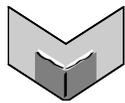


The Journal of Machine Learning Research
Volume 17
Print-Archive Edition

Issues 121–239



Microtome Publishing
Brookline, Massachusetts
www.mtome.com

The Journal of Machine Learning Research
Volume 17
Print-Archive Edition

The Journal of Machine Learning Research (JMLR) is an open access journal. All articles published in JMLR are freely available via electronic distribution. This Print-Archive Edition is published annually as a means of archiving the contents of the journal in perpetuity. The contents of this volume are articles published electronically in JMLR in 2016.

JMLR is abstracted in ACM Computing Reviews, INSPEC, and Psychological Abstracts/PsycINFO.

JMLR is a publication of Journal of Machine Learning Research, Inc. For further information regarding JMLR, including open access to articles, visit <http://www.jmlr.org/>.

JMLR Print-Archive Edition is a publication of Microtome Publishing under agreement with Journal of Machine Learning Research, Inc. For further information regarding the Print-Archive Edition, including subscription and distribution information and background on open-access print archiving, visit Microtome Publishing at <http://www.mtome.com/>.

Collection copyright © 2016 The Journal of Machine Learning Research, Inc. and Microtome Publishing. Copyright of individual articles remains with their respective authors.

ISSN 1532-4435 (print)
ISSN 1533-7928 (online)

JMLR Editorial Board

Editor-in-Chief

Bernhard Schölkopf, MPI for Intelligent Systems, Germany

Editor-in-Chief

Kevin Murphy, Google Research, USA

Managing Editor

Aron Culotta, Illinois Institute of Technology, USA

Production Editor

Charles Sutton, University of Edinburgh, UK

JMLR Web Master

Chiyuan Zhang, Massachusetts Institute of Technology, USA

JMLR Action Editors

Edoardo M. Airoldi, Harvard University, USA **Peter Auer**, University of Leoben, Austria **Francis Bach**, INRIA, France **Andrew Bagnell**, Carnegie Mellon University, USA **David Barber**, University College London, UK **Mikhail Belkin**, Ohio State University, USA **Yoshua Bengio**, Université de Montréal, Canada **Samy Bengio**, Google Research, USA **Jeff Bilmes**, University of Washington, USA **David Blei**, Princeton University, USA **Karsten Borgwardt**, MPI For Intelligent systems, Germany **Léon Bottou**, Microsoft Research, USA **Michael Bowling**, University of Alberta, Canada **Lawrence Carin**, Duke University, USA **Francois Caron**, University of Bordeaux, France **David Maxwell Chickering**, Microsoft Research, USA **Andreas Christmann**, University of Bayreuth, Germany **Alexander Clark**, King's College London, UK **William W. Cohen**, Carnegie-Mellon University, USA **Corinna Cortes**, Google Research, USA **Koby Crammer**, Technion, Israel **Sanjoy Dasgupta**, University of California, San Diego, USA **Rina Dechter**, University of California, Irvine, USA **Inderjit S. Dhillon**, University of Texas, Austin, USA **David Dunson**, Duke University, USA **Charles Elkan**, University of California at San Diego, USA **Rob Fergus**, New York University, USA **Nando de Freitas**, Oxford University, UK **Kenji Fukumizu**, The Institute of Statistical Mathematics, Japan **Sara van de Geer**, ETH Zürich, Switzerland **Amir Globerson**, The Hebrew University of Jerusalem, Israel **Moises Goldszmidt**, Microsoft Research, USA **Russ Greiner**, University of Alberta, Canada **Arthur Gretton**, University College London, UK **Maya Gupta**, Google Research, USA **Isabelle Guyon**, ClopiNet, USA **Moritz Hardt**, Google Research, USA **Matthias Hein**, Saarland University, Germany **Thomas Hofmann**, ETH Zurich, Switzerland **Bert Huang**, Virginia Tech, Virginia **Aapo Hyvärinen**, University of Helsinki, Finland **Alex Ihler**, University of California, Irvine, USA **Tommi Jaakkola**, Massachusetts Institute of Technology, USA **Samuel Kaski**, Aalto University, Finland **Sathiya Keerthi**, Microsoft Research, USA **Andreas Krause**, ETH Zurich, Switzerland **Christoph Lampert**, Institute of Science and Technology, Austria **Gert Lanckriet**, University of California, San Diego, USA **Pavel Laskov**, University of Tübingen, Germany **Neil Lawrence**, University of Sheffield, UK **Guy Lebanon**, LinkedIn, USA **Daniel Lee**, University of Pennsylvania, USA **Jure Leskovec**, Stanford University, USA **Qiang Liu**, Dartmouth College, USA **Gábor Lugosi**, Pompeu Fabra University, Spain **Ulrike von Luxburg**, University of Hamburg, Germany **Shie Mannor**, Technion, Israel **Robert E. McCulloch**, University of Chicago, USA **Chris Meek**, Microsoft Research, USA **Nicolai Meinshausen**, University of Oxford, UK **Vahab Mirrokni**, Google Research, USA **Mehryar Mohri**, New

York University, USA **Sebastian Nowozin**, Microsoft Research, Cambridge, UK **Una-May O'Reilly**, Massachusetts Institute of Technology, USA **Laurent Orseau**, Google Deepmind, USA **Manfred Opper**, Technical University of Berlin, Germany **Martin Pelikan**, Google Inc, USA **Jie Peng**, University of California, Davis, USA **Jan Peters**, Technische Universitaet Darmstadt, Germany **Avi Pfeffer**, Charles River Analytics, USA **Joelle Pineau**, McGill University, Canada **Massimiliano Pontil**, University College London, UK **Yuan (Alan) Qi**, Purdue University, USA **Luc de Raedt**, Katholieke Universiteit Leuven, Belgium **Alexander Rakhlin**, University of Pennsylvania, USA **Ben Recht**, University of California, Berkeley, USA **Saharon Rosset**, Tel Aviv University, Israel **Ruslan Salakhutdinov**, University of Toronto, Canada **Sujay Sanghavi**, University of Texas, Austin, USA **Marc Schoenauer**, INRIA Saclay, France **Matthias Seeger**, Amazon, Germany **John Shawe-Taylor**, University College London, UK **Xi-aotong Shen**, University of Minnesota, USA **Yoram Singer**, Google Research, USA **David Sontag**, New York University, USA **Peter Spirtes**, Carnegie Mellon University, USA **Nathan Srebro**, Toyota Technical Institute at Chicago, USA **Ingo Steinwart**, University of Stuttgart, Germany **Amos Storkey**, University of Edinburgh, UK **Csaba Szepesvari**, University of Alberta, Canada **Yee Whye Teh**, University of Oxford, UK **Olivier Teytaud**, INRIA Saclay, France **Ivan Titov**, University of Amsterdam, Netherlands **Koji Tsuda**, National Institute of Advanced Industrial Science and Technology, Japan **Zhuowen Tu**, University of California at San Diego, USA **Nicolas Vayatis**, Ecole Normale Supérieure de Cachan, France **S V N Vishwanathan**, Purdue University, USA **Manfred Warmuth**, University of California at Santa Cruz, USA **Stefan Wrobel**, Fraunhofer IAIS and University of Bonn, Germany **Eric Xing**, Carnegie Mellon University, USA **Bin Yu**, University of California at Berkeley, USA **Tong Zhang**, Rutgers University, USA **Zhihua Zhang**, Shanghai Jiao Tong University, China **Hui Zou**, University of Minnesota, USA

JMLR MLOSS Editors

Geoffrey Holmes, University of Waikato, New Zealand **Antti Honkela**, University of Helsinki, Finland **Balázs Kégl**, University of Paris-Sud, France **Cheng Soon Ong**, University of Melbourne, Australia **Mark Reid**, Australian National University, Australia

JMLR Editorial Board

Naoki Abe, IBM TJ Watson Research Center, USA **Yasemin Altun**, Google Inc, Switzerland **Jean-Yves Audibert**, CERTIS, France **Jonathan Baxter**, Australian National University, Australia **Richard K. Belew**, University of California at San Diego, USA **Kristin Bennett**, Rensselaer Polytechnic Institute, USA **Christopher M. Bishop**, Microsoft Research, Cambridge, UK **Lashon Booker**, The Mitre Corporation, USA **Henrik Boström**, Stockholm University/KTH, Sweden **Craig Boutilier**, Google Research, USA **Nello Cristianini**, University of Bristol, UK **Peter Dayan**, University College, London, UK **Dennis DeCoste**, eBay Research, USA **Thomas Dietterich**, Oregon State University, USA **Jennifer Dy**, Northeastern University, USA **Saso Dzeroski**, Jozef Stefan Institute, Slovenia **Ran El-Yaniv**, Technion, Israel **Peter Flach**, Bristol University, UK **Emily Fox**, University of Washington, USA **Dan Geiger**, Technion, Israel **Claudio Gentile**, Università degli Studi dell'Insubria, Italy **Sally Goldman**, Google Research, USA **Thore Graepel**, Microsoft Research, UK **Tom Griffiths**, University of California at Berkeley, USA **Carlos Guestrin**, University of Washington, USA **Stefan Harmeling**, University of Düsseldorf, Germany **David Heckerman**, Microsoft Research, USA **Katherine Heller**, Duke University, USA **Philipp Hennig**, MPI for Intelligent Systems, Germany **Larry Hunter**, University of Colorado, USA **Risi Kondor**, University of Chicago, USA **Aryeh Kontorovich**, Ben-Gurion University of

the Negev, Israel **Samory Kpotufe**, Princeton University, USA **Andreas Krause**, ETH Zürich, Switzerland **John Lafferty**, University of Chicago, USA **Erik Learned-Miller**, University of Massachusetts, Amherst, USA **Fei Fei Li**, Stanford University, USA **Yi Lin**, University of Wisconsin, USA **Wei-Yin Loh**, University of Wisconsin, USA **Richard Maclin**, University of Minnesota, USA **Sridhar Mahadevan**, University of Massachusetts, Amherst, USA **Michael W Mahoney**, University of California at Berkeley, USA **Vikash Mansinghka**, Massachusetts Institute of Technology, USA **Yishay Mansour**, Tel-Aviv University, Israel **Jon McAuliffe**, University of California, Berkeley, USA **Andrew McCallum**, University of Massachusetts, Amherst, USA **Joris Mooij**, Radboud University Nijmegen, Netherlands **Raymond J. Mooney**, University of Texas, Austin, USA **Klaus-Robert Muller**, Technical University of Berlin, Germany **Guillaume Obozinski**, Ecole des Ponts - ParisTech, France **Pascal Poupart**, University of Waterloo, Canada **Konrad Rieck**, University of Göttingen, Germany **Cynthia Rudin**, Massachusetts Institute of Technology, USA **Robert Schapire**, Princeton University, USA **Mark Schmidt**, University of British Columbia, Canada **Fei Sha**, University of Southern California, USA **Shai Shalev-Shwartz**, Hebrew University of Jerusalem, Israel **Padhraic Smyth**, University of California, Irvine, USA **Le Song**, Georgia Institute of Technology, USA **Bharath Sriperumbudur**, Pennsylvania State University, USA **Alexander Statnikov**, New York University, USA **Jean-Philippe Vert**, Mines ParisTech, France **Martin J. Wainwright**, University of California at Berkeley, USA **Chris Watkins**, Royal Holloway, University of London, UK **Kilian Weinberger**, Washington University, St Louis, USA **Max Welling**, University of Amsterdam, Netherlands **Chris Williams**, University of Edinburgh, UK **David Wipf**, Microsoft Research Asia, China **Alice Zheng**, GraphLab, USA

JMLR Advisory Board

Shun-Ichi Amari, RIKEN Brain Science Institute, Japan **Andrew Barto**, University of Massachusetts at Amherst, USA **Thomas Dietterich**, Oregon State University, USA **Jerome Friedman**, Stanford University, USA **Stuart Geman**, Brown University, USA **Geoffrey Hinton**, University of Toronto, Canada **Michael Jordan**, University of California at Berkeley at USA **Leslie Pack Kaelbling**, Massachusetts Institute of Technology, USA **Michael Kearns**, University of Pennsylvania, USA **Steven Minton**, InferLink, USA **Tom Mitchell**, Carnegie Mellon University, USA **Stephen Muggleton**, Imperial College London, UK **Nils Nilsson**, Stanford University, USA **Tomaso Poggio**, Massachusetts Institute of Technology, USA **Ross Quinlan**, Rulequest Research Pty Ltd, Australia **Stuart Russell**, University of California at Berkeley, USA **Lawrence Saul**, University of California at San Diego, USA **Terrence Sejnowski**, Salk Institute for Biological Studies, USA **Richard Sutton**, University of Alberta, Canada **Leslie Valiant**, Harvard University, USA

Journal of Machine Learning Research

Volume 17, 2017

Part A

- 17(1):1–42 **On the Complexity of Best-Arm Identification in Multi-Armed Bandit Models**
Emilie Kaufmann, Olivier Cappé, Aurélien Garivier
- 17(2):1–51 **Multiscale Dictionary Learning: Non-Asymptotic Bounds and Robustness**
Mauro Maggioni, Stanislav Minsker, Nate Strawn
- 17(3):1–32 **Consistent Algorithms for Clustering Time Series**
Azadeh Khaleghi, Daniil Ryabko, Jérémie Mary, Philippe Preux
- 17(4):1–26 **Random Rotation Ensembles**
Rico Blaser, Piotr Fryzlewicz
- 17(5):1–10 **Should We Really Use Post-Hoc Tests Based on Mean-Ranks?**
Alessio Benavoli, Giorgio Corani, Francesca Mangili
- 17(6):1–31 **Minimax Rates in Permutation Estimation for Feature Matching**
Olivier Collier, Arnak S. Dalalyan
- 17(7):1–33 **Consistency and Fluctuations For Stochastic Gradient Langevin Dynamics**
Yee Whye Teh, Alexandre H. Thiery, Sebastian J. Vollmer
- 17(8):1–32 **Knowledge Matters: Importance of Prior Information for Optimization**
Çağlar Gülçehre, Yoshua Bengio
- 17(9):1–5 **Harry: A Tool for Measuring String Similarity**
Konrad Rieck, Christian Wressnegger
- 17(10):1–29 **Herded Gibbs Sampling**
Yutian Chen, Luke Bornn, Nando de Freitas, Mareija Eskelin, Jing Fang, Max Welling
- 17(11):1–28 **Complexity of Representation and Inference in Compositional Models with Part Sharing**
Alan Yuille, Roozbeh Mottaghi
- 17(12):1–41 **Noisy Sparse Subspace Clustering**
Yu-Xiang Wang, Huan Xu
- 17(13):1–36 **Learning the Variance of the Reward-To-Go**
Aviv Tamar, Dotan Di Castro, Shie Mannor

- 17(14):1–45 **Convex Calibration Dimension for Multiclass Loss Matrices**
Harish G. Ramaswamy, Shivani Agarwal
- 17(15):1–24 **LLORMA: Local Low-Rank Matrix Approximation**
Joonseok Lee, Seungyeon Kim, Guy Lebanon, Yoram Singer, Samy Bengio
- 17(16):1–26 **A Consistent Information Criterion for Support Vector Machines in Diverging Model Spaces**
Xiang Zhang, Yichao Wu, Lan Wang, Runze Li
- 17(17):1–51 **Extremal Mechanisms for Local Differential Privacy**
Peter Kairouz, Sewoong Oh, Pramod Viswanath
- 17(18):1–40 **Loss Minimization and Parameter Estimation with Heavy Tails**
Daniel Hsu, Sivan Sabato
- 17(19):1–30 **Analysis of Classification-based Policy Iteration Algorithms**
Alessandro Lazaric, Mohammad Ghavamzadeh, Rémi Munos
- 17(20):1–54 **Operator-valued Kernels for Learning from Functional Response Data**
Hachem Kadri, Emmanuel Duflos, Philippe Preux, Stéphane Canu, Alain Rakotomamonjy, Julien Audiffren
- 17(21):1–5 **MEKA: A Multi-label/Multi-target Extension to WEKA**
Jesse Read, Peter Reutemann, Bernhard Pfahringer, Geoff Holmes
- 17(22):1–34 **Gradients Weights improve Regression and Classification**
Samory Kpotufe, Abdeslam Boularias, Thomas Schultz, Kyoungok Kim
- 17(23):1–21 **A Closer Look at Adaptive Regret**
Dmitry Adamskiy, Wouter M. Koolen, Alexey Chernov, Vladimir Vovk
- 17(24):1–42 **Learning Using Anti-Training with Sacrificial Data**
Michael L. Valenzuela, Jerzy W. Rozenblit
- 17(25):1–72 **A Unifying Framework in Vector-valued Reproducing Kernel Hilbert Spaces for Manifold Regularization and Co-Regularized Multi-view Learning**
Hà Quang Minh, Loris Bazzani, Vittorio Murino
- 17(26):1–41 **Quantifying Uncertainty in Random Forests via Confidence Intervals and Hypothesis Tests**
Lucas Mentch, Giles Hooker
- 17(27):1–57 **Statistical-Computational Tradeoffs in Planted Problems and Submatrix Localization with a Growing Number of Clusters and Submatrices**
Yudong Chen, Jiaming Xu

- 17(28):1–39 **Non-linear Causal Inference using Gaussianity Measures**
Daniel Hernández-Lobato, Pablo Morales-Mombiela, David Lopez-Paz, Alberto Suárez
- 17(29):1–54 **Consistent Distribution-Free K -Sample and Independence Tests for Univariate Random Variables**
Ruth Heller, Yair Heller, Shachar Kaufman, Barak Brill, Malka Gorfine
- 17(30):1–39 **A Gibbs Sampler for Learning DAGs**
Robert J. B. Goudie, Sach Mukherjee
- 17(31):1–32 **Dimension-free Concentration Bounds on Hankel Matrices for Spectral Learning**
Francois Denis, Mattias Gybels, Amaury Habrard
- 17(32):1–102 **Distinguishing Cause from Effect Using Observational Data: Methods and Benchmarks**
Joris M. Mooij, Jonas Peters, Dominik Janzing, Jakob Zscheischler, Bernhard Schölkopf
- 17(33):1–30 **Multi-task Sparse Structure Learning with Gaussian Copula Models**
André R. Goncalves, Fernando J. Von Zuben, Arindam Banerjee
- 17(34):1–7 **MLlib: Machine Learning in Apache Spark**
Xiangrui Meng, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, Davies Liu, Jeremy Freeman, DB Tsai, Manish Amde, Sean Owen, Doris Xin, Reynold Xin, Michael J. Franklin, Reza Zadeh, Matei Zaharia, Ameet Talwalkar
- 17(35):1–5 **OLPS: A Toolbox for On-Line Portfolio Selection**
Bin Li, Doyen Sahoo, Steven C.H. Hoi
- 17(36):1–39 **A Bounded p -norm Approximation of Max-Convolution for Sub-Quadratic Bayesian Inference on Additive Factors**
Julianus Pfeuffer, Oliver Serang
- 17(37):1–33 **Hybrid Orthogonal Projection and Estimation (HOPE): A New Framework to Learn Neural Networks**
Shiliang Zhang, Hui Jiang, Lirong Dai
- 17(38):1–15 **The Optimal Sample Complexity of PAC Learning**
Steve Hanneke
- 17(39):1–40 **End-to-End Training of Deep Visuomotor Policies**
Sergey Levine, Chelsea Finn, Trevor Darrell, Pieter Abbeel
- 17(40):1–45 **On Quantile Regression in Reproducing Kernel Hilbert Spaces with the Data Sparsity Constraint**
Chong Zhang, Yufeng Liu, Yichao Wu
- 17(41):1–6 **BayesPy: Variational Bayesian Inference in Python**
Jaakko Luttinen

- 17(42):1–62 **Variational Inference for Latent Variables and Uncertain Inputs in Gaussian Processes**
Andreas C. Damianou, Michalis K. Titsias, Neil D. Lawrence
- 17(43):1–28 **On the Estimation of the Gradient Lines of a Density and the Consistency of the Mean-Shift Algorithm**
Ery Arias-Castro, David Mason, Bruno Pelletier
- 17(44):1–35 **Scalable Learning of Bayesian Network Classifiers**
Ana M. Martínez, Geoffrey I. Webb, Shenglei Chen, Nayyar A. Zaidi
- 17(45):1–32 **A Unified View on Multi-class Support Vector Classification**
Ürün Doğan, Tobias Glasmachers, Christian Igel
- 17(46):1–31 **Addressing Environment Non-Stationarity by Repeating Q-learning Updates**
Sherief Abdallah, Michael Kaisers
- 17(47):1–43 **Large Scale Online Kernel Learning**
Jing Lu, Steven C.H. Hoi, Jialei Wang, Peilin Zhao, Zhi-Yong Liu
- 17(48):1–41 **Kernel Mean Shrinkage Estimators**
Krikamol Muandet, Bharath Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf
- 17(49):1–49 **SPSD Matrix Approximation via Column Selection: Theories, Algorithms, and Extensions**
Shusen Wang, Luo Luo, Zhihua Zhang
- 17(50):1–33 **Combinatorial Multi-Armed Bandit and Its Extension to Probabilistically Triggered Arms**
Wei Chen, Yajun Wang, Yang Yuan, Qinshi Wang
- 17(51):1–42 **Differentially Private Data Releasing for Smooth Queries**
Ziteng Wang, Chi Jin, Kai Fan, Jiaqi Zhang, Junliang Huang, Yiqiao Zhong, Liwei Wang
- 17(52):1–21 **Subspace Learning with Partial Information**
Alon Gonen, Dan Rosenbaum, Yonina C. Eldar, Shai Shalev-Shwartz
- 17(53):1–38 **Iterative Hessian Sketch: Fast and Accurate Solution Approximation for Constrained Least-Squares**
Mert Pilanci, Martin J. Wainwright
- 17(54):1–23 **Estimating Causal Structure Using Conditional DAG Models**
Chris. J. Oates, Jim Q. Smith, Sach Mukherjee
- 17(55):1–46 **Adaptive Lasso and group-Lasso for functional Poisson regression**
Stéphane Ivanoff, Franck Picard, Vincent Rivoirard
- 17(56):1–53 **Causal Inference through a Witness Protection Program**
Ricardo Silva, Robin Evans

- 17(57):1–47 **Structure Discovery in Bayesian Networks by Sampling Partial Orders**
Teppo Niinimäki, Pekka Parviainen, Mikko Koivisto
- 17(58):1–47 **Estimation from Pairwise Comparisons: Sharp Minimax Bounds with Topology Dependence**
Nihar B. Shah, Sivaraman Balakrishnan, Joseph Bradley, Abhay Parekh, Kannan Ramchandran, Martin J. Wainwright
- 17(59):1–35 **Domain-Adversarial Training of Neural Networks**
Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, Victor Lempitsky
- 17(60):1–34 **Probabilistic Low-Rank Matrix Completion from Quantized Measurements**
Sonia A. Bhaskar
- 17(61):1–35 **DSA: Decentralized Double Stochastic Averaging Gradient Algorithm**
Aryan Mokhtari, Alejandro Ribeiro
- 17(62):1–29 **The Statistical Performance of Collaborative Inference**
Gérard Biau, Kevin Bleakley, Benoît Cadre
- 17(63):1–53 **Convergence of an Alternating Maximization Procedure**
Andreas Andreassen, Vladimir Spokoiny
- 17(64):1–5 **StructED: Risk Minimization in Structured Prediction**
Yossi Adi, Joseph Keshet
- 17(65):1–32 **Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches**
Jure Žbontar, Yann LeCun
- 17(66):1–53 **Bayesian Policy Gradient and Actor-Critic Algorithms**
Mohammad Ghavamzadeh, Yaakov Engel, Michal Valko
- 17(67):1–70 **Practical Kernel-Based Reinforcement Learning**
André M.S. Barreto, Doina Precup, Joelle Pineau
- 17(68):1–30 **An Information-Theoretic Analysis of Thompson Sampling**
Daniel Russo, Benjamin Van Roy
- 17(69):1–26 **Compressed Gaussian Process for Manifold Regression**
Rajarshi Guhaniyogi, David B. Dunson
- 17(70):1–32 **On the Characterization of a Class of Fisher-Consistent Loss Functions and its Application to Boosting**
Matey Neykov, Jun S. Liu, Tianxi Cai
- 17(71):1–19 **Exact Inference on Gaussian Graphical Models of Arbitrary Topology using Path-Sums**
P.-L. Giscard, Z. Choo, S. J. Thwaite, D. Jaksch

- 17(72):1–54 **Challenges in multimodal gesture recognition**
Sergio Escalera, Vassilis Athitsos, Isabelle Guyon
- 17(73):1–29 **An Emphatic Approach to the Problem of Off-policy Temporal-Difference Learning**
Richard S. Sutton, A. Rupam Mahmood, Martha White
- 17(74):1–25 **Learning Algorithms for Second-Price Auctions with Reserve**
Mehryar Mohri, Andres Munoz Medina
- 17(75):1–25 **Distributed Coordinate Descent Method for Learning with Big Data**
Peter Richtárik, Martin Takáč
- 17(76):1–36 **Scaling-up Empirical Risk Minimization: Optimization of Incomplete U -statistics**
Stephan Cléménçon, Igor Colin, Aurélien Bellet
- 17(77):1–38 **Iterative Regularization for Learning with Convex Loss Functions**
Junhong Lin, Lorenzo Rosasco, Ding-Xuan Zhou
- 17(78):1–41 **Latent Space Inference of Internet-Scale Networks**
Qirong Ho, Junming Yin, Eric P. Xing
- 17(79):1–23 **Patient Risk Stratification with Time-Varying Parameters: A Multitask Learning Approach**
Jenna Wiens, John Guttag, Eric Horvitz
- 17(80):1–33 **Multiplicative Multitask Feature Learning**
Xin Wang, Jinbo Bi, Shipeng Yu, Jiangwen Sun, Minghu Song
- 17(81):1–32 **The Benefit of Multitask Representation Learning**
Andreas Maurer, Massimiliano Pontil, Bernardino Romera-Paredes
- 17(82):1–24 **Model-free Variable Selection in Reproducing Kernel Hilbert Space**
Lei Yang, Shaogao Lv, Junhui Wang
- 17(83):1–5 **CVXPY: A Python-Embedded Modeling Language for Convex Optimization**
Steven Diamond, Stephen Boyd
- 17(84):1–42 **Lenient Learning in Independent-Learner Stochastic Cooperative Games**
Ermo Wei, Sean Luke
- 17(85):1–15 **Structure-Leveraged Methods in Breast Cancer Risk Prediction**
Jun Fan, Yirong Wu, Ming Yuan, David Page, Jie Liu, Irene M. Ong, Peggy Peissig, Elizabeth Burnside

- 17(86):1–5 **LIBMF: A Library for Parallel Matrix Factorization in Shared-memory Systems**
Wei-Sheng Chin, Bo-Wen Yuan, Meng-Yuan Yang, Yong Zhuang, Yu-Chin Juan, Chih-Jen Lin
- 17(87):1–37 **L1-Regularized Least Squares for Support Recovery of High Dimensional Single Index Models with Gaussian Designs**
Matey Neykov, Jun S. Liu, Tianxi Cai
- 17(88):1–45 **Spectral Ranking using Seriation**
Fajwel Fogel, Alexandre d'Aspremont, Milan Vojnovic
- 17(89):1–34 **Sparsity and Error Analysis of Empirical Feature-Based Regularization Schemes**
Xin Guo, Jun Fan, Ding-Xuan Zhou
- 17(90):1–29 **Estimating Diffusion Networks: Recovery Conditions, Sample Complexity and Soft-thresholding Algorithm**
Manuel Gomez-Rodriguez, Le Song, Hadi Daneshmand, Bernhard Schölkopf
- 17(91):1–42 **Rounding-based Moves for Semi-Metric Labeling**
M. Pawan Kumar, Puneet K. Dokania
- 17(92):1–27 **Rate Optimal Denoising of Simultaneously Sparse and Low Rank Matrices**
Dan Yang, Zongming Ma, Andreas Buja
- 17(93):1–50 **Hierarchical Relative Entropy Policy Search**
Christian Daniel, Gerhard Neumann, Oliver Kroemer, Jan Peters
- 17(94):1–31 **Convex Regression with Interpretable Sharp Partitions**
Ashley Petersen, Noah Simon, Daniela Witten
- 17(95):1–5 **JCLAL: A Java Framework for Active Learning**
Oscar Reyes, Eduardo Pérez, María del Carmen Rodríguez-Hernández, Habib M. Fardoun, Sebastián Ventura
- 17(96):1–37 **Integrated Common Sense Learning and Planning in POMDPs**
Brendan Juba
- 17(97):1–37 **Cells in Multidimensional Recurrent Neural Networks**
Gundram Leifert, Tobias Strauß, Tobias Grüning, Welf Wustlich, Roger Labahn
- 17(98):1–37 **Learning Taxonomy Adaptation in Large-scale Classification**
Rohit Babbar, Ioannis Partalas, Eric Gaussier, Massih-Reza Amini, Cécile Amblard
- 17(99):1–61 **How to Center Deep Boltzmann Machines**
Jan Melchior, Asja Fischer, Laurenz Wiskott

- 17(100):1–35 **Control Function Instrumental Variable Estimation of Nonlinear Causal Effect Models**
Zijian Guo, Dylan S. Small
- 17(101):1–54 **Structure Learning in Bayesian Networks of a Moderate Size by Efficient Sampling**
Ru He, Jin Tian, Huaiqing Wu
- 17(102):1–44 **Spectral Methods Meet EM: A Provably Optimal Algorithm for Crowdsourcing**
Yuchen Zhang, Xi Chen, Dengyong Zhou, Michael I. Jordan
- 17(103):1–38 **Bayesian Leave-One-Out Cross-Validation Approximations for Gaussian Latent Variable Models**
Aki Vehtari, Tommi Mononen, Ville Tolvanen, Tuomas Sivula, Ole Winther
- 17(104):1–32 **e-PAL: An Active Learning Approach to the Multi-Objective Optimization Problem**
Marcela Zuluaga, Andreas Krause, Markus Püschel
- 17(105):1–41 **Trend Filtering on Graphs**
Yu-Xiang Wang, James Sharpnack, Alexander J. Smola, Ryan J. Tibshirani
- 17(106):1–37 **Multi-Task Learning for Straggler Avoiding Predictive Job Scheduling**
Neeraja J. Yadwadkar, Bharath Hariharan, Joseph E. Gonzalez, Randy Katz
- 17(107):1–31 **Interleaved Text/Image Deep Mining on a Large-Scale Radiology Database for Automated Image Interpretation**
Hoo-Chang Shin, Le Lu, Lauren Kim, Ari Seff, Jianhua Yao, Ronald M. Summers
- 17(108):1–30 **Distribution-Matching Embedding for Visual Domain Adaptation**
Mahsa Baktashmotlagh, Mehrtash Harandi, Mathieu Salzmann
- 17(109):1–47 **Monotonic Calibrated Interpolated Look-Up Tables**
Maya Gupta, Andrew Cotter, Jan Pfeifer, Konstantin Voevodski, Kevin Canini, Alexander Mangylov, Wojciech Moczydlowski, Alexander van Esbroeck
- 17(110):1–5 **Are Random Forests Truly the Best Classifiers?**
Michael Wainberg, Babak Alipanahi, Brendan J. Frey
- 17(111):1–43 **Minimax Adaptive Estimation of Nonparametric Hidden Markov Models**
Yohann De Castro, Élisabeth Gassiat, Claire Lacour

- 17(112):1–30 **Decrypting “Cryptogenic” Epilepsy: Semi-supervised Hierarchical Conditional Random Fields For Detecting Cortical Lesions In MRI-Negative Patients**
Bilal Ahmed, Thomas Thesen, Karen E. Blackmon, Ruben Kuzniecky, Orrin Devinsky, Carla E. Brodley
- 17(113):1–23 **Fused Lasso Approach in Regression Coefficients Clustering – Learning Parameter Heterogeneity in Data Integration**
Lu Tang, Peter X.K. Song
- 17(114):1–5 **The LRP Toolbox for Artificial Neural Networks**
Sebastian Lapuschkin, Alexander Binder, Grégoire Montavon, Klaus-Robert Müller, Wojciech Samek
- 17(115):1–21 **Equivalence of Graphical Lasso and Thresholding for Sparse Graphs**
Somayeh Sojoudi
- 17(116):1–13 **A Network That Learns Strassen Multiplication**
Veit Elser
- 17(117):1–65 **Revisiting the Nystrom Method for Improved Large-scale Machine Learning**
Alex Gittens, Michael W. Mahoney
- 17(118):1–20 **Improving Structure MCMC for Bayesian Networks through Markov Blanket Resampling**
Chengwei Su, Mark E. Borsuk
- 17(119):1–34 **Volumetric Spanners: An Efficient Exploration Basis for Learning**
Elad Hazan, Zohar Karnin
- 17(120):1–38 **Quasi-Monte Carlo Feature Maps for Shift-Invariant Kernels**
Haim Avron, Vikas Sindhwani, Jiyang Yang, Michael W. Mahoney

Part B

- 17(121):1–36 **Variational Dependent Multi-output Gaussian Process Dynamical Systems**
Jing Zhao, Shiliang Sun
- 17(122):1–35 **Multiple Output Regression with Latent Noise**
Jussi Gillberg, Pekka Marttinen, Matti Pirinen, Antti J. Kangas, Pasi Soinen, Mehreen Ali, Aki S. Havulinna, Marjo-Riitta Järvelin, Mika Ala-Korpela, Samuel Kaski
- 17(123):1–22 **The Constrained Dantzig Selector with Enhanced Consistency**
Yinfei Kong, Zemin Zheng, Jinchi Lv
- 17(124):1–29 **Bootstrap-Based Regularization for Low-Rank Matrix Estimation**
Julie Josse, Stefan Wager

- 17(125):1–47 **Bayesian Optimization for Likelihood-Free Inference of Simulator-Based Statistical Models**
Michael U. Gutmann, Jukka Corander
- 17(126):1–51 **On Lower and Upper Bounds in Smooth and Strongly Convex Optimization**
Yossi Arjevani, Shai Shalev-Shwartz, Ohad Shamir
- 17(127):1–30 **Dual Control for Approximate Bayesian Reinforcement Learning**
Edgar D. Klenske, Philipp Hennig
- 17(128):1–50 **Multiple-Instance Learning from Distributions**
Gary Doran, Soumya Ray
- 17(129):1–23 **An Online Convex Optimization Approach to Blackwell’s Approachability**
Nahum Shimkin
- 17(130):1–28 **A Well-Conditioned and Sparse Estimation of Covariance and Inverse Covariance Matrices Using a Joint Penalty**
Ashwini Maurya
- 17(131):1–87 **String and Membrane Gaussian Processes**
Yves-Laurent Kom Samo, Stephen J. Roberts
- 17(132):1–25 **Extracting PICO Sentences from Clinical Trial Reports using Supervised Distant Supervision**
Byron C. Wallace, Joël Kuiper, Aakash Sharma, Mingxi (Brian) Zhu, Iain J. Marshall
- 17(133):1–38 **Cross-Corpora Unsupervised Learning of Trajectories in Autism Spectrum Disorders**
Huseyin Melih Elibol, Vincent Nguyen, Scott Linderman, Matthew Johnson, Anna Hashmi, Finale Doshi-Velez
- 17(134):1–32 **Adjusting for Chance Clustering Comparison Measures**
Simone Romano, Nguyen Xuan Vinh, James Bailey, Karin Verspoor
- 17(135):1–55 **Refined Error Bounds for Several Learning Algorithms**
Steve Hanneke
- 17(136):1–33 **Synergy of Monotonic Rules**
Vladimir Vapnik, Rauf Izmailov
- 17(137):1–5 **Pymanopt: A Python Toolbox for Optimization on Manifolds using Automatic Differentiation**
James Townsend, Niklas Koep, Sebastian Weichwald
- 17(138):1–49 **CrossCat: A Fully Bayesian Nonparametric Method for Analyzing Heterogeneous, High Dimensional Data**
Vikash Mansinghka, Patrick Shafto, Eric Jonas, Cap Petschulat, Max Gasner, Joshua B. Tenenbaum

- 17(139):1–66 **Regularized Policy Iteration with Nonparametric Function Spaces**
Amir-massoud Farahmand, Mohammad Ghavamzadeh, Csaba Szepesvári, Shie Mannor
- 17(140):1–38 **Multiscale Adaptive Representation of Signals: I. The Basic Framework**
Cheng Tai, Weinan E
- 17(141):1–41 **Sparse PCA via Covariance Thresholding**
Yash Deshpande, Andrea Montanari
- 17(142):1–31 **Large Scale Visual Recognition through Adaptation using Joint Representation and Multiple Instance Learning**
Judy Hoffman, Deepak Pathak, Eric Tzeng, Jonathan Long, Sergio Guadarrama, Trevor Darrell, Kate Saenko
- 17(143):1–21 **Covariance-based Clustering in Multivariate and Functional Data Analysis**
Francesca Ieva, Anna Maria Paganoni, Nicholas Tarabelloni
- 17(144):1–51 **MOCCA: Mirrored Convex/Concave Optimization for Nonconvex Composite Functions**
Rina Foygel Barber, Emil Y. Sidky
- 17(145):1–40 **True Online Temporal-Difference Learning**
Harm van Seijen, A. Rupam Mahmood, Patrick M. Pilarski, Marlos C. Machado, Richard S. Sutton
- 17(146):1–51 **Penalized Maximum Likelihood Estimation of Multi-layered Gaussian Graphical Models**
Jiahe Lin, Sumanta Basu, Moulinath Banerjee, George Michailidis
- 17(147):1–28 **Local Network Community Detection with Continuous Optimization of Conductance and Weighted Kernel K-Means**
Twan van Laarhoven, Elena Marchiori
- 17(148):1–5 **Megaman: Scalable Manifold Learning in Python**
James McQueen, Marina Meilă, Jacob VanderPlas, Zhongyue Zhang
- 17(149):1–37 **Kernel Estimation and Model Combination in A Bandit Problem with Covariates**
Wei Qian, Yuhong Yang
- 17(150):1–34 **A General Framework for Consistency of Principal Component Analysis**
Dan Shen, Haipeng Shen, J. S. Marron
- 17(151):1–20 **Conditional Independencies under the Algorithmic Independence of Conditionals**
Jan Lemeire

- 17(152):1–40 **Learning Theory for Distribution Regression**
Zoltán Szabó, Bharath K. Sriperumbudur, Barnabás Póczos, Arthur Gretton
- 17(153):1–43 **A Differential Equation for Modeling Nesterov’s Accelerated Gradient Method: Theory and Insights**
Weijie Su, Stephen Boyd, Emmanuel J. Candès
- 17(154):1–21 **Importance Weighting Without Importance Weights: An Efficient Algorithm for Combinatorial Semi-Bandits**
Gergely Neu, Gábor Bartók
- 17(155):1–38 **New Perspectives on k-Support and Cluster Norms**
Andrew M. McDonald, Massimiliano Pontil, Dimitris Stamos
- 17(156):1–33 **Minimum Density Hyperplanes**
Nicos G. Pavlidis, David P. Hofmeyr, Sotiris K. Tasoulis
- 17(157):1–36 **Theoretical Analysis of the Optimal Free Responses of Graph-Based SFA for the Design of Training Graphs**
Alberto N. Escalante-B., Laurenz Wiskott
- 17(158):1–21 **Universal Approximation Results for the Temporal Restricted Boltzmann Machine and the Recurrent Temporal Restricted Boltzmann Machine**
Simon Odense, Roderick Edwards
- 17(159):1–45 **Exploration of the (Non-)Asymptotic Bias and Variance of Stochastic Gradient Langevin Dynamics**
Sebastian J. Vollmer, Konstantinos C. Zygalakis, Yee Whye Teh
- 17(160):1–53 **A General Framework for Constrained Bayesian Optimization using Information-based Search**
José Miguel Hernández-Lobato, Michael A. Gelbart, Ryan P. Adams, Matthew W. Hoffman, Zoubin Ghahramani
- 17(161):1–29 **Optimal Estimation and Completion of Matrices with Biclustering Structures**
Chao Gao, Yu Lu, Zongming Ma, Harrison H. Zhou
- 17(162):1–25 **The Teaching Dimension of Linear Learners**
Ji Liu, Xiaojin Zhu
- 17(163):1–44 **Augmentable Gamma Belief Networks**
Mingyuan Zhou, Yulai Cong, Bo Chen
- 17(164):1–25 **Optimal Estimation of Derivatives in Nonparametric Regression**
Wenlin Dai, Tiejun Tong, Marc G. Genton
- 17(165):1–52 **Double or Nothing: Multiplicative Incentive Mechanisms for Crowdsourcing**
Nihar B. Shah, Dengyong Zhou

- 17(166):1–48 **Joint Structural Estimation of Multiple Graphical Models**
Jing Ma, George Michailidis
- 17(167):1–37 **Support Vector Hazards Machine: A Counting Process Framework for Learning Risk Scores for Censored Outcomes**
Yuanjia Wang, Tianle Chen, Donglin Zeng
- 17(168):1–36 **Stable Graphical Models**
Navodit Misra, Ercan E. Kuruoglu
- 17(169):1–40 **Bounding the Search Space for Global Optimization of Neural Networks Learning Error: An Interval Analysis Approach**
Stavros P. Adam, George D. Magoulas, Dimitrios A. Karras, Michael N. Vrahatis
- 17(170):1–5 **mlr: Machine Learning in R**
Bernd Bischl, Michel Lang, Lars Kotthoff, Julia Schiffner, Jakob Richter, Erich Studerus, Giuseppe Casalicchio, Zachary M. Jones
- 17(171):1–32 **Feature-Level Domain Adaptation**
Wouter M. Kouw, Laurens J.P. van der Maaten, Jesse H. Krijthe, Marco Loog
- 17(172):1–47 **Semiparametric Mean Field Variational Bayes: General Principles and Numerical Issues**
David Rohde, Matt P. Wand
- 17(173):1–49 **Online PCA with Optimal Regret**
Jiazhong Nie, Wojciech Kotlowski, Manfred K. Warmuth
- 17(174):1–29 **Efficient Computation of Gaussian Process Regression for Large Spatial Data Sets by Patching Local Gaussian Processes**
Chiwoo Park, Jianhua Z. Huang
- 17(175):1–5 **bandicoot: a Python Toolbox for Mobile Phone Metadata**
Yves-Alexandre de Montjoye, Luc Rocher, Alex Sandy Pentland
- 17(176):1–48 **Input Output Kernel Regression: Supervised and Semi-Supervised Structured Output Prediction with Operator-Valued Kernels**
Céline Brouard, Marie Szafranski, Florence d'Alché-Buc
- 17(177):1–18 **A Note on the Sample Complexity of the Er-SpUD Algorithm by Spielman, Wang and Wright for Exact Recovery of Sparsely Used Dictionaries**
Radoslaw Adamczak
- 17(178):1–35 **The Asymptotic Performance of Linear Echo State Neural Networks**
Romain Couillet, Gilles Wainrib, Harry Sevi, Hafiz Tiomoko Ali
- 17(179):1–34 **On the Consistency of the Likelihood Maximization Vertex Nomination Scheme: Bridging the Gap Between Maximum Likelihood Estimation and Graph Matching**
Vince Lyzinski, Keith Levin, Donniell E. Fishkind, Carey E. Priebe

- 17(180):1–28 **Characteristic Kernels and Infinitely Divisible Distributions**
Yu Nishiyama, Kenji Fukumizu
- 17(181):1–46 **Consistency of Cheeger and Ratio Graph Cuts**
Nicolás García Trillos, Dejan Slepčev, James von Brecht, Thomas Laurent, Xavier Bresson
- 17(182):1–39 **Jointly Informative Feature Selection Made Tractable by Gaussian Modeling**
Leonidas Lefakis, François Fleuret
- 17(183):1–40 **Learning with Differential Privacy: Stability, Learnability and the Sufficiency and Necessity of ERM Principle**
Yu-Xiang Wang, Jing Lei, Stephen E. Fienberg
- 17(184):1–5 **fastFM: A Library for Factorization Machines**
Immanuel Bayer
- 17(185):1–24 **The Factorized Self-Controlled Case Series Method: An Approach for Estimating the Effects of Many Drugs on Many Outcomes**
Ramin Moghaddass, Cynthia Rudin, David Madigan
- 17(186):1–32 **Electronic Health Record Analysis via Deep Poisson Factor Models**
Ricardo Henao, James T. Lu, Joseph E. Lucas, Jeffrey Ferranti, Lawrence Carin
- 17(187):1–25 **Low-Rank Doubly Stochastic Matrix Decomposition for Cluster Analysis**
Zhirong Yang, Jukka Corander, Erkki Oja
- 17(188):1–25 **A New Algorithm and Theory for Penalized Regression-based Clustering**
Chong Wu, Sunghoon Kwon, Xiaotong Shen, Wei Pan
- 17(189):1–40 **Classification of Imbalanced Data with a Geometric Digraph Family**
Artür Manukyan, Elvan Ceyhan
- 17(190):1–37 **A Variational Approach to Path Estimation and Parameter Inference of Hidden Diffusion Processes**
Tobias Sutter, Arnab Ganguly, Heinz Koeppl
- 17(191):1–21 **One-class classification of point patterns of extremes**
Stijn Luca, David A. Clifton, Bart Vanrumste
- 17(192):1–66 **On the Influence of Momentum Acceleration on Online Learning**
Kun Yuan, Bicheng Ying, Ali H. Sayed
- 17(193):1–54 **Data-driven Rank Breaking for Efficient Rank Aggregation**
Ashish Khetan, Sewoong Oh

- 17(194):1–44 **Optimal Learning Rates for Localized SVMs**
Mona Meister, Ingo Steinwart
- 17(195):1–102 **Bipartite Ranking: a Risk-Theoretic Perspective**
Aditya Krishna Menon, Robert C. Williamson
- 17(196):1–47 **Bayesian group factor analysis with structured sparsity**
Shiwen Zhao, Chuan Gao, Sayan Mukherjee, Barbara E Engelhardt
- 17(197):1–37 **Machine Learning in an Auction Environment**
Patrick Hummel, R. Preston McAfee
- 17(198):1–38 **Wavelet decompositions of Random Forests - smoothness analysis, sparse approximation and applications**
Oren Elisha, Shai Dekel
- 17(199):1–31 **Mutual Information Based Matching for Causal Inference with Observational Data**
Lei Sun, Alexander G. Nikolaev
- 17(200):1–51 **Online Trans-dimensional von Mises-Fisher Mixture Models for User Profiles**
Xiangju Qin, Pádraig Cunningham, Michael Salter-Townshend
- 17(201):1–30 **Multivariate Spearman's rho for Aggregating Ranks Using Copulas**
Justin Bedö, Cheng Soon Ong
- 17(202):1–21 **Nonparametric Network Models for Link Prediction**
Sinead A. Williamson
- 17(203):1–34 **Guarding against Spurious Discoveries in High Dimensions**
Jianqing Fan, Wen-Xin Zhou
- 17(204):1–27 **Bayesian Graphical Models for Multivariate Functional Data**
Hongxiao Zhu, Nate Strawn, David B. Dunson
- 17(205):1–37 **Neural Autoregressive Distribution Estimation**
Benigno Uria, Marc-Alexandre Côté, Karol Gregor, Iain Murray, Hugo Larochelle
- 17(206):1–4 **ERRATA: On the Estimation of the Gradient Lines of a Density and the Consistency of the Mean-Shift Algorithm**
Ery Arias-Castro, David Mason, Bruno Pelletier
- 17(207):1–31 **Modelling Interactions in High-dimensional Data with Backtracking**
Rajen D. Shah
- 17(208):1–50 **Choice of V for V-Fold Cross-Validation in Least-Squares Density Estimation**
Sylvain Arlot, Matthieu Lerasle

- 17(210):1–49 **Towards More Efficient SPSD Matrix Approximation and CUR Matrix Decomposition**
Shusen Wang, Zihua Zhang, Tong Zhang
- 17(211):1–28 **Multi-Objective Markov Decision Processes for Data-Driven Decision Support**
Daniel J. Lizotte, Eric B. Laber
- 17(212):1–63 **Measuring Dependence Powerfully and Equitably**
Yakir A. Reshef, David N. Reshef, Hilary K. Finucane, Pardis C. Sabeti, Michael Mitzenmacher
- 17(213):1–39 **Neyman-Pearson Classification under High-Dimensional Settings**
Anqi Zhao, Yang Feng, Lie Wang, Xin Tong
- 17(214):1–31 **A Statistical Perspective on Randomized Sketching for Ordinary Least-Squares**
Garvesh Raskutti, Michael W. Mahoney
- 17(215):1–26 **Learning Planar Ising Models**
Jason K. Johnson, Diane Oyen, Michael Chertkov, Praneeth Netrapalli
- 17(216):1–52 **Newton-Stein Method: An Optimization Method for GLMs via Stein’s Lemma**
Murat A. Erdogdu
- 17(217):1–40 **Bayesian Decision Process for Cost-Efficient Dynamic Ranking via Crowdsourcing**
Xi Chen, Kevin Jiao, Qihang Lin
- 17(218):1–30 **Multi-scale Classification using Localized Spatial Depth**
Subhajit Dutta, Soham Sarkar, Anil K. Ghosh
- 17(219):1–58 **On Bayes Risk Lower Bounds**
Xi Chen, Adityanand Guntuboyina, Yuchen Zhang
- 17(220):1–58 **Weak Convergence Properties of Constrained Emphatic Temporal-difference Learning with Constant and Slowly Diminishing Step-size**
Huizhen Yu
- 17(221):1–5 **RLScore: Regularized Least-Squares Learners**
Tapio Pahikkala, Antti Airola
- 17(222):1–52 **Stability and Generalization in Structured Prediction**
Ben London, Bert Huang, Lise Getoor
- 17(223):1–52 **Composite Multiclass Losses**
Robert C. Williamson, Elodie Vernet, Mark D. Reid
- 17(224):1–52 **Learning Latent Variable Models by Pairwise Cluster Comparison: Part I - Theory and Overview**
Nuaman Asbeh, Boaz Lerner

- 17(225):1–42 **GenSVM: A Generalized Multiclass Support Vector Machine**
Gerrit J.J. van den Burg, Patrick J.F. Groenen
- 17(226):1–49 **Scalable Approximate Bayesian Inference for Outlier Detection under Informative Sampling**
Terrance D. Savitsky
- 17(227):1–51 **Approximate Newton Methods for Policy Search in Markov Decision Processes**
Thomas Furnston, Guy Lever, David Barber
- 17(228):1–15 **Structure-Leveraged Methods in Breast Cancer Risk Prediction**
Jun Fan, Yirong Wu, Ming Yuan, David Page, Jie Liu, Irene M. Ong, Peggy Peissig, Elizabeth Burnside
- 17(229):1–32 **Gains and Losses are Fundamentally Different in Regret Minimization: The Sparse Case**
Joon Kwon, Vianney Perchet
- 17(230):1–24 **Linear Convergence of Randomized Feasible Descent Methods Under the Weak Strong Convexity Assumption**
Chenxin Ma, Rachael Tappenden, Martin Takáč
- 17(231):1–20 **A Practical Scheme and Fast Algorithm to Tune the Lasso With Optimality Guarantees**
Michael Chichignoud, Johannes Lederer, Martin J. Wainwright
- 17(232):1–17 **A Characterization of Linkage-Based Hierarchical Clustering**
Margareta Ackerman, Shai Ben-David
- 17(233):1–45 **Learning Latent Variable Models by Pairwise Cluster Comparison: Part II - Algorithm and Evaluation**
Nuaman Asbeh, Boaz Lerner
- 17(234):1–35 **Integrative Analysis using Coupled Latent Variable Models for Individualizing Prognoses**
Peter Schulam, Suchi Saria
- 17(235):1–15 **Structure-Leveraged Methods in Breast Cancer Risk Prediction**
Jun Fan, Yirong Wu, Ming Yuan, David Page, Jie Liu, Irene M. Ong, Peggy Peissig, Elizabeth Burnside
- 17(236):1–26 **An Error Bound for L1-norm Support Vector Machine Coefficients in Ultra-high Dimension**
Bo Peng, Lan Wang, Yichao Wu
- 17(237):1–25 **Blending Learning and Inference in Conditional Random Fields**
Tamir Hazan, Alexander G. Schwing, Raquel Urtasun
- 17(238):1–44 **Distributed Submodular Maximization**
Baharan Mirzasoleiman, Amin Karbasi, Rik Sarkar, Andreas Krause

17(239):1–41

On the properties of variational approximations of Gibbs posteriors

Pierre Alquier, James Ridgway, Nicolas Chopin

Variational Dependent Multi-output Gaussian Process Dynamical Systems

Jing Zhao *

JZHAO2011@GMAIL.COM

Shiliang Sun *

SHILIANGSUN@GMAIL.COM

Department of Computer Science and Technology

East China Normal University

500 Dongchuan Road, Shanghai 200241, P. R. China

Editor: Neil Lawrence

Abstract

This paper presents a dependent multi-output Gaussian process (GP) for modeling complex dynamical systems. The outputs are dependent in this model, which is largely different from previous GP dynamical systems. We adopt convolved multi-output GPs to model the outputs, which are provided with a flexible multi-output covariance function. We adapt the variational inference method with inducing points for learning the model. Conjugate gradient based optimization is used to solve parameters involved by maximizing the variational lower bound of the marginal likelihood. The proposed model has superiority on modeling dynamical systems under the more reasonable assumption and the fully Bayesian learning framework. Further, it can be flexibly extended to handle regression problems. We evaluate the model on both synthetic and real-world data including motion capture data, traffic flow data and robot inverse dynamics data. Various evaluation methods are taken on the experiments to demonstrate the effectiveness of our model, and encouraging results are observed.

Keywords: Gaussian process, variational inference, dynamical system, multi-output modeling

1. Introduction

Dynamical systems are widespread in the research area of machine learning. Multi-output time series such as motion capture data, traffic flow data and video sequences are typical examples generated from these systems. Data generated from these dynamical systems usually have the following characteristics. 1) Implicit dynamics exist in the data, and the relationship between the observations and the time indices is nonlinear. For example, the transformation of the frames of a video over time is complex. 2) Possible dependency exists among multiple outputs. For example, for motion capture data, the position of the hand is often closely related to the position of the arm. A simple and straightforward method to model this kind of dynamical systems is to use Gaussian processes (GPs), since GPs provide an elegant method for modeling nonlinear mappings in the Bayesian nonparametric learning framework (Rasmussen and Williams, 2006). Some extensions of GPs have been developed in recent years to better model the dynamical systems. The dynamical systems

modeled by GPs are called the Gaussian process dynamical systems (GPDSs). However, the existing GPDSs have a limitation of ignoring the dependency among the multiple outputs, that is, they may not make full use of the characteristics of data. Our work aims to model the complex dynamical systems more reasonably and flexibly.

Gaussian process dynamical models (GPDMS) as extensions of the GP latent variable model (GP-LVM) (Lawrence, 2004, 2005) were proposed to model human motion (Wang et al., 2006, 2008). The GP-LVM is a nonlinear extension of the probabilistic principal component analysis (Tipping and Bishop, 1999) and is a probabilistic model where the outputs are observed while the inputs are hidden. It introduces latent variables and performs a nonlinear mapping from the latent space to the observation space. The GP-LVM provides an unsupervised non-linear dimensionality reduction method by optimizing the latent variables with the maximum a posteriori (MAP) solution. The GPDM allows to model nonlinear dynamical systems by adding a Markov dynamical prior on the latent space in the GP-LVM. It captures the variability of outputs by constructing the variance of outputs with different parameters. Some research of adapting GPDMs to specific applications was developed, such as object tracking (Urtasun et al., 2006), activity recognition (Gamage et al., 2011) and synthesis, and computer animation (Henter et al., 2012).

Similarly, Damianou et al. (2011, 2014) extended the GP-LVM by imposing a dynamical prior on the latent space to the variational GP dynamical system (VGPDS). The nonlinear mapping from the latent space to the observation space in the VGPDS allows the model to capture the structures and characteristics of data in a relatively low dimensional space. Instead of seeking a MAP solution for the latent variables as in GPDMs, VGPDSs used a variational method for model training. This follows the variational Bayesian method for training the GP-LVM (Titsias and Lawrence, 2010), in which a lower bound of the logarithmic marginal likelihood is computed by variationally integrating out the latent variables that appear nonlinearly in the inverse kernel matrix of the model. The variational Bayesian method was built on the method of variational inference with inducing points (Titsias, 2009). The VGPDS approximately marginalizes out the latent variables and leads to a rigorous lower bound on the logarithmic marginal likelihood. The model and variational parameters of the VGPDS can be learned through maximizing the variational lower bound. This variational method with inducing points was also employed to integrate out the warping functions in the warped GP (Snelson et al., 2003; Lázaro-gredilla, 2012). Park et al. (2012) developed an almost direct application of VGPDSs to phoneme classification, in which a variance constraint in the VGPDS was introduced to eliminate the sparse approximation error in the kernel matrix. Besides variational approaches, expectation propagation based methods (Deisenroth and Mohamed, 2012) are also capable of conducting approximate inference in GPDSs.

However, all the models mentioned above for GPDSs ignore the dependency among multiple outputs, which usually assume that the outputs are conditionally independent. Actually, modeling the dependency among outputs is necessary in many applications such as sensor networks, geostatistics and time-series forecasting, which helps to make better predictions (Boyle, 2007). Indeed, there are some recent works that explicitly considered the dependency of multiple outputs in GPs (Álvarez et al., 2009; Álvarez and Lawrence, 2011; Wilson et al., 2012). Latent force models (LFMs) (Álvarez et al., 2009) are a recent state-of-the-art modeling framework, which can model multi-output dependencies. Later, a

*. The authors contributed equally to this work.

series of extensions of LFMs were presented such as linear, nonlinear, cascaded and switching dynamical LFMs (Álvarez et al., 2011, 2013). In addition, sequential inference methods for LFMs have also been developed (Hartikainen and Särkkä, 2012). Álvarez and Lawrence (2011) employed convolution processes to account for the correlations among outputs to construct a convolved multiple outputs GP (CMOGP) which can be regarded as a specific case of LFMs. To overcome the difficulty of time and storage complexities for large data sets, some efficient approximations for the CMOGP were constructed through the convolution formalism (Álvarez et al., 2010; Álvarez and Lawrence, 2011). This leads to a form of covariance similar in spirit to the so called deterministic training conditional (DTC) approximation (Csató and Oppner, 2001), fully independent training conditional (FITC) approximation (Quinero-Candela and Rasmussen, 2005; Snelson and Ghahramani, 2006) and partially independent training conditional (PITC) approximation (Quinero-Candela and Rasmussen, 2005) for a single output (Lawrence, 2007). The CMOGP is then enhanced and extended to the collaborative multi-output Gaussian process (COGP) for handling large scale cases (Nguyen and Bonilla, 2014). Besides CMOGPs, Wilson et al. (2012) combined neural networks with GPs to construct a GP regression network (GPRN). Outputs in the GPRN are linear combinations of the shared adaptive latent basis functions with input dependent weights. However, these two models are neither introduced nor directly suitable for complex dynamical system modeling. When a dynamical prior is imposed, marginalizing over the latent variables is needed, which can be very challenging.

In this paper, we propose a variational dependent multi-output GP dynamical system (VDM-GPDS). It is a hierarchical Gaussian process model in which the dependency among all the observations is well captured. Specifically, the convolved process covariance function (Álvarez and Lawrence, 2011) is employed to capture the dependency among all the data points across all the outputs. To learn the VDM-GPDS, we first approximate the latent functions in the convolution processes, and then variationally marginalize out the latent variables in the model. This leads to a convenient lower bound of the logarithmic marginal likelihood, which is then maximized by the scaled conjugate gradient method to find out the optimal parameters.

The highlights of this paper are summarized as follows. 1) We explicitly take the dependency among multiple outputs into consideration while other methods (Damiannou et al., 2011; Park et al., 2012) for GPDS modeling assume that the outputs are conditionally independent. In particular, the convolved process covariance functions are used to construct the covariance matrix of the outputs. 2) We use the variational method to compute a lower bound of the logarithmic marginal likelihood of the GPDS model. Compared to Damiannou et al. (2011), our model is more reasonable in specific practical settings, and more challenging as a result of involving complex formulations and computations. 3) Our model can be seen as a multi-layer regression model which regards time indices as inputs and observations as outputs. It can be flexibly extended to handle regression problems. Compared with other dependent multi-output models such as the CMOGP, the VDM-GPDS can achieve much better performance attributed to its latent layers. 4) Our model is applicable to general dependent multi-output dynamical systems and multi-output regression tasks, rather than being specially tailored to a particular application. In this paper, we adapt the model to different applications and obtain promising results.

An earlier short version of this work appeared in Zhao and Sun (2014). In this paper, we add detailed derivations of the variational inference and provide the gradients of the objective function with respect to the parameters. Moreover, we analyze the performance and efficiency of the proposed model. In addition, we supplement experiments on real-world data and all the experimental results are measured under various evaluation criteria.

The rest of the paper is organized as follows. First, we give the model for nonlinear dynamical systems in Section 2, where we use convolution process covariance functions to construct the covariance matrix of the dependent multi-output latent variables. Section 3 gives the derivation of the variational lower bound of the marginal likelihood function and optimization methods. Prediction formulations are introduced in Section 4. Related work is analyzed and compared in Section 5. Experimental results are reported in Section 6 and finally conclusions are presented in Section 7.

2. The Proposed Model

Suppose we have multi-output time series data $\{Y_n, t_n\}_{n=1}^N$, where $Y_n \in \mathbb{R}^D$ is an observation at time $t_n \in \mathbb{R}^+$. We assume that there are low dimensional latent variables that govern the generation of the observations and a GP prior for the latent variables conditional on time captures the dynamical driving force of the observations, as in Damiannou et al. (2011). However, a large difference compared with their work is that we explicitly model the dependency among the outputs through convolution processes (Álvarez and Lawrence, 2011).

Our model is a four-layer GP dynamical system. Here $\mathbf{t} \in \mathbb{R}^N$ represents the input variables in the first layer. The matrix $X \in \mathbb{R}^{N \times Q}$ represents the low dimensional latent variables in the second layer with element $x_{nq} = x_q(t_n)$. Similarly, the matrix $F \in \mathbb{R}^{N \times D}$ denotes the latent variables in the third layer, with element $f_{nd} = f_d(\mathbf{x}_n)$ and the matrix $Y \in \mathbb{R}^{N \times D}$ denotes the observations in the fourth layer whose n th row corresponds to Y_n . The model is composed of an independent multi-output GP mapping from \mathbf{t} to X , a dependent multi-output GP mapping from X to F , and a linear mapping from F to Y .

Specifically, for the first mapping, \mathbf{x} is assumed to be a multi-output GP indexed by time t similarly to Damiannou et al. (2011), that is

$$x_q(t) \sim GP(0, \kappa_x(t, t')), \quad q = 1, \dots, Q, \quad (1)$$

where individual components of the latent function $\mathbf{x}(t)$ are independent sample paths drawn from a GP with a certain covariance function $\kappa_x(t, t')$ parameterized by θ_x . There are several commonly used covariance functions such as the squared exponential covariance function (RBF), the Matérn $3/2$ function and the periodic covariance function (RBFperiodic), which can be adopted to model the time evolution of sequences. For example, an RBF or a Matérn $3/2$ function is usually appropriate for a long time dependent sequence, which will lead to a full covariance matrix. For modeling the evolution of multiple independent sequences, a block-diagonal covariance matrix should be chosen, where each block can be constructed by an RBF or a Matérn $3/2$ function. RBFperiodic is useful to capture the periodicity of the sequences, and multiple kernels can be used to model different time cycles. These kernel

functions take the following forms.

$$\begin{aligned} \kappa_x(RBF)(t_i, t_j) &= \sigma_{rbf}^2 \frac{(t_i - t_j)^2}{2t^2}, \\ \kappa_x(\text{Matérn3/2})(t_i, t_j) &= \sigma_{mat}^2 \left(1 + \frac{\sqrt{3}|t_i - t_j|}{\ell}\right) e^{-\frac{\sqrt{3}|t_i - t_j|}{\ell}}, \\ \kappa_x(\text{RBFperiodic})(t_i, t_j) &= \sigma_{per}^2 \frac{1 - \sin^2(\frac{2\pi}{\ell}(t_i - t_j))}{\ell}. \end{aligned} \quad (2)$$

According to the conditional independency assumption among the latent variables $\{\mathbf{x}_q\}_{q=1}^Q$, we have

$$p(\mathbf{X}|\mathbf{t}) = \prod_{q=1}^Q p(\mathbf{x}_q|\mathbf{t}) = \prod_{q=1}^Q \mathcal{N}(\mathbf{x}_q|\mathbf{0}, \mathbf{K}_{\mathbf{t},\mathbf{t}}), \quad (3)$$

where $\mathbf{K}_{\mathbf{t},\mathbf{t}}$ is the covariance matrix constructed by $\kappa_x(\mathbf{t}, \mathbf{t}')$.

For the second mapping, we assume that \mathbf{f} is another multi-output GP indexed by \mathbf{x} , whose outputs are dependent, that is

$$f_d(\mathbf{x}) \sim \mathcal{GP}(0, \kappa_{f_d, f_{d'}}(\mathbf{x}, \mathbf{x}')), \quad d, d' = 1, \dots, D, \quad (4)$$

where $\kappa_{f_d, f_{d'}}(\mathbf{x}, \mathbf{x}')$ is a convolved process covariance function. The convolved process covariance function captures the dependency among all the data points across all the outputs with parameters $\boldsymbol{\theta}_f = \{\{\Lambda_k\}, \{P_d\}, \{S_{d,k}\}\}$. The detailed formulation of this covariance function will be given in Section 2.1. From the conditional dependency among the latent variables $\{f_{nd}\}_{n=1, d=1}^{N, D}$, we have

$$p(F|X) = p(\mathbf{f}|X) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}_{\mathbf{f},\mathbf{f}}), \quad (5)$$

where \mathbf{f} is a shorthand for $[\mathbf{f}_1^\top, \dots, \mathbf{f}_D^\top]^\top$ and $\mathbf{K}_{\mathbf{f},\mathbf{f}}$ sized $ND \times ND$ is the covariance matrix in which the elements are calculated by the covariance function $\kappa_{f_d, f_{d'}}(\mathbf{x}, \mathbf{x}')$.

The third mapping, which is from the latent variable f_{nd} to the observation y_{nd} , can be written as

$$y_{nd} = f_{nd} + \epsilon_{nd}, \quad \epsilon_{nd} \sim \mathcal{N}(0, \beta^{-1}). \quad (6)$$

Since the observations $\{y_{nd}\}_{n=1, d=1}^{N, D}$ are conditionally independent on F , we get

$$p(Y|F) = \prod_{d=1}^D \prod_{n=1}^N \mathcal{N}(y_{nd}|f_{nd}, \beta^{-1}). \quad (7)$$

Given the above setting, the graphical model for the proposed VDM-GPDS on the training data $\{\mathbf{y}_n, t_n\}_{n=1}^N$ can be depicted as Figure 1. From (3), (5) and (7), the joint probability distribution for the VDM-GPDS model is given by

$$p(\mathbf{Y}, F, X|\mathbf{t}) = p(\mathbf{Y}|F)p(F|X)p(X|\mathbf{t}) = p(\mathbf{f}|X) \prod_{d=1}^D \prod_{n=1}^N p(y_{nd}|f_{nd}) \prod_{q=1}^Q p(\mathbf{x}_q|\mathbf{t}). \quad (8)$$

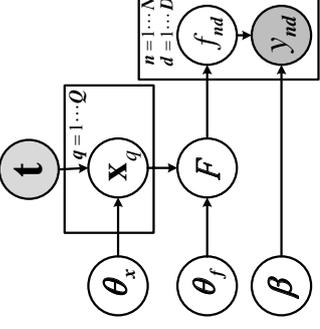


Figure 1: The graphical model for VDM-GPDS.

2.1 Convolved Process Covariance Function

Since the outputs in our model are dependent, we need to capture the correlations among all the data points across all the outputs. Bonilla et al. (2007) and Luttinen and Ilin (2012) used a Kronecker product covariance matrix with the form of $K_{FF} = K_{DD} \otimes K_{NN}$, where K_{DD} is the covariance matrix among the output dimensions, and K_{NN} is the covariance matrix calculated solely from the data inputs. This Kronecker form kernel is constructed from the processes which involve some form of instantaneous mixing of a series of independent processes. This is very limited and actually a special case of some general covariances when covariances calculated from outputs and inputs are independent (Álvarez et al., 2012). For example, if we want to model two output processes in such a way that one process was a blurred version of the other, we cannot achieve this through the instantaneous mixing (Álvarez and Lawrence, 2011). In this paper, we use a more general and flexible kernel in which K_{DD} and K_{NN} are not separated. In particular, the convolution processes (Álvarez and Lawrence, 2011) are employed to model the latent function $F(X)$.

Now we introduce how to construct the convolved process covariance functions. By using independent latent functions $\{u_k(\mathbf{x})\}_{k=1}^K$ and smoothing kernels $\{G_{d,k}(\mathbf{x})\}_{d=1, k=1}^{D, K}$ in the convolution processes, each latent function in (4) in the VDM-GPDS is expressed through a convolution integral,

$$f_d(\mathbf{x}) = \sum_{k=1}^K \int_{\mathbf{X}} G_{d,k}(\mathbf{x} - \tilde{\mathbf{x}}) u_k(\tilde{\mathbf{x}}) d\tilde{\mathbf{x}}. \quad (9)$$

The most common construction is to use Gaussian forms for $\{u_k(\mathbf{x})\}_{k=1}^K$ and $\{G_{d,k}(\mathbf{x})\}_{d=1, k=1}^{D, K}$. So the smoothing kernel is assumed to be

$$\tilde{G}_{d,k}(\mathbf{x}) = S_{d,k} \mathcal{N}(\mathbf{x}|\mathbf{0}, P_d), \quad (10)$$

where $S_{d,k}$ is a scalar value that depends on the output index d and the latent function index k , and P_d is assumed to be diagonal. The latent process $u_k(\mathbf{x})$ is assumed to be

Gaussian with covariance function

$$\kappa_k(\mathbf{x}, \mathbf{x}') = \mathcal{N}(\mathbf{x} - \mathbf{x}' | \mathbf{0}, \Lambda_k). \quad (11)$$

Thus, the covariance between $f_d(\tilde{\mathbf{x}})$ and $f_{d'}(\mathbf{x}')$ is

$$\kappa_{f_d, f_{d'}}(\mathbf{x}, \mathbf{x}') = \sum_{k=1}^K S_{d, k} S_{d', k} \mathcal{N}(\mathbf{x} | \mathbf{x}', P_d + P_{d'} + \Lambda_k). \quad (12)$$

The covariance between $f_d(\mathbf{x})$ and $u_k(\mathbf{x}')$ is

$$\kappa_{f_d, u_k}(\mathbf{x}, \mathbf{x}') = S_{d, k} \mathcal{N}(\mathbf{x} - \mathbf{x}' | \mathbf{0}, P_d + \Lambda_k). \quad (13)$$

These covariance functions (11), (12) and (13) will later be used for approximate inference in Section 3. Compared with Kronecker form kernels, our used convolved kernels have the following advantages. From the perspective of constructing the process f_d , convolved kernels are constructed using the convolution process f_d in which the smoothing kernels $G_{d, k}(\mathbf{x})$ related to \mathbf{x} are employed while Kronecker form kernels are constructed using $f_d(x) = \mathbf{a}_d \mathbf{u}_k(\mathbf{x})$ in which \mathbf{a}_d has no relation to \mathbf{x} (Álvarez and Lawrence, 2011). From the perspective of kernels, for different dimensions d and d' , convolved kernels allow that the covariances $\{\kappa_{f_d, f_{d'}}(\mathbf{x}, \mathbf{x}')\}$ are related to different terms $\mathcal{N}(\mathbf{x} | \mathbf{x}', P_d + P_{d'} + \Lambda_k)$ while Kronecker form kernels indicate that different covariances $\{\kappa_{f_d, f_{d'}}(\mathbf{x}, \mathbf{x}')\}$ share the same term $\kappa_k(\mathbf{x}, \mathbf{x}')$. Thus, our used convolved kernels are more general.

3. Inference and Optimization

As described above, the proposed VDM-GPPDS explicitly models the dependency among multiple outputs, which makes it largely distinct to the previous VGPDS and other GP dynamical systems. In order to make it easy to implement by extending the existing framework of the VGPDS, in the current and the following sections, we will deduce the variational lower bound for the logarithmic marginal likelihood and the posterior distribution for prediction in a formulation similar to the VGPDS. However, many details as described in the paper are specific to our model, and some calculations are more involved.

The fully Bayesian learning for our model requires maximizing the logarithm of the marginal likelihood (Bishop, 2006)

$$p(Y|\mathbf{t}) = \int p(Y|F)p(F|X)p(X|\mathbf{t})dXdF. \quad (14)$$

Note that the integration w.r.t X is intractable, because X appears nonlinearly in the inverse of the matrix $\mathbf{K}_F F$. We attempt to make some approximations for (14).

To begin with, we approximate $p(F|X)$ which is constructed by convolution process $f_d(\mathbf{x})$ in (9). Similarly to Álvarez and Lawrence (2011), a generative approach is used to approximate $f_d(\mathbf{x})$ as follows. We first draw a sample, $\mathbf{u}_k(Z) = [u_k(\mathbf{z}_1), \dots, u_k(\mathbf{z}_M)]^\top$, where $Z = \{\mathbf{z}_m\}_{m=1}^M$ are introduced as a set of input vectors for $u_k(\tilde{\mathbf{x}})$ and will be learned as parameters. We next sample $u_k(\tilde{\mathbf{x}})$ from the conditional prior $p(u_k(\tilde{\mathbf{x}})|\mathbf{u}_k)$. According to the above generating process, $u_k(\tilde{\mathbf{x}})$ in (9) can be approximated by the expectation

$\mathcal{E}(u_k(\tilde{\mathbf{x}})|\mathbf{u}_k)$. Let $U = \{\mathbf{u}_k\}_{k=1}^K$ and $\mathbf{u} = [\mathbf{u}_1^\top, \dots, \mathbf{u}_K^\top]^\top$. The probability distribution of \mathbf{u} can be expressed as

$$p(\mathbf{u}|Z) = \mathcal{N}(\mathbf{0}, \mathbf{K}_{\mathbf{u}, \mathbf{u}}), \quad (15)$$

where $\mathbf{K}_{\mathbf{u}, \mathbf{u}}$ is constructed by $\kappa_k(\mathbf{x}, \mathbf{x}')$ in (11). Combining (9) and (15), we get the probability distribution of \mathbf{f} conditional on \mathbf{u}, X, Z as

$$p(\mathbf{f}|\mathbf{u}, X, Z) = \mathcal{N}(\mathbf{f} | \mathbf{K}_F \mathbf{u}, \mathbf{K}_{\mathbf{u}, \mathbf{u}}^{-1} \mathbf{u}, \mathbf{K}_F F - \mathbf{K}_F \mathbf{u}, \mathbf{K}_{\mathbf{u}, \mathbf{u}}^{-1} \mathbf{K}_F F), \quad (16)$$

where $\mathbf{K}_F \mathbf{u}$ is the cross-covariance matrix between $f_d(\mathbf{x})$ and $u_k(\mathbf{z})$ with element $\kappa_{f_d, u_k}(\mathbf{x}, \mathbf{x}')$ in (13), the block-diagonal matrix $\mathbf{K}_{\mathbf{u}, \mathbf{u}}$ is the covariance matrix between $u_k(\mathbf{z})$ and $u_k(\mathbf{z}')$ with element $\kappa_k(\mathbf{x}, \mathbf{x}')$ in (11), and $\mathbf{K}_F F$ is the covariance matrix between $f_d(\mathbf{x})$ and $f_{d'}(\mathbf{x}')$ with element $\kappa_{f_d, f_{d'}}(\mathbf{x}, \mathbf{x}')$ in (12). Therefore, $p(F|X)$ is approximated by

$$p(F|X) \approx p(\mathbf{f}|\mathbf{u}, X, Z) = \int p(\mathbf{f}|\mathbf{u}, X, Z)p(\mathbf{u}|Z)d\mathbf{u}, \quad (17)$$

and $p(Y|\mathbf{t})$ is converted to

$$p(Y|\mathbf{t}) \approx p(Y|\mathbf{t}, Z) = \int p(Y|\mathbf{f})p(\mathbf{f}|\mathbf{u}, X, Z)p(\mathbf{u}|Z)p(X|\mathbf{t})dFdUdX, \quad (18)$$

where $p(\mathbf{u}|Z) = \mathcal{N}(\mathbf{0}, \mathbf{K}_{\mathbf{u}, \mathbf{u}})$ and $\mathbf{y} = [\mathbf{y}_1^\top, \dots, \mathbf{y}_D^\top]^\top$. It is worth noting that the marginal likelihood in (18) is still intractable as the integration with respect to X remains difficult.

Then, we introduce a lower bound of the logarithmic marginal likelihood $\log p(Y|\mathbf{t})$ using variational methods. We construct a variational distribution $q(F, U, X|Z)$ to approximate the posterior distribution $p(F, U, X|Y, \mathbf{t})$ and compute the Jensen's lower bound on $\log p(Y|\mathbf{t})$ as

$$\mathcal{L} = \int q(F, U, X|Z) \log \frac{p(Y, F, U, X|\mathbf{t}, Z)}{q(F, U, X|Z)} dXdUdF. \quad (19)$$

The variational distribution is assumed to be factorized as

$$q(F, U, X|Z) = p(\mathbf{f}|\mathbf{u}, X, Z)q(\mathbf{u}|Z)q(X). \quad (20)$$

The distribution $p(\mathbf{f}|\mathbf{u}, X, Z)$ in (20) is the same as the second term in (18), which will be eliminated in the term $\log \frac{p(Y, F, U, X|\mathbf{t}, Z)}{q(F, U, X|Z)}$ in (19). The distribution $q(\mathbf{u})$ is an approximation to the posterior distribution $p(\mathbf{u}|\mathbf{u}, X, Y)$, which is arguably Gaussian by maximizing the variational lower bound (Titsias and Lawrence, 2010; Damaniou et al., 2011). The distribution $q(X)$ is an approximation to the posterior distribution $p(X|Y)$, which is assumed to be a product of independent Gaussian distributions $q(X) = \prod_{q=1}^Q \mathcal{N}(\mathbf{x}_q | \boldsymbol{\mu}_q, S_0)$.

After some calculations and simplifications, the lower bound with X, U and F integrated out becomes

$$\begin{aligned} \mathcal{L} = & \log \left[\frac{\beta^{\frac{ND}{2}} |\mathbf{K}_{\mathbf{u}, \mathbf{u}}|^{-\frac{1}{2}}}{(2\pi)^{\frac{ND}{2}} |\beta\psi_2 + \mathbf{K}_{\mathbf{u}, \mathbf{u}}|^{-\frac{1}{2}}} \exp\left\{-\frac{1}{2} \mathbf{y}^\top W \mathbf{y}\right\} \right] \\ & - \frac{\beta\psi_0}{2} + \frac{\beta}{2} \text{Tr}(\mathbf{K}_{\mathbf{u}, \mathbf{u}}^{-1} \psi_2) - \mathbf{KL}[q(X) \| p(X|\mathbf{t})], \end{aligned} \quad (21)$$

where $W = \beta I - \beta^2 \psi_1 (\beta \psi_2 + \mathbf{K}_{u,u})^{-1} \psi_1^\top$, $\psi_0 = \text{Tr}(\mathbf{K}_{f,f} q(X))$, $\psi_1 = (\mathbf{K}_{f,u})_{q(X)}$ and $\psi_2 = (\mathbf{K}_{u,f} \mathbf{K}_{f,u})_{q(X)}$. $\mathbf{KL}[q(X) \| p(X|\mathbf{t})]$ defined by $\int q(X) \log \frac{q(X)}{p(X|\mathbf{t})} dX$ is expressed as

$$\begin{aligned} \mathbf{KL}[q(X) \| p(X|\mathbf{t})] &= \frac{Q}{2} \log |\mathbf{K}_{t,t}| - \frac{1}{2} \sum_{q=1}^Q \log |S_q| \\ &\quad + \frac{1}{2} \sum_{q=1}^Q [\text{Tr}(\mathbf{K}_{t,t}^{-1} S_q) + \text{Tr}(\mathbf{K}_{t,t}^{-1} \boldsymbol{\mu}_q \boldsymbol{\mu}_q^\top)] + \text{const}. \end{aligned} \quad (22)$$

The detailed derivation of this variational lower bound is described in Appendix A where \mathcal{L} is expressed as $\mathcal{L} = \hat{\mathcal{L}} - \mathbf{KL}[q(X) \| p(X|\mathbf{t})]$.

Note that although the lower bound in (21) and the one in VGPPDS (Damianou et al., 2011) look similar, they are essentially distinct and have different meanings. In particular, the variables U in this paper are the samples of the latent functions $\{u_k(\mathbf{x})\}_{k=1}^K$ in the convolution process while in VGPPDS they are samples of the latent variables F . Moreover, the covariance functions of F involved in this paper are multi-output covariance functions while VGPPDS adopts single-output covariance functions. As a result, our model is more flexible and challenging. For example, the calculation of statistics of ψ_0 , ψ_1 and ψ_2 is more complex, as well as the derivatives of the parameters.

3.1 Computation of ψ_0 , ψ_1 , ψ_2

Recall that the lower bound in (21) requires computing the statistics $\{\psi_0, \psi_1, \psi_2\}$. We now detail how to calculate them. ψ_0 is a scalar that can be calculated as

$$\begin{aligned} \psi_0 &= \sum_{n=1}^N \sum_{d=1}^D \sum_{k=1}^K \kappa_{f_d, f_d}(\mathbf{x}_n, \mathbf{x}_n) \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_n, S_n) d\mathbf{x}_n \\ &= \sum_{d=1}^D \sum_{k=1}^K \frac{N S_{d,k} S_{dk}}{(2\pi)^{\frac{D}{2}} |2P_d + \Lambda_k|^{\frac{1}{2}}}. \end{aligned} \quad (23)$$

ψ_1 is a $V \times W$ matrix whose elements are calculated as¹

$$\begin{aligned} (\psi_1)_{v,w} &= \int \kappa_{f_d, u_k}(\mathbf{x}_n, \mathbf{z}_m) \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_n, S_n) d\mathbf{x}_n \\ &= S_{d,k} \mathcal{N}(\mathbf{z}_m | \boldsymbol{\mu}_n, P_d + \Lambda_k + S_n), \end{aligned} \quad (24)$$

where $V = N \times D$, $W = M \times K$, $d = \lfloor \frac{v-1}{N} \rfloor + 1$, $n = v - (d-1)N$, $k = \lfloor \frac{w-1}{M} \rfloor + 1$ and $m = w - (k-1)M$. Here the symbol “ $\lfloor \cdot \rfloor$ ” means rounding down. ψ_2 is a $W \times W$ matrix whose elements are calculated as

$$\begin{aligned} (\psi_2)_{w,w'} &= \sum_{d=1}^D \sum_{n=1}^N \sum_{n'=1}^N \int \kappa_{f_d, u_k}(\mathbf{x}_n, \mathbf{z}_m) \kappa_{f_d, u_{k'}}(\mathbf{x}_n, \mathbf{z}_{m'}) \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_n, S_n) d\mathbf{x}_n \\ &= \sum_{d=1}^D \sum_{n=1}^N \sum_{n'=1}^N S_{d,k} S_{d,k'} \mathcal{N}(\mathbf{z}_m | \mathbf{z}_{m'}, 2P_d + \Lambda_k + \Lambda_{k'}) \mathcal{N}\left(\frac{\mathbf{z}_m + \mathbf{z}_{m'}}{2} | \boldsymbol{\mu}_n, \Sigma_{\psi_2}\right), \end{aligned} \quad (25)$$

1. We borrow the density formulations to express ψ_1 as well as ψ_2 .

where $k = \lfloor \frac{w'-1}{M} \rfloor + 1$, $m = w - (k-1)M$, $k' = \lfloor \frac{w'-1}{M} \rfloor + 1$, $m' = w' - (k'-1)M$ and $\Sigma_{\psi_2} = (P_d + \Lambda_k)^{-1} (2P_d + \Lambda_k + \Lambda_{k'})^{-1} (P_d + \Lambda_{k'}) + S_n$.

3.2 Conjugate Gradient Based Optimization

The parameters involved in (21) include the model parameters $\{\beta, \boldsymbol{\theta}_x, \boldsymbol{\theta}_f\}$ and the variational parameters $\{\{\boldsymbol{\mu}_q, S_q\}_{q=1}^Q, Z\}$. In order to reduce the variational parameters to be optimized and speed up convergence, we reparameterize the variational parameters $\boldsymbol{\mu}_q$ and S_q to $\bar{\boldsymbol{\mu}}_q$ and $\boldsymbol{\lambda}_q$ as done in Oppor and Archambeau (2009) and Damianou et al. (2011)

$$\boldsymbol{\mu}_q = \mathbf{K}_{t,t} \bar{\boldsymbol{\mu}}_q, \quad (26)$$

$$S_q = \left(\mathbf{K}_{t,t}^{-1} + \text{diag}(\boldsymbol{\lambda}_q) \right)^{-1}, \quad (27)$$

where $\text{diag}(\boldsymbol{\lambda}_q) = -2 \frac{\partial \hat{\mathcal{L}}}{\partial S_q}$ is an $N \times N$ diagonal, positive matrix whose N -dimensional diagonal is denoted by $\boldsymbol{\lambda}_q$, and $\bar{\boldsymbol{\mu}}_q = \frac{\partial \hat{\mathcal{L}}}{\partial \boldsymbol{\mu}_q}$ is an N -dimensional vector. Now the variational parameters to be optimized are $\{\{\bar{\boldsymbol{\mu}}_q, \boldsymbol{\lambda}_q\}_{q=1}^Q, Z\}$. Then the derivatives of the lower bound \mathcal{L} with respect to the variational parameters $\bar{\boldsymbol{\mu}}_q$ and $\boldsymbol{\lambda}_q$ become

$$\frac{\partial \mathcal{L}}{\partial \bar{\boldsymbol{\mu}}_q} = \mathbf{K}_{t,t} \left(\frac{\partial \hat{\mathcal{L}}}{\partial \boldsymbol{\mu}_q} - \bar{\boldsymbol{\mu}}_q \right), \quad (28)$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}_q} = -(S_q \circ S_q) \left(\frac{\partial \hat{\mathcal{L}}}{\partial S_q} + \frac{1}{2} \boldsymbol{\lambda}_q \right). \quad (29)$$

All the parameters are jointly optimized by the scaled conjugate gradient method to maximize the lower bound in (21). The detailed gradients with respect to the parameters are given in Appendix B.

4. Prediction

The proposed model can perform prediction for complex dynamical systems in two situations. One is prediction with only time and the other is prediction with time and partial observations. In addition, it can be adapted to regression models.

4.1 Prediction with Only Time

If the model is learned with training data Y , one can predict new outputs with only given time. In the Bayesian framework, we need to compute the posterior distribution of the predicted outputs $Y_* \in \mathbb{R}^{N_* \times D}$ on some given time instants $\mathbf{t}_* \in \mathbb{R}^{N_*}$. The expectation is used as the estimate and the autocovariance is used to show the prediction uncertainty. With the parameters as well as time \mathbf{t} and \mathbf{t}_* omitted, the posterior density is given by

$$p(Y_* | Y) = \int p(Y_* | F_*) p(F_* | X_*, Y) p(X_* | Y) dF_* dX_*, \quad (30)$$

where $F_* \in \mathbb{R}^{N_* \times D}$ denotes the set of latent variables (the noise-free version of Y_*) and $X_* \in \mathbb{R}^{N_* \times Q}$ represents the latent variables in the low dimensional space.

The distribution $p(F_*|X_*, Y)$ is approximated by the variational distribution

$$p(F_*|X_*, Y) \approx q(\mathbf{f}_*|X_*) = \int p(\mathbf{f}_*|\mathbf{u}, X_*)q(\mathbf{u})d\mathbf{u}, \quad (31)$$

where $\mathbf{f}_*^\top = [\mathbf{f}_{*1}^\top, \dots, \mathbf{f}_{*D}^\top]$, and $p(\mathbf{f}_*|\mathbf{u}, X_*)$ is Gaussian. Since the optimal setting for $q(\mathbf{u})$ in our variational framework is also found to be Gaussian, $q(\mathbf{f}_*|X_*)$ is Gaussian that can be computed analytically. The distribution $p(X_*|Y)$ is approximated by the variational distribution $q(X_*)$ which is Gaussian. Given $p(F_*|X_*, Y)$ approximated by $q(\mathbf{f}_*|X_*)$ and $p(X_*|Y)$ approximated by $q(X_*)$, the posterior density of \mathbf{f}_* (the noise-free version of \mathbf{y}_*) is now approximated by

$$p(\mathbf{f}_*|Y) = \int q(\mathbf{f}_*|X_*)q(X_*)dX_*. \quad (32)$$

The specific formulations of the distributions $p(\mathbf{f}_*|\mathbf{u}, X_*)$, $q(\mathbf{f}_*|X_*)$ and $q(X_*)$ are given in Appendix C as a more comprehensive treatment.

However, the integration of $q(\mathbf{f}_*|X_*)$ w.r.t $q(X_*)$ is not analytically feasible. Following Daniannou et al. (2011), we give the expectation of \mathbf{f}_* as $\mathcal{E}(\mathbf{f}_*)$ and its element-wise autocovariance as vector $\mathcal{C}(\mathbf{f}_*)$ whose $(\tilde{n} \times d)$ th entry is $\mathcal{C}(f_{\tilde{n}d})$ with $\tilde{n} = 1, \dots, N_*$ and $d = 1, \dots, D$.

$$\mathcal{E}(\mathbf{f}_*) = \psi_{1*}\mathbf{b}, \quad (33)$$

$$\mathcal{C}(f_{\tilde{n}d}) = \mathbf{b}^\top (\psi_{2\tilde{n}}^{d2} - (\psi_{1\tilde{n}}^{d1})^\top \psi_{1\tilde{n}}^{d1}) \mathbf{b} + \psi_{0*}^d - \text{Tr} \left[(\mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} - (\mathbf{K}_{\mathbf{u}\mathbf{u}} + \beta\psi_2) \mathbf{1}^{-1}) \psi_{2*}^d \right], \quad (34)$$

where $\psi_{1*} = \langle \mathbf{K}_{\mathbf{f}_*\mathbf{u}} \rangle_{q(X_*)}$, $\mathbf{b} = \beta \mathbf{K}_{\mathbf{u}\mathbf{u}} + \beta\psi_2 \mathbf{1}^{-1} \psi_{1*}^\top \mathbf{y}$, $\psi_{1\tilde{n}}^{d1} = \langle \mathbf{K}_{f_{\tilde{n}d}\mathbf{u}} \rangle_{q(X_*)}$, $\psi_{2\tilde{n}}^{d2} = \langle \mathbf{K}_{\mathbf{u}f_{\tilde{n}d}} \rangle_{q(X_*)}$, $\psi_{0*}^d = \text{Tr}(\langle \mathbf{K}_{\mathbf{f}_*\mathbf{f}_*} \rangle_{q(X_*)})$ and $\psi_{2*}^d = \langle \mathbf{K}_{\mathbf{u}\mathbf{f}_*} \rangle_{q(X_*)}$. Since \mathbf{y}_* is the noisy version of F_* , the expectation and element-wise autocovariance of \mathbf{y}_* are $\mathcal{E}(\mathbf{y}_*) = \mathcal{E}(\mathbf{f}_*)$ and $\mathcal{C}(\mathbf{y}_*) = \mathcal{C}(\mathbf{f}_*) + \beta^{-1} \mathbf{1}_{N_*D}$, where $\mathbf{y}_*^\top = [\mathbf{y}_{*1}^\top, \dots, \mathbf{y}_{*D}^\top]$.

4.2 Prediction with Time and Partial Observations

Prediction with time and partial observations can be divided into two cases. In one case, we need to predict $Y_*^m \in \mathbb{R}^{N_* \times D_m}$ which represents the outputs on missing dimensions, given $Y_*^m \in \mathbb{R}^{N_* \times D_p}$ which represents the outputs observed on partial dimensions. We call this task reconstruction. In the other case, we need to predict $Y_*^m \in \mathbb{R}^{N_* \times D}$ which means the outputs at the next time, given $Y_*^m \in \mathbb{R}^{N_* \times D}$ which means the outputs observed on all dimensions at the previous time. We call this task forecasting.

For the task of reconstruction, we should compute the posterior density of Y_*^m which is given below (Daniannou et al., 2011)

$$p(Y_*^m|Y_*^p, Y) = \int p(Y_*^m|F_*^m)p(F_*^m|X_*, Y_*^p, Y)p(X_*|Y_*^p, Y)dF_*^m dX_*. \quad (35)$$

$p(X_*|Y_*^p, Y)$ is approximated by a Gaussian distribution $q(X_*)$ whose parameters need to be optimized for the sake of considering the partial observations Y_*^p . This requires maximizing a new lower bound of $\log p(Y, Y_*^p)$ which can be expressed as

$$\begin{aligned} \tilde{\mathcal{L}} = \log & \left[\frac{\beta^{\frac{ND+N_*D_p}{2}} |\mathbf{K}_{\mathbf{u}\mathbf{u}}|^{\frac{1}{2}}}{(2\pi)^{\frac{ND+N_*D_p}{2}} |\beta\psi_2 + \mathbf{K}_{\mathbf{u}\mathbf{u}}|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2} \tilde{\mathbf{y}}^\top \tilde{\mathbf{M}} \tilde{\mathbf{y}}\right\} \right] \\ & - \frac{\beta\psi_0}{2} + \frac{\beta}{2} \text{Tr}(\mathbf{K}_{\mathbf{u}\mathbf{u}}^{-1} \psi_2) - \mathbf{KL}[q(X, X_*)||p(X, X_*)|\mathbf{t}, \mathbf{t}_*], \end{aligned} \quad (36)$$

where $\tilde{\mathbf{W}} = \beta \mathbf{I} - \beta^2 \psi_1 (\beta\psi_2 + \mathbf{K}_{\mathbf{u}\mathbf{u}})^{-1} \psi_1^\top$, $\psi_0 = \text{Tr}(\langle \mathbf{K}_{\mathbf{f}_*\mathbf{f}_*} \rangle_{q(X, X_*)})$, $\psi_1 = \langle \mathbf{K}_{\mathbf{f}_*\mathbf{u}} \rangle_{q(X, X_*)}$ and $\psi_2 = \langle \mathbf{K}_{\mathbf{u}\mathbf{f}_*} \rangle_{q(X, X_*)}$. The vector $\tilde{\mathbf{y}}$ splices the vectorization of matrix Y and the vectorization of matrix Y_*^p , i.e. $\tilde{\mathbf{y}} = [\text{vec}(Y); \text{vec}(Y_*^p)]$. The vector $\tilde{\mathbf{f}}$ corresponds to the noise-free version of $\tilde{\mathbf{y}}$. Moreover, parameters of the new variational distribution $q(X, X_*)$ are jointly optimized because of the coupling of X and X_* . Then the marginal distribution $q(X_*)$ is obtained from $q(X, X_*)$. Note that when multiple sequences such as X_* and X are independent, only the separated variational distribution $q(X_*)$ is optimized.

For the task of forecasting, we focus on real-time forecasting for which the outputs are dependent on the previous ones and the training set Y is not used in the prediction stage. The variational distribution $q(X_*)$ can be directly computed as (70). Then the posterior density of Y_*^m is computed as (66), but with Y_* and Y replaced with Y_*^n and Y_*^p , respectively. $\mathcal{E}(Y_*^m)$ is the estimate of the output Y_*^m . An application for forecasting is given in Section 6.3.

4.3 Adaptation to Regression Models

Since the VDM-GPDS can be seen as a multi-layer regression model which regards time indices as inputs and observations as outputs. It can be flexibly extended to solve regression problems. Specifically, the time indices in the dynamical systems are replaced with the observed input data V . In addition, the kernel functions for the latent variables X are replaced by some appropriate functions such as automatic relevance determination (ARD) kernels:

$$K_{\tilde{n}}(\mathbf{v}, \mathbf{v}') = \sigma_{\tilde{n}}^2 \rho^{\frac{1}{2}} \sum_{p=1}^P \rho^{\frac{1}{2}} \exp(-\rho^p |v_p - v'_p|). \quad (37)$$

Model inference and optimization remain the same except for some changes for model parameters $\theta_{\tilde{n}}$. Compared with other dependent multi-output regression models such as the CMOGP, the VDM-GPDS can achieve much better performance. This could be attributed to its use of latent layers.

5. Related Work

Daniannou et al. (2011) described a GP dynamical system with variational Bayesian inference called VGPDS in which the latent variables X are imposed a GP prior to model the dynamical driving force and capture the high dimensional data's characteristics. After introducing inducing points, the latent variables are variationally integrated out. The outputs of VGPDS are generated from multiple independent GPs with the same latent variables X and the same parameters, resulting in the advantage that VGPDS can handle high dimensional situations. However, the explicit dependency among the multiple outputs is ignored in this model while this kind of dependency is very important for many applications. In contrast, the CMOGP (Alvarez and Lawrence, 2011) and GPRN (Wilson et al., 2012) model the dependency of different outputs through convolved process covariance functions and an adaptive network, respectively. Nevertheless, these two methods are not directly suitable for dynamical system modeling. If applied to dynamical systems with time as inputs, they cannot well capture the complexity of dynamical systems because there is only one nonlinear mapping between the input and output included.

Our model is capable of capturing the dependency among outputs as well as modeling the dynamical characteristics. It is also very different from the GPDM (Wang et al., 2006, 2008) which models the variance of each output with different scale parameters and employs Markov dynamical prior on the latent variables. The Gaussian prior for the latent variables in the VDM-GPDS can model the dynamical characteristics in the systems better than the Markov dynamical prior, since it can model different kinds of dynamics by using different kernels such as using periodic kernels to model periodicity. Moreover, in contrast to the GPDM that estimates the latent variables X through the MAP, the VDM-GPDS integrates out the latent variables with variational methods. This is in the same spirit of the technique used in Damianou et al. (2011), which can provide a principled approach to handle uncertainty in the latent space and determine its principal dimensions automatically. In addition, the multiple outputs in our model are modeled by convolution processes as in Álvarez and Lawrence (2011), which can flexibly capture the correlations among the outputs.

6. Experiments

In this part, we design five experiments to evaluate our model for four different kinds of applications including prediction with only time as inputs, reconstruction of the missing data, real-time forecasting and solving robot inverse dynamics problem. Two experiments are performed on synthetic data and three on real-world data. A number of models such as the CMOGP/COGP, GPDM, VGPDS and VDM-GPDS are tested on the data. The root mean square error (RMSE) and mean standardized log loss (MSLL) (Rasmussen and Williams, 2006) are taken as the performance measures. In particular, let \hat{Y}^* be the estimate of matrix Y^* , and then the RMSE can be formulated as

$$\text{RMSE}(Y^*, \hat{Y}^*) = \left[\frac{1}{D} \frac{1}{N} \sum_d \sum_n (y_n^{*d} - \hat{y}_n^{*d})^2 \right]^{\frac{1}{2}}. \quad (38)$$

MSLL is the mean negative log probability of all the test data under the learned model Γ and training data Y , which can be formulated as

$$\text{MSLL}(Y^*, \Gamma) = \frac{1}{N} \sum_n \{-\log p(\mathbf{Y}_n^* | \Gamma, Y)\}. \quad (39)$$

The lower value of the RMSE and MSLL we get, the better the performance of the model is. Our code is implemented based on the framework of publicly available code for the VGPDS and CMOGP.

6.1 Synthetic Data

In this section, we evaluate our method on synthetic data generated from a complex dynamical system. The latent variables X are independently generated by the Ornstein-Uhlenbeck (OU) process (Archambeau et al., 2007)

$$dx_q = -\gamma x_q dt + \sqrt{\sigma^2} dW, \quad q = 1, \dots, Q, \quad (40)$$

The outputs Y are generated through a multi-output GP

$$y_d(\mathbf{x}) \sim \mathcal{GP}(0, \kappa_{f_d, f_d'}(\mathbf{x}, \mathbf{x}')), \quad d, d' = 1, \dots, D, \quad (41)$$

	Spline	CMOGP	GPDM	VGPDS	VDM-GPDS
RMSE(y_1)	1.91±0.43	1.75±0.38	1.70±0.18	1.51±0.31	1.43 ± 0.23
RMSE(y_2)	4.23±1.01	3.46±0.67	3.32±0.27	2.99±0.53	2.82 ± 0.35
RMSE(y_3)	6.88±1.91	5.19±0.99	4.83±0.28	4.24±0.85	4.09 ± 0.59
RMSE(y_4)	6.99±1.52	7.50±0.94	5.98±0.55	5.16±0.92	5.00 ± 0.60

Table 1: Averaged RMSE (%) with the standard deviation (%) for predictions on the output-dependent synthetic data.

	CMOGP	GPDM	VGPDS	VDM-GPDS
MSLL(y_1)	-2.63±0.22	$2.21 \times 10^4 \pm 4.86 \times 10^4$	-2.73±0.08	-2.79 ± 0.08
MSLL(y_2)	-1.99±0.13	$1.93 \times 10^4 \pm 5.23 \times 10^4$	-2.14±0.15	-2.18 ± 0.15
MSLL(y_3)	-1.49±0.21	$3.92 \times 10^4 \pm 8.17 \times 10^4$	-1.66±0.24	-1.66 ± 0.21
MSLL(y_4)	-1.08±0.21	$9.90 \times 10^4 \pm 1.98 \times 10^5$	-1.31±0.41	-1.32 ± 0.25

Table 2: Averaged MSLL with the standard deviation for predictions on the output-dependent synthetic data.

where $\kappa_{f_d, f_d'}(\mathbf{x}, \mathbf{x}')$ defined in (12) is the multi-output covariance function. In this paper, the number of the latent functions in (9) is set to one, i.e., $K = 1$, which is also the common setting used in Álvarez and Lawrence (2011).

We sample the synthetic data by two steps. First we use the differential equation with parameters $\gamma = 0.5$, $\sigma = 0.01$ to sample $N = 200$, $Q = 2$ latent variables at time interval $[-1, 1]$. Then we sample $D = 4$ dimensional outputs, each of which has 200 observations through the multi-output GP with the following parameters $S_{1,1} = 1$, $S_{2,1} = 2$, $S_{3,1} = 3$, $S_{4,1} = 4$, $P_1 = [5, 1]^\top$, $P_2 = [5, 1]^\top$, $P_3 = [3, 1]^\top$, $P_4 = [2, 1]^\top$ and $\Lambda = [4, 5]^\top$. For approximation, 30 random inducing points are used. In addition, white Gaussian noise is added to each output.

6.1.1 PREDICTION

Here we evaluate the performance of our method for predicting the outputs given only time over the synthetic data. We randomly select 50 points from each output for training with the remaining 150 points for testing. This is repeated for ten times. The CMOGP, GPDM and VGPDS are performed as comparisons. The cubic spline interpolation (spline for short) is also chosen as a baseline. The latent variables X in the GPDM, VGPDS and VDM-GPDS with two dimensions are initialized by using the principal component analysis on the observations. Moreover, the Matérn 3/2 covariance function is used in the VGPDS and VDM-GPDS.

Table 1 and Table 2 present the RMSE and MSLL for predictions, respectively. The best results are shown in bold. From the tables, we can find that for prediction on the data of each dimension, our model obtains the lowest RMSE and MSLL. We analyze the reasons as follows. First, since the data in this experiment are generated from a complex

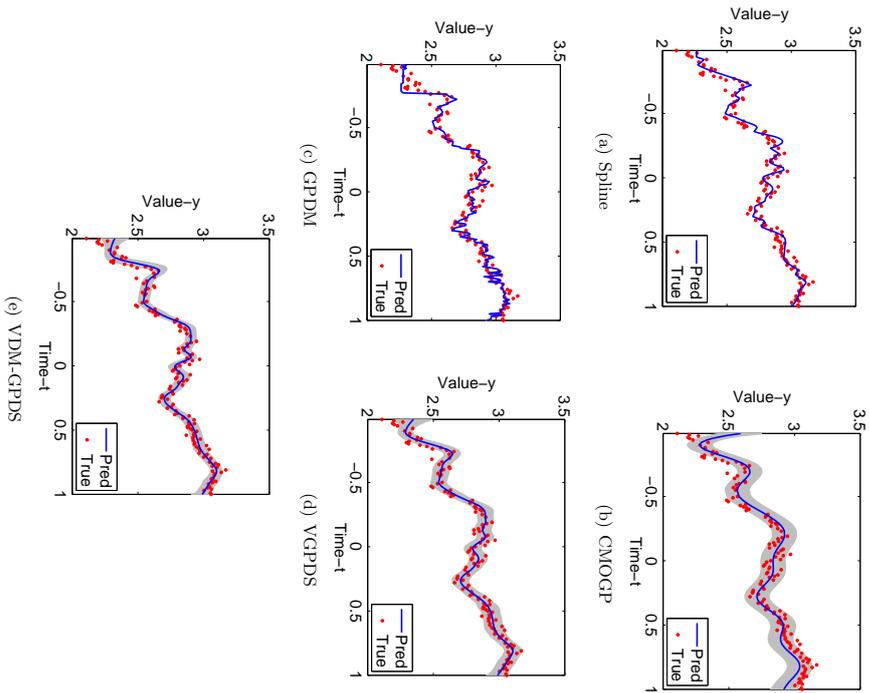


Figure 2: Predictions for $y_4(t)$ with the five methods. Pred and True indicate predicted and observed values, respectively. The shaded regions represent two standard deviations for the predictions.

dynamical system that combines two GP mappings, the CMOGP which consists of only one GP mapping cannot capture the complexity well. Moreover, the VDM-GPDS models the explicit dependency among the multiple outputs while the VGPDS and GPDM do not. The assumption of multi-output dependency is appropriate for the generative model. Further,

Table 3: Averaged RMSE (%) with the standard deviation (%) for predictions on the output-independent synthetic data.

	Spline	CMOGP	GPDM	VGPDS	VDM-GPDS
RMSE(y_1)	3.81±0.82	11.49±0.26	3.82±1.55	2.18 ± 0.06	2.21±0.06
RMSE(y_2)	2.58±0.68	3.59±0.72	3.45±1.70	2.06±0.19	2.05 ± 0.13
RMSE(y_3)	3.26±0.90	1.75±0.10	3.57±1.71	1.68 ± 0.09	1.72±0.12
RMSE(y_4)	9.06±1.17	8.34±10.89	7.10±1.28	4.48±0.23	4.45 ± 0.20

	CMOGP	GPDM	VGPDS	VDM-GPDS
MSLL(y_1)	-0.74±0.02	5.22×10 ² ±5.23×10 ²	-2.34 ± 0.04	-2.33±0.04
MSLL(y_2)	-1.91±0.24	1.10×10 ³ ±2.22×10 ³	-2.36±0.10	-2.36 ± 0.13
MSLL(y_3)	-2.62±0.05	2.10×10 ² ±3.43×10 ²	-2.50±0.08	-2.52 ± 0.11
MSLL(y_4)	-1.38±0.66	5.19×10 ² ±1.16×10 ³	-1.46±0.17	-1.48 ± 0.18

Table 4: Averaged MSLL with the standard deviation for predictions on the output-independent synthetic data.

the GPDM cannot work well in the case in which data on many time intervals are lost. Prediction with the GPDM results in very high MSLL. To sum up, our model gives the best performance among the five models as expected.

In order to give intuitive representations, we draw one prediction result from the ten experiments in Figure 2 where the shaded regions in 2(b), 2(c), 2(d) and 2(e) represent two standard deviations for the predictions. Through the figures, it is clear that the VDM-GPDS has higher accuracies and smaller variances. Note that the GPDM has very small variances, but low accuracies, which leads to the high MSLL as in Table 2. With all the evaluation measures considered, the VDM-GPDS gives the best performance of prediction with only time as inputs.

In addition, to verify the flexibility of the VDM-GPDS, we perform experiments on the output-independent data which are generated analogously to Section 6.1. In particular, the output-independent data are generated using Equation (41) but with $r_{t,t'}^d(\mathbf{x}, \mathbf{x}') = 0$ for $d \neq d'$ after generating X . Note that the GPDM and VGPDS do not make the assumption of output dependency. The results in terms of RMSE and MSLL are shown in Table 3 and Table 4 where we can see that our model performs as well as the VGPDS and significantly better than the CMOGP and GPDM.

6.1.2 RECONSTRUCTION

In this section, we compare the VDM-GPDS with the k -nearest neighbor best (k -NNbest) method which chooses the best k from $\{1, \dots, 5\}$, the CMOGP, GPDM and VGPDS for recovering missing points given time and partially observed outputs. We set $S_{4,1} = -4$ to generate data in this part, which makes the output y_4 be negatively correlated with the others. We remove all outputs y_1 or y_4 at time interval $[0.5, 1]$ from the 50 training points,

	k -NNbest	CMOGP	GPDM	VGPDs	VDM-GPDS
RMSE(y_1)	1.87±0.62	1.90±0.31	2.69±3.67	1.49±0.94	0.98 ± 0.34
RMSE(y_4)	13.51±2.54	9.31±0.87	12.61±2.43	6.79±6.07	5.56 ± 1.88

Table 5: Averaged RMSE (%) with the standard deviation (%) for reconstructions of the missing points for y_1 and y_4 .

	CMOGP	GPDM	VGPDs	VDM-GPDS
MSLL(y_1)	-1.74±0.20	$1.40 \times 10^3 \pm 5.96 \times 10^4$	-2.29±0.46	-2.86 ± 0.09
MSLL(y_4)	0.31±0.62	$7.40 \times 10^4 \pm 8.34 \times 10^4$	-1.64±0.69	-2.35 ± 0.10

Table 6: Averaged MSLL with the standard deviation for reconstructions of the missing points for y_1 and y_4 .

resulting in 35 points as training data. Note that the CMOGP considers all the present outputs as the training set while the GPDM, VGPDs and VDM-GPDS only consider the outputs at time interval $[-1, 0.5]$ as the training set.

Table 5 and Table 6 show the averaged RMSE and MSLL with the standard deviation for reconstructions of the missing points for y_1 and y_4 . The proposed model performs best with the lowest RMSE and MSLL. Specifically, our model can make full use of the present data on some dimensions to reconstruct the missing data through the dependency among outputs. This advantage is shown by comparing with the GPDM and VGPDs. In addition, the two Gaussian process mappings in the VDM-GPDS help to well model the dynamical characteristics and complexity of the data. This advantage is shown by comparing to the CMOGP.

Figure 3 shows one reconstruction result for y_4 from the ten experiments by five different methods. It can be seen that the results of the VDM-GPDS are the closest to the true values among the compared methods. This indicates the superior performance of our model for the reconstruction task.

6.2 Human Motion Capture Data

In order to demonstrate the validity of the proposed model on real-world data, we employ ten sequences of runs/jogs from subject 35 (see Figure 4 for a skeleton) and two sequences of runs/jogs from subject 16 in the CMU motion capture database for the reconstruction task. In particular, our task is to reconstruct the right leg or the upper body of one test sequence on the motion capture data given training sequences. We preprocess the data as in Lawrence (2007) and divide the sequences into training and test data. Nine independent training sequences are all from subject 35 and the remaining three testing sequences are from subject 35 and subject 16 (one from subject 35 and two from subject 16). The average length of each sequence is 40 frames and the output dimension is 59.

We conduct this reconstruction with six different methods, the nearest neighbor in the angle space (NN) and the scaled space (NN sc.) (Taylor et al., 2006), the CMOGP, GPDM,

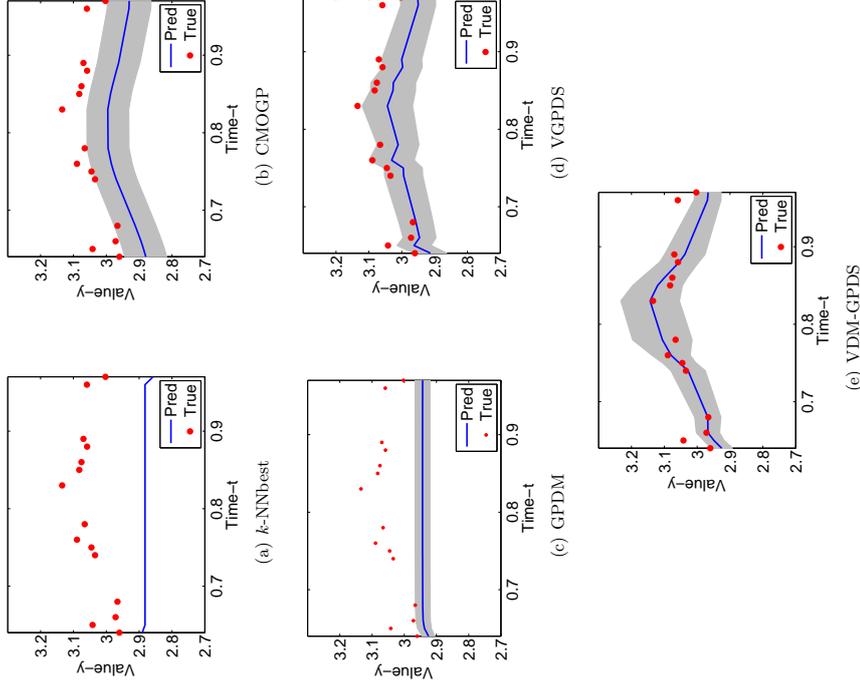


Figure 3: Reconstructions of the missing points for $y_4(t)$ with the five methods. Pred and True indicate predicted and observed values, respectively. The shaded regions represent two standard deviations for the predictions.

VGPDs and VDM-GPDS. For the CMOGP, periodic time indices with different cycles are used as inputs where the length of each sequence is a cycle. For the GPDM, parameters and latent variables are set as in Wang et al. (2006). For the VGPDs and VDM-GPDS, the RBF kernel is adopted in this set of experiments to construct $\mathbf{K}_{t,t}$ which is a block-

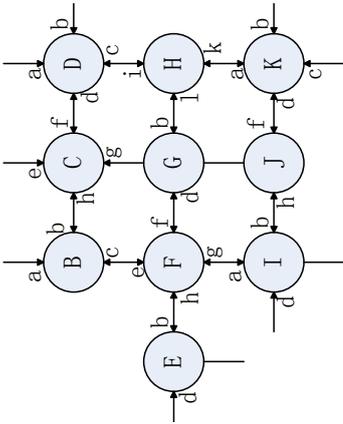


Figure 5: A patch taken from the urban traffic map of highways.

	RW	VGPDS	VDM-GPDS
RMSE(Bb)	84.95	81.90	80.88
RMSE(Ch)	72.49	65.33	62.08
RMSE(Dd)	66.45	61.68	57.97
RMSE(Eb)	153.46	148.42	140.69
RMSE(Fe)	151.16	143.35	131.74
RMSE(Gd)	174.18	162.81	147.14
RMSE(Hi)	95.57	92.89	85.19
RMSE(Ib)	142.85	129.21	121.15
RMSE(Jh)	146.52	141.50	128.66
RMSE(Ka)	94.15	88.23	75.23

Table 9: The RMSE for forecasting results on the traffic flow data.

the historic traffic flows of the four road links to predict the flows of Gd in the next interval. The historic time for forecasting is fixed as four intervals. We compare our model with the Random Walk (RW) and VGPDS. The RW is to forecast the current value using the last value (Williams, 1999), which is chosen as a baseline. According to the descriptions about real-time forecasting in Section 4.2, the VGPDS can be adapted to apply to this experiment. Moreover, in the previous experiments, the VGPDS performs best among the compared models except the VDM-GPDS. Therefore, it is sufficient to compare our model with the RW and VGPDS. Note that the realization of the VGPDS also takes the periodicity into consideration.

Table 9 and Table 10 show the RMSE and MSLL for forecasting results with three methods over the testing sets, respectively. It is obvious that the VDM-GPDS achieves the best performance, even for the road links with large volumes (and large fluctuations) such as Gd . This is attributed to the fact that our model well captures both the temporal and spatial dependency of the traffic flow data. In particular, the relationship between the traffic flows of the objective road link and its upstream links is captured by the multi-

	RW	VGPDS	VDM-GPDS
MSLL(Bb)	.	5.87	5.85
MSLL(Ch)	.	6.04	5.90
MSLL(Dd)	.	5.64	5.57
MSLL(Eb)	.	6.40	6.33
MSLL(Fe)	.	6.41	6.37
MSLL(Gd)	.	6.53	6.44
MSLL(Hi)	.	5.93	5.90
MSLL(Ib)	.	6.33	6.25
MSLL(Jh)	.	6.38	6.30
MSLL(Ka)	.	6.07	5.86

Table 10: The MSLL for forecasting results on the traffic flow data.

output dependency in the VDM-GPDS; the relationship between the traffic flows of the objective road link and its own historical series is captured by the dynamical characteristics modeled in the VDM-GPDS. Therefore, the entire cause information is well collected by the VDM-GPDS to predict the traffic flows of the objective road link.

To be intuitive, we give the final forecasting results of the performed models for the road link Gd in the last three days in Figure 6. The VDM-GPDS has shown great superiority to the compared models. As seen from Figure 6(a) and Figure 6(b), the forecasting results with the RW and VGPDS often lag.

6.4 Robot Inverse Dynamics Problem

The robot inverse dynamics problem is an important task in the robot areas (Sciavicco and Vijayakumar, 2000). For a goal of touching or grasping a subject using a robotic manipulator, it usually needs the following procedures. First, the inverse kinematic calculates the robot joint coordinates given the pose of the end-effector. Then trajectory planning decides a trajectory describing how a robot should move to achieve the desired task. Finally, given the trajectory, i.e., the motion specified by the joint angles, velocities and accelerations, the torques needed at the joints to drive it along the trajectory are computed by the inverse dynamics. What we concerned here is the robot inverse dynamics problem. Analytical models for the inverse dynamics are often infeasible, for example due to uncertainty in the physical parameters of the robot, or the difficulty of modeling frictions. This leads to the need to learn the inverse dynamics by some machine learning methods (Chai et al., 2009; Nguyen and Bonilla, 2014).

We approximate the inverse dynamics model of a 7-degree-of-freedom anthropomorphic robot arm (see Figure 7). The inverse dynamics model of the robot is strongly nonlinear due to a vast amount of superpositions of sine and cosine functions in robot dynamics. The data consist of 21 input dimensions: 7 joint positions, velocities, and accelerations. The goal of learning is to approximate the appropriate torque command of one robot motor in response to the input vector. We choose 4449 data points from the original data set which consists of 48933 data points. We use 100, 300 and 500 points for training, respectively and the rest for testing. All the experiments are repeated for ten times. We consider joint

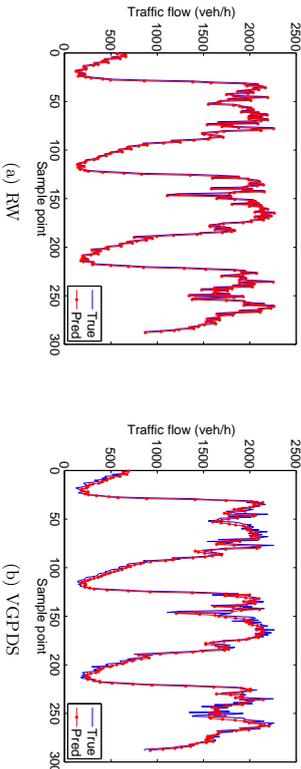


Figure 6: Forecasting results of the performed models for the road link GdI on the last three days.

learning for the two couples, the 2nd and 3rd, the 4th and 7th torques. Note that, the 2nd and 3rd torques are negatively correlated while the 4th and 7th torques are positively correlated.

We adapt our dynamical model to a regression model as described in Section 4.3. In order to demonstrate the performance of our model on the regression problem. We compare our model with the single Gaussian process regression (sGPR), multi-task Gaussian process (MTGP), collaborative multi-output Gaussian process (COGP) and VGPPDS. The sGPR is to learn the torque for each joint separately. The MTGP regards the torques from different joints as different tasks. The COGP is a scalable method which is extended from the CMOGP by introducing stochastic variational inference. As the COGP is an enhanced version of the CMOGP for robot inverse dynamics problems (Nguyen and Bonilla, 2014), we do not include the CMOGP in this experiment. For fairness, we set the batch size in the COGP the same as the number of training points. Note that the exact inference is used for the sGPR and MTGP. For the COGP, VGPDS and VDM-GPDS, variational inference is employed and the same size of inducing points are included. Particularly, 15, 20 and

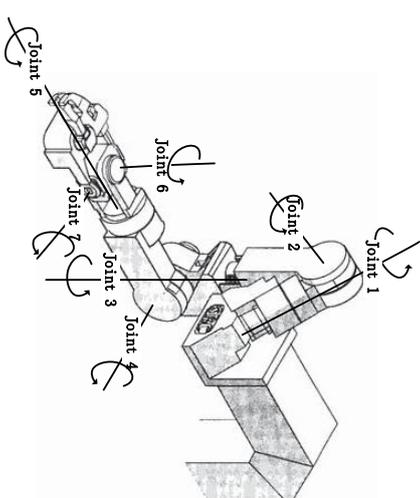


Figure 7: Sketch of the SARCOS dextrous arm (Vijayakumar and Schaal, 2000).

Method ($N = 100$)	2nd joint		3rd joint	
	RMSE	MSLL	RMSE	MSLL
sGPR	6.33±1.33	3.85±0.86	4.48±1.04	4.06±1.18
MTGP	5.52±0.79	3.88±0.19	3.20±0.38	3.10±0.27
COGP	5.11±0.44	4.17±0.69	3.18±0.23	3.39±0.38
VGPDS	4.89±0.36	6.20±0.87	2.96±0.26	5.73±0.89
VDM-GPDS	4.55 ± 0.34	2.95 ± 0.14	2.68 ± 0.22	2.34 ± 0.15

Method ($N = 100$)	4th joint		7th joint	
	RMSE	MSLL	RMSE	MSLL
sGPR	5.01±2.06	4.33±1.54	1.04±0.22	2.76±1.53
MTGP	3.27±0.35	3.44±0.49	0.72±0.07	2.11±0.55
COGP	3.26±0.25	2.68±0.24	0.68±0.05	1.30±0.21
VGPDS	3.19 ± 0.20	2.36 ± 0.08	0.65 ± 0.03	0.86 ± 0.66
VDM-GPDS	3.19±0.26	2.44±0.12	0.66±0.04	0.95±0.10

Table 11: Averaged RMSE and MSLL with the standard deviation for robot inverse dynamics learning with 100 training points.

30 inducing points are used for 100, 300 and 500 training points, respectively. For the VDM-GPDS and VGPDS, the dimensionality of the latent space is set to two. Table 11, 12 and 13 show the results for different methods in terms of averaged RMSE and MSLL. For intuition, we also plot the RMSE results in Figure 8.

From the tables and figures, we find that our model performs best on the whole, which confirms that the VDM-GPDS also works well for regression tasks. Comparing the VDM-

Method ($N = 300$)	2nd joint		3rd joint	
	RMSE	MSLL	RMSE	MSLL
sGPR	4.19±0.83	2.85±0.32	2.61±0.51	2.44±0.37
MTGP	4.37±0.89	3.36±0.35	2.53±0.56	2.77±0.19
COGP	3.78±0.14	5.08±0.79	2.27±0.09	3.49±0.43
VGPDs	3.67±0.18	2.62±0.04	2.11±0.12	2.06±0.04
VDM-GPDS	3.52 ± 0.09	2.59 ± 0.04	2.01 ± 0.06	2.00 ± 0.03
Method ($N = 300$)	4th joint		7th joint	
	RMSE	MSLL	RMSE	MSLL
sGPR	2.21±0.37	2.39±0.59	0.55±0.13	0.92±0.51
MTGP	2.01±0.20	2.40±0.12	0.47±0.04	1.27±0.19
COGP	2.25±0.18	2.65±0.25	0.52±0.02	1.71±0.21
VGPDs	2.31±0.38	2.12±0.15	0.48±0.06	0.59±0.10
VDM-GPDS	1.95 ± 0.09	1.92 ± 0.04	0.45 ± 0.01	0.53 ± 0.04

Table 12: Averaged RMSE and MSLL with the standard deviation for robot inverse dynamics learning with 300 training points.

Method ($N = 500$)	2nd joint		3rd joint	
	RMSE	MSLL	RMSE	MSLL
sGPR	4.01±0.71	2.78±0.27	2.08±0.45	2.05±0.32
MTGP	3.50±0.32	3.42±0.27	1.97±0.27	2.59±0.24
COGP	3.33±0.08	5.24±0.49	1.89±0.06	3.54±0.34
VGPDs	3.47±0.17	2.57±0.03	1.98±0.12	2.02±0.03
VDM-GPDS	3.20 ± 0.11	2.52 ± 0.04	1.77 ± 0.08	1.93 ± 0.04
Method ($N = 500$)	4th joint		7th joint	
	RMSE	MSLL	RMSE	MSLL
sGPR	2.08±0.38	2.28±0.51	0.47±0.08	0.76±0.45
MTGP	1.66±0.17	2.28±0.19	0.39±0.02	1.27±0.15
COGP	1.77±0.06	2.46±0.10	0.45±0.02	2.20±0.22
VGPDs	1.93±0.14	2.04±0.08	0.44±0.01	0.55±0.04
VDM-GPDS	1.64 ± 0.06	1.83 ± 0.02	0.39 ± 0.01	0.44 ± 0.03

Table 13: Averaged RMSE and MSLL with the standard deviation for robot inverse dynamics learning with 500 training points.

GPDS with the COGP, we further verify the assumption that the latent space can well grasp the characteristics of the data generation. Thus, the VDM-GPDS can well model the inverse dynamics model and make better prediction. Compared with the VGPDs, our model still shows advantages. This is attributed to the assumption of the dependency among

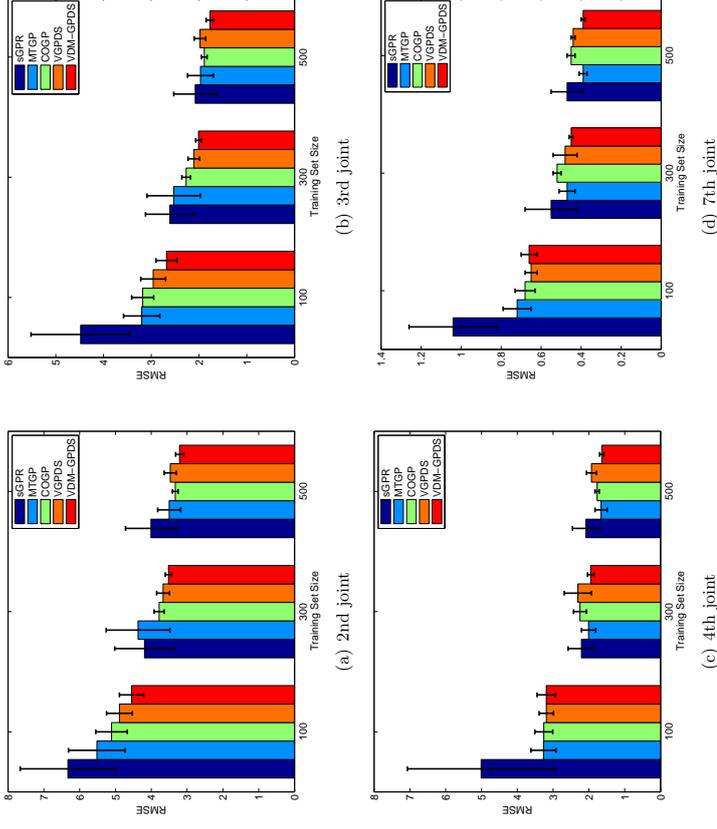


Figure 8: Averaged RMSE with the standard deviation for the 2nd, 3rd, 4th and 7th joint torques prediction. (better in color)

the multiple outputs. No matter for dynamical system modeling or static data regression, the proposed model is reasonable and applicable.

6.5 Performance and Efficiency Analysis

The proposed VDM-GPDS outperforms several previous methods for predicting outputs and recovering missing points for dynamical systems. In order to quantify the superior results, we evaluate the performance increases with the averaged performance increasing ratio to the VGPDs in terms of RMSE. The increasing ratios of the four experiments (Sections 6.1.1, 6.1.2, 6.2, 6.3 and 6.4) are 4.49%, 26.17%, 10.40%, 7.35% and 7.97%, respectively.

However, high effectiveness often comes together with low efficiency. The VDM-GPDS is a four-layer GP system that is more complex than the conventional methods. Particularly,

	CMOGP	GPDM	VGPPDS	VDM-GPDS
computational complexity	$O(D^3N^3)$	$O(N^3)$	$O(M^2NQ)$	$O(M^2NDQK)$
execution time	50.7	26.8	74.2	181.1

Table 14: Computational complexities and execution time (in ms) for different models.

since the proposed method explicitly models the dependency among outputs, the dependent multi-output covariance matrix in the VDM-GPDS is a full matrix with size $ND \times ND$ and operations involving it cannot be factorized. This is in contrast to the independent multi-output covariance matrix in the GPDM and VGPPDS, which is block-diagonal. As in Titsias (2009), inducing points are employed for the variational inference for the VDM-GPDS. The number of the inducing points M is much smaller than that of the data points N , which can improve the computational efficiency. For the VDM-GPDS, the most time-consuming calculation is to compute ψ_2 whose computational complexity is $O(M^2NDQK)$.

In order to give clear comparisons in terms of efficiency, we list the computational complexities of four models and the execution time (in ms) of one step for learning the models on the synthetic data in Table 14. Through the table, we find that the VDM-GPDS costs a lot. Nevertheless, our model can obtain high performance improvements as discussed above. We believe that getting performance improvements is worth the time cost.

7. Conclusion

In this paper, we have proposed a dependent multi-output GP for modeling complex dynamical systems. We give the reasonable assumption that the different outputs of the systems are generally dependent. The convolved process covariance function is employed to model the dependency among all the data points across all the outputs. We adapt the variational inference method involving inducing points to our model so that the latent variables are variationally integrated out. The model and variational parameters are jointly optimized with the scaled conjugate gradient method. Through small adaptations, our model can handle regression problems.

Modeling the possible dependency among multiple outputs can help to make better predictions. The effectiveness of the proposed model for complex dynamical systems is empirically demonstrated through multiple experiments. However, when the dimensionality of the output is very high, our model may take a long time to converge. This opens the possibility for future work to accelerate training for high dimensional dynamical systems.

Acknowledgments

The corresponding author Shitang Sun would like to thank supports from National Natural Science Foundation of China under Projects 61370175 and 61075005, and Shanghai Knowledge Service Platform Project (No. ZF1213).

Appendix A. Derivation of the Lower Bound

In order to approximately compute the marginal likelihood $p(\mathbf{Y}|\mathbf{t})$, we compute the variational lower bound of it by involving the variational distribution $q(F, U, X|Z)$. The variational lower bound \mathcal{L} can be expressed as

$$\begin{aligned} \mathcal{L} &= \int q(F, U, X|Z) \log \frac{p(\mathbf{Y}, F, U, X|\mathbf{t}, Z)}{q(F, U, X|Z)} dX dU dF \\ &= \int p(\mathbf{f}|\mathbf{u}, X, Z) q(\mathbf{u}) q(X) \log \frac{p(\mathbf{y}|\mathbf{f}) p(\mathbf{u}|Z) p(X|\mathbf{t})}{q(\mathbf{u}) q(X)} d\mathbf{f} d\mathbf{u} dX, \end{aligned} \quad (42)$$

since

$$p(\mathbf{Y}, F, U, X|\mathbf{t}, Z) = p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}|\mathbf{u}, X, Z) p(\mathbf{u}|Z) p(X|\mathbf{t}), \quad (43)$$

and

$$q(F, U, X|Z) = p(\mathbf{f}|\mathbf{u}, X, Z) q(\mathbf{u}) q(X). \quad (44)$$

For neatness, the above expression is split into two parts as $\mathcal{L} = \hat{\mathcal{L}} - \mathbf{KL}[q(X)||p(X|\mathbf{t})]$. Specifically, $\hat{\mathcal{L}}$ is expressed by

$$\hat{\mathcal{L}} = \int p(\mathbf{f}|\mathbf{u}, X, Z) q(\mathbf{u}) q(X) \log \frac{p(\mathbf{y}|\mathbf{f}) p(\mathbf{u}|Z)}{q(\mathbf{u})} d\mathbf{f} d\mathbf{u} dX. \quad (45)$$

$\mathbf{KL}[q(X)||p(X|\mathbf{t})]$ is the relative entropy of $q(X)$ and $p(X|\mathbf{t})$, expressed as

$$\begin{aligned} \mathbf{KL}[q(X)||p(X|\mathbf{t})] &= \int q(X) \log \frac{q(X)}{p(X|\mathbf{t})} \\ &= \frac{Q}{2} \log |\mathbf{K}_{\mathbf{t}, \mathbf{t}}| - \frac{1}{2} \sum_{q=1}^Q \log |S_q| \\ &\quad + \frac{1}{2} \sum_{q=1}^Q [\text{Tr}(\mathbf{K}_{\mathbf{t}, \mathbf{t}}^{-1} S_q) + \text{Tr}(\mathbf{K}_{\mathbf{t}, \mathbf{t}}^{-1} \boldsymbol{\mu}_q \boldsymbol{\mu}_q^T)] + \text{const}, \end{aligned} \quad (46)$$

since

$$p(X|\mathbf{t}) = \prod_{q=1}^Q \mathcal{N}(\mathbf{x}_q | \mathbf{0}, \mathbf{K}_{\mathbf{t}, \mathbf{t}}), \quad (47)$$

and

$$q(X) = \prod_{q=1}^Q \mathcal{N}(\mathbf{x}_q | \boldsymbol{\mu}_q, S_q). \quad (48)$$

So far, $\mathbf{KL}[q(X)||p(X|\mathbf{t})]$ can be calculated analytically as the above, we need to calculate $\hat{\mathcal{L}}$. By using the facts that $\log \frac{p(\mathbf{y}|\mathbf{f}) p(\mathbf{u}|Z)}{q(\mathbf{u})} = \log p(\mathbf{y}|\mathbf{f}) + \log \frac{p(\mathbf{u}|Z)}{q(\mathbf{u})}$ and $\int p(\mathbf{f}|\mathbf{u}, X, Z) d\mathbf{f} = 1$, $\hat{\mathcal{L}}$ is converted into

$$\begin{aligned} \hat{\mathcal{L}} &= \int q(\mathbf{u}) q(X) \int p(\mathbf{f}|\mathbf{u}, X, Z) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} d\mathbf{u} dX \\ &\quad + \int q(\mathbf{u}) q(X) \log \frac{p(\mathbf{u}|Z)}{q(\mathbf{u})} d\mathbf{u} dX. \end{aligned} \quad (49)$$

We know that $p(\mathbf{y}|\mathbf{f})$ and $p(\mathbf{f}|\mathbf{u}, X, Z)$ are both Gaussian, and then

$$\begin{aligned} & \int p(\mathbf{f}|\mathbf{u}, X, Z) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} \\ &= \log \mathcal{N}(\mathbf{y}|\mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}, \beta^{-1}I) - \frac{\beta}{2} \text{Tr}(\mathbf{K}_{\mathbf{f},\mathbf{f}} - \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}}). \end{aligned} \quad (50)$$

Thus $\hat{\mathcal{L}}$ in (49) can be simplified as

$$\begin{aligned} \hat{\mathcal{L}} &= \int q(\mathbf{u}) q(X) \log \frac{\mathcal{N}(\mathbf{y}|\mathbf{a}, B) p(\mathbf{u})}{q(\mathbf{u})} d\mathbf{u} dX \\ &\quad - \int \frac{\beta}{2} \text{Tr}(\mathbf{K}_{\mathbf{f},\mathbf{f}} - \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}}) q(X) dX, \end{aligned} \quad (51)$$

where $\mathbf{a} = \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}$ and $B = \beta^{-1}I$. By changing the integration order, we get

$$\begin{aligned} \hat{\mathcal{L}} &= \int q(\mathbf{u}) \left[\log \frac{e^{(\log \mathcal{N}(\mathbf{y}|\mathbf{a}, B))_{q(X)} p(\mathbf{u})}}{q(\mathbf{u})} \right] d(\mathbf{u}) \\ &\quad - \frac{\beta}{2} \text{Tr}(\langle \mathbf{K}_{\mathbf{f},\mathbf{f}} \rangle_{q(X)} - \langle \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}} \rangle_{q(X)}). \end{aligned} \quad (52)$$

We compute the optimal bound using the reserved Jensen's inequality as in Titsias and Lawrence (2010). This gives

$$\begin{aligned} \hat{\mathcal{L}} &\leq \log \int e^{(\log \mathcal{N}(\mathbf{y}|\mathbf{a}, B))_{q(X)} p(\mathbf{u})} d\mathbf{u} \\ &\quad - \frac{\beta}{2} \text{Tr}(\langle \mathbf{K}_{\mathbf{f},\mathbf{f}} \rangle_{q(X)}) + \frac{\beta}{2} \text{Tr}(\langle \mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}} \rangle_{q(X)}). \end{aligned} \quad (53)$$

The optimal distribution $q(\mathbf{u})$ that gives rise to this lower bound is given by $q(\mathbf{u}) = e^{(\log \mathcal{N}(\mathbf{y}|\mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}, \beta^{-1}I))_{q(X)} p(\mathbf{u})}$, which is analytically Gaussian

$$q(\mathbf{u}) \propto \mathcal{N}(\beta \mathbf{y}^\top \psi_1 \mathbf{K}_{\mathbf{u},\mathbf{u}} (\beta \psi_2 + \mathbf{K}_{\mathbf{u},\mathbf{u}})^{-1} \psi_1^\top \mathbf{y}, \mathbf{K}_{\mathbf{u},\mathbf{u}} (\beta \psi_2 + \mathbf{K}_{\mathbf{u},\mathbf{u}})^{-1}), \quad (54)$$

where $\psi_0 = \text{Tr}(\langle \mathbf{K}_{\mathbf{f},\mathbf{f}} \rangle_{q(X)})$, $\psi_1 = \langle \mathbf{K}_{\mathbf{f},\mathbf{u}} \rangle_{q(X)}$ and $\psi_2 = \langle \mathbf{K}_{\mathbf{u},\mathbf{u}} \mathbf{K}_{\mathbf{f},\mathbf{u}} \rangle_{q(X)}$. The closed-form of the lower bound of the approximated marginal log-likelihood defined as \mathcal{L} is given by

$$\begin{aligned} \mathcal{L} &= \log \left[\frac{\beta^{\frac{ND}{2}} |\mathbf{K}_{\mathbf{u},\mathbf{u}}|^{\frac{1}{2}}}{2\pi^{\frac{ND}{2}} |\beta \psi_2 + \mathbf{K}_{\mathbf{u},\mathbf{u}}|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2} \mathbf{y}^\top W \mathbf{y}\right\} \right] \\ &\quad - \frac{\beta \psi_0}{2} + \frac{\beta}{2} \text{Tr}(\langle \mathbf{K}_{\mathbf{u},\mathbf{u}} \rangle_{\psi_2}) - \mathbf{KL}[q(X)||p(X|\mathbf{t})], \end{aligned} \quad (55)$$

where $W = \beta I - \beta^2 \psi_1 (\beta \psi_2 + \mathbf{K}_{\mathbf{u},\mathbf{u}})^{-1} \psi_1^\top$. Given the above, we can obtain the final formulation of the lower bound in (21) which has the similar formulation with Damianou et al. (2011). But actually they are not the same.

Appendix B. Gradients with Respect to the Parameters

The parameters involved in the proposed model include the model parameters $\{\beta, \boldsymbol{\theta}_x, \boldsymbol{\theta}_f\}$ and the variational parameters $\{\boldsymbol{\mu}_q, \lambda_q\}_{q=1}^Q$ after reparameterizing $\{\boldsymbol{\mu}_q, S_q\}_{q=1}^Q$. All the parameters are jointly optimized by maximizing the lower bound in (21) with the scaled conjugate gradient method. Here we give the detailed gradients of all the parameters. Note that in our model the dimensionality of the latent variable \mathbf{U} is one ($K=1$). So the statistics such as $S_{d,k}$, Λ_k , W are changed to s_d , Λ , M here. In order to simplify the expressions, we define $\Sigma_{\psi_0} = 2P_d + \Lambda$, $\Sigma_{\psi_1} = P_d + \Lambda + S_n$, $\Sigma_{\psi_{21}} = 2(P_d + \Lambda)$, $\Sigma_{\psi_{22}} = \frac{P_d + \Lambda}{2} + S_n$, $C^{-1} = \beta^{-1} \mathbf{K}_{\mathbf{u},\mathbf{u}} + \psi_2$. \mathbf{z} is equivalent to \mathbf{z}_m and \mathbf{z}' is equivalent to $\mathbf{z}_{m'}$. The symbol $[\cdot]_q$ after a matrix means the q th column of the matrix.

Because of the reparameterization, we need to calculate the gradients of $\hat{\mathcal{L}}$ with respect to $\boldsymbol{\mu}_q$, S_q and then obtain the gradients of \mathcal{L} with respect to $\boldsymbol{\mu}_q$, λ_q using (28) and (29). Given that $\mathbf{KL}[q(X)||p(X|\mathbf{t})]$ does not involve the parameters $\boldsymbol{\theta}_f$, β and Z , its gradients with respect to $\boldsymbol{\theta}_f$, β and Z are zero. Therefore, $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_f} = \frac{\partial \hat{\mathcal{L}}}{\partial \boldsymbol{\theta}_f}$, $\frac{\partial \mathcal{L}}{\partial \beta} = \frac{\partial \hat{\mathcal{L}}}{\partial \beta}$ and $\frac{\partial \mathcal{L}}{\partial Z} = \frac{\partial \hat{\mathcal{L}}}{\partial Z}$.

First, we give the gradients of $\hat{\mathcal{L}}$ with respect to $\boldsymbol{\mu}_q$, S_q , P_d , s_d through the formulation

$$\frac{\partial \hat{\mathcal{L}}}{\partial \boldsymbol{\theta}} = -\frac{\beta}{2} \frac{\partial \psi_0}{\partial \boldsymbol{\theta}} + \beta \text{Tr} \left[\frac{\partial \psi_1^\top}{\partial \boldsymbol{\theta}} \mathbf{y} \mathbf{y}^\top \psi_1 C \right] + \frac{\beta}{2} \text{Tr} \left[\frac{\partial \psi_2}{\partial \boldsymbol{\theta}} (\mathbf{K}_{\mathbf{u},\mathbf{u}}^{-1} - \frac{C}{\beta} - C \psi_1^\top \mathbf{y} \mathbf{y}^\top \psi_1 C) \right], \quad (56)$$

where $\boldsymbol{\theta}$ represents $\boldsymbol{\mu}_{nq}$, S_{nq} , P_{dq} and s_d . The detailed derivatives of ψ_0 , ψ_1 and ψ_2 with respect to $\boldsymbol{\mu}_{nq}$, S_{nq} , P_{dq} and s_d are different. The derivatives of ψ_0 , $(\psi_1)_{vm}$ and $(\psi_2)_{mm'}$ with respect to $\boldsymbol{\mu}_{nq}$ are

$$\begin{aligned} \frac{\partial \psi_0}{\partial \boldsymbol{\mu}_{nq}} &= 0, \\ \frac{\partial (\psi_1)_{vm}}{\partial \boldsymbol{\mu}_{nq}} &= s_d \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_n, \Sigma_{\psi_1}) (\mathbf{z} - \boldsymbol{\mu}_n)^\top \Sigma_{\psi_1}^{-1} \mathbf{1}_q, \\ \frac{\partial (\psi_2)_{mm'}}{\partial \boldsymbol{\mu}_{nq}} &= \sum_{d=1}^D s_d^2 \mathcal{N}(\mathbf{z}|\mathbf{z}', \Sigma_{\psi_{21}}) \mathcal{N} \left(\frac{\mathbf{z} + \mathbf{z}'}{2} | \boldsymbol{\mu}_n, \Sigma_{\psi_{22}} \right) \left(\left(\frac{\mathbf{z} + \mathbf{z}'}{2} - \boldsymbol{\mu}_n \right)^\top \Sigma_{\psi_{22}}^{-1} \mathbf{1}_q \right). \end{aligned} \quad (57)$$

The derivatives of ψ_0 , $(\psi_1)_{vm}$ and $(\psi_2)_{mm'}$ with respect to S_{nq} are

$$\begin{aligned} \frac{\partial \psi_0}{\partial S_{nq}} &= 0, \\ \frac{\partial (\psi_1)_{vm}}{\partial S_{nq}} &= s_d \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_n, \Sigma_{\psi_1}) \left(-\frac{|\Sigma_{\psi_1}^{-1} \partial \Sigma_{\psi_1}|}{2} \frac{1}{\partial S_{nq}} - \frac{1}{2} (\mathbf{z} - \boldsymbol{\mu}_n)^\top \frac{\partial \Sigma_{\psi_1}^{-1}}{\partial S_{nq}} (\mathbf{z} - \boldsymbol{\mu}_n) \right), \\ \frac{\partial (\psi_2)_{mm'}}{\partial S_{nq}} &= \sum_{d=1}^D s_d^2 \mathcal{N}(\mathbf{z}|\mathbf{z}', \Sigma_{\psi_{21}}) \mathcal{N} \left(\frac{\mathbf{z} + \mathbf{z}'}{2} | \boldsymbol{\mu}_n, \Sigma_{\psi_{22}} \right) \\ &\quad \left(-\frac{|\Sigma_{\psi_{22}}^{-1} \partial \Sigma_{\psi_{22}}|}{2} \frac{1}{\partial S_{nq}} - \frac{1}{2} (\mathbf{z} + \mathbf{z}')^\top \frac{\partial \Sigma_{\psi_{22}}^{-1}}{\partial S_{nq}} (\mathbf{z} + \mathbf{z}') - \boldsymbol{\mu}_n \right). \end{aligned} \quad (58)$$

The derivatives of ψ_0 , $(\psi_1)_{om}$ and $(\psi_2)_{mm'}$ with respect to P_{dq} are

$$\begin{aligned} \frac{\partial \psi_0}{\partial P_{dq}} &= \sum_{d=1}^D \frac{N}{2} \frac{s_d s_d}{|2\pi|^{\frac{Q}{2}} |\Sigma_{\psi_0}|^{\frac{3}{2}}} \frac{\partial |\Sigma_{\psi_0}|}{\partial P_{dq}}, \\ \frac{\partial (\psi_1)_{om}}{\partial P_{dq}} &= s_d \mathcal{N}(\mathbf{z} | \boldsymbol{\mu}_n, \Sigma_{\psi_1}) \left(-\frac{1}{2} \frac{|\Sigma_{\psi_1}|^{-1} \partial |\Sigma_{\psi_1}|}{\partial P_{dq}} - \frac{1}{2} (\mathbf{z} - \boldsymbol{\mu}_n)^\top \frac{\partial \Sigma_{\psi_1}^{-1}}{\partial P_{dq}} (\mathbf{z} - \boldsymbol{\mu}_n) \right), \\ \frac{\partial (\psi_2)_{mm'}}{\partial P_{dq}} &= \sum_{n=1}^N s_d^2 \mathcal{N}(\mathbf{z} | \boldsymbol{\mu}_n, \Sigma_{\psi_2}) \mathcal{N} \left(\frac{\mathbf{z} + \mathbf{z}'}{2} \middle| \boldsymbol{\mu}_n, \Sigma_{\psi_{22}} \right) \\ &\quad \left(-\frac{|\Sigma_{\psi_{21}}|^{-1} \partial |\Sigma_{\psi_{21}}|}{2 \partial P_{dq}} - \frac{1}{2} (\mathbf{z} - \mathbf{z}')^\top \frac{\partial \Sigma_{\psi_{21}}^{-1}}{\partial P_{dq}} (\mathbf{z} - \mathbf{z}') \right) \\ &\quad - \frac{|\Sigma_{\psi_{22}}|^{-1} \partial |\Sigma_{\psi_{22}}|}{2 \partial P_{dq}} - \frac{1}{2} (\mathbf{z} + \mathbf{z}')^\top \frac{\partial \Sigma_{\psi_{22}}^{-1}}{\partial P_{dq}} (\mathbf{z} + \mathbf{z}') - \boldsymbol{\mu}_n \Big). \end{aligned} \quad (59)$$

The derivatives of ψ_0 , $(\psi_1)_{om}$ and $(\psi_2)_{mm'}$ with respect to s_d are

$$\begin{aligned} \frac{\partial \psi_0}{\partial s_d} &= \frac{2N s_d}{|2\pi|^{\frac{Q}{2}} |\Sigma_{\psi_0}|^{\frac{3}{2}}}, \\ \frac{\partial (\psi_1)_{om}}{\partial s_d} &= \mathcal{N}(\mathbf{z} | \boldsymbol{\mu}_n, \Sigma_{\psi_1}), \\ \frac{\partial (\psi_2)_{mm'}}{\partial s_d} &= \sum_{n=1}^N 2s_d \mathcal{N}(\mathbf{z} | \boldsymbol{\mu}_n, \Sigma_{\psi_{21}}) \mathcal{N} \left(\frac{\mathbf{z} + \mathbf{z}'}{2} \middle| \boldsymbol{\mu}_n, \Sigma_{\psi_{22}} \right). \end{aligned} \quad (60)$$

Then, we give the gradients of \mathcal{L} with respect to Λ and Z through the formulation

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \theta} &= -\frac{\beta}{2} \frac{\partial \psi_0}{\partial \theta} + \beta \text{Tr} \left[\frac{\partial \psi_1^\top}{\partial \theta} \mathbf{y} \mathbf{y}^\top \psi_1 C \right] + \frac{\beta}{2} \text{Tr} \left[\frac{\partial \psi_2}{\partial \theta} (\mathbf{K}_{\mathbf{u}, \mathbf{u}}^{-1} - C \psi_1^\top \mathbf{y} \mathbf{y}^\top \psi_1 C - \beta^{-1} C) \right] \\ &\quad + \frac{1}{2} \text{Tr} \left[\frac{\partial \mathbf{K}_{\mathbf{u}, \mathbf{u}}}{\partial \theta} (\mathbf{K}_{\mathbf{u}, \mathbf{u}}^{-1} - C \psi_1^\top \mathbf{y} \mathbf{y}^\top \psi_1 C - \beta^{-1} C - \beta \mathbf{K}_{\mathbf{u}, \mathbf{u}}^{-1} \psi_2 \mathbf{K}_{\mathbf{u}, \mathbf{u}}^{-1}) \right], \end{aligned} \quad (61)$$

where θ represents Λ_q , Z_{mq} . The detailed derivatives of the ψ_0 , ψ_1 , ψ_2 and $\mathbf{K}_{\mathbf{u}, \mathbf{u}}$ with respect to Λ_q , Z_{mq} are given separately. The derivatives of ψ_0 , $(\psi_1)_{om}$, $(\psi_2)_{mm'}$ and $(\mathbf{K}_{\mathbf{u}, \mathbf{u}})_{mm'}$

with respect to Λ_q are

$$\begin{aligned} \frac{\partial \psi_0}{\partial \Lambda_q} &= \sum_{d=1}^D \frac{N}{2} \frac{s_d s_d}{|2\pi|^{\frac{Q}{2}} |\Sigma_{\psi_0}|^{\frac{3}{2}}} \frac{\partial |\Sigma_{\psi_0}|}{\partial \Lambda_q}, \\ \frac{\partial (\psi_1)_{om}}{\partial \Lambda_q} &= s_d \mathcal{N}(\mathbf{z} | \boldsymbol{\mu}_n, \Sigma_{\psi_1}) \left(-\frac{1}{2} \frac{|\Sigma_{\psi_1}|^{-1} \partial |\Sigma_{\psi_1}|}{\partial \Lambda_q} - \frac{1}{2} (\mathbf{z} - \boldsymbol{\mu}_n)^\top \frac{\partial \Sigma_{\psi_1}^{-1}}{\partial \Lambda_q} (\mathbf{z} - \boldsymbol{\mu}_n) \right), \\ \frac{\partial (\psi_2)_{mm'}}{\partial \Lambda_q} &= \sum_{n=1}^N \sum_{d=1}^D s_d^2 \mathcal{N}(\mathbf{z} | \boldsymbol{\mu}_n, \Sigma_{\psi_{21}}) \mathcal{N} \left(\frac{\mathbf{z} + \mathbf{z}'}{2} \middle| \boldsymbol{\mu}_n, \Sigma_{\psi_{22}} \right) \\ &\quad \left(-\frac{\partial |\Sigma_{\psi_{21}}|}{2 |\Sigma_{\psi_{21}}| \partial \Lambda_q} - \frac{1}{2} (\mathbf{z} - \mathbf{z}')^\top \frac{\partial \Sigma_{\psi_{21}}^{-1}}{\partial \Lambda_q} (\mathbf{z} - \mathbf{z}') \right) \\ &\quad - \frac{\partial |\Sigma_{\psi_{22}}|}{2 |\Sigma_{\psi_{22}}| \partial \Lambda_q} - \frac{1}{2} (\mathbf{z} + \mathbf{z}')^\top \frac{\partial \Sigma_{\psi_{22}}^{-1}}{\partial \Lambda_q} (\mathbf{z} + \mathbf{z}') - \boldsymbol{\mu}_n \Big), \\ \frac{\partial (\mathbf{K}_{\mathbf{u}, \mathbf{u}})_{mm'}}{\partial \Lambda_q} &= \mathcal{N}(\mathbf{z} | \boldsymbol{\mu}_n, \Lambda) \left(-\frac{1}{2} |\Lambda|^{-1} \frac{\partial |\Lambda|}{\partial \Lambda_q} - \frac{1}{2} (\mathbf{z} - \mathbf{z}')^\top \frac{\partial \Lambda^{-1}}{\partial \Lambda_q} (\mathbf{z} - \mathbf{z}') \right). \end{aligned} \quad (62)$$

The derivatives of ψ_0 , $(\psi_1)_{om}$, $(\psi_2)_{mm'}$ and $(\mathbf{K}_{\mathbf{u}, \mathbf{u}})_{mm'}$ with respect to z_{mq} are

$$\begin{aligned} \frac{\partial \psi_0}{\partial z_{mq}} &= 0, \\ \frac{\partial (\psi_1)_{om}}{\partial z_{mq}} &= -s_d \mathcal{N}(\mathbf{z} | \boldsymbol{\mu}_n, \Sigma_{\psi_1}) \left((\mathbf{z} - \boldsymbol{\mu}_n)^\top \Sigma_{\psi_1}^{-1} \mathbf{q} \right), \\ \frac{\partial (\psi_2)_{mm'}}{\partial z_{mq}} &= \sum_{d=1}^D s_d^2 \mathcal{N}(\mathbf{z} | \boldsymbol{\mu}_n, \Sigma_{\psi_{21}}) \mathcal{N} \left(\frac{\mathbf{z} + \mathbf{z}'}{2} \middle| \boldsymbol{\mu}_n, \Sigma_{\psi_{22}} \right) \\ &\quad \left(-\frac{1}{2} (\mathbf{z} + \mathbf{z}')^\top \Sigma_{\psi_{22}}^{-1} \mathbf{q} + (\mathbf{z} - \mathbf{z}')^\top \Sigma_{\psi_{21}}^{-1} \mathbf{q} \right), \\ \frac{\partial (\mathbf{K}_{\mathbf{u}, \mathbf{u}})_{mm'}}{\partial z_{mq}} &= -\frac{1}{\Lambda_q} \mathcal{N}(\mathbf{z} | \boldsymbol{\mu}_n, \Lambda) (z_{mq} - z_{m'q}). \end{aligned} \quad (63)$$

Finally, we give the gradients of \mathcal{L} with respect to β and $\boldsymbol{\theta}_x$ as follows.

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \beta} &= \frac{1}{2} \text{Tr} (\mathbf{K}_{\mathbf{u}, \mathbf{u}}^{-1} \psi_2) + (Y - M) \beta^{-1} - \text{Tr} (\mathbf{y} \mathbf{y}^\top) + \text{Tr} (C \psi_1^\top \mathbf{y} \mathbf{y}^\top \psi_1) \\ &\quad + \beta^{-2} \text{Tr} (\mathbf{K}_{\mathbf{u}, \mathbf{u}} C) + \beta^{-1} \text{Tr} (\mathbf{K}_{\mathbf{u}, \mathbf{u}} C \psi_1^\top \mathbf{y} \mathbf{y}^\top \psi_1 C), \\ \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_x} &= \sum_{q=1}^Q \text{Tr} \left[-\frac{1}{2} (\hat{\beta}_q \mathbf{K}_{\mathbf{t}, \mathbf{t}} \hat{\beta}_q + \hat{\boldsymbol{\mu}}_q \hat{\boldsymbol{\mu}}_q^\top) + (I - \hat{\beta}_q \mathbf{K}_{\mathbf{t}, \mathbf{t}}) \frac{\partial \hat{\mathcal{L}}}{\partial S_q} (I - \hat{\beta}_q \mathbf{K}_{\mathbf{t}, \mathbf{t}})^\top \frac{\partial \mathbf{K}_{\mathbf{t}, \mathbf{t}}}{\partial \boldsymbol{\theta}_x} \right] \\ &\quad + \left(\frac{\partial \hat{\mathcal{L}}}{\partial \boldsymbol{\mu}_q} \right)^\top \frac{\partial \mathbf{K}_{\mathbf{t}, \mathbf{t}}}{\partial \boldsymbol{\theta}_x} \bar{\boldsymbol{\mu}}_q, \end{aligned} \quad (64)$$

where $\hat{\beta}_q = \Lambda_q^{\frac{1}{2}} (I + \Lambda_q^{\frac{1}{2}} \mathbf{K}_{\mathbf{t}, \mathbf{t}} \Lambda_q^{\frac{1}{2}})^{-1} \Lambda_q^{\frac{1}{2}}$.

Appendix C. Derivations of Prediction with Only Time

With the parameters as well as time \mathbf{t} and \mathbf{t}_* omitted, the posterior density for prediction is given by

$$p(Y_*|Y) = \int p(Y_*|F_*)p(F_*|X_*, Y)p(X_*|Y)dF_*dX_* \quad (66)$$

where $F_* \in \mathbb{R}^{N_* \times D}$ denotes the set of latent variables (the noise-free version of Y_*) and $X_* \in \mathbb{R}^{N_* \times Q}$ represents the latent variables in the low dimensional space.

The distribution $p(F_*|X_*, Y)$ in (66) is approximated by the variational distribution

$$p(F_*|X_*, Y) \approx q(\mathbf{f}_*|X_*) = \int p(\mathbf{f}_*|\mathbf{u}, X_*)q(\mathbf{u})d\mathbf{u} \quad (67)$$

where $\mathbf{f}_*^\top = [\mathbf{f}_{*1}^\top, \dots, \mathbf{f}_{*D}^\top]$, and $p(\mathbf{f}_*|\mathbf{u}, X_*)$ is Gaussian with the formulation

$$p(\mathbf{f}_*|\mathbf{u}, X_*) = \mathcal{N}(\mathbf{f}_*|\mathbf{K}_{\mathbf{f}_*, \mathbf{u}}\mathbf{K}_{\mathbf{u}, \mathbf{u}}^{-1}\mathbf{u}, \mathbf{K}_{\mathbf{f}_*, \mathbf{f}_*} - \mathbf{K}_{\mathbf{f}_*, \mathbf{u}}\mathbf{K}_{\mathbf{u}, \mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u}, \mathbf{f}_*}). \quad (68)$$

Since the optimal setting for $q(\mathbf{u})$ in our variational framework is also found to be Gaussian, $q(\mathbf{f}_*|X_*)$ is Gaussian that can be computed analytically

$$q(\mathbf{f}_*|X_*) = \mathcal{N}(\beta\mathbf{K}_{\mathbf{f}_*, \mathbf{u}}(\mathbf{K}_{\mathbf{u}, \mathbf{u}} + \beta\psi_2)^{-1}\psi_1^\top\mathbf{y}, \mathbf{K}_{\mathbf{f}_*, \mathbf{u}}(\mathbf{K}_{\mathbf{u}, \mathbf{u}} + \beta\psi_2)^{-1}\mathbf{K}_{\mathbf{f}_*, \mathbf{u}}^\top - \mathbf{K}_{\mathbf{f}_*, \mathbf{f}_*} - \mathbf{K}_{\mathbf{f}_*, \mathbf{u}}\mathbf{K}_{\mathbf{u}, \mathbf{u}}^{-1}\mathbf{K}_{\mathbf{u}, \mathbf{f}_*}). \quad (69)$$

The distribution $p(X_*|Y)$ in (66) is approximated by the variational distribution $q(X_*)$ which is Gaussian and can be explicitly formulated as

$$q(X_*) = \mathcal{N}(\boldsymbol{\mu}_{X_*}, \boldsymbol{\Sigma}_{X_*}), \quad (70)$$

where $\boldsymbol{\mu}_{X_*}$ is composed of column vector $\boldsymbol{\mu}_{\mathbf{x}_{*q}}$ and block-diagonal matrix $\boldsymbol{\Sigma}_{X_*}$ has diagonal element $\boldsymbol{\Sigma}_{\mathbf{x}_{*q}}$ with

$$\boldsymbol{\mu}_{\mathbf{x}_{*q}} = \mathbf{K}_{\mathbf{t}_*, \mathbf{t}_*}\mathbf{K}_{\mathbf{t}_*, \mathbf{t}}^{-1}\boldsymbol{\mu}_q, \quad (71)$$

$$\boldsymbol{\Sigma}_{\mathbf{x}_{*q}} = \mathbf{K}_{\mathbf{t}_*, \mathbf{t}_*} - \mathbf{K}_{\mathbf{t}_*, \mathbf{t}}\mathbf{K}_{\mathbf{t}, \mathbf{t}}^{-1}(\mathbf{K}_{\mathbf{t}, \mathbf{t}_*} - \mathbf{S}_q\mathbf{K}_{\mathbf{t}, \mathbf{t}}^{-1}\mathbf{K}_{\mathbf{t}, \mathbf{t}_*}). \quad (72)$$

Given $p(F_*|X_*, Y)$ approximated by $q(\mathbf{f}_*|X_*)$ and $p(X_*|Y)$ approximated by $q(X_*)$, the posterior density of \mathbf{f}_* (the noise-free version of \mathbf{y}_*) is now approximated by

$$p(\mathbf{f}_*|Y) = \int q(\mathbf{f}_*|X_*)q(X_*)dX_*. \quad (73)$$

So far, following Damianou et al. (2011), the expectation of \mathbf{f}_* and its element-wise autocovariance are given in (33) and (34).

References

M. A. Álvarez and N. D. Lawrence. Computationally efficient convolved multiple output Gaussian processes. *Journal of Machine Learning Research*, 12:1459–1500, 2011.

- M. A. Álvarez, D. Luengo, and N. D. Lawrence. Latent force models. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, pages 9–16, 2009.
- M. A. Álvarez, D. Luengo, M. K. Titsias, and N. D. Lawrence. Efficient multioutput Gaussian processes through variational inducing kernels. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pages 25–32, 2010.
- M. A. Álvarez, J. Peters, B. Schölkopf, and N. D. Lawrence. Switched latent force models for movement segmentation. *Advances in Neural Information Processing Systems*, 23: 55–63, 2011.
- M. A. Álvarez, L. Rosasco, and N. D. Lawrence. Kernels for vector-valued functions: a review. *Foundations and Trends® in Machine Learning*, 4:195–266, 2012.
- M. A. Álvarez, D. Luengo, and N. D. Lawrence. Linear latent force models using Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:2693–2705, 2013.
- C. Archambeau, D. Cornford, M. Opper, and J. Shawe-Taylor. Gaussian process approximations of stochastic differential equations. *Journal of Machine Learning Research Workshop and Conference Proceedings*, 1:1–16, 2007.
- C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.
- E. V. Bonilla, K. M. A. Chai, and C. K. I. Williams. Multi-task Gaussian process prediction. *Advances in Neural Information Processing Systems*, 20:153–160, 2007.
- P. Boyle. *Gaussian Process for Regression and Optimisation*. PhD thesis, Victoria University of Wellington, 2007.
- K. M. A. Chai, C. K. I. Williams, S. Klanke, and S. Vijayakumar. Multi-task Gaussian process learning of robot inverse dynamics. *Advances in Neural Information Processing Systems*, 21:265–272, 2009.
- L. Csató and M. Opper. Sparse representation for Gaussian process models. *Advances in Neural Information Processing Systems*, 13:444–450, 2001.
- A. C. Damianou, M. K. Titsias, and N. D. Lawrence. Variational Gaussian process dynamical systems. *Advances in Neural Information Processing Systems*, 24:2510–2518, 2011.
- A. C. Damianou, M. K. Titsias, and N. D. Lawrence. Variational inference for uncertainty on the inputs of Gaussian process models. <http://arxiv.org/abs/1409.2287>, 2014.
- M. P. Deisenroth and S. Mohamed. Expectation propagation in Gaussian process dynamical systems. *Advances in Neural Information Processing Systems*, 25:2618–2626, 2012.
- N. Gamage, T. C. Kuang, R. Akmeiliawati, and S. Demidenko. Gaussian process dynamical models for hand gesture interpretation in sign language. *Pattern Recognition Letters*, 32: 2009–2014, 2011.

- J. Hartikainen and S. Särkkä. Sequential inference for latent force models. <http://arxiv.org/abs/1202.3730>, 2012.
- G. E. Henter, M. R. Frean, and W. B. Kleijn. Gaussian process dynamical models for nonparametric speech representation and synthesis. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4505–4508, 2012.
- N. D. Lawrence. Gaussian process latent variable models for visualisation of high dimensional data. *Advances in Neural Information Processing Systems*, 17:329–336, 2004.
- N. D. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research*, 6:1783–1816, 2005.
- N. D. Lawrence. Learning for larger dataset with the Gaussian process latent variable model. In *Proceedings of the 11th International Workshop on Artificial Intelligence and Statistics*, pages 243–250, 2007.
- M. Lázaro-gredilla. Bayesian warped Gaussian processes. *Advances in Neural Information Processing Systems*, 25:1628–1636, 2012.
- J. Luttinen and A. Iljin. Efficient Gaussian process inference for short-scale spatio-temporal modeling. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, pages 741–750, 2012.
- V. T. Nguyen and E. Bonilla. Collaborative multi-output Gaussian processes. In *Proceedings of Uncertainty in Artificial Intelligence*, pages 1–10, 2014.
- M. Opper and A. Archanbear. The variational Gaussian approximation revisited. *Neural Computation*, 21:786–792, 2009.
- H. Park, S. Yun, S. Park, J. Kim, and C. D. Yoo. Phoneme classification using constrained variational Gaussian process dynamical system. *Advances in Neural Information Processing Systems*, 25:2015–2023, 2012.
- J. Quiñero-Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.
- C. E. Rasmussen and C. K. I. Williams. *Gaussian Process for Machine Learning*. MIT Press, 2006.
- L. Scavićco and B. Vijayakumar. *Modeling and Control of Robot Manipulators*. Springer, 2000.
- E. Snelson and Z. Ghahramani. Sparse Gaussian process using pseudo-inputs. *Advances in Neural Information Processing Systems*, 18:444–450, 2006.
- E. Snelson, Z. Ghahramani, and C. Rasmussen. Warped Gaussian processes. *Advances in Neural Information Processing Systems*, 16:337–344, 2003.
- S. Sun, C. Zhang, and G. Yu. A Bayesian network approach to traffic flow forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 7:124–132, 2006.
- G. W. Taylor, G. E. Hinton, and S. Roweis. Modeling human motion using binary latent variables. *Advances in Neural Information Processing Systems*, 19:1345–1352, 2006.
- M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society*, 61:611–622, 1999.
- M. K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, pages 567–574, 2009.
- M. K. Titsias and N. D. Lawrence. Bayesian Gaussian process latent variable model. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pages 844–851, 2010.
- R. Urtasun, D. J. Fleet, and P. Fua. 3D people tracking with Gaussian process dynamic models. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 238–245, 2006.
- S. Vijayakumar and S. Schaal. Locally weighted projection regression: An $O(n)$ algorithm for incremental real time learning in high dimensional space. In *Proceedings of the 17th International Conference on Machine Learning*, pages 1079–1086, 2000.
- J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models. *Advances in Neural Information Processing Systems*, 19:1441–1448, 2006.
- J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:283–398, 2008.
- B. M. Williams. *Modeling and forecasting vehicular traffic flow as a seasonal stochastic time series process*. PhD thesis, University of Virginia, 1999.
- A. G. Wilson, D. A. Knowles, and Z. Ghahramani. Gaussian process regression networks. In *Proceedings of the 29th International Conference on Machine Learning*, pages 599–606, 2012.
- J. Zhao and S. Sun. Variational dependent multi-output Gaussian process dynamical systems. In *Proceedings of the 17th International Conference on Discovery Science*, pages 350–361, 2014.

Multiple Output Regression with Latent Noise

Jussi Gillberg

Pekka Marttinen

Helsinki Institute for Information Technology HIIT

Department of Computer Science

PO Box 15600, Aalto University, 00076 Aalto, Finland

JUSSI.GILLBERG@AALTO.FI

PEKKA.MARTTINEN@AALTO.FI

Matti Pirinen

Institute for Molecular Medicine Finland (FIMM)

University of Helsinki, Finland

MATTI.PIRINEN@HELSINKI.FI

Antti J. Kangas

Pasi Soininen *

Computational Medicine

Faculty of Medicine

University of Oulu & Biocenter Oulu, Oulu, Finland

ANTTI.KANGAS@COMPUTATIONALMEDICINE.FI

PASI.SOININEN@COMPUTATIONALMEDICINE.FI

Mehreen Ali

Institute for Molecular Medicine Finland (FIMM)

University of Helsinki, Finland

MEHREEN.ALI@HELSINKI.FI

Aki S. Havulinna

Department of Health

National Institute for Health and Welfare, Helsinki, Finland

AKI.HAVULINNA@THL.FI

Marjo-Riitta Järvelin *

Department of Epidemiology and Biostatistics

MRC-PHE Centre for Environment & Health, School of Public Health,

Imperial College London, UK

M.JARVELIN@IMPERIAL.AC.UK

Mika Ala-Korpela *

Computational Medicine

Faculty of Medicine

University of Oulu & Biocenter Oulu, Oulu, Finland

MIKA.ALA-KORPELA@COMPUTATIONALMEDICINE.FI

Samuel Kaski

Helsinki Institute for Information Technology HIIT

Department of Computer Science

PO Box 15600, Aalto University, 00076 Aalto, Finland

SAMUEL.KASKI@AALTO.FI

Editor: Karsten Borgwardt

Abstract

In high-dimensional data, structured noise caused by observed and unobserved factors affecting multiple target variables simultaneously, imposes a serious challenge for modeling, by masking the often weak signal. Therefore, (1) explaining away the structured noise in multiple-output regression is of paramount importance. Additionally, (2) assumptions about the correlation structure of the regression weights are needed. We note that both can be formulated in a natural way in a latent variable model, in which both the interesting signal and the noise are mediated through the same latent factors. Under this assumption, the signal model then borrows strength from the noise model by encouraging similar effects on correlated targets. We introduce a hyperparameter for the *latent signal-to-noise ratio* which turns out to be important for modelling weak signals, and an ordered infinite-dimensional shrinkage prior that resolves the rotational unidentifiability in reduced-rank regression models. Simulations and prediction experiments with metabolite, gene expression, fMRI measurement, and macroeconomic time series data show that our model equals or exceeds the state-of-the-art performance and, in particular, outperforms the standard approach of assuming independent noise and signal models.

Keywords: Bayesian reduced-rank regression, latent variable models, latent signal-to-noise ratio, multiple-output regression, nonparametric Bayes, shrinkage priors, structured noise, weak effects

1. Introduction

Explaining away structured noise is one of the cornerstones for successful modeling of high-dimensional output data in the regression framework (Fusi et al., 2012; Klami et al., 2013; Rai et al., 2012; Rakitsch et al., 2013; Stegle et al., 2012; Virtanen et al., 2011). The structured noise refers to dependencies between response variables, which are unrelated to the dependencies of interest between the response variables and the covariates. It is noise caused by observed and unobserved confounders that affect multiple variables simultaneously. Common observed confounders in medical and biological data include age and sex of an individual, whereas unobserved confounders include, for example, the state of the cell being measured, measurement artifacts influencing multiple probes, or other unrecorded experimental conditions. When not accounted for, structured noise may both hide interesting relationships and result in spurious findings (Leek and Storey, 2007; Kang et al., 2008).

The effects of known confounders can be removed straightforwardly by using supervised methods. For the unobserved confounders, a routinely used approach for explaining away structured noise has been to assume *a priori* independent effects for the interesting and uninteresting factors. For example, in the factor regression setup (West, 2003; Stegle et al., 2010; Fusi et al., 2012), the target variables Y are assumed to have been generated as

$$Y = X\Theta + H\Lambda + E, \quad (1)$$

where $Y_{N \times K}$ is the matrix of K target variables (or dependent variables) and $X_{N \times P}$ contains the covariates (or independent variables), for the N observations. The model parameter matrix $H_{N \times S_2}$ comprises the unknown latent factors and $\Lambda_{S_2 \times K}$ the factor loadings, which are used to model away the structured noise. The term $E_{N \times K}$ represents independent unstructured noise and the elements of E are independently distributed, $\text{vec}(E) \sim \mathcal{N}(0, I_{NK})$. In this paper we call this model **independent-noise BRRR**. To reduce the effective number of parameters in the regression coefficient matrix $\Theta_{P \times K}$, a low-rank structure may be

*. PS and MAK are also at NMR Metabolomics Laboratory, School of Pharmacy, University of Eastern Finland, Kuopio, Finland; MRJ is also at Center for Life Course Epidemiology, Faculty of Medicine, University of Oulu, Finland and Biocenter Oulu, University of Oulu, Finland and Unit of Primary Care, Oulu University Hospital, Oulu, Finland; MAK is also at Computational Medicine, School of Social and Community Medicine and the Medical Research Council Integrative Epidemiology Unit, University of Bristol, Bristol, UK

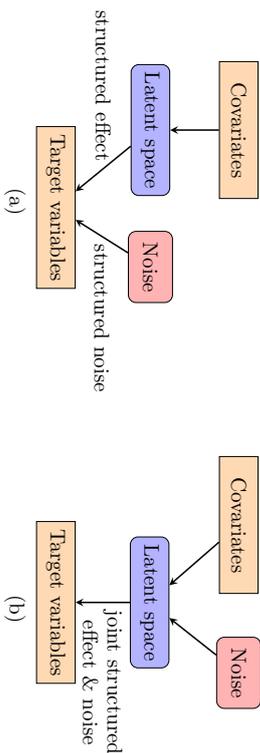


Figure 1: Illustration of (a) *a priori* independent interesting and uninteresting effects and (b) the latent noise assumption. Latent noise is mediated to the target variable measurements through a common subspace with the interesting effects.

assumed:

$$\Theta = \Psi \Gamma, \quad (2)$$

where the rank S_1 of parameters $\Psi_{P \times S_1}$ and $\Gamma_{S_1 \times K}$ is substantially lower than the number of target variables K and covariates P . The low-rank decomposition of the regression coefficient matrix (2) may be given an interpretation whereby the covariates X affect S_1 latent components with coefficients specified in Ψ , and the components, in turn, affect the target Y with coefficients Γ . Another line of work in multiple output prediction has focused on borrowing information from the correlation structure of the target variables when learning the regression model. The intuition stems from the observation that correlated targets are often seen to be affected similarly by the covariates, for example in genetic applications (see, e.g. Davis et al., 2014; Imoye et al., 2012). One popular method, GFlasso (Kim et al., 2009), learns the regression coefficients using

$$\hat{\Theta} = \arg \min_{\Theta} \sum_k (\mathbf{y}_k - X\theta_k)^T (\mathbf{y}_k - X\theta_k) + \lambda \sum_j \sum_k |\theta_{jk}| + \gamma \sum_{(m,j) \in E} r_{mj}^2 \sum_j |\theta_{jm} - \text{sign}(r_{mj})\theta_{jl}|, \quad (3)$$

where the θ_k are the columns of $\hat{\Theta}$. Two regularization parameters are introduced: λ represents the standard Lasso penalty, and γ encourages the effects θ_{jm} and θ_{jl} of the j th covariate on correlated outputs m and l to be similar. Here r_{mj} represents the correlation between the m th and l th phenotypes. The E is an *a priori* specified correlation graph for the output variables, with edges representing correlations to be accounted for in the model.

In this paper we propose a model that simultaneously learns the structured noise and encourages the sharing of information between the noise and the regression models. To motivate the new model, we note that by assuming independent prior distributions on Γ and Λ in model (1), one implicitly assumes independence of the interesting and uninteresting effects, caused by covariates X and unknown factors H , respectively (Fig. 1a). The assumption is appealing for example when explaining away batch effects (Fusi et al., 2012)

in high-dimensional data, but may be inadequate in the presence of other types of noise in molecular biology, where gene expression and metabolomics measurements record concentrations of compounds generated by ongoing latent biological processes. In this kind of situations, a limited set of covariates, such as single nucleotide polymorphisms (SNPs), determines the activity of the latent process only partially and all other activity of the process is due to unrecorded factors. In such cases, the noise affects the measurement levels through the very same process as the interesting signal (Fig. 1b), and rather than assuming independence of the effects, an assumption about parallel effects would be more appropriate. We refer to this type of noise as *latent noise* as it can be considered to affect the same latent subspace as the interesting effects. We note that in practice both types of structured noise are likely to be present. In this work, our main focus is on the latent noise, but we also present a comparison with a model that includes both types of structured noise simultaneously.

A natural way to encode the assumption of latent noise is to use the following model structure:

$$Y = (X\Psi + \Omega) \Gamma + E, \quad (4)$$

where the $\Omega_{N \times S_1}$ is a matrix consisting of unknown latent factors. In (4), Γ mediates the effects of both the interesting and uninteresting signals on the target variables. We note that the change required in the model structure is small, and has in fact been presented earlier (Bo and Sminchisescu 2009; recently extended with an Indian Buffet Process prior on the latent space by Baryg et al. 2014). We now proceed to using the structure (4) for GFlasso-type sharing of information (3) between the regression and noise models while simultaneously explaining away structured noise. To see that the information sharing between noise and regression models follows immediately from model (4), one can consider simulations generated from the model. The *a priori* independence assumption of model (1) results in uncorrelated regression weights regardless of the correlations between target variables (Figure 2a). The assumption of latent noise (4), however, encourages the regression weights to be correlated in a similar way as the target variables are (Figure 2c).

In this work, we focus on modelling weak signals in high-dimensional data with structured noise, where we consider effects that explain a tiny portion, say $< 1\%$, of the variance of the target variables as weak. We have hypothesized above that a model with the structure (4) might be particularly well-suited for this purpose. Additionally, (i) particular emphasis must be put on defining adequate prior distributions to distinguish the weak effects from noise as effectively as possible, and (ii) scalability to large sample size is needed in order to have any chance of learning the weak effects. For (i), we define *latent signal-to-noise ratio* β as a generalization of the standard signal-to-noise ratio in the latent space:

$$\beta = \frac{\text{Trace}(\text{Var}(X\Psi))}{\text{Trace}(\text{Var}(\Omega))}, \quad (5)$$

We use the latent signal-to-noise ratio as a hyperparameter in our model, and show that it is a key parameter affecting model performance. It can be either learned or set using prior knowledge. In addition, we introduce an ordered infinite-dimensional shrinkage prior that resolves the inherent rotational ambiguity in the model (4), by sorting both signal and noise components by their importance. Finally, we present efficient inference methods for the model.

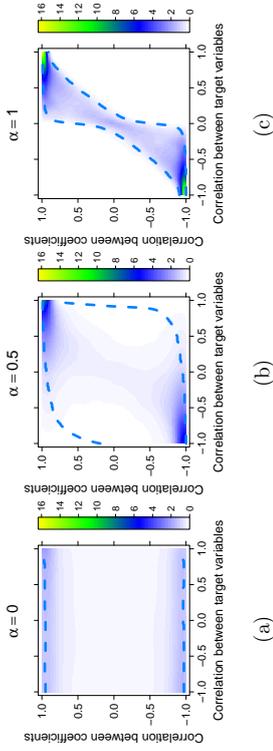


Figure 2: Conditional distribution of the correlation between regression coefficients, given the correlation between the corresponding target variables. In (a) the model (1) assumes *a priori* independent regression and noise models, and in (c) the model (4) makes the latent noise assumption. (b) A mixture of the models in a and c. The data were generated using equation (18), as described in Section 5.3, and α denotes the relative proportion of latent noise in data generation. The dashed lines denote the 95% confidence intervals of the conditional distributions.

2. Related work

Simultaneously solving multiple real-valued prediction tasks with the same set of covariates is called multiple-output regression (Breiman and Friedman, 1997); and more generally sharing of statistical strength between related tasks is called multitask learning (Baxter, 1996; Caruana, 1997). The data consist of N input-output pairs $(\mathbf{x}_n, \mathbf{y}_n)_{n=1, \dots, N}$; the P -dimensional input vectors \mathbf{x} (covariates) are used for predicting K -dimensional vectors \mathbf{y} of target variables. The common approach to dealing with structured noise due to unobserved confounders is to apply factor regression modeling (1) (West, 2003) and to explain away the structured noise using a noise model that is assumed to be *a priori* independent of the regression model (Stegle et al., 2010; Fusi et al., 2012; Rai et al., 2012; Virtanen et al., 2011; Klami et al., 2013; Rakitsch et al., 2013). A recent Bayesian reduced-rank regression (BRRR) model (Marttinen et al., 2014) implements the routine assumption of the independence of the regression and noise models; we will include it in the comparison studies of this paper.

Methods for multiple-output regression without the structured noise model have been proposed in other fields. In the application fields of genomic selection and multi-trait quantitative trait loci mapping, solutions (Yi and Banerjee, 2009; Xu et al., 2009; Calus and Veerkamp, 2011; Stephens, 2013) for low-dimensional target variable vectors ($K < 10$) have been proposed, but these methods do not scale up to the currently emerging needs of analyzing higher-dimensional target variable data. Additionally, sparse multiple-output regression models have been proposed for prediction of phenotypes from genomic data (Kim et al., 2009; Sohn and Kim, 2012).

Many methods for multi-task learning have been proposed in the field of kernel methods (Evgeniou and Pontil, 2007). These methods do not, however, scale up to data sets with

several thousands of samples, required for predicting the weak effects. Other relevant work include a recent method based on the BRRR presented by Foygel et al. (2012), but it does not scale to the dimensionalities of our experiments either. Methods for high-dimensional phenotypes have been proposed in the field of expression quantitative trait loci mapping (Botto et al., 2011) for the related task of finding associations (and avoiding false positives) rather than prediction, which is our main focus. Also functional assumptions (Wang et al., 2012) have been used to constrain related learning problems.

3. Model

In this Section, we present the details of our new model, Bayesian reduced rank regression with latent noise (latent-noise BRRR), show how the hyperparameters can be set using the latent signal-to-noise ratio, and analyze theoretically some properties of the infinite-dimensional shrinkage prior.

3.1 Model details: latent-noise BRRR

Our model is given by

$$Y = (X\Psi + \Omega)\Gamma + E, \quad (6)$$

where $Y_{N \times K}$ contains the K -dimensional response variables for N observations, and $X_{N \times P}$ contains the predictor variables. The product $\Theta = \Psi\Gamma$, of $\Psi_{P \times S_1}$ and $\Gamma_{S_1 \times K}$, results in a regression coefficient matrix with rank S_1 . The $\Omega_{N \times S_1}$ contains unknown latent factors representing the latent noise. Finally, $E_{N \times K} = [e_1, \dots, e_N]^T$, with $e_t \sim N(0, \Sigma)$, where $\Sigma = \text{diag}(\sigma_1^2, \dots, \sigma_K^2)$ is a matrix of uncorrelated target variable-specific noise vectors. Figure 3 displays graphically the structure of the model. In the figure, the node corresponding to the parameter Γ that is shared by the regression and noise models is highlighted with green.

Similarly to a recent BRRR model (Marttinen et al., 2014) and the Bayesian infinite sparse factor analysis model (Bhattacharya and Dunson, 2011), we assume the number of components S_1 connecting the covariates to the targets to be infinite. Accordingly, the number of rows in the weight matrix Γ , and the numbers of columns in Ψ and Ω , are infinite. The low-rank nature of the model is enforced by shrinking the columns of Ψ and rows of Γ and Ω increasingly with the growing column/row index, such that only a small number of columns/rows are influential in practice. The increasing shrinkage also solves any rotational unidentifiability issues by enforcing the model to mediate the strongest effects through the first columns/rows. In Section 3.4 we explore the basic properties of the infinite-dimensional prior, to ensure its soundness. The hierarchical priors for the projection weight matrix Γ , where $\Gamma = [\gamma_{lj}]$, are set as follows:

$$\begin{aligned} \gamma_{lj} | \phi_{lj}^{\Gamma}, \tau_h &\sim N\left(0, (\phi_{lj}^{\Gamma} \tau_h)^{-1}\right), \quad \phi_{lj}^{\Gamma} \sim \text{Ga}(\nu/2, \nu/2), \\ \tau_h &= \prod_{l=1}^h \delta_l, \quad \delta_1 \sim \text{Ga}(a_1, 1), \quad \delta_l \sim \text{Ga}(a_2, 1), \quad l \geq 2. \end{aligned} \quad (7)$$

Here τ_h is a global shrinkage parameter for the h th row of Γ and the ϕ_{lj}^{Γ} s are local shrinkage parameters for the individual elements of Γ , to provide additional flexibility over the global

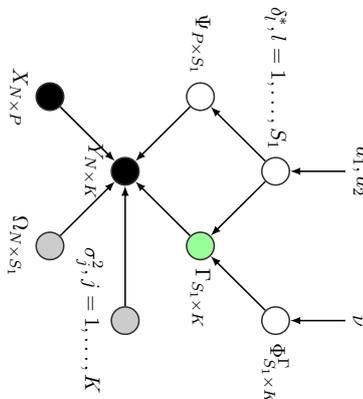


Figure 3: Graphical representation of latent-noise BRRR. The observed data are denoted by black circles, variables related to the reduced-rank regression part of the model by white circles, variables related only to the noise model are denoted by gray circles, and variables related to both the regression and the structured noise model are denoted with green circles. The matrix $\Phi_{S_1 \times K}^T$ comprises the sparsity parameters for the K target variables for the components.

shrinkage priors. The same parameters τ_h are used to shrink the columns of the matrices $\Psi = [\psi_{jh}]$ and $\Omega = [\omega_{jh}]$, because the scales of Γ and Ψ (or Ω) are not identifiable separately:

$$\psi_{jh}|\tau_h \sim N\left(0, (\tau_h)^{-1}\right), \quad \text{and} \quad \omega_{jh}|\tau_h \sim N\left(0, \sigma_\Omega^2 (\tau_h)^{-1}\right),$$

where σ_Ω^2 is a parameter that specifies the amount of latent noise, which is used to regularize the model (see the next Section). With the priors specified, the hidden factors Ω can be integrated out analytically, yielding

$$y_i \sim N\left((\Psi\Gamma)^T x_i, \sigma_\Omega^2 (\Gamma^*)^T (\Gamma^*) + \Sigma\right), \quad i = 1, \dots, N, \quad (8)$$

where Γ^* is obtained from Γ by multiplying the rows of Γ with the shrinkages $(\tau_h)^{-1/2}$ of the columns of Ω .

Finally, conjugate prior distributions

$$\sigma_j^{-2} \sim \text{Ga}(a_\sigma; b_\sigma), \quad j = 1, \dots, K, \quad (9)$$

are placed on the noise parameters of the target variables.

3.2 Regularization of latent-noise BRRR through the variance of Ω

The latent signal-to-noise ratio β in Equation (5) has an intuitive interpretation: given our prior distributions for Ψ and Ω , the prior latent SNR indicates the extent to which we believe

the noise to explain variation in Y , as compared to the variance explained by the covariates X . Thus, the latent SNR acts as a regularization parameter: when the latent variables Ω are allowed to have a large variance, the data will be explained by the noise model rather than the covariates. We note that this approach to regularization is non-standard and it may have favourable characteristics compared to the commonly used L1/L2 regularization of regression weights. First of all, the regression weights remain relatively unbiased as they need not be enforced to zero to control for overfitting. This is important when the effects are weak: if the effects were shrunk towards zero, they might be lost completely.

Secondly, while regularizing with the *a priori* selected latent SNR, the regularization parameter itself remains interpretable: every value of the variance parameter of Ω can be immediately interpreted as the percentage of variance explained by the noise model as compared to the covariates. In our experiments, we use cross-validation to select the variance of Ω and the interpretability of the parameter makes it easy to express beliefs of the plausible values based on prior knowledge. Making similar educated guesses for L1/L2 regularization parameters is not straightforward.

3.3 Difference between latent-noise BRRR and independent-noise BRRR

We call the standard Bayesian reduced rank regression (Equation 1), which assumes independent noise and signal models, the *independent-noise BRRR*. The new latent-noise BRRR differs from it in two ways: in the latent-noise BRRR

1. the structure of the model is different in that the noise model uses the same projection parameters as the regression model, and
2. the model is regularized by modifying the variance of the noise model. This is achieved by learning the latent signal-to-noise ratio parameter β .

In Section 5.5 we show that both of these improvements are needed to reach the performance differences observed.

We emphasize that although the technical difference between the two models is minor, the models are very different from the conceptual point of view, as discussed in the Introduction, as well as from the practical point of view. In particular, it has been reported before that with weak effects the independent-noise BRRR may suffer from severe instability, resulting from a highly multi-modal posterior distribution and, consequently, poor convergence and mixing properties of the learning algorithms (Koop et al., 2006; Marttinen et al., 2014). In Section 5.11, we demonstrate how the latent noise assumption provides just the required additional regularization to make the formal Bayesian inference tractable even with weak effects.

As both independent structured noise and latent noise could be present, a logical extension to the models presented so far is to consider both noise types simultaneously,

$$Y = (X\Psi + \Omega)\Gamma + HA + E, \quad (10)$$

where the distributional assumptions for Ψ , Ω and Γ are the same as in latent-noise BRRR, and for H and Λ they follow independent-noise BRRR. The Gibbs updates for this model are straightforward modifications of those for the latent-noise BRRR and independent-noise BRRR. We have implemented also this model and study its performance in Section 5.9.

We note that the latent-noise model is, in principle, able to express data generated by the independent-noise BRRR model, and vice versa. The latent-noise BRRR model may learn noise components that are independent from the signal in practice, having negligible contribution from the regression part $X\Psi$. On the other hand, nothing prevents the independent noise model to learn some correlated regression and noise components. Therefore, the family of models defined by Equation (10) that simultaneously includes both kinds of structured noise may have redundancy in its parameters. Indeed, the experiments in Section 5.9 demonstrate only minor improvements from this model.

3.4 Proofs of the soundness of the infinite prior

In this Section we verify the sensibility of the infinite non-parametric prior, which we introduce for ordering the components according to decreasing importance, and of a computational approximation resulting from truncation of the infinite model.

It has been proven that in Bayesian factor models $a_1 > 2$ and $a_2 > 3$ (in our case defined in eqn 7) is sufficient for the elements of $\Lambda\Lambda^T$ to have finite variance in a Bayesian factor model (1), even if an infinite number of columns with a prior similar to our model is assumed for Λ (Bhattacharya and Dunson, 2011). In this Section we present similar characteristics for the infinite reduced-rank regression model. The detailed proofs can be found in the Supplementary material. First, in analogy to the infinite Bayesian factor analysis model, we show that

$$a_1 > 2 \quad \text{and} \quad a_2 > 3 \quad (11)$$

is sufficient for the prediction of any of the response variables to have finite variance under the prior distribution (Proposition 1). Second, we show that the underestimation of uncertainty (variance) resulting from using a finite rank approximation to the infinite reduced-rank regression model decays exponentially with the rank of the approximation (Proposition 2). For notational clarity, let Ψ_h denote the h^{th} column of the Ψ matrix in the following. With this notation, the prediction for the i th response variable can be written as

$$\begin{aligned} \tilde{y}_i &= x^T \Theta_i \\ &= x^T \sum_{h=1}^{\infty} \Psi_h \gamma_{hi}. \end{aligned}$$

Furthermore, let $\Gamma(\cdot)$ denote below the gamma function (not to be confused with the matrix Γ used in all other Sections of this paper).

Proposition 1: Finite variance of predictions Suppose that $a_1 > 2$ and $a_2 > 3$. Then

$$\text{Var}(\tilde{y}_i) = \frac{\nu}{\nu-2} \sum_{j=1}^P \text{Var}(x_j) \frac{\Gamma(a_1-2)/\Gamma(a_1)}{1-\Gamma(a_2-2)/\Gamma(a_2)}. \quad (12)$$

A detailed proof is provided in the Supplementary material.

Proposition 2: Truncation error of the finite rank approximation Let $\tilde{y}_i^{S_1}$ denote the prediction for the i th target variable when using an approximation for Ψ and Γ consisting of the first S_1 columns or rows only, respectively. Then,

$$\frac{\text{Var}(\tilde{y}_i) - \text{Var}(\tilde{y}_i^{S_1})}{\text{Var}(\tilde{y}_i)} = \left[\frac{\Gamma(a_2-2)}{\Gamma(a_2)} \right]^{S_1},$$

that is, the reduction in the variance of the prediction resulting from using the approximation, relative to the infinite model, decays exponentially with the rank of the approximation. A detailed proof is provided in the Supplementary material.

4. Efficient computation by reparameterization

For estimating the parameters of the latent-noise BRRR, we use Gibbs sampling, updating the parameters one by one by sampling them from their conditional posterior probability distributions, given the current values of all other parameters. The bottleneck of the computation is in updating the matrix Ψ , and below we present a novel efficient update for this parameter.

4.1 Update of Γ

The conditional distribution of the parameter matrix Γ of latent-noise BRRR can be updated using a standard result for Bayesian linear models (Bishop et al., 2006) which states that if

$$\beta \sim N(0, \Sigma_\beta), \quad \text{and} \quad y|X^*, \beta \sim N(X^* \beta, \Sigma_y), \quad (13)$$

then

$$\beta|y, X^* \sim N(\Sigma_{\beta|y}(X^{*T} \Sigma_y^{-1} y), \Sigma_{\beta|y}), \quad (14)$$

where

$$\Sigma_{\beta|y} = (\Sigma_\beta^{-1} + X^{*T} \Sigma_y^{-1} X^*)^{-1}. \quad (15)$$

Because in our model (6) the columns E_i of the noise matrix are assumed independent with variances $\sigma_1^2, \dots, \sigma_K^2$, we get

$$Y_i \sim N((X\Psi + \Omega)\Gamma_i, \sigma_i^2 I_N). \quad (16)$$

Thus, by substituting

$$X^* \leftarrow X\Psi + \Omega, \quad \beta \leftarrow \Gamma_i, \quad \text{and} \quad \Sigma_y \leftarrow \sigma_i^2 I_N$$

into (13), together with prior covariance Σ_{β} derived from (7), we immediately obtain the posterior of Γ_i from (14) and (15).

4.2 Updates of Φ^1, δ, σ and Ω

The updates of the hyperparameters are the same as in Bayesian Reduced Rank Regression, and the conditional posterior distributions of the hyperparameters can be found in the Supplementary material of Marttinen et al. (2014). The Ω has the same conditional posterior distribution as the model parameter H of Marttinen et al. (2014).

4.3 Improved update of Ψ

The computational bottleneck of the naive Gibbs sampler is the update of parameter Ψ , which has PS_1 elements with a joint multivariate Gaussian distribution, conditionally on the other parameters (Geweke, 1996; Marttinen et al., 2014). Thus, the inversion of the

precision matrix of the joint distribution has a computational cost of $O(P^3 S_1^3)$. To remove the bottleneck, we reparameterize our model after which a linear algebra trick by Stegle et al. (2011) can be used to reduce the computational cost of the bottleneck to $O(P^3 + S_1^3)$. When sampling Ψ we also integrate over the distribution of Ω following the standard result from Equation (8). The reparameterization and the new posteriors are presented in the Supplementary material.

In brief, the trick is that the eigenvalue decomposition of a matrix of the form

$$C \otimes R + \sigma I \quad (17)$$

can be evaluated inexpensively. After reparameterizing the model in the proposed way the posterior covariance matrix of Ψ becomes of the form (17) and the eigenvalue decomposition can then be used to efficiently generate samples from the posterior distribution of Ψ . We note that the trick can also be applied to the original formulation of the Bayesian reduced-rank regression model by Geweke (1996) and the R-code published with this article allows generating samples from the original model as well. In the next Section, we compare the computational cost of the algorithm using the naive Gibbs sampler and the improved version that uses the new parameterization.

4.4 Sampling the maximum rank of the model

The sparse infinite factor analysis model presented by Bhattacharya and Dunson (2011) uses a certain adaptation procedure to update the maximum rank, i.e., the truncation point of their infinite-rank factor model. The idea is to update the maximum rank occasionally during the algorithm such that ranks having all elements of the corresponding projection vectors within some pre-specified distance from zero are removed from the model and, if none of the ranks has all elements within the threshold, another rank is added into the model. We have implemented a modification of this approach where we adapt the maximum rank of our infinite reduced rank regression model using a pre-specified cutoff for the amount of variance explained by the corresponding rank. With a slight abuse of terminology, we shall call this updating of the rank as sampling in the sequel.

5. Experiments

We start with a basic validation of the latent-noise BRRR model, and its relative merits over alternatives in a prediction task, using simulations with the ground truth available (Section 5.3), and a real-world omics dataset (Section 5.4). Section 5.5 analyses these results in more detail and identifies the characteristics of the proposed latent-noise BRRR model that are responsible for the performance differences observed, by considering the impact of each novel model aspect in isolation. Section (5.7) investigates another application domain, the detection of multivariate associations. In order to assess the prediction performance in more general, we analyse several additional real-world data sets from different domains in Section 5.8.

Different aspects of the inference algorithm are considered in three sub-sections: sampling vs. cross-validation of the rank and the latent signal-to-noise ratio (Section 5.6), speedup resulting from the proposed re-parameterization of the algorithm (Section 5.10),

and convergence diagnostics (Section 5.11). To assess the value of further extensions, Section 5.9 considers a model that includes both latent and independent structured noise simultaneously. Finally, Section 5.12 summarizes the findings on all real data sets.

5.1 Data sets

Experiments were performed on the following data sets:

NFBFC1966 [$N = 4702, P = 101, K = 96$, metabolomics prediction from SNPs] The NFBFC1966 data set comprises genome-wide SNP data along with metabolomics measurements for a cohort of 4,702 individuals (Rantakallio, 1969; Soininen et al., 2009). With these data, 96 metabolites belonging to the subclasses VLIDL, IDL, LDL and HDL (Houye et al., 2012) were used as the target variables and SNPs known to be associated with lipid metabolism (Teslovich et al., 2010; Ketunen et al., 2012; Global Lipids Genetics Consortium, 2013) were used as the covariates. Effects of age, sex, and lipid lowering medication were regressed out from the metabolomics data as a preprocessing step. For the genotype data, SNPs with low minor allele frequency (<0.01) were removed as a preprocessing step. For this data set, the comparison method GFLasso required excessive training time and we used 5-fold cross-validation to evaluate test set performances. Where cross-validation was needed for selecting model parameter values, the validation data performance was measured as an average over 3 validation sets, each comprising $\frac{1}{10}$ of the training samples.

DILGOM [$N = 509, P = 65, K = 18 \dots 137$, metabolomics and gene expression prediction from SNPs] The DILGOM data set (Houye et al., 2010) consists of genome-wide SNP data along with metabolomics and gene expression measurements. For details concerning metabolomics and gene expression data collection, see Soininen et al. (2009) and Ketunen et al. (2012). In total 509 individuals had all three measurement types. The DILGOM metabolomics data comprises 137 metabolites, most of which represent NMR-quantified levels of lipoproteins classified into 4 subclasses (VLIDL, IDL, LDL, HDL), together with quantified levels of amino acids, some serum extracts, and a set of quantities derived as ratios of the aforementioned metabolites. All 137 metabolites were used simultaneously as prediction targets. In gene expression prediction, in total 387 probes corresponding to curated gene sets of 8 KEGG lipid metabolism pathways were used as the prediction targets. A separate model was learnt for each pathway. The average number of probes in a pathway was 48. For details about the pathways, see the Supplementary material. On these data sets, 10-fold cross-validation was used to evaluate test set performances. To select values of the parameters that required evaluation on validation data, the training data was then further divided into 9 folds, on which cross-validation was performed to select parameters according to averaged validation set performance.

fMRI [$N = 1307, P = 776, K = 250$, fMRI response prediction from text stimuli] The cognitive neuroscience data set (Wehbe et al., 2014) consists of a time series of fMRI measurements from 8 subjects reading a chapter from ‘‘Harry Potter and the Sorcerers Stone’’ using *Rapid Serial Visual Presentation*: words of the text are presented one by one in the center of a screen. Brain voxel activations were measured

every 2 seconds. The 250 most accurately predictable voxels (see Supplementary material of Wehbe et al., 2014) of the fMRI measurements were used as prediction targets. The fMRI measurements from all patients were predicted simultaneously from features of the words being shown, such as semantic and syntactic properties, visual properties and discourse level features. The data were divided into 10 folds, only two of which were used to measure test data performance. This computational compromise was needed as the preprocessing (Wehbe et al., 2014) for each fold required about 10,000 hours of computation. To select the values of parameters that required evaluation on validation data, the training data were further divided into 10 folds, on which cross-validation was performed to select parameters according to averaged validation set performance.

econ [$N = 120, P = 52, K = 52$, macroeconomic time series prediction] The macroeconomic time series data set (Stock and Watson, 2006) consists of monthly values of 52 macroeconomic indicators. Prediction performance of these values from their earlier values was measured with different lags (1 month, 2 months, etc.). The data were processed as described by Carriero et al. (2011). Data for each month were used as a test set (395 test sets) while using data from the previous 10 years for training. Where cross-validation was needed for learning the values of model parameters, data from the last 2 years before the month-to-be-predicted were used for validation and data from the previous 8 years for training.

5.2 Methods included in comparison

We compared the latent-noise BRRR with a state-of-the-art sparse multiple-output regression method Graph-guided Fused Lasso (GFlasso) (Kim and Xing, 2009), BRRR/factor regression model (Marttinen et al., 2014) with and without the *a priori* independent noise model ('independent-noise BRRR', 'BRRR without noise model'), standard Bayesian linear model ('blm') (Gelman et al., 2004), standard ridge regression ('ridge regression') (Hoerl and Kennard, 1970), elastic-net-penalized multi-task learning (L2/L1 MTL), kernel regression with linear and Gaussian kernels combined with a process for removing confounding factors (Stegle et al., 2012) ('KRR with linear kernel + PEER', 'KRR with Gaussian kernel + PEER') and a baseline method of predicting with target data mean. GFlasso constitutes a suitable comparison as it encourages sharing of information between correlated responses, as our model, but does that within the Lasso-type penalized regression framework without the use of a noise model to explain away the structured noise. L2/L1 MTL is a multitask regression method implemented in the `glmnet` package (Friedman et al., 2010) that allows elastic net regularization. It does not use a noise model to explain away confounders either. The blm method and ridge regression were selected as a simple single-task baselines.

In one of the experiments, on an association study, latent-noise BRRR is compared with independent-noise BRRR and canonical correlation analysis ('cca'), considered the state-of-the-art methods for the detection of multivariate associations (Marttinen et al., 2013, 2014). Additionally, the simple univariate linear model ('lm') is included as it represents the common baseline in association analysis.

We compare latent-noise BRRR also with two other new models for structured noise modeling. In the simulations, we study the performance of correlated Bayesian reduced

rank regression ('correlated BRRR'), which is presented in more detail in the Supplementary material. In brief, in the correlated BRRR, the correlation structure of the target variables learnt by an *a priori* independent noise model is used as a prior for the regression weight parameters. With the NFBC1966 data and the macroeconomic time series data sets, we also study the performance of the method presented in Equation (10) in Section 3.3 that explicitly models both latent and independent structured noise, abbreviated as 'latent+independent-noise BRRR'.

Parameters for the different methods were specified as follows:

GFlasso: The regularization parameters of the `gw2` model were selected from the default grid using cross-validation. The method has been developed for genomic data indicating the default values should be appropriate. However, for NFBC1966 data, we were unable to run the method with the smallest values of the regularization parameters {110, 60, 10} due to lengthy runtime with these values. With this computational compromise of leaving out these three values, the average training time for the largest training data sets was ~ 650 h. With NFBC1966 data, the pre-specified correlation network required by the GFlasso was constructed to match the VLDL, IDL, LDL, and HDL metabolite clusters from Inouye et al. (2012). Within these clusters, the correlation network was fixed to the empirical correlations, and to 0 otherwise. With DILGOM data, we used the empirical correlation network, with correlations below 0.8 fixed to 0 to reduce the number of edges in the network for computational speedup.

independent-noise BRRR, BRRR without noise model: Hyperparameters α_1 and α_2 of all the BRRR models were fixed to 10 and 4, respectively. In total 1,000 MCMC samples were generated and 500 were discarded as burn-in. In preliminary tests similar results were obtained with 50,000 samples. The remaining samples, thinned by a factor of 10, were used for prediction. The maximum rank of the infinite-rank BRRR model was learned using cross-validation from the set of values {5, 10, 15} for the NFBC1966 data set, {2, 4, 8} for the metabolomics prediction task on the DILGOM data set and {2, 5, 10, 20} for the gene expression prediction task on the DILGOM data set. These grids were selected based on initial experiments. For the fMRI response prediction, the possible values for the maximal rank were limited to {2, 4} in order to save computational time. For the econometrics data set, maximum ranks of {5, 10, 20} were used. In the association detection task, the rank of independent noise BRRR was fixed to 1 as this was already sufficient for the task.

latent-noise BRRR: With the NFBC1966 data, the latent signal-to-noise ratio β was selected using cross-validation from a range of values from 100 to $\frac{1}{100}$, $\beta = \{100, 10, 2, 1, \frac{1}{7.5}, \frac{1}{15}, \frac{1}{30}, \frac{1}{60}, \frac{1}{100}\}$, in order to thoroughly evaluate the sensitivity of the model to this parameter. For the other data sets and tasks, the sets of values were as follows: DILGOM metabolomics prediction: $\beta = \{10, 2, 1, \frac{1}{7.5}, \frac{1}{15}, \frac{1}{30}, \frac{1}{60}, \frac{1}{100}\}$, DILGOM gene expression prediction $\beta = \{10, 1, \frac{1}{2}, \frac{1}{10}, \frac{1}{30}, \frac{1}{50}, \frac{1}{100}, \frac{1}{300}\}$ and for macroeconomic time series prediction $\beta = \{10, 2, 1, \frac{1}{7.5}, \frac{1}{15}, \frac{1}{30}, \frac{1}{60}, \frac{1}{100}\}$. For fMRI response prediction, the set of values was limited to $\beta = \{10, 1, \frac{1}{10}\}$ to save computation time. Other parameters, including the number of iterations, were set as for the independent-noise BRRR. The performance of the model was evaluated both by sampling the

maximum rank and by learning it with cross-validation from the same range of values as with independent-noise BRRR. Shrinkage hyperparameters were set to non-informative values, $a_1 = 10$ and $a_2 = 4$, similarly to the corresponding parameters a_3 and a_4 of independent-noise BRRR.

blm: The variance hyperparameter of BLM was integrated over using MCMC. The variance hyperparameter was assigned a Gamma prior with both shape and rate parameters set to 1. In total 1,000 posterior samples were generated and 500 were discarded as burn-in.

ridge regression: Ridge regression was used as implemented in the `glmnet` package with default parameters. The default convergence threshold parameters of `glmnet` were used and no warnings/numerical problems occurred.

L1/L2 MTL: The effects of different types of regularization penalties are an active research topic and we ran a continuum of mixtures of L1 and L2 penalties ranging from group lasso to ridge regression. The mixture parameter α controlling the balance between L1 and L2 regularization was evaluated on the grid $[0, 0.1, \dots, 0.9, 1.0]$ and selected using a 10-fold cross validation. The default convergence threshold parameters of `glmnet` were used and neither warnings nor numerical problems occurred.

KRR with linear kernel + PEER: First, the PEER software (Stegle et al., 2012) was used to remove the effects of confounders using 15 components. Then kernel ridge regression with a normalized linear kernel (Bishop et al., 2006) was applied using the residuals from PEER as the target variables. Kernel ridge regression was regularized according to the standard approach of adding parameter λ to the diagonal elements of the kernel. The value of λ was selected using cross-validation from a set of 10 values ranging from 0.1 to 100, $[10^{-1}, 10^{-0.66}, \dots, 10^{1.67}, 10^2]$. To share information between the different target variables, the approach of using the same kernel for all target variables was adopted.

KRR with Gaussian kernel + PEER: Kernel ridge regression using a Gaussian kernel was used. Regularization and the use of PEER were otherwise similar to KRR with linear kernel + PEER. The radius parameter of the Gaussian kernel was selected using cross-validation from a set of 30 values ranging from 0.001 to 1000, $[10^{-3}, 10^{-2.79}, \dots, 10^{2.79}, 10^3]$

cca: This is the conventional classical correlation analysis that attempts to identify linear combinations of the columns of the input and output matrices that are maximally correlated with each other.

correlated BRRR: Rank and hyperparameters a_1, a_2, a_3 and a_4 were set as with the independent-noise BRRR. This model is presented in detail in Supplementary Section 1.

latent+independent-noise BRRR: With the NFBG1966 data, the hyperparameters a_1, a_2, a_3 and a_4 were set as with the independent-noise BRRR. The latent signal-to-noise ratio β was selected using cross-validation from a range of values from 100 to $\frac{1}{100}$, $\beta = \{100, 10, 2, 1, \frac{1}{7.5}, \frac{1}{30}, \frac{1}{60}, \frac{1}{100}\}$ and the maximum rank was fixed to 10. For the econometrics data set, maximum ranks of $\{5, 10, 20\}$ were used and the signal-to-noise ratio β was selected using cross-validation from the values $\beta = \{10, 2, 1, \frac{1}{5}, \frac{1}{10}, \frac{1}{30}\}$. For both data sets, the variance parameter of the *a priori* independent noise H was selected from values $\{10^{-6}, 1\}$, value 10^{-6} corresponding to the extreme case of latent-noise BRRR.

5.3 Simulation experiment: impact of the noise model assumptions

In this Section, we study the implications of different noise model assumptions. Performances of models with different noise model assumptions are measured on simulated data sets generated from a continuum of models between the two extremes of assuming either latent noise, or *a priori* independent regression and noise models. The synthetic data are generated according to

$$Y = (X\Psi + \alpha\Omega)\Gamma + (1 - \alpha)HA + E, \quad (18)$$

where $\text{vec}(E) \sim \mathcal{N}(0, I_{NK})$ and the parameter $\alpha \in [0, 1]$ defines the proportion of variance attributed to the latent noise versus independent noise. We study a continuum of problems using the values of parameter $\alpha = 0, 0.1, \dots, 1$. The parameters Γ and Λ are orthogonalized using Gram-Schmidt orthogonalization. The parameters are scaled so that covariates X explain 3 % of the variance of Y through $X\Psi\Gamma$, the diagonal Gaussian noise $\mathcal{N}(0, I_{NK})$ explains 20 % of the total variance of Y and the structured noise $\alpha\Omega + (1 - \alpha)HA$ explains the remaining 77 % of the total variance of Y . The simulation was repeated 100 times and training data sets of 500 and 2000 samples were generated for each replicate. To compare the methods, performance in mean squared error (MSE) of the models learned with each method was compared to that of the true model on a test set of 15 000 samples. The number of covariates was fixed to 30 and the number of dependent variables to 60. Rank of the regression coefficient matrix and structured noise was set to 3 when simulating the data sets.

For independent-noise BRRR, the rank of the regression coefficient parameters Ψ and Γ was fixed to the true value while the rank of the noise model was learnt from the data. For latent-noise BRRR, the performance of the model was evaluated both by fixing the rank of the regression coefficient matrix to its true value and by learning it from the data. The variance of Ω was selected using 10-fold cross-validation. The grid for latent signal-to-noise ratios β was $\beta = \frac{1}{5}$ to $\frac{1}{15}$, $\beta = \{\frac{1}{5}, \frac{1}{7.5}, \frac{1}{10}, \frac{1}{12.5}, \frac{1}{15}\}$. More specifically, $\text{Var}(\text{vec}(\Omega)) = \sigma_{\Omega}^2 I_{NK}$ where $\sigma_{\Omega}^2 = \frac{1}{2} \times \text{Trace}(\text{Var}(X))$. The grid was chosen according to the interpretation given in Section 3.2, it corresponds to assuming that the latent noise explains 5 to 15 times the variance explained by the covariates.

Figures 4 (a) and (b) present the results of a simulation study with training sets of 500 and 2000 samples, respectively. When the structured noise is generated according to the conventional assumption of independent signal and noise, the model making the independence assumption (i.e., the independent-noise BRRR) performs equally well to the true

model with both 500 and 2000 samples. However, when the assumption is violated and the proportion of latent noise increases, the performance of the independent-noise BRRR breaks down, whereas the latent-noise BRRR performs consistently well. The method that does not explain away the structured noise at all (BRRR without noise model) is always inferior to the null model with the training set of 500 samples. When the number of training samples is increased to 2000 and the noise is generated according to the latent-noise assumption, the model, however, outperforms even the independent-noise BRRR. Thus, having no noise model is in this case better than having the noise model based on the incorrect independence assumption, which emphasizes the importance of the assumptions on which the noise model is based. Interestingly, with $n=2000$ the BRRR without noise model is among the best performing methods whereas with $n=500$ it is clearly the worst, highlighting the fact that the smaller n gets, the more important the right assumptions become.

The latent-noise end of the continuum appears to be more difficult for the methods that do not account for the structured noise (blm, BRRR without noise model). This weak but consistent trend can be seen in Figure 4(a,b) where the difference between the oracle and these methods increases with the percentage of latent noise. This behaviour is, however, rather intuitive in terms of Equation (18); by rewriting

$$\begin{aligned} Y &= (X\Psi + \alpha\Omega)\Gamma + (1 - \alpha)HA + E, \\ &= X\Psi\Gamma + \alpha\Omega\Gamma + (1 - \alpha)HA + E \end{aligned}$$

it is obvious that as $\alpha \rightarrow 1$, the structured noise (coming from Ω and H) will with certainty be projected on the particular target variables that are affected by the covariates X . In other words, latent noise blurs exactly the relationships of interest, being very disruptive.

Figure 4 shows results also for an alternative novel model that shares information between the noise and regression models (correlated BRRR, see Supplementary material for a detailed description). The model includes a separate noise model for the structured noise, as in (1), but achieves the information sharing by assuming a joint prior for the noise and regression models. In detail, conditional on the noise model, the current residual correlation matrix between the response variables is used as a prior for the rows of Γ . This way the correlations between target variables are propagated into the corresponding regression weights; however, the strongest noise components are not automatically coupled with the strongest signal components. Notably, the performance of the correlated BRRR model is very similar to the regular BRRR model that does not have any dependence between the noise and signal components.

5.4 NFBC1966: metabolomics prediction

In this Section, the models accounting for latent noise are evaluated in terms of predictive performance on the NFBC1966 data with different training set sizes. Figure 5 presents the test data MSE for the different methods. With the larger training set sizes, latent-noise BRRR outperforms the other methods. With the smallest training data size, ridge regression and latent-noise BRRR perform equally outperforming all other methods. However, ridge regression is unable to improve its performance as the number of training data points increases, and with the larger training sets it is outperformed by the more complex methods.

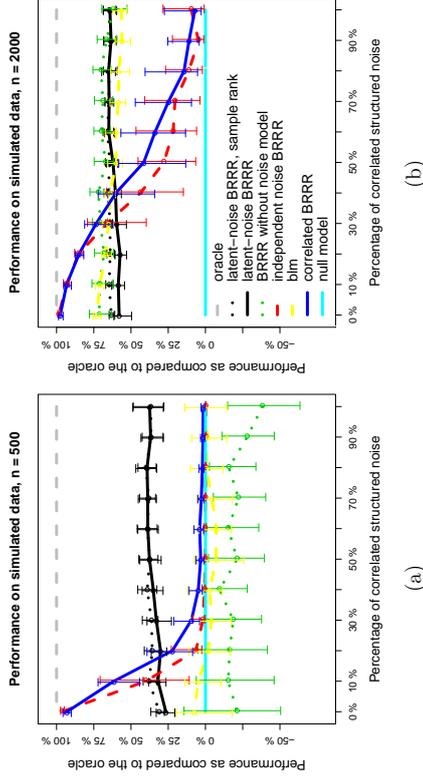


Figure 4: Performance of different methods, compared to the true model, as a function of the proportion of latent noise with a training set of (a) 500 and (b) 2000 samples. The x-axis indicates the proportion of noise generated according to the latent noise assumptions (100% corresponds to $\alpha = 1$). Bars denote ± 1 standard deviation, computed independently for each x-coordinate. The performance of 100% means the amount of variance explained by the model is equal to the amount explained by the true model. The performance of 0% means that the method does not explain any variance of the target variables, whereas negative values indicate the variance actually increases after taking the predictions into account.

Method blm performs worse than the baseline (null model, prediction with training set mean), even with the largest training data set containing 3761 individuals, and BRRR without noise model requires the largest training set size in order to outperform the baseline. A paired t-test for the performance difference between latent-noise BRRR and independent-noise BRRR yields a p-value of 0.03 suggesting a statistically significant difference.

5.5 Differences between latent-noise BRRR and independent-noise BRRR on NFBC1966 metabolomics prediction

The two differences between our new approach, latent-noise BRRR, and independent-noise BRRR are (1) model structure (latent-noise BRRR shares parameters between the regression and noise models) and (2) using the latent signal-to-noise ratio parameter β to regularize the model. In order to identify how these developments lead to the observed performance differences on the NFBC1966 data, we performed a sensitivity analysis for the two methods with respect to the assumed amount of variance attributed to the noise model.

Figure 6 presents the results of this sensitivity analysis. For latent-noise BRRR, the assumed variance of the noise model controlled by the *a priori* signal-to-noise ratio β affects

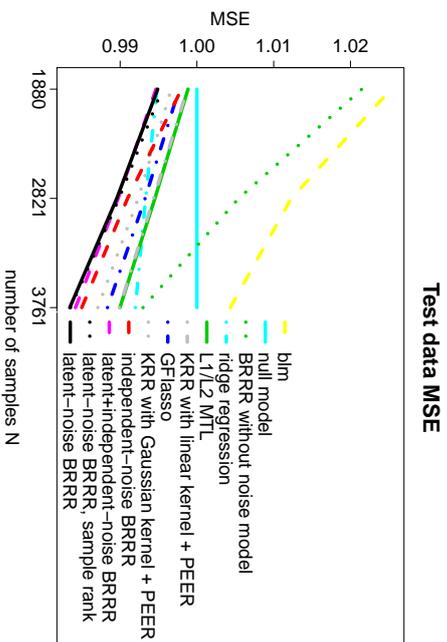


Figure 5: Test data MSE for different amounts of training data on the NFBG1966 metabolomics data. The MSEs have been scaled to give the null model a MSE of 1.

performance in a consistent way, whereas for independent-noise BRRR the impact appears random. If the performance difference stemmed mainly from controlling the variance of the noise model, controlling that parameter for both models should lead to similar results. On the other hand, if the difference in the model structure alone sufficed to explain the performance difference, the difference should not be sensitive to the variance of the noise model. Hence, we conclude that, on this data set, both the new model structure and regularization by using the latent signal-to-noise ratio are required for improved performance.

We also studied the variability of the estimated latent SNR on different folds. The optimal L-SNR was estimated very consistently; the results are presented in Supplementary Figure 3.

5.6 Evaluation of the chosen inference procedures for rank and noise parameters

Inference for the proposed model could naturally be done in several alternative ways. In this Section we justify the proposed inference procedure.

In the simulations (Section 5.3), sampling the maximum rank of the infinite prior worked well, measured in terms of predictive performance. Figure 4 shows that sampling the maximum rank actually improves performance, as compared to fixing it to the value used in the generative process, when the latent noise assumption is wrong (left end), both when $N = 500$ and when $N = 2000$. When the latent noise assumption holds (right end), the two

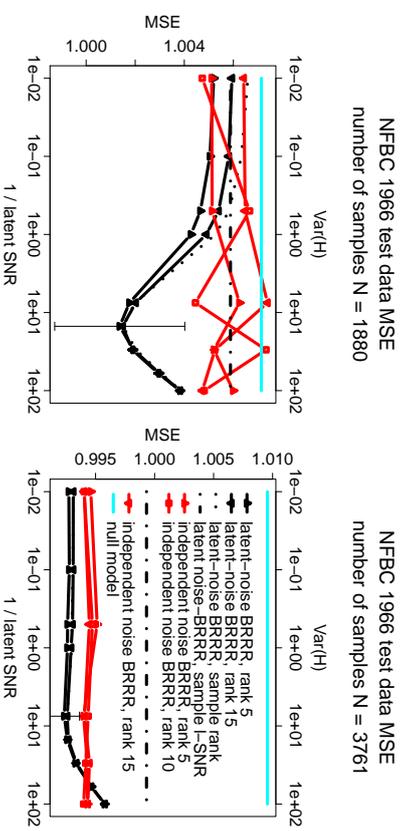


Figure 6: Sensitivity of latent-noise BRRR and independent-noise BRRR to the variance of the structured noise with different maximum ranks. The results are on NFBG1966 test data MSE ($N = 1880$ and $N = 3761$) as a function of the noise model variance. Lower axis: $a \text{ priori}$ latent signal-to-noise ratio of latent-noise BRRR. The upper axis: variance of the model parameter H of independent-noise BRRR. The bar denotes the standard deviation of the test set performance difference observed between the two models in cross-validation. The figures also present the unscaled performance of the null model and the performance of the latent-noise BRRR when using sampling to infer the latent signal-to-noise ratio (latent-noise BRRR, sample L-SNR) and when using sampling to infer the maximum rank (latent-noise BRRR, sample rank). When $N = 3761$, sampling the rank results in similar performance as obtained with the fixed values and thus the curves overlap.

inference procedures perform equally well. With the NFBG1966 data set (Figure 5), learning the maximum rank of the infinite prior by sampling (latent-noise BRRR, sample rank) or by cross-validation (latent-noise BRRR) results in very similar test set performances, similarly to in the simulation experiment in Section 5.3. Hence, we conclude that for learning the maximum rank, both sampling and cross-validation are appropriate techniques. We also ran the independent-noise BRRR so that the rank was sampled instead of selecting it using cross-validation on this data. However, the results were poor, with the test data MSE equal to 1.019 with $N = 1880$ and 1.005 with $N = 3761$. The lines were omitted for clarity. We hypothesize that the problems with the instability of the model (see Section 5.11) were accentuated when the rank was sampled.

The key parameter of our model, the latent signal-to-noise ratio, was estimated using cross-validation. In the simulations, the cross-validation based scheme allowed estimation of the latent signal-to-noise ratio to a reasonable accuracy. The estimated values are included in Figure 2 in the Supplementary material. While the latent signal-to-noise ratio of the

generative process was $\approx 1/25$, the estimated posterior latent signal-to-noise ratios ranged from $\frac{1}{14}$ to $\frac{1}{19}$ in the parts of the domain where the percentage of correlated structured noise was 100-80%. When the percentage of correlated structured noise was 0-10 %, the model correctly learnt lower variance for the latent noise and a corresponding stronger latent signal-to-noise ratio β .

We also studied the performance of latent-noise BRRR while sampling the variance of the noise model. A non-informative prior was assigned for the variance of Ω , $\Omega \sim \mathcal{N}(0, \sigma_\Omega^2)$ and $\sigma_\Omega^2 \sim \text{Gamma}(\text{shape} = 0.001, \text{rate} = 0.001)$. The performance of this model is presented in Figure 6. The performance of latent-noise BRRR when sampling the variance of Ω is consistently worse than when using cross-validation to select the value of the latent signal-to-noise ratio. Hence, we conclude that, as opposed to other parameters, cross-validation is needed to learn the latent-signal to ratio to reach the improved performance.

5.7 NFBC1966: multivariate association detection

Detection of associations between multiple SNPs and metabolites is a topic that has received attention recently (see, e.g., Kim et al., 2009; Inouye et al., 2012; Marttinen et al., 2014). Here we demonstrate the potential of the new method in this task using two illustrative example genes for which ground truth is available. Associations between SNPs within two genes, *LIPC* and *XRCC4*, and the metabolites in the NFBC1966 data are investigated in the experiment. Note that the covariates (SNPs) used in this experiment are different from the ones used in the prediction experiment: here SNPs in individual genes are used, whereas in the prediction experiment all known lipid-associated SNPs were used. *LIPC* was selected as a reference, because it is one of the most strongly lipid-associated genes. On the contrary, *XRCC4* was discovered only recently using three cohorts of individuals (Marttinen et al., 2014), and it was selected to serve as an example of a complex association detectable only by associating multiple SNPs with multiple metabolites, and not visible using simpler methods.

We use the proportion of total variance explained (PTVE) as the test score (Marttinen et al., 2014), and sample 100 permutations to measure the power to detect the associations. Furthermore, we use downsampling to evaluate the impact of the amount of training data. For comparison, we select the BRRR, the exhaustive pairwise (univariate) linear regression ('lm'), and canonical correlation analysis (CCA) (Ferreira and Purcell, 2009), these being the methods that have been proposed for the task and having a sensible runtime in putative genome-wide applications. For lm, the minimum p-value of the regression coefficient over all SNP-metabolite pairs, and for the CCA, the minimum p-value over all SNPs (each SNP associated with all metabolites jointly) are used as the test scores. The association involving the *XRCC4* gene was originally detected using the BRRR model; however, unlike here, informative priors were used for the regression coefficients.

Table 1 presents the ranking of the original data among the permuted data with different sample sizes and methods. Ten MCMC chains were computed for both models to account for sampling variability on this difficult and relatively strongly collinear data. The association score was obtained by averaging over the scores for different chains. As expected, all methods were able to detect the association involving *LIPC* with both training set sizes. However, latent-noise BRRR had the highest power to detect the *XRCC4* gene.

Table 1: Power of different methods to detect the association between metabolomics profiles and *XRCC4* or *LIPC* genes with $N = 4702$ and $N = 2351$ samples. Power is measured as the proportion of association test scores in permuted data sets smaller than the test score in the original data set. Value 1 indicates that the association score of the unpermuted data was higher than the score in any permutation.

	<i>XRCC4</i>		<i>LIPC</i>	
	$N = 4702$	$N = 2351$	$N = 4702$	$N = 2351$
latent-noise BRRR	0.98	0.94	1	1.00
independent-noise BRRR	0.41	0.32	1	0.99
lm	0.62	0.74	1	1.00
cca	0.20	0.24	1	1.00

5.8 Results: other real-world data sets

To thoroughly study the empirical value of the new method, we compared it to alternative methods on macroeconomic time series prediction, metabolomics and gene expression prediction experiments on the DILGOM data set and the fMRI response prediction. In these domains, explaining away structured noise is of crucial importance.

With the DILGOM data, the prediction of the weak effects was challenging for all methods. Indeed, we noticed that the null model using the average training data value for prediction was better than any other method in terms of MSE over all target variables with the single exception of L1/L2 MTL, which set all regression coefficients to zero thus reducing to the null model. However, a detailed investigation of the results revealed that while many of the target variables could not be predicted at all (as indicated by the worse than null model MSE) some of the target variables could still be predicted better than the null model, and by focusing the analysis on the MSE computed over the predictable target variables (*i.e.*, those that could be predicted better than the null by at least one method), comparisons regarding the model performances could still be made. For consistency, both metrics were computed also with the fMRI and econometrics data sets. To save computation time, we chose to evaluate only the cross-validation based variant of our model for the fMRI data, as this approach had already been identified as the most promising implementation of our method.

Table 2 and supplementary Table 1 present the results of the macroeconomic time series prediction experiment, metabolomics and gene expression prediction experiments on the DILGOM data set and the fMRI response prediction experiments. The results have been normalized so that the score for the null model (prediction using the mean) is 1. Table 2 presents the results for the predictable target variables and supplementary Table 1 presents the results obtained by averaging test data MSE over all target variables.

Latent-noise BRRR outperforms independent-noise BRRR consistently on the gene expression (on 8/10 folds), metabolomics (10/10 folds) and fMRI response prediction (2/2 folds) tasks on both scores. In the fMRI response prediction, the latent-noise BRRR and L1/L2 MTL are the only methods that outperform the null model. With the DILGOM data none of the methods outperformed the null model when averaged over all target variables and when concentrating on the predictable target variables, only the latent-noise BRRR and

KRR with Gaussian kernel were able to outperform the null model. With gene expression prediction, latent noise BRRR (sample rank), GFlasso and the kernel methods outperform the null model, latent-noise BRRR being the best. The economics data is the only case in which the independent-noise BRRR is more accurate than the latent-noise BRRR on both metrics, and the latent-noise BRRR is the third best method. In this data set, however, the effects appear rather strong as different methods explain up to 10-32% of the variance of the target variables and the best method is, in fact, ridge regression.

On the small DLGOM data sets, L1/L2 MTL sets all regression weights to zero as hypothesized in Section 3.2. This demonstrates the need to develop new alternatives to L1/L2 regularization: when modeling weak effects on small data sets, using L1/L2 penalties can prevent analysis altogether. Ridge regression appears to suffer from the same problem on the NFBC data set: although shrinking weights towards zero efficiently avoids overfitting, the model is only able to learn the strongest effects. Thus ridge regression outperforms most methods on the smallest training set (where the complex methods easily overfit), but heavily loses as the training set increases and the more complex methods become able to also benefit from the weak effects. Regularization by making the noise model stronger as in latent-noise BRRR avoids this problem.

The standard method blm performs surprisingly poorly especially as compared to ridge regression. The implementation was checked carefully. Predictive performance with the standard least squares linear model was also evaluated for some of the data sets (results not shown) and we found that it performed even worse than the blm. We hypothesize that the collinearity present in all of the data sets analyzed harmed the performance of blm and lm more than that of ridge regression.

5.9 Results: simultaneous modeling of both latent and independent structured noise

As both latent and independent structured noise can be present simultaneously, we evaluated the possible gains from taking both noise types simultaneously into account. A model that incorporates both latent and independent structured noise, here called latent+independent-noise BRRR, was evaluated for the metabolomics prediction task on the NFBC1966 data and on the macroeconomic time series prediction task, the strong domains of the methods of interest.

Results of this experiment are presented in Table 3. In metabolomics prediction, accounting for both noise types improved results slightly on the smallest training data size as compared to the best performing method latent-noise BRRR. On the larger training data sets, the more flexible latent+independent-noise BRRR model performed worse than the latent-noise BRRR that only accounts for latent noise. On the macroeconomic time series prediction task, accounting for both noise types improved performance as compared to only accounting for the dominant noise type (independent structured noise) on the smaller training data set. For summary, even though slight performance improvements were seen with the smallest training set sizes, the results indicate that as the size of the training data set increases, the advantages disappear. We hypothesize that the potential under-identifiability issues discussed in Section 3.3 hinder model performance more than the increased flexibility improves it.

	economics	DLGOM: gene expression	DLGOM: metabolomics	FMRI
latent-noise BRRR	0.73320±0.22564	0.99990±0.00057	1.00046±0.00130	0.99798±0.00282
latent-noise BRRR, sample rank	0.73453±0.21219	1.00039±0.00107	0.99995±0.00100	
independent-noise BRRR	0.71072±0.20549	1.00051±0.00038	1.04163±0.03781	1.00215±0.00183
L1/L2 MTL	0.75035±0.15651	1.00000±0.00000	1.00000±0.00000	0.99786±0.00090
GFlasso		1.00010±0.00106	0.99996±0.00221	
KRR with linear kernel + PEPER	0.88138±0.11021	1.00093±0.00057	0.99995±0.00006	1.00236±0.00112
KRR with Gaussian kernel + PEPER	0.90497±0.09707	0.99985±0.00016	0.99998±0.00004	1.00649±0.00179
BRRR without noise model	0.81818±0.34747	1.00568±0.00274	1.30795±0.08802	1.06722±0.05586
ridge regression	0.689771±0.202603	1.001798±0.001090	1.000445±0.003089	1.003388±0.007420
blm	1.59040±1.23041	1.04245±0.00914	1.52573±0.08859	2.08650±0.25396
null model	1.00000±0.00000	1.00000±0.00000	1.00000±0.00000	1.00000±0.00000

Table 2: Test data MSE computed on the predictable target variables on the economics, DLGOM and FMRI data sets. Bold font indicates better than baseline accuracy achieved by predicting with the training data mean.

5.10 Improvement in computational efficiency resulting from the reparameterization of model

To confirm the computational speed-up resulting from the reparameterization presented in Section 4, we performed an experiment where the algorithm implementing the naive

	NFBC $N = 3761$	NFBC $N = 1880$	econometrics $N = 120$	econometrics $N = 60$
latent-noise BRRR	0.9833±0.0077	0.9949±0.0037	0.7536±0.2143	0.8374±0.1816
latent+independent- noise BRRR	0.9840±0.0072	0.9947±0.0019	0.7445±0.1918	0.7889±0.1561
independent- noise BRRR	0.9849±0.0078	0.9980±0.0059	0.7339±0.1977	0.8097±0.2085

Table 3: Performance of the most flexible modeling assumptions. Test data MSE on the NFBC and econometrics data sets. On the larger training data sets, latent-noise BRRR and independent noise BRRR outperform the model that accounts for both noise types, latent+independent-noise BRRR. On the smaller training data sets, however, this model outperforms the models that only account for one noise type.

Gibbs sampling updates for the Bayesian reduced-rank regression (Geweke, 1996; Karlsson, 2012) was compared with the new algorithm that uses the reparameterization. Similar improvements were achieved with all other BRRR models as well.

Ten simulated data replicates were generated from the prior. The number of samples in the training set was fixed to 5000 and the number of target variables was set to 12. Rank of the regression coefficient matrix was 2. Runtime was measured as a function of the number of covariates, which was varied from 100 to 300; 1000 posterior samples were generated. The new algorithm that reparameterizes the model clearly outperformed the naïve Gibbs sampler (Figure 7). As a sanity check, the regression coefficient matrices estimated by the algorithms were compared, and found to be similar.

5.11 Efficiency of the algorithm

To investigate the efficiency of the proposed algorithm and to compare it with the alternative methods, we recorded the wall-clock run times with the NFBC1966 data set, shown in Figure 8. In addition, we studied the conventional convergence diagnostics. To assess convergence and mixing, we re-computed four MCMC chains of 2000 posterior samples each, for each of the BRRR methods. Averaged effective sample sizes (ESS) and potential scale reduction factors (PSRF) were computed for 200 randomly selected parameters of the regression coefficient matrix (Gelman et al., 2004). These results are presented in Table 4.

All BRRR methods, except for independent-noise BRRR, converge (PSRF < 1.1) and mix acceptably efficiently ($\frac{N_{\text{effective}}}{N_{\text{samples}}} \approx \frac{40}{1000}$). Independent noise BRRR, however, showed poor mixing and convergence. In initial experiments we observed that the PSRF for the independent-noise BRRR did not necessarily ever reach values indicating convergence even when sampled for 15,000 iterations. Thus, we decided to simply use the same number of MCMC iterations for each method in our experiments. The reason for the bad behaviour was the multimodality of the posterior distribution, caused by the too flexible model structure of the independent noise model, and the resulting convergence of the different chains into different modes.

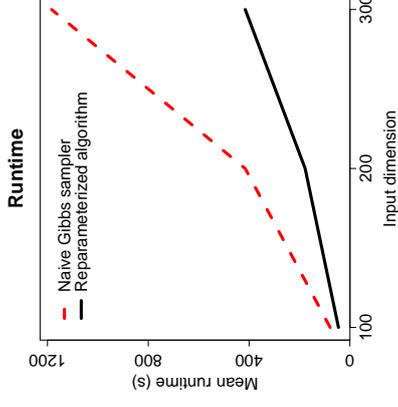


Figure 7: Runtime of the algorithm implementing the naïve Gibbs sampler with computational complexity and the new algorithm that reparameterizes the model. The naïve algorithm has a computational complexity of $O(P^3S_1^3)$ and the new algorithm $O(P^3 + S_1^3)$. Random variation over the repetitions was minimal and the error bars were omitted for clarity.

	independent-noise BRRR	BRRR without noise model	latent-noise BRRR	latent-noise BRRR, sample rank
1000 samples	4.46 ± 0.32	1.03 ± 0.03	1.06 ± 0.05	1.01 ± 0.004
2000 samples	3.42 ± 0.18	1.02 ± 0.02	1.05 ± 0.05	1.01 ± 0.003

Table 4: Averaged PSRF.

To further demonstrate the difference between the latent-noise and independent-noise BRRR methods, we visualized the MCMC trace of the association metric used in Section 5.7. The instability of independent-noise BRRR is strikingly visible in Figure 9. The chains converge to different modes and mix very slowly. On the other hand, the latent-noise BRRR appears to mix adequately and always converges to the same mode, except for one of the ten chains with the XRCC4 gene, which converges to a mode with a lower value of the explained variance.

5.12 Results: summary of the results with the real data sets

To provide an overview of the performances of the different methods on the various data sets and tasks, the methods' performances were ranked for each task/data set. For the prediction tasks, methods were ranked according to the MSE on the test set. When none of the methods outperformed the null model, the scores on the predictable target variables

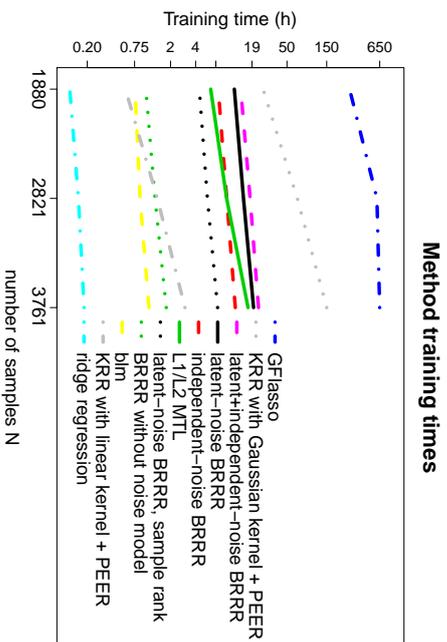


Figure 8: Computation times of the methods for different training set sizes N on the NFBG1966 metabolomics data.

	independent-noise BRRR	BRRR without noise model	latent-noise BRRR	latent-noise BRRR, sample rank
1000 samples	4.32 ± 0.32	43.88 ± 0.93	44.83 ± 0.25	40.74 ± 1.18
2000 samples	5.15 ± 0.66	84.39 ± 1.61	86.40 ± 0.43	77.77 ± 1.25

Table 5: Effective sample sizes for the Bayesian reduced rank regression methods. Independent-noise BRRR mixes substantially worse than the other methods.

were compared instead. In the association detection task, estimated statistical power was used as the ranking criterion. Table 6 presents the overview results.

Averaged over all data sets and tasks, latent-noise BRRR outperforms the comparison methods. In particular, the latent-noise BRRR outperforms the independent-noise BRRR on all setups except for the macroeconomic time series prediction task, where independent-noise BRRR is the best method and the two variants of latent-noise BRRR follow. The difference between the latent-noise BRRR and the independent-noise BRRR is consistent, present on 4/5 test folds on the NFBG1966 metabolite prediction, 8/10 test folds on the DLGOM gene expression prediction, 10/10 test folds on DLGOM metabolite prediction and on 2/2 folds on the FMRI response prediction. On macroeconomic time series prediction, independent-noise BRRR is better on 218/395 test folds. In the association detection task the latent-noise BRRR has higher power with both training set sizes on the challenging XRCC4 gene (0.94 vs. 0.32 with $n=2,351$; 0.98 vs. 0.41 with $n=4,702$).

Simultaneously accounting for both latent and independent structured noise improves performance on the smallest training data sets considered in the macroeconomic time series prediction and metabolomics prediction (NFBG1966) as compared to accounting for only one type of noise. On the other hand, with the larger training set sizes, the models with just the dominant noise type present perform better than the model including both noise types simultaneously.

Selecting the rank for latent-noise BRRR by sampling or by cross-validation results in comparable performance. Average performance ranks for cross-validation based and sampling-based inferences are 2.5 and 3.5, respectively. For the NFBG1966 data set and gene expression prediction task on the DLGOM data set, cross-validation yields better performance. On metabolomics prediction on DLGOM and the macroeconomic time series prediction, on the other hand, the sampling-based approach works better. It is also intriguing that similarly to the simulations, the sampling based variant of the model works better with independent structured noise (macroeconomic time series prediction) than the cross-validation based approach.

Latent-noise BRRR outperforms the null model on all test cases except for the metabolomics prediction on the DLGOM data. Even on that data set, however, the variant of the model that samples the maximum rank of the infinite prior outperforms the null model. We hypothesize that the poor performance may have resulted from convergence to some inferior mode of the posterior distribution; this can happen to latent-noise BRRR (as demonstrated in Figure 9) although the sharing of information between the signal and noise models makes it substantially more stable than the independent-noise BRRR.

6. Discussion

In this work, we evaluated the performance of multiple-output regression with different assumptions for the structured noise. While most existing methods assume *a priori* independence of the interesting effects and the uninteresting structured noise, we started from the opposite assumption of strong dependence between the components of the model. This assumption may be deemed appropriate for instance with the molecular biological data sets often analyzed with such methods. Using simulations we demonstrated the harmfulness of the independence assumption when latent noise was present. In real data experiments the model assuming latent noise outperformed state-of-the-art methods in prediction of metabolite measurements from genotype (SNP) data and FMRI response prediction, and showed consistently good performance in the different domains. In an illustrative multivariate association detection task, the latent noise model had increased power to detect associations invisible to other methods. To better address the computational needs, we presented a new algorithm reducing the runtime considerably, and improving the scalability of the BRRR models as the number of variables increases. The prior distributions were parameterized in terms of the new concept of *latent signal-to-noise ratio*, which was a key ingredient for optimal model performance. In addition, the rotational unidentifiability of the model was solved using ordered infinite-dimensional shrinkage priors. We also demonstrated that the two modifications (model structure, regularization through the latent signal-to-noise ratio) made to the existing state-of-the-art noise modeling approach were both needed in order to reach the optimal performance.

	NFBC $N = 3761$	econometrics $N = 120$	DILGOM: gene expression $N = 458$	DILGOM: metabolomics $N = 458$	fMRI $N = 1307$	NFBC: XRCC4 association detection $N = 4702$	Average rank
latent-noise BRRR	1	3	2	8	2	1	2.8
latent-noise BRRR, sample rank independent noise	2	4	6	1			3.2
BRRR	3	2	7	9	4	3	4.7
L1/L2 MTL	7	5	4	5	1		4.4
GFlasso	4		5	3			4.0
KRR with linear kernel + PEER	6	7	8	2	5		5.6
KRR with Gaussian kernel + PEER	5	8	1	4	7		5.0
BRRR without noise model	9	6	10	10	8		8.6
blm	11	10	11	11	9		10.4
null model	10	9	3	6	3		6.2
ridge regression	8	1	9	7	6		6.2
cca						4	
lm						2	

Table 6: Summary: ranking of methods according to performance in each studied data set and task.

In real data both latent and independent structured noise can be present. We studied a model incorporating both types simultaneously, and, based on these results, we concluded that the possible gains in predictive power as compared to modeling only the dominant type of noise were not worthwhile. In fact, results were also found to degrade when both noise types were included, which we hypothesize to be the result of poor identifiability of the corresponding model.

The new model implementing the concept of latent noise was studied using high-dimensional data containing weak signal (weak effects). The new model exploits a ubiquitous character-

istic of such data: while the interesting effects are weak, the noise is strong. Latent-noise BRRR borrows statistical strength from the noise model so as to alleviate learning of the weak effects, by automatically enforcing the regression coefficients on correlated target variables to be correlated. This intuitive characteristic can be seen as a counterpart of the powered correlation priors (Krishna et al., 2009) in the target variable space: Krishna et al. used the correlation structure of the covariates as a prior for the regression weights to enforce correlated covariates to have correlated weights.

The latent-noise BRRR is an extension of several common model families. By removing the covariates, the model reduces to a standard factor analysis model, which explains the output data with underlying factors. Thus, the latent-noise BRRR can be seen as a reversed analogy of PCA regression (West, 2003), in which components of the input space are used as covariates in prediction; in latent-noise BRRR components derived from the output space are predicted using the covariates (see Bo and Sminchisescu, 2009). Allowing the noise term to affect the latent space directly results in interesting connections to *linear mixed models* (LMMs) and *best linear unbiased prediction* (BLUP) (Robinson, 1991); using the latent noise formulation, the model can explain away bias in the residuals as in BLUP. On the other hand, LMMs have a random term for each sample and target variable. While LMMs are not computationally feasible to generalize for high-dimensional targets due to the NK random effect parameters and the associated inversion of an $NK \times NK$ covariance matrix, the latent-noise BRRR can be seen as a low-rank generalization of LMMs for high-dimensional target variables: the covariates are used for prediction in the latent space and in this space there is a noise term for each sample and dimension. Therefore, the number of random effect parameters stays at NS_1 and inference remains tractable.

In summary, our findings extend the existing literature on modeling structured noise in an important way by showing that structured noise can, and should, be taken advantage of when learning the interesting effects between the covariates and the target variables, and how this can be done. Code in R for the new method is available for download at <http://research.cs.aalto.fi/pml/software/latentNoise/>.

Acknowledgments

This work was financially supported by the Academy of Finland (the Finnish Centre of Excellence in Computational Inference Research COIN; grant numbers 259272 and 286607 to PM; grant number 257654 to MP; grants numbered 140057, 294238 and 292334 to SK).

This work was also supported by the "Machine Learning for Augmented Science and Knowledge Work" project of Aalto University funded by Tekes the Finnish Funding Agency for Innovation (Duro 1718/31/2014).

NFBC1966 received financial support from the Academy of Finland (project grants 104781, 120315, 129269, 1114194, 24300796, Center of Excellence in Complex Disease Genetics and SALVE), University Hospital Oulu, Biocenter, University of Oulu, Finland (75617), NHLBI grant 5R01HL087679-02 through the STAMPEED program (1RL1MH083268-01), NIH/NIMH (5R01MH63706:02), ENGAGE project and grant agreement HEALTH-F4-2007-201413, EU FP7 EurHEALTHAgeing-277849 and the Medical Research Council, UK (G0500539, G0600705, G1002319, PrevMetSyn/SALVE).

The development and applications of the quantitative serum NMR metabolomics platform are supported by the Academy of Finland, the Sigrid Juselius Foundation, Strategic Research Funding from the University of Oulu, the British Heart Foundation, the Wellcome Trust and the Medical Research Council, UK.

We acknowledge the computational resources provided by the Aalto Science-IT project. Disclosure: AJK, PS and MAK are shareholders of Brainshake Ltd., a company offering NMR-based metabolite profiling.

References

- A. Bargi, R. Y. Xu, Z. Ghahramani, and M. Piccardi. A non-parametric conditional factor regression model for multi-dimensional input and response. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33, pages 77–85. JMLR W&CP, 2014.
- J. Baxter. A Bayesian/information theoretic model of bias learning. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, COLT '96, pages 77–88. New York, NY, USA, 1996. ACM.
- A. Bhattacharya and D. Dunson. Sparse Bayesian infinite factor models. *Biometrika*, 98(2):291–306, 2011.
- C. M. Bishop et al. *Pattern recognition and machine learning*, volume 1. Springer, 2006.
- L. Bo and C. Smunichsescu. Supervised spectral latent variable models. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5, pages 33–40. JMLR W&CP, 2009.
- L. Bottolo, E. Petretto, S. Blankenberg, F. Cambien, S. Cook, L. Tiret, and S. Richardson. Bayesian detection of expression quantitative trait loci hot spots. *Genetics*, 189(4):1449–1459, 2011.
- L. Breiman and J. Friedman. Predicting multivariate responses in multiple linear regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(1):3–54, 1997.
- M. Calus and R. Veerkamp. Accuracy of multi-trait genomic selection using different methods. *Genetics Selection Evolution*, 43(1):26, 2011.
- A. Carriero, G. Kapetanios, and M. Marcellino. Forecasting large datasets with Bayesian reduced rank multivariate models. *Journal of Applied Econometrics*, 26(5):735–761, 2011.
- R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.
- O. Davis, G. Band, M. Pirinen, C. Haworth, E. Meaburn, Y. Kovas, N. Harlaar, S. Docherty, K. Hanscombe, M. Tzaskowski, et al. The correlation between reading and mathematics ability at age twelve has a substantial genetic component. *Nature Communications*, 5, 2014.
- A. Evgeniou and M. Pontil. Multi-task feature learning. In *Advances in Neural Information Processing Systems 19*, volume 19, pages 41–48, Cambridge, MA, 2007. The MIT Press.
- M. Ferreira and S. Purcell. A multivariate test of association. *Bioinformatics*, 25(1):132–133, 2009.
- R. Foygel, M. Horrell, M. Drton, and J. Lafferty. Nonparametric reduced rank regression. In *Advances in Neural Information Processing Systems 25*, pages 1637–1645, 2012.
- J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.
- N. Fusi, O. Stegle, and N. Lawrence. Joint modelling of confounding factors and prominent genetic regulators provides increased accuracy in genetical genomics studies. *PLoS Computational Biology*, 8(1):e1002330, 2012.
- A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian data analysis*. Chapman & Hall/CRC, 2004.
- J. Geweke. Bayesian reduced rank regression in econometrics. *Journal of Econometrics*, 75(1):121–146, 1996.
- Global Lipids Genetics Consortium. Discovery and refinement of loci associated with lipid levels. *Nature Genetics*, 45(11):1274–1283, 2013.
- A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.
- M. Inouye, J. Kettunen, P. Soininen, K. Silander, S. Ripatti, L. S Kumpula, E. Hämmäläinen, P. Joussilahti, A. J. Kangas, S. Männistö, et al. Metabonomic, transcriptomic, and genomic variation of a population cohort. *Molecular Systems Biology*, 6(1), 2010.
- M. Inouye, S. Ripatti, J. Kettunen, L. Lyytikäinen, N. Oksala, P. Laurila, A. Kangas, P. Soininen, M. Savolainen, J. Viikari, et al. Novel loci for metabolic networks and multi-tissue expression studies reveal genes for atherosclerosis. *PLoS Genetics*, 8(8):e1002907, 2012.
- H. M. Kang, C. Ye, and E. Eskin. Accurate discovery of expression quantitative trait loci under confounding from spurious and genuine regulatory hotspots. *Genetics*, 180(4):1909–1925, 2008.
- S. Karlsson. Conditional posteriors for the reduced rank regression model. Technical Report Working Papers 2012:11, Örebro University Business School, 2012.
- J. Kettunen, T. Tukiainen, A-P. Sarin, A. Ortega-Alonso, E. Tikkanen, L-P. Lyytikäinen, A. J. Kangas, P. Soininen, P. Würtz, K. Silander, et al. Genome-wide association study identifies multiple loci influencing human serum metabolite levels. *Nature genetics*, 44(3):269–276, 2012.
- S. Kim and E. Xing. Statistical estimation of correlated genome associations to a quantitative trait network. *PLoS Genetics*, 5(8):e1000587, 2009.

- S. Kim, K. Sohn, and E. Xing. A multivariate regression approach to association analysis of a quantitative trait network. *Bioinformatics*, 25(12):i204–i212, 2009.
- A. Klami, S. Virtanen, and S. Kaski. Bayesian canonical correlation analysis. *The Journal of Machine Learning Research*, 14(1):965–1003, 2013.
- G.M. Koop, Rodney W. Strachan, Herman Van Dijk, Mattias Villani, K. Patterson, and T. Mills. *Bayesian approaches to cointegration*, pages 871–898. ISBN 1403941556. Working paper version - Department of Economics, University of Leicester: Discussion Papers in Economics number 04/27.
- A. Krishna, H. D. Bondell, and S. K. Ghosh. Bayesian variable selection using an adaptive powered correlation prior. *Journal of Statistical Planning and Inference*, 139(8):2665–2674, 2009.
- J. T. Leek and J. D. Storey. Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genetics*, 3(9):e161, 2007.
- P. Marttinen, J. Gillberg, A. Havulinna, J. Corander, and S. Kaski. Genome-wide association studies with high-dimensional phenotypes. *Statistical Applications in Genetics and Molecular Biology*, 12(4):413–431, 2013.
- P. Marttinen, M. Pirinen, A-P. Sarin, J. Gillberg, J. Kettunen, I. Surakka, A. J. Kangas, P. Soiminen, P. O'Reilly, M. Kaakinen, M. Kähönen, T. Lehtimäki, M. Ala-Korpela, O. T. Raitakari, V. Salomaa, M. R. Järvelin, S. Ripatti, and S. Kaski. Assessing multivariate gene-metabolome associations with rare variants using Bayesian reduced rank regression. *Bioinformatics*, 2014.
- P. Rai, A. Kumar, and H. Daume III. Simultaneously leveraging output and task structures for multiple-output regression. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 3194–3202. Curran Associates, Inc., 2012.
- B. Rakitsch, C. Lippert, K. Borgwardt, and O. Stegle. It is all in the noise: Efficient multi-task gaussian process inference with structured residuals. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1466–1474. Curran Associates, Inc., 2013.
- P. Rautakallio. Groups at risk in low birth weight infants and perinatal mortality. *Acta Paediatrica Scandinavica*, 193:Suppl–193, 1969.
- G. K. Robinson. That blup is a good thing: The estimation of random effects. *Statistical science*, pages 15–32, 1991.
- K.-A. Sohn and S. Kim. Joint estimation of structured sparsity and output structure in multiple-output regression via inverse-covariance regularization. In N. Lawrence and M. Girolami, editors, *International Conference on Artificial Intelligence and Statistics*, volume 22, pages 1081–1089. JMLR W&CP, 2012.

- P. Soiminen, A. Kangas, P. Würtz, T. Tukiaainen, T. Tynkkynen, R. Laatikainen, M. Järvelin, M. Kähönen, T. Lehtimäki, J. Viikari, et al. High-throughput serum NMR metabolomics for cost-effective holistic studies on systemic metabolism. *Analyst*, 134(9):1781–1785, 2009.
- O. Stegle, L. Parts, R. Durbin, and J. Winn. A Bayesian framework to account for complex non-genetic factors in gene expression levels greatly increases power in eqtl studies. *PLoS Computational Biology*, 6(5):e1000770, 2010.
- O. Stegle, C. Lippert, J. M. Mooij, N. D. Lawrence, and K. M. Borgwardt. Efficient inference in matrix-variate gaussian models with iid observation noise. In J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 630–638. Curran Associates, Inc., 2011.
- O. Stegle, L. Parts, M. Pipari, J. Winn, and R. Durbin. Using probabilistic estimation of expression residuals (peer) to obtain increased power and interpretability of gene expression analyses. *Nature Protocols*, 7(3):500–507, 2012.
- M. Stephens. A unified framework for association analysis with multiple related phenotypes. *PLoS ONE* 8(7): e65245, 2013.
- J. H. Stock and M. W. Watson. Forecasting with many predictors. *Handbook of economic forecasting*, 1:515–554, 2006.
- T. Teslovich, K. Musunuru, A. Smith, A. Edmondson, I. Stylianou, M. Koseki, J. Pirruccello, S. Ripatti, D. Chasman, C. Willer, et al. Biological, clinical and population relevance of 95 loci for blood lipids. *Nature*, 466(7307):707–713, 2010.
- S. Virtanen, A. Klami, and S. Kaski. Bayesian CCA via group sparsity. In Lise Getoor and Tobias Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 457–464, New York, NY, 2011. ACM.
- W. Wang, V. Baladandayuthapani, J. Morris, B. Broom, G. Manyam, and K. Do. Integrative Bayesian analysis of high-dimensional multi-platform genomics data. *Bioinformatics*, 29(2):149–159, 2012.
- L. Wehbe, B. Murphy, P. Talukdar, A. Fyshe, A. Ramdas, and T. Mitchell. Simultaneously uncovering the patterns of brain regions involved in different story reading subprocesses. *PLoS ONE*, 9(11):e112575, 11 2014. doi: 10.1371/journal.pone.0112575. URL <http://dx.doi.org/10.1371/journal.pone.0112575>.
- M. West. Bayesian factor regression models in the large p, small n paradigm. *Bayesian Statistics*, 7:733–742, 2003.
- C. Xu, X. Wang, Z. Li, and S. Xu. Mapping QTL for multiple traits using Bayesian statistics. *Genetical Research*, 91(1):23–37, 2009.
- N. Yi and S. Banerjee. Hierarchical generalized linear models for multiple quantitative trait locus mapping. *Genetics*, 181(3):1101–1113, 2009.

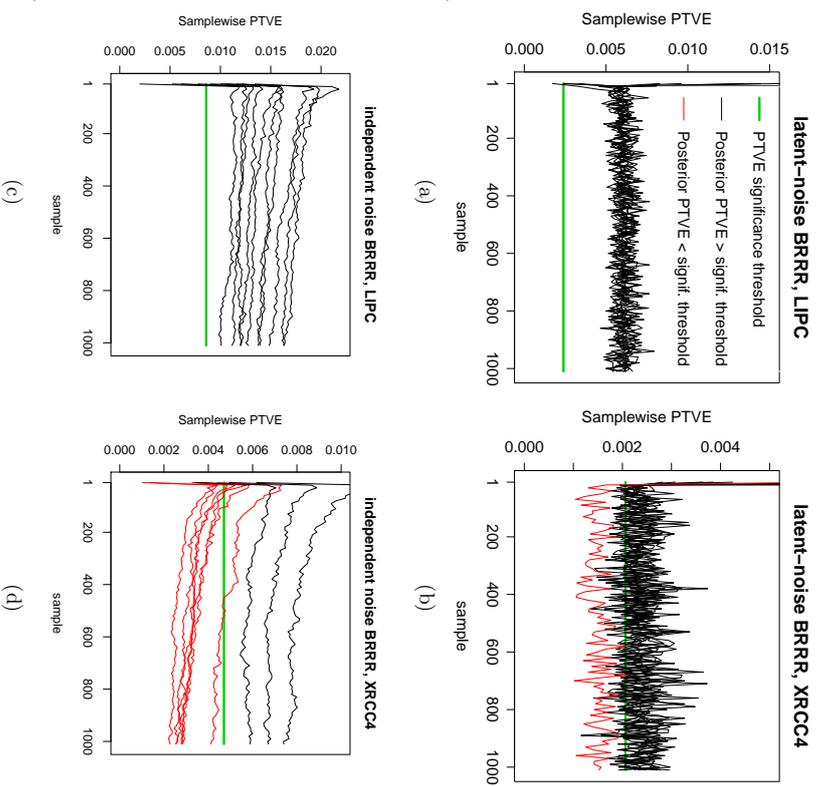


Figure 9: Convergence plots of the association score parameter, that is, the proportion of the total variance explained (PTVE), for latent-noise BRRR and independent-noise BRRR. 10 MCMC chains were computed using data sets with 4702 samples from genes *LIPC* and *XRCC4*. The green line marks the 0.05 significance level of the test score, obtained by permutation sampling. The chains, whose association scores exceed the significance threshold, are drawn in black, whereas the chains that do not exceed it are drawn in red. Latent-noise BRRR converges and mixes appropriately: chains with different initializations converge and traverse the posterior. On the contrary, the independent-noise BRRR behaves rather pathologically: different chains converge to different solutions and explore the posterior poorly.

The Constrained Dantzig Selector with Enhanced Consistency *

Yinfei Kong

*Department of Information Systems and Decision Sciences
Mihaylo College of Business and Economics
California State University at Fullerton
Fullerton, CA 92831, USA*

YIKONG@FULLERTON.EDU

Zemin Zheng

*Department of Statistics and Finance
University of Science and Technology of China
Hefei, Anhui 230026, China*

ZHENGZM@USTC.EDU.CN

Jinchi Lv

*Data Sciences and Operations Department
Marshall School of Business
University of Southern California
Los Angeles, CA 90089, USA*

JINCHILV@MARSHALL.USC.EDU

Editor: Sara van de Geer

Abstract

The Dantzig selector has received popularity for many applications such as compressed sensing and sparse modeling, thanks to its computational efficiency as a linear programming problem and its nice sampling properties. Existing results show that it can recover sparse signals mimicking the accuracy of the ideal procedure, up to a logarithmic factor of the dimensionality. Such a factor has been shown to hold for many regularization methods. An important question is whether this factor can be reduced to a logarithmic factor of the sample size in ultra-high dimensions under mild regularity conditions. To provide an affirmative answer, in this paper we suggest the constrained Dantzig selector, which has more flexible constraints and parameter space. We prove that the suggested method can achieve convergence rates within a logarithmic factor of the sample size of the oracle rates and improved sparsity, under a fairly weak assumption on the signal strength. Such improvement is significant in ultra-high dimensions. This method can be implemented efficiently through sequential linear programming. Numerical studies confirm that the sample size needed for a certain level of accuracy in these problems can be much reduced.

Keywords: Sparse Modeling, Compressed Sensing, Ultra-high Dimensionality, Dantzig Selector, Regularization Methods, Finite Sample

1. Introduction

Due to the advances of technologies, big data problems appear increasingly common in the domains of molecular biology, machine learning, and economics. It is appealing to design procedures that can provide a recovery of important signals, among a pool of potentially huge number of signals, to a desired level of accuracy. As a powerful tool for producing interpretable models, sparse modeling via regularization has gained popularity for analyzing large-scale data sets. A common feature of the theories for many regularization methods is that the rates of convergence under the prediction or estimation loss usually involve the logarithmic factor of the dimensionality $\log p$; see, for example, Bickel et al. (2009), van de Geer et al. (2011), and Fan and Lv (2013), among many others. From the asymptotic point of view, such a factor may be negligible or insignificant when p is not too large. It can, however, become no longer negligible or even significant in finite samples with large dimensionality, particularly when a relatively small sample size is considered or preferred. In such cases, the recovered signals and sparse models may tend to be noisy. Another consequence of this effect is that many noise variables tend to appear together with recovered important ones (Candès et al., 2008). An important and interesting question is whether such a factor can be reduced, say, to a logarithmic factor of the sample size $\log n$, from ultra-high dimensions under mild regularity conditions.

In high-dimensional variable selection, there is a fast growing literature for different kinds of regularization methods with well established rates of convergence under the estimation and prediction losses. For instance, Candès and Tao (2007) proposed the L_1 -regularization approach of the Dantzig selector by relaxing the normal equation to allow for correlation between the residual vector and all the variables, which can recover sparse signals as accurately as the ideal procedure, knowing the true underlying sparse model, up to a logarithmic factor $\log p$. Later, Bickel et al. (2009) showed that the popularly used Lasso estimator (Tibshirani, 1996) exhibits similar behavior as the Dantzig selector with a $\log p$ factor in the oracle inequalities for the prediction risk and bounds on the estimation losses in general nonparametric regression models. Furthermore, it has been proved in Raskutti et al. (2011) that the minimax rates of convergence for estimating the true coefficient vector in the high-dimensional linear regression model involve a logarithmic factor $\log p$ in the L_2 -loss and prediction loss, under some regularity conditions on the design matrix. Other work for desired properties such as the oracle property on the L_1 and more general regularization includes Fan and Li (2001), Fan and Peng (2004), Zou and Hastie (2005), Zou (2006), van de Geer (2008), Lv and Fan (2009), Antoniadis et al. (2010), Städler et al. (2010), Fan and Lv (2011), Negahban et al. (2012), Chen et al. (2013), and Fan and Tang (2013), among many others.

A typical way for reducing the logarithmic factor $\log p$ to $\log n$ is through model selection consistency, where the estimator selects exactly the support of the true coefficient vector, that is, the set of variables with nonzero coefficients. We refer the reader to Zhao and Yu (2006), Wainwright (2009), Zhang (2010), and Zhang (2011) for analysis of model selection consistency of regularization methods. Since the true parameters are assumed to be sparse in the high-dimensional setting, consistent variable selection can greatly lessen the analytical complexity from large dimensionality p to around oracle model size. Once model selection consistency is established for the estimator with significant probability, an analysis

*. This work was supported by the NSF CAREER Award DMS-0955316, a grant from the Simons Foundation, and USC Diploma in Innovation Grant. We sincerely thank the Action Editor and two referees for their valuable comments that have helped improve the paper significantly.

constrained on that event will give a factor $\log n$ instead of $\log p$ in the rates of convergence under various estimation and prediction losses. However, to obtain model selection consistency it usually requires a uniform signal strength condition that the minimum magnitude of nonzero coefficients is at least of order $\{s(\log p)/n\}^{1/2}$ (Fan and Lv, 2013) and (Zheng et al., 2014), where s stands for the size of the low-dimensional parameter vector. We doubt the necessity of the model selection consistency and the uniform signal strength of order $\{s(\log p)/n\}^{1/2}$ for achieving the logarithmic factor $\log n$ in ultra-high dimensionality.

In this paper, we suggest the constrained Dantzig selector to study the rates of convergence with weaker signal strength assumption. The constrained Dantzig selector replaces the constraint Dantzig constraint on correlations between the variables and the residual vector by a more flexible one, and considers a constrained parameter space distinguishing between zero parameters and significantly nonzero parameters. The main contributions of this paper are threefold. First, the convergence rates for the constrained Dantzig selector are shown to be within a factor of $\log n$ of the oracle rates instead of $\log p$, a significant improvement in the case of ultra-high dimensionality and relatively small sample size. It is appealing that such an improvement is made with a fairly weak assumption on the signal strength without requiring model selection consistency. To the best of our knowledge, this assumption seems to be the weakest one in the literature of similar results; see, for example, Bickel et al. (2009) and Zheng et al. (2014). Two parallel theorems, under the uniform uncertainty principle condition and the restricted eigenvalue assumption, are established on the properties of the constrained Dantzig selector for compressed sensing and sparse modeling, respectively. Second, compared to the Dantzig selector, theoretical results of this paper show that the number of falsely discovered signs of our new selector, with an explicit inverse relationship to the signal strength, is controlled as a possibly asymptotically vanishing fraction of the true model size. Third, an active-set based algorithm is introduced to implement the constrained Dantzig selector efficiently. An appealing feature of this algorithm is that its convergence can be checked easily.

The rest of the paper is organized as follows. In Section 2, we introduce the constrained Dantzig selector. We present its compressed sensing and sampling properties in Section 3. In Section 4, we discuss the implementation of the method and present several simulation and real data examples. We provide some discussions of our results and some possible extensions of our method in Section 5. All technical details are relegated to the Appendix.

2. The Constrained Dantzig selector

To simplify the technical presentation, we adopt the model setting in Candès and Tao (2007) and present the main ideas focusing on the linear regression model

$$y = \mathbf{X}\beta + \epsilon, \quad (1)$$

where $y = (y_1, \dots, y_n)^T$ is an n -dimensional response vector, $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_p)$ is an $n \times p$ design matrix consisting of p covariate vectors \mathbf{x}_j^T 's, $\beta = (\beta_1, \dots, \beta_p)^T$ is a p -dimensional regression coefficient vector, and $\epsilon = (\epsilon_1, \dots, \epsilon_n)^T \sim N(\mathbf{0}, \sigma^2 \mathbf{I}_n)$ for some positive constant σ is an n -dimensional error vector independent of \mathbf{X} . The normality assumption is considered for simplicity, and all the results in the paper can be extended to the cases of bounded errors

or light-tailed error distributions without much difficulty. See, for example, the technical analysis in Fan and Lv (2011) and Fan and Lv (2013).

In both problems of compressed sensing and sparse modeling, we are interested in recovering the support and nonzero components of the true regression coefficient vector $\beta_0 = (\beta_{0,1}, \dots, \beta_{0,p})^T$, which we assume to be sparse with s nonzero components, for the case when the dimensionality p may greatly exceed the sample size n . Throughout this paper, p is implicitly understood as $\max\{n, p\}$ and $s \leq \min\{n, p\}$ to ensure model identifiability. To align the scale of all covariates, we assume that each column of \mathbf{X} , that is, each covariate vector \mathbf{x}_j , is rescaled to have L_2 -norm $n^{1/2}$, matching that of the constant covariate vector $\mathbf{1}$. The Dantzig selector (Candès and Tao, 2007) is defined as

$$\hat{\beta}_{\text{DS}} = \operatorname{argmin}_{\beta \in \mathbb{R}^p} \|\beta\|_1 \quad \text{subject to} \quad \|n^{-1}\mathbf{X}^T(y - \mathbf{X}\beta)\|_\infty \leq \lambda_1, \quad (2)$$

where $\lambda_1 \geq 0$ is a regularization parameter. The above constant Dantzig selector constraint on correlations between all covariates and the residual vector may not be flexible enough to differentiate important covariates and noise covariates. We introduce an extension of the Dantzig selector, the constrained Dantzig selector, defined as

$$\begin{aligned} \hat{\beta}_{\text{CDS}} = \operatorname{argmin}_{\beta \in \mathbb{R}^p} \quad & \|\beta\|_1 \\ \text{subject to} \quad & |n^{-1}\mathbf{x}_j^T(y - \mathbf{X}\beta)| \leq \lambda_0 \mathbf{1}_{\{|\beta_j| \geq \lambda\}} + \lambda_1 \mathbf{1}_{\{|\beta_j| = 0\}} \\ & \text{for } j = 1, \dots, p, \end{aligned} \quad (3)$$

where $\lambda_0 \geq 0$ is a regularization parameter and $\mathcal{B}_\lambda = \{\beta \in \mathbb{R}^p : \beta_j = 0 \text{ or } |\beta_j| \geq \lambda \text{ for each } j\}$ is the constrained parameter space for some $\lambda \geq 0$. When we choose $\lambda = 0$ and $\lambda_0 = \lambda_1$, the constrained Dantzig selector becomes the Dantzig selector. Throughout this paper, we choose the regularization parameters λ_0 and λ_1 as $c_0\{(\log n)/n\}^{1/2}$ and $c_1\{(\log p)/n\}^{1/2}$, respectively, with $\lambda_0 \leq \lambda_1$ as well as c_0 and c_1 two sufficiently large positive constants, and assume that λ is a parameter greater than λ_1 . The two parameters λ_0 and λ_1 differentially bound two types of correlations: on the support of the constrained Dantzig selector, the correlations between covariates and residuals are bounded, up to a common scale, by λ_0 ; on its complement, however, the correlations are bounded through λ_1 . In the ultra-high dimensional case, meaning $\log p = O(n^\alpha)$ for some $0 < \alpha < 1$, the constraints involving λ_0 are tighter than those involving λ_1 , in which λ_1 is a universal regularization parameter for the Dantzig selector; see Candès and Tao (2007) and Bickel et al. (2009).

We now provide more insights into the new constraints in the constrained Dantzig selector. First, it is worthwhile to notice that if $\beta_0 \in \mathcal{B}_\lambda$, β_0 can satisfy new constraints with large probability in model setting (1); see the proof of Theorem 1. With the tighter constraints, the feasible set of the constrained Dantzig selector problem is a subset of that of the Dantzig selector problem, resulting in a search of the solution in a reduced space. Second, it is appealing to extract more information in important covariates, leading to lower correlations between those variables and the residual vector. In this spirit, the constrained Dantzig selector puts tighter constraints on the correlations between selected variables and residuals. Third, the constrained Dantzig selector is defined on the constrained parameter space \mathcal{B}_λ , which has been introduced in Fan and Lv (2013). Such a space also shares some similarity to the union of coordinate subspaces considered in Fan and Lv (2011) for characterizing the restricted global optimality of nonconcave penalized likelihood estimators.

The threshold λ in \mathcal{B}_λ distinguishes between important covariates with strong effects and noise covariates with weak effects. As shown in Fan and Lv (2013), this feature can lead to improved sparsity and effectively prevent overfitting by making it harder for noise covariates to enter the model.

3. Main results

In this section, two parallel theoretical results are introduced based on the uniform uncertainty principle (UUP) condition and restricted eigenvalue assumption, respectively. Although the UUP condition may be relatively stringent in some applications, we still present one theorem for our method under this condition in Section 3.1 for the purpose of comparison with the original Dantzig selector.

3.1 Nonasymptotic compressed sensing properties

Since the Dantzig selector was introduced partly for applications in compressed sensing, we first study the nonasymptotic compressed sensing properties of the constrained Dantzig selector by adopting the theoretical framework in Candès and Tao (2007). They introduced the uniform uncertainty principle condition defined as follows. Denote by \mathbf{X}_T a submatrix of \mathbf{X} consisting of columns with indices in a set $T \subset \{1, \dots, p\}$. For the true model size s , define the s -restricted isometry constant of \mathbf{X} as the smallest constant δ_s such that

$$(1 - \delta_s) \|\mathbf{h}\|_2^2 \leq n^{-1} \|\mathbf{X}_T \mathbf{h}\|_2^2 \leq (1 + \delta_s) \|\mathbf{h}\|_2^2$$

for any set T with size at most s and any vector \mathbf{h} . This condition requires that each submatrix of \mathbf{X} with at most s columns behaves similarly as an orthonormal system. Another constant, the s -restricted orthogonality constant, is defined as the smallest quantity $\theta_{s,2s}$ such that

$$n^{-1} |(\mathbf{X}_T \mathbf{h}, \mathbf{X}_{T'} \mathbf{h}')| \leq \theta_{s,2s} \|\mathbf{h}\|_2 \|\mathbf{h}'\|_2$$

for all pairs of disjoint sets $T, T' \subset \{1, \dots, p\}$ with $|T| \leq s$ and $|T'| \leq 2s$ and any vectors \mathbf{h}, \mathbf{h}' . The uniform uncertainty principle condition is simply stated as

$$\delta_s + \theta_{s,2s} < 1. \quad (4)$$

For notational simplicity, we drop the subscripts and denote these two constants by δ and θ , respectively. Without loss of generality, assume that $\text{supp}(\beta_0) = \{1 \leq j \leq p : \beta_{0,j} \neq 0\} = \{1, \dots, s\}$ hereafter. To evaluate the sparse recovery accuracy, we consider the number of falsely discovered signs defined as $\text{FS}(\hat{\beta}) = |\{j = 1, \dots, p : \text{sgn}(\hat{\beta}_j) \neq \text{sgn}(\beta_{0,j})\}|$ for an estimator $\hat{\beta} = (\hat{\beta}_1, \dots, \hat{\beta}_p)^T$. Now we are ready to present the nonasymptotic compressed sensing properties of the constrained Dantzig selector.

Theorem 1 *Assume that the uniform uncertainty principle condition (4) holds and $\beta_0 \in \mathcal{B}_\lambda$ with $\lambda \geq C^{1/2}(1 + \lambda_1/\lambda_0)\lambda_1$ for some positive constant C . Then with probability at least*

$1 - O(n^{-c})$ for $c = (c_0^2 \wedge c_1^2)/(2\sigma^2) - 1$, the constrained Dantzig selector $\hat{\beta}$ satisfies that

$$\begin{aligned} \|\hat{\beta} - \beta_0\|_1 &\leq 2\sqrt{5}(1 - \delta - \theta)^{-1} c_0 s \sqrt{(\log n)/n}, \\ \|\hat{\beta} - \beta_0\|_2 &\leq \sqrt{5}(1 - \delta - \theta)^{-1} c_0 \sqrt{s(\log n)/n}, \\ \text{FS}(\hat{\beta}) &\leq C c_1^2 s (\log p) / (n \lambda^2). \end{aligned}$$

If in addition $\lambda > \sqrt{5}(1 - \delta - \theta)^{-1} c_0 \sqrt{s(\log n)/n}$, then with the same probability, it also holds that $\text{sgn}(\hat{\beta}) = \text{sgn}(\beta_0)$ and $\|\hat{\beta} - \beta_0\|_\infty \leq 2c_0 \|(n^{-1} \mathbf{X}_1^T \mathbf{X}_1)^{-1}\|_\infty \sqrt{(\log n)/n}$, where \mathbf{X}_1 is an $n \times s$ submatrix of \mathbf{X} corresponding to s nonzero $\beta_{0,j}$'s.

The constant c in the above probability bound can be sufficiently large since both constants c_0 and c_1 are assumed to be large, while the constant C comes from Theorem 1.1 in Candès and Tao (2007); see the proof in the Appendix for details. In the above bound on the L_∞ -estimation loss, it holds that $\|(n^{-1} \mathbf{X}_1^T \mathbf{X}_1)^{-1}\|_\infty \leq s^{1/2} \|(n^{-1} \mathbf{X}_1^T \mathbf{X}_1)^{-1}\|_2 \leq (1 - \delta)^{-1} s^{1/2}$. See Section 3.2 for more discussion on this quantity.

From Theorem 1, we see improvements of the constrained Dantzig selector over the Dantzig selector, which has a convergence rate, in terms of the L_2 -estimation loss, up to a factor $\log p$ of that for the ideal procedure. However, the sparsity property of the Dantzig selector was not investigated in Candès and Tao (2007). In contrast, the constrained Dantzig selector is shown to have an inverse quadratic relationship between the number of falsely discovered signs and the threshold λ , revealing that its model selection accuracy increases with the signal strength. The number of falsely discovered signs can be controlled below or as an asymptotically vanishing fraction of the true model size, since $\text{FS}(\hat{\beta}) \leq Cs(\lambda_1/\lambda)^2 \leq (1 + \lambda_1/\lambda)^{-2} s < s$ by assuming $\lambda > C^{1/2}(1 + \lambda_1/\lambda_0)\lambda_1$.

Another advantage of the constrained Dantzig selector lies in its convergence rates. In the case of ultra-high dimensionality which is typical in compressed sensing applications, its prediction and estimation losses can be reduced from the logarithmic factor $\log p$ to $\log n$ with overwhelming probability. In particular, only a fairly weak assumption on the signal strength is imposed to attain such improved convergence rates. In fact, it has been shown in Raskutti et al. (2011) that without any condition on the signal strength, the minimax convergence rate of L_2 risk has an upper bound of order $O\{s^{1/2}(\log p)/n\}$. Especially, they claimed that the Dantzig selector can achieve such minimax rate, but requires a relatively stronger condition on the design matrix than nonconvex optimization algorithms to determine the minimax upper bounds. We push a step forward that the constrained Dantzig selector, with additional signal strength conditions, can attain the L_2 -estimation loss of a smaller order than $O\{s^{1/2}(\log p)/n\}$ in ultra-high dimensions.

There exist other methods which have been shown to enjoy convergence rates of the same order as well, for example, in Zheng et al. (2014) for high-dimensional thresholded regression. However, these results usually rely on a stronger condition on signal strength, such as, the minimum signal strength is at least of order $\{s(\log p)/n\}^{1/2}$. In another work, Fan and Lv (2011) showed that the nonconcave penalized estimator can have a consistency rate of $O_p\{s^{1/2} n^{-\gamma} \log n\}$ for some $\gamma \in (0, 1/2]$ under the L_2 -estimation loss, which can be slower than our rate of convergence. More detailed discussion on the relationship between the faster rates of convergence and the assumptions on the signal strength and sparsity can be found in Section 3.2. A main implication of our improved convergence rates is that

a smaller number of observations will be needed for the constrained Dantzig selector to attain the same level of accuracy in compressed sensing, as the Dantzig selector, which is demonstrated in Section 4.2.

3.2 Sampling properties

The properties of the Dantzig selector have also been extensively investigated in Bickel et al. (2009). They introduced the restricted eigenvalue assumption with which the oracle inequalities under various prediction and estimation losses were derived. We adopt their theoretical framework and study the sampling properties of the constrained Dantzig selector under the restricted eigenvalue assumption stated below. A positive integer m is said to be in the same order of s if m/s can be bounded from both above and below by some positive constants.

Condition 1 For some positive integer m in the same order of s , there exists some positive constant κ such that $\|n^{-1/2}\mathbf{X}\boldsymbol{\delta}\|_2 \geq \kappa \max\{\|\boldsymbol{\delta}\|_{1/2}, \|\boldsymbol{\delta}'_1\|_2\}$ for all $\boldsymbol{\delta} \in \mathbb{R}^p$ satisfying $\|\boldsymbol{\delta}_2\|_1 \leq \|\boldsymbol{\delta}'_1\|_1$, where $\boldsymbol{\delta} = (\boldsymbol{\delta}'_1, \boldsymbol{\delta}'_2)^T$, $\boldsymbol{\delta}_1$ is a subvector of $\boldsymbol{\delta}$ consisting of the first s components, and $\boldsymbol{\delta}'_1$ is a subvector of $\boldsymbol{\delta}_2$ consisting of the $\max\{m, C_m s(\lambda_1/\lambda)^2\}$ largest components in magnitude, with C_m some positive constant.

Condition 1 is a basic assumption on the design matrix \mathbf{X} for deriving the oracle inequalities of the Dantzig selector. Since we assume that $\text{supp}(\boldsymbol{\beta}_0) = \{1, \dots, s\}$, Condition 1 indeed plays the same role as the restricted eigenvalue assumption $\text{RE}(s, m, 1)$ in Bickel et al. (2009), which assumes the inequality in Condition 1 holds for any subset with size no larger than s to cover all possibilities of $\text{supp}(\boldsymbol{\beta}_0)$. See (10) in the Appendix for insights into the basic inequality $\|\boldsymbol{\delta}_2\|_1 \leq \|\boldsymbol{\delta}'_1\|_1$ and Bickel et al. (2009) for more detailed discussions on this assumption.

Theorem 2 Assume that Condition 1 holds and $\boldsymbol{\beta}_0 \in \mathcal{B}_\lambda$ with $\lambda \geq C_m^{1/2}(1 + \lambda_1/\lambda_0)\lambda_1$. Then the constrained Dantzig selector $\widehat{\boldsymbol{\beta}}$ satisfies with the same probability as in Theorem 1 that

$$\begin{aligned} n^{-1/2}\|\mathbf{X}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0)\|_2 &= O(\kappa^{-1}\sqrt{s(\log n)/n}), & \|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\|_1 &= O(\kappa^{-2}s\sqrt{(\log n)/n}), \\ \|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\|_2 &= O(\kappa^{-2}\sqrt{s(\log n)/n}), & \text{FS}(\widehat{\boldsymbol{\beta}}) &\leq C_m c_1^2 s(\log \rho)/(n\lambda^2). \end{aligned}$$

If in addition $\lambda > 2\sqrt{5}\kappa^{-2}c_0s^{1/2}\sqrt{(\log n)/n}$, then with the same probability it also holds that $\text{sgn}(\widehat{\boldsymbol{\beta}}) = \text{sgn}(\boldsymbol{\beta}_0)$ and $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\|_\infty = O\{(n^{-1}\mathbf{X}_1^T\mathbf{X}_1)^{-1}\|_\infty\sqrt{(\log n)/n}$.

Theorem 2 establishes asymptotic results on the sparsity and oracle inequalities for the constrained Dantzig selector under the restricted eigenvalue assumption. This assumption, which is an alternative to the uniform uncertainty principle condition, has also been widely employed in high-dimensional settings. In Bickel et al. (2009), an approximate equivalence of the Lasso estimator (Tibshirani, 1996) and Dantzig selector was proved under this assumption, and the Lasso estimator was shown to be sparse with size $O(\phi_{\max}^s)$, where ϕ_{\max} is the largest eigenvalue of the Gram matrix $n^{-1}\mathbf{X}^T\mathbf{X}$. In contrast, the constrained Dantzig selector gives a sparser model under the restricted eigenvalue assumption, since its number

of falsely discovered signs $\text{FS}(\widehat{\boldsymbol{\beta}}) \leq C_m s(\lambda_1/\lambda)^2 = o(s)$ when $\lambda_1 = o(\lambda)$. Similar as in Theorem 1, the constrained Dantzig selector improves over both Lasso and the Dantzig selector in terms of convergence rates, a reduction of the log p factor to log n .

Let us now take a closer look at the relationship between the faster rates of convergence and the assumptions on the signal strength and sparsity. Observe that the enhanced consistency results require the minimum signal strength to be at least of order $(\log p)/\sqrt{n \log n}$, in view of the assumption $\lambda \geq C_m^{1/2}(1 + \lambda_1/\lambda_0)\lambda_1$. Assume for simplicity that $\|\boldsymbol{\beta}_0\|_2$ is bounded away from both zero and ∞ . Then the order of the minimum signal strength yields an upper bound on the sparsity level $s = O\{n(\log n)/(\log p)^2\}$, which means that the sparsity s is required to be smaller when the dimensionality p becomes larger. In the ultra-high dimensional setting of $\log p = O(n^\alpha)$ with $0 < \alpha < 1$, we then have $s = O(n^{1-2\alpha} \log n)$. Thus the classical convergence rates involving $s(\log p)/n = O(n^{-\alpha} \log n)$ still go to zero asymptotically, while the rates established in our paper are improved with $s(\log p)/n$ replaced by $s(\log n)/n = O\{n^{-2\alpha}(\log n)^2\}$. We see a gain on the convergence rate of a factor $n^\alpha/(\log n)$ at the cost of the aforementioned signal strength and sparsity assumptions.

One can see that results in Theorems 1 and 2 are approximately equivalent, while the latter presents an additional oracle inequality on the prediction loss. An interesting phenomenon is that if adopting a simpler version, that is, the Dantzig selector equipped with the thresholding constraint only, we can also obtain similar results as in Theorems 1 and 2, but with a stronger condition on signal strength such as $\lambda \gg s^{1/2}\lambda_1$. In this sense, the constrained Dantzig selector is also an extension of the Dantzig selector equipped with the thresholding constraint only, but enjoys better properties. Some comprehensive results on the prediction and variable selection properties have also been established in Fan and Lv (2013) for various regularization methods, revealing their asymptotic equivalence in the thresholded parameter space. However, as mentioned in Section 3.1, improved rates as in Theorem 2 commonly require a stronger assumption on the signal strength, which is $\boldsymbol{\beta}_0 \in \mathcal{B}_\lambda$ with $\lambda \gg s^{1/2}\lambda_1$; see, for example, Theorem 2 of Fan and Lv (2013).

For the quantity $\|(n^{-1}\mathbf{X}_1^T\mathbf{X}_1)^{-1}\|_\infty$ in the above bound on the L_∞ -estimation loss, if \mathbf{X}_1 takes the form of a common correlation matrix $(1 - \rho)\mathbf{I}_s + \rho\mathbf{1}_s\mathbf{1}_s^T$ for some $\rho \in [0, 1)$, it is easy to check that $\|(n^{-1}\mathbf{X}_1^T\mathbf{X}_1)^{-1}\|_\infty = (1 - \rho)^{-1}\{1 + (2s - 3)\rho\}/(1 + (s - 1)\rho)$, which is bounded regardless of the value of s .

3.3 Asymptotic properties of computable solutions

The nonasymptotic and sampling properties of the constrained Dantzig selector, as the global minimizer, have been established in Sections 3.1 and 3.2, respectively. However, it is not guaranteed that the global minimizer can be generated by a computational algorithm. Moreover, a computable solution, generated by any algorithm, may only be a local minimizer in many cases. Under certain regularity conditions, we demonstrate that the local minimizer of our method can still share the same nice asymptotic properties as the global one.

Theorem 3 Let $\widehat{\boldsymbol{\beta}}$ be a computable local minimizer of (3). Assume that there exist some positive constants c_2, κ_0 and sufficiently large positive constant c_3 such that $\|\widehat{\boldsymbol{\beta}}\|_0 \leq c_2s$ and $\min_{\|\boldsymbol{\delta}\|_2=1, \|\boldsymbol{\delta}'_0\|_{\leq c_3s}} n^{-1/2}\|\mathbf{X}\boldsymbol{\delta}\|_2 \geq \kappa_0$. Then under conditions of Theorem 2, $\widehat{\boldsymbol{\beta}}$ enjoys the same properties as the global minimizer in Theorem 2.

In Section 4.1, we introduce an efficient algorithm that gives us a local minimizer. Theorem 3 indicates that the obtained solution can also enjoy the asymptotic properties in Theorem 2 under the extra assumptions that $\widehat{\beta}$ is a sparse solution with the number of nonzero components comparable with s and the design matrix \mathbf{X} satisfies a sparse eigenvalue condition. Similar results for the computable solution can be found in Fan and Lv (2013, 2014), where the local minimizer is additionally assumed to satisfy certain constraint on the correlation between the residual vector and all the covariates.

4. Numerical studies

In this section, we first introduce an algorithm which can efficiently implement the constrained Dantzig selector. Then several simulation studies and two real data examples are presented to evaluate the performance of our method.

4.1 Implementation

The constrained Dantzig selector defined in (3) depends on tuning parameters λ_0 , λ_1 , and λ . We suggest some fixed values for λ_0 and λ to simplify the computation, since the proposed method is generally not that sensitive to λ_0 and λ as long as they fall in certain ranges. In simulation studies to be presented, a value around $\{(\log p)/n\}^{1/2}$ for λ and a smaller value for λ_0 , say $0.05\{(\log p)/n\}^{1/2}$ or $0.1\{(\log p)/n\}^{1/2}$, can provide us nice prediction and estimation results. The value of λ_0 is chosen to be smaller than $\{(\log p)/n\}^{1/2}$ to mitigate the selection of noise variables and facilitate sparse modeling. The performance of our method with respect to different values of λ_0 and λ is shown in simulation Example 2 in Section 4.2, which is a typical example that illustrates the robustness of the proposed method with respect to λ_0 and λ .

For fixed λ_0 and λ , we exploit the idea of sequential linear programming to produce the solution path of the constrained Dantzig selector as λ_1 varies. Choose a grid of values for the tuning parameter λ_1 in decreasing order with the first one being $\|n^{-1}\mathbf{X}^T\mathbf{y}\|_\infty$. It is easy to check that $\beta = \mathbf{0}$ satisfies all the constraints in (3) for $\lambda_1 = \|n^{-1}\mathbf{X}^T\mathbf{y}\|_\infty$, and thus the solution is $\widehat{\beta}_{\text{CNS}} = \mathbf{0}$ in this case. For each λ_1 in the grid, we use the solution from the previous one in the grid as an initial value to speed up the convergence. For a given λ_1 , we define an active set, iteratively update this set, and solve the constrained Dantzig selector problem. We name this algorithm as the CDS algorithm which is detailed in four steps below.

1. For a fixed λ_1 in the grid, denote by $\widehat{\beta}_{\lambda_1}^{(0)}$ the initial value. Let $\widehat{\beta}_{\lambda_1}^{(0)}$ be zero when $\lambda_1 = \|n^{-1}\mathbf{X}^T\mathbf{y}\|_\infty$, and the estimate from previous λ_1 in the grid otherwise.
2. Denote by $\widehat{\beta}_{\lambda_1}^{(k)}$ the estimate from the k th iteration. Define the active set \mathcal{A} as the support of $\widehat{\beta}_{\lambda_1}^{(k)}$ and \mathcal{A}^c its complement. Let \mathbf{b} be a vector with constant components λ_0 on \mathcal{A} and λ_1 on \mathcal{A}^c . For the $(k+1)$ th iteration, update \mathcal{A} as $\mathcal{A} \cup \{j \in \mathcal{A}^c : |n^{-1}\mathbf{x}_j^T(\mathbf{y} - \mathbf{X}\widehat{\beta}_{\lambda_1}^{(k)})| > \lambda_1\}$, where the subscript \mathcal{A} indicates a subvector restricted on

4. Solve the following linear program on the new set \mathcal{A} :

$$\widehat{\beta}_{\mathcal{A}} = \operatorname{argmin} \|\beta_{\mathcal{A}}\|_1 \text{ subject to } |n^{-1}\mathbf{X}_{\mathcal{A}}^T(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\beta_{\mathcal{A}})| \leq \mathbf{b}_{\mathcal{A}}, \quad (5)$$

where \leq is understood as componentwise no larger than and the subscript \mathcal{A} also indicates a submatrix with columns corresponding to \mathcal{A} . For the solution obtained in (5), set all its components smaller than λ in magnitude to zero.

3. Update the active set \mathcal{A} as the support of $\widehat{\beta}_{\mathcal{A}}$. Solve the Dantzig selector problem on this active set with λ_0 as the regularization parameter:

$$\widehat{\beta}_{\mathcal{A}} = \operatorname{argmin} \|\beta_{\mathcal{A}}\|_1 \text{ subject to } \|n^{-1}\mathbf{X}_{\mathcal{A}}^T(\mathbf{y} - \mathbf{X}_{\mathcal{A}}\beta_{\mathcal{A}})\|_\infty \leq \lambda_0. \quad (6)$$

Let $\widehat{\beta}_{\mathcal{A}}^{(k+1)} = \widehat{\beta}_{\mathcal{A}}$ and $\widehat{\beta}_{\mathcal{A}^c}^{(k+1)} = \mathbf{0}$, which give the solution for the $(k+1)$ th iteration.

4. Repeat steps 2 and 3 until convergence for a fixed λ_1 and record the estimate from the last iteration as $\widehat{\beta}_{\lambda_1}$. Jump to the next λ_1 if $\widehat{\beta}_{\lambda_1} \in \mathcal{B}_{\lambda_1}$, and stop the algorithm otherwise.

With the solution path produced, we use the cross-validation to select the tuning parameter λ_1 . One can also tune λ_0 and λ similarly as for λ_1 , but as suggested before, some fixed values for them suffice to obtain satisfactory results.

The rationales of the constrained Dantzig selector algorithm are as follows. Step 1 defines the initial value (0th iteration) for each λ_1 in the grid. In step 2, starting with a smaller active set, we add variables that violate the constrained Dantzig selector constraints to eliminate such conflict. As a consequence, some components of $\mathbf{b}_{\mathcal{A}}$ are of value λ_1 instead of λ_0 . Therefore, we need to further solve (6) in step 3 by noting that restricted on its support, the constrained Dantzig selector should be a solution to the Dantzig selector problem with parameter λ_0 . An early stopping of the solution path is imposed in step 4 to make this algorithm computationally more efficient.

An appealing feature of this algorithm is that its convergence can be checked easily. Once there are no more variables violating the constrained Dantzig selector constraints, that is, $\{j \in \mathcal{A}^c : |n^{-1}\mathbf{x}_j^T(\mathbf{y} - \mathbf{X}\widehat{\beta}_{\mathcal{A}})| > \lambda_1\} = \emptyset$, the iteration stops and the algorithm converges. In other words, the convergence of the algorithm is equivalent to that of the active set which can be checked directly. When the algorithm converges, the solution lies in the feasible set of the constrained Dantzig selector problem and is a global minimizer restricted on the active set, and is thus a local minimizer.

In simulation Example 2 of Section 4.2, we tracked the convergence property of the algorithm on 100 data sets for $p = 1000$, 5000, and 10000, respectively. In all cases, we observe that the algorithm always converged over all 100 simulations, indicating considerable stability of this algorithm. Another advantage of the algorithm is that it is built upon the Dantzig selector in lower dimensions, so it inherits the computational efficiency.

4.2 Simulation studies

To better illustrate the performance of the constrained Dantzig selector, we consider the thresholded Dantzig selector which simply sets components of the Dantzig selector estimate to zeros if smaller than a threshold in magnitude. We evaluated the performance

of the constrained Dantzig selector in comparison with the Dantzig selector, thresholded Dantzig selector, Lasso, elastic net (Zou and Hastie, 2005), and adaptive Lasso (Zou, 2006). Two simulation studies were considered, with the first one investigating sparse recovery for compressed sensing and the second one examining sparse modeling.

The setting of the first simulation example is similar to that of the sparse recovery example in Lv and Fan (2009). The noiseless sparse recovery example is considered here since the Dantzig selector problem originated from compressed sensing. We want to evaluate the capability of our constrained Dantzig selector in recovering sparse signals as well. We generated 100 data sets from model (1) without noise, that is, the linear equation $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}_0$ with $(s, p) = (7, 1000)$. The nonzero components of $\boldsymbol{\beta}_0$ were set to be $(1, -0.5, 0.7, -1.2, -0.9, 0.3, 0.55)^T$ lying in the first seven components, and n were chosen to be even integers between 30 and 80. The rows of the design matrix \mathbf{X} were sampled as independent and identically distributed (i.i.d.) copies from $N(\mathbf{0}, \mathbf{I}_r)$, with \mathbf{I}_r a $p \times p$ matrix with diagonal elements being 1 and off-diagonal elements being r , and then each column was rescaled to have L_2 -norm $\sqrt{\pi}$. Three levels of population collinearity, $r = 0, 0.2$, and 0.5 , were considered. Let λ_0 and λ be in two small grids $\{0.001, 0.005, 0.01, 0.05, 0.1\}$ and $\{0.05, 0.1, 0.15, 0.2\}$, respectively. We chose two grids of values for them since in the literature of compressed sensing, it is desirable to have the true support included among a set of estimators. The value 0.2 was chosen because it is close to $\{(\log p)/n\}^{1/2}$, which is about $\{(\log 1000)/80\}^{1/2}$ in this example. Smaller values for λ and λ_0 are also included in the grid for conservativeness. We set the grid of values for λ_1 as described in Section 4.1. If any of the solutions in the path had exactly the same support as $\boldsymbol{\beta}_0$, it is counted as successful recovery. This criterion was applied to all other methods in this example for fair comparison.

Figure 1 presents the probabilities of exact recovery of sparse $\boldsymbol{\beta}_0$ based on 100 simulations by all methods. We see that all methods performed well in relatively large samples and had lower probability of successful sparse recovery when the sample size becomes smaller. The constrained Dantzig selector performed better than other methods over different sample sizes and three levels of population collinearity. In particular, the thresholded Dantzig selector performed similarly to the Dantzig selector, revealing that simple thresholding alone, instead of flexible constraints as in the constrained Dantzig selector, does not help much on signal recovery in this case.

The second simulation example adopts a similar setting to that in Zheng et al. (2014). We generated 100 data sets from the linear regression model (1) with Gaussian error $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^2 \mathbf{I}_n)$. The coefficient vector $\boldsymbol{\beta}_0 = (\mathbf{v}^T, \dots, \mathbf{v}^T, \mathbf{0}^T)^T$ with the pattern $\mathbf{v} = (\boldsymbol{\beta}_{\text{strong}}^T, \boldsymbol{\beta}_{\text{weak}}^T)^T$ repeated three times, where $\boldsymbol{\beta}_{\text{strong}} = (0.6, 0, 0, -0.6, 0, 0)^T$ and $\boldsymbol{\beta}_{\text{weak}} = (0.05, 0, 0, -0.05, 0, 0)^T$. The coefficient subvectors $\boldsymbol{\beta}_{\text{strong}}$ and $\boldsymbol{\beta}_{\text{weak}}$ stand for the strong signals and weak signals in $\boldsymbol{\beta}_0$, respectively. The sample size and noise level were chosen as $(n, \sigma) = (100, 0.4)$, while the dimensionality p was set to be 1000, 5000, and 10000. The rows of the $n \times p$ design matrix \mathbf{X} were sampled as i.i.d. copies from a multivariate normal distribution $N(\mathbf{0}, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma} = (0.5^{|i-j|})_{1 \leq i, j \leq p}$. We applied all methods as in simulation example 1 and set $\lambda_0 = 0.01$ and $\lambda = 0.2$ for our method. Similarly as before, the value of 0.2 was selected since it is close to $\{(\log 1000)/100\}^{1/2}$. The ideal procedure, which knows the true underlying sparse model in advance, was also used as a benchmark.

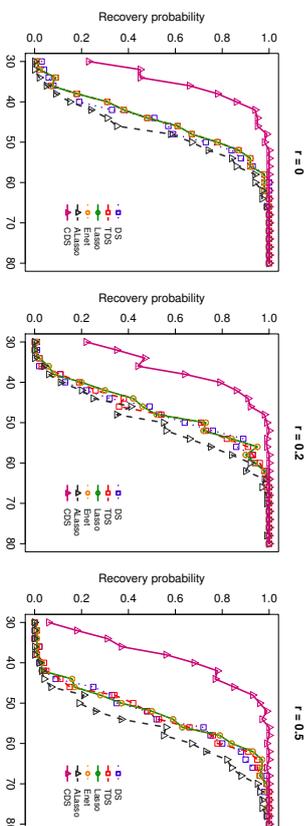


Figure 1: Probabilities of exact sparse recovery for the Dantzig selector (DS), thresholded Dantzig selector (TDS), Lasso, elastic net (Enet), adaptive Lasso (ALasso), and constrained Dantzig selector (CDS) in simulation example 1 of Section 4.2.

To compare these methods, we considered several performance measures: the prediction error, the L_q -estimation loss with $q = 1, 2, \infty$, number of false positives, and number of false negatives for strong or weak signals. The prediction error is defined as $E\|\mathbf{Y} - \mathbf{X}^T \hat{\boldsymbol{\beta}}\|^2$ with $\hat{\boldsymbol{\beta}}$ an estimate and (\mathbf{x}^T, Y) an independent observation, and the expectation was calculated using an independent test sample of size 10000. A false positive means a falsely selected noise covariate in the model, while a false negative means a missed true covariate. Table 1 summarizes the comparison results by all methods. We observe that most weak covariates were missed by all methods. This is reasonable since the weak signals are around the noise level, making it difficult to distinguish them from the noise covariates. However, the constrained Dantzig selector outperformed other methods in terms of other prediction and estimation measures, and followed very closely the ideal procedure in all cases of $p = 1000, 5000$, and 10000. In particular, the L_∞ -estimation loss for the constrained Dantzig selector was similar to that for the oracle procedure, confirming tight bounds on this loss in Theorems 1 and 2. As the dimensionality grows higher, the constrained Dantzig selector performed similarly, while other methods suffered from high dimensionality. In particular, the thresholded Dantzig selector has been shown to improve over the Dantzig selector, but was still outperformed by the adaptive Lasso and constrained Dantzig selector in this example, revealing the necessity to introduce more flexible constraints instead of simple thresholding.

Recall that we have fixed $\lambda_0 = 0.01$ and $\lambda = 0.2$ in the simulation example 2 across all settings. We now study the robustness of the constrained Dantzig with respect to λ_0 and λ in this typical example. For simplicity, we only consider Example 2 with dimensionality $p = 1000$. Instead of fixing $\lambda_0 = 0.01$ and $\lambda = 0.2$, we let λ_0 be a value in the grid $\{0.001, 0.005, 0.01, 0.015, 0.02, 0.025, 0.03\}$ and λ a value in $\{0.1, 0.15, 0.2, 0.25, 0.3, 0.35\}$, respectively. Therefore, we study $7 \times 6 = 42$ different combinations of λ_0 and λ in total to

Table 1: Means and standard errors (in parentheses) of different performance measures by all methods in simulation example 2 of Section 4.2.

Measure	DS	TDS	Lasso	Enet	ALasso	CDS	Oracle
$p = 1000$							
PE ($\times 10^{-2}$)	30.8 (0.6)	28.5 (0.5)	30.3 (0.5)	32.9 (0.7)	19.1 (0.1)	18.5 (0.1)	18.2 (0.1)
L_1 ($\times 10^{-2}$)	201.9 (5.4)	137.2 (3.4)	186.1 (5.0)	211.1 (6.0)	58.0 (1.1)	51.3 (0.6)	41.5 (0.9)
L_2 ($\times 10^{-2}$)	40.1 (0.7)	37.2 (0.7)	39.8 (0.7)	43.0 (0.8)	18.3 (0.3)	16.3 (0.2)	14.7 (0.3)
L_∞ ($\times 10^{-2}$)	19.0 (0.5)	18.6 (0.4)	19.3 (0.5)	20.7 (0.5)	9.1 (0.3)	7.5 (0.2)	8.4 (0.2)
FP	44.4 (1.8)	5.5 (3.8)	36.3 (1.6)	44.1 (1.9)	0.5 (0.1)	0 (0)	0 (0)
FN.strong	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
FN.weak	5.3 (0.1)	5.9 (0.4)	5.4 (0.1)	5.3 (0.1)	6.0 (0.0)	6.0 (0.0)	0 (0)
$p = 5000$							
PE ($\times 10^{-2}$)	45.1 (1.1)	39.3 (1.1)	44.8 (1.1)	44.9 (1.1)	21.3 (0.6)	18.4 (0.1)	18.3 (0.1)
L_1 ($\times 10^{-2}$)	289.3 (6.4)	184.6 (4.5)	270.8 (6.8)	273.2 (6.9)	71.2 (2.1)	50.4 (0.7)	41.7 (1.1)
L_2 ($\times 10^{-2}$)	56.3 (1.1)	50.6 (1.1)	56.1 (1.1)	56.2 (1.1)	22.9 (0.9)	16.0 (0.2)	14.9 (0.4)
L_∞ ($\times 10^{-2}$)	27.4 (0.7)	25.1 (0.6)	27.8 (0.7)	27.8 (0.7)	12.7 (0.7)	7.3 (0.2)	8.8 (0.3)
FP	60.6 (1.7)	7.0 (3.5)	53.5 (2.3)	53.8 (2.1)	1.0 (0.1)	0 (0)	0 (0)
FN.strong	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
FN.weak	5.2 (0.1)	6.0 (0.1)	5.4 (0.1)	5.3 (0.1)	5.9 (0.0)	6.0 (0.0)	0 (0)
$p = 10000$							
PE ($\times 10^{-2}$)	54.4 (16.2)	54.8 (17.4)	56.7 (23.1)	63.3 (21.7)	32.4 (32.3)	19.0 (0.5)	18.3 (0.1)
L_1 ($\times 10^{-2}$)	322.3 (67.9)	218 (60.3)	296.3 (72.7)	334.8 (79.9)	86.6 (67.6)	51.9 (1.6)	41.3 (0.9)
L_2 ($\times 10^{-2}$)	62.7 (14.1)	63.5 (14.9)	64.2 (17.7)	70.3 (16.8)	29.1 (26.3)	16.6 (0.6)	14.7 (0.3)
L_∞ ($\times 10^{-2}$)	30.5 (8.2)	32.5 (8.5)	31.8 (9.5)	34.4 (9.3)	15.8 (13.2)	7.8 (0.6)	8.5 (0.2)
FP	69.9 (14.3)	6.9 (4.6)	56.89 (20.66)	64.06 (18.64)	0.78 (1.25)	0.01 (0.10)	0 (0)
FN.strong	0 (0.1)	0 (0.1)	0.02 (0.14)	0.02 (0.14)	0.21 (0.95)	0.01 (0.10)	0 (0)
FN.weak	6 (0.2)	6 (0.1)	5.98 (0.14)	5.97 (0.17)	6 (0)	6 (0)	0 (0)

evaluate the robustness. The same performance measures were calculated. Here we only present the prediction error results to save space, and the other results are available upon request. The tuning parameter λ_1 was chosen by cross-validation, similar as in Example 2. We can see from Table 2 that for all $\lambda \geq 0.15$ and all λ_0 in the grids, the means and standard errors of the prediction error are very close or even identical to those for $\lambda_0 = 0.01$ and $\lambda = 0.2$. The prediction error in the case of $\lambda = 0.1$ is slightly higher since when the threshold becomes lower, some noise variables can be included. For the results on estimation losses as well as false positives and false negatives, we observe similar patterns confirming the robustness of our method with respect to the choices of λ_0 and λ .

4.3 Real data analyses

We applied the same methods as in Section 4.2 to two real data sets: one real PCR data set and another gene expression data set, both in the high-dimensional setting with relatively small sample size. In both data sets, we found that the proposed method enjoys smaller prediction errors and the differences are statistically significant.

Table 2: Means and standard errors (in parentheses) of prediction error of the constrained Dantzig selector for different choices of λ_0 and λ in simulation example 2 with $p = 1000$.

λ_0	λ					
	0.10	0.15	0.20	0.25	0.30	0.35
0.001	0.192 (0.002)	0.186 (0.001)	0.185 (0.001)	0.185 (0.001)	0.185 (0.001)	0.185 (0.001)
0.005	0.191 (0.003)	0.188 (0.001)	0.185 (0.001)	0.185 (0.001)	0.185 (0.001)	0.185 (0.001)
0.010	0.192 (0.002)	0.187 (0.001)	0.185 (0.001)	0.185 (0.001)	0.188 (0.003)	0.185 (0.001)
0.015	0.196 (0.007)	0.186 (0.001)	0.185 (0.001)	0.185 (0.001)	0.185 (0.001)	0.185 (0.001)
0.020	0.192 (0.002)	0.188 (0.001)	0.185 (0.001)	0.185 (0.001)	0.185 (0.001)	0.185 (0.001)
0.025	0.190 (0.002)	0.187 (0.001)	0.185 (0.001)	0.185 (0.001)	0.185 (0.001)	0.185 (0.001)
0.030	0.191 (0.002)	0.187 (0.001)	0.191 (0.007)	0.188 (0.003)	0.185 (0.001)	0.185 (0.001)

The real PCR data set, originally studied in Lan et al. (2006), examines the genetics of two inbred mouse populations. This data set is comprised of $n = 60$ samples with 29 males and 31 females. Expression levels of 22,575 genes were measured. Following Song and Liang (2015), we study the linear relationship between the numbers of Phosphoenolpyruvate carboxykinase (PEPCK), a phenotype measured by quantitative real-time PCR, and the gene expression levels. Both the phenotype and predictors are continuous. As suggested in Song and Liang (2015), we only picked $p = 2000$ genes having the highest marginal correlations with PEPCK as predictors. The response was standardized to have zero mean and unit variance before we conducted the analysis. The 2000 predictors were standardized to have zero mean and L_2 -norm \sqrt{n} in each column. Then the data set was randomly split into a training set of 55 samples and a test set with the remaining 5 samples for 100 times. We set $\lambda_0 = 0.001$ and $\lambda = 0.02$ in this real data analysis as well as the other one below for conservativeness. Methods under comparison are the same as in the simulation studies.

Table 3 reports the means and standard errors of the prediction error on the test data as well as the median model size for each method. We see that the method CDS gave the lowest mean prediction error. Paired t -tests were conducted for the prediction errors of CDS versus DS, TDS, Lasso, and Enet, respectively, to test the differences in performance across various methods. The corresponding p -values were 0.0243, 0.0014, 0.0081, 0.0001, and 0.0102, respectively, indicating significantly different prediction error from that for CDS.

The second data set has been studied in Scheetz et al. (2006) and Huang et al. (2008). In this data set, 120 twelve-week-old male rats were selected for tissue harvesting from the eyes and for microarray analysis. There are 31,042 different probe sets in the microarrays from the RNA of those rat eyes. Following Huang et al. (2008), we excluded the probes that were not expressed sufficiently or that lacked sufficient variation, leaving 18,976 probes which satisfy these two criteria. The response variable TRIM32, which was recently found to cause Bardet-Biedl syndrome, is one of the selected 18,976 probes. We then selected 3,000 probes with the largest variances from the remaining 18,975 probes. The goal of our analysis is to identify the genes that are most relevant to the expression level of TRIM32 from the 3,000 candidate genes.

Table 3: Means and standard errors of the prediction error and median model size over 100 random splits for the real PCR data set.

Method	PE	Model Size
DS	0.773 (0.040)	54
TDS	0.897 (0.063)	14
Lasso	0.802 (0.043)	58
ALasso	0.922 (0.056)	5
Enet	0.793 (0.041)	58
CDS	0.660 (0.036)	21

Similarly as in the analysis of the first real data set, we standardized the response and predictors beforehand. The training set contains 100 samples and was sampled randomly 100 times from the full data set. The remaining 20 samples at each time served as the test set. Results of prediction errors and median model sizes are presented in Table 4. It is clear that the proposed method CDS enjoys the lowest prediction error with a small model size. We conducted the same paired t -tests as in the real PCR data set for comparison. The corresponding p -values were 0.0351, 0.0058, 0.0164, 0.0004, and 0.0344, respectively, showing significant improvement.

Table 4: Means and standard errors of the prediction error and median model size over 100 random splits for the gene expression data set of rat eyes.

Method	PE	Model Size
DS	0.582 (0.044)	28.5
TDS	0.627 (0.051)	9
Lasso	0.590 (0.043)	33
ALasso	0.652 (0.051)	8.5
Enet	0.576 (0.040)	67.5
CDS	0.520 (0.025)	9

5. Discussion

We have shown that the suggested constrained Dantzig selector can achieve convergence rates within a logarithmic factor of the sample size of the oracle rates in ultra-high dimensions under a fairly weak assumption on the signal strength. Our work provides a partial answer to an interesting question of whether convergence rates involving a logarithmic factor of the dimensionality are optimal for regularization methods in ultra-high dimensions. It would be interesting to investigate such a phenomenon for more general regularization methods.

Our formulation of the constrained Dantzig selector uses the L_1 -norm of the parameter vector. A natural extension of the method is to exploit the weighted L_1 -norm of the

parameter to allow for different regularization on different covariates, as is in the adaptive Lasso (Zou, 2006). It would be interesting to investigate the behavior of these methods in more general model settings including generalized linear models and survival analysis. These problems are beyond the scope of the current paper and will be interesting topics for future research.

Appendix: Proofs of main results

Proof of Theorem 1

High probability event \mathcal{E} . Recall that $\lambda_0 = c_0 \sqrt{(\log n)/n}$ and $\lambda_1 = c_1 \sqrt{(\log p)/n}$. All the results in Theorems 1 and 2 will be shown to hold on a key event

$$\mathcal{E} = \{\|n^{-1}\mathbf{X}_1^T \boldsymbol{\epsilon}\|_\infty \leq \lambda_0 \text{ and } \|n^{-1}\mathbf{X}_2^T \boldsymbol{\epsilon}\|_\infty \leq \lambda_1\}, \quad (7)$$

where \mathbf{X}_1 is a submatrix of \mathbf{X} consisting of columns corresponding to $\text{supp}(\beta_0)$ and \mathbf{X}_2 consists of the remaining columns. Thus we will have the same probability bound in both theorems. The probability bound on the event \mathcal{E} in (7) can be easily calculated, using the classical Gaussian tail probability bound (see, for example, (Dudley, 1999)) and the Bonferroni inequality, as

$$\begin{aligned} \text{pr}(\mathcal{E}) &\geq 1 - \{\text{pr}(\|n^{-1}\mathbf{X}_1^T \boldsymbol{\epsilon}\|_\infty > \lambda_0) + \text{pr}(\|n^{-1}\mathbf{X}_2^T \boldsymbol{\epsilon}\|_\infty > \lambda_1)\} \\ &= 1 - \{s(2/\pi)^{1/2} \sigma \lambda_0^{-1} n^{-1/2} e^{-\lambda_0^2 n / (2\sigma^2)} + (p-s)(2/\pi)^{1/2} c \lambda_1^{-1} n^{-1/2} e^{-\lambda_1^2 n / (2c^2 \sigma^2)} - 1\} \\ &= 1 - O\{s n^{-c_0^2 / (2\sigma^2)} (\log n)^{-1/2} + (p-s) p^{-c_1^2 / (2\sigma^2)} (\log p)^{-1/2}\}, \end{aligned} \quad (8)$$

where the last equality follows from the definitions of λ_0 and λ_1 . Let $c = (c_0^2 \wedge c_1^2) / (2\sigma^2) - 1$ be a sufficiently large positive constant, since the two positive constants c_0 and c_1 are chosen large enough. Recall that p is understood implicitly as $\max\{n, p\}$ throughout the paper. Thus it follows from (8), $s \leq n$, and $n \leq p$ that

$$\text{pr}(\mathcal{E}) = 1 - O(n^{-c}). \quad (9)$$

From now on, we derive all the bounds on the event \mathcal{E} . In particular, in light of (7) and $\beta_0 \in \mathcal{B}_\lambda$, it is easy to verify that conditional on \mathcal{E} , the true regression coefficient vector β_0 satisfies the constrained Dantzig selector constraints; in other words, β_0 lies in the feasible set in (3).

Nonasymptotic properties of $\hat{\beta}$. We first make a simple observation on the constrained Dantzig selector $\hat{\beta} = (\hat{\beta}_1, \dots, \hat{\beta}_p)^T$. Recall that without loss of generality, we assume $\text{supp}(\beta_0) = \{1, \dots, s\}$. Let $\beta_0 = (\beta_1^T, \mathbf{0}^T)^T$ with each component of β_1 being nonzero, and $\hat{\beta} = (\hat{\beta}_1^T, \hat{\beta}_2^T)^T$ with $\hat{\beta}_1$ a subvector of $\hat{\beta}$ consisting of its first s components. Denote by $\boldsymbol{\delta} = (\boldsymbol{\delta}_1^T, \boldsymbol{\delta}_2^T)^T = \hat{\beta} - \beta_0$ the estimation error, where $\boldsymbol{\delta}_1 = \hat{\beta}_1 - \beta_1$ and $\boldsymbol{\delta}_2 = \hat{\beta}_2$. It follows from the global optimality of $\hat{\beta}$ that $\|\beta_1\|_1 + \|\beta_2\|_1 = \|\hat{\beta}\|_1 \leq \|\beta_0\|_1 = \|\beta_1\|_1$, which entails that

$$\|\boldsymbol{\delta}_2\|_1 = \|\hat{\beta}_2\|_1 \leq \|\beta_1\|_1 - \|\hat{\beta}_1\|_1 \leq \|\hat{\beta}_1 - \beta_1\|_1 = \|\boldsymbol{\delta}_1\|_1. \quad (10)$$

We will see that this basic inequality $\|\delta_2\|_1 \leq \|\delta_1\|_1$ plays a key role in the technical derivations of both Theorem 1 and 2. Equipped with this inequality and conditional on \mathcal{E} , we are now able to start the derivation of all results in Theorem 1 as follows.

The main idea is to first prove its sparsity property which will be presented in the next paragraph. Then, with the control on the number of false positives and false negatives, we derive an upper bound for the L_2 -estimation loss using the conclusion in Lemma 3.1 of Candès and Tao (2007). Results on other types of losses follow accordingly.

(1) Sparsity. Recall that under the assumption of Theorem 1, β_0 lies in its feasible set conditional on \mathcal{E} . Since $\lambda_0 \leq \lambda_1$, by the definition of the constrained Dantzig selector, we have $\|n^{-1}\mathbf{X}^T(\mathbf{y} - \mathbf{X}\hat{\beta})\| \leq \lambda_1$, where \leq is understood as componentwise no larger than. Conditional on the event \mathcal{E} , substituting \mathbf{y} by $\mathbf{X}\beta_0 + \varepsilon$ and applying the triangle inequality yields

$$\|n^{-1}\mathbf{X}^T\mathbf{X}\delta\|_\infty \leq 2\lambda_1. \quad (11)$$

Furthermore, Lemma 3.1 in Candès and Tao (2007) still applies for δ as long as the uniform uncertainty principle condition (4) holds. Together with (10) and (11), by applying the same argument as in the proof of Theorem 1.1 in Candès and Tao (2007), we obtain an L_2 -estimation loss bound of $\|\delta\|_2 \leq (Cs)^{1/2}\lambda_1$ with $C = 4^2/(1 - \delta - \theta)^2$ some positive constant.

Since both the true regression coefficient vector β_0 and the constrained Dantzig selector $\hat{\beta}$ lie in the constrained parameter space \mathcal{B}_λ , the magnitude of any nonzero component in both β_0 and $\hat{\beta}$ is no smaller than λ . It follows that on the set of falsely discovered signs, the component of $\delta = \hat{\beta} - \beta_0$ will be no smaller than λ . Then making use of the obtained L_2 -estimation loss bound $\|\delta\|_2 \leq (Cs)^{1/2}\lambda_1$, it is immediate that the number of falsely discovered signs is bounded from above by $Cs(\lambda_1/\lambda)^2$. Under the assumption of $\lambda \geq C^{1/2}(1 + \lambda_1/\lambda_0)\lambda_1$, we have $Cs(\lambda_1/\lambda)^2 \leq s(1 + \lambda_1/\lambda_0)^{-2} \leq s$.

(2) L_2 -estimation loss. We further exploit the technical tool of Lemma 3.1 in Candès and Tao (2007) to analyze the behavior of the estimation error $\delta = \hat{\beta} - \beta_0$. Let δ'_1 be a subvector of δ_2 consisting of the s largest components in magnitude, $\delta_3 = (\delta'_1, (\delta'_1)^T)^T$, and \mathbf{X}_3 a submatrix of \mathbf{X} consisting of columns corresponding to δ_3 . We emphasize that δ_3 covers all nonzero components of δ since the number of falsely discovered signs is upper bounded by s , as showed in the previous paragraph. Therefore, $\|\delta_3\|_q = \|\delta\|_q$ for all $q > 0$. In view of the uniform uncertainty principle condition (4), an application of Lemma 3.1 in Candès and Tao (2007) results in

$$\|\delta_3\|_2 \leq (1 - \delta)^{-1} \|n^{-1}\mathbf{X}_3^T\mathbf{X}\delta\|_2 + \theta(1 - \delta)^{-1} s^{-1/2} \|\delta_2\|_1. \quad (12)$$

On the other hand, from the basic inequality (10) it is easy to see that $s^{-1/2} \|\delta_2\|_1 \leq s^{-1/2} \|\delta_1\|_1 \leq \|\delta_3\|_2$. Substituting it into (12) leads to

$$\|\delta\|_2 = \|\delta_3\|_2 \leq (1 - \delta - \theta)^{-1} \|n^{-1}\mathbf{X}_3^T\mathbf{X}\delta\|_2 \quad (13)$$

Hence, to develop a bound for the L_2 -estimation loss it suffices to find an upper bound for $\|n^{-1}\mathbf{X}_3^T\mathbf{X}\delta\|_2$.

Denote by A_1 , A_2 , and A_3 the index sets of correctly selected variables, missed true variables, and falsely selected variables, respectively. Let $A_{23} = A_2 \cup A_3$. Then we can

obtain from the definition of the constrained Dantzig selector along with its thresholding feature that $\|n^{-1}\mathbf{X}_{A_1}^T(\mathbf{y} - \mathbf{X}\hat{\beta})\| \leq \lambda_0$, $\|n^{-1}\mathbf{X}_{A_2}^T(\mathbf{y} - \mathbf{X}\hat{\beta})\| \leq \lambda_1$, and $\|n^{-1}\mathbf{X}_{A_3}^T(\mathbf{y} - \mathbf{X}\hat{\beta})\| \leq \lambda_0$. Conditional on \mathcal{E} , substituting \mathbf{y} by $\mathbf{X}\beta_0 + \varepsilon$ and applying the triangle inequality give

$$\|n^{-1}\mathbf{X}_{A_1}^T\mathbf{X}\delta\|_\infty \leq 2\lambda_0 \quad \text{and} \quad \|n^{-1}\mathbf{X}_{A_{23}}^T\mathbf{X}\delta\|_\infty \leq \lambda_0 + \lambda_1. \quad (14)$$

We now make use of the technical result on sparsity. Since A_{23} denotes the index set of false positives and false negatives, its cardinality is also bounded by $Cs(\lambda_1/\lambda)^2$ with probability at least $1 - O(n^{-c})$. Therefore, by (14) we have

$$\begin{aligned} \|n^{-1}\mathbf{X}_3^T\mathbf{X}\delta\|_2^2 &\leq \|n^{-1}\mathbf{X}_{A_1}^T\mathbf{X}\delta\|_2^2 + \|n^{-1}\mathbf{X}_{A_{23}}^T\mathbf{X}\delta\|_2^2 \\ &\leq s\|n^{-1}\mathbf{X}_{A_1}^T\mathbf{X}\delta\|_\infty^2 + Cs(\lambda_1/\lambda)^2\|n^{-1}\mathbf{X}_{A_{23}}^T\mathbf{X}\delta\|_\infty^2 \\ &\leq 4s\lambda_0^2 + Cs(\lambda_1/\lambda)^2(\lambda_0 + \lambda_1)^2. \end{aligned}$$

Substituting this inequality into (13) yields

$$\|\delta\|_2 \leq (1 - \delta - \theta)^{-1} \{4s\lambda_0^2 + Cs(\lambda_1/\lambda)^2(\lambda_0 + \lambda_1)^2\}^{1/2}.$$

Since we assume $\lambda \geq C^{1/2}(1 + \lambda_1/\lambda_0)\lambda_1$, it follows that $C(\lambda_1/\lambda)^2(\lambda_0 + \lambda_1)^2 \leq \lambda_0^2$. Therefore, we conclude that $\|\delta\|_2 \leq (1 - \delta - \theta)^{-1}(5s\lambda_0^2)^{1/2}$.

(3) Other losses. Applying the basic inequality (10), we establish an upper bound for the L_1 -estimation loss

$$\begin{aligned} \|\delta\|_1 &= \|\delta_1\|_1 + \|\delta_2\|_1 \leq 2\|\delta_1\|_1 \leq 2s^{1/2}\|\delta_1\|_2 \\ &\leq 2s^{1/2}\|\delta\|_2 \leq 2(1 - \delta - \theta)^{-1} s(5\lambda_0^2)^{1/2}. \end{aligned}$$

For the L_∞ -estimation loss, we additionally assume that $\lambda > (1 - \delta - \theta)^{-1}(5s\lambda_0^2)$ which can lead to the sign consistency, $\text{sgn}(\hat{\beta}) = \text{sgn}(\beta_0)$, in view of the L_2 -estimation loss inequality above. Therefore, by the constrained Dantzig selector constraints we have $\|n^{-1}\mathbf{X}_1^T(\mathbf{y} - \mathbf{X}_1\hat{\beta}_1)\|_\infty \leq \lambda_0$ and thus $\|n^{-1}\mathbf{X}_1^T(\varepsilon - \mathbf{X}_1\delta_1)\|_\infty \leq \lambda_0$. Then conditional on \mathcal{E} , it follows from the triangle inequality that $\|n^{-1}\mathbf{X}_1^T\mathbf{X}_1\delta_1\|_\infty \leq 2\lambda_0$. Hence, $\|\delta\|_\infty = \|\delta_1\|_\infty \leq 2\|(n^{-1}\mathbf{X}_1^T\mathbf{X}_1)^{-1}\|_\infty\lambda_0$, which completes the proof.

Proof of Theorem 2

We continue to use the technical setup and notation introduced in the proof of Theorem 1. Results are parallel to those in Theorem 1 but presented in the asymptotic manner. The similarity lies in the rationale of the proof as well. The key element is to derive the sparsity and then construct an inequality for L_2 -estimation loss through the bridge $n^{-1}\|\mathbf{X}\delta\|_2^2$. Inequalities for other types of losses are built upon this bound.

(1) Sparsity. Conditional on the event \mathcal{E} , we know that (10) and (11) still hold by the definition of the constrained Dantzig selector. Moreover, Condition I is similar to the restricted eigenvalue assumption $\text{RE}(s, m, 1)$ in Bickel et al. (2009), except that $\text{RE}(s, m, 1)$ assumes the inequality holds for any subset with size no larger than s to cover different possibilities of $\text{supp}(\beta_0)$. Since we assume without loss of generality that $\text{supp}(\beta_0) =$

$\{1, \dots, s\}$, an application of similar arguments as in the proof of Theorem 7.1 in Bickel et al. (2009) yields the L_2 oracle inequality $\|\delta\|_2 \leq (C_m s)^{1/2} \lambda_1$ with C_m some positive constant dependent on m . Therefore, by the same arguments as in the proof of Theorem 1, it can be shown that the number of falsely discovered signs is bounded from above by $C_m s(\lambda_1/\lambda)^2$ and further by $s(1 + \lambda_1/\lambda_0)^{-2}$ since $\lambda \geq C_m^{1/2}(1 + \lambda_1/\lambda_0)\lambda_1$. Next we will go through the proof of Theorem 7.1 in Bickel et al. (2009) in a more cautious manner and make some improvements in some steps with the aid of the obtained bound on the number of false positives and false negatives.

(2) L_2 -estimation loss. By Condition 1, we have a lower bound for $n^{-1}\|\mathbf{X}\delta\|_2^2$. It is also natural to derive an upper bound for it and build an inequality related to the L_2 -estimation loss. It follows from (14) that

$$\begin{aligned} n^{-1}\|\mathbf{X}\delta\|_2^2 &\leq \|n^{-1}\mathbf{X}_A^T \mathbf{X} \delta\|_\infty \|\delta_{A_1}\|_1 + \|n^{-1}\mathbf{X}_{A_{23}}^T \mathbf{X} \delta\|_\infty \|\delta_{A_{23}}\|_1 \\ &\leq 2\lambda_0 \|\delta_{A_1}\|_1 + (\lambda_0 + \lambda_1) \|\delta_{A_{23}}\|_1. \end{aligned} \quad (15)$$

Since the cardinality of A_{23} is bounded by $C_m s(\lambda_1/\lambda)^2$, applying the Cauchy-Schwarz inequality to (15) leads to $n^{-1}\|\mathbf{X}\delta\|_2^2 \leq 2\lambda_0 \|\delta_{A_1}\|_1 + (\lambda_0 + \lambda_1) \|\delta_{A_{23}}\|_1 \leq 2\lambda_0 s^{1/2} \|\delta_{A_1}\|_2 + C_m^{1/2} (\lambda_0 + \lambda_1) s^{1/2} \lambda_1 / \lambda \|\delta_{A_{23}}\|_2$. This gives an upper bound for $n^{-1}\|\mathbf{X}\delta\|_2^2$. Combining this with Condition 1 results in

$$2^{-1} \kappa^{-2} (\|\delta_{A_1}\|_2^2 + \|\delta_{A_{23}}\|_2^2) \leq 2\lambda_0 s^{1/2} \|\delta_{A_1}\|_2 + C_m^{1/2} (\lambda_0 + \lambda_1) s^{1/2} \lambda_1 / \lambda \|\delta_{A_{23}}\|_2. \quad (16)$$

Consider (16) in a two-dimensional space with respect to $\|\delta_{A_1}\|_2$ and $\|\delta_{A_{23}}\|_2$. Then the quadratic inequality (16) defines a circular area centered at $(2\kappa^{-2}\lambda_0 s^{1/2}, \kappa^{-2} C_m^{1/2} (\lambda_0 + \lambda_1) s^{1/2} \lambda_1 / \lambda)$. The term $\|\delta_{A_1}\|_2^2 + \|\delta_{A_{23}}\|_2^2$ is nothing but the squared distance between the point in this circular area and the origin. One can easily identify the largest squared distance which is also the upper bound for the L_2 -estimation loss

$$\|\delta\|_2^2 = \|\delta_{A_1}\|_2^2 + \|\delta_{A_{23}}\|_2^2 \leq 4\kappa^{-4} \left\{ (2\lambda_0 s^{1/2})^2 + [C_m^{1/2} (\lambda_0 + \lambda_1) s^{1/2} \lambda_1 / \lambda]^2 \right\}.$$

With the assumption of $\lambda \geq C_m^{1/2} (1 + \lambda_1/\lambda_0) \lambda_1$, we can show that $\{C_m^{1/2} (\lambda_0 + \lambda_1) s^{1/2} \lambda_1 / \lambda\}^2 \leq s\lambda_0^2$ and thus $\|\delta\|_2 = O(\kappa^{-2} s^{1/2} \lambda_0)$. This bound has significant improvement with the factor $\log p$ reduced to $\log n$ in the ultra-high dimensional setting.

(3) Other losses. With the L_2 oracle inequality at hand, one can derive the L_1 oracle inequality $\|\delta\|_1 \leq s^{1/2} \|\delta\|_2 = O(\kappa^{-2} s \lambda_0)$ in a straightforward manner. For the prediction loss, it follows from (15) that

$$n^{-1}\|\mathbf{X}\delta\|_2^2 \leq 2\lambda_0 s^{1/2} \|\delta_{A_1}\|_2 + C_m^{1/2} (\lambda_0 + \lambda_1) s^{1/2} \lambda_1 / \lambda \|\delta_{A_{23}}\|_2.$$

Consider the problem of minimizing $2\lambda_0 s^{1/2} \|\delta_{A_1}\|_2 + C_m^{1/2} (\lambda_0 + \lambda_1) s^{1/2} \lambda_1 / \lambda \|\delta_{A_{23}}\|_2$ subject to (16). It is simply a two-dimensional linear optimization problem with a circular area as its feasible set. One can easily solve this minimization problem and obtain

$$n^{-1}\|\mathbf{X}\delta\|_2^2 \leq 2\kappa^{-2} \left[(2\lambda_0 s^{1/2})^2 + \{C_m^{1/2} (\lambda_0 + \lambda_1) s^{1/2} \lambda_1 / \lambda\}^2 \right] = O(\kappa^{-2} s \lambda_0^2).$$

Finally, for the L_∞ oracle inequality, it follows from similar arguments as in the proof of Theorem 1 that $\|\delta\|_\infty \leq 2\| (n^{-1} \mathbf{X}_1^T \mathbf{X}_1)^{-1} \|_\infty \lambda_0 = O\{ \| (n^{-1} \mathbf{X}_1^T \mathbf{X}_1)^{-1} \|_\infty \lambda_0 \}$, which concludes the proof.

Proof of Theorem 3

Denote by $\widehat{\beta}_{\text{global}}$ the global minimizer of (3). Under the conditions of Theorem 2, $\widehat{\beta}_{\text{global}}$ enjoys the same oracle inequalities and properties as in Theorem 2 conditional on the event defined in (7). In particular, we have $F_S(\widehat{\beta}_{\text{global}}) = O(s)$. It follows that

$$\|\widehat{\beta}_{\text{global}}\|_0 \leq \|\widehat{\beta}_0\|_0 + F_S(\widehat{\beta}_{\text{global}}) = O(s).$$

Let $\widehat{\beta}$ be a computable local minimizer of (3) produced by any algorithm satisfying $\|\widehat{\beta}\|_0 \leq c_3 s$. Denote by $A = \text{supp}(\widehat{\beta}) \cup \text{supp}(\widehat{\beta}_{\text{global}})$. Then we have

$$|A| \leq \|\widehat{\beta}\|_0 + \|\widehat{\beta}_{\text{global}}\|_0 = O(s) \leq c_3 s$$

for some large enough positive constant c_3 .

We next analyze the difference between the two estimators, that is, $\delta = \widehat{\beta} - \widehat{\beta}_{\text{global}}$. Let \mathbf{X}_A be a submatrix of \mathbf{X} consisting of columns in A and δ_A a subvector of δ consisting of components in A . Since $\|\mathbf{X}^T(\mathbf{y} - \mathbf{X}\widehat{\beta})\|_\infty = O(\lambda_1)$ and $\|\mathbf{X}^T(\mathbf{y} - \mathbf{X}\widehat{\beta}_{\text{global}})\|_\infty = O(\lambda_1)$, we can show that

$$\|n^{-1} \mathbf{X}_A^T \mathbf{X}_A \delta_A\|_2 \leq |A|^{1/2} \|n^{-1} \mathbf{X}_A^T \mathbf{X}_A \delta_A\|_\infty = O(s^{1/2} \lambda_1).$$

By the assumption that $\min_{\|\delta\|_0=1, \|\delta\|_{0 \leq c_3 s}} n^{-1/2} \|\mathbf{X}\delta\|_2 \geq \kappa_0$, the smallest singular value of $n^{-1/2} \mathbf{X}_A$ is bounded from below by κ_0 . Thus we have $\|\delta\|_2 = \|\delta_A\|_2 = O(s^{1/2} \lambda_1)$. Together with the thresholding feature of the constrained Dantzig selector, it follows that the number of different indices between $\text{supp}(\widehat{\beta})$ and $\text{supp}(\widehat{\beta}_{\text{global}})$ is bounded by $O\{s(\lambda_1/\lambda)^2\}$. This sparsity property is essential for our proof and similar sparsity results can be found in Theorem 2.

Based on the aforementioned sparsity property and by the facts that $\|\mathbf{X}_{\widehat{A}_1}^T \mathbf{X}\delta\|_\infty \leq 2\lambda_0$ and $\|\mathbf{X}_{\widehat{A}_{23}}^T \mathbf{X}\delta\|_\infty \leq \lambda_0 + \lambda_1$ with $\widehat{A}_1 = \text{supp}(\widehat{\beta}) \cap \text{supp}(\widehat{\beta}_{\text{global}})$ and $\widehat{A}_{23} = [\text{supp}(\widehat{\beta}) \setminus \text{supp}(\widehat{\beta}_{\text{global}})] \cup [\text{supp}(\widehat{\beta}_{\text{global}}) \setminus \text{supp}(\widehat{\beta})]$, the same arguments as in the proof of Theorem 2 apply to show that $\|\delta\|_2 = O(\kappa^{-2} s^{1/2} \lambda_0)$. It is clear that this bound is of the same order as that for the difference between $\widehat{\beta}_{\text{global}}$ and β_0 in Theorem 2. Thus $\widehat{\beta}$ enjoys the same asymptotic bound on the L_2 -estimation loss. Similarly, the asymptotic bounds for the other losses in Theorem 2 also apply to $\widehat{\beta}$, since those inequalities are rooted on the bounds for the sparsity and L_2 -estimation loss. This completes the proof.

References

A. Antoniadis, P. Fyzlewicz, and F. Letué. The dantzig selector in cox's proportional hazards model. *Scandinavian Journal of Statistics*, 37(4):531–552, 2010.

- P. J. Bickel, Y. Ritov, and A. B. Tsybakov. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, pages 1705–1732, 2009.
- E. J. Candès and T. Tao. The dantzig selector: Statistical estimation when p is much larger than n . *The Annals of Statistics*, pages 2313–2351, 2007.
- E. J. Candès, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier analysis and applications*, 14(5-6):877–905, 2008.
- K. Chen, H. Dong, and K.-S. Chan. Reduced rank regression via adaptive nuclear norm penalization. *Biometrika*, pages 1–20, 2013.
- R. M. Dudley. *Uniform central limit theorems*, volume 23. Cambridge Univ Press, 1999.
- J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.
- J. Fan and J. Lv. Nonconcave penalized likelihood with np -dimensionality. *IEEE Transactions on Information Theory*, 57(8):5467–5484, 2011.
- Y. Fan and J. Lv. Asymptotic equivalence of regularization methods in thresholded parameter space. *Journal of the American Statistical Association*, 108(503):1044–1061, 2013.
- Y. Fan and C.-Y. Tang. Tuning parameter selection in high dimensional penalized likelihood. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(3):531–552, 2013.
- J. Huang, S. Ma, and C.-H. Zhang. Adaptive lasso for sparse high-dimensional regression models. *Statistica Sinica*, pages 1603–1618, 2008.
- H. Lan, M. Chen, et al. Combined expression trait correlations and expression quantitative trait locus mapping. *PLoS Genet*, 2(1):e6, 2006.
- J. Lv and Y. Fan. A unified approach to model selection and sparse recovery using regularized least squares. *The Annals of Statistics*, pages 3498–3528, 2009.
- S. N. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu. A unified framework for high-dimensional analysis of m -estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557, 2012.
- G. Raskutti, M. J. Wainwright, and B. Yu. Minimax rates of convergence for high-dimensional regression under ℓ_q -ball sparsity. *IEEE transactions on information theory*, 57(10):6976–6994, 2011.
- T. E. Scheetz, K.-Y. Kim, et al. Regulation of gene expression in the mammalian eye and its relevance to eye disease. *Proceedings of the National Academy of Sciences*, 103(39):14429–14434, 2006.
- Q. Song and F. Liang. High-dimensional variable selection with reciprocal ℓ_1 -regularization. *Journal of the American Statistical Association*, 110(512):1607–1620, 2015.
- R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, pages 267–288, 1996.
- S. van de Geer. High-dimensional generalized linear models and the lasso. *The Annals of Statistics*, pages 614–645, 2008.
- S. van de Geer, P. Bühlmann, S. Zhou, et al. The adaptive and the thresholded lasso for potentially misspecified models (and a lower bound for the lasso). *Electronic Journal of Statistics*, 5:688–749, 2011.
- M. J. Wainwright. Sharp thresholds for noisy and high-dimensional recovery of sparsity using ℓ_1 -constrained quadratic programming (lasso). *IEEE Transactions on Information Theory*, 55(5):2183–2202, 2009.
- C.-H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of statistics*, pages 894–942, 2010.
- T. Zhang. Adaptive forward-backward greedy algorithm for learning sparse representations. *IEEE transactions on information theory*, 57(7):4689–4708, 2011.
- P. Zhao and B. Yu. On model selection consistency of lasso. *Journal of Machine Learning Research*, 7(Nov):2541–2563, 2006.
- Z. Zheng, Y. Fan, and J. Lv. High dimensional thresholded regression and shrinkage effect. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(3):627–649, 2014.
- H. Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.
- H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

Bootstrap-Based Regularization for Low-Rank Matrix Estimation

Julie Josse

Department of Applied Mathematics, Agrocampus Ouest, Rennes, France,
INRIA Saclay, Université Paris-Sud, Orsay, France.

josse@agrocampus-ouest.fr

Stefan Wager

Department of Statistics, Stanford University, Stanford, U.S.A.

swager@stanford.edu

Abstract

We develop a flexible framework for low-rank matrix estimation that allows us to transform noise models into regularization schemes via a simple bootstrap algorithm. Effectively, our procedure seeks an autoencoding basis for the observed matrix that is stable with respect to the specified noise model; we call the resulting procedure a *stable autoencoder*. In the simplest case, with an isotropic noise model, our method is equivalent to a classical singular value shrinkage estimator. For non-isotropic noise models—e.g., Poisson noise—the method does not reduce to singular value shrinkage, and instead yields new estimators that perform well in experiments. Moreover, by iterating our stable autoencoding scheme, we can automatically generate low-rank estimates without specifying the target rank as a tuning parameter.

KEYWORDS: Correspondence analysis, empirical Bayes, Lévy bootstrap, singular-value decomposition.

1. Introduction

Low-rank matrix estimation plays a key role in many scientific and engineering tasks, including collaborative filtering (Koren et al., 2009), genome-wide studies (Leek and Storey, 2007; Price et al., 2006), and magnetic resonance imaging (Candès et al., 2013; Lustig et al., 2008). Low-rank procedures are often motivated by the following statistical model. Suppose that we observe a noisy matrix $X \in \mathbb{R}^{n \times p}$ drawn from some distribution $\mathcal{L}(\mu)$ with $\mathbb{E}_\mu[X] = \mu$, and that we have scientific reason to believe that μ admits a parsimonious, low-rank representation. Then, we can frame our statistical goal as trying to recover the underlying μ from the observed X . D’Aspremont et al. (2012), Candès and Tao (2010), Chatterjee (2015), Gavish and Donoho (2014b), Shabalin and Nobel (2013), and others have studied regimes where it is possible to accurately do so.

Singular-value shrinkage Classical approaches to estimating μ from X are centered around singular-value decomposition (SVD) algorithms. Let

$$X = \sum_{l=1}^{\min\{n,p\}} u_l d_l v_l^\top \quad (1)$$

denote the SVD of X . Then, if we believe that μ should have rank k , the standard SVD estimator $\hat{\mu}_k$ for μ is

$$\hat{\mu}_k = \sum_{l=1}^k u_l d_l v_l^\top. \quad (2)$$

In other words, we estimate μ using the closest rank- k approximation to X . Often, however, the plain rank- k estimator (2) is found to be noisy, and its performance can be improved by regularization. Existing approaches to regularizing $\hat{\mu}_k$ focus on singular value shrinkage, and use

$$\hat{\mu}^{\text{shrink}} = \sum_{l=1}^{\min\{n,p\}} u_l \psi(d_l) v_l^\top, \quad (3)$$

where ψ is a shrinkage function that is usually chosen in a way that makes $\hat{\mu}^{\text{shrink}}$ the closest to μ according to a loss function. Several authors have proposed various choices for ψ (e.g., Candès et al., 2013; Chatterjee, 2015; Gavish and Donoho, 2014a; Josse and Sardy, 2015; Shabalin and Nobel, 2013; Verbanck et al., 2013).

Methods based on singular-value shrinkage have achieved considerable empirical success. They also have provable optimality properties in the Gaussian noise model where $X = \mu + \varepsilon$ and the ε_{ij} are independent and identically distributed Gaussian noise terms (Shabalin and Nobel, 2013). However, in the non-Gaussian case, mere singular-value shrinkage can prove to be limiting, and we may also need to rotate the singular vectors u_l and v_l in order to achieve good performance.

Stable autoencoding In this paper, we propose a new framework for regularized low-rank estimation that does not start from the singular-value shrinkage point of view. Rather, our approach is motivated by a simple plug-in bootstrap idea (Efron and Tibshirani, 1993). It is well known that the classical SVD estimator $\hat{\mu}_k$ can be written as (Boulevard and Kamp, 1988; Baldi and Hornik, 1989)

$$\hat{\mu}_k = X B_k, \quad \text{where } B_k = \operatorname{argmin}_B \left\{ \|X - X B\|_2^2 : \operatorname{rank}(B) \leq k \right\}, \quad (4)$$

where $\|M\|_2^2 = \operatorname{tr}(M^\top M)$ denotes the Frobenius norm. The matrix B , called a linear *autoencoder* of X , allows us to encode the features of X using a low-rank representation.

Now, in the context of our noise model $X \sim \mathcal{L}(\mu)$, we do not just want to compress X , and instead want to recover μ from X . From this perspective, we would much prefer to estimate μ using an oracle encoder matrix that formally provides the best linear approximation of μ given our noise model

$$\hat{\mu}_k^* = X B_k^*, \quad \text{where } B_k^* = \operatorname{argmin}_B \left\{ \mathbb{E}_{X \sim \mathcal{L}(\mu)} \left[\|\mu - X B\|_2^2 \right] : \operatorname{rank}(B) \leq k \right\}. \quad (5)$$

We of course cannot solve for B_k^* because we do not know μ . But we can seek to approximate B_k^* by solving the optimization problem (5) on a well-chosen bootstrap distribution.

More specifically, our goal is to create bootstrap samples $\tilde{X} \sim \tilde{\mathcal{L}}(X)$ such that the distribution of \tilde{X} around X is representative of the distribution of X around μ . Then, we can solve an analogue to (5) on the bootstrap samples \tilde{X} :

$$\hat{\mu}_k^{\text{stable}} = X \hat{B}_k, \quad \text{where } \hat{B}_k = \operatorname{argmin}_B \left\{ \mathbb{E}_{\tilde{X} \sim \tilde{\mathcal{L}}(X)} \left[\|\tilde{X} - \tilde{X} B\|_2^2 \right] : \operatorname{rank}(B) \leq k \right\}. \quad (6)$$

We call this choice of \tilde{B}_k a *stable autoencoder* of X , as it provides a parsimonious encoding of the features of X that is stable when perturbed with bootstrap noise $\mathcal{L}(X)$. The motivation behind this approach is that we want to shrink $\hat{\mu}_k$ aggressively along directions where the bootstrap reveals instability, but do not want to shrink $\hat{\mu}_k$ too much along directions where our measurements are already accurate.

Bootstrap models for stable autoencoding A challenge in carrying out the program (6) is in choosing how to generate the bootstrap samples \tilde{X} . In the classical statistical setting, we have access to $m \gg 1$ independent training samples and can thus create a bootstrap dataset by simply re-sampling the training data with replacement. In our setting, however, we only have a single matrix X , i.e., $m = 1$. Thus, we must find another avenue for creating bootstrap samples \tilde{X} .

To get around this limitation, we use a *Lévy bootstrap* (Wager et al., 2016). Before defining the abstract bootstrap scheme below, we first discuss some simple examples. In the **Gaussian** case $X_{ij} \sim \mathcal{N}(\mu_{ij}, \sigma^2)$, Lévy bootstrapping is equivalent to a parametric bootstrap:

$$\tilde{X}_{ij} = X_{ij} + \tilde{\epsilon}_{ij}, \quad \text{where } \tilde{\epsilon}_{ij} \stackrel{\text{iid}}{\sim} \mathcal{N}\left(0, \frac{\delta}{1-\delta} \sigma^2\right), \quad (7)$$

and $\delta \in (0, 1)$ is a tuning parameter that governs the regularization strength. Meanwhile, in the **Poisson** case $X_{ij} \sim \text{Poisson}(\mu_{ij})$, Lévy bootstrapping involves randomly deleting a fraction δ of the counts comprising the matrix X , and up-weighting the rest:

$$\tilde{X}_{ij} \sim \frac{1}{1-\delta} \text{Binomial}(X_{ij}; 1-\delta), \quad (8)$$

where again $\delta \in (0, 1)$ governs the regularization strength. In the case $\delta = 0.5$, this is equivalent to the “double-or-nothing” bootstrap on the individual counts of X (e.g., Owen and Eckles, 2012).

These two examples already reveal a variety of different phenomena. On one hand, stable autoencoding with the Gaussian noise model (7) reduces to a singular-value shrinkage estimator (Section 2), and thus leads us back to the classical literature on low-rank matrix estimation. Conversely, with the Poisson-adapted bootstrap (8), the method (6) rotates singular vectors instead of just shrinking singular values; and in our experiments, it outperforms several variants of singular-value shrinkage that have been proposed in the literature.

The Lévy bootstrap Having first surveyed our two main examples above, we now present the more general Lévy bootstrap that can be used to carry out stable autoencoding in a wider variety of exponential family models. To motivate this approach, suppose that we can generate our matrix X as a sum of independent components:

$$X = \sum_{b=1}^B X_b^{(B)}, \quad \text{where } X_1^{(B)}, \dots, X_B^{(B)} \quad (9)$$

are independent and identically distributed. For example, if X is Gaussian with $X_{ij} \sim \mathcal{N}(\mu_{ij}, \sigma)$, then (9) holds with $(X_b^{(B)})_{ij} \sim \mathcal{N}(\mu_{ij}/B, \sigma/B)$. If the above construction were to hold and we also knew the individual components $X_b^{(B)}$, we could easily create bootstrap

samples \tilde{X} as

$$\tilde{X} = \frac{1}{1-\delta} \sum_{b=1}^B \tilde{W}_b X_b^{(B)}, \quad \text{where } \tilde{W}_b \stackrel{\text{iid}}{\sim} \text{Bernoulli}(1-\delta), \quad (10)$$

and $\delta \in (0, 1)$ governs the noising strength. Now in reality, we do not know the terms in (9), and so cannot carry out (10).

However, Wager et al. (2016) establish conditions under which this limitation does not matter. Suppose that X is drawn from an exponential family distribution

$$X \sim f_{\Theta}(\cdot), \quad f_{\Theta}(x) = h(X) \exp\left[\sum_{i,j} \Theta_{ij} X_{ij} - \zeta(\Theta)\right], \quad (11)$$

where $\Theta \in \mathbb{R}^{n \times p}$ is an unknown parameter vector, $h(\cdot)$ is a carrier distribution, and $\zeta(\cdot)$ is the log-partition function. Suppose, moreover, that X has an infinitely divisible distribution, or, equivalently, that $X = A(1)$ for some matrix-valued Lévy process $A(t) \in \mathbb{R}^{n \times p}$ for $t > 0$ (e.g., Durrett, 2010). Then, we can always generate bootstrap replicates \tilde{X} distributed as

$$\tilde{X} \sim \tilde{\delta} h(X) := \frac{1}{1-\delta} A(1-\delta) \mid A(1) = X, \quad \text{for any } \delta \in (0, 1), \quad (12)$$

without requiring knowledge of the underlying Θ ; Wäger et al. (2016) provide explicit formulas for carrying out (12) that only depend on the carrier $h(\cdot)$.

The upshot of this bootstrap scheme is that it allows us to preserve the generative structure encoded by Θ without needing to know the true parameter. In the Gaussian and Poisson cases, (12) reduces to (7) and (8) respectively; however, this Lévy bootstrap framework also induces other noising schemes, such as multiplicative noising when X has a Gamma distribution.

Rank selection via iterated stable autoencoding Returning now to our main focus, namely regularized low-rank matrix estimation, we note that one difficulty with the estimator $\hat{\mu}_k^{\text{stable}}$ (6) is that we need to select the rank k of the estimator beforehand in addition to the shrinkage parameter δ . Surprisingly, we can get around this issue by iterating the optimization problem (6) until we converge to a limit:

$$\hat{\mu}^{\text{iter}} = X \hat{B}, \quad \text{where } \hat{B} = \underset{B}{\text{argmin}} \left\{ \mathbb{E}_{\tilde{X} \sim \mathcal{L}(\hat{\mu}^{\text{iter}}, X)} \left[\left\| \hat{\mu}^{\text{iter}} - \tilde{X} \hat{B} \right\|_2^2 \right] \right\}. \quad (13)$$

In Section 3, we establish conditions under which the iterative algorithm implied above in fact converges and, moreover, the resulting fixed point $\hat{\mu}^{\text{iter}}$ is low rank. In our experiments, this *iterated stable autoencoder* does a good job at estimating k of the underlying signal; thus, all the statistician needs to do is to specify a single regularization parameter $\delta \in (0, 1)$ that simultaneously controls both the amount of shrinkage and the rank k of the final estimate $\hat{\mu}$.

To summarize, our approach as instantiated in (6) and (13) provides us with a flexible framework for transforming noise models $\mathcal{L}(\cdot)$ into regularized matrix estimators via the

Lévy bootstrap. In the Gaussian case, our framework yields estimators that resemble best-practice singular-value shrinkage methods. Meanwhile, in the non-Gaussian case, stable autoencoding allows us to learn new singular vectors for $\hat{\mu}$. In our experiments, this allowed us to substantially improve over existing techniques.

Finally, we also discuss extensions to stable autoencoding: In Section 4, we show how to use our method to regularize correspondence analysis (Greenacre, 1984, 2007), which is one of the most popular ways to analyze multivariate count data and underlies several modern machine learning algorithms.

A software implementation of the proposed methods is available through the R-package `denoiseR` (Josse et al., 2016).

1.1 Related work

There is a well-known duality between regularization and feature noising schemes. As shown by Bishop (1995), linear regression with features perturbed with Gaussian noise, i.e.,

$$\hat{\beta} = \operatorname{argmin}_{\beta} \left\{ \mathbb{E}_{\varepsilon_{ij} \stackrel{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)} \left[\|Y - (X + \varepsilon)\beta\|_2^2 \right] \right\},$$

is equivalent to ridge regularization with Lagrange parameter $\lambda = n\sigma^2$:

$$\hat{\beta}_{\lambda}^{(R)} = \operatorname{argmin}_{\beta} \left\{ \|Y - X\beta\| + \lambda \|\beta\|_2 \right\}.$$

Because ridge regression is equivalent to adding homoskedastic noise to X , we can think of ridge regression as making the estimator robust against round perturbations to the data.

However, if we perturb the features X using non-Gaussian noise or are working with a non-quadratic loss function, artificial feature noising can yield new regularizing schemes with desirable properties (Globerson and Roweis, 2006; Simard et al., 2000; van der Maaten et al., 2013; Wager et al., 2013; Wang et al., 2013). Our proposed estimator $\hat{\mu}_k^{\text{stable}}$ can be seen as an addition to this literature, as we seek to regularize $\hat{\mu}_k$ by perturbing the autoencoder optimization problem. The idea of regularizing via feature noising is also closely connected to the dropout learning algorithm for training neural networks (Srivastava et al., 2014), which aims to regularize a neural network by randomly omitting hidden nodes during training time. Dropout and its generalizations have been found to work well in many large-scale prediction tasks (e.g., Baldi and Sadowski, 2014; Goodfellow et al., 2013; Krizhevsky et al., 2012).

Our method can be interpreted as an empirical Bayes estimator (Efron, 2012; Robbins, 1985), in that the stable autoencoder problem (6) seeks to find the best linear shrinker in an empirically chosen Bayesian model. There is also an interesting connection between stable autoencodings, and more traditional Bayesian modeling such as latent Dirichlet allocation (LDA) (Blei et al., 2003), in that the Lévy bootstrap (12) uses a generalization of the LDA generative model to draw bootstrap samples \tilde{X} . Thus, our method can be seen as benefiting from the LDA generative structure without committing to full Bayesian inference (Kucukelbir and Blei, 2015; Wager et al., 2014, 2016).

There is a large literature on low-rank exponential family estimation (Collins et al., 2001; de Leeuw, 2006; Fithian and Mazumder, 2013; Goodman, 1985; Li and Tao, 2013). In

the simplest form of this idea, each matrix entry is modeled using the generic exponential family distribution (11); the goal is then to maximize the log-likelihood of X subject to a low-rank constraint on the natural parameter matrix Θ , rather than the mean parameter matrix μ as in our setting. The main difficulty with this approach is that the resulting rank-constrained problem is no longer efficiently solvable; one way to avoid this issue is to relax the rank constraint into a nuclear norm penalty. Extending our bootstrap-based regularization framework to low-rank exponential family estimation would present an interesting avenue for further work. We also note the work of Buntine (2002), who seeks to maximize the multinomial log-likelihood of X subject to a low-rank constraint on μ using an approximate variational method.

Finally, one of the advantages of our stable autoencoding approach is that it lets us move beyond singular value shrinkage, and learn better singular vectors than those provided by the SVD. Another approach to way to improve on the quality of the learned singular vectors is to impose structural constraints on them, such as sparsity (Jolliffe et al., 2003; Uçell et al., 2014; Witten et al., 2009; Zou et al., 2006).

2. Fitting stable autoencoders

In this section, we show how to solve (6) under various bootstrap models $\tilde{\mathcal{L}}(\cdot)$. This provides us with estimators $\hat{\mu}_k^{\text{stable}}$ that are interesting in their own right, and also serves as a stepping stone to the iterative solutions from Section 3 that do not require pre-specifying the rank k of the underlying signal.

Isotropic stable autoencoders and singular-value shrinkage At first glance, the estimator $\hat{\mu}_k^{\text{stable}}$ defined in (6) may seem like a surprising idea. It turns out, however, that under the isotropic Gaussian¹ noise model

$$X = \mu + \varepsilon, \quad \text{with } \varepsilon_{ij} \stackrel{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2) \quad \text{for all } i = 1, \dots, n \text{ and } j = 1, \dots, p, \quad (14)$$

$\hat{\mu}_k^{\text{stable}}$ with bootstrap noise as in (7) is equivalent to a classical singular-value shrinkage estimator (3) with $\psi(d) = d/(1 + \lambda/d^2)$ and $\lambda = \delta/(1 - \delta)n\sigma^2$.

Theorem 1 *Let $\hat{\mu}_k^{\text{stable}}$ be the rank- k estimator for μ induced by the stable autoencoder (6) with a bootstrap model $\tilde{\mathcal{L}}_{\delta}(\cdot)$ as defined in (7) with some $0 < \delta < 1$. This estimator can also be written as the solution to a ridge-regularized autoencoder:*

$$\hat{\mu}_k^{\text{stable}} = X\hat{B}_k, \quad \text{where } \hat{B}_k = \operatorname{argmin}_B \left\{ \|X - XB\|_2^2 + \lambda \|B\|_2^2 : \operatorname{rank}(B) \leq k \right\} \quad (15)$$

with $\lambda = \delta/(1 - \delta)n\sigma^2$. Moreover, using notation from (1), we can write $\hat{\mu}_k^{\text{stable}}$ as

$$\hat{\mu}_k^{\text{stable}} = \sum_{l=1}^k u_l \frac{d_l}{1 + \lambda/d_l^2} v_l^{\top}. \quad (16)$$

1. Theorem 1 holds for all isotropic noise models with $\operatorname{Var}[\varepsilon_{ij}] = \sigma^2$ for all i and j , and not just the Gaussian one. However, in practice, isotropic noise is almost always modeled as Gaussian.

In the isotropic Gaussian noise case, singular-value shrinkage methods were shown by Shabalin and Nobel (2013) to have strong optimality properties for estimating μ . Thus, it is reassuring that our framework recovers an estimator of this class in the Gaussian case. In fact, the induced shrinkage function resembles a first-order approximation to the one proposed by Verbanck et al. (2013).

Non-isotropic stable autoencoders Stable autoencoders with isotropic noise are attractive in the sense that we can carefully analyze their behavior in closed form. However, from a practical point of view, our procedure is most useful outside of the isotropic regime, as it induces new estimators $\hat{\mu}$ that do not reduce to singular-value shrinkage. Even in the non-isotropic noise model, low-rank stable autoencoders can still be efficiently solved, as shown below.

Theorem 2 For a generic bootstrap model $\tilde{\mathcal{L}}(\cdot)$, the matrix \hat{B}_k from (6) can be obtained as follows:

$$\hat{B}_k = \operatorname{argmin}_B \left\{ \|X - XB\|_2^2 + \|S^{\frac{1}{2}}B\|_2^2 : \operatorname{rank}(B) \leq k \right\}, \quad (17)$$

where S is a $p \times p$ diagonal matrix with

$$S_{ij} = \sum_{t=1}^n \operatorname{Var}_{\tilde{X} \sim \tilde{\mathcal{L}}(X)} [\tilde{X}_{ij}]. \quad (18)$$

From a computational point of view, we can write the solution \hat{B}_k of (17) as

$$\hat{B}_k = \operatorname{argmin}_B \left\{ \operatorname{tr} \left((B - \hat{B})^\top (X^\top X + S) (B - \hat{B}) \right) : \operatorname{rank}(B) \leq k \right\}, \quad \text{where} \quad (19)$$

$$\hat{B} = (X^\top X + S)^{-1} X^\top X \quad (20)$$

is the solution of (17) without the rank constraint.

The optimization problem in (19) can be easily solved by taking the top k terms from the eigenvalue decomposition of $\hat{B}^\top (X^\top X + S) \hat{B}$: the matrix \hat{B}_k can then be recovered by solving a linear system (e.g., Takane, 2013). Thus, despite what we might have expected, solving the low-rank constrained stable autoencoder problem (6) with a generic noise model is not substantially more computationally demanding than singular-value shrinkage. Note that in (20), the matrix S is not equal to a constant times the identity matrix due to the non-isotropic noise, and so the resulting singular vectors of $\hat{\mu}_k^{\text{stable}} = X\hat{B}_k$ are not in general the same as those of X .

Selecting the tuning parameter Our stable autoencoder depends on a tuning parameter $\delta \in (0, 1)$, corresponding to the fraction of the information in the full data X that we throw away when creating pseudo-datasets \tilde{X} using the Lévy bootstrap (12). More prosaically, the parameter δ manifests itself as a multiplier $\delta/(1 - \delta) \in (0, \infty)$ on the effective stable autoencoding penalty, either explicitly in (15) or implicitly in (18) through the dependence on $\tilde{\mathcal{L}}_\delta(\cdot)$.

Algorithm 1 Low-rank matrix estimation via iterated stable autoencoding.

```

 $\hat{\mu} \leftarrow X$ 
for all  $j = 1, \dots, p$ 
   $S_{jj} \leftarrow \sum_{t=1}^n \operatorname{Var}_{\tilde{X} \sim \tilde{\mathcal{L}}_\delta(X)} [\tilde{X}_{tj}]$ 
while algorithm has not converged do
   $\hat{B} \leftarrow (\hat{\mu}^\top \hat{\mu} + S)^{-1} \hat{\mu}^\top \hat{\mu}$ 
   $\hat{\mu} \leftarrow X \hat{B}$ 
end while
```

One plausible default value is to set $\delta = 1/2$. This corresponds to using half of the information in the full data X to generate each bootstrap sample \tilde{X} , and is closely related to bagging (Breiman, 1996); see Buja and Snetzle (2006) for a discussion of the connections between half-sampling and bagging. Conversely, we could also opt for a data-driven choice of δ . The software implementation of stable autoencoding in `denoiser` (Josse et al., 2016) provides a cell-wise cross-validation algorithm for picking δ .

Finally, we note that our estimator is not in general invariant to transposition $X \rightarrow X^\top$. For example, in the isotropic case (15), we see that λ depends on n but not on p . Meanwhile, in the non-isotropic case (17), transposition may also affect the learned singular vectors. By default, we transpose X such that $n > p$, i.e., we pick the transposition of X that makes the matrix B smaller.

3. Iterated stable autoencoding

One shortcoming of the stable autoencoders discussed in the previous section is that we need to specify the rank k as a tuning parameter. Selecting the rank for multivariate methods is often a difficult problem, and many heuristics are available in the literature (Jolliffe, 2002; Josse and Husson, 2011). The stable autoencoding framework, however, induces a simple solution to the rank-selection problem: As we show here, iterating our estimation scheme from the previous section automatically yields low-rank solutions, and allows us to specify a single tuning parameter δ instead of both δ and k .

At a high level, our goal is to find a solution to

$$\hat{\mu}^{\text{iter}} = X \hat{B}, \quad \text{where } \hat{B} = \operatorname{argmin}_B \left\{ \mathbb{E}_{\tilde{X} \sim \tilde{\mathcal{L}}_\delta(\hat{\mu}^{\text{iter}}, X)} \left[\left\| \hat{\mu}^{\text{iter}} - \tilde{X} B \right\|_2^2 \right] \right\} \quad (21)$$

by iteratively updating \hat{B} and $\hat{\mu}$. As seen in the previous section, stable autoencoding only depends on $\mathcal{L}_\delta(\hat{\mu}, X)$ through the first two moments of \tilde{X} ; here, we simply specify them as

$$\mathbb{E}_{\tilde{X} \sim \tilde{\mathcal{L}}_\delta(\hat{\mu}, X)} [\tilde{X}] = \hat{\mu} \quad \text{and} \quad \operatorname{Var}_{\tilde{X} \sim \tilde{\mathcal{L}}_\delta(\hat{\mu}, X)} [\tilde{X}] = \operatorname{Var}_{\tilde{X} \sim \tilde{\mathcal{L}}_\delta(X)} [\tilde{X}], \quad (22)$$

where $\tilde{\mathcal{L}}_\delta(X)$ is obtained using the Lévy bootstrap as before.

Now, using the unconstrained solution (20) from Theorem 2 to iterate on the relation (21), we get the formal procedure described in Algorithm 1. Note that we do not update the matrix S , which encodes the variance of the noise distribution, and only update $\hat{\mu}$. As shown below, our algorithm converges to a well-defined solution; moreover, the solution is

regularized in that $\hat{\mu}^\top \hat{\mu}$ is smaller than $X^\top X$ with respect to the positive semi-definite cone ordering.

Theorem 3 *Algorithm 1 converges to a fixed point $\hat{\mu} = X\hat{B}$. Moreover,*

$$\hat{\mu}^\top \hat{\mu} \preceq X^\top X.$$

Moreover, iterated stable autoencoding can provide generic low-rank solutions $\hat{\mu}$.

Theorem 4 *Let $\hat{\mu}$ be the limit of our iterative algorithm, and let $u \in \mathbb{R}^p$ be any (normalized) eigenvector of $\hat{\mu}^\top \hat{\mu}$. Then, either*

$$\|\hat{\mu} u\|_2 = 0, \text{ or } \|\hat{\mu} u\|_2 \geq \frac{1}{\|XS^{-1}u\|_2}.$$

The reason our algorithm converges to low-rank solutions is that our iterative scheme does not have any fixed points “near” low-dimensional subspaces. Specifically, as shown in Theorem 4, for any eigenvector of $\hat{\mu}^\top \hat{\mu}$, either $\|\hat{\mu} u\|_2$ must be larger than some cutoff, or it must be exactly zero. Thus, $\hat{\mu}$ cannot have any small but non-zero singular values. In our experiments, we have found that our algorithm in fact conservatively estimates the true rank of the underlying signal.

Finally we note that, in the isotropic case, our iterative algorithm again admits a closed-form solution. Looking at this solution can give us more intuition about what our algorithm does in the general case. In particular, we note that the algorithm never shrinks a singular value by more than a factor $1/2$ without pushing it all the way to 0.

Proposition 5 *In the isotropic Gaussian case (14) with $\delta = 1/2$, our iterative algorithms converges to*

$$\hat{\mu}^{\text{iter}} = \sum_{l=1}^{\min\{n,p\}} u_l \psi(d_l) v_l^\top, \text{ where } \psi(d) = \begin{cases} \frac{1}{2} (d + \sqrt{d^2 - 4n\sigma^2}) & \text{for } d^2 \geq 4n\sigma^2, \\ 0 & \text{else.} \end{cases} \quad (23)$$

Since the isotropic Gaussian matrix estimation problem has been thoroughly studied, we can compare the shrinkage rule $\psi(\cdot)$ with known asymptotically optimal ones. Gavish and Donoho (2014b) provide a comprehensive treatment of optimal singular-value shrinkage for different loss functions in a Marcenko-Pastur asymptotic regime, where n and p both diverge to infinity such that $p/n \rightarrow \beta$ for some $0 < \beta \leq 1$ while the rank and the scale of the signal remains fixed. This specific asymptotic setting has also been investigated by, among others, Johnstone (2001) and Shabalin and Nobel (2013).

In what appears to be a remarkable coincidence, for the square case $\beta = 1$, our shrinkage rule (23) corresponds exactly to the Marcenko-Pastur optimal shrinkage rule under operator-norm loss $\|\hat{\mu} - \mu\|_{\text{op}}$; see the proof of Proposition 5 for a derivation. At the very least, this connection is reassuring as it suggests that our iterative scheme may yield statistically reasonable estimates $\hat{\mu}^{\text{iter}}$ for other noise models too. It remains to be seen whether this connection reflects a deeper theoretical phenomenon.

4. Application: regularizing correspondence analysis

When X contains count data, we have a natural noise model $X_{ij} \sim \text{Poisson}(\mu_{ij})$ that is compatible with the Lévy bootstrap, and so our stable autoencoding framework is easy to apply. In this situation, however, X is often analyzed by correspondence analysis (Greenacre, 1984, 2007) rather than using a direct singular-value decomposition. Correspondence analysis, a classical statistics technique pioneered by Hirschfeld (1935) and Benzécri (1986, 1973), underlies variants of many modern machine learning applications such as spectral clustering on graphs (e.g., Ng et al., 2002; Shi and Malik, 2000) or topic modeling for text data (see Section 6.1). In this section, we show how to regularize correspondence analysis by stable autoencoding. This discussion also serves as a blueprint for extending our method to other low-rank multivariate techniques such as principal component analysis or canonical correlation analysis.

Correspondence analysis involves taking the singular-value decomposition of a transformed matrix M :

$$M = R^{-\frac{1}{2}} \left(X - \frac{1}{N} rc^\top \right) C^{-\frac{1}{2}}, \quad \text{where } R = \text{diag}(r), C = \text{diag}(c), \quad (24)$$

N is the total number of counts, and r and c are vectors containing the row and column sums of X . This transformation M has several motivations. For example, suppose that X is a 2-way contingency table, i.e., that we have N samples for which we measure two discrete features $A \in \{1, \dots, n\}$ and $B \in \{1, \dots, p\}$, and X_{ij} counts the number of samples with $A = i$ and $B = j$. Then M measures the distance between X and a hypothetical contingency table where A and B are independently generated with the same marginal distributions as before; in fact, the standard χ^2 -test for independence of X uses $\|M\|_2^2$ as its test statistic. Meanwhile, if X is the adjacency matrix of a graph, then M is a version of the symmetric normalized graph Laplacian where we have projected out the first trivial eigenvector. Once we have a rank- k estimate of \hat{M}_k obtained as in (2), we get

$$\hat{\mu}_k^{CA} = R^{\frac{1}{2}} \hat{M}_k C^{\frac{1}{2}} + \frac{1}{N} rc^\top. \quad (25)$$

Our goal is to get a better estimator \widehat{M} for the matrix M of the population; we then transform \widehat{M} into an estimate of μ using the same formula (25).

Following (6), we propose regularizing the choice of M as follows:

$$\widehat{M}_k^{\text{stable}} = M \widehat{B}_k, \quad \text{where} \quad (26)$$

$$\widehat{B}_k = \text{argmin}_B \left\{ \mathbb{E}_{\tilde{X} \sim \tilde{\mathcal{L}}_k(X)} \left[\left\| M - R^{-\frac{1}{2}} \left(\tilde{X} - \frac{1}{N} rc^\top \right) C^{-\frac{1}{2}} B \right\|_2^2 \right] : \text{rank}(B) \leq k \right\}.$$

Just as in Theorem 2, we can show that \widehat{B}_k solves

$$\widehat{B}_k = \text{argmin}_B \left\{ \|M - MB\|_2^2 + \left\| S_M^{\frac{1}{2}} B \right\|_2^2 : \text{rank}(B) \leq k \right\}, \quad (27)$$

where S_M is a diagonal matrix with $(S_M)_{jj} = c_j^{-1} \sum_{i=1}^n \text{Var}_{\tilde{X} \sim \tilde{\mathcal{L}}_k(X)}[\tilde{X}_{ij}]/r_i$. We can efficiently solve for (27) using the same method as in (19) and (20). Finally, if we do not want to fix the rank k , we can use an iterative scheme as in Section 3.

Since X contains count data, we generate the bootstrap samples $\tilde{X} \sim \tilde{\mathcal{L}}_{\delta}(X)$ using the Poisson-compatible bootstrap algorithm (8), i.e., $\tilde{X}_{ij} \sim (1 - \delta)^{-1} \text{Binomial}(X_{ij}, 1 - \delta)$. Interestingly, if we had chosen to sample \tilde{X} from an independent contingency table with

$$\mathbb{E}_{\tilde{X} \sim \tilde{\mathcal{L}}_{\delta}}[\tilde{X}] = \frac{1}{N} r c^T, \quad \text{Var}_{\tilde{X} \sim \tilde{\mathcal{L}}_{\delta}}[\tilde{X}_{ij}] = \frac{\delta}{1 - \delta} \frac{r_i c_j}{N}, \quad (28)$$

we would have obtained a regularization matrix $S_M = n\delta/(N(1 - \delta))I_{p \times p}$. Because S_M is diagonal, the resulting estimator \widehat{M}_A could then be obtained from M by singular value shrinkage. Thus, if we want to regularize correspondence analysis applied to a nearly independent table, singular value shrinkage based methods can achieve good performance; however, if the table has strong dependence, our framework provides a more principled way of being robust to sampling noise.

5. Simulation experiments

To assess our proposed methods, we first run comparative simulation studies for different noise models. We begin with a sanity check: in Section 5.1, we reproduce the isotropic Gaussian noise experiments of Candès et al. (2013), and find that our method is competitive with existing approaches on this standard benchmark.

We then move to the non-isotropic case, where we can take advantage of our method’s ability to adapt to different noise structures. In Section 5.2 we show results on experiments with Poisson noise, and find that our method substantially outperforms its competitors. Finally, in Section 6, we apply our method to real-world applications motivated by topic modeling and sensory analysis.

5.1 Gaussian noise

We compare our estimators to existing ones by reproducing the simulations of Candès et al. (2013). For this experiment, we generated data matrices of size 200×500 according to the Gaussian noise model (14) with four signal-to-noise ratios $\text{SNR} \in \{0.5, 1, 2, 4\}$ calculated as $1/(\sigma\sqrt{np})$, and two values for the underlying rank $k \in \{10, 100\}$; results are in Table 1.

Methods under consideration: Our goal is to evaluate the performance of the stable autoencoder (SA) as defined in (6) and the iterated stable autoencoder (ISA) described in Algorithm 1. As discussed in Section 2, we applied our stable autoencoding methods to X^T rather than X , so that n was larger than p ; and set the tuning parameter to $\delta = 1/2$. For ISA, we ran the iterative Algorithm 1 for 100 steps, although the algorithm appeared to become stable after 10 steps already.

In addition to our two methods, we also consider the following estimators:

- Truncated SVD with fixed rank k (TSVD- k). This is the classical approach (2).
- Adaptively truncated SVD (TSVD- τ), using the asymptotically optimal threshold of Gavish and Donoho (2014a).
- Asymptotically optimal singular-value shrinkage (ASYMIP) in the Marchenko-Pastur asymptotic regime given the Frobenius norm loss (Shabalin and Nobel, 2013; Gavish

and Donoho, 2014b), with shrinkage function

$$\psi(d) = \begin{cases} \frac{1}{d} \sqrt{(d^2 - (1 + \beta)n\sigma^2)^2 - 4\beta n^2 \sigma^4} & \text{for } d^2 \geq (1 + \sqrt{\beta})^2 n\sigma^2, \\ 0 & \text{else,} \end{cases} \quad (29)$$

where $\beta = p/n$ is the aspect ratio, assuming without loss of generality that $p \leq n$.

- The shrinkage scheme of Verbanck et al. (2013) motivated by low-noise (LN) asymptotics. It uses for $\psi(d)$ in (3),

$$\psi(d) = \begin{cases} d \left(1 - \frac{\sigma^2}{d^2}\right) & \text{for } l \leq k, \\ 0 & \text{else.} \end{cases} \quad (30)$$

All the estimators are defined assuming the variance of the noise scale σ^2 to be known. In addition, TSVD- k , SA, and LN require the rank k as a tuning parameter. In this case, we set k to the true rank of the underlying signal.

As our simulation study makes clear, the proposed methods have very different strengths and weaknesses. Both methods that apply a hard thresholding rule to the singular values, namely TSVD- k and TSVD- τ , provide accurate MSE when the SNR is high but break down in low SNR settings. Conversely, the SVST behaves well in low SNR settings, but struggles in other regimes. This is not surprising, as the method over-estimates the rank of μ . This behavior is reminiscent of what happens in lasso regression (Tibshirani, 1996) when too many variables are selected (Zou, 2006; Zhang and Huang, 2008).

Meanwhile, the estimators with non-linear singular-value shrinkage functions, namely SA, ISA, ASYMIP, and LN are more flexible and perform well except in the very difficult scenario where the signal is overwhelmed by the noise. Both ASYMIP and ISA estimate the rank accurately except when the signal is nearly indistinguishable from the noise ($\text{SNR} = 0.5$ and $k = 100$).

5.2 Poisson noise

Once we move beyond the isotropic Gaussian case, our method can both learn better singular vectors and out-perform its competitors in terms of MSE. We illustrate this phenomenon with a simple simulation example, where we drew X of size $n = 50$ and $p = 20$ from a Poisson distribution with expectation μ of rank 3 represented in Figure 1. Because the three components of μ have different levels of concentration—the first component is rather diffuse, while the third one is concentrated in a corner—adapting to the Poisson variance structure is important.

We varied the effective signal-to-noise ratio by adjusting the mean number of counts in X , i.e., $N = \sum_{ij} \mu_{ij}$. We then report results for the normalized mean matrix μ/N . We used both SA and ISA to estimate μ from X ; in both cases, we generated \tilde{X} with the Poisson-compatible bootstrap noise model (8), and set $\delta = 1/2$. We also used LN, ASYMIP and

k	SNR	Stable		TSVD		ASYMP	SVST	LN
		SA	ISA	k	τ			
MSE								
10	4	0.004	0.004	0.004	0.004	0.004	0.008	0.004
100	4	0.037	0.036	0.038	0.038	0.037	0.045	0.037
10	2	0.017	0.017	0.017	0.016	0.017	0.033	0.017
100	2	0.142	0.143	0.152	0.158	0.146	0.156	0.141
10	1	0.067	0.067	0.072	0.072	0.067	0.116	0.067
100	1	0.511	0.775	0.733	0.856	0.600	0.448	0.491
10	0.5	0.277	0.251	0.321	0.321	0.250	0.353	0.257
100	0.5	1.600	1.000	3.164	1.000	0.961	0.852	1.477
Rank								
10	4		10		10	10	65	
100	4		100		100	100	193	
10	2		10		10	10	63	
100	2		100		100	100	181	
10	1		10		10	10	59	
100	1		29.6		38	64	154	
10	0.5		10		10	10	51	
100	0.5		0		0	15	86	

Table 1: Mean cell-wise squared error (top) and rank estimates (bottom) obtained by the methods described in Section 5.1, averaged over 50 simulation replications. The best results for each row are indicated in bold.

TSVD- τ as baselines, although they are only formally motivated in the Gaussian model. These methods require a value for σ . For LN, we used the method recommended by Josse and Husson (2011):

$$\hat{\sigma}^2 = \frac{\|X - \sum_{l=1}^k u_l d_l v_l\|_2^2}{np - nk - kp + k^2}. \quad (31)$$

For ASYMP and TSVD- τ , we used the estimator suggested in Gavish and Donoho (2014b), $\hat{\sigma} = d_{med} / \sqrt{n\mu_\beta}$, where d_{med} is the median of the singular values of X and μ_β is the median of the Marcenko-Pastur distribution with aspect ratio β .

In addition to providing MSE (Table 2), we also report the alignment of the row/column directions U and V with those of the true mean matrix μ (Table 3). We measured alignment using the RV coefficient, which is a matrix version of Pearson's squared correlation coefficient ρ^2 that takes values between 0 and 1 (Escoufier (1973); see Josse and Holmes (2013) for a review):

$$RV(U, \hat{U}) = \frac{\text{tr}(U^T \hat{U} \hat{U}^T U)}{\sqrt{\text{tr}((U^T U)^2) \text{tr}((\hat{U}^T \hat{U})^2)}}. \quad (32)$$

Finally, we also report the mean estimated ranks in Table 4.

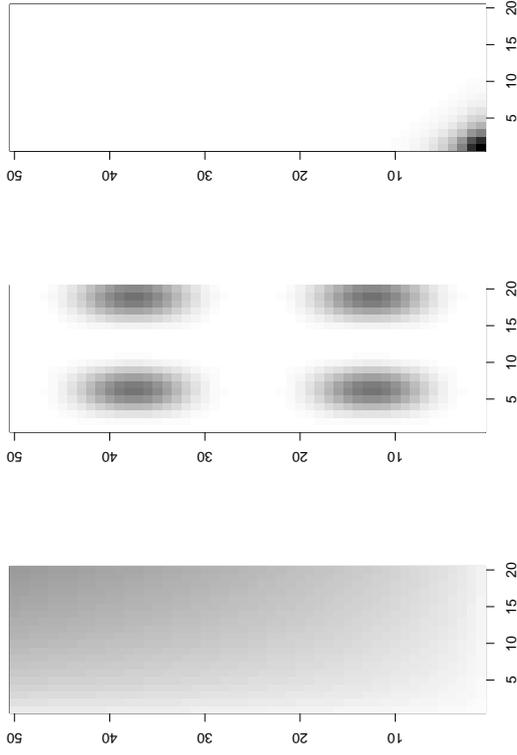


Figure 1: The 3 components of the mean of the underlying Poisson process; the dark arcs have the highest intensity. The corresponding singular values have relative magnitudes 1.1 : 1.4 : 1.

We see that our methods based on stable autoencoding do well across all noise levels. In the high-noise setting (i.e., with a small number of count observations N), the iterated stable autoencoder does particularly well, as it is able to use a lower rank in response to the weaker signal. As seen in Table 3, the ability to learn new singular vectors appears to have been useful here, as the \hat{U} and \hat{V} matrices obtained by stable autoencoding are much better aligned with the population ones than those produced by the SVD arc. We also see that, in the low noise setting where N is large, ISA recovers the true rank $k = 3$ almost exactly, whereas ASYMP and TSVD- τ do not. Finally, we note that all shrinkage methods did better than the baseline, namely the simple rank-3 SVD. Thus, even though LN, ASYMP and TSVD- τ are only formally motivated in the Gaussian noise case, our results suggest that they are still better than no regularization on generic problems.

N	Stable		TSVD		ASYNP		LN	
	SA	ISA	k	τ				
200	1.83	1.13	2.62	1.99	1.71		2.12	
400	0.76	0.51	1.08	0.93	0.77		0.88	
600	0.46	0.36	0.63	0.58	0.48		0.52	
800	0.33	0.29	0.44	0.42	0.35		0.37	
1000	0.25	0.24	0.32	0.33	0.27		0.28	
1200	0.20	0.19	0.25	0.27	0.22		0.22	
1400	0.16	0.15	0.20	0.22	0.19		0.18	
1600	0.14	0.13	0.17	0.19	0.16		0.15	
1800	0.12	0.11	0.14	0.16	0.14		0.13	
2000	0.11	0.10	0.13	0.15	0.13		0.12	

Table 2: Mean cell-wise squared error, averaged over 1000 simulation replications.

N	RV for U				RV for V			
	SVD	SA	ISA	SVD	SA	ISA	SVD	ISA
200	0.29	0.34	-	0.34	0.40	-	0.40	-
400	0.48	0.53	0.51	0.53	0.57	0.54	0.54	0.54
600	0.60	0.64	0.71	0.64	0.69	0.79	0.69	0.79
800	0.67	0.71	0.79	0.72	0.76	0.87	0.76	0.87
1000	0.74	0.77	0.82	0.79	0.83	0.89	0.83	0.89
1200	0.78	0.81	0.85	0.84	0.87	0.90	0.87	0.90
1400	0.82	0.85	0.86	0.87	0.90	0.92	0.90	0.92
1600	0.85	0.87	0.88	0.90	0.91	0.93	0.91	0.93
1800	0.87	0.88	0.89	0.92	0.93	0.94	0.93	0.94
2000	0.88	0.89	0.90	0.93	0.94	0.94	0.94	0.94

Table 3: RV coefficients between the estimated and true U and V matrices, averaged over 1000 simulation replications. For ISA, we only averaged performance over examples where the estimated rank was at least 3. In the $N = 200$ case, no results is given for ISA since the estimated rank is always less than 3.

6. Real-world examples

To highlight the wide applicability of our method, we use it on two real-world problems from different fields. We begin with a larger natural language application, where we use the iterated stable autoencoder to improve the quality of topics learned by latent semantic analysis, and evaluate results by end-to-end classifier performance. Next, in Section 6.2, we analyze a smaller dataset from a consumer survey, and show how our regularization schemes can improve the faithfulness of correspondence analysis graphical outputs commonly used by statisticians.

N	TSVD- τ	ASYNP	ISA
200	1.78	3.11	1.40
400	2.23	3.55	1.96
600	2.54	3.77	2.01
800	2.76	3.9	2.10
1000	2.94	3.99	2.36
1200	3.11	4.02	2.71
1400	3.17	4.04	2.92
1600	3.17	4.06	2.98
1800	3.22	4.08	3.00
2000	3.23	4.07	3.00

Table 4: Mean rank estimates for the Poisson simulation, averaged over 1000 simulation replications. The true rank of the underlying signal is 3.

	Document Averaged	Corresp. Analysis	Corresp. Analysis + ISA
Accuracy	62.1 %	61.8 %	67.0 %
Times Best	2/10,000	1/10,000	9,997/10,000

Table 5: Test set accuracy of a logistic regression classifier trained on topics learned by latent semantic analysis, averaged over 10,000 train/test splits. The topic models were run only once on all the (unlabeled) data; thus we are in a transductive setting. Each method used $k = 5$ topics; this number was automatically picked by ISA.

6.1 Learning topics for sentiment analysis

Many tasks in natural language processing involve computing a low-rank approximation to a *document/term-frequency* matrix X , i.e., X_{ij} counts the number of times word j appears in document i . The singular rows of X can then be interpreted as *topics* characterized by the prevalence of different words, and each document is described as a mixture of topics. The idea of learning topics using an SVD of (a normalized version of) the matrix X is called latent semantic analysis (Deerwester et al., 1990). Here, we argue that we can make the topics discovered by latent semantic analysis better by regularizing the SVD of X using an iterated stable autoencoder.

To do so, we examine the Rotten Tomatoes movie review dataset collected by Pang and Lee (2004), with $n = 2,000$ documents and $p = 50,921$ unique words. We learned topics with three variants of latent semantic analysis, which involve using different transformations/regularization schemes while taking an SVD.

- Document averaging: in order to avoid large documents dominating the fit, we compute the matrix $\Pi_{ij} = X_{ij} / \sum_j X_{ij}$. We then perform a rank- k SVD of Π .

- Correspondence analysis: we run a rank- k SVD on M from (24). This approach normalizes by both $R^{-\frac{1}{2}}$ and $C^{-\frac{1}{2}}$ in order to counteract the excess influence of both long documents and common words, instead of just using $\Pi = R^{-1}X$ as above.
- ISA-regularized correspondence analysis ($\delta = 0.5$).

Correspondence analysis with ISA picked $k = 5$ topics; we also used $k = 5$ for the other methods. The document averaging method did not appear to benefit much from regularization; presumably, this is because the matrix Π does not up-weight rare words.

Because n and p are both fairly large, it is difficult to evaluate the quality of the learned topics directly. To get around this, we used a more indirect approach and examined the quality of the learned decompositions of X by using them for sentiment classification. Specifically, each method produces a low-rank decomposition UDV^T , where U is an $n \times k$ orthonormal matrix; we then used the columns of U as features in a logistic regression. We trained the logistic regression on one half of the data and then tested it on the other half, repeating this process over 10,000 random splits. We are in a transductive setting because we used all the data (but not the labels) for learning the topics.

The results, shown in Table 5, suggests that ISA substantially improves the performance of latent semantic analysis for this dataset. In a somewhat surprising twist, we may have expected the decomposition based on correspondence analysis to out-perform the baseline that just divides by document length; however, correspondence analysis ended up doing slightly worse. The problem appears to have been that, because correspondence analysis up-weights less common words relative to the common ones, its topics become more vulnerable to noise. Thus, it is not able to beat document-wise averaging although it has a seemingly better normalization scheme. But, once we use ISA to regularize it, correspondence analysis is able to fully take advantage of the down-weighting of common words.

We can visualize the effect of regularization using Figure 2, which shows the distribution of $\log \|U_i\|_2^2$ for the U -matrices produced by correspondence analysis with and without ISA. The quantity $\|U_i\|_2^2$ measures the importance of the i -th document in learning the topics. We see that plain correspondence analysis has some documents that dominate the resulting U matrix, whereas with ISA the magnitudes of the contributions of different documents are more evenly spread out. Thus, assuming that we do not want topics to be dominated by just a few documents, Figure 2 corroborates our intuition that ISA improves the topics learned by correspondence analysis.

Finally, we note that there exist several topic models that do not reduce to an SVD (e.g., Blei et al., 2003; Hofmann, 2001; Xu et al., 2003); in fact, one of the motivations for latent Dirichlet allocation (Blei et al., 2003) was to add regularization to topic modeling using a hierarchical Bayesian approach. Moreover, methods that only rely on unsupervised topic learning do not in general achieve state-of-the-art accuracy for sentiment classification on their own (e.g., Wang and Manning, 2012). Thus, our goal here is not to advocate an end-to-end methodology for sentiment classification, but only to show that stable autoencoding can substantially improve the quality of topics learned by a simple SVD in cases where a practitioner may want to use them.

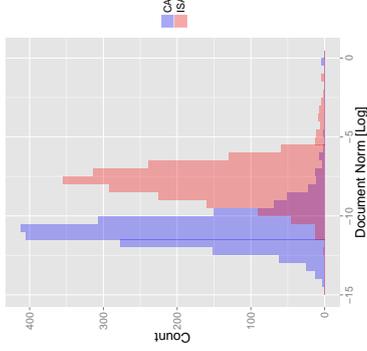


Figure 2: Distribution of $\log \|U_i\|_2^2$ for correspondence analysis with and without ISA. Using ISA increases the influence of the median document in learning topics.

6.2 A sensory analysis of perfumes

Finally, we use stable autoencoding to regularize a sensory analysis of perfumes. The data for the analysis was collected by asking consumers to describe 12 luxury perfumes such as *Chanel Number 5* and *J'adore* with words. The answers were then organized in a 12×39 (39 words unique were used) data matrix where each cell represents the number of times a word is associated to a perfume; a total of $N = 1075$ were used overall. The dataset is available at <http://factominer.free.fr/docs/perfume.txt>. We used correspondence analysis (CA) to visualize the associations between words and perfumes. Here, the technique allows to highlight perfumes that were described using a similar profile of words, and to find words that describe the differences between groups of perfumes.

In order to get a better idea of which regularization method is the most trustworthy here, we ran a small bootstrap simulation study built on top of the perfume dataset. We used the full $N = 1075$ perfume dataset as the population dataset, and then generated samples of size $N = 200$ by subsampling the original dataset without replacement. Then, on each sample, we performed a classical correspondence analysis by performing a rank- k truncated SVD of the matrix M (24), as well as several regularized alternatives described in Section 5.1.

For each estimator, we report its singular values, as well as the RV-coefficients between its row (respectively column) coordinates and the population ones. All the methods except for ISA require us to specify the rank k as an input parameter. Here, of course, k is unknown since we are working with a real dataset; however, examining the full-population dataset suggests that using $k = 2$ components is appropriate. For LN, SA and ISA, we set tuning parameters as in Section 5.2, namely LN uses $\hat{\sigma}$ from (31), while SA is performed with $\delta = 0.5$. For ISA we used $\delta = 0.3$; this latter choice was made to get rank-2 estimates. In practice, one could also consider cross-validation to find a good value for δ .

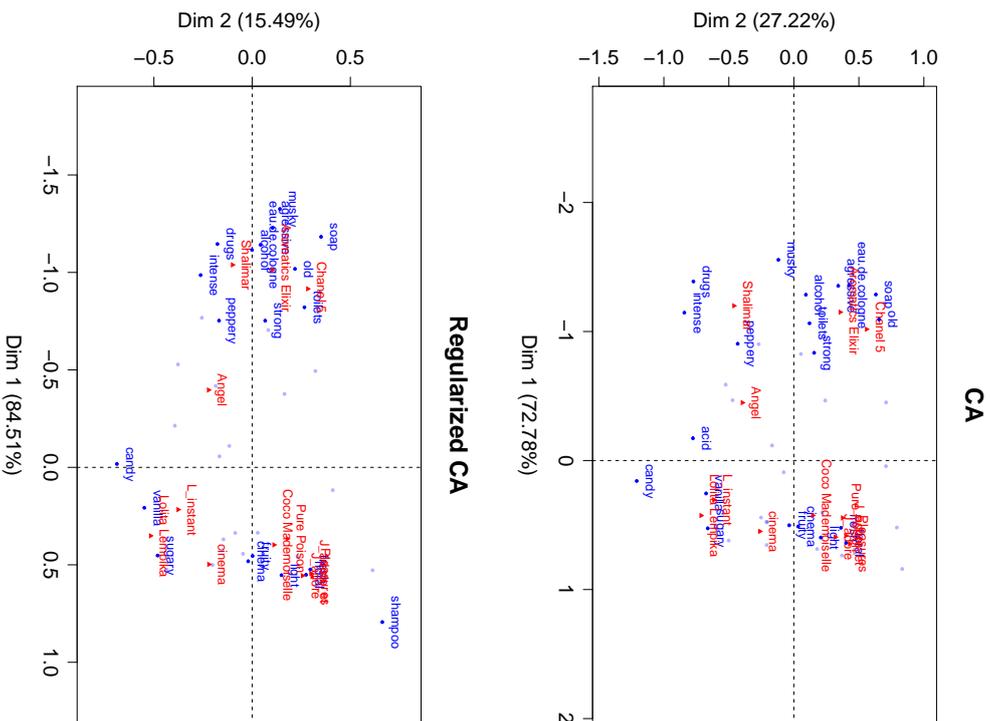


Figure 3: Results for CA on a sample data set (top) and Regularized CA (bottom) using ISA on a single subsample of size $N = 200$. Only the 20 words that contribute the most to the dimensions of variability are represented.

	d_1	d_2	RV _{row}	RV _{col}	k
TRUE	0.44	0.15			
CA	0.62	0.42	0.41	0.72	
LN	0.28	0.11	0.47	0.79	
SA	0.34	0.18	0.50	0.79	
ISA	0.40	0.18	0.52	0.81	2.43

Table 6: Performance of standard correspondence analysis (CA) as well as regularized alternatives on the perfume dataset. We report singular values, RV-coefficients and rank estimates; results correspond to the mean over the 1000 simulations.

Results are shown in Table 6. From a practical point of view, it is also interesting to compare the graphical output of correspondence analysis with and without regularization. Figure 3 (top) shows two-dimensional CA representation on one sample and Figure 3 (bottom) shows the representation obtained with ISA. Only the 20 words that contribute the most to the first two dimensions are represented. The analysis is performed using the R package FactoMineR (Le et al., 2008).

Our results emphasize that, although correspondence analysis is often used as a visualization technique, appropriate regularization is still important, as regularization may substantially affect the graphical output. For example, on the basis of the CA plot, the perfume *Shalimar* looks like an outlier, whereas after regularization it seems to fit in a cluster with *Chanel 5* and *Elixir*. We know from Table 6 that the regularized CA plots are better aligned with the population ones than the unregularized ones are; thus, we may be more inclined to trust insights from the regularized analysis.

7. Discussion

In this paper, we introduced a new framework for low-rank matrix estimation that works by transforming noise models into regularizers via a bootstrap scheme. Our method can adapt to non-isotropic noise structures, thus enabling it to substantially outperform its competitors on problems with, e.g., Poisson noise.

At a high level, our framework works by creating pseudo-datasets \tilde{X} from X using the bootstrap distribution $\tilde{\mathcal{L}}(X)$. If two pseudo-datasets \tilde{X}_1 and \tilde{X}_2 are both likely given $\tilde{\mathcal{L}}(X)$, then we want the induced mean estimates $\hat{\mu}_n^1 = \tilde{X}_1 \hat{B}_n$ and $\hat{\mu}_n^2 = \tilde{X}_2 \hat{B}_n$ to be close to each other. The stable autoencoder (6) enables us to turn this intuition into a concrete regularizer by using the Lévy bootstrap. It remains to be seen whether this idea of regularization via bootstrapping pseudo-datasets can be extended to other classes of low-rank matrix algorithms, e.g., those discussed by Collins et al. (2001), de Leeuw (2006), or Udell et al. (2014).

8. Appendix: Proofs

8.1 Proof of Theorem 1

We begin by establishing the equivalence between (6) and (15). By bias-variance decomposition, we can check that

$$\begin{aligned} \mathbb{E}_{\varepsilon_{ij} \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2)} \left[\|X - (X + \varepsilon)B\|_2^2 \right] &= \|X - XB\|_2^2 + \mathbb{E}_\varepsilon \left[\|\varepsilon B\|_2^2 \right] \\ &= \|X - XB\|_2^2 + \sum_{i,j,k} \text{Var}[\varepsilon_{ij}] B_{jk}^2 \\ &= \|X - XB\|_2^2 + n \frac{\delta \sigma^2}{1 - \delta} \|B\|_2^2, \end{aligned}$$

and so the two objectives are equivalent.

To show that $\hat{\mu}_k^{\text{stable}}$ can be written as (16), we solve for $\hat{\mu}_k^{\text{stable}} = X\hat{B}_k$ explicitly, where

$$\hat{B}_k = \text{argmin}_B \left\{ \|X - XB\|_2^2 + \lambda \|B\|_2^2 : \text{rank}(B) \leq k \right\}.$$

Let $X = UDV^\top$ be the SVD of X . For any matrix M of the same dimension as D , $\|UMV^\top\|_2^2 = \|M\|_2^2$. Thus, we can equivalently write the problem

$$\hat{B}_k = V\hat{Q}_kV^\top, \text{ where } \hat{Q}_k = \text{argmin}_Q \left\{ \|D - DQ\|_2^2 + \lambda \|Q\|_2^2 : \text{rank}(Q) \leq k \right\}.$$

Now, because D is diagonal, $(DQ)_{ij} = D_{ii}Q_{ij}$. Thus, we conclude that $\hat{Q}_{ij} = 0$ for all $i \neq j$, while the problem separates for all the diagonal terms. Without the rank constraint on Q , we find that the diagonal terms \hat{Q}_{ii} are given by

$$\hat{Q}_{ii} = \text{argmin}_{Q_{ii}} \left\{ (1 - Q_{ii})^2 D_{ii}^2 + \lambda Q_{ii}^2 \right\} = \frac{D_{ii}^2}{\lambda + D_{ii}^2}.$$

Meanwhile, we can check that adding the rank constraint amounts to zeroing out all but the k largest of the \hat{Q}_{ii} . Thus, plugging this into our expression of $\hat{\mu}$, we get that

$$\hat{\mu}_k^{\text{stable}} = \sum_{i=1}^k \frac{U_i}{1 + \lambda/D_{ii}^2} V_i^\top.$$

8.2 Proof of Theorem 2

We start by showing that \hat{B} is the solution to the unconstrained version of (17). Let V be a matrix defined by

$$V_{ij} = \text{Var}_{\tilde{X} \sim \tilde{\mathcal{L}}(X)} \left[\tilde{X}_{ij} \right].$$

Because \tilde{X} has mean X , we can check that

$$\mathbb{E}_{\tilde{X} \sim \tilde{\mathcal{L}}(X)} \left[\left\| X - \tilde{X}B \right\|_2^2 \right] = \|X - XB\|_2^2 + \sum_{i,j,k} V_{ij} B_{jk}^2.$$

Thus,

$$\frac{1}{2} \frac{\partial}{\partial B_{jk}} \mathbb{E}_{\tilde{X} \sim \tilde{\mathcal{L}}(X)} \left[\left\| X - \tilde{X}B \right\|_2^2 \right] = - \sum_i X_{ij} (X - XB)_{ik} + \sum_i V_{ij} B_{jk}.$$

Setting gradients to zero, we find an equilibrium

$$X^\top X = X^\top XB + SB, \text{ where } S_{jk} = \begin{cases} \sum_{i=1}^n V_{ij} & \text{for } j = k, \\ 0 & \text{else.} \end{cases}$$

Thus, we conclude that

$$\hat{B} = (X^\top X + S)^{-1} X^\top X$$

is in fact the solution to (17) without the rank constraint.

Next, we show how we can get from \hat{B} to \hat{B}_k using (19). For any matrix B , we can verify by quadratic expansion that

$$\begin{aligned} \|X - XB\|_2^2 &= \|X - X\hat{B}\|_2^2 + \|X(\hat{B} - B)\|_2^2 + 2 \text{tr} \left((X - X\hat{B})^\top (X\hat{B} - XB) \right) \\ &= \|X - X\hat{B}\|_2^2 + \|X(\hat{B} - B)\|_2^2 \\ &\quad + 2 \text{tr} \left((X^\top X (X^\top X + S)^{-1} X^\top X - X^\top X) (B - \hat{B}) \right) \end{aligned}$$

Meanwhile,

$$\begin{aligned} \|S^{\frac{1}{2}}B\|_2^2 &= \|S^{\frac{1}{2}}\hat{B}\|_2^2 + \|S^{\frac{1}{2}}(B - \hat{B})\|_2^2 + 2 \text{tr} \left(\hat{B}^\top S (B - \hat{B}) \right) \\ &= \|S^{\frac{1}{2}}\hat{B}\|_2^2 + \|S^{\frac{1}{2}}(B - \hat{B})\|_2^2 + 2 \text{tr} \left(X^\top X (X^\top X + S)^{-1} S (B - \hat{B}) \right). \end{aligned}$$

Summing everything together, we find that

$$\|X - XB\|_2^2 + \|S^{\frac{1}{2}}B\|_2^2 = \|X(B - \hat{B})\|_2^2 + \|S^{\frac{1}{2}}(B - \hat{B})\|_2^2 + R(\hat{B}, X)$$

where R is a residual term that does not depend on B . Thus, we conclude that

$$\begin{aligned} \hat{B}_k &= \text{argmin}_B \left\{ \|X - XB\|_2^2 + \|S^{\frac{1}{2}}B\|_2^2 : \text{rank}(B) \leq k \right\} \\ &= \text{argmin}_B \left\{ \text{tr} \left((B - \hat{B})^\top (X^\top X + S) (B - \hat{B}) \right) : \text{rank}(B) \leq k \right\}. \end{aligned}$$

As shown in, e.g., Takane (2013), we can solve this last problem by taking the top k terms of the eigendecomposition of $\hat{B}^\top (X^\top X + S)^{-1} \hat{B}$.

8.3 Proof of Theorem 3

For iterates $t = 0, 1, \dots$, define $M_t = \hat{\mu}_t^\top \hat{\mu}_t$. Here, we will show that M_t converges to a fixed point M^* , and that $M^* \preceq X^\top X$; the desired conclusion then follows immediately. First, by construction, we have that

$$M_0 = X^\top X \quad \text{and} \quad M_t = X^\top X \left(X^\top X + S \right)^{-1} X^\top X \left(X^\top X + S \right)^{-1} X^\top X,$$

and so we immediately see that $M_t \preceq M_0$. The general update for M_t is

$$M_{t+1} = g(M_t)^\top g(M_t), \quad \text{where } g(M) = \Sigma^{\frac{1}{2}} (M + S)^{-1} M, \quad (33)$$

where $\Sigma^{\frac{1}{2}}$ is a positive semi-definite solution to $(\Sigma^{\frac{1}{2}})^\top \Sigma^{\frac{1}{2}} = X^\top X$. Now, because matrix inversion is a monotone decreasing function over the positive semi-definite cone and $S \succ 0$, we find that

$$g(M) = \Sigma^{\frac{1}{2}} \left(I - (M + S)^{-1} S \right)$$

is monotone increasing in M over the positive semi-definite cone. In particular

$$\text{if } M_t \leq M_{t-1}, \text{ then } M_{t+1} \leq M_t.$$

By induction, the sequence M_t is monotone decreasing with respect to the positive semi-definite cone order; by standard arguments, it thus follows that this sequence must converge to a limit M^* . Finally, we note that convergence of M_t also implies convergence of $\hat{\mu}_t$, since $\hat{\mu}_{t+1} = X \hat{B}_t$ and \hat{B}_t only depends on $\hat{\mu}_t$ through M_t .

8.4 Proof of Theorem 4

As in the proof of Theorem 3, let $M^* = \hat{\mu}^\top \hat{\mu}$. Because M^* is a fixed point, we know that

$$M^* = M^* (M^* + S)^{-1} X^\top X (M^* + S)^{-1} M^*.$$

Now, because M^* is symmetric with eigenvector u , we can decompose it as

$$M^* = M^\perp + \lambda_u u u^\top, \quad \text{where } \lambda_u = u^\top M^* u \quad \text{and} \quad \|M^\perp u\|_2 = 0.$$

By combining these equalities and using the monotonicity of matrix inversion, we find that

$$\begin{aligned} \lambda_u &= u^\top M^* u \\ &= \lambda_u^2 u^\top (M^* + S)^{-1} X^\top X (M^* + S)^{-1} u \\ &\leq \lambda_u^2 u^\top S^{-1} X^\top X S^{-1} u. \end{aligned}$$

This relation can only hold if $\lambda_u = 0$, or $1 \leq \lambda_u u^\top S^{-1} X^\top X S^{-1} u$, and so our desired conclusion must hold.

8.5 Proof of Proposition 5

First, we note that in the setting (14) with $\delta = 1/2$, we have $S = n\sigma^2 I$. Using Theorem 1, we can verify that the singular vectors of $\hat{\mu}_t$ are the same as those of X for each iterate $t = 1, 2, \dots$ of our algorithm; and so we can write the limit $\hat{\mu}^{\text{iter}}$ as singular value shrinker

$$\hat{\mu}^{\text{iter}} = \sum_{i=1}^{\min\{n, p\}} u_i \psi(d_i) v_i^\top.$$

It remains to derive the form of ψ . Now, using (16), we can verify that the fixed point condition on $\hat{\mu}^{\text{iter}}$ can be expressed in terms of ψ as

$$\psi(d) = \frac{\psi^2(d)}{\lambda + \psi^2(d)} d, \quad \text{with } \lambda = n\sigma^2.$$

This is a cubic equation, with solutions

$$\psi(d) = 0, \quad \text{and} \quad \psi(d) = \frac{1}{2} \left(d \pm \sqrt{d^2 - 4n\sigma^2} \right) \quad \text{for } d^2 \geq 4n\sigma^2.$$

Finally, we can verify that our iterative procedure cannot jump over the largest root, thus resulting in the claimed shrinkage function.

We also check here that, in the case $n = p$, our shrinker $\psi(\cdot)$ is equivalent to the asymptotically optimal shrinker for operator loss $\psi_{\text{opt}}^*(d)$ (Gavish and Donoho, 2014b) which is 0 for $d^2 < 4n\sigma^2$, and else

$$\begin{aligned} \psi_{\text{opt}}(d) &= \frac{1}{\sqrt{2}} \sqrt{d^2 - 2n\sigma^2 + \sqrt{(d^2 - 2n\sigma^2)^2 - 4n^2\sigma^4}} \\ &= \frac{1}{\sqrt{2}} \sqrt{d^2 - 2n\sigma^2 + d \sqrt{d^2 - 4n\sigma^2}}. \end{aligned}$$

Squaring our iterative shrinker, we see that

$$\psi^2(d) = \frac{1}{2} \left(d^2 - 2n\sigma^2 + d \sqrt{d^2 - 4n\sigma^2} \right) = \psi_{\text{opt}}^2(d)$$

for $d^2 \geq 4n\sigma^2$, and 0 else.

Acknowledgment

The authors are grateful for helpful feedback from David Donoho, Bradley Efron, William Fithian and Jean-Philippe Vert, as well as two anonymous referees and the JMLR action editor. Part of this work was performed while J.J. was visiting Stanford University, with support from an AgreenSkills fellowship of the European Union Marie-Curie FP7 COFUND People Programme. S.W. was partially supported by a B.G. and E.J. Eaves Stanford Graduate Fellowship.

References

- Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, 2(1):53–58, 1989.
- Pierre Baldi and Peter Sadowski. The dropout learning algorithm. *Artificial Intelligence*, 210:78–122, 2014.
- J-P Benzécri. *L'analyse des données. Tome II: L'analyse des correspondances*. Dumod, 1973.
- J.-P. Benzécri. Statistical analysis as a tool to emerge patterns from the data. In S. Watanabe (ed), editor, *Methodologies of Pattern recognition*, pages 34–74. New-York, Academic Press, 1986.
- Chris M Bishop. Training with noise is equivalent to Tikhonov regularization. *Neural Computation*, 7(1):108–116, 1995.
- David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.
- Hervé Boullard and Yves Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59(4-5):291–294, 1988.
- Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.
- Andreas Buja and Werner Stuetzle. Observations on bagging. *Statistica Sinica*, pages 323–351, 2006.
- Wray Buntine. Variational extensions to EM and multinomial PCA. In Tapio Elomaa, Heikki Mannila, and Hannu Toivonen, editors, *Machine Learning: ECML 2002*, Lecture Notes in Computer Science, pages 23–34. Springer Berlin Heidelberg, 2002.
- Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.
- Emmanuel J Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.
- Emmanuel J Candès, Carlos A Sing-Long, and Joshua D Trzasko. Unbiased risk estimates for singular value thresholding and spectral estimators. *IEEE Transactions on Signal Processing*, 61(19):4643–4657, 2013.
- Sourav Chatterjee. Matrix estimation by universal singular value thresholding. *The Annals of Statistics*, 43(1):177–214, 2015.
- Michael Collins, Sanjoy Dasgupta, and Robert E. Schapire. A generalization of principal component analysis to the exponential family. In *Advances in Neural Information Processing Systems*. MIT Press, 2001.
- Alexandre d’Aspremont, Francis Bach, and Laurent El Ghaoui. Approximation bounds for sparse principal component analysis. *Mathematical Programming*, pages 1–22, 2012.
- Jan de Leeuw. Principal component analysis of binary data by iterated singular value decomposition. *Computational Statistics and Data Analysis*, 50(1):21–39, 2006.
- Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.
- Rick Durrett. *Probability: Theory and Examples*. Cambridge University Press, 2010.
- Bradley Efron. *Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction*. Cambridge University Press, 2012.
- Bradley Efron and Robert Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC, 1993.
- Yves Escoufier. Le traitement des variables vectorielles. *Biometrics*, 29:751–760, 1973.
- William Fithian and Rahul Mazumder. Scalable convex methods for flexible low-rank matrix modeling. *arXiv preprint arXiv:1308.4211*, 2013.
- Matan Gavish and David L Donoho. The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Transactions on Information Theory*, 60(8), 2014a.
- Matan Gavish and David L Donoho. Optimal shrinkage of singular values. *arXiv:1405.7511v2*, 2014b.
- Amir Globerson and Sam Roweis. Nightmare at test time: Robust learning by feature deletion. In *Proceedings of the International Conference on Machine Learning*, 2006.
- Ian Goodfellow, David Warde-farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1319–1327, 2013.
- L. A. Goodman. The analysis of cross-classified data having ordered and/or unordered categories: Association models, correlation models, and asymmetry models for contingency tables with or without missing entries. *Annals of Statistics*, 13:10–69, 1985.
- Michael J Greenacre. *Theory and Applications of Correspondence Analysis*. Academic Press, 1984.
- Michael J Greenacre. *Correspondence Analysis in Practice, Second Edition*. Chapman & Hall, 2007.
- Hermann O Hirschfeld. A connection between correlation and contingency. *Mathematical Proceedings of the Cambridge Philosophical Society*, 31(4):520–524, 1935.
- Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1-2):177–196, 2001.
- Ian Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *The Annals of Statistics*, 29(2):295–327, 2001.

- Ian Jolliffe. *Principal Component Analysis*. Springer, 2002.
- Ian T Jolliffe, Nickolay T Trendafilov, and Muddassar Uddin. A modified principal component technique based on the lasso. *Journal of Computational and Graphical Statistics*, 12(3): 531–547, 2003.
- Julie Josse and Susan Holmes. Measures of dependence between random vectors and tests of independence. Literature review. *arXiv preprint arXiv:1307.7383*, 2013.
- Julie Josse and François Husson. Selecting the number of components in PCA using cross-validation approximations. *Computational Statistics and Data Analysis*, 56(6):1869–1879, 2011.
- Julie Josse and Sylvain Sardy. Adaptive shrinkage of singular values. *Statistics and Computing*, pages 1–10, 2015.
- Julie Josse, Sylvain Sardy, and Stefan Wäger. **denoiser**: A package for low rank matrix estimation. *arXiv preprint arXiv:1602.01206*, 2016.
- Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.
- Alp Kucukelbir and David M Blei. Population empirical Bayes. *Uncertainty in Artificial Intelligence*, 2015.
- Sébastien Lê, Julie Josse, and François Husson. FactoMineR: An R package for multivariate analysis. *Journal of Statistical Software*, 25(1):1–18, 2008.
- Jeffrey T Leek and John D Storey. Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS genetics*, 3(9):e161, 2007.
- J. Li and D. Tao. Simple exponential family PCA. *IEEE Transactions on Neural Networks and Learning Systems*, 24(3):485–497, 2013.
- Michael Lustig, David L Donoho, Juan M Santos, and John M Pauly. Compressed sensing MRI. *Signal Processing Magazine, IEEE*, 25(2):72–82, 2008.
- Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 2:849–856, 2002.
- Art B Owen and Dean Eckles. Bootstrapping data arrays of arbitrary order. *The Annals of Applied Statistics*, 6(3):895–927, 2012.
- Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the Association for Computational Linguistics*, page 271. Association for Computational Linguistics, 2004.
- Alkes L Price, Nick J Patterson, Robert M Plenge, Michael E Weinblatt, Nancy A Shadick, and David Reich. Principal components analysis corrects for stratification in genome-wide association studies. *Nature Genetics*, 38(8):904–909, 2006.
- Herbert Robbins. *The Empirical Bayes Approach to Statistical Decision Problems*. Springer, 1985.
- Andrey A Shabalin and Andrew B Nobel. Reconstruction of a low-rank matrix in the presence of Gaussian noise. *Journal of Multivariate Analysis*, 118:67–76, 2013.
- Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.
- Patrice Y Simard, Yann A Le Cun, John S Denker, and Bernard Victorri. Transformation invariance in pattern recognition: Tangent distance and propagation. *International Journal of Imaging Systems and Technology*, 11(3):181–197, 2000.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- Yoshio Takane. *Constrained Principal Component Analysis and Related Techniques*. Chapman & Hall, 2013.
- Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, pages 267–288, 1996.
- Madeleine Udell, Corinne Horn, Reza Zadeh, and Stephen Boyd. Generalized low rank models. *arXiv preprint arXiv:1410.0342*, 2014.
- Laurens van der Maaten, Minmin Chen, Stephen Tyree, and Kilian Q Weinberger. Learning with marginalized corrupted features. In *Proceedings of the International Conference on Machine Learning*, 2013.
- Marie Verbanck, Julie Josse, and François Husson. Regularised PCA to denoise and visualise data. *Statistics and Computing*, pages 1–16, 2013.
- Stefan Wäger, Sida Wang, and Percy Liang. Dropout training as adaptive regularization. In *Advances in Neural Information Processing Systems*, pages 351–359, 2013.
- Stefan Wäger, William Fithian, Sida Wang, and Percy Liang. Altitude training: Strong bounds for single-layer dropout. In *Advances in Neural Information Processing Systems*, volume 27, pages 100–108, 2014.
- Stefan Wäger, William Fithian, and Percy Liang. Data augmentation via Lévy processes. *arXiv preprint arXiv:1603.06340*, 2016.
- Sida Wang and Christopher D Manning. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the Association for Computational Linguistics*, pages 90–94. Association for Computational Linguistics, 2012.

- Sida I Wang, Mengqiu Wang, Stefan Wager, Percy Liang, and Christopher D Manning. Feature noising for log-linear structured prediction. In *Empirical Methods in Natural Language Processing*, 2013.
- Daniela M Witten, Robert Tibshirani, and Trevor Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 2009.
- Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 267–273. ACM, 2003.
- C. H. Zhang and J. Huang. The sparsity and bias of the lasso selection in high-dimensional linear regression. *The Annals of Statistics*, 36(4):1567–1594, 2008.
- H. Zou. The adaptive LASSO and its oracle properties. *Journal of the American Statistical Association*, 101:1418–1429, 2006.
- H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.

Bayesian Optimization for Likelihood-Free Inference of Simulator-Based Statistical Models

Michael U. Gutmann

Helsinki Institute for Information Technology HIIT

Department of Mathematics and Statistics, University of Helsinki

Department of Information and Computer Science, Aalto University

MICHAEL.GUTMANN@HELSINKI.FI

Jukka Corander

Helsinki Institute for Information Technology HIIT

Department of Mathematics and Statistics, University of Helsinki

JUKKA.CORANDER@HELSINKI.FI

Editor: Nando de Freitas

Abstract

Our paper deals with inferring simulator-based statistical models given some observed data. A simulator-based model is a parametrized mechanism which specifies how data are generated. It is thus also referred to as generative model. We assume that only a finite number of parameters are of interest and allow the generative process to be very general; it may be a noisy nonlinear dynamical system with an unrestricted number of hidden variables. This weak assumption is useful for devising realistic models but it renders statistical inference very difficult. The main challenge is the intractability of the likelihood function. Several likelihood-free inference methods have been proposed which share the basic idea of identifying the parameters by finding values for which the discrepancy between simulated and observed data is small. A major obstacle to using these methods is their computational cost. The cost is largely due to the need to repeatedly simulate data sets and the lack of knowledge about how the parameters affect the discrepancy. We propose a strategy which combines probabilistic modeling of the discrepancy with optimization to facilitate likelihood-free inference. The strategy is implemented using Bayesian optimization and is shown to accelerate the inference through a reduction in the number of required simulations by several orders of magnitude.

Keywords: intractable likelihood, latent variables, Bayesian inference, approximate Bayesian computation, computational efficiency

1. Introduction

We consider the statistical inference of a finite number of parameters of interest $\theta \in \mathbb{R}^d$ of a simulator-based statistical model for observed data \mathbf{y}_0 which consist of m possibly dependent data points. A simulator-based statistical model is a parametrized stochastic data generating mechanism. Formally, it is a family of probability density functions (pdfs) $\{p_{\mathbf{y}|\theta}\}_{\theta}$ of unknown analytical form which allow for exact sampling of data $\mathbf{y}\theta \sim p_{\mathbf{y}|\theta}$. In practical terms, it is a computer program which takes a value of θ and a state of the random number generator as input and returns data $\mathbf{y}\theta$ as output. Simulator-based models are also called implicit models because the pdf of $\mathbf{y}\theta$ is not specified explicitly (Diggle and Gratton, 1984), or generative models because they specify how data are generated.

Simulator-based models are useful because they interface easily with models typically encountered in the natural sciences. In particular, hypotheses of how the observed data \mathbf{y}_0 were generated can be implemented without making excessive compromises in order to have an analytically tractable model pdf $p_{\mathbf{y}|\theta}$.

Since the analytical form of $p_{\mathbf{y}|\theta}$ is unknown, inference using the likelihood function $\mathcal{L}(\theta)$,

$$\mathcal{L}(\theta) = p_{\mathbf{y}|\theta}(\mathbf{y}_0|\theta), \quad (1)$$

is not possible. The likelihood function is also not available for a large class of other statistical models which are known as unnormalized models. In these models, $p_{\mathbf{y}|\theta}$ is only known up to a normalizing scaling factor (the partition function) which guarantees that $p_{\mathbf{y}|\theta}$ is a valid pdf for all values of θ . Simulator-based models differ from unnormalized models in that not only is the scaling factor unknown but also the shape of $p_{\mathbf{y}|\theta}$. Likelihood-free inference methods developed for unnormalized models (for example Hinton, 2002; Hyvärinen, 2005; Pihlaja et al., 2010; Gutmann and Hirayama, 2011; Gutmann and Hyvärinen, 2012) are thus not applicable to simulator-based models.

For simulator-based models, likelihood-free inference methods have emerged in multiple disciplines. “Indirect inference” originated in economics (Gouriéroux et al., 1993), “approximate Bayesian computation” (ABC) in genetics (Beaumont et al., 2002; Marjoram et al., 2003; Sisson et al., 2007), or the “synthetic likelihood” approach in ecology (Wood, 2010), for an overview, see, for example, the review by Hartig et al. (2011). The different methods share the basic idea to identify the model parameters by finding values which yield simulated data that resemble the observed data.

The generality of simulator-based models comes with the expense of two major difficulties in the inference. One difficulty is the assessment of the discrepancy between the observed and simulated data (Joyce and Marjoram, 2008; Wegmann et al., 2009; Nunes and Balding, 2010; Fearnhead and Prangle, 2012; Aeschbacher et al., 2012; Gutmann et al., 2014). The other difficulty is that the inference methods tend to be slow due to the need to simulate a large collection of data sets and due to the lack of knowledge about the relation between the model parameters and the corresponding discrepancies.

In this paper, we address the computational difficulty of the likelihood-free inference methods. We propose a strategy which combines probabilistic modeling of the discrepancies with optimization to facilitate likelihood-free inference. The strategy is implemented using Bayesian optimization (see, for example, Brochu et al., 2010). We show that using Bayesian optimization in likelihood-free inference (BOLFI) can reduce the number of required simulations by several orders of magnitude, which accelerates the inference substantially.¹

The rest of the paper is organized as follows: In Section 2, we present examples of simulator-based statistical models to help clarify their properties. In Section 3, we provide a unified review of existing inference methods for simulator-based models, and use the examples to point out computational issues. The computational difficulties are summarized in Section 4, and a framework to address them is outlined in Section 5. Section 6 implements

1. Preliminary results were presented at “Approximate Bayesian Computation in Rome”, 2013, and MCMCski IV, 2014, as a poster “Bayesian optimization for efficient likelihood-free inference”, and at the NIPS workshop “ABC in Montreal”, 2014, as part of an oral presentation.

the framework using Bayesian optimization. Applications of the developed methodology are given in Section 7, and Section 8 concludes the paper.

2. Examples of Simulator-Based Statistical Models

We present here three examples of simulator-based statistical models. The first example is an artificial one, but useful because it allows us to illustrate the central concepts. The other two are examples from real data analysis with intractable models (Wood, 2010; Numminen et al., 2013). The examples will be used throughout the paper and the model details can be looked up here when needed.

Example 1 (Normal distribution). A standard way to sample data $\mathbf{y}_\theta = (y_\theta^{(1)}, \dots, y_\theta^{(n)})$ from a normal distribution with mean θ and variance one is to sample n standard normal random variables $\boldsymbol{\omega} = (\omega^{(1)}, \dots, \omega^{(n)})$ and to add θ to the obtained samples,

$$\mathbf{y}_\theta = \theta + \boldsymbol{\omega}, \quad \boldsymbol{\omega} \sim \mathcal{N}(0, \mathbf{I}_n). \quad (2)$$

The symbol $\mathcal{N}(0, \mathbf{I}_n)$ denotes a n -variate normal distribution with mean zero and identity covariance matrix. After sampling of the random quantities $\boldsymbol{\omega}$, the observed data \mathbf{y}_θ are a deterministic transformation of $\boldsymbol{\omega}$ and the parameter θ . For more general simulators, the same principle applies. In particular, the data \mathbf{y}_θ are a deterministic transformation of θ if the random quantities are kept fixed, for example by fixing the seed of the random number generator.

Example 2 (Ricker model). In this example, the simulator consists of a latent stochastic time series and an observation model. The latent time series is a stochastic version of the Ricker map which is a classical model in ecology (Ricker, 1954). The stochastic version can be described as a nonlinear autoregressive model,

$$\log N^{(t)} = \log r + \log N^{(t-1)} - N^{(t-1)} + \sigma e^{(t)}, \quad t = 1, \dots, n, \quad N^{(0)} = 0, \quad (3)$$

where $N^{(t)}$ is the size of some animal population at time t and the $e^{(t)}$ are independent standard normal random variables. The latent time series has two parameters: $\log r$ which is related to the log growth rate and σ for the standard deviation of the innovations. A Poisson observation model is assumed, such that given $N^{(t)}$, $y_\theta^{(t)}$ is drawn from a Poisson distribution with mean $\varphi N^{(t)}$,

$$y_\theta^{(t)} | N^{(t)}, \varphi \sim \text{Poisson}(\varphi N^{(t)}), \quad (4)$$

where φ is a scaling parameter. The model is thus in total parametrized by $\boldsymbol{\theta} = (\log r, \sigma, \varphi)$. Figure 1(a) shows example data generated from the model. Inference of $\boldsymbol{\theta}$ is difficult because the $N^{(t)}$ are not directly observed and because of the strong nonlinearity in the autoregressive model. Wood (2010) used this example to illustrate his “synthetic likelihood” approach to inference.

Example 3 (Bacterial infections in day care centers). The data generating process is here defined via a latent continuous-time Markov chain and an observation model. The model was developed by Numminen et al. (2013) to infer the transmission dynamics of bacterial infections in day care centers.

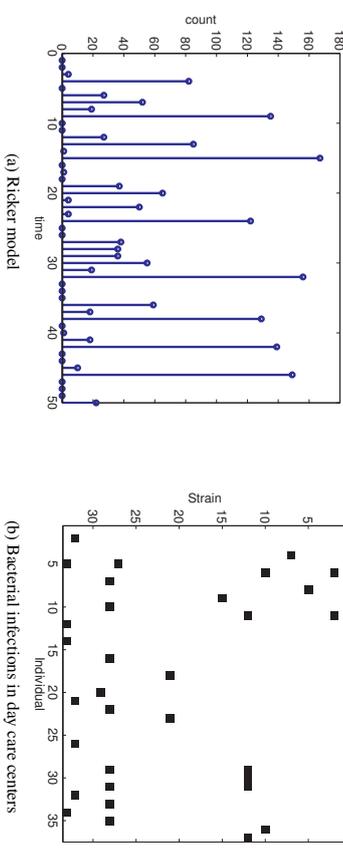


Figure 1: Examples of simulator-based statistical models. (a) Data generated from the Ricker model in Example 2 with $n = 50$ and $\boldsymbol{\theta}_0 = (\log r_0, \sigma_0, \varphi_0) = (3.8, 0.3, 10)$. (b) Data generated from the model in Example 3 on bacterial infections in day care centers. There are 33 different strains of the bacterium in circulation and $M_{\text{DCC}} = 36$ of the 53 attendees of the day care center were sampled (Numminen et al., 2013). Each black square indicates a sampled attendee who is infected with a particular strain. The data were generated with $\boldsymbol{\theta}_0 = (\beta_0, \Delta_0, \theta_0) = (3.6, 0.6, 0.1)$.

The variables of the latent Markov chain are the binary indicator variables I_{is}^t which specify whether attendee i of a day care center is infected with the bacterial strain s at time t ($I_{is}^t = 1$), or not ($I_{is}^t = 0$). Starting with zero infected individuals, $I_{is}^0 = 0$ for all i and s , the states evolve in a stochastic manner according to the rate equations

$$P(I_{is}^{t+h} = 0 | I_{is}^t = 1) = h + o(h), \quad (5)$$

$$P(I_{is}^{t+h} = 1 | I_{is}^t = 0 \forall s') = R_s(t)h + o(h), \quad (6)$$

$$P(I_{is}^{t+h} = 1 | I_{is}^t = 0, \exists s' : I_{is}^t = 1) = \theta R_s(t)h + o(h), \quad (7)$$

where h is a small time interval and $R_s(t)$ the rate of infection with strain s at time t . The three equations model the probability to clear a strain s during time t and $t+h$ (Equation 5), the probability to be infected with a strain s if not colonized by other strains (Equation 6), and the probability to be infected if colonized with other strains (Equation 7). The rate of infection is a weighted combination of the probability P_s for an infection happening outside the day care center and the probability $E_s(t)$ for an infection from within,

$$R_s(t) = \beta E_s(t) + \Delta P_s. \quad (8)$$

We refer the reader to the original publication by Numminen et al. (2013) for more details and the expression for $E_s(t)$. The observation model was random sampling of M_{DCC} individuals without replacement from all the individuals attending a day care center at some sufficiently large random time (endemic situation). The model has three parameters

$\theta = (\beta, \Lambda, \theta)$: the internal infection parameter β , the external infection parameter Λ , and the co-infection parameter θ . Figure 1(b) shows an example of data generated from the model.

Numminen et al. (2013) applied the model to data on colonizations with the bacterium *Streptococcus pneumoniae*. The observed data \mathbf{y}_o were the states of the sampled attendees of 29 day care centers, that is, 29 binary matrices as in Figure 1(b) but with varying numbers of sampled attendees per day care center. Inference of the parameters is difficult because the data are a snapshot of the state of some of the attendees at a single time point only. Since the process evolves in continuous-time, the modeled system involves infinitely many correlated unobserved variables. \blacktriangle

3. Inference Methods for Simulator-Based Statistical Models

This section organizes the foundations and the previous work. We first point out properties common to all inference methods for simulator-based models, one being the general manner of constructing approximate likelihood functions. We then explain parametric and nonparametric approximations of the likelihood and discuss the relation between the two approaches. This is followed by a summary of currently used posterior inference schemes.

3.1 General Properties of the Different Inference Methods

Inference of simulator-based statistical models is generally based on some measurement of discrepancy $\Delta\theta$ between the observed data \mathbf{y}_o and data \mathbf{y}_θ simulated with parameter value θ . The discrepancy is used to define an approximation $L(\theta)$ of the likelihood $\mathcal{L}(\theta)$. The approximation happens on multiple levels.

On a statistical level, the approximation consists of reducing the observed data \mathbf{y}_o to some features, or summary statistics Φ_o , before performing inference. The purpose of the summary statistics is to reduce the dimensionality and to filter out information which is not deemed relevant for the inference of θ . That is, in this first approximation, the likelihood $\mathcal{L}(\theta)$ is replaced with $L(\theta)$,

$$L(\theta) = p_{\Phi|\theta}(\Phi_o|\theta), \tag{9}$$

where $p_{\Phi|\theta}$ is the pdf of the summary statistics. The function $L(\theta)$ is a valid likelihood function, but for the inference of θ given Φ_o , and not for the inference of θ given \mathbf{y}_o , in contrast to $\mathcal{L}(\theta)$, unless the chosen summary statistics happened to be sufficient in the standard statistical sense.

The likelihood function $L(\theta)$, however, is also not known, because the pdf $p_{\Phi|\theta}$ is of unknown analytical form, which is a property inherited from $p_{\mathbf{y}|\theta}$. Thus, $L(\theta)$ needs to be approximated by some method. We denote practical approximations obtained with finite computational resources by $\hat{L}(\theta)$. Limiting approximations if infinitely many computational resources were available will be denoted by $\tilde{L}(\theta)$.

In the paper, we will encounter several methods to construct $\hat{L}(\theta)$. They all base the approximation on simulated summary statistics Φ_θ , generated with parameter value θ . The simulation of summary statistics is generally done by simulating a data set \mathbf{y}_θ , followed by its reduction to summary statistics. Table 1 provides an overview of the different “likelihoods” appearing in the paper.

Symbol	Meaning	Definition
\mathcal{L}	true likelihood based on observed data	Eq (1)
L	true likelihood based on summary statistics	Eq (9)
\tilde{L}	approximation of L requiring infinite computing power	Sec 3.1
$\tilde{L}_s(\tilde{\ell}_s)$	parametric approx/synthetic (log) likelihood	Eq (13)
\tilde{L}_κ	nonparametric approx with kernel κ	Eq (22)
\tilde{L}_u	nonparametric approx with uniform kernel	Eq (25)
\hat{L}	computable approximation of L	Sec 3.1
$\hat{L}_s^N(\hat{\ell}_s^N)$	parametric approx/synthetic (log) likelihood with sample averages	Eq (15)
\hat{L}_κ^N	nonparametric approx with kernel κ and sample averages	Eq (21)
\hat{L}_u^N	nonparametric approx with uniform kernel and sample averages	Eq (24)
$\hat{L}_s^{(t)}(\hat{\ell}_s^{(t)})$	parametric approx/synthetic (log) likelihood with regression	Sec 5.2
$\hat{L}_\kappa^{(t)}$	nonparametric approx with kernel κ and regression	Sec 5.3
$\hat{L}_u^{(t)}$	nonparametric approx with uniform kernel and regression	Sec 5.3

Table 1: The main (approximate) likelihood functions appearing in the paper. The superscript “ N ” indicates that the sample average is computed using N simulated data sets per model parameter θ . The superscript “ (t) ” indicates that regression is performed with a training set containing t simulated data sets. The parametric approximations will be used together with the Gaussian and the Ricker model, the nonparametric approximations together with the Gaussian and the day care center model.

After construction of \hat{L} , inference can be performed in the usual manner by replacing \mathcal{L} with \hat{L} . Approximate posterior inference can be performed via Markov chain Monte Carlo (MCMC) algorithms or via an importance sampling approach (see, for example, Robert and Casella, 2004). The posterior expectation of a function $g(\theta)$ given \mathbf{y}_o can be computed via importance sampling with auxiliary pdf $q(\theta)$,

$$E(g(\theta)|\mathbf{y}_o) \approx \sum_{m=1}^M g(\theta^{(m)})w^{(m)}, \quad w^{(m)} = \frac{\mathcal{L}(\theta^{(m)})p_{\theta}(\theta^{(m)})}{\sum_{i=1}^M \mathcal{L}(\theta^{(i)})p_{\theta}(\theta^{(i)})}, \quad \theta^{(m)} \stackrel{i.i.d.}{\sim} q(\theta), \tag{10}$$

where p_θ denotes the prior pdf. This approach also yields an estimate of the posterior distribution via the “particles” $\theta^{(m)}$ and the associated weights $w^{(m)}$. A computable version is obtained by replacing \mathcal{L} with \hat{L} , giving $E(g(\theta)|\mathbf{y}_o) \approx E(g(\theta)|\Phi_o)$,

$$E(g(\theta)|\Phi_o) \approx \sum_{m=1}^M g(\theta^{(m)})\hat{w}^{(m)}, \quad \hat{w}^{(m)} = \frac{\hat{L}(\theta^{(m)})p_{\theta}(\theta^{(m)})}{\sum_{i=1}^M \hat{L}(\theta^{(i)})p_{\theta}(\theta^{(i)})}, \quad \theta^{(m)} \stackrel{i.i.d.}{\sim} q(\theta). \tag{11}$$

There is some flexibility in the choice of the auxiliary pdf $q(\theta)$ in Equations (10) and (11) which enables iterative adaptive algorithms where the accepted $\theta^{(m)}$ of one iteration are used to define the auxiliary distribution $q(\theta)$ of the next iteration (population or sequential Monte Carlo algorithms; Cappé et al., 2004; Del Moral et al., 2006).

3.2 Parametric Approximation of the Likelihood

The pdf $p_{\Phi\theta}$ of the summary statistics is of unknown analytical form but it may be reasonably assumed that it belongs to a certain parametric family. For instance, if Φ_θ is obtained via averaging, the central limit theorem suggests that the pdf may be well approximated by a Gaussian distribution if the number of samples n is sufficiently large,

$$p_{\Phi\theta}(\phi|\theta) \approx \frac{1}{(2\pi)^{p/2} |\det \Sigma_\theta|^{1/2}} \exp\left(-\frac{1}{2}(\phi - \mu_\theta)^\top \Sigma_\theta^{-1}(\phi - \mu_\theta)\right), \quad (12)$$

where p is the dimension of Φ_θ . The corresponding likelihood function is $\hat{L}_s = \exp(\hat{\ell}_s)$,

$$\hat{\ell}_s(\theta) = -\frac{p}{2} \log(2\pi) - \frac{1}{2} \log |\det \Sigma_\theta| - \frac{1}{2} (\Phi_o - \mu_\theta)^\top \Sigma_\theta^{-1} (\Phi_o - \mu_\theta), \quad (13)$$

which is an approximation of $L(\theta)$ unless the summary statistics are indeed Gaussian. The mean μ_θ and the covariance matrix Σ_θ are generally not known. But the simulator can be used to estimate them via a sample average E^N over N independently generated summary statistics,

$$\hat{\mu}_\theta = E^N[\Phi_\theta] = \frac{1}{N} \sum_{i=1}^N \Phi_\theta^{(i)}, \quad \Phi_\theta^{(i)} \stackrel{i.i.d.}{\sim} p_{\Phi\theta}, \quad \hat{\Sigma}_\theta = E^N[(\Phi_\theta - \hat{\mu}_\theta)(\Phi_\theta - \hat{\mu}_\theta)^\top]. \quad (14)$$

A computable estimate \hat{L}_s^N of the likelihood function $L(\theta)$ is then given by $\hat{L}_s^N = \exp(\hat{\ell}_s^N)$,

$$\hat{\ell}_s^N(\theta) = -\frac{p}{2} \log(2\pi) - \frac{1}{2} \log |\det \hat{\Sigma}_\theta| - \frac{1}{2} (\Phi_o - \hat{\mu}_\theta)^\top \hat{\Sigma}_\theta^{-1} (\Phi_o - \hat{\mu}_\theta). \quad (15)$$

This approximation was named synthetic likelihood (Wood, 2010), hence our subscript “s”. Due to the approximation of the expectation with a sample average, $\hat{\ell}_s^N$ is a stochastic process (a random function). We illustrate this in Example 4 below. We there also show that the number of simulated summary statistics (data sets) N is a trade-off parameter: The computational cost decreases as N decreases but the variability of the estimate increases as a consequence. It further turns out that the sample curves of $\hat{\ell}_s^N$ may not be smooth for finite N and that decreasing N may worsen their roughness. We illustrate this in Example 5 using the Ricker model.

Example 4 (Synthetic likelihood for the mean of a normal distribution). The sample average is a sufficient statistic for the task of inferring the mean θ from a sample $\mathbf{y}_o = (y_o^{(1)}, \dots, y_o^{(n)})$ of a normal distribution with assumed variance one. We thus reduce the observed and simulated data \mathbf{y}_o and \mathbf{y}_θ to the empirical means Φ_o and Φ_θ , respectively,

$$\Phi_o = \frac{1}{n} \sum_{i=1}^n y_o^{(i)}, \quad \Phi_\theta = \frac{1}{n} \sum_{i=1}^n y_\theta^{(i)}. \quad (16)$$

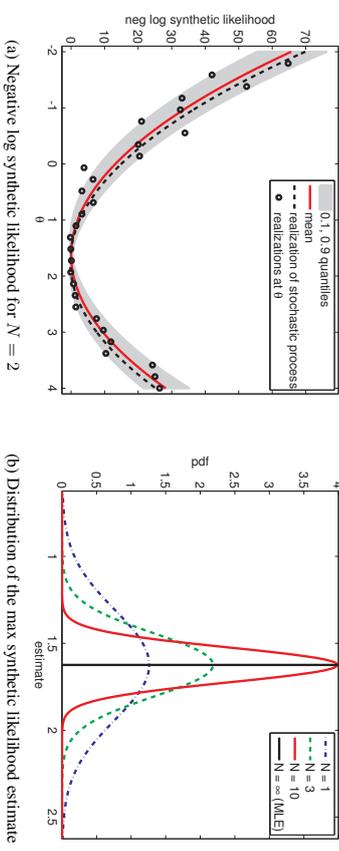


Figure 2: Estimation of the mean of a Gaussian. (a) The figure shows the negative log synthetic likelihood $-\hat{\ell}_s^N$. It illustrates that $\hat{\ell}_s^N$ is a random function. (b) The randomness makes the estimate $\hat{\theta} = \operatorname{argmax}_\theta \hat{\ell}_s^N(\theta)$ a random variable. Its variability increases as N decreases.

In this special case, no information is lost with the reduction to the summary statistic, that is, $L(\theta) \propto \mathcal{L}(\theta)$. Furthermore, the distribution of the summary statistic Φ_θ is here known, $\Phi_\theta \sim \mathcal{N}(\theta, 1/n)$ so that the Gaussian model assumption holds and $L_s(\theta) = L(\theta)$.

Using for simplicity the true variance of Φ_θ , we have $\hat{\ell}_s^N(\theta) = -1/2 \log(2\pi/n) - n/2(\Phi_o - \hat{\mu}_\theta)^2$. Since $\hat{\mu}_\theta$ is an average of N realizations of Φ_θ , $\hat{\mu}_\theta \sim \mathcal{N}(\theta, 1/(nN))$, and we can write $\hat{\ell}_s^N$ as a quadratic function subject to a random shift g ,

$$\hat{\ell}_s^N(\theta) = -\frac{1}{2} \log\left(\frac{2\pi}{n}\right) - \frac{n}{2} (\Phi_o - \theta - g)^2, \quad g \sim \mathcal{N}\left(0, \frac{1}{nN}\right). \quad (17)$$

Each realization of g yields a different mapping $\theta \rightarrow \hat{\ell}_s^N$ which illustrates that the (log) synthetic likelihood is a random function. Figure 2(a) shows the 0.1 and 0.9 quantiles of $-\hat{\ell}_s^N$ for $N=2$. The dashed curve visualizes $\theta \rightarrow -\hat{\ell}_s^N$ for a fixed realization of g . The circles show values of $-\hat{\ell}_s^N(\theta)$ when g is not kept fixed as θ changes. The results are for sample size $n=10$.

The optimizer $\hat{\theta}$ of each realization of $\hat{\ell}_s^N$ depends on g , $\hat{\theta} = \Phi_o - g$. That is, $\hat{\theta}$ is a random variable with distribution $\mathcal{N}(\Phi_o, 1/(Nn))$. In the limit of an infinite amount of available computational resources, that is $N \rightarrow \infty$, g equals zero, and the distribution has a point-mass at $\hat{\theta}_{\text{MLE}} = \Phi_o$ which is indicated with the black vertical line in Figure 2(b). As N decreases, variance is added to the point-estimate $\hat{\theta}$. This added variability is due to the use of finite computational resources; it does not reflect uncertainty about θ due to the finite sample size n . The variability causes an inflation of the mean squared estimation error by a factor of $(1 + 1/N)$, $E((\hat{\theta} - \theta_o)^2) = 1/n(1 + 1/N)$. \blacktriangle

Example 5 (Synthetic likelihood for the Ricker model). Wood (2010) used the synthetic likelihood to perform inference of the Ricker model and other simulator-based models with

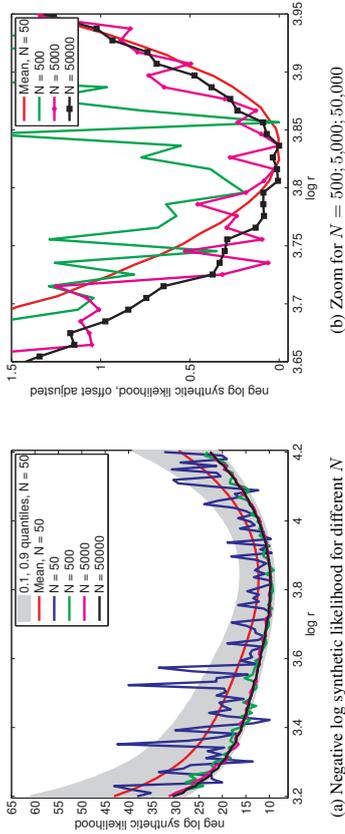


Figure 3: Using less computational resources may reduce the smoothness of the approximate likelihood function. The figures show the negative log synthetic likelihood $-\hat{\ell}_s^N$ for the Ricker model. Only the first parameter ($\log r$) was varied, the others were kept fixed at the data generating values. (a) The use of simulations makes the synthetic likelihood a stochastic process. Realizations of $-\hat{\ell}_s^N$ for different N are shown together with the variability for $N = 50$. (b) The curves become more and more smooth as the number N of simulated data sets increases even though the curve for $N = 50,000$ is still rugged. It is reasonable to assume though that the limit for $N \rightarrow \infty$ is smooth.

complex dynamics. Time series data $\mathbf{y}_\theta = (y_\theta^{(T_0+1)}, \dots, y_\theta^{(T_0+n)})$ from the Ricker model after some “burn-in” time T_0 were summarized in the form of the coefficients of the autocorrelation function and the coefficients of fitted nonlinear autoregressive models, thereby reducing the data to fourteen summary statistics Φ_θ (see the supplementary material of Wood, 2010, for their exact definition).

Figure 3 shows the negative log synthetic likelihood $-\hat{\ell}_s^N$ for the Ricker model as a function of the log growth rate $\log r$ for \mathbf{y}_o in Figure 1(a). The parameters σ and φ were kept fixed at the values $\sigma_o = 0.3$ and $\varphi_o = 10$ which we used to generate \mathbf{y}_o ($\log r^o$ was 3.8). The figures show that the realizations of the synthetic likelihood become less smooth as N decreases.

The lack of smoothness makes the minimization of the different realizations of $-\hat{\ell}_s^N$ difficult. A grid-search is feasible for very large N but this approach does not scale to higher dimensions. Gradient-based optimization is tricky because the functional form of $\hat{\ell}_s^N$ is unknown. Finite differences may not yield a reliable approximation of the gradient because of the lack of smoothness. Instead of optimizing a single realization of the objective, one could use an approximate stochastic gradient approach. That is, approximate gradients are computed with different random seeds at different values of the parameter. For small N , however, the gradients are unreliable so that the stepsize has to be very small, which makes the optimization rather costly again. To resolve the issue, we suggest a more efficient approach by combining probabilistic modeling with optimization. \blacktriangleleft

3.3 Nonparametric Approximation of the Likelihood

An alternative to assuming a parametric model for the pdf $p_{\Phi|\theta}$ of the summary statistics is to approximate it by a kernel density estimate (Rosenblatt, 1956; Parzen, 1962; Mack and Rosenblatt, 1979; Wand and Jones, 1995),

$$p_{\Phi|\theta}(\phi|\theta) \approx \mathbb{E}^N [K(\phi, \Phi_\theta)], \quad (18)$$

where K is a suitable kernel and \mathbb{E}^N denotes empirical expectation as before,

$$\mathbb{E}^N [K(\phi, \Phi_\theta)] = \frac{1}{N} \sum_{i=1}^N K(\phi, \Phi_\theta^{(i)}), \quad \Phi_\theta^{(i)} \stackrel{i.i.d.}{\sim} p_{\Phi|\theta}. \quad (19)$$

An approximation of the likelihood function $L(\theta)$ is given by $\hat{L}_K^N(\theta)$,

$$\hat{L}_K^N(\theta) = \mathbb{E}^N [K(\Phi_o, \Phi_\theta)]. \quad (20)$$

We may re-write $K(\Phi_o, \Phi)$ in another form as $\kappa(\Delta_\theta)$ where $\Delta_\theta \geq 0$ depends on Φ_o and Φ_θ , and κ is a univariate non-negative function not depending on θ . The kernels K are generally such that κ has a maximum at zero (the maximum may be not unique though). Taking the empirical expectation in Equation (20) with respect to Δ_θ instead of Φ_θ , we have $\hat{L}_K^N(\theta) = \hat{L}_\kappa^N(\theta)$,

$$\hat{L}_\kappa^N(\theta) = \mathbb{E}^N [\kappa(\Delta_\theta)]. \quad (21)$$

As the number N grows, \hat{L}_κ^N converges to \bar{L}_κ ,

$$\bar{L}_\kappa(\theta) = \mathbb{E}[\kappa(\Delta_\theta)], \quad (22)$$

which is \hat{L}_κ^N where the empirical average \mathbb{E}^N is replaced by the expectation \mathbb{E} . The limiting approximate likelihood $\bar{L}_\kappa(\theta)$ does not necessarily equal the likelihood $L(\theta) = p_{\Phi|\theta}(\Phi_o|\theta)$. For example, if $\kappa(\Delta_\theta)$ is obtained from a translation invariant kernel K , that is, $\kappa(\Delta_\theta) = K(\Phi_o - \Phi_\theta)$, \bar{L}_κ is the likelihood for a summary statistics whose pdf is obtained by convolving $p_{\Phi|\theta}$ with K .

For convex functions κ , Jensen’s inequality yields a lower bound for \hat{L}_κ^N and its logarithm,

$$\hat{L}_\kappa^N(\theta) \geq \kappa(\hat{J}^N(\theta)), \quad \log \hat{L}_\kappa^N(\theta) \geq \log \kappa(\hat{J}^N(\theta)), \quad \hat{J}^N(\theta) = \mathbb{E}^N [\Delta_\theta]. \quad (23)$$

Since κ is maximal at zero, the lower bound is maximized by minimizing the conditional empirical expectation $\hat{J}^N(\theta)$. The advantage of the lower bound is that it can be maximized irrespective of κ , which is often difficult to choose in practice.

A popular choice of κ for likelihood-free inference is the uniform kernel $\kappa = \kappa_u$ which yields the approximate likelihood \hat{L}_u^N ,

$$\kappa_u(u) = c \chi_{[0,h)}(u), \quad \hat{L}_u^N(\theta) = c \mathbb{P}^N(\Delta_\theta < h), \quad (24)$$

where the indicator function $\chi_{[0,h)}(u)$ equals one if $u \in [0, h)$ and zero otherwise. The scaling parameter c does not depend on θ , and the positive scalar h is the bandwidth of

the kernel and acts as acceptance/rejection threshold. The approximate likelihood \hat{L}_n^N is proportional to the empirical probability that the discrepancy is below the threshold. The limiting approximate likelihood is denoted by $\bar{L}_n(\theta)$,

$$\bar{L}_n(\theta) = c \mathbb{P}(\Delta_\theta < h). \quad (25)$$

The lower bound for convex κ is not applicable but we can obtain an equivalent bound by Markov's inequality,

$$\hat{L}_n^N(\theta) = c \left[1 - \mathbb{P}^N(\Delta_\theta^2 \geq h) \right] \geq c \left[1 - \frac{1}{h} \mathbb{E}^N[\Delta_\theta] \right]. \quad (26)$$

The lower bound of the approximate likelihood can be maximized by minimizing $\hat{J}^N(\theta)$ as for convex κ .

We illustrate the approximation of the likelihood via \hat{L}_n^N in Example 6 below. It is pointed out that good approximations are computationally very expensive because of the very small probability for Δ_θ to be below small thresholds h , or, in other words, because of the large rejection probability. We then use the model for bacterial infections in day care centers to show in Example 7 that the minimizer of $\hat{J}^N(\theta)$ can provide a good approximation of the maximizer of $\hat{L}_n^N(\theta)$. This is important because \hat{J}^N does not require choosing the bandwidth h or involve any rejections.

Example 6 (Approximate likelihood for the mean of a Gaussian). For the inference of the mean of a Gaussian, we can use as discrepancy Δ_θ the squared difference between the empirical mean of the observed and simulated data \mathbf{y}_o and \mathbf{y}_θ , that is the squared difference between the two summary statistics Φ_o and Φ_θ in Example 4: $\Delta_\theta = (\Phi_o - \Phi_\theta)^2$. Because of the use of simulated data, like the synthetic likelihood, the discrepancy Δ_θ is a stochastic process. We visualize its distribution in Figure 4(a). The observed data \mathbf{y}_o were the same as in Example 4.

For this simple example, we can compute the limiting approximate likelihood \bar{L}_n in Equation (25) in closed form,

$$\bar{L}_n(\theta) \propto F\left(\sqrt{n}(\Phi_o - \theta) + \sqrt{nh}\right) - F\left(\sqrt{n}(\Phi_o - \theta) - \sqrt{nh}\right), \quad (27)$$

where $F(x)$ is the cumulative distribution function (cdf) of a standard normal random variable,

$$F(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}u^2\right) du. \quad (28)$$

For small nh , $\bar{L}_n(\theta)$ becomes proportional to the likelihood $L(\theta)$. This is visualized in Figure 4(b).² However, the probability to actually observe a realization of Δ_θ which is below the threshold h becomes vanishingly small. For realistic models, \bar{L}_n is not available in closed form but needs to be estimated. The vanishingly small probability indicates that the inference procedure will be computationally expensive when \bar{L}_n is estimated via the sample average approximation \hat{L}_n^N . \blacktriangle

Example 7 (Approximate univariate likelihoods for the day care centers). In the model for bacterial infections in day care centers, the observed data were centered to summary

² Using $h = 0.1$ for illustrative purposes. For threshold choice in real applications, see Example 7 and Section 5.3.

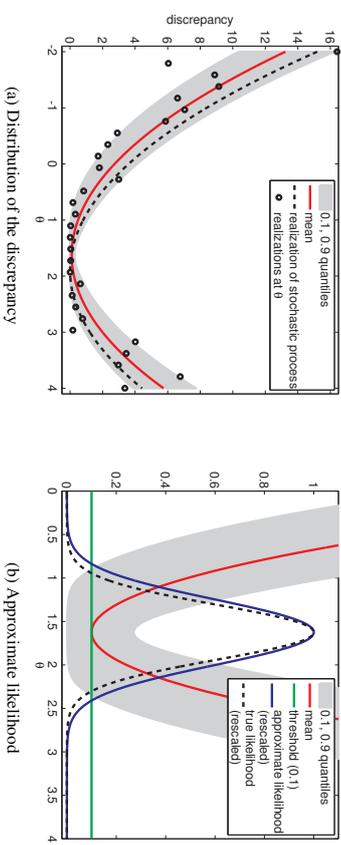


Figure 4: Nonparametric approximation of the likelihood to estimate the mean θ of a Gaussian. The discrepancy Δ_θ is the squared difference between the empirical means of the observed and simulated data. (a) The discrepancy is a random function. (b) The probability that the discrepancy is below some threshold h approximates the likelihood. Note the different range of the axes.

statistics Φ_o by representing each day care center (binary matrix) with four statistics. This gives $4 \cdot 29 = 116$ summary statistics in total (see Numminen et al., 2013, for details).

Since the day care centers can be considered to be independent, the 29 observations can be used to estimate the distribution of the four statistics and their cdfs. Numminen et al. (2013) assessed the difference between Φ_θ and Φ_o by the L_1 distance between the estimated cdfs. Each L_1 distance had its own uniform kernel and corresponding bandwidth, which means that a product kernel was used overall. We here work with a simplified discrepancy measure: The different scales of the four statistics were normalized by letting the maximal value of each of the four statistics be one for \mathbf{y}_o . The discrepancy Δ_θ was then the L_1 norm between Φ_θ and Φ_o divided by their dimension, $\Delta_\theta = 1/116 \|\Phi_\theta - \Phi_o\|_1$.

Figure 5 shows the distributions of the discrepancies Δ_θ if one of the three parameters is varied at a time. The results are for the real data used by Numminen et al. (2013). The parameters were varied on a grid around the (rounded) mean (3.6, 0.6, 0.1) which was inferred by Numminen et al. (2013). The distributions were estimated using $N = 300$ realizations of Δ_θ per parameter value. The red solid lines show the empirical average \hat{L}_n^N . The black lines with circles show \hat{L}_n^N with bandwidths (thresholds) equal to the 0.1 quantile of the sampled discrepancies. While subjective, this is a customary choice (Marin et al., 2012). The thresholds were $h_g = 1.16$, $h_\lambda = 1.18$, and $h_\theta = 1.20$, and are marked with green lines. It can be seen that the optima of \hat{J}^N and L_n^N are attained at about the same parameter values which is advantageous because \hat{J}^N is independent of kernel and bandwidth.

Since the functional form of \hat{J}^N and its gradients are, however, not known, the minimization becomes a difficult problem in higher dimensions. We will show that the idea of

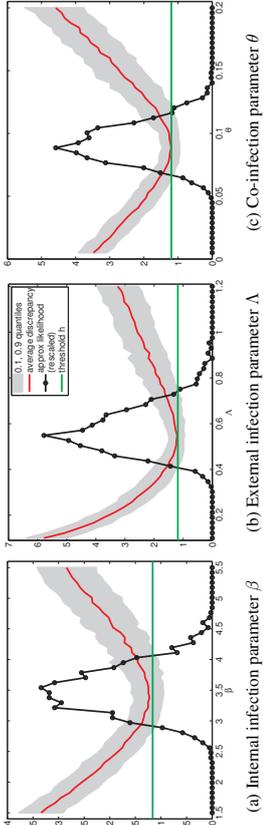


Figure 5: Approximate likelihoods \hat{L}_u^N and distributions of the discrepancy Δ_θ for the day care center example. The green horizontal lines indicate the thresholds used. The optima of the average discrepancies and the approximate likelihoods occur at about the same parameter values.

combining probabilistic modeling with optimization, which we mentioned in Example 5 for the log synthetic likelihood, is also helpful here. \blacktriangle

3.4 Relation between Nonparametric and Parametric Approximation

Kernel density estimation with Gaussian kernels is interesting for two reasons in the context of likelihood-free inference. First, the Gaussian kernel is positive definite, so that the estimated density is a member of a reproducing kernel Hilbert space. This means that more robust approximations of $p_{\Phi|\theta}$ than the one in Equation (18) would exist (Kim and Scott, 2012), and that there might be connections to the inference approach of Fukumizu et al. (2013). Second, it allows us to embed the synthetic likelihood approach of Section 3.2 into the nonparametric approach of Section 3.3.

For the Gaussian kernel, we have that $K(\Phi_o, \Phi_\theta) = K_g(\Phi_o, -\Phi_\theta)$,

$$K_g(\Phi_o - \Phi_\theta) = \frac{1}{(2\pi)^{p/2}} \frac{1}{|\det \mathbf{C}_\theta|^{1/2}} \exp\left(-\frac{(\Phi_o - \Phi_\theta)^\top \mathbf{C}_\theta^{-1} (\Phi_o - \Phi_\theta)}{2}\right), \quad (29)$$

where \mathbf{C}_θ is a positive definite bandwidth matrix possibly depending on θ . The kernel K_g corresponds to $\kappa = \kappa_g$ and $\Delta_\theta = \Delta_\theta^g$,

$$\kappa_g(u) = \frac{1}{(2\pi)^{p/2}} \exp\left(-\frac{u}{2}\right), \quad \Delta_\theta^g = \log|\det \mathbf{C}_\theta| + (\Phi_o - \Phi_\theta)^\top \mathbf{C}_\theta^{-1} (\Phi_o - \Phi_\theta). \quad (30)$$

The function κ_g is convex so that Equation (23) yields a lower bound for $\hat{L}_g^N(\theta) = \hat{L}_g^N(\theta)$ and its logarithm,

$$\log \hat{L}_g^N(\theta) \geq -\frac{p}{2} \log(2\pi) - \frac{1}{2} j_g^N(\theta), \quad (31)$$

$$j_g^N(\theta) = \mathbb{E}^N \left[\log|\det \mathbf{C}_\theta| + (\Phi_o - \Phi_\theta)^\top \mathbf{C}_\theta^{-1} (\Phi_o - \Phi_\theta) \right]. \quad (32)$$

We used the subscript “g” to highlight that \hat{j}^N in Equation (23) is computed for the particular discrepancy Δ_θ^g . The form of \hat{j}_g^N is reminiscent of the log synthetic likelihood $\hat{\ell}_s^N$ in Equation (15). The following proposition shows that there is indeed a connection.

Proposition 1 (Synthetic likelihood as lower bound). For $\mathbf{C}_\theta = \hat{\Sigma}_\theta$,

$$\hat{\ell}_s^N(\theta) = \frac{p}{2} - \frac{p}{2} \log(2\pi) - \frac{1}{2} \hat{j}_g^N(\theta), \quad (33)$$

$$\log \hat{L}_g^N(\theta) \geq -\frac{p}{2} + \hat{\ell}_s^N(\theta) \quad (34)$$

The proposition is proved in Appendix A. It shows that maximizing the synthetic log-likelihood corresponds to maximizing a lower bound of a nonparametric approximation of the log likelihood. The proposition embeds the parametric approach to likelihood approximation conceptually in the nonparametric one and shows furthermore that $\hat{\ell}_s^N$ can be computed via an empirical expectation over Δ_θ^g .

3.5 Posterior Inference using Sample Average Approximations of the Likelihood

Several computable approximations \hat{L} of the likelihood L were constructed in the previous two sections. Table 1 provides an overview. Intractable expectations were replaced with sample averages using N simulated data sets which we denoted by the superscript “ N ” in the symbols for the approximations.

Wood (2010) used the synthetic likelihood \hat{L}_s^N together with a Metropolis MCMC algorithm for posterior computations. We here focus on posterior inference via importance sampling. Using \hat{L}_u^N as \hat{L} in Equation (11), we have

$$\mathbb{E}(g(\theta) | \Phi_o) \approx \sum_{m=1}^M g(\theta^{(m)}) \hat{w}_u^{(m)}, \quad \hat{w}_u^{(m)} = \frac{\sum_{j=1}^N \chi_{[0,h)}(\Delta_\theta^{(j,m)}) \frac{p_{\Phi}(\theta^{(m)})}{q(\theta^{(m)})}}{\sum_{i=1}^M \sum_{j=1}^N \chi_{[0,h)}(\Delta_\theta^{(j,i)}) \frac{p_{\Phi}(\theta^{(i)})}{q(\theta^{(i)})}}, \quad (35)$$

where $\chi_{[0,h)}$ is the indicator function of the interval $[0, h)$ as before, and the $\Delta_\theta^{(j,m)}$, $j = 1, \dots, N$, are the observed discrepancies for the sampled parameter $\theta^{(m)} \sim q(\theta)$. Instead of sampling several discrepancies for the same $\theta^{(m)}$, sampling M' pairs $(\Delta_\theta^{(i)}, \theta^{(i)})$ with $N = 1$ is also possible and corresponds to an asymptotically equivalent solution. With $q = p_\theta$, the approximation is a Nadaraya–Watson kernel estimate of the conditional expectation (see, for example, Wasserman, 2004, Chapter 21).

Approximate Bayesian computation (ABC) is intrinsically linked to kernel density estimation and kernel regression (Blum, 2010). A basic ABC rejection sampler (Pritchard et al., 1999; Marin et al., 2012, Algorithm 2) is obtained from Equation (35) with $N = 1$, $q = p_\theta$, and $\Delta_\theta = \|\Phi_o - \Phi_\theta\|$ where $\|\cdot\|$ is some norm. Approximate samples from the posterior pdf of θ given Φ_o can thus be obtained by retaining those $\theta^{(m)}$ for which the $\Phi_\theta^{(m)}$ are within distance h from Φ_o . In an iterative approach, the accepted particles can be used to define the auxiliary pdf $q(\theta)$ of the next iteration by letting it be a mixture of Gaussians with weights $\hat{w}_u^{(m)}$, center points $\theta^{(m)}$, and a covariance determined by the $\theta^{(m)}$ (Beaumont et al., 2009). This gives the population Monte Carlo (PMC) ABC algorithm (Marin et al., 2012, Algorithm 4). Related sequential Monte Carlo (SMC) ABC algorithms were proposed

by Sisson et al. (2007) and Toni et al. (2009). Working with $q = p_\theta$, Beaumont et al. (2002) introduced ABC with more general kernels, which corresponds to using L_κ^N instead of L_n^N .

Example 6 showed that approximating the likelihood via sample averages is computationally expensive because of the required small thresholds. The auxiliary pdf $q(\theta)$ specifies where in the parameter space the likelihood is predominantly evaluated. The following example shows that avoiding regions in the parameter space where the likelihood is vanishingly small allows for considerable computational savings.

Example 8 (Univariate approximate posteriors for the day care centers). For the inference of the model of bacterial infections in day care centers, Numminen et al. (2013) used uniform priors for the parameters $\beta \in (0, 11)$, $\Lambda \in (0, 2)$, and $\theta \in (0, 1)$. The likelihoods \hat{L}_n^N shown in Figure 5 are thus proportional to the posterior pdfs. The posterior pdfs of the univariate unknowns are conditional on the remaining fixed parameters. For example, the posterior pdf for β is conditional on $(\Lambda, \theta) = (\Lambda_\theta, \theta_\theta) = (0.6, 0.1)$. In Section 7, we consider inference of all three parameters at the same time.

In Figure 5, each parameter is evaluated on a sub-interval of the domain of the prior. The sub-intervals were chosen such that the fat tails of the likelihoods were excluded. Parameter β , for example, was evaluated on the interval (1.5, 5.5) only. Evaluating the discrepancy Δ_θ on the complete interval (0, 11) is not very meaningful since the probability that it is above the chosen threshold is vanishingly small outside the interval (1.5, 5.5). In fact, out of $M = 5,000$ discrepancies Δ_θ which we simulated for β uniformly on (0, 11), not a single one was accepted for $\beta \notin (1.5, 5.5)$. Hence, taking for instance a uniform distribution on (1.5, 5.5) instead of the prior as auxiliary distribution leads to considerable computational savings. Motivated by this, we propose a method which automatically avoids regions in the parameter space where the likelihood is vanishingly small. \blacktriangle

4. Computational Difficulties in the Standard Inference Approach

We have seen that the approximate likelihood functions $\hat{L}(\theta)$ which are used to infer simulator-based statistical models are stochastic processes indexed by the model parameters θ . Their properties, in particular their functional form and gradients, are generally not known: they behave like stochastic black-box functions. The stochasticity is due to the use of simulations to approximate intractable expectations. In the standard approach presented in the previous section, the expectations are approximated by sample averages so that a single evaluation of \hat{L} requires the simulation of N data sets. The standard approach makes minimal assumptions but suffers from a couple of limiting factors.

1. There is an inherent trade-off between computational and statistical efficiency: Reducing N reduces the computational cost of the inference methods, but it can also decrease the accuracy of the estimates (Figure 2).
2. For finite N , the approximate likelihoods may not be smooth (Figure 3).
3. Simulating N data sets uniformly in the parameter space is an inefficient use of computational resources and particularly costly if simulating a single data set already takes a long time. In some regions in the parameter space, far fewer simulations suffice to conclude that it is very unlikely for the approximate likelihood to take a significant value (Figures 2 to 5).

5. Framework to Increase the Computationally Efficiency

We present a framework which combines optimization with probabilistic modeling in order to increase the efficiency of likelihood-free inference of simulator-based statistical models.

5.1 From Sample Average to Regression Based Approximations

The standard approach to obtain a computable approximate likelihood function \hat{L} relies on sample averages, yielding the parametric approximation $\hat{L}_\kappa^N = \exp(\hat{\ell}_\kappa^N)$ in Equation (15) or the nonparametric approximation \hat{L}_κ^N in Equation (21). The approximations are computable versions of $\hat{L}_s = \exp(\hat{\ell}_s)$ in Equation (13) and \hat{L}_κ in Equation (22), which both involve intractable expectations. But sample averages are not the only way to approximate the intractable expectations. We here consider approximations based on regression.

Equation (22) shows that $\hat{L}_\kappa(\theta)$ has a natural interpretation as a regression function where the model parameters θ are the covariates (the independent variables) and $\kappa(\Delta_\theta)$ is the response variable. The expectation can thus also be approximated by solving a regression problem. Further, \hat{J}^N in Equation (23) can be seen as the sample average approximation of the regression function $J(\theta)$,

$$J(\theta) = \mathbb{E}[\Delta_\theta], \quad (36)$$

where the discrepancy Δ_θ is the response variable. The arguments which we used to show that \hat{J}^N provides a lower bound for \hat{L}_κ carry directly over to J and \hat{L}_κ : J provides a lower bound for \hat{L}_κ if κ is convex or the uniform kernel.

Proposition 1 establishes a relation between the sample average quantities \hat{J}_g^N in Equation (32) and \hat{J}_s^N in Equation (15). In the proof of the proposition in Appendix A, we show that the relation extends to the limiting quantities $J_g(\theta) = \mathbb{E}[\Delta_\theta^g]$ and \hat{L}_s in Equation (13). Thus, for $\mathbf{C}_\theta = \Sigma_\theta$ and up to constants and the sign, $\hat{\ell}_s(\theta)$ can be seen as a regression function with the particular discrepancy Δ_θ^g as the response variable.

We next discuss the general strategy to infer the regression functions while avoiding unnecessary computations. For nonparametric approximations to the likelihood, inferring J is simpler than inferring \hat{L}_κ since the function κ and its corresponding bandwidth do not need to be chosen. We thus propose to first infer the regression function J of the discrepancies and then, in a second step, to leverage the obtained solution to infer \hat{L}_κ . For the parametric approximation to the likelihood, this extra step is not needed since J_g is a special instance of the regression function J .

5.2 Inferring the Regression Function of the Discrepancies

Inferring $J(\theta)$ via regression requires training data in the form of tuples $(\theta^{(i)}, \Delta_\theta^{(i)})$. Since we are mostly interested in the region of the parameter space where Δ_θ tends to be small, we propose to actively construct the training data such that they are more densely clustered around the minimizer of $J(\theta)$. As $J(\theta)$ is unknown in the first place, our proposal amounts to performing regression and optimization at the same time: Given an initial guess that the minimizer is in some bounded subset of the parameter space, we can sample some evidence $\mathcal{E}^{(i)}$ of the relation between θ and Δ_θ ,

$$\mathcal{E}^{(i)} = \left\{ (\theta^{(1)}, \Delta_\theta^{(1)}), \dots, (\theta^{(i)}, \Delta_\theta^{(i)}) \right\}, \quad (37)$$

and use this evidence to obtain an estimate $\hat{J}^{(t)}$ of J via regression. The estimated $\hat{J}^{(t)}$ and some measurement of uncertainty about it can then be used to produce a new guess about the potential location of the minimizer, from where the process re-starts. In some cases, it may be advantageous to include the prior pdf of the parameters in the process. We explore this topic in Appendix B.

The evidence set $\mathcal{E}^{(t)}$ grows at every iteration and we may stop at $t = T$. The value of T can be chosen based on computational considerations, by checking whether the learned model predicts the acquired points reasonably well, or by monitoring the change in the minimizer $\hat{\theta}_J^{(t)}$ of $\hat{J}^{(t)}$ as the evidence set grows,

$$\hat{\theta}_J^{(t)} = \operatorname{argmin}_{\theta} \hat{J}^{(t)}(\theta). \quad (38)$$

Given our examples so far, it is further reasonable to assume that J is a smooth function. Even for the Ricker model, the mean objective was smooth although the individual realizations were not (Figure 3). The smoothness assumption about J can be used in the regression and enables its efficient minimization.

For the special case where $\bar{\ell}_s$ is the target, several observed values of $\Delta_{\theta} = \Delta_{\theta}^g$ may be available for any given $\theta^{(t)}$. This is because the covariance matrix Σ_{θ} may be still estimated as a sample average so that multiple simulated summary statistics, and hence discrepancies, are available per $\theta^{(t)}$. They can be used as discussed above with the only minor modification that the training data are updated with several tuples at a time. But it is also possible to only use the average value of the observed discrepancies, which amounts to using the observed values of $\bar{\ell}_s^V$ for training. The estimated regression function $\hat{J}^{(t)}$ provides an estimate for $\bar{\ell}_s$ in either case. We denote the estimate by $\bar{\ell}_s^{(t)}$ and the corresponding estimate of \bar{L}_s by $\bar{L}_s^{(t)}$.

Combining nonlinear regression with the acquisition of new evidence in order to optimize a black-box function is known as Bayesian optimization (see, for example, Brochu et al., 2010). We can thus leverage results from Bayesian optimization to implement the proposed approach, which we will do in Section 6.

5.3 Inferring the Regression Function for Nonparametric Likelihood Approximation

The evidence set $\mathcal{E}^{(t)}$ can be used in two possible ways in the nonparametric setting: The first possibility is to compute for each $\Delta_{\theta}^{(t)}$ in $\mathcal{E}^{(t)}$ the value $\kappa^{(t)} = \kappa(\Delta_{\theta}^{(t)})$ and to thereby produce a new evidence set which can be used to approximate \bar{L}_{κ} by fitting a regression function. The second possibility is to estimate a probabilistic model of Δ_{θ} from the evidence $\mathcal{E}^{(t)}$. The estimated model can be used to approximate \bar{L}_{κ} by replacing the expectation in Equation (22) with the expectation under the model. We denote either approximation by $\bar{L}_{\kappa}^{(t)}$ where the superscript “ (t) ” indicates that the approximation was obtained via regression with t training points. Since $\mathcal{E}^{(t)}$ is such that the approximation of the regression function is accurate where it takes small values, the approximation of \bar{L}_{κ} will be accurate where it takes large values; that is, in the modal areas.

For nonparametric likelihood approximation, kernels and bandwidths need to be selected (see Section 3.3). The choice of the kernel is generally thought to be less critical than the choice of the bandwidth (Wand and Jones, 1995). Bandwidth selection has received considerable attention in the literature on kernel density estimation (for an introduction,

see, for example, Wand and Jones, 1995). The results from that literature are, however, not straightforwardly applicable to our work: We may only be given a certain discrepancy measure Δ_{θ} without underlying summary statistics Φ_{θ} (Gutmann et al., 2014). And even if the discrepancy Δ_{θ} is constructed via summary statistics, the kernel density estimate is only evaluated at Φ_{θ} which is kept fixed while θ is varied. Furthermore, we usually only have very few observations available for any given θ which is generally not the case in kernel density estimation. These differences warrant further investigations into which extend the bandwidth selection methods from the kernel density estimation literature are applicable to likelihood-free inference. We focus in this paper on the uniform kernel and generally choose h via the quantiles of the $\Delta_{\theta}^{(t)}$, which is common practice in approximate Bayesian computation (see, for example, Marin et al., 2012). The approximate likelihood function for the uniform kernel will be denoted by $\hat{L}_u^{(t)}$.

5.4 Benefits and Limitations of the Proposed Approach

The difference between the proposed approach and the standard approach to likelihood-free inference of simulator-based statistical models lies in the way the intractable J and \bar{L} are approximated. We use regression with actively acquired training data while the standard approach relies on computing sample averages. Our approach allows to incorporate a smoothness assumption about J and \bar{L} in the region of their optima. The smoothness assumption allows to “share” observed Δ_{θ} among multiple θ which suggests that fewer $\Delta_{\theta}^{(t)}$, that is, fewer simulated data sets $\mathbf{y}_{\theta}^{(t)}$, are needed to reach a certain level of accuracy. A second benefit of the proposed approach is that it directly targets the region in the parameter space where the discrepancy Δ_{θ} tends to be small, which is very important if simulating data sets is time consuming.

Regression and deciding on the training data are not free of computational cost. While the additional expense is often justified by the net savings made, it goes without saying that if simulating the model is very cheap, methods for regression and decision making need to be used which are not disproportionately costly. Furthermore, prioritizing the low-discrepancy areas of the parameter space is often meaningful, but it also implies that the tails of the likelihood (posterior) will not be as well approximated as the modal areas. The proposed approach thus had to be modified if the computation of small probability events was of primary interest.

Section 4 lists three computational difficulties occurring in the standard approach. Our approach addresses the smoothness issues via smooth regression. The inefficient use of resources is addressed by focusing on regions in the parameter space where Δ_{θ} tends to be small. The trade-off between computational and statistical performance is still present but in modified form: The trade-off is the size of the training set $\mathcal{E}^{(t)}$ used in the regression. The regression functions can be estimated more accurately as the size of the training set grows but this also requires more computation. The size of the training set as trade-off parameter has the advantage that we are free to choose in which areas of the parameter space we would like to approximate the regression function more accurately and in which areas an accurate approximation is not needed. This is in contrast to the standard approach where a computational cost of N simulated data sets needs to be paid per θ irrespective of its value.

6. Implementing the Framework with Bayesian Optimization

We start with introducing Bayesian optimization and then use it to implement our framework. This is followed by a discussion of possible extensions.

6.1 Brief Introduction to Bayesian Optimization

We briefly introduce the elements of Bayesian optimization which are needed in the paper. A more thorough introduction can be found in the review articles by Jones (2001) and Brochu et al. (2010). While the presented version of Bayesian optimization is rather straightforward and textbook-like, our framework can also be implemented with more advanced versions, see Section 6.4.

Bayesian optimization comprises a set of methods to minimize black-box functions $f(\boldsymbol{\theta})$. With a black-box function, we mean a function which we can evaluate but whose form and gradients are unknown. The basic idea in Bayesian optimization is to use a probabilistic model of f to select points where the objective is evaluated, and to use the obtained values to update the model by Bayes' theorem.

The objective f is often modeled as a Gaussian process which is also done in this paper: We assume that f is a Gaussian process with prior mean function $m(\boldsymbol{\theta})$ and covariance function $k(\boldsymbol{\theta}, \boldsymbol{\theta}')$ subject to additive Gaussian observation noise with variance σ_n^2 . The joint distribution of f at any t points $\boldsymbol{\theta}^{(1)}, \dots, \boldsymbol{\theta}^{(t)}$ is thus assumed Gaussian with mean \mathbf{m}_t and covariance \mathbf{K}_t ,

$$(f^{(1)}, \dots, f^{(t)})^\top \sim \mathcal{N}(\mathbf{m}_t, \mathbf{K}_t), \quad (39)$$

$$\mathbf{m}_t = \begin{pmatrix} m(\boldsymbol{\theta}^{(1)}) \\ \vdots \\ m(\boldsymbol{\theta}^{(t)}) \end{pmatrix}, \quad \mathbf{K}_t = \begin{pmatrix} k(\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(1)}) & \dots & k(\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(t)}) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(1)}) & \dots & k(\boldsymbol{\theta}^{(t)}, \boldsymbol{\theta}^{(t)}) \end{pmatrix} + \sigma_n^2 \mathbf{I}_t. \quad (40)$$

We used $f^{(t)}$ to denote $f(\boldsymbol{\theta}^{(t)})$ and \mathbf{I}_t is the $t \times t$ identity matrix. While other choices are possible, we assume that $m(\boldsymbol{\theta})$ is either a constant or a sum of convex quadratic polynomials in the elements θ_j of $\boldsymbol{\theta}$, cross-terms were not included, and that $k(\boldsymbol{\theta}, \boldsymbol{\theta}')$ is a squared exponential covariance function,

$$m(\boldsymbol{\theta}) = \sum_j a_j \theta_j^2 + b_j \theta_j + c, \quad k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \sigma_f^2 \exp\left(\sum_j \frac{1}{\lambda_j^2} (\theta_j - \theta'_j)^2\right). \quad (41)$$

These are standard choices (see, for example, Rasmussen and Williams, 2006, Chapter 2). Since we are interested in minimization, we constrain the a_j to be non-negative. In the last equation, θ_j and θ'_j are the elements of $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$, respectively; σ_f^2 is the signal variance, and the λ_j are the characteristic length scales. The length scales control the amount of correlation between $f(\boldsymbol{\theta})$ and $f(\boldsymbol{\theta}')$, in other words, they control the wiggliness of the realizations of the Gaussian process. The signal variance is the marginal variance of f at a point $\boldsymbol{\theta}$ if the observation noise was zero.

The quantities a_j , b_j , c , σ_f^2 , λ_j , and σ_n^2 are hyperparameters. For the results in this paper, they were learned by maximizing the leave-one-out log predictive probability (a form

of cross-validation, see Rasmussen and Williams, 2006, Section 5.4.2). The hyperparameters were slowly updated as new data were acquired, as done in previous work, for example by Wang et al. (2013). This yielded satisfactory results but there are several alternatives, including Bayesian methods to learn the hyperparameters (for an overview, see Rasmussen and Williams, 2006, Chapter 5), and we did not perform any systematic comparison.

Given evidence $\mathcal{E}_f^{(t)} = \{(\boldsymbol{\theta}^{(1)}, f^{(1)}), \dots, (\boldsymbol{\theta}^{(t)}, f^{(t)})\}$, the posterior pdf of f at a point $\boldsymbol{\theta}$ is Gaussian with posterior mean $\mu_t(\boldsymbol{\theta})$ and posterior variance $v_t(\boldsymbol{\theta}) + \sigma_n^2$,

$$f(\boldsymbol{\theta}) | \mathcal{E}_f^{(t)} \sim \mathcal{N}(\mu_t(\boldsymbol{\theta}), v_t(\boldsymbol{\theta}) + \sigma_n^2), \quad (42)$$

where (see, for example, Rasmussen and Williams, 2006, Section 2.7),

$$\mu_t(\boldsymbol{\theta}) = m(\boldsymbol{\theta}) + \mathbf{k}_t(\boldsymbol{\theta})^\top \mathbf{K}_t^{-1} (\mathbf{f}_t - \mathbf{m}_t), \quad v_t(\boldsymbol{\theta}) = k(\boldsymbol{\theta}, \boldsymbol{\theta}) - \mathbf{k}_t(\boldsymbol{\theta})^\top \mathbf{K}_t^{-1} \mathbf{k}_t(\boldsymbol{\theta}), \quad (43)$$

$$\mathbf{f}_t = (f^{(1)}, \dots, f^{(t)})^\top, \quad \mathbf{k}_t(\boldsymbol{\theta}) = (k(\boldsymbol{\theta}, \boldsymbol{\theta}^{(1)}), \dots, k(\boldsymbol{\theta}, \boldsymbol{\theta}^{(t)}))^\top. \quad (44)$$

The posterior mean μ_t emulates f and can be minimized with powerful gradient-based optimization methods.

The evidence set can be augmented by selecting a new point $\boldsymbol{\theta}^{(t+1)}$ where f is next evaluated. The point is chosen based on the posterior distribution of f given $\mathcal{E}_f^{(t)}$. While other choices are equally possible, we use the acquisition function $\mathcal{A}_t(\boldsymbol{\theta})$ to select the next point,

$$\mathcal{A}_t(\boldsymbol{\theta}) = \mu_t(\boldsymbol{\theta}) - \sqrt{\eta_t^2 v_t(\boldsymbol{\theta})}, \quad (45)$$

where $\eta_t^2 = 2 \log[\rho^{d/2+2} \pi^2 / (3\epsilon_\eta)]$ with ϵ_η being a small constant (we used $\epsilon_\eta = 0.1$). This acquisition function is known as the lower confidence bound selection criterion (Cox and John, 1992, 1997; Srinivas et al., 2010, 2012).³ Classically, $\boldsymbol{\theta}^{(t+1)}$ is chosen deterministically as the minimizer of $\mathcal{A}_t(\boldsymbol{\theta})$. The minimization of $\mathcal{A}_t(\boldsymbol{\theta})$ yields a compromise between exploration and exploitation: Minimization of the posterior mean $\mu_t(\boldsymbol{\theta})$ corresponds to exploitation of the current belief and ignores its uncertainty. Minimization of $-\sqrt{v_t(\boldsymbol{\theta})}$, on the other hand, corresponds to exploration where we seek a point where we are uncertain about f . The coefficient η_t implements the trade-off between these two desiderata.

There is usually no restriction that $\boldsymbol{\theta}^{(t+1)}$ must be different from previously acquired $\boldsymbol{\theta}^{(t)}$. We found, however, that this may result in a poor exploration of the parameter space (see Figure 7 and Example 10 below). Employing a stochastic acquisition rule avoids getting stuck at one point. We used the simple heuristic that $\boldsymbol{\theta}^{(t+1)}$ is sampled from a Gaussian with diagonal covariance matrix and mean equal to the minimizer of the acquisition function. The standard deviations were determined by finding the end-points of the interval where the acquisition function was within a certain (relative) tolerance. Other stochastic acquisition rules, like for example Thompson sampling (Thompson, 1933; Chapelle and Li, 2011; Russo and Van Roy, 2014), could alternatively be used.

The algorithm was initialized with an evidence set $\mathcal{E}_f^{(0)}$ where the parameters $\boldsymbol{\theta}^{(1)}, \dots, \boldsymbol{\theta}^{(k_0)}$ were chosen as a Sobol quasi-random sequence (see, for example, Niederreiter, 1988). Compared to uniformly distributed (pseudo) random numbers, the Sobol sequence covers the

³ In the literature, maximization instead of minimization problems are often considered. For maximization problems, the acquisition function becomes $\mu_t(\boldsymbol{\theta}) + \sqrt{\eta_t^2 v_t(\boldsymbol{\theta})}$ and needs to be maximized. The formula for η_t^2 is used in the review by Brochu et al. (2010) and is part of Theorem 2 of Srinivas et al. (2010).

parameter space in a more even fashion. This kind of initialization is, however, not critical to our approach, and only few initial points were used in our simulations.

6.2 Inferring the Regression Function of the Discrepancies

Letting $f(\theta) = \Delta_\theta$, Bayesian optimization yields immediately an estimate of $J(\theta)$ in Equation (36). Since Δ_θ is non-negative, working with $f = \log \Delta_\theta$ seems to be theoretically more sound. In practice, however, both approaches were found to work well, albeit we do not aim at any systematic comparison here. If $f = \Delta_\theta$, the estimate $\hat{J}^{(t)}$ of J is given by the posterior mean μ_t , and if $f = \log \Delta_\theta$, the estimate is given by the mean of a log-normal random variable,

$$\hat{J}^{(t)}(\theta) = \begin{cases} \mu_t(\theta) & \text{if } f(\theta) = \Delta_\theta, \\ \exp\left(\mu_t(\theta) + \frac{1}{2}(\sigma_t(\theta) + \sigma_n^2)\right) & \text{if } f(\theta) = \log \Delta_\theta. \end{cases} \quad (46)$$

As discussed in Section 5.2, in the parametric approach to likelihood approximation, $\hat{J}^{(t)}$ equals the computable approximation $\hat{\ell}_s^{(t)}$ of $\bar{\ell}_s$.

We illustrate the basic principles of Bayesian optimization in Example 9 below. In Example 10, we illustrate log-Gaussian modeling and the stochastic acquisition rule.

Example 9 (Bayesian optimization to infer the mean of a Gaussian). For inference of the mean of a univariate Gaussian, the squared difference of the empirical means was used as the discrepancy measure Δ_θ , as in Example 6. We modeled the discrepancy Δ_θ as a Gaussian process with constant prior mean and performed Bayesian optimization with the deterministic acquisition rule. Figure 6 shows the first iterations: When only a single observation of Δ_θ is available, $t = 1$ and Δ_θ is believed to be constant but there is considerable uncertainty about it (upper-left panel). The posterior distribution of the Gaussian process yields the acquisition function $\mathcal{A}_1(\theta)$ according to Equation (45) (curve in magenta). Its minimization gives the value $\theta^{(2)}$ where Δ_θ is evaluated next (blue rectangle). After including the observed value of Δ_θ into the evidence set, $t = 2$ and the posterior distribution of the Gaussian process is re-calculated using Equation (42), that is, the belief about Δ_θ is updated using Bayes' theorem (upper-right panel). The updated belief becomes the current belief and the process restarts. A movie showing the process over several iterations is available at <http://www.jmlr.org/papers/volume17/15-017/supplementary/Gauss.avi>. \blacktriangle

Example 10 (Bayesian optimization to infer the growth rate in the Ricker model). Example 5 introduced the synthetic likelihood for the Ricker model. We have seen that individual realizations of $\hat{\ell}_s^N$ are rather noisy, in particular for $N = 50$, but that their average, which represents an estimate of $\bar{\ell}_s$, is smooth with its optimum in the right region (Figure 3). We here obtain estimates $\hat{\ell}_s^{(t)}$ of $\bar{\ell}_s$ with Bayesian optimization. The maximal training data are $T = 150$ tuples $(\log r, \hat{\ell}_s^N)$, where the first nine are from the initialization. The log synthetic likelihood was computed using code of Wood (2010) which only returned $\hat{\ell}_s^N$ and not the multiple discrepancies prior to averaging.

Figures 7(a) and (b) show $-\hat{\ell}_s^{(t)}$ after initialization without and with log-transformation, respectively (black solid lines). In both cases, we used a quadratic prior mean function. The estimated limiting negative log synthetic likelihood $-\bar{\ell}_s$ from Figure 3 is shown in red for

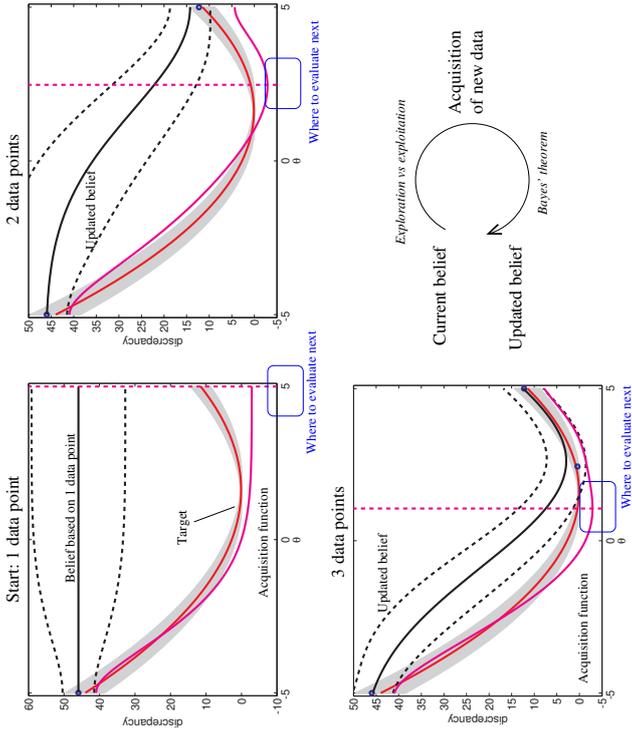


Figure 6: The first iterations of Bayesian optimization to estimate the mean of a Gaussian. The distribution of Δ_θ and its regression function $J(\theta)$ are reproduced from Figure 4 for reference (labeled “Target”). Bayesian optimization consists in acquiring new data based on the current belief, followed by an update of the belief by Bayes’ theorem. The acquisition of new data is based on an acquisition function which implements a trade-off between exploration and exploitation. Exploitation after two data points would consist in evaluating the objective again at $\theta = 5$. Exploration would consist in evaluating it where the posterior variance is large, that is, somewhere between minus five and zero. The point selected (blue rectangle) strikes a compromise between the two extremes.

reference. Figure 7(c) shows that the deterministic decision rule can lead to acquisitions with very little spatial exploration. The reason for the poor exploration is presumably the rather large variance of $\hat{\ell}_s^N$ for $N = 50$. Working with a log-Gaussian process leads to a better exploration and also to a better approximation (Figure 7(d)). The acquisitions happen, however, still in a cluster-like manner, which can also be seen in Figure 14 in Appendix C where we provide a more detailed analysis. Working with a stochastic decision rule leads to acquired points which are spread out more evenly in the area of interest. This results in both more stable and more accurate approximations (Figures 7(e-f) and Figure 15 in Appendix C). \blacktriangle

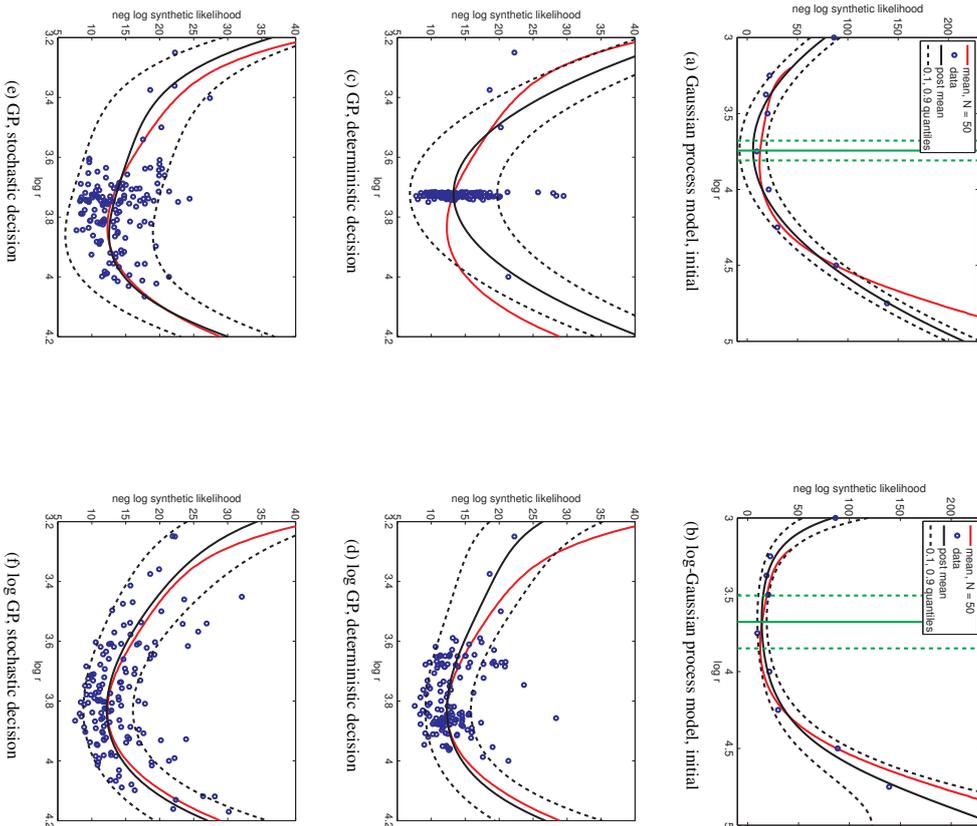


Figure 7. Approximation of the limiting negative log synthetic likelihood $-\hat{\ell}_s$ for the Ricker example. The approximations are shown as black solid curves. The black dashed curves indicate the variability of $-\hat{\ell}_s^N$, and the red curves show $-\hat{\ell}_s$ from Figure 3 for reference. (a–b) The approximation after initialization with 9 data points. The green vertical lines indicate the minimizer of the acquisition function. The dashed vertical lines show the mean plus-or-minus one standard deviation in the stochastic decision rule. (c–f) The approximations are based on 150 data points (blue circles).

6.3 Model-Based Nonparametric Likelihood Approximation

Bayesian optimization yields a probabilistic model for the discrepancy Δ_θ . As discussed in Section 5.3, we can use this model to obtain the computable likelihood approximation $\hat{L}_n^{(t)}$,

$$\hat{L}_n^{(t)}(\theta) \propto \begin{cases} F\left(\frac{h - \mu_t(\theta)}{\sqrt{v_t(\theta) + \sigma_n^2}}\right) & \text{if } f(\theta) = \Delta_\theta, \\ F\left(\frac{\log h - \mu_t(\theta)}{\sqrt{v_t(\theta) + \sigma_n^2}}\right) & \text{if } f(\theta) = \log \Delta_\theta, \end{cases} \quad (47)$$

where h is the bandwidth (threshold). The function $F(x)$ was defined in Equation (28) and denotes the cdf of a standard normal random variable, and μ_t and $v_t + \sigma_n^2$ are the posterior mean and variance of the Gaussian process.

Both $\hat{L}_n^{(t)}$ in the nonparametric approach and $\hat{\ell}_s^{(t)} = \exp(-\hat{\ell}_s^{(t)})$ in the parametric approach are computable approximations \hat{L} of the likelihood L . Evaluating them is cheap since no further runs of the simulator are needed. Derivatives can also be computed since the derivatives of the posterior mean and variance are tractable for Gaussian processes. A given approximate likelihood function can thus be used in various ways for inference: We can maximize it and compute its curvature (Hessian matrix) to obtain error bars, we can perform inference with a hybrid Monte Carlo algorithm in a MCMC framework, or use it according to Equation (11) in an importance sampling approach.

For the results in this paper, we used iterative importance sampling where in each iteration, the auxiliary pdf q was a mixture of Gaussians as in Section 3.5. The initial auxiliary pdf was defined as a mixture of Gaussians in the same manner by associating uniform weights with the $\theta^{(i)}$ acquired in the Bayesian optimization step. Samples from the prior pdf p_θ are not needed in such an approach, which can be advantageous if obtaining them is expensive.

We next illustrate model-based likelihood approximation using the example about bacterial infections in day care centers.

Example 11 (Model-based approximate univariate likelihoods for the day care centers). We inferred the likelihood function via Bayesian optimization using a Gaussian process model with quadratic prior mean and $T = 50$ data points (10 initial points and 40 acquisitions). The bandwidths and general setup were as in Example 7. The left column of Figure 8 shows the estimated models of the discrepancies for the different parameters and compares them with the empirical distributions reported in Figure 5. The right column of Figure 8 shows the estimated likelihood functions $\hat{L}_n^{(t)}$, $t = 50$ (blue solid curves), and compares them with the sample average based approximations $\hat{L}_n^{(t)}$ from Figure 5 (black dots). For Bayesian optimization, the computational cost for an entire likelihood curve was 50 simulations. This is in stark contrast to the computational cost of $N = 300$ simulations for a single evaluation of \hat{L}_n in the sample-based approach. Since \hat{L}_n was evaluated on a grid of 50 points, the model-based results required 300 times fewer simulations. The computational savings were achieved through the use of smooth regression and the active construction of the training data in Bayesian optimization. \blacktriangle

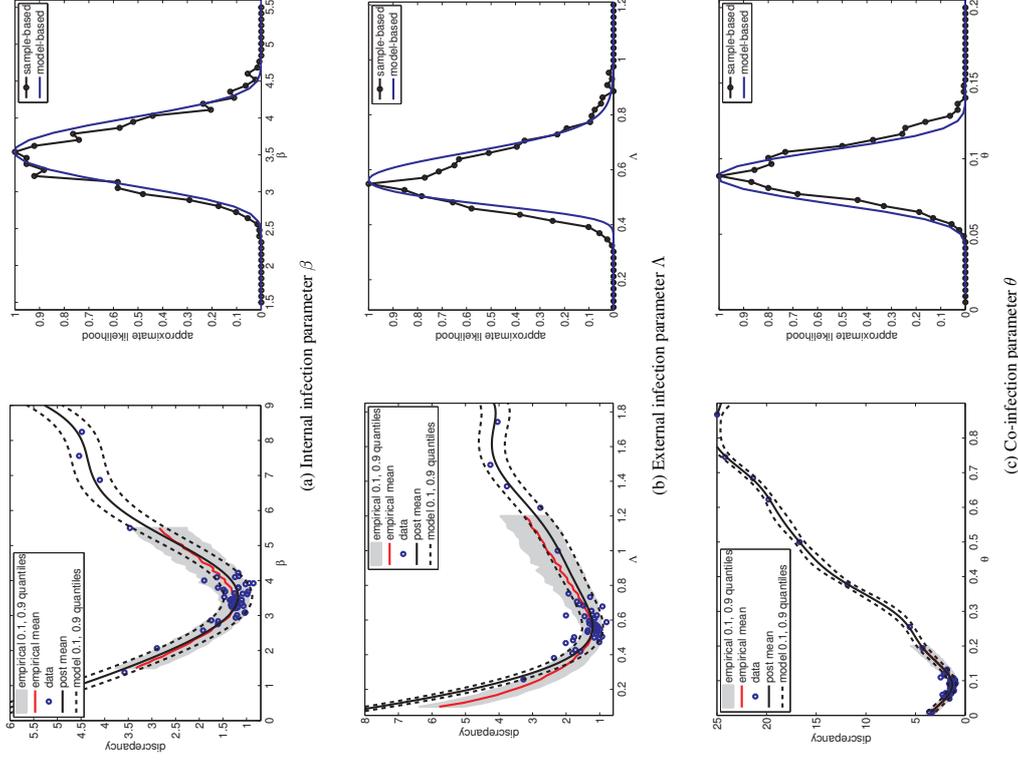


Figure 8: Distributions of the discrepancies and approximate likelihoods for the day care center example. For reference, the sample average results are reproduced from Figure 5. In the standard sample average approach, each likelihood curve required 15,000 simulations (right column, black lines with markers). In the proposed model-based approach, each likelihood curve required 50 simulations (right column, blue solid lines). This yields a factor of 300 in computational savings.

6.4 Possible Extensions

We use in this paper a basic version of Bayesian optimization to do likelihood-free inference. But more advanced versions exist which opens up a range of possible extensions.

6.4.1 SCALABILITY WITH THE NUMBER OF ACQUISITIONS

The straightforward approach of Section 6.1 to Bayesian optimization with a Gaussian process model requires the inversion of the $t \times t$ matrix \mathbf{K}_t . The inversion has complexity $\mathcal{O}(t^3)$ which limits the number of acquisitions to a few thousands. For the applications in this paper, this has not been an issue but we would like to be able to acquire more than a few thousand points if necessary.

Research on Gaussian processes has produced numerous methods to deal with the inversion of \mathbf{K}_t (for an overview, see Rasmussen and Williams, 2006, Chapter 8). Importantly, we can directly use any of these methods for the purpose of likelihood-free inference. For example, sparse Gaussian process regression employs $m < t$ “inducing variables” to reduce the complexity from $\mathcal{O}(t^3)$ to $\mathcal{O}(tm^2)$ (see, for example, Quinero-Candela and Rasmussen, 2005). The inducing variables and the hyperparameters of the Gaussian process can be optimized using variational learning (Titsias, 2009), which is also amenable to stochastic optimization to further reduce the computational cost (Hensman et al., 2013).

An alternative approach to Gaussian process regression is Bayesian linear regression with a set of $m < t$ suitably chosen basis functions. The two approaches are closely related (see, for example, Rasmussen and Williams, 2006, Chapter 2), but instead of a $t \times t$ matrix, a $m \times m$ matrix needs to be inverted. This reduces the computational complexity again to $\mathcal{O}(tm^2)$. In order to keep the number of required basis functions small, adaptive basis regression with deep neural networks has been employed to perform Bayesian optimization (Snoek et al., 2015).

6.4.2 HIGH-DIMENSIONAL INFERENCE

Likelihood-free inference is in general very difficult when the dimensionality d of the parameter space is large. This difficulty manifests itself in our approach in the form of a nonlinear regression problem which needs to be solved. While we are only interested in accurate regression results in the areas of the parameter space where the discrepancy is small, discovering these areas becomes more difficult as the dimension increases.

In general, more training data are needed with increasing dimensions so that a method which can handle a large number of acquisitions is likely required (see above). Furthermore, the optimization of acquisition functions is also more difficult in higher dimensions.

Bayesian optimization in high dimensions typically relies on structural assumptions about the objective function. In recent work, it was assumed that the objective varies along a low dimensional subspace only (Chen et al., 2012; Wang et al., 2013; Djolonga et al., 2013), or that it takes the form of an additive model (Kandasamy et al., 2015). This work and further developments in high-dimensional Bayesian optimization can be leveraged for the challenging problem of high-dimensional likelihood-free inference.

6.4.3 PARALLELIZATION AND ACQUISITION RULES

Bayesian optimization lends itself to parallelization. In particular the acquisition of new data points can be performed in parallel. While several well-known acquisition rules are sequential, they can also be parallelized. Our stochastic acquisition rule provides an easy mechanism by using a sequential rule to define a probability distribution for the location of the next acquisition. Several points can then be drawn in parallel from that distribution. We employ the lower confidence bound selection criterion in Equation (45) to drive the stochastic acquisitions, but alternative rules, for example the maximization of expected improvement, can be used in an analogous way. Other stochastic acquisition rules, like for instance Thompson sampling (Thompson, 1933; Chapelle and Li, 2011; Russo and Van Roy, 2014), enable similarly the concurrent acquisition of multiple data points.

A more elaborate way to parallelize a sequential acquisition rule is to design the joint acquisitions such that the resulting algorithm behaves as if the points are chosen sequentially (Azimi et al., 2010), or to integrate out the possible outcomes of the pending function evaluations (Snoek et al., 2012). Moreover, parallel versions of the lower/upper confidence bound criterion have been proposed by Cortal et al. (2013) and Desautels et al. (2014).

In most theoretical studies on acquisition rules, the objective function in Bayesian optimization is modeled as a Gaussian process with uncorrelated Gaussian observation noise. The distribution of the (log) discrepancy, however, may not follow this assumption. This implies on the one hand that the probabilistic modeling of the discrepancy could be improved (see below). On the other hand, it also means that further research would be needed about optimal acquisition rules in the context of likelihood-free inference.

6.4.4 PROBABILISTIC MODEL

We modeled the discrepancy $\Delta\theta$ as a Gaussian or log-Gaussian process using a squared exponential covariance function and uncorrelated Gaussian observation noise. While simple and often used, we are not limited to these choices. The literature on Gaussian process regression and Bayesian optimization provides several alternatives and extensions (for an overview, see Rasmussen and Williams, 2006). Modeling of $\Delta\theta$ is important because the model affects the inferences made.

In the employed model, a stationary prior distribution is assumed. However, depending on the simulator, the discrepancy may behave differently in different parameter regions. In particular its variance may be input dependent (heteroscedasticity). Such cases can be handled by non-stationary covariance functions or by using different stationary processes in different regions of the parameter space (see, for example, Rasmussen and Williams, 2006, Chapters 6 and 9).

Equation (42) shows that for the Gaussian process model, the posterior variance does not depend on the observed function values but only on the acquisition locations. As more points are acquired in a neighborhood of a point, the posterior variance may shrink even if the observed function values have a larger than expected spread. A dependency on the observed values can be obtained indirectly by updating the hyperparameters of the covariance function. But a more direct dependency may be preferable. An option is to use Student's t processes instead where the posterior variance depends on the observed function values through a global scaling factor (Shah et al., 2014).

7. Applications

We here apply the developed methodology to infer the complete Ricker model and the complete model of bacterial infections in day care centers. As in the previous section, using Bayesian optimization in likelihood-free inference (BOLEFI) reduces the amount of required simulations by several orders of magnitude.

7.1 Ricker Model

We introduced the Ricker model in Example 2. It has three parameters: $\log r$, σ , and ϕ . The difficulty in the inference stems from the dynamics of the latent time series and the unobserved variables. We inferred the parameters using the synthetic likelihood of Wood (2010) from the data shown in Figure 1(a) which were generated with $\theta_0 = (3.8, 0.3, 10)$.

Wood (2010) inferred the model with a random walk Markov chain Monte Carlo algorithm using $\hat{\ell}_s^N(\theta)$ with $N = 500$. The random walk was defined on the log-parameters due to their positivity. In a baseline study with the computer code made publicly available by Wood (2010), we were not able to infer the parameters with the settings in the original publication (Wood, 2010, Section 1.1 in the supplementary material). Reducing the proposal standard deviation for σ by a factor of ten enabled inference even though different Markov chains still led to rather different marginal posterior pdfs for σ . These issues were observed for $N \in \{500, 1000, 5000\}$ and for Markov chains run twice as long as in the original publication (100,000 versus 50,000 iterations). In addition to the usual random effects in MCMC, the variability in the outcomes of the different chains may be due to his approach of working on a single realization of the random log synthetic likelihood function (see Figure 3 for example realizations when only $\log r$ is varied). The results of our baseline study are reported in Appendix D. Given the nature of the baseline results, we should not expect that the results from our method match them exactly.

For BOLEFI, we modeled the random log synthetic likelihood $\hat{\ell}_s^N$ as a log-Gaussian process with a quadratic prior mean function (using $N = 500$ as Wood, 2010). Bayesian optimization was performed with the stochastic acquisition rule and 20 initial data points. Figure 9 shows $-\hat{\ell}_s^N(\theta)$ for $t \in \{50, 150, 500\}$. The results for $t = 50$ and $t = 500$ differ more in the shape of the estimated regression functions than in the location of the optima. As the evidence set grows, the algorithm learns that the log synthetic likelihood is less confined along σ and that the curvature along the other dimensions should be larger. The plot also shows that there is a negative correlation between $\log r$ and ϕ (conditional on σ). This is reasonable since a larger growth rate r can be compensated with a smaller value of the observation scalar ϕ and vice versa.

The approximation $\hat{\ell}_s^N$ was used to perform posterior inference of the parameters via the iterative importance sampling scheme of Section 6.3 (using three iterations with 25,000 samples each). This sampling is purely model-based and does not require further runs of the simulator. The computed marginal posterior pdfs are shown in Figure 10 (curves in gray) together with a MCMC solution for reference (blue dashed). It can be seen that already after $t = 150$ acquired data points, we obtain a solution which matches the MCMC solution well at a fraction of the computational cost. About 600 times fewer calls to the simulator were needed.

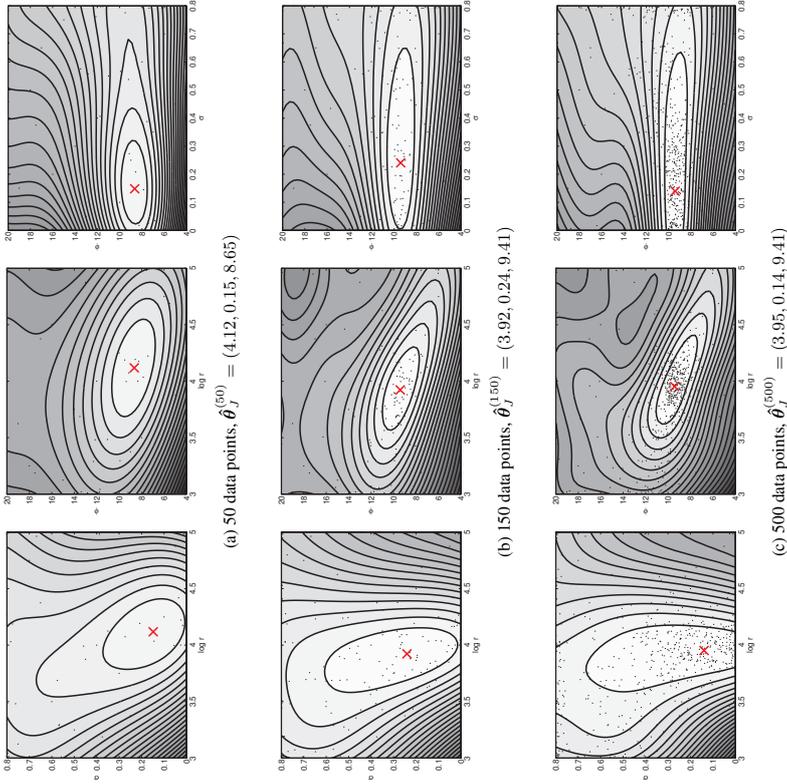


Figure 9: Isocontours of the estimated negative log synthetic likelihood function for the Ricker model. Each panel shows slices of $-\hat{\ell}_s^{(t)}$ with $\text{argmax} \hat{\ell}_s^{(t)}$ as center point when two of the three variables are varied at a time. The center points are marked with a red cross. The dots mark the location of the acquired parameters $\theta^{(t)}$ (projected onto the plane). The intensity map is the same in all figures; white corresponds to the smallest value.

The largest differences between the model-based and the MCMC solution occur for parameter σ (Figure 10(b)). But we have seen that this is a difficult parameter to infer and that the MCMC solution may actually not correspond to ground truth. The two posteriors inferred by MCMC have, for instance, posterior means (blue diamonds) which are further from the data generating parameter $\sigma_o = 0.3$ (green circle) than our model-based solution (black square). For the other parameters, the posterior means of the model-based solution are also closer to ground truth than the posterior means of the MCMC solution.

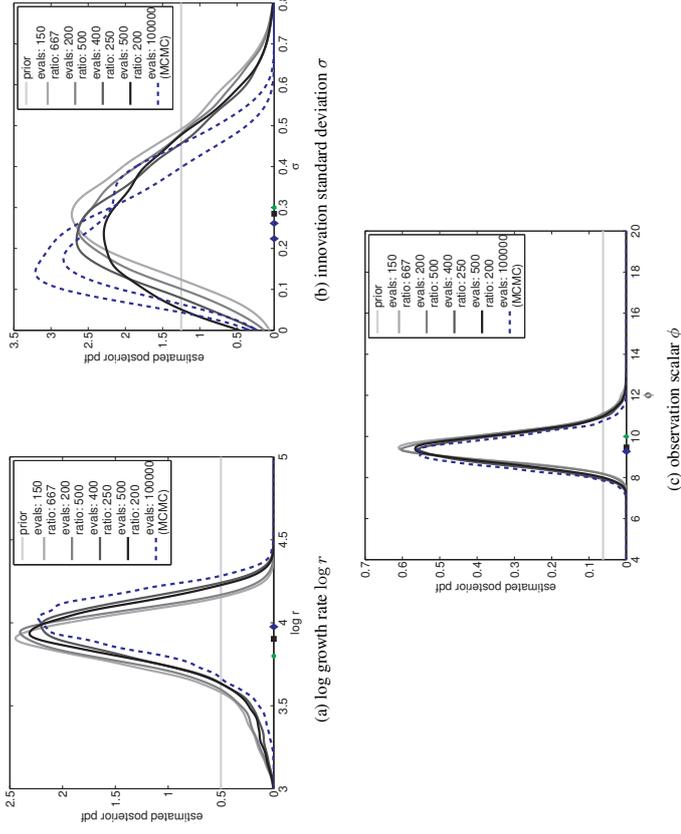


Figure 10: Marginal posterior pdfs for the Ricker model. The model-based solutions are shown in gray, the blue dashed curves are the MCMC solution. The green circles on the x-axes mark the location of $\theta_o = (3.8, 0.3, 10)$. The blue diamonds mark the value of the posterior mean for the MCMC solution while the black squares indicate the posterior means of the model-based solution. For the MCMC solution, 100,000 simulated data sets are needed. Bayesian optimization yields informative solutions using 150 simulated data sets only, which corresponds to 667 times fewer simulations than with MCMC.

7.2 Bacterial Infections in Day Care Centers

The model for bacterial infections in day care centers was described in Example 3. It has three parameters of interest: β , Λ , and θ . The likelihood function is intractable due to the infinitely many unobserved correlated variables. We inferred the model using the discrepancy Δ_θ described in Example 7 from the same real data as Numminen et al. (2013).

For BOLEFI, we modeled the discrepancy Δ_θ as a Gaussian process with a quadratic prior mean function and used the stochastic acquisition rule. The algorithm was initialized with 20 data points. Figure 11 shows the estimated regression functions $\hat{J}^{(t)}$ for

$t \in \{50, 100, 150, 500\}$. For $t = 50$, the optimal co-infection parameter θ is at a boundary of the parameter space. As more training data are acquired, the shape of the estimated regression function changes. The algorithm learns that the optimal θ is located away from the boundary, and the isocontours become oblique which indicates a negative (conditional) correlation between all three parameters. A negative correlation between β and Λ given the estimate of θ is reasonable because an increase in transmissions inside the day care centers (increase of β) can be compensated with a decrease of transmissions from an outside source (decrease of Λ). The co-infection parameter θ is negatively correlated with β given the estimate of Λ because a decrease in the tendency to be infected by multiple strains of the bacterium (decrease of θ) can be offset by an increase of the transmission rate (increase of β). The same reasoning applies to Λ given a fixed value of β .

We used the Gaussian process model of the discrepancy to compute the model-based likelihood $\hat{L}_h^{(t)}$. The threshold h was chosen as the 0.05 quantile of the modeled discrepancy at the minimizer of the estimated regression function. Model-based posterior inference was then performed via iterative importance sampling as described in Section 6:3 (using three iterations with 25,000 samples each). Figure 12 (left column) shows the inferred marginal posterior pdfs. They stabilize quickly as the amount of acquired data increases.

The right column in Figure 12 compares our model-based results with the solution by Numminen et al. (2013) (blue horizontal lines with triangles) and with results by the population Monte Carlo (PMC) ABC algorithm of Section 3.5 (black curves with diamonds). Numminen et al. (2013) used a PMC-ABC algorithm as well but with a slightly different discrepancy measure (see Example 7). Both PMC results were obtained using 10,000 initial simulations to set the initial threshold, followed by four more iterations with shrinking thresholds where in each iteration, data sets were simulated till 10,000 accepted parameters were obtained. It can be seen that the posterior mean and the credibility intervals of the two PMC results match in the fourth generation, which indicates that our modification of the discrepancy measurement had a negligible influence. For the PCM results shown in black, iteration one to four required 121,374; 277,997; 572,007; and 1,218,382 simulations each, giving a total computational cost of 2,199,760 simulations for the results of iteration four. In terms of computing time, the PMC computations took about 4.5 days on a cluster with 200 cores. Our model-based results for $t = 1,000$ were obtained with one tenth of the initial simulations of the reference methods and took only about 1.5 hours on a desktop computer.⁴ Out of the computing time, 93% were spent on simulating the day care centers, and 7% on regression and optimization of the acquisition function.

The posterior means of our model-based approach match quickly the reference results (red curves with circles versus blue curves with triangles). The focus on the modal region yields, however, broader credibility intervals. The broader model-based posterior pdfs suggest that they could be used as auxiliary pdf for PMC-ABC or other iterative ABC algorithms which are based on importance sampling. Moreover, one could evaluate the discrepancy at the sampled points to obtain additional training data in order to refine the model.

⁴ The simulation of the 29 day care centers in the model was partly parallelized by means of seven cores.

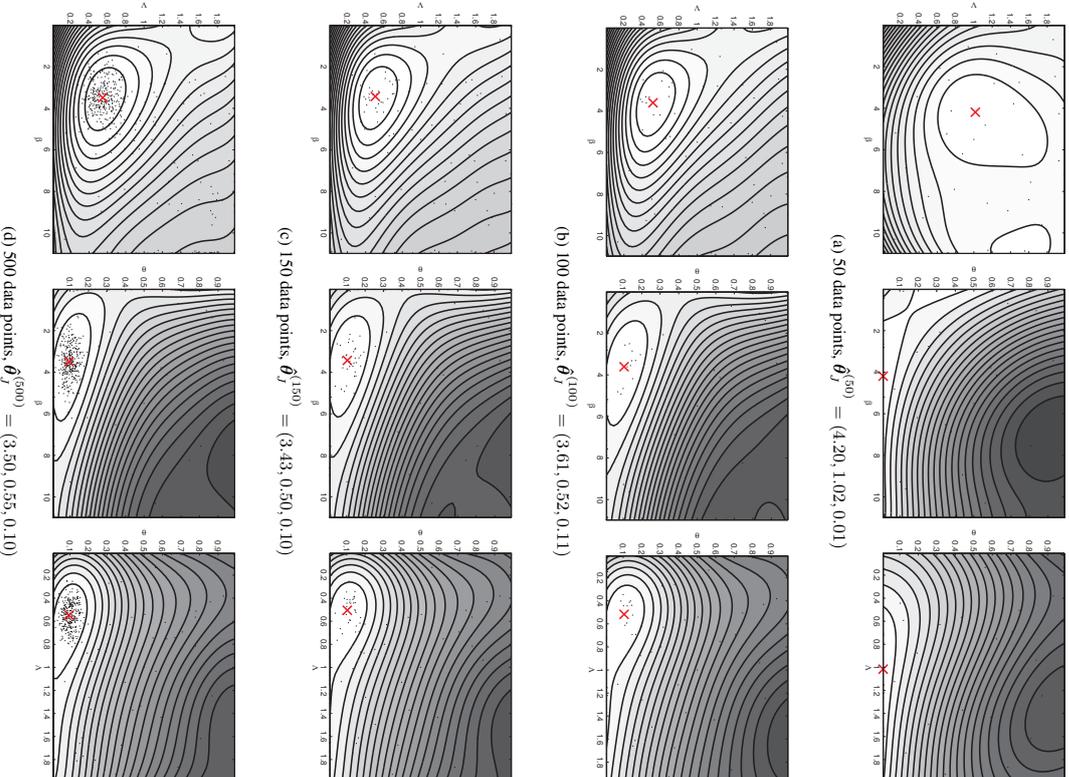


Figure 11: Isocontours of the estimated regression function $\hat{J}^{(t)}$ for the day care center model. Visualization is as in Figure 9.

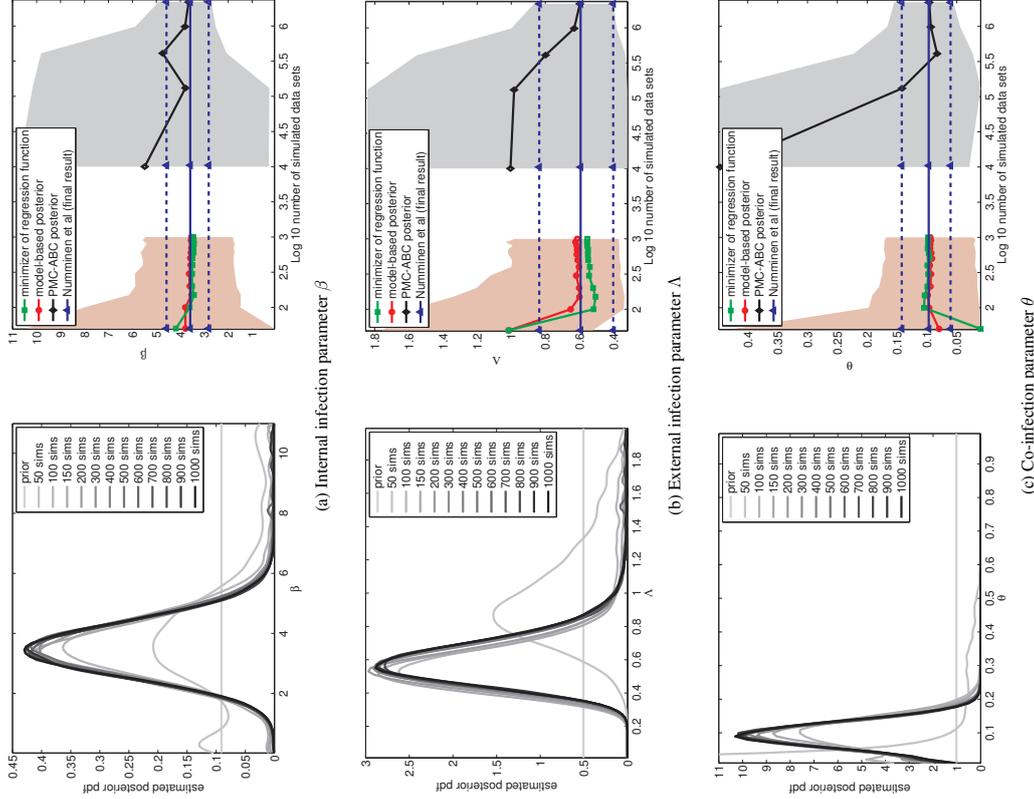


Figure 12: Marginal posterior results for the day care center model. The left column shows the obtained model-based posterior pdfs. The right column compares the posterior mean and the 95% credibility interval with results from PMc-ABC algorithms. We obtain conservative estimates of the model parameters at a fraction of the computational cost. Posterior means are shown as solid lines with markers, credibility intervals as shaded areas or dashed lines.

8. Conclusions

Our paper dealt with inferring the parameters of simulator-based (generative) statistical models. Inference is difficult for such models because of the intractability of the likelihood function. While it is an open question whether variational principles are also applicable, the parameters of simulator-based statistical models are typically inferred by finding values for which the discrepancy between simulated and observed data tends to be small. We have seen that such an approach is computationally costly. The high cost is largely due to a lack of knowledge about the functional relation between the model parameters and the discrepancies. We proposed to use regression to infer the relation using training data which are actively acquired. The acquisition is performed such that the focus in the regression is on regions in the parameter space where the discrepancy tends to be small. We implemented the proposed strategy using Bayesian optimization where the discrepancy is modeled with a Gaussian process. The posterior distribution of the Gaussian process was used to construct a model-based approximation of the intractable likelihood. This combination of probabilistic modeling and optimization reduced the number of simulated data sets by several orders of magnitude in our applications. The reduction in the number of required simulations accelerated the inference substantially.

Our approach is related to the work by Rasmussen (2003) and the two recent papers by Wilkinson (2014) and Meeds and Welling (2014) (which became only available after we first proposed our approach at “ABC in Rome” in 2013): Rasmussen (2003) used a Gaussian process to model the logarithm of the target pdf in a hybrid Monte Carlo algorithm. There are two main differences to our work. First, a scenario was considered where the target can be evaluated exactly at a finite computational cost, even though the cost might be high. In our case, exact evaluation of the likelihood function is not assumed possible at finite cost. This difference is important because approximate likelihood evaluations might be rather noisy. The second difference is that we used Bayesian optimization to focus on the modal areas of the target.

Related to the approach of Rasmussen (2003), Wilkinson (2014) modeled the log likelihood as a Gaussian process. This is different from our work where we model the discrepancies. We believe that modeling the discrepancies is advantageous because it allows to delay the selection of the kernel and bandwidth which are needed in the nonparametric setting. This is important because it enables one to make use of all simulated data. In the parametric setting, the two modeling strategies lead to identical solutions. We found further that accurate point estimates can be obtained by modeling the discrepancies only. In particular, minimizing their regression function corresponds to maximizing a lower bound of the approximate nonparametric likelihood under mild conditions. As a second difference, Wilkinson (2014) used space-filling points together with a plausibility criterion to obtain the parameter values for the regression. This is in contrast to Bayesian optimization where powerful optimization methods are employed to quickly identify the areas of interest.

Meeds and Welling (2014) proposed an alternative to the sample average approximation of the (limiting) synthetic likelihood by modeling each element of the intractable mean and covariance matrix of the summary statistics with a Gaussian process. The resulting likelihood approximation was used together with a Markov chain Monte Carlo algorithm for

posterior inference. The differences to our approach lie in the quantities modeled and in the use Bayesian optimization to actively design the training data for the Gaussian processes.

There are also connections to the body of work on Bayesian analysis of computer codes (for an introduction to this field of research, see for example the paper by O’Hagan, 2006): Sacks et al. (1989) and Currin et al. (1991) modeled the outputs of general deterministic computer codes as Gaussian processes. The computer codes were, for example, solving complex partial differential equations, and the papers were about finding an emulator for the heavy computations. Inference of unknown parameters of the computer codes given observed data was only considered later by Cox et al. (2001) and Kennedy and O’Hagan (2001). The observed and simulated data were modeled using Gaussian processes, and space-filling points were used to choose the parameters for which the computer code was run. The main differences to our approach are again the quantities modeled, and the use of Bayesian optimization.

We employed a rather basic algorithm to perform Bayesian optimization. This does, however, not mean that Bayesian optimization for likelihood-free inference is limited to that particular algorithm. We discussed a number of alternatives, as well as more advanced algorithms which could be used instead, and outlined a general framework for increasing the computational efficiency of likelihood-free inference.

Our paper opens up a wide range of extensions and opportunities for future research. One possibility is to use the tools provided by Bayesian optimization to tackle the challenging problem of likelihood-free inference in high dimensions. More foundational research topics would revolve around the modeling of the discrepancies and the development of acquisition rules which are tailored to the problem of likelihood-free inference. We focused on approximating the modal areas of the intractable likelihoods more accurately than the tails. It is an open question of how to best increase the accuracy in the tail areas. One possibility is to use the samples from the approximate posterior to update the training data for the regression, which would naturally lead to a recursion where the current method would only provide the initial approximation.

Acknowledgements

This work was partially supported by ERC grant no. 239784 and the Academy of Finland (Finnish Centre of Excellence in Computational Inference Research COIN, 251170). MTUG thanks Paul Blomstedt for helpful comments on an early draft of the paper.

Author contributions: MTUG proposed, designed, and performed research, and wrote the paper; JC contributed to research design and writing.

Appendix A. Proof of Proposition 1

We split the objective \hat{J}_g^N , defined in Equation (32), into two terms,

$$\hat{J}_g^N(\boldsymbol{\theta}) = T_1(\boldsymbol{\theta}) + T_2(\boldsymbol{\theta}), \quad (48)$$

$$T_1(\boldsymbol{\theta}) = \log |\det \mathbf{C}_{\theta_1}|, \quad (49)$$

$$T_2(\boldsymbol{\theta}) = E^N \left[(\Phi_o - \Phi_{\theta})^\top \mathbf{C}_{\theta}^{-1} (\Phi_o - \Phi_{\theta}) \right]. \quad (50)$$

Term T_2 can be rewritten using the empirical mean $\hat{\boldsymbol{\mu}}_{\theta}$ and the covariance matrix $\hat{\boldsymbol{\Sigma}}_{\theta}$ in Equation (14),

$$T_2(\boldsymbol{\theta}) = E^N \left[(\Phi_o - \hat{\boldsymbol{\mu}}_{\theta} + \hat{\boldsymbol{\mu}}_{\theta} - \Phi_{\theta})^\top \mathbf{C}_{\theta}^{-1} (\Phi_o - \hat{\boldsymbol{\mu}}_{\theta} + \hat{\boldsymbol{\mu}}_{\theta} - \Phi_{\theta}) \right] \quad (51)$$

$$= E^N \left[(\Phi_o - \hat{\boldsymbol{\mu}}_{\theta})^\top \mathbf{C}_{\theta}^{-1} (\Phi_o - \hat{\boldsymbol{\mu}}_{\theta}) + (\hat{\boldsymbol{\mu}}_{\theta} - \Phi_{\theta})^\top \mathbf{C}_{\theta}^{-1} (\hat{\boldsymbol{\mu}}_{\theta} - \Phi_{\theta}) \right. \\ \left. + 2(\Phi_o - \hat{\boldsymbol{\mu}}_{\theta})^\top \mathbf{C}_{\theta}^{-1} (\hat{\boldsymbol{\mu}}_{\theta} - \Phi_{\theta}) \right] \quad (52)$$

$$= (\Phi_o - \hat{\boldsymbol{\mu}}_{\theta})^\top \mathbf{C}_{\theta}^{-1} (\Phi_o - \hat{\boldsymbol{\mu}}_{\theta}) + \text{tr} \left(\mathbf{C}_{\theta}^{-1} E^N \left[(\hat{\boldsymbol{\mu}}_{\theta} - \Phi_{\theta}) (\hat{\boldsymbol{\mu}}_{\theta} - \Phi_{\theta})^\top \right] \right) \quad (53)$$

$$= (\Phi_o - \hat{\boldsymbol{\mu}}_{\theta})^\top \mathbf{C}_{\theta}^{-1} (\Phi_o - \hat{\boldsymbol{\mu}}_{\theta}) + \text{tr} \left(\mathbf{C}_{\theta}^{-1} \hat{\boldsymbol{\Sigma}}_{\theta} \right), \quad (54)$$

where we have used that $E^N[\Phi_{\theta}] = \hat{\boldsymbol{\mu}}_{\theta}$. For $\mathbf{C}_{\theta} = \hat{\boldsymbol{\Sigma}}_{\theta}$, we have

$$T_2(\boldsymbol{\theta}) = (\Phi_o - \hat{\boldsymbol{\mu}}_{\theta})^\top \hat{\boldsymbol{\Sigma}}_{\theta}^{-1} (\Phi_o - \hat{\boldsymbol{\mu}}_{\theta}) + p. \quad (55)$$

Hence, for $\mathbf{C}_{\theta} = \hat{\boldsymbol{\Sigma}}_{\theta}$, \hat{J}_g^N equals

$$\hat{J}_g^N(\boldsymbol{\theta}) = \log |\det \hat{\boldsymbol{\Sigma}}_{\theta}| + (\Phi_o - \hat{\boldsymbol{\mu}}_{\theta})^\top \hat{\boldsymbol{\Sigma}}_{\theta}^{-1} (\Phi_o - \hat{\boldsymbol{\mu}}_{\theta}) + p. \quad (56)$$

On the other hand, the log synthetic likelihood $\hat{\ell}_s^N$ is

$$\hat{\ell}_s^N(\boldsymbol{\theta}) = -\frac{p}{2} \log(2\pi) - \frac{1}{2} \log |\det \hat{\boldsymbol{\Sigma}}_{\theta}| - \frac{1}{2} (\Phi_o - \hat{\boldsymbol{\mu}}_{\theta})^\top \hat{\boldsymbol{\Sigma}}_{\theta}^{-1} (\Phi_o - \hat{\boldsymbol{\mu}}_{\theta}), \quad (57)$$

so that

$$\hat{J}_g^N(\boldsymbol{\theta}) = p - p \log(2\pi) - 2\hat{\ell}_s^N(\boldsymbol{\theta}). \quad (58)$$

The claimed result follows now from Equation (31),

$$\log L_g^N(\boldsymbol{\theta}) \geq -\frac{p}{2} + \hat{\ell}_s^N(\boldsymbol{\theta}). \quad (59)$$

Replacing the empirical average E^N with the expectation shows that the limiting quantities $\hat{\ell}_s$ and $J_g(\boldsymbol{\theta})$,

$$J_g(\boldsymbol{\theta}) = E[\Delta_{\theta_1}^g], \quad (60)$$

are related by an analogous result. In more detail,

$$J_g(\boldsymbol{\theta}) = \log |\det \mathbf{C}_{\theta_1}| + E \left[(\Phi_o - \Phi_{\theta})^\top \mathbf{C}_{\theta}^{-1} (\Phi_o - \Phi_{\theta}) \right] \quad (61)$$

$$= \log |\det \mathbf{C}_{\theta_1}| + (\Phi_o - \boldsymbol{\mu}_{\theta})^\top \mathbf{C}_{\theta}^{-1} (\Phi_o - \boldsymbol{\mu}_{\theta}) + \text{tr} \left(\mathbf{C}_{\theta}^{-1} \boldsymbol{\Sigma}_{\theta} \right), \quad (62)$$

where we used the same development which led to Equation (54) but with the expectation instead of E^N . Hence, for $\mathbf{C}_\theta = \Sigma_\theta$, we have the analogous result by definition of $\tilde{\ell}_s$ in Equation (13),

$$J_g(\theta) = p - p \log(2\pi) - 2\tilde{\ell}_s(\theta). \quad (63)$$

It follows by definition of J_g that $\tilde{\ell}_s$ can be seen as a regression function where θ is the vector of covariates and Δ_θ^g is, up to constants and the sign, the response variable.

Appendix B. Using the Prior Distribution of the Parameters in Bayesian Optimization

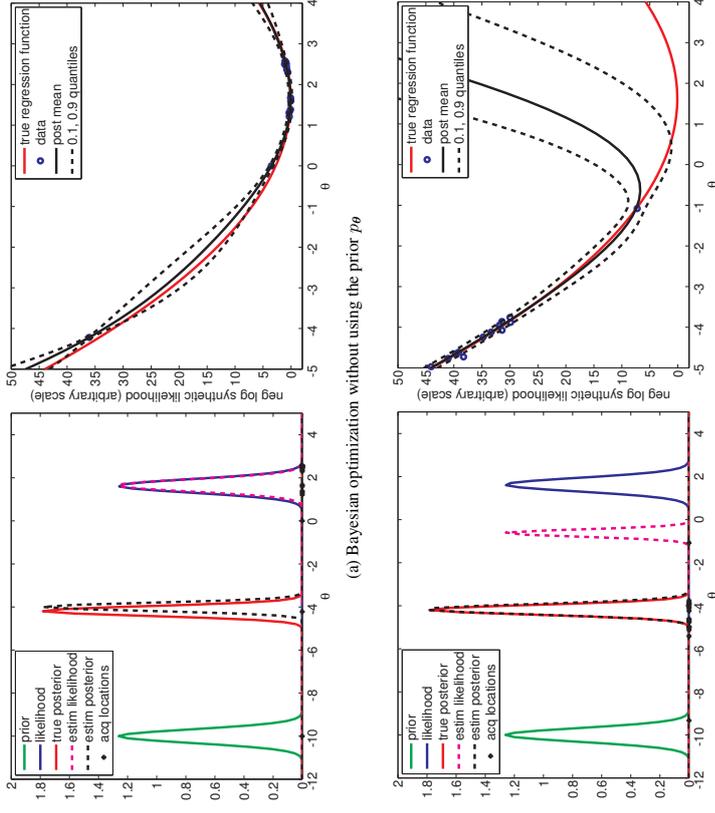
In the main text, we focused on acquiring training data in regions in the parameter space where the discrepancy Δ_θ tends to be small, which corresponds to the modal regions of the approximate likelihoods. For highly informative priors p_θ with modal regions far away from the peaks of the likelihood such an approach is suboptimal for posterior inference. Since the prior is typically fairly broad and the likelihood peaked, this situation is not usual. But if it happens, it is better to directly acquire the training data in the modal areas of the posterior. For inference via the synthetic likelihood, this can be straightforwardly done by approximating $\tilde{\ell}_s + \log p_\theta$. In Bayesian optimization with Δ_θ^g as the response variable, the posterior mean μ_t in Equation (43) would then be replaced by $\hat{\mu}_t(\theta) = \mu_t(\theta) - 2 \log p_\theta$. For inference via a nonparametric approximation of the likelihood, the same approach may also work but this warrants further investigations because the regression function J provides only a lower bound for the likelihood. We also note that using the prior p_θ can be helpful if it is known that the parameters do not influence the model independently, causing for instance the discrepancy to be nearly constant along certain directions in the parameter space.

Figure 13 illustrates the basic idea using Example 1 and a prior pdf p_θ (blue curve) which has practically no overlap with the true likelihood L (green curve). The results are for Bayesian optimization with 20 deterministic acquisitions and a Gaussian process model with constant mean function.

Appendix C. Bayesian Optimization with a Deterministic versus a Stochastic Acquisition Rule

Example 10 illustrated log-Gaussian modeling and the stochastic acquisition rule by means of the Ricker model with the log growth rate $\log r$ as only unknown. We here show the differences between stochastic and deterministic acquisitions in greater detail. The results are for a log-Gaussian process model.

Figure 14 shows the estimated regression functions $\hat{J}^{(t)}$ as obtained with a deterministic acquisition rule like in Figure 7(d) for different t . The acquired data points are vertically clustered because the acquisition rule often proposed nearly identical parameters. Figure 15 shows $\hat{J}^{(t)}$ obtained with a stochastic acquisition rule as in Figure 7(f). While both methods lead to a satisfactory approximation of the negative log synthetic likelihood around its minimum, the result with the stochastic acquisition rule seems more stable because the acquired training data are spread out more evenly in the interval of interest.



(a) Bayesian optimization without using the prior p_θ

(b) Bayesian optimization with the prior p_θ during the acquisitions

Figure 13: Using the prior density p_θ in Bayesian optimization. (a) If p_θ is not used (or if uniform), the focus is on the modal region of the likelihood. If the prior is far from the mode of the likelihood, the learned model is less accurate in the modal areas of the posterior (black dashed versus red solid curve). (b) The prior pdf p_θ was used to shift the data acquisitions in Bayesian optimization to the modal area of the posterior (see the circles in the figures on the right or on the x-axes on the left). This results in a more accurate approximation of the posterior pdf but a less accurate approximation of the mode of the likelihood (dashed magenta versus blue solid curve).

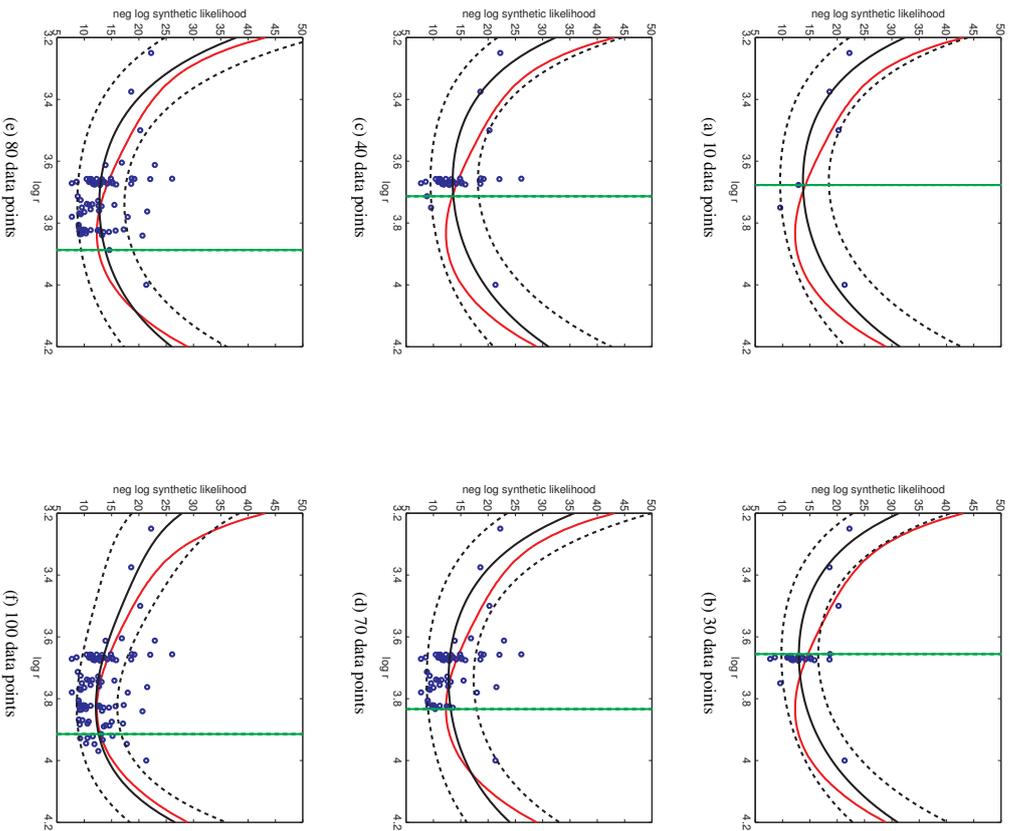


Figure 14: Log-Gaussian process model for the log synthetic likelihood of the Ricker model with $\log r$ as only unknown. The results are for the deterministic acquisition rule consisting of minimization of the acquisition function in Equation (45). Note the vertical clusters. The visualization is as in Figure 7. The plot range was restricted to $(3.2, 4.2)$ so that not all acquisition may be shown.

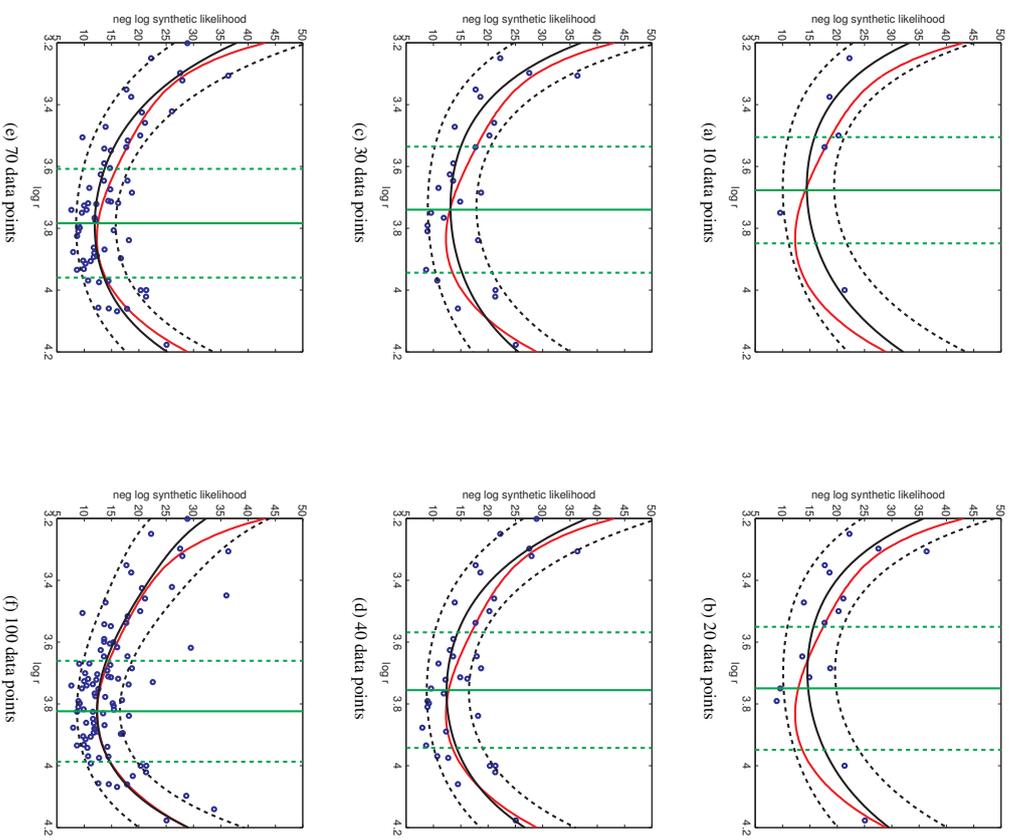


Figure 15: Log-Gaussian process model for the log synthetic likelihood of the Ricker model with $\log r$ as only unknown. The setup and visualization is as in Figure 14 but the stochastic acquisition rule is used. A movie showing the acquisitions and the updating of the model is available at <http://www.jmlr.org/papers/volume17/15-017/supplementary/RickerID.avi>.

Appendix D. Ricker Model Inferred with a Markov Chain Monte Carlo Algorithm

We here report the simulation results for the Ricker model inferred with the log synthetic likelihood \hat{L}_s^N and a random walk MCMC algorithm with the code made publicly available by Wood (2010). We ran the algorithm for 100,000 iterations, starting at $\theta_0 = (3.8, 0.3, 10)$. The first 25,000 samples were discarded. In the work by Wood (2010), the proposal standard deviation for σ was 0.1. Figure 16 shows that this choice led to a chain which got stuck close to $\sigma = 0$ even when $N = 5,000$ (blue, squares). Reducing the proposal standard deviation by a factor of 10 allowed us to obtain reasonable results (red, circles). The proposal standard deviations for the remaining parameters were the same as in the original publication. We then investigated the stability of the inferred posteriors when N is reduced from $N = 5,000$ to $N = 500$ and when the simulator is run with different realizations of the random log synthetic likelihood. Figure 17 shows that the posteriors are stable for $\log r$ and ϕ but that there is some variation for σ .

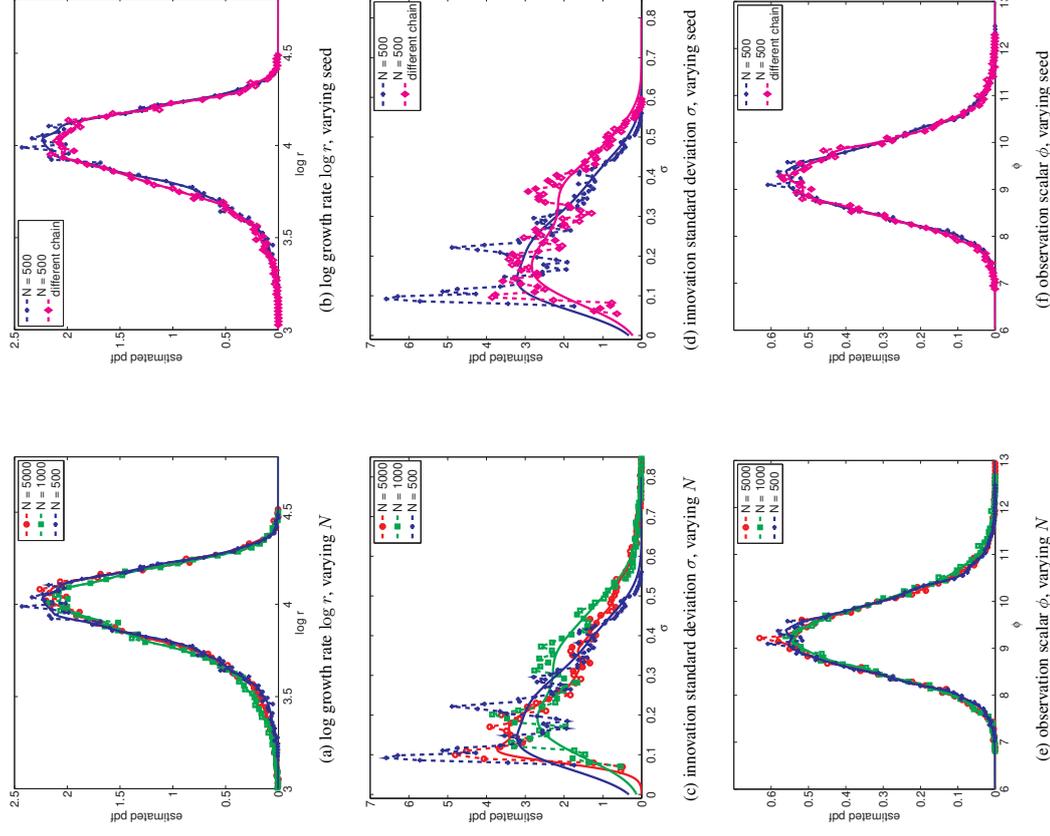


Figure 17: Effect of the number of simulated data sets N (left column) and the seed of the random number generator (right column) for the Ricker example when inferred with the method by Wood (2010). Visualization is as in Figure 16.

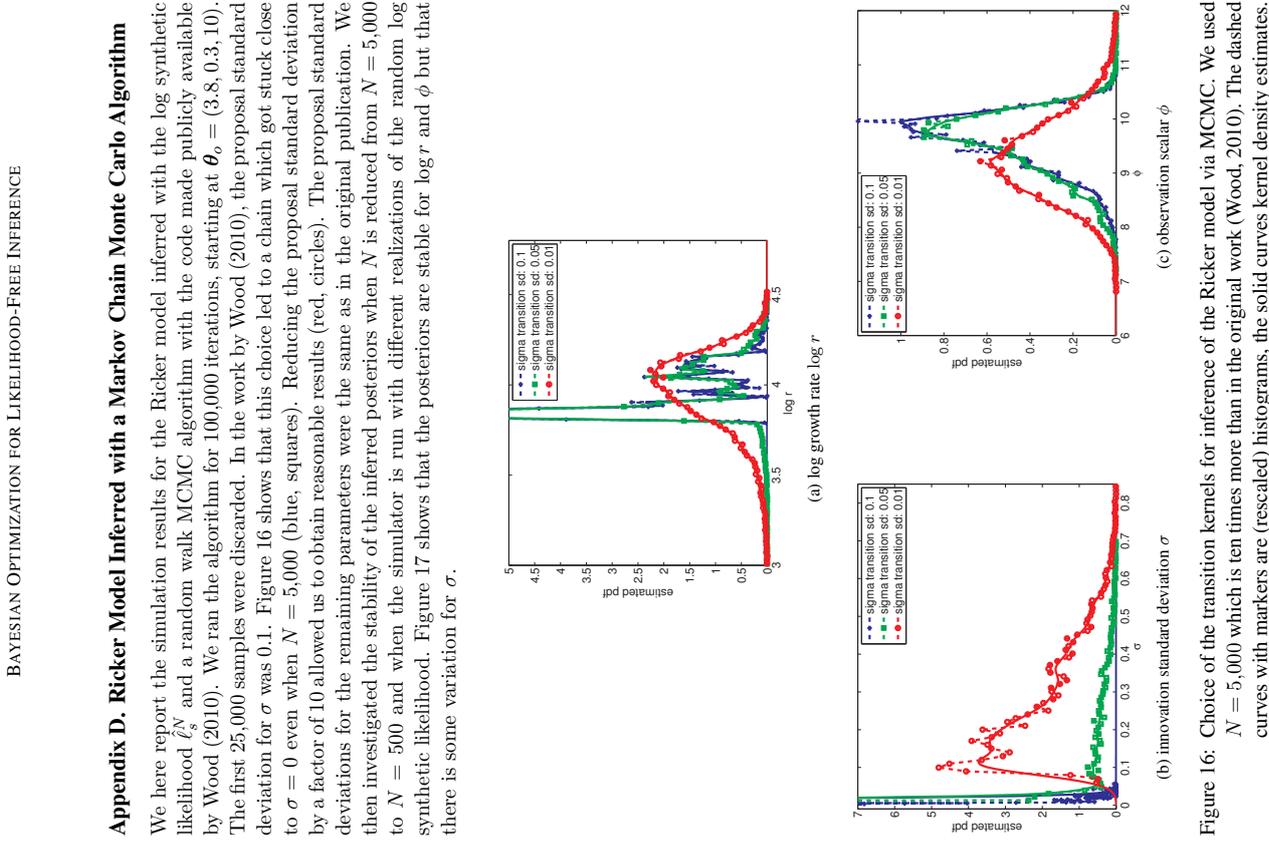


Figure 16: Choice of the transition kernels for inference of the Ricker model via MCMC. We used $N = 5,000$ which is ten times more than in the original work (Wood, 2010). The dashed curves with markers are (rescaled) histograms, the solid curves kernel density estimates.

References

- S. Aeschbacher, M.A. Beaumont, and A. Futschik. A novel approach for choosing summary statistics in approximate Bayesian computation. *Genetics*, 192(3):1027–1047, 2012.
- J. Azimi, A. Fern, and X.Z. Fern. Batch Bayesian optimization via simulation matching. In *Advances in Neural Information Processing Systems 23 (NIPS)*, 2010.
- M.A. Beaumont, W. Zhang, and D.J. Balding. Approximate Bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, 2002.
- M.A. Beaumont, J.-M. Cornuet, J.-M. Marin, and C.P. Robert. Adaptive approximate Bayesian computation. *Biometrika*, 96(4):983–990, 2009.
- M. Blum. Approximate Bayesian computation: A nonparametric perspective. *Journal of the American Statistical Association*, 105(491):1178–1187, 2010.
- E. Brochu, V.M. Cora, and N. de Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv:1012.2599*, 2010.
- O. Cappé, A. Guillin, J.M. Marin, and C.P. Robert. Population Monte Carlo. *Journal of Computational and Graphical Statistics*, 13(4):907–929, 2004.
- O. Chapelle and L. Li. An empirical evaluation of Thompson sampling. In *Advances in Neural Information Processing Systems 24 (NIPS)*, 2011.
- B. Chen, A. Krause, and R.M. Castro. Joint optimization and variable selection of high-dimensional Gaussian processes. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012.
- E. Contal, D. Buffoni, A. Robicquet, and N. Vayatis. Parallel Gaussian process optimization with upper confidence bound and pure exploration. In *Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*, 2013.
- D.D. Cox and S. John. A statistical method for global optimization. In *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, 1992.
- D.D. Cox and S. John. SDO: A statistical method for global optimization. In *Multiscale-Planary Design Optimization: State-of-the-Art*, 1997.
- D.D. Cox, J.-S. Park, and C.E. Singer. A statistical method for tuning a computer code to a data base. *Computational Statistics & Data Analysis*, 37(1):77–92, 2001.
- C. Curran, T. Mitchell, M. Morris, and D. Yvisaker. Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association*, 86(416):953–963, 1991.
- P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.
- T. Desautels, A. Krause, and J.W. Burdick. Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization. *Journal of Machine Learning Research*, 15:3873–3923, 2014.
- P.J. Diggle and R.J. Gratton. Monte Carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 46(2):193–227, 1984.
- J. Djolonga, A. Krause, and V. Cevher. High-dimensional Gaussian process bandits. In *Advances in Neural Information Processing Systems 26 (NIPS)*, 2013.
- P. Fearnhead and D. Prangle. Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(3):419–474, 2012.
- K. Fukumizu, L. Song, and A. Gretton. Kernel Bayes’ rule: Bayesian inference with positive definite kernels. *Journal of Machine Learning Research*, 14:3753–3783, 2013.
- C. Gouriéroux, A. Monfort, and E. Renault. Indirect inference. *Journal of Applied Econometrics*, 8(S1):S85–S118, 1993.
- M.U. Gutmann and J. Hyyriäinen. Bregman divergence as general framework to estimate unnormalized statistical models. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2011.
- M.U. Gutmann and A. Hyyriäinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13:307–361, 2012.
- M.U. Gutmann, R. Dutta, S. Kaski, and J. Corander. Likelihood-free inference via classification. *arXiv:1407.4981*, 2014.
- F. Harig, J.M. Calabrese, B. Reinking, T. Wiegand, and A. Huth. Statistical inference for stochastic simulation models – theory and application. *Ecology Letters*, 14(8):816–827, 2011.
- J. Hensman, N. Fusi, and N.D. Lawrence. Gaussian processes for big data. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2013.
- G.E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.
- A. Hyyriäinen. Estimation of non-normalized statistical models using score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.
- D.R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.
- P. Joyce and P. Martjoram. Approximately sufficient statistics and Bayesian computation. *Statistical Applications in Genetics and Molecular Biology*, 7(1):Article 26, 2008.

- K. Kandasamy, J. Schneider, and B. Póczos. High dimensional Bayesian optimisation and bandits via additive models. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.
- M.C. Kennedy and A. O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.
- J. Kim and C.D. Scott. Robust kernel density estimation. *Journal of Machine Learning Research*, 13:2529–2565, 2012.
- Y.P. Mack and M. Rosenblatt. Multivariate k-nearest neighbor density estimates. *Journal of Multivariate Analysis*, 9(1):1–15, 1979.
- J.-M. Marin, P. Pudlo, C.P. Robert, and R.J. Ryder. Approximate Bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180, 2012.
- P. Marjoram, J. Molitor, V. Plagnol, and S. Tavaré. Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26):15324–15328, 2003.
- E. Meeds and M. Welling. GPS-ABC: Gaussian process surrogate approximate Bayesian computation. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2014.
- H. Niederreiter. Low-discrepancy and low-dispersion sequences. *Journal of Number Theory*, 30(1):51–70, 1988.
- E. Numminen, L. Cheng, M. Gyllenberg, and J. Corander. Estimating the transmission dynamics of Streptococcus pneumoniae from strain prevalence data. *Biometrics*, 69(3):748–757, 2013.
- M.A. Nunes and D.J. Balding. On optimal selection of summary statistics for approximate Bayesian computation. *Statistical Applications in Genetics and Molecular Biology*, 9(1): Article 34, 2010.
- A. O'Hagan. Bayesian analysis of computer code outputs: A tutorial. *Reliability Engineering & System Safety*, 91(10–11):1290–1300, 2006.
- E. Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.
- M. Phlajaja, M.U. Gutmann, and A. Hyvärinen. A family of computationally efficient and simple estimators for unnormalized statistical models. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2010.
- J.K. Pritchard, M.T. Seielstad, A. Perez-Lezaun, and M.W. Feldman. Population growth of human Y chromosomes: A study of Y chromosome microsatellites. *Molecular Biology and Evolution*, 16(12):1791–1798, 1999.
- J. Quiñero-Candela and C.E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.
- GUTMANN AND CORANDER
- C.E. Rasmussen. Gaussian processes to speed up hybrid Monte Carlo for expensive Bayesian integrals. In *Bayesian Statistics 7: The 7th Valencia International Meeting*, 2003.
- C.E. Rasmussen and C.K.I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.
- W.E. Ricker. Stock and recruitment. *Journal of the Fisheries Research Board of Canada*, 11(5):559–623, 1954.
- C.P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, 2004.
- M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837, 1956.
- D. Russo and B. Van Roy. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014.
- J. Sacks, W.J. Welch, T.J. Mitchell, and H.P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4:409–435, 1989.
- A. Shah, A. Wilson, and Z. Ghahramani. Student-t processes as alternatives to Gaussian processes. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2014.
- S.A. Sisson, Y. Fan, and M.M. Tanaka. Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 104(6):1760–1765, 2007.
- J. Snoek, H. Larochelle, and R.P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25 (NIPS)*, 2012.
- J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M.M.A. Patwary, Prabhakar, and R.P. Adams. Scalable Bayesian optimization using deep neural networks. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.
- N. Srinivas, A. Krause, M. Seeger, and S.M. Kakade. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv:0912.3995v2*, 2010.
- N. Srinivas, A. Krause, S.M. Kakade, and M. Seeger. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.
- W.R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.
- M. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009.

- T. Toni, D. Welch, N. Strelkowa, A. Ipsen, and M.P.H. Stumpf. Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, 6(31):187–202, 2009.
- M.P. Wand and M.C. Jones. *Kernel Smoothing*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability, 1995.
- Z. Wang, M. Zoghi, F. Hutter, D. Matheson, and N. de Freitas. Bayesian optimization in high dimensions via random embeddings. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2013.
- L. Wasserman. *All of statistics*. Springer, 2004.
- D. Wegmann, C. Lenzenberger, and L. Excoffier. Efficient approximate Bayesian computation coupled with Markov chain Monte Carlo without likelihood. *Genetics*, 182(4): 1207–1218, 2009.
- R. Wilkinson. Accelerating ABC methods using Gaussian processes. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2014.
- S.N. Wood. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466(7310):1102–1104, 2010.

On Lower and Upper Bounds in Smooth and Strongly Convex Optimization

Yossi Arjevani

*Department of Computer Science and Applied Mathematics
Weizmann Institute of Science
Rehovot 7610001, Israel*

YOSSI.ARJEVANI@WEIZMANN.AC.IL

Shai Shalev-Shwartz

*School of Computer Science and Engineering
The Hebrew University
Givat Ram, Jerusalem 9190401, Israel*

SHAIS@CS.HUJI.AC.IL

Ohad Shamir

*Department of Computer Science and Applied Mathematics
Weizmann Institute of Science
Rehovot 7610001, Israel*

OHAD.SHAMIR@WEIZMANN.AC.IL

Editor: Mark Schmidt

Abstract

We develop a novel framework to study smooth and strongly convex optimization algorithms. Focusing on quadratic functions we are able to examine optimization algorithms as a recursive application of linear operators. This, in turn, reveals a powerful connection between a class of optimization algorithms and the analytic theory of polynomials whereby new lower and upper bounds are derived. Whereas existing lower bounds for this setting are only valid when the dimensionality scales with the number of iterations, our lower bound holds in the natural regime where the dimensionality is fixed. Lastly, expressing it as an optimal solution for the corresponding optimization problem over polynomials, as formulated by our framework, we present a novel systematic derivation of Nesterov's well-known Accelerated Gradient Descent method. This rather natural interpretation of AGD contrasts with earlier ones which lacked a simple, yet solid, motivation.

Keywords: smooth and strongly convex optimization, full gradient descent, accelerated gradient descent, heavy ball method

1. Introduction

In the field of mathematical optimization one is interested in efficiently solving a minimization problem of the form

$$\min_{\mathbf{x} \in X} f(\mathbf{x}), \quad (1)$$

where the *objective function* f is some real-valued function defined over the *constraints set* X . Many core problems in the field of Computer Science, Economic, and Operations Research can be readily expressed in this form, rendering this minimization problem far-reaching. That being said, in its full generality this problem is just too hard to solve or

even to approximate. As a consequence, various structural assumptions on the objective function and the constraints set, along with better-suited optimization algorithms, have been proposed so as to make this problem viable.

One such case is smooth and strongly convex functions over some d -dimensional Euclidean space¹. Formally, we consider continuously differentiable $f : \mathbb{R}^d \rightarrow \mathbb{R}$ which are L -smooth, i.e.,

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|, \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d,$$

and μ -strongly convex, that is,

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \mathbf{y} - \mathbf{x}, \nabla f(\mathbf{x}) \rangle + \frac{\mu}{2} \|\mathbf{y} - \mathbf{x}\|^2, \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d.$$

A wide range of applications together with very efficient solvers have made this family of problems very important. Naturally, an interesting question arises: how fast can these kind of problems be solved? better said, what is the computational complexity of minimizing smooth and strongly-convex functions to a given degree of accuracy?² Prior to answering these, otherwise ill-defined, questions, one must first address the exact nature of the underlying computational model.

Although being a widely accepted computational model in the theoretical computer sciences, the Turing Machine Model presents many obstacles when analyzing optimization algorithms. In their seminal work, Nemirovsky and Yudin (1983) evaded some of these difficulties by proposing the *black box computational model*, according to which information regarding the objective function is acquired iteratively by querying an *oracle*. This model does not impose any computational resource constraints³. Nemirovsky and Yudin showed that for any optimization algorithm which employs a first-order oracle, i.e. receives $(f(\mathbf{x}), \nabla f(\mathbf{x}))$ upon querying at a point $\mathbf{x} \in \mathbb{R}^d$, there exists an L -smooth μ -strongly convex quadratic function $f : \mathbb{R}^d \rightarrow \mathbb{R}$, such that for any $\epsilon > 0$ the number of oracle calls needed for obtaining an ϵ -optimal solution $\bar{\mathbf{x}}$, i.e.,

$$f(\bar{\mathbf{x}}) < \min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x}) + \epsilon, \quad (2)$$

must satisfy

$$\# \text{ Oracle Calls} \geq \tilde{\Omega} \left(\min \{d, \sqrt{\kappa} \ln(1/\epsilon)\} \right), \quad (3)$$

where $\kappa \triangleq L/\mu$ denotes the so-called *condition number*.

1. More generally, one may consider smooth and strongly convex functions over some Hilbert space.
2. Natural as these questions might look today, matters were quite different only few decades ago. In his book 'Introduction to Optimization' which dates back to 87', Polyak B.T devotes a whole section as to: 'Why Are Convergence Theorems Necessary?' (See section 1.6.2 in Polyak (1987)).
3. In a sense, this model is dual to the Turing Machine model where all the information regarding the parameters of the problem is available prior to the execution of the algorithm, but the computational resources are limited in time and space.

The result of Nemirovsky and Yudin can be seen as the starting point of the present paper. The restricted validity of this lower bound to the first $\mathcal{O}(d)$ iterations is not a mere artifact of the analysis. Indeed, from an information point of view, a minimizer of any convex quadratic function can be found using no more than $\mathcal{O}(d)$ first-order queries. Noticing that this bound is attained by the Conjugate Gradient Descent method (CGD, see Polyak 1987), it seems that one cannot get a non-trivial lower bound once the number of queries exceeds the dimension d . Moreover, a similar situation can be shown to occur for more general classes of convex functions. However, the known algorithms which attain such behavior (such as CGD and the center-of-gravity method, e.g., Nemirovski 2005) require computationally intensive iterations, and are quite different than many common algorithms used for large-scale optimization problems, such as gradient descent and its variants. Thus, to capture the attainable performance of such algorithms, we must make additional assumptions on their structure. This can be made more solid using the following simple observation.

When applied on quadratic functions, the update rule of many optimization algorithms reduces to a recursive application of a linear transformation which depends, possibly randomly, on the previous p query points.

Indeed, the update rule of CGD for quadratic functions is *non-stationary*, i.e. uses a different transformation at each iteration, as opposed to other optimization algorithms which utilize less complex update rules such as: stationary updates rule, e.g., Gradient Descent, Accelerated Gradient Descent, Newton's method (see Nesterov 2004), The Heavy Ball method Polyak (1987), SDCA (see Shalev-Shwartz and Zhang 2013) and SAG (see Roux et al. 2012); cyclic update rules, e.g., SVRG (see Johnson and Zhang 2013); and piecewise-stationary update rules, e.g., Accelerated SDCA. Inspired by this observation, in the present work we explore the boundaries of optimization algorithms which admit stationary update rules. We call such algorithms p -Stationary Canonical Linear Iterative optimization algorithms (abbr. p -SCLI), where p designates the number of previous points which are necessary to generate new points. The quantity p may be instructively interpreted as a limit on the amount of memory at the algorithm's disposal.

Similar to the analysis of power iteration methods, the convergence properties of such algorithms are intimately related to the eigenvalues of the corresponding linear transformation. Specifically, as the convergence rate of a recursive application of a linear transformation is essentially characterized by its largest magnitude eigenvalue, the asymptotic convergence rate of p -SCLI algorithms can be bounded from above and from below by analyzing the spectrum of the corresponding linear transformation. It should be noted that the technique of linearizing iterative procedures and analyzing their convergence behavior accordingly, which dates back to the pioneering work of the Russian mathematician Lyapunov, has been successfully applied in the field of mathematical optimization many times, e.g., Polyak (1987) and more recently Lessard et al. (2014). However, whereas previous works were primarily concerned with deriving upper bounds on the magnitude of the corresponding eigenvalues, in this work our reference point is lower bounds.

As eigenvalues are merely roots of characteristic polynomials⁴, our approach involves establishing a lower bound on the maximal modulus (absolute value) of the roots of polynomials. Clearly, in order to find a meaningful lower bound, one must first find a condition which is satisfied by all characteristic polynomials that correspond to p -SCLIs. We show that such condition does exist by proving that characteristic polynomials of consistent p -SCLIs, which correctly minimize the function at hand, must have a specific evaluation at $\lambda = 1$. This in turn allows us to analyze the convergence rate purely in terms of the analytic theory of polynomials, i.e.,

$$\mathbf{Find} \quad \min \{ \rho(q(z)) \mid q(z) \text{ is a real monic polynomial of degree } p \text{ and } q(1) = r^p \}, \quad (4)$$

where $r \in \mathbb{R}$ and $\rho(q(z))$ denotes the maximum modulus over all roots of $q(z)$. Although a vast range of techniques have been developed for bounding the moduli of roots of polynomials (e.g., Marden 1966; Rahman and Schmeisser 2002; Mhovanovic et al. 1994; Walsh 1922; Mhovanovic and Rassias 2000; Fell 1980), to the best of our knowledge, few of them address lower bounds (see Higham and Tisseur 2003). Minimization problem (4) is also strongly connected with the question of bounding the spectral radius of 'generalized' companion matrices from below. Unfortunately, this topic too lacks an adequate coverage in the literature (see Volkovitz and Stryan 1980; Zhong and Huang 2008; Horne 1997; Huang and Wang 2007). Consequently, we devote part of this work to establish new tools for tackling (4). It is noteworthy that these tools are developed by using elementary arguments. This sharply contrasts with previously proof techniques used for deriving lower bounds on the convergence rate of optimization algorithms which employed heavy machinery from the field of extremal polynomials, such as Chebyshev polynomials (e.g., Mason and Handscomb 2002).

Based on the technique described above we present a novel lower bound on the convergence rate of p -SCLI optimization algorithms. More formally, we prove that any p -SCLI optimization algorithm over \mathbb{R}^d , whose iterations can be executed efficiently, requires

$$\#\text{Oracle Calls} \geq \tilde{\Omega} \left(\sqrt[p]{\kappa} \ln(1/\epsilon) \right) \quad (5)$$

in order to obtain an ϵ -optimal solution, *regardless of the dimension of the problem*. This result partially complements the lower bound presented earlier in Inequality (3). More specifically, for $p = 1$, we show that the runtime of algorithms whose update rules do not depend on previous points (e.g. Gradient Descent) and can be computed efficiently scales linearly with the condition number. For $p = 2$, we get the optimal result for smooth and strongly convex functions. For $p > 2$, this lower bound is clearly weaker than the lower bound shown in (3) at the first d iterations. However, we show that it can be indeed attained by p -SCLI schemes, some of which can be executed efficiently for certain classes of quadratic functions. Finally, we believe that a more refined analysis of problem (4) would show that this technique is powerful enough to meet the classical lower bound $\sqrt[p]{\kappa}$ for any p , in the worst-case over all quadratic problems.

4. In fact, we will use a polynomial matrix analogous of characteristic polynomials which will turn out to be more useful for our purposes.

The last part of this work concerns a cornerstone in the field of mathematical optimization, i.e., Nesterov's well-known Accelerated Gradient Descent method (AGD). Prior to the work of Nemirovsky and Yudin, it was known that full Gradient Descent (FGD) obtains an ϵ -optimal solution by issuing no more than

$$\mathcal{O}(\kappa \ln(1/\epsilon))$$

first-order queries. The gap between this upper bound and the lower bound shown in (3) has intrigued many researchers in the field. Eventually, it was this line of inquiry that led to the discovery of AGD by Nesterov (see Nesterov 1983), a slight modification of the standard GD algorithm, whose iteration complexity is

$$\mathcal{O}(\sqrt{\kappa} \ln(1/\epsilon)).$$

Unfortunately, AGD lacks the strong geometrical intuition which accompanies many optimization algorithms, such as FGD and the Heavy Ball method. Primarily based on sophisticated algebraic manipulations, its proof strives for a more intuitive derivation (e.g. Beck and Teboulle 2009; Baes 2009; Tseng 2008; Sutskever et al. 2013; Allen-Zhu and Orecchia 2014). This downside has rendered the generalization of AGD to different optimization scenarios, such as constrained optimization problems, a highly non-trivial task which up to the present time does not admit a complete satisfactory solution. Surprisingly enough, by designing optimization algorithms whose characteristic polynomials are optimal with respect to a constrained version of (4), we have uncovered a novel simple derivation of AGD. This reformulation as an optimal solution for a constrained optimization problem over polynomials, shows that AGD and the Heavy Ball are essentially two sides of the same coin.

To summarize, our main contributions, in order of appearance, are the following:

- We define a class of algorithms (p -SCLI) in terms of linear operations on the last p iterations, and show that they subsume some of the most interesting algorithms used in practice.
- We prove that any p -SCLI optimization algorithm must use at least
$$\tilde{\Omega}(\sqrt{\kappa} \ln(1/\epsilon))$$
 iterations in order to obtain an ϵ -optimal solution. As mentioned earlier, unlike existing lower bounds, our bound holds for every fixed dimensionality.
- We show that there exist matching p -SCLI optimization algorithms which attain the convergence rates stated above for all p . Alas, for $p \geq 3$, an expensive pre-calculation task renders these algorithms inefficient.
- As a result, we focus on a restricted subclass of p -SCLI optimization algorithms which can be executed efficiently. This yields a novel systematic derivation of Full Gradient Descent, Accelerated Gradient Descent, The Heavy-Ball method (and potentially other efficient optimization algorithms), each of which corresponds to an optimal solution of optimization problems on the moduli of polynomials' roots.

- We present new schemes which offer better utilization of second-order information by exploiting breaches in existing lower bounds. This leads to a new optimization algorithm which obtains a rate of $\sqrt[3]{\kappa} \ln(1/\epsilon)$ in the presence of large enough spectral gaps.

1.1 Notation

We denote scalars with lower case letters and vectors with bold face letters. We use \mathbb{R}^{++} to denote the set of all positive real numbers. All functions in this paper are defined over Euclidean spaces equipped with the standard Euclidean norm and all matrix-norms are assumed to denote the spectral norm.

We denote a block-diagonal matrix whose blocks are A_1, \dots, A_k by the conventional direct sum notation, i.e., $\oplus_{i=1}^k A_k$. We devote a special operator symbol for scalar matrices $\text{Diag}(a_1, \dots, a_d) = \oplus_{i=1}^d a_i$. The spectrum of a square matrix A and its spectral radius, the maximum magnitude over its eigenvalues, are denoted by $\sigma(A)$ and $\rho(A)$, respectively. Recall that the eigenvalues of a square matrix $A \in \mathbb{R}^{d \times d}$ are exactly the roots of the characteristic polynomial which is defined as follows

$$\chi_A(\lambda) = \det(A - \lambda I_d),$$

where I_d denotes the identity matrix. Since polynomials in this paper have their origins as characteristic polynomials of some square matrices, by a slight abuse of notation, we will denote the roots of a polynomial $q(z)$ and its root radius, the maximum modulus over its roots, by $\sigma(q(z))$ and $\rho(q(z))$, respectively, as well.

The following notation for quadratic functions and matrices will be of frequent use,

$$\begin{aligned} \mathcal{S}^d(\Sigma) &\triangleq \left\{ A \in \mathbb{R}^{d \times d} \mid A \text{ is symmetric and } \sigma(A) \subseteq \Sigma \right\}, \\ \mathcal{Q}^d(\Sigma) &\triangleq \left\{ f_{A,\mathbf{b}}(\mathbf{x}) \mid A \in \mathcal{S}^d(\Sigma), \mathbf{b} \in \mathbb{R}^d \right\}, \end{aligned}$$

where Σ denotes a non-empty set of positive reals, and where $f_{A,\mathbf{b}}(\mathbf{x})$ denotes the following quadratic function

$$f_{A,\mathbf{b}}(\mathbf{x}) \triangleq \frac{1}{2} \mathbf{x}^\top A \mathbf{x} + \mathbf{b}^\top \mathbf{x}, \quad A \in \mathcal{S}^d(\Sigma).$$

2. Framework

In the sequel we establish our framework for analyzing optimization algorithms for minimizing smooth and strongly convex functions. First, to motivate this technique, we show that the analysis of SDCA presented in Shalev-Shwartz and Zhang (2013) is tight by using a similar method. Next, we lay the foundations of the framework by generalizing and formalizing various aspects of the SDCA case. We then examine some popular optimization algorithms through this formulation. Apart from setting the boundaries for this work, this inspection gives rise to, otherwise subtle, distinctions between different optimization algorithms. Lastly, we discuss the computational complexity of p -SCLIs, as well as their convergence properties.

2.1 Case Study - Stochastic Dual Coordinate Ascent

We consider the optimization algorithm Stochastic Dual Coordinates Ascent (SDCA⁵) for solving Regularized Loss Minimization (RLM) problems (6), which are of great significance for the field of Machine Learning. It is shown that applying SDCA on quadratic loss functions allows one to reformulate it as a recursive application of linear transformations. The relative simplicity of such processes is then exploited to derive a lower bound on the convergence rate.

A smooth-RLM problem is an optimization task of the following form:

$$\min_{\mathbf{w} \in \mathbb{R}^n} P(\mathbf{w}) \triangleq \frac{1}{n} \sum_{i=1}^n \phi_i(\mathbf{w}^\top \mathbf{x}_i) + \frac{\lambda}{2} \|\mathbf{w}\|^2, \quad (6)$$

where ϕ_i are $1/\gamma$ -smooth and convex, $\mathbf{x}_1, \dots, \mathbf{x}_n$ are vectors in \mathbb{R}^d and λ is a positive constant. For ease of presentation, we further assume that ϕ_i are non-negative, $\phi_i(0) \leq 1$ and $\|\mathbf{x}_i\| \leq 1$ for all i .

The optimization algorithm SDCA works by minimizing an equivalent optimization problem

$$\min_{\alpha \in \mathbb{R}^n} D(\alpha) \triangleq \frac{1}{n} \sum_{i=1}^n \phi_i^*(\alpha_i) + \frac{1}{2\lambda n^2} \left\| \sum_{i=1}^n \alpha_i \mathbf{x}_i \right\|^2,$$

where ϕ_i^* denotes the Fenchel conjugate of ϕ_i , by repeatedly picking $z \sim \mathcal{U}([n])$ uniformly and minimizing $D(\alpha)$ over the z 'th coordinate. The latter optimization problem is referred to as the *dual problem*, while the problem presented in (6) is called the *primal problem*. As shown in Shalev-Shwartz and Zhang (2013), it is possible to convert a high quality solution of the dual problem into a high quality solution of the primal problem. This allows one to bound from above the number of iterations required for obtaining a prescribed level of accuracy $\epsilon > 0$ by

$$\tilde{O}\left(\left(n + \frac{1}{\lambda\gamma}\right) \ln(1/\epsilon)\right).$$

We now show that this analysis is indeed tight. First, let us define the following 2-smooth functions:

$$\phi_i(y) = y^2, \quad i = 1, \dots, n$$

and let us define $\mathbf{x}_1 = \mathbf{x}_2 = \dots = \mathbf{x}_n = \frac{1}{\sqrt{n}} \mathbb{1}$. This yields

$$D(\alpha) = \frac{1}{2} \alpha^\top \left(\frac{1}{2n} I + \frac{1}{\lambda n^2} \mathbb{1} \mathbb{1}^\top \right) \alpha. \quad (7)$$

⁵ For a detailed analysis of SDCA, please refer to Shalev-Shwartz and Zhang 2013.

Clearly, the unique minimizer of $D(\alpha)$ is $\alpha^* \triangleq 0$. Now, given $i \in [n]$ and $\alpha \in \mathbb{R}^n$, it is easy to verify that

$$\operatorname{argmin}_{\alpha' \in \mathbb{R}^n} D(\alpha_1, \dots, \alpha_{i-1}, \alpha', \alpha_{i+1}, \dots, \alpha_n) = \frac{-2}{2 + \lambda n} \sum_{j \neq i} \alpha_j. \quad (8)$$

Thus, the next test point α^+ , generated by taking a step along the i 'th coordinate, is a linear transformation of the previous point, i.e.,

$$\alpha^+ = (I - \mathbf{e}_i \mathbf{u}_i^\top) \alpha, \quad (9)$$

where

$$\mathbf{u}_i^\top \triangleq \left(\frac{2}{2 + \lambda n}, \dots, \frac{2}{2 + \lambda n}, \underbrace{\frac{1}{2 + \lambda n}}_{i's \text{ entry}}, \frac{2}{2 + \lambda n}, \dots, \frac{2}{2 + \lambda n} \right).$$

Let α^k , $k = 1, \dots, K$ denote the k 'th test point. The sequence of points $(\alpha^k)_{k=1}^K$ is randomly generated by minimizing $D(\alpha)$ over the z_k 'th coordinate at the k 'th iteration, where $z_1, z_2, \dots, z_K \sim \mathcal{U}([n])$ is a sequence of K uniform distributed i.i.d random variables. Applying (9) over and over again starting from some initialization point α^0 we obtain

$$\alpha^k = (I - \mathbf{e}_{z_K} \mathbf{u}_{z_K}^\top) (I - \mathbf{e}_{z_{K-1}} \mathbf{u}_{z_{K-1}}^\top) \dots (I - \mathbf{e}_{z_1} \mathbf{u}_{z_1}^\top) \alpha^0.$$

To compute $\mathbb{E}[\alpha^k]$ note that by the i.i.d hypothesis and by the linearity of the expectation operator,

$$\begin{aligned} \mathbb{E}[\alpha^k] &= \mathbb{E} \left[(I - \mathbf{e}_{z_K} \mathbf{u}_{z_K}^\top) (I - \mathbf{e}_{z_{K-1}} \mathbf{u}_{z_{K-1}}^\top) \dots (I - \mathbf{e}_{z_1} \mathbf{u}_{z_1}^\top) \alpha^0 \right] \\ &= \mathbb{E} \left[(I - \mathbf{e}_{z_K} \mathbf{u}_{z_K}^\top) \right] \mathbb{E} \left[(I - \mathbf{e}_{z_{K-1}} \mathbf{u}_{z_{K-1}}^\top) \right] \dots \mathbb{E} \left[(I - \mathbf{e}_{z_1} \mathbf{u}_{z_1}^\top) \right] \alpha^0 \\ &= \mathbb{E} \left[(I - \mathbf{e}_z \mathbf{u}_z^\top) \right]^K \alpha^0. \end{aligned} \quad (10)$$

The convergence rate of the latter is governed by the spectral radius of

$$E \triangleq \mathbb{E} \left[I - \mathbf{e}_z \mathbf{u}_z^\top \right].$$

A straightforward calculation shows that the eigenvalues of E , ordered by magnitude, are

$$\underbrace{1 - \frac{1}{2/\lambda + n}, \dots, 1 - \frac{1}{2/\lambda + n}}_{n-1 \text{ times}}, 1 - \frac{2 + \lambda}{2 + \lambda n}. \quad (11)$$

By choosing α^0 to be the following normalized eigenvector which corresponds to the largest eigenvalue

$$\alpha^0 = \left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 0, \dots, 0 \right),$$

and plugging it into Equation (10), we can now bound from below the distance of $\mathbb{E}[\boldsymbol{\alpha}^k]$ to the optimal point $\boldsymbol{\alpha}^* = 0$,

$$\begin{aligned} \|\mathbb{E}[\boldsymbol{\alpha}^k] - \boldsymbol{\alpha}^*\| &= \left\| \mathbb{E} \left[\left(I - \mathbf{e}_z \mathbf{u}_z^\top \right)^K \boldsymbol{\alpha}^0 \right] \right\| \\ &= \left(1 - \frac{1}{2/\lambda + n} \right)^K \|\boldsymbol{\alpha}^0\| \\ &= \left(1 - \frac{2}{(4/\lambda + 2n - 1) + 1} \right)^K \\ &\geq \left(\exp \left(\frac{-1}{2/\lambda + n - 1} \right) \right)^K, \end{aligned} \quad (12)$$

where the last inequality is due to the following inequality,

$$1 - \frac{2}{x+1} \geq \exp \left(\frac{-2}{x-1} \right), \quad \forall x \geq 1.$$

We see that the minimal number of iterations required for obtaining a solution whose distance from the $\boldsymbol{\alpha}^*$ is less than $\epsilon > 0$ must be greater than

$$(2/\lambda + n - 1) \ln(1/\epsilon),$$

thus showing that, up to logarithmic factors, the analysis of the convergence rate of SDCA is tight.

2.2 Definitions

In the sequel we introduce the framework of p -SCLI optimization algorithms which generalizes the analysis shown in the preceding section.

We denote the set of $d \times d$ symmetric matrices whose spectrum lies in $\Sigma \subseteq \mathbb{R}^{++}$ by $\mathcal{S}^d(\Sigma)$ and denote the following set of quadratic functions

$$f_{A,\mathbf{b}}(\mathbf{x}) \triangleq \frac{1}{2} \mathbf{x}^\top A \mathbf{x} + \mathbf{b}^\top \mathbf{x}, \quad A \in \mathcal{S}^d(\Sigma),$$

by $\mathcal{Q}^d(\Sigma)$. Note that since twice continuous differentiable functions $f(\mathbf{x})$ are L -smooth and μ -strongly convex if and only if

$$\sigma(\nabla^2 f(\mathbf{x})) \subseteq [\mu, L] \subseteq \mathbb{R}^{++}, \quad \mathbf{x} \in \mathbb{R}^d,$$

we have that $\mathcal{Q}^d(\mu, L)$ comprises L -smooth μ -strongly convex quadratic functions. Thus, any optimization algorithm designed for minimizing smooth and strongly convex functions can be used to minimize functions in $\mathcal{Q}^d(\mu, L)$. The key observation here is that since the gradient of $f_{A,\mathbf{b}}(\mathbf{x})$ is linear in \mathbf{x} , when applied to quadratic functions, the update rules of many optimization algorithms also become linear in \mathbf{x} . This formalizes as follows.

Definition 1 (p -SCLI optimization algorithms) An optimization algorithm \mathcal{A} is called a p -stationary canonical linear iterative (abbr. p -SCLI) optimization algorithm over \mathbb{R}^d if there exist $p+1$ mappings $C_0(X), C_1(X), \dots, C_{p-1}(X), N(X)$ from $\mathbb{R}^{d \times d}$ to $\mathbb{R}^{d \times d}$ -valued random variables, such that for any $f_{A,\mathbf{b}}(\mathbf{x}) \in \mathcal{Q}^d(\Sigma)$ the corresponding initialization and update rules take the following form:

$$\mathbf{x}^0, \mathbf{x}^1, \dots, \mathbf{x}^{p-1} \in \mathbb{R}^d \quad (13)$$

$$\mathbf{x}^k = \sum_{j=0}^{p-1} C_j(A) \mathbf{x}^{k-p+j} + N(A) \mathbf{b}, \quad k = p, p+1, \dots \quad (14)$$

We further assume that in each iteration $C_j(A)$ and $N(A)$ are drawn independently of previous realizations⁶, and that $\mathbb{E}C_i(A)$ are finite and simultaneously triangularizable⁷.

Let us introduce a few more definitions and terminology which will be used throughout this paper. The number of previous points p by which new points are generated is called the lifting factor. The matrix-valued random variables $C_0(X), C_1(X), \dots, C_{p-1}(X)$ and $N(X)$ are called coefficient matrices and inversion matrix, respectively. The term inversion matrix refers to the mapping $N(X)$, as well as to a concrete evaluation of it. It will be clear from the context which interpretation is being used. The same holds for coefficient matrices.

As demonstrated by the following definition, coefficients matrices of p -SCLIs can be equivalently described in terms of polynomial matrices⁸. This correspondence will soon play a pivotal role in the analysis of p -SCLIs.

Definition 2 The characteristic polynomial of a given p -SCLI optimization algorithm \mathcal{A} is defined by

$$\mathcal{L}_{\mathcal{A}}(\lambda, X) \triangleq I_d \lambda^p - \sum_{j=0}^{p-1} \mathbb{E}C_j(X) \lambda^j, \quad (15)$$

where $C_j(X)$ denote the coefficient matrices. Moreover, given $X \in \mathbb{R}^{d \times d}$ we define the root radius of $\mathcal{L}_{\mathcal{A}}(\lambda, X)$ by

$$\rho_{\mathcal{A}}(\mathcal{L}_{\mathcal{A}}(\lambda, X)) = \rho(\det \mathcal{L}_{\mathcal{A}}(\lambda, X)) = \max \{ |\lambda| \mid \det \mathcal{L}_{\mathcal{A}}(\lambda, X) = 0 \}.$$

For the sake of brevity, we sometimes specify a given p -SCLI optimization algorithm \mathcal{A} using an ordered pair of a characteristic polynomial and an inversion matrix as follows

$$\mathcal{A} \triangleq (\mathcal{L}_{\mathcal{A}}(\lambda, X), N(X)).$$

Furthermore, we may omit the subscript \mathcal{A} , when it is clear from the context.

6. We shall refer to this assumption as *stationarity*.

7. Intuitively, having this technical requirement is somewhat similar to assuming that the coefficients matrices commute (see Draxin et al. 1951 for a precise statement), and as such does not seem to restrict the scope of this work. Indeed, it is common to have $\mathbb{E}C_i(A)$ as polynomials in A or as diagonal matrices, in which case the assumption holds true.

8. For a detailed cover of polynomial matrices see Golberg et al. (2009).

Lastly, note that nowhere in the definition of p -SCLIs did we assume that the optimization process converges to the minimizer of the function under consideration - an assumption which we refer to as *consistency*.

Definition 3 (Consistency of p -SCLI optimization algorithms) A p -SCLI optimization algorithm \mathcal{A} is said to be consistent with respect to a given $A \in \mathbf{S}^d(\Sigma)$ if for any $\mathbf{b} \in \mathbb{R}^d$, \mathcal{A} converges to the minimizer of $f_{A,\mathbf{b}}(\mathbf{x})$, regardless of the initialization point. That is, for $(\mathbf{x}^k)_{k=1}^\infty$ as defined in (13,14) we have that

$$\mathbf{x}^k \rightarrow -A^{-1}\mathbf{b},$$

for any $\mathbf{b} \in \mathbb{R}^d$. Furthermore, if \mathcal{A} is consistent with respect to all $A \in \mathbf{S}^d(\Sigma)$, then we say that \mathcal{A} is consistent with respect to $\mathcal{Q}^d(\Sigma)$.

2.3 Specifications for Some Popular Optimization Algorithms

Having defined the framework of p -SCLI optimization algorithms, a natural question now arises: how broad is the scope of this framework and what does characterize optimization algorithms which it applies to? Loosely speaking, any optimization algorithm whose update rules depend linearly on the first and the second order derivatives of the function under consideration is eligible for this framework. Instead of providing a precise characterization for such algorithms, we apply various popular optimization algorithms on a general quadratic function $f_{A,\mathbf{b}}(\mathbf{x}) \in \mathcal{Q}^d(\mu, L)$ and then express them as p -SCLI optimization algorithms.

Full Gradient Descent (FGD) is a 1-SCLI optimization algorithm with

$$\begin{aligned} \mathbf{x}^0 &\in \mathbb{R}^d, \\ \mathbf{x}^{k+1} &= \mathbf{x}^k - \beta \nabla f(\mathbf{x}^k) = \mathbf{x}^k - \beta(A\mathbf{x}^k + \mathbf{b}) = (I - \beta A)\mathbf{x}^k - \beta\mathbf{b}, \\ \beta &= \frac{2}{\mu + L}. \end{aligned}$$

See Nesterov (2004) for more details.

Newton method is a 0-SCLI optimization algorithm with

$$\begin{aligned} \mathbf{x}^0 &\in \mathbb{R}^d, \\ \mathbf{x}^{k+1} &= \mathbf{x}^k - (\nabla^2 f(\mathbf{x}^k))^{-1} \nabla f(\mathbf{x}^k) = \mathbf{x}^k - A^{-1}(A\mathbf{x}^k + \mathbf{b}) \\ &= (I - A^{-1}A)\mathbf{x}^k - A^{-1}\mathbf{b} = -A^{-1}\mathbf{b}. \end{aligned}$$

Note that Newton method can be also formulated as a degenerate p -SCLI for some $p \in \mathbb{N}$, whose coefficients matrices vanish. See Nesterov (2004) for more details.

The Heavy Ball Method is a 2-SCLI optimization algorithm with

$$\begin{aligned} \mathbf{x}^{k+1} &= \mathbf{x}^k - \alpha \nabla f(\mathbf{x}^k) + \beta(\mathbf{x}^k - \mathbf{x}^{k-1}) \\ &= \mathbf{x}^k - \alpha(A\mathbf{x}^k + \mathbf{b}) + \beta(\mathbf{x}^k - \mathbf{x}^{k-1}) \\ &= ((1 + \beta)I - \alpha A)\mathbf{x}^k - \beta I\mathbf{x}^{k-1} - \alpha\mathbf{b}, \\ \alpha &= \frac{4}{(\sqrt{L} + \sqrt{\mu})^2}, \quad \beta = \left(\frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}} \right)^2. \end{aligned}$$

See Polyak (1987) for more details.

Accelerated Gradient Descent (AGD) is a 2-SCLI optimization algorithm with

$$\begin{aligned} \mathbf{x}^0 &= \mathbf{y}^0 \in \mathbb{R}^d, \\ \mathbf{y}^{k+1} &= \mathbf{x}^k - \frac{1}{L} \nabla f(\mathbf{x}^k), \\ \mathbf{x}^{k+1} &= (1 + \alpha)\mathbf{y}^{k+1} - \alpha\mathbf{y}^k, \\ \alpha &= \frac{\sqrt{L} - \sqrt{\mu}}{\sqrt{L} + \sqrt{\mu}}, \end{aligned}$$

which can be rewritten as follows:

$$\begin{aligned} \mathbf{x}^0 &\in \mathbb{R}^d, \\ \mathbf{x}^{k+1} &= (1 + \alpha) \left(\mathbf{x}^k - \frac{1}{L} \nabla f(\mathbf{x}^k) \right) - \alpha \left(\mathbf{x}^{k-1} - \frac{1}{L} \nabla f(\mathbf{x}^{k-1}) \right) \\ &= (1 + \alpha) \left(\mathbf{x}^k - \frac{1}{L}(A\mathbf{x}^k + \mathbf{b}) \right) - \alpha \left(\mathbf{x}^{k-1} - \frac{1}{L}(A\mathbf{x}^{k-1} + \mathbf{b}) \right) \\ &= (1 + \alpha) \left(I - \frac{1}{L}A \right) \mathbf{x}^k - \alpha \left(I - \frac{1}{L}A \right) \mathbf{x}^{k-1} - \frac{1}{L}\mathbf{b}. \end{aligned}$$

Note that here we employ a stationary variant of AGD. See Nesterov (2004) for more details.

Stochastic Coordinate Descent (SCD) is a 1-SCLI optimization algorithm. This is a generalization of the example shown in Section 2.1. SCD acts by repeatedly minimizing a uniformly randomly drawn coordinate in each iteration. That is,

$$\begin{aligned} \mathbf{x}^0 &\in \mathbb{R}^d, \\ \text{Pick } i &\sim \mathcal{U}([d]) \text{ and set } \mathbf{x}^{k+1} = \left(I - \frac{1}{A_{i,i}} \mathbf{e}_i \mathbf{a}_{i,*}^\top \right) \mathbf{x}^k - \frac{b_i}{A_{i,i}} \mathbf{e}_i, \end{aligned}$$

where $\mathbf{a}_{i,*}^\top$ denotes the i 'th row of A and $\mathbf{b} \triangleq (b_1, b_2, \dots, b_d)$. Note that the expected update rule of this method is equivalent to the well-known Jacobi's iterative method.

We now describe some popular optimization algorithms which do not fit this framework, mainly because the stationarity requirement fails to hold. The extension of this framework to cyclic and piecewise stationary optimization algorithms is left to future work.

Conjugate Gradient Descent (CGD) can be expressed as a non-stationary iterative method

$$\mathbf{x}^{k+1} = ((1 + \beta_k)I - \alpha_k A) \mathbf{x}^k - \beta_k I \mathbf{x}^{k-1} - \alpha_k \mathbf{b},$$

where α_k and β_k are computed at each iteration based on $\mathbf{x}^k, \mathbf{x}^{k-1}, A$ and \mathbf{b} . Note the similarity of CGD and the heavy ball method. See Polyak (1987); Nemirovski (2005) for more details. In the context of this framework, CGD forms the ‘most non-stationary’ kind of method in that its coefficients α_k, β_k are highly dependent on time and the function at hand.

Stochastic Gradient Descent (SGD) A straightforward extension of the deterministic FGD. Specifically, let $(\Omega, \mathcal{F}, \mathcal{P})$ be a probability space and let $G(\mathbf{x}, \omega) : \mathbb{R}^d \times \Omega \rightarrow \mathbb{R}^d$ be an unbiased estimator of $\nabla f(\mathbf{x})$ for any \mathbf{x} . That is,

$$\mathbb{E}[G(\mathbf{x}, \omega)] = \nabla f(\mathbf{x}) = A\mathbf{x} + \mathbf{b}, \quad \mathbf{x} \in \mathbb{R}^d.$$

Equivalently, define $\mathbf{e}(\mathbf{x}, \omega) = G(\mathbf{x}, \omega) - (A\mathbf{x} + \mathbf{b})$ and assume $\mathbb{E}[\mathbf{e}(\mathbf{x}, \omega)] = 0, \mathbf{x} \in \mathbb{R}^d$. SGD may be defined using a suitable sequence of step sizes $(\gamma_i)_{i=1}^\infty$ as follows:

$$\begin{aligned} \text{Generate } \omega_k \text{ randomly and set } \mathbf{x}^{k+1} &= \mathbf{x}^k - \gamma_k G(\mathbf{x}^k, \omega_k) \\ &= (I - \gamma_k A) \mathbf{x}^k - \gamma_k \mathbf{b} - \gamma_k \mathbf{e}(\mathbf{x}, \omega). \end{aligned}$$

Clearly, some types of noise may not form a p -SCLI optimization algorithm. However, for some instances, e.g., quadratic learning problems, we have

$$\mathbf{e}(\mathbf{x}, \omega) = A_\omega \mathbf{x} + \mathbf{b}_\omega,$$

such that

$$\mathbb{E}[A_\omega] = 0, \quad \mathbb{E}[\mathbf{b}_\omega] = 0.$$

If, in addition, the step size is fixed then we get a 1-SCLI optimization algorithm. See Kushner and Yin (2003); Spall (2005); Nemirovski (2005) for more details.

2.4 Computational Complexity

The stationarity property of general p -SCLIs optimization algorithms implies that the computational cost of minimizing a given quadratic function $f_{A,\mathbf{b}}(\mathbf{x})$, assuming $\Theta(1)$ cost for all arithmetic operations, is

$$\# \text{ Iterations} \times \begin{cases} \text{Generating coefficient and inversion matrices randomly} \\ + \\ \text{Executing update rule (14) based on the previous } p \text{ points} \end{cases}$$

The computational cost of the execution of update rule (14) scales quadratically with d , the dimension of the problem, and linearly with p , the lifting factor. Thus, the running time of p -SCLIs is mainly affected by the iterations number and the computational cost of

randomly generating coefficient and inversion matrices each time. Notice that for deterministic p -SCLIs one can save running time by computing the coefficient and inversion matrices once, prior to the execution of the algorithm. Not surprisingly, but interesting nonetheless, there is a law of conservation which governs the total amount of computational cost invested in both factors: the more demanding is the task of randomly generating coefficient and inversion matrices, the less is the total number of iterations required for obtaining a given level of accuracy, and vice versa. Before we can make this statement more rigorous, we need to present a few more facts about p -SCLIs. For the time being, let us focus on the *iteration complexity*, i.e., the total number iterations, which forms our analogy for black box complexity.

The *iteration complexity* of a p -SCLI optimization algorithm \mathcal{A} with respect to an accuracy level ϵ , initialization points \mathcal{X}^0 and a quadratic function $f_{A,\mathbf{b}}(\mathbf{x})$, symbolized by

$$\mathcal{IC}_{\mathcal{A}}(\epsilon, f_{A,\mathbf{b}}(\mathbf{x}), \mathcal{X}^0),$$

is defined to be the minimal number of iterations K such that

$$\|\mathbb{E}[\mathbf{x}^k - \mathbf{x}^*]\| < \epsilon, \quad \forall k \geq K,$$

where $\mathbf{x}^* = -A^{-1}\mathbf{b}$ is the minimizer of $f_{A,\mathbf{b}}(\mathbf{x})$, assuming \mathcal{A} is initialized at \mathcal{X}^0 . We would like to point out that although iteration complexity is usually measured through

$$\mathbb{E}\|\mathbf{x}^k - \mathbf{x}^*\|,$$

here we employ a different definition. We will discuss this issue shortly.

In addition to showing that the iteration complexity of p -SCLI algorithms scales logarithmically with $1/\epsilon$, the following theorem provides a characterization for the iteration complexity in terms of the root radius of the characteristic polynomial.

Theorem 4 *Let \mathcal{A} be a p -SCLI optimization algorithm over \mathbb{R}^d and let $f_{A,\mathbf{b}}(\mathbf{x}) \in \mathcal{Q}^d(\Sigma)$, $(\Sigma \subseteq \mathbb{R}^{++})$ be a quadratic function. Then, there exists $\mathcal{X}^0 \in (\mathbb{R}^d)^p$ such that*

$$\mathcal{IC}_{\mathcal{A}}(\epsilon, f_{A,\mathbf{b}}(\mathbf{x}), \mathcal{X}^0) = \tilde{\Omega}\left(\frac{\rho}{1-\rho} \ln(1/\epsilon)\right),$$

and for all $\mathcal{X}^0 \in (\mathbb{R}^d)^p$, it holds that

$$\mathcal{IC}_{\mathcal{A}}(\epsilon, f_{A,\mathbf{b}}(\mathbf{x}), \mathcal{X}^0) = \tilde{\mathcal{O}}\left(\frac{1}{1-\rho} \ln(1/\epsilon)\right),$$

where ρ denotes the root radius of the characteristic polynomial evaluated at $X = A$.

The full proof for this theorem is somewhat long and thus provided in Section C.1. Nevertheless, the intuition behind it is very simple and may be sketched as follows:

- First, we express update rule (14) as a single step rule by introducing new variables in some possibly higher-dimensional Euclidean space $(\mathbb{R}^d)^p$,

$$\mathbf{z}^0 = (\mathbf{x}^0, \mathbf{x}^1, \dots, \mathbf{x}^{p-1})^\top \in \mathbb{R}^{pd}, \quad \mathbf{z}^k = M(X)\mathbf{z}^{k-1} + UN(X)\mathbf{b}, \quad k = 1, 2, \dots$$

Recursively applying this rule and taking expectation w.r.t. the coefficient matrices and the inversion matrix yields

$$\mathbb{E}[\mathbf{z}^k - \mathbf{z}^*] = \mathbb{E}[M]^k (\mathbf{z}^0 - \mathbf{z}^*).$$

- Then, to derive the lower bound, we use the Jordan form of $\mathbb{E}[M]$ to show that there exists some non-zero vector $\mathbf{r} \in (\mathbb{R}^d)^p$ such that if $\langle \mathbf{z}^0 - \mathbf{z}^*, \mathbf{r} \rangle \neq 0$, then $\|\mathbb{E}[M]^k (\mathbf{z}^0 - \mathbf{z}^*)\|$ is asymptotically bounded from below by some geometric sequence. The upper bound follows similarly.
- Finally, we express the bound on the convergence rate of $\langle \mathbf{z}^k \rangle$ in terms of the original space.

Carefully inspecting the proof idea shown above reveals that the lower bound remains valid even in cases where the initialization points are drawn randomly. The only condition for this to hold is that the underlying distribution is reasonable, in the sense that it is absolutely continuous w.r.t. the Lebesgue measure, which implies that $\Pr[\langle \mathbf{z}^0 - \mathbf{z}^*, \mathbf{r} \rangle \neq 0] = 1$.

We remark that the constants in the asymptotic behavior above may depend on the quadratic function under consideration, and that the logarithmic terms depend on the distance of the initialization points from the minimizer, as well as the lifting factor and the spectrum of the Hessian. For the sake of clarity, we omit the dependency on these quantities.

There are two, rather subtle, issues regarding the definition of iteration complexity which we would like to address. First, observe that in many cases a given point $\bar{\mathbf{x}} \in \mathbb{R}^d$ is said to be ϵ -optimal w.r.t. some real function $f: \mathbb{R}^d \rightarrow \mathbb{R}$ if

$$f(\bar{\mathbf{x}}) < \min_{\mathbf{x} \in \mathbb{R}^d} f(\mathbf{x}) + \epsilon.$$

However, here we employ a different measure for optimality. Fortunately, in our case either can be used without essentially affecting the iteration complexity. That is, although in general the gap between these two definitions can be made arbitrarily large, for L -smooth μ -strongly convex functions we have

$$\frac{\mu}{2} \|\mathbf{x} - \mathbf{x}^*\|^2 \leq f(\mathbf{x}) - f(\mathbf{x}^*) \leq \frac{L}{2} \|\mathbf{x} - \mathbf{x}^*\|^2.$$

Combining these two inequalities with the fact that the iteration complexity of p -SCLIs depends logarithmically on $1/\epsilon$ implies that in this very setting these two distances are interchangeable, up to logarithmic factors.

Secondly, here we measure the sub-optimality of the k 'th iteration by $\|\mathbb{E}[\mathbf{x}^k - \mathbf{x}^*]\|$, whereas in many other stochastic settings it is common to derive upper and lower bounds on $\mathbb{E}[\|\mathbf{x}^k - \mathbf{x}^*\|]$. That being the case, by

$$\mathbb{E}[\|\mathbf{x}^k - \mathbf{x}^*\|^2] = \mathbb{E}[\|\mathbf{x}^k - \mathbb{E}\mathbf{x}^k\|^2] + \|\mathbb{E}[\mathbf{x}^k - \mathbf{x}^*]\|^2,$$

we see that if the variance of the k 'th point is of the same order of magnitude as the norm of the expected distance from the optimal point, then both measures are equivalent. Consequently, our upper bounds imply upper bounds on $\mathbb{E}[\|\mathbf{x}^k - \mathbf{x}^*\|^2]$ for deterministic algorithms (where the variance term is zero), and our lower bounds imply lower bounds on $\mathbb{E}[\|\mathbf{x}^k - \mathbf{x}^*\|^2]$, for both deterministic and stochastic algorithms (since the variance is non-negative). We defer a more adequate treatment for this matter to future work.

3. Deriving Bounds for p -SCLI Algorithms

The goal of the following section is to show how the framework of p -SCLI optimization algorithms can be used to derive lower and upper bounds. Our presentation follows from the simplest setting to the most general one. First, we present a useful characterization of consistency (see Definition 3) of p -SCLIs using the characteristic polynomial. Next, we demonstrate the importance of consistency through a simplified one dimensional case. This line of argument is then generalized to any finite dimensional space and is used to explain the role of the inversion matrix. Finally, we conclude this section by providing a schematic description of this technique for the most general case which is used both in Section (4) to establish lower bounds on the convergence rate of p -SCLIs with diagonal inversion matrices, and in Section (5) to derive efficient p -SCLIs.

3.1 Consistency

Closely inspecting various specifications for p -SCLI optimization algorithms (see Section (2.3)) reveals that the coefficient matrices always sum up to $I + \mathbb{E}N(X)X$, where $N(X)$ denotes the inversion matrix. It turns out that this is not a mere coincidence, but an extremely useful characterization for consistency of p -SCLIs. To see why this condition must hold, suppose \mathcal{A} is a deterministic p -SCLI algorithm over \mathbb{R}^d whose coefficient matrices and inversion matrix are $C_0(X), \dots, C_{p-1}(X)$ and $N(X)$, respectively, and suppose that \mathcal{A} is consistent w.r.t. some $A \in S^d(\Sigma)$. Recall that every $p + 1$ consecutive points generated by \mathcal{A} are related by (14) as follows

$$\mathbf{x}^k = \sum_{j=0}^{p-1} C_j(A)\mathbf{x}^{k-p+j} + N(A)\mathbf{b}, \quad k = p, p+1, \dots$$

Taking limit of both sides of the equation above and noting that by consistency

$$\mathbf{x}^k \rightarrow -A^{-1}\mathbf{b}$$

for any $\mathbf{b} \in \mathbb{R}^d$, yields

$$-A^{-1}\mathbf{b} = -\sum_{j=0}^{p-1} C_j(A)A^{-1}\mathbf{b} + N(A)\mathbf{b}.$$

Thus,

$$-A^{-1} = -\sum_{j=0}^{p-1} C_j(A)A^{-1} + N(A).$$

Multiplying by A and rearranging, we obtain

$$\sum_{j=0}^{p-1} C_j(A) = I_d + N(A)A. \quad (16)$$

On the other hand, if instead of assuming consistency we assume that \mathcal{A} generates a convergent sequence of points and that Equation (16) holds, then the arguments used above show that the limit point must be $-A^{-1}\mathbf{b}$. In terms of the characteristic polynomial of p -SCLIs, this is formalized as follows.

Theorem 5 (Consistency via Characteristic Polynomials) *Suppose $\mathcal{A} \triangleq (\mathcal{L}(\lambda, X), N(X))$ is a p -SCLI optimization algorithm. Then, \mathcal{A} is consistent with respect to $A \in \mathcal{S}^d(\Sigma)$ if and only if the following two conditions hold:*

1. $\mathcal{L}(1, A) = -\mathbb{E}N(A)A$ (17)
2. $\rho_\lambda(\mathcal{L}(\lambda, A)) < 1$ (18)

The proof for the preceding theorem is provided in Section C.2. This result will be used extensively throughout the remainder of this work.

3.2 Simplified One-Dimensional Case

To illustrate the significance of consistency in the framework of p -SCLIs, consider the following simplified case. Suppose \mathcal{A} is a deterministic 2-SCLI optimization algorithm over $\mathcal{Q}^1(\mu, L)$, such that its inversion matrix $N(x)$ is some constant scalar $\nu \in \mathbb{R}$ and its coefficient matrices $c_0(x), c_1(x)$ are free to take any form. The corresponding characteristic polynomial is

$$\mathcal{L}(\lambda, x) = \lambda^2 - c_1(x)\lambda - c_0(x).$$

Now, let $f_{a,b}(x) \in \mathcal{Q}^1(\mu, L)$ be a quadratic function. By Theorem 4, we know that \mathcal{A} converges to the minimizer of $f_{a,b}(x)$ with an asymptotic geometric rate of $\rho_\lambda(\mathcal{L}(\lambda, a))$, the maximal modulus root. Thus, ideally we would like to set $c_j(x) = 0$, $j = 0, 1$. However, this might violate the consistency condition (17), according to which, one must maintain

$$\mathcal{L}(1, a) = -\nu a.$$

That being the case, how little can $\rho_\lambda(\mathcal{L}(\lambda, a))$ be over all possible choices for $c_j(a)$ which satisfy $\mathcal{L}(1, a) = -\nu a$? Formally, we seek to solve the following minimization problem

$$\rho_* = \min \{ \rho_\lambda(\mathcal{L}(\lambda, a)) \mid \mathcal{L}(\lambda, a) \text{ is a real monic quadratic polynomial in } \lambda \text{ and } \mathcal{L}(1) = -\nu a \}.$$

By consistency we also have that ρ_* must be strictly less than one. This readily implies that $-\nu a > 0$. In which case, Lemma 6 below gives

$$\rho_* \geq \rho \left((\lambda - 1 - \sqrt{-\nu a})^2 \right) = \left| \sqrt{-\nu a} - 1 \right|. \quad (19)$$

The key observation here is that ν cannot be chosen so as to be optimal for all $\mathcal{Q}^1(\mu, L)$ simultaneously. Indeed, the preceding inequality holds in particular for $a = \mu$ and $a = L$, by which we conclude that

$$\rho_* \geq \max \left\{ \left| \sqrt{-\nu\mu} - 1 \right|, \left| \sqrt{-\nu L} - 1 \right| \right\} \geq \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}, \quad (20)$$

where $\kappa \triangleq L/\mu$. Plugging in Inequality (20) into Theorem 4 implies that there exists $f_{a,b}(x) \in \mathcal{Q}^1(\mu, L)$ such that the iteration complexity of \mathcal{A} for minimizing it is

$$\tilde{\Omega} \left(\frac{\sqrt{\kappa} - 1}{2} \ln(1/\epsilon) \right).$$

To conclude, by applying this rather natural line of argument we have established a lower bound on the convergence rate of any 2-SCLI optimization algorithms for smooth and strongly convex function over \mathbb{R} , e.g., AGD and HB.

3.3 The General Case and the Role of the Inversion Matrix

We now generalize the analysis shown in the previous simplified case to any deterministic p -SCLI optimization algorithm over any finite dimensional space. This generalization relies on a useful decomposability property of the characteristic polynomial, according to which deriving a lower bound on the convergence rate of p -SCLIs over \mathbb{R}^d is essentially equivalent for deriving d lower bounds on the maximal modulus of the roots of d polynomials over \mathbb{R} .

Let $\mathcal{A} \triangleq (\mathcal{L}(\lambda, X), N(X))$ be a consistent deterministic p -SCLI optimization algorithm and let $f_{A,\mathbf{b}}(\mathbf{x}) \in \mathcal{Q}^d(\Sigma)$ be a quadratic function. By consistency (see Theorem 5) we have

$$\mathcal{L}(1, A) = -NA$$

(for brevity we omit the functional dependency on X). Since coefficient matrices are assumed to be simultaneously triangularizable, there exists an invertible matrix $Q \in \mathbb{R}^{d \times d}$ such that

$$T_j \triangleq Q^{-1}C_jQ, \quad j = 0, 1, \dots, p-1$$

are upper triangular matrices. Thus, by the definition of the characteristic polynomial (Definition 2) we have

$$\det \mathcal{L}(\lambda, X) = \det (Q^{-1}\mathcal{L}(\lambda, X)Q) = \det \left(I_d \lambda^p - \sum_{j=0}^{p-1} T_j \lambda^j \right) = \prod_{j=1}^d \ell_j(\lambda), \quad (21)$$

where

$$\ell_j(\lambda) = \lambda^p - \sum_{k=0}^{p-1} \sigma_k^j \lambda^k, \quad (22)$$

and where $\sigma_0^j, \dots, \sigma_{d-1}^j$, $j = 0, \dots, p-1$ denote the elements on the diagonal of \mathcal{T}_j , or equivalently the eigenvalues of G_j ordered according to Q_j . Hence, the root radius of the characteristic polynomial of \mathcal{A} is

$$\rho_\lambda(\mathcal{L}(\lambda, X)) = \max\{|\lambda| \mid \ell_i(\lambda) = 0 \text{ for some } i \in [d]\}. \quad (23)$$

On the other hand, by consistency condition (17) we get that for all $i \in [d]$,

$$\ell_i(1) = \sigma_i(\mathcal{L}(1)) = \sigma_i(-NA). \quad (24)$$

It remains to derive a lower bound on the maximum modulus of the roots of $\ell_i(\lambda)$, subject to constraint (24). To this end, we employ the following lemma whose proof can be found in Section C.3.

Lemma 6 *Suppose $q(z)$ is a real monic polynomial of degree p . If $q(1) < 0$, then*

$$\rho(q(z)) > 1.$$

Otherwise, if $q(1) \geq 0$, then

$$\rho(q(z)) \geq \left| \sqrt[p]{q(1)} - 1 \right|.$$

In which case, equality holds if and only if

$$q(z) = \left(z - (1 - \sqrt[p]{q(1)}) \right)^p.$$

We remark that the second part of Lemma 6 implies that subject to constraint (24), the lower bound stated above is unimprovable. This property is used in Section 5 where we aim to obtain optimal p -SCLIs by designing $\ell_j(\lambda)$ accordingly. Clearly, in the presence of additional constraints, one might be able to improve on this lower bound (see Section 4.2).

Since \mathcal{A} is assumed to be consistent, Lemma 6 implies that $\sigma(-N(A)A) \subseteq \mathbb{R}^{++}$, as well as the following lower bound on the root radius of the characteristic polynomial,

$$\rho_\lambda(\mathcal{L}(\lambda, X)) \geq \max_{i \in [d]} \left| \sqrt[p]{\sigma_i(-N(A)A)} - 1 \right|. \quad (25)$$

Noticing that the reasoning above can be readily applied to stochastic p -SCLI optimization algorithms, we arrive at the following corollary which combines Theorem 4 and Inequality (25).

Corollary 7 *Let \mathcal{A} be a consistent p -SCLI optimization algorithm with respect to some $A \in \mathcal{S}^d(\Sigma)$, let $N(X)$ denote the corresponding inversion matrix and let*

$$\rho^* = \max_{i \in [d]} \left| \sqrt[p]{\sigma_i(-\mathbb{E}N(A)A)} - 1 \right|,$$

then the iteration complexity of \mathcal{A} for any $f_{A,b}(\mathbf{x}) \in \mathcal{Q}^d(\Sigma)$ is lower bounded by

$$\tilde{\Omega} \left(\frac{\rho^*}{1 - \rho^*} \ln(1/\epsilon) \right). \quad (26)$$

Using Corollary 7, we are now able to provide a concise ‘plug-and-play’ scheme for deriving lower bounds on the iteration complexity of p -SCLI optimization algorithms. To motivate this scheme, note that the effectiveness of the lower bound stated in Corollary 7 is directly related to the magnitude of the eigenvalues of $-N(X)X$. To exemplify this, consider the inversion matrix of Newton method (see Section 2.3)

$$N(X) = -X^{-1}.$$

Since

$$\sigma(-N(X)X) = \{1\},$$

the lower bound stated above is meaningless for this case. Nevertheless, the best computational cost for computing the inverse of $d \times d$ regular matrices known today is super-quadratic in d . As a result, this method might become impractical in large scale scenarios where the dimension of the problem space is large enough. A possible solution is to employ inversion matrices whose dependence on X is simpler. On the other hand, if $N(X)$ approximates $-X^{-1}$ very badly, then the root radius of the characteristic polynomial might get too large. For instance, if $N(X) = 0$ then

$$\sigma(-N(X)X) = \{0\},$$

contradicting the consistency assumption, regardless of the choice of the coefficient matrices.

In light of the above, many optimization algorithms can be seen as strategies for balancing the computational cost of obtaining a good approximation for the inverse of X and executing large number of iterations. Put differently, various structural restrictions on the inversion matrix yield different $\sigma(-N(X)X)$, which in turn lead to a lower bound on the root radius of the corresponding characteristic polynomial. This gives rise to the following scheme:

Scheme 1	Lower bounds
Parameters:	<ul style="list-style-type: none"> • A family of quadratic functions $\mathcal{Q}^d(\Sigma)$ • An inversion matrix $N(X)$
Choose	• A lifting factor $p \in \mathbb{N}$,
Verify	$S' \subseteq \mathcal{S}^d(\Sigma)$
Bound	$\forall A \in S', \sigma(-\mathbb{E}N(A)A) \subseteq (0, 2^p)$ to ensure consistency (Theorem 5)
Lower bound:	$\max_{A \in S', i \in [d]} \left \sqrt[p]{\sigma_i(-\mathbb{E}N(A)A)} - 1 \right $ from below by some $\rho^* \in [0, 1)$
Lower bound:	$\tilde{\Omega} \left(\frac{\rho^*}{1 - \rho^*} \ln(1/\epsilon) \right)$

This scheme is implicitly used in the previous Section (3.2), where we established a lower bound on the convergence rate of 2-SCLI optimization algorithms over \mathbb{R} with constant inversion matrix and the following parameters

$$\Sigma = [\mu, L], \quad S' = \{\mu, L\}.$$

In Section 4 we will make this scheme concrete for scalar and diagonal inversion matrices.

3.4 Bounds Schemes

In spite of the fact that Scheme 1 is expressive enough for producing meaningful lower bounds under various structures of the inversion matrix, it does not allow one to incorporate other lower bounds on the root radius of characteristic polynomials whose coefficient matrices admit certain forms, e.g., linear coefficient matrices (see 35 below). Abstracting away from Scheme 1, we now formalize one of the main pillars of this work, i.e., the relation between the amount of computational cost one is willing to invest in executing each iteration and the total number of iterations needed for obtaining a given level of accuracy. We use this relation to form two schemes for establishing lower and upper bounds for p -SCLIs.

Given a compatible set of parameters: a lifting factor p , an inversion matrix $N(X)$, set of quadratic functions $\mathcal{Q}^d(\Sigma)$ and a set of coefficient matrices \mathcal{C} , we denote by $\mathfrak{A}(p, N(X), \mathcal{Q}^d(\Sigma), \mathcal{C})$ the set of consistent p -SCLI optimization algorithms for $\mathcal{Q}^d(\Sigma)$ whose inversion matrix are $N(X)$ and whose coefficient matrices are taken from \mathcal{C} . Furthermore, we denote by $\mathfrak{L}(p, N(X), \mathcal{Q}^d(\Sigma), \mathcal{C})$ the following set of polynomial matrices

$$\left\{ \mathcal{L}(\lambda, X) \triangleq I_d \lambda^p - \sum_{j=0}^{p-1} \mathbb{E} C_j(X) \lambda^j \mid C_j(X) \in \mathcal{C}, \mathcal{L}(1, A) = -N(A)A, \forall A \in \mathcal{S}^d(\Sigma) \right\}.$$

Since both sets are determined by the same set of parameters, the specifications of which will be occasionally omitted for brevity. The natural one-to-one correspondence between these two sets, as manifested by Theorem 4 and Corollary 5, yields

$$\boxed{\min_{A \in \mathfrak{A}} \max_{f, \lambda, b(x) \in \mathcal{Q}^d(\Sigma)} \rho_\lambda(\mathcal{L}_A(\lambda, A)) = \min_{\mathcal{L}(\lambda, X) \in \mathfrak{L}} \max_{A \in \mathcal{S}^d(\Sigma)} \rho_\lambda(\mathcal{L}(\lambda, A))} \quad (27)$$

The importance of Equation (27) stems from its ability to incorporate any bound on the maximal modulus root of polynomial matrices into a general scheme for bounding the iteration complexity of p -SCLIs. This is summarized by the following scheme.

Scheme 2	Lower bounds
Given	a set of p -SCLI optimization algorithms $\mathfrak{A}(p, N(X), \mathcal{Q}^d(\Sigma), \mathcal{C})$
Find	$\rho_* \in [0, 1)$ such that
	$\min_{\mathcal{L}(\lambda, X) \in \mathfrak{L}} \max_{A \in \mathcal{S}^d(\Sigma)} \rho_\lambda(\mathcal{L}(\lambda, A)) \geq \rho_*$
Lower bound:	$\tilde{\Omega} \left(\frac{\rho_*}{1-\rho_*} \ln(1/\epsilon) \right)$

Thus, Scheme 1 is in effect an instantiation of the scheme shown above using Lemma 6. This correspondence of p -SCLI optimization algorithms and polynomial matrices can be also used contrarily to derive efficient algorithm optimization. Indeed, in Section 2.3 we show how FGD, HB and AGD can be formed as optimal instantiations of the following dual scheme.

Scheme 3	Optimal p -SCLI Optimization Algorithms
Given	a set of polynomial matrices $\mathfrak{L}(p, N(X), \mathcal{Q}^d(\Sigma), \mathcal{C})$
Compute	$\rho^* = \min_{\mathcal{L}(\lambda, X) \in \mathfrak{L}} \max_{A \in \mathcal{S}^d(\Sigma)} \rho_\lambda(\mathcal{L}(\lambda, A))$ and denote its minimizer by $\mathcal{L}^*(\lambda, A)$
Upper bound:	The corresponding p -SCLI algorithm for $\mathcal{L}^*(\lambda, A)$
Convergence rate:	$\mathcal{O} \left(\frac{1}{1-\rho^*} \ln(1/\epsilon) \right)$

4. Lower Bounds

In the sequel we derive lower bounds on the convergence rate of p -SCLI optimization algorithms whose inversion matrices are scalar or diagonal, and discuss the assumptions under which these lower bounds meet matching upper bounds. It is likely that this approach can be also effectively applied for block-diagonal inversion, as well as for a much wider set of inversion matrices whose entries depend on a relatively small set of entries of the matrix to be inverted.

4.1 Scalar and Diagonal Inversion Matrices

We derive a lower bound on the convergence rate of p -SCLI optimization algorithms for L -smooth μ -strongly convex functions over \mathbb{R}^d with a scalar inversion matrix $N(X)$ by employing Scheme 1 (see Section 3.3). Note that since the one-dimensional case was already proven in Section 3.2, we may assume that $d \geq 2$.

First, we need to pick a ‘hard’ matrix in $\mathcal{S}^d([\mu, L])$. It turns out that any positive-definite matrix $A \in \mathcal{S}^d([\mu, L])$ for which

$$\{\mu, L\} \subseteq \sigma(A), \quad (28)$$

will meet this criterion. For the sake of concreteness, let us define

$$A \triangleq \text{Diag}(L, \underbrace{\mu, \dots, \mu}_{d-1 \text{ times}}).$$

In which case,

$$-\nu\{\mu, L\} = \sigma(-\mathbb{E}N(A)A),$$

where $\nu I = \mathbb{E}[N(A)]$. Thus, to maintain consistency, it must hold that⁹

$$\nu \in \left(\frac{-2\rho}{L}, 0 \right). \quad (29)$$

9. On a side note, this reasoning also implies that if the spectrum of a given matrix A contains both positive and negative eigenvalues then $A^{-1}b$ cannot be computed using p -SCLIs with scalar inversion matrices.

Next, to bound from below

$$\rho_* \triangleq \max_{\kappa \in [\underline{\mu}]} \left| \frac{\sqrt{\sigma_i}(\sqrt{-\nu}A) - 1}{\sqrt{\sigma_i}} \right| = \max \left\{ \sqrt{\kappa^{-\nu}\mu} - 1, \left| \sqrt{\kappa^{-\nu}L} - 1 \right| \right\},$$

we split the feasible range of ν (29) into three different sub-ranges as follows:

$\sqrt{\kappa^{-\nu}\mu} - 1 < 0$	Case 1 Range: $[-1/L, 0)$ Minimizer: $\nu^* = -1/L$ Lower bounds: $1 - \sqrt{\frac{\mu}{L}}$	$\sqrt{\kappa^{-\nu}\mu} - 1 \geq 0$	N/A
$\sqrt{\kappa^{-\nu}L} - 1 \leq 0$	Case 2 Range: $(-1/\mu, -1/L)$ Minimizer: $-\left(\frac{\sqrt{\kappa^{-\nu}L} + \sqrt{\mu}}{2}\right)^p$ Lower bound: $\frac{\sqrt{\kappa^{-\nu}L/\mu} - 1}{\sqrt{\kappa^{-\nu}L/\mu} + 1}$	Case 3 (requires: $p \geq \log_2 \kappa$) Range: $(-2^p/L, -1/\mu]$ Minimizer: $-1/\mu$ Lower Bound: $\sqrt{\frac{\mu}{L}} - 1$	
$\sqrt{\kappa^{-\nu}L} - 1 > 0$			

Table 1: Lower bound for ρ_* by subranges of ν

Therefore,

$$\rho_* \geq \min \left\{ 1 - \sqrt{\frac{\mu}{L}}, \frac{\sqrt{\kappa^{-\nu}L/\mu} - 1}{\sqrt{\kappa^{-\nu}L/\mu} + 1}, \sqrt{\frac{L}{\mu}} - 1, \sqrt{\frac{\kappa}{\mu}} - 1 \right\} = \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}, \quad (30)$$

where $\kappa \triangleq L/\mu$, upper bounds the condition number of functions in $\mathcal{Q}^d(\mu, L)$. Thus, by Scheme 1, we get the following lower bound on the worse-case iteration complexity,

$$\tilde{\Omega} \left(\frac{\sqrt{\kappa} - 1}{2} \ln(1/\epsilon) \right). \quad (31)$$

As for the diagonal case, it turns out that for any quadratic $f_{A,b}(\mathbf{x}) \in \mathcal{Q}^d(\mu, L)$ which has

$$\begin{pmatrix} \frac{L+\mu}{2} & \frac{L-\mu}{2} \\ \frac{L-\mu}{2} & \frac{L+\mu}{2} \end{pmatrix} \quad (32)$$

as a principal sub-matrix of A , the best p -SCLI optimization algorithm with a diagonal inversion matrix does not improve on the optimal asymptotic convergence rate achieved by scalar inversion matrices (see Section C.4). Overall, we obtain the following theorem.

Theorem 8 *Let A be a consistent p -SCLI optimization algorithm for L -smooth μ -strongly convex functions over \mathbb{R}^d . If the inversion matrix of A is diagonal, then there exists a quadratic function $f_{A,b}(\mathbf{x}) \in \mathcal{Q}^d(\mu, L)$ such that*

$$\mathcal{IC}_A(\epsilon, f_{A,b}(\mathbf{x})) = \tilde{\Omega} \left(\frac{\sqrt{\kappa} - 1}{2} \ln(1/\epsilon) \right), \quad (33)$$

where $\kappa = L/\mu$.

4.2 Is This Lower Bound Tight?

A natural question now arises: is the lower bound stated in Theorem 8 tight? In short, it turns out that for $p = 1$ and $p = 2$ the answer is positive. For $p > 2$, the answer heavily depends on whether a suitable spectral decomposition is within reach. Obviously, computing the spectral decomposition for a given positive definite matrix A is at least as hard as finding the minimizer of a quadratic function whose Hessian is A . To avoid this, we will later restrict our attention to linear coefficient matrices which allow efficient implementation.

A matching upper bound for $p = 1$ In this case the lower bound stated in Theorem 8 is simply attained by FGD (see Section 2.3).

A matching upper bound for $p = 2$ In this case there are two 2-SCLI optimization algorithms which attain this bound, namely, Accelerated Gradient Descent and The Heavy Ball method (see Section 2.3), whose inversion matrices are scalar and correspond to Case 1 and Case 2 in Table 1, i.e.,

$$N_{\text{HB}} = - \left(\frac{2}{\sqrt{L} + \sqrt{\mu}} \right)^2 I_d, \quad N_{\text{AGD}} = \frac{-1}{L} I_d.$$

Although HB obtains the best possible convergence rate in the class of 2-SCLIs with diagonal inversion matrices, it has a major disadvantage. When applied to general smooth and strongly-convex functions, one cannot guarantee global convergence. That is, in order to converge to the corresponding minimizer, HB must be initialized close enough to the minimizer (see Section 3.2.1 in Polyak 1987). Indeed, if the initialization point is too far from the minimizer then HB may diverge as shown in Section 4.5 in Lessard et al. (2014). In contrast to this, AGD attains a global linear convergence with a slightly worse factor. Put differently, the fact HB is highly adapted to quadratic functions prevents it from converging globally to the minimizers of general smooth and strongly convex functions.

A matching upper bound for $p > 2$ In Subsection A we show that when no restriction on the coefficient matrices is imposed, the lower bound shown in Theorem 8 is tight, i.e., for any $p \in \mathbb{N}$ there exists a matching p -SCLI optimization algorithm with scalar inversion matrix whose iteration complexity is

$$\tilde{\mathcal{O}} \left(\frac{\sqrt{\kappa} - 1}{2} \ln(1/\epsilon) \right). \quad (34)$$

In light of the existing lower bound which scales according to $\sqrt{\kappa}$, this result may seem surprising at first. However, there is a major flaw in implementing these seemingly ideal p -SCLIs. In order to compute the corresponding coefficients matrices one has to obtain a very good approximation for the spectral decomposition of the positive definite matrix which defines the optimization problem. Clearly, this approach is rarely practical. To remedy this situation we focus on linear coefficient matrices which admit a relatively low computational cost per iteration. That is, we assume that there exist real scalars $\alpha_1, \dots, \alpha_{p-1}$ and $\beta_1, \dots, \beta_{p-1}$ such that

$$C_j(X) = \alpha_j X + \beta_j I_d, \quad j = 0, 1, \dots, p-1, \quad (35)$$

We believe that for these type of coefficient matrices the lower bound derived in Theorem 8 is not tight. Precisely, we conjecture that for any $0 < \mu < L$ and for any consistent p -SCLI optimization algorithm \mathcal{A} with diagonal inversion matrix and linear coefficients matrices, there exists $f_{A,B}(\mathbf{x}) \in \mathcal{Q}^d([\mu, L])$ such that

$$\rho_\lambda(\mathcal{L}_A(\lambda, X)) \geq \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1},$$

where $\kappa \triangleq L/\mu$. Proving this may allow to derive tight lower bounds for many optimization algorithm in the field of machine learning. Using Scheme 2, which allows to incorporate various lower bounds on the root radius of polynomials, one is able to equivalently express this conjecture as follows: suppose $q(z)$ is a p -degree monic real polynomial such that $q(1) = 0$. Then, for any polynomial $r(z)$ of degree $p-1$ and for any $0 < \mu < L$, there exists $\eta \in [\mu, L]$ such that

$$\rho(q(z) - \eta r(z)) \geq \frac{\sqrt{L/\mu} - 1}{\sqrt{L/\mu} + 1}.$$

That being so, can we do better if we allow families of quadratic functions $\mathcal{Q}^d(\Sigma)$ where Σ are not necessarily continuous intervals? It turns out that the answer is positive. Indeed, in Section B we present a 3-SCLI optimization algorithm with linear coefficient matrices which, by being intimately adjusted to quadratic functions whose Hessian admits large enough spectral gap, beats the lower bound of Nemirovsky and Yudin (3). This apparently contradicting result is also discussed in Section B, where we show that lower bound (3) is established by employing quadratic functions whose Hessian admits spectrum which densely populates $[\mu, L]$.

5. Upper Bounds

Up to this point we have projected various optimization algorithms on the framework of p -SCLI optimization algorithms, thereby converting questions on convergence properties into questions on moduli of roots of polynomials. In what follows, we shall head in the opposite direction. That is, first we define a polynomial (see Definition (2)) which meets a prescribed set of constraints, and then we form the corresponding p -SCLI optimization algorithm. As stressed in Section 4.2, we will focus exclusively on linear coefficient matrices which admit low per-iteration computational cost and allow a straightforward extension to general smooth and strongly convex functions. Surprisingly enough, this allows a systematic recovering of FGD, HB, AGD, as well as establishing new optimization algorithms which allow better utilization of second-order information. This line of inquiry is particularly important due to the obscure nature of AGD, and further emphasizes its algebraic characteristic. We defer stochastic coefficient matrices, as in SDCA, (Section 2.1) to future work.

This section is organized as follows. First we apply Scheme 3 to derive general p -SCLIs with linear coefficients matrices. Next, we recover AGD and HB as optimal instantiations under this setting. Finally, although general p -SCLI algorithms are exclusively specified for quadratic functions, we show how p -SCLIs with linear coefficient matrices can be extended to general smooth and strongly convex functions.

5.1 Linear Coefficient Matrices

In the sequel we instantiate Scheme 3 (see Section 3.4) for $\mathcal{C}_{\text{Linear}}$, the family of deterministic linear coefficient matrices.

First, note that due to consistency constraints, inversion matrices of constant p -SCLIs with linear coefficient matrices must be either constant scalar matrices or else be computationally equivalent to A^{-1} . Therefore, since our motivation for resorting to linear coefficient matrices was efficiency, we can safely assume that $N(X) = \nu I_d$ for some $\nu \in (-2^p/L, 0)$. Following Scheme 3, we now seek the optimal characteristic polynomial in $\mathfrak{E}_{\text{Linear}} \triangleq \mathfrak{E}(p, \nu I_d, \mathcal{Q}^d([\mu, L]), \mathcal{C}_{\text{Linear}})$ with a compatible set of parameters (see Section 3.4). In the presence of linearity, the characteristic polynomials takes the following simplified form

$$\mathcal{L}(\lambda, X) = \lambda^p - \sum_{j=0}^{p-1} (a_j X + b_j I_d) \lambda^j, \quad a_j, b_j \in \mathbb{R}.$$

By (23) we have

$$\rho_\lambda(\mathcal{L}(\lambda, X)) = \max\{|\lambda| \mid \exists i \in [d], \ell_i(\lambda) = 0\},$$

where $\ell_i(\lambda)$ denote the factors of the characteristic polynomial as in (22). That is, denoting the eigenvalues of X by $\sigma_1, \dots, \sigma_d$ we have

$$\ell_i(\lambda) = \lambda^p - \sum_{j=0}^{p-1} (a_j \sigma_i + b_j) \lambda^j = \lambda^p - \sigma_i \sum_{j=0}^{p-1} a_j \lambda^j + \sum_{j=0}^{p-1} b_j \lambda^j.$$

Thus, we can express the maximal root radius of the characteristic polynomial over $\mathcal{Q}^d([\mu, L])$ in terms of the following polynomial

$$\ell(\lambda, \eta) = \lambda^p - (\eta a(\lambda) + b(\lambda)), \quad (36)$$

for some real univariate $p-1$ degree polynomials $a(\lambda)$ and $b(\lambda)$, whereby

$$\max_{A \in \mathfrak{S}^d(\Sigma)} \rho_\lambda(\mathcal{L}(\lambda, A)) = \max_{\eta \in [\mu, L]} \rho(\ell(\lambda, \eta)).$$

That being the case, finding the optimal characteristic polynomial in $\mathfrak{E}_{\text{Linear}}$ translates to the following minimization problem,

$\begin{aligned} & \text{minimize} && \max_{\eta \in [\mu, L]} \rho_\lambda(\ell(\lambda, \eta)) \\ & \text{s.t.} && \ell(1, \eta) = -\nu\eta, \quad \eta \in [\mu, L] \end{aligned} \quad (37)$ $\rho_\lambda(\ell(\lambda, \eta)) < 1 \quad (38)$

(Note that in this case we think of $\mathfrak{E}_{\text{Linear}}$ as a set of polynomials whose variable assumes scalars).

This optimization task can be readily solved for the setting where the lifting factor is $p = 1$, the family of quadratic functions under considerations is $\mathcal{Q}^d(\mu, L)$ and the inversion matrix is $N(X) = \nu I_d$, $\nu \in (-2/L, 0)$. In which case (36) takes the following form

$$\ell(\lambda, \eta) = \lambda - \eta a_0 - b_0,$$

where a_0, b_0 are some real scalars. In order to satisfy (37) for all $\eta \in [\mu, L]$, we have no other choice but to set

$$a_0 = \nu, \quad b_0 = 1,$$

which implies

$$\rho_\lambda(\ell(\lambda, \eta)) = 1 + \nu\eta.$$

Since $\nu \in (-2/L, 0)$, condition 38 follows, as well. The corresponding 1-SCLI optimization algorithm is

$$\mathbf{x}^{k+1} = (I + \nu A)\mathbf{x}^k + \nu \mathbf{b},$$

and its first-order extension (see Section 5.3 below) is precisely FGD (see Section 2.3). Finally, note that the corresponding root radius is bounded from above by

$$\frac{\kappa - 1}{\kappa}$$

for $\nu = -1/L$, the minimizer in Case 2 of Table 1, and by

$$\frac{\kappa - 1}{\kappa + 1}$$

for $\nu = \frac{-2}{\mu+L}$, the minimizer in Case 3 of Table 1. This proves that FGD is optimal for the class of 1-SCLIs with linear coefficient matrices. Figure 5.1 shows how the root radius of the characteristic polynomial of FGD is related to the eigenvalues of the Hessian of the quadratic function under consideration.

5.2 Recovering AGD and HB

Let us now calculate the optimal characteristic polynomial for the setting where the lifting factor is $p = 2$, the family of quadratic functions under considerations is $\mathcal{Q}^d(\mu, L)$ and the inversion matrix is $N(X) = \nu I_d$, $\nu \in (-4/L, 0)$ (recall that the restricted range of ν is due to consistency). In which case (36) takes the following form

$$\ell(\lambda, \eta) = \lambda^2 - \eta(a_1\lambda + a_0) - (b_1\lambda + b_0), \quad (39)$$

for some real scalars a_0, a_1, b_0, b_1 . Our goal is to choose a_0, a_1, b_0, b_1 so as to minimize

$$\max_{\eta \in [\mu, L]} \rho_\lambda(\ell(\lambda, \eta))$$

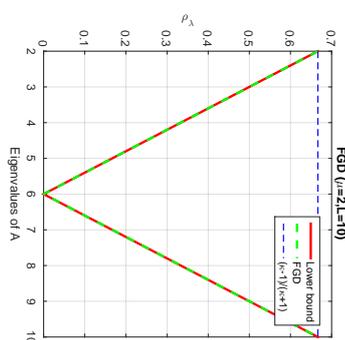


Figure 1: The root radius of FGD vs. various eigenvalues of the corresponding Hessian.

while preserving conditions (37) and (38). Note that $\ell(\lambda, \eta)$, when seen as a function of η , forms a linear path of quadratic functions. Thus, a natural way to achieve this goal is to choose $\ell(\lambda, \eta)$ so that $\ell(\lambda, \mu)$ and $\ell(\lambda, L)$ take the form of the ‘economic’ polynomials introduced in Lemma 6, namely

$$(\lambda - (1 - \sqrt{\tau}))^2$$

for $\tau = -\nu\mu$ and $\tau = -\nu L$, respectively, and hope that for others $\eta \in (\mu, L)$, the roots of $\ell(\lambda, \eta)$ would still be of small magnitude. Note that due to the fact that $\ell(\lambda, \eta)$ is linear in η , condition (37) readily holds for any $\eta \in (\mu, L)$. This yields the following two equations

$$\begin{aligned} \ell(\lambda, \mu) &= (\lambda - (1 - \sqrt{-\nu\mu}))^2, \\ \ell(\lambda, L) &= (\lambda - (1 - \sqrt{-\nu L}))^2. \end{aligned}$$

Substituting (39) for $\ell(\lambda, \eta)$ and expanding the r.h.s. of the equations above we get

$$\begin{aligned} \lambda^2 - (a_1\mu + b_1)\lambda - (a_0\mu + b_0) &= \lambda^2 - 2(1 - \sqrt{-\nu\mu})\lambda + (1 - \sqrt{-\nu\mu})^2, \\ \lambda^2 - (a_1L + b_1)\lambda - (a_0L + b_0) &= \lambda^2 - 2(1 - \sqrt{-\nu L})\lambda + (1 - \sqrt{-\nu L})^2. \end{aligned}$$

Which can be equivalently expressed as the following system of linear equations

$$-(a_1\mu + b_1) = -2(1 - \sqrt{-\nu\mu}), \quad (40)$$

$$-(a_0\mu + b_0) = (1 - \sqrt{-\nu\mu})^2, \quad (41)$$

$$-(a_1L + b_1) = -2(1 - \sqrt{-\nu L}), \quad (42)$$

$$-(a_0L + b_0) = (1 - \sqrt{-\nu L})^2. \quad (43)$$

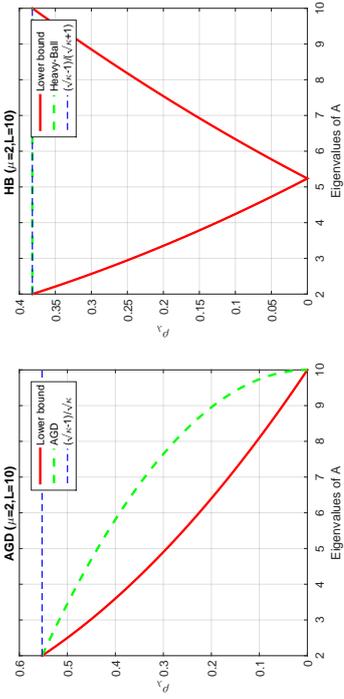


Figure 2: The root radius of AGD and HB vs. various eigenvalues of the corresponding Hessian.

Multiplying Equation (40) by -1 and add to it Equation (42). Next, multiply Equation (41) by -1 and add to it Equation (43) yields

$$\begin{aligned} a_1(\mu - L) &= 2\sqrt{-\nu}(\sqrt{L} - \sqrt{\mu}), \\ a_0(\mu - L) &= (1 - \sqrt{-\nu L})^2 - (1 - \sqrt{-\nu\mu})^2. \end{aligned}$$

Thus,

$$a_1 = \frac{-2\sqrt{-\nu}}{\sqrt{\mu} + \sqrt{L}}, \quad a_0 = \frac{2\sqrt{-\nu}}{\sqrt{\mu} + \sqrt{L}} + \nu.$$

Plugging in $\nu = -1/L$ (see Table 1) into the equations above and solving for b_1 and b_0 yields a 2-SCLI optimization algorithm whose extension (see Section 5.3 below) is precisely AGD. Following the same derivation only this time by setting (see again Table 1)

$$\nu = -\left(\frac{2}{\sqrt{L} + \sqrt{\mu}}\right)^2$$

yields the Heavy-Ball method.

Moreover, using standard formulae for roots of quadratic polynomials one can easily verify that

$$\rho_\lambda(\ell(\lambda, \eta)) \leq \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa}}, \quad \eta \in [\mu, L],$$

for AGD, and

$$\rho_\lambda(\ell(\lambda, \eta)) \leq \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}, \quad \eta \in [\mu, L],$$

for HB. In particular, Condition 38 holds. Figure 5.2 shows how the root radii of the characteristic polynomials of AGD and HB are related to the eigenvalues of the Hessian of the quadratic function under consideration.

Unfortunately, finding the optimal p -SCLIs for $p > 2$ is open and is closely related to the conjecture presented in the end of Section 4.2.

5.3 First-Order Extension for p -SCLIs with Linear Coefficient Matrices

As mentioned before, since the coefficient matrices of p -SCLIs can take any form, it is not clear how to use a given p -SCLI algorithm, efficient as it may be, for minimizing general smooth and strongly convex functions. That being the case, one could argue that recovering the specifications of, say, AGD for quadratic functions does not necessarily imply how to recover AGD itself. Fortunately, consistent p -SCLIs with linear coefficients can be reformulated as optimization algorithms for general smooth and strongly convex functions in a natural way by substituting $\nabla f(\mathbf{x})$ for $A\mathbf{x} + \mathbf{b}$, while preserving the original convergence properties to a large extent. In the sequel we briefly discuss this appealing property, namely, canonical first-order extension, which completes the path from the world of polynomials to the world optimization algorithm for general smooth and strongly convex functions.

Let $\mathcal{A} \triangleq (\mathcal{L}_A(\lambda, X), N(X))$ be a consistent p -SCLI optimization algorithm with a scalar inversion matrix, i.e., $N(X) \triangleq \nu I_d$, $\nu \in (-2^p/L, 0)$, and linear coefficient matrices

$$C_j(X) = a_j X + b_j I_d, \quad j = 0, \dots, p-1, \quad (44)$$

where $a_0, \dots, a_{p-1} \in \mathbb{R}$ and $b_0, \dots, b_{p-1} \in \mathbb{R}$ denote real scalars. Recall that by consistency, for any $f_{A,\mathbf{b}}(\mathbf{x}) \in \mathcal{Q}^d(\Sigma)$, it holds that

$$\begin{aligned} \text{Thus,} \quad & \sum_{j=0}^{p-1} C_j(A) = I + \nu A. \\ & \sum_{j=0}^{p-1} b_j = 1 \text{ and } \sum_{j=0}^{p-1} a_j = \nu. \end{aligned} \quad (45)$$

By the definition of p -SCLIs (Definition 1), we have that

$$\mathbf{x}^k = C_0(A)\mathbf{x}^{k-p} + C_1(A)\mathbf{x}^{k-(p-1)} + \dots + C_{p-1}(A)\mathbf{x}^{k-1} + \nu \mathbf{b}.$$

Substituting $C_j(A)$ for (44), gives

$$\mathbf{x}^k = (a_0 A + b_0)\mathbf{x}^{k-p} + (a_1 A + b_1)\mathbf{x}^{k-(p-1)} + \dots + (a_{p-1} A + b_{p-1})\mathbf{x}^{k-1} + \nu \mathbf{b}.$$

Rearranging and plugging in 45, we get

$$\begin{aligned} \mathbf{x}^k &= a_0(A\mathbf{x}^{k-p} + \mathbf{b}) + a_1(A\mathbf{x}^{k-(p-1)} + \mathbf{b}) + \dots + a_{p-1}(A\mathbf{x}^{k-1} + \mathbf{b}) \\ &\quad + b_0\mathbf{x}^{k-p} + b_1\mathbf{x}^{k-(p-1)} + \dots + b_{p-1}\mathbf{x}^{k-1}. \end{aligned}$$

Finally, by substituting $A\mathbf{x} + \mathbf{b}$ for its analog $\nabla f(\mathbf{x})$, we arrive at the following canonical first-order extension of \mathcal{A}

$$\mathbf{x}^k = \sum_{j=0}^{p-1} b_j \mathbf{x}^{k-(p-j)} + \sum_{j=0}^{p-1} a_j \nabla f(\mathbf{x}^{k-(p-j)}). \quad (46)$$

Being applicable to a much wider collection of functions, how well should we expect the canonical extensions to behave? The answer is that when initialized close enough to the minimizer, one should expect a linear convergence of essentially the same rate. A formal statement is given by the theorem below which easily follows from Theorem 1 in Section 2.1, Polyak (1987) for

$$g(\mathbf{x}^{k-p}, \mathbf{x}^{k-(p-1)}, \dots, \mathbf{x}^{k-1}) = \sum_{j=0}^{p-1} b_j \mathbf{x}^{k-(p-j)} + \sum_{j=0}^{p-1} a_j \nabla f(\mathbf{x}^{k-(p-j)}).$$

Theorem 9 *Suppose $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is an L -smooth μ -strongly convex function and let \mathbf{x}^* denotes its minimizer. Then, for every $\epsilon > 0$, there exist $\delta > 0$ and $C > 0$ such that if*

$$\|\mathbf{x}^j - \mathbf{x}^*\| \leq \delta, \quad j = 0, \dots, p-1,$$

then

$$\|\mathbf{x}^k - \mathbf{x}^0\| \leq C(\rho^* + \epsilon)^k, \quad k = p, p+1, \dots,$$

where

$$\rho^* = \sup_{\eta \in \Sigma} \rho \left(\lambda^p - \sum_{j=0}^{p-1} (a_j \eta + b_j) \lambda^j \right).$$

Unlike general p -SCLIs with linear coefficient matrices which are guaranteed to converge only when initialized close enough to the minimizer, AGD converges linearly, regardless of the initialization points, for any smooth and strongly convex function. This fact merits further investigation as to the precise principles which underlie p -SCLIs of this kind.

Appendix A. Optimal p -SCLI for Unconstrained Coefficient Matrices

In the sequel we use Scheme 3 (see Section 3.4) to show that, when no constraints are imposed on the functional dependency of the coefficient matrices, the lower bound shown in Theorem 8 is tight. To this end, recall that in Lemma 6 we showed that the lower bound on the maximal modulus of roots of a polynomials which evaluate at $z = 1$ to some $r \geq 0$ is uniquely attained by the following polynomial

$$q_r^*(z) \triangleq (z - (1 - \sqrt[r]{r}))^p$$

Thus, by choosing coefficient matrices which admit the same form, we obtain the optimal convergence rate as stated in Theorem 8.

Concretely, let $p \in \mathbb{N}$ be some lifting factor, let $N(X) = \nu L$, $\nu \in (-2^p/L, 0)$ be a fixed scalar matrix and let $f_{A,\mathbf{b}}(\mathbf{x}) \in \mathcal{Q}^d(2)$ be some quadratic function. Lemma 6 implies that for each $\eta \in \sigma(-\nu A)$ we need the corresponding factor of the characteristic polynomial to be

$$\begin{aligned} \ell_j(\lambda) &= (\lambda - (1 - \sqrt[p]{\eta}))^p \\ &= \sum_{k=0}^p \binom{p}{k} (\sqrt[p]{-\nu\eta} - 1)^{p-k} \lambda^k \end{aligned} \quad (47)$$

This is easily accomplished using the spectral decomposition of A by

$$\Lambda \triangleq U^T A U$$

where U is an orthogonal matrix and Λ is a diagonal matrix. Note that since A is a positive definite matrix such a decomposition must always exist. We define p coefficient matrices C_0, C_1, \dots, C_{p-1} in accordance with Equation (47) as follows

$$C_k = U \begin{pmatrix} -\binom{p}{k} (\sqrt[p]{-\nu\Lambda_{11}} - 1)^{p-k} & & & \\ & -\binom{p}{k} (\sqrt[p]{-\nu\Lambda_{22}} - 1)^{p-k} & & \\ & & \ddots & \\ & & & -\binom{p}{k} (\sqrt[p]{-\nu\Lambda_{dd}} - 1)^{p-k} \end{pmatrix} U^T.$$

By using Theorem 5, it can be easily verified that these coefficient matrices form a consistent p -SCLI optimization algorithm whose characteristic polynomial's root radius is

$$\max_{j=1, \dots, d} \left| \sqrt[p]{-\nu \mu_j} - 1 \right|.$$

Choosing

$$\nu = - \left(\frac{2}{\sqrt[p]{L} + \sqrt[p]{\mu}} \right)^p$$

according to Table 1, produces an optimal p -SCLI optimization algorithm for this set of parameters. It is noteworthy that other suitable decompositions can be used for deriving optimal p -SCLIs, as well.

As a side note, since the cost of computing each iteration in \mathbb{R}^{pd} grows linearly with the lifting factor p , the optimal choice of p with respect to the condition number κ yields a p -SCLI optimization algorithm whose iteration complexity is $\Theta(\ln(\kappa) \ln(1/\epsilon))$. Clearly, this result is of theoretical interest only, as this would require a spectral decomposition of A , which, if no other structural assumptions are imposed, is an even harder task than computing the minimizer of $f_{A,\mathbf{b}}(\mathbf{x})$.

Appendix B. Lifting Factor ≥ 3

In Section 4.2 we conjecture that for any p -SCLI optimization algorithm $\mathcal{A} \triangleq (\mathcal{L}(\lambda, X), N(X))$, with diagonal inversion matrix and linear coefficient matrices there exists some $A \in \mathcal{Q}^d([\mu, L])$ such that

$$\rho_\lambda(\mathcal{L}(\lambda, X)) \geq \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}, \quad (48)$$

where $\kappa \triangleq L/\mu$. However, it may be possible to overcome this barrier by focusing on a subclass of $\mathcal{Q}^d([\mu, L])$. Indeed, recall that the polynomial analogy of this conjecture states that for any monic real p degree polynomial $q(z)$ such that $q(1) = 0$ and for any polynomial $r(z)$ of degree $p - 1$, there exists $\eta \in [\mu, L]$ such that

$$\rho(q(z) - \eta r(z)) \geq \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}.$$

This implies that we may be able to tune $q(z)$ and $r(z)$ so as to obtain a convergence rate, which breaks Inequality (48), for quadratic function whose Hessian's spectrum does not spread uniformly across $[\mu, L]$.

Let us demonstrate this idea for $p = 3$, $\mu = 2$ and $L = 100$. Following the exact same derivation used in the last section, let us pick

$$q(z, \eta) \triangleq z^3 - (\eta a(z) + b(z))$$

numerically, so that

$$\begin{aligned} q(z, \mu) &= \left(z - (1 - \sqrt[3]{-\nu\mu}) \right)^3 \\ q(z, L) &= \left(z - (1 - \sqrt[3]{-\nu\mu}) \right)^3 \end{aligned}$$

where

$$\nu = - \left(\frac{2}{\sqrt[3]{L} + \sqrt[3]{\mu}} \right)^3$$

The resulting 3-CLI optimization algorithm \mathcal{A}_3 is

$$\mathbf{x}^k = C_2(X)\mathbf{x}^{k-1} + C_1(X)\mathbf{x}^{k-2} + C_0(X)\mathbf{x}^{k-3} + N(X)b$$

where

$$\begin{aligned} C_0(X) &\approx 0.1958L_d - 0.0038X \\ C_1(X) &\approx -0.9850L_d \\ C_2(X) &\approx 1.7892L_d - 0.0351X \\ N(X) &\approx -0.0389L_d \end{aligned}$$

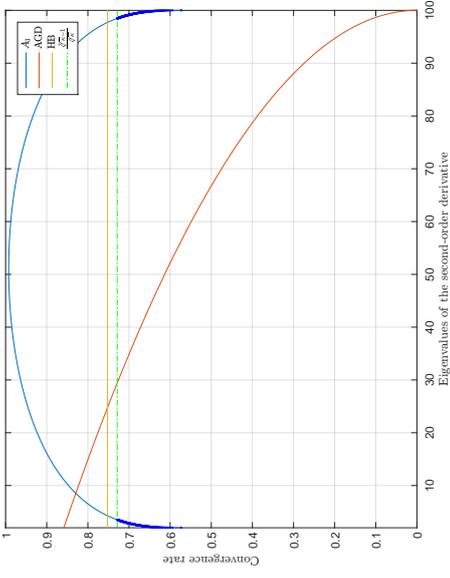


Figure 3: The convergence rate of AGD and \mathcal{A}_3 vs. the eigenvalues of the second-order derivatives. It can be seen that the asymptotic convergence rate of \mathcal{A}_3 for quadratic functions whose second-order derivative comprises eigenvalues which are close to the edges of $[2, 100]$, is faster than AGD and goes below the theoretical lower bound for first-order optimization algorithm $\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$.

As opposed to the algorithm described in Section A, when employing linear coefficient matrices no knowledge regarding the eigenvectors of A is required. As each eigenvalue of the second-order derivative corresponds to a bound on the convergence rate, one can verify by Figure 3 that

$$\rho_\lambda(\mathcal{L}_{\mathcal{A}_3}(\lambda, X)) \leq \frac{\sqrt[3]{\kappa} - 1}{\sqrt[3]{\kappa}}$$

for any $X \in \mathcal{Q}^d([2, 100])$ which satisfies

$$\sigma(A) \subseteq \hat{\Sigma} \triangleq [2, 2 + \epsilon] \cup [100 - \epsilon, 100], \quad \epsilon \approx 1.5.$$

Thus, \mathcal{A}_3 outperforms AGD for this family of quadratic functions.

Let us demonstrate the gain in the performance allowed by \mathcal{A}_3 in a very simple setting. Define A to be $\text{Diag}(\mu, L)$ rotated counter-clockwise by 45° , that is

$$A = \mu \begin{pmatrix} \frac{1}{\sqrt{2}} & \\ & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \\ & \frac{1}{\sqrt{2}} \end{pmatrix}^\top + L \begin{pmatrix} \frac{1}{\sqrt{2}} & \\ & \frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}} & \\ & \frac{1}{\sqrt{2}} \end{pmatrix}^\top = \begin{pmatrix} \frac{\mu+L}{2} & \\ & \frac{\mu-L}{2} \end{pmatrix}.$$

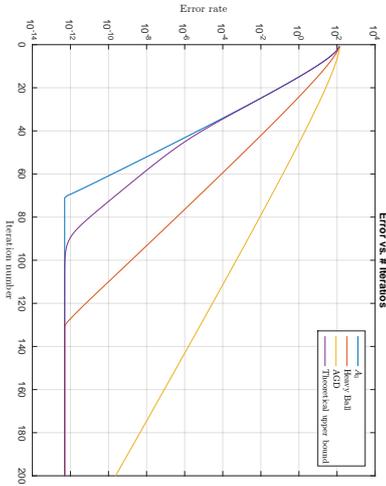


Figure 4: The error rate of \mathcal{A}_3 , AGD and HB vs. # iterations for solving a simple quadratic minimization task. The convergence rate of \mathcal{A}_3 is bounded from above by $\frac{\sqrt[3]{\kappa}-1}{\sqrt[3]{\kappa}}$ as implied by theory.

Furthermore, define $\mathbf{b} = -\mathbf{A}(100, 100)^\top$. Note that $f_{\mathbf{A}, \mathbf{b}}(\mathbf{x}) \in \mathcal{Q}^2(\hat{\Sigma})$ and that its minimizer is simply $(100, 100)^\top$. Figure 4 shows the error of \mathcal{A}_3 , AGD and HB vs. iteration number. All algorithms are initialized at $\mathbf{x}^0 = 0$. Since \mathcal{A}_3 is a first-order optimization algorithm, by the lower bound shown in (3) there must exist some quadratic function $f_{\mathbf{A}_{\text{lb}}, \mathbf{b}_{\text{lb}}}(\mathbf{x}) \in \mathcal{Q}^2(\mu, L)$ such that

$$\mathcal{EC}_{\mathcal{A}_3}(\epsilon, f_{\mathbf{A}_{\text{lb}}, \mathbf{b}_{\text{lb}}}(\mathbf{x})) \geq \tilde{\Omega}(\sqrt{\kappa} \ln(1/\epsilon)). \tag{49}$$

But, since

$$\mathcal{EC}_{\mathcal{A}_3}(\epsilon, f_{\mathbf{A}, \mathbf{b}}(\mathbf{x})) \leq \mathcal{O}(\sqrt[3]{\kappa} \ln(1/\epsilon)) \tag{50}$$

for every $f_{\mathbf{A}, \mathbf{b}}(\mathbf{x}) \in \mathcal{Q}^2(\hat{\Sigma})$, we must have $f_{\mathbf{A}_{\text{lb}}, \mathbf{b}_{\text{lb}}}(\mathbf{x}) \in \mathcal{Q}^2(\mu, L) \setminus \mathcal{Q}^2(\hat{\Sigma})$. Indeed, in the somewhat simpler form of the general lower bound for first-order optimization algorithms,

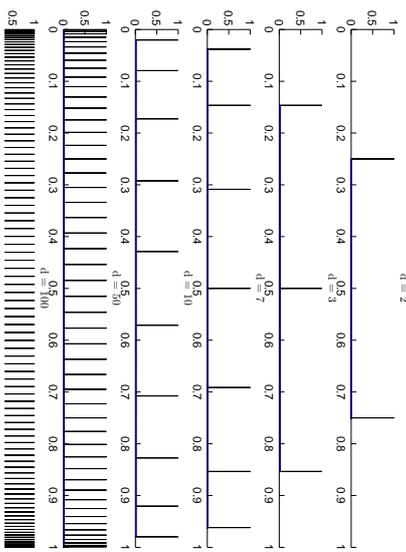


Figure 5: The spectrum of \mathbf{A}_{lb} , as used in the derivation of Nesterov’s lower bound, for problem space of various dimensions.

Nesterov (see Nesterov 2004) considers the following 1-smooth 0-strongly convex function¹⁰

$$\mathbf{A}_{\text{lb}} = \frac{1}{4} \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ 0 & -1 & 2 & -1 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & -1 & 2 & -1 \\ 0 & \cdots & \cdots & 0 & -1 & 2 \end{pmatrix}, \mathbf{b}_{\text{lb}} = - \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

As demonstrated by Figure 5, $\sigma(\mathbf{A}_{\text{lb}})$ densely fills $[\mu, L]$.

Consequently, we expect that whenever adjacent eigenvalues of the second-order derivatives are relatively distant, one should be able to minimize the corresponding quadratic function faster than the lower bound stated in 3. This technique can be further generalized to $p > 3$ using the same ideas. Also, a different approach is to use quadratic (or even higher degree) coefficient matrices to exploit other shapes of spectra. Clearly, the applicability of both approaches heavily depends the existence of spectra of this type in real applications.

¹⁰. Although $f_{\mathbf{A}_{\text{lb}}, \mathbf{b}_{\text{lb}}}(\mathbf{x})$ is not strongly convex, the lower bound for strongly convex function is obtained by shifting the spectrum using a regularization term $\mu/2 \|\mathbf{x}\|^2$. In which case, the shape of the spectrum is preserved.

Appendix C. Proofs

C.1 Proof of Theorem 4

The simple idea behind proof of Theorem 4 is to express the dynamic of a given p -SCLI optimization algorithm as a recurrent application of linear operator. To analyze the latter, we employ the Jordan form which allows us to bind together the maximal magnitude eigenvalue and the convergence rate. Prior to proving this theorem, we first need to introduce some elementary results in linear algebra.

C.1.1 LINEAR ALGEBRA PRELIMINARIES

We prove two basic lemmas which allow to determine under what conditions does a recurrence application of linear operators over finite dimensional spaces converge, as well as to compute the limit of matrices powers series. It is worth noting that despite of being a very elementary result in Matrix theory and in the theory of power methods, the lower bound part of the first lemma does not seem to appear in this form in standard linear algebra literature.

Lemma 10 *Let A be a $d \times d$ square matrix.*

- *If $\rho(A) > 0$ then there exists $C_A > 0$ such that for any $\mathbf{u} \in \mathbb{R}^d$ and for any $k \in \mathbb{N}$ we have*

$$\|A^k \mathbf{u}\| \leq C_A k^{m-1} \rho(A)^k \|\mathbf{u}\|,$$

where m denotes the maximal index of eigenvalues whose modulus is maximal.

In addition, there exists $c_A > 0$ and $\mathbf{r} \in \mathbb{R}^d$ such that for any $\mathbf{u} \in \mathbb{R}^d$ which satisfies $\langle \mathbf{u}, \mathbf{r} \rangle \neq 0$ we have

$$\|A^k \mathbf{u}\| \geq c_A k^{m-1} \rho(A)^k \|\mathbf{u}\|,$$

for sufficiently large $k \in \mathbb{N}$.

- *If $\rho(A) = 0$ then A is a nilpotent matrix. In which case, both lower and upper bounds mentioned above hold trivially for any $\mathbf{u} \in \mathbb{R}^d$ for sufficiently large k .*

Proof Let P be a $d \times d$ invertible matrix such that

$$P^{-1}AP = J,$$

where J is a Jordan form of A , namely, J is a block-diagonal matrix such that $J = \bigoplus_{i=1}^s J_{k_i}(\lambda_i)$ where $\lambda_1, \lambda_2, \dots, \lambda_s$ are eigenvalues of A in a non-increasing order, whose indices are k_1, \dots, k_s , respectively. w.l.o.g. we may assume that $|\lambda_1| = \rho(A)$ and that the corresponding index, which we denote by m , is maximal over all eigenvalues of maximal magnitude. Let Q_1, Q_2, \dots, Q_s and R_1, R_2, \dots, R_s denote partitioning of the columns of P and the rows of P^{-1} , respectively, which conform with the Jordan blocks of A .

Note that for all $i \in [d]$, $J_{k_i}(0)$ is a nilpotent matrix of an order k_i . Therefore, for any (λ_i, k_i) and $k \geq k_i - 1$ we have

$$\begin{aligned} J_{k_i}(\lambda_i)^k &= (\lambda_i J_{k_i} + J_{k_i}(0))^k \\ &= \sum_{j=0}^k \binom{k}{j} \lambda_i^{k-j} J_{k_i}(0)^j \\ &= \sum_{j=0}^{k_i-1} \binom{k}{j} \lambda_i^{k-j} J_{k_i}(0)^j. \end{aligned}$$

Thus, for non-zero eigenvalues we have

$$\begin{aligned} J_{k_i}(\lambda_i)^k / (k^{m-1} \lambda_i^k) &= \sum_{j=0}^{k_i-1} \frac{\binom{k}{j} \lambda_i^{k-j} J_{k_i}(0)^j}{k^{m-1} \lambda_i^k} \\ &= \sum_{j=0}^{k_i-1} \binom{k}{j} \left(\frac{\lambda_i}{\lambda_1} \right)^k \frac{J_{k_i}(0)^j}{\lambda_i^j}. \end{aligned} \quad (51)$$

The rest of the proof pivots around the following equality which holds for any $\mathbf{u} \in \mathbb{R}^{pd}$,

$$\begin{aligned} \|A^k \mathbf{u}\| &= \|PJ^k P^{-1} \mathbf{u}\| \\ &= \left\| \sum_{i=1}^{s'} Q_i J_{k_i}(\lambda_i)^k R_i \mathbf{u} \right\| \\ &= k^{m-1} \rho(A)^k \left\| \sum_{i=1}^{s'} Q_i \left(J_{k_i}(\lambda_i) / (k^{m-1} \lambda_i^{k_i}) \right) R_i \mathbf{u} \right\|, \end{aligned} \quad (52)$$

where s' denotes the smallest index such that $\lambda_i = 0$ for $i > s'$, in case there are zero eigenvalues. Plugging in 51 yields,

$$\|A^k \mathbf{u}\| = k^{m-1} \rho(A)^k \underbrace{\left\| \sum_{i=1}^{s'} Q_i \left(\sum_{j=0}^{k_i-1} \frac{\binom{k}{j}}{k^{m-1}} \left(\frac{\lambda_i}{\lambda_1} \right)^k \frac{J_{k_i}(0)^j}{\lambda_i^j} \right) R_i \mathbf{u} \right\|}_{\mathbf{w}_k}. \quad (53)$$

Let us denote the sequence of vectors in the r.h.s of the preceding inequality by $\{\mathbf{w}_k\}_{k=1}^\infty$. Showing that the norm of $\{\mathbf{w}_k\}_{k=1}^\infty$ is bounded from above and away from zero will conclude the proof. Deriving an upper bound is straightforward.

$$\begin{aligned} \|\mathbf{w}_k\| &\leq \sum_{i=1}^{s'} \left\| Q_i \left(\sum_{j=0}^{k_i-1} \frac{\binom{k}{j}}{k^{m-1}} \left(\frac{\lambda_i}{\lambda_1} \right)^k \frac{J_{k_i}(0)^j}{\lambda_i^j} \right) R_i \mathbf{u} \right\| \\ &\leq \|\mathbf{u}\| \sum_{i=1}^{s'} \|Q_i\| \|R_i\| \sum_{j=0}^{k_i-1} \left\| \frac{\binom{k}{j}}{k^{m-1}} \left(\frac{\lambda_i}{\lambda_1} \right)^k \frac{J_{k_i}(0)^j}{\lambda_i^j} \right\|. \end{aligned} \quad (54)$$

Since for all $i \in [d]$ we have

$$\frac{\binom{k}{i}}{k^{m-1}} \left(\frac{\lambda_i}{\lambda_1}\right)^k \rightarrow 0 \quad \text{or} \quad \left| \frac{\binom{k}{i}}{k^{m-1}} \left(\frac{\lambda_i}{\lambda_1}\right)^k \right| \rightarrow 1$$

it holds that Inequality (54) can be bounded from above by some positive scalar C_A . Plugging it in into 53 yields

$$\|A^k \mathbf{u}\| \leq C_A k^{m-1} \rho(A)^k \|\mathbf{u}\|.$$

Deriving a lower bound on the norm of $\{\mathbf{w}_k\}$ is a bit more involved. First, we define the following set of Jordan blocks which govern the asymptotic behavior of $\|\mathbf{w}_k\|$

$$\mathcal{I} \triangleq \{i \in [s] \mid |\lambda_i| = \rho(A) \text{ and } k_i = m\}.$$

Equation (51) implies that for all $i \notin \mathcal{I}$

$$J_{k_i}(\lambda_i)^k / (k^{m-1} \lambda_i^k) \rightarrow 0 \text{ as } k \rightarrow \infty.$$

As for $i \in \mathcal{I}$, the first $k_i - 1$ terms in Equation (51) tend to zero. The last term is a matrix whose entries are all zeros, except for the last entry in the first row which equals

$$\frac{\binom{k}{m-1}}{k^{m-1}} \left(\frac{\lambda_i}{\lambda_1}\right)^k \frac{1}{(\lambda_i^{m-1})} \sim \left(\frac{\lambda_i}{\lambda_1}\right)^k \frac{1}{(\lambda_i^{m-1})}$$

(here, two positive sequences a_k, b_k are asymptotic equivalence, i.e., $a_k \sim b_k$, if $a_k/b_k \rightarrow 1$). By denoting the first column of each Q_i by \mathbf{q}_i and the last row in each R_i by \mathbf{r}_i^\top , we get

$$\begin{aligned} \|\mathbf{w}_k\| &\sim \left\| \sum_{i \in \mathcal{I}} \left(\frac{\lambda_i}{\lambda_1}\right)^k \frac{1}{\lambda_i^{m-1}} Q_i J_m(0)^{m-1} R_i \mathbf{u} \right\| \\ &= \left\| \sum_{i \in \mathcal{I}} \left(\frac{\lambda_i}{\lambda_1}\right)^k \frac{\mathbf{q}_i \mathbf{r}_i^\top \mathbf{u}}{\lambda_i^{m-1}} \right\|. \end{aligned}$$

Now, if \mathbf{u} satisfies $\mathbf{r}_1^\top \mathbf{u} \neq 0$ then since $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_I$ are linearly independent, we see that the preceding can be bounded from below by some positive constant $c_A > 0$ which does not depend on k . That is, there exists $c_A > 0$ such that $\|\mathbf{w}_k\| > c_A$ for sufficiently large k . Plugging it in into Equation (53) yields

$$\|A \mathbf{u}\| \geq c_A k^{m-1} \rho(A)^k \|\mathbf{u}\|$$

for any $\mathbf{u} \in \mathbb{R}^d$ such that $\langle \mathbf{u}, \mathbf{r}_1 \rangle \neq 0$ and for sufficiently large k . ■

The following is a well-known fact regarding *Neuman series*, sum of powers of square matrices, which follows easily from Lemma 10.

Lemma 11 *Suppose A is a square matrix. Then, the following statements are equivalent:*

1. $\rho(A) < 1$.
2. $\lim_{k \rightarrow \infty} A^k = 0$.
3. $\sum_{k=0}^{\infty} A^k$ converges.

In which case, $(I - A)^{-1}$ exists and $(I - A)^{-1} = \sum_{k=0}^{\infty} A^k$.

Proof First, note that all norms on a finite-dimensional space are equivalent. Thus, the claims stated in (2) and (3) are well-defined.

The fact that (1) and (2) are equivalent is a direct implication of Lemma 10. Finally, the equivalence of (2) and (3) may be established using the following identity

$$(I - A) \sum_{k=0}^{m-1} A^k = I - A^m, \quad m \in \mathbb{N}.$$

■

C.1.2 CONVERGENCE PROPERTIES

Let us now analyze the convergence properties of p -SCLI optimization algorithms. First, note that update rule (14) can be equivalently expressed as a single step rule by introducing new variables in some possibly higher-dimensional Euclidean space \mathbb{R}^{pd} ,

$$\mathbf{z}^0 = (\mathbf{x}^0, \mathbf{x}^1, \dots, \mathbf{x}^{p-1})^\top \in \mathbb{R}^{pd}, \quad \mathbf{z}^k = M(X) \mathbf{z}^{k-1} + U N(X) \mathbf{b}, \quad k = 1, 2, \dots \quad (55)$$

where

$$U \triangleq \underbrace{(0_d, \dots, 0_d, I_d)}_{p-1 \text{ times}}^\top \in \mathbb{R}^{pd \times d}, \quad (56)$$

and where $M(X)$ is a mapping from $\mathbb{R}^{d \times d}$ to $\mathbb{R}^{pd \times pd}$ -valued random variables which admits the following generalized form of companion matrices

$$\begin{pmatrix} 0_d & I_d & & & \\ & 0_d & I_d & & \\ & & \ddots & \ddots & \\ & & & 0_d & \ddots \\ C_0(X) & \dots & C_{p-2}(X) & C_{p-1}(X) & I_d \end{pmatrix}. \quad (57)$$

Following the convention in the field of linear iterative methods, we call $M(X)$ the *iteration matrix*. Note that in terms of the formulation given in (55), consistency w.r.t. $A \in \mathcal{S}^d(\Sigma)$ is equivalent to

$$\mathbb{E} \mathbf{z}^k \rightarrow \underbrace{(-A^{-1} \mathbf{b}, \dots, -A^{-1} \mathbf{b})^\top}_{p \text{ times}} \quad (58)$$

regardless of the initialization points and for any $\mathbf{b} \in \mathbb{R}^d$ and $\mathbf{z}^0 \in \mathbb{R}^{pd}$.

To improve readability, we shall omit the functional dependency of the iteration, inversion and coefficient matrices on X in the following discussion. Furthermore, Equation (55) can be used to derive a simple expression of \mathbf{z}^k , in terms of previous iterations as follows

$$\begin{aligned} \mathbf{z}^1 &= M^{(0)}\mathbf{z}^0 + UN^{(0)}\mathbf{b}, \\ \mathbf{z}^2 &= M^{(1)}\mathbf{z}^1 + UN^{(1)}\mathbf{b} = M^{(1)}M^{(0)}\mathbf{z}^0 + M^{(1)}UN^{(0)}\mathbf{b} + UN^{(1)}\mathbf{b}, \\ \mathbf{z}^3 &= M^{(2)}\mathbf{z}^2 + UN^{(2)}\mathbf{b} = M^{(2)}M^{(1)}M^{(0)}\mathbf{z}^0 + M^{(2)}M^{(1)}UN^{(0)}\mathbf{b} + M^{(2)}UN^{(1)}\mathbf{b} + UN^{(2)}\mathbf{b}, \\ &\vdots \\ \mathbf{z}^k &= \prod_{j=0}^{k-1} M^{(j)}\mathbf{z}^0 + \sum_{m=1}^{k-1} \prod_{j=m}^{k-1} M^{(j)}UN^{(m-1)}\mathbf{b} + UN^{(k-1)}\mathbf{b}, \\ &= \prod_{j=0}^{k-1} M^{(j)}\mathbf{z}^0 + \sum_{m=1}^k \left(\prod_{j=m}^{k-1} M^{(j)} \right) UN^{(m-1)}\mathbf{b}. \end{aligned}$$

where $(M^{(0)}, N^{(0)}), \dots, (M^{(k-1)}, N^{(k-1)})$ are k i.i.d realizations of the corresponding iteration matrix and inversion matrix, respectively. We follow the convention of defining an empty product as the identity matrix and defining the multiplication order of factors of abbreviated product notation as multiplication from the highest index to the lowest, i.e., $\prod_{j=1}^k M^{(j)} = M^{(k)} \dots M^{(1)}$. Taking the expectation of both sides yields

$$\mathbb{E}\mathbf{z}^k = \mathbb{E}[M]^{k-1}\mathbf{z}^0 + \left(\sum_{j=0}^{k-1} \mathbb{E}[M]^j \right) \mathbb{E}[UN]\mathbf{b}. \quad (59)$$

By Lemma 11, if $\rho(\mathbb{E}M) < 1$ then the first term in the r.h.s of Equation (59) vanishes for any initialization point \mathbf{z}^0 , whereas the second term converges to

$$(I - \mathbb{E}M)^{-1} \mathbb{E}[UN]\mathbf{b},$$

the fixed point of the update rule. On the other hand, suppose that $(\mathbb{E}z^k)_{k=0}^\infty$ converges for any $\mathbf{z}^0 \in \mathbb{R}^d$. Then, this is also true for $\mathbf{z}^0 = 0$. Thus, the second summand in the r.h.s of Equation (59) must converge. Consequently, the sequence $\mathbb{E}[M]^k\mathbf{z}^0$, being a difference of two convergent sequences, converges for all \mathbf{z}^0 , which implies $\rho(\mathbb{E}[M]) < 1$. This proves the following theorem.

Theorem 12 *With the notation above, $(\mathbb{E}z^k)_{k=0}^\infty$ converges for any $\mathbf{z}^0 \in \mathbb{R}^d$ if and only if $\rho(\mathbb{E}[M]) < 1$. In which case, for any initialization point $\mathbf{z}^0 \in \mathbb{R}^d$, the limit is*

$$\mathbf{z}^* \triangleq (I - \mathbb{E}M)^{-1} \mathbb{E}[UN]\mathbf{b}. \quad (60)$$

We now address the more delicate question as to how fast do p -SCLIs converge. To this end, note that by Equation (59) and Theorem 12 we have

$$\begin{aligned} \mathbb{E}[\mathbf{z}^k - \mathbf{z}^*] &= \mathbb{E}[M]^{k-1}\mathbf{z}^0 + \left(\sum_{i=0}^{k-1} \mathbb{E}[M]^i \right) \mathbb{E}[UN]\mathbf{b} - (I - \mathbb{E}M)^{-1} \mathbb{E}[UN]\mathbf{b} \\ &= \mathbb{E}[M]^{k-1}\mathbf{z}^0 + (I - \mathbb{E}M)^{-1} \left((I - \mathbb{E}M) \sum_{i=0}^{k-1} \mathbb{E}[M]^i - I \right) \mathbb{E}[UN]\mathbf{b} \\ &= \mathbb{E}[M]^{k-1}\mathbf{z}^0 - (I - \mathbb{E}M)^{-1} (\mathbb{E}M)^k \mathbb{E}[UN]\mathbf{b} \\ &= \mathbb{E}[M]^{k-1}(\mathbf{z}^0 - \mathbf{z}^*). \end{aligned} \quad (61)$$

Hence, to obtain a full characterization of the convergence rate of $\|\mathbb{E}[\mathbf{z}^k - \mathbf{z}^*]\|$ in terms of $\rho(\mathbb{E}M)$, all we need is to simply apply Lemma 10 with $\mathbb{E}M$.

C.1.3 PROOF

We are now in position to prove Theorem 4. Let $\mathcal{A} \triangleq (\mathcal{L}(\lambda, X), N(X))$ be a p -SCLI algorithm over \mathbb{R}^d , let $M(X)$ denote its iteration matrix and let $f_{A,\mathbf{b}}(\mathbf{x})$ be some quadratic function. According to the previous discussion, there exist $m \in \mathbb{N}$ and $C(A), c(A) > 0$ such that the following hold:

1. For any initialization point $\mathbf{z}^0 \in \mathbb{R}^{pd}$, we have that $(\mathbb{E}z^k)_{k=1}^\infty$ converges to $\mathbf{z}^* \triangleq (I - \mathbb{E}M(A))^{-1} \mathbb{E}[UN(A)]\mathbf{b}$. (62)
2. For any initialization point $\mathbf{z}^0 \in \mathbb{R}^{pd}$ and for any $h \in \mathbb{N}$, (63)

$$\|\mathbb{E}[\mathbf{z}^k - \mathbf{z}^*]\| \leq C_A k^{m-1} \rho(M(A))^k \|\mathbf{z}^0 - \mathbf{z}^*\|.$$
3. There exists $\mathbf{r} \in \mathbb{R}^{pd}$ such that for any initialization point $\mathbf{z}^0 \in \mathbb{R}^{pd}$ which satisfies $\langle \mathbf{z}^0 - \mathbf{z}^*, \mathbf{r} \rangle \neq 0$ and sufficiently large $k \in \mathbb{N}$, (64)

$$\|\mathbb{E}[\mathbf{x}^k - \mathbf{x}^*]\| \geq c_A k^{m-1} \rho(M(A))^k \|\mathbf{z}^0 - \mathbf{z}^*\|.$$

Since iteration complexity is defined over the problem space, we need to derive the same inequalities in terms of

$$\mathbf{x}^k = U^\top \mathbf{z}^k.$$

Note that by linearity we have $\mathbf{x}^* = U^\top \mathbf{z}^*$. For bounding $(\mathbf{x}^k)_{k=1}^\infty$ from above we use (63),

$$\begin{aligned} \|\mathbb{E}[\mathbf{x}^k - \mathbf{x}^*]\| &= \|\mathbb{E}[U^\top \mathbf{z}^k - U^\top \mathbf{z}^*]\| \\ &\leq \|U^\top\| \|\mathbb{E}[\mathbf{z}^k - \mathbf{z}^*]\| \\ &\leq \|U^\top\| C_A k^{m-1} \rho(M)^k \|\mathbf{z}^0 - \mathbf{z}^*\| \\ &= \|U^\top\| C_A k^{m-1} \rho(M)^k \|U\mathbf{x}^0 - U\mathbf{x}^*\| \\ &\leq \|U^\top\| \|U\| C_A k^{m-1} \rho(M)^k \|\mathbf{x}^0 - \mathbf{x}^*\|. \end{aligned} \quad (65)$$

Thus, the same rate as in (63), with a different constant, holds in the problem space. Although the corresponding lower bound takes a slightly different form, its proof is done similarly. Pick $\mathbf{x}^0, \mathbf{x}^1, \dots, \mathbf{x}^{p-1}$ such that the corresponding \mathbf{z}^0 is satisfied the condition in (64). For sufficiently large $k \in \mathbb{N}$, it holds that

$$\begin{aligned} \max_{k=0, \dots, p-1} \left\| \mathbb{E} \mathbf{x}^{k+j} - \mathbb{E} \mathbf{x}^* \right\| &\geq \frac{1}{\sqrt{p}} \sqrt{\sum_{j=0}^{p-1} \left\| \mathbb{E} \mathbf{x}^{k+j} - \mathbb{E} \mathbf{x}^* \right\|^2} \\ &= \frac{1}{\sqrt{p}} \left\| \mathbb{E} [\mathbf{z}^k] - \mathbf{z}^* \right\| \\ &\geq \frac{c_A}{\sqrt{p}} k^{m-1} \rho(M)^k \|\mathbf{z}^0 - \mathbf{z}^*\| \\ &= \frac{c_A}{\sqrt{p}} k^{m-1} \rho(M)^k \sum_{j=0}^{p-1} \|\mathbf{x}^j - \mathbf{x}^*\|^2. \end{aligned} \quad (66)$$

We arrived at the following corollary which states that the asymptotic convergence rate of any p -SCLI optimization algorithm is governed by the spectral radius of its iteration matrix.

Theorem 13 *Suppose \mathcal{A} is a p -SCLI optimization algorithm over $\mathcal{Q}^d(\Sigma)$ and let $M(X)$ denotes its iteration matrix. Then, there exists $m \in \mathbb{N}$ such that for any quadratic function $f_{\mathcal{A},b}(\mathbf{x}) \in \mathcal{Q}^d(\Sigma)$ it holds that*

$$\left\| \mathbb{E} [\mathbf{x}^k - \mathbf{x}^*] \right\| = \mathcal{O} \left(k^{m-1} \rho(M(X))^k \|\mathbf{x}^0 - \mathbf{x}^*\| \right),$$

where \mathbf{x}^* denotes the minimizer of $f_{\mathcal{A},b}(\mathbf{x})$. Furthermore, there exists an initialization point $\mathbf{x}^0 \in \mathbb{R}^d$, such that

$$\max_{k=0, \dots, p-1} \left\| \mathbb{E} \mathbf{x}^{k+j} - \mathbb{E} \mathbf{x}^* \right\| = \Omega \left(\frac{k^{m-1}}{\sqrt{p}} \rho(M(X))^k \|\mathbf{x}^0 - \mathbf{x}^*\| \right).$$

Finally, in the next section we prove that the spectral radius of the iteration matrix equals the root radius of the determinant of the characteristic of polynomial by showing that

$$\det(\lambda I - M(X)) = \det(\mathcal{L}(\lambda, X)).$$

Combining this with the corollary above and by applying Inequality (12) and the like, concludes the proof for Theorem 4.

C.1.4 THE CHARACTERISTIC POLYNOMIAL OF THE ITERATION MATRIX

The following lemma provides an explicit expression for the characteristic polynomial of iteration matrices. The proof is carried out by applying elementary determinant manipulation rules.

Lemma 14 *Let $M(X)$ be the matrix defined in (57) and let A be a given $d \times d$ square matrix. Then, the characteristic polynomial of $\mathbb{E}M(A)$ can be expressed as the following matrix polynomial*

$$\chi_{\mathbb{E}M(A)}(\lambda) = (-1)^{pd} \det \left(\lambda^p I_d - \sum_{k=0}^{p-1} \lambda^k \mathbb{E} C_k(A) \right). \quad (67)$$

Proof As usual, for the sake of readability we omit the functional dependency on A , as well as the expectation operator symbol. For $\lambda \neq 0$ we get,

$$\begin{aligned} \chi_M(\lambda) &= \det(M - \lambda I_{pd}) \\ &= \det \begin{pmatrix} -\lambda I_d & I_d & & & \\ & -\lambda I_d & I_d & & \\ & & \ddots & \ddots & \\ & & & -\lambda I_d & I_d \\ C_0 & & \dots & C_{p-2} & C_{p-1} - \lambda I_d \end{pmatrix} \\ &= \det \begin{pmatrix} -\lambda I_d & I_d & & & \\ & -\lambda I_d & I_d & & \\ & & \ddots & \ddots & \\ & & & -\lambda I_d & I_d \\ 0_d & C_1 + \lambda^{-1} C_0 & \dots & C_{p-2} & C_{p-1} - \lambda I_d \end{pmatrix} \\ &= \det \begin{pmatrix} -\lambda I_d & I_d & & & \\ & -\lambda I_d & I_d & & \\ & & \ddots & \ddots & \\ & & & -\lambda I_d & I_d \\ 0_d & 0_d & C_2 + \lambda^{-1} C_1 + \lambda^{-2} C_0 & \dots & C_{p-2} & C_{p-1} - \lambda I_d \end{pmatrix} \\ &= \det \begin{pmatrix} -\lambda I_d & I_d & & & \\ & -\lambda I_d & I_d & & \\ & & \ddots & \ddots & \\ & & & -\lambda I_d & I_d \\ 0_d & \dots & 0_d & \dots & \sum_{k=1}^p \lambda^{k-p} C_{k-1} - \lambda I_d \end{pmatrix} \\ &= \det(-\lambda I_d)^{p-1} \det \left(\sum_{k=1}^p \lambda^{k-p} C_{k-1} - \lambda I_d \right) \end{aligned}$$

$$\begin{aligned}
&= (-1)^{(p-1)d} \det \left(\sum_{k=1}^p \lambda^{k-1} C_{k-1} - \lambda^p I_d \right) \\
&= (-1)^{pd} \det \left(\lambda^p I_d - \sum_{k=0}^{p-1} \lambda^k C_k \right).
\end{aligned}$$

By continuity we have that the preceding equality holds for $\lambda = 0$ as well. \blacksquare

C.2 Proof of Theorem 5

We prove that consistent p -SCLI optimization algorithms must satisfy conditions (17) and (18). The reverse implication is proven by reversing the steps of the proof.

First, note that (18) is an immediate consequence of Corollary 13, according to which p -SCLIs converge if and only if the root radius of the characteristic polynomial is strictly smaller than 1. As for (18), let $\mathcal{A} \triangleq (\mathcal{L}(\lambda, X), N(X))$ be a consistent p -SCLI optimization algorithm over $\mathcal{Q}^d(\Sigma)$ and let $f_{A, \mathbf{b}}(\mathbf{x}) \in \mathcal{Q}^d(\Sigma)$ be a quadratic function. Furthermore, let us denote the corresponding iteration matrix by $M(X)$ as in (57). By Theorem 12, for any initialization point we have

$$\mathbb{E} \mathbf{z}^k \rightarrow (I - \mathbb{E}M(A))^{-1} U \mathbb{E}[N(A)] \mathbf{b},$$

where U is as defined in (56), i.e.,

$$U \triangleq \underbrace{(0_d, \dots, 0_d, I_d)^\top}_{p-1 \text{ times}} \in \mathbb{R}^{pd \times d}.$$

For the sake of readability we omit the functional dependency on A , as well as the expectation operator symbol. Combining this with Equation (58) yields

$$U^\top (I - M)^{-1} U N \mathbf{b} = -A^{-1} \mathbf{b}.$$

Since this holds for any $\mathbf{b} \in \mathbb{R}^d$, we get

$$U^\top (I - M)^{-1} U N = -A^{-1}.$$

Evidently, N is an invertible matrix. Therefore,

$$U^\top (I - M)^{-1} U = -(NA)^{-1}. \quad (68)$$

Now, recall that

$$M = \begin{pmatrix} 0_d & I_d & & & \\ 0_d & I_d & & & \\ & & \ddots & & \\ & & & \ddots & \\ C_0 & \dots & C_{p-2} & C_{p-1} \end{pmatrix},$$

where C_j denote the coefficient matrices. We partition M as follows

$$\left(\begin{array}{c|c|c} M_{11} & M_{12} & \\ \hline M_{21} & M_{22} & \\ \hline C_0 & \dots & C_{p-2} \end{array} \middle| \begin{array}{c} I_d \\ I_d \\ \vdots \\ 0_d \\ C_{p-1} \end{array} \right).$$

The l.h.s of Equation (68) is in fact the inverse of the Schur Complement of $I - M_{11}$ in $I - M$, i.e.,

$$\begin{aligned}
(I - M_{22} - M_{21}(I - M_{11})^{-1}M_{12})^{-1} &= -(NA)^{-1} \\
I - M_{22} - M_{21}(I - M_{11})^{-1}M_{12} &= -NA \\
M_{22} + M_{21}(I - M_{11})^{-1}M_{12} &= I + NA.
\end{aligned} \quad (69)$$

Moreover, it is straightforward to verify that

$$(I - M_{11})^{-1} = \begin{pmatrix} I_d & I_d & & \\ & I_d & I_d & \\ & & \ddots & \\ & & & I_d \end{pmatrix}$$

Plugging in this into (69) yields

$$\sum_{i=0}^{p-1} C_i = I + NA,$$

or equivalently,

$$\mathcal{L}(1, A) = -NA \quad (70)$$

This concludes the proof.

C.3 Proof of Lemma 6

First, we prove the following Lemma. Let us denote

$$q_r^*(z) \triangleq (z - (1 - \sqrt[r]{r}))^p,$$

where r is some non-negative constant.

Lemma 15 Suppose $q(z)$ is a monic polynomial of degree p with complex coefficients. Then,

$$\rho(q(z)) \leq \left| \sqrt[p]{|q(1)|} - 1 \right| \iff q(z) = q_{|q(1)|}^*(z).$$

Proof As the \Leftarrow statement is clear, we prove here only the \Rightarrow part. By the fundamental theorem of algebra $q(z)$ has p roots. Let us denote these roots by $\zeta_1, \zeta_2, \dots, \zeta_p \in \mathbb{C}$. Equivalently,

$$q(z) = \prod_{i=1}^p (z - \zeta_i).$$

Let us denote $r \triangleq |q(1)|$. If $r \geq 1$ we get

$$\begin{aligned} r &= \left| \prod_{i=1}^p (1 - \zeta_i) \right| = \prod_{i=1}^p |1 - \zeta_i| \leq \prod_{i=1}^p (1 + |\zeta_i|) \\ &\leq \prod_{i=1}^p (1 + |\zeta_i|^r) = \prod_{i=1}^p (1 + \zeta_i^r - 1) = r. \end{aligned} \tag{71}$$

Consequently, Inequality (71) becomes an equality. Therefore,

$$|1 - \zeta_i| = 1 + |\zeta_i| = \zeta_i^r, \quad \forall i \in [p]. \tag{72}$$

Now, for any two complex numbers $w, z \in \mathbb{C}$ it holds that

$$|w + z| = |w| + |z| \iff \text{Arg}(w) = \text{Arg}(z).$$

Using this fact in the first equality of Equation (72), we get that $\text{Arg}(-\zeta_i) = \text{Arg}(1) = 0$, i.e., ζ_i are negative real numbers. Writing $-\zeta_i$ in the second equality of Equation (72) instead of $|\zeta_i|$, yields $1 - \zeta_i = \zeta_i^r$, concluding this part of the proof.

The proof for $r \in [0, 1)$ follows along the same lines, only this time we use the reverse triangle inequality,

$$\begin{aligned} r &= \prod_{i=1}^p |1 - \zeta_i| \geq \prod_{i=1}^p (1 - |\zeta_i|) \geq \prod_{i=1}^p (1 - |\zeta_i|^r) \\ &= \prod_{i=1}^p (1 - (1 - \zeta_i^r)) = r. \end{aligned}$$

Note that in the first inequality, we used the fact that $r \in [0, 1) \implies |\zeta_i| \leq 1$ for all i . ■

The proof for Lemma 6 now follows easily. In case $q(1) \geq 0$, if $q(z) = (z - (1 - \zeta^r))^p$ then, clearly,

$$\rho(q(z)) = \rho((z - (1 - \zeta^r))^p) = |1 - \zeta^r|.$$

Otherwise, according to Lemma 15

$$\rho(q(z)) > |1 - \zeta^r|.$$

In case $q(1) \leq 0$, we must use the assumption that the coefficients are reals (see Remark 16), in which case the mere fact that

$$\lim_{z \in \mathbb{R}, z \rightarrow \infty} q(z) = \infty$$

combined with the Mean-Value theorem implies $\rho(q(z)) \geq 1$. This concludes the proof.

Remark 16 The requirement that the coefficients of $q(z)$ should be real is inevitable. To see why, consider the following polynomial,

$$u(z) = \left(z - \left(1 - 0.5e^{\frac{ix}{3}} \right) \right)^3.$$

Although $u(1) = \left(1 - \left(1 - 1/2e^{\frac{ix}{3}} \right) \right)^3 = -1/8 \leq 0$, it holds that $\rho(u(z)) < 1$. Indeed, not all the coefficients of $u(z)$ are real. Notice that the claim does hold for degree ≤ 3 , regardless of the additional assumption on the coefficients of $u(z)$.

C.4 Bounding the Spectral Radius of Diagonal Inversion Matrices from below Using Scalar Inversion Matrices

We prove a lower bound on the convergence rate of p -SCLI optimization algorithm with diagonal inversion matrices. In particular, we show that for any p -SCLI optimization algorithm whose inversion matrix is diagonal there exists a quadratic function for which it does not perform better than p -SCLI optimization algorithms with scalar inversion matrix. We prove the claim for $d = 2$. The general case follows by embedding the 2-dimensional case as a principal sub-matrix in some higher dimensional matrix in $S^d(\mu, L)$. Also, although here we prove for deterministic p -SCLIs, the stochastic case is straightforward.

Let \mathcal{A} be a p -SCLI optimization algorithm with iteration matrix $M(X)$ (defined in (57)) and diagonal inversion matrix $N(X)$. Define the following positive definite matrix

$$B = \begin{pmatrix} \frac{L+\mu}{2} & \frac{L-\mu}{2} \\ \frac{L-\mu}{2} & \frac{L+\mu}{2} \end{pmatrix}, \tag{73}$$

and note that $\sigma(B) = \{\mu, L\}$. As usual, we wish to derive a lower bound on $\rho(M(B))$. To this end, denote

$$N \triangleq N(B) = \begin{pmatrix} \alpha & 0 \\ 0 & \beta \end{pmatrix},$$

where $\alpha, \beta \in \mathbb{R}$. By a straightforward calculation we get that the eigenvalues of $-NB$ are

$$\begin{aligned} \sigma_{1,2}(\alpha, \beta) &= \frac{-(\alpha + \beta)(L + \mu)}{4} \pm \sqrt{\left(\frac{(\alpha + \beta)(L + \mu)}{4} \right)^2 - \alpha\beta L\mu} \\ &= \frac{-(\alpha + \beta)(L + \mu)}{4} \pm \sqrt{(\alpha + \beta)^2 \frac{(L - \mu)^2}{16} + \frac{1}{4}(\alpha - \beta)^2 L\mu}. \end{aligned} \tag{74}$$

Using similar arguments to the ones which were applied in the scalar case, we get that both eigenvalues of $-NB$ must be strictly positive as well as satisfy

$$\rho(M) \geq \min_{\alpha, \beta} \max \left\{ \left| \sqrt{\sigma_1(\alpha, \beta)} - 1 \right|, \left| \sqrt{\sigma_2(\alpha, \beta)} - 1 \right| \right\}. \quad (75)$$

Equation (74) shows that the minimum of the preceding is obtained for $\nu = \frac{\alpha+\beta}{2}$, which simplifies to

$$\begin{aligned} \max \left\{ \left| \sqrt{\sigma_1(\alpha, \beta)} - 1 \right|, \left| \sqrt{\sigma_2(\alpha, \beta)} - 1 \right| \right\} &\geq \max \left\{ \left| \sqrt{\sigma_1(\nu, \nu)} - 1 \right|, \left| \sqrt{\sigma_2(\nu, \nu)} - 1 \right| \right\} \\ &= \max \left\{ \left| \sqrt{-\nu\mu} - 1 \right|, \left| \sqrt{-\nu L} - 1 \right| \right\}. \end{aligned}$$

The rest of the analysis is carried out similarly to the scalar case, resulting in

$$\rho(M(B)) \geq \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}.$$

References

- Zeyuan Allen-Zhu and Lorenzo Orecchia. A novel, simple interpretation of nesterov’s accelerated method as a combination of gradient and mirror descent. *arXiv preprint arXiv:1407.1537*, 2014.
- Michel Baes. Estimate sequence methods: extensions and approximations. 2009.
- Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.
- MP Drazin, JW Dungey, and KW Gruenberg. Some theorems on commutative matrices. *Journal of the London Mathematical Society*, 1(3):221–228, 1951.
- Harriet Fell. On the zeros of convex combinations of polynomials. *Pacific Journal of Mathematics*, 89(1):43–50, 1980.
- Israel Gohberg, Pnusteter Lancaster, and Leiba Rodman. *Matrix polynomials*, volume 58. SIAM, 2009.
- Nicholas J Higham and Françoise Tisseur. Bounds for eigenvalues of matrix polynomials. *Linear Algebra and its applications*, 358(1):5–22, 2003.
- Bill G Horne. Lower bounds for the spectral radius of a matrix. *Linear algebra and its applications*, 263:261–273, 1997.
- Ting-Zhu Huang and Lin Wang. Improving bounds for eigenvalues of complex matrices using traces. *Linear Algebra and its Applications*, 426(2):841–854, 2007.
- Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.
- Harold J Kushner and George Yin. *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer, 2003.
- Laurent Lessard, Benjamin Recht, and Andrew Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *arXiv preprint arXiv:1408.3595*, 2014.
- Morris Marden. *Geometry of polynomials*. Number 3 in @. American Mathematical Soc., 1966.
- John C Mason and David C Handscomb. *Chebyshev polynomials*. CRC Press, 2002.
- Gradimir V Milovanović and Themistocles M Rassias. Distribution of zeros and inequalities for zeros of algebraic polynomials. In *Functional equations and inequalities*, pages 171–204. Springer, 2000.
- Gradimir V Milovanović, DS Mitrinović, and Th M Rassias. Topics in polynomials. *Extremal Problems, Inequalities, Zeros, World Scientific, Singapore*, 1994.

- Arkadi Nemirovski. Efficient methods in convex programming. 2005.
- AS Nemirovsky and DB Yudin. Problem complexity and method efficiency in optimization. 1983. *Wiley-Interscience*. New York, 1983.
- Yurii Nesterov. *A method of solving a convex programming problem with convergence rate $O(1/k^2)$* . ©, 1983.
- Yurii Nesterov. *Introductory lectures on convex optimization*, volume 87. Springer Science & Business Media, 2004.
- Boris T Polyak. *Introduction to optimization*. Optimization Software New York, 1987.
- Qazi Badur Rahnman and Gerhard Schmeisser. *Analytic theory of polynomials*. Number 26 in @. Oxford University Press, 2002.
- Nicolas Le Roux, Mark Schmidt, and Francis Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. *arXiv preprint arXiv:1202.6258*, 2012.
- Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss. *The Journal of Machine Learning Research*, 14(1):567–599, 2013.
- James C Spall. *Introduction to stochastic search and optimization: estimation, simulation, and control*, volume 65. John Wiley & Sons, 2005.
- Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1139–1147, 2013.
- Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. submitted to *siam j. J. Optim.*, 2008.
- JL Walsh. On the location of the roots of certain types of polynomials. *Transactions of the American Mathematical Society*, 24(3):163–180, 1922.
- Henry Wolkowicz and George PH Styan. Bounds for eigenvalues using traces. *Linear Algebra and Its Applications*, 29:471–506, 1980.
- Qin Zhong and Ting-Zhu Huang. Bounds for the extreme eigenvalues using the trace and determinant. *Journal of Information and Computing Science*, 3(2):118–124, 2008.

Dual Control for Approximate Bayesian Reinforcement Learning

Edgar D. Klenske

Max-Planck-Institute for Intelligent Systems

Spemannstraße 38

72076 Tübingen, Germany

EDGAR.KLENSKE@TUEBINGEN.MPG.DE

Philipp Hennig

Max-Planck-Institute for Intelligent Systems

Spemannstraße 38

72076 Tübingen, Germany

PHILIPP.HENNIG@TUEBINGEN.MPG.DE

Editor: Manfred Opper

Abstract

Control of non-episodic, finite-horizon dynamical systems with uncertain dynamics poses a tough and elementary case of the exploration-exploitation trade-off. Bayesian reinforcement learning, reasoning about the effect of actions and future observations, offers a principled solution, but is intractable. We review, then extend an old approximate approach from control theory—where the problem is known as *dual control*—in the context of modern regression methods, specifically generalized linear regression. Experiments on simulated systems show that this framework offers a useful approximation to the intractable aspects of Bayesian RL, producing structured exploration strategies that differ from standard RL approaches. We provide simple examples for the use of this framework in (approximate) Gaussian process regression and feedforward neural networks for the control of exploration.

Keywords: reinforcement learning, control, Gaussian processes, filtering, Bayesian inference

1. Introduction

The exploration-exploitation trade-off is a central problem of learning in interactive settings, where the learner’s actions influence future observations. In episodic settings, where the control problem is re-instantiated repeatedly with unchanged dynamics, comparably simple notions of exploration can succeed. E.g., assigning an *exploration bonus* to uncertain options (Macready and Wolpert, 1998; Audibert et al., 2009) or acting optimally under one sample from the current probabilistic model of the environment (*Thompson sampling*, see Thompson, 1933; Chapelle and Li, 2011), can perform well (Dearden et al., 1999; Kolter and Ng, 2009; Srinivas et al., 2010). Such approaches, however, do not model the effect of actions on future beliefs, which limits the potential for the balancing of exploration and exploitation. This issue is most drastic in the non-episodic case, the control of a single, ongoing trial. Here, the controller cannot hope to be returned to known states, and exploration must be carefully controlled to avoid disaster.

A principled solution to this problem is offered by *Bayesian reinforcement learning* (Duff, 2002; Poupart et al., 2006; Hennig, 2011): A probabilistic belief over the dynamics and cost of the environment can be used not just to simulate and plan trajectories, but also to reason about changes to the belief from future observations, and their influence on future decisions. An elegant formulation is to combine the physical state with the parameters of the probabilistic model into an augmented dynamical description, then aim to control this system. Due to the inference, the augmented system invariably has strongly nonlinear dynamics, causing prohibitive computational cost—even for finite state spaces and discrete time (Poupart et al., 2006), all the more for continuous space and time (Hennig, 2011).

The idea of augmenting the physical state with model parameters was noted early, and termed *dual control*, by Feldbaum (1960–1961). It seems both conceptual and—by the standards of the time—computational complexity hindered its application. An exception is a strand of several works by Meier, Bar-Shalom, and Tse (Tse et al., 1973; Tse and Bar-Shalom, 1973; Bar-Shalom and Tse, 1976; Bar-Shalom, 1981). These authors developed techniques for limiting the computational cost of dual control that, from a modern perspective, can be seen as a form of approximate inference for Bayesian reinforcement learning. While the Bayesian reinforcement learning community is certainly aware of their work (Duff, 2002; Hennig, 2011), it has not found widespread attention. The first purpose of this paper is to cast their dual control algorithm as an approximate inference technique for Bayesian RL in parametric Gaussian (general least-squares) regression. We then extend the framework with ideas from contemporary machine learning. Specifically, we explain how it can in principle be formulated non-parametrically in a Gaussian process context, and then investigate simple, practical finite-dimensional approximations to this result. We also give a simple, small-scale example for the use of this algorithm for dual control if the environment model is constructed with a feedforward neural network rather than a Gaussian process.

2. Model and Notation

Throughout, we consider discrete-time, finite-horizon dynamic systems (POMDPs) of form

$$x_{k+1} = f_k(x_k, u_k) + \xi_k \quad (\text{state dynamics}) \quad y_k = Cx_k + \gamma_k \quad (\text{observation model}).$$

At time $k \in \{0, \dots, T\}$, $x_k \in \mathbb{R}^n$ is the state, $\xi_k \sim \mathcal{N}(0, Q)$ is a Gaussian disturbance. The control input (continuous action) is denoted u_k ; for simplicity we will assume scalar $u_k \in \mathbb{R}$ throughout. Measurements $y_k \in \mathbb{R}^d$ are observations of x_k , corrupted by Gaussian noise $\gamma_k \sim \mathcal{N}(0, R)$. The generative model thus reads $p(x_{k+1} | x_k, u_k) = \mathcal{N}(x_{k+1}; f_k(x_k, u_k), Q)$ and $p(y_k | x_k) = \mathcal{N}(y_k; Cx_k, R)$, with a linear map $C \in \mathbb{R}^{d \times n}$. Trajectories are vectors $\mathbf{x} = [x_0, \dots, x_T]$, and analogously for \mathbf{u}, \mathbf{y} . We will occasionally use the subset notation $\mathbf{y}_{k:j} = [y_k, \dots, y_j]$. We further assume that dynamics f_k are not known, but can be described up to Gaussian uncertainty by a general linear model with nonlinear features $\phi: \mathbb{R}^n \rightarrow \mathbb{R}^m$ and uncertain matrices A_k, B_k .

$$x_{k+1} = A_k \phi(x_k) + B_k u_k + \xi_k, \quad A_k \in \mathbb{R}^{n \times m}, B_k \in \mathbb{R}^{n \times 1}. \quad (1)$$

To simplify notation, we reshape the elements of A_k and B_k into a parameter vector $\theta_k = [\text{vec}(A_k); \text{vec}(B_k)] \in \mathbb{R}^{(m+1)n}$, and define the reshaping transformations $A(\theta_k): \theta_k \mapsto A_k$

and $B(\theta_k) : \theta_k \mapsto B_k$. At initialization, $k = 0$, the belief over states and parameters is assumed to be Gaussian

$$p \left(\begin{bmatrix} x_0 \\ \theta_0 \end{bmatrix} \right) = \mathcal{N} \left(\begin{bmatrix} x_0 \\ \theta_0 \end{bmatrix}; \begin{bmatrix} \hat{x}_0 \\ \hat{\theta}_0 \end{bmatrix}, \begin{bmatrix} \Sigma_{x_0}^{xx} & \Sigma_{x_0}^{x\theta} \\ \Sigma_{\theta_0}^{x0} & \Sigma_{\theta_0}^{\theta\theta} \end{bmatrix} \right). \quad (2)$$

The control response $B_k u_k$ is linear, a common assumption for physical systems. Nonlinear mappings can be included in a generic form $\phi(x_k, u_k)$, but complicate the following derivations and raise issues of identifiability. For simplicity, we also assume that the dynamics do not change through time: $p(\theta_{k+1} | \theta_k) = \delta(\theta_{k+1} - \theta_k)$. This could be relaxed to an autoregressive model $p(\theta_{k+1} | \theta_k) = \mathcal{N}(\theta_{k+1}; D\theta_k, \Xi)$, which would give additive terms in the derivations below. Throughout, we assume a finite horizon with terminal time T and a quadratic cost function in state and control

$$\mathcal{L}(\mathbf{x}, \mathbf{u}) = \left[\sum_{k=0}^{T-1} (x_k - r_k)^\top W_k (x_k - r_k) + \sum_{k=0}^{T-1} u_k^\top U_k u_k \right],$$

where $\mathbf{r} = [r_0, \dots, r_{T-1}]$ is a target trajectory. W_k and U_k define state and control cost, they can be time-varying. The goal, in line with the standard in both optimal control and reinforcement learning, is to find the control sequence \mathbf{u} that, at each k , minimizes the *expected cost* to the horizon

$$J_k(\mathbf{u}_{k:T-1}; p(x_k)) = \mathbb{E}_{x_k} \left[(x_k - r_k)^\top W_k (x_k - r_k) + u_k^\top U_k u_k + J_{k+1}(\mathbf{u}_{k+1:T-1}; p(x_{k+1})) \right] | p(x_k), \quad (3)$$

where past measurements $\mathbf{y}_{1:k}$, controls $\mathbf{u}_{1:k-1}$ and prior information $p(x_0)$ are incorporated into the belief $p(x_k)$, relative to which the expectation is calculated. Effectively, $p(x_k)$ serves as a bounded rationality approximation to the true information state. Since the equation above is recursive, the final element of the cost has to be defined differently, as

$$J_T(p(x_T)) = \mathbb{E}_{x_T} \left[(x_T - r_T)^\top W_T (x_T - r_T) \right] | p(x_T)$$

(that is, without control input and future cost). The optimal control sequence minimizing this cost will be denoted \mathbf{u}^* , with associated cost

$$J_k^*(p(x_k)) = \min_{\mathbf{u}_k} \mathbb{E}_{x_k} \left[(x_k - r_k)^\top W_k (x_k - r_k) + u_k^\top U_k u_k + J_{k+1}^*(p(x_{k+1})) \right] | p(x_k). \quad (4)$$

This recursive formulation, if written out, amounts to alternating minimization and expectation steps. As u_k influences x_{k+1} and y_{k+1} , it enters the latter expectation nonlinearly. Classic optimal control is the linear base case ($\phi(x) = x$) with known θ , where \mathbf{u}^* can be found by dynamic programming (Bellman, 1961; Bertsekas, 2005).

3. Bayesian RL and Dual Control

Feldbaum (1960–1961) coined the term *dual control* to describe the idea now also known as Bayesian reinforcement learning in the machine learning community: While adaptive control only considers past observations, dual control also takes future observations into account. This is necessary because all other ways to deal with uncertain parameters have substantial

drawbacks. Robust controllers, for example, sacrifice performance due to their conservative design; adaptive controllers based on *certainly equivalence* (where the uncertainty of the parameters is not taken into account but only their mean estimates) do not move exploration, so that all learning is purely passive. For most systems it is obvious that more excitation leads to better estimation, but also to worse control performance. Attempts at finding a compromise between exploration and exploitation are generally subsumed under the term “dual control” in the control literature. It can only be achieved by taking the future effect of current actions into account.

It has been shown that optimal dual control is practically unsolvable for most cases (Aoki, 1967), with a few examples where solutions were found for simple systems (e.g., Sternby, 1976). Instead, a large number of *approximate* formulations of the dual control problem were formulated in the decades since then. This includes the introduction of perturbation signals (e.g., Jacobs and Patchell, 1972), constrained optimization to limit the minimal control signal or the maximum variance, serial expansion of the loss function (e.g., Tse et al., 1973) or modifications of the loss function (e.g., Fliatow and Unbehauen, 2004). A comprehensive overview of dual control methods is given by Wittemark (1995). A historical side-effect of these numerous treatments is that the meaning of the term “dual control” has evolved over time, and is now applied both to the fundamental concept of optimal exploration, and to methods that only approximate this notion to varying degree. Our treatment below studies one such class of practical methods that aim to approximate the true dual control solution.

The central observation in Bayesian RL / dual control is that both the states x and the parameters θ are subject to uncertainty. While part of this uncertainty is caused by randomness, part by lack of knowledge, both can be captured in the same way by probability distributions. States and parameters can thus be subsumed in an *augmented state* (Feldbaum, 1960–1961; Duff, 2002; Pompart et al., 2006) $z_k^\top = (x_k^\top \ \theta_k^\top) \in \mathbb{R}^{(m+2)n}$. In this notation, the optimal exploration–exploitation trade-off—relative to the probabilistic priors defined above—can be written compactly as optimal control of the augmented system with a new observation model $p(y_k | z_k) = \mathcal{N}(y_k; \tilde{C}z_k, R)$ using $\tilde{C} = [C \ \mathbf{0}]$ and a cost analogous to Eq. (3).

Unfortunately, the dynamics of this new system are nonlinear, even if the original physical system is linear. This is because inference is always nonlinear and future states influence future parameter beliefs, and vice versa. A first problem, not unique to dual control, is thus that inference is not analytically tractable to use under the Gaussian assumptions above (Aoki, 1967). The standard remedy is to get approximations, most popularly the linearization of the extended Kalman filter (e.g., Särkkä, 2013). This gives a sequence of approximate Gaussian likelihood terms. But even so, incorporating these Gaussian likelihood terms into future dynamics is still intractable, because it involves expectations over rational polynomial functions, whose degree increases with the length of the prediction horizon. The following section provides an intuition for this complexity, but also the descriptive power of the augmented state space.

As an aside, we note that several authors (Kaippen, 2011; Hennig, 2011) have previously pointed out another possible construction of an augmented state: incorporating not the actual *value* of the parameters θ_k in the state, but the parameters μ_k, Σ_k of a Gaussian belief $p(\theta_k | \mu_k, \Sigma_k) = \mathcal{N}(\theta_k; \mu_k, \Sigma_k)$ over them. The advantage of this is that, if the state x_k is observed without noise, these belief parameters follow stochastic differential equations—more

precisely, Σ_k follows an ordinary (deterministic) differential equation, while μ_k follows a stochastic differential equation—and it can then be attempted to solve the control problem for these differential equations more directly.

While it can be a numerical advantage, this formulation of the augmented state also has some drawbacks, which is why we have here decided not to adopt it: First, the simplicity of the directly formalizable SDE vanishes in the POMDP settings, i.e. if the state is not observed without noise. If the state observations are corrupted, the exact belief state is not a Gaussian process, so that the parameters μ_k and Σ_k have no natural meaning. Approximate methods can be used to retain a Gaussian belief (and we will do so below), but the dynamics of μ_k , Σ_k are then intertwined with the chosen approximation (i.e. changing the approximation changes their dynamics), which causes additional complication. More generally speaking, it is not entirely natural to give differing treatment to the state x_k and parameters θ_k : Both state and parameters should thus be treated within the same framework; this also allows extending the framework to the case where also the parameters do follow an SDE.

3.1 A Toy Problem

To provide an intuition for sheer complexity of optimal dual control, consider the perhaps simplest possible example: the linear, scalar system

$$x_{k+1} = ax_k + bu_k + \xi_k, \quad (5)$$

with target $r_k = 0$ and noise-free observations ($R = 0$). If a and b are known, the optimal u_k to drive the current state x_k to zero in one step can be trivially verified to be

$$u_{k,\text{oracle}}^* = -\frac{abx_k}{U + b^2}.$$

Let now parameter b be uncertain, with current belief $p(b) = \mathcal{N}(b; \mu_k, \sigma_k^2)$ at time k . The naive option of simply replacing the parameter with the current mean estimate is known as *certainty equivalence (CE)* control in the dual control literature (e.g., Bar-Shalom and Tse, 1974). The resulting control law is

$$u_{k,\text{CE}}^* = -\frac{a\mu_k x_k}{U + \mu_k^2}.$$

It is used in many adaptive control settings in practice, but has substantial deficiencies: If the uncertainty is large, the mean is not a good estimate, and the CE controller might apply completely useless control signals. This often results in large overshoots at the beginning or after parameter changes.

A slightly more elaborate solution is to compute the expected cost $\mathbb{E}_b[x_{k+1}^2 + Uu_k^2 | \mu_k, \sigma_k^2]$ and then optimize for u_k . This gives *optimal feedback (OF)* or “cautious” control (Dreyfus, 1964)¹:

$$u_{k,\text{OF}}^* = -\frac{a\mu_k x_k}{U + \sigma_k^2 + \mu_k^2}. \quad (6)$$

1. Dreyfus used the term “open loop optimal feedback” for his approach, a term that is misleading to modern readers, because it is in fact a closed-loop algorithm.

This control law reduces control actions in cases of high parameter uncertainty. This mitigates the main drawback of the CE controller, but leads to another problem: Since the OF controller decreases control with rising uncertainty, it can entirely prevent learning. Consider the posterior on b after observing x_{k+1} , which is a closed-form Gaussian (because u_k is chosen by the controller and has no uncertainty):

$$p(b | \mu_{k+1}, \sigma_{k+1}^2) = \mathcal{N}\left(b; \frac{\sigma_k^2 u_k (bu_k + \xi_k) + \mu_k Q}{u_k^2 \sigma_k^2 + Q}, \frac{\sigma_k^2 Q}{u_k^2 \sigma_k^2 + Q}\right) \quad (7)$$

(b shows up in the fully observed $x_{k+1} = ax_k + bu_k + \xi_k$). The dual effect here is that the updated σ_{k+1}^2 depends on u_k . For large values of σ_k^2 , according to (6), $u_{k,\text{OF}} \rightarrow 0$, and the new uncertainty $\sigma_{k+1}^2 \rightarrow \sigma_k^2$. The system thus will never learn or act, even for large x_k . This is known as the “turn-off phenomenon” (Aoki, 1967; Bar-Shalom, 1981).

However, the derivation for OF control above amounts to minimizing Eq. (3) for the myopic controller, where the horizon is only a single step long ($T = 1$). Therefore, OF control is indeed optimal for this case. By the optimality principle (e.g., Bertsekas, 2005), this means that Eq. (6) is the optimal solution for the last step of every controller. But since it does not show any form of exploration or “probing” (Bar-Shalom and Tse, 1976), a myopic controller is not enough to show the dual properties.

In order to expose the dual features, the horizon has to be at least of length $T = 2$. Since the optimal controller follows Bellman’s equation, the solution proceeds backwards. The solution for the second control action u_1 is identical to the solution of the myopic controller (6); but after applying the first control action u_0 , the belief over the unknown parameter b needs an update according to Eq. (7), resulting in

$$u_1^* = -\left[U + \frac{\sigma_0^2 Q}{u_0^2 \sigma_0^2 + Q} + \left(\frac{\sigma_0^2 u_0 (bu_0 + \xi_0) + \mu_0 Q}{u_0^2 \sigma_0^2 + Q} \right)^2 \right]^{-1} \left[a \frac{\sigma_0^2 u_0 (bu_0 + \xi_0) + \mu_0 Q}{u_0^2 \sigma_0^2 + Q} x_1 \right]. \quad (8)$$

Inserting into Eq. (4) gives

$$\begin{aligned} J_0^*(x_0) &= \min_{u_0} \mathbb{E}_{x_0} \left[Wx_0^2 + Uu_0^2 + \min_{u_1} \mathbb{E}_{x_1} \left[Wx_1^2 + Uu_1^2 + \mathbb{E}_{x_2} [Wx_2^2] \right] \right] \\ &= \min_{u_0} \left[Wx_0^2 + Uu_0^2 + \mathbb{E}_{\xi_1, b} [Wx_1^2 + U(u_1^*)^2 + \mathbb{E}_{\xi_1, b} [W(x_1 + bu_1^* + \xi_1)^2 | \mu_1, \sigma_1] | \mu_0, \sigma_0] \right]. \end{aligned} \quad (9)$$

Since u_1^* from Eq. (8) is already a rational function of fourth order in b , and shows up quadratically in Eq. (9), the relevant expectations cannot be computed in closed form (Aoki, 1967). For this simple case though, it is possible to compute the optimal dual control by performing the expectation through sampling b, ξ_0, ξ_1 from the prior. Fig. 1 shows such samples of $\mathcal{L}(u_0)$ (in gray; one single sample highlighted in orange), and the empirical expectation $J(u_0)$ in dashed green. Each sample is a rational function of even leading order. In contrast to the CE cost, the dual cost is much narrower, leading to more cautious behavior of the dual controller. The average dual cost has its minima not at zero, but to either side of it, reflecting the optimal amount of exploration in this particular belief state.

While it is not out of the question that the Monte Carlo solution can remain feasible for larger horizons, we are not aware of successful solutions for continuous state spaces

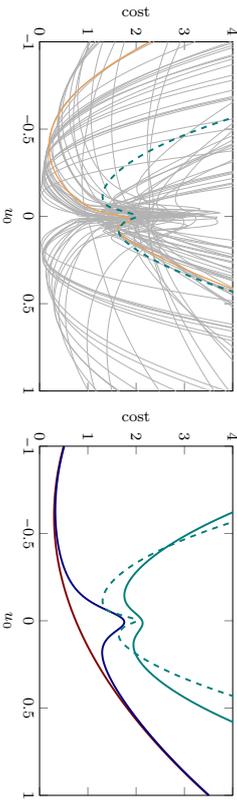


Figure 1: **Left:** Computing the $T = 2$ dual cost for the simple system of Eq. (5). Costs $\mathcal{L}(u_0)$ under optimal control on u_1 for sampled parameter b (thin gray; one sample highlighted, orange). Expected dual cost $J(u_0)$ under u_1^* (dashed green). The optimal u_0^* lies at the minimum of the dashed green line. **Right:** Comparison of sampling (dashed green; thin gray; samples) to three approximations: CE (red) and CE with Bayesian exploration bonus (blue). The solid green line is the approximate dual control constructed in Section 4. See also Sec. 6.1 for details.

(however, see Pompart et al., 2006, for a sampling solution to Bayesian reinforcement learning in discrete spaces, including notes on the considerable computational complexity of this approach). The next section describes a tractable *analytic* approximation that does not involve samples.

4. Approximate Dual Control for Linear Systems

In 1973, Tse et al. (1973) constructed theory and an algorithm (Tse and Bar-Shalom, 1973) for approximate dual (AD) control, based on the series expansion of the cost-to-go. This is related to differential dynamic programming for the control of nonlinear dynamic systems (Mayne, 1966). It separates into three conceptual steps (described in Sec. 4.1–4.3), which together yield what, from a contemporary perspective, amounts to a structured Gaussian approximation to Bayesian RL.

- ① Find an optimal trajectory for the deterministic part of the system under the mean model: the *nominal* trajectory under certainty equivalent control. For linear systems this is easy (see below), for nonlinear ones it poses a nontrivial, but feasible nonlinear model predictive control problem (Allgöwer et al., 1999; Diehl et al., 2009). It yields a nominal trajectory, relative to which the following step constructs a tractable quadratic expansion.
- ② Around the nominal trajectory, construct a local *quadratic expansion* that approximates the effects of future observations. Because the expansion is quadratic, an optimal control law relative to the deterministic system—the *perturbation control*—can be constructed by dynamic programming. Plugging this perturbation control into the residual dynamics of the approximate quadratic system gives an approximation for the cost-to-go. This step adds the cost of uncertainty to the deterministic control cost.

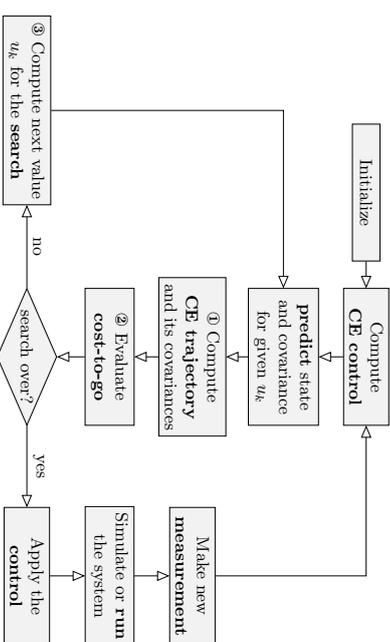


Figure 2: Flow-chart of the approximate dual control algorithm to show the overall structure. Adapted from Tse and Bar-Shalom (1973). The left cycle is the inner loop, performing the nonlinear optimization.

- ③ In the current time step k , perform the prediction for an arbitrary control input u_k (as opposed to the analytically computed control input for later steps). Optimize u_k numerically by repeated computation of steps ① and ② at varying u_k to minimize the approximate cost.

These three steps will be explained in detail in the subsequent sections. The interplay between the different parts of the algorithm is shown in Figure 2.

The abstract introductory work Tse et al. (1973) is relatively general, but the explicit formulation in Tse and Bar-Shalom (1973) only applies to linear systems. Since both works are difficult to parse for contemporary readers, the following sections thus first provide a short review, before we extend to more modern concepts. In this section, we follow the more transparent case of a linear system from Tse and Bar-Shalom (1973), i.e. $\phi(x) = x$ in Eq. (1). For the augmented state z , this still gives a nonlinear system, because θ and x interact multiplicatively

$$z_{k+1} = \begin{pmatrix} x_{k+1} \\ \theta_{k+1} \end{pmatrix} = \begin{pmatrix} A(\theta_k) & 0 \\ 0 & \mathbf{I} \end{pmatrix} z_k + \begin{pmatrix} B(\theta_k) \\ 0 \end{pmatrix} u_k + \begin{pmatrix} \xi_k \\ 0 \end{pmatrix} =: \tilde{f}(z_k, u_k). \quad (10)$$

The parameters θ are assumed to be deterministic, but not known to the controller. This uncertainty is captured by the distribution $p(\theta)$ representing the lack of knowledge.

4.1 Certainty Equivalent Control Gives a Nominal Reference Trajectory

The certainty equivalent model is built on the assumption that the uncertain θ coincide with their most likely value, the mean $\bar{\theta}$ of $p(\theta)$, and that the system propagates deterministically without noise. This means that the nominal parameters $\bar{\theta}$ are the current mean values $\hat{\theta}$,

which decouples θ entirely from x in Eq. (10), and the optimal control for the finite horizon problem can be computed by dynamic programming (DP) (Aoki, 1967), yielding an optimal linear control law

$$\bar{u}_j^* = -(\bar{B}^\top \bar{K}_{j+1} \bar{B} + U_j)^{-1} \bar{B}^\top [\bar{K}_{j+1} \bar{A} \bar{x}_j + \bar{p}_{j+1}],$$

where we have momentarily simplified notation to $\bar{A} = A(\bar{\theta}_j)$, $\bar{B} = B(\bar{\theta}_j)$, $\forall j$, because the $\bar{\theta}_j$ are constant. The \bar{K}_j and \bar{p}_j for $j = k+1, \dots, T$ are defined and computed recursively as

$$\begin{aligned} \bar{K}_j &= \bar{A}^\top (\bar{K}_{j+1} - \bar{K}_{j+1} \bar{B} (\bar{B}^\top \bar{K}_{j+1} \bar{B} + U_j)^{-1} \bar{B}^\top \bar{K}_{j+1}) \bar{A} + W_j & \bar{K}_T &= W_T \\ \bar{p}_j &= \bar{A}^\top (\bar{p}_{j+1} - \bar{K}_{j+1} \bar{B} (\bar{B}^\top \bar{K}_{j+1} \bar{B} + U_j)^{-1} \bar{B}^\top \bar{K}_{j+1}) \bar{p}_{j+1} - W_j r_j & \bar{p}_T &= -W_T r_T, \end{aligned}$$

where r is the reference trajectory to be followed. This CE controller gives the *nominal trajectory* of inputs $\bar{u}_{k:T-1}$ and states $\bar{x}_{k:T}$, from the current time k to the horizon T . The true future trajectory is subject to stochasticity and uncertainty, but the deterministic nominal trajectory \bar{x} , with its optimal control \bar{u}^* and associated nominal cost $\bar{J}_k^* = \mathcal{L}(\bar{x}_{k:T}, \bar{u}_{k:T}^*)$ provides a base, relative to which an approximation will be constructed.

4.2 Quadratic Expansion Around the Nominal Defines Cost of Uncertainty

The central idea of AD control is to project the nonlinear objective $J_k(\mathbf{u}_{k:T-1}, p(x_k))$ of Eq. (3) into a quadratic, by locally linearizing around the nominal trajectory \mathbf{x} and maintaining a joint Gaussian belief.

To do so, we introduce small perturbations around nominal cost, states, and control: $\Delta J_j = J_j - \bar{J}_j$, $\Delta z_j = z_j - \bar{z}_j$, and $\Delta u_j = u_j - \bar{u}_j$. These perturbations arise from both the stochasticity of the state and the parameter uncertainty. Note that a change in the state results in a change of the control signal, because the optimal control signal in each step depends on the state. Even though the origin of the uncertainties is different (Δx arises from stochasticity and $\Delta \theta$ from the lack of knowledge), both can be modeled in a joint probability distribution.

Approximate Gaussian filtering ensures that beliefs over Δz remain Gaussian:

$$p(\Delta z_j) = \mathcal{N} \left[\begin{pmatrix} \Delta x_j \\ \Delta \theta_j \end{pmatrix}; \begin{pmatrix} \Sigma_{xx} & \Sigma_{x\theta} \\ 0 & \Sigma_{\theta\theta} \end{pmatrix} \right].$$

Note that shifting the mean to the nominal trajectory does not change the uncertainty. Note further that the expected perturbation in the parameters is nil. This is because the parameters are assumed to be deterministic and are not affected by any state or input.

Calculating the Gaussian filtering updates is in principle not possible for future measurements, since it violates the causality principle (Glad and Ljung, 2000). Nonetheless, it is possible to use the *expected* measurements to simulate the effects of the future measurements on the uncertainty, since these effects are deterministic. This is sometimes referred to as preposterior analysis (Raiffa and Schlaifer, 1961).

To second order around the nominal trajectory, the cost is approximated by

$$J_k(\mathbf{u}_{k:T-1}, p(x_k)) = \bar{J}_k^* + \Delta J_k \approx \bar{J}_k^* + \Delta \bar{J}_k,$$

where \bar{J}_k^* is the optimal cost for the nominal system and $\Delta \bar{J}_k$ is the approximate additional cost from the perturbation:

$$\Delta \bar{J}_k := \mathbb{E}_{\mathbf{x}_{k:T}} \left[\sum_{j=k}^T \left\{ (\hat{x}_j - r_j)^\top W_j \Delta x_j + \frac{1}{2} \Delta \hat{x}_j^\top W_j \Delta x_j \right\} + \sum_{j=k}^{T-1} \left\{ \bar{u}_j^\top U_j \Delta u_j + \frac{1}{2} \Delta \bar{u}_j^\top U_j \Delta u_j \right\} \right]. \quad (12)$$

Although the uncertain parameters θ do not show up explicitly in the above equation, this step captures dual effects: The uncertainty of the trajectory $\Delta \mathbf{x}$ depends on θ via the dynamics. Higher uncertainty over θ at time $j-1$ causes higher predictive uncertainty over Δx_j (for each j), and thus increases the expectation of the quadratic term $\Delta \bar{u}_j^\top W_j \Delta x_j$. Control that decreases uncertainty in θ can lower this approximate cost, modeling the benefit of exploration. For the same reason, Eq. (12) is in fact still not a quadratic function and has no closed form solution. To make it tractable, Tse and Bar-Shalom (1973) make the ansatz that all terms in the expectation of Eq. (12) can be written as $g_j + p_j^\top \Delta z_j + 1/2 \Delta z_j^\top K_j \Delta z_j$. This amounts to applying dynamic programming on the perturbed system. Expectations over the cost under Gaussian beliefs on Δz can then be computed analytically. Because all $\Delta \theta$ have zero mean, linear terms in these quantities vanish in the expectation. This allows analytic minimization of the approximate optimal cost for each time step

$$\begin{aligned} \Delta \bar{J}_j^*(p(x_j)) &= \min_{\Delta u_j} \left\{ (x_j - r_j)^\top W_j \Delta \hat{x}_{j|j} + \frac{1}{2} \Delta \hat{x}_{j|j}^\top W_j \Delta \hat{x}_{j|j} + u_j^\top U \Delta u_j + \frac{1}{2} \Delta \bar{u}_j^\top U \Delta u_j \right. \\ &\quad \left. + \frac{1}{2} \text{tr} \left[W_j \Sigma_{j-1|j}^{xx} \right] + \mathbb{E}_{\Delta \bar{x}_{j+1}} \left[\Delta \bar{J}_{j+1}^*(\mathbf{y}_{1:j+1}) \mid p(x_j) \right] \right\}, \quad (13) \end{aligned}$$

which is feasible given an explicit description of the Gaussian filtering update. It is important to note that, assuming extended Kalman filtering, the update to the mean from *expected* future observations y_{j+1} is nil. This is because we expect to see measurements consistent with the current mean estimate. Nonetheless, the (co-)variance changes depending on the control input u_j , which is the dual effect.

Following the dynamic programming equations for the perturbed problem, including the additional cost from uncertainty, the resulting cost amounts to (Tse et al., 1973)

$$\begin{aligned} \Delta \bar{J}_k^*(p(x_k)) &= \hat{g}_{k+1} + \bar{p}_{k+1}^\top \Delta \hat{z}_k + \frac{1}{2} \Delta \hat{z}_k^\top \bar{K}_{k+1} \Delta \hat{z}_k \\ &\quad + \frac{1}{2} \text{tr} \left\{ W_T \Sigma_{T|T}^{xx} + \sum_{j=k}^{T-1} \left[W_j \Sigma_{j|j}^{xx} + (\Sigma_{j+1|j} - \Sigma_{j+1|j+1}) \bar{K}_{j+1} \right] \right\} \end{aligned}$$

(where we have neglected second-order effects of the dynamics). Recalling that $\Delta \hat{z} = 0$ and dropping the constant part, the dual cost can be approximated to be

$$J_k^d = \frac{1}{2} \text{tr} \left\{ W_T \Sigma_{T|T}^{xx} + \sum_{j=k}^{T-1} \left[W_j \Sigma_{j|j}^{xx} + (\Sigma_{j+1|j} - \Sigma_{j+1|j+1}) \bar{K}_{j+1} \right] \right\} \quad (= \Delta \bar{J}_k^* - \text{const})$$

where the recursive equation

$$\bar{K}_j = \bar{A}^\top (\bar{K}_{j+1} - \bar{K}_{j+1} \bar{B} (\bar{B}^\top \bar{K}_{j+1} \bar{B} + U_j)^{-1} \bar{B}^\top \bar{K}_{j+1}) \bar{A} + \bar{W}_j \quad \bar{K}_T = \bar{W}_T$$

is defined for the augmented system (10), with $\tilde{A} = \frac{\partial \tilde{f}}{\partial \tilde{x}}$, $\tilde{B} = \frac{\partial \tilde{f}}{\partial \tilde{u}}$ and $\tilde{W}_j = \text{blkdiag}(W_j, \mathbf{0})$. The approximation to the overall cost is then $J_k^* + J_k^d$, which is used in the subsequent optimization procedure.

4.3 Optimization of the Current Control Input Gives Approximate Dual Control

The last step ③ amounts to the outer loop of the overall algorithm. A gradient-free black-box optimization algorithm is used to find the minimum of the dual cost function. In every step, this algorithm proposes a control input u_k for which the dual cost is evaluated.

Depending on u_k , approximate filtering is carried out to the horizon. The perturbation control is plugged into Eq. (13) to give an analytic, recursive definition for K_j , and an approximation for the dual cost J_k^d as a function of the current control input u_k .

Nonlinear optimization—through repetitions of steps ① and ② for proposed locations u_k —then yields an approximation to the optimal dual control u_k^* . Conceptually the simplest part of the algorithm, this outer loop dominates computational cost, because for every location u_k the whole machinery of ① and ② has to be evaluated.

5. Extension to Contemporary Machine Learning Models

The preceding section reviewed the treatment of dual control in linear dynamical systems from Tse and Bar-Shalom (1973). In this section, we extend the approach to inference on, and dual control of, the dynamics of *nonlinear* dynamical systems. This extension is guided by the desire to use a number of popular, standard regression frameworks in machine learning: Parametric general least-squares regression, nonparametric Gaussian process regression, and feedforward neural networks (including the base case of logistic regression).

5.1 Parametric Nonlinear Systems

We begin with the generalized linear model mentioned in Eq. (1). The nonlinear features ϕ can in principle be any function (popular choices include sines and cosines, radial basis functions, sigmoids, polynomials and others), with the caveat that their structure crucially influences the properties of the model. From a modeling perspective, this approach is quite standard for machine learning. However, the dynamical learning setting requires a few adaptations: First, to allow the modeling of higher-order dynamical systems, the original states must be included. This gives features of the form $\phi(x)^T = (x^T \ \varphi(x)^T)$, consisting of the linear representation, augmented by general features φ .

The next challenge is that the optimal control for nonlinear dynamical systems cannot be optimized in closed form using dynamic programming, not even for the deterministic nominal system. Instead, we find the nominal reference trajectory using nonlinear model predictive control (Allgöwer et al., 1999; Diehl et al., 2009). In our case, we begin with a dynamic programming on a locally linearized system, then optimize nonlinearly with a numerical method across the trajectory. This adds computational cost, and requires some care to achieve stable optimization performance for specific system setups.

Filtering from observations is also more involved in the case of nonlinear dynamics. In the experiments reported below, we stayed within the extended Kalman filtering framework

to retain Gaussian beliefs over the states and parameters. Extensions of this approach to more elaborate filtering methods are an interesting direction for future work. This includes relatively standard options like unscented Kalman filtering (Uhlmann, 1995), but also more recent developments in machine learning and probabilistic control, such as analytic moment propagation where the features φ allow this (e.g., Deisenroth and Rasmussen, 2011).

The final problem is the generalization of the derivations from the preceding sections to the nonlinear dynamics. We take a relatively simplistic approach, which nevertheless turns out to work well. A linearization gives locally linear dynamics whose structure closely matches Eq. (10):

$$\begin{aligned} z_{k+1} &= \begin{pmatrix} \bar{x}_{k+1} + \Delta x_{k+1} \\ \bar{\theta}_{k+1} + \Delta \theta_{k+1} \end{pmatrix} = \begin{pmatrix} A(\theta_k) & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} \phi(\bar{x}_k + \Delta x_k) \\ \bar{\theta}_k + \Delta \theta_k \end{pmatrix} + \begin{pmatrix} B(\theta_k) \\ 0 \end{pmatrix} u_k + \begin{pmatrix} \xi_k \\ 0 \end{pmatrix} \\ &\approx \begin{pmatrix} A(\bar{\theta}_k) & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} \phi(\bar{x}_k) \\ \bar{\theta}_k \end{pmatrix} + \begin{pmatrix} B(\bar{\theta}_k) \\ 0 \end{pmatrix} u_k + \begin{pmatrix} \xi_k \\ 0 \end{pmatrix} \\ &\quad + \begin{pmatrix} A(\bar{\theta}_k) \frac{\partial}{\partial \bar{x}_k} \phi(\bar{x}_k) & \frac{\partial}{\partial \bar{\theta}_k} (A(\bar{\theta}_k) \phi(\bar{x}_k) + B(\bar{\theta}_k) u_k) \\ 0 & I \end{pmatrix} \begin{pmatrix} \Delta x_k \\ \Delta \theta_k \end{pmatrix}. \end{aligned}$$

This essentially amounts to extended Kalman filtering on the augmented state. Using this linearization, the approximation described in Sec. 4 can be applied analogously.

5.2 Nonparametric Gaussian Process Dynamics Models

The above treatment of parametric linear models makes it comparably easy to extend the description from finitely many feature functions to an infinite-dimensional feature space defining a Gaussian process (GP) dynamics model: Assume that the true dynamics function f is a draw from a Gaussian process prior $p(f) = \mathcal{GP}(f; m, \bar{\kappa})$ with prior mean function $m: \mathbb{R}^n \rightarrow \mathbb{R}^n$, and prior covariance function (kernel) $\bar{\kappa}: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n \times \mathbb{R}^n$. This is using the widely used notion of “multi-output regression” (Rasmussen and Williams, 2006, § 9.1), i.e. formulating the covariance as

$$\text{cov}(f_k(x), f_j(x')) = \bar{\kappa}_{kj}(x, x').$$

To simplify the treatment, we will assume that the covariance factors between inputs and outputs, i.e. $\bar{\kappa}_{ij}(x, x') = V_{ij} \kappa(x, x')$ with a univariate kernel $\kappa: \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ and a positive semi-definite matrix $V \in \mathbb{R}^{m \times m}$ of output covariances. By Mercer’s theorem (e.g., König, 1986; Rasmussen and Williams, 2006), the kernel can be decomposed into a converging series over eigenfunctions $\phi(x)$, as

$$\kappa(x, x') = \sum_{\ell=1}^{\infty} \lambda_{\ell} \phi_{\ell}(x) \phi_{\ell}^*(x'), \quad (14)$$

where $\phi_{\ell}: \mathbb{R}^n \rightarrow \mathbb{R}$ are functions that are orthonormal relative to some measure μ over \mathbb{R}^n (the precise choice of which is irrelevant for the time being), with the property

$$\int \kappa(x, x') \phi_{\ell}(x') d\mu(x') = \lambda_{\ell} \phi_{\ell}(x).$$

Precisely in this sense, Gaussian process regression can be written as “infinite-dimensional” Bayesian linear regression. We will use the suggestive, and somewhat abusive notation $f_k(x_k) = L\Omega_k\Phi(x_k)$ for this generative model, defined as

$$f_k^i(x_k) = \sum_{j=1}^T L_{ij} \sum_{\ell=1}^{\infty} \Omega_k^{\ell} \phi_{\ell}(x_k) \quad (15)$$

where L is a matrix satisfying $LL^{\top} = V$ (e.g., the Cholesky decomposition), and the elements of Ω are draws from the “white” Gaussian process $\Omega_k^{\ell} \sim \mathcal{N}(0, \lambda_{\ell})$. Because of Mercer’s theorem above, Eq. (15) exists in μ^2 expectation, and is well-defined in this sense. This notation allows writing Eq. (2) as a nonparametric prior with mean θ_0 and covariance $\Sigma_k^{\theta_0} = V \otimes (\Phi\Lambda\Phi^{\top})$ where Λ is an infinite diagonal matrix with diagonal elements $\Lambda_{\ell\ell} = \lambda_{\ell}$ (the matrix multiplication $\Phi\Lambda\Phi^{\top}$ is here defined as in Eq. (15)).

Using this notation, a tedious but straightforward linear algebra derivation (see Appendix A) shows that the posterior over $z^{\top} = (x^{\top} \ \theta^{\top})$ after a number k of EKF-linearized Gaussian observations is a tractable Gaussian process, for which the Gram matrix

$$\mathcal{G} = \mathcal{P} + \mathcal{Q} + \mathcal{K} + \mathcal{F}^{-1}\mathcal{R}\mathcal{F}^{-\top}$$

consists of the parts

$$\mathcal{P} = \begin{bmatrix} R_0 & 0 \\ 0 & 0 \end{bmatrix} \quad \mathcal{Q} = (\mathcal{Q} \otimes I) \quad \mathcal{K} = \kappa(\mathbf{y}_{1:m}, \mathbf{y}_{1:m}) \quad \mathcal{R} = (R \otimes I)$$

of appropriate size, depending on the current time k . The multi-step state transition matrix

$$\mathcal{F} = \begin{bmatrix} I & 0 & 0 & \dots & 0 \\ A_1 & I & 0 & \dots & 0 \\ A_2 A_1 & A_2 & I & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_m \dots A_1 & A_m \dots A_2 & \dots & \dots & I \end{bmatrix} \quad \text{with} \quad \mathcal{F}^{-1} = \begin{bmatrix} I & 0 & 0 & \dots & 0 \\ -A_1 & I & 0 & \dots & 0 \\ 0 & -A_2 & I & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & -A_m & I \end{bmatrix}$$

is needed to account for the effect of the measurement noise R over time. The A -matrices are the Jacobians $\nabla_x f(x)_{|x_k}$.

The posterior mean now evaluates to

$$\begin{bmatrix} \hat{x}_k \\ \hat{\theta}_k \end{bmatrix} = \begin{bmatrix} \hat{x}_{k-1} \\ 0 \end{bmatrix} + \begin{bmatrix} \Phi(\hat{x}_{k-1})\Lambda\Phi(\mathbf{y}_{1:k-1})^{\top} \\ \Lambda\Phi(\mathbf{y}_{1:k-1})^{\top} \end{bmatrix} \mathcal{G}^{-1} [\Phi(\mathbf{y}_{1:k-1})\Lambda\Phi(\hat{x}_{k-1})^{\top} \ \Phi(\mathbf{y}_{1:k-1})\Lambda] \quad (16)$$

and the posterior covariance is comprised of

$$\bar{\Sigma}_k^{xx} = A_k \Sigma_k^{xx} A_k^{\top} + Q + \Phi_k \Sigma_k^{\theta_0} A_k^{\top} + A_k \Sigma_k^{\theta_0} \Phi_k^{\top} + \Phi_k \Sigma_k^{\theta_0} \Phi_k^{\top} \quad (17a)$$

$$\Sigma_k^{xx} = \bar{\Sigma}_{k-1}^{xx} - \bar{\Sigma}_{k-1}^{xx} [\bar{\Sigma}_{k-1}^{xx} + R]^{-1} \bar{\Sigma}_{k-1}^{xx} \quad (17b)$$

$$\Sigma_k^{\theta_0} = \mathcal{F}_k \Phi_k \Phi_k^{\top} \Lambda - \mathcal{F}_k; [\mathcal{P} + \mathcal{K} + \mathcal{Q}] \mathcal{G}^{-1} \Phi_k \Lambda \quad (17c)$$

$$\Sigma_k^{\theta_0} = (\Sigma_k^{\theta_0})^{\top} \quad (17d)$$

$$\Sigma_k^{\theta_0} = \Lambda - \Lambda \Phi(\mathbf{y})^{\top} \mathcal{G}^{-1} \Phi(\mathbf{y}) \Lambda. \quad (17e)$$

This formulation, together with the expositions in the preceding sections, defines a nonparametric dual control algorithm for Gaussian process priors. It is important to stress that this posterior is indeed “tractable” in so far as it depends only on a Gram matrix of size $nT \times nT$, and the posterior over any $f(x)$ can be computed in time $\mathcal{O}((nT)^3)$, despite the infinite-dimensional state space.

5.2.1 AN APPROXIMATION OF CONSTANT COST

In practical control applications, continuously rising inference cost is rarely acceptable. It is thus necessary to project the GP belief onto a finite representation, replacing the infinite sum in Eq. (14) with a finite one, to bound the computational cost of the matrix inversion in Eqs. (16), (17c) and (17e). We do so by projecting into a pre-defined finite basis of functions drawn from the eigen-spectrum of the kernel with respect to the Lebesgue measure. This approach has been recently popular elsewhere in regression (Rahimi and Recht, 2008). For readers unaware of this line of work, here is a short, self-contained introduction:

By Bochner’s theorem (e.g., Stein, 1999; Rasmussen and Williams, 2006), the covariance function $k(r)$ (with $r = |x - x'|$) of a stationary μ^2 continuous random process can be represented as the Fourier transform of a positive finite measure and, if that measure has a density $S(s)$, as the Fourier dual of S :

$$\kappa(r) = \int S(s) e^{2\pi i s r} ds,$$

This means that the eigenfunctions of the kernel are trigonometric functions, and stationary covariance functions, like the commonly used square exponential kernel

$$\kappa_{\text{SE}}(x, x') = \exp\left(-\frac{(x - x')^2}{2\lambda^2}\right),$$

can be approximated by sine and cosine basis functions as

$$\kappa(x, x') \approx \tilde{\kappa}(x, x') = \sqrt{\frac{2}{F}} \sum_{i=1}^{F/2} \sin(\omega_{2i-1}|x - x'|) + \cos(\omega_{2i}|x - x'|),$$

where the frequencies ω_i of the feature functions is sampled from the power spectrum of the process. An example of such kernel approximation is shown in Fig. 3. With increasing number of features, the approximation can be chosen as closely to the true covariance function as needed, while keeping the number of features in a range that is still feasible within the time constraints of the control algorithm.

5.3 Dual Control of Feedforward Neural Networks

Another extension of the parametric linear models of Section 5.1 is to allow for a nonlinear parametrization of the dynamics function:

$$f(x; \theta) = \sum_{\ell} \theta_{\ell}^{\text{lin}} \phi_{\ell}(x; \theta_{\ell}^{\text{nonlin}}).$$

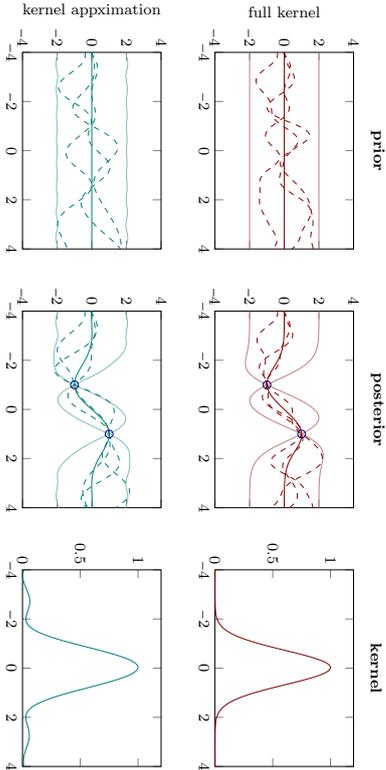


Figure 3: Prior (left), posterior (middle) and kernel function (right) of both the full kernel function (top row) and the approximate kernel (bottom row). The thick lines represent the mean and the thin lines show two standard deviations. The dashed lines are samples from the shown distributions.

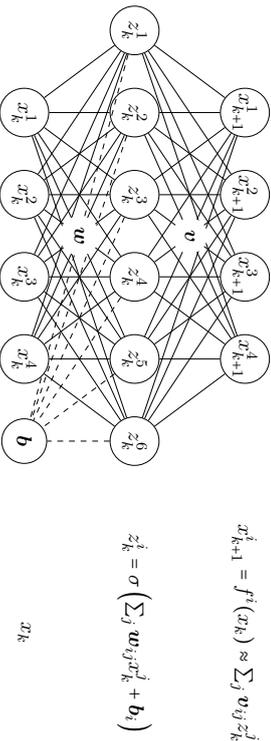


Figure 4: A two-layer feedforward neural network. Sketch to illustrate the structure of Eq. (18).

A particularly interesting example of this structure are multilayer perceptrons. Consider a two-layer network with logistic link function

$$f(x) = \sum_i v_i \sigma(\mathbf{w}_i x + \mathbf{b}_i), \quad (18)$$

where \mathbf{v} are the weights from the latent to the output layer, \mathbf{w} are the weights from input to hidden units, and \mathbf{b} are the biases of the hidden units (see Fig. 4).

Neural networks are used in control quite regularly, see e.g., Nguyen and Widrow (1990). Instead of using backpropagation and stochastic gradient descent as in most applications of neural networks (Rumelhart et al., 1986; Robbins and Monro, 1951), the EKF inference procedure can be used to train the weights as well (Singhal and Wu, 1989). This is possible because the EKF linearization can also be applied for the nonlinear link function, e.g., the logistic function. Speaking in terms of feature functions, not only the weight of each feature but also the shape (steepness) can be inferred. A limiting factor for this inference naturally is the number of data points: the more features and parameters are introduced, the more data points are necessary to learn.

Using the state augmentation $z^T = (x^T \ v^T \ w^T)$, and linearizing w.r.t. all parameters in each step, the EKF inference on the neural network parameters allows us not only to apply relatively cheap inference on them, but also to use the dual control framework to plan control signals, accounting for the effect of future observations and the subsequent change in the belief. This means the adaptive dual controller described in Sec. 4 can identify those parts of the neural net that are relevant for applying optimal control to the problem at hand. In Sec. 6.3, we show an experiment with these properties.

6. Experiments

A series of experiments on single-episode tasks with continuous state space highlights qualitative differences between the adaptive dual (AD) controller and three other controllers: An oracle controller with access to the true parameters, which provides an unattainable lower bound (LB) on the achievable performance, a certainty equivalent (CE) controller as described in Sec. 4.1, and a controller minimizing the sum of GE cost and the Bayesian exploration bonus (BEB) (see Köler and Ng, 2009):

$$l_{\text{BEB}} = \tau \left[\text{sqrt}(\text{diag}(\Sigma^{t\theta})) \right]^T \left[\text{sqrt}(\text{diag}(\Sigma^{t\theta})) \right]$$

(τ is a scalar exploration weight). The additional cost term l_{BEB} is evaluated for the predicted parameter covariance where the prediction time is chosen according to the order of the system such that the effect of the current control signal shows up in the belief over the parameters. This type of controller is sometimes also called dual control, while being referred to as *explicit dual control*, where the dual features are obtained by a modified cost function (Filiatov and Ushbaevan, 2000).

Every experiment was repeated 50 times with different random seeds, which were shared across controllers for comparability. All systems presented below are very simple setups. Their primary point is to show qualitative differences of the controllers' behavior. The experiments were done with different approximations from the preceding section to show experimental feasibility for each of them.

The feature set used for a specific application is part of the prior assumptions for that application. Large uncertainty requires flexible models (which take longer to converge, and require more exploration). Feature selection is important, but since it is independent of the dual control framework itself and a broad topic on its own, it is beyond the scope of this paper. In the following experiments, different feature sets are used both as examples for the flexibility of the framework, but also to model different structural knowledge about the problems at hand.

6.1 On a Simple Scalar System, AD Control Matches Exact Dual Control Well

For the noise-free linear system of Sec. 3.1, ($a = 1$ (known), $b = 2$, $p(b) = \mathcal{N}(b; 1, 10)$, $Q = 10^{-1}$, $R = 0$, $W = 1$, $\Lambda = 1$, $T = 2$), Fig. 1, right, compares the cost functions of the various controllers and the approximately exact sampling solution (which is only available for this very simple setup). All cost functions are shifted by an irrelevant constant. The CE cost is quadratic and indifferent about zero. The BEB ($\tau = 0.1$) gives additional structure near zero that encourages learning. While qualitatively similar to the dual cost, its global minimum is almost at the same location as that of CE. The dual control approximates the sampling solution much closer.

6.2 Faced with Time-Varying State Cost, AD Control Holds Off Exploration Until Suitable

A cart on a rail is a simple example for a dynamical system. Combined with a nonlinearly varying slope, a simple but nonlinear system can be constructed. The dynamics, prior beliefs, and true values for the parameters are chosen to be

$$x_{k+1} = \begin{bmatrix} 1 & 0.4 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 & 0 \\ \theta^1 & \theta^2 \end{bmatrix} \begin{bmatrix} \varphi^1(x_k^1) \\ \varphi^2(x_k^1) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_k \quad \theta \sim \mathcal{N} \left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \quad \theta_{\text{true}} = \begin{bmatrix} 0.8 \\ 0.4 \end{bmatrix},$$

$$\varphi^1(x) = -\frac{1}{1 + e^{-(x+5)}} \quad \varphi^2(x) = \frac{1}{1 + e^{-(x-5)}}, \quad (19)$$

where superscripts denote vector elements. The nonlinear functions φ are shifted logistic functions of the form

and disturbance/noise is chosen to be $R = Q = 10^{-2}I$. We use this setup as a testbed for a time-structured exploration problem. The actual system and its dynamics are relatively irrelevant here, as we will focus on a complication caused by the cost function: The reference to be tracked is

$$\mathbf{r}_{0:11} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{r}_{12:14} = \begin{bmatrix} 10 \\ 0 \end{bmatrix} \quad \mathbf{r}_{15} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{r}_{16:18} = \begin{bmatrix} -10 \\ 0 \end{bmatrix} \quad \mathbf{r}_{19:20} = \begin{bmatrix} 0 \\ 0 \end{bmatrix};$$

it is also shown in each plot of Fig. 5 as dashed orange line. The state weighting is time-dependent

$$\mathbf{W}_{0:5} = \begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix} \quad \mathbf{W}_{5:10} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \mathbf{W}_{11:20} = \begin{bmatrix} 100 & 0 \\ 0 & 0 \end{bmatrix},$$

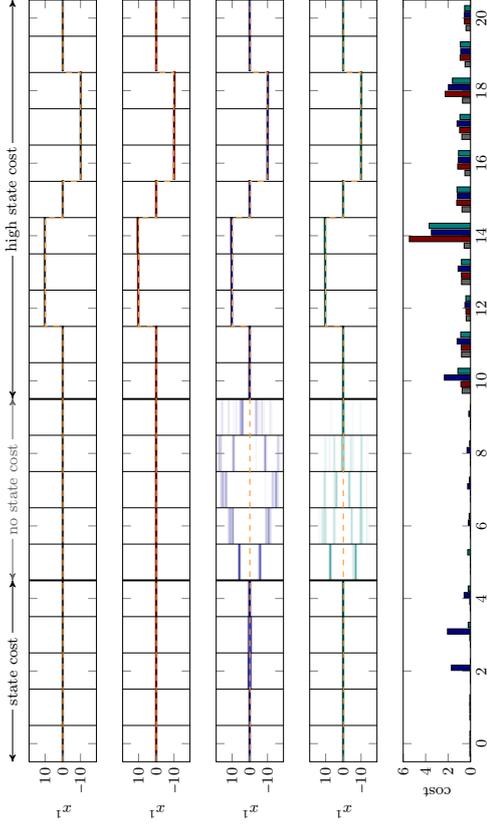


Figure 5: **Top four:** Density estimate for 50 trajectories (first state). From top to bottom: optimal oracle control (gray), certainty equivalent control (red), CE with Bayesian exploration bonus (blue), approximate dual control (green). Reference trajectory in dashed orange. **Bottom:** The mean cost per time step is shown in the bottom plot, with colors matching the controllers noted above.

and control cost is relatively low: $\Lambda = 10^{-3}$. The task, thus, is to first keep the cart fixed in the starting position to high precision, for the first 4 time steps. This is followed by a “loose” period between time steps 5 and 10. Then, the cart has to be moved to one side, back to the center, to the other side, and back again, all at high cost. A good exploration strategy in this setting is to act cautiously for the first 5 time steps, then aggressively explore in the “loose” phase, to finally be able to control the motion with high precision.

The inference model is a GP with approximated SE kernel, as described in Sec. 5.2.1. We use 30 alternating sine and cosine features that are distributed according to the power spectrum of the full SE kernel. Since the true nonlinearity of Eq. (19) is not of this form, the approximation is out of model and the lower bound controller only represents a perfectly learned, but still not exact, model.

Fig. 5 shows a density estimated from 50 state trajectories for the four different controllers. The lower bound controller (top) controls precisely at times of high cost, and does nothing for times with zero cost, controlling perfectly up to the measurement and state disturbances. The certainty equivalent controller (second from top) never explores actively, it only learns “accidentally” from observations arising during the run. Since the initial trajectory requires little action, it is left with a bad model when the reference starts to move at time step 12. The exploration bonus controller (second from bottom) continuously explores, because it

has no way of knowing about the “loose” phase ahead. Of course, this strategy incurs a higher cost initially. The dual controller (bottom) efficiently holds off exploration until it reaches the “loose” phase, where it explores aggressively.

6.3 AD Control Distinguishes Necessary and Unnecessary Parameter Exploration

The system including nonlinearities for this experiment is the same as before, although with noise parameters $R = Q = 10^{-3}I$. The reference trajectory and state weighting are much simpler, though:

$$r_{0:11} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad r_{12:18} = \begin{bmatrix} 10 \\ 0 \end{bmatrix} \quad r_{18:20} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

with the time-dependent weighting

$$W_{0:10} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad W_{11:20} = \begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix},$$

allowing for identification in the beginning, while penalizing deviations of the first state in later time steps.

Important to note here is that the reference trajectory only passes areas of the state space where φ^1 is strong, and φ^2 is negligible. Good exploration thus will ignore θ^2 , but this can only be found through reasoning about future trajectories.

In this experiment, the learned model is of the neural network form described in Sec. 5.3. We use 4 logistic features (see Eq. (18)) with two free parameters each (w_i and v_i) and equally spaced b_i between -5 and 5 , the locations of the true nonlinear features. This means it is possible to learn the perfect model in this case.

Fig. 6 shows a density estimated from 50 state trajectories for the four different controllers. Because of symmetry in the cost function and feature functions, BEB (with $\tau = 1$) cannot “decide” between the relevant θ^1 and the irrelevant θ^2 , choosing the exploration direction stochastically. It thus sometimes reduces the uncertainty on θ^2 , which does not help the subsequent control. The AD controller ignores θ^2 completely and only identifies θ^1 in early phases, leading to good control performance.

6.4 AD Control Maintains Only Useful Knowledge

The last experiment is again similar to Sec. 6.2, but uses a different set of nonlinear functions, namely shifted Gaussian functions (a.k.a. radial basis functions)

$$\varphi^1(x) = e^{-\frac{(x-2)^2}{2}} \quad \varphi^2(x) = e^{-\frac{(x+2)^2}{2}} \quad \theta_{\text{true}} = \begin{bmatrix} 1.0 \\ 0.8 \end{bmatrix}.$$

For this experiment, the model is learned with parametric linear regression, according to Sec. 5.1. The fundamental difference to the other experimental setups is that the model now assumes *parameter drift*. This results in growing uncertainty for the parameters over time. (The true parameters are kept constant for simplicity.)

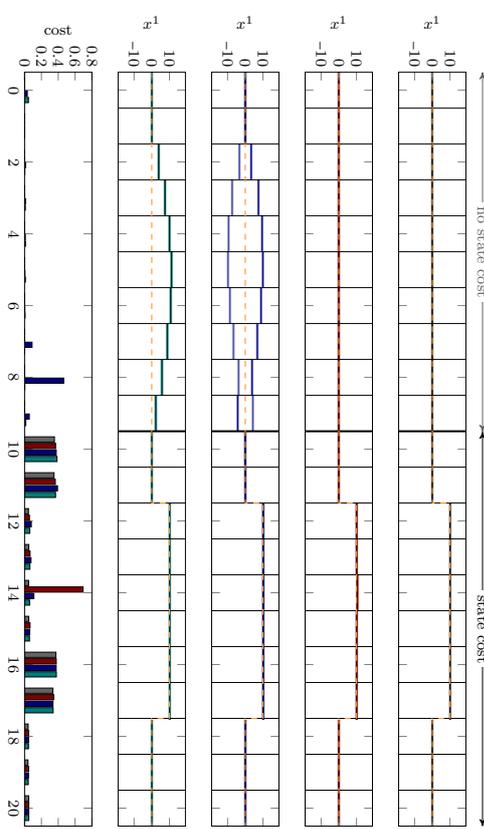


Figure 6: **Top four:** Density estimate for 50 trajectories (second state). From top to bottom: optimal oracle control (gray), certainty equivalent control (red), CE with Bayesian exploration bonus (blue), approximate dual control (green). Reference trajectory in dashed orange. **Bottom:** The mean cost per time step is shown in the bottom plot, with colors matching the controllers noted above.

The reference to be tracked passes through both nonlinear features but then stays at one of them:

$$r_{0:6} = \begin{bmatrix} -5 \\ 0 \end{bmatrix} \quad r_7 = \begin{bmatrix} -4 \\ 0 \end{bmatrix} \quad r_8 = \begin{bmatrix} -2 \\ 0 \end{bmatrix} \quad r_9 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad r_{10:20} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}.$$

The cost structure is

$$W_{0:5} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad W_{6:20} = \begin{bmatrix} 10 & 0 \\ 0 & 0 \end{bmatrix},$$

such that there is cost starting with the linear reference trajectory at time instant 6.

Fig. 7 shows the parameter belief and relevant state of a single run of this experiment over time. It shows clearly that the in the beginning necessary parameter θ^1 is learned early by BEB and the AD controller, while CE learns only “accidentally”. The BEB controller also learns the second parameter in the beginning, even though the knowledge will be lost over time. When the trajectory reaches the zone of the second parameter, the BEB controller tries to lower the growing uncertainty every now and then (visible by the drops in state x^1), incurring high cost. AD control completely ignores the growing uncertainty on θ^1 after reaching the area of θ^2 , thus preventing unnecessary exploration.

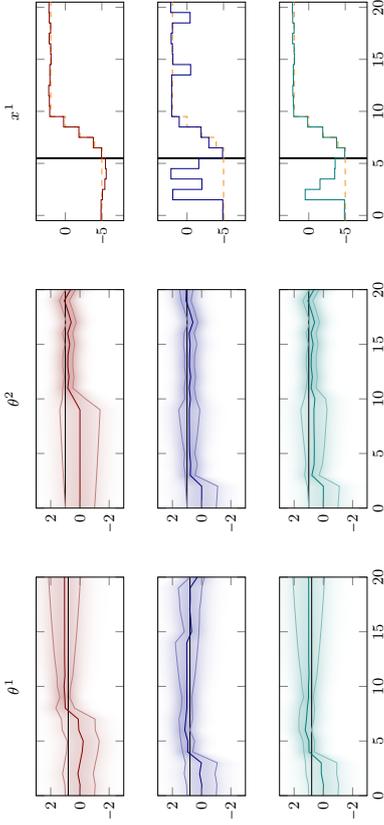


Figure 7: Parameter knowledge (left, middle) and state trajectory (right) for different controllers. From top to bottom: certainty equivalent control (red), CE with Bayesian exploration bonus (blue), approximate dual control (green). The true parameters are the black lines.

	Exp. 6.1		Exp. 6.2		Exp. 6.3		Exp. 6.4	
	mean	std	mean	std	mean	std	mean	std
Oracle	0.67	0.23	1.75	0.33	7.15	3.85	0.66	0.51
CE	0.84	0.73	2.49	0.74	15.72	5.20	1.76	0.90
CE-BEB	0.99	0.92	2.64	0.37	20.88	6.74	84.91	6.77
AD	0.77	0.43	1.96	0.34	14.33	5.40	1.62	0.56

Table 1: Average and standard deviation of costs in the experiments for 50 runs.

6.5 Quantitative Comparison

The above experiments aim to emphasize qualitative strengths of AD control over simpler approximations. It is desirable for a controllers to deal with flexible models of many parameters, many of which will invariably be superfluous. For reference, Table 6.5 also shows quantitative results: Averages and standard deviations of the cost, from the 50 runs for each controller. The AD controller shows good performance overall; interestingly, it also has low variance. CE and BEB were more prone to instabilities.

7. Conclusion

Bayesian reinforcement learning, or dual control, offers an elegant answer to the exploration-exploitation trade-off, relative to prior probabilistic beliefs. Its intricate, intractable structure requires approximations to balance another kind of trade-off, between computation and performance. This work investigated an old approximate framework from control, re-phrased

it in the language of reinforcement learning, and extended it to apply to contemporary inference methods from machine learning, including approximate Gaussian process regression and multi-layer networks. The result is a tractable approximation that captures notions of structured exploration, like the value of waiting for future exploration opportunities, and distinguishing relevant from irrelevant model parameters.

The dual control framework, in its now clearer form, offers interesting directions for research in reinforcement learning, including its combination with recent new developments in learning and planning. Following this conceptual work, the main challenge for further development is the still comparably high (but tractable) numerical load of dual control, particularly in problems of higher dimensionality.

Appendix A. Nonparametric EKF Form

The standard Kalman filter (KF) can be found in many textbooks (e.g., Särkkä, 2013) and therefore will not be restated here. Starting from the standard equations, we derive a general multi-step formulation of the classic KF with

$$p(z_k) = \mathcal{N}(z_k; m_k, P_k).$$

From there, state augmentation with an infinite-dimensional weight vector gives the expected result.

A.1 Derivation of the Multi-Step KF Formulation

Assuming that the result of the KF and the Gaussain process framework should be identical under certain circumstances, we wish to transform the KF to a formulation with full Gram matrix. Therefore, the prediction and update step have to be combined to

$$P_1 = (A_0 P_0 A_0^\top + Q) - (A_0 P_0 A_0^\top + Q) H^\top S_1^{-1} H (A_0 P_0 A_0^\top + Q)^\top$$

$$S_1 = H (A_0 P_0 A_0^\top + Q) H^\top + R,$$

which is pretty straightforward. We're adopting a standard notation, where P_k is the covariance at time step k , A_k is the Jacobian, Q is the drift and R is the measurement covariance. The same can be done for the second time step, but it is beneficial introducing a compact notation for the predictive covariance first

$$\begin{aligned} & (A_1 P_1 A_1^\top + Q) \\ &= (A_1 \left[(A_0 P_0 A_0^\top + Q) - (A_0 P_0 A_0^\top + Q) H^\top S_1^{-1} H (A_0 P_0 A_0^\top + Q)^\top \right] A_1^\top + Q) \\ &= \underbrace{A_1 (A_0 P_0 A_0^\top + Q) A_1^\top}_{=:g_{11}} + \underbrace{Q - A_1 (A_0 P_0 A_0^\top + Q) H^\top (S_1^{-1})^{-1} H (A_0 P_0 A_0^\top + Q) A_1^\top}_{=:g_{10}} \\ &= g_{11} - g_{10} H^\top g_{00}^{-1} H g_{01}. \end{aligned}$$

Using the compact notation and defining S_2 analogously to S_1 , we can write the two-step update as

$$\begin{aligned} P_2 &= (g_{11} - g_{10} H^\top S_1^{-1} H g_{01}) - (g_{11} - g_{10} H^\top S_1^{-1} H g_{01}) H^\top S_2^{-1} H (g_{11} - g_{10} H^\top S_1^{-1} H g_{01}) \\ P_2 &= g_{11} - g_{10} H^\top S_1^{-1} H g_{01} - g_{11} H^\top S_2^{-1} H g_{11} - g_{10} H^\top S_1^{-1} H g_{01} H^\top S_2^{-1} H g_{10} H^\top S_1^{-1} H g_{01} \\ &\quad + g_{11} H^\top S_2^{-1} H g_{10} H^\top S_1^{-1} H g_{01} + g_{10} H^\top S_1^{-1} H g_{01} H^\top S_2^{-1} H g_{11} \\ P_2 &= g_{11} - \underbrace{[g_{10} H^\top \quad g_{11} H^\top]}_{=:G^{-\top}} \begin{bmatrix} S_1^{-1} + S_1^{-1} H g_{01} H^\top S_2^{-1} H g_{01} H^\top S_1^{-1} & -S_1^{-1} H g_{01} H^\top S_2^{-1} \\ -S_2^{-1} H g_{10} H^\top S_1^{-1} & S_2^{-1} \end{bmatrix} \begin{bmatrix} H g_{01} \\ H g_{11} \end{bmatrix}. \end{aligned}$$

Application of Schur's lemma gives

$$G = \begin{bmatrix} H g_{00} H^\top + R & H g_{01} H^\top \\ H g_{10} H^\top & H g_{11} H^\top + R \end{bmatrix}.$$

Assuming full state measurement ($H = I$) for compactness of notation, the two-step update is

$$\begin{aligned} P_2 &= g_{11} - \begin{bmatrix} g_{10}^\top & g_{11}^\top \end{bmatrix} \begin{bmatrix} g_{00} + R & g_{01} \\ g_{10} & g_{11} + R \end{bmatrix}^{-1} \begin{bmatrix} g_{01} \\ g_{11} \end{bmatrix} \\ &= A_1 (A_0 P_0 A_0^\top + Q) A_1^\top + Q - [A_1 (A_0 P_0 A_0^\top + Q) \quad (A_1 (A_0 P_0 A_0^\top + Q) A_1^\top + Q)] \\ &\quad \cdot \begin{bmatrix} (A_0 P_0 A_0^\top + Q) + R & (A_0 P_0 A_0^\top + Q) A_1^\top \\ A_1 (A_0 P_0 A_0^\top + Q) A_1^\top + Q + R \end{bmatrix}^{-1} \begin{bmatrix} (A_0 P_0 A_0^\top + Q) A_1^\top \\ (A_1 (A_0 P_0 A_0^\top + Q) A_1^\top + Q) \end{bmatrix}, \end{aligned} \quad (20)$$

which already looks similar to GP inference. We can now generalize this two-step result to the general form by building the Gram matrix according to

$$G = \mathcal{F} \mathcal{P} \mathcal{F}^\top + \mathcal{F} \mathcal{Q} \mathcal{F}^\top + \mathcal{R},$$

where the individual parts are

$$\mathcal{P} = \begin{bmatrix} P_0 & 0 \\ 0 & 0 \end{bmatrix} \quad \mathcal{Q} = (Q \otimes I) \quad \mathcal{R} = (R \otimes I)$$

of appropriate size, depending on the current time k . The multi-step state transition matrix

$$\mathcal{F} = \begin{bmatrix} I & 0 & 0 & \dots & 0 \\ A_1 & I & 0 & \dots & 0 \\ A_2 & A_1 & A_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_m \dots A_1 & A_m \dots A_2 & \dots & I \end{bmatrix}, \quad \text{with} \quad \mathcal{F}^{-1} = \begin{bmatrix} I & 0 & 0 & \dots & 0 \\ -A_1 & I & 0 & \dots & 0 \\ 0 & -A_2 & I & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & -A_m & I \end{bmatrix},$$

is also needed so shift the initial covariance and drift covariances through time. Put together, this results in

$$P_k = \mathcal{F}_{k,:} (\mathcal{P} + \mathcal{Q}) \mathcal{F}_{k,:}^\top - \mathcal{F}_{k,:} (\mathcal{P} + \mathcal{Q}) \mathcal{F} (\mathcal{F} \mathcal{P} \mathcal{F}^\top + \mathcal{F} \mathcal{Q} \mathcal{F}^\top + \mathcal{R})^{-1} \mathcal{F}^\top (\mathcal{P} + \mathcal{Q}) \mathcal{F}_{k,:}$$

A more compact notation can be achieved by using \mathcal{F}^{-1} to obtain

$$P_k = \mathcal{F}_{k,:} (\mathcal{P} + \mathcal{Q}) \mathcal{F}_{k,:}^\top - \mathcal{F}_{k,:} (\mathcal{P} + \mathcal{Q}) (\mathcal{P} + \mathcal{Q} + \mathcal{F}^{-1} \mathcal{R} \mathcal{F}^{-\top})^{-1} (\mathcal{P} + \mathcal{Q}) \mathcal{F}_{k,:} \quad (21)$$

Calculating the mean prediction is done analogously:

$$m_k = \mathcal{F}_{k,0} m_0 + \mathcal{F}_{k,:} (\mathcal{P} + \mathcal{Q}) (\mathcal{P} + \mathcal{Q} + \mathcal{F}^{-1} \mathcal{R} \mathcal{F}^{-\top})^{-1} \mu.$$

A.2 Augmenting the State

Instead of tracking only the state covariance, in the GP setting also the dynamics function has to be inferred. The system equations of the nonlinear system are now

$$\begin{aligned} x_k &= f(x_{k-1}) + q_{k-1} & q_{k-1} &\sim \mathcal{N}(0, Q) \\ y_k &= H x_k + r_k & r_k &\sim \mathcal{N}(0, R), \end{aligned}$$

where $f \sim \mathcal{GP}(0, k)$. The inference in this model can be done through the EKF with augmented state. We adopt the weight-space view with $f = \Phi^\top w$ (see Rasmussen and Williams, 2006) to augment the state with the infinite-dimensional weight vector w :

$$z = \begin{pmatrix} x \\ w \end{pmatrix} \quad \Sigma = \begin{pmatrix} P & \Sigma^{vw} \\ \Sigma^{wx} & \Sigma^{ww} \end{pmatrix} \quad J_A = \begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial w} \\ 0 & I \end{pmatrix} = \begin{pmatrix} A & \Phi \\ 0 & I \end{pmatrix},$$

where, in (20), the original x is replaced by the augmented z , P by Σ and A by J_A .

Choosing $H = [I \ 0]$ so that H recovers the original states from the augmented state vector, we obtain, after calculations similar to those above, a Gram matrix with additional terms including feature functions and the prior on them:

$$G^* = G + \begin{pmatrix} \Phi_0 \Sigma_0^{ww} \Phi_0^\top & \Phi_0 \Sigma_0^{vw} \Phi_0^\top \\ (A_1 \Phi_0 + \Phi_1) \Sigma_0^{vw} \Phi_0^\top & (A_1 \Phi_0 + \Phi_1) \Sigma_0^{ww} (\Phi_0^\top A_1^\top + \Phi_1^\top) \end{pmatrix}.$$

At this point it is important to note that the infinite inner product $\Phi_k \Sigma_k^{ww} \Phi_k^\top$ corresponds to an evaluation of the kernel

$$\Phi(x) \Sigma_0^{ww} \Phi(x')^\top = \sum \phi_i(x) \Sigma_{0,i,j}^{ww} \phi_j(x') = \kappa(x, x').$$

This means we can write the Gram matrix as

$$G^* = G + \begin{pmatrix} \kappa_{00} & \kappa_{00} A_1^\top + \kappa_{01} \\ A_1 \kappa_{00} + \kappa_{10} & A_1 \kappa_{00} A_1^\top + \kappa_{10} A_1^\top + A_1 \kappa_{01} + \kappa_{11} \end{pmatrix},$$

where we have written κ_{\cdot} for $\kappa(\cdot, \cdot)$ to save space. In total, the Gram matrix is then

$$G^* = \mathcal{F} \mathcal{P} \mathcal{F}^\top + \mathcal{F} \mathcal{Q} \mathcal{F}^\top + \mathcal{F} \mathcal{K} \mathcal{F}^\top + \mathcal{R},$$

with $\mathcal{K} = \kappa(\mathbf{y}_{1:m}, \mathbf{y}_{1:m})$. Since inference is more compact and numerically stable if we absorb \mathcal{F} into the Gram matrix as in Eq. (21), we define

$$\mathcal{G} = \mathcal{P} + \mathcal{Q} + \mathcal{K} + \mathcal{F}^{-1} \mathcal{R} \mathcal{F}^{-\top}.$$

Inference is done according to

$$P_k = \mathcal{F}_{k,\cdot} (\mathcal{P} + \mathcal{Q} + \mathcal{K}) \mathcal{F}_{k,\cdot}^\top - \mathcal{F}_{k,\cdot} (\mathcal{P} + \mathcal{Q} + \mathcal{K}) \mathcal{G}^{-1} (\mathcal{P} + \mathcal{Q} + \mathcal{K}) \mathcal{F}_{\cdot,k}$$

$$\Sigma_k^{wx} = (\Sigma_k^{ww})^\top = \mathcal{F}_{k,\cdot} \Phi(\mathbf{y}) \Sigma_0^{ww} - \mathcal{F}_{k,\cdot} (\mathcal{P} + \mathcal{K} + \mathcal{Q}) \mathcal{G}^{-1} \Phi(\mathbf{y}) \Sigma_0^{ww}$$

$$\Sigma_k^{ww} = \Sigma_0^{ww} - \Sigma_0^{ww} \Phi(\mathbf{y})^\top \mathcal{G}^{-1} \Phi(\mathbf{y}) \Sigma_0^{ww}$$

for the covariance and

$$m_k = \mathcal{F}_{k,0} m_0 + \mathcal{F}_{k,\cdot} (\mathcal{P} + \mathcal{Q} + \mathcal{K}) \mathcal{G}^{-1} \mathbf{y}$$

$$\bar{w}_k = \bar{w}_0 + \Sigma_0^{ww} \Phi(\mathbf{y})^\top \mathcal{G}^{-1} \mathbf{y}$$

for the mean.

Appendix B. Gradients and Hessians of Dynamics Functions

B.1 Neural Network Basis Functions

The neural network dynamics function is

$$f(x) = \sum_{i=1}^F v_i \sigma(w_i(x - b_i)), \quad \sigma(a) = \frac{1}{1 + e^{-a}},$$

with the well-known derivatives of the logistic

$$\frac{\partial}{\partial a} \sigma(a) = \sigma(a)(1 - \sigma(a)), \quad \frac{\partial^2}{\partial a^2} \sigma(a) = \sigma(a)(1 - \sigma(a))(3 - 2\sigma(a)).$$

The gradient of $f(x)$ can easily found to be

$$\nabla f(x) = \begin{bmatrix} \sum_{i=1}^F v_i w_i \sigma(w_i(x - b_i))(1 - \sigma(w_i(x - b_i))) \\ \sigma(w_1(x - b_1)) \\ \vdots \\ v_1(x - b_1) \sigma(w_1(x - b_1))(1 - \sigma(w_1(x - b_1))) \\ \vdots \\ v_F(x - b_F) \sigma(w_F(x - b_F))(1 - \sigma(w_F(x - b_F))) \end{bmatrix}.$$

The Hessian, written in parts, using $a_i = w_i x + b_i$, is:

$$\nabla_x^2 f(x) = \sum_{i=1}^F v_i w_i^2 \sigma(a_i)(1 - \sigma(a_i))(3 - 2\sigma(a_i))$$

$$\nabla_x \nabla_{v_i} f(x) = w_i \sigma(a_i)(1 - \sigma(a_i))$$

$$\nabla_x \nabla_{w_i} f(x) = v_i \sigma(a_i)(1 - \sigma(a_i)) + (x - b_i) w_i v_i \sigma(a_i)(1 - \sigma(a_i))(1 - 2\sigma(a_i))$$

$$\nabla_{v_i} \nabla_{v_i} f(x) = 0$$

$$\nabla_{v_i} \nabla_{w_i} f(x) = (x - b_i) \sigma(a_i)(1 - \sigma(a_i))$$

$$\nabla_{w_i} \nabla_{w_i} f(x) = v_i (x - b_i)^2 \sigma(a_i)(1 - \sigma(a_i))(3 - 2\sigma(a_i)).$$

B.2 Fourier Basis Functions

The Fourier approximation to the dynamics function has the form

$$f(x) = \sqrt{\frac{F}{2}} \sum_{i=1}^{F/2} v_{2i-1} \sin(\omega_{2i-1} x) + v_{2i} \cos(\omega_{2i} x).$$

The gradient of $f(x)$ can easily verified to be

$$\nabla f(x) = \begin{bmatrix} \sqrt{\frac{2}{F}} \sum_{i=1}^{F/2} v_{2i-1} \omega_{2i-1} \cos(\omega_{2i-1} x) - v_{2i} \omega_{2i} \sin(\omega_{2i} x) \\ \sqrt{\frac{2}{F}} \sin(\omega_1 x) \\ \sqrt{\frac{2}{F}} \cos(\omega_2 x) \\ \vdots \\ \sqrt{\frac{2}{F}} \sin(\omega_{F-1} x) \\ \sqrt{\frac{2}{F}} \cos(\omega_F x) \end{bmatrix}.$$

The Hessian, written in parts, using $c = \sqrt{2/F}$ for normalization, is:

$$\begin{aligned} \nabla_x^2 f(x) &= c \sum_{i=1}^{F/2} -v_{2i-1} \omega_{2i-1}^2 \sin(\omega_{2i-1} x) - v_{2i} \omega_{2i}^2 \cos(\omega_{2i} x) \\ \nabla_x \nabla_{v_i} f(x) &= \begin{cases} c \omega_i \cos(\omega_i x) & i \text{ odd} \\ -c \omega_i \sin(\omega_i x) & i \text{ even} \end{cases} \\ \nabla_{v_i} \nabla_{v_i} f(x) &= 0. \end{aligned}$$

B.3 Radial Basis Functions

With radial basis function features, the dynamics function is

$$f(x) = \sum_{i=1}^F v_i \exp\left(-\frac{(x-c_i)^2}{2\lambda^2}\right).$$

The gradient of $f(x)$ is

$$\nabla f(x) = \begin{bmatrix} \sum_{i=1}^F v_i \exp\left(-\frac{(x-c_i)^2}{2\lambda^2}\right) \frac{(c_i-x)}{2\lambda^2} \\ \exp\left(-\frac{(x-c_1)^2}{2\lambda^2}\right) \\ \vdots \\ \exp\left(-\frac{(x-c_F)^2}{2\lambda^2}\right) \end{bmatrix}.$$

The Hessian, written in parts, is:

$$\begin{aligned} \nabla_x^2 f(x) &= \sum_{i=1}^F v_i \exp\left(-\frac{(x-c_i)^2}{2\lambda^2}\right) \left[\left(\frac{c_i-x}{2\lambda^2}\right)^2 - \frac{1}{\lambda^2} \right] \\ \nabla_x \nabla_{v_i} f(x) &= \exp\left(-\frac{(x-c_i)^2}{2\lambda^2}\right) \frac{c_i-x}{2\lambda^2} \\ \nabla_{v_i} \nabla_{v_i} f(x) &= 0 \end{aligned}$$

References

- F. Allgöwer, T.A. Badgwell, J.S. Qin, J.B. Rawlings, and S.J. Wright. Nonlinear predictive control and moving horizon estimation: an introductory overview. In *Advances in Control*, pages 391–449. Springer, 1999.
- M. Aoki. *Optimization of Stochastic Systems*. Academic Press, New York - London, 1967.
- J.-Y. Audibert, R. Munos, and G. Szepesvári. Exploration-exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009.
- Y. Bar-Shalom. Stochastic dynamic programming: Caution and probing. *IEEE Transactions on Automatic Control*, 26(5):1184–1195, 1981.
- Y. Bar-Shalom and E. Tse. Dual effect, certainty equivalence, and separation in stochastic control. *IEEE Transactions on Automatic Control*, 19(5):494–500, 1974.
- Y. Bar-Shalom and E. Tse. Caution, probing, and the value of information in the control of uncertain systems. *Annals of Economic and Social Measurement*, 5(3):323–337, 1976.
- R.E. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.
- D.P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 3rd edition, 2005.
- O. Chapelle and L. Li. An Empirical Evaluation of Thompson Sampling. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2249–2257, 2011.
- R. Dearden, N. Friedman, and D. Andre. Model based Bayesian exploration. In *Uncertainty in Artificial Intelligence (UAI)*, volume 15, pages 150–159, 1999.
- M.P. Deisenroth and C.E. Rasmussen. PILCO: A Model-Based and Data-Efficient Approach to Policy Search. In *International Conference on Machine Learning (ICML)*, 2011.
- M. Diehl, H.J. Ferreau, and N. Haverbeke. Efficient numerical methods for nonlinear MPC and moving horizon estimation. In *Nonlinear Model Predictive Control*, volume 384, pages 391–417. Springer, 2009.
- S.E. Dreyfus. Some types of optimal control of stochastic systems. *Journal of the Society for Industrial & Applied Mathematics, Series A: Control*, 21(1):120–134, 1964.
- M.O.G. Duff. *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes*. PhD thesis, University of Massachusetts, Amherst, 2002.
- A.A. Feldbaum. Dual Control Theory I-IV. *Avtomatika i Telemekhanika*, 21(9), 21(11), 22(1), 22(2), 1960–1961.
- N.M. Filatov and H. Unbehauen. Survey of adaptive dual control methods. *IEEE Proceedings on Control Theory and Applications*, 147(1):118–128, 2000.
- N.M. Filatov and H. Unbehauen. *Adaptive Dual Control*. Springer Verlag, Berlin, 2004.
- T. Glad and L. Ljung. *Control theory: Multivariable and Nonlinear Methods*. Taylor and Francis, New York, London, 2000.
- P. Hennig. Optimal Reinforcement Learning for Gaussian Systems. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.
- O.L.R. Jacobs and J.W. Patehall. Caution and probing in stochastic control. *International Journal of Control*, 16(1):189–199, 1972.
- H.J. Kappen. Optimal control theory and the linear Bellman equation. *Inference and Learning in Dynamic Models*, pages 363–387, 2011.

- J.Z. Kolter and A.Y. Ng. Near-Bayesian exploration in polynomial time. In *International Conference on Machine Learning (ICML)*, 2009.
- H. König. *Eigenvalue Distribution of Compact Operators*. Birkhäuser, 1986.
- W.G. Macready and D.H. Wolpert. Bandit problems and the exploration/exploitation tradeoff. *IEEE Transactions on Evolutionary Computation*, 2(1):2–22, 1998.
- D.Q. Mayne. A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems. *International Journal of Control*, 3(1):85–95, 1966.
- D.H. Nguyen and B. Widrow. Neural networks for self-learning control systems. *IEEE Control Systems Magazine*, 10(3):18–23, 1990.
- P. Poupart, N. Vlassis, J. Hoey, and K. Regan. An analytic solution to discrete Bayesian reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2006.
- A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1177–1184, 2008.
- H. Raiffa and R. Schlaifer. *Applied statistical decision theory*. Studies in managerial economics. Harvard University, Boston, 1961.
- C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. MIT, 2006.
- H. Robbins and S. Mouro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, Sep. 1951.
- D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.
- S. Särkkä. *Bayesian filtering and smoothing*. Cambridge University Press, 2013.
- S. Singhal and L. Wu. Training multilayer perceptrons with the extended kalman algorithm. In *Advances in Neural Information Processing Systems (NIPS)*, pages 133–140, 1989.
- N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. In *International Conference on Machine Learning (ICML)*, 2010.
- M.L. Stein. *Interpolation of spatial data: some theory for Kriging*. Springer Verlag, 1999.
- J. Sternby. A simple dual control problem with an analytical solution. *IEEE Transactions on Automatic Control*, 21(6):840–844, 1976.
- W.R. Thompson. On the Likelihood that one unknown probability exceeds another in view of two samples. *Biometrika*, 25:275–294, 1933.
- E. Tse and Y. Bar-Shalom. An actively adaptive control for linear systems with random parameters via the dual control approach. *IEEE Transactions on Automatic Control*, 18(2):109–117, 1973.
- E. Tse, Y. Bar-Shalom, and L. Meier III. Wide-sense adaptive dual control for nonlinear stochastic systems. *IEEE Transactions on Automatic Control*, 18(2):98–108, 1973.
- J. Uhlmann. *Dynamic Map Building and Localization: New Theoretical Foundations*. PhD thesis, University of Oxford, 1995.
- B. Wittenmark. Adaptive dual control methods: An overview. In *IFAC symposium on Adaptive Systems in Control and Signal Processing*, pages 67–72, 1995.

Multiple-Instance Learning from Distributions

Gary Doran

Soumya Ray

*Department of Electrical Engineering and Computer Science
Case Western Reserve University
10900 Euclid Ave, Glennan 320
Cleveland, OH 44106, USA*

GARY.DORAN@CASE.EDU

SRAY@CASE.EDU

Editor: Luc De Raedt

Abstract

We propose a new theoretical framework for analyzing the multiple-instance learning (MIL) setting. In MIL, training examples are provided to a learning algorithm in the form of labeled sets, or “bags,” of instances. Applications of MIL include 3-D quantitative structure–activity relationship prediction for drug discovery and content-based image retrieval for web search. The goal of an algorithm is to learn a function that correctly labels new bags or a function that correctly labels new instances. We propose that bags should be treated as latent distributions from which samples are observed. We show that it is possible to learn accurate instance- and bag-labeling functions in this setting as well as functions that correctly rank bags or instances under weak assumptions. Additionally, our theoretical results suggest that it is possible to learn to rank efficiently using traditional, well-studied “supervised” learning approaches. We perform an extensive empirical evaluation that supports the theoretical predictions entailed by the new framework. The proposed theoretical framework leads to a better understanding of the relationship between the MI and standard supervised learning settings, and it provides new methods for learning from MI data that are more accurate, more efficient, and have better understood theoretical properties than existing MI-specific algorithms.

Keywords: multiple-instance learning, learning theory, ranking, classification

1. Introduction

The standard supervised learning setting, in which labeled training examples are represented with individual feature vectors, is well-studied with numerous applications. However, there remain many compelling real-world problems that require learning from more structured data. For example, in text categorization, a document might contain a set of passages or paragraphs. A typical approach is to simply ignore the internal structure within the document by treating it as a “bag of words,” then to represent the document with a single feature vector based on word frequencies. However, because such an approach destroys the internal structure of the document, it becomes challenging to determine *which* passages or paragraphs correspond to the category of interest. Similarly, for the content-based image retrieval (CBIR) domain, the goal is to learn to retrieve images that contain some object of interest to the user. One approach is to use a flat feature vector representation of each

image given pixel color values. Again, using such an approach, it is not clear how one might identify *which* object in or region of the image was of interest to a user.

For such problems, the multiple-instance (MI) setting offers a richer representation for structure objects as sets, or “bags,” of feature vectors, each of which is called an “instance” (Dietterich et al., 1997). In the text categorization example above, a document is a bag of passages or paragraphs, which are the instances. For CBIR, an image is a bag of segments or objects. The MI setting further assumes that labels exist at both the level of instances and bags, where a bag’s label is the logical conjunction of Boolean instance labels. That is, a bag is positive if *at least one* instance in the bag is positive and negative if *all* of the instances in the bag are negative. This logical relationship corresponds to the fact that a document or an image is of the class of interest if and only if at least one of the passages or objects it contains is of the class of interest.

In the standard supervised setting, there is typically only one target concept of interest. For MI learning, one might be interested in learning either a bag or an instance concept from MI data. For example, in the 3-Dimensional Quantitative Structure–Activity Relationship (3D-QSAR) domain, the goal is to learn to predict whether a molecule will bind to a given target receptor (Dietterich et al., 1997). Because a molecule has flexible bonds, it exists in multiple shapes, or *conformations*, in solution. Thus, mapping this problem into the MI setting, conformations are instances and molecules are bags. A *bag-labeling* function can be used to predict whether a given molecule will bind to a target receptor. On the other hand, an *instance-labeling* function can be used to predict which specific conformations will bind to a receptor, providing useful, difficult-to-measure information about the receptor’s physical structure.

Despite the importance of these two learning tasks, only the bag-labeling task has received much attention in recent prior work characterizing the learnability of MI concepts (Sabato and Tishby, 2012). When learnability of instance concepts has been addressed, it has been under the strict, unrealistic assumption that instances across all bags are independent and identically distributed (IID) samples *from the same underlying distribution* (Blum and Kalai, 1998). However, as far as we know, the result of Blum and Kalai (1998) has remained the only positive result on instance concept learnability in the MI setting for over a decade. In this paper, we describe new *positive* results for both instance- and bag-concept learnability. Our contributions are summarized as follows:

1. We describe a new generative model for MI data and show that it subsumes some previously proposed generative models for MI learning (MIL).
2. We provide novel results for learning accurate bag-level concepts from MI data.
3. We describe the first positive instance concept learnability results since those of Blum and Kalai (1998).¹
4. We prove the first results, to our knowledge, that formally describe the ability to rank both instances and bags in the MI setting.
5. We empirically evaluate a surprising implication of our theoretical results: that *standard supervised approaches* can effectively rank both instances and bags in the MI setting.

¹. These results were also presented in Doran and Ray (2014).

setting. Our evaluation uses 55 data sets from a wide variety of domains, and supports both our theoretical results as well as the assumptions made by our generative model.

2. Bags as Distributions

In this section, we describe a generative model for MI data in which bags are viewed as *distributions over instances* rather than as sets of instances. We show that the proposed generative model actually encompasses previous, standard models of MI learning in which bags are sets or tuples. The choice of framing a problem within a particular theoretical model has significant practical consequences for designing or selecting an algorithm to solve the problem. This section provides a theoretical framework in which the MI classification problem can be analyzed. The model allows us to derive positive instance- and bag-concept learnability results for the MI setting as described in Section 3. Furthermore, as Section 4 shows, the generative model leads to a surprising yet testable hypothesis that standard supervised algorithms can learn from MI data. This hypothesis is evaluated experimentally, supporting the assumptions made by the model.

2.1 The Generative Model

At the heart of this work is the claim that bags are best viewed as distributions rather than as finite sets of instances. Below, we formally define what we mean by this statement. But first, the example domain of drug activity prediction provides an intuitive justification for this claim. As described in Section 1, in the drug activity prediction domain, the goal is to predict the ability of molecules to activate, or bind to, a receptor. To cast the problem as binary classification, we select some threshold so that each molecule’s activity level either corresponds to an “active” or “inactive” label. In this case, we can think of each molecule (bag) as being drawn from a distribution D_B over molecules. Ignoring for the moment that each molecule has numerous conformations, this molecule either activates the receptor or not, so in nature the labeling function is defined at the level of bags. Prior models represent each molecule as a set or multiset of conformations, so they implicitly assume that each molecule exists in only a finite number of conformations. In reality, a molecule can transform continuously from conformation to conformation, producing an infinite set of conformations. In particular, each molecule exists in a state of dynamic equilibrium in which the amount of time it spends in each conformation is distributed according to Gibbs free energy such that low-energy conformations are preferred. Hence, the molecule (bag) corresponds to a *distribution over instances*. Constructing a bag from low-energy conformations, the common procedure for constructing bags in the drug activity domain, can be thought of as sampling instances from this distribution. Note that each molecule will have a *unique* distribution over conformations; thus, prior generative models for MIL that assume all instances are drawn from the *same* distribution are not applicable (Blum and Kalai, 1998). In Section 2.5, we describe how this view of the MI generative process can be applied to other problem domains.

More abstractly, our generative process needs several components. First, *bags are distributions over instances*, and we will assume that these bags are sampled from a *distribution over bags* and labeled according to a *bag-labeling function*. In addition to the bag-labeling

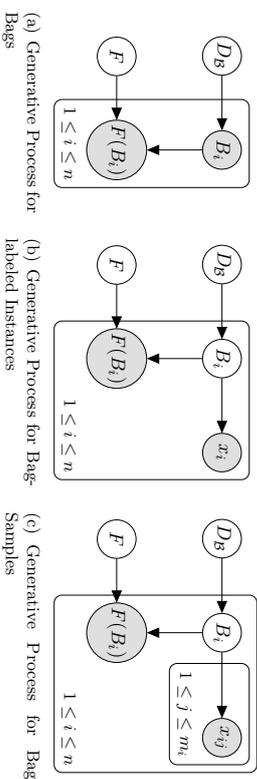


Figure 1: A comparison of the generative processes for bags, individual bag-labeled instances, and bag samples.

function, in the MI setting there is also an *instance-labeling* function with the standard *MI assumption* relating the bag- and instance-labeling functions. We describe the *instance distribution* that is consistent with the generative process, which will be useful for discussing instance concept learnability. In addition to these essential components, we introduce two *additional weak assumptions* that make efficient learning in this setting possible.

Bags as Distributions. To formalize the intuition above, suppose we have an instance space \mathcal{X} . Typically, the space of bags is some subset of \mathcal{X}^* , the set of all finite subsets of \mathcal{X} . However, here, we let the space of bags be $\mathcal{B} = \mathcal{P}(\mathcal{X})$, the set of probability distributions on the input space. Hence, each bag $B \in \mathcal{B}$ is a probability distribution over instances, denoted $P(x | B)$.

Bag Distribution. We propose that, at the level of bags, the MI generative process is similar to that for supervised learning. In particular, bags are sampled from some fixed distribution D_B , which is a distribution over instance distributions ($D_B \in \mathcal{P}(\mathcal{P}(\mathcal{X}))$). From this distribution D_B , we sample some set of bags $\{B_i\}_{i=1}^n$, as illustrated by the plate model in Figure 1(a).

Bag-Labeling Function. As in supervised learning, we assume that there exists some labeling function $F : \mathcal{B} \rightarrow \{0, 1\}$ that labels bags. Thus, a supervised data set $\{(B_i, F(B_i))\}_{i=1}^n$ could be produced by sampling bags IID from D_B and applying the labeling function F .

Instance-Labeling Function. In the MI setting, we assume that in addition to the bag-labeling function F , there also exists an *instance-labeling* function $f : \mathcal{X} \rightarrow \{0, 1\}$. A key component of the MI setting is not only the existence of both bag and instance labeling functions, but the relationship between the two as well. Traditionally, the MI assumption is stated with respect to particular sets of instances so that a bag label $F(B_i)$ is the logical OR (for boolean labels), or maximum (for numerical labels), of its instances’ labels: $F(B_i) = \max_j f(x_{ij})$. However, in the proposed generative model, bags are distributions with *a priori* labels regardless of the instances sampled from them. Therefore, our generative model requires a more nuanced description of the relationship between bag and instance labels.

The MI Assumption. We state the relationship between F and f at the level of the generative model. Accordingly, a bag is negative ($F(B) = 0$) if and only if probability of sampling a positive instance within the bag is zero: $\mathbb{P}_{x \sim B}[f(x) = 1] = 0$. In measure theoretic terms, instances sampled within negative bags are almost surely negative, which implies that positive instances are almost surely sampled only within positive bags. This condition corresponds to the standard MI assumption that negative bags contain only negative instances.

Instance Distribution. In order to talk about the learnability of f , we must define some instance distribution with respect to which we will measure risk. An instance distribution naturally arises from our generative model if we first sample a bag B randomly from D_B , then sample an instance x randomly from the distribution corresponding to B . The instance distribution D_X resulting from this two-level sampling procedure is effectively the distribution that marginalizes out the individual bag distributions. That is, given a probability distribution P_B over bags corresponding to D_B , we can define a distribution P_X corresponding to D_X as

$$P_X(x) = \int_B P(x | B) dP_B(B). \quad (1)$$

Given that “ x ” is used to denote instances and “ B ” is used to denote bags, we subsequently drop subscripts from P when the sample space can be inferred from context. As we discuss in Section 3.4, the ability to marginalize out bag-specific distributions in our model plays a vital role in proving the learnability of instance- and bag-labeling functions. Given a bag distribution, the existence of such an instance distribution is guaranteed under relatively weak assumptions on the instance space \mathcal{X} (Diestel and Uhl, 1977). Furthermore, note that while we can view instances in our generative model as being sampled IID from D_X , this does not require the assumption that instances are IID across all bag distributions, as in prior generative models for MIL (Blum and Kalai, 1998). We discuss this point in detail in Section 3.6.

Additional Assumptions. As is the case in the standard MI framework, in our generative model, only bag labels are observed. Suppose we sample individual instances as illustrated in Figure 1(b) where we first sample a bag, record its label, and then sample an instance from the bag-specific distribution $P(x | B)$ and assign the bag label to the instance. Then the resulting bag-labeled instances $\{(x_{i1}, F(B_i))\}_{i=1}^n$ are distributed according to D_X , and will appear in positive bags some of the time and negative bags the remaining fraction of the time. Therefore, each instance will have some probability $c(x) \in [0, 1]$ of appearing with a positive label, which can be formally expressed as a probabilistic concept (p -concept) like the kind described by Kearns and Schapire (1994):

$$c(x) \triangleq \mathbb{P}[F(B) = 1 | x]. \quad (2)$$

That is, the probability of observing a positive label for instance x is the conditional probability that the bag B in the two-level sampling procedure was positive, given that x was observed within B . This conditional probability can be derived from the joint distribution over instances and bag labels corresponding to the generative process in Figure 1(b).

It follows from the previously-stated relationship between F and f that for any positive instance x_+ , $c(x_+) = 1$, since each positive instance is observed almost surely (with probability 1) within a positive bag. In order to distinguish positive and negative instances, we

make the following weak assumption: there exists some $\gamma > 0$ such that for every negative instance x_- , $c(x_-) \leq 1 - \gamma$. Intuitively, this corresponds to the assumption that every negative instance is observed with some nonzero probability in a negative bag.

To see why negative instances must appear in negative bags in order to learn a concept, consider trying to learn the instance concept “spoon” in the CBIR domain, as described in Section 1. To learn this concept, you are given a set of images containing spoons, and a set of images not containing spoons. However, suppose that in every image containing a spoon, there is also a fork nearby. Furthermore, forks never appear alone in images without spoons. In this unfortunate scenario, you have no means of determining which of the fork or spoon is the positive instance given only image-level labels. However, if there is a guarantee that eventually you will see a negative image containing a fork but not a spoon, you will be able to learn that the fork is not the positive instance. We discuss learnability further in Section 3 and Section 4.

Finally, for learning bag-level concepts, we show in Section 3.2 that we require one additional assumption that there is some minimum fraction π of positive instances in each positive bag. That is, for every positive bag B_+ , $\mathbb{P}[f(x) = 1 | B_+] \geq \pi$. Without this assumption, there might be positive bags that only contain negative instances. However, this would make them indistinguishable during bag labeling from negative bags, which by definition only contain negative instances. Interestingly, this assumption is not required if we are only interested in learning an instance-level concept.

Now, we can formally define MI-GEN, the set of generative processes for MI data consistent with the assumptions described above,

Definition 1 (MI-GEN) *Given any $\gamma \in (0, 1]$ and $\pi \in [0, 1]$, MI-GEN(γ, π) is the set of all tuples (D_X, D_B, f, F) , each consisting of an instance distribution D_X (with corresponding $P(x)$), bag distribution D_B (with corresponding $P(B)$), instance-labeling function f , and bag-labeling function F , that satisfy the conditions:*

1. $\mathbb{P}(x) = \int_B \mathbb{P}(x | B) dP(B)$
2. $\forall x : f(x) = 1 \implies \mathbb{P}[F(B) = 0 | x] = 0$
3. $\forall x : f(x) = 0 \implies \mathbb{P}[F(B) = 0 | x] \geq \gamma$
4. $\forall B : F(B) = 1 \implies \mathbb{P}[f(x) = 1 | B] \geq \pi$.

For simplicity, we will write MI-GEN(γ) for the case when $\pi = 0$, which corresponds to the weakest Condition 4. That is, for any fixed γ , MI-GEN(γ) \supseteq MI-GEN(γ, π) for every $\pi \geq 0$. Such notation will be used when discussing instance-concept learnability, which does not require the $\pi > 0$ assumption. That is, instance-concept learning under our model is naturally tolerant to “bag label noise” of the form where positive bags contain only negative instances.

Finally, note that for any $\gamma \in (0, 1]$, $\pi \in [0, 1]$, MI-GEN(γ, π) \supseteq MI-GEN(1, 1). That is, $\gamma = \pi = 1$ corresponds to the strongest constraints on the generative process. Even in this case, for any D_X and f , there exist D_B and F such that $(D_X, D_B, f, F) \in \text{MI-GEN}(1, 1)$. In particular, given a point mass δ_x centered on x , we can define D_B so that $P_B(\delta_x) = P_X(x)$ and F such that $F(\delta_x) = f(x)$. This choice of (D_B, F) corresponds to supervised learning

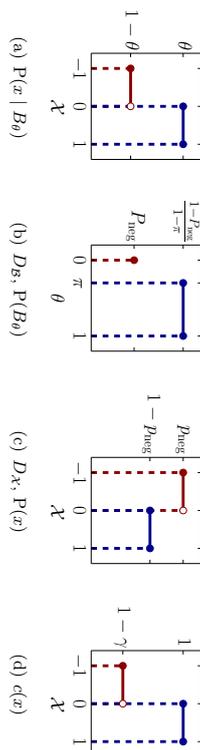


Figure 2: An example generative process for MI data. Each bag distribution (a) is parameterized by θ , and the distribution over bags (b) corresponds to a distribution over values of θ . The resulting distribution over instance (c) is derived in Equation 4 and Equation 5. The probability of instances appearing in positive bags (d) is derived in Equation 6 and Equation 7.

expressed in our generative model. That is, sampling from our generative process in that case is indistinguishable from sampling directly from D_X with labels assigned according to f . Below, we discuss the relationship between our generative model and other proposed models for MI learning.

2.2 An Example of the Generative Process

As a concrete example, suppose the instance space is the closed real-valued interval $\mathcal{X} = [-1, 1]$ and each bag B_θ is a distribution parameterized by a single real-valued parameter $\theta \in [0, 1]$. As illustrated in Figure 2(a), the bag distribution $P(x | B_\theta)$ assigns $(1 - \theta)$ of the probability mass uniformly to the interval $[-1, 0)$, and θ of the mass uniformly to the interval $[0, 1]$. Each value of θ corresponds to a different bag, which is a different distribution over instances.

In this example, a distribution over bags is essentially a distribution over the bag parameter θ . Such a distribution is illustrated in Figure 2(b), and assigns P_{neg} of the mass to the set $\{0\}$ and the remaining $1 - P_{\text{neg}}$ portion of the mass uniformly to the interval $[\pi, 1]$. The probability of sampling a bag, $P(B_\theta)$, corresponds to the probability of sampling the corresponding value of θ . Similarly, a bag-labeling function F can be defined in terms of θ as follows:

$$F(B_\theta) = \begin{cases} 0 & \text{if } \theta = 0 \\ 1 & \text{if } \theta > 0. \end{cases} \quad (3)$$

Thus, for this example, $P_{\text{neg}} = P[F(B_\theta) = 0]$.

For the sake of the example, we choose the instance-labeling function to be

$$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0. \end{cases}$$

This choice is consistent with the bag-labeling function defined in Equation 3, since $F(B_\theta) = 0$ implies $\theta = 0$, which implies that the probability of sampling a positively labeled $x \in [0, 1]$ is zero as required.

If we marginalize out the bag distribution, we obtain the single instance distribution in Figure 2(c). Analytically, for any $x_- \in [-1, 0)$, we have

$$\begin{aligned} P(x_-) &= \int_0^1 P(x_- | B_\theta) P(B_\theta) d\theta \\ &= 1 \cdot P_{\text{neg}} + \int_\pi^1 (1 - \theta) \frac{1 - P_{\text{neg}}}{1 - \pi} d\theta \\ &= P_{\text{neg}} + \frac{1}{2}(1 - P_{\text{neg}})(1 - \pi) \triangleq P_{\text{neg}}. \end{aligned} \quad (4)$$

Similarly, for $x_+ \in [0, 1]$,

$$\begin{aligned} P(x_+) &= \int_0^1 P(x_+ | B_\theta) P(B_\theta) d\theta \\ &= 0 \cdot P_{\text{neg}} + \int_\pi^1 \frac{\theta - P_{\text{neg}}}{1 - \pi} d\theta \\ &= \frac{1}{2}(1 - P_{\text{neg}})(1 + \pi) = 1 - P_{\text{neg}}. \end{aligned} \quad (5)$$

Since probability density functions exist for this example, we can analytically compute $c(x)$ given the following expression:

$$c(x) = P[F(B) = 1 | x] = \frac{\int_{B_+} P(x | B) dP(B)}{P(x)},$$

where $B_+ = \{B : F(B) = 1\}$. As described, for positive instances $x_+ \in [0, 1]$, we have

$$c(x_+) = \frac{\int_\pi^1 P(x_+ | B_\theta) P(B_\theta) d\theta}{\frac{1}{2}(1 - P_{\text{neg}})(1 + \pi)} = 1, \quad (6)$$

since positive instances always appear in positive bags. On the other hand, for negative instances,

$$\begin{aligned} c(x_-) &= \frac{\int_\pi^1 P(x_- | B_\theta) P(B_\theta) d\theta}{P_{\text{neg}} + \frac{1}{2}(1 - P_{\text{neg}})(1 - \pi)} \\ &= \frac{\frac{1}{2}(1 - P_{\text{neg}})(1 - \pi)}{P_{\text{neg}} + \frac{1}{2}(1 - P_{\text{neg}})(1 - \pi)} \triangleq 1 - \gamma. \end{aligned} \quad (7)$$

The resulting values of $c(x)$ are shown in Figure 2(d). Note that for this generative process, except for the trivial case in which $P_{\text{neg}} = 0$, $1 - \gamma = c(x_-) < 1$, so $\gamma > 0$. Thus, the assumption that negative instances appear in negative bags is automatically satisfied for the example in Figure 2. By construction, this example also satisfies the $\pi > 0$ assumption since there is zero probability of sampling a bag with $\theta \in (0, \pi)$ mass over positive bags. Hence, this example is an element of MI-GEN.

2.3 The Empirical Bag-Labeling Function

In MI-GEN, the instance- and bag-labeling functions are defined independently at the level of the generative model, and must satisfy the relationships indicated in Definition 1.

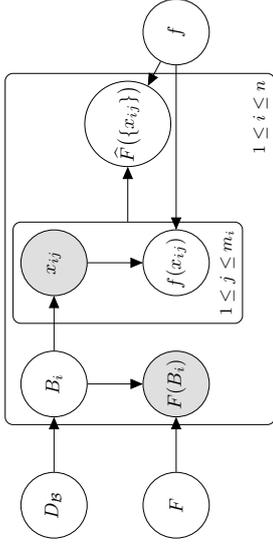


Figure 3: An illustration of the instance-, bag-, and empirical bag-labeling functions in MI-GEN.

However, in the standard MI setting, bag labels are typically viewed as being derived from instance labels. That is, if a bag is a set, then it is positive if it contains at least one positive instance, and negative otherwise.

Unlike the plate model in Figure 1(a), we do not typically observe bags directly in the MI setting. In the typical case, we only have access to samples $X_i = \{x_{ij}\}_{j=1}^{m_i}$, each drawn independently according to the distribution corresponding to each bag B_i , so that $\{\{x_{ij}\}_{j=1}^{m_i}, F(B_i)\}_{i=1}^n$ is the observed MI data set, as shown in Figure 1(c). Each bag can be a different size, but we will use $m_l \leq m_i \leq m_u$ to denote the lower and upper bounds on bag sizes, respectively.

If we think of “bags” (in the sense of the standard generative model) of instances $\{x_{ij}\}_{j=1}^{m_i}$ as empirical samples drawn from the underlying bag distributions B_i in our model, then it is possible that samples from positive bags do not contain any positive instances. Hence, such “bags” would be negative in the sense of the standard model. To more harmoniously account for the standard notion of bag labels within our model, we introduce the empirical bag-labeling function, $\widehat{F} : \mathcal{X}^* \rightarrow \{0, 1\}$:

$$\widehat{F}(X_i) = \max_j f(x_{ij}), \tag{8}$$

where $X_i = \{x_{ij}\}_{j=1}^{m_i}$ is any finite set of instances.

We can think of \widehat{F} as the bag labels that would be assigned by an oracle that had perfect information about the instance-labeling function f , but only an empirical sample from each bag. An illustration of the empirical bag-labeling function is shown in Figure 3. Figure 3 is a version of Figure 1(c) that shows the contributions of the instance-labeling and empirical bag-labeling functions. For every instance x_{ij} in an empirical bag sample, f assigns the label $f(x_{ij})$ to x_{ij} . On the other hand, \widehat{F} is a function of the entire bag sample $X_i = \{x_{ij}\}_{j=1}^{m_i}$ as well as the instance-labeling function f , as specified in Equation 8.

The labeling functions F and \widehat{F} will always agree on negative bags, since only negative instances are observed in negative bags. However, there might be some discrepancy between

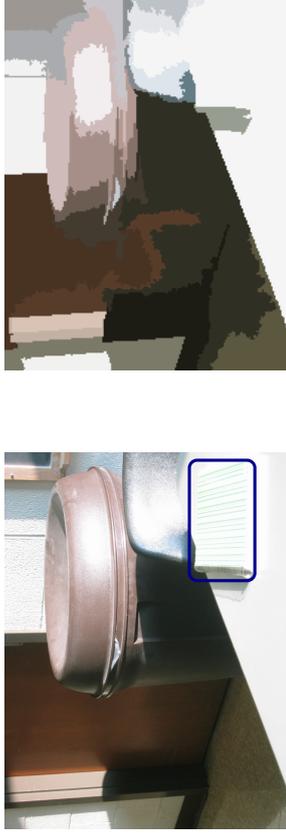


Figure 4: An example from the CBIR domain when a positive image does not contain a positive instance (the notebook, annotated in blue) after segmentation.

F and \widehat{F} on positive bags if only negative instances are sampled within a positive bag. We return to characterizing the discrepancy between F and \widehat{F} in Section 3.2.

The discrepancy between F and \widehat{F} is essentially bag-label noise that naturally results from our generative model. Previous generative models do not account for this potential source of label noise, despite its presence in some domains. For example, in the drug activity prediction domain, even if it is known that a molecule activates a receptor, a sample of conformations from this molecule might not contain the particular positive conformation that causes activation. Likewise, for the CBIR domain, extracting a set of objects from images is often performed using segmentation achieved through local optimization (Andrews et al., 2003; Carson et al., 2002). Therefore, it is possible that no single instance in a bag generated from a positive image will correspond to the positive instance. Figure 4 shows an example from the SIVAL data set when, due to lighting conditions, the positive “notebook” instance in the image is grouped with the table during segmentation (Settles et al., 2008). This kind of “noise” is naturally captured by our generative model as the discrepancy between \widehat{F} and F .

2.4 Relationship to Prior Models

The most general model in which instance learnability results have been previously shown is the “IID r -tuple” model (Blum and Kalai, 1998). The model, illustrated in Figure 5(a), assumes that each bag is generated by randomly sampling r instances in every bag from the same underlying instance distribution, D_X . However, this is an unrealistic assumption for many domains. For example, consider the drug activity prediction setting. In this domain, that would mean that the conformations of every molecule are sampled independently from the same distribution, which is not true as it requires that different molecules share the same conformations. Likewise, for CBIR, the IID assumption asserts that all segments in *all* images are sampled from the same distribution, when the distributions over objects/segments clearly change between images.

To show that our model is more general than the IID r -tuple model, we now describe how to simulate this model within our model. First, we define each bag to be a probability distribution parameterized by an r -tuple of instances $B_{(x_1, \dots, x_r)}$. This distribution will be a weighted sum of point masses over each of the r instances: $\mathbf{P}(x | B_{(x_1, \dots, x_r)}) = \frac{1}{r} \sum_{i=1}^r \delta_{x_i}(x)$. Then, for any distribution D_X over instances (with $\mathbf{P}(x)$ and instance-labeling function f), we let the distribution over bags D_B be defined as $\mathbf{P}(B_{(x_1, \dots, x_r)}) \triangleq \prod_{i=1}^r \mathbf{P}(x_i)$, which is the probability that the corresponding r -tuple would have been sampled from D_X , and the bag-labeling function F to be $F(B_{(x_1, \dots, x_r)}) = \max_{1 \leq i \leq r} f(x_i)$. Let $p_{\text{neg}} = \mathbf{P}[f(x) = 0]$, then we claim that the (D_X, D_B, f, F) described above is in ML-GEN $(\mu_{\text{neg}}^{\text{at}}, \frac{1}{r})$.

First, we need to show that D_B as defined satisfies Condition 1 of Definition 1:

$$\begin{aligned} \mathbf{P}(x) &\stackrel{?}{=} \int_B \mathbf{P}(x | B) d\mathbf{P}(B) \\ &= \int_B \frac{1}{r} \sum_{i=1}^r \delta_{x_i}(x) d\mathbf{P}(B_{(x_1, \dots, x_r)}) \\ &= \frac{1}{r} \sum_{i=1}^r \int_{\mathcal{X}} \dots \int_{\mathcal{X}} \delta_{x_i}(x) d\mathbf{P}(x_r) \dots d\mathbf{P}(x_1) \\ &= \frac{1}{r} \sum_{i=1}^r \left(\prod_{j \neq i} \int_{\mathcal{X}} d\mathbf{P}(x_j) \right) \left(\int_{\mathcal{X}} \delta_{x_i}(x) d\mathbf{P}(x_i) \right) \\ &= \frac{1}{r} \sum_{i=1}^r (1^{r-1}) \mathbf{P}(x) = \mathbf{P}(x). \end{aligned}$$

So sampling instances under our two-step generative process is equivalent to sampling according to the original instance distribution.

Condition 2 of Definition 1 is trivially satisfied, since by the definition of F , positive instances never appear in negative bags. To show that Condition 3 holds, we must compute the probability that negative instances appear in a negative bag. Using the definition of conditional probability, this is,

$$\mathbf{P}[F(B) = 0 | x] = \frac{\int_B \mathbf{P}(x | B) d\mathbf{P}(B)}{\mathbf{P}(x)}.$$

Using the fact that in a negative bag $B_{(x_1, \dots, x_r)}$, all instances must be negative, we can compute the numerator for a negative instance as

$$\begin{aligned} \int_B \mathbf{P}(x | B) d\mathbf{P}(B) &= \int_B \frac{1}{r} \sum_{i=1}^r \delta_{x_i}(x) d\mathbf{P}(B_{(x_1, \dots, x_r)}) \\ &= \frac{1}{r} \sum_{i=1}^r \int_{\mathcal{X}} \dots \int_{\mathcal{X}} \delta_{x_i}(x) d\mathbf{P}(x_r) \dots d\mathbf{P}(x_1) \\ &= \frac{1}{r} \sum_{i=1}^r \left(\prod_{j \neq i} \int_{\mathcal{X}} d\mathbf{P}(x_j) \right) \left(\int_{\mathcal{X}} \delta_{x_i}(x) d\mathbf{P}(x_i) \right) \\ &= \frac{1}{r} \sum_{i=1}^r (p_{\text{neg}}^{r-1}) \mathbf{P}(x) = p_{\text{neg}}^{r-1} \mathbf{P}(x). \end{aligned}$$

Thus, $\mathbf{P}[F(B) = 0 | x] = \frac{p_{\text{neg}}^{r-1} \mathbf{P}(x)}{\mathbf{P}(x)} = p_{\text{neg}}^{r-1}$. Since this probability is the same across all negative instances, this means that $\gamma = p_{\text{neg}}^{r-1}$. This quantity is positive as long as $p_{\text{neg}} > 0$. Otherwise, all instances are positive, so the $\gamma > 0$ assumption is vacuously satisfied.

Finally, to show that Condition 4 of Definition 1 is satisfied, we see that for a positive bag, B_i ,

$$\begin{aligned} \mathbf{P}[f(x) = 1 | B] &= \int_{\mathcal{X}} f(x) \left(\frac{1}{r} \sum_{i=1}^r \delta_{x_i}(x) \right) dx \\ &= \frac{1}{r} \sum_{i=1}^r \int_{\mathcal{X}} f(x_i) \geq \frac{1}{r} = \pi, \end{aligned} \tag{9}$$

since at least one instance in the bag is such that $f(x) = 1$. Therefore, the IID r -tuple model is a special case of our model in which γ and π are positive, and determined by the fraction of negative instances and bag size r .

Another generative model, used to show the learnability of bag-level concepts (Sabato and Tishby, 2012), allows arbitrary distributions over r -tuples. The model further relaxes the r -tuple model by allowing bag sizes to vary from 1 to R , some maximum bag size. The model is illustrated in Figure 5(b), where D_X^* denotes the distribution over tuples of size at most R . However, this model is also restrictive for many problem domains like drug activity prediction, since it enforces that bag sizes are finite and bounded, whereas molecules can exist in infinitely many conformations.

We can also represent the generative model of Sabato and Tishby (2012) in a similar way as for the IID r -tuple model. We simplify the space of bags to be atomic distributions over $r \leq R$ tuples, and allow an arbitrary distribution D_B over bags rather than requiring that $\mathbf{P}(B_{(x_1, \dots, x_r)}) = \prod_{i=1}^r \mathbf{P}(x_i)$. Now, D_X is not fixed, so we can define it in terms of Condition 1 of ML-GEN so that that condition is automatically satisfied. The bag-labeling function F is still defined in terms of the arbitrary instance-labeling function f , so Condition 2 is still trivially satisfied. Furthermore, by similar reasoning as in Equation 9, $\pi = \frac{1}{r}$ in this generative model, so Condition 4 is satisfied. However, the $\gamma > 0$ assumption (Condition 3) is no longer automatically satisfied by this generative process, since arbitrary distributions over tuples are allowed. Hence, while Sabato and Tishby (2012) analyze bag

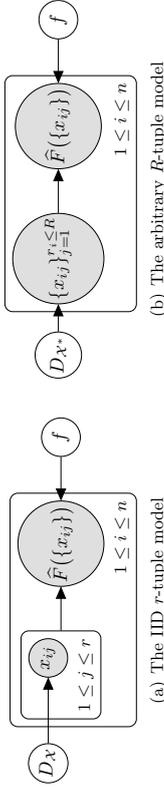


Figure 5: Previous generative models for MI data.

concept learnability with MI-GEN $(0, \frac{1}{R})$, we require MI-GEN $(\gamma, \frac{1}{R}) \subset$ MI-GEN $(0, \frac{1}{R})$ for instance concept learnability.

Babenko et al. (2011) propose treating bags in the MI setting as *manifolds* in the instance space \mathcal{X} . While this allows describing a bag with an infinite number of instances, it assigns an equal “weight” to every instance. However, for a domain like drug activity prediction, a molecule is more likely to exist in certain conformations than in others. The varying weight of instances is naturally handled in our setting, but we do not treat bags as manifolds over instances, so our results may not apply to the generative process in which bags are manifolds.

Some prior work proposes additional generative models for MIL, some of which model bags as distributions over instances. For example, some work uses Gaussian distributions (Maron and Lozano-Pérez, 1998; Xu, 2003) or mixture models (Foulds and Smyth, 2011) to represent distributions over instances, while other work uses more complex graphical (Yang et al., 2009; Adel et al., 2013; Kandemir and Hamprecht, 2014) or hierarchical Bayesian (Kueck and de Freitas, 2005) models. However, our work differs from these prior investigations in two important ways. First, we focus on the theoretical properties of our generative model, whereas prior work has only empirically explored the performance of algorithms tailored to specific generative models. Secondly, while prior generative models require that bags or instances are sampled from specific, parametric probability distributions, our generative model does not require such assumptions. Thus, the theoretical results presented below apply to more general scenarios than the models previously explored.

2.5 Applicability to Problem Domains

At the beginning of this section, we motivated MI-GEN using the 3D-QSAR domain. In this section, we elaborate on how bags can naturally be viewed as distributions in various other problem domains and which labeling tasks illustrated in Figure 3 are of interest in each domain. Of course, as described in Section 2.4, standard generative models are special cases of MI-GEN; so previous applications of MIL for which it is most natural to think of bags as finite sets of instances can still be incorporated in this model.

2.5.1 DRUG ACTIVITY PREDICTION

For the drug activity prediction or the 3D-QSAR problem, it is natural to think of each molecule as a distribution over conformations. While it is natural to view learning molecule-level activity F as the ultimate goal of 3D-QSAR, it is also important to learn the instance-

labeling function f . Knowing whether an individual conformation binds to a receptor provides information about the structure of the receptor’s binding site, which is practically difficult to measure directly. Hence, learning both instance- and bag-labeling functions are important in the 3D-QSAR domain.

2.5.2 TEXT CATEGORIZATION

While it is popular to represent documents as a flat “bag of words” using a single feature vector comprised of word frequencies (Salton and McGill, 1983), prior work has acknowledged the benefits of representing document-specific structure. In particular, latent Dirichlet allocation (LDA) models each document as a mixture of distributions over words (Blei et al., 2003). Of course, LDA can also be applied to a coarser-grained representation in which documents are distributions over n -grams or paragraphs, which are like individual instances in the MI setting (Blei et al., 2003). Other work attempts to infer the level of granularity in a document in addition to modeling the distributions over the discovered “segments” (Du et al., 2013). Hence, treating documents as distributions is already a natural and popular representation for text. On the other hand, LDA treats each document distribution as taking a specific parametric form, whereas our results and analysis do not make any parametric assumptions about bag-level distributions.

As for 3D-QSAR, both document-level and instance-level categorization is important in the text categorization domain. For example, if certain types of documents like survey articles discuss various subjects, then it might be important to determine not just that the document as a whole discusses a particular subject, but also which specific passage or paragraph discusses the subject.

2.5.3 CONTENT-BASED IMAGE RETRIEVAL

Applying our generative framework to the CBIR task requires viewing images as distributions over objects such that the objects in each image are a sample from the corresponding distribution. As with LDA for the text categorization domain, analogous probabilistic models have been proposed for categorizing natural scenes (Fei-Fei and Perona, 2005). Thus, while treating images as distributions is not unprecedented, our analysis is novel in that it discusses learnability under such a model without assuming that image distributions take a specific parametric form.

Furthermore, as for the other domains discussed, the bag-labeling function F is not the only latent variable of interest in CBIR. In addition to labeling new images, a CBIR system might be interested in determining the location of the object of interest within an image, which requires learning the instance-labeling function f .

3. Learning Accurate Concepts from MI Data

In this section, we describe new theoretical results that highlight the advantages of the generative model proposed in Section 2. In particular, the new generative model allows new results about instance- and bag-concept learnability that previously only held under a much stronger set of assumptions. As we describe in Section 4, additional theoretical results imply the surprising but testable ability of standard supervised approaches to learn

Table 1: A summary of the learnability results in Section 3 and Section 4.

	Accuracy		AUC
Instance	f	Theorem 1	Theorem 4
	\hat{F}	Theorem 2	Theorem 5
Bag	F	Theorem 3	Theorem 6

Table 2: Legend of the basic notation used in Section 3.

Symbol	Description/Definition
\mathcal{X}	Space of instances
\mathcal{B}	Space of bags (distributions over instances)
\mathcal{X}^*	Set of bag samples (sets of instances)
x_{ij}	Instance $x_{ij} \in \mathcal{X}$
B_i	Bag $B_i \in \mathcal{B}$
X_i	Bag sample $\{x_{ij}\}_{j=1}^{m_i} \in \mathcal{X}^*$, $x_{ij} \sim B_i$
m_i	Bag Sample Size $(m_i \leq X_i \leq m_u)$
f	Instance-Labeling Concept $m_i = X_i $
F	Bag-Labeling Concept
\hat{F}	Empirical Bag-Labeling Concept $\hat{F}(X_i) \triangleq \max_j f(x_{ij})$
g	Instance-Labeling Hypothesis
\hat{G}	Empirical Bag-Labeling Hypothesis $\hat{G}(X_i) \triangleq \max_j g(x_{ij})$
\mathcal{F}	Instance-Labeling Concept Class
$\text{VC}(\mathcal{F})$	Vapnik-Chervonenkis (VC) Dimension (Vapnik and Chervonenkis, 1971)

to rank instances and bags from MI data. Table 1 summarizes the theoretical contributions made in this and the following sections, which demonstrate the learnability of the instance concept f , empirical bag-labeling function \hat{F} , and bag-labeling function F with respect to both accuracy and ranking as measured by area under ROC (AUC). The results in this section and the following section use a model of the instance labeling function f to derive models for the bag-labeling functions \hat{F} and F .

Defining the ability of an algorithm to learn a good approximation of a target concept requires some metric by which the quality of the approximation is to be measured. Traditionally, the quality of a classifier is measured in terms of expected 0–1 loss. We begin by investigating the ability of algorithms to learn accurate concepts from MI data in this sense. While there is only one learning task in the supervised setting, there are now both instance- and bag-concept learning tasks in the MI setting, which we explore separately in the following sections. Table 2 shows the notation used for the concepts in this section.

3.1 Learning Accurate Instance Concepts

The probably approximately correctly (PAC) framework describes one sense in which it is possible to learn accurate concepts from supervised data. Since the generative process

described in Section 2 differs from that for supervised learning, we must restate what it means to “PAC” learn an accurate instance concept under this model.

In the supervised setting, the learnability of some fixed concept class \mathcal{F} is discussed without making any assumptions about the distribution over instances. The definition of MI-GEN in Definition 1 similarly allows any instance distribution, with which many bag distributions are consistent in the sense of Condition 1. To ensure that the target concept f is a member of the concept class \mathcal{F} , we must further restrict the set of models allowed by the generative process as follows:

Definition 2 (MI-GEN \mathcal{F}) For any $\gamma \in (0, 1]$ and $\pi \in [0, 1]$:

$$\text{MI-GEN}_{\mathcal{F}(\gamma, \pi)} \triangleq \{(D_X, D_B, f, F) \in \text{MI-GEN}(\gamma, \pi) : f \in \mathcal{F}\}.$$

Now, we can formally define PAC learnability for the MI setting:

Definition 3 (Instance MI PAC-learning) We say that an algorithm \mathcal{A} MI PAC-learns instance concept class \mathcal{F} from MI data when for any $(D_X, D_B, f, F) \in \text{MI-GEN}_{\mathcal{F}(\gamma)}$ with $\gamma > 0$, and $\epsilon_1, \delta > 0$, \mathcal{A} requires $O(\text{poly}(\frac{1}{\epsilon_1}, \frac{1}{\delta}))$ bag-labeled instances sampled independently from the MI generative process in Figure 1(b) to produce an instance hypothesis g with risk $R_I(g) < \epsilon_1$ with probability at least $1 - \delta$ over samples.²

Note that because our generative model allows us to discuss the marginalized instance distribution D_X , the risk $R_I(g) = \mathbb{E}_{x \sim D_X} [\mathbb{1}[f(x) \neq g(x)]]$ is measured with respect to this distribution as in the supervised setting. Now we show that instance concepts are MI PAC-learnable in the sense of Definition 3:

Theorem 1 An instance concept class \mathcal{F} with VC dimension $\text{VC}(\mathcal{F})$ is Instance MI PAC-learnable using $O\left(\frac{1}{\epsilon_1 \gamma} \left(\text{VC}(\mathcal{F}) \log \frac{1}{\epsilon_1 \gamma} + \log \frac{1}{\delta}\right)\right)$ examples.

Proof By Condition 1 in Definition 1, we can treat bag-labeled instances as being drawn from the underlying instance distribution D_X . Instances are observed with some label noise with respect to true labels given by f . Since positive instances never appear in negative bags (by Condition 2 of Definition 1), noise on instances is one-sided. If every negative instance appears in negative bags at least some γ fraction of the time (by Condition 3), then the maximum one-sided noise rate is $\eta = 1 - \gamma$. Since $\gamma > 0$, $\eta < 1$, which is required for learnability. Under our generative assumptions, the noise is “semi-random” in that noise rate might vary across instances, but is bounded by $\eta < 1$. Thus, learning an instance concept is equivalent to learning from data with one-sided label noise in this sense.

The result of Simon (2012) shows that in the presence of one-sided, semi-random noise, when a concept class \mathcal{F} has a VC dimension of $\text{VC}(\mathcal{F})$, \mathcal{F} is PAC-learnable from $O\left(\frac{1}{\epsilon_1(1-\eta)} \left(\text{VC}(\mathcal{F}) \log \frac{1}{\epsilon_1(1-\eta)} + \log \frac{1}{\delta}\right)\right)$ examples using a “minimum one-sided disagreement” strategy. This strategy entails choosing a classifier that minimizes the number of disagreements on positively-labeled examples while perfectly classifying all negatively-labeled examples. This strategy also works in the special case that all instances and bags are positive ($\eta = 0$, or $\gamma = 1$, since there are no negative instances). Substituting $1 - \gamma$ for η in the

² Alternate definitions of PAC-learnability require that \mathcal{A} also take at most polynomial time to produce hypothesis g . We defer the discussion of time complexity to Section 3.5.

bound of Simon (2012) yields the bound in terms of γ . ■

We note that Theorem 1 allows for “noisy” positive bags without positive instances ($\pi = 0$), since the additional bag-level noise is essentially absorbed into η .

3.2 Learning Accurate Bag Concepts

As for instance concept learnability, we must formally define what we mean to learn accurate bag concepts in the MI setting. As described in Section 2, there are two bag-labeling functions we might be interested in learning. In our generative model, we assume that the MI relationship between bag and instance labels holds at the level of the generative process. That is, bags are directly assigned labels by a bag concept F . On the other hand, given a set of instances sampled from a bag, we might be interested in learning the more traditional bag-labeling concept in the MI setting, $\widehat{F}(X_i) = \max_j f(x_{ij})$, which we have called the empirical bag-labeling function (Equation 8). We will first analyze learnability with respect to the empirical bag-labeling function and then extend this result to the true bag-labeling function.

We can define the risk of a bag-labeling concept \widehat{G} with respect to the underlying empirical bag-labeling concept \widehat{F} as follows:

$$R_{\widehat{F}}(\widehat{G}) = \mathbb{E}[\mathbb{1}[\widehat{F}(X) \neq \widehat{G}(X)]] \tag{10}$$

$$= \int_B \int_{\mathcal{X}^*} \mathbb{1}[\widehat{F}(X) \neq \widehat{G}(X)] d\mathbb{P}(X | B) d\mathbb{P}(B),$$

where $\mathbb{P}(X | B)$ is the probability of sampling the set of instances X from bag B . Since we assume that instances are sampled IID according to B , $\mathbb{P}(X | B) = \prod_{x \in X} \mathbb{P}(x | B)$. Given a formal definition of the risk of an empirical bag-labeling function, we can define learnability with respect to this function below:

Definition 4 (Empirical Bag MI PAC-learning) *We say that an algorithm \mathcal{A} MI PAC-learns empirical bag-labeling functions derived from instance concept class \mathcal{F} when for any $(D_{\mathcal{X}}, D_B, f, F) \in \text{MI-GEN}_{\mathcal{F}}(\gamma)$ with $\gamma > 0$, and $\epsilon_B, \delta > 0$, \mathcal{A} requires $O(\text{poly}(\frac{1}{\gamma}, \frac{1}{\epsilon_B}, \frac{1}{\delta}))$ bag-labeled instances sampled independently from the MI generative process in Figure 1(b) to produce an empirical bag-labeling function \widehat{G} with risk $R_{\widehat{F}}(\widehat{G}) < \epsilon_B$ with probability at least $1 - \delta$ over samples.*

To show empirical bag concept learnability under our generative model, we will show that by learning an accurate enough instance concept g , the resulting empirical bag-labeling concept given by $\widehat{G}(X_i) = \max_j g(x_{ij})$ will have low risk with respect to \widehat{F} . Thus, we start with a bound on $R_{\widehat{F}}(\widehat{G})$ in terms of $R_f(g)$.

Lemma 1 *Let $R_f(g)$ be the risk of an instance labeling concept g , and $R_{\widehat{F}}(\widehat{G})$ be the risk of the empirical bag-labeling function $\widehat{G}(X_i) = \max_j g(x_{ij})$. Then if bag sample sizes are bounded by m_o ($\forall i : |X_i| \leq m_o$), $R_{\widehat{F}}(\widehat{G}) \leq m_o R_f(g)$.* ■

Proof See Appendix A.

Given the bound demonstrated in Lemma 1, we can derive the following result:

Theorem 2 *Empirical bag-labeling functions derived from instance concept class \mathcal{F} with VC dimension $\text{VC}(\mathcal{F})$ are PAC-learnable from MI data using*

$$O\left(\frac{m_u}{\epsilon_B \gamma} \left(\text{VC}(\mathcal{F}) \log \frac{m_u}{\epsilon_B \gamma} + \log \frac{1}{\delta}\right)\right)$$

examples.

Proof The general strategy is to learn an approximation g for $f \in \mathcal{F}$ using minimum one-sided disagreement as mentioned in the proof of Theorem 1 and then to derive an empirical bag-labeling function \widehat{G} from g .

For a desired bound ϵ_B on $R_{\widehat{F}}(\widehat{G})$, by using $\epsilon_1 = \frac{\epsilon_B}{m_u}$ in Theorem 1, this ensures that the resulting instance classifier is such that $R_f(g) < \frac{\epsilon_B}{m_u}$ with high probability. Combined with the result in Lemma 1, this implies that $R_{\widehat{F}}(\widehat{G}) \leq m_u R_f(g) < m_u \left(\frac{\epsilon_B}{m_u}\right) = \epsilon_B$, so $R_{\widehat{F}}(\widehat{G}) < \epsilon_B$ as desired. Substituting $\epsilon_1 = \frac{\epsilon_B}{m_u}$ into the bound in Theorem 1 gives the bound as stated in Theorem 2. ■

Again, Theorem 2 allows for noisy positive bags without positive instances ($\pi = 0$). Furthermore, in the special case when every bag sample is a singleton $X = \{x\}$, then $m_u = 1$ and $\widehat{F}(\{x\}) = f(x)$. Thus, the instance concept learnability result in Theorem 1 is really just a special case of learning an empirical bag-labeling function with bags of size 1 as in Theorem 2.

Next, we turn our attention to learning the underlying bag-labeling function F . We will still use the instance-labeling function g to derive this bag-labeling function. It is possible to consider learning F without the use of an instance concept g , as we discuss in Section 3.3. During both training and testing, we are only given access to a sample X_i from each bag B_i with which we can estimate $F(B_i)$. Therefore, we will again learn an empirical bag labeling function $\widehat{G}(X_i)$. However, now we will assess the quality of \widehat{G} with respect to the underlying bag-labeling function F as follows:

$$R_F(\widehat{G}) = \mathbb{E}[\mathbb{1}[F(B) \neq \widehat{G}(X)]] \tag{11}$$

$$= \int_B \int_{\mathcal{X}^*} \mathbb{1}[F(B) \neq \widehat{G}(X)] d\mathbb{P}(X | B) d\mathbb{P}(B).$$

The definition of bag concept learnability then takes the same form as that in Definition 4 with the risk as given in Equation 11. As we will show in Lemma 2, we now also require the further assumption that π , minimum fraction of positive instances in positive bags, is nonzero.

Definition 5 (Bag MI PAC-learning) *We say that an algorithm \mathcal{A} MI PAC-learns bag-labeling functions derived from instance concept class \mathcal{F} when for any $(D_{\mathcal{X}}, D_B, f, F) \in \text{MI-GEN}_{\mathcal{F}}(\gamma, \pi)$ with $\gamma, \pi > 0$, and $\epsilon_B, \delta > 0$, algorithm \mathcal{A} requires $O(\text{poly}(\frac{1}{\gamma}, \frac{1}{\pi}, \frac{1}{\epsilon_B}, \frac{1}{\delta}))$ bag-labeled instances sampled independently from the MI generative process in Figure 1(b) to produce an empirical bag-labeling function \widehat{G} with risk $R_F(\widehat{G}) < \epsilon_B$ with probability at least $1 - \delta$ over samples.*

In order to show learnability of the bag-labeling concept F , we adopt a similar strategy as for Theorem 2 in which we first learn an instance-labeling concept g , then use g to derive

an empirical bag-labeling concept \widehat{G} . Since Theorem 2 shows that we can learn a concept G that accurately models F , what remains to be shown is that F is an accurate model of F under some additional conditions. First, we prove the following lemma, which decomposes the risk $R_F(\widehat{G})$ into the discrepancy between \widehat{G} and \widehat{F} , and the discrepancy between \widehat{F} and F .

Lemma 2 For any empirical bag-labeling concept \widehat{G} ,

$$R_F(\widehat{G}) \leq R_{\widehat{F}}(\widehat{G}) + R_F(\widehat{F}).$$

Proof See Appendix A. ■

Now, we derive a bound on the discrepancy between the empirical bag-labeling function \widehat{F} and the underlying bag-labeling function F . Since this discrepancy arises when we do not sample a positive instance within a positive bag, the bound depends on the minimum bag sample size and the minimum fraction π of positive instances in every positive bag.

Lemma 3 Suppose bag samples are of size at least m_i ($\forall i : m_i \leq |X_i|$), then $R_F(\widehat{F}) \leq (1 - \pi)^{m_i}$.

Proof See Appendix A. ■

Finally, we can now show the following learnability result with respect to the underlying bag-labeling function. However, note that in Lemma 3, the error $R_F(\widehat{F})$ decreases with the minimum bag size m_i . Thus, in order to achieve low error with respect to F , we must ensure that bags in the test set are sufficiently large. Therefore, the following result is stated under the additional condition that the test bag sample sizes m_i satisfy some constraints. Note that these constraints arise naturally from the process that samples instances from bag distributions.

Theorem 3 Bag-labeling functions derived from instance concept class \mathcal{F} with VC dimension $VC(\mathcal{F})$ are PAC-learnable from MI data using

$$O\left(\frac{1}{\epsilon_B^2 \pi} \left(VC(\mathcal{F}) \log \frac{1}{\epsilon_B \pi} + \log \frac{1}{\delta}\right)\right) \tag{12}$$

examples when test bag sample sizes are bounded by $m_i \leq m \leq m_u$ and m_i is large enough such that $m_i \geq \frac{1}{\pi} \log \frac{2}{\epsilon_B}$.

Proof Intuitively, we can learn an instance-labeling function g according to Theorem 1 and then use the resulting empirical bag-labeling function \widehat{G} . By combining the previously stated results, we can bound $R_F(\widehat{G})$ as

$$\begin{aligned} R_F(\widehat{G}) &\leq R_{\widehat{F}}(\widehat{G}) + R_F(\widehat{F}) && \text{(by Lemma 2)} \\ &\leq m_u R_g(g) + R_F(\widehat{F}) && \text{(by Lemma 1)} \\ &\leq m_u R_g(g) + (1 - \pi)^{m_i}. && \text{(by Lemma 3)} \end{aligned}$$

In the case that $\pi = 1$, then the second term in the sum is zero. Otherwise, suppose the minimum bag size is such that

$$m_i \geq \frac{1}{\pi} \log \frac{2}{\epsilon_B} \geq \frac{\log \frac{\epsilon_B}{2}}{\log(1 - \pi)} = \log_{1 - \pi} \frac{\epsilon_B}{2},$$

where the second inequality follows from the fact that $\pi \leq -\log(1 - \pi)$ for $\pi \in (0, 1)$. Therefore, since $(1 - \pi) < 1$, we have that

$$(1 - \pi)^{m_i} \leq (1 - \pi)^{\log_{1 - \pi} \frac{\epsilon_B}{2}} = \frac{\epsilon_B}{2}.$$

Furthermore, when learning the instance concept g , we can choose ϵ_l to be such that $\epsilon_l = \frac{\epsilon_B}{2m_u}$. Since g will be such that $R_g(g) < \epsilon_l$ with probability $(1 - \delta)$, with the same probability we have that

$$\begin{aligned} R_F(\widehat{G}) &\leq m_u R_g(g) + (1 - \pi)^{m_i} \\ &< m_u \left(\frac{\epsilon_B}{2m_u}\right) + \frac{\epsilon_B}{2} = \epsilon_B. \end{aligned}$$

Substituting the expression for ϵ_l in terms of ϵ_B into the bound in Theorem 1 gives the sample complexity bound:

$$O\left(\frac{m_u}{\epsilon_B^2 \gamma} \left(VC(\mathcal{F}) \log \frac{m_u}{\epsilon_B \gamma} + \log \frac{1}{\delta}\right)\right),$$

which is the same bound as stated in Theorem 2.

It is also possible to sub-sample large bags such that there is also a conservative upper bound on sample size $m_u = O\left(\frac{1}{\epsilon_B \pi}\right)$, which is consistent with $m_i \geq \frac{1}{\pi} \log \frac{2}{\epsilon_B}$. Then, we can derive an expression for learnability in terms of π : Substituting this bound into that of Theorem 2 gives the second sample complexity bound as stated in Equation 12. ■

3.3 Discussion

The results presented in Section 3.1 and Section 3.2 follow the same basic strategy. First, minimum one-sided disagreement is used to learn an accurate instance concept g in the presence of one-sided noise on bag-labeled instances. Then, for the bag-labeling task, instance labels are aggregated using an empirical bag-labeling function \widehat{G} to approximate the empirical bag-labeling function \widehat{F} or the underlying bag-labeling function F . The idea of combining instance labels to produce a bag-labeling function is used by many existing MI algorithms.

However, under the generative model that treats bags as distributions, the bag-labeling results derived in Section 3.2 are somewhat counterintuitive. On the one hand, if bags are distributions from which we observe samples, then the larger the samples, the more information an algorithm has about the underlying bag distribution. Intuitively, it seems that the better an algorithm can estimate the underlying bag distribution, which is the object of interest for classification, the better it can learn a concept to label new bags. On

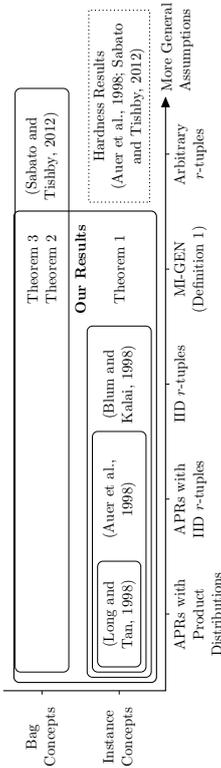


Figure 6: Relation to prior learnability results.

the other hand, the result in Theorem 2 suggests that it is harder to learn from larger bag sizes, since roughly $O(m_n \log m_n)$ more examples are required to learn an accurate concept.

Essentially, the source of this incoherence in reasoning is the use of an instance-labeling concept g to derive the bag-labeling concept \widehat{G} . In the process of combining instance labels to label a bag, small errors in the instance labeling function g compound quickly. For example, g must label *all* instances in a negative bag correctly for \widehat{G} to label the bag correctly. As bag size increases, it becomes less likely that g will agree with f across all instances.

Therefore, despite our positive results suggesting that learning an accurate instance-labeling function is sufficient to learn an accurate bag-labeling function, in practice, it is possible to imagine an alternative approach in which bag-labeling functions are learned directly by representing bags in a supervised fashion. Several practical approaches, such as bag-level kernels (Gärtner et al., 2002; Zhou et al., 2009) and embeddings (Chen et al., 2006; Foulds, 2008) do precisely this, and essentially turn the bag-label learning problem into a supervised learning problem. Further investigating the trade-offs between techniques that classify bags using instance classifiers and those that directly learn bag classifiers is an interesting direction for future work.

3.4 Relation to Prior Learnability Results

An overview of our results in the context of prior work is shown in Figure 6. Early work on instance learnability shows that axis-parallel rectangles (APRs) are learnable from MI data, but under the restrictive assumption that each bag contains r IID instances sampled from a product distribution (Long and Tan, 1998). Later work by Auer et al. (1998) extends these results to the case when the instance distribution is no longer a product distribution, but the instances are still sampled IID from a single distribution across bags. The most recent results on instance concept learnability in the MI setting are described by Blum and Kalai (1998). Like the proof of Theorem 1, Blum and Kalai (1998) also reduce the problem of learning instance concepts to learning from noisy examples. However, the proof in Blum and Kalai (1998) requires that the label noise on negative examples be uniformly random. This condition is met under the strong assumption made in that work, that instances in all bags are drawn IID from the same distribution over instances. On the other hand, the result in Theorem 1 applies to our more general model in which the noise rate can vary

across instances. Hence, our results rely on the recent work of Simon (2012), which shows that it is possible to learn from instances corrupted with semi-random one-sided noise.

The bag learnability results in Section 3.2 show that an accurate bag concept can be learned by learning an accurate instance concept and deriving a bag concept by combining instance labels within a bag. Other recent work on bag concept learnability takes a different approach. The strategy of Sabato and Tishby (2012) is to directly learn empirical bag-labeling concepts using empirical risk minimization (ERM). That is, they suppose that an algorithm selects an instance-labeling function $g \in \mathcal{F}$ that minimizes $R_{\widehat{F}}(\widehat{G})$. Since general sample complexity bounds exist for ERM in terms of capacity measures such as VC dimension of a hypothesis class (Blumer et al., 1989), Sabato and Tishby (2012) proceed by proving that the capacity of the function class $\{\widehat{G} : \widehat{G}(X_i) = \max_j g(x_{ij}), g \in \mathcal{F}\}$ is bounded in terms of the capacity of \mathcal{F} . In fact, the results of Sabato and Tishby (2012) apply to more general cases in which the combining function used to derive a bag-labeling function from an instance-labeling function is other than the max function. However, the results of Sabato and Tishby (2012) do not prove positive results about instance-labeling concepts.

As indicated in Figure 6, the results in Section 3.2 are not a strict generalization of those in Sabato and Tishby (2012), nor are those in Sabato and Tishby (2012) a generalization of those in Section 3.2. In particular, since MI-GEN treats bags as distributions, the results in Section 3.2 apply to cases not considered in Sabato and Tishby (2012), in which bags are assumed to have finite size. On the other hand, while our generative model can encapsulate aspects of the generative model in Sabato and Tishby (2012) (see Section 2.4), arbitrary distributions over r -tuples are not permitted. However, it may be that we are able to prove positive instance learnability results precisely *because* of this constraint on the generative process.

Other recent work has discussed the difficulty, both theoretically and in practice, to relate the performance of the same classifier on the instance- and bag-labeling tasks (Tragante et al., 2011). In contrast, Lemma 1 illustrates a clear connection between the accuracy of an instance concept and that of the resulting empirical bag concept. This new connection is made possible by the relationship between bag and instance distributions in our generative model, as highlighted in Lemma 1. In particular, Condition 1 of Definition 1 is employed to marginalize out the effect of individual bag distributions so that error on bags can be expressed directly in terms of the error on instances. There are at least two reasons why this relationship is not obvious from empirical results. First, contrasting the sample complexity expressions Theorem 1 and Theorem 3, we see that larger samples are required to learn accurate bag concepts. Thus, for a fixed sample size, there might be a significant discrepancy in performance of a single algorithm on the two learning tasks. Secondly, the relationship we demonstrate holds when an accurate instance concept is learned and *then* applied to the bag-labeling task. However, in practice, many algorithms attempt to learn an instance function to label bags using ERM at the *bag-level*. It is not clear that the instance-labeling function found via bag-level ERM in this way will successfully label instances.

3.5 Relation to Prior Hardness Results

The positive learnability results in Section 3.1 and Section 3.2 do not contradict existing hardness results about learning in the MI setting. Essentially, most hardness results are shown under the scenarios that lie on the far right of Figure 6. For example, Sabato and Tishby (2012) observe that if only positive bags are generated, then learning the bag-labeling function is trivial, but no label information about instances is provided. In this case, learning instance labels is equivalent to learning in the *unsupervised learning* setting, for which no PAC-style guarantees can be made. However, the additional assumptions in MI-GEN preclude the case when only positive bags appear, since the negative instances would never appear in negative bags as required by Condition 3 in Definition 1.

Similarly, under the weak assumption in which arbitrary distributions over r -tuples are allowed, Auer et al. (1998) show that that efficiently PAC-learning MI instance concepts is impossible (unless $\text{NP} = \text{RP}^?$). While the results on instance and bag learnability stemming from Theorem 1 show that a polynomial number of examples can be used to learn accurate concepts, they do not bound the computational complexity of learning from the examples. In particular, minimum one-sided disagreement is known to be NP-hard for certain concept classes and loss functions (Simon, 2012). Therefore, for some concept classes, instance and bag concepts are not *efficiently* PAC-learnable: learnable with a polynomial number of examples *in polynomial time*.

The apparent contradiction between our learnability results and the hardness results of Auer et al. (1998) is resolved by observing that MI-GEN precludes the scenario used to reduce learning disjunctive normal form (DNF) formulae to learning APPRs from MI data. In the reduction used by Auer et al. (1998), each instance corresponds to a (variable assignment, clause) pair, and a bag is formed for each variable assignment by including a pair with that variable assignment for each clause. Bags are sampled uniformly over all variable assignments. Suppose a particular variable assignment v satisfies the first clause c_1 , but not the second clause c_2 . Then the instance (v, c_1) is positive, but (v, c_2) is negative. However, (v, c_2) only ever appears in bags along with (v, c_1) ; that is, in positive bags. This violates the condition that $\gamma > 0$, or that negative instances appear with some probability in negative bags, so our results do not apply to this hard scenario.

Similarly, our generative model precludes scenarios used to show the hardness of learning hyperplane concepts for MI data (Kumudacioglu et al., 2010; Dicochos et al., 2012; Doran and Ray, 2013). It is unknown whether there is an algorithm to efficiently learn hyperplanes that minimize one-sided disagreement. However, even ERM under 0–1 loss is NP-hard for the concept class of hyperplanes (Ben-David et al., 2003), which are widely used in practice for supervised learning. Thus, while previous results have characterized the hardness of MI learning as resulting from arbitrary distributions across bags, our results suggest that the hardness arises from cases in which $\gamma = 0$, or when negative instances only occur in negative bags.

3. **RP** is the class of decision problems for which a probabilistic Turing machine terminates in polynomial time, always returns **NO** when the answer is **NO**, and returns **YES** with probability at least $\frac{1}{2}$ when the answer is **YES**.

3.6 Must Instances be Dependent Samples?

As observed in prior work, most real-world examples of MIIL have bags that contain non-IID instances (Zhou et al., 2009). Thus, our assumption that bag samples X_i are drawn IID according to their corresponding bag distributions B_i might seem unrealistic. However, note that our generative model *does* allow for dependencies between instances at the level of bag distributions, B_i . That is, although the samples X_i are drawn from bag distributions independently, we can use such independent samples to *approximate* the behavior of empirical bag-labeling functions on *nonindependent* samples.

The arbitrary R -tuple model, as illustrated in Figure 5(b), allows for arbitrary distributions over tuples of size at most R , which can be used to represent any generative model in which there is a relationship between instances in bags (i.e., bags in which instances are non-IID). As described in Section 2.4, it is possible to represent this model within MI-GEN where each bag is an atomic distribution over the instances in the tuple and the distribution over bags corresponds to the original distribution over tuples. Given this representation, $\pi = \frac{1}{R}$ in our model. In the traditional MI setting, we would directly observe these R instances. Our generative model, on the other hand, assumes that we perform the equivalent of repeatedly sampling an instance from these R instances uniformly and independently at random. In this case, we have the following result:

Corollary 1 (MI PAC-learning from Dependent Instances) *When distributions of instances in bags are defined by a set of R dependent instances sampled from a distribution over R tuples, bag-labeling functions derived from instance concept class \mathcal{F} with VC dimension $\text{VC}(\mathcal{F})$ are PAC-learnable from MI data using*

$$O\left(\frac{R}{\epsilon_B \gamma} \log \frac{1}{\epsilon_B} \left(\text{VC}(\mathcal{F}) \log \frac{R}{\epsilon_B \gamma} + \log \frac{1}{\delta}\right)\right)$$

examples with test bags of size $m = \lceil R \log \frac{2}{\epsilon_B} \rceil$ drawn independently with replacement from the R dependent instances.

Proof Following the same line of reasoning as in Theorem 3, we can derive the sample complexity bound

$$O\left(\frac{m}{\epsilon_B \gamma} \left(\text{VC}(\mathcal{F}) \log \frac{m}{\epsilon_B \gamma} + \log \frac{1}{\delta}\right)\right), \quad (13)$$

when there are m instances per bag. Choosing $m = \lceil R \log \frac{2}{\epsilon_B} \rceil$ satisfies the conditions of that theorem, since $\pi = \frac{1}{R}$. Substituting m into Equation 13 implies the sample complexity as stated in the corollary. ■

Thus, even though the bag distribution is representable using only R dependent instances, when sampling independently, we must sample a factor of $O\left(\log \frac{1}{\epsilon_B}\right)$ more instances to ensure that we can learn an accurate bag-labeling concept with high probability.

4. Learning to Rank from MI Data

Learnability results are often stated as in Section 3 with respect to the accuracy metric. However, other metrics often provide a more useful characterization of algorithm performance in practice. For example, for the 3D-QSAR problem, it is not necessary to accurately

Table 3: Legend of the basic notation used in Section 4.

Symbol	Description/Definition
\mathcal{X}	Space of instances
\mathcal{B}	Space of bags (distributions over instances)
\mathcal{X}^*	Set of bag samples (sets of instances)
x_{ij}	Instance $x_{ij} \in \mathcal{X}$
B_i	Bag $B_i \in \mathcal{B}$
X_i	Bag sample $\{x_{ij}\}_{j=1}^{m_i} \in \mathcal{X}^*$, $x_{ij} \sim B_i$ ($m_i \leq X_i \leq m_u$)
m_i	Bag Sample Size $m_i = X_i $
c	p -concept for bag-labeled instances $c(x) \triangleq \mathbb{P}[F(B) = 1 x]$
h	Instance-Labeling p -concept
\hat{H}	Empirical Bag-Labeling p -concept $\hat{H}(X_i) \triangleq \max_j h(x_{ij})$
p_{neg}, p	$p_{\text{neg}} \triangleq \mathbb{P}[f(x) = 0]$ $p \triangleq \min\{p_{\text{neg}}, 1 - p_{\text{neg}}\}$
$\hat{P}_{\text{neg}}, \hat{P}$	$\hat{P}_{\text{neg}} \triangleq \mathbb{P}[\hat{F}(X) = 0]$ $\hat{P} \triangleq \min\{\hat{P}_{\text{neg}}, 1 - \hat{P}_{\text{neg}}\}$
P_{neg}, P	$P_{\text{neg}} \triangleq \mathbb{P}[F(B) = 0]$ $P \triangleq \min\{P_{\text{neg}}, 1 - P_{\text{neg}}\}$
\mathcal{C}	Instance-Labeling p -concept Class
$\text{PD}(\mathcal{C})$	Pseudo-Dimension of \mathcal{C} (Haussler, 1992)

predict the activity of every molecule. Instead, a classifier can produce a ranked list indicating its confidence that each molecule is active. The set of active molecules with the highest predicted activity can then be investigated further by chemists. Unlike the prior work on learning accurate concepts from MI data as shown in Figure 6, there has been virtually no prior work on learning to rank in the MI setting. That is, although ranking algorithms have been developed for MIL (Bergeron et al., 2008), there is no formal analysis of the performance of such approaches. We provide such an analysis in this section.

In the 3D-QSAR example, a desirable property of a classifier is that it appropriately *rank*s bags or instances. That is, it assigns a higher real-valued confidence that a conformation is positive to actual positive conformations than to negative conformations. The AUC metric is commonly used to measure the ranking performance of a classifier. We show in this section that classifiers with high AUC are also learnable from MI data under our generative model. Furthermore, we show that learning high-AUC concepts from MI data is easier than learning accurate concepts in the sense that it can be achieved using standard ERM approaches. This suggests that standard supervised algorithms can learn high-AUC concepts from MI data generated according to MI-GEN, a surprising hypothesis that we evaluate in the final section.

4.1 Learning High-AUC Instance Concepts

Prior work has shown that the AUC is equivalent to the probability that a randomly selected positive example will be assigned a higher confidence than a randomly selected negative example (Hanley and McNeil, 1982). We can define a corresponding instance AUC error of a real-valued hypothesis h as $1 - \text{AUC}$, or the probability that a negative instance is assigned a higher confidence than a positive instance:

$$\begin{aligned} R_f^{\text{AUC}}(h) &= \int_{\mathcal{X}} \int_{\mathcal{X}} \mathbb{1}[h(x_-) > h(x_+)] d\mathbb{P}(x_+ | f(x_+) = 1) d\mathbb{P}(x_- | f(x_-) = 0) \\ &= \frac{\int_{\mathcal{X}_-} \int_{\mathcal{X}_+} \mathbb{1}[h(x_-) > h(x_+)] d\mathbb{P}(x_+) d\mathbb{P}(x_-)}{\mathbb{P}[f(x) = 1] \mathbb{P}[f(x) = 0]} \\ &= \frac{1}{(1 - p_{\text{neg}}) p_{\text{neg}}} \int_{\mathcal{X}_-} \int_{\mathcal{X}_+} \mathbb{1}[h(x_-) > h(x_+)] d\mathbb{P}(x_+) d\mathbb{P}(x_-). \end{aligned} \quad (14)$$

The first step follows from the definition of conditional probability, and we introduce $p_{\text{neg}} = \mathbb{P}[f(x) = 0]$ for notational convenience (see Table 3 for a list of notation used in this section). By definition, this quantity is zero in the cases when either all instances are positive or all instances are negative.

Given the formal definition of AUC, we can begin to describe how it is possible to learn high-AUC instance concepts from MI data. Since a classifier’s confidence values are relevant for the AUC metric, we will consider the hypothesis class corresponding to a classifier to be a p -concept class \mathcal{C} . The p -concept model is a model for binary classification in which a p -concept $c: \mathcal{X} \rightarrow [0, 1]$ represents the probability that an instance x is observed with a positive label (Kearns and Schapire, 1994). For high-AUC instance learnability, we will show that it is sufficient to learn a p -concept $h \in \mathcal{C}$ that models the p -concept $c(x) = \mathbb{P}[F(B) = 1 | x]$, the probability of observing instance x in a positive bag as defined in Equation 2.

To ensure that the target concept c is also a member of \mathcal{C} , we must formally restrict the set of bag labeling functions and distributions that are permitted by the generative model as follows:

Definition 6 (MI-GEN \mathcal{C}) For any $\gamma \in (0, 1]$ and $\pi \in [0, 1]$:

$$\text{MI-GEN}_{\mathcal{C}}(\gamma, \pi) \triangleq \left\{ (D_{\mathcal{X}}, D_{\mathcal{B}}, f, F) \in \text{MI-GEN}(\gamma, \pi) : (x \mapsto \mathbb{P}[F(B) = 1 | x]) \in \mathcal{C} \right\}.$$

Learnability of a p -concept with high AUC is then defined with respect to p -concept class \mathcal{C} :

Definition 7 (Instance MI AUC-PAC-learning) We say that an algorithm \mathcal{A} MI AUC-PAC-learns instance p -concept class \mathcal{C} from MI data when for any $(D_{\mathcal{X}}, D_{\mathcal{B}}, f, F) \in \text{MI-GEN}_{\mathcal{C}}(\gamma)$ with $\gamma > 0$, and $\epsilon_1, \delta > 0$, algorithm \mathcal{A} requires $O(\text{poly}(\frac{1}{\epsilon_1}, \frac{1}{\delta}))$ bag-labeled instances sampled independently from the MI generative process in Figure 1(b) to produce an instance p -concept hypothesis h with risk $R_f^{\text{AUC}}(h) < \epsilon_1$ with probability at least $1 - \delta$ over samples.

Whereas learning accurate instance concepts as in Definition 3 required the use of minimum one-sided disagreement, we show in Theorem 4 that it is possible to learn high-AUC

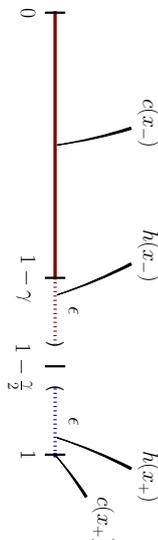


Figure 7: The intuition behind Theorem 4. A hypothesis h that closely approximates c will correctly rank instances with high probability.

concepts using ERM. In particular, the strategy used in the following theorem is to learn a p -concept h that models the concept c defined in Equation 2 using standard ERM. The intuition is that c already achieves perfect AUC; that is, $R_{\text{AUC}}^{\text{AUC}}(c) = 0$. The reason is that for any negative instance x_- and positive instance x_+ , $c(x_-) \leq 1 - \gamma < 1 = c(x_+)$; see Figure 7 for an illustration. If we learn a p -concept h that closely approximates c to within some ϵ , then with high probability, h will also correctly rank instances.

Stating the learnability of a p -concept with ERM requires use of the pseudo-dimension of the concept class \mathcal{C} ; just as VC dimension can be used to characterize the capacity of a deterministic concept class. The pseudo-dimension is similar to the VC dimension, but uses a different notion of “shattering.” In particular, for a set of points with real-valued labels, $\{(x_i, y_i)\}_{i=1}^n$, a p -concept class \mathcal{C} shatters the points if for any binary labeling of the points $\{b_i\}$, there exists some $c \in \mathcal{C}$ such that $c(x_i) \geq y_i$ if $b_i = 1$ and $c(x_i) < y_i$ if $b_i = 0$ (Haussler, 1992). The pseudo-dimension of \mathcal{C} , denoted $\text{PD}(\mathcal{C})$, is the size of the largest set such that \mathcal{C} shatters some set of that size.

Theorem 4 *An instance p -concept class \mathcal{C} with pseudo-dimension $\text{PD}(\mathcal{C})$ is Instance MI AUC-PAC-learnable using $O\left(\frac{1}{(\epsilon\gamma)^p} \left(\text{PD}(\mathcal{C}) \log \frac{1}{\epsilon\gamma p} + \log \frac{1}{\delta}\right)\right)$ examples with standard ERM approaches, where $p = \min\{P_{\text{neg}}, 1 - P_{\text{neg}}\}$.*

Proof See Appendix A. ■

Comparing Theorem 4 with Theorem 1 on learning accurate instance concepts, we see that neither results require that positive instances appear in positive bags ($\pi > 0$). In both cases, the addition label noise affects γ , but is tolerated by the underlying algorithm. The key difference between these results is that high-AUC concepts can be learned via standard ERM approaches, whereas accurate concept learning requires minimum one-sided disagreement. Additionally, the sample complexity bound in Theorem 4 contains an additional factor p that accounts for class imbalance. Intuitively, this factor appears because it is difficult to learn to effectively rank instances from different classes when one class appears very infrequently in the training set (p is small).

4.2 Learning High-AUC Bag Concepts

As for accuracy, we might be interested in learning either high-AUC instance *or* bag concepts from MI data. Following a similar strategy as employed in Section 3.2 for learning accurate bag concepts, here we will consider two measures of bag-level performance of a bag concept \hat{H} derived from an instance concept h . The same combining function as in Section 3, $\hat{H}(X_i) = \max_j h(x_{ij})$, is commonly used to derive real-valued bag-labeling functions in prior work (Ray and Craven, 2005). Following the analysis in Section 3.2, we will measure performance of \hat{H} with respect to both \hat{F} , the empirical bag-labeling function, and later F , the underlying bag-labeling function.

For the empirical bag-labeling function, \hat{F} , the intuitive definition of AUC is the probability that a bag-level hypothesis \hat{H} assigns a higher value to a bag sample given that it is labeled positive by \hat{F} (that is, containing at least one positive instance) than another bag sample labeled negative by \hat{F} (containing no positive instances). Formally, we can define the corresponding AUC-based risk as follows:

$$\begin{aligned} R_{\hat{F}}^{\text{AUC}}(\hat{H}) &= \frac{\int_{\mathcal{B}} \int_{\mathcal{B}} \int_{\mathcal{X}_+} \int_{\mathcal{X}_+} \mathbf{1}[\hat{H}(X_-) > \hat{H}(X_+)] \dots \\ &\quad \dots d\mathbb{P}(X_+ | B_+) d\mathbb{P}(X_- | B_-) d\mathbb{P}(B_+) d\mathbb{P}(B_-)}{\mathbb{P}[\hat{F}(X) = 1] \mathbb{P}[\hat{F}(X) = 0]} \\ &= \frac{\int_{\mathcal{B}} \int_{\mathcal{B}} \int_{\mathcal{X}_+} \int_{\mathcal{X}_+} \mathbf{1}[\hat{H}(X_-) > \hat{H}(X_+)] \dots \\ &\quad \dots d\mathbb{P}(X_+ | B_+) d\mathbb{P}(X_- | B_-) d\mathbb{P}(B_+) d\mathbb{P}(B_-)}{(1 - \hat{P}_{\text{neg}}) \hat{P}_{\text{neg}}}. \end{aligned} \quad (15)$$

Above, \mathcal{X}_+^* is the set of all negative bag samples, and \mathcal{X}_+^* the set of all positive bag samples. The notation $\hat{P}_{\text{neg}} = \Pr[\hat{F}(X) = 0]$ is used for convenience. Now, we can define learnability with respect to this metric:

Definition 8 (Empirical Bag MI AUC-PAC-learning) *We say that an algorithm A MI AUC-PAC-learns empirical bag-labeling functions derived from p -concept class \mathcal{C} when for any $(D_{\mathcal{X}}, D_{\mathcal{B}}, f, F) \in \text{MI-GEN}(\gamma)$ with $\gamma > 0$, and $\epsilon_{\mathcal{B}}, \delta > 0$, algorithm A requires $O(\text{poly}(\frac{1}{\gamma}, \frac{1}{\epsilon_{\mathcal{B}}}, \frac{1}{\delta}))$ bag-labeled instances sampled independently from the MI generative process in Figure 1(b) to produce an empirical bag-labeling function \hat{H} with risk $R_{\hat{F}}^{\text{AUC}}(\hat{H}) < \epsilon_{\mathcal{B}}$ with probability at least $1 - \delta$ over samples.*

We will now show learnability of empirical bag-labeling functions by reducing the problem to learning an accurate model of the p -concept c . Hence, the approach of the proof follows that for learning accurate empirical bag-labeling functions.

Theorem 5 *Empirical bag-labeling functions derived from p -concept class \mathcal{C} with pseudo-dimension $\text{PD}(\mathcal{C})$ are AUC-PAC-learnable from MI data using*

$$O\left(\frac{m_{\mathcal{B}}^4}{(\epsilon_{\mathcal{B}}\gamma)^4} \left(\text{PD}(\mathcal{C}) \log \frac{m_{\mathcal{B}}}{\epsilon_{\mathcal{B}}\gamma} + \log \frac{1}{\delta}\right)\right)$$

examples with standard ERM approaches, where

$$\hat{P} \triangleq \min\{\hat{P}_{\text{neg}}, 1 - \hat{P}_{\text{neg}}\} \geq \min\{P_{\text{neg}}, 1 - P_{\text{neg}}\},$$

and $m_{\mathcal{B}}$ is an upper bound on bag sample size.

Proof See Appendix A. ■

Note that Theorem 4 is a special case of Theorem 5 when $m_u = 1$. In this case $\widehat{P}_{\text{neg}} = p_{\text{neg}}$ when samples all have size 1, so $\widehat{P} = p$ and the sample complexity is the same.

Now, we can examine AUC-learnability with respect to the true bag-labeling function, F . To define AUC with respect to F , we measure the probability that a sample X_+ is labeled higher by \widehat{H} than X_- is, given that X_+ is sampled from a positive bag and X_- is sampled from a negative bag. Formally, the AUC risk of \widehat{H} with respect to F is

$$\begin{aligned} R_F^{\text{AUC}}(\widehat{H}) &= \frac{\int_{\mathcal{B}_-} \int_{\mathcal{B}_+} \int_{\mathcal{X}^*} \mathbb{1}[\widehat{H}(X_-) > \widehat{H}(X_+)] \dots}{\dots \text{dP}(X_+ | B_+) \text{dP}(X_- | B_-) \text{dP}(B_+) \text{dP}(B_-)} \dots \\ &= \frac{\text{P}[F(B) = 1] \text{P}[F(B) = 0]}{\int_{\mathcal{B}_-} \int_{\mathcal{B}_+} \int_{\mathcal{X}^*} \mathbb{1}[\widehat{H}(X_-) > \widehat{H}(X_+)] \dots \dots \text{dP}(X_+ | B_+) \text{dP}(X_- | B_-) \text{dP}(B_+) \text{dP}(B_-)} \dots \end{aligned} \quad (16)$$

The notation $P_{\text{neg}} = \text{P}[F(B) = 0]$ is used to denote the probability of sampling a negative bag from the distribution over bags. We define the risk to be zero in the case that this probability is equal to either 0 or 1.

As for accuracy, the risk of an empirical bag-labeling function now depends on how representative a sample is of the underlying bag. Thus, in the definition of AUC-learnability with respect to F (Definition 9), we now again require an additional assumption that positive instances appear some $\pi > 0$ fraction of the time in positive bags.

Definition 9 (Bag MI AUC-PAC-learning) *We say that an algorithm \mathcal{A} MI AUC-PAC-learns bag-labeling functions derived from the p -concept class \mathcal{C} when for any tuple $(D_X, D_B, f, F) \in \text{MI-GEN}_c(D_X, f, \gamma, \pi)$ with $\gamma, \pi > 0$, and $\epsilon_B, \delta > 0$, algorithm \mathcal{A} requires $O(\text{poly}(\frac{1}{\epsilon_B}, \frac{1}{\pi}, \frac{1}{\delta}, \frac{1}{\delta}))$ bag-labeled instances sampled independently from the MI generative process in Figure 1(b) to produce an empirical bag-labeling function \widehat{H} with risk $R_F^{\text{AUC}}(\widehat{H}) < \epsilon_B$ with probability at least $1 - \delta$ over samples.*

Again, we will learn an instance p -concept h that models c , and then show that a sufficiently accurate p -concept can produce an empirical bag-labeling function \widehat{H} that models F with high AUC. To do this, we will show that the AUC error of \widehat{H} with respect to F , $R_F^{\text{AUC}}(\widehat{H})$, is bounded in terms of the AUC error of \widehat{H} with respect to \widehat{F} , $R_{\widehat{F}}^{\text{AUC}}(\widehat{H})$.

Lemma 4 *Suppose bag samples are of size at least $m_i : m_i \leq |X_i|$, then $R_F^{\text{AUC}}(\widehat{H}) \leq \frac{1}{P_{\text{neg}}} R_{\widehat{F}}^{\text{AUC}}(\widehat{H}) + (1 - \pi)^{m_i}$.* ■

Proof See Appendix A.

Finally, given the bound in Lemma 4, we can derive a result on learning high-AUC bag concepts with respect to the underlying bag-labeling function F . As with the results in Theorem 3, we state the results conditioned on the fact that bag sizes m_i respect some constraints to account for the error that naturally results from insufficiently large samples of instances in positive bags.

Theorem 6 *Bag-labeling functions derived from p -concept class \mathcal{C} with pseudo-dimension $\text{PD}(\mathcal{C})$ are AUC-PAC-learnable from MI data using*

$$O\left(\frac{1}{(\epsilon_B^2 \gamma \pi P_{\text{neg}})^4} \left(\text{PD}(\mathcal{C}) \log \frac{1}{(\epsilon_B \gamma \pi P_{\text{neg}})} + \log \frac{1}{\delta} \right) \right). \quad (17)$$

examples using standard ERM approaches when bag sample sizes are bounded by $m_i \leq m \leq m_u$ and $m_i \geq \frac{1}{\pi} \log \frac{2}{\epsilon_B}$, where $\widehat{P} = \min\{\widehat{P}_{\text{neg}}, 1 - \widehat{P}_{\text{neg}}\}$. ■

Proof See Appendix A.

4.3 Discussion

The results on AUC learnability in the MI setting are surprising, because they imply the testable hypothesis that *standard supervised approaches* can be used to learn about instance and bag labels in the MI setting. The work that introduced the MI setting evaluated the performance, in terms of *accuracy*, of supervised approaches on MI problems and found them to perform poorly (Dietterich et al., 1997). Later empirical work found that supervised algorithms actually performed quite well on MI problems, *in terms of AUC* (Ray and Craven, 2005). This apparent discrepancy can be explained with the results in this section. Supervised approaches can perform well in terms of AUC on MI problems, but not, it seems, with respect to accuracy.

While the results in Section 3 do not formally show that supervised approaches cannot learn high-accuracy concepts, we conjecture that this is the case due to the one-sided noise inherent in learning to discriminate classes. As illustrated in Figure 7, the fact that negative instances appear some $\gamma > 0$ fraction of the time in negative bags means that learning to approximate c can be used to rank instances. However, accurately labeling instances using an approximation of c requires choosing a *threshold* to discriminate between positive and negative instances. If the value of γ were known in advance, then such a threshold might be selected at $1 - \frac{\gamma}{2}$, for example. However, without knowledge of γ or other further assumptions about the generative process, proving that such a threshold might be selected accurately is a direction for future work.

5. Empirical Evaluation

Because the results in this work imply the surprising fact that standard supervised algorithms can be used to learn concepts with high-AUC, but not high accuracy, from MI data, we explicitly evaluate this hypothesis using real-world MI data sets. As always, there are some differences between theory and practice that might confound the experimental results. Below we first explain these two key differences and argue why they do not threaten the validity of our experimental results. Then, we discuss the remainder of our experimental methodology and results.

5.1 Single Instance Learning

In these experiments, we use single-instance learning (SIL) to apply supervised algorithms to MI data. The SIL procedure takes an MI data set and applies to every instance the

label of its bag. Hence, like in the generative model described in Section 2 used to show the theoretical results in this work, the SIL training set consists of bag-labeled instances. However, unlike in the generative model used in this work, SIL samples more than one instance per bag. As a result, SIL potentially introduces some “correlation” between instances in the training set. Figure 1 provides a comparison of the generative model of Section 2 (Figure 1(b)) and that of SIL (Figure 1(c)).

We could make SIL more closely resemble our generative model by randomly discarding all but one instance in every bag. However, this would dramatically reduce the size of most practical MI data sets and would needlessly “throw away” the information associated with the discarded instances. Instead, we use all instances in the data set, and ignore the fact that they are potentially correlated, thereby assuming that every instance is sampled from an independent bag. The correlation could change the training distribution over instances, but this should *hurt* the performance of the supervised algorithm if it has any significant effect at all. Therefore, comparing SIL to MI-specific algorithms provides a comparison that is fair to the MI approaches.

5.2 Risk Minimization Approaches

The results on AUC learnability for MI data use results on learning via empirical risk minimization (ERM). ERM requires that some concept class C is fixed in advance, and a hypothesis $h \in C$ that minimizes empirical risk (in terms of accuracy) is selected. In practice, however, C might not be known *a priori*. Thus, structural risk minimization (SRM) strategies are often used in practice, which simultaneously select a hypothesis h that minimizes empirical risk while controlling the capacity of the class from which h is selected. The standard support vector machine (SVM) is an SRM approach, where the parameter C , selected via cross-validation, controls the trade-off between risk minimization and regularization. Although our theoretical result holds for ERM, we will use the SRM-based SVM for these experiments. The same SRM strategy is used across all of the baseline algorithms.

Similarly, the SVM outputs confidence values that range from $(-\infty, \infty)$ rather than from $[0, 1]$. Thus, the SVM technically does not learn a p -concept. However, prior work has shown how it is possible to fit a logistic regression model to an SVM’s outputs to derive associated probabilities (Platt, 1999). However, since rescaling the data does not affect the relative rankings of the real-valued outputs produced by the SVM, the AUC of the classifier does not change. Accordingly, we report results using the raw confidence values produced by the SVM in these experiments.

5.3 Methodology

To evaluate our hypothesis that a supervised SVM can perform well with respect to AUC for learning instance- and bag-labeling functions, we use a total of 55 real-world data sets across a variety of problem domains, including 3D-QSAR (Dietterich et al., 1997), CBR (Andrews et al., 2003; Maron and Ratan, 1998; Rahmani et al., 2005), text categorization (Andrews et al., 2003; Settles et al., 2008), and audio classification (Briggs et al., 2012). Of the 55 data sets, 45 of them have instance labels, which are only used to test the instance-level performance of classifiers, not for training.

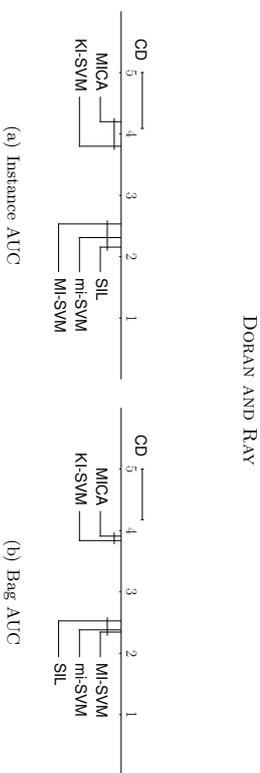


Figure 8: The average ranks (lower is better) of approaches on the instance- and bag-labeling tasks, evaluated using AUC. Statistically insignificant differences in performance are indicated with horizontal lines.

The SIL approach combined with a standard supervised SVM is compared with four popular baseline MI SVM approaches: mi-SVM (Andrews et al., 2003), MICA (Mangasarian and Wild, 2008), and the “instance” variant of KI-SVM (Li et al., 2009), which have been specifically designed to learn bag or instance labels from MI data. Prior empirical results show that these approaches constitute the state-of-the-art in instance-based MI SVM approaches (Doran and Ray, 2013).

The experiments used for this work were implemented in Python using NumPy (Ascher et al., 2001) and SciPy (Jones et al., 2001) for general matrix computations and the CVXOPT library (Dahl and Vandenberghe, 2009) for solving quadratic programs (QPs). We use the authors’ original MATLAB code, found at http://lamba.nju.edu.cn/code-KTSVM_astx, for the key instance SVM (KI-SVM) approach (Lin et al., 2012). We evaluate algorithms using 10-fold stratified cross-validation, with 5-fold inner-validation used to select parameters using random search (Bergstra and Bengio, 2012). Parameter selection is performed with respect to bag-level labels (since instance-level labels are unavailable at training time, even during cross-validation). We use the radial basis function (RBF) kernel with all algorithms, with scale parameter $\gamma \in [10^{-6}, 10^1]$, and regularization-loss trade-off parameter $C \in [10^{-2}, 10^5]$. The L_2 norm is used for regularization in all algorithms.

To statistically compare the classifiers, we use the approach described by Demšar (2006). We use the nonparametric Friedman test to reject the null hypothesis that the algorithms perform equally at an $\alpha = 0.001$ significance level. Finally, we plot the average ranks using a *critical difference* diagram, which uses the Nemenyi test to identify statistically equivalent groups of classifiers at an $\alpha = 0.05$ significance level.

5.4 Results and Discussion

The results are summarized using critical difference diagrams in Figure 8. Using AUC to measure performance, the ranks of the approaches are averaged across the 45 instance-labeled data sets for the instance-level metrics and across the 55 data sets for the bag-level metrics. Lower ranks indicate better performance. Full results can be found in Table 4 and Table 5.

Prior work has found that with respect to accuracy, the naïve SIL approach applied to MI data does not perform well (Dietterich et al., 1997; Doran and Ray, 2014). On the other hand, with respect to AUC, the relative performance of SIL increases significantly, and SIL

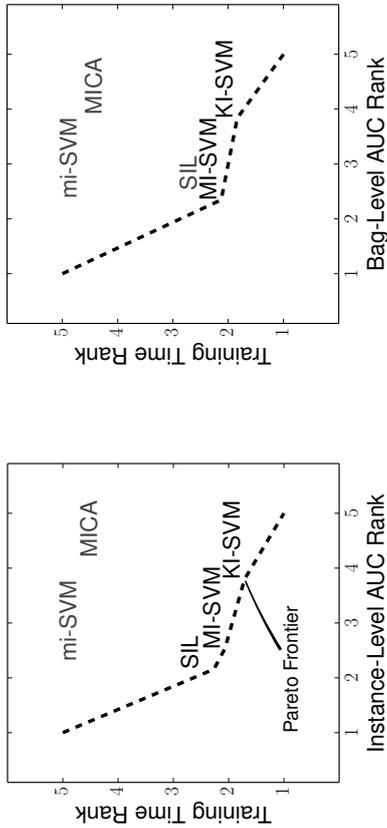


Figure 9: Comparison of supervised and MI-specific approaches in terms of running time and classification performance (AUC). Lower ranks correspond to better performance and faster training time. The Pareto frontier shows algorithms that are not dominated by any other algorithm along both dimensions.

performs as well the best MI approaches. For instance-level AUC, SIL is the highest-ranked approach. For bag-level AUC, SIL is not the best approach on average, but it is statistically equivalent to the top MI approaches. Since more samples are required to learn a bag-level concept using SIL, it could be that performance would improve even more with a larger training sample.

These surprising results support the theoretical framework described in Section 2. In particular, the experimental results suggest that the assumptions made by our generative model hold in practice in many cases. For example, we claim that the assumption that negative instances appear in negative bags ($\gamma > 0$) is weak and reasonable for many MI domains. In CBIR, negative background segments are likely to appear at least some of the time in images without the object of interest. The experimental results provide empirical support for this claim across the four domains on which we evaluate the classifiers. Of course, there might be domains for which this assumption does not hold. Determining whether any learnability results can be derived under weaker assumptions is an interesting question for future work.

There are also several ways that the theoretical and empirical results in this work can inform future work on MI learning. As mentioned earlier, the first work on MIL used accuracy as a performance measure, and found the SIL approach to be inaccurate in the MI setting for labeling bags (Dietterich et al., 1997). As a result, subsequent studies rarely used it as a baseline when evaluating new MI techniques. However, our results suggest that SIL should be used as a baseline when evaluating new MI approaches, especially if the intended application involves ranking bags or instances.

For researchers looking to learn high-AUC instance concepts from MI data, our results suggest that supervised approaches often suffice for this purpose in practice. Since supervised approaches are typically more computationally efficient than their MI counterparts, our theoretical and empirical justification for using supervised approaches with MI data provides a valuable practical benefit. The results in Figure 9 support this claim. The training time required by the algorithms for each data set is ranked with 1 corresponding to the fastest algorithm, and these ranks are averaged across data sets. Then, the combined performance of each approach in terms of both AUC and training time is shown in Figure 9. The Pareto frontier, the set of algorithms for which there does not exist any other algorithm that has better performance along both dimensions, is indicated in the figure. SIL is at or near the Pareto frontier for both instance- and bag-labeling.

For learning high-AUC bag-labeling concepts, MI algorithms still had a slight (but statistically insignificant) advantage over SIL in terms of classifier performance and training time. However, we have observed that even better performance can be attained by applying standard supervised approaches directly to the bag-level learning task using kernel methods (Doran and Ray, 2013).

Table 4: Instance-level AUC results for Section 5.4. The best result is indicated in boldface.

Dataset	SIL	MI-SVM	mi-SVM	KI-SVM	MICA
SIVAL01	0.758	0.872	0.836	0.758	0.898
SIVAL02	0.867	0.841	0.782	0.761	0.815
SIVAL03	0.676	0.588	0.795	0.690	0.934
SIVAL04	0.647	0.651	0.859	0.595	0.836
SIVAL05	0.954	0.810	0.961	0.906	0.754
SIVAL06	0.619	0.489	0.603	0.516	0.703
SIVAL07	0.895	0.784	0.903	0.780	0.725
SIVAL08	0.868	0.852	0.768	0.556	0.759
SIVAL09	0.829	0.730	0.824	0.771	0.581
SIVAL10	0.882	0.788	0.948	0.686	0.721
SIVAL11	0.965	0.795	0.952	0.746	0.600
SIVAL12	0.566	0.541	0.515	0.690	0.623
Newsgroups01	0.980	0.953	0.968	0.834	0.584
Newsgroups02	0.904	0.899	0.864	0.850	0.572
Newsgroups03	0.866	0.783	0.782	0.686	0.576
Newsgroups04	0.923	0.883	0.885	0.846	0.612
Newsgroups05	0.951	0.922	0.906	0.796	0.537
Newsgroups06	0.946	0.948	0.895	0.824	0.587
Newsgroups07	0.907	0.853	0.835	0.827	0.604
Newsgroups08	0.753	0.881	0.909	0.808	0.551
Newsgroups09	0.711	0.962	0.979	0.869	0.560
Newsgroups10	0.660	0.947	0.908	0.746	0.565
Newsgroups11	0.728	0.971	0.980	0.968	0.702
Newsgroups12	0.958	0.961	0.942	0.767	0.536

continued...

Table 4: Instance-level AUC results (continued).

Dataset	SIL	MI-SVM	mi-SVM	KI-SVM	MICA
Newsgrroups13	0.970	0.939	0.911	0.920	0.608
Newsgrroups14	0.823	0.903	0.884	0.902	0.614
Newsgrroups15	0.736	0.949	0.955	0.930	0.553
Newsgrroups16	0.454	0.938	0.906	0.940	0.600
Newsgrroups17	0.946	0.913	0.921	0.854	0.565
Newsgrroups18	0.964	0.914	0.922	0.797	0.605
Newsgrroups19	0.931	0.914	0.826	0.803	0.558
Newsgrroups20	0.573	0.884	0.914	0.912	0.556
Birdsong01	0.762	0.925	0.907	0.704	0.708
Birdsong02	0.895	0.884	0.849	0.574	0.748
Birdsong03	0.782	0.729	0.673	0.636	0.599
Birdsong04	0.966	0.927	0.932	0.905	0.858
Birdsong05	0.686	0.439	0.641	0.422	0.498
Birdsong06	0.627	0.741	0.581	0.540	0.719
Birdsong07	0.782	0.570	0.857	0.441	0.814
Birdsong08	0.836	0.615	0.796	0.552	0.774
Birdsong09	0.920	0.940	0.915	0.889	0.702
Birdsong10	0.858	0.859	0.879	0.763	0.757
Birdsong11	0.989	0.971	0.970	0.982	0.712
Birdsong12	0.954	0.907	0.918	0.490	0.745
Birdsong13	0.799	0.640	0.806	0.605	0.589

Table 5: Bag-level AUC results for Section 5.4. The best result is indicated in boldface.

Dataset	SIL	MI-SVM	mi-SVM	KI-SVM	MICA
mnski1	0.922	0.845	0.943	0.836	0.849
mnsk2	0.897	0.949	0.661	0.665	0.913
elephant	0.919	0.912	0.916	0.676	0.871
fox	0.662	0.589	0.632	0.500	0.615
tiger	0.859	0.856	0.853	0.673	0.688
field	0.923	0.871	0.908	0.687	0.847
flower	0.907	0.873	0.921	0.810	0.759
mountain	0.916	0.915	0.935	0.759	0.830
SIVAL01	0.626	0.954	0.875	0.643	0.933
SIVAL02	0.826	0.952	0.731	0.747	0.863
SIVAL03	0.785	0.666	0.716	0.708	0.906
SIVAL04	0.657	0.683	0.831	0.697	0.957
SIVAL05	0.985	0.964	1.000	0.938	0.914
SIVAL06	0.648	0.756	0.753	0.542	0.918
SIVAL07	0.793	0.993	0.972	0.969	0.974

continued...

Table 5: Bag-level AUC results (continued).

Dataset	SIL	MI-SVM	mi-SVM	KI-SVM	MICA
SIVAL08	0.874	0.998	0.812	0.488	0.865
SIVAL09	0.907	0.979	0.822	0.768	0.709
SIVAL10	0.819	0.772	0.930	0.643	0.785
SIVAL11	0.981	1.000	0.987	0.817	0.736
SIVAL12	0.601	0.621	0.516	0.692	0.710
Newsgrroups01	0.928	0.931	0.870	0.746	0.535
Newsgrroups02	0.873	0.794	0.878	0.826	0.538
Newsgrroups03	0.755	0.715	0.805	0.640	0.517
Newsgrroups04	0.765	0.767	0.727	0.631	0.511
Newsgrroups05	0.776	0.842	0.760	0.800	0.538
Newsgrroups06	0.814	0.862	0.837	0.741	0.521
Newsgrroups07	0.802	0.798	0.789	0.844	0.576
Newsgrroups08	0.674	0.810	0.837	0.759	0.532
Newsgrroups09	0.728	0.925	0.918	0.784	0.552
Newsgrroups10	0.752	0.904	0.900	0.696	0.550
Newsgrroups11	0.709	0.975	0.957	0.816	0.669
Newsgrroups12	0.844	0.805	0.858	0.695	0.530
Newsgrroups13	0.971	0.911	0.914	0.930	0.602
Newsgrroups14	0.648	0.825	0.861	0.869	0.600
Newsgrroups15	0.742	0.886	0.913	0.928	0.555
Newsgrroups16	0.564	0.860	0.538	0.838	0.543
Newsgrroups17	0.630	0.787	0.751	0.674	0.533
Newsgrroups18	0.864	0.874	0.797	0.771	0.558
Newsgrroups19	0.754	0.812	0.745	0.640	0.548
Newsgrroups20	0.575	0.807	0.757	0.770	0.520
OHSUMED1	0.958	0.914	0.954	0.779	0.816
OHSUMED2	0.740	0.638	0.775	0.520	0.715
Birdsong01	0.894	0.974	0.976	0.807	0.824
Birdsong02	0.902	0.895	0.895	0.570	0.827
Birdsong03	0.908	0.916	0.909	0.747	0.755
Birdsong04	0.999	0.998	0.989	0.992	0.945
Birdsong05	0.664	0.623	0.542	0.529	0.699
Birdsong06	0.926	0.915	0.964	0.728	0.975
Birdsong07	0.931	0.928	0.934	0.617	0.919
Birdsong08	0.913	0.901	0.921	0.723	0.908
Birdsong09	0.984	0.941	0.962	0.897	0.787
Birdsong10	0.947	0.975	0.956	0.819	0.862
Birdsong11	0.991	0.988	0.991	0.995	0.803
Birdsong12	0.996	0.961	0.993	0.609	0.737
Birdsong13	0.980	0.948	0.985	0.885	0.780

6. Conclusion

In this work, we describe a new generative model for the MI setting in which bags are viewed as distributions over instances. The sets of instances observed in a training sample are then viewed as samples from each underlying bag distribution. We then introduce several additional assumptions that we show entail instance and bag concept learnability. We discuss the relationship between the proposed model and those found in prior work.

Next, we describe new positive learnability results for learning instance or bag concepts from data generated by MI-GEN. We describe how our generative model allows for learnability while excluding scenarios used to show hardness under other generative models. Nevertheless, our generative process extends prior results on instance concept learnability that are over a decade old (Blum and Kalai, 1998). We also show that MI-GEN can incorporate the non-IID instance assumption within bag-specific distributions over instances, so assuming that samples from individual bags are drawn independently is not a restrictive assumption in our model.

Finally, we argue that for many real-world applications of MIL, it is sufficient to *rank* instances or bags rather than assign accurate binary labels. Accordingly, we derived results demonstrating the ability to learn high-AUC rankings of instances or bags from data generated by a process in MI-GEN. The surprising aspect of these results is that such rankings can be found via *standard supervised approaches*. We evaluate this surprising hypothesis empirically and find that supervised approaches *can* in fact learn to rank from MI data in practice. Thus, the empirical results support the assumptions made by MI-GEN.

Our work provides a starting point for many future investigations of learning in the MI framework. For example, we plan to extend our learnability results to the multi-class and multi-label settings. Furthermore, we plan to investigate generalizations of our generative model that allow for other previously studied instance- and bag-label relationships (Scott et al., 2005). Finally, some recent work has investigated learnability of real-valued bag-level concepts under a similar generative process for MI regression (Szabó et al., 2015). We plan to investigate instance-level learning in the MI regression setting.

Acknowledgements

We thank the anonymous reviewers for their feedback. G. Doran was supported by GAANN grant P2000A090265 from the US Department of Education and NSF grant CNS-1035602. S. Ray was partially supported by CWRU award OSA110264. This work made use of the High Performance Computing Resource in the Core Facility for Advanced Research Computing at Case Western Reserve University.

Appendix A. Detailed Proofs

Lemma 1 *Let $R_f(g)$ be the risk of an instance labeling concept g , and $R_{\widehat{F}}(\widehat{G})$ be the risk of the empirical bag-labeling function $\widehat{G}(X_i) = \max_j g(x_{ij})$. Then if bag sample sizes are bounded by m_u ($\forall i : |X_i| \leq m_u$), $R_{\widehat{F}}(\widehat{G}) \leq m_u R_f(g)$.*

Proof First, observe that when all elements of an empirical bag X_i are labeled correctly by g , $\widehat{F}(X_i) = \widehat{G}(X_i)$, so when $\widehat{F}(X_i) \neq \widehat{G}(X_i)$, at least one instance in X_i is labeled

incorrectly by g . In set notation, this implication is equivalent to the statement

$$\left\{ X_i : \widehat{F}(X_i) \neq \widehat{G}(X_i) \right\} \subseteq \left\{ X_i : (f(x_{i1}) \neq g(x_{i1})) \vee \dots \vee (f(x_{im}) \neq g(x_{im})) \right\}.$$

Using indicator function ($\mathbb{1}[\cdot]$) notation, the statement above implies

$$\begin{aligned} \mathbb{1}[\widehat{F}(X_i) \neq \widehat{G}(X_i)] &\leq \mathbb{1}[(f(x_{i1}) \neq g(x_{i1})) \vee \dots \vee (f(x_{im}) \neq g(x_{im}))] \\ &= \mathbb{1}\left[\bigvee_{x_{ij} \in X_i} (f(x_{ij}) \neq g(x_{ij}))\right] \\ &\leq \sum_{x_{ij} \in X_i} \mathbb{1}[f(x_{ij}) \neq g(x_{ij})]. \end{aligned}$$

Using this inequality in the definition of risk for empirical bag-labeling functions (Equation 10) yields

$$\begin{aligned} R_{\widehat{F}}(\widehat{G}) &= \int_{\mathcal{B}} \int_{\mathcal{X}^*} \mathbb{1}[\widehat{F}(X_i) \neq \widehat{G}(X_i)] d\mathbb{P}(X_i | B) d\mathbb{P}(B) \\ &\leq \int_{\mathcal{B}} \int_{\mathcal{X}^*} \sum_{x_{ij} \in X_i} \mathbb{1}[f(x_{ij}) \neq g(x_{ij})] d\mathbb{P}(X_i | B) d\mathbb{P}(B). \end{aligned}$$

By the independence of the instances $x_{ij} \in X_i$, and the bound m_u on bag sample sizes, we can rewrite the inner integral to conclude that

$$\begin{aligned} R_{\widehat{F}}(\widehat{G}) &\leq \int_{\mathcal{B}} m_u \int_{\mathcal{X}^*} \mathbb{1}[f(x) \neq g(x)] d\mathbb{P}(x | B) d\mathbb{P}(B) \\ &= m_u \int_{\mathcal{X}^*} \mathbb{1}[f(x) \neq g(x)] d\mathbb{P}(x) \\ &= m_u R_f(g). \end{aligned}$$

Exchanging the order of the integrals and marginalizing out the individual bag distributions to obtain an integral with respect to the instance distribution follows from Condition 1 in Definition 1. \blacksquare

Lemma 2 *For any empirical bag-labeling concept \widehat{G} ,*

$$R_{\widehat{F}}(\widehat{G}) \leq R_{\widehat{F}}(\widehat{G}) + R_{\widehat{F}}(\widehat{F}).$$

Proof First, note that if $\widehat{G}(X) = \widehat{F}(X)$ and $\widehat{F}(X) = F(B)$, then $\widehat{G}(X) = F(B)$. Thus, if $\widehat{G}(X) \neq F(B)$, then either $\widehat{G}(X) \neq \widehat{F}(X)$ or $\widehat{F}(X) \neq F(B)$. In set notation, this is equivalent to the statement

$$\left\{ (X, B) : \widehat{G}(X) \neq F(B) \right\} \subseteq \left\{ (X, B) : (\widehat{G}(X) \neq \widehat{F}(X)) \vee (\widehat{F}(X) \neq F(B)) \right\}.$$

Using indicator function notation, the statement above implies

$$\begin{aligned} \mathbb{1}[\widehat{G}(X) \neq F(B)] &\leq \mathbb{1}[(\widehat{G}(X) \neq \widehat{F}(X)) \vee (\widehat{F}(X) \neq F(B))] \\ &\leq \mathbb{1}[(\widehat{G}(X) \neq \widehat{F}(X))] + \mathbb{1}[(\widehat{F}(X) \neq F(B))]. \end{aligned}$$

Finally, substituting the expression above into the definitions of risk yields

$$\begin{aligned} R_F(\widehat{G}) &= \int_{\mathcal{B}} \int_{\mathcal{X}^*} \mathbb{1}[\widehat{G}(X) \neq F(B)] \, d\mathbb{P}(X | B) \, d\mathbb{P}(B) \\ &\leq \int_{\mathcal{B}} \int_{\mathcal{X}^*} \mathbb{1}[(\widehat{G}(X) \neq \widehat{F}(X))] \, d\mathbb{P}(X | B) \, d\mathbb{P}(B) \\ &\quad + \int_{\mathcal{B}} \int_{\mathcal{X}^*} \mathbb{1}[(\widehat{F}(X) \neq F(B))] \, d\mathbb{P}(X | B) \, d\mathbb{P}(B) \\ &= R_{\widehat{F}}(\widehat{G}) + R_F(\widehat{F}). \end{aligned}$$

■

Lemma 3 *Suppose bag samples are of size at least m_i ($\forall i : m_i \leq |X_i|$), then $R_F(\widehat{F}) \leq (1 - \pi)^{m_i}$.*

Proof Given the definition of $R_F(\widehat{F})$, we can decompose it as such

$$\begin{aligned} R_F(\widehat{F}) &= \int_{\mathcal{B}} \int_{\mathcal{X}^*} \mathbb{1}[F(B_i) \neq \widehat{F}(X_i)] \, d\mathbb{P}(X_i | B_i) \, d\mathbb{P}(B_i) \\ &= \int_{\mathcal{B}_+} \int_{\mathcal{X}^*} \mathbb{1}[F(B_i) \neq \widehat{F}(X_i)] \, d\mathbb{P}(X_i | B_i) \, d\mathbb{P}(B_i) \\ &\quad + \int_{\mathcal{B}_-} \int_{\mathcal{X}^*} \mathbb{1}[F(B_i) \neq \widehat{F}(X_i)] \, d\mathbb{P}(X_i | B_i) \, d\mathbb{P}(B_i). \end{aligned}$$

On the set of negative bags \mathcal{B}_- , F and \widehat{F} always agree, since only negative instances are sampled within negative bags. Therefore, the second term of the decomposition can be eliminated and we are left with

$$R_F(\widehat{F}) = \int_{\mathcal{B}_+} \int_{\mathcal{X}^*} \mathbb{1}[F(B_i) \neq \widehat{F}(X_i)] \, d\mathbb{P}(X_i | B_i) \, d\mathbb{P}(B_i).$$

Now, we observe that for a positive bag B_i , the only way that F and \widehat{F} can disagree is if every instance in X_i is negative. Using basic properties of indicator functions (namely, that $\mathbb{1}[\bigvee_i E_i] = \prod_i \mathbb{1}[E_i^c]$), we can use this fact to rewrite the expression above as

$$\begin{aligned} R_F(\widehat{F}) &= \int_{\mathcal{B}_+} \int_{\mathcal{X}^*} \mathbb{1}[F(B_i) \neq \widehat{F}(X_i)] \, d\mathbb{P}(X_i | B_i) \, d\mathbb{P}(B_i) \\ &= \int_{\mathcal{B}_+} \int_{\mathcal{X}^*} \mathbb{1}[\bigwedge_{x_{ij} \in X_i} (f(x_{ij}) = 0)] \, d\mathbb{P}(X_i | B_i) \, d\mathbb{P}(B_i) \\ &= \int_{\mathcal{B}_+} \int_{\mathcal{X}^*} \prod_{x_{ij} \in X_i} \mathbb{1}[f(x_{ij}) = 0] \, d\mathbb{P}(X_i | B_i) \, d\mathbb{P}(B_i). \end{aligned}$$

Since the instances $x_{ij} \in X_i$ are independent, we can rewrite the integral as

$$\begin{aligned} R_F(\widehat{F}) &= \int_{\mathcal{B}_+} \int_{\mathcal{X}^*} \prod_{x_{ij} \in X_i} \mathbb{1}[f(x_{ij}) = 0] \, d\mathbb{P}(X_i | B_i) \, d\mathbb{P}(B_i) \\ &= \int_{\mathcal{B}_+} \prod_{x_{ij} \in X_i} \left(\int_{\mathcal{X}} \mathbb{1}[f(x_{ij}) = 0] \, d\mathbb{P}(x_{ij} | B_i) \right) \, d\mathbb{P}(B_i) \\ &\leq \int_{\mathcal{B}_+} \prod_{x_{ij} \in X_i} (1 - \pi) \, d\mathbb{P}(B_i) \\ &\leq \int_{\mathcal{B}_+} (1 - \pi)^{m_i} \, d\mathbb{P}(B_i) \\ &= (1 - \pi)^{m_i} \int_{\mathcal{B}_+} \, d\mathbb{P}(B_i) \\ &\leq (1 - \pi)^{m_i}. \end{aligned}$$

■

Theorem 4 *An instance p -concept class \mathcal{C} with pseudo-dimension $\text{PD}(\mathcal{C})$ is Instance MI AUC-P4C-learnable using $O\left(\frac{1}{\epsilon^2 p^2}\left(\text{PD}(\mathcal{C}) \log \frac{1}{\epsilon} + \log \frac{1}{\delta}\right)\right)$ examples with standard ERM approaches, where $p = \min\{p_{\text{neg}}, 1 - p_{\text{neg}}\}$.*

Proof For any $c \in \mathcal{C}$, we can use ERM with respect to the quadratic loss function to learn a hypothesis h such that $\mathbb{E}[|h(x) - c(x)|^2] < \epsilon$ with probability $1 - \delta$ across samples. By Jensen's inequality, this bounds the expected absolute deviation between h and c :

$$\mathbb{E}[|h(x) - c(x)|] \leq \sqrt{\mathbb{E}[|h(x) - c(x)|^2]} < \sqrt{\epsilon}.$$

Then, by Markov's inequality, this expression bounds the probability over examples that $|h(x) - c(x)|$ exceeds some constant t :

$$\mathbb{P}[|h(x) - c(x)| > t] \leq \frac{\mathbb{E}[|h(x) - c(x)|]}{t} < \frac{\sqrt{\epsilon}}{t}. \quad (18)$$

Therefore, with high probability, $|h(x) - c(x)|$ is small for small ϵ .

Now, we can proceed by following the intuition illustrated in Figure 7. In particular, we will show that the AUC risk is bounded when h and c agree on examples with high probability. First, suppose $|h(x) - c(x)| \leq \frac{\gamma}{2}$ for both of a pair (x_+, x_-) of positive and negative instances. Then for the negative instance, x_- , by Definition 1, Condition 3,

$$h(x_-) \leq c(x_-) + \frac{\gamma}{2} \leq (1 - \gamma) + \frac{\gamma}{2} = 1 - \frac{\gamma}{2}.$$

Similarly, for the positive instance, x_+ , by Definition 1, Condition 2,

$$h(x_+) \geq c(x_+) - \frac{\gamma}{2} = 1 - \frac{\gamma}{2}.$$

Hence, we have that $h(x_-) \leq h(x_+)$.

By contraposition of the conclusion above, if $h(x_-) > h(x_+)$, then it is either the case that $|h(x_-) - c(x_-)| > \frac{\gamma}{2}$ or that $|h(x_+) - c(x_+)| > \frac{\gamma}{2}$. In set theoretic terms, this means

$$\{(x_+, x_-) : h(x_-) > h(x_+)\} \subseteq \{(x_+, x_-) : |h(x_-) - c(x_-)| > \frac{\gamma}{2} \vee |h(x_+) - c(x_+)| > \frac{\gamma}{2}\}$$

In indicator function notation, this implies

$$\begin{aligned} \mathbb{1}[h(x_-) > h(x_+)] &\leq \mathbb{1}[|h(x_-) - c(x_-)| > \frac{\gamma}{2} \vee |h(x_+) - c(x_+)| > \frac{\gamma}{2}] \\ &\leq \mathbb{1}[|h(x_-) - c(x_-)| > \frac{\gamma}{2}] + \mathbb{1}[|h(x_+) - c(x_+)| > \frac{\gamma}{2}]. \end{aligned}$$

Substituting this expression into the definition of $R_f^{\text{AUC}}(h)$ (Equation 14) yields

$$\begin{aligned} R_f^{\text{AUC}}(h) &= \frac{\int_{X_-} \int_{X_+} \mathbb{1}[h(x_-) > h(x_+)] d\mathbb{P}(x_+) d\mathbb{P}(x_-)}{(1 - p_{\text{neg}})p_{\text{neg}}} \\ &\leq \frac{\int_{X_-} \int_{X_+} \mathbb{1}[|h(x_-) - c(x_-)| > \frac{\gamma}{2}] d\mathbb{P}(x_+) d\mathbb{P}(x_-)}{(1 - p_{\text{neg}})p_{\text{neg}}} \\ &\quad + \frac{\int_{X_+} \int_{X_+} \mathbb{1}[|h(x_+) - c(x_+)| > \frac{\gamma}{2}] d\mathbb{P}(x_+) d\mathbb{P}(x_-)}{(1 - p_{\text{neg}})p_{\text{neg}}} \\ &= \frac{\int_{X_-} \mathbb{1}[|h(x_-) - c(x_-)| > \frac{\gamma}{2}] d\mathbb{P}(x_-)}{1 - p_{\text{neg}}} \\ &\quad + \frac{\int_{X_+} \mathbb{1}[|h(x_+) - c(x_+)| > \frac{\gamma}{2}] d\mathbb{P}(x_+)}{1 - p_{\text{neg}}}. \end{aligned}$$

Then, using the definition $p = \min\{p_{\text{neg}}, 1 - p_{\text{neg}}\}$, this becomes

$$\begin{aligned} R_f^{\text{AUC}}(h) &\leq \frac{\int_{X_-} \mathbb{1}[|h(x_-) - c(x_-)| > \frac{\gamma}{2}] d\mathbb{P}(x_-)}{p} \\ &\quad + \frac{\int_{X_+} \mathbb{1}[|h(x_+) - c(x_+)| > \frac{\gamma}{2}] d\mathbb{P}(x_+)}{p} \\ &= \frac{\int_{X_-} \mathbb{1}[|h(x_-) - c(x_-)| > \frac{\gamma}{2}] d\mathbb{P}(x_-) + \int_{X_+} \mathbb{1}[|h(x_+) - c(x_+)| > \frac{\gamma}{2}] d\mathbb{P}(x_+)}{p}. \end{aligned}$$

Finally, using the inequality derived in Equation 18, we have

$$R_f^{\text{AUC}}(h) \leq \frac{\mathbb{P}[|h(x) - c(x)| > \frac{\gamma}{2}]}{p} < \frac{2\sqrt{\epsilon}}{\gamma p}.$$

Therefore, it is sufficient to choose $\epsilon = \frac{(\gamma p)^2}{4}$ when learning h via ERM as so that $R_f^{\text{AUC}}(h) < \epsilon$.

Finally, the sample complexity bound results from substituting $\epsilon = \frac{(\gamma p)^2}{4}$ into the existing bound $O(\frac{1}{\epsilon} (\text{PD}(\mathcal{C}) \log \frac{1}{\epsilon} + \log \frac{1}{\delta}))$ for learning p -concepts using ERM (Kearns and Schapire, 1994). ■

Theorem 5 Empirical bag-labeling functions derived from p -concept class \mathcal{C} with pseudo-dimension $\text{PD}(\mathcal{C})$ are AUC-PAC-learnable from MI data using

$$O\left(\frac{m_u}{(\epsilon \gamma \hat{P})^4} \left(\text{PD}(\mathcal{C}) \log \frac{m_u}{(\epsilon \gamma \hat{P})} + \log \frac{1}{\delta}\right)\right)$$

examples with standard ERM approaches, where

$$\hat{P} \triangleq \min\{\hat{P}_{\text{neg}}, 1 - \hat{P}_{\text{neg}}\} \geq \min\{P_{\text{neg}}, 1 - p_{\text{neg}}\},$$

and m_u is an upper bound on bag sample size.

Proof As in Theorem 4, we will learn a p -concept h to model c accurately with high probability. Then, given bag samples X_+ with at least one positive instance and X_- with all negative instances, suppose that $|h(x) - c(x)| \leq \frac{\gamma}{2}$ for all instances across both samples. Then by the same argument as in Theorem 4 as illustrated in Figure 7, at least one instance in X_+ is assigned a label by h that is at least $1 - \frac{\gamma}{2}$, and all instances in X_- are assigned a label by h of at most $1 - \frac{\gamma}{2}$. Therefore, the maximum label assigned in X_+ , $\hat{H}(X_+)$, is greater than or equal to the maximum label in X_- , $\hat{H}(X_-)$.

By contraposition, if $\hat{H}(X_-) > \hat{H}(X_+)$, then the label $h(x)$ of some instance x in either X_+ or X_- deviates by more than $\frac{\gamma}{2}$ from $c(x)$. That is,

$$\begin{aligned} \{(X_+, X_-) : \hat{H}(X_-) > \hat{H}(X_+)\} \\ \subseteq \left\{ (X_+, X_-) : \left(\bigvee_{x \in X_+} |h(x) - c(x)| > \frac{\gamma}{2} \right) \vee \left(\bigvee_{x \in X_-} |h(x) - c(x)| > \frac{\gamma}{2} \right) \right\} \end{aligned}$$

Therefore, in indicator function notation,

$$\mathbb{1}[\hat{H}(X_-) > \hat{H}(X_+)] \leq \sum_{x \in X_+} \mathbb{1}[|h(x) - c(x)| > \frac{\gamma}{2}] + \sum_{x \in X_-} \mathbb{1}[|h(x) - c(x)| > \frac{\gamma}{2}].$$

Using the inequality above in the definition of $R_F^{\text{AUC}}(\hat{H})$ in Equation 15 gives

$$\begin{aligned} R_F^{\text{AUC}}(\hat{H}) &\leq \frac{(1 - \hat{P}_{\text{neg}})\hat{P}_{\text{neg}}}{\dots d\mathbb{P}(X_+ | B_+) d\mathbb{P}(X_- | B_-) d\mathbb{P}(B_+) d\mathbb{P}(B_-)} \\ &\quad \int_{B_-} \int_{B_-} \int_{X_+} \int_{X_+} \mathbb{1}[|h(x) - c(x)| > \frac{\gamma}{2}] \dots \\ &\quad \dots d\mathbb{P}(X_+ | B_+) d\mathbb{P}(X_- | B_-) d\mathbb{P}(B_+) d\mathbb{P}(B_-) \\ &\quad + \frac{(1 - \hat{P}_{\text{neg}})\hat{P}_{\text{neg}}}{\dots d\mathbb{P}(X_+ | B_+) d\mathbb{P}(X_- | B_-) d\mathbb{P}(B_+) d\mathbb{P}(B_-)} \\ &\quad \dots d\mathbb{P}(X_+ | B_+) d\mathbb{P}(X_- | B_-) d\mathbb{P}(B_+) d\mathbb{P}(B_-) \end{aligned}$$

Since the integrands above only depend on X_+ and X_- , we can rewrite the expression using the fact that

$$\begin{aligned} \int_{X_-} d\mathbb{P}(X_- | B_-) d\mathbb{P}(B_-) &= \mathbb{P}[\hat{F}(X) = 0] = \hat{P}_{\text{neg}} \\ \int_{X_+} d\mathbb{P}(X_+ | B_+) d\mathbb{P}(B_+) &= \mathbb{P}[\hat{F}(X) = 1] = 1 - \hat{P}_{\text{neg}}. \end{aligned}$$

The result is

$$\begin{aligned}
 R_{\hat{P}}^{\text{AVC}}(\hat{H}) &\leq \frac{\int_{\mathcal{B}} \int_{\mathcal{X}^*} \sum_{x \in \mathcal{X}_+} \mathbb{1}[|h(x) - c(x)| > \frac{\gamma}{2}] d\mathbb{P}(X_+ | \mathcal{B}_+) d\mathbb{P}(\mathcal{B}_+)}{(1 - \hat{P}_{\text{neg}})} \\
 &\quad + \frac{\int_{\mathcal{B}} \int_{\mathcal{X}^*} \sum_{x \in \mathcal{X}_-} \mathbb{1}[|h(x) - c(x)| > \frac{\gamma}{2}] d\mathbb{P}(X_- | \mathcal{B}_-) d\mathbb{P}(\mathcal{B}_-)}{\hat{P}_{\text{neg}}} \\
 &\leq \frac{\int_{\mathcal{B}} \int_{\mathcal{X}^*} \sum_{x \in \mathcal{X}_+} \mathbb{1}[|h(x) - c(x)| > \frac{\gamma}{2}] d\mathbb{P}(X_+ | \mathcal{B}_+) d\mathbb{P}(\mathcal{B}_+)}{\hat{P}} \\
 &\quad + \frac{\int_{\mathcal{B}} \int_{\mathcal{X}^*} \sum_{x \in \mathcal{X}_-} \mathbb{1}[|h(x) - c(x)| > \frac{\gamma}{2}] d\mathbb{P}(X_- | \mathcal{B}_-) d\mathbb{P}(\mathcal{B}_-)}{\hat{P}} \\
 &= \frac{\int_{\mathcal{B}} \int_{\mathcal{X}^*} \sum_{x \in \mathcal{X}} \mathbb{1}[|h(x) - c(x)| > \frac{\gamma}{2}] d\mathbb{P}(X | \mathcal{B}) d\mathbb{P}(\mathcal{B})}{\hat{P}}.
 \end{aligned}$$

By the independence of instances $x \in \mathcal{X}$, the upper bound m_u on bag size, and Condition 1 in Definition 1, we can rewrite the expression above as

$$\begin{aligned}
 R_{\hat{P}}^{\text{AVC}}(\hat{H}) &\leq \frac{m_u}{\hat{P}} \int_{\mathcal{B}} \int_{\mathcal{X}} \mathbb{1}[|h(x) - c(x)| > \frac{\gamma}{2}] d\mathbb{P}(x | \mathcal{B}) d\mathbb{P}(\mathcal{B}) \\
 &= \frac{m_u}{\hat{P}} \int_{\mathcal{X}} \mathbb{1}[|h(x) - c(x)| > \frac{\gamma}{2}] d\mathbb{P}(x) \\
 &= \frac{m_u}{\hat{P}} \mathbb{P}[|h(x) - c(x)| > \frac{\gamma}{2}].
 \end{aligned}$$

Then, by Markov's inequality in Equation 18,

$$R_{\hat{P}}^{\text{AVC}}(\hat{H}) \leq \frac{m_u}{\hat{P}} \mathbb{P}[|h(x) - c(x)| > \frac{\gamma}{2}] < \frac{2m_u\sqrt{\epsilon}}{\gamma\hat{P}}. \quad (19)$$

Therefore, choosing $\epsilon = \frac{(\gamma\hat{P})^2}{4m_u^2}$, is sufficient to learn \hat{H} with $R_{\hat{P}}^{\text{AVC}}(\hat{H}) < \epsilon\mathcal{B}$. Substituting this ϵ into the bound of Kearns and Schapire (1994) gives the sample complexity of learning \hat{H} as stated in the theorem.

Finally, we show that $\hat{P} \geq \min\{P_{\text{neg}}, 1 - P_{\text{neg}}\}$ as asserted in the theorem, which demonstrates that \hat{P} is independent of the bag size m (so there is no hidden dependence on bag size). First, observe that $\hat{P}_{\text{neg}} \geq P_{\text{neg}}$. The reason is that whenever a negative bag is sampled, a sample of only negative instances is guaranteed to be sampled from the bag. Thus, the probability of a negative sample of instances is at least the probability of sampling a negative bag.

Additionally, $1 - \hat{P}_{\text{neg}} \geq 1 - P_{\text{neg}}$. This is true because the probability of a sample containing a positive instance is at least the probability that the very first instance sampled is positive, which is $1 - P_{\text{neg}}$.

Combining the observations above, we get

$$\begin{aligned}
 \hat{P} &\triangleq \min\{\hat{P}_{\text{neg}}, 1 - \hat{P}_{\text{neg}}\} \\
 &\geq \min\{P_{\text{neg}}, 1 - P_{\text{neg}}\}.
 \end{aligned}$$

Lemma 4 *Suppose bag samples are of size at least m_l ($\forall i: m_l \leq |X_i|$), then $R_{\hat{P}}^{\text{AVC}}(\hat{H}) \leq \frac{1}{P_{\text{neg}}} R_{\hat{P}}^{\text{AVC}}(\hat{H}) + (1 - \pi)^{m_l}$.*

Proof We can derive the inequality in Lemma 4 by transforming the definition of $R_{\hat{P}}^{\text{AVC}}(\hat{H})$ in Equation 16 to that of $R_{\hat{P}}^{\text{AVC}}(\hat{H})$ in Equation 15. Starting from $R_{\hat{P}}^{\text{AVC}}(\hat{H})$, we get

$$\begin{aligned}
 R_{\hat{P}}^{\text{AVC}}(\hat{H}) &= \frac{\int_{\mathcal{B}_-} \int_{\mathcal{B}_+} \int_{\mathcal{X}^*} \int_{\mathcal{X}^*} \mathbb{1}[\hat{H}(X_-) > \hat{H}(X_+)] \dots \\
 &\quad \dots d\mathbb{P}(X_+ | \mathcal{B}_+) d\mathbb{P}(X_- | \mathcal{B}_-) d\mathbb{P}(\mathcal{B}_+) d\mathbb{P}(\mathcal{B}_-)}{(1 - P_{\text{neg}})P_{\text{neg}}} \dots \\
 &= \frac{\int_{\mathcal{B}_-} \int_{\mathcal{B}_+} \int_{\mathcal{X}^*} \int_{\mathcal{X}^*} \mathbb{1}[\hat{H}(X_-) > \hat{H}(X_+)] \dots \\
 &\quad \dots d\mathbb{P}(X_+ | \mathcal{B}_+) d\mathbb{P}(X_- | \mathcal{B}_-) d\mathbb{P}(\mathcal{B}_+) d\mathbb{P}(\mathcal{B}_-)}{(1 - P_{\text{neg}})P_{\text{neg}}} \\
 &\quad + \frac{\int_{\mathcal{B}_-} \int_{\mathcal{B}_+} \int_{\mathcal{X}^*} \int_{\mathcal{X}^*} \mathbb{1}[\hat{H}(X_-) > \hat{H}(X_+)] \dots \\
 &\quad \dots d\mathbb{P}(X_+ | \mathcal{B}_+) d\mathbb{P}(X_- | \mathcal{B}_-) d\mathbb{P}(\mathcal{B}_+) d\mathbb{P}(\mathcal{B}_-)}{(1 - P_{\text{neg}})P_{\text{neg}}}. \quad (B)
 \end{aligned} \quad (A)$$

Starting with (B), we see that since $\mathbb{1}[\hat{H}(X_-) > \hat{H}(X_+)] \leq 1$, we can rewrite this term as

$$\begin{aligned}
 (B) &\leq \frac{\int_{\mathcal{B}_-} \int_{\mathcal{B}_+} \int_{\mathcal{X}^*} \int_{\mathcal{X}^*} d\mathbb{P}(X_+ | \mathcal{B}_+) d\mathbb{P}(X_- | \mathcal{B}_-) d\mathbb{P}(\mathcal{B}_+) d\mathbb{P}(\mathcal{B}_-)}{(1 - P_{\text{neg}})P_{\text{neg}}} \\
 &= \frac{\int_{\mathcal{B}_+} \int_{\mathcal{X}^*} d\mathbb{P}(X_+ | \mathcal{B}_+) d\mathbb{P}(\mathcal{B}_+)}{(1 - P_{\text{neg}})} \\
 &\leq \frac{\int_{\mathcal{B}_+} (1 - \pi)^{m_l} d\mathbb{P}(\mathcal{B}_+)}{(1 - P_{\text{neg}})} = (1 - \pi)^{m_l}.
 \end{aligned}$$

The second step follows from the fact that $\int_{\mathcal{X}^*} d\mathbb{P}(X_+ | \mathcal{B}_+)$ is the probability of sampling only negative instances within a positive bag of size at least m_l , which is at most $(1 - \pi)^{m_l}$. Continuing with term (A), we can rewrite this as

$$\begin{aligned}
 (A) &= \frac{\int_{\mathcal{B}_-} \int_{\mathcal{B}_+} \int_{\mathcal{X}^*} \int_{\mathcal{X}^*} \mathbb{1}[\hat{H}(X_-) > \hat{H}(X_+)] \dots \\
 &\quad \dots d\mathbb{P}(X_+ | \mathcal{B}_+) d\mathbb{P}(X_- | \mathcal{B}_-) d\mathbb{P}(\mathcal{B}_+) d\mathbb{P}(\mathcal{B}_-)}{(1 - P_{\text{neg}})P_{\text{neg}}} \dots \\
 &\quad + \frac{\int_{\mathcal{B}_-} \int_{\mathcal{B}_+} \int_{\mathcal{X}^*} \int_{\mathcal{X}^*} \mathbb{1}[\hat{H}(X_-) > \hat{H}(X_+)] \dots \\
 &\quad \dots d\mathbb{P}(X_+ | \mathcal{B}_+) d\mathbb{P}(X_- | \mathcal{B}_-) d\mathbb{P}(\mathcal{B}_+) d\mathbb{P}(\mathcal{B}_-)}{(1 - P_{\text{neg}})P_{\text{neg}}}. \quad (D)
 \end{aligned} \quad (C)$$

Now, we see that (D) = 0, since it involves an integral over bags with positive instances in negative bags, which occurs with probability zero by Condition 2 in Definition 1. For (C),

- C. Bergeron, J. Zaretzki, C. Breeman, and K. P. Bennett. Multiple instance ranking. In *Proceedings of the International Conference on Machine Learning*, pages 48–55, 2008.
- J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.
- D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
- A. Blum and A. Kalai. A note on learning from multiple-instance examples. *Machine Learning Journal*, 30:23–29, 1998.
- A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the Association for Computing Machinery*, 36(4): 929–965, 1989.
- F. Biggs, X. Z. Fern, and R. Raich. Rank-loss support instance machines for MIML instance annotation. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 534–542, 2012.
- C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, 2002.
- Y. Chen, J. Bi, and J. Z. Wang. MILES: Multiple-instance learning via embedded instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12): 1931–1947, 2006.
- J. Dahl and L. Vandenbergh. CVXOPT: A Python package for convex optimization, 2009. <http://abel.ee.ucla.edu/cvxopt>.
- J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.
- J. Diestel and J. J. Uhl. *Vector Measures*. Mathematical surveys and monographs. American Mathematical Society, 1977.
- T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1–2):31–71, 1997.
- D. Dioninos, R. Sloan, and G. Turán. On multiple-instance learning of halfspaces. *Information Processing Letters*, 2012.
- G. Doran and S. Ray. A theoretical and empirical analysis of support vector machine methods for multiple-instance classification. *Machine Learning Journal*, pages 1–24, 2013.
- G. Doran and S. Ray. Learning instance concepts from multiple-instance data with bags as distributions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1802–1808, 2014.
- L. Du, W. L. Bunine, and M. Johnson. Topic segmentation with a structured topic model. In *Proceedings of the Human Language Technologies Conference of the North American Chapter of the Association for Computational Linguistics*, pages 190–200, 2013.
- L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 524–531, 2005.
- J. Foulds and P. Smyth. Multi-instance mixture models and semi-supervised learning. In *SIAM International Conference on Data Mining*. SIAM, 2011.
- J. R. Foulds. Learning instance weights in multi-instance learning. Master’s thesis, The University of Waikato, 2008.
- T. Gärtner, P. Flach, A. Kowalczyk, and A. Smola. Multi-instance kernels. In *Proceedings of the International Conference on Machine Learning*, pages 179–186, 2002.
- J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36, 1982.
- D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and computation*, 100(1):78–150, 1992.
- E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001. <http://www.scipy.org/>.
- M. Kandemir and F. A. Hamprecht. Instance label prediction by Dirichlet process multiple instance learning. In *Uncertainty in Artificial Intelligence*, 2014.
- M. J. Kearns and R. E. Schapire. Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and System Sciences*, 48(3):464–497, 1994.
- H. Knuck and N. de Freitas. Learning about individuals from group statistics. In *Uncertainty in Artificial Intelligence*, 2005.
- O. Kundakcioglu, O. Seref, and P. Pardalos. Multiple instance learning via margin maximization. *Applied Numerical Mathematics*, 60(4):358–369, 2010.
- Y.-F. Li, J. T. Kwok, J. W. Tsang, and Z.-H. Zhou. A convex method for locating regions of interest with multi-instance learning. In *Machine Learning and Knowledge Discovery in Databases*, pages 15–30. Springer, 2009.
- G. Liu, J. Wu, and Z.-H. Zhou. Key instance detection in multi-instance learning. In *Proceedings of the Asian Conference on Machine Learning*, pages 253–268, 2012.
- P. Long and L. Tan. PAC learning axis-aligned rectangles with respect to product distributions from multiple-instance examples. *Machine Learning Journal*, 30(1):7–21, 1998.
- O. Mangasarian and E. Wild. Multiple instance classification via successive linear programming. *Journal of Optimization Theory and Applications*, 137:555–568, 2008.

- O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. In *Advances in Neural Information Processing Systems*, pages 570–576, 1998.
- O. Maron and A. L. Ratan. Multiple-instance learning for natural scene classification. In *Proceedings of the International Conference on Machine Learning*, pages 341–349, 1998.
- J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, 1999.
- R. Rahmani, S. A. Goldman, H. Zhang, J. Krettek, and J. E. Fritts. Localized content based image retrieval. In *Proceedings of the 7th ACM SIGMM International Workshop on Multimedia Information Retrieval*, pages 227–236. ACM, 2005.
- S. Ray and M. Craven. Supervised versus multiple instance learning: an empirical comparison. In *Proceedings of the International Conference on Machine Learning*, pages 697–704, 2005.
- S. Sabato and N. Tishby. Multi-instance learning with any hypothesis class. *Journal of Machine Learning Research*, 13:2999–3039, 2012.
- G. Salton and M. McGill. *Introduction to modern information retrieval*. McGraw-Hill Computer Science Series. McGraw-Hill, 1983.
- S. Scott, J. Zhang, and J. Brown. On generalized multiple-instance learning. *International Journal of Computational Intelligence and Applications*, 5(1):21–35, 2005.
- B. Settles, M. Craven, and S. Ray. Multiple-instance active learning. In *Advances in Neural Information Processing Systems*, pages 1289–1296, 2008.
- H. U. Simon. PAC-learning in the presence of one-sided classification noise. *Annals of Mathematics and Artificial Intelligence*, pages 1–18, 2012.
- Z. Szabó, A. Gretton, B. Póczos, and B. Sriperumbudur. Two-stage sampled learning theory on distributions. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2015.
- V. Tragaute do Ó, D. Fierens, and H. Blockeel. Instance-level accuracy versus bag-level accuracy in multi-instance learning. In *Proceedings of the 23rd Benelux Conference on Artificial Intelligence*, 2011.
- V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971.
- X. Xu. Statistical learning in multiple instance problems. Master’s thesis, The University of Waikato, 2003.
- S.-H. Yang, H. Zha, and B.-G. Hu. Dirichlet-Bernoulli alignment: A generative model for multi-class multi-label multi-instance corpora. In *Advances in Neural Information Processing Systems*, pages 2143–2150, 2009.
- Z. Zhou, Y. Sun, and Y. Li. Multi-instance learning by treating instances as non-IID samples. In *Proceedings of the International Conference on Machine Learning*, pages 1249–1256, 2009.

An Online Convex Optimization Approach to Blackwell's Approachability

Nahum Shimkin

Faculty of Electrical Engineering

Technion—Israel Institute of Technology

Haiifa 32000, ISRAEL

SHIMKIN@EE.TECHNION.AC.IL

Abstract

The problem of approachability in repeated games with vector payoffs was introduced by Blackwell in the 1950s, along with geometric conditions and corresponding approachability strategies that rely on computing a sequence of *direction vectors* in the payoff space. For convex target sets, these vectors are obtained as projections from the current average payoff vector to the set. A recent paper by Abernethy, Batlett and Hazan (2011) proposed a class of approachability algorithms that rely on Online Linear Programming for obtaining alternative sequences of direction vectors. This is first implemented for target sets that are convex cones, and then generalized to any convex set by embedding it in a higher-dimensional convex cone. In this paper we present a more direct formulation that relies on general Online Convex Optimization (OCO) algorithms, along with basic properties of the support function of convex sets. This leads to a general class of approachability algorithms, depending on the choice of the OCO algorithm and the used norms. Blackwell's original algorithm and its convergence are recovered when Follow The Leader (or a regularized version thereof) is used for the OCO algorithm.

Keywords: approachability, online convex optimization, repeated games with vector payoffs

1. Introduction

Blackwell's approachability theory and the regret-based framework of online learning both address a repeated decision problem in the presence of an arbitrary (namely, unpredictable) adversary. Approachability, as introduced by Blackwell (1956), considers a fundamental feasibility issue for repeated matrix games with vector-valued payoffs. Referring to one player as the *agent* and to the other as *Nature*, a set S in the payoff space is *approachable* by the agent if it can ensure that the average payoff vector converges (with probability 1) to S , irrespectively of Nature's strategy. Blackwell's seminal paper provided geometric conditions for approachability, which are both necessary and sufficient for *convex* target sets S , and a corresponding approachability strategy for the agent. Approachability has found important applications in the theory of learning in games (Aumann and Maschler, 1995; Fudenberg and Levine,

1998; Peyton Young, 2004), and in particular in relation with no-regret strategies in repeated games as further elaborated below. A recent textbook exposition of approachability and some of its applications can be found in Maschler et al. (2013), and a comprehensive survey is provided by Perchet (2014).

Concurrently to Blackwell's paper, Hauman (1957) introduced the concept of no-regret strategies in the context of repeated matrix games. The *regret* of the agent is the shortfall of the cumulative payoff that was actually obtained relative to the one that would have been obtained with the best (fixed) action in hindsight, given Nature's observed action sequence. A no-regret strategy, or algorithm, ensures that the regret grows sub-linearly in time. The no-regret criterion has been widely adopted in the machine learning literature as a standard measure for the performance of online learning algorithms, and its scope has been greatly extended accordingly. Of specific relevance here is the Online Convex Optimization (OCO) framework, where Nature's discrete action is replaced by the choice of a convex function at each stage, and the agent's decision is a point in a convex set. The influential text of Cesa-Bianchi and Lugosi (2006) offers a broad overview of regret in online learning. Extensive surveys of OCO algorithms are provided by Shalev-Shwartz (2011); Hazan (2012, April 2016).

It is well known that no-regret strategies for repeated games can be obtained as particular instances of the approachability problem. A specific scheme was already given by Blackwell (1954), and an alternative formulation that leads to more explicit strategies was proposed by Hart and Mas-Colell (2001). The present paper considers the opposite direction, namely how no-regret algorithms for OCO can be used as a basis for an approachability strategy. Specifically, the OCO algorithm is used to generate a sequence of vectors that replace the projection-based direction vectors in Blackwell's algorithm. This results in a general class of approachability algorithms, that includes Blackwell's algorithm (and some generalizations thereof by Hart and Mas-Colell (2001)) as special cases.

The idea of using an online-algorithm to provide the sequence of direction vectors originated in the work of Abernethy et al. (2011), who showed how any no-regret algorithm for the online *linear* optimization problem can be used as a basis for an approachability algorithm. The scheme suggested in Abernethy et al. (2011) first considers target sets S that are convex cones. The generalization to any convex set is carried out by embedding the original target set as a convex cone in a higher dimensional payoff space. Here, we propose a more direct scheme that avoids the above-mentioned embedding. This construction relies on the *support function* of the target set, which is related to Blackwell's approachability conditions on the one hand, and on the other provides a variational expression for the point-to-set distance. Consequently, the full range of OCO algorithms can be used to provide a suitable sequence of direction vectors.

As we shall see, Blackwell's original algorithm is recovered from our scheme when the standard Follow the Leader (FTL) algorithm is used for the OCO part. Recovering the (known) convergence of this algorithm directly from the OCO viewpoint is a bit

more intricate. First, when the target set has a smooth boundary, we show that FTL converges at a “fast” (logarithmic) rate, hence leading to a correspondingly fast convergence of the average reward to the target set. To address the general case, we further show that Blackwell’s algorithm is still exactly recovered when an appropriately *regularized* version of FTL is used, from which the standard $O(T^{-1/2})$ convergence rate may be deduced.

The basic results of approachability theory have been extended in numerous directions. These include additional theoretical results, such as the characterization of non-convex approachable sets; extended models, such as stochastic (Markov) games and games with partial monitoring; and additional approachability algorithms for the basic model. For concreteness we will expand only on the latter (below, in Subsection 2.1), and refer the reader to the above-mentioned overviews for further information.

The paper proceeds as follows. In Section 2 we review the relevant background on Blackwell’s approachability and Online Convex Optimization. Section 3 presents the proposed scheme, in the form of a meta-algorithm that relies on a generic OCO algorithm, discusses the relation to the scheme of Abernethy et al. (2011), and demonstrates a specific algorithm that is obtained by using Generalized Gradient Descent for the OCO algorithm. In Section 4 we describe the relations with Blackwell’s original algorithm and its convergence. Section 5 outlines the extension of the proposed framework to general (rather than Euclidean) norms, followed by some concluding remarks.

Notation: The standard (dot) inner product in \mathbb{R}^d is denoted by $\langle \cdot, \cdot \rangle$, $\| \cdot \|_2$ is the Euclidean norm, $d(z, S) = \inf_{s \in S} \|z - s\|_2$ denotes the corresponding point-to-set distance, $B_2 = \{w \in \mathbb{R}^d : \|w\|_2 \leq 1\}$ denotes the Euclidean unit ball, $\Delta(I)$ is the set of probability distributions over a finite set I , $\text{diam}(S) = \sup_{s, s' \in S} \|s - s'\|_2$ is the diameter of the set S , and $\|\mathcal{R} - S\|_2 = \sup_{s \in \mathcal{R}, s' \in S} \|s - s'\|_2$ denotes the maximal distance between points in sets \mathcal{R} and S .

2. Model and Background

We start with brief reviews of Blackwell’s approachability theory and Online Convex Programming, focusing on those aspects that are most relevant to this paper.

2.1 Approachability

Consider a repeated game with *vector-valued* rewards that is played by two players, the *agent* and *Nature*. Let I and J denote the finite action sets of these players, respectively, with corresponding mixed actions $x = (x(I), \dots, x(\{I\})) \in \Delta(I)$ and $y = (y(1), \dots, y(\{J\})) \in \Delta(J)$. Let $r : I \times J \rightarrow \mathbb{R}^d$, $d \geq 1$, be the vector-valued reward function of the single-stage game, which is extended to mixed action as usual through the bilinear function

$$r(x, y) = \sum_{i,j} x(i)y(j)r(i, j).$$

Similarly, we denote $r(x, j) = \sum_i x(i)r(i, j)$. The specific meaning of $r(\cdot, \cdot)$ should be clear by its argument.

The game is repeated in stages $t = 1, 2, \dots$, where at stage t actions i_t and j_t are chosen by the players, and the reward vector $r(i_t, j_t)$ is obtained. A pure strategy for the agent is a mapping from each possible history $(i_1, j_1, \dots, i_{t-1}, j_{t-1})$ to an action i_t , and a mixed strategy is a probability distribution over the pure strategies. Nature’s strategies are similarly defined. Any pair of strategies for the agent and Nature thus induce a probability measure on the game sequence $(i_t, j_t)_{t=1}^{\infty}$.

Let

$$\bar{r}_T = \frac{1}{T} \sum_{t=1}^T r(i_t, j_t)$$

denote the T -stage average reward vector. We may now recall Blackwell’s definition of an approachable set.

Definition 1 (Approachability) *A set $S \subset \mathbb{R}^d$ is **approachable** if there exists a strategy for the agent such that \bar{r}_t converges to S with probability 1, at a uniform rate over Nature’s strategies. That is, for any $\epsilon > 0$ and $\delta > 0$ there exists $T \geq 1$ such that*

$$\text{Prob} \left\{ \sup_{t \geq T} d(\bar{r}_t, S) > \epsilon \right\} \leq \delta, \quad (1)$$

for any strategy of Nature. A strategy of the agent that satisfies this property is an approachability strategy for S .

Remarks:

1. It is evident that approachability of a set and its closure are equivalent, hence we shall henceforth consider only closed target sets S .
2. In some treatments of approachability, convergence of the expected distance $E(d(\bar{r}_t, S))$ and its rates are of central interest; see Perchet (2014). We shall consider these rates as well in the following.
3. In some models of interest, the decision variable of the agent may actually be the continuous variable x (in place of i), so that the actual reward is $r(x, j)$. All definitions and results below easily extend to this case, as long as x remains in a compact and convex set, and $r(x, j)$ is linear in x over that set.

For convex sets, approachability is fully characterized by the following result, which also provides an explicit strategy for the agent.

Theorem 2 (Blackwell, 1956) *A closed convex set $S \subset \mathbb{R}^d$ is approachable if and only if either one of the following equivalent conditions holds:*

(i) For each unit vector $u \in \mathbb{R}^d$, there exists a mixed action $x = x_S(u) \in \Delta(I)$ such that

$$\langle u, r(x, j) \rangle \leq \sup_{s \in S} \langle u, s \rangle, \quad \text{for all } j \in J. \quad (2)$$

(ii) For each $y \in \Delta(J)$ there exists $x \in \Delta(I)$ such that $r(x, y) \in S$.

If S is approachable, then the following strategy is an approachability strategy for S : For $z \notin S$, let $u_S(z)$ denote the unit vector that points to z from $\text{Proj}_S(z)$, the closest point to z in S . For $t \geq 2$, if $\bar{r}_{t-1} \notin S$, choose i_t according to the mixed action $x_t = x_S(u_S(\bar{r}_{t-1}))$; otherwise, choose i_t arbitrarily.

Blackwell's approachability strategy relies on the sequence of direction vectors $u_t = u_S(\bar{r}_{t-1})$, obtained through Euclidean projections onto the set S . A number of extensions and alternative algorithms for the basic game model have been proposed since. Most related to the present paper is the use of more general direction vectors. In Hart and Mas-Colell (2001), the direction vectors are obtained as the gradient of a suitable potential function; Blackwell's algorithm is recovered when the potential is taken as the Euclidean distance to the target set, while the use of other norms provides a range of useful variants. We will relate these variants to the present work in Section 5. As mentioned in the introduction, Abernethy et al. (2011) introduced the use of no-regret algorithms to generate the sequence of direction vectors.

A different class of approachability algorithms relies on Blackwell's dual condition in Theorem 2(ii), thereby avoiding the computation of direction vectors as projections (or related operations) to the target set S . Based on that condition, one can define a *response map* that assigns to each mixed action y of Nature a mixed action x of the agent such that the reward vector $r(x, y)$ belongs to S . An approachability algorithm that applies the response map to a *calibrated forecast* of the opponents' actions was proposed in Perchet (2009), and further analyzed in Bernstein et al. (2014). A computationally feasible response-based scheme that avoids the hard computation of calibrated forecasts is provided by Bernstein and Shimkin (2015). This paper also demonstrates the utility of the response-based approach for a class of generalized no-regret problems, where the set S is geometrically complicated, hence computing a projection is hard, but the response function is readily available. The response-based viewpoint is pursued further in the work of Mannor et al. (2014), which aims to approach the best-in-hindsight target set in an unknown game.

2.2 Online Convex Optimization (OCO)

OCO extends the framework of no-regret learning to function minimization. Let W be a convex and compact set in \mathbb{R}^d , and let \mathcal{F} be a set of convex functions $f : W \rightarrow \mathbb{R}$. Consider a sequential decision problem, where at each stage $t \geq 1$ the agent chooses a point $w_t \in W$, and then observes a function $f_t \in \mathcal{F}$. An *Algorithm* for the agent is a rule for choosing w_t , $t \geq 1$, based on the history $\{f_k, w_k\}_{k \leq t-1}$. The regret of an

algorithm \mathcal{A} is defined as

$$\text{Regret}_{\mathcal{T}}(\mathcal{A}) = \sup_{f_1, \dots, f_{\mathcal{T}} \in \mathcal{F}} \left\{ \sum_{t=1}^{\mathcal{T}} f_t(w_t) - \min_{w \in W} \sum_{t=1}^{\mathcal{T}} f_t(w) \right\}, \quad (3)$$

where the supremum is taken over all possible functions $f_t \in \mathcal{F}$. An effective algorithm should guarantee a small regret, and in particular one that grows sub-linearly in \mathcal{T} .

The OCO problem was introduced in this generality in Zinkevich (2003), along with the following Online Gradient Descent algorithm:

$$w_{t+1} = \text{Proj}_W(w_t - \eta_t g_t), \quad g_t \in \partial f_t(w_t). \quad (4)$$

Here $\partial f_t(w_t)$ is the subdifferential of f_t at w_t , (η_t) is a diminishing gain sequence, and Proj_W denotes the Euclidean projection onto the convex set W . To state a regret bound for this algorithm, let $\text{diam}(W)$ denote the diameter of W , and suppose that all subgradients of the functions f_t are uniformly bounded in norm by a constant G .

Proposition 3 (Zinkevich, 2003) For the Online Gradient Descent algorithm in (4) with gain sequence $\eta_t = \frac{\eta}{\sqrt{t}}$, $\eta > 0$, the regret is upper bounded as follows:

$$\text{Regret}_{\mathcal{T}}(\text{OGD}) \leq \left(\frac{\text{diam}(W)^2}{\eta} + 2\eta G^2 \right) \sqrt{\mathcal{T}}. \quad (5)$$

Several classes of OCO algorithms are now known, as surveyed in Cesa-Bianchi and Lugosi (2006); Shalev-Shwartz (2011); Hazan (2012). Of particular relevance here is the Regularized Follow the Leader (RFTL) algorithm, specified by

$$w_{t+1} = \arg \min_{w \in W} \left\{ \sum_{k=1}^t f_k(w) + R_t(w) \right\}, \quad (6)$$

where $R_t(w)$, $t \geq 1$ is a sequence of regularization functions. With $R_t \equiv 0$, the algorithm reduces to the basic Follow the Leader (FTL) algorithm, which does not generally lead to sublinear regret, unless additional requirements such as strong convexity are imposed on the functions f_t (we will revisit the convergence of FTL in Section 4). For RFTL, we will require the following standard convergence result. Recall that a function $R(w)$ over a convex set W is called ρ -strongly convex if $R(w) - \frac{\rho}{2} \|w\|_2^2$ is convex there.

Proposition 4 Suppose that each function f_t is Lipschitz-continuous over W , with Lipschitz coefficient L_f . Let $R_t(w) = \rho_t R(w)$, where $0 < \rho_t < \rho_{t+1}$, and the function $R : W \rightarrow [0, R_{\max}]$ is 1-strongly convex and Lipschitz continuous with coefficient L_R . Then

$$\text{Regret}_{\mathcal{T}}(\text{RFTL}) \leq 2L_f \sum_{t=1}^{\mathcal{T}} \frac{L_f + (\rho_t - \rho_{t-1})L_R}{\rho_t + \rho_{t-1}} + \rho_{\mathcal{T}} R_{\max}. \quad (7)$$

The proof of this bound is outlined in the Appendix.

3. OCO-Based Approachability

In this section we present the proposed OCO-based approachability algorithm. We start by introducing the support function and its relevant properties, and express Blackwell's separation condition in terms of this function. We then present the proposed algorithm, in the form of a meta-algorithm that incorporates a generic OCO algorithm. As a concrete example, we consider the specific algorithm obtained when Online Gradient Descent is used for the OCO part.

3.1 The Support Function

Let set $S \subset \mathbb{R}^d$ be a closed and convex set. The *support function* $h_S : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{\infty\}$ of S is defined as

$$h_S(w) \triangleq \sup_{s \in S} \langle w, s \rangle, \quad w \in \mathbb{R}^d.$$

It is evident that h_S is a convex function (as a pointwise supremum of linear functions), and is positive homogeneous: $h_S(aw) = ah_S(w)$ for $a \geq 0$. Furthermore, the Euclidean distance from a point $z \in \mathbb{R}^d$ to S can be expressed as

$$d(z, S) = \max_{w \in B_2} \{ \langle w, z \rangle - h_S(w) \}, \quad (8)$$

where B_2 is the closed Euclidean unit ball (see, e.g., Boyd and Vandenberghe (2004), Section 8.1.3); see also Lemma 16 below). It follows that

$$\operatorname{argmax}_{w \in B_2} \{ \langle w, z \rangle - h_S(w) \} = \begin{cases} 0 & : z \in S \\ u_S(z) & : z \notin S \end{cases} \quad (9)$$

with $u_S(z)$ as defined in Theorem 2, namely the unit vector pointing from $\operatorname{Proj}_S(z)$ to z .

Blackwell's separation condition in (2) can now be written in terms of the support function as follows:

$$\langle w, r(x, j) \rangle \leq \sup_{s \in S} \langle w, s \rangle \equiv h_S(w).$$

We can now rephrase the primal condition in Theorem 2 in the following form.

Corollary 5 *A closed and convex set S is approachable if and only if for every vector $w \in B_2$ there exists a mixed action $x \in \Delta(I)$ so that*

$$\langle w, r(x, j) \rangle - h_S(w) \leq 0, \quad \forall j \in J. \quad (10)$$

We note that equation (10) defines a linear inequality for x , so that a mixed action $x \in \Delta(I)$ that satisfies (10) for a given direction w can be computed using linear programming. More concretely, existence of a mixed action x that satisfies (10) can be equivalently stated as

$$\operatorname{val}(w \cdot r) \triangleq \min_{x \in \Delta(I)} \max_{j \in J} \langle w, r(x, j) \rangle \leq h_S(w),$$

where $\operatorname{val}(w \cdot r)$ is the minimax value of the matrix game with a scalar payoff that is obtained by projecting the reward vectors $r(i, j)$ onto w . Consequently, the mixed action x that satisfies (10) can be taken as a minimax strategy for the agent in this game.

3.2 The General Algorithm

The proposed algorithm (see Algorithm 1 below) builds on the following idea. First, we apply an OCO algorithm to generate a sequence of *direction vectors* $w_t \in B_2$, so that

$$\sum_{t=1}^T (\langle w_t, r_t \rangle - h_S(w_t)) \geq T \max_{w \in B_2} \{ \langle w, \bar{r} \rangle - h_S(w) \} - a(T), \quad (11)$$

where $r_t = r(x_t, j_t)$ is considered (within the OCO algorithm) an arbitrary vector that is revealed after w_t is specified, and $a(T)$ is of order $o(T)$. The mixed action $x_t \in \Delta(I)$, in turn, is chosen (after w_t is revealed) to satisfy (10), so that $\langle w_t, r(x_t, j_t) \rangle - h_S(w_t) \leq 0$, hence

$$\langle w_t, r_t \rangle - h_S(w_t) \leq \langle w_t, r(x_t, j_t) \rangle \triangleq \delta_t.$$

Using this inequality in (11), and observing the distance formula (8), yields

$$d(\bar{r}, S) \leq \frac{a(T)}{T} + \Delta(T) \rightarrow 0,$$

where $\Delta(T) = \frac{1}{T} \sum_{t=1}^T \delta_t$, a stochastic term that converges to 0, as discussed below.

To secure (11), observe that the function $f(w; r) = -\langle w, r \rangle + h_S(w)$ is convex in w for each vector r . Therefore, an OCO algorithm can be applied to the sequence of convex functions $f_t(w) = -\langle w, r_t \rangle + h_S(w)$, where $r_t = r(x_t, j_t)$ is considered an arbitrary vector which is revealed only after w_t is specified. Applying an OCO algorithm \mathcal{A} with $\operatorname{Regret}_T(\mathcal{A}) \leq a(T)$ to this setup, we obtain a sequence (w_t) such that

$$\sum_{t=1}^T f_t(w_t) \leq \min_{w \in B_2} \sum_{t=1}^T f_t(w) + a(T),$$

where

$$\begin{aligned} \sum_{t=1}^T f_t(w_t) &= -\sum_{t=1}^T (\langle w_t, r_t \rangle - h_S(w_t)), \\ \sum_{t=1}^T f_t(w) &= -\sum_{t=1}^T (\langle w, r_t \rangle - h_S(w)) = -T(\langle w, \bar{r} \rangle - h_S(w)). \end{aligned}$$

This can be seen to imply (11).

The discussion above leads to the following generic approachability algorithm.

Algorithm 1 (OCO-based Approachability Meta-Algorithm)

- Given: A closed, convex and approachable set S ; a procedure (e.g., a linear program) to compute $x \in \Delta(I)$, for a given vector w , so that (10) is satisfied; an OCO algorithm \mathcal{A} for the functions $f_t(w) = -\langle w, r_t \rangle + h_S(w)$, with $\text{Regret}_T(\mathcal{A}) \leq a(T)$.
- Repeat for $t = 1, 2, \dots$:
 1. Obtain w_t from the OCO algorithm applied to the convex functions $f_k(w) = -\langle w, r_k \rangle + h_k(w)$, $k \leq t - 1$, so that inequality (11) is satisfied.
 2. Choose x_t according to (10), so that $\langle w_t, r(x_t, j) \rangle - h_S(w_t) \leq 0$ holds for all $j \in J$.
 3. Observe Nature's action j_t , and set $r_t = r(x_t, j_t)$.

To state our convergence result for this algorithm, we first consider the term Δ_t that arises due to the difference between $r_t = r(i_t, j_t)$ and $r(x_t, j_t)$. The analysis follows by standard convergence results for martingale difference sequences.

Lemma 6 *Let*

$$\Delta_T = \frac{1}{T} \sum_{t=1}^T \delta_t, \quad \delta_t = \langle w_t, r(i_t, j_t) \rangle - r(x_t, j_t).$$

Then $E(\Delta_T) = 0$, and $\Delta_T \rightarrow 0$ w.p. 1, at a uniform rate independent of Nature's strategy. Specifically,

$$P\{|\Delta_T| > \epsilon\} \leq \frac{6\rho_0^2}{\epsilon^2 T}, \quad (12)$$

where $\rho_0 = \max_{j \in J} \max_{i, i' \in I} \|r(i, j) - r(i', j)\|_2$.

Proof Let $H_t = \langle i_t, j_t, w_t \rangle_{1 \leq k \leq t}$. Observe that w_t and j_t are chosen based only on H_{t-1} , hence do not depend on i_t , and similarly i_t is randomly and independently chosen according to x_t . It follows that $E(\delta_t | H_{t-1}) = 0$, which implies that $E(\Delta_T) = 0$. Furthermore, (δ_t) is a Martingale difference sequence, uniformly bounded by

$$|\delta_t| \leq \|w_t\|_2 \|r(i_t, j_t) - r(x_t, j_t)\|_2 \leq \max_{j \in J} \max_{i, i'} \|r(i, j) - r(i', j)\|_2 \triangleq \rho_0,$$

(where $w_t \in B_2$ was used in the second inequality). Convergence of Δ_t now follows by standard results for martingale difference sequences; the specific rate bound in (12) follows from Proposition 4.1 and Equation (4.7) in Shimkin and Shwartz (1993), upon noting that $X_t \triangleq \sum_{k=1}^t \delta_k$ satisfies $E(X_{t+1}^2 | H_t) = X_t^2 + 0 + E(\delta_{t+1}^2 | H_t) \leq X_t^2 + \rho_0^2$. ■

Convergence of Algorithm 1 may now be summarised as follows.

Theorem 7 *Under Algorithm 1, for any $T \geq 1$ and any strategy of the opponent, it holds w.p. 1 that*

$$d(\bar{r}_T, S) \leq \frac{a(T)}{T} + \Delta_T,$$

where Δ_T is defined in Lemma 6 and is a zero-mean random variable that converges to zero at a uniform rate, as specified there. In particular,

$$E(d(\bar{r}_T, S)) \leq \frac{a(T)}{T}.$$

Proof As observed above, application of the OCO algorithm implies (11). Recalling (8), we obtain

$$\begin{aligned} d(\bar{r}_T, S) &= \max_{w \in B_2} \{ \langle w, \bar{r}_T \rangle - h_S(w) \} \\ &\leq \frac{1}{T} \sum_{t=1}^T (\langle w_t, r_t \rangle - h_S(w_t)) + \frac{a(T)}{T} \\ &= \frac{1}{T} \sum_{t=1}^T (\langle w_t, r(x_t, j_t) \rangle - h_S(w_t)) + \frac{a(T)}{T} + \frac{1}{T} \sum_{t=1}^T (w_t, r_t - r(x_t, j_t)). \end{aligned}$$

But since $\langle w_t, r(x_t, j_t) \rangle - h_S(w_t) \leq 0$ by choice of x_t in the algorithm, and using the definition of Δ_t , we obtain that $d(\bar{r}_T, S) \leq \frac{a(T)}{T} + \Delta_T$, as claimed. The rest now follows by the properties of Δ_t , stated in Lemma 6. ■

To recap, any OCO algorithm that guarantees (11) with $\frac{a(T)}{T} \rightarrow 0$, induces an approachability strategy with rate of convergence bounded by the sum of two terms: the first is $\frac{a(T)}{T}$, related to the regret bound of the OCO algorithm, and the second is a zero-mean stochastic term of order $T^{-1/2}$ (at most), which arises due to the difference between the actual rewards $r_t = r(i_t, j_t)$ and their means $r(x_t, j_t)$.

We conclude this subsection with a few remarks. The first two concern instances where the stochastic term Δ_T is nullified.

Remark 8 (Pure Actions) *Suppose that the inequality $\max_j \langle w_t, r(x, j) \rangle - h_S(w_t) \leq 0$ in step 2 of the Algorithm can always be satisfied by pure actions (so that x_t assigns probability 1 to a single action, i_t). Then choosing such x_t 's clearly implies that $r(x_t, j_t) = r(i_t, j_t)$, hence the term Δ_t in Theorem 7 becomes identically zero.*

Remark 9 (Smooth Rewards) *In some problems, the rewards of interest may actually be the smoothed rewards $r(x_t, j_t)$ or $r(x_t, y_t)$, rather than $r(i_t, j_t)$. Focusing on the first case for concreteness, let us redefine r_t as $r(x_t, j_t)$, and assume that this reward vector can be computed or observed by the agent following each stage t . Applying Algorithm 1 with these modified rewards now leads to the same bound as in Theorem 7, but with $\Delta_T = 0$.*

Remark 10 (Convex Cones) The approachability algorithm of Abernethy et al. (2011) starts with target sets S that are restricted to be convex cones. For S a closed convex cone, the support function is given by

$$h_S(w) = \begin{cases} 0 & : w \in S^\circ \\ \infty & : w \notin S^\circ \end{cases}$$

where S° is the polar cone of S . The required inequality in (11) thus reduces to

$$\sum_{t=1}^T \langle u_t, r_t \rangle \geq T \max_{w \in B_2 \cap S^\circ} \langle w, \bar{r}_T \rangle - a(T).$$

The sequence (u_t) can be obtained in this case by applying an online linear optimization algorithm restricted to $w_t \in B_2 \cap S^\circ$. This is the algorithm proposed by Abernethy et al. (2011). The extension to general convex sets is handled there by lifting the problem to a $(d+1)$ -dimensional space, with payoff vector $r^t(x, y) = (\kappa, r^t(x, y))$ and target set $S' = \text{cone}(\kappa \times S)$, where $\kappa = \max_{s \in S} \|s\|_2$, for which it holds that $d(u, S) \leq 2d(u', S')$.

3.3 An OGD-based Approachability Algorithm

As a concrete example, let us apply the Online Gradient Descent algorithm specified in (4) to our problem. With $W = B_2$ and $f_t^i(w) = -\langle w, r_t \rangle - h_S(w)$, we obtain in step 1 of Algorithm 1,

$$w_{t+1} = \text{Proj}_{B_2} \{w_t + \eta_t(r_t - y_t)\}, \quad y_t \in \partial h_S(w_t).$$

Observe that $\text{Proj}_{B_2}(v) = v / \max\{1, \|v\|_2\}$, and (e.g., by Corollary 8.25 in Rockafellar and Wets (1997))

$$\partial h_S(w) = \arg\max_{s \in S} \langle s, w \rangle.$$

To evaluate the convergence rate in (5), observe that $\text{diam}(B_2) = 2$, and, since $y_t \in S$, $\|y_t\|_2 = \|r_t - y_t\|_2 \leq \|\mathcal{R} - S\|_2$, where $\mathcal{R} = \{r(x, y) : x \in \Delta(U), y \in \Delta(U)\}$ is the reward set. Assuming for the moment that the goal set S is bounded, we obtain

$$E(d(\bar{r}_T, S)) \leq \frac{b(\eta)}{\sqrt{T}}, \quad \text{with } b(\eta) = \frac{4}{\eta} + 2\eta \|\mathcal{R} - S\|_2^2.$$

For $\eta = \sqrt{2} \|\mathcal{R} - S\|_2$, we thus obtain $b(\eta) = 4\sqrt{2} \|\mathcal{R} - S\|_2$.

If S is not bounded, it can always be intersected with \mathcal{R} (without affecting its approachability), yielding $\|\mathcal{R} - S\|_2 \leq \text{diam}(\mathcal{R})$. This amounts to modifying the choice of y_t in the algorithm to

$$y_t \in \partial h_{S \cap \mathcal{R}}(w_t) = \arg\max_{y \in S \cap \mathcal{R}} \langle y, w_t \rangle.$$

Alternatively, one may restrict attention (by projection) to vectors u_t in the set $\{w \in B_2 : h_S(w) < \infty\}$, similarly to the case of convex cones mentioned in Remark 10 above; we will not go here into further details.

4. Blackwell's Algorithm and (R)FTL

We next examine the relation between Blackwell's approachability algorithm and the proposed OCO-based scheme. We first show that Blackwell's algorithm coincides with OCO-based approachability when FTL is used as the OCO algorithm. We use this equivalence to establish fast (logarithmic) convergence rates for Blackwell's algorithm when the target set S has a smooth boundary. Interestingly, this equivalence does not provide a convergence result for general convex sets. To complete the picture, we show that Blackwell's algorithm can more generally be obtained via a *regularized* version of FTL, which leads to an alternative proof of convergence of the algorithm in the general case.

4.1 Blackwell's algorithm as FTL

Recall Blackwell's algorithm as specified in Theorem 2, namely x_{t+1} is chosen as a mixed action that satisfies (2) for $u = u_S(\bar{r}_t)$ (with x_{t+1} chosen arbitrarily if $\bar{r}_t \in S$, which is equivalent to setting $u = 0$ in that case).

Similarly, in Algorithm 1, x_{t+1} is chosen as a mixed action that satisfies (2) for $u = w_{t+1}$. Using FTL (i.e., Equation (6) with $R_t = 0$) for the OCO part gives

$$w_{t+1} = \arg\min_{w \in B_2} \sum_{k=1}^t f_k(w), \quad \text{with } f_k(w) = -\langle w, r_k \rangle + h_S(w).$$

Equivalence of the two algorithms now follows directly from the following observation.

Lemma 11 *With $f_k(w)$ as above,*

$$\arg\min_{w \in B_2} \sum_{k=1}^t f_k(w) = \begin{cases} u_S(\bar{r}_t) & : \bar{r}_t \notin S \\ 0 & : \bar{r}_t \in S \end{cases}.$$

Proof Observe that $\sum_{k=1}^t f_k(w) = -t(\langle w, \bar{r}_t \rangle - h_S(w))$, so that

$$\arg\min_{w \in B_2} \sum_{k=1}^t f_k(w) = \arg\max_{w \in B_2} \{w, \bar{r}_t\} - h_S(w).$$

The required equality now follows from (9). \blacksquare

To establish convergence of Blackwell's algorithm via this equivalence, one needs to show that FTL guarantees the regret bound in (11) for an arbitrary reward sequence $(r_t) \subset \mathcal{R}$, with a sublinear rate sequence $a(T)$. It is well known, however, that (unregularized) FTL does not guarantee sublinear regret, without some additional assumptions on the function f_t . A simple counter-example, reformulated to the present case, is devised as follows: Let $S = \{0\} \subset \mathbb{R}$, so that $h_S(w) = 0$, and suppose that

$r_t = -1$ and $r_t = 2(-1)^t$ for $t > 1$. Since $w_t = \text{sign}(\bar{r}_{t-1})$ and $\text{sign}(r_t) = -\text{sign}(\bar{r}_{t-1})$, we obtain that $f_t(w_t) = -r_t w_t = 1$, leading to a linearly-increasing regret.

The failure of FTL in this example is clearly due to the fast changes in the predictors w_t . We now add some smoothness assumptions on the set S that can mitigate such abrupt changes.

Assumption 1 *Let S be a compact and convex set. Suppose that the boundary ∂S of S is smooth with curvature bounded by κ_0 , namely:*

$$\|\bar{\pi}(s_1) - \bar{\pi}(s_2)\|_2 \leq \kappa_0 \|s_1 - s_2\|_2 \quad \text{for all } s_1, s_2 \in \partial S, \quad (13)$$

where $\bar{\pi}(s)$ is the unique unit outer normal to S at $s \in \partial S$.

For example, for a closed Euclidean ball of radius ρ , (13) is satisfied with equality for $\kappa_0 = \rho^{-1}$. The assumed smoothness property may in fact be formulated in terms of an interior sphere condition: For any point in $s \in S$ there exists a ball $B(\rho) \subset S$ with radius $\rho = \kappa_0^{-1}$ such that $s \in B(\rho)$.

Proposition 12 *Let Assumption 1 hold. Consider Blackwell's algorithm as specified in Theorem 2, and denote $w_t = u_S(\bar{r}_{t-1})$ (with w_1 arbitrary). Then, for any time $T \geq 1$ such that $\bar{r}_T \notin S$, (11) holds with*

$$a(T) = C_0(1 + \ln T), \quad (14)$$

where $C_0 = \text{diam}(\mathcal{R}) \|\mathcal{R} - S\|_2 \kappa_0$, and $\ln(\cdot)$ is the natural logarithm. Consequently,

$$E(\text{d}(\bar{r}_T, S)) \leq C_0 \frac{1 + \ln T}{T}, \quad T \geq 1. \quad (15)$$

Proof Observe first that the regret bound in (14) implies (15). Indeed, for $\bar{r}_T \notin S$, $\text{d}(\bar{r}_T, S) \leq a(T)/T$ follows as in Theorem 7, while if $\bar{r}_T \in S$ then $\text{d}(\bar{r}_T, S) = 0$ and (15) holds trivially.

We proceed to establish the logarithmic regret bound in (14). Let $f_t(w) = -\langle w, r_t \rangle + h_S(w)$, $W = B_2$, and denote

$$\text{Regret}_T(f_{1:T}) = \sum_{t=1}^T f_t(w_t) - \min_{w \in W} \sum_{t=1}^T f_t(w) = \sum_{t=1}^T (f_t(w_t) - f_t(w_{T+1})). \quad (16)$$

A standard induction argument (e.g., Lemma 2.1 in Shalev-Shwartz (2011)) verifies that

$$\sum_{t=1}^T (f_t(w_t) - f_t(u)) \leq \sum_{t=1}^T (f_t(w_t) - f_t(w_{t+1})) \quad (17)$$

holds for any $u \in W$, and in particular for $u = w_{T+1}$. It remains to upper-bound the differences in the last sum.

Consider first the case where $\bar{r}_t \notin S$ for all $1 \leq t \leq T$. We first show that $\|w_t - w_{t+1}\|_2$ is small, which implies the same for $|f_t(w_t) - f_t(w_{t+1})|$. By its definition, $w_{t+1} = u_S(\bar{r}_t)$, the unit vector pointing to \bar{r}_t from $c_t \stackrel{\Delta}{=} \text{Proj}_S(\bar{r}_t)$, which clearly coincides with the outer unit normal $\bar{\pi}(c_t)$ to S at c_t . It follows that

$$\|w_t - w_{t+1}\|_2 = \|\bar{\pi}(c_{t-1}) - \bar{\pi}(c_t)\|_2 \leq \kappa_0 \|c_{t-1} - c_t\|_2 \leq \kappa_0 \|\bar{r}_{t-1} - \bar{r}_t\|_2,$$

where the first inequality follows by Assumption 1, and the second due to the shrinking property of the projection. Substituting $\bar{r}_t = \bar{r}_{t-1} + \frac{1}{t}(r_t - \bar{r}_{t-1})$ obtains

$$\|w_t - w_{t+1}\|_2 \leq \frac{\kappa_0}{t} \|\bar{r}_t - \bar{r}_{t-1}\|_2 \leq \frac{\kappa_0}{t} \text{diam}(\mathcal{R}). \quad (18)$$

Next, observe that for any pair of unit vectors w_1 and w_2 ,

$$\begin{aligned} f_t(w_1) - f_t(w_2) &= -\langle w_1 - w_2, r_t \rangle + h_S(w_1) - h_S(w_2) \\ &= -\langle w_1 - w_2, r_t \rangle + \max_{s \in S} \langle w_1, s \rangle - \max_{s \in S} \langle w_2, s \rangle \\ &\leq -\langle w_1 - w_2, r_t \rangle + \langle w_1, s_1 \rangle - \langle w_2, s_1 \rangle \\ &= \langle w_1 - w_2, s_1 - r_t \rangle \leq \|w_1 - w_2\|_2 \|\mathcal{R} - S\|_2, \end{aligned}$$

where $s_1 \in S$ attains the first maximum. Since the same bound holds for $f_t(w_2) - f_t(w_1)$, it holds also for the absolute value. In particular,

$$|f_t(w_t) - f_t(w_{t+1})| \leq \|w_t - w_{t+1}\|_2 \|\mathcal{R} - S\|_2, \quad (19)$$

and together with (18) we obtain

$$|f_t(w_t) - f_t(w_{t+1})| \leq \frac{\kappa_0}{t} \text{diam}(\mathcal{R}) \|\mathcal{R} - S\|_2 = \frac{C_0}{t}.$$

Substituting in (17) and summing over t^{-1} yields the regret bound

$$\text{Regret}_T(f_{1:T}) \leq C_0(1 + \ln T). \quad (20)$$

We next extend this bound to case where $\bar{r}_t \in S$ for some t . In that case $w_{t+1} = 0$, and $w_t - w_{t+1}$ may not be small. However, since $f_t(0) = 0$, such terms will not affect the sum in (17). Recall that we need to establish (14) for T such that $\bar{r}_T \notin S$. In that case, any time t for which $\bar{r}_t \in S$ is followed by some time $m \leq T$ with $\bar{r}_m \notin S$. Let $1 \leq k < m \leq T$ be indices such that $\bar{r}_k, \dots, \bar{r}_{m-1} \in S$, but $\bar{r}_{k-1} \notin S$ (or $k = 1$) and $\bar{r}_m \notin S$. Then $w_{k+1}, \dots, w_m = 0$, and

$$\sum_{t=k}^m (f_t(w_t) - f_t(w_{t+1})) = f_k(w_k) - f_m(w_{m+1}).$$

Proceeding as above, we obtain similarly to (18),

$$\|w_k - w_{m+1}\|_2 \leq \kappa_0 \|\bar{r}_{k-1} - \bar{r}_m\|_2 \leq \text{diam}(\mathcal{R}) \sum_{t=k}^{m-1} \frac{\kappa_0}{t},$$

and the regret bound in (20) may be obtained as above. \blacksquare

The last result establishes a fast convergence rate (of order $\log T/T$) for Blackwell's approachability algorithm, under the assumed smoothness of the target set. We note that conditions for fast approachability (of order T^{-1}) were derived in Perchet and Mannor (2013), but are of different nature than the above.

Logarithmic convergence rates were derived for OCO algorithms in Hazan et al. (2007), under strong convexity conditions on the function f_t . This is apparently related to the present result, especially given the equivalence between strong convexity of a function and strong smoothness of its Legendre-Fenchel transform (cf. Shalev-Shwartz (2011), Lemma 2.19). However, we observe that the support function is *not* strongly convex, so that the logarithmic regret bound in (20) does not seem to follow from existing results. Rather, a basic property which underlies both cases is insensitvity of the maximum point to small perturbations in f_t , which here leads to the inequality (18).

4.2 Blackwell's algorithm as RFTL

The smoothness requirement in Assumption 1 does not hold for important classes of target sets, such as polyhedra and cones. As observed above, in absence of such additional smoothness properties the interpretation of Blackwell's algorithm through an FTL scheme does not entail its convergence, as the regret of FTL (and the corresponding bound $a(T)$ in (11)) might increase linearly in general.

To accommodate general (non-smooth) sets, we show next that Blackwell's algorithm can be identified more generally with a *regularized* version of FTL. This algorithm does guarantee an $O(\sqrt{T})$ regret in (11), and consequently leads to the standard $O(T^{-1/2})$ rate of convergence of Blackwell's approachability algorithm.

Let us apply the RFTL algorithm in equation (6) as the OCO part in Algorithm 1, with a quadratic regularization function $R_t(w) = \frac{\rho_t}{2} \|w\|_2^2$. This gives

$$w_{t+1} = \underset{w \in B_2}{\operatorname{argmin}} \left\{ \sum_{k=1}^t f_k(w) + \frac{\rho_t}{2} \|w\|_2^2 \right\}, \quad f_k(w) = -\langle w, r_k \rangle + h_S(w).$$

The following equality is the key to the required equivalence. It relies essentially on the positive-homogeneity property of the support function h_S , and consequently of the functions f_k above.

Lemma 13 For $\rho_t > 0$, and w_{t+1} as defined above,

$$w_{t+1} = \begin{cases} \beta_t u_S(\bar{r}_t) & : \bar{r}_t \notin S \\ 0 & : \bar{r}_t \in S \end{cases}, \quad (21)$$

where $\beta_t = \min\{1, \frac{t}{\rho_t} d(\bar{r}_t, S)\} > 0$.

Proof Recall that $\sum_{k=1}^t f_k(w) = -t\langle w, \bar{r}_t \rangle - h_S(w)$, so that

$$\underset{w \in B_2}{\operatorname{argmin}} \left\{ \sum_{k=1}^t f_k(w) + \frac{\rho_t}{2} \|w\|_2^2 \right\} = \underset{w \in B_2}{\operatorname{argmax}} \left\{ \langle w, \bar{r}_t \rangle - h_S(w) - \frac{\rho_t}{2t} \|w\|_2^2 \right\}.$$

To compute the right-hand side, we first maximize over $\{w : \|w\|_2 = \beta\}$, and then optimize over $\beta \in [0, 1]$. Denote $z = \bar{r}_t$, and $\eta = \rho_t/t$. Similarly to Lemma 11,

$$\underset{\|w\|_2=\beta}{\operatorname{argmax}} \left\{ \langle w, z \rangle - h_S(w) - \frac{\eta}{2} \|w\|_2^2 \right\} = \underset{\|w\|_2=\beta}{\operatorname{argmax}} \{ \langle w, z \rangle - h_S(w) \} = \begin{cases} \beta u_S(z) & : z \notin S \\ 0 & : z \in S \end{cases}.$$

Now, for $z \notin S$,

$$\max_{\|w\|_2=\beta} \left\{ \langle w, z \rangle - h_S(w) - \frac{\eta}{2} \|w\|_2^2 \right\} = \beta d(z, S) - \frac{\eta}{2} \beta^2.$$

Maximizing the latter over $0 \leq \beta \leq 1$ gives $\beta^* = \min\{1, \frac{d(z,S)}{\eta}\}$. Substituting back z and η gives (21). \blacksquare

This immediately leads to the required conclusion.

Proposition 14 Algorithm 1 with quadratically regularized FTL is equivalent to Blackwell's algorithm.

Proof Observe that the vector w_{t+1} in Equation (21) is equal to $u_S(\bar{r}_t)$ from Blackwell's algorithm in Theorem 2, up to a positive scaling by β_t . This scaling does not affect the choice of x_{t+1} according to (10), as the support function $h_S(w)$ is positive homogeneous. \blacksquare

Compared to non-regularized FTL (or Blackwell's algorithm), we see the direction vectors in Equation (21) are scaled by a positive constant. Essentially, the effect of this scaling is to reduce the magnitude of w_{t+1} when \bar{r}_t is close to S . While such scaling does not affect the choice of action x_t , it does lead to sublinear-regret for the OLO algorithm, and consequently convergence of the approachability algorithm. This is summarized as follows.

Proposition 15 *Let S be a convex and compact set. Consider the RFTL algorithm specified in equation (21), with $\rho_t = \rho\sqrt{t}$, $\rho > 0$. The regret of this algorithm is bounded by*

$$\text{Regret}_T(\text{RFTL}) \leq \left(\frac{2L_f^2}{\rho} + \frac{\rho}{2} \right) \sqrt{T} + \frac{L_f}{2} \ln(T) + 4L_f \triangleq a_0(T),$$

where $L_f = \|\mathcal{R} - S\|_2$. Consequently, if this RFTL algorithm is used in step 1 of Algorithm 1 to compute w_t , we obtain

$$E(\text{d}(\bar{r}_T, S)) \leq \frac{a_0(T)}{T} = O(T^{-\frac{1}{2}}), \quad T \geq 1. \quad (22)$$

Proof The regret bound follows from the one in Proposition 4, evaluated for $f_t(w) = -\langle r_t, w \rangle + h_S(w)$, $W = B_2$, $R(w) = \frac{1}{2}\|w\|_2^2$, and $\rho_t = \rho\sqrt{t}$. Recalling that $\partial f_t(w) = -r_t + \text{argmax}_{s \in S} \langle w, s \rangle$, the Lipschitz constant of f_t is upper bounded by $\|\mathcal{R} - S\|_2 \triangleq L_f$. Furthermore, $R_{\max} = \frac{1}{2}$ and $L_R = 1$. Therefore,

$$\text{Regret}_T(\text{RFTL}) \leq 2L_f \sum_{t=1}^T \frac{L_f + \rho(\sqrt{t} - \sqrt{t-1})}{\rho(\sqrt{t} + \sqrt{t-1})} + \frac{\rho}{2}\sqrt{T}. \quad (23)$$

To upper bound the sum, we note that

$$\sum_{t=1}^T \frac{1}{(\sqrt{t} + \sqrt{t-1})} = \sum_{t=1}^T (\sqrt{t} - \sqrt{t-1}) = \sqrt{T},$$

and

$$\begin{aligned} \sum_{t=1}^T \left(\frac{\sqrt{t} - \sqrt{t-1}}{\sqrt{t} + \sqrt{t-1}} \right) &= \sum_{t=1}^T \frac{1}{(\sqrt{t} + \sqrt{t-1})^2} \leq 2 + \sum_{t=3}^T \frac{1}{(2\sqrt{t-1})^2} \\ &\leq 2 + \frac{1}{4} \int_{t=1}^T \frac{1}{t} dt = 2 + \frac{1}{4} \ln(T). \end{aligned}$$

Substituting in (23) gives the stated regret bound. The second part now follows directly from Theorem 7. \blacksquare

With $\rho = 2L_f$, we obtain in (22) the convergence rate

$$E(\text{d}(\bar{r}_T, S)) \leq \frac{2\|\mathcal{R} - S\|_2}{\sqrt{T}} + o\left(\frac{1}{\sqrt{T}}\right).$$

We emphasize that the algorithm discussed in this section is equivalent to Blackwell's algorithm, hence its convergence is known. The proof of convergence here

is certainly not the simplest, nor does it lead to the best constants in the convergence rate. Indeed, Blackwell's proof (which recursively bounds the square distance $\text{d}(\bar{r}_T, S)^2$) leads to the bound $\sqrt{E(\text{d}(\bar{r}_T, S)^2)} \leq \frac{\|\mathcal{R} - S\|_2}{\sqrt{T}}$. Rather, our main purpose was to provide an alternative view and analysis of Blackwell's algorithm, which rely on a standard OCO algorithm. That said, the logarithmic convergence rate of the expected distance that was obtained under the smoothness Assumption 1 appears to be new.

5. Extensions with General Norms

As was mentioned in Subsection 2.1, a class of approachability algorithms that generalizes Blackwell's strategy was introduced by Hart and Mas-Colell (2001). The direction vectors (u_t) in Blackwell's algorithm, that are defined through Euclidean projection, are replaced in that paper by the gradient of a smooth *potential function*; Blackwell's algorithm is recovered when the potential is taken as the Euclidean distance to the target set. Other instances of interest are obtained by defining the potential through the p -norm distance; this, in turn, was used as a basis for a general class of no-regret algorithms in repeated games.

In this section we provide an extension of the OCO-based approachability algorithm from Section 3, which relies on a general norm rather than the Euclidean one to obtain the direction vectors (w_t) . The proposed algorithms coincide with those of Hart and Mas-Colell (2001) when the RFTL algorithm is used for the OCO part.

Let $\|\cdot\|$ denote some norm on \mathbb{R}^d . The dual norm, denoted $\|\cdot\|_*$, is defined as

$$\|x\|_* = \max_{w \in \mathbb{R}^d: \|w\| \leq 1} \langle w, x \rangle.$$

For example, if the primal norm is the p -norm $\|x\|_p = (\sum_{i=1}^d x_i^p)^{\frac{1}{p}}$ with $p \in (0, 1)$, the dual norm is the q -norm, with $q \in (0, 1)$ that satisfies $\frac{1}{p} + \frac{1}{q} = 1$.

The following relations between the support function h_S and the point-to-set distance d_* will be required. The first is needed to show convergence of the algorithm, and the second for the interpretation of the FTL-based variant.

Lemma 16 *Let S be a closed convex set with support function h_S , and let $d_*(z, S) = \min_{s \in S} \|z - s\|_*$ denote the point-to-set distance with respect to the dual norm. Then, for any $z \in \mathbb{R}^d$,*

$$d_*(z, S) = \max_{\|w\| \leq 1} \{ \langle w, z \rangle - h_S(w) \}, \quad (24)$$

and

$$\partial d_*(z, S) = \text{argmax}_{\|w\| \leq 1} \{ \langle w, z \rangle - h_S(w) \}, \quad (25)$$

where $\partial d_*(z, S)$ is the subgradient of $d_*(\cdot, S)$ at z .

Proof We first note that the maximum in (24) is attained since h_S is a lower semi-continuous function, and $\{\|w\| \leq 1\}$ is a compact set. To establish (24) we invoke the minimax theorem. By definition of h_S ,

$$\max_{\|w\| \leq 1} \langle w, z \rangle - h_S(w) = \max_{\|w\| \leq 1} \inf_{s \in S} \langle w, z - s \rangle.$$

Observe that $\{\|w\| \leq 1\}$ is a convex and compact set, S is convex by definition, and $\langle w, z - s \rangle$ is linear both in w and in s . We may thus apply Sion's minimax theorem to obtain that the last expression equals

$$\inf_{s \in S} \sup_{\|w\| \leq 1} \langle w, z - s \rangle = \inf_{s \in S} \|z - s\|_* = d_*(z, S),$$

where the definition of the dual norm was used in the last step, and (24) is obtained.

Proceeding to (25), we observe (24) implies that $d_*(\cdot, S)$ is the Legendre-Fenchel transform of an appropriately modified function \bar{h}_S , namely $d_*(z, S) = \max_{w \in \mathbb{R}^d} \{\langle w, z \rangle - \bar{h}_S(w)\}$ where $\bar{h}_S(w) = h_S(w)$ if $\|w\| \leq 1$ and $\bar{h}_S(w) = \infty$ for $\|w\| > 1$. Evidently h_S is a convex and lower semi-continuous function, which follows since both S and $\{\|w\| \leq 1\}$ are closed and convex sets. The equality in (25) now follows directly by Proposition 11.3 in Rockafellar and Wets (1997). ■

The algorithm: We can now repeat the blueprints of Subsection 3.2 to obtain an approachability algorithm for a convex target set S , which here relies on any norm $\|\cdot\|$. First, we apply an OCO algorithm to the functions $f_t(w) = -\langle w, \tau_t \rangle + h_S(w)$ over the convex compact set $\{w \in \mathbb{R}^d : \|w\| \leq 1\}$ to obtain, analogously to equation (11), a sequence of vectors (w_t) such that

$$\sum_{t=1}^T (\langle w_t, \tau_t \rangle - h_S(w_t)) \geq T \max_{\|w\| \leq 1} \{\langle w, \bar{\tau} \rangle - h_S(w)\} - a(T). \quad (26)$$

Next, each w_t is used as the direction vector for stage t , and the mixed action x_t is chosen so that $\langle w_t, \tau_t \rangle - h_S(w_t) \leq 0$ holds for any action of Nature. Observing (24), we obtain that

$$d_*(\bar{\tau}, S) \leq \frac{a(T)}{T} \rightarrow 0.$$

Follow the Leader: Consider the specific case that where FTL is used for the OCO algorithm. That is,

$$w_{t+1} \in \operatorname{argmin}_{\|w\| \leq 1} \sum_{k=1}^t f_k(w) \equiv \operatorname{argmax}_{\|w\| \leq 1} \{\langle w, \tau_t \rangle - h_S(w)\}. \quad (27)$$

By (25), this is equivalent to $w_{t+1} \in \partial d_*(\bar{\tau}, S)$. In particular, if $d_*(z, S)$ is differentiable at $z = \bar{\tau}$, then $w_{t+1} = \nabla d_*(\bar{\tau}, S)$. We therefore recover the approachability algorithm of Hart and Mas-Colell (2001) for the potential function $P(z) = d_*(z, S)$.

Convergence of the approachability algorithm of Hart and Mas-Colell (2001) requires the potential function $P(z)$ to be continuously differentiable. As observed there, for $P(z) = d_*(z, S)$ this holds if either the norm $\|\cdot\|_*$ is smooth (e.g., the q -norm for $1 < q < \infty$), or the boundary of S is smooth.

In our framework, convergence analysis of the FTL-based OCO algorithm can be carried out similarly to that of Section 4. In particular, similarly to the procedure of Subsection 4.2, if the norm $\|\cdot\|$ is smooth we can guarantee convergence of the OCO algorithm without affecting the induced approachability algorithm by adding an appropriate regularization term in (27), namely setting

$$w_{t+1} \in \operatorname{argmin}_{\|w\| \leq 1} \left\{ \sum_{k=1}^t f_k(w) - \frac{\rho_t}{2} \|w\|^2 \right\}.$$

By analogy to Lemma 13, the added regularization does not modify the direction of w_t but only its magnitude, hence the choice of actions x_t is the induced approachability algorithm remains the same. Convergence rates can be obtained along the lines of Section 4, and will not be considered in detail here.

Acknowledgments

The author wishes to thank Elad Hazan for helpful comments on a preliminary version of this work, and to the anonymous referees for many useful comments that helped improve the presentation. This research was supported by the Israel Science Foundation grant No. 1319/11.

Appendix A.

Proof of Proposition 4: We follow the outline of the proof of Lemma 2.10 in Shalev-Shwartz (2011), modified to accommodate a non-constant regularization sequence ρ_t . The starting point is the inequality, proved by induction,

$$\sum_{t=1}^T (f_t(w_t) - f_t(u)) \leq \sum_{t=1}^T (f_t(w_t) - f_t(w_{t+1})) + \rho_t R(u), \quad (28)$$

which holds for any $u \in W$. Therefore,

$$\sum_{t=1}^T (f_t(w_t) - f_t(u)) \leq L_T \sum_{t=1}^T \|w_t - w_{t+1}\|_2 + \rho_t R(u). \quad (29)$$

Denote $F_t(w) = \sum_{k=1}^{t-1} f_k(w) + \rho_{t-1} R(w)$. Then F_t is ρ_{t-1} -strongly convex, and w_t is its minimizer by definition. Hence, it holds generally that

$$F_t(w) \geq F_t(w_t) + \frac{\rho_{t-1}}{2} \|w - w_t\|_2^2,$$

and in particular,

$$F_t(w_{t+1}) \geq F_t(w_t) + \frac{\rho_{t-1}}{2} \|w_{t+1} - w_t\|_2^2, \quad (30)$$

$$F_{t+1}(w_t) \geq F_{t+1}(w_{t+1}) + \frac{\rho_t}{2} \|w_t - w_{t+1}\|_2^2. \quad (31)$$

Summing and cancelling terms, we obtain

$$f_t(w_t) - f_t(w_{t+1}) + (\rho_t - \rho_{t-1})(R(w_t) - R(w_{t+1})) \geq \frac{\rho_t + \rho_{t-1}}{2} \|w_{t+1} - w_t\|_2^2.$$

But the left-hand side is upper-bounded by $(L_f + (\rho_t - \rho_{t-1})L_R)\|w_{t+1} - w_t\|_2$, which implies that

$$\|w_{t+1} - w_t\|_2 \leq 2 \frac{L_f + (\rho_t - \rho_{t-1})L_R}{\rho_t + \rho_{t-1}}.$$

Substituting in (29) gives the bound stated in the Proposition. ■

References

- J. Abernethy, P. L. Bartlett, and E. Hazan. Blackwell approachability and low-regret learning are equivalent. In *Conference on Learning Theory (COLT)*, pages 27–46, June 2011.
- R.J. Aumann and M. Maschler. *Repeated Games with Incomplete Information*. MIT Press, Boston, MA, 1995.
- A. Bernstein and N. Shimkin. Response-based approachability with applications to generalized no-regret problems. *Journal of Machine Learning Research*, 16:747–773, 2015.
- A. Bernstein, S. Mannor, and N. Shimkin. Opportunistic approachability and generalized no-regret problems. *Mathematics of Operations Research*, 39(4):1057–1093, 2014.
- D. Blackwell. Controlled random walks. In *Proceedings of the International Congress of Mathematicians*, volume III, pages 335–338, 1954.
- D. Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 6:1–8, 1956.
- S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.
- N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, 2006.
- D. Fudenberg and D. K. Levine. *The Theory of Learning in Games*. MIT Press, Boston, MA, 1998.
- J. Hannan. Approximation to Bayes risk in repeated play. *Contributions to the Theory of Games*, 3:97–139, 1957.
- S. Hart and A. Mas-Colell. A general class of adaptive strategies. *Journal of Economic Theory*, 98:26–54, 2001.
- E. Hazan. The convex optimization approach to regret minimization. In S. Sra et al., editor, *Optimization for Machine Learning*, chapter 10. MIT Press, Cambridge, MA, 2012.
- E. Hazan. *Introduction to Online Convex Optimization*. Online book draft, <http://ocobook.cs.princeton.edu/>, April 2016.
- E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.
- S. Mannor, V. Perchet, and G. Stoltz. Approachability in unknown games: Online learning meets multi-objective optimization. In *Conference on Learning Theory (COLT)*, pages 339–355, Barcelona, Spain, May 2014.
- M. Maschler, E. Solan, and S. Zamir. *Game Theory*. Cambridge University Press, Cambridge, UK, 2013.
- V. Perchet. Calibration and internal no-regret with partial monitoring. In *International Conference on Algorithmic Learning Theory (ALT)*, Porto, Portugal, October 2009.
- V. Perchet. Approachability, regret and calibration: Implications and equivalences. *Journal of Dynamics and Games*, 1:181–254, 2014.
- V. Perchet and S. Mannor. Approachability, fast and slow. In *Proc. COLT 2013: JMLR Workshop and Conference Proceedings*, volume 30, pages 474–488, 2013.
- H. Peyton Young. *Strategic Learning and Its Limits*. Oxford University Press, 2004.
- R. T. Rockafellar and R. Wets. *Variational Analysis*. Springer-Verlag, 1997.
- S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4:107–194, 2011.
- N. Shimkin and A. Shwartz. Guaranteed performance regions in Markovian systems with competing decision makers. *IEEE Transactions on Automatic Control*, 38(1): 84–95, 1993.

OCO-BASED APPROACHABILITY

M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning (ICML)*, pages 928–936, 2003.

A Well-Conditioned and Sparse Estimation of Covariance and Inverse Covariance Matrices Using a Joint Penalty

Ashwini Maurya

Department of Statistics and Probability
Michigan State University
East Lansing, MI 48824, USA

AKMAURYA07@GMAIL.COM

Editor: Jie Peng

Abstract

We develop a method for estimating well-conditioned and sparse covariance and inverse covariance matrices from a sample of vectors drawn from a sub-Gaussian distribution in high dimensional setting. The proposed estimators are obtained by minimizing the quadratic loss function and joint penalty of l_1 norm and variance of its eigenvalues. In contrast to some of the existing methods of covariance and inverse covariance matrix estimation, where often the interest is to estimate a sparse matrix, the proposed method is flexible in estimating both a sparse and well-conditioned covariance matrix simultaneously. The proposed estimators are optimal in the sense that they achieve the mini-max rate of estimation in operator norm for the underlying class of covariance and inverse covariance matrices. We give a very fast algorithm for computation of these covariance and inverse covariance matrices which is easily scalable to large scale data analysis problems. The simulation study for varying sample sizes and variables shows that the proposed estimators performs better than several other estimators for various choices of structured covariance and inverse covariance matrices. We also use our proposed estimator for tumor tissues classification using gene expression data and compare its performance with some other classification methods.

Keywords: Sparsity, Eigenvalue Penalty, Penalized Estimation

1. Introduction

With the recent surge in data technology and storage capacity, today's statisticians often encounter data sets where sample size n is small and number of variables p is very large: often hundreds, thousands and even million or more. Examples include gene expression data and web search problems [Clarke et al. (2008), Pass et al. (2006)]. For many of the high dimensional data problems, the choice of classical statistical methods becomes inappropriate for making valid inference. The recent developments in asymptotic theory deal with increasing p as long as both p and n tend to infinity at some rate depending upon the parameters of interest.

The estimation of covariance and inverse covariance matrix is a problem of primary interest in multivariate statistical analysis. Some of the applications include: (i) Principal component analysis (PCA) [Johnstone and Lu (2004), Zou et al. (2006)]; where the goal is to project the data on "best" k -dimensional subspace, and where best means the projected data explains as much of the variation in original data without increasing k . (ii) Discriminant analysis [Mardia et al. (1979)]; where the goal is to classify observations into different

classes. Here estimates of covariance and inverse covariance matrices play an important role as the classifier is often a function of these entities. (iii) Regression analysis: If interest focuses on estimation of regression coefficients with correlated (or longitudinal) data, a sandwich estimator of the covariance matrix may be used to provide standard errors for the estimated coefficients that are robust in the sense that they remain consistent under misspecification of the covariance structure. (iv) Gaussian graphical modeling [Meinshausen and Bühlmann (2006), Wainwright et al. (2006), Yuan and Lin (2007), Yuan (2009)]; the relationship structure among nodes can be inferred from inverse covariance matrix. A zero entry in the inverse covariance matrix implies conditional independence between the corresponding nodes.

The estimation of large dimensional covariance matrix based on few sample observations is a difficult problem, especially when $n \asymp p$ (here $a_n \asymp b_n$ means that there exist positive constants c and C such that $c \leq a_n/b_n \leq C$). In these situations, the sample covariance matrix becomes unstable which explodes the estimation error. It is well known that the eigenvalues of sample covariance matrix are over-dispersed which means that the eigen-spectrum of sample covariance matrix is not a good estimator of its population counterpart [Marcenko and Pastur (1967), Karoui (2008a)]. To illustrate this point, consider $\Sigma_p = I_p$, so all the eigenvalues are 1. A result from [Geman (1980)] shows that if entries of X_i 's are i.i.d (let X_i 's have mean zero and variance 1) with a finite fourth moment and if $p/n \rightarrow \theta < 1$, then the largest sample eigenvalue l_1 satisfies:

$$l_1 \rightarrow (1 + \sqrt{\theta})^2, \quad a.s.$$

This suggests that l_1 is not a consistent estimator of the largest eigenvalue σ_1 of population covariance matrix. In particular if $n = p$ then l_1 tends to 4 whereas $\sigma_1 = 1$. This is also evident in the eigenvalue plot in Figure 2.1. The distribution of l_1 also depends on the underlying structure of the true covariance matrix. From Figure 2.1, it is evident that the smaller sample eigenvalues tend to underestimate the true eigenvalues for large p and small n . For more discussion on this topic, see Karoui (2008a).

To correct for this bias, a natural choice would be to shrink the sample eigenvalues towards some suitable constant to reduce the over-dispersion. For instance, Stein (1975) proposed an estimator of the form $\tilde{\Sigma} = \tilde{U}\Lambda(\lambda)\tilde{U}$, where $\Lambda(\lambda)$ is a diagonal matrix with diagonal entries as transformed function of the sample eigenvalues and \tilde{U} is the matrix of the eigen-vectors. In another interesting paper Ledoit and Wolf (2004) proposed an estimator that shrinks the sample covariance matrix towards the identity matrix. In another paper, Karoui (2008b) proposed a non-parametric estimation of spectrum of eigenvalues and show that his estimator is consistent in the sense of weak convergence of distributions.

The covariance matrix estimates based on eigen-spectrum shrinkage are well-conditioned in the sense that their eigenvalues are well bounded away from zero. These estimates are based on the shrinkage of the eigenvalues and therefore invariant under some orthogonal group i.e. the shrinkage estimators shrink the eigenvalues but eigenvectors remain unchanged. In other words, the basis (eigenvector) in which the data are given is not taken advantage of and therefore the methods rely on premise that one will be able to find a good estimate in any basis. In particular, it is reasonable to believe that the basis generating the data is somewhat nice. Often this translates into the assumption that the covariance matrix

has particular structure that one should be able to take advantage of. In these situations, it becomes natural to perform certain form of regularization directly on the entries of the sample covariance matrix.

Much of the recent literature focuses on two broad classes of regularized covariance matrix estimation. i) The one class relies on natural ordering among variables, where one often assumes that the variables far apart are weakly correlated and ii) the other class where there is no assumption on the natural ordering among variables. The first class includes the estimators based on banding and tapering [Bickel and Levina (2008b), Cai et al. (2011)]. These estimators are appropriate for a number of applications for ordered data (time series, spectroscopy, climate data). However for many applications including gene expression data, prior knowledge of any canonical ordering is not available and searching for all permutation of possible ordering would not be feasible. In these situations, an ℓ_1 penalized estimator becomes more appropriate which yields a permutation-invariant estimate.

To obtain a suitable estimate which is both well-conditioned and sparse, we introduce two regularization terms: i) ℓ_1 penalty for each of the off-diagonal elements of matrix and, ii) penalty proportional to the variance of the eigenvalues. The ℓ_1 minimization problems are well studied in the covariance and inverse covariance matrix estimation literature [Friedman et al. (2008), Banerjee et al. (2008), Ravikumar et al. (2011), Bein and Tibshirani (2011), Maurya (2014) etc.]. Rothman (2012) proposes an ℓ_1 penalized log-likelihood estimator and shows that estimator is consistent in Frobenius norm at the rate of $O_p\left(\sqrt{(\bar{p} + s) \log \bar{p}/n}\right)$, as both p and n approach to infinity. Here s is the number of non-zero off-diagonal elements in the true covariance matrix. In another interesting paper Bein and Tibshirani (2011) propose an estimator of covariance matrix as penalized maximum likelihood estimator with a weighted lasso type penalty. In these optimization problems, the ℓ_1 penalty results in sparse and a permutation-invariant estimator as compared to other $\ell_p, q \neq 1$ penalties. Another advantage is that the ℓ_1 norm is a convex function which makes it suitable for large scale optimization problems. A number of fast algorithms exist in the literature for covariance and inverse covariance matrix estimation [Friedman et al. (2008), Rothman et al. (2008)]. The eigenvalues variance penalty overcomes the over-dispersion in the sample covariance matrix so that the estimator remains well-conditioned.

Ladroit and Wolf (2004) proposed an estimator of covariance matrix as a linear combination of sample covariance and identity matrix. Their estimator of covariance matrix is well-conditioned but it is not sparse. Rothman et al. (2008) proposed estimator of covariance matrix based on quadratic loss function and ℓ_1 penalty with a log-barrier on the determinant of covariance matrix. The log-determinant barrier is a valid technique to achieve positive definiteness but it is still unclear whether the iterative procedure proposed in Rothman et al. (2008) actually finds the right solution to the corresponding optimization problem. In another interesting paper, Xie et al. (2012) proposed an estimator of covariance matrix as a minimizer of penalized quadratic loss function over set of positive definite matrices. In their paper, the authors solve a positive definite constrained optimization problem and establish the consistency of estimator. The resulting estimator is sparse and positive definite but whether it overcomes the over-dispersion of the eigen-spectrum of sample covariance matrix, is hard to justify. Maurya (2014) proposed a joint convex penalty as function of ℓ_1 and trace norm (defined as sum of singular values of a matrix) for inverse covariance matrix estimation based on penalized likelihood approach.

In this paper, we propose the JPEN (Joint PENalty) estimators for covariance and inverse covariance matrices estimation and derive an explicit rate of convergence in both the operator and Frobenius norm. The JPEN estimators achieves mini-max rate of convergence under operator norm for the underlying class of sparse covariance and inverse covariance matrices and hence is optimal. For more details see section §3. One of the major advantage of the proposed estimators is that the proposed algorithm is very fast, efficient and easily scalable to a large scale data analysis problem.

The rest of the paper is organized as following. The next section highlights some background and problem set-up for covariance and inverse covariance matrix estimation. In section 3, we describe the proposed estimators and establish their theoretical consistency. In section 4, we give an algorithm and compare its computational time with some other existing algorithms. Section 5 highlights the performance of the proposed estimators on simulated data while an application of proposed estimator to real life data is given in section 6.

Notation: For a matrix M , let $\|M\|_1$ denote its ℓ_1 norm defined as the sum of absolute values of the entries of M , $\|M\|_F$ denote its Frobenius norm, defined as the sum of square of elements of M , $\|M\|$ denote its operator norm (also called spectral norm), defined as the largest absolute eigenvalue of M , M^- denotes matrix M where all diagonal elements are set to zero, M^+ denote matrix M where all off-diagonal elements are set to zero, $\sigma_i(M)$ denote the i^{th} largest eigenvalue of M , $\text{tr}(M)$ denotes its trace, $\det(M)$ denote its determinant, $\sigma_{\min}(M)$ and $\sigma_{\max}(M)$ denote the minimum and maximum eigenvalues of M , $|M|$ be its cardinality, and let $\text{sign}(M)$ be matrix of signs of elements of M . For any real x , let $\text{sign}(x)$ denotes sign of x , and let $|x|$ denotes its absolute value.

2. Background and Problem Set-up

Let $X = (X_1, X_2, \dots, X_p)$ be a zero-mean p -dimensional random vector. The focus of this paper is the estimation of the covariance matrix $\Sigma := \mathbb{E}(XX^T)$ and its inverse Σ^{-1} from a sample of independently and identically distributed data $\{X^{(k)}\}_{k=1}^n$. In this section we provide some background and problem setup more precisely.

The choice of loss function is very crucial in any optimization problem. An optimal estimator for a particular loss function may not be optimal for another choice of loss function. Recent literature in covariance matrix and inverse covariance matrix estimation mostly focuses on estimation based on likelihood function or quadratic loss function [Friedman et al. (2008), Banerjee et al. (2008), Bickel and Levina (2008b), Ravikumar et al. (2011), Rothman et al. (2008), Maurya (2014)]. The maximum likelihood estimation requires a tractable probability distribution of observations whereas quadratic loss function does not have any such requirement and therefore fully non-parametric. The quadratic loss function is convex and due to this analytical tractability, it is a widely applicable choice for many data analysis problems.

2.1 Proposed Estimators

Let S be the sample covariance matrix. Consider the following optimization problem.

$$\hat{\Sigma}_{\lambda,\gamma} = \underset{\Sigma = \Sigma^T, \text{tr}(\Sigma) = \text{tr}(S)}{\text{argmin}} \left[\|\Sigma - S\|_F^2 + \lambda \|\Sigma^{-1}\|_1 + \gamma \sum_{i=1}^p \{\sigma_i(\Sigma) - \sigma_i(S)\}^2 \right], \quad (2.1)$$

where σ_Σ is the mean of eigenvalues of Σ , λ and γ are some positive constants. Note that by penalty function $\|\Sigma^{-1}\|_1$, we only penalize off-diagonal elements of Σ . The eigenvalues of variance penalty term for eigen-spectrum shrinkage is chosen from the following points of interest: i) It is easy to interpret and ii) this choice of penalty function yields a very fast optimization algorithm. By constraint $\text{tr}(\Sigma) = \text{tr}(S)$, the total variation in $\hat{\Sigma}_{\lambda,\gamma}$ is same as that in sample covariance matrix S , however the eigenvalues of $\hat{\Sigma}_{\lambda,\gamma}$ are well-conditioned than those of S . From here onwards we suppress the dependence of λ, γ on Σ and denote $\hat{\Sigma}_{\lambda,\gamma}$ by $\hat{\Sigma}$.

For $\gamma = 0$, the solution to (2.1) is the standard soft-thresholding estimator for quadratic loss function and its solution is given by (see §4 for derivation of this estimator):

$$\begin{aligned} \hat{\Sigma}_{ii} &= s_{ii} \\ \hat{\Sigma}_{ij} &= \text{sign}(s_{ij}) \max \left(|s_{ij}| - \frac{\lambda}{2}, 0 \right), \quad i \neq j. \end{aligned} \quad (2.2)$$

It is clear from this expression that a sufficiently large value of λ will result in sparse covariance matrix estimate. But estimator $\hat{\Sigma}$ of (2.2) is not necessarily positive definite [for more details here see [Xue et al. \(2012\)](#)]. Moreover it is hard to say whether it overcomes the over-dispersion in the sample eigenvalues. The following eigenvalue plot (Figure (2.1)) illustrates this phenomenon for a neighbourhood type (see §5 for details on description of neighbourhood type of covariance matrix) covariance matrix. Here we simulated random vectors from multivariate normal distribution with sample size $n = 50$ and number of covariates $p = 20$. As is evident from Figure 2.1, eigenvalues of sample covariance matrix are over-dispersed as most of them are either too large or close to zero. Eigenvalues of the proposed Joint Penalty (JPEN) estimator and PDSCE (Positive Definite Sparse Covariance Estimator ([Rothman \(2012\)](#))) of the covariance matrix are well aligned with those of true covariance matrix. See §5 for detailed discussion. Another drawback of the estimator (2.2) is that the estimate can be negative definite.

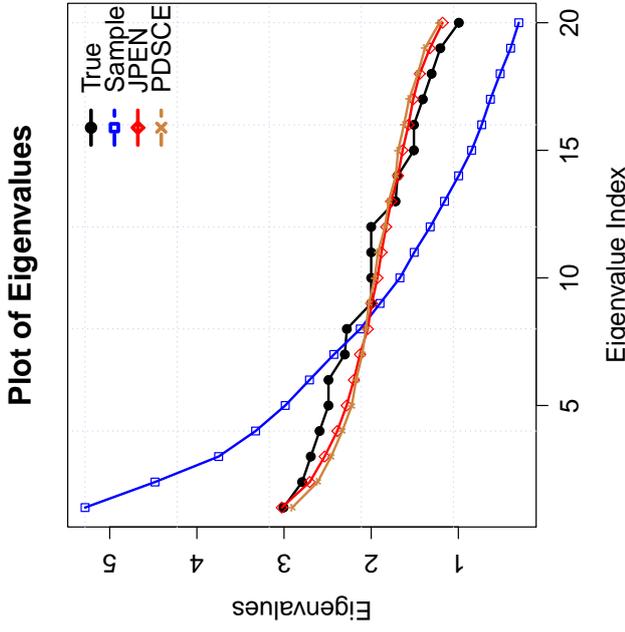
As argued earlier, to overcome the over-dispersion in eigen-spectrum of sample covariance matrix, we include eigenvalues variance penalty. To illustrate its advantage, consider $\lambda = 0$. After some algebra, let $\hat{\Sigma}$ be the minimizer of (2.1), then it is given by:

$$\hat{\Sigma} = (S + \gamma t I)/(1 + \gamma), \quad (2.3)$$

where I is the identity matrix, and $t = \sum_{i=1}^p S_{ii}/p$. After some algebra, conclude that for any $\gamma > 0$:

$$\begin{aligned} \sigma_{\min}(\hat{\Sigma}) &= \sigma_{\min}(S + \gamma t I)/(1 + \gamma) \\ &\geq \frac{\gamma t}{1 + \gamma} > 0 \end{aligned}$$

Figure 2.1: Comparison of Eigenvalues of Covariance Matrices



This means that the eigenvalues variance penalty improves S to a positive definite estimator $\hat{\Sigma}$. However the estimator (2.3) is well-conditioned but need not be sparse. Sparsity can be achieved by imposing ℓ_1 penalty on the entries of covariance matrix. Simulations have shown that, in general the minimizer of (2.1) is not positive definite for all values of $\lambda > 0$ and $\gamma > 0$. Here onwards we focus on correlation matrix estimation, and later generalize the method for covariance matrix estimation.

To achieve both well-conditioned and sparse positive definite estimator we optimize the following objective function in R over specific region of values of (λ, γ) which depends upon sample correlation matrix K and λ, γ . Here the condition $\text{tr}(\Sigma) = \text{tr}(S)$ reduces to $\text{tr}(R) = p$, and $t = 1$. Consider the following optimization problem:

$$\hat{R}_K = \underset{R = R^T, \text{tr}(R) = p(\lambda, \gamma) \in \delta_K^+}{\text{argmin}} \left[\|R - K\|_F^2 + \lambda \|R^{-1}\|_1 + \gamma \sum_{i=1}^p \{\sigma_i(R) - \sigma_i(S)\}^2 \right], \quad (2.4)$$

where

$$\hat{\Sigma}_K^K = \left\{ (\lambda, \gamma) : \lambda, \gamma > 0, \lambda \asymp \gamma \asymp \sqrt{\frac{\log p}{n}}, \forall \epsilon > 0, \sigma_{\min}\{K + \gamma I\} - \frac{\lambda}{2} * \text{sign}(K + \gamma I) > \epsilon \right\},$$

and $\bar{\sigma}_R$ is mean of the eigenvalues of R . For instance when K is diagonal matrix, the set $\hat{\Sigma}_K^K$ is given by:

$$\hat{\Sigma}_K^K = \left\{ (\lambda, \gamma) : \lambda, \gamma > 0, \lambda \asymp \gamma \asymp \sqrt{\frac{\log p}{n}}, \forall \epsilon > 0, \lambda < 2(\gamma - \epsilon) \right\}.$$

The minimization in (2.4) over R is for fixed $(\lambda, \gamma) \in \hat{\Sigma}_K^K$. The proposed estimator of covariance matrix (based on regularized correlation matrix estimator \hat{R}_K) is given by $\hat{\Sigma}_K = (S^+)^{1/2} \hat{R}_K (S^+)^{1/2}$, where S^+ is the diagonal matrix of the diagonal elements of S . Furthermore Lemmas 3.1 and 3.2, respectively show that the objective function (2.4) is convex and estimator given in (2.4) is positive definite.

2.2 Our Contribution

The main contributions are the following:

- i) The proposed estimators are both sparse and well-conditioned simultaneously. This approach allows to take advantage of a prior structure if known on the eigenvalues of the true covariance and the inverse covariance matrices.
- ii) We establish theoretical consistency of proposed estimators in both operator and Frobenius norm. The proposed JPEN estimators achieves the mini-max rate of convergence in operator norm for the underlying class of sparse and well-conditioned covariance and inverse covariance matrices and therefore is optimal.
- iii) The proposed algorithm is very fast, efficient and easily scalable to large scale optimization problems.

3. Analysis of JPEN Method

Def: A random vector X is said to have sub-Gaussian distribution if for each $t \geq 0$ and $y \in \mathbb{R}^p$ with $\|y\|_2 = 1$, there exist $0 < \tau < \infty$ such that

$$\mathbb{P}\{|y^T(X - \mathbb{E}(X))| > t\} \leq e^{-t^2/2\tau} \tag{3.1}$$

Although the JPEN estimators exists for any finite $2 \leq n < p < \infty$, for theoretical consistency in operator norm we require $s \log p = o(n)$ and for Frobenius norm we require $(p + s) \log p = o(n)$ where s is the upper bound on the number of non-zero off-diagonal entries in true covariance matrix. For more details, see the remark after Theorem 3.1.

3.1 Covariance Matrix Estimation

We make the following assumptions about the true covariance matrix Σ_0 .

A0. Let $X := (X_1, X_2, \dots, X_p)$ be a mean zero vector with covariance matrix Σ_0 such that each $X_i/\sqrt{\Sigma_{0ii}}$ has sub-Gaussian distribution with parameter τ as defined in (3.1).

A1. With $E = \{(i, j) : \Sigma_{0ij} \neq 0, i \neq j\}$, the $|E| \leq s$ for some positive integer s .

A2. There exists a finite positive real number $\bar{k} > 0$ such that $1/\bar{k} \leq \sigma_{\min}(\Sigma_0) \leq \sigma_{\max}(\Sigma_0) \leq k$.

Assumption A2 guarantees that the true covariance matrix Σ_0 is well-conditioned (i.e. all the eigenvalues are finite and positive). A well-conditioned means that [Ledoit and Wolf \(2004\)](#)) inverting the matrix does not explode the estimation error. Assumption A1 is more of a definition which says that the number of non-zero off diagonal elements are bounded by some positive integer. Theorem 3.1 gives the rate of convergence of the proposed correlation based covariance matrix estimator (2.4). The following Lemmas show that optimization problem in (2.4) is convex and the proposed JPEN estimator (2.4) is positive definite.

Lemma 1. The optimization problem in (2.4) is convex.

Lemma 2. The estimator given by (2.4) is positive definite for any $2 \leq n < \infty$ and $p < \infty$.

Theorem 3.1. Let $(\lambda, \gamma) \in \hat{\Sigma}_K^K$ and $\hat{\Sigma}_K$ be as defined in (2.4). Under Assumptions A0, A1, A2,

$$\|\hat{R}_K - R_0\|_F = O_P\left(\sqrt{\frac{s \log p}{n}}\right) \quad \text{and} \quad \|\hat{\Sigma}_K - \Sigma_0\| = O_P\left(\sqrt{\frac{(s+1)\log p}{n}}\right), \tag{3.2}$$

where R_0 is true correlation matrix.

Remark 1. The JPEN estimator $\hat{\Sigma}_K$ is mini-max optimal under the operator norm. In [Cai et al. \(2015\)](#), the authors obtain the mini-max rate of convergence in the operator norm of their covariance matrix estimator for the particular construction of parameter space $\mathcal{H}_0(c_{n,p}) := \left\{ \Sigma : \max_{1 \leq i \leq p} \sum_{j=1}^p I(\sigma_{ij} \neq 0) \leq c_{n,p} \right\}$. They show that this rate in operator norm is $c_{n,p} \sqrt{\log p/n}$ which is same as that of $\hat{\Sigma}_K$ for $1 \leq c_{n,p} = \sqrt{s}$.

2. Bickel and Levina (2008a) proved that under the assumption of $\sum_{j=1}^p |\sigma_{ij}|^q \leq c_0(p)$ for some $0 \leq q \leq 1$, the hard thresholding estimator of the sample covariance matrix for tuning parameter $\lambda \asymp \sqrt{(\log p)/n}$ is consistent in operator norm at a rate no worse than $O_P\left(c_0(p) \sqrt{p} (\frac{\log p}{n})^{(1-q)/2}\right)$ where $c_0(p)$ is the upper bound on the number of non-zero elements in each row. Here the truly sparse case corresponds to $q = 0$. The rate of convergence of $\hat{\Sigma}_K$ is same as that of [Bickel and Levina \(2008a\)](#) except in the following cases:

Case (i) The covariance matrix has all off diagonal elements zero except last row which has \sqrt{p} non-zero elements. Then $c_0(p) = \sqrt{p}$ and $\sqrt{s} = \sqrt{2} \sqrt{p} - 1$. The operator norm rate of convergence for JPEN estimator is $O_P\left(\sqrt{\sqrt{p}} (\log p)/n\right)$ where as rate of [Bickel and Levina's](#) estimator is $O_P\left(\sqrt{p} (\log p)/n\right)$.

Case (ii) When the true covariance matrix is tri-diagonal, we have $c_0(p) = 2$ and $s = 2p - 2$, the JPEN estimator has rate of $\sqrt{p} \log p/n$ whereas the [Bickel and Levina's](#) estimator has rate of $\sqrt{\log p/n}$.

For the case $\sqrt{s} \asymp c_0(p)$ and JPEN has the same rate of convergence as that of [Bickel and Levina's](#) estimator.

3. The operator norm rate of convergence is much faster than Frobenius norm. This is due to the fact that Frobenius norm convergence is in terms of all eigenvalues of the covariance matrix whereas the operator norm gives the convergence of the estimators in terms of the largest eigenvalue.

4. Our proposed estimator is applicable to estimate any non-negative definite covariance matrix.

Note that the estimator $\hat{\Sigma}_K$ is obtained by regularization of sample correlation matrix in (2.4). In some application it is desirable to directly regularize the sample covariance matrix. The JPEN estimator of the covariance matrix based on regularization of sample covariance matrix is obtained by solving the following optimization problem:

$$\hat{\Sigma}_S = \arg \min_{\Sigma = \Sigma^T, \text{tr}(\Sigma) = \text{tr}(S)(\lambda, \gamma) \in \hat{S}_S^K} \left[\|\Sigma - S\|_F^2 + \lambda \|\Sigma\|_1 + \gamma \sum_{i=1}^p \{\sigma_i(\Sigma) - \bar{\sigma}_S\}^2 \right], \quad (3.3)$$

where

$$\hat{S}_S^K = \left\{ (\lambda, \gamma) : \lambda, \gamma > 0, \lambda \asymp \gamma \asymp \sqrt{\frac{\log p}{n}}, \forall \epsilon > 0, \sigma_{\min} \left\{ (S + \gamma t I) - \frac{\lambda}{2} * \text{sign}(S + \gamma t I) \right\} > \epsilon \right\},$$

and S is sample covariance matrix. The minimization in (3.3) over Σ is for fixed $(\lambda, \gamma) \in \hat{S}_S^K$. The estimator $\hat{\Sigma}_S$ is positive definite and well-conditioned. Theorem 3.2 gives the rate of convergence of the estimator $\hat{\Sigma}_S$ in Frobenius norm.

Theorem 3.2. *Let $(\lambda, \gamma) \in \hat{S}_S^K$, and let $\hat{\Sigma}_S$ be as defined in (3.3). Under Assumptions A0, A1, A2,*

$$\|\hat{\Sigma}_S - \Sigma_0\|_F = O_P \left(\sqrt{\frac{(s+p)\log p}{n}} \right) \quad (3.4)$$

As noted in Rothman (2012) the worst part of convergence here comes from estimating the diagonal entries.

3.1.1 WEIGHTED JPEN ESTIMATOR FOR THE COVARIANCE MATRIX ESTIMATION

A modification of estimator \hat{R}_K is obtained by adding positive weights to the term $(\sigma_i(R) - \bar{\sigma}_R)^2$. This leads to weighted eigenvalues variance penalty with larger weights amounting to greater shrinkage towards the center and vice versa. Note that the choice of the weights allows one to use any prior structure of the eigenvalues (if known) in estimating the covariance matrix. The weighted JPEN correlation matrix estimator \hat{R}_A is given by :

$$\hat{R}_A = \arg \min_{R = R^T, \text{tr}(R) = \text{tr}(S)(\lambda, \gamma) \in \hat{S}_A^{K,A}} \left[\|R - K\|_F^2 + \lambda \|R\|_1 + \gamma \sum_{i=1}^p a_i \{\sigma_i(R) - \bar{\sigma}_R\}^2 \right], \quad (3.5)$$

where

$$\hat{S}_A^{K,A} = \left\{ (\lambda, \gamma) : \lambda \asymp \gamma \asymp \sqrt{\frac{\log p}{n}}, \lambda \leq \frac{(2 \sigma_{\min}(K)(1+\gamma \max(A_{ii})^{-1}) + \gamma \min(A_{ii}))}{(1+\gamma \min(A_{ii}))^{-p}}}, \right\},$$

and $A = \text{diag}(A_{11}, A_{22}, \dots, A_{pp})$ with $A_{ii} = a_i$. The proposed covariance matrix estimator is $\hat{\Sigma}_{K,A} = (S^+)^{1/2} \hat{R}_A (S^+)^{1/2}$. The optimization problem in (3.5) is convex and yields a positive definite estimator for each $(\lambda, \gamma) \in \hat{S}_A^{K,A}$. A simple exercise shows that the estimator $\hat{\Sigma}_{K,A}$ has same rate of convergence as that of $\hat{\Sigma}_S$.

3.2 Estimation of Inverse Covariance Matrix

We extend the JPEN approach to estimate a well-conditioned and sparse inverse covariance matrix. Similar to the covariance matrix estimation, we first propose an estimator for inverse covariance matrix based on regularized inverse correlation matrix and discuss its rate of convergence in Frobenius and operator norm.

Notation: We shall use Z and Ω for inverse correlation and inverse covariance matrix respectively.

Assumptions: We make the following assumptions about the true inverse covariance matrix Ω_0 . Let $\Sigma_0 = \Omega_0^{-1}$.

B0. Same as the assumption A0.

B1. With $H = \{(i, j) : \Omega_{0ij} \neq 0, i \neq j\}$, the $|H| \leq s$, for some positive integer s .

B2. There exist $0 < \bar{k} < \infty$ large enough such that $(1/\bar{k}) \leq \sigma_{\min}(\Omega_0) \leq \sigma_{\max}(\Omega_0) \leq \bar{k}$.

Let \hat{R}_K be a JPEN estimator for the true correlation matrix. By Lemma 3.2, \hat{R}_K is positive definite. Define the JPEN estimator of inverse correlation matrix as the solution to the following optimization problem,

$$\hat{Z}_K = \arg \min_{Z = Z^T, \text{tr}(Z) = \text{tr}(\hat{R}_K^{-1})(\lambda, \gamma) \in \hat{S}_Z^K} \left[\|Z - \hat{R}_K^{-1}\|^2 + \lambda \|Z\|_1 + \gamma \sum_{i=1}^p \{\sigma_i(Z) - \bar{\sigma}(Z)\}^2 \right] \quad (3.6)$$

where

$$\hat{S}_Z^K = \left\{ (\lambda, \gamma) : \lambda, \gamma > 0, \lambda \asymp \gamma \asymp \sqrt{\frac{\log p}{n}}, \forall \epsilon > 0, \right. \\ \left. \sigma_{\min} \left\{ (\hat{R}_K^{-1} + \gamma t_1 I) - \frac{\lambda}{2} * \text{sign}(\hat{R}_K^{-1} + \gamma t_1 I) \right\} > \epsilon \right\},$$

and t_1 is average of the diagonal elements of \hat{R}_K^{-1} . The minimization in (3.6) over Z is for fixed $(\lambda, \gamma) \in \hat{S}_Z^K$. The proposed JPEN estimator of inverse covariance matrix (based on regularized inverse correlation matrix estimator \hat{Z}_K) is given by $\hat{\Omega}_K = (S^+)^{-1/2} \hat{Z}_K (S^+)^{-1/2}$, where S^+ is a diagonal matrix of the diagonal elements of S . Moreover (3.6) is a convex optimization problem and \hat{Z}_K is positive definite.

Next we state the consistency of estimators \hat{Z}_K and $\hat{\Omega}_K$.

Theorem 3.3. *Under Assumptions B0, B1, B2 and for $(\lambda, \gamma) \in \hat{S}_Z^K$,*

$$\|\hat{Z}_K - R_0^{-1}\|_F = O_P \left(\sqrt{\frac{s \log p}{n}} \right) \quad \text{and} \quad \|\hat{\Omega}_K - \Omega_0\| = O_P \left(\sqrt{\frac{(s+1) \log p}{n}} \right) \quad (3.7)$$

where R_0^{-1} is the inverse of true correlation matrix.

Remark-1. Note that the JPEN estimator $\hat{\Omega}_K$ achieves mini-max rate of convergence for the class of covariance matrices satisfying assumption B_0 , B_1 , and B_2 and therefore optimal. The similar rate is obtained in [Cai et al. \(2015\)](#) for their class of sparse inverse covariance matrices.

Next we give another estimate of inverse covariance matrix based on $\hat{\Sigma}_S$. Consider the following optimization problem:

$$\hat{\Omega}_S = \underset{\Omega \in \Omega^T, \text{tr}(\Omega) = \text{tr}(\hat{\Sigma}_S^{-1}), (\lambda, \gamma) \in \hat{\mathcal{E}}_S^2}{\text{arg min}} \left[\|\Omega - \hat{\Sigma}_S^{-1}\|_F^2 + \lambda \|\Omega^{-1}\|_1 + \gamma \sum_{i=1}^p \{\sigma_i(\Omega) - \sigma_i(R)\}^2 \right], \quad (3.8)$$

where

$$\hat{\mathcal{E}}_S^2 = \left\{ (\lambda, \gamma) : \lambda, \gamma > 0, \lambda \asymp \gamma \asymp \sqrt{\frac{\log p}{n}}, \forall \epsilon > 0, \right. \\ \left. \sigma_{\min}\{(\hat{\Sigma}_S^{-1} + \gamma t_2 I) - \frac{\lambda}{2} * \text{sign}(\hat{\Sigma}_S^{-1} + \gamma t_2 I)\} > \epsilon \right\},$$

and t_2 is average of the diagonal elements of $\hat{\Sigma}_S$. The minimization in (3.8) over Ω is for fixed $(\lambda, \gamma) \in \hat{\mathcal{E}}_S^2$. The estimator in (3.8) is positive definite and well-conditioned. The consistency result of the estimator $\hat{\Omega}_S$ is given in following theorem.

Theorem 3.4. Let $(\lambda, \gamma) \in \hat{\mathcal{E}}_S^2$ and let $\hat{\Omega}_S$ be as defined in (3.8). Under Assumptions B_0 , B_1 , and B_2 ,

$$\|\hat{\Omega}_S - \Omega_0\|_F = O_P \left(\sqrt{\frac{(s+p)\log p}{n}} \right). \quad (3.9)$$

3.2.1 WEIGHTED JPEN ESTIMATOR FOR THE INVERSE COVARIANCE MATRIX

Similar to weighted JPEN covariance matrix estimator $\hat{\Sigma}_{K,A}$, a weighted JPEN estimator of the inverse covariance matrix is obtained by adding positive weights a_i to the term $(\sigma_i(Z) - 1)^2$ in (3.8). The weighted JPEN estimator is $\hat{\Omega}_{K,A} := (S^+)^{-1/2} \hat{Z}_A (S^+)^{-1/2}$, where

$$\hat{Z}_A = \underset{Z = Z^T, \text{tr}(Z) = \text{tr}(R_K^{-1}), (\lambda, \gamma) \in \hat{\mathcal{E}}_{K,A}^2}{\text{arg min}} \left[\|Z - \hat{R}_K^{-1}\|_F^2 + \lambda \|Z^{-1}\|_1 + \gamma \sum_{i=1}^p a_i \{\sigma_i(Z) - 1\}^2 \right], \quad (3.10)$$

with

$$\hat{\mathcal{E}}_{K,A}^2 = \left\{ (\lambda, \gamma) : \lambda \asymp \gamma \asymp \sqrt{\frac{\log p}{n}}, \lambda \leq \frac{(2 \sigma_{\min}(R_K^{-1}) (1 + \gamma \text{max}(A_{ii}))^{-1})}{(1 + \gamma \text{min}(A_{ii}))^{-\gamma}} + \frac{2 \text{min}(A_{ii})}{p} \right\},$$

and $A = \text{diag}(A_{11}, A_{22}, \dots, A_{pp})$ with $A_{ii} = a_i$. The optimization problem in (3.10) is convex and yields a positive definite estimator for $(\lambda, \gamma) \in \hat{\mathcal{E}}_{K,A}^2$. A simple exercise shows that the estimator \hat{Z}_A has similar rate of convergence as that of \hat{Z}_K .

4. An Algorithm

4.1 Covariance Matrix Estimation:

The optimization problem (2.4) can be written as:

$$\hat{R}_K = \underset{R = R^T, (\lambda, \gamma) \in \hat{\mathcal{E}}_K^2}{\text{arg min}} f(R), \quad (4.1)$$

where

$$f(R) = \|R - K\|_F^2 + \lambda \|R^{-1}\|_1 + \gamma \sum_{i=1}^p \{\sigma_i(R) - \sigma_i(R)\}^2.$$

Note that $\sum_{i=1}^p \{\sigma_i(R) - \sigma_i(R)\}^2 = \text{tr}(R^2) - 2 \text{tr}(R) + p$, where we have used the constraint $\text{tr}(R) = p$. Therefore,

$$f(R) = \|R - K\|_F^2 + \lambda \|R^{-1}\|_1 + \gamma \text{tr}(R^2) - 2 \gamma \text{tr}(R) + p \\ = \text{tr}(R^2(1 + \gamma)) - 2\text{tr}\{R(K + \gamma I)\} + \text{tr}(K^T K) + \lambda \|R^{-1}\|_1 + p \\ = (1 + \gamma) \{\text{tr}(R^2) - 2/(1 + \gamma) \text{tr}\{R(K + \gamma I)\} + (1/(1 + \gamma)) \text{tr}(K^T K)\} \\ + \lambda \|R^{-1}\|_1 + p \\ = (1 + \gamma) \{ \|R - (K + \gamma I)/(1 + \gamma) \|_F^2 + (1/(1 + \gamma)) \text{tr}(K^T K) \} \\ + \lambda \|R^{-1}\|_1 + p.$$

The solution of (4.1) is soft thresholding estimator and it is given by:

$$\hat{R}_K = \frac{1}{1 + \gamma} \text{sign}(K) * \text{pmax}\{\text{abs}(K + \gamma I) - \frac{\lambda}{2}, 0\} \quad (4.2)$$

with $(\hat{R}_K)_{ii} = (K_{ii} + \gamma)/(1 + \gamma)$; $\text{pmax}(A, b)_{ij} := \text{max}(A_{ij}, b)$ is elementwise max function for each entry of the matrix A . Note that for each $(\lambda, \gamma) \in \hat{\mathcal{E}}_K^2$, \hat{R}_K is positive definite.

Choice of λ and γ : For a given value of γ , we can find the value of λ satisfying:

$$\sigma_{\min}\{(K + \gamma I) - \frac{\lambda}{2} * \text{sign}(K + \gamma I)\} > 0 \quad (4.3)$$

which can be simplified to

$$\lambda < \frac{\sigma_{\min}(K + \gamma I)}{C_{12} \sigma_{\max}(\text{sign}(K))}.$$

For some $C_{12} \geq 0.5$. Such choice of $(\lambda, \gamma) \in \hat{\mathcal{E}}_K^2$, and the estimator \hat{R}_K is positive definite. Smaller values of C_{12} yield a solution which is more sparse but may not be positive definite.

Choice of weight matrix A : For optimization problem in (3.5), the weights are chosen in following way:

Let \mathcal{E} be the set of sorted diagonal elements of the sample covariance matrix S .

i) Let k be largest index of \mathcal{E} such that k^{th} elements of \mathcal{E} is less than 1. For $i \leq k$, $a_i = \mathcal{E}_i$. For $k < i \leq p$, $a_i = 1/\mathcal{E}_i$.

ii) $A = \text{diag}(a_1, a_2, \dots, a_p)$, where $a_i = a_j / \sum_{i=1}^p a_i$. Such choice of weights allows more shrinkage of extreme sample eigenvalues than the ones in center of eigen-spectrum.

4.2 Inverse Covariance Matrix Estimation:

To get an expression of inverse covariance matrix estimate, we replace K by \hat{R}_K^{-1} in (4.2), where \hat{R}_K is a JPEN estimator of correlation matrix. We chose $(\lambda, \gamma) \in \hat{S}_K^2$. For a given γ , we chose $\lambda > 0$ satisfying:

$$\sigma_{\min}\{(\hat{R}_K^{-1} + \gamma t_1 I) - \frac{\lambda}{2} * \text{sign}(\hat{R}_K^{-1} + \gamma t_1 I)\} > 0 \tag{4.4}$$

which can be simplified to

$$\lambda < \frac{\sigma_{\min}(\hat{R}_K^{-1} + \gamma t_1 I)}{C_{12} \sigma_{\max}(\text{sign}(\hat{R}_K^{-1}))}$$

4.3 Computational Complexity

The JPEN estimator $\hat{\Sigma}_K$ has computational complexity of $O(p^2)$ as there are at most $3p^2$ multiplications for computing the estimator $\hat{\Sigma}_K$. The other existing algorithm Glasso (Friedman et al. (2008)), PDSCE (Rothman (2012)) have computational complexity of $O(p^3)$. We compare the computational timing of our algorithm to some other existing algorithms Glasso (Friedman et al. (2008)), PDSCE (Rothman (2012)). The exact timing of these algorithm also depends upon the implementation, platform etc. (we did our computations in R on a AMD 2.8GHz processor). Following the approach Bickel and Levina (2008a), the optimal tuning parameter (λ, γ) was obtained by minimizing the 5-fold cross validation error

$$(1/5) \sum_{i=1}^5 \|\hat{\Sigma}_i^v - \Sigma_i^{-v}\|_1,$$

where $\hat{\Sigma}_i^v$ is JPEN estimate of the covariance matrix based on $v = 4n/5$ observations, Σ_i^{-v} is the sample covariance matrix using $(n/5)$ observations. Figure 4.1 illustrates the total computational time taken to estimate the covariance matrix by *Glasso*, *PDSCE* and *JPEN* algorithms for different values of p for Toeplitz type of covariance matrix on log-log scale (see section §5 for Toeplitz type of covariance matrix). Although the proposed method requires optimization over a grid of values of $(\lambda, \gamma) \in \hat{S}_K^2$, our algorithm is very fast and easily scalable to large scale data analysis problems.

5. Simulation Results

We compare the performance of the proposed method to other existing methods on simulated data for five types of covariance and inverse covariance matrices.

(i) **Hub Graph:** Here the rows/columns of Σ_0 are partitioned into J equally-sized disjoint groups: $\{V_1 \cup V_2 \cup \dots \cup V_J\} = \{1, 2, \dots, p\}$, each group is associated with a **pivotal** row k . Let size $|V_1| = s$. We set $\sigma_{0k,j} = \sigma_{0j,i} = \rho$ for $i \in V_k$ and $\sigma_{0k,i} = \sigma_{0j,i} = 0$ otherwise. In our experiment, $J = \lfloor p/s \rfloor$, $k = 1, s + 1, 2s + 1, \dots$, and we always take $\rho = 1/(s + 1)$ with $J = 20$.

(ii) **Neighborhood Graph:** We first uniformly sample (y_1, y_2, \dots, y_n) from a unit square. We then set $\sigma_{0k,j} = \sigma_{0j,i} = \rho$ with probability $(\sqrt{2\pi})^{-1} \exp(-4\|y_k - y_j\|^2)$. The remaining entries of Σ_0 are set to be zero. The number of nonzero off-diagonal elements of each row or column is restricted to be smaller than $\lfloor 1/\rho \rfloor$ where ρ is set to be 0.245.

(iii) **Toeplitz Matrix:** We set $\sigma_{0k,j} = 2$ for $i = j$; $\sigma_{0k,j} = \lfloor 0.75 \rfloor^{|i-j|}$ for $|i - j| = 1, 2$; and $\sigma_{0k,j} = 0$ otherwise.

(iv) **Block Diagonal Matrix:** In this setting Σ_0 is a block diagonal matrix with varying block size. For $p = 500$ number of blocks is 4 and for $p = 1000$ the number of blocks is 6. Each block of covariance matrix is taken to be Toeplitz type matrix as in case (iii).

(v) **Cov-I type Matrix:** In this setting, we first simulate a random sample (y_1, y_2, \dots, y_p) from standard normal distribution. Let $x_i = |y_i|^{3/2} * (1 + 1/p)^{1 + \log(1 + 1/p^2)}$. Next we generate multivariate normal random vectors $\tilde{z} = (z_1, z_2, \dots, z_{5p})$ with mean vector zero and identity covariance matrix. Let U be eigenvector corresponding to sample covariance matrix of \tilde{z} . We take $\Sigma_0 = UDU'$, where $D = \text{diag}(x_1, x_2, \dots, x_p)$. This is not a sparse setting but the covariance matrix has most of eigenvalues close to zero and hence allows us to compare the performance of various methods in a setting where most of eigenvalues are close to zero and widely spread as compared to structured covariance matrices in (i)-(iv).

We chose similar structure of Ω_0 for simulations. For all these choices of covariance and inverse covariance matrices, we generate random vectors from multivariate normal distribution with varying n and p . We chose $n = 50, 100$ and $p = 500, 1000$. We compare the performance of proposed covariance matrix estimator $\hat{\Sigma}_K$ to graphical lasso [Friedman et al. (2008)], PDSCE Estimate [Rothman (2012)], Bickel and Levina's thresholding estimator (BLThresh) [Bickel and Levina (2008a)] and Ledoit-Wolf [Ledoit and Wolf (2004)] estimate of covariance matrix. The JPEN estimate $\hat{\Sigma}_K$ was computed using R software (version 3.0.2). The graphical lasso estimate of the covariance matrix was computed using R package "glasso" (<http://statweb.stanford.edu/tibs/glasso/>). The Ledoit-Wolf estimate was obtained using code from (<http://econ.uzh.ch/faculty/wolf/publications.html#9>). The PDSCE estimate was obtained using PDSCE package (<http://cran.r-project.org/web/packages/PDSCE/index.html>). The Bickel and Levina's estimator was computed as per the algorithm given in their paper. For inverse covariance matrix performance comparison

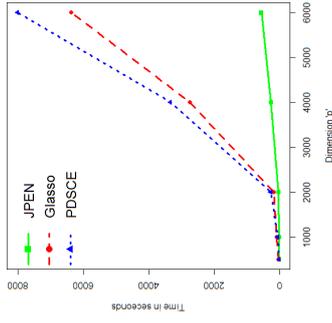


Figure 4.1: Timing comparison of JPEN, Glasso, and PDSCE.

Table 5.1: Covariance Matrix Estimation

Block type covariance matrix			
	n=50	n=1000	n=5000
Ledoit-Wolf	1.54(0.102)	2.96(0.0903)	4.271(0.0394)
Giasso	0.322(0.0235)	3.618(0.073)	0.227(0.008)
PDSCe	3.622(0.231)	4.968(0.017)	1.806(0.21)
BLThresh	2.747(0.093)	3.131(0.122)	0.887(0.04)
JPEN	2.378(0.138)	3.203(0.144)	1.124(0.088)
Hub type covariance matrix			
	n=50	n=1000	n=5000
Ledoit-Wolf	2.13(0.103)	2.43(0.043)	1.07(0.165)
Giasso	0.511(0.047)	0.551(0.005)	0.325(0.053)
PDSCe	0.735(0.106)	0.686(0.006)	0.36(0.035)
BLThresh	1.782(0.047)	2.389(0.036)	0.875(0.102)
JPEN	0.732(0.111)	0.688(0.006)	0.356(0.058)
Neighborhood type covariance matrix			
	n=50	n=1000	n=5000
Ledoit-Wolf	1.36(0.054)	2.89(0.028)	1.1(0.0331)
Giasso	0.698(0.054)	0.63(0.005)	0.428(0.047)
PDSCe	0.373(0.085)	0.468(0.007)	0.11(0.056)
BLThresh	1.526(0.074)	2.902(0.033)	0.870(0.028)
JPEN	0.454(0.0423)	0.501(0.018)	0.086(0.045)
Toepfliz type covariance matrix			
	n=50	n=1000	n=5000
Ledoit-Wolf	1.526(0.074)	2.902(0.033)	1.967(0.041)
Giasso	2.351(0.156)	3.581(0.079)	1.78(0.087)
PDSCe	3.108(0.449)	5.027(0.016)	0.795(0.076)
BLThresh	0.858(0.040)	1.206(0.059)	0.703(0.039)
JPEN	2.517(0.214)	3.205(0.16)	1.82(0.084)
Cov-I type covariance matrix			
	n=50	n=1000	n=5000
Ledoit-Wolf	33.2(0.04)	36.7(0.03)	36.2(0.03)
Giasso	15.4(0.25)	16.1(0.4)	14.0(0.03)
PDSCe	16.5(0.05)	16.33(0.04)	16.9(0.03)
BLThresh	15.7(0.04)	17.1(0.03)	13.4(0.02)
JPEN	7.1(0.042)	11.5(0.07)	8.4(0.042)

$$ARE(\hat{\Sigma}, \hat{\Sigma}) = |\log(f(S, \hat{\Sigma})) - \log(f(S, \Sigma_0))| / (|\log(f(S, \Sigma_0))|),$$

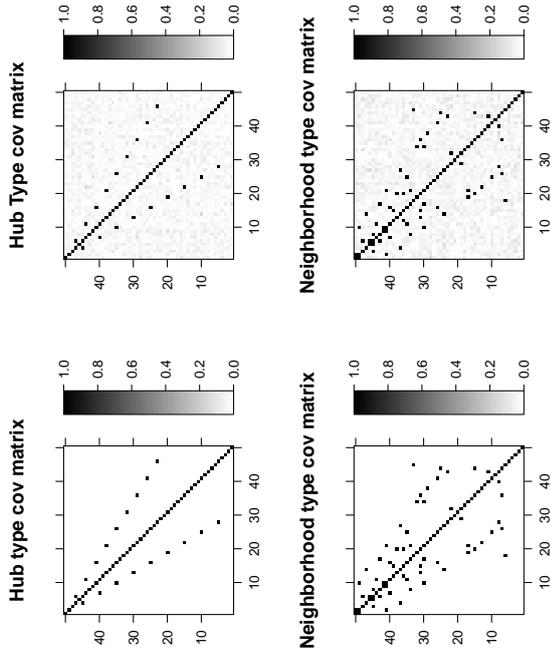
where $f(S, \cdot)$ is multivariate normal density given the sample covariance matrix S . Σ_0 is the true covariance, $\hat{\Sigma}$ is the estimate of Σ_0 based on one of the methods under consideration. Other choices of performance criteria are Kullback-Leibler used by [Yuan and Lin \(2007\)](#) and [Bickel and Levina \(2008a\)](#). The optimal values of tuning parameters were obtained over a grid of values by minimizing 5-fold cross-validation as explained in §4. The average relative

Table 5.2: Inverse Covariance Matrix Estimation

Block type covariance matrix			
	n=50	n=1000	n=5000
Giasso	4.144(0.523)	1.202(0.042)	0.168(0.136)
PDSCe	1.355(0.497)	1.201(0.044)	0.516(0.196)
CLIME	4.24(0.23)	6.56(0.25)	6.88(0.802)
JPEN	1.248(0.33)	1.106(0.029)	0.562(0.183)
Hub type covariance matrix			
	n=50	n=1000	n=5000
Giasso	1.122(0.082)	0.805(0.007)	0.07(0.038)
PDSCe	0.717(0.108)	0.702(0.007)	0.358(0.046)
CLIME	10.5(0.329)	10.6(0.219)	6.98(0.237)
JPEN	0.684(0.051)	0.669(0.003)	0.34(0.024)
Neighborhood type covariance matrix			
	n=50	n=1000	n=5000
Giasso	1.597(0.109)	0.879(0.013)	1.29(0.847)
PDSCe	0.587(0.13)	0.736(0.014)	0.094(0.058)
CLIME	10.5(0.535)	11.5(0.233)	10.5(0.563)
JPEN	0.551(0.075)	0.691(0.008)	0.066(0.042)
Toepfliz type covariance matrix			
	n=50	n=1000	n=5000
Giasso	2.862(0.475)	2.89(0.048)	2.028(0.267)
PDSCe	1.223(0.5)	1.238(0.065)	0.49(0.269)
CLIME	4.91(0.22)	7.597(0.34)	5.27(1.14)
JPEN	1.151(0.333)	2.718(0.032)	0.607(0.196)
Cov-I type covariance matrix			
	n=50	n=1000	n=5000
Giasso	54.0(0.19)	190.(5.91)	14.7(0.37)
PDSCe	28.8(0.19)	45.8(0.32)	16.9(0.04)
CLIME	59.8(0.82)	207.5(3.44)	15.4(0.03)
JPEN	26.3(0.36)	7.0(0.07)	15.7(0.08)

error and their standard deviations (in percentage) for covariance and inverse covariance matrix estimates are given in Table 5.1 and Table 5.2, respectively. The numbers in the bracket are the standard errors of relative error based on the estimates using different methods. Among all the methods JPEN and PDSCe perform similar for most of choices of n and p for all five type of covariance matrices. This is due to the fact that both PDSCe and JPEN use quadratic optimization function with a different penalty function. The behavior of Bickel and Levina's estimator is quite good in Toepfliz case where it performs better than the other methods. For this type of covariance matrix, the entries away from the diagonal decay to zero and therefore soft-thresholding estimators like BLThresh perform better in this setting. However for neighborhood and hub type covariance matrix which are not necessarily banded type, Bickel and Levina estimator is not a natural choice as their estimator would fail to recover the underlying sparsity pattern. The performance of Ledoit-Wolf estimator is not very encouraging for Cov-I type matrix. The Ledoit-Wolf estimator shrinks the sample covariance matrix towards identity and hence the eigenvalues estimates are highly shrunk

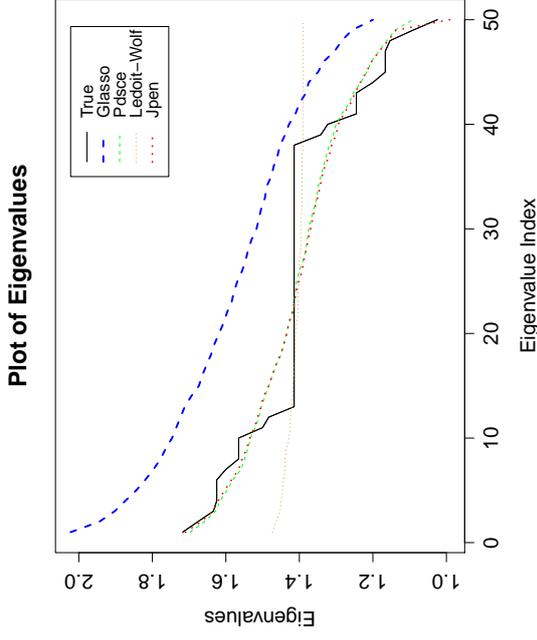
Figure 5.1: Heat-map of zeros identified in covariance matrix out of 50 realizations. White color is 50/50 zeros identified, black color is 0/50 zeros identified.



towards one. This is also visible in eigenvalues plot in Figure 5.2 and Figure 5.3. For Cov-I type covariance matrix where most of eigenvalues are close to zero and widely spread, the performance of JPEN estimator is impressive. The eigenplot in Figure 5.3 shows that among all the methods, estimates of eigenvalues of JPEN estimator are most consistent with true eigenvalues. This clearly shows the advantage of JPEN estimator of covariance matrix when the true eigenvalues are dispersed or close to zero. The eigenvalues plot in Figure 5.2 shows that when eigen-spectrum of true covariance matrix are not highly dispersed, the JPEN and PDSCE estimates of eigenvalues are almost the same. This phenomenon is also apparent in Figure 2.1. Also Ledoit-Wolf estimator heavily shrinks the eigenvalues towards the center and thus underestimates the true eigen-spectrum.

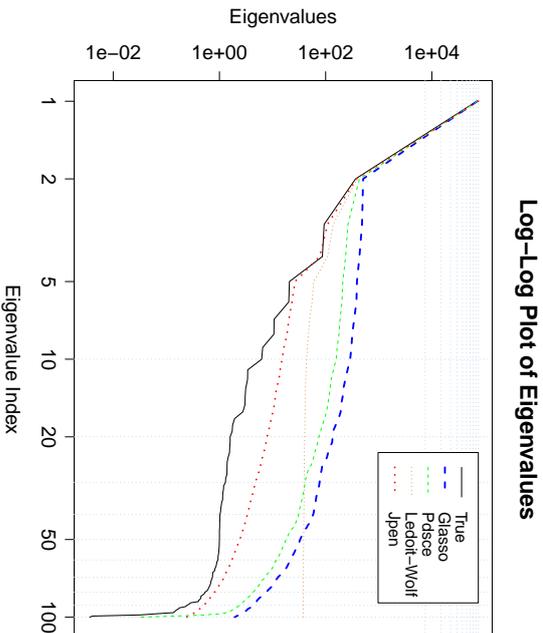
For inverse covariance matrix, we compare glasso, CLIME and PDSCE estimates with proposed JPEN estimator. The JPEN estimator $\hat{\Omega}_K$ outperforms other methods for the most of the choices of n and p for all five types of inverse covariance matrices. Additional simulations (not included here) show that for $n \approx p$, all the underlying methods perform similarly and the estimates of their eigenvalues are also well aligned with true values. However in high dimensional setting, for large p and small n , their performance is different as seen in simulations of Table 5.1 and Table 5.2. Figure 5.1 shows the recovery of non-zero and zero

Figure 5.2: Eigenvalues plot for $n = 100, p = 50$ based on 50 realizations for neighborhood type of covariance matrix



entries of true covariance matrix based on JPEN estimator $\hat{\Sigma}_K$ based on 50 realizations. The estimator recovers the true zeros for about 90% of times for Hub and Neighborhood type of covariance matrix. It also reflect the recovery of true structure of non-zero entries and actual pattern among the rows/columns of covariance matrix. To see the implication of eigenvalues shrinkage penalty as compared to other methods, we plot (Figure 5.2) the eigenvalues of estimated covariance matrix for $n = 100, p = 50$ for neighborhood type of covariance matrix. The JPEN estimates of eigen-spectrum are well aligned with true ones and closest being PDSCE estimates of eigenvalues. Figure 5.3 shows the recovery of eigenvalues based on estimates using different methods for Cov-I type covariance matrix. For this particular simulation, the eigenvalues are chosen differently than the one described in (v) of §5. The eigenvalues of true covariance matrix are taken to be very diverse with maximum about 10^6 and smallest eigenvalue about 10^{-6} . For Cov-I type of matrix, JPEN estimates of eigenvalues are better than other methods.

Figure 5.3: Eigenvalues plot for $n = 100, p = 100$ based on 50 realizations for *Cov-I* type matrix



6. Colon Tumor Classification Example

In this section, we compare performance of JPEN estimator of inverse covariance matrix for tumors classification using Linear Discriminant Analysis (LDA). The gene expression data (Alon et al. (1999)) consists of 40 tumorous and 22 non-tumorous adenocarcinoma tissue. After preprocessing, data was reduced to a subset of 2,000 gene expression values with the largest minimal intensity over the 62 tissue samples (source: <http://genomics-prubz.princeton.edu/oncology/affydata/index.html>). In our analysis, we reduced the number of genes by selecting p most significant genes based on logistic regression. We obtain estimates of inverse covariance matrix for $p = 50, 100, 200$ and then use LDA to classify these tissues as either tumorous or non-tumorous (normal). We classify each test observation x to either class $k = 0$ or $k = 1$ using the LDA rule

$$\hat{\delta}_k(x) = \arg \max_k \left\{ x^T \hat{\Omega} \hat{\mu}_k - \frac{1}{2} \hat{\mu}_k^T \hat{\Omega} \hat{\mu}_k + \log(\hat{\pi}_k) \right\}, \tag{6.1}$$

where $\hat{\pi}_k$ is the proportion of class k observations in the training data, $\hat{\mu}_k$ is the sample mean for class k on the training data, and $\hat{\Omega} := \hat{\Sigma}^{-1}$ is an estimator of the inverse of the common covariance matrix on the training data computed. Tuning parameters λ and γ were

chosen using 5-fold cross validation. To create training and test sets, we randomly split the data into a training and test set of sizes 42 and 20 respectively; following the approach used by Wang et al. (2007), the training set has 27 tumor samples and 15 non-tumor samples. We repeat the split at random 100 times and measure the average classification error. Since

Table 6.1: Averages and standard errors of classification errors over 100 replications in %.

Method	p=50	p=100	p=200
	Logistic Regression	21.0(0.84)	19.31(0.89)
SVM	16.70(0.85)	16.76(0.97)	18.18(0.96)
Naive Bayes	13.3(0.75)	14.33(0.85)	14.63(0.75)
Graphical Lasso	10.9(1.3)	9.4(0.89)	9.8(0.90)
Joint Penalty	9.9(0.98)	8.9(0.93)	8.2(0.81)

we do not have separate validation set, we do the 5-fold cross validation on training data. At each split, we divide the training data into 5 subsets (fold) where 4 subsets are used to estimate the covariance matrix and one subset is used to measure the classifier's performance. For each split, this procedure is repeated 5 times by taking one of the 5 subsets as validation data. An optimal combination of λ and γ is obtained by minimizing the 5-fold cross validation error.

The average classification errors with standard errors over the 100 splits are presented in Table 6.1. Since the sample size is less than the number of genes, we omit the inverse sample covariance matrix as it is not well defined and instead include the naive Bayes' and support vector machine classifiers. Naive Bayes has been shown to perform better than the sample covariance matrix in high-dimensional settings (Bickel and Levina (2004)). Support Vector Machine (SVM) is another popular choice for high dimensional classification tool. Among all the methods covariance matrix based LDA classifiers perform far better than Naive Bayes, SVM and Logistic Regression. For all other classifiers the classification performance deteriorates for increasing p . For larger p , i.e., when more genes are added to the data set, the classification performance of JPEN estimate based LDA classifier initially improves but it deteriorates for large p . For $p = 2000$, the classifier based on inverse covariance matrix has accuracy of 30%. This is due to the fact that as dimension of covariance matrix increases, the estimator does not remain very informative.

7. Summary

We have proposed and analyzed regularized estimation of large covariance and inverse covariance matrix using joint penalty. The proposed JPEN estimators are optimal under spectral norm for underlying classes of sparse and well-conditioned covariance and inverse covariance matrices. We also establish its theoretical consistency in Frobenius norm. One of its biggest advantage is that the optimization carries no computational burden and and the resulting algorithm is very fast and easily scalable to large scale data analysis problems. The extensive simulation shows that the proposed estimators performs well for a number

of structured covariance and inverse covariance matrices. Also when the eigenvalues of underlying true covariance matrix are highly dispersed, it outperforms other methods (based on simulation analysis). The JPEN estimator recovers the sparsity pattern of the true covariance matrix and provides a good approximation of the underlying eigen-spectrum and hence we expect that PCA will be one of the most important application of the method. Although the proposed JPEN estimators of covariance and inverse covariance matrix do not require any assumption on the structure of true covariance and inverse covariance matrices respectively, any prior knowledge of structure of true covariance matrix might be helpful to choose a suitable weight matrix and hence improve estimation.

Acknowledgments

The author would like to express the deep gratitude to Professor Hira L. Koul for his valuable and constructive suggestions during the planning and development of this research work. The author would like to thank Dr. Adam Rothman for his valuable discussion and suggestion. The author would also like to thank the two anonymous referees and the action editor Dr. Jie Peng for insightful reviews that helped to improve the original manuscript substantially.

References

- U. Alon, Barkai N., Notterman D., Gish K., Ybarra S., Mack D., and Levine A. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceeding of National Academy of Science USA*, 96(12):6745-6750, 1999.
- O. Banerjee, L. El Ghaoui, and A. dAspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine Learning Research*, 9:485-516, 2008.
- J. Bein and R. Tibshirani. Sparse estimation of a covariance matrix. *Biometrika*, 98: 807-820, 2011.
- P. Bickel and E. Levina. Some theory for fishers linear discriminant function, "naive bayes, and some alternatives when there are many more variables than observations. *Bernoulli*, 10:989-1010, 2004.
- P. Bickel and E. Levina. Covariance regularization by thresholding. *The Annals of Statistics*, 36(Mar):2577-2604, 2008a.
- P. Bickel and E. Levina. Regularized estimation of large covariance matrices. *Annals of Statistics*, 36:199-227, 2008b.
- T. Cai, W. Liu, and X. Luo. , a constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of American Statistical Association*, 106:2594-607, 2011.
- T. Cai, Z. Ren, and H. Zhou. Estimating structured high-dimensional covariance and precision matrices: Optimal rates and adaptive estimation. *Electronic Journal of Statistics*, 2015.
- R. Clarke, Ransom H., Wang A., Xuan J., Liu M., Gehan E., and Wang Y. The properties of high-dimensional data spaces: implications for exploring gene and protein expression data. *Nat Rev Cancer*, 8:37-49, 2008.
- J. Friedman, Hastie T., and Tibshirani R. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432-441, 2008.
- S. Geman. A limit theorem for the norm of random matrices. *The Annals of Statistics*, 8(2):252-261, 1980.
- I. Johnstone and Y. Lu. Sparse principal components analysis. *Unpublished Manuscript*, 2004.
- N. Karoui. Operator norm consistent estimation of large dimensional sparse covariance matrices. *The Annals of Statistics*, 36:2717-2756, 2008a.
- N. Karoui. Spectrum estimation for large dimensional covariance matrices using random matrix theory. *The Annals of Statistics*, 36(6):2757-2790, 2008b.
- O. Ledoit and M. Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88:365-411, 2004.
- V. Marcenko and L. Pastur. Distributions of eigenvalues of some sets of random matrices. *Math. USSR-Sb*, 1:507-536, 1967.
- K. Mardia, Kent J., and Bibby J. *Multivariate Analysis*, volume 1. Academic Press, New York, NY, 1979.
- Ashwini Maurya. A joint convex penalty for inverse covariance matrix estimation. *Computational Statistics and Data Analysis*, 75:15-27, 2014.
- Meinshausen and P. Bühlmann. High dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34:1436-1462, 2006.
- G. Pass, Chowdhury A., and Torgeson C. "a picture of search". *The First International Conference on Scalable Information Systems*, 6, 2006.
- P. Ravikumar, Wainwright M. and Raskutti G., and Yu B. High-dimensional covariance estimation by minimizing ℓ_1 -penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935-980, 2011.
- A. Rothman. Positive definite estimators of large covariance matrices. *Biometrika*, 99: 733-740, 2012.
- A. Rothman, Bickel P. J., Levina E., and Zhu J. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494-515, 2008.

C. Stein. Estimation of a covariance matrix. *Rietz lecture, 39th Annual Meeting IMS, Atlanta, Georgia, 1975.*

M. Wainwright, Ravikumar P, and Lafferty J. High-dimensional graphical model selection using l_1 -regularized logistic regression. *Proceedings of Advances in Neural In formation Processing Systems*, 2006.

L. Xue, Ma S., and Zou Hui. Positive-definite l_1 -penalized estimation of large covariance matrices. *Journal of American Statistical Association*, 107(500):983-990, 2012.

M. Yuan. Sparse inverse covariance matrix estimation via linear programming. *Journal of Machine Learning Research*, 11:2261-2286, 2009.

M. Yuan and Y. Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19-35, 2007.

H. Zou, Hastie T., and Tibshirani R. Sparse principal components analysis. *Journal of Computational and Graphical Statistics*, 15:265-286, 2006.

Appendix A.

Proof of Lemma 3.1

Let

$$f(R) = \|R - K\|^2 + \lambda \|R^{-1}\|_1 + \gamma \sum_{i=1}^p \{\sigma_i(R) - \sigma_{R_i}\}^2. \tag{.1}$$

where $\bar{\sigma}_R$ is the mean of eigenvalues of R . Due to the constraint $\text{tr}(R) = p$, we have $\bar{\sigma}_R = 1$. The third term of (.1) can be written as

$$\sum_{i=1}^p \{\sigma_i(R) - \sigma_{R_i}\}^2 = \text{tr}(R^2) - 2 \text{tr}(R) + p$$

We obtain,

$$\begin{aligned} f(R) &= \text{tr}(R^2) - 2 \text{tr}(RK) + \text{tr}(K^2) + \lambda \|R^{-1}\|_1 + \gamma \{\text{tr}(R^2) - 2 \text{tr}(R) + p\} \\ &= \text{tr}(R^2(1 + \gamma)) - 2 \text{tr}(K + \gamma I) + \text{tr}(K^2) + \lambda \|R^{-1}\|_1 + p \\ &= (1 + \gamma) \|R - (K + \gamma I)/(1 + \gamma)\|^2 + \text{tr}(K^2) + \lambda \|R^{-1}\|_1 + p \end{aligned} \tag{.2}$$

This is quadratic in R with a l_1 penalty to the off-diagonal entries of R , therefore a convex function in R .

Proof of Lemma 3.2 The solution to (.2) satisfies:

$$2(R - (K + \gamma I))(1 + \gamma)^{-1} + \lambda \frac{\partial \|R^{-1}\|_1}{\partial R} = 0 \tag{.3}$$

where $\frac{\partial \|R^{-1}\|_1}{\partial R}$ is given by:

$$\frac{\partial \|R^{-1}\|_1}{\partial R} = \begin{cases} 1 & : \text{if } R_{ij} > 0 \\ -1 & : \text{if } R_{ij} < 0 \\ \tau \in (-1, 1) & : \text{if } R_{ij} = 0 \end{cases}$$

Note that $\|R^{-1}\|_1$ has same value irrespective of sign of R , therefore the right hand side of (.2) is minimum if :

$$\text{sign}(R) = \text{sign}(K + \gamma I) = \text{sign}(K)$$

$\forall \epsilon > 0$, using (.3), $\sigma_{\min}\{K + \gamma I - \frac{\lambda}{2} \text{sign}(K)\} > \epsilon$ gives a $(\lambda, \gamma) \in \hat{S}_\epsilon^K$ and such a choice of (λ, γ) guarantees the estimator to be positive definite.

Remark: Intuitively, a larger γ shrinks the eigenvalues towards center which is 1, a larger γ would result in positive definite estimator, whereas a larger λ results in sparse estimate. A combination of (λ, γ) results in a sparse and well-conditioned estimator. In particular case, when K is diagonal matrix: the $\lambda < 2 * \gamma$.

Proof of Theorem 3.1 Define the function $Q(\cdot)$ as following:

$$Q(R) = f(R) - f(R_0)$$

where R_0 is the true correlation matrix and R is any other correlation matrix. Let $R = UDU^T$ be eigenvalue decomposition of R , D is diagonal matrix of eigenvalues and U is matrix of eigenvectors. We have,

$$Q(R) = \|R - K\|_F^2 + \lambda \|R^-\|_1 + \gamma \operatorname{tr}(D^2 - 2D + p) - \|R_0 - K\|_F^2 - \lambda \|R_0^-\|_1 - \gamma \operatorname{tr}(D_0^2 - 2D_0 + p) \quad (4)$$

$R_0 = U_0 D_0 U_0^T$ is eigenvalue decomposition of R_0 . Let $\Theta_n(M) := \{\Delta : \Delta = \Delta^T, \|\Delta\|_2 = M/n, 0 < M < \infty\}$. The estimate \hat{R} minimizes the $Q(R)$ or equivalently $\hat{\Delta} = \hat{R} - R_0$ minimizes the $G(\Delta) = Q(R_0 + \Delta)$. Note that $G(\Delta)$ is convex and if $\hat{\Delta}$ be its solution, then we have $G(\hat{\Delta}) \leq G(0) = 0$. Therefore if we can show that $G(\Delta)$ is non-negative for $\Delta \in \Theta_n(M)$, this will imply that the $\hat{\Delta}$ lies within sphere of radius M/n . We require $r_n = o\left(\sqrt{(p+s) \log p/n}\right)$.

$$\begin{aligned} \|R - K\|_F^2 - \|R_0 - K\|_F^2 &= \operatorname{tr}(R^T R - 2R^T K + K^T K) - \operatorname{tr}(R_0^T R_0 - 2R_0^T K + K^T K) \\ &= \operatorname{tr}(R^T R - R_0^T R_0) - 2 \operatorname{tr}((R - R_0)^T K) \\ &= \operatorname{tr}((R_0 + \Delta)^T (R_0 + \Delta) - R_0^T R_0) - 2 \operatorname{tr}(\Delta^T K) \\ &= \operatorname{tr}(\Delta^T \Delta) - 2 \operatorname{tr}(\Delta^T (K - R_0)) \end{aligned}$$

Next, we bound term involving K in above expression, we have

$$\begin{aligned} |\operatorname{tr}(\Delta^T (R_0 - K))| &\leq \sum_{i \neq j} |\Delta_{ij} (R_{0ij} - K_{ij})| \\ &\leq \max_{i \neq j} (|R_{0ij} - K_{ij}|) \|\Delta^-\|_1 \\ &\leq C_0 (1 + \tau) \sqrt{\frac{\log p}{n}} \|\Delta^-\|_1 \end{aligned}$$

holds with high probability by a result (Lemma 1) from Ravikumar et al. (2011) on the tail inequality for sample covariance matrix of sub-Gaussian random vectors and where $C_1 = C_0(1 + \tau)$, $C_0 > 0$. Next we obtain upper bound on the terms involving γ in (4). we have,

$$\begin{aligned} &\operatorname{tr}(D^2 - 2D) - \operatorname{tr}(D_0^2 - 2D_0) \\ &= \operatorname{tr}\{R^2 - R_0^2\} - 2 \operatorname{tr}\{R - R_0\} = \operatorname{tr}(R_0 + \Delta)^2 - \operatorname{tr}(R_0^2) \\ &= 2 \operatorname{tr}(R_0 \Delta) + \operatorname{tr}(\Delta^T \Delta) \leq 2 \sqrt{s} \|\Delta\|_F + \|\Delta\|_F^2. \end{aligned}$$

using Cauchy-Schwarz inequality. To bound the term $\lambda(\|R^-\|_1 - \|R_0^-\|_1) = \lambda(\|\Delta^-\|_1 - \|R_0^-\|_1)$, let E be index set as defined in Assumption A.2 of Theorem 3.2. Then using the triangle inequality, we obtain,

$$\begin{aligned} \lambda(\|\Delta^-\|_1 - \|R_0^-\|_1) &= \lambda(\|\Delta_E^-\|_1 + \|\Delta_{E^c}^-\|_1 - \|R_0^-\|_1) \\ &\geq \lambda(\|R_0^-\|_1 - \|\Delta_{E^c}^-\|_1 + \|\Delta_E^-\|_1 - \|R_0^-\|_1) \\ &\geq \lambda(\|\Delta_E^-\|_1 - \|\Delta_{E^c}^-\|_1) \end{aligned}$$

Let $\lambda = (C_1/\epsilon) \sqrt{\log p/n}$, $\gamma = (C_1/\epsilon_1) \sqrt{\log p/n}$, where $(\lambda, \gamma) \in \hat{S}_K^K$, we obtain,

$$\begin{aligned} G(\Delta) &\geq \operatorname{tr}(\Delta^T \Delta) (1 + \gamma) - 2C_1 \left\{ \sqrt{\frac{\log p}{n}} (\|\Delta^-\|_1) + \frac{1}{\epsilon_1} \sqrt{\frac{s \log p}{n}} \|\Delta\|_F \right\} \\ &\quad + \frac{C_1}{\epsilon} \sqrt{\frac{\log p}{n}} (\|\Delta_E^-\|_1 - \|\Delta_{E^c}^-\|_1) \\ &\geq \|\Delta\|_F^2 (1 + \gamma) - 2C_1 \sqrt{\frac{\log p}{n}} (\|\Delta_E^-\|_1 + \|\Delta_{E^c}^-\|_1) \\ &\quad - \frac{C_1}{\epsilon} \sqrt{\frac{\log p}{n}} (\|\Delta_E^-\|_1 - \|\Delta_{E^c}^-\|_1) - \frac{2C_1}{\epsilon_1} \sqrt{\frac{s \log p}{n}} \|\Delta\|_F. \end{aligned}$$

Also because $\|\Delta_E^-\|_1 = \sum_{(i,j) \in E, i \neq j} \Delta_{ij} \leq \sqrt{s} \|\Delta^-\|_F$,

$$-2C_1 \sqrt{\frac{\log p}{n}} \|\Delta_E^-\|_1 + \frac{C_1}{\epsilon} \sqrt{\frac{\log p}{n}} \|\Delta_{E^c}^-\|_1 \geq \sqrt{\frac{\log p}{n}} \|\Delta_{E^c}^-\|_1 (-2C_1 + \frac{C_1}{\epsilon}) \geq 0$$

for sufficiently small ϵ . Therefore,

$$\begin{aligned} G(\Delta) &\geq \|\Delta\|_F^2 (1 + \frac{C_1}{\epsilon_1} \sqrt{\frac{\log p}{n}}) - C_1 \sqrt{\frac{s \log p}{n}} \|\Delta^+\|_F \{1 + 1/\epsilon + 2/\epsilon_1\} \\ &\geq \|\Delta\|_F^2 \left[1 + \frac{C_1}{\epsilon_1} \sqrt{\frac{\log p}{n}} - \frac{C_1}{M} \{1 + 1/\epsilon + 2/\epsilon_1\} \right] \\ &\geq 0, \end{aligned}$$

for all sufficiently large n and M . Which proves the first part of theorem. To prove the operator norm consistency, we have,

$$\begin{aligned} \|\hat{\Sigma}_K - \Sigma_0\| &= \|\hat{W} \hat{R} \hat{W} - W K W\| \\ &\leq \|\hat{W} - W\| \|\hat{R} - K\| \|\hat{W} - W\| \\ &\quad + \|\hat{W} - W\| (\|\hat{R}\| \|W\| + \|\hat{W}\| \|K\|) + \|\hat{R} - K\| \|\hat{W}\| \|W\|. \end{aligned}$$

using sub-multiplicative norm property $\|AB\| \leq \|A\| \|B\|$. Since $\|K\| = O(1)$ and $\|\hat{R} - K\|_F = O(\sqrt{\frac{s \log p}{n}})$ these together implies that $\|\hat{R}\| = O(1)$. Also,

$$\begin{aligned} \|\hat{W} - W\| &= \max_{1 \leq i \leq p} \sum_{i=1}^p (|\hat{w}_i^2 - w_i^2|) x_i^2 \leq \max_{1 \leq i \leq p} (|\hat{w}_i^2 - w_i^2|) \sum_{i=1}^p x_i^2 \\ &= \max_{1 \leq i \leq p} (|\hat{w}_i^2 - w_i^2|) = O\left(\sqrt{\frac{\log p}{n}}\right). \end{aligned}$$

holds with high probability by using a result (Lemma 1) from Ravikumar et al. (2011). Next we shall show that $\|\hat{W} - W\| \asymp \|\hat{W}^2 - W^2\|$, (where $A \succ B$ means $A = O_P(B)$ and

$B=O_p(A)$). We have,

$$\begin{aligned} \|\hat{W} - W\| &= \max_{\|x\|_2=1} \sum_{i=1}^p |(\hat{w}_i - w_i)x_i^2| = \max_{\|x\|_2=1} \sum_{i=1}^p \left(\frac{\hat{w}_i^2 - w_i^2}{\hat{w}_i + w_i} \right) |x_i^2| \\ &\asymp \sum_{i=1}^p |(\hat{w}_i^2 - w_i^2)| x_i^2 = C_3 \|\hat{W}^2 - W^2\|. \end{aligned}$$

where we have used the fact that the true standard deviations are well above zero, i.e., $\exists 0 < C_3 < \infty$ such that $1/C_3 \leq w_i^{-1} \leq C_3 \forall i = 1, 2, \dots, p$, and sample standard deviation are all positive, i.e, $\hat{w}_i > 0 \forall i = 1, 2, \dots, p$. Now since $\|\hat{W}^2 - W^2\| \asymp \|\hat{W} - W\|$, this follows that $\|\hat{W}\| = O(1)$ and we have $\|\sum_{i=1}^p \sigma_i(\Sigma) - \sigma_\Sigma\|^2 = O\left(\frac{s \log p}{n} + \frac{\log p}{n}\right)$. This completes the proof.

Proof of Theorem 3.2 Let

$$f(\Sigma) = \|\Sigma - S\|_F^2 + \lambda \|\Sigma^-\|_1 + \gamma \sum_{i=1}^p \{\sigma_i(\Sigma) - \sigma_\Sigma\}^2,$$

Similar to the proof of theorem (3.1), define the function $Q_1(\cdot)$ as following:

$$Q_1(\Sigma) = f(\Sigma) - f(\Sigma_0)$$

where Σ_0 is the true covariance matrix and Σ is any other covariance matrix. Let $\Sigma = UDU^T$ be eigenvalue decomposition of Σ , D is diagonal matrix of eigenvalues and U is matrix of eigenvectors. We have,

$$\begin{aligned} Q_1(\Sigma) &= \|\Sigma - S\|_F^2 + \lambda \|\Sigma^-\|_1 + \gamma \operatorname{tr}(D^2) - (\operatorname{tr}(D))^2/p \\ &\quad - \|\Sigma_0 - S\|_F^2 - \lambda \|\Sigma_0^-\|_1 - \gamma \operatorname{tr}(D_0^2) - (\operatorname{tr}(D_0))^2/p \end{aligned} \tag{.5}$$

where $A = \operatorname{diag}(a_1, a_2, \dots, a_p)$ and $\Sigma_0 = U_0 D_0 U_0^T$ is eigenvalue decomposition of Σ_0 . Write $\Delta = \Sigma - \Sigma_0$, and let $\Theta_n(M) := \{\Delta : \Delta = \Delta^T, \|\Delta\|_2 = M r_n, 0 < M < \infty\}$. The estimate $\hat{\Sigma}$ minimizes the $Q(\Sigma)$ or equivalently $\hat{\Delta} = \hat{\Sigma} - \Sigma_0$ minimizes the $G(\Delta) = Q(\Sigma_0 + \Delta)$. Note that $G(\Delta)$ is convex and if $\hat{\Delta}$ be its solution, then we have $G(\Delta) \leq G(0) = 0$. Therefore if we can show that $G(\Delta)$ is non-negative for $\Delta \in \Theta_n(M)$, this will imply that the $\hat{\Delta}$ lies within sphere of radius $M r_n$. We require $\sqrt{(p+s) \log p} = o(\sqrt{r_n})$.

$$\begin{aligned} \|\Sigma - S\|_F^2 - \|\Sigma_0 - S\|_F^2 &= \operatorname{tr}(\Sigma^T \Sigma - 2\Sigma^T S + S^T S) - \operatorname{tr}(\Sigma_0^T \Sigma_0 - 2\Sigma_0^T S + S^T S) \\ &= \operatorname{tr}(\Sigma^T \Sigma - \Sigma_0^T \Sigma_0) - 2 \operatorname{tr}(\Sigma - \Sigma_0) S \\ &= \operatorname{tr}((\Sigma_0 + \Delta)^T (\Sigma_0 + \Delta) - \Sigma_0^T \Sigma_0) - 2 \operatorname{tr}(\Delta^T S) \\ &= \operatorname{tr}(\Delta^T \Delta) - 2 \operatorname{tr}(\Delta^T (S - \Sigma_0)) \end{aligned}$$

Next, we bound term involving S in above expression, we have

$$\begin{aligned} |\operatorname{tr}(\Delta(\Sigma_0 - S))| &\leq \sum_{i \neq j} |\Delta_{ij}(\Sigma_{0ij} - S_{ij})| + \sum_{i=1}^p |\Delta_{ii}(\Sigma_{0ii} - S_{ii})| \\ &\leq \max_{i \neq j} (|\Sigma_{0ij} - S_{ij}|) \|\Delta^-\|_1 + \sqrt{p} \max_{i=1}^p (|\Sigma_{0ii} - S_{ii}|) \sqrt{\sum_{i=1}^p \Delta_{ii}^2} \\ &\leq C_0(1 + \tau) \max(\Sigma_{0ii}) \left\{ \sqrt{\frac{\log p}{n}} \|\Delta^-\|_1 + \sqrt{\frac{p \log p}{n}} \|\Delta^+\|_2 \right\} \\ &\leq C_1 \left\{ \sqrt{\frac{\log p}{n}} \|\Delta^-\|_1 + \sqrt{\frac{p \log p}{n}} \|\Delta^+\|_2 \right\} \end{aligned}$$

holds with high probability by a result (Lemma 1) from Ravikumar et al. (2011) where $C_1 = C_0(1 + \tau) \max(\Sigma_{0ii})$, $C_0 > 0$ and Δ^+ is matrix Δ with all off-diagonal elements set to zero. Next we obtain upper bound on the terms involving γ in (3.7). we have,

$$\begin{aligned} \text{(i)} \quad &\operatorname{tr}(D^2) - (\operatorname{tr}(D))^2/p - \operatorname{tr}(D_0^2) - (\operatorname{tr}(D_0))^2/p \\ &= \operatorname{tr}(\Sigma^2) - \operatorname{tr}(\Sigma_0^2) - (\operatorname{tr}(\Sigma))^2/p + (\operatorname{tr}(\Sigma_0))^2/p \\ &\leq \operatorname{tr}(\Sigma_0 + \Delta)^2 - \operatorname{tr}(\Sigma_0)^2 \\ &= \operatorname{tr}(\Delta)^2 + 2 \operatorname{tr}(\Delta^2 \Sigma_0) \leq \operatorname{tr}(\Delta)^2 + C_1 \sqrt{s} \|\Delta\|_F \end{aligned}$$

$$\begin{aligned} \text{(ii)} \quad &\operatorname{tr}(\Sigma^2) - (\operatorname{tr}(\Sigma_0))^2 \\ &= (\operatorname{tr}(\Sigma_0 + \Delta))^2 - (\operatorname{tr}(\Sigma_0))^2 \\ &\leq (\operatorname{tr}(\Delta))^2 + 2 \operatorname{tr}(\Sigma_0) \operatorname{tr}(\Delta) \leq p \|\Delta\|_F^2 + 2 \bar{k} p \sqrt{p} \|\Delta^+\|_F. \end{aligned}$$

Therefore the γ term can be bounded by $2\|\Delta\|_F^2 + (C_1 \sqrt{s} + 2\sqrt{p} \bar{k}) \|\Delta\|_F$. We bound the term involving λ as in similar to the proof of Theorem 3.1. For $\lambda > \gamma \asymp \sqrt{\frac{\log p}{n}}$, the proof follows very similar to Theorem 3.1.

Proof of Theorem 3.3. To bound the cross product term involving Δ and \hat{R}_K^{-1} , we have,

$$\begin{aligned} |\operatorname{tr}((R_0^{-1} - \hat{R}_K^{-1})\Delta)| &= |\operatorname{tr}(R_0^{-1}(\hat{R}_K - R_0)\hat{R}_K^{-1}\Delta)| \\ &\leq \sigma_1(R_0^{-1}) |\operatorname{tr}((\hat{R}_K - R_0)\hat{R}_K^{-1}\Delta)| \\ &\leq \bar{k} \sigma_1(\hat{R}_K^{-1}) |\operatorname{tr}((\hat{R}_K - R_0)\Delta)| \\ &\leq \bar{k} \bar{k}_1 |\operatorname{tr}((\hat{R}_K - R_0)\Delta)|. \end{aligned}$$

where $\sigma_{\min}(\hat{R}_K) \geq (1/\bar{k}_1) > 0$, is a positive lower bound on the eigenvalues of JPEN estimate \hat{R}_K of correlation matrix R_0 . Such a constant exist by Lemma 3.2. Rest of the proof closely follows as that of Theorem 3.1.

Proof of Theorem 3.4. We bound the term $\operatorname{tr}((\hat{\Omega}_S - \Omega_0)\Delta)$ similar to that in proof of Theorem 3.3. Rest of the proof closely follows to that Theorem 3.2.

String and Membrane Gaussian Processes

Yves-Laurent Kom Samo

Stephen J. Roberts

Department of Engineering Science and Oxford-Man Institute
University of Oxford
Eagle House, Walton Well Road,
OX2 6ED, Oxford, United Kingdom

YLKS@ROBOTS.OX.AC.UK

SJROB@ROBOTS.OX.AC.UK

nonstationary covariance functions, the development of which is still an active subject of research. Secondly, inference under GP priors often consists of looking at the values of the GP at all input points as a jointly Gaussian vector with fully dependent coordinates, which induces a memory requirement and time complexity respectively squared and cubic in the training data size, and thus is intractable for large data sets. We refer to this approach as the *standard GP paradigm*. The framework we introduce in this paper addresses both of the above limitations.

Our work is rooted in the observation that, from a Bayesian nonparametric perspective, it is inefficient to define a stochastic process through *fully-dependent* marginals, as it is the case for Gaussian processes. Indeed, if a stochastic process $(f(x))_{x \in \mathbb{R}^d}$ has fully dependent marginals and exhibits no additional conditional independence structure then, when f is used as functional prior and some observations related to $(f(x_1), \dots, f(x_n))$ are gathered, namely (y_1, \dots, y_n) , the *additional* memory required to take into account an additional piece of information (y_{n+1}, x_{n+1}) grows in $\mathcal{O}(n)$, as one has to keep track of the extent to which y_{n+1} informs us about $f(x_i)$ for every $i \leq n$, typically through a covariance matrix whose size will increase by $2n + 1$ terms. Clearly, this is inefficient, as y_{n+1} is unlikely to be informative about $f(x_i)$, unless x_i is sufficiently close to x_{n+1} . More generally, the larger n , the less information a single additional pair (y_{n+1}, x_{n+1}) will add to existing data, and yet the increase in memory requirement will be much higher than that required while processing earlier and more informative data. This inefficiency in resource requirements extends to computational time, as the *increase* in computational time resulting from adding (y_{n+1}, x_{n+1}) typically grows in $\mathcal{O}(n^2)$, which is the difference between the numbers of operations required to invert a $n \times n$ matrix and to invert a $(n+1) \times (n+1)$ matrix. A solution for addressing this inefficiency is to appropriately limit the extent to which values $f(x_1), \dots, f(x_n)$ are related to each other. Existing approaches such as sparse Gaussian processes (see [Quinero-Candela and Rasmussen \(2005\)](#) for a review), resort to an *ex-post* approximation of fully-dependent Gaussian marginals with multivariate Gaussians exhibiting conditional independence structures. Unfortunately, these approximations trade-off accuracy for scalability through a control variable, namely the number of inducing points, whose choice is often left to the user. The approach we adopt in this paper consists of going back to stochastic analysis basics, and constructing stochastic processes whose finite-dimensional marginals exhibit suitable conditional independence structures so that we need not resorting to *ex-post* approximations. Incidentally, unlike sparse GP techniques, the conditional independence structures we introduce also allow for flexible and principled learning of local patterns, and this increased flexibility does not come at the expense of scalability.

The contributions of this paper are as follows. We introduce a novel class of stochastic processes, string Gaussian processes (*string GPs*), that may be used as priors over latent functions within a Bayesian nonparametric framework, especially for large scale problems and in the presence of possibly multiple types of local patterns. We propose a framework for analysing the flexibility of random functions and surfaces, and prove that our approach yields more flexible stochastic processes than isotropic Gaussian processes. We demonstrate that exact inference under a *string GP* prior scales considerably better than in the *standard GP paradigm*, and is amenable to distributed computing. We illustrate that popular stationary kernels can be well approximated within our framework, making *string GPs* a scalable alternative to commonly used GP models. We derive the joint law of a *string GP* and its gradient, thereby allowing for explanatory analysis on the learned latent function. We propose a reversible-jump Markov Chain Monte Carlo sampler for automatic learning of model complexity and local patterns from data.

Abstract

In this paper we introduce a novel framework for making exact nonparametric Bayesian inference on latent functions that is particularly suitable for *Big Data* tasks. Firstly, we introduce a class of stochastic processes we refer to as *string Gaussian processes (string GPs)* which are not to be mistaken for Gaussian processes operating on text). We construct *string GPs* so that their finite-dimensional marginals exhibit suitable *local* conditional independence structures, which allow for *scalable, distributed, and flexible* nonparametric Bayesian inference, without resorting to approximations, and while ensuring some mild global regularity constraints. Furthermore, *string GP* priors naturally cope with heterogeneous input data, and the gradient of the learned latent function is readily available for explanatory analysis. Secondly, we provide some theoretical results relating our approach to the *standard GP paradigm*. In particular, we prove that some *string GPs* are Gaussian processes, which provides a complementary *global* perspective on our framework. Finally, we derive a scalable and distributed MCMC scheme for supervised learning tasks under *string GP* priors. The proposed MCMC scheme has computational time complexity $\mathcal{O}(N)$ and memory requirement $\mathcal{O}(dN)$, where N is the data size and d the dimension of the input space. We illustrate the efficacy of the proposed approach on several synthetic and real-world data sets, including a data set with 6 millions input points and 8 attributes.

Keywords: String Gaussian processes, scalable Bayesian nonparametrics, Gaussian processes, nonstationary kernels, reversible-jump MCMC, point process priors

1. Introduction

Many problems in statistics and machine learning involve inferring a latent function from training data (for instance regression, classification, inverse reinforcement learning, inference on point processes to name but a few). Real-valued stochastic processes, among which Gaussian processes (GPs), are often used as functional priors for such problems, thereby allowing for a full Bayesian nonparametric treatment. In the machine learning community, interest in GPs grew out of the observation that some Bayesian neural networks converge to GPs as the number of hidden units approaches infinity ([Neal \(1996\)](#)). Since then, other similarities have been established between GPs and popular models such as Bayesian linear regression, Bayesian basis function regression, spline models and support vector machines ([Rasmussen and Williams \(2006\)](#)). However, they often perform poorly on *Big Data* tasks primarily for two reasons. Firstly, large data sets are likely to exhibit multiple types of local patterns that should appropriately be accounted for by flexible and possibly

The rest of the paper is structured as follows. In Section 2 we review recent advances on Gaussian processes in relation to inference on large data sets. In Section 3 we formally construct *string* GPs and derive some important results. In Section 4 we provide detailed illustrative and theoretical comparisons between *string* GPs and the *standard GP paradigm*. In Section 5 we propose methods for inferring latent functions under *string* GP priors with time complexity and memory requirement that are linear in the size of the data set. The efficacy of our approach compared to competing alternatives is illustrated in Section 6. Finally, we finish with a discussion in Section 7.

2. Related Work

The two primary drawbacks of the *standard GP paradigm* on large scale problems are the lack of scalability resulting from postulating a full multivariate Gaussian prior on function values at *all* training inputs, and the difficulty postulating *a priori* a class of covariance functions capable of capturing intricate and often local patterns likely to occur in large data sets. A tremendous amount of work has been published that attempt to address either of the aforementioned limitations. However, scalability is often achieved either through approximations or for specific applications, and nonstationarity is usually introduced at the expense of scalability, again for specific applications.

2.1 Scalability Through Structured Approximations

As far as scalability is concerned, sparse GP methods have been developed that approximate the multivariate Gaussian probability density function (pdf) over training data with the marginal over a smaller set of inducing points multiplied by an approximate conditional pdf (Smola and Bartlett (2001); Lawrence et al. (2003); Seeger (2003b,a); Snelson and Ghahramani (2006)). This approximation yields a time complexity linear—rather than cubic—in the data size and squared in the number of inducing points. We refer to Quinero-Candela and Rasmussen (2005) for a review of sparse GP approximations. More recently, Hensman et al. (2013, 2015) combined sparse GP methods with Stochastic Variational Inference (Hoffman et al. (2013)) for GP regression and GP classification. However, none of these sparse GP methods addresses the selection of the number of inducing points (and the size of the minibatch in the case of Hensman et al. (2013, 2015)), although this may greatly affect scalability. More importantly, although these methods do not impose strong restrictions on the covariance function of the GP model to approximate, they do not address the need for flexible covariance functions inherent to large scale problems, which are more likely to exhibit intricate and local patterns, and applications considered by the authors typically use the vanilla squared exponential kernel.

Lazaro-Gredilla et al. (2010) proposed approximating stationary kernels with truncated Fourier series in Gaussian process regression. An interpretation of the resulting sparse spectrum Gaussian process model as Bayesian basis function regression with a finite number K of trigonometric basis functions allows making inference in time complexity and memory requirement that are both linear in the size of the training sample. However, this model has two major drawbacks. Firstly, it is prone to over-fitting. In effect, the learning machine will aim at inferring the K major spectral frequencies evidenced in the training data. This will only lead to appropriate prediction out-of-sample when the underlying latent phenomenon can be appropriately characterised by a finite discrete spectral decomposition that is expected to be the same everywhere on the domain. Secondly, this model implicitly postulates that the covariance between the values of the GP at two points does not vanish as the distance between the points becomes arbitrarily large. This imposes *a priori* the

view that the underlying function is highly structured, which might be unrealistic in many real-life non-periodic applications. This approach is generalised by the so-called *random Fourier features* methods (Rahimi and Recht (2007); Le et al. (2013); Yang et al. (2015)). Unfortunately all existing random Fourier features methods give rise to stationary covariance functions, which might not be appropriate for data sets exhibiting local patterns.

The bottleneck of inference in the *standard GP paradigm* remains inverting and computing the determinant of a covariance matrix, normally achieved through the Cholesky decomposition or Singular Value Decomposition. Methods have been developed that speed-up these decompositions through low rank approximations (Williams and Seeger (2001)) or by exploiting specific structures in the covariance function and in the input data (Satchi (2011); Wilson et al. (2014)), which typically give rise to Kronecker or Toeplitz covariance matrices. While the Kronecker method used by Satchi (2011) and Wilson et al. (2014) is restricted to inputs that form a Cartesian grid and to separable kernels,¹ low rank approximations such as the Nyström method used by Williams and Seeger (2001) modify the covariance function and hence the functional prior in a non-trivial way. Methods have also been proposed to interpolate the covariance matrix on a uniform or Cartesian grid in order to benefit from some of the computational gains of Toeplitz and Kronecker techniques even when the input space is not structured (Wilson and Nickisch (2015)). However, none of these solutions is general as they require that either the covariance function be separable (Kronecker techniques), or the covariance function be stationary and the input space be one-dimensional (Toeplitz techniques).

2.2 Scalability Through Data Distribution

A family of methods have been proposed to scale-up inference in GP models that are based on the observation that it is more computationally efficient to compute the pdf of K independent small Gaussian vectors with size n than to compute the pdf of a single bigger Gaussian vector of size nK . For instance, Kim et al. (2005) and Gramacy and Lee (2008) partitioned the input space, and put independent stationary GP priors on the restrictions of the latent function to the subdomains forming the partition, which can be regarded as independent *local GP experts*. Kim et al. (2005) partitioned the domain using Voronoi tessellations, while Gramacy and Lee (2008) used tree based partitioning. These two approaches are provably equivalent to postulating a (nonstationary) GP prior on the whole domain that is discontinuous along the boundaries of the partition, which might not be desirable if the latent function we would like to infer is continuous, and might affect predictive accuracy. The more local experts there are, the more scalable the model will be, but the more discontinuities the latent function will have, and subsequently the less accurate the approach will be.

Mixtures of Gaussian process experts models (MoE) (Tresp (2001); Rasmussen and Ghahramani (2001); Meeds and Osindero (2006); Ross and Dy (2013)) provide another implementation of this idea. MoE models assume that there are multiple latent functions to be inferred from the data, on which it is placed independent GP priors, and each training input is associated to one latent function. The number of latent functions and the repartition of data between latent functions can then be performed in a full Bayesian nonparametric fashion (Rasmussen and Ghahramani (2001); Ross and Dy (2013)). When there is a single continuous latent function to be inferred, as it is the case for most regression models, the foregoing Bayesian nonparametric approach will learn a single latent function, thereby leading to a time complexity and a memory requirement that are the same as in the *standard GP paradigm*, which defies the scalability argument.

¹. That is multivariate kernel that can be written as product of univariate kernels.

The last implementation of the idea in this section consists of distributing the training data over multiple independent but identical GP models. In regression problems, examples include the *Bayesian Committee Machines* (BCM) of [Tresp \(2000\)](#), the *generalized product of experts* (gPoE) model of [Cao and Fleet \(2014\)](#), and the *robust Bayesian Committee Machines* (rBCM) of [Deisenroth and Ng \(2015\)](#). These models propose splitting the training data in small subsets, each subset being assigned to a different GP regression model—referred to as an expert—that has the same hyper-parameters as the other experts, although experts are assumed to be mutually independent. Training is performed by maximum marginal likelihood, with time complexity (resp. memory requirement) linear in the number of experts and cubic (resp. squared) in the size of the largest data set processed by an expert. Predictions are then obtained by aggregating the predictions of all GP experts in a manner that is specific to the method used (that is the BCM, the gPoE or the rBCM). However, these methods present major drawbacks in the training and testing procedures. In effect, the assumption that experts have identical hyper-parameters is inappropriate for data sets exhibiting local patterns. Even if one would allow GP experts to be driven by different hyper-parameters as in [Nguyen and Bonilla \(2014\)](#) for instance, learned hyper-parameters would lead to overly simplistic GP experts and poor aggregated predictions when the number of training inputs assigned to each expert is small—this is a direct consequence of the (desirable) fact that maximum marginal likelihood GP regression abides by Occam’s razor. Another critical pitfall of BCM, gPoE and rBCM is that their methods for aggregating expert predictions are Kolmogorov *inconsistent*. For instance, denoting \hat{p} the predictive distribution in the BCM, it can be easily seen from Equations (2.4) and (2.5) in [Tresp \(2000\)](#) that the predictive distribution $\hat{p}(f(x_1^*)|\mathcal{D})$ (resp. $\hat{p}(f(x_2^*)|\mathcal{D})$)² provided by the aggregation procedure of the BCM is *not* the marginal over $f(x_2^*)$ (resp. over $f(x_1^*)$) of the multivariate predictive distribution $\hat{p}(f(x_1^*), f(x_2^*)|\mathcal{D})$ obtained from experts multivariate predictions $p_k(f(x_1^*), f(x_2^*)|\mathcal{D})$ using the same aggregation procedure: $\hat{p}(f(x_1^*)|\mathcal{D}) \neq \int \hat{p}(f(x_1^*), f(x_2^*)|\mathcal{D}) df(x_2^*)$. Without Kolmogorov consistency, it is impossible to make principled Bayesian inference of latent function values. A principled Bayesian nonparametric model should not provide predictions about $f(x_1^*)$ that differ depending on whether or not one is also interested in predicting other values $f(x_2^*)$ simultaneously. This pitfall might be the reason why [Cao and Fleet \(2014\)](#) and [Deisenroth and Ng \(2015\)](#) restricted their expositions to predictive distributions about a single function value at a time $\hat{p}(f(x^*)|\mathcal{D})$, although their procedures (Equation 4 in [Cao and Fleet \(2014\)](#) and Equation 20 in [Deisenroth and Ng \(2015\)](#)) are easily extended to posterior distributions over multiple function values. These extensions would also be Kolmogorov *inconsistent*, and restricting the predictions to be of exactly one function value is unsatisfactory as it does not allow determining the posterior covariance between function values at two test inputs.

2.3 Expressive Stationary Kernels

In regards to flexibly handling complex patterns likely to occur in large data sets, [Wilson and Adams \(2013\)](#) introduced a class of expressive stationary kernels obtained by summing up convolutions of Gaussian basis functions with Dirac delta functions in the spectral domain. The sparse spectrum kernel can be thought of as the special case where the convolving Gaussian is degenerate. Although such kernels perform particularly well in the presence of globally repeated patterns in the data, their stationarity limits their utility on data sets with local patterns. Moreover the proposed covariance

2. Here f is the latent function to be inferred, x_1^*, x_2^* are test points and \mathcal{D} denotes training data.

functions generate infinitely differentiable random functions, which might be too restrictive in some applications.

2.4 Application-Specific Nonstationary Kernels

As for nonstationary kernels, [Paciorek and Schervish \(2004\)](#) proposed a method for constructing nonstationary covariance functions from any stationary one that involves introducing n input dependent $d \times d$ covariance matrices that will be inferred from the data. [Plagemann et al. \(2008\)](#) proposed a faster approximation to the model of [Paciorek and Schervish \(2004\)](#). However, both approaches scale poorly with the input dimension and the data size as they have time complexity $\mathcal{O}(\max(n d^3, n^3))$. [MacKay \(1998\)](#), [Schmidt and O’Hagan \(2003\)](#), and [Calandra et al. \(2014\)](#) proposed kernels that can be regarded as stationary after a non-linear transformation d on the input space: $k(x, x') = h(\|d(x) - d(x')\|)$, where h is positive semi-definite. Although for a given deterministic function d the kernel k is nonstationary, [Schmidt and O’Hagan \(2003\)](#) put a GP prior on d with mean function $m(x) = x$ and covariance function invariant under translation, which unfortunately leads to a kernel that is (unconditionally) stationary, albeit more flexible than $h(\|x - x'\|)$. To model nonstationarity, [Adams and Stegle \(2008\)](#) introduced a functional prior of the form $g(x) = f(x) \exp g(x)$ where f is a stationary GP and g is some scaling function on the domain. For a given non-constant function g such a prior indeed yields a nonstationary Gaussian process. However, when a stationary GP prior is placed on the function g as [Adams and Stegle \(2008\)](#) did, the resulting functional prior $g(x) = f(x) \exp g(x)$ becomes stationary. The piecewise GP ([Kim et al. \(2005\)](#)) and treed GP ([Gramacy and Lee \(2008\)](#)) models previously discussed also introduce nonstationarity. The authors’ premise is that heterogeneous patterns might be locally homogeneous. However, as previously discussed such models are inappropriate for modelling continuous latent functions.

2.5 Our Approach

The approach we propose in this paper for inferring latent functions in large scale problems, possibly exhibiting locally homogeneous patterns, consists of constructing a novel class of *smooth, nonstationary* and *flexible* stochastic processes we refer to as *string Gaussian processes* (*string GPs*), whose finite dimensional marginals are structured enough so that full Bayesian nonparametric inference scales linearly with the sample size, without resorting to approximations. Our approach is analogous to MoE models in that, when the input space is one-dimensional, a *string GP* can be regarded as a *collaboration of local GP experts* on non-overlapping supports, that implicitly exchange messages with one another, and that are independent conditional on the aforementioned messages. Each local GP expert only shares just enough information with adjacent local GP experts for the whole stochastic process to be sufficiently smooth (for instance continuously differentiable), which is an important improvement over MoE models as the latter generate discontinuous latent functions. These messages will take the form of boundary conditions, conditional on which each local GP expert will be independent from any other local GP expert. Crucially, unlike the BCM, the gPoE and the rBCM, we do not assume that local GP experts share the same prior structure (that is mean function, covariance function, or hyper-parameters). This allows each local GP expert to flexibly learn local patterns from the data if there are any, while preserving global smoothness, which will result in improved accuracy. Similarly to MoEs, the computational gain in our approach stems from the fact that the conditional independence of the local GP experts conditional on shared boundary

conditions will enable us to write the joint distribution over function and derivative values at a large number of inputs as the product of pdfs of much smaller Gaussian vectors. The resulting effect on time complexity is a decrease from $\mathcal{O}(N^3)$ to $\mathcal{O}(\max_k n_k^3)$, where $N = \sum_k n_k$, $n_k \ll N$. In fact, in Section 5 we will propose Reversible-Jump Monte Carlo Markov Chain (RJ-MCMC) inference methods that achieve memory requirement and time complexity $\mathcal{O}(N)$, without any loss of flexibility. All these results are preserved by our extension of *string GPs* to multivariate input spaces, which we will occasionally refer to as *membrane Gaussian processes* (or membrane GPs). Unlike the BCM, the gPbE and the rBCM, the approach we propose in this paper, which we will refer to as the *string GP paradigm*, is Kolmogorov consistent, and enables principled inference of the posterior distribution over the values of the latent function at multiple test inputs.

3. Construction of String and Membrane Gaussian Processes

In this section we formally construct *string* Gaussian processes, and we provide some important theoretical results including smoothness, and the joint law of *string GPs* and their gradients. We construct *string GPs* indexed on \mathbb{R} , before generalising to *string GPs* indexed on \mathbb{R}^d , which we will occasionally refer to as *membrane GPs* to stress that the input space is multivariate. We start by considering the joint law of a differentiable GP on an interval and its derivative, and introducing some related notions that we will use in the construction of *string GPs*.

Proposition 1 (Derivative Gaussian processes)

Let I be an interval, $k : I \times I \rightarrow \mathbb{R}$ a \mathcal{C}^2 symmetric positive semi-definite function,³ $m : I \rightarrow \mathbb{R}$ a \mathcal{C}^1 function.

(A) There exists a \mathbb{R}^2 -valued stochastic process $(D_t)_{t \in I}$, $D_t = (z_t, z'_t)$, such that for all $t_1, \dots, t_n \in I$, $(z_{t_1}, \dots, z_{t_n}, z'_{t_1}, \dots, z'_{t_n})$ is a Gaussian vector with mean $(m(t_1), \dots, m(t_n), \frac{dm}{dt}(t_1), \dots, \frac{dm}{dt}(t_n))$ and covariance matrix such that

$$\text{cov}(z_{t_1}, z_{t_2}) = k(t_1, t_2), \quad \text{cov}(z_{t_1}, z'_{t_2}) = \frac{\partial k}{\partial y}(t_1, t_2), \quad \text{and} \quad \text{cov}(z'_{t_1}, z'_{t_2}) = \frac{\partial^2 k}{\partial x^2 \partial y^2}(t_1, t_2),$$

where $\frac{\partial}{\partial x}$ (resp. $\frac{\partial}{\partial y}$) refers to the partial derivative with respect to the first (resp. second) variable of k . We herein refer to $(D_t)_{t \in I}$ as a **derivative Gaussian process**.

(B) $(z_t)_{t \in I}$ is a Gaussian process with mean function m , covariance function k and that is \mathcal{C}^1 in the L^2 (mean square) sense.

(C) $(z'_t)_{t \in I}$ is a Gaussian process with mean function $\frac{dm}{dt}$ and covariance function $\frac{\partial^2 k}{\partial x^2 \partial y^2}$. Moreover, $(z'_t)_{t \in I}$ is the L^2 derivative of the process $(z_t)_{t \in I}$.

Proof Although this result is known in the Gaussian process community, we provide a proof for the curious reader in Appendix B. ■

We will say of a kernel k that it is **degenerate at a** when a *derivative Gaussian process* $(z_t, z'_t)_{t \in I}$

³. \mathcal{C}^1 (resp. \mathcal{C}^2) functions denote functions that are once (resp. twice) continuously differentiable on their domains.

with kernel k is such that z_a and z'_a are perfectly correlated,⁴ that is

$$|\text{corr}(z_a, z'_a)| = 1.$$

As an example, the linear kernel $k(u, v) = \sigma^2(u - c)(v - c)$ is degenerate at 0. Moreover, we will say of a kernel k that it is **degenerate at b given a** when it is not degenerate at a and when the *derivative Gaussian process* $(z_t, z'_t)_{t \in I}$ with kernel k is such that the variances of z_b and z'_b conditional on (z_a, z'_a) are both zero.⁵ For instance, the periodic kernel proposed by MacKay (1998) with period T is degenerate at $u + T$ given u .

An important subclass of *derivative Gaussian processes* in our construction are the processes resulting from conditioning paths of a *derivative Gaussian process* to take specific values at certain times (t_1, \dots, t_c) . We herein refer to those processes as **conditional derivative Gaussian process**. As an illustration, when k is \mathcal{C}^3 on $I \times I$ with $I = [a, b]$, and neither degenerate at a nor degenerate at b given a , the *conditional derivative Gaussian process* on $I = [a, b]$ with unconditional mean function m and unconditional covariance function k that is conditioned to start at $(\tilde{z}_a, \tilde{z}'_a)$ is the *derivative Gaussian process* with mean function

$$\forall t \in I, \quad m_c^a(t; \tilde{z}_a, \tilde{z}'_a) = m(t) + \tilde{\mathbf{K}}_{t;a} \mathbf{K}_{a;a}^{-1} \begin{bmatrix} \tilde{z}_a - m(a) \\ \tilde{z}'_a - \frac{dm}{dt}(a) \end{bmatrix}, \quad (1)$$

and covariance function k_c^a that reads

$$\forall t, s \in I, \quad k_c^a(t, s) = k(t, s) - \tilde{\mathbf{K}}_{t;a} \mathbf{K}_{a;a}^{-1} \tilde{\mathbf{K}}_{s;a}^T \quad (2)$$

where $\mathbf{K}_{a;a} = \begin{bmatrix} k(a, a) & \frac{\partial k}{\partial x}(a, a) \\ \frac{\partial k}{\partial x}(a, a) & \frac{\partial^2 k}{\partial x^2 \partial y^2}(a, a) \end{bmatrix}$, and $\tilde{\mathbf{K}}_{t;a} = \begin{bmatrix} k(t, a) & \frac{\partial k}{\partial y}(t, a) \end{bmatrix}$. Similarly, when the process is conditioned to start at $(\tilde{z}_a, \tilde{z}'_a)$ and to end at $(\tilde{z}_b, \tilde{z}'_b)$, the mean function reads

$$\forall t \in I, \quad m_c^{a,b}(t; \tilde{z}_a, \tilde{z}'_a, \tilde{z}_b, \tilde{z}'_b) = m(t) + \tilde{\mathbf{K}}_{t;(a,b)} \mathbf{K}_{(a,b);(a,b)}^{-1} \begin{bmatrix} \tilde{z}_a - m(a) \\ \tilde{z}'_a - \frac{dm}{dt}(a) \\ \tilde{z}_b - m(b) \\ \tilde{z}'_b - \frac{dm}{dt}(b) \end{bmatrix}, \quad (3)$$

and the covariance function $k_c^{a,b}$ reads

$$\forall t, s \in I, \quad k_c^{a,b}(t, s) = k(t, s) - \tilde{\mathbf{K}}_{t;(a,b)} \mathbf{K}_{(a,b);(a,b)}^{-1} \tilde{\mathbf{K}}_{s;(a,b)}^T \quad (4)$$

where $\mathbf{K}_{(a,b);(a,b)} = \begin{bmatrix} \mathbf{K}_{a;a} & \mathbf{K}_{a;b} \\ \mathbf{K}_{b;a} & \mathbf{K}_{b;b} \end{bmatrix}$, and $\tilde{\mathbf{K}}_{t;(a,b)} = [\tilde{\mathbf{K}}_{t;a} \quad \tilde{\mathbf{K}}_{t;b}]$. It is important to note that both $\mathbf{K}_{a;a}$ and $\mathbf{K}_{(a,b);(a,b)}$ are indeed invertible because the kernel is assumed to be neither degenerate at a nor degenerate at b given a . Hence, the support of (z_a, z'_a, z_b, z'_b) is \mathbb{R}^4 , and any function and derivative values can be used for conditioning. Figure 1 illustrates example independent draws from a *conditional derivative Gaussian process*.

⁴. Or equivalently when the Gaussian vector (z_a, z'_a) is degenerate.

⁵. Or equivalently when the Gaussian vector (z_a, z'_a) is not degenerate but (z_a, z'_a, z_b, z'_b) is.

3.1 String Gaussian Processes on \mathbb{R}

The intuition behind string Gaussian processes on an interval comes from the analogy of collaborative local GP experts we refer to as *strings* that are connected but independent of each other conditional on some regularity boundary conditions. While each string is tasked with representing local patterns in the data, a string only shares the *states* of its extremities (value and derivative) with adjacent strings. Our aim is to preserve global smoothness and limit the amount of information shared between strings, thus reducing computational complexity. Furthermore, the conditional independence between strings will allow for distributed inference, greater flexibility and principled nonstationarity construction.

The following theorem at the core of our framework establishes that it is possible to connect together GPs on a partition of an interval I , in a manner consistent enough that the newly constructed stochastic object will be a stochastic process on I and in a manner restrictive enough that any two connected GPs will share just enough information to ensure that the constructed stochastic process is continuously differentiable (\mathcal{C}^1) on I in the L^2 sense.

Theorem 2 (String Gaussian process)

Let $a_0 < \dots < a_k < \dots < a_K, I = [a_0, a_K]$ and let $p_{\mathcal{N}}(x; \mu, \Sigma)$ be the multivariate Gaussian density with mean vector μ and covariance matrix Σ . Furthermore, let $(m_k : [a_{k-1}, a_k] \rightarrow \mathbb{R})_{k \in [1..K]}$ be \mathcal{C}^1 functions, and $(k_k : [a_{k-1}, a_k] \times [a_{k-1}, a_k] \rightarrow \mathbb{R})_{k \in [1..K]}$ be \mathcal{C}^3 symmetric positive semi-definite functions, neither degenerate at a_{k-1} , nor degenerate at a_k , given a_{k-1} .

(A) There exists an \mathbb{R}^2 -valued stochastic process $(SD_t)_{t \in I}$, $SD_t = (z_t, \dot{z}_t)$ satisfying the following conditions:

1) The probability density of $(SD_{a_0}, \dots, SD_{a_K})$ reads:

$$p_k(x_0, \dots, x_K) := \prod_{k=0}^K p_{\mathcal{N}}\left(x_k; \mu_k, \Sigma_k\right) \quad (5)$$

where: $\Sigma_0 = {}_1\mathbf{K}_{a_0; a_0}$, $\forall k > 0$ $\Sigma_k^b = {}_k\mathbf{K}_{a_k; a_k} - {}_k\mathbf{K}_{a_k; a_{k-1}} {}_k\mathbf{K}_{a_{k-1}; a_{k-1}}^{-1} {}_k\mathbf{K}_{a_{k-1}; a_k}^T$, (6)

$\mu_0^b = {}_1\mathbf{M}_{a_0}$, $\forall k > 0$ $\mu_k^b = {}_k\mathbf{M}_{a_k} + {}_k\mathbf{K}_{a_k; a_{k-1}} {}_k\mathbf{K}_{a_{k-1}; a_{k-1}}^{-1} (x_{k-1} - {}_k\mathbf{M}_{a_{k-1}})$, (7)

$$\text{with } {}_k\mathbf{K}_{a;v} = \begin{bmatrix} k_k(u, v) & \frac{\partial k_k}{\partial v}(u, v) \\ \frac{\partial k_k}{\partial x}(u, v) & \frac{\partial^2 k_k}{\partial x \partial y}(u, v) \end{bmatrix}, \text{ and } {}_k\mathbf{M}_u = \begin{bmatrix} m_k(u) \\ \frac{dm_k}{dt}(u) \end{bmatrix}.$$

2) Conditional on $(SD_{a_k} = x_k, k \in [0..K])$, the restrictions $(SD_t)_{t \in [a_{k-1}, a_k]}$, $k \in [1..K]$ are independent conditional derivative Gaussian processes, respectively with unconditional mean function m_k and unconditional covariance function k_k and that are conditioned to take values x_{k-1} and x_k at a_{k-1} and a_k respectively. We refer to $(SD_t)_{t \in I}$ as a **string derivative Gaussian process**, and to its first coordinate $(z_t)_{t \in I}$ as a **string Gaussian process** namely,

$$(z_t)_{t \in I} \sim \text{SGP}(\{a_k\}, \{m_k\}, \{k_k\}).$$

(B) The **string Gaussian process** $(z_t)_{t \in I}$ defined in (A) is \mathcal{C}^1 in the L^2 sense and its L^2 derivative is the process $(\dot{z}_t)_{t \in I}$ defined in (A).

Proof See Appendix C. ■

In our collaborative local GP experts analogy, Theorem 2 stipulates that each local expert takes as message from the previous expert its left hand side boundary conditions, conditional on which it generates its right hand side boundary conditions, which it then passes on to the next expert. Conditional on their boundary conditions local experts are independent of each other, and resemble vibrating pieces of string on fixed extremities, hence the name *string Gaussian process*.

3.2 Pathwise Regularity

Thus far we have dealt with regularity only in the L^2 sense. However, we note that a sufficient condition for the process $(z_t)_{t \in I}$ in Theorem 2 to be almost surely continuous (i.e. sample paths are continuous with probability 1) and to be the almost sure derivative of the string Gaussian process $(z_t)_{t \in I}$, is that the Gaussian processes on $I_k = [a_{k-1}, a_k]$ with mean and covariance functions $m_{ok}^{a_{k-1}, a_k}$ and $k_{ok}^{a_{k-1}, a_k}$ (as per Equations 3 and 4 with $m := m_k$ and $k := k_k$) are themselves almost surely \mathcal{C}^1 for every boundary condition.⁶ We refer to (Adler and Taylor, 2011, Theorem 2.5.2) for a sufficient condition under which a \mathcal{C}^1 in L^2 Gaussian process is also almost surely \mathcal{C}^1 . As the above question is provably equivalent to that of the almost sure continuity of a Gaussian process (see Adler and Taylor, 2011, p. 30), Kolmogorov's continuity theorem (see Øksendal, 2003, Theorem 2.2.3) provides a more intuitive, albeit stronger, sufficient condition than that of (Adler and Taylor, 2011, Theorem 2.5.2).

3.3 Illustration

Algorithm 1 illustrates sampling jointly from a string Gaussian process and its derivative on an interval $I = [a_0, a_K]$. We start off by sampling the string boundary conditions (z_{a_k}, \dot{z}_{a_k}) sequentially, conditional on which we sample the values of the stochastic process on each string. This we may do in parallel as the strings are independent of each other conditional on boundary conditions. The resulting time complexity is the sum of $\mathcal{O}(\max_k n_k^2)$ for sampling values within strings, and $\mathcal{O}(n)$ for sampling boundary conditions, where the sample size is $n = \sum_k n_k$. The memory requirement grows as the sum of $\mathcal{O}(\sum_k n_k^2)$, required to store conditional covariance matrices of the values within strings, and $\mathcal{O}(K)$ corresponding to the storage of covariance matrices of boundary conditions. In the special case where strings are all empty, that is inputs and boundary times are the same, the resulting time complexity and memory requirement are $\mathcal{O}(n)$. Figure 2 illustrates a sample from a string Gaussian process, drawn using this approach.

3.4 String Gaussian Processes on \mathbb{R}^d

So far the input space has been assumed to be an interval. We generalise *string GPs* to hyperrectangles in \mathbb{R}^d as stochastic processes of the form:

$$f(t_1, \dots, t_d) = \phi\left(z_{t_1}^1, \dots, z_{t_d}^d\right), \quad (10)$$

where the *link function* $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$ is a \mathcal{C}^1 function and (z_t^i) are d independent (\perp) latent string Gaussian processes on intervals. We will occasionally refer to *string GPs* indexed on \mathbb{R}^d with $d > 1$ as *membrane GPs* to avoid any ambiguity. We note that when $d = 1$ and when the link function

⁶ The proof is provided in Appendix D.

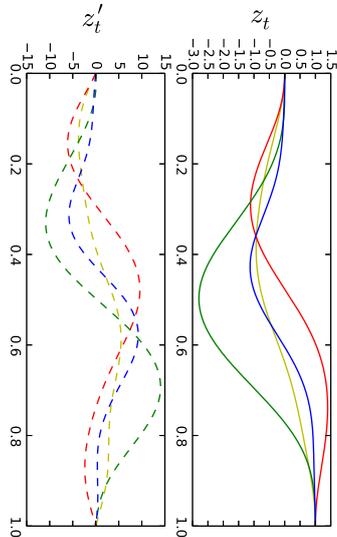


Figure 1: Draws from a conditional derivative GP conditioned to start at 0 with derivative 0 and to finish at 1.0 with derivative 0.0. The unconditional kernel is the squared exponential kernel with variance 1.0 and input scale 0.2.

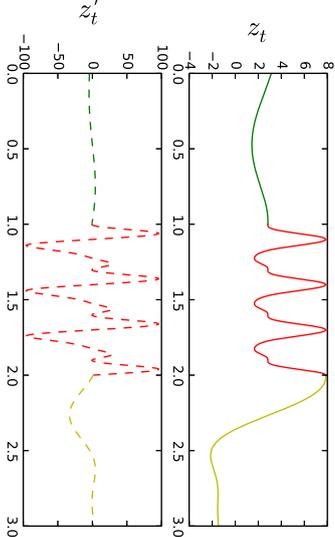


Figure 2: Draw from a *string GP* (z_t) with 3 strings and its derivative (z'_t), under squared exponential kernels (green and yellow strings), and the periodic kernel of MacKay (1998) (red string).

Algorithm 1 Simulation of a string derivative Gaussian process

Inputs: boundary times $a_0 < \dots < a_K$, string times $\{t_j^k \in [a_{k-1}, a_k] \mid j \in [1, n_k], k \in [1, K]\}$, unconditional mean (resp. covariance) functions m_k (resp. k_k)

Output: $\{\dots, z_{a_k}, z'_{a_k}, \dots, z_{t_j^k}, z'_{t_j^k}, \dots\}$.

Step 1: sample the boundary conditions sequentially.

for $k = 0$ to K do

Sample $(z_{a_k}, z'_{a_k}) \sim \mathcal{N}(\mu_k^b, \Sigma_k^b)$, with μ_k^b and Σ_k^b as per Equations (7) and (6).

end for

Step 2: sample the values on each string conditional on the boundary conditions in parallel.

parfor $k = 1$ to K do

Let ${}_k M_{a_k}$ and ${}_k K_{a_k; y}$ be as in Theorem 2.

$${}_k \Lambda = \begin{bmatrix} {}_k \mathbf{K}_{t_1^k; a_k-1} & \dots & {}_k \mathbf{K}_{t_1^k; a_k} \\ \dots & \dots & \dots \\ {}_k \mathbf{K}_{t_{n_k}^k; a_k-1} & \dots & {}_k \mathbf{K}_{t_{n_k}^k; a_k} \end{bmatrix}^{-1} \begin{bmatrix} {}_k \mathbf{K}_{a_{k-1}; a_k-1} & \dots & {}_k \mathbf{K}_{a_{k-1}; a_k} \\ \dots & \dots & \dots \\ {}_k \mathbf{K}_{a_k; a_k-1} & \dots & {}_k \mathbf{K}_{a_k; a_k} \end{bmatrix}^{-1},$$

$$\mu_k^s = \begin{bmatrix} {}_k \mathbf{M}_{t_1^k} \\ \dots \\ {}_k \mathbf{M}_{t_{n_k}^k} \end{bmatrix} + {}_k \Lambda \begin{bmatrix} z_{a_{k-1}}^{2a_{k-1}} - m_k(a_{k-1}) \\ z_{a_{k-1}}^{2a_k} - m_k(a_k) \\ z_{a_k}^{2a_k} - \frac{dm_k}{dt}(a_k) \end{bmatrix}, \quad (8)$$

$$\Sigma_k^s = \begin{bmatrix} {}_k \mathbf{K}_{t_1^k; t_1^k} & \dots & {}_k \mathbf{K}_{t_1^k; t_{n_k}^k} \\ \dots & \dots & \dots \\ {}_k \mathbf{K}_{t_{n_k}^k; t_1^k} & \dots & {}_k \mathbf{K}_{t_{n_k}^k; t_{n_k}^k} \end{bmatrix} - {}_k \Lambda \begin{bmatrix} {}_k \mathbf{K}_{a_{k-1}; a_k-1} & \dots & {}_k \mathbf{K}_{a_{k-1}; a_k} \\ \dots & \dots & \dots \\ {}_k \mathbf{K}_{a_k; a_k-1} & \dots & {}_k \mathbf{K}_{a_k; a_k} \end{bmatrix}^T. \quad (9)$$

Sample $(z_{t_1^k}, z'_{t_1^k}, \dots, z_{t_{n_k}^k}, z'_{t_{n_k}^k}) \sim \mathcal{N}(\mu_k^s, \Sigma_k^s)$.

end parfor

is $\phi(x) = x$, we recover *string GPs* indexed on an interval as previously defined. When the *string GPs* (z_t^j) are a.s. \mathcal{C}^1 , the *membrane GP* f in Equation (10) is also a.s. \mathcal{C}^1 , and the partial derivative with respect to the j -th coordinate reads:

$$\frac{\partial f}{\partial t_j}(t_1, \dots, t_d) = z_{t_j}^{j'} \frac{\partial \phi}{\partial t_j}(z_{t_1}^1, \dots, z_{t_d}^d). \quad (11)$$

Thus in high dimensions, *string GPs* easily allow an explanation of the sensitivity of the learned latent function to inputs.

3.5 Choice of Link Function

Our extension of *string GPs* to \mathbb{R}^d departs from the *standard GP paradigm* in that we did not postulate a covariance function on $\mathbb{R}^d \times \mathbb{R}^d$ directly. Doing so usually requires using a metric on \mathbb{R}^d , which is often problematic for heterogeneous input dimensions, as it introduces an arbitrary comparison between distances in each input dimension. This problem has been partially addressed by approaches such as Automatic Relevance Determination (ARD) kernels, that allow for a linear rescaling of input dimensions to be learned jointly with kernel hyper-parameters. However, inference under a *string GP* prior can be thought of as learning a coordinate system in which the latent function f resembles the link function ϕ through non-linear rescaling of input dimensions. In particular, when ϕ is symmetric, the learned univariate *string GPs* (being interchangeable in ϕ) implicitly aim at normalizing input data across dimensions, making *string GPs* naturally cope with heterogeneous data sets.

An important question arising from our extension is whether or not the link function ϕ needs to be learned to achieve a flexible functional prior. The flexibility of a *string GP* as a functional prior depends on both the *link function* and the covariance structures of the underlying *string GP* building blocks (z_t^j). To address the impact of the choice of ϕ on flexibility, we constrain the *string GP* building blocks by restricting them to be independent identically distributed *string GPs* with one string each (i.e. (z_t^j) are i.i.d Gaussian processes). Furthermore, we restrict ourselves to isotropic kernels as they provide a consistent basis for putting the same covariance structure in \mathbb{R} and \mathbb{R}^d . One question we might then ask, for a given *link function* ϕ_0 , is whether or not an isotropic GP indexed on \mathbb{R}^d with covariance function k yields more flexible random surfaces than the stationary *string GP* $f(t_1, \dots, t_d) = \phi_0(z_{t_1}^1, \dots, z_{t_d}^d)$, where (z_t^j) are stationary GPs indexed on \mathbb{R} with the same covariance function k . If we find a *link function* ϕ_0 generating more flexible random surfaces than isotropic GP counterparts it would suggest ϕ need not to be inferred in dimension $d > 1$ to be more flexible than any GP using one of the large number of commonly used isotropic kernels, among which squared exponential kernels, rational quadratic kernels, and Matérn kernels to name but a few.

Before discussing whether such a ϕ_0 exists, we need to introduce a rigorous meaning to ‘flexibility’. An intuitive qualitative definition of the flexibility of a stochastic process indexed on \mathbb{R}^d is the ease with which it can generate surfaces with varying shapes from one random sample to another independent one. We recall that the tangent hyperplane to a \mathcal{C}^1 surface $y = f(x) = 0$, $x \in \mathbb{R}^d$ at some point $x_0 = (t_1^0, \dots, t_d^0)$ has equation $\nabla f(x_0)^T (x - x_0) - (y - f(x_0)) = 0$ and admits as normal vector $(\frac{\partial f}{\partial t_1}(t_1^0), \dots, \frac{\partial f}{\partial t_d}(t_d^0), -1)$. As tangent hyperplanes approximate a surface locally, a first criterion of flexibility for a random surface $y = f(x) = 0$, $x \in \mathbb{R}^d$ is the proclivity of the (random) direction of its tangent hyperplane at any point x —and hence the proclivity of $\nabla f(x)$ —to vary.

This criterion alone, however, does not capture the difference between the local shapes of the random surface at two distinct points. A complementary second criterion of flexibility is the proclivity of the (random) directions of the tangent hyperplanes at any two distinct points $x_0, x_1 \in \mathbb{R}^d$ —and hence the proclivity of $\nabla f(x_0)$ and $\nabla f(x_1)$ —to be independent. The first criterion can be measured using the entropy of the gradient at a point, while the second criterion can be measured through the mutual information between the two gradients. The more flexible a stochastic process, the higher the entropy of its gradient at any point, and the lower the mutual information between its gradients at any two distinct points. This is formalised in the definition below.

Definition 3 (Flexibility of stochastic processes)

Let f and g be two real valued, almost surely \mathcal{C}^1 , stochastic processes indexed on \mathbb{R}^d , and whose gradients have a finite entropy everywhere (i.e. $\forall x, H(\nabla f(x)), H(\nabla g(x)) < \infty$). We say that f is more flexible than g if the following conditions are met:

- 1) $\forall x, H(\nabla f(x)) \geq H(\nabla g(x))$,
 - 2) $\forall x \neq y, I(\nabla f(x); \nabla f(y)) \leq I(\nabla g(x); \nabla g(y))$,
- where H is the entropy operator, and $I(X; Y) = H(X) + H(Y) - H(X, Y)$ stands for the mutual information between X and Y .

The following proposition establishes that the *link function* $\phi_s(x_1, \dots, x_d) = \sum_{i=j}^d x_j$ yields more flexible stationary *string GPs* than their isotropic GP counterparts, thereby providing a theoretical underpinning for not inferring ϕ .

Proposition 4 (Additively separable string GPs are flexible)

Let $k(x, y) := \rho(\|x - y\|_{L^2}^2)$ be a stationary covariance function generating a.s. \mathcal{C}^1 GP paths indexed on \mathbb{R}^d , $d > 0$, and ρ a function that is \mathcal{C}^2 on $]0, +\infty[$ and continuous at 0. Let $\phi_s(x_1, \dots, x_d) = \sum_{j=1}^d x_j$; let $(z_t^j)_{t \in \mathbb{R}, j \in \{1, \dots, d\}}$ be independent stationary Gaussian processes with mean 0 and covariance function k (where the L^2 norm is on \mathbb{R}), and let $f(t_1, \dots, t_d) = \phi_s(z_{t_1}^1, \dots, z_{t_d}^d)$ be the corresponding stationary *string GP*. Finally, let g be an isotropic Gaussian process indexed on \mathbb{R}^d with mean 0 and covariance function k (where the L^2 norm is on \mathbb{R}^d). Then:

- 1) $\forall x \in \mathbb{R}^d, I(\nabla f(x)) = H(\nabla g(x))$,
- 2) $\forall x \neq y \in \mathbb{R}^d, I(\nabla f(x); \nabla f(y)) \leq I(\nabla g(x); \nabla g(y))$.

■ See Appendix E.

Although the link function need not be inferred in a full nonparametric fashion to yield comparable if not better results than most isotropic kernels used in the *standard GP paradigm*, for some problems certain link functions might outperform others. In Section 4.2 we analyse a broad family of link functions, and argue that they extend successful anisotropic approaches such as the Automatic Relevance Determination (MacKay (1998)) and the additive kernels of Duvenaud et al. (2011). Moreover, in Section 5 we propose a scalable inference scheme applicable to any link function.

4. Comparison with the Standard GP Paradigm

We have already established that sampling *string GPs* scales better than sampling GPs under the *standard GP paradigm* and is amenable to distributed computing. We have also established that

stationary additively separable *string GPs* are more flexible than their isotropic counterparts in the *standard GP paradigm*. In this section, we provide further theoretical results relating the *string GP paradigm* to the *standard GP paradigm*. Firstly we establish that *string GPs* with link function $\phi_s(x_1, \dots, x_d) = \sum_{j=1}^d x_j$ are GPs. Secondly, we derive the global mean and covariance functions induced by the *string GP* construction for a variety of link functions. Thirdly, we provide a sense in which the *string GP paradigm* can be thought of as extending the *standard GP paradigm*. And finally, we show that the *string GP paradigm* may serve as a scalable approximation of commonly used stationary kernels.

4.1 Some String GPs are GPs

On one hand we note from Theorem 2 that the restriction of a *string GP* defined on an interval to the support of the first string—in other words the first local GP expert—is a Gaussian process. On the other hand, the messages passed on from one local GP expert to the next are not necessarily consistent with the unconditional law of the receiving local expert, so that overall a *string GP* defined on an interval, that is when looked at globally and unconditionally, might not be a Gaussian process. However, the following proposition establishes that some *string GPs* are indeed Gaussian processes.

Proposition 5 (Additively separable string GPs are GPs)

String Gaussian processes on \mathbb{R} are Gaussian processes. Moreover, string Gaussian processes on \mathbb{R}^d with link function $\phi_s(x_1, \dots, x_d) = \sum_{j=1}^d x_j$ are also Gaussian processes.

Proof. The intuition behind this proof lies in the fact that if X is a multivariate Gaussian, and if conditional on X , Y is a multivariate Gaussian, providing that the conditional mean of Y depends linearly on X and the conditional covariance matrix of Y does not depend on X , the vector (X, Y) is jointly Gaussian. This will indeed be the case for our collaboration of local GP experts as the boundary conditions picked up by an expert from the previous will not influence the conditional covariance structure of the expert (the conditional covariance structure depends only on the partition of the domain, not the values of the boundary conditions) and will affect the mean linearly. See Appendix H for the full proof. ■

The above result guarantees that commonly used closed form predictive equations under GP priors are still applicable under some *string GP* priors, providing the global mean and covariance functions, which we derive in the following section, are available. Proposition 5 also guarantees stability of the corresponding *string GPs* in the GP family under addition of independent Gaussian noise terms as in regression settings. Moreover, it follows from Proposition 5 that inference techniques developed for Gaussian processes can be readily used under *string GP* priors. In Section 5 we provide an additional MCMC scheme that exploits the conditional independence between strings to yield greater scalability and distributed inference.

4.2 String GP kernels and String GP Mean Functions

The approach we have adopted in the construction of *string GPs* and *membrane GPs* did not require explicitly postulating a global mean function or covariance function. In Appendix I we derive the global mean and covariance functions that result from our construction. The global covariance function could be used for instance as a stand-alone kernel in any kernel method, for instance GP

models under the *standard GP paradigm*, which would provide a flexible and nonstationary alternative to commonly used kernels that may be used to learn local patterns in data sets—some successful example applications are provided in Section 5. That being said, adopting such a global approach should be limited to small scale problems as the conditional independence structure of *string GPs* does not easily translate into structures in covariance matrices over *string GP* values (without derivative information) that can be exploited to speed-up SVD or Cholesky decomposition. Crucially, marginalising out all derivative information in the distribution of *derivative string GP* values at some inputs would destroy any conditional independence structure, thereby limiting opportunities for scalable inference. In Section 5 we will provide a RJ-MCMC inference scheme that fully exploits the conditional independence structure in *string GPs* and scales to very large data sets.

4.3 Connection Between Multivariate String GP kernels and Existing Approaches

We recall that for $n \leq d$, the n -th order *elementary symmetric polynomial* (Macdonald (1995)) is given by

$$e_0(x_1, \dots, x_d) := 1, \quad \forall 1 \leq n \leq d \quad e_n(x_1, \dots, x_d) = \sum_{1 \leq i_1 < i_2 < \dots < i_n \leq d} \prod_{k=1}^n x_{i_k}. \quad (12)$$

As an illustration,

$$\begin{aligned} e_1(x_1, \dots, x_d) &= \sum_{j=1}^d x_j = \phi_s(x_1, \dots, x_d), \\ e_2(x_1, \dots, x_d) &= x_1 x_2 + x_1 x_3 + \dots + x_1 x_d + \dots + x_{d-1} x_d, \\ &\dots \\ e_d(x_1, \dots, x_d) &= \prod_{j=1}^d x_j = \phi_p(x_1, \dots, x_d). \end{aligned}$$

Covariance kernels of *string GPs*, using as link functions *elementary symmetric polynomials* e_n , extend most popular approaches that combine unidimensional kernels over features for greater flexibility or cheaper design experiments.

The first-order polynomial e_1 gives rise to additively separable Gaussian processes, that can be regarded as Bayesian nonparametric *generalised additive models* (GAM), particularly popular for their interpretability. Moreover, as noted by Durrande et al. (2012), additively separable Gaussian processes are considerably cheaper than alternate transformations in design experiments with high-dimensional input spaces. In addition to the above, additively separable *string GPs* also allow postulating the existence of local properties in the experimental design process at no extra cost.

The d -th order polynomial e_d corresponds to a product of unidimensional kernels, also known as separable kernels. For instance, the popular squared exponential kernel is separable. Separable kernels have been successfully used on large scale inference problems where the inputs form a grid (Saatchi, 2011; Wilson et al., 2014), as they yield covariance matrices that are Kronecker products, leading to maximum likelihood inference in linear time complexity and with linear memory requirement. Separable kernels are often used in conjunction with the *automatic relevance determination* (ARD) model, to learn the relevance of features through global linear rescaling. However, ARD

kernels might be limited in that we might want the relevance of a feature to depend on its value. As an illustration, the market value of a watch can be expected to be a stronger indicator of its owner's wealth when it is in the top 1 percentile, than when it is in the bottom 1 percentile; the rationale being that possessing a luxurious watch is an indication that one can afford it, whereas possessing a cheap watch might be either an indication of lifestyle or an indication that one cannot afford a more expensive one. Separable *string GP* kernels extend ARD kernels, in that strings between input dimensions and within an input dimension may have unconditional kernels with different hyperparameters, and possibly different functional forms, thereby allowing for *automatic local relevance determination* (ALRD).

More generally, using as link function the n -th order elementary symmetric polynomial e_n corresponds to the n -th order interaction of the *additive kernels* of Duvenaud et al. (2011). We also note that the class of link functions $\phi(x_1, \dots, x_d) = \sum_{i=1}^d \sigma_i e_i(x_1, \dots, x_d)$ yield full *additive kernels*. Duvenaud et al. (2011) noted that such kernels are 'exceptionally well-suited' to learn non-local structures in data. *String GPs* complement *additive kernels* by allowing them to learn local structures as well.

4.4 String GPs as Extension of the Standard GP Paradigm

The following proposition provides a perspective from which *string GPs* may be considered as extending Gaussian processes on an interval.

Proposition 6 (Extension of the standard GP paradigm)

Let $K \in \mathbb{N}^*$, let $I = [a_0, a_K]$ and $I_k = [a_{k-1}, a_k]$ be intervals with $a_0 < \dots < a_K$. Furthermore, let $m : I \rightarrow \mathbb{R}$ be a C^1 function, m_k the restriction of m to I_k , $h : I \times I \rightarrow \mathbb{R}$ a C^3 symmetric positive semi-definite function, and h_k the restriction of h to $I_k \times I_k$. If

$$(z_t)_{t \in I} \sim \mathcal{SGP}(\{a_k\}, \{m_k\}, \{h_k\}),$$

then

$$\forall k \in [1..K], (z_t)_{t \in I_k} \sim \mathcal{GP}(m, h).$$

Proof. See Appendix F. ■

We refer to the case where unconditional string mean and kernel functions are restrictions of the same functions as in Proposition 6 as *uniform string GPs*. Although uniform *string GPs* are not guaranteed to be as much regular at boundary times as their counterparts in the *standard GP paradigm*, we would like to stress that they may well generate paths that are. In other words, the functional space induced by a uniform *string GP* on an interval extends the functional space of the GP with the same mean and covariance functions m and h taken globally and unconditionally on the whole interval as in the *standard GP paradigm*. This allows for (but does not enforce) less regularity at the boundary times. When *string GPs* are used as functional prior, the posterior mean can in fact have more regularity at the boundary times than the continuous differentiability enforced in the *string GP paradigm*, providing such regularity is evidenced in the data.

We note from Proposition 6 that when m is constant and h is stationary, the restriction of the uniform *string GP* $(z_t)_{t \in I}$ to any interval whose interior does not contain a boundary time, the largest of which being the intervals $[a_{k-1}, a_k]$, is a stationary GP. We refer to such cases as *partition stationary string GPs*.

4.5 Commonly Used Covariance Functions and their String GP Counterparts

Considering the superior scalability of the *string GP paradigm*, which we may anticipate from the scalability of sampling *string GPs*, and which we will confirm empirically in Section 5, a natural question that comes to mind is whether or not kernels commonly used in the *standard GP paradigm* can be well approximated by *string GP* kernels, so as to take advantage of the improved scalability of the *string GP paradigm*. We examine the distortions to commonly used covariance structures resulting from restricting strings to share only C^1 boundary conditions, and from increasing the number of strings.

Figure 3 compares some popular stationary kernels on $[0, 1] \times [0, 1]$ (first column) to their uniform *string GP* kernel counterparts with 2, 4, 8 and 16 strings of equal length. The popular kernels considered are the squared exponential kernel (SE), the rational quadratic kernel $k_{RQ}(u, v) = \left(1 + \frac{2(u-v)^2}{\alpha}\right)^{-\alpha}$ with $\alpha = 1$ (RQ 1) and $\alpha = 5$ (RQ 5), the Matérn 3/2 kernel (MA 3/2), and the Matérn 5/2 kernel (MA 5/2), each with output scale (variance) 1 and input scale 0.5. Firstly, we observe that each of the popular kernels considered coincides with its uniform *string GP* counterparts regardless of the number of strings, so long as the arguments of the covariance function are less than an input scale apart. Except for the Matérn 3/2, the loss of information induced by restricting strings to share only C^1 boundary conditions becomes noticeable when the arguments of the covariance function are more than 1.5 input scales apart, and the effect is amplified as the number of strings increases. As for the Matérn 3/2, no loss of information can be noticed, as further attests Table 1. In fact, this comes as no surprise given that stationary Matérn 3/2 GP are 1-Markov, that is the corresponding derivative Gaussian process is a Markov process so that the vector (z_t, \dot{z}_t) contains as much information as all *string GP* or derivative values prior to t (see Doob (1944)). Table 1 provides some statistics on the absolute errors between each of the popular kernels considered and uniform *string GP* counterparts.

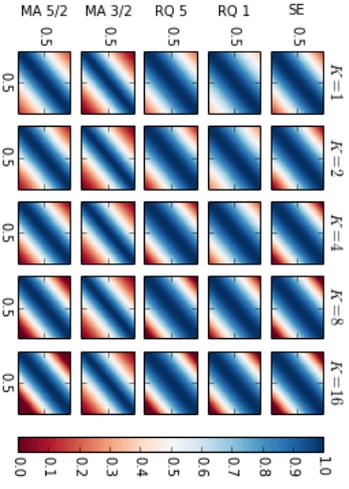


Figure 3: Commonly used covariance functions on $[0, 1] \times [0, 1]$ with the same input and output scales (first column) and their uniform *string GP* counterparts with $K > 1$ strings of equal length.

	$K = 2$			$K = 4$			$K = 8$			$K = 16$		
	min	avg	max	min	avg	max	min	avg	max	min	avg	max
SE	0	0.01	0.13	0	0.02	0.25	0	0.03	0.37	0	0.04	0.44
RQ 1	0	0.01	0.09	0	0.03	0.20	0	0.05	0.37	0	0.07	0.52
RQ 5	0	0.01	0.12	0	0.02	0.24	0	0.04	0.37	0	0.05	0.47
MA 3/2	0	0	0	0	0	0	0	0	0	0	0	0
MA 5/2	0	0.01	0.07	0	0.03	0.15	0	0.05	0.29	0	0.08	0.48

Table 1: Minimum, average, and maximum absolute errors between some commonly used stationary covariance functions on $[0, 1] \times [0, 1]$ (with unit variance and input scale 0.5) and their uniform *string GP* counterparts with $K > 1$ strings of equal length.

5. Inference under String and Membrane GP Priors

In this section we move on to developing inference techniques for Bayesian nonparametric inference of latent functions under *string GP* priors. We begin with marginal likelihood inference in regression problems. We then propose a novel reversible-jump MCMC sampler that enables automatic learning of model complexity (that is the number of different unconditional kernel configurations) from the data, with a time complexity and memory requirement both linear in the number of training inputs.

5.1 Maximum Marginal Likelihood for Small Scale Regression Problems

Firstly, we leverage the fact that additively separable *string GPs* are Gaussian processes to perform Bayesian nonparametric regressions in the presence of local patterns in the data, using standard Gaussian process techniques (see [Rasmussen and Williams, 2006](#), p.112 §5.4.1). We use as generative model

$$y_i = f(x_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma_{k_i}^2), \quad \sigma_{k_i}^2 > 0, \quad x_i \in I^1 \times \dots \times I^d, \quad y_i, \epsilon_i \in \mathbb{R}$$

we are given the training data set $\mathcal{D} = \{\tilde{x}_i, \tilde{y}_i\}_{i \in [1..N]}$, and we place a mean-zero additively separable *string GP* prior on f , namely

$$f(x) = \sum_{j=1}^d z_{x[j]}, \quad (z_i^j \sim \text{SGP}(\{a_k^j\}, \{0\}, \{k_k^j\})), \quad \forall j < l, \quad (z_i^j \perp (z_i^l))$$

which we assume to be independent of the measurement noise process. Moreover, the noise terms are assumed to be independent, and the noise variance $\sigma_{k_i}^2$ affecting $f(x_i)$ is assumed to be the same for any two inputs whose coordinates lie on the same string intervals. Such a heteroskedastic noise model fits nicely within the *string GP paradigm*, can be very useful when the dimension of the input space is small, and may be replaced by the typical constant noise variance assumption in high-dimensional input spaces.

Let us define $\mathbf{y} = (\tilde{y}_1, \dots, \tilde{y}_N)$, $\mathbf{X} = (\tilde{x}_1, \dots, \tilde{x}_N)$, $\mathbf{f} = (f(\tilde{x}_1), \dots, f(\tilde{x}_N))$ and let $\bar{\mathbf{K}}_{\mathbf{X};\mathbf{X}}$ denote the auto-covariance matrix of \mathbf{f} (which we have derived in [Section 4.2](#)), and let $\mathbf{D} = \text{diag}(\{\sigma_{k_i}^2\})$ denote the diagonal matrix of noise variances. It follows that \mathbf{y} is a Gaussian vector with mean $\mathbf{0}$ and auto-covariance matrix $\mathbf{K}_{\mathbf{y}} := \bar{\mathbf{K}}_{\mathbf{X};\mathbf{X}} + \mathbf{D}$ and that the log marginal likelihood reads:

$$\log p(\mathbf{y} | \mathbf{X}, \{\sigma_{k_i}\}, \{\theta_k^j\}, \{a_k^j\}) = -\frac{1}{2} \mathbf{y}^T \mathbf{K}_{\mathbf{y}}^{-1} \mathbf{y} - \frac{1}{2} \log \det(\mathbf{K}_{\mathbf{y}}) - \frac{n}{2} \log 2\pi. \quad (13)$$

We obtain estimates of the string measurement noise standard deviations $\{\hat{\sigma}_{k_i}\}$ and estimates of the string hyper-parameters $\{\hat{\theta}_k^j\}$ by maximising the marginal likelihood for a given domain partition $\{a_k^j\}$, using gradient-based methods. We deduce the predictive mean and covariance matrix of the latent function values \mathbf{f}^* at test points \mathbf{X}^* , from the estimates $\{\hat{\theta}_k^j\}, \{\hat{\sigma}_{k_i}\}$ as

$$\mathbf{E}(\mathbf{f}^* | \mathbf{y}) = \bar{\mathbf{K}}_{\mathbf{X}^*; \mathbf{X}} \mathbf{K}_{\mathbf{y}}^{-1} \mathbf{y} \quad \text{and} \quad \text{cov}(\mathbf{f}^* | \mathbf{y}) = \bar{\mathbf{K}}_{\mathbf{X}^*; \mathbf{X}^*} - \bar{\mathbf{K}}_{\mathbf{X}^*; \mathbf{X}} \mathbf{K}_{\mathbf{y}}^{-1} \bar{\mathbf{K}}_{\mathbf{X}; \mathbf{X}^*}, \quad (14)$$

using the fact that $(\mathbf{f}^*, \mathbf{y})$ is jointly Gaussian, and that the cross-covariance matrix between \mathbf{f}^* and \mathbf{y} is $\bar{\mathbf{K}}_{\mathbf{X}^*; \mathbf{X}}$ as the additive measurement noise is assumed to be independent from the latent process f .

5.1.1 REMARKS

The above analysis and equations still hold when a GP prior is placed on f with one of the multivariate *string GP* kernels derived in [Section 4.2](#) as covariance function.

It is also worth noting from the derivation of *string GP* kernels in [Appendix I](#) that the marginal likelihood [Equation \(13\)](#) is continuously differentiable in the locations of boundary times. Thus, for a given number of boundary times, the positions of the boundary times can be determined as part of the marginal likelihood maximisation. The derivatives of the marginal log-likelihood ([Equation 13](#)) with respect to the aforementioned locations $\{\sigma_{k_i}^2\}$ can be determined from the recursions of [Appendix I](#), or approximated numerically by finite differences. The number of boundary times in each input dimension can then be learned by trading off model fit (the maximum marginal log likelihood) and model simplicity (the number of boundary times or model parameters), for instance using information criteria such as AIC and BIC. When the input dimension is large, it might be advantageous to further constrain the hypothesis space of boundary times before using information criteria, for instance by assuming that the number of boundary times is the same in each dimension. An alternative Bayesian nonparametric approach to learning the number of boundary times will be discussed in [section 5.4](#).

This method of inference cannot exploit the structure of *string GPs* to speed-up inference, and as a result it scales like the *standard GP paradigm*. In fact, any attempt to marginalize out univariate derivative processes, including in the prior, will inevitably destroy the conditional independence structure. Another perspective to this observation is found by noting from the derivation of global *string GP* covariance functions in [Appendix I](#) that the conditional independence structure does not easily translate in a matrix structure that may be exploited to speed-up matrix inversion, and that marginalizing out terms relating to derivatives processes as in [Equation \(13\)](#) can only make things worse.

5.2 Generic Reversible-Jump MCMC Sampler for Large Scale Inference

More generally, we consider learning a smooth real-valued latent function f , defined on a d -dimensional hyper-rectangle, under a generative model with likelihood $p(\mathcal{D} | \mathbf{f}, \mathbf{u})$, where \mathbf{f} denotes values of f at training inputs points and \mathbf{u} denotes other likelihood parameters that are not related to f . A large class of machine learning problems aiming at inferring a latent function have a likelihood model of this form. Examples include celebrated applications such as nonparametric regression and nonparametric binary classification problems, but also more recent applications such as learning a profitable portfolio generating-function in *stochastic portfolio theory* ([Karatzas and Fernholz \(2009\)](#)) from the data. In particular, we do not assume that $p(\mathcal{D} | \mathbf{f}, \mathbf{u})$ factorizes over training inputs. Extensions to likelihood models that depend on the values of multiple latent functions are straight-forward and will be discussed in [Section 5.3](#).

5.2.1 PRIOR SPECIFICATION

We place a prior $p(\mathbf{u})$ on other likelihood parameters. For instance, in regression problems under a Gaussian noise model, \mathbf{u} can be the noise variance and we may choose $p(\mathbf{u})$ to be the inverse-Gamma distribution for conjugacy. We place a mean-zero *string GP* prior on f

$$f(x) = \phi \left(z_{x[1]}^1, \dots, z_{x[d]}^d \right), \quad (z_i^j \sim \text{SGP}(\{a_k^j\}, \{0\}, \{k_k^j\})), \quad \forall j < l, \quad (z_i^j \perp (z_i^l)). \quad (15)$$

As discussed in Section 3.5, the link function ϕ need not be inferred as the symmetric sum was found to yield a sufficiently flexible functional prior. Nonetheless, in this section we do not impose any restriction on the link function ϕ other than continuous differentiability. Denoting \mathbf{z} the vector of univariate *string GP* processes and their derivatives, evaluated at all distinct input coordinate values, we may re-parametrize the likelihood as $p(\mathcal{D}|\mathbf{z}, \mathbf{u})$, with the understanding that \mathbf{f} can be recovered from \mathbf{z} through the link function ϕ . To complete our prior specification, we need to discuss the choice of boundary times $\{a_k^j\}$ and the choice of the corresponding unconditional kernel structures $\{k_k^j\}$. Before doing so, we would like to stress that key requirements of our sampler are that i) it should decouple the need for scalability from the need for flexibility, ii) it should scale linearly with the number of training and test inputs, and iii) the user should be able to express prior views on model complexity/flexibility in an intuitive way, but the sampler should be able to validate or invalidate the prior model complexity from the data. While the motivations for the last two requirements are obvious, the first requirement is motivated by the fact that a massive data set may well be more homogeneous than a much smaller data set.

5.2.2 SCALABLE CHOICE OF BOUNDARY TIMES

To motivate our choice of boundary times that achieves great scalability, we first note that the evaluation of the likelihood, which will naturally be needed by the MCMC sampler, will typically have at least linear time complexity and linear memory requirement, as it will require performing computations that use each training sample at least once. Thus, the best we can hope to achieve overall is linear time complexity and linear memory requirement. Second, in MCMC schemes with functional priors, the time complexity and memory requirements for sampling from the posterior

$$p(\mathbf{f}|\mathcal{D}) \propto p(\mathcal{D}|\mathbf{f})p(\mathbf{f})$$

are often the same as the resource requirements for sampling from the prior $p(\mathbf{f})$, as evaluating the model likelihood is rarely the bottleneck. Finally, we note from Algorithm 1 that, when each input coordinate in each dimension is a boundary time, the sampling scheme has time complexity and memory requirement that are linear in the maximum number of unique input coordinates across dimensions, which is at most the number of training samples. In effect, each univariate derivative *string GP* is sampled in *parallel* at as many times as there are unique input coordinates in that dimension, before being combined through the link function. In a given input dimension, univariate derivative *string GP* values are sampled sequentially, one boundary time conditional on the previous. The foregoing sampling operation is very scalable not only asymptotically but also in absolute terms: it merely requires storing and inverting at most as many 2×2 matrices as the number of input points. We will evaluate the actual overall time complexity and memory requirement when we discuss our MCMC sampler in greater detail. For now, we would like to stress that i) choosing each distinct input coordinate value as a boundary time in the corresponding input dimension before training is a perfectly valid choice, ii) we expect this choice to result in resource requirements that grow linearly with the sample size and iii) in the *string GP* theory we have developed thus far there is no requirement that two adjacent strings be driven by different kernel hyper-parameters.

5.2.3 MODEL COMPLEXITY LEARNING AS A CHANGE-POINT PROBLEM

The remark iii) above pertains to model complexity. In the simplest case, all strings are driven by the same kernel and hyper-parameters as it was the case in Section 4.5, where we discussed how this

setup departs from postulating the unconditional string covariance function k_k^j globally similarly to the *standard GP paradigm*. The more distinct unconditional covariance structures there are, the more complex the model is, as it may account for more types of local patterns. Thus, we may identify model complexity to the number of different kernel configurations across input dimensions. In order to learn model complexity, we require that some (but not necessarily all) strings share their kernel configuration.⁷ Moreover, we require kernel membership to be dimension-specific in that two strings in different input dimensions may not explicitly share a kernel configuration in the prior specification, although the posterior distribution over their hyper-parameters might be similar if the data support it.

In each input dimension j , kernel membership is defined by a partition of the corresponding domain operated by a (possibly empty) set of change-points,⁸ as illustrated in Figure 4. When there is no change-point as in Figure 4-(a), all strings are driven by the same kernel and hyper-parameters. Each change-point c_b^j induces a new kernel configuration θ_b^j that is shared by all strings whose boundary times a_k^j and a_{k+1}^j both lie in $[c_b^j, c_{b+1}^j]$.⁹ When one or multiple change-points c_b^j occur between two adjacent boundary times as illustrated in Figures 4-(b-d), for instance $a_k^j \leq c_b^j \leq a_{k+1}^j$, the kernel configuration of the string defined on $[a_k^j, a_{k+1}^j]$ is that of the largest change-point that lies in $[a_k^j, a_{k+1}^j]$ (see for instance Figure 4-(d)). For consistency, we denote θ_b^j the kernel configuration driving the first string in the j -th dimension; it also drives strings that come before the first change-point, and all strings when there is no change-point.

To place a prior on model complexity, it suffices to define a joint probability measure on the set of change-points and the corresponding kernel configurations. As kernel configurations are not shared across input dimensions, we choose these priors to be independent across input dimensions. Moreover, $\{c_b^j\}$ being a random collection of points on an interval whose number and positions are both random, it is *de facto* a point process (Daley and Vere-Jones (2008)). To keep the prior specification of change-points uninformative, it is desirable that conditional on the number of change-points, the positions of change-points be i.i.d. uniform on the domain. As for the number of change-points, it is important that the support of its distribution not be bounded, so as to allow for an arbitrarily large model complexity if warranted. The two requirements above are satisfied by a homogeneous Poisson process or HPP (Daley and Vere-Jones (2008)) with constant intensity λ^j . More precisely, the prior probability measure on $(\{c_b^j, \theta_b^j\}, \lambda^j)$ is constructed as follows:

$$\begin{cases} \lambda^j \sim \Gamma(\alpha^j, \beta^j), \\ \{c_b^j\} | \lambda^j \sim \text{HPP}(\lambda^j) \\ \theta_b^j | \{c_b^j\}, \lambda^j \stackrel{\text{i.i.d.}}{\sim} \log \mathcal{N}(0, \rho^j) \\ \forall (j, p) \neq (1, q) \quad \theta_b^j \perp \theta_q^j \end{cases} \quad (16)$$

where we choose the Gamma distribution Γ as prior on the intensity λ^j for conjugacy, we assume all kernel hyper-parameters are positive as is often the case in practice,⁹ the coordinates of the hyper-parameters of a kernel configuration are assumed i.i.d., and kernel hyper-parameters are assumed

⁷ That is, the functional form of the unconditional kernel k_k^j and its hyper-parameters.

⁸ We would like to stress that change-points do not introduce new input points or boundary times, but solely define a partition of the domain of each input dimension.

⁹ This may easily be relaxed if needed, for instance by putting normal priors on parameters that may be negative and log-normal priors on positive parameters.

independent between kernel configurations. Denoting the domain of the j -th input $[a^j, b^j]$, it follows from applying the laws of total expectation and total variance on Equation (16) that the expected number of change-points in the j -th dimension under our prior is

$$E(\#\{c_p^j\}) = (b^j - a^j) \frac{\alpha^j}{\beta^j}, \tag{17}$$

and the variance of the number of change-points in the j -dimension under our prior is

$$\text{Var}(\#\{c_p^j\}) = (b^j - a^j) \frac{\alpha^j}{\beta^j} \left(1 + \frac{(b^j - a^j)}{\beta^j} \right). \tag{18}$$

The two equations above may guide the user when setting the parameters α^j and β^j . For instance, these values may be set so that the expected number of change-points in a given input dimension be a fixed fraction of the number of boundary times in that input dimension, and so that the prior variance over the number of change-points be large enough that overall the prior isn't too informative.

We could have taken a different approach to construct our prior on change-points. In effect, assuming for the sake of the argument that the boundaries of the domain of the j -th input, namely a^j and b^j , are the first and last change-point in that input dimension, we note that the mapping

$$(\dots, c_p^j, \dots) \rightarrow (\dots, p_p^j, \dots) := \left(\dots, \frac{c_{p+1}^j - c_p^j}{b^j - a^j}, \dots \right)$$

defines a bijection between the set of possible change-points in the j -th dimension and the set of all discrete probability distributions. Thus, we could have placed as prior on (\dots, p_p^j, \dots) a Dirichlet process (Ferguson (1973)), a Pitman-Yor process (Pitman and Yor (1997)), more generally *normalized completely random measures* (Kingman (1967)) or any other probability distribution over partitions. We prefer the point process approach primarily because it provides an easier way of expressing prior belief about model complexity through the expected number of change-points $\#\{c_p^j\}$, while remaining uninformative about positions thereof.

One might also be tempted to regard change-points in an input dimension j as inducing a partition, not of the domain $[a^j, b^j]$, but of the set of boundary times a_k^j in the same dimension, so that one may define a prior over kernel memberships through a prior over partitions of the set of boundary times. However, this approach would be inconsistent with the aim to learn local patterns in the data if the corresponding random measure is *exchangeable*. In effect, as boundary times are all input coordinates, local patterns may only arise in the data as a result of adjacent strings sharing kernel configurations. An exchangeable random measure would postulate a priori that two kernel membership assignments that have the same kernel configurations (i.e. the same number of configurations and the same set of hyper-parameters) and the same *number* of boundary times in each kernel cluster (although not exactly the same boundary times), are equally likely to occur, thereby possibly putting more probability mass on kernel membership assignments that do not respect boundary time adjacency. Unfortunately, *exchangeable* random measures (among which the Dirichlet process and the Pitman-Yor process) are by far more widely adopted by the machine learning community than non-exchangeable random measures. Thus, this approach might be perceived as overly complex. That being said, as noted by Foti and Williamson (2015), non-exchangeable normalized random measures may be regarded as Poisson point processes (with varying intensity functions) on some

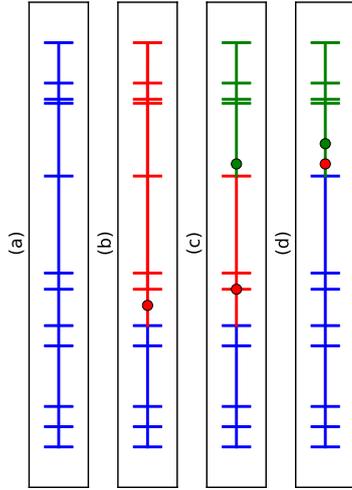


Figure 4: Effects of domain partition through change-points (coloured circles), on kernel membership. Each vertical bar corresponds to a distinct boundary time a_k^j . For the same collection of boundary times, we consider four scenarios: (a) no partition, (b) partition of the domain in two by a single change-point that does not coincide with any existing boundary time, (c) partition of the domain in three by two change-points, one of which coincides with an existing boundary time, and (d) partition of the domain in two by two distinct change-points. In each scenario, kernel membership is illustrated by colour-coding. The colour of the interval between two consecutive boundary times a_k^j and a_{k+1}^j reflects what kernel configuration drives the corresponding string; in particular, the colour of the vertical bar corresponding to boundary time a_{k+1}^j determines what kernel configuration should be used to compute the conditional distribution of the value of the derivative string GP (z_t^j, z_t^j) at a_{k+1}^j , given its value at a_k^j .

augmented spaces, which makes this choice of prior specification somewhat similar, but stronger (that is more informative) than the one we adopt in this paper.

Before deriving the sampling algorithm, it is worth noting that the prior defined in Equation (16) does not admit a density with respect to the same base measure,¹⁰ as the number of change-points $\#\{d_j^i\}$, and subsequently the number of kernel configurations, may vary from one sample to another. Nevertheless, the joint distribution over the data \mathcal{D} and all other model parameters is well defined and, as we will see later, we may leverage reversible-jump MCMC techniques (Green (1995); Green and Hastie (2009)) to construct a Markov chain that converges to the posterior distribution.

5.2.4 OVERALL STRUCTURE OF THE MCMC SAMPLER

To ease notations, we denote \mathbf{c} the set of all change-points in all input dimensions, we denote $\mathbf{n} = (\dots, \#\{d_j^i\}, \dots) \in \mathbb{N}^d$ the vector of the numbers of change-points in each input dimension, we denote $\boldsymbol{\theta}$ the set of kernel hyper-parameters,¹¹ and $\boldsymbol{\rho} := (\dots, \rho^1, \dots)$ the vector of variances of the independent log-normal priors on $\boldsymbol{\theta}$. We denote $\boldsymbol{\lambda} := (\dots, \lambda^1, \dots)$ the vector of change-points intensities, we denote $\boldsymbol{\alpha} := (\dots, \alpha^d, \dots)$ and $\boldsymbol{\beta} := (\dots, \beta^1, \dots)$ the vectors of parameters of the Gamma priors we placed on the change-points intensities across the d input dimensions, and we recall that \mathbf{u} denotes the vector of likelihood parameters other than the values of the latent function f .

We would like to sample from the posterior distribution $p(\mathbf{f}, \mathbf{f}^*, \nabla \mathbf{f}, \nabla \mathbf{f}^* | \mathcal{D}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho})$, where \mathbf{f} and \mathbf{f}^* are the vectors of values of the latent function f at training and test inputs respectively, and $\nabla \mathbf{f}, \nabla \mathbf{f}^*$ the corresponding gradients. Denoting \mathbf{z} the vector of univariate *string GP* processes and their derivatives, evaluated at all distinct training and test input coordinate values, we note that to sample from $p(\mathbf{f}, \mathbf{f}^*, \nabla \mathbf{f}, \nabla \mathbf{f}^* | \mathcal{D}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho})$, it suffices to sample from $p(\mathbf{z} | \mathcal{D}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho})$, compute \mathbf{f} and \mathbf{f}^* using the link function, and compute the gradients using Equation (11). To sample from $p(\mathbf{z} | \mathcal{D}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho})$, we may sample from the target distribution

$$\pi(\mathbf{n}, \mathbf{c}, \boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{z}, \mathbf{u}) := p(\mathbf{n}, \mathbf{c}, \boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{z}, \mathbf{u} | \mathcal{D}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}), \quad (19)$$

and discard variables that are not of interest. As previously discussed, π is not absolutely continuous with respect to the same base measure, though we may still decompose it as

$$\pi(\mathbf{n}, \mathbf{c}, \boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{z}, \mathbf{u}) = \frac{1}{p(\mathcal{D} | \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho})} p(\mathbf{n} | \boldsymbol{\lambda}) p(\boldsymbol{\lambda} | \boldsymbol{\alpha}, \boldsymbol{\beta}) p(\mathbf{c} | \mathbf{n}) p(\boldsymbol{\theta} | \mathbf{n}, \boldsymbol{\rho}) p(\mathbf{u} | \mathbf{z}, \mathbf{c}, \boldsymbol{\theta}) p(\mathcal{D} | \mathbf{z}, \mathbf{u}), \quad (20)$$

where we use the notation $p(\cdot)$ and $p(\cdot | \cdot)$ to denote probability measures rather than probability density functions or probability mass functions, and where product and scaling operations are usual measure operations. Before proceeding any further, we will introduce a slight re-parametrization of Equation (20) that will improve the inference scheme.

Let $\mathbf{n}_a = (\dots, \#\{d_k^j\}, \dots)$ be the vector of the numbers of unique boundary times in all d input dimensions. We recall from our prior on f that

$$p(\mathbf{z} | \mathbf{c}, \boldsymbol{\theta}) = \prod_{j=1}^d p \left(\begin{matrix} z_{a_0^j}^j, z_{a_0^j}^{j'} \\ z_{a_k^j}^j, z_{a_k^j}^{j'} \end{matrix} \middle| \begin{matrix} z_{a_{k-1}^j}^j, z_{a_{k-1}^j}^{j'} \\ z_{a_{k-1}^j}^j, z_{a_{k-1}^j}^{j'} \end{matrix} \right), \quad (21)$$

10. That is the joint prior probability measure is neither discrete, nor continuous.

11. To simplify the exposition, we assume without loss of generality that each kernel configuration has the same kernel functional form, so that configurations are defined by kernel hyper-parameters.

where each factor in the decomposition above is a bivariate Gaussian density whose mean vector and covariance matrix is obtained from the partitions \mathbf{c} , the kernel hyper-parameters $\boldsymbol{\theta}$, and the kernel membership scheme described in Section 5.2.3 and illustrated in Figure 4, and using Equations (6-7). Let \mathbf{K}_{cov} be the unconditional covariance matrix between $\begin{pmatrix} z_{a_0^j}^j, z_{a_0^j}^{j'} \end{pmatrix}$ and $\begin{pmatrix} z_{a_k^j}^j, z_{a_k^j}^{j'} \end{pmatrix}$ as per the unconditional kernel structure driving the string defined on the interval $[a_k^j, a_{k+1}^j]$. Let $\Sigma_k^j := {}^0 \mathbf{K}_{a_0^j, a_0^j}$ be the auto-covariance matrix of $\begin{pmatrix} z_{a_0^j}^j, z_{a_0^j}^{j'} \end{pmatrix}$. Let

$$\Sigma_k^j := {}^j \mathbf{K}_{a_k^j, a_k^j} - {}^j \mathbf{K}_{a_k^j, a_{k-1}^j} {}^j \mathbf{K}_{a_{k-1}^j, a_{k-1}^j}^{-1} {}^j \mathbf{K}_{a_{k-1}^j, a_k^j}^T$$

be the covariance matrix of $\begin{pmatrix} z_{a_k^j}^j, z_{a_k^j}^{j'} \end{pmatrix}$ given $\begin{pmatrix} z_{a_{k-1}^j}^j, z_{a_{k-1}^j}^{j'} \end{pmatrix}$, and

$$M_k^j := {}^j \mathbf{K}_{a_k^j, a_{k-1}^j} {}^j \mathbf{K}_{a_{k-1}^j, a_{k-1}^j}^{-1}$$

Finally, let $L_k^j := U_k^j (D_k^j)^{\frac{1}{2}}$ with $\Sigma_k^j = U_k^j D_k^j (U_k^j)^T$ the singular value decomposition (SVD) of Σ_k^j . We may choose to represent $\begin{pmatrix} z_{a_0^j}^j, z_{a_0^j}^{j'} \end{pmatrix}$ as

$$\begin{bmatrix} z_{a_0^j}^j \\ z_{a_0^j}^{j'} \end{bmatrix} = L_0^j \mathbf{x}_{a_0^j}^j, \quad (22)$$

and for $k > 0$ we may also choose to represent $\begin{pmatrix} z_{a_k^j}^j, z_{a_k^j}^{j'} \end{pmatrix}$ as

$$\begin{bmatrix} z_{a_k^j}^j \\ z_{a_k^j}^{j'} \end{bmatrix} = M_k^j \begin{bmatrix} z_{a_{k-1}^j}^j \\ z_{a_{k-1}^j}^{j'} \end{bmatrix} + L_k^j \mathbf{x}_{a_k^j}^j, \quad (23)$$

where $\{\mathbf{x}_{a_k^j}^j\}$ are independent bivariate standard normal vectors. Equations (22-23) provide an equivalent representation. In effect, we recall that if $Z = M + LX$, where $X \sim \mathcal{N}(0, I)$ is a standard multivariate Gaussian, M is a real vector, and L is a real matrix, then $Z \sim \mathcal{N}(M, LL^T)$. Equations (22-23) result from applying this result to $\begin{pmatrix} z_{a_0^j}^j, z_{a_0^j}^{j'} \end{pmatrix}$ and $\begin{pmatrix} z_{a_k^j}^j, z_{a_k^j}^{j'} \end{pmatrix} \middle| \begin{pmatrix} z_{a_{k-1}^j}^j, z_{a_{k-1}^j}^{j'} \end{pmatrix}$. We note that at training time, M_k^j and L_k^j only depend on kernel hyper-parameters. Denoting \mathbf{x} the vector of all $\mathbf{x}_{a_k^j}^j$, \mathbf{x} is a so-called ‘whitened’ representation of \mathbf{z} , which we prefer for reasons we will discuss shortly. In the whitened representation, the target distribution π is re-parameterized as

$$\pi(\mathbf{n}, \mathbf{c}, \boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{x}, \mathbf{u}) = \frac{1}{p(\mathcal{D} | \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho})} p(\mathbf{n} | \boldsymbol{\lambda}) p(\boldsymbol{\lambda} | \boldsymbol{\alpha}, \boldsymbol{\beta}) p(\mathbf{c} | \mathbf{n}) p(\boldsymbol{\theta} | \mathbf{n}, \boldsymbol{\rho}) p(\mathbf{u} | \mathbf{x}) p(\mathcal{D} | \mathbf{x}, \mathbf{c}, \boldsymbol{\theta}, \mathbf{u}), \quad (24)$$

where the dependency of the likelihood term on the partitions and the hyper-parameters stems from the need to recover \mathbf{z} and subsequently \mathbf{f} from \mathbf{x} through Equations (22) and (23). The whitened representation Equation (24) has two primary advantages. Firstly, it is robust to ill-conditioning of

Σ_k^j , which would typically occur when two adjacent boundary times are too close to each other. In the representation of Equation (20), as one needs to evaluate the density $p(\mathbf{z}|\mathbf{c}, \boldsymbol{\theta})$, ill-conditioning of Σ_k^j would result in numerical instabilities. In contrast, in the whitened representation, one needs to evaluate the density $p(\mathbf{x})$, which is that of i.i.d. standard Gaussians and as such can be evaluated robustly. Moreover, the SVD required to evaluate L_k^j is also robust to ill-conditioning of Σ_k^j , so that Equations (22) and (23) hold and can be robustly evaluated for degenerate Gaussians too. The second advantage of the whitened representation is that it improves mixing by establishing a link between kernel hyper-parameters and the likelihood.

Equation (24) allows us to cast our inference problem as a Bayesian model selection problem under a countable family of models indexed by $\mathbf{n} \in \mathbb{N}^d$, each defined on a different parameter subspace $\mathcal{C}_{\mathbf{n}}$, with cross-model normalizing constant $p(\mathcal{D}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho})$, model probability driven by $p(\mathbf{n}|\boldsymbol{\lambda})p(\boldsymbol{\lambda}|\boldsymbol{\alpha}, \boldsymbol{\beta})$, model-specific prior $p(\mathbf{c}|\mathbf{n})p(\boldsymbol{\theta}|\mathbf{n}, \boldsymbol{\rho})p(\mathbf{u}|\mathbf{x})$, and likelihood $p(\mathcal{D}|\mathbf{x}, \mathbf{c}, \boldsymbol{\theta}, \mathbf{u})$. Critically, it can be seen from Equation (24) that the conditional probability distribution

$$\pi(\mathbf{c}, \boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{x}, \mathbf{u}|\mathbf{n})$$

admits a density with respect to Lebesgue's measure on $\mathcal{C}_{\mathbf{n}}$.

Our setup is therefore analogous to that which motivated the seminal paper Green (1995), so that to sample from the posterior $\pi(\mathbf{c}, \boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{x}, \mathbf{u}, \mathbf{n})$ we may use any Reversible-Jump Metropolis-Hastings (RJ-MH) scheme satisfying detailed balance and dimension-matching as described in section 3.3 of Green (1995). To improve mixing of the Markov chain, we will alternate between a *between-models* RJ-MH update with target distribution $\pi(\mathbf{n}, \mathbf{c}, \boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{x}, \mathbf{u})$, and a *within-model* MCMC-within-Gibbs sampler with target distribution $\pi(\mathbf{c}, \boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{x}, \mathbf{u}|\mathbf{n})$. Constructing reversible-jump samplers by alternating between within-model sampling and between-models sampling is standard practice, and it is well-known that doing so yields a Markov chain that converges to the target distribution of interest (see Brooks et al., 2011, p. 50).

In a slight abuse of notation, in the following we might use the notations $p(\cdot|\cdot)$ and $p(\cdot)$, which we previously used to denote probability measures, to refer to the corresponding probability density functions or probability mass functions.

5.2.5 WITHIN-MODEL UPDATES

We recall from Equation (24) that $\mathbf{c}, \boldsymbol{\theta}, \boldsymbol{\lambda}, \mathbf{x}, \mathbf{u}|\mathcal{D}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\rho}, \mathbf{n}$ has probability density function

$$p(\mathbf{n}|\boldsymbol{\lambda})p(\boldsymbol{\lambda}|\boldsymbol{\alpha}, \boldsymbol{\beta})p(\mathbf{c}|\mathbf{n})p(\boldsymbol{\theta}|\mathbf{n}, \boldsymbol{\rho})p(\mathbf{u}|\mathbf{x})p(\mathcal{D}|\mathbf{x}, \mathbf{c}, \boldsymbol{\theta}, \mathbf{u}), \quad (25)$$

up to a normalizing constant.

Updating $\boldsymbol{\lambda}$: By independence of the priors over $(\boldsymbol{\lambda}[j], \mathbf{n}[j])$, the distributions $\boldsymbol{\lambda}[j] | \mathbf{n}[j]$ are also independent, so that the updates may be performed in *parallel*. Moreover, recalling that the prior number of change-points in the j -th input dimension is Poisson distributed with intensity $\boldsymbol{\lambda}[j](b^j - a^j)$, and by conjugacy of the Gamma distribution to the Poisson likelihood, it follows that

$$\boldsymbol{\lambda}[j] | \mathbf{n}[j] \sim \Gamma\left(\frac{\mathbf{n}[j]}{b^j - a^j} + \boldsymbol{\alpha}[j], 1 + \boldsymbol{\beta}[j]\right). \quad (26)$$

This update step has memory requirement and time complexity both constant in the number of training and test samples.

Updating \mathbf{u} : When the likelihood has additional parameters \mathbf{u} , they may be updated with a Metropolis-Hastings step. Denoting $q(\mathbf{u} \rightarrow \mathbf{u}')$ the proposal probability density function, the acceptance ratio reads

$$r_{\mathbf{u}} = \min\left(1, \frac{p(\mathbf{u}')p(\mathcal{D}|\mathbf{x}, \mathbf{c}, \boldsymbol{\theta}, \mathbf{u}')q(\mathbf{u} \rightarrow \mathbf{u}')}{p(\mathbf{u})p(\mathcal{D}|\mathbf{x}, \mathbf{c}, \boldsymbol{\theta}, \mathbf{u})q(\mathbf{u}' \rightarrow \mathbf{u})}\right). \quad (27)$$

In some cases however, it might be possible and more convenient to choose $p(\mathbf{u})$ to be conjugate to the likelihood $p(\mathcal{D}|\mathbf{x}, \mathbf{c}, \boldsymbol{\theta}, \mathbf{u})$. For instance, in regression problems under a Gaussian noise model, we may take \mathbf{u} to be the noise variance on which we may place an inverse-gamma prior. Either way, the computational bottleneck of this step is the evaluation of the likelihood $p(\mathcal{D}|\mathbf{x}, \mathbf{c}, \boldsymbol{\theta}, \mathbf{u}')$, which in most cases can be done with a time complexity and memory requirement that are both linear in the number of training samples.

Updating \mathbf{c} : We update the positions of change-points sequentially using the Metropolis-Hastings algorithm, one input dimension j at a time, and for each input dimension we proceed in increasing order of change-points. The proposal new position for the change-point c_j^b is sampled uniformly at random on the interval $[c_{j-p-1}^b, c_{j+p+1}^b]$, where c_{j-p-1}^b (resp. c_{j+p+1}^b) is replaced by a^j (resp. b^j) for the first (resp. last) change-point. The acceptance probability of this proposal is easily found to be

$$r_{c_j^b} = \min\left(1, \frac{p(\mathcal{D}|\mathbf{x}, \mathbf{c}', \boldsymbol{\theta}, \mathbf{u})}{p(\mathcal{D}|\mathbf{x}, \mathbf{c}, \boldsymbol{\theta}, \mathbf{u})}\right), \quad (28)$$

where \mathbf{c}' is identical to \mathbf{c} except for the change-point to update. This step requires computing the factors $\{J_k^j, M_k^j\}$ corresponding to inputs in j -th dimension whose kernel configuration would change if the proposal were to be accepted, the corresponding vector of *derivative string GP* values \mathbf{z} , and the observation likelihood under the proposal $p(\mathcal{D}|\mathbf{x}, \mathbf{c}, \boldsymbol{\theta}, \mathbf{u})$. The computational bottleneck of this step is therefore once again the evaluation of the new likelihood $p(\mathcal{D}|\mathbf{x}, \mathbf{c}', \boldsymbol{\theta}, \mathbf{u})$.

Updating \mathbf{x} : The target conditional density of \mathbf{x} is proportional to

$$p(\mathbf{x})p(\mathcal{D}|\mathbf{x}, \mathbf{c}, \boldsymbol{\theta}, \mathbf{u}). \quad (29)$$

Recalling that $p(\mathbf{x})$ is a multivariate standard normal, it follows that the form of Equation (29) makes it convenient to use elliptical slice sampling (Murray et al. (2010)) to sample from the unnormalized conditional $p(\mathbf{x})p(\mathcal{D}|\mathbf{x}, \mathbf{c}, \boldsymbol{\theta}, \mathbf{u})$. The two bottlenecks of this update step are sampling a new proposal from $p(\mathbf{x})$ and evaluating the likelihood $p(\mathcal{D}|\mathbf{x}, \mathbf{c}, \boldsymbol{\theta}, \mathbf{u})$. Sampling from the multivariate standard normal $p(\mathbf{x})$ may be *massively parallelized*, for instance by using GPU Gaussian random number generators. When no parallelism is available, the overall time complexity reads $\mathcal{O}\left(\sum_{j=1}^d \mathbf{n}_a[j]\right)$, where we recall that $\mathbf{n}_a[j]$ denotes the number of distinct training and testing input coordinates in the j -th dimension. In particular, if we denote N the total number of training and testing d -dimensional input samples, then $\sum_{j=1}^d \mathbf{n}_a[j] \leq dN$, although for many classes of data sets with sparse input values such as images, where each input (single-colour pixel value) may have at most 256 distinct values, we might have $\sum_{j=1}^d \mathbf{n}_a[j] \ll dN$. As for the memory required to sample from $p(\mathbf{x})$, it grows proportionally to the size of \mathbf{x} , that is in $\mathcal{O}\left(\sum_{j=1}^d \mathbf{n}_a[j]\right)$. In regards to the evaluation of the likelihood $p(\mathcal{D}|\mathbf{x}, \mathbf{c}, \boldsymbol{\theta}, \mathbf{u})$, as previously discussed its resource requirements are application-specific, but it will typically have time complexity that grows in $\mathcal{O}(N)$ and memory requirement that grows in $\mathcal{O}(dN)$. For instance, the foregoing resource requirements always hold

for i.i.d. observation models such as in nonparametric regression and nonparametric classification problems.

Updating θ . We note from Equation (25) that the conditional distribution of θ given everything else has unnormalized density

$$p(\theta|n, \rho)p(\mathcal{D}|\mathbf{x}, c, \theta, \mathbf{u}), \quad (30)$$

which we may choose to represent as

$$p(\log \theta|n, \rho)p(\mathcal{D}|\mathbf{x}, c, \log \theta, \mathbf{u}). \quad (31)$$

As we have put independent log-normal priors on the coordinates of θ (see Equation 16), we may once again use elliptical slice sampling to sample from $\log \theta$ before taking the exponential. The time complexity of generating a new sample from $p(\log \theta|n, \rho)$ will typically be at most linear in the total number of distinct kernel hyper-parameters. Overall, the bottleneck of this update is the evaluation of the likelihood $p(\mathcal{D}|\mathbf{x}, c, \log \theta, \mathbf{u})$. In this update, the latter operation requires recomputing the factors M_k^i and L_k^j of Equations (22) and (23), which requires computing and taking the SVD of unrelated 2×2 matrices, computations we may perform in *parallel*. Once the foregoing factors have been computed, we evaluate \mathbf{z} , the *derivative string GP* values at boundary times, parallelizing over input dimensions, and running a sequential update within an input dimension using Equations (22) and (23). Updating \mathbf{z} therefore has time complexity that is, in the worst case where no distributed computing is available, $\mathcal{O}(dN)$, and $\mathcal{O}(N)$ when there are up to d computing cores. The foregoing time complexity will also be that of this update step, unless the observation likelihood is more expensive to evaluate. The memory requirement, as in previous updates, is $\mathcal{O}(dN)$.

Overall resource requirement. To summarize previous remarks, the overall computational bottleneck of a *within-model* iteration is the evaluation of the likelihood $p(\mathcal{D}|\mathbf{x}, c, \theta, \mathbf{u})$. For i.i.d. observation models such as classification and regression problems for instance, the corresponding time complexity grows in $\mathcal{O}(N)$ when d computing cores are available, or $\mathcal{O}(dN)$ otherwise, and the memory requirement grows in $\mathcal{O}(dN)$.

5.2.6 BETWEEN-MODELS UPDATES

Our reversible-jump Metropolis-Hastings update proceeds as follows. We choose an input dimension, say j , uniformly at random. If j has no change-points, that is $\mathbf{n}[j] = 0$, we randomly choose between not doing anything, and adding a change-point, each outcome having the same probability. If $\mathbf{n}[j] > 0$, we either do nothing, add a change-point, or delete a change-point, each outcome having the same probability of occurrence.

Whenever we choose not to do anything, the acceptance ratio is easily found to be one:

$$r_{j0} = 1. \quad (32)$$

Whenever we choose to add a change-point, we sample the position c_k^j of the proposal new change-point uniformly at random on the domain $[a^j, b^j]$ of the j -th input dimension. This proposal will almost surely break an existing kernel membership cluster, say the p -th, into two: that is $c_p^j < c_k^j < c_{p+1}^j$ where we may have $a^j = c_p^j$ and/or $b^j = c_{p+1}^j$. In the event c_k^j coincides with an existing change-point, which should happen with probability 0, we do nothing. When adding a change-point, we sample a new vector of hyper-parameters θ_k^j from the log-normal prior of Equation (16), and we

propose as hyper-parameters for the tentative new clusters $[c_p^j, c_k^j]$ and $[c_k^j, c_{p+1}^j]$ the vectors $\theta_{\text{add-left}}^j$ and $\theta_{\text{add-right}}^j$ defined as

$$\log \theta_{\text{add-left}}^j := \cos(\alpha) \log \theta_p^j - \sin(\alpha) \log \theta_k^j \quad (33)$$

and

$$\log \theta_{\text{add-right}}^j := \sin(\alpha) \log \theta_p^j + \cos(\alpha) \log \theta_k^j \quad (34)$$

respectively, where $\alpha \in [0, \frac{\pi}{2}]$ and θ_p^j is the vector of hyper-parameters currently driving the kernel membership defined by the cluster $[c_p^j, c_{p+1}^j]$. We note that if θ_p^j is distributed as per the prior in Equation (16) then $\theta_{\text{add-left}}^j$ and $\theta_{\text{add-right}}^j$ are i.i.d. distributed as per the foregoing prior. More generally, this elliptical transformation determines the extent to which the new proposal kernel configurations should deviate from the current configuration θ_p^j . α is restricted to $[0, \frac{\pi}{2}]$ so as to give a positive weight to the current vector of hyper-parameters θ_p^j . When $\alpha = 0$, the left hand-side cluster $[c_p^j, c_k^j]$ will fully exploit the current kernel configuration, while the right hand-side cluster $[c_k^j, c_{p+1}^j]$ will use the prior to explore a new set of hyper-parameters. When $\alpha = \frac{\pi}{2}$ the reverse occurs. To preserve symmetry between the left and right hand-side kernel configurations, we choose

$$\alpha = \frac{\pi}{4}. \quad (35)$$

Whenever we choose to delete a change-point, we choose an existing change-point uniformly at random, say c_p^j . Deleting c_p^j would merge the clusters $[c_{p-1}^j, c_p^j]$ and $[c_p^j, c_{p+1}^j]$, where we may have $a^j = c_{p-1}^j$ and/or $b^j = c_{p+1}^j$. We propose as vector of hyper-parameters for the tentative merged cluster $[c_{p-1}^j, c_{p+1}^j]$ the vector $\theta_{\text{del-merged}}^j$ satisfying:

$$\log \theta_{\text{del-merged}}^j = \cos(\alpha) \log \theta_{p-1}^j + \sin(\alpha) \log \theta_p^j, \quad (36)$$

which together with

$$\log \theta_{\text{del}*}^j = -\sin(\alpha) \log \theta_{p-1}^j + \cos(\alpha) \log \theta_p^j, \quad (37)$$

constitute the inverse of the transformation defined by Equations (33) and (34).

Whenever a proposal to add or delete a change-point occurs, the factors L_k^j and M_k^i that would be affected by the change in kernel membership structure are recomputed, and so are the affected coordinates of \mathbf{z} .

This scheme satisfies the reversibility and dimension-matching requirements of Green (1995). Moreover, the absolute value of the Jacobian of the mapping

$$(\log \theta_p^j, \log \theta_k^j) \rightarrow (\log \theta_{\text{add-left}}^j, \log \theta_{\text{add-right}}^j)$$

of the move to add a change-point in $[c_p^j, c_{p+1}^j]$ reads

$$\left| \frac{\partial (\log \theta_{\text{add-left}}^j, \log \theta_{\text{add-right}}^j)}{\partial (\log \theta_p^j, \log \theta_k^j)} \right| = 1. \quad (38)$$

Similarly, the absolute value of the Jacobian of the mapping corresponding to a move to delete change-point c_p^j , namely

$$\begin{aligned} & \left(\log \theta_{p-1}^j, \log \theta_p^j \right) \rightarrow \left(\log \theta_{\text{del-merged}}^j, \log \theta_{\text{del}^*}^j \right), \\ & \text{reads:} \\ & \left| \frac{\partial \left(\log \theta_{\text{del-merged}}^j, \log \theta_{\text{del}^*}^j \right)}{\partial \left(\log \theta_{p-1}^j, \log \theta_p^j \right)} \right| = 1. \end{aligned} \quad (39)$$

Applying the standard result Equation (8) of Green (1995), the acceptance ratio of the move to add a change-point is found to be

$$r_{j+} = \min \left(1, \frac{p(\mathcal{D}|\mathbf{x}, c_+, \theta_+, \mathbf{u}) \lambda[j] (b^j - a^j) p_{\log \theta_+} \left(\log \theta_{\text{add-left}}^j \right) p_{\log \theta_+} \left(\log \theta_{\text{add-right}}^j \right)}{p(\mathcal{D}|\mathbf{x}, c, \theta, \mathbf{u}) \left(1 + \mathbf{n}[j] p_{\log \theta_+} \left(\log \theta_p^j \right) p_{\log \theta_+} \left(\log \theta_c^j \right) \right)} \right) \quad (40)$$

where $p_{\log \theta_+}$ is the prior over log hyper-parameters in the j -th input dimension (as per the prior specification Equation 16), which we recall is i.i.d. centred Gaussian with variance $\rho[j]$, and c_+ and θ_+ denote the proposal new vector of change-points and the corresponding vector of hyper-parameters. The three coloured terms in the acceptance probability are very intuitive. The green term $\frac{p(\mathcal{D}|\mathbf{x}, c_+, \theta_+, \mathbf{u})}{p(\mathcal{D}|\mathbf{x}, c, \theta, \mathbf{u})}$ represents the fit improvement that would occur if the new proposal is accepted.

In the red term $\frac{\lambda[j](b^j - a^j)}{1 + \mathbf{n}[j]}$, $\lambda[j] (b^j - a^j)$ represents the average number of change-points in the j -th input dimension as per the HPP prior, while $1 + \mathbf{n}[j]$ corresponds to the proposal new number of change-points in the j -th dimension, so that the whole red term acts as a complexity regulariser. Finally, the blue term $\frac{p_{\log \theta_+} \left(\log \theta_{\text{add-left}}^j \right) p_{\log \theta_+} \left(\log \theta_{\text{add-right}}^j \right)}{p_{\log \theta_+} \left(\log \theta_p^j \right) p_{\log \theta_+} \left(\log \theta_c^j \right)}$ plays the role of hyper-parameters regulariser.

Similarly, the acceptance ratio of the move to delete change-point c_p^j , thereby changing the number of change-points in the j -th input dimension from $\mathbf{n}[j]$ to $\mathbf{n}[j] - 1$, is found to be

$$r_{j-} = \min \left(1, \frac{p(\mathcal{D}|\mathbf{x}, c_-, \theta_-, \mathbf{u}) \mathbf{n}[j] p_{\log \theta_-} \left(\log \theta_{\text{del-merged}}^j \right) p_{\log \theta_-} \left(\log \theta_{\text{del}^*}^j \right)}{p(\mathcal{D}|\mathbf{x}, c, \theta, \mathbf{u}) \lambda[j] (b^j - a^j) p_{\log \theta_-} \left(\log \theta_{p-1}^j \right) p_{\log \theta_-} \left(\log \theta_c^j \right)} \right), \quad (41)$$

where c_- and θ_- denote the proposal new vector of change-points and the corresponding vector of hyper-parameters. Once more, each coloured term plays the same intuitive role as its counterpart in Equation (40).

Overall resource requirement: The bottleneck of between-models updates is the evaluation of the new likelihoods $p(\mathcal{D}|\mathbf{x}, c_+, \theta_+, \mathbf{u})$ or $p(\mathcal{D}|\mathbf{x}, c_-, \theta_-, \mathbf{u})$, whose resource requirements, which are the same as those of within-models updates, we already discussed.

Algorithm 2 summarises the proposed MCMC sampler.

5.3 Multi-Output Problems

Although we have restricted ourselves to cases where the likelihood model depends on a single real-valued function for brevity and to ease notations, cases where the likelihood depends on vector-valued functions, or equivalently multiple real-valued functions, present no additional theoretical or

Algorithm 2 MCMC sampler for nonparametric Bayesian inference of a real-valued latent function under a string GP prior

Inputs: Likelihood model $p(\mathcal{D}|\mathbf{f}, \mathbf{u})$, link function ϕ , training data \mathcal{D} , test inputs, type of unconditional kernel, prior parameters α, β, ρ .

Outputs: Posterior samples of the values of the latent function at training and test inputs \mathbf{f} and \mathbf{f}^* , and the corresponding gradients $\nabla \mathbf{f}$ and $\nabla \mathbf{f}^*$.

Step 0: Set $n = 0$ and $c = \emptyset$, and sample $\theta, \lambda, \mathbf{x}, \mathbf{u}$ from their priors.

repeat

Step 1: Perform a within-model update.

1.1: Update each $\lambda[j]$ by sampling from the Gamma distribution in Equation (26).

1.2: Update \mathbf{u} , the vector of other likelihood parameters, if any, using Metropolis-Hastings (MH) with proposal q and acceptance ratio Equation (27) or by sampling directly from the posterior when $p(\mathbf{u})$ is conjugate to the likelihood model.

1.3: Update θ , using elliptical slice sampling (ESS) with target distribution Equation (31), and record the newly computed factors $\{L_k^j, M_k^j\}$ that relate \mathbf{z} to its whitened representation \mathbf{x} .

1.4: Update \mathbf{x} using ESS with target distribution Equation (29).

1.5: Update change-point positions \mathbf{c} sequentially using MH, drawing a proposal update for c_p^j uniformly at random on $\{c_{p-1}^j, c_{p+1}^j\}$, and accepting the update with probability $r_{c_p^j}$ (defined Equation 28). On accept, update the factors $\{L_k^j, M_k^j\}$.

Step 2: Perform a between-models update.

2.1: Sample a dimension to update, say j , uniformly at random.

2.2: Consider adding or deleting a change-point

if $\mathbf{n}[j] = 0$ **then**

Randomly choose to add a change-point with probability $1/2$.

if we should consider adding a change-point **then**

Construct proposals to update following Section 5.2.6.

Accept proposals with probability r_+^j (see Equation 40).

end if

else

Randomly choose to add/delete a change-point with probability $1/3$.

if we should consider adding a change-point **then**

Construct proposals to update following Section 5.2.6.

Accept proposals with probability r_+^j (see Equation 40).

else if we should consider deleting a change-point **then**

Construct proposals to update following Section 5.2.6.

Accept proposals with probability r_-^j (see Equation 41).

else

Continue.

end if

end if

Step 3: Compute $\mathbf{f}, \nabla \mathbf{f}$ and $\nabla \mathbf{f}^*$, first recovering \mathbf{z} from \mathbf{x} , and then recalling that $f(x) =$

$$\phi \left(z_{x[1]}, \dots, z_{x[d]} \right) \text{ and } \nabla f(x) = \left(z_{x[1]} \frac{\partial \phi}{\partial x[1]}(x), \dots, z_{x[d]} \frac{\partial \phi}{\partial x[d]}(x) \right).$$

until enough samples are generated after mixing.

practical challenge. We may simply put independent *string GP* priors on each of the latent functions. An MCMC sampler almost identical to the one introduced herein may be used to sample from the posterior. All that is required to adapt the proposed MCMC sampler to multi-outputs problems is to redefine \mathbf{z} to include all univariate *derivative string GP* values across input dimensions and across latent functions, perform step 1.1 of Algorithm 2 for each of the latent function, and update step 2.1 so as to sample uniformly at random not only what dimension to update but also what latent function. Previous analyses and derived acceptance ratios remain unchanged. The resource requirements of the resulting multi-outputs MCMC sampler on a problem with K latent functions, N training and test d -dimensional inputs, are the same as those of the MCMC sampler for a single output (Algorithm 2) with N training and test dK -dimensional inputs. The time complexity is $\mathcal{O}(N)$ when dK computing cores are available, $\mathcal{O}(dKN)$ when no distributed computing is available, and the memory requirement becomes $\mathcal{O}(dKN)$.

5.4 Flashback to Small Scale GP Regressions with String GP Kernels

In Section 5.1 we discussed maximum marginal likelihood inference in Bayesian nonparametric regressions under additively separable *string GP* priors, or GP priors with *string GP* covariance functions. We proposed learning the positions of boundary times, conditional on their number, jointly with kernel hyper-parameters and noise variances by maximizing the marginal likelihood using gradient-based techniques. We then suggested learning the number of strings in each input dimension by trading off goodness-of-fit with model simplicity using information criteria such as AIC and BIC. In this section, we propose a fully Bayesian nonparametric alternative.

Let us consider the Gaussian process regression model

$$y_i = f(x_i) + \epsilon_i, \quad f \sim \mathcal{GP}(0, k_{\text{SGP}}(\cdot, \cdot)), \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2), \quad (42)$$

$$x_i \in [a^1, b^1] \times \dots \times [a^d, b^d], \quad y_i, \epsilon_i \in \mathbb{R}, \quad (43)$$

where k_{SGP} is the covariance function of some *string GP* with boundary times $\{a_k^j\}$ and corresponding unconditional kernels $\{k_k^j\}$ in the j -th input dimension. It is worth stressing that we place a GP (not *string GP*) prior on the latent function f , but the covariance function of the GP is a *string GP* covariance function (as discussed in Section 4.2 and as derived in Appendix I). Of course when the *string GP* covariance function k_{SGP} is separably additive, the two functional priors are the same. However, we impose no restriction on the link function of the *string GP* that k_{SGP} is the covariance function of, other than continuous differentiability. To make full Bayesian nonparametric inference, we may place on the boundary times $\{a_k^j\}$ independent homogeneous Poisson process priors, each with intensity λ^j . Similarly to the previous section (Equation 16) our full prior specification of the *string GP* kernel reads

$$\begin{cases} \lambda^j \sim \Gamma(\alpha^j, \beta^j), \\ \{a_k^j\} | \lambda^j \sim \text{HPP}(\lambda^j) \\ \theta_k^j | \{a_k^j\}, \lambda^j \stackrel{\text{i.i.d.}}{\sim} \log \mathcal{N}(0, \rho^j) \\ \forall (j, k) \neq (l, p) \theta_k^j \perp \theta_p^l \end{cases}, \quad (44)$$

where θ_k^j is the vector of hyper-parameters driving the unconditional kernel k_k^j . The method developed in the previous section and the resulting MCMC sampling scheme (Algorithm 2) may be

reused to sample from the posterior over function values, pending the following two changes. First, gradients ∇f and ∇f^* are no longer necessary. Second, we may work with function values (f, f^*) directly (that is in the original as opposed to whitened space). The resulting (Gaussian) distribution of function values (f, f^*) conditional on all other variables is then analytically derived using standard Gaussian identities. Like it is done in vanilla Gaussian process regression, so that the within-model update of (f, f^*) is performed using a single draw from a multivariate Gaussian.

This approach to model complexity learning is advantageous over the information criteria alternative of Section 5.1 in that it scales better with large input-dimensions. Indeed, rather than performing complete maximum marginal likelihood inference a number of times that grows exponentially with the input dimension, the approach of this section alternates between exploring a new combination of numbers of kernel configurations in each input dimension, and exploring function values and kernel hyper-parameters (given their number). That being said, this approach should only be considered as an alternative to commonly used kernels for *small scale* regression problems to enable the learning of local patterns. Crucially, it scales as poorly as the *standard GP paradigm*, and Algorithm 2 should be preferred for large scale problems.

6. Experiments

We now move on to presenting empirical evidence for the efficacy of *string GPs* in coping with local patterns in data sets, and in doing so in a scalable manner. Firstly we consider maximum marginal likelihood inference on two small scale problems exhibiting local patterns. We begin with a toy experiment that illustrates the limitations of the *standard GP paradigm* in extrapolating and interpolating simple local periodic patterns. Then, we move on to comparing the accuracy of Bayesian nonparametric regression under a *string GP* prior to that of the standard Gaussian process regression model and existing mixture-of-experts alternatives on the motorcycle data set of Silverman (1985), commonly used for the local patterns and heteroskedasticity it exhibits. Finally, we illustrate the performance of the previously derived MCMC sampler on two large scale Bayesian inference problems, namely the prediction of U.S. commercial airline arrival delays of Hensman et al. (2013) and a new large scale dynamic asset allocation problem.

6.1 Extrapolation and Interpolation of Synthetic Local Patterns

In our first experiment, we illustrate a limitation of the standard approach consisting of postulating a global covariance structure on the domain, namely that this approach might result in unwanted global extrapolation of local patterns, and we show that this limitation is addressed by the *string GP paradigm*. To this aim, we use 2 toy regression problems. We consider the following functions:

$$f_0(t) = \begin{cases} \sin(60\pi t) & t \in [0, 0.5] \\ \frac{1}{2} \sin(16\pi t) & t \in [0.5, 1] \end{cases}, \quad f_1(t) = \begin{cases} \sin(16\pi t) & t \in [0, 0.5] \\ \frac{1}{2} \sin(32\pi t) & t \in [0.5, 1] \end{cases}. \quad (45)$$

f_0 (resp. f_1) undergoes a sharp (resp. mild) change in frequency and amplitude at $t = 0.5$. We consider using their restrictions to $[0.25, 0.75]$ for training. We sample those restrictions with frequency 300, and we would like to extrapolate the functions to the rest of their domains using Bayesian non-parametric regression.

We compare marginal likelihood *string GP* regression models, as described in Section 5.1, to vanilla GP regression models using popular and expressive kernels. All *string GP* models have two strings and the partition is learned in the marginal likelihood maximisation. Figure 5 illustrates

plots of the posterior means for each kernel used, and Table 2 compares predictive errors. Overall, it can be noted that the *string GP kernel* with the periodic kernel (MacKay (1998)) as building block outperforms competing kernels, including the expressive spectral mixture kernel

$$k_{SM}(\tau) = \sum_{k=1}^K \sigma_k^2 \exp(-2\pi^2 \tau^2 \gamma_k^2) \cos(2\pi \tau \mu_k)$$

of Wilson and Adams (2013) with $K = 5$ mixture components.¹²

The comparison between the spectral mixture kernel and the string spectral mixture kernel is of particular interest, since spectral mixture kernels are pointwise dense in the family of stationary kernels, and thus can be regarded as flexible enough for learning *stationary* kernels from the data. In our experiment, the *string spectral mixture kernel* with a single mixture component per string significantly outperforms the spectral mixture kernel with 5 mixture components. This intuitively can be attributed to the fact that, regardless of the number of mixture components in the spectral mixture kernel, the learned kernel must account for both types of patterns present in each training data set. Hence, each local extrapolation on each side of 0.5 will attempt to make use of both amplitudes and both frequencies evidenced in the corresponding training data set, and will struggle to recover the true local sine function. We would expect that the performance of the spectral mixture kernel in this experiment will not improve drastically as the number of mixture components increases. However, under a *string GP* prior, the left and right hand side strings are independent conditional on the (unknown) boundary conditions. Therefore, when the *string GP* domain partition occurs at time 0.5, the training data set on $[0.25, 0.5]$ influences the hyper-parameters of the string to the right of 0.5 only to the extent that both strings should agree on the value of the latent function and its derivative at 0.5. To see why this is a weaker condition, we consider the family of pair of functions:

$$(\alpha\omega_1 \sin(\omega_2 t), \alpha\omega_2 \sin(\omega_1 t)), \quad \omega_i = 2\pi k_i, \quad k_i \in \mathbb{N}, \quad \alpha \in \mathbb{R}.$$

Such functions always have the same value and derivative at 0.5, regardless of their frequencies, and they are plausible GP paths under a spectral mixture kernel with one single mixture component ($\mu_k = k_i$ and $\gamma_k \ll 1$), and under a periodic kernel. As such it is not surprising that extrapolation under a string spectral mixture kernel or a string periodic kernel should perform well.

To further illustrate that *string GPs* are able to learn local patterns that GPs with commonly used and expressive kernels can't, we consider interpolating two bivariate functions f_2 and f_3 that exhibit local patterns. The functions are defined as:

$$\forall u, v \in [0.0, 1.0] \quad f_2(u, v) = f_0(u)f_1(v), \quad f_3(u, v) = \sqrt{f_0(u)^2 + f_1(v)^2}. \quad (46)$$

We consider recovering the original functions as the posterior mean of a GP regression model trained on $[0.0, 0.4] \cup [0.6, 1.0] \times [0.0, 0.4] \cup [0.6, 1.0]$. Each bivariate kernel used is a product of two univariate kernels in the same family, and we used standard Kronecker techniques to speed-up inference (see Saatchi, 2011, p.134). The univariate kernels we consider are the same as previously. Each univariate *string GP* kernel has one change-point (two strings) whose position is learned by maximum marginal likelihood. Results are illustrated in Figures 6 and 7. Once again it can be seen that unlike any competing kernel, the product of string periodic kernels recover both functions almost perfectly. In particular, it is impressive to see that, despite f_3 not being a separable function,

a product of string periodic kernels recovered it almost perfectly. The interpolations performed by the spectral mixture kernel (see Figures 6 and 7) provide further evidence for our previously developed narrative: the spectral mixture kernel tries to blend all local patterns found in the training data during the interpolation. The periodic kernel learns a single global frequency characteristic of the whole data set, ignoring local patterns, while the squared exponential, Matérn and rational quadratic kernels merely attempt to perform interpolation by smoothing.

Although we used synthetic data to ease illustrating our argument, it is reasonable to expect that in real-life problems the bigger the data set, the more likely there might be local patterns that should not be interpreted as noise and yet are not indicative of the data set as whole.

12. The sparse spectrum kernel of Lazaro-Gredilla et al. (2010) can be thought of as the special case $\gamma_k \ll 1$.

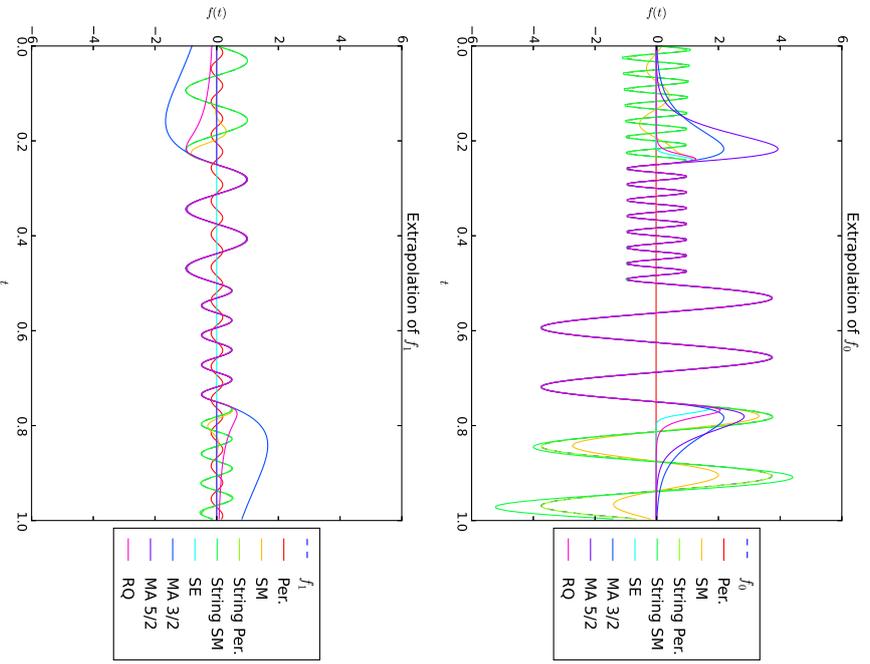


Figure 5: Extrapolation of two functions f_0 and f_1 through Bayesian nonparametric regression under *string GP* priors and vanilla GP priors with popular and expressive kernels. Each model is trained on $[0.25, 0.5]$ and extrapolates to $[0, 1.0]$.

Kernel	Absolute Error		Squared Error	
	f_0	f_1	f_0	f_1
Squared exponential	1.44 ± 2.40	0.48 ± 0.58	3.50 ± 9.20	0.31 ± 0.64
Rational quadratic	1.39 ± 2.31	0.51 ± 0.83	3.28 ± 8.79	0.43 ± 1.15
Matérn 3/2	1.63 ± 2.53	1.26 ± 1.37	4.26 ± 11.07	2.06 ± 3.55
Matérn 5/2	1.75 ± 2.77	0.48 ± 0.58	5.00 ± 12.18	0.31 ± 0.64
Periodic	1.51 ± 2.45	0.53 ± 0.60	3.79 ± 9.62	0.37 ± 0.72
Spec. Mix. (5 comp.)	0.75 ± 1.15	0.39 ± 0.57	0.94 ± 2.46	0.24 ± 0.58
String Spec. Mix. (2 strings, 1 comp.)	0.23 ± 0.84	0.01 ± 0.03	0.21 ± 1.07	0.00 ± 0.00
String Periodic	0.02 ± 0.02	0.00 ± 0.01	0.00 ± 0.00	0.00 ± 0.00

Table 2: Predictive accuracies in the extrapolation of the two functions f_0 and f_1 of Section 6.1 through Bayesian nonparametric regression under *string GP* priors and vanilla GP priors with popular and expressive kernels. Each model is trained on $[0.25, 0.5]$ and extrapolates to $[0, 1.0]$. The predictive errors are reported as average ± 2 standard deviations.

6.2 Small Scale Heteroskedastic Regression

In our second experiment, we consider illustrating the advantage of the *string GP paradigm* over the *standard GP paradigm*, but also over the alternatives of Kim et al. (2005), Gramacy and Lee (2008), Tresp (2000) and Deisenroth and Ng (2015) that consist of considering independent GP experts on disjoint parts of the domain or handling disjoint subsets of the data. Using the motorcycle data set of Silverman (1985), commonly used for the local patterns and heteroskedasticity it exhibits, we show that our approach outperforms the aforementioned competing alternatives, thereby providing empirical evidence that the collaboration between consecutive GP experts introduced in the *string GP paradigm* vastly improves predictive accuracy and certainty in regression problems with local patterns. We also illustrate learning of the derivative of the latent function, solely from noisy measurements of the latent function.

The observations consist of accelerometer readings taken through time in an experiment on the efficacy of crash helmets. It can be seen at a glance in Figure 8 that the data set exhibits roughly 4 regimes. Firstly, between 0ms and 15ms the acceleration was negligible. Secondly, the impact slowed down the helmet, resulting in a sharp deceleration between 15ms and 28ms. Thirdly, the helmet seems to have bounced back between 28ms and 32ms, before it finally gradually slowed down and came to a stop between 32ms and 60ms. It can also be noted that the measurement noise seems to have been higher in the second half of the experiment.

We ran 50 independent random experiments, leaving out 5 points selected uniformly at random from the data set for prediction, the rest being used for training. The models we considered include the vanilla GP regression model, the *string GP* regression model with marginal maximum likelihood inference as described in Section 5.1, mixtures of independent GP experts acting on disjoint subsets of the data both for training and testing, the Bayesian committee machine (Tresp (2000)), and the robust Bayesian committee machine (Deisenroth and Ng (2015)). We considered *string GPs* with 4 and 6 strings whose boundary times are learned as part of the maximum likelihood inference. For consistency, we used the resulting partitions of the domain to define the independent experts in the competing alternatives we considered. The Matérn 3/2 kernel was used throughout. The results are reported in Table 3. To gauge the ability of each model to capture the physics of the helmets crash experiment, we have also trained all models with all data points. The results are illustrated in Figures 8 and 9.

It can be seen at a glance from Figure 9 that mixtures of independent GP experts are inappropriate for this experiment as i) the resulting posterior means exhibit discontinuities (for instance at $t = 30$ ms and $t = 40$ ms) that are inconsistent with the physics of the underlying phenomenon, and ii) they overfit the data towards the end. The foregoing discontinuities do not come as a surprise as each GP regression expert acts on a specific subset of the domain that is disjoint from the ones used by the other experts, both for training and prediction. Thus, there is no guarantee of consistency between expert predictions at the boundaries of the domain partition. Another perspective to this observation is found in noting that postulating independent GP experts, each acting on an element of a partition of the domain, is equivalent to putting as prior on the whole function a stochastic process that is discontinuous at the boundaries of the partition. Thus, the posterior stochastic process should not be expected to be continuous at the boundaries of the domain either.

This discontinuity issue is addressed by the Bayesian committee machine (BCM) and the robust Bayesian committee machine (rBCM) because, despite each independent expert being trained on a disjoint subset of the data, each expert is tasked with making predictions about all test inputs, not

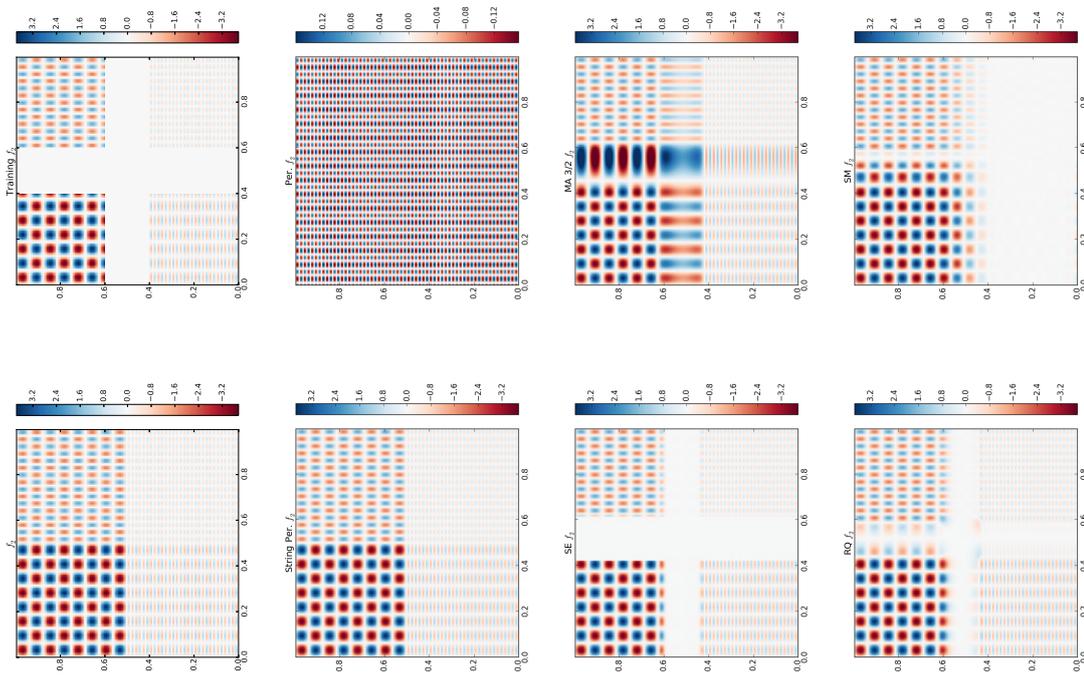


Figure 6: Extrapolation of a synthetic function f_2 (top left corner), cropped in the middle for training (top right corner), using *string GP* regression and vanilla GP regression with various popular and expressive kernels.

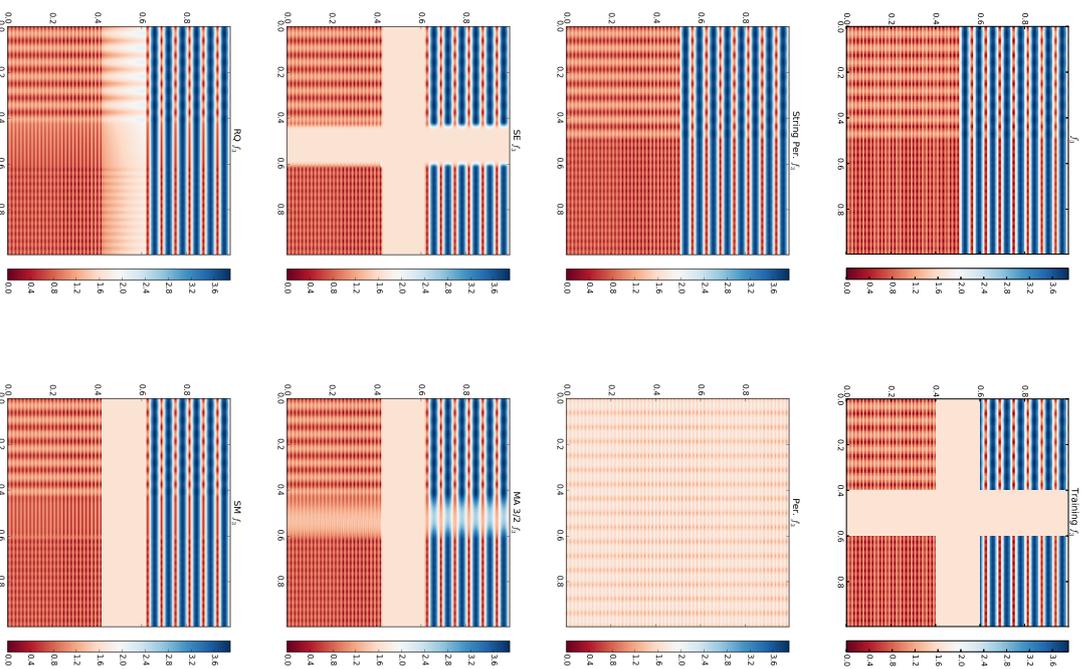


Figure 7: Extrapolation of a synthetic function f_3 (top left corner), cropped in the middle for training (top right corner), using *string GP* regression and vanilla GP regression with various popular and expressive kernels.

just the ones that fall into its input subspace. Each GP expert prediction is therefore continuous on the whole input domain,¹³ and the linear weighting schemes operated by the BCM and the rBCM on expert predictions to construct the overall predictive mean preserve continuity. However, we found that the BCM and the rBCM suffer from three pitfalls. First, we found them to be less accurate than any other alternative out-of-sample on this data set (see Table 3). Second, their predictions of latent function values are overly uncertain. This might be due to the fact that, each GP expert being trained only with training samples that lie on its input subspace, its predictions about test inputs that lie farther away from its input subspace will typically be much more uncertain, so that, despite the weighting scheme of the Bayesian committee machine putting more mass on ‘confident’ experts, overall the posterior variance over latent function values might still be much higher than in the *standard GP paradigm* for instance. This is well illustrated by both the last column of Table 3 and the BCM and rBCM plots in Figure 9. On the contrary, no *string GP* model suffers from this excess uncertainty problem. Third, the posterior means of the BCM, the rBCM and the vanilla GP regression exhibit oscillations towards the end ($t > 40$ ms) that are inconsistent with the experimental setup; the increases in acceleration as the helmet slows down suggested by these posterior means would require an additional source of energy after the bounce.

In addition to being more accurate and more certain about predictions than vanilla GP regression, the BCM and the rBCM (see Table 3), *string GP* regressions yield posterior mean acceleration profiles that are more consistent with the physics of the experiment: steady speed prior to the shock, followed by a deceleration resulting from the shock, a brief acceleration resulting from the change in direction after the bounce, and finally a smooth slow down due to the dissipation of kinetic energy. Moreover, unlike the vanilla GP regression, the BCM and the BCM, *string GP* regressions yield smaller posterior variances towards the beginning and the end of the experiment than in the middle, which is consistent with the fact that the operator would be less uncertain about the acceleration at the beginning and at the end of the experiment—one would indeed expect the acceleration to be null at the beginning and at the end of the experiment. This desirable property can be attributed to the heteroskedasticity of the noise structure in the *string GP* regression model.

We also learned the derivative of the latent acceleration with respect to time, purely from noisy acceleration measurements using the joint law of a *string GP* and its derivative (Theorem 2). This is illustrated in Figure 8.

¹³. So long as the functional prior is continuous, which is the case here.

		Prediction	
		Absolute Error	Squared Error
		Log. lik.	Pred. Std
Training			
String GP (4 strings)	-388.36 ± 0.36	15.70 ± 1.05	0.70/2.25/3.39
String GP (6 strings)	-367.21 ± 0.43	15.89 ± 1.06	475.59 ± 51.95
Vanilla GP	-420.69 ± 0.24	16.84 ± 1.09	524.18 ± 58.33
Mix. of 4 GPs	-388.37 ± 0.38	16.61 ± 1.10	512.30 ± 56.08
Mix. of 6 GPs	-369.05 ± 0.45	16.05 ± 1.11	500.43 ± 58.26
BCM with 4 GPs	-419.08 ± 0.30	17.17 ± 1.13	538.94 ± 61.91
BCM with 6 GPs	-422.15 ± 0.30	16.93 ± 1.12	533.21 ± 61.78
rBCM with 4 GPs	-419.08 ± 0.30	17.29 ± 1.11	546.95 ± 61.21
rBCM with 6 GPs	-422.15 ± 0.30	16.79 ± 1.12	542.95 ± 61.95

Table 3: Performance comparison between *string GPs*, vanilla GPs, mixture of independent GPs, the Bayesian committee machine (Tresp (2000)) and the robust Bayesian committee machine (Deisenroth and Ng (2015)) on the motorcycle data set of Silverman (1985). The Matérn 3/2 kernel was used throughout. The domain partitions were learned in the *string GP* experiments by maximum likelihood. The learned partitions were then reused to allocate data between GP experts in other models. 50 random runs were performed, each run leaving 5 data points out for testing and using the rest for training. All results (except for predictive standard deviations) are reported as average over the 50 runs ± standard error. The last column contains the minimum, average and maximum of the predictive standard deviation of the values of the latent (noise-free) function at all test points across random runs.

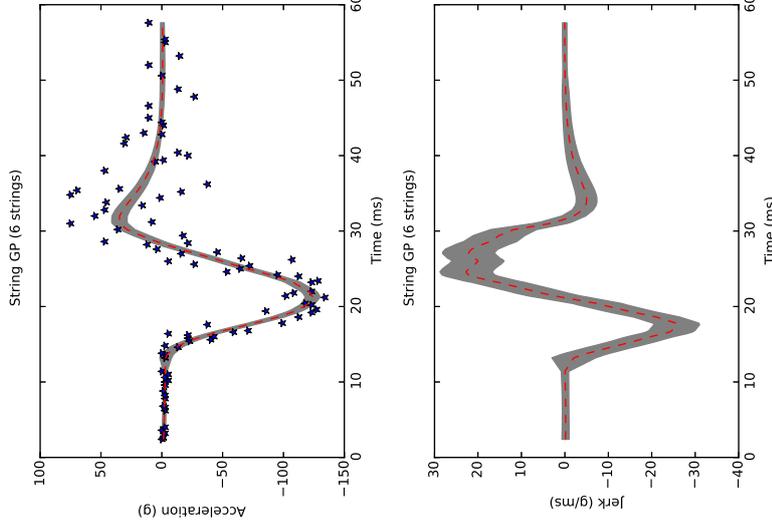


Figure 8: Posterior mean ± 2 predictive standard deviations on the motorcycle data set (see Silverman, 1985), under a Matérn 3/2 derivative *string GP* prior with 6 learned strings. The top figure shows the noisy accelerations measurements and the learned latent function. The bottom function illustrates the derivative of the acceleration with respect to time learned from noisy acceleration samples. Posterior credible bands are over the latent functions rather than noisy measurements, and as such they do not include the measurement noise.

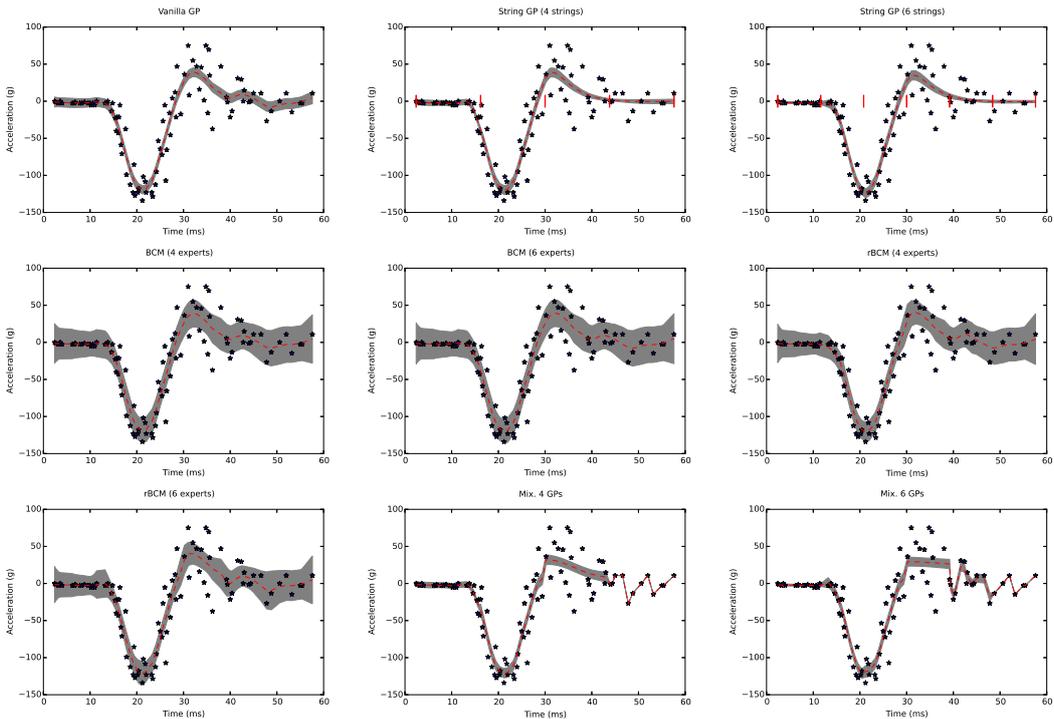


Figure 9: Bayesian nonparametric regressions on the motorcycle data set of Silverman (1985). Models compared are *string GP* regression, vanilla GP regression, mixture of independent GP regression experts on a partition of the domain, the Bayesian committee machine (BCM) and the robust Bayesian committee machine (rBCM). Domain partitions were learned during *string GP* maximum likelihood inference (red vertical bars), and reused in other experiments. Blue stars are noisy samples, red lines are posterior means of the latent function and grey bands correspond to ± 2 predictive standard deviations of the (noise-free) latent function about its posterior mean.

6.3 Large Scale Regression

To illustrate how our approach fares against competing alternatives on a standard large scale problem, we consider predicting arrival delays of commercial flights in the USA in 2008 as studied by Hensman et al. (2013). We choose the same covariates as in Hensman et al. (2013), namely the age of the aircraft (number of years since deployment), distance that needs to be covered, airline, departure time, arrival time, day of the week, day of the month and month. Unlike Hensman et al. (2013) who only considered commercial flights between January 2008 and April 2008, we consider commercial throughout the whole year, for a total of 5.93 million records. In addition to the whole data set, we also consider subsets so as to empirically illustrate the sensitivity of computational time to the number of samples. Selected subsets consist of 10,000, 100,000 and 1,000,000 records selected uniformly at random. For each data set, we use $2/3$ of the records selected uniformly at random for training and we use the remaining $1/3$ for testing. In order to level the playing field between stationary and nonstationary approaches, we normalize training and testing data sets.¹⁴ As competing alternatives to *string GPs* we consider the SVIGP of Hensman et al. (2013), the Bayesian committee machines (BCM) of Tresp (2000), and the robust Bayesian committee machines (rBCM) of Deisenroth and Ng (2015).

As previously discussed the prediction scheme operated by the BCM is Kolmogorov-inconsistent in that the resulting predictive distributions are not consistent by marginalization.¹⁵ Moreover, jointly predicting all function values by using the set of all test inputs as query set, as originally suggested in Tresp (2000), would be impractical in this experiment given that the BCM requires inverting a covariance matrix of the size of the query set which, considering the numbers of test inputs in this experiment (which we recall can be as high as 1.97 million), would be computationally intractable. To circumvent this problem we use the BCM algorithm to query one test input at a time. This approach is in-line with that adopted by Deisenroth and Ng (2015), where the authors did not address determining joint predictive distributions over multiple latent function values. For the BCM and rBCM, the number of experts is chosen so that each expert processes 200 training points. For SVIGP we use the implementation made available by the The GPy authors (2012–2016), and we use the same configuration as in Hensman et al. (2013). As for *string GPs*, we use the symmetric sum as link function, and we run two types of experiments, one allowing for inference of change-points (String GP), and the other enforcing a single kernel configuration per input dimension (String GP*). The parameters α and β are chosen so that the prior mean number of change-points in each input dimension is 5% of the number of distinct training and testing values in that input dimension, and so that the prior variance of the foregoing number of change-points is 50 times the prior mean—the aim is to be uninformative about the number of change-points. We run 10,000 iterations of our R1-MCMC sampler and discarded the first 5,000 as ‘burn-in’. After burn-in we record the states of the Markov chains for analysis using a 1-in-100 thinning rate. Predictive accuracies are reported in Table 4 and CPU time requirements¹⁶ are illustrated in Figure 10. We stress that all experiments were run on a multi-core machine, and that we prefer using the cumulative CPU clock resource

14. More precisely, we subtract from every feature sample (both in-sample and out-of-sample) the in-sample mean of the feature and we divide the result by the in-sample standard deviation of the feature.

15. For instance the predictive distribution of the value of the latent function at a test input x_1 , namely $f(x_1)$, obtained by using $\{x_1\}$ as set of test inputs in the BCM, differs from the predictive distribution obtained by using $\{x_1, x_2\}$ as set of test inputs in the BCM and then marginalising with respect to the second input x_2 .

16. We define CPU time as the cumulative CPU clock resource usage of the whole experiment (training and testing), across child processes and threads, and across CPU cores.

as time complexity metric, instead of wall-clock time, so as to be agnostic to the number of CPU cores used in the experiments. This metric has the merit of illustrating how the number of CPU cores required grows as a function of the number of training samples for a fixed/desired wall-clock execution time, but also how the wall-clock execution time grows as a function of the number of training samples for a given number of available CPU cores.

The BCM and the rBCM perform the worst in this experiment both in terms of predictive accuracy (Table 4) and total CPU time (Figure 10). The poor scalability of the BCM and the rBCM is primarily due to the testing phase. Indeed, if we denote M the total number of experts, then $M = \lceil \frac{N}{300} \rceil$, as each expert processes 200 training points, of which there are $\frac{2}{3}N$. In the prediction phase, each expert is required to make predictions about all $\frac{1}{3}N$ test inputs, which requires evaluating M products of an $\frac{1}{3}N \times 200$ matrix with a 200×200 matrix, which results in a total CPU time requirement that grows in $\mathcal{O}(M \frac{1}{3} N 200^2)$, which is the same as $\mathcal{O}(N^2)$. Given that training CPU time grows linearly in N the cumulative training and testing CPU time grows quadratically in N . This is well illustrated in Figure 10, where it can be seen that the slopes of total CPU time profiles of the BCM and the rBCM in log-log scale are approximately 2. The airline delays data set was also considered by Deisenroth and Ng (2015), but the authors restricted themselves to a fixed size of the test set of 100,000 points. However, this limitation might be restrictive as in many ‘smoothing’ applications, the test data set can be as large as the training data set—neither the BCM nor the rBCM would be sufficiently scalable in such applications.

As for SVIGP, although it was slightly more accurate than *string GPs* on this data set, it can be noted from Figure 10 that *string GPs* required 10 times less CPU resources. In fact we were unable to run the experiment on the full data set with SVIGP—we gave up after 500 CPU hours, or more than a couple of weeks wall-clock time given that the GPy implementation of SVIGP makes little use of multiple cores. As a comparison, the full experiment took 91.0 hours total CPU time (≈ 15 hours wall-clock time on our 8 cores machine) when change-points were inferred and 83.11 hours total CPU time (≈ 14 hours wall-clock time on our 8 cores machine) when change-points were not inferred. Another advantage of additively separable *string GPs* over GPs, and subsequently over SVIGP, is that they are more interpretable. Indeed, one can determine at a glance from the learned posterior mean *string GPs* of Figure 11 the effect of each of the 8 covariates considered on arrival delays. It turns out that the three most informative factors in predicting arrival delays are departure time, distance and arrival time, while the age of the aircraft, the day of the week and the day of the month seem to have little to no effect. Finally, posterior distributions of the number of change-points are illustrated in Figure 12, and posterior distributions of the locations of change-points are illustrated in Figure 13.

N	String GP	String GP*	BCM	rBCM	SVIGP
10,000	1.03 ± 0.10	1.06 ± 0.10	1.06 ± 0.10	1.06 ± 0.10	0.90 ± 0.09
100,000	0.93 ± 0.03	0.96 ± 0.03	1.66 ± 0.03	1.04 ± 0.04	0.88 ± 0.03
1,000,000	0.93 ± 0.01	0.92 ± 0.01	N/A	N/A	0.82 ± 0.01
5,929,413	0.90 ± 0.01	0.93 ± 0.01	N/A	N/A	N/A

Table 4: Predictive mean squared errors (MSEs) \pm one standard error on the airline arrival delays experiment. Squared errors are expressed as fraction of the sample variance of airline arrival delays, and hence are unitless. With this normalisation, a MSE of 1.00 is as good as using the training mean arrival delays as predictor. The * in String GP* indicates that inference was performed without allowing for change-points. N/A entries correspond to experiments that were not over after 500 CPU hours.

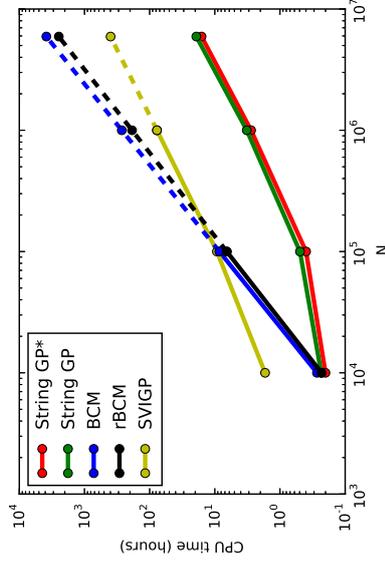


Figure 10: Total CPU time (training and testing) taken by various regression approaches on the airline delays data set as a function of the size of the subset considered, in log-log scale. The experimental setup is described in Section 6.3. The CPU time reflects actual CPU clock resource usage in each experiment, and is therefore agnostic to the number of CPU cores used. It can be regarded as the wall-clock time the experiment would have taken to complete on a single-core computer (with the same CPU frequency). Dashed lines are extrapolated values, and correspond to experiments that did not complete after 500 hours of CPU time.

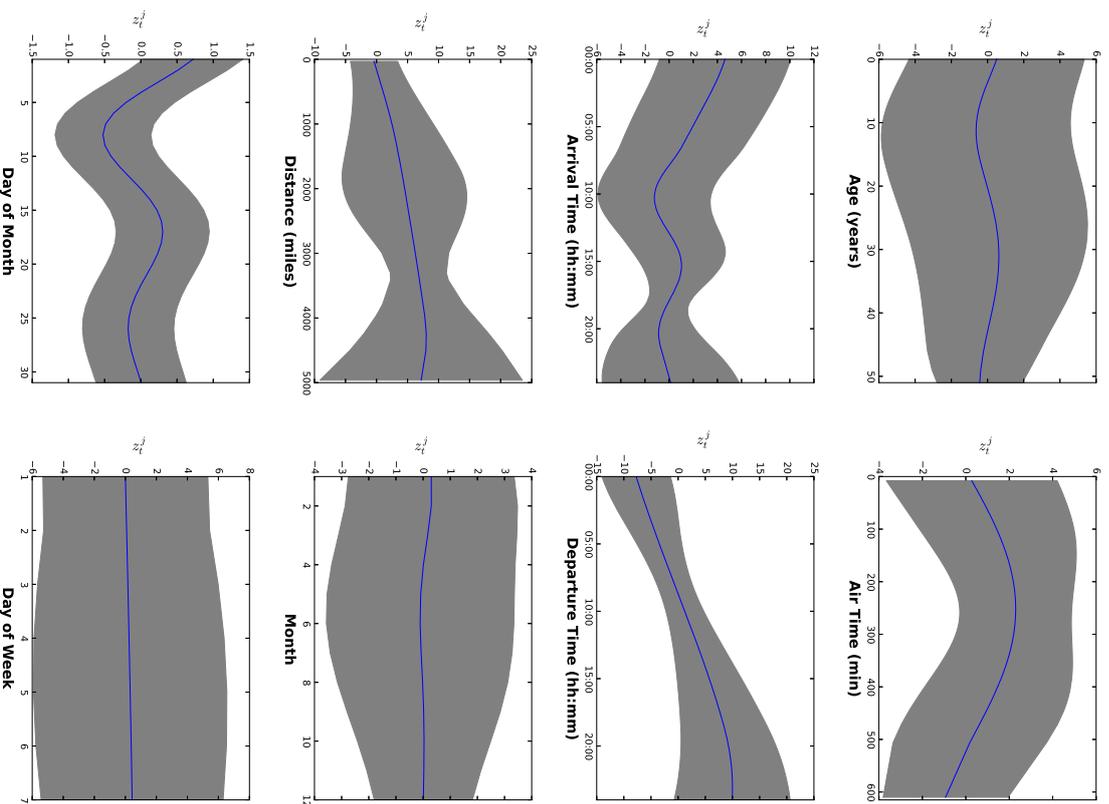


Figure 11: Posterior mean \pm one posterior standard deviation of univariate *string GPs* in the airline delays experiment of Section 6.3. Change-points were automatically inferred in this experiment

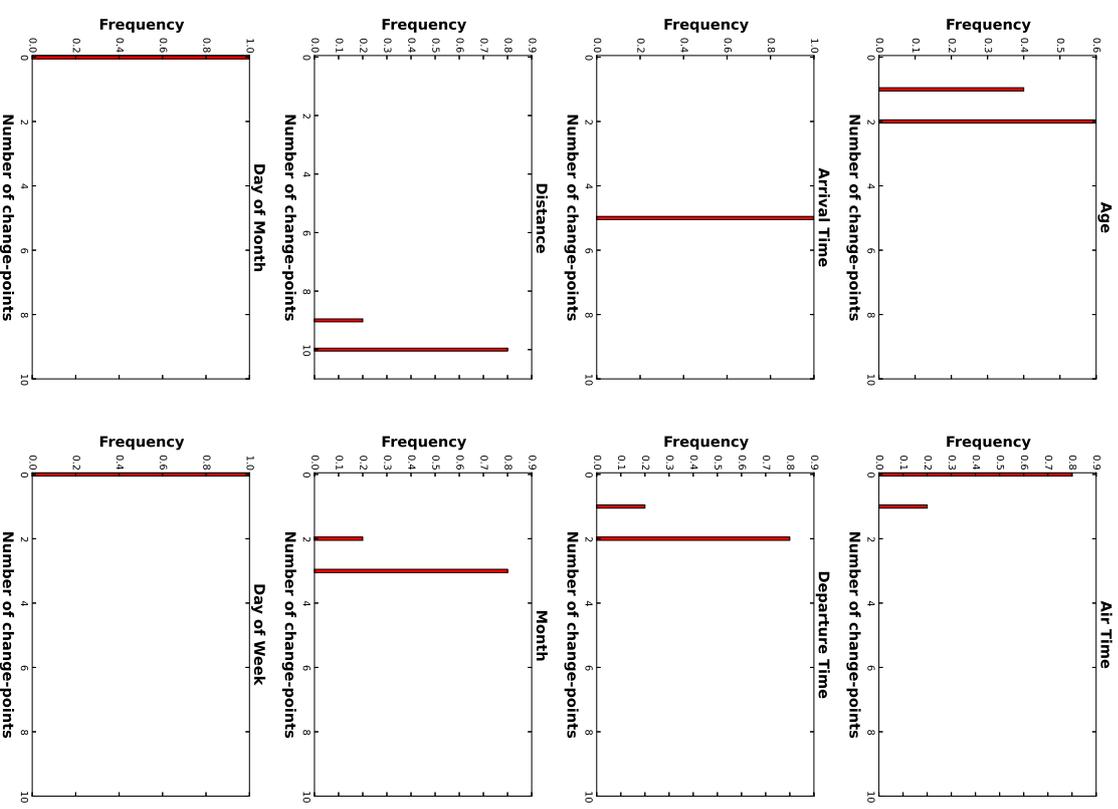


Figure 12: Posterior distributions of the numbers of change-points in each input dimension in the airline delays experiment of Section 6.3.

6.4 Large Scale Dynamic Asset Allocation

An important feature of our proposed RJ-MCMC sampler (Algorithm 2) is that, unlike the BCM, the rBCM and SVIGP, which are restricted to Bayesian nonparametric regression and classification, Algorithm 2 is agnostic with regard to the likelihood model, so long as it takes the form $p(D|\mathbf{u})$. Thus, it may be used as is on a wide variety of problems that go beyond classification and regression. In this experiment we aim to illustrate the efficacy of our approach on one such large scale problem in quantitative finance.

6.4.1 BACKGROUND

Let $(x_i(t))_{t>0}$ for $i = 1, \dots, n$ be n stock price processes. Let $(X_i(t))_{t>0}$ for $i = 1, \dots, n$ denote the market capitalisation processes, that is $X_i(t) = n_i(t)x_i(t)$ where $n_i(t)$ is the number of shares in company i trading in the market at time t . We call long-only portfolio any vector-valued stochastic process $\pi = (\pi_1, \dots, \pi_n)$ taking value on the unit simplex on \mathbb{R}^n , that is

$$\forall i, t, \pi_i(t) \geq 0 \text{ and } \sum_{i=1}^n \pi_i(t) = 1.$$

Each process π_i represents the proportion of an investor's wealth invested in (holding) shares in asset i . An example long-only portfolio is the market portfolio $\mu = (\mu_1, \dots, \mu_n)$ where

$$\mu_i(t) = \frac{X_i(t)}{X_1(t) + \dots + X_n(t)} \tag{47}$$

is the market weight of company i at time t , that is its size relative to the total market size (or that of the universe of stocks considered). The market portfolio is very important to practitioners as it is often perceived not to be subject to idiosyncracies, but only to systemic risk. It is often used as an indicator of how the stock market (or a specific universe of stocks) performs as a whole. We denote Z_π the value process of a portfolio π with initial capital $Z_\pi(0)$. That is, $Z_\pi(t)$ is the wealth at time t of an investor who had an initial wealth of $Z_\pi(0)$, and dynamically re-allocated all his wealth between the n stocks in our universe up to time t following the continuous-time strategy π .

A mathematical theory has recently emerged, namely *stochastic portfolio theory* (SPT) (see Karatzas and Fernholz, 2009) that studies the stochastic properties of the wealth processes of certain portfolios called *functionally-generated portfolio* under realistic assumptions on the market capitalisation processes $(X_i(t))_{t>0}$. Functionally-generated portfolios are rather specific in that the allocation at time t , namely $(\pi_1(t), \dots, \pi_n(t))$, solely depends on the market weights vector $(\mu_1(t), \dots, \mu_n(t))$. Nonetheless, some functionally-generated portfolios π^* have been found that, under the mild (so-called diversity) condition

$$\exists \mu_{\max}, 0 < \mu_{\max} < 1 \text{ s.t. } \forall i \leq n, t \leq T, \mu_i(t) \leq \mu_{\max}, \tag{48}$$

outperform the market portfolio over the time horizon $[0, T]$ with probability one (see Vervuurt and Karatzas, 2015; Karatzas and Fernholz, 2009). More precisely,

$$\mathbb{P}(Z_{\pi^*}(T) \geq Z_\mu(T)) = 1 \text{ and } \mathbb{P}(Z_{\pi^*}(T) > Z_\mu(T)) > 0. \tag{49}$$

Galvanized by this result, we here consider the inverse problem consisting of learning from historical market data a portfolio whose wealth process has desirable user-specified properties. This

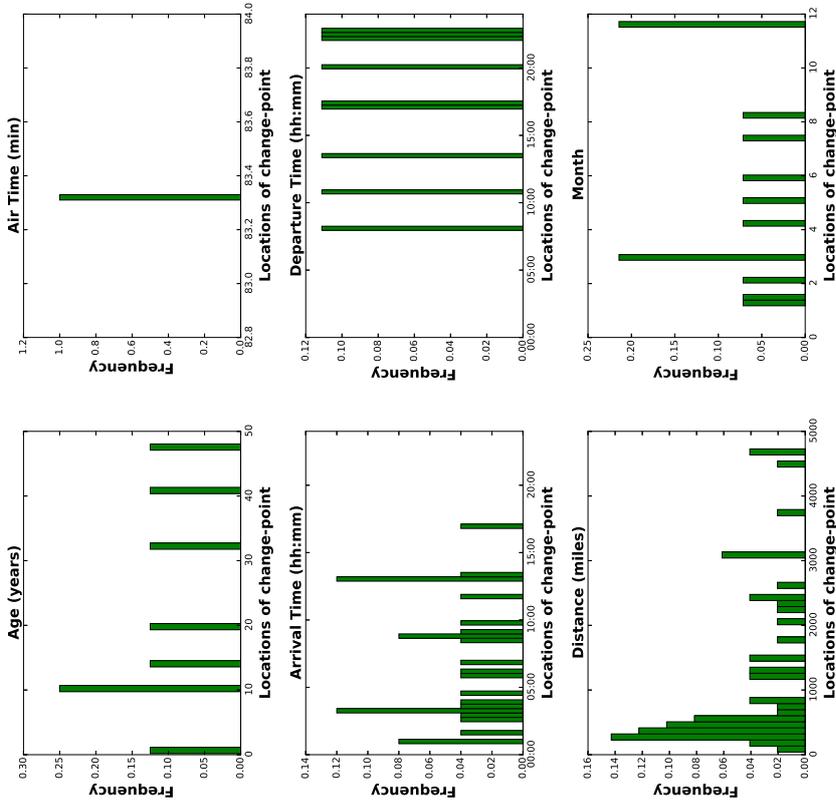


Figure 13: Posterior distributions of the locations of change-points in each input dimension in the airline delays experiment of Section 6.3. Dimensions that were learned to exhibit no change-point have been omitted here.

inverse problem is perhaps more akin to the problems faced by investment professionals: i) their benchmarks depend on the investment vehicles pitched to investors and may vary from one vehicle to another, ii) they have to take into account liquidity costs, and iii) they often find it more valuable to go beyond market weights and leverage multiple company characteristics in their investment strategies.

6.4.2. MODEL CONSTRUCTION

We consider portfolios $\pi^f = (\pi_1^f, \dots, \pi_n^f)$ of the form

$$\pi_i^f(t) = \frac{f(c_i(t))}{f(c_1(t)) + \dots + f(c_n(t))}, \quad (50)$$

where $c_i(t) \in \mathbb{R}^d$ are some quantifiable characteristics of asset i that may be observed in the market at time t , and f is a positive-valued function. Portfolios of this form include all functionally-generated portfolios studied in SPT as a special case.¹⁷ A crucial departure of our approach from the aforementioned type of portfolios is that the market characteristics processes c_i need not be restricted to size-based information, and may contain additional information such as social media sentiments, stock price path-properties, but also characteristics relative to other stocks such as performance relative to the best/worst performing stock last week/month/year etc. We place a mean-zero *string GP* prior on $\log f$. Given some historical data \mathcal{D} corresponding to a training time horizon $[0, T]$, the likelihood model $p(\mathcal{D}|\pi^f)$ is defined by the investment professional and reflects the extent to which applying the investment strategy π^f over the training time horizon would have achieved a specific investment objective. An example investment objective is to achieve a high excess return relative to a benchmark portfolio α

$$U_{\text{ER}}(\pi^f) = \log Z_{\pi^f}(T) - \log Z_{\alpha}(T). \quad (51)$$

α can be the market portfolio (as in SPT) or any stock index. Other risk-adjusted investment objectives may also be used. One such objective is to achieve a high Sharpe-ratio, defined as

$$U_{\text{SR}}(\pi^f) = \frac{\bar{r} \sqrt{252}}{\sqrt{\frac{1}{T} \sum_{t=1}^T (\bar{r}(t) - \bar{r})^2}}, \quad (52)$$

where the time t is in days, $\bar{r}(t) := \log Z_{\pi^f}(t) - \log Z_{\pi^f}(t-1)$ are the daily returns the portfolio π^f and $\bar{r} = \frac{1}{T} \sum_{t=1}^T \bar{r}(t)$ its average daily return. More generally, denoting $\mathcal{U}(\pi^f)$ the performance of the portfolio π^f over the training horizon $[0, T]$ (as per the user-defined investment objective), we may choose as likelihood model a distribution over $\mathcal{U}(\pi^f)$ that reflects what the investment professional considers good and bad performance. For instance, in the case of the excess return relative to a benchmark portfolio or the Sharpe ratio, we may choose $\mathcal{U}(\pi^f)$ to be supported on $]0, +\infty[$ (for instance $\mathcal{U}(\pi^f)$ can be chosen to be Gamma distributed) so as to express that portfolios that do not outperform the benchmark or loose money overall in the training data are not of interest. We may then choose the mean and standard deviation of the Gamma distribution based on our

expectation as to what performance a good candidate portfolio can achieve, and how confident we feel about this expectation. Overall we have,

$$p(\mathcal{D}|\pi^f) = \gamma(\mathcal{U}(\pi^f); \alpha_{\epsilon_1}, \beta_{\epsilon_2}), \quad (53)$$

where $\gamma(\cdot; \alpha, \beta)$ is the probability density function of the Gamma distribution. Noting, from Equation (50) that $\pi_i^f(t)$ only depends on f through its values at $(c_1(t), \dots, c_n(t))$, and assuming that $\mathcal{U}(\pi^f)$ only depends on π^f evaluated at a finite number of times (as it is the case for excess returns and the Sharpe ratio), it follows that $\mathcal{U}(\pi^f)$ only depends on \mathbf{f} , a vector of values of f at a finite number of points. Hence the likelihood model, which we may rewrite as

$$p(\mathcal{D}|\mathbf{f}) = \gamma(\mathcal{U}(\pi_i^f); \alpha_{\epsilon_1}, \beta_{\epsilon_2}), \quad (54)$$

is of the form required by the RL-MCMC sampler previously developed. By sampling from the posterior distribution $p(\mathbf{f}, \mathbf{f}^*, \mathbb{V}\mathbf{f}, \mathbb{V}\mathbf{f}^*|\mathcal{D})$, the hope is to learn a portfolio that did well during the training horizon, to analyse the sensitivity of its investment strategy to the underlying market characteristics through the gradient of f , and to evaluate the learned investment policy on future market conditions.

6.4.3. EXPERIMENTAL SETUP

The universe of stocks was considered for this experiment are the constituents of the S&P 500 index, accounting for changes in constituents over time and corporate events. We used the period 1st January 1990 to 31st December 2004 for training and we tested the learned portfolio during the period 1st January 2005 to 31st December 2014. We rebalanced the portfolio daily, for a total of 2,52 million input points at which the latent function f must be learned. We considered as market characteristics the market weight (CAP), the latest return on asset (ROA) defined as the ratio between the net yearly income and the total assets as per the latest balance sheet of the company known at the time of investment, the previous close-to-close return (PR), the close-to-close return before the previous (PR2), and the S&P long and short term credit rating (LCR and SCR). While the market weight is a company size characteristic, the ROA reflects how well a company performs relative to its size, and we hope that S&P credit ratings will help further discriminate successful companies from others. The close-to-close returns are used to learn possible ‘momentum’ patterns from the data. The data originate from the CRSP and Compustat databases. In the experiments we considered as performance metric the annualised excess return $U_{\text{ER-EMP}}$ relative to the equally-weighted portfolio. We found the equally-weighted portfolio to be a harder benchmark to outperform than the market portfolio. We chose α_{ϵ} and β_{ϵ} in Equation (54) so that the mean of the Gamma distribution is 10.0 and its variance 0.5, which expresses a very greedy investment target.

It is worth pointing out that none of the scalable GP alternatives previously considered can cope with our likelihood model Equation (54). We compared the performance of the learned *string GP* portfolio out-of-sample to those of the best three SPT portfolios studied in [Vervuurt and Karatzas \(2015\)](#), namely the equally weighted portfolio

$$\pi_i^{\text{EMP}}(t) = \frac{1}{n}, \quad (55)$$

and the diversity weighted portfolios

$$\pi_i^{\text{DWP}}(t; p) = \frac{\mu_i(t)^p}{\mu_1(t)^p + \dots + \mu_n(t)^p}, \quad (56)$$

¹⁷ We refer the reader to [Karatzas and Fernholz \(2009\)](#) for the definition of functionally-generated portfolios.

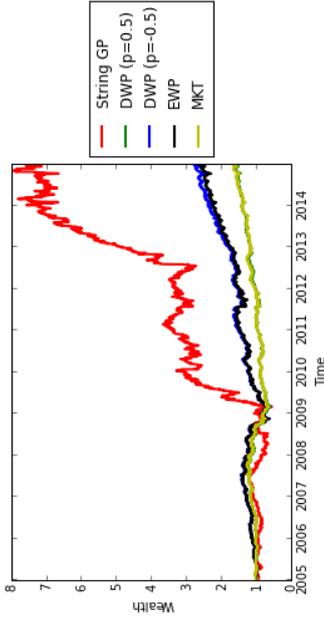


Figure 14: Evolution of the wealth processes of various long-only trading strategies on the S&P 500 universe of stocks between 1st January 2005 (where we assume a starting wealth of 1) and 31st December 2014. The String GP strategy was learned using market data from 1st January 1990 to 31st December 2004 as described in Section 6.4. EWP refers to the equally-weighted portfolio, MKT refers to the market portfolio (which weights stocks proportionally to their market capitalisations) and DWP (p) refers to the diversity-weighted portfolio with exponent p (which weights stocks proportionally to the p -th power of their market weights).

with parameter p equals to -0.5 and 0.5 , and the market portfolio. Results are provided Table 5, and Figure 14 displays the evolution of the wealth process of each strategy. It can be seen that the learned *string GP* strategy considerably outperforms the next best SPT portfolio. This experiment not only demonstrates (once more) that *string GPs* scale to large scale problems, it also illustrates that our inference scheme is able to unlock commercial value in new intricate large scale applications where no alternative is readily available. In effect, this application was first introduced by Kom Samo and Vervuurt (2016), where the authors used a Gaussian process prior on a Cartesian grid and under a separable covariance function so as to speed up inference with Kronecker techniques. Although the resulting inference scheme has time complexity that is linear in the total number of points on the grid, for a given grid resolution, the time complexity grows *exponentially* in the dimension of the input space (that is the number of trading characteristics), which is impractical for $d \geq 4$. On the other hand, *string GPs* allow for a time complexity that grows linearly with the number of trading characteristics, thereby enabling the learning of subtler market inefficiencies from the data.

Strategy	Sharpe Ratio	$Z_\pi(T)/Z_{\text{EWP}}(T)$	Avg. Ann. Ret.
String GP	0.73	2.87	22.07%
DWP ($p = -0.5$)	0.55	1.07	10.56%
EWP	0.53	1.00	9.84%
MKT	0.34	0.62	4.77%
DWP ($p = 0.5$)	0.33	0.61	4.51%

Table 5: Performance of various long-only trading strategies on the S&P 500 universe of stocks between 1st January 2005 (where we assume a starting wealth of 1) and 31st December 2014. The String GP strategy was learned using market data from 1st January 1990 to 31st December 2004 as described in Section 6.4. EWP refers to the equally-weighted portfolio, MKT refers to the market portfolio (which weights stocks proportionally to their market capitalisations) and DWP (p) refers to the diversity-weighted portfolio with exponent p (which weights stocks proportionally to the p -th power of the market weight of the asset). $Z_\pi(T)$ denotes the terminal wealth of strategy π , and Avg. Ann. Ret. is the strategy's equivalent constant annual return over the test horizon.

7. Discussion

In this paper, we introduce a novel class of smooth functional priors (or stochastic processes), which we refer to as *string GPs*, with the aim of simultaneously addressing the lack of scalability and the lack of flexibility of Bayesian kernel methods. Unlike existing approaches, such as Gaussian process priors (Rasmussen and Williams (2006)) or student-t process priors (Shah et al. (2014)), which are parametrised by *global* mean and covariance functions, and which postulate *fully dependent* finite-dimensional marginals, the alternative construction we propose adopts a *local* perspective and the resulting finite-dimensional marginals exhibit *conditional independence* structures. Our local approach to constructing *string GPs* provides a principled way of postulating that the latent function we wish to learn might exhibit locally homogeneous patterns, while the conditional independence structures constitute the core ingredient needed for developing scalable inference methods. Moreover, we provide theoretical results relating our approach to Gaussian processes, and we illustrate that our approach can often be regarded as a more scalable and/or more flexible extension. We argue and empirically illustrate that *string GPs* present an unparalleled opportunity for learning local patterns in small scale regression problems using nothing but standard Gaussian process regression techniques. More importantly, we propose a novel *scalable* RJ-MCMC inference scheme to learn latent functions in a wide variety of machine learning tasks, while simultaneously determining *whether* the data set exhibits local patterns, *how many* types of local patterns the data might exhibit, and *where* do changes in these patterns are likely to occur. The proposed scheme has time complexity and memory requirement that are both *linear* in the sample size N . When the number of available computing cores is at least equal to the dimension d of the input space, the time complexity is independent from the dimension of the input space. Else, the time complexity grows in $\mathcal{O}(dN)$. The memory requirement grows in $\mathcal{O}(dN)$. We empirically illustrate that our approach scales considerably better than competing alternatives on a standard benchmark data set, and is able to process data sizes that competing approaches cannot handle in a reasonable time.

7.1 Limitations

The main limitation of our approach is that, unlike the *standard GP paradigm* in which the time complexity of marginal likelihood evaluation does not depend on the dimension of the input space (other than through the evaluation of the Gram matrix), the *string GP paradigm* requires a number of computing cores that increases linearly with the dimension of the input space, or alternatively has a time complexity linear in the input space dimension on single-core machines. This is a by-product of the fact that in the *string GP paradigm*, we jointly infer the latent function and its gradient. If the gradient of the latent function is inferred in the *standard GP paradigm*, the resulting complexity will also be linear in the input dimension. That being said, overall our RJ-MCMC inference scheme will typically scale better per iteration to large input dimensions than gradient-based marginal likelihood inference in the *standard GP paradigm*, as the latter typically requires numerically evaluating an Hessian matrix, which requires computing the marginal likelihood a number of times per iterative update that grows quadratically with the input dimension. In contrast, a Gibbs cycle in our MCMC sampler has worst case time complexity that is linear in the input dimension.

7.2 Extensions

Some of the assumptions we have made in the construction of *string GPs* and *membrane GPs* can be relaxed, which we consider in detail below.

7.2.1 STRONGER GLOBAL REGULARITY

We could have imposed more (multiple continuous differentiability) or less (continuity) regularity as boundary conditions in the construction of *string GPs*. We chose continuous differentiability as it is a relatively mild condition guaranteed by most popular kernels, and yet the corresponding treatment can be easily generalised to other regularity requirements. It is also possible to allow for discontinuity at a boundary time a_k by replacing μ_k^b and Σ_k^b in Equation (5) with ${}_k M_{a_k}$ and ${}_k \mathbf{K}_{a_k, a_k}$ respectively, or equivalently by preventing any communication between the k -th and the $(k+1)$ -th strings. This would effectively be equivalent to having two independent *string GPs* on $[a_0, a_k]$ and $[a_k, a_k]$.

7.2.2 DIFFERENTIAL OPERATORS AS LINK FUNCTIONS

Our framework can be further extended to allow differential operators as link functions, thereby considering the latent multivariate function to infer as the response of a differential system to independent univariate *string GP* excitations. The RJ-MCMC sampler we propose in Section 5 would still work in this framework, with the only exception that, when the differential operator is of first order, the latent multivariate function will be continuous but not differentiable, except if global regularity is upgraded as discussed above. Moreover, Proposition 5 can be generalised to first order linear differential operators.

7.2.3 DISTRIBUTED STRING GPs

The RJ-MCMC inference scheme we propose may be easily adapted to handle applications where the data set is so big that it has to be stored across multiple clusters, and inference techniques have to be developed as data flow graphs¹⁸ (for instance using libraries such as TensorFlow).

To do so, the choice of string boundary times can be adapted so that each string has the same number of inner input coordinates, and such that in total there are as many strings across dimensions as a target number of available computing cores. We may then place a prior on kernel memberships similar to that of previous sections. Here, the change-points may be restricted to coincide with boundary times, and we may choose priors such that the sets of change-points are independent between input dimensions. In each input dimension the prior on the number of change-points can be chosen to be a truncated Poisson distribution (truncated to never exceed the total number of boundary times), and conditional on their number we may choose change-points to be uniformly distributed in the set of boundary times. In so doing, any two strings whose shared boundary time is not a change-point will be driven by the same kernel configuration.

This new setup presents no additional theoretical or practical challenges, and the RJ-MCMC techniques previously developed are easily adaptable to jointly learn change-points and function values. Unlike the case we developed in previous sections where an update of the univariate *string GP* corresponding to an input dimension, say the j -th, requires looping through all distinct j -th input coordinates, here no step in the inference scheme requires a full view of the data set in any input dimension. Full RJ-MCMC inference can be constructed as a data flow graph. An example such graph is constructed as follows. The leaves correspond to computing cores responsible for generating change-points and kernel configurations, and mapping strings to kernel configurations. The following layer is made of compute cores that use kernel configurations coming out of the previous layer to sequentially compute boundary conditions corresponding to a specific input dimension—there are d such compute cores, where d is the input dimension. These compute cores then pass computed boundary conditions to subsequent compute cores we refer to as string compute cores. Each string compute core is tasked with computing *derivative string GP* values for a specific input dimension and for a specific string in that input dimension, conditional on previously computed boundary conditions. These values are then passed to a fourth layer of compute cores, each of which being tasked with computing function and gradient values corresponding to a small subset of training inputs from previously computed *derivative string GP* values. The final layers then computes the log-likelihood using a distributed algorithm such as Map-Reduce when possible. This proposal data flow graph is illustrated Figure 15.

We note that the approaches of Kim et al. (2005), Gramacy and Lee (2008), Tresp (2000), and Deisenroth and Ng (2015) also allow for fully-distributed inference on regression problems. Distributed *string GP* RJ-MCMC inference improves on these in that it places little restriction on the type of likelihood. Moreover, unlike Kim et al. (2005) and Gramacy and Lee (2008) that yield discontinuous latent functions, *string GPs* are continuously differentiable, and unlike Tresp (2000) and Deisenroth and Ng (2015), local experts in the *string GP paradigm* (i.e. strings) are driven by possibly different sets of hyper-parameters, which facilitates the learning of local patterns.

¹⁸ A data flow graph is a computational (directed) graph whose nodes represent calculations (possibly taking place on different computing units) and directed edges correspond to data flowing between calculations or computing units.

7.2.4 APPROXIMATE MCMC FOR I.I.D. OBSERVATIONS LIKELIHOODS

As discussed in Section 5.2, the bottleneck of our proposed inference scheme is the evaluation of likelihood. When the likelihood factorises across training samples, the linear time complexity of our proposed approach can be further reduced using a Monte Carlo approximation of the log-likelihood (see for instance [Bardenet et al. \(2014\)](#) and references therein). Although the resulting Markov chain will typically not converge to the true posterior distribution, in practice its stationary distribution can be sufficiently close to the true posterior when reasonable Monte Carlo sample sizes are used. Convergence results of such approximations have recently been studied by [Bardenet et al. \(2014\)](#) and [Alquier et al. \(2016\)](#). We expect this extension to speed-up inference when the number of compute cores is in the order of magnitude of the input dimension, but we would recommend the previously mentioned fully-distributed string GP inference extension when compute cores are not scarce.

7.2.5 VARIATIONAL INFERENCE

It would be useful to develop suitable variational methods for inference under *string GP* priors, that we hope will scale similarly to our proposed RJ-MCMC sampler but will converge faster. We anticipate that the main challenge here will perhaps be the learning of model complexity, that is the number of distinct kernel configurations in each input dimension.

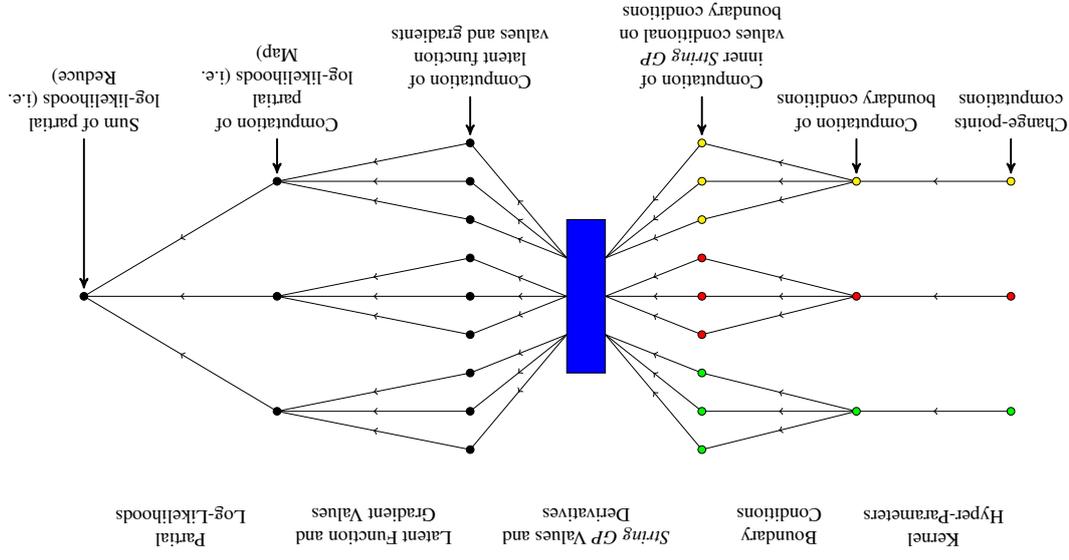


Figure 15: Example data flow graph for fully-distributed *string GP* inference under an i.i.d observations likelihood model. Here the input space is three-dimensional to ease illustration. Filled circles represent compute cores, and edges correspond to flows of data. Compute cores with the same colour (green, red or yellow) perform operations pertaining to the same input dimension, while black-filled circles represent compute cores performing cross-dimensional operations. The blue rectangle plays the role of a hub that relays *string GP* values to the compute cores that need them to compute the subset of latent function values and gradients they are responsible for. These values are then used to compute the log-likelihood in a distributed fashion using the Map-Reduce algorithm. Each calculation in the corresponding RJ-MCMC sampler would be initiated at one of the compute cores, and would trigger updates of all edges accessible from that compute core.

Acknowledgments

Yves-Laurent is a Google Fellow in Machine Learning and would like to acknowledge support from the Oxford-Man Institute. Wharton Research Data Services (WRDS) was used in preparing the data for Section 6.4 of this paper. This service and the data available thereon constitute valuable intellectual property and trade secrets of WRDS and/or its third-party suppliers.

Appendix A.

We begin by recalling Kolmogorov's extension theorem, which we will use to prove the existence of derivative Gaussian processes and string Gaussian processes.

Theorem 7 (Kolmogorov's extension theorem, (Øksendal, 2003, Theorem 2.1.5))

Let I be an interval, let all $t_1, \dots, t_i \in I$, $i, n \in \mathbb{N}^*$, let ν_{t_1, \dots, t_i} be probability measures on \mathbb{R}^{ni} such that:

$$\nu_{\pi(1), \dots, \pi(i)}(F_{\pi(1)}, \dots, F_{\pi(i)}) = \nu_{t_1, \dots, t_i}(F_{t_1}, \dots, F_{t_i}) \quad (57)$$

for all permutations π on $\{1, \dots, i\}$ and

$$\nu_{t_1, \dots, t_i}(F_{t_1}, \dots, F_{t_i}) = \nu_{t_1, \dots, t_i, t_{i+1}, \dots, t_{i+m}}(F_{t_1}, \dots, F_{t_i}, \mathbb{R}^n, \dots, \mathbb{R}^n) \quad (58)$$

for all $m \in \mathbb{N}^*$ where the set on the right hand side has a total of $i + m$ factors. Then there exists a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and an \mathbb{R}^n -valued stochastic process $(X_t)_{t \in I}$ on Ω ,

$$X_t : \Omega \rightarrow \mathbb{R}^n$$

such that

$$\nu_{t_1, \dots, t_i}(F_{t_1}, \dots, F_{t_i}) = \mathbb{P}(X_{t_1} \in F_{t_1}, \dots, X_{t_i} \in F_{t_i}) \quad (59)$$

for all $t_1, \dots, t_i \in I$, $i \in \mathbb{N}^*$ and for all Borel sets F_1, \dots, F_i .

It is easy to see that every stochastic process satisfies the permutation and marginalisation conditions (57) and (58). The power of Kolmogorov's extension theorem is that it states that those two conditions are sufficient to guarantee the existence of a stochastic process.

Appendix B. Proof of Proposition 1

In this section we prove Proposition 1, which we recall below.

Proposition 1 (Derivative Gaussian processes)

Let I be an interval, $k : I \times I \rightarrow \mathbb{R}$ a C^2 symmetric positive semi-definite function,¹⁹ $m : I \rightarrow \mathbb{R}$ a C^1 function.

(A) There exists a \mathbb{R}^2 -valued stochastic process $(D_t)_{t \in I}$, $D_t = (z_t, z'_t)$, such that for all $t_1, \dots, t_n \in I$,

$$(z_{t_1}, \dots, z_{t_n}, z'_{t_1}, \dots, z'_{t_n})$$

¹⁹ C^1 (resp. C^2) functions denote functions that are once (resp. twice) continuously differentiable on their domains.

is a Gaussian vector with mean

$$\begin{pmatrix} m(t_1), \dots, m(t_n), \frac{dm}{dt}(t_1), \dots, \frac{dm}{dt}(t_n) \end{pmatrix}$$

and covariance matrix such that

$$\text{cov}(z_{t_i}, z_{t_j}) = k(t_i, t_j), \quad \text{cov}(z_{t_i}, z'_{t_j}) = \frac{\partial k}{\partial y}(t_i, t_j), \quad \text{and} \quad \text{cov}(z'_{t_i}, z'_{t_j}) = \frac{\partial^2 k}{\partial x \partial y}(t_i, t_j).$$

We herein refer to $(D_t)_{t \in I}$ as a **derivative Gaussian process**.

(B) $(z_t)_{t \in I}$ is a Gaussian process with mean function m , covariance function k and that is C^1 in the L^2 (mean square) sense.

(C) $(z'_t)_{t \in I}$ is a Gaussian process with mean function $\frac{dm}{dt}$ and covariance function $\frac{\partial^2 k}{\partial x \partial y}$. Moreover, $(z'_t)_{t \in I}$ is the L^2 derivative of the process $(z_t)_{t \in I}$.

Proof

Appendix B.1 Proof of Proposition 1 (A)

Firstly, we need to show that the matrix suggested in the proposition as the covariance matrix of $(z_{t_1}, \dots, z_{t_n}, z'_{t_1}, \dots, z'_{t_n})$ is indeed positive semi-definite. To do so, we will show that it is the limit of positive definite matrices (which is sufficient to conclude it is positive semi-definite, as $x^T M_n x \geq 0$ for a convergent sequence of positive definite matrices implies $x^T M_\infty x \geq 0$).

Let k be as in the proposition, h such that $\forall i \leq n$, $t_i + h \in I$ and $(\tilde{z}_i)_{i \in I}$ be a Gaussian process with covariance function k . The vector

$$\begin{pmatrix} \tilde{z}_{t_1}, \dots, \tilde{z}_{t_n}, \frac{\tilde{z}_{t_1+h} - \tilde{z}_{t_1}}{h}, \dots, \frac{\tilde{z}_{t_n+h} - \tilde{z}_{t_n}}{h} \end{pmatrix}$$

is a Gaussian vector whose covariance matrix is positive definite and such that

$$\text{cov}(\tilde{z}_{t_i}, \tilde{z}_{t_j}) = k(t_i, t_j), \quad (60)$$

and

$$\text{cov}\left(\tilde{z}_{t_i}, \frac{\tilde{z}_{t_i+h} - \tilde{z}_{t_i}}{h}\right) = \frac{k(t_i, t_j + h) - k(t_i, t_j)}{h}, \quad (61)$$

$$\begin{aligned} & \text{cov}\left(\frac{\tilde{z}_{t_i+h} - \tilde{z}_{t_i}}{h}, \frac{\tilde{z}_{t_j+h} - \tilde{z}_{t_j}}{h}\right) \\ &= \frac{1}{h^2} (k(t_i + h, t_j + h) - k(t_i + h, t_j) - k(t_i, t_j + h) + k(t_i, t_j)). \end{aligned} \quad (62)$$

As k is C^2 , $h \rightarrow k(x, y + h)$ admits a second order Taylor expansion about $h = 0$ for every x , and we have:

$$k(x, y + h) = k(x, y) + \frac{\partial k}{\partial y}(x, y)h + \frac{1}{2} \frac{\partial^2 k}{\partial y^2}(x, y)h^2 + o(h^2) = k(y + h, x). \quad (63)$$

Similarly, $h \rightarrow k(x+h, y+h)$ admits a second order Taylor expansion about $h=0$ for every x, y and we have:

$$k(x+h, y+h) = k(x, y) + \left[\frac{\partial k}{\partial x}(x, y) + \frac{\partial k}{\partial y}(x, y) \right] h + \left[\frac{\partial^2 k}{\partial x \partial y}(x, y) + \frac{1}{2} \frac{\partial^2 k}{\partial x^2}(x, y) \right. \\ \left. + \frac{1}{2} \frac{\partial^2 k}{\partial y^2}(x, y) \right] h^2 + o(h^2). \quad (64)$$

Hence,

$$k(t_i, t_j+h) - k(t_i, t_j) = \frac{\partial k}{\partial y}(t_i, t_j)h + o(h), \quad (65)$$

and

$$k(t_i+h, t_j+h) - k(t_i+h, t_j) - k(t_i, t_j+h) + k(t_i, t_j) = \frac{\partial^2 k}{\partial x \partial y}(t_i, t_j)h^2 + o(h^2). \quad (66)$$

Dividing Equation (65) by h , dividing Equation (66) by h^2 , and taking the limits, we obtain:

$$\lim_{h \rightarrow 0} \text{cov} \left(\frac{\tilde{z}_{t_i} + h - \tilde{z}_{t_j}}{h}, \frac{\tilde{z}_{t_j} + h - \tilde{z}_{t_i}}{h} \right) = \frac{\partial k}{\partial y}(t_i, t_j),$$

and

$$\lim_{h \rightarrow 0} \text{cov} \left(\frac{\tilde{z}_{t_i+h} - \tilde{z}_{t_i}}{h}, \frac{\tilde{z}_{t_j+h} - \tilde{z}_{t_j}}{h} \right) = \frac{\partial^2 k}{\partial x \partial y}(t_i, t_j),$$

which corresponds to the covariance structure of Proposition 1. In other words the proposed covariance structure is indeed positive semi-definite.

Let ν_{t_1, \dots, t_n}^N be the Gaussian probability measure corresponding to the joint distribution of $(z_{t_1}, \dots, z_{t_n}, z'_{t_1}, \dots, z'_{t_n})$ as per the Proposition 1, and let ν_{t_1, \dots, t_n}^D be the measure on the Borel σ -algebra $\mathcal{B}(\mathbb{R}^2 \times \dots \times \mathbb{R}^2)$ such that for any $2n$ intervals $I_{11}, I_{12}, \dots, I_{n1}, I_{n2}$,

$$\nu_{t_1, \dots, t_n}^D(I_{11} \times I_{12} \times \dots \times I_{n1} \times I_{n2}) := \nu_{t_1, \dots, t_n}^N(I_{11}, \dots, I_{n1}, I_{12}, \dots, I_{n2}). \quad (67)$$

The measures ν_{t_1, \dots, t_n}^D are the finite dimensional measures corresponding to the stochastic object $(D_t)_{t \in I}$ sampled at times t_1, \dots, t_n . They satisfy the time permutation and marginalisation conditions of Kolmogorov's extension theorem as the Gaussian measures ν_{t_1, \dots, t_n}^N do. Hence, the \mathbb{R}^2 -valued stochastic process $(D_t)_{t \in I}$ defined in Proposition 1 does exist.

Appendix B.2 Proof of Proposition 1 (B)

That $(z_t)_{t \in I}$ is a Gaussian process results from the fact that the marginals $(z_{t_1}, \dots, z_{t_n})$ are Gaussian vectors with mean $(m(t_1), \dots, m(t_n))$ and covariance matrix $[k(t_i, t_j)]_{i, j \in [1, n]}$. The fact that $(z_t)_{t \in I}$ is \mathcal{C}^1 in the L^2 sense is a direct consequence of the twice continuous differentiability of k .

Appendix B.3 Proof of Proposition 1 (C)

In effect, it follows from Proposition 1(A) that $\frac{z_{t+h} - z_t}{h} - z'_t$ is a Gaussian random variable with mean

$$\frac{m(t+h) - m(t)}{h} - \frac{dm}{dt}(t)$$

and variance

$$\frac{k(t+h, t+h) - 2k(t+h, t) + k(t, t) - 2\frac{\partial k}{\partial y}(t+h, t)h + 2\frac{\partial k}{\partial y}(t, t)h + \frac{\partial^2 k}{\partial x \partial y}(t, t)h^2}{h^2}.$$

Taking the second order Taylor expansion of the numerator in the fraction above about $h=0$ we get $o(h^2)$, hence

$$\lim_{h \rightarrow 0} \text{Var} \left(\frac{z_{t+h} - z_t}{h} - z'_t \right) = 0.$$

We also have

$$\lim_{h \rightarrow 0} \mathbb{E} \left(\frac{z_{t+h} - z_t}{h} - z'_t \right) = \frac{dm}{dt}(t) - \mathbb{E}(z'_t) = 0.$$

Therefore,

$$\lim_{h \rightarrow 0} \mathbb{E} \left[\left(\frac{z_{t+h} - z_t}{h} - z'_t \right)^2 \right] = 0,$$

which proves that (z'_t) is the L^2 derivative of (z_t) . The fact that (z'_t) is a Gaussian process with mean function $\frac{dm}{dt}$ and covariance function $\frac{\partial^2 k}{\partial x \partial y}$ is a direct consequence of the distribution of the marginals $(z'_{t_1}, \dots, z'_{t_n})$. Moreover, the continuity of (z'_t) in the L^2 sense is a direct consequence of the continuity of $\frac{\partial^2 k}{\partial x \partial y}$ (see Rasmussen and Williams, 2006, p. 81 4.1.1). ■

Appendix C. Proof of Theorem 2

In this section we prove Theorem 2 which we recall below.

Theorem 2 (String Gaussian process)

Let $a_0 < \dots < a_k < \dots < a_K < \dots < a_K$ and let $p_N(x; \mu, \Sigma)$ be the multivariate Gaussian density with mean vector μ and covariance matrix Σ . Furthermore, let $(m_k : [a_{k-1}, a_k] \rightarrow \mathbb{R})_{k \in [1, K]}$ be \mathcal{C}^1 functions, and $(k_k : [a_{k-1}, a_k] \times [a_{k-1}, a_k] \rightarrow \mathbb{R})_{k \in [1, K]}$ be \mathcal{C}^3 symmetric positive semi-definite functions, neither degenerate at a_{k-1} , nor degenerate at a_k given a_{k-1} .

(A) There exists an \mathbb{R}^2 -valued stochastic process $(SD_t)_{t \in I}$, $SD_t = (z_t, z'_t)$ satisfying the following conditions:

1) The probability density of $(SD_{a_0}, \dots, SD_{a_K})$ reads:

$$p_0(x_0, \dots, x_K) := \prod_{k=0}^K p_N \left(x_k; \mu_k^b, \Sigma_k^b \right)$$

where: $\Sigma_0^b = \mathbf{1} \mathbf{K}_{a_0; a_0}$, $\forall k > 0$ $\Sigma_k^b = \mathbf{K}_{a_k; a_k} - \mathbf{K}_{a_k; a_{k-1}} \mathbf{K}_{a_{k-1}; a_{k-1}}^{-1} \mathbf{K}_{a_{k-1}; a_k}^T$,

$$\begin{aligned} \mu_0^k &= \mathbf{1}_{\mathbf{M}_{a_0}}, \quad \forall k > 0 \quad \mu_k^k = k \mathbf{M}_{a_k} + k \mathbf{K}_{a_k; a_{k-1}} \mathbf{K}_{a_{k-1}}^{-1} (x_{k-1} - k \mathbf{M}_{a_{k-1}}), \\ &\text{with } k \mathbf{K}_{uv} = \begin{bmatrix} k_k(u, v) & \frac{\partial k_k}{\partial u}(u, v) \\ \frac{\partial k_k}{\partial x}(u, v) & \frac{\partial^2 k_k}{\partial x \partial y}(u, v) \end{bmatrix}, \quad \text{and } k \mathbf{M}_u = \begin{bmatrix} m_k(u) \\ \frac{dm_k}{du}(u) \end{bmatrix}. \end{aligned}$$

2) Conditional on $(SD)_{a_k} = x_{a_k} \in [0, K_1]$, the restrictions $(SD)_{t_i \in [a_{k-1}, a_k]}$, $k \in [1..K]$ are **independent conditional derivative Gaussian processes**, respectively with unconditional mean function m_k and unconditional covariance function k_k and that are conditioned to take values x_{k-1} and x_k at a_{k-1} and a_k respectively. We refer to $(SD)_{t_i \in I}$ as a **string derivative Gaussian process**, and to its first coordinate $(z)_{t_i \in I}$ as a **string Gaussian process** namely,

$$(z)_{t_i \in I} \sim \mathcal{SGP}(\{a_k\}, \{m_k\}, \{k_k\}).$$

(B) The **string Gaussian process** $(z)_{t_i \in I}$ defined in (A) is C^1 in the L^2 sense and its L^2 derivative is the process $(z')_{t_i \in I}$ defined in (A).

Proof

Appendix C.1 Proof of Theorem 2 (A)

We will once again turn to Kolmogorov's extension theorem to prove the existence of the stochastic process $(SD)_{t_i \in I}$. The core of the proof is in the finite dimensional measures implied by Theorem 2 (A-1) and (A-2). Let $\{t_i^k \in [a_{k-1}, a_k]\}_{i \in [1..N_k], k \in [1..K]}$ be n times. We first formally construct the finite dimensional measures implied by Theorem 2 (A-1) and (A-2), and then verify that they satisfy the conditions of Kolmogorov's extension theorem.

Let us define the measure $\nu_{t_1^1, \dots, t_1^K, \dots, t_n^1, \dots, t_n^K}^{SD}$ as the probability measure having density with respect to the Lebesgue measure on $\mathcal{E}(\mathbb{R}^2 \times \dots \times \mathbb{R}^2)$ that reads:

$$\begin{aligned} p_{SD}(x_1^1, \dots, x_{N_1}^1, \dots, x_1^K, \dots, x_{N_K}^K, x_{a_0}, \dots, x_{a_K}) &= p_b(x_{a_0}, \dots, x_{a_K}) \times \\ &\prod_{k=1}^K p_{N_k}^{x_{a_{k-1}}, x_{a_k}}(x_{t_1^k}, \dots, x_{t_{N_k}^k}) \end{aligned} \quad (68)$$

where p_b is as per Theorem 2 (A-1) and $p_{N_k}^{x_{a_{k-1}}, x_{a_k}}(x_{t_1^k}, \dots, x_{t_{N_k}^k})$ is the (Gaussian) pdf of the joint distribution of the values at times $\{t_i^k \in [a_{k-1}, a_k]\}$ of the *conditional derivative Gaussian process* with unconditional mean functions m_k and unconditional covariance functions k_k that is conditioned to take values $x_{a_{k-1}} = (z_{a_{k-1}}, z'_{a_{k-1}})$ and $x_{a_k} = (z_{a_k}, z'_{a_k})$ at times a_{k-1} and a_k respectively (the corresponding—conditional—mean and covariance functions are derived from Equations 3 and 4). Let us extend the family of measures ν^{SD} to cases where some or all boundary times a_k are missing, by integrating out the corresponding variables in Equation (68). For instance when a_0 and a_1 are missing

$$\begin{aligned} \nu_{t_1^1, \dots, t_{N_1}^1, \dots, t_1^K, \dots, t_{N_K}^K, a_2, \dots, a_K}^{SD} & (T_1^1, \dots, T_{N_1}^1, \dots, T_1^K, \dots, T_{N_K}^K, A_2, \dots, A_K) \\ &= \nu_{t_1^1, \dots, t_{N_1}^1, \dots, t_1^K, \dots, t_{N_K}^K, a_0, \dots, a_K}^{SD} (T_1^1, \dots, T_{N_1}^1, \dots, T_1^K, \dots, T_{N_K}^K, \mathbb{R}^2, \mathbb{R}^2, A_2, \dots, A_K) \end{aligned} \quad (69)$$

where A_i and T_j^i are rectangle in \mathbb{R}^2 . Finally, we extend the family of measures ν^{SD} to any arbitrary set of indices $\{t_1, \dots, t_n\}$ as follows:

$$\nu_{t_1, \dots, t_n}^{SD}(T_1, \dots, T_n) := \nu_{t_{\pi^*(1)}, \dots, t_{\pi^*(n)}}^{SD}(T_{\pi^*(1)}, \dots, T_{\pi^*(n)}), \quad (70)$$

where π^* is a permutation of $\{1, \dots, n\}$ such that $\{t_{\pi^*(1)}, \dots, t_{\pi^*(n)}\}$ verify the following conditions:

1. $\forall i, j$, if $t_i \in [a_{k_1-1}, a_{k_1}]$, $t_j \in [a_{k_2-1}, a_{k_2}]$, and $k_1 < k_2$, then $\text{Idx}(t_i) < \text{Idx}(t_j)$. Where $\text{Idx}(t_i)$ stands for the index of t_i in $\{t_{\pi^*(1)}, \dots, t_{\pi^*(n)}\}$;
2. if $t_i \notin \{a_0, \dots, a_K\}$ and $t_j \in \{a_0, \dots, a_K\}$ then $\text{Idx}(t_i) < \text{Idx}(t_j)$;
3. if $t_i \in \{a_0, \dots, a_K\}$ and $t_j \in \{a_0, \dots, a_K\}$ then $\text{Idx}(t_i) < \text{Idx}(t_j)$ if and only if $t_i < t_j$.

Any such measure $\nu_{t_{\pi^*(1)}, \dots, t_{\pi^*(n)}}^{SD}$ will fall in the category of either Equation (68) or Equation (69). Although π^* is not unique, any two permutations satisfying the above conditions will only differ by a permutation of times belonging to the same string interval $[a_{k-1}, a_k]$. Moreover, it follows from Equations (68) and (69) that the measures $\nu_{t_{\pi^*(1)}, \dots, t_{\pi^*(n)}}^{SD}$ are invariant by permutation of times belonging to the same string interval $[a_{k-1}, a_k]$, and as a result any two π^* satisfying the above conditions will yield the same probability measure.

The finite dimensional probability measures $\nu_{t_1, \dots, t_n}^{SD}$ are the measures implied by Theorem 2. The permutation condition (57) of Kolmogorov's extension theorem is met by virtue of Equation (70). In effect for every permutation π of $\{1, \dots, n\}$, if we let $\pi' : \{\pi(1), \dots, \pi(n)\} \rightarrow \{1, \dots, n\}$, then

$$\begin{aligned} \nu_{\pi(1), \dots, \pi(n)}^{SD} (T_{\pi(1)}, \dots, T_{\pi(n)}) &:= \nu_{t_{\pi^*(\pi(1))}, \dots, t_{\pi^*(\pi(n))}}^{SD} (T_{\pi^*(\pi(1))}, \dots, T_{\pi^*(\pi(n))}) \\ &= \nu_{t_{\pi^*(1)}, \dots, t_{\pi^*(n)}}^{SD} (T_{\pi^*(1)}, \dots, T_{\pi^*(n)}) \\ &= \nu_{t_1, \dots, t_n}^{SD} (T_1, \dots, T_n). \end{aligned}$$

As for the marginalisation condition (58), it is met for every boundary time by virtue of how we extended ν^{SD} to missing boundary times. All we need to prove now is that the marginalisation condition is also met at any non-boundary time. To do so, it is sufficient to prove that the marginalisation condition holds for t_1^1 , that is:

$$\begin{aligned} \nu_{t_1^1, \dots, t_1^1, \dots, t_1^K, \dots, t_{N_K}^K, a_0, \dots, a_K}^{SD} & (\mathbb{R}^2, T_1^1, T_2^1, \dots, T_{N_1}^1, \dots, T_1^K, \dots, T_{N_K}^K, A_0, \dots, A_K) \\ &= \nu_{t_2^1, \dots, t_{N_1}^1, \dots, t_1^K, \dots, t_{N_K}^K, a_0, \dots, a_K}^{SD} (T_2^1, \dots, T_{N_1}^1, \dots, T_1^K, \dots, T_{N_K}^K, A_0, \dots, A_K) \end{aligned} \quad (71)$$

for every rectangles A_i and T_j^i in \mathbb{R}^2 . In effect, cases where some boundary times are missing are special cases with the corresponding rectangles A_j set to \mathbb{R}^2 . Moreover, if we prove Equation (71), the permutation property (57) will allow us to conclude that the marginalisation also holds true for any other (single) non-boundary time. Furthermore, if Equation (71) holds true, it can be shown that the marginalisation condition will also hold over multiple non-boundary times by using the permutation property (57) and marginalising one non-boundary time after another.

By Fubini's theorem, and considering Equation (68), showing that Equation (71) holds true is equivalent to showing that:

$$\int_{\mathbb{R}^2} p_{\mathcal{N}}^{x_0, x_{a_1}}(x_{t_1}^1, \dots, x_{t_{N_1}}^1) dx_{t_1}^1 = p_{\mathcal{N}}^{x_0, x_{a_1}}(x_{t_2}^1, \dots, x_{t_{N_1}}^1) \quad (72)$$

which holds true as $p_{\mathcal{N}}^{x_0, x_{a_1}}(x_{t_1}^1, \dots, x_{t_{N_1}}^1)$ is a multivariate Gaussian density, and the corresponding marginal is indeed the density of the same *conditional derivative Gaussian process* at times $t_2^1, \dots, t_{N_1}^1$.

This concludes the proof of the existence of the stochastic process $(SD_t)_{t \in I}$.

Appendix C.2 Proof of Theorem 2 (B)

As conditional on boundary conditions the restriction of a *string derivative Gaussian process* on a string interval $[a_{k-1}, a_k]$ is a *derivative Gaussian process*, it follows from Proposition 1 (C) that

$$\forall \tilde{x}_{a_0}, \dots, \tilde{x}_{a_K}, \forall t, t+h \in [a_{k-1}, a_k], \quad \lim_{h \rightarrow 0} \mathbf{E} \left(\left[\frac{z_{t+h} - z_t}{h} - z'_t \right]^2 \middle| x_{a_0} = \tilde{x}_{a_0}, \dots, x_{a_K} = \tilde{x}_{a_K} \right) = 0, \quad (73)$$

or equivalently that:

$$\Delta z'_h := \mathbf{E} \left(\left[\frac{z_{t+h} - z_t}{h} - z'_t \right]^2 \middle| x_{a_0}, \dots, x_{a_K} \right) \xrightarrow{a.s.} 0. \quad (74)$$

Moreover,

$$\Delta z_h = \text{Var} \left(\frac{z_{t+h} - z_t}{h} - z'_t \middle| x_{a_0}, \dots, x_{a_K} \right) + \mathbf{E} \left(\frac{z_{t+h} - z_t}{h} - z'_t \middle| x_{a_0}, \dots, x_{a_K} \right)^2. \quad (75)$$

As both terms in the sum of the above equation are non-negative, it follows that

$$\text{Var} \left(\frac{z_{t+h} - z_t}{h} - z'_t \middle| x_{a_0}, \dots, x_{a_K} \right) \xrightarrow{a.s.} 0 \quad \text{and} \quad \mathbf{E} \left(\frac{z_{t+h} - z_t}{h} - z'_t \middle| x_{a_0}, \dots, x_{a_K} \right)^2 \xrightarrow{a.s.} 0.$$

From which we deduce

$$\mathbf{E} \left(\frac{z_{t+h} - z_t}{h} - z'_t \middle| x_{a_0}, \dots, x_{a_K} \right) \xrightarrow{a.s.} 0.$$

As $\mathbf{E} \left(\frac{z_{t+h} - z_t}{h} - z'_t \middle| x_{a_0}, \dots, x_{a_K} \right)$ depends linearly on the boundary conditions, and as the boundary conditions are jointly-Gaussian (see Appendix H step 1), it follows that

$$\mathbf{E} \left(\frac{z_{t+h} - z_t}{h} - z'_t \middle| x_{a_0}, \dots, x_{a_K} \right)$$

is Gaussian. Finally we note that

$$\text{Var} \left(\frac{z_{t+h} - z_t}{h} - z'_t \middle| x_{a_0}, \dots, x_{a_K} \right)$$

does not depend on the values of the boundary conditions x_{a_k} (but rather on the boundary times), and we recall that convergence almost sure of Gaussian random variables implies convergence in L^2 . Hence, taking the expectation on both side of Equation (75) and then the limit as h goes to 0 we get

$$\mathbf{E} \left(\left[\frac{z_{t+h} - z_t}{h} - z'_t \right]^2 \right) = \mathbf{E}(\Delta z'_h) \xrightarrow{h \rightarrow 0} 0,$$

which proves that the *string GP* $(z_t)_{t \in I}$ is differentiable in the L^2 sense on I and has derivative $(z'_t)_{t \in I}$.

We prove the continuity in the L^2 sense of $(z'_t)_{t \in I}$ in a similar fashion, noting that conditional on the boundary conditions, $(z'_t)_{t \in I}$ is a Gaussian process whose mean function $\frac{d \text{tr}_{\text{ck}}}{dt} a_{k-1, a_k}$ and covariance function $\frac{\partial^2 \text{ck}_{a_{k-1}, a_k}}{\partial z_t \partial y_t}$ are continuous, thus is continuous in the L^2 sense on $[a_{k-1}, a_k]$ (conditional on the boundary conditions). We therefore have that:

$$\forall \tilde{x}_{a_0}, \dots, \tilde{x}_{a_K}, \forall t, t+h \in [a_{k-1}, a_k], \quad \lim_{h \rightarrow 0} \mathbf{E} \left((z'_{t+h} - z'_t)^2 \middle| x_{a_0} = \tilde{x}_{a_0}, \dots, x_{a_K} = \tilde{x}_{a_K} \right) = 0, \quad (76)$$

from which we get that:

$$\Delta z'_h := \mathbf{E} \left([z'_{t+h} - z'_t]^2 \middle| x_{a_0}, \dots, x_{a_K} \right) \xrightarrow{a.s.} 0. \quad (77)$$

Moreover,

$$\Delta z'_h = \text{Var} \left(z'_{t+h} - z'_t \middle| x_{a_0}, \dots, x_{a_K} \right) + \mathbf{E} \left(z'_{t+h} - z'_t \middle| x_{a_0}, \dots, x_{a_K} \right)^2, \quad (78)$$

which implies that

$$\text{Var} \left(z'_{t+h} - z'_t \middle| x_{a_0}, \dots, x_{a_K} \right) \xrightarrow{a.s.} 0$$

and

$$\mathbf{E} \left(z'_{t+h} - z'_t \middle| x_{a_0}, \dots, x_{a_K} \right)^2 \xrightarrow{a.s.} 0,$$

as both terms in the sum in Equation (78) are non-negative. Finally,

$$\text{Var} \left(z'_{t+h} - z'_t \middle| x_{a_0}, \dots, x_{a_K} \right)$$

does not depend on the values of the boundary conditions, and

$$\mathbf{E} \left(z'_{t+h} - z'_t \middle| x_{a_0}, \dots, x_{a_K} \right)$$

is Gaussian for the same reason as before. Hence, taking the expectation on both sides of Equation (78), we get that

$$\mathbf{E} \left([z'_{t+h} - z'_t]^2 \right) = \mathbf{E}(\Delta z'_h) \xrightarrow{h \rightarrow 0} 0,$$

which proves that (z'_t) is continuous in the L^2 sense. ■

Appendix D. Proof of the Condition for Pathwise Regularity Upgrade of String GPs from L^2

In this section we prove that a sufficient condition for the process $(z_t^j)_{t \in I}$ in Theorem 2 to be almost surely continuous and to be the almost sure derivative of the string Gaussian process $(z_t^j)_{t \in I}$, is that the Gaussian processes on $I_k = [a_{k-1}, a_k]$ with mean and covariance functions m_{a_{k-1}, a_k}^k and k_{a_{k-1}, a_k}^k (as per Equations 3 and 4 with $m := m_k$ and $k := k_k$) are themselves almost surely C^1 for every boundary condition.

Firstly we note that the above condition guarantees that the result holds at non-boundary times. As for boundary times, the condition implies that the string GP is almost surely right differentiable (resp. left differentiable) at every left (resp. right) boundary time, including a_0 and a_K . Moreover, the string GP being differentiable in L^2 , the right hand side and left hand side almost sure derivatives are the same, and are equal to the L^2 derivative, which proves that the L^2 derivatives at inner boundary times are also in the almost sure sense. A similar argument holds to conclude that the right (resp. left) hand side derivative at a_0 (resp. a_K) is also in the almost sure sense. Moreover, the derivative process $(z_t^j)_{t \in I}$ admits an almost sure right hand side limit and an almost sure left hand side limit at every inner boundary time and both are equal as the derivative is continuous in L^2 , which proves its almost sure continuity at inner boundary times. Almost sure continuity of $(z_t^j)_{t \in I}$ on the right (resp. left) of a_0 (resp. a_K) is a direct consequence of the above condition.

Appendix E. Proof of Proposition 4

In this section, we prove Proposition 4, which we recall below.

Proposition 4 (Additively separable string GPs are flexible)

Let $k(x, y) := \rho(|x - y|_{I^d}^2)$ be a stationary covariance function generating a.s. C^1 GP paths indexed on \mathbb{R}^d , $d > 0$, and ρ a function that is C^2 on $[0, +\infty[$ and continuous at 0. Let $\phi_s(x_1, \dots, x_d) = \sum_{j=1}^d \phi_s(z_j^1)_{j \in [1, d]}$ be independent stationary Gaussian processes with mean 0 and covariance function k (where the L^2 norm is on \mathbb{R}), and let $f(t_1, \dots, t_d) = \phi_s(z_{t_1}^1, \dots, z_{t_d}^d)$ be the corresponding stationary string GP. Finally, let g be an isotropic Gaussian process indexed on $I^1 \times \dots \times I^d$ with mean 0 and covariance function k (where the L^2 norm is on \mathbb{R}^d). Then:

- 1) $\forall x \in I^1 \times \dots \times I^d$, $H(\nabla f(x)) = H(\nabla g(x))$,
- 2) $\forall x \neq y \in I^1 \times \dots \times I^d$, $I(\nabla f(x); \nabla f(y)) \leq I(\nabla g(x); \nabla g(y))$.

To prove Proposition 4 we need a lemma, which we state and prove below.

Lemma 8 Let X_n be a sequence of Gaussian random vectors with auto-covariance matrix Σ_n and mean μ_n , converging almost surely to X_∞ . If $\Sigma_n \rightarrow \Sigma_\infty$ and $\mu_n \rightarrow \mu_\infty$ then X_∞ is Gaussian with mean μ_∞ and auto-covariance matrix Σ_∞ .

Proof We need to show that the characteristic function of X_∞ is

$$\phi_{X_\infty}(t) := \mathbb{E}(e^{it^T X_\infty}) = e^{it^T \mu_\infty - \frac{1}{2} t^T \Sigma_\infty t}.$$

As Σ_n is positive semi-definite, $\forall n$, $|e^{it^T \mu_n - \frac{1}{2} t^T \Sigma_n t}| = e^{-\frac{1}{2} t^T \Sigma_n t} \leq 1$. Hence, by Lebesgue's dominated convergence theorem,

$$\begin{aligned} \phi_{X_\infty}(t) &= \mathbb{E}(\lim_{n \rightarrow +\infty} e^{it^T X_n}) = \lim_{n \rightarrow +\infty} \mathbb{E}(e^{it^T X_n}) = \lim_{n \rightarrow +\infty} e^{it^T \mu_n - \frac{1}{2} t^T \Sigma_n t} = e^{it^T \mu_\infty - \frac{1}{2} t^T \Sigma_\infty t}. \end{aligned}$$

Appendix E.1 Proof of Proposition 4.1)

Let $x = (t_1^1, \dots, t_1^d) \in I^1 \times \dots \times I^d$. We want to show that $H(\nabla f(x)) = H(\nabla g(x))$ where f and g are as per Proposition 4, and H is the entropy operator. Firstly, we note from Equation (11) that

$$\nabla f(x) = \begin{pmatrix} z_{t_1^1}^1 \\ \vdots \\ z_{t_1^d}^d \end{pmatrix}, \quad (79)$$

where the joint law of the GP $(z_t^j)_{t \in I^j}$ and its derivative $(z_t^j)_{t \in I^j}$ is provided in Proposition 1. As the processes $\left(z_t^j, z_t^j \right)_{t \in I^j}$, $j \in [1, d]$ are assumed to be independent of each other, $\nabla f(x)$ is a Gaussian vector and its covariance matrix reads:

$$\Sigma_{\nabla f(x)} = -2 \frac{dp}{dx}(0) \mathbf{I}_d, \quad (80)$$

where \mathbf{I}_d is the $d \times d$ identity matrix. Hence,

$$H(\nabla f(x)) = \frac{d}{2} (1 + \ln(2\pi)) + \frac{1}{2} \ln |\Sigma_{\nabla f(x)}|. \quad (81)$$

Secondly, let e_j denote the d -dimensional vector whose j -th coordinate is 1 and every other coordinate is 0, and let $h \in \mathbb{R}$. As the proposition assumes the covariance function k generates almost surely C^1 surfaces, the vectors $\left(\frac{g(x+he_1) - g(x)}{h}, \dots, \frac{g(x+he_d) - g(x)}{h} \right)$ are Gaussian vectors converging almost surely as $h \rightarrow 0$. Moreover, their mean is 0 and their covariance matrices have as element on the i -th row and j -th column ($i \neq j$):

$$\text{cov} \left(\frac{g(x+he_i) - g(x)}{h}, \frac{g(x+he_j) - g(x)}{h} \right) = \frac{\rho(2h^2) - 2\rho(h^2) + \rho(0)}{h^2} \quad (82)$$

and as diagonal terms:

$$\text{Var} \left(\frac{g(x+he_j) - g(x)}{h} \right) = 2 \frac{\rho(0) - \rho(h^2)}{h^2}. \quad (83)$$

Taking the limit of Equations (82) and (83) using the first order Taylor expansion of ρ (which the Proposition assumes is C^2), we get that:

$$\Sigma_{\nabla g(x)} = -2 \frac{dp}{dx}(0) \mathbf{I}_d = \Sigma_{\nabla f(x)}, \quad (84)$$

It then follows from Lemma 8 that the limit $\nabla g(x)$ of

$$\left(\frac{g(x+he_1) - g(x)}{h}, \dots, \frac{g(x+he_d) - g(x)}{h} \right)$$

is also a Gaussian vector, which proves that $H(\nabla f(x)) = H(\nabla g(x))$.

Appendix E.2 Proof of Proposition 4.2)

We start by stating and proving another lemma we will later use.

Lemma 9 *Let A and B be two d -dimensional jointly Gaussian vectors with diagonal covariance matrices Σ_A and Σ_B respectively. Let $\Sigma_{A,B}$ be the cross-covariance matrix between A and B , and let $\text{diag}(\Sigma_{A,B})$ be the diagonal matrix whose diagonal is that of $\Sigma_{A,B}$. Then:*

$$\det \left(\begin{array}{c|c} \Sigma_A & \text{diag}(\Sigma_{A,B}) \\ \hline \text{diag}(\Sigma_{A,B}) & \Sigma_B \end{array} \right) \geq \det \left(\begin{array}{c|c} \Sigma_A & \Sigma_{A,B} \\ \hline \Sigma_{A,B}^T & \Sigma_B \end{array} \right).$$

Proof Firstly we note that

$$\det \left(\begin{array}{c|c} \Sigma_A & \text{diag}(\Sigma_{A,B}) \\ \hline \text{diag}(\Sigma_{A,B}) & \Sigma_B \end{array} \right) = \det(\Sigma_A) \det(\Sigma_B - \text{diag}(\Sigma_{A,B}) \Sigma_A^{-1} \text{diag}(\Sigma_{A,B}))$$

and

$$\det \left(\begin{array}{c|c} \Sigma_A & \Sigma_{A,B} \\ \hline \Sigma_{A,B} & \Sigma_B \end{array} \right) = \det(\Sigma_A) \det(\Sigma_B - \Sigma_{A,B}^T \Sigma_A^{-1} \Sigma_{A,B}).$$

As the matrix Σ_A is positive semi-definite, $\det(\Sigma_A) \geq 0$. The case $\det(\Sigma_A) = 0$ is straight-forward. Thus we assume that $\det(\Sigma_A) > 0$, so that all we need to prove is that

$$\det(\Sigma_B - \text{diag}(\Sigma_{A,B}) \Sigma_A^{-1} \text{diag}(\Sigma_{A,B})) \geq \det(\Sigma_B - \Sigma_{A,B}^T \Sigma_A^{-1} \Sigma_{A,B}).$$

Secondly, the matrix $\Sigma_{B|A}^{\text{diag}} := \Sigma_B - \text{diag}(\Sigma_{A,B}) \Sigma_A^{-1} \text{diag}(\Sigma_{A,B})$ being diagonal, its determinant is the product of its diagonal terms:

$$\det(\Sigma_{B|A}^{\text{diag}}) = \prod_{i=1}^d \Sigma_{B|A}[i, i] = \prod_{i=1}^d \left(\Sigma_B[i, i] - \frac{\Sigma_{A,B}[i, i]^2}{\Sigma_A[i, i]} \right).$$

As for the matrix $\Sigma_{B|A} := \Sigma_B - \Sigma_{A,B}^T \Sigma_A^{-1} \Sigma_{A,B}$, we note that it happens to be the covariance matrix of the (Gaussian) distribution of B given A , and thus is positive semi-definite and admits a Cholesky decomposition $\Sigma_{B|A} = LL^T$. It follows that

$$\begin{aligned} \det(\Sigma_{B|A}) &= \prod_{i=1}^d L[i, i]^2 \leq \prod_{i=1}^d \Sigma_{B|A}[i, i] = \prod_{i=1}^d \left(\Sigma_B[i, i] - \sum_{j=1}^d \frac{\Sigma_{A,B}[j, i]^2}{\Sigma_A[j, j]} \right) \\ &\leq \prod_{i=1}^d \left(\Sigma_B[i, i] - \frac{\Sigma_{A,B}[i, i]^2}{\Sigma_A[i, i]} \right) = \det(\Sigma_{B|A}^{\text{diag}}), \end{aligned} \quad (85)$$

where the first inequality results from the fact that $\Sigma_{B|A}[i, i] = \sum_{j=1}^d L[j, i]^2$ by definition of the Cholesky decomposition. This proves that

$$\det(\Sigma_B - \text{diag}(\Sigma_{A,B}) \Sigma_A^{-1} \text{diag}(\Sigma_{A,B})) \geq \det(\Sigma_B - \Sigma_{A,B}^T \Sigma_A^{-1} \Sigma_{A,B}),$$

which as previously discussed concludes the proof of the lemma. \blacksquare

Proof of Proposition 4.2): Let $x = (t_1^x, \dots, t_d^x)$, $y = (t_1^y, \dots, t_d^y) \in I^1 \times \dots \times I^d$, $x \neq y$. We want to show that $I(\nabla f(x); \nabla f(y)) \leq I(\nabla g(x); \nabla g(y))$ where f and g are as per Proposition 4, and

$$I(X; Y) = H(X) + H(Y) - H(X, Y)$$

is the mutual information between X and Y . As we have proved that $\forall x, H(\nabla f(x)) = H(\nabla g(x))$, all we need to prove now is that

$$H(\nabla f(x), \nabla f(y)) \geq H(\nabla g(x), \nabla g(y)).$$

Firstly, it follows from Equation (79) and the fact that the *derivative Gaussian processes* $(z_t^f, z_t^g)_{t \in I^d}$ are independent that $(\nabla f(x), \nabla f(y))$ is a jointly Gaussian vector. Moreover, the cross-covariance matrix $\Sigma_{\nabla f(x), \nabla f(y)}$ is diagonal with diagonal terms:

$$\Sigma_{\nabla f(x), \nabla f(y)}[i, i] = -2 \left[\frac{d\rho}{dx} (\|x - y\|_{L^2}^2) + 2(t_i^x - t_i^y)^2 \frac{d^2\rho}{dx^2} (\|x - y\|_{L^2}^2) \right]. \quad (86)$$

Secondly, it follows from a similar argument to the previous proof that $(\nabla g(x), \nabla g(y))$ is also a jointly Gaussian vector, and the terms $\Sigma_{\nabla g(x), \nabla g(y)}[i, j]$ are evaluated as limit of the cross-covariance terms $\text{cov} \left(\frac{g(x+he_i) - g(x)}{h}, \frac{g(y+he_j) - g(y)}{h} \right)$ as $h \rightarrow 0$. For $i = j$,

$$\begin{aligned} \text{cov} \left(\frac{g(x+he_i) - g(x)}{h}, \frac{g(y+he_i) - g(y)}{h} \right) &= \frac{1}{h^2} \left\{ 2\rho \left(\sum_{k \neq i} (t_k^x - t_k^y)^2 \right) \right. \\ &\quad \left. - \rho \left(\sum_{k \neq i} (t_k^x - t_k^y)^2 + (t_i^x + h - t_i^y)^2 \right) - \rho \left(\sum_{k \neq i} (t_k^x - t_i^y)^2 + (t_i^x - h - t_k^y)^2 \right) \right\}, \end{aligned} \quad (87)$$

As ρ is assumed to be \mathcal{C}^2 , the below Taylor expansions around $h = 0$ hold true:

$$\rho \left(\sum_k (t_k^x - t_k^y)^2 \right) - \rho \left(\sum_{k \neq i} (t_k^x - t_k^y)^2 + (t_i^x - h - t_i^y)^2 \right) = 2(t_i^x - t_i^y) h \frac{d\rho}{dx} \left(\sum_k (t_k^x - t_k^y)^2 \right) \quad (88)$$

$$- \left[\frac{d\rho}{dx} \left(\sum_k (t_k^x - t_k^y)^2 \right) + 2(t_i^x - t_i^y)^2 \frac{d^2\rho}{dx^2} \left(\sum_k (t_k^x - t_k^y)^2 \right) \right] h^2 + o(h^2)$$

$$\rho \left(\sum_k (t_k^x - t_k^y)^2 \right) - \rho \left(\sum_{k \neq i} (t_k^x - t_k^y)^2 + (t_i^x + h - t_i^y)^2 \right) = -2(t_i^x - t_i^y) h \frac{d\rho}{dx} \left(\sum_k (t_k^x - t_k^y)^2 \right) \quad (89)$$

$$- \left[\frac{d\rho}{dx} \left(\sum_k (t_k^x - t_k^y)^2 \right) + 2(t_i^x - t_i^y)^2 \frac{d^2\rho}{dx^2} \left(\sum_k (t_k^x - t_k^y)^2 \right) \right] h^2 + o(h^2)$$

Plugging Equations (88) and (89) into Equation (87) and taking the limit we obtain:

$$\begin{aligned} \Sigma_{\nabla g(x), \nabla g(y)}[i, j] &= -2 \left[\frac{d\rho}{dx} (|x - y|_{L^2}^2) + 2(t_i^x - t_j^y)^2 \frac{d^2\rho}{dx^2} (|x - y|_{L^2}^2) \right] \\ &= \Sigma_{\nabla f(x), \nabla f(y)}[i, j]. \end{aligned} \quad (90)$$

Similarly for $i \neq j$,

$$\begin{aligned} \text{cov} \left(\frac{g(x + h e_i) - g(x)}{h}, \frac{g(y + h e_j) - g(y)}{h} \right) \\ &= \frac{1}{h^2} \left\{ \rho \left(\sum_{k \neq i, j} (t_k^x - t_k^y)^2 + (t_i^x + h - t_j^y)^2 + (t_j^x - h - t_i^y)^2 \right) \right. \\ &\quad - \rho \left(\sum_{k \neq i} (t_k^x - t_k^y)^2 + (t_i^x + h - t_j^y)^2 \right) - \rho \left(\sum_{k \neq j} (t_k^x - t_k^y)^2 + (t_i^x - h - t_j^y)^2 \right) \\ &\quad \left. + \rho \left(\sum_k (t_k^x - t_k^y)^2 \right) \right\}, \end{aligned} \quad (91)$$

and

$$\begin{aligned} &\rho \left(\sum_{k \neq i, j} (t_k^x - t_k^y)^2 + (t_i^x + h - t_j^y)^2 + (t_j^x - h - t_i^y)^2 \right) - \rho \left(\sum_k (t_k^x - t_k^y)^2 \right) \\ &= \left[2 \frac{d\rho}{dx} \left(\sum_k (t_k^x - t_k^y)^2 \right) + 2 \left((t_i^x - t_j^y) - (t_j^x - t_i^y) \right)^2 \times \frac{d^2\rho}{dx^2} \left(\sum_k (t_k^x - t_k^y)^2 \right) \right] h^2 \\ &\quad + 2 \left(t_i^x - t_j^y - t_j^x + t_i^y \right) \frac{d\rho}{dx} \left(\sum_k (t_k^x - t_k^y)^2 \right) h + o(h^2). \end{aligned} \quad (92)$$

Plugging Equations (88), (89) and (92) in Equation (91) and taking the limit we obtain:

$$\Sigma_{\nabla g(x), \nabla g(y)}[i, j] = -4(t_i^x - t_j^y)(t_j^x - t_i^y) \frac{d^2\rho}{dx^2} (|x - y|_{L^2}^2). \quad (93)$$

To summarize, $(\nabla f(x), \nabla f(y))$ and $(\nabla g(x), \nabla g(y))$ are both jointly Gaussian vectors; $\nabla f(x)$, $\nabla g(x)$, $\nabla f(y)$, and $\nabla g(y)$ are (Gaussian) identically distributed with a diagonal covariance matrix; $\Sigma_{\nabla f(x), \nabla f(y)}$ is diagonal; $\Sigma_{\nabla g(x), \nabla g(y)}$ has the same diagonal as $\Sigma_{\nabla f(x), \nabla f(y)}$ but has possibly non-zero off-diagonal terms. Hence, it follows from Lemma 9 that the determinant of the auto-covariance matrix of $(\nabla f(x), \nabla f(y))$ is higher than that of the auto-covariance matrix of $(\nabla g(x), \nabla g(y))$; or equivalently the entropy of $(\nabla f(x), \nabla f(y))$ is higher than that of $(\nabla g(x), \nabla g(y))$ (as both are Gaussian vectors), which as previously discussed is sufficient to conclude that the mutual information between $\nabla f(x)$ and $\nabla f(y)$ is smaller than that between $\nabla g(x)$ and $\nabla g(y)$.

Appendix F. Proof of Proposition 6

In this section, we prove Proposition 6, which we recall below.

Proposition 6 (Extension of the standard GP paradigm)

Let $K \in \mathbb{N}^*$, let $I = [a_0, a_K]$ and $I_k = [a_{k-1}, a_k]$ be intervals with $a_0 < \dots < a_K$. Furthermore, let $m : I \rightarrow \mathbb{R}$ be a C^1 function, m_k the restriction of m to I_k , $h : I \times I \rightarrow \mathbb{R}$ a C^3 symmetric positive semi-definite function, and h_k the restriction of h to $I_k \times I_k$. If

$$(z_t)_{t \in I} \sim \text{SGP}(\{a_k\}, \{m_k\}, \{h_k\}),$$

then

$$\forall h \in [1, K], (z_t)_{t \in I_k} \sim \text{GP}(m, h).$$

Proof

To prove Proposition 6, we consider the string derivative Gaussian process (Theorem 2) $(SD_t)_{t \in I}$, $SD_t = (z_t, \dot{z}_t)$ with unconditional string mean and covariance functions as per Proposition 6 and prove that its restrictions on the intervals $I_k = [a_{k-1}, a_k]$ are derivative Gaussian processes with the same mean function m and covariance function h . Proposition 1(B) will then allow us to conclude that $(z_t)_{t \in I_k}$ are GPs with mean m and covariance function h .

Let $t_1, \dots, t_n \in [a_{k-1}, a_k]$ and let $p_D(x_{a_{k-1}})$ (respectively $p_D(x_{a_k} | x_{a_{k-1}})$) and $p_D(x_{t_1}, \dots, x_{t_n} | x_{a_{k-1}}, x_{a_k})$ denote the pdf of the derivative Gaussian process with mean function m and covariance function h at a_{k-1} (respectively its value at a_k conditional on its value at a_{k-1} , and its values at t_1, \dots, t_n conditional on its values at a_{k-1} and a_k). Saying that the restriction of the string derivative Gaussian process (SD_t) on $[a_{k-1}, a_k]$ is the derivative Gaussian process with mean m and covariance h is equivalent to saying that all finite dimensional marginals of the string derivative Gaussian process $PSD(x_{a_{k-1}}, x_{t_1}, \dots, x_{t_n}, x_{a_k})$, $t_i \in [a_{k-1}, a_k]$, factorise as²⁰:

$$p_{PSD}(x_{a_{k-1}}, x_{t_1}, \dots, x_{t_n}, x_{a_k}) = p_D(x_{a_{k-1}}) p_D(x_{t_1}, \dots, x_{t_n} | x_{a_{k-1}}, x_{a_k}).$$

Moreover, we know from Theorem 2 that by design, $p_{PSD}(x_{a_{k-1}}, x_{t_1}, \dots, x_{t_n}, x_{a_k})$ factorises as

$$p_{PSD}(x_{a_{k-1}}, x_{t_1}, \dots, x_{t_n}, x_{a_k}) = p_{SD}(x_{a_{k-1}} | x_{a_{k-1}}) p_D(x_{t_1}, \dots, x_{t_n} | x_{a_{k-1}}, x_{a_k}).$$

In other words, all we need to prove is that

$$p_{SD}(x_{a_k}) = p_D(x_{a_k})$$

for every boundary time, which we will do by induction. We note by integrating out every boundary condition but the first in p_k (as per Theorem 2 (a-1)) that

$$p_{SD}(x_{a_0}) = p_D(x_{a_0}).$$

If we assume that $p_{SD}(x_{a_{k-1}}) = p_D(x_{a_{k-1}})$ for some $k > 0$, then as previously discussed the restriction of the string derivative Gaussian process on $[a_{k-1}, a_k]$ will be the derivative Gaussian process with the same mean and covariance functions, which will imply that $p_{SD}(x_{a_k}) = p_D(x_{a_k})$. This concludes the proof. \blacksquare

²⁰ We emphasize that the terms on the right hand-side of this equation involve p_D not p_{SD} .

Appendix G. Proof of Lemma 10

In this section, we will prove Lemma 10 that we recall below.

Lemma 10 Let \tilde{X} be a multivariate Gaussian with mean μ_X and covariance matrix Σ_X . If conditional on \tilde{X} , Y is a multivariate Gaussian with mean $M\tilde{X} + A$ and covariance matrix Σ_Y^c , where M , A and Σ_Y^c do not depend on \tilde{X} , then (X, Y) is a jointly Gaussian vector with mean

$$\mu_{X;Y} = \begin{bmatrix} \mu_X \\ M\mu_X + A \end{bmatrix},$$

and covariance matrix

$$\Sigma_{X;Y} = \begin{bmatrix} \Sigma_X & \Sigma_X M^T \\ M\Sigma_X & \Sigma_Y^c + M\Sigma_X M^T \end{bmatrix}.$$

Proof To prove this lemma we introduce two vectors \tilde{X} and \tilde{Y} whose lengths are the same as those of X and Y respectively, and such that (\tilde{X}, \tilde{Y}) is jointly Gaussian with mean $\mu_{X;Y}$ and covariance matrix $\Sigma_{X;Y}$. We then prove that the (marginal) distribution of \tilde{X} is the same as the distribution of X and that the distribution of $\tilde{Y}|\tilde{X} = x$ is the same as $Y|X = x$ for any x , which is sufficient to conclude that (X, Y) and (\tilde{X}, \tilde{Y}) have the same distribution.

It is obvious from the joint (\tilde{X}, \tilde{Y}) that \tilde{X} is Gaussian distribution with mean μ_X and covariance matrix Σ_X . As for the distribution of \tilde{Y} conditional on $\tilde{X} = x$, it follows from the usual Gaussian identities that it is Gaussian with mean

$$M\mu_X + c + M\Sigma_X \Sigma_X^{-1}(x - \mu_X) = Mx + c,$$

and covariance matrix

$$\Sigma_Y^c + M\Sigma_X M^T - M\Sigma_X \Sigma_X^{-1} \Sigma_X^T M^T = \Sigma_Y^c,$$

which is the same distribution as that of $Y|X = x$ since the covariance matrix Σ_X is symmetric. This concludes our proof. \blacksquare

Appendix H. Proof of Proposition 5

In this section we will prove that *string GPs* with link function ϕ_s are GPs, or in other words that if f is a *string GP* indexed on \mathbb{R}^d , $d > 0$ with link function $\phi_s(x_1, \dots, x_d) = \sum_{j=1}^d x_j$, then $(f(x_1), \dots, f(x_n))$ has a multivariate Gaussian distribution for every set of distinct points $x_1, \dots, x_n \in \mathbb{R}^d$.

Proof As the sum of independent Gaussian processes is a Gaussian process, a sufficient condition for additively separable *string GPs* to be GPs in dimensions $d > 1$ is that *string GPs* be GPs in dimension 1. Hence, all we need to do is to prove that *string GPs* are GPs in dimension 1.

Let $(z_t^j, z_t^j)_{t \in \mathcal{I}^j}$ be a string derivative GP in dimension 1, with boundary times $a_0^j, \dots, a_{K^j}^j$, and unconditional string mean and covariance functions m_k^j and k_k^j respectively. We want to prove that $(z_{t_1}^j, \dots, z_{t_n}^j)$ is jointly Gaussian for any $t_1, \dots, t_n \in \mathcal{I}^j$.

APPENDIX H.0.1 STEP 1 $(z_{a_0^j}^j, z_{a_0^j}^j, \dots, z_{a_{K^j}^j}^j, z_{a_{K^j}^j}^j)$ IS JOINTLY GAUSSIAN

We first prove recursively that the vector $(z_{a_0^j}^j, z_{a_0^j}^j, \dots, z_{a_{K^j}^j}^j, z_{a_{K^j}^j}^j)$ is jointly Gaussian. We note from Theorem 2 that $(z_t^j, z_t^j)_{t \in [a_0, a_1]}$ is the *derivative Gaussian process* with mean m_1^j and covariance function k_1^j . Hence, $(z_{a_0^j}^j, z_{a_0^j}^j, \dots, z_{a_1^j}^j, z_{a_1^j}^j)$ is jointly Gaussian. Moreover, let us assume that $\mathcal{B}_{k-1} := (z_{a_0^j}^j, z_{a_0^j}^j, \dots, z_{a_{k-1}^j}^j, z_{a_{k-1}^j}^j)$ is jointly Gaussian for some $k > 1$. Conditional on \mathcal{B}_{k-1} , $(z_{a_k^j}^j, z_{a_k^j}^j)$ is Gaussian with covariance matrix independent of \mathcal{B}_{k-1} , and with mean

$$\begin{bmatrix} m_k^j(a_k^j) \\ \frac{dm_k^j}{dt}(a_k^j) \end{bmatrix} + {}^j\mathbf{K}_{a_k^j, a_{k-1}^j} {}^j\mathbf{K}_{a_{k-1}^j, a_{k-1}^j}^{-1} \begin{bmatrix} z_{a_{k-1}^j}^j - m_k^j(a_{k-1}^j) \\ z_{a_{k-1}^j}^j - \frac{dm_k^j}{dt}(a_{k-1}^j) \end{bmatrix},$$

which depends linearly on $(z_{a_0^j}^j, z_{a_0^j}^j, \dots, z_{a_{k-1}^j}^j, z_{a_{k-1}^j}^j)$. Hence by Lemma 10,

$$(z_{a_0^j}^j, z_{a_0^j}^j, \dots, z_{a_k^j}^j, z_{a_k^j}^j)$$

is jointly Gaussian.

APPENDIX H.0.2 STEP 2 $(z_{a_0^j}^j, z_{a_0^j}^j, \dots, z_{a_{K^j}^j}^j, z_{a_{K^j}^j}^j, \dots, z_{t_k^j}^j, z_{t_k^j}^j, \dots)$ IS JOINTLY GAUSSIAN

Let $t_k^j, \dots, t_n^j \in]a_{k-1}^j, a_k^j[$, $k \leq K^j$ be distinct string times. We want to prove that the vector $(z_{a_0^j}^j, z_{a_0^j}^j, \dots, z_{a_{K^j}^j}^j, z_{a_{K^j}^j}^j, \dots, z_{t_k^j}^j, z_{t_k^j}^j, \dots)$ where all boundary times are represented, and for any finite number of string times is jointly Gaussian. Firstly, we have already proved that

$(z_{a_0^j}^j, z_{a_0^j}^j, \dots, z_{a_{K^j}^j}^j, z_{a_{K^j}^j}^j)$ is jointly Gaussian. Secondly, we note from Theorem 2 that conditional on $(z_{a_0^j}^j, z_{a_0^j}^j, \dots, z_{a_{K^j}^j}^j, z_{a_{K^j}^j}^j)$, $(\dots, z_{t_k^j}^j, z_{t_k^j}^j, \dots)$ is a Gaussian vector whose covariance matrix does not depend on $(z_{a_0^j}^j, z_{a_0^j}^j, \dots, z_{a_{K^j}^j}^j, z_{a_{K^j}^j}^j)$, and whose mean depends linearly on

$$(z_{a_0^j}^j, z_{a_0^j}^j, \dots, z_{a_{K^j}^j}^j, z_{a_{K^j}^j}^j).$$

Hence,

$$(z_{a_0^j}^j, z_{a_0^j}^j, \dots, z_{a_{K^j}^j}^j, z_{a_{K^j}^j}^j, \dots, z_{t_k^j}^j, z_{t_k^j}^j, \dots)$$

is jointly Gaussian (by Lemma 10).

APPENDIX H.0.3 STEP 3 $(z_{t_1}^j, \dots, z_{t_n}^j)$ IS JOINTLY GAUSSIAN

$(z_{t_1}^j, z_{t_1}^j, \dots, z_{t_n}^j, z_{t_n}^j)$ is jointly Gaussian as it can be regarded as the marginal of some joint distribution of the form $(z_{a_0^j}^j, z_{a_0^j}^j, \dots, z_{a_{K^j}^j}^j, z_{a_{K^j}^j}^j, \dots, z_{t_k^j}^j, z_{t_k^j}^j, \dots)$. Hence, its marginal $(z_{t_1}^j, \dots, z_{t_n}^j)$ is also jointly Gaussian, which concludes our proof. \blacksquare

Appendix I. Derivation of Global String GP Mean and Covariance Functions

We begin with *derivative string GPs* indexed on \mathbb{R} . Extensions to *membrane GPs* are easily achieved for a broad range of link functions. In our exposition, we focus on the class of *elementary symmetric*

polynomials (Macdonald (1995)). In addition to containing the link function ϕ_s previously introduced, this family of polynomials yields global covariance structures that have many similarities with existing kernel approaches, which we discuss in Section 4.3.

For $n \leq d$, the n -th order elementary symmetric polynomial is given by

$$e_0(x_1, \dots, x_d) := 1, \quad \forall 1 \leq n \leq d \quad e_n(x_1, \dots, x_d) = \sum_{1 \leq i_1 < i_2 < \dots < i_n \leq d} \prod_{k=1}^n x_{i_k}. \quad (94)$$

As an illustration,

$$\begin{aligned} e_1(x_1, \dots, x_d) &= \sum_{j=1}^d x_j = \phi_8(x_1, \dots, x_d), \\ e_2(x_1, \dots, x_d) &= x_1 x_2 + x_1 x_3 + \dots + x_1 x_d + \dots + x_{d-1} x_d, \\ &\dots \\ e_d(x_1, \dots, x_d) &= \prod_{j=1}^d x_j = \phi_9(x_1, \dots, x_d). \end{aligned}$$

Let f denote a *membrane GP* indexed on \mathbb{R}^d with link function e_n and by $(z_1^1), \dots, (z_1^d)$, its independent building block string GPs. Furthermore, let m_k^j and k_k^j denote the unconditional mean and covariance functions corresponding to the k -th string of (z_k^j) defined on $[a_{k-1}^j, a_k^j]$. Finally, let us define

$$\bar{m}^j(t) := \mathbb{E}(z_k^j), \quad \bar{m}^{j'}(t) := \mathbb{E}(z_k^{j'}),$$

the global mean functions of the j -th building block string GP and of its derivative, where $\forall t \in I^j$. It follows from the independence of the building block string GPs that:

$$\bar{m}^f(t_1, \dots, t_d) := \mathbb{E}(f(t_1, \dots, t_d)) = e_n(\bar{m}^1(t_1), \dots, \bar{m}^d(t_d)).$$

Moreover, noting that

$$\frac{\partial e_n}{\partial x_j} = e_{n-1}(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_d),$$

it follows that:

$$\bar{m}^{\nabla f}(t_1, \dots, t_d) := \mathbb{E}(\nabla f(t_1, \dots, t_d)) = \begin{bmatrix} \bar{m}^{1'}(t_1) e_{n-1}(\bar{m}^2(t_2), \dots, \bar{m}^d(t_d)) \\ \bar{m}^{2'}(t_2) e_{n-1}(\bar{m}^1(t_1), \dots, \bar{m}^{d-1}(t_{d-1})) \\ \dots \\ \bar{m}^{d'}(t_d) e_{n-1}(\bar{m}^1(t_1), \dots, \bar{m}^{d-1}(t_{d-1})) \end{bmatrix}.$$

Furthermore, for any $u_j, v_j \in I^j$ we also have that

$$\text{cov}(f(u_1, \dots, u_d), f(v_1, \dots, v_d)) = e_n(\text{cov}(z_{u_1}^1, z_{v_1}^1), \dots, \text{cov}(z_{u_d}^d, z_{v_d}^d)),$$

$$\text{cov}\left(\frac{\partial f}{\partial x_i}(u_1, \dots, u_d), f(v_1, \dots, v_d)\right) = e_n(\text{cov}(z_{u_1}^{i-1}, z_{v_1}^1), \dots, \text{cov}(z_{u_i}^i, z_{v_i}^i), \dots, \text{cov}(z_{u_d}^d, z_{v_d}^d)),$$

and for $i \leq j$

$$\text{cov}\left(\frac{\partial f}{\partial x_i}(u_1, \dots, u_d), \frac{\partial f}{\partial x_j}(v_1, \dots, v_d)\right) = \begin{cases} e_n(\text{cov}(z_{u_1}^1, z_{v_1}^1), \dots, \text{cov}(z_{u_i}^i, z_{v_i}^i), \dots, \text{cov}(z_{u_d}^d, z_{v_d}^d)), & \text{if } i < j \\ e_n(\text{cov}(z_{u_1}^1, z_{v_1}^1), \dots, \text{cov}(z_{u_i}^i, z_{v_i}^i), \dots, \text{cov}(z_{u_d}^d, z_{v_d}^d)), & \text{if } i = j \end{cases}.$$

Overall, for any elementary symmetric polynomial link function, multivariate mean and covariance functions are easily deduced from the previously boxed equations and the univariate quantities

$$\bar{m}^j(u), \quad \bar{m}^{j'}(u), \quad \text{and } {}^j\mathbf{K}_{uv} := \begin{bmatrix} \text{cov}(z_u^j, z_u^j) & \text{cov}(z_u^j, z_v^j) \\ \text{cov}(z_u^j, z_v^j) & \text{cov}(z_v^j, z_v^j) \end{bmatrix} = {}^j\mathbf{K}_{v;u}^T$$

which we now derive. In this regards, we will need the following lemma.

Lemma 10 *Let X be a multivariate Gaussian with mean μ_X and covariance matrix Σ_X . If conditional on X , Y is a multivariate Gaussian with mean $MX + A$ and covariance matrix Σ_Y^X where M, A and Σ_Y^X do not depend on X , then (X, Y) is a jointly Gaussian vector with mean*

$$\mu_{X;Y} = \begin{bmatrix} \mu_X \\ M\mu_X + A \end{bmatrix},$$

and covariance matrix

$$\Sigma_{X;Y} = \begin{bmatrix} \Sigma_X & \Sigma_X M^T \\ M \Sigma_X & \Sigma_Y^X + M \Sigma_X M^T \end{bmatrix}.$$

Proof See Appendix G. ■

APPENDIX I.0.1 GLOBAL STRING GP MEAN FUNCTIONS

We now turn to evaluating the univariate global mean functions \bar{m}^j and $\bar{m}^{j'}$. We start with boundary times and then generalise to other times.

Boundary times: We note from Theorem 2 that the restriction $(z_k^j, z_k^{j'})_{k \in [a_0^j, a_1^j]}$ is the derivative Gaussian process with mean and covariance functions m_1^j and k_1^j . Thus,

$$\begin{bmatrix} \bar{m}^j(a_0^j) \\ \bar{m}^{j'}(a_0^j) \end{bmatrix} = \begin{bmatrix} m_1^j(a_0^j) \\ \frac{dm_1^j}{dt}(a_0^j) \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} \bar{m}^j(a_1^j) \\ \bar{m}^{j'}(a_1^j) \end{bmatrix} = \begin{bmatrix} m_1^j(a_1^j) \\ \frac{dm_1^j}{dt}(a_1^j) \end{bmatrix}.$$

For $k > 1$, we recall that conditional on $(z_{a_{k-1}^j}^j, z_{a_{k-1}^j}^{j'})$, $(z_{a_k^j}^j, z_{a_k^j}^{j'})$ is Gaussian with mean

$$\begin{bmatrix} \bar{m}_k^j(a_k^j) \\ \bar{m}_k^{j'}(a_k^j) \end{bmatrix} + {}^j\mathbf{K}_{a_k^j, a_{k-1}^j} \begin{bmatrix} z_{a_{k-1}^j}^j \\ z_{a_{k-1}^j}^{j'} \end{bmatrix} - \begin{bmatrix} m_k^j(a_{k-1}^j) \\ \frac{dm_k^j}{dt}(a_{k-1}^j) \end{bmatrix} + {}^j\mathbf{K}_{a_{k-1}^j, a_k^j} \begin{bmatrix} z_{a_{k-1}^j}^j \\ z_{a_{k-1}^j}^{j'} \end{bmatrix} - \begin{bmatrix} m_k^j(a_{k-1}^j) \\ \frac{dm_k^j}{dt}(a_{k-1}^j) \end{bmatrix},$$

$$\text{with } {}^j\mathbf{K}_{u,v} = \begin{bmatrix} k_k^j(u, v) & \frac{\partial k_k^j}{\partial x}(u, v) \\ \frac{\partial k_k^j}{\partial x}(u, v) & \frac{\partial^2 k_k^j}{\partial x^2}(u, v) \end{bmatrix}.$$

It then follows from the law of total expectations that for all $k > 1$

$$\begin{bmatrix} \bar{m}_k^j(a_k^j) \\ \bar{m}_k^{j'}(a_k^j) \end{bmatrix} = \begin{bmatrix} m_k^j(a_k^j) \\ \frac{dm_k^j}{dt}(a_k^j) \end{bmatrix} + {}^j\mathbf{K}_{a_k^j, a_{k-1}^j} {}^j\mathbf{K}_{a_{k-1}^j}^{-1} \begin{bmatrix} \bar{m}^j(a_{k-1}^j) - m_k^j(a_{k-1}^j) \\ \bar{m}^{j'}(a_{k-1}^j) - \frac{dm_k^j}{dt}(a_{k-1}^j) \end{bmatrix}.$$

String times: As for non-boundary times $t \in]a_{k-1}^j, a_k^j[$, conditional on $(z_{a_{k-1}^j}^j, z_{a_{k-1}^j}^{j'})$ and $(z_{a_k^j}^j, z_{a_k^j}^{j'})$, $(z_t^j, z_t^{j'})$ is Gaussian with mean

$$\begin{bmatrix} m_k^j(t) \\ \frac{dm_k^j}{dt}(t) \end{bmatrix} + {}^j\mathbf{K}_{t, (a_{k-1}^j, a_k^j)} {}^j\mathbf{K}_{(a_{k-1}^j, a_k^j)}^{-1} (a_{k-1}^j, a_k^j); (a_{k-1}^j, a_k^j) \begin{bmatrix} z_{a_{k-1}^j}^j - m_k^j(a_{k-1}^j) \\ z_{a_{k-1}^j}^{j'} - \frac{dm_k^j}{dt}(a_{k-1}^j) \\ z_{a_k^j}^j - m_k^j(a_k^j) \\ z_{a_k^j}^{j'} - \frac{dm_k^j}{dt}(a_k^j) \end{bmatrix},$$

with

$${}^j\mathbf{K}_{(a_{k-1}^j, a_k^j); (a_{k-1}^j, a_k^j)} = \begin{bmatrix} {}^j\mathbf{K}_{a_{k-1}^j, a_{k-1}^j} & {}^j\mathbf{K}_{a_{k-1}^j, a_k^j} \\ {}^j\mathbf{K}_{a_k^j, a_{k-1}^j} & {}^j\mathbf{K}_{a_k^j, a_k^j} \end{bmatrix}$$

and

$${}^j\mathbf{K}_{t, (a_{k-1}^j, a_k^j)} = \begin{bmatrix} {}^j\mathbf{K}_{t, a_{k-1}^j} & {}^j\mathbf{K}_{t, a_k^j} \end{bmatrix}.$$

Hence, using once again the law of total expectation, it follows that for any $t \in]a_{k-1}^j, a_k^j[$,

$$\begin{bmatrix} \bar{m}_k^j(t) \\ \bar{m}_k^{j'}(t) \end{bmatrix} = \begin{bmatrix} m_k^j(t) \\ \frac{dm_k^j}{dt}(t) \end{bmatrix} + {}^j\mathbf{K}_{t, (a_{k-1}^j, a_k^j); (a_{k-1}^j, a_k^j)} {}^j\mathbf{K}_{(a_{k-1}^j, a_k^j)}^{-1} \begin{bmatrix} \bar{m}^j(a_{k-1}^j) - m_k^j(a_{k-1}^j) \\ \bar{m}^{j'}(a_{k-1}^j) - \frac{dm_k^j}{dt}(a_{k-1}^j) \\ \bar{m}^j(a_k^j) - m_k^j(a_k^j) \\ \bar{m}^{j'}(a_k^j) - \frac{dm_k^j}{dt}(a_k^j) \end{bmatrix}.$$

We note in particular that when $\forall j, k, m_k^j = 0$, it follows that $\bar{m}^j = 0, \bar{m}^{j'} = 0, \bar{m} \nabla f = 0$.

APPENDIX I.0.2 GLOBAL STRING GP COVARIANCE FUNCTIONS

As for the evaluation of ${}^j\mathbf{K}_{u,v}$, we start by noting that the covariance function of a univariate *string GP* is the same as that of another *string GP* whose strings have the same unconditional kernels but unconditional mean functions $m_k^j = 0$, so that to evaluate univariate *string GP* kernels we may assume that $\forall j, k, m_k^j = 0$ without loss of generality. We start with the case where u and v are both boundary times, after which we will generalise to other times.

Boundary times: As previously discussed, the restriction $(z_t^j, z_t^{j'})_{t \in \{a_0^j, a_1^j\}}$ is the *derivative Gaussian process* with mean 0 and covariance function $k_{a_0^j}^j$. Thus,

$${}^j\bar{\mathbf{K}}_{a_0^j, a_0^j} = {}^j\mathbf{K}_{a_0^j, a_0^j}, \quad {}^j\bar{\mathbf{K}}_{a_1^j, a_1^j} = {}^j\mathbf{K}_{a_1^j, a_1^j}, \quad {}^j\bar{\mathbf{K}}_{a_0^j, a_1^j} = {}^j\mathbf{K}_{a_0^j, a_1^j}. \quad (95)$$

We recall that conditional on the boundary conditions at or prior to a_{k-1}^j , $(z_{a_k^j}^j, z_{a_k^j}^{j'})$ is Gaussian with mean

$$\begin{bmatrix} z_{a_{k-1}^j}^j \\ z_{a_{k-1}^j}^{j'} \end{bmatrix} \quad \text{with} \quad b_k M = {}^j\mathbf{K}_{a_k^j, a_{k-1}^j} {}^j\mathbf{K}_{a_{k-1}^j, a_{k-1}^j}^{-1},$$

and covariance matrix

$${}^j\mathbf{K}_{a_k^j, a_k^j} = {}^j\mathbf{K}_{a_k^j, a_{k-1}^j} - b_k M {}^j\mathbf{K}_{a_{k-1}^j, a_{k-1}^j}^{-1} b_k M^T.$$

Hence using Lemma 10 with $M = \begin{bmatrix} b_k M & 0 & \dots & 0 \end{bmatrix}$ where there are $(k-1)$ null block 2×2 matrices, and noting that $(z_{a_0^j}^j, z_{a_0^j}^{j'}, \dots, z_{a_{k-1}^j}^j, z_{a_{k-1}^j}^{j'})$ is jointly Gaussian, it follows that the vector

$(z_{a_0^j}^j, z_{a_0^j}^{j'}, \dots, z_{a_k^j}^j, z_{a_k^j}^{j'})$ is jointly Gaussian, that $(z_{a_k^j}^j, z_{a_k^j}^{j'})$ has covariance matrix

$${}^j\bar{\mathbf{K}}_{a_k^j, a_k^j} = \begin{bmatrix} b_k \Sigma & b_k M {}^j\bar{\mathbf{K}}_{a_{k-1}^j, a_{k-1}^j}^{-1} b_k M^T \\ b_k M {}^j\bar{\mathbf{K}}_{a_{k-1}^j, a_{k-1}^j}^{-1} b_k M^T & {}^j\mathbf{K}_{a_{k-1}^j, a_{k-1}^j} \end{bmatrix},$$

and that the covariance matrix between the boundary conditions at a_k^j and at any earlier boundary time $a_l^j, l < k$ reads:

$${}^j\bar{\mathbf{K}}_{a_l^j, a_k^j} = \begin{bmatrix} b_l M & b_l M {}^j\bar{\mathbf{K}}_{a_{k-1}^j, a_{k-1}^j}^{-1} b_k M^T \\ b_l M {}^j\bar{\mathbf{K}}_{a_{k-1}^j, a_{k-1}^j}^{-1} b_k M^T & {}^j\mathbf{K}_{a_{k-1}^j, a_{k-1}^j} \end{bmatrix}.$$

String times: Let $u \in]a_{p-1}^j, a_p^j[$, $v \in]a_{q-1}^j, a_q^j[$. By the law of total expectation, we have that

$${}^j\bar{\mathbf{K}}_{u,v} := \mathbb{E} \left(\begin{bmatrix} z_u^j \\ z_u^{j'} \end{bmatrix} \begin{bmatrix} z_v^j \\ z_v^{j'} \end{bmatrix} \right) = \mathbb{E} \left(\mathbb{E} \left(\begin{bmatrix} z_u^j \\ z_u^{j'} \end{bmatrix} \begin{bmatrix} z_v^j \\ z_v^{j'} \end{bmatrix} \middle| \mathcal{B}(p, q) \right) \right),$$

where $\mathcal{B}(p, q)$ refers to the boundary conditions at the boundaries of the p -th and q -th strings, in other words $\left\{ z_x^j, z_x^{j'} \mid x \in \{a_{p-1}^j, a_p^j, a_{q-1}^j, a_q^j\} \right\}$. Furthermore, using the definition of the covariance matrix under the conditional law, it follows that

$$\mathbb{E} \left(\begin{bmatrix} z_u^j \\ z_u^{j'} \end{bmatrix} \begin{bmatrix} z_v^j \\ z_v^{j'} \end{bmatrix} \middle| \mathcal{B}(p, q) \right) = {}^j\bar{\mathbf{K}}_{u,v} + \mathbb{E} \left(\begin{bmatrix} z_u^j \\ z_u^{j'} \end{bmatrix} \mathcal{B}(p, q) \right) \mathbb{E} \left(\begin{bmatrix} z_v^j \\ z_v^{j'} \end{bmatrix} \middle| \mathcal{B}(p, q) \right), \quad (96)$$

where ${}^j\bar{\mathbf{K}}_{u,v}$ refers to the covariance matrix between $(z_u^j, z_u^{j'})$ and $(z_v^j, z_v^{j'})$ conditional on the boundary conditions $\mathcal{B}(p, q)$, and can be easily evaluated from Theorem 2. In particular,

$$\text{if } p \neq q, \quad {}^j\bar{\mathbf{K}}_{u,v} = 0, \quad \text{and if } p = q, \quad {}^j\bar{\mathbf{K}}_{u,v} = {}^j\mathbf{K}_{u,v} - \begin{bmatrix} {}^j\mathbf{K}_{u, a_{p-1}^j}^T \\ {}^j\mathbf{K}_{u, a_{p-1}^j} \end{bmatrix}, \quad (97)$$

where

$$\forall x, l, \quad {}^j\Delta_x = \begin{bmatrix} {}^j\mathbf{K}_{x:\alpha^j} \\ {}^j\mathbf{K}_{x:\alpha^j} \end{bmatrix} \begin{bmatrix} {}^j\mathbf{K}_{\alpha^j-1:\alpha^j-1} & {}^j\mathbf{K}_{\alpha^j-1:\alpha^j} \\ {}^j\mathbf{K}_{\alpha^j:\alpha^j-1} & {}^j\mathbf{K}_{\alpha^j:\alpha^j} \end{bmatrix}^{-1}$$

We also note that

$$\mathbb{E} \left(\begin{bmatrix} z_{\alpha^j}^j \\ z_{\beta^j}^j \end{bmatrix} \middle| \mathcal{B}(p, q) \right) = {}^j\Lambda_u \begin{bmatrix} z_{\alpha^j-1}^j & z_{\alpha^j-1}^j \\ z_{\alpha^j-1}^j & z_{\alpha^j-1}^j \\ z_{\alpha^j}^j & z_{\alpha^j}^j \\ z_{\alpha^j}^j & z_{\alpha^j}^j \\ z_{\beta^j}^j & z_{\beta^j}^j \\ z_{\beta^j}^j & z_{\beta^j}^j \end{bmatrix} \text{ and } \mathbb{E} \left(\begin{bmatrix} z_v^j \\ z_{\beta^j}^j \end{bmatrix} \middle| \mathcal{B}(p, q) \right) = \begin{bmatrix} z_{\alpha^j-1}^j & z_{\beta^j}^j & z_{\beta^j}^j \\ z_{\alpha^j-1}^j & z_{\alpha^j}^j & z_{\alpha^j}^j \\ z_{\alpha^j}^j & z_{\alpha^j}^j & z_{\beta^j}^j \\ z_{\alpha^j}^j & z_{\alpha^j}^j & z_{\beta^j}^j \end{bmatrix} {}^j\Lambda_v^T.$$

Hence, taking the expectation with respect to the boundary conditions on both sides of Equation (96), we obtain:

$$\forall u \in [\alpha_{p-1}^j, \alpha_{p-1}^j], v \in [\alpha_{q-1}^j, \alpha_{q-1}^j], \quad {}^j\bar{\mathbf{K}}_{u:v} = {}^j\bar{\mathbf{K}}_{u:v} + {}^j\Lambda_u \begin{bmatrix} {}^j\bar{\mathbf{K}}_{\alpha^j-1:\alpha^j-1} & {}^j\bar{\mathbf{K}}_{\alpha^j-1:\alpha^j} \\ {}^j\bar{\mathbf{K}}_{\alpha^j:\alpha^j-1} & {}^j\bar{\mathbf{K}}_{\alpha^j:\alpha^j} \end{bmatrix} {}^j\Lambda_v^T,$$

where ${}^j\bar{\mathbf{K}}_{u:v}$ is provided in Equation (97).

References

R. P. Adams and O. Segle. Gaussian process product models for nonparametric nonstationarity. In *International Conference on Machine Learning (ICML)*, pages 1–8, 2008.

R. J. Adler and J. E. Taylor. *Topological Complexity of Smooth Random Functions: École D'Été de Probabilités de Saint-Flour*. Springer, 2011.

P. Alquier, N. Friel, R. Everitt, and A. Boland. Noisy monte carlo: Convergence of Markov chains with approximate transition kernels. *Statistics and Computing*, 26(1-2):29–47, 2016.

R. Bardenet, A. Doucet, and C. Holmes. Towards scaling up Markov chain monte carlo: an adaptive subsampling approach. In *International Conference on Machine Learning (ICML)*, pages 405–413, 2014.

S. Brooks, A. Gelman, G. Jones, and X.-L. Meng. *Handbook of Markov Chain Monte Carlo*. CRC press, 2011.

R. Calandra, J. Peters, C. E. Rasmussen, and M. P. Deisenroth. Manifold Gaussian processes for regression. *arXiv preprint arXiv:1402.5876*, 2014.

Y. Cao and D. J. Fleet. Generalized product of experts for automatic and principled fusion of Gaussian process predictions. *arXiv preprint arXiv:1410.7827*, 2014.

D. J. Daley and D. Vere-Jones. *An Introduction to the Theory of Point Processes*. Springer-Verlag, 2008.

M. Deisenroth and J. W. Ng. Distributed Gaussian processes. In *International Conference on Machine Learning (ICML)*, pages 1481–1490, 2015.

J. L. Doob. The elementary Gaussian processes. *The Annals of Mathematical Statistics*, 15(3): 229–282, 1944.

N. Durand, D. Ginsbourger, and O. Roustant. Additive covariance kernels for high-dimensional Gaussian process modeling. *Annales de la Faculté de Sciences de Toulouse*, 21(3), 2012.

D. Duvenaud, H. Nickisch, and C. E. Rasmussen. Additive Gaussian processes. In *Advances in Neural Information Processing Systems (NIPS)*, pages 226–234, 2011.

T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, pages 209–230, 1973.

N. J. Foti and S. Williamson. A survey of non-exchangeable priors for Bayesian nonparametric models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):359–371, 2015.

R. B. Gramacy and H. K. H. Lee. Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103(483), 2008.

P. J. Green. Reversible jump Markov chain monte carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

- P. J. Green and D. I. Hastie. Reversible jump MCMC. *Genetics*, 155(3):1391–1403, 2009.
- J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. In *Uncertainty in Artificial Intelligence (UAI)*, pages 282–290, 2013.
- J. Hensman, A. Matthews, and Z. Ghahramani. Scalable variational Gaussian process classification. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 351–360, 2015.
- M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013.
- I. Karatzas and R. Fernholz. Stochastic Portfolio Theory: An Overview. *Handbook of Numerical Analysis*, 15:89–167, 2009.
- H. Kim, Mallick B. K., and Holmes C. C. Analyzing nonstationary spatial data using piecewise Gaussian processes. *Journal of the American Statistical Association*, 100(470):653–668, 2005.
- J. Kingman. Completely random measures. *Pacific Journal of Mathematics*, 21(1):59–78, 1967.
- Y.-L. Kom Samo and A. Vervuurt. Stochastic portfolio theory : a machine learning perspective. In *Uncertainty in Artificial Intelligence (UAI)*, pages 657–665, 2016.
- N. Lawrence, M. Seeger, and R. Herbrich. Fast sparse Gaussian process methods: the informative vector machine. In *Advances in Neural Information Processing Systems (NIPS)*, pages 625–632, 2003.
- M. Lazaro-Gredilla, J. Quinero-Candela, C. E. Rasmussen, and A. R. Figueiras-Vida. Sparse spectrum Gaussian process regression. *Journal of Machine Learning Research*, 11:1866–1881, 2010.
- Q. Le, T. Sarlós, and A. Smola. Fastfood-approximating kernel expansions in loglinear time. In *International Conference on Machine Learning (ICML)*, 2013.
- I. G. Macdonald. *Symmetric Functions and Hall Polynomials*. Oxford University Press, 1995.
- D. J. C. MacKay. Introduction to Gaussian processes. In *NATO ASI Series F: Computer and Systems Sciences*, pages 133–166. Springer, Berlin, 1998.
- E. Meeds and S. Osindero. An alternative infinite mixture of Gaussian process experts. In *Advances in Neural Information Processing Systems (NIPS)*, 2006.
- I. Murray, R. P. Adams, and D. J. C. MacKay. Elliptical slice sampling. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 9–16, 2010.
- R. Neal. *Bayesian learning for neural networks*. Lecture notes in Statistics. Springer, 1996.
- T. Nguyen and E. Bomilla. Fast allocation of Gaussian process experts. In *International Conference on Machine Learning (ICML)*, pages 145–153, 2014.
- B. Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. Hochschultext / Universitext. Springer, 2003.
- C. Paciorek and M. Schervish. Nonstationary covariance functions for Gaussian process regression. In *Advances in Neural Information Processing Systems (NIPS)*, pages 273–280, 2004.
- J. Pitman and M. Yor. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, pages 855–900, 1997.
- C. Plagemann, K. Kersting, and W. Burgard. Nonstationary Gaussian process regression using point estimate of local smoothness. In *European Conference on Machine Learning (ECML)*, pages 204–219, 2008.
- J. Quinero-Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.
- A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1177–1184, 2007.
- C. E. Rasmussen and Z. Ghahramani. Infinite mixtures of Gaussian process experts. In *Advances in Neural Information Processing Systems (NIPS)*, pages 881–888, 2001.
- C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.
- J. Ross and J. Dy. Nonparametric mixture of Gaussian processes with constraints. In *International Conference on Machine Learning (ICML)*, pages 1346–1354, 2013.
- Y. Saatchi. *Scalable Inference for Structured Gaussian Process Models*. PhD thesis, University of Cambridge, 2011.
- A. M. Schmidt and A. O’Hagan. Bayesian inference for nonstationary spatial covariance structure via spatial deformations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(3):743–758, 2003.
- M. Seeger. Bayesian Gaussian process models: Pac-Bayesian generalisation error bounds and sparse approximations. Technical report, 2003a.
- M. Seeger. PAC-Bayesian generalisation error bounds for Gaussian process classification. *Journal of Machine Learning Research*, 3:233–269, 2003b.
- A. Shah, A. G. Wilson, and Z. Ghahramani. Student-t processes as alternatives to Gaussian processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 877–885, 2014.
- B. W. Silverman. Some Aspects of the Spline Smoothing Approach to Non-Parametric Regression Curve Fitting. *Journal of the Royal Statistical Society. Series B (Methodological)*, 47(1):1–52, 1985.
- A. J. Smola and P. Bartlett. Sparse greedy Gaussian process regression. In *Advances in Neural Information Processing Systems (NIPS)*, pages 619–625. MIT Press, 2001.
- E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1257–1264, 2006.

- The GPY authors. GPY: A Gaussian process framework in python. <http://github.com/sheffieldml/gpy>, 2012–2016.
- V. Tresp. A Bayesian Committee Machine. *Neural Computation*, 12(11):2719–2741, 2000.
- V. Tresp. Mixtures of Gaussian processes. In *Advances in Neural Information Processing Systems (NIPS)*, pages 654–660, 2001.
- A. Vervuur and I. Karatzas. Diversity-weighted portfolios with negative parameter. *Annals of Finance*, 11(3):411–432, 2015.
- C. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems (NIPS)*, pages 682–688, 2001.
- A. G. Wilson and R. P. Adams. Gaussian process kernels for pattern discovery and extrapolation. In *International Conference on Machine Learning (ICML)*, pages 1067–1075, 2013.
- A. G. Wilson and H. Nickisch. Kernel interpolation for scalable structured Gaussian processes. In *International Conference on Machine Learning (ICML)*, pages 1775–1784, 2015.
- A. G. Wilson, E. Gilboa, and J. P. Nehorai. A. and Cunningham. Fast kernel learning for multidimensional pattern extrapolation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3626–3634, 2014.
- Z. Yang, A. Smola, L. Song, and A. G. Wilson. A la carte – learning fast kernels. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1098–1106, 2015.

Extracting PICO Sentences from Clinical Trial Reports using *Supervised Distant Supervision*

Byron C. Wallace

*College of Computer and Information Science
Northeastern University
Boston, MA, USA*

BYRON@CCS.NEU.EDU

Joël Kuiper

*Doctor Evidence
Santa Monica, CA, USA*

JKUIPER@DOCTOREVIDENCE.COM

Aakash Sharma

*Department of Chemistry
University of Texas at Austin
Austin, TX, USA*

A.SHARMA@UTEXAS.EDU

Mingxi (Brian) Zhu

*Department of Computer Science
University of Texas at Austin
Austin, TX, USA*

BRIAN.ZHU@UTEXAS.EDU

Iain J. Marshall

*Department of Primary Care & Public Health Sciences, Faculty of Life Sciences & Medicine
King's College London
London, UK*

IAIN.MARSHALL@KCL.AC.UK

Editor: Benjamin M. Marlin, C. David Page, and Suchi Saria

Abstract

Systematic reviews underpin Evidence Based Medicine (EBM) by addressing precise clinical questions via comprehensive synthesis of all relevant published evidence. Authors of systematic reviews typically define a Population/Problem, Intervention, Comparator, and Outcome (a *PICO* criteria) of interest, and then retrieve, appraise and synthesize results from all reports of clinical trials that meet these criteria. Identifying *PICO* elements in the full-texts of trial reports is thus a critical yet time-consuming step in the systematic review process. We seek to expedite evidence synthesis by developing machine learning models to automatically extract sentences from articles relevant to *PICO* elements. Collecting a large corpus of training data for this task would be prohibitively expensive. Therefore, we derive *distant supervision* (DS) with which to train models using previously conducted reviews. DS entails heuristically deriving 'soft' labels from an available structured resource. However, we have access only to unstructured, free-text summaries of *PICO* elements for corresponding articles; we must derive from these the desired sentence-level annotations.

To this end, we propose a novel method – *supervised distant supervision* (SDS) – that uses a small amount of direct supervision to better exploit a large corpus of distantly labeled instances by *learning* to pseudo-annotate articles using the available DS. We show that this approach tends to outperform existing methods with respect to automated *PICO* extraction.

Keywords: Evidence-based medicine, distant supervision, data extraction, text mining, natural language processing

1. Introduction and Motivation

Evidence-based medicine (EBM) looks to inform patient care using the totality of the available evidence. Typically, this evidence comprises the results of Randomized Control Trials (RCTs) that investigate the efficacy of a particular treatment (or treatments) in people with a specific clinical problem. *Systematic reviews* are transparently undertaken, rigorous statistical syntheses of such evidence; these underpin EBM by providing quantitative summaries of the entirety of the current evidence base pertaining to particular conditions, treatments and populations.

Systematic reviews are especially critical in light of the data deluge in biomedicine: over 27,000 clinical trials were published in 2012 alone, or roughly 74 per day on average (Bastian et al., 2010). There is thus simply no way that a physician could keep current with the body of primary evidence. Reviews mitigate this problem by providing up-to-date, comprehensive summaries of all evidence addressing focused clinical questions. These reviews are considered the highest level of evidence and now inform all aspects of healthcare, from bedside treatment decisions to national policies and guidelines.

However, the same deluge of clinical evidence that has made reviews indispensable has made producing and maintaining them increasingly onerous. An estimate from 1999 suggests that producing a single review requires thousands of person hours (Allen and Olkin, 1999); this has surely increased since. Producing and keeping evidence syntheses current is thus hugely expensive, especially because reviews are performed by highly-trained individuals (often doctors). Machine learning methods to automate aspects of the systematic review process are therefore needed if EBM is to keep pace with the torrent of newly published evidence (Tsafnat et al., 2013; Bastian et al., 2010; Elliott et al., 2014; Wallace et al., 2013).

A cornerstone of the systematic review paradigm is the notion of precise clinical questions. These are typically formed by decomposing queries into *PICO frames* that define the Population, Intervention, Comparison, and Outcome of interest. Interventions and comparison treatments (e.g., placebo) are often discussed together: we therefore group I and C for the remainder of this paper, and refer to these jointly as simply *interventions*. Once specified, these criteria form the basis for retrieval and inclusion of published evidence in a systematic review. The *PICO* framework is an invaluable tool in the EBM arsenal generally (Huang et al., 2006), and is specifically a pillar of the systematic review process.

Unfortunately, results from RCTs are predominantly disseminated as unstructured free text in scientific publications. This makes identifying relevant studies and extracting the target data for evidence syntheses burdensome. For example, free text does not lend itself to structured search over PICO elements. Structured PICO summaries of articles describing clinical trials would vastly improve access to the biomedical literature base. Additionally, methods to extract PICO elements for subsequent inspection could facilitate inclusion assessments for systematic reviews by allowing reviewers to rapidly judge relevance with respect to each PICO element. Furthermore, automated PICO identification could expedite *data extraction* for systematic reviews, in which reviewers manually extract structured data to be reported and synthesized. Consider the task of extracting dosage information for a given clinical trial: currently reviewers must identify passages in the article that discuss the interventions and then extract from these the sought after information. This is time-consuming and tedious; a tool that automatically identified PICO related sentences and guided the reviewer to these would expedite data extraction.

In this work we present a novel machine learning approach that learns to automatically extract sentences pertaining to PICO elements from full-text articles describing RCTs. We exploit an existing (semi-)structured resource – the Cochrane Database of Systematic Reviews (CDSR) – to derive *distant supervision* (DS) with which to train our PICO extraction model. DS is generated by using heuristics to map from existing structured data \mathcal{D} to pseudo-annotations that approximate the target labels \mathcal{Y} . These derived labels will be imperfect, because the structured data to which we have access comprises free-text summaries describing each PICO element; this text does not appear verbatim in the corresponding articles. Thus, using simple string matching methods to induce supervision will introduce noise. We therefore propose a new method that *learns* to map from \mathcal{D} to \mathcal{Y} using a small amount of direct supervision, thus deriving from the free-text summaries in the CDSR the desired sentence-level annotations. We refer to this as *supervised distant supervision* (SDS).

We empirically evaluate our approach both retrospectively (using previously collected data) and via a prospective evaluation. We demonstrate that SDS consistently improves performance with respect to baselines that exploit *only* distant or (a small amount of) direct supervision. We also show that our flexible SDS approach performs at least as well – and usually better – than a previously proposed model for jointly learning from distant and direct supervision. While our focus here is on the particular task of PICO identification in biomedical texts, we believe that the proposed SDS method represents a generally useful new paradigm for distantly supervised machine learning.

The remainder of this paper is structured as follows. We review related work in the following section. We introduce our source of distant supervision, the CDSR, in Section 3. This motivates the development of our SDS model, which we present in Section 4. We discuss experimental details (including features and baseline methods to which we compare) in Section 5, and report experimental results in Section 6. Finally, we conclude with additional discussion in Section 7.

2. Related Work

We briefly review two disparate threads of related work: automatic identification of PICO elements for EBM (Section 2.1) and work on *distant supervision* (Section 2.2), paying particular attention to recent efforts to develop models that combine distant and direct supervision.

2.1 Automatic PICO Identification

The practical need for language technologies posed by EBM-related tasks has motivated several recent efforts to identify PICO elements in biomedical text (Denner-Fushman and Lin, 2007; Chung, 2009; Boudin et al., 2010b,a; Kim et al., 2011). However, nearly all of these works have considered only the abstracts of articles, limiting their utility. Such approaches could not be used, for example, to support data extraction for systematic reviews, because clinically salient data is often not available in the abstract. Furthermore, it is likely that identifying PICO elements in the full-texts of articles could support rich information retrieval support, beyond what is achievable using abstracts alone.

Nonetheless, identifying PICO sentences in abstracts has proven quite useful for supporting biomedical literature retrieval. For example, Denner-Fushman and Lin (2007) developed and evaluated a tool that extracts clinically salient snippets (including PICO elements) from MEDLINE abstracts. They showed that these extractions can assist with information retrieval and clinical question answering. Similarly, Boudin et al. (2010b,a) showed that automatically generated PICO annotation of abstracts can improve biomedical information retrieval, even if these annotations are noisy.

Moving beyond abstracts, one system that does operate over full texts to summarize clinical trials is ExaCT (Kiritchenko et al., 2010). ExaCT aims to extract variables describing clinical trials. It requires HTML or XML formatted documents as input. The system splits full-text articles into sentences and classifies these as *relevant* or *not* using a model trained on a small set (132) of manually annotated articles. ExaCT does not attempt to identify PICO sentences, but rather aims to map directly to a semi-structured template describing trial attributes. The work is therefore not directly comparable to the present effort.

Our work here differs from the efforts just reviewed in a few key ways:

1. In contrast to previous work, we aim to identify sentences in *full-text* articles that are pertinent to PICO elements. This may be used to facilitate search, but we are more immediately interested in using this technology to semi-automate data extraction for systematic reviews.
2. Previous work has leveraged small corpora (on the order of tens to hundreds of manually annotated abstracts) to train machine learning systems. By contrast, we exploit a large ‘distantly supervised’ training corpus derived from an existing database. In Section 6 we demonstrate the advantage of this novel approach, and show that using a small set of direct supervision alone fares comparatively poorly here.

Additionally, we introduce a novel paradigm for distantly supervised machine learning, which we review next.

2.2 Distant Supervision

Distant supervision (DS) refers to learning from indirect or weak supervision derived from existing structured resources. These derived ‘labels’ are often noisy, i.e., imperfect. But the advantage is that by exploiting existing resources one can capitalize on a potentially large labeled training dataset effectively ‘for free’. The general approach in DS is to develop heuristics to map existing, structured resources onto the target labels of interest and then use these derived labels to train a model (Figure 1a).

This paradigm was first introduced by Craven and Kumlien (1999) in their work on building models for information extraction for biological knowledge base construction.¹ Specifically they considered the task of extracting relationships between biological entities, such as subcellular-structures and proteins. To generate (noisy) training data for this task they exploited the Yeast Protein Database (YPD), which contains propositions expressing relationships of interest between pairs of biological entities. For each known relationship expressed in the YPD they searched PubMed, a repository of biomedical literature, to identify abstracts that mentioned both entities. They made the simplifying assumption that any such co-occurrence expressed the target relationship (this being the heuristic means of inducing positive instances). They demonstrated that training their model with these pseudo-positive instances resulted in performance comparable to models trained using manually labeled examples.

Much of the more recent work on distant supervision since has been focused on the task of *relation extraction* (Mintz et al., 2009; Nguyen and Moschitti, 2011; Riedel et al., 2010; Bunesco and Mooney, 2007; Angeli et al., 2014) and classification of Twitter/microblog texts (Purver and Battersby, 2012; Marchetti-Bowick and Chambers, 2012). Our focus here aligns with previous attempts to reduce the noise present in distantly labeled datasets, although so far as we are aware these have been exclusively applied for the task of relation extraction (Roth et al., 2013). These methods have tended to exploit a class of generative *latent-variable* models specifically developed for the task of relation extraction (Surdeanu et al., 2012; Min et al., 2013; Takamatsu et al., 2012; Angeli et al., 2014). Unfortunately, these models do not naturally generalize to other tasks because they are predicated on the assumption that the structured resource to be exploited comprises *entity-pairs* to be identified in unlabeled instances. Such entity-pairs have no analog in the case of sentence extraction. For example, Angeli et al. (2014) combine direct and distant supervision for relation extraction by building on the Multi-Instance Multi-Label Learning (MIML-RE) originally proposed by Surdeanu et al. (2012). They estimate the parameters in a fully generative model that includes variables corresponding to entities and their co-occurrences in

1. Craven and Kumlien called this ‘weakly supervised’ learning. The term ‘distant supervision’ was later coined by Mintz et al. (2009).

Target description from the CDSR. Patients ($n = 24$, 15 females) with neck pain of > 3 months’ duration, who had pain in one or more cervical (C3-C7) zygapophysial joints after a car accident and whose pain perception had been confirmed by placebo-controlled diagnostic blocks.

C1: The study patients were selected from among patients whose cervical zygapophysial-joint pain had been confirmed with the use of local anesthetic blocks at either the unit or a private radiology practice in Newcastle.

C2: We studied 24 patients (9 men and 15 women; mean age, 43 years) who had pain in one or more cervical zygapophysial joints after an automobile accident (median duration of pain, 34 months).

C3: The significant rate of response to the control treatment, even among patients who had been tested with placebo-controlled diagnostic blocks to confirm their perceptions of pain, is a sobering reminder of the complex and inconstant dynamics of placebo phenomena.

Table 1: Example *population* target text (summary) from the CDSR and three candidate sentences from the corresponding full-text article generated via distant supervision.

texts. It is not clear how one might modify this model to accommodate our task of *sentence extraction*.

Here we will therefore be interested in guiding DS for general learning tasks using a small set of direct annotations. Most relevant to our work is therefore that of Nguyen and Moschitti (2011), in which they proposed a general method for combining direct and distant supervision. Their approach involves training two conditional models: one trained on directly labeled instances and the other on a mixed set comprising both directly and distantly labeled examples. They then linearly combine probability estimates from these classifiers to produce a final estimate. The key point here is that the derivation of the DS – i.e., the process of moving from extant data to noisy, distant labels – was still a heuristic procedure in this work. By contrast, we propose *learning* an explicit mapping from directly labeled data to distant labels, as we discuss further in Section 4.

3. Learning from the Cochrane Database of Systematic Reviews

We next describe the Cochrane Database of Systematic Reviews (CDSR) (The Cochrane Collaboration, 2014), which is the database we used to derive DS.

3.1 PICO and the CDSR

The CDSR is produced and maintained by the *Cochrane Collaboration*, a global network of 30,000+ researchers who work together to produce systematic reviews. The group has collectively generated nearly 6,000 reviews, which describe upwards of 50,000 clinical trials. These reviews (and the data extracted to produce them) are published as the CDSR.

The CDSR contains structured and semi-structured data for every clinical trial included in each systematic review. To date we have obtained corresponding full-text articles (PDFs) for 12,808 of the clinical trials included in the CDSR. In previous work (Marshall et al., 2014, 2015) we demonstrated that supervision derived from the CDSR on linked full-text

PICO element	Number of distantly labeled articles
<i>Population</i>	12,474
<i>Intervention</i>	12,378
<i>Outcomes</i>	12,572

Table 2: The number of full-text articles for which a corresponding free-text summary is available in the CDSR for each PICO element (studies overlap substantially); this provides our DS.

articles can be exploited to learn models for automated *risk of bias* (RoB) assessment of clinical trials and supporting sentence extraction. However, in the case of RoB, supervision was comparatively easy to derive from the CDSR: this required only literal string matching, because by convention reviewers often store verbatim sentences extracted from articles that support their RoB assessments. In the case of PICO, however, reviewers generate free-text summaries for each element (not verbatim quotes) that are then stored in the CDSR. Therefore, we must map from these summaries to sentence labels (*relevant* or *not*) for each PICO domain.

Table 1 provides an example of a *population* summary stored in the Cochrane database for a specific study, along with potentially ‘positive’ sentence instances from the corresponding article. Such summaries are typically, but not always, generated for articles and this varies somewhat by PICO element. In Table 2 we report the number of studies for which we have access to human-generated summaries for each PICO element.

Articles used for the population, intervention and outcomes domains were (automatically) segmented into 333, 335 and 338 sentences on average, respectively. We adopted a straightforward heuristic approach to generating DS (in turn generating candidate sets) of sentences using the CDSR. Specifically, for a given article a_i and matched PICO element summary s_i stored in the CDSR, we soft-labeled as positive (designated as candidates) up to k sentences in a_i that were most similar to s_i . To mitigate noise, we introduced a threshold such that candidate sentences had to be at least ‘reasonably’ similar to the CDSR text to be included in the candidate set. Operationally, we ranked all sentences in a given article with respect to the raw number of word (nigram) tokens shared with the CDSR summary, excluding stop words. The top 10 sentences that shared at least 4 tokens with the summary were considered positive (members of the candidate set). These were somewhat arbitrary decisions reflecting intuitions gleaned through working with the data; other approaches to generating candidate sets could have of course been used here instead. However, it is likely that any reasonable heuristic based on token similarity would result in DS with similar properties.

7

3.2 Annotation

We labeled a subset of the candidate sets generated via DS from the CDSR for two reasons: (1) this constitutes the *direct* supervision which we aim to combine with DS to train an accurate model; and, (2) cross-fold validation using these labels may be used as a proxy evaluation for different models. We say ‘proxy’ because implicitly we assume that *all* sentences not among the candidate sets are true negatives, which is almost certainly not the case (although given the relatively low threshold for inclusion in the candidate set, this assumption is not entirely unreasonable).

The annotation process involved rating the quality of automatically derived candidate sentences for each PICO element and article. Annotations were on a 3-point scale designed to differentiate between *irrelevant*, *relevant* and *best available* candidate sentences (coded as 0, 1 and 2, respectively). This assessment was made in light of the corresponding summaries for each PICO field published in the CDSR. In Table 1, we show three candidate sentences (distantly labeled ‘positive’ instances) and the target summary. Here, candidate 1 (C_1) is *relevant*, C_2 is the *best available* and C_3 is in fact *irrelevant*.

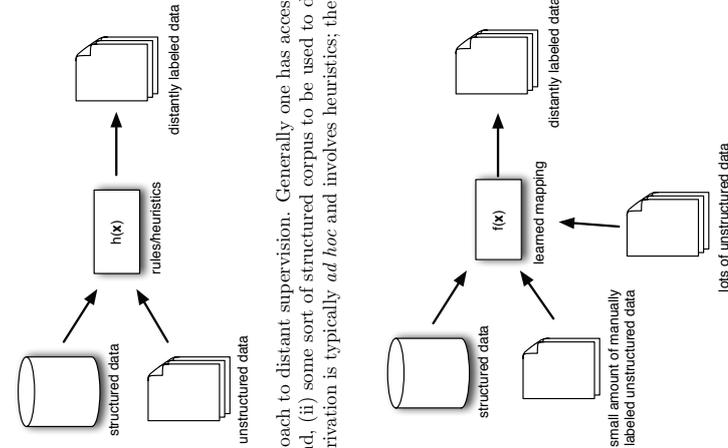
Two of the co-authors (BZ and AS) worked with BW to develop and refine the labeling scheme. This refinement process involved conducting a few pilot rounds to clarify labeling criteria.² We conducted these until what we deemed acceptable pairwise agreement was reached, and subsequently discarded the annotations collected in these early rounds. After this pilot phase, a subset of 1,071 total candidate sentences were labeled independently by both annotators. Additional sentences were later labeled individually. On the multiply labeled subset, observed annotator agreement was high: pairwise $\kappa = 0.74$ overall, and $\kappa = 0.81$ when we group *relevant* sentences with *best available* – in practice, we found distinguishing between these was difficult and so we focus on discriminating between *irrelevant* and *relevant/best available* sentences. Ultimately, we acquired a set of 2,821 labels on sentences from 133 unique articles; these comprise 1009, 1006 and 806 sentences corresponding to ‘participants’, ‘interventions’ and ‘outcomes’, respectively.

4. Supervised Distant Supervision

We now describe the novel approach of *supervised distant supervision* (SDS) that we propose for capitalizing on a small set of directly labeled candidate instances in conjunction with a large set of distantly supervised examples to induce a more accurate model for the target task. Figure 1b describes the idea at a high-level. The intuition is to train a model that maps from the heuristically derived and hence noisy DS to ‘true’ target labels. This may be viewed as learning a filtering model that winnows a candidate set of positive instances automatically generated via DS to a higher-precision subset of (hopefully) true positive

² The annotation guideline developed during our pilot annotation phase is available at: <http://byron.ischool.utexas.edu/stratic/sds-guidelines.pdf>

8



(a) The standard approach to distant supervision. Generally one has access to (i) a (large) set of unlabeled instances and, (ii) some sort of structured corpus to be used to derive distant labels on said instances. This derivation is typically *ad hoc* and involves heuristics; the derived labels are thus usually noisy.

(b) The proposed *supervised distant supervision* (SDS) approach. We aim to leverage a small amount of annotated data – which provides alignments between unlabeled instances with the structured corpus to be used to derive distant labels – to induce a model that maps that from paired entries in the available structured data and unlabeled corpus to the target labels on the latter.

Figure 1: Standard distant supervision (top) and the proposed *supervised distant supervision* approach (bottom).

instances, using attributes derived from instances and the available distant supervision on them.

We will denote instances (documents) by $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$. Each $\mathbf{x}_i \in \mathcal{X}$ comprises m_i sentences: $\{\mathbf{x}_{i,1}, \dots, \mathbf{x}_{i,m_i}\}$. We will index sentences by j , so $\mathbf{x}_{i,j}$ denotes the (vector

representation of) sentence j in document i . We treat the sentence extraction tasks for the respective PICO elements as independent, and therefore do not introduce notation to differentiate between them.

We will denote the database of semi-structured information from which we are to derive DS by \mathcal{D} . We assume that \mathcal{D} contains an entry for all n linked articles under consideration. We denote the set of distantly derived labels on sentences by $\tilde{\mathcal{Y}} = \{\tilde{y}_{1,1}, \dots, \tilde{y}_{1,m_1}, \dots, \tilde{y}_{n,1}, \dots, \tilde{y}_{n,m_n}\}$, and corresponding true target labels by $\mathcal{Y} = \{y_{1,1}, \dots, y_{1,m_1}, \dots, y_{n,1}, \dots, y_{n,m_n}\}$. The former are assumed to have been derived from \mathcal{D} via the heuristic labeling function h , while the latter are assumed to be unobserved. In DS one generally hopes that $\tilde{\mathcal{Y}}$ and \mathcal{Y} agree well enough to train a model that can predict target labels for future examples.

Our innovation here is to exploit a small amount of direct supervision to *learn* a model to improve DS by filtering the candidates generated by h using a function f that operates over features capturing similarities between entries in \mathcal{D} and instances to generate a more precise label set. Specifically we aim to learn a function $f : (\mathcal{X}, \tilde{\mathcal{Y}}) \rightarrow \mathcal{Y}$, where we have introduced new instance representations \mathcal{X} which incorporate features derived from pairs of instances and database entries (we later enumerate these). We emphasize that this representation differs from \mathcal{X} , which cannot exploit features that rely on \mathcal{D} because DS will not generally be available for new instances. The parameters of f are to be estimated using a small amount of direct (manual) supervision which we will denote by \mathcal{L} . These labels indicate whether or not distantly derived labels are correct. Put another way, this is *supervision for distant supervision*.

We will assume that the heuristic function h can generate a *candidate set* of positive instances, many of which will in fact be negative. This assumption is consistent with previous efforts (Bunescu and Mooney, 2007). In our case, we will have a candidate set of sentence indices \mathcal{C}_i associated with each entry i in \mathcal{D} (note that we will have different candidate sets for each PICO element, but the modeling approach will be the same for each). These are the sentences for which \tilde{y} is positive. The supervision \mathcal{L} will comprise annotations on entries in these candidate sets with respect to target labels y . Thus the learning task that we will be interested in is a mapping between $\mathcal{C}_1, \dots, \mathcal{C}_l$ and corresponding target label sets $\mathcal{Y}_1, \dots, \mathcal{Y}_l$.

4.1 Intuition

To better motivate this SDS approach, consider a scenario in which one has access to a (very) large set of unlabeled instances \mathcal{X} and a database \mathcal{D} from which noisy, distant supervision $\tilde{\mathcal{Y}}$ may be derived (along with feature vectors jointly describing instances and their entries in \mathcal{D} , \mathcal{X}). In such scenarios, we will be able to efficiently generate a very large training set for ‘free’ by exploiting \mathcal{D} ; hence the appeal of DS. However, if our rule h for deriving $\tilde{\mathcal{Y}}$ is only moderately accurate, we may be introducing excessive noise into the training set, in turn hindering model performance. At the same time, it may be that one

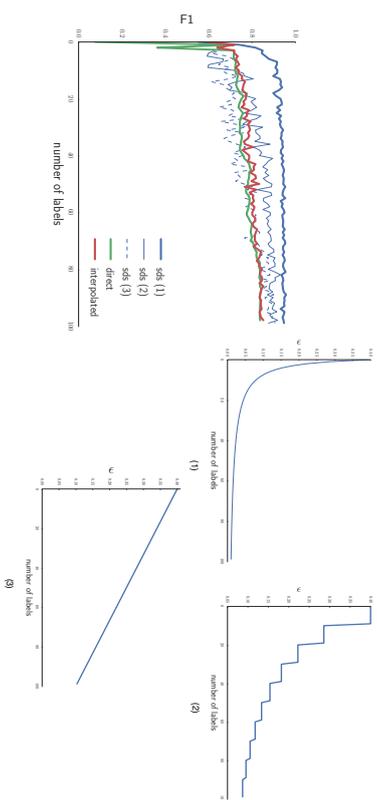


Figure 2: Plots from a simulation highlighting the intuition behind SDS. We consider different learning rates for the SDS task: e.g., in scenario 1 we assume the error ϵ in the distantly derived labels can be reduced drastically with relatively few direct labels. Put another way, this assumes that an accurate heuristic is relatively easy to learn. On this assumption, performance (F1) on the target task can be improved drastically compared to the alternative approach of, e.g., interpolating models trained on the distant and direct label sets. SDS still performs well under less optimistic assumptions, as can be seen in scenarios 2 and 3, which assume step and linear reduction relationships between ϵ and the number of labels provided, respectively. See the text for additional explanation.

can dramatically improve the pseudo-labeling accuracy by *learning* a mapping from a small amount of direct supervision, \mathcal{L} . Providing supervision for the mapping, rather than the actual task at hand, may be worthwhile if the former allows us to effectively exploit the large set of distantly labeled data.

To make this intuition more concrete, we conducted an illustrative simulation experiment using the classic twenty-newsgroups corpus.³ We used the standard train and test splits of the data, and consider the binary classification task of discriminating between messages from the *alt.athletics* and those from the *soc.religion.christian* boards. This subset comprises 1079 messages in the training set and 717 in the testing set.

We assume that a DS heuristic h assigns the true label with probability $1-\epsilon$; thus ϵ encodes the noise present in $\tilde{\mathcal{Y}}$. Intuitively, SDS will work well when we can efficiently learn a model that operates over $\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}$ to better approximate the true labels \mathcal{Y} , i.e., reduce ϵ . This may be possible when the features comprising \mathcal{X} are predictive. We assume $\epsilon = 0.4$ for

3. <http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html>

DS to begin with and we consider three scenarios which differ in their assumed relationship between the number of annotations and the induced reduction in ϵ . Respectively, these simulations assume: (1) a smooth and (2) step-wise exponentially decreasing function (representing rapid learning rates, implying that $\tilde{\mathcal{X}}$ is rich with signal), and (3) a linearly decreasing function. These simulated learning rates are depicted in Figure 2.

We report the performance (F1 score, i.e., the harmonic mean of precision and recall) on the held-out test data using SDS under these three scenarios. That is, we re-label the instances in the large training corpus with noise equal to the corresponding ϵ under each scenario; this simulates re-labeling the (distantly supervised) training corpus using the trained SDS model. We compare this to using direct supervision (no noise) only and to interpolating independent predictions from models trained on direct and distant supervision, respectively (see Section 5.2). We allowed the latter model access to a small validation set with which to tune the interpolation parameter. We simulated annotating (directly, with no noise) up to 100 instances and show learning curves under each strategy/scenario.

As one would expect, SDS works best when one can efficiently reduce the noise in the relatively large set of distantly labeled instances, as in simulations (1) and (2), which assume noise exponentially decays with labeled instances. In these cases, effort is best spent on learning a model to reduce ϵ . However, note that even when the signal is not quite as strong – as in scenario 3 where we assume a linear relationship between noise reduction and collected annotations – we can see that the SDS model ultimately outperforms the other approaches. The comparative advantage of the SDS strategy will depend on the noise introduced by DS to begin with and the efficiency with which a model can be learned to reduce this noise. We emphasize that this scenario is intended to highlight the intuition behind SDS and scenarios in which it might work well, not necessarily to provide empirical evidence for its efficacy.

4.2 Model

We now turn to formally defining an SDS model. This entails first specifying the form of f . Here we use a log-linear model to relate instances comprising candidate sets to their associated label qualities. Specifically, we assume:

$$\hat{p}_{i,j}^{sds} = p(y_{i,j} | \mathcal{C}_i, \tilde{\mathbf{w}}) = \begin{cases} \propto \exp(\tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}}_{i,j}) & \text{if } j \in \mathcal{C}_i \text{ (i.e., } \tilde{y}_{i,j} = 1) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $\tilde{\mathbf{w}}$ is a weight vector to be estimated from the training data \mathcal{L} .⁴ More precisely, we use regularized logistic regression as our conditional probability model for instances comprising the candidate set. Note that for brevity we will denote the estimated conditional

4. Note that $\tilde{\mathbf{w}}$ differs from the weight vector parameterizing the final model, \mathbf{w} , because the former comprises coefficients for features in $\tilde{\mathcal{X}}$ which are at least partially derived from information in the available structured resource. These would not be available at test-time (i.e., in \mathcal{X}).

probability for sentence j in document i by $\hat{p}_{i,j}^{sds}$. The idea is that once we have estimated $\tilde{\mathbf{w}}$ (and hence $\hat{p}_{i,j}^{sds}$ for all i and j) we can use this to improve the quality of DS by effectively filtering the candidate sets.

Consider first a standard objective that aims to directly estimate the parameters \mathbf{w} of a linear model for the target task relying only on the distant supervision:

$$\operatorname{argmin}_{\mathbf{w}} \mathcal{R}(\mathbf{w}) + C \sum_{i=1}^n \sum_{j=1}^{m_i} \operatorname{loss}(\mathbf{w} \cdot \mathbf{x}_{i,j}, \tilde{y}_{i,j}) \quad (2)$$

where a loss function (e.g., hinge or log loss) is used to incur a penalty for disagreement between model predictions and the derived (distant) labels, \mathcal{R} is a regularization penalty (such as the squared ℓ_2 norm) and C is a scalar encoding the emphasis placed on minimizing loss versus achieving model simplicity. We will be concerned primarily with the parameterization of the *loss* function here, and therefore omit the regularization term (and associated hyper-parameter C) for brevity in the following equations.

Again grouping all distantly labeled ‘positive’ sentences for document i in the set \mathcal{C}_i and decomposing the loss into that incurred for false negatives and false positives, we can re-write this as:

$$\sum_{i=1}^n \left\{ \sum_{j \in \mathcal{C}_i} c_{fn} \cdot \operatorname{loss}(\mathbf{w} \cdot \mathbf{x}_{i,j}, 1) + \sum_{j \notin \mathcal{C}_i} c_{fp} \cdot \operatorname{loss}(\mathbf{w} \cdot \mathbf{x}_{i,j}, -1) \right\} \quad (3)$$

Where we are denoting the cost of a false negative by c_{fn} and the cost of a false positive by c_{fp} . Minimizing this objective over \mathbf{w} provides a baseline approach to learning under DS.

We propose an alternative objective that leverages the mapping model discussed above (Equation 1). The most straight-forward approach would be to use binary (0/1) classifier output to completely drop out instances in the candidate set that are deemed likely to be irrelevant by the model, i.e.:

$$\sum_{i=1}^n \left\{ \sum_{j \in \mathcal{C}_i} c_{fn} \cdot \operatorname{sign}^{0/1}(\tilde{\mathbf{w}} \cdot \tilde{\mathbf{x}}_{i,j}) \cdot \operatorname{loss}(\mathbf{w} \cdot \mathbf{x}_{i,j}, 1) + \sum_{j \notin \mathcal{C}_i} c_{fp} \cdot \operatorname{loss}(\mathbf{w} \cdot \mathbf{x}_{i,j}, -1) \right\} \quad (4)$$

Where $\operatorname{sign}^{0/1}$ denotes a sign function that returns 0 when its argument is negative and 1 otherwise. We take a finer-grained approach in which we scale the contribution to the total loss due to ‘positive’ instances by probability estimates that these indeed represent true positive examples, conditioned on the available distant supervision:

$$\sum_{i=1}^n \left\{ \sum_{j \in \mathcal{C}_i} c_{fn} \cdot \hat{p}_{i,j}^{sds} \cdot \operatorname{loss}(\mathbf{w} \cdot \mathbf{x}_{i,j}, 1) + \sum_{j \notin \mathcal{C}_i} c_{fp} \cdot \operatorname{loss}(\mathbf{w} \cdot \mathbf{x}_{i,j}, -1) \right\} \quad (5)$$

We extend this objective to penalize more for mistakes on explicitly labeled instances. Recall that we denote by \mathcal{L} the small set of directly annotated articles; here we assume that this set comprises indices of directly labeled articles. Let us also denote by \mathcal{L}_i^+ and \mathcal{L}_i^- the set of positive and negative sentence indices for labeled article i , respectively. Further, denote by $\tilde{\mathcal{L}}$ the set of article indices for which we *only* have distant supervision (so that $\mathcal{L} \cap \tilde{\mathcal{L}} = \emptyset$ by construction). Putting everything together forms our complete objective:

$$\operatorname{argmin}_{\mathbf{w}} \mathcal{R}(\mathbf{w}) + C \left(\lambda \sum_{i \in \mathcal{L}} \left\{ \sum_{j \in \mathcal{L}_i^+} c_{fn} \cdot \operatorname{loss}(\mathbf{w} \cdot \mathbf{x}_{i,j}, 1) + \sum_{j \in \mathcal{L}_i^-} c_{fp} \cdot \operatorname{loss}(\mathbf{w} \cdot \mathbf{x}_{i,j}, -1) \right\} + \sum_{i \in \tilde{\mathcal{L}}} \left\{ \sum_{j \in \mathcal{C}_i} c_{fn} \cdot \hat{p}_{i,j}^{sds} \cdot \operatorname{loss}(\mathbf{w} \cdot \mathbf{x}_{i,j}, 1) + \sum_{j \notin \mathcal{C}_i} c_{fp} \cdot \operatorname{loss}(\mathbf{w} \cdot \mathbf{x}_{i,j}, -1) \right\} \right) \quad (6)$$

Here we used log loss throughout and ℓ_2 regularization for the penalty \mathcal{R} . The λ and C are hyper-parameters to be tuned via grid-search (details in Section 5.3).

The key element of this objective is the use of the $\hat{p}_{i,j}^{sds}$ (Equation 1) estimates to scale loss contributions from distantly supervised data. This is particularly important because in general there will exist far more distantly supervised instances than directly labeled examples, i.e., $|\tilde{\mathcal{L}}| \gg |\mathcal{L}|$. One practical advantage of this approach is that once training is complete, the model is defined by a single weight-vector \mathbf{w} , even though two models, parameterized independently by \mathbf{w} and $\tilde{\mathbf{w}}$ are used during training.

Recall that $\hat{p}_{i,j}^{sds}$ estimates the probability of a candidate sentence (potentially positive instance, as per the distant supervision heuristic h) indeed being a ‘true’ positive. As mentioned above, the feature space that we use for this task can differ from the feature space used for the target task. That is, the attributes comprising \mathcal{X} need not be the same as those in $\tilde{\mathcal{X}}$. Indeed, features in $\tilde{\mathcal{X}}$ should capture signal gleaned from attributes derived via the available distant supervision \mathcal{D} for any given instance, but at test time we would not be able to capitalize on such features. In the next section we describe the features we used for PICO sentence classification, both for \mathcal{X} and $\tilde{\mathcal{X}}$.

5. Experimental Details and Setup

5.1 Features

Table 3 enumerates the feature sets we use. All models leverage those in the top part of the table. The bottom part describes those features that are derived using \mathcal{D} , our source of DS. Therefore, these are only used in the SDS approach, and only present in $\tilde{\mathcal{X}}$.

5.2 Baselines

We compare the proposed *supervised distant supervision* (SDS) approach to the following baseline methods:

Feature	Description
Bag-of-Words	Term-Frequency Inverse-Document-Frequency (TF-IDF) weighted uni- and bi-gram count features extracted for each sentence. We include up to 50,000 unique tokens that appear in at least three unique sentences.
Positional	Indicator variable coding for the decile (with respect to length) of the article where the corresponding sentence is located.
Line lengths	Variables indicating if a sentence contains 10%, 25% or a greater percentage of 'short' lines (operationally defined as comprising 10 or fewer characters); a heuristic for identifying tabular data
Numbers	Indicators encoding the fraction of numerical tokens in a sentence (fewer than 20% or fewer than 40%).
New-line count	Binned indicators for new-line counts in sentences. Bins were: 0-1, fewer than 20 and fewer than 40 new-line characters.
Drugbank	An indicator encoding whether the sentence contains any known drug names (as enumerated in a stored list of drug names from http://www.drugbank.ca/).
Shared tokens	<i>Additional features used for SDS task (enclosed by X)</i> TF-IDF weighted features capturing the uni- and bi-grams present both in a sentence and in the Cochrane summary for the target field.
Relative similarity score	'Score' (here, token overlap count) for sentences with respect to target summary in the CDSR. Specifically, we use the score for the sentence minus the average score over all candidate sentences.

Table 3: Features we used for the target learning tasks and additional features we used in learning to map from candidate sets (the distant supervision) to 'true' labels. We set discrete ('binned') feature thresholds heuristically, reflecting intuition; we did not experiment at length with alternative coding schemes. Note that separate models were learned for each PICO domain.

- Distant supervision only (DS) (Mintz et al., 2009; Craven and Kunhlen, 1999). This simply relies on the heuristic labeling function h . We define the corresponding objective formally in Equation 3. We also experimented with a variant that naively incorporates the direct labels when available, but does not explicitly distinguish these from the distant labels. These two approaches performed equivalently, likely due to the relative volume of the distantly labeled instances.
- Direct supervision only. This uses only the instances for which we have direct supervision and so represents standard supervised learning.
- Joint distant and direct supervision, via the pooling method due to Nguyen and Moschitti (2011). In this approach one leverages the direct and indirect supervision to estimate separate (probabilistic) models, and then generates a final predicted probability by linearly interpolating the estimates from the two models:

$$\hat{p}_{k,j}^{\text{pooled}} = \alpha \cdot \hat{p}_{k,j}^{\text{direct}} + (1 - \alpha) \cdot \hat{p}_{k,j}^{\text{distant}} \quad (7)$$

Where α is to be tuned on a validation set (Section 5.3).

These baselines allow us to evaluate (1) whether and to what degree augmenting a large set of DS with a small set of direction annotations can improve model performance; (2) the relative accuracy of the proposed SDS approach, in comparison to the pooling mechanism proposed by Nguyen and Moschitti (2011).

5.3 Parameter Estimation and Hyper-Parameter Tuning

We performed parameter estimation for all models concerned with the target task (i.e., estimating \mathbf{w}) via Stochastic Gradient Descent (SGD).⁵ For all models, class weights were set inversely to their prevalences in the training dataset (mistakes on the rare class – positive instances – were thus more severely penalized). For distant and direct only models, we conducted a line-search over C values from 10 up to 10^5 , taking logarithmically spaced steps. We selected from these the value that maximized the harmonic mean of precision and recall (F1 score); this was repeated independently for each fold.

SDS. For the SDS model (Equation 6) we performed grid search over λ and C values. Specifically we searched over $\lambda = \{2, 10, 50, 100, 200, 500\}$ and the same set of C values specified above. For each point on this grid, we assessed performance with respect to squared error on a validation set comprising 25% of the available training data for a given fold. We kept the λ and C values that minimized expected squared error

$$\beta \{y_{k,j} = 1 | \hat{\mathbf{w}}, \mathbf{x}_{k,j}\} - \text{sign}^{0/1}(y_{k,j})^2 \quad (8)$$

⁵ Specifically, we used the implementation in the Python machine learning library `scikit-learn` (Pedregosa et al., 2011) v0.17, with default estimation parameters save for `class_weight` which we set to 'balanced'.

Where $\hat{p}(y_{i,j} = 1|\hat{\mathbf{w}})$ denotes the predicted probability of sentence j in article i being relevant – that is, predicted by the linear model for the target task where $\hat{\mathbf{w}}$ has been selected to maximize the objective parameterized by a specific pair of (λ, C) values. We emphasize that this estimated probability is with respect to the target label, and thus differs from the $\hat{p}_{i,j}^{sds}$ defined in Equation 1, which relies on an estimate of $\tilde{\mathbf{w}}$. We scaled this per-instance error to account for imbalance, so that the total contribution to the overall error that could be incurred from mistakes made on (the relatively few) positive instances was equal to the potential contribution due to mistakes made on negative examples.

We also note that the parameters of the SDS model (i.e., $\tilde{\mathbf{w}}$ in Equation 1) were estimated using LIBLINEAR (Fan et al., 2008);⁶ for this model we searched over a slightly different range of C values, ranging from 10^0 to 10^4 , taking logarithmically spaced steps.

Nguyen. We performed a line-search for α (Equation 7) ranging from 0 to 1 taking 50 equispaced steps using the same strategy and objective as just described for tuning the SDS hyper-parameters. (Note that the two constituent models that form the Nguyen ensemble have their own respective regularizer and associated scalar hyper-parameter; we tune these independently of α , also via line-search as described above).

5.4 Evaluation and Metrics

We performed both retrospective and prospective evaluation. For retrospective evaluation, we performed cross-fold validation of the directly labeled candidate instances (see Section 3). Note that this evaluation is somewhat noisy, due to the way in which the ‘ground truth’ was derived. Therefore, we also conducted a prospective evaluation, which removed noise from the test set but required additional annotation effort. For the latter evaluation we considered only the two most competitive methods. The top-3 sentences retrieved by each of these methods were directly labeled for relevance, using the same criteria as we used in collecting our direct supervision over candidate instances (Section 3.2). The annotator was blinded to which method selected which sentences.

In our retrospective evaluation, we report several standard metrics: Area Under the Receiver Operating Characteristic Curve (AUC) to assess overall discriminative performance and normalized Discounted Cumulative Gain (NDCG) (Järvelin and Kekäläinen, 2000, 2002), which incorporates relevance scores and discounts relative rankings that appear lower in the ranking. More specifically, we report NDCG@20, which evaluates the rankings of the top 20 sentences induced by each method. We also report precision@3, precision@10 and precision@20, which correspond to the fraction of relevant sentences retrieved amongst the top 3, 10 and 20 sentences retrieved by each method, respectively. All metrics are calculated for each article separately and we then report averages and standard deviations over these.

For our prospective evaluation, we report precision@3, which was practical from an annotation vantage point. We allowed the two most competitive models to select three

sentences from as-yet unlabeled articles to be manually assessed for relevance (the assessor was blinded as to which model selected which sentences). We report the average fraction of these (over articles) deemed at least *relevant*.

6. Results

We present retrospective results in Section 6.1 and prospective results in Section 6.2. For the latter, we tasked one of our trained annotators with labeling (for each PICO domain) the three top-ranked sentences selected from 50 held-out articles by the two most competitive approaches, SDS and the pooling approach Nguyen and Moschitti Nguyen and Moschitti (2011).

6.1 Retrospective evaluation

We performed five-fold validation on the 133 articles for which candidate sentences were directly labeled across all three PICO elements (recall that we group Intervention and Comparator together). We treat all explicitly labeled *relevant* and *best available* sentences (as described in Section 3) instances as positive and all other examples as negative, including those that did not score sufficiently high to be included in a candidate set (i.e., distantly labeled negative instances).

We report results averaged over these folds with respect to the metrics discussed in Section 5.4. We report all results observed on the retrospective data in Table 4; we reiterate that these are averages taken across all 133 articles. In general, we note that across all metrics and domains, SDS most often results in the best performance, although the comparative gain is often small.

6. Again with default parameters provided in the interface from scikit-learn (Pedregosa et al., 2011) v0.17, and again specifying a ‘balanced’ *class_weight*.

Method	Mean AUC (SD)	Mean NDCG@20 (SD)	Precision@3 (SD)	Precision@10 (SD)	Precision@20 (SD)
<i>Population</i>					
Direct only	0.904 (0.106)	0.530 (0.270)	0.347 (0.298)	0.183 (0.126)	0.116 (0.070)
DS	0.941 (0.063)	0.484 (0.243)	0.256 (0.242)	0.129 (0.075)	0.117 (0.072)
Ngyuen	0.917 (0.091)	0.537 (0.275)	0.328 (0.281)	0.180 (0.128)	0.117 (0.072)
SDS	0.947 (0.050)	0.548 (0.263)	0.336 (0.276)	0.212 (0.133)	0.132 (0.076)
<i>Interventions</i>					
Direct only	0.838 (0.089)	0.493 (0.265)	0.397 (0.293)	0.216 (0.148)	0.139 (0.086)
DS	0.933 (0.068)	0.507 (0.239)	0.344 (0.295)	0.250 (0.164)	0.172 (0.099)
Ngyuen	0.921 (0.073)	0.536 (0.254)	0.419 (0.300)	0.248 (0.162)	0.158 (0.097)
SDS	0.936 (0.063)	0.530 (0.249)	0.389 (0.323)	0.252 (0.164)	0.172 (0.099)
<i>Outcomes</i>					
Direct only	0.837 (0.096)	0.261 (0.241)	0.180 (0.244)	0.114 (0.117)	0.080 (0.072)
DS	0.896 (0.078)	0.308 (0.223)	0.117 (0.203)	0.148 (0.133)	0.120 (0.091)
Ngyuen	0.870 (0.085)	0.339 (0.256)	0.228 (0.288)	0.151 (0.137)	0.106 (0.084)
SDS	0.900 (0.070)	0.333 (0.233)	0.138 (0.212)	0.160 (0.134)	0.124 (0.092)

Table 4: Retrospective results, with respect to: per-article AUC, NDCG@20, precision@10 and precision@20. For each we report the means and standard deviations over the 133 articles for which candidate sets were annotated for the respective domains. All sentences not in candidate sets are assumed to be *irrelevant*, these results are therefore noisy and likely pessimistic. We **bold** cells corresponding to the best performing methods for each metric, PICO element pair.

EXTRACTING PICO SENTENCES FROM CLINICAL TRIAL REPORTS USING SDS

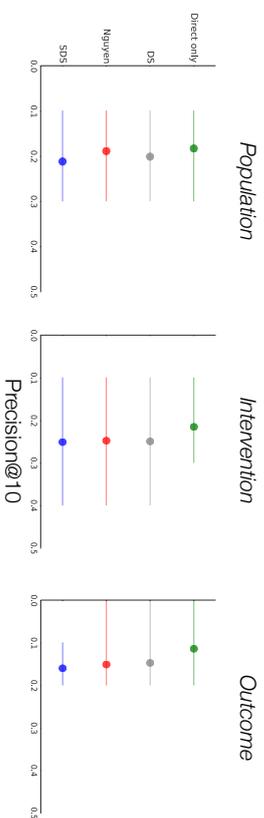


Figure 3: A subset of the retrospective results presented in Table 4 depicted graphically. Top: mean precision@10 for each method and domain (thin lines span the 25th to 75th percentiles, over articles). Bottom: density plots of per-article AUCs. Note that while Ngyuen is the most competitive method with SDS with respect to precision@10, simple DS outperforms this method in terms of overall ranking performance (AUC). SDS maintains a modest but consistent edge over other approaches.

For clarity we also present a subset of the retrospective results graphically in Figure 3. The top row of this figure depicts the mean precision@10 (and 25th/75th percentiles across articles) realized by each model in each domain. The bottom row of this figure describes the distributions of AUCs realized by each strategy for each domain. These are density plots (smoothed histograms) showing the empirical density of AUCs (calculated per-article) achieved by each strategy.

We observe that the top two models with respect to precision@10 (and precision@k in general) are SDS and the interpolation approach proposed by Ngyuen and Moschitti (Ngyuen and Moschitti, 2011). But in terms of overall ranking performance (AUC), vanilla DS outperforms the latter but not the former. Put another way: SDS appears to perform well both in terms of overall ranking and with respect to discriminative performance amongst the top k sentences.

Method	Precision@3
<i>Population</i>	
Nguyen	0.907 (0.222)
SDS	0.927 (0.214)
<i>Interventions</i>	
Nguyen	0.854 (0.254)
SDS	0.903 (0.245)
<i>Outcomes</i>	
Nguyen	0.880 (0.208)
SDS	0.887 (0.196)

Table 5: Averages (and standard deviations) of the proportion of the top-3 sentences extracted via the respective models from 50 prospectively annotated articles that were deemed *relevant* or *best available* by an annotator. The annotator was blinded to which model selected which sentences.

6.2 Prospective results

We prospectively evaluated the top-3 sentences retrieved by the Nguyen and SDS methods (as these were the best performing in our retrospective evaluation). We report precision@3 for each approach in Table 5, calculated over 50 prospectively annotated articles. One can see that here SDS consistently includes *more* relevant sentences among the top-3 than does the pooling approach, and this holds across all domains. The difference is in some cases substantial; e.g., we see a 5% absolute gain in precision@3 for Interventions (a gain of nearly 3% for Population. For Outcomes the difference is less pronounced (nearing 1 point in precision@3).

7. Discussion

We have presented and evaluated a new approach to automating the extraction of sentences describing the PICO elements from the full-texts of biomedical publications that report the conduct and results of clinical trials. As far as we are aware, this is the first effort to build models that automatically extract PICO sentences from full-texts.

We demonstrated the efficacy of using distant supervision (DS) for this task and we introduced *supervised distant supervision* (SDS), a new, flexible approach to distant supervision that capitalizes on a small set of direct annotation to mitigate noise in distantly derived annotations. We demonstrated that this consistently improves performance compared to baseline models that exploit either distant or direct supervision only, and generally also outperforms a previously proposed approach to combining direct and distant supervision. While this work has been motivated by EBM and specifically the task of PICO extraction,

we believe that the proposed SDS approach represents a generally useful strategy for learning jointly from distant and direct supervision.

A natural extension to SDS would be to explore *active* SDS, in which one would aim to selectively acquire the small set of directly annotated instances with which to estimate the parameters of the mapping function f . This may further economize efforts by capitalizing on a small set of examples cleverly selected instances to learn a model that can subsequently ‘clean’ a very large set of distantly generated labels.

For the present application of PICO extraction, we would also like in future work to introduce dependencies across sentences into the model. The model we have proposed ignores such structure. We also note that we hope to extend the present approach by mapping tokens comprising the identified PICO sentences to normalized terms from a structured biomedical vocabulary (namely, MeSH⁷).

With respect to next steps toward automating EBM, we hope to develop models that take as input the PICO sentences extracted from articles to improve ‘downstream’ tasks. For example, we have already incorporated these models into our *RobotReviewer* (Marshall et al., 2015; Kuiper et al., 2014) tool,⁸ which aims to facilitate semi-automated data extraction from full-text articles for biomedical evidence synthesis. This tool uses machine learning models to automatically identify and highlight passages likely to contain the information of interest, thus expediting the extraction process. Additionally, extracted PICO sentences could be used to improve article indexing for search, or fed as input to models for extracting structured bits of information, such as outcome metrics.

Realizing the aim of evidence-based care in an era of information overload necessitates the development of new machine learning and natural language processing technologies to optimize aspects of evidence synthesizes. This work represents one step toward this goal, but much work remains.

8. Acknowledgements

We thank our anonymous JMLR reviewers for thoughtful feedback. This work was made possible by support from the National Library of Medicine (NLM), NIH/NLM grant R01LM012086.

References

IE Allen and I Olkin. Estimating time to conduct a meta-analysis from number of citations retrieved. *JAMA: The Journal of the American Medical Association*, 282(7):634–635, 1999.

7. <http://www.ncbi.nlm.nih.gov/mesh>

8. <https://robot-reviewer.vortext.systems/>

- G Angeli, J Tibshirani, J Wu, and CD Manning. Combining distant and partial supervision for relation extraction. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*, pages 1556–1567. ACL, 2014.
- H Bastian, P Glasziou, and I Chalmers. Seventy-five trials and eleven systematic reviews a day: how will we ever keep up? *PLoS medicine*, 7(9), 2010.
- F Boudin, J-Y Nie, and M Dawes. Positional language models for clinical information retrieval. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 108–115. Association for Computational Linguistics, 2010a.
- F Boudin, L Shi, and J-Y Nie. Improving medical information retrieval with pico element detection. In *Advances in Information Retrieval*, pages 50–61. Springer, 2010b.
- RC Bunescu and R Mooney. Learning to extract relations from the web using minimal supervision. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 576–587. Association for Computational Linguistics, 2007.
- GY Chung. Sentence retrieval for abstracts of randomized controlled trials. *BMC Medical Informatics and Decision Making*, 9(1):10, 2009.
- M Craven and J Kunjlen. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the Annual Meeting of the International Society for Computational Biology (ISCB)*, pages 77–86, 1999.
- D Demner-Fushman and J Lin. Answering clinical questions with knowledge-based and statistical techniques. *Computational Linguistics*, 33(1):63–103, 2007.
- J Elljoh, I Sim, J Thomas, N Owens, G Dooley, J Riis, B Wallace, J Thomas, A Noel-Storr, and G Rada. CochrameTech: technology and the future of systematic reviews. *The Cochrane database of systematic reviews*, 9:ED000091, 2014.
- R-E Fan, K-W Chang, G-J Hsieh, X-R Wang, and G-J Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research (JMLR)*, 9:1871–1874, 2008.
- X Huang, J Lin, and D Demner-Fushman. Evaluation of PICO as a knowledge representation for clinical questions. In *Proceedings of the Annual Meeting of the American Medical Informatics Association (AMIA)*, volume 2006, page 359. AMIA, 2006.
- K Järvelin and J Kekäläinen. In evaluation methods for retrieving highly relevant documents. In *Proceedings of the annual international ACM SIGIR conference on Research and development in information retrieval*, pages 41–48. ACM, 2000.
- K Järvelin and J Kekäläinen. Cumulated gain-based evaluation of ir techniques. *Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.
- SN Kim, D Martinez, L Cavedon, and L Yencken. Automatic classification of sentences to support evidence based medicine. *BMC Bioinformatics*, 12(Suppl 2):S5, 2011.
- S Kritchenko, B de Bruijn, S Carini, J Martin, and I Sim. ExaCT: automatic extraction of clinical trial characteristics from journal publications. *BMC medical informatics and decision making*, 10(1):56, 2010.
- J Kuiper, IJ Marshall, BC Wallace, and MIA Swertz. Spá: A web-based viewer for text mining in evidence based medicine. In *Proceedings of the The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*, pages 452–455. Springer, 2014.
- M Marchetti-Bowick and N Chambers. Learning for microblogs with distant supervision: Political forecasting with twitter. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 603–612. Association for Computational Linguistics, 2012.
- IJ Marshall, J Kuiper, and BC Wallace. Automating risk of bias assessment for clinical trials. In *Proceedings of the ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 88–95. Association for Computing Machinery, 2014.
- IJ Marshall, J Kuiper, and BC Wallace. Robotreviewer: evaluation of a system for automatically assessing bias in clinical trials. In *The Journal of the American Medical Informatics Association (JAMIA)*, 2015. doi: <http://dx.doi.org/10.1093/jamia/ocv04>.
- B Min, R Grishman, L Wan, C Wang, and D Gondak. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 777–782, 2013.
- M Mintz, S Bills, R Snow, and D Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the Association of Computational Linguistics (ACL) and the International Joint Conference on Natural Language Processing (IJCNLP)*, pages 1003–1011. Association for Computational Linguistics, 2009.
- TVT Nguyen and A Moschitti. Joint distant and direct supervision for relation extraction. In *Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP)*, pages 732–740. Asian Federation of Natural Language Processing, 2011.
- F Pedregosa, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, M Blondel, P Prettenhofer, R Weiss, V Dubourg, et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research (JMLR)*, 12:2825–2830, 2011.
- M Purver and S Battersby. Experimenting with distant supervision for emotion classification. In *Proceedings of the Conference of the European Chapter of the Association*

- for *Computational Linguistics (ACL)*, pages 482–491. Association for Computational Linguistics, 2012.
- S Riedel, L Yao, and A McCallum. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer, 2010.
- B Roth, T Barth, M Wiegand, and D Klakow. A survey of noise reduction methods for distant supervision. In *Proceedings of the workshop on Automated knowledge base construction*, pages 73–78. ACM, 2013.
- M Surdeanu, J Tibshirani, R Nallapati, and CD Manning. Multi-instance multi-label learning for relation extraction. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 455–465. Association for Computational Linguistics, 2012.
- S Takamatsu, I Sato, and H Nakagawa. Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 721–729. Association for Computational Linguistics, 2012.
- The Cochrane Collaboration. The Cochrane Database of Systematic Reviews, 2014. URL <http://www.thecochranelibrary.com>.
- G Tsafnat, A Dunn, P Glasziou, and E Coiera. The automation of systematic reviews. *British Medical Journal (BMJ)*, 346, 2013.
- BC Wallace, IJ Dahabreh, CH Schmid, J Lau, and TA Trikalinos. Modernizing the systematic review process to inform comparative effectiveness: tools and methods. *Journal of Comparative Effectiveness Research*, 2(3):273–282, 2013.

Cross-Corpora Unsupervised Learning of Trajectories in Autism Spectrum Disorders

Huseyin Melih Elibol

Vincent Nguyen

Scott Linderman

Matthew Johnson

Anna Hashmi

Finale Doshi-Velez

Paulson School of Engineering and Applied Sciences

Harvard University

Cambridge, MA 02138, USA

ELIBOL@G.HARVARD.EDU

VINCENTNGUYEN@ALUMNI.HARVARD.EDU

SLINDERMAN@SEAS.HARVARD.EDU

MAFTJJ@CSAIL.MIT.EDU

AMINHASHMI@ALUMNI.HARVARD.EDU

FTNALE@SEAS.HARVARD.EDU

Editor: Benjamin M. Marlin, C. David Page, and Suchi Soria

Abstract

Patients with developmental disorders, such as autism spectrum disorder (ASD), present with symptoms that change with time even if the named diagnosis remains fixed. For example, language impairments may present as delayed speech in a toddler and difficulty reading in a school-age child. Characterizing these trajectories is important for early treatment. However, deriving these trajectories from observational sources is challenging: electronic health records only reflect observations of patients at irregular intervals and only record what factors are clinically relevant at the time of observation. Meanwhile, caretakers discuss daily developments and concerns on social media.

In this work, we present a fully unsupervised approach for learning disease trajectories from incomplete medical records and social media posts, including cases in which we have only a single observation of each patient. In particular, we use a dynamic topic model approach which embeds each disease trajectory as a path in \mathbb{R}^D . A Pólya-gamma augmentation scheme is used to efficiently perform inference as well as incorporate multiple data sources. We learn disease trajectories from the electronic health records of 13,435 patients with ASD and the forum posts of 13,743 caretakers of children with ASD, deriving interesting clinical insights as well as good predictions.

Keywords: Disease progression model, Dynamic topic model

1. Introduction

Psychiatric conditions that arise in childhood, generally termed developmental disorders, are increasingly common. The parent-reported rates of developmental disorders are now nearly 15%, which includes learning disabilities (affecting 7.66% of children) and attention deficit hyperactivity disorder (ADHD, 6.69% of children) (Boyle et al., 2011). CDC estimates for the prevalence of autism spectrum disorder (ASD) is now 1 in 68 children which is over 1% of the US population (Baio, 2014).

Characterizing these disorders is challenging because, unlike many adult disorders, the symptoms of developmental disorders are inextricably linked to the developmental processes

of childhood. For example, a language-related impairment may present as delayed speech in a toddler and difficulty reading in a school-age child. A neurological condition may manifest as convulsions at age three and intellectual disability at age seven. Characterizing the evolution of distinct disease courses is a critical step toward personalizing treatments; with developmental disorders the early identification of appropriate therapy can significantly increase a child's IQ and ability to communicate (Peters-Scheffer et al., 2011).

However, constructing these trajectories from data is challenging. Clinical studies tend to have the cleanest sources of data: patients are followed regularly, and measurements are consistently recorded. Unfortunately, most clinical studies involve small cohorts—under 200 individuals—which can make it difficult or impossible to distinguish heterogeneous disease courses from variance. In contrast, electronic health records (EHRs) and social media (SM) provide valuable windows to study populations of thousands of individuals. However, these less-structured sources are much more challenging to analyze due to several factors:

- *(Extremely) Partial trajectories.* EHRs are often confined to a single medical system; if a patient switches providers then their history will no longer be available. Similarly, patients and caregivers may be active on social media at some times and not others.
- *Irregular interactions.* Patients generally only visit clinics or post to social media when they have complaints; we do not observe data from patients between these times.
- *Partially structured, noisy, high-dimensional information.* The space of clinical symptoms is large, and with both clinician and caregiver-generated text, information may also be entered or described incorrectly. Clinicians and patients use very different vocabularies when describing the same symptoms.

To address these challenges, we develop an unsupervised approach that models each source—electronic health records and social media—with a cross-corpora dynamic topic model. Our model can be scientifically interpreted as positing that there are a few underlying disease processes that characterize the signs and symptoms that we observe in our patient population. Each disease is a process that evolves over time; we posit that each disease process k at each time t is associated with a distribution over possible signs and symptoms it may emit. The same disease process may be described differently in electronic health records and social media, and multiple diseases may be simultaneously present in a patient.

Specifically, we assume data in the form of (patient , time , sign) tuples. For some patients, we have may have data at multiple times; for other patients, we may only have data at one time. Similarly, some patients may have many signs, others just a few. Our approach derives distinct disease trajectories *without* linking individual identities between social media and electronic health records, and it can also derive disease trajectories in the limit of only a single note per patient. Thus, we do not have to restrict ourselves to patients with longitudinal data; we are able to incorporate all patient data that we have.

For inference in our model, we explore the use of Pólya-gamma augmentation scheme (Polson et al., 2013; Zhou et al., 2012b; Chen et al., 2013; Linderman et al., 2015) to easily adapt the model to have different correlation structures. We detail our approach in Sections 3 and 4, and review related work in Section 6. In Section 5, we apply our approach to a large data set of electronic health records from 13,435 individuals with ASD and 13,743

forum posts by 2,391 caretakers of children with ASD. To our knowledge, this is the first study to jointly model temporal patterns in electronic health record and social media data at this scale.

2. Background

Our technical approach uses Pólya-gamma augmentation to construct an efficient and easily extensible sampler for dynamic topic models and related models. In this section we briefly review topic models and Pólya-gamma augmentation.

2.1 Topic Models and Dynamic Topic Models

Latent Dirichlet Allocation (LDA) The latent Dirichlet allocation (LDA) topic model (Blei et al., 2003) is one of the most successful and widely used models in machine learning. Its basic aim is to decompose a corpus of natural language documents, like a collection of news articles or scientific papers, into an interpretable collection of topics as well as identify what topics are present in each document. For example, a corpus of scientific papers may contain topics like atomic physics, cosmology, and neural chemistry. For modeling purposes, each such topic is identified with a distribution over words: for example, the word “experiment” might have high probability in all three topics, while only the cosmology topic might have frequent occurrences of words like “star” and “galaxy.” In this simplified view, to identify the topics present in a document, it is not necessary to model the details of language or even the order of the words in each document; instead, a document can be summarized by “bag of words:” a histogram counting the words that it contains.

The LDA topic model of Blei et al. (2003) posits that each document can be characterized by a distribution over the topics it contains, and each topic can be characterized by a distribution over the words associated with it. In symbols, each document d has a distribution over topics θ_d ($d = 1, 2, \dots, D$), and each topic β_k ($k = 1, 2, \dots, K$) is a distribution over a vocabulary of V possible words. Given Dirichlet priors on the topics β and topic proportions θ with parameters α_β and α_θ , the full generative model (also illustrated in figure 1a) is

$$\begin{aligned} \beta_k &\sim \text{Dir}(\alpha_\beta), \\ \theta_d &\sim \text{Dir}(\alpha_\theta), \\ z_{n,d} | \theta_d &\sim \text{Cat}(\theta_d), \\ w_{n,d} | z_{n,d}, \{\beta\} &\sim \text{Cat}(\beta_{z_{n,d}}). \end{aligned} \quad (1)$$

where $w_{n,d}$ is n^{th} word in document d , $z_{n,d}$ is the topic associated with the word $w_{n,d}$, $\text{Cat}(\pi)$ draws one sample from a vector of probabilities π , and Dir is the Dirichlet distribution. The Dirichlet-multinomial conjugacy in the generative process makes it straight-forward to perform inference via a blocked Gibbs sampling scheme that, given a set of words $\{w_{n,d}\}$, can sample the latent topic-word distributions $\{\beta_{k_i}\}$, the document-topic proportions $\{\theta_d\}$, and the word-topic assignments $\{z_{n,d}\}$.

Dynamic Topic Model (DTM) Blei and Lafferty (2006b) expand upon LDA to model temporal evolution in the topics β . Each multinomial topic distribution β_k is modeled

through its natural parameter ψ_k ; the mapping from ψ_k to β_k is a multi-class logistic function given by

$$\beta_k(v) \equiv \beta(\psi_k(v)) \equiv \frac{\exp(\psi_k(v))}{\sum_{v'} \exp(\psi_k(v'))}. \quad (2)$$

where $\beta_k(v)$ is the probability of word v in topic k . The natural parameters ψ_k are unconstrained—they can be positive or negative, and they do not need to sum to one.

Next, Blei and Lafferty (2006b) model the evolution of each topic β_k as a random walk on its natural parameters ψ_k . Let $\psi_{k,t}$ denote the values of the natural parameters ψ for topic k at time t . The DTM posits the following generative process on ψ , also illustrated in figure 1b:

$$\begin{aligned} \psi_{k,t} | \psi_{k,t-1} &\sim \mathcal{N}(\psi_{k,t-1}, \sigma^2 I), \\ \theta_d &\sim \text{Dir}(\alpha_\theta), \\ z_{n,d} | \theta_d &\sim \text{Cat}(\theta_d), \\ w_{n,d} | z_{n,d}, \{\psi_{k,t}\} &\sim \text{Cat}(\beta(\psi_{z_{n,d},t}(d))). \end{aligned} \quad (3)$$

Here, $t(d)$ is the time associated with document d and $\beta(\psi_{k,t})$ is the transformation of $\psi_{k,t}$ back to a multinomial using equation 2. We will use $\beta_{k,t}$ as shorthand for $\beta(\psi_{k,t})$.

This DTM construction captures the temporal evolution of topics while retaining the interpretable structure of LDA. However, the DTM construction in equation 3 does not enjoy the conjugacy structure of the original LDA model in equation 1: the DTM replaces LDA’s factored Dirichlet prior on the topics β_k with a Gaussian linear dynamical system (LDS) mapped through a multi-class logistic function. While inference in Gaussian linear dynamical systems coupled with linear Gaussian observations can be performed efficiently using message passing algorithms, the nonlinear mapping in equation 2 does not allow such algorithms to be applied directly.

2.2 Pólya-gamma Augmentation

Pólya-gamma augmentation is an auxiliary variable scheme that allows multinomial observations to appear as Gaussian likelihoods. This scheme has recently been used to develop Gibbs samplers and variational inference algorithms for Bernoulli, binomial, negative binomial, and multinomial regression models with logit link functions (Polson et al., 2013). Chen et al. (2013) use Pólya-gamma augmentation for multinomial models in the context of LDA, but in a way that only provides limited single-site inference updates. More recently, Linderman et al. (2015) extend the Pólya-gamma augmentation scheme for multinomial models in such a way that allows block updates and hence readily extends to dynamic topic models, in which entire state trajectories must be updated as a block for inference to be efficient. Here, we use the augmentation strategy of Linderman et al. (2015) to enable such block updating in our dynamic topic models.

The Pólya-gamma augmentation scheme is based on an integral identity derived from the Laplace transform of the Pólya-gamma distribution. Specifically, if $p(\omega | b, 0)$ is the density of the Pólya-gamma distribution $\text{PG}(b, 0)$, then

$$\frac{(e^\omega)^a}{(1 + e^\omega)^b} = 2^{-b} e^{a\omega} \int_0^\infty e^{-\omega\omega'/2} p(\omega' | b, 0) d\omega', \quad (4)$$

where $\kappa = a - b/2$. The integral on the right-hand side is the Laplace transform of the Pólya-gamma density evaluated at $\psi^2/2$, and the left-hand side is a functional form that often appears in logistic likelihoods. Importantly, viewed as a function of ψ for fixed ω , the right-hand side is an unnormalized Gaussian density. Thus, the identity in equation 4 transforms a logistic likelihood to a Gaussian likelihood conditioned on an auxiliary variable, ω .

While we focus on Gibbs sampling inference here, the Pólya-gamma augmentation scheme also enables efficient mean-field variational inference (Linderman et al., 2015; Zhou et al., 2012b), including scalable stochastic variational inference (Hoffman et al., 2013; Linderman et al., 2015). These algorithms could be adapted to provide scalable inference for the dynamic topic model case that we study here.

Binomial Case For the binomial case, Polson et al. (2013) let $\psi_0 = 0$ and write $\psi_1 = \psi$. Let $x = (x_0, x_1)$ be the number of zeros and ones that have been observed. Then we can write the likelihood of the natural parameter ψ given the data x as

$$p(x | \psi) = \binom{x_0 + x_1}{x_1} \frac{(e^\psi)^{x_1}}{(1 + e^\psi)^{x_0}} = c(x) \frac{(e^\psi)^{a(x)}}{(1 + e^\psi)^{b(x)}}$$

Given a prior $p(\psi)$ on the natural parameter ψ , then the joint density of (ψ, x) can be written as

$$p(\psi, x) = p(\psi) c(x) \frac{(e^\psi)^{a(x)}}{(1 + e^\psi)^{b(x)}} = \int_0^\infty p(\psi) c(x) 2^{-b(x)} e^{\kappa(x)\psi} e^{-\omega\psi^2/2} p(\omega | b(x), 0) d\omega. \quad (5)$$

The integrand of (5) defines a joint density on (ψ, x, ω) . If we condition on the auxiliary variables ω , then the conditional density $p(\psi | x, \omega)$ on the natural Bernoulli parameter ψ is given by

$$p(\psi | x, \omega) \propto p(\psi) e^{\kappa(x)\psi} e^{-\omega\psi^2/2} \quad (6)$$

which is Gaussian if $p(\psi)$ is Gaussian. By the exponential tilting property of the Pólya-gamma distribution, we have $\omega | \psi, x \sim \text{PG}(b(x), \psi)$. Efficient samplers exist for Pólya-gamma distributed variables (Windle et al., 2014), and thus we can alternate between sampling $\omega | \psi, x$ from a Pólya-gamma distribution and sampling $\psi | \omega, x$ from a Gaussian distribution.

Multinomial Case For the multinomial case, Linderman et al. (2015) rewrite the K -dimensional multinomial likelihood recursively in terms of $K - 1$ binomial densities using the following stick-breaking representation. Let β be a vector describing the probability of each outcome $1 \dots K$. Then we can define $\tilde{\beta}_k$ to be probability of choosing option k given that we have not selected any option $j < k$:

$$\tilde{\beta}_k = \frac{\beta_k}{1 - \sum_{j < k} \beta_j} \quad (7)$$

Writing the probabilities β in this way allows us to write the multinomial density as a product of binomial densities:

$$\text{Mult}(\mathbf{x} | N, \beta) = \prod_{k=1}^{K-1} \text{Bin}(x_k | N_k, \tilde{\beta}_k), \quad (8)$$

$$N_k = N - \sum_{j < k} x_j, \quad k = 1, 2, \dots, K, \quad (9)$$

where we can interpret N_k as the number of observations remaining after the observations where $j = 1, 2, \dots, k$ have been removed. Substituting $\tilde{\beta}_k = \sigma(\psi_k)$, we can write the multinomial likelihood as

$$\begin{aligned} \text{Mult}(\mathbf{x} | N, \psi) &= \prod_{k=1}^{K-1} \text{Bin}(x_k | N_k, \sigma(\psi_k)) = \prod_{k=1}^{K-1} \binom{N_k}{x_k} \sigma(\psi_k)^{x_k} (1 - \sigma(\psi_k))^{N_k - x_k} \\ &= \prod_{k=1}^{K-1} \binom{N_k}{x_k} \frac{(e^{\psi_k})^{x_k}}{(1 + e^{\psi_k})^{N_k}}. \end{aligned}$$

Linderman et al. (2015) next let $\mathbf{a}_k(\mathbf{x}) = \mathbf{x}_k$ and $\mathbf{b}_k(\mathbf{x}) = N_k$ for each $k = 1, 2, \dots, K - 1$ and introduce Pólya-gamma auxiliary variables ω_k corresponding to each coordinate ψ_k . Then the probability of the data \mathbf{x} and the auxiliary variables ω given the natural parameters ψ has a diagonal Gaussian likelihood:

$$p(\mathbf{x}, \omega | \psi) \propto \prod_{k=1}^{K-1} e^{(\kappa_k - N_k/2)\psi_k - \omega_k \psi_k^2/2} \propto \mathcal{N}\left(\psi \mid \Omega^{-1} \kappa(\mathbf{x}), \Omega^{-1}\right),$$

where $\Omega \equiv \text{diag}(\omega)$ and $\kappa(\mathbf{x}) \equiv \mathbf{x} - N(\mathbf{x})/2$. Thus, if we begin with a Gaussian prior $p(\psi)$ on the stick-breaking parameters ψ , then the posterior will remain Gaussian.

Finally, given the parameters ψ , we can recover the parameters β through the stick-breaking construction:

$$\begin{aligned} \tilde{\beta}_j &= \sigma(\psi_j) \\ \beta_k &= \tilde{\beta}_k \prod_{j < k} (1 - \tilde{\beta}_j) \end{aligned} \quad (10)$$

We denote this recovery process in equation 10 by the function $\beta \equiv \pi_{\text{SB}}(\psi)$.

3. Model: Stick-breaking Construction for Dynamic Topic Models

The Pólya-gamma augmentation scheme allows us to take a Gaussian graphical model in which efficient inference is well-developed and apply it to models with multinomial likelihoods. However, we must first convert the dynamic topic model from Section 2.1 into the appropriate stick-breaking form. In this section we describe this stick-breaking construction and a natural cross-corpora extension; for completeness we also include the parts of the dynamic topic model that remain unchanged.

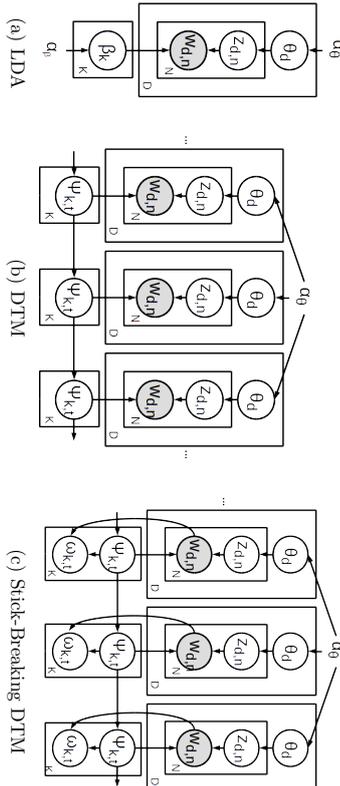


Figure 1: Graphical Models of latent Dirichlet allocation, the dynamic topic model, and our stick-breaking dynamic topic model. The natural parameters ψ are converted to multinomials β through the stick-breaking process in equation 10

3.1 Document-Specific parameters $\{\theta_d\}$ and $\{z_{n,d}\}$

As in the standard LDA approach, we continue to model the proportion of each topic in each document θ_d as being drawn independently from Dirichlet distributions with parameters α_θ , and the topic $z_{n,d}$ for each word $w_{n,d}$ drawn from θ_d :

$$\begin{aligned} \theta_d &\sim \text{Dir}(\alpha_\theta), \\ z_{n,d} | \theta_d &\sim \text{Cat}(\theta_d). \end{aligned}$$

3.2 Topic Parameters $\{\beta_k\}$

Static Stick-Breaking LDA Model In standard LDA, the likelihood associated with each topic β_k depends on the words assigned to that topic:

$$p(\{w_d\}_{d=1}^D | \{z_d\}_{d=1}^D, \{\beta_k\}_{k=1}^K) \propto \prod_{d=1}^D \prod_{k=1}^{N_d} \beta_k^{\mathbb{1}\{z_{d,n}=k\}} \propto \text{Mult} \left(\sum_{d=1}^D \mathbf{b}_{d,k}, \sum_{d=1}^D N_{d,k}, \beta_k \right)$$

where $\{w_{n,d}\}$ are all of the words in document d and $\{z_{n,d}\}$ are all of their assignments. Let N_d be the number of words in document d . The count vectors $\mathbf{b}_{d,k,v}$ and $N_{d,k}$ count the number of occurrences of word v in document d assigned to topic k and the number of occurrences of the topic k in document d , respectively:

$$\begin{aligned} \mathbf{b}_{d,k,v} &= \sum_{n=1}^{N_d} \mathbb{1}\{w_{d,n} = v\} \mathbb{1}\{z_{d,n} = k\}, \\ N_{d,k} &= \sum_{n=1}^{N_d} \mathbb{1}\{z_{d,n} = k\}. \end{aligned} \tag{11}$$

We transform the word probability vectors such that $\beta_k \equiv \pi_{\text{SB}}(\psi_k)$, introduce auxiliary variables ω_k , and set a Gaussian prior $\psi_k \sim \mathcal{N}(\mu, \Sigma)$ on the stick-breaking parameters ψ . Then the posterior over ψ given the counts $\{\mathbf{b}_d\}$ is given by the Gaussian

$$p(\psi_k | \{\mathbf{b}_d\}, \{z_d\}, \omega_k, \mu, \Sigma) \propto \mathcal{N} \left(\psi_k | \Omega_k^{-1} \cdot \kappa \left(\sum_{d=1}^D \mathbf{b}_{d,k} \right), \Omega_k^{-1} \right) \mathcal{N}(\psi_k | \mu, \Sigma) \tag{12}$$

Dynamic Stick-Breaking Topic Model Let $t(d) \in \mathbb{N}$ denote the discrete time index of document d and $\beta_{t,k} \in [0, 1]^V$ denote the word probability vector of topic k at time t . Then we can define the following dynamical system model

$$\begin{aligned} \psi_{t,k} &\sim \mathcal{N}(\mathbf{A}\psi_{t-1,k}, \mathbf{B}\mathbf{B}^\top) \\ \beta_{t,k} &\equiv \pi_{\text{SB}}(\psi_{t,k}) \end{aligned} \tag{13}$$

where $\mathbf{u}_{t,k}$ is a latent state of topic k at time t . Then the likelihood associated with latent state vectors $\{\mathbf{u}_{t,k}\}$ given the word-topic assignments $\{z_{n,d}\}$ is given by the diagonal Gaussian potential

$$p(\mathbf{b}_{d,k} | \{\mathbf{u}_{t(d),k}, \omega_{t(d),k}\}) \propto \mathcal{N} \left(\mathbf{\Omega}^{-1} \cdot \kappa \left(\sum_{t:t(d)=t} \mathbf{b}_{d,k} \right) | \psi_{t(d),k}, \mathbf{\Omega}^{-1} \right). \tag{14}$$

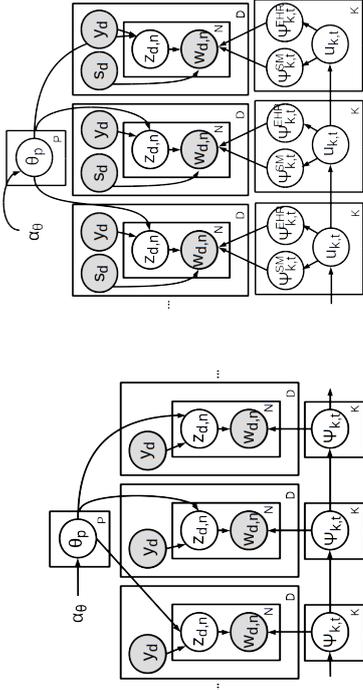
where $\mathbf{\Omega} \equiv \text{diag}(\omega_{t,k})$. As in equation 11, $\mathbf{b}_{d,k}$ counts how often each word v is assigned to topic k in document d , and the likelihood for $\psi_{t,k}$ only depends on the documents for which $t(d) = t$. Figure 1c shows the graphical model of the stick-breaking DTM with the associated Pólya-gamma variables.

3.3 Extensions

Shared Topic Proportions Among Documents In the dynamic topic model, temporal coherence arises due to the smoothness prior on β . While this approach allows us to build temporal models from cross-sectional data, it does not use longitudinal information about whether documents are associated with the same patient when it is available.

One extension we consider is that the proportion of each disease in a patient does not change over time, that is, instead of considering a distinct document-topic vector θ_d for each document, we have a single patient-topic vector θ_p for each patient. However, the probability of a word given the topic— β —will still change with time. This extension is shown in figure 2a, where we introduce the variable y_d to indicate which patient p is associated with each document d ; that is, the indicator y_d selects which θ_p to apply to document d .

Relationships between Multiple Corpora Given multiple corpora, one simple extension of the model from Section 3.2 is to posit that each disease has some canonical temporal process, but the probabilities of the terms associated with that process may vary across different corpora. For example, posts from social media may talk more about the behaviors associated with a disease, while diagnoses may focus on comorbidities. To model differences



(a) DTM with per-patient θ_p ; the observed variable y_d indicates which patient the document came from.
 (b) Cross-corpora extension; the observed variable s_d indicates to which corpus the document belongs.

Figure 2: Graphical Models for the DTMs in which topic proportions are shared across all notes from the same patient (2a) and DTMs that combine multiple corpora (2b). To reduce clutter, we do not include the associated Pólya-gamma variables; these are the same as in figure 1c

in term usage between corpora, we consider a dynamical system structured as

$$\begin{aligned} \mathbf{u}_{t,k} &\sim \mathcal{N}(\mathbf{u}_{t,k} | \mathbf{A}\mathbf{u}_{t-1,k}, \mathbf{B}\mathbf{B}^\top) \\ \epsilon_{t,k,l} &\sim \mathcal{N}(0, \sigma_l^2) \\ \psi_{t,k,l} &\equiv \mathbf{u}_{t,k} + \epsilon_{t,k,l} \\ \beta_{t,k,l} &\equiv \pi\text{SB}(\psi_{t,k,l}) \end{aligned} \quad (15)$$

where now topic proportions $\beta_{t,k,l}$ and their natural parameters $\psi_{t,k,l}$ are associated with a specific corpus l .

Our stick-breaking construction using Pólya-gamma augmentation again renders the relevant likelihoods Gaussian: for each corpus l , the probability of the words associated with the corpus given $\psi_{t,k,l}$ is given by

$$p(\mathbf{b}_{d,k} | \psi_{t(d),k,l(d)}, \boldsymbol{\omega}_{t(d),k,l(d)}) \propto \mathcal{N}\left(\boldsymbol{\Omega}_{t(d),k,l(d)}^{-1} \cdot \boldsymbol{\kappa} \left(\sum_{d:l(d)=l} \mathbf{b}_{d,k} \right) \middle| \psi_{t(d),k,l(d)}, \boldsymbol{\Omega}_{t(d),k,l(d)}^{-1}\right)$$

where $l(d)$ is the corpus associated with document d , $\mathbf{b}_{d,k}$ is again a vector of the number of times each word v is assigned to topic k in document d from equation 11, and $\boldsymbol{\Omega} \equiv \text{diag}(\boldsymbol{\omega}_{t,k})$.

Finally, the likelihood associated with the underlying temporal process $\mathbf{u}_{t,k}$ is simply

$$p(\psi_{t,k}, |\mathbf{u}_{t,k}, \sigma_l^2) = \prod_l \mathcal{N}(\psi_{t,k,l} | \mathbf{u}_{t,k}, \sigma_l^2).$$

The cross-corpora extension of the dynamic topic model is shown in figure 2b, where we explicitly show the parameters $\psi_{t,k}^{SM}$ and $\psi_{t,k}^{EHR}$ for just two corpora. The variable s_d indicates which source— $\psi_{t,k}^{SM}$ or $\psi_{t,k}^{EHR}$ —should be used to model document d .

4. Inference

Given the stick-breaking dynamic topic model construction in Section 3.2, inference is straight-forward; the simplicity of inference is a key advantage of the Pólya-gamma augmentation approach. Below we summarize the inference process for the latent variables in our model: the topic proportions θ_d , the topic assignments $\{z_{nd}\}$, the topic parameters \mathbf{u} (which can be deterministically converted into the topic proportions $\boldsymbol{\beta} = \pi\text{SB}(\mathbf{u})$), and the augmentation variables $\boldsymbol{\omega}$. The variables θ , $\{z_{nd}\}$, and $\boldsymbol{\omega}$ are resampled using Gibbs sampling, and \mathbf{u} is resampled using a Gaussian linear dynamical system.

4.1 Resampling Document-Specific Parameters $\{z_{n,d}\}$ and $\{\theta_d\}$

The word-topic assignments $\{z_{n,d}\}$ are resampled exactly as in the Gibbs sampler for LDA:

$$z_{nd} \sim \text{Mult}(\{\beta_{k,v(w_{nd})} \theta_{d,k}\})$$

where $v(w_{n,d})$ is the word associated with the token $w_{n,d}$. Likewise, the topic proportions θ_d are also sampled exactly as in LDA:

$$\theta_d \sim \text{Dir}(\alpha_\theta + \mathbf{N}_d),$$

where \mathbf{N}_d is the vector of counts with $N_{dk} = \sum_{z_{nd} \in d} \mathbb{I}(z_{nd} = k)$. If we are sampling topic proportions per patient rather than per document, then we simply replace N_{dk} with $N_{pk} = \sum_{z_{nd} \in p} \mathbb{I}(z_{nd} = k)$, the number of times that a topic has been observed with each patient.

4.2 Resampling Topic Parameters

In the static LDA case, we can resample the natural parameters ψ from the Gaussian distribution given equation 12. In the dynamic case, we must incorporate the linear dynamical system prior.

Resampling ψ : Dynamic Topic Model The formulas in equation 13 describe a linear Gaussian system, and the likelihoods in equation 14 are also Gaussian, and thus inference on \mathbf{u} can be performed using off-the-shelf algorithms for linear dynamical systems. For completeness, we write the forward-filtering backward-sampling equations here, setting \mathbf{A} from equation 13 to be the identity \mathbf{I} and $\mathbf{B} = \text{diag}(\sigma_n \dots \sigma_n)$. Define the covariance of the random walk $\boldsymbol{\Sigma} \equiv \mathbf{B}\mathbf{B}^\top = \text{diag}(\sigma_n^2 \dots \sigma_n^2)$. For each topic k , we first compute the mean $q_{t,k}$ and variance $Q_{t,k}$ of the ψ_k in the forward pass:

$$\begin{aligned} q_{t,k} &= q_{t-1,k} + (Q_{t-1,k} + \boldsymbol{\Sigma})(Q_{t-1,k} + \boldsymbol{\Sigma} + \boldsymbol{\Omega}_{t(d),k}^{-1})^{-1}(y_{t,k} - q_{t-1,k}) \\ Q_{t,k} &= (\mathbf{I} - (Q_{t-1,k} + \boldsymbol{\Sigma})(Q_{t-1,k} + \boldsymbol{\Sigma} + \boldsymbol{\Omega}_{t(d),k}^{-1})^{-1})(Q_{t-1,k} + \boldsymbol{\Sigma}) \end{aligned} \quad (16)$$

where we start with some q_1 and Q_1 as the prior mean and variance of $\psi_{t=1,k}$, $\mathbf{\Omega}_{t(0),k}^{-1}$ is computed from the auxiliary variables according to equation 14, and we use $q_{t,k} \equiv \mathbf{\Omega}_{t(0),k}^{-1} \cdot \kappa(\sum_{d:t(d)=t} \mathbf{b}_{d,k})$. Importantly, if the initial covariance Q_1 is diagonal, then because the transition covariance $\mathbf{\Sigma}$ and the likelihood covariance $\mathbf{\Omega}$ are also diagonal, the covariance $Q_{t,k}$ remains diagonal for all times t . Thus the updates in equation 16 can be computed in time linear in the size of the vocabulary $|V|$.

Similarly, the backward sampling pass can be efficiently computed by sampling $\psi_{T,k} \sim \mathcal{N}(q_{T,k}, Q_{T,k})$ and then recursively sampling $\psi_{t,k} \sim \mathcal{N}(q'_{t,k}, Q'_{t,k})$ where the mean $q'_{t,k}$ and variance $Q'_{t,k}$ are given by

$$\begin{aligned} q'_{t,k} &= q_{t,k} + Q_{t,k}(Q_{t,k} + \mathbf{\Sigma})^{-1}(\psi_{t+1,k} - q_{t,k}) \\ Q'_{t,k} &= (\mathbf{I} - Q_{t,k}(Q_{t,k} + \mathbf{\Sigma})^{-1})Q_{t,k} \end{aligned} \quad (17)$$

Resampling ω , ψ : Cross-Corpora Dynamic Topic Model In the cross-corpora dynamic topic model from section 3.3, we have separate variables $\omega_{t,k}$ describing the underlying dynamical system and natural parameters $\psi_{t,k,l}$ for each corpus. Conditioned on $\omega_{t,k}$, the distribution over the parameters for $\psi_{t,k,l}$ for each time t are independent. They can be computed using equation 12 for the static LDA case and substituting the appropriate mean and variance:

$$p(\psi_{t,k,l} | \{z_d\}, \omega_{t,k}, \mathbf{\Sigma}) \propto \mathcal{N} \left(\psi_k | \mathbf{\Omega}_{t,k,l}^{-1} \cdot \kappa \left(\sum_{d \in I} \mathbf{b}_d \right), \mathbf{\Omega}_k^{-1} \right) \cdot \mathcal{N}(\psi_{t,k,l} | \omega_{t,k}, \mathbf{\Sigma})$$

where $\mathbf{\Sigma}$ is the diagonal covariance $\text{diag}(\sigma_1^2 \dots \sigma_l^2)$ from equation 15 and \mathbf{b}_d sums over the word counts for topic k at time t in corpus l in document d .

Conditioned on the topic proportions $\psi_{t,k,l}$, the evolving terms $\omega_{t,k}$ can be resampled using a linear dynamical system with $\psi_{t,k,l}$ as the emissions.

Resampling ω In both the cross-corpora and the standard dynamic topic models, we achieve Gaussian likelihoods by augmenting the model with Pólya-gamma distributed variables $\omega_{t,k}$ or $\omega_{t,k,l}$ respectively. The posterior distributions of these variables are given by

$$\omega_{t,k} | \omega_{t,k}, \sim \text{PG}(\mathbf{N}_{t,k}, \omega_{t,k})$$

where $\mathbf{N}_{t,k}$ is a vector of how often each word appeared in all documents at time t that were assigned to topic k . In the cross-corpora case, this becomes

$$\omega_{t,k,l} | \psi_{t,k,l}, \sim \text{PG}(\mathbf{N}_{t,k,l}, \psi_{t,k,l}).$$

5. Application to Learning Trajectories in Autism Spectrum Disorders

5.1 Data Description

Electronic Health Records We analyze the ICD-9CM diagnostic codes from 13,435 patients with at least one ICD-9CM code for autism spectrum disorder (299.0, 299.8, 299.9) from the Boston Children’s Hospital. The Institutional Review Boards of Boston Children’s

Hospital, Harvard Medical School, and the Harvard Paulson School of Engineering and Applied Sciences reviewed this study and approved it as not-human subjects research.

Each ICD-9CM code was converted into a concept unique identifier (CUIs) using the UMLS (Bodenreider, 2004) and filtered for the semantic type “Disease or Syndrome.” For each code, we computed the age of the patient given the patient’s birth date and the date associated with the visit that produced the code. As current evidence (e.g. Stoner et al. (2014)) suggests that ASD develops from birth, we used the age of the child as the time index for the ICD-9CM code.

To form documents, we grouped all codes associated with a patient for each year of age between the ages 0 and 15 into a “document.” For example, if a patient had three visits that generated a total of ten ICD9-CM codes between ages one and two, and two more visits that generated a total of five ICD9-CM codes between ages two and three, then that patient would be associated with two documents: one at time index “age 1,” with ten codes, and one at time index “age 2,” with five codes. Grouping all diagnostic codes from a year into one document smoothed over variations due to visits to specialties that focused on different aspects of the child’s care. This processing procedure resulted in 63,941 documents with an average of 5.3 CUIs each and 7,037 unique CUIs.

Social Media We scraped all subforums of the websites www.asd-forum.org.uk, www.autismweb.com, and www.asdfriendly.org, resulting in 664,954 posts from 80,927 threads. An example post is given in Appendix A.1. The forum posts contained the date of posting but not the child’s date of birth; thus additional processing was required to determine the age of the child—and thus the time-index—for the documents. Regular expressions (see Appendix A.2) were used to extract ages from the posts, and posts with multiple ages were excluded. This procedure resulted in 13,743 posts with a single mention of age. Approximately 1,000 of these posts were hand-checked for accuracy; the regular expressions were adjusted to avoid any errors that were discovered in the hand-checked posts.

We filtered for patients between 0 and 15 years of age, and as with the electronic health records, we combined all the posts written about the same patient with the same age into one document to smooth over variations due to the caregiver’s particular concerns at the time. This processing resulted in a data set of 5,461 documents (each containing possibly multiple posts written in the same year) by 2,391 unique users.

Clinically-relevant terms were extracted from these posts by finding terms that matched the consumer health vocabulary (Zeng, 2015), which has mappings into the UMLS CUIs. A trie was used to quickly match terms to the dictionary of words, and only terms with the semantic type “Disease or Syndrome” were included. The average number of CUIs per document was 1.8. Of the 7,372 CUIs across the EHR and SM data sets, 284 were unique to the forum posts and 2,407 were unique to the EHR codes.

5.2 Methods

Models We considered three variants of dynamic topic models:

- *SB-DTM- θ* The stick-breaking DTM from Section 3.2.

- **SB-DTM- θ_p** The stick-breaking DTM in which we assume that distribution over diseases in each patient remains constant over time, as described in Section 3.3.
- **SB-ccDTM** The stick-breaking DTM in which the EHR and SM corpora are modeled as having distinct topics with shared underlying dynamics, as described in Section 3.3.

These variants were compared to two versions of LDA: in the first version, LDA-K was trained with K topics that did not evolve over time. LDA-K15 was trained with $15K$ topics, accounting for the fact that the dynamic topic model could have a different topic for each year in ages 0 to 15.¹

Evaluations Our first evaluation metric was simply predictive log-likelihoods. We randomly held-out 10% percent of the words from 10% percent of the documents. Once the model was trained, we had a value of the topic proportions θ_d for every document d . Thus, probability of a held-out word w_{nd} was given by

$$p(w_{nd} | \theta_d, \beta) = \sum_z p(w_{nd} | z, \beta_{z,t(d)}) p(z | \theta_d)$$

Our second evaluation metric simulated the more clinically relevant task of stratifying patient risk for various future outcomes. For this evaluation, we considered only patients with at least one document during early childhood—under the age of five—and one document from later childhood—over the age of seven. For 10% of these patients, we held out *all* the documents for after the child was six years old. The documents from when the child was five years old or younger were included in the DTM training. Following training, we computed the average document-topic proportions θ_p for each patient as

$$\theta_p = \frac{1}{N_p^{<5}} \sum_{ds,t,t(d) \leq 5} \theta_d$$

where $N_p^{<5}$ is the number of documents associated with patient p where $t(d) \leq 5$. This averaging corresponds to the assumption that the patient’s disease proportions do not change over time; note that in the shared-proportions DTM from Section 3.3, we can simply use the learned θ_p .

Given a patient-topic vector θ_p , we can compute the likelihood of the future, *unseen* notes

$$p(w_{nd} | \theta_p, \beta) = \sum_{z, ds,t,t(d) \geq 7} p(w_{nd} | z, \beta_{z,t(d)}) p(z | \theta_p)$$

If our temporal models were capturing time-varying patterns in disease processes, we would expect our model to better predict the content of future documents than a static model.

1. We also ran tests using the C implementation of dynamic topic models available at <https://github.com/balei-1ab/dtm> but were unable to achieve satisfying likelihoods with several parameter settings.

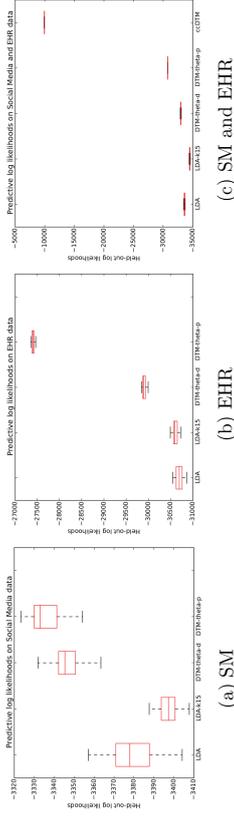


Figure 3: Boxplots of held-out test likelihoods for the different models on SM data alone, EHR data alone, and both data sets combined. Across all versions, the dynamic models have higher predictive performance.

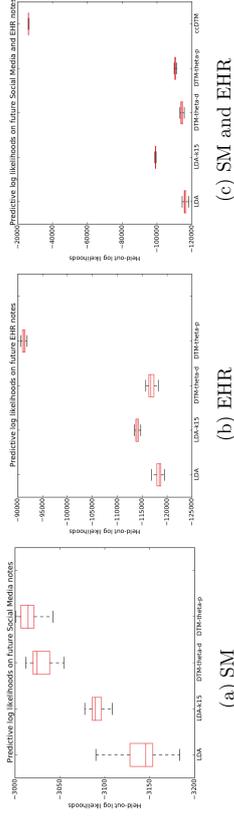


Figure 4: Boxplots of predictions of future patient notes on SM data alone, EHR data alone, and both data sets combined. Models which are trained with the assumption that topic proportions θ_p for each patient remain constant over time do best in the individual data sets, and the transfer learning in the combined case has the best predictive performance.

5.3 Results and Analysis

We completed 10 runs each of LDA, LDA-K15, the standard DTM, the SB-DTM- θ_d , the SB-DTM- θ_p , and the SB-ccDTM. We completed runs on the EHR data alone, the SM data alone, and the SM and EHR data combined. The results of LDA were used to initialize the dynamic topic models, and the results of basic DTM were used to initialize the ccDTM. Preliminary tests of 300 iterations showed that the samplers mixed by around 50 iterations (see figure 9 in appendix B for an example plot); in the results below each sampler was run for 100 iterations. The transition noise parameter in the linear dynamical system was set to $\sigma_n^2 = 0.1$, the cross-corpora noise parameter in SB-ccDTM was set to $\sigma_c^2 = 1$, and the number of topics K was set to 10 based on initial parameter exploration of $K = 5, 10, 15$.

Predictive Performance: Held-out Data Figure 3 shows the held-out test likelihoods for the SM, EHR, and combined cohorts, respectively, for $K = 15$. We see that the dynamic models outperform the static models, including an LDA model with as many topics as the DTM. Indeed, LDA-K15 has the lowest overall predictive likelihoods, suggesting that it may be overfitting. Incorporating links between notes from the same patient (DTM- θ_p)

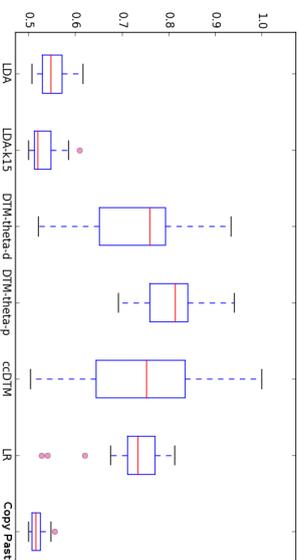


Figure 5: Boxplots of AUCs for predicting future patient conditions for conditions that occurred in at least 10% of the patients. Even without explicitly trying to optimize future predictions, the DTM-based approaches are comparable to—or better than—a discriminative baseline such as logistic regression.

improves prediction quality in both the individual and combined data sets, and the added flexibility of the cross collection ccDTM model further improves prediction accuracy in the combined data set.

Predicting Future notes Figure 4 shows the held-out test likelihoods for the content of *all* patients notes associated with age seven and above given all the notes from that patient under the age of five. Predicting the content of an *entirely* held-out note is much harder than predicting the missing contents of a partially held-out note. We see that training the models with the assumption that topic proportions stay constant—as in the DTM- θ_p model—results in the best predictive performance on these entirely held-out notes in both data sets. In the combined data set, the ccDTM model, which also allows for transfer learning between the SM and EHR data sets, achieves the highest predictive likelihoods.

Figure 5 shows AUCs for the same task of predicting the contents of future notes given current ones. We see that the DTM-based models again perform better than their static counterparts because they are able to imagine what future diseases may occur (the boxplots are over all CTUs with at least 10% prevalence the future notes). The DTM model which takes advantage of the links between patients performs the best, better than the logistic regression discriminative baseline. Finally, we see that simply assuming that a patient’s past condition will continue into the future (copy past) does not produce high AUCs; both the DTM and the logistic regression are learning meaningful predictive relationships.

Transfer Learning between EHR and Social Media The previous analyses showed that our dynamic models better predict held-out and future patient data than a static model. It is also interesting to test whether combining the two data sets increases predictive performance on data from each of the individual data sets, that is, for the same held-out

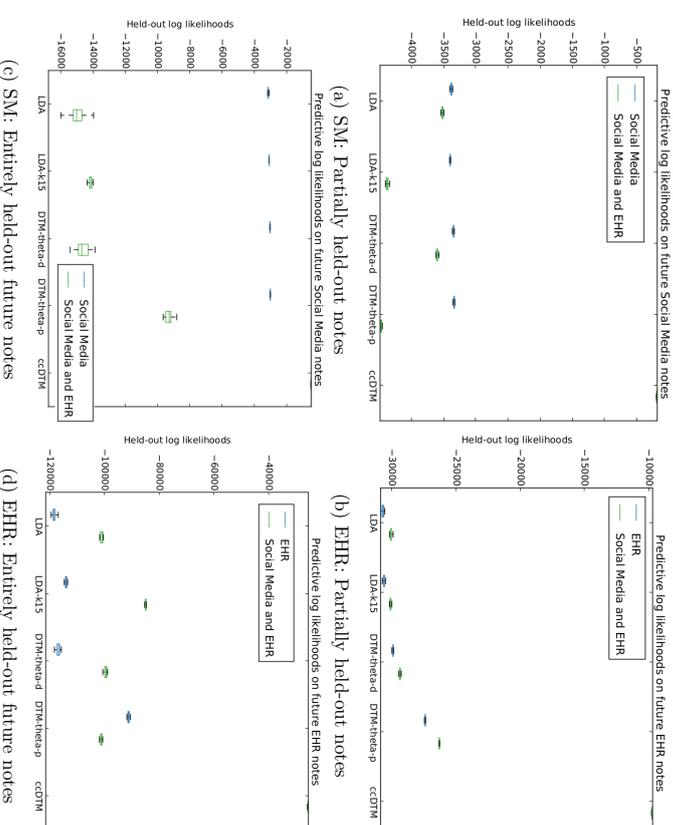


Figure 6: Boxplots comparing of predictions of randomly held-out data and future notes for each cohort vs. the combined cohort. In general, transfer is positive for EHRs and negative for SM; however, the flexibility of the cross-corpora DTM results in positive transfer in all scenarios (tiny bar at the top-right in all the plots).

EHR data, is there a benefit to training on EHR and SM data rather than training on EHR data alone? Likewise, for some held-out SM data, does adding EHR data into the training set benefit predictive performance? (Note that there is no reason, a priori, to assume that combining collections will be beneficial.)

The boxplots in figure 6 show the results of this test for both randomly held-out data and for predicting entire future notes. The blue boxplots correspond to training only on the target data set, and the green boxplots correspond to training on the combined data set. In the EHR cohort, the transfer is positive in almost all cases (the green boxplots are higher than their corresponding blue boxplots), even among models such as standard LDA. The opposite is true in the SM cohort: training on the combined set decreases predictive accuracy among the flat models, likely because the SM data set had many fewer documents than the EHR data set. However, in all cases, we observe that cross-collection DTM (ccDTM)—

whose hierarchy allows greater flexibility in how information is shared across the two data sources—has the highest predictive performance.

Computational Time The static LDA models had the fastest wall-clock times, with a five-topic LDA model on the full corpus taking 0.279 seconds per iteration and the larger LDA-15 taking 0.414 seconds per iteration. The standard DTM- θ_d took 2.00 seconds per iteration, and adding the patient links in the DTM- θ_p increased the per iteration runtime to 2.14 seconds per iteration. Interestingly, the cDTM required only 0.953 seconds per iteration, because the forward-backward pass over the \mathbf{u} variables only had two emissions—the ψ from each corpus—rather than inputs from all of the documents.

Qualitative Examination of Topics: Electronic Health Records We show the top-4 words for the EHR-only θ -p DTM in table 1. (We choose a small K for brevity, larger K have similar and additional patterns.) Topic 0 corresponds to the trajectory of patients with ASD who also have Down’s syndrome. ASD and Down’s syndrome are known to be comorbid with each other (Kent et al., 1999; Rasmussen et al., 2001). Expressive disorder, a feature of both ASD and Down’s syndrome, shows up in the top-4 list as children are learning language at age 2; later the top-4 list is dominated by clinical features such as infections. The overall prevalence of infection-related terms is consistent with associations of immunodeficiency with both Down’s syndrome (Ram and Chinen, 2011) and ASD (Gupta et al., 2010), including increased ear infections specifically (Konstantareas and Homatidis, 1987b). Children with Down’s syndrome are more likely to have a variety of abnormal ocular features such as myopia (Shapiro and France, 1985) and abnormalities of the ear such as eustachian tube dysfunction (Pueschel, 1990; Shott et al., 2001). Sleep apnea is also common in children with Down’s syndrome (Marcus et al., 1991).

Topic 1 corresponds to children with ASD who go on to develop psychiatric disorders, and is very similar to the psychiatric subgroup reported by Doshi-Velez et al. (2013). As expected, there is a progression from ADHD at age 4, anxiety and conduct disorders at age 10, to episodic mood disorders at age 15 (other prevalent, but not top-4 terms at age 15 included depressive disorder and childhood psychoses). Psychiatric disorders are commonly reported among higher functioning children with ASD (Gillott et al., 2001; DeLong and Dwyer, 1988), and the progression of diagnoses makes sense because clinicians will usually avoid giving a young child a diagnosis for a severe psychiatric illness.

Topic 2 contains a combination of intellectual disability and epilepsy. It is similar to neurological subgroup reported by Doshi-Velez et al. (2013). Epilepsy is a common comorbidity of autism (Sherr, 2003a; Mouridsen SE, 1999), affecting close to 20% of children with ASD. Sherr (2003b) suggest that these three disorders—epilepsy, intellectual disability, and ASD—are linked through the ARX gene. Launonnier et al. (2004) find common genes between ASD and intellectual disability, and Sharp et al. (2008) report genomic underpinnings for epilepsy and intellectual disability. Again, a young child is less likely to be given a diagnosis of intellectual disability—it appears in our top-4 list at age 4—but other signs, such as symbolic dysfunction and developmental delays are noted from infancy.

Topic 3 tracks the progression of children with ASD and cerebral palsy. There are known correlations between cerebral palsy and infantile autism (Surén et al., 2012; Talkowski et al., 2012); early infections (seen at age 0) have also been associated with both cerebral palsy and autism spectrum disorders (Konstantareas and Homatidis, 1987a; Rosenhall et al., 1999).

Table 1: Top words from Dynamic Topic Model trained only on Electronic Health Records.

Topic	Year 0	Year 2	Year 4	Year 10	Year 15
Topic 0	Otitis Media, Down Syndrome, Acute upper respiratory infection, Unspecified viral infection	Expressive Language Disorder, Otitis Media, Down Syndrome, Chronic serous otitis media	Otitis Media, Expressive Language Disorder, Down Syndrome, Eustachian tube disorder	Down Syndrome, Eustachian tube disorder, Sensorineural Hearing Loss, Otitis Media	Down Syndrome, Eustachian tube disorder, Sleep Apnea, Myopia
Topic 1	Acute bronchitis, Asthma	Other specified pervasive developmental disorders, Asthma, Urea Cycle Disorders, Autistic Disorder	Other specified pervasive developmental disorders, Attention deficit hyperactivity disorder, Autistic Disorder, Developmental delay (disorder)	Attention deficit hyperactivity disorder, Other specified pervasive developmental disorders, Anxiety state, Conduct Disorder	Attention deficit hyperactivity disorder, Other specified pervasive developmental disorders, Anxiety state, episodic mood disorders
Topic 2	Other specified delays in development, Mixed development disorder, Viral and chlamydial infection, Developmental delay (disorder), Symbolic dysfunction	Infantile autism, Symbolic dysfunction, Developmental delay (disorder), Other specified delays in development	Symbolic dysfunction, Infantile autism, Unspecified intellectual disabilities, Epilepsy	Infantile autism, Unspecified intellectual disabilities, Epilepsy, Symbolic dysfunction	Infantile autism, Unspecified intellectual disabilities, Epilepsy, Unspecified, Generalized convulsive epilepsy,
Topic 3	Infantile cerebral palsy, Gastroesophageal reflux disease, Chronic respiratory disease in perinatal period, Deglutition Disorders	Infantile cerebral palsy, Quadriplegic Infantile Cerebral Palsy, Diplegic Infantile Cerebral Palsy, Deglutition Disorders	Infantile cerebral palsy, Quadriplegic Infantile Cerebral Palsy, Diplegic Infantile Cerebral Palsy, Deglutition Disorders	Quadraplegic Infantile Cerebral Palsy, Infantile cerebral palsy, allergic rhinitis, hay fever	Quadraplegic Infantile Cerebral Palsy, Infantile cerebral palsy, Hemiplegic cerebral palsy, Gastroesophageal reflux disease
Topic 4	Gastroesophageal reflux disease, Atrial septal defect within oval fossa, Hypoplastic Left Heart Syndrome, DiGeorge Syndrome	Gastroesophageal reflux disease, Deglutition Disorders, Asthma, Failure to Thrive	Gastroesophageal reflux disease, Muscle, ligation and fascia disorders, Developmental Coordination Disorder, Deglutition Disorders	Gastroesophageal reflux disease, Hypogammaglobulinemia, Asthma, Hematological Disease	Hypogammaglobulinemia, Gastroesophageal reflux disease, Adjustment Disorder With Mixed Anxiety and Depression, Hypertopia

Children with cerebral palsy are known to have difficulty swallowing (Sochaniwskyj et al., 1986) and reflux (Reyes et al., 1993). Horrath et al. (1999) also note an association between ASD and a number of gastrointestinal symptoms, including increased reflux.

Finally, topic 4 initially contains a variety of more severe multi-system disorders. Many are congenital anomalies (e.g. DigGeorge Syndrome and septal defects), which are more prevalent in ASD (Wier et al., 2006). It makes sense to see “failure to thrive”—usually diagnosed in early childhood—as one of the top diagnoses in this topic of severe illnesses. The later terms contain features common in ASD (GI symptoms, immunodeficiency) seen in earlier topics but without the associated Down’s syndrome or cerebral palsy. This topic is somewhat reminiscent of the multi-system subgroup in Doshi-Velez et al. (2013). More broadly, analyzing the same data set, we recover topics that resemble the subgroups discovered in Doshi-Velez et al. (2013) with the addition of specific trajectories for patients with ASD and Down’s syndrome and ASD and cerebral palsy.

Qualitative Examination of Topics: Social Media Table 2 shows a similar table for the θ_i DTM trained on the social media data alone. Even after filtering for only signs and symptoms, the extracted terms from the forum posts tend to focus more the symptoms of the child’s ASD rather than other comorbid conditions. ² Topic 3 seems to correspond to the most “traditional” ASD trajectory: with speech delays and tantrums early on. Emotional distress is a constant, and we see that bullying makes the top-4 list at age 10. Children with ASD are both more likely to bully (Montes and Haltnerman, 2007; Van Roessel et al., 2010) and be bullied (Lee et al., 2008), especially as they reach later grade school and early middle school years.

In general, terms such as tantrums and mental suffering are common in many of the topics. For example, topic 0 follows the trajectory of children with stereotypics (tic disorder, apraxias) common in ASD (Goldman et al., 2009). Pagramenta et al. (2010) suggest genetic commonalities between ASD and dyslexia, and Gillon and Moriarty (2007) note that children with speech apraxias are also at higher risk for dyslexia. However, there also exists a parallel set of terms starting with mental suffering starting at age 2 and ending with psychiatric problem at age 15. Even if some of the mental suffering terms are a mistaken reference to the challenges experienced by the caregiver, rather than the child, we can still say that forums generally contain more language pertaining to mental health.

Topic 1 describes emotional distress, including nightmares (while nightmares are not reported as common in the clinical literature, sleep disorders are very common and it may be that parents attribute sleep disorders to nightmares (Gail Williams et al., 2004)), as well as reactions that children may have to stress—temper tantrums and aggressive behaviors. These terms turn to phobic anxiety at later ages. We conjecture that this topic is the care-giver analog of EHR Topic 1 above, which followed the trajectories of patients with psychiatric disorders.

Like Topic 2 in the EHR, Topic 2 here describes developmental delays and epilepsy. However, we see abstract thought disorder rather than intellectual disability as well as symptoms such as staring. At age 15, emotional distress again makes the top-4 list, suggesting that most children with ASD face challenges as they grow older and interact more

² We were not able to incorporate clinical notes in this study, but it is possible that the clinical note would also tip the balance toward terms describing the patient’s ASD rather than other comorbidities.

with society. Topic 4 also has some psychiatric disorders, including aggressive behavior turning into emotional distress, bullying, and depression as the child ages.

Table 2: Top words from Dynamic Topic Model trained on only Social Media

Topic	Year 0	Year 2	Year 4	Year 10	Year 15
0	Infection, Apraxias, Developmental delay (disorder), Autistic Disorder	Autistic Disorder, Infection, Apraxias, Mental Suffering	Autistic Disorder, Apraxias, Tic disorder, Mental Suffering	Autistic Disorder, Dyslexia, Apraxias, Mental Suffering	Autistic Disorder, Apraxias, Tic disorder, Psychiatric problem
1	Autistic Disorder, Emotional distress, Abstract thought disorder, Temper tantrum	Autistic Disorder, Emotional distress, Abstract thought disorder, Aggressive behavior	Autistic Disorder, Emotional distress, Abstract thought disorder, Aggressive behavior	Autistic Disorder, Emotional distress, Temper tantrum, Aggressive behavior	Autistic Disorder, Emotional distress, Phobic anxiety disorder, Nightmares
2	Autistic Disorder, Abstract thought disorder, Temper tantrum, Staring	Autistic Disorder, Temper tantrum, Abstract thought disorder, Epilepsy	Autistic Disorder, Abstract thought disorder, Temper tantrum, Staring	Autistic Disorder, Abstract thought disorder, Temper tantrum, Epilepsy	Autistic Disorder, Abstract thought disorder, Asperger Syndrome, Emotional distress
3	Autistic Disorder, Speech Delay, Emotional distress, Temper tantrum	Autistic Disorder, Speech Delay, Emotional distress, Temper tantrum	Autistic Disorder, Emotional distress, Psychiatric problem, Speech Delay	Autistic Disorder, Emotional distress, Bullying, Asperger Syndrome	Autistic Disorder, Mental Suffering, Emotional distress, Apraxias
4	Autistic Disorder, Aggressive behavior, Nightmares, Apraxias	Autistic Disorder, Forgetting, Aggressive behavior, Mental Suffering	Autistic Disorder, Emotional distress, Temper tantrum, Confusion	Aggressive behavior, Emotional distress, Violent, Bullying, Forgetting	Emotional distress, Mental Depression, Violent, Mental Suffering

Qualitative Examination of Topics: Cross-corpora model Finally, we show the matching topics of the cross-corpora model in tables 3 and 4, as well as the overall proportions of each topic in figure 7. Again, we limit ourselves to a smaller topic model and show only a few top words, but we emphasize that in a clinical application these choices can be expanded and each topic examined in significantly more detail. What is most interesting for our purposes is the cross-corpora DTM allows us to see where top words in the corpora match, and where they do not.

Overall, the topics are closer to the EHR topics—likely a reflection of the fact that we had more EHR data. For example, topic 0 appears to be epilepsy topic (with pervasive developmental disorders replacing intellectual disability as a topic term, but reflecting a similar set of conditions). Epilepsy-related terms are also present in the social media version of the topic; however, we also see ADHD—also comorbid with epilepsy (Suren et al., 2012; Dunn et al., 2003)—present in both topics, likely because ADHD is commonly discussed on forums. We also dental caries, which are also associated with epilepsy (Anjounshoaa et al.,

2009), in the social media version of the topic. Such dental terms would not occur as often in the clinical records because children see their dentists outside the hospital system.

Topic 1 contains several psychiatric disorders with increasing severity (especially prominent in the EHR version of the topic). These show up as more general emotional distress and mental suffering in the forum topic. While most of the topics are present in similar relative proportions in both corpora (figure 7), topic 1 is the most common topic in the social media source and the least common topic in the electronic health records. We posit this difference may be because caregivers in general may be more focused on the mental health of their children (as seen in the social media-only topics), while the EHRs contain a range of specialties seen by the patient and perhaps disproportionately little about their mental health.

Topic 2 contains many infections, in both the social media and the EHR, which are consistent with the immunodeficiency-related topics discovered from EHR alone. Interestingly, asthma, an autoimmune disease, also appears in this topic; Becker (2007) posits that some ASDs, asthma, and inflammation may have a common autoimmune component. Doshi-Velez et al. (2013) also found a subgroup enriched for asthma. Obesity, associated with asthma (Beuther et al., 2006), also appears in this topic; here it seems that combining the sources resulted in a much clearer infections and autoimmune topic rather than the more diluted multi-system EHR topic 4.

Finally, topic 3 mirrors the cerebral palsy topic from the EHRs and topic 4 mirrors the Down’s syndrome topic. In the cerebral palsy topic (topic 3), we see more differences in the topics early on. Caregivers mention temper tantrums, speech delays, and abstract thought disorder—all features consistent with ASD and cerebral palsy—early on but the term cerebral palsy does not make the top-4 list. Later the terms are more similar across the two sources. Similarly, the caregiver version of the top-4 list for the ASD and Down’s syndrome topic (topic 4) includes more terms like expressive language disorder and symbolic dysfunction early on as well as stereotypic movements.

6. Related Work

Disease Progression Models Disease progression modeling is an important area in medical informatics. When biomarkers of interest are known, or disease stages have been labeled, supervised approaches can be used to predict disease stages given signs and symptoms; such supervised approaches have been applied to modeling the progression of Alzheimer’s disease (Zhou et al., 2012a). Other approaches use physiological models (De Winter et al., 2006) or meta-analyses of existing literature (Ito et al., 2010) to derive disease progression models.

One of the most popular data-driven approaches to learning disease progression models is to fit a hidden Markov Model (HMM) to the observations. The states of the HMM correspond to different stages of chronic diseases, and often left-to-right HMMs are used to model the fact that many disease progression processes are not reversible. Such models have been used to model disease progression in chronic kidney disease (Luo et al., 2013; Yang et al., 2014), Alzheimer’s disease (Sukkar et al., 2012), aneurysm screening (Jackson et al., 2003), and flu (Fan et al., 2015). Yang et al. (2014) allow the patient to have multiple conditions at the same time, treating each patient as having a mixture of disease pathways. Luo et al. (2013) take into account irregular sampling of data.

Table 3: Top words from Dynamic Topic Model trained on both SM and EHR data.

	Year 0	Year 2	Year 4	Year 10	Year 15
EHR Topic 0	Acute upper respiratory infection, Hearing Loss, gastroenteritis, Hirschsprung’s disease	Expressive Language Disorder, Developmental delay, Hearing Loss, Mixed development disorder	Expressive Language Disorder, Developmental delay, Epilepsy, Localization-related epilepsy	Epilepsy, Hearing Loss, Conduct Disorder, ADHD	Generalized intractable convulsive epilepsy, Conduct Disorder, Generalized intractable convulsive epilepsy, Epilepsy
SM Topic 0	Epilepsy, Acute upper respiratory infection, Mixed development disorder, Hemophilia B	Ehlers-Danlos Syndrome, Developmental delay, Ritual compulsion, Dental caries	Exanthema, Developmental delay, Pervasive Development Disorder, Metabolic Diseases	Hearing Loss, Mixed Conductive-Sensorineural Disorder, Hearing Loss, Dental caries	Generalized intractable convulsive epilepsy, Dental caries, ADHD, Grand Status Epilepticus
EHR Topic 1	Acute bronchiolitis, Redundant prepuce and phimosis, Common Cold, Epilepsy	Autistic Disorder, pervasive developmental disorders, Asthma, Urea Cycle Disorders	ADHD, Autistic Disorder, speech or language disorder, Urea Cycle Disorders	ADHD, Other specified pervasive developmental disorders, Tic disorder, Autistic Disorder	pervasive developmental disorder, ADHD, Psychotic Disorders, Depressive disorder, Emotional distress
SM Topic 1	Autistic Disorder, Emotional distress, Abstract thought disorder, Epilepsy	Autistic Disorder, Emotional distress, Mental Suffering, Aggressive behavior	Autistic Disorder, Emotional distress, Aggressive behavior, Tic disorder	Autistic Disorder, Emotional distress, Bullying, Aggressive behavior	Autistic Disorder, Emotional distress, Aggressive behavior, Mental Suffering
EHR Topic 2	Otitis Media, Atrial defect within oval fossa, Acute upper respiratory infection, Viral infection	Otitis Media, Asthma, Acute upper respiratory infection, Spina bifida	Otitis Media, Asthma, Spina bifida, Urinary specified viral infection	Asthma, Otitis Media, Developmental delay, Obesity	Hypogammaglobulinemia, Sleep Apnea, Asthma, Obesity
SM Topic 2	Acute upper respiratory infection, Atrial septal defect within oval fossa, Otitis Media, Vesicoureteral reflux,	Common Cold, Forgetting, Exanthema, Asthma	Common Cold, Forgetting, Asthma, Urinary tract infection	Exanthema, Common Cold, Obesity, Asthma	Developmental delay, Hypogammaglobulinemia, Enlargement of tonsil or adenoid, Anomalous pulmonary artery

Table 4: Top words from Dynamic Topic Model trained on both SM and EHR data.

	Year 0	Year 2	Year 4	Year 10	Year 15
EHR Topic 3	Gastroesophageal reflux disease, Deglutition Disorders, Congenital Hypothyroidism, Chronic respiratory disease in perinatal period	Infantile cerebral palsy, Deglutition Disorders, Gastroesophageal reflux disease, Quadriplegic Infantile Cerebral Palsy	Infantile cerebral palsy, Quadriplegic Infantile Cerebral Palsy, Gastroesophageal reflux disease, Deglutition Disorders	Infantile cerebral palsy, Quadriplegic Infantile Cerebral Palsy, Gastroesophageal reflux disease, Diplegic Infantile Cerebral Palsy	Quadriplegic Infantile Cerebral Palsy, Gastroesophageal reflux disease, Hemiplegic cerebral palsy, Intellectual disabilities
SM Topic 3	Disorders, Infantile cerebral palsy, Congenital Hypothyroidism, Gastroesophageal reflux disease	Speech Delay, Temper tantrum, Abstract thought disorder, Developmental delay	Abstract thought disorder, Temper tantrum, Developmental delay, Gastroesophageal reflux disease, Muscle, ligament and fascia disorders	Diplegic Infantile Cerebral Palsy, Myopia, Failure to Thrive, Other specified delays in development	Quadriplegic Infantile Cerebral Palsy, Infantile cerebral palsy, Generalized convulsive epilepsy, Other specified delays in development
EHR Topic 4	Down Syndrome, Atresia and stenosis of large intestine, Contact dermatitis, Middle ear conductive hearing loss	Down Syndrome, Infantile autism, Synbiotic dysfunction, Eustachian tube disorder	Synbiotic dysfunction, Infantile autism, Other speech pervasive developmental disorders,, Down Syndrome	Infantile autism, Down Syndrome, pervasive developmental disorders, Anxiety state	Infantile autism, Anxiety state, Down Syndrome, Intellectual disabilities
SM Topic 4	Down Syndrome, Middle ear conductive hearing loss, Eustachian tube disorder, Unspecified intellectual disabilities	Autistic disorder, Infantile autism, Expressive Language Disorder, Symbolic dysfunction	Autistic disorder, Eustachian tube disorder, Stereotypic Movement Disorder, Hay fever, Down Syndrome	Pervasive developmental disorders,, Stereotypic Movement Disorder, Hay fever, Asthma	Psychotic Disorders, Down Syndrome, Unspecified childhood psychosis, Other specific pervasive developmental disorders

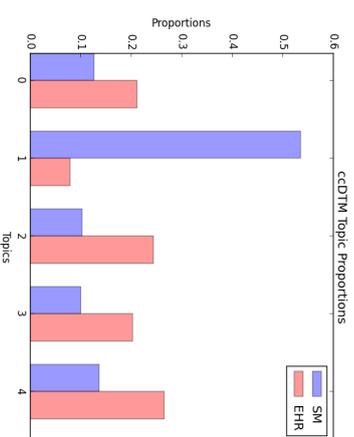


Figure 7: Overall topic popularities in both electronic health record and social media documents. Except for topic 1, most of the topics are present in similar proportions.

Others model disease progression with continuous time processes. Liu et al. (2013) model the progression of glaucoma with continuous-time HMMs, while Wang et al. (2014) use continuous-time Markov jump processes to model the progression of chronic obstructive pulmonary disease. Saeedi and Bouchard-Côté (2011) introduce gamma-exponential processes to model recurrent disease processes multiple sclerosis. These models can be adapted to incorporate individual-specific progression rates and treatment effects (Post et al., 2005).

Whether discrete or continuous time, all of these approaches involve discrete disease stages. However, often diseases evolve slowly over time. While we use a discrete time model in our work, a fundamental difference in our approach from those above is that we do not attempt to divide disease progression into stages, which might be artificial distinctions. Especially in developmental disorders, a more continuous progression model is more natural as a child’s development is a continuously evolving process. In this sense, perhaps closest in spirit to our work is the work of Zhou et al. (2014), which models disease progression with a matrix factorization that is smooth in the time dimension. Che et al. (2015) embed each time point of a patient into a latent space using a deep network.

In addition to being a natural way to model smoothly evolving diseases, our smoothness assumption allows us to easily incorporate cross-sectional data as well as longitudinal data. Requiring multiple visits to derive trajectories is often one of the factors that greatly limits the amount of data that can be used from a cohort: Doshi-Velez et al. (2013) used EHRs from the same hospital as us but were limited to only 4,927 patients with many visits rather than the 13,435 patients we study here (unlike Doshi-Velez et al. (2013) and other clustering-based studies, we do also not rely on ad-hoc patient similarity functions and intensive data pre-processing). Other studies that use smoothness assumptions in similar ways are Ross et al. (2014) and Li et al. (2012). Li et al. (2012) derive trajectories and then define an HMM from cross-sectional data through temporal bootstrap method that connects patients with similar features; their approach has no underlying model but rather

relies on patient similarities to build trajectories. Ross et al. (2014) derive lung capacity trajectories in chronic obstructive pulmonary disease from a cross-sectional cohort using Gaussian processes to encourage smoothness.

Dynamic Topic Models and Dynamical Systems Several techniques exist to model the temporal evolution of topics. Wang and McCallum (2006) consider the case in which the popularity of a topic changes over time, but each topic’s word proportions remain stationary. In contrast, dynamic topic models (Blei and Lafferty, 2006b; Wang et al., 2012) assume a topic’s word proportions smoothly evolve over time. Dynamic topic models have been applied to applications including discovering themes in research communities, (Furukawa et al., 2015), evolving patterns in software programs (Thomas et al., 2014), and the adoption of applications by smart phone users (Chua et al., 2015).

Topic models have also been developed for modeling multiple corpora. Wang et al. (2009) model correlations between the natural parameters for multiple corpora as a Gaussian random field. Paul (2009); Paul and Girju (2009); Zhai et al. (2004) model correlations between multiple corpora through a mixture of base and corpora-specific topics. Zhang et al. (2010) model the changing popularities of topics across three corpora—blogs, news, and message boards—using evolutionary hierarchical Dirichlet processes.

There also exists a related literature on modeling text as dynamical systems. Mikolov (2012) model dependencies in text as a recurrent neural network. Belanger and Kakade (2015) model text as a Gaussian linear dynamical system. Their model is misspecified in that it attributes zero probability mass to any observation, but they note the computational convenience of modeling occurrences of words with Gaussian variables rather than multinomials. While we are not modeling sequences of words, the idea modeling trends as linear dynamical system is close in spirit to our work.

In this context, we emphasize that the models we described in Section 3 are not novel—dynamic topic models and cross-corpora topic models both have well-established literatures. However, each topic model variant above relies on its own bespoke, implementation-intensive inference techniques that are often specific to that model. By using Pólya-gamma augmentation in our inference, we are able easily explore a variety of models. Moreover, to our knowledge, the application of dynamical system models of text to characterize disease progression is novel.

Disease Models from Social Media There exists a large body of work analyzing social media for information related to diseases. Chee et al. (2011) use personal health messages to predict adverse drug events, while Wilson and Brownstein (2009); Paul et al. (2015) use social media for disease surveillance. Elhadad et al. (2014); Jha and Elhadad (2010) characterize the linguistic properties of online forum text and use it to predict the cancer stage of the patient. Coppersmith et al. (2015) describe the task of identifying patients with depression and post-traumatic stress disorder from their Twitter posts. Unlike these works, our objective is understanding disease phenotypes and disease progression from social media, not prevalence or diagnosis.

7. Discussion and Conclusions

Modeling Choices Using dynamic topic models for modeling disease progression offers several advantages over more traditional clustering and HMM-based approaches. We do not require patients to belong to a single cluster or health state; they may have multiple disease processes varying in intensity over time, and each disease process is a smoothly varying, rather than discrete, structure. Because we can combine longitudinal and cross-sectional data, we can take advantage of much larger cohorts. Unlike clustering approaches, no ad-hoc patient similarity metrics are required, and unlike HMM-based approaches, we do not need to perform inference about what may have happened to patients in the gaps between visits.

Using Pólya-gamma augmentation allowed us to explore a variety of model choices without significantly changing our inference procedure: the static LDA, the DTM, and the ccDTM all used the same underlying Gibbs samplers and forward-backward code for Gaussian distributions. It would be interesting to investigate other alternatives, such as correlating intra-document topic proportions with a Pólya-gamma version of the correlated topic model (Blei and Lafferty, 2006a; Linderman et al., 2015) and correlating inter-document topic proportions from the same patient with an author topic model (Rosen-Zvi et al., 2004).

Another interesting direction for exploration is how the same topic appears in different corpora. In our work, we used the simplest approach in which topics for each corpus were isotropically perturbed versions of the latent disease process topic. This approach had the advantage of being able to easily interpret the latent topics probabilities $\mathbf{u}_{t,k}$. However, another option might be to learn a static emission matrix \mathbf{C}_l for each corpus l :

$$\begin{aligned}\beta_{l,k,l} &\equiv \pi_{SB}(\psi_{l,k,l}) \\ \psi_{l,k,l} &\equiv \mathbf{C}_l \mathbf{u}_{t,k} \\ \mathbf{u}_{t,k} &\sim \mathcal{N}(\mathbf{u}_{t,k} \mid \mathbf{A}, \mathbf{u}_{t-1,k}, \mathbf{B}\mathbf{B}^\top)\end{aligned}$$

Such an approach could allow the statistics of the pathological process $\mathbf{u}_{t,k}$ to have much lower dimensionality than the corpus-specific topic-word parameters ψ if \mathbf{C} were rectangular. It could also model systematic differences between document collections. For example, it would be exciting to incorporate general terms from social media that are not diseases or syndromes. However, the statistics $\mathbf{u}_{t,k}$ would be much harder to interpret; we chose our simpler model because $\mathbf{u}_{t,k}$ can readily be interpreted as the key terms of the disease process k . To create interpretable reduced-rank models, one approach might be to require that the emission matrix \mathbf{C}_l respect some clinician-interpretable ontology, as was done for static topic models in Doshi-Velez et al. (2015).

While Pólya-gamma augmentation allows for the exploration of many exciting models, there are some aspects of the inference that must be treated with care. Our application had a much higher dimensionality than the work of Linderman et al. (2015), and numerical errors accumulated during the recursive stick-breaking construction. Ordering the vocabulary by the prevalence of terms had a large impact on inference performance; deeply understanding the limitations of this augmentation approach on high-dimensionality data sets remains an interesting and open question. Our sampler was also fully uncollapsed; it would be interesting to see whether parameters in the cross-corpora models can be collapsed for

faster-mixing inference. As an alternative inference strategy, black-box variational inference (BBVI) (Ranganath et al., 2014) may offer convenient ways to work with such non-conjugate models.

Clinical Relevance: Autism Spectrum Disorders Clinical manifestations of autism spectrum disorders (ASD) beyond the core DSM criteria have been gaining increasing attention in recent years (Ming et al., 2008; Bauman, 2010; Coury, 2010; Smith, 1981; Kohane et al., 2012). Prior work in clustering phenotypes in ASD has largely relied on surveys and diagnostic tests. Miles et al. (2005) divide ASD into two clusters, “essential” and “complex” based on the manifestation of significant dysmorphology or microcephaly. They find that patients with “complex” ASDs have poorer outcomes, including lower IQ and more seizures. Wiggins et al. (2012) find clusters along disease severity, while Lane et al. (2010) discover sensory processing subtypes. Other studies find clusters along cognitive, language, and behavioral criteria (Wing and Gould, 1979; Ben-Sasson et al., 2008; Bitsika et al., 2008; Hu and Steinberg, 2009). Sacco et al. (2012) find patterns among both neurodevelopmental factors as well as immune and circadian dysfunction.

The phenotypes we find are consistent with these studies as well as the neurological, multi-system, and psychiatric disorder clusters characterized by Doshi-Velez et al. (2013). In addition, we find trajectories for patients with ASD and Down’s syndrome and ASD and cerebral palsy, two common comorbidities. Meanwhile, the topics associated with the social media—containing terms such as tantrums and bullying—provide a more complete window in the lives of these children. The fact that mental health terms dominate the social media topics is an indication of important stressors for these children and caregivers.

While it is reassuring that the topics associated with the clinical data are consistent with prior work, this study still has important limitations. Diagnostic codes are extremely noisy measures of disease state, and information extraction from social media is also a challenging process. In particular, our extraction is agnostic to whether a term applies to a current or past condition, to the child or to the caregiver. Our coarse processing was sufficient to discover credible trends, but better extraction methods will be required to validate the patterns we have discussed. Furthermore, while we have shown that our dynamic topic modeling approaches do better at predicting a patient’s future diagnoses than static models, there is still an important gap between improved predictions and clinically-useful predictions. Filling this gap will require using additional features in the models and rigorous data validation (e.g. through chart review).

Other Phenotyping Applications While we have focused on developmental disorders, the approaches described here could be relevant to discover the disease trajectories in other conditions. Indeed, almost all disease processes are likely best modeled as continuously evolving rather than having discrete stages. However, applying our approach to complex, chronic diseases such as chronic obstructive pulmonary disease, chronic kidney disease, or diabetes will have several challenges. First, unlike developmental disorders, which start at birth, one must now infer the age of onset from observational sources. Second, while disease processes are continuous, patients often visit when their situation has changed, leading clinicians to observe discrete changes. We hypothesize that a cross-corpora approach, using patient or caregiver-generated text or even outputs of patient-worn sensors (such as glucose monitors), could help discover these continuously evolving processes between sporadic

patient visits. Finally, many of these adult chronic diseases may have periods of remission between periods of high disease activity; these will also need to be modeled.

Conclusions In this work, we presented a dynamic topic modeling approach to modeling disease evolution. Our application of Poly-r-gamma augmentation to these models created a simple, unified framework for inference in dynamic topic models and cross-collection topic models. Applied to large collection of EHR and online forum posts describing patients with ASD, our models discovered disease trajectories that make sense in the context of the existing autism literature, and our cross-collection dynamic topic model had both high overall predictive performance and high predictive performance on predicting future patient trajectories. We are excited by the opportunity created by our approach to discover cross-corpora patterns of disease evolution in ASD as well as other diseases.

Acknowledgments

We are grateful to John Bickel and the Boston Children’s Hospital i2b2 team for providing the electronic health record data. Joy Ming, Sam Wiseman, and Andy Miller’s work on understanding autism forum data was directly valuable for in our pre-processing pipeline. We would also like to acknowledge support for this project from the National Science Foundation (NSF grant ACl-1544628).

References

- Ida Anjomshoa, Margaret E Cooper, and Alexandre R Vieira. Caries is associated with asthma and epilepsy. *European journal of dentistry*, 3(4):297, 2009.
- Jon Bato. Prevalence of autism spectrum disorders among children aged 8 years – autism and developmental disabilities monitoring network, 11 sites, united states, 2010. *CDC Morbidity and Mortality Weekly Report*, 63:1–21, March 2014.
- M. L. Bauman. Medical comorbidities in autism: challenges to diagnosis and treatment. *Neurotherapeutics*, 7:320327., 2010.
- Kevin G Becker. Autism, asthma, inflammation, and the hygiene hypothesis. *Medical hypotheses*, 69(4):731–740, 2007.
- David Belanger and Shann Kakade. A linear dynamical system model for text. In *Proceedings of the International Conference on Machine Learning*, 2015.
- A. Ben-Sasson, S. A. Cernak, G. I. Orsmond, H. Tager-Flusberg, M. B. Kadlec, and A. S. Carter. Sensory clusters of toddlers with autism spectrum disorders: differences in affective symptoms. *J Child Psychol Psychiatry*, 49(8):817–25, Aug 2008.
- David A Beutner, Scott T Weiss, and E Rand Sutherland. Obesity and asthma. *American Journal of Respiratory and Critical Care Medicine*, 174(2):112–119, 2006.

- V. Bitsika, C. F. Sharpley, and S. Orapeleng. An exploratory analysis of the use of cognitive, adaptive and behavioural indices for cluster analysis of asd subgroups. *J Intellect Disabil Res.*, 52(11):973–85, Nov 2008.
- David Blei and John Lafferty. Correlated topic models. *Advances in neural information processing systems*, 18:147, 2006a.
- David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the International Conference on Machine Learning*, pages 113–120. ACM, 2006b.
- David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.
- O. Bodenreider. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research*, 32:D267–D270, 2004.
- Coleen A Boyle, Sheree Boulet, Laura A Schieve, Robin A Cohen, Stephen J Blumberg, MarshalyN Yeargin-Allsopp, Susanna Visser, and Michael D Kogan. Trends in the prevalence of developmental disabilities in us children, 1997–2008. *Pediatrics*, pages peds–2010, 2011.
- Z. Che, D. Kale, W. Li, M. T. Bahadori, and Y. Liu. Deep computational phenotyping. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2015.
- Braut W Chee, Richard Berlin, and Bruce Schatz. Predicting adverse drug events from personal health messages. In *AMIA Annual Symposium Proceedings*, volume 2011, page 217. American Medical Informatics Association, 2011.
- Jianfei Chen, Jun Zhu, Zi Wang, Xun Zheng, and Bo Zhang. Scalable inference for logistic-normal topic models. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2445–2453, 2013.
- Freddy Chong Tat Chua, Richard J Oentaryo, and Ee-Peng Lim. Using linear dynamical topic model for inferring temporal social correlation in latent space. *arXiv preprint arXiv:1501.01270*, 2015.
- Glen Coppensmith, Mark Dredze, Craig Harman, Kristy Hollingshead, and Margaret Mitchell. Clpsych 2015 shared task: Depression and ptsd on twitter. In *NAACL Workshop on Computational Linguistics and Clinical Psychology*, 2015.
- D. Coury. Medical treatment of autism spectrum disorders. *Curr Opin Neurol*, 23:131136, 2010.
- Willem De Winter, Joost DeJongh, Teun Post, Bart Ploeger, Richard Urquhart, Ian Moules, David Eckland, and Meindert Danhof. A mechanism-based disease progression model for comparison of long-term effects of pioglitazone, metformin and gliclazide on disease processes underlying type 2 diabetes mellitus. *Journal of pharmacokinetics and pharmacodynamics*, 33(3):313–343, 2006.
- Elibol, NGUYEN, LINDERMAN, JOHNSON, HASHMI, AND DOSHI-VELEZ
- G. Robert DeLong and Judith T. Dwyer. Correlation of family history with specific autistic subgroups: Asperger’s syndrome and bipolar affective disease. *Journal of Autism and Developmental Disorders*, 18:593–600, 1988.
- Finale Doshi-Velez, Yaorong Ge, and Isaac Kohane. Comorbidity clusters in autism spectrum disorders: An electronic health record time-series analysis. *Pediatrics*, 10.1542.2013.
- Finale Doshi-Velez, Byron Wallace, and Ryan Adams. Graph-sparse lda: A topic model with structured sparsity. *AAAI*, 2015.
- David W Dunn, Joan K Austin, Jaroslaw Harezlak, and Walter T Ambrosius. Adhd and epilepsy in childhood. *Developmental Medicine & Child Neurology*, 45(01):50–54, 2003.
- Noémi Elhadad, Shaodian Zhang, Patricia Driscoll, and Samuel Brody. Characterizing the sublanguage of online breast cancer forums for medications, symptoms, and emotions. In *AMIA Annual Symposium Proceedings*, volume 2014, page 516. American Medical Informatics Association, 2014.
- Kai Fan, Marisa Eisenberg, Alison Walsh, Allison Aiello, and Katherine Heller. Hierarchical graph-coupled hmms for heterogeneous personalized health data. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2015.
- Takao Furukawa, Kaoru Mori, Kazuma Arino, Kazuhiro Hayashi, and Nobuyuki Shirakawa. Identifying the evolutionary process of emerging technologies: A chronological network analysis of world wide web conference sessions. *Technological Forecasting and Social Change*, 91:280–294, 2015.
- P Gail Williams, Lonnie L Sears, and AnnaMary Allard. Sleep problems in children with autism. *Journal of sleep research*, 13(3):265–268, 2004.
- Gail T Gillon and Brigid C Moriarty. Childhood apraxia of speech: children at risk for persistent reading and spelling disorder. In *Seminars in speech and language*, volume 28, pages 48–57, 2007.
- A Gilloff, F Furniss, and A Walter. Anxiety in high-functioning children with autism. *Autism*, 5(3):277–286, September 2001.
- Sylvie Goldman, Cuiling Wang, Miran W Salgado, Paul E Greene, Mimi Kim, and Isabelle Rapin. Motor stereotypes in children with autism and other developmental disorders. *Developmental Medicine & Child Neurology*, 51(1):30–38, 2009.
- Sudhir Gupta, Daljeet Samra, and Sudhanshu Agrawal. Adaptive and innate immune responses in autism: rationale for therapeutic use of intravenous immunoglobulin. *Journal of Clinical Immunology*, 30(1):90–96, 2010.
- Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

- Karoly Horvath, John C Pappadimitriou, Anna Rabszryn, Cintia Drachenberg, and J Tyson Tildon. Gastrointestinal abnormalities in children with autistic disorder. *The Journal of pediatrics*, 135(5):559–563, 1999.
- V.W. Hu and M.E. Steinberg. Novel clustering of items from the autism diagnostic interview-revised to define phenotypes within autism spectrum disorders. *Autism Res.*, 2(2):67–77, Apr 2009.
- Kaori Ito, Sima Ahadiel, Brian Corrigan, Jonathan French, Terence Fullerton, Thomas Tensfeldt, Alzheimer’s Disease Working Group, et al. Disease progression meta-analysis model in alzheimer’s disease. *Alzheimer’s & Dementia*, 6(1):39–53, 2010.
- Christopher H Jackson, Linda D Sharples, Simon G Thompson, Stephen W Duffy, and Elisabeth Couto. Multistate markov models for disease progression with classification error. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 52(2):193–209, 2003.
- Mukund Jha and Noémie Elhadad. Cancer stage prediction based on patient online discourse. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, pages 64–71. Association for Computational Linguistics, 2010.
- Lindsey Kent, Joanne Evans, Mohi Paul, and Margot Sharp. Comorbidity of autistic spectrum disorders in children with down syndrome. *Developmental Medicine & Child Neurology*, 41(3):153–158, 1999.
- I.S. Kohane, A. McMurry, G. Weber, D. MacFadden, and L. Rappaport. The co-morbidity burden of children and young adults with autism spectrum disorders. *PLoS ONE*, 7(4):2012.
- M M Konstantareas and S Homatidis. Ear infections in autistic and normal children. *J Autism Dev Disord*, 17(4):585–94, Dec 1987a.
- M Mary Konstantareas and Soula Homatidis. Brief report: Ear infections in autistic and normal children. *Journal of autism and developmental disorders*, 17(4):585–594, 1987b.
- A.E. Lane, R.L. Young, A.E. Baker, and M. T. Angley. Sensory processing subtypes in autism: association with adaptive behavior. *J Autism Dev Disord.*, 40(1):112–22, Jan 2010.
- Frédéric Laumonnier, Frédérique Bonnet-Brihault, Marie Gomot, Romuald Blanc, Albert David, Marie-Pierre Moizard, Martine Raynaud, Nathalie Ronce, Eric Lecomnier, Patrick Carvas, et al. X-linked mental retardation and autism are associated with a mutation in the nglr4 gene, a member of the neuroigin family. *The American Journal of Human Genetics*, 74(3):552–557, 2004.
- Li-Ching Lee, Rebecca A Harrington, Brian B Louie, and Craig J Newschaffer. Children with autism: Quality of life and parental concerns. *Journal of autism and developmental disorders*, 38(6):1147–1160, 2008.
- Y Li, S Swift, and A Tucker. Modelling and analysing the dynamics of disease progression from cross-sectional studies. *Journal of Biomedical Informatics*, 24(2), 2012.
- Scott W. Linderman, Matthew J. Johnson, and Ryan P. Adams. Dependent multinomial models made easy: Stick breaking with the py-gamma augmentation. In *arXiv:1506.05843*, 2015.
- Yu-Ying Liu, Hiroshi Ishikawa, Mei Chen, Gadi Wollstein, Joel S Schuman, and James M Rehg. Longitudinal modeling of glaucoma progression using 2-dimensional continuous-time hidden markov model. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2013*, pages 444–451. Springer, 2013.
- Lola Luo, Dylan Small, Walter F Stewart, and Jason A Roy. Methods for estimating kidney disease stage transition probabilities using electronic medical records. *EGEMS*, 1(3), 2013.
- Carole L Marcus, Thomas G Keens, Daisy B Bantista, Walter S von Pechmann, and Sally L Davidson Ward. Obstructive sleep apnea in children with down syndrome. *Pediatrics*, 88(1):132–139, 1991.
- Tomáš Mikolov. Statistical language models based on neural networks. *Presentation at Google, Mountain View, 2nd April, 2012*.
- J H Miles, T N Takahashi, S Bagley, P K Sahota, D F Vastow, C H Wang, R E Hillman, and J E Farmer. Essential versus complex autism: definition of fundamental prognostic subtypes. *Ann J Med Genet A.*, 135:171–180, June 2005.
- X. Ming, M. Brinacombe, J. Chaaban, B. Zimmerman-Bier, and G. C. Wagner. Autism spectrum disorders: concurrent clinical disorders. *Journal of Child Neurology*, 23:6–13, 2008.
- Guillermo Montes and Jill S Halperman. Bullying among children with autism and the influence of comorbidity with adhd: A population-based study. *Ambulatory Pediatrics*, 7(3):253–257, 2007.
- Isager T Mouridsen SE, Rich B. Epilepsy in disintegrative psychosis and infantile autism: a long-term validation study. *Dev Med Child Neurol*, 41:110114, 1999.
- Alistair T Pagnamenta, Elena Bacchelli, Maretha V de Jonge, Ghazala Mirza, Thomas S Scerif, Fiorella Minopoli, Andreas Chiochetti, Kerstin U Ludwig, Per Hoffmann, Silvia Paracchini, et al. Characterization of a family with rare deletions in chr1p5 and dock4 suggests novel risk loci for autism and dyslexia. *Biological psychiatry*, 68(4):320–328, 2010.
- Michael Paul. Cross-collection topic models: Automatically comparing and contrasting text. *Urbana*, 51:61801, 2009.
- Michael Paul and Roxana Girju. Cross-cultural analysis of blogs and forums with mixed-collection topic models. In *Proceedings of the 2009 Conference on Empirical Methods*

- in *Natural Language Processing: Volume 3-Volame 3*, pages 1408–1417. Association for Computational Linguistics, 2009.
- Michael Paul, Mark Dredze, David Broniatowski, and Nicholas Genovous. Worldwide influenza surveillance through twitter. In *AAAI Workshop on the World Wide Web and Public Health Intelligence*, 2015.
- Nienke Peters-Scheffer, Robert Didden, Hubert Korzilius, and Peter Sturmey. A meta-analytic study on the effectiveness of comprehensive aba-based early intervention programs for children with autism spectrum disorders. *Research in Autism Spectrum Disorders*, 5(1):60–69, 2011.
- Nicholas G Polson, James G Scott, and Jesse Windle. Bayesian inference for logistic models using Pólya-gamma latent variables. *Journal of the American Statistical Association*, 108(504):1339–1349, 2013.
- Teun M Post, Jan I Freijer, Joost DeJongh, and Meindert Danhof. Disease system analysis: basic disease progression models in degenerative disease. *Pharmaceutical research*, 22(7):1038–1049, 2005.
- Siegfried M Pueschel. Clinical aspects of down syndrome from infancy to adulthood. *American Journal of Medical Genetics*, 37(57):52–56, 1990.
- G Ram and J Chinen. Infections and immunodeficiency in down syndrome. *Clinical & Experimental Immunology*, 164(1):9–16, 2011.
- Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pages 814–822, 2014.
- Peder Rasmussen, Ola Börjesson, Elisabet Wentz, and Christopher Gillberg. Autistic disorders in down syndrome: background factors and clinical correlates. *Developmental Medicine & Child Neurology*, 43(11):750–754, 2001.
- AL Reyes, AJ Cash, SH Green, and IW Booth. Gastroesophageal reflux in children with cerebral palsy. *Child: care, health and development*, 19(2):109–118, 1993.
- Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 487–494. AUAI Press, 2004.
- Ulf Rosenhall, Viviann Nordin, Mikael Sandstrom, Gunilla Ahlsten, and Christopher Gillberg. Autism and hearing loss. *J Autism Dev Disord*, 29(5):349–357, October 1999.
- James C. Ross, Peter J. Castaldi, Michael H. Cho, and Jennifer G. Dy. Dual beta process priors for latent cluster discovery in chronic obstructive pulmonary disease. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 155–162, 2014.
- ELIBOL, NGUYEN, LINDERMAN, JOHNSON, HASHMI, AND DOSHI-VELEZ
- R. Sacco, C. Lenti, M. Saccaui, C. Paolo, B. Manzi, C. Bravaccio, and A. M. Pestic. Cluster analysis of autistic patients based on principal pathogenetic components. *Autism Research*, 5:137–147, 2012.
- Ardavan Saeedi and Alexandre Bouchard-Côté. Priors over recurrent continuous time processes. In *Advances in Neural Information Processing Systems*, pages 2052–2060, 2011.
- Michael B Shapiro and Thomas D France. The ocular features of down’s syndrome. *American journal of ophthalmology*, 99(6):659–663, 1985.
- Andrew J Sharp, Heather C Mefford, Kelly Li, Carl Baker, Cindy Skinner, Roger E Stevenson, Richard J Schroer, Francesca Novara, Manuela De Gregori, Roberto Ciccone, et al. A recurrent 15q13.3 microdeletion syndrome associated with mental retardation and seizures. *Nature genetics*, 40(3):322–328, 2008.
- E H Sherr. The arx story (epilepsy, mental retardation, autism, and cerebral malformations): one gene leads to many phenotypes. *Curr Opin Pediatr*, 6(15):567–571, December 2003a.
- Elliott H Sherr. The arx story (epilepsy, mental retardation, autism, and cerebral malformations): one gene leads to many phenotypes. *Current opinion in pediatrics*, 15(6):567–571, 2003b.
- Sally R Shott, Aileen Joseph, and Dorsey Heithaus. Hearing loss in children with down syndrome. *International journal of pediatric otorhinolaryngology*, 61(3):199–205, 2001.
- R.D. Smith. Abnormal head circumference in learning-disabled children. *Dev Med Child Neurol*, 23:626632, 1981.
- Alexander E Sochaniwskyj, Ruth M Koheil, Kazek Bablich, Morris Milner, and David J Kenny. Oral motor functioning, frequency of swallowing and drooling in normal children and in children with cerebral palsy. *Archives of physical medicine and rehabilitation*, 67(12):866–874, 1986.
- Rich Stoner, Maggie L Chow, Maureen P Boyle, Susan M Sunkin, Peter R Mouton, Subhojit Roy, Anthony Wynshaw-Boris, Sophia A Colamarino, Ed S Lein, and Eric Courchesne. Patches of disorganization in the neocortex of children with autism. *New England Journal of Medicine*, 370(13):1209–1219, 2014.
- Rafid Sukkar, Edward Katz, Yanwei Zhang, David Raunig, and Bradley T Wyman. Disease progression modeling using hidden markov models. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pages 2845–2848. IEEE, 2012.
- Pål Surén, Inger Johanne Bakken, Heidi Aase, Richard Chin, Nina Gunnes, Kari Kveim Lie, Per Magnus, Ted Reichborn-Kjennerud, Synnve Schjølberg, Anne-Siri Øyen, et al. Autism spectrum disorder, adhd, epilepsy, and cerebral palsy in norwegian children. *Pediatrics*, 130(1):e152–e158, 2012.

- Michael E. Talkowski, Gilles Maussion, Liam Crapper, Jill A. Rosenfeld, Ian Blumenthal, Carrie Hanscom, Colby Chang, Amelia Lindgren, Shabrin Pereira, Douglas Ruderer, Alpha B. Diallo, Juan Pablo Lopez, Gustavo Thredek, Elizabeth S. Chen, Carolina Giegik, David J. Harris, Va Lip, Yu An, Marta Biagioli, Marcy E. MacDonald, Michael Lin, Stephen J. Haggarty, Pamela Sklar, Shann Purcell, Manolis Kellis, Stuart Schwartz, Lisa G. Shaffer, Marvin R. Natowitz, Yiping Shen, Cynthia C. Morton, James F. Gusella, and Carl Ernst. Disruption of a large intergenic noncoding rna in subjects with neurodevelopmental disabilities. *The American Journal of Human Genetics*, 91:1128–1134, December 2012.
- Stephen W Thomas, Brann Adams, Ahmed E Hassan, and Dorothea Bloststein. Studying software evolution using topic models. *Science of Computer Programming*, 80:457–479, 2014.
- Eske Van Roekel, Ron HJ Scholte, and Robert Didden. Bullying among adolescents with autism spectrum disorders: Prevalence and perception. *Journal of autism and developmental disorders*, 40(1):63–73, 2010.
- Chong Wang, Bo Thiesson, Christopher Meek, and David Blei. Markov topic models. In *Proceedings of The Twelfth International Conference on Artificial Intelligence and Statistics*, 2009.
- Chong Wang, David Blei, and David Heckerman. Continuous time dynamic topic models. *arXiv preprint arXiv:1206.3298*, 2012.
- Xiang Wang, David Sontag, and Fei Wang. Unsupervised learning of disease progression models. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 85–94. ACM, 2014.
- Xuerui Wang and Andrew McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 424–433. ACM, 2006.
- Megan L Wier, Roxana Odouli, Cathleen K Yoshida, Judith K Grether, and Lisa A Croen. Congenital anomalies associated with autism spectrum disorders. *Developmental Medicine & Child Neurology*, 48(6):500–507, 2006.
- L. D. Wiggins, D. L. Robbins, L. B. Adamson, R. Bakeman, and C. C. Henrich. Support for a dimensional view of autism spectrum disorders in toddlers. *J Autism Dev Disord.*, 42(2):191–200, Feb 2012.
- Kunanan Wilson and John S Brownstein. Early detection of disease outbreaks using the internet. *Canadian Medical Association Journal*, 180(8):829–831, 2009.
- Jesse Windle, Nicholas G Polson, and James G Scott. Sampling Pólya-gamma random variates: alternate and approximate techniques. *arXiv preprint arXiv:1405.0506*, 2014.
- L. Wing and J Gould. Severe impairments of social interaction and associated abnormalities in children: epidemiology and classification. *J Autism Dev Disord.*, 9(1):11–29, Mar 1979.
- Jaewon Yang, Julian McAuley, Jure Leskovec, Paek LePendu, and Nigam Shah. Finding progression stages in time-evolving event sequences. In *Proceedings of the 23rd international conference on World wide web*, pages 783–794. ACM, 2014.
- Qing Theater Zeng. Consumer health vocabulary initiative. 2015. URL <http://consumerhealthvocab.org/>.
- ChengXiang Zhai, Anulya Vellyelli, and Bei Yu. A cross-collection mixture model for comparative text mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 743–748. ACM, 2004.
- Jianwen Zhang, Yangqin Song, Changshui Zhang, and Shixia Liu. Evolutionary hierarchical dirichlet processes for multiple correlated time-varying corpora. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1079–1088. ACM, 2010.
- Jiayu Zhou, Jun Liu, Vaibhav A Narayan, and Jieping Ye. Modeling disease progression via fused sparse group lasso. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1095–1103. ACM, 2012a.
- Jiayu Zhou, Fei Wang, Jianying Hu, and Jieping Ye. From micro to macro: data driven phenotyping by densification of longitudinal electronic medical records. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 135–144. ACM, 2014.
- Mingyan Zhou, Lingbo Li, David Dunson, and Lawrence Carin. Lognormal and gamma mixed negative binomial regression. In *Proceedings of the International Conference on Machine Learning*, volume 2012, page 1343, 2012b.

Appendix A. Data and Data Processing

A.1 Example Forum Post

Below is an example of a post. The age and CUIs that we extracted from the post are listed below.

hi my son is 13 nearly 14 and has this year become increasingly anxious and withdrawn in july his psychiatrist said to put him on prozac saying it might take the edge off his anxieties and allow him some positive experiences thus helping to lift the depression he seemed to be in i was not all that keen to be honest but my son who had been reluctant to take his other meds said he wanted to try it so we did he started on a small liquid dose and is now on tab a day will check exact dose if you want to know it despite my reservations his mood has really lifted he is still really challenging aggressive one track mind struggles to leave the house though maybe not so much but

to be honest he is back to where he was before the dip in terms of talking to me etc i am probably not explaining very well in feb half term adn easter hols the only interaction at all was to be negative call us names adn swear at us now he still does that but he also chats and has a laugh again which had stopped i have not seen any side effects and he says he likes taking it cos he feels better he cant explain anymore than that i discussed it with autism outreach recently and she said it is being used effectively in a lot of kids with asd and anxieties to take the edge off the anxieties dont get me wrong it hasn t solved all our issues at all but he just doesnt seem so saddont know if this is of any help at all so hard to put into words lol ps if you google most meds for kids ritalin prozac respiridone etc you get a lot of negatives adn not many positives and not a lot of balanced comment

age: 13

CUI: C0870663, C0424092, C06883607, C0023133, C0234856, C1273517, C1304698, C0001807, C0080151, C0011570, C0233730, C0004352

A.2 Forum Data Pre-Processing and Age Extraction

We used BeautifulSoup to obtain and parse all subforums of the websites www.asd-forum.org.uk, www.autismweb.com, and www.asdfriendly.org on June 29, 2015. We extracted the text, the user-id, and the time and date of posting for 21,206 threads from [asd-forum](http://asd-forum.com), 26,807 threads from [asd-friendly](http://asd-friendly.com), and 32,914 threads from [autismweb](http://autismweb.com), for a total of 80,927 threads. These threads contained a total of 664,954 posts. Figure 8 shows the regular expressions used to extract ages from the posts. Next, the outputs were filtered through a trie for a list of error terms such as *sec*, *wks*, *ft*, and *m* that might indicate another unit of measure; posts with such terms after the identified age were excluded. Finally, only posts with only one age were included, to avoid conflating information from multiple ages or multiple people.

Appendix B. MCMC Convergence

Figure 9 show the log-likelihoods for a characteristic run. Based on plots such as these, we determined that 100 iterations seemed to be more than enough for the sampler to find an optima.

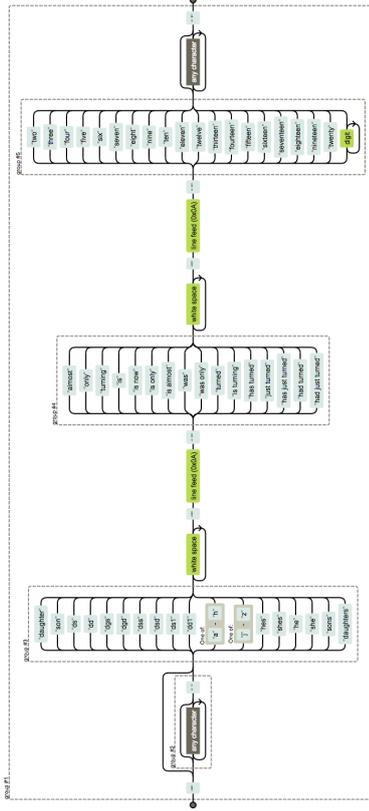


Figure 8: Chart showing the of regular expressions used to extract potential ages from posts. Outputs passing this filter were filtered through a second stage of processing to identify and remove cases where the number corresponded to a unit other than age in years.

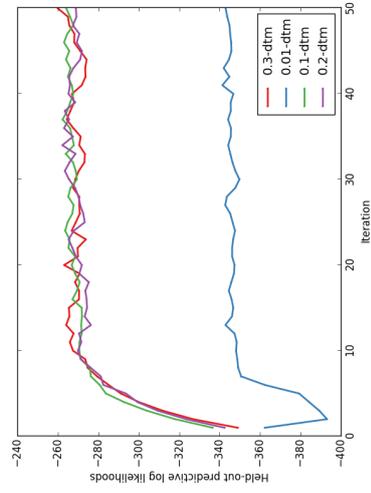


Figure 9: Log-likelihoods for a characteristic run.

Adjusting for Chance Clustering Comparison Measures

Simone Romano

SIMONE.ROMANO@UNIMELB.EDU.AU

Nguyen Xuan Vinh

VINH.NGUYEN@UNIMELB.EDU.AU

James Bailey

BAILEY.J@UNIMELB.EDU.AU

Karin Verspoor

KARIN.VERSPOOR@UNIMELB.EDU.AU

Dept. of Computing and Information Systems, The University of Melbourne, VIC, Australia.

Editor: Sebastian Nowozin

Abstract

Adjusted for chance measures are widely used to compare partitions/clustering of the same data set. In particular, the Adjusted Rand Index (ARI) based on pair-counting, and the Adjusted Mutual Information (AMI) based on Shannon information theory are very popular in the clustering community. Nonetheless it is an open problem as to what are the best application scenarios for each measure and guidelines in the literature for their usage are sparse, with the result that users often resort to using both. Generalized Information Theoretic (IT) measures based on the Tsallis entropy have been shown to link pair-counting and Shannon IT measures. In this paper, we aim to bridge the gap between adjustment of measures based on pair-counting and measures based on information theory. We solve the key technical challenge of analytically computing the expected value and variance of generalized IT measures. This allows us to propose adjustments of generalized IT measures, which reduce to well known adjusted clustering comparison measures as special cases. Using the theory of generalized IT measures, we are able to propose the following guidelines for using ARI and AMI as external validation indices: ARI should be used when the reference clustering has large equal sized clusters; AMI should be used when the reference clustering is unbalanced and there exist small clusters.

Keywords: Clustering Comparison, Clustering Validation, Adjustment for Chance, Generalized Information Theoretic Measures, Pair-Counting Measures

1. Introduction

Clustering comparison measures are used to compare partitions/clustering of the same data set. In the clustering community (Aggarwal and Reddy, 2013), they are extensively used for external validation when the ground truth clustering is available. A family of popular clustering comparison measures are measures based on pair-counting (Albatineh et al., 2006). This category comprises the well known similarity measures Rand Index (RI) (Rand, 1971) and the Jaccard coefficient (J) (Ben-Hur et al., 2001). Recently, information theoretic (IT) measures have been also extensively used to compare partitions (Strehl and Ghosh, 2003; Vinh et al., 2010). Given the variety of different possible measures, it is very challenging to identify the best choice for a particular application scenario (Wu et al., 2009).

The picture becomes even more complex if adjusted for chance measures are also considered. Adjusted for chance measures are widely used external clustering validation techniques

because they improve the interpretability of the results. Indeed, two important properties hold true for adjusted measures: they have constant baseline equal to 0 value when the partitions are random and independent, and they are equal to 1 when the compared partitions are identical. Notable examples are the Adjusted Rand Index (ARI) (Hubert and Arabie, 1985) and the Adjusted Mutual Information (AMI) (Vinh et al., 2009). It is common to see published research that validates clustering solutions against a reference ground truth clustering with the ARI or the AMI. Nonetheless there are still open problems: *there are no guidelines for their best application scenarios shown in the literature to date and authors often resort to employing them both and leaving the reader to interpret.*

Moreover, some clustering comparisons measures are susceptible to selection bias: when selecting the most similar partition to a given ground truth partition, clustering comparison measures are more likely to select partitions with many clusters (Romano et al., 2014). In Romano et al. (2014) it was shown that it is beneficial to perform statistical standardization to IT measures to correct for this bias. In particular, standardized IT measures help in decreasing this bias when the number of objects in the data set is small. Statistical standardization has not been applied to pair-counting measures yet in the literature. We solve this challenge in the current paper, and provide further results about the utility of measure adjustment by standardization.

In this work, we aim to *bridge the gap between the adjustment of pair-counting measures and the adjustment of IT measures.* In Furuichi (2006) and Simovici (2007) it has been shown that generalized IT measures based on the Tsallis q -entropy (Tsallis et al., 2009) are a further generalization of IT measures and some pair-counting measures such as RI. In this paper, we will exploit this useful idea to connect ARI and AMI. Furthermore using the same idea, we can perform statistical adjustment by standardization to a broader class of measures, including pair-counting measures.

A key technical challenge is to analytically compute the expected value and variance for generalized IT measures when the clusterings are random. To solve this problem, we propose a technique applicable to a broader class of measures we name \mathcal{L}_ϕ , which includes generalized IT measures as a special case. This generalizes previous work which provided analytical adjustments for narrower classes: measures based on pair-counting from the family \mathcal{L} (Albatineh et al., 2006), and measures based on the Shannon mutual information (Vinh et al., 2009, 2010). Moreover, we define a family of measures \mathcal{N}_ϕ which generalizes many clustering comparison measures. For measures which belong in this family, the expected value can be analytically approximated when the number of objects is large. Figure 1 depicts the families of measures discussed in this paper. Table 1 summarizes the development of this line of work over the past 30 years and positions our contribution. In summary, we make the following contributions:

- We define families of measures for which the expected value and variance can be computed analytically when the clusterings are random;
- We propose generalized adjusted measures to correct for the baseline property and for selection bias. This captures existing well known measures as special cases;
- We provide insights into the open problem of identifying the best application scenarios for clustering comparison measures, in particular the application scenarios for ARI and AMI.

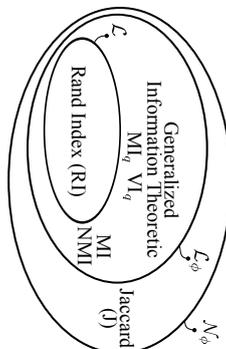


Figure 1: Families of clustering comparison measures discussed in this paper. We show how to analytically adjust measures in \mathcal{L}_ϕ and how to obtain approximations for the family \mathcal{N}_ϕ .

Year	Contribution	Reference
1985	Expectation of Rand Index (RI)	(Hubert and Arabie, 1985)
2006	Expectation and variance of $S \in \mathcal{L}$	(Albattineh et al., 2006)
2009	Expectation of Shannon Mutual Information (MI)	(Vinh et al., 2009)
2010	Expectation of Normalized Shannon MI (NMI)	(Vinh et al., 2010)
2014	Variance of Shannon MI	(Romano et al., 2014)
2016	Expectation and variance of $S \in \mathcal{L}_\phi$ Asymptotic expectation of $S \in \mathcal{N}_\phi$	This Work

Table 1: Work on adjusting clustering comparison measures carried out over the past 30 years. Information theoretic measures have been only recently adjusted for chance. In this paper, we bridge the gap between adjustment of pair-counting measures and information theoretic measures.

2. Comparing Partitions

Given two partitions (clusterings) U and V of the same data set of N objects, let $\{u_1, \dots, u_r\}$ and $\{v_1, \dots, v_c\}$ be the disjoint sets (clusters) for U and V respectively. Let $|u_i| = a_i$ for $i = 1, \dots, r$ denote the number of objects in the set u_i and $|v_j| = b_j$ for $j = 1, \dots, c$ denote the number of objects in v_j . Naturally, $\sum_{i=1}^r a_i = \sum_{j=1}^c b_j = N$. The overlap between the two partitions U and V can be represented in matrix form by a $r \times c$ contingency table \mathcal{M} where n_{ij} represents the number of objects in both u_i and v_j , i.e. $n_{ij} = |u_i \cap v_j|$. Also, we refer to $a_i = \sum_{j=1}^c n_{ij}$ as the row marginals and to $b_j = \sum_{i=1}^r n_{ij}$ as the column marginals. A contingency table \mathcal{M} is shown in Table 2.

Pair-counting measures between partitions, such as the Rand Index (RI) (Rand, 1971), might be defined using the following quantities: k_{11} , the pairs of objects in the same set in both U and V ; k_{00} the pairs of objects not in the same set in U and not in the same set in V ; k_{10} , the pairs of objects in the same set in U and not in the same set in V ; and k_{01} the pairs of objects not in the same set in U and in the same set in V . All these quantities can

		V				
		b_1	\dots	b_j	\dots	b_c
U	a_1	n_{11}	\dots	\cdot	\dots	n_{1c}
	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
	a_i	\cdot	\dots	n_{ij}	\cdot	\cdot
	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
a_r	n_{r1}	\dots	\cdot	\dots	n_{rc}	

Table 2: $r \times c$ contingency table \mathcal{M} related to two clusterings U and V . $a_i = \sum_j n_{ij}$ are the row marginals and $b_j = \sum_i n_{ij}$ are the column marginals.

be computed using the contingency table \mathcal{M} , for example:

$$k_{11} = \frac{1}{2} \sum_{i=1}^r \sum_{j=1}^c n_{ij}(n_{ij} - 1), \quad k_{00} = \frac{1}{2} \left(N^2 + \sum_{i=1}^r \sum_{j=1}^c n_{ij}^2 - \left(\sum_{i=1}^r a_i^2 + \sum_{j=1}^c b_j^2 \right) \right) \quad (1)$$

Using k_{00} , k_{11} , k_{10} , and k_{01} it is possible to compute similarity measures, e.g. RI, or distance measures, e.g. the Mirkin index $\text{MK}(U, V) \triangleq \sum_i a_i^2 + \sum_j b_j^2 - 2 \sum_{i,j} n_{ij}^2$, between partitions (Meiliš, 2007):

$$\text{RI}(U, V) \triangleq (k_{11} + k_{00}) / \binom{N}{2}, \quad \text{MK}(U, V) = 2(k_{10} + k_{01}) = N(N-1)(1 - \text{RI}(U, V)) \quad (2)$$

Information theoretic measures are instead defined for random variables but can also be used to compare partitions when we employ the empirical probability distributions associated to U , V , and the joint partition (U, V) . Let $\frac{a_i}{N}$, $\frac{b_j}{N}$, and $\frac{n_{ij}}{N}$ be the probability that an object falls in the set u_i , v_j , and $u_i \cap v_j$ respectively. We can therefore define the Shannon entropy with natural logarithms for a partition V as follows: $H(V) \triangleq -\sum_j \frac{b_j}{N} \ln \frac{b_j}{N}$. Similarly, we can define the entropy $H(U)$ for the partition U , the joint entropy $H(U, V)$ for the joint partition (U, V) , and the conditional entropies $H(U|V)$ and $H(V|U)$. Shannon entropy can be used to define the well known Mutual Information (MI) and employ it to compute similarity between partitions U and V :

$$\text{MI}(U, V) \triangleq H(U) - H(U|V) = H(V) - H(V|U) = H(U) + H(V) - H(U, V) \quad (3)$$

On contingency tables, MI is linearly related to G -statistics used for likelihood-ratio tests: $G = 2N\text{MI}$. In Meiliš (2007), using the Shannon entropy it was shown that the following distance, namely the Variation of Information (VI) is a metric:

$$\text{VI}(U, V) \triangleq 2H(U, V) - H(U) - H(V) = H(U|V) + H(V|U) = H(U) + H(V) - 2\text{MI}(U, V) \quad (4)$$

Information theoretic measures are extensively used to compare crisp partitions (Strehl and Ghosh, 2003; Vinh et al., 2010). Very recently they have also been used to compare fuzzy partitions (Lei et al., 2014a, 2016).

2.1 Generalized Information Theoretic Measures

Generalized Information Theoretic (IT) measures based on the generalized Tsallis q -entropy (Tsallis, 1988) can be defined for random variables (Furuchi, 2006) and also be applied to the task of comparing partitions (Simovici, 2007). Indeed, these measures have also seen recent application in the machine learning community. More specifically, it has been shown that they can act as proper kernels (Martins et al., 2009). Furthermore, empirical studies demonstrated that careful choice of q yields successful results when comparing the similarity between documents (Vila et al., 2011), decision tree induction (Maszczyk and Duch, 2008; Wang et al., 2015), and reverse engineering of biological networks (Lopes et al., 2011). It is important to note that the Tsallis q -entropy is equivalent to the Havrda-Charvat-Daroczy generalized entropy proposed in Havrda and Charvat (1967); Daroczy (1970). Results available in literature about these generalized entropies are equivalently valid for all the proposed versions.

Given $q \in \mathbb{R}^+ - \{1\}$, the generalized Tsallis q -entropy for a partition V is defined as follows: $H_q(V) \triangleq \frac{1}{q-1} \left(1 - \sum_j \left(\frac{a_j}{N}\right)^q\right)$. Similarly to the case of Shannon entropy, we have the joint q -entropy $H_q(U, V)$ and the conditional q -entropies $H_q(U|V)$ and $H_q(V|U)$. Conditional q -entropy is computed according to a weighted average parametrized on q . More specifically the formula for $H_q(V|U)$ is:

$$H_q(V|U) \triangleq \sum_{i=1}^r \left(\frac{a_i}{N}\right)^q H_q(V|u_i) = \sum_{i=1}^r \left(\frac{a_i}{N}\right)^q \frac{1}{q-1} \left(1 - \sum_{j=1}^c \left(\frac{n_{ij}}{a_i}\right)^q\right) \quad (5)$$

The q -entropy reduces to the Shannon entropy computed in nats for $q \rightarrow 1$.

In Furuchi (2006), using the fact that $q > 1$ implies $H_q(U) \geq H_q(U|V)$, it is shown that non-negative MI can be naturally generalized with q -entropy when $q > 1$:

$$\text{MI}_q(U, V) \triangleq H_q(U) - H_q(U|V) = H_q(V) - H_q(V|U) = H_q(U) + H_q(V) - H_q(U, V) \quad (6)$$

However, q values smaller than 1 are allowed if the assumption that $\text{MI}_q(U, V)$ is always positive can be dropped. In addition, generalized IT measures can be used to define the generalized Variation of Information distance (VI_q) which tends to VI in Eq. (4) when $q \rightarrow 1$:

$$\text{VI}_q(U, V) \triangleq H_q(U|V) + H_q(V|U) = 2H_q(U, V) - H_q(U) - H_q(V) = H_q(U) + H_q(V) - 2\text{MI}_q(U, V) \quad (7)$$

In Simovici (2007) it was shown that VI_q is a proper metric and interesting links were identified between measures for comparing partitions U and V . We state these links in Proposition 1 given that they set the fundamental motivation of our paper:

Proposition 1 (Simovici, 2007) *When $q = 2$ the generalized variation of information, the Minkin index, and the Rand index are linearly related: $\text{VI}_2(U, V) = \frac{1}{N^2} \text{MK}(U, V) = \frac{N-1}{N} (1 - \text{RI}(U, V))$.*

Generalized IT measures are not only a generalization of IT measures in the Shannon sense but also a generalization of pair-counting measures for particular values of q . Note that in literature there exist another well know generalization of entropy: the Renyi entropy (Renyi,

1961). This entropy is again parametrized on a real number q and is defined as follows: $R_q(V) \triangleq \frac{1}{1-q} \ln \left(\sum_j \left(\frac{b_j}{N}\right)^q\right)$. Because of the use of the logarithm for any value of q , the Renyi entropy does enable the generalization of pair-counting measures when $q = 2$. Therefore, in this paper we make use of generalized IT measures based on the Tsallis entropy.

2.2 Normalized Generalized IT Measures

To allow a more interpretable range of variation, a clustering similarity measure should be normalized: it should achieve its maximum at 1 when $U = V$. An upper bound to the generalized mutual information MI_q is used to obtain a normalized measure. MI_q can take different possible upper bounds (Furuchi, 2006). Here, we choose to derive another possible upper bound using Eq. (7) when we use the minimum value of $\text{VI}_q = 0$: $\max \text{MI}_q = \frac{1}{2}(H_q(U) + H_q(V))$. This upper bound is valid for any $q \in \mathbb{R}^+ - \{1\}$ and allows us to link different existing measures as we will show in the next sections of the paper. The Normalized Mutual Information with q -entropy (NMI_q) is defined as follows:

$$\text{NMI}_q(U, V) \triangleq \frac{\text{MI}_q(U, V)}{\max \text{MI}_q(U, V)} = \frac{\text{MI}_q(U, V)}{\frac{1}{2}(H_q(U) + H_q(V))} = \frac{H_q(U) + H_q(V) - H_q(U, V)}{\frac{1}{2}(H_q(U) + H_q(V))} \quad (8)$$

Even if $\text{NMI}_q(U, V)$ achieves its maximum 1 when the partitions U and V are identical, $\text{NMI}_q(U, V)$ is not a suitable clustering comparison measure. Indeed, it does not show constant baseline value equal to 0 when partitions are random. We explore this through an experiment. Given a dataset of $N = 100$ objects, we randomly generate uniform partitions U with $r = 2, 4, 6, 8, 10$ sets and V with $c = 6$ sets *independently* of each others. The average value of NMI_q over 1,000 simulations for different values of q is shown in Figure 2. It is reasonable to expect that when the partitions are independent, the average value of NMI_q is constant irrespectively of the number of sets r of the partition U . This is not the case. This behavior is unintuitive and misleading when comparing partitions. Computing the

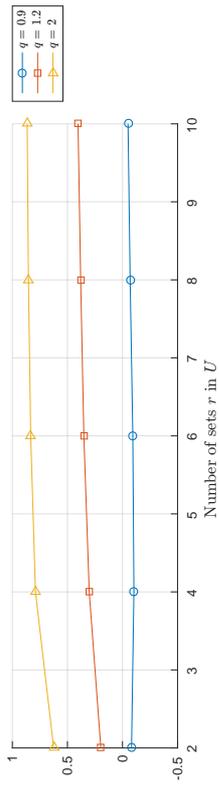


Figure 2: The baseline value of $\text{NMI}_q(U, V)$ between independent random partitions U and V . Despite the partitions are random, the baseline of NMI_q is not constant and depends on the number of sets of the partitions.

analytical expected value of generalized IT measures under the null hypothesis of random and independent U and V is important; it can be subtracted from the measure itself to adjust its baseline for chance such that the result is 0 when U and V are random. Given

Proposition 1, this strategy also allows us to generalize adjusted for chance pair-counting and Shannon IT measures.

3. Baseline Adjustment

In order to adjust the baseline of a similarity measure $S(U, V)$, we have to compute its expected value under the null hypothesis of independent random partitions U and V . We adopt the assumption of randomness used to adjust RI (Hubert and Arabie, 1985) and the Shannon MI (Vinh et al., 2009). This is formalized as follows:

Definition 1 (Random partitions) *The partitions U and V are generated independently and at random fixing the number of objects N and the marginals a_i and b_j .*

This is also denoted as the permutation or the hypergeometric model of randomness. We are able to compute the exact expected value for a similarity measure in the family \mathcal{L}_ϕ :

Definition 2 *Let \mathcal{L}_ϕ be the family of similarity measures $S(U, V) = \alpha + \beta \sum_{ij} \phi_{ij}(r_{ij})$ where α and β do not depend on the entries r_{ij} of the contingency table \mathcal{M} and $\phi_{ij}(\cdot)$ are bounded real functions.*

Intuitively, \mathcal{L}_ϕ represents the class of measures that can be written as a linear combination of $\phi_{ij}(r_{ij})$. A measure between partitions uniquely determines α , β , and ϕ_{ij} . However, not every choice of α , β , and ϕ_{ij} yields a meaningful similarity measure. \mathcal{L}_ϕ is a superset of the set \mathcal{L} defined in Albatineh et al. (2006) as the family of measures $S(U, V) = \alpha + \beta \sum_{ij} r_{ij}^2$, i.e. $S \in \mathcal{L}$ are special cases of measures in \mathcal{L}_ϕ with $\phi_{ij}(\cdot) = (\cdot)^2$. Figure 1 shows a diagram of the similarity measures discussed in Section 2.1 and their relationships.

Lemma 1 *If $S(U, V) \in \mathcal{L}_\phi$, when partitions U and V are random:*

$$E[S(U, V)] = \alpha + \beta \sum_{ij} E[\phi_{ij}(r_{ij})] \quad \text{where} \quad E[\phi_{ij}(r_{ij})] \quad \text{is} \quad (9)$$

$$\sum_{n_{ij}=\max\{0, a_i+b_j-N\}}^{\min\{a_i, b_j\}} \phi_{ij}(r_{ij}) \frac{a_i! b_j! (N - a_i)! (N - b_j)!}{N! n_{ij}! (a_i - n_{ij})! (b_j - n_{ij})! (N - a_i - b_j + n_{ij})!} \quad (10)$$

Lemma 1 extends the results in Albatineh and Niewiadomska-Bęgał (2011) showing exact computation of the expected value of measures in the family \mathcal{L} . Given that generalized IT measures belong in \mathcal{L}_ϕ we can employ this result to adjust them.

3.1 Baseline Adjustment for Generalized IT measures

Using Lemma 1 it is possible to compute the exact expected value of $H_q(U, V)$, $VI_q(U, V)$ and $MI_q(U, V)$:

Theorem 1 *When the partitions U and V are random:*

- i) $E[H_q(U, V)] = \frac{1}{q-1} \left(1 - \frac{1}{N^q} \sum_{ij} E[n_{ij}^q] \right)$ with $E[n_{ij}^q]$ from Eq. (10) with $\phi_{ij}(r_{ij}) = r_{ij}^q$;
- ii) $E[MI_q(U, V)] = H_q(U) + H_q(V) - E[H_q(U, V)]$;

$$\text{ii) } E[VI_q(U, V)] = 2E[H_q(U, V)] - H_q(U) - H_q(V).$$

It is worth noting that this approach is valid for any $q \in \mathbb{R}^+ - \{1\}$. We can use these expected values to adjust for baseline generalized IT measures. We use the method proposed in Hubert and Arabie (1985) to adjust similarity measures, such as MI_q and distance measures, such as VI_q :

$$AMI_q \triangleq \frac{MI_q - E[MI_q]}{\max MI_q - E[MI_q]} \quad AVI_q \triangleq \frac{E[VI_q] - VI_q}{E[VI_q] - \min VI_q} \quad (11)$$

VI_q is a distance measure, thus $\min VI_q = 0$. For MI_q we use the upper bound $\max MI_q = \frac{1}{2}(H_q(U) + H_q(V))$ as for NMI_q in Eq. (8). An exhaustive list of adjusted versions of Shannon MI can be found in Vinh et al. (2010), when the upper bound $\frac{1}{2}(H_q(U) + H_q(V))$ is used the authors named the adjusted MI as AMI_{sum} .

It is important to note that this type of adjustment turns distance measures into similarity measures, i.e., AVI_q is a similarity measure. It is also possible to maintain both the distance properties and the baseline adjustment using $NVI_q \triangleq VI_q/E[VI_q]$ which can be seen as a normalization of VI_q with the stochastic upper bound $E[VI_q]$ (Vinh et al., 2009). It is also easy to see that $AVI_q = 1 - NVI_q$. The adjustments in Eq. (11) also enable the measures to be normalized. AMI_q and AVI_q achieve their maximum at 1 when $U = V$ and their minimum is 0 when U and V are random partitions.

According to the chosen upper bound for MI_q , we obtain the nice analytical form shown in Theorem 2. Our adjusted measures quantify the discrepancy between the values of the actual contingency table and their expected value in relation to the maximum discrepancy possible, i.e. the denominator in Eq. (12). It is also easy to see that all measures in \mathcal{L}_ϕ resemble this form when adjusted.

Theorem 2 *Using $E[n_{ij}^q]$ in Eq. (10) with $\phi_{ij}(r_{ij}) = r_{ij}^q$, the adjustments for chance for $MI_q(U, V)$ and $VI_q(U, V)$ are:*

$$AMI_q(U, V) = AVI_q(U, V) = \frac{\sum_{ij} n_{ij}^q - \sum_{ij} E[n_{ij}^q]}{\frac{1}{2} \left(\sum_i a_i^q + \sum_j b_j^q \right) - \sum_{ij} E[n_{ij}^q]} \quad (12)$$

From now on we only discuss AMI_q given that it is identical to AVI_q . There are notable special cases for our proposed adjusted generalized IT measures. In particular, the Adjusted Rand Index (ARI) (Hubert and Arabie, 1985) is equal to AMI_2 . ARI is a classic measure, heavily used for validation in social sciences and the most popular clustering validity index.

Corollary 1 *It holds true that:*

- i) $\lim_{q \rightarrow 1} AMI_q = \lim_{q \rightarrow 1} AVI_q = AMI = AVI$ with Shannon entropy;
- ii) $AMI_2 = AVI_2 = ARI$.

Therefore, using the permutation model we can perform baseline adjustment to generalized IT measures. Our generalized adjusted IT measures are a further generalization of particular well known adjusted measures such as AMI and ARI. It is worth noting, that ARI is equivalent to other known measures for comparing partitions (Albatineh et al., 2006).

Furthermore, there is also a strong connection between ARI and Cohen's κ statistics used to quantify inter-rater agreement (Warrens, 2008). As final remark, we point out that our baseline adjustments can also be seen as statistical corrections for generalized information theoretic measures. It is indeed well known that information theoretic measures are severely biased when plug-in estimators are used, and many have worked on correcting this bias for decades: there exist in literature frequentist approaches (Paininski, 2003) as well as Bayesian approaches (Archer et al., 2013; Cerqueti, 2014) to reduce bias. In this section, we discussed an adjustment to obtain exact bias correction in particular when U and V are independent.

Computational complexity: The computational complexity of AMI_q in Eq. (12) is dominated by the computation of the sum of the expected value of each cell.

Proposition 2 *The computational complexity of AMI_q is $O(N \cdot \max\{r, c\})$.*

If all the possible contingency tables \mathcal{M} obtained by permutations were generated, the computational complexity of the exact expected value would be $O(N!)$. However, this can be dramatically reduced using properties of the expected value.

3.2 Experiments on Measure Baseline

Here we show that our adjusted generalized IT measures have a baseline value of 0 when comparing random partitions U and V . In Figure 3 we show the behavior of AMI_q , ARI, and AMI on the same experiment proposed in Section 2.2. They are all close to 0 with negligible variation when the partitions are random and independent. Moreover, it is interesting to see the equivalence of AMI_2 and ARI. On the other hand, the equivalence of AMI_q and AMI with Shannon entropy is obtained only at the limit $q \rightarrow 1$.

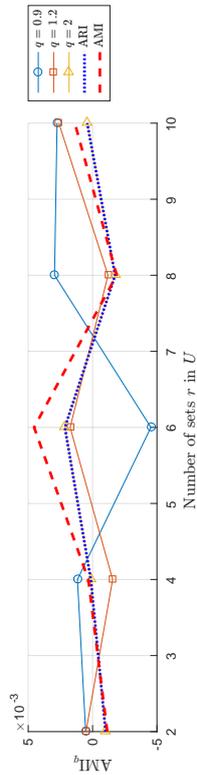


Figure 3: Baseline value of adjusted clustering comparison measures between two random partitions. When varying the number of sets for the random partition U , the value of $\text{AMI}_q(U, V)$ is always very close to 0 with negligible variation for any q .

We also point out that NMI_q does not show constant baseline when the relative size of the sets in U varies when U and V are random. In Figure 4, we generate random partitions V with $c = 6$ sets on $N = 100$ points, and random binary partitions U independently. $\text{NMI}_q(U, V)$ shows different behavior at the variation of the relative size of the biggest set

in U . This is unintuitive given that the partitions U and V are random and independent. We obtain the desired property of a baseline value of 0 with AMI_q .

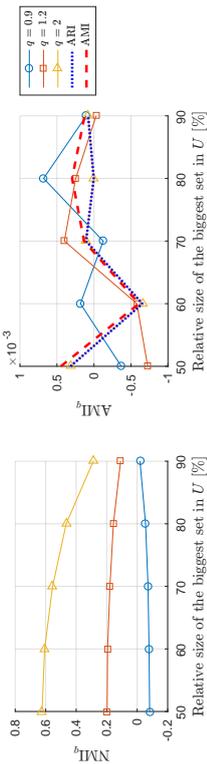


Figure 4: The left panel shows the baseline value of $\text{NMI}_q(U, V)$ between the random partitions U and V at the variation of the size of the sets in U . The right panel shows the baseline value of adjusted clustering comparison measures. When varying the relative size of one cluster for the random partition U , the value of $\text{AMI}_q(U, V)$ is always very close to 0 with negligible variation for any q .

3.3 Large Number of Objects

In this section, we introduce a very general family of measures which includes \mathcal{L}_ϕ . For measures belonging to this family, it is possible to find an approximation of their expected value when the number of objects N is large. This allows us to identify approximations for the expected value of measures in \mathcal{L}_ϕ as well as for measures not in \mathcal{L}_ϕ , such as the Jaccard coefficient as shown in Figure 1.

Let \mathcal{N}_ϕ be the family of measures which are *non-linear* combinations of $\phi_{ij}(n_{ij})$:

Definition 3 *Let \mathcal{N}_ϕ be the family of similarity measures $S(U, V) = \phi(\frac{a_i b_j}{N}, \dots, \frac{n_{ij}}{N}, \dots, \frac{b_i c_j}{N})$ where ϕ is a bounded real function as N reaches infinity.*

Note that \mathcal{N}_ϕ is a generalization of \mathcal{L}_ϕ . At the limit of large number of objects N , it is possible to compute the expected value of measures in \mathcal{N}_ϕ under random partitions U and V using only the marginals of the contingency table \mathcal{M} :

Lemma 2 *If $S(U, V) \in \mathcal{N}_\phi$, then $\lim_{N \rightarrow \infty} E[S(U, V)] = \phi(\frac{a_i b_j}{N}, \dots, \frac{a_i b_j}{N}, \dots, \frac{a_i b_j}{N})$.*

In Morey and Agresti (1984) the expected value of the RI was computed using an approximated value based on the multinomial distribution. It turns out this approximated value is equal to what we obtain for RI using Lemma 2. The authors of Albatineh et al. (2006) noticed that the difference between the approximation and the expected value obtained with the hypergeometric model is small on empirical experiments when N is large. We point out that this is a natural consequence of Lemma 2 given that $\text{RI} \in \mathcal{L}_\phi \subseteq \mathcal{N}_\phi$. Moreover, the multinomial distribution was also used to compute the expected value of the Jaccard coefficient (J) in Albatineh and Niewiadomska-Bugaj (2011), obtaining good results on empirical experiments with many objects. Again, this is a natural consequence of Lemma 2

given that $J \in \mathcal{N}_\phi$ but $J \notin \mathcal{L}_\phi$. Indeed, the Jaccard coefficient does not allow analytical adjustment using the hypergeometric model but it allows an approximation using Lemma 2.

Generalized IT measures belong in $\mathcal{L}_\phi \subseteq \mathcal{N}_\phi$. Therefore we can employ Lemma 2. When the number of objects is large, the expected value under random partitions U and V of $H_q(U, V)$, $MI_q(U, V)$, and $VI_q(U, V)$ in Theorem 1 depends only on the entropy of the partitions U and V , i.e., just the marginals of the contingency table must be taken into account:

Theorem 3 *It holds true that:*

- i) $\lim_{N \rightarrow +\infty} E[H_q(U, V)] = H_q(U) + H_q(V) - (q-1)H_q(U)H_q(V)$;
- ii) $\lim_{N \rightarrow +\infty} E[MI_q(U, V)] = (q-1)H_q(U)H_q(V)$;
- iii) $\lim_{N \rightarrow +\infty} E[VI_q(U, V)] = H_q(U) + H_q(V) - 2(q-1)H_q(U)H_q(V)$.

Result i) recalls the property of non-additivity that holds true for random variables (Furuidi, 2006). Figure 5 shows the behavior of $E[H_q(U, V)]$ when the partitions U and V are generated uniformly at random. V has $c = 6$ sets and U has r sets. In this case, $H_q(U) + H_q(V) - (q-1)H_q(U)H_q(V)$ appears to be a good approximation already for $N = 1000$. In particular, the approximation is good when the number of objects N is big with regards to the number of cells of the contingency table in Table 2: i.e., when $\frac{N}{rc}$ is large enough.

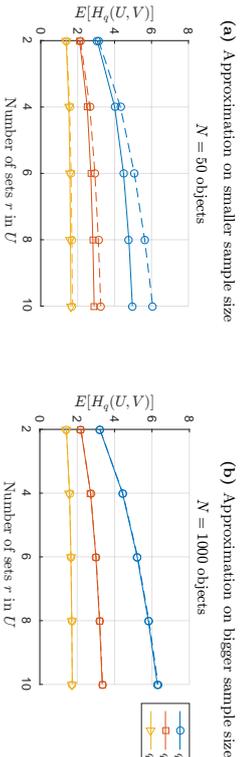


Figure 5: The left panel shows the average value of $E[H_q(U, V)]$ between random partitions U and V when they are induced on $N = 50$ objects. The right panel shows results for $N = 1000$ objects. $E[H_q(U, V)]$ are plotted using a solid line and their limit value $H_q(U) + H_q(V) - (q-1)H_q(U)H_q(V)$ is plotted using a dashed line. The solid line coincides approximately with the dashed one in (4b) when $N = 1000$. The limit value is a good approximation for $E[H_q(U, V)]$ when $\frac{N}{rc}$ is large enough.

From point ii) follows the result proved in Vinh et al. (2010) to connect the adjusted mutual information to the widely used Normalized Mutual Information (NMI) based on Shannon entropy (Strehl and Ghosh, 2003):

Theorem 4 (Vinh et al., 2010) *It holds true that:*

$$\lim_{N \rightarrow +\infty} \text{AMI}(U, V) = \text{NMI}(U, V)$$

NMI is easier to compute than AMI and it is less prone to computer precision errors. The analysis provided in this section aims at broadening the possible clustering comparison measures that can be adjusted: as long as a measure belongs in \mathcal{N}_ϕ , its expected value at large N can be computed and thus it can be adjusted. Moreover, we saw that adjusted measures in $\mathcal{L}_\phi \subseteq \mathcal{N}_\phi$ have simpler formulas at large N . The adjustments at large N are faster to compute and less prone to computer precision errors.

In the next section we put forward a theoretical analysis on the best choice between AMI and ARI when validating clustering solutions. Moreover being NMI equal to AMI at large N , the next section helps also to understand when to use either NMI or ARI.

4. Application Scenarios for AMI_q

In this section we aim to answer to the question: *Given a reference ground truth clustering V , which is the best choice for q in $\text{AMI}_q(U, V)$ to validate the clustering solution U ?* By answering this question, we implicitly identify the application scenarios for ARI and AMI given the results in Corollary 1. This is particularly important for external clustering validation. Nonetheless, there are a number of other applications where the task is to find the most similar partition to a reference ground truth partition: e.g., categorical feature selection (Vinh et al., 2014), decision tree induction (Criminisi et al., 2012), generation of alternative or multi-view clusterings (Miller et al., 2013), or the exploration of the clustering space with the Meta-Clustering algorithm (Caruana et al., 2006; Lei et al., 2014b) to list a few.

Different values for q in AMI_q yield to different biases. The source of these biases can be identified by analyzing the properties of the q -entropy. In Figure 6 we show the q -entropy for a binary partition at the variation of the relative size p of one cluster. This can be analytically computed: $H_q(p) = \frac{1}{q-1}(1-p)^p - (1-p)^q$. *The range of variation for $H_q(p)$ is much bigger if q is small.* More specifically, when q is small, the difference in entropy between an unbalanced partition and a balanced partition is big.

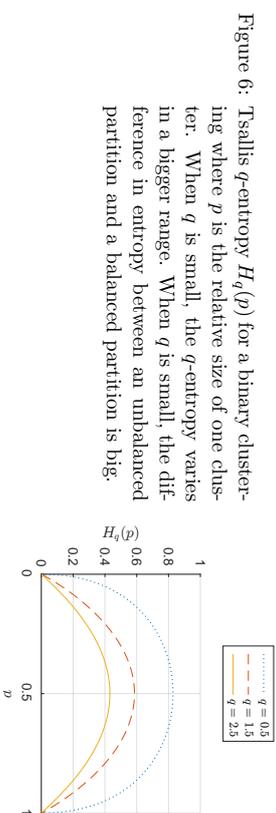


Figure 6: Tsallis q -entropy $H_q(p)$ for a binary clustering where p is the relative size of one cluster. When q is small, the q -entropy varies in a bigger range. When q is small, the difference in entropy between an unbalanced partition and a balanced partition is big.

Let us focus on an example. Let V be a reference clustering with 3 clusters of size 50 each, and let U_1 and U_2 be two clustering solutions with the same number of clusters and same cluster sizes. The contingency tables for U_1 and U_2 are shown on Figure 7. Given that both contingency tables have the same marginals, the only difference between $\text{AMI}_q(U_1, V)$ and $\text{AMI}_q(U_2, V)$ according to Eq. (11) lies in MI_q . Given that both solutions U_1 and U_2

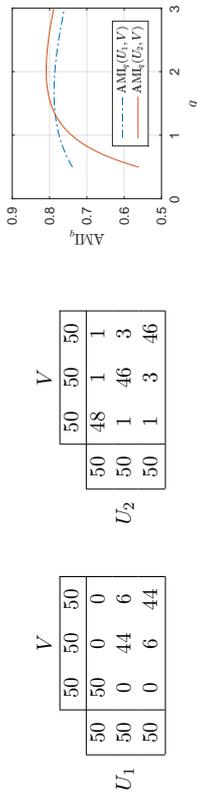


Figure 7: Ground truth clustering V compared in turn to the clustering solutions U_1 and U_2 . AMI_q with small q prefers the solution U_1 because there exists one pure cluster: i.e., there is one cluster which contains elements from only one cluster in the reference clustering V .

are compared against V , the only term that varies in $\text{MI}_q(U, V) = H_q(V) - H_q(V|U)$ is $H_q(V|U)$. In order to identify the clustering solution that maximizes AMI_q we have to analyze the solution that decreases $H_q(V|U)$ the most. $H_q(V|U)$ is a weighted average of the entropies $H_q(V|u_i)$ computed on the rows of the contingency table as shown in Eq. (5), and this is sensitive to values equal to 0. Given the bigger range of variation of H_q for small q , small q implies higher sensitivity to row entropies of 0. Therefore, small values of q tends to decrease $H_q(V|U)$ much more if the clusters in the solution U are pure: i.e., clusters contain elements from only one cluster in the reference clustering V . In other words, AMI_q with *small q prefers pure clusters in the clustering solution*.

When the marginals in the contingency tables for two solutions are different, another important factor in the computation of AMI_q is the normalization coefficient $\frac{1}{2}(H_q(U) + H_q(V))$. Balanced solutions U will be penalized more by AMI_q when q is small. Therefore, AMI_q with *small q prefers unbalanced clustering solutions*. To summarize, AMI_q with small q such as $\text{AMI}_{0.5}$ or $\text{AMI}_1 = \text{AMI}$ with Shannon entropy:

- Is biased towards pure clusters in the clustering solutions;
 - Prefers unbalanced clustering solutions.
- By contrary, AMI_q with bigger q such as $\text{AMI}_{2.5}$ or $\text{AMI}_2 = \text{ARI}$:
- Is less biased towards pure clusters in the clustering solution;
 - Prefers balanced clustering solutions.

Given a reference clustering V , these biases can guide the choice of q in AMI_q to identify more suitable clustering solutions.

4.1 Use AMI_q with small q such as $\text{AMI}_{0.5}$ or $\text{AMI}_1 = \text{AMI}$ when the reference clustering is unbalanced and there exist small clusters

If the reference cluster V is unbalanced and presents small clusters, AMI_q with small q might prefer more appropriate clustering solutions U . For example, in Figure 8 we show two contingency tables associated to two clustering solutions U_1 and U_2 for the reference

clustering V with 4 clusters of size $[10, 10, 10, 70]$ respectively. When there exist small clusters in the reference V their identification has to be *precise* in the clustering solution. In the solution U_1 looks arguably better than U_2 because it shows many pure clusters. In this scenario we advise the use of $\text{AMI}_{0.5}$ or $\text{AMI}_1 = \text{AMI}$ with Shannon entropy because it gives more weight to the clustering solution U_1 .

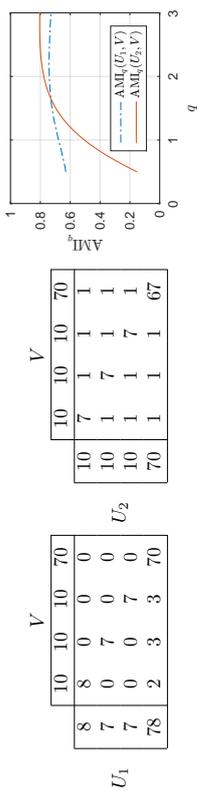


Figure 8: Ground truth clustering V compared in turn to the clustering solutions U_1 and U_2 . V is unbalanced and presents small clusters. When the reference clustering has small clusters their identification in the solution has to be *precise*. Therefore U_1 appears to be a better solution than U_2 . AMI_q with small q prefers the solution U_1 because its clusters are pure. In this scenario we advise the use of $\text{AMI}_{0.5}$ or $\text{AMI}_1 = \text{AMI}$.

If the number of objects N is large, AMI is equivalent to NMI according to Theorem 4. Therefore, when the reference clustering is unbalanced, there exist small clusters, and N is large, it is advisable to use NMI rather than ARI .

4.2 Use AMI_q with big q such as $\text{AMI}_{2.5}$ or $\text{AMI}_2 = \text{ARI}$ when the reference clustering has big equal sized clusters

If V is a reference clustering with big equal size clusters it is less crucial to have precise clusters in the solution. Indeed, precise clusters in the solution penalize the *recall* of clusters from the reference. In this case, AMI_q with bigger q might prefer more appropriate solutions. In Figure 9 we show two clustering solutions U_1 and U_2 for the reference clustering V with 4 equal size clusters of size 25. The solution U_2 looks better than U_1 because each of its clusters identifies more elements from particular clusters in the reference. Moreover, U_2 has to be preferred to U_1 because it consists in 4 equal sized clusters as the reference clustering V consists in equal sized clusters. In this scenario we advise the use of $\text{AMI}_{2.5}$ or $\text{AMI}_2 = \text{ARI}$ because it gives more importance to the solution U_2 .

If the number of objects N is large, AMI is equivalent to NMI according to Theorem 4. Therefore, when the reference clustering is balanced with big equal sized clusters and N is large, it is advisable to use ARI rather than NMI .

5. Standardization of Clustering Comparison Measures

The selection of the most similar partition U to a reference partition V is biased according to the chosen similarity measure, the number of sets r in U , and their relative size.

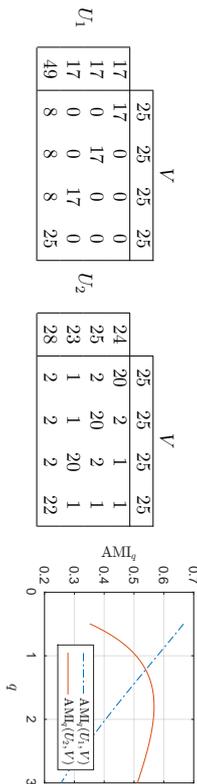


Figure 9: Ground truth clustering V compared in turn to the clustering solutions U_1 and U_2 . V shows equal size clusters. U_2 appears to be a better solution than U_1 because its clusters are more balanced in size than the clusters in U_1 . When the reference clustering has big equal sized clusters their precise identification is less crucial. AMI_q with big q prefers the solution U_2 because it is less biased to pure clusters in the solution. In this scenario we advise the use of $AMI_{2.5}$ or $AMI_2 = ARI$.

This phenomena is known as *selection bias* and it has been extensively studied in decision trees (White and Liu, 1994). Researchers in this area agree that in order to achieve unbiased selection of partitions, distribution properties of similarity measures have to be taken into account (Dobra and Gehrke, 2001; Shih, 2004; Hothorn et al., 2006). Using the permutation model, we proposed in Romano et al. (2014) to analytically standardize the Shannon MI by subtraction of its expected value and division by its standard deviation. In this section, we discuss how to achieve analytical standardization of measures $S \in \mathcal{L}_{\phi^*}$.

In order to standardize a measure, we must analytically compute its variance:

Lemma 3 *If $S(U, V) \in \mathcal{L}_{\phi^*}$, when partitions U and V are random:*

$$\text{Var}(S(U, V)) = \beta^2 \left(E \left[\left(\sum_{i,j} \phi_{ij}(n_{ij}) \right)^2 \right] - \left(\sum_{i,j} E[\phi_{ij}(n_{ij})] \right)^2 \right),$$

where

$$E \left[\left(\sum_{i,j} \phi_{ij}(n_{ij}) \right)^2 \right]$$

is equal to

$$\begin{aligned} & \sum_{i,j} \sum_{\tilde{n}_{i,j}} \phi(n_{ij}) P(n_{ij}) \cdot \left[\phi_{ij}(n_{ij}) + \sum_{\tilde{n}_{i,j} \neq n_{i,j}} \sum_{\tilde{n}_{i,j}} \phi_{ij}(\tilde{n}_{i,j}) P(\tilde{n}_{i,j}) + \right. \\ & \quad \left. + \sum_{\tilde{n}_{i,j} \neq \tilde{n}_{i,j'}} \sum_{\tilde{n}_{i,j'}} P(\tilde{n}_{i,j'}) \left(\phi_{ij}(\tilde{n}_{i,j}) + \sum_{\tilde{n}_{i,j} \neq \tilde{n}_{i,j'}} \sum_{\tilde{n}_{i,j'}} \phi_{ij}(\tilde{n}_{i,j'}) P(\tilde{n}_{i,j'}) \right) \right] \end{aligned} \quad (13)$$

with $n_{ij} \sim \text{HYP}(a_i, b_j, N)$, $\tilde{n}_{i,j} \sim \text{HYP}(b_j - n_{ij}, a_i, N - a_i)$, $\tilde{n}_{i,j'} \sim \text{HYP}(a_i - n_{ij}, b_j', N - b_j)$, $\tilde{n}_{i,j'} \sim \text{HYP}(a_i, b_j', N - a_i)$ *hypergeometric random variables*.

We can use the expected value to standardize measures $S \in \mathcal{L}_{\phi^*}$ such as generalized IT measures.

5.1 Standardization of Generalized IT Measures

The variance under the permutation model of generalized IT measures is:

Theorem 5 *Using Eqs. (10) and (13) with $\phi_{ij}(\cdot) = (\cdot)^q$, when the partitions U and V are random:*

- i) $\text{Var}(H_q(U, V)) = \frac{1}{(q-1)^2 N^{2q}} \left(E[(\sum_{i,j} n_{ij}^q)^2] - (\sum_{i,j} E[n_{ij}^q])^2 \right)$;
- ii) $\text{Var}(MI_q(U, V)) = \text{Var}(H_q(U, V))$
- iii) $\text{Var}(VI_q(U, V)) = 4\text{Var}(H_q(U, V))$

We define the standardized version of the similarity measure MI_q (SMI_q), and the standardized version of the distance measure VI_q (SVI_q) as follows:

$$SMI_q \triangleq \frac{MI_q - E[MI_q]}{\sqrt{\text{Var}(MI_q)}}, \quad SVI_q \triangleq \frac{E[VI_q] - VI_q}{\sqrt{\text{Var}(VI_q)}}, \quad (14)$$

As for the case of AMI_q and AVI_q , it turns out that SMI_q is equal to SVI_q :

Theorem 6 *Using Eqs. (10) and (13) with $\phi_{ij}(\cdot) = (\cdot)^q$, the standardized $MI_q(U, V)$ and the standardized $VI_q(U, V)$ are:*

$$SMI_q(U, V) = SVI_q(U, V) = \frac{\sum_{i,j} n_{ij}^q - \sum_{i,j} E[n_{ij}^q]}{\sqrt{E[(\sum_{i,j} n_{ij}^q)^2] - (\sum_{i,j} E[n_{ij}^q])^2}} \quad (15)$$

This formula shows that we are interested in maximizing the difference between the sum of the cells of the actual contingency table and the sum of the expected cells under randomness. Standardized measures differs from their adjusted counterpart because of the denominator, i.e. the standard deviation of the sums of the cells. Indeed, SMI_q and SVI_q measure the number of standard deviations MI_q and VI_q are from their mean.

There are some notable special cases for particular choices of q . Indeed, our generalized standardization of IT measures allows us to generalize also the standardization of pair-counting measures such as the Rand index. To see this, let us define the Standardized Rand Index (SRI): $SRI \triangleq \frac{RI - E[RI]}{\sqrt{\text{Var}(RI)}}$ and recall that the standardized G -statistic is defined as $SG \triangleq \frac{G - E[G]}{\sqrt{\text{Var}(G)}}$ (Romano et al., 2014):

Corollary 2 *It holds true that:*

- i) $\lim_{q \rightarrow 1} SMI_q = \lim_{q \rightarrow 1} SVI_q = SMI = SVI = SG$ with Shannon entropy;
- ii) $SMI_2 = SVI_2 = SRI$.

Computational complexity: The computational complexity of SMI_q is dominated by computation of the second moment of the sum of the cells defined in Eq. (13):

Proposition 3 *The computational complexity of SMI_q is $O(N^3 c \cdot \max\{c, r\})$.*

Note that the complexity is quadratic in c and linear in r . This happens because of the way we decided to condition the probabilities in Eq. (13) in the proof of Lemma 3. With different conditions, it is possible to obtain a formula symmetric to Eq. (13) with complexity $O(N^3 r \cdot \max\{r, c\})$ (Romano et al., 2014).

Statistical inference: All IT measures computed on *partitions* can be seen as estimators of their true value computed using the random *variables* associated to the partitions U and V . Therefore, SMI_q can be used as non-parametric independence test for MI_q . We formalize this with the following proposition:

Proposition 4 *The p -value associated to the test for independence between U and V using $\text{MI}_q(U, V)$ is smaller than:*

$$\frac{1}{1 + (\text{SMI}_q(U, V))^2}.$$

For example, if SMI_q is equal to 4.46 the associated p -value is smaller than 0.05. Neural time series data is often analyzed making use of the Shannon MI (e.g. see Chapter 29 in Cohen (2014)). It is common practice to test the independence of two time series by computing SMI via Monte Carlo permutations, sampling from the space of $N!$ cardinality. Our SMI_q can be effectively and efficiently used in this application because it is exact and obtains $O(N^3 r \cdot \max\{r, c\})$ complexity.

5.2 Experiments on Selection Bias

In this section, we evaluate the performance of standardized measures on selection bias correction when partitions U are generated at random and independently from the reference partition V . This hypothesis has been employed in previous published research to study selection bias (White and Liu, 1994; Frank and Witten, 1998; Dobra and Gehrké, 2001; Shih, 2004; Hothorn et al., 2006; Romano et al., 2014). In particular, we experimentally demonstrate that NMI_q is biased towards the selection of partitions U with more clusters at any q . Therefore, in this scenario it is beneficial to perform standardization. Mind though that the choice of whether performing standardization or not is application dependent (Romano et al., 2015). For example, it has been argued that in some cases the selection of clustering solutions should be biased towards clusterings with the same number of clusters as in the reference (Amelio and Pizzati, 2015). In this section we aim to show the effects of selection bias when clusterings are independent and that standardization helps in reducing it. Moreover, we will see in Section 5.3 that it is particularly important to correct for selection bias when the number of objects N is small.

Given a reference partition V on $N = 100$ objects with $c = 4$ sets, we generate a pool of random partitions U with r ranging from 2 to 10 sets. Then, we use $\text{NMI}_q(U, V)$ to select the closest partition to the reference V . The plot at the bottom of Figure 10 shows the probability of selection of a partition U with r sets using NMI_q computed on 5000 simulations. We do not expect any partition to be the best given that they are all generated at random: *i.e., the plot is expected to be flat if a measure is unbiased*. Nonetheless, we see that there is a clear bias towards partitions with 10 sets if we use NMI_q with q respectively equal to 1.001, 2, or 3. We can see that the use of the adjusted measures such as AMI_q helps in decreasing this bias, in particular when $q = 2$. On this experiment when $q = 2$, baseline adjustment seems to be effective in decreasing the selection bias because the variance of $\text{AMI}_2 = \text{ARI}$ is almost constant. However for all q , using SMI_q we obtain close to uniform probability of selection of each random partition U .

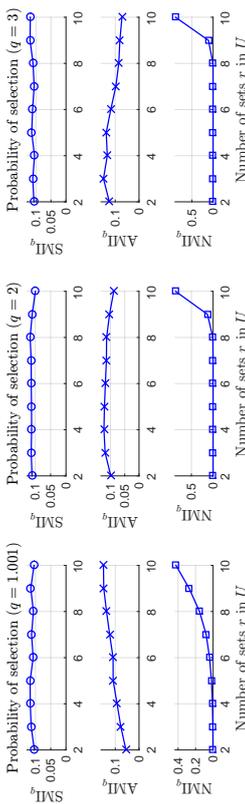


Figure 10: Selection bias towards random partitions U with different number of sets r when compared to a reference V . The probability of selection should be uniform when partitions are random. Using SMI_q we achieve close to uniform probability of selection for q equal to 1.001, 2 and 3 respectively.

5.3 Large Number of Objects

It is likely to expect that the variance of generalized IT measures decreases when partitions are generated on a large number of objects N . Here we prove a general result about measures of the family \mathcal{N}_ϕ .

Lemma 4 *If $S(U, V) \in \mathcal{N}_\phi$, then $\lim_{N \rightarrow +\infty} \text{Var}(S(U, V)) = 0$.*

Given that generalized IT measures belong in the family \mathcal{N}_ϕ , we can prove the following:

Theorem 7 *It holds true that:*

$$\lim_{N \rightarrow +\infty} \text{Var}(H_q(U, V)) = \lim_{N \rightarrow +\infty} \text{Var}(\text{MI}_q(U, V)) = \lim_{N \rightarrow +\infty} \text{Var}(\text{VI}_q(U, V)) = 0 \quad (16)$$

Therefore, SMI_q attains very large values when N is large. In practice of course, N is finite, so the use of SMI_q is beneficial. However, it is less important to correct for selection bias if the number of objects N is big with regards to the number of cells in the contingency table in Table 2: i.e., when $\frac{N}{rc}$ is large. Indeed, when the number of objects is large AMI_q might be sufficient to avoid selection bias and any test for independence between partitions has high power. In this scenario, SMI_q is not needed and AMI_q might be preferred as it can be computed more efficiently.

6. Conclusion

In this paper, we computed the exact expected value and variance of measures of the family \mathcal{L}_ϕ , which contains generalized IT measures. We also showed how the expected value for measures $S \in \mathcal{N}_\phi$ can be computed for large N . Using these statistics, we proposed AMI_q and SMI_q to adjust generalized IT measures both for baseline and for selection bias. AMI_q is a further generalization of well known measures for clustering comparisons such as ARI and AMI. This analysis allowed us to provide guidelines for their best application in different scenarios. In particular ARI might be used as external validation index when the reference

clustering shows big equal sized clusters. AMI can be used when the reference clustering is unbalanced and there exist small clusters. The standardized SMI_q can instead be used to correct for selection bias among many possible candidate clustering solutions when the number of objects is small. Furthermore, it can also be used to test the independence between two partitions. All code has been made available online¹.

Acknowledgments

James Bailey's work was supported by an Australian Research Council Future Fellowship. Experiments were carried out on Amazon cloud supported by AWS in Education Grant Award.

Appendix A. Theorem Proofs

Proposition 1 (Simowicz, 2007) *When $q = 2$ the generalized variation of information, the Mfrkin index, and the Rand index are linearly related: $VI_2(U, V) = \frac{1}{N^2}MK(U, V) = \frac{N-1}{N}(1 - RI(U, V))$.*

Proof

$$\begin{aligned} VI_q(U, V) &= 2H_q(U, V) - H_q(U) - H_q(V) \\ &= \frac{2}{q-1} \left(1 - \sum_{i=1}^r \sum_{j=1}^c \binom{n_{ij}}{N} \right)^q - \frac{1}{q-1} \left(1 - \sum_{i=1}^r \binom{a_i}{N} \right)^q - \frac{1}{q-1} \left(1 - \sum_{j=1}^c \binom{b_j}{N} \right)^q \\ &= \frac{1}{(q-1)N^q} \left(\sum_{i=1}^r a_i^q + \sum_{j=1}^c b_j^q - 2 \sum_{i=1}^r \sum_{j=1}^c n_{ij}^q \right) \end{aligned}$$

When $q = 2$, $VI_2(U, V) = \frac{1}{N^2}(\sum_i a_i^2 + \sum_j b_j^2 - 2 \sum_{i,j} n_{ij}^2) = \frac{1}{N^2}MK(U, V) = \frac{N-1}{N}(1 - RI(U, V))$. ■

Lemma 1 *If $S(U, V) \in \mathcal{L}_\phi$, when partitions U and V are random:*

$$E[S(U, V)] = \alpha + \beta \sum_{i,j} E[\phi_{ij}(n_{ij})] \quad \text{where} \quad E[\phi_{ij}(n_{ij})] \quad \text{is} \quad (9)$$

$$\sum_{n_{ij}=\max\{0, a_i+b_j-N\}}^{\min\{a_i, b_j\}} \phi_{ij}(n_{ij}) \frac{a_i! b_j! (N - a_i)! (N - b_j)!}{N! n_{ij}! (a_i - n_{ij})! (b_j - n_{ij})! (N - a_i - b_j + n_{ij})!} \quad (10)$$

Proof The expected value of $S(U, V)$ according to the hypergeometric model of randomness is $E[S(U, V)] = \sum_{\mathcal{M}} S(\mathcal{M}) P(\mathcal{M})$ where \mathcal{M} is a contingency table generated via permutations. This is reduced to $E[S(U, V)] = \sum_{\mathcal{M}} (\alpha + \beta \sum_{i,j} \phi_{ij}(n_{ij})) P(\mathcal{M}) = \alpha + \beta \sum_{\mathcal{M}} \sum_{i,j} \phi_{ij}(n_{ij}) P(\mathcal{M})$. Because of linearity of the expected value, it is possible to swap the summation over \mathcal{M} and the one over cells obtaining $\alpha + \beta \sum_{i,j} \sum_{n_{i,j}} \phi_{ij}(n_{i,j}) P(n_{i,j}) = \alpha + \beta \sum_{i,j} E[\phi_{ij}(n_{i,j})]$ where $n_{i,j}$ is a hypergeometric distribution with the marginals a_i , b_j and N as parameters, i.e. $n_{i,j} \sim \text{Hyp}(a_i, b_j, N)$. ■

Theorem 1 *When the partitions U and V are random:*

- i) $E[H_q(U, V)] = \frac{1}{q-1} \left(1 - \frac{1}{N^q} \sum_{i,j} E[n_{ij}^q] \right)$ with $E[n_{ij}^q] = n_{ij}^q$;
- ii) $E[MI_q(U, V)] = H_q(U) + H_q(V) - E[H_q(U, V)]$;
- iii) $E[VI_q(U, V)] = 2E[H_q(U, V)] - H_q(U) - H_q(V)$.

Proof The results easily follow from Lemma 1 and the hypothesis of fixed marginals. ■

¹ <https://sites.google.com/site/adjgent/>

Theorem 2 Using $E[n_{ij}^q]$ in Eq. (10) with $\phi_{ij}(n_{ij}) = n_{ij}^q$, the adjustments for chance for $\text{MI}_q(U, V)$ and $\text{VI}_q(U, V)$ are:

$$\text{AMI}_q(U, V) = \text{AVI}_q(U, V) = \frac{\sum_{i,j} n_{ij}^q - \sum_{i,j} E[n_{ij}^q]}{\frac{1}{2} \left(\sum_i a_i^q + \sum_j b_j^q \right) - \sum_{i,j} E[n_{ij}^q]} \quad (12)$$

Proof The using the upper bound $\frac{1}{2}(H_q(U) + H_q(V))$ to MI_q , AMI_q and AVI_q are equivalent. Therefore we compute AVI_q . The denominator is equal to $E[\text{VI}_q] = \frac{2}{(q-1)N^q} \left(\frac{1}{2} \sum_i a_i^q + \sum_j b_j^q \right) - \sum_{i,j} E[n_{ij}^q]$. The numerator is instead $\frac{2}{(q-1)N^q} \left(\sum_{i,j} n_{ij}^q - \sum_{i,j} E[n_{ij}^q] \right)$. ■

Corollary 1 It holds true that:

- i) $\lim_{q \rightarrow 1} \text{AMI}_q = \lim_{q \rightarrow 1} \text{AVI}_q = \text{AMI} = \text{AVI}$ with Shannon entropy;
- ii) $\text{AMI}_2 = \text{AVI}_2 = \text{ARI}$.

Proof Point i) follows from the limit of the q -entropy when $q \rightarrow 1$. Point ii) follows from:

$$\text{AVI}_2 = \frac{E[\text{VI}_2] - \text{VI}_2}{E[\text{VI}_2] - \min \text{VI}_2} = \frac{\frac{N-1}{N}(\text{RI} - E[\text{RI}])}{\frac{N-1}{N}(\max \text{RI} - E[\text{RI}])} = \text{ARI}$$

■

Proposition 2 The computational complexity of AMI_q is $O(N \cdot \max\{r, c\})$.

Proof The computation of $P(n_{ij})$ where n_{ij} is a hypergeometric distribution $\text{Hyp}(a_i, b_j, N)$ is linear in N . However, the computation of the expected value $E[n_{ij}^q] = \sum_{n_{ij}} n_{ij}^q P(n_{ij})$ can exploit the fact that $P(n_{ij})$ are computed iteratively: $P(n_{ij}+1) = P(n_{ij}) \frac{(a_i - n_{ij})(b_j - n_{ij})}{(n_{ij}+1)(N - a_i - b_j + n_{ij} + 1)}$. We compute $P(n_{ij})$ only for $\max\{0, a_i + b_j - N\}$. In both cases $P(n_{ij})$ can be computed in $O(\max\{a_i, b_j\})$. We can compute all other probabilities iteratively as shown above in constant time. Therefore:

$$\begin{aligned} \sum_{i=1}^r \sum_{j=1}^c \left(O(\max\{a_i, b_j\}) + \sum_{n_{ij}=0}^{\min\{a_i, b_j\}} O(1) \right) &= \sum_{i=1}^r \sum_{j=1}^c O(\max\{a_i, b_j\}) = \sum_{i=1}^r O(\max\{ca_i, N\}) \\ &= O(\max\{cN, rN\}) = O(N \cdot \max\{c, r\}) \end{aligned}$$

■

Lemma 2 If $S(U, V) \in \mathcal{N}_\phi$, then $\lim_{N \rightarrow +\infty} E[S(U, V)] = \phi\left(\frac{a_1}{N}, \dots, \frac{a_r}{N}, \frac{b_1}{N}, \dots, \frac{b_c}{N}\right)$.

Proof $S(U, V)$ can be written as $\phi\left(\frac{n_{11}}{N}, \dots, \frac{n_{1r}}{N}, \dots, \frac{n_{rc}}{N}\right)$. Let $\mathbf{X} = (X_1, \dots, X_{rc}) = \left(\frac{n_{11}}{N}, \dots, \frac{n_{1r}}{N}, \dots, \frac{n_{rc}}{N}\right)$ be a vector of rc random variables where n_{ij} is a hypergeometric distribution with the marginals as parameters: a_i, b_j and N . The expected value of $\frac{n_{ij}}{N}$ is $E\left[\frac{n_{ij}}{N}\right] = \frac{1}{N} \frac{a_i b_j}{N}$. Let $\boldsymbol{\mu} = (\mu_1, \dots, \mu_{rc}) = (E[X_1], \dots, E[X_{rc}]) = \left(\frac{a_1 b_1}{N}, \dots, \frac{a_1 b_r}{N}, \dots, \frac{a_r b_c}{N}\right)$ be the vector of the expected values. The Taylor approximation of $S(U, V) = \phi(\mathbf{X})$ around $\boldsymbol{\mu}$ is:

$$\phi(\mathbf{X}) \simeq \phi(\boldsymbol{\mu}) + \sum_{t=1}^{rc} (X_t - \mu_t) \frac{\partial \phi}{\partial X_t} + \frac{1}{2} \sum_{t=1}^{rc} \sum_{s=1}^{rc} (X_t - \mu_t)(X_s - \mu_s) \frac{\partial^2 \phi}{\partial X_t \partial X_s} + \dots$$

Its expected value is (see Section 4.3 of (Ang and Tang, 2006)):

$$E[\phi(\mathbf{X})] \simeq \phi(\boldsymbol{\mu}) + \frac{1}{2} \sum_{t=1}^{rc} \sum_{s=1}^{rc} \text{Cov}(X_t, X_s) \frac{\partial^2 \phi}{\partial X_t \partial X_s} + \dots$$

We just analyse the second order remainder given that it dominates the higher order ones. Using the Cauchy-Schwartz inequality we have that $|\text{Cov}(X_t, X_s)| \leq \sqrt{\text{Var}(X_t)\text{Var}(X_s)}$. Each X_t and X_s is equal to $\frac{n_{ij}}{N}$ for some indexes i and j . The variance of each X_t and X_s is therefore equal to $\text{Var}\left(\frac{n_{ij}}{N}\right) = \frac{1}{N^2} \frac{a_i b_j}{N} \frac{N - a_i - b_j}{N - 1}$. When the number of records is large also the marginals increase: $N \rightarrow +\infty \Rightarrow a_i \rightarrow +\infty$, and $b_j \rightarrow +\infty \forall i, j$. However, because of the permutation model, all the fractions $\frac{a_i}{N}$ and $\frac{b_j}{N}$ stay constant $\forall i, j$. Therefore, because $\boldsymbol{\mu}$ is constant. However, at the limit of large N , the variance of $\frac{n_{ij}}{N}$ tends to 0: $\text{Var}\left(\frac{n_{ij}}{N}\right) = \frac{1}{N} \frac{a_i b_j}{N} \left(1 - \frac{a_i}{N}\right) \left(1 + \frac{1}{N-1} - \frac{b_j}{N}\right) \rightarrow 0$. Therefore, at large N :

$$E[\phi(\mathbf{X})] \simeq \phi(\boldsymbol{\mu}) = \phi\left(\frac{a_1}{N}, \frac{b_1}{N}, \dots, \frac{a_r}{N}, \frac{b_c}{N}\right)$$

■

Theorem 3 It holds true that:

- i) $\lim_{N \rightarrow +\infty} E[H_q(U, V)] = H_q(U) + H_q(V) - (q-1)H_q(U)H_q(V)$;
- ii) $\lim_{N \rightarrow +\infty} E[\text{MI}_q(U, V)] = (q-1)H_q(U)H_q(V)$;
- iii) $\lim_{N \rightarrow +\infty} E[\text{VI}_q(U, V)] = H_q(U) + H_q(V) - 2(q-1)H_q(U)H_q(V)$.

Proof $E[H_q(U, V)] = \frac{1}{q-1} \left(1 - \sum_{i,j} E\left[\left(\frac{n_{ij}}{N}\right)^q\right]\right)$ and according to Lemma 2 for large N : $E[H_q(U, V)] \simeq \frac{1}{q-1} \left(1 - \sum_{i,j} \left(\frac{a_i b_j}{N^q}\right)^q\right) = \frac{1}{q-1} \left(1 - \sum_i \left(\frac{a_i}{N}\right)^q \sum_j \left(\frac{b_j}{N}\right)^q\right)$. If we add an subtract

$1 - \sum_i \left(\frac{a_i}{N}\right)^q - \sum_j \left(\frac{b_j}{N}\right)^q$ in the parenthesis above:

$$\begin{aligned} E[H_q(U, V)] &\simeq \frac{1}{q-1} \left(1 - \sum_i \left(\frac{a_i}{N}\right)^q \sum_j \left(\frac{b_j}{N}\right)^q\right. \\ &\quad \left.+ 1 - \sum_i \left(\frac{a_i}{N}\right)^q - \sum_j \left(\frac{b_j}{N}\right)^q\right) \\ &\quad - 1 + \sum_i \left(\frac{a_i}{N}\right)^q + \sum_j \left(\frac{b_j}{N}\right)^q \\ &= \frac{1}{q-1} \left(1 - \sum_i \left(\frac{a_i}{N}\right)^q\right) + \frac{1}{q-1} \left(1 - \sum_j \left(\frac{b_j}{N}\right)^q\right) \\ &\quad + \frac{1}{q-1} \left(-1 - \sum_i \left(\frac{a_i}{N}\right)^q \sum_j \left(\frac{b_j}{N}\right)^q + \sum_i \left(\frac{a_i}{N}\right)^q + \sum_j \left(\frac{b_j}{N}\right)^q\right) \\ &= H_q(U) + H_q(V) + \frac{1}{q-1} \left(\left(1 - \sum_i \left(\frac{a_i}{N}\right)^q\right) \left(\sum_j \left(\frac{b_j}{N}\right)^q\right)\right) \\ &= H_q(U) + H_q(V) - (q-1)H_q(U)H_q(V) \end{aligned}$$

Point *ii*) and *iii*) follow from Equations (6) and (7). \blacksquare

Lemma 3 *If $S(U, V) \in \mathcal{L}_{\phi^*}$, when partitions U and V are random:*

$$\text{Var}(S(U, V)) = \beta^2 \left(E \left[\left(\sum_{i,j} \phi_{ij}(n_{ij}) \right)^2 \right] - \left(\sum_{i,j} E[\phi_{ij}(n_{ij})] \right)^2 \right),$$

where

$$E \left[\left(\sum_{i,j} \phi_{ij}(n_{ij}) \right)^2 \right]$$

is equal to

$$\begin{aligned} &\sum_{i,j} \sum_{n_{ij}} \phi(n_{ij}) P(n_{ij}) \cdot \left[\phi_{ij}(n_{ij}) + \sum_{i' \neq i} \sum_{j' \neq j} \phi_{i'j'}(\tilde{n}_{i'j'}) P(\tilde{n}_{i'j'}) + \right. \\ &\quad \left. + \sum_{j' \neq j} \sum_{i' \neq i} P(\tilde{n}_{i'j'}) \left(\phi_{ij'}(\tilde{n}_{i'j'}) + \sum_{i'' \neq i} \sum_{j'' \neq j} \phi_{i''j''}(\tilde{\tilde{n}}_{i''j''}) P(\tilde{\tilde{n}}_{i''j''}) \right) \right] \end{aligned} \quad (13)$$

with $n_{ij} \sim \text{HYP}(a_i, b_j, N)$, $\tilde{n}_{i'j'} \sim \text{HYP}(b_j - n_{ij}, a_i, N - a_i)$, $\tilde{\tilde{n}}_{i''j''} \sim \text{HYP}(a_i - n_{ij}, b_j, N - b_j)$, $\tilde{n}_{i'j'} \sim \text{HYP}(a_i, b_j - \tilde{n}_{i'j'}, N - a_i)$ hypergeometric random variables.

Proof The proof follows Theorem 1 proof in Romano et al. (2014). Using the properties of the variance we can show that $\text{Var}(S(U, V)) = \beta^2 \text{Var}(\sum_{i,j} \phi_{ij}(n_{ij})) = \beta^2 \left(E[(\sum_{i,j} \phi_{ij}(n_{ij}))^2] -$

$(\sum_{i,j} E[\phi_{ij}(n_{ij})])^2$). $(E[\sum_{i,j} \phi_{ij}(n_{ij})])^2 = (\sum_{i,j} E[\phi_{ij}(n_{ij})])^2$ can be computed using Eq. (10). The first term in the sum is instead:

$$E[(\sum_{i,j} \phi_{ij}(n_{ij}))^2] = \sum_{i,j} \sum_{i',j'} E[\phi_{ij}(n_{ij})\phi_{i'j'}(n_{i'j'})] = \sum_{i,j} \sum_{i',j'} \sum_{n_{ij}} \sum_{n_{i'j'}} \phi_{ij}(n_{ij})\phi_{i'j'}(n_{i'j'}) P(n_{ij}, n_{i'j'})$$

We cannot find the exact form of the joint probability $P(n_{ij}, n_{i'j'})$ thus we rewrite it as $P(n_{ij})P(n_{i'j'}|n_{ij}) = P(n_{ij})P(\tilde{n}_{i'j'})$. The random variable $n_{i'j'}$ is a hypergeometric distribution that simulates the experiment of sampling without replacement the a_i objects in the set u_i from a total of N objects. Sampling one of the b_j objects from v_j is defined as a success: $n_{i'j'} \sim \text{HYP}(a_i, b_j, N)$. The random variable $\tilde{n}_{i'j'}$ has a different distribution depending on the possible combinations of indexes i, i', j, j' . Thus $E[(\sum_{i,j} \phi_{ij}(n_{ij}))^2]$ is equal to:

$$\begin{aligned} &\sum_{i,j} \sum_{n_{ij}} \sum_{i',j'} \sum_{n_{i'j'}} \phi_{ij}(n_{ij})\phi_{i'j'}(n_{i'j'}) P(n_{ij}, n_{i'j'}) = \sum_{i,j} \sum_{n_{ij}} \phi_{ij}(n_{ij}) P(n_{ij}) \sum_{i',j'} \sum_{n_{i'j'}} \phi_{i'j'}(\tilde{n}_{i'j'}) P(\tilde{n}_{i'j'}) \\ &\quad \text{which, by taking care of all possible combinations of } i, i', j, j', \text{ is equal to :} \\ &\sum_{i,j} \sum_{n_{ij}} \phi_{ij}(n_{ij}) P(n_{ij}) \cdot \left[\sum_{i'=i, j'=j} \sum_{\tilde{n}_{i'j'}} \phi_{ij}(\tilde{n}_{i'j'}) P(\tilde{n}_{i'j'}) \right. \\ &\quad \left. + \sum_{i' \neq i, j'=j} \sum_{\tilde{n}_{i'j'}} \phi_{ij}(\tilde{n}_{i'j'}) P(\tilde{n}_{i'j'}) \right] \end{aligned} \quad (17)$$

$$\begin{aligned} &\quad + \sum_{i'=i, j' \neq j} \sum_{\tilde{n}_{i'j'}} \phi_{ij}(\tilde{n}_{i'j'}) P(\tilde{n}_{i'j'}) \\ &\quad + \sum_{i' \neq i, j' \neq j} \sum_{\tilde{n}_{i'j'}} \phi_{ij}(\tilde{n}_{i'j'}) P(\tilde{n}_{i'j'}) \end{aligned} \quad (18)$$

Case 1: $i' = i \wedge j' = j$
 $P(\tilde{n}_{i'j'}) = 1$ if and only if $\tilde{n}_{i'j'} = n_{ij}$ and 0 otherwise. This case produces the first term $\phi_{ij}(n_{ij})$ enclosed in square brackets.

Case 2: $i' = i \wedge j' \neq j$
In this case, the possible successes are the objects from the set v_j . We have already sampled n_{ij} objects and we are sampling from the whole set of objects excluding the set v_j . Thus, $\tilde{n}_{i'j'} \sim \text{HYP}(a_i - n_{ij}, b_j, N - b_j)$.

Case 3: $i' \neq i \wedge j' = j$
This case is symmetric to the previous one where $a_{i'}$ is now the possible number of successes. Therefore $\tilde{n}_{i'j'} \sim \text{HYP}(b_j - n_{ij}, a_i, N - a_i)$.

Case 4: $i' \neq i \wedge j' \neq j$
In order compute $P(\tilde{n}_{i'j'})$, we have to impose a further condition:

$$P(\tilde{n}_{i'j'}) = \sum_{\tilde{n}_{i'j'}} P(\tilde{n}_{i'j'} | \tilde{n}_{i'j'}) P(\tilde{n}_{i'j'}) = \sum_{\tilde{n}_{i'j'}} P(\tilde{\tilde{n}}_{i'j'}) P(\tilde{\tilde{n}}_{i'j'})$$

We are considering sampling the $a_{i'}$ objects in $u_{i'}$ from the whole set of objects excluding the a_i objects from u_i . Just knowing that n_{ij} objects have already been sampled from u_i does not allow us to know how many objects from $v_{i'}$ have also been sampled. If we know that $n_{i'j'}$ is the number of objects sampled from $v_{j'}$, we know there are $b_{j'} - n_{i'j'}$ possible successes and thus $\tilde{n}_{i'j'} | \tilde{n}_{i'j'} \sim \text{Hyp}(a_{i'}, b_{j'} - \tilde{n}_{i'j'}, N - a_i)$. So the last two terms in Eq. (18) can be put together:

$$\begin{aligned} & \sum_{j' \neq i, j' \neq j} \sum_{\tilde{n}_{i'j'}} \phi_{ij'}(\tilde{n}_{i'j'}) P(\tilde{n}_{i'j'}) + \sum_{i' \neq i, j' \neq j} \sum_{\tilde{n}_{i'j'}} \phi_{ij'}(\tilde{n}_{i'j'}) P(\tilde{n}_{i'j'}) \\ &= \sum_{j' \neq j} \sum_{\tilde{n}_{i'j'}} \phi_{ij'}(\tilde{n}_{i'j'}) P(\tilde{n}_{i'j'}) + \sum_{i' \neq i, j' \neq j} \sum_{\tilde{n}_{i'j'}} \phi_{ij'}(\tilde{n}_{i'j'}) P(\tilde{n}_{i'j'}) P(\tilde{n}_{i'j'}) \\ &= \sum_{j' \neq j} \sum_{\tilde{n}_{i'j'}} \phi_{ij'}(\tilde{n}_{i'j'}) P(\tilde{n}_{i'j'}) + \sum_{i' \neq i, j' \neq j} \sum_{\tilde{n}_{i'j'}} \phi_{ij'}(\tilde{n}_{i'j'}) P(\tilde{n}_{i'j'}) P(\tilde{n}_{i'j'}) \\ &= \sum_{j' \neq j} \sum_{\tilde{n}_{i'j'}} P(\tilde{n}_{i'j'}) \phi_{ij'}(\tilde{n}_{i'j'}) + \sum_{i' \neq i, j' \neq j} \sum_{\tilde{n}_{i'j'}} P(\tilde{n}_{i'j'}) \sum_{i' \neq i} \phi_{ij'}(\tilde{n}_{i'j'}) P(\tilde{n}_{i'j'}) \\ &= \sum_{j' \neq j} \sum_{\tilde{n}_{i'j'}} P(\tilde{n}_{i'j'}) \left(\phi_{ij'}(\tilde{n}_{i'j'}) + \sum_{i' \neq i} \sum_{\tilde{n}_{i'j'}} \phi_{ij'}(\tilde{n}_{i'j'}) P(\tilde{n}_{i'j'}) \right) \end{aligned}$$

By putting everything together we get that $E[(\sum_{ij} \phi_{ij}(n_{ij}))^2]$ is equal to:

$$\begin{aligned} & \sum_{ij} \sum_{i'j'} \phi(n_{ij}) P(n_{ij}) \cdot \left[\phi_{ij}(n_{ij}) + \sum_{i' \neq i} \sum_{\tilde{n}_{i'j'}} \phi_{ij}(\tilde{n}_{i'j'}) P(\tilde{n}_{i'j'}) + \right. \\ & \quad \left. + \sum_{j' \neq j} \sum_{\tilde{n}_{i'j'}} P(\tilde{n}_{i'j'}) \left(\phi_{ij'}(\tilde{n}_{i'j'}) + \sum_{i' \neq i} \sum_{\tilde{n}_{i'j'}} \phi_{ij'}(\tilde{n}_{i'j'}) P(\tilde{n}_{i'j'}) \right) \right] \end{aligned}$$

Theorem 5 Using Eqs. (10) and (13) with $\phi_{ij}(\cdot) = (\cdot)^q$, when the partitions U and V are random:

- i) $\text{Var}(H_q(U, V)) = \frac{1}{(q-1)^2 N^q} \left(E[(\sum_{ij} n_{ij}^q)^2] - (\sum_{ij} E[n_{ij}^q])^2 \right)$;
- ii) $\text{Var}(\text{ML}_q(U, V)) = \text{Var}(H_q(U, V))$
- iii) $\text{Var}(\text{VI}_q(U, V)) = 4\text{Var}(H_q(U, V))$

Proof The results follow from Lemma 3, the hypothesis of fixed marginals and properties of the variance. ■

Theorem 6 Using Eqs. (10) and (13) with $\phi_{ij}(\cdot) = (\cdot)^q$, the standardized $\text{ML}_q(U, V)$ and the standardized $\text{VI}_q(U, V)$ are:

$$\text{SMI}_q(U, V) = \text{SVI}_q(U, V) = \frac{\sum_{ij} n_{ij}^q - \sum_{ij} E[n_{ij}^q]}{\sqrt{E[(\sum_{ij} n_{ij}^q)^2] - (\sum_{ij} E[n_{ij}^q])^2}} \quad (15)$$

Proof

For SMI_q , the numerator is equal to $H_q(U, V) - E[H_q(U, V)] = \frac{1}{(q-1)N^q} \left(\sum_{ij} n_{ij}^q - \sum_{i,j} E[n_{ij}^q] \right)$. According Theorem 5, the denominator is instead:

$$\sqrt{\text{Var}(\text{ML}_q(U, V))} = \sqrt{\text{Var}(H_q(U, V))} = \frac{1}{(q-1)N^q} \sqrt{E[(\sum_{ij} n_{ij}^q)^2] - (E[\sum_{ij} n_{ij}^q])^2}.$$

For SVI_q , the numerator is equal to $2H_q(U, V) - 2E[H_q(U, V)] = \frac{2}{(q-1)N^q} \left(\sum_{ij} n_{ij}^q - \sum_{i,j} E[n_{ij}^q] \right)$. According Theorem 5, the denominator is instead:

$$\sqrt{\text{Var}(\text{VI}_q(U, V))} = \sqrt{4\text{Var}(H_q(U, V))} = \frac{2}{(q-1)N^q} \sqrt{E[(\sum_{ij} n_{ij}^q)^2] - (E[\sum_{ij} n_{ij}^q])^2}.$$

Therefore, SMI_q and SVI_q are equivalent. ■

Corollary 2 It holds true that:

- i) $\lim_{q \rightarrow 1} \text{SMI}_q = \lim_{q \rightarrow 1} \text{SVI}_q = \text{SMI} = \text{SVI} = \text{SG}$ with Shannon entropy;
- ii) $\text{SMI}_2 = \text{SVI}_2 = \text{SRI}$.

Proof Point i) follows from the limit of the q -entropy when $q \rightarrow 1$ and the linear relation of G -statistic to MI: $G = 2N\text{MI}$. Point ii) follows from:

$$\text{SVI}_2 = \frac{E[\text{VI}_2] - \text{VI}_2}{\sqrt{\text{Var}(\text{VI}_2)}} = \frac{N-1}{N} \frac{(\text{RI} - E[\text{RI}])}{\sqrt{\text{Var}(\text{RI})}} = \text{SRI}$$

Proposition 3 The computational complexity of SMI_q is $O(N^3 c \cdot \max\{c, r\})$.

Proof Each summation in Eq. (13) can be bounded above by the maximum value of the cell marginals and each sum can be done in constant time. The last summation in Eq. (13) is:

$$\begin{aligned} & \sum_{j'=1}^c \sum_{\tilde{n}_{i'j'=0}} \sum_{i'=1}^r \sum_{\tilde{n}_{i'j'=0}} \sum_{i'=1}^c \max_{\{a_{i'}, b_{j'}\}} \sum_{i'=1}^r \sum_{\tilde{n}_{i'j'=0}} O(\max\{a_{i'}, b_{j'}\}) \\ &= \sum_{j'=1}^c \sum_{\tilde{n}_{i'j'=0}} \sum_{i'=1}^r \sum_{\tilde{n}_{i'j'=0}} O(\max\{N, r b_{j'}\}) \\ &= \sum_{j'=1}^c O(\max\{a_i, b_{j'}\}) \\ &= \sum_{j'=1}^c O(\max\{a_i, a_i r b_{j'}, b_{j'} N, r b_{j'}^2\}) \\ &= O(\max\{c a_i, N, a_i r N, r N^2\}) \end{aligned}$$

The above term is thus the computational complexity of the inner loop. Using the same machinery one can prove that:

$$\sum_{j=1}^c \sum_{i=1}^r \max_{n_{ij}=0}^{\max\{a_i, b_j\}} O(\max\{a_i N, a_j r N, r N^2\}) = O(\max\{c^2 N^3, r c N^3\}) = O(N^3 c \cdot \max\{c, r\})$$

■

Proposition 4 *The p -value associated to the test for independence between U and V using $\text{MI}_q(U, V)$ is smaller than: $\frac{1}{1 + (\text{SMI}_q(U, V))^2}$.*

Proof Let MI_q^0 be the random variable under the null hypothesis of independence between partitions associated to the test statistic $\text{MI}_q(U, V)$. The p -value is defined as:

$$\begin{aligned} p\text{-value} &= P\left(\text{MI}_q^0 \geq \text{MI}_q(U, V)\right) = P\left(\text{MI}_q^0 - E[\text{MI}_q(U, V)] \geq \text{MI}_q(U, V) - E[\text{MI}_q(U, V)]\right) \\ &= P\left(\frac{\text{MI}_q^0 - E[\text{MI}_q(U, V)]}{\sqrt{\text{Var}(\text{MI}_q(U, V))}} \geq \frac{\text{MI}_q(U, V) - E[\text{MI}_q(U, V)]}{\sqrt{\text{Var}(\text{MI}_q(U, V))}}\right) \\ &= P\left(\frac{\text{MI}_q^0 - E[\text{MI}_q(U, V)]}{\sqrt{\text{Var}(\text{MI}_q(U, V))}} \geq \text{SMI}_q(U, V)\right) \end{aligned}$$

Let Z be the standardized random variable $\frac{\text{MI}_q^0 - E[\text{MI}_q(U, V)]}{\sqrt{\text{Var}(\text{MI}_q(U, V))}}$, then using the one side Chebyshev's inequality also known as the Cantelli's inequality (Ross, 2012):

$$p\text{-value} = P(Z \geq \text{SMI}_q(U, V)) < \frac{1}{1 + (\text{SMI}_q(U, V))^2}$$

■

Lemma 4 *If $S(U, V) \in \mathcal{N}_{\phi}$, then $\lim_{N \rightarrow +\infty} \text{Var}(S(U, V)) = 0$.*

Proof Let $\mathbf{X} = (X_1, \dots, X_{rc}) = (\frac{n_{11}}{N}, \dots, \frac{n_{1c}}{N}, \dots, \frac{n_{r1}}{N}, \dots, \frac{n_{rc}}{N})$ be a vector of rc random variables where n_{ij} is a hypergeometric distribution with the marginals as parameters: a_i , b_j and N . Using the Taylor approximation (Ang and Tang, 2006) of $S(U, V) = \phi(\mathbf{X})$, it is possible to show that:

$$\text{Var}(\phi(\mathbf{X})) \simeq \sum_{i=1}^{rc} \sum_{s=1}^{rc} \text{Cov}(X_i, X_s) \frac{\partial \phi}{\partial X_i} \frac{\partial \phi}{\partial X_s} + \dots$$

Using the Cauchy-Schwarz inequality we have that $|\text{Cov}(X_i, X_s)| \leq \sqrt{\text{Var}(X_i) \text{Var}(X_s)}$. Each X_i and X_s is equal to $\frac{n_{ij}}{N}$ for some indexes i and j . The variance of each X_i and X_s is

therefore equal to $\text{Var}\left(\frac{n_{ij}}{N}\right) = \frac{1}{N^2} \frac{a_i b_j}{N} \frac{N - a_i}{N} \frac{N - b_j}{N - 1}$. When the number of records is large also the marginals increase: $N \rightarrow +\infty \Rightarrow a_i \rightarrow +\infty$, and $b_j \rightarrow +\infty \forall i, j$. However because of the permutation model, all the fractions $\frac{a_i}{N}$ and $\frac{b_j}{N}$ stay constant $\forall i, j$. Therefore, at the limit of large N , the variance of $\frac{n_{ij}}{N}$ tends to 0: $\text{Var}\left(\frac{n_{ij}}{N}\right) = \frac{1}{N} \frac{a_i}{N} \frac{b_j}{N} \left(1 - \frac{a_i}{N}\right) \left(1 + \frac{1}{N-1} - \frac{b_j}{N}\right) \rightarrow 0$ and thus $\text{Var}(\phi(\mathbf{X}))$ tends to 0. ■

Theorem 7 *It holds true that:*

$$\lim_{N \rightarrow +\infty} \text{Var}(H_q(U, V)) = \lim_{N \rightarrow +\infty} \text{Var}(\text{MI}_q(U, V)) = \lim_{N \rightarrow +\infty} \text{Var}(\text{VI}_q(U, V)) = 0 \quad (16)$$

Proof Trivially follows from Lemma 4. ■

References

- Charu C. Aggarwal and Chandan K. Reddy. *Data Clustering: Algorithms and Applications*. CRC Press, 2013.
- Ahmed N. Albatineh and Magdalena Niewiadomska-Bugaj. Correcting Jaccard and other similarity indices for chance agreement in cluster analysis. *Advances in Data Analysis and Classification*, 5(3):179–200, 2011.
- Ahmed N. Albatineh, Magdalena Niewiadomska-Bugaj, and Daniel Mihalco. On similarity indices and correction for chance agreement. *Journal of Classification*, 23(2):301–313, 2006.
- Alessia Amelio and Clara Pizzuti. Is normalized mutual information a fair measure for comparing community detection methods? In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, pages 1584–1585. ACM, 2015.
- Alfredo H-S. Ang and Wilson H. Tang. *Probability Concepts in Engineering: Emphasis on Applications to Civil and Environmental Engineering*. John Wiley and Sons, 2006.
- Evan Archer, Il Memming Park, and Jonathan W Pillow. Bayesian and quasi-bayesian estimators for mutual information from discrete data. *Entropy*, 15(5):1738–1755, 2013.
- Asa Ben-Hur, André Elisseeff, and Isabelle Guyon. A stability based method for discovering structure in clustered data. In *Pacific symposium on bioinformatics*, volume 7, pages 6–17, 2001.
- Rich Caruana, M Elhaway, Nam Nguyen, and Casey Smith. Meta clustering. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 107–118. IEEE, 2006.
- Annalisa Cerquetti. Bayesian nonparametric estimation of tsallis diversity indices under gnedin-pitman priors. *arXiv preprint arXiv:1404.3441*, 2014.
- Mike X Cohen. *Analyzing neural time series data: theory and practice*. MIT Press, 2014.
- Antonio Criminisi, Jamie Shotton, and Ender Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, 7(2-3): 81–227, 2012.
- Zoltán Daróczy. Generalized information functions. *Information and control*, 16(1):36–51, 1970.
- Alin Dobra and Johannes Gehrke. Bias correction in classification tree construction. In *Proceedings of the International Conference on Machine Learning*, pages 90–97, 2001.
- Eibe Frank and Ian H. Witten. Using a permutation test for attribute selection in decision trees. In *Proceedings of the International Conference on Machine Learning*, pages 152–160, 1998.
- Shigeru Furuchi. Information theoretical properties of tsallis entropies. *Journal of Mathematical Physics*, 47(2):023302, 2006.
- Jan Havrda and František Charvát. Quantification method of classification processes. concept of structural α -entropy. *Kybernetika*, 3(1):30–35, 1967.
- Torsten Hothorn, Kurt Hornik, and Achim Zeileis. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15(3):651–674, 2006.
- Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2: 193–218, 1985.
- Yang Lei, James C Bezdek, Jeffrey Chan, Nguyen Xuan Vinh, Simone Romano, and James Bailey. Generalized information theoretic cluster validity indices for soft clusterings. In *IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, pages 24–31. IEEE, 2014a.
- Yang Lei, Nguyen Xuan Vinh, Jeffrey Chan, and James Bailey. Filta: Better view discovery from collections of clusterings via filtering. In *Machine Learning and Knowledge Discovery in Databases*, pages 145–160. Springer, 2014b.
- Yang Lei, James Bezdek, Jeffrey Chan, Nguyen Vinh, Simone Romano, and James Bailey. Extending information-theoretic validity indices for fuzzy clustering. *IEEE Transactions on Fuzzy Systems*, 2016.
- Fabrizio M Lopes, Evaldo A de Oliveira, and Roberto M Cesar. Inference of gene regulatory networks from time series by Tsallis entropy. *BMC systems biology*, 5(1):61, 2011.
- André FT Martins, Noah A Smith, Eric P Xing, Pedro MQ Aguiar, and Mário AT Figureiredo. Nonextensive information theoretic kernels on measures. *The Journal of Machine Learning Research*, 10:935–975, 2009.
- Tomasz Maszczyk and Włodzisław Duch. Comparison of Shannon, Renyi and Tsallis entropy used in decision trees. In *Artificial Intelligence and Soft Computing-ICAISC 2008*, pages 643–651. Springer, 2008.
- Marina Meilă. Comparing clusterings – an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, 2007.
- Leslie C Morey and Alan Agresti. The measurement of classification agreement: an adjustment to the rand statistic for chance agreement. *Educational and Psychological Measurement*, 44(1):33–37, 1984.
- Emmanuel Müller, Stephan Günnemann, Ines Färber, and Thomas Seidl. Discovering multiple clustering solutions: Grouping objects in different views of the data. Tutorial at ICML, 2013. URL <http://dme.rwth-aachen.de/en/DMCS>.
- Liam Paninski. Estimation of entropy and mutual information. *Neural computation*, 15(6): 1191–1253, 2003.

- William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.
- Alfred Renyi. On measures of entropy and information. 1961.
- Simone Romano, James Bailey, Vinh Nguyen, and Karin Verspoor. Standardized mutual information for clustering comparisons: One step further in adjustment for chance. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1143–1151, 2014.
- Simone Romano, Nguyen Xuan Vinh, James Bailey, and Karin Verspoor. A framework to adjust dependency measure estimates for chance. *arXiv preprint arXiv:1510.07786*, 2015.
- Sheldon Ross. *A first course in probability*. Pearson, 2012.
- Y-S Shih. A note on split selection bias in classification trees. *Computational statistics & data analysis*, 45(3):457–466, 2004.
- Dan Simovici. On generalized entropy and entropic metrics. *Journal of Multiple Valued Logic and Soft Computing*, 13(4/6):295, 2007.
- Alexander Strahl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3:583–617, 2003.
- Constantino Tsallis. Possible generalization of Boltzmann-Gibbs statistics. *Journal of statistical physics*, 52(1-2):479–487, 1988.
- Constantino Tsallis et al. *Introduction to Nonextensive Statistical Mechanics*. Springer, 2009.
- Marius Vila, Anton Bardera, Miquel Feixas, and Mateu Sbert. Tsallis mutual information for document classification. *Entropy*, 13(9):1694–1707, 2011.
- Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the International Conference on Machine Learning*, pages 1073–1080. ACM, 2009.
- Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11:2837–2854, 2010.
- Nguyen Xuan Vinh, Jeffrey Chan, Simone Romano, and James Bailey. Effective global approaches for mutual information based feature selection. In *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 512–521. ACM, 2014.
- Yisen Wang, Chaohing Song, and Shu-Tao Xia. Improving decision trees using tsallis entropy. *arXiv preprint arXiv:1511.08136*, 2015.
- Matthew J Warrens. On the equivalence of Cohens kappa and the Hubert-Arable adjusted Rand index. *Journal of Classification*, 25(2):177–183, 2008.
- Allan P. White and Wei Zhong Liu. Bias in information-based measures in decision tree induction. *Machine Learning*, pages 321–329, 1994.
- Junjie Wu, Hui Xiong, and Jian Chen. Adapting the right measures for k-means clustering. In *Knowledge Discovery and Data Mining*, pages 877–886, 2009.

Refined Error Bounds for Several Learning Algorithms

Steve Hanneke

STEVE.HANNEKE@GMAIL.COM

Editor: John Shawe-Taylor

Abstract

This article studies the achievable guarantees on the error rates of certain learning algorithms, with particular focus on refining logarithmic factors. Many of the results are based on a general technique for obtaining bounds on the error rates of sample-consistent classifiers with monotonic error regions, in the realizable case. We prove bounds of this type expressed in terms of either the VC dimension or the sample compression size. This general technique also enables us to derive several new bounds on the error rates of general sample-consistent learning algorithms, as well as refined bounds on the label complexity of the CAL active learning algorithm. Additionally, we establish a simple necessary and sufficient condition for the existence of a distribution-free bound on the error rates of all sample-consistent learning rules, converging at a rate inversely proportional to the sample size. We also study learning in the presence of classification noise, deriving a new excess error rate guarantee for general VC classes under Tsybakov's noise condition, and establishing a simple and general necessary and sufficient condition for the minimax excess risk under bounded noise to converge at a rate inversely proportional to the sample size.

Keywords: sample complexity, PAC learning, statistical learning theory, active learning, minimax analysis

1. Introduction

Supervised machine learning is a classic topic, in which a learning rule is tasked with producing a classifier that mimics the classifications that would be assigned by an expert for a given task. To achieve this, the learner is given access to a collection of examples (assumed to be i.i.d.) labeled with the correct classifications. One of the major theoretical questions of interest in learning theory is: How many examples are necessary and sufficient for a given learning rule to achieve low classification error rate? This quantity is known as the *sample complexity*, and varies depending on how small the desired classification error rate is, the type of classifier we are attempting to learn, and various other factors. Equivalently, the question is: How small of an error rate can we guarantee a given learning rule will achieve, for a given number of labeled training examples?

A particularly simple setting for supervised learning is the *realizable case*, in which it is assumed that, within a given set \mathcal{C} of classifiers, there resides some classifier that is *always* correct. The optimal sample complexity of learning in the realizable case has recently been completely resolved, up to constant factors, in a sibling paper to the present article (Hanneke, 2016). However, there remains the important task of identifying interesting general families of algorithms achieving this optimal sample complexity. For instance, the best known general upper bounds for the general family of *empirical risk minimization* algorithms differ from the optimal sample complexity by a logarithmic factor, and it is

known that there exist spaces \mathcal{C} for which this is unavoidable (Auer and Ortner, 2007). This same logarithmic factor gap appears in the analysis of several of other learning methods as well. The present article focuses on this logarithmic factor, arguing that for certain types of learning rules, it can be entirely removed in some cases, and for others it can be somewhat refined. The technique leading to these results is rooted in an idea introduced in the author's doctoral dissertation (Hanneke, 2009). By further exploring this technique, we also obtain new results for the related problem of *active learning*. We also derive interesting new results for learning with classification noise, where again the focus is on a logarithmic factor gap between upper and lower bounds.

1.1 Basic Notation

Before further discussing the results, we first introduce some essential notation. Let \mathcal{X} be any nonempty set, called the *instance space*, equipped with a σ -algebra defining the measurable sets; for simplicity, we will suppose the sets in $\{\{x\} : x \in \mathcal{X}\}$ are all measurable. Let $\mathcal{Y} = \{-1, +1\}$ be the label space. A *classifier* is any measurable function $h : \mathcal{X} \rightarrow \mathcal{Y}$. Following Vapnik and Chervonenkis (1971), define the VC dimension of a set \mathcal{A} of subsets of \mathcal{X} , denoted $\text{vc}(\mathcal{A})$, as the maximum cardinality $|S|$ over subsets $S \subseteq \mathcal{X}$ such that $\{S \cap A : A \in \mathcal{A}\} = 2^S$ (the power set of S); if no such maximum cardinality exists, define $\text{vc}(\mathcal{A}) = \infty$. For any set \mathcal{H} of classifiers, denote by $\text{vc}(\mathcal{H}) = \text{vc}(\{\{x\} : h(x) = +1\} : h \in \mathcal{H})$ the VC dimension of \mathcal{H} . Throughout, we fix a set \mathcal{C} of classifiers, known as the *concept space*, and abbreviate $d = \text{vc}(\mathcal{C})$. To focus on nontrivial cases, throughout we suppose $|\mathcal{C}| \geq 3$, which implies $d \geq 1$. We will also generally suppose $d < \infty$ (though some of the results would still hold without this restriction).

For any $L_m = \{(x_1, y_1), \dots, (x_m, y_m)\} \in (\mathcal{X} \times \mathcal{Y})^m$, and any classifier h , define $\text{er}_{L_m}(h) = \frac{1}{m} \sum_{(x,y) \in L_m} \mathbb{1}[h(x) \neq y]$. For completeness, also define $\text{er}_{\mathcal{H}}(h) = 0$. Also, for any set \mathcal{H} of classifiers, denote $\mathcal{H}[L_m] = \{h \in \mathcal{H} : \forall (x, y) \in L_m, h(x) = y\}$, referred to as the set of classifiers in \mathcal{H} *consistent* with L_m ; for completeness, also define $\mathcal{H}[\{\}] = \mathcal{H}$. Fix an arbitrary probability measure \mathcal{P} on \mathcal{X} (called the *data distribution*), and a classifier $f^* \in \mathcal{C}$ (called the *target function*). For any classifier h , denote $\text{er}(h) = \mathcal{P}(x : h(x) \neq f^*(x))$, the *error rate* of h . Let X_1, X_2, \dots be independent \mathcal{P} -distributed random variables. We generally denote $\mathcal{L}_m = \{(X_1, f^*(X_1)), \dots, (X_m, f^*(X_m))\}$, and $V_m = \mathcal{C}[\mathcal{L}_m]$ (called the *version space*). The general setting in which we are interested in producing a classifier \hat{h} with small $\text{er}(\hat{h})$, given access to the data \mathcal{L}_m , is a special case of supervised learning known as the *realizable case* (in contrast to settings where the observed labeling might not be realizable by any classifier in \mathcal{C} , due to label noise or model misspecification, as discussed in Section 6).

We adopt a few convenient notational conventions. For any $m \in \mathbb{N}$, denote $[m] = \{1, \dots, m\}$; also denote $[0] = \{\}$. We adopt a shorthand notation for sequences, so that for a sequence x_1, \dots, x_m , we denote $x_{[m]} = (x_1, \dots, x_m)$. For any \mathbb{R} -valued functions f, g , we write $f(z) \lesssim g(z)$ or $g(z) \gtrsim f(z)$ if there exists a finite numerical constant $c > 0$ such that $f(z) \leq cg(z)$ for all z . For any $x, y \in \mathbb{R}$, denote $x \vee y = \max\{x, y\}$ and $x \wedge y = \min\{x, y\}$. For $x \geq 0$, denote $\text{Log}(x) = \ln(x \vee e)$ and $\text{Log}_2(x) = \log_2(x \vee 2)$. We also adopt the conventions that for $x > 0$, $x/0 = \infty$, and $0\text{Log}(x/0) = 0\text{Log}(\infty) = 0 \cdot \infty = 0$. It will also be convenient to use the notation $\mathcal{Z}^0 = \{\emptyset\}$ for a set \mathcal{Z} , where $(\)$ is the empty sequence.

Throughout, we also make the usual implicit assumption that all quantities required to be measurable in the proofs and lemmas from the literature are indeed measurable. See, for instance, van der Vaart and Wellner (1996, 2011), for discussions of conditions on \mathbb{C} that typically suffice for this.

1.2 Background and Summary of the Main Results

This work concerns the study of the error rates achieved by various *learning rules*: that is, mappings from the data set \mathcal{L}_m to a classifier \hat{h}_m ; for simplicity, we sometimes refer to \hat{h}_m itself as a learning rule, leaving dependence on \mathcal{L}_m implicit. There has been a substantial amount of work on bounding the error rates of various learning rules in the realizable case. Perhaps the most basic and natural type of learning rule in this setting is the family of *consistent* learning rules: that is, those that choose $\hat{h}_m \in V_m$. There is a general upper bound for all consistent learning rules \hat{h}_m , due to Vapnik and Chervonenkis (1974); Blumer, Ehrenfeucht, Haussler, and Warmuth (1989), stating that with probability at least $1 - \delta$,

$$\text{er}(\hat{h}_m) \lesssim \frac{1}{m} \left(d \log \left(\frac{m}{d} \right) + \text{Log} \left(\frac{1}{\delta} \right) \right). \quad (1)$$

This is complemented by a general lower bound of Ehrenfeucht, Haussler, Kearns, and Valiant (1989), which states that for any learning rule (consistent or otherwise), there exists a choice of \mathcal{P} and $f^* \in \mathbb{C}$ such that, with probability greater than δ ,

$$\text{er}(\hat{h}_m) \gtrsim \frac{1}{m} \left(d + \text{Log} \left(\frac{1}{\delta} \right) \right). \quad (2)$$

Resolving the logarithmic factor gap between (2) and (1) has been a challenging subject of study for decades now, with many interesting contributions resolving special cases and proposing sometimes-better upper bounds (e.g., Haussler, Littlestone, and Warmuth, 1994; Giné and Koltchinskii, 2006; Auer and Ortner, 2007; Long, 2003). It is known that the lower bound is sometimes *not* achieved by certain consistent learning rules (Auer and Ortner, 2007). The question of whether the lower bound (2) can always be achieved by *some* algorithm remained open for a number of years (Ehrenfeucht, Haussler, Kearns, and Valiant, 1989; Warmuth, 2004), but has recently been resolved in a sibling paper to the present article (Hanneke, 2016). That work proposes a learning rule \hat{h}_m based on a majority vote of classifiers consistent with carefully-constructed subsamples of the data, and proves that with probability at least $1 - \delta$,

$$\text{er}(\hat{h}_m) \lesssim \frac{1}{m} \left(d + \text{Log} \left(\frac{1}{\delta} \right) \right).$$

However, several avenues for investigation remain open, including identifying interesting general families of learning rules able to achieve this optimal bound under general conditions on \mathbb{C} . In particular, it remains an open problem to determine necessary and sufficient conditions on \mathbb{C} for the entire family of consistent learning rules to achieve the above optimal error bound.

The work of Giné and Koltchinskii (2006) includes a bound that refines the logarithmic factor in (1) in certain scenarios. Specifically, it states that, for any consistent learning rule

\hat{h}_m , with probability at least $1 - \delta$,

$$\text{er}(\hat{h}_m) \lesssim \frac{1}{m} \left(d \log \left(\theta \left(\frac{d}{m} \right) \right) + \text{Log} \left(\frac{1}{\delta} \right) \right), \quad (3)$$

where $\theta(\cdot)$ is the *disagreement coefficient* (defined below in Section 4). The doctoral dissertation of Hanneke (2009) contains a simple and direct proof of this bound, based on an argument which splits the data set in two parts, and considers the second part as containing a subsequence sampled from the conditional distribution given the region of disagreement of the version space induced by the first part of the data. Many of the results in the present work are based on variations of this argument, including a variety of interesting new bounds on the error rates achieved by certain families of learning rules.

As one of the cornerstones of this work, we find that a variant of this argument for consistent learning rules with *monotonic* error regions leads to an upper bound that *matches* the lower bound (2) up to constant factors. For such monotonic consistent learning rules to exist, we would need a very special kind of concept space. However, they do exist in some important cases. In particular, in the special case of learning *intersection-closed* concept spaces, the *Closure* algorithm (Natarajan, 1987; Auer and Ortner, 2004, 2007) can be shown to satisfy this monotonicity property. Thus, this result immediately implies that, with probability at least $1 - \delta$, the Closure algorithm achieves

$$\text{er}(\hat{h}_m) \lesssim \frac{1}{m} \left(d + \text{Log} \left(\frac{1}{\delta} \right) \right),$$

which was an open problem of Auer and Ortner (2004, 2007); this fact was recently also obtained by Darnstädt (2015), via a related direct argument. We also discuss a variant of this result for monotone learning rules expressible as *compression schemes*, where we remove a logarithmic factor present in a result of Littlestone and Warmuth (1986) and Floyd and Warmuth (1995), so that for \hat{h}_m based on a compression scheme of size n , which has monotonic error regions (and is permutation-invariant), with probability at least $1 - \delta$,

$$\text{er}(\hat{h}_m) \lesssim \frac{1}{m} \left(n + \text{Log} \left(\frac{1}{\delta} \right) \right).$$

This argument also has implications for *active learning*. In many active learning algorithms, the *region of disagreement* of the version space induced by m samples, $\text{DIS}(V_m) = \{x \in \mathcal{X} : \exists h, g \in V_m \text{ s.t. } h(x) \neq g(x)\}$, plays an important role. In particular, the label complexity of the CAL active learning algorithm (Cohn, Atlas, and Laderer, 1994) is largely determined by the rate at which $\mathcal{P}(\text{DIS}(V_m))$ decreases, so that any bound on this quantity can be directly converted into a bound on the label complexity of CAL (Hanneke, 2011, 2009, 2014; El-Yaniv and Wiener, 2012). Wiener, Hanneke, and El-Yaniv (2015) have argued that the region $\text{DIS}(V_m)$ can be described as a compression scheme, where the size of the compression scheme, denoted \hat{n}_m , is known as the *version space compression set size* (Definition 6 below). By further observing that $\text{DIS}(V_m)$ is monotonic in m , applying our general argument yields the fact that, with probability at least $1 - \delta$, letting $\hat{n}_{1:m} = \max_{t \in [m]} \hat{n}_t$,

$$\mathcal{P}(\text{DIS}(V_m)) \lesssim \frac{1}{m} \left(\hat{n}_{1:m} + \text{Log} \left(\frac{1}{\delta} \right) \right), \quad (4)$$

which is typically an improvement over the best previously-known general bound by a logarithmic factor.

In studying the distribution-free minimax label complexity of active learning, Hanneke and Yang (2015) found that a simple combinatorial quantity \mathfrak{s} , which they term the *star number*, is of fundamental importance. Specifically (see also Definition 9), \mathfrak{s} is the largest number s of distinct points $x_1, \dots, x_s \in \mathcal{X}$ such that $\exists h_0, h_1, \dots, h_s \in \mathbb{C}$ with $\forall i \in [s]$, $\text{DIS}(\{h_0, h_i\}) \cap \{x_1, \dots, x_s\} = \{x_i\}$, or else $\mathfrak{s} = \infty$ if no such largest s exists. Interestingly, the work of Hanneke and Yang (2015) also establishes that the largest possible value of \hat{h}_m (over m and the data set) is exactly \mathfrak{s} . Thus, (4) also implies a *data-independent* and *distribution-free* bound: with probability at least $1 - \delta$,

$$\mathcal{P}(\text{DIS}(V_m)) \lesssim \frac{1}{m} \left(\mathfrak{s} + \text{Log} \left(\frac{1}{\delta} \right) \right).$$

Now one interesting observation at this point is that the direct proof of (3) from Hanneke (2009) involves a step in which $\mathcal{P}(\text{DIS}(V_m))$ is relaxed to a bound in terms of $\theta(d/m)$. If we instead use (4) in this step, we arrive at a new bound on the error rates of *all* consistent learning rules \hat{h}_m : with probability at least $1 - \delta$,

$$\text{er}(\hat{h}_m) \lesssim \frac{1}{m} \left(d \text{Log} \left(\frac{\hat{h}_{1,m}}{d} \right) + \text{Log} \left(\frac{1}{\delta} \right) \right). \quad (5)$$

Since Hanneke and Yang (2015) have shown that the maximum possible value of $\theta(d/m)$ (over m , \mathcal{P} , and f^*) is also exactly the star number \mathfrak{s} , while $\hat{h}_{1,m}/d$ has as its maximum possible value \mathfrak{s}/d , we see that the bound in (5) sometimes reflects an improvement over (3). It further implies a new data-independent and distribution-free bound for any consistent learning rule \hat{h}_m : with probability at least $1 - \delta$,

$$\text{er}(\hat{h}_m) \lesssim \frac{1}{m} \left(d \text{Log} \left(\frac{\min\{\mathfrak{s}, m\}}{d} \right) + \text{Log} \left(\frac{1}{\delta} \right) \right).$$

Interestingly, we are able to complement this with a *lower bound* in Section 5.1. Though not quite matching the above in terms of its joint dependence on d and \mathfrak{s} (and necessarily so), this lower bound does provide the interesting observation that $\mathfrak{s} < \infty$ is *necessary and sufficient* for there to exist a distribution-free bound on the error rates of all consistent learning rules, converging at a rate $\Theta(1/m)$, and otherwise (when $\mathfrak{s} = \infty$) the best such bound is $\Theta(\text{Log}(m)/m)$.

Continuing with the investigation of general consistent learning rules, we also find a variant of the argument of Hanneke (2009) that refines (3) in a different way: namely, replacing $\theta(\cdot)$ with a quantity based on considering a well-chosen *subregion* of the region of disagreement, as studied by Balcan, Broder, and Zhang (2007); Zhang and Chaudhuri (2014). Specifically, in the context of active learning, Zhang and Chaudhuri (2014) have proposed a general quantity $\varphi_c(\cdot)$ (Definition 15 below), which is never larger than $\theta(\cdot)$, and is sometimes significantly smaller. By adapting our general argument to replace $\text{DIS}(V_m)$ with this well-chosen subregion, we derive a bound for all consistent learning rules \hat{h}_m : with probability at least $1 - \delta$,

$$\text{er}(\hat{h}_m) \lesssim \frac{1}{m} \left(d \text{Log} \left(\varphi_c \left(\frac{d}{m} \right) \right) + \text{Log} \left(\frac{1}{\delta} \right) \right).$$

In particular, as a special case of this general result, we recover the theorem of Balcan and Long (2013) that all consistent learning rules have optimal sample complexity (up to constants) for the problem of learning homogeneous linear separators under isotropic log-concave distributions, as $\varphi_c(d/m)$ is bounded by a finite numerical constant in this case. In Section 6, we also extend this result to the problem of learning with *classification noise*, where there is also a logarithmic factor gap between the known general-case upper and lower bounds. In this context, we derive a new general upper bound under the Bernstein class condition (a generalization of Tsybakov’s noise condition), expressed in terms of a quantity related to $\varphi_c(\cdot)$, which applies to a particular learning rule. This sometimes reflects an improvement over the best previous general upper bounds (Massart and Nédélec, 2006; Giné and Koltchinskii, 2006; Hanneke and Yang, 2012), and again recovers a result of Balcan and Long (2013) for homogeneous linear separators under isotropic log-concave distributions, as a special case.

For many of these results, we also state bounds on the *expected* error rate: $\mathbb{E}[\text{er}(\hat{h}_m)]$. In this case, the optimal distribution-free bound is known to be within a constant factor of d/m (Haussler, Littlestone, and Warmuth, 1994; Li, Long, and Srinivasan, 2001), and this rate is achieved by the one-inclusion graph prediction algorithm of Haussler, Littlestone, and Warmuth (1994), as well as the majority voting method of Hanneke (2016). However, there remain interesting questions about whether other algorithms achieve this optimal performance, or require an extra logarithmic factor. Again we find that *monotone* consistent learning rules indeed achieve this optimal d/m rate (up to constant factors), while a distribution-free bound on $\mathbb{E}[\text{er}(\hat{h}_m)]$ with $\Theta(1/m)$ dependence on m is achieved by all consistent learning rules if and only if $\mathfrak{s} < \infty$, and otherwise the best such bound has $\Theta(\text{Log}(m)/m)$ dependence on m .

As a final interesting result, in the context of learning with classification noise, under the *bounded noise assumption* (Massart and Nédélec, 2006), we find that the condition $\mathfrak{s} < \infty$ is actually *necessary and sufficient* for the *minimax optimal* excess error rate to decrease at a rate $\Theta(1/m)$, and otherwise (if $\mathfrak{s} = \infty$) it decreases at a rate $\Theta(\text{Log}(m)/m)$. This result generalizes several special-case analyses from the literature (Massart and Nédélec, 2006; Raginsky and Rakhlin, 2011). Note that the “necessity” part of this statement is significantly stronger than the above result for consistent learning rules in the realizable case, since this result applies to the best error guarantee achievable by *any* learning rule.

2. Bounds for Consistent Monotone Learning

In order to state our results for monotonic learning rules in an abstract form, we introduce the following notation. Let \mathcal{Z} denote any space, equipped with a σ -algebra defining the measurable subsets. For any collection \mathcal{A} of measurable subsets of \mathcal{Z} , a *consistent monotone rule* is any sequence of functions $\psi_t : \mathcal{Z}^t \rightarrow \mathcal{A}$, $t \in \mathbb{N}$, such that $\forall z_1, z_2, \dots \in \mathcal{Z}$, $\forall t \in \mathbb{N}$, $\psi_t(z_1, \dots, z_t) \cap \{z_1, \dots, z_t\} = \emptyset$, and $\forall t \in \mathbb{N}$, $\psi_{t+1}(z_1, \dots, z_{t+1}) \subseteq \psi_t(z_1, \dots, z_t)$. We begin with the following very simple result, the proof of which will also serve to introduce, in its simplest form, the core technique underlying many of the results presented in later sections below.

Theorem 1 Let \mathcal{A} be a collection of measurable subsets of \mathcal{Z} , and let $\psi_t : \mathcal{Z}^t \rightarrow \mathcal{A}$ (for $t \in \mathbb{N}$) be any consistent monotone rule. Fix any $m \in \mathbb{N}$, any $\delta \in (0, 1)$, and any probability measure P on \mathcal{Z} . Letting Z_1, \dots, Z_m be independent P -distributed random variables, and denoting $A_m = \psi_m(Z_1, \dots, Z_m)$, with probability at least $1 - \delta$,

$$P(A_m) \leq \frac{4}{m} \left(17\text{vc}(\mathcal{A}) + 4 \ln \left(\frac{4}{\delta} \right) \right). \quad (6)$$

Furthermore,

$$\mathbb{E}[P(A_m)] \leq \frac{68(\text{vc}(\mathcal{A}) + 1)}{m}. \quad (7)$$

The overall structure of this proof is based on an argument of Hanneke (2009). The most-significant novel element here is the use of monotonicity to further refine a logarithmic factor. The proof relies on the following classic result. Results of this type are originally due to Vapnik and Chervonenkis (1974); the version stated here features slightly better constant factors, due to Blumer, Ehrenfeucht, Haussler, and Warmuth (1989).

Lemma 2 For any collection \mathcal{A} of measurable subsets of \mathcal{Z} , any $\delta \in (0, 1)$, any $m \in \mathbb{N}$, and any probability measure P on \mathcal{Z} , letting Z_1, \dots, Z_m be independent P -distributed random variables, with probability at least $1 - \delta$, every $A \in \mathcal{A}$ with $A \cap \{Z_1, \dots, Z_m\} = \emptyset$ satisfies

$$P(A) \leq \frac{2}{m} \left(\text{vc}(\mathcal{A}) \text{Log}_2 \left(\frac{2em}{\text{vc}(\mathcal{A})} \right) + \text{Log}_2 \left(\frac{2}{\delta} \right) \right).$$

We are now ready for the proof of Theorem 1.

Proof of Theorem 1 Fix any probability measure P , let Z_1, Z_2, \dots be independent P -distributed random variables, and for each $m \in \mathbb{N}$ denote $A_m = \psi_m(Z_1, \dots, Z_m)$. We begin with the inequality in (6). The proof proceeds by induction on m . If $m \leq 200$, then since $\log_2(400e) < 34$ and $\log_2(\frac{2}{\delta}) < 8 \ln(\frac{4}{\delta})$, and since the definition of a consistent monotone rule implies $A_m \cap \{Z_1, \dots, Z_m\} = \emptyset$, the stated bound follows immediately from Lemma 2 for any $\delta \in (0, 1)$. Now, as an inductive hypothesis, fix any integer $m \geq 201$ such that, $\forall m' \in [m-1]$, $\forall \delta \in (0, 1)$, with probability at least $1 - \delta$,

$$P(A_{m'}) \leq \frac{4}{m'} \left(17\text{vc}(\mathcal{A}) + 4 \ln \left(\frac{4}{\delta} \right) \right).$$

Now fix any $\delta \in (0, 1)$ and define

$$N = |\{Z_{[m/2]+1}, \dots, Z_m\} \cap A_{[m/2]}|,$$

and enumerate the elements of $\{Z_{[m/2]+1}, \dots, Z_m\} \cap A_{[m/2]}$ as $\hat{Z}_1, \dots, \hat{Z}_N$ (retaining their original order).

Note that $N = \sum_{i=[m/2]+1}^m \mathbb{1}_{A_{[m/2]}}(Z_i)$ is conditionally Binomial($[m/2]$, $P(A_{[m/2]})$)-distributed given $Z_1, \dots, Z_{[m/2]}$. In particular, with probability one, if $P(A_{[m/2]}) = 0$, then $N = 0$. Otherwise, if $P(A_{[m/2]}) > 0$, then note that Z_1, \dots, Z_N are conditionally independent and $P(\cdot | A_{[m/2]})$ -distributed given $Z_1, \dots, Z_{[m/2]}$ and N . Thus, since $A_m \cap \{\hat{Z}_1, \dots, \hat{Z}_N\} \subseteq A_m \cap \{Z_1, \dots, Z_m\} = \emptyset$, applying Lemma 2 (under the conditional

distribution given N and $Z_1, \dots, Z_{[m/2]}$), combined with the law of total probability, we have that on an event E_1 of probability at least $1 - \delta/2$, if $N > 0$, then

$$P(A_m | A_{[m/2]}) \leq \frac{2}{N} \left(\text{vc}(\mathcal{A}) \text{Log}_2 \left(\frac{2eN}{\text{vc}(\mathcal{A})} \right) + \log_2 \left(\frac{4}{\delta} \right) \right).$$

Additionally, again since N is conditionally Binomial($[m/2]$, $P(A_{[m/2]})$)-distributed given $Z_1, \dots, Z_{[m/2]}$, applying a Chernoff bound (under the conditional distribution given $Z_1, \dots, Z_{[m/2]}$), combined with the law of total probability, we obtain that on an event E_2 of probability at least $1 - \delta/4$, if $P(A_{[m/2]}) \geq \frac{16}{m} \ln(\frac{4}{\delta})$, then

$$N \geq P(A_{[m/2]}) \lceil m/2 \rceil / 2 \geq P(A_{[m/2]}) m / 4.$$

In particular, if $P(A_{[m/2]}) \geq \frac{16}{m} \ln(\frac{4}{\delta})$, then $P(A_{[m/2]}) m / 4 > 0$, so that if this occurs with E_2 , then we have $N > 0$. Noting that $\text{Log}_2(x) \leq \text{Log}(x) / \ln(2)$, then by monotonicity of $x \mapsto \text{Log}(x)/x$ for $x > 0$, we have that on $E_1 \cap E_2$, if $P(A_{[m/2]}) \geq \frac{16}{m} \ln(\frac{4}{\delta})$, then

$$P(A_m | A_{[m/2]}) \leq \frac{8}{P(A_{[m/2]}) m \ln(2)} \left(\text{vc}(\mathcal{A}) \text{Log} \left(\frac{eP(A_{[m/2]}) m}{2\text{vc}(\mathcal{A})} \right) + \ln \left(\frac{4}{\delta} \right) \right).$$

The monotonicity property of ψ_t implies $A_m \subseteq A_{[m/2]}$. Together with monotonicity of probability measures, this implies $P(A_m) \leq P(A_{[m/2]})$. It also implies that, if $P(A_{[m/2]}) > 0$, then $P(A_m) = P(A_m | A_{[m/2]}) P(A_{[m/2]})$. Thus, on $E_1 \cap E_2$, if $P(A_m) \geq \frac{16}{m} \ln(\frac{4}{\delta})$, then

$$P(A_m) \leq \frac{8}{m \ln(2)} \left(\text{vc}(\mathcal{A}) \text{Log} \left(\frac{eP(A_{[m/2]}) m}{2\text{vc}(\mathcal{A})} \right) + \ln \left(\frac{4}{\delta} \right) \right).$$

The inductive hypothesis implies that, on an event E_3 of probability at least $1 - \delta/4$,

$$P(A_{[m/2]}) \leq \frac{4}{\lceil m/2 \rceil} \left(17\text{vc}(\mathcal{A}) + 4 \ln \left(\frac{16}{\delta} \right) \right).$$

Since $m \geq 201$, we have $\lceil m/2 \rceil \geq (m-2)/2 \geq (199/402)m$, so that the above implies

$$P(A_{[m/2]}) \leq \frac{4 \cdot 402}{199m} \left(17\text{vc}(\mathcal{A}) + 4 \ln \left(\frac{16}{\delta} \right) \right).$$

Thus, on $E_1 \cap E_2 \cap E_3$, if $P(A_m) \geq \frac{16}{m} \ln(\frac{4}{\delta})$, then

$$P(A_m) \leq \frac{8}{m \ln(2)} \left(\text{vc}(\mathcal{A}) \text{Log} \left(\frac{2 \cdot 402e}{199} \left(17 + \frac{4}{\text{vc}(\mathcal{A})} \ln \left(\frac{16}{\delta} \right) \right) \right) + \ln \left(\frac{4}{\delta} \right) \right).$$

Lemma 20 in Appendix A allows us to simplify the logarithmic term here, revealing that the right hand side is at most

$$\begin{aligned} & \frac{8}{m \ln(2)} \left(\text{vc}(\mathcal{A}) \text{Log} \left(\frac{2 \cdot 402e}{199} \left(17 + 4 \ln(4) + \frac{4}{\ln(4/e)} \right) \right) + \left(1 + \ln \left(\frac{4}{e} \right) \right) \ln \left(\frac{4}{\delta} \right) \right) \\ & \leq \frac{4}{m} \left(17\text{vc}(\mathcal{A}) + 4 \ln \left(\frac{4}{\delta} \right) \right). \end{aligned}$$

Since $\frac{16}{m} \ln(\frac{4}{\delta}) \leq \frac{4}{m} (17\text{vc}(\mathcal{A}) + 4 \ln(\frac{4}{\delta}))$, we have that, on $E_1 \cap E_2 \cap E_3$, regardless of whether or not $P(A_m) \geq \frac{16}{m} \ln(\frac{4}{\delta})$, we have

$$P(A_m) \leq \frac{4}{m} \left(17\text{vc}(\mathcal{A}) + 4 \ln\left(\frac{4}{\delta}\right) \right).$$

Noting that, by the union bound, the event $E_1 \cap E_2 \cap E_3$ has probability at least $1 - \delta$, this extends the inductive hypothesis to $m' = m$. By the principle of induction, this completes the proof of the first claim in Theorem 1.

For the bound on the expectation in (7), we note that, letting $\varepsilon_m = \frac{4}{m} (17\text{vc}(\mathcal{A}) + 4 \ln(\frac{4}{\delta}))$, by setting the bound in (6) equal to a value ε and solving for δ , the value of which is in $(0, 1)$ for any $\varepsilon > \varepsilon_m$, the result just established can be restated as: $\forall \varepsilon > \varepsilon_m$,

$$\mathbb{P}(P(A_m) > \varepsilon) \leq 4 \exp\{(17/4)\text{vc}(\mathcal{A}) - \varepsilon m/16\}.$$

Furthermore, for any $\varepsilon \leq \varepsilon_m$, we of course still have $\mathbb{P}(P(A_m) > \varepsilon) \leq 1$. Therefore, we have that

$$\begin{aligned} \mathbb{E}[P(A_m)] &= \int_0^\infty \mathbb{P}(P(A_m) > \varepsilon) d\varepsilon \leq \varepsilon_m + \int_{\varepsilon_m}^\infty 4 \exp\{(17/4)\text{vc}(\mathcal{A}) - \varepsilon m/16\} d\varepsilon \\ &= \varepsilon_m + \frac{4 \cdot 16}{m} \exp\{(17/4)\text{vc}(\mathcal{A}) - \varepsilon_m m/16\} = \frac{4}{m} (17\text{vc}(\mathcal{A}) + 4 \ln(\frac{4}{\delta})) + \frac{16}{m} \\ &= \frac{4}{m} (17\text{vc}(\mathcal{A}) + 4 \ln(\frac{4}{\delta})) \leq \frac{68\text{vc}(\mathcal{A}) + 39}{m} \leq \frac{68\text{vc}(\mathcal{A}) + 1}{m}. \end{aligned}$$

■

We can also state a variant of Theorem 1 applicable to *sample compression schemes*, which will in fact be more useful for our purposes below. To state this result, we first introduce the following additional terminology. For any $t \in \mathbb{N}$, we say that a function $\psi: \mathcal{Z}^t \rightarrow \mathcal{A}$ is *permutation-invariant* if every $z_1, \dots, z_t \in \mathcal{Z}$ and every bijection $\kappa: [t] \rightarrow [t]$ satisfy $\psi(z_{\kappa(1)}, \dots, z_{\kappa(t)}) = \psi(z_1, \dots, z_t)$. For any $n \in \mathbb{N} \cup \{0\}$, a *consistent monotone sample compression rule of size n* is a consistent monotone rule ψ_t with the additional properties that, $\forall t \in \mathbb{N}$, ψ_t is permutation-invariant, and $\forall z_1, \dots, z_t \in \mathcal{Z}$, $\exists n_t(z_{[t]}) \in [\min\{n, t\}] \cup \{0\}$ such that

$$\psi_t(z_1, \dots, z_t) = \phi_{t, n_t(z_{[t]})}(z_{i_1(z_{[t]})}, \dots, z_{i_{n_t(z_{[t]})}(z_{[t]})}(z_{[t]}),$$

where $\phi_{t,k}: \mathcal{Z}^k \rightarrow \mathcal{A}$ is a permutation-invariant function for each $k \in [\min\{n, t\}] \cup \{0\}$, and i_1, \dots, i_{n_t} are functions $\mathcal{Z}^t \rightarrow [t]$ such that $\forall z_1, \dots, z_t \in \mathcal{Z}$, $i_{1,1}(z_{[t]}), \dots, i_{t, n_t}(z_{[t]})(z_{[t]})$ are all distinct. In words, the element of \mathcal{A} mapped to by $\psi_t(z_1, \dots, z_t)$ depends only on the unordered (multi)set $\{z_1, \dots, z_t\}$, and can be specified by an unordered subset of $\{z_1, \dots, z_t\}$ of size at most n . Following the terminology from the literature on sample compression schemes, we refer to the collection of functions $\{(n_t, i_{1,1}, \dots, i_{t, n_t}) : t \in \mathbb{N}\}$ as the *compression function* of ψ_t , and to the collection of permutation-invariant functions $\{\phi_{t,k} : t \in \mathbb{N}, k \in [\min\{n, t\}] \cup \{0\}\}$ as the *reconstruction function* of ψ_t .

This kind of ψ_t is a type of sample compression scheme (see Littlestone and Warmuth, 1986; Floyd and Warmuth, 1995), though certainly not all permutation-invariant compression schemes yield consistent monotone rules. Below, we find that consistent monotone

sample compression rules of a quantifiable size arise naturally in the analysis of certain learning algorithms (namely, the Closure algorithm and the CAL active learning algorithm). With the above terminology in hand, we can now state our second abstract result.

Theorem 3 Fix any $n \in \mathbb{N} \cup \{0\}$, let \mathcal{A} be a collection of measurable subsets of \mathcal{Z} , and let $\psi_t: \mathcal{Z}^t \rightarrow \mathcal{A}$ (for $t \in \mathbb{N}$) be any consistent monotone sample compression rule of size n . Fix any $m \in \mathbb{N}$, $\delta \in (0, 1)$, and any probability measure P on \mathcal{Z} . Letting Z_1, \dots, Z_m be independent P -distributed random variables, and denoting $A_m = \psi_m(Z_1, \dots, Z_m)$, with probability at least $1 - \delta$,

$$P(A_m) \leq \frac{1}{m} \left(21n + 16 \ln\left(\frac{3}{\delta}\right) \right). \quad (8)$$

Furthermore,

$$\mathbb{E}[P(A_m)] \leq \frac{21n + 34}{m}. \quad (9)$$

The proof of Theorem 3 relies on the following classic result due to Littlestone and Warmuth (1986); Floyd and Warmuth (1995) (see also Herbrich, 2002; Wiener, Hanneke, and El-Yaniv, 2015, for a clear and direct proof).

Lemma 4 Fix any collection \mathcal{A} of measurable subsets of \mathcal{Z} , any $m \in \mathbb{N}$ and $n \in \mathbb{N} \cup \{0\}$ with $n < m$, and any permutation-invariant functions $\phi_k: \mathcal{Z}^k \rightarrow \mathcal{A}$, $k \in [n] \cup \{0\}$. For any probability measure P on \mathcal{Z} , letting Z_1, \dots, Z_m be independent P -distributed random variables, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, every $k \in [n] \cup \{0\}$, and every distinct $i_1, \dots, i_k \in [m]$ with $\phi_k(Z_{i_1}, \dots, Z_{i_k}) \cap \{Z_1, \dots, Z_m\} = \emptyset$ satisfy

$$P(\phi_k(Z_{i_1}, \dots, Z_{i_k})) \leq \frac{1}{m-n} \left(n \text{Log}\left(\frac{em}{n}\right) + \text{Log}\left(\frac{1}{\delta}\right) \right).$$

With this lemma in hand, we are ready for the proof of Theorem 3.

Proof of Theorem 3 The proof follows analogously to that of Theorem 1, but with several additional complications due to the form of Lemma 4 being somewhat different from that of Lemma 2. Let $\{(n_t, i_{1,1}, \dots, i_{t, n_t}) : t \in \mathbb{N}\}$ and $\{\phi_{t,k} : t \in \mathbb{N}, k \in [\min\{n, t\}] \cup \{0\}\}$ be the compression function and reconstruction function of ψ_t , respectively. For convenience, also denote $\psi_0() = \mathcal{Z}$, and note that this extends the monotonicity property of ψ_t to $t \in \mathbb{N} \cup \{0\}$. Fix any probability measure P , let Z_1, Z_2, \dots be independent P -distributed random variables, and for each $m \in \mathbb{N}$ denote $A_m = \psi_m(Z_1, \dots, Z_m)$.

We begin with the inequality in (8). The special case of $n = 0$ is directly implied by Lemma 4, so for the remainder of the proof of (8), we suppose $n \geq 1$. The proof proceeds by induction on m . Since $P(A) \leq 1$ for all $A \in \mathcal{A}$, and since $21 + 16 \ln(3) > 38$, the stated bound is trivially satisfied for all $\delta \in (0, 1)$ if $m \leq \max\{38, 21n\}$. Now, as an inductive hypothesis, fix any integer $m > \max\{38, 21n\}$ such that, $\forall m' \in [m-1]$, $\forall \delta \in (0, 1)$, with probability at least $1 - \delta$,

$$P(A_{m'}) \leq \frac{1}{m'} \left(21m' + 16 \ln\left(\frac{3}{\delta}\right) \right).$$

Fix any $\delta \in (0, 1)$ and define

$$N = \left\lfloor \left\{ Z_{\lfloor m/2 \rfloor + 1}, \dots, Z_m \right\} \cap A_{\lfloor m/2 \rfloor} \right\rfloor,$$

and enumerate the elements of $\{Z_{\lfloor m/2 \rfloor + 1}, \dots, Z_m\} \cap A_{\lfloor m/2 \rfloor}$ as $\hat{Z}_1, \dots, \hat{Z}_N$. Also enumerate the elements of $\{Z_{\lfloor m/2 \rfloor + 1}, \dots, Z_m\} \setminus A_{\lfloor m/2 \rfloor}$ as $\hat{Z}'_1, \dots, \hat{Z}'_{\lfloor m/2 \rfloor - N}$. Now note that, by the monotonicity property of ψ_t , we have $A_m \subseteq A_{\lfloor m/2 \rfloor}$. Furthermore, by permutation-invariance of ψ_t , we have that

$$A_m = \psi_m \left(\hat{Z}_1, \dots, \hat{Z}_N, Z_1, \dots, Z_{\lfloor m/2 \rfloor}, \hat{Z}'_1, \dots, \hat{Z}'_{\lfloor m/2 \rfloor - N} \right).$$

Combined with the monotonicity property of ψ_t , this implies that $A_m \subseteq \psi_N \left(\hat{Z}_1, \dots, \hat{Z}_N \right)$. Altogether, we have that

$$A_m \subseteq A_{\lfloor m/2 \rfloor} \cap \psi_N \left(\hat{Z}_1, \dots, \hat{Z}_N \right). \quad (10)$$

Note that $N = \sum_{i=\lfloor m/2 \rfloor + 1}^m \mathbb{1}_{A_{\lfloor m/2 \rfloor}}(Z_i)$ is conditionally Binomial($\lfloor m/2 \rfloor, P(A_{\lfloor m/2 \rfloor})$)-distributed given $Z_1, \dots, Z_{\lfloor m/2 \rfloor}$. In particular, with probability one, if $P(A_{\lfloor m/2 \rfloor}) = 0$, then $N = 0 \leq n$. Otherwise, if $P(A_{\lfloor m/2 \rfloor}) > 0$, then note that $\hat{Z}_1, \dots, \hat{Z}_N$ are conditionally independent and $P(\cdot | A_{\lfloor m/2 \rfloor})$ -distributed given N and $Z_1, \dots, Z_{\lfloor m/2 \rfloor}$. Since ψ_t is a consistent monotone rule, we have that $\psi_N(\hat{Z}_1, \dots, \hat{Z}_N) \cap \{\hat{Z}'_1, \dots, \hat{Z}'_N\} = \emptyset$. We also have, by definition of ψ_N , that $\psi_N(\hat{Z}_1, \dots, \hat{Z}_N) = \phi_{N, n, N}(\hat{Z}_N) \left(\hat{Z}_{i_{N,1}(\hat{Z}_N)}, \dots, \hat{Z}_{i_{N,n}(\hat{Z}_N)}(\hat{Z}_N) \right)$. Thus, applying Lemma 4 (under the conditional distribution given N and $Z_1, \dots, Z_{\lfloor m/2 \rfloor}$), combined with the law of total probability, we have that on an event E_1 of probability at least $1 - \delta/3$, if $N > n$, then

$$P \left(\psi_N \left(\hat{Z}_1, \dots, \hat{Z}_N \right) \middle| A_{\lfloor m/2 \rfloor} \right) \leq \frac{1}{N-n} \left(n \ln \left(\frac{eN}{n} \right) + \ln \left(\frac{3}{\delta} \right) \right).$$

Combined with (10) and monotonicity of measures, this implies that on E_1 , if $N > n$, then

$$\begin{aligned} P(A_m) &\leq P \left(A_{\lfloor m/2 \rfloor} \cap \psi_N \left(\hat{Z}_1, \dots, \hat{Z}_N \right) \right) = P \left(A_{\lfloor m/2 \rfloor} \cap \psi_N \left(\hat{Z}_1, \dots, \hat{Z}_N \right) \middle| A_{\lfloor m/2 \rfloor} \right) \\ &\leq P \left(A_{\lfloor m/2 \rfloor} \right) \frac{1}{N-n} \left(n \ln \left(\frac{eN}{n} \right) + \ln \left(\frac{3}{\delta} \right) \right). \end{aligned}$$

Additionally, again since N is conditionally Binomial($\lfloor m/2 \rfloor, P(A_{\lfloor m/2 \rfloor})$)-distributed given $Z_1, \dots, Z_{\lfloor m/2 \rfloor}$, applying a Chernoff bound (under the conditional distribution given $Z_1, \dots, Z_{\lfloor m/2 \rfloor}$), combined with the law of total probability, we obtain that on an event E_2 of probability at least $1 - \delta/3$, if $P(A_{\lfloor m/2 \rfloor}) \geq \frac{16}{m} \ln \left(\frac{3}{\delta} \right) \geq \frac{8}{\lfloor m/2 \rfloor} \ln \left(\frac{3}{\delta} \right)$, then

$$N \geq P(A_{\lfloor m/2 \rfloor}) \lfloor m/2 \rfloor / 2 \geq P(A_{\lfloor m/2 \rfloor}) m / 4.$$

Also note that if $P(A_m) \geq \frac{1}{m} (21n + 16 \ln \left(\frac{3}{\delta} \right))$, then (10) and monotonicity of probability measures imply $P(A_{\lfloor m/2 \rfloor}) \geq \frac{1}{m} (21n + 16 \ln \left(\frac{3}{\delta} \right))$ as well. In particular, if this occurs with

E_2 , then we have $N \geq P(A_{\lfloor m/2 \rfloor}) m / 4 > 5n$. Thus, by monotonicity of $x \mapsto \text{Log}(x)/x$ for $x > 0$, we have that on $E_1 \cap E_2$, if $P(A_m) \geq \frac{1}{m} (21n + 16 \ln \left(\frac{3}{\delta} \right))$, then

$$\begin{aligned} P(A_m) &< P(A_{\lfloor m/2 \rfloor}) \frac{1}{N - (N/5)} \left(n \text{Log} \left(\frac{eN}{n} \right) + \ln \left(\frac{3}{\delta} \right) \right) \\ &\leq \frac{5}{m} \left(n \text{Log} \left(\frac{eP(A_{\lfloor m/2 \rfloor}) m}{4n} \right) + \ln \left(\frac{3}{\delta} \right) \right). \end{aligned}$$

The inductive hypothesis implies that, on an event E_3 of probability at least $1 - \delta/3$,

$$P(A_{\lfloor m/2 \rfloor}) \leq \frac{1}{\lfloor m/2 \rfloor} \left(21n + 16 \ln \left(\frac{9}{\delta} \right) \right).$$

Since $m \geq 39$, we have $\lfloor m/2 \rfloor \geq (m-2)/2 \geq (37/78)m$, so that the above implies

$$P(A_{\lfloor m/2 \rfloor}) \leq \frac{78}{37m} \left(21n + 16 \ln \left(\frac{9}{\delta} \right) \right).$$

Thus, on $E_1 \cap E_2 \cap E_3$, if $P(A_m) \geq \frac{1}{m} (21n + 16 \ln \left(\frac{3}{\delta} \right))$, then

$$\begin{aligned} P(A_m) &< \frac{5}{m} \left(n \text{Log} \left(\frac{78e}{4 \cdot 37} \left(21 + \frac{16}{n} \ln \left(\frac{9}{\delta} \right) \right) \right) + \ln \left(\frac{3}{\delta} \right) \right) \\ &\leq \frac{5}{m} \left(n \text{Log} \left(\frac{78 \cdot 20}{37 \cdot 11} \left(\frac{21 \cdot 11e}{16 \cdot 5} + \frac{11e}{5} \ln(3) + \frac{11e}{5n} \ln \left(\frac{3}{\delta} \right) \right) \right) + \ln \left(\frac{3}{\delta} \right) \right). \end{aligned}$$

By Lemma 20 in Appendix A, this last expression is at most

$$\begin{aligned} &\frac{5}{m} \left(n \text{Log} \left(\frac{78 \cdot 20}{37 \cdot 11} \left(\frac{21 \cdot 11e}{16 \cdot 5} + \frac{11e}{5} \ln(3) + e \right) \right) + \frac{16}{5} \ln \left(\frac{3}{\delta} \right) \right) < \frac{1}{m} \left(21n + 16 \ln \left(\frac{3}{\delta} \right) \right), \\ &\text{contradicting the condition } P(A_m) \geq \frac{1}{m} (21n + 16 \ln \left(\frac{3}{\delta} \right)). \text{ Therefore, on } E_1 \cap E_2 \cap E_3, \\ &P(A_m) < \frac{1}{m} \left(21n + 16 \ln \left(\frac{3}{\delta} \right) \right). \end{aligned}$$

Noting that, by the union bound, the event $E_1 \cap E_2 \cap E_3$ has probability at least $1 - \delta$, this extends the inductive hypothesis to $m' = m$. By the principle of induction, this completes the proof of the first claim in Theorem 3.

For the bound on the expectation in (9), we note that (as in the proof of Theorem 1), letting $\varepsilon_m = \frac{1}{m} (21n + 16 \ln(3))$, the result just established can be restated as: $\forall \varepsilon > \varepsilon_m$,

$$\mathbb{P}(P(A_m) > \varepsilon) \leq 3 \exp \{ (21/16)n - \varepsilon m / 16 \}.$$

Specifically, this is obtained by setting the bound in (8) equal to ε and solving for δ , the value of which is in $(0, 1)$ for any $\varepsilon > \varepsilon_m$. Furthermore, for any $\varepsilon \leq \varepsilon_m$, we of course still have $\mathbb{P}(P(A_m) > \varepsilon) \leq 1$. Therefore, we have that

$$\begin{aligned} \mathbb{E}[P(A_m)] &= \int_0^\infty \mathbb{P}(P(A_m) > \varepsilon) d\varepsilon \leq \varepsilon_m + \int_{\varepsilon_m}^\infty 3 \exp \{ (21/16)n - \varepsilon m / 16 \} d\varepsilon \\ &= \varepsilon_m + \frac{3 \cdot 16}{m} \exp \{ (21/16)n - \varepsilon_m m / 16 \} = \frac{1}{m} (21n + 16 \ln(3)) + \frac{16}{m} \\ &= \frac{1}{m} (21n + 16 \ln(3e)) \leq \frac{21n + 34}{m}. \quad \blacksquare \end{aligned}$$

3. Application to the Closure Algorithm for Intersection-Closed Classes

One family of concept spaces studied in the learning theory literature, due to their interesting special properties, is the *intersection-closed* classes (Natarajan, 1987; Helmbold, Sloan, and Warmuth, 1990; Haussler, Littlestone, and Warmuth, 1994; Kuhlmann, 1999; Auer and Ortner, 2007). Specifically, the class \mathbb{C} is called *intersection-closed* if the collection of sets $\{(x : h(x) = +1) : h \in \mathbb{C}\}$ is closed under intersections: that is, for every $h, g \in \mathbb{C}$, the classifier $x \mapsto 2\mathbb{1}[h(x) = g(x)] - 1$ is also contained in \mathbb{C} . For instance, the class of conjunctions on $\{0, 1\}^d$, the class of axis-aligned rectangles on \mathbb{R}^d , and the class $\{h : |\{x : h(x) = +1\}| \leq d\}$ of classifiers labeling at most d points positive, are all intersection-closed.

In the context of learning in the realizable case, there is a general learning strategy, called the *Closure* algorithm, designed for learning with intersection-closed concept spaces, which has been a subject of frequent study. Specifically, for any $m \in \mathbb{N} \cup \{0\}$, given any data set $L_m = \{(x_1, y_1), \dots, (x_m, y_m)\} \in (\mathcal{X} \times \mathcal{Y})^m$ with $\mathbb{C}[L_m] \neq \emptyset$, the Closure algorithm $\mathbb{A}(L_m)$ for \mathbb{C} produces the classifier $\hat{h}_m : \mathcal{X} \rightarrow \mathcal{Y}$ with $\{x : \hat{h}_m(x) = +1\} = \bigcap_{h \in \mathbb{C}[L_m]} \{x : h(x) = +1\}$: that is, $\hat{h}_m(x) = +1$ if and only if every $h \in \mathbb{C}$ consistent with L_m (i.e., $\text{er}_{L_m}(h) = 0$) has $h(x) = +1$.¹ Defining $\bar{\mathbb{C}}$ as the set of all classifiers $h : \mathcal{X} \rightarrow \mathcal{Y}$ for which there exists a nonempty $\mathcal{G} \subseteq \mathbb{C}$ with $\{x : h(x) = +1\} = \bigcap_{g \in \mathcal{G}} \{x : g(x) = +1\}$, Auer and Ortner (2007) have argued that $\bar{\mathbb{C}}$ is an intersection-closed concept space containing \mathbb{C} , with $\text{vc}(\bar{\mathbb{C}}) = \text{vc}(\mathbb{C})$. Thus, for $\hat{h}_m = \mathbb{A}(L_m)$ (where \mathbb{A} is the Closure algorithm), since $\hat{h}_m \in \bar{\mathbb{C}}[L_m]$, Lemma 2 immediately implies that, for any $m \in \mathbb{N}$, with probability at least $1 - \delta$, $\text{er}(\hat{h}_m) \lesssim \frac{1}{m}(d \text{Log}(\frac{m}{\delta}) + \text{Log}(\frac{1}{\delta}))$. However, by a more-specialized analysis, Auer and Ortner (2004, 2007) were able to show that, for intersection-closed classes \mathbb{C} , the Closure algorithm in fact achieves $\text{er}(\hat{h}_m) \lesssim \frac{1}{m}(d \text{Log}(d) + \text{Log}(\frac{1}{\delta}))$ with probability at least $1 - \delta$, which is an improvement for large m . They also argued that, for a special subfamily of intersection-closed classes (namely, those with *homogeneous spans*), this bound can be further refined to $\frac{1}{m}(d + \text{Log}(\frac{1}{\delta}))$, which matches (up to constant factors) the lower bound (2). However, they left open the question of whether this refinement is achievable for general intersection-closed concept spaces (by Closure, or any other algorithm).

In the following result, we prove that the Closure algorithm indeed always achieves the optimal bound (up to constant factors) for intersection-closed concept spaces, as a simple consequence of either Theorem 1 or Theorem 3. This fact was very recently also obtained by Darstadt (2015) via a related direct approach; however, we note that the constant factors obtained here are significantly smaller (by roughly a factor of 15.5, for large d).

Theorem 5 *If \mathbb{C} is intersection-closed and \mathbb{A} is the Closure algorithm, then for any $m \in \mathbb{N}$ and $\delta \in (0, 1)$, letting $\hat{h}_m = \mathbb{A}(\{(X_1, f^*(X_1)), \dots, (X_m, f^*(X_m))\})$, with probability at least $1 - \delta$,*

$$\text{er}(\hat{h}_m) \leq \frac{1}{m} \left(21d + 16 \ln \left(\frac{3}{\delta} \right) \right).$$

1. For simplicity, we suppose \mathbb{C} is such that this set $\bigcap_{h \in \mathbb{C}[L_m]} \{x : h(x) = +1\}$ is measurable for every L_m , which is the case for essentially all intersection-closed concept spaces of practical interest.

Furthermore,

$$\mathbb{E} \left[\text{er}(\hat{h}_m) \right] \leq \frac{21d + 34}{m}.$$

Proof For each $t \in \mathbb{N} \cup \{0\}$ and $x_1, \dots, x_t \in \mathcal{X}$, define $\psi_t(x_1, \dots, x_t) = \{x \in \mathcal{X} : \hat{h}_{x_{[t]}}(x) \neq f^*(x)\}$, where $\hat{h}_{x_{[t]}} = \mathbb{A}(\{(x_1, f^*(x_1)), \dots, (x_t, f^*(x_t))\})$. Fix any $x_1, x_2, \dots \in \mathcal{X}$, let $L_t = \{(x_1, f^*(x_1)), \dots, (x_t, f^*(x_t))\}$ for each $t \in \mathbb{N}$, and note that for any $t \in \mathbb{N}$, the classifier $\hat{h}_{x_{[t]}}$ produced by $\mathbb{A}(L_t)$ is consistent with L_t , which implies $\psi_t(x_1, \dots, x_t) \cap \{x_1, \dots, x_t\} = \emptyset$. Furthermore, since $f^* \in \mathbb{C}[L_t]$, we have that $\{x : \hat{h}_{x_{[t]}}(x) = +1\} \subseteq \{x : f^*(x) = +1\}$, which together with the definition of $\hat{h}_{x_{[t]}}$ implies

$$\begin{aligned} \psi_t(x_1, \dots, x_t) &= \{x \in \mathcal{X} : \hat{h}_{x_{[t]}}(x) = -1, f^*(x) = +1\} \\ &= \bigcup_{h \in \mathbb{C}[L_t]} \{x \in \mathcal{X} : h(x) = -1, f^*(x) = +1\} \end{aligned} \quad (11)$$

for every $t \in \mathbb{N}$. Furthermore, for any $t \in \mathbb{N}$, $\mathbb{C}[L_{t+1}] \subseteq \mathbb{C}[L_t]$. Together with monotonicity of the union, these two observations imply

$$\begin{aligned} \psi_{t+1}(x_1, \dots, x_{t+1}) &= \bigcup_{h \in \mathbb{C}[L_{t+1}]} \{x \in \mathcal{X} : h(x) = -1, f^*(x) = +1\} \\ &\subseteq \bigcup_{h \in \mathbb{C}[L_t]} \{x \in \mathcal{X} : h(x) = -1, f^*(x) = +1\} = \psi_t(x_1, \dots, x_t). \end{aligned}$$

Thus, ψ_t defines a consistent monotone rule. Also, since \mathbb{A} always produces a function in $\bar{\mathbb{C}}$, we have $\psi_t(x_1, \dots, x_t) \in \{\{x \in \mathcal{X} : h(x) \neq f^*(x)\} : h \in \bar{\mathbb{C}}\}$ for every $t \in \mathbb{N}$, and it is straightforward to show that the VC dimension of this collection of sets is exactly $\text{vc}(\bar{\mathbb{C}})$ (see Vidyasagar, 2003, Lemma 4.12), which Auer and Ortner (2007) have argued equals d . From this, we can already infer a bound $\frac{4}{m}(17d + 4 \ln(\frac{4}{\delta}))$ via Theorem 1. However, we can refine the constant factors in this bound by noting that ψ_t can also be represented as a consistent monotone sample compression rule of size d , and invoking Theorem 3. The rest of this proof focuses on establishing this fact.

Fix any $t \in \mathbb{N}$. It is well known in the literature (see e.g., Auer and Ortner, 2007, Theorem 1) that there exist $k \in [d] \cup \{0\}$ and distinct $i_1, \dots, i_k \in [d]$ such that $f^*(x_{i_j}) = +1$ for all $j \in [k]$, and letting $L_{i_{[k]}} = \{(x_{i_1}, +1), \dots, (x_{i_k}, +1)\}$, we have $\bigcap_{h \in \mathbb{C}[L_{i_{[k]}}]} \{x : h(x) = +1\} = \bigcap_{h \in \mathbb{C}[L_t]} \{x : h(x) = +1\}$; in particular, letting $\hat{h}_{x_{[i_{[k]}}]} = \mathbb{A}(L_{i_{[k]}})$, this implies $\hat{h}_{x_{[i_{[k]}}]} = \hat{h}_{x_{[t]}}$. This further implies $\psi_t(x_1, \dots, x_t) = \psi_k(x_{i_1}, \dots, x_{i_k})$, so that defining the compression function $(n_k(x_{i_{[k]}}), i_{k,1}(x_{i_{[k]}}), \dots, i_{k,n_k}(x_{i_{[k]}})(x_{i_{[k]}})) = (k, i_1, \dots, i_k)$ as above, for each $x_1, \dots, x_t \in \mathcal{X}$, and defining the reconstruction function $\phi_{t,k}(x'_1, \dots, x'_k) = \psi_k(x'_1, \dots, x'_k)$ for each $t \in \mathbb{N}$, $k' \in [d] \cup \{0\}$, and $x'_1, \dots, x'_k \in \mathcal{X}$, we have that $\psi_t(x_1, \dots, x_t) = \phi_{t,n_k(x_{i_{[k]}})}(x_{i_{k,1}(x_{i_{[k]}}), \dots, x_{i_{k,n_k}(x_{i_{[k]}})}(x_{i_{[k]}}))$ for all $t \in \mathbb{N}$ and $x_1, \dots, x_t \in \mathcal{X}$. Furthermore, since $(x_1, \dots, x_t) \mapsto \mathbb{C}(\{(x_1, f^*(x_1)), \dots, (x_t, f^*(x_t))\})$ is invariant to permutations of its arguments, it follows from (11) that ψ_t is permutation-invariant for every $t \in \mathbb{N}$; this also means that, for the choice of $\phi_{t,k'}$ above, the function $\phi_{t,k'}$ is also permutation-invariant. Altogether, we have that ψ_t is a consistent monotone sample compression rule of size d . Thus,

since $\text{er}(\hat{h}_m) = \mathcal{P}(\psi_m(X_1, \dots, X_m))$ for $m \in \mathbb{N}$, the stated result follows directly from Theorem 3 (with $\mathcal{Z} = \mathcal{X}$, $\mathcal{P} = \mathcal{P}$, and ψ defined as above). ■

4. Application to the CAL Active Learning Algorithm

As another interesting application of Theorem 3, we derive an improved bound on the label complexity of a well-studied active learning algorithm, usually referred to as CAL after its authors Cohn, Atlas, and Laderer (1994). Formally, in the active learning protocol, the learning algorithm \mathbb{A} is given access to the *unlabeled* data sequence X_1, X_2, \dots (or some sufficiently-large finite initial segment thereof), and then sequentially requests to observe the labels: that is, it selects an index t_1 and requests to observe the label $f^*(X_{t_1})$, at which time it is permitted access to $f^*(X_{t_1})$; it may then select another index t_2 and request to observe the label $f^*(X_{t_2})$, is then permitted access to $f^*(X_{t_2})$, and so on. This continues until at most some given number n of labels have been requested (called the *label budget*), at which point the algorithm should halt and return a classifier h ; we denote this as $h = \mathbb{A}(n)$ (leaving the dependence on the unlabeled data implicit, for simplicity). We are then interested in characterizing a sufficient size for the budget n so that, with probability at least $1 - \delta$, $\text{er}(h) \leq \epsilon$; this size is known as the *label complexity* of \mathbb{A} .

The CAL active learning algorithm is based on a very elegant and natural principle: never request a label that can be deduced from information already obtained. CAL is defined solely by this principle, employing no additional criteria in its choice of queries. Specifically, the algorithm proceeds by considering randomly-sampled data points one at a time, and to each it applies the above principle, skipping over the labels that can be deduced, and requesting the labels that cannot be. In favorable scenarios, as the number of label requests grows, the frequency of encountering a sample whose label cannot be deduced should diminish. The key to bounding the label complexity of CAL is to characterize the rate at which this frequency shrinks. To further pursue this discussion with rigor, let us define the *region of disagreement* for any set \mathcal{H} of classifiers:

$$\text{DIS}(\mathcal{H}) = \{x \in \mathcal{X} : \exists h, g \in \mathcal{H} \text{ s.t. } h(x) \neq g(x)\}.$$

Then the CAL active learning algorithm is formally defined as follows.

Algorithm: CAL(n)

0. $m \leftarrow 0$, $t \leftarrow 0$, $V_0 \leftarrow \mathcal{C}$
1. While $t < n$ and $m < 2^n$
 2. $m \leftarrow m + 1$
 3. If $X_m \in \text{DIS}(V_{m-1})$
 4. Request label $Y_m = f^*(X_m)$; let $V_m \leftarrow V_{m-1} \setminus \{(X_m, Y_m)\}$, $t \leftarrow t + 1$
 5. Else $V_m \leftarrow V_{m-1}$
 6. Return any $\hat{h} \in V_m$

This algorithm has several attractive properties. One is that, since it only removes classifiers from V_m upon disagreement with f^* , it maintains the invariant that $f^* \in V_m$.

Another property is that, since it maintains $f^* \in V_m$, and it only refrains from requesting a label if every classifier in V_m agrees on the label (and hence agrees with f^* so that requesting the label would not affect V_m anyway), it maintains the invariant that $V_m = \mathcal{C}[\mathcal{L}_m]$, where $\mathcal{L}_m = \{(X_1, f^*(X_1)), \dots, (X_m, f^*(X_m))\}$.

This algorithm has been studied a great deal in the literature (Cohn, Atlas, and Laderer, 1994; Hanneke, 2009, 2011, 2012, 2014; El-Yaniv and Wiener, 2012; Wiener, Hanneke, and El-Yaniv, 2015), and has inspired an entire genre of active learning algorithms referred to as *disagreement-based* (or sometimes as *mellow*), including several methods possessing desirable properties such as robustness to classification noise (e.g., Balcan, Beygelzimer, and Langford, 2006, 2009; Dasgupta, Hsu, and Monteleoni, 2007; Koltchinskii, 2010; Hanneke and Yang, 2012; Hanneke, 2014). There is a substantial literature studying the label complexity of CAL and other disagreement-based active learning algorithms; the interested reader is referred to the recent survey article of Hanneke (2014) for a thorough discussion of this literature. Much of that literature discusses characterizations of the label complexity in terms of a quantity known as the *disagreement coefficient* (Hanneke, 2007b, 2009). However, Wiener, Hanneke, and El-Yaniv (2015) have recently discovered that a quantity known as the *version space compression set size* (a.k.a. *empirical teaching dimension*) can sometimes provide a smaller bound on the label complexity of CAL. Specifically, the following quantity was introduced in the works of El-Yaniv and Wiener (2010); Hanneke (2007a).

Definition 6 For any $m \in \mathbb{N}$ and $\mathcal{L} \in (\mathcal{X} \times \mathcal{Y})^m$, the version space compression set $\hat{\mathcal{C}}_{\mathcal{L}}$ is a smallest subset of \mathcal{L} satisfying $\mathcal{C}[\hat{\mathcal{C}}_{\mathcal{L}}] = \mathcal{C}[\mathcal{L}]$. We then define $\hat{n}(\mathcal{L}) = |\hat{\mathcal{C}}_{\mathcal{L}}|$, the version space compression set size. In the special case $\mathcal{L} = \mathcal{L}_m$, we abbreviate $\hat{n}_m = \hat{n}(\mathcal{L}_m)$. Also define $\hat{n}_{1:m} = \max_{i \in [m]} \hat{n}_i$, and for any $\delta \in (0, 1)$, define $\hat{n}_m(\delta) = \min\{b \in [m] \cup \{0\} : \mathbb{P}(\hat{n}_m \leq b) \geq 1 - \delta\}$ and $\hat{n}_{1:m}(\delta) = \min\{b \in [m] \cup \{0\} : \mathbb{P}(\hat{n}_m \leq b) \geq 1 - \delta\}$.

The recent work of Wiener, Hanneke, and El-Yaniv (2015) studies this quantity for several concept spaces and distributions, and also identifies general relations between \hat{n}_m and the more-commonly studied disagreement coefficient θ of (Hanneke, 2007b, 2009). Specifically, for any $r > 0$, define $\mathcal{B}(f^*, r) = \{h \in \mathcal{C} : \mathcal{P}(x : h(x) \neq f^*(x)) \leq r\}$. Then the disagreement coefficient is defined, for any $r_0 \geq 0$, as

$$\theta(r_0) = \sup_{r > r_0} \frac{\mathcal{P}(\text{DIS}(\mathcal{B}(f^*, r)))}{r} \vee 1.$$

Both $\hat{n}_{1:m}(\delta)$ and $\theta(r_0)$ are complexity measures dependent on f^* and \mathcal{P} . Wiener, Hanneke, and El-Yaniv (2015) relate them by showing that

$$\theta(1/m) \lesssim \hat{n}_{1:m}(1/20) \vee 1, \quad (12)$$

and for general $\delta \in (0, 1)$,²

$$\hat{n}_{1:m}(\delta) \lesssim \theta(d/m) \left(d \text{Log}(\theta(d/m)) + \text{Log} \left(\frac{\text{Log}(m)}{\delta} \right) \right) \text{Log}(m). \quad (13)$$

² The original claim from Wiener, Hanneke, and El-Yaniv (2015) involved a maximum of minimal $(1 - \delta)$ -confidence bounds on \hat{n}_i over $t \in [m]$, but the same proof can be used to establish this slightly stronger claim.

Wiener, Hanneke, and El-Yaniv (2015) prove that, for $\text{CAL}(n)$ to produce \hat{h} with $\text{er}(\hat{h}) \leq \varepsilon$ with probability at least $1 - \delta$, it suffices to take a budget m of size proportional to

$$\left(\max_{m \in [M(\varepsilon, \delta/2)]} \tilde{n}_m(\delta_m) \text{Log} \left(\frac{m}{\tilde{n}_m(\delta_m)} \right) + \text{Log} \left(\frac{\text{Log}(M(\varepsilon, \delta/2))}{\delta} \right) \right) \text{Log}(M(\varepsilon, \delta/2)), \quad (14)$$

where the values $\delta_m \in (0, 1]$ are such that $\sum_{i=0}^{\lfloor \log_2(M(\varepsilon, \delta/2)) \rfloor} \delta_{2^i} \leq \delta/4$, and $M(\varepsilon, \delta/2)$ is the smallest $m \in \mathbb{N}$ for which $\mathbb{P}(\sup_{h \in \mathbb{C}[\mathcal{L}_m]} \text{er}(h) \leq \varepsilon) \geq 1 - \delta/2$; the quantity $M(\varepsilon, \delta)$ is discussed at length below in Section 5. They also argue that this is essentially a *tight* characterization of the label complexity of CAL, up to logarithmic factors.

The key to obtaining this result is establishing an upper bound on $\mathcal{P}(\text{DIS}(V_m))$ as a function of m , where (as in CAL) $V_m = \mathbb{C}[\mathcal{L}_m]$. One basic observation indicating that $\mathcal{P}(\text{DIS}(V_m))$ can be related to the version space compression set size is that, by exchangeability of the X_i random variables,

$$\begin{aligned} \mathbb{E}[\mathcal{P}(\text{DIS}(V_m))] &= \mathbb{E}[\mathbb{1}[X_{m+1} \in \text{DIS}(\mathbb{C}[\mathcal{L}_m])]] \\ &= \frac{1}{m+1} \sum_{i=1}^{m+1} \mathbb{E}[\mathbb{1}[X_i \in \text{DIS}(\mathbb{C}[\mathcal{L}_{m+1} \setminus \{(X_i, f^*(X_i))\}]]]] \\ &\leq \frac{1}{m+1} \sum_{i=1}^{m+1} \mathbb{E}[\mathbb{1}[(X_i, f^*(X_i)) \in \hat{\mathcal{C}}_{\mathcal{L}_{m+1}}]] = \frac{\mathbb{E}[\hat{n}_{m+1}]}{m+1}, \end{aligned}$$

where the inequality is due to the observation that any $X_i \in \text{DIS}(\mathbb{C}[\mathcal{L}_{m+1} \setminus \{(X_i, f^*(X_i))\}])$ is necessarily in the version space compression set $\hat{\mathcal{C}}_{\mathcal{L}_{m+1}}$, and the last equality is by linearity of the expectation. However, obtaining the bound (14) required a more-involved argument from Wiener, Hanneke, and El-Yaniv (2015), to establish a high-confidence bound on $\mathcal{P}(\text{DIS}(V_m))$, rather than a bound on its expectation. Specifically, by combining a perspective introduced by El-Yaniv and Wiener (2010, 2012), with the observation that $\text{DIS}(V_m)$ may be represented as a sample compression scheme of size \hat{n}_m , and invoking Lemma 4, Wiener, Hanneke, and El-Yaniv (2015) prove that, with probability at least $1 - \delta$,

$$\mathcal{P}(\text{DIS}(V_m)) \lesssim \frac{1}{m} \left(\hat{n}_m \text{Log} \left(\frac{m}{\hat{n}_m} \right) + \text{Log} \left(\frac{1}{\delta} \right) \right). \quad (15)$$

In the present work, we are able to entirely eliminate the factor $\text{Log} \left(\frac{m}{\hat{n}_m} \right)$ from the first term, simply by observing that the region $\text{DIS}(V_m)$ is *monotonic* in m . Specifically, by combining this monotonicity observation with the description of $\text{DIS}(V_m)$ as a compression scheme from Wiener, Hanneke, and El-Yaniv (2015), the refined bound follows from arguments similar to the proof of Theorem 3. Formally, we have the following result.

Theorem 7 For any $m \in \mathbb{N}$ and $\delta \in (0, 1)$, with probability at least $1 - \delta$,

$$\mathcal{P}(\text{DIS}(V_m)) \leq \frac{16}{m} \left(2\hat{n}_{1:m} + \ln \left(\frac{3}{\delta} \right) \right).$$

We should note that, while Theorem 7 indeed eliminates a logarithmic factor compared to (15), this refinement is also accompanied by an increase in the complexity measure, replacing \hat{n}_m with $\hat{n}_{1:m}$. This arises from our proof, since (as in the proof of Theorem 3) the argument relies on $\hat{n}_{1:m}$ being a sample compression set size, not just for the full sample, but also for any prefix of the sample. The effect of this increase is largely benign in this context, since the bound (14) on the label complexity of CAL, derived from (15), involves maximization over the sample size anyway.

Although Theorem 7 follows from the same principles as Theorem 3 (i.e., DIS(V_t) being a consistent monotone rule expressible as a sample compression scheme), it does not quite follow as an immediate consequence of Theorem 3, due fact that the size $\hat{n}_{1:m}$ of the sequence of sample compression schemes can vary based on the specific samples (including their *order*). For this reason, we provide a specialized proof of this result in Appendix B, which follows an argument nearly-identical to that of Theorem 3, with only a few minor changes to account for this variability of $\hat{n}_{1:m}$ using special properties of the sets DIS(V_t).

Based on this result, and following precisely the same arguments as Wiener, Hanneke, and El-Yaniv (2015),³ we arrive at the following bound on the label complexity of CAL. For brevity, we omit the proof, referring the interested reader to the original exposition of Wiener, Hanneke, and El-Yaniv (2015) for the details.

Theorem 8 There is a universal constant $c \in (0, \infty)$ such that, for any $\varepsilon, \delta \in (0, 1)$, for any $n \in \mathbb{N}$ with

$$n \geq c \left(\tilde{n}_{1:M(\varepsilon, \delta/2)}(\delta/4) + \text{Log} \left(\frac{\text{Log}(M(\varepsilon, \delta/2))}{\delta} \right) \right) \text{Log}(M(\varepsilon, \delta/2)),$$

with probability at least $1 - \delta$, the classifier $\hat{h}_n = \text{CAL}(n)$ has $\text{er}(\hat{h}_n) \leq \varepsilon$.

It is also possible to state a distribution-free variant of Theorem 7. Specifically, consider the following definition, from Hanneke and Yang (2015).

Definition 9 The star number \mathfrak{s} is the largest integer s such that there exist distinct points $x_1, \dots, x_s \in \mathcal{X}$ and classifiers $h_0, h_1, \dots, h_s \in \mathbb{C}$ with the property that $\forall i \in [s]$, $\text{DIS}(\{h_0, h_i\}) \cap \{x_1, \dots, x_s\} = \{x_i\}$; if no such largest integer exists, define $\mathfrak{s} = \infty$.

The star number is a natural combinatorial complexity measure, corresponding to the largest possible degree in the data-induced one-inclusion graph. Hanneke and Yang (2015) provide several examples of concept spaces exhibiting a variety of values for the star number (though it should be noted that many commonly-used concept spaces have $\mathfrak{s} = \infty$: e.g., linear separators). As a basic relation, one can easily show that $\mathfrak{s} \geq d$. Hanneke and Yang (2015) also relate the star number to many other complexity measures arising in the learning theory literature, including \hat{n}_m . Specifically, they prove that, for every $m \in \mathbb{N}$ and

3. The only small twist is that we replace $\max_{m \leq M(\varepsilon, \delta/2)} \tilde{n}_m(\delta_m)$ from (14) with $\tilde{n}_{1:M(\varepsilon, \delta/2)}(\delta/4)$. As the purpose of these $\tilde{n}_m(\delta_m)$ values in the original proof is to provide bounds on their respective \hat{n}_m values (which in our context, are $\hat{n}_{1:m}$ values), holding simultaneously for all $m = 2^i \in [M(\varepsilon, \delta/2)]$ with probability at least $1 - \delta/4$, the value $\tilde{n}_{1:M(\varepsilon, \delta/2)}(\delta/4)$ can clearly be used instead. If desired, by a union bound we can of course bound $\tilde{n}_{1:M(\varepsilon, \delta/2)}(\delta/4) \leq \max_{m \in [M(\varepsilon, \delta/2)]} \tilde{n}_m(\delta_m)$, for any sequence δ_m in $(0, 1]$ with $\sum_{m \in [M(\varepsilon, \delta/2)]} \delta_m \leq \delta/4$.

$\mathcal{L} \in (\mathcal{X} \times \mathcal{Y})^m$ with $\mathbb{C}[\mathcal{L}] \neq \emptyset$, $\hat{n}(\mathcal{L}) \leq \mathfrak{s}$, with equality in the worst case (over m and \mathcal{L}). Based on this fact, Theorem 3 implies the following result.

Theorem 10 *For any $m \in \mathbb{N}$ and $\delta \in (0, 1)$, with probability at least $1 - \delta$,*

$$P(\text{DIS}(V_m)) \leq \frac{1}{m} \left(21\mathfrak{s} + 16 \ln \binom{3}{\delta} \right).$$

Proof For every $t \in \mathbb{N}$ and $x_1, \dots, x_t \in \mathcal{X}$, define $\psi_t(x_1, \dots, x_t) = \text{DIS}(\mathbb{C}[\mathcal{L}_{x_t}])$, where $\mathcal{L}_{x_t} = \{(x_1, f^*(x_1)), \dots, (x_t, f^*(x_t))\}$; ψ_t is clearly permutation-invariant, and satisfies $\psi_t(x_1, \dots, x_t) \cap \{x_1, \dots, x_t\} = \emptyset$ (since every $h \in \mathbb{C}[\mathcal{L}_{x_t}]$ agrees with f^* on $\{x_1, \dots, x_t\}$). Furthermore, monotonicity of $\mathcal{L} \mapsto \mathbb{C}[\mathcal{L}]$ and $\mathcal{H} \mapsto \text{DIS}(\mathcal{H})$ imply that any $t \in \mathbb{N}$ and $x_1, \dots, x_{t+1} \in \mathcal{X}$ satisfy $\psi_{t+1}(x_1, \dots, x_{t+1}) \subseteq \psi_t(x_1, \dots, x_t)$, so that ψ_t is a consistent monotone rule. Also define $\phi_{t,k}(x_1, \dots, x_k) = \psi_k(x_1, \dots, x_k)$ for any $k \in [t]$ and $x_1, \dots, x_k \in \mathcal{X}$, and $\phi_{t,0}() = \text{DIS}(\mathbb{C})$. Since ψ_k is permutation-invariant for every $k \in [t]$, so is $\phi_{t,k}$. For any $x_1, \dots, x_t \in \mathcal{X}$, from Definition 6, there exist distinct $i_{t,1}(x_t), \dots, i_{t,\hat{n}(\mathcal{L}_{x_t})}(x_t) \in [t]$ such that $\hat{\mathcal{C}}_{\mathcal{L}_{x_t}} = \{(x_{i_{t,j}}(x_t), f^*(x_{i_{t,j}}(x_t))) : j \in \{1, \dots, \hat{n}(\mathcal{L}_{x_t})\}\}$, and since $\mathbb{C}[\hat{\mathcal{C}}_{\mathcal{L}_{x_t}}] = \mathbb{C}[\mathcal{L}_{x_t}]$, it follows that $\phi_{t,\hat{n}(\mathcal{L}_{x_t})}(x_{i_{t,1}}(x_t), \dots, x_{i_{t,\hat{n}(\mathcal{L}_{x_t})}}(x_t)) = \psi_t(x_1, \dots, x_t)$. Thus, since $\hat{n}(\mathcal{L}_{x_t}) \leq \mathfrak{s}$ for all $t \in \mathbb{N}$ (Hanneke and Yang, 2015), ψ_t is a consistent monotone sample compression rule of size \mathfrak{s} . The result immediately follows by applying Theorem 3 with $\mathcal{Z} = \mathcal{X}$, $\mathcal{P} = \mathcal{P}$, and ψ_t as above. ■

As a final implication for CAL, we can also plug the inequality $\hat{n}(\mathcal{L}) \leq \mathfrak{s}$ into the bound from Theorem 8 to reveal that CAL achieves a label complexity upper-bounded by a value proportional to $\mathfrak{s} \log(M(\varepsilon, \delta/2)) + \log \left(\frac{\log(M(\varepsilon, \delta/2))}{\delta} \right) \log(M(\varepsilon, \delta/2))$.

Remark: In addition to the above applications to active learning, it is worth noting that, combined with the work of El-Yaniv and Wiener (2010), the above results also have implications for the setting of *selective classification*: that is, the setting in which, for each $t \in \mathbb{N}$, given access to $(X_1, f^*(X_1)), \dots, (X_{t-1}, f^*(X_{t-1}))$ and X_t , a learning algorithm is required either to make a prediction \hat{Y}_t for $f^*(X_t)$, or to “abstain” from prediction; after each round t , the algorithm is permitted access to the value $f^*(X_t)$. Then the error rate is the probability the prediction \hat{Y}_t is incorrect (conditioned on $X_{[t-1]}$), given that the algorithm chooses to predict, and the *coverage* is the probability on $X_{[t-1]}$ the algorithm chooses to predict, and the *coverage* is the probability on $X_{[t-1]}$ the algorithm explores an extreme variant, called *perfect selective classification*, in which the algorithm is required to only make predictions that will be correct with *certainly* (i.e., for any data sequence x_1, x_2, \dots , the algorithm will never misclassify a point it chooses to predict for). El-Yaniv and Wiener (2010) find that a selective classification algorithm based on principles analogous to the CAL active learning algorithm obtains the optimal coverage among all perfect selective classification algorithms; the essential strategy is to predict only if $X_t \notin \text{DIS}(V_{t-1})$, taking Y_t as the label agreed-upon by every $h \in V_{t-1}$. In particular, this implies that the optimal coverage rate in perfect selective classification, on round t , is $1 - \mathcal{P}(\text{DIS}(V_{t-1}))$. Thus, combined with Theorem 7 or Theorem 10, we can immediately obtain bounds on the optimal coverage rate for perfect selective classification as well: in particular, this typically refines the bound of

El-Yaniv and Wiener (2010) (and a later refinement by Wiener, Hanneke, and El-Yaniv, 2015) by at least a logarithmic factor (though again, it is not a “pure” improvement, as Theorem 7 uses $\hat{n}_{1:m}$ in place of \hat{n}_m).

5. Application to General Consistent PAC Learners

In general, a *consistent learning algorithm* \mathbb{A} is a learning algorithm such that, for any $m \in \mathbb{N}$ and $L \in (\mathcal{X} \times \mathcal{Y})^m$ with $\mathbb{C}[L] \neq \emptyset$, $\mathbb{A}(L)$ produces a classifier \hat{h} consistent with L (i.e., $\hat{h} \in \mathbb{C}[L]$). In the context of learning in the realizable case, this is equivalent to \mathbb{A} being an instance of the well-studied method of *empirical risk minimization*. The study of general consistent learning algorithms focuses on the quantity $\sup_{h \in V_m} \text{er}(h)$, where $V_m = \mathbb{C}[\mathcal{L}_m]$, as above. It is clear that the error rate achieved by any consistent learning algorithm, given \mathcal{L}_m as input, is at most $\sup_{h \in V_m} \text{er}(h)$. Furthermore, it is not hard to see that, for any given \mathcal{P} and $f^* \in \mathbb{C}$, there exist consistent learning rules obtaining error rates arbitrarily close to $\sup_{h \in V_m} \text{er}(h)$, so that obtaining guarantees on the error rate that hold *generally* for all consistent learning algorithms requires us to bound this value.

Based on Lemma 2 (taking $\mathcal{A} = \{x : h(x) \neq f^*(x)\} : h \in \mathbb{C}\}$), one immediately obtains a classic result (due to Vapnik and Chervonenkis, 1974; Blumer, Ehrenfeucht, Haussler, and Warmuth, 1989), that with probability at least $1 - \delta$,

$$\sup_{h \in V_m} \text{er}(h) \lesssim \frac{1}{m} \left(d \log \left(\frac{m}{d} \right) + \log \left(\frac{1}{\delta} \right) \right).$$

This has been refined by Giné and Koltchinskii (2006),⁴ who argue that, with probability at least $1 - \delta$,

$$\sup_{h \in V_m} \text{er}(h) \lesssim \frac{1}{m} \left(d \log \left(\theta \left(\frac{d}{m} \right) \right) + \log \left(\frac{1}{\delta} \right) \right). \quad (16)$$

In the present work, by combining an argument of Hanneke (2009) with Theorem 7 above, we are able to obtain a new result, which replaces $\theta \left(\frac{d}{m} \right)$ in (16) with $\frac{\mathfrak{s} d}{d}$. Specifically, we have the following result.

Theorem 11 *For any $\delta \in (0, 1)$ and $m \in \mathbb{N}$, with probability at least $1 - \delta$,*

$$\sup_{h \in V_m} \text{er}(h) \leq \frac{8}{m} \left(d \ln \left(\frac{49\mathfrak{s} \hat{n}_{1:m}}{d} + 37 \right) + 8 \ln \left(\frac{6}{\delta} \right) \right).$$

The proof of Theorem 11 follows a similar strategy to the inductive step from the proofs of Theorems 1, 3, and 7. The details are included in Appendix C.

Additionally, since Hanneke and Yang (2015) prove that $\max_{Y \in [m]} \max_{Y \in (\mathcal{X} \times \mathcal{Y})^Y} \hat{n}(L) = \min\{\mathfrak{s}, m\}$, where \mathfrak{s} is the star number, the following new distribution-free bound immediately follows.⁵

4. See also Hanneke (2009), for a simple direct proof of this result.

5. The bound on the expectation follows by integrating the exponential bound on $\mathbb{P}(\sup_{h \in V_m} \text{er}(h) > \varepsilon)$ implied by the first statement in the corollary, as was done, for instance, in the proofs of Theorems 1 and 3. We also note that, by using Theorem 10 in place of Theorem 7 in the proof of Theorem 11, one can obtain mildly better numerical constants in the logarithmic term in this corollary.

Corollary 12 For any $m \in \mathbb{N}$ and $\delta \in (0, 1)$, with probability at least $1 - \delta$,

$$\sup_{h \in V_m} \text{er}(h) \lesssim \frac{1}{m} \left(d \text{Log} \left(\frac{\min\{\mathfrak{s}, m\}}{d} \right) + \text{Log} \left(\frac{1}{\delta} \right) \right).$$

Furthermore,

$$\mathbb{E} \left[\sup_{h \in V_m} \text{er}(h) \right] \lesssim \frac{d}{m} \text{Log} \left(\frac{\min\{\mathfrak{s}, m\}}{d} \right).$$

Let us compare this result to (16). Since Hanneke and Yang (2015) prove that

$$\max_{\mathcal{P}} \max_{f^* \in \mathbb{C}} \theta(r_0) = \min \left\{ \mathfrak{s}, \frac{1}{r_0} \right\},$$

and also (as mentioned) that $\max_{L \in (\mathcal{X} \times \mathcal{Y})} \hat{r}_1(L) = \min\{\mathfrak{s}, m\}$, we see that, at least in some scenarios (i.e., for some choices of \mathcal{P} and f^*), the new bound in Theorem 11 represents an improvement over (16). In particular, the best *distribution-free* bound obtainable from (16) is proportional to

$$\frac{1}{m} \left(d \text{Log} \left(\frac{\min\{d\mathfrak{s}, m\}}{d} \right) + \text{Log} \left(\frac{1}{\delta} \right) \right), \quad (17)$$

which is somewhat larger than the bound stated in Corollary 12 (which has \mathfrak{s} in place of $d\mathfrak{s}$). Also, recalling that Wiener, Hanneke, and El-Yaniv (2015) established that $\theta(1/m) \lesssim \hat{r}_{1,m}(\delta) \lesssim d\theta(d/m) \text{polylog}(m, 1/\delta)$, we should expect that the bound in Theorem 11 is typically not much larger than (16) (and indeed will be smaller in many interesting cases).

5.1 Necessary and Sufficient Conditions for $1/m$ Rates for All Consistent Learners

Corollary 12 provides a sufficient condition for every consistent learning algorithm to achieve error rate with $O(1/m)$ asymptotic dependence on m : namely, $\mathfrak{s} < \infty$. Interestingly, we can show that this condition is in fact also *necessary* for every consistent learner to have a distribution-free bound on the error rate with $O(1/m)$ dependence on m . To be clear, in this context, we only consider m as the asymptotic variable: that is, $m \rightarrow \infty$ while δ and \mathbb{C} (including d and \mathfrak{s}) are held fixed. This result is proven via the following theorem, establishing a worst-case lower bound on $\sup_{h \in V_m} \text{er}(h)$.

Theorem 13 For any $m \in \mathbb{N}$ and $\delta \in (0, 1/100)$, there exists a choice of \mathcal{P} and $f^* \in \mathbb{C}$ such that, with probability greater than δ ,

$$\sup_{h \in V_m} \text{er}(h) \gtrsim \frac{d + \text{Log}(\min\{\mathfrak{s}, m\}) + \text{Log} \left(\frac{1}{\delta} \right)}{m} \wedge 1.$$

Furthermore,

$$\mathbb{E} \left[\sup_{h \in V_m} \text{er}(h) \right] \gtrsim \frac{d + \text{Log}(\min\{\mathfrak{s}, m\})}{m} \wedge 1.$$

Proof Since any $a, b, c \in \mathbb{R}$ have $a + b + c \leq 3 \max\{a, b, c\}$ and $a + b \leq 2 \max\{a, b\}$, it suffices to establish $\frac{d}{m} \wedge 1$, $\frac{\text{Log}(\frac{1}{\delta})}{m} \wedge 1$, and $\frac{\text{Log}(\min\{\mathfrak{s}, m\})}{m}$ as lower bounds separately for the first bound, and $\frac{d}{m} \wedge 1$ and $\frac{\text{Log}(\min\{\mathfrak{s}, m\})}{m}$ as lower bounds separately for the second bound. Lower bounds proportional to $\frac{d}{m} \wedge 1$ (in both bounds) and $\frac{\text{Log}(\frac{1}{\delta})}{m} \wedge 1$ (in the first bound) are known in the literature (Blumer, Ehrenfeucht, Haussler, and Warmuth, 1989; Ehrenfeucht, Haussler, Kearns, and Valiant, 1989; Haussler, Littlestone, and Warmuth, 1994), and in fact hold as lower bounds on the error rate guarantees achievable by *any* learning algorithm.

For the remaining term, note that this term (with appropriately small constant factors) follows immediately from the others if $\mathfrak{s} \leq 56$, so suppose $\mathfrak{s} \geq 57$. Fix any $\varepsilon \in (0, 1/48)$, let $M_\varepsilon = \lfloor \frac{\mathfrak{s} - \varepsilon}{2} \rfloor$, and let $x_1, \dots, x_{\min\{\mathfrak{s}, M_\varepsilon\}} \in \mathcal{X}$ and $h_0, h_1, \dots, h_{\min\{\mathfrak{s}, M_\varepsilon\}} \in \mathbb{C}$ be as in Definition 9. Choose the probability measure \mathcal{P} such that $\mathcal{P}(\{x_i\}) = \varepsilon$ for every $i \in \{2, \dots, \min\{\mathfrak{s}, M_\varepsilon\}\}$, and $\mathcal{P}(\{x_1\}) = 1 - (\min\{\mathfrak{s}, M_\varepsilon\} - 1)\varepsilon \geq 0$. Choose the target function $f^* = h_0$. Then note that, for any $m \in \mathbb{N}$, if $\exists i \in \{2, \dots, \min\{\mathfrak{s}, M_\varepsilon\}\}$ with $x_i \notin \{X_1, \dots, X_m\}$, then $h_i \in V_m$, so that $\sup_{h \in V_m} \text{er}(h) \geq \text{er}(h_i) = \varepsilon$.

Characterizing the probability that $\{x_2, \dots, x_{\min\{\mathfrak{s}, M_\varepsilon\}} \subseteq \{X_1, \dots, X_m\}$ can be approached as an instance of the so-called *coupon collector's problem*. Specifically, let

$$\hat{M} = \min \{m \in \mathbb{N} : \{x_2, \dots, x_{\min\{\mathfrak{s}, M_\varepsilon\}} \subseteq \{X_1, \dots, X_m\}\}.$$

Note that \hat{M} may be represented as a sum $\sum_{k=1}^{\min\{\mathfrak{s}, M_\varepsilon\} - 1} G_k$ of independent geometric random variables $G_k \sim \text{Geometric}(\varepsilon \min\{\mathfrak{s}, M_\varepsilon\} - k)$, where G_k corresponds to the waiting time between encountering the $(k - 1)$ th and k th distinct elements of $\{x_2, \dots, x_{\min\{\mathfrak{s}, M_\varepsilon\}}\}$ in the X_t sequence. A simple calculation reveals that $\mathbb{E}[\hat{M}] = \frac{1}{\varepsilon} H_{\min\{\mathfrak{s}, M_\varepsilon\} - 1}$, where H_t is the t th harmonic number; in particular, $H_t \geq \ln(t)$. Another simple calculation with this sum of independent geometric random variables reveals $\text{Var}(\hat{M}) < \frac{\pi^2}{6\varepsilon^2}$. Thus, Chebyshev's inequality implies that, with probability greater than $1/2$, $\hat{M} \geq \frac{1}{\varepsilon} \ln(\min\{\mathfrak{s}, M_\varepsilon\} - 1) - \frac{\pi}{\sqrt{3}\varepsilon}$. Since $\ln(\min\{\mathfrak{s}, M_\varepsilon\} - 1) \geq \ln(48) > 2\frac{\pi}{\sqrt{3}}$, the right hand side of this inequality is at least $\frac{1}{2\varepsilon} \ln(\min\{\mathfrak{s}, M_\varepsilon\} - 1) = \frac{1}{2\varepsilon} \ln(\min\{\mathfrak{s} - 1, \lfloor \frac{1}{\varepsilon} \rfloor\})$. Altogether, we have that for any $m < \frac{1}{2\varepsilon} \ln(\min\{\mathfrak{s} - 1, \lfloor \frac{1}{\varepsilon} \rfloor\})$, with probability greater than $1/2$, $\sup_{h \in V_m} \text{er}(h) \geq \varepsilon$. By Markov's inequality, this further implies that, for any such m , $\mathbb{E}[\sup_{h \in V_m} \text{er}(h)] > \varepsilon/2$.

For any $m \leq 47$, the $\frac{\text{Log}(\min\{\mathfrak{s}, m\})}{m}$ term in both lower bounds (with appropriately small constant factors) follows from the lower bound proportional to $\frac{d}{m} \wedge 1$, so suppose $m \geq 48$. In particular, for any $c \in (4, \ln(56))$, letting $\varepsilon = \frac{cm}{\ln(\min\{\mathfrak{s} - 1, m\})}$, one can easily verify that $0 < \varepsilon < 1/48$, and $m < \frac{1}{2\varepsilon} \ln(\min\{\mathfrak{s} - 1, \lfloor \frac{1}{\varepsilon} \rfloor\})$. Therefore, with probability greater than $1/2 > \delta$,

$$\sup_{h \in V_m} \text{er}(h) \geq \frac{\ln(\min\{\mathfrak{s} - 1, m\})}{cm},$$

and furthermore,

$$\mathbb{E} \left[\sup_{h \in V_m} \text{er}(h) \right] > \frac{\ln(\min\{\mathfrak{s} - 1, m\})}{2cm}.$$

The result follows by noting $\ln(\min\{\mathfrak{s} - 1, m\}) \geq \ln(\min\{\mathfrak{s}, m\}/2) \geq \ln(\min\{\mathfrak{s}, m\})/2$ for $\mathfrak{s}, m \geq 4$. \blacksquare

Comparing Theorem 13 with Corollary 12, we see that the asymptotic dependences on m are identical, though they differ in their joint dependences on d and m . The precise dependence on both d and m from Corollary 12 can be included in the lower bound of Theorem 13 for certain types of concept spaces \mathbb{C} , but not all: the interested reader is referred to the recent article of Hanneke and Yang (2015) for discussions relevant to this type of gap, and constructions of concept spaces which (one can easily verify) span this gap; that is, for some spaces \mathbb{C} the lower bound is tight, while for other spaces \mathbb{C} the upper bound is tight, up to numerical constant factors.

An immediate corollary of Theorem 13 and Corollary 12 is that $\mathfrak{s} < \infty$ is *necessary and sufficient* for arbitrary consistent learners to achieve $O(1/m)$ rates. Formally, for any $\delta \in (0, 1)$, let $R_m(\delta)$ denote the smallest value such that, for all \mathcal{P} and all $f^* \in \mathbb{C}$, with probability at least $1 - \delta$, $\sup_{h \in V_m} \text{er}(h) \leq R_m(\delta)$. Also let R_m denote the supremum value of $\mathbb{E}[\sup_{h \in V_m} \text{er}(h)]$ over all \mathcal{P} and all $f^* \in \mathbb{C}$. We have the following corollary (which applies to any \mathbb{C} with $0 < d < \infty$).

Corollary 14 $R_m = \Theta\left(\frac{1}{m}\right)$ if and only if $\mathfrak{s} < \infty$, and otherwise $R_m = \Theta\left(\frac{\text{Log}^2(m)}{m}\right)$. Likewise, $\forall \delta \in (0, 1/100)$, $R_m(\delta) = \Theta\left(\frac{1}{m}\right)$ if and only if $\mathfrak{s} < \infty$, and otherwise $R_m(\delta) = \Theta\left(\frac{\text{Log}^2(m)}{m}\right)$.

5.2 Using Subregions Smaller than the Region of Disagreement

In recent work, Zhang and Chandhuri (2014) have proposed a general active learning strategy, which revises the CAL strategy so that the algorithm only requests a label if the corresponding X_m is in a well-chosen *subregion* of $\text{DIS}(V_{m-1})$. This general idea was first explored in the more-specific context of learning linear separators under a uniform distribution by Balcan, Broder, and Zhang (2007) (see also Dasgupta, Kalai, and Monteleoni, 2005, for related arguments). Furthermore, following up on Balcan, Broder, and Zhang (2007), the work of Balcan and Long (2013) has also used this subregion idea to argue that any consistent learning algorithm achieves the optimal sample complexity (up to constants) for the problem of learning linear separators under isotropic log-concave distributions. In this section, we combine the abstract perspective of Zhang and Chandhuri (2014) with our general bounding technique, to generalize the result of Balcan and Long (2013) by expressing a bound holding for arbitrary concept spaces \mathbb{C} , distributions \mathcal{P} , and target functions $f^* \in \mathbb{C}$. First, we need to introduce the following complexity measure $\varphi_c(r_0)$ based on the work of Zhang and Chandhuri (2014). As was true of $\theta(r_0)$ above, this complexity measure $\varphi_c(r_0)$ generally depends on both \mathcal{P} and f^* .

Definition 15 For any nonempty set \mathcal{H} of classifiers, and any $\eta \geq 0$, letting $X \sim \mathcal{P}$, define

$$\Phi(\mathcal{H}, \eta) = \min \left\{ \mathbb{E}[\gamma(X)] : \sup_{h \in \mathcal{H}} \mathbb{E}[\mathbb{1}[h(X) = +1]]\zeta(X) + \mathbb{1}[h(X) = -1]\xi(X) \leq \eta, \right.$$

$$\left. \text{where } \forall x \in \mathcal{X}, \gamma(x) + \zeta(x) + \xi(x) = 1 \text{ and } \gamma(x), \zeta(x), \xi(x) \in [0, 1] \right\}.$$

Then, for any $r_0 \in [0, 1)$ and $c > 1$, define

$$\varphi_c(r_0) = \sup_{r_0 < r \leq 1} \frac{\Phi(\mathbb{B}(f^*, r), r/c)}{r} \vee 1.$$

One can easily observe that, for the optimal choices of γ , ζ , and ξ in the definition of Φ , we have $\gamma(x) = 0$ for (almost every) $x \notin \text{DIS}(\mathcal{H})$. In the special case that γ is binary-valued, the aforementioned well-chosen “subregion” of $\text{DIS}(\mathcal{H})$ corresponds to the set $\{x : \gamma(x) = 1\}$. In general, the definition also allows for $\gamma(x)$ values in between 0 and 1, in which case γ essentially re-weights the conditional distribution $\mathcal{P}(\cdot | \text{DIS}(\mathcal{H}))$.⁶ As an example where this quantity is informative, Zhang and Chandhuri (2014) argue that, for \mathbb{C} the class of homogeneous linear separators in \mathbb{R}^k ($k \in \mathbb{N}$) and \mathcal{P} any isotropic log-concave distribution, $\varphi_c(r_0) \lesssim \text{Log}(c)$ (which follows readily from arguments of Balcan and Long, 2013). Furthermore, they observe that $\varphi_c(r_0) \leq \theta(r_0)$ for any $c \in (1, \infty]$.

Zhang and Chandhuri (2014) propose the above quantities for the purpose of proving a bound on the label complexity of a certain active learning algorithm, inspired both by the work of Balcan, Broder, and Zhang (2007) on active learning with linear separators, and by the connection between selective classification and active learning exposed by El-Yaniv and Wiener (2012). However, since the idea of using well-chosen subregions of $\text{DIS}(V_m)$ in the analysis of consistent learning algorithms lead Balcan and Long (2013) to derive improved sample complexity bounds for these methods in the case of linear separators under isotropic log-concave distributions, and since the corresponding improvements for active learning are reflected in the general results of Zhang and Chandhuri (2014), it is natural to ask whether the sample complexity improvements of Balcan and Long (2013) for that special scenario can also be extended to the general case by incorporating the complexity measure $\varphi_c(r_0)$. Here we provide such an extension. Specifically, following the same basic strategy from Theorem 7, with a few adjustments inspired by Zhang and Chandhuri (2014) to allow us to consider only a *subregion* of $\text{DIS}(V_m)$ in the argument (or more generally, a reweighting of the conditional distribution $\mathcal{P}(\cdot | \text{DIS}(V_m))$), we arrive at the following result. The proof is included in Appendix D.

Theorem 16 For any $\delta \in (0, 1)$ and $m \in \mathbb{N}$, for $c = 16$, with probability at least $1 - \delta$,

$$\sup_{h \in V_m} \text{er}(h) \leq \frac{21}{m} \left(d \ln \left(83\varphi_c \left(\frac{d}{m} \right) \right) + 3 \ln \left(\frac{4}{\delta} \right) \right).$$

In particular, in the special case of \mathbb{C} the space of homogeneous linear separators on \mathbb{R}^k , and \mathcal{P} an isotropic log-concave distribution, Theorem 16 recovers the bound of Balcan and Long (2013) proportional to $\frac{1}{m}(k + \text{Log}(\frac{1}{\delta}))$ as a special case. Furthermore, one can easily construct scenarios (concept spaces \mathbb{C} , distributions \mathcal{P} , and target functions $f^* \in \mathbb{C}$) where $\varphi_c(\frac{d}{m})$ is bounded while $\frac{d_{\text{hom}}}{m} = \frac{d}{m}$ almost surely (e.g., $\mathbb{C} = \{x \mapsto 2\mathbb{1}\{x\} - 1 : t \in \mathbb{R}\}$ the class of impulse functions on \mathbb{R} , and \mathcal{P} uniform on $(0, 1)$), so that Theorem 16 sometimes reflects a significant improvement over Theorem 11.

6. Allowing these more-general values of $\gamma(x)$ typically does not affect the qualitative behavior of the minimal $\mathbb{E}[\gamma(X)]$ value: for instance, we argue in Lemma 24 of Appendix E that the minimal $\mathbb{E}[\gamma(X)]$ value achievable under the additional constraint that $\gamma(x) \in (0, 1)$ is at most $26/74\eta/2$. Thus, we do not lose much by thinking of $\Phi(\mathcal{H}, \eta)$ as describing the measure of a subregion of $\text{DIS}(\mathcal{H})$.

One can easily show that we always have $\varphi_c(r_0) \leq (1 - \frac{1}{c})\theta(r_0)$, so that Theorem 16 is never worse than the bound (16) of Giné and Koltchinskii (2006). However, we argue in Appendix D.1 that $\forall c \geq 2, \forall r_0 \in [0, 1)$,

$$\left(1 - \frac{1}{c}\right) \min \left\{ \mathfrak{s}, \frac{1}{r_0} - \frac{1}{c-1} \right\} \leq \sup_{\mathcal{P}} \sup_{f^* \in \mathbb{C}} \varphi_c(r_0) \leq \left(1 - \frac{1}{c}\right) \min \left\{ \mathfrak{s}, \frac{1}{r_0} \right\}. \quad (18)$$

Thus, at least in some cases, the bound in Theorem 11 is smaller than that in Theorem 16 (as the former leads to Corollary 12 in the worst case, while the latter leads to (17) in the worst case). In fact, if we let $\varphi_c^{01}(r_0)$ be defined identically to $\varphi_c(r_0)$, except that γ is restricted to be $\{0, 1\}$ -valued in Definition 15, then the same argument from Appendix D.1 reveals that, for any $c \geq 4$,

$$\sup_{\mathcal{P}} \sup_{f^* \in \mathbb{C}} \varphi_c^{01}(r_0) = \min \left\{ \mathfrak{s}, \frac{1}{r_0} \right\}.$$

Relation to the Doubling Dimension: To further put Theorem 16 in context, we also note that it is possible to relate $\varphi_c(r_0)$ to the *doubling dimension*. Specifically, the doubling dimension (also known as the local metric entropy) of \mathbb{C} at f^* under \mathcal{P} , denoted $D(r_0)$, is defined as

$$D(r_0) = \max_{r \geq r_0} \log_2 (\mathcal{N}(r/2, \mathbb{B}(f^*, r), \mathcal{P})),$$

for $r_0 > 0$, where $\mathcal{N}(r/2, \mathbb{B}(f^*, r), \mathcal{P})$ is the smallest $n \in \mathbb{N}$ such that there exist classifiers h_1, \dots, h_n for which $\sup_{x \in \mathbb{B}(f^*, r)} \min_{1 \leq i \leq n} \mathcal{P}(x : h_i(x) \neq h_i(x)) \leq r/2$, known as the $(r/2)$ -covering number for $\mathbb{B}(f^*, r)$ under the $L_1(\mathcal{P})$ pseudo-metric. The notion of doubling dimension has been explored in a variety of contexts in the literature (e.g., LeCam, 1973; Yang and Barron, 1999; Gupta, Krauthgamer, and Lee, 2003; Bshouty, Li, and Long, 2009). We always have $D(r_0) \lesssim d\text{Log}(1/r_0)$ (Haussler, 1995), though it can often be smaller than this, and in many interesting contexts, it can even be bounded by an r_0 -invariant value (Bshouty, Li, and Long, 2009). Bshouty, Li, and Long (2009) construct a particular \mathcal{P} -dependent learning rule \mathbb{A} such that, for any $\varepsilon, \delta \in (0, 1)$, and any

$$m \gtrsim \frac{1}{\varepsilon} \left(D(\varepsilon/c) + \text{Log} \left(\frac{1}{\delta} \right) \right), \quad (19)$$

where $c > 0$ is a specific constant, with probability at least $1 - \delta$, the classifier $\hat{h}_m = \mathbb{A}(\mathcal{L}_m)$ satisfies $\text{er}(\hat{h}_m) \leq \varepsilon$. They also establish a weaker bound holding for all consistent learning rules: for any $\varepsilon > 0$, denoting $\varepsilon_0 = \varepsilon \exp \left\{ -\sqrt{\ln(1/\varepsilon)} \right\}$, for any

$$m \gtrsim \frac{1}{\varepsilon} \left(\max \{d, D(\varepsilon_0)\} \sqrt{\text{Log} \left(\frac{1}{\varepsilon} \right)} + \text{Log} \left(\frac{1}{\delta} \right) \right), \quad (20)$$

with probability at least $1 - \delta$, $\sup_{h \in \mathcal{V}_m} \text{er}(h) \leq \varepsilon$.

Hanneke and Yang (2015) have proven that we always have $D(r_0) \lesssim d\text{Log}(\theta(r_0))$, which immediately implies that (19) is never larger than the bound (16) for consistent learning rules (aside from constant factors), though (16) may often offer improvements over the

weaker bound (20). Here we note that a related argument can be used to prove the following bound: for any $r_0 > 0$ and $c \geq 8$,

$$D(r_0) \leq 2d \log_2(96\varphi_c(r_0)). \quad (21)$$

In particular, this implies that the bound (19) is never larger than the bound in Theorem 16 for consistent learning rules (aside from constant factors), though again Theorem 16 may often offer improvements over the weaker bound (20). We also note that, combined with the above mentioned result of Zhang and Chaudhuri (2014) that $\varphi_c(r_0) \lesssim \text{Log}(c)$ for \mathbb{C} the class of homogeneous linear separators in \mathbb{R}^k and \mathcal{P} any isotropic log-concave distribution, (21) immediately implies a bound $D(r_0) \lesssim k$ for the doubling dimension in this scenario (recalling that $d = k$ for this class, from Cover, 1965), which appears to be new to the literature. The proof of (21) is included in Appendix D.2.

6. Learning with Noise

The previous sections demonstrate how variations on the basic technique of Hanneke (2009) lead to refined analyses of certain learning methods, in the *realizable case*, where $\exists f^* \in \mathbb{C}$ with $\text{er}(f^*) = 0$. We can also apply this general technique in the more-general setting of learning with *classification noise*. Specifically, in this setting, there is a *joint* distribution \mathcal{P}_{XY} on $\mathcal{X} \times \mathcal{Y}$, and the error rate of a classifier h is then defined as $\text{er}(h) = \mathbb{P}(h(X) \neq Y)$ for $(X, Y) \sim \mathcal{P}_{XY}$. As above, we denote by \mathcal{P} the marginal distribution $\mathcal{P}_{XY}(\cdot \times \mathcal{Y})$ on \mathcal{X} . We then let $(X_1, Y_1), (X_2, Y_2), \dots$ denote a sequence of independent \mathcal{P}_{XY} -distributed random samples, and denoting $\mathcal{L}_m = \{(X_1, Y_1), \dots, (X_m, Y_m)\}$, we are interested in obtaining bounds on $\text{er}(\hat{h}_m) - \inf_{f \in \mathbb{C}} \text{er}(f)$ (the *excess error rate*), where $\hat{h}_m = \mathbb{A}(\mathcal{L}_m)$ for some learning rule \mathbb{A} . This notation is consistent with the above, which represents the special case in which $\mathbb{P}(Y = f^*(X)|X) = 1$ almost surely (i.e., the *realizable case*). While there are various noise models commonly studied in the literature, for our present discussion, we are primarily interested in two such models.

- For $\beta \in (0, 1/2)$, \mathcal{P}_{XY} satisfies the *β -bounded noise* condition if $\exists h^* \in \mathbb{C}$ such that $\mathbb{P}(Y \neq h^*(X)|X) \leq \beta$ almost surely, where $(X, Y) \sim \mathcal{P}_{XY}$.
- For $a \in [1, \infty)$ and $\alpha \in [0, 1]$, \mathcal{P}_{XY} satisfies the *(a, α) -Bernstein class* condition if, for $h^* = \arg \min_{h \in \mathbb{C}} \text{er}(h)$,⁷ we have $\forall h \in \mathbb{C}, \mathcal{P}(x : h(x) \neq h^*(x)) \leq a(\text{er}(h) - \text{er}(h^*))^\alpha$.

Note that β -bounded noise distributions also satisfy the Bernstein class condition, with $\alpha = 1$ and $a = \frac{1}{1-2\beta}$. These two conditions have been studied extensively in both the passive and active learning literatures (e.g., Massarnau and Tsybakov, 1999; Tsybakov, 2004; Bartlett, Jordan, and McAuliffe, 2006; Massart and Nédélec, 2006; Koltchinskii, 2006; Bartlett and Mendelson, 2006; Giné and Koltchinskii, 2006; Hanneke, 2009, 2011, 2012, 2014; El-Yaniv and Wiener, 2011; Ailon, Begleiter, and Ezra, 2014; Zhang and Chaudhuri, 2014; Hanneke and Yang, 2015). In particular, for passive learning, much of this literature

⁷ For simplicity, we suppose the minimum error rate is achieved in \mathbb{C} . One can easily generalize the condition to the more-general case where the minimum is not necessarily achieved (see e.g., Koltchinskii, 2006), and the results below continue to hold with only minor technical adjustments to the proofs.

focuses on the analysis of *empirical risk minimization*. Specifically, for any $m \in \mathbb{N}$ and $L \in (\mathcal{X} \times \mathcal{Y})^m$, define $\text{ERM}(\mathbb{C}, L) = \{h \in \mathbb{C} : \text{er}_L(h) = \min_{g \in \mathbb{C}} \text{er}_L(g)\}$, the set of *empirical risk minimizers*. Massart and Nédélec (2006) established that, for any \mathcal{P}_{XY} satisfying the (a, α) -Bernstein class condition, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$,

$$\sup_{h \in \text{ERM}(\mathbb{C}, \mathcal{L}_m)} \text{er}(h) - \inf_{h \in \mathbb{C}} \text{er}(h) \lesssim \left(\frac{a \left(d \text{Log} \left(\frac{1}{d} \left(\frac{2m}{d} \right)^{\frac{2-\alpha}{\alpha}} \right) + \text{Log} \left(\frac{1}{\delta} \right) \right)}{m} \right)^{\frac{1}{2-\alpha}}. \quad (22)$$

In the case of β -bounded noise, Giné and Koltchinskii (2006) showed that the logarithmic factor $\text{Log} \left(\frac{m(1-2\beta)^2}{d} \right)$ implied by (22) can be replaced by $\text{Log} \left(\theta \left(\frac{\theta}{m(1-2\beta)^2} \right) \right)$, where the disagreement coefficient $\theta(r_0)$ is defined as above, except with h^* in place of f^* in the definition. Furthermore, applying their arguments to the general case of the (a, α) -Bernstein class condition (see Hanneke and Yang, 2012, for an explicit derivation), one arrives at the fact that, with probability at least $1 - \delta$,

$$\sup_{h \in \text{ERM}(\mathbb{C}, \mathcal{L}_m)} \text{er}(h) - \inf_{h \in \mathbb{C}} \text{er}(h) \lesssim \left(\frac{a \left(d \text{Log} \left(\theta \left(a \left(\frac{2d}{m} \right)^{\frac{2-\alpha}{\alpha}} \right) \right) + \text{Log} \left(\frac{1}{\delta} \right) \right)}{m} \right)^{\frac{1}{2-\alpha}}. \quad (23)$$

Since Hanneke and Yang (2015) have argued that $\theta(r_0) \leq \min \left\{ \mathfrak{s}, \frac{1}{\alpha} \right\}$ (with equality in the worst case), (23) further implies that, with probability at least $1 - \delta$,

$$\sup_{h \in \text{ERM}(\mathbb{C}, \mathcal{L}_m)} \text{er}(h) - \inf_{h \in \mathbb{C}} \text{er}(h) \lesssim \left(\frac{a \left(d \text{Log} \left(\min \left\{ \mathfrak{s}, \frac{1}{\alpha} \left(\frac{2d}{m} \right)^{\frac{2-\alpha}{\alpha}} \right\} \right) + \text{Log} \left(\frac{1}{\delta} \right) \right)}{m} \right)^{\frac{1}{2-\alpha}}. \quad (24)$$

Via the same integration argument used in Corollary 12, this further implies

$$\mathbb{E} \left[\sup_{h \in \text{ERM}(\mathbb{C}, \mathcal{L}_m)} \text{er}(h) - \inf_{h \in \mathbb{C}} \text{er}(h) \right] \lesssim \left(\frac{a d \text{Log} \left(\min \left\{ \mathfrak{s}, \frac{1}{\alpha} \left(\frac{2d}{m} \right)^{\frac{2-\alpha}{\alpha}} \right\} \right)}{m} \right)^{\frac{1}{2-\alpha}}. \quad (25)$$

It is worth noting that the bound (24) does not quite recover the bound of Corollary 12 in the realizable case (corresponding to $a = \alpha = 1$). Specifically, it contains a logarithmic factor $\text{Log} \left(\frac{\min\{d, m\}}{d} \right)$, rather than $\text{Log} \left(\frac{\min\{s, m\}}{d} \right)$. I conjecture that this logarithmic factor in (24) can generally be improved so that, for any a and α , it is bounded by a numerical constant whenever $\mathfrak{s} \lesssim d$. This problem is intimately connected to a conjecture in active learning, proposed by Hanneke and Yang (2015), concerning the joint dependence on \mathfrak{s} and d in the minimax label complexity of active learning under the Bernstein class condition.

6.1 Necessary and Sufficient Conditions for $1/m$ Minimax Rates under Bounded Noise

In the case of bounded noise (where $a = \frac{1}{1-2\beta}$ and $\alpha = 1$), Massart and Nédélec (2006) have shown that for some concept spaces \mathbb{C} , the factor $\text{Log} \left(\frac{m(1-2\beta)^2}{d} \right)$ is present even in

a lower bound on the minimax excess error rate, so that it cannot generally be removed. Raginsky and Rakhlin (2011) further discuss a range of lower bounds on the minimax excess error rate for various spaces \mathbb{C} they construct, where the appropriate factor ranges between $\text{Log} \left(\frac{m(1-2\beta)^2}{d} \right)$ at the highest, to a constant factor at the lowest. The bound in (24) provides a *sufficient* condition for all empirical risk minimization algorithms to achieve excess error rate with $O(1/m)$ asymptotic dependence on m under β -bounded noise: namely $\mathfrak{s} < \infty$. Recall that this condition was both *sufficient and necessary* for $O(1/m)$ error rates to be achievable by every algorithm of this type for all distributions in the realizable case (Corollary 14). It is therefore natural to wonder whether this remains the case for bounded noise as well. In this section, we find this is indeed the case. In fact, following a generalization of the technique of Raginsky and Rakhlin (2011) explored by Hanneke and Yang (2015) for active learning, we are here able to provide a general lower bound on the *minimax* excess error rate of passive learning, expressed in terms of \mathfrak{s} . This immediately implies a corollary that $\mathfrak{s} < \infty$ is both *necessary and sufficient* for the *minimax optimal* bound on the excess error rate to have dependence on m of $\Theta(1/m)$ under bounded noise, and otherwise the minimax optimal bound is $\Theta(\text{Log}(m)/m)$. Note that this is a stronger type of result than that given by Corollary 14, as the lower bounds here apply to *all* learning rules. Formally, we have the following theorem. The proof is included in Appendix E.1.

Theorem 17 *For any $\beta \in (0, 1/2)$, $m \in \mathbb{N}$, and $\delta \in (0, 1/24]$, for any (passive) learning rule Δ , there exists a choice of \mathcal{P}_{XY} satisfying the β -bounded noise condition such that, denoting $\hat{h}_m = \Delta(\mathcal{L}_m)$, with probability greater than δ ,*

$$\text{er}(\hat{h}_m) - \inf_{h \in \mathbb{C}} \text{er}(h) \gtrsim \frac{d + \beta \text{Log} \left(\min \left\{ \mathfrak{s}, (1 - 2\beta)^2 m \right\} \right) + \text{Log} \left(\frac{1}{\delta} \right)}{(1 - 2\beta)m} \wedge (1 - 2\beta).$$

Furthermore,

$$\mathbb{E} \left[\text{er}(\hat{h}_m) \right] - \inf_{h \in \mathbb{C}} \text{er}(h) \gtrsim \frac{d + \beta \text{Log} \left(\min \left\{ \mathfrak{s}, (1 - 2\beta)^2 m \right\} \right)}{(1 - 2\beta)m} \wedge (1 - 2\beta).$$

As was the case in Theorem 13, the joint dependence on d and m in this lower bound does not match that in (24) in the case $\mathfrak{s} = \infty$. One can show that the dependence in this lower bound can be made to nearly match that in (24) for certain specially-constructed spaces \mathbb{C} under bounded noise (Massart and Nédélec, 2006; Raginsky and Rakhlin, 2011; Hanneke and Yang, 2015) (the only gap being that \mathfrak{s} is replaced by \mathfrak{s}/d in (24) to obtain the lower bound); however, there also exist spaces \mathbb{C} where these lower bounds are nearly tight (for β bounded away from 0), so that they cannot be improved in the general case (see Hanneke and Yang, 2015, for construction of spaces \mathbb{C} with arbitrary d and \mathfrak{s} , for which one can show this is the case).

As mentioned above, an immediate corollary of Theorem 17, in combination with (24), is that $\mathfrak{s} < \infty$ is necessary and sufficient for the minimax excess error rate to have $O(1/m)$ dependence on m for bounded noise. Formally, for $m \in \mathbb{N}$, $\beta \in [0, 1/2]$, and $\delta \in (0, 1)$, let $R_m(\delta, \beta)$ denote the smallest value such that there exists a learning rule Δ for which, for all \mathcal{P}_{XY} satisfying the β -bounded noise condition, with probability at least $1 - \delta$, $\text{er}(\Delta(\mathcal{L}_m)) - \inf_{h \in \mathbb{C}} \text{er}(h) \leq R_m(\delta, \beta)$. Also let $\bar{R}_m(\beta)$ denote the smallest value such that

there exists a learning rule \mathbb{A} for which, for all \mathcal{P}_{XY} satisfying the β -bounded noise condition, $\mathbb{E}[\text{er}(\mathbb{A}(\mathcal{L}_m))] - \inf_{h \in \mathbb{C}} \text{er}(h) \leq R_m(\beta)$. We have the following corollary (which applies to any \mathbb{C} with $0 < d < \infty$).

Corollary 18 Fix any $\beta \in (0, 1/2)$. $\bar{R}_m(\beta) = \Theta\left(\frac{1}{m}\right)$ if and only if $\mathfrak{s} < \infty$, and otherwise $\bar{R}_m(\beta) = \Theta\left(\frac{\text{Log}(m)}{m}\right)$. Likewise, $\forall \delta \in (0, 1/24]$, $R_m(\delta, \beta) = \Theta\left(\frac{1}{m}\right)$ if and only if $\mathfrak{s} < \infty$, and otherwise $R_m(\delta, \beta) = \Theta\left(\frac{\text{Log}(m)}{m}\right)$.

Again, note that this is a stronger type of result than Corollary 14 above, which only found $\mathfrak{s} < \infty$ as necessary and sufficient for a *particular family* of learning rules to obtain $O(1/m)$ rates. In contrast, this result applies even to the *minimal optimal* learning rule.

We conclude this section by noting that the technique leading to Theorem 17 appears not to straightforwardly extend to the general (a, α) -Bernstein class condition. Indeed, though one can certainly exhibit specific spaces \mathbb{C} for which the minimax excess risk has $\Theta\left(\frac{\text{Log}(m)}{m}\right)^{\frac{1}{2-\alpha}}$ dependence on m (e.g., impulse functions on \mathbb{R} ; see Hanneke and Yang, 2015, for related discussions), it appears a much more challenging problem to construct general lower bounds describing the range of possible dependences on m . Thus, the more general question of establishing necessary and sufficient conditions for $O\left(\frac{1}{m}\right)^{\frac{1}{2-\alpha}}$ excess error rates under the (a, α) -Bernstein class condition remains open.

6.2 Using Subregions to Achieve Improved Excess Error Bounds

In general, note that plugging into (23) the parameters $a = \alpha = 1$ admitted by the realizable case, (23) recovers the bound (16). Recalling that we were able to refine the bound (16) via techniques from the subregion-based analysis of Zhang and Chaudhuri (2014), yielding Theorem 16 above, it is natural to consider whether we might be able to refine (23) in a similar way. We find that this is indeed the case, though we establish this refinement for a different learning rule (described in Appendix E.2). Letting $c = 128$, for any $r_0 \in [0, 1]$, $a \geq 1$ and $\alpha \in (0, 1]$, define

$$\hat{\varphi}_{a,\alpha}(r_0) = \sup_{h \in \mathbb{C}: r > r_0} \frac{\Phi(\mathbb{B}(h, r), (r/a)^{1/\alpha}/c)}{r} \vee 1.$$

For completeness, also define $\hat{\varphi}_{a,\alpha}(r_0) = 1$ for any $r_0 \geq 1$, $a \geq 1$, and $\alpha \in [0, 1]$. We have the following theorem.

Theorem 19 For any $a \geq 1$ and $\alpha \in (0, 1]$, for any probability measure \mathcal{P} over \mathcal{X} , for any $\delta \in (0, 1)$, there exists a learning rule \mathbb{A} such that, for any \mathcal{P}_{XY} satisfying the (a, α) -Bernstein class condition with marginal distribution \mathcal{P} over \mathcal{X} , for any $m \in \mathbb{N}$, letting $\hat{h}_m = \mathbb{A}(\mathcal{L}_m)$, with probability at least $1 - \delta$,

$$\text{er}(\hat{h}_m) - \inf_{h \in \mathbb{C}} \text{er}(h) \lesssim \left(\frac{a \left(d \text{Log} \left(\hat{\varphi}_{a,\alpha} \left(a \left(\frac{ad}{m} \right)^{\frac{1}{2-\alpha}} \right) \right) + \text{Log} \left(\frac{1}{\delta} \right) \right)^{\frac{1}{2-\alpha}}}{m} \right).$$

The proof is included in Appendix E.2. We should emphasize that the bound in Theorem 19 is established for a particular learning method (described in Appendix E.2), *not* for empirical risk minimization. Thus, whether or not this bound can be established for the general family of empirical risk minimization rules remains an open question. We should also note that $\hat{\varphi}_{a,\alpha}(r_0)$ involves a supremum over $h \in \mathbb{C}$ only so that we may allow the algorithm to explicitly depend on $\hat{\varphi}_{a,\alpha}(r_0)$ (noting that, as stated, Theorem 19 allows \mathcal{P} -dependence in the algorithm). It is conceivable that this dependence on $\hat{\varphi}_{a,\alpha}(r_0)$ in \mathbb{A} can be removed, for instance via a stratification and model selection technique (see e.g., Koltchinskii, 2006), in which case this supremum over h would be replaced by fixing $h = h^*$.

We conclude this section with some basic observations about the bound in Theorem 19. First, in the special case of \mathbb{C} the class of homogeneous linear separators on \mathbb{R}^k and \mathcal{P} any isotropic log-concave distribution, Theorem 19 recovers a bound of Balcan and Long (2013) (established for a closely related method), since a result of Zhang and Chaudhuri (2014) implies $\hat{\varphi}_{a,\alpha}(a\varepsilon^\alpha) \lesssim \text{Log}(a\varepsilon^{\alpha-1})$ in that case. Additionally, we note that a result similar to (24) also generally holds for the method \mathbb{A} from Theorem 19, since (18) implies we always have

$$\frac{\Phi(\mathbb{B}(h, a\varepsilon^\alpha), \varepsilon/c)}{a\varepsilon^\alpha} \leq \left(1 - \frac{1}{ca} \varepsilon^{1-\alpha} \right) \min \left\{ \mathfrak{s}, \frac{1}{a\varepsilon^\alpha} \right\}.$$

Appendix A. A Technical Lemma

The following lemma is useful in the proofs of several of the main results of this paper.⁸

Lemma 20 For any $a, b, c_1 \in [1, \infty)$ and $c_2 \in [0, \infty)$,

$$a \ln \left(c_1 \left(c_2 + \frac{b}{a} \right) \right) \leq a \ln(c_1(c_2 + e)) + \frac{1}{e}b.$$

Proof By subtracting $a \ln(c_1)$ from both sides, we see that it suffices to verify that $a \ln \left(c_2 + \frac{b}{a} \right) \leq a \ln(c_2 + e) + \frac{1}{e}b$. If $\frac{b}{a} \leq e$, then monotonicity of $\ln(\cdot)$ implies

$$a \ln \left(c_2 + \frac{b}{a} \right) \leq a \ln(c_2 + e),$$

which is clearly no greater than $a \ln(c_2 + e) + \frac{1}{e}b$. On the other hand, if $\frac{b}{a} > e$, then

$$a \ln \left(c_2 + \frac{b}{a} \right) \leq a \ln \left(\max\{c_2, 2\} \frac{b}{a} \right) = a \ln(\max\{c_2, 2\}) + a \ln \left(\frac{b}{a} \right).$$

The first term in the rightmost expression is at most $a \ln(c_2 + 2) \leq a \ln(c_2 + e)$. The second term in the rightmost expression can be rewritten as $b \frac{\ln(b/a)}{b/a}$. Since $x \mapsto \ln(x)/x$ is nonincreasing on (e, ∞) , in the case $\frac{b}{a} > e$ this is at most $\frac{1}{e}b$. Together, we have that

$$a \ln \left(c_2 + \frac{b}{a} \right) \leq a \ln(c_2 + e) + \frac{1}{e}b$$

in this case as well. \blacksquare

⁸ This lemma and proof also appear in a sibling paper (Hanneke, 2016).

Appendix B. Proof of Theorem 7

Here we present the proof of Theorem 7.

Proof of Theorem 7 The structure of the proof is nearly identical to that of Theorem 3, with only a few small changes to account for the fact that $\hat{n}_{1:m}$ depends on the specific samples, and in particular, on the order of the samples.

The proof proceeds by induction on m . Since $\mathcal{P}(\text{DIS}(V_m)) \leq 1$ always, the stated bound is trivially satisfied for all $\delta \in (0, 1)$ if $m \leq 16$. Now, as an inductive hypothesis, fix any integer $m \geq 17$ such that, $\forall \delta \in (0, 1)$, with probability at least $1 - \delta$,

$$\mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor})) \leq \frac{16}{\lfloor m/2 \rfloor} \left(2^{\hat{n}_{1:\lfloor m/2 \rfloor}} + \ln \left(\frac{3}{\delta} \right) \right).$$

Fix any $\delta \in (0, 1)$. Define

$$N = \left| \{X_{\lfloor m/2 \rfloor+1}, \dots, X_m\} \cap \text{DIS}(V_{\lfloor m/2 \rfloor}) \right|,$$

and enumerate the elements of $\{X_{\lfloor m/2 \rfloor+1}, \dots, X_m\} \cap \text{DIS}(V_{\lfloor m/2 \rfloor})$ as $\hat{X}_1, \dots, \hat{X}_N$. Let $\mathcal{L}_t = \{(X_1, f^*(X_1)), \dots, (X_t, f^*(X_t))\}$ for every $t \in [m]$, and $\hat{n}'_m = \left| \hat{C}_{\mathcal{L}_m} \setminus \mathcal{L}_{\lfloor m/2 \rfloor} \right|$, and enumerate as $i'_1, \dots, i'_{\hat{n}'_m}$ the indices $i \in \{\lfloor m/2 \rfloor+1, \dots, m\}$ with $(X_i, f^*(X_i)) \in \hat{C}_{\mathcal{L}_m} \setminus \mathcal{L}_{\lfloor m/2 \rfloor}$. In particular, note that $\hat{n}'_m \leq \hat{n}_m$ and $\hat{C}_{\mathcal{L}_m} \subseteq \mathcal{L}_{\lfloor m/2 \rfloor} \cup \{(X_{i'_1}, f^*(X_{i'_1})), \dots, (X_{i'_{\hat{n}'_m}}, f^*(X_{i'_{\hat{n}'_m}}))\}$, so that

$$\mathcal{C} \left[\mathcal{L}_{\lfloor m/2 \rfloor} \cup \{(X_{i'_1}, f^*(X_{i'_1})), \dots, (X_{i'_{\hat{n}'_m}}, f^*(X_{i'_{\hat{n}'_m}}))\} \right] = V_m.$$

Next, let $\hat{n}''_m = \left| \{j \in [\hat{n}'_m] : X_{i'_j} \in \text{DIS}(V_{\lfloor m/2 \rfloor})\} \right|$, and enumerate as $i''_1, \dots, i''_{\hat{n}''_m}$ the indices $i \in [N]$ such that $(\hat{X}_i, f^*(\hat{X}_i)) \in \{(X_{i'_1}, f^*(X_{i'_1})), \dots, (X_{i'_{\hat{n}'_m}}, f^*(X_{i'_{\hat{n}'_m}}))\}$. Note that, since every $j \in [\hat{n}'_m]$ with $X_{i'_j} \notin \text{DIS}(V_{\lfloor m/2 \rfloor})$ has $h(X_{i'_j}) = f^*(X_{i'_j})$ for every $h \in \mathcal{C}[\mathcal{L}_{\lfloor m/2 \rfloor} \cup \{(X_{i'_1}, f^*(X_{i'_1})), \dots, (X_{i'_{\hat{n}'_m}}, f^*(X_{i'_{\hat{n}'_m}}))\}]$ (by definition of DIS and monotonicity of $\mathcal{L} \mapsto \mathcal{C}[\mathcal{L}]$), we have

$$\begin{aligned} \mathcal{C}[\mathcal{L}_{\lfloor m/2 \rfloor} \cup \{(X_{i''_1}, f^*(X_{i''_1})), \dots, (X_{i''_{\hat{n}''_m}}, f^*(X_{i''_{\hat{n}''_m}}))\}] \\ = \mathcal{C} \left[\mathcal{L}_{\lfloor m/2 \rfloor} \cup \{(X_{i'_1}, f^*(X_{i'_1})), \dots, (X_{i'_{\hat{n}'_m}}, f^*(X_{i'_{\hat{n}'_m}}))\} \right] = V_m, \end{aligned}$$

so that $\text{DIS}(V_m)$ may be expressed as a fixed function of $X_1, \dots, X_{\lfloor m/2 \rfloor}$ and $\hat{X}_{i''_1}, \dots, \hat{X}_{i''_{\hat{n}''_m}}$. Furthermore, note that the set $\text{DIS} \left(\mathcal{C}[\mathcal{L}_{\lfloor m/2 \rfloor} \cup \{(X_{i''_1}, f^*(X_{i''_1})), \dots, (X_{i''_{\hat{n}''_m}}, f^*(X_{i''_{\hat{n}''_m}}))\}] \right)$ is invariant to permutations of the $i''_1, \dots, i''_{\hat{n}''_m}$ indices.

Now note that N is conditionally Binomial($\lfloor m/2 \rfloor, \mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor}))$)-distributed given $X_1, \dots, X_{\lfloor m/2 \rfloor}$. In particular, with probability one, if $\mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor})) = 0$, then $N = 0$. Otherwise, if $\mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor})) > 0$, then note that $\hat{X}_1, \dots, \hat{X}_N$ are conditionally independent and $\mathcal{P}(\cdot | \text{DIS}(V_{\lfloor m/2 \rfloor}))$ -distributed given $X_1, \dots, X_{\lfloor m/2 \rfloor}$ and N . Thus, since $\text{DIS}(V_m) \cap \{X_1, \dots, X_N\} = \emptyset$ (since every $h \in V_m$ agrees with f^* on X_1, \dots, X_m), combining the above with Lemma 4 (applied under the conditional distribution given $X_1, \dots, X_{\lfloor m/2 \rfloor}$

and N), combined with the law of total probability, implies that for every $n \in [m] \cup \{0\}$, with probability at least $1 - \delta/(n+3)^2$, if $\hat{n}''_m = n$ and $N > n$, then

$$\mathcal{P}(\text{DIS}(V_m) | \text{DIS}(V_{\lfloor m/2 \rfloor})) \leq \frac{1}{N-n} \left(n \text{Log} \left(\frac{eN}{n} \right) + \text{Log} \left(\frac{(n+3)^2}{\delta} \right) \right).$$

By a union bound, this holds simultaneously for all $n \in [m] \cup \{0\}$ on an event E_1 of probability at least $1 - \sum_{n=0}^m \frac{\delta}{(n+3)^2} > 1 - \frac{2}{3}\delta$. In particular, since the right hand side of the above inequality is nondecreasing in n , and $\hat{n}''_m \leq \hat{n}_m$, and since $\text{DIS}(V_m) \subseteq \text{DIS}(V_{\lfloor m/2 \rfloor})$, we have that on E_1 , if $N > \hat{n}_m$, then

$$\mathcal{P}(\text{DIS}(V_m)) \leq \mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor})) \frac{1}{N - \hat{n}_m} \left(\hat{n}_m \text{Log} \left(\frac{eN}{\hat{n}_m} \right) + \text{Log} \left(\frac{(\hat{n}_m + 3)^2}{\delta} \right) \right).$$

Next, again since N is conditionally Binomial($\lfloor m/2 \rfloor, \mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor}))$)-distributed given $X_1, \dots, X_{\lfloor m/2 \rfloor}$, by a Chernoff bound (applied under the conditional distribution given $X_1, \dots, X_{\lfloor m/2 \rfloor}$), combined with the law of total probability, we obtain that on an event E_2 of probability at least $1 - \delta/3$, if $\mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor})) \geq \frac{16}{m} \ln \left(\frac{3}{\delta} \right) \geq \frac{8}{\lfloor m/2 \rfloor} \ln \left(\frac{3}{\delta} \right)$, then

$$N \geq \mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor})) \lfloor m/2 \rfloor / 2 \geq \mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor})) m/4.$$

Also note that if $\mathcal{P}(\text{DIS}(V_m)) \geq \frac{16}{m} (2\hat{n}_m + \ln \left(\frac{3}{\delta} \right))$, then monotonicity of $t \mapsto \text{DIS}(V_t)$ and monotonicity of probability measures imply $\mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor})) \geq \frac{16}{m} (2\hat{n}_m + \ln \left(\frac{3}{\delta} \right))$ as well. In particular, if this occurs with E_2 , then we have $N \geq \mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor})) m/4 > 8\hat{n}_m$. Thus, by monotonicity of $x \mapsto \text{Log}(x)/x$ for $x > 0$, we have that on $E_1 \cap E_2$, if $\mathcal{P}(\text{DIS}(V_m)) \geq \frac{16}{m} (2\hat{n}_m + \ln \left(\frac{3}{\delta} \right))$, then

$$\begin{aligned} \mathcal{P}(\text{DIS}(V_m)) &< \mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor})) \frac{8}{7N} \left(\hat{n}_m \text{Log} \left(\frac{eN}{\hat{n}_m} \right) + \ln \left(\frac{(\hat{n}_m + 3)^2}{\delta} \right) \right) \\ &\leq \frac{32}{7m} \left(\hat{n}_m \text{Log} \left(\frac{e \mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor})) m}{4\hat{n}_m} \right) + \ln \left(\frac{(\hat{n}_m + 3)^2}{\delta} \right) \right). \end{aligned}$$

The inductive hypothesis implies that, on an event E_3 of probability at least $1 - \delta/4$,

$$\mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor})) \leq \frac{16}{\lfloor m/2 \rfloor} \left(2^{\hat{n}_{1:\lfloor m/2 \rfloor}} + \ln \left(\frac{12}{\delta} \right) \right).$$

Since $m \geq 17$, we have $\lfloor m/2 \rfloor \geq (m-2)/2 \geq (15/34)m$, so that the above implies

$$\mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor})) \leq \frac{544}{15m} \left(2^{2\hat{n}_{1:\lfloor m/2 \rfloor}} + \ln \left(\frac{12}{\delta} \right) \right).$$

Thus, on $E_1 \cap E_2 \cap E_3$, if $\mathcal{P}(\text{DIS}(V_m)) \geq \frac{16}{m} (2\hat{n}_m + \ln \left(\frac{3}{\delta} \right))$, then

$$\begin{aligned} \mathcal{P}(\text{DIS}(V_m)) &< \frac{32}{7m} \left(\hat{n}_m \text{Log} \left(\frac{136e}{15} \left(2^{\frac{\hat{n}_{1:\lfloor m/2 \rfloor}}{\hat{n}_m}} + \frac{1}{\hat{n}_m} \ln \left(\frac{12}{\delta} \right) \right) \right) + \ln \left(\frac{(\hat{n}_m + 3)^2}{\delta} \right) \right) \\ &\leq \frac{32}{7m} \left(\hat{n}_{1:m} \text{Log} \left(\frac{136e}{15} \left(2 + \frac{1}{\hat{n}_{1:m}} \ln(4) + \frac{1}{\hat{n}_{1:m}} \ln \left(\frac{3}{\delta} \right) \right) \right) + \ln \left(\frac{(\hat{n}_{1:m} + 3)^2}{\delta} \right) \right). \end{aligned} \quad (26)$$

By straightforward calculus, one can easily verify that, when $\hat{n}_{1:m} \in \{0, 1\}$, the right hand side of (26) is at most $\frac{16}{m}(2\hat{n}_{1:m} + \ln(\frac{3}{\delta}))$ (recalling our conventions that $1/0 = \infty$ and $0\text{Log}(\infty) = 0$). Otherwise, supposing $\hat{n}_{1:m} \geq 2$, Lemma 20 in Appendix A (applied with $b = \frac{56}{2} \ln(3/\delta)$) implies the right hand side of (26) is at most

$$\begin{aligned} & \frac{32}{7m} \left(\hat{n}_{1:m} \text{Log} \left(\frac{136e}{15} \left(2 + \ln(4) + \frac{2}{5} \right) \right) + 2 \ln(\hat{n}_{1:m} + 3) + \frac{7}{2} \ln \left(\frac{3}{\delta} \right) \right) \\ & \leq \frac{32}{7m} \left(5\hat{n}_{1:m} + 2 \ln(\hat{n}_{1:m} + 3) + \frac{7}{2} \ln \left(\frac{3}{\delta} \right) \right). \end{aligned}$$

Since $5x + 2 \ln(x + 3) < 7x$ for any $x \geq 2$, the above is at most

$$\frac{32}{7m} \left(7\hat{n}_{1:m} + \frac{7}{2} \ln \left(\frac{3}{\delta} \right) \right) = \frac{16}{m} \left(2\hat{n}_{1:m} + \ln \left(\frac{3}{\delta} \right) \right).$$

Thus, since $\frac{16}{m}(2\hat{n}_m + \ln(\frac{3}{\delta})) \leq \frac{16}{m}(2\hat{n}_{1:m} + \ln(\frac{3}{\delta}))$ as well, in either case we have that, on $E_1 \cap E_2 \cap E_3$,

$$\mathcal{P}(\text{DIS}(V_m)) \leq \frac{16}{m} \left(2\hat{n}_{1:m} + \ln \left(\frac{3}{\delta} \right) \right).$$

Noting that, by a union bound, the event $E_1 \cap E_2 \cap E_3$ has probability at least $1 - \frac{2}{3}\delta - \frac{1}{3}\delta - \frac{1}{4}\delta > 1 - \delta$, this extends the result to m . By the principle of induction, this completes the proof of Theorem 7. \blacksquare

Appendix C. Proof of Theorem 11

We now present the proof of Theorem 11.

Proof of Theorem 11 The result trivially holds for $m \leq \lfloor 8(\ln(37) + 8 \ln(6)) \rfloor = 143$, so suppose $m \geq 144$. Let $N = \{X_{\lfloor m/2 \rfloor + 1}, \dots, X_m\} \cap \text{DIS}(V_{\lfloor m/2 \rfloor})$ and enumerate the elements of $\{X_{\lfloor m/2 \rfloor + 1}, \dots, X_m\} \cap \text{DIS}(V_{\lfloor m/2 \rfloor})$ as $\hat{X}_1, \dots, \hat{X}_N$. Note that N is conditionally Binomial($\lfloor m/2 \rfloor, \mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor}))$)-distributed given $X_1, \dots, X_{\lfloor m/2 \rfloor}$. In particular, with probability one, if $\mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor})) = 0$, then $N = 0$. Otherwise, if $\mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor})) > 0$, then note that $\hat{X}_1, \dots, \hat{X}_N$ are conditionally independent $\mathcal{P}(\cdot | \text{DIS}(V_{\lfloor m/2 \rfloor}))$ -distributed random variables, given $X_1, \dots, X_{\lfloor m/2 \rfloor}$ and N . Also, note that (one can easily show) $\text{ve}(\{x : h(x) \neq f^*(x) : h \in \mathbb{C}\}) = d$. Together with Lemma 2 (applied under the conditional distribution given $X_1, \dots, X_{\lfloor m/2 \rfloor}$ and N), combined with the law of total probability, these observations imply that there is an event H_1 of probability at least $1 - \delta/3$, on which, if $N > 0$, then $\forall h \in V_m$,

$$\mathcal{P}(\text{DIS}(\{h, f^*\}) | \text{DIS}(V_{\lfloor m/2 \rfloor})) \leq \frac{2}{N} \left(d \text{Log}_2 \left(\frac{2eN}{d} \right) + \text{Log}_2 \left(\frac{6}{\delta} \right) \right).$$

In particular, noting that $\forall h \in V_m$, since $f^* \in V_m$ as well, $\text{DIS}(\{h, f^*\}) \subseteq \text{DIS}(V_m) \subseteq \text{DIS}(V_{\lfloor m/2 \rfloor})$, we have that on H_1 , $\forall h \in V_m$,

$$\begin{aligned} \text{er}(h) &= \mathcal{P}(\text{DIS}(\{h, f^*\})) = \mathcal{P}(\text{DIS}(\{h, f^*\}) | \text{DIS}(V_{\lfloor m/2 \rfloor})) \mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor})) \\ &\leq \mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor})) \frac{2}{N} \left(d \text{Log}_2 \left(\frac{2eN}{d} \right) + \text{Log}_2 \left(\frac{6}{\delta} \right) \right). \end{aligned}$$

Next, again since N is conditionally Binomial($\lfloor m/2 \rfloor, \mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor}))$)-distributed given $X_1, \dots, X_{\lfloor m/2 \rfloor}$, by a Chernoff bound (applied under the conditional distribution given $X_1, \dots, X_{\lfloor m/2 \rfloor}$), combined with the law of total probability, there is an event H_2 of probability at least $1 - \delta/3$, on which, if $\mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor})) \geq \frac{32}{\lfloor m/2 \rfloor} \ln(\frac{3}{\delta})$, then

$$N \geq (3/4) \mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor})) \lfloor m/2 \rfloor \geq (3/8) \mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor})) m,$$

which (by $\text{Log}_2(x) \leq \text{Log}(x)/\ln(2)$ and monotonicity of $x \mapsto \text{Log}(x)/x$ for $x > 0$) implies

$$\begin{aligned} & \frac{2}{N} \left(d \text{Log}_2 \left(\frac{2eN}{d} \right) + \text{Log}_2 \left(\frac{6}{\delta} \right) \right) \\ & \leq \frac{2}{3 \ln(2) \mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor})) m} \left(d \ln \left(\frac{3e \mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor})) m}{4d} \right) + \ln \left(\frac{6}{\delta} \right) \right). \end{aligned}$$

Also, by Theorem 7, on an event H_3 of probability at least $1 - \delta/3$,

$$\mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor})) \leq \frac{16}{\lfloor m/2 \rfloor} \left(2\hat{n}_{1:\lfloor m/2 \rfloor} + \ln \left(\frac{9}{\delta} \right) \right).$$

Together with the facts that $\frac{16}{3 \ln(2)} < 8$ and $\lfloor m/2 \rfloor \geq \frac{m-2}{m} \frac{m}{2} \geq \frac{142}{144} \frac{m}{2}$, we have that, on $H_1 \cap H_2 \cap H_3$, if $\mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor})) \geq \frac{32}{\lfloor m/2 \rfloor} \ln(\frac{3}{\delta})$, then

$$\begin{aligned} \sup_{h \in V_m} \text{er}(h) &\leq \frac{8}{m} \left(d \ln \left(\frac{e24 \cdot 144 (2\hat{n}_{1:\lfloor m/2 \rfloor} + \ln(9/\delta))}{142d} \right) + \ln \left(\frac{6}{\delta} \right) \right) \\ &= \frac{8}{m} \left(d \ln \left(\frac{24 \cdot 144 \left(\frac{14e\hat{n}_{1:\lfloor m/2 \rfloor} + 7e \ln(3/2)}{d} + \frac{7e \ln(6/\delta)}{d} \right) \right) + \ln \left(\frac{6}{\delta} \right) \right). \end{aligned}$$

By Lemma 20 in Appendix A, this last expression is at most

$$\begin{aligned} & \frac{8}{m} \left(d \ln \left(\frac{24 \cdot 144 \left(\frac{14e\hat{n}_{1:\lfloor m/2 \rfloor} + 7e \ln(3/2)}{d} + e \right) \right) + 8 \ln \left(\frac{6}{\delta} \right) \right) \\ & \leq \frac{8}{m} \left(d \ln \left(\left(\frac{49e\hat{n}_{1:\lfloor m/2 \rfloor}}{d} + 37 \right) \right) + 8 \ln \left(\frac{6}{\delta} \right) \right). \end{aligned}$$

Furthermore, since $\text{DIS}(\{h, f^*\}) \subseteq \text{DIS}(V_{\lfloor m/2 \rfloor})$ for every $h \in V_m$, if $\mathcal{P}(\text{DIS}(V_{\lfloor m/2 \rfloor})) < \frac{32}{\lfloor m/2 \rfloor} \ln(\frac{3}{\delta}) \leq \frac{64}{m} \ln(\frac{3}{\delta})$, then

$$\sup_{h \in V_m} \text{er}(h) < \frac{64}{m} \ln \left(\frac{3}{\delta} \right) < \frac{8}{m} \left(d \text{Log} \left(\frac{49e\hat{n}_{1:\lfloor m/2 \rfloor}}{d} + 37 \right) + 8 \ln \left(\frac{6}{\delta} \right) \right).$$

Thus, in either case, we have that, on $H_1 \cap H_2 \cap H_3$,

$$\sup_{h \in V_m} \text{er}(h) \leq \frac{8}{m} \left(d \text{Log} \left(\frac{49e\hat{n}_{1:\lfloor m/2 \rfloor}}{d} + 37 \right) + 8 \ln \left(\frac{6}{\delta} \right) \right).$$

The proof is completed by noting that $\hat{n}_{1:\lfloor m/2 \rfloor} \leq \hat{n}_{1:m}$, and that, by the union bound, the event $H_1 \cap H_2 \cap H_3$ has probability at least $1 - \delta$. \blacksquare

Appendix D. Proof of Theorem 16

We now present the proof of Theorem 16.

Proof of Theorem 16 The proof essentially combines the argument of Hanneke (2009) (which proves (16)) with the subsample-based ideas of Zhang and Chaudhuri (2014). Fix $c = 16$. The proof proceeds by induction on m . Since $\sup_{h \in \mathcal{C}} \text{er}(h) \leq 1$, the result trivially holds for $m < 21(d \ln(83) + 3 \ln(4))$. Now, as an inductive hypothesis, fix any $m \geq 21(d \ln(83) + 3 \ln(4))$ such that $\forall m' \in [m-1]$, $\forall \delta \in (0, 1)$, with probability at least $1 - \delta$,

$$\sup_{h \in \mathcal{V}_{m'}} \text{er}(h) \leq \frac{21}{m'} \left(d \text{Log} \left(83 \varphi_c \left(\frac{d}{m'} \right) \right) + 3 \text{Log} \left(\frac{4}{\delta} \right) \right).$$

Fix any $\delta \in (0, 1)$ and $\eta \in [0, 1]$. Let γ^*, ζ^*, ξ^* be the functions γ, ζ , and ξ from Definition 13 (each mapping $\mathcal{X} \rightarrow [0, 1]$) with $\gamma^*(x) + \zeta^*(x) + \xi^*(x) = 1$ for all $x \in \mathcal{X}$, and $\mathbb{E}[\gamma^*(X)]X_1, \dots, X_{\lfloor m/2 \rfloor}$ minimal subject to

$$\sup_{h \in \mathcal{V}_{\lfloor m/2 \rfloor}} \mathbb{E}[\mathbb{1}[h(X) = +1][\zeta^*(X) + \mathbb{1}[h(X) = -1]\xi^*(X)]X_1, \dots, X_{\lfloor m/2 \rfloor} \leq \eta,$$

where $X \sim \mathcal{P}$ is independent of X_1, X_2, \dots .⁹ Note that these functions are themselves random, having dependence on $X_1, \dots, X_{\lfloor m/2 \rfloor}$. In particular, $\mathbb{E}[\gamma^*(X)]X_1, \dots, X_{\lfloor m/2 \rfloor} = \Phi(V_{\lfloor m/2 \rfloor}, \eta)$.

Let $\Gamma_{\lfloor m/2 \rfloor+1}, \dots, \Gamma_m$ be conditionally independent random variables given X_1, \dots, X_m , with Γ_i having conditional distribution $\text{Bernoulli}(\gamma^*(X_i))$ given X_1, \dots, X_m , for each $i \in \{\lfloor m/2 \rfloor + 1, \dots, m\}$. Let $N = \{i \in \{\lfloor m/2 \rfloor + 1, \dots, m\} : \Gamma_i = 1\}$, and enumerate the elements of $\{X_i : i \in \{\lfloor m/2 \rfloor + 1, \dots, m\}, \Gamma_i = 1\}$ as $\hat{X}_1, \dots, \hat{X}_N$ (retaining their original order). For $X \sim \mathcal{P}$ independent of X_1, X_2, \dots , let $\Gamma(X)$ denote a random variable that is conditionally $\text{Bernoulli}(\gamma^*(X))$ given X and $X_1, \dots, X_{\lfloor m/2 \rfloor}$. Also define a (random) probability measure $P_{\lfloor m/2 \rfloor}$ such that, given $X_1, \dots, X_{\lfloor m/2 \rfloor}$, $P_{\lfloor m/2 \rfloor}(A) = \mathbb{P}(X \in A | \Gamma(X) = 1, X_1, \dots, X_{\lfloor m/2 \rfloor})$ for all measurable $A \subseteq \mathcal{X}$.

Note that $N = \sum_{i=\lfloor m/2 \rfloor+1}^m \Gamma_i$ is conditionally $\text{Binomial}(\lfloor m/2 \rfloor, \Phi(V_{\lfloor m/2 \rfloor}, \eta))$ given $X_1, \dots, X_{\lfloor m/2 \rfloor}$. In particular, with probability one, if $\Phi(V_{\lfloor m/2 \rfloor}, \eta) = 0$, then $N = 0$. Otherwise, if $\Phi(V_{\lfloor m/2 \rfloor}, \eta) > 0$, then $\hat{X}_1, \dots, \hat{X}_N$ are conditionally i.i.d. given $X_1, \dots, X_{\lfloor m/2 \rfloor}$ and N , each with conditional distribution $P_{\lfloor m/2 \rfloor}$ given $X_1, \dots, X_{\lfloor m/2 \rfloor}$ and N . Thus since every $h \in \mathcal{V}_m$ has $\{x : h(x) \neq f^*(x)\} \cap \{\hat{X}_1, \dots, \hat{X}_N\} \subseteq \{x : h(x) \neq f^*(x)\} \cap \{X_1, \dots, X_m\} = \emptyset$, and (one can easily show) $\text{vc}(\{x : h(x) \neq f^*(x)\} : h \in \mathcal{C}) = d$, applying Lemma 2 (under the conditional distribution given N and $X_1, \dots, X_{\lfloor m/2 \rfloor}$), combined with the law of total probability, we have that on an event E_1 of probability at least $1 - \delta/2$, if $N > 0$, then

$$\sup_{h \in \mathcal{V}_m} P_{\lfloor m/2 \rfloor}(x : h(x) \neq f^*(x)) \leq \frac{2}{N} \left(d \text{Log}_2 \left(\frac{2eN}{d} \right) + \text{Log}_2 \left(\frac{4}{\delta} \right) \right).$$

Next, since N is conditionally $\text{Binomial}(\lfloor m/2 \rfloor, \Phi(V_{\lfloor m/2 \rfloor}, \eta))$ given $X_1, \dots, X_{\lfloor m/2 \rfloor}$, applying a Chernoff bound (under the conditional distribution given $X_1, \dots, X_{\lfloor m/2 \rfloor}$), combined with the law of total probability, we obtain that on an event E_2 of probability at least

⁹ Note that the minimum is actually achieved here, since the objective function is continuous and convex, and the feasible region is nonempty, closed, bounded, and convex (see Bowers and Kalloni, 2014, Proposition 5.50).

$1 - \delta/4$, if $\Phi(V_{\lfloor m/2 \rfloor}, \eta) \geq \frac{18}{\lfloor m/2 \rfloor} \ln \left(\frac{4}{\delta} \right)$, then

$$N \geq (2/3) \Phi(V_{\lfloor m/2 \rfloor}, \eta) \lfloor m/2 \rfloor \geq \Phi(V_{\lfloor m/2 \rfloor}, \eta) m/3.$$

In particular, if $\Phi(V_{\lfloor m/2 \rfloor}, \eta) \geq \frac{18}{\lfloor m/2 \rfloor} \ln \left(\frac{4}{\delta} \right)$, then the right hand side is strictly greater than 0, so that if this occurs with E_2 , then we have $N > 0$. Thus, by the fact that $\text{Log}_2(x) \leq \text{Log}(x)/\ln(2)$, combined with monotonicity of $x \mapsto \text{Log}(x)/x$ for $x > 0$, we have that on $E_1 \cap E_2$, if $\Phi(V_{\lfloor m/2 \rfloor}, \eta) \geq \frac{18}{\lfloor m/2 \rfloor} \ln \left(\frac{4}{\delta} \right)$, then

$$\sup_{h \in \mathcal{V}_m} P_{\lfloor m/2 \rfloor}(x : h(x) \neq f^*(x)) \leq \frac{6/\ln(2)}{\Phi(V_{\lfloor m/2 \rfloor}, \eta) m} \left(d \text{Log} \left(\frac{2e\Phi(V_{\lfloor m/2 \rfloor}, \eta) m}{3d} \right) + \ln \left(\frac{4}{\delta} \right) \right).$$

Next (following an argument of Zhang and Chaudhuri, 2014), note that $\forall h \in \mathcal{V}_m$,

$$\begin{aligned} \text{er}(h) &= \mathbb{E}[\mathbb{1}[h(X) \neq f^*(X)](\gamma^*(X) + \zeta^*(X) + \xi^*(X)) | X_1, \dots, X_{\lfloor m/2 \rfloor}] \\ &= P_{\lfloor m/2 \rfloor}(x : h(x) \neq f^*(x)) \mathbb{P}(\Gamma(X) = 1 | X_1, \dots, X_{\lfloor m/2 \rfloor}) \\ &\quad + \mathbb{E}[\mathbb{1}[h(X) = +1] \mathbb{1}[f^*(X) = -1] \\ &\quad + \mathbb{1}[h(X) = -1] \mathbb{1}[f^*(X) = +1]] (\zeta^*(X) + \xi^*(X)) | X_1, \dots, X_{\lfloor m/2 \rfloor}] \\ &\leq P_{\lfloor m/2 \rfloor}(x : h(x) \neq f^*(x)) \Phi(V_{\lfloor m/2 \rfloor}, \eta) \\ &\quad + \mathbb{E}[\mathbb{1}[h(X) = +1] \zeta^*(X) + \mathbb{1}[h(X) = -1] \xi^*(X) | X_1, \dots, X_{\lfloor m/2 \rfloor}] \\ &\quad + \mathbb{E}[\mathbb{1}[f^*(X) = +1] \zeta^*(X) + \mathbb{1}[f^*(X) = -1] \xi^*(X) | X_1, \dots, X_{\lfloor m/2 \rfloor}]. \end{aligned}$$

Since $h, f^* \in \mathcal{V}_{\lfloor m/2 \rfloor}$, the definition of ζ^* and ξ^* implies this last expression is at most

$$P_{\lfloor m/2 \rfloor}(x : h(x) \neq f^*(x)) \Phi(V_{\lfloor m/2 \rfloor}, \eta) + 2\eta.$$

Therefore, on $E_1 \cap E_2$, if $\Phi(V_{\lfloor m/2 \rfloor}, \eta) \geq \frac{18}{\lfloor m/2 \rfloor} \ln \left(\frac{4}{\delta} \right)$, then

$$\sup_{h \in \mathcal{V}_m} \text{er}(h) \leq 2\eta + \frac{6/\ln(2)}{m} \left(d \text{Log} \left(\frac{2e\Phi(V_{\lfloor m/2 \rfloor}, \eta) m}{3d} \right) + \ln \left(\frac{4}{\delta} \right) \right).$$

The inductive hypothesis implies that, on an event E_3 of probability at least $1 - \delta/4$,

$$\sup_{h \in \mathcal{V}_{\lfloor m/2 \rfloor}} \text{er}(h) \leq \frac{21}{\lfloor m/2 \rfloor} \left(d \text{Log} \left(83 \varphi_c \left(\frac{d}{\lfloor m/2 \rfloor} \right) \right) + 3 \text{Log} \left(\frac{16}{\delta} \right) \right).$$

Since $m \geq \lceil 21(d \ln(83) + 3 \ln(4)) \rceil \geq 181$, we have $\lfloor m/2 \rfloor \geq (m-2)/2 \geq (179/362)m$, so that (together with monotonicity of $\varphi_c(\cdot)$) the above implies $\mathcal{V}_{\lfloor m/2 \rfloor} \subseteq \mathcal{B}(f^*, r_{\lfloor m/2 \rfloor}^c)$, where

$$r_{\lfloor m/2 \rfloor}^c = \frac{21 \cdot 362}{179m} \left(d \ln \left(83 \varphi_c \left(\frac{d}{m} \right) \right) + 3 \ln \left(\frac{16}{\delta} \right) \right).$$

Altogether, plugging in $\eta = (r_{\lfloor m/2 \rfloor}^c/c) \wedge 1$, and noting that $\mathcal{H} \rightarrow \Phi(\mathcal{H}, \eta)$ is nondecreasing in \mathcal{H} , and that $d/m \leq r_{\lfloor m/2 \rfloor}^c$, we have that on $E_1 \cap E_2 \cap E_3$, if $\Phi(V_{\lfloor m/2 \rfloor}, (r_{\lfloor m/2 \rfloor}^c/c) \wedge 1) \geq$

$\frac{18}{\lceil m/2 \rceil} \ln \left(\frac{4}{\delta} \right)$, then

$$\begin{aligned} \sup_{h \in V_m} \text{er}(h) &\leq \frac{2r_{\lceil m/2 \rceil}}{c} + \frac{6/\ln(2)}{m} \left(d \log \left(\frac{2e\Phi(\mathbf{B}(f^*, r_{\lceil m/2 \rceil}) \wedge (r_{\lceil m/2 \rceil}/c) \wedge 1)m}{3d} \right) + \ln \left(\frac{4}{\delta} \right) \right) \\ &\leq \frac{2r_{\lceil m/2 \rceil}}{c} + \frac{6/\ln(2)}{m} \left(d \ln \left(\frac{2e\varphi_c(d/m)r_{\lceil m/2 \rceil}m}{3d} \right) + \ln \left(\frac{4}{\delta} \right) \right). \end{aligned} \quad (27)$$

The second term in this last expression equals

$$\begin{aligned} \frac{6/\ln(2)}{m} \left(d \ln \left(\frac{14 \cdot 362}{179} \varphi_c \left(\frac{d}{m} \right) \right) + e \ln \left(83\varphi_c \left(\frac{d}{m} \right) \right) + \frac{3e}{d} \ln \left(\frac{16}{\delta} \right) \right) + \ln \left(\frac{4}{\delta} \right) \\ \leq \frac{6/\ln(2)}{m} \left(d \ln \left(\frac{14 \cdot 362 \cdot 6}{179 \cdot 7} \varphi_c \left(\frac{d}{m} \right) \right) + \frac{7e}{6} \ln \left(64 \cdot 83\varphi_c \left(\frac{d}{m} \right) \right) + \frac{7e}{2d} \ln \left(\frac{4}{\delta} \right) \right) + \ln \left(\frac{4}{\delta} \right). \end{aligned}$$

Applying Lemma 20 (with $b = (7e/2) \ln(4/\delta)$), this is at most

$$\frac{6/\ln(2)}{m} \left(d \ln \left(\frac{14 \cdot 362 \cdot 6}{179 \cdot 7} \varphi_c \left(\frac{d}{m} \right) \right) + \frac{7e}{6} \ln \left(64 \cdot 83\varphi_c \left(\frac{d}{m} \right) \right) + e \right) + \frac{9}{2} \ln \left(\frac{4}{\delta} \right),$$

and a simple relaxation of the expression in the logarithm reveals this is at most

$$\frac{6/\ln(2)}{m} \left(\frac{3}{2} d \ln \left(83\varphi_c \left(\frac{d}{m} \right) \right) + \frac{9}{2} \ln \left(\frac{4}{\delta} \right) \right) \leq \frac{13}{m} \left(d \ln \left(83\varphi_c \left(\frac{d}{m} \right) \right) + 3 \ln \left(\frac{4}{\delta} \right) \right).$$

Additionally, some straightforward reasoning about numerical constants reveals that

$$\frac{2r_{\lceil m/2 \rceil}}{c} \leq \frac{8}{m} \left(d \ln \left(83\varphi_c \left(\frac{d}{m} \right) \right) + 3 \ln \left(\frac{4}{\delta} \right) \right).$$

Plugging these two facts back into (27), we have that on $E_1 \cap E_2 \cap E_3$, if $\Phi(V_{\lceil m/2 \rceil}, (r_{\lceil m/2 \rceil}/c) \wedge 1) \geq \frac{18}{\lceil m/2 \rceil} \ln \left(\frac{4}{\delta} \right)$, then

$$\sup_{h \in V_m} \text{er}(h) \leq \frac{21}{m} \left(d \ln \left(83\varphi_c \left(\frac{d}{m} \right) \right) + 3 \ln \left(\frac{4}{\delta} \right) \right). \quad (28)$$

On the other hand, if $\Phi(V_{\lceil m/2 \rceil}, (r_{\lceil m/2 \rceil}/c) \wedge 1) < \frac{18}{\lceil m/2 \rceil} \ln \left(\frac{4}{\delta} \right)$, then recalling that (as established above) $\sup_{h \in V_m} \text{er}(h) \leq 2\eta + \sup_{h \in V_m} P_{\lceil m/2 \rceil}(x : h(x) \neq f^*(x))\Phi(V_{\lceil m/2 \rceil}, \eta)$, plugging in $\eta = (r_{\lceil m/2 \rceil}/c) \wedge 1$ and noting that $P_{\lceil m/2 \rceil}(x : h(x) \neq f^*(x)) \leq 1$, we have

$$\begin{aligned} \sup_{h \in V_m} \text{er}(h) &\leq \frac{2r_{\lceil m/2 \rceil}}{c} + \Phi(V_{\lceil m/2 \rceil}, (r_{\lceil m/2 \rceil}/c) \wedge 1) \\ &< \frac{8}{m} \left(d \ln \left(83\varphi_c \left(\frac{d}{m} \right) \right) + 3 \ln \left(\frac{4}{\delta} \right) \right) + \frac{18}{\lceil m/2 \rceil} \ln \left(\frac{4}{\delta} \right) \\ &\leq \frac{21}{m} \left(d \ln \left(83\varphi_c \left(\frac{d}{m} \right) \right) + 3 \ln \left(\frac{4}{\delta} \right) \right). \end{aligned}$$

Thus, in either case, on $E_1 \cap E_2 \cap E_3$, (28) holds. Noting that, by the union bound, the event $E_1 \cap E_2 \cap E_3$ has probability at least $1 - \delta$, this extends the inductive hypothesis to m . The result then follows by the principle of induction. \blacksquare

D.1 The Worst-Case Value of φ_c

Next, we prove (18). Fix any $c \geq 2$. First, suppose $r_0 \in (0, 1)$, and let $m = \min \left\{ \mathfrak{s}, \left\lceil \frac{1}{r_0} \right\rceil \right\}$; note that our assumption that $|C| \geq 3$ implies $\mathfrak{s} \geq 2$, so that $m \geq 2$ here. Let $x_1, \dots, x_m \in \mathcal{X}$ and $h_0, h_1, \dots, h_m \in \mathbb{C}$ be as in Definition 9. Let $\mathcal{P}(\{x_i\}) = 1/m$ for each $i \in [m]$, and take $f^* = h_0$.

Let r_1 be any value satisfying $\max\{1/m, r_0\} < r_1 \leq 1$ chosen sufficiently close to $\max\{1/m, r_0\}$ so that $\frac{mr_1}{c} < 1$. Consider now the definition of $\Phi(\mathbf{B}(f^*, r_1), r_1/c)$ from Definition 15. For any functions $\chi_0, \chi_1 : \mathcal{X} \rightarrow [0, 1]$, let $\zeta(x) = \mathbb{1}[h_0(x) = -1]\chi_0(x) + \mathbb{1}[h_0(x) = +1]\chi_1(x)$ and $\xi(x) = \mathbb{1}[h_0(x) = -1]\chi_1(x) + \mathbb{1}[h_0(x) = +1]\chi_0(x)$. In particular, note that it is possible to specify any functions $\zeta, \xi : \mathcal{X} \rightarrow [0, 1]$ by choosing appropriate χ_0, χ_1 values (namely, $\chi_0(x) = \mathbb{1}[h_0(x) = -1]\zeta(x) + \mathbb{1}[h_0(x) = +1]\xi(x)$ and $\chi_1(x) = \mathbb{1}[h_0(x) = -1]\xi(x) + \mathbb{1}[h_0(x) = +1]\zeta(x)$). Noting that, for any classifier h and any $x \in \mathcal{X}$, $\mathbb{1}[h(x) = -1]\zeta(x) + \mathbb{1}[h(x) = +1]\xi(x) = \mathbb{1}[h(x) \neq h_0(x)]\chi_0(x) + \mathbb{1}[h(x) = h_0(x)]\chi_1(x)$, and $\zeta(x) + \xi(x) = \chi_0(x) + \chi_1(x)$, we may re-express the constraints in the optimization problem defining $\Phi(\mathbf{B}(f^*, r_1), r_1/c)$ in Definition 15 as $\sup_{h \in \mathbf{B}(f^*, r_1)} \mathbb{E}[\mathbb{1}[h(X) \neq h_0(X)]\chi_0(X) + \mathbb{1}[h(X) = h_0(X)]\chi_1(X)] \leq r_1/c$ and $\forall x \in \mathcal{X}, \gamma(x) + \chi_0(x) + \chi_1(x) = 1$ while $\gamma(x), \chi_0(x), \chi_1(x) \in [0, 1]$. We may further simplify the problem by noting that $\gamma(x) = 1 - \chi_0(x) - \chi_1(x)$, so that these last two constraints become $\chi_0(x) + \chi_1(x) \leq 1$ while $\chi_0(x), \chi_1(x) \geq 0$, and the value $\Phi(\mathbf{B}(f^*, r_1), r_1/c)$ is the minimum achievable value of $\mathbb{E}[1 - \chi_0(X) - \chi_1(X)]$ subject to these constraints. Furthermore, noting that $h_i \in \mathbf{B}(f^*, r_1)$ for every $i \in [m]$, we have that

$$\begin{aligned} \Phi(\mathbf{B}(f^*, r_1), r_1/c) &\geq \min \left\{ \mathbb{E}[1 - \chi_0(X) - \chi_1(X)] : \right. \\ &\quad \left. \max_{i \in [m]} \mathbb{E}[\mathbb{1}[h_i(X) \neq h_0(X)]\chi_0(X) + \mathbb{1}[h_i(X) = h_0(X)]\chi_1(X)] \leq \frac{r_1}{c}, \right. \\ &\quad \left. \text{where } \forall x \in \mathcal{X}, \chi_0(x) + \chi_1(x) \leq 1 \text{ and } \chi_0(x), \chi_1(x) \geq 0 \right\} \\ &= \min \left\{ \sum_{i=1}^m \frac{1}{m} (1 - \chi_0(x_i) - \chi_1(x_i)) : \right. \\ &\quad \left. \forall i \in [m], \chi_0(x_i) + \sum_{j \neq i} \chi_1(x_j) \leq \frac{mr_1}{c}, \chi_0(x_i) + \chi_1(x_i) \leq 1, \chi_0(x_i), \chi_1(x_i) \geq 0 \right\}. \end{aligned}$$

This is a simple linear program with linear inequality constraints. We can explicitly solve this problem to find an optimal solution with $\chi_1(x_i) = 0$ and $\chi_0(x_i) = \frac{mr_1}{c}$ for all $i \in [m]$, at which the value of the objective function $\sum_{i=1}^m \frac{1}{m} (1 - \chi_0(x_i) - \chi_1(x_i))$ is $1 - \frac{mr_1}{c}$. One can easily verify that this choice of χ_0 and χ_1 satisfies the constraints above. To see that this is an optimal choice, we note that the objective function can be re-expressed as $\sum_{i=1}^m \frac{1}{m} (1 - \chi_0(x_i) - \chi_1(x_{\sigma(i)}))$, where $\sigma(i) = i + 1$ for $i \in [m-1]$, and $\sigma(m) = 1$. In particular, since $m \geq 2$, we have $\sigma(i) \neq i$ for each $i \in [m]$. Thus, for any χ_0 and χ_1 satisfying the constraints above, we have $\chi_0(x_i) + \chi_1(x_{\sigma(i)}) \leq \chi_0(x_i) + \sum_{j \neq i} \chi_1(x_j) \leq \frac{mr_1}{c}$

for each $i \in [m]$, so that $\sum_{i=1}^m \frac{1}{m} (1 - \chi_0(x_i) - \chi_1(x_i)) \geq 1 - \frac{mm_1}{c}$, which is precisely the value obtained with the above choices of χ_0 and χ_1 .

Thus, since the above argument holds for any choice of $r_1 > \max\{1/m, r_0\}$ sufficiently close to $\max\{1/m, r_0\}$, we have

$$\varphi_c(r_0) = \sup_{r_0 < r \leq 1} \frac{\Phi(\mathcal{B}(f^*, r), r/c)}{r} \vee 1 \geq \lim_{r_1 \searrow \max\{1/m, r_0\}} \frac{1 - \frac{mm_1}{c}}{r_1} = \frac{1 - \frac{1}{c} \max\{1, mm_0\}}{\max\{1/m, r_0\}}.$$

If $s < \frac{1}{r_0}$, then $m = s$, and the rightmost expression above equals $(1 - 1/c)s$. Otherwise, if $s \geq \frac{1}{r_0}$, then $m = \lceil \frac{1}{r_0} \rceil$, and the rightmost expression above equals

$$\left(1 - \frac{1}{c} \left\lceil \frac{1}{r_0} \right\rceil r_0\right) \frac{1}{r_0} \geq \left(1 - \frac{1+r_0}{c}\right) \frac{1}{r_0} = \left(1 - \frac{1}{c}\right) \left(\frac{1}{r_0} - \frac{1}{c-1}\right).$$

Either way, we have

$$\varphi_c(r_0) \geq \left(1 - \frac{1}{c}\right) \min\left\{s, \frac{1}{r_0} - \frac{1}{c-1}\right\}.$$

For the case $r_0 = 0$, we note that $\forall \varepsilon > 0$, any $c \geq 2$ has

$$\sup_{\mathcal{P}} \sup_{f^* \in \mathcal{C}} \varphi_c(0) \geq \sup_{\mathcal{P}} \sup_{f^* \in \mathcal{C}} \varphi_c(\varepsilon) \geq \left(1 - \frac{1}{c}\right) \min\left\{s, \frac{1}{\varepsilon} - \frac{1}{c-1}\right\}.$$

Taking the limit $\varepsilon \rightarrow 0$ yields $\sup_{\mathcal{P}} \sup_{f^* \in \mathcal{C}} \varphi_c(0) \geq (1 - \frac{1}{c})s = (1 - \frac{1}{c}) \min\left\{s, \frac{1}{r_0} - \frac{1}{c-1}\right\}$.

For the upper bound, we clearly have $\varphi_c(r_0) \leq (1 - 1/c)\theta(r_0)$ for every $c > 1$. To see this, take $\zeta(x) = (1/c)\mathbb{1}[x \in \text{DIS}(\mathcal{B}(f^*, r))]\mathbb{1}[f^*(x) = -1] + \mathbb{1}[x \notin \text{DIS}(\mathcal{B}(f^*, r))]\mathbb{1}[f^*(x) = -1]$ and $\xi(x) = (1/c)\mathbb{1}[x \in \text{DIS}(\mathcal{B}(f^*, r))]\mathbb{1}[f^*(x) = +1] + \mathbb{1}[x \notin \text{DIS}(\mathcal{B}(f^*, r))]\mathbb{1}[f^*(x) = +1]$ in the optimization problem defining $\Phi(\mathcal{B}(f^*, r), r/c)$ in Definition 15. With these choices of ζ and ξ , we have $\mathbb{E}[\gamma(X)] = (1 - 1/c)\mathcal{P}(\text{DIS}(\mathcal{B}(f^*, r)))$; also, for any $h \in \mathcal{B}(f^*, r)$, since $\text{DIS}(\{h, f^*\}) \subseteq \text{DIS}(\mathcal{B}(f^*, r))$, we have $\mathbb{E}[\mathbb{1}[h(X) = +1]\zeta(X) + \mathbb{1}[h(X) = -1]\xi(X)] = \mathbb{E}[(1/c)\mathbb{1}[h(X) \neq f^*(X)]] = (1/c)\mathcal{P}(x : h(x) \neq f^*(x)) \leq r/c$; one can easily verify that the remaining constraints are also satisfied. Thus, since Hanneke and Yang (2015) prove $\sup_{\mathcal{P}} \sup_{f^* \in \mathcal{C}} \theta(r_0) = \min\left\{s, \frac{1}{r_0}\right\}$, we have $\sup_{\mathcal{P}} \sup_{f^* \in \mathcal{C}} \varphi_c(r_0) \leq (1 - 1/c) \min\left\{s, \frac{1}{r_0}\right\}$.

We also note that, if we define $\varphi_c^{\text{th}}(r_0)$ identically to $\varphi_c(r_0)$ except that γ is restricted to have *binary* values (i.e., in $\{0, 1\}$), then for $c \geq 4$, this same construction giving the lower bound above must have $\gamma(x_i) = 1$ for every $i \in [m]$, which implies $\varphi_c^{\text{th}}(r_0) \geq \min\left\{s, \frac{1}{r_0}\right\}$ in this case. To see this, consider any $r_1 > \max\{1/m, r_0\}$ sufficiently small so that $\frac{mm_1}{c} < \frac{1}{2}$; then to satisfy the constraints $\chi_0(x_i) + \sum_{j \neq i} \chi_1(x_j) \leq \frac{mm_1}{c} < \frac{1}{2}$ for every $i \in [m]$, while $\chi_0(x_i), \chi_1(x_i) \geq 0$, we must have every $\chi_0(x_i)$ and $\chi_1(x_i)$ strictly less than $\frac{1}{2}$, so that $\gamma(x_i) = 1 - \chi_0(x_i) - \chi_1(x_i) > 0$ (and hence, $\gamma(x_i) = 1$, due to the constraint to binary values). As we always have $\varphi_c^{\text{th}}(r_0) \leq \theta(r_0)$, and Hanneke and Yang (2015) have shown $\sup_{\mathcal{P}} \sup_{f^* \in \mathcal{C}} \theta(r_0) = \min\left\{s, \frac{1}{r_0}\right\}$, this implies $\sup_{\mathcal{P}} \sup_{f^* \in \mathcal{C}} \varphi_c^{\text{th}}(r_0) = \min\left\{s, \frac{1}{r_0}\right\}$ as well.

D.2 Relation of $\varphi_c(r_0)$ to the Doubling Dimension

Here we present the proof of (21), via a modification of an argument of Hanneke and Yang (2015). We in fact prove the following slightly stronger inequality: for any $c \geq 8$ and $r > 0$,

$$\log_2(\mathcal{N}(r/2, \mathcal{B}(f^*, r), \mathcal{P})) \leq 2d \log_2 \left(96 \left(\frac{\Phi(\mathcal{B}(f^*, r), r/c)}{r} \vee 1 \right) \right), \quad (29)$$

which will immediately imply (21) by taking the supremum of both sides over $r > r_0$ (with some careful consideration of the special case $r = r_0$; see below).

Fix any $c > 4$ and $r \in (0, 1]$. Let G_r denote any maximal $(r/2)$ -packing of $\mathcal{B}(f^*, r)$: that is, G_r is a subset of $\mathcal{B}(f^*, r)$ of maximal cardinality such that $\min_{h, g \in G_r: h \neq g} \mathcal{P}(x : h(x) \neq g(x)) > r/2$. It is known that any such set G_r satisfies

$$\mathcal{N}(r/2, \mathcal{B}(f^*, r), \mathcal{P}) \leq |G_r| \leq \mathcal{N}(r/4, \mathcal{B}(f^*, r), \mathcal{P}) \quad (30)$$

(see e.g., Kolmogorov and Tikhomirov, 1959, 1961; Vidyasagar, 2003). In particular, since we have assumed $d < \infty$, in our case this further implies $|G_r| < \infty$ (Haussler, 1995). Also, this implies that if $|G_r| = 1$, then (29) trivially holds, so let us suppose $|G_r| \geq 2$.

Now fix any measurable functions γ, ζ, ξ mapping $\mathcal{X} \rightarrow [0, 1]$ satisfying the constraint $\sup_{h \in \mathcal{B}(f^*, r)} \mathbb{E}[\mathbb{1}[h(X) = +1]\zeta(X) + \mathbb{1}[h(X) = -1]\xi(X)] \leq r/c$, where $X \sim \mathcal{P}$, and $\forall x \in \mathcal{X}$, $\gamma(x) + \zeta(x) + \xi(x) = 1$; for simplicity, also suppose $\mathbb{E}[\gamma(X)] \geq r$. As above, for $m \in \mathbb{N}$, let X_1, \dots, X_m be independent \mathcal{P} -distributed random variables. Then let $\Gamma_1, \dots, \Gamma_m$ be conditionally independent given X_1, \dots, X_m , with the conditional distribution of each Γ_i as Bernoulli($\gamma(X_i)$) given X_1, \dots, X_m . Let $N_m = \{i \in [m] : \Gamma_i = 1\}$, and let $\hat{X}_1, \dots, \hat{X}_m$ denote the subsequence of X_1, \dots, X_m for which the respective $\Gamma_i = 1$.

By two applications of the Chernoff bound, combined with the union bound, the event $E_1 = \{m\mathbb{E}[\gamma(X)]/2 \leq N_m \leq 2m\mathbb{E}[\gamma(X)]\}$ has probability at least $1 - 2\exp\{-m\mathbb{E}[\gamma(X)]/8\}$. Additionally, $\forall f, g \in G_r$ with $f \neq g$, $\forall i \in [m]$,

$$\begin{aligned} \mathbb{P}(f(X_i) \neq g(X_i) \text{ and } \Gamma_i = 0) \\ &= \mathbb{E}[\mathbb{1}[f(X) \neq g(X)](1 - \gamma(X))] = \mathbb{E}[\mathbb{1}[f(X) \neq g(X)](\zeta(X) + \xi(X))] \\ &= \mathbb{E}[\mathbb{1}[f(X) = +1]\mathbb{1}[g(X) = -1] + \mathbb{1}[f(X) = -1]\mathbb{1}[g(X) = +1]](\zeta(X) + \xi(X)) \\ &\leq \mathbb{E}[\mathbb{1}[f(X) = +1]\zeta(X) + \mathbb{1}[f(X) = -1]\xi(X)] + \mathbb{E}[\mathbb{1}[g(X) = -1]\xi(X) + \mathbb{1}[g(X) = +1]\zeta(X)] \\ &\leq \frac{2r}{c} \end{aligned}$$

so that

$$\mathbb{P}(f(X_i) \neq g(X_i) \text{ and } \Gamma_i = 1) = \mathbb{P}(f(X_i) \neq g(X_i)) - \mathbb{P}(f(X_i) \neq g(X_i) \text{ and } \Gamma_i = 0) > \frac{r}{2} - \frac{2r}{c}.$$

In particular, this implies

$$\mathbb{P}(f(X_i) \neq g(X_i) | \Gamma_i = 1) \geq \left(\frac{1}{2} - \frac{2}{c}\right) \frac{r}{\mathbb{E}[\gamma(X)]}.$$

Therefore,

$$\begin{aligned} \mathbb{P}\left(\exists i \in [N_m] : f(\hat{X}_i) \neq g(\hat{X}_i) \mid N_m\right) &= 1 - (1 - \mathbb{P}(f(X_i) \neq g(X_i) | \Gamma_i = 1))^{N_m} \\ &\geq 1 - \left(1 - \left(\frac{1}{2} - \frac{2}{c}\right) \frac{r}{\mathbb{E}[\gamma(X)]}\right)^{N_m} \geq 1 - \exp\left\{-\left(\frac{1}{2} - \frac{2}{c}\right) \frac{r}{\mathbb{E}[\gamma(X)]} N_m\right\}. \end{aligned}$$

On the event E_1 , this is at least $1 - \exp\{-\frac{1}{4} - \frac{1}{c}\}rm$. Altogether, we have that

$$\begin{aligned} \mathbb{P}(E_1 \text{ and } \exists i \in [N_m] : f(\hat{X}_i) \neq g(\hat{X}_i)) &= \mathbb{E}[\mathbb{1}_{E_1} \cdot \mathbb{P}(\exists i \in [N_m] : f(\hat{X}_i) \neq g(\hat{X}_i) | N_m)] \\ &\geq \left(1 - \exp\left\{-\left(\frac{1}{4} - \frac{1}{c}\right)rm\right\}\right) \mathbb{P}(E_1) \\ &\geq 1 - \exp\left\{-\left(\frac{1}{4} - \frac{1}{c}\right)rm\right\} - 2 \exp\{-m\mathbb{E}[\gamma(X)]/8\} \\ &\geq 1 - \exp\left\{-\left(\frac{c-4}{4c}\right)rm\right\} - 2 \exp\{-mr/8\}. \end{aligned}$$

In particular, choosing

$$m = \left\lceil \frac{1}{r} \left(\frac{4c}{c-4} \vee 8\right) \ln(2|G_r|^2) \right\rceil,$$

we have that $\mathbb{P}(E_1 \text{ and } \exists i \in [N_m] : f(\hat{X}_i) \neq g(\hat{X}_i)) \geq 1 - \frac{2}{|G_r|^2}$. By a union bound, this implies that with probability at least $1 - \frac{2}{|G_r|^2}$ ($\frac{|G_r|}{2} > 0$, E_1 holds and, for every $f, g \in G_r$ with $f \neq g$, $\exists i \in [N_m]$ for which $f(\hat{X}_i) \neq g(\hat{X}_i)$): that is, every $f \in G_r$ classifies $\hat{X}_1, \dots, \hat{X}_{N_m}$ distinctly. But for this to be the case, $|G_r|$ can be at most the number of distinct classifications of a sequence of N_m points in \mathcal{X} realizable by classifiers in \mathbb{C} , where (since E_1 also holds) $N_m \leq 2m\mathbb{E}[\gamma(X)]$. Together with the VC-Sauer lemma (Vapnik and Chervonenkis, 1971; Sauer, 1972), this implies that

$$\begin{aligned} \log_2(|G_r|) &\leq d \log_2 \left(\frac{2em\mathbb{E}[\gamma(X)]}{d} \vee 2 \right) \\ &\leq d \log_2 \left(\frac{35 \cdot 4e}{33} \left(\frac{4c}{c-4} \vee 8\right) \frac{\mathbb{E}[\gamma(X)]}{r} \frac{1}{d} (\ln(\sqrt{2}) + \ln(|G_r|)) \vee 2 \right) \\ &= d \log_2 \left(\frac{35 \cdot 4e}{33 \log_2(e)} \left(\frac{4c}{c-4} \vee 8\right) \frac{\mathbb{E}[\gamma(X)]}{r} \frac{1}{d} ((1/2) + \log_2(|G_r|)) \vee 2 \right), \end{aligned}$$

where the second inequality follows from the fact that $8 \ln(2|G_r|^2) > 16.5$ (since $|G_r| \geq 2$), so that $m \leq \frac{17.5}{16.5} \frac{1}{r} \left(\frac{4c}{c-4} \vee 8\right) \ln(2|G_r|^2) = \frac{35}{33} \frac{1}{r} \left(\frac{4c}{c-4} \vee 8\right) \ln(2|G_r|^2)$.

If $\log_2(|G_r|) \leq d$, then together with (30), the inequality (29) trivially holds. Otherwise, if $\log_2(|G_r|) > d$, then letting $K = \frac{1}{2} \log_2(|G_r|) \geq 1$, the above implies

$$\begin{aligned} K &\leq \log_2 \left(\frac{35 \cdot 4e}{33 \log_2(e)} \left(\frac{4c}{c-4} \vee 8\right) \frac{\mathbb{E}[\gamma(X)]}{r} \frac{3}{2} K \right) \\ &= \log_2 \left(\frac{35 \cdot 4e}{22 \log_2(e)} \left(\frac{4c}{c-4} \vee 8\right) \frac{\mathbb{E}[\gamma(X)]}{r} \right) + \log_2(K). \end{aligned}$$

Via some simple calculus (see e.g., Vidyasagar, 2003, Lemma 4.6), this implies

$$K \leq 2 \log_2 \left(\frac{35 \cdot 4e}{22 \log_2(e)} \left(\frac{4c}{c-4} \vee 8\right) \frac{\mathbb{E}[\gamma(X)]}{r} \right).$$

Noting that $\frac{35 \cdot 4e}{22 \log_2(e)} < 12$, together with (30), we have that

$$\log_2(\mathcal{N}(r/2, \mathbb{B}(f^*, r), \mathcal{P})) \leq 2d \log_2 \left(12 \left(\frac{4c}{c-4} \vee 8\right) \frac{\mathbb{E}[\gamma(X)]}{r} \right). \quad (31)$$

This inequality holds for any choice of γ, ζ, ξ satisfying the constraints in the definition of $\Phi(\mathbb{B}(f^*, r), r/c)$ from Definition 15, with the additional constraint that $\mathbb{E}[\gamma(X)] \geq r$. Thus, if $\Phi(\mathbb{B}(f^*, r), r/c) \geq r$, then by minimizing the right hand side of (31) over the choice of γ, ζ, ξ , it follows that

$$\log_2(\mathcal{N}(r/2, \mathbb{B}(f^*, r), \mathcal{P})) \leq 2d \log_2 \left(12 \left(\frac{4c}{c-4} \vee 8\right) \frac{\Phi(\mathbb{B}(f^*, r), r/c)}{r} \right).$$

Otherwise, if $\Phi(\mathbb{B}(f^*, r), r/c) < r$, then we note that, for any functions γ^*, ζ^*, ξ^* satisfying the constraints from the definition of $\Phi(\mathbb{B}(f^*, r), r/c)$ such that $\mathbb{E}[\gamma^*(X)] = \Phi(\mathbb{B}(f^*, r), r/c)$, there exists functions γ, ζ, ξ satisfying the constraints from the definition of $\Phi(\mathbb{B}(f^*, r), r/c)$ for which $\mathbb{E}[\gamma(X)] = r$. For instance, we can take γ based on a convex combination of γ^* and 1: $\gamma(x) = \frac{1-r}{1-\mathbb{E}[\gamma^*(X)]} \gamma^*(x) + \frac{r-\mathbb{E}[\gamma^*(X)]}{1-\mathbb{E}[\gamma^*(X)]}$, $\zeta(x) = (\zeta^*(x) - (\gamma(x) - \gamma^*(x))) \vee 0$, $\xi(x) = 1 - \gamma(x) - \zeta(x)$; one can easily verify that, since $0 \leq \zeta(x) \leq \zeta^*(x)$ and $0 \leq \xi(x) \leq \xi^*(x)$, this choice of γ, ζ, ξ still satisfy the requirements for γ, ζ, ξ above, and that furthermore, $\mathbb{E}[\gamma(X)] = r$. Therefore, (31) implies $\log_2(\mathcal{N}(r/2, \mathbb{B}(f^*, r), \mathcal{P})) \leq 2d \log_2 \left(12 \left(\frac{4c}{c-4} \vee 8\right) \right)$. Thus, either way, we have established that

$$\log_2(\mathcal{N}(r/2, \mathbb{B}(f^*, r), \mathcal{P})) \leq 2d \log_2 \left(12 \left(\frac{4c}{c-4} \vee 8\right) \frac{\Phi(\mathbb{B}(f^*, r), r/c)}{r} \vee 1 \right). \quad (32)$$

Noting that, for any $c \geq 8$, $\frac{4c}{c-4} \leq 8$, this establishes (29) for any $c \geq 8$ and $r \in (0, 1]$.

In the case of $r > 1$, a result of Haussler (1995) implies that

$$\begin{aligned} \log_2(\mathcal{N}(r/2, \mathbb{B}(f^*, r), \mathcal{P})) &\leq \log_2(\mathcal{N}(1/2, \mathbb{C}, \mathcal{P})) \leq d \log_2(4e) + \log_2(\epsilon(d+1)) \\ &\leq d \log_2(4e) + d + \log_2(\epsilon) \leq d \log_2(8e^2) \leq d \log_2(96) \leq 2d \log_2 \left(96 \left(\frac{\Phi(\mathbb{B}(f^*, r), r/c)}{r} \vee 1\right) \right), \end{aligned}$$

so that both (29) and (32) are also valid for $r > 1$. This completes the proof of (29).

As a final step in the proof of (21), we note that there is a slight complication to be resolved, since the definition of $D(r_0)$ includes r_0 in the range of r , while the definition of $\varphi_c(r_0)$ does not. However, we note that, for any $c > 4$, any $r_0 > 0$, and any $r > r_0$ sufficiently close to r_0 , we have $c > cr_0/r > 4$, so that (32) would imply

$$\begin{aligned} \log_2(\mathcal{N}(r_0/2, \mathbb{B}(f^*, r_0), \mathcal{P})) &\leq 2d \log_2 \left(12 \left(\frac{4(cr_0/r)}{(cr_0/r)-4} \vee 8\right) \frac{\Phi(\mathbb{B}(f^*, r_0), r_0/(cr_0/r))}{r_0} \vee 1 \right) \\ &\leq 2d \log_2 \left(12 \left(\frac{4c}{(cr_0/r)-4} \vee \frac{8r}{r_0}\right) \frac{\Phi(\mathbb{B}(f^*, r), r/c)}{r} \vee 1 \right). \end{aligned}$$

Then taking the limit as $r \searrow r_0$ implies

$$\begin{aligned} \log_2(\mathcal{N}(r_0/2, \mathbb{B}(f^*, r_0), \mathcal{P})) &\leq 2d \log_2 \left(12 \left(\frac{4c}{c-4} \vee 8\right) \lim_{r \searrow r_0} \left(\frac{\Phi(\mathbb{B}(f^*, r), r/c)}{r} \vee 1 \right) \right) \\ &\leq 2d \log_2 \left(12 \left(\frac{4c}{c-4} \vee 8\right) \varphi_c(r_0) \right). \end{aligned}$$

In particular, for any $c \geq 8$, $\frac{4c}{c-4} \leq 8$, so that

$$\log_2(\mathcal{N}(r_0/2, \mathcal{B}(f^*, r_0), \mathcal{P})) \leq 2d \log_2(96\varphi_c(r_0)).$$

Together with the above, we therefore have that, for any $c \geq 8$ and $r_0 > 0$,

$$\begin{aligned} D(r_0) &= \max \left\{ \log_2(\mathcal{N}(r_0/2, \mathcal{B}(f^*, r_0), \mathcal{P})), \sup_{r>r_0} \log_2(\mathcal{N}(r/2, \mathcal{B}(f^*, r), \mathcal{P})) \right\} \\ &\leq \max \left\{ 2d \log_2(96\varphi_c(r_0)), \sup_{r>r_0} 2d \log_2 \left(96 \left(\frac{\Phi(\mathcal{B}(f^*, r), r/c)}{r} \vee 1 \right) \right) \right\} \\ &= 2d \log_2(96\varphi_c(r_0)). \end{aligned}$$

Thus, we have established (21).

Appendix E. Proofs of Results on Learning with Noise

This appendix includes the proofs of results in Section 6: namely, Theorems 17 and 19.

E.1 Proof of Theorem 17

We begin with the proof of Theorem 17. The proof follows a technique of Hanneke and Yang (2015), which identifies a subset of classifiers in \mathbb{C} , corresponding to a certain concept space for which Raginsky and Rakhlin (2011) have established lower bounds. Specifically, the following setup is taken directly from Hanneke and Yang (2015). Fix $\zeta \in (0, 1]$, $\beta \in [0, 1/2]$, and $k \in \mathbb{N}$ with $k \leq \min\{1/\zeta, |\mathcal{X}|-1\}$. Let $\mathcal{X}_k = \{x_1, \dots, x_{k+1}\}$ be a set of $k+1$ distinct elements of \mathcal{X} , and define $\mathbb{C}_k = \{x \mapsto 2\mathbb{1}\{x\}(x) - 1 : x \in [k]\}$. Let $\mathcal{P}_{k,\zeta}$ be a probability measure over \mathcal{X} with $\mathcal{P}\{x_i\} = \zeta$ for each $i \in [k]$, and $\mathcal{P}_{k,\zeta}\{x_{k+1}\} = 1 - \zeta k$. For each $t \in [k]$, let $\mathcal{P}'_{k,\zeta,t}$ be a probability measure over $\mathcal{X} \times \mathcal{Y}$ with marginal distribution $\mathcal{P}_{k,\zeta}$ over \mathcal{X} , such that for $(X, Y) \sim \mathcal{P}'_{k,\zeta,t}$ every $i \in [k]$ has $\mathbb{P}\{Y = 2\mathbb{1}\{x_i\}(X) - 1 | X = x_i\} = 1 - \beta$, and $\mathbb{P}\{Y = -1 | X = x_{k+1}\} = 1$. Raginsky and Rakhlin (2011) prove the following result (see the proof of their Theorem 1),¹⁰

Lemma 21 For k, ζ, β as above, with $k \geq 2$, for any $\delta \in (0, 1/4)$, for any (passive) learning rule \mathbb{A} , and any $m \in \mathbb{N}$ with

$$m < \max \left\{ \frac{\beta \ln(\frac{1}{\delta})}{2\zeta(1-2\beta)^2}, \frac{3\beta \ln(\frac{k}{96})}{16\zeta(1-2\beta)^2} \right\},$$

if $\mathbb{C}_k \subseteq \mathbb{C}$, then there exists a $t \in [k]$ such that, if $\mathcal{P}_{XY} = \mathcal{P}'_{k,\zeta,t}$, then denoting $\hat{h}_m = \mathbb{A}(\mathcal{L}_m)$, with probability greater than δ ,

$$\text{er}(\hat{h}_m) - \inf_{h \in \mathbb{C}} \text{er}(h) \geq (\zeta/2)(1-2\beta).$$

10. As noted by Hanneke and Yang (2015), although technically the proof of this result by Raginsky and Rakhlin (2011) relies on a lemma (their Lemma 4) that imposes additional restrictions on k and a parameter ν, μ , one can easily verify that the conclusions of that lemma continue to hold in the special case considered here (corresponding to $d = 1$ and arbitrary $k \in \mathbb{N}$) by defining $\mathcal{M}_{k,1} = \{0, 1\}^k$ in their construction.

Continuing to follow Hanneke and Yang (2015), we embed the above scenario into the general case, so that Lemma 21 provides a lower bound. Fix any $\zeta \in (0, 1]$, $\beta \in [0, 1/2]$, and $k \in \mathbb{N}$ with $k \leq \min\{s-1, \lfloor 1/\zeta \rfloor\}$, and let x_1, \dots, x_{k+1} and h_0, h_1, \dots, h_k be as in Definition 9. Let $\mathcal{P}_{k,\zeta}$ be as above (for this choice of x_1, \dots, x_{k+1}), and for each $t \in [k]$, let $\mathcal{P}_{k,\zeta,t}$ denote a probability measure over $\mathcal{X} \times \mathcal{Y}$ with marginal distribution $\mathcal{P}_{k,\zeta}$ over \mathcal{X} such that, for $(X, Y) \sim \mathcal{P}_{k,\zeta,t}$, $\mathbb{P}\{Y = h_t(X) | X = x_i\} = 1 - \beta$ for every $i \in [k]$, while $\mathbb{P}\{Y = h_0(X) | X = x_{k+1}\} = 1$.

Lemma 22 For k, ζ, β as above, with $k \geq 96e$, for any $\delta \in (0, 1/4)$, for any (passive) learning rule \mathbb{A} , and any $m \in \mathbb{N}$ with

$$m < \frac{3\beta \ln(\frac{k}{96})}{16\zeta(1-2\beta)^2},$$

there exists a $t \in [k]$ such that, if $\mathcal{P}_{XY} = \mathcal{P}_{k,\zeta,t}$, then denoting $\hat{h}_m = \mathbb{A}(\mathcal{L}_m)$, with probability greater than δ ,

$$\text{er}(\hat{h}_m) - \inf_{h \in \mathbb{C}} \text{er}(h) \geq (\zeta/2)(1-2\beta).$$

The proof of Lemma 22 is essentially identical to the proof of Hanneke and Yang (2015, Lemma 26), except that the algorithm \mathbb{A} here is restricted to be a passive learning rule so that Lemma 21 can be applied (in place of Lemma 25 there). As such, we omit the details here for brevity.

We are now ready for the proof of Theorem 17.

Proof of Theorem 17 Fix any $\beta \in (0, 1/2)$, $\delta \in (0, 1/24)$, $m \in \mathbb{N}$, and any (passive) learning rule \mathbb{A} . First consider the case of $s \geq 97e$. Fix $\varepsilon \in (0, (1-2\beta)/(384e^2))$, and let $\zeta = \frac{2\varepsilon}{1-2\beta}$ and $k = \min\{s-1, \lfloor 1/\zeta \rfloor\}$. Then, noting that the distributions $\mathcal{P}_{k,\zeta,t}$ above satisfy the β -bounded noise condition, Lemma 22 implies that if

$$m < \frac{3\beta \ln(\frac{k}{96})}{32\varepsilon(1-2\beta)}, \quad (33)$$

then there exists a choice of \mathcal{P}_{XY} satisfying the β -bounded noise condition such that, with probability greater than δ , the classifier $\hat{h}_m = \mathbb{A}(\mathcal{L}_m)$ has

$$\text{er}(\hat{h}_m) - \inf_{h \in \mathbb{C}} \text{er}(h) \geq \varepsilon.$$

Note that for any $m \in \mathbb{N}$ and $\varepsilon \in (0, (1-2\beta)/(384e^2))$, it holds that (see e.g., Vidyasagar, 2003, Corollary 4.1)

$$\begin{aligned} m &\leq \frac{3\beta}{64\varepsilon(1-2\beta)} \ln \left(\frac{(1-2\beta)^2 m}{18\beta} \right) \\ &\implies m < \frac{3\beta \ln(\frac{1-2\beta}{384\varepsilon})}{32\varepsilon(1-2\beta)} \leq \frac{3\beta \ln(\frac{1/\zeta}{96})}{32\varepsilon(1-2\beta)}. \end{aligned}$$

Thus, the inequality in (33) is satisfied if both

$$m < \frac{3\beta \ln(\frac{s-1}{96})}{32\varepsilon(1-2\beta)}$$

and

$$m \leq \frac{3\beta}{64\epsilon(1-2\beta)} \ln \left(\frac{(1-2\beta)^2 m}{18\beta} \right).$$

Solving for a value $\epsilon \in (0, (1-2\beta)/(384e^2))$ that satisfies both of these, we have that for any $m \in \mathbb{N}$ with $m \geq \frac{18e\beta}{(1-2\beta)^2}$, there is a choice of \mathcal{P}_{XY} satisfying the β -bounded noise condition such that, with probability greater than δ ,

$$\begin{aligned} \text{er}(\hat{h}_m) - \inf_{h \in \mathbb{C}} \text{er}(h) &\geq \frac{3\beta \ln \left(\min \left\{ \frac{s-1}{96}, \frac{(1-2\beta)^2 m}{18\beta} \right\} \right)}{64(1-2\beta)m} \wedge \frac{1-2\beta}{384e^2} \\ &\geq \frac{\beta \text{Log}(\min \{s, (1-2\beta)^2 m\})}{(1-2\beta)m} \wedge (1-2\beta). \end{aligned}$$

Furthermore, for $m < \frac{18e\beta}{(1-2\beta)^2}$, we may also think of \hat{h}_m as the output of $\mathcal{A}'(\mathcal{L}_{m'})$ for $m' = \left\lceil \frac{18e\beta}{(1-2\beta)^2} \right\rceil > m$, for a learning rule \mathcal{A}' which simply discards the last $m' - m$ samples and runs $\mathcal{A}(\mathcal{L}_m)$ to produce its return classifier. Thus, the above result implies that for $m < \frac{18e\beta}{(1-2\beta)^2}$, with probability greater than δ ,

$$\text{er}(\hat{h}_m) - \inf_{h \in \mathbb{C}} \text{er}(h) \geq \frac{3\beta \ln \left(\min \left\{ \frac{s-1}{96}, \frac{(1-2\beta)^2 m'}{18\beta} \right\} \right)}{64(1-2\beta)m'} \wedge \frac{1-2\beta}{384e^2}.$$

Since $m, m' \in \mathbb{N}$ and $m' > m$, we know that $m' \geq 2$, so that $\frac{18e\beta}{(1-2\beta)^2} \leq m' \leq \frac{36e\beta}{(1-2\beta)^2}$. Therefore,

$$\frac{3\beta \ln \left(\min \left\{ \frac{s-1}{96}, \frac{(1-2\beta)^2 m'}{18\beta} \right\} \right)}{64(1-2\beta)m'} \geq \frac{3\beta}{64(1-2\beta)m'} \geq \frac{3(1-2\beta)}{64 \cdot 36e} \frac{(1-2\beta)}{384e^2}.$$

Thus, in this case, we have that with probability greater than δ ,

$$\text{er}(\hat{h}_m) - \inf_{h \in \mathbb{C}} \text{er}(h) \geq \frac{(1-2\beta)}{384e^2} \gtrsim (1-2\beta) \geq \frac{\beta \text{Log}(\min \{s, (1-2\beta)^2 m\})}{(1-2\beta)m} \wedge (1-2\beta).$$

Next, we return to the general case of arbitrary $s \in \mathbb{N} \cup \{\infty\}$. In particular, since any $s < 97e$ has $\frac{\beta \text{Log}(\min \{s, (1-2\beta)^2 m\})}{(1-2\beta)m} \lesssim \frac{d}{(1-2\beta)m}$, to complete the proof it suffices to establish a lower bound

$$\text{er}(\hat{h}_m) - \inf_{h \in \mathbb{C}} \text{er}(h) \gtrsim \frac{1}{(1-2\beta)m} \left(d + \text{Log} \left(\frac{1}{\delta} \right) \right) \wedge (1-2\beta),$$

holding with probability greater than δ . This lower bound is already known, and frequently referred to in the literature; it follows from well-known constructions (see e.g., Anthony and Bartlett, 1999; Massart and Nédélec, 2006; Hanneke, 2011, 2014). The case $\beta < 3/8$ is covered by the classic minimax lower bound of Ehrenfeucht, Haussler, Kearns, and Valiant (1989) for the realizable case, while the case $\beta \geq 3/8$ is addressed by Hanneke (2014, Theorem 3.5). However, it seems an explicit proof of this latter result has not actually

appeared in the literature. As such, for completeness, we include a brief sketch of the argument here.

Suppose $\beta \geq 3/8$. We begin with the term $\frac{1}{(1-2\beta)m} \text{Log} \left(\frac{1}{\delta} \right)$. Since we have assumed $|\mathbb{C}| \geq 3$, there must exist $x_0, x_1 \in \mathcal{X}$ and $h_0, h_1 \in \mathbb{C}$ such that $h_0(x_0) = h_1(x_0)$ while $h_0(x_1) \neq h_1(x_1)$. Now fix $\epsilon = \frac{3}{8(1-2\beta)m} \ln \left(\frac{1}{5\delta} \right) \wedge (1-2\beta)$, let $\mathcal{P}(\{x_1\}) = \frac{\epsilon}{1-2\beta}$, and let $\mathcal{P}(\{x_0\}) = 1 - \mathcal{P}(\{x_1\})$. Then, for $b \in \{0, 1\}$, we let P_b be a distribution on $\mathcal{X} \times \mathcal{Y}$ with marginal \mathcal{P} over \mathcal{X} , and with $P_b(\{(x_0, h_0(x_0))\} \times \mathcal{Y}) = 1$ and $P_b(\{(x_1, h_b(x_1))\} \times \mathcal{Y}) = 1 - \beta$. Then one can easily check that, for $\mathcal{P}_{XY} = P_b$, any classifier h with $h(x_1) \neq h_b(x_1)$ has $\text{er}(h) - \inf_{g \in \mathbb{C}} \text{er}(g) \geq \epsilon$. But since $\text{KL}(P_0^m \| P_1^m) = m \text{KL}(P_0 \| P_1) = m\epsilon \ln \left(\frac{1-\beta}{\beta} \right)$, and $\ln \left(\frac{1-\beta}{\beta} \right) \leq \frac{1-\beta}{\beta} - 1 = \frac{1-2\beta}{\beta} \leq \frac{8}{3}(1-2\beta)$ (since $\beta \geq 3/8$), classic hypothesis testing lower bounds (see Tsybakov, 2009, Theorem 2.2) imply that there exists a choice of $b \in \{0, 1\}$ such that, with $\mathcal{P}_{XY} = P_b$ and $\hat{h}_m = \mathcal{A}(\mathcal{L}_m)$, $\mathbb{P}(\hat{h}_m(x_1) \neq h_b(x_1)) \geq \frac{1}{4} \exp \{-m\epsilon \frac{8}{3}(1-2\beta)\} \geq (5/4)\delta > \delta$. Thus, with probability greater than δ , $\text{er}(\hat{h}_m) - \inf_{g \in \mathbb{C}} \text{er}(g) \geq \epsilon \gtrsim \frac{1}{(1-2\beta)m} \text{Log} \left(\frac{1}{\delta} \right)$.

Next, we present a proof for the term $\frac{1}{(1-2\beta)m}$, again for $\beta \geq 3/8$. This term is trivially implied by the term $\frac{1}{(1-2\beta)m} \text{Log} \left(\frac{1}{\delta} \right)$ in the case $d = 1$, so suppose $d \geq 2$. This time, we let $\{x_0, \dots, x_{d-1}\}$ denote a subset of \mathcal{X} shatterable by \mathbb{C} , fix $\epsilon = \frac{3(d-1)}{64\epsilon(1-2\beta)m} \wedge \frac{1-2\beta}{8e}$, and let $\mathcal{P}(\{x_i\}) = \frac{8\epsilon}{(d-1)(1-2\beta)}$ for $i \in \{1, \dots, d-1\}$, and $\mathcal{P}(\{x_0\}) = 1 - \frac{1-2\beta}{1-2\beta}$. Now for each $\bar{b} = (b_1, \dots, b_{d-1}) \in \{0, 1\}^{d-1}$, let $P_{\bar{b}}$ denote a probability measure on $\mathcal{X} \times \mathcal{Y}$ with marginal \mathcal{P} over \mathcal{X} , and with $P_{\bar{b}}(\{(x_i, 2b_i - 1)\} \times \mathcal{Y}) = 1 - \beta$ for every $i \in \{1, \dots, d-1\}$, and $P_{\bar{b}}(\{(x_0, -1)\} \times \mathcal{Y}) = 1$. In particular, note that any $\bar{b}, \bar{b}' \in \{0, 1\}^{d-1}$ with Hamming distance $\|\bar{b} - \bar{b}'\|_1 = 1$ have $\text{KL}(P_{\bar{b}}^m \| P_{\bar{b}'}^m) = m \text{KL}(P_{\bar{b}} \| P_{\bar{b}'}) = m \frac{8\epsilon}{d-1} \ln \left(\frac{1-\beta}{\beta} \right)$, and as above, $\ln \left(\frac{1-\beta}{\beta} \right) \leq \frac{8}{3}(1-2\beta)$. Now Assouad's lemma (see Tsybakov, 2009, Theorem 2.12) implies that there exists a $\bar{b} \in \{0, 1\}^{d-1}$ such that, with $\mathcal{P}_{XY} = P_{\bar{b}}$ and $\hat{h}_m = \mathcal{A}(\mathcal{L}_m)$, denoting $\hat{b} = ((1 + \hat{h}_m(x_1))/2, \dots, (1 + \hat{h}_m(x_{d-1}))/2)$, we have $\mathbb{E}[\|\hat{b} - \bar{b}\|_1] \geq \frac{d-1}{4} \exp \left\{ -m \frac{8\epsilon}{d-1} \frac{8}{3}(1-2\beta) \right\} \geq \frac{d-1}{4e}$. Noting that $0 \leq \|\hat{b} - \bar{b}\|_1 \leq d-1$, this further implies that $\mathbb{P} \left(\|\hat{b} - \bar{b}\|_1 \geq \frac{d-1}{8e} \right) \geq \frac{1}{8e}$. Furthermore, note that $\text{er}(\hat{h}_m) - \inf_{g \in \mathbb{C}} \text{er}(g) \geq \|\hat{b} - \bar{b}\|_1 \frac{8\epsilon}{d-1}$. Thus, $\mathbb{P} \left(\text{er}(\hat{h}_m) - \inf_{g \in \mathbb{C}} \text{er}(g) \geq \epsilon \right) \geq \frac{1}{8e} > \delta$. Finally, note that $\epsilon \gtrsim \frac{d}{(1-2\beta)m} \wedge (1-2\beta)$.

Altogether, by choosing which ever of these lower bounds is greatest, we have that for any $m \in \mathbb{N}$, there exists a choice of \mathcal{P}_{XY} satisfying the β -bounded noise condition such that, with probability greater than δ ,

$$\text{er}(\hat{h}_m) - \inf_{h \in \mathbb{C}} \text{er}(h) \gtrsim \frac{\max \{d, \beta \text{Log}(\min \{s, (1-2\beta)^2 m\}), \text{Log} \left(\frac{1}{\delta} \right)\}}{(1-2\beta)m} \wedge (1-2\beta).$$

Applying the relaxation $\max\{a, b, c\} \geq (1/3)(a + b + c)$ (for nonnegative values a, b, c) then completes the proof of the first lower bound stated in the theorem.

For the second inequality, note that by taking $\delta = 1/24$, the inequality proven above implies that there exists a distribution \mathcal{P}_{XY} satisfying the β -bounded noise condition such that, with probability greater than $1/24$,

$$\text{er}(\hat{h}_m) - \inf_{h \in \mathbb{C}} \text{er}(h) \gtrsim \frac{d + \beta \text{Log}(\min \{s, (1-2\beta)^2 m\})}{(1-2\beta)m} \wedge (1-2\beta).$$

Furthermore, since bounded noise distributions have $\inf_{h \in \mathcal{C}} \text{er}(h)$ equal the Bayes risk, $\text{er}(h_m) - \inf_{h \in \mathcal{C}} \text{er}(h)$ is always nonnegative. We therefore have

$$\begin{aligned} \mathbb{E} \left[\text{er}(\hat{h}_m) - \inf_{h \in \mathcal{C}} \text{er}(h) \right] &\geq \frac{23}{24} - 0 + \frac{1}{24} \frac{d + \beta \text{Log}(\min\{5, (1 - 2\beta)^2 m\})}{(1 - 2\beta)m} \wedge (1 - 2\beta) \\ &\geq \frac{d + \beta \text{Log}(\min\{5, (1 - 2\beta)^2 m\})}{(1 - 2\beta)m} \wedge (1 - 2\beta). \end{aligned}$$

Finally, since $\inf_{h \in \mathcal{C}} \text{er}(h)$ is nonrandom, $\mathbb{E} \left[\text{er}(\hat{h}_m) \right] - \inf_{h \in \mathcal{C}} \text{er}(h) = \mathbb{E} \left[\text{er}(\hat{h}_m) - \inf_{h \in \mathcal{C}} \text{er}(h) \right]$. ■

E.2 Proof of Theorem 19

Next, we present the proof of Theorem 19. We begin by stating a classic result, due to Ginié and Koltchinski (2006) (see also van der Vaart and Wellner, 2011; Hanneke and Yang, 2012). For any set \mathcal{H} of classifiers, denote $\text{diam}_{\mathcal{P}}(\mathcal{H}) = \sup_{h, g \in \mathcal{H}} \mathcal{P}(x : h(x) \neq g(x))$.

Lemma 23 *There is a universal constant $c_0 \in (1, \infty)$ such that, for any set \mathcal{H} of classifiers, for any $\delta \in (0, 1)$ and $m \in \mathbb{N}$, defining*

$$U(\mathcal{H}, m, \delta; R) = 1 \wedge \inf_{r > \text{diam}_{\mathcal{P}}(\mathcal{H})} c_0 \sqrt{r} \frac{\text{vc}(\mathcal{H}) \text{Log}\left(\frac{\mathcal{P}(R)}{r}\right) + \text{Log}\left(\frac{1}{\delta}\right)}{m} + c_0 \frac{\text{vc}(\mathcal{H}) \text{Log}\left(\frac{\mathcal{P}(R)}{r}\right) + \text{Log}\left(\frac{1}{\delta}\right)}{m}$$

for every measurable $R \subseteq \mathcal{X}$, with probability at least $1 - \delta$, $\forall h \in \mathcal{H}$,

$$\begin{aligned} \text{er}(h) - \inf_{g \in \mathcal{H}} \text{er}(g) &\leq \max \left\{ 2 \left(\text{er}_{\mathcal{L}_m}(h) - \min_{g \in \mathcal{H}} \text{er}_{\mathcal{L}_m}(g) \right), U(\mathcal{H}, m, \delta; \text{DIS}(\mathcal{H})) \right\}, \\ \text{er}_{\mathcal{L}_m}(h) - \min_{g \in \mathcal{H}} \text{er}_{\mathcal{L}_m}(g) &\leq \max \left\{ 2 \left(\text{er}(h) - \inf_{g \in \mathcal{H}} \text{er}(g) \right), U(\mathcal{H}, m, \delta; \text{DIS}(\mathcal{H})) \right\}. \end{aligned}$$

Next, we note that we lose very little by requiring the γ function in Definition 15 to be binary. This allows us to simplify certain parts of the proof of Theorem 19 below.

Lemma 24 *For any set \mathcal{H} of classifiers, and any $\eta \in [0, 1]$, for $X \sim \mathcal{P}$, letting*

$$\begin{aligned} \Phi_{(0,1)}(\mathcal{H}, \eta) &= \inf_{h \in \mathcal{H}} \mathbb{E} \left[\mathbb{I}[\gamma(X)] : \sup_{h \in \mathcal{H}} \mathbb{E} \left[\mathbb{I}[h(X) = +1] \zeta(X) + \mathbb{I}[h(X) = -1] \xi(X) \right] \leq \eta, \right. \\ &\quad \left. \text{where } \forall x \in \mathcal{X}, \gamma(x) + \zeta(x) + \xi(x) = 1 \text{ and } \zeta(x), \xi(x) \in [0, 1], \gamma(x) \in \{0, 1\} \right], \end{aligned}$$

we have that

$$\Phi(\mathcal{H}, \eta) \leq \Phi_{(0,1)}(\mathcal{H}, \eta) \leq 2\Phi(\mathcal{H}, \eta/2).$$

Proof The left inequality is clear from the definitions. For the right inequality, let γ^*, ζ^*, ξ^* be the functions at the optimal solution achieving $\Phi(\mathcal{H}, \eta/2)$ in Definition 15. For every

$x \in \mathcal{X}$, if $\gamma^*(x) \geq 1/2$, define $\gamma(x) = 1$ and $\zeta(x) = \xi(x) = 0$, and otherwise define $\gamma(x) = 0$, $\zeta(x) = \zeta^*(x)/(\zeta^*(x) + \xi^*(x))$, and $\xi(x) = \xi^*(x)/(\zeta^*(x) + \xi^*(x))$. By design, we have that $\gamma(x) \in \{0, 1\}$, $\zeta(x), \xi(x) \in [0, 1]$, and $\gamma(x) + \zeta(x) + \xi(x) = 1$ for every $x \in \mathcal{X}$. Since every $x \in \mathcal{X}$ has $\gamma(x) \leq 2\gamma^*(x)$, we have $\mathbb{E}[\gamma(X)] \leq 2\mathbb{E}[\gamma^*(X)] = 2\Phi(\mathcal{H}, \eta/2)$. Furthermore, for every $x \in \mathcal{X}$, we either have $\zeta(x) = 0 \leq 2\zeta^*(x)$ and $\xi(x) = 0 \leq 2\xi^*(x)$, or else $\gamma^*(x) < 1/2$, in which case $\zeta^*(x) + \xi^*(x) = 1 - \gamma^*(x) > 1/2$, so that $\zeta(x) = \zeta^*(x)/(\zeta^*(x) + \xi^*(x)) \leq 2\zeta^*(x)$ and $\xi(x) = \xi^*(x)/(\zeta^*(x) + \xi^*(x)) \leq 2\xi^*(x)$. Therefore,

$$\begin{aligned} \sup_{h \in \mathcal{H}} \mathbb{E} \left[\mathbb{I}[h(X) = +1] \zeta(X) + \mathbb{I}[h(X) = -1] \xi(X) \right] \\ \leq 2 \sup_{h \in \mathcal{H}} \mathbb{E} \left[\mathbb{I}[h(X) = +1] \zeta^*(X) + \mathbb{I}[h(X) = -1] \xi^*(X) \right] \leq \eta. \end{aligned}$$

Thus, γ, ζ, ξ are functions in the feasible region of the optimization problem defining $\Phi_{(0,1)}(\mathcal{H}, \eta)$, so that $\Phi_{(0,1)}(\mathcal{H}, \eta) \leq \mathbb{E}[\gamma(X)] \leq 2\Phi(\mathcal{H}, \eta/2)$. ■

We will establish the claim in Theorem 19 for the following algorithm (which has the data set \mathcal{L}_m as input). For simplicity, this algorithm is stated in a way that makes it \mathcal{P} -dependent (which is consistent with the statement of Theorem 19). It may be possible to remove this dependence by replacing the \mathcal{P} -dependent quantities with empirical estimates, but we leave this task to future work (e.g., see the work of Koltchinski, 2006, for discussion of empirical estimation of $U(\mathcal{H}, m, \delta; R)$; Zhang and Chandhuri, 2014, additionally discuss estimating the minimizing function γ from the definition of Φ , through some refinement to their concentration arguments would be needed for our purposes). For any $k \in \{0, 1, \dots, \lfloor \log_2(m) \rfloor - 1\}$, define $\delta_k = \frac{\delta}{(\log_2(2m) - \delta_k)^2}$, and fix a value $\eta_k \geq 0$ (to be specified in the proof below).

Algorithm 1:

0. $\mathcal{G}_0 \leftarrow \mathcal{C}$
1. For $k = 0, 1, \dots, \lfloor \log_2(m) \rfloor - 1$
2. Let γ_k be the function γ at the solution defining $\Phi_{(0,1)}(\mathcal{G}_k, \eta_k)$
3. $R_k \leftarrow \{x \in \mathcal{X} : \gamma_k(x) = 1\}$
4. $D_k \leftarrow \{(X_i, Y_i) : 2^k + 1 \leq i \leq 2^{k+1}, X_i \in R_k\}$
5. $\mathcal{G}_{k+1} \leftarrow \left\{ h \in \mathcal{G}_k : 2^{-k} |D_k(\text{er}_{D_k}(h) - \min_{g \in \mathcal{G}_k} \text{er}_{D_k}(g)) \leq \max\{4\eta_k, U(\mathcal{G}_k, 2^k, \delta_k, R_k)\} \right\}$
6. Return any $\hat{h} \in \mathcal{G}_{\lfloor \log_2(m) \rfloor}$

For simplicity, we suppose the function γ_k in Step 2 actually minimizes $\mathbb{E}[\gamma_k(X)]$ subject to the constraints in the definition of $\Phi_{(0,1)}(\mathcal{G}_k, \eta_k)$. However, the proof below would remain valid for any γ_k satisfying these constraints, with $\mathbb{E}[\gamma_k(X)] \leq 2\Phi(\mathcal{G}_k, \eta_k/2)$: for instance, the proof of Lemma 24 reveals this would be satisfied by $\gamma_k(x) = \mathbb{I}[\gamma^*(x) \geq 1/2]$ for the γ^* achieving the minimum value of $\mathbb{E}[\gamma^*(X)]$ in the definition of $\Phi(\mathcal{G}_k, \eta_k/2)$. Indeed, it would even suffice to choose γ_k satisfying the constraints of $\Phi_{(0,1)}(\mathcal{G}_k, \eta_k)$ with $\mathbb{E}[\gamma_k(X)] \leq c^k \Phi(\mathcal{G}_k, \eta_k/2)$, for any finite numerical constant c^k , as this would only affect the numerical constant factors in Theorem 19.

We are now ready for the proof of Theorem 19.

Proof of Theorem 19 The proof is similar to those given above (e.g., that of Theorem 16), except that the stronger form of Lemma 23 (compared to Lemma 2) affords us a simplification that avoids the step in which we lower-bound the sample size under the conditional distribution given $\Gamma_i = 1$.

Fix any $a \geq 1$ and $\alpha \in (0, 1]$, and fix $c = 128$. We establish the claim for Algorithm 1, described above. Define $\eta_0 = 2/c$ and $\tilde{U}_0 = 1$, and for each $k \in \{1, \dots, \lfloor \log_2(m) \rfloor\}$, inductively define

$$\begin{aligned}\tilde{U}_k &= \min \left\{ 1, 2\eta_{k-1} + \max \left\{ 8\eta_{k-1}, 2U(\mathcal{G}_{k-1}, 2^{k-1}, \delta_{k-1}; R_{k-1}) \right\} \right\}, \\ r_k &= ac_1 \left(a2^{1-k} \left(d \text{Log} \left(\hat{\varphi}_{a,\alpha} \left(a \left(ad2^{1-k} \right)^{\frac{\alpha}{2-\alpha}} \right) \right) + \text{Log} \left(\frac{1}{\delta_{k-1}} \right) \right) \right)^{\frac{\alpha}{2-\alpha}}, \\ \eta_k &= \frac{2}{c} \left(\frac{r_k}{a} \right)^{1/\alpha},\end{aligned}$$

where $c_1 = (32c_0)^{\frac{2-\alpha}{\alpha}}$. We proceed by induction on k in the algorithm. Suppose that, for some $k \in \{0, 1, \dots, \lfloor \log_2(m) \rfloor - 1\}$, there is an event E_k of probability at least $1 - \sum_{k'=0}^{k-1} \delta_{k'}$ (or probability 1 if $k = 0$), on which $h^* \in \mathcal{G}_k$, and for some universal constant $c_1 \in (1, \infty)$, every $k' \in \{0, \dots, k\}$ has

$$\tilde{U}_{k'} \leq (c/2)\eta_{k'},$$

and

$$\mathcal{G}_{k'} \subseteq \left\{ h \in \mathbb{C} : \text{er}(h) - \text{er}(h^*) \leq \tilde{U}_{k'} \right\}.$$

In particular, these conditions are trivially satisfied for $k = 0$, so this may serve as a base case for this inductive argument. Next we must extend these conditions to $k + 1$.

For each $h \in \mathcal{G}_k$, define $h_{R_k}(x) = h(x)\mathbb{1}[x \in R_k] + h^*(x)\mathbb{1}[x \notin R_k]$, and denote $\mathcal{H}_k = \{h_{R_k} : h \in \mathcal{G}_k\}$. Noting that $R_k \supseteq \text{DIS}(\mathcal{H}_k)$, and that this implies $U(\mathcal{H}_k, 2^k, \delta_k; R_k) \geq U(\mathcal{H}_k, 2^k, \delta_k; \text{DIS}(\mathcal{H}_k))$, Lemma 23 (applied under the conditional distribution given \mathcal{G}_k) and the law of total probability imply that there exists an event E_{k+1}^* of probability at least $1 - \delta_k$, on which, $\forall h_{R_k} \in \mathcal{H}_k$, denoting $\tilde{\mathcal{L}}_k = \{(X_i, Y_i) : 2^k + 1 \leq i \leq 2^{k+1}\}$ (which is distributionally equivalent to \mathcal{L}_{2^k} but independent of \mathcal{G}_k),

$$\begin{aligned}\text{er}_{\tilde{\mathcal{L}}_k}(h_{R_k}) - \inf_{g_{R_k} \in \mathcal{H}_k} \text{er}(g_{R_k}) &\leq \max \left\{ 2 \left(\text{er}_{\tilde{\mathcal{L}}_k}(h_{R_k}) - \min_{g_{R_k} \in \mathcal{H}_k} \text{er}_{\tilde{\mathcal{L}}_k}(g_{R_k}) \right), U(\mathcal{H}_k, 2^k, \delta_k; R_k) \right\}, \\ \text{er}_{\tilde{\mathcal{L}}_k}(h_{R_k}) - \min_{g_{R_k} \in \mathcal{H}_k} \text{er}_{\tilde{\mathcal{L}}_k}(g_{R_k}) &\leq \max \left\{ 2 \left(\text{er}(h_{R_k}) - \inf_{g_{R_k} \in \mathcal{H}_k} \text{er}(g_{R_k}) \right), U(\mathcal{H}_k, 2^k, \delta_k; R_k) \right\}.\end{aligned}$$

First we note that, since every h_{R_k} and g_{R_k} in \mathcal{H}_k agree on the labels of all samples in $\tilde{\mathcal{L}}_k \setminus D_k$, and they each agree with their respective classifiers h and g in \mathcal{G}_k on D_k , we have that

$$\text{er}_{\tilde{\mathcal{L}}_k}(h_{R_k}) - \min_{g_{R_k} \in \mathcal{H}_k} \text{er}_{\tilde{\mathcal{L}}_k}(g_{R_k}) = 2^{-k}|D_k| \left(\text{er}_{D_k}(h) - \min_{g \in \mathcal{G}_k} \text{er}_{D_k}(g) \right).$$

Next, let ζ_k and ξ_k denote the functions ζ and ξ from the definition of $\Phi_{(0,1)}(\mathcal{G}_k, \eta_k)$ at the solution with γ equal γ_k . Note that ζ_k and ξ_k are themselves random, but are completely

determined by \mathcal{G}_k . The definition of R_k guarantees that for every $h, g \in \mathcal{G}_k$, for $X \sim \mathcal{P}$ (independent from \mathcal{L}_m)

$$\begin{aligned}\mathcal{P}(x \notin R_k : h(x) \neq g(x)) &= \mathbb{E}[\mathbb{1}[h(X) \neq g(X)](\zeta_k(X) + \xi_k(X))|\mathcal{G}_k] \\ &= \mathbb{E}[\mathbb{1}[h(X) = +1]\mathbb{1}[g(X) = -1] + \mathbb{1}[h(X) = -1]\mathbb{1}[g(X) = +1]](\zeta_k(X) + \xi_k(X))|\mathcal{G}_k] \\ &\leq \mathbb{E}[\mathbb{1}[h(X) = +1]\zeta_k(X) + \mathbb{1}[h(X) = -1]\xi_k(X)|\mathcal{G}_k] \\ &\quad + \mathbb{E}[\mathbb{1}[g(X) = +1]\zeta_k(X) + \mathbb{1}[g(X) = -1]\xi_k(X)|\mathcal{G}_k] \leq 2\eta_k.\end{aligned}$$

Therefore,

$$\text{er}(h_{R_k}) - \text{er}(g_{R_k}) \leq \text{er}(h) - \text{er}(g) + \mathcal{P}(x \notin R_k : h(x) \neq g(x)) \leq \text{er}(h) - \text{er}(g) + 2\eta_k,$$

and similarly

$$\text{er}(h_{R_k}) - \text{er}(g_{R_k}) \geq \text{er}(h) - \text{er}(g) - \mathcal{P}(x \notin R_k : h(x) \neq g(x)) \geq \text{er}(h) - \text{er}(g) - 2\eta_k.$$

In particular, noting that $\text{er}(h_{R_k}) - \inf_{g_{R_k} \in \mathcal{H}_k} \text{er}(g_{R_k}) = \sup_{g \in \mathcal{G}_k} \text{er}(h_{R_k}) - \text{er}(g_{R_k})$ and $\sup_{g \in \mathcal{G}_k} \text{er}(h) - \text{er}(g) = \text{er}(h) - \inf_{g \in \mathcal{G}_k} \text{er}(g)$, this implies

$$\text{er}(h) - \inf_{g \in \mathcal{G}_k} \text{er}(g) - 2\eta_k \leq \text{er}(h_{R_k}) - \inf_{g_{R_k} \in \mathcal{H}_k} \text{er}(g_{R_k}) \leq \text{er}(h) - \inf_{g \in \mathcal{G}_k} \text{er}(g) + 2\eta_k.$$

We also note that $\text{vc}(\mathcal{H}_k) \leq \text{vc}(\mathcal{G}_k)$ and $\text{diam}_{\mathcal{P}}(\mathcal{H}_k) \leq \text{diam}_{\mathcal{P}}(\mathcal{G}_k)$, which together imply $U(\mathcal{H}_k, 2^k, \delta_k; R_k) \leq U(\mathcal{G}_k, 2^k, \delta_k; R_k)$. Altogether, we have that on E_{k+1}^* , $\forall h \in \mathcal{G}_k$,

$$\begin{aligned}\text{er}(h) - \inf_{g \in \mathcal{G}_k} \text{er}(g) &\leq 2\eta_k + \max \left\{ 2^{1-k}|D_k| \left(\text{er}_{D_k}(h) - \min_{g \in \mathcal{G}_k} \text{er}_{D_k}(g) \right), U(\mathcal{G}_k, 2^k, \delta_k; R_k) \right\}, \\ 2^{-k}|D_k| \left(\text{er}_{D_k}(h) - \min_{g \in \mathcal{G}_k} \text{er}_{D_k}(g) \right) &\leq \max \left\{ 2 \left(\text{er}(h) - \inf_{g \in \mathcal{G}_k} \text{er}(g) + 2\eta_k \right), U(\mathcal{G}_k, 2^k, \delta_k; R_k) \right\}.\end{aligned}$$

In particular, defining $E_{k+1} = E_{k+1}^* \cap E_k$, we have that on E_{k+1} , $h^* \in \mathcal{G}_k$, and

$$2^{-k}|D_k| \left(\text{er}_{D_k}(h^*) - \min_{g \in \mathcal{G}_k} \text{er}_{D_k}(g) \right) \leq \max \left\{ 4\eta_k, U(\mathcal{G}_k, 2^k, \delta_k; R_k) \right\},$$

so that $h^* \in \mathcal{G}_{k+1}$ as well. Furthermore, combined with the definition of \mathcal{G}_{k+1} , this further implies that on E_{k+1} ,

$$\begin{aligned}\mathcal{G}_{k+1} &\subseteq \left\{ h \in \mathbb{C} : \text{er}(h) - \text{er}(h^*) \leq 2\eta_k + \max \left\{ 8\eta_k, 2U(\mathcal{G}_k, 2^k, \delta_k; R_k) \right\} \right\} \\ &= \left\{ h \in \mathbb{C} : \text{er}(h) - \text{er}(h^*) \leq \tilde{U}_{k+1} \right\}.\end{aligned}$$

It remains only to establish the bound on \tilde{U}_{k+1} . For this, we first note that, combining the inductive hypothesis with the (a, α) -Bernstein class condition, on E_{k+1} we have

$$\mathcal{G}_k \subseteq \mathbb{B} \left(h^*, a\tilde{U}_k^\alpha \right) \subseteq \mathbb{B} \left(h^*, r_k \right).$$

Combining this with Lemma 24 and monotonicity of $\Phi(\cdot, \eta_k/2)$, we have that

$$\mathcal{P}(R_k) \leq 2\Phi \left(\mathbb{B} \left(h^*, r_k \right), \eta_k/2 \right) = 2\Phi \left(\mathbb{B} \left(h^*, r_k \right), (r_k/a)^{1/\alpha}/c \right) \leq 2\hat{\varphi}_{a,\alpha}(r_k)r_k.$$

The above also implies that $\text{diam}_{\mathcal{P}}(\mathcal{G}_k) \leq 2r_k$ on E_{k+1} . Together with the fact that $\text{vc}(\mathcal{G}_k) \leq d$, we have that on E_{k+1} ,

$$\begin{aligned} U(\mathcal{G}_k, 2^k, R_k) &\leq c_0 \sqrt{2r_k 2^{-k} \left(d \text{Log}(\hat{\varphi}_{\alpha, \alpha}(r_k)) + \text{Log}\left(\frac{1}{\delta_k}\right) \right)} \\ &\quad + c_0 2^{-k} \left(d \text{Log}(\hat{\varphi}_{\alpha, \alpha}(r_k)) + \text{Log}\left(\frac{1}{\delta_k}\right) \right). \end{aligned} \quad (34)$$

Furthermore, monotonicity of $\hat{\varphi}_{\alpha, \alpha}(\cdot)$ implies $\hat{\varphi}_{\alpha, \alpha}(r_k) \leq \hat{\varphi}_{\alpha, \alpha}(a(ad2^{-k})^{\frac{2}{2-\alpha}})$. Plugging the definition of r_k into (34) along with this relaxation of $\hat{\varphi}_{\alpha, \alpha}(r_k)$ and simplifying, the minimum of 1 and the right hand side of (34) is at most

$$\begin{aligned} 8c_0 \sqrt{c_1} \left(a 2^{-k} \left(d \text{Log}\left(\hat{\varphi}_{\alpha, \alpha}\left(a(ad2^{-k})^{\frac{2}{2-\alpha}}\right)\right) + \text{Log}\left(\frac{1}{\delta_k}\right) \right) \right)^{\frac{1}{2-\alpha}} \\ = 8c_0 \sqrt{c_1} \left(\frac{r_{k+1}}{c_1 a} \right)^{1/\alpha} = \frac{4c_0 c}{c_1^{\frac{2-\alpha}{2\alpha}}} \eta_{k+1} = \frac{c}{8} \eta_{k+1}. \end{aligned}$$

We may also observe that

$$\eta_k \leq 4^{2^{-\alpha}} \eta_{k+1} \leq 4/\eta_{k+1}.$$

Combining the above with the definition of \tilde{U}_{k+1} , we have that on E_{k+1} ,

$$\tilde{U}_{k+1} \leq 8r_{k+1} + \max\left\{32\eta_{k+1}, \frac{c}{4}\eta_{k+1}\right\} = 40\eta_{k+1} \leq 64\eta_{k+1} = \frac{c}{2}\eta_{k+1}.$$

Finally, noting that the union bound implies E_{k+1} has probability at least $1 - \sum_{k'=0}^k \delta_{k'}$ completes the inductive step.

By the principle of induction, we have thus established that, on an event $E_{\lfloor \log_2(m) \rfloor}$ of probability at least $1 - \sum_{k=0}^{\lfloor \log_2(m) \rfloor - 1} \delta_k > 1 - \delta \sum_{i=2}^{\infty} \frac{1}{i^2} > 1 - \delta$,

$$h^* \in \mathcal{G}_{\lfloor \log_2(m) \rfloor} \subseteq \left\{ h \in \mathcal{C} : \text{er}(h) - \text{er}(h^*) \leq \frac{c}{2} \eta_{\lfloor \log_2(m) \rfloor} \right\}.$$

In particular, this implies that \hat{h} exists in Step 6, and satisfies $\text{er}(\hat{h}) - \inf_{g \in \mathcal{C}} \text{er}(g) = \text{er}(\hat{h}) - \text{er}(h^*) \leq \frac{c}{2} \eta_{\lfloor \log_2(m) \rfloor}$. Noting that

$$\begin{aligned} \frac{c}{2} \eta_{\lfloor \log_2(m) \rfloor} &\leq c_1^{1/\alpha} \left(\frac{4a \left(d \text{Log}\left(\hat{\varphi}_{\alpha, \alpha}\left(a\left(\frac{ad}{m}\right)^{\frac{2}{2-\alpha}}\right)\right) + \text{Log}\left(\frac{4}{\delta}\right) \right)}{m} \right)^{\frac{1}{2-\alpha}} \\ &\leq 6(32c_0)^2 \left(\frac{a \left(d \text{Log}\left(\hat{\varphi}_{\alpha, \alpha}\left(a\left(\frac{ad}{m}\right)^{\frac{2}{2-\alpha}}\right)\right) + \text{Log}\left(\frac{4}{\delta}\right) \right)}{m} \right)^{\frac{1}{2-\alpha}} \end{aligned}$$

completes the proof. \blacksquare

References

- N. Alon, R. Beigleiter, and E. Ezra. Active learning using smooth relative regret approximations with applications. *Journal of Machine Learning Research*, 15(3):885–920, 2014. 6
- M. Anthony and P. L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999. E.1
- P. Auer and R. Ortner. A new PAC bound for intersection-closed concept classes. In *Proceedings of the 17th Conference on Learning Theory*, 2004. 1, 2, 3
- P. Auer and R. Ortner. A new PAC bound for intersection-closed concept classes. *Machine Learning*, 66(2-3):151–163, 2007. 1, 1.2, 1.2, 3, 3
- M.-F. Balcan and P. M. Long. Active and passive learning of linear separators under log-concave distributions. In *Proceedings of the 26th Conference on Learning Theory*, 2013. 1.2, 5.2, 5.2, 5.2, 6.2
- M.-F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. In *Proceedings of the 23rd International Conference on Machine Learning*, 2006. 4
- M.-F. Balcan, A. Broder, and T. Zhang. Margin based active learning. In *Proceedings of the 20th Conference on Learning Theory*, 2007. 1.2, 5.2, 5.2
- M.-F. Balcan, A. Beygelzimer, and J. Langford. Agnostic active learning. *Journal of Computer and System Sciences*, 75(1):78–89, 2009. 4
- P. Bartlett, M. I. Jordan, and J. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006. 6
- P. L. Bartlett and S. Mendelson. Discussion: Local Rademacher complexities and oracle inequalities in risk minimization. *The Annals of Statistics*, 34(6):2657–2663, 2006. 6
- A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the Association for Computing Machinery*, 36(4):929–965, 1989. 1.2, 2, 5, 5.1
- A. Bowers and N. J. Kalton. *An Introductory Course in Functional Analysis*. Springer, 2014. 9
- N. H. Bshouty, Y. Li, and P. M. Long. Using the doubling dimension to analyze the generalization of learning algorithms. *Journal of Computer and System Sciences*, 75(6):323–335, 2009. 5.2
- D. Cohn, L. Atlas, and R. Laderer. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994. 1.2, 4
- T. M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, EC-14(3):326–334, 1965. 5.2

- M. Darnstädt. The optimal PAC bound for intersection-closed concept classes. *Information Processing Letters*, 115(4):458–461, 2015. 1.2, 3
- S. Dasgupta, A. T. Kalai, and C. Monteleoni. Analysis of perceptron-based active learning. In *Proceedings of the 18th Conference on Learning Theory*, 2005. 5.2
- S. Dasgupta, D. Hsu, and C. Monteleoni. A general agnostic active learning algorithm. In *Advances in Neural Information Processing Systems* 20, 2007. 4
- A. Ehrenfeucht, D. Haussler, M. Kearns, and L. Valiant. A general lower bound on the number of examples needed for learning. *Information and Computation*, 82(3):247–261, 1989. 1.2, 1.2, 5.1, E.1
- R. El-Yaniv and Y. Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(5):1605–1641, 2010. 4, 4, 4
- R. El-Yaniv and Y. Wiener. Agnostic selective classification. In *Advances in Neural Information Processing Systems* 24, 2011. 6
- R. El-Yaniv and Y. Wiener. Active learning via perfect selective classification. *Journal of Machine Learning Research*, 13(2):255–279, 2012. 1.2, 4, 4, 5.2
- S. Floyd and M. Warmuth. Sample compression, learnability, and the Vapnik-Chervonenkis dimension. *Machine Learning*, 21(3):269–304, 1995. 1.2, 2, 2
- E. Giné and V. Koltchinskii. Concentration inequalities and asymptotic results for ratio type empirical processes. *The Annals of Probability*, 34(3):1143–1216, 2006. 1.2, 1.2, 5, 5.2, 6, 6, E.2
- A. Gupta, R. Krauthgamer, and J. R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, 2003. 5.2
- S. Hanneke. Teaching dimension and the complexity of active learning. In *Proceedings of the 20th Conference on Learning Theory*, 2007a. 4
- S. Hanneke. A bound on the label complexity of agnostic active learning. In *Proceedings of the 24th International Conference on Machine Learning*, 2007b. 4, 4
- S. Hanneke. *Theoretical Foundations of Active Learning*. PhD thesis, Machine Learning Department, School of Computer Science, Carnegie Mellon University, 2009. 1, 1.2, 1.2, 1.2, 2, 4, 4, 5, 4, 6, D
- S. Hanneke. Rates of convergence in active learning. *The Annals of Statistics*, 39(1):333–361, 2011. 1.2, 4, 6, E.1
- S. Hanneke. Activized learning: Transforming passive to active with improved label complexity. *Journal of Machine Learning Research*, 13(5):1469–1587, 2012. 4, 6
- S. Hanneke. Theory of disagreement-based active learning. *Foundations and Trends in Machine Learning*, 7(2–3):131–309, 2014. 1.2, 4, 6, E.1
- S. Hanneke. The optimal sample complexity of PAC learning. *Journal of Machine Learning Research*, 17(38):1–15, 2016. 1, 1.2, 1.2, 8
- S. Hanneke and L. Yang. Surrogate losses in passive and active learning. *arXiv:1207.3772*, 2012. 1.2, 4, 6, E.2
- S. Hanneke and L. Yang. Minimax analysis of active learning. *Journal of Machine Learning Research*, 16(12):3487–3602, 2015. 1.2, 1.2, 4, 4, 4, 5, 5.1, 5.2, 6, 6, 6.1, 6.1, 6.1, D.1, D.2, E.1, 10, E.1, E.1
- D. Haussler. Sphere packing numbers for subsets of the Boolean n-cube with bounded Vapnik-Chervonenkis dimension. *Journal of Combinatorial Theory A*, 69(2):217–232, 1995. 5.2, D.2, D.2
- D. Haussler, N. Littlestone, and M. Warmuth. Predicting $\{0, 1\}$ -functions on randomly drawn points. *Information and Computation*, 115(2):248–292, 1994. 1.2, 1.2, 3, 5.1
- D. Helmbold, R. Sloan, and M. Warmuth. Learning nested differences of intersection-closed concept classes. *Machine Learning*, 5(2):165–196, 1990. 3
- R. Herbrich. *Learning Kernel Classifiers*. The MIT Press, Cambridge, MA, 2002. 2
- A. N. Kolmogorov and V. M. Tikhomirov. ε -entropy and ε -capacity of sets in function spaces. *Uspekhi Matematicheskikh Nauk*, 14(2):3–86, 1959. D.2
- A. N. Kolmogorov and V. M. Tikhomirov. ε -entropy and ε -capacity of sets in function spaces. *American Mathematical Society Translations, Series 2*, 17:277–364, 1961. D.2
- V. Koltchinskii. Local Rademacher complexities and oracle inequalities in risk minimization. *The Annals of Statistics*, 34(6):2593–2656, 2006. 6, 7, 6.2, E.2
- V. Koltchinskii. Rademacher complexities and bounding the excess risk in active learning. *Journal of Machine Learning Research*, 11(9):2457–2485, 2010. 4
- C. Kuhlmann. On teaching and learning intersection-closed concept classes. In *Proceedings of the 12th Conference on Learning Theory*, 1999. 3
- L. LeCam. Convergence of estimates under dimensionality restrictions. *The Annals of Statistics*, 1(1):38–53, 1973. 5.2
- Y. Li, P. M. Long, and A. Srinivasan. The one-inclusion graph algorithm is near-optimal for the prediction model of learning. *IEEE Transactions on Information Theory*, 47(3):1257–1261, 2001. 1.2
- N. Littlestone and M. Warmuth. Relating data compression and learnability. *Unpublished manuscript*, 1986. 1.2, 2, 2
- P. M. Long. An upper bound on the sample complexity of PAC learning halfspaces with respect to the uniform distribution. *Information Processing Letters*, 87(5):229–234, 2003. 1.2

- E. Mannen and A.B. Tsybakov. Smooth discrimination analysis. *The Annals of Statistics*, 27(6):1808–1829, 1999. 6
- P. Massart and E. Nédélec. Risk bounds for statistical learning. *The Annals of Statistics*, 34(5):2326–2366, 2006. 1.2, 6, 6.1, 6.1, E.1
- B. K. Natarajan. On learning Boolean functions. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, 1987. 1.2, 3
- M. Raginsky and A. Rakhlin. Lower bounds for passive and active learning. In *Advances in Neural Information Processing Systems 24*, 2011. 1.2, 6.1, 6.1, E.1, 10
- N. Sauer. On the density of families of sets. *Journal of Combinatorial Theory (A)*, 13(1): 145–147, 1972. D.2
- A. B. Tsybakov. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1):135–166, 2004. 6
- A. B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009. E.1
- A. van der Vaart and J. A. Wellner. A local maximal inequality under uniform entropy. *Electronic Journal of Statistics*, 5:192–203, 2011. 1.1, E.2
- A. W. van der Vaart and J. A. Wellner. *Weak Convergence and Empirical Processes*. Springer, 1996. 1.1
- V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971. 1.1, D.2
- V. Vapnik and A. Chervonenkis. *Theory of Pattern Recognition*. Nauka, Moscow, 1974. 1.2, 2, 5
- M. Vidyasagar. *Learning and Generalization with Applications to Neural Networks*. Springer-Verlag, 2nd edition, 2003. 3, D.2, E.1
- M. Warmuth. The optimal PAC algorithm. In *Proceedings of the 17th Conference on Learning Theory*, 2004. 1.2
- Y. Wiener, S. Hanneke, and R. El-Yaniv. A compression technique for analyzing disagreement-based active learning. *Journal of Machine Learning Research*, 16(4):713–745, 2015. 1.2, 2, 4, 4, 2, 4, 4, 4, 4, 4, 5
- Y. Yang and A. Barron. Information-theoretic determination of minimax rates of convergence. *The Annals of Statistics*, 27(5):1564–1599, 1999. 5.2
- C. Zhang and K. Chandhuri. Beyond disagreement-based agnostic active learning. In *Advances in Neural Information Processing Systems 27*, 2014. 1.2, 5.2, 5.2, 5.2, 6, 6.2, 6.2, D, E.2

In memory of Alexey Chervonensis

Synergy of Monotonic Rules

Vladimir Vapnik

Columbia University
New York, NY 10027, USA
Facebook AI Research
New York, NY 10017, USA

VLADIMIR.VAPNIK@GMAIL.COM

Rauf Izmailov

Applied Communication Sciences
Basking Ridge, NJ 07920-2021, USA

RIZMAILOV@APPCOMSCI.COM

Editor: Andreas Christmann

Abstract

This article describes a method for constructing a special rule (we call it synergy rule) that uses as its input information the outputs (scores) of several monotonic rules which solve the same pattern recognition problem. As an example of scores of such monotonic rules we consider here scores of SVM classifiers.

In order to construct the optimal synergy rule, we estimate the conditional probability function based on the direct problem setting, which requires solving a Fredholm integral equation. Generally, solving a Fredholm equation is an ill-posed problem. However, in our model, we look for the solution of the equation in the set of monotonic and bounded functions, which makes the problem well-posed. This allows us to solve the equation accurately even with training data sets of limited size.

In order to construct a monotonic solution, we use the set of functions that belong to Reproducing Kernel Hilbert Space (RKHS) associated with the INK-spline kernel (splines with Infinite Numbers of Knots) of degree zero. The paper provides details of the methods for finding multidimensional conditional probability in a set of monotonic functions to obtain the corresponding synergy rules. We demonstrate effectiveness of such rules for

- 1) solving standard pattern recognition problems,
- 2) constructing multi-class classification rules,
- 3) constructing a method for knowledge transfer from multiple intelligent teachers in the LUPI paradigm.

Keywords: conditional probability, synergy, ensemble learning, intelligent teacher, privileged information, knowledge transfer, support vector machines, SVM+, classification, learning theory, kernel functions, regression

1. Introduction

The standard setting of pattern recognition problem requires, in the given set of functions $f(x, \alpha)$, $\alpha \in \Lambda$ defined in the space $X \in R^n$, to find the function $f(x, \alpha_0)$ such that the indicator function $y = \theta(f(x, \alpha)) \in \{0, 1\}$ (in this paper, the indicator function is defined

as follows: $\theta(x) = 0$ for $x < 0$ and $\theta(x) = 1$ for $x \geq 0$) minimizes the loss functional

$$R(\alpha) = \int |y - f(x, \alpha)| dp(x, y)$$

if the probability measure $p(x, y)$, $x \in X$, $y \in \{0, 1\}$ is unknown but iid data

$$(x_1, y_1), \dots, (x_\ell, y_\ell), \quad x_i \in X, y_i \in \{0, 1\}$$

generated according to $p(x, y) = p(y|x)p(x)$ are given (in the standard pattern recognition terminology, conditional probability $P(y|x)$ defines an unknown law of classification given by Teacher and $P(x)$ defines an unknown generator of events that should be classified by the learning machine).

In this article, we illustrate our approach using Support Vector Machine (SVM) algorithms. The SVM algorithm constructs an approximation of the desired classification function by first mapping vectors $x \in X$ into vectors $z \in Z$ and then constructing a separating hyperplane in space (Z, y) . The obtained rule is used for classification of unknown iid vectors distributed according to the same unknown probability measure $p(x, y)$.

The conditional probability of class $y = 1$ given x depends on the position of vector z relative to the obtained hyperplane

$$s_i = (w, z_i) + b,$$

where w, b are the parameters estimated by SVM: if $s_i \geq 0$, vector z_i belongs to class $y_i = 1$, otherwise it belongs to the opposite class $y_i = 0$.

As Platt (1999) observed, the smaller is the (negative) score s_i for vector z_i , the closer is the conditional probability $P(y = 1|s_i)$ to zero and, the larger is the (positive) score s_i , the closer is the conditional probability $P(y = 1|s_i)$ to one. Platt introduced a method for mapping SVM scores into values of conditional probability based on two hypotheses, a general one and a special one.

The general hypothesis: Conditional probability function $p(y = 1|s)$ is a monotonic function of variable s .

The special hypothesis: Conditional probability function can be approximated well with sigmoid functions with two parameters:

$$P(y = 1|s) = \frac{1}{1 + \exp\{-As + B\}}, \quad A, B \in R^1.$$

Using the maximum likelihood technique, Platt (1999) introduced effective methods to estimate both parameters A, B (see Lin et al., 2007).

Platt's approach was shown to be useful for calibration of SVM scores. Nevertheless, this method has certain drawbacks: even if the conditional probability function for SVM is monotonically increasing, it does not necessarily have the form of a two-parametric sigmoid function.

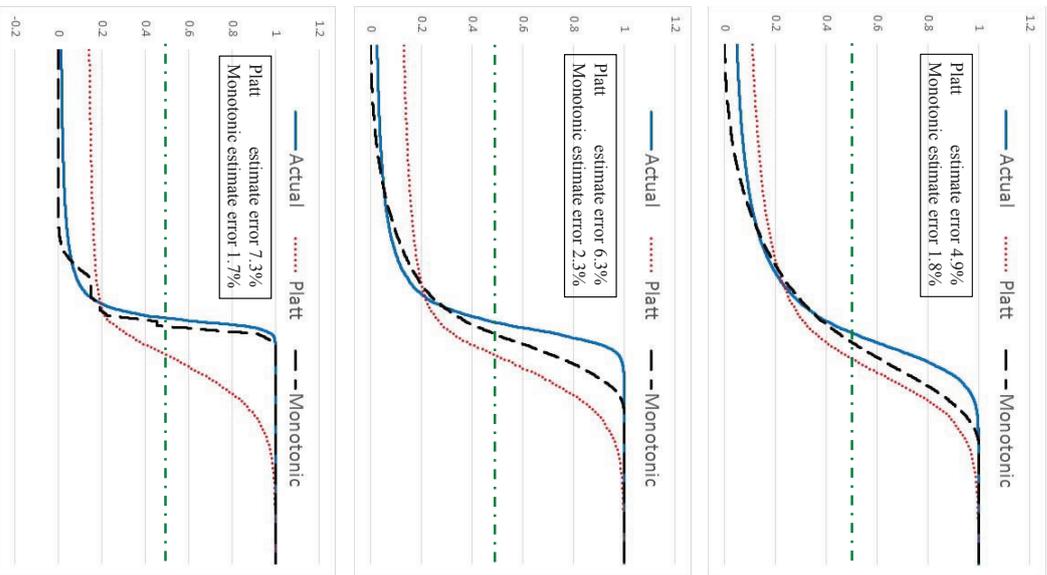


Figure 1: Comparison of conditional probability estimates.

It is easy to construct examples where suggested sigmoid function does not approximate well the desired monotonic conditional probability function. Figure 1 illustrates the conditional probability approximations (for a sample consisting of 96 random numbers that are evenly split between both classes) for Platt’s approach and for the special one-dimensional case of the algorithm described further in this paper.

The one-dimensional problem mentioned in the previous paragraph has the following form: given pairs (values s_i of SVM scores and corresponding classifications y_i)

$$(s_1, y_1), \dots, (s_t, y_t),$$

find an accurate approximation of the monotonic conditional probability function $p(y = 1|s)$. Section 3 describes a technique for construction of a monotonic approximation of the desired function. This approximation provides a more accurate estimate than the one based on sigmoid functions.

In this paper, we consider a more general (and more important) problem than this one-dimensional one. Suppose we have d different SVMs, solving the same classification problem. Also, suppose that the probability of class $y = 1$ given scores $s = (s^1, \dots, s^d)$ of d SVMs is a multidimensional monotonic conditional probability function: for any coordinate k and any fixed values of the other coordinates $(s^1, \dots, s^{k-1}, s^{k+1}, \dots, s^d)$, the higher is the value of score s^k , the higher is the probability $P(y = 1|s)$.

The goal of this article is to find a method for estimation of the *monotonic* conditional probability function $P(y = 1|s)$ for multidimensional vectors $s = (s^1, \dots, s^d)$; that is, to combine, in a single probability value, the results of multiple (namely, d) SVMs. We show that estimating conditional probability function in a set of monotonic functions has a significant advantage over estimating conditional probability function in a general, non-monotonic set of functions: it forms a *well-posed* problem rather than an *ill-posed* problem.

The decision rule for a two-class pattern recognition problem can be obtained using the estimated conditional probability function $P(y = 1|s)$ as

$$y = \theta \left(P(y = 1|s) - \frac{1}{2} \right).$$

This article is organized as follows. In Section 2, we consider the problem of estimating conditional probability function. We show that the problem of conditional probability estimation in general sets of functions is *ill-posed*. However, this problem is *well-posed* for sets of nonnegative bounded (by 1) monotonic functions. Therefore, the problem of estimating the monotonic conditional probability function can be solved more accurately than the general problem of estimating conditional probability function. In Section 3, we describe methods of estimating monotonic conditional probability functions. In Section 4, we apply methods of estimating monotonic conditional probability function based on the scores generated by several different SVMs solving the same pattern recognition problem. Here we estimate monotonic conditional probability function of class $y = 1$ given all the scores and we obtain the so-called *synergy rule* of classification. In Section 5, we consider a method for knowledge transfer from multiple intelligent teachers.

Remark. It is important to note that, in classical machine learning literature, there are *ensemble methods* that combine several rules (see Dietterich (2000), Zhang and Ma (2012),

Tsybakov (2003), Lecué (2007)). The difference between ensemble rules and synergy rules is in the following:

- 1) Ensemble rule is a result of structural combination (such as voting or weighted aggregation) of several classification rules.
- 2) Synergy rule defines the *optimal* solution to the problem of combining several scores of *monotonic rules*. It is based on effective methods of conditional probability estimation in the set of monotonic functions.
- 3) Synergy rule is constructed only for *monotonic rules* (such as SVM) in contrast to ensemble rule which combines *any* rules. Synergy is the property of monotonicity of the solution.

2. Overview of Methods

In this section, we present a short overview of the direct constructive setting of estimation of conditional probability, as presented in (Vapnik and Izmailov, 2015c) and (Vapnik and Izmailov, 2015a). The method is quite general, and, in this section, we do not even assume that the probability has to belong to $[0, 1]$.

2.1 Glivenko-Cantelli Theory

In (Vapnik et al., 2015), (Vapnik and Izmailov, 2015c), (Vapnik and Izmailov, 2015a), we introduced direct constructive methods for solving the main problems of statistical inference. All these methods are based on Glivenko-Cantelli theory, which forms the foundation of classical statistics. This theory states that the *joint cumulative distribution function of several variables* $X = (X^1, \dots, X^n)$

$$F(x) = P\{X^1 \leq x^1, \dots, X^n \leq x^n\}$$

can be estimated from the observations

$$X_1^1, \dots, X_\ell^1$$

by *empirical cumulative distribution function*

$$F_\ell(x) = \frac{1}{\ell} \sum_{k=1}^{\ell} \prod_{i=1}^n \theta(x^k - X_i^i), \quad (1)$$

where $\theta(x^k - X_i^i)$ is the step function (indicator function for $x \geq 0$).

Classical statistical theory provides bounds on the rate of convergence of $F_\ell(x)$ to the desired function $F(x)$ for the one-dimensional case (Massart, 1990):

$$P\left\{\sup_x |F_\ell(x) - F(x)| \geq \varepsilon\right\} \leq 2 \exp\{-2\varepsilon^2 \ell\}. \quad (2)$$

Application of VC theory to n -dimensional case (Vapnik, 1998) gives the bound

$$P\left\{\sup_x |F_\ell(x) - F(x)| \geq \varepsilon\right\} \leq \exp\left\{-\left(\varepsilon^2 - \frac{n \ln \ell}{\ell}\right) \ell\right\}. \quad (3)$$

2.2 Direct Setting of Conditional Probability Estimation

Estimation of cumulative distribution function using empirical data is a foundation for estimation of more sophisticated characteristics of stochastic events such as *density function*, *conditional density function*, *regression function*, *conditional probability function* etc.

1. We call function $p(x)$ the *density function* of the random events $X \sim F(x)$ if its integral defines the cumulative distribution function

$$\int \theta(x - X)p(X)dX = F(x), \quad X \in R^n.$$

2. Let pair (x, y) , $x \in X$, $y \in Y \in R^1$ be a random event. We call

$$p(y|x) = \frac{p(x, y)}{p(x)}, \quad p(x) > 0,$$

the conditional density function; it defines the conditional density of the value of y given observation x .

3. Let pair (x, y) , $x \in X$, $y \in \{0, 1\}$ be a random event. We call

$$p(y = 1|x) = \frac{p(x, y = 1)}{p(x)}, \quad p(x) > 0,$$

the conditional probability function; it defines conditional probability of $y = 1$ given observation x .

4. We call the integral

$$f(x) = \int yp(y|x)dy,$$

the regression function $f(x)$; it defines conditional expectation of value y given observation x .

The definition of conditional probability can be rewritten (see Vapnik and Izmailov (2015c), Vapnik and Izmailov (2015a)) in the form of the solution of integral equation

$$\int \theta(x - X)p(y = 1|X)dF(X) = F(x, y = 1). \quad (4)$$

These papers describe the direct constructive way of estimation of the conditional probability function as solving a multidimensional Fredholm integral equation (4) when cumulative distribution functions $F(x)$ and $F(x, y = 1)$ are unknown but data

$$(x_1, y_1), \dots, (x_\ell, y_\ell)$$

are given. This setting is called *direct* because it is based on the definition of conditional probability. It is called *constructive* because there exists an *empirical cumulative distribution function*, defined as (1), that converges to the real cumulative distribution function with the rates (2) and (3).

In order to find the conditional probability (i.e., to solve equation (4)), we use the approximations $F_k(x)$ and $F_k(x, y = 1) = F_k(x|y = 1)P_k(y = 1)$ instead of unknown functions $F(x)$, $F(x, y = 1)$. As shown in (Vapnik and Izmailov, 2015c) and (Vapnik and Izmailov, 2015a), statistical inference problems, such as (1) conditional density estimation, (2) conditional probability estimation, (3) regression estimation, (4) ratio of two densities estimation, can be formulated as follows: solve the integral equation

$$\int \theta(z - Z)F(z)dF(z) = (E\eta)^{-1}F^*(z)$$

in the situation when the cumulative distribution functions $F(z)$, $F^*(z)$, and the value $E(\eta)$ are unknown but their approximations in the form of empirical cumulative functions $F_k(z)$, $F_k^*(z)$ and empirical average $P_k(y = 1)$ can be obtained using data

$$(z_1, y_1), \dots, (z_\ell, y_\ell).$$

2.3 Fredholm Integral Equations of the First Kind

We consider the linear operator equations

$$Af = \Phi, \tag{5}$$

where A maps elements f of the metric space E_1 into elements Φ of the metric space E_2 . Let A be a continuous one-to-one operator, which maps a set $M \subset E_1$ onto a set $N \subset E_2$, i.e., $AM = N$. The solution of such operator equation exists and is unique, i.e., inverse operator A^{-1} is defined:

$$M = A^{-1}N.$$

The crucial question is whether this inverse operator A^{-1} is continuous. If it is, then close functions in N are mapped by A^{-1} to close functions in M : that is, a ‘‘small’’ change in the right-hand side of (5) results in a ‘‘small’’ change of its solution. In this case, the operator A^{-1} is called *stable* (Tikhonov and Arsenin, 1977). If, however, the inverse operator is discontinuous, then ‘‘small’’ changes in the right-hand side of (5) can cause a significant change of its solution. In this case, the operator A^{-1} is *unstable*.

The equation (5) is *well-posed* if its solution (1) exists, (2) is unique, and (3) is stable. Otherwise, the equation (5) is *ill-posed*.

We are interested in the situation when the solution of operator equation *exists*, and *is unique*. In this case, the *stability* of the operator A^{-1} determines whether (5) is ill-posed or well-posed. If the operator is unstable, then, generally speaking, any numerical solution of (5) is meaningless (a small error in the right-hand side of (5) can cause a large change of its solution).

Here we consider the linear integral operator

$$Af(x) = \int_a^b K(x, u)f(u)du$$

defined by the kernel $K(t, u)$, which is a symmetric positive definite function that is continuous almost everywhere on $a \leq t \leq b$, $c \leq x \leq d$. This kernel maps the set of functions

$\{f(t)\}$, continuous on $[a, b]$, unto the set of functions $\{\Phi(x)\}$, also continuous on $[c, d]$. The corresponding Fredholm equation of the first kind (Tikhonov and Arsenin, 1977)

$$\int_a^b K(x, u)f(u)du = \Phi(x)$$

requires finding the solution $f(u)$ given the right-hand side $\Phi(x)$. It is known that these integral equations are ill-posed.

In our problem, not only the right-hand side of (5) is an approximation but also the operator of (5) is defined approximately. In (Vapnik, 1995), such equations are called stochastic ill-posed problems.

2.4 Methods of Solving Ill-Posed Problems

In this subsection, we consider methods for solving ill-posed operator equations.

2.4.1 INVERSE OPERATOR LEMMA

The following Inverse Operator Lemma (see Tikhonov and Arsenin, 1977) is the key enabler for solving ill-posed problems.

Lemma. *If A is a continuous one-to-one operator defined on a compact set $M^* \subset M$, then the inverse operator A^{-1} is continuous on the set $N^* = AM^*$.*

It is known that bounded monotonic functions form a compact set. Therefore, if we restrict the set of solutions of Fredholm integral equations to the class of bounded monotonic functions, we will make the corresponding equation well-posed. This is exactly the reason of our targeting the *monotonic* solutions in this paper.

Thus, as follows from Inverse Operator Lemma, the conditions of existence and uniqueness of the solution of an operator equation imply that the problem is well-posed on the compact M^* . The third condition (stability of the solution) is automatically satisfied. This lemma is the basis for all constructive ideas of solving ill-posed problems. We describe one of them in the next subsection.

2.4.2 REGULARIZATION METHOD

Suppose that we have to solve the operator equation (4) defined by a continuous one-to-one operator A mapping M into N , where we assume that the solution of (5) exists. Also, suppose that, instead of the right-hand side $\Phi(x)$, we are given its approximations $\Phi_\delta(x)$, where

$$\rho_{E_2}(\Phi(x), \Phi_\delta(x)) \leq \delta.$$

Our goal is to solve the equations

$$Af = \Phi_\delta$$

when $\delta \rightarrow 0$.

Consider a lower semi-continuous functional $W(f)$ (called the *regularizer*) that has the following three properties:

1. the solution of (5) belongs to the domain $D(W)$ of the functional $W(f)$;

2. the values $W(f)$ of W are non-negative in the domain of W ;
3. the sets $\mathcal{M}_c = \{f : W(f) \leq c\}$ are compact for any $c \geq 0$.

The idea of regularization is to find a solution for (5) as an element minimizing the so-called regularized functional

$$R_\gamma(\hat{f}, \Phi_\delta) = \rho_{E_2}^2(A\hat{f}, \Phi_\delta) + \gamma_\delta W(\hat{f}), \quad \hat{f} \in D(W) \quad (6)$$

with regularization parameter $\gamma_\delta > 0$.

The following theorem holds true (Tikhonov and Arsenin, 1977).

Theorem 1. *Let E_1 and E_2 be metric spaces, and suppose that, for $\Phi \in \mathcal{N}$, there exists a solution of (5) that belongs to compact \mathcal{M}_c for some c . Suppose that, instead of the exact right-hand side Φ in (5), its approximations¹ $\Phi_\delta \in E_2$ are such that $\rho_{E_2}(\Phi, \Phi_\delta) \leq \delta$. Consider the sequence of parameters γ such that*

$$\gamma(\delta) \rightarrow 0 \text{ for } \delta \rightarrow 0, \text{ and } \lim_{\delta \rightarrow 0} \frac{\delta^2}{\gamma(\delta)} \leq r < \infty. \quad (7)$$

Then the sequence of solutions $f_\delta^{(\delta)}$ minimizing the functionals $R_{\gamma(\delta)}(f, \Phi_\delta)$ on $D(W)$ converges to the exact solution f (in the metric of space E_1) as $\delta \rightarrow 0$.

In a Hilbert space, the functional $W(f)$ may be chosen as $\|f\|^2$ for a linear operator A . Although the sets \mathcal{M}_c are only weakly compact in this case, regularized solutions converge to the desired one. Such a choice of regularized functional is convenient since its domain $D(W)$ is the whole space E_1 . In this case, however, the conditions on the parameters γ are more restrictive than in the case of Theorem 1: γ should converge to zero slower than δ^2 . Thus the following theorem holds true (Tikhonov and Arsenin, 1977).

Theorem 2. *Let E_1 be a Hilbert space and $W(f) = \|f\|^2$. Then, if $\gamma(\delta)$ satisfies (7) with $r = 0$, the regularized elements $f_\delta^{(\delta)}$ converge to the exact solution f in E_1 as $\delta \rightarrow 0$.*

2.4.3 STOCHASTIC ILL-POSED PROBLEMS

Let A_ℓ be approximations of A and Φ_ℓ be approximations of Φ . In order to solve stochastic ill-posed problems

$$A_\ell f = \Phi_\ell, \quad (8)$$

we will also use the regularization method minimizing the functional

$$T_\ell f = \|A_\ell f - \Phi_\ell\|_{E_2}^2 + \gamma_\ell \Omega(f). \quad (9)$$

Here, with increasing number of observations ℓ , functions Φ_ℓ converge to the actual function Φ and operator A_ℓ converges to the actual operators A in the sense that

$$\|A_\ell - A\|^2 = \sup_{f \in \{\Omega(f) \leq C\}} \frac{\|A_\ell f - A f\|^2}{\Omega(f)} \rightarrow \ell \rightarrow \infty 0. \quad (10)$$

As was shown in (Vapnik (1998)), if the desired solution belongs to one of the compacts $\{f; \Omega(f) \leq C\}$, the sequence of approximations Φ_ℓ of the right-hand side of equation and

¹. The elements Φ_δ do not have to belong to the set \mathcal{N} .

the sequence of approximations A_ℓ of the operators converge in probability to, respectively, Φ and A , and $\gamma_\ell \rightarrow 0$ in (9) is such that

$$\lim_{\ell \rightarrow \infty} \frac{\|A_\ell - A\|^2}{\gamma_\ell} = 0,$$

then the sequence of minima converges to the desired function.

In (Vapnik and Izmailov (2015c)), (Vapnik and Izmailov (2015a)), it was shown that our specific integral equations satisfy the required conditions.

2.5 V-Matrix Method of Estimation of Conditional Probability Function

In order to find the conditional probability from the observations, we solve stochastic ill-posed problem (8) using regularization method (9), where approximations of the right-hand side of equation Φ and operator A are defined. We define two terms of (9) as:

1. the square of the distance $\rho^2(A_\ell f, \Phi_\ell)$ in space E_2 between functions (we omit the common normalizing multiplier $1/\ell$ in these definitions since it does not affect the subsequent derivations):

$$A_\ell f(x) = \sum_{i=1}^{\ell} \theta(x - X_i) f(X_i) \quad \text{and} \quad \Phi_\ell(x) = \sum_{j=1}^{\ell} y_j \theta(x - X_j),$$

where $(X_i, y_i), \dots, i = 1, \dots, \ell, X_i \in R^d, y_i \in \{0, 1\}$ are training data;

2. the regularization functional $\Omega(f)$, to be defined below.

2.5.1 CHOICE OF DISTANCE AND DEFINITION OF V-MATRIX

Below, we use L_2 -distance in space E_2 in the general form

$$\rho^2(A_\ell f, \Phi_\ell) = \int (A_\ell f(x) - \Phi_\ell(x))^2 \sigma(x) d\mu(x),$$

where $\sigma(x)$ is a non-negative function and $\mu(x)$ is a probability measure; some choices for $\sigma(x)$ and $\mu(x)$ were considered in (Vapnik and Izmailov (2015c)), (Vapnik and Izmailov (2015a)). To simplify computations, we chose

$$\sigma(x) = \prod_{k=1}^d \sigma_k(x^k) \quad \text{and} \quad \mu(x) = \prod_{k=1}^d \mu_k(x^k), \quad \text{where } x = (x^1, \dots, x^d).$$

Then we can rewrite the square of distance ρ^2 in the explicit form

$$\int \left(\sum_{i=1}^{\ell} f(X_i) \prod_{k=1}^d \theta(x^k - X_i^k) - \sum_{j=1}^{\ell} y_j \prod_{k=1}^d \theta(x^k - X_j^k) \right)^2 \prod_{k=1}^d \sigma_k(x^k) d\mu_k(x^k) = \sum_{i,j=1}^{\ell} f(X_i) f(X_j) V_{i,j} - 2 \sum_{i,j=1}^{\ell} y_j f(X_i) V_{i,j} + \sum_{i,j=1}^{\ell} y_i y_j V_{i,j},$$

where we have denoted

$$V_{i,j} = \prod_{k=1}^{\ell} V_{k,j}^k, \quad V_{k,j}^k = \int \theta \left(x^k - \min\{X_k^k, X_j^k\} \right) \sigma_k(x^k) d\mu(x^k).$$

We denote by V the $(\ell \times \ell)$ -dimensional matrix of elements $V_{i,j}$, by \mathbf{f} the ℓ -dimensional vector $\mathbf{f} = (f(X_1), \dots, f(X_\ell))^T$, and by Y the ℓ -dimensional vector $Y = (y_1, \dots, y_\ell)^T$. In matrix notations, we can rewrite the square of distance as follows:

$$\rho^2 = \mathbf{f}^T V \mathbf{f} - 2\mathbf{f}^T V Y + Y^T V Y.$$

2.5.2 CHOICE OF REGULARIZATION FUNCTIONAL

Suppose that the solution of our integral equation (4) belongs to the RKHS (Reproducing Kernel Hilbert Space) associated with kernel $K(x, x^*)$ (symmetric positive definite function of vector variables $x, x^* \in X$). This means that RKHS has inner product such that for any function $f(x)$ from the space, the equality

$$(K(\cdot, y), f) = f(y)$$

holds true. According to Mercer theorem, any positive definite kernel $K(x, x^*)$ can be represented as

$$K(x, x^*) = \sum_{k=1}^{\infty} \lambda_k \phi_k(x) \phi_k(x^*),$$

where $\{\phi_k(x)\}$ is a system of orthonormal functions in E_1 and $\{\lambda_k\}$ is a sequence of non-negative values converging to zero, where $k = 1, 2, \dots$

It is easy to check that the functions

$$f(x, a) = \sum_{k=1}^{\infty} a_k \phi_k(x),$$

belong to RKHS associated with kernel $K(x, x^*)$ if the inner product between two functions $f(x, a)$ and $f(x, b)$ has the form

$$(f(x, a), f(x, b)) = \sum_{k=1}^{\infty} \frac{a_k b_k}{\lambda_k},$$

and, therefore, the norm of function $f(x, a)$ is

$$\|f(x, a)\|^2 = \sum_{k=1}^{\infty} \frac{a_k^2}{\lambda_k}. \quad (11)$$

We will chose the norm of function from RKHS as the regularizer $\Omega(f) = \|f\|^2$. As follows from (11), the set of functions with their norm bounded by C

$$\|f(x, a)\|^2 = \sum_{k=1}^{\infty} \frac{a_k^2}{\lambda_k} \leq C$$

is a compact. Therefore, we use as a regularizer in (9) the norm of function in RKHS

$$\rho^2 + \gamma_\ell \|f\|^2 = \mathbf{f}^T V \mathbf{f} - 2\mathbf{f}^T V Y + Y^T V Y + \gamma_\ell \|f\|^2. \quad (12)$$

An important property of RKHS for applications is defined by the so-called Representer Theorem (Kornfeldorff and Wahba (1970)), according to which the minimum of (12) has an expansion on elements $K(x_i, x)$ defined on the training data x_1, \dots, x_ℓ

$$f(x, a) = \sum_{i=1}^{\ell} \alpha_i K(x_i, x), \quad (13)$$

and the norm of function f in RKHS is defined as

$$\|f\|^2 = \sum_{i,j=1}^{\ell} \alpha_i \alpha_j K(x_i, x_j). \quad (14)$$

To simplify the notations, we introduce ℓ -dimensional vector $\Lambda = (\alpha_1, \dots, \alpha_\ell)$, ℓ -dimensional vector functions $K(x) = (K(x_1, x), \dots, K(x_\ell, x))^T$ and $(\ell \times \ell)$ -dimensional matrix $K = (K(x_i, x_j))$.

Using these notations, we can rewrite (13) and (14) as

$$f(x) = K^T(x) \Lambda, \quad \mathbf{f} = K \Lambda, \quad \|f\|^2 = \Lambda^T K \Lambda.$$

2.5.3 V-MATRIX KERNEL REGRESSION

In order to solve our integral equation using the regularization technique, we have to minimize, with respect to vector Λ , the functional

$$W(\Lambda) = \Lambda^T K V K \Lambda - 2\Lambda^T K V Y + \gamma_\ell \Lambda^T K \Lambda; \quad (15)$$

in this functional, the third term of (12) was omitted since it does not depend on Λ . The solution has the form

$$f(x) = \Lambda^T K(x), \quad (16)$$

where one has to minimize functional (15) in order to find Λ .

The minimum of (15) has the closed-form representation

$$\Lambda = (V K + \gamma_\ell I)^{-1} V Y. \quad (17)$$

Note that for $y_i \in \{0, 1\}$ in $Y = (y_1, \dots, y_\ell)^T$, expression (17) estimates the conditional probability; for $y_i \in R^1$, this expression estimates the regression.

2.6 Estimation in a Set of Functions with Bias Term

Below we consider sets of functions $\{f(x) + b\}$, where b is a value of bias (to be estimated from data) and function $f(x)$ belongs to RKHS (note that $f(x) + b$ does not have to belong to RKHS). Replacing $f(x)$ with $f(x) + b$ in (16), we can rewrite (15) in the form

$$W = (K \Lambda + b \mathbf{1}_\ell)^T V (K \Lambda + b \mathbf{1}_\ell) - 2(K \Lambda + b \mathbf{1}_\ell)^T V Y + \gamma_\ell \Lambda^T K \Lambda. \quad (18)$$

Finding the expression for b by minimizing (18)

$$b = \frac{1}{\ell} \mathbf{1}_\ell^T (Y - K\Lambda) \quad (19)$$

and putting it into equation (18), we obtain the functional for minimization:

$$\begin{aligned} W = & \left(K\Lambda + \frac{1}{\ell} \mathbf{1}_\ell \mathbf{1}_\ell^T (Y - K\Lambda) \right)^T \left(K\Lambda + \frac{1}{\ell} \mathbf{1}_\ell \mathbf{1}_\ell^T (Y - K\Lambda) \right) - \\ & 2 \left(K\Lambda + \frac{1}{\ell} \mathbf{1}_\ell \mathbf{1}_\ell^T (Y - K\Lambda) \right)^T VY + \gamma_\ell \Lambda^T K\Lambda. \end{aligned} \quad (20)$$

In order to simplify this expression, we introduce the notations

$$E = I - \frac{1}{\ell} \mathbf{1}_\ell \mathbf{1}_\ell^T \quad \text{and} \quad \mathcal{V} = EV E.$$

Then

$$W = \Lambda^T K \mathcal{V} K \Lambda - 2\Lambda^T K \mathcal{V} Y + \gamma \Lambda^T K \Lambda + C, \quad (21)$$

where C are the terms that do not depend on Λ . Taking the derivative of W over Λ and equating it to zero, we see that, in order to minimize (21), vector Λ has to satisfy the equation

$$2K \mathcal{V} K \Lambda - 2K \mathcal{V} Y + 2\gamma K \Lambda = 0.$$

Solving this equation with respect to Λ , we obtain the closed-form solution

$$\Lambda = (\mathcal{V} K + \gamma I)^{-1} \mathcal{V} Y, \quad (22)$$

which differs from (17) just by using matrix \mathcal{V} instead of matrix V .

Therefore, in order to find the conditional probability in the form $f(x) = \Lambda^T \mathcal{K}(x) + b$, we have to estimate the vector Λ using (22) and estimate the bias b using (19).

Remark. The described solution for conditional probability is also applicable for estimating regression. In that case, coordinates y_i of vector $Y = (y_1, \dots, y_\ell)^T$ belong to \mathbb{R}^1 and the set of functions $f(x, \alpha)$, $\alpha \in \Lambda$ is a set of real-valued functions from RKHS.

2.7 Indirect Methods of Estimation of Conditional Probability

In addition to direct setting of conditional probability problem, indirect settings also exist. They are based on the fact that for some loss functions $\rho(y - f(x, \alpha))$, under a wide range of conditions, the minimum of the functional

$$R = \int \rho(y - f(x, \alpha)) d\rho(x, y) \quad (23)$$

in the set $f(x, \alpha)$, $\alpha \in \Lambda$ defines conditional probability function $f(x, \alpha_0)$ (provided that $\alpha_0 \in \Lambda$). In order to estimate the conditional probability, one has to find the function that minimizes functional (23) if the probability measure $P\rho(x, y)$ is unknown but iid sample

$$(x_1, y_1), \dots, (x_\ell, y_\ell) \quad (24)$$

is given. The standard idea for solving this problem is to minimize the functional

$$\sum_{i=1}^{\ell} \rho(y_i - f(x_i, \alpha)) + \gamma \|f(x, \alpha)\|^2.$$

There are two classical ideas of choosing the term $\rho(y - f(x, \alpha))$:

1. $\rho(y - f(x, \alpha)) = (y - f(x, \alpha))^2$, which leads to regularized *kernel least square* method.
2. $\rho(y - f(x, \alpha)) = |y - f(x, \alpha)|$, which leads to a more robust regularized *kernel least modulo* method.

2.7.1 REGULARIZED KERNEL LEAST SQUARE METHOD

We minimize the functional (23) based on empirical data (24) in the set of functions belonging to RKHS associated with the kernel $K(x, x^*)$. For this set, we minimize the empirical functional

$$\sum_{i=1}^{\ell} (y_i - f(x_i, \alpha) - b)^2 + \gamma \|f(x, \alpha)\|^2.$$

Minimizing this expression over b , we obtain

$$\sum_{i=1}^{\ell} (f(x_i, \alpha) + b) = \sum_{i=1}^{\ell} y_i,$$

where we again assume that functions $f(x, \alpha)$, $\alpha \in \Lambda$ belong to RKHS associated with kernel $K(x, x^*)$. Using the same reasoning as in the previous section, we obtain that the solution has the form (16), with the expansion coefficients $\Lambda = (\alpha_1, \dots, \alpha_\ell)^T$ maximizing the functional

$$W = \Lambda^T K I K \Lambda - 2\Lambda^T K I Y + \gamma \Lambda^T K \Lambda,$$

where we have denoted

$$\mathcal{I} = E I E.$$

The vector of coefficients Λ in closed form is

$$\Lambda = (\mathcal{I} K + \gamma I)^{-1} \mathcal{I} Y.$$

2.7.2 REGULARIZED KERNEL LEAST MODULO METHOD

In classical statistics, besides L_2 -norm loss function for estimating regression, L_1 -norm loss is considered as well. In many situations, L_1 -norm regression has an advantage over L_2 -norm: it provides the so-called *robust regression* (Andersen (2008)). As in previous sections, we estimate the regression in the set of functions $\{f(x, \alpha) + b\}$, where each $f(x, \alpha)$ belongs to RKHS associated with kernel $K(x, x^*)$. In order to do that, we minimize the functional

$$R = C \sum_{i=1}^{\ell} |y_i - f(x_i, \alpha) - b| + \|f(x, \alpha)\|^2.$$

We rewrite this problem in an equivalent form: we map vectors $x \in X$ into Hilbert space $z \in Z$ defined by the inner product $\langle z_i, z_j \rangle = K(x_i, x_j)$ given by a non-negative definite kernel $K(x, x_*)$. We look for a solution in the form $f(x, \alpha) = (w, z) + b$, where $w, z \in Z$. In these notations, we rewrite our minimization problem as follows: minimize the functional

$$R = C \sum_{i=1}^{\ell} \xi_i + (w, w)$$

subject to the constraints

$$-\xi_i \leq y_i - (w, z_i) - b \leq \xi_i, \quad i = 1, \dots, \ell.$$

Using Lagrange multiplier method, we construct the Lagrangian

$$\mathcal{L} = C \sum_{i=1}^{\ell} \xi_i + (w, w) - \sum_{i=1}^{\ell} \alpha_i [y_i - (w, z_i) - b] + \xi_i - \sum_{i=1}^{\ell} \alpha_i^+ [-y_i + (w, z_i) + b] + \xi_i],$$

the saddle point of which (minimum with respect to ξ and w and maximum with respect to α) defines the solution.

The solution has the form

$$f(x, \alpha) = \sum_{i=1}^{\ell} \delta_i K(x_i, x) + b,$$

where, in order to find $\delta_i = \alpha_i^* - \alpha_i$, one has to maximize the functional

$$R = \sum_{i=1}^{\ell} y_i \delta_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \delta_i \delta_j K(x_i, x_j)$$

subject to the constraints

$$-C \leq \delta_i \leq C, \quad \sum_{i=1}^{\ell} \delta_i = 0.$$

The bias b can be computed as

$$b = y_k - \sum_{i=1}^{\ell} \delta_i K(x_i, x_k),$$

where k is an index for which $|\delta_k| \neq C$.

3. Estimation of Monotonic Conditional Probability Functions

Our goal is to minimize functional (15) in the set of monotonically increasing functions. We do this by using expansion of desired function on kernels that generate splines with infinite number of knots (INK-spline) of degree zero. The reason we use these kernels is that they enable an efficient and straightforward construction of multidimensional monotonic functions: it is possible that some other kernels might be used for that purpose as well.

3.1 Kernels for Estimating INK-Splines

According to the definition in the one-dimensional case, splines of degree r with m knots are defined by the expansion (in this section, we assume that $0 \leq x \leq 1$)

$$S(x|r, m) = \sum_{s=0}^r c_s x^s + \sum_{k=0}^m e_k (x - a_k)_+^r, \quad (25)$$

where

$$(x - a_k)_+^r = \begin{cases} (x - a_k)^r & \text{if } x - a_k \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

We generalize expansion (25) using infinite number of knots:

$$S_{\infty}(x) = \sum_{s=0}^r c_s x^s + \int_0^{\infty} g(\tau) (x - \tau)_+^r d\tau.$$

Following the approach from (Vapnik (1998)), (Izmailov et al. (2013)), we define the kernel with infinite number of knots (INK-spline) of degree r for expansion of the function of one variable $x \geq 0$ in the form

$$K_r(x_i, x_j) = \int_0^{\infty} (x_i - \tau)_+^r (x_j - \tau)_+^r d\tau = \sum_{k=0}^r \frac{C_r^k}{2^r - k + 1} [\min\{x_i, x_j\}]^{2r-k+1} |x_i - x_j|^k$$

(here we modified the definition of INK-kernel from (Vapnik (1998)), (Izmailov et al. (2013)) by omitting its polynomial portion).

For $r = 0$, the INK-spline kernel has the form

$$K_0(x_i, x_j) = \min\{x_i, x_j\}; \quad (26)$$

for $r = 1$, the INK-spline kernel has the form

$$K_1(x_i, x_j) = \frac{1}{3} (\min\{x_i, x_j\})^3 + \frac{1}{2} (\min\{x_i, x_j\})^2 |x_i - x_j|.$$

In the multidimensional case, the INK-spline of degree r is defined as

$$K_r(x_i, x_j) = \prod_{k=1}^d K_{r_k}(x_i^k, x_j^k), \quad x = (x^1, \dots, x^d).$$

3.2 Estimating One-Dimensional Monotonic Conditional Probability Function

In classical statistics, there are methods for estimation of monotonic (isotonic) regression (see Best and Chakravarti, Mair et al. (2009), Sysoev et al. (2011), Meyer (2013)), focusing on maintaining the monotonicity on the observed sample points. Below we describe a method of estimating conditional probability that is a monotonic function in the whole space; the method is based on INK-splines with infinite number of knots.

We estimate the monotonic conditional probability function in the set of INK-splines of degree zero (piecewise constant spline function with infinite number of knots). We start with one-dimensional case where $x \geq 0$. For this kernel, the solution is defined as

$$f(x) = \Lambda^T \mathcal{K}(x) + b = \sum_{i=1}^{\ell} \alpha_i \min\{x_i, x\} + b. \quad (27)$$

To specify monotonically increasing function, we impose additional constraints for (27): specifically, we consider the subset of functions (27) for which the inequality

$$\frac{df(x, \alpha)}{dx} \geq 0, \quad \forall x \geq 0 \quad (28)$$

is valid. Since any function (27) is a piecewise linear continuous function, in order for it to be monotonic, it is sufficient for that function to satisfy the constraints

$$\begin{aligned} \frac{df(x_j, \alpha)}{dx} &= \sum_{i=1}^{\ell} \alpha_i \theta(x_i - x_j) \geq 0, \quad j = 1, \dots, \ell, \\ \frac{df(0, \alpha)}{dx} &= \sum_{i=1}^{\ell} \alpha_i \geq 0. \end{aligned} \quad (29)$$

Indeed, consider three possible cases:

1. Let $x \leq \min\{x_1, \dots, x_L\}$. Then, since $\theta(x_i - x) = 1$ for all $i = 1, \dots, L$, the value (28) is non-negative according to the second inequality in (29).
2. Let $\min\{x_1, \dots, x_L\} < x < \max\{x_1, \dots, x_L\}$. Without loss of generality, assume the ordering $x_1 \leq x_2 \leq \dots \leq x_L$ and the position of x within that ordering as $x_j \leq x \leq x_{j+1}$. The function (28) is linear on the interval (x_j, x_{j+1}) and its values at the ends of the interval are

$$\sum_{i=1}^L \alpha_i \theta(x_i - x_j) \quad \text{and} \quad \sum_{i=1}^L \alpha_i \theta(x_i - x_{j+1}),$$

which are non-negative, according to the first inequality in (29). Therefore, the function is also non-negative at any internal point x of the interval (x_j, x_{j+1}) .

3. Let $x \geq \max\{x_1, \dots, x_L\}$. Then, since $\theta(x_i - x) = 0$ for all $i = 1, \dots, L$, the value (28) is zero.

We introduce the notations

$$\Theta(x_j) = (\theta(x_1 - x_j), \dots, \theta(x_\ell - x_j))^T, \quad j = 1, \dots, \ell.$$

Using these notations, we rewrite the constraints (29) in the form

$$\Lambda^T \Theta(0) \geq 0, \quad \Lambda^T \Theta(x_j) \geq 0, \quad j = 1, \dots, \ell. \quad (30)$$

3.2.1 ESTIMATING MONOTONIC CONDITIONAL PROBABILITY USING V-MATRIX METHOD

In order to find a monotonic solution, we use our method for estimating conditional probability function with INK-spline kernel of degree zero with additional ℓ monotonicity constraints (30). That is, we have to minimize the functional

$$W = (K\Lambda + b\mathbf{1}_\ell)^T V(K\Lambda + b\mathbf{1}_\ell) - 2(K\Lambda + b\mathbf{1}_\ell)^T VY + \gamma_\ell \Lambda^T K\Lambda \quad (31)$$

(here coordinates of vector Y are $y_i \in \{0, 1\}$ subject to $\ell + 1$ inequality constraints

$$\Lambda^T \Theta(0) \geq 0, \quad \Lambda^T \Theta(x_j) \geq 0, \quad j = 1, \dots, \ell.$$

Let $x \geq 0$. Then, in order to construct the conditional probability in the set of non-negative monotonic functions bounded by the value 1, we have to enforce the constraint $P(y = 1|x) \leq 1$. Thus, taking into account nonnegativity and monotonicity (31), we add the constraint

$$\Lambda^T \mathcal{K}(x = 1) + b = \Lambda^T \bar{\mathbf{x}} + b \leq 1, \quad (32)$$

where $\bar{\mathbf{x}} = (x_1, \dots, x_\ell)^T$.

3.2.2 ESTIMATING MONOTONIC CONDITIONAL PROBABILITY USING L₂-NORM SVM

Using L_2 -norm SVM for estimating monotonic conditional probability function, we minimize the functional

$$W(\Lambda) = \Lambda^T K K \Lambda - 2\Lambda^T K Y + \gamma_\ell \Lambda^T K \Lambda,$$

with coordinates of Y are $y_i \in \{0, 1\}$ subject to $\ell + 2$ inequality constraints (31), (32).

3.2.3 ESTIMATING MONOTONIC CONDITIONAL PROBABILITY USING L₁-NORM SVM

We look for a solution of the following quadratic optimization problem: minimize the functional

$$C \sum_{i=1}^{\ell} |y_i - f(x_i, \alpha) - b| + \|f(x, \alpha)\|^2$$

subject to $\ell + 2$ inequality constraints (31) and (32), where $f(x, \alpha)$ belongs to RKHS associated with the kernel INK-spline of degree zero $K(x_i, x_j) = \min(x_i, x_j)$.

In matrix form, this problem can be rewritten as follows: minimize the functional

$$R(\xi, \Lambda) = C \mathbf{1}_\ell^T \xi + \gamma \Lambda^T K \Lambda$$

subject to the constraints

$$-\xi \leq Y - K\Lambda - b\mathbf{1}_\ell \leq \xi,$$

and $\ell + 2$ constraints

$$\Lambda^T \Theta(0) \geq 0, \quad \Lambda^T \Theta(x_i) \geq 0, \quad i = 1, \dots, \ell,$$

$$\Lambda^T \bar{\mathbf{x}} + b \leq 1,$$

where we have denoted

$$\bar{\xi} = (\xi_1, \dots, \xi_\ell)^T, \quad \bar{x} = (x_1, \dots, x_\ell)^T, \quad Y = (y_1, \dots, y_\ell)^T.$$

3.3 Estimating Multidimensional Monotonic Conditional Probability Functions

3.3.1 ESTIMATION OF MONOTONIC FUNCTIONS FROM RKHS ASSOCIATED WITH MULTIPLICATIVE KERNELS

In multidimensional case, $x_i = (x_i^1, \dots, x_i^d)^T \in H \subset \mathbb{R}^d$, where we can assume (by proper normalization) that $H = [0, 1]^d$. We consider the solution of the equation in the form

$$f(x) = \sum_{i=1}^{\ell} \alpha_i K(x_i, x) + b,$$

where the kernel generating d -dimensional INK-spline of degree zero has the multiplicative form

$$K(x_i, x) = \prod_{k=1}^d \min(x_i^k, x^k).$$

We have

$$f(x) = \sum_{i=1}^{\ell} \alpha_i \prod_{k=1}^d \min(x_i^k, x^k). \quad (33)$$

Note that the set of functions (33) is monotonic in H if d inequalities

$$\frac{df(x)}{dx^k} = \sum_{i=1}^{\ell} \alpha_i \theta(x_i^k - x^k) \prod_{m \neq k} \min(x_i^m, x^m) \geq 0, \quad k = 1, \dots, d \quad (34)$$

hold true for an function and any $x = (x^1, \dots, x^d) \in H$. To keep the matrix notations, we consider diagonal matrix D_k with diagonal elements h_{ii}

$$h_{ii} = \prod_{m \neq k} \min(x_i^m, x^m), \quad i = 1, \dots, \ell, \quad k = 1, \dots, d.$$

Using this notation, we rewrite (34) as

$$\frac{df(x)}{dx^k} = A^T D_k \Theta(x^k) \geq 0, \quad k = 1, \dots, d \quad (35)$$

for all $x \in H$.

In order to find the monotonic conditional probability function, we minimize

$$R(\Lambda, b) = (K\Lambda + b\mathbf{1}_\ell)^T V (K\Lambda + b\mathbf{1}_\ell) - 2(K\Lambda + b\mathbf{1}_\ell)^T VY + \gamma(A^T K\Lambda) \quad (36)$$

subject to d constrains

$$\inf_{x \in H} A^T D_k \Theta(x^k) \geq 0, \quad k = 1, \dots, d.$$

This is a difficult problem to solve. Instead, one could construct an approximate solution by using Monte Carlo ideas: consider $N \approx n^d$ random (or pseudo-random) elements $x_i =$

$(x_i^1, \dots, x_i^d)^T$, $t = 1, \dots, N$ belonging to H (Sobol points (Jäckel (2004)) could be used, for instance) and instead of d constrains (35) consider Nd constrains

$$A^T D_k \Theta(x_i^k) \geq 0, \quad k = 1, \dots, d, \quad t = 1, \dots, N. \quad (37)$$

As in the one-dimensional case, in order to enforce that the value of conditional probability does not exceed one, we add one more constraint

$$A^T \bar{\mathbf{x}}^* + b \leq 1, \quad (38)$$

where we have denoted by $\bar{\mathbf{x}}^*$ the ℓ -dimensional vector of products

$$\bar{\mathbf{x}}^* = \left[\left(\prod_k x_1^k \right), \dots, \left(\prod_k x_\ell^k \right) \right]^T.$$

Therefore, in order to construct d -dimensional *approximation* of monotonic conditional probability function, one has to minimize functional (36), subject to Nd constrains (37) and one constraint (38).

3.3.2 ESTIMATING MONOTONIC FUNCTIONS FROM RKHS ASSOCIATED WITH ADDITIVE KERNELS

Along with functions defined by (multiplicative) INK-spline kernels of degree zero (26) that can construct approximations to monotonic functions, we consider functions defined by the additive kernel (which is a sum of one-dimensional kernels)

$$f(x) = \sum_{i=1}^{\ell} \sum_{k=1}^d \alpha_i^k \min(x_i^k, x^k) + b. \quad (39)$$

In order to find $d \times \ell$ coefficients α_i^k , $k = 1, \dots, d$, $i = 1, \dots, \ell$ of expansion in estimating conditional probability function in the direct setting, we minimize the functional

$$R(\Lambda_1, \dots, \Lambda_\ell, b) = \left[\sum_{k=1}^d K_k \Lambda_k + b\mathbf{1}_\ell \right]^T V \left[\sum_{k=1}^d K_k \Lambda_k + b\mathbf{1}_\ell \right] - 2 \left[\sum_{k=1}^d K_k \Lambda_k + b\mathbf{1}_\ell \right]^T VY + \gamma \sum_{k=1}^d (\Lambda_k^T K_k \Lambda_k) \quad (40)$$

subject to $d \times (\ell + 1)$ inequality constrains

$$\begin{aligned} \frac{\partial f(x_j, \alpha)}{\partial x^k} &= \sum_{i=1}^{\ell} \alpha_i^k \theta(x_i^k - x^k) = \Lambda_k^T \Theta(x_j^k) \geq 0, \quad j = 1, \dots, \ell, \quad k = 1, \dots, d, \\ \frac{\partial f(\bar{0}, \alpha)}{\partial x^k} &= \sum_{i=1}^{\ell} \alpha_i^k = \Lambda_k^T \Theta(\bar{0}^k) \geq 0, \quad k = 1, \dots, d. \end{aligned} \quad (41)$$

where we have denoted by Λ_k the ℓ -dimensional vector of $\alpha^k = (\alpha_1^k, \dots, \alpha_\ell^k)^T$, $k = 1, \dots, d$, by K_k the $(\ell \times \ell)$ -dimensional matrix of elements $K_k(x_i^k, x_j^k) = \min(x_i^k, x_j^k)$, and by

$$\Theta(x_j^k) = (\theta(x_1^k - x_j^k), \dots, \theta(x_\ell^k - x_j^k))^T, \quad j = 1, \dots, \ell, \quad k = 1, \dots, d$$

we have denoted the $d \times \ell$ vectors of dimensionality ℓ .

Let vector $x = (x^1, \dots, x^d)$ have bounded coordinates $0 \leq x^k \leq c_k$, $k = 1, \dots, d$. Since conditional probability does not exceed 1, we need one more constraint $P(y = 1|c_1, \dots, c_d) \leq 1$. That is, we have to add the constraint

$$\sum_{k=1}^d \Lambda_k^T \mathbf{X}^k + b \leq 1, \quad (42)$$

where we have denoted $\mathbf{X}^k = (x_1^k, \dots, x_\ell^k)^T$. A function satisfying the conditions (41) and (42), is monotonic (it can be proven in the same way it was done in Section 3.2).

3.3.3 ESTIMATION OF MULTIDIMENSIONAL MONOTONIC CONDITIONAL PROBABILITY USING L_2 -NORM SVM

In order to estimate the conditional probability in indirect setting, one minimizes functional (36), subject to constraints (37), (38) for multiplicative kernel (or functional (40) subject to constraints (41), (42) for additive kernel), where V -matrix is replaced with identity matrix (I -matrix).

3.3.4 ESTIMATION OF MULTIDIMENSIONAL MONOTONIC CONDITIONAL PROBABILITY USING L_1 -NORM SVM

In order to estimate multidimensional conditional monotonic function using L_1 -norm and multiplicative INK-spline kernel (33), one minimizes the functional

$$R(\xi, \Lambda) = C \mathbf{1}_\ell^T \xi + \gamma \Lambda^T K \Lambda$$

subject to the constraints

$$-\bar{\xi} \leq Y - K \Lambda - b \mathbf{1}_\ell \leq \bar{\xi},$$

and constraints (37), (38). Here we used the notations

$$\bar{\xi} = (\xi_1, \dots, \xi_\ell)^T, \quad Y = (y_1, \dots, y_\ell)^T.$$

To estimate multidimensional conditional monotonic function using L_1 -norm and additive INK-spline kernel (39), one minimizes the functional

$$R(\xi, \Lambda) = C \mathbf{1}_\ell^T \xi + \gamma \left[\sum_{k=1}^d \Lambda_k^T K_k \Lambda_k \right]$$

subject to the constraints

$$-\bar{\xi} \leq Y - \sum_{k=1}^d K_k \Lambda_k - b \mathbf{1}_\ell \leq \bar{\xi},$$

and constraints (41), (42).

3.3.5 COMPUTATIONAL ISSUES

Quadratic optimization problem. In order to estimate a multidimensional monotonic function using multiplicative kernel, one has to solve a quadratic optimization problem of order ℓ subject to $N = \ell d$ inequality constraints.

With additive kernel, one has to estimate $d \times (\ell + 1)$ parameters under $d \times (\ell + 1)$ constraints. To decrease the computation amount:

1. One can replace V -matrix with I -matrix.
2. For additive kernel, one can estimate multidimensional conditional probability function in the *restricted set of functions* where $\alpha_i^k = \alpha_i$, for some or for all t .
3. One can consider linear structure of the solution using d one-dimensional estimates of conditional probability $P(y = 1|s^t)$ obtained by solving one-dimensional estimation problems as described in Section 3.2.1, and then approximate the multidimensional conditional probability function as

$$P(y = 1|s^1, \dots, s^d) = \sum_{t=1}^d \beta_t P(y = 1|s^t),$$

where its weights $\beta_t \geq 0$, $\sum \beta_t = 1$ are computed by solving an d -dimensional quadratic optimization problem under $d + 1$ constraints. That optimization problem is formulated as follows: minimize the functional

$$B^T P V P B - 2B^T P V Y + \gamma B^T B$$

subject to the constraints

$$B \geq 0, \quad B^T \mathbf{1}_\ell = 1,$$

where we have denoted by B vector of coefficients $B = (\beta_1, \dots, \beta_d)^T$, by P the $(d \times \ell)$ -dimensional matrix $P = p(x_i^t)$, $t = 1, \dots, d$, $i = 1, \dots, \ell$.

Estimation of both the SVMs and the conditional probability using the same data set. In the examples considered in this section, we construct synergy rules for SVMs where we use the same training set both for constructing SVM rules $s_k = f_t(x)$, $t = 1, \dots, d$ and for estimating the conditional probability $P(y = 1|s_1, \dots, s_d)$.

Suppose that our rules were constructed using different SVM kernels $K_t(x, y)$ and the same training set

$$(x_1, y_1), \dots, (x_\ell, y_\ell) \quad (43)$$

and let

$$s_1^t, \dots, s_\ell^t, \quad t = 1, \dots, d$$

be the scores $s^t = f_t(x)$ obtained using vectors x from (43).

Note that these scores are statistically different from the scores obtained using ℓ elements of test set (support vectors s^* are biased: in the separable case, all $|s^*| = 1$). Therefore, it is reasonable to use scores obtained in the procedure of k -fold cross-validation for estimating parameters of SVM algorithm.

Also, note that while individual components of the same d -dimensional vector $S^d = (s_1^d, \dots, s_d^d)$ are interdependent, the vectors S^d themselves are not (they are i.i.d.), so the general theory developed in the previous sections is applicable here for computing conditional probabilities.

4. Synergy of Several SVMs

In this section, we consider several examples of synergy of d SVM rules obtained under different circumstances:

1. Synergy of d rules obtained using the same training data but different kernels.
2. Synergy of d rules obtained using different training data but the same kernel.
3. Synergy of d classes classification problem using d *one versus the rest* rules.

In all these examples, the synergy of the rules is based on estimating the corresponding monotonic conditional probability function from RKHS associated with additive kernel, as described in Section 3.3.2.

4.1 Synergy of SVM Rules with Different Kernels

In this section, we show that the accuracy of classification using synergy of SVM rules that use different kernels can be much higher than the accuracy of a rule based on any kernel.² The effect of synergy, which is estimated by the number of additional training examples in training data required to achieve comparable to synergy level of accuracy, can be significant.

We selected the following 9 calibration data sets from UCI Machine Learning Repository (Lichman (2013)): Covertypе, Adult, Tic-tac-toe, Diabetes, Australian, Spambase, MONK's-1, MONK's-2, and Bank marketing. Our selection of these specific data sets was driven by the desire to ensure statistical reliability of targeted estimates, which translated into availability of relatively large test data set (containing at least 150 samples). Specific breakdowns for the corresponding training and test sets are listed in Table 1.

For each of these 9 data sets, we constructed 10 random realizations of training and test data sets; for each of these 10 realizations, we trained three SVMs with different kernels: with RBF kernel, with INK-Spline kernel, and with linear kernel. The averaged test errors of the constructed SVMs are listed in Table 1.

Constructed SVMs provide binary classifications g and scores s . Additional performance improvements are possible by intelligent leveraging of the results of these classifications.

We compared our approach with the baseline method of voting on classification results of all three classifications obtained from three different kernels (since we had odd number of kernels, we did not need any tie-breaking in that vote). The first column of Table 2 shows the averaged test errors of that voting approach.

The second column of Table 2 shows the averaged test errors of our synergy approach. Specifically, the data in the second column are based on constructing a 3-dimensional mono-

² The idea of using several SVMs as ensemble SVM (such as (Wang et al. (2009)) and Stork et al. (2013)) was used in the past for providing improved classification performance; however, these approaches did not leverage the main monotonicity property of SVM.

Data set	Training	Test	Features
Covertypе	300	3000	54
Adult	300	26147	123
Tic-tac-toe	300	658	27
Diabetes	576	192	8
Australian	517	173	14
Spambase	300	4301	57
MONK's-1	124	432	6
MONK's-2	169	432	6
Bank	300	4221	16

Table 1: Calibration data sets from UCI Machine Learning repository.

Data set	Voting	Synergy	Gain
Covertypе	27.83%	28.96%	-4.05%
Adult	20.07%	19.08%	4.93%
Tic-tac-toe	1.95%	1.75%	10.16%
Diabetes	24.53%	23.39%	4.67%
Australian	12.02%	12.54%	-4.33%
Spambase	8.96%	8.44%	5.80%
MONK's-1	22.80%	20.16%	11.57%
MONK's-2	19.31%	16.23%	15.95%
Bank	12.79%	11.73%	8.29%

Table 2: Synergy of SVMs with RBF, INK-spline, and linear kernels.

tonic conditional probability function from RKHS associated with additive kernel, as described in Section 3.3.2, on triples of SVM scores s . In this column, we assigned the classification labels g based on the sign of the difference between 3-dimensional conditional probability and the threshold value $1/2$.

The last column of Table 2 contains relative performance gain (i.e., relative decrease of error rate) delivered by the proposed synergy approach over the benchmark voting algorithm.

The results demonstrate the consistent performance advantage of synergy approach over its empirical alternative in most of the cases (for 7 data sets out of 9); for some data sets this advantage is relatively small, but for others it is substantial (in relative terms).

This substantial performance improvement of synergy can be also viewed as a viable alternative to brute force approaches relying on accumulation of (big) data. Indeed, for the already considered Adult data set, we compared results of our synergy approach on a training data set consisting of 300 samples to an alternative approach relying on training SVM algorithms on larger training data sets. Specifically, we trained SVMs with RBF kernel and INK-Spline kernel on Adult data sets containing 1,000 and 3,000 samples. The results, shown in Table 3, suggest that synergy of two rules, even on training data set of

Training size	300	1000	3000
RBF	20.95%	19.21%	18.49%
INK-Spline	19.77%	18.72%	18.38%
Synergy	17.92%	-	-

Table 3: Synergy versus training size increase.

limited size, can be better than straightforward SVMs on training data sets of much larger sizes (in this example, equivalent to the increase of training sample by more than a factor of 10).

4.2 Synergy of SVM Rules Obtained on Different Training Data

Suppose we are dealing with “big data” situation, where the number L of elements in the training data set

$$(x_1, y_1), \dots, (x_L, y_L), \quad (44)$$

is large. Consider the SVM method that uses a universal kernel³. Generally speaking, with the increase of size ℓ of training data, the expected error rate of the obtained SVM rule monotonically converges to the Bayesian rule (here the expectation is taken both over the rules obtained from different training data of the same size ℓ and over test data). The typical *learning curve* shows the dependence of that expected error rate on the size ℓ of training data as a hyperbola-looking curve consisting of two parts: the beginning of the curve, where the error rate falls steeply with the increase of ℓ , and the tail of the curve, where the error rate slowly converges to the Bayesian solution. Suppose that the transition from the “steeply falling” part of the curve to the “slowly decreasing” part of the curve (sometimes referred to as the “knee” of the curve) occurs for some ℓ^* . Assuming that large number L in (44) is greater than ℓ^* , we partition the training data (44) into J subsets containing ℓ elements each (here $L = J\ell$ and $\ell > \ell^*$ as well):

$$(x_{(t-1)\ell+1}, y_{(t-1)\ell+1}), \dots, (x_{t\ell}, y_{t\ell}), \quad t = 1, \dots, J \quad (45)$$

On each of these J disjoint training subsets we construct its own SVM rule (independent of other rules)

$$y = \theta(f_t(x, \alpha_\ell)), \quad t = 1, \dots, J.$$

For each of these SVM rules, we construct (as described in Section 3.3.2) its own one-dimensional monotonic conditional probability function $P_t(y = 1|s^t)$, $t = 1, \dots, J$.

Then, using these J one-dimensional monotonic conditional probability functions, we construct the J -dimensional $(s = (s^1, \dots, s^J))$ conditional probability function as follows:

$$P_{syn}(y = 1|s) = \frac{1}{J} \sum_{t=1}^J P_t(y = 1|s^t). \quad (46)$$

3. A universal kernel (for example, RBF) can approximate well any bounded continuous function.

Training size	300	300	300	900	1000	3000
RBF SVM	20.77%	19.06%	21.40%	20.01%	19.21%	18.49%
Voting on 3 subsets	N/A	N/A	N/A	19.44%	-	-
Synergy on 3 subsets	N/A	N/A	N/A	18.52%	-	-

Table 4: Synergy versus training size increase.

The Synergy decision rule in this case has the form

$$y = \theta \left(P_{syn}(y = 1|s) - \frac{1}{2} \right).$$

Note that (46) forms an unbiased estimate of the values of learning curve describing conditional probability for training data of (different) size ℓ . Since the training data (45) for different t are independent, the averaging of J conditional probability values decreases the variance of resulting conditional probability by a factor of J . In this approach, by choosing an appropriate value of ℓ , one can optimally solve the bias-variance dilemma.

To illustrate this approach, we again used Adult data set. Specifically, we trained SVMs with RBF kernel on Adult data sets containing 900, 1,000 and 3,000 samples. For the first of these samples (containing 900 elements), we also executed the following procedure: we split it into three subsets containing 300 elements each, trained RBF SVM on each of them, and then constructed two combined decision rules: (1) voting on the labels of three auxiliary SVMs, and (2) synergy of three SVMs as described in this section. The results, shown in Table 4, suggest that Synergy of rules on disjoint data sets can be better than straightforward SVMs on training data sets of much larger sizes (in this example, equivalent to the increase of training sample by a factor of 3).

Comparison of Table 3 and Table 4 suggests that synergy of SVMs with different SVM kernels obtained on the same data set may be more beneficial (equivalent to ten-fold increase of training sample size) than the synergy of SVMs with the same kernel obtained on different subsets of of that data set (equivalent to three-fold increase of training sample size).

Thus it is reasonable to assume that, for big data set (44), Synergy of SVM rules obtained on different training data and Synergy of SVM rules with different kernels (described in previous Section 4.1) can be unified to create an even more accurate synergy rule. This unification can be implemented in the following manner.

Consider d kernels $K_r(x, x')$, $k = 1, \dots, d$. For each of these kernels, using the method described in Section 3.3.2, we construct the corresponding condition probability function

$$P_{syn}(y = 1|s(r)) = \frac{1}{J} \sum_{t=1}^J P_t(y = 1|s^t(r)),$$

where we have denoted by $P_t(y = 1|s^t(r))$ the conditional probability function estimated for the rule with kernel $K_r(x, x')$ and for the t th subset of training data (44) with the fixed t . Let introduce the vector $p = (p^1, \dots, p^d)$ where

$$p^r = P_{syn}(y = 1|s(r)), \quad r = 1, \dots, d.$$

Using these vectors, we estimate the d -dimensional conditional probability function $P_{sym}(y = 1|p) = P_{sym}(y = 1|p^1, \dots, p^d)$.

The resulting double reinforced Synergy rule has the form

$$y = \theta \left(P_{sym}(y = 1|p) - \frac{1}{2} \right).$$

4.3 Multi-Class Classification Rules

Constructing decision rules for multi-class classification is an important problem in pattern recognition. In contrast to methods for constructing two-class classification rules, which have solid statistical justifications, existing methods for constructing $d > 2$ class classification rules are based on heuristics.

One of the most popular heuristics, *one versus rest* (OVR), suggests first to solve the following d two-class classification problems: in problem number k (where $k = 1, \dots, d$), the examples of class k are considered as examples of the first class and examples of the all other classes $1, \dots, (k - 1), (k + 1), \dots, d$ are considered as the second class. Using OVR approach, one constructs d different two-class classification rules

$$y = \theta(f_k(x)) \quad k = 1, \dots, d.$$

The new object x_* is assigned to the class k , where k th rule provides the maximum score for x_* :

$$k = \operatorname{argmax}\{s_1^1, \dots, s_1^d\}, \quad \text{where } s_i^k = f_i(x_*).$$

This method of d -class classification is not based on a clear statistical foundation⁴.

Here we implement the following multi-class classification procedure. For every k (where $k = 1, \dots, d$), we solve the corresponding OVR SVM problem, for which all the elements with the original label k are marked with $y = 1$, while all the other elements are marked with $y = 0$. Upon solving all these d problems, we can, for any given vector x and any class k , compute its score $s_k(x)$ provided by the k th SVM rule.

After that, for every k (where $k = 1, \dots, d$) we use the obtained scores for estimating conditional probability of the class k based on the scores $(\bar{s}^1, \dots, \bar{s}^d)$ where

$$\bar{s}^m = \begin{cases} s_m & \text{if } m = k \\ -s_m & \text{if } m \neq k \end{cases}$$

This transformation of scores is used to maintain the monotonicity of the overall conditional probability function. To estimate the function

$$P(k) = P(k|\bar{s}^1, \dots, \bar{s}^d),$$

as in Section 3.3.2, we use the representation

$$P(k|\bar{s}^1, \dots, \bar{s}^d) = \sum_{i=1}^d \left[\alpha_i \min\{\bar{s}_i^k, s_i^k\} + \beta_i \sum_{i \neq k} \min\{\bar{s}_i^k, s_i^k\} \right], \quad k = 1, \dots, d.$$

⁴ Another common heuristics called *one versus one* (OVO): it suggests to solve C_2^d two-class classification problems separating all possible pairs of classes. To classify a new object x^* , one uses a voting scheme based on the obtained C_2^d rules.

Data set	Classes	Features	Training	Test	OVR	Synergy	Gain
Vehicle	4	18	709	236	17.45%	14.15%	18.91%
Waveform	3	40	200	4800	20.10%	18.31%	8.90%
Cardiotocography	3	21	300	1826	15.83%	12.05%	23.87%

Table 5: Synergy for multi-class classification.

Finally, we replace the heuristic procedure of choosing the class k based on maximization of underlying scores with the following procedure that is based on the framework described above: this procedure uses estimated d conditional probabilities $P(k|s_1, \dots, s_d)$ (probability of class $k = 1, \dots, d$ given all d scores) and chooses the class t corresponding to the maximum value of the conditional probability:

$$t = \operatorname{argmax}\{P(1|\bar{s}_1^1, \dots, \bar{s}_1^d), \dots, P(d|\bar{s}_1^1, \dots, \bar{s}_1^d)\}.$$

We compared our synergy approach with the standard OVR approach for the data sets Vehicle, Waveform, and Cardiotocography from UCI Machine Learning Repository (Lichman (2013)). Training and test sets were selected randomly from these data sets; the number of elements in each are shown in Table 5; the table also shows the error rates achieved by OVR and synergy algorithm, along with relative performance gain obtained with our approach. The results confirm the viability of our framework.

5. Synergy of Learning from Several Intelligent Teachers

In (Vapnik and Izmailov (2015d)), (Vapnik and Izmailov (2015b)), we introduced the concept of *knowledge transfer* from Intelligent Teacher to student. Knowledge transfer is possible in the framework of Learning Using Privileged Information (LUPI) paradigm introduced in (Vapnik (2006)) and (Vapnik and Vashist (2009)). According to this paradigm, iid training examples are generated by some unknown generator $P(x)$, $x \in X$ and Intelligent Teacher who supplies vectors x with information (x^*, y) according to some (unknown) *Intelligence generator* $P(x^*, y|x)$, $x^* \in X^*$, $y \in \{-1, 1\}$, forming training triples

$$(x_1, x_1^*, y_1), \dots, (x_\ell, x_\ell^*, y_\ell). \quad (47)$$

Vector x_i^* corresponding to vector x_i is called *privileged information*, and generator $P(x^*, y|x)$ is called generator of intelligent (due to x^*) information. Privileged information is available only for training examples and is *not available* for test examples. In contrast to LUPI, classical learning paradigm considers a primitive teacher that just generates classification y for any x according to $P(y|x)$ (with no additional explanation x^*), forming training pairs

$$(x_1, y_1), \dots, (x_\ell, y_\ell).$$

Knowledge transfer mechanism. Consider the second⁵ mechanism in LUPI paradigm the *knowledge transfer mechanism* to construct a better decision rule. Given triplets (47), we can consider two pattern recognition problems:

1. *Pattern recognition problem defined in space X*: Using data, $(x_1, y_1), \dots, (x_\ell, y_\ell)$, find in the set of functions $f(x, \alpha), \alpha \in \Lambda$ the rule $y = \text{sgn}\{f(x)\}$ that minimizes the probability of test errors (in space X).
2. *Pattern recognition problem defined in space X**: Using data, $(x_1^*, y_1), \dots, (x_\ell^*, y_\ell)$, find in the set of functions $f^*(x^*, \alpha^*), \alpha^* \in \Lambda^*$ the rule $y = \text{sgn}\{f^*(x^*)\}$ that minimizes the probability of test errors (in space X*).

Suppose that, in space X*, one can find a rule $y = \text{sgn}\{f_0^*(x^*)\}$ that is better (more accurate) than the corresponding rule $y = \text{sgn}\{f_0(x)\}$ in space⁶ X.

The question arises: *Can the knowledge about a good rule*

$$f_0^*(x^*) = \sum_{i=1}^{\ell} y_i \alpha_i^* K^*(x_i^*, x^*) + b^* \quad (48)$$

in space X* help to find a good rule

$$f_\ell(x) = \sum_{i=1}^{\ell} y_i \alpha_i K(x_i, x) + b \quad (49)$$

in space X?

Consider the following example. Suppose that our goal is to classify images x_i of biopsy in pixel space X into two categories: cancer and non-cancer.

Suppose that, along with images x_i in pixel space X, we are given description of the images $x_i^* \in X^*$ (privileged information), reflecting the existing model of developing cancer:

- *Aggressive proliferation of A-cells into B-cells.*
- *Absence of any dynamic in standard picture of sells distribution.*

Since pixel space X is universal (it can be used for many problems, for example, in pixel space, one can distinguish male faces from female ones), and space of descriptions X* reflects just the model of cancer development⁷, the VC dimension of the corresponding set of functions in X space has to be larger than the VC dimension of the corresponding set of functions in X*.

Therefore the rule constructed from ℓ examples in space X* will be more accurate than the rule constructed from ℓ examples in space X. That is why transferring the rule from space X* into space X can be helpful.

5. The first mechanism is called *similarity control* described in (Vapnik and Izmailov (2015d)), (Vapnik and Izmailov (2015b)). The second mechanism of knowledge transfer, described there and further in this paper, is related to SVM technology. However, the idea of knowledge transfer is general and can be implemented for other learning algorithms.
6. This is always the case if space X is a subset of X*.
7. In this example, generator $P(x^*, y|x)$ is intelligent since for any *picture* of the event x it describes the *essence* of the event. Using descriptions of the essence of an event makes classification of the event a relatively easy problem.

Knowledge representation in space X*. To transfer knowledge from space X* into space X, we use three elements of knowledge representation developed in 1950's in Artificial Intelligence (see Brachman and Levesque, 2004):

1. Fundamental elements of the knowledge in X*.
2. Main frames (fragments) of the knowledge in X*.
3. Structure of knowledge: combination of the frames in X*.

For LUPI using SVM⁸:

1. The *fundamental elements* are defined by k support vectors of rule (48) in X*.
2. The *frames* in X* are defined by the functions $K^*(x_s^*, x^*)$, $s = 1, \dots, k$.
3. The structure of the knowledge (48) is linear in the frames.

Algorithm for knowledge transfer. In order to transfer knowledge from space X* to space X, one has to make two transformations in the training triplets (47):

1. To transform n -dimensional vectors of $x_i = (x_i^1, \dots, x_i^n)^T$ into k -dimensional vectors $\mathcal{F}x_i = (\phi_1(x_i), \dots, \phi_k(x_i))^T$;
2. Use the target values $f_\ell^*(x_i^*)$ obtained for x_i^* in rule (48) instead of the values y_i , given for x_i in triplet (47).

(1) In order to transform vector x , one constructs k -dimensional space as follows: for any frame $K^*(x_s^*, x_s^*)$, $s = 1, \dots, k$ in space X*, one constructs its image (function) $\phi_s(x)$ in space X that is defined by the relationship

$$\phi_s(x) = \int K(x_s^*, x^*) P(x^*|x) dx^*, \quad s = 1, \dots, k.$$

This requires to solve the following regression estimation problem: given data

$$(x_1, z_1^s), \dots, (x_\ell, z_\ell^s), \quad \text{where } z_i^s = K(x_s^*, x_i^*),$$

find k regression functions $\phi_s(x)$, $s = 1, \dots, k$, forming the space $\mathcal{F}(x) = (\phi_1(x), \dots, \phi_k(x))^T$.
(2) Replace target value y_i in triplets (47) with scores $f_\ell^*(x_i^*)$ given (48).

Therefore the knowledge transfer algorithm transforms the training triplet⁹

$$((\mathcal{F}x_1, x_1^*, f_\ell^*(x_1^*)), \dots, (\mathcal{F}x_\ell, x_\ell^*, f_\ell^*(x_\ell^*))). \quad (50)$$

It uses triplets (50) instead of triplets (47).

Synergy of several Intelligent Teachers. Suppose now that Student tries to learn how to solve the same problem from several (say two) Intelligent Teachers. For simplicity, let both Teachers use the same training data (x_i, y_i) , $i = 1, \dots, \ell$ but different privileged information (different explanations)

$$(x_1, x_1^*, y_1), \dots, (x_\ell, x_\ell^*, y_\ell)$$

and

$$(x_1, x_1^{**}, y_1), \dots, (x_\ell, x_\ell^{**}, y_\ell).$$

Constructing, using these triplets, two different rules and corresponding synergy rule, one obtains the synergy effect of two Intelligent Teachers.

8. Different concepts of fundamental elements, frames, and structure of knowledge can be applied for different algorithms.
9. In the simplified version, pairs $(\mathcal{F}x_i, \delta_i^*)$, $i = 1, \dots, \ell$.

6. Conclusion

In this paper, we showed that:

1. Scores $s = (s^1, \dots, s^d)$ of several monotonic classifiers (for example, SVMs) that solve the same pattern recognition problem can be transformed into multi-dimensional monotonic conditional probability functions $P(y|s)$ (probability of class y given scores s).
2. There exists an effective algorithm for such transformation.
3. Classification rules obtained on the basis of constructed conditional probability functions significantly improve performance, especially in multi-class classification cases.

Acknowledgments

This material is based upon work partially supported by AFRL and DARPA under contract FA8750-14-C-0008, and by AFRL under contract FA9550-15-1-0502. Any opinions, findings and / or conclusions in this material are those of the authors and do not necessarily reflect the views of AFRL and DARPA.

The authors thank the reviewers for their outstanding diligence, which helped to correct multiple errors and typos in the paper.

References

- R. Andersen. *Modern methods for robust regression*. Quantitative Applications in the Social Sciences. SAGE, 2008.
- M. Best and N. Chakravarti. Active set algorithms for isotonic regression: a unifying framework. *Mathematical Programming*, 47(1):425–439.
- R. Braachman and H. Levesque. *Knowledge Representation and Reasoning*. Morgan Kaufman Publishers, San Francisco, CA, 2004.
- T. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems, LBCS-1857*, pages 1–15. Springer, 2000.
- R. Izmailov, V. Vapnik, and A. Yashit. Multidimensional splines with infinite number of knots as SVM kernels. In *International Joint Conference on Neural Networks*, pages 1096–1102, 2013.
- P. Jäckel. *Monte Carlo methods in finance*. Wiley, 2004.
- G. Kinneldorf and G. Wähba. A correspondence between bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, pages 495–502, 1970.
- G. Lecué. Optimal rates of aggregation in classification under low noise assumption. *Bernoulli*, 13(4):1000–1022, 11 2007.
- M. Lichman. UCI machine learning repository, 2013. URL <http://archive.ics.uci.edu/ml>.
- H. Lin, C. Lin, and R. Weng. A note on Platt’s probabilistic outputs for support vector machines. *Machine Learning*, 68(3):267–276, 2007.
- P. Mair, K. Hornik, and J. de Leeuw. Isotone optimization in R: pool-adjacent-violators algorithm (PAVA) and active set methods. *Journal of Statistical Software*, 32(5):1–24, October 2009.
- P. Massart. The tight constant in the Dvoretzky-Kiefer-Wolfowitz inequality. *The Annals of Probability*, 18(3):1269–1283, 1990.
- M. Meyer. Semi-parametric additive constrained regression. *Journal of Nonparametric Statistics*, 25(3):715–730, 2013.
- J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Machine Classifiers*, pages 61–74. MIT Press, 1999.
- J. Stork, R. Rammes, P. Koch, and W. Könen. SVM ensembles are better when different kernel types are combined. In *ECD4, Studies in Classification, Data Analysis, and Knowledge Organization*, pages 191–201. Springer, 2013.
- O. Sysoev, O. Burdakov, and A. Grimvall. A segmentation-based algorithm for large-scale partially ordered monotonic regression. *Computational Statistics & Data Analysis*, 55(8): 2463–2476, 2011.
- A. Tikhonov and V. Arsenin. *Solutions of Ill-Posed Problems*. W.H. Winston, 1977.
- A. Tsybakov. *Optimal Rates of Aggregation*, pages 303–313. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.
- V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.
- V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.
- V. Vapnik. *Estimation of Dependencies Based on Empirical Data*. Springer-Verlag, 2nd edition, 2006.
- V. Vapnik and R. Izmailov. V-matrix method of solving statistical inference problems. *Journal of Machine Learning Research*, 16:1683–1730, 2015a.
- V. Vapnik and R. Izmailov. Learning using privileged information: Similarity control and knowledge transfer. *Journal of Machine Learning Research*, 16:2023–2049, 2015b.
- V. Vapnik and R. Izmailov. Statistical inference problems and their rigorous solutions. In A. Gannemman, V. Vovk, and H. Papadopoulos, editors, *Statistical Learning and Data Sciences*, volume 9047 of *Lecture Notes in Computer Science*, pages 33–71. Springer International Publishing, 2015c.

- V. Vapnik and R. Izmailov. Learning with intelligent teacher: Similarity control and knowledge transfer. In A. Gammerman, V. Vovk, and H. Papadopoulos, editors, *Statistical Learning and Data Sciences*, volume 9047 of *Lecture Notes in Computer Science*, pages 3–32. Springer International Publishing, 2015d.
- V. Vapnik and A. Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5-6):544–557, 2009.
- V. Vapnik, I. Braga, and R. Izmailov. Constructive setting for problems of density ratio estimation. *Statistical Analysis and Data Mining*, 8(3):137–146, 2015.
- S. Wang, A. Mathew, Y. Chen, L. Xi, L. Ma, and J. Lee. Empirical analysis of support vector machine ensemble classifiers. *Expert Systems with Applications*, 36(3 Pt2):6466–6476, 2009.
- C. Zhang and Y. Ma. *Ensemble Machine Learning: Methods and Applications*. Springer New York, 2012.

Pymanopt: A Python Toolbox for Optimization on Manifolds using Automatic Differentiation

James Townsend

University College London, London, UK

Niklas Koep

RWTH Aachen University, Germany

Sebastian Weichwald

Max Planck Institute for Intelligent Systems, Tübingen, Germany

JAMES.TOWNSEND.14@UCL.AC.UK

NIKLAS.KOEP@RWTH-AACHEN.DE

SWEICHWALD@TUE.MPG.DE

Editor: Antti Honkela

Abstract

Optimization on manifolds is a class of methods for optimization of an objective function, subject to constraints which are smooth, in the sense that the set of points which satisfy the constraints admits the structure of a differentiable manifold. While many optimization problems are of the described form, technicalities of differential geometry and the laborious calculation of derivatives pose a significant barrier for experimenting with these methods.

We introduce PYMANOPT (available at [pymanopt.github.io](https://github.com/pymanopt)), a toolbox for optimization on manifolds, implemented in Python, that—similarly to the Manopt¹ Matlab toolbox—implements several manifold geometries and optimization algorithms. Moreover, we lower the barriers to users further by using automated differentiation² for calculating derivative information, saving users time and saving them from potential calculation and implementation errors.

Keywords: Riemannian optimization, non-convex optimization, manifold optimization, projection matrices, symmetric matrices, rotation matrices, positive definite matrices

1. Introduction

Optimization on manifolds, or Riemannian optimization, is a method for solving problems of the form

$$\min_{x \in \mathcal{M}} f(x)$$

where $f: \mathcal{M} \rightarrow \mathbb{R}$ is a (cost) function and the search space \mathcal{M} is smooth, in the sense that it admits the structure of a differentiable manifold. Although the definition of differentiable manifold is technical and abstract, many familiar sets satisfy this definition and are therefore compatible with the methods of optimization on manifolds. Examples include the *sphere* (the set of points with unit Euclidean norm) in \mathbb{R}^n , the set of *positive definite matrices*, the set of *orthogonal matrices* as well as the set of p -dimensional subspaces of \mathbb{R}^n with $p < n$, also known as the *Grassmann* manifold.

To perform optimization, the function f needs to be defined for points on the manifold \mathcal{M} . Elements of \mathcal{M} are often represented by elements of \mathbb{R}^n or $\mathbb{R}^{m \times n}$, and f is often well defined on some or all of this “ambient” Euclidean space. If f is also differentiable, it makes sense for an optimization algorithm to use the derivatives of f and adapt them to the manifold setting in order to iteratively refine solutions based on curvature information. This is one of the key aspects of Manopt (Boumal et al., 2014), which allows the user to pass a function’s gradient and Hessian to state of the art

solvers which exploit this information to optimize over the manifold \mathcal{M} . However, working out and implementing gradients and higher order derivatives is a laborious and error prone task, particularly when the objective function acts on matrices or higher rank tensors. Manopt’s state of the art Riemannian Trust Regions solver, described in Absil et al. (2007), requires second order directional derivatives (or a numerical approximation thereof), which are particularly challenging to work out for the average user, and more error prone and tedious even for an experienced mathematician.

It is these difficulties which we seek to address with this toolbox. PYMANOPT supports a variety of modern Python libraries for automated differentiation of cost functions acting on vectors, matrices or higher rank tensors. Combining optimization on manifolds and automated differentiation enables a convenient workflow for rapid prototyping that was previously unavailable to practitioners. All that is required of the user is to instantiate a manifold, define a cost function, and choose one of PYMANOPT’s solvers. This means that the Riemannian Trust Regions solver in PYMANOPT is just as easy to use as one of the derivative-free or first order methods.

2. The Potential of Optimization on Manifolds and Pymanopt Use Cases

Much of the theory of how to adapt Euclidean optimization algorithms to (matrix) manifolds can be found in Smith (1994); Edelman et al. (1998); Absil et al. (2008). The approach of optimization on manifolds is superior to performing free (Euclidean) optimization and projecting the parameters back onto the search space after each iteration (as in the projected gradient descent method), and has been shown to outperform standard algorithms for a number of problems.

Hosseini and Sra (2015) demonstrate this advantage for a well-known problem in machine learning, namely inferring the maximum likelihood parameters of a mixture of Gaussian (MoG) model. Their alternative to the traditional expectation maximization (EM) algorithm uses optimization over a product manifold of positive definite (covariance) matrices. Rather than optimizing the likelihood function directly, they optimize a reparameterized version which shares the same local optima. The proposed method, which is on par with EM and shows less variability in running times, is a striking example why we think a toolbox like PYMANOPT, which allows the user to readily experiment with and solve problems involving optimization on manifolds, can accelerate and pave the way for improved machine learning algorithms.³

Further successful applications of optimization on manifolds include matrix completion tasks (Vandereycken, 2013; Boumal and Absil, 2015), robust PCA (Podosinnikova et al., 2014), dimension reduction for independent component analysis (ICA) (Theis et al., 2009), kernel ICA (Shen et al., 2007) and similarity learning (Shalit et al., 2012).

Many more applications to machine learning and other fields exist. While a full survey on the usefulness of these methods is well beyond the scope of this manuscript, we highlight that at the time of writing, a search for the term “manifold optimization” on the IEEE Xplore Digital Library lists 1065 results; the Manopt toolbox itself is referenced in 90 papers indexed by Google Scholar.

3. Implementation

Our toolbox is written in Python and uses NumPy and SciPy for computation and linear algebra operations. Currently PYMANOPT is compatible with cost functions defined using Autograd (Maclaurin et al., 2015), Theano (AI-Rfou et al., 2016) or TensorFlow (Abadi et al., 2015). PYMANOPT itself and all the required software is open source, with no dependence on proprietary software.

To calculate derivatives, Theano uses symbolic differentiation, combined with rule-based optimizations, while both Autograd and TensorFlow use reverse-mode automatic differentiation. For a discussion of the distinctions between the two approaches and an overview of automatic differentiation in the context of machine learning, we refer the reader to Baydin et al. (2015).

3. A quick example implementation for inferring MoG parameters is available at [pymanopt.github.io/MeG.html](https://github.com/pymanopt/pymanopt/blob/master/MeG.html).

Much of the structure of PYMANOPT is based on that of the Manopt Matlab toolbox. For this early release, we have implemented all of the solvers and a number of the manifolds found in Manopt, and plan to implement more, based on the needs of users. The codebase is structured in a modular way and thoroughly commented to make extension to further solvers, manifolds, or backends for automated differentiation as straightforward as possible. Both a user and developer documentation are available. The GitHub repository at github.com/pymanopt/pymanopt offers a convenient way to ask for help or request features by raising an issue, and contains guidelines for those wishing to contribute to the project.

4. Usage: A Simple Instructive Example

All automated differentiation in PYMANOPT is performed behind the scenes so that the amount of setup code required by the user is minimal. Usually only the following steps are required:

- (a) Instantiation of a manifold \mathcal{M}
- (b) Definition of a cost function $f: \mathcal{M} \rightarrow \mathbb{R}$
- (c) Instantiation of a PYMANOPT solver

We briefly demonstrate the ease of use with a simple example. Consider the problem of finding an $n \times n$ positive semi-definite (PSD) matrix S of rank $k < n$ that best approximates a given $n \times n$ (symmetric) matrix A , where closeness between A and its low-rank PSD approximation S is measured by the following loss function

$$L_\delta(S, A) \triangleq \sum_{i=1}^n \sum_{j=1}^n H_\delta(s_{i,j} - a_{i,j})$$

for some $\delta > 0$ and $H_\delta(x) \triangleq \sqrt{x^2 + \delta^2} - \delta$ the pseudo-Huber loss function. This loss function is robust against outliers as $H_\delta(x)$ approximates $|x| - \delta$ for large values of x while being approximately quadratic for small values of x (Huber, 1964).

This can be formulated as an optimization problem on the manifold of PSD matrices:

$$\min_{S \in \text{PSD}_k^n} L_\delta(S, A)$$

where $\text{PSD}_k^n \triangleq \{M \in \mathbb{R}^{n \times n} : M \succeq 0, \text{rank}(M) = k\}$. This task is easily solved using PYMANOPT:

```
from pymanopt.manifolds import PSDFixedRank
import autograd.numpy as np
from pymanopt import Problem
from pymanopt.solvers import TrustRegions

# Let A be a (n x n) matrix to be approximated

# (a) Instantiation of a manifold
# points on the manifold are parameterized as YY^T
# where Y is a matrix of size n x k
manifold = PSDFixedRank(A.shape[0], k)
# (b) Definition of a cost function (here using autograd.numpy)
def cost(Y):
    S = np.dot(Y, Y.T)
    delta = .5
    return np.sum(np.sqrt((S - A)**2 + delta**2)) - delta
```

```
# define the Pymanopt problem
problem = Problem(manifold=manifold, cost=cost)
# (c) Instantiation of a Pymanopt solver
solver = TrustRegions()

# Let Pymanopt do the rest
Y = solver.solve(problem)
S = np.dot(Y, Y.T)
```

The examples folder within the PYMANOPT toolbox holds further instructive examples, such as performing inference in mixture of Gaussian models using optimization on manifolds instead of the expectation maximization algorithm. Also see the examples section on [pymanopt.github.io](https://github.com/pymanopt/pymanopt).

5. Conclusion

PYMANOPT enables the user to experiment with different state of the art solvers for optimization problems on manifolds, like the Riemannian Trust Regions solver, without any extra effort. Experimenting with different cost functions, for example by changing the pseudo-Huber loss $L_\delta(S, A)$ in the code above to the Frobenius norm $\|S - A\|_F$, a p -norm $\|S - A\|_p$ or some more complex function, requires just a small change in the definition of the cost function. For problems of greater complexity, PYMANOPT offers a significant advantage over toolboxes that require manual differentiation by enabling users to run a series of related experiments without returning to pen and paper each time to work out derivatives. Gradients and Hessians only need to be derived if they are required for other analysis of a problem. We believe that these advantages, coupled with the potential for extending PYMANOPT to large-scale applications using TensorFlow, could lead to significant progress in applications of optimization on manifolds.

Acknowledgments

We would like to thank the developers of the Manopt Matlab toolbox, in particular Nicolas Boumal and Pierre-Antoine Absil, for developing Manopt, and for the generous help and advice they have given. We would also like to thank Helko Strathmann for his thoughtful advice as well as the anonymous reviewers for their constructive feedback and idea for a more suitable application example.

References

- M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudrur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, P. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. URL <http://tensorflow.org>.
- P.-A. Absil, C.G. Baker, and K.A. Gallivan. Trust-Region Methods on Riemannian Manifolds. *Foundations of Computational Mathematics*, 7(3):303–330, 2007.
- P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008. ISBN 978-0-691-13298-3.
- R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, F. Bastien, J. Bayer, A. Belikov, A. Belopolsky, Y. Bengio, A. Bergeron, J. Bergstra, V. Bisson, J. Blecher Snyder, N. Bouchard, N. Boulanger-Levandowski, X. Bouthillier, A. de Brébisson, O. Breuleux, P.-L. Carrier, K. Cho, J. Chorowski, P. Christiano, T. Cojmann, M.-A. Côté, M. Côté, A. Courville, Y.N. Dauphin, O. Delalleau, J. Demouth, G. Desjardins, S. Dieleman, L. Dinh, M. Ducoffe, V. Dumoulin, S. Ebrahimi Kahou, D. Erhan, Z. Fan, O. Firat, M. Gernain, X. Glorot, I. Goodfellow, M. Graham, C. Gulcehre, P. Hamel, J. Haroutchiet, J.-P. Heng, B. Hidasi, S. Honari, A. Jain, S. Jean, K. Jia, M. Korobov, V. Kulkarni, A. Lamb, P. Lamblin, E. Larsen, C. Laurent, S. Lee, S. Lefrançois, S. Lemaire, N. Léonard, Z. Lin, J. A. Livezey, C. Lorenz, J. Lowin, Q. Ma, P.-A. Manzagol, O. Mastroietro, R.T. McElhobon, R. Memisevic, B. van Merriënboer, V. Michalski, M. Mirza, A. Orlandi, C. Pal, R. Pascanu, M. Pezeshki, C. Raffel, D. Renshaw, M. Rocklin, A. Romero, M. Roth, P. Sadowski, J. Salvatier, F. Savard, J. Schlüter, J. Schulman, G. Schwartz, I.V. Serban, D. Sedivyuk, S. Shabanian, É. Simon, S. Spöckermann, S.R. Subramanyam, J. Szymanski, J. Tanguay, G. van Tulder, J. Turian, S. Urban, P. Vincent, F. Visin, H. de Vries, D. Warde-Farley, D.J. Webb, M. Willson, K. Xu, L. Xue, L. Yao, S. Zhang, and Y. Zhang. Theano: A Python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688*, 2016. URL <http://deeplearning.net/software/theano>.
- A.G. Baydin, B.A. Pearlmutter, A.A. Radul, and J.M. Siskind. Automatic differentiation in machine learning: a survey. *arXiv preprint arXiv:1502.05767*, 2015.
- N. Boumal and P.-A. Absil. Low-rank matrix completion via preconditioned optimization on the Grassmann manifold. *Linear Algebra and its Applications*, 475:200–239, 2015. doi: 10.1016/j.laa.2015.02.027.
- N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a Matlab Toolbox for Optimization on Manifolds. *Journal of Machine Learning Research*, 15:1455–1459, 2014. URL <http://manopt.org>.
- A. Edelman, T.A. Arias, and S.T. Smith. The Geometry of Algorithms with Orthogonality Constraints. *SIAM J. Matrix Anal. & Appl.*, 20(2):303–353, 1998.
- R. Hesseini and S. Sra. Matrix Manifold Optimization for Gaussian Mixtures. In *Advances in Neural Information Processing Systems*, pages 910–918, 2015.
- P.J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.
- D. Maclaurin, D. Duvenaud, M. Johnson, and R.P. Adams. Autograd: Reverse-mode differentiation of native Python, 2015. URL <http://github.com/HIPS/autograd>.
- A. Podosinnikova, S. Setzer, and M. Hein. Robust PCA: Optimization of the Robust Reconstruction Error over the Stiefel Manifold. In *36th German Conference on Pattern Recognition (GCPR)*, 2014.
- U. Shalit, D. Weinshall, and G. Chechik. Online Learning in the Embedded Manifold of Low-rank Matrices. *Journal of Machine Learning Research*, 13(1):429–458, 2012.
- H. Shen, S. Jegelka, and A. Gretton. Fast Kernel ICA using an Approximate Newton Method. In *International Conference on Artificial Intelligence and Statistics*, pages 476–483, 2007.
- S.T. Smith. Optimization techniques on Riemannian manifolds. *Fields institute communications*, 3(3):113–135, 1994.
- F.J. Theis, T.P. Cason, and P.-A. Absil. Soft dimension reduction for ICA by joint diagonalization on the Stiefel manifold. In *Independent Component Analysis and Signal Separation*, pages 354–361. Springer, 2009.
- B. Vandereycken. Low-Rank Matrix Completion by Riemannian Optimization. *SIAM J. Optim.*, 23(2):1214–1236, 2013.

CrossCat: A Fully Bayesian Nonparametric Method for Analyzing Heterogeneous, High Dimensional Data

Vikash Mansinghka
Patrick Shafto
Eric Jonas
Cap Petschulat
Max Gasner
Joshua B. Tenenbaum

VKM@MIT.EDU
P.SHAFTO@LOUISVILLE.EDU
JONAS@PRIORKNOWLEDGE.NET
CAP@PRIORKNOWLEDGE.NET
MAX@PRIORKNOWLEDGE.NET
JBT@MIT.EDU

Editor: David Blei

Abstract

There is a widespread need for statistical methods that can analyze high-dimensional datasets without imposing restrictive or opaque modeling assumptions. This paper describes a domain-general data analysis method called CrossCat. CrossCat infers multiple non-overlapping views of the data, each consisting of a subset of the variables, and uses a separate nonparametric mixture to model each view. CrossCat is based on approximately Bayesian inference in a hierarchical, nonparametric model for data tables. This model consists of a Dirichlet process mixture over the columns of a data table in which each mixture component is itself an independent Dirichlet process mixture over the rows; the inner mixture components are simple parametric models whose form depends on the types of data in the table. CrossCat combines strengths of mixture modeling and Bayesian network structure learning. Like mixture modeling, CrossCat can model a broad class of distributions by positing latent variables, and produces representations that can be efficiently conditioned and sampled from for prediction. Like Bayesian networks, CrossCat represents the dependencies and independencies between variables, and thus remains accurate when there are multiple statistical signals. Inference is done via a scalable Gibbs sampling scheme; this paper shows that it works well in practice. This paper also includes empirical results on heterogeneous tabular data of up to 10 million cells, such as hospital cost and quality measures, voting records, unemployment rates, gene expression measurements, and images of handwritten digits. CrossCat infers structure that is consistent with accepted findings and common-sense knowledge in multiple domains and yields predictive accuracy competitive with generative, discriminative, and model-free alternatives.

Keywords: Bayesian nonparametrics, Dirichlet processes, Markov chain Monte Carlo, multivariate analysis, structure learning, unsupervised learning, semi-supervised learning

1. Introduction

High-dimensional datasets containing data of multiple types have become commonplace. These datasets are often presented as tables, where rows correspond to data vectors, columns correspond to observable variables or features, and the whole table is treated as a random subsample from a statistical population (Hastie, Tibshirani, Friedman, and Franklin, 2005). This setting brings new opportunities as well as new statistical challenges (NRC Committee on the Analysis of Massive Data, 2013; Wasserman, 2011). In principle, the dimensionality and coverage of some of these datasets is sufficient to rigorously answer to fine-grained questions about small sub-populations.

This size and richness also enables the detection of subtle predictive relationships, including those that depend on aggregating individually weak signals from large numbers of variables. Challenges include integrating data of heterogeneous types (NRC Committee on the Analysis of Massive Data, 2013), suppressing spurious patterns (Benjamini and Hochberg, 1995; Attia, Ioannidis, et al., 2009), selecting features (Wasserman, 2011; Weston, Mukherjee, et al., 2001), and the prevalence of non-ignorable missing data.

This paper describes CrossCat, a general-purpose Bayesian method for analyzing high-dimensional mixed-type datasets that aims to mitigate these challenges. CrossCat is based on approximate inference in a hierarchical, nonparametric Bayesian model. This model is comprised of an “outer” Dirichlet process mixture over the columns of a table, with components that are themselves independent “inner” Dirichlet process mixture models over the rows. CrossCat is parameterized on a per-table basis by data type specific component models — for example, Beta-Bernoulli models for binary values and Normal-Gamma models for numerical values. Each “inner” mixture is solely responsible for modeling a subset of the variables. Each hypothesis assumes a specific set of marginal dependencies and independencies. This formulation supports scalable algorithms for learning and prediction, specifically a collapsed MCMC scheme that marginalizes out all but the latent discrete state and hyper parameters.

The name “CrossCat” is derived from the combinatorial skeleton of this probabilistic model. Each approximate posterior sample represents a *cross-categorization* of the input data table. In a cross-categorization, the variables are partitioned into a set of *views*, with a separate partition of the entities into *categories* with respect to the variables in each *view*. Each (*category, variable*) pair contains the sufficient statistics or latent state needed by its associated component model. See Figure 1 for an illustration of this structure. From the standpoint of structure learning, CrossCat finds multiple, cross-cutting categorizations or clusterings of the data table. Each non-overlapping system of categories is context-sensitive in that it explains a different subset of the variables. Conditional densities are straightforward to calculate and to sample from. Doing so first requires dividing the conditioning and target variables into views, then sampling a category for each view. The distribution on categories must reflect the values of the conditioning variables. After choosing a category it is straightforward to sample predictions or evaluate predictive densities for each target variable by using the appropriate component model.

Standard approaches for inferring representations for joint distributions from data, such as Bayesian networks, mixture models, and sparse multivariate Gaussians, each exhibit complementary strengths and limitations. Each method exhibits distinct strengths and weaknesses:

1. Bayesian networks and structure learning.

The main advantage offered by Bayesian networks in this setting is that they can use a separate network to model each group of variables. From a statistical perspective, Bayes nets may be effective for sufficiently large, purely discrete datasets where all variables are observed and no hidden variables are needed to accurately model the true data generator. The core modeling difficulty is that many relevant joint distributions are either impossible to represent using a Bayesian network or require prohibitively complex parameterizations. For example, without hidden variables, Bayes nets must emulate the effect of any hidden variables by implicitly marginalizing them out, yielding a dense set of connections. These artificial edges can reduce statistical efficiency: each new parent for a given node can multiplicatively increase

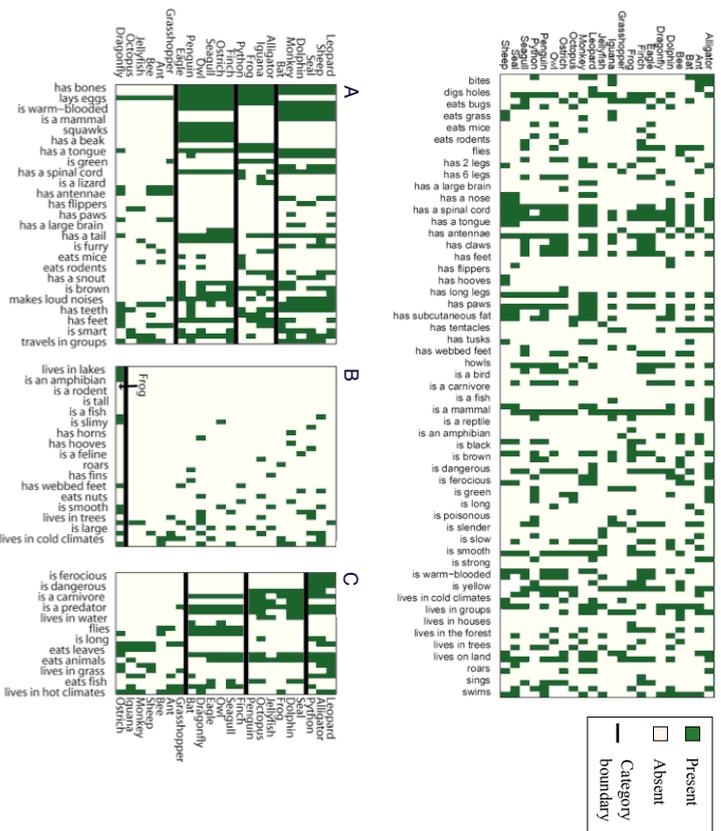


Figure 1: An illustration of the latent structure posited by cross-categorization on a dataset of common-sense judgments about animals. The figure shows the raw input data and one posterior sample from a dataset containing animals and their properties. CrossCat finds three independent signals, or views. A taxonomic clustering (left, A) comprises groups of mammals, amphibians and reptiles, birds, and invertebrates, and explains primarily anatomical and physiological variables. An ecological clustering (right, C) cross-cuts the taxonomic groups and specifies groups of carnivorous land predators, sea animals, flying animals, and other land animals, and explains primarily habitat and behavior variables. Finally all animals (except frogs) are lumped together into a cluster of miscellaneous features (center, B) that accounts for a set of idiosyncratic or sparse “noise” variables.

the number of parameters to estimate (Elidan, Lohner, Friedman, and Koller, 2001; Elidan and Friedman, 2005). There are also computational difficulties. First, there are no known scalable techniques for fully Bayesian learning of Bayesian networks, so posterior uncertainty about the dependence structure is lost. Second, even when the training data is fully observed, i.e. all variables are observed, search through the space of networks is computationally demanding. Third, if the data is incomplete (or hidden variables are posited), a complex inference subproblem needs to be solved in the inner loop of structure search.

2. Parametric and semi-parametric mixture modeling.

Mixtures of simple parametric models have several appealing properties in this setting. First, they can accurately emulate the joint distribution within each group of variables by introducing a sufficiently large number of mixture components. Second, heterogeneous data types are naturally handled using independent parametric models for each variable chosen based on the type of data it contains. Third, learning and prediction can both be done via MCMC techniques that are linear time per iteration, with constant factors and iteration counts that are often acceptable in practice. Unfortunately, mixture models assume that all variables are (marginally) coupled through the latent mixture component assignments. As a result, posterior samples will often contain good categorizations for one group of variables, but these same categories treat all other groups of mutually dependent variables as noise. This can lead to dramatic under-fitting in high dimensions or when missing values are frequent; this paper includes experiments illustrating this failure mode. Thus if the total number of variables is small enough, and the natural cluster structure of all groups of variables is sufficiently similar, and there is enough data, mixture models may perform well.

3. Multivariate Gaussians with sparse inverse covariances.

High-dimensional continuous distributions are often modeled as multivariate Gaussians with sparse conditional dependencies (Meinshausen and Bühlmann, 2006). Several parameter estimation techniques are available; see e.g. (Friedman, Hastie, and Tibshirani, 2008). The pairwise dependencies produced by these methods form an undirected graph. The underlying assumptions are most appropriate when the number of variables and observations are sufficiently large, the data is naturally continuous and fully observed, and the joint distribution is approximately unimodal. A key advantage of these methods is the availability of fast algorithms for parameter estimation (though extensions for handling missing values require solving challenging non-convex optimization problems (Städler and Bühlmann, 2012)). These methods also have two main limitations. First, the assumption of joint Gaussianity is unrealistic in many situations (Wasserman, 2011). Second, discrete values must be transformed into numerical values; this invalidates estimates of predictive uncertainty, and can generate other surprising behaviors.

CrossCat combines key computational and statistical strengths of each of these methods. As with nonparametric mixture modeling, CrossCat admits a scalable MCMC algorithm for posterior sampling, handles incomplete data, and does not impose restrictions on data types. CrossCat also preserves the asymptotic consistency of density estimation via Dirichlet process mixture modeling

(Dunson and Xing, 2009), and can emulate a broad class of generative processes and joint distributions given enough data. However, unlike mixture modeling but like Bayesian network structure learning, CrossCat can also detect independencies between variables. The “outer” Dirichlet process mixture partitions variables into groups that are independent of one another. As with estimation of sparse multivariate Gaussians (but unlike Bayesian network modeling), CrossCat can handle complex continuous distributions and report pairwise measures of association between variables. However, in CrossCat, the couplings between variables can be nonlinear and heteroscedastic, and induce complex, multi-modal distributions. These statistical properties are illustrated using synthetic tests designed to strain the CrossCat modeling assumptions and inference algorithm.

This paper illustrates the flexibility of CrossCat by applying it to several exploratory analysis and predictive modeling tasks. Results on several real-world datasets of up to 10 million cells are described. Examples include measures of hospital cost and quality, voting records, US state-level unemployment time series, and handwritten digit images. These experiments show that CrossCat can extract latent structures that are consistent with accepted findings and common-sense knowledge in multiple domains. They also show that CrossCat can yield favorable predictive accuracy as compared to generative, discriminative, and model-free baselines.

The remainder of this paper is organized as follows. This section concludes with a discussion of related work. Section 2 focuses on generative model and approximate inference scheme behind CrossCat. Section 3 describes empirical results, and section 4 contains a broad discussion and summary of contributions.

1.1 Related Work

The observation that multiple alternative clusterings can often explain data better than a single clustering is not new to this paper. Methods for finding multiple clusterings have been developed in several fields, including by the authors of this paper (see e.g. Niu, Dy, and Jordan, 2010; Cui, Fern, and Dy, 2007; Guan, Dy, Niu, and Ghahramani, 2010; Li and Shafto, 2011; Rodriguez and Ghosh, 2009; Shafto, Kemp, Mansinghka, Gordon, and Tenenbaum, 2006; Shafto, Kemp, Mansinghka, and Tenenbaum, 2011; Ross and Zemel, 2006). For example, Ross and Zemel (2006) used an EM approach to fit a parametric mixture of mixtures and applied it to image modeling. As nonparametric mixtures and model selection over finite mixtures can behave similarly, it might seem that a nonparametric formulation is a small modification. In fact, nonparametric formulation presented here is based on a super-exponentially larger space of model complexities that includes all possible numbers and sizes of views, and for each view, all possible numbers and sizes of categories. This expressiveness is necessary for the broad applicability of CrossCat. Cross-validation over this set is intractable, motivating the nonparametric formulation and sampling scheme used in this paper.

It is instructive to compare and contrast CrossCat with related hierarchical Bayesian models that link multiple Dirichlet process mixture models, such as the nested Dirichlet process (Rodriguez, Dunson, and Gelfand, 2008) and the hierarchical Dirichlet process (Teh, Jordan, Beal, and Blei, 2006). See Jordan (2010) for a thorough review. This section contains a brief discussion of the most important similarities and differences. The hierarchical Dirichlet process applies independent Dirichlet processes to each dataset, whose atoms are themselves draws from a single shared Dirichlet process. It thus enables a single pool of clusters to be shared and sparsely expressed in otherwise independent clusterings of several datasets. The differences are substantial. For example, with the hierarchical Dirichlet process, the number of Dirichlet processes is fixed in advance. In CrossCat,

each atom on one Dirichlet process is associated with its own Dirichlet process, and inference is used to determine the number that will be expressed in a given finite dataset.

The nested Dirichlet process shares this combinatorial structure with CrossCat, but has been used to build very different statistical models. (Rodriguez et al., 2008) introduces it as a model for multiple related datasets. The model consists of a Dirichlet process mixture over datasets where each component is another Dirichlet process mixture models over the items in that dataset. From a statistical perspective, it can be helpful to think of this construction as follows. First, a top-level Dirichlet process is used to cluster datasets. Second, all datasets in the same cluster are pooled and their contents are modeled via a single clustering, provided by the lower-level Dirichlet process mixture model associated with that dataset cluster.

The differences between CrossCat and the nested Dirichlet process are clearest in terms of the nested Chinese restaurant process representation of the nested DP (Blei, Griffiths, Jordan, and Tenenbaum, 2004; Blei, Griffiths, and Jordan, 2010). In a 2-layer nested Chinese restaurant process, there is one customer per data vector. Each customer starts at the top level, sits a table at their current level according to a CRP, and descends to the CRP at the level below that the chosen table contains. In CrossCat, the top level CRP partitions the variables into views, and the lower level CRPs partition the data vectors into categories for each view. If there are K tables in top CRP, i.e. the dataset is divided into K views, then adding one datapoint leads to the seating of K new customers at level 2. Each of these customers is deterministically assigned to a distinct table. Also, whenever a new customer is created at the top restaurant, in addition to creating a new CRP at the level below, R customers are immediately seated below it (one per row in the dataset).

Close relatives of CrossCat have been introduced by the authors of this paper in the cognitive science literature, and also by other authors in machine learning and statistics. This paper goes beyond this previous work in several ways. Guan et al. (2010) uses a variational algorithm for inference, while Rodriguez and Ghosh (2009) uses a sampler for the stick breaking representation for a Pitman-Yor (as opposed to Dirichlet Process) variant of the model. CrossCat is instead based on samplers that (i) operate over the combinatorial (i.e. Chinese restaurant) representation of the model, not the stick-breaking representation, and (ii) perform fully Bayesian inference over all hyper-parameters. This formulation leads to CrossCat’s scalability and robustness. This paper includes results on tables with millions of cells, without any parameter tuning, in contrast to the 148x500 gene expression subsample analyzed in Rodriguez and Ghosh (2009). These other papers include empirical results comparable in size to the authors’ experiments from Shafto et al. (2006) and Mansinghka, Jonas, Petschulat, Cronin, Shafto, and Tenenbaum (2009); these are 10-100x smaller than some of the examples from this paper. Additionally, all the previous work on variants of the CrossCat model focused on clustering, and did not articulate its use as a general model for high-dimensional data generators. For example, Guan et al. (2010) does not include predictions, although Rodriguez and Ghosh (2009) does discuss an example of imputation on a 51x26 table.

To the best of our knowledge, this paper is the first to introduce a fully Bayesian, domain-general, semi-parametric method for estimating the joint distributions of high-dimensional data. This method appears to be the only joint density estimation technique that simultaneously supports heterogeneous data types, detects independencies, and produces representations that support efficient prediction. This paper is also the first to empirically demonstrate the effectiveness of the underlying probabilistic model and inference algorithm on multiple real-world datasets with mixed types, and the first to compare predictions and latent structures from this kind of model against multiple generative, discriminative and model-free baselines.

2. The CrossCat Model and Inference Algorithm

CrossCat is based on inference in a column-wise Dirichlet process mixture of Dirichlet process mixture models (Escobar and West, 1995; Rasmussen, 2000) over the rows. The “outer” or “column-wise” Dirichlet process mixture determines which dimensions/variables should be modeled together at all, and which should be modeled independently. The “inner” or “row-wise” mixtures are used to summarize the joint distribution of each group of dimensions/variables that are stochastically assigned to the same modeling subproblem.

This paper presents the Dirichlet processes in CrossCat via the convenient Chinese restaurant process representation (Pitman, 1996). Recall that the Dirichlet process is a stochastic process that maps an arbitrary underlying base measure into a measure over discrete atoms, where each atom is associated with a single draw from the base measure. In a set of repeated draws from this discrete measure, some atoms are likely to occur multiple times. In nonparametric Bayesian mixture modeling, each atom corresponds to a set of parameters for some mixture component; “popular” atoms correspond to mixture components with high weight. The Chinese restaurant process is a stochastic process that corresponds to the discrete residue of the Dirichlet process. It is sequential, easy to describe, easy to simulate, and exchangeable. It is often used to represent nonparametric mixture models as follows. Each data item is viewed as a customer at a restaurant with an infinite number of tables. Each table corresponds to a mixture component; the customers at each table thus comprise the groups of data that are modeled by the same mixture component. The choice probabilities follow a simple “rich-gets-richer” scheme. Let m_j be the number of customers (data items) seated at a given table j , and z_i be the table assignment of customer i (with the first table $z_0 = 0$), then the conditional probability distribution governing the Chinese restaurant process with concentration parameter α is:

$$Pr(z_i = j) \propto \begin{cases} \alpha & \text{if } j = \max(\bar{z}) + 1 \\ m_j & \text{o.w.} \end{cases}$$

This sequence of conditional probabilities induces a distribution over the partitions of the data that is equivalent to the marginal distribution on equivalence classes of atom assignments under the Dirichlet process. The Chinese restaurant process provides a simple but flexible modeling tool: the number of components in a mixture can be determined by the data, with support over all logically possible clusterings. In CrossCat, the *number* of Chinese restaurant processes (over the table assignment determined by the number of tables in a Chinese restaurant process over the columns. The data itself is modeled by datatype-specific component models for each dimension (column) of the target table.

2.1 The Generative Process

The generative process behind CrossCat unfolds in three steps:

1. **Generating hyper-parameters and latent structure.** First, the hyper-parameters $\tilde{\lambda}_d$ for the component models for each dimension are chosen from a vague hyper-prior V_d that is appropriate¹ for the type of data in d . Second, the concentration parameter α for the outer

¹ The hyper-prior must only assign positive density to valid hyper-parameter values and be sufficiently broad for the marginal distribution for a single data cell has comparable spread to the actual data being analyzed. We have explored multiple hyper-priors that satisfy these constraints on analyses similar to those from this paper; there was little

Chinese restaurant process is sampled from a vague gamma hyper-prior. Third, a partition of the variables into views, \bar{z} , is sampled from this outer Chinese restaurant process. Fourth, for each view, $v \in \bar{z}$, a concentration parameter α_v is sampled from a vague hyper-prior. Fifth, for each view v , a partition of the rows y^v is drawn using the appropriate inner Chinese restaurant process with concentration α_v .

2. **Generating category parameters for uncollapsed variables.** This paper uses u_d as an indicator of whether a given variable/dimension d is uncollapsed ($u_d = 1$) or collapsed ($u_d = 0$). For each uncollapsed variable, parameters θ_c^d must be generated for each category c from a datatype-compatible prior model M_d .

3. **Generating the observed data given hyper-parameters, latent structure, and parameters.** The dataset $\mathbf{X} = \{x^{(i,d)}\}$ is generated separately for each variable d and for each category $c \in \mathcal{Y}^d$ in the view $v = z_d$ for that variable. For uncollapsed dimensions, this is done by repeatedly simulating from a likelihood model L_d . For collapsed dimensions, we use an exchangeably coupled model ML_d to generate all the data in each category at once.

The details of the CrossCat generative process are as follows:

1. Generate α_D , the concentration hyper-parameter for the Chinese Restaurant Process over dimensions, from a generic Gamma hyper-prior: $\alpha_D \sim \text{Gamma}(k = 1, \theta = 1)$.
2. For each dimension $d \in D$:
 - (a) Generate hyper-parameters $\tilde{\lambda}_d$ from a data type appropriate hyper-prior with density $p(\tilde{\lambda}_d) = V_d(\tilde{\lambda}_d)$, as described above. Binary data is handled by an asymmetric Beta-Bernoulli model with pseudocounts $\tilde{\lambda}_d = [\alpha_d, \beta_d]$. Discrete data is handled by a symmetric Dirichlet-Discrete model with concentration parameter $\tilde{\lambda}_d$. Continuous data is handled by a Normal-Gamma model with $\tilde{\lambda}_d = (\mu_d, K_d, \nu_d, \tau_d)$, where μ_d is the mean, K_d is the effective number of observations, ν_d is the degrees of freedom, and τ_d is the sum of squares.
 - (b) Assign dimension d to a view z_d from a Chinese Restaurant Process with concentration hyper-parameter α_D , conditional on all previous draws: $z_d \sim \text{CRP}\{z_0, \dots, z_{d-1}\}; \alpha_D$
3. For each view v in the dimension partition \bar{z} :
 - (a) Generate α_v , the concentration hyper-parameter for the Chinese Restaurant Process over categories in view v , from a generic hyper-prior: $\alpha_v \sim \text{Gamma}(k = 1, \theta = 1)$.
 - (b) For each observed data point (i.e. row of the table) $r \in R$, generate a category assignment y_r^v from a Chinese Restaurant Process with concentration parameter α_v , conditional on all previous draws: $y_r^v \sim \text{CRP}(\{y_0^v, \dots, y_{r-1}^v\}; \alpha_v)$

apparent variation. Examples for strictly positive, real-valued hyper-parameters include vague Gamma($k = 1, \theta = 1$) hyper-prior, uniform priors over a broad range, and both linear and logarithmic discretizations. Our reference implementation uses a set of simple data-dependent heuristics to determine sufficiently broad ranges. Chinese restaurant process concentration parameters are given 100-bin log-scale grid discrete priors; concentration parameters for the finite-dimensional Dirichlet distributions used to generate component parameters for discrete data have the same form. For Normal-Gamma models, $\min(\tilde{K}_{(d)}) \leq \mu_d \leq \max(\tilde{K}_{(d)})$.

- (c) For each category c in the row partition for this view \bar{y}^c :
- i. For each dimension d such that $u_d = 1$ (i.e. its component models are uncollapsed), generate component model parameters θ_c^d from the appropriate prior with density $M_d(\cdot; \bar{\lambda}_d)$ using hyper-parameters $\bar{\lambda}_d$, as follows:
 - A. For binary data, we have a scalar θ_c^d equal to the probability that dimension d is equal to 1 for rows from category c , drawn from a Beta distribution: $\theta_c^d \sim \text{Beta}(\alpha_d, \beta_d)$, where values from the hyper-parameter vector $\bar{\lambda}_d = [\alpha_d, \beta_d]$.
 - B. For categorical data, we have a vector-valued θ_c^d of probabilities, drawn from a symmetric Dirichlet distribution with concentration parameter λ_d : $\theta_c^d \sim \text{Dirichlet}(\lambda_d)$.
 - C. For continuous data, we have $\theta_c^d = (\mu_c^d, \sigma_c^d)$, the mean and variance of the data in the component, drawn from a Normal-Gamma distribution $(\mu_c^d, \sigma_c^d) \sim \text{NormalGamma}(\bar{\lambda}_d)$.
 - ii. Let $\bar{x}_{(\cdot, d)}$ contain all $x_{(r, d)}$ in this component, i.e. for r such that $y_r^d = c$. Generate the data in this component, as follows:
 - A. If $u_d = 1$, i.e. d is uncollapsed, then generate each $x_{(r, d)}$ from the appropriate likelihood model $L_d(\cdot; \theta_c^d)$. For binary data, we have $x_{(r, d)} \sim \text{Bernoulli}(\theta_c^d)$; for categorical data, we have $x_{(r, d)} \sim \text{Multinomial}(\theta_c^d)$; for continuous data, we have $x_{(r, d)} \sim \text{Normal}(\mu_c^d, \sigma_c^d)$.
 - B. If $u_d = 0$, so d is collapsed, generate the entire contents of $\bar{x}_{(\cdot, d)}$ by directly simulating from the marginalized component model that with density $ML_d(\bar{x}_{(\cdot, d)}; \bar{\lambda}_d)$. One approach is to sample from the sequence of predictive distributions $P(x_{(r, d)} | \bar{x}_{(\cdot, d)}^r; \bar{\lambda}_d)$, induced by M_d and L_d , indexing over rows r , in c .
- The key steps in this process can be concisely described:

$$\begin{aligned}
 \alpha_D &\sim \text{Gamma}(k = 1, \theta = 1) \\
 \bar{\lambda}_d &\sim V_d(\cdot) \\
 z_d &\sim \text{CRP}(\{z_i \mid i \neq d\}; \alpha_D) \\
 \alpha_r &\sim \text{Gamma}(k = 1, \theta = 1) \\
 y_r^c &\sim \text{CRP}(\{y_i^c \mid i \neq r\}; \alpha_r) \\
 \bar{\theta}_c^d &\sim M_d(\cdot; \bar{\lambda}_d) \\
 \bar{x}_{(\cdot, d)} = \{x_{(r, d)} \mid y_r^d = c\} &\sim \begin{cases} \prod_r L_d(\bar{\theta}_c^d) & \text{if } u_d = 1 \\ ML_d(\bar{\lambda}_d) & \text{if } u_d = 0 \end{cases}
 \end{aligned}$$

2.2 The Joint Probability Density

Recall that the following dataset-specific information is needed to fully specify the CrossCat model:

1. $V_d(\cdot)$, a generic hyper-prior of the appropriate type for variable/dimension d .
2. $\{u_d\}$, the indicators for which variables are uncollapsed.
3. $M_d(\cdot)$ and $L_D(\cdot) \forall d$ s.t. $u_d = 1$, a datatype-appropriate parameter prior (e.g. a Beta prior for binary data, Normal-Gamma for continuous data, or Dirichlet for discrete data) and likelihood model (e.g. Bernoulli, Normal or Multinomial).
4. $ML_d(\cdot) \forall d$ s.t. $u_d = 0$, a datatype-appropriate marginal likelihood model, e.g. the collapsed version of the conjugate pair formed by some M_d and L_d .
5. $T_d(\{x\})$, the sufficient statistics for the component model for some collapsed dimension d from a subset of the data $\{x\}$. Arbitrary non-conjugate component models can be numerically collapsed by choosing $T_d(\{x\}) = \{x\}$.

This paper will use **CC** to denote the information necessary to capture the dependence of CrossCat on the data \mathbf{X} . This includes the view concentration parameter α_D , the variable-specific hyperparameters $\{\bar{\lambda}_d\}$, the view partition \bar{z} , the view-specific concentration parameters $\{\alpha_r\}$ and row partition $\{y^r\}$, and the category-specific parameters $\{\theta_c^d\}$ or sufficient statistics $T_d(\bar{x}_{(\cdot, d)})$. This paper will also overload ML_d, M_d, V_d, L_d , and CRP to each represent both probability density functions and stochastic simulators; the distinction should be clear based on context. Given this notation, we have:

$$\begin{aligned}
 P(\mathbf{CC}, \mathbf{X}) &= P(\mathbf{X}; \{\theta_c^d\}, \{y^r\}, \{\bar{\lambda}_d\}, \bar{z}, \alpha_D) \\
 &= e^{-\alpha_D (\prod_{d \in D} V_d(\bar{\lambda}_d))} \text{CRP}(\bar{z}; \alpha_D) (\prod_{r \in \bar{z}} e^{-\alpha_r} \text{CRP}(y^r; \alpha_r)) \\
 &\quad \times \prod_{v \in \bar{z}} \prod_{c \in \bar{y}^v} \prod_{d \in \{i \text{ s.t. } z_i = v\}} \begin{cases} ML_d(T_d(\bar{x}_{(\cdot, d)}); \bar{\lambda}_d) & \text{if } u_d = 1 \\ M_d(\theta_c^d; \bar{\lambda}_d) \prod_{r \in c} L_d(x_{(r, d)}; \theta_c^d) & \text{if } u_d = 0 \end{cases}
 \end{aligned}$$

2.3 Hypothesis Space and Modeling Capacity

The modeling assumptions encoded in CrossCat are designed to enable it to emulate a broad class of data generators. One way to assess this class is to study the full hypothesis space of CrossCat, that is, all logically possible cross-categorizations. Figure 2 illustrates the version of this space that is induced by a 4 row, 3 column dataset. Each cross-categorization corresponds to a model structure — a set of dependence and independence assumptions — that is appropriate for some set of statistical situations. For example, conditioned on the hyper-parameters, the dependencies between variables and data values can be either dense or sparse. A group of dependencies will exhibit a unimodal joint distribution if they are modeled using only a single cluster. Strongly bimodal or multi-modal distributions as well as nearly unimodal distributions with some outliers are recovered by varying the number of clusters and their size. The prior favors stochastic relationships between groups of variables, but also supports (nearly) deterministic models; these correspond to structures with a large number of clusters that share low-entropy component models.

The CrossCat generative process favors hypotheses with multiple views and multiple categories per view. A useful rule of thumb is to expect $O(\log(D))$ views with $O(\log(R))$ categories each a priori. Asserting that a dataset has several views and several categories per view corresponds to asserting that the underlying data generator exhibits several important statistical properties. The first is that the dataset contains variables that arise from several distinct causal processes, not just a

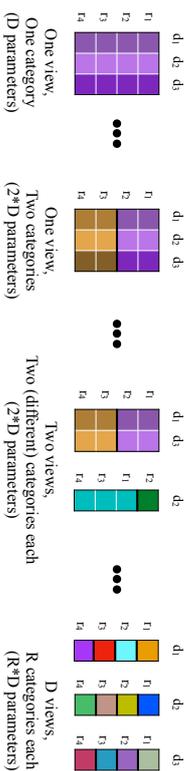


Figure 2. **Model structures drawn from the space of all logically possible cross-categorizations of a 4 row, 3 column dataset.** In each structure, all data values (cells) that are governed by the same parametric model are shown in the same color. If two cells have different colors, they are modeled as conditionally independent given the model structure and hyper-parameters. In general, the space of all cross-categorizations contains a broad class of simple and complex data generators. See the main text for details.

single one. The second is that these processes cannot be summarized by a single parametric model, and thus induce non-Gaussian or multi-modal dependencies between the variables.

2.4 Posterior Inference Algorithm

Posterior inference is carried out by simulating an ergodic Markov chain that converges to the posterior (Gilks, 1999; Neal, 1998). The state of the Markov chain is a data structure storing the cross-categorization, sufficient statistics, and all uncollapsed parameters and hyper-parameters. Figure 3 shows several sampled states from a typical run of the inference scheme on the dataset from Figure 1.

The CrossCat inference Markov chain initializes a candidate state by sampling it from the prior². The transition operator that it iterates consists of an outer cycle of several kernels, each performing cycle sweeps that apply other transition operators to each segment of the latent state. The first is a cycle kernel for inference over the outer CRP concentration parameter α and a cycle of kernels over the inner CRP concentration parameters $\{\alpha_r\}$ for each view. The second is a cycle of kernels for inference over the hyper-parameters $\{\tilde{\lambda}_d\}$ for each dimension. The third is a kernel for inference over any uncollapsed parameters $\{\tilde{\theta}_r^i\}$. The fourth is a cycle over dimensions of an inter-view auxiliary variable Gibbs kernel that shuffles dimensions between views. The fifth is itself a cycle over views of cycles that sweep a single-site Gibbs sampler over all the rows in the given view. This chain corresponds to the default auxiliary variable Gibbs sampler that the Venture probabilistic programming platform (Mansinghka, Selsam, and Perov, 2014) produces when given the CrossCat model written as a probabilistic program.

More formally, the Markov chain used for inference is a cycle over the following kernels:

1. Ancestrally, this initialization appears to yield the best inference performance overall. One explanation can be found by considering a representative subproblem of inference in CrossCat: performing inference in one of the inner CRP mixture models. A maximally dispersed initialization, with each of the N rows in its own category, requires $O(N^2)$ time for its first Gibbs sweep. An initialization that places all rows in a single category requires $O(1)$ time for its first sweep but can spend many iterations stuck in or near the “low resolution” model encoded by this initial configuration.

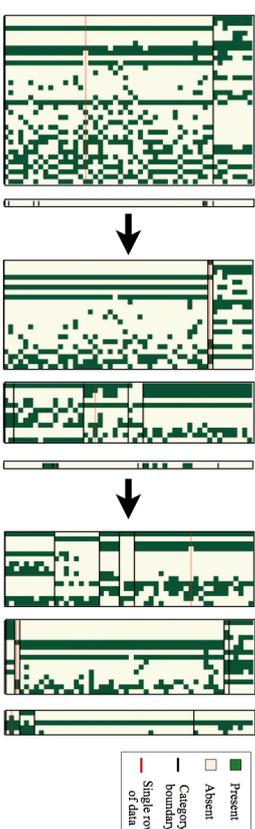


Figure 3. **Snapshots of the Markov chain for cross-categorization on a dataset of human object-feature judgments.** Each of the three states shows a particular cross-categorization that arose during a single Markov chain run, automatically rendered using the latent structure from cross-categorization to inform the layout. Black horizontal lines separate categories within a view. The red horizontal line follows one row of the dataset. Taken from left to right, the three states span a typical run of roughly 100 iterations: the first is while the chain appears to be converging to a high probability region, while the last two illustrate variability within that region.

1. **Concentration hyper-parameter inference: updating α_d and each element of $\{\alpha_r\}$.** Sample α_d and all the $\alpha_{r,s}$ for each view via a discretized Gibbs approximation to the posterior, $\alpha_d \sim P(\alpha_d | \bar{z})$ and $\alpha_r \sim P(\alpha_r | \bar{y}^r)$. For each α , this involves scoring the CRP marginal likelihood at a fixed number of grid points — typically ~ 100 — and then re-normalizing and sampling from the resulting discrete approximation.

2. **Component model hyper-parameter inference: updating the elements of $\{\tilde{\lambda}_d\}$.** For each dimension, for each hyper-parameter, discretize the support of the hyper-prior and numerically sample from an approximate hyper-posterior distribution. That is, implement an appropriately-binned discrete approximation to a Gibbs sampler for $\tilde{\lambda}_d \sim P(\tilde{\lambda}_d | \bar{x}_d, \bar{y}^d)$ (i.e. we condition on the vertical slice of the input table described by the hyper-parameters, and the associated latent variables). For conjugate component models, the probabilities depend only on the sufficient statistics needed to evaluate this posterior. Each hyper-parameter adjustment requires an operation linear in the number of categories, since the scores for each category (i.e. the marginal probabilities) must be recalculated, after each category’s statistics are updated. Thus each application of this kernel takes time proportional to the number of dimensions times the maximum number of categories in any view.

3. **Category inference: updating the elements of $\{y^r\}$ via Gibbs with auxiliary variables.** For each entry in each view, this transition operator samples a new category assignment from its conditional posterior. A variant of Algorithm 8 from (Neal, 1998) (with $m=1$) is used to handle uncollapsed dimensions.

The category inference transition operator will sample y_r^i , the categorization for row r in view v , according to its conditioned distribution given the other category assignments y^r , parameters $\{\tilde{\theta}_r^i\}$ and auxiliary parameters. If $u_d = 0 \forall d$ s.t. $z_d = v$, i.e. there are no uncollapsed dimensions in this view, then this reduces to the usual collapsed Gibbs sampler applied to

the subset of data within the view. Otherwise, let $\{\tilde{\phi}^d\}$ denote auxiliary parameters for each uncollapsed dimension d (where $u_d = 1$) of the same form as θ_c^d . Before each transition, these parameters are chosen as follows:

$$\tilde{\phi}^d \sim \begin{cases} \delta_{\theta_c^d} & \text{if } y_r^+ = y_j^+ \iff r = j \\ M_d(\tilde{\lambda}_d) & \text{o.w. } (y_r^+ \in \tilde{y}_{=r}^-) \end{cases}$$

In this section, c^+ will denote the category associated with the auxiliary variable. If $y_r^+ \in \tilde{y}_{=r}^-$ then $c^+ = \max(\tilde{y}_v^-) + 1$, i.e. a wholly new category will be created, and by sampling $\tilde{\phi}^d$ this category will have newly sampled parameters. Otherwise, $c^+ = y_r^+$, i.e. row r was a singleton, so its previous category assignment and parameters will be reused.

Given the auxiliary variables, we can derive the target density of the transition operator by expanding the joint probability density:

$$\begin{aligned} y_r^+ &\sim P(y_r^+ | \tilde{y}_{=r}^-, \tilde{\lambda}_d, \{x_{(c,d)}\} | d \text{ s.t. } z_d = v), \{\{\theta_c^d | c \in \tilde{y}_{=r}^-\} | d \text{ s.t. } z_d = v \text{ and } u_d = 1\}, \{\tilde{\phi}^d\} \\ &\propto \text{CRP}(y_r^+, \tilde{y}_{=r}^-, \alpha_0) \\ &\quad \times \prod_{d \in \{l \text{ s.t. } z_d = v\}} \begin{cases} M_d(T_d(\tilde{x}_{(c,d)}^-, \tilde{\lambda}_d)) & \text{if } u_d = 0 \\ M_d(\tilde{\theta}_c^d; \tilde{\lambda}_d) \prod_{c \in L_d(x_{(c,d)})} & \text{if } u_d = 1 \text{ and } y_r^+ \in \tilde{y}_{=r}^- \\ M_d(\tilde{\theta}_c^d; \tilde{\lambda}_d) \prod_{c \in L_d(x_{(c,d)})} & \text{if } u_d = 1 \text{ and } y_r^+ = c^+ \notin \tilde{y}_{=r}^- \end{cases} \end{aligned}$$

The probabilities this transition operator needs can be obtained by iterating over possible values for y_r^+ , calculating their joint densities, and re-normalizing numerically. These operations can be implemented efficiently by maintaining and incrementally modifying a representation of CC, updating sufficient statistics and a joint probability accumulator after each change (Mansinghka, 2007). The complexity of resampling y_r^+ for all rows r and views v is $O(VRCD)$, where V is the number of views, R the number of rows, C the maximum number of categories in any view, and D is the number of dimensions.

4. **Inter-view inference: updating the elements of \tilde{z} via Gibbs with auxiliary variables.** For each dimension d , this transition operator samples a new view assignment z_d from its conditional posterior. As with the category inference kernel, this can be viewed as a variant of Algorithm 8 from (Neal, 1998) (with $m = 1$), applied to the ‘‘outer’’ Dirichlet process mixture model in CrossCat. This mixture has uncollapsed, non-conjugate component models that are themselves Dirichlet process mixtures.

Let v^+ be the index of the new view. The auxiliary variables are α_{v^+} , \tilde{y}^{v^+} and $\{\theta_c^d | c \in \tilde{y}^{v^+}\}$ (if $u_d = 1$). If $z_d \in \tilde{z}^{-d}$, then $v^+ = \max(\tilde{z}) + 1$, and the auxiliary variables are sampled from their priors. Otherwise, $v^+ = z_d$, and the auxiliary variables are deterministically set to the values associated with z_d . Given values for these variables, the conditional distribution for z_d can be derived as follows:

$$\begin{aligned} z_d &\sim P(z_d | \alpha_D, \tilde{\lambda}_d, \tilde{z}^{-d}, \alpha_{v^+}, \{\tilde{y}^{v^+}\}, \{\{\theta_c^d | c \in \tilde{y}^c\} | j \in D\}, \mathbf{X}) \\ &\propto \text{CRP}(z_d; \tilde{z}^{-d}, \alpha_D) \prod_{c \in \tilde{y}^{v^+}} \begin{cases} M_{L_d}(T_d(\tilde{x}_{(c,d)}^-, \tilde{\lambda}_d)) & \text{if } u_d = 1 \\ M_d(\tilde{\theta}_c^d; \tilde{\lambda}_d) \prod_{c \in L_d(x_{(c,d)})} & \text{if } u_d = 0 \end{cases} \end{aligned}$$

This transition operator shuffles individual columns between views, weighing their compatibility with each view by multiplying likelihoods for each category. A full sweep thus has time complexity $O(DVCR)$. Note that if a given variable is a poor fit for its current view, its hyper-parameters and parameters will be driven to reduce the dependence of the likelihood for that variable on its clustering. This makes it more likely for row categorizations proposed from the prior to be accepted.

Inference over the elements of \tilde{z} can also be done via a mixture of a Metropolis-Hastings birth-death kernel to create new views with a standard Gibbs kernel to reassign dimensions among pre-existing views. In our experience, both transition operators yield comparable results on real-world data; the Gibbs auxiliary variable kernel is presented here for simplicity.

5. **Component model parameter inference: updating $\{\theta_c^d | u_d = 1\}$.** Each dimension or variable whose component models are uncollapsed must be equipped with a suitable ergodic transition operator T that converges to the local parameter posterior $P(\tilde{\theta}_c^d | x_{(c,d)}^-, \tilde{\lambda}_d)$. Exact Gibbs sampling is often possible when L_d and M_d are conjugate.

CrossCat’s scalability can be assessed by multiplying an estimate of how long each transition takes with an estimate of how many transitions are needed to get good results. The experiments in this paper use ~ 10 -100 independent samples. Each sample was based on runs of the inference Markov chain with ~ 100 -1,000 transitions. Taking these numbers as rough constants, scalability is governed by the asymptotic orders of growth. Let R be the number of rows, D the number of dimensions, V the maximum number of views and C the maximum number of categories. The memory needed to store the latent state is the sum of the memory needed to store the D hyper-parameters and view assignments, the VC parameters/sufficient statistics, and the VR category assignments, or $O(D + VC + VR)$. Assuming a fully dense data matrix, the loops in the transition operator described above scale as $O(DC + RDVC + RDVC + DC) = O(RDVC)$, with the RD terms scaling down following the data density.

This paper shows results from both open-source and commercial implementations on datasets of up to ~ 10 million cells³. Because this algorithm is asymptotically linear in runtime with low memory requirements, a number of performance engineering and distributed techniques can be applied to reach larger scales at low latencies. Performance engineering details are beyond the scope of this paper.

2.5 Exploration and Prediction Using Posterior Samples

Each approximate posterior sample provides an estimate of the full joint distribution of the data. It also contains a candidate latent structure that characterizes the dependencies between variables and provides an independent clustering of the rows with respect to each group of dependent variables. This section gives examples of exploratory and predictive analysis problems that can be solved by using these samples. Prediction is based on calculating or sampling from the conditional densities implied by each sample and then either averaging or resampling from the results. Exploratory queries typically involve Monte Carlo estimation of posterior probabilities that assess structural

3. A variation on CrossCat was the basis of Veritable, a general-purpose machine learning system built by Navia Systems/Prior Knowledge Inc. This implementation became a part of Salesforce.com’s predictive analytics infrastructure. At Navia, CrossCat was applied to proprietary datasets from domains such as operations management for retail, clinical virology, and quantitative finance.

properties of the latent variables posited by CrossCat and the dependencies they imply. Examples include obtaining a global map of the pairwise dependencies between variables, selecting those variables that are probably predictive of some target, and identifying rows that are similar in light of some variables of interest.

2.5.1 PREDICTION

Recall that \mathbf{CC} represents a model for the joint distribution over the variables along with sufficient statistics, parameters, a partition of variables into views, and categorizations of the rows in the data \mathbf{X} . Variables representing the latent structure associated with a particular posterior sample $\mathcal{C}C_s$ will all be indexed by s , e.g. $z_{q_s}^s$. Also let \mathcal{Y}^+ represent the category assignment of a new row in view v , and let $\{t_i\}$ and $\{g_j\}$ be the sets of target variables and given variables in a given predictive query.

To generate predictions by sampling from the conditional density on targets given the data, we must simulate

$$\{\hat{x}_i\} \sim p(\{X_{t_i}\} | \{X_{g_j} = x_{g_j}\}, \mathbf{X})$$

Given a set of models, this can be done in two steps. First, from each model, sample a categorization from each view conditioned on the values of the given variables. Second, sample values for each target variable by simulating from the target variable's component model for the sampled category:

$$\begin{aligned} \mathcal{C}C_s &\sim p(\mathbf{CC} | \mathbf{X}) \\ c_s^v &\sim p(\mathcal{Y}^+ | \{X_{g_j} = x_{g_j} | z_{q_s}^s = v\}) \\ \hat{x}_i^v &\sim p(X_{t_i} | z_{q_s}^s = v) = \int L(x_{t_i}; \hat{\theta}_{c_s^v}^v) M(\hat{\theta}_{c_s^v}^v; \vec{\lambda}_{q_i}, \vec{\mu}) d\hat{\theta} \end{aligned}$$

The category kernel from the MCMC inference algorithm can be re-used to sample from c_s^v . Also, sampling from \hat{x}_i^v can be done directly given the sufficient statistics for data types whose likelihood models and parameter priors are conjugate. In other cases, either θ will be represented as part of $\mathcal{C}C_s$ or sampled on demand.

The same latent variables are also useful for evaluating the conditional density for a desired set of predictions:

$$\begin{aligned} p(\{X_{t_i} = x_{t_i}\} | \{X_{g_j} = x_{g_j}\}, \mathbf{X}) \\ &\approx \frac{1}{N} \sum_s p(\{X_{t_i} = x_{t_i}\} | \{X_{g_j} = x_{g_j}\}, \mathbf{CC} = \mathcal{C}C_s) \\ &= \frac{1}{N} \sum_s \prod_{v \in \mathcal{C}^v} \prod_c p(\{X_{t_i} = x_{t_i} | z_{q_s}^s = v\} | \mathcal{Y}^+ = c) p(\mathcal{Y}^+ = c | \{X_{g_j} = x_{g_j} | z_{q_s}^s = v\}) \end{aligned}$$

Many problems of prediction can be reduced to sampling from and/or calculating conditional densities. Examples include classification, regression and imputation. Each can be implemented by forming estimates $\{X_{t_i}^v\}$ of the target variables. By default, the implementation from this paper implements the mean of the predictive to impute continuous values. This is equivalent to choosing the value that minimizes the expected square loss under the empirical distribution induced by a set of predictive samples. For discrete values, the implementation uses the most probable value.

equivalent to minimizing 0-1 loss, and calculates it by directly evaluating the conditional density of each possible value. This approach to prediction can also handle nonlinear and/or stochastic relationships within the set of target variables $\{X_{t_i}\}$ and between the given variables $\{X_{g_j}\}$ and the targets. It is easy to implement in terms of the same sampling and probability calculation kernels that are necessary for inference.

This formulation of prediction scales linearly in the number of variables, categories, and view. It is also sub-linear in the number of variables when dependencies are sparse, and parallelizable over the views, the posterior samples, and the generated samples from the conditional density. Future work will explore the space of tradeoffs between accuracy, latency and throughput that can be achieved using this basic design.

2.5.2 DETECTING DEPENDENCIES BETWEEN VARIABLES

To detect dependencies between groups of variables, it is natural to use a Monte Carlo estimate of the marginal posterior probability that a set of variables $\{q_i\}$ share the same posterior view. Using s as a superscript to select values from a specific sample, we have:

$$\begin{aligned} Pr[z_{q_0}^s = z_{q_1} = \dots = z_{q_n} | \mathbf{X}] &\approx \frac{1}{N} \sum_s Pr[z_{q_0}^s = z_{q_1}^s = \dots = z_{q_n}^s | \mathcal{C}C_s] \\ &= \frac{\#\{s | z_{q_0}^s = z_{q_1}^s = \dots = z_{q_n}^s\}}{N} \end{aligned}$$

These probabilities also characterize the marginal dependencies and independencies that are explicitly represented by CrossCat. For example, pairwise co-assignment in \vec{z} determines⁴ pairwise marginal independence under the generative model:

$$X_{q_i} \perp\!\!\!\perp X_{q_j} \iff z_{q_i} \neq z_{q_j}$$

The results in this paper often include the “z-matrix” of marginal dependence probabilities $\mathbf{Z} = [Z_{(i,j)}]$, where $Z_{(i,j)} = 1 - Pr[X_i \perp\!\!\!\perp X_j | \mathbf{X}]$. This measure is used primarily for simplicity; other measures of the presence or strength of predictive relationships are possible.

2.5.3 ESTIMATING SIMILARITY BETWEEN ROWS

Exploratory analyses often make use of “similarity” functions defined over pairs of rows. One useful measure of similarity is given by the probability that two pieces of data were generated from the same statistical model (Tenenbaum and Griffiths, 2001; Ghahramani and Heller, 2006). CrossCat naturally induces a context-sensitive similarity measure between rows that has this form: the probability that two items come from the same category in some context. Here, contexts are defined by target variables, and comprise the set of views in which that variable participates (weighted by their probability). This probability is straightforward to estimate given a collection of samples:

⁴ This paper defines independence in terms of the generative process and latent variables. Two variables in different views are explicitly independent, but two variables in the same view are coupled through the latent cluster assignment. This is clear if there are multiple clusters. Even if there is just one cluster, if q_i remains nonzero as N goes to infinity, then eventually there will be more than one cluster. A predictive definition of independence in terms of nonzero mutual information will differ in some cases: a comparison between these candidate measures is beyond the scope of this paper.

$$1 - Pr[x_{(r,c)} \perp x_{(r',c)} | \mathbf{X}, \tilde{\lambda}_c] \approx \frac{\#(\{s|y_{(s,r)}^{sc} = y_{(s,r')}^{sc}\})}{N}$$

This measure relies on CrossCat’s detection of marginal dependencies to determine which variables are relevant in any given context. The component models largely determine how differences in each variable in that view will be weighted when calculating similarity.

2.6 Assessing Inference Quality

A central concern is that the single-site Gibbs sampler used for inference might not produce high-quality models or stable posterior estimates within practical running times. For example, the CrossCat inference algorithm might rapidly converge to a local minimum in which all proposals to create new views are rejected. In this case, even though the Gibbs sampler will appear to have converged, the models it produces could yield poor inference quality.

This section reports four experiments that illustrate key algorithmic and statistical properties of CrossCat. The first experiment gives a rough sense of inference efficiency by comparing the energies of ground truth states to the energies of states sampled from CrossCat on data generated by the model. The second experiment assesses the convergence rate and the reliability of estimates of posterior expectations on a real-world dataset. The third experiment explores CrossCat’s resistance to under-fitting and over-fitting by running inference on datasets of Gaussian noise. The fourth experiment assesses CrossCat’s predictive accuracy in a setting with a large number of distractors and a small number of signal variables. It shows that CrossCat yields favorable accuracy compared to several baseline methods.

The next experiment assesses the stability and efficiency of CrossCat inference on real-world data. Figure 5a shows the evolution of Monte Carlo dependence probability estimates as a function of the number of Markov chain iterations. Figure 5b shows traces of the number of views for each chain in the same set of runs. ~ 100 iterations appears sufficient for initializations to be forgotten, regardless of the number of views sampled from the CrossCat prior. At this point, Monte Carlo estimates appear to stabilize, and the majority of states (~ 40 of 50 total) appear to have 4, 5 or 6 views. This stability is not simply due to a local minimum: after 700 iterations, transitions that create or destroy views are still being accepted. However, the frequency of these transitions does decrease. It thus seems likely that the standard MCMC approach of averaging over a single long chain run might require significantly more computation than parallel chains. This behavior is typical for applications to real-world data. We typically use 10–100 chains, each run for 100–1,000 iterations, and have consistently obtained stable estimates.

The convergence measures from (Geweke, 1992) are also included for comparison, specifically the numerical standard error (NSE) and relative numerical efficiency (RNE) for the view CRP parameter α to assess autocorrelations (LeSage, 1999). NSE values near 0 and RNE values near 1 indicate approximately independent draws. These values were computed using a 0%, 4%, 8%, and 15% autocorrelation taper. NSE values were near zero and did not differ markedly: .023, .021, .018, and .018. Similarly, RSE values were near 1 and did not differ markedly: 1, 1.23, 1.66, and 1.54. These results suggest that there is acceptably low autocorrelation in the sampled values of the hyper-parameters.

The reliability of CrossCat reflects simple but important differences between the way single-site Gibbs sampling is used here and standard MCMC practice in machine learning. First, CrossCat

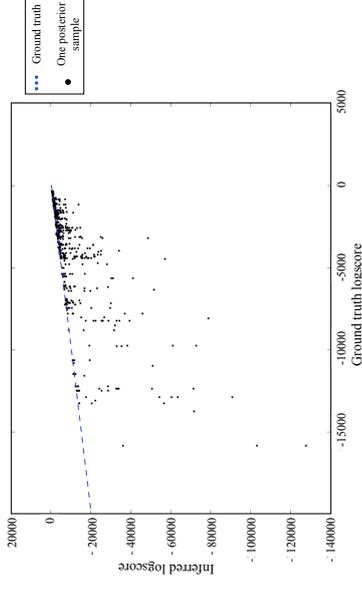


Figure 4: **The joint density of the latent cross-categorization and training data for $\sim 1,000$ samples from CrossCat’s inference algorithm, compared to ground truth.** Each point corresponds to a sample drawn from an approximation of the CrossCat posterior distribution after 200 iterations on data from a randomly chosen CrossCat model. Table sizes range from 10×10 to 512×512 . Points on the blue line correspond to samples with the same joint density as the ground truth state. Points lying above the line correspond to models that most likely underestimate the entropy of the underlying generator, i.e. they have over-fit the data. CrossCat rarely produces such samples. Some points lie significantly below the line, overestimating the entropy of the generator. These do not necessarily correspond to “under-fit” models, as the true posterior will be broad (and may also induce broad predictions) when data is scarce.

uses independent samples from parallel chains, each initialized with an independent sample from the CrossCat prior. In contrast, typical MCMC schemes from nonparametric Bayesian statistics use dependent samples obtained by thinning a single long chain that was deterministically initialized. For example, Gibbs samplers for Dirichlet process mixtures are often initialized to a state with a single cluster; this corresponds to a single-view single-category state for CrossCat. Second, CrossCat performs inference over hyper-parameters that control the expected predictability of each dimension, as well as the concentration parameters of all Dirichlet processes. Many machine learning applications of nonparametric Bayes do not include inference over these hyper-parameters; instead, they are set via cross-validation or other heuristics.

There are mechanisms by which these differences could potentially explain the surprising reliability and speed of CrossCat inference as compared to typical Gibbs samplers. Recall that the regeneration time of a Markov chain started at its equilibrium distribution is the (random) amount of time it needs to “forget” its current state and arrive at an independent sample. For CrossCat, this regeneration time appears to be substantially longer than convergence time from the prior. States from the prior are unlikely to have high energy or be near high energy regions, unlike states drawn from the posterior. Second, hyper-parameter inference — especially those controlling the expected noise in the component models, not just the Dirichlet process concentrations — provides a simple mechanism that helps the sampler exit local minima. Consider a dimension that is poorly explained

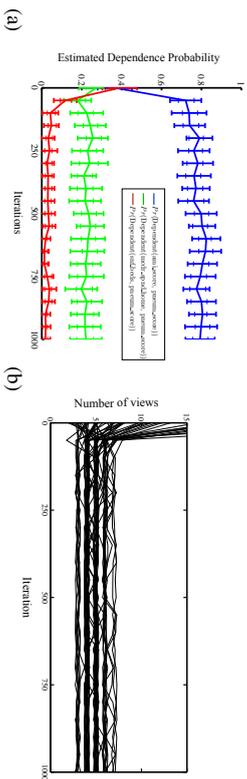


Figure 5: A quantitative assessment of the convergence rate of CrossCat inference and the stability of posterior estimates on a real-world health economics dataset. (a) shows the evolution of simple Monte Carlo estimates of the probability of dependence of three pairs of variables, made from independent chains initialized from the prior, as a function of the number of iterations of inference. Thick error bars show the standard deviation of estimates across 50 repetitions, each with 20 samples; thin lines show the standard deviation of estimates from 40 samples. Estimates stabilize after ~ 100 iterations. (b) shows the number of views for 50 of the same Markov chain runs. After ~ 100 iterations, states with 4, 5 or 6 views dominate the sample, and chains still can switch into and out of this region after 700 iterations.

by the categorization in its current view. Conditioned on such a categorization, the posterior on the hyper-parameter will favor increasing the expected noisiness of the clusters, to better accommodate the data. Once the hyper-parameter enters this regime, the model becomes less sensitive to the specific clustering used to explain this dimension. This therefore also increases the probability that the dimension will be reassigned to any other pre-existing view. It also increases the acceptance probability for proposals that create a new view with a random categorization. Once a satisfactory categorization is found, however, the Bayesian Occam’s Razor favors reducing the expected entropy of the clusters. Similar dynamics were described in (Mansinghka, Kulkarni, Perov, and Tenenbaum, 2013); a detailed study is beyond the scope of this paper.

The third simulation, shown in Figure 7, illustrates CrossCat’s behavior on datasets with low-dimensional signals amidst high-dimensional random noise. In each case, CrossCat rapidly and confidently detects the independence between the “distractor” dimensions, i.e. it does not over-fit. Also, when the signal is strong or there are few distractors, CrossCat confidently detects the true predictive relationships. As the signals become weaker, CrossCat’s confidence decreases, and variance increases. These examples qualitatively support the use of CrossCat’s estimates of dependence probabilities as indicators of the presence or absence of predictive relationships. A quantitative characterization of CrossCat’s sensitivity and specificity, as a function of both sample size and strength of dependence, is beyond the scope of this paper.

Many data analysis problems require sifting through a large pool of candidate variables in settings where only a small fraction are relevant for any given prediction. The fourth experiment, shown in Figure 7, illustrates CrossCat’s behavior in this setting. The test datasets contain 10 “signal” dimensions generated from a 5-component mixture model, plus 10–1,000 “distractor” dimensions generated by an independent 3-component mixture that clusters the data differently. As the number of distractors increases, the likelihood becomes dominated by the distractors. The experiment compares imputation accuracy for several methods — CrossCat, mixture modeling, column-wise averaging, imputation by randomly chosen values, and a popular model-free imputation tech-

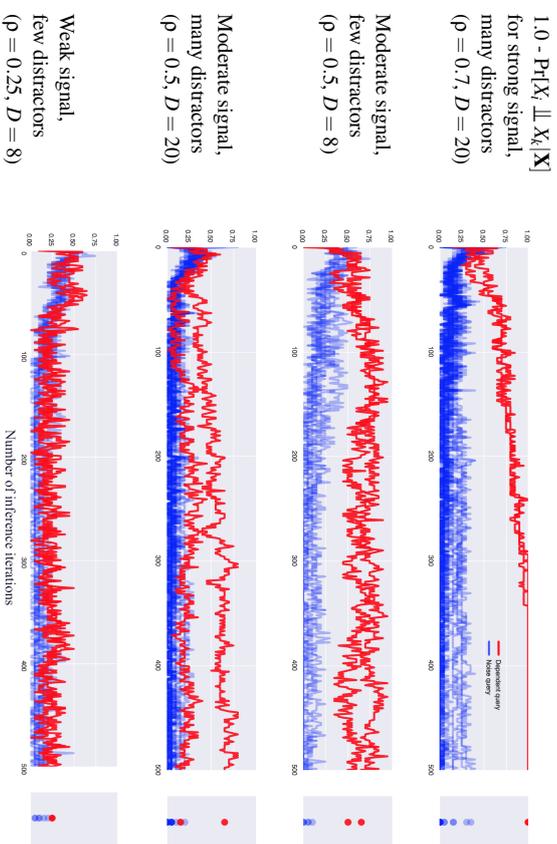


Figure 6: **Detected dependencies given two correlated signal variables and multiple independent distractors.** This experiment illustrates CrossCat’s sensitivity and specificity to pairwise relationships on multivariate Gaussian datasets with 100 rows. In each dataset, two pairs of variables have nonzero correlation ρ . The remaining $D - 4$ dimensions are uncorrelated distractors. Each row shows the inferred dependences between 20 randomly sampled pairs of distractors (blue) and the two pairs of signal variables (red). See main text for further discussion.

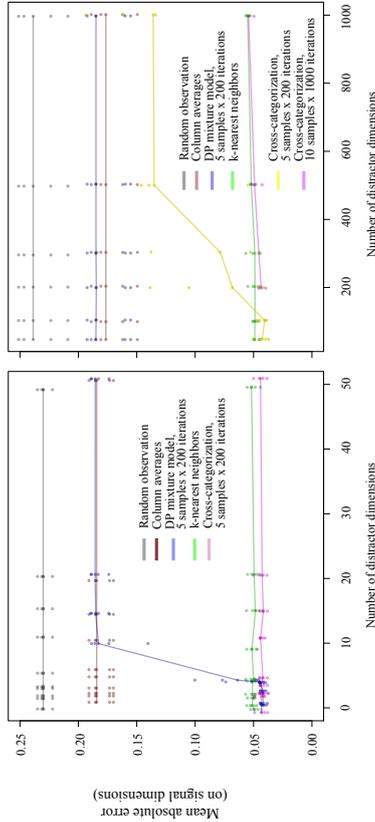


Figure 7: **Predictive accuracy for low-dimensional signals embedded in high-dimensional noise.** The data generator contains 10 “signal” dimensions described by a 5-cluster model to which distractor dimensions described by an independent 3-cluster model have been appended. The left plot shows imputation accuracies for up to 50 distractor dimensions; the right shows accuracies for 50-1,000 distractors. CrossCat is compared to mixture models as well as multiple non-probabilistic baselines (column-wise averaging, imputing via a random value, and a state-of-the-art extension to k-nearest-neighbors). The accuracy of mixture modeling drops when the number of distractors D becomes comparable to the number of signal variables S , i.e. when $D \approx S$. When $D > S$, the distractors get modeled instead of the signal. In contrast, CrossCat remains accurate when the number of distractors is 100 times larger than the number of signal variables. See main text for additional discussion.

nique (Hastie, Tibshirani, Sherlock, Eisen, Brown, and Bostein, 1999) — on problems with varying numbers of distractors. CrossCat remains accurate when the number of distractors is 100x larger than the number of signal variables. As expected, mixtures are effective in low dimensions, but inaccurate in high dimensions. When the number of distractors equals the number of signal variables, the mixture posterior grows bimodal, including one mode that treats the signal variables as noise. This mode dominates when the number of distractors increases further.

3. Empirical Results on Real-World Datasets

This section describes the results from CrossCat-based analyses of several datasets. Examples are drawn from multiple fields, including health economics, pattern recognition, political science, and econometrics. These examples involve both exploratory analysis and predictive modeling. The primary aim is to illustrate CrossCat and assess its efficacy on real-world problems. A secondary aim is to verify that CrossCat produces useful results on data generating processes with diverse statistical characteristics. A third aim is to compare CrossCat with standard generative, discriminative, and model-free methods.

3.1 Dartmouth Atlas of Health Care

The Dartmouth Atlas of Health Care (Fisher, Goodman, Wennberg, and Bronner, 2011) is one output from a long-running effort to understand the efficiency and effectiveness of the US health care system. The overall dataset covers ~4300 hospitals that can be aggregated into ~300 hospital-reporting regions. The extract analyzed here contains 74 variables that collectively describe a hospital’s capacity, quality of care, and cost structure. These variables contain information about multiple functional units of a hospital, such as the intensive care unit (ICU), its surgery department, and any hospice services it offers. For several of these units, the amount each hospital bills to a federal program called Medicare is also available. The continuous variables in this dataset range over multiple orders of magnitude. Specific examples include counts of patients, counts of beds, dollar amounts, percentages that are ratios of counts in the dataset, and numerical aggregates from survey instruments that assess quality of care.

Due to its broad coverage of hospitals and their key characteristics, this dataset illustrates some of the opportunities and challenges described by the NRC Committee on the Analysis of Massive Data (2013). For example, given the range of cost variables and quality surveys it contains, this data could be used to study the relationship between cost and quality of care. The credibility of any resulting inferences would rest partly on the comprehensiveness of the dataset in both rows (hospitals) and columns (variables). However, it can be difficult to establish the absence of meaningful predictive relationships in high-dimensional data on purely empirical grounds. Many possible sets of predictors and forms of relationships need to be considered and rejected, without sacrificing either sensitivity or specificity. If the dataset had fewer variables, a negative finding would be easier to establish, both statistically and computationally, as there are fewer possibilities to consider. However, such a negative finding would be less convincing.

The dependencies detected by CrossCat reflect accepted findings about health care that may be surprising. The inferred pairwise dependence probabilities, shown in Figure 8, depict strong evidence for a dissociation between cost and quality. Specifically, the variables in block A are aggregated quality scores, for congestive heart failure (CHE_SCORE), pneumonia (PNEUM_SCORE), acute myocardial infarction (AMI_SCORE), and an overall quality metric (QUAL_SCORE). The probability that they depend on any other variable in the dataset is low. This finding has been reported consistently across multiple studies and distinct patient populations (Fisher, Goodman, Skinner, and Bronner, 2009). Partly due to its coverage in the popular press (Gawande, 2009), it also informed the design of performance-based funding provisions in the 2009 Affordable Care Act.

CrossCat identifies several other clear, coherent blocks of variables whose dependencies are broadly consistent with common sense. For example, Section B of Figure 8 shows that CrossCat has inferred probable dependencies between three variables that all measure hospice usage. The dependencies within Section C reflect the proposition that the presence of home health aides — often consisting of expensive equipment — and overall equipment spending are probably dependent. The dark green bar for MD_CRFND_AMBLINC with the variables in section C is also intuitive: home health care easily leads to ambulance transport during emergencies. Section D shows probable dependencies between the length of hospital stays, hospital bed usage, and surgery. This section and section E, which contains measures of ICU usage, are probably predictive of the general spending metrics in section F, such as total Medicare reimbursement, use of doctors’ time, and total full time equivalent (FTE) head count. Long hospital stays, surgery, and time in the intensive care unit (ICU) are key drivers of costs, but not quality of care.

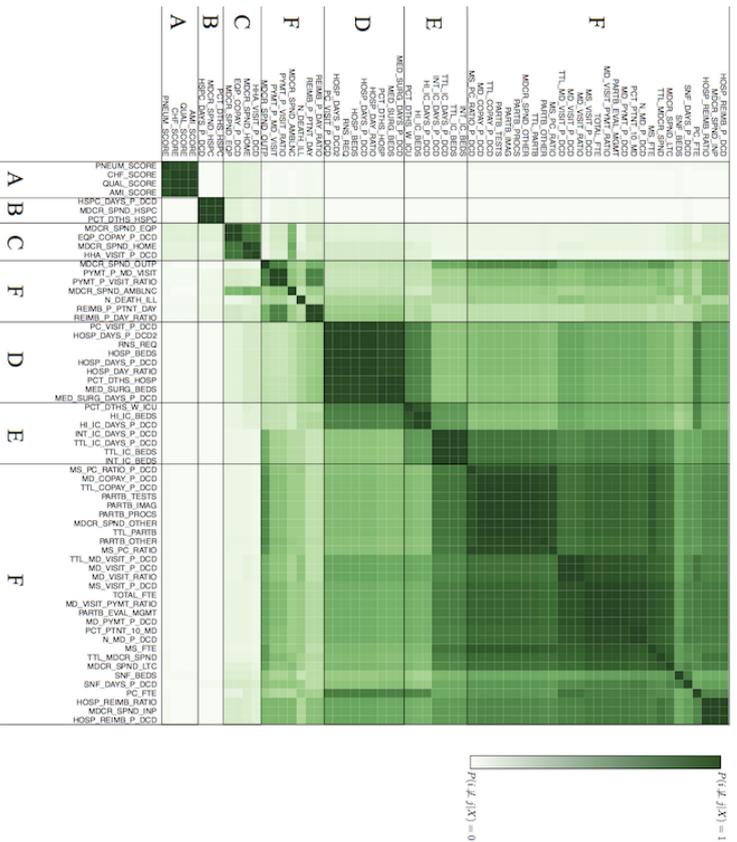


Figure 8: **Dependencies between variables the Dartmouth Atlas data aggregated by hospital federal region.** This figure shows the z-matrix $Z = [Z(i, j)]$ of pairwise dependence probabilities, where darker green represents higher probability. Rows and columns are sorted by hierarchical clustering and several portions of the matrix have been labeled. The isolation of [A] indicates that the quality score variables are almost certainly mutually dependent but independent of the variables describing capacity and cost structure. [B] contains three distinct but dependent measures of hospice cost and capacity: the percent of deaths in hospice, the number of hospice days per decedent, and the total Medicare spending on hospice usage. [C] contains spending on home health aides, equipment, and ambulance care. [D] shows dependencies between hospital stays, surgeries, and in-hospital deaths. [E] contains variables characterizing intensive care, including some that probably interact with surgery, and others that interact with general spending metrics [F], such as usage of doctors' time and total full time equivalency (FTE) head count.

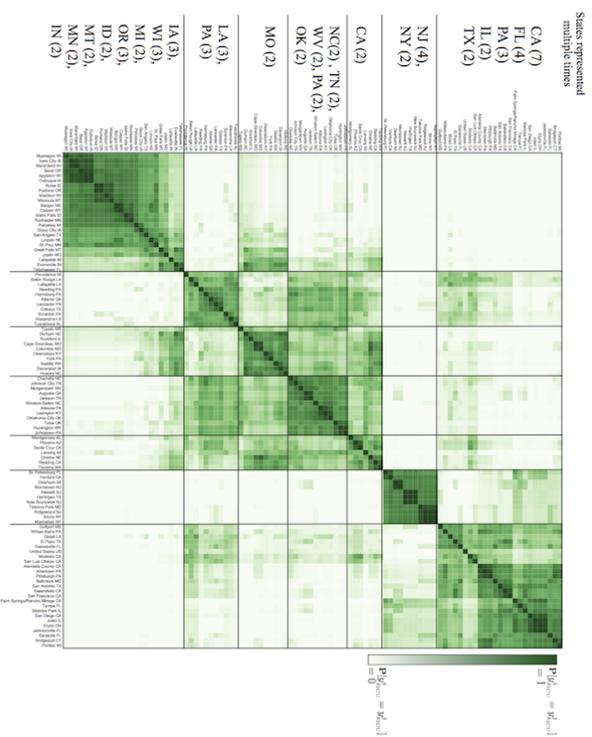


Figure 9: **The pairwise similarity measure inferred by CrossCat in the context of ICU utilization.** Each cell contains an estimate of the marginal probability that the hospital reporting regions corresponding to the row and column come from the same category in the view. The block structure in this matrix reflects regional variation in ICU utilization and in other variables that are probably predictive of it; examples include measures of hospital and intensive care capacity and usage.

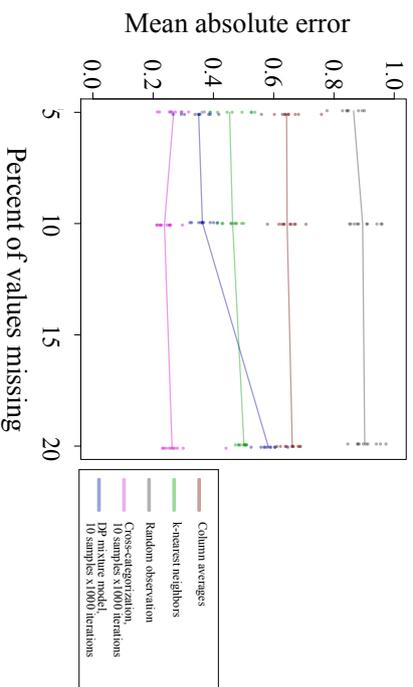


Figure 12: **A comparison of imputation accuracy under random censoring.** The error (y axis) as a function of the fraction of missing values (x axis) is measured on a scale that has been normalized by column-wise variance, so that high-variance variables do not dominate the comparison. CrossCat is more accurate than baselines such as column-wise averaging, imputation using a randomly chosen observation, a state-of-the-art variant of k-nearest-neighbors, and Dirichlet process mixtures of Gaussians. Also note the collapse of mixture modeling to column-wise averaging when the fraction of missing values grows sufficiently large.

3.2 Classifying Images of Handwritten Digits

The MNIST collection of handwritten digit images (LeCun and Cortes, 2011) can be used to explore CrossCat’s applicability to high-dimensional prediction problems from pattern recognition. Figure 13a shows example digits. For all experiments, each image was downsampled to 16x16 pixels and represented as a 256-dimensional binary vector. The digit label was treated as an additional categorical variable, observed for training examples and treated as missing for testing. Figure 13b shows the inferred dependence probabilities among pixels and between the digit label and the pixels. The pixels that are identified as independent of the digit class label lie on the boundary of the image, as shown in Figure 13c.

A set of approximate posterior samples from CrossCat can be used to complete partially observed images by sampling predictions for arbitrary subsets of pixels. Figure 14 illustrates this: each panel shows the data, marginal predictive images, and predicted image completions, for 10 images from the dataset, one per digit. With no data, all 10 predictive distributions are equivalent, but as additional pixels are observed, the predictions for most images concentrate on representations of the correct digit. Some digits remain ambiguous when $\sim 30\%$ of the pixels have been observed. The predictive distributions begin as highly multi-modal distributions when there is little to no data, but concentrate on roughly unimodal distributions given sufficiently many features.

The predictive distribution can also be used to infer the most probable digit, i.e. solve the standard MNIST multi-class classification problem. Figure 15 shows ROC curves for CrossCat on this problem. Each panel shows the tradeoff between true and false positives for each digit, aggregated from the overall performance on the underlying multi-class problem. The figure also includes ROC curves for Support Vector Machines with linear and Gaussian kernels. For these methods, the standard one-vs-all approach was used to reduce the multi-class problem into a set of binary classification problems. The regularization and kernel bandwidth parameters for the SVMs were set via cross-validation using 10% of the training data. 10 posterior samples from CrossCat were used, each obtained after 1,000 iterations of inference from a random initialization. CrossCat was more accurate than the linear discriminative technique; this is expected, as CrossCat induces a nonlinear decision boundary even if classifying based on a single posterior sample. Overall, the 10-sample model used here made less accurate predictions than the Gaussian SVM baseline. Also, in anecdotal runs that were scored by overall 0-1 loss rather than per-digit accuracy, performance was similarly mixed, and less favorable for CrossCat. However, the size of the kernel matrix for the Gaussian SVM scales quadratically, while CrossCat scales linearly. As a classifier, CrossCat thus offers different tradeoffs between accuracy, amount of training data, test-time parallelism (via the number of independent samples), and latency than standard techniques.

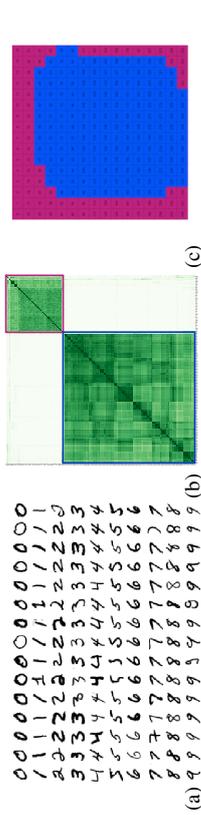


Figure 13: **MNIST handwritten digits, feature z-matrix, and color-coded image pixel locations.** (a) Fifteen visually rendered examples of handwritten digits for each number in the MNIST data set. Each image was converted to a binary feature vector for predictive modeling. CrossCat additionally treated the digit label as an additional feature; this value was observed for training examples and treated as missing for testing. (b) The dependence probabilities between pixel values distinguish two blocks of pixels, one containing the digit label. (c) Coloring the pixels from each block reveals the spatial structure in pixel dependencies. Blue pixels — pixels from the block with a blue border from figure (b) — pick out the foreground, i.e. pixels whose values depend on what digit the image contains. Magenta pixels pick out the common background, i.e. pixels whose values are independent of what digit is drawn.

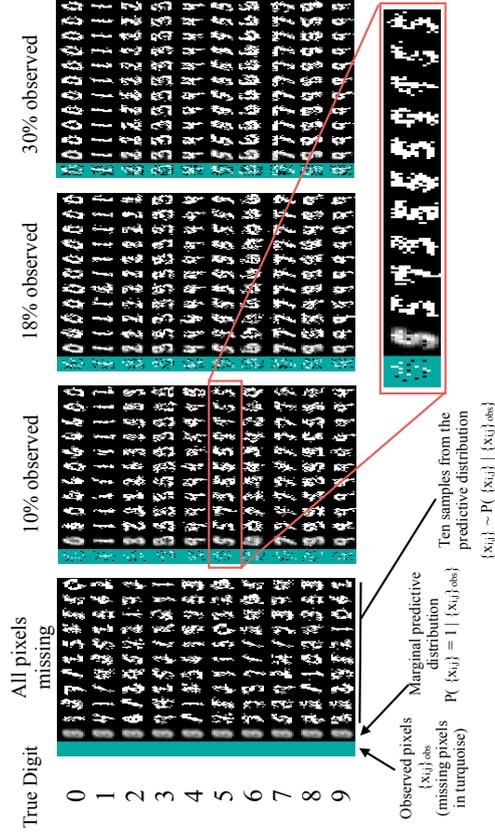


Figure 14: **Predicted images of handwritten digits given sparse observations.** CrossCat can be used to fill in missing pixels by sampling from their conditional density given the observed pixels. Each panel shows completion results for one image per digit; across panels, the fraction of observed pixels grows from 0 to 30%. The leftmost column shows the observed pixels in black and white, with missing pixels in turquoise. The second column from the left shows the marginal probabilities for each pixel. The 10 remaining columns show independent sampled predictions. In the leftmost panel, with no data, all 10 predictive distributions are equivalent. The predictive distribution collapses onto single digit classes (except for 8 and 6) after 18% of the digits have been observed. The marginal images for 1, 7, and 9 become resolvable to a single prototypical example after 10% of the 256 pixels have been observed. Others, such as 8, remain ambiguous.

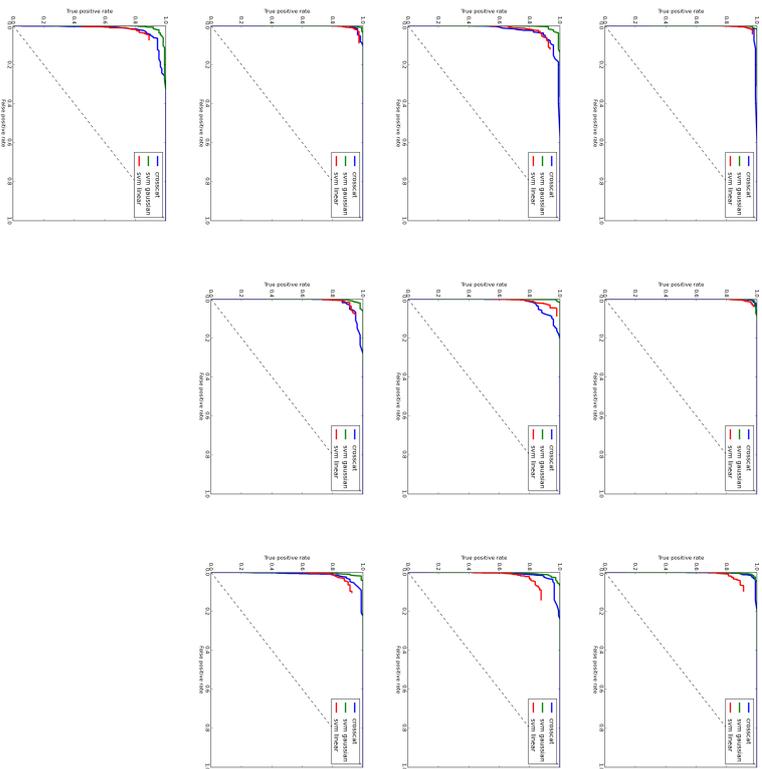


Figure 15: **Classification accuracy on handwritten digits from MNIST.** Each panel shows the true positive/false positive tradeoff curves for classifying each digit from 0 through 9. Digit images were represented as binary vectors, with one dimension per pixel. As with the image completion example from Figure 14, CrossCat was applied directly to this data, with the digit label appended as a categorical variable; no weighting or tuning for the supervised setting was done. Support vector machines (SVMs) with both linear and Gaussian kernels are provided as baselines. Regularization and kernel bandwidth parameters were chosen via cross-validation on 10% of the training data, with multiple classes treated via a one-versus-all reduction. See the main text for further discussion.

3.3 Voting Records for the 111th Senate

Voters are often members of multiple issue-dependent coalitions. For example, US senators sometimes vote according to party lines, and at other times vote according to regional interests. Because this common-sense structure is typical for the CrossCat prior, voting records are an important test case.

This set of experiments describes the results of a CrossCat analysis of the 397 votes held by the 111th Senate during the 2009-2010 session. In this dataset, each column is a vote or bill, and each row is a senator. Figure 16 shows the raw voting data, with several votes and senators highlighted. There are 106 senators; this is senators is larger than the size of the senate by 6, due to deaths, replacement appointments, party switches, and special elections. When a senator did not vote on a given issue, that datum is treated as missing. Figure 16 also includes two posterior samples, one that reflects partisan alignment and another that posits a higher-resolution model for the votes.

This kind of structure is also apparent in estimates that aggregate across samples. Dependence probabilities between votes are shown in Figure 17. The visible independencies between blocks are compatible with a common-sense understanding of US politics. The two votes in orange are partisan issues. The two votes in green have broad bipartisan support. The vote in yellow aimed at removing an energy subsidy for rural areas, an issue that cross-cuts party lines. The vote in purple stipulates that the Department of Homeland Security must spend its funding through competitive processes, with an exception for small businesses and women or minority-owned businesses. This issue subdivides the republican party, isolating many of the most fiscally conservative. Similarity matrices for the senators with respect to S. 160 (orange) and an amendment to H.R. 2997 (yellow) are shown in Figure 18, with the senators whose similarity values changed the most between these two bills highlighted in grey.

It is instructive to compare the latent structure and predictions inferred by CrossCat with structures and predictions from other learning techniques. As an individual voting record can be described by 397 binary variables and the missing values are negligible, Bayesian network structure learning is a suitable benchmark. Figure 19a shows the best Bayesian network structure found by structure learning using the search-and-score method implemented in the Bayes Net Toolbox for MATLAB (Murphy, 2001). This search is based on local moves similar to the transition operators from Gaudici and Green (1999). The highest scoring graphs after 500 iterations contained between 143 and 193 links. Figure 19b shows the marginal dependencies between votes induced by this Bayesian network; these are sparser than those from CrossCat. Figure 19c shows the mean absolute errors for this Bayes net and for CrossCat on a predictive test where 25% of the votes were held out and predicted for senators in the test set. Bayes net CPTs were estimated using a symmetric Beta-Bernoulli model with hyper-parameter $\alpha = 1$. CrossCat predictions were based on four samples, each obtained after 250 iterations of inference. Compared to Bayesian network structure learning, CrossCat makes more accurate predictions and also finds more intuitive latent structures.

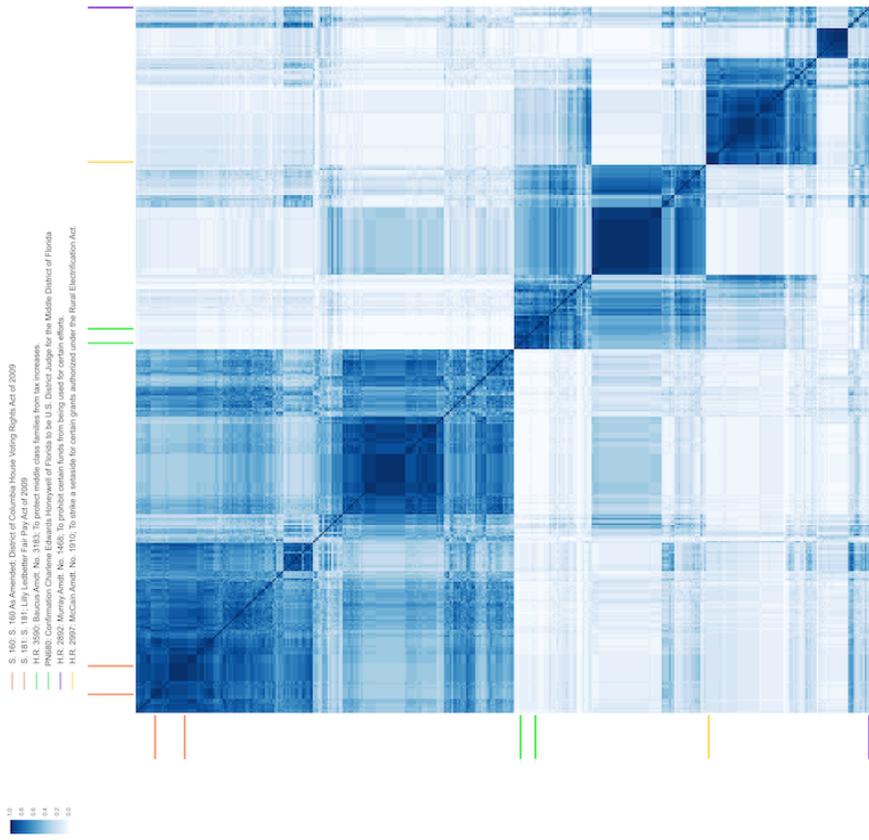


Figure 17: **Pairwise dependence probabilities between bills.** Blocks of bills with high probability of dependence include predominantly partisan issues (orange), issues with broad bipartisan support (green), and bills that divide senators along ideological or regional lines.

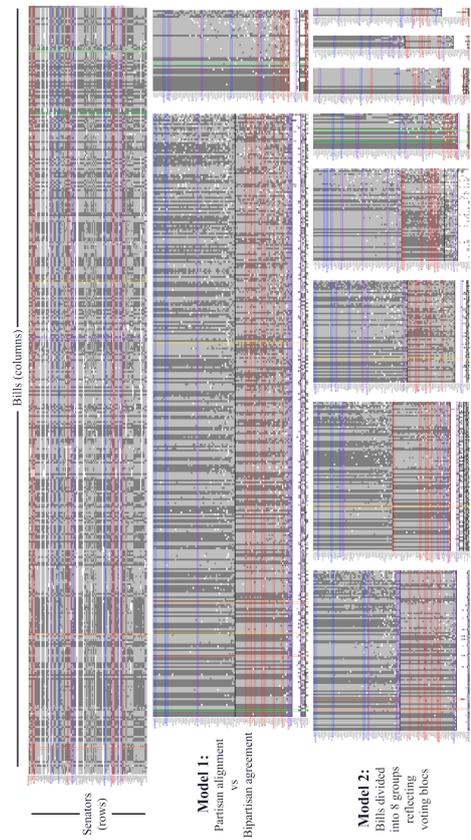


Figure 16: **Voting records for the 111th US Senate (2009).** (top) This includes 397 votes (yea in light grey, nay in dark grey) for 106 senators, including separate records for senators who changed parties or assumed other offices mid-term. Some senators are highlighted in colors based on their generally accepted identification as democrats (blue), moderates (purple), or republicans (red). See main text for an explanation of the colored bills. (middle) This row shows a simple or low resolution posterior sample that divides bills into those that exhibit partisan alignment and those with bipartisan agreement. Clusters of senators, generated automatically, are separated by thick black horizontal lines. (bottom) This row shows a sample that includes additional views and clusters, positing a finer-grained predictive model for votes that are treated as random in the middle row.

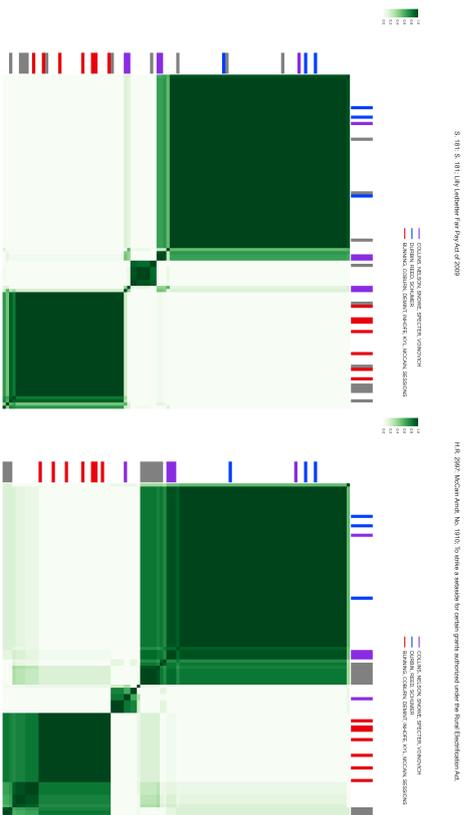


Figure 18: **Context-sensitive similarity measures for senators with respect to partisan and special-interest issues.** The left matrix shows senator similarity for S. 181, the Lilly Ledbetter Fair Pay Act of 2009, a bill whose senator clusters tend to respect party lines. The right matrix is for H.R. 2997, a bill designed to remove a subsidy for energy generation systems in rural areas. The grey senators are those whose similarities changed the most between these two bills.

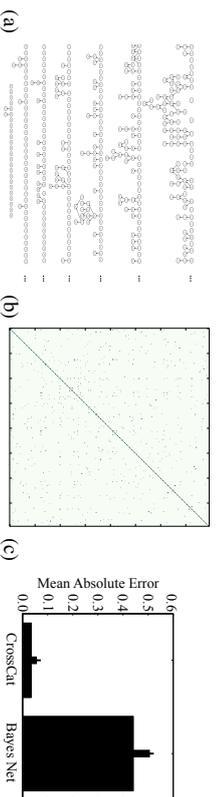


Figure 19: **Comparison of latent structures and predictive accuracy for CrossCat and Bayesian network structure learning.** (a) The Bayesian network structure found by structure learning; each node is a vote, and edge indicates a conditional dependence. (b) The sparse marginal dependencies induced by this Bayes net. (c) A comparison of the predictive accuracy of CrossCat and Bayesian networks. See main text for details and discussion.

High-throughput measurement techniques in modern biology generate datasets of unusually high dimension. The number of variables typically is far greater than the sample size. For example, microarrays can be used to assess the expression levels of 10,000 to 100,000 probe sequences, but due to the cost of clinical data acquisition, typical datasets may have just tens or hundreds of tissue samples. Exploration and analysis of this data can be both statistically and computationally challenging.

3.4 High-Dimensional Gene Expression Data

Individual samples from CrossCat can aid exploration of this kind of data. Figure 20 shows one typical sample obtained from the data in Raponi et al. (2009). The dataset had ~ 1 million cells, with 10,000 probes (columns) and 100 tissue samples (rows). This sample has multiple visually coherent views with ~ 50 genes, each reflecting a particular low-dimensional pattern. Some of these views divide the rows into clusters with “low”, “medium” or “high” expression levels; others reflect more complex patterns. The co-assignment of many probes to a single view could indicate the existence of a latent co-regulating mechanism. The structure in a single sample could thus potentially inform pathway searches and help generate testable hypotheses.

Posterior estimates from CrossCat can also facilitate exploration. CrossCat was applied to estimate dependence probabilities for a subset of the arthritis dataset from (Bienkowska, Dalgin, Battiwalla, Allaire, Roubenoff, Gregersen, and Carulli, 2009) (NCBI GEO accession number GSE15258). This dataset contains expression levels for $\sim 55,000$ probes, each measured for 87 patients. It also contains standard measures of initial and final disease levels and a categorical “response” variable with 3 classes. CrossCat was applied to subsets with 1,000 and 5,000 columns.

Figure 20 shows the 100 variables most probably predictive of a 3-class treatment response variable. The dependence probabilities with response (outlined in red) are all low, i.e. according to CrossCat, there is little evidence in favor of the existence of any prognostic biomarker. At first glance this may seem to contradict (Bienkowska et al., 2009), which reports 8-gene and 24-gene biomarkers with prognostic accuracies of 83%-91%. However, the test set from (Bienkowska et al., 2009) has 11 out-of-sample patients, 9 of whom are responders. Predicting according to class marginal probabilities would yield compatible accuracy. The final disease activation level, outlined in blue, does appear within the selected set of variables. CrossCat infers that it probably depends on many other gene expression levels; these genes could potentially reflect the progression of arthritis in physiologically or clinically useful ways.

3.5 Longitudinal Unemployment Data by State

This experiment explores CrossCat's behavior on data where variables are tracked over time. The data are monthly state-level unemployment rates from 1976 to 2011, without seasonal adjustment, obtained from the US Bureau of Labor Statistics. The data also includes annual unemployment rates for every state. Figure 21 shows time series for 5 states, along with national unemployment rate and real Gross Domestic Product (GDP). Typical analyses of raw macroeconomic time series are built on assumptions about temporal dependence and incorporate multiple model-based smoothing techniques; see e.g. (Bureau of Labor Statistics, 2014). Cyclical macroeconomic dynamics, such as the business cycle, are demarcated using additional formal and informal techniques for assessing agreement across multiple indicators (National Bureau of Economic Research, 2010). For this analysis, the input data was organized as a table, where each row r represents a state, each column c represents a month, and each cell $x_{r,c}$ represents the unemployment rate for state r in month c . This representation removes all temporal cues.

CrossCat posits short-range, common-sense temporal structure in state-level employment rates. The top panel of Figure 21 shows the largest and smallest month in each of the views in a single posterior sample as vertical dashes. Figure 22a shows the frequency of years in each view; each view contains one or two temporally contiguous blocks. Figure 22b shows the raw unemployment rates sorted according to the cross-categorization from this sample. Different groups of states are affected by each phase of the business cycle in different ways, inducing different natural clusterings of unemployment rates.

CrossCat also detects long-range temporal structure that is largely in agreement with the officially designated phases of the business cycle. Figure 23 shows the dependence probabilities for all months, in temporal order, with business cycle peaks in black and troughs in red. The beginning of the 1980 recession aligns closely with a sharp drop in dependence probability; this indicates that during 1980 the states naturally cluster differently than they do in 1979. Three major US recessions — 1980, 1990, and late 2001 — align with these breakpoints. The beginning of the 2008 recession and the end of the 1980s recession (in 1984) both fall near sub-block boundaries; these are best seen at high resolution. Correspondence is not perfect, but this is expected: CrossCat is not analyzing the same data used to determine business cycle peaks and troughs, nor is it explicitly assuming any temporal dynamics.

Time series analysis techniques commonly assume temporal smoothness and sometimes also incorporate the possibility of abrupt changes (Ahn and Thompson, 1988; Wang and Zivot, 2000). CrossCat provides an alternative approach that makes weaker dynamical assumptions: temporal smoothness is not assumed at the outset but must be inferred from the data. This cross-sectional approach to the analysis of natively longitudinal data may open up new possibilities in econometrics. For example, it could be fruitful to apply CrossCat to a longitudinal dataset with multiple macroeconomic variables for each state, or to use CrossCat to combine temporal information at different timescales.

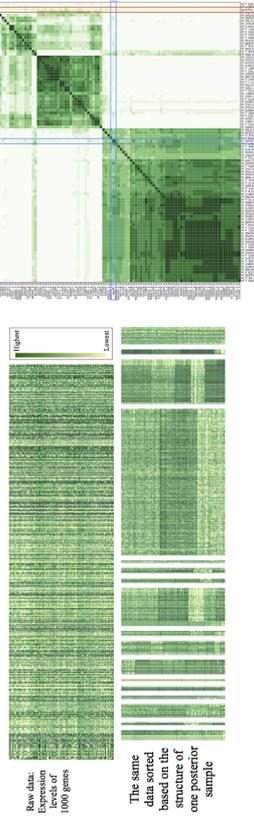


Figure 20: **CrossCat inference results on gene expression data.** (left) Expression levels for the top 1,000 highest-variance genes from Raponi et al. (2009) in original form and in the order induced by a single CrossCat sample. CrossCat was applied to a subset with roughly 1 million cells ($\sim 10,000$ probes by ~ 100 tissue samples). (right) Pairwise dependence probabilities between probe values and treatment response inferred from the GSE15258 dataset. The 100 probes most probably dependent on a 3-class treatment response variable, based on analysis of a subset with 1,000 probes, are shown outlined in red. The low inferred dependence probability suggests that the data does not support the existence of any prognostic biomarker. A standard disease activity score is shown outlined in blue; this measure is naturally dependent on many of the probes.

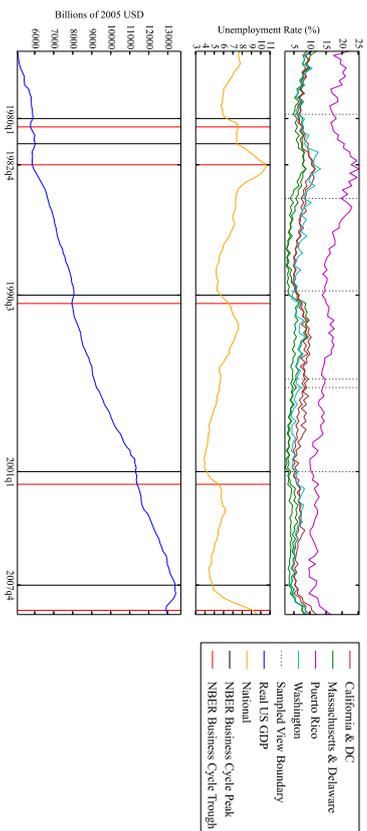


Figure 21: **US state-level unemployment data aligned with official business cycle peaks and troughs.** (top) Unemployment rates for five states from 1976 to 2009, colored according to one posterior sample. (middle) The national unemployment rate and business cycle peaks and troughs during the same period. Business cycle peaks and troughs are identified using multiple macroeconomic signals; see main text for details. Unemployment grows during recessions (intervals bounded on the left by black and on the right by red) and shrinks during periods of growth (intervals bounded on the left by red and on the right by black). (bottom) Real US gross domestic product (GDP) similarly decreases or stays constant during recessions and increases during periods of growth. View boundaries from the CrossCat sample pick out the business cycle turning points around 1980, 1990 and 2001.

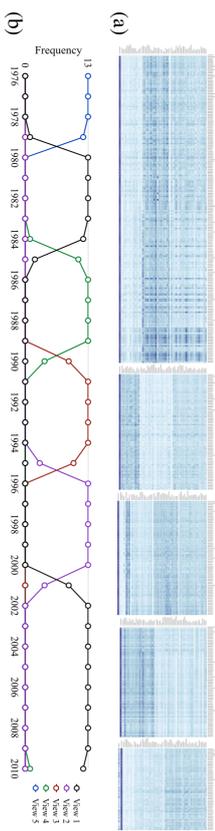


Figure 22: **The temporal structure in a single posterior sample.** (a) The complete state-level monthly employment rate dataset, sorted according to one posterior sample. This sample divides the months into 5 time periods, each inducing a different clustering of the states. (b) The frequency of years and quarters for each view shows that each view reflects temporal contiguity: each view either corresponds to a single, temporally contiguous interval or to the union of two such intervals. This temporal structure is not given to CrossCat, but rather inferred from patterns of unemployment across clusters of states.

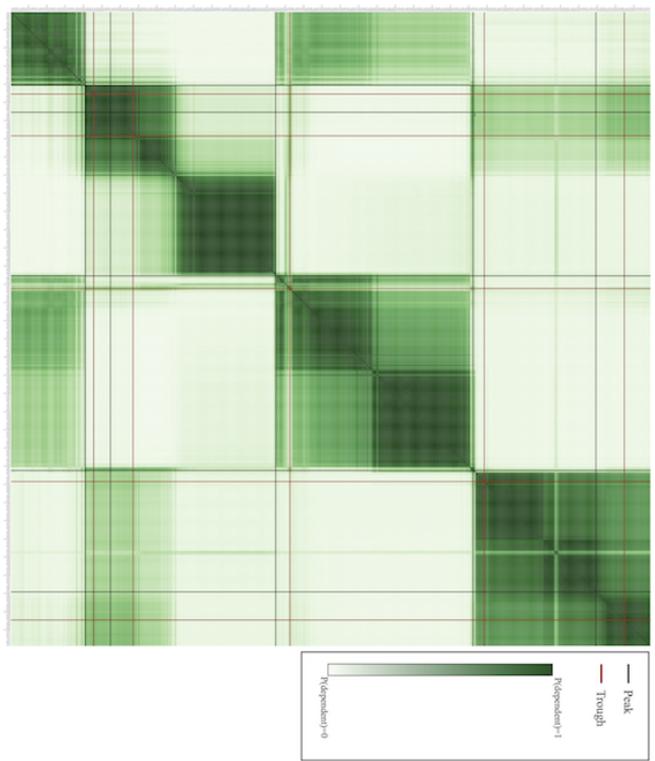


Figure 23: **Dependence probabilities between monthly unemployment rates.** Unemployment rates are sorted by date, beginning with 1976 in the bottom left corner, with business cycle peaks and troughs identified by the NBER in black and red. The beginnings of three major recessions — in the early 1980s, 1990s, and late 2001 — are identified by breaks in dependence probability: unemployment rates are dependent with high probability within these periods, and independent of rates from the previous period. See main text for more discussion.

4. Discussion

This paper contains two contributions. First, it describes CrossCat, a model and inference algorithm that together comprise a domain-general method for characterizing the full joint distribution of the variables in a high-dimensional dataset. CrossCat makes it possible to draw a broad class of Bayesian inferences and to solve prediction problems without domain-specific modeling. Second, it describes applications to real-world datasets and analysis problems from multiple fields. CrossCat finds latent structures that are consistent with accepted findings as well as common-sense knowledge and that can yield favorable predictive accuracy compared to generative and discriminative baselines.

CrossCat is expressive enough to recover several standard statistical methods by fixing its hyperparameters or adding other deterministic constraints:

1. **Semi-supervised naive Bayes:** $\alpha = 0 \implies \#(\text{unique}(\bar{z})) = 1$ and $\alpha_0 = \epsilon$ with $\lambda_{\text{class}} = 0$

Assume that the dimension in the dataset labeled *class* contains categorical class labels, with a multinomial component model and symmetric Dirichlet prior, with concentration parameter λ_{class} . Because the outer CRP concentration parameter is 0, all features as well as the class label will be assigned to a single view. Because $\lambda_{\text{class}} = 0$, each category in this view will have a component model for the class label dimension that constrains the category to only contain data items whose class labels all agree:

$$y_i^j = y_i^j \iff x_{(i,\text{class})} = x_{(j,\text{class})}$$

This yields a model that is appropriate for semi-supervised classification, and related to previously proposed techniques based on EM for mixture models (Nigam, McCallum, Thrun, and Mitchell, 1999). Data from each class will be modeled by a separate set of clusters, with feature hyper-parameters (e.g. overall scales for continuous values, or levels of noise for discrete values) shared across classes. Data items whose class labels are missing will be stochastically assigned to classes based on how compatible their features are with the features for other data items in the same class, marginalizing out parameter uncertainty. Forcing the concentration parameter α_0 for the sole inner CRP to have a sufficiently small value ϵ ensures that there will only be a single cluster per class (with arbitrarily high probability depending on N and ϵ). These restrictions thus recover a version of the naive Bayesian classifier (for discrete data) and linear discriminant analysis (for continuous data), adding hyper-parameter inference.

2. **Nonparametric mixture modeling:** $\alpha = 0 \implies \#(\text{unique}(\bar{z})) = 1$

If the constraints on categorizations are relaxed, but the outer CRP is still constrained to generate a single view, then CrossCat recovers a standard nonparametric Bayesian mixture model. The current formulation of CrossCat additionally enforces independence between features. This assumption is standard for mixtures over high-dimensional discrete data. Mixtures of high-dimensional continuous distributions sometimes support dependence between variables within each component, rather than model all dependence using the latent component assignments. It would be easy and natural to relax CrossCat to support these component models and to revise the calculations of marginal dependence accordingly.

3. **Independent univariate mixtures:** $\alpha = \infty \implies \#(\text{unique}(\bar{z})) = D$

The outer CRP can be forced to assign each variable to a separate view by setting its concentration parameter α to ∞ . With this setting, each customer (variable) will choose a new table (view) with probability 1. In this configuration, CrossCat reduces to a set of independent Dirichlet process mixture models, one per variable. A complex dataset with absolutely no dependencies between variables can induce a CrossCat posterior that concentrates near this subspace.

4. **Clustering with unsupervised feature selection:** $\#(\text{unique}(\bar{z})) = 2$, with $\alpha_0 > 0$ but $\alpha_1 = 0$
- A standard mixture must model noisy or independent variables using the same cluster assignments as the variables that support the clustering. It can therefore be useful to integrate mixture modeling with feature selection, by permitting inference to select variables that should be modeled independently. The “irrelevant” features can be modeled in multiple ways; one natural approach is to use a single parametric model that can independently adjust the entropy of its model for each dimension. CrossCat contains this extension to mixtures as a subspace.

The empirical results suggest that CrossCat’s flexibility in principle manifests in practice. The experiments show that can effectively emulate many qualitatively different data generating processes, including processes with varying degrees of determinism and diverse dependence structures. However, it will still be important to quantitatively characterize CrossCat’s accuracy as a density estimator.

Accuracy assessments will be difficult for two main reasons. First, it is not clear how to define a space of data generators that spans a sufficiently broad class of applied statistics problems. CrossCat itself could be used as a starting point, but key statistical properties such as the marginal and conditional entropies of groups of variables are only implicitly controllable. Second, it is not clear how to measure the quality of an emulator for the full joint distribution. Metrics from collaborative filtering and imputation, such as the mean squared error on randomly censored cells, do not account for predictive uncertainty. Also, the accuracy of estimates of joint distributions and conditional distributions can diverge. Thus the natural metric choice of KL divergence between the emulated and true joint distributions may be misleading in applications where CrossCat is used to respond to a stream of queries of different structures. Because there are exponentially many possible query structures, random sampling will most likely be needed. Modeling the likely query sequences or weighting queries based on their importance seems ultimately necessary but difficult.

In addition to these questions, CrossCat has several known limitations that could be addressed by additional research:

1. *Real-world datasets may contain types and/or shapes of data that CrossCat can only handle by transformation.*

First, several common data types are poorly modeled by the set of component models that CrossCat currently supports. Examples include timestamps, geographical locations, currency values, and categorical variables drawn from open sets. Additional parametric component models — or nonparametric models, e.g. for open sets of discrete values — could be integrated.

Second, it is unclear how to best handle time series data or panel/longitudinal settings. In the analysis of state-level monthly unemployment data, each state was represented as a row, and

each time point was a separate and a priori independent column. The authors were surprised that CrossCat inferred the temporal structure rather than under-fit by ignoring it. In retrospect, this is to be expected in circumstances where the temporal signal is sufficiently strong, such that it can be recovered by inference over the views. However, computational and statistical limitations seem likely to lead to under-fitting on panel data with sufficiently many time points and variables per time point.

One pragmatic approach to resolving this issue is to transform panel data into cross-sectional data by replacing the time series for each member of the population with the parameters of a time-series model. Separately fitting the time-series model could be done as a pre-processing step, or alternated with CrossCat inference to yield a joint Gibbs sampler. Another approach would be to develop a sequential extension to CrossCat. For example, the inner Dirichlet process mixtures could be replaced with nonparametric state machines (Beal, Ghahramani, and Rasmussen, 2001). Each view would share a common state machine, with common states, transition models, and observation models. Each group of dependent variables would thus induce a division of the data into subpopulations, each with a distinct hidden state sequence.

2. Discriminative learning can be more accurate than CrossCat on standard classification and regression problems.

Discriminative techniques can deliver higher predictive accuracy than CrossCat when input features are fully observed during both training and testing and when there is enough labeled training data. One possible remedy is to integrate CrossCat with discriminative techniques, e.g. by allowing “discriminative target” variables to be modeled by generic regressions (e.g. GLMs or Gaussian processes). These regressions would be conditioned on the non-discriminative variables that would still be modeled by CrossCat.

An alternative approach is to distinguish prediction targets within the CrossCat model. At present, the CrossCat likelihood penalizes prediction errors in all features equally. This could be fixed by e.g. deterministically constraining the Dirichlet concentration hyper-parameters for class-label columns to be equal to 0. This forces CrossCat to assign 0 probability density to states that put items from different classes into the same category in the view containing the class label. These purity constraints can be met by using categories that either exactly correspond to the finest-grained classes in the dataset or subdivide these classes. Conditioned on the hyper-parameters, this modification reduces joint density estimation to independent nonparametric Bayesian estimation of class-conditional densities (Mansinghka, Roy, Rifkin, and Tenenbaum, 2007).

3. Natural variations are challenging to test due to the cost and difficulty of developing fast implementations.

The authors found it surprising that a reliable and scalable implementation was possible. Several authors were involved in the engineering of multiple high-performance commercial implementations. One of these can be applied to multiple real-world, million-row datasets with typical runtimes ranging from minutes to hours (Obermeyer, Glidden, and Ionsa, 2014). No fundamental changes to the Gibbs sampling algorithm were necessary to make it possible to do to inference on these scales. Instead, the gains were due to standard software performance engineering techniques. Examples include custom numerical libraries, careful data structure design, and adopting a streaming architecture and compact latent state representation that

reduce the time spent waiting for memory retrieval. The simplicity of the Gibbs sampling algorithm thus turned out to be an asset for achieving high performance. Unfortunately, this implementation took man-years of software engineering, and is harder to extend and modify than slower, research-oriented implementations.

Probabilistic programming technology could potentially simplify the process of prototyping variations on CrossCat and incrementally optimizing them. For example, Venture (Mansinghka et al., 2014) can express the CrossCat model and inference algorithm from this paper in ~ 40 lines of probabilistic code. At the time of writing, the primary open-source implementation of CrossCat is ~ 4000 lines of C++. New datatypes, model variations, and perhaps even more sophisticated inference strategies could potentially be tested this way. However, the performance engineering will still be difficult. As an alternative to careful performance engineering, the authors experimented with more sophisticated algorithms and initializations. Transition operators such as the split-merge algorithm from (Jain and Neal, 2000) and initialization schemes based on high-quality approximations to Dirichlet process posteriors (Li and Shafiq, 2011) did not appear to help significantly. These complex approaches are also more difficult to debug and optimize than single-site Gibbs. This may be a general feature: reductions in the total number of iterations can easily be offset by increases in the computation time required for each transition.

There is a widespread need for statistical methods that are effective in high dimensions but do not rely on restrictive or opaque assumptions (NRC Committee on the Analysis of Massive Data, 2013; Wasserman, 2011). CrossCat attempts to address these requirements via a divide-and-conquer strategy. Each high-dimensional modeling problem is decomposed into multiple independent sub-problems, each of lower dimension. Each of these subproblems is itself decomposed by splitting the data into discrete categories that are separately modeled using parametric Bayesian techniques. The hypothesis space induced by these stochastic decompositions contains proxies for a broad class of data generators, including some generators that are simple and others that are complex. The transparency of simple parametric models is largely preserved, without sacrificing modeling flexibility. It may be possible to design other statistical models around this algorithmic motif.

CrossCat formulates a broad class of supervised, semi-supervised, and unsupervised learning problems in terms of a single set of models and a single pair of algorithms for learning and prediction. The set of models and queries that can be implemented may be large enough to sustain a dedicated probabilistic programming language. Probabilistic programs in such a language could contain modeling constraints, translated into hyper-parameter settings, but leave the remaining modeling details to be filled in via approximate Bayesian inference. Data exploration using CrossCat samples can be cumbersome, and would be simplified by a query language where each query could reference previous results.

This flexibility comes with costs, especially in applications where only a single repeated prediction problem is important. In these cases, it can be more effective to use a statistical procedure that is optimized for this task, such as a discriminative learning algorithm. It seems unlikely that even a highly optimized CrossCat implementation will be able to match the performance of best-in-class supervised learning algorithms when data is plentiful and all features are fully observed.

However, just as with software, sophisticated optimizations also come with costs, and can be premature. For example, some researchers have suggested that there is an “illusion of progress” in classifier technology (Hand, 2006) in which algorithmic and statistical improvements documented

in classification research papers frequently do not hold up in practice. Instead, classic methods seem to give the best and most robust performance. One interpretation is that this is the result of prematurely optimizing based on particular notions of expected statistical accuracy. The choice to formalize a problem as supervised classification may similarly be premature. It is not uncommon for the desired prediction targets to change after deployment, or for the typical patterns of missing values to shift. In both these cases, a collection of CrossCat samples can be used unmodified, while supervised methods need to be retrained.

It is unclear how far this direct Bayesian approach to data analysis can be taken, or how broad is the class of data generating processes that CrossCat can emulate in practice. Some statistical inference problems may be difficult to pose in terms of approximately Bayesian reasoning over a space of proxy generators. Under-fitting may be difficult to avoid, especially for problems with complex couplings between variables that exceed the statistical capacity of fully factored models. Despite these challenges, our experiences with CrossCat have been encouraging. It is fortunate that, paraphrasing Box (1979), the statistical models that CrossCat produces can be simplistic yet still flexible and useful. We thus hope that CrossCat proves to be an effective tool for the analysis of high-dimensional data. We also hope that the results in this paper will encourage the design of other fully Bayesian, general-purpose statistical methods.

Acknowledgments

The authors thank Kevin Murphy, Ryan Rifkin, Cameron Freer, Daniel Roy, Bill Lazarus, David Jensen, Beau Cronin, Rax Dillon, and the anonymous reviewers and editor for valuable suggestions. This work was supported in part by gifts from Google, NTT Communication Sciences Laboratory, and Eli Lilly & Co; by the Army Research Office under agreement number W911NF-13-1-0212; and by DARPA via the XDATA and PPAML programs. Any opinions, findings, and conclusions or recommendations expressed in this work are those of the authors and do not necessarily reflect the views of any of the above sponsors.

References

- Chang Mo Ahn and Howard E. Thompson. Jump-Diffusion Processes and the Term Structure of Interest Rates. *Journal of Finance*, pages 155–174, 1988.
- John Attia, John P. A. Ioannidis, et al. How to Use an Article About Genetic Association B: Are the Results of the Study Valid? *JAMA: The Journal of the American Medical Association*, 301(2): 191, 2009.
- Matthew J. Beal, Zoubin Ghahramani, and Carl E. Rasmussen. The Infinite Hidden Markov Model. In *Advances in Neural Information Processing Systems*, pages 577–584, 2001.
- Yoav Benjamini and Yoşef Hochberg. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 289–300, 1995.
- Jadwiga R. Bienkowska, Gul S. Dalgin, Franak Batliwalla, Normand Allaire, Ronem Roubenoff, Peter K. Gregersen, and John P. Carulli. Convergent Random Forest Predictor: Methodology for

Predicting Drug Response from Genome-Scale Data Applied to Anti-TNF Response. *Genomics*, 94(6):423–432, 2009.

David M. Blei, Thomas L. Griffiths, Michael I. Jordan, and Joshua B. Tenenbaum. Hierarchical Topic Models and the Nested Chinese Restaurant Process. In *Advances in Neural Information Processing Systems 16*, volume 16, page 17. The MIT Press, 2004.

David M. Blei, Thomas L. Griffiths, and Michael I. Jordan. The Nested Chinese Restaurant Process and Bayesian Nonparametric Inference of Topic Hierarchies. *Journal of the ACM (JACM)*, 57(2): 7, 2010.

George E. P. Box. Robustness in the Strategy of Scientific Model Building. *Robustness in Statistics*, 1:201–236, 1979.

Bureau of Labor Statistics. U.S. Bureau of Labor Statistics Unemployment Analysis Methodology, 2014. URL <http://www.bls.gov/lau/lauseas.htm>.

Ying Cui, Xiaoli Z. Fern, and Jennifer G. Dy. Non-Redundant Multi-View Clustering via Orthogonalization. In *icdm*, pages 133–142. IEEE Computer Society, 2007.

David B. Dunson and Chuanhua Xing. Nonparametric Bayes Modeling of Multivariate Categorical Data. *Journal of the American Statistical Association*, 104(487):1042–1051, 2009.

Gal Elidan and Nir Friedman. Learning Hidden Variable Networks: The Information Bottleneck Approach. *The Journal of Machine Learning Research*, 6:81–127, 2005.

Gal Elidan, Noam Lotner, Nir Friedman, and Daphne Koller. Discovering Hidden Variables: A Structure-Based Approach. *Advances in Neural Information Processing Systems*, pages 479–485, 2001.

Michael D. Escobar and Mike West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90:577–588, 1995.

Elliott Fisher, David Goodman, Jonathan Skinner, and Kristen Bronner. Health Care Spending, Quality, and Outcomes: More Isn't Always Better. 2009. URL http://www.dartmouthatlas.org/downloads/reports/Spending_Brief_022709.pdf.

Elliott S. Fisher, David C. Goodman, John E. Wennberg, and Kristen K. Bronner. The Dartmouth Atlas of Health Care, August 2011. URL <http://www.dartmouthatlas.org/>.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse Inverse Covariance Estimation with the Graphical Lasso. *Biostatistics*, 9(3):432–441, 2008.

Atul Gawande. The Cost Conundrum. *The New Yorker*, June 2009.

John Geweke. Evaluating the Accuracy of Sampling-Based Approaches to the Calculation of Posterior Moments. In J.O. Berger, J.M. Bernardo, A.P. Dawid, and A.F.M. Smith, editors, *Proceedings of the Fourth Valencia International Meeting on Bayesian Statistics*, pages 169–194. Oxford University Press, 1992.

- Zoubin Ghahramani and Katherine A. Heller. Bayesian Sets. In *Advances in Neural Information Processing Systems*, pages 435–442, 2006.
- Walker R. Gilks. *Markov Chain Monte Carlo In Practice*. Chapman and Hall/CRC, 1999. ISBN 0412055511.
- Paolo Giudici and Peter J. Green. Decomposable Graphical Gaussian Model Determination. *Biometrika*, 86:785–801, 1999.
- Yue Guan, Jennifer G. Dy, Dongjin Niu, and Zoubin Ghahramani. Variational Inference for Non-parametric Multiple Clustering. In *KDD10 Workshop on Discovering, Summarizing, and Using Multiple Clusterings*, 2010.
- David J. Hand. Classifier Technology and the Illusion of Progress. *Statistical science*, 21(1):1–14, 2006.
- T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.
- Trevor Hastie, Robert Tibshirani, Gavin Sherlock, Michael Eisen, Patrick Brown, and David Botstein. Imputing Missing Data for Gene Expression Arrays, 1999.
- Sonia Jain and Radford M. Neal. A Split-Merge Markov Chain Monte Carlo procedure for the Dirichlet Process Mixture Model. *Journal of Computational and Graphical Statistics*, 2000.
- Michael I. Jordan. Hierarchical Models, Nested Models and Completely Random Measures. *Frontiers of Statistical Decision Making and Bayesian Analysis: in Honor of James O. Berger*. New York: Springer, 2010.
- Yann LeCun and Corinna Cortes. The MNIST Database. August 2011. URL <http://yann.lecun.com/exdb/mnist/>.
- James P. LeSage. Applied Econometrics Using MATLAB. Unpublished manuscript, 1999.
- Dazhou Li and Patrick Shafiq. Bayesian Hierarchical Cross-Clustering. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011.
- Vikash Mansinghka, Tejas D. Kulkarni, Yura N. Perov, and Josh Tenenbaum. Approximate Bayesian image interpretation using generative probabilistic graphics programs. In *Advances in Neural Information Processing Systems*, pages 1520–1528, 2013.
- Vikash K. Mansinghka. Efficient Monte Carlo Inference for Infinite Relational Models, 2007. URL <http://www.cs.ubc.ca/~murphyk/nips07NetworkWorkshop/talks/vikash.pdf>.
- Vikash K. Mansinghka, Daniel M. Roy, Ryan Rifkin, and Joshua B. Tenenbaum. AClass: An online algorithm for generative classification. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2007.
- Vikash K. Mansinghka, Eric Jonas, Cap Petschnat, Beau Cronin, Patrick Shafiq, and Joshua B. Tenenbaum. Cross-categorization: A Method for Discovering Multiple Overlapping Clusters. In *Advances in Neural Information Processing Systems*, volume 22, 2009.
- Vikash K. Mansinghka, Daniel Selsam, and Yura Perov. Venture: a higher-order probabilistic programming platform with programmable inference. *arXiv preprint arXiv:1404.0099*, 2014.
- Nicolai Meinshausen and Peter Bühlmann. High-Dimensional Graphs and Variable Selection With the Lasso. *The Annals of Statistics*, pages 1436–1462, 2006.
- Kevin P. Murphy. The Bayes Net Toolbox for MATLAB. *Computing Science and Statistics*, 33: 2001, 2001.
- National Bureau of Economic Research. National Bureau of Economic Research Business Cycle Methodology, 2010. URL <http://www.nber.org/cycles/sept2010.html>.
- Radford M. Neal. Markov Chain Sampling Methods for Dirichlet Process Mixture Models, 1998.
- Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom Mitchell. Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*, 39(2/3):103–134, 1999. URL <http://www.kamal.nigam.com/papers/emcat-nl199.pdf>.
- Dongjin Niu, Jennifer G. Dy, and Michael I. Jordan. Multiple non-redundant spectral clustering views. In *Proc. ICML*. Citeseer, 2010.
- NRC Committee on the Analysis of Massive Data. *Frontiers in Massive Data Analysis*. National Academies Press, 2013. URL <http://www.nap.edu/catalog.php?recordid=18374>.
- Fritz Obermeyer, Jonathan Glidden, and Eric Jonas. Scaling Nonparametric Bayesian Inference via Subsample-Annealing. *arXiv preprint arXiv:1402.5473*, 2014.
- Jim Pitman. Some Developments of the Blackwell-MacQueen Urn Scheme. In T. S. Ferguson, L. S. Shapley, and J. B. MacQueen, editors, *Statistics, probability and game theory: Papers in honor of David Blackwell*, pages 245–267. Institute of Mathematical Statistics, 1996.
- Mich Raponi, Lesley Dossey, Tim Jakoe, Xiaoying Wul, Goan Chen, Hongtao Fan, and David G. Beer. MicroRNA classifiers for predicting prognosis of squamous cell lung cancer. *Cancer research*, 69(14):5776, 2009.
- Carl E. Rasmussen. The Infinite Gaussian Mixture Model. In *Advances in Neural Processing Systems* 12, 2000.
- Abel Rodriguez and Kaushik Ghosh. Nested Partition Models. *UCSC School of Engineering Technical Report*, 2009.
- Abel Rodriguez, David B. Dunson, and Alan E. Gelfand. The Nested Dirichlet Process. *Journal of the American Statistical Association*, 103(483):1131–1154, 2008.
- David A. Ross and Richard S. Zemel. Learning Parts-Based Representations of Data. *Journal of Machine Learning Research*, 7:2369–2397, 2006.
- Patrick Shafiq, Charles Kemp, Vikash K. Mansinghka, Matthew Gordon, and Joshua B. Tenenbaum. Learning Cross-Cutting Systems of Categories. In *Proceedings of the 28th Annual Conference of the Cognitive Science Society*, 2006.

- Patrick Shafto, Charles Kemp, Vikash K. Mansinghka, and Joshua B. Tenenbaum. A Probabilistic Model of Cross-Categorization. *Cognition*, 120:1–25, 2011.
- Nicolas Städler and Peter Bühlmann. Missing Values: Sparse Inverse Covariance Estimation and an Extension to Sparse Regression. *Statistics and Computing*, 22(1):219–235, 2012.
- Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.
- Joshua B. Tenenbaum and Thomas L. Griffiths. Generalization, Similarity, and Bayesian Inference. *Behavioral and Brain Sciences*, 24:629–641, 2001.
- Jiahui Wang and Eric Zivot. A Bayesian Time Series Model of Multiple Structural Changes in Level, Trend, and Variance. *Journal of Business & Economic Statistics*, pages 374–386, 2000.
- Larry Wasserman. Low Assumptions, High Dimensions. *Rationality, Markets and Morals Special Topic: Statistical Science and Philosophy of Science*, 2011. URL http://www.rmm-journal.de/downloads/Article_Wasserman.pdf.
- Jason Weston, Shayan Mukherjee, et al. Feature Selection for SVMs. *Advances in neural information processing systems*, pages 668–674, 2001.

Regularized Policy Iteration with Nonparametric Function Spaces

Amir-massoud Farahmand
Mitsubishi Electric Research Laboratories (MERL)
201 Broadway, 8th Floor
Cambridge, MA 02139, USA

FARAHMAND@MERL.COM

Mohammad Ghavamzadeh
Adobe Research
321 Park Avenue
San Jose, CA 95110, USA

GHAVAMZA@ADOBE.COM

Csaba Szepesvári
Department of Computing Science
University of Alberta
Edmonton, AB, T6G 2E8, Canada

SZEPESVA@UALBERTA.CA

Shie Mannor
Department of Electrical Engineering
The Technion
Haifa 32000, Israel

SHIE@EE.TECHNION.AC.IL

Editor: Peter Auer

Abstract

We study two regularization-based approximate policy iteration algorithms, namely REG-LSPI and REG-BRM, to solve reinforcement learning and planning problems in discounted Markov Decision Processes with large state and finite action spaces. The core of these algorithms are the regularized extensions of the Least-Squares Temporal Difference (LSTD) learning and Bellman Residual Minimization (BRM), which are used in the algorithms' policy evaluation steps. Regularization provides a convenient way to control the complexity of the function space to which the estimated value function belongs and as a result enables us to work with rich nonparametric function spaces. We derive efficient implementations of our methods when the function space is a reproducing kernel Hilbert space. We analyze the statistical properties of REG-LSPI and provide an upper bound on the policy evaluation error and the performance loss of the policy returned by this method. Our bound shows the dependence of the loss on the number of samples, the capacity of the function space, and some intrinsic properties of the underlying Markov Decision Process. The dependence of the policy evaluation bound on the number of samples is minimax optimal. This is the first work that provides such a strong guarantee for a nonparametric approximate policy iteration algorithm.¹

Keywords: reinforcement learning, approximate policy iteration, regularization, nonparametric method, finite-sample analysis

1. Introduction

We study the approximate policy iteration (API) approach to find a close to optimal policy in a Markov Decision Process (MDP), either in a reinforcement learning (RL) or in a planning scenario. The basis of API, which is explained in Section 3, is the policy iteration algorithm that iteratively evaluates a policy (i.e., finding the value function of the policy—the *policy evaluation* step) and then improves it (i.e., computing the *greedy* policy with respect to (w.r.t.) the recently obtained value function—the *policy improvement* step). When the state space is large (e.g., a subset of \mathbb{R}^d or a finite state space that has too many states to be exactly represented), the policy evaluation step cannot be performed exactly, and as a result the use of function approximation is inevitable. The appropriate choice of the function approximation method, however, is far from trivial. The best choice is problem-dependent and it also depends on the number of samples in the input data.

In this paper we propose a *nonparametric regularization*-based approach to API. This approach provides a flexible and easy way to implement the policy evaluation step of API. The advantage of nonparametric methods over parametric methods is that they are flexible: Whereas a parametric model, which has a fixed and finite parameterization, limits the range of functions that can be represented, irrespective of the number of samples, the nonparametric models avoid such undue restrictions by increasing the power of the function approximation as necessary. Moreover, the regularization-based approach to nonparametrics is elegant and powerful: It has a simple algorithmic form and the estimator achieves minimax optimal rates in a number of scenarios. Further discussion of and specific results about nonparametric methods, particularly in the supervised learning scenario, can be found in the books by Györfi et al. (2002) and Wasserman (2007).

The nonparametric approaches to solve RL/Planning problems have received some attention in the RL community. For instance, Petrik (2007); Mahadevan and Maggioni (2007); Parr et al. (2007); Mahadevan and Liu (2010); Geramifard et al. (2011); Farahmand and Precup (2012); Böhrer et al. (2013) and Milani Fard et al. (2013) suggest methods to generate data-dependent basis functions, to be used in general linear models. Ormonett and Sen (2002) use smoothing kernel-based estimate of the model and then use value iteration to find the value function. Barreto et al. (2011, 2012) benefit from “stochastic factorization trick” to provide computationally efficient ways to scale up the approach of Ormonett and Sen (2002). In the context of approximate value iteration, Ernst et al. (2005) consider growing ensembles of trees to approximate the value function. In addition, there have been some works where regularization methods have been applied to the RL/Planning problems, e.g., Engel et al. (2005); Jung and Polani (2006); Loth et al. (2007); Farahmand et al. (2009a,b); Taylor and Parr (2009); Kolter and Ng (2009); Johns et al. (2010); Ghavamzadeh et al. (2011); Farahmand (2011b); Ávila Pires and Szepesvári (2012); Hoffman et al. (2012); Geist and Scherrer (2012). Nevertheless, most of these papers are algorithmic results and do not analyze the statistical properties of these methods (the exceptions are Farahmand et al. 2009a,b; Farahmand 2011b; Ghavamzadeh et al. 2011; Ávila Pires and Szepesvári 2012). We compare these methods with ours in more detail in Sections 5.3.1 and 6.

It is worth mentioning that one might use a regularized estimator alongside a feature generation approach to control the complexity of function space induced by the features. An approach alternative to regularization for controlling the complexity of a function space is to

¹ This work is an extension of the NIPS 2008 conference paper by Farahmand et al. (2009b).

use greedy algorithms, such as Matching Pursuit (Mallat and Zhang, 1993) and Orthogonal Matching Pursuit (Pati et al., 1993), to select features from a large set of features. Greedy algorithms have recently been developed for the value function estimation by Johns (2010); Painter-Wakefield and Parr (2012); Farahmand and Precup (2012); Geramifard et al. (2013). We do not discuss these methods any further.

1.1 Contributions

The algorithmic contribution of this work is to introduce two regularization-based nonparametric approximate policy iteration algorithms, namely *Regularized Least-Squares Policy Improvement (REG-LSPI)* and *Regularized Bellman Residual Minimization (REG-BRM)*. These are flexible methods that, upon the proper selection of their parameters, are sample efficient. Each of REG-BRM and REG-LSPI is formulated as two coupled regularized optimization problems (Section 4). As we argue in Section 4.1, having a regularized objective in both optimization problems is necessary for rich nonparametric function spaces. Despite the unusual coupled formulation of the underlying optimization problems, we prove that the solutions can be computed in a closed-form when the estimated action-value function belongs to the family of *reproducing kernel Hilbert spaces (RKHS)* (Section 4.2).

The theoretical contribution of this work (Section 5) is to analyze the statistical properties of REG-LSPI and to provide upper bounds on the policy evaluation error and the performance difference between the optimal policy and the policy returned by this method (Theorem 14). The result demonstrates the dependence of the bounds on the number of samples, the capacity of the function space to which the estimated action-value function belongs, and some intrinsic properties of the MDP. It turns out that the dependence of the policy evaluation error bound on the number of samples is minimax optimal. This paper, alongside its conference (Farahmand et al., 2009b) and the dissertation (Farahmand, 2011b) versions, is the first work that analyzes a nonparametric regularized API algorithm and provides such a strong guarantee for it.

2. Background and Notation

In the first part of this section, we provide a brief summary of some of the concepts and definitions from the theory of MDPs and RL (Section 2.1). For more information, the reader is referred to Bertsekas and Shreve (1978); Bertsekas and Tsitsiklis (1996); Sutton and Barto (1998); Szepesvári (2010). In addition to this background on MDPs, we introduce the notations we use to denote function spaces and their corresponding norms (Section 2.2) as well as the considered learning problem (Section 2.3).

2.1 Markov Decision Processes

For a space Ω , with a σ -algebra σ_Ω , we define $\mathcal{M}(\Omega)$ as the set of all probability measures over σ_Ω . We let $B(\Omega)$ denote the space of bounded measurable functions w.r.t. σ_Ω and we denote $B(\Omega, L)$ as the space of bounded measurable functions with bound $0 < L < \infty$.

Definition 1 A finite-action discounted MDP is a 4-tuple $(\mathcal{X}, \mathcal{A}, P, \gamma)$, where \mathcal{X} is a measurable state space, \mathcal{A} is a finite set of actions, $P : \mathcal{X} \times \mathcal{A} \rightarrow \mathcal{M}(\mathbb{R} \times \mathcal{X})$ is a mapping with domain $\mathcal{X} \times \mathcal{A}$, and $0 \leq \gamma < 1$ is a discount factor. Mapping P evaluated

at $(x, a) \in \mathcal{X} \times \mathcal{A}$ gives a distribution over $\mathbb{R} \times \mathcal{X}$, which we shall denote by $P(\cdot, \cdot | x, a)$. We denote the marginals of P by the overloaded symbol $P : \mathcal{X} \times \mathcal{A} \rightarrow \mathcal{M}(\mathcal{X})$ defined as $P(\cdot | x, a) = P_{x,a}(\cdot) = \int_{\mathbb{R}} P(dr, \cdot | x, a)$ (transition probability kernel) and $\mathcal{R} : \mathcal{X} \times \mathcal{A} \rightarrow \mathcal{M}(\mathbb{R})$ defined as $\mathcal{R}(\cdot | x, a) = \int_{\mathcal{X}} P(\cdot, dy | x, a)$ (reward distribution).

An MDP together with an initial distribution P_1 of states encode the laws governing the temporal evolution of a discrete-time stochastic process controlled by an agent as follows: The controlled process starts at time $t = 1$ with random initial state $X_1 \sim P_1$ (here and in what follows $X \sim Q$ denotes that the random variable X is drawn from distribution Q). At stage t , action $A_t \in \mathcal{A}$ is selected by the agent controlling the process. In response, the pair (R_t, X_{t+1}) is drawn from $P(\cdot, \cdot | X_t, A_t)$, i.e., $(R_t, X_{t+1}) \sim P(\cdot, \cdot | X_t, A_t)$, where, R_t is the reward that the agent receives at time t and X_{t+1} is the state at time $t + 1$. The process then repeats with the agent selecting action A_{t+1} , etc.

In general, the agent can use all past states, actions, and rewards in deciding about its current action. However, for our purposes it will suffice to consider action-selection procedures, or policies, that select an action deterministically and time-invariantly solely based on the current state:

Definition 2 (Deterministic Markov Stationary Policy) A measurable mapping $\pi : \mathcal{X} \rightarrow \mathcal{A}$ is called a deterministic Markov stationary policy, or just policy in short. Following a policy π in an MDP means that at each time step t it holds that $A_t = \pi(X_t)$.

Policy π induces the transition probability kernels $P^\pi : \mathcal{X} \times \mathcal{A} \rightarrow \mathcal{M}(\mathcal{X} \times \mathcal{A})$ defined as follows: For a measurable subset C of $\mathcal{X} \times \mathcal{A}$, let $(P^\pi)(C | x, a) \triangleq \int P(dy | x, a) \mathbb{1}_C(y, \pi(y)) \in \mathcal{C}$. The m -step transition probability kernels $(P^\pi)^m : \mathcal{X} \times \mathcal{A} \rightarrow \mathcal{M}(\mathcal{X} \times \mathcal{A})$ for $m = 2, 3, \dots$ are defined inductively by $(P^\pi)^m(C | x, a) \triangleq \int_{\mathcal{X}} P(dy | x, a) (P^\pi)^{m-1}(C | y, \pi(y))$. Also given a probability transition kernel $P : \mathcal{X} \times \mathcal{A} \rightarrow \mathcal{M}(\mathcal{X} \times \mathcal{A})$, we define the right-linear operator $P : B(\mathcal{X} \times \mathcal{A}) \rightarrow B(\mathcal{X} \times \mathcal{A})$ by $(PQ)(x, a) \triangleq \int_{\mathcal{X} \times \mathcal{A}} P(dy, da | x, a) Q(y, a)$. For a probability measure $\rho \in \mathcal{M}(\mathcal{X} \times \mathcal{A})$ and a measurable subset C of $\mathcal{X} \times \mathcal{A}$, we define the left-linear operators $\cdot P : \mathcal{M}(\mathcal{X} \times \mathcal{A}) \rightarrow \mathcal{M}(\mathcal{X} \times \mathcal{A})$ by $(\rho P)(C) = \int \rho(dx, da) P(dy, da | x, a) \mathbb{1}_C(y, a) \in \mathcal{C}$. To study MDPs, two auxiliary functions are of central importance: the value and the action-value functions of a policy π .

Definition 3 (Value Functions) For a policy π , the value function V^π and the action-value function Q^π are defined as follows: Let $(R_t; t \geq 1)$ be the sequence of rewards when the Markov chain is started from a state X_1 (or state-action (X_1, A_1) for the action-value function) drawn from a positive probability distribution over \mathcal{X} (or $\mathcal{X} \times \mathcal{A}$) and the agent follows policy π . Then, $V^\pi(x) \triangleq \mathbb{E} \left[\sum_{t=1}^{\infty} \gamma^{t-1} R_t \mid X_1 = x \right]$ and $Q^\pi(x, a) \triangleq \mathbb{E} \left[\sum_{t=1}^{\infty} \gamma^{t-1} R_t \mid X_1 = x, A_1 = a \right]$.

It is easy to see that for any policy π , if the magnitude of the immediate expected reward $r(x, a) = \int r P(dr, dy | x, a)$ is uniformly bounded by R_{\max} , then the functions V^π and Q^π are bounded by $V_{\max} = Q_{\max} = R_{\max} / (1 - \gamma)$, independent of the choice of π .

For a discounted MDP, we define the optimal value and optimal action-value functions by $V^*(x) = \sup_{\pi} V^\pi(x)$ for all states $x \in \mathcal{X}$ and $Q^*(x, a) = \sup_{\pi} Q^\pi(x, a)$ for all state-actions $(x, a) \in \mathcal{X} \times \mathcal{A}$. We say that a policy π^* is optimal if it achieves the best values in every

state, i.e., if $V^{\pi^*} = V^*$. We say that a policy π is *greedy* w.r.t. an action-value function Q if $\pi(x) = \arg\max_{a \in \mathcal{A}} Q(x, a)$ for all $x \in \mathcal{X}$. We define function $\tilde{\pi}(x; Q) \triangleq \arg\max_{a \in \mathcal{A}} Q(x, a)$ (for all $x \in \mathcal{X}$) that returns a greedy policy of an action-value function Q (If there exist multiple maximizers, a maximizer is chosen in an arbitrary deterministic manner). Greedy policies are important because a greedy policy w.r.t. the optimal action-value function Q^* is an optimal policy. Hence, knowing Q^* is sufficient for behaving optimally (cf. Proposition 4.3 of Bertsekas and Shreve 1978).²

Definition 4 (Bellman Operators) For a policy π , the Bellman operators $T^\pi : B(\mathcal{X}) \rightarrow B(\mathcal{X})$ (for value functions) and $T^\pi : B(\mathcal{X} \times \mathcal{A}) \rightarrow B(\mathcal{X} \times \mathcal{A})$ (for action-value functions) are defined as

$$\begin{aligned} (T^\pi V)(x) &\triangleq r(x, \pi(x)) + \gamma \int_{\mathcal{X}} V(y) P(dy|x, \pi(x)), \\ (T^\pi Q)(x, a) &\triangleq r(x, a) + \gamma \int_{\mathcal{X}} Q(y, \pi(y)) P(dy|x, a). \end{aligned}$$

To avoid unnecessary clutter, we use the same symbol to denote both operators. However, this should not introduce any ambiguity: Given some expression involving T^π one can always determine which operator T^π means by looking at the type of function T^π is applied to. It is known that the fixed point of the Bellman operator is the (action-)value function of the policy π , i.e., $T^\pi Q^\pi = Q^\pi$ and $T^\pi V^\pi = V^\pi$, see e.g., Proposition 4.2(b) of Bertsekas and Shreve (1978). We will also need to define the so-called Bellman optimality operators:

Definition 5 (Bellman Optimality Operators) The Bellman optimality operators $T^* : B(\mathcal{X}) \rightarrow B(\mathcal{X})$ (for value functions) and $T^* : B(\mathcal{X} \times \mathcal{A}) \rightarrow B(\mathcal{X} \times \mathcal{A})$ (for action-value functions) are defined as

$$\begin{aligned} (T^* V)(x) &\triangleq \max_a \left\{ r(x, a) + \gamma \int_{\mathcal{X}} V(y) P(dy|x, a) \right\}, \\ (T^* Q)(x, a) &\triangleq r(x, a) + \gamma \int_{\mathcal{X}} \max_{a'} Q(y, a') P(dy|x, a). \end{aligned}$$

Again, we use the same symbol to denote both operators; the previous comment that no ambiguity should arise because of this still applies. The Bellman optimality operators enjoy a fixed-point property similar to that of the Bellman operators. In particular, $T^* V^* = V^*$ and $T^* Q^* = Q^*$, see e.g., Proposition 4.2(a) of Bertsekas and Shreve (1978). The Bellman optimality operator thus provides a vehicle to compute the optimal action-value function and therefore to compute an optimal policy.

² Measurability issues are dealt with in Section 9.5 of the same book. In the case of finitely many actions, no additional condition is needed besides the obvious measurability assumptions on the immediate reward function and the transition kernel (Bertsekas and Shreve, 1978, Corollary 9.17.1), which we will assume from now on.

2.2 Norms and Function Spaces

In what follows we use $\mathcal{F} : \mathcal{X} \rightarrow \mathbb{R}$ to denote a subset of measurable functions. The exact specification of this set will be clear from the context. Further, we let $\mathcal{F}^{\mathcal{A}} : \mathcal{X} \times \mathcal{A} \rightarrow \mathbb{R}^{|\mathcal{A}|}$ to be a subset of vector-valued measurable functions with the identification of

$$\mathcal{F}^{\mathcal{A}} = \{(Q_1, \dots, Q_{|\mathcal{A}|}) : Q_i \in \mathcal{F}, i = 1, \dots, |\mathcal{A}|\}.$$

We shall use $\|Q\|_{p, \nu}$ to denote the $L_p(\nu)$ -norm ($1 \leq p < \infty$) of a measurable function $Q : \mathcal{X} \times \mathcal{A} \rightarrow \mathbb{R}$, i.e., $\|Q\|_{p, \nu} \triangleq \int_{\mathcal{X} \times \mathcal{A}} |Q(x, a)|^p d\nu(x, a)$.

Let $z_{1:n}$ denote the \mathcal{Z} -valued sequence (z_1, \dots, z_n) . For $\mathcal{D}_n = z_{1:n}$, define the empirical norm of function $f : \mathcal{Z} \rightarrow \mathbb{R}$ as

$$\|f\|_{p, \mathcal{D}_n}^p \triangleq \frac{1}{n} \sum_{i=1}^n |f(z_i)|^p. \quad (1)$$

When there is no chance of confusion about \mathcal{D}_n , we may denote the empirical norm by $\|f\|_{p, n}$. Based on this definition, one may define $\|Q\|_{p, \mathcal{D}_n}$ with the choice of $\mathcal{Z} = \mathcal{X} \times \mathcal{A}$. Note that if $\mathcal{D}_n = (Z_i)_{i=1}^n$ is random with $Z_i \sim \nu$, the empirical norm is random too, and for any fixed function f , we have $\mathbb{E}[\|f\|_{p, \mathcal{D}_n}] = \|f\|_{p, \nu}$. When $p = 2$, we simply use $\|\cdot\|_{\mathcal{D}_n}$.

2.3 Offline Learning Problem and Empirical Bellman Operators

We consider the *offline learning* scenario when we are only given a batch of data³

$$\mathcal{D}_n = \{(X_1, A_1, R_1, X'_1), \dots, (X_n, A_n, R_n, X'_n)\}, \quad (2)$$

with $X_i \sim \nu_{\mathcal{X}}$, $A_i \sim \pi_0(\cdot|X_i)$, and $(R_i, X'_i) \sim P(\cdot, \cdot|X_i, A_i)$ for $i = 1, \dots, n$. Here $\nu_{\mathcal{X}} \in \mathcal{M}(\mathcal{X})$ is a fixed distribution over the states and π_0 is the data generating behavior policy, which is a stochastic stationary Markov policy, i.e., given any state $x \in \mathcal{X}$, it assigns a probability distribution over \mathcal{A} . We shall also denote the common distribution underlying (X_i, A_i) by $\nu \in \mathcal{M}(\mathcal{X} \times \mathcal{A})$.

Samples X_i and X'_{i+1} may be sampled independently (we call this the “*Planning scenario*”), or may be coupled through $X'_i = X_{i+1}$ (“*RL scenario*”). In the latter case the data comes from a single trajectory. Under either of these scenarios, we say that the data \mathcal{D}_n meets the *standard offline sampling assumption*. We analyze the Planning scenario, where the states are independent, but one may also analyze dependent processes by considering mixing processes and using tools such as the independent blocks technique (Yu, 1994; Doukhan, 1994), as has been done by Antos et al. (2008b); Farahmand and Szepesvári (2012).

The data set \mathcal{D}_n allows us to define the so-called *empirical Bellman operators*, which can be thought of as empirical approximations to the true Bellman operators.

³ In what follows, when $\{\cdot\}$ is used in connection to a data set, we treat the set as an ordered multiset, where the ordering is given by the time indices of the data points.

Definition 6 (Empirical Bellman Operators) Let \mathcal{D}_n be a data set as above. Define the ordered multiset $S_n = \{(X_1, A_1), \dots, (X_n, A_n)\}$. For a given fixed policy π , the empirical Bellman operator $\hat{T}^\pi : \mathbb{R}^{S_n} \rightarrow \mathbb{R}^n$ is defined as

$$(\hat{T}^\pi Q)(X_i, A_i) \triangleq R_i + \gamma Q(X_i', \pi(X_i')), \quad 1 \leq i \leq n.$$

Similarly, the empirical Bellman optimality operator \hat{T}^* : $\mathbb{R}^{S_n} \rightarrow \mathbb{R}^n$ is defined as

$$(\hat{T}^* Q)(X_i, A_i) \triangleq R_i + \gamma \max_{a'} Q(X_i', a'), \quad 1 \leq i \leq n.$$

In words, the empirical Bellman operators get an n -element list S_n and return an n -dimensional real-valued vector of the single-sample estimate of the Bellman operators applied to the action-value function Q at the selected points. It is easy to see that the empirical Bellman operators provide an unbiased estimate of the Bellman operators in the following sense: For any fixed bounded measurable deterministic function $Q : \mathcal{X} \times \mathcal{A} \rightarrow \mathbb{R}_k$, policy π and $1 \leq i \leq n$, it holds that $\mathbb{E}[\hat{T}^\pi Q(X_i, A_i) | X_i, A_i] = T^\pi Q(X_i, A_i)$ and $\mathbb{E}[\hat{T}^* Q(X_i, A_i) | X_i, A_i] = T^* Q(X_i, A_i)$.

3. Approximate Policy Iteration

The policy iteration algorithm computes a sequence of policies such that the new policy in the iteration is greedy w.r.t. the action-value function of the previous policy. This procedure requires one to compute the action-value function of the most recent policy (policy evaluation step) followed by the computation of the greedy policy (policy improvement step). In API, the exact, but infeasible, policy evaluation step is replaced by an approximate one. Thus, the skeleton of API methods is as follows: At the k^{th} iteration and given a policy π_k , the API algorithm approximately evaluates π_k to find a Q_k . The action-value function Q_k is typically chosen to be such that $Q_k \approx T^{\pi_k} Q_k$, i.e., it is an approximate fixed point of T^{π_k} . The API algorithm then calculates the greedy policy w.r.t. the most recent action-value function to obtain a new policy π_{k+1} , i.e., $\pi_{k+1} = \hat{\pi}(\cdot; Q_k)$. The API algorithm continues by repeating this process again and generating a sequence of policies and their corresponding approximate action-value functions $Q_0 \rightarrow \pi_1 \rightarrow Q_1 \rightarrow \pi_2 \rightarrow \dots$ ⁴.

The success of an API algorithm hinges on the way the approximate policy evaluation step is implemented. Approximate policy evaluation is non-trivial for at least two reasons. First, policy evaluation is an inverse problem,⁵ so the underlying learning problem is unlike a standard supervised learning problem in which the data take the form of input-output pairs. The second problem is the off-policy sampling problem: The distribution of (X_i, A_i) in the data samples (possibly generated by a behavior policy) is typically different from the distribution that would be induced if we followed the to-be-evaluated policy (i.e., target policy). This causes a problem since the methods must be able to handle this mismatch of

distributions.⁶ In the rest of this section, we review generic LSTD and BRM methods for approximate policy evaluation. We introduce our regularized version of LSTD and BRM in Section 4.

3.1 Bellman Residual Minimization

The idea of BRM goes back at least to the work of [Schweitzer and Seidmann \(1985\)](#). It was later used in the RL community by [Williams and Baird \(1994\)](#) and [Baird \(1995\)](#). The basic idea of BRM comes from noticing that the action-value function is the unique fixed point of the Bellman operator: $Q^\pi = T^\pi Q^\pi$ (or similarly $V^\pi = T^\pi V^\pi$ for the value function). Whenever we replace Q^π by an action-value function Q different from Q^π , the fixed-point equation would not hold anymore, and we have a non-zero residual function $Q - T^\pi Q$. This quantity is called the *Bellman residual* of Q . The same is true for the Bellman optimality operator T^* .

The BRM algorithm minimizes the norm of the Bellman residual of Q , which is called the *Bellman error*. It can be shown that if $\|Q - T^* Q\|$ is small, then the value function of the greedy policy w.r.t. Q , that is $V^{\hat{\pi}(\cdot; Q)}$, is also in some sense close to the optimal value function V^* , see e.g., [Williams and Baird \(1994\)](#); [Munos \(2003\)](#); [Antos et al. \(2008b\)](#); [Farahmand et al. \(2010\)](#), and [Theorem 13](#) of this work. The BRM algorithm is defined as the procedure minimizing the following loss function:

$$L_{BRM}(Q; \pi) \triangleq \|Q - T^\pi Q\|_\nu^2,$$

where ν is the distribution of state-actions in the input data. Using the empirical L_2 -norm defined in (1) with samples \mathcal{D}_n defined in (2), and by replacing $(T^\pi Q)(X_i, A_i)$ with the empirical Bellman operator (Definition 6), the empirical estimate of $L_{BRM}(Q; \pi)$ can be written as

$$\hat{L}_{BRM}(Q; \pi, n) \triangleq \|Q - \hat{T}^\pi Q\|_{\mathcal{D}_n}^2 = \frac{1}{n} \sum_{i=1}^n [Q(X_i, A_i) - (R_i + \gamma Q(X_i', \pi(X_i')))]^2. \quad (3)$$

Nevertheless, it is well-known that \hat{L}_{BRM} is not an unbiased estimate of L_{BRM} when the MDP is not deterministic ([Lagoudakis and Parr, 2003](#); [Antos et al., 2008b](#)). To address this issue, [Antos et al. \(2008b\)](#) propose the modified BRM loss that is a new empirical loss function with an extra *de-biasing* term. The idea of the modified BRM is to cancel the unwanted variance by introducing an auxiliary function h and a new loss function

$$L_{BRM}(Q, h; \pi) = L_{BRM}(Q; \pi) - \|h - T^\pi Q\|_\nu^2, \quad (4)$$

and approximating the action-value function Q^π by solving

$$Q_{BRM} = \operatorname{argmin}_{Q \in \mathcal{F}^{\mathcal{A}}} \sup_{h \in \mathcal{F}^{\mathcal{A}}} L_{BRM}(Q, h; \pi), \quad (5)$$

4. In an actual API implementation, one does not need to compute π_{k+1} for all states, which in fact is infeasible for large state spaces. Instead, one uses Q_k to compute π_{k+1} at some select states, as required in the approximate policy evaluation step.

5. Given an operator $\mathcal{L} : \mathcal{F} \rightarrow \mathcal{F}$, the inverse problem is the problem of solving $g = \mathcal{L}f$ for f when g is known. In the policy evaluation problem, $\mathcal{L} = \mathbf{I} - \gamma T^{\pi^*}$, $g(\cdot) = r(\cdot, \pi(\cdot))$, and $f = Q^{\pi^*}$.

6. A number of works in the domain adaptation literature consider this scenario under the name of covariate shift problem, see e.g., [Ben-David et al. 2006](#); [Mansour et al. 2009](#); [Ben-David et al. 2010](#); [Cortes et al. 2015](#).

where the supremum comes from the negative sign of $\|h - T^\pi Q\|_\nu^2$. They have shown that optimizing the new loss function still makes sense and the empirical version of this loss is unbiased.

The min-max optimization problem (5) is equivalent to the following coupled (nested) optimization problems:

$$\begin{aligned} h(\cdot; Q) &= \operatorname{argmin}_{h \in \mathcal{F}^{|\mathcal{A}|}} \|h' - T^\pi Q\|_\nu^2, \\ Q_{BRM} &= \operatorname{argmin}_{Q \in \mathcal{F}^{|\mathcal{A}|}} [\|Q - T^\pi Q\|_\nu^2 - \|h(\cdot; Q) - T^\pi Q\|_\nu^2]. \end{aligned} \quad (6)$$

In practice, the norm $\|\cdot\|_\nu$ is replaced by the empirical norm $\|\cdot\|_{\mathcal{D}_n}$ and $T^\pi Q$ is replaced by its sample-based approximation $\hat{T}^\pi Q$, i.e.,

$$\begin{aligned} \hat{h}_n(\cdot; Q) &= \operatorname{argmin}_{h \in \mathcal{F}^{|\mathcal{A}|}} \|h - \hat{T}^\pi Q\|_{\mathcal{D}_n}^2, \\ \hat{Q}_{BRM} &= \operatorname{argmin}_{Q \in \mathcal{F}^{|\mathcal{A}|}} [\|Q - \hat{T}^\pi Q\|_{\mathcal{D}_n}^2 - \|\hat{h}_n(\cdot; Q) - \hat{T}^\pi Q\|_{\mathcal{D}_n}^2]. \end{aligned} \quad (7) \quad (8)$$

From now on, whenever we refer to the BRM algorithm, we are referring to this modified BRM.

3.2 Least-Squares Temporal Difference Learning

The Least-Squares Temporal Difference learning (LSTD) algorithm for policy evaluation was first proposed by [Bradtke and Barto \(1996\)](#), and later used in an API procedure by [Lagoudakis and Parr \(2003\)](#) and was called Least-Squares Policy Iteration (LSPI).

The original formulation of LSTD finds a solution to the fixed-point equation $Q = \Pi_\nu T^\pi Q$, where Π_ν is the simplified notation for ν -weighted projection operator onto the space of admissible functions $\mathcal{F}^{|\mathcal{A}|}$, i.e., $\Pi_\nu \triangleq \Pi_{\mathcal{F}^{|\mathcal{A}|}} : B(\mathcal{X} \times \mathcal{A}) \rightarrow B(\mathcal{X} \times \mathcal{A})$ is defined by $\Pi_{\mathcal{F}^{|\mathcal{A}|}} Q = \operatorname{argmin}_{h \in \mathcal{F}^{|\mathcal{A}|}} \|h - Q\|_\nu^2$ for $Q \in B(\mathcal{X} \times \mathcal{A})$. We, however, use a different optimization-based formulation. The reason is that whenever ν is not the stationary distribution induced by π , the operator $(\Pi_\nu T^\pi)$ does not necessarily have a fixed point, but the optimization problem is always well-defined.

We define the LSTD solution as the minimizer of the L_2 -norm between Q and $\Pi_\nu T^\pi Q$:

$$L_{LSTD}(Q; \pi) \triangleq \|Q - \Pi_\nu T^\pi Q\|_\nu^2. \quad (9)$$

The minimizer of $L_{LSTD}(Q; \pi)$ is well-defined, and whenever ν is the stationary distribution of π (i.e., on-policy sampling), the solution to this optimization problem is the same as the solution to $Q = \Pi_\nu T^\pi Q$. The LSTD solution can therefore be written as the solution to the following set of coupled optimization problems:

$$\begin{aligned} h(\cdot; Q) &= \operatorname{argmin}_{h \in \mathcal{F}^{|\mathcal{A}|}} \|h' - T^\pi Q\|_\nu^2, \\ Q_{LSTD} &= \operatorname{argmin}_{Q \in \mathcal{F}^{|\mathcal{A}|}} \|Q - h(\cdot; Q)\|_\nu^2, \end{aligned} \quad (10)$$

Algorithm 1 Regularized Policy Iteration($K, \hat{Q}^{(K-1)}, \mathcal{F}^{|\mathcal{A}|}, J, \{(\lambda_{Q_n}^{(k)}, \lambda_{h_n}^{(k)})\}_{k=0}^{K-1}$)

```

// K: Number of iterations
//  $\hat{Q}^{(K-1)}$ : Initial action-value function
//  $\mathcal{F}^{|\mathcal{A}|}$ : The action-value function space
// J: The regularizer
//  $\{(\lambda_{Q_n}^{(k)}, \lambda_{h_n}^{(k)})\}_{k=0}^K$ : The regularization coefficients
for  $k = 0$  to  $K - 1$  do
   $\pi_k(\cdot) \leftarrow \hat{\pi}(\cdot; \hat{Q}^{(k-1)})$ 
  Generate training samples  $\mathcal{D}_n^{(k)}$ 
   $\hat{Q}^{(k)} \leftarrow \text{REG-LSTD/BRM}(\pi_k; \mathcal{D}_n^{(k)}; \mathcal{F}^{|\mathcal{A}|}, J, \lambda_{Q_n}^{(k)}, \lambda_{h_n}^{(k)})$ 
end for
return  $\hat{Q}^{(K-1)}$  and  $\pi_K(\cdot) = \hat{\pi}(\cdot; \hat{Q}^{(K-1)})$ 

```

where the first equation finds the projection of $T^\pi Q$ onto $\mathcal{F}^{|\mathcal{A}|}$, and the second one minimizes the distance of Q and the projection. The corresponding empirical version based on data set \mathcal{D}_n is

$$\hat{h}_n(\cdot; Q) = \operatorname{argmin}_{h \in \mathcal{F}^{|\mathcal{A}|}} \|h - \hat{T}^\pi Q\|_{\mathcal{D}_n}^2, \quad (11)$$

$$\hat{Q}_{LSTD} = \operatorname{argmin}_{Q \in \mathcal{F}^{|\mathcal{A}|}} \|Q - \hat{h}_n(\cdot; Q)\|_{\mathcal{D}_n}^2. \quad (12)$$

For general spaces $\mathcal{F}^{|\mathcal{A}|}$, these optimization problems can be difficult to solve, but when $\mathcal{F}^{|\mathcal{A}|}$ is a linear subspace of $B(\mathcal{X} \times \mathcal{A})$, the minimization problem becomes computationally feasible.

Comparison of BRM and LSTD is noteworthy. The population version of LSTD loss minimizes the distance between Q and $\Pi_\nu T^\pi Q$, which is $\|Q - \Pi_\nu T^\pi Q\|_\nu^2$. Meanwhile, BRM minimizes another distance function that is the distance between $T^\pi Q$ and $\Pi_\nu T^\pi Q$ subtracted from the distance between Q and $T^\pi Q$, i.e., $\|Q - T^\pi Q\|_\nu^2 - \|\hat{h}_n(\cdot; Q) - T^\pi Q\|_\nu^2$. See [Figure 1a](#) for a pictorial presentation of these distances. When $\mathcal{F}^{|\mathcal{A}|}$ is linear, because of the Pythagorean theorem, the solution to the modified BRM (6) coincides with the LSTD solution (10) ([Antos et al., 2008b](#)).

4. Regularized Policy Iteration Algorithms

In this section we introduce two *Regularized Policy Iteration* algorithms, which are instances of the generic API algorithms. These algorithms are built on the regularized extensions of BRM ([Section 3.1](#)) and LSTD ([Section 3.2](#)) for the task of approximate policy evaluation.

The pseudo-code of the Regularized Policy Iteration algorithms is shown in [Algorithm 1](#). The algorithm receives K (the number of API iterations), an initial action-value function $\hat{Q}^{(K-1)}$, the function space $\mathcal{F}^{|\mathcal{A}|}$, the regularizer $J : \mathcal{F}^{|\mathcal{A}|} \rightarrow \mathbb{R}$, and a set of regularization coefficients $\{(\lambda_{Q_n}^{(k)}, \lambda_{h_n}^{(k)})\}_{k=0}^{K-1}$. Each iteration starts with a step of policy improvement, i.e.,

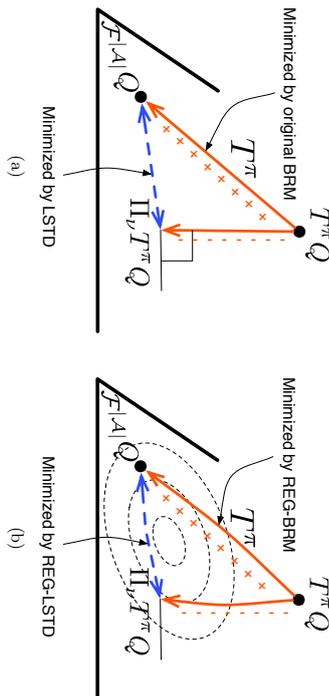


Figure 1: (a) This figure shows the loss functions minimized by the original BRM, the modified BRM, and the LSTD methods: The function space $\mathcal{F}^{|\mathcal{A}|}$ is represented by the plane. The Bellman operator T^π maps an action-value function $Q \in \mathcal{F}^{|\mathcal{A}|}$ to a function $T^\pi Q$. The function $T^\pi Q - \Pi_{\mathcal{F}^{|\mathcal{A}|}} T^\pi Q$ is orthogonal to $\mathcal{F}^{|\mathcal{A}|}$. The original BRM loss function is $\|Q - T^\pi Q\|_2^2$ (solid line), the modified BRM loss is $\|Q - T^\pi Q\|_2^2 - \|\Pi_{\mathcal{F}^{|\mathcal{A}|}} T^\pi Q - \Pi_{\mathcal{F}^{|\mathcal{A}|}} Q\|_2^2$ (the difference of two solid line segments; note the + and - symbols), and the LSTD loss is $\|Q - \Pi_{\mathcal{F}^{|\mathcal{A}|}} T^\pi Q\|_2^2$ (dashed line). LSTD and the modified BRM are equivalent for linear function spaces. (b) REG-LSTD and REG-BRM minimize regularized objective functions. Regularization makes the function $T^\pi Q - \Pi_{\mathcal{F}^{|\mathcal{A}|}} T^\pi Q$ to be non-orthogonal to $\mathcal{F}^{|\mathcal{A}|}$. The dashed ellipsoids represent the level-sets defined by the regularization functional J .

$\pi_k \leftarrow \hat{\pi}(\cdot; \hat{Q}^{(k-1)}) = \operatorname{argmax}_{\sigma \in \mathcal{A}} \hat{Q}^{(k-1)}(\cdot, \sigma)$. For the first iteration ($k=0$), one may ignore this step and provide an initial policy π_0 instead of $\hat{Q}^{(-1)}$. Afterwards, we have a data generating step: At each iteration $k=0, \dots, K-1$, the agent follows the data generating policy π_k to obtain $\mathcal{D}_n^{(k)} = \{(X_t^{(k)}, A_t^{(k)}, R_t^{(k)}, X_{t+1}^{(k)})\}_{1 \leq t \leq n}$. For the k th iteration of the algorithm, we use training samples $\mathcal{D}_n^{(k)}$ to evaluate policy π_k . In practice, one might want to change π_k at each iteration in such a way that the agent ultimately achieves a better performance. The relation between the performance and the choice of data samples, however, is complicated. For simplicity of analysis, in the rest of this work we assume that a fixed behavior policy is used in all iterations, i.e., $\pi_k = \pi_0$.⁷ This leads to K independent data sets $\mathcal{D}_n^{(0)}, \dots, \mathcal{D}_n^{(K-1)}$. From now on, to avoid clutter, we use symbols $\mathcal{D}_n, X_t, \dots$ instead of $\mathcal{D}_n^{(k)}, X_t^{(k)}, \dots$ with the understanding that each \mathcal{D}_n in various iterations is referring to an independent set of data samples, which should be clear from the context.

The approximate policy evaluation step is performed by REG-LSTD/BRM, which will be discussed shortly. REG-LSTD/BRM receives policy π_k , the training samples $\mathcal{D}_n^{(k)}$, the function space $\mathcal{F}^{|\mathcal{A}|}$, the regularizer J , and the regularization coefficients $(\lambda_{Q,n}^{(k)}, \lambda_{h,n}^{(k)})$, and

returns an estimate of the action-value function of policy π_k . This procedure repeats for K iterations.

REG-BRM approximately evaluates policy π_k by solving the following coupled optimization problems:

$$\hat{h}_n(\cdot; Q) = \operatorname{argmin}_{h \in \mathcal{F}^{|\mathcal{A}|}} \left[\|h - \hat{T}^{\pi_k} Q\|_{\mathcal{D}_n}^2 + \lambda_{h,n}^{(k)} J^2(h) \right], \quad (13)$$

$$\hat{Q}^{(k)} = \operatorname{argmin}_{Q \in \mathcal{F}^{|\mathcal{A}|}} \left[\|Q - \hat{T}^{\pi_k} Q\|_{\mathcal{D}_n}^2 - \|\hat{h}_n(\cdot; Q) - \hat{T}^{\pi_k} Q\|_{\mathcal{D}_n}^2 + \lambda_{Q,n}^{(k)} J^2(Q) \right], \quad (14)$$

where $J : \mathcal{F}^{|\mathcal{A}|} \rightarrow \mathbb{R}$ is the regularization functional (or simply regularizer or penalizer), and $\lambda_{h,n}^{(k)}, \lambda_{Q,n}^{(k)} > 0$ are regularization coefficients. The regularizer can be any pseudo-norm defined on $\mathcal{F}^{|\mathcal{A}|}$, and \mathcal{D}_n is defined as (2).⁸ The regularizer is often chosen such that the functions that we believe are more “complex” have larger values of J . The notion of complexity, however, is subjective and depends on the choice of $\mathcal{F}^{|\mathcal{A}|}$ and J . Finally note that we call $J(Q)$ the smoothness of Q , even though it might not coincide with the conventional derivative-based notions of smoothness.

An example of the case that J has a derivative-based interpretation is when the function space $\mathcal{F}^{|\mathcal{A}|}$ is a Sobolev space and the regularizer J is defined as its corresponding norm. In this case, we are penalizing the weak-derivatives of the estimate (Györfi et al., 2002; van de Geer, 2000). One can generalize the notion of smoothness beyond the usual derivative-based ones (cf. Chapter 1 of Triebel 2006) and define function spaces such as the family of Besov spaces (Devore, 1998). The RKHS norm for shift-invariant and radial kernels can also be interpreted as a penalizer of higher-frequency terms of the function (i.e., a low-pass filter Evgenion et al. 1999), so they effectively encourage “smoother” functions. The choice of kernel determines the frequency response of the filter. One may also use other data-dependent regularizers such as manifold regularization (Balkin et al., 2006) and Sample-based Approximate Regularization (Bachman et al., 2014). As a final example, for the functions in the form of $Q(x; a) = \sum_{i \geq 1} \phi_i(x; a) w_i$, if we choose a sparsity-inducing regularizer such as $J(Q) \triangleq \sum_{i \geq 1} |w_i|$ as the measure of smoothness, then a function that has a sparse representation in the dictionary $\{\phi_i\}_{i \geq 1}$ is, by definition, a smooth function—even though there is not necessarily any connection to the derivative-based smoothness.

REG-LSTD approximately evaluates the policy π_k by solving the following coupled optimization problems:

$$\hat{h}_n(\cdot; Q) = \operatorname{argmin}_{h \in \mathcal{F}^{|\mathcal{A}|}} \left[\|h - \hat{T}^{\pi_k} Q\|_{\mathcal{D}_n}^2 + \lambda_{h,n}^{(k)} J^2(h) \right], \quad (15)$$

$$\hat{Q}^{(k)} = \operatorname{argmin}_{Q \in \mathcal{F}^{|\mathcal{A}|}} \left[\|Q - \hat{h}_n(\cdot; Q)\|_{\mathcal{D}_n}^2 + \lambda_{Q,n}^{(k)} J^2(Q) \right]. \quad (16)$$

Note that the difference between (7)-(8) ((11)-(12)) and (13)-(14) ((15)-(16)) is the addition of the regularizers $J^2(h)$ and $J^2(Q)$.

Unlike the non-regularized case described in Section 3, the solutions of REG-BRM and REG-LSTD are not the same. As a result of the *regularized* projection, (13) and

⁷ So we are in the so-called *off-policy sampling* scenario.

(15), the function $\hat{h}_n(\cdot; Q) - \hat{T}^{\pi_k} Q$ is not orthogonal to the function space $\mathcal{F}^{|\mathcal{A}|}$ —even if $\mathcal{F}^{|\mathcal{A}|}$ is a linear space. Therefore, the Pythagorean theorem is not applicable anymore: $\|Q - \hat{h}_n(\cdot; Q)\|^2 \neq \|Q - \hat{T}^{\pi_k} Q\|^2 - \|\hat{h}_n(\cdot; Q) - \hat{T}^{\pi_k} Q\|^2$ (See Figure 1b).

One may ask why we have regularization terms in both optimization problems, as opposed to only in the projection term (15) (similar to the Lasso-TD algorithm [Kolter and Ng 2009](#); [Ghavamzadeh et al. 2011](#)) or only in (16) (similar to [Geist and Scherrer 2012](#); [Ávila Pires and Szepesvári 2012](#)). We discuss this question in Section 4.1. Briefly speaking, for large function spaces such as the Sobolev spaces or the RKHS with universal kernels, if we remove the regularization term in (15), the coupled optimization problems reduces to (unmodified) BRM, which is biased as discussed earlier; whereas if the regularization term in (16) is removed, the solution can be arbitrary bad due to overfitting.

Finally note that the choice of the function space $\mathcal{F}^{|\mathcal{A}|}$, the regularizer J , and the regularization coefficients $\lambda_{Q,n}^{(k)}$ and $\lambda_{h,n}^{(k)}$ all affect the sample efficiency of the algorithms. If one knew $J(Q^\pi)$, the regularization coefficients could be chosen optimally. Nonetheless, the value of $J(Q^\pi)$ is often not known, so one has to use a model selection procedure to set the best function space and the regularization coefficients. The situation is similar to the problem of model selection in supervised learning (though the solutions are different). After developing some tools necessary for discussing this issue in Section 5, we return to the problem of choosing the regularization coefficients after Theorem 11 as well as in Section 6.

Remark 7 *To the best of our knowledge, [Antos et al. \(2008b\)](#) were the first who explicitly considered LSTD as the optimizer of the loss function (9). Their discussion was mainly to prove the equivalence of modified BRM (5) and LSTD when $\mathcal{F}^{|\mathcal{A}|}$ is a linear function space. In their work, the loss function is not used to derive any new algorithm. [Farahmand et al. \(2009b\)](#) used this loss function to develop the regularized variant of LSTD (15)-(16). This loss function was later called mean-square projected Bellman error by [Sutton et al. \(2009\)](#), and was used to derive the GTD2 and TDC algorithms.*

4.1 Why Two Regularizers?

We discuss why using regularizers in both optimization problems (15) and (16) of REG-LSTD is necessary for large function spaces such as the Sobolev spaces and the RKHS with universal kernels. Here we show that for large function spaces, depending on which regularization term we remove, either the coupled optimization problems reduces to the regularized variant of the unmodified BRM, which has a bias, or the solution can be arbitrary bad.

Let us focus on REG-LSTD for a given policy π . Assume that the function space $\mathcal{F}^{|\mathcal{A}|}$ is rich enough in the sense that it is dense in the space of continuous functions w.r.t. the supremum norm. This is satisfied by many large function spaces such as RKHS with universal kernels (Definition 4.52 of [Steinwart and Christmann 2008](#)) and the Sobolev spaces on compact domains. We consider what would happen if instead of the current formulation of REG-LSTD (15)-(16), we only used a regularizer either in the first or second optimization problem. We study each case separately. For notational simplicity, we omit the dependence on the iteration number k .

Case 1. In this case, we only regularize the empirical error $\|Q - \hat{h}_n(\cdot; Q)\|_{\mathcal{D}_n}^2$, but we do not regularize the projection, i.e.,

$$\begin{aligned} \hat{h}_n(\cdot; Q) &= \operatorname{argmin}_{h \in \mathcal{F}^{|\mathcal{A}|}} \|h - \hat{T}^{\pi} Q\|_{\mathcal{D}_n}^2, \\ \hat{Q} &= \operatorname{argmin}_{Q \in \mathcal{F}^{|\mathcal{A}|}} \left[\|Q - \hat{h}_n(\cdot; Q)\|_{\mathcal{D}_n}^2 + \lambda_{Q,n} J^2(Q) \right]. \end{aligned} \quad (17)$$

When the function space $\mathcal{F}^{|\mathcal{A}|}$ is rich enough, there exists a function $\hat{h}_n \in \mathcal{F}^{|\mathcal{A}|}$ that fits perfectly well to its target values at data points $\{(X_i, A_i)\}_{i=1}^n$, that is, $\hat{h}_n(X_i, A_i; Q) = (\hat{T}^{\pi} Q)(X_i, A_i)$ for $i = 1, \dots, n$.⁹ Such a function is indeed the minimizer of the loss $\|Q - \hat{h}_n(\cdot; Q)\|_{\mathcal{D}_n}^2$. The second optimization problem (17) becomes

$$\hat{Q} = \operatorname{argmin}_{Q \in \mathcal{F}^{|\mathcal{A}|}} \left[\|Q - \hat{T}^{\pi} Q\|_{\mathcal{D}_n}^2 + \lambda_{Q,n} J^2(Q) \right].$$

This is the regularized version of the original (i.e., unmodified) formulation of the BRM algorithm. As discussed in Section 3.1, the unmodified BRM algorithm is biased when the MDP is not deterministic. Adding a regularizer does not solve the biasedness problem of the unmodified BRM loss. So without regularizing the first optimization problem, the function \hat{h}_n overfits to the noise and as a result the whole algorithm becomes incorrect.

Case 2. In this case, we only regularize the empirical projection $\|h - \hat{T}^{\pi} Q\|_{\mathcal{D}_n}^2$, but we do not regularize $\|Q - \hat{h}_n(\cdot; Q)\|_{\mathcal{D}_n}^2$, i.e.,

$$\begin{aligned} \hat{h}_n(\cdot; Q) &= \operatorname{argmin}_{h \in \mathcal{F}^{|\mathcal{A}|}} \left[\|h - \hat{T}^{\pi} Q\|_{\mathcal{D}_n}^2 + \lambda_{h,n} J^2(h) \right], \\ \hat{Q} &= \operatorname{argmin}_{Q \in \mathcal{F}^{|\mathcal{A}|}} \|Q - \hat{h}_n(\cdot; Q)\|_{\mathcal{D}_n}^2. \end{aligned} \quad (18)$$

For a fixed Q , the first optimization problem is the standard regularized regression estimator with the regression function $\mathbb{E}[(\hat{T}^{\pi} Q)(X, A) | X = x, A = a] = (T^{\pi} Q)(x, a)$. Therefore, if the function space $\mathcal{F}^{|\mathcal{A}|}$ is rich enough and we set the regularization coefficient $\lambda_{h,n}$ properly, $\|h - T^{\pi} Q\|_{\nu}$ and $\|h - \hat{T}^{\pi} Q\|_{\mathcal{D}_n}$ go to zero as the sample size grows (the rate of convergence depends on the complexity of the target function; cf. Lemma 15 and Theorem 16). So we can expect $\hat{h}_n(\cdot; Q)$ to get closer to $T^{\pi} Q$ as the sample size grows.

For simplicity of discussion, suppose that we are in the ideal situation where for any Q , we have $\hat{h}_n(\cdot; Q) = (T^{\pi} Q)(x, a)$ for all $(x, a) \in \{(X_i, A_i)\}_{i=1}^n \cup \{(X'_i, \pi(X'_i))\}_{i=1}^n$, that

9. To be more precise: First, for an $\varepsilon > 0$, we construct a continuous function $\tilde{h}_\varepsilon(z) = \sum_{z_i \in \{(X_i, A_i)\}_{i=1}^n} \max\left\{1 - \frac{\|z - z_i\|}{\varepsilon}, 0\right\} (\hat{T}^{\pi} Q)(z_i)$. We then use the denseness of the function space $\mathcal{F}^{|\mathcal{A}|}$ in the supremum norm to argue that there exists $h_\varepsilon \in \mathcal{F}^{|\mathcal{A}|}$ such that $\|h_\varepsilon - \tilde{h}_\varepsilon\|_\infty$ is arbitrarily close to zero. So when $\varepsilon \rightarrow 0$, the value of function h_ε is arbitrarily close to $T^{\pi} Q$ at data points. We then choose $\hat{h}_n(\cdot; Q) = h_\varepsilon$. This construction is similar to Theorem 2 of [Nadler et al. \(2009\)](#). See also the argument in Case 2 for more detail.

is, we precisely know $T^\pi Q$ at all data points.¹⁰ Substituting this $\hat{h}_n((x, a), Q)$ in the second optimization problem (17), we get that we are solving the following optimization problem:

$$\hat{Q} = \operatorname{argmin}_{Q \in \mathcal{F}^{\mathcal{A}}} \|Q - T^\pi Q\|_{\mathcal{D}_n}^2. \quad (19)$$

This is the Bellman error minimization problem. We do not have the biasedness problem here as we have $T^\pi Q$ instead of $\hat{T}^\pi Q$ in the loss. Nonetheless, we face another problem: Minimizing this empirical risk minimization without controlling the complexity of the function space might lead to an overfitted solution, very similar to the same phenomenon in supervised learning.

To see it more precisely, we first construct a continuous function

$$\bar{Q}_\varepsilon(z) = \sum_{Z_i \in (\{X_i, A_i\})_{i=1}^n \cup (\{X_i^\dagger, \pi(X_i)\})_{i=1}^n} \max \left\{ 1 - \frac{\|z - Z_i\|}{\varepsilon}, 0 \right\} Q^\pi(Z_i),$$

which for small enough $\varepsilon > 0$ has the property that $\|\bar{Q}_\varepsilon - T^\pi \bar{Q}_\varepsilon\|_{\mathcal{D}_n}^2$ is zero, i.e., it is a minimizer of the empirical loss. Due to the denseness of $\mathcal{F}^{\mathcal{A}}$, we can find a $Q_\varepsilon \in \mathcal{F}^{\mathcal{A}}$ that is arbitrarily close to the continuous function \bar{Q}_ε . Therefore, for small enough ε , the function Q_ε is a minimizer of (19), i.e., the value of $\|Q_\varepsilon - T^\pi Q_\varepsilon\|_{\mathcal{D}_n}^2$ is zero. But Q_ε is not a good approximation of Q^π because Q_ε consists of spikes in the ε -neighbourhood of data points and zero elsewhere. In other words, Q_ε does not generalize well beyond the data points when ε is chosen to be small.

Of course the solution is to control the complexity of $\mathcal{F}^{\mathcal{A}}$ so that spiky functions such as Q_ε are not selected as the solution of the optimization problem. When we regularize both optimization problems, as we do in this work, none of these problems happen.

This argument applies to rich function spaces that can approximate any reasonably complex functions (e.g., continuous functions) arbitrarily well. If the function space $\mathcal{F}^{\mathcal{A}}$ is much more limited, for example if it is a parametric function space, we *may* not need to regularize both optimization problems. An example of such an approach for parametric spaces has been analyzed by [Avila Pires and Szepesvári \(2012\)](#).

4.2 Closed-Form Solutions

In this section we provide a closed-form solution for (13)-(14) and (15)-(16) for two cases: 1) When $\mathcal{F}^{\mathcal{A}}$ is a finite dimensional linear space and $J^2(\cdot)$ is defined as the weighted squared sum of parameters describing the function (a setup similar to the ridge regression [Hoerl and Kennard 1970](#)) and 2) $\mathcal{F}^{\mathcal{A}}$ is an RKHS and $J(\cdot)$ is the corresponding inner-product norm, i.e., $J^2(\cdot) = \|\cdot\|_H^2$. Here we use a generic π and \mathcal{D}_n instead of π_k and $\mathcal{D}_n^{(k)}$ at the k th iteration.

¹⁰. This is an ideal situation because 1) $\|h - \hat{T}^\pi Q\|_v$ is equal to zero only asymptotically and not in finite samples regime, and 2) even if $\|h - \hat{T}^\pi Q\|_v = 0$, it does not imply that $h_n(x, a; Q) = (T^\pi Q)(x, a)$ almost surely on $\mathcal{X} \times \mathcal{A}$. Nonetheless, these simplifications are only in favour of the algorithm considered in this case, so for simplicity of discussion we assume that they hold.

4.2.1 A PARAMETRIC FORMULATION FOR REG-BRM AND REG-LSTD

In this section we consider the case when h and Q are both given as linear combinations of some basis functions:

$$h(\cdot) = \phi(\cdot)^\top \mathbf{u}, \quad Q(\cdot) = \phi(\cdot)^\top \mathbf{w}, \quad (20)$$

where $\mathbf{u}, \mathbf{w} \in \mathbb{R}^p$ are parameter vectors and $\phi(\cdot) \in \mathbb{R}^p$ is a vector of p linearly independent basis functions defined over the space of state-action pairs.¹¹ These basis functions might be predefined (e.g., Fourier ([Kondratiev et al., 2011](#)) or wavelets) or constructed data-dependently by one of already mentioned feature generation methods. We further assume that the regularization terms take the form

$$J^2(h) = \mathbf{u}^\top \Psi \mathbf{u}, \\ J^2(Q) = \mathbf{w}^\top \Psi \mathbf{w}.$$

for some user-defined choice of positive definite matrix $\Psi \in \mathbb{R}^{p \times p}$. A simple and common choice would be $\Psi = \mathbf{I}$. Define $\Phi, \Phi' \in \mathbb{R}^{n \times p}$ and $\mathbf{r} \in \mathbb{R}^n$ as follows:

$$\Phi = \left(\phi(Z_1), \dots, \phi(Z_n) \right)^\top, \quad \Phi' = \left(\phi(Z'_1), \dots, \phi(Z'_n) \right)^\top, \quad \mathbf{r} = \left(R_1, \dots, R_n \right)^\top, \quad (21)$$

with $Z_i = (X_i, A_i)$ and $Z'_i = (X'_i, \pi(X'_i))$.

The solution to REG-BRM is given by the following proposition.

Proposition 8 (Closed-form solution for REG-BRM) *Under the setting of this section, the approximate action-value function returned by REG-BRM is $\hat{Q}(\cdot) = \phi(\cdot)^\top \mathbf{w}^*$, where*

$$\mathbf{w}^* = \left[\mathbf{B}^\top \mathbf{B} - \gamma^2 \mathbf{C}^\top \mathbf{C} + n \lambda_{Q,n} \Psi \right]^{-1} \left(\mathbf{B}^\top + \gamma \mathbf{C}^\top (\Phi \mathbf{A} - \mathbf{D}) \right) \mathbf{r},$$

with $\mathbf{A} = (\Phi^\top \Phi + n \lambda_{h,n} \Psi)^{-1} \Phi^\top$, $\mathbf{B} = \Phi - \gamma \Phi'$, $\mathbf{C} = (\Phi \mathbf{A} - \mathbf{D}) \Phi'$.

Proof Using (20) and (21), we can rewrite (13)-(14) as

$$\mathbf{u}^*(\mathbf{w}) = \operatorname{argmin}_{\mathbf{u} \in \mathbb{R}^p} \left\{ \frac{1}{n} [\Phi \mathbf{u} - (\mathbf{r} + \gamma \Phi' \mathbf{w})]^\top [\Phi \mathbf{u} - (\mathbf{r} + \gamma \Phi' \mathbf{w})] + \lambda_{h,n} \mathbf{u}^\top \Psi \mathbf{u} \right\}, \quad (22)$$

$$\mathbf{w}^* = \operatorname{argmin}_{\mathbf{w} \in \mathbb{R}^p} \left\{ \frac{1}{n} [\Phi \mathbf{w} - (\mathbf{r} + \gamma \Phi' \mathbf{w})]^\top [\Phi \mathbf{w} - (\mathbf{r} + \gamma \Phi' \mathbf{w})] - \frac{1}{n} [\Phi \mathbf{u}^*(\mathbf{w}) - (\mathbf{r} + \gamma \Phi' \mathbf{w})]^\top [\Phi \mathbf{u}^*(\mathbf{w}) - (\mathbf{r} + \gamma \Phi' \mathbf{w})] + \lambda_{Q,n} \mathbf{w}^\top \Psi \mathbf{w} \right\}. \quad (23)$$

Taking the derivative of (22) w.r.t. \mathbf{u} and equating it to zero, we obtain \mathbf{u}^* as a function of \mathbf{w} :

$$\mathbf{u}^*(\mathbf{w}) = \left(\Phi^\top \Phi + n \lambda_{h,n} \Psi \right)^{-1} \Phi^\top (\mathbf{r} + \gamma \Phi' \mathbf{w}) = \mathbf{A} (\mathbf{r} + \gamma \Phi' \mathbf{w}). \quad (24)$$

Plug $\mathbf{u}^*(\mathbf{w})$ from (24) into (23), take the derivative w.r.t. \mathbf{w} and equate it to zero to obtain the parameter vector \mathbf{w}^* as announced above. ■

The solution returned by REG-LSTD is given in the following proposition.

¹¹. At the cost of using generalized inverses, everything in this section extends to the case when the basis functions are not linearly independent.

Proposition 9 (Closed-form solution for REG-LSTD) Under the setting of this section, the approximate action-value function returned by REG-LSTD is $\hat{Q}(\cdot) = \phi(\cdot)^\top \mathbf{w}^*$, where

$$\mathbf{w}^* = [\mathbf{E}^\top \mathbf{E} + n\lambda_{Q,n} \mathbf{\Psi}]^{-1} \mathbf{E}^\top \mathbf{A} \mathbf{r},$$

with $\mathbf{A} = (\mathbf{\Phi}^\top \mathbf{\Phi} + n\lambda_{h,n} \mathbf{\Psi})^{-1} \mathbf{\Phi}^\top$ and $\mathbf{E} = (\mathbf{\Phi} - \gamma \mathbf{A} \mathbf{\Phi}')$.

Proof Using (20) and (21), we can rewrite (15)-(16) as

$$\mathbf{u}^*(\mathbf{w}) = \operatorname{argmin}_{\mathbf{u} \in \mathbb{R}^p} \left\{ \frac{1}{n} [\mathbf{\Phi} \mathbf{u} - (\mathbf{r} + \gamma \mathbf{\Phi}' \mathbf{w})]^\top [\mathbf{\Phi} \mathbf{u} - (\mathbf{r} + \gamma \mathbf{\Phi}' \mathbf{w})] + \lambda_{h,n} \mathbf{u}^\top \mathbf{\Psi} \mathbf{u} \right\}, \quad (25)$$

$$\mathbf{w}^* = \operatorname{argmin}_{\mathbf{w} \in \mathbb{R}^p} \left\{ [\mathbf{\Phi} \mathbf{w} - \mathbf{\Phi} \mathbf{u}^*(\mathbf{w})]^\top [\mathbf{\Phi} \mathbf{w} - \mathbf{\Phi} \mathbf{u}^*(\mathbf{w})] + \lambda_{Q,n} \mathbf{w}^\top \mathbf{\Psi} \mathbf{w} \right\}. \quad (26)$$

Similar to the parametric REG-BRM, we solve (25) and obtain $\mathbf{u}^*(\mathbf{w})$ which is the same as (24). If we plug this $\mathbf{u}^*(\mathbf{w})$ into (26), take derivative w.r.t. \mathbf{w} , and find the minimizer, the parameter vector \mathbf{w}^* will be as announced. \blacksquare

4.2.2 RKHS FORMULATION FOR REG-BRM AND REG-LSTD

The class of reproducing kernel Hilbert spaces provides a flexible and powerful family of function spaces to choose $\mathcal{F}^{|\mathcal{A}|}$ from. An RKHS $\mathcal{H} : \mathcal{X} \times \mathcal{A} \rightarrow \mathbb{R}$ is defined by a positive definite kernel $\kappa : (\mathcal{X} \times \mathcal{A}) \times (\mathcal{X} \times \mathcal{A}) \rightarrow \mathbb{R}$. With such a choice, we can use the corresponding squared RKHS norm $\|\cdot\|_{\mathcal{H}}^2$ as the regularizer $J^2(\cdot)$. REG-BRM with an RKHS function space $\mathcal{F}^{|\mathcal{A}|} = \mathcal{H}$ would be

$$\hat{h}_n(\cdot; Q) = \operatorname{argmin}_{h \in \mathcal{F}^{|\mathcal{A}|} = \mathcal{H}} \left[\|h - \hat{T}^\pi Q\|_{\mathcal{D}_n}^2 + \lambda_{h,n} \|h\|_{\mathcal{H}}^2 \right], \quad (27)$$

$$\hat{Q} = \operatorname{argmin}_{Q \in \mathcal{F}^{|\mathcal{A}|} = \mathcal{H}} \left[\|Q - \hat{T}^\pi Q\|_{\mathcal{D}_n}^2 - \|\hat{h}_n(\cdot; Q) - \hat{T}^\pi Q\|_{\mathcal{D}_n}^2 + \lambda_{Q,n} \|Q\|_{\mathcal{H}}^2 \right], \quad (28)$$

and the coupled optimization problems for REG-LSTD are

$$\hat{h}_n(\cdot; Q) = \operatorname{argmin}_{h \in \mathcal{F}^{|\mathcal{A}|} = \mathcal{H}} \left[\|h - \hat{T}^\pi Q\|_{\mathcal{D}_n}^2 + \lambda_{h,n} \|h\|_{\mathcal{H}}^2 \right], \quad (29)$$

$$\hat{Q} = \operatorname{argmin}_{Q \in \mathcal{F}^{|\mathcal{A}|} = \mathcal{H}} \left[\|Q - \hat{h}_n(\cdot; Q)\|_{\mathcal{D}_n}^2 + \lambda_{Q,n} \|Q\|_{\mathcal{H}}^2 \right]. \quad (30)$$

We can solve these coupled optimization problems by the application of the generalized representer theorem for RKHS (Schölkopf et al., 2001). The result, which is stated in the next theorem, shows that the infinite dimensional optimization problem defined on $\mathcal{F}^{|\mathcal{A}|} = \mathcal{H}$ boils down to a finite dimensional problem with the dimension twice the number of data points.

Theorem 10 Let \tilde{Z} be a vector defined as $\tilde{Z} = (Z_1, \dots, Z_n, Z'_1, \dots, Z'_n)^\top$. Then the optimizer $\hat{Q} \in \mathcal{H}$ of (27)-(28) can be written as $\hat{Q}(\cdot) = \sum_{i=1}^{2n} \tilde{\alpha}_i \kappa(\tilde{Z}_i, \cdot)$ for some values of

$\tilde{\alpha} \in \mathbb{R}^{2n}$. The same holds for the solution to (29)-(30). Further, the coefficient vectors can be obtained in the following form:

$$\begin{aligned} \text{REG-BRM:} \quad \tilde{\alpha}_{BRM} &= (\mathbf{C} \mathbf{K}_Q + n\lambda_{Q,n} \mathbf{I})^{-1} (\mathbf{D}^\top + \gamma \mathbf{C}_2^\top \mathbf{B}) \mathbf{r}, \\ \text{REG-LSTD:} \quad \tilde{\alpha}_{LSTD} &= (\mathbf{F}^\top \mathbf{F} \mathbf{K}_Q + n\lambda_{Q,n} \mathbf{I})^{-1} \mathbf{F}^\top \mathbf{E} \mathbf{r}, \end{aligned}$$

where $\mathbf{r} = (R_1, \dots, R_n)^\top$ and the matrices $\mathbf{K}_Q, \mathbf{B}, \mathbf{C}, \mathbf{C}_2, \mathbf{D}, \mathbf{E}, \mathbf{F}$ are defined as follows: $\mathbf{K}_h \in \mathbb{R}^{n \times n}$ is defined as $[\mathbf{K}_h]_{ij} = \kappa(Z_i, Z_j)$, $1 \leq i, j \leq n$, and $\mathbf{K}_Q \in \mathbb{R}^{2n \times 2n}$ is defined as $[\mathbf{K}_Q]_{ij} = \kappa(\tilde{Z}_i, \tilde{Z}_j)$, $1 \leq i, j \leq 2n$. Let $\mathbf{C}_1 = (\mathbf{I}_{n \times n} \quad \mathbf{0}_{n \times n})$ and $\mathbf{C}_2 = (\mathbf{0}_{n \times n} \quad \mathbf{I}_{n \times n})$. Denote $\mathbf{D} = \mathbf{C}_1 - \gamma \mathbf{C}_2$, $\mathbf{E} = \mathbf{K}_h (\mathbf{K}_h + n\lambda_{h,n} \mathbf{I})^{-1}$, $\mathbf{F} = \mathbf{C}_1 - \gamma \mathbf{E} \mathbf{C}_2$, $\mathbf{B} = \mathbf{K}_h (\mathbf{K}_h + n\lambda_{h,n} \mathbf{I})^{-1} - \mathbf{I}$, and $\mathbf{C} = \mathbf{D}^\top \mathbf{D} - \gamma^2 (\mathbf{B} \mathbf{C}_2)^\top (\mathbf{B} \mathbf{C}_2)$.

Proof See Appendix A. \blacksquare

5. Theoretical Analysis

In this section, we analyze the statistical properties of REG-LSPI and provide a finite-sample upper bound on the performance loss $\|Q^* - Q^{\pi_K}\|_{1,\rho}$. Here, π_K is the policy greedy w.r.t. $\hat{Q}^{(K-1)}$ and ρ is the performance evaluation measure. The distribution ρ is chosen by the user and is often different from the sampling distribution ν .

Our study has two main parts. First, we analyze the policy evaluation error of REG-LSTD in Section 5.1. We suppose that given any policy π , we obtain \hat{Q} by solving (15)-(16) with π_k in these equations being replaced by π . Theorem 11 provides an upper bound on the Bellman error $\|\hat{Q} - T^\pi \hat{Q}\|_{\nu}$. We discuss the optimality of this upper bound for policy evaluation for some general classes of function spaces. We show that the result is not only optimal in its convergence rate, but also in its dependence on $J(Q^\pi)$. After that in Section 5.2, we show how the Bellman errors of the policy evaluation procedure propagate through the API procedure (Theorem 13). The main result of this paper, which is an upper bound on the performance loss $\|Q^* - Q^{\pi_K}\|_{1,\rho}$, is stated as Theorem 14 in Section 5.3, followed by its discussion. We compare this work's statistical guarantee with some other papers' in Section 5.3.1.

To analyze the statistical performance of the REG-LSPI procedure, we make the following assumptions. We discuss their implications and the possible relaxations after stating each of them.

Assumption A1 (MDP Regularity) The set of states \mathcal{X} is a compact subset of \mathbb{R}^d . The random immediate rewards $R_t \sim \mathcal{R}(\cdot | X_t, A_t)$ ($t = 1, 2, \dots$) as well as the expected immediate rewards $r(x, a)$ are uniformly bounded by R_{\max} , i.e., $|R_t| \leq R_{\max}$ ($t = 1, 2, \dots$) and $\|r\|_\infty \leq R_{\max}$.

Even though the algorithms were presented for a general measurable state space \mathcal{X} , the theoretical results are stated for the problems whose state space is a compact subset of \mathbb{R}^d . Generalizing Assumption A1 to other state spaces should be possible under certain regularity conditions. One example could be any Polish space, i.e., separable completely

metrizable topological space. Nevertheless, we do not investigate such generalizations here. The boundedness of the rewards is a reasonable assumption that can be replaced by a more relaxed condition such as its sub-Gaussianity (Vershynin, 2012; van de Geer, 2000). This relaxation, however, increases the technicality of the proofs without adding much to the intuition. We remark on the compactness assumption after stating Assumption A4.

Assumption A2 (Sampling) At iteration k of REG-LSPI (for $k = 0, \dots, K - 1$), n fresh independent and identically distributed (i.i.d.) samples are drawn from distribution $\nu \in \mathcal{M}(\mathcal{X} \times \mathcal{A})$, i.e., $\mathcal{D}_n^{(k)} = \left\{ \left(Z_t^{(k)}, R_t^{(k)}, X_t^{(k)} \right)_{t=1}^n \text{ with } Z_t^{(k)} = (X_t^{(k)}, A_t^{(k)}) \stackrel{\text{i.i.d.}}{\sim} \nu \text{ and } X_t^{(k)} \sim P(\cdot | X_t^{(k)}, A_t^{(k)}) \right\}$.

The i.i.d. requirement of Assumption A2 is primarily used to simplify the proofs. With much extra effort, these results can be extended to the case when the data samples belong to a single trajectory generated by a fixed policy. In the single trajectory scenario, samples are not independent anymore, but under certain conditions on the Markov process, the process (X_t, A_t) gradually “forgets” its past. One way to quantify this forgetting is through mixing processes. For these processes, tools such as the *independent blocks* technique (Yu, 1994; Donkhan, 1994) or information theoretical inequalities (Samson, 2000) can be used to carry on the analysis—as have been done by Amos et al. (2008b) in the API context, by Farahmand and Szepesvári (2012) for analyzing the regularized regression problem, and by Farahmand and Szepesvári (2011) in the context of model selection for RL problems.

It is worthwhile to emphasize that we do not require that the distribution ν to be known. The sampling distribution is also generally different from the distribution induced by the target policy π_* . For example, it might be generated by drawing state samples from a given $\nu_{\mathcal{X}}$ and choosing actions according to a behavior policy π , which is different from the policy being evaluated. So we are in the off-policy sampling setting. Moreover, changing ν at each iteration based on the previous iterations is a possibility with potential practical benefits, which has theoretical justifications in the context of imitation learning (Ross et al., 2011). For simplicity of the analysis, however, we assume that ν is fixed in all iterations. Finally, we note that the proofs work fine if we reuse the same data sets in all iterations. We comment on it later after the proof of Theorem 11 in Appendix B.

Assumption A3 (Regularizer) Define two regularization functionals $J : B(\mathcal{X}) \rightarrow \mathbb{R}$ and $J : B(\mathcal{X} \times \mathcal{A}) \rightarrow \mathbb{R}$ that are pseudo-norms on \mathcal{F} and $\mathcal{F}^{\mathcal{A}}$, respectively.¹² For all $Q \in \mathcal{F}^{\mathcal{A}}$ and $a \in \mathcal{A}$, we have $J(Q(\cdot, a)) \leq J(Q)$.

The regularizer $J(Q)$ measures the complexity of an action-value function Q . The functions that are more complex have larger values of $J(Q)$. We also need to define a related regularizer for value functions $Q(\cdot, a)$ ($a \in \mathcal{A}$). The latter regularizer is not explicitly used in the algorithm, and is only used in the analysis. This assumption imposes some mild restrictions on these regularization functionals. The condition that the regularizers be pseudo-norms is satisfied by many commonly-used regularizers such as the Sobolev norms,

¹² Note that here we are slightly abusing the notations as the same symbol is used for the regularizer over both $B(\mathcal{X})$ and $B(\mathcal{X} \times \mathcal{A})$. However, this should not cause any confusion since in any specific expression the identity of the regularizer should always be clear from the context.

the RKHS norms, and the l_2 -regularizer defined in Section 4.2.1 with a positive semi-definite choice of matrix Ψ . Moreover, the condition $J(Q(\cdot, a)) \leq J(Q)$ essentially states that the complexity of Q should upper bound the complexity of $Q(\cdot, a)$ for all $a \in \mathcal{A}$. If the regularizer $J : B(\mathcal{X} \times \mathcal{A}) \rightarrow \mathbb{R}$ is derived from a regularizer $J' : B(\mathcal{X}) \rightarrow \mathbb{R}$ through $J(Q) = \|(J'(Q(\cdot, a)))_{a \in \mathcal{A}}\|_p$ for some $p \in [1, \infty]$, then J will satisfy the second part of the assumption. From a computational perspective, a natural choice for RKHS is to choose $p = 2$ and to define $J^2(Q) = \sum_{a \in \mathcal{A}} \|Q(\cdot, a)\|_{\mathcal{H}_a}^2$ for \mathcal{H} being the RKHS defined on \mathcal{X} .

Assumption A4 (Capacity of Function Space) For $R > 0$, let $\mathcal{F}_R = \{f \in \mathcal{F} : J(f) \leq R\}$. There exist constants $C > 0$ and $0 < \alpha < 1$ such that for any $u, R > 0$ the following metric entropy condition is satisfied:

$$\log \mathcal{N}_\infty(u, \mathcal{F}_R) \leq C \left(\frac{R}{u} \right)^{2\alpha}.$$

This assumption characterizes the capacity of the ball with radius R in \mathcal{F} . The value of α is an essential quantity in our upper bounds. The metric entropy is precisely defined in Appendix G, but roughly speaking it is the logarithm of the minimum number of balls with radius u that are required to completely cover a ball with radius R in \mathcal{F} . This is a measure of complexity of a function space as it is more difficult to estimate a function when the metric entropy grows fast when u decreases. As a simple example, when the function space is finite, we effectively need to have good estimate of $|\mathcal{F}|$ functions in order not to choose the wrong one. In this case, $\mathcal{N}_\infty(u, \mathcal{F}_R)$ can be replaced by $|\mathcal{F}|$, so $\alpha = 0$ and $C = \log |\mathcal{F}|$. When the state space \mathcal{X} is finite and all functions are bounded by Q_{\max} , we have $\log \mathcal{N}_\infty(u, \mathcal{F}_R) \leq \log \mathcal{N}_\infty(u, \mathcal{F}) = |\mathcal{X}| \log \left(\frac{2Q_{\max}}{u} \right)$. This shows that the metric entropy for problems with finite state spaces grows much slower than what we consider here. Assumption A4 is suitable for large function spaces and is indeed satisfied for the Sobolev spaces and various RKHS. Refer to van de Geer (2000); Zhou (2002, 2003); Steinwart and Christmann (2008) for many examples.

An alternative assumption would be to have a similar metric entropy for the balls in $\mathcal{F}^{\mathcal{A}}$ (instead of \mathcal{F}). This would slightly change a few steps of the proofs, but leave the results essentially the same. Moreover, it makes the requirement that $J(Q(\cdot, a)) \leq J(Q)$ in Assumption A3 unnecessary. Nevertheless, as results on the capacity of \mathcal{F} is more common in the statistical learning theory literature, we stick to the combination of Assumptions A3 and A4.

The metric entropy here is defined w.r.t. the supremum norm. All proofs, except that of Lemma 23, only require the same bound to hold when the supremum norm is replaced by the more relaxed empirical L_2 -norm, i.e., those results require that there exist constants $C > 0$ and $0 < \alpha < 1$ such that for any $u, R > 0$ and all $x_1, \dots, x_n \in \mathcal{X}$, we have $\log \mathcal{N}_2(u, \mathcal{F}_R, x_{1:n}) \leq C \left(\frac{R}{u} \right)^{2\alpha}$. Of course, the metric entropy w.r.t. the supremum norm implies the one with the empirical norm. It is an interesting question to relax the supremum norm assumption in Lemma 23.

We can now remark on the requirement that \mathcal{X} is compact (Assumption A1). We stated that requirement mainly because most of the metric entropy results in the literature are for compact spaces (one exception is Theorem 7.34 of Steinwart and Christmann (2008), which relaxes the compactness requirement by adding some assumptions on the tail of $\nu_{\mathcal{X}}$ on \mathcal{X}).

So we could remove the compactness requirement from Assumption A1 and implicitly let Assumption A4 satisfy it, but we preferred to be explicit about it at the cost of a bit of redundancy in our set of assumptions.

Assumption A5 (Function Space Boundedness) The subset $\mathcal{F}^{|\mathcal{A}|} \subset B(\mathcal{X} \times \mathcal{A}; Q_{\max})$ is a separable and complete Carathéodory set with $R_{\max} \leq Q_{\max} < \infty$.

Assumption A5 requires all the functions in $\mathcal{F}^{|\mathcal{A}|}$ to be bounded so that the solutions of optimization problems (15)-(16) stay bounded. If they are not, they should be truncated, and thus, the truncation argument should be used in the analysis, see e.g., the proof of Theorem 21.1 of Györfi et al. (2002). The truncation argument does not change the final result, but complicates the proof at several places, so we stick to the above assumption to avoid unnecessary clutter. Moreover, in order to avoid the measurability issues resulting from taking supremum over an uncountable function space $\mathcal{F}^{|\mathcal{A}|}$, we require the space to be a separable and complete Carathéodory set (cf. Section 7.3 of Steinwart and Christmann 2008).

Assumption A6 (Function Approximation Property) The action-value function of any policy π belongs to $\mathcal{F}^{|\mathcal{A}|}$, i.e., $Q^\pi \in \mathcal{F}^{|\mathcal{A}|}$.

This “no function approximation error” assumption is standard in analyzing regularization-based nonparametric methods. This assumption is realistic and is satisfied for rich function spaces such as RKHS defined by universal kernels, e.g., Gaussian or exponential kernels (Section 4.6 of Steinwart and Christmann 2008). On the other hand, if the space is not large enough, we might have function approximation error. The behavior of the function approximation error for certain classes of “small” RKHS has been discussed by Smale and Zhou (2003); Steinwart and Christmann (2008). We stick to this assumption to simplify many key steps in the proofs.

Assumption A7 (Expansion of Smoothness) For all $Q \in \mathcal{F}^{|\mathcal{A}|}$, there exist constants $0 \leq L_R, L_P < \infty$, depending only on the MDP and $\mathcal{F}^{|\mathcal{A}|}$, such that for policy π ,

$$J(T^\pi Q) \leq L_R + \gamma L_P J(Q).$$

We require that the complexity of $T^\pi Q$ to be comparable to the complexity of Q itself. In other words, we require that if Q is smooth according to the regularizer J of a function space $\mathcal{F}^{|\mathcal{A}|}$, it stays smooth after the application of the Bellman operator. We believe that this is a reasonable assumption for many classes of MDPs with “sufficient” stochasticity and when $\mathcal{F}^{|\mathcal{A}|}$ is rich enough. The intuition is that if the Bellman operator has a “smoothing” effect, the norm of $T^\pi Q$ does not blow up and the function can still be represented well within $\mathcal{F}^{|\mathcal{A}|}$. Proposition 25 in Appendix F presents the conditions that for the so-called *convolutional* MDPs, Assumption A7 is satisfied. Briefly speaking, the conditions are 1) the transition probability kernel should have a finite gain (in the control-theoretic sense) in its frequency response, and 2) the reward function should be smooth according to the regularizer J . Of course, this is only an example of the class of problems for which this assumption holds.

5.1 Policy Evaluation Error

In this section, we focus on the k^{th} iteration of REG-LSPI. To simplify the notation, we use $\mathcal{D}_n = \{(Z_t, R_t, X_t)\}_{t=1}^n$ to refer to $\mathcal{D}_n^{(k)}$. The policy π_k depends on data used in the earlier iterations, but since we use independent set of samples $\mathcal{D}_n^{(k)}$ for the k^{th} iteration and π_k is independent of $\mathcal{D}_n^{(k)}$, we can safely ignore the randomness of π_k by working on the probability space obtained by conditioning on $\mathcal{D}_n^{(0)}, \dots, \mathcal{D}_n^{(k-1)}$, i.e., the probability space used in the k^{th} iteration is $(\Omega, \sigma_{\Omega}, \mathbb{P}_k)$ with $\mathbb{P}_k = \mathbb{P} \left\{ \cdot \mid \mathcal{D}_n^{(0)}, \dots, \mathcal{D}_n^{(k-1)} \right\}$. In order to avoid clutter, we do not use the conditional probability symbol. In the rest of this section, π refers to a $\sigma(\mathcal{D}_n^{(0)}, \dots, \mathcal{D}_n^{(k-1)})$ -measurable policy and is independent of \mathcal{D}_n ; \hat{Q} and $\hat{h}_n(\cdot) = \hat{h}_n(\cdot; Q)$ refer to the solution to (15)-(16) when $\pi, \lambda_{h,n}$, and $\lambda_{Q,n}$ replace $\pi_k, \lambda_{h,n}$, and $\lambda_{Q,n}$ in that set of equations, respectively.

The following theorem is the main result of this section and provides an upper bound on the statistical behavior of the policy evaluation procedure REG-LSTD.

Theorem 11 (Policy Evaluation) For any fixed policy π , let \hat{Q} be the solution to the optimization problem (15)-(16) with the choice of

$$\lambda_{h,n} = \lambda_{Q,n} = \left[\frac{1}{n J^2(Q^\pi)} \right]^{\frac{1}{1+\alpha}}.$$

If Assumptions A1-A7 hold, there exists $c(\delta) > 0$ such that for any $n \in \mathbb{N}$ and $0 < \delta < 1$, we have

$$\|\hat{Q} - T^\pi \hat{Q}\|_{\nu}^2 \leq c(\delta) n^{-\frac{1}{1+\alpha}},$$

with probability at least $1 - \delta$. Here $c(\delta)$ is equal to

$$c(\delta) = c_1 (1 + (\gamma L_P)^2) J^{\frac{2\alpha}{1+\alpha}}(Q^\pi) \ln(1/\delta) + c_2 \left(L_R^{\frac{2\alpha}{1+\alpha}} + \frac{L_R^2}{[J(Q^\pi)]^{\frac{2\alpha}{1+\alpha}}} \right),$$

for some constants $c_1, c_2 > 0$.

Theorem 11, which is proven in Appendix B, indicates how the number of samples and the difficulty of the problem as characterized by $J(Q^\pi)$, L_P , and L_R influence the policy evaluation error.¹³

This upper bound provides some insights about the behavior of the REG-LSTD algorithm. To begin with, it shows that under the specified conditions, REG-LSTD is a consistent algorithm: As the number of samples increases, the Bellman error decreases and asymptotically converges to zero. This is due to the use of a nonparametric function space and the proper control of its complexity through regularization. A parametric function space, e.g., a linear function approximator with a fixed number of features, does not generally have a similar guarantee unless the value function happens to belong to the span of the features. Achieving consistency for parametric function spaces requires careful choice

13. Without loss of generality and for simplicity we assumed that $J(Q^\pi) > 0$.

of features, and might be difficult. On the other hand, a rich enough nonparametric function space, for example one defined by a universal kernel (cf. Assumption A6), ensures the consistency of the policy evaluation algorithm.

This theorem, however, is much more powerful than a consistency result as it provides a finite-sample upper bound guarantee for the error, too. If the parameters of the REG-LSTD algorithm are selected properly, one may achieve the sample complexity upper bound of $O(n^{-1/(1+\alpha)})$. For the case of the Sobolev space $\mathbb{W}^k(\mathcal{X})$ with \mathcal{X} being an open Euclidean ball in \mathbb{R}^d and $k > d/2$, one may choose $\alpha = d/2k$ to obtain the error upper bound of $O(n^{-d/(2k+d)})$.¹⁴

To study the upper bound a bit closer, let us focus on the special case of $\gamma = 0$. For this choice of the discount factor, $T^\pi Q$ is equal to r^π and $(T^\pi Q)(X_i, A_i)$ is equal to R_i . One can see that the policy evaluation problem becomes a regression problem with the regression function r^π . The guarantee of this theorem would be then on $\|\hat{Q} - T^\pi Q\|_2^2 = \|\hat{Q} - r^\pi\|_2^2$, which is the usual squared error in the regression literature. Hence we reduced a regression problem to a policy evaluation problem. Because of this reduction, any lower bound on the regression would also be a lower bound on the policy evaluation problem.

It is well-known that the convergence rate of $n^{-d/(2k+d)}$ is asymptotically minimax optimal for the regression estimation for target functions belonging to the Sobolev space $\mathbb{W}^k(\mathcal{X})$ as well as some other smoothness classes with the k order of smoothness; cf. e.g., [Nussbaum \(1999\)](#) for the results for the Sobolev spaces, [Stone \(1982\)](#) for a closely related Hölder space $C^{p,\alpha}$, which with the choice of $k = p + \alpha$ ($k \in \mathbb{N}$ and $0 < \alpha \leq 1$) has the same rate, and [Tsybakov \(2009\)](#) for several results on minimax optimality of nonparametric estimators. More generally, the rate of $O(n^{-1/(1+\alpha)})$ is optimal too. For a regression function belonging to a function space \mathcal{F} with a packing entropy in the same form as in the upper bound of Assumption A4, the rate $\Omega(n^{-1/(1+\alpha)})$ is its minimax lower bound ([Yang and Barron, 1999](#)), making the upper bound optimal. Comparing these lower bounds with the upper bound $O(n^{-1/(1+\alpha)})$ (or $O(n^{-d/(2k+d)})$ for the Sobolev space) of this theorem indicates that REG-LSTD algorithm has the optimal error rate as a function of the number of samples n , which is a remarkable result.

Furthermore, to understand the fine behavior of the upper bound, beyond the dependence of the rate on n and α , we focus on the multiplicative term $c(\delta)$. Again we consider the special case of regression estimation as it is the only case we have some known lower bounds. With the choice of $\gamma = 0$, we have $\hat{Q}^\pi = r^\pi$, so $J(\hat{Q}^\pi) = J(r^\pi)$. Moreover, since $T^\pi Q = r^\pi + 0P^\pi Q = r^\pi$, we can choose $L_R = J(r^\pi)$ in Assumption A7. As a result $c(\delta) = c_1 J^{\frac{2\alpha}{1+\alpha}}(r^\pi) \ln(1/\delta)$ for a constant $c_1 > 0$. We are interested in studying the dependence of the upper bound on $J(r^\pi)$. We study its behavior when the function space is the Sobolev space $\mathbb{W}^k([0, 1])$ and $J(\cdot)$ is the corresponding Sobolev space norm. We choose $\alpha = 1/2k$ to get $J^{\frac{2\alpha}{1+\alpha}}(r^\pi)$ dependence of $c(\delta)$. On the other hand, for the regression estimation problem within the subset $\mathcal{F}_T^{\mathcal{A}} = \{Q(\cdot, a) \in \mathbb{W}^k([0, 1]) : J(Q) \leq J(r^\pi), \forall a \in \mathcal{A}\}$ of this Sobolev space, the fine behavior of the asymptotic minimax rate is determined by

the so-called Pinsker constant, whose dependence on J is in fact $J^{\frac{2\alpha}{1+\alpha}}(r^\pi)$, cf. e.g., [Nussbaum \(1999, 1985\)](#), [Golubov and Nussbaum \(1990\)](#), or Section 3.1 of [Tsybakov \(2009\)](#).¹⁵ Therefore, not only the exponent of the rate is optimal for this function space, but also its multiplicative dependence on the smoothness $J(r^\pi)$ is optimal.

For function spaces other than this choice of Sobolev space (i.e., the general case of α), we are not aware of any refined lower bound that indicates the optimality of $J^{\frac{2\alpha}{1+\alpha}}(r^\pi)$. We note that some available upper bounds for regression with comparable assumptions on the metric entropy have the same dependence on $J(r^\pi)$, e.g., [Steinwart et al. \(2009\)](#)¹⁶ or [Farahmand and Szepesvári \(2012\)](#), whose result is for the regression setting with exponential β -mixing input, but can also be shown for i.i.d. data. We conjecture that under our assumptions this dependence is optimal.

One may note that the proper selection of the regularization coefficients to achieve the optimal rate requires the knowledge of an unknown quantity $J(Q^\pi)$. This, however, is not a major concern as a proper model selection procedure finds parameters that result in a performance which is almost the same as the optimal performance. We comment on this issue in more detail in Section 6.

The proof of this theorem requires several auxiliary results, which are presented in the appendices, but the main idea behind the proof is as follows. Since $\|\hat{Q} - T^\pi Q\|_2^2 \leq 2\|\hat{Q} - \hat{h}_n(\cdot; \hat{Q})\|_2^2 + 2\|\hat{h}_n(\cdot; \hat{Q}) - T^\pi \hat{Q}\|_2^2$, we may upper bound the Bellman error by upper bounding each term in the right-hand side (RHS). One can see that for a fixed \hat{Q} , the optimization problem (15) essentially solves a regularized least-squares regression problem, which leads to small value of $\|\hat{h}_n(\cdot; \hat{Q}) - T^\pi \hat{Q}\|_2$, when there are enough samples and under proper conditions. The relation of the optimization problem (16) with $\|\hat{Q} - \hat{h}_n(\cdot; \hat{Q})\|_2$ is evident too. The difficulty, however, is that these two optimization problems are coupled: $\hat{h}_n(\cdot; \hat{Q})$ is a function of \hat{Q} which itself is a function of $\hat{h}_n(\cdot; \hat{Q})$. Thus, \hat{Q} appearing in (15) is not fixed, but is a random function \hat{Q} . The same is true for the other optimization problem as well. The coupling of the optimization problems makes the analysis more complicated than the usual supervised learning type of analysis. The dependencies between all the results that lead to the proof of Theorem 14 is depicted in Figure 2 in Appendix B.

In order to obtain fast convergence rates, we use concepts and techniques from the empirical process theory such as the peeling device, the chaining technique, and the modulus of continuity of the empirical process, cf. e.g., [van de Geer \(2000\)](#). By focusing on the behavior of the empirical process over local subsets of the function space, these techniques allow us to study the deviations of the process in a more refined way compared to a global approach that studies the supremum of the empirical process in the whole function space. These techniques are crucial to obtain a fast rate for large function spaces. We discuss them in more detail as we proceed in the proofs.

15. The Pinsker constant determines the effect of the noise variance too. We do not present such information in our bounds. Also note that most aforementioned results, except [Golubov and Nussbaum \(1990\)](#), consider a normal noise model, which is different from our bounded noise.

16. This is obtained by using Corollary 3 of [Steinwart et al. \(2009\)](#) after substituting $A_2(\lambda)$ by its upper bound $\lambda \|\cdot\|_{\mathcal{H}}^2$, which is valid whenever $f^* \in \mathcal{H}$, as is in our case. This result can be used after one converts the metric entropy condition to the condition on the decay rate of eigenvalues of a certain integral operator.

14. For examples of the metric entropy results for the Sobolev spaces, refer to Section A.5.6 alongside Lemma 6.21 of [Steinwart and Christmann \(2008\)](#), or Theorem 2.4 of [van de Geer \(2000\)](#) for $\mathcal{X} = [0, 1]$ or Lemma 20.6 of [Györfi et al. \(2002\)](#) for $\mathcal{X} = [0, 1]^d$. Also in this paper we use the notation $\mathbb{W}^k(\mathcal{X})$ to refer to $\mathbb{W}^{k,2}(\mathcal{X})$, the Sobolev space defined based on the L_2 -norm of the weak derivatives.

We mentioned earlier that one can actually reuse a single data set in all iterations. To keep the presentation more clear, we keep the current setup. The reason behind this can be explained better after the proof of Theorem 11. But note that from the convergence-rate point of view, the difference between reusing data or not is insignificant. If we have a batch of data with size n and we divide it into K chunks and only use one chunk per iteration of API, the rate would be $O\left(\left(\frac{n}{K}\right)^{-\frac{1}{1+\alpha}}\right)$. For finite K , or slowly growing K , this is essentially the same as $O\left(n^{-\frac{1}{1+\alpha}}\right)$.

5.2 Error Propagation in API

Consider an API algorithm that generates the sequence $\hat{Q}^{(0)} \rightarrow \pi_1 \rightarrow \hat{Q}^{(1)} \rightarrow \pi_2 \rightarrow \dots \rightarrow \hat{Q}^{(K-1)} \rightarrow \pi_K$, where π_k is the greedy policy w.r.t. $\hat{Q}^{(k-1)}$ and $\hat{Q}^{(k)}$ is the approximate action-value function for policy π_k . For the sequence $(\hat{Q}^{(k)})_{k=0}^{K-1}$, denote the Bellman Residual (BR) of the k^{th} action-value function by

$$\varepsilon_k^{\text{BR}} = \hat{Q}^{(k)} - T^{\pi_k} \hat{Q}^{(k)}. \quad (31)$$

The goal of this section is to study the effect of the ν -weighted L_2 -norm of the Bellman residual sequence $(\varepsilon_k^{\text{BR}})_{k=0}^{K-1}$ on the performance loss $\|Q^* - Q^{\pi_K}\|_{1,\rho}$ of the resulting policy π_K . Because of the dynamical nature of the MDP, the performance loss $\|Q^* - Q^{\pi_K}\|_{1,\rho}$ depends on the difference between the sampling distribution ν and the future state-action distribution in the form of $\rho P^{\pi_1} P^{\pi_2} \dots$. The precise form of this dependence is formalized in Theorem 13, which is a slight modification of a result by Farahmand et al. (2010).¹⁷

Before stating the results, we define the following *concentrability* coefficients that are used in a change of measure argument, see e.g., Munos (2007); Antos et al. (2008b); Farahmand et al. (2010).

Definition 12 (Expected Concentrability of Future State-Action Distributions)
Given the distributions $\rho, \nu \in \mathcal{M}(\mathcal{X} \times \mathcal{A})$, an integer $m \geq 0$, and an arbitrary sequence of stationary policies $(\pi_m)_{m \geq 1}$, let $\rho P^{\pi_1} P^{\pi_2} \dots P^{\pi_m} \in \mathcal{M}(\mathcal{X} \times \mathcal{A})$ denote the future state-action distribution obtained when the first state-action is distributed according to ρ and then we follow the sequence of policies $(\pi_k)_{k=1}^m$. Define the following *concentrability coefficients*:

$$c_{PI,\rho,\nu}(m_1, m_2; \pi) \triangleq \left(\mathbb{E} \left[\frac{d(\rho(P^{\pi^*})^{m_1}(P^\pi)^{m_2})}{d\nu} (X, A) \right] \right)^{\frac{1}{2}},$$

with $(X, A) \sim \nu$. If the future state-action distribution $\rho(P^{\pi^*})^{m_1}(P^\pi)^{m_2}$ is not absolutely continuous w.r.t. ν , then we take $c_{PI,\rho,\nu}(m_1, m_2; \pi) = \infty$.

In order to compactly present our results, we define the following notation:

$$a_k = \frac{(1-\gamma)\gamma^{K-k-1}}{1-\gamma^{K+1}}, \quad (0 \leq k < K) \quad (32)$$

17. The difference of these two results is in the way the norm of functions from the space $\mathcal{F}^{\mathcal{A}}$ is defined, which in turn corresponds to whether the distributions ν and ρ are defined over the state space \mathcal{X} , as Farahmand et al. (2010) defined, or over the state-action space $\mathcal{X} \times \mathcal{A}$, as we define here. These differences do not change the general form of the proof. See Theorem 3.2 in Chapter 3 of Farahmand (2011b) for the proof of the current result.

Theorem 13 (Error Propagation for API—Theorem 3 of Farahmand et al. 2010)
Let $p \geq 1$ be a real number, K be a positive integer, and $Q_{\max} \leq \frac{R_{\max}}{1-\gamma}$. Then for any sequence $(\hat{Q}^{(k)})_{k=0}^{K-1} \subset B(\mathcal{X} \times \mathcal{A}, Q_{\max})$ and the corresponding sequence $(\varepsilon_k^{\text{BR}})_{k=0}^{K-1}$ defined in (31), we have

$$\|Q^* - Q^{\pi_K}\|_{p,\rho} \leq \frac{2\gamma}{(1-\gamma)^2} \left[\inf_{r \in [0,1]} C_{PI,\rho,\nu}^{2r}(K; r) \mathcal{E}^{\frac{1}{2r}}(\varepsilon_0^{\text{BR}}, \dots, \varepsilon_{K-1}^{\text{BR}}; r) + \gamma^{\frac{K-1}{p}} R_{\max} \right],$$

where $\mathcal{E}(\varepsilon_0^{\text{BR}}, \dots, \varepsilon_{K-1}^{\text{BR}}; r) = \sum_{k=0}^{K-1} a_k^{2r} \|\varepsilon_k^{\text{BR}}\|_{2p,\nu}^{2r}$ and

$$C_{PI,\rho,\nu}(K; r) = \left(\frac{1-\gamma}{2} \right)^2 \sup_{\pi_0^*, \dots, \pi_{K-1}^*} \left[\sum_{m \geq 0} \gamma^m \left(c_{PI,\rho,\nu}(K-k-1, m+1; \pi_{k+1}^*) + c_{PI,\rho,\nu}(K-k, m; \pi_k^*) \right) \right]^2.$$

For better understanding of the intuition behind the error propagation results in general, refer to Munos (2007); Antos et al. (2008b); Farahmand et al. (2010). The significance of this particular theorem and the ways it improves previous similar error propagation results such as that of Antos et al. (2008b) (for API) and Munos (2007) (for AVI) is thoroughly discussed by Farahmand et al. (2010). We briefly comment on it in Section 5.3.

5.3 Performance Loss of REG-LSPI

In this section, we use the error propagation result (Theorem 13 in Section 5.2) together with the upper bound on the policy evaluation error (Theorem 11 in Section 5.1) to derive an upper bound on the performance loss $\|Q^* - Q^{\pi_K}\|_{1,\rho}$ of REG-LSPI. This is the main theoretical result of this work. Before stating the theorem, let us denote $\hat{\Pi}(\mathcal{F}^{\mathcal{A}})$ as the set of all policies that are greedy w.r.t. a member of $\mathcal{F}^{\mathcal{A}}$, i.e., $\hat{\Pi}(\mathcal{F}^{\mathcal{A}}) = \{\hat{\pi}(\cdot; Q) : Q \in \mathcal{F}^{\mathcal{A}}\}$.

Theorem 14 Let $(\hat{Q}^{(k)})_{k=0}^{K-1}$ be the solutions of the optimization problem (15)-(16) with the choice of

$$\lambda_{h,n}^{(k)} = \lambda_{Q,n}^{(k)} = \left[\frac{1}{nJ^2(Q^{\pi_k})} \right]^{\frac{1}{1+\alpha}}.$$

Let Assumptions A1-A5 hold; Assumptions A6 and A7 hold for any $\pi \in \hat{\Pi}(\mathcal{F}^{\mathcal{A}})$, and $\inf_{r \in [0,1]} C_{PI,\rho,\nu}(K; r) < \infty$. Then there exists $C_{LSPI}(\delta, K, \rho, \nu)$ such that for any $n \in \mathbb{N}$ and $0 < \delta < 1$, we have

$$\|Q^* - Q^{\pi_K}\|_{1,\rho} \leq \frac{2\gamma}{(1-\gamma)^2} \left[C_{LSPI}(\delta, K; \rho, \nu) n^{-\frac{1}{2(1+\alpha)}} + \gamma^{K-1} R_{\max} \right],$$

with probability at least $1 - \delta$.

In this theorem, the function $C_{LSPI}(\delta, K; \rho, \nu) = C_{LSPI}(\delta, K; \rho, \nu; L_R, L_P, \alpha, \beta, \gamma)$ is

$$C_{LSPI}(\delta, K; \rho, \nu; L_R, L_P, \alpha, \beta, \gamma) = C_1^{\frac{1}{2}}(\delta) \inf_{r \in [0,1]} \left\{ \left(\frac{1-\gamma}{1-\gamma^{K+1}} \right)^r \sqrt{\frac{1-(\gamma^{2r})^K}{1-\gamma^{2r}}} C_{PI,\rho,\nu}^{\frac{1}{2}}(K; r) \right\},$$

with $C_1(\delta)$ being defined as

$$C_1(\delta) = \sup_{\pi \in \Pi(\mathcal{F}^{1,4})} \left[c_1 (1 + (\gamma L_P)^2) J^{\frac{2\alpha}{1+\alpha}}(Q^\pi) \ln \left(\frac{K}{\delta} \right) + c_2 \left(L_R^{\frac{2\alpha}{1+\alpha}} + \frac{L_R^2}{[J(Q^\pi)]^{1+\alpha}} \right) \right],$$

in which $c_1, c_2 > 0$ are universal constants.

Proof Fix $0 < \delta < 1$. For each iteration $k = 0, \dots, K-1$, invoke Theorem 11 with the confidence parameter δ/k and take the supremum over all policies to upper bound the Bellman residual error $\|\varepsilon_k^{\text{BR}}\|_\nu$ as

$$\left\| \hat{Q}^{(k)} - T^{\pi_k} \hat{Q}^{(k)} \right\|_\nu^2 \leq \underbrace{\sup_{\pi \in \Pi(\mathcal{F}^{1,4})} c \left(J(Q^\pi), L_R, L_P, \alpha, \beta, \gamma, \frac{\delta}{K} \right)}_{\triangleq c'} n^{-\frac{1}{1+\alpha}},$$

which holds with probability at least $1 - \frac{\delta}{K}$. Here $c(\cdot)$ is defined as in Theorem 11. For any $r \in [0, 1]$, we have

$$\begin{aligned} \mathcal{E}(\varepsilon_0^{\text{BR}}, \dots, \varepsilon_{K-1}^{\text{BR}}; r) &= \sum_{k=0}^{K-1} \frac{2^r}{a_k} \|\varepsilon_k^{\text{BR}}\|_\nu^2 \leq c' n^{-\frac{1}{1+\alpha}} \sum_{k=0}^{K-1} a_k^{\frac{1}{2}} \\ &= c' n^{-\frac{1}{1+\alpha}} \left(\frac{1-\gamma}{1-\gamma^{K+1}} \right)^{2r} \frac{1-(\gamma^{2r})^K}{1-\gamma^{2r}}, \end{aligned}$$

where we used the definition of a_k (32). We then apply Theorem 13 with the choice of $p = 1$ to get that with probability at least $1 - \delta$, we have

$$\|Q^* - Q^{\pi_K}\|_{1,\rho} \leq \frac{2\gamma}{(1-\gamma)^2} \left[C_{\text{LSPI}}(\rho, \nu; K) n^{-\frac{1}{1+\alpha}} + \gamma^{K-1} R_{\max} \right].$$

Here

$$C_{\text{LSPI}}(\rho, \nu; K) = \sqrt{\sup_{\pi \in \Pi(\mathcal{F}^{1,4})} c \left(J(Q^\pi), L_R, L_P, \alpha, \gamma, \frac{\delta}{K} \right) \inf_{r \in [0,1]} \left\{ \left(\frac{1-\gamma}{1-\gamma^{K+1}} \right)^r \sqrt{\frac{1-(\gamma^{2r})^K}{1-\gamma^{2r}}} C_{\text{PI},\rho,\nu}^{\frac{1}{2}}(K; r) \right\}}.$$

■

Theorem 14 upper bounds the performance loss and relates it to the number of samples n , the capacity of the function space quantified by α , the number of iterations K , the concentrability coefficients, and some other properties of the MDP such as L_R, L_P , and γ .

This theorem indicates that the behavior of the upper bound as a function of the number of samples is $O(n^{-\frac{1}{2(1+\alpha)}})$. This upper bound is notable because of its minimax optimality, as discussed in detail after Theorem 11.

The term C_{LSPI} has two main components. The first is $C_{\text{PI},\rho,\nu}(\cdot; r)$, which describes the effect of the sampling distribution ν and the evaluation distribution ρ , as well as the

transition probability kernel of the MDP itself on the performance loss. This term has been thoroughly discussed by Farahmand et al. (2010), but briefly speaking it indicates that ν and ρ affect the performance through a weighted summation of $c_{\text{PI},\rho,\nu}$ (Definition 12). The concentrability coefficients $c_{\text{PI},\rho,\nu}$ is defined as the square root of the expected squared Radon-Nikodym of the future state-action distributions starting from ρ w.r.t. the sampling distribution ν . This may be much tighter compared to the previous results (e.g., Antos et al. 2008b) that depend on the *supremum* of the Radon-Nikodym derivative. One may also notice that Theorem 13 actually provides a stronger result than what is reported in Theorem 14: The effect of errors at earlier iterations on the performance loss is geometrically decayed. So one may potentially use a fewer number of samples in the earlier iterations of REG-LSPI (or any other API algorithm) to get the same guarantee on the performance loss. We ignore this effect to simplify the result.

The other important term is C_1 , which mainly describes the effect of L_R, L_P , and $\sup_{\pi \in \Pi(\mathcal{F}^{1,4})} J(Q^\pi)$ on the performance loss. These quantities depend on the MDP, as well as the function space $\mathcal{F}^{1,4}$. If the function space is “mismatched” with the MDP, these quantities would be small, otherwise they may even be infinity.

Note that C_1 provides an upper bound on the constant in front of REG-LSTD procedure by taking supremum over all policies in $\Pi(\mathcal{F}^{1,4})$. This might be a conservative estimate as the actual encountered policies are the rather restricted random sequence $\pi_0, \pi_1, \dots, \pi_{K-1}$ generated by the REG-LSPI procedure. One might expect that as the sequence $\hat{Q}^{(k-1)}$ converge to a neighbourhood of Q^* , the value function Q^{π_k} of the greedy policy $\pi_k = \hat{\pi}(\cdot; \hat{Q}^{(k-1)})$, which is the policy being evaluated, converges to a neighbourhood of Q^* too. Thus with certain assumptions, one might be able to show that its smoothness $J(Q^{\pi_k})$, the quantity that appears in the upper bound of Theorem 11, belongs to a neighbourhood of $J(Q^*)$. If $J(Q^*)$ is small, the value of $J(Q^{\pi_k})$ in that neighbourhood can be smaller than $\sup_{\pi \in \Pi(\mathcal{F}^{1,4})} J(Q^\pi)$. We postpone the analysis of this finer structure of the problem to future work.

Finally we note that the optimality of the error bound for the policy evaluation task, as shown by Theorem 11, does not necessarily imply that the REG-LSPI algorithm has the optimal sample complexity rate for the corresponding RL/Planning problem as well. The reason is that it is sufficient to get close to the optimal policy, which is the ultimate goal in RL/Planning, even though the estimate of the action-value function is still inaccurate. To act optimally, it is sufficient to have an action-value function whose greedy policy is the same as the optimal policy. This can happen even if there is some error in the estimated action-value function. This is called the *action-gap phenomenon* and has been analyzed in the reinforcement learning context by Farahmand (2011a).

5.3.1 COMPARISON WITH SIMILAR STATISTICAL GUARANTEES

Theorem 14 might be compared with the results of Antos et al. (2008b), who introduced a BRM-based API procedure and studied its statistical properties, Lazarevic et al. (2012), who analyzed LSPI with linear function approximators, Ávila Pires and Szepesvári (2012), who studied a regularized variant of LSTD, and Ghavamzadeh et al. (2011), who analyzed the statistical properties of Lasso-TD. Although these results address different algorithms, comparing them with the results of this work is insightful.

We first focus on [Antos et al. \(2008b\)](#). Their simplified upper bound for $\|Q^* - Q^{\pi_K}\|_{1,\rho}$ is $C_{\rho,\nu}^{1/2} \sqrt{V_{\mathcal{F}} \log(n) + \ln(K/\delta)} n^{-1/4}$, in which $V_{\mathcal{F}}$ is the “effective” dimension of \mathcal{F} and is defined based on the pseudo-dimension of sub-graphs of \mathcal{F} and the so-called “VC-crossing dimension” of \mathcal{F} ; and $C_{\rho,\nu}$ is a concentrability coefficient and plays a similar role to our $C_{\text{PI},\rho,\nu}(K; r)$. In contrast, our simplified upper bound is $C_{\text{LSP1}}(\delta) n^{-\frac{1}{2(1+\tau)}}$, in which $C_{\text{LSP1}}(\delta)$ can roughly be factored into $C_{\text{PI},\rho,\nu}^{\frac{1}{2}}(K; r) C_1(J(Q^r), L_R, L_P) \sqrt{\ln(K/\delta)}$.

One important difference between these two results is that [Antos et al. \(2008b\)](#) considered parametric function spaces, which have finite effective dimension $V_{\mathcal{F}}$, while this work considers nonparametric function spaces, which essentially are infinite dimensional. The way they use the parametric function space assumption is equivalent to assuming that $\log M_1(u, \mathcal{F}, x_{1:n}) \leq V_{\mathcal{F}} \log(\frac{1}{u})$ as opposed to $\log \mathcal{N}_{\infty}(u, \mathcal{F}_B, x_{1:n}) \leq C \left(\frac{B}{u}\right)^{2\alpha}$ of [Assumption A4](#). Our assumption lets us describe the capacity of infinite dimensional function spaces \mathcal{F} . Disregarding this crucial difference, one may also note that our upper bound’s dependence on the number of samples (i.e., $O(n^{-\frac{1}{2(1+\tau)}}$) is much faster than theirs (i.e., $O(n^{-1/4})$). This is more noticeable when we apply our result to a finite dimensional function space, which can be done by letting $\alpha \rightarrow 0$ at a certain rate, to recover the error upper bound of $n^{-1/2}$.¹⁸ This improvement is mainly because of more advanced techniques used in our analysis, i.e., the relative deviation tail inequality and the peeling device in this work in contrast with the uniform deviation inequality of [Antos et al. \(2008b\)](#).

The other difference is in the definition of concentrability coefficients ($C_{\text{PI},\rho,\nu}(K)$ vs. $C_{\rho,\nu}$). In [Definition 12](#), we use the expectation of Radon-Nikodym derivative of two distributions while their definition uses the supremum of a similar quantity. This can be a significant improvement in the multiplicative constant of the upper bound. For more information regarding this improvement, which can be used to improve the result of [Antos et al. \(2008b\)](#) too, refer to [Farahmand et al. \(2010\)](#).

[Lazaric et al. \(2012\)](#) analyzed unregularized LSTD/LSPI specialized for linear function approximators with finite number of basis functions (parametric setting). Their rate of $O(n^{-1/2})$ for $\|V^* - V^{\pi_K}\|_{2,\rho}$ is faster than the rate in the work of [Antos et al. \(2008b\)](#), and is comparable to our rate for $\|Q^* - Q^{\pi_K}\|_{1,\rho}$ when $\alpha \rightarrow 0$. The difference of their work with ours is that they focus on a parametric class of function approximators as opposed to the nonparametric class in this work. Moreover, because they formulate the LSTD as a fixed-point problem, in contrast to this work and that of [Antos et al. \(2008b\)](#), their algorithm and results are only applicable to on-policy sampling scenario.

[Ávila Pires and Szepesvári \(2012\)](#) studied a regularized version of LSTD in the parametric setting that works for both on-policy and off-policy sampling. Beside the difference between the class of function spaces with this work (parametric vs. nonparametric), another algorithmic difference is that they only use a regularizer for the projected Bellman error term, similar to [\(16\)](#), as opposed to using regularizers in both terms of REG-LSTD [\(15\)-\(16\)](#) (cf. [Section 4.1](#)). Also the weight used in their loss function, the matrix M in their paper, is not necessarily the one induced by data. Their result indicates $O(n^{-1/2})$ for the projected Bellman error, which is comparable, though with some subtle differences, to [Lazaric et al.](#)

18. For problems with finite state space, we have $\log \mathcal{N}_{\infty}(u, \mathcal{F}_B) \leq |\mathcal{X}| \log(\frac{Q_{\max}}{u})$, so with a similar $\alpha \rightarrow 0$ argument, we get $O(n^{-1/2})$ error upper bound (disregarding the logarithmic terms).

[\(2012\)](#). It is remarkable that they separate the error bound analysis to deterministic and probabilistic parts. In the deterministic part, they use perturbation analysis to relate the loss to the error in the estimation of certain parameters used by the algorithms. In the probabilistic part, they provide upper bounds on the error in estimation of the parameters. We conjecture that their proof technique, even though simple and elegant, cannot easily be extended to provide the right convergence rate for large function spaces because the current analysis is based on a uniform bound on the error of a noisy matrix. Providing a tight uniform bound for a matrix (or operator) for large state spaces might be difficult or impossible to achieve.

[Ghavamzadeh et al. \(2011\)](#) analyzed Lasso-TD, a policy evaluation algorithm that uses linear function approximators and enforces sparsity by the l_1 -regularization, and provided error upper bounds w.r.t. the empirical measure (or what they call Markov design). Their error upper bound is $O(\|w^*\|_1^2 \log(p))^{1/4} n^{-1/4}$, where w^* is the weight vector describing the projection of Q^{π^*} onto the span of p basis functions. With some extra assumptions on the Gramian of the basis functions, they obtain faster rate of $O(\sqrt{\|w^*\|_0 \log(p)} n^{-1/2})$. These results indicate that by using the sparsity-inducing regularizer, the dependence of the error bound on the number of features becomes logarithmic.

We conjecture that if one uses REG-LSTD with a linear function space (similar to [Section 4.2.1](#)) with $J^2(h) = \|h\|_1$ and $J^2(Q) = \|w\|_1$, the current analysis leads to the error upper bound $O(\|w^*\|_1^{1/2} n^{-1/4})$ with a logarithmic dependence on p . This result might be obtained using [Corollary 5 of Zhang \(2002\)](#) as [Assumption A4](#). To get a faster rate of $O(n^{-1/2})$, one should make extra assumptions on the Gramian—as was done by [Ghavamzadeh et al. \(2011\)](#). We should emphasize that even with the choice of linear function approximators and the l_1 -regularization, REG-LSTD would not be the same algorithm as Lasso-TD since REG-LSTD uses regularization in both optimization problems [\(15\)-\(16\)](#). Also note that the error upper bound of [Ghavamzadeh et al. \(2011\)](#) is on the empirical norm $\|\cdot\|_{2,\mathcal{D}_n}$ as opposed to the norm $\|\cdot\|_{2,\nu}$, which is w.r.t. the measure ν . This means that their result does not provide a generalization upper bound on the quality of the estimated value function over the whole state space, but provides an upper bound only on the training data.

Comparing this work with its conference version ([Farahmand et al., 2009b](#)), we observe that the main difference in the theoretical guarantees is that the current results are for more general function spaces than the Sobolev spaces considered in the conference paper. [Assumption A4](#) specifies the requirement on the capacity of the function space, which is satisfied not only by the Sobolev spaces (with the choice of $\alpha = d/2k$ for $\mathbb{W}^k(\mathcal{X})$ with \mathcal{X} being an open Euclidean ball in \mathbb{R}^d and $k > d/2$; cf. [Section A.5.6](#) alongside [Lemma 6.21 of Steinwart and Christmann \(2008\)](#), or [Theorem 2.4 of van de Geer \(2000\)](#) for $\mathcal{X} = [0, 1]$ or [Lemma 20.6 of Györfi et al. \(2002\)](#) for $\mathcal{X} = [0, 1]^d$), but also many other large function spaces including several commonly-used RKHS.

6. Conclusion and Future Work

We introduced two regularization-based API algorithms, namely REG-LSPI and REG-BRM, to solve RL/Planning problems with large state spaces. Our formulation was general and could incorporate many types of function spaces and regularizers. We specifically showed how these algorithms can be implemented efficiently when the function space is the

span of a finite number of basis functions (parametric model) or an RKHS (nonparametric model).

We then focused on the statistical properties of REG-LSPI and provided its performance loss upper bound (Theorem 14). The error bound demonstrated the role of the sample size, the complexity of function space to which the action-value function belongs (quantified by its metric entropy in Assumption A4), and the intrinsic properties of the MDP such as the behavior of concentrability coefficients and the smoothness-expansion property of the Bellman operator (Definition 12 and Assumption A7). The result indicated that the dependence on the sample size for the task of policy evaluation is optimal.

This work (and its conference (Farahmand et al., 2009b) and the dissertation (Farahmand, 2011b)) versions alongside the work on the Regularized Fitted Q-Iteration algorithm (Farahmand et al., 2008, 2009a) are the first that address the statistical performance of a *regularized* RL algorithm. Nevertheless, there have been a few other work that also used regularization for RL/Planning problems, most often without analyzing their statistical properties.

Jung and Polani (2006) studied adding regularization to BRM, but their solution is restricted to deterministic problems. The main contribution of that work was the development of fast incremental algorithms using the *sparsification* technique. The l_1 -regularization has been considered by Loh et al. (2007), who were similarly concerned with incremental implementations and computational efficiency. Xu et al. (2007) provided a kernel-based, but not regularized, formulation of LSPI. They used sparsification to provide basis functions for the LSTD procedure. Sparsification leads to a selection of only a subset of data points to be used as the basis functions, thus indirectly controls the complexity of the resulting function space. This should be contrasted with a regularization-based approach in which the regularizer interacts with the empirical loss to jointly determine the subset of the function space to which the estimate belongs.

Koller and Ng (2009) formulated an l_1 -regularization fixed-point formulation LSTD, which is called Lasso-TD by Ghavamzadeh et al. (2011), and provided LARS-like algorithm (Efron et al., 2004) to compute the solutions. Johns et al. (2010) considered the same fixed-point formulation and cast it as a linear complementarity problem. The statistical properties of this l_1 -regularized fixed-point formulation is studied by Ghavamzadeh et al. (2011), as discussed earlier. Lasso-TD has a fixed-point formulation, which looks different from our coupled optimization formulation (15)-(16), but under on-policy sampling scenario, it is equivalent to a particular version of REG-LSTD: If we choose a fixed linear function approximator (parametric), use the l_1 -norm in the projection optimization problem (15), but do not regularize optimization problem (16) (i.e., $\lambda_{Q,n} = 0$), we get Lasso-TD. Geist and Scherrer (2012) suggested a different algorithm where the projection is not regularized (i.e., $\lambda_{h,n} = 0$), but the optimization problem (16) is regularized with the l_1 -norm of the parameter weights. The choice of only regularizing (16) is the same as the one in the algorithm introduced and analyzed by Avila Pires and Szepesvári (2012), except that the latter work uses the l_2 -norm. Hofmann et al. (2012) introduced an algorithm similar to that of Geist and Scherrer (2012) with the difference that the projection optimization (15) uses the l_2 -norm (so it is a mixed l_1/l_2 -regularized algorithm). All these algorithms are parametric. Several TD-based algorithms and their regularized variants are discussed in a survey by Dann et al. (2014).

Taylor and Parr (2009) unified several kernelized reinforcement learning algorithms, and showed the equivalence of kernelized value function approximators such as GP-TD (Engel et al., 2005), the work of Xu et al. (2007), and a few other methods with a model-based reinforcement learning algorithm that has certain regularization on the transition kernel estimator, reward estimator, or both. Their result was obtained by considering two separate regularized regression problems: One that predicts the reward function given the current state and the other that predicts the next-state kernel values given the current-state ones. Their formulation is different from our formulation that is stated as a coupled optimization problem in an RKHS.

Similar to other kernel-based algorithms (e.g., SVMs, Gaussian Process Regressions, Splines, etc.), devising a computationally efficient implementation of REG-LSPI/BRM is important to ensure that it is a practical algorithm for large-scale problems. A naive implementation of these algorithms requires the computation time of $O(n^3K)$, which is prohibitive for large sample sizes. One possible workaround is to reduce the effective number of samples by the sparsification technique (Engel et al., 2005; Jung and Polani, 2006; Xu et al., 2007). The other is to use elegant vector-matrix multiplication methods, which are used in iterative methods for matrix inversion, such as those based on the Fast Multipole Methods (Beason and Greengard, 1997) and the Fast Gauss Transform (Yang et al., 2004). These methods can reduce the computational cost of vector-matrix multiplication from $O(n^2)$ to $O(n \log n)$, which results in computation time of $O(n^2K \log n)$ for REG-LSPI/BRM, at the cost of some small, but controlled, numerical error. Another possibility is to use stochastic gradient-like algorithms similar to the works of Lin et al. (2012); Qin et al. (2014). The use of stochastic gradient-like algorithms is especially appealing in the light of results such as Borton and Bousquet (2008); Shalev-Shwartz and Strehl (2008). They analyze the tradeoff between the statistical error and the optimization error caused by the choice of optimization method. They show that one might achieve lower generalization error by using a faster stochastic gradient-like algorithm, which processes more data points less accurately, rather than a slower but more accurate optimization algorithm, which can only process fewer data points. Designing scalable optimization algorithms for REG-LSPI/BRM is a topic for future work.

An important issue in the successful application of any RL/Planning algorithm, including REG-LSPI and REG-BRM, is the proper choice of parameters. In REG-BRM and REG-LSTD we are faced with the choice of $\mathcal{F}^{|\mathcal{A}|}$ and the corresponding regularization parameters $\lambda_{Q,n}$ and $\lambda_{h,n}$. The proper choice of these parameters, however, depends on quantities that are not known, e.g., $J(Q^\pi)$ and the choice of $\mathcal{F}^{|\mathcal{A}|}$ that “matches” with the MDP. This problem in the RL/Planning context has been addressed by Farahmand and Szepesvári (2011). They introduced a complexity-regularization-based model selection algorithm that allows one to design adaptive algorithms: Algorithms that perform almost the same as the one with the prior knowledge of the best parameters.

Another important question is how to extend these algorithms to deal with continuous action MDPs. There are two challenges: Computational and statistical. The computational challenge is finding the greedy action at each state in the policy improvement step. In general, this is an intractable optimization problem, which cannot be solved exactly or even with any suboptimality guarantee. To analyze this inexact policy improvement some parts of the theory, especially the error propagation result, should be modified. Moreover, we also have a statistical challenge: One should specifically control the complexity of the

policy space as the complexity of $\{\max_{a \in \mathcal{A}} Q(\cdot, a) : Q \in \mathcal{F}^{\mathcal{A}}\}$ might be infinity even though $\mathcal{F}^{\mathcal{A}}$ has a finite complexity (Antos et al., 2008a). A properly modified algorithm might be similar to the continuous-action extension of Farahmand et al. (2015), an API algorithm that explicitly controls the complexity of the policy space.

Finally an open theoretical question is to characterize the properties of the MDP that determine the function space to which action-value function belong. A similar question is how the values of L_P and L_R in Assumption A7 are related to the intrinsic properties of the MDP. We partially addressed this question for the convolutional MDPs, but analysis of more general MDPs is remained to be done.

Acknowledgments

We thank the members of the Reinforcement Learning and Artificial Intelligence (RLAI) research group at the University of Alberta and the Reasoning and Learning Lab at McGill University for fruitful discussions. We thank the anonymous reviewers and the editor for their helpful comments and suggestions, which improved the quality of the paper. We gratefully acknowledge funding from the National Science and Engineering Research Council of Canada (NSERC) and Alberta Innovates Centre for Machine Learning (AICML). Shie Mannor was partially supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement 306638 (SUPREL).

Proofs and Auxiliary Results

In these appendices, we first prove Theorem 10, which provides the closed-form solutions for REG-LSTD and REG-BRM when the function space is an RKHS (Appendix A). We then attend to the proof of Theorem 11 (Policy Evaluation error for REG-LSTD). The main body of the proof for Theorem 11 is in Appendix B. To increase the readability and flow, the proofs of some of the auxiliary and more technical results are postponed to Appendices C, D, and E.

More specifically, we prove an extension of Theorem 21.1 of Györfi et al. (2002) in Appendix C (Lemma 15). We present a modified version of Theorem 10.2 of van de Geer (2000) in Appendix D. We then provide a covering number result in Appendix E (Lemma 20). The reason we require these results will become clear in Appendix B. Finally, we introduce convolutional MDPs as an instance of problems that satisfy Assumption A7 (Appendix F).

We would like to remark that the generic ‘‘constants’’ $c, c' > 0$ in the proofs, especially those related to the statistical guarantees, might change from line to line, if their exact value is not important in the bound. These values are constant as a function of important quantities of the upper bound (such as $n, \alpha, J(Q^F)$, etc.), but may depend on Q_{\max} or $|\mathcal{A}|$.

Appendix A. Proof of Theorem 10 (Closed-Form Solutions for RKHS Formulation of REG-LSTD/BRM)

Proof REG-BRM: First, notice that the optimization problem (28) can be written in the form $c_n(Q) + \lambda_{Q,n} \|Q\|_{\mathcal{H}}^2 \stackrel{Q}{\rightarrow} \min!$ with an appropriately defined functional c_n .¹⁹ In order to apply the representer theorem (Schölkopf et al., 2001), we require to show that c_n depends on Q only through the data-points $Z_1, Z'_1, \dots, Z_n, Z'_n$. This is immediate for all the terms that define c_n except the term that involves $\hat{h}_n(\cdot; Q)$. However, since \hat{h}_n is defined as the solution to the optimization problem (27), calling for the representer theorem once again, we observe that \hat{h}_n can be written in the form

$$\hat{h}_n(\cdot; Q) = \sum_{t=1}^n \beta_t^* \kappa(Z_t, \cdot),$$

where $\beta^* = (\beta_1^*, \dots, \beta_n^*)^\top$ satisfies

$$\beta^* = \operatorname{argmin}_{\beta \in \mathbb{R}^n} \left[\left\| \mathbf{K}_h \beta - \hat{T}^\pi Q \right\|_n^2 + \lambda_{h,n} \beta^\top \mathbf{K}_h \beta \right].$$

Solving this minimization problem leads to

$$\beta^* = (\mathbf{K}_h + n\lambda_{h,n} \mathbf{I})^{-1} (\hat{T}^\pi Q).$$

In both equations ($\hat{T}^\pi Q$) is viewed as the n -dimensional vector

$$\left((\hat{T}^\pi Q)(Z_1), \dots, (\hat{T}^\pi Q)(Z_n) \right)^\top = (R_1 + \gamma Q(Z_1), \dots, R_n + \gamma Q(Z_n))^\top.$$

Thus, β^* depends on Q only through $Q(Z_1), \dots, Q(Z_n)$. Plugging this solution into (28), we get that $c_n(Q)$ indeed depends on Q through

$$Q(Z_1), Q(Z'_1), \dots, Q(Z_n), Q(Z'_n),$$

and thus on data points $Z_1, Z'_1, \dots, Z_n, Z'_n$. The representer theorem then implies that the minimizer of $c_n(Q) + \lambda_{Q,n} \|Q\|_{\mathcal{H}}^2$ can be written in the form $Q(\cdot) = \sum_{i=1}^{2n} \hat{\alpha}_i \kappa(\tilde{Z}_i, \cdot)$, where $\tilde{Z}_i = Z_i$ if $i \leq n$ and $\tilde{Z}_i = Z'_{i-n}$, otherwise.

Let $\tilde{\alpha} = (\alpha_1, \dots, \alpha_n, \alpha'_1, \dots, \alpha'_n)^\top$. Using the reproducing kernel property of κ , we get the optimization problem

$$\|C_1 \mathbf{K}_Q \tilde{\alpha} - (r + \gamma C_2 \mathbf{K}_Q \tilde{\alpha})\|_n^2 - \|\mathbf{B}(r + \gamma C_2 \mathbf{K}_Q \tilde{\alpha})\|_n^2 + \lambda_{Q,n} \tilde{\alpha}^\top \mathbf{K}_Q \tilde{\alpha} \stackrel{\tilde{\alpha}}{\rightarrow} \min!.$$

Solving this for $\tilde{\alpha}$ concludes the proof for REG-BRM.

REG-LSTD: The first part of the proof that shows c_n depends on Q only through the data-points $Z_1, Z'_1, \dots, Z_n, Z'_n$ is exactly the same as the proof of REG-BRM. Thus, using the representer theorem, the minimizer of (30) can be written in the form $Q(\cdot) = \sum_{i=1}^{2n} \hat{\alpha}_i \kappa(\tilde{Z}_i, \cdot)$, where $\tilde{Z}_i = Z_i$ if $i \leq n$ and $\tilde{Z}_i = Z'_{i-n}$, otherwise. Let $\tilde{\alpha} = (\alpha_1, \dots, \alpha_n, \alpha'_1, \dots, \alpha'_n)^\top$. Using the reproducing kernel property of κ , we get the optimization problem

$$\|(C_1 - \gamma \mathbf{E} C_2) \mathbf{K}_Q \tilde{\alpha} - \mathbf{E} r\|_n^2 + \lambda_{Q,n} \tilde{\alpha}^\top \mathbf{K}_Q \tilde{\alpha} \stackrel{\tilde{\alpha}}{\rightarrow} \min!.$$

Replacing $C_1 - \gamma \mathbf{E} C_2$ with \mathbf{F} and solving for $\tilde{\alpha}$ concludes the proof. \blacksquare

¹⁹ Here $f(Q) \stackrel{Q}{\rightarrow} \min!$ indicates that Q is a minimizer of $f(Q)$.

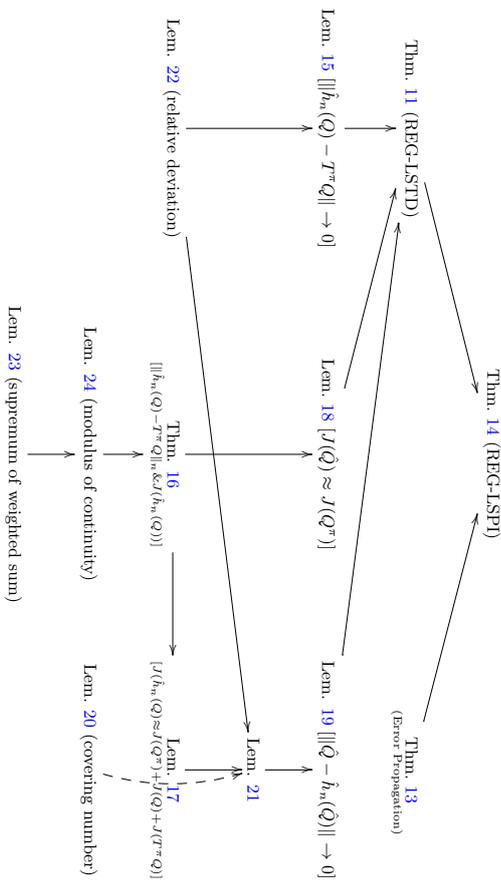


Figure 2: Dependencies of results used to prove the statistical guarantee for REG-LSP (Theorem 14).

Appendix B. Proof of Theorem 11 (Statistical Guarantee for REG-LSTD)

The goal of Theorem 11 is to provide a finite-sample upper bound on the Bellman error $\|\hat{Q} - T^\pi \hat{Q}\|_\pi$ for REG-LSTD defined by the optimization problems (15) and (16). Since $\|\hat{Q} - T^\pi \hat{Q}\|_\pi^2 \leq 2\|\hat{Q} - \hat{h}_n(\cdot; \hat{Q})\|_\pi^2 + 2\|\hat{h}_n(\cdot; \hat{Q}) - T^\pi \hat{Q}\|_\pi^2$, we may upper bound the Bellman error by upper bounding each term in the RHS. Recall from the discussion after Theorem 11 that the analysis is more complicated than the conventional supervised learning setting because the corresponding optimization problems are coupled: $\hat{h}_n(\cdot; \hat{Q})$ is a function of \hat{Q} which itself is a function of $\hat{h}_n(\cdot; \hat{Q})$.

Theorem 11 is proven using Lemma 15, which upper bounds $\|\hat{h}_n(\cdot; \hat{Q}) - T^\pi \hat{Q}\|_\pi$, and Lemma 19, which upper bounds $\|\hat{Q} - \hat{h}_n(\cdot; \hat{Q})\|_\pi$. We also require to relate the smoothness $J(\hat{Q})$ to the smoothness $J(Q^\pi)$. Lemma 18 specifies this relation. The proof of these lemmas themselves require further developments, which will be discussed when we encounter them. Figure 2 shows the dependencies between all results that lead to the proof of Theorem 11 and consequently Theorem 14.

The following lemma controls the error behavior resulting from the optimization problem (15). This lemma, which is a result on the error upper bound of a regularized regression estimator, is similar to Theorem 21.1 of Györfi et al. (2002) with two main differences. First,

it holds uniformly over $T^\pi Q$ (as opposed to a fixed function $T^\pi Q$); second, it holds for function spaces that satisfy a general metric entropy condition (as opposed to the special case of the Sobolev spaces).

Lemma 15 (Convergence of $\hat{h}_n(\cdot; Q)$ to $T^\pi Q$) For any random $Q \in \mathcal{F}^{|A|}$, let $\hat{h}_n(Q)$ be defined according to (15). Under Assumptions A1–A5 and A7, there exist finite constants $c_1, c_2 > 0$ such that for any $n \in \mathbb{N}$ and $0 < \delta < 1$, we have

$$\|\hat{h}_n(\cdot; Q) - T^\pi Q\|_\pi^2 \leq 4\lambda_{h,n} J^2(T^\pi Q) + 2\lambda_{h,n} J^2(Q) + c_1 \frac{1}{n\lambda_{h,n}^\alpha} + c_2 \frac{\ln(1/\delta)}{n},$$

with probability at least $1 - \delta$.

Proof See Appendix C. \blacksquare

When we use this lemma to prove Theorem 11, the action-value function Q that appears in the bound is the result of the optimization problems defined in (16), that is \hat{Q} , and so is random. Lemma 18, which we will prove later, provides a deterministic upper bound for the smoothness $J(\hat{Q})$ of this random quantity.

It turns out that to derive our main result, we require to know more about the behavior of the regularized regression estimator than what is shown in Lemma 15. In particular, we need an upper bound on the empirical error of the regularized regression estimator $\hat{h}_n(\cdot; Q)$ (cf. (33) below). Moreover, we should bound the random smoothness $J(\hat{h}_n(\cdot; Q))$ by some deterministic quantities, which turns out to be a function of $J(T^\pi Q)$ and $J(Q)$. Theorem 16 provides us with the required upper bounds. This theorem is a modification of Theorem 10.2 by van de Geer (2000), with two main differences: 1) It holds uniformly over Q and 2) $h_n(\cdot; Q)$ uses the same data \mathcal{D}_n that is used to estimate Q itself.

We introduce the following notation: Let $w = (x, a, r, x')$ and define the random variables $w_i = (X_i, A_i, R_i, X'_i)$ for $1 \leq i \leq n$. The data set \mathcal{D}_n would be $\{w_1, \dots, w_n\}$. For a measurable function $g : \mathcal{X} \times \mathcal{A} \times \mathbb{R} \times \mathcal{X} \rightarrow \mathbb{R}$, let $\|g\|_m^2 = \frac{1}{n} \sum_{i=1}^n |g(w_i)|^2$. Consider the regularized least squares estimator:

$$\hat{h}_n(\cdot; Q) = \operatorname{argmin}_{h \in \mathcal{F}^{|A|}} \left[\|h - [R_\gamma + \gamma Q(X'_i, \pi(X'_i))]\|_n^2 + \lambda_{h,n} J^2(h) \right], \quad (33)$$

which is the same as (15) with π replacing π_π .

Theorem 16 (Empirical error and smoothness of $\hat{h}_n(\cdot; Q)$) For a random function $Q \in \mathcal{F}^{|A|}$, let $\hat{h}_n(\cdot; Q)$ be defined according to (33). Suppose that Assumptions A1–A5 and A7 hold. Then there exist constants $c_1, c_2 > 0$, such that for any $n \in \mathbb{N}$ and $0 < \delta < 1$, we have

$$\|\hat{h}_n(\cdot; Q) - T^\pi Q\|_\pi \leq c_1 \max \left\{ \frac{Q_{\max}^{1+\alpha} \sqrt{\ln(1/\delta)}}{\lambda_{h,n}^{\frac{\alpha}{2}}}, \frac{Q_{\max} (J(Q) + J(T^\pi Q))^{\frac{1+\alpha}{2}}}{\lambda_{h,n}^{\frac{\alpha}{2}}} \left(\frac{\ln(1/\delta)}{n} \right)^{\frac{1}{2(1+\alpha)}} \right\},$$

$$J(\hat{h}_n(\cdot; Q)) \leq c_2 \max \left\{ J(Q) + J(T^\pi Q), \frac{Q_{\max}^{1+\alpha} \sqrt{\frac{\ln(1/\delta)}{n}}}{\lambda_{h,n}^{\frac{1+\alpha}{2}}} \right\},$$

with probability at least $1 - \delta$.

Proof See Appendix D. ■

The following lemma, which is an immediate corollary of Theorem 16, indicates that with the proper choice of the regularization coefficient, the complexity of the regression function $\hat{h}_n(\cdot; Q)$ is in the same order as the complexities of Q , $T^\pi Q$, and Q^π . This result will be used in the proof of Lemma 21, which itself is used in the proof of Lemma 19.

Lemma 17 (Smoothness of $\hat{h}_n(\cdot; Q)$) For a random $Q \in \mathcal{F}^{|\mathcal{A}|}$, let $\hat{h}_n(\cdot; Q)$ be the solution to the optimization problem (15) with the choice of regularization coefficient

$$\lambda_{h,n} = \left[\frac{1}{n J^2(Q^\pi)} \right]^{\frac{1}{1+\alpha}}.$$

Let Assumptions A1–A5 and A7 hold. Then, there exists a finite constant $c > 0$, depending on Q_{\max} , such that for any $n \in \mathbb{N}$ and $0 < \delta < 1$, the upper bound

$$J(\hat{h}_n(\cdot; Q)) \leq c \left(J(T^\pi Q) + J(Q) + J(Q^\pi) \sqrt{\ln(1/\delta)} \right)$$

holds with probability at least $1 - \delta$.

Proof With the choice of $\lambda_{h,n} = \left[\frac{1}{n J^2(Q^\pi)} \right]^{\frac{1}{1+\alpha}}$, Theorem 16 implies that there exist some finite constant $c_1 > 0$ as well as $c_2 > 0$, which depends on Q_{\max} , such that for any $n \in \mathbb{N}$ and $0 < \delta < 1$, the inequality

$$\begin{aligned} J(\hat{h}_n(\cdot; Q)) &\leq c_1 \max \left\{ J(Q) + J(T^\pi Q), \frac{Q_{\max}^{1+\alpha} \sqrt{\frac{\ln(1/\delta)}{n}}}{\left[\frac{1}{n J^2(Q^\pi)} \right]^{\frac{1}{2}}} \right\} \\ &\leq c_2 \left(J(T^\pi Q) + J(Q) + J(Q^\pi) \sqrt{\ln(1/\delta)} \right) \end{aligned}$$

holds with probability at least $1 - \delta$. ■

An intuitive understanding of this result might be gained if we consider $\hat{h}_n(\cdot; Q^\pi)$, which is the regression estimate for $T^\pi Q^\pi = Q^\pi$. This lemma then indicates that the smoothness of $\hat{h}_n(\cdot; Q^\pi)$ is comparable to the smoothness of its target function Q^π . This is intuitive whenever the regularization coefficients are chosen properly.

The following lemma relates $J(\hat{Q})$ and $J(T^\pi \hat{Q})$, which are random, to the complexity of the action-value function of the policy π , i.e., $J(Q^\pi)$. This result is used in the proof of Theorem 11.

Lemma 18 (Smoothness of \hat{Q}) Let Assumptions A1–A7 hold, and let \hat{Q} be the solution to (16) with the choice of

$$\lambda_{h,n} = \left[\frac{1}{n J^2(Q^\pi)} \right]^{\frac{1}{1+\alpha}}.$$

Then, there exists a finite constant $c > 0$ such that for any $n \in \mathbb{N}$ and $0 < \delta < e^{-1}$, we have

$$\lambda_{Q,n} J^2(\hat{Q}) \leq \lambda_{Q,n} J^2(Q^\pi) + c \frac{J^{\frac{2\alpha}{1+\alpha}}(Q^\pi) \ln(1/\delta)}{n^{\frac{1}{1+\alpha}}},$$

with probability at least $1 - \delta$.

Proof By Assumption A6 we have $Q^\pi \in \mathcal{F}^{|\mathcal{A}|}$, so by the optimizer property of \hat{Q} (cf. (16)), we get

$$\lambda_{Q,n} J^2(\hat{Q}) \leq \left\| \hat{Q} - \hat{h}_n(\cdot; \hat{Q}) \right\|_{\mathcal{D}_n}^2 + \lambda_{Q,n} J^2(\hat{Q}) \leq \left\| Q^\pi - \hat{h}_n(\cdot; Q^\pi) \right\|_{\mathcal{D}_n}^2 + \lambda_{Q,n} J^2(Q^\pi). \quad (34)$$

Since $Q^\pi = T^\pi Q^\pi$, we have $\|Q^\pi - \hat{h}_n(\cdot; Q^\pi)\|_{\mathcal{D}_n} = \|T^\pi Q^\pi - \hat{h}_n(\cdot; Q^\pi)\|_{\mathcal{D}_n}$. So Theorem 16 shows that with the choice of $\lambda_{h,n} = \left[\frac{1}{n J^2(Q^\pi)} \right]^{\frac{1}{1+\alpha}}$, there exists a finite constant $c > 0$ such that for any $n \in \mathbb{N}$ and for $0 < \delta < e^{-1} \approx 0.3679$, we have

$$\left\| Q^\pi - \hat{h}_n(\cdot; Q^\pi) \right\|_{\mathcal{D}_n}^2 \leq c_1 \left(1 \vee Q_{\max}^2 \right) \frac{J^{\frac{2\alpha}{1+\alpha}}(Q^\pi) \ln(1/\delta)}{n^{\frac{1}{1+\alpha}}}, \quad (35)$$

with probability at least $1 - \delta$. Chaining inequalities (34) and (35) finishes the proof. ■

The other main ingredient of the proof of Theorem 11 is an upper bound to $\|\hat{Q} - \hat{h}_n(\cdot; \hat{Q})\|_\nu$, which is closely related to the optimization problem (16). This task is done by Lemma 19. In the proof of this lemma, we call Lemma 21, which shall be stated and proven right after this result.

Lemma 19 (Convergence of $\|\hat{Q} - \hat{h}_n(\cdot; \hat{Q})\|_\nu$) Let \hat{Q} be the solution to the set of coupled optimization problems (15)–(16). Suppose that Assumptions A1–A7 hold. Then there exists a finite constant $c > 0$ such that for any $n \in \mathbb{N}$ and $0 < \delta < 2e^{-1}$ and with the choice of

$$\lambda_{h,n} = \lambda_{Q,n} = \left[\frac{1}{n J^2(Q^\pi)} \right]^{\frac{1}{1+\alpha}},$$

we have

$$\left\| \hat{Q} - \hat{h}_n(\cdot; \hat{Q}) \right\|_\nu^2 \leq c \frac{(1 + \gamma^2 L_P^2)^\alpha J^{\frac{2\alpha}{1+\alpha}}(Q^\pi) \ln(1/\delta) + L_R^{\frac{2\alpha}{1+\alpha}}}{n^{\frac{1}{1+\alpha}}},$$

with probability at least $1 - \delta$.

Proof Decompose

$$\left\| \hat{Q} - \hat{h}_n(\cdot; \hat{Q}) \right\|_{\nu}^2 = I_{1,n} + I_{2,n},$$

with

$$\begin{aligned} \frac{1}{2} I_{1,n} &= \left\| \hat{Q} - \hat{h}_n(\cdot; \hat{Q}) \right\|_{\mathcal{D}_n}^2 + \lambda_{Q,n} J^2(\hat{Q}), \\ I_{2,n} &= \left\| \hat{Q} - \hat{h}_n(\cdot; \hat{Q}) \right\|_{\nu}^2 - I_{1,n}. \end{aligned} \quad (36)$$

In what follows, we upper bound each of these terms.

$I_{1,n}$: Use the optimizer property of \hat{Q} to get

$$\frac{1}{2} I_{1,n} = \left\| \hat{Q} - \hat{h}_n(\cdot; \hat{Q}) \right\|_{\mathcal{D}_n}^2 + \lambda_{Q,n} J^2(\hat{Q}) \leq \left\| Q^\pi - \hat{h}_n(\cdot; Q^\pi) \right\|_{\mathcal{D}_n}^2 + \lambda_{Q,n} J^2(Q^\pi).$$

To upper bound $\left\| Q^\pi - \hat{h}_n(\cdot; Q^\pi) \right\|_{\mathcal{D}_n}^2 = \left\| T^\pi Q^\pi - \hat{h}_n(\cdot; Q^\pi) \right\|_{\mathcal{D}_n}^2$, we evoke [Theorem 16](#). For our choice of $\lambda_{Q,n}$, there exists a constant $c_1 > 0$ such that for any $n \in \mathbb{N}$ and $0 < \delta_1 < 1$, we have

$$\frac{1}{2} I_{1,n} \leq \lambda_{Q,n} J^2(Q^\pi) + c_1 \frac{J^{\frac{2\alpha}{1+\alpha}}(Q^\pi) \ln(1/\delta_1)}{n^{\frac{1+\alpha}{1+\alpha}}}, \quad (37)$$

with probability at least $1 - \delta_1$.

$I_{2,n}$: With our choice of $\lambda_{Q,n}$ and $\lambda_{h,n}$, [Lemma 21](#), which shall be proven later, indicates that there exist some finite constants $c_2, c_3, c_4 > 0$ such that for any $n \in \mathbb{N}$ and finite $J(Q^\pi)$, L_R , and L_P , and $0 < \delta_2 < 1$, we have

$$I_{2,n} \leq c_2 \frac{L_R^{\frac{2\alpha}{1+\alpha}} + [J(Q^\pi)]^{\frac{2\alpha}{1+\alpha}} [\ln(1/\delta_2)]^{\frac{1+\alpha}{1+\alpha}}}{n^{\frac{1+\alpha}{1+\alpha}}} + c_3 \frac{(1 + \gamma^2 L_P^2)^\alpha}{n \lambda_{Q,n}} + c_4 \frac{\ln(1/\delta_2)}{n}, \quad (38)$$

with probability at least $1 - \delta_2$. For $\delta_2 < e^{-1}$ and $\alpha \geq 0$, we have $[\ln(1/\delta_2)]^{\frac{1+\alpha}{1+\alpha}} \leq \ln(1/\delta_2)$, and also

$$\frac{1}{n \lambda_{Q,n}^\alpha} = \frac{[J(Q^\pi)]^{\frac{2\alpha}{1+\alpha}}}{n^{\frac{1+\alpha}{1+\alpha}}} \leq \frac{[J(Q^\pi)]^{\frac{2\alpha}{1+\alpha}}}{n^{\frac{1+\alpha}{1+\alpha}}} \ln(1/\delta_2). \quad (39)$$

With the right choice of constants, $\frac{\ln(1/\delta_2)}{n}$ can be absorbed into the other terms. Select $\delta_1 = \delta_2 = \delta/2$. Inequalities [\(37\)](#), [\(38\)](#), and [\(39\)](#) imply that with the specified choice of $\lambda_{Q,n}$ and $\lambda_{h,n}$, there exists a finite constant $c_5 > 0$ such that for any $0 < \delta < 2e^{-1}$, we have

$$\left\| \hat{Q} - \hat{h}_n(\cdot; \hat{Q}) \right\|_{\nu}^2 \leq c_5 \frac{(1 + \gamma^2 L_P^2)^\alpha J^{\frac{2\alpha}{1+\alpha}}(Q^\pi) \ln(1/\delta) + L_R^{\frac{1+2\alpha}{1+\alpha}}}{n^{\frac{1+\alpha}{1+\alpha}}},$$

with probability at least $1 - \delta$. \blacksquare

To upper bound $I_{2,n}$, defined in [\(36\)](#), we simultaneously apply the peeling device (cf. [Section 5.3](#) of [van de Geer 2000](#)) on two different, but coupled, function spaces (one to which

\hat{Q} belongs and the other to which $\hat{h}_n(\cdot; \hat{Q})$ belongs). In each layer of peeling, we apply an exponential tail inequality to control the relative deviation of the empirical mean from the true mean ([Lemma 22](#) in [Appendix C](#)). We also require a covering number result, which is stated as [Lemma 20](#). The final result of this procedure is a tight upper bound on $I_{2,n}$, as stated in [Lemma 21](#).

To prepare for the peeling argument, define the following subsets of \mathcal{F} and $\mathcal{F}^{\mathcal{A}}$:

$$\begin{aligned} \mathcal{F}_\sigma &\triangleq \{f : f \in \mathcal{F}, J^2(f) \leq \sigma\}, \\ \mathcal{F}_\sigma^{\mathcal{A}} &\triangleq \{f : f \in \mathcal{F}^{\mathcal{A}}, J^2(f) \leq \sigma\}. \end{aligned}$$

Let

$$g_{Q,h}(x, a) \triangleq \sum_{j=1}^{|\mathcal{A}|} \mathbb{1}_{\{a=q_j\}} [Q_j(x) - h_j(x)]^2. \quad (40)$$

To simplify the notation, we use $z = (x, a)$ and $Z = (X, A)$ in the rest of this section. Define G_{σ_1, σ_2} as the space of $g_{Q,h}$ functions with $J(Q) \leq \sigma_1$ and $J(h) \leq \sigma_2$, i.e.,

$$G_{\sigma_1, \sigma_2} \triangleq \left\{ g_{Q,h} : \mathbb{R}^d \times \mathcal{A} \rightarrow \mathbb{R}; Q \in \mathcal{F}_{\sigma_1}^{\mathcal{A}}, h \in \mathcal{F}_{\sigma_2}^{\mathcal{A}} \right\}. \quad (41)$$

The following lemma provides an upper bound on the covering numbers of G_{σ_1, σ_2} .

Lemma 20 (Covering Number) *Let Assumptions A3, A4, and A5 hold. Then there exists a constant $c_1 > 0$, independent of $\sigma_1, \sigma_2, \alpha, Q_{\max}$, and $|\mathcal{A}|$, such that for any $u > 0$ and all $((x_1, a_1), \dots, (x_n, a_n)) \in \mathcal{X} \times \mathcal{A}$, the empirical covering number of the class of functions G_{σ_1, σ_2} defined in [\(41\)](#) w.r.t. the empirical norm $\|\cdot\|_{2, z_{1:n}}$ is upper bounded by*

$$\log N_2(u, G_{\sigma_1, \sigma_2}, (x, a)_{1:n}) \leq c_1 |\mathcal{A}|^{1+\alpha} Q_{\max}^{2\alpha} (\sigma_1^\alpha + \sigma_2^\alpha) u^{-2\alpha}.$$

Proof See [Appendix E](#). \blacksquare

Next, we state and prove [Lemma 21](#), which provides a high probability upper bound on $I_{2,n}$.

Lemma 21 *Let $I_{2,n}$ be defined according to [\(36\)](#). Under Assumptions A1–A5 and A7 and with the choice of*

$$\lambda_{h,n} = \lambda_{Q,n} = \left[\frac{1}{n J^2(Q^\pi)} \right]^{\frac{1}{1+\alpha}},$$

there exist constants $c_1, c_2, c_3 > 0$, such that for any $n \in \mathbb{N}$, finite $J(Q^\pi)$, L_R , and L_P , and $\delta > 0$ we have

$$I_{2,n} \leq c_1 \frac{L_R^{\frac{2\alpha}{1+\alpha}} + [J(Q^\pi)]^{\frac{2\alpha}{1+\alpha}} [\ln(1/\delta)]^{\frac{1+\alpha}{1+\alpha}}}{n^{\frac{1+\alpha}{1+\alpha}}} + c_2 \frac{(1 + \gamma^2 L_P^2)^\alpha}{n \lambda_{Q,n}} + c_3 \frac{\ln(1/\delta)}{n},$$

with probability at least $1 - \delta$.

Proof Let $Z = (X, A)$ be a random variable with distribution ν that is independent from \mathcal{D}_n . Without loss of generality, we assume that $Q_{\max} \geq 1/2$. We use the peeling device in conjunction with Lemmas 20 and 22 to obtain a tight high-probability upper bound on $I_{2,n}$. Based on the definition of $I_{2,n}$ in (36) we have

$$\mathbb{P}\{I_{2,n} > t\} = \mathbb{P}\left\{\frac{\mathbb{E}\left[g_{\hat{Q}, \hat{h}_n(\cdot; \hat{Q})}(Z)|\mathcal{D}_n\right] - \frac{1}{n}\sum_{i=1}^n g_{\hat{Q}, \hat{h}_n(\cdot; \hat{Q})}(Z_i)}{t + 2\lambda_{Q,n} J^2(\hat{Q})} + \mathbb{E}\left[g_{\hat{Q}, \hat{h}_n(\cdot; \hat{Q})}(Z)|\mathcal{D}_n\right]}\right\} > \frac{1}{2}. \quad (42)$$

To benefit from the peeling device, we relate the complexity of $\hat{h}_n(\cdot; \hat{Q})$ to the complexity of \hat{Q} . For a fixed $\delta_1 > 0$ and some constant $c > 0$, to be specified shortly, define the following event:

$$\mathcal{A}_0 = \left\{\omega : J^2(\hat{h}_n(\cdot; \hat{Q})) \leq c \left(J^2(T^\pi \hat{Q}) + J^2(\hat{Q}) + J^2(Q^\pi) \ln(1/\delta_1)\right)\right\}.$$

Lemma 17 indicates that $\mathbb{P}\{\mathcal{A}_0\} \geq 1 - \delta_1$, where the constant c here can be chosen to be three times of the squared value of the constant in the lemma. We have $\mathbb{P}\{I_{2,n} > t\} = \mathbb{P}\{I_{2,n} > t, \mathcal{A}_0^c\} + \mathbb{P}\{I_{2,n} > t, \mathcal{A}_0\} \leq \delta_1 + \mathbb{P}\{I_{2,n} > t, \mathcal{A}_0\}$, so we focus on upper bounding $\mathbb{P}\{I_{2,n} > t, \mathcal{A}_0\}$.

Since $\hat{Q} \in \mathcal{F}^{|\mathcal{A}|}$, there exists $l \in \mathbb{N}_0$ such that $2^{l\mathbb{H}}(l \neq 0) \leq 2\lambda_{Q,n} J^2(\hat{Q}) < 2^{l+1}t$. Fix $l \in \mathbb{N}_0$. For any $Q \in \mathcal{F}^{|\mathcal{A}|}$, Assumption A7 relates $J(T^\pi Q)$ to $J(Q)$:

$$J^2(Q) \leq \frac{2^l t}{\lambda_{Q,n}} \Rightarrow J^2(T^\pi Q) \leq 2 \left(L_R^2 + \gamma^2 L_P^2 \frac{2^l t}{\lambda_{Q,n}}\right).$$

Thus on the event \mathcal{A}_0 , if $\hat{Q} \in \mathcal{F}_{\sigma_1^l}^{|\mathcal{A}|}$ where $\sigma_1^l = \frac{2^l t}{\lambda_{Q,n}}$, we also have $\hat{h}_n(\hat{Q}) \in \mathcal{F}_{\sigma_2^l}^{|\mathcal{A}|}$ with

$$\sigma_2^l = c \left[2 \left(L_R^2 + (1 + \gamma^2 L_P^2) \frac{2^l t}{\lambda_{Q,n}}\right) + J^2(Q^\pi) \ln(1/\delta_1)\right]. \quad (43)$$

Apply the peeling device on (42). Use (43) and note that if for an $l \in \mathbb{N}_0$ we have $2\lambda_{Q,n} J^2(\hat{Q}) \geq 2^{l\mathbb{H}}(l \neq 0)$, we also have $t + 2\lambda_{Q,n} J^2(\hat{Q}) \geq 2^l t$ to get

$$\begin{aligned} \mathbb{P}\{I_{2,n} > t\} &= \mathbb{P}\{I_{2,n} > t, \mathcal{A}_0^c\} + \mathbb{P}\{I_{2,n} > t, \mathcal{A}_0\} \\ &\leq \delta_1 + \sum_{l=0}^{\infty} \mathbb{P}\left\{\mathcal{A}_0, 2^{l\mathbb{H}}(l \neq 0) \leq 2\lambda_{Q,n} J^2(\hat{Q}) < 2^{l+1}t\right\} \\ &\leq \delta_1 + \sum_{l=0}^{\infty} \mathbb{P}\left\{\frac{\mathbb{E}\left[g_{\hat{Q}, \hat{h}_n(\cdot; \hat{Q})}(Z)|\mathcal{D}_n\right] - \frac{1}{n}\sum_{i=1}^n g_{\hat{Q}, \hat{h}_n(\cdot; \hat{Q})}(Z_i)}{t + 2\lambda_{Q,n} J^2(\hat{Q})} + \mathbb{E}\left[g_{\hat{Q}, \hat{h}_n(\cdot; \hat{Q})}(Z)|\mathcal{D}_n\right]}\right\} > \frac{1}{2} \\ &\leq \delta_1 + \sum_{l=0}^{\infty} \mathbb{P}\left\{\sup_{g_{Q,n} \in \mathcal{G}_{\sigma_1^l, \sigma_2^l}} \frac{\mathbb{E}[g_{Q,h}(Z)|\mathcal{D}_n] - \frac{1}{n}\sum_{i=1}^n g_{Q,h}(Z_i)}{2^l t + \mathbb{E}[g_{Q,h}(Z)|\mathcal{D}_n]}\right\} > \frac{1}{2}. \end{aligned} \quad (44)$$

Let us study the behavior of the l^{th} term of the above summation by verifying the conditions of Lemma 22 with the choice of $\varepsilon = \frac{1}{2}$ and $\eta = 2^l t$.

Condition (A1): Since all functions involved are bounded by Q_{\max} , it is easy to see that $|g_{Q,h}(x, u)| \leq \sum_{j=1}^{|\mathcal{A}|} \mathbb{I}_{\{a=a_j\}} |Q_j(x) - h_j(x)|^2 \leq 4Q_{\max}^2$. Therefore, K_1 , defined in Lemma 22, can be set to $K_1 = 4Q_{\max}^2$.

Condition (A2): We have $\mathbb{E}\left[|Q(Z) - h(Z)|^2\right] \leq 4Q_{\max}^2 \mathbb{E}\left[|Q(Z) - h(Z)|^2\right]$. Therefore, K_2 can be set to $K_2 = 4Q_{\max}^2$.

Condition (A3): We should satisfy $\frac{\sqrt{2}}{4} \sqrt{\eta \eta} \geq 288 \max\{8Q_{\max}^2, \sqrt{8}Q_{\max}\}$. Since $\eta = 2^l t \geq t$, it is sufficient to have

$$t \geq \frac{c}{n}, \quad (C1)$$

in which c is a function of Q_{\max} (we can choose $c = 2 \times 4608^2 Q_{\max}^4$).

Condition (A4): We shall verify that for $\varepsilon' \geq \frac{1}{8}\eta = \frac{1}{8}2^l t$, and $\sigma_1 = \sigma_1^l$ and $\sigma_2 = \sigma_2^l$, the following holds:

$$\begin{aligned} \frac{\sqrt{n}(\frac{1}{2})\varepsilon'}{96\sqrt{2} \max\{K_1, 2K_2\}} &\geq \\ \int_{\frac{1}{16 \max\{K_1, 2K_2\}} \varepsilon'}^{\sqrt{\varepsilon'}} \left(\log \mathcal{N}_2\left(u, \left\{g \in \mathcal{G}_{\sigma_1, \sigma_2} : \frac{1}{n} \sum_{i=1}^n g^2(z_i) \leq 16\varepsilon'\right\}, z_{1:n}\right)\right)^{1/2} &du. \end{aligned} \quad (45)$$

Notice that there exists a constant $c > 0$ such that for any $u, \varepsilon' > 0$

$$\begin{aligned} \log \mathcal{N}_2\left(u, \left\{g \in \mathcal{G}_{\sigma_1, \sigma_2} : \frac{1}{n} \sum_{i=1}^n g^2(z_i) \leq 16\varepsilon'\right\}, z_{1:n}\right) &\leq \log \mathcal{N}_2(u, \mathcal{G}_{\sigma_1, \sigma_2}, z_{1:n}) \\ &\leq c(\sigma_1^l + \sigma_2^l)u^{-2\alpha}, \end{aligned} \quad (46)$$

where we used Lemma 20 in the second inequality.

Plug (46) into (45) with the choice of $\sigma_1 = \sigma_1^l = \frac{2^l t}{\lambda_{Q,n}}$ and $\sigma_2 = \sigma_2^l = c[2(L_R^2 + (1 + \gamma^2 L_P^2) \frac{2^l t}{\lambda_{Q,n}}) + J^2(Q^\pi) \ln(1/\delta_1)]$. Therefore, for some constant $c' = c'(Q_{\max}) > 0$, the inequality

$$c' \sqrt{n} \varepsilon' \geq \int_0^{\sqrt{\varepsilon'}} \underbrace{\left(\frac{2^l t}{\lambda_{Q,n}}\right)^\alpha + c \left[2 \left(L_R^2 + (1 + \gamma^2 L_P^2) \frac{2^l t}{\lambda_{Q,n}}\right) + J^2(Q^\pi) \ln(1/\delta_1)\right]^\alpha}_{(b)} u^{-\alpha} du,$$

implies (45). Because $(a + b)^{\frac{1}{2}} \leq (a^{\frac{1}{2}} + b^{\frac{1}{2}})$ for non-negative a and b , it suffices to verify the following two conditions:

(a) We shall verify that for $\varepsilon' \geq \frac{1}{8}2^l t$, we have

$$c \sqrt{n} \varepsilon' \geq \left(\frac{2^l t}{\lambda_{Q,n}}\right)^\alpha \varepsilon'^{\frac{1-2\alpha}{2}} \Leftrightarrow c \frac{\sqrt{n} \varepsilon'^{\frac{1+2\alpha}{2}} \lambda_{Q,n}^{\frac{\alpha}{2}}}{(2^l t)^{\frac{\alpha}{2}}} \geq 1$$

for some $c > 0$. Substituting ε' with $2^t t$, we see that it is enough if for some constant $c > 0$,

$$t \geq \frac{c}{2^t n \lambda_{Q,n}^{\alpha}}. \quad (D1)$$

(b) We should verify that for $\varepsilon' \geq \frac{5}{8} 2^t t$, the following is satisfied:

$$\sqrt{n\varepsilon'} \geq c \left[\underbrace{L_R^2 + (1 + \gamma^2 L_P^2)}_{(b_1)} \underbrace{\lambda_{Q,n}^{-2t}}_{(b_2)} + \underbrace{J^2(Q^\pi) \ln(1/\delta)}_{(b_3)} \right]^{\alpha/2} \varepsilon'^{\frac{1-\alpha}{2}},$$

for some $c > 0$. After some manipulations, we get that the previous inequality holds if the following three inequalities are satisfied:

$$(b_1) : \quad t \geq c_1' \frac{L_R^{\frac{2\alpha}{1+\alpha}}}{n^{\frac{1+\alpha}{1+\alpha}}}, \quad (D2)$$

$$(b_2) : \quad t \geq c_2' \frac{(1 + \gamma^2 L_P^2)^\alpha}{n \lambda_{Q,n}^\alpha}, \quad (D3)$$

$$(b_3) : \quad t \geq c_3' \frac{[J(Q^\pi)]^{\frac{2\alpha}{1+\alpha}} \ln(1/\delta)]^{\frac{\alpha}{1+\alpha}}}{n^{\frac{1+\alpha}{1+\alpha}}}, \quad (D4)$$

for some constants $c_1', c_2', c_3' > 0$.

Fix $\delta > 0$ and let $\delta_1 \equiv \delta/2$. Whenever (C1), (D1), (D2), (D3), and (D4) are satisfied, for some choice of constants $c, c' > 0$ we have

$$\begin{aligned} \mathbb{P}\{I_{2n} > t\} &\leq \frac{\delta}{2} + \sum_{l=0}^{\infty} 60 \exp\left(-\frac{n(2^l)(\frac{1}{4})(1-\frac{1}{2})}{128 \times 2304 \times \max\{16Q_{\max}^4, 4Q_2^2\}}\right) \\ &\leq \frac{\delta}{2} + c \exp(-c'n t). \end{aligned}$$

Let the left-hand side be equal δ and solve for t . Considering all aforementioned conditions, we get that there exist constants $c_1, c_2, c_3 > 0$ such that for any $n \in \mathbb{N}$, finite $J(Q^\pi)$, L_R , and L_P , and $\delta > 0$, we have

$$I_{2n} \leq c_1 \frac{L_R^{\frac{2\alpha}{1+\alpha}} + [J(Q^\pi)]^{\frac{2\alpha}{1+\alpha}} \ln(1/\delta)]^{\frac{2\alpha}{1+\alpha}}}{n^{\frac{1+\alpha}{1+\alpha}}} + c_2 \frac{(1 + \gamma^2 L_P^2)^\alpha}{n \lambda_{Q,n}^\alpha} + c_3 \frac{\ln(1/\delta)}{n},$$

with probability at least $1 - \delta$. ■

After developing these tools, we are ready to prove Theorem 11.

Proof [Proof of Theorem 11] We want to show that $\|\hat{Q} - T^\pi \hat{Q}\|_\nu$ is small. Since (15)-(16) minimize $\|h_n(\cdot; \hat{Q}) - T^\pi \hat{Q}\|_\nu$ and $\|\hat{Q} - h_n(\cdot; \hat{Q})\|_\nu$, we upper bound $\|\hat{Q} - T^\pi \hat{Q}\|_\nu$ in terms of these quantities as follows:

$$\|\hat{Q} - T^\pi \hat{Q}\|_\nu^2 \leq 2 \|\hat{Q} - \hat{h}_n(\cdot; \hat{Q})\|_\nu^2 + 2 \|\hat{h}_n(\cdot; \hat{Q}) - T^\pi \hat{Q}\|_\nu^2. \quad (47)$$

Let us upper bound each of these two terms in the RHS. Fix $0 < \delta < 1$. **Bounding** $\|\hat{h}_n(\cdot; \hat{Q}) - T^\pi \hat{Q}\|_\nu$: Lemma 15 indicates that there exist constants $c_1, c_2 > 0$ such that for any random $\hat{Q} \in \mathcal{F}^{|\mathcal{A}|}$ and any fixed $n \in \mathbb{N}$, we have

$$\|\hat{h}_n(\cdot; \hat{Q}) - T^\pi \hat{Q}\|_\nu^2 \leq \lambda_{h,n} \left(2J^2(\hat{Q}) + 4J^2(T^\pi \hat{Q}) \right) + c_1 \frac{1}{n \lambda_{h,n}^\alpha} + c_2 \frac{\ln(3/\delta)}{n}, \quad (48)$$

with probability at least $1 - \delta/3$. Note that $T^\pi \hat{Q} \in \mathcal{F}^{|\mathcal{A}|}$ is implied by Assumption A7 and $\hat{Q} \in \mathcal{F}^{|\mathcal{A}|}$.

Because \hat{Q} is random itself, the terms $J(\hat{Q})$ and $J(T^\pi \hat{Q})$ in the upper bound of (48) are also random. In order to upper bound them, we use Lemma 18, which states that upon the choice of $\lambda_{h,n} = \lambda_{Q,n} = \lfloor \frac{1}{nJ^2(\hat{Q}^\pi)} \rfloor^{\frac{1}{1+\alpha}}$, there exists a constant $c_3 > 0$ such that for any $n \in \mathbb{N}$,

$$\lambda_{h,n} J^2(\hat{Q}) = \lambda_{Q,n} J^2(\hat{Q}) \leq \lambda_{Q,n} J^2(Q^\pi) + c_3 \frac{J^{\frac{2\alpha}{1+\alpha}}(Q^\pi)}{n^{\frac{1+\alpha}{1+\alpha}}} \ln(3/\delta) \quad (49)$$

holds with probability at least $1 - \delta/3$. We use Assumption A7 to show that we have

$$\lambda_{h,n} J^2(T^\pi \hat{Q}) \leq 2\lambda_{Q,n} L_R^2 + 2(\gamma L_P)^2 \left(\lambda_{Q,n} J^2(Q^\pi) + c_3 \frac{J^{\frac{2\alpha}{1+\alpha}}(Q^\pi)}{n^{\frac{1+\alpha}{1+\alpha}}} \ln(3/\delta) \right), \quad (50)$$

with the same probability. Plugging (49) and (50) into (48) and using the selected schedule for $\lambda_{Q,n}$ and $\lambda_{h,n}$, we get

$$\begin{aligned} \|\hat{h}_n(\cdot; \hat{Q}) - T^\pi \hat{Q}\|_\nu^2 &\leq \\ &\left[(2 + c_1 + 8(\gamma L_P)^2) J^{\frac{2\alpha}{1+\alpha}}(Q^\pi) + c_3 (2 + 8(\gamma L_P)^2) J^{\frac{2\alpha}{1+\alpha}}(Q^\pi) \ln(3/\delta) + \frac{8L_R^2}{J^{\frac{2\alpha}{1+\alpha}}(Q^\pi)} \frac{1}{n^{\frac{1+\alpha}{1+\alpha}}} \right. \\ &\quad \left. + c_2 \frac{\ln(3/\delta)}{n} \right] \end{aligned}$$

with probability at least $1 - \frac{2}{3}\delta$. By the proper choice of constants, the term $c_2 n^{-1} \ln(3/\delta)$ can be absorbed into $n^{\frac{1}{1+\alpha}} \ln(3/\delta)$. Therefore, there exists a constant $c_4 > 0$ such that

$$\|\hat{h}_n(\cdot; \hat{Q}) - T^\pi \hat{Q}\|_\nu^2 \leq \left[c_4 [1 + (\gamma L_P)^2] J^{\frac{2\alpha}{1+\alpha}}(Q^\pi) \ln(1/\delta) + \frac{8L_R^2}{[J(Q^\pi)]^{\frac{2}{1+\alpha}}} \frac{1}{n^{\frac{1+\alpha}{1+\alpha}}} \right], \quad (51)$$

with probability at least $1 - \frac{2}{3}\delta$.

Bounding $\|\hat{Q} - \hat{h}_n(\cdot; \hat{Q})\|_\nu$: With our choice of $\lambda_{Q,n}$ and $\lambda_{h,n}$, Lemma 19 states that there exists a constant $c_5 > 0$ such that for any $n \in \mathbb{N}$,

$$\|\hat{Q} - \hat{h}_n(\cdot; \hat{Q})\|_\nu^2 \leq c_5 \frac{(1 + \gamma^2 L_P^2)^\alpha J^{\frac{2\alpha}{1+\alpha}}(Q^\pi) \ln(1/\delta) + L_R^{\frac{2\alpha}{1+\alpha}}}{n^{\frac{1+\alpha}{1+\alpha}}}, \quad (52)$$

holds with probability at least $1 - \delta/3$.

Thus, inequality (47) alongside upper bounds (51) and (52) indicate that there exist constants $c_6, c_7 > 0$ such that for any $n \in \mathbb{N}$ and $\delta > 0$, we have

$$\|\hat{Q} - T^\pi \hat{Q}\|_{\infty}^2 \leq \frac{c_6 \left[1 + (\gamma L_P)^2 \right] J_{1+\frac{2}{1-\alpha}}(Q^\pi) \ln(1/\delta) + c_7 \left(\frac{L_R^2}{J(Q^\pi)^{1+\frac{2}{1-\alpha}}} + \frac{L_h^2}{[J(Q^\pi)]^{1+\frac{2}{1-\alpha}}} \right)}{n^{\frac{1}{1+\alpha}}},$$

with probability at least $1 - \delta$. ■

A careful study of the proof of Theorem 11 and the auxiliary results used in it reveals that one can indeed reuse a single data set in all iterations. Recall that at the k^{th} iteration of an API procedure such as REG-LSPI, the policy $\pi = \pi_k$ is the greedy policy w.r.t. $\hat{Q}^{(k-1)}$, so it depends on earlier data sets. This implies that a function such as $T^\pi \hat{Q} = T^{\pi(\cdot; \hat{Q}^{(k-1)})} \hat{Q}$ is random with two sources of randomness: One source is the data set used in the current iteration, which defines the empirical loss functions. This directly affects \hat{Q} . The other source is $\hat{\pi}(\cdot; \hat{Q}^{(k-1)})$, which depends on the data sets in earlier iterations. When we assume that all data sets are independent from each other, the randomness of π does not cause any problem because we can work on the probability space conditioned on the data sets of the earlier iterations. Conditioned on that randomness, the policy π becomes a deterministic function. This is how we presented the statement of Theorem 11 by stating that π is fixed. Nonetheless, the proofs can handle the dependence with no change. Briefly speaking, the reason is that when we want to provide a high probability upper bounds on certain random quantities, we take the supremum over both \hat{Q} and $T^\pi \hat{Q}$ and consider them as two separate functions, even though they are related through a random T^π operator.

To see this more clearly, notice that in the proof of Lemma 15, which is used in the proof of this theorem, we define the function spaces \mathcal{G}_l that chooses the functions h, Q , and $T^\pi Q$ separately. We then take the supremum over all functions in \mathcal{G}_l . This means that for the probabilistic upper bound, the randomness of π in $T^\pi Q$ becomes effectively irrelevant as we are providing a uniform over \mathcal{G}_l guarantee. In the proof of this theorem, we also use Lemma 19, which itself uses Theorem 16 and Lemma 21 that have a similar construct.

Appendix C. Proof of Lemma 15 (Convergence of $\hat{h}_n(\cdot; Q)$ to $T^\pi Q$)

The following lemma, quoted from Györfi et al. (2002), provides an exponential probability tail inequality for the relative deviation of the empirical mean from the true mean. A slightly modified version of this result was published as Theorem 2 of Kohler (2000). This result is used in the proof of Lemmas 15 and 21.

Lemma 22 (Theorem 19.3 of Györfi et al. 2002) *Let Z, Z_1, \dots, Z_n be independent and identically distributed random variables with values in \mathcal{Z} . Let $0 < \varepsilon < 1$ and $\eta > 0$. Assume that $K_1, K_2 \geq 1$ and let \mathcal{F} be a permissible class of functions $f: \mathcal{Z} \rightarrow \mathbb{R}$ with the following properties:*

- (A1) $\|f\|_{\infty} \leq K_1$,
- (A2) $\mathbb{E}[f(Z)^2] \leq K_2 \mathbb{E}[f(Z)]$,

(A3) $\sqrt{n\varepsilon} \sqrt{1 - \varepsilon} \sqrt{\eta} \geq 288 \max\{2K_1, \sqrt{2K_2}\}$,

(A4) For all $z_1, \dots, z_n \in \mathcal{Z}$ and all $\delta \geq \eta/8$,

$$\frac{\sqrt{n\varepsilon(1 - \varepsilon)\delta}}{96\sqrt{2} \max\{K_1, 2K_2\}} \geq \int_{\frac{\varepsilon(1 - \varepsilon)\delta}{16 \max\{K_1, 2K_2\}}}^{\sqrt{\delta}} \log \mathcal{N}_2 \left(u, \left\{ f \in \mathcal{F} : \frac{1}{n} \sum_{i=1}^n f^2(z_i) \leq 16\delta \right\}, z_{1:n} \right) du.$$

Then

$$\mathbb{P} \left\{ \sup_{f \in \mathcal{F}} \frac{|\mathbb{E}[f(Z)] - \frac{1}{n} \sum_{i=1}^n f(Z_i)|}{\eta + \mathbb{E}[f(Z)]} > \varepsilon \right\} \leq 60 \exp \left(- \frac{n \eta \varepsilon^2 (1 - \varepsilon)}{128 \times 2304 \max\{K_1^2, K_2\}} \right).$$

Let us now turn to the proof of Lemma 15. This proof follows similar steps to the proof of Theorem 21.1 of Györfi et al. (2002).

Proof [Proof of Lemma 15] Without loss of generality, assume that $Q_{\max} \geq 1/2$. Denote $z = (x, a)$ and let $Z = (X, A) \sim \nu$, $R \sim \mathcal{R}(\cdot|X, A)$, and $X' \sim P(\cdot|X, A)$ be random variables that are independent of $\mathcal{D}_n = \{(X_i, A_i, R_i, X'_i)\}_{i=1}^n$. Define the following error decomposition

$$\begin{aligned} \int_{\mathcal{X} \times \mathcal{A}} |\hat{h}_n(z; Q) - T^\pi Q(z)|^2 d\nu(z) &= \mathbb{E} \left[|\hat{h}_n(Z; Q) - [R + \gamma Q(X', \pi(X'))]|^2 \middle| \mathcal{D}_n \right] - \\ &= \mathbb{E} \left[|T^\pi Q(Z) - [R + \gamma Q(X', \pi(X'))]|^2 \right] \\ &= I_{1,n} + I_{2,n}, \end{aligned}$$

with

$$\begin{aligned} \frac{1}{2} I_{1,n} &= \frac{1}{n} \sum_{i=1}^n |\hat{h}_n(Z_i; Q) - [R_i + \gamma Q(X'_i, \pi(X'_i))]|^2 - |T^\pi Q(Z_i) - [R_i + \gamma Q(X'_i, \pi(X'_i))]|^2 + \\ &\quad \lambda_{h,n} \left(J^2(\hat{h}_n(\cdot; Q)) + J^2(Q) + J^2(T^\pi Q) \right), \\ I_{2,n} &= \mathbb{E} \left[|\hat{h}_n(Z; Q) - \hat{T}^\pi Q(Z)|^2 - |T^\pi Q(Z) - \hat{T}^\pi Q(Z)|^2 \middle| \mathcal{D}_n \right] - I_{1,n}. \end{aligned}$$

By the optimizer property of $\hat{h}_n(\cdot; Q)$, we get the following upper bound by substituting $\hat{h}_n(\cdot; Q)$ with $T^\pi Q \in \mathcal{F}^{-|A|}$:

$$\begin{aligned} I_{1,n} &\leq 2 \left[\frac{1}{n} \sum_{i=1}^n |T^\pi Q(Z_i) - \hat{T}^\pi Q(Z_i)|^2 - |T^\pi Q(Z_i) - \hat{T}^\pi Q(Z_i)|^2 \right] + \\ &\quad \lambda_{h,n} (J^2(T^\pi Q) + J^2(Q) + J^2(T^\pi Q)) \\ &= 4\lambda_{h,n} J^2(T^\pi Q) + 2\lambda_{h,n} J^2(Q). \end{aligned} \tag{53}$$

We now turn to upper bounding $\mathbb{P}\{I_{2,n} > t\}$. Given a policy π and functions $h, Q, Q' \in \mathcal{F}^{|\mathcal{A}|}$, for $w = (x, a, r, x')$ define $g : \mathcal{X} \times \mathcal{A} \times \mathbb{R} \times \mathbb{R} \times \mathcal{X} \rightarrow \mathbb{R}$ as

$$g_{h,Q,Q'}(w) = |h(z) - [r + \gamma Q(a', \pi(x'))]|^2 - |Q'(z) - [r + \gamma Q(a', \pi(x'))]|^2.$$

Note that $g_{h_n, Q, Q'}(w)$ is the function appearing in the definition of $I_{2,n}$. Define the following function spaces for $l = 0, 1, \dots$:

$$\mathcal{G}_l \triangleq \left\{ g_{h,Q,Q'} : \mathcal{X} \times \mathcal{A} \times \mathbb{R} \times \mathcal{X} \rightarrow \mathbb{R} : h, Q, Q' \in \mathcal{F}^{|\mathcal{A}|}, J^2(h), J^2(Q), J^2(Q') \leq \frac{2^l t}{\lambda_{h,n}} \right\}.$$

Denote $W = (X, A, R, X')$ and $W_i = (X_i, A_i, R_i, X'_i)$. Apply the peeling device to get

$$\begin{aligned} \mathbb{P}\{I_{2,n} > t\} &\leq \sum_{l=0}^{\infty} \mathbb{P}\left(\exists h, Q \in \mathcal{F}^{|\mathcal{A}|}, 2^l \mathbb{1}_{\{t \neq 0\}} \leq 2\lambda_{h,n} (J^2(h) + J^2(Q)) + J^2(T^\pi Q)\right) < 2^{l+1} t; \\ &\quad \text{s.t. } \frac{\mathbb{E}[g_{h,Q,T^\pi Q}(W)|\mathcal{D}_n] - \frac{1}{n} \sum_{i=1}^n g_{h,Q,T^\pi Q}(W_i)}{t + 2\lambda_{h,n} (J^2(h) + J^2(Q)) + J^2(T^\pi Q)} + \mathbb{E}[g_{h,Q'}(W)|\mathcal{D}_n] > \frac{1}{2} \\ &\leq \sum_{l=0}^{\infty} \mathbb{P}\left(\sup_{g \in \mathcal{G}_l} \frac{\mathbb{E}[g(W)|\mathcal{D}_n] - \frac{1}{n} \sum_{i=1}^n g(W_i)}{2^l t + \mathbb{E}[g(W)|\mathcal{D}_n]} > \frac{1}{2}\right). \end{aligned}$$

Here we used the simple fact that if $2\lambda_{h,n} (J^2(h) + J^2(Q)) + J^2(T^\pi Q) < 2^{l+1} t$, then $J^2(h)$, $J^2(Q)$, and $J^2(T^\pi Q)$ are also less than $\frac{2^l t}{\lambda_{h,n}}$, so $g_{h,Q,T^\pi Q} \in \mathcal{G}_l$.

We study the behavior of the l th term of the above summation by verifying the conditions of Lemma 22—similar to what we did in the proof of Lemma 21.

It is easy to verify that (A1) and (A2) are satisfied with the choice of $K_1 = K_2 = 4Q_{\max}^2$. Condition (A3) is satisfied whenever

$$t \geq \frac{c_1}{n}, \quad (54)$$

for some constant $c_1 > 0$ depending on Q_{\max} (the constant can be set to $c_1 = 2 \times 4608^2 Q_{\max}^2$).

To verify condition (A4), we first require an upper bound on $N_2^2(u, \mathcal{G}_l, w_{1:n})$ for any sequence $w_{1:n}$. This can be done similar to the proof of Lemma 20: Denote $\mathcal{F}_l = \{f : f \in \mathcal{F}, J^2(f) \leq \frac{2^l t}{\lambda_{h,n}}\}$. For $g_{h_1, Q_1, T^\pi Q_1}, g_{h_2, Q_2, T^\pi Q_2} \in \mathcal{G}_l$ and any sequence $w_{1:n}$ we have

$$\begin{aligned} &\frac{1}{2} \sum_{i=1}^n |g_{h_1, Q_1, T^\pi Q_1}(w_i) - g_{h_2, Q_2, T^\pi Q_2}(w_i)|^2 \\ &\leq 12(2 + \gamma)^2 Q_{\max}^2 \frac{1}{n} \sum_{i=1}^n \left[|h_1(z_i) - h_2(z_i)|^2 + 4\gamma^2 |Q_1(x'_i, \pi(x'_i)) - Q_2(x'_i, \pi(x'_i))|^2 + \right. \\ &\quad \left. |T^\pi Q_1(z_i) - T^\pi Q_2(z_i)|^2 \right] \\ &\leq 12(2 + \gamma)^2 Q_{\max}^2 \frac{1}{n} \sum_{a \in \mathcal{A}} \sum_{i=1}^n \left[|h_1(x_i, a) - h_2(x_i, a)|^2 + 4\gamma^2 |Q_1(x'_i, a) - Q_2(x'_i, a)|^2 + \right. \\ &\quad \left. |T^\pi Q_1(x_i, a) - T^\pi Q_2(x_i, a)|^2 \right]. \end{aligned}$$

With the same covering set argument as in the proof of Lemma 20, we get that for any $u > 0$,

$$N_2(\sqrt{2|\mathcal{A}|} Q_{\max} u, \mathcal{G}_l, w_{1:n}) \leq N_2^2(u, \mathcal{F}_l, x_{1:n})^{|\mathcal{A}|} \times N_2^2(u, \mathcal{F}_l, x'_{1:n})^{|\mathcal{A}|} \times N_2^2(u, \mathcal{F}_l, x_{1:n})^{|\mathcal{A}|}.$$

Invoke Assumption A4 to get

$$\log N_2(u, \mathcal{G}_l, w_{1:n}) \leq c(|\mathcal{A}|, Q_{\max}) \left(\frac{2^l t}{\lambda_{h,n}} \right)^\alpha u^{-2\alpha}.$$

Plugging this covering number result into condition (A4), one can verify that the condition is satisfied if

$$t \geq \frac{c_2}{n \lambda_{h,n}^\alpha}, \quad (55)$$

for a constant $c_2 > 0$, which is only a function of Q_{\max} and $|\mathcal{A}|$. Therefore, Lemma 22 indicates that

$$\mathbb{P}\{I_{2,n} > t\} \leq 60 \sum_{l=0}^{\infty} \exp\left(-\frac{n(2^l t)(1/4)(1/2)}{128 \times 2304 \times \max\{16Q_{\max}^4, 4Q_{\max}^2\}}\right) \leq c_3 \exp(-c_4 n t). \quad (56)$$

for some constants $c_3, c_4 > 0$.

Combining (53), (54), (55), and (56), we find that there exist $c_5, c_6 > 0$ such that for any $n \in \mathbb{N}$ and $0 < \delta < 1$, we have

$$\| \hat{h}_n(Q) - T^\pi Q \|_{\mu}^2 \leq 4\lambda_{h,n} J^2(T^\pi Q) + 2\lambda_{h,n} J^2(Q) + c_5 \frac{1}{n \lambda_{h,n}^\alpha} + c_6 \frac{\ln(1/\delta)}{n}.$$

Here, c_5 is only a function of Q_{\max} and $|\mathcal{A}|$, and c_6 is a function of Q_{\max} . ■

Appendix D. Proof of Theorem 16 (Empirical Error and Smoothness of $\hat{h}_n(\cdot; Q)$)

To prove Theorem 16, which is a modification of Theorem 10.2 by van de Geer (2000), we first need to modify and specialize Lemma 3.2 by van de Geer (2000) to be suitable to our problem. The modification is required because Q in (33) is a random function in $\mathcal{F}^{|\mathcal{A}|}$ as opposed to being a fixed function as in Theorem 10.2 of van de Geer (2000).

Let us denote $z = (x, a) \in \mathcal{Z} = \mathcal{X} \times \mathcal{A}$ and $Z' = (x, a, R, X') \in \mathcal{Z}' = \mathcal{X} \times \mathcal{A} \times \mathbb{R} \times \mathcal{X}$ with $(R, X') \sim P(\cdot, \cdot | x, a)$. Let \mathcal{D}_n denote the set $\{(x_i, a_i, R_i, X'_i)\}_{i=1}^n$ of independent random variables. We use z_i to refer to (x_i, a_i) and Z'_i to refer to (x_i, a_i, R_i, X'_i) . Let P_n be the probability measure that puts mass $1/n$ on z_1, \dots, z_n , i.e., $P_n = \frac{1}{n} \sum_{i=1}^n \delta_{z_i}$, in which δ_z is the Dirac's delta function that puts a mass of 1 at z .

Denote $\mathcal{G} : \mathcal{Z} \rightarrow \mathbb{R}$ and $\mathcal{G}' : \mathcal{Z}' \rightarrow \mathbb{R}^{3|\mathcal{A}|}$, which is defined as the set

$$\mathcal{G}' = \left\{ (Q, T^\pi Q, \mathbf{1}) : Q \in \mathcal{F}^{|\mathcal{A}|} \right\}$$

with $\mathbf{1} : \mathcal{X} \times \mathcal{A} \rightarrow \mathbb{R}^{|\mathcal{A}|}$ being a bounded constant function (and not necessarily equal to 1). We use $\|g\|_\infty$ to denote the supremum norm of functions in \mathcal{G} . The supremum norm of vector-valued functions in \mathcal{G}' is defined by taking the supremum norm over the l_∞ -norm of each vector. Similarly, the supremum norm of $(g, g') \in \mathcal{G} \times \mathcal{G}'$ is defined by $\|(g, g')\|_\infty \triangleq \max\{\|g\|_\infty, \|g'\|_\infty\}$.

For $g \in \mathcal{G}$, we define $\|g\|_{P_n} \triangleq [\frac{1}{n} \sum_{i=1}^n g^2(z_i)]^{1/2}$. To simplify the notation, we use the following definition of the inner product: Fix $n \in \mathbb{N}$. Consider z_1, \dots, z_n as a set of points in \mathcal{Z} , and a real-valued sequence $w = (w_1, \dots, w_n)$. For a function $g \in \mathcal{G}$, define $\langle w, g \rangle_n \triangleq \frac{1}{n} \sum_{i=1}^n w_i g(z_i)$.

For any $g' = (Q, T^\pi Q, \mathbf{1}) \in \mathcal{G}'$, define the mapping $\bar{W}(g')(x, a, r, x') : \mathcal{X} \times \mathcal{A} \times \mathbb{R} \times \mathcal{X} \rightarrow \mathbb{R}$ by $\bar{W}(g')(x, a, r, x') = r + \gamma Q(x', \pi(x')) - T^\pi Q(x, a)$. For any fixed $g' \in \mathcal{G}'$ and $i = 1, \dots, n$, define the random variables $W_i(g') = \bar{W}(g')(Z_i)$ and let $W(g')$ denote the random vector $[W_1(g') \dots W_n(g')]^\top$. Notice that $W_i(g')$ can be re-written as $W_i(g') = (R_i - r(z_i)) + \gamma(Q(X_i, \pi(X_i)) - (P^\pi Q)(z_i))$, thus for any fixed g' , $\mathbb{E}[W_i(g')] = 0$ ($i = 1, \dots, n$). For notational simplification, we use $a \vee b = \max\{a, b\}$.

Lemma 23 (Modified Lemma 3.2 of van de Geer 2000) Fix the sequence $(z_i)_{i=1}^n \subset \mathcal{Z}$ and let $(Z_i)_{i=1}^n \subset \mathcal{Z}'$ be the sequence of independent random variables defined as above. Assume that for some constants $0 < R \leq L$, it holds that $\sup_{g \in \mathcal{G}} \|g\|_{P_n} \leq R$, $\sup_{g' \in \mathcal{G}'} \|g'\|_\infty \leq L$, and $|R_i| \leq L$ ($1 \leq i \leq n$) almost surely. There exists a constant C such that for all $0 \leq \varepsilon < \delta$ satisfying

$$\sqrt{n}(\delta - \varepsilon) \geq C L \left[\int_{\frac{\varepsilon \delta}{25L}}^R \log \mathcal{N}_\infty(u, \mathcal{G} \times \mathcal{G}')^{1/2} du \vee R \right], \quad (57)$$

we have

$$\mathbb{P} \left\{ \sup_{(g, g') \in \mathcal{G} \times \mathcal{G}'} \left| \frac{1}{n} \sum_{i=1}^n W_i(g') g(z_i) \right| \geq \delta \right\} \leq 4 \exp \left(- \frac{n(\delta - \varepsilon)^2}{27 \times 3^5 (RL)^2} \right).$$

The main difference between this lemma and Lemma 3.2 of van de Geer (2000) is that the latter provides a maximal inequality for $\sup_{g \in \mathcal{G}} \frac{1}{n} \sum_{i=1}^n W_i g(z_i)$, with W_i being random variables that satisfy a certain exponential probability inequality, while our result is a maximal inequality for $\sup_{(g, g') \in \mathcal{G} \times \mathcal{G}'} \frac{1}{n} \sum_{i=1}^n W_i(g') g(z_i)$, i.e., the random variables $W_i(g')$ are functions of an arbitrary $g' \in \mathcal{G}'$. The current proof requires us to have a condition on the metric entropy w.r.t. the supremum norm (cf. (57)) instead of w.r.t. the empirical L_2 -norm used in Lemma 3.2 of van de Geer (2000). The possibility of relaxing this requirement is an interesting question. We now prove this result.

Proof First, note that for any $g_1, g_2 \in \mathcal{G}$, and $g'_1, g'_2 \in \mathcal{G}'$ (with the identification of g' with its corresponding Q and $T^\pi Q$), we have

$$\begin{aligned} & \frac{1}{n} \sum_{i=1}^n W_i(g'_1) g_1(z_i) - W_i(g'_2) g_2(z_i) = \\ & \frac{1}{n} \sum_{i=1}^n (R_i - r(z_i))(g_1(z_i) - g_2(z_i)) + \\ & \frac{1}{n} \sum_{i=1}^n \gamma [(Q_1(X'_i, \pi(X'_i)) - P^\pi Q_1(z_i)) - (Q_2(X'_i, \pi(X'_i)) - P^\pi Q_2(z_i))] g_1(z_i) + \\ & \frac{1}{n} \sum_{i=1}^n \gamma (Q_2(X'_i, \pi(X'_i)) - P^\pi Q_2(z_i))(g_1(z_i) - g_2(z_i)) \leq \\ & 2L \|g_1 - g_2\|_{P_n} + \gamma R \|Q_1 - Q_2\|_\infty + \|P^\pi Q_1 - P^\pi Q_2\|_\infty + 3\gamma L \|g_1 - g_2\|_{P_n} = \\ & (2 + 3\gamma)L \|g_1 - g_2\|_{P_n} + \gamma R \|Q_1 - Q_2\|_\infty + R \|T^\pi Q_1 - T^\pi Q_2\|_\infty, \end{aligned} \quad (58)$$

where we used the boundedness assumptions, the definition of the supremum norm, the norm inequality $\frac{1}{n} \sum_{i=1}^n |g_1(z_i) - g_2(z_i)| \leq \|g_1 - g_2\|_{P_n}$, and the fact that $|\gamma(Q(X'_i, \pi(X'_i)) - P^\pi Q(z_i))| = |r(z_i) + \gamma(Q(X'_i, \pi(X'_i)) - T^\pi Q(z_i))| \leq (2 + \gamma)L \leq 3L$ for any L -bounded Q and $T^\pi Q$ to get the inequality. We used $\|P^\pi Q^s - P^\pi Q^{s-1}\|_\infty = \gamma^{-1} \|T^\pi Q^s - T^\pi Q^{s-1}\|_\infty$ to get the last equality.

Let $\{(g'_j, g_j^s)\}_{j=1}^{N_s}$ with $N_s = \mathcal{N}_\infty(2^{-s}R, \mathcal{G} \times \mathcal{G}')$ be a minimal $2^{-s}R$ -covering of $\mathcal{G} \times \mathcal{G}'$ w.r.t. the supremum norm. For any $(g, g') \in \mathcal{G} \times \mathcal{G}'$, there exists a $(g^s, (Q^s, T^\pi Q^s, \mathbf{1})) = (g^s, g^s) \in \{(g'_j, g_j^s)\}_{j=1}^{N_s}$ such that $\|(g, g') - (g^s, g^s)\|_\infty \leq 2^{-s}R$. This implies that both $\|Q^s - Q\|_\infty$ and $\|T^\pi Q^s - T^\pi Q\|_\infty$ are smaller than $2^{-s}R$ as well. Moreover, $\|g^s - g\|_{P_n} \leq \|g^s - g\|_\infty \leq 2^{-s}R$. By (58) we get

$$\begin{aligned} \left| \frac{1}{n} \sum_{i=1}^n W_i(g^s) g^s(z_i) - W_i(g) g(z_i) \right| & \leq [(2 + 3\gamma)L + (1 + \gamma)R](2^{-s}R) \leq (3 + 4\gamma)L(2^{-s}R) \\ & \leq 7RL 2^{-s}. \end{aligned}$$

Choose $S = \min\{s \geq 1 : 2 \cdot 2^{-s} \leq \frac{\varepsilon}{7RL}\}$, which entails that for any $(g, g') \in \mathcal{G} \times \mathcal{G}'$, the covering set defined by $\{(g'_j, g_j^s)\}_{j=1}^{N_s}$ approximates the inner product of $[g(z_1) \dots g(z_n)]^\top$ and $W(g')$ with an error less than ε . So it suffices to prove the exponential inequality for

$$\mathbb{P} \left\{ \max_{j=1, \dots, N_s} \left| \frac{1}{n} \sum_{i=1}^n W_i(g'_j) g_j^s(z_i) \right| \geq \delta - \varepsilon \right\}.$$

We use the chaining technique (e.g., see [van de Geer 2000](#)) as follows (we choose $g^0 = 0$, so $W_i(g^0)g^0(z_i) = 0$ for all $1 \leq i \leq n$):

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n W_i(g^S)g^S(z_i) &= \frac{1}{n} \sum_{i=1}^n \sum_{s=1}^S (W_i(g^s)g^s(z_i) - W_i(g^{s-1})g^{s-1}(z_i)) \\ &= \sum_{s=1}^S \left[\frac{1}{n} \sum_{i=1}^n (R_i - r(z_i))(g^s(z_i) - g^{s-1}(z_i)) + \right. \\ &\quad \left. \frac{1}{n} \sum_{i=1}^n \gamma [(Q^s(X'_i, \pi(X'_i)) - (P^\pi Q^s)(z_i)) - (Q^{s-1}(X'_i, \pi(X'_i)) - (P^\pi Q^{s-1})(z_i))] g^s(z_i) + \right. \\ &\quad \left. \frac{1}{n} \sum_{i=1}^n \gamma (Q^{s-1}(X'_i, \pi(X'_i)) - (P^\pi Q^{s-1})(z_i))(g^s(z_i) - g^{s-1}(z_i)) \right]. \end{aligned}$$

Because each of these summations consists of bounded random variables with expectation zero, we may use Hoeffding's inequality alongside the union bound to upper bound them. To apply Hoeffding's inequality, we require an upper bound on the sum of squared values of random variables involved. To begin, we have $|g^s(z_i) - g^{s-1}(z_i)| = |g^s(z_i) - g(z_i) + g(z_i) - g^{s-1}(z_i)| \leq 2^{-s}R + 2^{-(s-1)}R = 3 \times 2^{-s}R$. Similarly, both $\|Q^s - Q^{s-1}\|_\infty$ and $\|T^\pi Q^s - T^\pi Q^{s-1}\|_\infty$ are smaller than $3 \times 2^{-s}R$. As a result, for the first term we get

$$\frac{1}{n} \sum_{i=1}^n [(R_i - r(z_i))(g^s(z_i) - g^{s-1}(z_i))]^2 \leq 36(RL)^2 2^{-2s}.$$

For the second term we have

$$\begin{aligned} &\frac{1}{n} \sum_{i=1}^n |\gamma [(Q^s(X'_i, \pi(X'_i)) - (P^\pi Q^s)(z_i)) - (Q^{s-1}(X'_i, \pi(X'_i)) - (P^\pi Q^{s-1})(z_i))] g^s(z_i)|^2 \\ &\leq 2\gamma^2 \left[\|Q^s - Q^{s-1}\|_\infty^2 + \gamma^{-2} \|T^\pi Q^s - T^\pi Q^{s-1}\|_\infty^2 \right] \|g^s\|_{P_n}^2 \\ &\leq 2(1 + \gamma^2) 3^2 (2^{-s}R)^2 R^2 \leq 36R^4 2^{-2s}, \end{aligned}$$

in which we used $\|P^\pi Q^s - P^\pi Q^{s-1}\|_\infty = \gamma^{-1} \|T^\pi Q^s - T^\pi Q^{s-1}\|_\infty$. And finally,

$$\begin{aligned} &\frac{1}{n} \sum_{i=1}^n |\gamma (Q^{s-1}(X'_i, \pi(X'_i)) - (P^\pi Q^{s-1})(z_i))(g^s(z_i) - g^{s-1}(z_i))|^2 \leq (3L)^2 3^2 (2^{-s}R)^2 \\ &= g^s(RL)^2 2^{-2s}, \end{aligned}$$

where we used the fact that $|\gamma Q(X'_i, \pi(X'_i)) - \gamma P^\pi Q(z_i)| \leq 3L$ for any L -bounded Q and $T^\pi Q$.

Let η_s be a sequence of positive real-valued numbers satisfying $\sum_{s=1}^S \eta_s \leq 1$. We continue the chaining argument by the use of the union bound and the fact that $N_s N_{s-1} \leq N_s^2$ to

get

$$\begin{aligned} P_1 &= \mathbb{P} \left\{ \sup_{(g,g') \in \mathcal{G} \times \mathcal{G}'} \left| \frac{1}{n} \sum_{i=1}^n W_i(g)g(z_i) \right| \geq \delta \right\} \\ &\leq \mathbb{P} \left\{ \max_{j=1, \dots, N_s} \left| \frac{1}{n} \sum_{i=1}^n W_i(g_j^S)g_j^S(z_i) \right| \geq \delta - \varepsilon \right\} \\ &\leq \sum_{s=1}^S \mathbb{P} \left\{ \max_{(g^s, g'^s)} \left| \frac{1}{n} \sum_{i=1}^n (R_i - r(z_i))(g^s(z_i) - g^{s-1}(z_i)) \right| \geq \frac{\eta_s(\delta - \varepsilon)}{3} \right\} + \\ &\quad \mathbb{P} \left\{ \max_{(g^s, g'^s)} \left| \frac{1}{n} \sum_{i=1}^n \gamma [(Q^s(X'_i, \pi(X'_i)) - (P^\pi Q^s)(z_i)) - \right. \right. \\ &\quad \left. \left. (Q^{s-1}(X'_i, \pi(X'_i)) - (P^\pi Q^{s-1})(z_i))] g^s(z_i) \right| \geq \frac{\eta_s(\delta - \varepsilon)}{3} \right\} + \\ &\quad \mathbb{P} \left\{ \max_{(g^s, g'^s)} \left| \frac{1}{n} \sum_{i=1}^n \gamma (Q^{s-1}(X'_i, \pi(X'_i)) - (P^\pi Q^{s-1})(z_i))(g^s(z_i) - g^{s-1}(z_i)) \right| \geq \frac{\eta_s(\delta - \varepsilon)}{3} \right\} \end{aligned}$$

$$\begin{aligned} &\mathbb{P} \left\{ \max_{(g^s, g'^s)} \left| \frac{1}{n} \sum_{i=1}^n \gamma (Q^{s-1}(X'_i, \pi(X'_i)) - (P^\pi Q^{s-1})(z_i))(g^s(z_i) - g^{s-1}(z_i)) \right| \geq \frac{\eta_s(\delta - \varepsilon)}{3} \right\} \\ &\leq \sum_{s=1}^S N_s N_{s-1} \exp \left(-\frac{2(\delta - \varepsilon)^2 \eta_s^2 n}{4 \times g^2 (RL)^2 2^{-2s}} \right) + N_s N_{s-1} \exp \left(-\frac{2(\delta - \varepsilon)^2 \eta_s^2 n}{4 \times g^2 R^2 2^{-2s}} \right) \\ &\leq \sum_{s=1}^S 3 \exp \left(-\frac{2(\delta - \varepsilon)^2 \eta_s^2 n}{3 \times g^2 (RL)^2 2^{-2s}} \right) \\ &\quad N_s N_{s-1} \exp \left(-\frac{2(\delta - \varepsilon)^2 \eta_s^2 n}{3 \times g^2 (RL)^2 2^{-2s}} \right). \end{aligned} \tag{59}$$

Here the $\max_{(g^s, g'^s)}$ is over the corresponding covering set $\{(g_j^s, g_j^s)\}_{j=1}^{N_s}$, which has N_s elements (and the same for $s-1$).

Choose

$$\eta_s = \frac{3^2 \sqrt{6RL} 2^{-s} (\log N_s)^{1/2}}{\sqrt{n}(\delta - \varepsilon)} \vee \frac{2^{-s} \sqrt{s}}{8}. \tag{60}$$

Take C in (57) sufficiently large such that

$$\sqrt{n}(\delta - \varepsilon) \geq 2 \times 3^2 \sqrt{6RL} \sum_{s=1}^S 2^{-s} [\log N_\infty (2^{-s}R, \mathcal{G} \times \mathcal{G}')]^{1/2} \vee 72 \sqrt{6 \log 4 RL}. \tag{61}$$

It can be shown that by this choice of η_s and the condition (61), we have $\sum_{s=1}^S \eta_s \leq 1$.

From (60), we have $\log N_s \leq \frac{n(\delta - \varepsilon)^2 \eta_s^2}{2 \times 3^2 (RL)^2 2^{-2s}}$, so P_1 in (59) can be upper bounded as follows

$$P_1 \leq \sum_{s=1}^S 3 \exp \left(-\frac{n(\delta - \varepsilon)^2 \eta_s^2}{2 \times 3^5 (RL)^2 2^{-2s}} \right).$$

Since $\eta_s \geq 2^{-s}\sqrt{s}/8$ too, we have

$$\begin{aligned} P_1 &\leq 3 \sum_{s=1}^S \exp\left(-\frac{n(\delta-\varepsilon)^2 2^{-2s}}{27 \times 3^5 (RL)^2 2^{-2s}}\right) \leq 3 \sum_{s=1}^{\infty} \exp\left(-\frac{n(\delta-\varepsilon)^2 s}{27 \times 3^5 (RL)^2}\right) \\ &\leq \frac{3 \exp\left(-\frac{n(\delta-\varepsilon)^2}{27 \times 3^5 (RL)^2}\right)}{1 - \exp\left(-\frac{n(\delta-\varepsilon)^2}{27 \times 3^5 (RL)^2}\right)} \leq 4 \exp\left(-\frac{n(\delta-\varepsilon)^2}{27 \times 3^5 (RL)^2}\right), \end{aligned}$$

where in the last inequality we used the assumption that $\sqrt{n}(\delta-\varepsilon) \geq 72\sqrt{6}\log 4 RL$ (cf. (61)).

One can show that (61) is satisfied if

$$\sqrt{n}(\delta-\varepsilon) \geq 36\sqrt{6}L \int_{\frac{\varepsilon}{32L}}^R |\log \mathcal{N}_{\infty}(u, \mathcal{G} \times \mathcal{G}')^{1/2} du| \vee 72\sqrt{6}\log 4 RL,$$

so C can be chosen as $C = 72\sqrt{6}\log 4$. \blacksquare

The following lemma, which is built on Lemma 23, is a result on the behavior of the modulus of continuity and will be used in the proof of Theorem 16. This lemma provides a high-probability upper bound on $\sup_{(g,g') \in \mathcal{G} \times \mathcal{G}'} \frac{|(W(g), g)_n|}{\|g\|_{P_n}^{1-\alpha} J^{\alpha}(g,g')}$. Here $J(g, g')$ is a regularizer that is defined on $\mathcal{G} \times \mathcal{G}'$ and is a pseudo-norm.

This result is similar in spirit to Lemma 8.4 of van de Geer (2000), with two main differences: The first is that here we provide an upper bound on

$$\sup_{(g,g') \in \mathcal{G} \times \mathcal{G}'} \frac{|(W(g'), g)_n|}{\|g\|_{P_n}^{1-\alpha} J^{\alpha}(g, g')},$$

whereas in Lemma 8.4 of van de Geer (2000), the upper bound is on

$$\sup_{g \in \mathcal{G}} \frac{|(W, g)_n|}{\|g\|_{P_n}^{1-\alpha}}.$$

The normalization by $\|g\|_{P_n}^{1-\alpha} J^{\alpha}(g, g')$ instead of $\|g\|_{P_n}^{1-\alpha}$ is important to get the right error bound in Theorem 16. The other crucial difference is that here W are random variables that are functions of $g' \in \mathcal{G}'$, while the result of van de Geer (2000) is for independent W . The proof technique is inspired by Lemmas 5.13, 5.14, and 8.4 of van de Geer (2000).

Lemma 24 (Modulus of Continuity for Weighted Sums) Fix the sequence $(z_i)_{i=1}^n \subset \mathcal{Z}$ and define $P_n = \frac{1}{n} \sum_{i=1}^n \delta_{z_i}$. Let $(Z_i)_{i=1}^n \subset \mathcal{Z}'$ be the sequence of independent random variables defined as before. Assume that for some constant $L > 0$, it holds that $\sup_{g \in \mathcal{G}} \|g\|_{P_n} \leq L$, $\sup_{g \in \mathcal{G}'} \|g'\|_{\infty} \leq L$, and $|R_i| \leq L$ ($1 \leq i \leq n$) almost surely. Furthermore, suppose that there exist $0 < \alpha < 1$ and a finite constant A such that for all $u > 0$,

$$\log \mathcal{N}_{\infty}(u, \{ (g, g') \in \mathcal{G} \times \mathcal{G}' : J(g, g') \leq B \}) \leq A \left(\frac{B}{u}\right)^{2\alpha}.$$

Then there exists a constant $c > 0$ such that for any $0 < \delta < 1$, we have

$$\sup_{(g,g') \in \mathcal{G} \times \mathcal{G}'} \frac{|(W(g'), g)_n|}{\|g\|_{P_n}^{1-\alpha} J^{\alpha}(g, g')} \leq cL^{1+\alpha} \sqrt{\frac{\ln(\frac{1}{\delta})}{n}},$$

with probability at least $1 - \delta$.

Proof The proof uses double-peeling, i.e., we peel on both $J(g, g')$ and $\|g\|_{P_n}$. Without loss of generality, we assume that $L \geq 1$. We use $c_1, c_2, \dots > 0$ as constants. First, we start by peeling on $J(g, g')$:

$$\begin{aligned} \delta &\triangleq \mathbb{P} \left\{ \sup_{(g,g') \in \mathcal{G} \times \mathcal{G}'} \frac{|(W(g'), g)_n|}{\|g\|_{P_n}^{1-\alpha} J^{\alpha}(g, g')} \geq t \right\} \\ &\leq \sum_{s=0}^{\infty} \mathbb{P} \left\{ \sup_{(g,g') \in \mathcal{G} \times \mathcal{G}'} \frac{|(W(g'), g)_n|}{\|g\|_{P_n}^{1-\alpha}} \geq \underbrace{t \cdot 2^{os}}_{\triangleq \tau_s}, 2^s \mathbb{I}_{\{s \neq 0\}} \leq J(g, g') < 2^{s+1} \right\}. \end{aligned} \quad (62)$$

Let us denote each term in the RHS by δ_s . To upper bound δ_s , notice that by assumption $\|g\|_{P_n} \leq L$. For each term, we peel again, this time on $\|g\|_{P_n}$, and apply Lemma 23:

$$\begin{aligned} \delta_s &\leq \sum_{r \geq 0} \mathbb{P} \left\{ \sup_{(g,g') \in \mathcal{G} \times \mathcal{G}'} \frac{|(W(g'), g)_n|}{\|g\|_{P_n}^{1-\alpha}} \geq \tau_s, 2^s \mathbb{I}_{\{s \neq 0\}} \leq J(g, g') < 2^{s+1}, \right. \\ &\quad \left. 2^{-(r+1)}L < \|g\|_{P_n} \leq 2^{-r}L \right\} \\ &\leq \sum_{r \geq 0} \mathbb{P} \left\{ \sup_{(g,g') \in \mathcal{G} \times \mathcal{G}'} |(W(g'), g)_n| \geq \tau_s \left(2^{-(r+1)}L\right)^{1-\alpha}, J(g, g') < 2^{s+1}, \|g\|_{P_n} \leq 2^{-r}L \right\} \\ &\leq \sum_{r \geq 0} 4 \exp \left(-\frac{n \left[\tau_s \left(2^{-(r+1)}L\right)^{1-\alpha} \right]^2}{27 \times 3^5 \left(2^{-r}L\right)^2 L^2} \right) \\ &= \sum_{r \geq 0} 4 \exp \left(-\frac{2^{2r\alpha} n \tau_s^2}{27 \times 3^5 \times 2^{2(1-\alpha)} L^{2(1+\alpha)}} \right) = c_2 \exp \left(-\frac{c_1 n \tau_s^2}{L^{2(1+\alpha)}} \right). \end{aligned} \quad (63)$$

The last inequality holds only if the covering number condition in Lemma 23 is satisfied, which is the case whenever

$$\sqrt{n} \left(\tau_s \left(2^{-(r+1)}L\right)^{1-\alpha} \right) \geq cL \left[\int_0^{2^{-r}L} \sqrt{A} \left(\frac{2^{s+1}}{u}\right)^{\alpha} du \vee 2^{-r}L \right].$$

Substituting $\tau_s = 2^{os}t$ and solving the integral, we get that the condition is

$$\sqrt{nt} 2^{os} \left(2^{-(r+1)}L\right)^{1-\alpha} \geq cL \sqrt{A} \left[(2^{s+1})^{\alpha} \left(2^{-r}L\right)^{1-\alpha} \vee 2^{-r}L\right],$$

which would be satisfied for

$$t \geq \frac{cL \sqrt{A} 2^{1+\alpha}}{\sqrt{n}} \vee \frac{2^{1-\alpha} cL^{1+\alpha}}{\sqrt{n}} = c_3 \frac{L^{1+\alpha}}{\sqrt{n}}. \quad (64)$$

Plug (63) into (62) to get that

$$\delta \leq \sum_{s=0}^{\infty} c_2 \exp\left(-\frac{c_1 n \ell^2 2^{2\alpha s}}{L^{2(1+\alpha)}}\right) = c_4 \exp\left(-\frac{c_1 n \ell^2}{L^{2(1+\alpha)}}\right).$$

Solving for δ , we have $t \leq c_5 L^{1+\alpha} \sqrt{\frac{\ln(1/\delta)}{n}}$ with probability at least $1 - \delta$. This alongside the condition (64) lead to the desired result. \blacksquare

Let us turn to the proof of Theorem 16. The proof is similar to the proof of Theorem 10.2 by van de Geer (2000), but with necessary modifications in order to get a high probability upper bound that holds uniformly over Q . We discuss the differences in more detail after the proof.

Proof [Proof of Theorem 16] Recall that in the optimization problem, we use $w_i = (X_i, A_i, R_i, X_i')$ ($i = 1, \dots, n$) to denote the i th elements of the data set $\mathcal{D}_n = \{(X_i, A_i, R_i, X_i')\}_{i=1}^n$. Also for a measurable function $f : \mathcal{X} \times \mathcal{A} \times \mathbb{R} \times \mathbb{R} \times \mathcal{X} \rightarrow \mathbb{R}$, we denote $\|f\|_n^2 = \frac{1}{n} \sum_{i=1}^n |f(w_i)|^2$. We also let $(X, A) \sim \nu$, $R \sim \mathcal{R}(\cdot | X, A)$, and $X' \sim P(\cdot | X, A)$ be random variables that are independent of \mathcal{D}_n .

For any $Q \in \mathcal{F}^{|\mathcal{A}|}$ and the corresponding $T^\pi Q \in \mathcal{F}^{|\mathcal{A}|}$, define the mapping $\bar{W}(Q, T^\pi Q, \mathbf{1}) : \mathcal{X} \times \mathcal{A} \times \mathbb{R} \times \mathcal{X} \rightarrow \mathbb{R}$ by $\bar{W}(Q, T^\pi Q, \mathbf{1})(X, A, R, X') = R + \gamma Q(X', \pi(X)) - T^\pi Q(X, A)$, in which $\mathbf{1} \in \mathcal{F}^{|\mathcal{A}|}$ is the constant function defined on $\mathcal{X} \times \mathcal{A}$ with the value of one. For any fixed Q and $t = 1, \dots, n$, define the random variables $W_t(Q) = \bar{W}(Q, T^\pi Q, \mathbf{1})(X_t, A_t, R_t, X_t')$ and let $W(Q)$ denote the random vector $[W_1(Q) \dots W_n(Q)]^\top$. Notice that $|W_t(Q)| \leq 3Q_{\max}$ and we have $\mathbb{E}[W_t(Q) | Q] = 0$ ($t = 1, \dots, n$).

From the optimizer property of $\hat{h}_n(\cdot, Q)$, we have

$$\begin{aligned} \|\hat{h}_n(Q) - [R + \gamma Q(X_i', \pi(X_i'))]\|_n^2 + \lambda_{h_n} J^2(\hat{h}_n(Q)) &\leq \\ \|T^\pi Q - [R + \gamma Q(X_i', \pi(X_i'))]\|_n^2 + \lambda_{h_n} J^2(T^\pi Q). \end{aligned}$$

After expanding and rearranging, we get

$$\|\hat{h}_n(Q) - T^\pi Q\|_n^2 + \lambda_{h_n} J^2(\hat{h}_n(Q)) \leq 2 \left\langle W(Q), \hat{h}_n(Q) - T^\pi Q \right\rangle_n + \lambda_{h_n} J^2(T^\pi Q). \quad (65)$$

We evoke Lemma 24 to upper bound $\left| \left\langle W(Q), \hat{h}_n(Q) - T^\pi Q \right\rangle_n \right|$. The function spaces \mathcal{G} and \mathcal{G}' in that lemma are set as $\mathcal{G} : \mathcal{X} \times \mathcal{A} \rightarrow \mathbb{R}$ and $\mathcal{G}' : \mathcal{X} \times \mathcal{A} \times \mathbb{R} \times \mathcal{X} \rightarrow \mathbb{R}^3$ with

$$\begin{aligned} \mathcal{G} &= \left\{ h - T^\pi Q : h, Q \in \mathcal{F}^{|\mathcal{A}|} \right\}, \\ \mathcal{G}' &= \left\{ (Q, T^\pi Q, \mathbf{1}) : Q, T^\pi Q \in \mathcal{F}^{|\mathcal{A}|} \right\}. \end{aligned}$$

All functions in $\mathcal{F}^{|\mathcal{A}|}$ are Q_{\max} -bounded, so the functions in \mathcal{G} and \mathcal{G}' are bounded by $2Q_{\max}$ and $(Q_{\max}, Q_{\max}, 1)$, respectively. Moreover for any $g \in \mathcal{G}$, $\frac{1}{n} \sum_{i=1}^n |g(X_i, A_i)|^2 \leq 4Q_{\max}^2$. So by setting L equal to $2Q_{\max}$ in that lemma, all boundedness conditions are satisfied.

Define $J(g, g') = J(h) + J(Q) + J(T^\pi Q)$ and denote

$$(\mathcal{G} \times \mathcal{G}')_B = \{(g, g') \in \mathcal{G} \times \mathcal{G}' : J(g, g') \leq B\}.$$

Lemma 24 requires an upper bound on $\log \mathcal{N}_\infty(u, (\mathcal{G} \times \mathcal{G}')_B)$. We relate the metric entropy of this space to that of $\mathcal{F}_B = \{f \in \mathcal{F} : J(f) \leq B\}$, which is specified by Assumption A4. Notice that if $J(g, g') \leq B$, each of $J(h)$, $J(Q)$, and $J(T^\pi Q)$ is also less than or equal to B . So we have

$$\begin{aligned} (\mathcal{G} \times \mathcal{G}')_B &= \{(h - T^\pi Q, Q, T^\pi Q, \mathbf{1}) : h, Q, T^\pi Q \in \mathcal{F}^{|\mathcal{A}|}, J(h) + J(Q) + J(T^\pi Q) \leq B\} \subset \\ &\left\{ h - T^\pi Q : h, T^\pi Q \in \mathcal{F}^{|\mathcal{A}|}, J(h) + J(T^\pi Q) \leq B \right\} \times \left\{ Q : Q \in \mathcal{F}^{|\mathcal{A}|}, J(Q) \leq B \right\} \times \\ &\left\{ T^\pi Q : T^\pi Q \in \mathcal{F}^{|\mathcal{A}|}, J(T^\pi Q) \leq B \right\} \times \{\mathbf{1}\}. \end{aligned}$$

Because $J(\cdot)$ is a pseudo-norm, we have $J(h - T^\pi Q) \leq J(h) + J(T^\pi Q)$, so

$$\left\{ h - T^\pi Q : h, T^\pi Q \in \mathcal{F}^{|\mathcal{A}|}, J(h) + J(T^\pi Q) \leq B \right\} \subset \left\{ Q : Q \in \mathcal{F}^{|\mathcal{A}|}, J(Q) \leq B \right\}.$$

As a result $(\mathcal{G} \times \mathcal{G}')_B$ is a subset of the product space $\{Q \in \mathcal{F}^{|\mathcal{A}|} : J(Q) \leq B\}^3$. Therefore by the usual covering argument, we get that

$$\log \mathcal{N}_\infty(u, (\mathcal{G} \times \mathcal{G}')_B) \leq 3 \log \mathcal{N}_\infty\left(u, \left\{ Q \in \mathcal{F}^{|\mathcal{A}|} : J(Q) \leq B \right\}\right).$$

It is easy to see that for finite $|\mathcal{A}|$, if $\log \mathcal{N}_\infty(u, \{f \in \mathcal{F} : J(f) \leq B\}) \leq C \left(\frac{B}{u}\right)^{2\alpha}$, then $\log \mathcal{N}_\infty(u, \{f_1, \dots, f_{|\mathcal{A}|}\} \in \mathcal{F}^{|\mathcal{A}|} : J((f_1, \dots, f_{|\mathcal{A}|})) \leq B\} \leq C_1 \left(\frac{B}{u}\right)^{2\alpha}$ (we benefit from the condition $J(Q(\cdot, a)) \leq J(Q)$ in Assumption A3; the proof is similar to the proof of Lemma 20 in Appendix E). Here the constant C_1 depends on $|\mathcal{A}|$. This along with the previous inequality show that for some constant $A > 0$, we have

$$\log \mathcal{N}_\infty(u, (\mathcal{G} \times \mathcal{G}')_B) \leq A \left(\frac{B}{u}\right)^{2\alpha}.$$

We are ready to apply Lemma 24 to upper bound the inner product term in (65). Fix $\delta > 0$. To simplify the notation, denote $L_n = \|\hat{h}_n(Q) - T^\pi Q\|_n$, set $t_0 = \sqrt{\frac{\ln(1/\delta)}{n}}$, and use \hat{h}_n to refer to $\hat{h}_n(Q)$. There exists a constant $c > 0$ such that with probability at least $1 - \delta$, we have

$$L_n^2 + \lambda_{h_n} J^2(\hat{h}_n) \leq 2cL^{1+\alpha} L_n^{1-\alpha} \left(J(\hat{h}_n) + J(Q) + J(T^\pi Q) \right)^\alpha t_0 + \lambda_{h_n} J^2(T^\pi Q). \quad (66)$$

Either the first term in the RHS is larger than the second one or the second term is larger than the first. We analyze each case separately.

Case 1. $2cL^{1+\alpha} L_n^{1-\alpha} (J(\hat{h}_n) + J(Q) + J(T^\pi Q))^\alpha t_0 \geq \lambda_{h_n} J^2(T^\pi Q)$. In this case we have

$$L_n^2 + \lambda_{h_n} J^2(\hat{h}_n) \leq 4cL^{1+\alpha} L_n^{1-\alpha} \left(J(\hat{h}_n) + J(Q) + J(T^\pi Q) \right)^\alpha t_0. \quad (67)$$

Again, two cases might happen:

Case 1.a. $J(\hat{h}_n) > J(Q) + J(T^\pi Q)$: From (67) we have $L_n^2 \leq 2^{2+\alpha} c L^{1+\alpha} L_n^{1-\alpha} J^\alpha(\hat{h}_n)/t_0$. Solving for L_n , we get that $L_n \leq \frac{2^{2+\alpha} c^{1+\alpha} L^{1+\alpha} J(\hat{h}_n)}{\lambda_{h,n}^{1+\alpha} t_0^{1+\alpha}}$. From (67) we also have $\lambda_{h,n} J^2(\hat{h}_n) \leq 2^{2+\alpha} c L^{1+\alpha} L_n^{1-\alpha} J^\alpha(\hat{h}_n)/t_0$. Plugging-in the recently obtained upper bound on L_n and solving for $J(\hat{h}_n)$, we get that

$$J(\hat{h}_n) \leq \frac{2^{2+\alpha} c L^{1+\alpha} t_0}{\lambda_{h,n}^{\frac{3}{2}}}. \quad (68)$$

Substituting this in the upper bound on L_n , we get that

$$L_n \leq \frac{2^{2+\alpha} c L^{1+\alpha} t_0}{\lambda_{h,n}^{\frac{3}{2}}}. \quad (69)$$

Case 1.b. $J(\hat{h}_n) \leq J(Q) + J(T^\pi Q)$: The upper bound on $J(\hat{h}_n)$ is obvious. From (67) we have $L_n^2 \leq 2^{2+\alpha} c L^{1+\alpha} L_n^{1-\alpha} (J(Q) + J(T^\pi Q))^\alpha t_0$. Solving for L_n , we obtain

$$L_n \leq 2^{\frac{2+\alpha}{1+\alpha}} c^{\frac{1}{1+\alpha}} L (J(Q) + J(T^\pi Q))^{\frac{\alpha}{1+\alpha}} t_0^{\frac{1}{1+\alpha}}. \quad (70)$$

Case 2. $2cL^{1+\alpha} L_n^{1-\alpha} J(\hat{h}_n) + J(Q) + J(T^\pi Q)^\alpha t_0 < \lambda_{h,n} J^2(T^\pi Q)$. In this case we have $L_n^2 + \lambda_{h,n} J^2(\hat{h}_n) \leq 2\lambda_{h,n} J^2(T^\pi Q)$, which implies that

$$L_n \leq \sqrt{2\lambda_{h,n}} J(T^\pi Q), \quad (71)$$

$$J(\hat{h}_n) \leq \sqrt{2} J(T^\pi Q). \quad (72)$$

By (69), (70), and (71) for L_n and (68), (72), and the condition $J(\hat{h}_n) \leq J(Q) + J(T^\pi Q)$ in Case 1.b. for $J(\hat{h}_n)$, we have that for any fixed $0 < \delta < 1$, with probability at least $1 - \delta$, the following inequalities hold:

$$\begin{aligned} \|\hat{h}_n(Q) - T^\pi Q\|_n &\leq \max \left\{ \frac{2^{2+\alpha} c L^{1+\alpha} \sqrt{\frac{\ln(1/\delta)}{n}}}{\lambda_{h,n}^{\frac{3}{2}}}, \right. \\ &\quad \left. \frac{2^{2+\alpha} c^{1+\alpha} L (J(Q) + J(T^\pi Q))^{\frac{\alpha}{1+\alpha}} \left(\frac{\ln(1/\delta)}{n} \right)^{\frac{1}{2(1+\alpha)}}}{\lambda_{h,n}^{\frac{3}{2}}}, \right. \\ &\quad \left. \sqrt{2\lambda_{h,n} J(T^\pi Q)} \right\}, \\ J(\hat{h}_n(Q)) &\leq \max \left\{ \frac{2^{2+\alpha} c L^{1+\alpha} \sqrt{\frac{\ln(1/\delta)}{n}}}{\lambda_{h,n}^{\frac{3}{2}}}, J(Q) + J(T^\pi Q), \sqrt{2} J(T^\pi Q) \right\}. \end{aligned}$$

Comparing this proof with that of Theorem 10.2 by van de Geer (2000), we see that here we do not normalize the function space $\mathcal{G} \times \mathcal{G}'$ to ensure that $J(g, g') \leq 1$ and then

use their Lemma 8.4, which provides a high-probability upper bound on $\sup_{g \in \mathcal{G}} \frac{\langle W, g \rangle_n}{\|g\|_{F_n}^{1-\alpha}}$. Instead we directly apply Lemma 24, which upper bounds $\sup_{(g, g') \in \mathcal{G} \times \mathcal{G}'} \frac{\langle W(g'), g \rangle_n}{\|g\|_{F_n}^{1-\alpha} J^\alpha(g, g')}$, on the (unnormalized) function space $\mathcal{G} \times \mathcal{G}'$. If we went through the former approach, in which the normalization is global, the first term in the RHS of (66) would be $L_n^{1-\alpha} (J(\hat{h}_n) + J(Q) + J(T^\pi Q))^{1+\alpha} t_0$ instead of $L_n^{1-\alpha} (J(\hat{h}_n) + J(Q) + J(T^\pi Q))^\alpha t_0$ of here, which is obtained by local normalization. This extra $J(\hat{h}_n) + J(Q) + J(T^\pi Q)$ would prevent us from getting proper upper bounds on L_n and $J(\hat{h}_n)$ in Case 1.a above. The reason that the original proof does not work is that here $W(g')$ is a function of $g' \in \mathcal{G}'$.

Appendix E. Proof of Lemma 20 (Covering Number of G_{σ_1, σ_2})

Here we prove Lemma 20, which relates the covering number of G_{σ_1, σ_2} to the covering number of \mathcal{F}_{σ_1} and \mathcal{F}_{σ_2} .

Proof [Proof of Lemma 20] Let $g_{Q_1, h_1}, g_{Q_2, h_2} \in G_{\sigma_1, \sigma_2}$. By the definition of G_{σ_1, σ_2} (41), the functions Q_1 and h_1 corresponding to g_{Q_1, h_1} satisfy $Q_1 \in \mathcal{F}_{\sigma_1}^{|\mathcal{A}|}$ and $h_1 \in \mathcal{F}_{\sigma_2}^{|\mathcal{A}|}$ (and similarly for Q_2 and h_2). Set $z_i = (x_i, a_i)$. We have

$$\begin{aligned} &\frac{1}{n} \sum_{i=1}^n |g_{Q_1, h_1}(z_i) - g_{Q_2, h_2}(z_i)|^2 \\ &= \frac{1}{n} \sum_{i=1}^n [(Q_1(z_i) - h_1(z_i))^2 - (Q_2(z_i) - h_2(z_i))^2]^2 \\ &\leq 16 Q_{\max}^2 \frac{1}{n} \sum_{i=1}^n [(Q_1(z_i) - Q_2(z_i)) + (h_1(z_i) - h_2(z_i))]^2 \\ &\leq 32 Q_{\max}^2 \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^{|\mathcal{A}|} [(Q_{1,j}(x_i) - Q_{2,j}(x_i))^2 + (h_{1,j}(x_i) - h_{2,j}(x_i))^2]. \end{aligned}$$

Assumption A3 implies that for $Q_1, Q_2 \in \mathcal{F}_{\sigma_1}^{|\mathcal{A}|}$, the functions $Q_{1,j}, Q_{2,j}$ are in \mathcal{F}_{σ_1} and for $h_1, h_2 \in \mathcal{F}_{\sigma_2}^{|\mathcal{A}|}$, the functions $h_{1,j}, h_{2,j}$ are in \mathcal{F}_{σ_2} —for all $j = 1, \dots, |\mathcal{A}|$. Therefore the previous inequality shows that u -covers on $Q_j \in \mathcal{F}_{\sigma_1}$ and $h_j \in \mathcal{F}_{\sigma_2}$ (for $j = 1, \dots, |\mathcal{A}|$) w.r.t. the empirical norms $\|\cdot\|_{x_{1:n}}$ define an $8Q_{\max} \sqrt{|\mathcal{A}|}$ u -cover on G_{σ_1, σ_2} w.r.t. $\|\cdot\|_{x_{1:n}}$. Thus,

$$\mathcal{N}_2 \left(8Q_{\max} \sqrt{|\mathcal{A}|} u, G_{\sigma_1, \sigma_2}, (x, a)_{1:n} \right) \leq \mathcal{N}_2(u, \mathcal{F}_{\sigma_1}, x_{1:n})^{|\mathcal{A}|} \times \mathcal{N}_2(u, \mathcal{F}_{\sigma_2}, x_{1:n})^{|\mathcal{A}|}.$$

Assumption A4 then implies that for a constant c_1 , independent of $u, |\mathcal{A}|, Q_{\max}$, and α , and for all $((x_1, a_1), \dots, (x_n, a_n)) \in \mathcal{X} \times \mathcal{A}$ we have

$$\log \mathcal{N}_2(u, G_{\sigma_1, \sigma_2}, (x, a)_{1:n}) \leq c_1 |\mathcal{A}|^{1+\alpha} Q_{\max}^{2\alpha} (\sigma_1^\alpha + \sigma_2^\alpha) u^{-2\alpha}.$$

Appendix F. Convolutional MDPs and Assumption A7

In this appendix, we show that Assumption A7 holds for a certain class of MDPs. This class is defined by one dimensional MDPs in which the increment of the next X' compared to the current state X is a function of chosen action only, i.e., $X' - X \sim W(\pi(X))$.

Proposition 25 Suppose that $\mathcal{X} = [-\pi, \pi]$ is the unit circle and \mathcal{F} is the Sobolev space $\mathcal{W}^{k,2}(\mathcal{X}) = \mathcal{W}^{k,2}(\mathcal{X})$ and $J(\cdot)$ is defined as the corresponding norm $\|\cdot\|_{\mathcal{W}^{k,2}}$. For a function $f \in \mathcal{F}$, let $\hat{f}(n)$ be the n^{th} Fourier coefficient, i.e., $\hat{f}(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-jn x} dx$. Consider the MDPs that have the convolutional transition probability kernel, that is, for any policy π and $V \in \mathcal{F}$, there exists $K_{\pi}(x, y) = K_{\pi}(x - y)$ such that

$$\int_{\mathcal{X}} P(dy|x; \pi(x)) V(y) = \int_{\mathcal{X}} K_{\pi}(x - y) V(y) dy = K_{\pi} * V.$$

Moreover, assume that $K_{\pi}, V \in L_1(\mathcal{X})$. For a given policy π , let $r^{\pi}(x) = r(x, \pi(x))$ ($x \in \mathcal{X}$). Assumption A7 is then satisfied with the choice of $L_R = \sup_{\pi} \|r^{\pi}\|_{\mathcal{W}^{k,2}}$ and $L_P = \sup_{\pi} \max_n |\hat{K}_{\pi}(n)|$.

Proof By the convolution theorem, $(\widehat{K_{\pi} * V})(n) = \hat{K}_{\pi}(n) \hat{V}(n)$. It is also known that for $V \in \mathcal{F}$, we have $\|V\|_{\mathcal{W}^{k,2}}^2 = \sum_{n=-\infty}^{\infty} (1 + |n|^2)^k |\hat{V}(n)|^2$. Thus,

$$\begin{aligned} \|K_{\pi} * V\|_{\mathcal{W}^{k,2}}^2 &= \sum_{n=-\infty}^{\infty} (1 + |n|^2)^k |\hat{K}_{\pi}(n)|^2 |\hat{V}(n)|^2 \leq \left[\max_n |\hat{K}_{\pi}(n)|^2 \right] \sum_{n=-\infty}^{\infty} (1 + |n|^2)^k |\hat{V}(n)|^2 \\ &= \left[\max_n |\hat{K}_{\pi}(n)|^2 \right] \|V\|_{\mathcal{W}^{k,2}}^2. \end{aligned}$$

Therefore, $\|r^{\pi} V\|_{\mathcal{W}^{k,2}} \leq \|r^{\pi}\|_{\mathcal{W}^{k,2}} + \gamma \left[\max_n |\hat{K}_{\pi}(n)| \right] \|V\|_{\mathcal{W}^{k,2}}$. Taking supremum over all policies finishes the proof. ■

The interpretation of $\max_n |\hat{K}_{\pi}(n)|$ is the maximum gain of the linear filter K_{π} that is applied to a value function V . The gain here is explicitly written in the frequency domain.

Appendix G. The Metric Entropy and the Covering Number

Definition 26 (Definition 9.3 of Györfi et al. 2002) Let $\varepsilon > 0$, \mathcal{F} be a set of real-valued functions defined on \mathcal{X} , and ν_X be a probability measure on \mathcal{X} . Every finite collection of $N_{\varepsilon} = \{f_1, \dots, f_{N_{\varepsilon}}\}$ defined on \mathcal{X} with the property that for every $f \in \mathcal{F}$, there is a function $f' \in N_{\varepsilon}$ such that $\|f - f'\|_{p, \nu_X} < \varepsilon$ is called an ε -cover of \mathcal{F} w.r.t. $\|\cdot\|_{p, \nu_X}$. Let $\mathcal{N}(\varepsilon, \mathcal{F}, \|\cdot\|_{p, \nu_X})$ be the size of the smallest ε -cover of \mathcal{F} w.r.t. $\|\cdot\|_{p, \nu_X}$. If no finite ε -cover exists, take $\mathcal{N}(\varepsilon, \mathcal{F}, \|\cdot\|_{p, \nu_X}) = \infty$. Then $\mathcal{N}(\varepsilon, \mathcal{F}, \|\cdot\|_{p, \nu_X})$ is called an ε -covering number of \mathcal{F} and $\log \mathcal{N}(\varepsilon, \mathcal{F}, \|\cdot\|_{p, \nu_X})$ is called the metric entropy of \mathcal{F} w.r.t. the same norm.

The ε -covering of \mathcal{F} w.r.t. the supremum norm $\|\cdot\|_{\infty}$ is denoted by $\mathcal{N}_{\infty}(\varepsilon, \mathcal{F})$. For $x_{1:n} = (x_1, \dots, x_n) \in \mathcal{X}^{1:n}$, one may also define the empirical measure $\nu_{X_{1:n}}(A) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{\{x_i \in A\}}$ for $A \subset \mathcal{X}$. This leads to the empirical covering number of \mathcal{F} w.r.t. the empirical norm $\|\cdot\|_{p, x_{1:n}}$ and is denoted by $\mathcal{N}_{p,n}(\varepsilon, \mathcal{F}, x_{1:n})$. If $X_{1:n} = (X_1, \dots, X_n)$ is a sequence of random variables, the covering number $\mathcal{N}_{p,n}(\varepsilon, \mathcal{F}, X_{1:n})$ is a random variable too.

References

- András Antos, Rémi Munos, and Csaba Szepesvári. Fitted Q-iteration in continuous action-space MDPs. In *Advances in Neural Information Processing Systems (NIPS - 20)*, pages 9–16. MIT Press, 2008a. 33
- András Antos, Csaba Szepesvári, and Rémi Munos. Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71:89–129, 2008b. 6, 8, 10, 13, 19, 25, 26, 28, 29
- Bernardo Ávila Pres and Csaba Szepesvári. Statistical linear estimation with penalized estimators: an application to reinforcement learning. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012. 2, 13, 15, 28, 29, 31
- Philip Bachman, Amir-massoud Farahmand, and Doña Precup. Sample-based approximate regularization. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, volume 32 of *JMLR: W @ CP*, 2014. 12
- Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Machine Learning (ICML)*, pages 30–37. Morgan Kaufmann, 1995. 8
- André MS. Barreto, Doña Precup, and Joelle Pineau. Reinforcement learning using kernel-based stochastic factorization. In *Advances in Neural Information Processing Systems (NIPS - 24)*, pages 720–728, 2011. 2
- André MS. Barreto, Doña Precup, and Joelle Pineau. On-line reinforcement learning using incremental kernel-based stochastic factorization. In *Advances in Neural Information Processing Systems (NIPS - 25)*, pages 1484–1492, 2012. 2
- Rick Beatson and Leslie Greengard. A short course on fast multipole methods. In *Wavelets, Multilevel Methods and Elliptic PDEs*, pages 1–37. Oxford University Press, 1997. 32
- Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research (JMLR)*, 7:2399–2434, 2006. 12
- Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems (NIPS - 19)*, pages 137–144. MIT Press, 2006. 8
- Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulcsza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175, 2010. 8
- Dimitri P. Bertsekas and Steven E. Shreve. *Stochastic Optimal Control: The Discrete-Time Case*. Academic Press, 1978. 3, 5
- Dimitri P. Bertsekas and John N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1996. 3

- Wendelin Böhmer, Steffen Grünewälder, Yun Shen, Marek Musial, and Klaus Obermayer. Construction of approximation spaces for reinforcement learning. *Journal of Machine Learning Research (JMLR)*, 14:2067–2118, 2013. [2](#)
- Leon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems (NIPS - 20)*, pages 161–168. MIT Press, 2008. [32](#)
- Steven J. Bradtko and Andrew G. Barto. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22:33–57, 1996. [9](#)
- Corinna Cortes, Mehryar Mohri, and Andres Muñoz Medina. Adaptation algorithm and theory based on generalized discrepancy. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 169–178, 2015. [8](#)
- Christoph Dann, Gerhard Neumann, and Jan Peters. Policy evaluation with temporal differences: A survey and comparison. *Journal of Machine Learning Research (JMLR)*, 15:809–883, 2014. [31](#)
- Ronald A. Devore. Nonlinear approximation. *Acta Numerica*, 7:51–150, 1998. [12](#)
- Paul Doukhan. *Mixing: Properties and Examples*, volume 85 of *Lecture Notes in Statistics*. Springer-Verlag, Berlin, 1994. [6, 19](#)
- Bradley Efron, Trevor Hastie, Iain M. Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407 – 499, 2004. [31](#)
- Yaakov Engel, Shie Mannor, and Ron Meir. Reinforcement learning with Gaussian processes. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pages 201–208. ACM, 2005. [2, 32](#)
- Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research (JMLR)*, 6:503–556, 2005. [2](#)
- Theodoros Evgeniou, Massimiliano Pontil, and Tomaso Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13(1):1–50, 1999. [12](#)
- Amir-massoud Farahmand. Action-gap phenomenon in reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS - 24)*, pages 172–180. Curran Associates, Inc., 2011a. [28](#)
- Amir-massoud Farahmand. *Regularization in Reinforcement Learning*. PhD thesis, University of Alberta, 2011b. [2, 3, 25, 31](#)
- Amir-massoud Farahmand and Doina Precup. Value pursuit iteration. In *Advances in Neural Information Processing Systems (NIPS - 25)*, pages 1349–1357. Curran Associates, Inc., 2012. [2, 3](#)
- Amir-massoud Farahmand and Csaba Szepesvári. Model selection in reinforcement learning. *Machine Learning*, 85(3):299–332, 2011. [19, 32](#)
- Amir-massoud Farahmand and Csaba Szepesvári. Regularized least-squares regression: Learning from a β -mixing sequence. *Journal of Statistical Planning and Inference*, 142(2):493 – 505, 2012. [6, 19, 24](#)
- Amir-massoud Farahmand, Mohammad Ghavamzadeh, Csaba Szepesvári, and Shie Mannor. Regularized fitted Q-iteration: Application to planning. In *Recent Advances in Reinforcement Learning, 8th European Workshop (EWRLL)*, volume 5323 of *Lecture Notes in Computer Science*, pages 55–68. Springer, 2008. [31](#)
- Amir-massoud Farahmand, Mohammad Ghavamzadeh, Csaba Szepesvári, and Shie Mannor. Regularized fitted Q-iteration for planning in continuous-space Markovian Decision Problems. In *Proceedings of American Control Conference (ACC)*, pages 725–730, June 2009a. [2, 31](#)
- Amir-massoud Farahmand, Mohammad Ghavamzadeh, Csaba Szepesvári, and Shie Mannor. Regularized policy iteration. In *Advances in Neural Information Processing Systems (NIPS - 21)*, pages 441–448. MIT Press, 2009b. [1, 2, 3, 13, 30, 31](#)
- Amir-massoud Farahmand, Rémi Munos, and Csaba Szepesvári. Error propagation for approximate policy and value iteration. In *Advances in Neural Information Processing Systems (NIPS - 23)*, pages 568–576. 2010. [8, 25, 26, 28, 29](#)
- Amir-massoud Farahmand, Doina Precup, Mohammad Ghavamzadeh, and André M.S. Barreto. Classification-based approximate policy iteration. *IEEE Transactions on Automatic Control*, 60(11):2989–2993, November 2015. [33](#)
- Matthieu Geist and Bruno Scherrer. ℓ_1 -penalized projected Bellman residual. In *Recent Advances in Reinforcement Learning*, volume 7188 of *Lecture Notes in Computer Science*, pages 89–101. Springer Berlin Heidelberg, 2012. [2, 13, 31](#)
- Alborz Geramifard, Finale Doshi, Joshua Redding, Nicholas Roy, and Jonathan How. Online discovery of feature dependencies. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 881–888. ACM, 2011. [2](#)
- Alborz Geramifard, Thomas J. Walsh, Nicholas Roy, and Jonathan P. How. Batch iFDD: A scalable matching pursuit algorithm for solving MDPs. In *29th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, 2013. [3](#)
- Mohammad Ghavamzadeh, Alessandro Lazaric, Rémi Munos, and Matthew Hoffman. Finite-sample analysis of Lasso-TD. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 1177–1184. ACM, 2011. [2, 13, 28, 30, 31](#)
- Grigori K. Golubev and Michael Nussbaum. A risk bound in Sobolev class regression. *The Annals of Statistics*, 18(2):758–778, 1990. [24](#)
- László Györfi, Michael Kohler, Adam Krzyżak, and Harro Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer Verlag, New York, 2002. [2, 12, 21, 23, 30, 33, 35, 45, 46, 59](#)

- Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1), 1970. 15
- Matthew W. Hoffman, Alessandro Lazaric, Mohammad Ghavamzadeh, and Rémi Munos. Regularized least squares temporal difference learning with nested ℓ_1 and ℓ_2 penalization. In *Recent Advances in Reinforcement Learning*, volume 7188 of *Lecture Notes in Computer Science*, pages 102–114. Springer Berlin Heidelberg, 2012. 2, 31
- Jeff Johns. *Basis Construction and Utilization for Markov Decision Processes using Graphs*. PhD thesis, University of Massachusetts Amherst, 2010. 3
- Jeff Johns, Christopher Painter-Wakefield, and Ronald Parr. Linear complementarity for regularized policy evaluation and improvement. In *Advances in Neural Information Processing Systems (NIPS - 23)*, pages 1009–1017, 2010. 2, 31
- Tobias Jung and Daniel Polani. Least squares SVM for least squares TD learning. In *Proceedings of the 17th European Conference on Artificial Intelligence (ECAI)*, pages 499–503, 2006. 2, 31, 32
- Michael Kohler. Inequalities for uniform deviations of averages from expectations with applications to nonparametric regression. *Journal of Statistical Planning and Inference*, 89:1–23, 2000. 45
- J. Zico Kolter and Andrew Y. Ng. Regularization and feature selection in least-squares temporal difference learning. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pages 521–528. ACM, 2009. 2, 13, 31
- George Konidaris, Sarah Osentoski, and Philipp Thomas. Value function approximation in reinforcement learning using the Fourier basis. In *AAAI Conference on Artificial Intelligence*, 2011. 16
- Michail G. Lagoudakis and Ronald Parr. Least-squares policy iteration. *Journal of Machine Learning Research (JMLR)*, 4:1107–1149, 2003. 8, 9
- Alessandro Lazaric, Mohammad Ghavamzadeh, and Rémi Munos. Finite-sample analysis of least-squares policy iteration. *Journal of Machine Learning Research (JMLR)*, 13:3041–3074, October 2012. 28, 29
- Bo Liu, Sridhar Mahadevan, and Ji Liu. Regularized off-policy TD-learning. In *Advances in Neural Information Processing Systems (NIPS - 25)*, pages 845–853, 2012. 32
- Mannel Loth, Mannel Davy, and Philippe Preux. Sparse temporal difference learning using LASSO. In *IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning*, pages 352–359, 2007. 2, 31
- Sridhar Mahadevan and Bo Liu. Basis construction from power series expansions of value functions. In *Advances in Neural Information Processing Systems (NIPS - 23)*, pages 1540–1548, 2010. 2
- Sridhar Mahadevan and Mauro Maggioni. Proto-value functions: A Laplacian framework for learning representation and control in Markov decision processes. *Journal of Machine Learning Research (JMLR)*, 8:2169–2231, 2007. 2
- Stéphane Mallat and Zhiqiang Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993. 3
- Yislay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *Proceedings of the 29th Annual Conference on Learning Theory (COLT)*, 2009. 8
- Mahdi Mifani Fard, Yuri Grinberg, Amir-massoud Farahmand, Joelle Pineau, and Doina Precup. Bellman error based feature generation using random projections on sparse spaces. In *Advances in Neural Information Processing Systems (NIPS - 20)*, pages 3030–3038, 2013. 2
- Rémi Munos. Error bounds for approximate policy iteration. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pages 560–567, 2003. 8
- Rémi Munos. Performance bounds in L_p norm for approximate value iteration. *SIAM Journal on Control and Optimization*, pages 541–561, 2007. 25, 26
- Boaz Nadler, Nathan Srebro, and Xueyan Zhou. Semi-supervised learning with the graph Laplacian: The limit of infinite unlabelled data. In *Advances in Neural Information Processing Systems (NIPS - 22)*, pages 1330–1338, 2009. 14
- Michael Nussbaum. Spline smoothing in regression models and asymptotic efficiency in L_2 . *Annals of Statistics*, 13(3):984–997, 1985. 24
- Michael Nussbaum. Minimax risk: Pinsker bound. In *Encyclopedia of Statistical Sciences*, volume 3, pages 451–460, 1999. 23, 24
- Dirk Ormoneit and Saunak Sen. Kernel-based reinforcement learning. *Machine Learning*, 49:161–178, 2002. 2
- Christopher Painter-Wakefield and Ronald Parr. Greedy algorithms for sparse reinforcement learning. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 2012. 3
- Ronald Parr, Christopher Painter-Wakefield, Lihong Li, and Michael Littman. Analyzing feature generation for value-function approximation. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, pages 737 – 744. ACM, 2007. 2
- Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of the 27th Annual Asilomar Conference on Signals, Systems, and Computers*, pages 40–44, 1993. 3
- Marek Petrik. An analysis of Laplacian methods for value function approximation in MDPs. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2574–2579, 2007. 2

- Zhiwei Qin, Weichang Li, and Firdaus Janoos. Sparse reinforcement learning via convex optimization. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, volume 32 of *JMLR: W & CP*, 2014. 32
- Stéphane Ross, Geoffrey Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 627–635, 2011. 19
- Paul-Marie Samson. Concentration of measure inequalities for Markov chains and ϕ -mixing processes. *The Annals of Probability*, 28(1):416–461, 2000. 19
- Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. A generalized representer theorem. In *COLT '01/EuroCOLT '01: Proceedings of the 14th Annual Conference on Computational Learning Theory and 5th European Conference on Computational Learning Theory*, pages 416–426. Springer-Verlag, 2001. 17, 34
- Paul J. Schweitzer and Abraham Seidmann. Generalized polynomial approximations in Markovian decision processes. *Journal of Mathematical Analysis and Applications*, 110:568–582, 1985. 8
- Shai Shalev-Shwartz and Nathan Srebro. SVM optimization: Inverse dependence on training set size. In *Proceedings of the 25th international conference on Machine learning (ICML)*, pages 928–935. ACM, 2008. 32
- Steve Smale and Ding-Xuan Zhou. Estimating the approximation error in learning theory. *Analysis and Applications*, 1(1):17–41, 2003. 21
- Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer, 2008. 13, 20, 21, 23, 30
- Ingo Steinwart, Don Hush, and Clint Scovel. Optimal rates for regularized least squares regression. In *in Proceedings of the 22nd Annual Conference on Learning Theory (COLT)*, 2009. 24
- Charles J. Stone. Optimal global rates of convergence for nonparametric regression. *Annals of Statistics*, 10(4):1040–1053, 1982. 23
- Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 1998. 3
- Richard S. Sutton, Hamid Reza Maei, Doina Precup, Shalabh Bhatnagar, David Silver, Csaba Szepesvári, and Eric Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pages 993–1000. ACM, 2009. 13
- Csaba Szepesvári. *Algorithms for Reinforcement Learning*. Morgan Claypool Publishers, 2010. 3
- Gavin Taylor and Ronald Parr. Kernelized value function approximation for reinforcement learning. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pages 1017–1024. ACM, 2009. 2, 31
- Hans Triebel. *Theory of Function Spaces III*. Springer, 2006. 12
- Alexander B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009. 23, 24
- Sara A. van de Geer. *Empirical Processes in M-Estimation*. Cambridge University Press, 2000. 12, 19, 20, 23, 24, 30, 33, 36, 39, 48, 49, 51, 53, 55, 57
- Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. In Y. Eldar and G. Kutyniok, editors, *Compressed Sensing, Theory and Applications*, chapter 5, pages 210–268. 2012. 19
- Larry Wasserman. *All of Nonparametric Statistics*. Springer, 2007. 2
- Ronald J. Williams and Leemon C. Baird. Tight performance bounds on greedy policies based on imperfect value functions. In *Proceedings of the Tenth Yale Workshop on Adaptive and Learning Systems*, 1994. 8
- Xin Xu, Dewen Hu, and Xicheng Lu. Kernel-based least squares policy iteration for reinforcement learning. *IEEE Transactions on Neural Networks*, 18:973–992, 2007. 31, 32
- Changjiang Yang, Ramani Duraiswami, and Larry Davis. Efficient kernel machines using the improved fast Gauss transform. In *Advances in Neural Information Processing Systems (NIPS - 17)*, pages 1561–1568. MIT Press, 2004. 32
- Yuhong Yang and Andrew R. Barron. Information-theoretic determination of minimax rates of convergence. *The Annals of Statistics*, 27(5):1564–1599, 1999. 23
- Bin Yu. Rates of convergence for empirical processes of stationary mixing sequences. *The Annals of Probability*, 22(1):94–116, January 1994. 6, 19
- Tong Zhang. Covering number bounds of certain regularized linear function classes. *Journal of Machine Learning Research (JMLR)*, 2(527 – 550), 2002. 30
- Ding-Xuan Zhou. The covering number in learning theory. *Journal of Complexity*, 18(3):739–767, 2002. 20
- Ding-Xuan Zhou. Capacity of reproducing kernel spaces in learning theory. *IEEE Transactions on Information Theory*, 49:1743–1752, 2003. 20

Multiscale Adaptive Representation of Signals: I. The Basic Framework

Cheng Tai

*PACM, Princeton University
Princeton, NJ 08544, USA*

CHENGT@MATH.PRINCETON.EDU

Weinan E

*School of Mathematical Sciences and BICMR
Peking University and*

Department of Mathematics and PACM

*Princeton University
Princeton, NJ 08544, USA*

WEINAN@MATH.PRINCETON.EDU

Editor: Tong Zhang

Abstract

We introduce a framework for designing multi-scale, adaptive, shift-invariant frames and bi-frames for representing signals. The new framework, called AdaFrame, improves over dictionary learning-based techniques in terms of computational efficiency at inference time. It improves classical multi-scale basis such as wavelet frames in terms of coding efficiency. It provides an attractive alternative to dictionary learning-based techniques for low level signal processing tasks, such as compression and denoising, as well as high level tasks, such as feature extraction for object recognition. Connections with deep convolutional networks are also discussed. In particular, the proposed framework reveals a drawback in the commonly used approach for visualizing the activations of the intermediate layers in convolutional networks, and suggests a natural alternative.

Keywords: AdaFrame, Dictionary Learning, Wavelet Frames/Bi-frames

1. Introduction

It is now well acknowledged that sparse and overcomplete representations of data play a key role in many signal processing applications. The ability to represent a signal as a sparse linear combination of a few atoms from a possibly overcomplete dictionary lies at the heart of many applications including image/audio compression, denoising, as well as higher level tasks such as object recognition.

One popular technique for representing signals is the use of dictionaries. Since the seminal work of Olshausen et al. (1996), the field of dictionary learning has seen many promising advances. The objective is to learn a dictionary such that the input

data can be written as a sparse linear combination of the dictionary atoms. More specifically, given the data represented as a matrix X , one finds the dictionary matrix D and coefficient matrix C simultaneously by solving:

$$\min_{D,C} \|X - DC\|_2^2 + \lambda \|C\|_1. \quad (1)$$

The solution is usually obtained by solving alternatively the minimization problem for D and the sparse coding problem for C with the other variable being kept fixed. After obtaining D , inference can be made by solving a sparse coding problem. Different dictionary learning models differ in the way the dictionary D is updated. Examples include: MOD (Engan et al., 1999a), K-SVD (Aharon et al., 2006) and their variants.

Dictionary learning techniques have been successfully applied to some low level image and video processing tasks, such as image/video denoising (Elad and Aharon, 2006), compression (Bryt and Elad, 2008a; Engan et al., 1999b), inpainting (Mairal et al., 2008) and other restoration tasks (Mairal et al., 2007) with state-of-the-art performances. In addition, dictionary learning and sparse coding techniques have been very popular in high level object recognition tasks where their function is to extract features from raw data. These techniques have been used successfully to extract visual features in Ranzato et al. (2007); Lee et al. (2009); Jarrett et al. (2009). At the other end are the more traditional methodologies of designing analytic tight frames, such as Fourier basis, wavelet frames and bi-frames (Daubechies et al., 2003), curvelets (Candes and Donoho, 2000), contourlets (Do and Vetterli, 2002), etc. These analytic tight frames are robust, easy to use and computationally efficient.

In some sense the analytic tight frames can also be viewed as a dictionary. The set of signals is a particular space of functions. A dictionary is found that gives rise to the optimal representation and approximation of the signals in that function class. The resulted dictionary is highly structured, and in particular, when used into applications, the dictionary atoms are never explicitly used. However, the two approaches do differ fundamentally in several aspects (see Table 1).

- **Computational cost.** For dictionary learning, the computational cost consists of two parts: the one time cost of learning the dictionary atoms and the repeated cost of solving the sparse coding problem for the test signal at inference time. Among the two, it is the latter that prevents it from being used in real time situations. Despite the efforts devoted to seeking more efficient sparse coding algorithms (Daubechies et al., 2004; Lee et al., 2006; Beck and Teboulle, 2009), none of the available techniques is efficient enough for large scale visual feature extraction. In fact, assuming that the signal x is of length N and the trained dictionary $D \in \mathbb{R}^{m \times N}$ is stored and used explicitly, then computing Dx alone requires $O(mN)$ operations. In comparison, analytic transforms are far more efficient: fast Fourier transform takes $O(N \log N)$ operations and one level wavelet transform takes only $O(N)$ operations. This is a huge efficiency gap. In addition, the computational cost of training cannot be ignored either. The

learning procedure requires solving a non-convex optimization problem, limiting dictionary atoms to low dimensions. Partly because of this, in image processing applications, dictionary atoms are only obtained for small image patches.

- **Multi-scale features.** Dictionaries as obtained by MOD and K-SVD operate at a single small scale. Since the dictionary atoms are limited to small sizes, there is not much room for multi-scale features. Past experience with wavelets has taught us that often times it is beneficial to process signals at several scales, and operate at each scale separately.

- **Artifacts.** In low level tasks such as image compression, the dictionary learning approach operates in a patch by patch manner, which produces visually unpleasant block effects along the borders of the patches (Bryt and Elad, 2008a). Post processing is often needed to remove these artifacts (Bryt and Elad, 2008b).

Adapted to data	Dictionary Learning Yes	Wavelet Tight Frames No
Computational speed	Slow	Fast
Multi-scale	No	Yes
Robustness to perturbation	Conditionally	Yes
Performance on real data	Better	Worse

Table 1: Comparison between dictionary learning and wavelet tight frames

Given the relative features of dictionary learning and wavelet tight frames, it is natural to ask whether one can design bases that have the benefits of both and avoid the problems. In other words, can one design bases that are adapted to the data but at the same time have the multi-scale structure that is essential for the efficient algorithms for wavelet tight frames?

We propose a framework of constructing adaptive frames and bi-frames (abbreviated as AdaFrame). This framework gives multi-scale, sparse representations of the signal, with an efficiency comparable to that of the wavelets at inference time.

The proposed framework is formally similar to the first few layers of a convolutional network. As a byproduct, we show that the proposed framework gives a better way of visualizing the activations of the intermediate layers of a neural net in terms of reconstruction error.

The framework presented here is best suited for datasets such that each data point has some structure. Obvious examples include time series, images and videos. However, as in the case of wavelets, it is also possible to extend this kind of ideas to less structured data such as graphs, etc (Coffman and Mlaggoni, 2006).

In Cai et al. (2014), a variational model is proposed to learn a tight frame system that is adapted to the input image. The model in Cai et al. (2014) and our model share

a similar objective and build upon similar mathematical foundations, but our work greatly extends the model proposed in Cai et al. (2014) where a sufficient condition for perfect reconstruction is replaced by a sufficient and necessary condition and the tight frame is extended to bi-frame, which is much more flexible.

Most examples discussed in this paper are still of the low level image processing type. In a subsequent paper, we will discuss more thoroughly higher level tasks such as image classification.

The organization of this paper is as follows. In section 2, we introduce shift-invariant frames and bi-frames. In section 3, we introduce the adaptive construction of shift-invariant frames. In section 4, we introduce the adaptive construction of shift-invariant bi-frames. In section 5, we discuss multi-level constructions. In section 6, we give some simple illustrative examples of the adaptively constructed frames and bi-frames. In section 7, we discuss the connection with predefined wavelets and wavelet frames. In section 8, we discuss applications to image processing and image classification. In section 9, we discuss connection with deconvolutional nets and reconstruction of input data from features in the intermediate layers of the convolutional nets. Some conclusions are drawn in section 10.

2. Shift-invariant Frames and Bi-frames

An important starting point is the concept of multi-resolution analysis (MRA) introduced in Mallat (1989) and Meyer (1995), of which wavelets are particularly popular examples. One main advantage of MRA is that it comes naturally with fast decomposition and reconstruction algorithms, and this has been essential for making wavelets a practical tool in signal processing (Daubechies et al., 2003; Shen, 2010). Although our work builds upon the theory of wavelet frames in the continuous setting, we decide to introduce our model in a purely discrete setup. This has the advantage that it is more direct and more easily linked with existing machine learning models, including dictionary learning and convolutional networks. However, as noted in Han (2010), there is a canonical link between affine systems in the continuous setting and fast algorithms in the discrete framework.

The signals and the filters are all assumed to be discrete sequences in $l_2(\mathbb{Z}^d)$, where d is the dimension. For audio, image and video signals, $d = 1, 2, 3$ respectively. First let us define the up- and down-sampling operators. Let M be an integer. The (one dimensional) down-sampling and up-sampling operator are defined by:

$$\begin{aligned} [v \downarrow_M](n) &:= v(Mn), & n \in \mathbb{Z} \\ [v \uparrow](n) &:= \begin{cases} v(k), & n = Mk, k \in \mathbb{Z} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (2)$$

respectively, for $v \in l_2(\mathbb{Z})$. M is the decimation factor. Similarly if $d > 1$, denote the decimation factor in each dimension by M_1, M_2, \dots, M_d . For convenience we define a matrix $M = \text{Diag}(M_1, \dots, M_d) \in \mathbb{R}^{d \times d}$. A common choice of M in image processing

is $M = 2I$. We call M the *sampling matrix* and use the same notation as in (2) where Mn is understood as the matrix-vector multiplication. In general M can be an invertible matrix whose entries are positive integers or rational numbers that are greater than 1.

Key to the decomposition and reconstruction algorithms are the transition and subdivision operators. For a data sequence $v \in l_2(\mathbb{Z}^d)$, a finitely supported filter $a \in l_2(\mathbb{Z}^d)$ and a sampling matrix $M \in \mathbb{R}^{d \times d}$, the transition operator $\mathcal{T}_a : l_2(\mathbb{Z}^d) \mapsto l_2(\mathbb{Z}^d)$ is defined by

$$(\mathcal{T}_{a,M}v)(n) := \downarrow_M [v * a](n) = \sum_{k \in \mathbb{Z}^d} v(k) \overline{a(k - Mn)}, \quad (3)$$

the subdivision operator $\mathcal{S}_a : l_2(\mathbb{Z}^d) \mapsto l_2(\mathbb{Z}^d)$ is defined by

$$(\mathcal{S}_{a,M}v)(n) := |\det(M)| [a * (\uparrow v)](n) = |\det(M)| \sum_{k \in \mathbb{Z}^d} v(k) a(n - Mk). \quad (4)$$

To make the notations more concise, we omit M in the subscript.

Given a set of finitely supported filters $A = \{a_1, \dots, a_m\}$ and the coefficient sequence $v \in l_2(\mathbb{Z}^d)$, which could be the input signal itself or the coefficients computed at some decomposition level, we compute coefficients of the next level by

$$v_l = \mathcal{T}_{a_l} v, \quad l = 1, \dots, m. \quad (5)$$

With this notation, the one-level decomposition operator $W_A : l_2(\mathbb{Z}^d) \mapsto \underbrace{l_2(\mathbb{Z}^d) \oplus \dots \oplus l_2(\mathbb{Z}^d)}_{m \text{ times}}$

is defined as:

$$W_A v := \{v_1, \dots, v_l\} = \{\mathcal{T}_{a_1} v, \mathcal{T}_{a_2} v, \dots, \mathcal{T}_{a_m} v\}. \quad (6)$$

Given a set of finitely supported filters $B = \{b_1, \dots, b_m\}$, the one-level reconstruction operator $R_B : \underbrace{l_2(\mathbb{Z}^d) \oplus \dots \oplus l_2(\mathbb{Z}^d)}_{m \text{ times}} \mapsto l_2(\mathbb{Z}^d)$ is defined as

$$R_B(v_1, \dots, v_m) := \sum_{l=1}^m \mathcal{S}_{b_l} v_l \quad (7)$$

In wavelet frames, the filters A used for decomposition and the filters B used for reconstruction are connected by : $b_l(\cdot) = a_l(-\cdot)$, $l = 1, \dots, m$, where $a_l(-\cdot)$ means flip the entries of a_l along each dimension. But this does not have to be the case: A and B can be different and together they constitute a *bi-frame*.

The main requirement is that of perfect reconstruction, by which we mean:

$$R_B W_A v = v \quad \forall v \in l_2(\mathbb{Z}^d). \quad (8)$$

The following result is crucial.

Theorem 1 (Ron and Shen, 1997) Let $M \in \mathbb{R}^{d \times d}$ be a sampling matrix, let $A = \{a_1, \dots, a_m\}$ and $B = \{b_1, \dots, b_m\}$ be two sets of finitely supported sequences in $l_2(\mathbb{Z}^d)$. Then the perfect reconstruction property

$$R_B W_A v = v, \quad \forall v \in l_2(\mathbb{Z}^d) \quad (9)$$

holds if and only if, for all $k, j \in \mathbb{Z}^d$,

$$\sum_{l=1}^m \sum_{n \in \mathbb{Z}^d} a_l(Mn + j) \overline{b_l(k + Mn + j)} = |\det(M)|^{-1} \delta_k \quad (10)$$

where $\delta_k = 1$ if $k = 0$ and $\delta_k = 0$ otherwise.

In the case of wavelet tight frames, $b_l(\cdot) = a_l(-\cdot)$, $l = 1, \dots, m$, and we have:

Theorem 2 (Ron and Shen, 1997) Let $M \in \mathbb{R}^{d \times d}$ be a sampling matrix, let $A = \{a_1, \dots, a_m\}$ be a set of finitely supported sequences in $l_2(\mathbb{Z}^d)$. Then the perfect reconstruction property

$$R_A W_A v = v, \quad \forall v \in l_2(\mathbb{Z}^d). \quad (11)$$

holds if and only if, for all $k, j \in \mathbb{Z}^d$,

$$\sum_{l=1}^m \sum_{n \in \mathbb{Z}^d} a_l(Mn + j) \overline{a_l(k + Mn + j)} = |\det(M)|^{-1} \delta_k \quad (12)$$

In particular, if the data are real numbers and no down-sampling is performed, then the perfect reconstruction condition (12) becomes

$$\sum_{i=1}^m \sum_{n \in \mathbb{Z}^d} a_i(k+n) a_i(n) = \delta_k, \quad \forall k \in \mathbb{Z}^d. \quad (13)$$

The proof of Theorem 1 and Theorem 2 can be found in Daubechies et al. (2003). For completeness, we give a direct proof for the discrete case in the appendix. These conditions are referred to as the unitary extension principle (UEP) in wavelet frame theory.

As an example, the linear B-spline wavelet tight frame used in many image restoration tasks is constructed via the UEP. Its associated filters are :

$$a_1 = \frac{1}{4}(1, 2, 1)^T; \quad a_2 = \frac{\sqrt{2}}{4}(1, 0, -1)^T; \quad a_3 = \frac{1}{4}(-1, 2, -1)^T.$$

This kind of tight frames are shift-invariant systems since the transforms are in the form of discrete convolution. They are suited for the case when, below certain scale, the statistical properties of the signals are translation invariant.

3. Adaptive Construction of Frames

Given a set of signals $X = \{x_1, \dots, x_N\}$, the goal is to construct wavelet frames that are adapted to this set of signals in the sense that signals in the given set have a sparse representation.

Define \mathcal{Q} to be the set of filters that satisfy the UEP condition:

$$\mathcal{Q} = \left\{ \{a_k\}_{k=1}^m : \sum_{l=1}^m \sum_{n \in \mathbb{Z}^d} a_l(Mn + j) \overline{a_l(Mn + k + j)} = |\det(M)|^{-1} \delta_k, \forall k, j \in \mathbb{Z}^d \right\}. \quad (14)$$

Filters in this set generate a wavelet frame that provide a faithful representation for all signals in $l_2(\mathbb{Z}^d)$. However, we are not interested in all signals in $l_2(\mathbb{Z}^d)$. We are only interested in X . Among all filters in \mathcal{Q} , we want to select the one that is most adapted to X .

In image restoration tasks, we are mostly interested in wavelet frames that give rise to a sparse representation of the input signal. Therefore we will use sparsity as our guiding principle for selecting the filters. Other guiding principles such as the discriminative criterion can also be used. But in this paper, we will focus on sparsity. Let Φ be a sparsity-inducing function. Examples of Φ include the l_1 norm, l_0 "norm", or the Huber loss function defined (component-wise) by:

$$L_\delta(x) = \begin{cases} \frac{1}{2}x^2, & |x| \leq \delta \\ \delta(|x| - \frac{1}{2}\delta), & \text{otherwise} \end{cases}. \quad (15)$$

Given the data X , the adaptive filters are chosen by solving the following optimization problem:

$$\begin{aligned} & \min_{a_1, \dots, a_m} \sum_{j=1}^N \sum_{i=1}^m \Phi(v_{i,j}) \\ & \text{subject to } v_{i,j} = T_{a_i} x_j, \quad i = 1, \dots, m \\ & \{a_i\}_{i=1}^m \in \mathcal{Q} \end{aligned} \quad (16)$$

In the following, without loss of generality, we will assume that there is only one data point in the signal set, i.e. $N = 1$, and we will omit the subscript j .

To be specific, we use l_1 norm as the measurement of sparsity and we will note the changes required if the l_0 norm is used. The above problem then becomes

$$\begin{aligned} & \min_{a_1, \dots, a_m} \sum_{i=1}^m \|\mathcal{T}_{a_i} x\|_1 \\ & \{a_i\}_{i=1}^m \in \mathcal{Q} \end{aligned} \quad (17)$$

This innocent looking optimization problem is difficult to solve because of the constraint. Consider the simplest case when the signals and the filters are all one-dimensional. Assume each filter has support length r , and we have r of them. For a

real symmetric matrix G , let us denote by $Tr(G; k)$ the sum of entries along the k -th sub-diagonal. For example, $Tr(G; 0)$ is the usual trace of G . Let $A := (a_1, \dots, a_m)$. Then the constraint $\{a_i\}_{i=1}^m \in \mathcal{Q}$ is equivalent to

$$Tr(AA^T, k) = \delta_k, \quad k = 0, \dots, r-1.$$

To see a nontrivial example where this constraint is satisfied, take an orthogonal matrix $U \in \mathbb{R}^{r \times r}$, and let $a_i = \frac{1}{\sqrt{r}} U_{:,i}$, $i = 1, \dots, m$, where $U_{:,i}$ means the i -th column of U . However, in general, the algebraic constraint above is difficult to deal with. Note also that this optimization problem is not convex.

We use the split Bregman algorithm (Goldstein and Osher, 2009) to solve (17). Introduce the auxiliary variable $D = (d_1, \dots, d_m)$ where $d_i = T_{a_i} x$, $i = 1, \dots, m$. Define the norm $\|D\|_{1,1} := \sum_{i=1}^m \|d_i\|_1$. Then (17) is equivalent to:

$$\begin{aligned} & \min_{A, D} \|D\|_{1,1} \\ & \text{subject to } D = W_A x \\ & A \in \mathcal{Q} \end{aligned} \quad (18)$$

Applying the split Bregman method, we obtain the following algorithm:

Algorithm 1 Adaptive construction of frames

- 1: **Input:** x .
- 2: **Initialize** $k = 0, B = 0, A = A^0, D = W_{A^0} x$.
- 3: **while** "not converge" **do**
- 4: $D^{k+1} \leftarrow \arg \min_D \|D\|_{1,1} + \frac{\eta}{2} \|D - W_{A^k} x - B^k\|_F^2$
- 5: $A^{k+1} \leftarrow \arg \min_A \|W_{A^k} x - D^{k+1} + B^k\|_F^2$ **s.t.** $A \in \mathcal{Q}$.
- 6: $B^{k+1} \leftarrow B^k + W_{A^{k+1}} - D^{k+1}$.
- 7: $k \leftarrow k + 1$
- 8: **return** A^k

To implement the algorithm, we must be able to solve each of the subproblems listed in steps 4, 5 and 6.

To solve the subproblem for D , note that the problem decouples for each d_i , $i = 1, \dots, m$. In fact,

$$d_i^{k+1} = \arg \min_d \left(\|d\|_1 + \frac{\eta}{2} \|\mathcal{T}_{a_i} x - d + b_i^k\| \right) \quad (19)$$

for $i = 1, \dots, m$. It is easy to see that (19) has a closed form solution given by

$$d_i^{k+1} = \text{shrink}(\mathcal{T}_{a_i} x + b_i^k, \frac{1}{\eta}) \quad (20)$$

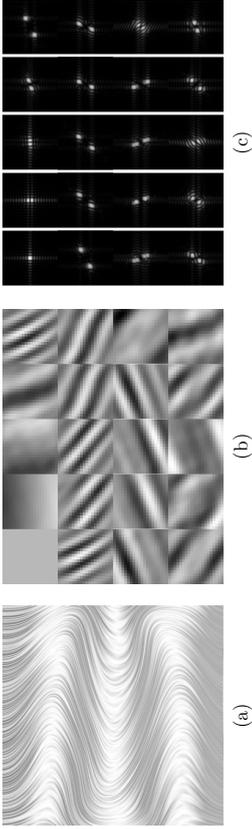


Figure 1: (a) The input image. (b) The filters learned using Algorithm 1, $m = 20$, $r = 20$. (c) The Fourier spectrum of the corresponding filters. Note the first filter is a low-pass filter, all other filters are high-pass filters as can be seen from the Fourier spectrum. The second and third filter look like edge detectors along the axis. Other filters detect oscillations along different directions.

where the function $\text{shrink} : \mathbb{R} \mapsto \mathbb{R}$ is defined as

$$\text{shrink}(x, a) = \begin{cases} (|x| - a)\text{sign}(x), & \text{if } |x| > a \\ 0, & \text{otherwise} \end{cases}. \quad (21)$$

When shrinkage-operator acts on a vector, it acts on each component of the vector according to (21).

The subproblem for updating A is most problematic due to the constraint. We use the interior-point method for this part of the algorithm. There is no guarantee of a global solution to this subproblem.

The update for B is straightforward. This is analogous to the step of “adding the noise back” in the ROF model for denoising (Osher et al., 2005).

Among the three subproblems, the update of A is the most time consuming. But as is observed by many authors, it is not necessary to solve A to full convergence, the intuitive reason being that if the error of the solution to the subproblem is smaller than $\|B^k - B^{k-1}\|$, the extra accuracy will be wasted. In fact, for updating A , we only run a few steps of the interior-point iterations and we still observe numerical convergence.

If we use the l_0 “norm” as the measurement of sparsity, the only change needed in the above algorithm is in the D step, where the soft-shrinkage operator is replaced by hard-thresholding defined as:

$$\text{Hard}(x, a) = \begin{cases} x, & \text{if } |x| > a \\ 0, & \text{otherwise} \end{cases}. \quad (22)$$

To give the readers some intuition about how the filters obtained look like, we show an example in Figure 1. More examples are given in section 5.

In some applications such as object recognition, perfect reconstruction is unnecessary. Instead, writing the input signal as a sparse linear combination of a few dictionary atoms is only a means to extract features to be used by other learning algorithms. Sparse coding has been quite popular in serving this purpose for visual object recognition tasks. In this case, it is possible to relax the constraint in (17). Instead of solving the constrained minimization problem, we can use a penalty method to solve an unconstrained problem. For example, in 1D, we can solve

$$\min_A \sum_{k=1}^m \|W_A x\|_{1,1} + \eta \sum_k (Tr(AA^T, k) - \delta_k)^2 \quad (23)$$

where η is a parameter that depends on our tolerance on the reconstruction error. This unconstrained problem is relatively easy to solve using first-order optimization methods.

The optimization problem may appear similar to reconstruction ICA (RICA) proposed in Le et al. (2011), but they are fundamentally different. There proposed model guarantees perfect reconstruction while RICA approximates the input signal. Perfect reconstruction in RICA can only be achieved in the limit where the weight of the reconstruction term goes to infinity. In addition, RICA does not have a multi-scale structure which is essential in wavelet tight frames. The goals of RICA and AdaFrame are different, in ICA, the goal is to find independent sources where as in AdaFrame, the goal is to build a wavelet tight frame that sparsely represent the signal.

4. Adaptive Construction of Bi-frames

In this section, we introduce the adaptive construction of wavelet bi-frames. Compared with the wavelet frames, the bi-frames offer two distinct advantages: The first is that the constraint for the filters becomes bi-linear making it easier to construct the filters. The second is that the added redundancy introduces more flexibility. These prove to be very important in practice.

Let \mathcal{Q} denote the set of pairs A and B , $A = (a_1, \dots, a_m)$, $B = (b_1, \dots, b_m)$, that satisfy (10):

$$\mathcal{Q} := \left\{ (A, B) : \sum_{l=1}^m \sum_{n \in \mathbb{Z}^d} a_l(Mn + j) \overline{b_l(k + Mn + j)} = |\det(M)|^{-1} \delta_k, \forall k, j \in \mathbb{Z}^d \right\} \quad (24)$$

We want to find filter pairs (A, B) with desired properties while respect the constraint $(A, B) \in \mathcal{Q}$. As before we will only consider sparsity. Given the data x and a sampling matrix M , we aim to solve :

$$\begin{aligned} & \min_{A, B} \|W_A x\|_{1,1} \\ & \text{subject to } (A, B) \in \mathcal{Q} \end{aligned} \quad (25)$$

The constraint $(A, B) \in \mathcal{Q}$ in bi-linear in A and B . Let us first count the number of equations.

We start with the simplest case where the signals and the filters are one dimensional. Let A, B be defined as before and assume that each filter $a_i, b_i, i = 1, \dots, m$ has support size r . Given the decimation factor M , define

$$\mathcal{S}(r) := \{(k, \gamma) : \exists n \in \mathbb{Z}, 1 \leq Mn + k + \gamma \leq r, 1 \leq Mn + \gamma \leq r\}, \quad (26)$$

then each $(k, \gamma) \in \mathcal{S}(r)$ constitutes an equation. This gives

$$|\mathcal{S}(r)| = (2r - M)M. \quad (27)$$

This is the total number of equations. The total number of unknowns in A and B is $2rm$. Therefore for (10) to have a solution, we expect:

$$2rm \geq (2r - M)M. \quad (28)$$

In the general case where the signals and the filters live in d dimensions, we can do a similar counting. Assume the support size of the filter $a_i, b_i, i = 1, \dots, m$ is $\mathbf{r} = (r_1, \dots, r_d)$, and assume that the sampling matrix is $M = \text{Diag}(M_1, \dots, M_d)$. Let

$$\mathcal{S}(\mathbf{r}) := \{(k, \gamma) \in \mathbb{Z}^d : \exists n \in \mathbb{Z}^d, \mathbf{1} \leq Mn + k + \gamma \leq \mathbf{r}, \mathbf{1} \leq Mn + \gamma \leq \mathbf{r}\}, \quad (29)$$

where the inequality is understood component-wise. Each $(k, \gamma) \in \mathcal{S}(\mathbf{r})$ gives rise to an equation. The total number of equations is

$$|\mathcal{S}(\mathbf{r})| = \prod_{i=1}^d (2r_i - M_i)M_i. \quad (30)$$

The number of unknowns in a and b is $2m \prod_{i=1}^d r_i$. Hence to have a solution to (10), we expect:

$$2m \prod_{i=1}^d r_i \geq \prod_{i=1}^d (2r_i - M_i)M_i. \quad (31)$$

Two cases are of special interest.

- **Redundant case.** In this case, the number of filters m is large. The number of decomposition coefficients is larger than the size of the input signal. Hence we call this the *redundant case*. For the optimization problem, we have more unknowns than equations. In particular, if $m \geq 2M - M^2/r$ in one dimension, and $m \prod_{i=1}^d r_i \geq \prod_{i=1}^d (2r_i - M_i)M_i$ in d dimensions, for most A , we expect (10) as a set of linear equations for B , to have a solution. Therefore, we can design A and B separately: We can design A first in whichever way we want as long as it is non-degenerate. We then solve (10) to get B .

- **Critically down-sampled case.** In this case, the number of filters m is small. The number of decomposition coefficients is the same as that of the input signal (depending on the boundary conditions). Hence we call this the *critically down-sampled case*. For example, in one dimension, $m = M$. In this down-sampled case, for a typical A , it is likely that (10), as a linear system for B , does not have a solution. This means that we must consider A and B simultaneously.

4.1 Redundant Case

4.1.1 DESIGN OF THE DECOMPOSITION FILTERS

As discussed above, we can design A in the first phase, and then choose B that satisfies the linear constraint (25) in the second phase.

However, the choice of A has a significant impact on the condition number of (25). Hence some constraints should be added. While there are a lot of flexibilities, we propose the following formulation:

$$\begin{aligned} \min_A \quad & \|W_A x\|_{1,1} \\ \text{subject to} \quad & A^T A = I \end{aligned} \quad (32)$$

The additional constraint $A^T A = I$ is chosen based on the consideration that the filters are most incoherent among themselves.

To solve (32) numerically, we apply the split Bregman method. But we need to handle the extra orthogonality constraint as well. To this end, we introduce the auxiliary variable $P = A$ as a means to split the orthogonality constraint. This trick has been used in other problems, see for example Lai and Osher (2014). The problem then becomes:

$$\begin{aligned} \min_{A, D, P} \quad & \|D\|_{1,1} \\ \text{subject to} \quad & D = W_A x, P = A, P^T P = I. \end{aligned} \quad (33)$$

The algorithm is then:

Algorithm 2 Adaptive construction of bi-frames: redundant case

- 1: **Input:** x .
 - 2: **Initialize** $k = 0, F = \mathbf{0}, C = \mathbf{0}, A = A^0, D = W_{A^0} x, P = A$.
 - 3: **while** “not converge” **do**
 - 4: **for** $n=1:N$ **do**
 - 5: $D^{k+1} \leftarrow \arg \min_D \|D\|_{1,1} + \frac{\lambda}{2} \|D - W_{A^k} x - F^k\|_F^2$
 - 6: $A^{k+1} \leftarrow \arg \min_A \eta \|W_A x - D^{k+1} + F^k\|_F^2 + \lambda \|A - P^k + C^k\|_F^2$
 - 7: $P^{k+1} \leftarrow \arg \min_P \|A^{k+1} - P + C^k\|_F^2$ **s.t.** $P^T P = I$
 - 8: $F^{k+1} \leftarrow F^k + W_{A^{k+1}} x - D^{k+1}$.
 - 9: $C^{k+1} \leftarrow C^k + A^{k+1} - P^{k+1}$
 - 10: $k \leftarrow k + 1$
 - 11: **return** A^k
-

To implement the algorithm, we must be able to solve each of the subproblems for D , A and P . Updating D is the same as in Algorithm 1. The subproblem for A is a quadratic program. It can be decoupled into m smaller problems, each of which involves one column of A . Writing $D = (d_1, \dots, d_m)$, $P = (p_1, \dots, p_m)$, $C = (c_1, \dots, c_m)$ and $F = (f_1, \dots, f_m)$, we can perform the optimization in a column by column fashion:

$$a_i^{k+1} = \arg \min_a \eta \|T_a x - d_i^k\|_2^2 + \lambda \|a - p_i^k + c_i^k\|_2^2, \quad i = 1, \dots, m \quad (34)$$

Each of the m smaller problems is an unconstrained quadratic program. Many optimization techniques can be used to solve this problem. Among the several choices of iterative algorithms, we use conjugate gradient (CG) method because the objective function value tends to decrease very quickly in the first few CG iterations, thus giving a good approximate solution quickly. For the same reason as in Algorithm 1, iteration to convergence is not necessary.

Next, we consider the subproblem for P . This problem is equivalent to:

$$\max_P \text{Trace}((A^{k+1} + C^k)^T P) \quad \text{subject to} \quad P^T P = I. \quad (35)$$

This is the classical orthogonal procrustes problem (Gower and Dijksterhuis, 2004) and has a closed form solution which we summarize in the following lemma. The proof can be found in linear algebra textbooks, e.g. Horn and Johnson, chapter 3.

Lemma 1 Let $Y \in \mathbb{R}^{n \times m}$, $n \geq m$ and $Y = UDV^T$ be the singular value decomposition of Y , then the constrained optimization problem

$$P^* = \arg \min_{P \in \mathbb{R}^{n \times m}} \|P - Y\|_F^2 \quad \text{subject to} \quad P^T P = I \quad (36)$$

has a closed form solution given by $P^* = UI_{n \times m} V^T$.

Substituting Y with $A^{k+1} + C^k$, we get the formula for updating P . Updating the auxiliary variable F and C is straightforward.

An illustration of such filters is shown in Figure 8(b).

4.1.2 DESIGN OF THE RECONSTRUCTION FILTERS

Once $A = (a_1, \dots, a_m)$ is obtained, we move on to second phase of designing the reconstruction filters B .

For fixed A and sampling matrix M , the constraint (10) is a linear system in B . Hence we will write it as $H(A)B = f$, where $H(A)$ denotes the coefficient matrix generated using A . To get some concrete ideas, let us look at a simple example.

Example. Consider a one dimensional situation where $m = 2$, $r = 3$. Assume $A = (a_1, a_2)$, $B = (b_1, b_2) \in \mathbb{R}^{3 \times 2}$,

$$A = \begin{pmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ a_{13} & a_{23} \end{pmatrix}, \quad B = \begin{pmatrix} b_{11} & b_{21} \\ b_{12} & b_{22} \\ b_{13} & b_{23} \end{pmatrix},$$

Assume $M = 1$, that is, no downsampling is performed. Then the linear equation $H(A)B = f$ is

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{21} & a_{22} & a_{23} \\ 0 & a_{11} & a_{12} & 0 & a_{21} & a_{23} \\ 0 & 0 & a_{11} & 0 & 0 & a_{21} \\ a_{12} & a_{13} & 0 & a_{22} & a_{23} & 0 \\ a_{13} & 0 & 0 & a_{23} & 0 & 0 \end{pmatrix} \begin{pmatrix} b_{11} \\ b_{12} \\ b_{13} \\ b_{21} \\ b_{22} \\ b_{23} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

This is a system of 5 equations with 6 unknowns. Therefore, we have one additional degree of freedom left to design B .

In general, since m is large, $H(A)B = f$ is an under-determined linear system. Moreover, since A is obtained by solving (32) with respect to the orthogonality constraint, the coefficient matrix $H(A)$ tend to have a good condition number. This well-behaved under-determined linear system gives us the freedom to design the reconstruction filters B with additional properties. The general formulation is:

$$\min_B G(B) \quad \text{subject to} \quad H(A)B = f \quad \text{and other constraints} \quad (37)$$

where $G(B)$ is the objective function that we use to impose the additional property that we expect B to have. For example, if we want the reconstruction filters to look like piecewise smooth function, we can use the following formulation:

$$\begin{aligned} \min_B \quad G(B) &:= \sum_{l=1}^m \|\nabla b_l\|_1 \\ \text{subject to} \quad & \|b_l\|_2 = \alpha, \quad l = 1, \dots, m \\ & H(A)B = f \end{aligned} \quad (38)$$

where ∇ is a discrete gradient operator and α is a predefined parameter whose purpose is to make the size of B compatible with the constraint $H(A)B = f$.

An illustration of the reconstruction filters is given in Figure 8(b).

4.2 Critically Down-sampled Case

In this case, we have less freedom and must consider the decomposition and reconstruction filters simultaneously. Since the constraint is bi-linear in A and B , in order to avoid the trivial situation where the objective function is minimized by scaling down the decomposition filters A and scaling up the reconstruction filters B , we require the filters A to have unit norm. Adopting the same notation as before, (25) becomes

$$\begin{aligned} \min_{A, B} \quad & \|W_A x\|_{1,1} \\ \text{subject to} \quad & H(A)B = f \\ & \|a_i\|_2 = 1, \quad i = 1, \dots, m \end{aligned} \quad (39)$$

Again, we apply the split Bregman algorithm to solve this problem. The procedures are similar to the redundant case. We will formulate the algorithm directly as follows:

Algorithm 3 Adaptive construction of bi-frames: critically down-sampled case

- 1: **Input:** x .
 - 2: **Initialize** $k = 0, F = \mathbf{0}, C = \mathbf{0}, A = A^0, B = B^0, D = W_{A^0}x$.
 - 3: **while** “not converge” **do**
 - 4: $D^{k+1} \leftarrow \arg \min_D \|D\|_{1,1} + \frac{\lambda}{2} \|D - W_{A^k}x - F^k\|_F^2$
 - 5: $A^{k+1} \leftarrow \arg \min_A \eta \|W_A x - D^{k+1} + F^k\|_F^2 + \lambda \|H(A)B^k - f + C^k\|_F^2$ **s.t.**
 $\|a_i\|_2 = 1, i = 1, \dots, m$
 - 6: $B^{k+1} \leftarrow \arg \min_B \lambda \|H(A^{k+1})B - f + C^k\|_F^2$
 - 7: $F^{k+1} \leftarrow F^k + W_{A^{k+1}}x - D^{k+1}$.
 - 8: $C^{k+1} \leftarrow C^k + H(A^{k+1})B^{k+1} - f - P^{k+1}$
 - 9: $k \leftarrow k + 1$
 - 10: **return** A^k, B^k
-

Updating D is again done by soft thresholding. The update of A is done by running a few iterations of the interior-point method, and updating B is done by running a few iterations of conjugate gradient method. The most computationally intensive step is updating A . But since in our applications, the support size and the number of filters are small, the total number of variables is normally a few hundred, hence the computational cost is reasonable.

5. Multi-level Adaptive Frames

Going to multi-level, the basic idea is to recursively use the framework of adaptive frames on the coefficients obtained by applying the adaptive filters to the signal. There are two practical issues that we need to consider. The first is whether one considers all the coefficients or a subset of coefficients when going to coarser level. In this regard the difference between low-pass and high-pass filters is particularly relevant. Recall that a low pass filter is defined by the condition that the Fourier coefficient $a(0) \neq 0$. The second issue is whether a new set of adaptive filters is learned and used at each level. We will discuss three different strategies that are motivated by three different examples.

5.1 The MRA approach

The basic idea of MRA is to apply the same set of filters at each level to the coefficients from the low-pass filters. When constructing traditional wavelet frames using MRA, there is only one low-pass filter at each level, the scaling function. All other filters are high pass filters associated with the wavelets. Our experience suggests that this is often the case for the adaptively learned filters. To makes sure that this is indeed

the case, we can also add the additional constraint

$$\hat{a}_1(0) \neq 0, \hat{a}_i(0) = 0, i = 2, \dots, m \quad (40)$$

to (17). As a linear constraint, this does not cause much trouble in the optimization algorithm. With this, the adaptive wavelet frames can be used in the same way as classical wavelet frames. Specifically, given the the input signal x , the multi-level decomposition proceeds as follows: We first perform a one-level decomposition to get the coefficients $v_i = T_{a_i}x, i = 1, \dots, m$. v_1 is associated with the low-pass filter, which provides the coarse-grained approximation of the signal, and $v_i, i = 2, \dots, m$ are associated with the high-pass filters, which provide the missing details from the coarse-graining. Next, we treat v_1 as the input signal and perform another one-level decomposition using the same set of filters to get the second-level coefficients. This procedure can then be continued. Schematically, this algorithm can be represented as a tree with one branching point at each level, as shown in Figure 2(a).

5.2 The scattering transform approach

By applying each fixed filter to the signal, one obtains a set of coefficients, called a feature map. If the input signal is an image, the feature map is also an image. One can then treat this new image as the input signal and find the corresponding adaptive filters. In some applications, this can be preceded by some component-wise nonlinear transformation. This is schematically shown in Figure 2(b). This structure is used in the scattering transforms proposed in Bruna and Mallat (2013).

The obvious drawback of this approach is that the degrees of freedom increase exponentially as the number of levels increases. Nevertheless, in classification tasks, it is generally believed that lifting the raw data to a high dimensional space using some nonlinear transforms can help by making the data more linearly separable. This is the underlying principle that makes kernel methods effective. Therefore this approach is potentially useful for classification tasks.

In practice, we can also apply some pruning procedure if there are many layers. For example, we can stop expanding the node if it has very small energy.

5.3 The convolutional net approach

The structure shown in Figure 2(c) resembles the first few layers of a convolutional net. The root node still represents the input signal, the first layer nodes represent the one-level decomposition coefficients. The coefficients together are then regarded as a multi-channel signal. For example, if the input signal is a monochrome two dimensional image, the first layer coefficients can be regards as a three dimensional image by stacking the m features maps. Once viewed as a three dimensional image, we can construct adaptive frames and bi-frames using three dimensional filters, except that the the filters might not be convolutional in the third dimension since the input signal is not expected to be translation invariant in that direction.

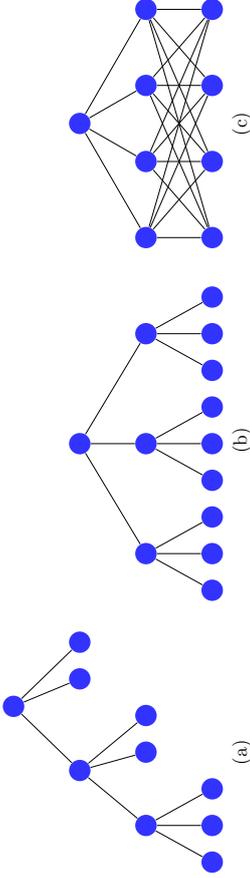


Figure 2: three structures

Figure 3: Illustration of the different multi-level structures. (a) The structure used in the MRA approach. (b) The structure used in the scattering transform approach. (c) The structure used in the convolutional net approach.

Obviously we are not limited by these three examples of multi-level structures. We call this way of representing the signal *multi-scale adaptive frames and bi-frames*. For convenience we abbreviate it as: *AdaFrame*.

6. Examples

6.1 The staircase signal

We consider a simple example where the signals are binary, each consists of long sequences of +1's separated by long sequences of -1's, as shown in Figure 4. Let s be the minimum length of consecutive +1 and -1 blocks. s is a measure of the lowest frequency of the signal. We use Algorithm 1 to learn the filters with $\eta = 10^2$. The filters learned are shown in Figure 5.



Figure 4: A binary signal

In the case when $m = 2, r = 2$, we recover the Haar wavelet basis as shown Figure 6. This example is simple enough to allow for analytic calculations. In fact, one can show that in the large s regime with the assumption that $r = m$, the filters learned should exactly be the ones shown in the figure. This simple example shows that adaptive filters do capture the special features of the data.

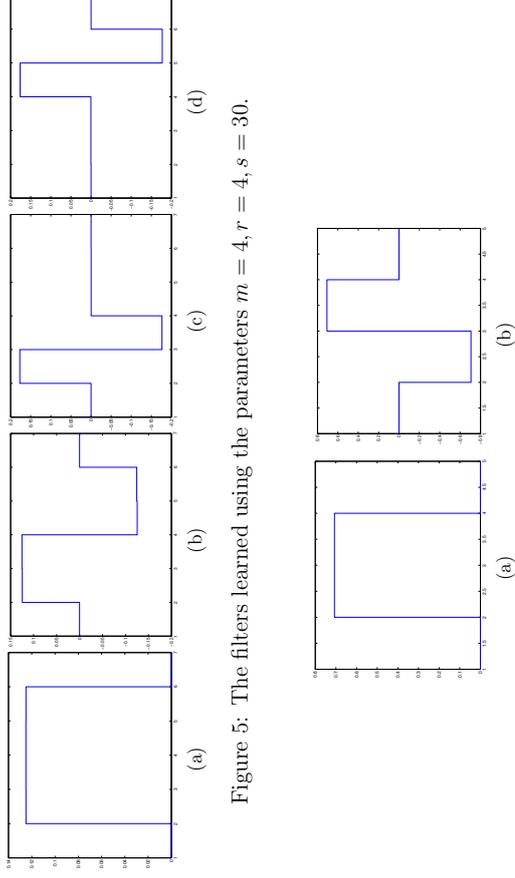


Figure 5: The filters learned using the parameters $m = 4, r = 4, s = 30$.

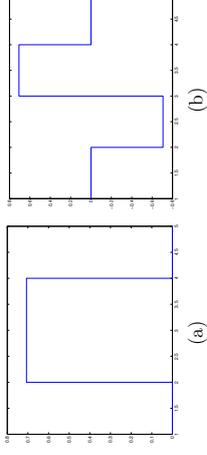


Figure 6: The filters learned using the parameters $m = 2, r = 2, s = 30$. In this case, we recover the Haar wavelets.

6.2 Fingerprint signal

Our next example is the fingerprint dataset (Maltoni et al., 2009). We use a fraction of the database. The input are 80 images of size 364×256 . Some sample images and filters learned are shown in Figure 7. The filters are learned using Algorithm 2 with parameters $\eta = 10^2, \lambda = 10^3$. The main feature of the fingerprint images is that they contain oscillations along different directions. As can be seen from the Figure 7, this feature is indeed captured by the learned filters.

6.3 Another test image

The next example is a well-known natural image shown in Figure 8. This is an example of the redundant bi-frame case. We learn the decomposition filters using Algorithm 2 with $\eta = 10^2, \lambda = 10^3$. Note that some filters look like edge detectors along different directions (e.g. the second and third filters in the first row act like edge detectors along the x and y axis). Most filters look like Gabor wavelets. They detect oscillations along different directions. Because this is an example of the redundant case, hence the reconstruction filters are not unique.

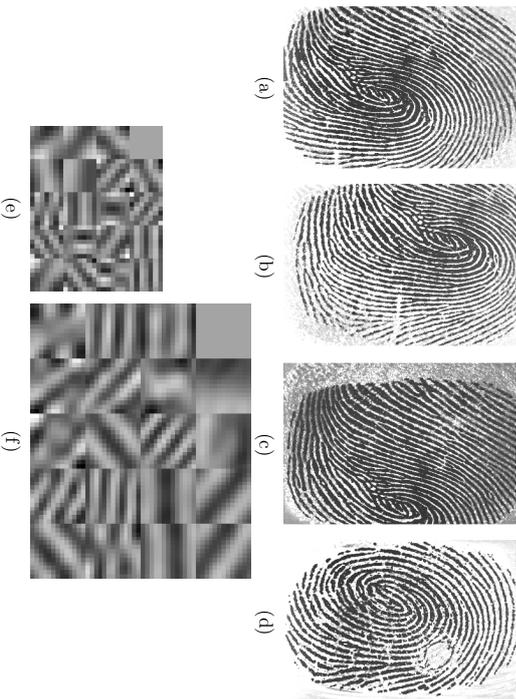


Figure 7: (a)(b)(c)(d) Sample images of finger print. (e) Decomposition filters learned with support size 7×7 . (f) Decomposition filters learned with support size 13×13 .

7. Recover Predefined Wavelets

The proposed framework is an adaptive extension of the well-known wavelets and wavelet frames. It is natural to ask whether the standard wavelet filters can be recovered using this framework. Naturally we expect that if the signal has a sparse representation in a predefined wavelet domain, then the adaptive frames and bi-frames would recover the predefined wavelets.

To see whether this is the case, we generate the signals using linear combinations of different wavelets with different levels of sparsity. Specifically, the signals are generated using 4 Daubechies wavelets of different support size, “db2”, “db3”, “db12”, “db24” in MATLAB syntax. Sparse random vectors with a given sparsity level are generated (the sparsity level is the ratio of the number of nonzero coefficients to the length of the coefficient vector, we also call it the density), and these vectors are used as the coefficients of the signals under the wavelet transform.

Given a signal, the adaptive filters are learned by solving (17). Since (17) is nonconvex, to avoid complications coming from local minimum, we used the simulated annealing algorithm to perform the global optimization. We then compare the filters obtained with the original wavelets used to construct the signal. We declare success if the l_2 norm of the difference between the adaptive filters and the predefined wavelets is smaller than 10^{-4} . Table 2 shows the success rate. 10 trials were performed for

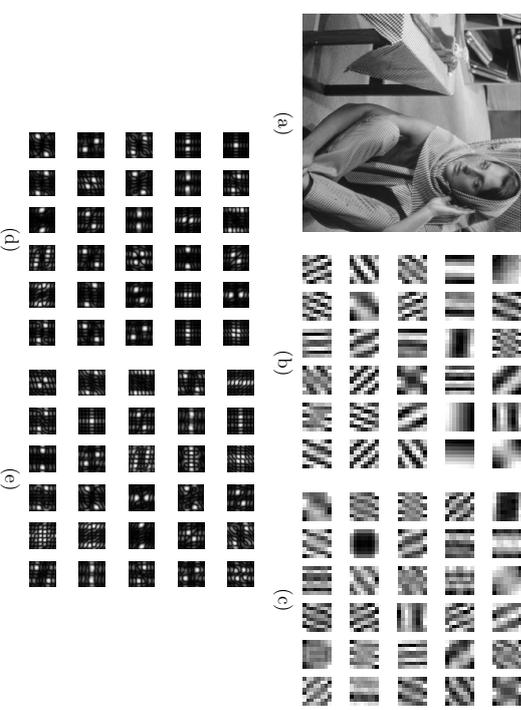


Figure 8: (a) Input image of size 512×512 . (b) 30 decomposition filters with support size 8×8 . (c) A specific set of reconstruction filters. (d) Fourier spectrum of the decomposition filters. (e) Fourier spectrum of the reconstruction filters.

Density	db2	db3	db12	db24
0.1	1	1	1	1
0.2	1	1	1	1
0.3	1	1	1	1
0.4	1	1	0	0
0.5	0	0	0	0

Table 2: Ratio of successful recovery of predefined wavelets.

each case. The result is indeed consistent with our expectation. It is interesting to see that the transition is very sharp.

Figure 9 shows the adaptive filters for the case when the signals are generated using a dense combination of the predefined wavelets. In this case, the predefined wavelets are not optimal, and the signals have a sparser representation under the adaptive filters, as can be seen from Figure 9(c). The L_1 norm of the wavelet coefficients is used as a robust measure of sparsity.

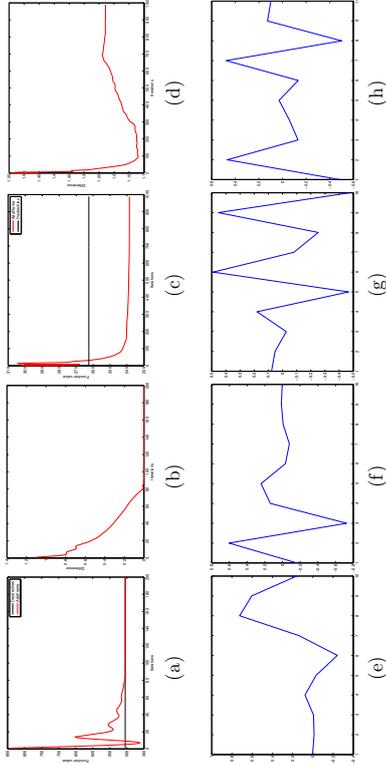


Figure 9: (a) The signal is generated using sparse linear combinations of the Daubechies wavelets. The black line is the objective function value evaluated using the Daubechies wavelets, which is optimal in this case. The value below the black line is due to infeasible intermediate solutions. (b) The filters learned also converge to the Daubechies wavelets, the figure shows the difference of the adaptive filters and the Daubechies wavelets measured in Frobenius norm. (c) The signal is generated using a dense linear combinations of the Daubechies wavelets. In this case, the objective function converges to a value lower than that of the the wavelets, indicated by the horizontal line. (d) The filters learned also converge, but to something different from the Daubechies wavelets. (e)(f) Decomposition filters of the Daubechies wavelet “db5”. The signal is generated using sparse linear combination of this wavelets, the filters learned are the same as the wavelets. (g)(h) The filters learned for signals generated using dense combinations of the “db5” wavelet. They are different from the wavelet filters.

8. Sample Applications

In this section, we discuss some examples of applications of the multi-scale adaptive frames, the AdaFrames. A thorough comparison of the proposed model and other existing models will be postponed to future publications.

8.1 Image Compression

AdaFrames are designed with the objective of making the decomposition coefficients sparse. Therefore they should be naturally suited for image compression tasks. As an initial step, we will compare the performance of AdaFrames with predefined Daubechies wavelets and Haar wavelets. We use the following simple compression scheme: Given an image x and the filters, we perform a decomposition to the coars-

est level to get the coefficients, but we keep only the coefficients with relatively large absolute values and set all the other coefficients to 0. The ratio of the total number of coefficients to the number of coefficients kept is called the “compression ratio” (the entropy coding stage is not considered here). We then perform a reconstruction step to get the reconstructed image \hat{x} . The quality of the compression is measured by the peak signal-to-noise ratio (PSNR). For monochrome 8 bit image, PSNR is defined as

$$\text{PSNR}(x, \hat{x}) = 10 \log_{10} \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (\hat{x}(i, j) - x(i, j))^2 \quad (41)$$

The filters are learned using image 8(a). 4 filters of support size 6×6 are learned using Algorithm 1 with $\eta = 10^2$. The coefficients are critically down-sampled with sampling matrix $M = \text{Diag}(2, 2)$. Initialization is done using the Daubechies filters db3. In general, we have found that using predefined wavelet frames as initialization works quite well. 7 levels of decompositions are performed using the architecture shown in Figure 2 and the same set of filters. The PSNR values are plotted against the “compression ratio” in Figure 10.

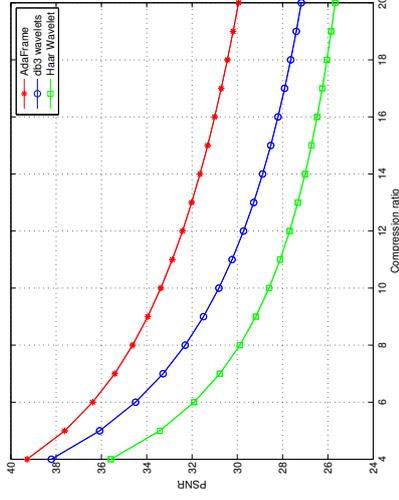


Figure 10: Image compression example. With the same image quality (measured in PSNR), AdaFrames achieves significantly higher compression ratio than the Haar wavelets and the Daubechies wavelets.

8.2 Image Denoising

As one of the simplest inverse problem, image denoising provides a convenient platform over which image processing ideas and techniques can be tested. Indeed, during the past few decades, many ideas from a diverse range of viewpoints have been proposed to address this problem, including wavelet domain thresholding, nonlocal means

(Buades et al., 2005), BM3D (Dabov et al., 2007), and the more recent ones based on dictionary learning (Elad and Aharon, 2006).

Among the various models, we select the K-SVD model (Elad and Aharon, 2006) as a benchmark for comparison since it is closely related to AdaFrames and since it has been shown to achieve the state of the art results.

Assume the image is corrupted by some additive noise:

$$g = f + n$$

where f is the clean image, g is our observation, and n is the noise with unknown distribution. First let us recall the procedure for wavelet domain denoising. Let W_A and R_A be the decomposition and reconstruction operators associated with the filters A respectively. Given an observed image x , the denoised image is then given by:

$$\hat{x} = R_A(\text{shrink}(W_A x)) \quad (42)$$

The procedure for AdaFrame denoising is exactly the same as that of wavelet domain denoising. Given the input image, we first learn the filters from the data using Algorithm 1 (or Algorithm 2 if we want to use bi-frames). We then use (42) to denoise.

In the first example, the input is a single image normalized to $[0, 1]$ and is corrupted with an additive Gaussian white noise with $\sigma = 0.1$. We train the filters both from the noisy image and the clean image with $m = 36, r = 6, \eta = 10^2, \lambda = 10^3$. A two-level decomposition is performed. The soft thresholding parameter is set to be 0.14. Initialization is done by setting the filters to be random orthogonal vectors. The result is shown in Figure 11. The performance of the K-SVD algorithm depends on the number of the atoms in the dictionary. Generally, the performance is better as we increase the number of atoms. In this example, 256 atoms with size 6×6 are used.

It is not surprising that the filters learned from a clean image produces better quality images: One can see from Figure 12 that the fine textures of the image are recovered. At a first sight, one might feel that this is impractical since we normally do not have access to the clean images. Nevertheless, there do exist realistic settings where learning from clean images makes sense. One such a situation is that filters learned from one set of clean images can then be used on another set of noisy images. We tested this idea on the extended Yale human face dataset B (Lee et al., 2005). It contains 16128 images of 28 human subjects. We used a subset of the images by picking the first 20 images of each of the subjects. We then added Gaussian white noise with $\sigma = 0.1$ to get the simulated noisy images. A glimpse of the dataset is in Figure 13.

To learn the filters, we pick the 100 clean images at random and use Algorithm 2 with $m = 36, r = 6, \eta = 10^2, \lambda = 10^3$. Two-levels of decompositions are performed. The soft-thresholding parameter is set to be 0.14. The results for the noisy images are reported in Table 3.



Figure 11: (a) Noisy input image, $\sigma = 0.1$. (b) K-SVD denoising result, PSNR=28.65dB. (c) AdaFrame denoising, filters learned from noisy image, PSNR=28.84dB. (d) AdaFrame denoising, filters learned from the clean image, PSNR=29.34dB.

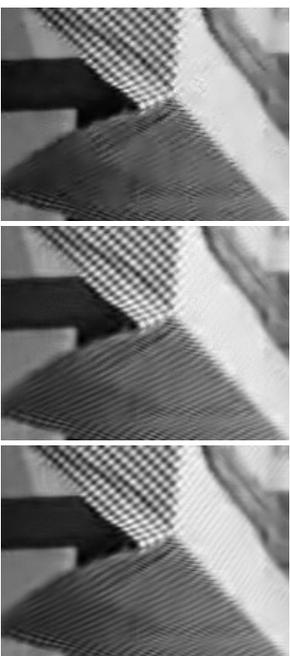


Figure 12: (a) Zoom in Figure 11(b). (b) Zoom in of Figure 11(c). (d) Zoom in of Figure 11(d).

	K-SVD, noisy	K-SVD, clean	AdaFrame, noisy	AdaFrame, clean
PSNR	31.4dB	32.01dB	31.35dB	32.07dB

Table 3: Average PSNR on the simulated noisy images on the extended Yale human face dataset B.

In another experiment, we test the performance of AdaFrame and K-SVD with different support sizes. We use some well-known benchmark images as test images. The images are normalized to $[0, 1]$ and the noise is Gaussian with $\sigma = 0.02, 0.05$ and 0.1 respectively. For K-SVD, 256 filters of support size 8×8 and 12×12 are used. For AdaFrame, 64 filters of support size 8×8 and 144 filters of support size 12×12



Figure 13: Simulated noisy images from extended Yale face dataset B

Test Image	K-SVD 8×8	K-SVD 12×12	AdaFrame 8×8	AdaFrame 12×12
Barbara $\sigma = 0.02$	38.02	38.00	37.34	38.21
Barbara $\sigma = 0.05$	33.28	33.01	31.87	33.22
Barbara $\sigma = 0.1$	29.47	29.24	29.18	29.70
Boat $\sigma = 0.02$	37.02	36.71	36.75	36.86
Boat $\sigma = 0.05$	32.53	32.11	32.50	32.59
Boat $\sigma = 0.1$	29.19	28.70	29.18	29.21
House $\sigma = 0.02$	39.45	39.25	39.18	39.17
House $\sigma = 0.05$	35.12	34.74	34.50	34.66
House $\sigma = 0.1$	32.15	32.05	31.19	31.45
Lena $\sigma = 0.02$	38.45	38.21	37.98	38.45
Lena $\sigma = 0.05$	34.46	34.18	33.21	34.34
Lena $\sigma = 0.1$	31.38	30.84	31.12	31.39
Peppers $\sigma = 0.02$	37.68	37.47	37.30	37.46
Peppers $\sigma = 0.05$	33.94	33.52	33.32	33.79
Peppers $\sigma = 0.1$	31.26	30.78	30.33	30.91

Table 4: Comparison of AdaFrame and K-SVD, performance measured in PSNR, the unit is dB.

are learned. λ is chosen based on the noise level and is set to be $\lambda = 0.005, 0.01, 0.025$ respectively. The result is shown in Table 5.

As a last denoising example, we apply AdaFrames to some examples of natural photos with unknown noise. The setting is the same as the previous example. We learn filters directly from the noisy images. Since the image has RGB channels, we learn the filters (of support size 9×9) for each channel separately with the same value of λ , which is chosen to yield a good visual impression. The results are shown in Figure 14.

As we emphasized earlier, the AdaFrame is faster than sparse coding technique at inference time. We record the computation time for the K-SVD denoising algorithm

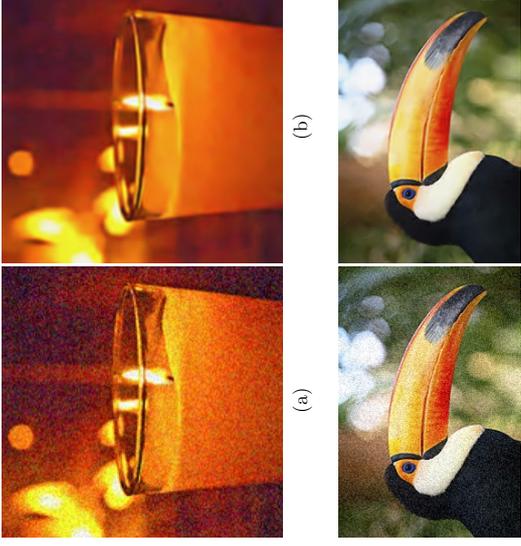


Figure 14: (a)(c) Two images from the Internet. (b)(d) Denoised images using AdaFrame.

and the AdaFrame denoising algorithm. In our laptop with the same setup, the K-SVD algorithm takes 25s to train a dictionary with 256 atoms of support size 8×8 and 6.5s to denoise the image. The software we use is downloaded from <http://www.cs.technion.ac.il/~ronrubin/software>. The AdaFrame takes 3.7s to train 64 filters with support size 8×8 and takes 0.6s to denoise. The time for denoising scales linearly with the number of filters.

8.3 Image Classification

Although AdaFrames are aimed to produce sparse representations, they can also be used to for other tasks such as extracting features for object recognition. In fact, it can provide a faster alternative to sparse coding.

To demonstrate this idea, in the following example, we apply AdaFrames to extract features in order to classify the handwritten digits. The dataset we used is MNIST (LeCun et al., 1998). It contains 70000 28×28 images of digits from 0 to 9, 60000 for training and 10000 for testing. A nonlinear transformation, the rectified linear function defined by $relu(x) = \max(x, 0)$ is applied to the coefficients obtained using

AdaFrames. The results are sent to a linear support vector machine (SVM) to perform the classification task. We discuss three different set of experiments.

In the first setup, we use Algorithm 2 to learn the filters with $m = 6$, $r = 6$, $\eta = 10^2$, $\lambda = 10^3$. Initialization is done with random orthogonal filters. For each image, we perform a one-level decomposition to get the coefficients.

The second setup is identical to the first one, except $m = 12$ instead of $m = 6$. It is generally believed that lifting the raw pixels to some higher-dimensional feature space will be helpful for classification. Since we use more filters in this setup, the features we get have higher dimensions. Indeed the results are better than the results of the previous setup.

In the third setup, we use a two-level decomposition. We use Algorithm 2 to learn the filters with $m = 6$, $r = 6$, $\eta = 10^2$, $\lambda = 10^3$. Same nonlinear transformation as in the previous setups are used. In this way, we obtain 6 feature maps, each of size 28×28 . Then the collection of the feature maps are treated as 6 sets of new input images. For each set, we use Algorithm 2 with $m = 4$, $r = 6$, $\eta = 10^2$, $\lambda = 10^3$ to learn the filters. Hence we have 24 filters in total. For each feature map, we perform a one-level decomposition using the corresponding 4 filters to get 4 feature maps. Again, we keep the positive coefficients and set the negative coefficients to 0. These positive coefficients in the first and second layers are the extracted features.

MNIST	Raw pixel	I	II	III
Precision	88.0 %	97.0 %	97.4 %	99.0 %

Table 5: Results of the MNIST classification. ‘‘Raw pixel’’ means that the features are the raw pixels.

These features are sent to a linear SVM. The results are reported in Table 5. Note that there is a significant reduction in the error rates compared to raw pixel features. As a point of comparison, the state-of-the-art result with preprocessing, is 0.23%, which is obtained using deep convolutional neural networks (Chresan et al., 2012).

9. Connection with De-convolutional net

Convolutional nets have had remarkable successes in a variety of challenging applications (LeCun et al., 1998; Lee et al., 2009; Krizhevsky et al., 2012). A typical supervised convolutional net consists of several convolutional layers and fully connected layers. A convolutional layer has the structure shown in Figure 15. It maps the feature maps produced by the previous layer to another set of feature maps. The input feature maps are first convolved with some filters, which are also obtained from training. A point-wise nonlinear function, called the ‘‘activation function’’, such as a rectified linear function is then applied, followed by a pooling procedure in order to down-sample the set of feature maps. Pooling is usually a local operation. Max pooling, namely picking the feature map with the maximum amplitude in a small

neighborhood of each node, is the most popular. It is similar to simple down-sampling but is nonlinear.

Although convolutional nets are designed for feature extraction and object recognition, it is an interesting question to ask how much of the input data can be reconstructed from the information in the intermediate layers of the network. For one thing, this can help us to gain some intuition about how convolutional nets work.

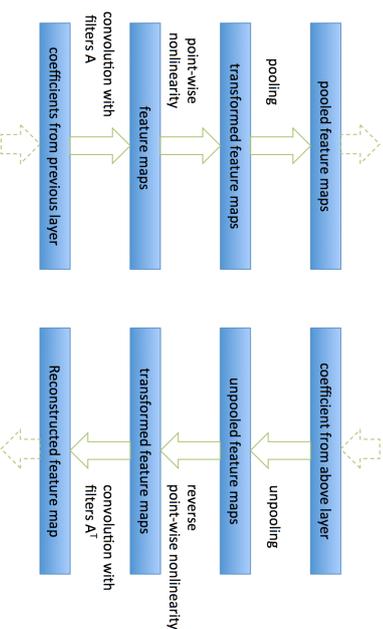


Figure 15: The left figure shows the typical structure of a convolutional layer from a convolutional net, the right figure shows the structure of a de-convolutional layer from a de-convolutional net.

In this regard the most popular approach in the literature is the ‘‘deconvolutional net’’ (Zelner et al., 2010). A deconvolutional net can be thought of as a convolutional net that uses the same components (filtering, nonlinear activation, pooling) but in reverse order. Specifically a deconvolutional net consists of the following steps: First, the pooling procedure is reversed. If averaging or other linear operator is used for pooling, then to reverse it, one simply applies its transpose operator. The max-pooling procedure is a non-linear operation. For an image I , the max-pooling operation has two outputs, the maximum value and the position where the maximum value is obtained, defined as

$$(v, p)(x) = (\text{sign}(I(x)) \cdot \max_{x \in \mathcal{N}} |I(x)|, \arg \max_{x \in \mathcal{N}} |I(x)|)$$

where \mathcal{N} is the neighborhood of x . To reverse max-pooling, we set

$$I(x) = \begin{cases} v & : x = p, x \in \mathcal{N} \\ 0 & : x \neq p, x \in \mathcal{N} \end{cases}$$

The second component is to reverse the activation function. For invertible functions such as the sigmoid or the tanh function (LeCun et al., 1998), we simply take

their inverse. The situation where the activation function is non-invertible as is the case of the absolute value function is more complicated and is discussed in Waldspurger et al. (2012).

The third component is to reverse the convolution operator, hence the name “deconvolution net”. Since convolution is a linear operator, to reverse it, one applies its transpose (Zeiler and Fergus, 2013).

The above procedure is summarized in a diagram in Figure 15. Notice the similarity with applying wavelet frame transforms. A single level decomposition and reconstruction step of the wavelet frame transform can be described as in Figure 16. We see that if we ignore the point-wise nonlinearity, a convolutional or a deconvolutional layer is very similar to a decomposition and reconstruction step in wavelet bi-frame transform respectively.

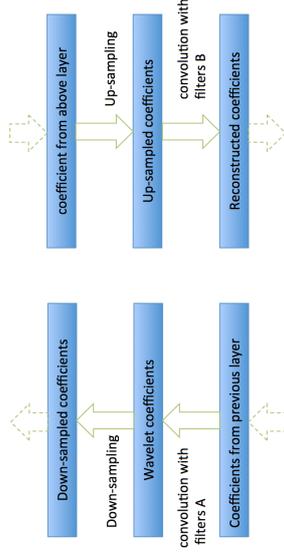


Figure 16: One level decomposition and reconstruction of AdaFrame

There is a subtle but important difference. In deconvolutional net, deconvolution is done by applying the transpose of the convolution operator. In the one level wavelet bi-frame reconstruction, this is done using the reconstruction filters, obtained by solving (10), as required by UEP. Since there is no guarantee that the UEP condition is satisfied by the filters obtained in the convolutional nets, one expects that there will be errors in the reconstruction process, i.e. the deconvolutional nets. This is indeed the case, as we show below.

The similarity between the convolutional layer and one level wavelet frame transform suggests a natural fix for this problem. Instead of using the flipped convolutional filters as the deconvolutional filters, we view the convolutional filters as the decomposition filters and solve (10) to obtain the reconstruction filters. These reconstruction filters are then used as the deconvolutional filters. Everything else is the same as in the original deconvolutional net. The existence of a solution to (10) is guaranteed by the fact that in a typical convolutional net, the number of filters is large, and hence we are in the the redundant case for the wavelet bi-frames. This small change to the deconvolutional net yields much better reconstruction as we now demonstrate.

We implemented a two-layer convolutional network. In the first layer, we have 12 filters of support size 6×6 , the pooling procedure is chosen to be the usual down-sampling with decimation factor (2, 2). To construct the second layer, we stack together the feature maps from the first layer and form a three-dimensional signal. We then learn 12 filters of support size $4 \times 4 \times 2$, the pooling procedure is also down-sampling with decimation factor (2, 1, 2). The activation function is the sigmoid function. The results of reconstructing the input image using the original deconvolutional net and the modified procedure described above are shown in Figure 17. As one can see, using the deconvolutional net approach, we gradually lose information as we ascend in the layers, while using the AdaFrame, we do not lose information.

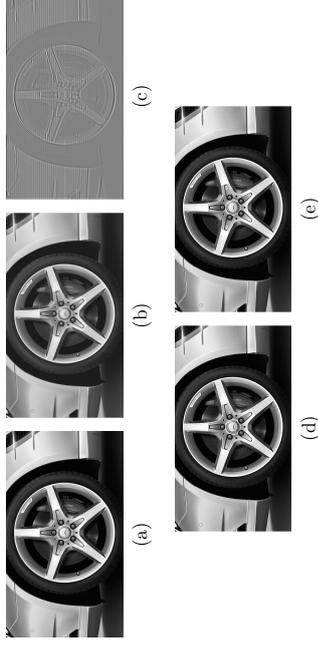


Figure 17: (a) The input image. (b) Reconstruction from the first layer activations using “deconvolutional net” approach. (c) Reconstruction from the second layer activations using the “deconvolutional net” approach. (d) Reconstruction from the first layer activations using the AdaFrame. (e) Reconstruction from the second layer activations using the AdaFrame.

In addition to near perfect reconstruction, AdaFrame has the potential to be used as an initialization method for the convolutional parts of a typical convolutional net. This is a direction for future research.

10. Conclusion

Predefined wavelets and dictionary learning have both been very successful in their own ways. In this paper, we have proposed a framework, the AdaFrame, that naturally combines the advantages of both. It is multi-scale and computationally efficient as pre-defined wavelets and wavelet frames, while being adaptive as in dictionary learning. Unlike dictionary learning, the proposed framework guarantees perfect reconstruction, which is an appealing property in many signal processing tasks.

Between adaptive frames and adaptive bi-frames, our experience suggests that adaptive bi-frames are much easier to use because of the additional flexibility. The

learning procedure is also easier, especially when the system is very redundant in which case the learning procedure can be carried out in two phases by learning the decomposition and reconstruction filters separately.

In addition to the examples given in this paper, we believe that the proposed framework can be useful in many other applications. It is not restricted to image processing, it can be used on time series, videos and even graphs. We will explore these applications in subsequent papers.

Another direction for future investigation is to use the proposed framework as feature extraction tools for machine learning tasks. Sparse coding has been popular for this purpose. But the proposed framework should be a promising alternative since it is more efficient and it has a multi-scale structure. It should be particularly appealing when the computation cost is the main bottleneck, as is the case in some real-time object recognition systems.

Acknowledgments

This work is supported in part by the 973 program of the Ministry of Science and Technology of China, the Major Program of NNSFC under grant 91130005 and an ONR grant N00014-13-1-0338.

Appendix

Proof of Theorem 1

For convenience, we need the following lemma.

Lemma 2 *Let M be $d \times d$ sampling matrix and $a, b \in l_2(\mathbb{Z}^d)$ be finitely supported sequences. Then*

$$\widehat{S_b v}(\xi) = |\det(M)| \widehat{v}(M^T \xi) \widehat{b}(\xi) \quad (43)$$

and

$$\widehat{T_a}(M^T \xi) = |\det(M)|^{-1} \sum_{\omega \in \Omega_M} \widehat{v}(\xi + 2\pi\omega) \overline{\widehat{a}(\xi + 2\pi\omega)} \quad (44)$$

where

$$\widehat{a}(\xi) := \sum_{k \in \mathbb{Z}^d} a(k) e^{-ik \cdot \xi}$$

and

$$\Omega_M := \{[M^T]^{-1} \mathbb{Z}^d\} \cap [0, 1]^d$$

Proof For a sequence $v \in l_2(\mathbb{Z}^d)$,

$$\begin{aligned} \widehat{S_b v}(\xi) &= \sum_{k \in \mathbb{Z}^d} (S_b v)(k) e^{-ik \cdot \xi} \\ &= |\det(M)| \sum_{k \in \mathbb{Z}^d} \sum_{j \in \mathbb{Z}^d} v(j) b(k - Mj) e^{-ik \cdot \xi} \\ &= |\det(M)| \sum_{k \in \mathbb{Z}^d} b(k - Mj) e^{-i(k - Mj) \cdot \xi} \sum_j v(j) e^{-iMj \cdot \xi} \\ &= |\det(M)| \widehat{b}(\xi) \widehat{v}(M^T \xi). \end{aligned} \quad (45)$$

Let $\widehat{u}(\xi) = \sum_{k \in \mathbb{Z}^d} v(k) \overline{\widehat{a}(k - n)}$, then $\widehat{u}(\xi) = \widehat{v}(\xi) \overline{\widehat{a}(\xi)}$. By definition of T_a , we have $(T_a v)(n) = u(Mn)$. So

$$\widehat{T_a v}(M^T \xi) = \sum_{n \in \mathbb{Z}^d} (T_a v)(n) e^{-in \cdot M^T \xi} = \sum_{n \in \mathbb{Z}^d} u(Mn) e^{-kMn \cdot \xi} \quad (46)$$

On the other hand,

$$\begin{aligned} \sum_{\omega \in \Omega_M} \widehat{u}(\xi + 2\pi\omega) &= \sum_{k \in \mathbb{Z}^d} \sum_{\omega \in \Omega_M} e^{-ik \cdot (\xi + 2\pi\omega)} \\ &= \sum_{k \in \mathbb{Z}^d} u(k) e^{-ik \cdot \xi} \sum_{\omega \in \Omega_M} e^{-ik \cdot 2\pi\omega}. \end{aligned} \quad (47)$$

If $k \in M\mathbb{Z}^d$, then $\sum_{\omega \in \Omega_M} e^{-ik \cdot 2\pi\omega} = |\det(M)|$; if $k \in \mathbb{Z}^d \setminus M\mathbb{Z}^d$, $\sum_{\omega \in \Omega_M} e^{-ik \cdot 2\pi\omega} = 0$, so we have

$$\sum_{\omega \in \Omega_M} \widehat{u}(\xi + 2\pi\omega) = |\det(M)| \sum_{k \in M\mathbb{Z}^d} u(k) e^{-ik \cdot \xi} = |\det(M)| \sum_{n \in \mathbb{Z}^d} u(Mn) e^{-iMn \cdot \xi}, \quad (48)$$

Combining this with (46), we get the desired result. \blacksquare

Lemma 3 Let M be $d \times d$ sampling matrix and $a_l, b_l, l = 1, \dots, m$ be m finitely supported sequences. Then

$$\sum_{l=1}^m \mathcal{S}_{b_l} \mathcal{T}_{a_l} v = v, \quad \forall v \in l_2(\mathbb{Z}^d) \quad (49)$$

$$\text{if and only if, for any } \omega \in \Omega_M := [(M^T)^{-1}\mathbb{Z}^d] \cap [0, 1)^d \quad (50)$$

$$\sum_{l=1}^m \hat{b}_l(\xi) \overline{\hat{a}_l(\xi + 2\pi\omega)} = \delta(\omega).$$

Proof By definition of the decomposition and reconstruction operators W_a and R_b , we have

$$R_b W_a v = \sum_{l=1}^m \mathcal{S}_{b_l} \mathcal{T}_{a_l} v. \quad (51)$$

which is equivalent to

$$\sum_{l=1}^m \widehat{\mathcal{S}_{b_l} \mathcal{T}_{a_l} v}(\xi) = \hat{v}(\xi), \quad \forall v \quad (52)$$

By the above lemma, we have

$$\begin{aligned} \hat{v}(\xi) &= \sum_{l=1}^m (\widehat{\mathcal{S}_{b_l} \mathcal{T}_{a_l} v})(\xi) \\ &= \sum_{l=1}^m |\det(M)| \widehat{\mathcal{T}_{a_l} v}(\xi) (M^T \xi) \hat{b}_l(\xi) \\ &= \sum_{l=1}^m \sum_{\omega \in \Omega_M} \hat{v}(\xi + 2\pi\omega) \hat{b}_l(\xi) \overline{\hat{a}_l(\xi + 2\pi\omega)} \end{aligned} \quad (53)$$

If (52) holds true, then

$$\sum_{l=1}^m \sum_{\omega \in \Omega} \hat{v}(\xi + 2\pi\omega) \hat{b}_l(\xi) \overline{\hat{a}_l(\xi + 2\pi\omega)} = \sum_{\omega \in \Omega_M} \hat{v}(\xi + 2\pi\omega) \delta(\omega) = \hat{v}(\xi). \quad (54)$$

holds for all $v \in l_2(\mathbb{Z}^d)$.

Conversely, if (51) is true, we can choose v that is close to a δ -function. Let $B_\epsilon(\xi_0)$ be the open ball centered at ξ_0 with radius ϵ . Fix $\omega_0 \in \Omega_M$ and $\xi_0 \in \mathbb{R}^d$, we can choose $v \in l_2(\mathbb{Z}^d)$ such that

1. $\hat{v}(\xi + 2\pi\omega_0) = 1$, for all $\xi \in B_\epsilon(\xi_0)$.
2. $\hat{v}(\xi + 2\pi\omega) = 0$, for all $\xi \in B_\epsilon(\xi_0), \omega \in \Omega \setminus \{\omega_0\}$.
3. $\text{supp}(\hat{v}) \subset 2\pi\omega_0 + B_{2\epsilon}(\xi_0)$

This is possible because the set Ω is discrete. \blacksquare

Hence, for $\xi \in B_\epsilon(\xi_0)$,

$$\begin{aligned} \hat{v}(\xi) &= \sum_{l=1}^m \sum_{\omega \in \Omega_M} \hat{v}(\xi + 2\pi\omega) \hat{b}_l(\xi) \overline{\hat{a}_l(\xi + 2\pi\omega)} \\ &= \sum_{l=1}^m \hat{b}_l(\xi) \overline{\hat{a}_l(\xi + 2\pi\omega_0)} \end{aligned} \quad (55)$$

Hence,

$$\sum_{l=1}^m \hat{b}_l(\xi) \overline{\hat{a}_l(\xi + 2\pi\omega_0)} = \delta(\omega)$$

for all $\xi \in B_\epsilon(\xi_0)$, since ξ_0 and ω_0 are arbitrary, we obtain the desired result. Proof of Theorem 1.

Proof We only need to establish that (10) is equivalent to (52).

$$\begin{aligned} \delta(\omega) &= \sum_{l=1}^m \sum_{k \in \mathbb{Z}^d} \overline{\hat{b}_l(k)} e^{ik \cdot \xi} \sum_{n \in \mathbb{Z}^d} a_l(n) e^{-in \cdot (\xi + 2\pi\omega)} \\ &= \sum_{l=1}^m \sum_{k, n \in \mathbb{Z}^d} \overline{\hat{b}_l(k)} a_l(n) e^{i(k-n) \cdot \xi} e^{-in \cdot 2\pi\omega} \end{aligned} \quad (56)$$

Denote by $\Gamma_M := (M[0, 1)^d) \cap \mathbb{Z}^d$, then we have $\mathbb{Z}^d = \Gamma_M + M\mathbb{Z}^d$, replace n by $Mn + \gamma$, we can rewrite the above equation as

$$\begin{aligned} \delta(\omega) &= \sum_{l=1}^m \sum_{\gamma \in \Gamma_M} \sum_{k, n \in \mathbb{Z}^d} \overline{\hat{b}_l(k)} a_l(Mn + \gamma) e^{i(k-Mn-\gamma) \cdot \xi} e^{-i(Mn+\gamma) \cdot 2\pi\omega} \\ &= \sum_{l=1}^m \sum_{\gamma \in \Gamma_M} \sum_{k, n \in \mathbb{Z}^d} \overline{\hat{b}_l(k + Mn + \gamma)} a_l(Mn + \gamma) e^{ik \cdot \xi} e^{-k\gamma \cdot 2\pi\omega} \end{aligned} \quad (57)$$

Note that $(e^{-i\gamma \cdot 2\pi\omega})_{\omega \in \Omega_M, \gamma \in \Gamma_M}$ is the Fourier matrix, and its inverse matrix is $|\det(M)|^{-1} (e^{i\gamma \cdot 2\pi\omega})_{\omega \in \Omega_M, \gamma \in \Gamma_M}$. Therefore,

$$\begin{aligned} & \left(\sum_{l=1}^m \sum_{k, n \in \mathbb{Z}^d} \overline{\hat{b}_l(k + Mn + \gamma)} a_l(Mn + \gamma) e^{ik \cdot \xi} \right)_{\gamma \in \Gamma_M} \\ &= |\det(M)|^{-1} (e^{i\gamma \cdot 2\pi\omega})_{\omega \in \Omega_M, \gamma \in \Gamma_M} (\delta(\omega))_{\omega \in \Omega} \\ &= |\det(M)|^{-1} (1, 1, \dots, 1)^T. \end{aligned} \quad (58)$$

Hence

$$\sum_{l=1}^m \sum_{k, n \in \mathbb{Z}^d} \frac{1}{b_l(k + Mn + \gamma)} a_l(Mn + \gamma) e^{ik \cdot \xi} = |\det(M)|^{-1} \cdot \forall \gamma \quad (59)$$

taking inverse Fourier transform, we get the desired result. ■

References

- Michal Aharon, Michael Elad, and Alfred Bruckstein. -svt: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54(11):4311–4322, 2006.
- Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.
- Jean Bruna and Stéphane Mallat. Invariant scattering convolution networks. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1872–1886, 2013.
- Ori Bryt and Michael Elad. Compression of facial images using the k-svd algorithm. *Journal of Visual Communication and Image Representation*, 19(4):270–282, 2008a.
- Ori Bryt and Michael Elad. Improving the k-svd facial image compression using a linear deblocking method. In *Electrical and Electronics Engineers in Israel, 2008. IEEE 2008. IEEE 25th Convention of*, pages 533–537. IEEE, 2008b.
- Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60–65. IEEE, 2005.
- Jian-Fang Cai, Hui Ji, Zuowei Shen, and Gui-Bo Ye. Data-driven tight frame construction and image denoising. *Applied and Computational Harmonic Analysis*, 37(1):89–105, 2014.
- Emmanuel J Candes and David L Donoho. Curvelets: A surprisingly effective non-adaptive representation for objects with edges. Technical report, DTIC Document, 2000.
- Dan Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649. IEEE, 2012.
- Ronald R Coifman and Mauro Maggioni. Diffusion wavelets. *Applied and Computational Harmonic Analysis*, 21(1):53–94, 2006.
- Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *Image Processing, IEEE Transactions on*, 16(8):2080–2095, 2007.
- Ingrid Daubechies, Bin Han, Amos Ron, and Zuowei Shen. Framelets: Mra-based constructions of wavelet frames. *Applied and Computational Harmonic Analysis*, 14(1):1–46, 2003.
- Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on pure and applied mathematics*, 57(11):1413–1457, 2004.
- Mihai N Do and Martin Vetterli. Contourlets: a new directional multiresolution image representation. In *Signals, Systems and Computers, 2002. Conference Record of the Thirty-Sixth Asilomar Conference on*, volume 1, pages 497–501. IEEE, 2002.
- Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *Image Processing, IEEE Transactions on*, 15(12):3736–3745, 2006.
- Kjersti Engan, Sven Ole Aase, and J Hakon Husoy. Method of optimal directions for frame design. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings, 1999 IEEE International Conference on*, volume 5, pages 2443–2446. IEEE, 1999a.
- Kjersti Engan, Sven Ole Aase, and JH Husoy. Frame based signal compression using method of optimal directions (mod). In *Circuits and Systems, 1999. ISCAS 99. Proceedings of the 1999 IEEE International Symposium on*, volume 4, pages 1–4. IEEE, 1999b.
- Tom Goldstein and Stanley Osher. The split bregman method for l1-regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.
- John C Gower and Garnt B Dijkstraerhuis. *Procrustes problems*, volume 3. Oxford University Press Oxford, 2004.
- Bin Han. Pairs of frequency-based nonhomogeneous dual wavelet frames in the distribution space. *Applied and Computational Harmonic Analysis*, 29(3):330–353, 2010.
- RA Horn and CR Johnson. Topics in matrix analysis, 1991. *Cambridge University Press, Cambridge*.
- Kevin Jarrett, Koray Kavukcuoglu, M Ranzato, and Yann LeCun. What is the best multi-stage architecture for object recognition? In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2146–2153. IEEE, 2009.

- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- Rongjie Lai and Stanley Osher. A splitting method for orthogonality constrained problems. *Journal of Scientific Computing*, 58(2):431–449, 2014.
- Quoc V Le, Alexandre Karpenko, Jiquan Ngiam, and Andrew Y Ng. Ica with reconstruction cost for efficient overcomplete feature learning. In *Advances in Neural Information Processing Systems*, pages 1017–1025, 2011.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient sparse coding algorithms. In *Advances in neural information processing systems*, pages 801–808, 2006.
- Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 609–616. ACM, 2009.
- Kuang-Chih Lee, Jeffrey Ho, and David Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(5):684–698, 2005.
- Julien Mairal, Guillermo Sapiro, and Michael Elad. Learning multiscale sparse representations for image and video restoration. Technical report, DTIC Document, 2007.
- Julien Mairal, Michael Elad, and Guillermo Sapiro. Sparse representation for color image restoration. *Image Processing, IEEE Transactions on*, 17(1):53–69, 2008.
- Stephane G Mallat. Multiresolution approximations and wavelet orthonormal bases of 2 (). *Transactions of the American mathematical society*, 315(1):69–87, 1989.
- Davide Maltoni, Dario Maio, Anil K Jain, and Salil Prabhakar. *Handbook of fingerprint recognition*. springer, 2009.
- Yves Meyer. *Wavelets and operators*, volume 1. Cambridge university press, 1995.
- Bruno A Olshausen et al. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.
- Stanley Osher, Martin Burger, Donald Goldfarb, Jinjun Xu, and Wotao Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 4(2):460–489, 2005.
- M Ranzato, Fu Jie Huang, Y-L Boureau, and Yann LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.
- Amos Ron and Zaowei Shen. Affine systems in l_2 (r d): The analysis of the analysis operator. *Journal of Functional Analysis*, 148(2):408–447, 1997.
- Zuowei Shen. Wavelet frames and image restorations. In *Proceedings of the International congress of Mathematicians*, volume 4, pages 2834–2863, 2010.
- Irène Waldspurger, Alexandre dAspremont, and Stéphane Mallat. Phase recovery, maxcut and complex semidefinite programming. *Mathematical Programming*, pages 1–35, 2012.
- Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional neural networks. *arXiv preprint ar-Xiv:1311.2901*, 2013.
- Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Robert Fergus. Deconvolutional networks. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2528–2535. IEEE, 2010.

Sparse PCA via Covariance Thresholding

Yash Deshpande

*Department of Electrical Engineering
Stanford University
Stanford, CA 94305, USA*

YASH.DESHPANDE@STANFORD.EDU

Andrea Montanari

*Departments of Electrical Engineering and Statistics
Stanford University
Stanford, CA 94305, USA*

MONTANARI@STANFORD.EDU

Editor: Alexander Rakhlin

Abstract

In sparse principal component analysis we are given noisy observations of a low-rank matrix of dimension $n \times p$ and seek to reconstruct it under additional sparsity assumptions. In particular, we assume here each of the principal components $\mathbf{v}_1, \dots, \mathbf{v}_r$ has at most s_0 non-zero entries. We are particularly interested in the high dimensional regime wherein p is comparable to, or even much larger than n .

In an influential paper, Johnstone and Lu (2004) introduced a simple algorithm that estimates the support of the principal vectors $\mathbf{v}_1, \dots, \mathbf{v}_r$ by the largest entries in the diagonal of the empirical covariance. This method can be shown to identify the correct support with high probability if $s_0 \leq K_1 \sqrt{n}/\log p$, and to fail with high probability if $s_0 \geq K_2 \sqrt{n}/\log p$ for two constants $0 < K_1, K_2 < \infty$. Despite a considerable amount of work over the last ten years, no practical algorithm exists with provably better support recovery guarantees.

Here we analyze a covariance thresholding algorithm that was recently proposed by Krauthgamer, Nadler, Vitenchik, et al. (2015). On the basis of numerical simulations (for the rank-one case), these authors conjectured that covariance thresholding correctly recover the support with high probability for $s_0 \leq K\sqrt{n}$ (assuming n of the same order as p). We prove this conjecture, and in fact establish a more general guarantee including higher-rank as well as n much smaller than p . Recent lower bounds (Berthet and Rigollet, 2013; Ma and Wigderson, 2015) suggest that no polynomial time algorithm can do significantly better.

The key technical component of our analysis develops new bounds on the norm of kernel random matrices, in regimes that were not considered before. Using these, we also derive sharp bounds for estimating the population covariance, and the principal component (with ℓ_2 -loss).

1. Introduction

In the spiked covariance model proposed by Johnstone and Lu (2004), we are given data $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ with $\mathbf{x}_i \in \mathbb{R}^p$ of the form¹:

$$\mathbf{x}_i = \sum_{q=1}^r \sqrt{\beta_q} u_{q,i} \mathbf{v}_q + \mathbf{z}_i, \quad (1)$$

Here $\mathbf{v}_1, \dots, \mathbf{v}_r \in \mathbb{R}^p$ is a set of orthonormal vectors, that we want to estimate, while $u_{q,i} \sim \mathcal{N}(0, 1)$ and $\mathbf{z}_i \sim \mathcal{N}(0, \mathbf{I}_p)$ are independent and identically distributed. The quantity $\beta_q \in \mathbb{R}_{>0}$ is a measure of signal-to-noise ratio. In the rest of this introduction, in order to simplify the exposition, we will refer to the rank one case and drop the subscript $q \in \{1, 2, \dots, r\}$. Further, we will assume n to be of the same order as p . Our results and proofs hold for a broad range of scalings of r, p, n , and will be stated in general form.

The standard method of principal component analysis involves computing the sample covariance matrix $\mathbf{G} = n^{-1} \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top$ and estimates $\mathbf{v} = \mathbf{v}_1$ by its principal eigenvector $\mathbf{v}_{\text{PC}}(\mathbf{G})$. It is a well-known fact that, in the high dimensional regime, this yields an inconsistent estimate (see Johnstone and Lu (2009)). Namely $\|\mathbf{v}_{\text{PC}} - \mathbf{v}\| \not\rightarrow 0$ unless $p/n \rightarrow 0$. Even worse, Baik, Ben Arous, and Pécché (2005) and Paul (2007) demonstrate the following phase transition phenomenon. Assuming that $p/n \rightarrow \alpha \in (0, \infty)$, if $\beta < \sqrt{\alpha}$ the estimate is *asymptotically orthogonal* to the signal, i.e. $\langle \mathbf{v}_{\text{PC}}, \mathbf{v} \rangle \rightarrow 0$. On the other hand, for $\beta > \sqrt{\alpha}$, $\langle \mathbf{v}_{\text{PC}}, \mathbf{v} \rangle$ remains bounded away from zero as $n, p \rightarrow \infty$. This phase transition phenomenon has attracted considerable attention recently within random matrix theory (see, e.g. Féral and Pécché, 2007; Capitaine et al., 2009; Benaych-Georges and Nadakuditi, 2011; Knowles and Yin, 2013).

These inconsistency results motivated several efforts to exploit additional structural information on the signal \mathbf{v} . In two influential papers, Johnstone and Lu (2004, 2009) considered the case of a signal \mathbf{v} that is sparse in a suitable basis, e.g. in the wavelet domain. Without loss of generality, we will assume here that \mathbf{v} is sparse in the canonical basis $\mathbf{e}_1, \dots, \mathbf{e}_p$. In a nutshell, Johnstone and Lu (2009) propose the following:

1. Order the diagonal entries of the Gram matrix $\mathbf{G}_{i(1),i(1)} \geq \mathbf{G}_{i(2),i(2)} \geq \dots \geq \mathbf{G}_{i(p),i(p)}$, and let $J \equiv \{i(1), i(2), \dots, i(k)\}$ be the set of indices corresponding to the s_0 largest entries.
2. Set to zero all the entries $\mathbf{G}_{i,j}$ of \mathbf{G} unless $i, j \in J$, and estimate \mathbf{v} with the principal eigenvector of the resulting matrix.

Johnstone and Lu formalized the sparsity assumption by requiring that \mathbf{v} belongs to a weak ℓ_q -ball with $q \in (0, 1)$. Instead, here we consider a strict sparsity constraint where \mathbf{v} has exactly s_0 non-zero entries, with magnitudes bounded below by $\theta/\sqrt{s_0}$ for some constant $\theta > 0$. Amini and Wainwright (2009) studied the more restricted case when every entry of \mathbf{v} has equal magnitude of $1/\sqrt{s_0}$. Within this restricted model, they proved diagonal thresholding successfully recovers the support of \mathbf{v} provided the sample size n satisfies²

1. Throughout the paper, we follow the convention of denoting scalars by lowercase, vectors by lowercase boldface, and matrices by uppercase boldface letters.

2. Throughout the introduction, we write $f(n) \gtrsim g(n)$ as a shorthand of ' $f(n) \geq K g(n)$ for a some constant $K = K(r, \beta)$ '.

$n \gtrsim s_0^2 \log p$ (see Amini and Wainwright, 2009). This result is a striking improvement over vanilla PCA. While the latter requires a number of samples scaling with the number of parameters $n \gtrsim p$, sparse PCA via diagonal thresholding achieves the same objective with a number of samples that scales with the number of *non-zero* parameters, $n \gtrsim s_0^2 \log p$.

At the same time, this result is not as strong as might have been expected. By searching exhaustively over all possible supports of size s_0 (a method that has complexity of order p^{s_0}) the correct support can be identified with high probability as soon as $n \gtrsim s_0 \log p$. No method can succeed for much smaller n , because of information theoretic obstructions. We refer the reader to Amini and Wainwright (2009) for more details.

Over the last ten years, a significant effort has been devoted to developing practical algorithms that outperform diagonal thresholding, see e.g. Moghaddam et al. (2005); Zou et al. (2006); d’Aspremont et al. (2007, 2008); Witten et al. (2009). In particular, d’Aspremont et al. (2007) developed a promising M-estimator based on a semidefinite programming (SDP) relaxation. Amini and Wainwright (2009) also carried out an analysis of this method and proved that, if³ (i) $n \geq K(\beta) s_0 \log(p - s_0)p$, and (ii) the SDP solution has rank one, then the SDP relaxation provides a consistent estimator of the support of \mathbf{v} .

At first sight, this appears as a satisfactory solution of the original problem. No procedure can estimate the support of \mathbf{v} from less than $s_0 \log p$ samples, and the SDP relaxation succeeds in doing it from –at most– a constant factor more samples. This picture was upset by a recent, remarkable result by Krauthgamer et al. (2015) who showed that the rank-one condition assumed by Amini and Wainwright does not hold for $\sqrt{n} \lesssim s_0 \lesssim (n/\log p)$. This result is consistent with recent work of Berthet and Rigollet (2013) demonstrating that sparse PCA cannot be performed in polynomial time in the regime $s_0 \gtrsim \sqrt{n}$, under a certain computational complexity conjecture for the so-called planted clique problem.

In summary, the sparse PCA problem demonstrates a fascinating interplay between computational and statistical barriers.

From a statistical perspective, and disregarding computational considerations, the support of \mathbf{v} can be estimated consistently if and only if $s_0 \lesssim n/\log p$. This can be done, for instance, by exhaustive search over all the $\binom{p}{s_0}$ possible supports of \mathbf{v} . We refer to Vu and Lei (2012); Cai et al. (2013) for a minimax analysis.

From a computational perspective, the problem appears to be much more difficult. There is rigorous evidence (Berthet and Rigollet, 2013; ?; Ma and Wigderson, 2015; Wang et al., 2014) that no polynomial algorithm can reconstruct the support unless $s_0 \lesssim \sqrt{n}$. On the positive side, a very simple algorithm (Johnstone and Lu’s diagonal thresholding) succeeds for $s_0 \lesssim \sqrt{n}/\log p$.

Of course, several elements are still missing in this emerging picture. In the present paper we address one of them, providing an answer to the following question:

Is there a polynomial time algorithm that is guaranteed to solve the sparse PCA problem with high probability for $\sqrt{n}/\log p \lesssim s_0 \lesssim \sqrt{n}$?

³ Throughout the paper, we denote by K constants that can depend on problem parameters r and β . We denote by upper case C (lower case c) generic absolute constants that are bigger (resp. smaller) than 1, but which might change from line to line.

We answer this question positively by analyzing a covariance thresholding algorithm that proceeds, briefly, as follows. (A precise, general definition, with some technical changes is given in the next section.)

1. Form the empirical covariance matrix \mathbf{G} and set to zero all its entries that are in modulus smaller than τ/\sqrt{n} , for τ a suitably chosen constant.
2. Compute the principal eigenvector $\hat{\mathbf{v}}_1$ of this thresholded matrix.
3. Denote by $\mathbf{B} \subseteq \{1, \dots, p\}$ be the set of indices corresponding to the s_0 largest entries of $\hat{\mathbf{v}}_1$.
4. Estimate the support of \mathbf{v} by ‘cleaning’ the set \mathbf{B} . (Briefly, \mathbf{v} is estimated by thresholding $\mathbf{G}^{\mathbf{B}} \hat{\mathbf{v}}_{\mathbf{B}}$ with $\hat{\mathbf{v}}_{\mathbf{B}}$ obtained by zeroing the entries outside \mathbf{B} .)

Such a covariance thresholding approach was proposed in Krauthgamer et al. (2015), and is in turn related to earlier work by Bickel and Levina (2008B); Cai et al. (2010). The formulation discussed in the next section presents some technical differences that have been introduced to simplify the analysis. Notice that, to simplify proofs, we assume s_0 to be known: this issue is discussed in the next two sections.

The rest of the paper is organized as follows. In the next section we provide a detailed description of the algorithm and state our main results. The proof strategy for our results is explained in Section 3. Our theoretical results assume full knowledge of problem parameters for ease of proof. In light of this, in Section 4 we discuss a practical implementation of the same idea that does not require prior knowledge of problem parameters, and is data-driven. We also illustrate the method through simulations. The complete proofs are in Sections 5, 7 and 6 respectively.

A preliminary version of this paper appeared in (Deshpande and Montanari, 2014). This paper extends significantly the results in Deshpande and Montanari (2014). In particular, by following an analogous strategy, we improve greatly the bounds obtained by Deshpande and Montanari (2014). This significantly improves the regimes of (s_0, p, n) on which we can obtain non-trivial results. The proofs follow a similar strategy but are, correspondingly, more careful.

2. Algorithm and main results

We provide a detailed description of the covariance thresholding algorithm for the general model (1) in Table 1. For notational convenience, we shall assume that $2n$ sample vectors are given (instead of n): $\{\mathbf{x}_j\}_{1 \leq j \leq 2n}$.

We start by splitting the data into two halves: $(\mathbf{x}_j)_{1 \leq j \leq n}$ and $(\mathbf{x}_j)_{n+1 \leq j \leq 2n}$ and compute the respective sample covariance matrices \mathbf{G} and \mathbf{G}' respectively. Define Σ to be the population covariance minus identity, i.e.

$$\Sigma \equiv \sum_{q=1}^r \beta_q \mathbf{v}_q \mathbf{v}_q^\top. \quad (2)$$

Throughout, we let \mathbf{Q}_q and s_q denote the support of \mathbf{v}_q and its size respectively, for $q \in \{1, 2, \dots, r\}$. We further let $\mathbf{Q} = \cup_{q=1}^r \mathbf{Q}_q$ and $s_0 = |\mathbf{Q}|$. The matrix \mathbf{G} is used, in steps 1 to

Algorithm 1 Covariance Thresholding

- 1: **Input:** Data $(\mathbf{x}_i)_{1 \leq i \leq 2n}$, parameter $s_0 \in \mathbb{N}$, $\tau, \rho \in \mathbb{R}_{>0}$;
- 2: Compute the empirical covariance matrices $\mathbf{G} \equiv \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top / n$, $\mathbf{G}' \equiv \sum_{i=n+1}^{2n} \mathbf{x}_i \mathbf{x}_i^\top / n$;
- 3: Compute $\widehat{\Sigma} = \mathbf{G} - \mathbf{I}_p$ (resp. $\widehat{\Sigma}' = \mathbf{G}' - \mathbf{I}_p$);
- 4: Compute the matrix $\eta(\widehat{\Sigma})$ by soft-thresholding the entries of $\widehat{\Sigma}$;

$$\eta(\widehat{\Sigma})_{ij} = \begin{cases} \widehat{\Sigma}_{ij} - \frac{\tau}{\sqrt{n}} & \text{if } \widehat{\Sigma}_{ij} \geq \tau / \sqrt{n}, \\ 0 & \text{if } -\tau / \sqrt{n} < \widehat{\Sigma}_{ij} < \tau / \sqrt{n}, \\ \widehat{\Sigma}_{ij} + \frac{\tau}{\sqrt{n}} & \text{if } \widehat{\Sigma}_{ij} \leq -\tau / \sqrt{n}, \end{cases}$$

- 5: Let $(\widehat{\mathbf{v}}_q)_{q \leq r}$ be the first r eigenvectors of $\eta(\widehat{\Sigma})$;
- 6: **Output:** $\widehat{\mathbf{Q}} = \{i \in [p] : \exists q \text{ s.t. } |(\widehat{\Sigma}' \widehat{\mathbf{v}}_q)_i| \geq \rho\}$.

4 to obtain a good estimate $\eta(\widehat{\Sigma})$ for the low rank part of the population covariance Σ . The algorithm first computes $\widehat{\Sigma}$, a centered version of the empirical covariance of the samples as follows:

$$\widehat{\Sigma} \equiv \mathbf{G} - \mathbf{I}_p, \quad (3)$$

where $\mathbf{G} = n^{-1} \sum_{i \leq n} \mathbf{x}_i \mathbf{x}_i^\top$ is the sample covariance matrix.

It then obtains the estimate $\eta(\widehat{\Sigma}) \in \mathbb{R}^{p \times p}$ by *soft thresholding* each entry of $\widehat{\Sigma}$ at a threshold τ / \sqrt{n} . Explicitly:

$$(\eta(\widehat{\Sigma}))_{ij} \equiv \eta\left(\widehat{\Sigma}_{ij}; \frac{\tau}{\sqrt{n}}\right). \quad (4)$$

Here $\eta: \mathbb{R} \times \mathbb{R}_+ \rightarrow \mathbb{R}$ is the soft thresholding function

$$\eta(z; \lambda) = \begin{cases} z - \lambda & \text{if } z \geq \lambda \\ -z + \lambda & \text{if } z \leq -\lambda \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

In step 5 of the algorithm, this estimate is used to construct good estimates $\widehat{\mathbf{v}}_q$ of the eigenvectors \mathbf{v}_q . Finally, in step 6, these estimates are combined with the (independent) second half of the data \mathbf{G}' to construct estimators $\widehat{\mathbf{Q}}_q$ for the support of the individual eigenvectors \mathbf{v}_q . In the first two subsections we will focus on the estimation of Σ and the individual principal components. Our results on support recovery are provided in the final subsection.

2.1 Estimating the population covariance

Our first result bounds the estimation error of the soft thresholding procedure in operator norm.

Theorem 1 *There exist numerical constants $C_1, C_2, C > 0$ such that the following happens. Assume $n > C \log p$, $n > s_0^2$ and let $\tau_* = C_1(\beta \vee 1)\sqrt{\log(p/s_0^2)}$. We keep the thresholding level τ according to*

$$\tau = \begin{cases} \tau_* & \text{when } \tau_* \leq \sqrt{\log p}/2, s_0^2 \leq p/e \\ C_2 \tau_* & \text{when } \tau_* \geq \sqrt{\log p}/2, s_0 \leq p/e \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

. Then with probability $1 - o(1)$:

$$\|\eta(\widehat{\Sigma}) - \Sigma\|_{op} \leq C \sqrt{\frac{s_0^2(\beta^2 \vee 1)}{n}} \left(\log \frac{p}{s_0^2} \vee 1 \right). \quad (7)$$

At this point, it is useful to compare Theorem 1 with available results in the literature. Classical denoising theory (Donoho and Johnstone, 1994; Johnstone, 2015) provides upper bounds on the estimation error of soft-thresholding. However, estimation error is measured by (element-wise) ℓ_p norm, while here we are interested in operator norm.

Bickel and Levina (2008a,b); Karoui (2008); Cai, Zhang, Zhou, et al. (2010); Cai and Liu (2011) considered the operator norm error of thresholding estimators for structured covariance matrices. Specializing to our case of exact sparsity, the result of Bickel and Levina (2008a) implies that, with high probability:

$$\|\eta(\widehat{\Sigma}) - \Sigma\|_{op} \leq C_0 \sqrt{\frac{s_0^2 \log p}{n}}. \quad (8)$$

Here $\eta_H(\cdot, \cdot)$ is the hard-thresholding function: $\eta_H(z) = z \mathbb{1}(|z| \geq \tau / \sqrt{n})$, and the threshold is chosen to be $\tau = C_1 \sqrt{\log p}$. Also, $\eta_H(\mathbf{M})$ is the matrix obtained by thresholding the entries of \mathbf{M} . In fact, Cai et al. (2012) showed that the rate in (8) is minimax optimal over the class of sparse population covariance matrices, with at most s_0 non-zero entries per row, under the assumption $s_0^2/n \leq C(\log p)^{-3}$.

Theorem 1 ensures consistency under a weaker sparsity condition, viz. $s_0^2/n \rightarrow 0$ is sufficient. Also, the resulting rate depends on $\log(p/s_0^2)$ instead of $\log p$. In other words, in order to achieve $\|\eta(\widehat{\Sigma}) - \Sigma\|_{op} < \varepsilon$ for a fixed ε , it is sufficient $s_0 \lesssim \varepsilon \sqrt{n}$ as opposed to $s_0 \lesssim \sqrt{n} / \log p$.

Crucially, in this regime for $s_0 = \Theta(\varepsilon \sqrt{n})$, Theorem 1 suggests a threshold of order $\tau = \Theta(\sqrt{\log(1/\varepsilon)})$ as opposed to $\tau = C_1 \sqrt{\log p}$ which is used in Bickel and Levina (2008a); Cai et al. (2012). As we will see in Section 3, this regime mathematically more challenging than the one of Bickel and Levina (2008a); Cai et al. (2012). By setting $\tau = C_1 \sqrt{\log p}$ for a large enough constant C_1 , all the entries of $\widehat{\Sigma}$ outside the support of Σ are set to 0. In contrast, a large part of our proof is devoted to control the operator norm of the noise part of $\widehat{\Sigma}$.

2.2 Estimating the principal components

We next turn to the question of estimating the principal components $\mathbf{v}_1, \dots, \mathbf{v}_r$. Of course, these are not identifiable if there are degeneracies in the population eigenvalues $\beta_1, \beta_2, \dots, \beta_r$. We thus introduce the following identifiability condition.

A1 The spike strengths $\beta_1 > \beta_2 > \dots > \beta_r$ are all *distinct*. We denote by $\beta \equiv \max(\beta_1, \dots, \beta_r)$ and $\beta_{\min} \equiv \min_{q \neq d'}(\beta_1 - \beta_2, \beta_2 - \beta_3, \dots, \beta_r)$. Namely, β is the largest signal strength and β_{\min} is the minimum gap.

We measure estimation error through the following loss, defined for $\mathbf{x}, \mathbf{y} \in S^{p-1} \equiv \{\mathbf{v} \in \mathbb{R}^p : \|\mathbf{v}\| = 1\}$:

$$L(\mathbf{x}, \mathbf{y}) \equiv \frac{1}{2} \min_{s \in \{+1, -1\}} \|\mathbf{x} - s\mathbf{y}\|^2 \quad (9)$$

$$= 1 - |\langle \mathbf{x}, \mathbf{y} \rangle|. \quad (10)$$

Notice the minimization over the sign $s \in \{+1, -1\}$. This is required because the principal components $\mathbf{v}_1, \dots, \mathbf{v}_r$ are only identifiable up to a sign. Analogous results can obtained for alternate loss functions such as the projection distance:

$$L_p(\mathbf{x}, \mathbf{y}) \equiv \frac{1}{\sqrt{2}} \|\mathbf{x}\mathbf{x}^\top - \mathbf{y}\mathbf{y}^\top\|_F = \sqrt{1 - \langle \mathbf{x}, \mathbf{y} \rangle^2}. \quad (11)$$

The theorem below is an immediate consequence of Theorem 1. In particular, it uses the guarantee of Theorem 1 to show that the corresponding principal components of $\eta(\widehat{\Sigma})$ provide good estimates of the principal components \mathbf{v}_q , $1 \leq q \leq r$.

Theorem 2 *There exists a numerical constant C such that the following holds. Suppose that Assumption A1 holds in addition to the conditions $n > C \log p$, $s_0^2 < n$, and $s_0^2 < p/e$. Set τ as according to Theorem 1, and let $\widehat{\mathbf{v}}_1, \dots, \widehat{\mathbf{v}}_r$ denote the r principal eigenvectors of $\eta(\widehat{\Sigma}; \tau/\sqrt{n})$. Then, with probability $1 - o(1)$*

$$\max_{q \in [r]} L(\widehat{\mathbf{v}}_q, \mathbf{v}_q) \leq \frac{C}{\beta_{\min}^2} \frac{s_0^2(\beta^2 \vee 1)}{n} \log \frac{p}{s_0^2}. \quad (12)$$

Proof Let $\Delta \equiv \eta(\widehat{\Sigma}; \tau/\sqrt{n}) - \Sigma$. By Davis-Kahn sin-theta theorem (Davis and Kahn, 1970), we have, for $\beta_{\min} > \|\Delta\|_{op}$,

$$L(\widehat{\mathbf{v}}_q, \mathbf{v}_q) \leq \frac{1}{2} \left(\frac{\|\Delta\|_{op}}{\beta_{\min} - \|\Delta\|_{op}} \right)^2. \quad (13)$$

For $\beta_{\min}^2 > 2C(s_0^2(\beta^2 \vee 1)/n) \log(p/s_0^2)$, the claim follows by using Theorem 1. If $\beta_{\min}^2 \leq 2C(s_0^2(\beta^2 \vee 1)/n) \log(p/s_0^2)$, the claim is obviously true since $L(\widehat{\mathbf{v}}_q, \mathbf{v}_q) \leq 1$ always. ■

2.3 Support recovery

Finally, we consider the question of support recovery of the principal components \mathbf{v}_q . The second phase of our algorithm aims at estimating union of the supports $\mathbf{Q} = \mathbf{Q}_1 \cup \dots \cup \mathbf{Q}_r$ from the estimated principal components $\widehat{\mathbf{v}}_q$. Note that, although $\widehat{\mathbf{v}}_q$ is not even expected to be sparse, it is easy to see that the largest entries of $\widehat{\mathbf{v}}_q$ should have significant overlap with $\text{supp}(\mathbf{v}_q)$. Step 6 of the algorithm exploit this property to construct a consistent estimator $\widehat{\mathbf{Q}}_q$ of the support of the spike \mathbf{v}_q .

We will require the following assumption to ensure support recovery.

A2 There exist constants $\theta, \gamma > 0$ such that the following holds. The non-zero entries of the spikes satisfy $|v_{q,i}| \geq \theta/\sqrt{s_0}$ for all $i \in \mathbf{Q}_q$. Further, for any q, q' $|v_{q,i}|/|v_{q',i}| \leq \gamma$ for every $i \in \mathbf{Q}_q \cap \mathbf{Q}_{q'}$. Without loss of generality, we will assume $\gamma \geq 1$.

Theorem 3 *Assume the spiked covariance model of Eq. (1) satisfying assumptions A1 and A2, and further $n > C \log p$, $s_0^2 < n$, and $s_0^2 < p/e$ for C a large enough numerical constant. Consider the Covariance Thresholding algorithm of Table 1, with τ as in Theorem 1 $\rho = \beta_{\min} \theta / (2\sqrt{s_0})$.*

Then there exists $K_0 = K_0(\theta, \gamma, \beta, \beta_{\min})$ such that, if

$$n \geq K_0 s_0^2 r \log \frac{p}{s_0^2} \quad (14)$$

then the algorithm recovers the union of supports of \mathbf{v}_q with probability $1 - o(1)$ (i.e. we have $\widehat{\mathbf{Q}} = \mathbf{Q}$).

The proof in Section 7 also provides an explicit expression for the constant K_0 .

Remark 4 *In Assumption A2, the requirement on the minimum size of $|v_{q,i}|$ is standard in support recovery literature (see, e.g. Wainwright, 2009; Memshausen and Billmann, 2006). Additionally, however, we require that when the supports of $\mathbf{v}_q, \mathbf{v}_{q'}$ overlap, they are of the same order, quantified by the parameter γ . Relaxing this condition is a potential direction for future work.*

Remark 5 *Recovering the signed supports $\mathbf{Q}_{q,+} = \{i \in [p] : v_{q,i} > 0\}$ and $\mathbf{Q}_{q,-} = \{i \in [p] : v_{q,i} < 0\}$, up to a sign flip, is possible using the same technique as recovering the supports $\text{supp}(\mathbf{v}_q)$ above, and poses no additional difficulty.*

3. Algorithm intuition and proof strategy

For the purposes of exposition, throughout this section, we will assume that $r = 1$ and drop the corresponding subscript q .

Denoting by $\mathbf{X} \in \mathbb{R}^{n \times p}$ the matrix with rows $\mathbf{x}_1, \dots, \mathbf{x}_n$, by $\mathbf{Z} \in \mathbb{R}^{n \times p}$ the matrix with rows $\mathbf{z}_1, \dots, \mathbf{z}_n$, and letting $\mathbf{u} = (u_1, u_2, \dots, u_n)$, the model (1) can be rewritten as

$$\mathbf{X} = \sqrt{\beta} \mathbf{u} \mathbf{v}^\top + \mathbf{Z}. \quad (15)$$

Recall that $\widehat{\Sigma} = n^{-1} \mathbf{X}^\top \mathbf{X} - I_p = \mathbf{G} - I_p$. For $\beta > \sqrt{p/n}$, the principal eigenvector of \mathbf{G} , and hence of $\widehat{\Sigma}$ is positively correlated with \mathbf{v} , i.e. $|\langle \widehat{\mathbf{v}}_1(\widehat{\Sigma}), \mathbf{v} \rangle|$ is bounded away from zero. However, for $\beta < \sqrt{p/n}$, the noise component in $\widehat{\Sigma}$ dominates and the two vectors become asymptotically orthogonal, i.e. for instance $\lim_{n \rightarrow \infty} |\langle \widehat{\mathbf{v}}_1(\widehat{\Sigma}), \mathbf{v} \rangle| = 0$. In order to reduce the noise level, we must exploit the sparsity of the spike \mathbf{v} .

Now, letting $\beta' \equiv \beta \|\mathbf{u}\|^2/n \approx \beta$, and $\mathbf{w} \equiv \sqrt{\beta} \mathbf{Z}^\top \mathbf{u}/n$, we can rewrite $\widehat{\Sigma}$ as

$$\widehat{\Sigma} = \beta' \mathbf{v} \mathbf{v}^\top + \mathbf{v} \mathbf{w}^\top + \mathbf{w} \mathbf{v}^\top + \frac{1}{n} \mathbf{Z}^\top \mathbf{Z} - I_p. \quad (16)$$

For a moment, let us neglect the cross terms $(\mathbf{v}\mathbf{v}^\top + \mathbf{w}\mathbf{w}^\top)$. The ‘signal’ component $\beta' \mathbf{v}\mathbf{v}^\top$ is sparse with s_0^2 entries of magnitude $\beta'\theta^2/s_0$, which (in the regime of interest $s_0 = \sqrt{n}/C$) is equivalent to $C\theta^2\beta/\sqrt{n}$. The ‘noise’ component $\mathbf{Z}^\top\mathbf{Z}/n - \mathbf{I}_p$ is dense with entries of order $1/\sqrt{n}$. Assuming $s_0/\sqrt{n} < c$ for some small constant c , it should be possible to remove most of the noise by thresholding the entries at level of order $1/\sqrt{n}$. For technical reasons, we use the soft thresholding function $\eta: \mathbb{R} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$, $\eta(z; \tau) = \text{sgn}(z)(|z| - \tau)_+$. We will omit the second argument from $\eta(\cdot; \cdot)$ wherever it is clear from context.

Consider again the decomposition (16). Since the soft thresholding function $\eta(z; \tau/\sqrt{n})$ is affine when $z \gg \tau/\sqrt{n}$, we would expect that the following decomposition holds approximately (for instance, in operator norm):

$$\eta(\widehat{\Sigma}) \approx \eta(\beta' \mathbf{v}\mathbf{v}^\top) + \eta\left(\frac{1}{n}\mathbf{Z}^\top\mathbf{Z} - \mathbf{I}_p\right). \quad (17)$$

Since $\beta' \approx \beta$ and each entry of $\mathbf{v}\mathbf{v}^\top$ has magnitude at least θ^2/s_0 , the first term is still approximately rank one, with

$$\left\| \eta(\beta' \mathbf{v}\mathbf{v}^\top) - \beta \mathbf{v}\mathbf{v}^\top \right\|_{op} \leq \frac{s_0\tau}{\sqrt{n}}. \quad (18)$$

This is straightforward to see since soft thresholding introduces a maximum bias of τ/\sqrt{n} per entry of the matrix, while the factor s_0 comes due to the support size of $\mathbf{v}\mathbf{v}^\top$ (see Proposition 14 below for a rigorous argument).

The main technical challenge now is to control the operator norm of the perturbation $\eta(\mathbf{Z}^\top\mathbf{Z}/n - \mathbf{I}_p)$. We know that $\eta(\mathbf{Z}^\top\mathbf{Z}/n - \mathbf{I}_p)$ has entries of variance $\delta(\tau)/n$, for $\delta(\tau) \approx \exp(-c\tau^2)$. If entries were independent with mild tail conditions, this would imply –with high probability–

$$\left\| \eta\left(\frac{1}{n}\mathbf{Z}^\top\mathbf{Z} - \mathbf{I}_p\right) \right\|_{op} \lesssim C\delta(\tau)\sqrt{\frac{p}{n}} = C\exp(-c\tau^2)\sqrt{\frac{p}{n}}, \quad (19)$$

for some constant C . Combining the bias bound from Eq. (18) and the heuristic decomposition of Eq. (19) with the decomposition (17) results in the bound

$$\left\| \eta(\widehat{\Sigma}) - \beta \mathbf{v}\mathbf{v}^\top \right\|_{op} \leq \frac{s_0\tau}{\sqrt{n}} + C\exp(-c\tau^2)\sqrt{\frac{p}{n}}. \quad (20)$$

Our analysis formalizes this argument and shows that such a bound is correct when $p < n$.

The matrix $\eta(\mathbf{Z}^\top\mathbf{Z}/n - \mathbf{I}_p)$ is a special case of so-called inner-product kernel random matrices, which have attracted recent interest within probability theory (see El Karoui, 2010a,b; Cheng and Singer, 2013; Fan and Montanari, 2015). The basic object of study in this line of work is a matrix $\mathbf{M} \in \mathbb{R}^{p \times p}$ of the type:

$$M_{ij} = f_n\left(\frac{\langle \tilde{\mathbf{z}}_i, \tilde{\mathbf{z}}_j \rangle}{n} - \mathbb{1}(i = j)\right). \quad (21)$$

In other words, $f_n: \mathbb{R} \rightarrow \mathbb{R}$ is a kernel function and is applied entry-wise to the matrix $\mathbf{Z}^\top\mathbf{Z}/n - \mathbf{I}_p$, with \mathbf{Z} a matrix with independent standard normal entries as above and $\tilde{\mathbf{z}}_i \in \mathbb{R}^n$ are the columns of \mathbf{Z} .

The key technical challenge in our proof is the analysis of the operator norm of such matrices, when f_n is the soft-thresholding function, with threshold of order $1/\sqrt{n}$. Earlier results are not general enough to cover this case. El Karoui (2010a,b) provide conditions under which the spectrum of $f_n(\mathbf{Z}^\top\mathbf{Z}/n - \mathbf{I}_p)$ is close to a rescaling of the spectrum of $(\mathbf{Z}^\top\mathbf{Z}/n - \mathbf{I}_p)$. We are interested instead in a different regime in which the spectrum of $f_n(\mathbf{Z}^\top\mathbf{Z}/n - \mathbf{I}_p)$ is very different from the one of $(\mathbf{Z}^\top\mathbf{Z}/n - \mathbf{I}_p)$. Cheng and Singer (2013) consider n -dependent kernels, but their results are asymptotic and concern the weak limit of the empirical spectral distribution of $f_n(\mathbf{Z}^\top\mathbf{Z}/n - \mathbf{I}_p)$. This does not yield an upper bound on the spectral norm of $f_n(\mathbf{Z}^\top\mathbf{Z}/n - \mathbf{I}_p)$. Finally, Fan and Montanari (2015) consider the spectral norm of kernel random matrices for smooth kernels f , only in the proportional regime $n/p \rightarrow c \in (0, \infty)$.

Our approach to proving Theorem 1 follows instead the ε -net method: we develop high probability bounds on the maximum Rayleigh quotient:

$$\max_{\mathbf{y} \in \mathbb{S}^{p-1}} \langle \mathbf{y}, \eta(\mathbf{Z}^\top\mathbf{Z}/n - \mathbf{I}_p)\mathbf{y} \rangle = \max_{\mathbf{y} \in \mathbb{S}^{p-1}} \eta\left(\frac{\langle \tilde{\mathbf{z}}_i, \tilde{\mathbf{z}}_j \rangle}{n}, \frac{\tau}{\sqrt{n}}\right) y_i y_j, \quad (22)$$

by discretizing $\mathbb{S}^{p-1} = \{\mathbf{y} \in \mathbb{R}^p : \|\mathbf{y}\| = 1\}$, the unit sphere in p dimensions. For a fixed \mathbf{y} , the Rayleigh quotient $\langle \mathbf{y}, \eta(\mathbf{Z}^\top\mathbf{Z}/n - \mathbf{I}_p)\mathbf{y} \rangle$ is a (complicated) function of the underlying Gaussian random variables \mathbf{Z} . One might hope that it is Lipschitz continuous with some Lipschitz constant $B = B(n, p, \tau, \mathbf{y})$, thereby implying, by Gaussian isoperimetry (Ledoux, 2005), that it concentrates to the scale B around its expectation (i.e. 0). Then, by a standard union bound argument over a discretization of the sphere, one would obtain that the operator norm of $\eta(\mathbf{Z}^\top\mathbf{Z}/n - \mathbf{I}_p)$ is typically no more than $\sqrt{p} \sup_{\mathbf{y} \in \mathbb{S}^{p-1}} B(n, p, \tau, \mathbf{y})$.

Unfortunately, this turns out not to be true over the whole space of \mathbf{Z} , i.e. the Rayleigh quotient is not Lipschitz continuous in the underlying Gaussian variables \mathbf{Z} . Our approach, instead, shows that for *typical* values of \mathbf{Z} , we can control the gradient of $\langle \mathbf{y}, \eta(\mathbf{Z}^\top\mathbf{Z}/n - \mathbf{I}_p)\mathbf{y} \rangle$ with respect to \mathbf{Z} , and extract the required concentration only from such local information of the function. This is formalized in our concentration lemma 9, which we apply extensively while proving Theorem 1. This lemma is a significantly improved version of the analogous result in Deshpande and Montanari (2014).

4. Practical aspects and empirical results

Specializing to the rank one case, Theorems 2 and 3 show that Covariance Thresholding succeeds with high probability for a number of samples $n \gtrsim s_0^2$, while Diagonal Thresholding requires $n \gtrsim s_0^2 \log p$. The reader might wonder whether eliminating the $\log p$ factor has any practical relevance or is a purely conceptual improvement. Figure 1 presents simulations on synthetic data under the strictly sparse model, and the Covariance Thresholding algorithm of Table 1, used in the proof of Theorem 3. The objective is to check whether the $\log p$ factor has an impact at moderate p . We compare this with Diagonal Thresholding.

We plot the empirical success probability as a function of s_0/\sqrt{n} for several values of p , with $p = n$. The empirical success probability was computed by using 100 independent instances of the problem. A few observations are of interest: (i) Covariance Thresholding appears to have a significantly larger success probability in the ‘difficult’ regime where Diagonal Thresholding starts to fail; (ii) The curves for Diagonal Thresholding appear to

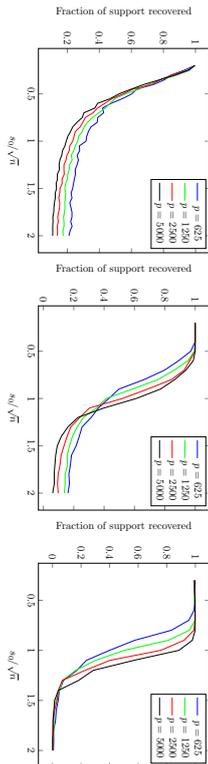


Figure 1: The support recovery phase transitions for Diagonal Thresholding (left) and Covariance Thresholding (center) and the data-driven version of Section 4 (right). For Covariance Thresholding, the fraction of support recovered correctly *increases* monotonically with p , as long as $s_0 \leq c\sqrt{n}$ with $c \approx 1.1$. Further, it appears to converge to one throughout this region. For Diagonal Thresholding, the fraction of support recovered correctly *decreases* monotonically with p for all s_0 of order \sqrt{n} . This confirms that Covariance Thresholding (with or without knowledge of the support size s_0) succeeds with high probability for $s_0 \leq c\sqrt{n}$, while Diagonal Thresholding requires a significantly sparser principal component.

decrease monotonically with p indicating that s_0 proportional to \sqrt{n} is not the right scaling for this algorithm (as is known from theory); (iii) In contrast, the curves for Covariance Thresholding become steeper for larger p , and, in particular, the success probability increases with p for $s_0 \leq 1.1\sqrt{n}$. This indicates a sharp threshold for $s_0 = \text{const} \cdot \sqrt{n}$, as suggested by our theory.

In terms of practical applicability, our algorithm in Table 1 has the shortcomings of requiring knowledge of problem parameters s_0, β, θ . Furthermore, the thresholds ρ, τ suggested by theory need not be optimal. We next describe a principled approach to estimating (where possible) the parameters of interest and running the algorithm in a purely data-dependent manner. Assume the following model, for $i \in [n]$

$$\mathbf{x}_i = \boldsymbol{\mu} + \sum_q \sqrt{\beta_q} u_{q,i} \mathbf{v}_q + \sigma \mathbf{z}_i,$$

where $\boldsymbol{\mu} \in \mathbb{R}^p$ is a fixed mean vector, $u_{q,i}$ have mean 0 and variance 1, and \mathbf{z}_i have mean 0 and covariance \mathbf{I}_p . Note that our focus in this section is not on rigorous analysis, but instead to demonstrate a principled approach to applying covariance thresholding in practice. We proceed as follows:

Estimating $\boldsymbol{\mu}, \sigma$: We let $\hat{\boldsymbol{\mu}} = \sum_{i=1}^n \mathbf{x}_i/n$ be the empirical mean estimate for $\boldsymbol{\mu}$. Further letting $\hat{\mathbf{X}} = \mathbf{X} - 1\hat{\boldsymbol{\mu}}^\top$ we see that $pm - (\sum_q k_q/n) \approx pm$ entries of $\hat{\mathbf{X}}$ are mean 0 and variance σ^2 . We let $\hat{\sigma} = \text{MAD}(\hat{\mathbf{X}})/\nu$ where $\text{MAD}(\cdot)$ denotes the median absolute deviation of the entries of the matrix in the argument, and ν is a constant scale factor. Guided by the Gaussian case, we take $\nu = \Phi^{-1}(3/4) \approx 0.6745$.

Choosing τ : Although in the statement of the theorem, our choice of τ depends on the SNR β/σ^2 , it is reasonable to instead threshold ‘at the noise level’, as follows. The

noise component of entry i, j of the sample covariance (ignoring lower order terms) is given by $\sigma^2 \langle \mathbf{z}_i, \mathbf{z}_j \rangle / n$. By the central limit theorem, $\langle \mathbf{z}_i, \mathbf{z}_j \rangle / \sqrt{n} \stackrel{d}{\sim} \mathcal{N}(0, 1)$. Consequently, $\sigma^2 \langle \mathbf{z}_i, \mathbf{z}_j \rangle / n \approx \mathcal{N}(0, \sigma^4/n)$, and we need to choose the (rescaled) threshold proportional to $\sqrt{\sigma^4} = \sigma^2$. Using previous estimates, we let $\tau = \nu' \cdot \hat{\sigma}^2$ for a constant ν' . In simulations, a choice $3 \lesssim \nu' \lesssim 4$ appears to work well.

Estimating τ : We define $\hat{\Sigma} = \hat{\mathbf{X}}^\top \hat{\mathbf{X}} / n - \hat{\sigma}^2 \mathbf{I}_p$ and soft threshold it to get $\eta(\hat{\Sigma})$ using τ as above. Our proof of Theorem 2 relies on the fact that $\eta(\hat{\Sigma})$ has τ eigenvalues that are separated from the bulk of the spectrum. Hence, we estimate τ using $\hat{\tau}$: the number of eigenvalues separated from the bulk in $\eta(\hat{\Sigma})$. The edge of the spectrum can be computed numerically using the Stieltjes transform method as in Cheng and Singer (2013).

Estimating \mathbf{v}_q : Let $\hat{\mathbf{v}}_q$ denote the q^{th} eigenvector of $\eta(\hat{\Sigma})$. Our theoretical analysis indicates that $\hat{\mathbf{v}}_q$ is expected to be close to \mathbf{v}_q . In order to denoise $\hat{\mathbf{v}}_q$, we assume $\hat{\Sigma} \hat{\mathbf{v}}_q \approx (1 - \delta) \mathbf{v}_q + \boldsymbol{\epsilon}_q$, where $\boldsymbol{\epsilon}_q$ is additive random noise (perhaps with some sparse correlations). We then threshold $\hat{\Sigma} \hat{\mathbf{v}}_q$ ‘at the noise level’ to recover a better estimate of \mathbf{v}_q . To do this, we estimate the standard deviation of the ‘noise’ $\boldsymbol{\epsilon}$ by $\hat{\sigma}_{\boldsymbol{\epsilon}} = \text{MAD}(\hat{\mathbf{v}}_q) / \nu$. Here we set –again guided by the Gaussian heuristic– $\nu \approx 0.6745$. Since \mathbf{v}_q is sparse, this procedure returns a good estimate for the size of the noise deviation. We let $\hat{\mathbf{v}}_q^*$ denote the vector obtained by hard thresholding $\hat{\mathbf{v}}_q$: set $\hat{\mathbf{v}}_q^* = \hat{\mathbf{v}}_q$, if $|\hat{v}_{q,i}| \geq \nu' \hat{\sigma}_{\boldsymbol{\epsilon}_q}$ and 0 otherwise. We then let $\hat{\mathbf{v}}_q^* = \hat{\mathbf{v}}_q^* / \|\hat{\mathbf{v}}_q^*\|$ and return $\hat{\mathbf{v}}_q^*$ as our estimate for \mathbf{v}_q .

Note that –while different in several respects– this empirical approach shares the same philosophy of the algorithm in Table 1. On the other hand, the data-driven algorithm presented in this section is less straightforward to analyze, a task that we defer to future work.

Figure 1 also shows results of a support recovery experiment using the ‘data-driven’ version of this section. Covariance thresholding in this form also appears to work for supports of size $s_0 \leq \text{const} \sqrt{n}$. Figure 2 shows the performance of vanilla PCA, Diagonal Thresholding and Covariance Thresholding on the ‘Three Peak’ example of Johnstone and Lu (2004). This signal is sparse in the wavelet domain and the simulations employ the data-driven version of covariance thresholding. A similar experiment with the ‘box’ example of Johnstone and Lu is provided in Figure 3. These experiments demonstrate that, while for large values of n both Diagonal Thresholding and Covariance Thresholding perform well, the latter appears superior for smaller values of n .

5. Proof preliminaries

In this section we review some notation and preliminary facts that we will use throughout the paper.

5.1 Notation

We let $[m] = \{1, 2, \dots, m\}$ denote the set of first m integers. We will represent vectors using boldface lower case letters, e.g. $\mathbf{u}, \mathbf{v}, \mathbf{x}$. The entries of a vector $\mathbf{u} \in \mathbb{R}^n$ will be represented

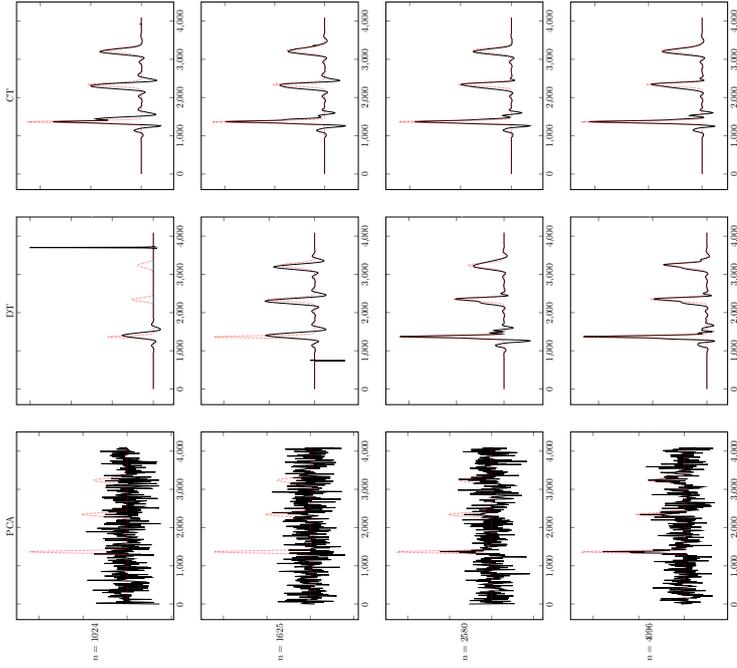


Figure 2: The results of Simple PCA, Diagonal Thresholding and Covariance Thresholding (respectively) for the “Three Peak” example of Johnstone and Lu (2009) (see Figure 1 of the paper). The signal is sparse in the ‘Symmet 8’ basis. We use $\beta = 1.4$, $p = 4096$, and the rows correspond to sample sizes $n = 1024, 1625, 2580, 4096$ respectively. Parameters for Covariance Thresholding are chosen as in Section 4, with $\nu' = 4.5$. Parameters for Diagonal Thresholding are from Johnstone and Lu (2009). On each curve, we superpose the clean signal (dotted).

by $u_i, i \in [n]$. Matrices are represented using boldface upper case letters e.g. \mathbf{A}, \mathbf{X} . The entries of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ are represented by \mathbf{A}_{ij} for $i \in [m], j \in [n]$. Given a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, we generically let $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m$ denote its rows, and $\tilde{\mathbf{a}}_1, \tilde{\mathbf{a}}_2, \dots, \tilde{\mathbf{a}}_n$ its columns.

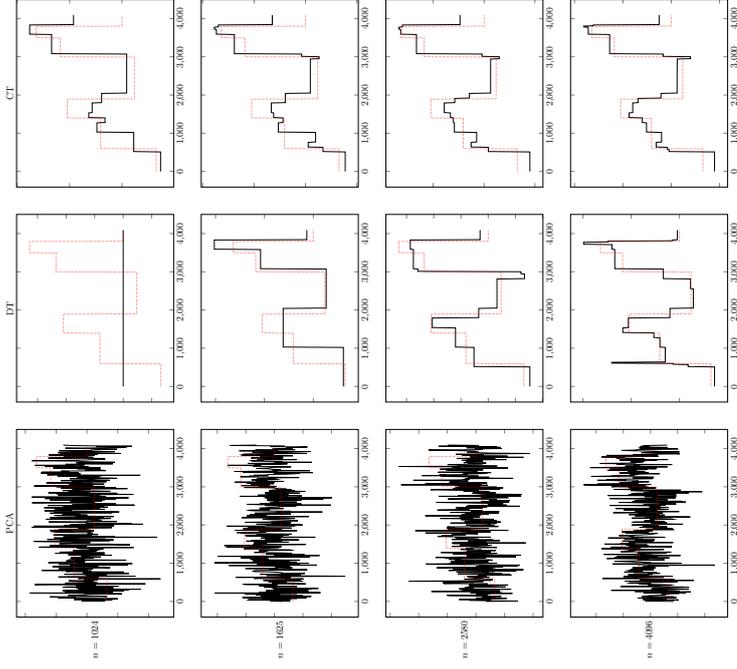


Figure 3: The results of Simple PCA, Diagonal Thresholding and Covariance Thresholding (respectively) for a synthetic block-constant function (which is sparse in the Haar wavelet basis). We use $\beta = 1.4, p = 4096$, and the rows correspond to sample sizes $n = 1024, 1625, 2580, 4096$ respectively. Parameters for Covariance Thresholding are chosen as in Section 4, with $\nu' = 4.5$. Parameters for Diagonal Thresholding are from Johnstone and Lu (2009). On each curve, we superpose the clean signal (dotted).

For $E \subseteq [m] \times [n]$, we define the projector operator $\mathcal{P}_E : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n}$ by letting $\mathcal{P}_E(\mathbf{A})$ be the matrix with entries

$$\mathcal{P}_E(\mathbf{A})_{ij} = \begin{cases} \mathbf{A}_{ij} & \text{if } (i, j) \in E, \\ 0 & \text{otherwise.} \end{cases} \quad (23)$$

For a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, and a set $E \subseteq [n]$, we define its column restriction $\mathbf{A}_E \in \mathbb{R}^{m \times n}$ to be the matrix obtained by setting to 0 columns outside E :

$$(\mathbf{A}_E)_{ij} = \begin{cases} \mathbf{A}_{ij} & \text{if } j \in E, \\ 0 & \text{otherwise.} \end{cases}$$

Similarly \mathbf{y}_E is obtained from \mathbf{y} by setting to zero all indices outside E . The operator norm of a matrix \mathbf{A} is denoted by $\|\mathbf{A}\|$ (or $\|\mathbf{A}\|_{op}$) and its Frobenius norm by $\|\mathbf{A}\|_F$. We write $\|\mathbf{x}\|$ for the standard ℓ_2 norm of a vector \mathbf{x} . Other vector norms such as ℓ_1 or ℓ_∞ are denoted with appropriate subscripts.

We let \mathbf{Q}_q denotes the support of the q^{th} spike \mathbf{v}_q . Also, we denote the union of the supports of \mathbf{v}_q by $\mathbf{Q} = \cup_q \mathbf{Q}_q$. The complement of a set $E \subseteq [n]$ is denoted by E^c .

We write $\eta(\cdot; \cdot)$ for the soft-thresholding function. By $\partial\eta(\cdot; \tau)$ we denote the derivative of $\eta(\cdot; \tau)$ with respect to the *first* argument, which exists Lebesgue almost everywhere. To simplify the notation, we omit the second argument when it is understood from context.

For a random variable Z and a measurable set \mathcal{A} we write $\mathbb{E}\{Z; \mathcal{A}\}$ to denote $\mathbb{E}\{Z\mathbb{1}(Z \in \mathcal{A})\}$, the expectation of Z constrained to the event \mathcal{A} .

In the statements of our results, consider the limit of large p and large n with certain conditions on p, n (as in Theorem 2). This limit will be referred to either as “*n* large enough” or “*p* large enough” where the phrase “large enough” indicates dependence of p (and thereby n) on specific problem parameters.

The Gaussian distribution function will be denoted by $\Phi(x) = \int_{-\infty}^x e^{-t^2/2} dt / \sqrt{2\pi}$.

5.2 Preliminary facts

Let \mathbb{S}^{N-1} denote the unit sphere in N dimensions, i.e. $\mathbb{S}^{N-1} = \{\mathbf{x} \in \mathbb{R}^N : \|\mathbf{x}\| = 1\}$. We use the following definition (see Vershynin, 2012, Definition 5.2) of the ε -net of a set $X \subseteq \mathbb{R}^n$:

Definition 6 (Nets, Covering numbers) A subset $T^\varepsilon(X) \subseteq X$ is called an ε -net of X if every point in X may be approximated by one in $T^\varepsilon(X)$ with error at most ε . More precisely:

$$\forall x \in X, \quad \inf_{y \in T^\varepsilon(X)} \|x - y\| \leq \varepsilon.$$

The minimum cardinality of an ε -net of X , if finite, is called its covering number.

The following two facts are useful while using ε -nets to bound the spectral norm of a matrix. For proofs, we refer the reader to (see Vershynin, 2012, Lemmas 5.2, 5.4).

Lemma 7 Let \mathbb{S}^{n-1} be the unit sphere in n dimensions. Then there exists an ε -net of \mathbb{S}^{n-1} , $T^\varepsilon(\mathbb{S}^{n-1})$ satisfying:

$$|T^\varepsilon(\mathbb{S}^{n-1})| \leq \left(1 + \frac{2}{\varepsilon}\right)^n.$$

Lemma 8 Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be a symmetric matrix. Then, there exists $\mathbf{x} \in T^\varepsilon(\mathbb{S}^{n-1})$ such that

$$|\langle \mathbf{x}, \mathbf{A}\mathbf{x} \rangle| \geq (1 - 2\varepsilon)\|\mathbf{A}\|. \quad (24)$$

Proof Firstly, we have $\|\mathbf{A}\| = \max_{\mathbf{x} \in \mathbb{S}^{n-1}} |\langle \mathbf{x}, \mathbf{A}\mathbf{x} \rangle| = \max_{\mathbf{x} \in \mathbb{S}^{n-1}} \|\mathbf{A}\mathbf{x}\|$. Let \mathbf{x}_* be the maximizer (which exists as \mathbb{S}^{n-1} is compact and $|\langle \mathbf{x}, \mathbf{A}\mathbf{x} \rangle|$ is continuous in \mathbf{x}). Choose $\mathbf{x} \in T_n^\varepsilon$ so that $\|\mathbf{x} - \mathbf{x}_*\| \leq \varepsilon$. Then:

$$\langle \mathbf{x}, \mathbf{A}\mathbf{x} \rangle = \langle \mathbf{x} - \mathbf{x}_*, \mathbf{A}(\mathbf{x} + \mathbf{x}_*) \rangle + \langle \mathbf{x}_*, \mathbf{A}\mathbf{x}_* \rangle. \quad (25)$$

The lemma then follows as $|\langle \mathbf{x}, \mathbf{A}(\mathbf{x} - \mathbf{x}_*) \rangle| \leq \|\mathbf{x} + \mathbf{x}_*\| \|\mathbf{A}\| \|\mathbf{x} - \mathbf{x}_*\| \leq 2\varepsilon \|\mathbf{A}\|$. ■

Throughout the paper we will denote by T_N^ε an ε -net on the unit sphere \mathbb{S}^{N-1} that satisfies Lemma 7. For a subset of indices $S \subseteq [N]$ we denote by $T_N^\varepsilon(S)$ the natural isometric embedding of T_S^ε in \mathbb{S}^{N-1} .

We now state a general concentration lemma. This will be our basic tool to establish Theorem 2, and thereby Theorem 3.

Lemma 9 Let $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I}_N)$ be the vector of N i.i.d. standard normal variables. Suppose S is a finite set and we have functions $F_s : \mathbb{R}^N \rightarrow \mathbb{R}$ for every $s \in S$. Assume $\mathcal{G} \in \mathbb{R}^N \times \mathbb{R}^N$ is a Borel set such that for Lebesgue-almost every $(\mathbf{x}, \mathbf{y}) \in \mathcal{G}$:

$$\max_{s \in S} \max_{t \in [0, 1]} \|\nabla F_s(\sqrt{t}\mathbf{x} + \sqrt{1-t}\mathbf{y})\| \leq L. \quad (26)$$

Then, for any $\Delta > 0$:

$$\mathbb{P}\left\{ \max_{s \in S} |F_s(\mathbf{z}) - \mathbb{E}F_s(\mathbf{z})| \geq \Delta \right\} \leq C|S| \exp\left(-\frac{\Delta^2}{CL^2}\right) + \frac{C}{\Delta^2} \mathbb{E}\left\{ \max_{s \in S} [(F_s(\mathbf{z}) - F_s(\mathbf{z}'))^2]; \mathcal{G}^c \right\}. \quad (27)$$

Here \mathbf{z}' is an independent copy of \mathbf{z} .

Proof We use the Mauney-Pisier method along with symmetrization. By centering, assume that $\mathbb{E}F_s(\mathbf{z}) = 0$ for all $s \in S$. Further, by including the functions $-F_s$ in the set S (at most doubling its size), it suffices to prove the one-sided version of the inequality:

$$\mathbb{P}\left\{ \max_{s \in S} F_s(\mathbf{z}) \geq \Delta \right\} \leq C|S| \exp\left(-\frac{\Delta^2}{CL^2}\right) + \frac{C}{\Delta^2} \mathbb{E}\left\{ \max_{s \in S} (F_s(\mathbf{z}) - F_s(\mathbf{z}'))^2; \mathcal{G}^c \right\}. \quad (28)$$

We first implement the symmetrization. Note that:

$$\{\mathbf{x} : \max_s F_s(\mathbf{x}) \geq \Delta\} \subseteq \{\mathbf{x} : \max_{x \in \mathbb{R}, s \in S} [2xF_s(\mathbf{x}) - x^2] \geq \Delta^2\} \quad (29)$$

$$\{\mathbf{x}, \mathbf{y} : \max_s [F_s(\mathbf{x}) - F_s(\mathbf{y})] \geq \Delta\} \subseteq \{\mathbf{x}, \mathbf{y} : \max_{x \in \mathbb{R}, s \in S} [2x(F_s(\mathbf{x}) - F_s(\mathbf{y})) - x^2] \geq \Delta^2\}. \quad (30)$$

Furthermore, by centering, $F_s(\mathbf{z}) = \mathbb{E}\{F_s(\mathbf{z}) - F_s(\mathbf{z}')|\mathbf{z}\}$. Hence for any non-decreasing convex function $\phi(z)$:

$$\mathbb{E}\left\{\phi\left(\max_{x,s} [2xF_s(\mathbf{z}) - x^2]\right)\right\} \leq \mathbb{E}\left\{\phi\left(\max_{x,s} [\mathbb{E}\{2xF_s(\mathbf{z}) - 2xF_s(\mathbf{z}') - x^2|\mathbf{z}\}]\right)\right\} \quad (31)$$

$$\stackrel{(a)}{\leq} \mathbb{E}\left\{\phi\left(\mathbb{E}\left\{\max_{x,s} [2x(F_s(\mathbf{z}) - F_s(\mathbf{z}') - x^2)|\mathbf{z}]\right\}\right)\right\} \quad (32)$$

$$\stackrel{(b)}{\leq} \mathbb{E}\left\{\phi\left(\max_{x,s} [2x(F_s(\mathbf{z}) - F_s(\mathbf{z}') - x^2)]\right)\right\}. \quad (33)$$

Here we use Jensen's inequality with the monotonicity of $\phi(\cdot)$ to obtain (a) and with the convexity of $\phi(\cdot)$ to obtain (b).

Now we choose $\phi(z) = (z - a)_+$, for $a = \Delta^2/2$.

$$\mathbb{P}\{\max_s F_s(\mathbf{z}) \geq \Delta\} \leq \mathbb{P}\{\max_{x,s} [2xF_s(\mathbf{z}) - x^2] \geq \Delta^2\} \quad (34)$$

$$\stackrel{(a)}{\leq} \phi(\Delta^2)^{-1} \mathbb{E}\left\{\phi\left(\max_{x,s} [2xF_s(\mathbf{z}) - x^2]\right)\right\} \quad (35)$$

$$\stackrel{(b)}{\leq} \phi(\Delta^2)^{-1} \mathbb{E}\left\{\phi\left(\max_{x,s} [2x(F_s(\mathbf{z}) - F_s(\mathbf{z}') - x^2)]\right)\right\} \quad (36)$$

$$= \phi(\Delta^2)^{-1} \mathbb{E}\left\{\phi\left(\max_s [(F_s(\mathbf{z}) - F_s(\mathbf{z}'))^2]\right)\right\} \quad (37)$$

$$= \phi(\Delta^2)^{-1} \left(\mathbb{E}\left\{\phi\left(\max_s [(F_s(\mathbf{z}) - F_s(\mathbf{z}')^2)]\right)\right\} \right. \\ \left. + \mathbb{E}\left\{\phi\left(\max_s [(F_s(\mathbf{z}) - F_s(\mathbf{z}'))^2]\right)\right\} \right). \quad (38)$$

Here (a) is Markov's inequality, and (b) is the symmetrization bound Eq.(33), where we use the fact that $\phi(z) = (z - a)_+$ is non-decreasing and convex in z .

At this point, it is easy to see that the lemma follows if we are able to control the first term in Eq.(38). We establish this via the Maurey-Pisier method. Define the path $\mathbf{z}(\theta) \equiv \mathbf{z} \sin \theta + \mathbf{z}' \cos \theta$, the velocity $\dot{\mathbf{z}} \equiv d\mathbf{z}/d\theta = \mathbf{z} \cos \theta - \mathbf{z}' \sin \theta$.

$$\mathbb{E}\left\{\phi\left(\max_s [(F_s(\mathbf{z}) - F_s(\mathbf{z}')^2)]\right)\right\} = \int_0^\infty \mathbb{P}\left\{\left(\max_s [(F_s(\mathbf{z}) - F_s(\mathbf{z}')^2)] - a\right)_+ \mathbb{I}(\mathcal{G}) \geq x\right\} dx \\ (39) \\ = \int_0^\infty \mathbb{P}\left\{\max_s [|F_s(\mathbf{z}) - F_s(\mathbf{z}')|] \geq \sqrt{x+a}; \mathcal{G}\right\} dx \quad (40) \\ \leq 2|S| \int_a^\infty e^{-\lambda\sqrt{x}} \max_s \left[\mathbb{E}\left\{\exp\{\lambda(F_s(\mathbf{z}) - F_s(\mathbf{z}'))\}; \mathcal{G}\right\} \right] dx, \quad (41)$$

where, in the last inequality we use the union bound followed by Markov's inequality. To control the exponential moment, note that $F_s(\mathbf{z}) - F_s(\mathbf{z}') = \int_0^{\pi/2} \langle \nabla F(\mathbf{z}(\theta)), \dot{\mathbf{z}}(\theta) \rangle d\theta$ whence,

using Jensen's inequality:

$$\mathbb{E}\left\{\exp\left\{\lambda(F_s(\mathbf{z}) - F_s(\mathbf{z}'))\right\}; \mathcal{G}\right\} = \mathbb{E}\left\{\exp\left(\int_0^{\pi/2} \lambda \langle \nabla F_s(\mathbf{z}(\theta)), \dot{\mathbf{z}}(\theta) \rangle d\theta\right); \mathcal{G}\right\} \quad (42)$$

$$\leq \frac{2}{\pi} \int_0^{\pi/2} \mathbb{E}\left\{\exp\left(\lambda\pi \langle \nabla F_s(\mathbf{z}(\theta)), \dot{\mathbf{z}}(\theta) \rangle / 2\right); \mathcal{G}\right\} d\theta. \quad (43)$$

Define the set $\mathcal{G}_\theta = \{(\mathbf{z}, \mathbf{z}') : \max_s \|\nabla F_s(\mathbf{z}(\theta))\| \leq L\}$. Then:

$$\mathbb{E}\left\{\exp\left\{\lambda(F_s(\mathbf{z}) - F_s(\mathbf{z}'))\right\}; \mathcal{G}\right\} \stackrel{(a)}{\leq} \frac{2}{\pi} \int_0^{\pi/2} \mathbb{E}\left\{\exp\left(\lambda\pi \langle \nabla F_s(\mathbf{z}(\theta)), \dot{\mathbf{z}}(\theta) \rangle / 2\right); \mathcal{G}_\theta\right\} d\theta \quad (44)$$

$$\stackrel{(b)}{=} \frac{2}{\pi} \int_0^{\pi/2} \mathbb{E}\left\{\exp\left(\frac{\lambda^2 \pi^2 \|\nabla F_s(\mathbf{z}(\theta))\|^2}{8}; \mathcal{G}_\theta\right)\right\} d\theta \quad (45)$$

$$\stackrel{(c)}{\leq} \exp\left(\frac{\lambda^2 \pi^2 L^2}{8}\right). \quad (46)$$

Here (a) follows as $\mathcal{G}_\theta \supseteq \mathcal{G}$. Equality (b) follows from noting that \mathcal{G}_θ is measurable with respect to $\mathbf{z}(\theta)$ and, hence, first integrating with respect to $\dot{\mathbf{z}}(\theta) = \mathbf{z} \cos \theta - \mathbf{z}' \sin \theta$, a Gaussian random variable that is independent of $\mathbf{z}(\theta)$. The final inequality (c) follows by using the fact that $\|\nabla F_s(\mathbf{z}(\theta))\| \leq L$ on the set \mathcal{G}_θ .

Since this bound is uniform over $s \in S$, we can use it in (41):

$$\mathbb{E}\left\{\phi\left(\max_s (F_s(\mathbf{z}) - F_s(\mathbf{z}'))\right); \mathcal{G}\right\} \leq 2|S| \int_a^\infty \exp\left(-\lambda\sqrt{x} + \frac{\lambda^2 \pi^2 L^2}{8}\right) dx \quad (47)$$

$$\leq \frac{4|S|}{\lambda^2} (1 + \lambda\sqrt{a}) \exp\left(-\lambda\sqrt{a} + \frac{\lambda^2 \pi^2 L^2}{8}\right) \quad (48)$$

We can now set $\lambda = 4\sqrt{a}/\pi^2 L^2$, to obtain the exponent above as $-2a/\pi^2 L^2 = -\Delta^2/\pi^2 L^2$. The prefactor $(1 + \lambda\sqrt{a})\lambda^{-2}$ is bounded by $CL^2 \max(L^2/\Delta^2)$ when $a = \Delta^2/2$. Therefore, as required, we obtain:

$$\mathbb{E}\left\{\phi\left(\max_s (F_s(\mathbf{z}) - F_s(\mathbf{z}'))\right); \mathcal{G}\right\} \leq C \max(1, L^4/\Delta^4) \exp\left(-\frac{\Delta^2}{CL^2}\right) \quad (49)$$

Combining this with Eq. (38) and the fact that $\phi(\Delta^2)^{-1} \leq C\Delta^{-2}$ gives Eq. (28) and, consequently, the lemma. \blacksquare

By a simple application of Cauchy-Schwarz, this lemma implies the following.

Corollary 10 *Under the same conditions as Lemma 9,*

$$\mathbb{P}\left\{\max_{s \in S} |F_s(\mathbf{z}) - \mathbb{E}F_s(\mathbf{z})| \geq \Delta\right\} \leq C|S| \exp\left(-\frac{\Delta^2}{CL^2}\right) \\ + \frac{C}{\Delta^2} \mathbb{P}\left\{\max_{s \in S} [(F_s(\mathbf{z}) - F_s(\mathbf{z}'))^4]^{1/2}\right\} \mathbb{P}\{\mathcal{G}^c\}^{1/2}. \quad (50)$$

The following two lemmas are well-known concentration of measure results. The forms below can be found in (Vershynin, 2012, Corollary 5.35), (Laurent and Massart, 2000, Lemma 1) respectively.

Lemma 11 Let $\mathbf{A} \in \mathbb{R}^{M \times N}$ be a matrix with i.i.d. standard normal entries, i.e. $\mathbf{A}_{ij} \sim \mathcal{N}(0, 1)$. Then, for every $t \geq 0$:

$$\mathbb{P}\left\{\|\mathbf{A}\|_{\text{op}} \geq \sqrt{M} + \sqrt{N} + t\right\} \leq \exp\left(-\frac{t^2}{2}\right). \quad (51)$$

Lemma 12 Let $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I}_N)$. Then

$$\mathbb{P}\{\|\mathbf{z}\|^2 \geq N + 2\sqrt{N}t + 2t\} \leq \exp(-t). \quad (52)$$

6. Proof of Theorem 1

Since $\widehat{\Sigma} = \mathbf{X}^T \mathbf{X} / n - \mathbf{I}_p$, we have:

$$\begin{aligned} \widehat{\Sigma} &= \sum_{q=1}^r \left\{ \frac{\beta_q \|\mathbf{u}_q\|^2}{n} \mathbf{v}_q(\mathbf{v}_q)^T + \frac{\sqrt{\beta_q}}{n} (\mathbf{z}^T \mathbf{u}_q)^T + (\mathbf{z}^T \mathbf{u}_q) \mathbf{v}_q^T \right\} \\ &\quad + \sum_{q \neq q'} \left\{ \frac{\sqrt{\beta_q \beta_{q'}} \langle \mathbf{u}_q, \mathbf{u}_{q'} \rangle}{n} \mathbf{v}_q(\mathbf{v}_{q'})^T \right\} + \frac{\mathbf{z}^T \mathbf{z}}{n} - \mathbf{I}_p. \end{aligned} \quad (53)$$

We let $\mathbf{D} = \{(i, j) : i \in [p] \setminus \mathbf{Q}\}$ be the diagonal entries not included in any support. (Recall that $\mathbf{Q} = \cup_q \mathbf{Q}_q$ denote the union of the supports.) Further let $\mathbf{E} = \mathbf{Q} \times \mathbf{Q}$, $\mathbf{F} = (\mathbf{Q}^c \times \mathbf{Q}^c) \setminus \mathbf{D}$, and $\mathbf{G} = [p] \times [p] \setminus (\mathbf{D} \cup \mathbf{E} \cup \mathbf{F})$, or, equivalently $\mathbf{G} = (\mathbf{Q} \times \mathbf{Q}^c) \cup (\mathbf{Q}^c \times \mathbf{Q})$. Since these are disjoint we have:

$$\eta(\widehat{\Sigma}) = \underbrace{\mathcal{P}_{\mathbf{E}}\{\eta(\widehat{\Sigma})\}}_{\mathbf{S}} + \underbrace{\mathcal{P}_{\mathbf{F}}\{\eta(\widehat{\Sigma})\}}_{\mathbf{N}} + \underbrace{\mathcal{P}_{\mathbf{G}}\{\eta(\widehat{\Sigma})\}}_{\mathbf{C}} + \underbrace{\mathcal{P}_{\mathbf{D}}\{\eta(\widehat{\Sigma})\}}_{\mathbf{D}}. \quad (54)$$

The first term corresponds to the ‘signal’ component, while the last three terms correspond to the ‘noise’ component.

Theorem 1 is a direct consequence of the next five propositions. The first demonstrates that, even for a low level of thresholding, viz. $\tau < \sqrt{\log p/2}$, the term \mathbf{N} has small operator norm. The second demonstrates that the soft thresholding operation preserves the signal in the term \mathbf{S} . The next two propositions show that the cross and diagonal terms \mathbf{C} and \mathbf{D} are negligible as well. Finally, in the last proposition, we demonstrate that, for the regime of thresholding far above the noise level, i.e. $\tau > C\sqrt{\log p}$, the noise terms \mathbf{N} and \mathbf{C} vanish entirely.

Proposition 13 Let \mathbf{N} denote the second term of Eq. (54). Since $\mathbf{F} = \mathbf{Q}^c \times \mathbf{Q}^c \setminus \mathbf{D}$,

$$\mathbf{N} = \mathcal{P}_{\mathbf{F}}\left(\eta(\widehat{\Sigma})\right) = \mathcal{P}_{\mathbf{F}}\left\{\eta\left(\frac{1}{n}\mathbf{Z}^T \mathbf{z}\right)\right\}. \quad (55)$$

Then, there exists an absolute constant C such that the following happens. Assuming that (i) $\tau < \sqrt{\log p/2}$ and (ii) $n > C \log p$, then with probability $1 - o(1)$

$$\|\mathbf{N}\|_{\text{op}} \leq C \left(\sqrt{\frac{p}{n}} \vee \frac{p}{n} \right) e^{-\tau^2/C}. \quad (56)$$

Proposition 14 Let \mathbf{S} denote the first term in Eq. (54):

$$\mathbf{S} = \mathcal{P}_{\mathbf{E}}\left\{\eta(\widehat{\Sigma})\right\}. \quad (57)$$

Assume that (i) $s_0/n < 1$ and (ii) $n > C \log p$. Then with probability $1 - o(1)$:

$$\|\mathbf{S} - \Sigma\|_{\text{op}} \leq \frac{2\tau s_0}{\sqrt{n}} + C(\beta \vee 1) \sqrt{\frac{s_0}{n}}. \quad (58)$$

Proposition 15 Let \mathbf{C} denote the matrix corresponding to the third term of Eq. (54):

$$\mathbf{C} = \mathcal{P}_{\mathbf{G}}\left\{\eta(\widehat{\Sigma})\right\}.$$

Assuming the conditions of Proposition 13 and, additionally, that $s_0^2 \leq p$, there exist constants C, c such that with probability $1 - o(1)$

$$\|\mathbf{C}\|_{\text{op}} \leq C \tau e^{-c\tau^2/(\beta \vee 1)} \sqrt{\frac{p}{n}} \vee \frac{p}{n}. \quad (59)$$

Proposition 16 Let \mathbf{D} denote the matrix corresponding to the third term of Eq. (54):

$$\mathbf{D} = \mathcal{P}_{\mathbf{D}}\left\{\eta(\widehat{\Sigma})\right\}.$$

With probability $1 - o(1)$ we have that $\|\mathbf{D}\|_{\text{op}} \leq C\sqrt{n-1} \log p$.

Proposition 17 For some absolute constant C_0 , we have for $\tau \geq C_0(\beta \vee 1)\sqrt{\log p}$ that, with probability $1 - o(1)$:

$$\forall i, j \quad N_{ij} = C_{ij} = 0. \quad (60)$$

Therefore, $\|\mathbf{N}\|_{\text{op}} = 0$ and $\|\mathbf{C}\|_{\text{op}} = 0$.

Remark 18 At this point we remark that the probability $1 - o(1)$ can be made quantitative, for e.g. of the form $1 - \exp(-\min(\sqrt{\beta}, n)/C_1)$, for every n large enough. For simplicity of exposition we do not pursue this in the paper.

We defer the proofs of Propositions 13, 14, 15, 16 and 17 to Sections 6.1, 6.2, 6.3, 6.4 and 6.5 respectively. By combining them for $\beta = O(1)$, we immediately obtain the following bound.

Theorem 19 There exist numerical constants C_0, C_1 such that the following happens. Assume $\beta \leq C_0$, $n > C_1 \log p$ and $\tau \leq \sqrt{\log p/2}$. Then with probability $1 - o(1)$:

$$\|\eta(\widehat{\Sigma}) - \Sigma\|_{\text{op}} \leq \frac{2\tau s_0}{\sqrt{n}} + C \left(\sqrt{\frac{p}{n}} \vee \frac{p}{n} \right) e^{-\tau^2/C} + C \sqrt{\frac{s_0 \vee \log p}{n}}. \quad (61)$$

Proof The proof is obtained by adding the error terms from Propositions 13, 14, 15 and 16, and noting that β is bounded. \blacksquare

Using Propositions 13, 14, 15 and 16, together with a suitable choice of τ , we obtain the proof of Theorem 1.

Proof [Proof of Theorem 1] Note that in the case $s_0^2 > p/e$ there is no thresholding and hence the result follows from the fact that $\|\tilde{\Sigma} - \Sigma\|_{op} \leq C\sqrt{p}/n$ (Vershynin, 2012, Remark 5.40).

We assume now that $s_0^2 \leq p/e$ and the case that $\tau_* = C_1(\beta \vee 1)\sqrt{\log(p/s_0^2)} \leq \sqrt{\log p}/2$. In that case we set $\tau = \tau_* \leq \sqrt{\log p}/2$. Below we will keep C_1 a large enough constant, and check that each of the error terms in Propositions 13, 14, 15 and 16 is upper bounded by (a constant times) the right-hand side of Eq. (7). Throughout C will denote a generic constant that can be made as large as we want, and can change from line to line.

We start from Proposition 13:

$$\|\mathbf{N}\|_{op} \leq C \left(\sqrt{\frac{p}{n}} \vee \frac{p}{n} \right) \left(\frac{s_0^2}{p} \right)^C \quad (62)$$

$$\leq C \sqrt{\frac{p}{n}} \left(\frac{p}{s_0^2} \right)^{-C-1} \vee C \sqrt{\left(\frac{p}{n} \right)^2 \left(\frac{p}{s_0^2} \right)^{-C-2}} \quad (63)$$

$$\leq C \sqrt{\frac{s_0^2}{n}} \left(\frac{p}{s_0^2} \right)^{-C} \vee C \sqrt{\left(\frac{s_0^2}{n} \right)^2 \left(\frac{p}{s_0^2} \right)^{-C}} \quad (64)$$

$$\leq C \sqrt{\frac{s_0^2}{n}} \log \frac{p}{s_0^2}, \quad (65)$$

where in the last step we used $(e s_0^2/p), (s_0^2/n) \leq 1$.

Next consider Proposition 14:

$$\begin{aligned} \|\mathbf{S} - \Sigma\|_{op} &\leq C \sqrt{\frac{s_0^2 \tau^2}{n}} + C \sqrt{\frac{s_0(\beta \vee 1)^2}{n}} \\ &\leq C \sqrt{\frac{s_0^2(\beta^2 \vee 1)}{n}} \log \frac{p}{s_0^2}. \end{aligned} \quad (66)$$

From Proposition 15, we get, using the same argument as in Eq. (65)

$$\begin{aligned} \|\mathbf{C}\|_{op} &\leq C \sqrt{\beta \vee 1} \left(\sqrt{\frac{p}{n}} \vee \frac{p}{n} \right) \left(\frac{s_0^2}{p} \right)^C \\ &\leq C(\beta \vee 1) \sqrt{\frac{s_0^2}{n}} \log \frac{p}{s_0^2}. \end{aligned} \quad (68)$$

Finally, the term of Proposition 16 is also bounded as desired using $\log p \leq s_0^2 \log(p/s_0^2)$ (dividing both sides by p and using the fact that $x \mapsto x \log(1/x)$ is increasing).

The case of $\tau_* \geq \sqrt{\log p}/2$ is easier. In that case, we can keep $\tau = C_2 \tau_*$ with C_2 large enough so that $\tau \geq C_0(\beta \vee 1)\sqrt{\log p}$ for C_0 of Proposition 17. Then, by Proposition 17, we

know that $\mathbf{N} = 0$ and $\mathbf{C} = 0$. Therefore we only need consider the terms $\mathbf{S} - \Sigma$ and \mathbf{D} . For these terms we can use Propositions 14 and 16 respectively and, arguing as in the earlier case $\tau_* \leq \sqrt{\log p}$, we obtain the desired result. \blacksquare

6.1 Proof of Proposition 13

Define $\tilde{\mathbf{N}}$ as

$$\tilde{\mathbf{N}} = \mathcal{P}_{nd} \left\{ \eta \left(\frac{1}{n} \mathbf{Z}^\top \mathbf{Z} \right) \right\}.$$

Since \mathbf{N} is a principal submatrix of $\tilde{\mathbf{N}}$, it suffices to prove the same bound for $\tilde{\mathbf{N}}$. Our main tool in the proof will be the concentration lemma 9 which we use on multiple occasions. With a view to using the lemma, we let $\mathbf{Z}' \in \mathbb{R}^{n \times p}$ denote an independent copy of \mathbf{Z} , and $\tilde{\mathbf{z}}'_i$ its i^{th} column. The proof relies on two preliminary lemmas. For some $A \geq 1$ (to be chosen later), we first state and prove the following lemma that controls the norm of *any principal submatrix* of $\tilde{\mathbf{N}}$ of size at most p/A .

Lemma 20 Fix any $A \geq 1$. There exists an absolute constants C, c such that:

$$\mathbb{P} \left\{ \max_{\mathbf{S} \subseteq [p], |\mathbf{S}| \leq p/A} \|\mathcal{P}_{\mathbf{S} \times \mathbf{S}}(\tilde{\mathbf{N}})\|_{op} \geq \Delta \right\} \leq C \exp \left(\frac{\log CA}{A} - \frac{n^2 \Delta^2}{C(n+p)} \right) + C \frac{(np)^C}{\Delta^2} \exp(-cn). \quad (70)$$

Proof For any subset $\mathbf{S} \subset [p]$ recall that $T_p^\varepsilon(\mathbf{S})$ denotes an ε -net of unit vectors in \mathbb{S}^{p-1} supported on the subset \mathbf{S} . For simplicity let $T(A) = \cup_{\mathbf{S}: |\mathbf{S}| \leq p/A} T_p^\varepsilon(\mathbf{S})$. It suffices, by Lemma 8, to control $\langle \mathbf{y}, \tilde{\mathbf{N}} \mathbf{y} \rangle$ on the set $T(A)$. In particular:

$$\mathbb{P} \left\{ \max_{\mathbf{S} \subseteq [p], |\mathbf{S}| \leq p/A} \|\mathcal{P}_{\mathbf{S} \times \mathbf{S}}(\tilde{\mathbf{N}})\|_{op} \geq \Delta \right\} \leq \mathbb{P} \left\{ \max_{\mathbf{y} \in T(A)} |\langle \mathbf{y}, \tilde{\mathbf{N}} \mathbf{y} \rangle| \geq \Delta(1 - 2\varepsilon) \right\}. \quad (71)$$

Consider the good set \mathcal{G}_1 given by:

$$\mathcal{G}_1 = \{(\mathbf{Z}, \mathbf{Z}') : \max(\|\mathbf{Z}\|, \|\mathbf{Z}'\|) \leq \sqrt{2}(\sqrt{n} + \sqrt{p})\}. \quad (72)$$

To use Lemma 9, we need to compute $\mathbb{E}\langle \mathbf{y}, \tilde{\mathbf{N}} \mathbf{y} \rangle$ and the gradient of $\langle \mathbf{y}, \tilde{\mathbf{N}} \mathbf{y} \rangle$ with respect to the underlying random variables \mathbf{Z} . Since $\eta(\cdot)$ is an odd function the expectation vanishes. To compute the gradient, we let $t \in [0, 1]$ and $\mathbf{W} = \sqrt{t} \mathbf{Z} + \sqrt{1-t} \mathbf{Z}'$, and consider $\langle \mathbf{y}, \tilde{\mathbf{N}} \mathbf{y} \rangle = \langle \mathbf{y}, \eta(\mathbf{W}^\top \mathbf{W}/n) \mathbf{y} \rangle$ as a function of the \mathbf{W} . Taking the gradient with respect to a column $\tilde{\mathbf{w}}_\ell$ for $\ell \in \mathbf{S}$:

$$\nabla_{\tilde{\mathbf{w}}_\ell} \langle \mathbf{y}, \tilde{\mathbf{N}} \mathbf{y} \rangle = \frac{y_\ell}{n} \sum_{i \neq \ell, i \in \mathbf{S}} \tilde{\mathbf{w}}_{ij} \partial \eta(\langle \tilde{\mathbf{w}}_i, \tilde{\mathbf{w}}_\ell \rangle / n) \quad (73)$$

$$= \frac{y_\ell}{n} \mathbf{W} \boldsymbol{\sigma}, \quad (74)$$

where

$$\sigma_i = \begin{cases} y_i \partial \eta(\langle \tilde{\mathbf{w}}_i, \tilde{\mathbf{w}}_\ell \rangle / n) & \text{if } i \neq \ell, i \in S \\ 0 & \text{otherwise.} \end{cases} \quad (75)$$

Since $\|\sigma\| \leq \|\mathbf{y}\| = 1$, we have that $\|\nabla_{\tilde{\mathbf{w}}_\ell} \langle \mathbf{y}, \tilde{\mathbf{N}}\mathbf{y} \rangle\|^2 \leq y_\ell^2 \|\mathbf{W}\|^2 / n^2$. Summing over $\ell \in S$ we obtain the gradient bound, holding on the good set \mathcal{G}_1 :

$$\|\nabla_{\mathbf{W}} \langle \mathbf{y}, \tilde{\mathbf{N}}\mathbf{y} \rangle\|^2 \leq \frac{\sum_{\ell} y_\ell^2}{n^2} \|\mathbf{W}\|^2 \quad (76)$$

$$\leq \frac{C(n+p)}{n^2}, \quad (77)$$

which holds because of triangle inequality and the fact that $\sqrt{t} + \sqrt{1-t} \leq \sqrt{2}$. We can now apply Lemma 9 to bound the RHS of Eq. (71) and get:

$$\begin{aligned} \mathbb{P}\left\{ \max_{S \subseteq [p], |S| \leq p/4} \mathcal{R}_{S \times S}(\tilde{\mathbf{N}}) \geq \Delta \right\} &\leq C|T(A)| \exp\left(-\frac{n^2 \Delta^2}{C(n+p)}\right) \\ &+ \frac{C}{\Delta^2} \mathbb{E}\left\{ \max_{y \in T} \langle \mathbf{y}, \tilde{\mathbf{N}}\mathbf{y} \rangle^2; \mathcal{G}_1^c \right\}. \end{aligned} \quad (78)$$

We can simplify the terms on the right-hand side to obtain the result of the lemma. With $\varepsilon = 1/4$, Stirling's approximation and Lemma 7 we have:

$$|T(A)| \leq \exp\left(p \frac{\log C A}{A}\right). \quad (79)$$

We use a crude bound on the complement of the good set \mathcal{G}_1 . It is easy to see that, for any unit vector \mathbf{y} , $\langle \mathbf{y}, \tilde{\mathbf{N}}\mathbf{y} \rangle^2 \leq \|\tilde{\mathbf{N}}\|_F^2 \leq \|\mathbf{Z}^T \mathbf{Z}\|_F^2 / n^2$. Cauchy-Schwarz then implies that

$$\mathbb{E}\{\max\langle \mathbf{y}, \tilde{\mathbf{N}}\mathbf{y} \rangle^2; \mathcal{G}_1^c\} \leq n^{-2} (\mathbb{E}\{\|\mathbf{Z}^T \mathbf{Z}\|_F^2\})^{1/2} \mathbb{P}\{\mathcal{G}_1^c\}^{1/2} \quad (80)$$

$$\leq (np)^C \exp(-c(n+p)), \quad (81)$$

where the bound on $\mathbb{P}\{\mathcal{G}_1^c\}$ follows from Lemma 11. This concludes the lemma. \blacksquare

Note that Lemma 20, with $A = 1$, tells us that $\|\tilde{\mathbf{N}}\|_{\text{op}}$ is of order $\sqrt{p/n} + (p/n)^2$ (uniformly in τ) with high probability. Already this non-asymptotic bound is non-trivial, since the previous results of Cheng and Singer (2013) and Fan and Montanari (2015) do not extend to this case. However, Proposition 13 is stronger, and establishes a rate of decay with the thresholding level τ .

The second lemma we require controls the Rayleigh quotient $\langle \mathbf{y}, \tilde{\mathbf{N}}\mathbf{y} \rangle$ when the entries of \mathbf{y} are ‘‘spread out’’.

Lemma 21 *Assume that $\tau \leq \sqrt{\log p}/2$. Given $A \geq 1$ and a unit vector \mathbf{y} , let $S = \{i : |y_i| \leq \sqrt{A/p}\}$ and $\mathbf{y}_S, \mathbf{y}_{S^c}$ denote the projections of \mathbf{y} onto supports S, S^c respectively. We have:*

$$\mathbb{P}\left\{ \max_{y \in T_p^{1/4}} |\langle \mathbf{y}_S, \tilde{\mathbf{N}}\mathbf{y}_S \rangle| \geq \Delta \right\} \leq C \exp\left(-\frac{n^2 \Delta^2}{L_1^2} + C p\right) + (np)^C \exp\left(-c \min(\sqrt{p}, n)\right), \quad (82)$$

for any $\Delta \geq L_1$ where $L_1 = C_1 \sqrt{A \exp(-\tau^2/16)(n+p)/n^2}$. The same bound holds for $\mathbb{P}\left\{ \max_{y \in T_p^{1/4}} |\langle \mathbf{y}_{S^c}, \tilde{\mathbf{N}}\mathbf{y}_{S^c} \rangle| \geq \Delta \right\}$.

Proof We first prove the claim for $\langle \mathbf{y}_S, \tilde{\mathbf{N}}\mathbf{y}_S \rangle$. Firstly, we have $\mathbb{E}\langle \mathbf{y}_S, \tilde{\mathbf{N}}\mathbf{y}_S \rangle = 0$. Consider the ‘‘good set’’ \mathcal{G}_2 of pairs $(\mathbf{W}, \mathbf{W}') \in \mathbb{R}^{n \times p} \times \mathbb{R}^{n \times p}$ satisfying the conditions:

$$\|\mathbf{W}\|, \|\mathbf{W}'\| \leq \sqrt{2}(\sqrt{n} + \sqrt{p}), \quad (83)$$

$$\forall i \in [p], \quad \frac{1}{p} \sum_{j \in [p] \setminus i} \mathbb{I}(|\langle \tilde{\mathbf{w}}_i, \tilde{\mathbf{w}}_j \rangle| \geq \tau \sqrt{n}/2) \leq 2 \exp(-\tau^2/16), \quad (84)$$

$$\forall i \in [p], \quad \frac{1}{p} \sum_{j \in [p] \setminus i} \mathbb{I}(|\langle \tilde{\mathbf{w}}'_i, \tilde{\mathbf{w}}'_j \rangle| \geq \tau \sqrt{n}/2) \leq 2 \exp(-\tau^2/16), \quad (85)$$

$$\forall i \in [p], \quad \frac{1}{p} \sum_{j \in [p]} \mathbb{I}(|\langle \tilde{\mathbf{w}}_i, \tilde{\mathbf{w}}_j \rangle| \geq \tau \sqrt{n}/2) \leq 2 \exp(-\tau^2/16). \quad (86)$$

Also, for any pair $\mathbf{W}, \mathbf{W}' \in \mathcal{G}_2$, for $\mathbf{W}(t) = \sqrt{t}\mathbf{W} + \sqrt{1-t}\mathbf{W}'$ (and its columns $\tilde{\mathbf{w}}(t)_i$ defined appropriately) we have:

$$\|\mathbf{W}(t)\| \leq \max(\sqrt{t} + \sqrt{1-t})(\sqrt{2n} + \sqrt{2p}) + \sqrt{2p} = 2(\sqrt{n} + \sqrt{p}), \quad (87)$$

$$\forall i \in [p] \quad \frac{1}{p} \sum_{j \in [p] \setminus i} \mathbb{I}(|\langle \tilde{\mathbf{w}}(t)_i, \tilde{\mathbf{w}}(t)_j \rangle| \geq \tau \sqrt{n}) \leq 6 \exp(-\tau^2/16). \quad (88)$$

Equation (87) follows by a simple application of triangle inequality and condition (83) defining \mathcal{G}_2 . For inequality (88), expanding the product $\langle \tilde{\mathbf{w}}(t)_i, \tilde{\mathbf{w}}(t)_j \rangle$:

$$\langle \tilde{\mathbf{w}}(t)_i, \tilde{\mathbf{w}}(t)_j \rangle = t \langle \tilde{\mathbf{w}}_i, \tilde{\mathbf{w}}_j \rangle + (1-t) \langle \tilde{\mathbf{w}}'_i, \tilde{\mathbf{w}}'_j \rangle + \sqrt{t(1-t)} \langle \tilde{\mathbf{w}}_i, \tilde{\mathbf{w}}'_j \rangle, \quad (89)$$

whence, by triangle inequality and $\sqrt{t(1-t)} < 1$

$$\begin{aligned} \mathbb{I}(|\langle \tilde{\mathbf{w}}(t)_i, \tilde{\mathbf{w}}(t)_j \rangle| \geq \tau \sqrt{n}) &\leq \mathbb{I}(|\langle \tilde{\mathbf{w}}_i, \tilde{\mathbf{w}}_j \rangle| \geq \tau \sqrt{n}/2) + \mathbb{I}(|\langle \tilde{\mathbf{w}}'_i, \tilde{\mathbf{w}}'_j \rangle| \geq \tau \sqrt{n}/2) \\ &+ \mathbb{I}(|\langle \tilde{\mathbf{w}}_i, \tilde{\mathbf{w}}'_j \rangle| \geq \tau \sqrt{n}/2). \end{aligned} \quad (90)$$

The gradient of $\langle \mathbf{y}_S, \eta(\mathbf{W}^T \mathbf{W}/n) \mathbf{y}_S \rangle$ with respect to a column $\tilde{\mathbf{w}}_\ell$ of \mathbf{W} is given by:

$$\nabla_{\tilde{\mathbf{w}}_\ell} \langle \mathbf{y}_S, \eta(\mathbf{W}^T \mathbf{W}/n) \mathbf{y}_S \rangle = \frac{y_\ell}{n} \sum_{j \in S \setminus \ell} y_j \partial \eta\left(\frac{\langle \tilde{\mathbf{w}}_j, \tilde{\mathbf{w}}_\ell \rangle}{n}; \frac{\tau}{\sqrt{n}}\right) \tilde{\mathbf{w}}_j \quad (91)$$

$$= \frac{y_\ell}{n} \mathbf{W} \boldsymbol{\sigma}, \quad (92)$$

$$\text{where } \sigma_i = \begin{cases} \partial \eta(\langle \tilde{\mathbf{w}}_i, \tilde{\mathbf{w}}_\ell \rangle / n; \tau / \sqrt{n}) y_i & \text{when } i \in S \setminus \ell \\ 0 & \text{otherwise.} \end{cases} \quad (93)$$

Therefore

$$\|\nabla_{\tilde{\mathbf{w}}_\ell} \langle \mathbf{y}_S, \tilde{\mathbf{N}}\mathbf{y}_S \rangle\|^2 \leq \frac{y_\ell^2}{n^2} \|\mathbf{W}\|^2 \|\boldsymbol{\sigma}\|^2 \quad (94)$$

$$\leq \frac{y_\ell^2 \|\mathbf{W}\|^2}{n^2} \sum_{i \neq \ell} (y_i \partial \eta(\langle \tilde{\mathbf{w}}_i, \tilde{\mathbf{w}}_\ell \rangle / n))^2 \quad (95)$$

$$\stackrel{(a)}{\leq} \frac{y_\ell^2 \|\mathbf{W}\|^2}{n^2} \sum_{i \neq \ell} \frac{A}{p} \mathbb{I}(|\langle \tilde{\mathbf{w}}_i, \tilde{\mathbf{w}}_\ell \rangle| \geq \tau \sqrt{n}) \quad (96)$$

$$\stackrel{(b)}{\leq} \frac{y_\ell^2}{n^2} C(n+p) A \exp(-\tau^2/16) \quad (97)$$

Here (a) follows from fact that the entries of \mathbf{y}_S are bounded by $\sqrt{A/p}$ and the definition of the soft thresholding function. Inequality (b) follows when we set $\mathbf{W} = \mathbf{Z}(t) = \sqrt{t}\mathbf{Z} + \sqrt{1-t}\mathbf{Z}'$ and $(\mathbf{Z}, \mathbf{Z}') \in \mathcal{G}_2$. Therefore, summing over ℓ we obtain the following bound for the gradient of $\langle \mathbf{y}_S, \tilde{\mathbf{N}}\mathbf{y}_S \rangle$

$$\|\nabla_{\mathbf{Z}(t)} \langle \mathbf{y}_S, \tilde{\mathbf{N}}\mathbf{y}_S \rangle\|^2 \leq C_1 \frac{A \exp(-\tau^2/16)(n+p)}{n^2} \equiv L_1^{-1}. \quad (98)$$

We can use now Lemma 9, to get, for $L_1 > 0$ as defined above and any $\Delta \geq L_1$:

$$\begin{aligned} \mathbb{P}\left\{ \max_{\mathbf{y} \in \mathcal{T}_1^{1/4}} \langle \mathbf{y}_S, \tilde{\mathbf{N}}\mathbf{y}_S \rangle \geq \Delta \right\} &\leq C \exp\left(-\frac{\Delta^2}{CL_1^2} + Cp\right) \\ &\quad + CL_1^{-2} \mathbb{E}\left\{ \max_{\mathbf{y} \in \mathcal{T}_1^{1/4}} \langle \mathbf{y}_S, \tilde{\mathbf{N}}\mathbf{y}_S \rangle^2; \mathcal{G}_2 \right\} \\ &\leq C \exp\left(-\frac{\Delta^2}{CL_1^2} + Cp\right) + C(np)^C \mathbb{P}\{\mathcal{G}_2^c\}^{1/2}, \end{aligned} \quad (100)$$

where the last line follows by Cauchy-Schwarz, as in the proof of Lemma 20, and the fact that $L_1 \geq (np)^{-C_2}$ using the upper bound $\tau \leq \sqrt{\log p/2}$.

To obtain the thesis, we need to now bound $\mathbb{P}\{\mathcal{G}_2^c\}$. It suffices to control the failure probability of conditions (83), (84), (85), (86) of the good set \mathcal{G}_2 individually, and apply the union bound. For \mathbf{Z}, \mathbf{Z}' independent, $\max(\|\mathbf{Z}\|, \|\mathbf{Z}'\|) \geq \sqrt{2}(\sqrt{n} + \sqrt{p})$ with probability at most $2 \exp(-c(n+p))$ by Lemma 11. Now consider condition (84) with $i = 1$, without loss of generality. First, for any $h > 0$ we have:

$$\begin{aligned} \mathbb{P}\left\{ \frac{1}{p} \sum_{j \neq 1} \mathbb{I}(|\langle \tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_j \rangle| \geq \tau \sqrt{n}/2) \geq h \right\} &\leq \mathbb{P}\left\{ \frac{1}{p} \sum_{j \neq 1} \mathbb{I}(|\langle \tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_j \rangle| \geq \tau \sqrt{n}/2) \geq 2h; \|\tilde{\mathbf{z}}_1\| \leq 2\sqrt{n} \right\} \\ &\quad + \mathbb{P}\{\|\tilde{\mathbf{z}}_1\| \geq \sqrt{2n}\}. \end{aligned} \quad (101)$$

Lemma 12 guarantees that the second term is at most $\exp(-cn)$. To control the first term, we note that, conditional on $\tilde{\mathbf{z}}_1, \langle \tilde{\mathbf{z}}_j, \tilde{\mathbf{z}}_1 \rangle, j \neq 1$ are independent Gaussian random variables with variance $\|\tilde{\mathbf{z}}_1\|^2$. Therefore, conditional on $\tilde{\mathbf{z}}_1, \mathbb{I}(|\langle \tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_j \rangle| \geq \tau \sqrt{n}/2)$ are independent Bernoulli random variables with success probability $h_0 = 2\Phi(-\tau \sqrt{n}/(2\|\tilde{\mathbf{z}}_1\|))$, where $\Phi(\cdot)$ is

the Gaussian cumulative distribution function. It follows, by the Chernoff-Hoeffding bound for Bernoulli random variables that

$$\mathbb{P}\left\{ \frac{1}{p} \sum_{j \neq 1} \mathbb{I}(|\langle \tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_j \rangle| \geq \tau \sqrt{n}/2) \geq h \mid \tilde{\mathbf{z}}_1 \right\} \leq \exp(-pD(h\|\tilde{\mathbf{z}}_1\|h_0)), \quad (102)$$

where $D(a\|b) = a \log(a/b) + (1-a) \log(1-a)/(1-b)$. Choosing $h = 4\Phi(-\tau/(2\sqrt{2}))$, and conditional on $\|\tilde{\mathbf{z}}_1\| \leq \sqrt{2n}$, $D(h\|h_0) \geq ch$ for a constant c , implying that

$$\mathbb{P}\left\{ \frac{1}{p} \sum_{j \neq 1} \mathbb{I}(|\langle \tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_j \rangle| \geq \tau \sqrt{n}/2) \geq h; \|\tilde{\mathbf{z}}_1\| \leq \sqrt{2n} \right\} \leq \exp(-cph). \quad (103)$$

By standard bounds $h = 4\Phi(-\tau/(2\sqrt{2})) \leq 2 \exp(-\tau^2/16)$ and, as $\tau \leq \sqrt{\log p/2}$, $h \geq 1/\sqrt{p}$, we have

$$\mathbb{P}\left\{ \frac{1}{p} \sum_{j \neq 1} \mathbb{I}(|\langle \tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_j \rangle| \geq \tau \sqrt{n}/2) \geq h; \|\tilde{\mathbf{z}}_1\| \leq \sqrt{2n} \right\} \leq \exp(-c\sqrt{p}). \quad (104)$$

Combining this with Eq. (101) we now get:

$$\mathbb{P}\left\{ \frac{1}{p} \sum_{j \neq 1} \mathbb{I}(|\langle \tilde{\mathbf{z}}_1, \tilde{\mathbf{z}}_j \rangle| \geq \tau \sqrt{n}/2) \geq h \right\} \leq 2 \exp(-c \min(n, \sqrt{p})). \quad (105)$$

A similar bound holds for $i \neq 1$ and the other conditions (85) and (86), whence we have by the union bound that $\mathbb{P}\{\mathcal{G}_2^c\} \leq p^2 \exp(-c \min(\sqrt{p}, n))$. This completes the proof of the claim (82).

The proof of the claim for $\langle \mathbf{y}_S, \tilde{\mathbf{N}}\mathbf{y}_S \rangle$ is analogous, so we only sketch the points at which it differs from that of Eq. (82). We use the same good set \mathcal{G}_2 , as defined earlier. Computing the gradient as for $\langle \mathbf{y}_S, \tilde{\mathbf{N}}\mathbf{y}_S \rangle$ we obtain:

$$\nabla_{\tilde{\mathbf{w}}_\ell} \langle \mathbf{y}_S, \tilde{\mathbf{N}}\mathbf{y}_S \rangle = \frac{y_\ell}{n} \sum_{j \in \mathcal{S}(\ell)} y_j \tilde{\mathbf{w}}_j \partial \eta\left(\frac{\langle \tilde{\mathbf{w}}_j, \tilde{\mathbf{w}}_\ell \rangle}{n}; \frac{\tau}{\sqrt{n}}\right). \quad (106)$$

Here $\mathcal{S}(\ell) = \mathcal{S}^c$ if $\ell \in \mathcal{S}$ and \mathcal{S} otherwise. Define the vector $\boldsymbol{\sigma}(\ell) \in \mathbb{R}^p$ as

$$(\boldsymbol{\sigma}(\ell))_j = \begin{cases} y_\ell y_j \partial \eta\left(\frac{\langle \tilde{\mathbf{w}}_j, \tilde{\mathbf{w}}_\ell \rangle}{n}; \frac{\tau}{\sqrt{n}}\right) & \text{if } j \in \mathcal{S}(\ell) \\ 0 & \text{otherwise.} \end{cases} \quad (107)$$

As before, we have that $\|\nabla_{\tilde{\mathbf{w}}_\ell} \langle \mathbf{y}_S, \tilde{\mathbf{N}}\mathbf{y}_S \rangle\| = n^{-1} \|\mathbf{W}\boldsymbol{\sigma}(\ell)\| \leq n^{-1} \|\mathbf{W}\| \|\boldsymbol{\sigma}(\ell)\|$. Therefore, summing over $\ell \in [p]$:

$$\|\nabla_{\mathbf{W}} \langle \mathbf{y}_S, \tilde{\mathbf{N}}\mathbf{y}_S \rangle\|^2 \leq \frac{\|\mathbf{W}\|^2}{n^2} \sum_{\ell \in [p]} \|\boldsymbol{\sigma}(\ell)\|^2 \quad (108)$$

$$\leq \frac{\|\mathbf{W}\|^2}{n^2} \sum_{\ell \in [p]} \sum_{j \in \mathcal{S}(\ell)} y_\ell^2 y_j^2 \partial \eta\left(\frac{\langle \tilde{\mathbf{w}}_j, \tilde{\mathbf{w}}_\ell \rangle}{n}; \frac{\tau}{\sqrt{n}}\right) \quad (109)$$

$$= 2 \frac{\|\mathbf{W}\|^2}{n^2} \sum_{\ell \in \mathcal{S}} \sum_{j \in \mathcal{S}^c} y_\ell^2 y_j^2 \partial \eta\left(\frac{\langle \tilde{\mathbf{w}}_j, \tilde{\mathbf{w}}_\ell \rangle}{n}; \frac{\tau}{\sqrt{n}}\right) \quad (110)$$

$$\leq \frac{2\|\mathbf{W}\|^2 A}{n^2} \max_{\ell \in [p]} \sum_{j \neq \ell} \partial \eta\left(\frac{\langle \tilde{\mathbf{w}}_j, \tilde{\mathbf{w}}_\ell \rangle}{n}; \frac{\tau}{\sqrt{n}}\right). \quad (111)$$

Under the condition of \mathcal{G}_2 , the gradient also satisfies, when evaluated at $\mathbf{W} = \mathbf{Z}(t) = \sqrt{t}\mathbf{Z} + \sqrt{1-t}\mathbf{Z}'$:

$$\|\nabla_{\mathbf{Z}(t)} \langle \mathbf{Y}_S, \tilde{\mathbf{N}}\mathbf{Y}_{S^c} \rangle\|^2 \leq \frac{CA \exp(-\tau^2/16)(n+p)}{n^2}. \quad (112)$$

The rest of the proof is then the same as before. \blacksquare

Given these lemmas, we can now establish Proposition 13.

Proof [Proof of Proposition 13] We use a variant of the ε -net argument of Lemma 20. To bound the probability that $\|\tilde{\mathbf{N}}\|_{op}$ is large, with Lemma 8, we obtain:

$$\mathbb{P}\{\|\tilde{\mathbf{N}}\|_{op} \geq \Delta\} \leq \mathbb{P}\left\{\max_{\mathbf{y} \in T_p^S} \langle \mathbf{y}, \tilde{\mathbf{N}}\mathbf{y} \rangle \geq \Delta(1-2\varepsilon)\right\}. \quad (113)$$

Let $\mathbf{S} = \{i : |y_i| \leq \sqrt{A/p}\}$ for some $A \geq 1$ to be chosen later. Then let $\mathbf{y} = \mathbf{y}_S^c + \mathbf{y}_S$ denote the projections of \mathbf{y} onto supports S, S^c respectively. Since $\langle \mathbf{y}, \tilde{\mathbf{N}}\mathbf{y} \rangle = \langle \mathbf{y}_{S^c}, \tilde{\mathbf{N}}\mathbf{y}_{S^c} \rangle + \langle \mathbf{y}_S, \tilde{\mathbf{N}}\mathbf{y}_S \rangle + 2\langle \mathbf{y}_S, \tilde{\mathbf{N}}\mathbf{y}_{S^c} \rangle$ by triangle inequality and union bound:

$$\mathbb{P}\{\|\tilde{\mathbf{N}}\|_{op} \geq \Delta\} \leq \mathbb{P}\left\{\max_{\mathbf{y} \in T_p^S} \langle \mathbf{y}_{S^c}, \tilde{\mathbf{N}}\mathbf{y}_{S^c} \rangle + \langle \mathbf{y}_S, \tilde{\mathbf{N}}\mathbf{y}_S \rangle + 2\langle \mathbf{y}_S, \tilde{\mathbf{N}}\mathbf{y}_{S^c} \rangle \geq \Delta(1-2\varepsilon)\right\} \quad (114)$$

$$\leq \mathbb{P}\left\{\max_{\mathbf{y} \in T_p^S} \langle \mathbf{y}_{S^c}, \tilde{\mathbf{N}}\mathbf{y}_{S^c} \rangle \geq \Delta(1-2\varepsilon)/4\right\} + \mathbb{P}\left\{\max_{\mathbf{y} \in T_p^S} \langle \mathbf{y}_S, \tilde{\mathbf{N}}\mathbf{y}_S \rangle \geq \Delta(1-2\varepsilon)/4\right\} \\ + \mathbb{P}\left\{\max_{\mathbf{y} \in T_p^S} \langle \mathbf{y}_S, \tilde{\mathbf{N}}\mathbf{y}_{S^c} \rangle \geq \Delta(1-2\varepsilon)/4\right\} \quad (115)$$

$$\leq \mathbb{P}\left\{\max_{S: |S| \leq p/4} \|\mathcal{P}_{S^c \times S}(\tilde{\mathbf{N}})\| \geq \Delta(1-2\varepsilon)/4\right\} + \mathbb{P}\left\{\max_{\mathbf{y} \in T_p^S} \langle \mathbf{y}_S, \tilde{\mathbf{N}}\mathbf{y}_S \rangle \geq \Delta(1-2\varepsilon)/4\right\} \\ + \mathbb{P}\left\{\max_{\mathbf{y} \in T_p^S} \langle \mathbf{y}_S, \tilde{\mathbf{N}}\mathbf{y}_{S^c} \rangle \geq \Delta(1-2\varepsilon)/4\right\}. \quad (116)$$

With $\varepsilon = 1/4$, the first term is controlled by Lemma 20 while the final two are controlled by Lemma 21. We choose $\varepsilon = 1/4$ in Eq. (116), and

$$\Delta = \Delta_* \equiv C \sqrt{\frac{p}{n} \left(1 + \frac{p}{n}\right) \left(\frac{\log A}{A} + A \exp\left(-\frac{\tau^2}{16}\right)\right)}, \quad (117)$$

for large enough C so that, using the bounds of Lemmas 20 and 21, we have:

$$\mathbb{P}\{\tilde{\mathbf{N}} \geq \Delta_*\} \leq C(np)^C \exp\left[-c \min\left(\sqrt{p}, n, p \frac{\log A}{A}\right)\right]. \quad (118)$$

This probability bound is $o(1)$ provided A is not too large: we choose $A = 0.25\sqrt{\tau \exp(\tau^2/16)} \ll \sqrt{p}$ which guarantees that the bound above is $o(1)$ when $n > C \log p$ for some C large enough. This concludes the proposition. \blacksquare

6.2 Proof of Proposition 14

We decompose the empirical covariance matrix (53) as

$$\mathcal{P}_E(\hat{\Sigma}) = \Sigma + \Delta_1 + \Delta_2 + \Delta_2^\top + \mathcal{P}_E\left(\frac{1}{n}\mathbf{Z}^\top\mathbf{Z} - \mathbf{I}_p\right), \quad (119)$$

$$\Delta_1 \equiv \sum_{q,q'=1}^r \sqrt{\beta_q \beta_{q'}} \left(\frac{1}{n} \langle \mathbf{u}_q, \mathbf{u}_{q'} \rangle - \mathbf{1}_{q=q'}\right) \mathbf{v}_q \mathbf{v}_{q'}^\top, \quad (120)$$

$$\Delta_2 \equiv \sum_{q=1}^r \frac{\sqrt{\beta_q}}{n} \mathbf{v}_q (\mathbf{Z}^\top \mathbf{u}_q)^\top. \quad (121)$$

Next notice that, for any $x \in \mathbb{R}$,

$$|\eta(x) - x| \leq \frac{\tau}{\sqrt{n}}. \quad (122)$$

With a view to employing this inequality, we use Eq. (119) and the triangle inequality:

$$\|\mathcal{P}_E(\eta(\hat{\Sigma})) - \Sigma\|_{op} = \left\| \mathcal{P}_E(\eta(\hat{\Sigma})) - \mathcal{P}_E(\hat{\Sigma}) - \Delta_1 - \Delta_2 - \Delta_2^\top - \mathcal{P}_E\left(\frac{1}{n}\mathbf{Z}^\top\mathbf{Z} - \mathbf{I}_p\right) \right\|_{op} \quad (123)$$

$$\leq \|\mathcal{P}_E(\eta(\hat{\Sigma}) - \hat{\Sigma})\|_{op} + \|\Delta_1\|_{op} + 2\|\Delta_2\|_{op} + \left\| \mathcal{P}_E\left(\frac{1}{n}\mathbf{Z}^\top\mathbf{Z} - \mathbf{I}_p\right) \right\|_{op} \quad (124)$$

$$\leq \frac{80\tau}{\sqrt{n}} + \|\Delta_1\|_{op} + 2\|\Delta_2\|_{op} + \left\| \mathcal{P}_E\left(\frac{1}{n}\mathbf{Z}^\top\mathbf{Z} - \mathbf{I}_p\right) \right\|_{op}, \quad (125)$$

where the last line follows by noticing that the first term is supported on \mathbf{E} of size 80×80 and then using bias bound Eq. (122) entry-wise. We next bound each of the three terms on the right hand side.

For the first term in Eq (125), note that with a change of basis to the orthonormal set $\mathbf{v}_1, \dots, \mathbf{v}_r$, Δ_1 is equivalent to an $r \times r$ matrix with entries $M_{qq'} \sqrt{\beta_q \beta_{q'}}$, where $M_{qq'} = \langle \mathbf{u}_q, \mathbf{u}_{q'} \rangle / n - \mathbf{1}_{q=q'}$. Denote by $\mathbf{B} \in \mathbb{R}^{r \times r}$ the diagonal matrix with $B_{qq} = \sqrt{\beta_q}$ and by $\mathbf{U} \in \mathbb{R}^{r \times n}$, the matrix with columns $\mathbf{u}_1, \dots, \mathbf{u}_r$. Then, we have, with high probability

$$\|\Delta_1\|_{op} = \|\mathbf{B}\mathbf{M}\mathbf{B}\|_{op} \quad (126)$$

$$\leq \|\mathbf{B}\|_{op}^2 \|\mathbf{M}\|_{op} = \beta \left\| \frac{1}{n} \mathbf{U}^\top \mathbf{U} - \mathbf{I}_{r \times r} \right\|_{op} \quad (127)$$

$$\leq C\beta \sqrt{\frac{\tau}{n}}. \quad (128)$$

The last inequality follows from the Bat-Yin law on eigenvalues of Wishart matrices (see Vershynin, 2012, Corollary 5.35).

Consider the second term in Eq (125). By orthonormality of $\mathbf{v}_1, \dots, \mathbf{v}_r$, the matrix Δ_2 is orthogonally equivalent to $\mathbf{B}\mathbf{Z}_Q^\top \mathbf{U} / n$, where we recall that \mathbf{Z}_Q denotes the submatrix of \mathbf{Z} formed by the columns in Q . Denoting by \mathbf{P}_U the orthogonal projector onto the column

space of \mathbf{U} , we then have, with high probability,

$$\|\Delta_2\|_{op} \leq \frac{1}{n} \|\mathbf{B}\|_{op} \|\mathbf{Z}_Q^T \mathbf{P}_U \mathbf{U}\|_{op} \quad (129)$$

$$\leq \frac{\beta}{n} \|\mathbf{P}_U \mathbf{Z}_Q\|_{op} \|\mathbf{U}\|_{op} \quad (130)$$

$$\leq \frac{C\beta}{n} (\sqrt{s_0} + \sqrt{r}) (\sqrt{n} + \sqrt{r}) \leq C\beta \sqrt{\frac{s_0}{n}}. \quad (131)$$

Here the penultimate inequality follows by Lemma 11 noting that, by invariance under rotations (and since \mathbf{P}_U project onto a random subspace of r dimensions independent of \mathbf{Z}), $\|\mathbf{P}_U \mathbf{Z}_Q\|_{op}$ is distributed as the norm of a matrix with i.i.d. standard normal entries, with dimensions $|\mathbf{Q}| \times r$, $|\mathbf{Q}| \leq s_0$.

Finally, for the third term of Eq. (125) we use the Bai-Yin law of Wishart matrices (see Vershynin, 2012, Corollary 5.35) to obtain, with high probability:

$$\left\| \mathcal{P}_E \left(\frac{1}{n} \mathbf{Z}^T \mathbf{Z} - \mathbf{I}_p \right) \right\|_{op} = \left\| \frac{1}{n} \mathbf{Z}_Q^T \mathbf{Z}_Q - \mathbf{I}_{s_0} \right\|_{op} \quad (132)$$

$$\leq C \sqrt{\frac{s_0}{n}}. \quad (133)$$

Finally, substituting the above bounds in Eq. (125), we get

$$\left\| \mathcal{P}_E(n\widehat{\Sigma}) - \Sigma \right\|_{op} = \frac{\tau s_0}{\sqrt{n}} + C(1 + \beta) \sqrt{\frac{s_0}{n}}, \quad (134)$$

which implies the proposition.

6.3 Proof of Proposition 15

Note that $\mathbf{C} = \widehat{\mathbf{C}} + \mathbf{C}^T$ where $\widehat{\mathbf{C}} = \mathcal{P}_{\mathbf{Q} \times \mathbf{Q}^c}(n\widehat{\Sigma})$. It is therefore sufficient to control $\widehat{\mathbf{C}}$, and then use triangle inequality. The proof is similar to that of Proposition 13. We let $\mathbf{U} \in \mathbb{R}^{n \times r}$ denote the matrix with columns $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_r$, and introduce the set

$$\mathcal{U} \equiv \left\{ \mathbf{U} \in \mathbb{R}^{n \times r} : \left\| \frac{1}{n} \mathbf{U}^T \mathbf{U} - \mathbf{I}_{r \times r} \right\|_{op} \leq 5 \sqrt{\frac{r}{n}} \right\}. \quad (135)$$

We then have

$$\mathbb{P}(\|\widehat{\mathbf{C}}\|_{op} \geq \Delta) \leq \sup_{\mathbf{U} \in \mathcal{U}} \mathbb{P}(\|\widehat{\mathbf{C}}\|_{op} \geq \Delta | \mathbf{U}) + \mathbb{P}(\mathbf{U} \notin \mathcal{U}). \quad (136)$$

Notice that, by the Bai-Yin law on eigenvalues of Wishart matrices (see Vershynin, 2012, Corollary 5.35), $\lim_{n \rightarrow \infty} \mathbb{P}(\mathbf{U} \in \mathcal{U}) = 1$ (throughout $r < cn$ for c a small constant). It is therefore sufficient to show $\sup_{\mathbf{U} \in \mathcal{U}} \mathbb{P}(\|\widehat{\mathbf{C}}\|_{op} \geq \Delta | \mathbf{U}) \rightarrow 0$ for Δ as in the statement of the theorem.

In order to lighten the notation, we will write $\widehat{\mathbb{P}}(\cdot) \equiv \mathbb{P}(\cdot | \mathbf{U})$ and bound the above probability uniformly over $\mathbf{U} \in \mathcal{U}$. (In other words $\widehat{\mathbb{P}}$ denotes expectation over \mathbf{Z} with \mathbf{U} fixed). We first control the norms of small submatrices of $\widehat{\mathbf{C}}$, following which we control the full matrix.

Lemma 22 Fix an $A \in [1, p^{1/3}]$, and let $L = \sqrt{((\beta \vee 1)n + p)/n^2}$. Then, there exists an absolute constant $C > 0$ such that, for any $\Delta > 0$:

$$\begin{aligned} \widehat{\mathbb{P}} \left\{ \max_{\mathbf{Q}^c \ni \mathbf{S}: |\mathbf{S}| \leq p/A} \|\mathcal{P}_{\mathbf{Q} \times \mathbf{S}}(n\widehat{\Sigma})\|_{op} \geq \Delta \right\} &\leq C \exp \left(C s_0 + \frac{p \log(CA)}{A} - \frac{\Delta^2}{CL^2} \right) \\ &\quad + L^{-2} (np)^C \exp(-n/C). \end{aligned} \quad (137)$$

Proof Let, as before, $T_p^{\varepsilon}(\mathbf{S})$ denote the ε -net of unit vectors supported on $\mathbf{S} \subset \mathbf{Q}^c$ of size at most p/A and let $T = \cup_{\mathbf{S} \in T_p^{\varepsilon}(\mathbf{S})}$. Then, by Lemma 8, with $\varepsilon = 1/4$:

$$\widehat{\mathbb{P}} \left\{ \max_{\mathbf{S} \subseteq \mathbf{Q}^c: |\mathbf{S}| \leq p/A} \|\mathcal{P}_{\mathbf{Q} \times \mathbf{S}}(n\widehat{\Sigma})\|_{op} \geq \Delta \right\} \leq \widehat{\mathbb{P}} \left\{ \max_{y \in T, \mathbf{w} \in T_p^{\varepsilon}} \langle \mathbf{w}, \widehat{\mathbf{C}} \mathbf{y} \rangle \geq \Delta(1 - 2\varepsilon)/2 \right\}. \quad (138)$$

It now suffices to control the right hand side via Lemma 9. We first compute the gradients with respect to $\bar{\mathbf{z}}_{\ell}$ as before:

$$\nabla_{\bar{\mathbf{z}}_{\ell}} \langle \mathbf{w}, \widehat{\mathbf{C}} \mathbf{y} \rangle = \begin{cases} \frac{w_{\ell}}{n} \sum_{i \in \mathbf{Q}^c} y_i \partial \eta(\langle \bar{\mathbf{x}}_{\ell}, \bar{\mathbf{z}}_i \rangle / n) \bar{\mathbf{z}}_i & \text{when } \ell \in \mathbf{Q}, \\ \frac{w_{\ell}}{n} \sum_{i \in \mathbf{Q}} w_i \partial \eta(\langle \bar{\mathbf{z}}_{\ell}, \bar{\mathbf{x}}_i \rangle / n) \bar{\mathbf{x}}_i & \text{when } \ell \in \mathbf{Q}^c, \end{cases} \quad (139)$$

Therefore, arguing as in proof of Proposition 13 (see Lemma 20):

$$\|\nabla_{\mathbf{Z}} \langle \mathbf{w}, \widehat{\mathbf{C}} \mathbf{y} \rangle\|_F^2 = \sum_{\ell} \|\nabla_{\bar{\mathbf{z}}_{\ell}} \langle \mathbf{w}, \widehat{\mathbf{C}} \mathbf{y} \rangle\|^2 \leq \frac{\|\mathbf{Z}\|^2 + \|\mathbf{X}_Q\|^2}{n^2}. \quad (140)$$

Let $\mathbf{B} \in \mathbb{R}^{r \times r}$ be the diagonal matrix with entries $B_{q,q} = \sqrt{\beta_{q,r}}$ and $\mathbf{V} \in \mathbb{R}^{p \times r}$ be the matrix with columns $\mathbf{v}_1, \dots, \mathbf{v}_r$. We then have $\mathbf{X} = \mathbf{UBV}^T + \mathbf{Z}$, whence, recalling $\mathbf{U} \in \mathcal{U}$, and $r \leq cn$ with c small enough

$$\|\mathbf{X}_Q\| \leq \|\mathbf{X}\| \leq \|\mathbf{UBV}^T\| + \|\mathbf{Z}\| \quad (141)$$

$$\leq \sqrt{\beta} \|\mathbf{U}\| + \|\mathbf{Z}\| \leq 5\sqrt{\beta}n + \|\mathbf{Z}\|. \quad (142)$$

Consider the good set \mathcal{G}_4 of pairs $(\mathbf{Z}, \mathbf{Z}')$ satisfying:

$$\max(\|\mathbf{Z}\|, \|\mathbf{Z}'\|) \leq \sqrt{2n} + \sqrt{2p}, \quad (143)$$

$$\max(\|\mathbf{Z}_Q\|, \|\mathbf{Z}'_Q\|) \leq \sqrt{2n} + \sqrt{2k}. \quad (144)$$

For $(\mathbf{Z}, \mathbf{Z}') \in \mathcal{G}_4$, and $t \in [0, 1]$, define $\mathbf{Z}(t) = \sqrt{t}\mathbf{Z} + \sqrt{1-t}\mathbf{Z}'$. Now Using Eqs. (140) and (142), the gradient $\nabla \langle \mathbf{w}, \widehat{\mathbf{C}} \mathbf{y} \rangle$ evaluated at $\mathbf{Z}(t)$ satisfies:

$$\|\nabla \langle \mathbf{w}, \widehat{\mathbf{C}} \mathbf{y} \rangle\|^2 \leq \frac{3\|\mathbf{Z}(t)\|^2 + 10\beta n}{n^2} \quad (145)$$

$$\leq C \frac{(n+p) + \beta n}{n^2} \quad (146)$$

$$\leq C \frac{(\beta \vee 1)n + p}{n^2}. \quad (147)$$

Now applying Corollary 10, for $L = C\sqrt{((\beta \vee 1)n + p)/n^2}$:

$$\begin{aligned} \mathbb{P}^c \left\{ \max_{S \subseteq \mathcal{Q}^c, |S| \leq p/A} \|\mathbf{P}_{\mathcal{Q} \times S}(\eta(\hat{\Sigma}))\|_{op} \geq \Delta \right\} &\leq C|T| \exp\left(-\frac{\Delta^2}{CL^2}\right) \\ &\quad + CL^{-2} \mathbb{E} \left[\max_{\mathbf{w}, \mathbf{y}} \langle \mathbf{w}, \bar{\mathbf{C}}\mathbf{y} \rangle^4 \right]^{1/4} \mathbb{P}\{G_A\}^{1/2}. \end{aligned} \quad (148)$$

Let $\varepsilon = 1/4$, observing that $T \subseteq \cup_{S: |S| \leq p/A} \mathcal{T}_p^\varepsilon(S)$, we have the bound (using Lemma 7 and Stirling's approximation):

$$|T| \leq \exp(Cs_0 + A^{-1}p \log CA), \quad (149)$$

for some absolute C . Now, as in the proof of Proposition 13, $|\langle \mathbf{w}, \bar{\mathbf{C}}\mathbf{y} \rangle| \leq \|\mathbf{C}\| \leq \|\mathbf{C}\|_F \leq \|\hat{\Sigma}\|_F$. From this it follows that $\mathbb{E} \left[\max_{\mathbf{w}, \mathbf{y}} \langle \mathbf{w}, \bar{\mathbf{C}}\mathbf{y} \rangle^4 \right] \leq (np)^C$ for some C . Finally $\mathbb{P}\{G_A\} \leq \exp(-cn)$ using Lemmas 11, 12 and the union bound. Combining these bounds in Eq. (148) yields the lemma. \blacksquare

Now we prove a similar lemma when \mathbf{y} has entries that are ‘‘spread out’’.

Lemma 23 Fix an $A \in [1, p^{1/3}]$, and a unit vector $\mathbf{y} \in \mathbb{R}^{\mathcal{Q}^c}$ let $\mathbf{S} = \{i : |y_i| \leq \sqrt{A}/p\}$, and \mathbf{y}_S denote the projection of \mathbf{y} on the set of indices S . Then there exists a numerical constant C such that, assuming $\tau \leq \sqrt{\log p}/2$, we have

$$\mathbb{P}^c \left\{ \max_{\mathbf{w} \in \mathcal{T}_{\mathcal{Q}^c}^c} \langle \mathbf{w}, \bar{\mathbf{C}}\mathbf{y}_S \rangle \geq \Delta \right\} \leq C \exp\left(-\frac{\Delta^2}{CL_*^2} + Cp\right) + (np)^C \exp(-c \min(\sqrt{p}, n)), \quad (150)$$

where $L_* = \sqrt{A \exp(-\tau^2/C(\beta \vee 1))(n(\beta \vee 1) + p)/n^2}$.

Proof For simplicity of notation, it is convenient to introduce the vector $\mathbf{y}' = \mathbf{y}_S$. Throughout the proof, we will use that $\|\mathbf{y}'\| \leq 1$ and $\|\mathbf{y}'\|_\infty \leq \sqrt{A}/p$. We compute the gradients as follows:

$$\nabla_{z_\ell} \langle \mathbf{w}, \bar{\mathbf{C}}\mathbf{y}' \rangle = \begin{cases} \frac{w_\ell}{n} \sum_{i \in \mathcal{Q}^c} y'_i \partial \eta(\langle \tilde{\mathbf{x}}_i, \tilde{\mathbf{z}}_i \rangle / n) \tilde{\mathbf{z}}_i & \text{when } \ell \in \mathcal{Q} \\ \frac{w_\ell}{n} \sum_{i \in \mathcal{Q}^c} w_i \partial \eta(\langle \tilde{\mathbf{z}}_i, \tilde{\mathbf{x}}_i \rangle / n) \tilde{\mathbf{x}}_i & \text{when } \ell \in \mathcal{Q}^c. \end{cases} \quad (151)$$

Therefore we have

$$\sum_{\ell \in \mathcal{Q}} \|\nabla_{z_\ell} \langle \mathbf{w}, \bar{\mathbf{C}}\mathbf{y}' \rangle\|^2 \leq \sum_{\ell \in \mathcal{Q}} \frac{w_\ell^2}{n^2} \|\mathbf{Z}\|^2 \sum_{i \in \mathcal{Q}^c} (y'_i \partial \eta(\langle \tilde{\mathbf{x}}_i, \tilde{\mathbf{z}}_i \rangle))^2 \quad (152)$$

$$\leq \frac{A \|\mathbf{Z}\|^2}{pn^2} \max_{\ell \in \mathcal{Q}} \sum_{i \in \mathcal{Q}^c} \partial \eta(\langle \tilde{\mathbf{x}}_i, \tilde{\mathbf{z}}_i \rangle / n), \quad (153)$$

where we used the fact that $|y'_i| \leq \sqrt{A}/p$ and that $\partial \eta(\cdot) \in \{0, 1\}$. Similarly, for $\ell \in \mathcal{Q}^c$:

$$\sum_{\ell \in \mathcal{Q}^c} \|\nabla_{z_\ell} \langle \mathbf{w}, \bar{\mathbf{C}}\mathbf{y}' \rangle\|^2 \leq \sum_{\ell \in \mathcal{Q}^c} \frac{(y'_\ell)^2 \|\mathbf{X}_{\mathcal{Q}}\|^2}{n^2} \sum_{i \in \mathcal{Q}} (w_i \partial \eta(\langle \tilde{\mathbf{z}}_i, \tilde{\mathbf{x}}_i \rangle / n))^2 \quad (154)$$

$$= \sum_{i \in \mathcal{Q}} \frac{w_i^2 \|\mathbf{X}_{\mathcal{Q}}\|^2}{n^2} \sum_{\ell \in \mathcal{Q}^c} (y'_\ell)^2 \partial \eta(\langle \tilde{\mathbf{z}}_i, \tilde{\mathbf{x}}_i \rangle / n)^2 \quad (155)$$

$$\leq \frac{A \|\mathbf{X}_{\mathcal{Q}}\|^2}{pn^2} \max_{\ell \in \mathcal{Q}} \sum_{i \in \mathcal{Q}^c} \partial \eta(\langle \tilde{\mathbf{z}}_i, \tilde{\mathbf{x}}_i \rangle / n), \quad (156)$$

Combining the bounds in Eqs. (153), (156), we obtain

$$\|\nabla_{\mathbf{z}} \langle \mathbf{w}, \bar{\mathbf{C}}\mathbf{y}' \rangle\|_F^2 = \sum_{\ell \in [p]} \|\nabla_{z_\ell} \langle \mathbf{w}, \bar{\mathbf{C}}\mathbf{y}' \rangle\|^2 \quad (157)$$

$$\leq \frac{2A}{pn^2} (\|\mathbf{X}_{\mathcal{Q}}\|^2 + \|\mathbf{Z}\|^2) \max_{i \in \mathcal{Q}} \sum_{j \in \mathcal{Q}^c} \partial \eta(\langle \tilde{\mathbf{x}}_i, \tilde{\mathbf{z}}_j \rangle / n). \quad (158)$$

With $K = C\beta \vee 1$, we define the good set \mathcal{G}_5 of pairs $(\mathbf{Z}, \mathbf{Z}')$ satisfying

$$\|\mathbf{Z}\|, \|\mathbf{Z}'\| \leq \sqrt{2n} + \sqrt{2p} \quad (159)$$

$$\forall i \in \mathcal{Q}, \quad \frac{1}{p} \sum_{j \in \mathcal{Q}^c} \mathbb{I}(\langle \tilde{\mathbf{x}}_i, \tilde{\mathbf{z}}_j \rangle \geq \tau\sqrt{n}/2) \leq 2 \exp(-\tau^2/K) \quad (160)$$

$$\forall i \in \mathcal{Q}, \quad \frac{1}{p} \sum_{j \in \mathcal{Q}^c} \mathbb{I}(\langle \tilde{\mathbf{x}}'_i, \tilde{\mathbf{z}}'_j \rangle \geq \tau\sqrt{n}/2) \leq 2 \exp(-\tau^2/K) \quad (161)$$

$$\forall i \in \mathcal{Q}, \quad \frac{1}{p} \sum_{j \in \mathcal{Q}^c} \mathbb{I}(\langle \tilde{\mathbf{x}}'_i, \tilde{\mathbf{z}}_j \rangle \geq \tau\sqrt{n}/4) \leq 2 \exp(-\tau^2/K) \quad (162)$$

$$\forall i \in \mathcal{Q}, \quad \frac{1}{p} \sum_{j \in \mathcal{Q}^c} \mathbb{I}(\langle \tilde{\mathbf{x}}_i, \tilde{\mathbf{z}}'_j \rangle \geq \tau\sqrt{n}/4) \leq 2 \exp(-\tau^2/K). \quad (163)$$

Define $\mathbf{Z}(t) = \sqrt{t}\mathbf{Z} + \sqrt{1-t}\mathbf{Z}'$ with $(\mathbf{Z}, \mathbf{Z}') \in \mathcal{G}_5$. By Eq. (158) the gradient evaluated at $\mathbf{Z}(t)$ is bounded by

$$\|\nabla \langle \mathbf{w}, \bar{\mathbf{C}}\mathbf{y} \rangle\|^2 \leq \frac{2A}{pn^2} (\|\mathbf{X}_{\mathcal{Q}}(t)\|^2 + \|\mathbf{Z}(t)\|^2) \max_{i \in \mathcal{Q}} \sum_{j \in \mathcal{Q}^c} \partial \eta(\langle \tilde{\mathbf{x}}(t)_i, \tilde{\mathbf{z}}(t)_j \rangle / n) \quad (164)$$

$$\leq \frac{CA}{pn^2} ((\beta \vee 1)n + p) \max_{i \in \mathcal{Q}} \sum_{j \in \mathcal{Q}^c} \partial \eta(\langle \tilde{\mathbf{x}}(t)_i, \tilde{\mathbf{z}}(t)_j \rangle / n), \quad (165)$$

where we bounded $\|\mathbf{X}_{\mathcal{Q}}(t)\|$ as in Eq. (142), and used $\|\mathbf{Z}(t)\|_{\text{op}} \leq 2(\sqrt{n} + \sqrt{p})$, which follows from Eq. (159) and triangle inequality. Furthermore, as $\langle \tilde{\mathbf{x}}(t)_i, \tilde{\mathbf{z}}(t)_j \rangle = t \langle \tilde{\mathbf{x}}_i, \tilde{\mathbf{z}}_j \rangle + (1-t) \langle \tilde{\mathbf{x}}'_i, \tilde{\mathbf{z}}'_j \rangle + \sqrt{t(1-t)} \langle \langle \tilde{\mathbf{x}}_i, \tilde{\mathbf{z}}'_j \rangle + \langle \tilde{\mathbf{x}}'_i, \tilde{\mathbf{z}}_j \rangle \rangle$, we have that:

$$\partial \eta(\langle \tilde{\mathbf{x}}(t)_i, \tilde{\mathbf{z}}(t)_j \rangle / n) = \mathbb{I}(\langle \tilde{\mathbf{x}}(t)_i, \tilde{\mathbf{z}}(t)_j \rangle \geq \tau\sqrt{n}) \quad (166)$$

$$\begin{aligned} &\leq \mathbb{I}(\langle \tilde{\mathbf{x}}_i, \tilde{\mathbf{z}}_j \rangle \geq \tau\sqrt{n}/2) + \mathbb{I}(\langle \tilde{\mathbf{x}}'_i, \tilde{\mathbf{z}}'_j \rangle \geq \tau\sqrt{n}/2) \\ &\quad + \mathbb{I}(\langle \tilde{\mathbf{x}}_i, \tilde{\mathbf{z}}_j \rangle \geq \tau\sqrt{n}/4) + \mathbb{I}(\langle \tilde{\mathbf{x}}'_i, \tilde{\mathbf{z}}'_j \rangle \geq \tau\sqrt{n}/4). \end{aligned} \quad (167)$$

Hence on the good set \mathcal{G}_5 , we have:

$$\max_{i \in \mathbf{Q}} \sum_{j \in \mathbf{Q}^c} \partial \eta_i(\tilde{\mathbf{x}}(t)_i, \tilde{\mathbf{z}}(t)_j/n) \leq 4p e^{-\tau^2/K}. \quad (168)$$

Therefore the gradient satisfies, on the good set:

$$\|\nabla_{\mathbf{Z}} \langle \mathbf{w}, \bar{\mathbf{C}}\mathbf{y}' \rangle\|^2 \leq C \frac{A}{n^2} ((\beta \vee 1)n + p) e^{-\tau^2/K} = CL_*^2. \quad (169)$$

Hence, by Lemma 9, we obtain:

$$\begin{aligned} \tilde{\mathbb{P}} \left\{ \max_{\mathbf{w} \in T_{\mathbf{Q}}^{\varepsilon}, \mathbf{y} \in T_{\mathbf{Q}^c}^{\varepsilon}} \langle \mathbf{w}, \bar{\mathbf{C}}\mathbf{y}' \rangle \geq \Delta \right\} &\leq C |T_{\mathbf{Q}}^{\varepsilon}| |T_{\mathbf{Q}^c}^{\varepsilon}| \exp\left(-\frac{\Delta^2}{CL_*^2}\right) \\ &+ CL_*^{-2} \tilde{\mathbb{P}} \left\{ \max_{\mathbf{w}, \mathbf{y}} \langle \mathbf{w}, \bar{\mathbf{C}}\mathbf{y}' \rangle^4 \right\}^{1/4} \mathbb{P}\{\mathcal{G}_5^c\}^{1/2}. \end{aligned} \quad (170)$$

By Lemma 7, keeping $\varepsilon = 1/4$ we have that the first term is at most $C \exp(Cp + \exp(-\Delta^2/CL_*^2))$. For the second term, we have $\langle \mathbf{w}, \bar{\mathbf{C}}\mathbf{y}' \rangle \leq \|\bar{\mathbf{C}}\|_F \leq \|\bar{\mathbf{C}}\|_F \leq \|\bar{\mathbf{C}}\|_F \leq \|\bar{\mathbf{C}}\|_F \leq \|\bar{\mathbf{C}}\|_F \leq (np)^C$, we have that $\mathbb{E}\{\max_{\mathbf{w}, \mathbf{y}} \langle \mathbf{w}, \bar{\mathbf{C}}\mathbf{y}' \rangle^4\}^{1/4} \leq (np)^C$. Also as $\tau < \sqrt{\log p}$, $L_* \geq (np)^{-C}$, implying that the second term is bounded above by $(np)^C \mathbb{P}\{\mathcal{G}_5^c\}^{1/2}$. Therefore:

$$\tilde{\mathbb{P}} \left\{ \max_{\mathbf{w} \in T_{\mathbf{Q}}^{\varepsilon}, \mathbf{y} \in T_{\mathbf{Q}^c}^{\varepsilon}} \langle \mathbf{w}, \bar{\mathbf{C}}\mathbf{y}' \rangle \geq \Delta \right\} \leq C \exp\left(Cp - \frac{\Delta^2}{CL_*^2}\right) + (np)^C \mathbb{P}\{\mathcal{G}_5^c\}^{1/2}. \quad (171)$$

It remains to control the probability of the bad set \mathcal{G}_5^c . For this, we control the probability of violating any one condition among (159), (160), (161), (162) and (163) defining \mathcal{G}_5 and then use the union bound. By Lemmas 11, condition (159) hold with probability $1 - C \exp(-cn)$. The argument controlling the probability for conditions (160), (161), (162) and (163) to hold are essentially the same, so we restrict ourselves to condition (160) keeping $i = 1 \in \mathbf{Q}$, without loss of generality. Conditional on $\tilde{\mathbf{x}}_1, (\tilde{\mathbf{x}}_1, \tilde{\mathbf{z}}_j)$ for $j \in \mathbf{Q}^c$ are independent $\mathcal{N}(0, \|\tilde{\mathbf{x}}_1\|^2)$ variables. Therefore, conditional on $\tilde{\mathbf{x}}_1$, $\mathbb{I}(\|\tilde{\mathbf{x}}_1, \tilde{\mathbf{z}}_j\| \geq \tau\sqrt{n}/2)$ are independent Bernoulli random variables with success probability $\Phi\{-\tau\sqrt{n}/2\|\tilde{\mathbf{x}}_1\|\}$. Define h_1 to be the success probability, i.e. $h_1 = \Phi(-\tau\sqrt{n}/(2\|\tilde{\mathbf{x}}_1\|))$.

Since $K = C(\beta \vee 1)$ we can enlarge C to a large absolute constant. Letting $\mathbf{V} \in \mathbb{R}^{n \times r}$ be the matrix with columns $\mathbf{v}_1, \dots, \mathbf{v}_r$, and \mathbf{B} the diagonal matrix with $B_{q,q} = \sqrt{\beta}_q$, we have, with probability at least $1 - \exp(-n/C)$,

$$\|\tilde{\mathbf{x}}_1\| \leq \|\mathbf{UBV}^T \mathbf{e}_1\| + \|\tilde{\mathbf{z}}_1\| \leq \|\mathbf{B}\| \|\mathbf{U}\| + \|\tilde{\mathbf{z}}_1\| \leq \sqrt{\frac{Kn}{4}}, \quad (172)$$

where the last equality holds since $\mathbf{U} \in \mathcal{U}$ and by tail bounds on chi-squared random variables. Further

$$\begin{aligned} \tilde{\mathbb{P}} \left\{ \sum_{j \in \mathbf{Q}^c} \mathbb{I}(\|\tilde{\mathbf{x}}_1, \tilde{\mathbf{z}}_j\| \geq \tau\sqrt{n}/2) \geq |\mathbf{Q}^c| h \right\} &\leq \tilde{\mathbb{P}} \left\{ \|\tilde{\mathbf{x}}_1\| \geq Kn \right\} \\ &+ \sup_{\|\tilde{\mathbf{x}}_1\|^2 \leq Kn} \tilde{\mathbb{P}} \left\{ \sum_{j \in \mathbf{Q}^c} \mathbb{I}(\|\tilde{\mathbf{x}}_1, \tilde{\mathbf{z}}_j\| \geq \tau\sqrt{n}/2) \geq |\mathbf{Q}^c| h \mid \tilde{\mathbf{x}}_1 \right\}. \end{aligned} \quad (173)$$

By the above argument, the first term is at most $\exp(-n/C)$ and we turn to the second term. By the Chernoff bound

$$\tilde{\mathbb{P}} \left\{ \sum_{j \in \mathbf{Q}^c} \mathbb{I}(\|\tilde{\mathbf{x}}_1, \tilde{\mathbf{z}}_j\| \geq \tau\sqrt{n}/2) \geq |\mathbf{Q}^c| h \mid \tilde{\mathbf{x}}_1 \right\} \leq \exp(-|\mathbf{Q}^c| D(h\|\tilde{\mathbf{x}}_1)), \quad (174)$$

with $h_1 < \exp(-\tau^2/K)$ when $\|\tilde{\mathbf{x}}_1\|^2 \leq Kn/4$. Choosing $h = 2 \exp(-\tau^2/K)$ implies that $h_1 \leq h/2$ when and, thereby, that $D(h\|\tilde{\mathbf{x}}_1) \geq h/C$. Further since $\tau < \sqrt{\log p}/2$, $h \geq 1/\sqrt{p}$. This implies that

$$\exp(-|\mathbf{Q}^c| D(h - h_1\|\tilde{\mathbf{x}}_1)) = \exp(-(p - s_0)h/C) \geq \exp(-\sqrt{p}/C). \quad (175)$$

Combining this with Eq. (173) we have that $\mathbb{P}\{\mathcal{G}_5^c\} \leq Cp^2 \exp(-\min(n, \sqrt{p})/C)$ for some absolute C . Plugging this in Eq. (171) yields the lemma. \blacksquare

We are now ready to prove Proposition 15. Indeed, as in Proposition 13, for any unit vector $\mathbf{y} \in \mathbb{R}^{\mathbf{Q}^c}$, let $\mathbf{S} = \{i : |y_i| \geq \sqrt{A/p}\}$ and $\mathbf{y}_S, \mathbf{y}_{S^c}$ denote the projections on the indices in \mathbf{S}, \mathbf{S}^c respectively.

$$\tilde{\mathbb{P}} \left\{ \|\bar{\mathbf{C}}_1\| \geq \Delta \right\} \leq \tilde{\mathbb{P}} \left\{ \max_{\mathbf{w} \in T_{\mathbf{Q}}^{\varepsilon}, \mathbf{y} \in T_{\mathbf{Q}^c}^{\varepsilon}} |\langle \mathbf{w}, \bar{\mathbf{C}}\mathbf{y} \rangle| \geq \Delta(1 - 2\varepsilon) \right\} \quad (176)$$

$$\begin{aligned} &\leq \tilde{\mathbb{P}} \left\{ \max_{\mathbf{w} \in T_{\mathbf{Q}}^{\varepsilon}, \mathbf{y} \in T_{\mathbf{Q}^c}^{\varepsilon}} |\langle \mathbf{w}, \bar{\mathbf{C}}\mathbf{y}_S \rangle| \geq \Delta(1 - 2\varepsilon)/2 \right\} \\ &\quad + \mathbb{P} \left\{ \max_{\mathbf{w} \in T_{\mathbf{Q}}^{\varepsilon}, \mathbf{y} \in T_{\mathbf{Q}^c}^{\varepsilon}} |\langle \mathbf{w}, \bar{\mathbf{C}}\mathbf{y}_{S^c} \rangle| \geq \Delta(1 - 2\varepsilon)/2 \right\}. \end{aligned} \quad (177)$$

As before, we will let $\varepsilon = 1/4$. The first term is controlled via Lemma 22, while the second is controlled by Lemma 23. We keep $\Delta = \Delta_*$ where

$$\Delta_* = C \left(L_* \sqrt{p} + L \sqrt{\frac{p \log A}{A}} \right). \quad (178)$$

so that, via the bounds of Lemmas 22, 23 and that $s_0^2 \leq p$:

$$\mathbb{P}\{\|\mathbf{C}_1\| \geq \Delta_*\} \leq C \exp\left(-c \frac{p \log A}{A}\right) + L_*^{-2} (np)^C \exp(-c \min(\sqrt{p}, n)). \quad (179)$$

We now set $A = ((\tau^2/K) \exp(\tau^2/K))^{1/2}$ with $K = C(\beta \vee 1)$ for a suitable constant C and, since $\tau \leq \sqrt{\log p}/2$, we get that $A \leq p^{1/3}$. Furthermore, it is straightforward to see that $L \geq (np)^{-C}$, and this implies that

$$\mathbb{P}\{\|\mathbf{C}_1\| \geq \Delta_*\} \leq (np)^C \exp(-c \min(\sqrt{p}, n)) = o(1). \quad (180)$$

With this setting of A , we get the form of Δ_* below, as required for the proposition.

$$\Delta_* \leq C e^{-c\tau^2/K} \sqrt{\frac{\tau^2 \vee 1}{K}} \cdot \frac{pm(\beta \vee 1) + p^2}{n^2} \quad (181)$$

$$\leq C (\tau \vee 1) e^{-c\tau^2/K} \sqrt{\frac{p}{n}} \vee \frac{p}{n}. \quad (182)$$

6.4 Proof of Proposition 16

Since \mathbf{D} is a diagonal matrix, its spectral norm is bounded by the maximum of its entries. This is easily done as, for every $i \in \mathbf{Q}^c$:

$$|(\mathbf{D})_{ii}| = \left| \eta \left(\frac{\|\tilde{\mathbf{z}}_i\|^2}{n} - 1; \frac{\tau}{\sqrt{n}} \right) \right| \quad (183)$$

$$\leq \frac{\|\tilde{\mathbf{z}}_i\|^2 - n}{n}. \quad (184)$$

By the Chernoff bound for χ^2 -squared random variables as in Lemma 12 followed by the union bound, with probability $1 - o(1)$:

$$\max_i \left| \frac{\|\tilde{\mathbf{z}}_i\|^2}{n} - 1 \right| \leq C \sqrt{\frac{\log p}{n}} \quad (185)$$

for some absolute C . Here we used the fact that $(\log p)/n < 1$.

6.5 Proof of Proposition 17

It suffices to show that with probability $1 - o(1)$

$$\max_{i,j \in \mathbf{U} \cap \mathbf{Q}} |\hat{\Sigma}_{ij}| \leq \frac{\tau}{\sqrt{n}} = C_0(\beta \vee 1) \sqrt{\frac{\log p}{n}}. \quad (186)$$

This is a standard argument (see Bickel and Levina, 2008b, Lemma A.3) where (following the dependence on β) it suffices to take $\tau \geq C_0(\beta \vee 1) \sqrt{\log p}$ for C_0 a sufficiently large absolute constant. We note here that the same can also be proved via the conditioning technique applied in the proofs of Propositions 13 and 15.

7. Proof of Theorems 3

Throughout this section, to lighten notation, we drop the prime from $\hat{\Sigma}'$ and \mathbf{X}' while keeping in mind that these are independent from $\hat{\mathbf{V}}_1, \dots, \hat{\mathbf{V}}_r$. We further write $\mathbf{X} = \mathbf{U}\mathbf{B}\mathbf{V}^T + \mathbf{Z}$, where $\mathbf{U} \in \mathbb{R}^{n \times r}$ is the matrix with columns $\mathbf{u}_1, \dots, \mathbf{u}_r$, \mathbf{B} is diagonal with $B_{ii} = \sqrt{\beta_i}$ and $\mathbf{V} \in \mathbb{R}^{p \times r}$ has columns $\mathbf{v}_1, \dots, \mathbf{v}_r$.

Define the event

$$\mathcal{U} \equiv \left\{ \mathbf{U} \in \mathbb{R}^{n \times r} : \left\| \frac{1}{n} \mathbf{U}^T \mathbf{U} - \mathbf{I}_{r \times r} \right\|_{\text{op}} \leq 3 \sqrt{\frac{\tau}{n}} \right\}. \quad (187)$$

By the Bai-Yin law on eigenvalues of Wishart matrices (see Vershynin, 2012), $\lim_{n \rightarrow \infty} \mathbb{P}(\mathbf{U} \in \mathcal{U}) = 1$. In the rest of the proof, we will therefore assume $\mathbf{U} \in \mathcal{U}$ fixed, and denote by $\mathbb{P}(\cdot) = \mathbb{P}(\cdot | \mathbf{U})$ the expectation conditional on \mathbf{U} . In other words, $\mathbb{E}(\cdot)$ denotes expectation with respect to \mathbf{Z} .

Note that

$$\hat{\Sigma} = \frac{1}{n} \mathbf{V}\mathbf{B}\mathbf{U}^T \mathbf{U}\mathbf{B}\mathbf{V}^T + \frac{1}{n} \mathbf{Z}^T \mathbf{U}\mathbf{B}\mathbf{V}^T + \frac{1}{n} \mathbf{V}\mathbf{B}\mathbf{U}^T \mathbf{Z} + \frac{1}{n} \mathbf{Z}^T \mathbf{Z} - \mathbf{I}. \quad (188)$$

We then have, for $q \in \{1, \dots, r\}$ and $i \in \{1, \dots, p\}$,

$$\left| \langle \hat{\Sigma} \hat{\mathbf{v}}_q \rangle_i - \beta_q \langle \mathbf{v}_q, \hat{\mathbf{v}}_q \rangle_{q_{q,i}} \right| \leq T_{i,q}^{(1)} + T_{i,q}^{(2)} + T_{i,q}^{(3)}, \quad (189)$$

$$T_{i,q}^{(1)} \equiv \left| \frac{1}{n} \langle \mathbf{e}_i, \mathbf{V}\mathbf{B}\mathbf{U}^T \mathbf{U}\mathbf{B}\mathbf{V}^T \hat{\mathbf{v}}_q \rangle - \beta_q \langle \mathbf{v}_q, \hat{\mathbf{v}}_q \rangle_{q_{q,i}} \right|, \quad (190)$$

$$T_{i,q}^{(2)} \equiv \left| \frac{1}{n} \langle \mathbf{Z}, [\mathbf{U}\mathbf{B}\mathbf{V}^T \mathbf{e}_i] \hat{\mathbf{v}}_q^T + (\mathbf{U}\mathbf{B}\mathbf{V}^T \hat{\mathbf{v}}_q) \mathbf{e}_i^T \rangle \right|, \quad (191)$$

$$T_{i,q}^{(3)} \equiv \left| \langle \mathbf{e}_i, \left(\frac{1}{n} \mathbf{Z}^T \mathbf{Z} - \mathbf{I} \right) \hat{\mathbf{v}}_q \rangle \right|. \quad (192)$$

We next bound, with high probability, $\max_{i,q} T_{i,q}^{(a)}$ for $a \in \{1, 2, 3\}$. Throughout we let $\varepsilon \equiv \max_{q \in [r]} \|\hat{\mathbf{v}}_q - \mathbf{v}_q\|$.

Considering the first term, we have

$$\begin{aligned} T_{i,q}^{(1)} &\leq \left| \langle \mathbf{e}_i, \mathbf{V}\mathbf{B} \left(\frac{1}{n} \mathbf{U}^T \mathbf{U} - \mathbf{I} \right) \mathbf{B}\mathbf{V}^T \hat{\mathbf{v}}_q \rangle \right| + \left| \langle \mathbf{e}_i, \mathbf{V}\mathbf{B}^2 \mathbf{V}^T \hat{\mathbf{v}}_q \rangle - \beta_q \langle \mathbf{v}_q, \hat{\mathbf{v}}_q \rangle_{q_{q,i}} \right| \\ &\leq 2\beta \sqrt{\frac{\tau}{n}} + \beta \varepsilon \sqrt{\tau} \max_{q \in [r]} \max_{i \in [p]} |v_{q,i}|, \end{aligned} \quad (194)$$

where in the last inequality we used $\sum_{q \in [r] \setminus \{q\}} \langle \mathbf{v}_q, \hat{\mathbf{v}}_q \rangle^2 \leq 1 - \langle \mathbf{v}_q, \hat{\mathbf{v}}_q \rangle^2 \leq \varepsilon^2/2$.

Consider next the second term. Since $Z_{ij} \sim_{iid} \mathcal{N}(0, 1)$, it follows that $T_{i,q}^{(2)} = |W_{i,q}|$ for $W_{i,q} \sim \mathcal{N}(0, \sigma_{i,q}^2)$ a Gaussian random variable with variance

$$\sigma_{i,q}^2 = \frac{1}{n^2} \left\| (\mathbf{U}\mathbf{B}\mathbf{V}^T \mathbf{e}_i) \hat{\mathbf{v}}_q^T + (\mathbf{U}\mathbf{B}\mathbf{V}^T \hat{\mathbf{v}}_q) \mathbf{e}_i^T \right\|^2 \quad (195)$$

$$\leq \frac{2}{n^2} \left\{ \|\mathbf{U}\mathbf{B}\mathbf{V}^T \mathbf{e}_i\|^2 + \|\mathbf{U}\mathbf{B}\mathbf{V}^T \hat{\mathbf{v}}_q\|^2 \right\} \quad (196)$$

$$\leq \frac{4}{n^2} \|\mathbf{U}\mathbf{B}\mathbf{V}^T\|_{\text{op}}^2 \quad (197)$$

$$\leq \frac{4}{n^2} \|\mathbf{U}\|_{\text{op}}^2 \|\mathbf{B}\|_{\text{op}}^2 \leq \frac{8\beta^2}{n}. \quad (198)$$

By union bound over $i \in [p]$, $q \in [r]$ we obtain

$$\max_{i \in [p], q \in [r]} T_{i,q}^{(2)} \leq 8\beta \sqrt{\frac{\log p}{n}}. \quad (199)$$

Finally, consider the last term. By rotational invariance of \mathbf{Z} , the distribution of $T_{i,q}^{(3)}$ only depends on the angle between \mathbf{e}_i and $\hat{\mathbf{v}}_q$. Calling this angle ϑ , we have

$$T_{i,q}^{(3)} \stackrel{d}{=} \left| \langle \mathbf{e}_i, \left(\frac{1}{n} \mathbf{Z}^T \mathbf{Z} - \mathbf{I} \right) \mathbf{e}_1 \rangle \cos \vartheta + \langle \mathbf{e}_i, \left(\frac{1}{n} \mathbf{Z}^T \mathbf{Z} - \mathbf{I} \right) \mathbf{e}_2 \rangle \sin \vartheta \right| \quad (200)$$

$$\leq \left| \frac{1}{n} \|\mathbf{z}_1\|^2 - 1 \right| + \left| \frac{1}{n} \langle \mathbf{z}_1, \mathbf{z}_2 \rangle \right|. \quad (201)$$

Both of these terms have Bernstein-type tail bounds, whence

$$\mathbb{P} \left(T_{i,q}^{(3)} \geq \frac{t}{\sqrt{n}} \right) \leq 2 \exp \left\{ -c \min(t\sqrt{n}, t^2) \right\}. \quad (202)$$

Using $t = C_0\sqrt{\log p}$, and recalling that $n \geq C \log p$ for C a large constant, we obtain $\mathbb{P}(T_{i,q}^{(3)} \geq C_0\sqrt{(\log p)/n}) \leq 2p^{-10}$. Hence by union bound

$$\max_{i \in [p], q \in [r]} T_{i,q}^{(3)} \leq C_0\sqrt{\frac{\log p}{n}}. \quad (203)$$

By putting together Eqs. (194), (199), (203), and using assumption A2, we get

$$|(\widehat{\Sigma}\widehat{\mathbf{v}}_q)_i - \beta_q \langle \mathbf{v}_q, \widehat{\mathbf{v}}_q \rangle v_{q,i}| \leq C\beta\sqrt{\frac{r}{n}} + C(\beta \vee 1)\sqrt{\frac{\log p}{n}} + \beta\epsilon\gamma\sqrt{r} |v_{q,i}| \mathbb{I}(i \in \mathbf{Q}). \quad (204)$$

Let $\widehat{\mathbf{Q}}_q = \{i \in [p] : |(\widehat{\Sigma}\widehat{\mathbf{v}}_q)_i| \geq \rho\}$. We claim that the above implies that, with high probability, $\mathbf{Q}_q \subseteq \widehat{\mathbf{Q}}_q \subseteq \mathbf{Q}$ for all q .

For $i \notin \mathbf{Q}$, we have

$$|(\widehat{\Sigma}\widehat{\mathbf{v}}_q)_i| \leq C\beta\sqrt{\frac{r}{n}} + C(\beta \vee 1)\sqrt{\frac{\log p}{n}} \quad (205)$$

$$< \frac{\beta_{\min}\theta}{2\sqrt{s_0}}, \quad (206)$$

where the last inequality follows from Eq. (14).

On the other hand, By Theorem 2 and using the assumption (14), we can guarantee

$$\epsilon \leq \frac{1}{8} \left(\frac{\beta_{\min}}{\beta\gamma\sqrt{r}} \wedge 1 \right). \quad (207)$$

Hence for $i \in \mathbf{Q}_q$, and considering –to be definite– $v_{q,i} > 0$, we get

$$(\widehat{\Sigma}\widehat{\mathbf{v}}_q)_i \geq \beta_q \langle \mathbf{v}_q, \widehat{\mathbf{v}}_q \rangle v_{q,i} - C\beta\sqrt{\frac{r}{n}} - C(\beta \vee 1)\sqrt{\frac{\log p}{n}} - \beta\epsilon\gamma\sqrt{r} |v_{q,i}| \quad (208)$$

$$\geq \beta_{\min} \left(1 - \epsilon - \frac{\beta}{\beta_{\min}} \epsilon\gamma\sqrt{r} \right) v_{q,i} - C\beta\sqrt{\frac{r}{n}} - C(\beta \vee 1)\sqrt{\frac{\log p}{n}} \quad (209)$$

$$\geq \frac{3\beta_{\min}\theta}{4\sqrt{s_0}} - C\beta\sqrt{\frac{r}{n}} - C(\beta \vee 1)\sqrt{\frac{\log p}{n}} \quad (210)$$

$$> \frac{\beta_{\min}\theta}{2\sqrt{s_0}}. \quad (211)$$

where, in the first inequality, we used $\langle \mathbf{v}_q, \widehat{\mathbf{v}}_q \rangle \geq 1 - \epsilon$.

This concludes the proof. Keeping track of the dependence on θ , γ , β , β_{\min} , we get that the following conditions are sufficient for the theorem's conclusion to hold (with C a

suitable numerical constant):

$$n \geq C \frac{(\beta^2 \vee 1)}{\beta_{\min}^2 \theta^2} s_0 \log p, \quad (212)$$

$$n \geq C \frac{\beta^2}{\beta_{\min}^2 \theta^2} r s_0, \quad (213)$$

$$n \geq C \left\{ \frac{\beta^4 \vee \beta^2}{\beta_{\min}^2} \gamma^2 \right\} r s_0^2 \log \frac{p}{s_0^2}, \quad (214)$$

$$n \geq C \frac{(\beta^2 \vee 1)}{\beta_{\min}^2} s_0^2 \log \frac{p}{s_0^2}. \quad (215)$$

All of these conditions are implied by the assumptions of Theorem 3, namely Eq. (14). In particular, this is shown by using the fact that $s_0 \log p \leq s_0^2 \log(p/s_0^2)$ for $s_0 \leq \sqrt{p}$.

Acknowledgments

We are grateful to David Donoho for his feedback on an early draft of this manuscript. This work was partially supported by the NSF CAREER award CCF-0743978, the NSF grant CCF-1319979, and the grants AFOSR/DARPA FA9550-12-1-0411 and FA9550-13-1-0036.

References

- Arash A. Amni and Martin J. Wainwright. High-dimensional analysis of semidefinite relaxations for sparse principal components. *The Annals of Statistics*, 37(5B):2877–2921, 2009.
- Jinhu Baik, Gérard Ben Arous, and Sandrine Péché. Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices. *Annals of Probability*, pages 1643–1697, 2005.
- Florent Benaych-Georges and Raj Rao Nadakuditi. The eigenvalues and eigenvectors of finite, low rank perturbations of large random matrices. *Advances in Mathematics*, 227(1):494–521, 2011.
- Quentin Berthet and Philippe Rigollet. Computational lower bounds for sparse pca. *arXiv preprint arXiv:1304.0828*, 2013.
- Peter J Bickel and Elizaveta Levina. Covariance regularization by thresholding. *The Annals of Statistics*, pages 2577–2604, 2008a.
- Peter J Bickel and Elizaveta Levina. Regularized estimation of large covariance matrices. *The Annals of Statistics*, pages 199–227, 2008b.
- T Tony Cai, Cun-Hui Zhang, Harrison H Zhou, et al. Optimal rates of convergence for covariance matrix estimation. *The Annals of Statistics*, 38(4):2118–2144, 2010.

- T Tony Cai, Harrison H Zhou, et al. Optimal rates of convergence for sparse covariance matrix estimation. *The Annals of Statistics*, 40(5):2389–2420, 2012.
- T Tony Cai, Zongming Ma, Yihong Wu, et al. Sparse pca: Optimal rates and adaptive estimation. *The Annals of Statistics*, 41(6):3074–3110, 2013.
- Tony Cai and Weidong Liu. Adaptive thresholding for sparse covariance matrix estimation. *Journal of the American Statistical Association*, 106(494):672–684, 2011.
- Mireille Capitaine, Catherine Donati-Martin, and Delphine Féral. The largest eigenvalues of finite rank deformation of large wigner matrices: convergence and nonuniversality of the fluctuations. *The Annals of Probability*, 37(1):1–47, 2009.
- Xinyuan Cheng and Amit Singer. The spectrum of random inner-product kernel matrices. *Random Matrices: Theory and Applications*, 2(04):1350010, 2013.
- Alexandre d’Aspremont, Laurent El Ghaoui, Michael I Jordan, and Gert RG Lanckriet. A direct formulation for sparse pca using semidefinite programming. *SIAM review*, 49(3):434–448, 2007.
- Alexandre d’Aspremont, Francis Bach, and Laurent El Ghaoui. Optimal solutions for sparse principal component analysis. *The Journal of Machine Learning Research*, 9:1269–1294, 2008.
- Chandler Davis and William Morton Kahan. The rotation of eigenvectors by a perturbation. iii. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.
- Yash Deshpande and Andrea Montanari. Sparse pca via covariance thresholding. In *Advances in Neural Information Processing Systems*, pages 334–342, 2014.
- David L. Donoho and Iain M. Johnstone. Minimax risk over l_p balls. *Prob. Th. and Rel. Fields*, 99:277–303, 1994.
- Noureddine El Karoui. On information plus noise kernel random matrices. *The Annals of Statistics*, 38(5):3191–3216, 2010a.
- Noureddine El Karoui. The spectrum of kernel random matrices. *The Annals of Statistics*, 38(1):1–50, 2010b.
- Zhou Fan and Andrea Montanari. The spectral norm of random inner-product kernel matrices. arXiv:1507.05343, 2015.
- Delphine Féral and Sandrine Péché. The largest eigenvalue of rank one deformation of large wigner matrices. *Communications in mathematical physics*, 272(1):185–228, 2007.
- Iain M. Johnstone. Function estimation and gaussian sequence models. *Unpublished manuscript*, 2015.
- Iain M Johnstone and Arthur Yu Lu. Sparse principal components analysis. *Unpublished manuscript*, 2004.
- Iain M Johnstone and Arthur Yu Lu. On consistency and sparsity for principal components analysis in high dimensions. *Journal of the American Statistical Association*, 104(486), 2009.
- Noureddine El Karoui. Operator norm consistent estimation of large-dimensional sparse covariance matrices. *The Annals of Statistics*, pages 2717–2756, 2008.
- Antti Knowles and Jun Yin. The isotropic semicircle law and deformation of wigner matrices. *Communications on Pure and Applied Mathematics*, 2013.
- Robert Krauthgamer, Boaz Nadler, Dan Vilenchik, et al. Do semidefinite relaxations solve sparse pca up to the information limit? *The Annals of Statistics*, 43(3):1300–1322, 2015.
- Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pages 1302–1338, 2000.
- Michel Ledoux. *The concentration of measure phenomenon*. Number 89. American Mathematical Soc., 2005.
- Tengyu Ma and Avi Wigderson. Sum-of-squares lower bounds for sparse pca. In *Advances in Neural Information Processing Systems*, pages 1603–1611, 2015.
- Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, pages 1436–1462, 2006.
- Baback Moghaddam, Yair Weiss, and Shai Avidan. Spectral bounds for sparse pca: Exact and greedy algorithms. In *Advances in neural information processing systems*, pages 915–922, 2005.
- Debashis Paul. Asymptotics of sample eigenstructure for a large dimensional spiked covariance model. *Statistica Sinica*, 17(4):1617, 2007.
- R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. In Y.C. Eldar and G. Kutyniok, editors, *Compressed Sensing: Theory and Applications*, pages 210–268. Cambridge University Press, 2012.
- Vincent Q Yu and Jing Lei. Minimax rates of estimation for sparse pca in high dimensions. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS) 2012*, 2012.
- Martin J Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1 -constrained quadratic programming (lasso). *Information Theory, IEEE Transactions on*, 55(5):2183–2202, 2009.
- Tengyao Wang, Quentin Berthet, and Richard J Samworth. Statistical and computational trade-offs in estimation of sparse principal components. arXiv preprint arXiv:1408.5369, 2014.
- Daniela M Witten, Robert Tibshirani, and Trevor Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009.

Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.

Large Scale Visual Recognition through Adaptation using Joint Representation and Multiple Instance Learning

Judy Hoffman
Deepak Pathak
Eric Tzeng
Jonathan Long
Sergio Guadarrama*
Trevor Darrell

*Department of Electrical Engineering and Computer Science
University of California
Berkeley, CA 94720, USA*

Kate Saenko

*Department of Computer Science
University of Massachusetts
Lowell, Massachusetts 01854, USA*

JHOFFMAN@EECS.BERKELEY.EDU
PATHAK@BERKELEY.EDU
ETZENG@EECS.BERKELEY.EDU
JONLONG@BERKELEY.EDU
SGUADA@EECS.BERKELEY.EDU
TREVOR@EECS.BERKELEY.EDU

SAENKO@CS.UML.EDU

Editor: Kevin Murphy

Abstract

A major barrier towards scaling visual recognition systems is the difficulty of obtaining labeled images for large numbers of categories. Recently, deep convolutional neural networks (CNNs) trained used 1.2M+ labeled images have emerged as clear winners on object classification benchmarks. Unfortunately, only a small fraction of those labels are available with bounding box localization for training the detection task and even fewer pixel level annotations are available for semantic segmentation. It is much cheaper and easier to collect large quantities of image-level labels from search engines than it is to collect scene-centric images with precisely localized labels. We develop methods for learning large scale recognition models which exploit joint training over both weak (image-level) and strong (bounding box) labels and which transfer learned perceptual representations from strongly-labeled auxiliary tasks. We provide a novel formulation of a joint multiple instance learning method that includes examples from object-centric data with image-level labels when available, and also performs domain transfer learning to improve the underlying detector representation. We then show how to use our large scale detectors to produce pixel level annotations. Using our method, we produce a >7.6K category detector and release code and models at lsva.berkeleyvision.org.

Keywords: Computer Vision, Deep Learning, Transfer Learning, Large Scale Learning

1. Introduction

It is well known that contemporary visual models thrive on large amounts of training data, especially those that directly include labels for the desired tasks. Many real world settings contain labels with varying specificity, e.g., “strong” bounding box detection labels,

and “weak” labels indicating presence somewhere in the image. We tackle the problem of *joint detector and representation learning*, and develop models which cooperatively exploit heterogeneous sources of training data, where some classes have no “strong” annotations. Our model optimizes a latent variable multiple instance learning model over image regions while simultaneously transferring a shared representation from detection-domain models to classification-domain models. The latter provides a key source of automatic and accurate initialization for latent variable optimization, which has heretofore been unavailable in such methods.

Both classification and detection are key visual recognition challenges, though historically very different architectures have been deployed for each. Recently, the R-CNN model (Girshick et al., 2014) showed how to adapt an ImageNet classifier into a detector, but required bounding box data for all categories. We ask, is there something generic in the transformation from classification to detection that can be learned on a subset of categories and then transferred to other classifiers?

One of the fundamental challenges in training object detection systems is the need to collect a large amount of images with bounding box annotations. The introduction of detection challenge datasets, such as PASCAL VOC (Everingham et al., 2010), has propelled progress by providing the research community a dataset with enough fully annotated images to train competitive models although only for 20 classes. Even though the more recent ILSVRC13 detection dataset (Russakovsky et al., 2014) has extended the set of annotated images, it only contains data for 200 categories. The larger ImageNet dataset contains some localization information for around 3000 object categories, though these are not exhaustively labeled. As we look forward towards the goal of scaling our systems to human-level category detection, it becomes impractical to collect a large quantity of bounding box labels for tens or hundreds of thousands of categories.

In contrast, image-level annotation is comparatively easy to acquire. The prevalence of image tags allows search engines to quickly produce a set of images that have some correspondence to any particular category. ImageNet (Berg et al., 2012), for example, has made use of these search results in combination with manual outlier detection to produce a large classification dataset comprised of over 20,000 categories. While this data can be effectively used to train object classifier models, it lacks the supervised annotations needed to train state-of-the-art detectors.

Previous methods employ varying combinations of weak and strong labels of the same object category to learn a detector. Such methods seldom exploit available strong-label data of different, auxiliary categories, despite the fact that such data is very often available in many practical scenarios. Deselaers et al. (2012) uses auxiliary data to learn generic objectness information just as an initial step, but doesn’t optimize jointly for weakly labeled data.

We introduce a new model for large-scale learning of detectors that can jointly exploit weak and strong labels, perform inference over latent regions in weakly labeled training examples, and can transfer representations learned from related tasks (see Figure 1). In practical settings, such as learning visual detector models for all available ImageNet categories, or for learning detector versions of other defined categories such as Sentibank’s adjective-noun-phrase models (Borth et al., 2013), our model makes greater use of available data and labels than previous approaches. Our method takes advantage of such data by

*. Now at Google Research.

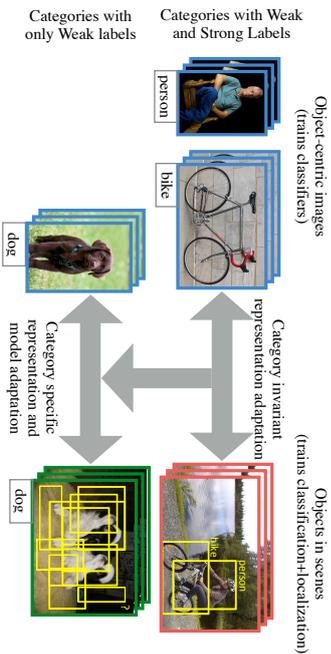


Figure 1: We learn detectors (models which classify and localize) for categories with only weak labels (*bottom row*). We use auxiliary categories with available paired strong and weak annotations (*top row*) to learn to adapt a visual representation from whole image classification to localized region detection. We then use the adapted representation to transform the classifiers trained for the categories with only weak labels and jointly solve an MIL problem to mine localized training data from the weakly labeled scene-centric training data (*green – bottom right*).

using the auxiliary strong labels to improve the feature representation for detection tasks, and uses the improved representation to learn a stronger detector from weak labels in a deep architecture.

We cast the task as a domain adaptation problem, considering the data used to train classifiers (images with category labels) as our source domain, and the data used to train detectors (images with bounding boxes and category labels) as our target domain. We then seek to find a general transformation from the source domain to the target domain, that can be applied to any image classifier to adapt it into a object detector (see Figure 1). R-CNN (Girshick et al., 2014) demonstrated that adaptation, in the form of fine-tuning, is very important for transferring deep features from classification to detection and partially inspired our approach. However, the R-CNN algorithm uses classification data only to pre-train a deep network and then requires a large number of bounding boxes to train each detection category.

To learn detectors, we exploit weakly labeled data for a concept, including both object-centric images (e.g., from ImageNet classification training data), and weakly labeled scene-centric imagery (e.g., from PASCAL or ImageNet detection training data with bounding box metadata removed). We define a novel multiple instance learning (MIL) framework that includes bags defined on both types of data, and also jointly optimizes an underlying perceptual representation using strong detection labels from related categories. We demonstrate that a good perceptual representation for detection tasks can be learned from a set of paired weak and strong labeled examples and the resulting adaptation can be transferred to new categories, even those for which no strong labels were available.

We additionally show that our large-scale detection models can be directly converted into models which produce pixel-level localization for each category. Following the recent result of Long et al. (2015), we run our models fully-convolutionally and directly use the learned detection weights to predict per-pixel labels.

We evaluate our detection model empirically on the largest set of available ground-truth visual data labeled with bounding box annotations, the ILSVRC13 detection dataset. Our method outperforms the previous best MIL-based approaches for weakly labeled detector learning (Wang et al., 2014) on ILSVRC13 (Russakovsky et al., 2014) by 200%. Our model is directly applicable to learning improved “detectors in the wild”, including categories in ImageNet but not in the ILSVRC13 detection dataset, or categories defined ad-hoc for a particular user or task with just a few training examples to fine-tune a new classification model. Such models can be promoted to detectors with no (or few) labeled bounding boxes.

The article builds on two conference publications. The generic feature adaptation for transforming a classifier into a detector was first presented in Hoffman et al. (2014). Hoffman et al. (2015) presented a further category specific detector and representation refinement with mined localization labels. In this work, we present and compare the two works and additionally present a novel extension for further producing per-pixel predictions for adapting to the semantic segmentation task.

2. Related Work

Since its inception, the multiple instance learning (MIL) problem (Diettrich et al., 1997), or learning from a set of labels that specify at least one instance in a bag of instances, has been attempted in several frameworks, including Noisy-OR and boosting (Ali and Saenko, 2014; Zhang et al., 2005). However, most commonly, it has been framed as a max-margin classification problem (Andrews et al., 2002), with latent parameters optimized using alternating optimization (Felzenszwalb et al., 2010; Yu and Joachims, 2009).

Recently, MIL has also been used in computer vision to train detectors using weak labels, i.e. images with category labels but without bounding box labels. The MIL paradigm estimates latent labels of examples in positive training bags, where each positive bag is known to contain at least one positive example. For example, Galleguillos et al. (2008) and Ali and Saenko (2014) construct positive bags from all object proposal regions in a weakly labeled image that is known to contain the object and use a version of MIL to learn an object detector. Overall, MIL is tackled in two stages: first, finding a good initialization, and second, using good heuristics for optimization. A number of methods have been proposed for initialization which include using a large image region excluding boundary (Pandey and Lazebnik, 2011), using a candidate set which covers the training data space (Song et al., 2014a,b), using unsupervised patch discovery (Siva et al., 2013; Singh et al., 2012), learning generic objectness knowledge from auxiliary categories (Alexe et al., 2010; Deselaers et al., 2012), learning latent categories from background to suppress it (Wang et al., 2014), or using class-specific similarity (Siva et al., 2012). Approaches to better optimize the non-convex problem involve using multi-fold learning as a measure of regularizing overfitting (Cimbis et al., 2014), optimizing Latent SVM for the area under the ROC curve (AUC) (Bilen et al., 2014), and training with easy examples initially to avoid bad local optima (Bengio et al., 2009; Kumar et al., 2010; Guillaumein et al., 2014).

While these approaches are promising, they often underperform on the full detection task in more challenging settings such as the PASCAL VOC dataset (Everingham et al., 2010), where objects only cover small portions of images, and many candidate bounding boxes contain no objects whatsoever. The major challenges faced by solutions to the MIL problem are the limitations of fixed feature representations and poor initializations, particularly in non-object centric images. Our algorithm provides solutions to both of these issues. We also provide an evaluation on the large-scale ILSVRC13 detection dataset, which many previous methods have not been evaluated on.

Deep convolutional neural networks (CNNs) have emerged as state of the art on popular object classification benchmarks such as ILSVRC (Krizhevsky et al., 2012) and MNIST. In fact, “deep features” extracted from CNNs trained on the object classification task are also state of the art on other tasks such as subcategory classification, scene classification, domain adaptation (Donahue et al., 2014), and even image matching (Fischer et al., 2014). Unlike the previously dominant features (SIFT (Lowe, 2004), HOG (Dalal and Triggs, 2005)), deep CNN features can be learned for each specific task, but only if sufficient labeled training data is available. R-CNN (Girshick et al., 2014) showed that fine-tuning deep features, pre-trained for classification, on a large amount of bounding box labeled data significantly improves detection performance.

Domain adaptation methods aim to reduce dataset bias caused by a difference in the statistical distributions between training and test domains. In this paper, we treat the transformation of classifiers into detectors as a domain adaptation task. Many approaches have been proposed for classifier adaptation, such as feature space transformations (Saenko et al., 2010; Kulis et al., 2011; Gong et al., 2012; Fernando et al., 2013), model adaptation approaches (Yang et al., 2007a; Aytar and Zisserman, 2011), and joint feature and model adaptation (Hoffman et al., 2013a; Duan et al., 2012). However, even the joint learning models are not able to modify the feature extraction process and so are limited to shallow adaptation techniques. Additionally, these methods only adapt between visual domains, keeping the task fixed, while we adapt both from a large visual domain to a smaller visual domain and from a classification task to a detection task.

However, domain adaptation techniques have seen recent success through the merger with deep CNNs. Hoffman et al. (2013b) showed that, when training data in the target domain is severely limited or unavailable, domain adaptation techniques as applied to CNNs can be more effective than the standard practice of fine-tuning. More recent works have seen success in augmenting deep architectures with additional regularization layers that are robust to the negative effects of domain shift (Chifary et al., 2014; Tzeng et al., 2014; Long and Wang, 2015; Ganin and Lempitsky, 2015). However, all of these methods focus on the standard visual domain adaptation problem, where one adapts between two versions of the same task with different statistics, and do not investigate the task adaptation setting.

Several supervised domain adaptation models have been proposed for object detection. Given a detector trained on a source domain, they adjust its parameters on labeled target domain data. These include variants for linear support vector machines (Yang et al., 2007b; Aytar and Zisserman, 2011; Donahue et al., 2013), as well as adaptive latent SVMs (Xu et al., 2014) and adaptive exemplar SVM (Aytar and Zisserman, 2012). A related recent method (Gochring et al., 2014) proposes a fast adaptation technique based on Linear Discriminant Analysis. These methods require strongly labeled data with bounding box an-

notations for all object categories, both in the source and target domains, which is absent in our scenario.

Other methods have been proposed that use the underlying semantic hierarchy of ImageNet to transfer localization information to classes for strong labels are available (Guillaumin and Ferrari, 2012; Vezhnevets and Ferrari, 2014). However, this necessarily limits their approaches to settings in which additional semantic information is available.

2.1 Background: MIL

We begin by briefly reviewing a standard solution to the multiple instance learning problem, Multiple Instance SVMs (MI-SVMs) (Andrews et al., 2002) or Latent SVMs (Felzenszwalb et al., 2010; Yu and Joachims, 2009). In this setting, each weakly labeled image is considered a collection of bounding boxes which form a positive ‘bag’. For a binary classification problem, the task is to maximize the bag margin which is defined by the instance with highest confidence. For each weakly labeled image $I \in \mathcal{W}$, we collect a set of bounding boxes and define the index set of those boxes as R_I . We next define a bag as $B_I = \{\mathbf{x}_i | i \in R_I\}$, with label Y_I , and let the i^{th} instance in the bag be $(\mathbf{x}_i, y_i) \in \mathcal{R}^p \times \{-1, +1\}$.

For an image with a negative image-level label, $Y_I = -1$, we label all bounding boxes in the image as negative. For an image with a positive image-level label, $Y_I = 1$, we create a constraint that at least one positive instance occurs in the image bag.

In a typical detection scenario, R_I corresponds to the set of possible bounding boxes inside the image, and maximizing over R_I is equivalent to discovering the bounding box that contains the positive object. We define a representation $\phi(\mathbf{x}_i) \in \mathcal{R}^d$ for each instance, which is the feature descriptor for the corresponding bounding box, and formulate the MI-SVM objective as follows:

$$\min_{\mathbf{w} \in \mathcal{R}^d} \frac{1}{2} \|\mathbf{w}\|_2^2 + \alpha \sum_I \ell \left(Y_I, \max_{i \in R_I} \phi(\mathbf{x}_i) \right) \quad (1)$$

where α is a hyper-parameter and $\ell(y, \hat{y})$ is the hinge loss. Interestingly, for negative bags i.e. $Y_I = -1$, the knowledge that all instances are negative allows us to unfold the max operation into a sum over each instance. Thus, Equation (1) reduces to a standard QP with respect to \mathbf{w} . For the case of positive bags, this formulation reduces to a standard SVM if the maximum scoring instance is known.

Based on this idea, Equation (1) is optimized using a classic concave-convex procedure (Yuille and Rangarajan, 2003), which decreases the objective value monotonically with a guarantee to converge to a local minima or saddle point. Due to this reason, weakly trained MIL detectors are sensitive to the feature representation and initial detector weights (i.e. initialization in MIL) (Cinbis et al., 2014; Song et al., 2014a). With our algorithm we mitigate these sensitivities by learning a representation that works well for detection and by proposing an initialization technique for the weakly trained detectors which proves to avoid many of the pitfalls of prior MIL techniques (see Fig 7).

3. Large Scale Detection through Adaptation

We propose a learning algorithm that uses a heterogeneous data source, containing only weak labels for some tasks, to produce strong visual recognition models for all. Our approach

is to cast the shift from tasks that require weak labels to tasks that require strong labels as a domain adaptation problem. We then consider transforming the models for the weakly labeled task into the models for the strongly labeled task. For concreteness, we will present our algorithm applied to the specific task shift of classification to detection, called Large Scale Detection through Adaptation (LSDA). In the following section, we will explain how to shift to a different strongly labeled task of semantic segmentation.

Let the set of images with only weak labels be denoted as \mathcal{W} and the set of images with strong labels (bounding box annotations) from auxiliary tasks be denoted as \mathcal{S} . We assume that the set of object categories that appear in the weakly labeled set, C_W , do not overlap with the set of object categories that appear in the strongly labeled set, C_S . For each image in the weakly labeled set, $I \in \mathcal{W}$, we have an image-level label per category, $k: Y_I^k \in \{1, -1\}$. For each image in the strongly labeled set, $I \in \mathcal{S}$, we have a label per category, k , per region in the image, $i \in R_I: y_i^k \in \{1, -1\}$. We seek to learn a representation, $\phi(\cdot)$ that can be used to train detectors for all object categories, $C = \{C_W \cup C_S\}$. For a category $k \in C$, we denote the category specific detection parameter as w_k and compute our final detection scores per region, x , as $score_k(x) = w_k^T \phi(x)$.

We propose a joint optimization algorithm which learns a feature representation, $\phi(\cdot)$, and detection model parameters, w_k , using the combination of strongly labeled scene-centric data, \mathcal{S} , with weakly labeled object and scene-centric data, \mathcal{W} . For a fixed representation, one can directly train detectors for all categories represented in the strongly labeled set, $k \in C_S$. Additionally, for the same fixed representations, we reviewed in the previous section techniques to train detectors for the categories in the weakly labeled data set, $k \in C_W$. Our insight is that the knowledge from the strong label set can be used to help guide the optimization for the weak labeled set, and we can explicitly adapt our representation for the categories of interest and for the generic detection task.

Below, we state our overall objective:

$$\min_{\substack{w_k, \phi \\ k \in C}} \sum_k \Gamma(w_k) + \alpha_1 \sum_{I \in \mathcal{W}, p \in C_W} \mathcal{F}(Y_I^p, w_p) + \alpha_2 \sum_{I \in \mathcal{S}} \sum_{i \in R_I, q \in C_S} \ell(y_i^q, w_q^T \phi(x_i)) \quad (2)$$

where $\ell(\cdot)$ is the cross-entropy loss function, \mathcal{F} is the region-based loss function over weak categories, α_1, α_2 are scalar hyper-parameters and $\Gamma(\cdot)$ is a regularization over the detector weights. We use convolutional neural networks (CNNs) to define our representation ϕ and thus the last layer weights serve as detection weights w . We adopt the CNN architecture of Krizhevsky et al. (2012) (referred to as *AlexNet*).

This formulation is difficult to optimize directly, so we propose to solve this objective by sequentially optimizing easier sub-problems which are less likely to diverge (see Figure 2).

Lets describe the sub-problems for our overall approach. We begin by initializing a feature representation ϕ and the detection weights w using auxiliary weakly labeled data (Figure 2: *blue boxes*). These weights can be used to compute scores per region proposal to produce initial detection scores. We next use available strongly labeled data from auxiliary tasks to transfer category invariant information about the detection problem. We accomplish this through further optimizing our feature representation and learning generic background detection weights, w, ϕ , (Figure 2: *red boxes*). We then use the well tuned detection feature space to perform MIL on our weakly labeled data to find positive instances (Figure 2: *yellow boxes*). Finally, we use our discovered positive instances together with the

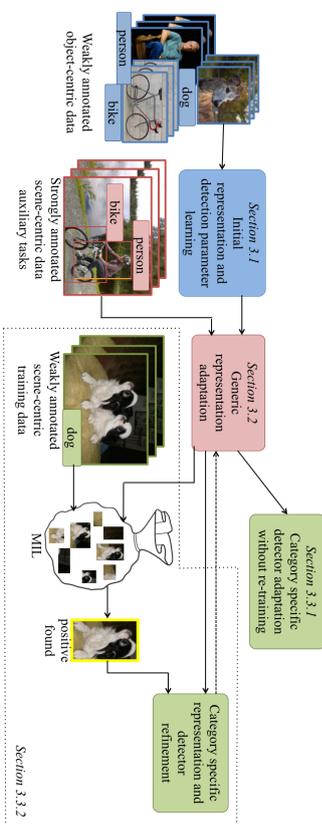


Figure 2: Our method (LSDA) jointly optimizes a representation and category specific detection parameters for categories with only weakly labeled data. We first learn a feature representation conducive to adaptation by initializing all parameters with weakly labeled data. We then collectively refine the feature space with strongly labeled data from auxiliary tasks to adapt the category invariant representation from classification to detection (red boxes). Finally, we perform category specific adaptation (green boxes) either without re-training or by solving MIL in our detection feature space and using the discovered bounding boxes to further refine the representation and detection weights.

strongly labeled data from auxiliary tasks to jointly optimize all parameters corresponding to feature representation and detection weights. We now describe each of these steps in detail in the follow subsections.

3.1 Initializing representation and detection parameters

As mentioned earlier, we use the AlexNet architecture to describe representation ϕ and detection weights w . Since this network requires a large amount of data and time to train its approximately 60 million parameters, we start by pre-training on the ILSVRC2012 classification dataset, which we refer to as auxiliary weakly labeled data. It contains 1.2 million weakly labeled images of 1000 categories. Pre-training on this dataset has been shown to be a very effective technique (Donahue et al., 2014; Sermanet et al., 2013; Girschick et al., 2014), both in terms of performance and in terms of limiting the amount of in-domain labeled data needed to successfully tune the network. This data is usually object centric and is therefore effective for training a network that is able to discriminate between different categories. Next, we replace the last weight layer (1000 linear classifiers) with $K = |C|$ randomly initialized linear classifiers, one for each category in our task.

We next learn initial values for all of the detection parameters for our particular categories of interest, $w_k, \forall k \in C$. We obtain such initialization by solving the simplified learning problem of image-level classification. The image, $I \in \mathcal{S}$, is labeled as positive for a category k if any of the regions in the image are labeled as positive for k and is labeled as negative otherwise, we denote the image level label as in the weakly labeled case: Y_I^k . Now, we can optimize over all images to refine the representation and learn category specific

parameters that can be used per region proposal to produce detection scores:

$$\min_{\mathbf{w}_k, \phi} \sum_k \left[\Gamma(\mathbf{w}_k) + \alpha \sum_{I \in \{W \cup S\}} \ell(Y_I^k, \mathbf{w}_k^T \phi(I)) \right] \quad (3)$$

We optimize Equation (3) through fine-tuning our CNN architecture with a new K -way last fully connected layer, where $K = |C|$. This serves as our initialization for solving sequential sub-problems to optimize overall objective (2). We find that even using the net trained on weakly labeled data in this way produces a strong baseline. We will refer this baseline as ‘*Classification Network*’ in the experiments; see Table 2.

3.2 Learning category specific representation and detection parameters

We next transform our classification network into a detection network and learn a representation which makes it possible to separate objects of interest from background and makes it easy to distinguish different object categories. We proceed by modifying the representation (layers 1-7), $\phi(\cdot)$, through finetuning, using the available strongly labeled data for categories in set C_S . Following the Regions-based CNN (R-CNN) (Girshick et al., 2014) algorithm, we collect positive bounding boxes for each category in set C_S as well as a set of background boxes using a region proposal algorithm, such as selective search (Uijlings et al., 2013). We use each labeled region as a fine-tuning input to the CNN after padding and warping it to the CNN’s input size. Note that the R-CNN fine-tuning algorithm requires bounding box annotated data for all categories and so can not directly be applied to train all K detectors. Fine-tuning transforms all network weights (except for the linear classifiers for categories in C_W) and produces a softmax detector for categories in set C_S , which includes a weight vector for the new background class. We find empirically that fine-tuning induces a generic, category invariant transformation of the classification network into a detection network. That is, even though fine-tuning sees no strongly labeled data for categories in set C_W , the network transforms in a way that automatically makes the original set C_W image classifiers much more effective at detection (see Figure 9). Fine-tuning for detection also learns a background weight vector that encodes a generic “background” category, \mathbf{w}_b . This background model is important for modeling the task shift from image classification, which does not include background distractors, to detection, which is dominated by background patches. This detector explicitly attempts to recognize all data labeled as negative in our bags. Since we initialize this detector with the strongly labeled data, we know precisely which regions correspond to background.

This can be summarized as the following intermediate sub-problem for objective (2):

$$\min_{\mathbf{w}_q, \phi} \sum_q \left[\Gamma(\mathbf{w}_q) + \alpha \sum_{I \in S} \sum_{i \in R_I} \ell(y_i^q, \mathbf{w}_q^T \phi(\mathbf{x}_i)) \right] \quad (4)$$

This is accomplished by fine-tuning our CNN architecture with the strongly labeled data, while keeping the detection weights for the categories with only weakly labeled data fixed. We will call this method as ‘*LSDA rep only*’ in our experiments.

3.3 Adapting category specific representation and detection parameters

Finally, we seek to adapt the category dependent representation and model parameters for the categories in our weakly labeled set, C_W . We will present two approaches to this problem of learning detection weights for weak categories. Specifically, we aim to update the weakly labeled category specific parameters. Section 3.3.1 presents a heuristic adaptation approach that requires no further CNN training with gradient descent and updates only the weakly labeled classification parameters. Section 3.3.2 describes a separate adaptation approach that directly optimizes a subproblem of our overall objective (2). It uses multiple instance learning to discover localized labeled regions in the weakly labeled training data and uses the discovered labels to adapt both the representation and the classification parameters for categories in the weakly labeled set.

3.3.1 K-NEAREST NEIGHBORS BASED ADAPTATION

In this section, we describe a technique for adapting the category specific parameters of the classifier model into the detector model parameters that are better suited for use with the detection feature representation based on a k -NN heuristic. We will determine a similarity metric between each category in the weakly labeled set, C_W , to the strongly labeled categories, C_S .

For simplicity, we separate the category specific output layer (8th layer of the network - f_{CS}) of the classification model into two components f_{CS} and f_{CW} , corresponding to model parameters for the categories in the strongly labeled set C_S and the weakly labeled set C_W , respectively. During our generic category adaptation of Section 3.1, we trained a new background prediction layer, f_{Cb} .

For categories in set C_S , adaptation to detectors can be learned directly through fine-tuning the category specific model parameters f_{CS} . This is equivalent to fixing f_{CS} and learning a new layer, zero initialized, δS , with equivalent loss to f_{CS} , and adding together the outputs of δS and f_{CS} .

Let us define the weights of the output layer of the original classification network as W^c , and the weights of the output layer of the adapted detection network as W^d . We know that for a category $i \in C_S$, the final detection weights should be computed as $W_i^d = W_i^c + \delta S_i$. However, since there is no strongly labeled data for categories in C_W , we cannot directly learn a corresponding δW layer during fine-tuning. Instead, we can approximate the fine-tuning that would have occurred to f_{CW} had strongly labeled data been available. We do this by finding the nearest neighbors categories in set C_S for each category in set C_W and applying the average change. We assume that there are categories in set C_S that are similar to those in set C_W and therefore have similar weights and similar gradient descent updates.

Here we define nearest neighbors as those categories with the nearest (minimal Euclidean distance) ℓ_2 -normalized f_{CS} parameters in the classification network. This corresponds to the classification model being most similar and hence, we assume, the detection model should be most similar. We denote the k^{th} nearest neighbor in set C_S of category $j \in C_W$ as $N_S(j, k)$, then we compute the final output detection weights for categories in set C_W as:

$$\forall j \in C_W : W_j^d = W_j^c + \frac{1}{k} \sum_{i=1}^k \delta S_{N_S(j, i)} \quad (5)$$

Thus, we adapt the category specific parameters even without bounding boxes for categories in set C_W . In section 5 we experiment with various values of k , including taking the full average: $k = |S|$. We will now refer to this method as ‘*LSDA rep+kNN*’ in our experiments.

3.3.2 MLL TRAINING BASED ADAPTATION

The previous section provides a technique for adapting the category specific model parameters for the weakly labeled categories without any further CNN training. However, we may want to modify our representation and model parameters by explicitly retraining with the weakly labeled data. To do this, we need to discover localization information from the image-level labels. Therefore, we will begin by solving a multiple instance learning (MIL) problem to discover the portion of each image most likely corresponding to the weak image-level label.

With the representation, ϕ , that has now been directly tuned for detection, we fix the parameter weights, $\phi(\cdot)$ and solve for the regions of interest in each weak labeled image. This corresponds to solving the following objective:

$$\mathcal{F} = \max_{I \in R_I} \min_{p \in \{C_W, I\}} \left[\Gamma(\mathbf{w}_p) + \alpha \sum_{I \in \mathcal{V}} \mathcal{F}(Y_I^p, \mathbf{w}_p) \right] \quad (6)$$

$$\mathcal{F} = \max_{I \in R_I} \mathbf{w}_p^T \phi(\mathbf{x}_I) \quad (7)$$

Note, we can decouple this optimization problem and independently solve for each category in our weakly labeled data set, $p \in C_W$. Let’s consider a single category p . Our goal is to minimize the loss for category p over images $I \in \mathcal{W}$. We will do this by considering two cases. First, if p is not in the weak label set of an image ($Y_I^p = -1$), then all regions in that image should be considered negative for category p . Second, if $Y_I^p = 1$, then we positively label a region \mathbf{x}_i if it has the highest confidence of containing object and negatively label all other regions. We perform the discovery of this top region in two steps. At first, we narrow down the set of candidate bounding boxes using the score, $\mathbf{w}_p^T \phi(\mathbf{x}_i)$, from our fixed representation and detectors from the previous optimization step. This set is then refined to estimate the most likely region to contain a positive instance in a Latent SVM formulation. The implementation details are discussed section 5.4.

Our final optimization step is to use the discovered bounding boxes from our weak dataset to refine our detectors and feature representation from the previous optimization step. This amounts to the subsequent step for minimization of the joint objective described in Equation (2). We collectively utilize the strong labels of images in S and estimated bounding boxes for the weakly labeled set, \mathcal{W} , to optimize for detector weights and feature representation, as follows:

$$\min_{\mathbf{w}_k, \phi} \sum_{k \in \{C, B\}} \left[\Gamma(\mathbf{w}_k) + \alpha \sum_{I \in \{\mathcal{W}, S\}} \sum_{r \in R_I} \ell(y_i^k, \mathbf{w}_k^T \phi(\mathbf{x}_i)) \right] \quad (8)$$

This is achieved by re-fine-tuning the CNN architecture. This final method is referred to as ‘*LSDA rep+joint fit*’ in our experiments.

Thus, the overall non-convex objective (2) is first approximated through initialization in (3). This initialization is then used to solve the sequential optimization problems defined in (4) and (6). Further, we present two ways to solve (6): k-NN based heuristic approach in (5) and MLL-based re-training approach in (7).

The sub-problem defined in (4) decreases the loss for strongly labeled categories and (8) decreases the loss for both weak-strong categories. Thus, this ensures that the overall objective (2) decreases. The refined detector weights and representation can be used to discover the bounding box annotations for weakly labeled data again, and this process can be iterated over (see Figure 2). We discuss re-training strategies and evaluate the contribution of this final optimization step in Section 5.5.

3.4 Detection with LSDA models

We now describe how our adapted network is used for detection at test time (depicted in Figure 3). For each test image we extract region proposals and generate $K + 1$ scores per region (similar to the R-CNN (Girshick et al., 2014) pipeline), one score for each category and an additional score for the background category. The score is generated by passing the properly warped image patch through our adapted representation layers and then through one of our proposed category specific adapted layers (described in the previous sections). Finally, for a given region, the score for category i is computed by linearly combining the per category score with the background score: $score_i - score_{Bg}$.

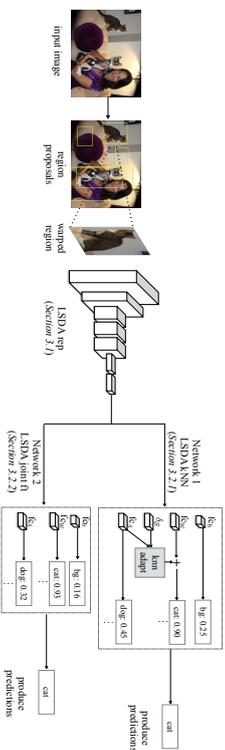


Figure 3: Detection with the LSDA network (test time). Given an image, extract region proposals, reshape the regions to fit into the network size and pass through our adapted network. Use the adapted representation and the category specific adaptation either through the no retraining nearest neighbor method or by retraining with our MLL based method. Finally produce detection scores per category for the region by considering background and category scores.

In contrast to the R-CNN (Girshick et al., 2014) model which trains SVMs on the extracted features from layer 7 and bounding box regression on the extracted features from layer 5, we directly use the final score vector to produce the prediction scores without either of the retraining steps. This choice results in a small performance loss, but offers the flexibility of being able to directly combine the classification portion of the network that has no detection labeled data, and reduces the training time from 3 days to roughly 5.5 hours.

4. Recognition Beyond Detection

In the previous section we outline an algorithm for producing weakly supervised detection models which label and coarsely localize objects in scene-centric images. While a bounding box around an object offers significantly more information than an image-level label, it is not sufficiently localized for tasks such as robotic manipulation and full scene parsing. Instead, we would like to produce semantic segmentation models which are capable of labeling each pixel in an image with the object category or background label.

Prior work has shown that convolutional networks can also be applied to arbitrary-sized inputs to allow for per-pixel spatial output. For example, Matan et al. (1992) augmented the LeNet digit classification model (LeCun et al., 1989), enabling recognition of strings of digits, and Wolf and Platt (1994) use networks to output 2-dimensional maps in order to identify the locations of postal address blocks. This technique has been used to produce semantic segmentation outputs of *C. elegans* (Ning et al., 2005) and more recently for generic object categories (Long et al., 2015). These “fully convolutional” networks can also be finetuned end-to-end on segmentation ground truth to produce fully supervised segmentation models (Long et al., 2015).

As we would like to produce pixel level labels from our LSDA model, we will build off of our recent work for object category semantic segmentation (Long et al., 2015). However, Long et al. (2015) requires full semantic segmentation (pixel-level) annotations to train the corresponding fully connected network. This form of supervision is particularly expensive to collect and in general very few data sources exist with these annotations.

Instead, we argue that much of the knowledge gained through training with pixel-level annotations can be transferred from the much weaker bounding box annotations. Therefore, we demonstrate that a reasonable semantic segmentation is possible by directly using detection parameters in a fully convolutional framework. Further, we show that even our weakly supervised detection models presented in the previous section are able to localize objects more precisely than a bounding box, despite never receiving pixel-level annotations and for many categories never even receiving bounding box annotations.

To produce such a network we take our final adapted LSDA model, which for the purpose of our experiments was trained using an AlexNet basic architecture (Krizhevsky et al., 2012), and we convert the model into the corresponding fully convolutional 32 stride network (FCN-32s) presented by Long et al. (2015). This amounts to relatively few changes to the network architecture. First, each input image is padded with 100 pixels before features are extracted. Next each of the three fully connected layers are converted into convolutional layers, where layer 6 has 4096 convolutions with 6×6 sized kernels, layer 7 has 4096 convolutions with 1×1 sized kernels, and the final score layer has $K + 1$ convolutions with 1×1 sized kernels (where K is the number of categories, plus one for background). Finally, additional deconvolution and crop layers are added which upsample the score map produced by the 8th layer (bilinear interpolation) and crops the pixel level score map to be the size of the input image. This means the final output of the network is a score per category per pixel, which allows us to perform semantic segmentation.

5. Experiments

To demonstrate the effectiveness of our approach we present quantitative results on the ILSVRC2013 detection dataset. The dataset offers images exhaustively labeled with bounding box annotations for 200 relevant object categories. The training set has $\sim 400K$ labeled images and on average 1.534 object classes per image. The validation set has 20K labeled images with $\sim 50K$ labeled objects. We simulate having access to weak labels for all 200 categories and having strong labels for only the first 100 categories (alphabetically sorted).

5.1 Experiment Setup & Implementation Details

We start by separating our data into classification and detection sets for training and a validation set for testing. Since the ILSVRC2013 training set has on average fewer objects per image than the validation set, we use this data as our classification data. To balance the categories we use ≈ 1000 images per class (200,000 total images). **Note:** for classification data we only have access to a single image-level annotation that gives a category label. In effect, since the training set may contain multiple objects, this single full-image label is a weak label, even compared to other classification training data sets. Next, we split the ILSVRC2013 validation set in half as (Girshick et al., 2014) did, producing two sets: `val1` and `val2`. To construct our detection training set, we take the images with bounding box labels from `val1` for only the first 100 categories (≈ 5000 images). Since the validation set is relatively small, we augment our detection set with 1000 bounding box labeled images per category from the ILSVRC2013 training set (following the protocol of (Girshick et al., 2014)). Finally we use the second half of the ILSVRC2013 validation set (`val2`) for our evaluation.

We implemented our CNN architectures and execute all fine-tuning using the open source software package Caffe (Jia et al., 2014) and have made our model definitions weights publicly available.

Train	Num images	395905
	Num objects	345854
Val	Num images	20121
	Num objects	55502

Table 1: Statistics of the ILSVRC13 detection dataset. Training set has fewer objects per image than validation set.

We use the ILSVRC13 detection dataset (Russakovsky et al., 2014) for our experiments. This dataset provides bounding box annotations for 200 categories. The dataset is separated into three pieces: `train`, `val`, `test` (see Table 1). The training images have fewer objects per image on an average than validation set images, so they constitute classification style data (Hoffman et al., 2014). Following prior work (Girshick et al., 2014), we use the further separation of the validation set into `val1` and `val2`. Overall, we use the `train` and `val1` set for our training data source and evaluate our performance of the data in `val2`.

Layers Adapted using Strongly Labeled Data	mAP (%) Weak Categories	mAP (%) All Categories
No Adapt (Classification Network)	10.31	11.90
f_{bgnd}	12.22	13.60
f_{bgnd}, f_{c6}	13.72	19.20
f_{bgnd}, f_{c7}	14.57	19.00
f_{bgnd}, f_{c5}	11.74	14.90
$f_{\text{bgnd}}, f_{c6}, f_{c7}$	14.20	20.00
$f_{\text{bgnd}}, f_{c6}, f_{c7}, f_{c5}$	14.42	20.40
$f_{\text{bgnd}}, \text{layers 1-7}, f_{c5}$	15.85	21.83

Table 2: Ablation study for different techniques for category independent adaptation of our model (LSDA rep only). We consider training with the first 100 (alphabetically) categories of the ILSVRC2013 detection validation set (on val) and report mean average precision (mAP) over the 100 weakly labeled categories (on val2). We find the best improvement is from fine-tuning all layers.

5.2 Quantitative Analysis of Adapted Representation

We evaluate the importance of each component of our algorithm through an ablation study. As a baseline, we consider training the network with only the weakly labeled data (no adaptation) and applying the network to the region proposals.

In Table 2, we present a detailed analysis of the different category independent adaptation techniques we could use to train the network. We call this method LSDA rep only. We find that the best category invariant adaptation approach is to learn the background category layer and adapt all convolutional and fully connected layers, bringing mAP on the weakly labeled categories from 10.31% up to 15.85% i.e. this achieves a 54% relative mAP boost over the classification only network. We later observe that the most important step of our algorithm proved to be adapting the feature representation, while the least important was adapting the category specific parameter. This fits with our intuition that the main benefit of our approach is to transfer category invariant information from categories with known bounding box annotation to those without the bounding box annotations.

We find that one of the biggest reasons our algorithm improves is from reducing localization error. For example, in Figure 4, we show that while the classification only trained net tends to focus on the most discriminative part of an object (ex: face of an animal) after our adaptation, we learn to localize the whole object (ex: entire body of the animal).

5.3 Error Analysis on Weakly Labeled Categories

We next present an analysis of the types of errors that our system (LSDA) makes on the weakly labeled object categories. First, in Figure 5, we consider three types of false positive errors: Loc (localization errors), BG (confusion with background), and Oth (other error types, which is essentially correctly localizing an object, but misclassifying it). After separating all false positives into one of these three error types we visually show the percentage

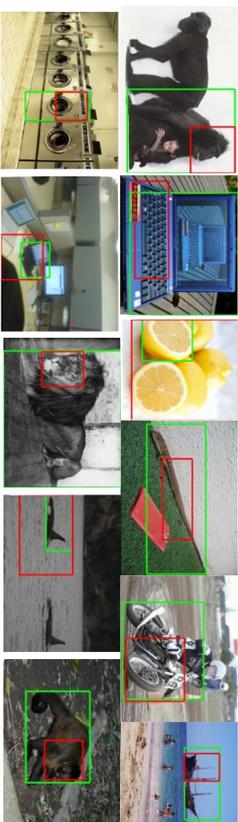


Figure 4: We show example detections on weakly labeled categories, for which we have **no detection training data**, where LSDA (shown with green box) correctly localizes and labels the object of interest, while the classification network baseline (shown in red) incorrectly localizes the object. This demonstrates that our algorithm learns to adapt the classifier into a detector which is sensitive to localization and background rejection.

of errors found in each type as you look at the top scoring 25-3200 false positives.¹ We consider the baseline of starting with the classification only network and show the false positive breakdown in Figure 5a. Note that the majority of false positive errors are confusion with background and localization errors. In contrast, after adapting the network using LSDA we find that the errors found in the top false positives are far less due to localization and background confusion (see Figure 5b). Arguably one of the biggest differences between classification and detection is the ability to accurately localize objects and reject background. Therefore, we show that our method successfully adapts the classification parameters to be more suitable for detection.

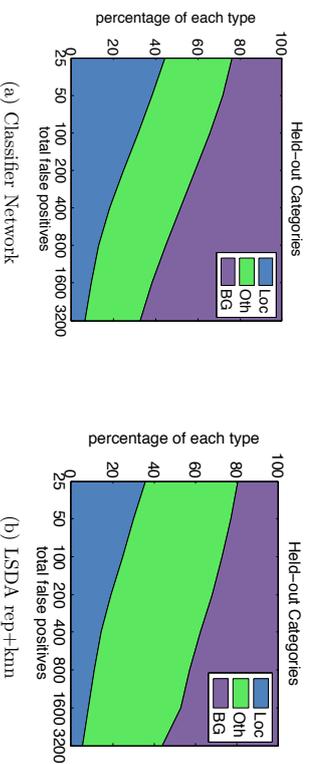


Figure 5: Comparison of error type breakdown on the categories which have no training bounding boxes available (weakly labeled data). After adapting all the layers in the network (LSDA), the percentage of false positive errors due to localization and background confusion is reduced (b) as compared to directly using the classification network for detection (a).

1. We modified the analysis software made available by Hoein et al. (2012) to work on ILSVRC-2013 detection

In Figure 6, we show examples of the top scoring O_{th} error types for LSDA on the weakly labeled data. This means the detector localizes an incorrect object type. For example, the motorcycle detector localized and mislabeled bicycle and the lemon detector localized and mislabeled an orange. In general, we noticed that many of the top false positives from the O_{th} error type were confusion with very similar categories. This is discussed in detail in next subsection.



Figure 6: Examples of the top scoring false positives from our LSDA rep+km network. Many of our top scoring false positives come from confusion with other categories.

5.4 Analysis of Discovered Boxes

We now analyze the quality of boxes discovered using adaptation of all layers including the background class. One of the key components of our system is using strong labels from auxiliary tasks to learn a representation where it’s possible to discover bounding boxes that correspond to the objects of interest in our weakly labeled data source. We begin our analysis by studying the bounding box discovery that our feature space enables, using selective search (Uijlings et al., 2013) to produce candidate regions. We optimize the bounding box discovery (Equations (6),(7)) using a one vs all Latent SVM formulation and optimize the formulation for AUC criterion (Bilen et al., 2014). This ensures that the top candidate regions chosen for joint fine-tuning have high precision. The feature descriptor used is the output of the fully connected layer, f_{cr} , of the CNN which is produced after fine-tuning the feature representation with strongly labeled data from auxiliary tasks. Following our alternating minimization approach, these discovered top boxes are then used to re-estimate the weights and feature representations of our CNN architecture.

	CorLoc over full dataset			Localization mAP (%)		
	ov=0.3	ov=0.5	ov=0.7	ov=0.9	ov=0.5	ov=0.5
Classification Network	29.63	26.10	24.28	23.43	13.13	
LSDA rep only	32.69	28.81	26.27	24.78	22.81	

Table 3: CorLoc over dataset and localization mAP (i.e. given the labels) performance of discovered bounding boxes in our weakly labeled training set (val1) of ILSVRC13 detection dataset. Comparison with varying amount of overlap with ground truth box. About 25% of our discovered boxes have an overlap of at least 0.9. Our method is able to significantly improve the quality of discovered boxes after incorporating strong labels from auxiliary tasks.

To evaluate the quality of discovered boxes, we do ablation study analyzing their overlap with ground truth which is measured using the standard intersection over union (IOU)

metric. Table 3 reports the CorLoc for varying overlapping thresholds. CorLoc across full dataset is defined as the accuracy of discovered boxes i.e. the accuracy that the box is correctly localized per image at different thresholds. Our optimization approach produces one positive bounding box per image with a weak label, and a discovered box is considered a true positive if it overlaps sufficiently with the ground truth box that corresponds to that label. Since each bounding box, once discovered, is considered an equivalent positive (regardless of score) for the purpose of retraining the ‘LSDA rep only’ model, this simple CorLoc metric is a good indication of the usefulness of our discovered bounding boxes. We note here that after re-training with our mined boxes the CorLoc will further improve, as indicated in the detection mAP reported in the next section. It is interesting that a significant fraction of discovered boxes have high overlap with the ground truth regions. For reference, we also computed the standard mean average precision over the discovered boxes for localization task i.e. when label is known. It is important to note that the improvement in localization mAP is much more significant than the CorLoc. This is because mAP is obtained by averaging over recall values, and the ‘LSDA rep only’ model achieves better overall recall than the ‘Classification Network’ model.

It is important to understand not only that our new feature space improves the quality of the resulting bounding boxes, but also what type of errors our method reduces. In Figure 7, we show the top 5 scoring discovered bounding boxes before and after modifying the feature space with strong labels from auxiliary tasks. We find that in many cases the improvement comes from better localization. For example without auxiliary strong labels we mostly discover the face of a lion rather than the body that we discover after our algorithm. Interestingly, there is also an issue with co-occurring classes. We are better able to localize “lion” body rather than the face. Most amazing results are for the “ping-pong” and “rugby” (second and third row) category where we are actually able to mine boxes for the racket and ball, while the classification net could only get the person boxes which is incorrect. Once we incorporate strong labels from auxiliary tasks we begin to be able to distinguish the person playing from the racket/ball itself. In the bottom row of Figure 7, we show the top 5 discovered bounding boxes for “tennis racket” where we are partially able to correct the images. Finally, there are some example discovered bounding boxes where we reduce quality after incorporating the strong labels from auxiliary tasks. For example, one of our strongly labeled categories is “computer keyboard”. Due to the strong training with keyboard images, some of our discovered boxes for “laptop” start to have higher scores on the keyboard rather than the whole laptop (see Figure 8). Also for the “water-craft” category, our adapted network ignores the mast but better localizes the boat itself, which slightly decreases the IOU of obtained box.

5.5 Detection Performance on ILSVRC13

Now that we have analyzed the intermediate result of our algorithm, we next study the full performance of our system. Figure 9 shows the mean average precision (mAP) percentage computed over the categories in val2 of ILSVRC13 for which we only have weakly labeled training data (categories 101-200). Previous method, LCL (Wang et al., 2014), detects in the standard weakly supervised setting – having no bounding box annotations for any of the 200 categories. This method also only reports results across all 200 categories on the full



Figure 7: Example discovered bounding boxes learned using our method. Left side shows the discovered boxes after fine-tuning with images in classification settings only, and right side shows the discovered boxes after fine-tuning with auxiliary strongly labeled dataset. We show top 5 discovered boxes across the dataset for corresponding category. Examples with a **green** outline are categories for which our algorithm was able to correctly discover bounding boxes of the object, while the feature space with only weak label training was not able to produce correct boxes. After incorporating the strong labels from auxiliary tasks, our method starts discovering “ping-pong” racket/ball and “rugby” ball, though still has some confusion with the person playing tennis. None of the discovered boxes from the original feature space correctly located racket/ball and instead included the person as well. In **yellow** we highlight the specific example of “tennis racket”, where some of the boxes get corrected not all top boxes.



Figure 8: Example discovered boxes of the category “laptop” where using auxiliary strongly labeled data causes bounding box discovery to diverge. *Left*: The discovered boxes obtained after fine-tuning with images in classification settings only. *Right*: The discovered boxes obtained after fine-tuning with the auxiliary strongly labeled dataset that contains the category “computer keyboard”. These boxes were low scoring examples, but we show them here to demonstrate a potential failure case – specifically, when one of the strongly labeled classes is a part of one of the weakly labeled classes. In the second example, adapted network better localizes the “water-craft” but misses the mast which decreases the IOU slightly.

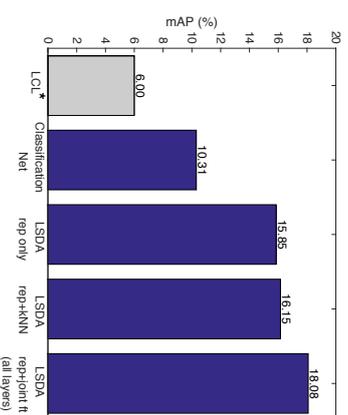


Figure 9: Comparison of mAP (%) for categories without any bounding box annotations (101-200 of val2) of ILSVRC13. The Joint representation and category-specific learning using MIL outperforms all other approaches. *As a reference we report the performance of LCL (Wang et al., 2014) which was computed across all 200 categories of the full validation set (val1+val2).

validation set. Our experiments indicate that the first 100 categories are easier on average than the second 100 categories, therefore the 6.0% mAP may actually be an upper bound of the performance of this approach. We also compare our algorithm against the scenario when the class-specific layer is adapted using nearest neighbors across all categories (LSDA rep+km). The joint representation and multiple instance learning approach achieves the highest results (LSDA rep+joint ft).

We next consider different re-training strategies for learning new features and detection weights after discovering the bounding boxes in the weakly labeled data. Table 4 reports the

Category Specific Adaptation Strategy	mAP (%) Weak Categories	mAP (%) All Categories
LSDA rep only	15.85	21.83
LSDA rep+kNN (k=5)	15.97	22.05
LSDA rep+kNN (k=10)	16.15	22.05
LSDA rep+kNN (k=100= fcs)	15.96	21.94
LSDA rep+joint fit (fcw)	17.01	22.43
LSDA rep+joint fit (all layers)	18.08	22.74
Baseline: Classification Network	10.31	11.90
Oracle: RCNN Full Detection Network	26.25	28.00

Table 4: Comparison of different ways to re-train after discovery of bounding boxes. We show mAP on val2 set from ILSVRC13. We find that the most effective way to re-train with discovered boxes is to modify the detectors and the feature representation.

mean average precision (mAP) percentage for no re-training (directly using the feature space learned after incorporating the strong labels), LSDA rep only, no retraining but last layer weights of weak categories adapted using nearest neighbors, LSDA rep+knn, re-training only the category-specific detection parameters, LSDA rep+joint fit (fcw), and retraining feature representations jointly with category-specific weights, LSDA rep+joint fit (all layers). In our experiments the improved performance is due to the first iteration of the overall algorithm. We find that the best approach is to jointly learn to refine the feature representation and the category-specific detection weights. More specifically, we learn a new feature representation by fine-tuning all fully connected layers in the CNN architecture. The last row shows the performance achievable by our detection network if it had access to bounding box annotated data for all 200 categories, and serves as a performance upper bound.² Our method achieves **18.08%** mAP on weakly labeled categories as compared to **10.31%** of baseline, but it is still significantly lower than fully-supervised oracle which gives **26.25%**.

We finally analyze examples where our full algorithm which jointly learns representation and class-specific layer using MIL (LSDA rep+joint fit) outperforms the previous approach where only representation is adapted without joint learning over weak labels (LSDA rep+knn). Figure 10 shows a sample of the types of errors our algorithm improves on. These include localization errors, confusion with other categories, and interestingly, confusion with co-occurring categories. In particular, our algorithm provides improvement when searching for a small object (ball or helmet) in a sports scene. Training only with weak labels causes the previous state-of-the-art to confuse the player and the object, resulting in a detection that includes both. Our algorithm is able to localize only the small object and recognize that the player is a separate object of interest.

2. To achieve R-CNN performance requires additionally learning SVMs on the activations of layer 7 and bounding box regression on the activations of layer 5. Each of these steps adds between 1-2mAP at high computation cost and using the SVMs removes the adaptation capacity of the system.

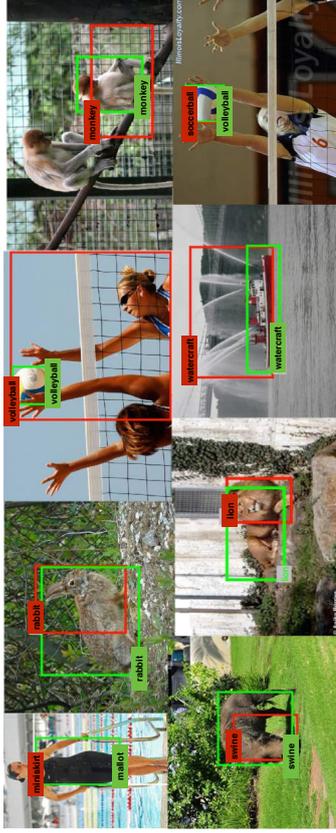


Figure 10: Examples where our algorithm after joint MIL adaptation (LSDA rep+joint fit) outperforms the representation only adaptation (LSDA rep only). We show the top scoring detection from LSDA rep only with a Red box and label, and the top scoring detection from LSDA rep+joint fit, as a Green box and label. Our algorithm improves localization (ex: rabbit, lion etc), confusion with other categories (ex: miniskirt vs maillot), and confusion with co-occurring classes (ex: volleyball vs volleyball player)

5.6 Large Scale Detection

To showcase the capabilities of our technique we produced a 7604 category detector. The first categories correspond to the 200 categories from the ILSVRC2013 challenge dataset which have bounding box labeled data available. The other 7404 categories correspond to leaf nodes in the ImageNet database and are trained using the available full image weakly labeled classification data. We trained a full detection network using the 200 strongly labeled categories and trained the other 7404 last layer nodes using only the weak labels. Note, the ImageNet dataset does contain other non-exhaustively labeled images for around 3000 object categories, 1825 of which overlap with the 7404 leaf node categories in our model. We do not use these labels during training of our large scale model. Quantitative evaluation for these categories is difficult to compute since they are not exhaustively labeled, however a followup work by Mrowca et al. (2015) evaluated F1 score of our model for the few object instances labeled per image to be 9.59%. Also note that while we have no bounding box annotations for the 7404 fine-grained categories, some may be related to the 200 basic level categories for which we use bounding box data to train – for example a particular breed of dog from 7404 weakly labeled data while ‘dog’ appears in the 200 strongly labeled categories.

We show qualitative results of our large scale detector by displaying the top detections per image in Figure 11. The results are filtered using non-max suppression across categories to only show the highest scoring categories.

The main contribution of our algorithm is the joint representation and multiple instance learning approach for modifying a convolutional neural network for detection. However, the choice of network and how the net is used at test time both effect the detection time



Figure 11: Example top detections from our 7604 category detector. Detections from the 200 categories that have bounding box training data available are shown in blue. Detections from the remaining 7404 categories for which only weakly labeled data is available are shown in red.

computation. We have therefore also implemented and released a version of our algorithm running with fast region proposals (Krishenbühl and Kohm, 2014) on a spatial pyramid pooling network (He et al., 2014), reducing our detection time down to half a second per image (from 4s per image) with nearly the same performance. We hope that this will allow the use of our 7.6K model on large data sources such as videos. We have released the 7.6K model and code to run detection (both the way presented in this paper and our faster version) at lsda.berkeleyvision.org.

5.7 Fully Convolutional LSDA for Semantic Segmentation

Bounding boxes localize objects to an inherently limited degree. While the system presented so far produces remarkably accurate bounding boxes from weak training labels, it does not address the ultimate goal of knowing exactly which pixels correspond to which objects.

Segmentation ground truth is unavailable for all but a few of the 7604 categories in our large scale detector, and segmentations are even more costly to annotate than bounding boxes. Nevertheless, as described in Section 5.7, we can convert our detection-adapted network into a fully-convolutional model following Long et al. (2015) and produce dense outputs for each of the 7604 categories plus 1 for background. We call this model LSDA7k FCN-32s since we use the 32 stride version of the fully convolutional networks proposed in Long et al. (2015). We next evaluate our semantic segmentation model using the PASCAL dataset Everingham et al. (2010) and the following metrics.

Metrics We compute both the commonly used mean intersection over union (mean IU) metric for semantic segmentation as well as three other metrics used by Long et al. (2015). The metrics are defined below, where n_{ij} denotes the number of pixels from class i predicted to belong to class j so that the number of pixels belonging to class i are $m_i = \sum_j n_{ij}$, and K denotes the number of classes.

- pixel accuracy: $\sum_i n_{ii} / \sum_i m_i$
- mean accuracy: $1/K \sum_i n_{ii} / t_i$
- mean IU: $1/K \sum_i n_{ii} / (m_i + \sum_j n_{ji} - n_{ii})$

- frequency weighted IU: $(\sum_i m_i)^{-1} \sum_i m_i n_{ii} / (m_i + \sum_j n_{ji} - n_{ii})$

We would like to understand how well our model can localize weakly trained objects so for each of the PASCAL 20 object categories we manually find the set of fine-grained categories from the 7404 weakly labeled leaf nodes in ImageNet that correspond to that category. Since layer 8 of our LSDA7k FCN-32s network produces 7605 outputs per region of the image, we insert an additional mapping layer which for each category c is the maximum score across all weakly labeled categories which correspond to that PASCAL category. Next, this reduced score map where each image region now has 21 scores is run through the deconvolution layer to produce the corresponding PASCAL per pixel scores. Finally, for each pixel we choose a label based on which of the categories or background has the highest pixel score.

We report results on both the PASCAL 2011 and 2012 validation sets. Note, our method was not trained on any PASCAL images and in general was trained for classification of 7404 fine-grained categories and then adapted using our algorithm for detection. Additionally, our model is trained using the AlexNet architecture while most state-of-the-art semantic segmentation models are trained using the larger VGG network (Simonyan and Zisserman, 2014).

For the PASCAL 2011 validation set, shown in Table 5, we first compare against the classification model trained for the 7404 category full image labels. We run this model fully convolutionally using the FCN-32s approach (AlexNet) and report the segmentation performance in the first row as *Classification 7K FCN-32s (AlexNet)*. This method gives a baseline for our LSDA approach which uses this model as the initialization prior to our adaptation approach. Next, we compare against the reported performance of the weakly trained models of Pathak et al. (2014) and for reference, the fully supervised AlexNet and VGG FCN-32s presented by Long et al. (2015). We report all four metrics for our work and report all available metrics for competing works. We see that our weakly trained model outperforms the baseline classification model run fully convolutionally and almost reaches the performance of the MIU-FCN method which uses the higher capacity VGG model and trains specifically for the segmentation task.

The per-category results of our method on the PASCAL 2012 validation set as compared to two state-of-the-art weakly trained semantic segmentation models is shown in Table 6. Not surprisingly, our LSDA7k FCN-32s underperforms these methods. No doubt adding the multiple instance loss of Pathak et al. (2014) or the object constraints of Pathak et al. (2015), while training directly on the PASCAL dataset would further improve our method. The purpose of these experiments is to give the reader an accurate picture of how well our large scale model performs at pixel level annotation without any tuning to the new situation.

We next show qualitative segmentation results across the fine-grained 7404 categories of our LSDA7k FCN-32s network in Figure 12³ and compare against the baseline Classification 7K FCN-32s network. We find that often the segmentation masks from our LSDA network are more precise (see “American egret” example) and the top scoring predicted class is often more accurately labeled. For example, the bottom image is labeled as “air conditioner” by the classification network and correctly as “venetian blind” by our network. These category models were trained without ever seeing any associated

3. The full network without the mapping layer to pascal 20 categories.

adaptation algorithm. Given the significant improvement on the weakly labeled categories, our algorithm enables detection of tens of thousands of categories. We produce a 7.6K

category detector and have released both code and models at lsda.berkeleyvision.org.

Our approach significantly reduces the overhead of producing a high quality detector. We hope that in doing so we will be able to minimize the gap between having strong large-scale classifiers and strong large-scale detectors. Further we show that large-scale detectors can be used to produce large-scale semantic segmenters. We present semantic segmentation performance for the large scale model on PASCAL VOC with a manual mapping from the 7404 weakly labeled object categories to the 20 categories in the PASCAL dataset. For future work we would like to experiment with incorporating some pixel-level annotations for a few object categories. Our intuition is that by doing so we will be able to further improve our large-scale models with minimal extra supervision.

References

- B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *Proc. CVPR*, 2010.
- K. Ali and K. Saenko. Confidence-rated multiple instance boosting for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.
- S. Andrews, I. Tsochanantaris, and T. Hofmann. Support vector machines for multiple-instance learning. In *Proc. NIPS*, pages 561–568, 2002.
- Y. Aytar and A. Zisserman. Tabula rasa: Model transfer for object category detection. In *ICCV*, 2011.
- Y. Aytar and A. Zisserman. Enhancing exemplar svms using part level transfer regularization. In *British Machine Vision Conference*, 2012.
- Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *In Proc. ICML*, 2009.
- A. Berg, J. Deng, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge, 2012. URL <http://www.image-net.org/challenges/LSVRC/2012/>.
- H. Bilal, V. P. Namboodiri, and L. J. Van Gool. Object and action classification with latent window parameters. *ICCV*, 106(3):237–251, 2014.
- D. Borth, R. Ji, T. Chen, T. Breuel, and S. F. Chang. Large-scale visual sentiment ontology and detectors using adjective noun pairs. In *ACM Multimedia Conference*, 2013.
- R. G. Cimbis, J. Verbeek, C. Schmid, et al. Multi-fold ml training for weakly supervised object localization. In *CVPR*, 2014.
- N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *In Proc. CVPR*, 2005.
- T. Deselaers, B. Alexe, and V. Ferrari. Weakly supervised localization and learning with generic knowledge. *ICV*, 2012.
- T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 1997.
- J. Donahue, J. Hoffman, E. Rodner, K. Saenko, and T. Darrell. Semi-supervised domain adaptation with instance constraints. In *Computer Vision and Pattern Recognition (CVPR)*, 2013.
- J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. In *Proc. ICML*, 2014.
- L. Duan, D. Xu, and I. W. Tsang. Learning with augmented features for heterogeneous domain adaptation. In *Proc. ICML*, 2012.
- M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *ICCV*, 88(2):303–338, June 2010.
- P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. PAMI*, 32(9):1627–1645, 2010.
- B. Fernando, A. Habrard, M. Sebban, and T. Thyrtelaers. Unsupervised visual domain adaptation using subspace alignment. In *Proc. ICCV*, 2013.
- P. Fischer, A. Dosovitskiy, and T. Brox. Descriptor matching with convolutional neural networks: a comparison to sift. *ArXiv e-prints*, abs/1405.5769, 2014.
- C. Gallgullies, B. Babenko, A. Rabinovich, and S. Belongie. Weakly supervised object localization with stable segmentations. In *ECCV*, 2008.
- Y. Ganin and V. Lempitsky. Unsupervised Domain Adaptation by Backpropagation. In *ICML*, 2015.
- M. Ghifary, W. B. Kleijn, and M. Zhang. Domain adaptive neural networks for object recognition. *CoRR*, abs/1409.6041, 2014. URL <http://arxiv.org/abs/1409.6041>.
- R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *In Proc. CVPR*, 2014.
- D. Goehringer, J. Hoffman, E. Rodner, K. Saenko, and T. Darrell. Interactive adaptation of real-time object detectors. In *ICRA*, 2014.
- B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Proc. CVPR*, 2012.
- M. Guillaumin and V. Ferrari. Large-scale knowledge transfer for object localization in imagenet. In *CVPR*, pages 3202–3209, June 2012. doi: 10.1109/CVPR.2012.6248055.
- M. Guillaumin, D. Ktuel, and V. Ferrari. Imagenet auto-annotation with segmentation propagation. *ICCV*, 110(3):328–348, 2014. ISSN 0920-5691. doi: 10.1007/s11263-014-0713-9. URL <http://dx.doi.org/10.1007/s11263-014-0713-9>.
- K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *In Proc. ECCV*, 2014.
- D. Hoem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In *In Proc. ECCV*, 2012.
- J. Hoffman, E. Rodner, J. Donahue, K. Saenko, and T. Darrell. Efficient learning of domain-invariant image representations. In *Proc. ICML*, 2013a.

- J. Hoffman, E. Tzeng, J. Donahue, Y. Jia, K. Saenko, and T. Darrell. One-shot adaptation of supervised deep convolutional models. *CoRR*, abs/1312.6204, 2013b. URL <http://arxiv.org/abs/1312.6204>.
- J. Hoffman, S. Guadarrama, E. Tzeng, R. Hu, J. Donahue, R. Girshick, T. Darrell, and K. Saenko. LSDA: Large scale detection through adaptation. In *Neural Information Processing Systems (NIPS)*, 2014.
- J. Hoffman, D. Pathak, T. Darrell, and K. Saenko. Detector discovery in the wild: Joint multiple instance and representation learning. In *CVPR*, 2015.
- Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.
- P. Krähenbühl and V. Koltun. Geodesic object proposals. In *In Proc. ECCV*, 2014.
- A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proc. NIPS*, 2012.
- B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Proc. CVPR*, 2011.
- M. P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In *In Proc. NIPS*, 2010.
- Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1989.
- J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *CVPR*, November 2015.
- M. Long and J. Wang. Learning transferable features with deep adaptation networks. In *ICML*, 2015.
- D. G. Lowe. Distinctive image features from scale-invariant key points. *IJCV*, 2004.
- O. Matan, C. J. Burgess, Y. L. Cun, and J. S. Denker. Multi-digit recognition using a space displacement neural network. In *Neural Information Processing Systems*, pages 488–495. Morgan Kaufmann, 1992.
- D. Mrowca, M. Rohrbach, J. Hoffman, R. Hu, K. Saenko, and T. Darrell. Spatial semantic regularization for large scale object detection. In *ICCV*, 2015.
- F. Ning, D. Delhomme, Y. LeCun, F. Piano, L. Bottou, and P. E. Barbano. Toward automatic phenotyping of developing embryos from videos. In *IEEE Transactions on Image Processing*, pages 14(9):1360–1371, 2005.
- M. Pauley and S. Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *Proc. ICCV*, 2011.
- G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille. Weakly- and semi-supervised learning of a dcn for semantic image segmentation. *CoRR*, abs/1502.02734, 2015.
- D. Pathak, E. Shelhamer, J. Long, and T. Darrell. Fully convolutional multi-class multiple instance learning. *CoRR*, abs/1412.7144, 2014. URL <http://arxiv.org/abs/1412.7144>.
- D. Pathak, P. Krähenbühl, and T. Darrell. Constrained convolutional neural networks for segmentation. In *ICCV*, 2015.
- O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. K. and Michael Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. arXiv:1409.0575, 2014.
- K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *Proc. ECCV*, 2010.
- P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013.
- K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
- S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*, 2012.
- P. Siva, C. Russell, and T. Xiang. In defence of negative mining for annotating weakly labelled data. In *ECCV*, 2012.
- P. Siva, C. Russell, T. Xiang, and L. Agapito. Looking beyond the image: Unsupervised learning for object saliency and detection. In *Proc. CVPR*, 2013.
- H. Song, R. Girshick, S. Jegelka, J. Mairal, Z. Harchaoui, and T. Darrell. On learning to localize objects with minimal supervision. In *ICML*, 2014a.
- H. O. Song, Y. J. Lee, S. Jegelka, and T. Darrell. Weakly-supervised discovery of visual pattern configurations. In *Proc. NIPS*, 2014b.
- E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *CoRR*, abs/1412.3474, 2014. URL <http://arxiv.org/abs/1412.3474>.
- J. Uijlings, K. van de Saude, T. Gevers, and A. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013.
- A. Vezhnevets and V. Ferrari. Associative embeddings for large-scale knowledge transfer with self-assessment. *CVPR*, June 2014.
- C. Wang, W. Ren, K. Huang, and T. Tan. Weakly supervised object localization with latent category learning. In *European Conference on Computer Vision (ECCV)*, 2014.
- R. Wolf and J. C. Platt. Postal address block location using a convolutional locator network. In *Advances in Neural Information Processing Systems 6*, pages 745–752. Morgan Kaufmann Publishers, 1994.
- J. Xu, S. Ramos, D. Vázquez, and A. López. Domain adaptation of deformable part-based models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, In Press, 2014.
- J. Yang, R. Yan, and A. Hauptmann. Adapting SVM classifiers to data with shifted distributions. In *ICDM Workshops*, 2007a.
- J. Yang, R. Yan, and A. G. Hauptmann. Cross-domain video concept detection using adaptive svms. *ACM Multimedia*, 2007b.

LARGE SCALE VISUAL RECOGNITION THROUGH ADAPTATION

- C.-N. J. Yu and T. Joachims. Learning structural svm with latent variables. In *Proc. ICML*, pages 1169–1176, 2009.
- A. L. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 15(4):915–936, 2003.
- C. Zhang, J. C. Platt, and P. A. Viola. Multiple instance boosting for object detection. In *Advances in neural information processing systems*, 2005.

Covariance-based Clustering in Multivariate and Functional Data Analysis

Francesca Ieva

*Department of Mathematics “F. Enriques”
Università degli Studi di Milano
Via Cesare Saldini 50, 20133 Milano, Italy*

FRANCESCA.IEVA@UNIMI.IT

Anna Maria Paganoni

*Nicholas Tarabelloni
MOX – Modeling and Scientific Computing
Department of Mathematics
Politecnico di Milano
Via Bonardi 9, 20133 Milano, Italy*

ANNA.PAGANONI@POLIMI.IT

NICHOLAS.TARABELLONI@POLIMI.IT

Editor: Sara van de Geer

Abstract

In this paper we propose a new algorithm to perform clustering of multivariate and functional data. We study the case of two populations different in their covariances, rather than in their means. The algorithm relies on a proper quantification of distance between the estimated covariance operators of the populations, and subdivides data in two groups maximising the distance between their induced covariances. The naive implementation of such an algorithm is computationally forbidding, so we propose a heuristic formulation with a much lighter complexity and we study its convergence properties, along with its computational cost. We also propose to use an enhanced estimator for the estimation of discrete covariances of functional data, namely a linear shrinkage estimator, in order to improve the precision of the clustering. We establish the effectiveness of our algorithm through applications to both synthetic data and a real data set coming from a biomedical context, showing also how the use of shrinkage estimation may lead to substantially better results.

Keywords: Clustering, covariance operator, operator distance, shrinkage estimation, functional data analysis

1. Introduction

The goal of performing clustering of data, in order to point out groups of observations based on some notion of similarity, has been of primary interest in applied statistics since ages. Literature is plenty of methods focusing their attention on the aggregation and separation of a sample into groups depending on similarities in locations of data (e.g., hierarchical clustering, *k*-means, PAM; see for instance Hartigan, 1975). Considerably less work can be found on methods attaining the clustering entirely on the basis of differences in the covariance structures of random models generating data. This target is not trivial, and less easy to translate into practice, since it calls for a proper quantification of differential correlation or distances between covariances of data. Nevertheless, it might happen to analyse groups

of data that are scarcely distinguishable in terms of locations, while showing differences in their variability.

Examples can be found in biostatistics where, for instance, the dichotomy between physiological and pathological features often shows an interesting change in pattern of variability. Also, this is of great interest in genomics, where instead of focusing on gene expression levels, one could be interested in finding different correlation structures among subsets of data. This is the core task in the analysis of the differential co-expression of genes, namely the differential correlation structure among expression levels in different subsets of experimental conditions. In (Watson, 2006), for instance, the author proposes a method to identify groups of genes that are correlated in a first group of microarrays, and are not in a second one. In (Mitra et al., 2016), the authors provide a more complex modeling strategy that is able to specify the differential dependence structure by using Bayesian graphical models, and in (Cai and Zhang, 2016) authors provide, within a supervised framework, a way to estimate the differential correlation matrices of data belonging to two differentially expressed groups.

In this paper we tackle the problem from a different point of view, and focus on differences between global covariance structures of data belonging to two unknown groups, which will also be identified. We focus on the specific case of a set of observations from two populations whose probability distributions have the same mean but differ in terms of covariances. The method we propose can be applied both to the traditional case of random vectors, and to the recently developed setting of functional data, arising as outcomes of infinite-dimensional stochastic processes (see, for instance, the monographs Ramsay and Silverman, 2005; Horváth and Kokoszka, 2012). We will introduce the method according to the latter case.

In particular, we first introduce a suitable notion of distance between covariance operators, i.e. the functional generalisation of covariance matrices, which is the instrument we use to measure dissimilarities. Then we make use of such distance to search, among two-class partitions of data, the one maximising the distance between the class-specific covariances, under the assumption that, if the two populations can be distinguished from their covariances, this would be the most likely subdivision detecting the true groups.

A naive implementation of this algorithm, involving an exhaustive sampling strategy inside the set of subsets of data, would face a combinatorial complexity with respect to the number of observations, thus forbidding the analysis of data sets with common sizes. We therefore transform the method into a heuristic, greedy algorithm, with greatly reduced complexity, which can be efficiently implemented and effectively applied.

Due to its construction, our algorithm benefits from the accuracy of the estimation of covariances. Owing to the typically large dimensionality (compared to the number of data available) of discrete approximations of functional observations, covariance estimation through classical sample estimators may be non-optimal. To remedy this shortcoming, we propose to replace standard, unbiased covariance estimator with a shrinkage estimator with enhanced accuracy properties (see, for instance, Ledoit and Wolf, 2003, 2004 and Schafer and Strimmer, 2005). We show through experiments that this choice leads to a substantially improved clustering.

The paper is organised as follows: in Section 2 we briefly recall some properties of covariance operators for functional data. In Section 3 we introduce the new clustering method for two groups of data which differ in variance-covariance structures, we derive its heuristic formulation and describe the shrinkage strategy we used to enhance the estimation performances. In Section 4 we assess the clustering performances through the application to both synthetic and real data sets. Discussion and conclusions are presented in Section 5.

2. Covariance Operators for Functional Data

Whenever our data can be interpreted as finite-dimensional samples of quantities that are intrinsically dependent on some continuous variable, such as time, we may resort to the model of functional data (see, for instance, Ramsay and Silverman, 2005; Horváth and Kokoszka, 2012). At its core is the assumption that data are sample measurements of trajectories of stochastic processes valued in suitable function spaces.

In the following we recall the definition of covariance operator for functional data, along with its most important properties (for more details see, e.g., Bosq, 2000). Let \mathcal{X} be a stochastic process taking values in $L^2(I)$, with $I \subset \mathbb{R}$ a compact interval, having mean function $\mathbb{E}[\mathcal{X}] = \mu$ and such that $\mathbb{E}\|\mathcal{X}\|^2 < \infty$, where we denote by $\|\cdot\|$ the $L^2(I)$ norm induced by the scalar product $\langle \cdot, \cdot \rangle$. Without loss of generality we can assume $\mu = 0$ and define the following covariance operator $\mathcal{C} \in \mathcal{L}(L^2(I); L^2(I))$:

$$\langle y, \mathcal{C}x \rangle = \mathbb{E}[\langle x, \mathcal{X} \rangle \langle y, \mathcal{X} \rangle], \quad \forall x, y \in L^2(I). \quad (1)$$

\mathcal{C} is a compact, self-adjoint, positive semidefinite linear operator between $L^2(I)$ and $L^2(I)$. Therefore it can be decomposed into:

$$\mathcal{C} = \sum_{k=1}^{\infty} \lambda_k e_k \otimes e_k, \quad (2)$$

where \otimes indicates an outer product in $L^2(I)$, $\{e_k\}_{k=1}^{\infty}$ is the sequence of orthonormal eigenfunctions, forming a basis of $L^2(I)$, and $\{\lambda_k\}_{k=1}^{\infty}$ is the sequence of eigenvalues. We assume that eigenvalues are sorted in decreasing order, so that:

$$\lambda_1 \geq \lambda_2 \geq \dots \geq 0.$$

By expressing \mathcal{X} with respect to the eigenfunctions basis, $\mathcal{X} = \sum_{k=1}^{\infty} \xi_k e_k$, it holds

$$\lambda_k = \langle e_k, \mathcal{C}e_k \rangle = \mathbb{E}[\xi_k^2],$$

thus, the covariance operator is nuclear, meaning that

$$\mathbb{E}\|\mathcal{X}\|^2 = \sum_{k=1}^{\infty} \lambda_k = \sum_{k=1}^{\infty} |\lambda_k| < \infty.$$

\mathcal{C} is also a Hilbert-Schmidt operator (see, for instance, Bosq, 2000), since it holds:

$$\sum_{k=1}^{\infty} \lambda_k^2 < \infty. \quad (3)$$

We equip the space of Hilbert-Schmidt operators with the Hilbert-Schmidt norm, defined as $\|\mathcal{U}\|_S^2 = \sum_{k=1}^{\infty} \lambda_k^2$, where $\{\lambda_k\}_{k=1}^{\infty}$ are the eigenvalues of \mathcal{U} . This is induced by the following scalar product:

$$\langle \mathcal{U}, \mathcal{V} \rangle_S = \sqrt{\text{Tr}(\mathcal{U} - \mathcal{V})(\mathcal{U} - \mathcal{V})^*}, \quad (4)$$

where $\text{Tr}(\cdot)$ denotes the trace operator, and \mathcal{U}^* is the Hilbertian adjoint of \mathcal{U} , i.e.,

$$\langle \mathcal{U}(x), y \rangle = \langle x, \mathcal{U}^*(y) \rangle \quad \forall x, y \in L^2(I).$$

The space of Hilbert-Schmidt operators on $L^2(I)$, endowed with the scalar product (4) and the associated norm, becomes a separable Hilbert space itself.

Within this theoretic framework, a natural definition of dissimilarity between Hilbert-Schmidt operators (among which are covariance operators) may be the Hilbert-Schmidt distance:

$$d(\mathcal{U}, \mathcal{V}) = \|\mathcal{U} - \mathcal{V}\|_S^2 = \sum_{k=1}^{\infty} \eta_k^2, \quad (5)$$

where $\{\eta_k\}_{k=1}^{\infty}$ is the sequence of eigenvalues of $\mathcal{U} - \mathcal{V}$.

3. Covariance-based Clustering

We face now the problem of classifying observations belonging to two different functional populations. Let \mathcal{X} and \mathcal{Y} be stochastic processes on $L^2(I)$ generated by the laws $\mathcal{P}_{\mathcal{X}}$ and $\mathcal{P}_{\mathcal{Y}}$. We imagine to have a set of N data in some data set D (i.e., a collection of observations) composed in the following way by an equal number of observations from two families: $D = \{X_1, \dots, X_K, Y_1, \dots, Y_K\}$ with $K = N/2$ and where $\{\mathcal{X}_j\}_{j=1}^K$ are i.i.d and follow respectively $\mathcal{P}_{\mathcal{X}}$ and $\mathcal{P}_{\mathcal{Y}}$. We introduce the following quantities:

$$\begin{aligned} \mu_1 &= \mathbb{E}[X_1], & C_1 &= \mathbb{E}[X_i \otimes X_i], & \forall i &= 1, \dots, K, \\ \mu_2 &= \mathbb{E}[Y_1], & C_2 &= \mathbb{E}[Y_j \otimes Y_j], & \forall j &= 1, \dots, K. \end{aligned}$$

Let us consider the vector of indexes of units constituting the two populations in D :

$$I^{(0)} = \left(\underbrace{1, 2, \dots, K}_{I_1^{(0)}}, \underbrace{K+1, \dots, N}_{I_2^{(0)}} \right), \quad (6)$$

which is unique, provided we don't distinguish among permutations of sub-intervals $I_1^{(0)}$ and $I_2^{(0)}$. In the following we shall consider recombinations of these indexes into two subsets:

$$I^{(i)} = \left(I_1^{(i)}, I_2^{(i)} \right), \quad i \in \{1, \dots, N_C\}, \quad (7)$$

where $I^{(i)}$ denotes the i -th combination out of $N_C = \binom{N}{K}$, however enumerated.

The sample estimators of means and covariance operators induced by this subdivision are denoted, respectively, with $\hat{\mu}_1^{(i)}, \hat{\mu}_2^{(i)}$ and $\hat{C}_1^{(i)}, \hat{C}_2^{(i)}$. We point out that, when $i = 0$, we recover the estimators of μ_1, μ_2 and C_1 and C_2 . For this reason we rename the latter quantities as

$\mu_1^{(0)}, \mu_2^{(0)}$, and $\mathcal{C}_1^{(0)}, \mathcal{C}_2^{(0)}$.

Our clustering method is based on the following, crucial assumption:

Assumption 1 *We assume that observations drawn from families $P_{\mathcal{X}}$ and $P_{\mathcal{Y}}$ constituting the data set D may be distinguished on the basis of their covariances, but not of their means, i.e. $\mu_1^{(0)} = \mu_2^{(0)}$ and $\mathcal{C}_1^{(0)} \neq \mathcal{C}_2^{(0)}$, and therefore $\|\mu_1^{(0)} - \mu_2^{(0)}\| = 0$ and $d(\mathcal{C}_1^{(0)}, \mathcal{C}_2^{(0)}) \gg 0$.*

As a consequence of this assumption we conveniently center data and assume they have zero means.

In order to illustrate the clustering method we propose, let us consider a situation where the original data set has been split according to a vector of indices $I^{(g)} = [I_1^{(g)}, I_2^{(g)}]$. For the sake of simplicity, let us encode this through the binary variables $w_{j,g} = \mathbb{1}(X_j \in I_g^{(g)})$, $j = 1, \dots, K$, $g = 1, 2$, and $v_{j,g} = \mathbb{1}(Y_j \in I_g^{(g)})$, $j = 1, \dots, K$, $g = 1, 2$. In other words, such variables express the fact that observation j from the original population \mathcal{X} or \mathcal{Y} belongs to the first ($I_1^{(g)}$) or second ($I_2^{(g)}$) group into which data are split. According to the setting previously introduced, it is:

$$\begin{aligned} K &= \sum_{g=1}^2 \sum_{j=1}^K w_{j,g}, & K &= \sum_{j=1}^2 \sum_{g=1}^K v_{j,g}, \\ K &= \sum_{j=1}^K w_{j,1} + \sum_{j=1}^K v_{j,1}, & K &= \sum_{j=1}^K w_{j,2} + \sum_{j=1}^K v_{j,2}. \end{aligned}$$

Then, we can re-write the sample covariances $\widehat{\mathcal{C}}_1^{(g)}$ and $\widehat{\mathcal{C}}_2^{(g)}$ as:

$$\begin{aligned} \widehat{\mathcal{C}}_1^{(g)} &= \frac{\sum_{j=1}^K w_{j,1} X_j \otimes X_j + \sum_{k=1}^K v_{k,1} Y_k \otimes Y_k}{K}, \\ \widehat{\mathcal{C}}_2^{(g)} &= \frac{\sum_{j=1}^K w_{j,2} X_j \otimes X_j + \sum_{k=1}^K v_{k,2} Y_k \otimes Y_k}{K}. \end{aligned}$$

If we compute the difference $\widehat{\mathcal{C}}_1^{(g)} - \widehat{\mathcal{C}}_2^{(g)}$, we obtain:

$$\widehat{\mathcal{C}}_1^{(g)} - \widehat{\mathcal{C}}_2^{(g)} = \frac{1}{K} \sum_{j=1}^K (w_{j,1} - w_{j,2}) X_j \otimes X_j + \frac{1}{K} \sum_{k=1}^K (v_{k,1} - v_{k,2}) Y_k \otimes Y_k,$$

hence, exploiting the distance between covariance operators:

$$\begin{aligned} K^2 \|\widehat{\mathcal{C}}_1^{(g)} - \widehat{\mathcal{C}}_2^{(g)}\|_S^2 &= \left\| \sum_{j=1}^K (w_{j,1} - w_{j,2}) X_j \otimes X_j \right\|_S^2 + \left\| \sum_{j=1}^K (v_{j,1} - v_{j,2}) Y_j \otimes Y_j \right\|_S^2 + \\ &+ 2 \sum_{j=1}^K \sum_{k=1}^K (w_{j,1} - w_{j,2})(v_{k,1} - v_{k,2}) \langle X_j, Y_k \rangle^2 \\ &= \sum_{j=1}^K \|X_j \otimes X_j\|_S^2 + 2 \sum_{j < k} (w_{j,1} - w_{j,2})(w_{k,1} - w_{k,2}) \langle X_j, X_k \rangle^2 + \\ &+ \sum_{j=1}^K \|Y_j \otimes Y_j\|_S^2 + 2 \sum_{j < k} (v_{j,1} - v_{j,2})(v_{k,1} - v_{k,2}) \langle Y_j, Y_k \rangle^2 + \\ &+ 2 \sum_{j=1}^K \sum_{k=1}^K (w_{j,1} - w_{j,2})(v_{k,1} - v_{k,2}) \langle X_j, Y_k \rangle^2. \end{aligned} \quad (8)$$

Let us now call $\delta_{j,k}^X = (w_{j,1} - w_{j,2})(w_{k,1} - w_{k,2})$, $\delta_{j,k}^Y = (v_{j,1} - v_{j,2})(v_{k,1} - v_{k,2})$ and $\delta_{j,k}^{XY} = (w_{j,1} - w_{j,2})(v_{k,1} - v_{k,2})$. Now, it is: $\delta_{j,k}^X = +1$ if observations X_j and X_k are assigned to the same group, while on the contrary it is $\delta_{j,k}^X = -1$. The same applies for $\delta_{j,k}^Y$ with Y_j and Y_k . Finally, $\delta_{j,k}^{XY} = +1$ if X_j and Y_k are assigned to different groups, and $\delta_{j,k}^{XY} = -1$ on the contrary. It is now clear that, the distance between covariance operators is increased when two observations of populations \mathcal{X} or \mathcal{Y} are both assigned to the same group, or when two observations of the opposite populations \mathcal{X} and \mathcal{Y} are assigned to different groups. These remarks suggest the idea that, under the previous assumption, by recovering the original labelling of the two populations \mathcal{X} and \mathcal{Y} the distance between the induced covariance operators is increased.

If we replace the estimators of covariance operators in (8) with their expected values,

$$\begin{aligned} \mathbb{E}[\widehat{\mathcal{C}}_1^{(g)}] &= \mathcal{C}_1^{(g)} = \frac{\sum_{j=1}^N w_{j,1}}{K} \mathcal{C}_1 + \frac{\sum_{j=1}^K v_{j,1}}{K} \mathcal{C}_2, \\ \mathbb{E}[\widehat{\mathcal{C}}_2^{(g)}] &= \mathcal{C}_2^{(g)} = \frac{\sum_{j=1}^N w_{j,2}}{K} \mathcal{C}_1 + \frac{\sum_{j=1}^K v_{j,2}}{K} \mathcal{C}_2, \end{aligned}$$

then, by denoting $N_{1,2} = \sum_{j=1}^K w_{j,2}$ and also considering the relations among the variables $w_{j,g}$ and $v_{j,g}$ we get:

$$d(\mathcal{C}_1^{(g)}, \mathcal{C}_2^{(g)}) = \left\| \mathbb{E}[\widehat{\mathcal{C}}_1^{(g)}] - \mathbb{E}[\widehat{\mathcal{C}}_2^{(g)}] \right\|_S^2 = \left(1 - 2 \frac{N_{1,2}}{K}\right)^2 \|\mathcal{C}_1 - \mathcal{C}_2\|_S^2, \quad (9)$$

specifying that the maximum distance between (exact) covariances is attained when the groupings coincide with the original but unknown indexing of the data set.

If assumption (1) is true and in view of (9), a natural way to recover the true indexing can be to find the recombination of data in two groups maximising the distance between the induced covariance operators, i.e., to solve the optimization problem:

$$[I_1^*; I_2^*] = \arg \max_{i \in R_C} \left\{ d \left(\mathcal{C}_1^{(i)}, \mathcal{C}_2^{(i)} \right) \right\}, \quad R_C = \{1, \dots, N_C\}. \quad (\mathbf{P})$$

Identity (9) ensures that either $I_1^* = I_1^{(0)}$ and $I_2^* = I_2^{(0)}$, or $I_1^* = I_2^{(0)}$ and $I_2^* = I_1^{(0)}$. The double solution is due to the symmetry of (9), yet the groups represent the same partition of data, for this reason in the following we will not distinguish between them.

Practically, only approximate estimates of $\mathcal{C}_1^{(i)}$ and $\mathcal{C}_2^{(i)}$ are available, thus we must recast problem (\mathbf{P}) into:

$$\left[\tilde{I}_1^*; \tilde{I}_2^* \right] = \arg \max_{i \in R_C} \left\{ d \left(\tilde{\mathcal{C}}_1^{(i)}, \tilde{\mathcal{C}}_2^{(i)} \right) \right\}, \quad R_C = \{1, \dots, N_C\}, \quad (\tilde{\mathbf{P}})$$

The method we propose coincides with finding a solution to problem $(\tilde{\mathbf{P}})$.

In general \tilde{I}_1^* and \tilde{I}_2^* may differ from $I_1^{(0)}$ and $I_2^{(0)}$, since they are determined based on estimates of covariance operators. Indeed, provided that the chosen distance is capable of emphasizing the actual differences between covariances of the two populations, results could be improved by enhancing the accuracy of estimators. In Subsection 3.2 we will address the former issue.

3.1 Greedy formulation

In order to solve problem $(\tilde{\mathbf{P}})$, it would be required to test each of the N_C recombinations of indexes in order to find the desired pair \tilde{I}_1^* and \tilde{I}_2^* . Of course, the number of tests to be performed, $N_C = \binom{N}{K}$, with $K = N/2$, undergoes a combinatorially-fast growth, as N increases. Thus, unless we have only a small number of observations in our data set, the naive approach of performing an exhaustive search in the set of recombinations is not feasible. This calls for a proper complexity-reduction strategy, aimed at restraining the complexity and enabling the application of our method also to data sets with a common size.

3.1.1 MAX-SWAP ALGORITHM

We propose to rephrase problem $(\tilde{\mathbf{P}})$ into a greedy algorithm, with a greatly reduced complexity: The driving idea is to interpret $d(\tilde{\mathcal{C}}_1^{(i)}, \tilde{\mathcal{C}}_2^{(i)})$ as an objective function of i , and, starting from an initial guess (I_1^0, I_2^0) , to iteratively increase it by allowing exchanges of units between the two groups. The exchange of data must preserve the total number of units inside each group, so each group discards and receives an equal number of units, say up to J per group.

We propose to choose the swapping units in such a way that the distance between the estimated covariance operators at the next step be strictly higher than the previous one and, heuristically, the highest possible. Convergence is reached when no further swap can increase that distance.

This strategy can be also motivated by (8), since it performs swaps between observations in order to increase $\|\tilde{\mathcal{C}}_1^{(i)} - \tilde{\mathcal{C}}_2^{(i)}\|$, thus trying to assign the correct values to $v_{j,g}$ and $v_{j,g}$.

Algorithm 1: Max-Swap algorithm

Input: Initial guess: (I_1^0, I_2^0)

Output: Estimated indexing $(\tilde{I}_1^*, \tilde{I}_2^*)$

Compute $(\tilde{\mathcal{C}}_1^0, \tilde{\mathcal{C}}_2^0)$ induced by (I_1^0, I_2^0) ;

$d^0 = d(\tilde{\mathcal{C}}_1^0, \tilde{\mathcal{C}}_2^0)$;

$k = 1$;

$(\Delta d)^k = 1$;

while $(\Delta d)^k > 0$ **do**

for $s \in 1, \dots, K$ **do**

for $t \in 1, \dots, K$ **do**

 Swap in first group: $\tilde{I}_1 = \bigcup_{p \neq s} I_1^{k-1}(p) \cup I_2^{k-1}(t)$;

 Swap in second group: $\tilde{I}_2 = \bigcup_{p \neq t} I_2^{k-1}(q) \cup I_1^{k-1}(s)$;

 Compute $(\tilde{\mathcal{C}}_1, \tilde{\mathcal{C}}_2)$ induced by $(\tilde{I}_1, \tilde{I}_2)$;

$D_{s,t} = d(\tilde{\mathcal{C}}_1, \tilde{\mathcal{C}}_2)$;

$(s^*, t^*) = \arg \max_{s,t} D_{s,t}$;

$d^k = D_{s^*, t^*}$;

$(\Delta d)^k = d^k - d^{k-1}$;

$I_1^k = \bigcup_{p \neq s^*} I_1^{k-1}(p) \cup I_2^{k-1}(t^*)$;

$I_2^k = \bigcup_{q \neq t^*} I_2^{k-1}(q) \cup I_1^{k-1}(s^*)$;

$k = k + 1$;

$d^{**} = d^{k-1}$;

$I_1^{**} = I_1^{k-1}$;

$I_2^{**} = I_2^{k-1}$;

When searching the best swap of size up to J , we must explore a number of combinations of the current groups equal to $\sum_{i=1}^J \binom{K}{i}^2$. Therefore it is evident that J affects both the computational effort and the robustness of our algorithm: the lower is J , the less permutations we have to search among to find the optimal swap; the greater is J , the more likely we are to detect and exchange at once a block of truly extraneous units. We point out that, for $J = K$ we recover the original complexity of solving problem $(\tilde{\mathbf{P}})$ in just one step, since it holds:

$$\binom{N}{K} = \sum_{i=0}^K \binom{K}{i}^2 = \sum_{i=1}^K \binom{K}{i}^2 + 1.$$

We propose to set $J = 1$, in order to save computations, and to choose the units to be exchanged by exploring the K^2 swaps of one unit from the first group with another unit of

the second group. Then we select the one yielding the maximum increment in the distance. The complete formulation of our Max-Swap algorithm is summarised in Algorithm 1, where we specify for the sake of clarity that the symbol $I_1^k(p)$, for instance, indicates the p -th element of the set of indexes I_1^k . In the following we will denote the estimated set of indexes at step k of algorithm with superscript k without brackets, (I_1^k, I_2^k) .

3.1.2 CONVERGENCE

We turn now to the study of the convergence of our proposed algorithm. With reference to the notation of Algorithm 1, it is easy to prove that

$$\textbf{Proposition 1} \quad \textit{The monotonicity constraint:} \quad (\Delta d)^k > 0 \quad \forall k \geq 1, \quad (10)$$

ensures that convergence always happens, at least to a local maximum of $d(\widehat{\mathcal{C}}_1^{(t)}, \widehat{\mathcal{C}}_2^{(t)})$.

Proof As a simple consequence of (10), the list of intermediate indexings:

$$(I_1^0, I_2^0), (I_1^1, I_2^1), \dots, (I_1^k, I_2^k), \dots$$

does not have cycles (a contiguous sub-sequence with equal extrema). In fact, let there be a cycle of minimal period L starting at iteration k_0 , then it should hold:

$$0 = d^{k_0+L} - d^{k_0} = \sum_{j=1}^L (\Delta d)^{j+k_0} > 0,$$

which is a contradiction. Thus each element in the list is unique and contained in the set of all the possible recombinations of data:

$$(I_1^{(0)}, I_2^{(0)}), (I_1^{(1)}, I_2^{(1)}), \dots, (I_1^{(N_C)}, I_2^{(N_C)}),$$

which has a finite number of elements. Therefore the algorithm, however initialised, converges. ■

Now that convergence has been established, we can formulate the following proposition:

Proposition 2 *When estimates of covariance operators, $\widehat{\mathcal{C}}_1$ and $\widehat{\mathcal{C}}_2$, are exact, i.e., when $\widehat{\mathcal{C}}_1 = \mathcal{C}_1$ and $\widehat{\mathcal{C}}_2 = \mathcal{C}_2$, the greedy algorithm converges to the exact solution (I_1^*, I_2^*) of problem (\mathbf{P}) in at most $K/2$ steps.*

Proof This is a consequence of Proposition 1 and the convexity of the objective function $d(\widehat{\mathcal{C}}_1^{(t)}, \widehat{\mathcal{C}}_2^{(t)})$ w.r.t $N_{1,2}$ showed in (9), making it impossible that local maxima exist. ■

A consequence of Proposition 1 is that in general $(I_1^{**}, I_2^{**}) \neq (\widehat{I}_1^*, \widehat{I}_2^*)$, since the algorithm may converge only to a local maximizer of the covariances' distance. This is a well-known drawback affecting greedy methods for optimization problems based on local-search patterns

and the development of possible remedies is a very active research field in algorithmics and optimization disciplines. A simple way to correct for the possibility to select only a local optimum is to implement a *restart*-like strategy, namely run multiple instances of the algorithm with different initialisations and to select the best results in terms of optimised objective function. This is a simple and classic solution to the drawbacks of general local-optimisation algorithms.

However, Proposition 2 assures that the algorithm converges to the exact solution, provided that we have a thorough knowledge of covariance operators. Therefore, the possible non-convexity of the objective function $d(\mathcal{C}_1^{(t)}, \mathcal{C}_2^{(t)})$, being the reason why local maxima exist, would be a direct consequence of the small precision in estimating covariances.

Under this light, we can rephrase the problem of enhancing our method from finding an algorithmics-like remedy to the aforementioned drawback, to the study of accuracy properties of covariance estimators. This will be the focus of Subsection 3.2. Another possibility to reduce the risk of selecting only local maximisers, which we don't investigate in the following, would be to assess the stability of the maximiser found when running a standard Max-Swap algorithm by running a general version of Max-Swap with $J > 1$.

3.1.3 RATE OF CONVERGENCE AND COMPLEXITY

Max-Swap algorithm increases the objective function starting from an initial guess, (I_1^0, I_2^0) . In general, we have no prior knowledge on the distribution of true groups among the data set, then we should choose the initial guess at random, and in particular by drawing from the set of indexes without replacement, assigning equal probability to each outcome. This strategy causes the quantity $N_{1,2}$ in (9) to follow a hypergeometric distribution:

$$N_{1,2} \sim \text{Hyper}(N, K, K),$$

i.e.,

$$\mathbb{P}(N_{1,2} = h) = \frac{\binom{K}{h} \binom{K-h}{K-h}}{\binom{N}{K}}, \quad \mathbb{E}[N_{1,2}] = \frac{K}{2},$$

with typically large values of N and $K = N/2$. Owing to the fast decay of hypergeometric mass function away from its mean, we can presume that the random initial draw will cause $N_{1,2}$ to be most likely in some neighbourhood of $K/2$ (we imagine K to be even).

Let us assume that hypotheses of Proposition 2 are true, then if we consider a value for $N_{1,2}$ resulting from the initial guess, the number of iterations to convergence is $N_{\text{it}} = \min(N_{1,2}, K - N_{1,2})$, corresponding to N_{it} consecutive and correct swaps. Then, by initialising at random the algorithm:

$$N_{\text{it}} \leq \frac{K}{2}.$$

If we summarise the complexity of solving problem (\mathbf{P}) with the number of combinations N_{comb} processed to recover the estimated groups (I_1^*, I_2^*) , and we compare our proposed method (MS) with the naive, exhaustive strategy (N), we have:

$$N_{\text{comb}}^{MS} = K^2 \quad N_{\text{it}} \leq \frac{K^3}{2}, \quad N_{\text{comb}}^N = \binom{N}{K},$$

where the multiplicative factor K^2 in $N^{M/S}_{\text{comb}}$ accounts for the number of combinations searched for to find the best swap at each step of Max-Swap algorithm. This shows how Max-Swap algorithm entails a far lower complexity than the brute force approach.

3.1.4 CLASSIFICATION OF NEW OBSERVATIONS

Once the algorithm has been run and the two groups constituting the mixed data set have been found, a simple rule can be applied to use the clusters in order to classify new observations. We suggest to consider each new unit separately, and compute the covariances obtained by including the unit either in the first or second group. For the sake of simplicity, we denote them, respectively, by \hat{C}_1 and \hat{C}_2 . Then we compute the distances $d_1 = d(\hat{C}_1, \hat{C})$ and $\tilde{d}_2 = d(\hat{C}_2, \hat{C}_2)$, where here we indicate by \hat{C}_1 and \hat{C}_2 the covariances of the two groups determined at the end of the clustering stage. Then we attribute the new unit to the group for which the distance d is minimum.

3.2 Shrinkage estimation of covariance

In this subsection we consider the problem of improving the estimation of covariance operators, so that clustering is more accurate. In particular, we will describe an alternative estimator of data covariance than the sample covariance, which is better conditioned and in some circumstances achieves lower MSE. We will make use of it in our clustering algorithm as an alternative to sample covariance.

Let us consider a generic family of functional data $\mathcal{X} \sim P_{\mathcal{X}}$, such that $\mathbb{E}[\mathcal{X}] = 0$, $\mathbb{E}\|\mathcal{X}\|^2 < \infty$. We denote its covariance with \mathcal{C} and we imagine to estimate it with $\hat{\mathcal{C}}$. Our purpose is to find its best possible approximation, or saying it otherwise, being:

$$\text{MSE}_{\mathcal{S}}(\hat{\mathcal{C}}) := \mathbb{E}\|\hat{\mathcal{C}} - \mathcal{C}\|_{\mathcal{S}}^2,$$

our measure of the estimation error of $\hat{\mathcal{C}}$, to solve the following estimation problem **(E)**:

$$\hat{\mathcal{C}}^* = \arg \min_{\hat{\mathcal{C}}} \text{MSE}_{\mathcal{S}}(\hat{\mathcal{C}}) = \arg \min_{\hat{\mathcal{C}}} \mathbb{E}\|\hat{\mathcal{C}} - \mathcal{C}\|_{\mathcal{S}}^2, \quad (\text{E})$$

where the minimum is sought among all possible estimators $\hat{\mathcal{C}}$ of \mathcal{C} . We point out that in MSE $_{\mathcal{S}}$ we use our selected distance to measure the discrepancy of estimation.

Of course, from a practical viewpoint, only a finite-dimensional estimation of \mathcal{C} can be attained, given data. In addition, functional data are often available from sources as discrete measurements of a signal over some one dimensional grid. Let us indicate by X_i the i -th (out of N) sample realisation of process \mathcal{X} , i.e.:

$$X_i = (X_i(t_j))_{j=1}^P, \quad I^h = [t_1, \dots, t_P], \quad (11)$$

where, for the sake of simplicity, we have imagined the grid I^h to be regularly spaced (although this is not mandatory), i.e., $t_{j+1} - t_j = h > 0$ for $j = 1, \dots, P - 1$. A crucial point when analysing functional data is to reconstruct functions from scattered measurements

X_i , which requires the use of some proper smoothing technique. Furthermore, the so called *phase variability* of reconstructed signals, involving the dispersion of features along the grid axis, can be separated from *amplitude variability*, appearing as the dispersion of magnitudes of values of X_i . This process is known as registration (see, for instance, Ramsay and Silverman, 2005). Once data have been smoothed and registered, they can be re-evaluated onto another one dimensional grid. To save notation we will assume that discrete representations in (11) have already been preprocessed.

It is clear that, within this habit, covariance estimators of \mathcal{C} are discrete, matrix-type approximations obtained starting from pointwise observations X_i . For instance, standard sample covariance estimator for zero-mean data is:

$$\mathbf{S} = \frac{1}{N} \sum_{i=1}^N X_i X_i^T. \quad (12)$$

If we denote the true, discrete covariance structure related to each X_i by \mathbf{C} the discrete version of problem **(E)** is:

$$\mathbf{C}^* = \arg \min_{\hat{\mathbf{C}}} \mathbb{E}\|\hat{\mathbf{C}} - \mathbf{C}\|_F^2, \quad (\text{E})$$

where the minimum is sought inside the set of symmetric and positively defined matrix-type estimators of dimension P . We point out that the subscript F in **(E)** indicates the Frobenius norm, that is the finite-dimensional counterpart of the Hilbert-Schmidt norm for operators.

When the sample size N is low compared to the number of features P , sample covariance may loose in accuracy, meaning that the actual estimate might be quite distant from the true covariance \mathbf{C} (this can be seen as a consequence of the so-called *Stein's phenomenon*, Stein, 1956).

A typical remedy to the poor performances of sample covariance, often used in the setting of *Large P - Small N* problems, is to replace it with a biased, shrinkage estimator. Other solutions might be jackknife or bootstrap, but their computational cost renders them practically useless in our clustering algorithm, which requires to repeatedly estimate covariance matrices. Shrinkage estimation has been explicitly applied to the context of large covariance matrices in (Ledoit and Wolf, 2003, 2004) and (Schäfer and Strimmer, 2005), turning out in a sufficiently lightweight procedure. In those works, authors start from problem **(E)** and build an estimator that is asymptotically more accurate and better conditioned than sample covariance. We follow the approach described in (Ledoit and Wolf, 2004) and consider the class of linear shrinkage estimators of the form:

$$\hat{\mathbf{C}} = \mu\gamma\mathbf{I} + (1 - \gamma)\mathbf{S}, \quad (13)$$

where \mathbf{I} is the $P \times P$ identity matrix and $\gamma \in [0, 1]$, $\mu \in \mathbb{R}^+$ and \mathbf{S} is the sample covariance estimator. Obviously, the class contains the sample covariance estimator itself. Then **(E)** is solved with respect to the optimal values of μ and γ :

$$(\mu^*, \gamma^*) = \arg \min_{\mu, \gamma} \frac{\mathbb{E}\|\mathbf{C} - \mu\gamma\mathbf{I} - (1 - \gamma)\mathbf{S}\|_F^2}{P}. \quad (14)$$

If we introduce the quantities:

$$\alpha^2 = \frac{\|\mathbf{C} - \mu\mathbf{I}\|_F^2}{P}, \quad \beta^2 = \frac{\mathbb{E}\|\mathbf{S} - \mathbf{C}\|_F^2}{P}, \quad \delta^2 = \frac{\mathbb{E}\|\mathbf{S} - \mu\mathbf{I}\|_F^2}{P}, \quad (15)$$

and note that these are subjected to $\alpha^2 + \beta^2 = \delta^2$, we can perform the explicit minimization in equation (14). The expressions of μ^* and γ^* are:

$$\mu^* = \frac{\langle \mathbf{C}; \mathbf{I} \rangle_F}{P} = \frac{\text{Tr}(\mathbf{C})}{P}, \quad \gamma^* = \frac{\beta^2}{\delta^2}, \quad (16)$$

where we have used $\mu = \mu^*$ in the computation of δ . The desired shrinkage estimator becomes:

$$\mathbf{S}^* = \mu^* \frac{\beta^2}{\delta^2} \mathbf{I} + \frac{\alpha^2}{\delta^2} \mathbf{S}. \quad (17)$$

Of course, estimator (17) depends on the unknown exact covariance matrix \mathbf{C} , even though only through four scalar functions. In (Ledoit and Wolf, 2004) authors solve this problem by proposing the following estimators for α , β , δ and μ^* :

$$\hat{\mu}^* = \frac{\text{Tr}(\mathbf{S})}{P}, \quad \hat{\delta}^2 = \frac{\|\mathbf{S} - \hat{\mu}^* \mathbf{I}\|_F^2}{P}, \quad (18)$$

$$\hat{\beta}^2 = \min \left(\hat{\delta}^2; \frac{1}{N^2} \sum_{k=1}^N \frac{\|X_k X_k^T - \mathbf{S}\|_F^2}{P} \right), \quad (19)$$

and $\hat{\alpha}^2 = \hat{\delta}^2 - \hat{\beta}^2$.

Then, the actual shrinkage estimator is:

$$\hat{\mathbf{S}}^* = \widehat{\mu}^* \frac{\hat{\beta}^2}{\hat{\delta}^2} \mathbf{I} + \frac{\hat{\alpha}^2}{\hat{\delta}^2} \mathbf{S}. \quad (20)$$

In (Ledoit and Wolf, 2004) authors show how estimates (18) are consistent, in the sense that they converge to the exact values in quadratic mean, under the general asymptotic limits of P and N , i.e., when both P and N are allowed to go to infinity but there exists a $c \in \mathbb{R}$ independent on N such that $P/N < c$ (see Ledoit and Wolf, 2004 and references therein for theoretical details on general asymptotics). Moreover, estimator $\hat{\mathbf{S}}^*$ is an asymptotically optimal linear shrinkage estimator for covariance matrix \mathbf{C} with respect to quadratic loss. Besides its asymptotic properties, extensive use in applications shows that the accuracy gain resulting from $\hat{\mathbf{S}}^*$ in terms of decrease in MSE is substantial also in many finite sample cases, and that standard covariance is almost always matched and often outperformed by $\hat{\mathbf{S}}^*$. For a detailed description of how and when shrinkage estimation of covariance is recommended over the sample estimation, see for instance (Ledoit and Wolf, 2004).

4. Case Studies

In this section we provide three simulations involving our proposed clustering method. In Subsection 4.1 we show a first example, regarding standard bivariate data, in order to

give a clear geometric idea of clustering based on covariance structures. In Subsection 4.2 we show an application to synthetic functional data. In these former two examples the true subdivision of samples is known, so the goodness of the clustering arising from Max-Swap algorithm is assessed against the true identities of data. In Subsection 4.3, instead, we apply the clustering algorithm on real functional data expressing the concentration of deoxygenated hemoglobin measured in human subjects' brains.

4.1 Multivariate data

This test is meant to provide a first, visual example of the features of the clustering arising from using Max-Swap algorithm. In order to ease the geometrical interpretation, we chose to focus on two bivariate data sets, composed of simulated data with a-priori designed covariances. Indeed, by representing bi-dimensional data we are able to support our considerations with a clear graphical counterpart.

We exploit two reference data sets having the same means but different variance-covariance structures. In particular, a generic clustering based on locations run on these data is meant to fail. The first set of data, hereafter *hourglass* data, has covariances whose difference lies in the directions along which variability expresses. We generated it according to the following laws:

$$X = \rho_x (\cos \theta_x, \sin \theta_x), \quad \rho_x \sim \mathcal{U}[-1, 1], \quad \theta_x \sim \mathcal{U} \left[\frac{\pi}{12}, \frac{5\pi}{12} \right], \\ Y = \rho_y (\cos \theta_y, \sin \theta_y), \quad \rho_y \sim \mathcal{U}[-1, 1], \quad \theta_y \sim \mathcal{U} \left[\frac{7\pi}{12}, \frac{11\pi}{12} \right],$$

where the four random variables, $\rho_x, \rho_y, \theta_x, \theta_y$ are independent. Simple calculations reveal that $\mathbb{E}[X] = 0$ and $\mathbb{E}[Y] = 0$, while covariances are:

$$\mathbf{C}_x = \begin{pmatrix} 1/6 & \sqrt{3}/4\pi \\ \sqrt{3}/4\pi & 1/6 \end{pmatrix}, \quad \mathbf{C}_y = \begin{pmatrix} 1/6 & -\sqrt{3}/4\pi \\ -\sqrt{3}/4\pi & 1/6 \end{pmatrix}.$$

Note that X and Y differ only in their covariances. Moreover, since only off-diagonal terms of \mathbf{C}_x and \mathbf{C}_y are different (and indeed opposed), the two families have the same kind of variability, only expressed along orthogonal directions in the plane. We generate a data set D of $N = 400$ data, according to the previous laws, made up of $K = 200$ samples from X and $K = 200$ samples from Y , which are displayed in Figure 1.

We considered also another data set, referred to as *bull's eye*, whose features are somehow complementary to the ones of *hourglass*, since variabilities of *bull's eye* sub-populations express along the same directions, though with different magnitudes. In particular, we considered the following laws:

$$X = \rho_x (\cos \theta_x, \sin \theta_x), \quad \rho_x \sim \mathcal{U} \left[0, \frac{1}{2} \right], \quad \theta_x \sim \mathcal{U} [0, 2\pi], \\ Y = \rho_y (\cos \theta_y, \sin \theta_y), \quad \rho_y \sim \mathcal{U} \left[2, \frac{5}{2} \right], \quad \theta_y \sim \mathcal{U} [0, 2\pi],$$

where, still, the four random variables $\rho_x, \rho_y, \theta_x, \theta_y$ are independent. This leads to covariances:

$$\mathbf{C}_x = \frac{1}{24} \mathbf{I}, \quad \mathbf{C}_y = \frac{61}{24} \mathbf{I}$$

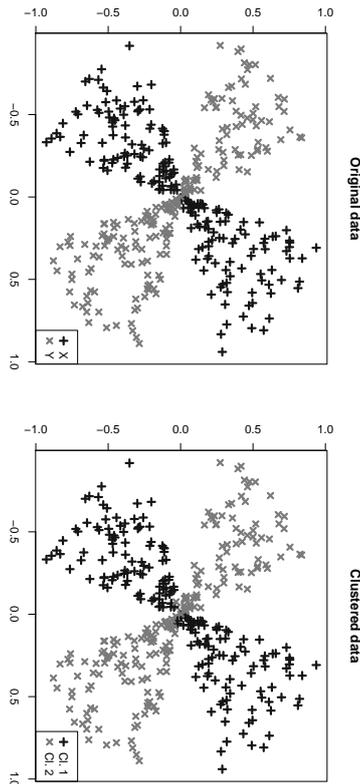


Figure 1: *Left*: Hourglass data set used in the first multivariate experiment, collecting $N = 400$ points subdivided into family X ($K = 200$ points, marked by +) and family Y ($K = 200$ points, marked by x). *Right*: Outcome of clustering via Max-Swap algorithm.

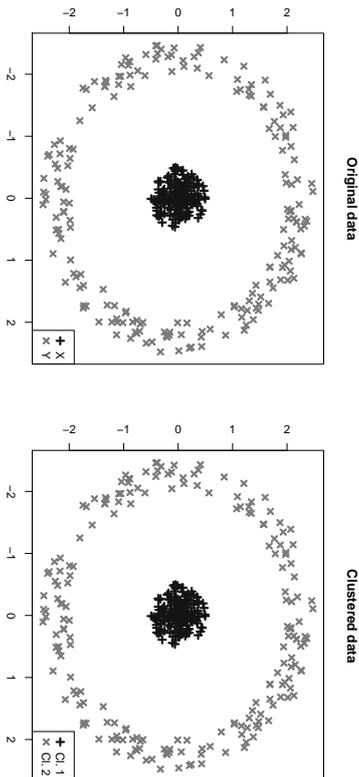


Figure 2: Bull's eye data set used in the second multivariate experiment, collecting $N = 400$ points subdivided into family X ($K = 200$ points, marked by +) and family Y ($K = 200$ points, marked by x). *Right*: Outcome of clustering via Max-Swap algorithm.

that clearly differ only in terms of their variability's magnitude. *Bull's eye* data set is generated according to these laws for an overall cardinality of $N = 400$ data subdivided in two groups of size $K = 200$ each, and it is shown in Figure 2.

We point out that, in order to improve the robustness of results with respect to the chance of selecting only a local maximiser of the distance between variance-covariance structures, Max-Swap algorithm was run for 10 times, keeping the result for which the objective function was highest. Since the number of data in each sub-population, K , is high with respect to their dimensionality, $P = 2$, we used Max-Swap algorithm in combination with the standard sample estimator of covariance, \mathbf{S} . The results of the clustering procedure are also shown in Figure 1 and Figure 2 (right panels), where it is clear how the clustering method is able to detect the observations belonging to the two populations, whose difference is its only in their covariances. Since in this simulated scenario we know the law generating observations, and therefore the labels of the generated data, we are able to assess the performances of the clustering procedure by comparing the two identified groups of data with the original labels. In particular, in the case of hourglass data, only 3 observations out of 200 in each cluster belong to the other population, and are all located very close to the data centre. In the bull's eye example, instead, the two clusters are composed of elements coming from only one population.

The outcome of standard clustering algorithms, like K-means or hierarchical clustering, show the complete inefficacy of location-based clustering for such data sets, where data are mostly different in terms of their variability. In particular, 20 K-means ($K = 2$) runs on Bull's eye and Hourglass data sets yield, if compared with the true labelling of observations, a mis-classification rate of 0.27 ± 0.01 and 0.47 ± 0.01 , respectively, while 20 runs of a hierarchical clustering with euclidean distance and Ward linkage give a mis-classification rate of 0.29 ± 0.05 and 0.46 ± 0.04 , respectively.

4.2 Synthetic functional data

In this subsection we apply our clustering algorithm to functional data. We use a data set composed of two populations of functions, \mathcal{X} and \mathcal{Y} , with null means and covariance operators:

$$C_x = \sum_{i=1}^L \sigma_i \varepsilon_i \otimes \varepsilon_i, \quad C_y = \sum_{i=1}^L \eta_i \varepsilon_i \otimes \varepsilon_i, \quad (21)$$

where $\{\varepsilon_i\}_{i=1}^L$ are the first L elements of the orthonormal Fourier basis on the interval $I = [0, 1]$, save for the constant, i.e.,:

$$e_{2k-1} = \sqrt{2} \sin(2k\pi x), \quad e_{2k} = \sqrt{2} \cos(2k\pi x), \quad x \in I,$$

for $k = 1, \dots, L/2$, and the eigenvalues are chosen as:

$$\sigma_i = 1, \quad \eta_i = \frac{\sigma_i}{\sqrt{5}}, \quad \forall i = 1, \dots, L. \quad (22)$$

In what follows we considered $L = 30$. A visual representation of the related covariance functions is in Figure 3. It is clear from (21) and from Figure 3 that covariances C_x and

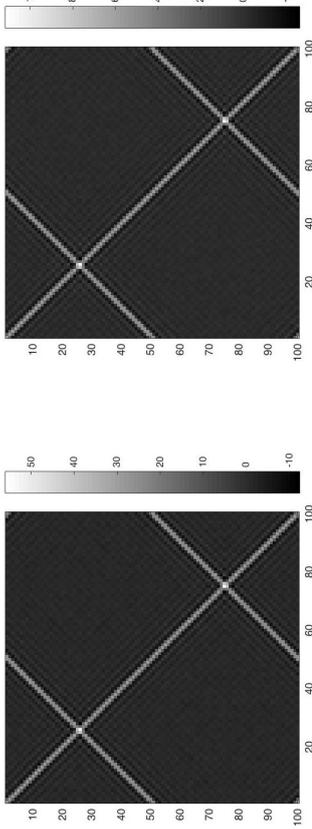


Figure 3: Contour plot of covariances \mathcal{C}_x and \mathcal{C}_y for the experiment with synthetic functional data. The different scales of contour plots show that the difference between the variance-covariance structures is only in magnitude.

\mathcal{C}_y are different only in terms of variability's magnitudes, while their eigenfunctions are the same.

We generated several sets of the following synthetic families of Gaussian functional data having covariances like in (21):

$$X_i = \sum_{j=1}^L \xi_{ij} \sqrt{\sigma_j} \epsilon_j, \quad Y_i = \sum_{j=1}^L \zeta_{ij} \sqrt{\tau_j} \epsilon_j, \quad (23)$$

for $i = 1, \dots, K$, where $\xi_{ij} \sim \mathcal{N}(0, 1)$, and are independent from $\zeta_{ij} \sim \mathcal{N}(0, 1)$. Each synthetic functional unit has been evaluated on a grid of $P = 100$ points, evenly spaced on I . The different sets have been generated choosing $K \in \{20, 25, 30, 35, 40, 45, 50\}$, corresponding to total cardinalities of $N \in \{40, 50, 60, 70, 80, 90, 100\}$.

We applied our clustering algorithm to each synthetic data set. The different values of K (i.e., of N) allow to study the performances of clustering as the sample size increases. This is of interest since our method relies on the estimation of covariance matrices, thus we expect that when the number of data increases the performances tends to improve. We used Max-Swap algorithm both with the standard sample covariance estimator, \mathbf{S} , and with the shrinkage covariance estimator $\hat{\mathbf{S}}^*$.

Like in the case of multivariate data, we know the laws generating observations, hence their original labels, therefore we can analyse the identified clusters in terms of the composition of units from X and Y . This allows us to understand whether the algorithm is able to detect groups of observations that we know *a priori* are different only in their variability. We report the results of the clustering procedure in Tab. 1. Similarly to the case of multivariate synthetic data of Subsection 4.1, each of them is related to the one trial in a set of 10 for which the distance between covariances was highest. This was done in order to take

N		K		Sample covariance (S)		Shrinkage estimator ($\hat{\mathbf{S}}^*$)	
		Misc.	(1, 2)	Err.	Misc.	(1, 2)	Err.
40	20	2	(1, 1)	5%	0	(0, 0)	0%
50	25	2	(1, 1)	4%	0	(0, 0)	0%
60	30	0	(0, 0)	0%	0	(0, 0)	0%
70	35	6	(3, 3)	8.5%	2	(1, 1)	3%
80	40	2	(1, 1)	2.5%	2	(1, 1)	2.5%
90	45	0	(0, 0)	0%	0	(0, 0)	0%
100	50	0	(0, 0)	0%	0	(0, 0)	0%

Table 1: Clustering performances results for the application with synthetic functional data.

account of the heuristic nature of the algorithm.

Results undoubtedly highlight that covariance-based clustering is effective, yielding groups which can easily be interpreted as the original ones up to an error always lower than 10%, also for challenging cases of scarce data. In addition, for reasonable sample sizes, the error tends to get lower. The results are even more satisfactory if related to the dimension of the covariance matrices of these data, i.e. $P = 100$, since a successful clustering may be carried out with only 25-30 units per family.

If we compare the performances gained when using \mathbf{S} with those attained by using $\hat{\mathbf{S}}^*$, we see that a substantial improvement in accuracy has been achieved. Moreover, in this experiment the performances of Max-Swap combined with \mathbf{S}^* were more stable across the trials, and almost always close to the best one for all trials. This may be an advantage with respect to \mathbf{S} , which in turn gave results more variable from trial to trial. On the contrary, from a computational point of view, resorting to \mathbf{S} leads to faster simulations, while using $\hat{\mathbf{S}}^*$ requires higher effort, especially when K is large.

4.3 Deoxygenated hemoglobin data

In this subsection we apply our clustering method to a real data set belonging to a biomedical context. In particular, we deal with data produced by functional near-infrared spectroscopy (fNIRS), an optical technique able to noninvasively monitor the cerebral hemodynamic at cortical level. Exploiting the relatively low absorption of biological tissues, light in the red and near-infrared wavelength range can penetrate the human head down to some centimeters and reach the cerebral cortex. Therefore, fNIRS can provide a measure of oxy- and deoxy-hemoglobin, the main chromophores contributing to light absorption at this wavelength range. In particular we study the measurements along time of the concentration of deoxygenated hemoglobin in the brain of a group of six right-handed healthy subjects (male, 44 years old) while they are carrying out a motor task (i.e., squeezing a soft ball in the right hand) at a rate of 2 Hz guided by a metronome. The measures of each subject were made on eight different points of the brain, four located in the central part of the left hemisphere and another four located in the central part of the right hemisphere. The

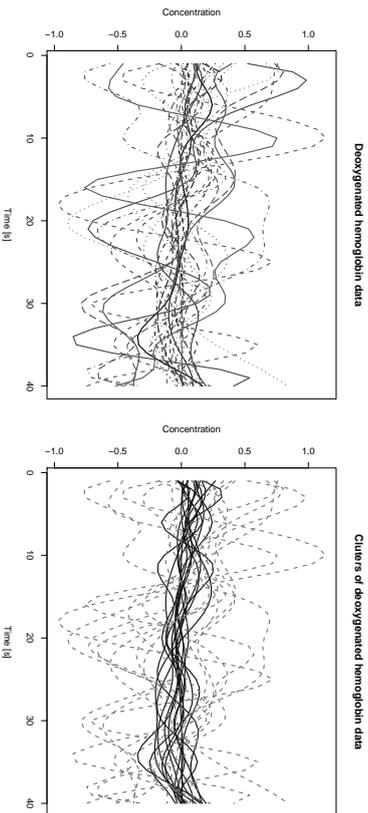


Figure 4: Deoxygenated hemoglobin's concentration data. In the left panel are represented the preprocessed data on which the clustering is carried out. In the right panel is shown the output.

measurement and preprocessing techniques as well as the experimental instruments used to collect data are described in (Torricelli et al., 2014; Zucchelli et al., 2013; Re et al., 2013).

Each statistical unit of the data set consists of a sampling along 40 seconds of deoxygenated hemoglobin's concentration at the related location on the brain. Our clustering purpose is to recognize the signals of patients whose trends in hemoglobin's concentration show wide fluctuations across their mean profiles from signals where the concentration varies little. The aim of the study was in fact to detect different behaviours corresponding to activated vs. non activated cerebral areas. This activation is reflected in difference in covariance operators more than in difference in the mean level of deoxygenated hemoglobin concentration, thus we wish to apply our clustering algorithm in order to detect the two clusters. In a pre-processing stage, signals affected by artifacts due to the measurement procedure were removed, while the others were de-trended and smoothed thanks to a B-Spline smoothing basis.

At the end of the pre-processing stage, a set of $N = 30$ signals, subdivided into two groups of $K = 15$ are available with a sampling rate of 1s, so that $P = 40$. These data are depicted in Figure 4.

We run the Max-Swap clustering algorithm on these data to perform clustering, both using \mathbf{S} and \mathbf{S}^* estimators, finding equal partitions of initial data. The results are shown in Figure 4, and highlight how the algorithm is able to answer to our request, i.e., to detect two clusters of functions that are well distinguishable in terms of their different variability. We interpret these as activated versus non activated brain regions and our results are in agreement with those obtained in (Bonomini et al., 2015).

5. Conclusions

In this paper we have studied the problem of performing clustering on two groups of data whose difference lies in their variance-covariance structures rather than in their means. We have formulated it according to the general statistical framework of functional data, yet it can be of interest also in other contexts, such as for multivariate data. We have shown how the naive clustering strategy is computationally intractable and we have proposed a new heuristic algorithm to override such issue. The algorithm is based on a proper quantification of the distance between estimates of covariance operators, which we assumed to be the natural Hilbert-Schmidt norm, and seeks for the partition of data producing the highest possible distance among estimated covariances. The partition is sought by modifying two initial guesses of the true groups with subsequent exchanges of units, in order to maximise the distance between estimated covariances. We have given its pseudo-code formulation and studied its convergence properties and complexity. A crucial point of the algorithm is the estimation of covariance operators, which can be done by standard sample covariance, but we have proposed a variant involving a linear shrinkage estimator, which promises to be at least as accurate as sample covariance, and often better in terms of mean square error. By means of some examples we have collected empirical evidence to prove that the algorithm is able to solve suitably the clustering problem, both when the variabilities are different in their magnitudes or in their directions. We compared the performances gained on functional data under the use of the sample estimator and of the linear shrinkage one, and found that both of them give definitely satisfactory results and that the use of linear shrinkage may provide a substantial improvement in terms of clustering performances.

References

- V. Bonomini, R. Re, L. Zucchelli, F. Leva, L. Spinelli, D. Contini, A. M. Paganoni, and Torricelli A. A new linear regression method for statistical analysis of firs data. *Biomedical optics express*, 2(6):615–630, 2015.
- D. Bosq. *Linear Processes in Function Spaces: Theory and Applications*, volume 149 of *Lecture Notes in Statistics*. Springer, 2000.
- T. T. Cai and A. Zhang. Inference for high-dimensional differential correlation matrices. *Journal of Multivariate Analysis*, 143:107–126, 2016.
- J. A. Hartigan. *Clustering Algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1975.
- L. Horváth and P. Kokoszka. *Inference for Functional Data with Applications*, volume 200 of *Springer Series in Statistics*. Springer, 2012.
- O. Ledoit and M. Wolf. Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *Journal of Empirical Finance*, 10(5):603–621, 2003.
- O. Ledoit and M. Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411, 2004.

- R. Mitra, P. Müller, and Y. Ji. Bayesian graphical models for differential pathways. *Bayesian Analysis*, 11(1):99–124, 2016.
- J. O. Ramsay and B.W. Silverman. *Functional Data Analysis*. Springer, New York, 2005.
- R. Re, D. Contini, M. Turolo, L. Spinelli, L. Zucchelli, M. Caffini, R. Cubeddu, and A. Torricelli. Multi-channel medical device for time domain functional near infrared spectroscopy based on wavelength space multiplexing. *Biomedical optics express*, 4(10):2231–2246, 2013.
- J. Schafer and K. Strimmer. A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, 4(1):1175–1189, 2005.
- C. Stein. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. In J. Neyman, editor, *Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 197–206, 1956.
- A. Torricelli, D. Contini, A. Pifferi, M. Caffini, R. Re, L. Zucchelli, and L. Spinelli. Time domain functional nirs imaging for human brain mapping. *Neuroimage*, 85:28–50, 2014.
- M. Watson. Coxpress: differential co-expression in gene expression data. *BMC bioinformatics*, 7(1):1, 2006.
- L. Zucchelli, D. Contini, R. Re, A. Torricelli, and L. Spinelli. Method for the discrimination of superficial and deep absorption variations by time domain fnirs. *Biomedical optics express*, 4(12):2893–2910, 2013.

MOCCA: Mirrored Convex/Concave Optimization for Nonconvex Composite Functions

Rina Foygel Barber

*Department of Statistics
University of Chicago
5747 South Ellis Avenue
Chicago, IL 60637, USA*

RINA@UCHICAGO.EDU

Emil Y. Sidky

*Department of Radiology
University of Chicago
5841 South Maryland Avenue
Chicago, IL 60637, USA*

SIDKY@UCHICAGO.EDU

Editor: Tong Zhang

Abstract

Many optimization problems arising in high-dimensional statistics decompose naturally into a sum of several terms, where the individual terms are relatively simple but the composite objective function can only be optimized with iterative algorithms. In this paper, we are interested in optimization problems of the form $F(Kx) + G(x)$, where K is a fixed linear transformation, while F and G are functions that may be nonconvex and/or nondifferentiable. In particular, if either of the terms are nonconvex, existing alternating minimization techniques may fail to converge; other types of existing approaches may instead be unable to handle nondifferentiability. We propose the MOCCA (mirrored convex/concave) algorithm, a primal/dual optimization approach that takes a local convex approximation to each term at every iteration. Inspired by optimization problems arising in computed tomography (CT) imaging, this algorithm can handle a range of nonconvex composite optimization problems, and offers theoretical guarantees for convergence when the overall problem is approximately convex (that is, any concavity in one term is balanced out by convexity in the other term). Empirical results show fast convergence for several structured signal recovery problems.

Keywords: MOCCA, ADMM, nonconvex, penalized likelihood, total variation, computed tomography

1. Introduction

We consider the problem of minimizing a composite objective function of the form

$$F(Kx) + G(x) \tag{1}$$

over $x \in \mathbb{R}^d$, where $K \in \mathbb{R}^{m \times d}$ is a fixed linear operator, and F and G are functions which are potentially nonconvex and/or nondifferentiable. Optimization problems of this form arise in many applications, and in particular, the algorithm developed here was motivated by an

image reconstruction problem for computed tomography (CT), an imaging technology used often in medicine and in other domains.

When F and G are both convex, many existing methods are well-equipped to handle this optimization problem, even in high dimensions. For example, the Alternating Direction Method of Multipliers (ADMM) (Boyd et al., 2011) and related primal/dual methods yield effective algorithms when the functions F and G both have inexpensive proximal maps, defined as

$$\text{Prox}_F(u) = \arg \min_w \left\{ \frac{1}{2} \|w - u\|^2 + F(w) \right\}$$

for any $u \in \mathbb{R}^m$, and same for Prox_G defined on \mathbb{R}^d . Methods such as ADMM are especially effective if F has an inexpensive proximal map but the linear transformation of the same function, i.e. the function $x \mapsto F(Kx)$, does not. If F does not offer an inexpensive proximal map but is instead smoothly differentiable (even if nonconvex), while G does have a fast proximal map, then methods such as proximal gradient descent (see e.g. Nesterov, 2013; Beck and Teboulle, 2009) can be applied instead of a primal/dual method.

In this paper, we consider a more challenging setting where F and G may both be nonconvex and nondifferentiable. For instance, we can consider working with functions that can be decomposed as $F = F_{\text{cox}} + F_{\text{diff}}$ and $G = G_{\text{cox}} + G_{\text{diff}}$, where we assume that $F_{\text{cox}}, G_{\text{cox}}$ are convex but do not need to be differentiable, while $F_{\text{diff}}, G_{\text{diff}}$ are potentially nonconvex but are differentiable. (If F_{diff} and G_{diff} are concave, then this type of optimization problem is often referred to as ‘‘convex/concave’’.) As we will see, this formulation arises naturally in a range of applications, but in general, cannot be handled by existing methods that are designed with convexity in mind. In this work, we generalize to a more flexible framework where F and G can each be locally approximated at any point with a convex function.

A special case is the setting where G is convex while F is nonconvex, but $x \mapsto F(Kx)$ is convex (i.e. if F is twice differentiable, then this is equivalent to assuming that $K^\top (\nabla^2 F) K \succeq 0$ but $\nabla^2 F \not\succeq 0$). In this case, the overall optimization problem, i.e. minimizing $F(Kx) + G(x)$, is a convex problem, that is, any local minimum is guaranteed to be a global minimum, and thus we might expect that this problem would be simple to optimize. Surprisingly, this may not be the case—if F is nondifferentiable, then we cannot use gradient descent on F , while the nonconvexity of F means that existing primal/dual optimization techniques might not be applicable.

In the nonconvex, or more specifically, convex/concave setting, one of the most common techniques used in place of convex methods is the majorization/minimization approach (Ortega and Rheinboldt, 1970; Hunter and Lange, 2000), where at each iteration, we work with a convex upper bound on the nonconvex objective function. Specifically, for the setting considered here, at iteration t we would choose some convex functions $F^{(t)}, G^{(t)}$ satisfying $F^{(t)} \geq F$ and $G^{(t)} \geq G$, then solve the modified optimization problem $\min_x \{F^{(t)}(Kx) + G^{(t)}(x)\}$. However, the modified optimization problem may itself be very challenging to solve in this setting, so we often cannot apply the majorization/minimization technique in a straightforward way. Our work combines the ideas of majorization/minimization with primal/dual techniques to handle this composite optimization problem.

1.1 The MOCCA Algorithm

In this work, we propose the mirrored convex/concave (MOCCA) algorithm, which offers an approach that combines some of the techniques described above. We work with a primal/dual formulation of the problem, and incorporate a majorization/minimization type step at each iteration. To motivate our method, we first present an existing method for the case where F and G are both convex, the Chambolle-Pock (CP) algorithm (Chambolle and Pock, 2011) (this method is closely related to other existing algorithms, which we discuss later on).

The CP algorithm is derived by considering the problem in a different form. From this point on, for clarity, we will use variables x, y, z to denote points in \mathbb{R}^d and u, v, w to denote points in \mathbb{R}^m . Since we are considering the setting where F is convex, by duality we can write

$$\min_x \{F(Kx) + G(x)\} = \min_w \max_x \{Kx, w\} - F^*(w) + G(x).$$

where F^* is the conjugate to F (Rockafellar, 1997), also known as the Legendre-Fenchel transform of F , and is defined as

$$F^*(w) = \max_v \{ \langle w, v \rangle - F(v) \}. \quad (2)$$

The primal variable x and dual variable w define a saddle-point problem. Given step sizes Σ, T which are positive diagonal matrices, the (preconditioned) CP algorithm (Chambolle and Pock, 2011; Pock and Chambolle, 2011) iterates the following steps:

$$\begin{cases} x_{t+1} = \arg \min_x \{ \langle Kx, w_t \rangle + G(x) + \frac{1}{2} \|x - x_t\|_{\Sigma^{-1}}^2 \}, \\ w_{t+1} = \arg \min_w \{ -\langle K\bar{x}_{t+1}, w \rangle + F^*(w) + \frac{1}{2} \|w - w_t\|_{\Sigma^{-1}}^2 \}, \end{cases} \quad (3)$$

where $\bar{x}_{t+1} = x_{t+1} + \theta(x_{t+1} - x_t)$ is an extrapolation term for some parameter $\theta \in [0, 1]$ (generally $\theta = 1$). Here the two norms are calculated via the definition $\|x\|_A := \sqrt{x^T A x}$ (for any positive semidefinite matrix $A \succeq 0$). When $\|\Sigma^{1/2} K^T T^{1/2}\| < 1$, convergence properties for this algorithm have been proved, e.g. Chambolle and Pock (2011); Pock and Chambolle (2011); He and Yuan (2012).¹ Setting $\theta = 0$ reduces to an earlier approach, the Primal-Dual Hybrid Gradient algorithm (Esser et al., 2009); with $\theta = 1$, the CP algorithm is equivalent to a modification of ADMM (discussed later in Section 3.1).

We now ask how we could modify the discussed methods to handle nonconvexity of F and/or G . One approach would be to approximate the problem with a convex optimization problem, that is, to take some approximation to F and G that are chosen to be a convex function. Of course, in general, there may not be a convex function that will provide a globally accurate approximation to a nonconvex function, but local convex approximations may be possible.

Consider a family of approximations to the functions F and G , indexed by $(z, v) \in \mathbb{R}^d \times \mathbb{R}^m$, the (primal) points at which the local approximations are taken. We write

1. The original form of the Chambolle-Pock algorithm (Chambolle and Pock, 2011), without preconditioning, can be obtained from (3) by replacing Σ, T with $\sigma I, \tau I$ for scalar step size parameters $\sigma, \tau > 0$, with convergence results proved if $\sigma\tau \|K\|^2 < 1$; however, in general the preconditioned form gives better performance and we only consider the preconditioned version here.

$F_v : \mathbb{R}^m \rightarrow \mathbb{R}$ and $G_z : \mathbb{R}^d \rightarrow \mathbb{R}$ for these approximations, and from this point on, we implicitly assume that the approximations satisfy, for any points $(z, v) \in \text{dom}(G) \times \text{dom}(F)$,

$$\begin{cases} \text{dom}(F_v) = \text{dom}(F) \text{ and } \text{dom}(G_z) = \text{dom}(G); \\ F_v \text{ and } G_z \text{ are convex and continuous functions}; \\ F_v \text{ and } G_z \text{ are accurate up to first order: } F - F_v \text{ and } G - G_z \text{ are differentiable,} \\ \text{with } (F - F_v)(v) = 0, \nabla(F - F_v)(v) = 0, (G - G_z)(z) = 0, \nabla(G - G_z)(z) = 0. \end{cases} \quad (4)$$

(Here $\text{dom}(\cdot)$ denotes the domain of a function, i.e. all points at which the function takes a finite value.) In particular this assumption implicitly requires that F and G both have convex domains.

As a special case, in some settings we can consider decomposing each function into the sum of a convex and a differentiable term, $F = F_{\text{cov}} + F_{\text{diff}}$ and $G = G_{\text{cov}} + G_{\text{diff}}$, as mentioned before; we can then take linear approximations to F_{diff} and to G_{diff} at the points v and z , respectively, to obtain

$$\begin{cases} F_v(w) := F_{\text{cov}}(w) + [F_{\text{diff}}(v) + \langle w - v, \nabla F_{\text{diff}}(v) \rangle], \\ G_z(x) := G_{\text{cov}}(x) + [G_{\text{diff}}(z) + \langle x - z, \nabla G_{\text{diff}}(z) \rangle]. \end{cases} \quad (5)$$

In particular, if F_{diff} and G_{diff} are both concave (and thus F and G are each convex/concave), these two approximations are standard convex majorizations to F and G taken at points v and z (as might be used in a majorization/minimization algorithm in some settings).

Since F_v, G_z are convex, we can substitute them in place of F, G in the iterations of the CP algorithm (3):

$$\begin{cases} x_{t+1} = \arg \min_x \{ \langle Kx, w_t \rangle + G_z(x) + \frac{1}{2} \|x - x_t\|_{\Sigma^{-1}}^2 \}, \\ w_{t+1} = \arg \min_w \{ -\langle K\bar{x}_{t+1}, w \rangle + F_v^*(w) + \frac{1}{2} \|w - w_t\|_{\Sigma^{-1}}^2 \}. \end{cases} \quad (6)$$

We will see later on (in Section 3.3) that this formulation is closely related to the ADMM algorithm, and in fact in the special case given in (5), if F_{diff} and G_{diff} are concave, then MOCCA can be viewed as a special case of Bregman ADMM (Wang and Banerjee, 2014; Wang et al., 2014a).

Of course, a key question remains: which primal points (z, v) should we use for constructing the convex approximations to F and to G , at iteration t of the algorithm? We find that, before solving for (x_{t+1}, w_{t+1}) , we should use expansion points

$$(z, v) = (x_t, \Sigma^{-1}(w_{t-1} - w_t) + K\bar{x}_t).$$

We will return shortly to the question of how these values were chosen; with this choice in place, the MOCCA algorithm is defined in Algorithm 1.

For reasons of stability, it may sometimes be desirable to update the expansion points (z, v) only periodically, and we will incorporate this option into a more general version of MOCCA, given in Algorithm 2. Specifically, at the t th stage, we repeat the (x, w) updates L_t many times; we refer to these repeated updates as the “inner loop”. In fact, the t th “inner

Algorithm 1 MOCCA algorithm

Input: Functions F, G with local convex approximations F_v, G_z , linear operator K , positive diagonal step size matrices Σ, T , extrapolation parameter $\theta \in [0, 1]$.

Initialize: Primal point $x_0 \in \mathbb{R}^d$, dual point $w_0 \in \mathbb{R}^m$, expansion points $(z_0, v_0) \in \mathbb{R}^d \times \mathbb{R}^m$.

for $t = 0, 1, 2, \dots$ **do**

Update x and w variables: writing $\bar{x}_{t+1} = x_{t+1} + \theta(x_{t+1} - x_t)$, define

$$\begin{cases} x_{t+1} = \arg \min_x \left\{ \langle Kx, w_t \rangle + G_z(x) + \frac{1}{2} \|x - x_t\|_{T^{-1}}^2 \right\}, \\ w_{t+1} = \arg \min_w \left\{ -\langle K\bar{x}_{t+1}, w \rangle + F_v^*(w) + \frac{1}{2} \|w - w_t\|_{\Sigma^{-1}}^2 \right\}. \end{cases}$$

Update expansion points: define

$$\begin{cases} z_{t+1} = x_{t+1}, \\ v_{t+1} = \Sigma^{-1}(w_t - w_{t+1}) + K\bar{x}_{t+1} \in \partial F_v^*(w_{t+1}). \end{cases}$$

until some convergence criterion is reached.

Algorithm 2 MOCCA algorithm (stable version with “inner loop”)

Input / Initialize: Same as for Algorithm 1, along with inner loop lengths L_1, L_2, \dots

for $\ell = 0, 1, 2, \dots$ **do**

Define $(x_{t+1;0}, w_{t+1;0}) = (x_t, w_t)$.

for $\ell = 1, 2, \dots, L_{t+1}$ **do**

Update x and w variables: writing $\bar{x}_{t+1;\ell} = x_{t+1;\ell} + \theta(x_{t+1;\ell} - x_{t+1;\ell-1})$, define

$$\begin{cases} x_{t+1;\ell} = \arg \min_x \left\{ \langle Kx, w_{t+1;\ell-1} \rangle + G_z(x) + \frac{1}{2} \|x - x_{t+1;\ell-1}\|_{T^{-1}}^2 \right\}, \\ w_{t+1;\ell} = \arg \min_w \left\{ -\langle K\bar{x}_{t+1;\ell}, w \rangle + F_v^*(w) + \frac{1}{2} \|w - w_{t+1;\ell-1}\|_{\Sigma^{-1}}^2 \right\}. \end{cases}$$

end for

Define $(x_{t+1}, w_{t+1}) = \frac{1}{L_{t+1}} \sum_{\ell=1}^{L_{t+1}} (x_{t+1;\ell}, w_{t+1;\ell})$.

Update expansion points: define

$$\begin{cases} z_{t+1} = \frac{1}{L_{t+1}} \sum_{\ell=1}^{L_{t+1}} x_{t+1;\ell}, \\ v_{t+1} = \frac{1}{L_{t+1}} \sum_{\ell=1}^{L_{t+1}} (\Sigma^{-1}(w_{t+1;\ell-1} - w_{t+1;\ell}) + K\bar{x}_{t+1;\ell}). \end{cases}$$

until some convergence criterion is reached.

loop” is simply running the CP algorithm for the convex problem $\min_x \{F_v(Kx) + G_z(x)\}$. Then, we average over the inner loop and calculate a single update of the expansion points (z, v) . Observe that, if we set $L_t = 1$ for all t , we do only a single (x, w) update in each “inner loop” and thus have reduced to the basic form of MOCCA. In practice, the basic form (Algorithm 1) performs well, and the more stable version (Algorithm 2) is primarily

proposed here for theoretical purposes, as some of our convergence guarantees do not hold for the basic version. However, in some settings the added stability does help empirically.

We remark that F, G (and their approximations F_v, G_z) are allowed to take the value $+\infty$, for instance, to reflect a constraint. For example we might have $G(x) = \delta(\|x\|_2 \leq 1)$, the convex indicator function taking the value $+\infty$ if the constraint $\|x\|_2 \leq 1$ is violated and zero otherwise; this has the effect of imposing a constraint on the x update step of our algorithm. Furthermore, in settings where F_v may not be strongly convex, its conjugate F_v^* may not be finitely valued; we would have $F_v^*(w) = +\infty$ (for some, but not all, w). For instance if $F_v(w) = \|w\|_1$ then $F_v^* = \delta\{\|w\|_\infty \leq 1\}$, which has the effect of imposing a constraint on w in the w update step of our algorithm. Our theoretical results in this paper hold across all these settings, i.e. we do not assume that any of the functions F, G, F_v, G_z, F_v^* are everywhere finite, but instead work in the domains of these functions.

1.1.1 SIMPLE SPECIAL CASES

Before discussing the implementation and behavior of MOCCA for general nonconvex and/or nonsmooth problems, we pause to illustrate that MOCCA can be viewed as a generalization of many existing techniques.

- If F, G are both convex with easy proximal maps, then we can of course choose the trivial convex families $F_v = F$ and $G_z = G$; the MOCCA algorithm then reduces to the Chambolle-Pock (Chambolle and Pock, 2011) or Primal-Dual Hybrid Gradient (Esser et al., 2009) algorithm (depending on our choice of the extrapolation parameter θ). These methods can handle composite objective functions with convex terms; MOCCA extends these methods to a setting with nonconvex terms.

- In the setting where we want to minimize a function $G(x)$ which is a sum of a convex term and a differentiable term, $G(x) = g(x) + h(x)$, we can show that MOCCA reduces to proximal gradient descent (Nesterov, 2013; Beck and Teboulle, 2009) as a special case. Specifically, we define the approximations $G_z(x) = g(x) + h(z) + \langle \nabla h(z), x - z \rangle$. In this setting there is no F function, and hence no w variable; taking $T = \tau \cdot \mathbf{I}_d$, the steps of Algorithm 1 become

$$\begin{cases} x_{t+1} = \arg \min_x \left\{ G_z(x) + \frac{1}{2\tau} \|x - x_t\|_2^2 \right\} \\ = \arg \min_x \left\{ g(x) + \langle \nabla h(z_t), x \rangle + \frac{1}{2\tau} \|x - x_t\|_2^2 \right\}, \\ z_{t+1} = x_{t+1}, \end{cases}$$

which simplifies to the update scheme

$$x_{t+1} = \text{Prox}_{\tau \cdot g}(x_t - \tau \cdot \nabla h(x_t)). \quad (7)$$

This is exactly the proximal gradient descent algorithm with step size τ .

Proximal gradient descent can handle a function which combines a differentiable term with a convex term as long as the convex term has an easy proximal map; MOCCA extends this method to a setting where the convex terms lack an easy proximal map due to linear transformations, leading to composite optimization problems.

We will discuss other existing methods in more detail later on in Section 3.

1.1.2 STEP SIZE PARAMETERS Σ AND \mathbf{T}

We now turn to the question of choosing the diagonal step size matrices Σ and \mathbf{T} . As we will see later on, good convergence properties are attained when Σ, \mathbf{T} are chosen sufficiently small, to satisfy $\|\Sigma^{1/2}KT^{1/2}\| < 1$ —this condition on Σ and \mathbf{T} is derived by Pock and Chambolle (2011) for the preconditioned CP algorithm, and appears in our theory as well. Here $\|\cdot\|$ is the matrix operator norm (i.e. the largest singular value). To choose matrices that satisfy this requirement, Pock and Chambolle (2011) propose a parametrized family of choices: after fixing some parameter $\lambda > 0$, define²

$$\Sigma_{ii} = \frac{\lambda}{\sum_j |K_{ij}|} \quad \text{and} \quad \mathbf{T}_{jj} = \frac{\lambda^{-1}}{\sum_i |K_{ij}|}. \quad (8)$$

Empirically, we find that higher values of λ are more stable but lead to slower convergence; it seems that the best choice is the smallest possible λ such that the algorithm does not diverge. It may also be interesting to consider varying λ adaptively over the iterations of the algorithm, but we do not study this extension here.

1.1.3 UNDERSTANDING THE CHOICE OF EXPANSION POINTS

We now return to our choice of the expansion points $(z, v) \in \mathbb{R}^d \times \mathbb{R}^m$. We will give an intuition for the choices of these points in the MOCCA algorithm. To examine this question, first consider the goal for optimization: we would like to find

$$x^* \in \arg \min_x \{F(Kx) + G(x)\},$$

or if this problem is nonconvex then we may be satisfied to let x^* be a local minimizer or critical point of this objective function. We then need to find primal points $z \in \mathbb{R}^d$ and $v \in \mathbb{R}^m$, such that replacing F with F_v and G with G_z still yields the same solution, i.e. so that

$$x^* \in \arg \min_x \{F_v(Kx) + G_z(x)\}. \quad (9)$$

Examining the first-order optimality conditions for each of these problems, it follows that we should set $(z, v) = (x^*, Kx^*)$ to ensure that (9) holds.

Of course, x^* is unknown and so we cannot set $(z, v) = (x^*, Kx^*)$. Instead, a logical approach would be to set $(z_i, v_i) = (x_i, Kx_i)$, before solving for (x_{i+1}, w_{i+1}) . Then, hopefully, as x_i converges to x^* we will also have (z_i, v_i) converging to (x^*, Kx^*) . However, in practice, we find that this approach does not always perform as well as expected. Specifically, the problem lies with the choice $v_i = Kx_i$, relative to the primal/dual structure of the algorithm.

To understand why, imagine that F and G are actually convex, but we nonetheless are taking local approximations F_v and G_z (which are also convex). Perhaps for computational reasons. Then we would like our x and w update steps (6) to coincide with the updates (3) of the original CP algorithm. Examining the optimality conditions, this will occur when

$$\partial G(x_{i+1}) = \partial G_{z_i}(x_{i+1}) \quad \text{and} \quad \partial F^*(w_{i+1}) = \partial F_{v_i}^*(w_{i+1}). \quad (10)$$

² In fact these choices for Σ, \mathbf{T} satisfy the matrix norm constraint more weakly, with $\|\Sigma^{1/2}KT^{1/2}\| \leq 1$ rather than a strict inequality, but this is sufficient in practice.

(Here we are ignoring issues of multivalued subdifferentials since our aim is only to give intuition.) Using the definitions of G and G_{z_i} , for the x step our requirement in (10) is equivalent to

$$\nabla(G - G_{z_i})(x_{i+1}) = 0,$$

which will certainly hold if $z_i = x_{i+1}$ by our assumption (4) on the function G_{z_i} . Since we have not yet solved for x_{i+1} , we instead choose the expansion point $z_i = x_i$ for the function G , as previously proposed.

For the w step, our outcome will be different. Subgradients satisfy a duality property, namely, $w \in \partial F(w)$ if and only if $u \in \partial F^*(w)$ for any convex function F and its conjugate F^* . The requirement $\partial F^*(w_{i+1}) = \partial F_{v_i}^*(w_{i+1})$ in (10) therefore yields $w_{i+1} \in \partial F(\partial F_{v_i}^*(w_{i+1}))$ by this duality property, and so we have

$$\begin{aligned} w_{i+1} &\in \partial F(\partial F_{v_i}^*(w_{i+1})) \\ &= \partial F_{v_i}(\partial F_{v_i}^*(w_{i+1})) + \nabla(F - F_{v_i})(\partial F_{v_i}^*(w_{i+1})) \\ &= w_{i+1} + \nabla(F - F_{v_i})(\partial F_{v_i}^*(w_{i+1})) \end{aligned}$$

where the last step again holds from the duality property of subgradients. So, we see that we would like

$$\nabla(F - F_{v_i})(\partial F_{v_i}^*(w_{i+1})) = 0$$

which, according to our assumption (4) on the expansions F_{v_i} , will hold if

$$v_i \in \partial F_{v_i}^*(w_{i+1}).$$

In other words, we would like v_i to be the primal point that corresponds to the dual point w_{i+1} —that is, the primal point that *mirrors* the dual point w_{i+1} . Of course, this is not possible since we have not yet computed w_{i+1} , and furthermore v_i appears on both sides of this equation. Instead, we take $v_i \in \partial F_{v_i-1}^*(w_i)$. Looking at the first-order optimality conditions for the update step for w_i , we see that we can satisfy this expression by choosing

$$v_i = \Sigma^{-1}(w_{i-1} - w_i) + Kx_i \in \partial F_{v_i-1}^*(w_i).$$

In fact, we will see in Section 3.3 that this choice for v_i is very logical in light of the connection between the CP algorithm and the Alternating Direction Method of Multipliers (ADMM) (Boyd et al., 2011).

For the stable form of MOCCA, Algorithm 2, our choice for expansion points (z, v) takes an average over each inner loop, which we will see gives sufficient stability for our convergence results to hold.

1.1.4 CHECKING CONVERGENCE

Here we give a simple way to check whether the basic MOCCA algorithm, Algorithm 1, is near a critical point (e.g. a local minimum). (We treat this question more formally, for the more general Algorithm 2, in our theoretical results later on.) Due to the first-order accuracy of the convex approximations to F and G as specified in (4), a critical point $x \in \mathbb{R}^d$ for the objective function $F(Kx) + G(x)$ is characterized by the first-order condition

$$0 \in K^\top \partial F_{Kx}(Kx) + \partial G_x(x). \quad (11)$$

Equivalently, we can search for a dual variable $w \in \mathbb{R}^m$ such that

$$-K^\top w \in \partial \mathbf{G}_x(x) \quad \text{and} \quad w \in \partial \mathbf{F}_{Kx}(Kx).$$

We can expand this condition to include additional variables $(z, v) \in \mathbb{R}^d \times \mathbb{R}^m$:

$$\begin{cases} -K^\top w \in \partial \mathbf{G}_z(x), \\ w \in \partial \mathbf{F}_v(Kx) \Leftrightarrow Kx \in \partial \mathbf{F}_v^*(w), \\ z = x, \\ v = Kx. \end{cases}$$

To check whether these conditions hold approximately, we can take the following ‘‘optimality gap’’:

$$\text{OptimalityGap}(x, w, z, v) = \left\| -K^\top w - \partial \mathbf{G}_z(x) \right\|_2^2 + \left\| Kx - \partial \mathbf{F}_v^*(w) \right\|_2^2 + \|z - x\|_2^2 + \|v - Kx\|_2^2.$$

Here, if any of the subdifferentials are multivalued, we can interpret these norms as choosing some element of the corresponding subdifferentials. Now we consider the value of this gap at an iteration of the MOCCA algorithm (in its original form, Algorithm 1). By the definitions of $x_{t+1}, w_{t+1}, z_t, v_t$, we can show that

$$\begin{cases} 0 \in K^\top w_t + \partial \mathbf{G}_{z_t}(x_{t+1}) + \mathbf{T}^{-1}(x_{t+1} - x_t), \\ 0 \in -K\bar{x}_{t+1} + \partial \mathbf{F}_{v_t}^*(w_{t+1}) + \Sigma^{-1}(w_{t+1} - w_t), \\ z_t = x_t, \\ v_t = \Sigma^{-1}(w_t - w_{t-1}) + K\bar{x}_t. \end{cases}$$

Therefore, plugging these calculations in to the definition of the optimality gap, we see that

$$\begin{aligned} \text{OptimalityGap}(x_{t+1}, w_{t+1}, z_t, v_t) &= \left\| -K(w_t - w_{t+1}) + \mathbf{T}^{-1}(x_t - x_{t+1}) \right\|_2^2 + \left\| K(x_t - x_{t+1}) + \Sigma^{-1}(w_t - w_{t+1}) \right\|_2^2 \\ &\quad + \|x_t - x_{t+1}\|_2^2 + \left\| K(x_{t-1} - 2x_t + x_{t+1}) + \Sigma^{-1}(w_{t-1} - w_t) \right\|_2^2 \\ &= \mathcal{O} \left(\left\| \begin{pmatrix} x_{t-1} - x_t \\ w_{t-1} - w_t \end{pmatrix} \right\|_2^2 + \left\| \begin{pmatrix} x_t - x_{t+1} \\ w_t - w_{t+1} \end{pmatrix} \right\|_2^2 \right). \end{aligned}$$

In other words, if the change in the variables (x_t, w_t) converges to zero as $t \rightarrow \infty$, then the optimality gap is also converging to zero.

1.1.5 PREVIEW OF THEORETICAL RESULTS

We present two theoretical results in this work. The first is fairly standard in the related literature: in Theorem 1 we show that if the algorithm does converge to a point, then we have reached a critical point of the original optimization problem. (Since the simple form, Algorithm 1, is a special case of the stable form, Algorithm 2, we prove this result for the stable algorithm only.)

The novelty of our theory lies in our convergence guarantee, given in Theorem 2, where we prove that the stable form of MOCCA, given in Algorithm 2, is guaranteed to converge to a nearly-globally-optimal solution, under some assumptions on convexity and curvature. Specifically, we consider a scenario where convexity and nonconvexity in \mathbf{F} and \mathbf{G} counter-balance each other, so that the overall function

$$x \mapsto \mathbf{F}(Kx) + \mathbf{G}(x) \quad (12)$$

is itself either strongly convex or satisfies restricted strong convexity assumptions (which we will discuss in detail in Section 4.2.1). It is important to note that even the globally convex setting is by no means trivial—even if (12) is strongly convex, if \mathbf{F} itself is nonconvex it may be the case that ADMM and other primal/dual or alternating minimization algorithms diverge or converge to the wrong solution, as we discuss later in Section 3.2. Crucially, our results allow \mathbf{F} and \mathbf{G} to be nondifferentiable as well as nonconvex, a setting that is necessary in practice but is not covered by existing theory.

1.1.6 OUTLINE OF PAPER

The remainder of the paper is organized as follows. In Section 2, we present several important applications where the minimization problem considered here, with a nonconvex composite objective function as in (1), arises naturally: regression problems with errors in covariates, isotropic total variation penalties, nonconvex total variation penalties, and image reconstruction problems in computed tomography (CT) imaging. In Section 3 we give background on several types of existing algorithms for convex and nonconvex composite objective functions, and compare a range of existing results to the work presented here. Theoretical results on the convergence properties of our algorithm are given in Section 4. We study the empirical performance of MOCCA in Section 5. Proofs are given in Section 6, with technical details deferred to the Appendix. In Section 7 we discuss our findings and outline directions for future research.

2. Applications

We now highlight several applications of the MOCCA algorithm, in high-dimensional statistics and in imaging.

2.1 Regression with Errors in Variables

Recent work by Loh and Wainwright (2013) considers optimization for nonconvex statistical problems, proving that under some special conditions, nonconvexity may not pose a challenge to recovering optimal parameters. In particular, they consider the following example (Loh and Wainwright, 2011, 2013): suppose that we observe a response $y \in \mathbb{R}^n$ which is generated with a Gaussian linear model,

$$b = Ax_{\text{true}} + \epsilon \quad \text{with} \quad \epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_n),$$

where $A \in \mathbb{R}^{n \times d}$ is a design matrix and $x_{\text{true}} \in \mathbb{R}^d$ is the unknown vector of coefficients. In this case, we might seek to recover x_{true} with the least squares estimator, perhaps with

some penalty added to promote some desired structure in x ,

$$\begin{aligned} \hat{x} = \arg \min_x & \left\{ \frac{1}{2\eta} \|b - Ax\|_2^2 + \text{Penalty}(x) \right\} \\ & = \arg \min_x \left\{ \frac{1}{2} x^\top \left(\frac{A^\top A}{\eta} \right) x - x^\top \left(\frac{A^\top b}{\eta} \right) + \text{Penalty}(x) \right\}. \end{aligned} \quad (13)$$

In some settings, however, the design matrix A itself may not be known with perfect accuracy. Instead, suppose we observe

$$Z = A + W$$

where $W \perp A$ has independent mean-zero entries, with $\mathbb{E}[W_{ij}^2] = \sigma_A^2$ for all i, j . In this case, a naive approach might be to substitute Z for A in (13), before finding the minimizer. However, unless σ_A^2 is negligible, this may not produce a good approximation to \hat{x} since, when substituting $Z^\top Z$ for $A^\top A$ in the quadratic term in (13), we have

$$\mathbb{E} \left[Z^\top Z \mid A \right] = \mathbb{E} \left[(A + W)^\top (A + W) \mid A \right] = A^\top A + \mathbb{E} \left[W^\top W \right] = A^\top A + n\sigma_A^2 \mathbf{I}_d \neq A^\top A.$$

In contrast, for the linear term in (13), we have $\mathbb{E}[Z^\top b \mid A] = A^\top b$, as desired. To correct for the bias in $Z^\top Z$, we should take

$$\hat{x}_{\text{noisy}} = \arg \min_x \{ \mathcal{L}(x) + \text{Penalty}(x) \},$$

where

$$\mathcal{L}(x) := \frac{1}{2} x^\top \left(Z^\top Z - \sigma_A^2 \mathbf{I}_d \right) x - x^\top \left(Z^\top b \right).$$

Of course, this optimization problem is no longer convex due to the negative quadratic term, and in particular, for a Lipschitz penalty and a high-dimensional setting ($n < d$), the value tends to $-\infty$ as x grows large in any direction in the null space of Z . Remarkably, Loh and Wainwright (2013) show that, if x_{true} is sparse and $\text{Penalty}(x)$ is similar to the ℓ_1 norm, then as long as $(Z^\top Z)$ satisfies a restricted strong convexity assumption (as is standard in the sparse regression literature), then x_{true} can be accurately recovered from *any* local minimum or critical point of the constrained optimization problem

$$\hat{x}_{\text{noisy}} = \arg \min_{\|x\|_1 \leq R} \{ \mathcal{L}(x) + \text{Penalty}(x) \}. \quad (14)$$

The approach taken by Loh and Wainwright (2013) is to perform proximal gradient descent, with steps taking the form

$$x_{t+1} = \arg \min_{\|x\|_1 \leq R} \left\{ \frac{1}{2} \left\| x - \left(x_t - \frac{1}{\eta} \nabla \mathcal{L}(x_t) \right) \right\|_2^2 + \frac{1}{\eta} \text{Penalty}(x) \right\}$$

where $\frac{1}{\eta}$ is a step size parameter. When $\text{Penalty}(x)$ is (some multiple of) the ℓ_1 norm, or some other function with a simple proximal operator (that is, it is simple to compute $\arg \min_x \{ \frac{1}{2} \|x - z\|_2^2 + \text{Penalty}(x) \}$ for any z), this algorithm is very efficient.

2.1.1 TOTAL VARIATION AND GENERALIZED CONVEX ℓ_1 PENALTIES

Consider a setting where the penalty term in (14) is given by a generalized ℓ_1 norm,

$$\text{Penalty}(x) = \nu \cdot \|Kx\|_1$$

for some matrix $K \in \mathbb{R}^{m \times d}$. In particular, if K is the one-dimensional differences matrix $\nabla_{1d} \in \mathbb{R}^{(d-1) \times d}$, which has entries $(\nabla_{1d})_{ii} = 1$ and $(\nabla_{1d})_{i,i+1} = -1$ for each i , then this defines the (one-dimensional) total variation norm on x , $\|x\|_{\text{TV}} = \|\nabla_{1d} x\|_1$; this method is also known as the fused Lasso (Tibshirani et al., 2005). We can also consider a two- or three-dimensional total variation norm, $K = \nabla_{2d}$ or $K = \nabla_{3d}$, defined analogously as the differences matrix for a two- or three-dimensional grid. Total variation type penalties are commonly used in imaging applications and many other fields to obtain solutions that are locally constant or locally smooth.

In this setting, proximal gradient descent is not practical except in some special cases, such as when K is diagonal, because the proximal operator $\arg \min_x \{ \frac{1}{2} \|z - x\|_2^2 + \nu \cdot \|Kx\|_1 \}$ does not have a closed form solution for general K and would itself require an iterative algorithm to be run to convergence. For a total variation penalty on a one-dimensional grid of points, e.g. $K = \nabla_{1d}$, some fast algorithms do exist for the proximal map (Johnson, 2013). Additional methods for convex problems with two-dimensional total variation and related penalties such as total variation over a graph can be found in Chambolle and Darbon (2009); Wang et al. (2014b, 2015b). We are not aware of a non-iterative algorithm for general K . Here we apply MOCCA to allow for arbitrary K and for a nonconvex loss term $\mathcal{L}(x)$.

2.1.2 APPLYING THE MOCCA ALGORITHM

We consider applying the MOCCA algorithm to this nonconvex optimization problem with $\text{Penalty}(x) = \nu \|Kx\|_1$. We define the convex function

$$F(w) = \nu \|w\|_1$$

with the trivial convex approximations $F_\nu(w) = F(w)$ at any expansion point $v \in \mathbb{R}^m$. We also let

$$G(x) = \begin{cases} \mathcal{L}(x), & \text{if } \|x\|_1 \leq R, \\ +\infty, & \text{if } \|x\|_1 > R, \end{cases}$$

with convex approximations given by taking the linear approximation to the loss,

$$G_z(x) = \begin{cases} \mathcal{L}(z) + \langle x - z, \nabla \mathcal{L}(z) \rangle, & \text{if } \|x\|_1 \leq R, \\ +\infty, & \text{if } \|x\|_1 > R, \end{cases}$$

for any expansion point $z \in \mathbb{R}^d$. Then the optimization problem (14) can be expressed as

$$\hat{x}_{\text{noisy}} = \arg \min_x \{ F(Kx) + G(x) \}.$$

Applying Algorithm 1, the update steps take the form

$$\begin{cases} x_{t+1} = \arg \min_{\|x\|_1 \leq R} \left\{ \|x - [x_t - \nabla(\mathcal{L}(x_t) + K^\top w_t)]\|_{l_1}^2 \right\}, \\ w_{t+1} = \text{Truncate}_\nu(w_t + \Sigma \hat{K} x_{t+1}), \\ z_{t+1} = x_{t+1}. \end{cases} \quad (15)$$

where $\text{Truncate}_\nu(w)$ truncates the entries of the vector w to the range $[-\nu, \nu]$. Note that the x update step is a simple shrinkage step and therefore easy to solve (and $\nabla \mathcal{L}(x_t)$ is simple to compute), while the w and z updates are computationally trivial.

As a second option, we can incorporate more convexity into our approximations \mathbf{G}_z by taking

$$\mathbf{G}_z(x) = \begin{cases} \mathcal{L}(x) + \frac{\sigma_z^2}{2} \|x - z\|_2^2, & \text{if } \|x\|_1 \leq R, \\ +\infty, & \text{if } \|x\|_1 > R, \end{cases}$$

which is convex since $\mathcal{L}(x)$ has negative curvature bounded by $\frac{\sigma_z^2}{2}$. In this case, after simplifying, our update steps become

$$\begin{cases} x_{t+1} = \arg \min_{\|x\|_1 \leq R} \left\{ \left\| x - \left[x_t - \left(\mathbf{T}^{-1} + \frac{z^T z}{n} \right)^{-1} (\nabla \mathcal{L}(x_t) + K^T w_t) \right] \right\|_{\left(\mathbf{T}^{-1} + \frac{z^T z}{n} \right)} \right\}^2, \\ w_{t+1} = \text{Truncate}_\nu(w_t + \Sigma K \bar{x}_{t+1}), \\ z_{t+1} = x_{t+1}. \end{cases} \quad (16)$$

While the x update step may appear difficult due to the combination of the non-diagonal matrix $\left(\mathbf{T}^{-1} + \frac{z^T z}{n} \right)$ which scales the norm, combined with the ℓ_1 constraint, in practice the constraint R is chosen to be large so that it is inactive in all or most steps; the x update step is then solved by $x_{t+1} = x_t - \left(\mathbf{T}^{-1} + \frac{z^T z}{n} \right)^{-1} (\nabla \mathcal{L}(x_t) + K^T w_t)$.

An important point is that MOCCA can be applied to this problem for arbitrary K , including a difference operator such as \mathbf{V}_{2d} or \mathbf{V}_{3d} ; in contrast, proximal gradient descent can only be performed approximately except for certain special cases, as mentioned above. We explore this setting's theoretical properties in Section 4.2.3, and give empirical results for this problem in Section 5.

2.2 Isotropic Total Variation and Generalized ℓ_1/ℓ_2 Penalties

For locally constant images or signals in two dimensions, the form of the total variation penalty given above is known as ‘anisotropic’, meaning that it imposes a sparsity pattern which is specific to the alignment of the image onto a horizontal and vertical axis. In contrast, the isotropic total variation penalty (Rudin et al., 1992), on an image x parametrized with values $x_{i,j}$ at grid location (i, j) , is given by

$$\|x\|_{\text{isoTV}} = \sum_{(i,j)} \sqrt{(x_{i,j} - x_{i,j+1})^2 + (x_{i,j} - x_{i+1,j})^2}.$$

Optimization methods for the denoising problem with an isotropic total variation penalty, i.e. problems of the form $\min_x \left\{ \frac{1}{2} \|b - x\|_2^2 + \nu \cdot \|x\|_{\text{isoTV}} \right\}$, were studied in Chambolle and Pock (2015). In practice the isotropic penalty is often preferred as it leads to smoother contours, avoiding the artificial horizontal or vertical edges that may result from anisotropic total variation regularization.

The isotropic total variation penalty can be generalized to penalties of the form

$$\text{Penalty}(x) = \nu \cdot \sum_{\ell=1}^L \|K_\ell x\|_2,$$

where each $K_\ell \in \mathbb{R}^{m_\ell \times d}$ is some fixed matrix. To see how this determines an isotropic total variation penalty in two dimensions, we let ℓ index all locations (i, j) ; then the corresponding matrix K_ℓ has two rows, which when multiplying the image x , extracts the differences $x_{i,j} - x_{i,j+1}$ and $x_{i,j} - x_{i+1,j}$. To see how this specializes to the usual generalized ℓ_1 penalty, $\|Kx\|_1$ for some fixed matrix $K \in \mathbb{R}^{m \times d}$, simply take K_ℓ to be the ℓ th row of the matrix K for each $\ell = 1, \dots, m$.

2.2.1 APPLYING THE MOCCA ALGORITHM

We now show the steps of the MOCCA algorithm to the problem of minimizing

$$\mathcal{L}(x) + \nu \cdot \sum_{\ell=1}^L \|K_\ell x\|_2, \quad (17)$$

where $\mathcal{L}(x)$ is a differentiable likelihood term (such the nonconvex likelihood for regression with errors in variables as above). We define the matrix K by vertically stacking the K_ℓ 's, and define the convex function

$$\mathbf{F}(w) = \nu \cdot \sum_{\ell=1}^L \|w_{B_\ell}\|_2,$$

where w_{B_ℓ} is understood to be the ℓ th block of the vector w , i.e.

$$w_{B_\ell} = (w_{m_1+\dots+m_{\ell-1}+1}, \dots, w_{m_1+\dots+m_\ell}).$$

We take the trivial convex approximations $\mathbf{F}_v(w) = \mathbf{F}(w)$ at any expansion point $v \in \mathbb{R}^m$, and define $\mathbf{G}(x)$ and $\mathbf{G}_z(x)$ exactly as before (as for the previous application, we allow the option of restricting to $\|x\|_1 \leq R$ if desired). Then the objective function (17) is equivalent to minimizing $\mathbf{F}(Kx) + \mathbf{G}(x)$.

Applying Algorithm 1, the update steps take the form

$$\begin{cases} x_{t+1} = \arg \min_{\|x\|_1 \leq R} \left\{ \|x - [x_t - \mathbf{T}(\nabla \mathcal{L}(x_t) + K^T w_t)]\|_{\mathbf{T}^{-1}} \right\}^2, \\ w_{t+1} = \text{Truncate}_{\nu, (\mathbb{B}_{m_1} \times \dots \times \mathbb{B}_{m_L})} (w_t + \Sigma K \bar{x}_{t+1}), \\ z_{t+1} = x_{t+1}, \end{cases}$$

where $\text{Truncate}_{\nu, (\mathbb{B}_{m_1} \times \dots \times \mathbb{B}_{m_L})}(w)$ projects each block $w_{B_\ell} \in \mathbb{R}^{m_\ell}$ of the vector w to the ball of radius ν , $\nu \cdot \mathbb{B}_{m_\ell} \subseteq \mathbb{R}^{m_\ell}$ (here \mathbb{B}_{m_ℓ} is the unit ball of dimension m_ℓ in the ℓ_2 norm). Specifically, the w update step can be computed for each block as

$$(w_{t+1})_{B_\ell} = (w_t + \Sigma K \bar{x}_{t+1})_{B_\ell} \cdot \min \left\{ 1, \frac{\nu}{\|(w_t + \Sigma K \bar{x}_{t+1})_{B_\ell}\|_2} \right\}$$

for each $\ell = 1, \dots, L$, and is therefore trivial to compute.

2.3 Nonconvex Total Variation Penalties

As discussed in Section 2.1, total variation penalties are common in many applications where the underlying signal exhibits smooth or locally constant spatial structure (in one, two, or more dimensions). In convex optimization, we are faced with a well-known tradeoff between sparsity and bias—using $\nu \cdot \|x\|_{\text{TV}}$ as our penalty function for some parameter $\nu > 0$, we want to be sure to choose ν large enough that the resulting solution is total-variation-sparsity, to avoid overfitting when the sample size is small relative to the dimension of the image; however, larger ν leads to increased shrinkage of the signal, leading to an estimate that is biased towards zero. One way to avoid this tradeoff is to use a nonconvex penalty, which should behave like the total variation norm in terms of promoting sparsity, but reduce the amount of shrinkage for larger signal strength. In this section, we will use ∇_{TV} to denote the differences matrix in the appropriate space (e.g. $\nabla_{\text{TV}} = \nabla_{2d}$ in two dimensions), so that $\|x\|_{\text{TV}} = \|\nabla_{\text{TV}}x\|_1$.

For sparse regression problems (i.e. where the signal x is itself sparse, rather than sparsity in $\nabla_{\text{TV}}x$), many nonconvex alternatives to the ℓ_1 norm penalty $\|x\|_1$ have been studied, demonstrating more accurate signal recovery empirically as well as theoretical properties of reduced bias, such as the Smoothly Clipped Absolute Deviation (SCAD) penalty (Fan and Li, 2001), the ℓ_q penalty which penalizing $\sum_i |x_i|^q$ for some $q \in (0, 1)$ (Knight and Fu, 2000; Chartrand, 2007), and the Minimax Concave Penalty (MCP) which seeks to minimize concavity while avoiding bias (Zhang, 2010). Another option is to use a reweighted ℓ_1 norm (Candès et al., 2008), where signals estimated to be large at the first pass are then penalized less in the next pass to reduce bias in their estimates; in fact, Candès et al. (2008) show that this procedure is related to a nonconvex log-sum penalty, given by penalizing each component of x_i as $\log(|x_i| + \epsilon)$ for some fixed $\epsilon > 0$. For the problem of total variation sparsity, a variety of nonconvex approaches have also been studied, including applying SCAD (Chopra and Lian, 2010), an ℓ_q norm penalty for $0 < q < 1$ (Sidky et al., 2014; Lu and Hwang, 2014), or a log-sum total variation penalty (Selesnick et al., 2015; Parekh and Selesnick, 2015) to the total variation sparsity setting.

We now consider the problem applying a log-sum penalty to the problem of total variation sparsity. Here we consider the form of this penalty given by

$$\log \text{TV}_{\beta}(x) = \log \mathbb{L}_{\beta}(\nabla_{\text{TV}}x) \text{ where } \log \mathbb{L}_{\beta}(w) = \sum_i \beta \log(1 + |w_i|/\beta),$$

where $\beta > 0$ is a nonconvexity parameter. (We can also consider applying this nonconvex penalty to the isotropic version of total variation, as discussed in Section 2.2, but for simplicity we do not give that version explicitly here.)

To understand this function, observe that for any t , the function

$$t \mapsto \beta \log(1 + |t|/\beta)$$

is approximately equal to $|t|$ when $t \approx 0$ (that is, near zero it behaves like the ℓ_1 norm), but is nonconvex and penalizes large values of t much less heavily than an absolute value penalty of $|t|$. The parameter β controls the amount of nonconvexity: small β gives a highly nonconvex penalty, while for large β the penalty is nearly convex, with $\log \text{TV}_{\beta}(x) \approx \|x\|_{\text{TV}}$; see Figure 1 for an illustration.

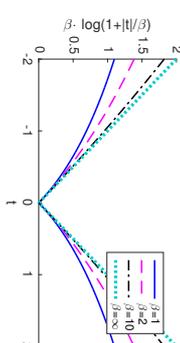


Figure 1: Illustration of the nonconvex sparsity-promoting penalty discussed in Section 2.3. The figure plots the function $t \mapsto \beta \cdot \log(1 + |t|/\beta)$ across a range of values of t , for $\beta \in \{1, 2, 10, \infty\}$; for $\beta = \infty$, we should interpret this as the absolute value function, $t \mapsto |t|$. We see that all the functions appear similar for $t \approx 0$, with a nondifferentiable point at $t = 0$ which ensures sparsity when this function is used as a penalty. For larger values of t , smaller values of β correspond to greater nonconvexity.

Consider the problem of minimizing an objective function

$$\mathcal{L}(x) + \nu \cdot \log \text{TV}_{\beta}(x), \quad (18)$$

where $\mathcal{L}(x)$ is some likelihood or loss term. In the image denoising setting, where $\mathcal{L}(x) = \frac{1}{2} \|y - x\|_2^2$ (i.e. when y is a noisy observation of the signal x), Parekh and Selesnick (2015) approach this optimization problem with a majorization/minimization algorithm, iterating the steps: (1) find a majorization of $\log \text{TV}_{\beta}(x)$ at the current estimate x_n , which takes the form of a reweighted TV norm, (2) compute x_{n+1} as the minimizer of $\mathcal{L}(x) + \nu \cdot$ (the majorized penalty). In other settings, however, for a general loss $\mathcal{L}(x)$, step (2) may not be possible.

We now show how the MOCCA algorithm can be used to optimize objective functions of the form (18). For simplicity we show the steps for the case that $\mathcal{L}(x)$ is convex and has a simple proximal operator, but this can be generalized as needed.

First, we define a new function

$$h_{\beta}(w) = \log \mathbb{L}_{\beta}(w) - \|w\|_1 = \sum_i \beta \log(1 + |w_i|/\beta) - \|w\|_1.$$

We note an important property of this function: $h_{\beta}(w)$ is differentiable with $(\nabla h_{\beta}(w))_i = \frac{-\frac{w_i}{|w_i|}}{\beta + |w_i|}$. For a first approach, we will take

$$F(w) = \nu \cdot \log \mathbb{L}_{\beta}(w) \text{ and } G(x) = \mathcal{L}(x).$$

Now we define a local convex approximation to F by writing $F(w) = \nu (\|w\|_1 + h_{\beta}(w))$ and taking the linear approximation to h_{β} , namely,

$$F_v(w) = \nu \cdot \|w\|_1 + \nu [h_{\beta}(v) + \langle w - v, \nabla h_{\beta}(v) \rangle].$$

And, since by assumption $\mathcal{L}(x)$ is convex and has a simple proximal operator, we can simply define

$$\mathbf{G}_z(x) = \mathbf{G}(x) = \mathcal{L}(x).$$

Then the objective function (18) is equal to $\mathbf{F}(\nabla_{\text{TV}}x) + \mathbf{G}(x)$, and can be minimized with MOCCA. Applying Algorithm 1, the update steps take the form

$$\begin{cases} x_{t+1} = \arg \min_x \left\{ \mathcal{L}(x) + \frac{1}{2} \|x - [x_t - \text{TV}_{\text{TV}}^{\top} w_t]\|_{\Gamma^{-1}}^2 \right\}, \\ w_{t+1} = \text{Truncate}_{\nu} (w_t + \Sigma \nabla_{\text{TV}} \bar{x}_{t+1} - \nu \nabla h_{\beta}(v_t)) + \nu \nabla h_{\beta}(v_t), \\ v_{t+1} = \Sigma^{-1}(w_t - w_{t+1}) + K \bar{x}_{t+1}. \end{cases}$$

Of course, we have the flexibility to arrange the functions differently if we wish—for example, we could instead define

$$\mathbf{F}(w) = \nu \cdot \|w\|_1 \quad \text{and} \quad \mathbf{G}(x) = \mathcal{L}(x) + \nu \cdot h_{\beta}(\nabla_{\text{TV}}x).$$

In this case, we will define the local approximations as

$$\mathbf{F}_v(w) = \mathbf{F}(w) = \nu \cdot \|w\|_1$$

and

$$\mathbf{G}_z(w) = \mathcal{L}(x) + \nu [h_{\beta}(\nabla_{\text{TV}}z) + \langle \nabla_{\text{TV}}(x - z), \nabla h_{\beta}(\nabla_{\text{TV}}z) \rangle].$$

In this case the objective function (18) is equal to $\mathbf{F}(\nabla_{\text{TV}}x) + \mathbf{G}(x)$ as before. In this case, the update steps are

$$\begin{cases} x_{t+1} = \arg \min_x \left\{ \mathcal{L}(x) + \frac{1}{2} \|x - [x_t - \text{TV}_{\text{TV}}^{\top}(\nu \nabla h_{\beta}(\nabla_{\text{TV}}z_t) + w_t)]\|_{\Gamma^{-1}}^2 \right\}, \\ w_{t+1} = \text{Truncate}_{\nu} (w_t + \Sigma \nabla_{\text{TV}} \bar{x}_{t+1}), \\ z_{t+1} = x_{t+1}. \end{cases}$$

However, in a sense this decomposition is less natural as it splits the penalty $\log \text{TV}_{\beta}(x)$ across \mathbf{F} and \mathbf{G} , and in fact in our experiments in Section 5, we will see that this second formulation gives poorer convergence results when the MOCCA algorithm is applied.

2.4 Application to CT Image Reconstruction

The initial motivation for developing the algorithm presented here, is a problem arising in computed tomography (CT) imaging. Here we briefly summarize the problem and our approach via the MOCCA algorithm; we describe this setting fully, and give detailed results, in Barber et al. (2016).

In CT imaging, an X-ray beam is sent along many rays through the object of interest. Typically, the measurement is the total energy that has passed through the object, along each ray; comparing the energy retrieved against the energy of the entering X-ray beam, gives information about the materials inside the object, since different materials have different beam attenuation properties. A recent technological development is the photon counting

detector, which measures the raw number of photons that successfully pass through the object rather than the total integrated energy. As a first pass, this transmission model can be written as follows, where the location $\vec{r}_{\ell}(t)$ inside the object parametrizes the ray ℓ :

$$\text{Count}_{\ell} \sim \text{Poisson} \left(\int_{\text{energy } E} \left(\frac{\text{beam intensity at energy } E}{\text{detector sensitivity (to energy } E)} \right) \cdot \exp \left\{ - \int_t \mu(E, \vec{r}_{\ell}(t)) dt \right\} dE \right), \quad (19)$$

where the only unknowns are the coefficients $\mu = (\mu(E, \vec{r}))$, indexed over energy level E and location \vec{r} inside the object. Here $\mu(E, \vec{r})$ is the attenuation for photons at energy E at the location \vec{r} —higher values of $\mu(E, \vec{r})$ indicate that a photon (at energy E) is more likely to be absorbed (at location \vec{r}). The expression $\exp\{-\int_t \mu(E, \vec{r}_{\ell}(t)) dt\}$ determines, for a photon at energy E that enters the object along the trajectory defined by ray ℓ , the probability that the photon will not be absorbed by the object (i.e. will pass through the object). These coefficients $\mu(E, \vec{r})$ can be further decomposed as

$$\mu(E, \vec{r}) = \sum_{\text{materials } m} \mu_m(E) \cdot x_m(\vec{r}) \quad (20)$$

where $\mu_m(E)$ is a known quantity determining the absorption properties of material m at energy E , while $x_m(\vec{r})$ is unknown, representing the amount of material m that is present at location \vec{r} . In practice, the object space is discretized into pixels, so that x is finite-dimensional.

In this application, the optimization problem is then given by

$$\hat{x} = \arg \min_x \left\{ \mathcal{L}(x) + \sum_{\text{materials } m} (\text{total variation constraint on material map } x_m) \right\},$$

perhaps with other constraints added as well (e.g. nonnegativity of the material maps). Here $\mathcal{L}(x)$ is the negative log-likelihood of x given the Poisson model for the observed photon counts as a function of x , given by (19) and (20). $\mathcal{L}(x)$ is a nonconvex function due to the integration across energy levels. Both $\mathcal{L}(x)$ and the total variation constraints are better represented as functions of linear transformations of x ; the presence of these multiple terms, including nonconvexity (from \mathcal{L}) and nondifferentiability (from total variation), mean that existing methods cannot be applied to solve this optimization problems.

The MOCCA algorithm, applied to this problem, gives strong performance in terms of fast convergence and accurate image reconstruction on simulated and real imaging data. We do not give the details of the algorithm implementation or any empirical results in this paper, but instead refer the reader to our work in Barber et al. (2016) for more details on the method and for empirical results on simulated CT image data (results on real data are forthcoming).

3. Background: Optimizing Convex and Nonconvex Composite Functions

In this sections, we give background on several related algorithms for solving the minimization problem in the simpler setting where \mathbf{F} and \mathbf{G} are convex, and in the more challenging nonconvex setting.

3.1 Optimization When F is Convex

When F is convex, a variety of existing methods are available to recover the (possibly non-unique) minimizer

$$x^* = \arg \min \{F(Kx) + G(x)\}.$$

In many settings, the functions F and G may each have easily computable proximal operators, but the linear transformation inside $x \rightarrow F(Kx)$ might make the function difficult to optimize directly—for example, we might have $F(u) = \|u\|_1$ with $K = \nabla_{2d}$ chosen so that $F(Kx) = \|x\|_{TV}$, the two-dimensional total variation norm of x . For this type of setting, the Alternating Direction Method of Multipliers (ADMM) reframes the problem as

$$(x^*, u^*) = \arg \min_{x, u} \{F(u) + G(x) : Kx = u\}$$

and solves for (x^*, u^*) by working with the augmented Lagrangian

$$\min_{x, u} \max_{\Delta} L(x, u, \Delta) = \left\{ F(u) + G(x) + \langle \Delta, Kx - u \rangle + \frac{\rho}{2} \|Kx - u\|_2^2 \right\}.$$

Here $\Delta \in \mathbb{R}^m$ is the dual variable whose role is to enforce the constraint $Kx = u$. The steps of the algorithm are, for each $t \geq 1$,

$$\begin{cases} x_t = \arg \min_x \{L(x, u_{t-1}, \Delta_{t-1})\}, \\ u_t = \arg \min_u \{L(x_t, u, \Delta_{t-1})\}, \\ \Delta_t = \Delta_{t-1} + \rho(Kx_t - u_t). \end{cases}$$

Examining the update step for for u , we see that this step entails a single use of the proximal map for F. However, the x update step is more complicated due to the linear operator K ; this step cannot be solved with one use of the proximal map for G, except in the special case that $K^T K$ is a multiple of the identity matrix. To resolve this, we can add additional curvature to the x update step:

$$x_t = \arg \min_x \left\{ L(x, u_{t-1}, \Delta_{t-1}) + \frac{1}{2} (x - x_{t-1})^T (\lambda \mathbf{I} - \rho K^T K) (x - x_{t-1}) \right\},$$

where $\lambda \geq \rho \|K\|^2$. In this setting, the x update step now becomes solvable with the proximal map for G. In fact, this preconditioned form of the ADMM algorithm is equivalent to the CP algorithm (3); for details of the equivalence, see Chambolle and Pock (2011).

In addition to the ADMM and CP algorithms, and their variants, we mention one other option here. If F is differentiable, then proximal gradient descent offers a simple procedure, alternating between taking a gradient descent step on the term $F(Kx)$ and a proximal operator step on G. As we discussed in Section 1.1.1, the proximal gradient descent algorithm can be viewed as a simple special case of MOCCA (in that section, the terms were arranged slightly differently, with both differentiable and convex terms all included in G, but the scenarios are equivalent). Of course, this type of method cannot handle scenarios with a non-differentiable F, which can arise through total variation penalties and in other settings.

3.2 The Issue of Nonconvexity

Next, we turn to the setting where F and/or G may be nonconvex.

To begin, we consider the following interesting scenario, introduced in Section 1: suppose that G is itself convex, with trivial approximations $G_2 = G$, and that $x \rightarrow F(Kx)$ is strictly convex even though F is nonconvex. Since $x \rightarrow F(Kx) + G(x)$ is therefore strictly convex, the optimization problem has a unique global minimizer $x^* \in \mathbb{R}^d$. However, since F itself is nonconvex, then the strategies for optimization described in Section 3.1 may not be directly applicable for the task of finding x^* .

Here we outline the difficulties faced by the main existing approaches outlined in Section 3.1 for settings where F and/or G are nonconvex (including the scenario outlined above), and summarize the most relevant results in the literature.

To see this, first consider the ADMM algorithm. Suppose that F is nonconvex, and in particular, for some vector $w \in \mathbb{R}^m$, the function $s \mapsto F(s \cdot w)$ is strongly concave—specifically,

$$F(s \cdot w) \leq C - c \cdot s^2 \tag{21}$$

for some $C < \infty, c > 0$ and all $s > 0$. Then we see that the update step

$$u_t = \arg \min_u \{L(x_t, u, \Delta_{t-1})\}$$

is not well-defined: the function $u \rightarrow L(x_t, u, \Delta_{t-1})$ diverges to $-\infty$ when we set $u = s \cdot w$ and let $s \rightarrow \infty$. Therefore, for some types of nonconvex functions, the ADMM algorithm will not be implementable due to this divergence.

In the literature, theoretical guarantees for the performance of ADMM on nonconvex objective functions have been considered under several different settings. Broadly speaking, we can summarize existing work as falling into one of two categories. First, there are settings where the original form of the ADMM updates perform well. For instance, Magidson et al. (2015) proves convergence results (for a slightly different algorithm) when optimization is over a bounded set, thus avoiding the issue of divergence arising from directions of strong concavity in F as mentioned above; this paper also proves results guaranteeing optimality of any limit point, if one exists, in the unbounded optimization setting, but does not guarantee that a limit point is reached. In Li and Pong (2015), convergence results are proved assuming that one of the two terms (i.e. F or G) is smooth; in contrast, for the applications considered here, including total variation type penalties incorporated into F or hard constraints in G, it is critical to allow for nondifferentiable F and G. The special case of applying ADMM to nonconvex consensus problems is considered by Hong et al. (2016), with convergence guarantees again in a bounded setting, in this case assuming that any nonconvex functions must obey a lower bound.

Second, when the original ADMM updates cannot be expected to perform well—if for instance F has directions of strong concavity—then the ADMM can be modified by adding curvature to each update via a Bregman divergence term, as studied by Wang et al. (2014a) (with extensions to multi-block ADMM, with more than two terms in the objective function (Wang et al., 2015a)). This work proves convergence guarantees for the algorithm, but requires that the function F (after converting to our notation) is differentiable and smooth, in contrast to our work where allowing for nondifferentiable F is critical.

Next, consider the CP algorithm. When F is nonconvex, it is no longer the case in general that $F(Kx) = \max_w \{ \langle Kx, w \rangle - F^*(w) \}$. In fact, by definition of conjugate functions, this maximum defines the “conjugate of the conjugate”, i.e.

$$F^{**}(Kx) = \max_w \{ \langle Kx, w \rangle - F^*(w) \}.$$

It is known that any conjugate function must be convex, i.e. F^{**} is convex. This implies that $F^{**} \neq F$. Therefore, the saddle-point problem does not correspond to the original optimization problem: we have

$$\min_x \max_w \{ \langle Kx, w \rangle - F^*(w) + G(x) \} = \min_x \{ F^{**}(Kx) + G(x) \},$$

which is different from the original optimization problem since $F^{**} \neq F$. If $F^*(w)$ and $F^{**}(w)$ take finite values on some domain, then the CP algorithm can be expected to converge, but it will converge to the solution of an optimization problem that is different from the one intended due to the issue that $F^{**} \neq F$. Problems also arise in a setting where F exhibits negative curvature as in (21), in which case $F^*(w) = \infty$ for all w , so we do not have a well-defined saddle point problem to begin with.

To our knowledge, no general results exist for the CP algorithm with nonconvex and nondifferentiable F and/or G . Valkonen (2014) considers an interesting related problem, namely, a variant of the CP algorithm, where the objective function is now $F(\mathcal{K}(x)) + G(x)$, for convex F, G but with a nonlinear map $\mathcal{K}(x)$ in place of the previous linear map Kx , as the argument to F . In this case, convergence to a stationary point is proved, even when the nonlinearity of $\mathcal{K}(x)$ may make the overall problem nonconvex. Relatedly, Ochs et al. (2015) study the setting where $F(\mathcal{K}(x))$ is a nonconvex elementwise penalty on the convex transform $\mathcal{K}(x)$ while G is convex; their approach uses the CP algorithm as a subroutine for solving convex approximations of the objective function, at each step.

Next, we consider the option of proximal gradient descent, in the case that G has a simple proximal operator. Loh and Wainwright (2013) study penalized likelihood problems,

$$\min_x \{ \mathcal{L}(x) + \text{Penalty}(x) \},$$

where the likelihood term $\mathcal{L}(x)$ and/or the penalty term $\text{Penalty}(x)$ may exhibit nonconvexity. In many settings that arise in high-dimensional statistics, for instance, the likelihood term $\mathcal{L}(x)$ may be strongly concave in some directions, but will be strongly convex in all “plausible” directions, that is, all directions x that are not prohibited by the penalty term $\text{Penalty}(x)$. For instance, if $\text{Penalty}(x)$ is a sparsity-promoting penalty, with low values only at solutions x with many (near-zero) values, then $\mathcal{L}(x)$ might be strongly convex in all sparse directions. This relates to the notion of *restricted strong concavity*, introduced by Negalban et al. (2009), which we discuss in greater detail in Section 4.2.1. Under restricted convexity and smoothness assumptions on the likelihood term, and with bounds on the amount of nonconvexity allowed in the penalty term, Loh and Wainwright (2013) prove convergence to a point that is near the global optimum, for a proximal gradient descent method, with some additional details restricting steps to a bounded set to avoid diverging towards directions of strong concavity. Ochs et al. (2014)’s iPiano method gives an inertial (i.e. accelerated) proximal gradient descent method for this same setting where the loss is

differentiable with the penalty has an easy proximal map. The (accelerated) proximal gradient descent method for nonconvex problems is studied also by Ghadimi and Lan (2016); Li and Lin (2015). Note that these algorithms are applicable only when the terms in the objective function are all either differentiable (the likelihood) or have an easy-to-compute proximal operator (the penalty), and therefore, cannot be applied to many of the problems that that we have considered.

Finally, Bolte et al. (2014) propose an algorithm, Proximal Alternating Linearized Minimization (PALM), to solve a related problem of the form

$$\min_{x,w} \{ F(w) + G(x) + H(w, x) \}, \quad (22)$$

where H is differentiable while F, G each have easy to compute proximal maps (and may be nonconvex); this formulation is related to the problem we study, but PALM cannot be used to solve general problems of the form $F(Kx) + G(x)$ where F is nondifferentiable (since, if we add a variable $w = Kx$, the constraint $w = Kx$ cannot be enforced with any differentiable function $H(w, x)$ unless we allow modifications such as a relaxation to a penalty on $\|w - Kx\|_2^2$), and therefore again cannot be applied to some of the problems considered here.

3.3 Connection Between MOCCA and ADMM

In the convex setting, the CP method with parameter $\theta = 1$ is known to be equivalent to a preconditioned ADMM algorithm (Chambolle and Pock, 2011). Specifically, reformulating the original optimization problem in the ADMM form

$$\min_{x,u} \{ F(u) + G(x) : u = Kx \} = \min_{x,u} \max_{\Delta} \left\{ F(u) + G(x) + \langle \Delta, Kx - u \rangle + \frac{1}{2} \|Kx - u\|_2^2 \right\},$$

the CP iterations given in (3) are equivalent to the following preconditioned ADMM iterations, where we choose a preconditioning matrix $T^{-1} = K^T \Sigma K \succeq 0$:

$$\begin{cases} x_{t+1} = \arg \min_x \left\{ G(x) + \langle K^T \Delta_t, x \rangle + \frac{1}{2} \|Kx - u_t\|_2^2 + \frac{1}{2} \|x - x_t\|_{T^{-1} - K^T \Sigma K}^2 \right\}, \\ \Delta_{t+1} = \Delta_t + \Sigma(Kx_{t+1} - u_t), \\ u_{t+1} = \arg \min_u \left\{ F(u) - \langle \Delta_{t+1}, u \rangle + \frac{1}{2} \|Kx_{t+1} - u\|_2^2 \right\}, \end{cases}$$

(Here we use a slightly nonstandard indexing, writing the variable updates in the order x, Δ, u for later convenience.) We do not derive the equivalence here (see Chambolle and Pock, 2011, for details), but remark that the variables (x_t, u_t) at iteration t of the CP algorithm (3) can be recovered by taking $(x_t, \Sigma(Kx_t - u_t) + \Delta_t)$ from the ADMM iterations.

Similarly, in the more general nonconvex setting considered here, we can equivalently formulate the MOCCA method as a combination of the preconditioned ADMM iterations and taking convex expansions to F and G . The steps are given in Algorithm 3. This algorithm is exactly equivalent to the basic version of MOCCA, Algorithm 1, with extrapolation parameter $\theta = 1$, but is expressed as an ADMM type algorithm with preconditioning and with convex approximations to F and G . (The stable “inner loop” version of MOCCA, Algorithm 2, can also be interpreted as an extension of ADMM, but we do not give details here). The

equivalence between Algorithms 1 and 3 is simply an extension of the connection between the Chambolle-Pock algorithm and a preconditioned ADMM, as shown in Chambolle and Pock (2011).

Algorithm 3 MOCCA algorithm: ADMM version

Input: Convex functions $F_{\text{cov}}, G_{\text{cov}}$, differentiable functions $F_{\text{diff}}, G_{\text{diff}}$, linear operator K , positive diagonal step size matrices Σ, Γ .

Initialize: Primal variables $x_0 \in \mathbb{R}^d$, $u_0 \in \mathbb{R}^m$, dual variable $\Delta_0 \in \mathbb{R}^m$.

for $t = 0, 1, 2, \dots$ **do**

Update all variables:

$$\begin{cases} x_{t+1} = \arg \min_x \left\{ G_{\text{cov}}(x) + \langle x, K^\top \Delta_t \rangle + \frac{1}{2} \|Kx - u_t\|_{\Sigma}^2 + \frac{1}{2} \|x - x_t\|_{\Gamma^{-1} - K^\top \Sigma K}^2 \right\}, \\ \Delta_{t+1} = \Delta_t + \Sigma(Kx_{t+1} - u_t), \\ u_{t+1} = \arg \min_u \left\{ F_{\text{cov}}(u) - \langle \Delta_{t+1}, u \rangle + \frac{1}{2} \|Kx_{t+1} - u\|_{\Sigma}^2 \right\}, \end{cases}$$

until some convergence criterion is reached.

This ADMM formulation of MOCCA gives us a clearer understanding of the choice of the expansion points (z, v) in the MOCCA algorithm. Recalling the simpler form of the MOCCA algorithm given in Algorithm 1 where the expansion points are updated at each iteration (i.e. there is no “inner loop”), the expansion points were defined as

$$z_{t+1} = x_{t+1} \text{ and } v_{t+1} = \Sigma^{-1}(u_t - u_{t+1}) + K\bar{x}_{t+1}.$$

Using our conversion between the CP variables (x, w) and the ADMM variables (x, Δ, u) , we see that

$$\begin{aligned} v_{t+1} &= \Sigma^{-1}(u_t - u_{t+1}) + K(2x_{t+1} - x_t) \quad \text{since } \theta = 1 \text{ so } \bar{x}_{t+1} = 2x_{t+1} - x_t \\ &= \Sigma^{-1}(\Sigma(Kx_t - u_t) + \Delta_t - \Sigma(Kx_{t+1} - u_{t+1}) - \Delta_{t+1}) + K(2x_{t+1} - x_t) \\ &= \Sigma^{-1}(\Sigma(Kx_t - u_t) + \Delta_t - \Sigma(Kx_{t+1} - u_{t+1})) \\ &\quad - (\Delta_t + \Sigma(Kx_{t+1} - u_t)) + K(2x_{t+1} - x_t) \\ &= u_{t+1}, \end{aligned}$$

where the next-to-last step uses the definition of the update step for Δ_{t+1} . In other words, after the t th step, our estimated minimizers for $\{F(u) + G(x) : Kx = u\}$ are given by u_{t+1} and x_{t+1} , and our convex approximations to F and to G for the next step are consequently taken at the values $v = u_{t+1}$ and $z = x_{t+1}$.

4. Theoretical Results

In this section we present our two main theoretical results.

4.1 Convergence to a Critical Point

First, we show that if the algorithm converges, then its limit point is a solution to our original problem.

Theorem 1 Assume that the families of approximations F_v and G_z satisfy (4), and furthermore that

$$\begin{cases} (v, w) \mapsto \nabla(F - F_v)(w) \text{ is continuous jointly in } (v, w), \text{ and} \\ (z, x) \mapsto \nabla(G - G_z)(x) \text{ is continuous jointly in } (z, x). \end{cases} \quad (23)$$

Suppose that Algorithm 2 converges to a point, with

$$\begin{aligned} x_{t;\ell} &\rightarrow \hat{x} \quad \text{where the sequence } (x_{t;\ell}) \text{ is interpreted as } (x_{1;0}, x_{1;1}, \dots, x_{1;L_1}, x_{2;0}, \dots), \\ u_{t;\ell} &\rightarrow \hat{w} \quad \text{where } (u_{t;\ell}) \text{ is interpreted analogously,} \\ z_t &\rightarrow \hat{z}, \\ v_t &\rightarrow \hat{v}. \end{aligned} \quad (23)$$

Then \hat{x} is a critical point of the original optimization problem, in the sense that

$$0 \in K^\top \partial F_{K\hat{x}}(K\hat{x}) + \partial G_{\hat{x}}(\hat{x}).$$

4.2 Guarantees of Convergence

We now turn to theoretical results proving that the algorithm converges (and proving rates of convergence) under specific assumptions on F and G . In this section, we only consider the “inner loop” form of MOCCA, given in Algorithm 2. We show that if our inner loop length (i.e. L_t) tends to infinity, then we can bound the error of the algorithm.

We begin with an assumption on the step size parameters:

Assumption 1 The extrapolation parameter is set at $\theta = 1$, and the diagonal matrices Σ and Γ are chosen such that

$$M = \begin{pmatrix} \Gamma^{-1} & -K^\top \\ -K & \Sigma^{-1} \end{pmatrix} \succ 0.$$

Pock and Chambolle (2011) introduce this assumption for the (convex) preconditioned Chambolle-Pock algorithm, and give a simple construction for one choice of Σ, Γ to satisfy this without calculating any matrix norms or other high-cost operations, specified in (8) above.

Next, we turn to the convexity and smoothness assumptions required for our convergence guarantee.

4.2.1 RESTRICTED CONVEXITY AND SMOOTHNESS

In practice, F and/or G may each consist of multiple terms, combining characteristics of the problem such as a likelihood calculation or a penalty or constraint on the underlying signal. To accommodate a range of potential applications, in particular those arising in the regression and imaging applications described in Section 2, we consider a broad setting where our main assumptions involve the interplay between convexity and negative curvature in the functions F, G .

The notion of restricted strong convexity (RSC), introduced by Negahban et al. (2009), has often been used in high-dimensional statistics to express the idea that likelihood functions and optimization problems, which may not have desirable strong convexity properties

globally, nonetheless exhibit strong convexity in “directions of interest”. For example, in a least-squares regression problem with design matrix $A \in \mathbb{R}^{n \times d}$, with $n \ll d$, the least squares loss function $\mathcal{L}(x) = \frac{1}{2} \|y - Ax\|_2^2$ is not strongly convex since $A^\top A$ is rank deficient, but can yield good statistical properties if $A^\top A$ is strongly convex in all sparse directions, that is, $x^\top A^\top A x \geq c \cdot \|x\|_2^2$ for all sparse (or approximately sparse) vectors x . In this case, the loss function is globally convex, but it is the RSC property that ensures high accuracy for sparse regression problems. More recently, Loh and Wainwright (2013) proved that the RSC property, along with an analogous restricted smoothness property, can in fact be leveraged even in nonconvex optimization problems, such as the regression-with-errors-in-variables scenario described in Section 2.1. Their work relies on optimizing the variable x within some bounded set, to ensure that the RSC property will push x towards a good (local) minimum rather than allowing x to diverge. For instance, if the loss function has some directions of strong concavity—as is the case for regression-with-errors-in-variables in (14)—then staying within a bounded set is critical. In theory, their work focuses on problems that take the form of minimizing a penalized loss function over a bounded set $\{x : \|x\|_1 \leq R\}$, where we think of R as a large bound, requiring only a loose bound on the ℓ_1 norm of the true signal. In practice, if an optimization algorithm is initialized at zero, then it is often the case that the iterations will never leave a bounded region, without imposing any explicit constraint.

In general, results using the RSC condition take the following form: first, the loss function or objective function $\mathcal{L}(x)$ is shown to satisfy a RSC property of the form

$$\langle x - x', \nabla \mathcal{L}(x) - \nabla \mathcal{L}(x') \rangle \geq c \|x - x'\|_2^2 - \tau^2 \|x - x'\|_{\text{restrict}}^2,$$

for some structured norm $\|\cdot\|_{\text{restrict}}$ (for example, the ℓ_1 norm). Here $c > 0$ is a constant while τ is vanishingly small, for instance $c \sim 1$ and $\tau \sim \sqrt{\frac{\log(d)}{n}}$ in many high dimensional regression applications with sample size n . The solution \hat{x} is then shown to converge to the true signal x^* up to an error of size τR , where R is some bound on the signal complexity, for instance $R \sim \sqrt{k}$ where k is the true sparsity level of a sparse regression problem. In these settings, it is assumed that $\tau R = o(1)$, and that errors of this magnitude are negligible. We will follow this general framework in our convergence guarantee as well. However, since we consider settings where the signals may not have natural sparsity but would instead have a different type of structure (such as total variation sparsity), we replace the ℓ_1 norm with a general measure of signal complexity, $\|\cdot\|_{\text{restrict}}$ chosen with respect to the problem at hand (for instance, a total variation norm).

We now specify our assumptions on convexity and smoothness for the functions involved in the optimization, using the restricted strong convexity / restricted smoothness framework from the literature. Roughly speaking, the following assumption requires that the errors of the convex approximations $F - F_v$, $G - G_z$ are counterbalanced by strong convexity in the composite approximations $F_v(Kx) + G_z(x)$. For the term G , we allow for some flexibility by considering restricted strong convexity and restricted smoothness, relative to the structured norm $\|x\|_{\text{restrict}}$.

Assumption 2 *The approximations F_v and G_z satisfy the conditions (4), with additional assumptions as follows. For the function F and its family of local approximations F_v , we*

assume that F_v is strongly convex, while $F - F_v$ is smooth: for all u, v, w ,

$$\begin{cases} \text{Strong convexity of } F_v: \langle u, \partial F_v(w + u) - \partial F_v(w) \rangle \geq \|u\|_{\Lambda_F}^2, \\ \text{Smoothness of } F - F_v: |\langle u, \nabla(F - F_v)(v + w) \rangle| \leq \frac{1}{2} (\|u\|_{\Theta_F}^2 + \|w\|_{\Theta_F}^2), \end{cases}$$

for some $\Lambda_F, \Theta_F \succeq 0$.⁴ We also assume a gradient condition on F_v ,

$$\left\{ F_v \text{ satisfies a gradient condition: } \|\partial F_v(w) - \partial F_v(w')\|_2 \leq C_{\text{Lip}} + C_{\text{grad}} \|w - w'\|_2, \right.$$

for some $C_{\text{Lip}}, C_{\text{grad}} < \infty$. (For example, this is satisfied if F_v can be written as the sum of a Lipschitz function and a smooth function.)

For the function G and its family of local approximations G_z , we assume that G_z satisfies restricted strong convexity, while $G - G_z$ satisfies a restricted smoothness assumption: for all x, y, z ,

$$\begin{cases} \text{Restricted strong convexity of } G_z: \langle y, \partial G_z(x + y) - \partial G_z(x) \rangle \geq \|y\|_{\Lambda_G}^2 - \tau^2 \|y\|_{\text{restrict}}^2, \\ \text{Restricted smoothness of } G - G_z: |\langle y, \nabla(G - G_z)(z + x) \rangle| \leq \frac{1}{2} (\|x\|_{\Theta_G}^2 + \|y\|_{\Theta_G}^2) + \frac{\tau^2}{2} (\|x\|_{\text{restrict}}^2 + \|y\|_{\text{restrict}}^2), \end{cases}$$

for some $\Lambda_G, \Theta_G \succeq 0$ and $\tau < \infty$.

Finally, the total convexity in the local approximations F_v and G_z must (approximately) outweigh the total curvature of the differences $F - F_v$ and $G - G_z$. Specifically, for all $x \in \mathbb{R}^d$, we require

$$x^\top (K^\top \Lambda_F K + \Lambda_G) x \geq x^\top (K^\top \Theta_F K + \Theta_G) x + C_{\text{cvx}} \|x\|_2^2 - \tau^2 \|x\|_{\text{restrict}}^2,$$

for some $C_{\text{cvx}} > 0$ and $\tau < \infty$.

In general, greater convexity (i.e. Λ_F, Λ_G as strongly positive definite as possible) and tighter bounds on smoothness (i.e. Θ_F, Θ_G as small as possible) allow for a better (i.e. larger) constant C_{cvx} and, therefore, faster convergence of the algorithm. The value of τ is typically of a very small order in many problems arising in high-dimensional statistics, as discussed above for the sparse regression setting.

It is critical to note that this assumption does *not* require either F_v or G_z to be strictly convex—if the matrices Λ_F or Λ_G are not full rank, then strict convexity has not been assumed. Instead, F_v is strongly convex in any direction of \mathbb{R}^m contained in the column span of Λ_F , and similarly for G_z and Λ_G in \mathbb{R}^d . Our assumption essentially requires that the combination of these directions leads to overall (approximate) convexity, after accounting for concavity that might be introduced by the errors $F - F_v$ and $G - G_z$.

For simplicity in the statements and proofs of our results, we group the norms of all matrices from Assumptions 1 and 2 into a single constant:

$$C_{\text{matrix}} = \max \{ \|\Lambda_F\|, \|\Theta_F\|, \|\Lambda_G\|, \|\Theta_G\|, \|M\|, \|M^{-1}\| \}.$$

Throughout, we will treat $C_{\text{matrix}}, C_{\text{cvx}}, C_{\text{Lip}}$, and C_{grad} as fixed finite positive constants, and dependence on these values will not be given explicitly except in the proofs. On the other hand, the role of the restricted convexity/smoothness parameter τ will be shown explicitly.

3. We implicitly restrict all variables to the domain of the appropriate function, throughout.

4. For a function F with a multivalued subdifferential, this notation is taken to mean that the statement must hold for any elements of the subdifferentials, throughout the paper.

4.2.2 CONVERGENCE GUARANTEE

Choose any point $x^* \in \mathbb{R}^d$ with $\|x^*\|_{\text{restrict}} \leq R$, which is a critical point for the optimization problem

$$\min_{\|x\|_{\text{restrict}} \leq R} \{F(Kx) + G(x)\}.$$

For convenience, we will now absorb the constraint $\|x\|_{\text{restrict}} \leq R$ into the functions themselves, by replacing G with the function

$$x \mapsto \begin{cases} G(x), & \text{if } \|x\|_{\text{restrict}} \leq R, \\ +\infty, & \text{if } \|x\|_{\text{restrict}} > R. \end{cases}$$

and replacing G_z (for each z) with the function

$$x \mapsto \begin{cases} G_z(x), & \text{if } \|x\|_{\text{restrict}} \leq R, \\ +\infty, & \text{if } \|x\|_{\text{restrict}} > R. \end{cases}$$

In practice, as mentioned before, we typically do not need to explicitly incorporate this constraint into the optimization algorithm, as we will generally only see updates that all lie within a bounded region. However, in our statements and proofs of theoretical results from this point onward, we will assume that G, G_z restrict the domain of the variable x , that is,

$$G(x) = G_z(x) = +\infty \text{ whenever } \|x\|_{\text{restrict}} > R. \quad (24)$$

We will also assume that τR is bounded by a constant without further comment, since our results give convergence guarantees up to the accuracy level τR , the results are meaningful only if τR is small.

We now state our convergence guarantee for the stable form of the MOCCA algorithm, given in Algorithm 2:

Theorem 2 *Assume that Assumptions 1 and 2, and that G, G_z satisfy (24). Then there exists constants $C_{\text{converge}}, L_{\text{min}} < \infty$ and $\delta > 0$, such that if $\min_{t \geq 1} L_t \geq L_{\text{min}}$, then for all $t \geq 1$, the iterations of Algorithm 2 satisfy*

$$\|x_t - x^*\|_2 \leq C_{\text{converge}} \left(\frac{1}{\sqrt{L_t}} + \tau R \right)$$

where

$$L_t' = \min \{L_t, (1 + \delta)L_{t-1}, \dots, (1 + \delta)^{t-1}L_1\}.$$

As an example, we can set $L_t \sim (1 + \delta)^t$. Then after the t th inner loop, $\|x_t - x^*\|_2 \sim (1 + \delta)^{-t/2} + \tau R$, and the total number of iterations taken is $L_1 + \dots + L_t \sim (1 + \delta)^t$. In other words, the error $\|x - x^*\|_2$ scales as $\frac{1}{\sqrt{T}} + \tau R$ where T is the total number of update steps, i.e. error is inversely proportional to the square root of computational cost, up to the accuracy level τR .

We also remark that, if we were to assume additionally that F is differentiable and smooth, the result would improve dramatically: we would obtain error decaying exponentially in the number of update steps (up to the accuracy level τR). The resulting convergence

guarantee would then be comparable to the results obtained in Loh and Wainwright (2013, Theorem 3), which show a result with error at time t scaling as $c^t + \tau R$ (for a constant $c < 1$). However, convergence in this setting is of limited interest for the applications we have in mind, since total variation penalties, and many other natural penalties or losses falling into the F term of the composite objective functions, are not differentiable.

4.2.3 CONVEXITY AND SMOOTHNESS ASSUMPTIONS: AN EXAMPLE

To illustrate the many different matrices and constants appearing in Assumption 2 with a concrete example, we return to the problem studied in Section 2.1, where a least squares regression with errors in variables is combined with a total variation (or generalized ℓ_1) penalty. Recalling this setting, we seek to minimize $\mathcal{L}(x) + \nu \cdot \|Kx\|_1$, and we set $F_\nu(w) = F(w) = \nu \cdot \|w\|_1$, and

$$G(x) = \begin{cases} \mathcal{L}(x), & \text{if } \|x\|_1 \leq R, \\ +\infty, & \text{if } \|x\|_1 > R, \end{cases}$$

and

$$G_z(x) = \begin{cases} \mathcal{L}(x) + \frac{\sigma^2}{2} \|x - z\|_2^2, & \text{if } \|x\|_{\text{restrict}} \leq R, \\ +\infty, & \text{if } \|x\|_1 > R, \end{cases}$$

where

$$\mathcal{L}(x) = \frac{1}{2} x^T \left(\frac{Z^T Z}{n} - \sigma_A^2 \mathbf{I}_d \right) x - x^T \left(\frac{Z^T b}{n} \right).$$

Under the Gaussian noise model, the noisy design matrix given by entries $Z_{ij} = A_{ij} + \text{Normal}(0, \sigma_A^2)$ and the response is given by $b = A \cdot \text{true} + \text{Normal}(0, \sigma^2 \mathbf{I}_d)$. The MOCCA update steps for this problem are given in (16).

In this setting, Assumption 2 is satisfied with the following parameters. First, since F is convex but not strongly convex, we set $A_F = \mathbf{0}$; we can also set $\Theta_F = \mathbf{0}$ as $F_\nu = F$ for any expansion point v . F is ν -Lipschitz so we can take $C_{\text{Lip}} = 2\nu, C_{\text{grad}} = 0$. Next, for G , we see that

$$G_z(x) = \frac{1}{2} x^T \left(\frac{Z^T Z}{n} \right) x - x^T \left(\frac{Z^T b}{n} + \sigma_A^2 z \right)$$

(on the domain $\|x\|_{\text{restrict}} \leq R$), and so we can set $A_G = \frac{Z^T Z}{n}$. To check the smoothness condition, we have

$$\langle y, \nabla(G - G_z)(z + x) \rangle = \langle y, \sigma_A^2 x \rangle \leq \sigma_A^2 \cdot \frac{1}{2} (\|x\|_2^2 + \|y\|_2^2),$$

and so we can take $\Theta_G = \sigma_A^2 \mathbf{I}_d$.

Finally, for the ‘‘total convexity’’ condition of Assumption 2, we need to check that $K^T(A_F - \Theta_F)K + (A_G - \Theta_G) = \frac{Z^T Z}{n} - \sigma_A^2 \mathbf{I}_d$ satisfies restricted strong convexity. In Loh and Wainwright (2011, Corollary 1), it is shown that if the rows of the (original) design matrix A are drawn i.i.d. from a subgaussian distribution with covariance Σ_A then, assuming that the sample size n satisfies $n \gg \log(d)$, the matrix $\frac{Z^T Z}{n} - \sigma_A^2 \mathbf{I}_d$ (which is an unbiased estimate of the desired term $\frac{A^T A}{n}$ using the unknown original design matrix A) satisfies

$$x^T \left(\frac{Z^T Z}{n} - \sigma_A^2 \mathbf{I}_d \right) x \geq \frac{1}{2} \lambda_{\min}(\Sigma_A) \cdot \|x\|_2^2 - (\text{constant}) \cdot \frac{\log(d)}{n} \cdot \|x\|_{\text{restrict}}^2 \quad \text{for all } x \in \mathbb{R}^d \quad (25)$$

with high probability, when we choose $\|x\|_{\text{restrict}} = \|x\|_1$; similar results will hold for other structured choices of $\|\cdot\|_{\text{restrict}}$ such as total variation norm or a generalized ℓ_1 norm. Therefore we can set $C_{\text{cvx}} = \frac{1}{2}\lambda_{\min}(\Sigma_A)$ and $\tau \sim \sqrt{\frac{\log(d)}{n}}$ to obtain the desired condition in Assumption 2. Note that the guarantees of Theorem 2 give a meaningful convergence result even if we choose a fairly large radius R .

5. Experiments

We now implement the MOCCA algorithm to examine its performance in practice. Throughout this section, we work with the simpler formulation of MOCCA, given in Algorithm 1, with no “inner loop”. All computations were performed in MATLAB (MATLAB, 2015).⁵

For all simulations, we choose not to place a bound on $\|x\|_{\text{restrict}}$, although technically this is required by our convergence guarantees and those of the related results in Loh and Wainwright (2013) (which we compare to, in Simulation 2). Empirically we observe good convergence without imposing such a bound, but can easily add such a bound if desired.

We consider two examples: Simulation 1 studies nonconvex total variation regularization with a least squares loss (as described in Section 2.3), and Simulation 2 considers convex total variation regularization with a nonconvex loss arising from regression with errors in variables (as described in Section 2.1). While other algorithms which are developed specifically for these problems are available—for example, denoising with total variation penalties is studied by e.g. Chambolle and Darbon (2009); Wang et al. (2014b, 2015b), and could be combined with a proximal gradient method for Simulation 2—here our purpose is simply to illustrate applications of MOCCA to several concrete examples in order to demonstrate its flexibility for a broad range of problems. Specific problems will often have specialized algorithms which would far outperform our general-purpose method; however, slight modifications to the optimization problem (for example, replacing total variation regularization with a more general penalty $\|Kx\|_1$ for a generic dense matrix K , or with isotropic total variation) will often mean that specialized algorithms can no longer be applied, while MOCCA can adapt easily to accommodate these changes.

5.1 Simulation 1: Nonconvex Total Variation Penalty

In the first simulation, we study the nonconvex total variation penalty considered in Section 2.3, using a two-dimensional spatial structure. We generate data as follows: first, we define the true signal $x_{\text{true}} \in \mathbb{R}^d$ with dimension $d = 625$, obtained by vectorizing the two-dimensional locally constant array

$$\begin{pmatrix} \mathbf{1}_{5 \times 5} & \mathbf{0}_{5 \times 15} & \mathbf{0}_{5 \times 5} \\ \mathbf{0}_{15 \times 5} & \mathbf{1}_{15 \times 15} & \mathbf{0}_{15 \times 5} \\ \mathbf{0}_{5 \times 5} & \mathbf{0}_{5 \times 15} & \mathbf{1}_{5 \times 5} \end{pmatrix} \in \mathbb{R}^{25 \times 25}.$$

The two-dimensional total variation of the true signal is very low, because $\nabla_{2d} x_{\text{true}}$ is sparse. We then take a linear regression model with $n = 200$ observations, with design matrix

$A \in \mathbb{R}^{n \times d}$ with $A_{ij} \stackrel{\text{iid}}{\sim} \text{Normal}(0, 1)$ and $b \in \mathbb{R}^n$ with entries

$$b_i = (A \cdot x_{\text{true}})_i + \text{Normal}(0, 1).$$

We would then like to solve a penalized least-squares problem using the nonconvex total variation penalty introduced in Section 2.3, namely,

$$\hat{x} = \arg \min_x \{\text{Obj}(x)\} \text{ for } \text{Obj}(x) = \frac{1}{2} \|b - Ax\|_2^2 + \nu \cdot \log \text{TV}_\beta(x), \quad (26)$$

where we choose penalty parameter $\nu = 20$ and nonconvexity parameter $\beta = 3$ (recall that a low value of β corresponds to greater nonconvexity), and where the $\log \text{TV}_\beta(\cdot)$ penalty is defined with respect to two-dimensional total variation—recall

$$\log \text{TV}_\beta(x) = \log \mathbf{L}_{1,\beta}(\nabla_{2d} x) = \sum_i \beta \log(1 + |(\nabla_{2d} x)_i|/\beta).$$

Here $\nabla_{2d} \in \mathbb{R}^{m \times d}$ is the two-dimensional first differences matrix for the vectorized $d_1 \times d_2$ grid, where $d = d_1 \cdot d_2$ is the total dimension of the signal while $m = d_1(d_2 - 1) + d_2(d_1 - 1)$ is the number of first-order differences measured; in our case, we have $d_1 = d_2 = 25$, $d = 625$, and $m = 1200$.

Next, we implement the MOCCA algorithm with the two variants described in Section 2.3: setting $K = \nabla_{2d}$, we consider the more natural form where the penalty term is contained in F , given by

$$\begin{cases} F(w) = \nu \cdot \log \mathbf{L}_{1,\beta}(w), & \text{with } F_v(w) = \nu \cdot \|w\|_1 + \nu [h_\beta(w) + \langle w - v, \nabla h_\beta(v) \rangle], \\ G(x) = G_z(x) = \frac{1}{2} \|b - Ax\|_2^2, \end{cases} \quad (27)$$

where $h_\beta(w) = \log \mathbf{L}_{1,\beta}(w) - \|w\|_1$ is a differentiable concave function as discussed in Section 2.3. We also consider the less natural form where the penalty term is split across F and G , given by

$$\begin{cases} F(w) = F_v(w) = \nu \cdot \|w\|_1, \\ G(x) = \frac{1}{2} \|b - Ax\|_2^2 + \nu \cdot h_\beta(\nabla_{2d} x), \\ \quad \text{with } G_z(x) = \frac{1}{2} \|b - Ax\|_2^2 + \nu [h_\beta(\nabla_{2d} z) + \langle \nabla_{2d}(x - z), \nabla h_\beta(\nabla_{2d} z) \rangle], \end{cases} \quad (28)$$

We will refer to these two versions as MOCCA(natural) and MOCCA(split), respectively. Finally, we choose step size parameters

$$\Sigma = \lambda \cdot \frac{1}{2} \mathbf{I}_m \quad \text{and} \quad \mathbf{T} = \lambda^{-1} \cdot \frac{1}{4} \mathbf{I}_d,$$

which ensures that the positive semidefinite assumption, Assumption 1, will hold (although perhaps not strictly) as in Pock and Chambolle (2011). We test the algorithm across a range of λ values, $\lambda \in \{4, 8, 16, 32, 64\}$.

The results are shown in Figure 2, which plots the log value of the objective function $\text{Obj}(x)$ at each iteration, and also plots the log of the change in each iteration,

$$\text{Change}_t = \left\| \begin{pmatrix} x_{t-1} - x_t \\ w_{t-1} - w_t \end{pmatrix} \right\|_2. \quad (29)$$

⁵ Code for fully reproducing these simulations is available at <http://www.stat.uchicago.edu/~rina/mocca.html>.

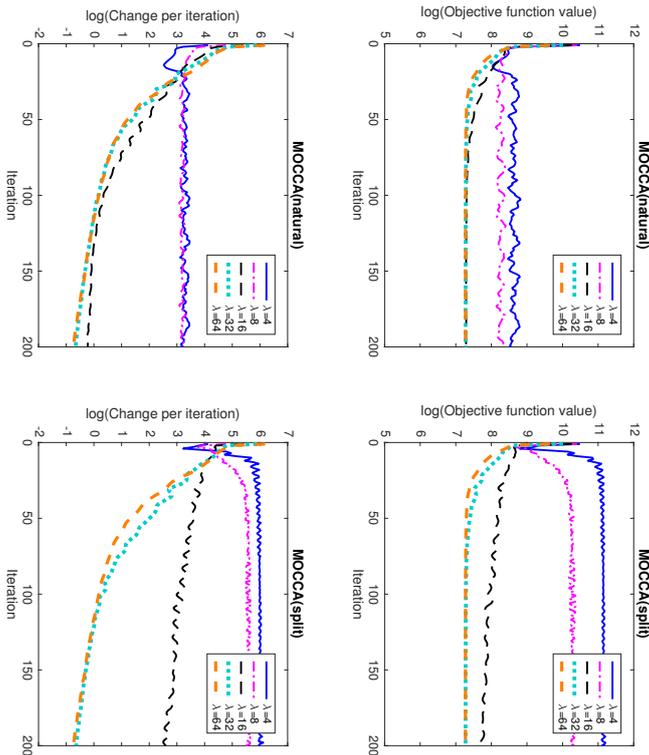


Figure 2: Results from Simulation 1. The top row plots the log of the objective function value defined in (26) against iteration number, while the bottom row plots the log of the change in the (x, u) variables (29) at each iteration, for several step size parameters λ . (Section 1.1.4 gives a direct correspondence between convergence of the (x, u) variables, and the optimality of the x variable.) The plots show results from the “natural” (left) and “split” (right) versions of the MOCCA algorithm, defined in (27) and (28), respectively.

(Recall from Section 1.1.4 that, if $\text{Change} \rightarrow 0$, then the optimality gap of the solution tends to zero; that is, the x variable is close to being a critical point for the optimization problem.) Looking first at the results for MOCCA(natural), we see that smaller λ values tend to lead to faster convergence at the very early stages, but poorer performance or instability at later stages. (In fact, this suggests the possibility of varying λ as we run more iterations, which we leave to future work.)

Turning to MOCCA(split), we see that the performance is worse at all λ values as compared with MOCCA(natural); the difference is minor for the largest λ values, but the lower λ values give far poorer results and far more instability for MOCCA(split) as compared to MOCCA(natural). This highlights the importance of the “mirroring” step in our algorithm,

which gives us the flexibility of placing the nonconvex terms into F , i.e. the function which will be optimized via the dual variable. In other scenarios, of course, a different arrangement of the terms may be preferable.

5.2 Simulation 2: Total Variation Penalty, with Errors in Variables

In the second simulation, we treat the errors-in-variables setting discussed in Section 2.1. We generate the signal x_{true} , the design matrix A , and the response vector b as in Simulation 1. Next, suppose our measurement of A is itself noisy: define $Z \in \mathbb{R}^{n \times d}$ with $Z_{ij} = A_{ij} + \text{Normal}(0, \sigma_A^2)$, where $\sigma_A = 0.2$. Finally, we would like to minimize the objective function

$$\frac{1}{2} x^T \left(Z^T Z - n \cdot \sigma_A^2 \mathbf{I}_d \right) x - x^T Z^T b + \nu \|x\|_{\text{TV}}, \quad (30)$$

with penalty parameter $\nu = 20$, where again we use two-dimensional total variation, $\|x\|_{\text{TV}} = \|\nabla_{2d} x\|_1$. (Here we use a different scaling of the likelihood term relative to Section 2.1 for simpler implementation and tuning.) Of course, due to the negative quadratic term, this objective function is strongly concave in some directions and so its global minimum is “at infinity”; within a bounded set, however, the penalty will ensure that the objective function is approximately convex. In practice, initializing our algorithm at $x = 0$ does not lead to any problems, and we converge to a bounded solution that can be viewed as a local minimum within a bounded set, e.g. $\{x : \|x\|_{\text{TV}} \leq R\}$ for some appropriate choice of R , as discussed in the context of restricted strong convexity in Section 4.2.1.

A proximal gradient descent method, as proposed by Loh and Wainwright (2013) for this type of nonconvex penalized likelihood, would in theory iterate the steps

$$\begin{cases} \tilde{x}_{t+1} = x_t - \frac{1}{\eta} \left((Z^T Z - n\sigma_A^2 \mathbf{I}_d) x_t - (Z^T b) \right), \\ x_{t+1} = \arg \min_x \left\{ \frac{1}{2} \|x - \tilde{x}_{t+1}\|_2^2 + \frac{\eta}{2} \|x\|_{\text{TV}} \right\}, \end{cases}$$

where $\frac{1}{\eta}$ is a step size parameter. However, the second step is a proximal operator for the total variation norm $\|x\|_{\text{TV}} = \|\nabla_{2d} x\|_1$, which cannot be calculated in closed form. Instead, we could apply the CP algorithm to the convex (sub)problem of this proximity operator with parameters $\Sigma = \lambda \cdot \frac{1}{2} \mathbf{I}_m$ and $\mathbf{T} = \lambda^{-1} \cdot \frac{1}{4} \mathbf{I}_d$, and could terminate this inner “prox loop” after some convergence criterion is reached, e.g. after some fixed number n_{step} of steps, or once the relative change in x is below ϵ_{mesh} . We do not show details of the derivation, but the complete procedure iterates these steps (taking extrapolation parameter $\theta = 1$):

$$\begin{cases} \text{Gradient step: } \tilde{x}_{t+1} = x_t - \frac{1}{\eta} \left((Z^T Z - n\sigma_A^2 \mathbf{I}_d) x_t - (Z^T b) \right), \\ \text{Initialize prox loop: } x_{t+1,0}^l = x_t, u_{t+1,0}^l = u_t, \\ \text{Run prox loop: for } \ell = 1, 2, \dots, \text{ writing } \tilde{x}_{t+1,\ell}^l = 2x_{t+1,\ell-1}^l - x_{t+1,\ell-1}^l, \\ \left\{ \begin{aligned} x_{t+1,\ell}^l &= (1 + \frac{1}{\lambda})^{-1} \left(x_{t+1,\ell-1}^l + \frac{1}{\lambda} \tilde{x}_{t+1} - \frac{1}{\lambda} \nabla_{2d}^T (u_{t+1,\ell-1}^l) \right), \\ u_{t+1,\ell}^l &= \text{Truncate}_{[-\nu/\eta, \nu/\eta]} \left(u_{t+1,\ell-1}^l + \frac{\nu}{2} \nabla_{2d} \tilde{x}_{t+1,\ell}^l \right), \end{aligned} \right. \\ \text{until a convergence criterion is reached} \\ \text{(i.e. } \ell = n_{\text{step}} \text{ OR } \left\| x_{t+1,\ell}^l - x_{t+1,\ell-1}^l \right\|_2 / \left\| x_{t+1,\ell-1}^l \right\|_2 \leq \epsilon_{\text{thresh}} \text{).} \end{cases} \quad (31)$$

Gather results from prox loop: $x_{t+1} = x_{t+1,n_{\text{step}}}^l$, $u_{t+1} = u_{t+1,n_{\text{step}}}^l$.

We will refer to this method as the Approximate Proximal Gradient Descent (APGD) algorithm, where “approximate” describes the fact that the proximal operator step is only ever solved approximately via a finite number of steps in the inner loop.

In fact, if we examine this algorithm carefully, we can find that by taking a single step of the inner “prox loop” (that is, setting $n_{\text{step}} = 1$), we arrive back at the steps of the MOCCA algorithm. Specifically, as in the implementation (15) in Section 2.1, we choose $K = \nabla_{2d}$, $F(w) = F_v(w) = \nu \|w\|_1$, and

$$G(x) = \frac{1}{2} x^\top (Z^\top Z - n\sigma_A^2 \mathbf{I}_d) x - x^\top (Z^\top b)$$

with local approximations given by linear expansions,

$$\begin{aligned} G_z(x) &= G(z) + \langle x - z, \nabla G(x) \rangle \\ &= \langle x, (Z^\top Z - n\sigma_A^2 \mathbf{I}_d) z - Z^\top b \rangle + (\text{terms constant with respect to } x). \end{aligned}$$

The update steps of the MOCCA algorithm are then given by

$$\begin{cases} x_{t+1} = \arg \min_x \left\{ \langle \nabla_{2d} x, w_t \rangle + G_z(x) + \frac{1}{2} \|x - x_t\|_{\mathbf{T}^{-1}}^2 \right\}, \\ w_{t+1} = \arg \min_y \left\{ -\langle \nabla_{2d} \bar{x}_{t+1}, w \rangle + F^*(w) + \frac{1}{2} \|w - w_t\|_{\Sigma^{-1}}^2 \right\}, \\ z_{t+1} = x_{t+1}, \end{cases}$$

which we can simplify to

$$\begin{cases} x_{t+1} = x_t - \mathbf{T} (\nabla_{2d}^\top w_t + (Z^\top Z - n\sigma_A^2 \mathbf{I}_d) x_t - (Z^\top b)), \\ w_{t+1} = \text{Truncate}_{[-\nu, \nu]} (w_t + \Sigma \nabla_{2d} \bar{x}_{t+1}). \end{cases}$$

If we choose

$$\Sigma = \frac{\lambda \eta}{2} \cdot \mathbf{I}_{d-1} \quad \text{and} \quad \mathbf{T} = \frac{1}{(4\lambda + 1)\eta} \cdot \mathbf{I}_d,$$

it can be shown that this is equivalent to the proximal gradient algorithm (31) with a single inner loop step, i.e. with $n_{\text{step}} = 1$ (specifically, the iterates x_t stay the same, while the other variables are related as $w_t = \eta \cdot u_t$).

Now we compare the performance of the approximate proximal gradient descent (APGD) algorithm, with various stopping criteria for the inner “prox loop”, against the performance of the MOCCA algorithm, which we can view as the APGD algorithm taking exactly one step in each inner “prox loop”. For simplicity, we consider only a few values for the step size parameters, setting $\eta = \lambda = 100$ or $\eta = \lambda = 200$. As for Simulation 1, we will see that higher values for these parameters gives more stability at the cost of slower convergence.

We consider stopping rules for the inner loop as follows: either we run the inner loop for a fixed number of steps, $n_{\text{step}} \in \{1, 5\}$ (with $n_{\text{step}} = 1$ yielding MOCCA), or we use a convergence criterion $\epsilon_{\text{fresh}} \in \{0.1, 0.05, 0.01\}$. Figure 3 shows the results; for the figure on the left, we see that running the inner loop longer does help to make our solutions more accurate (i.e. the objective function is lower) over the range of iterations. However, each iteration has greater computational cost when we increase the time spent running the inner loop. Since we would like to see the performance as a function of computational cost, the

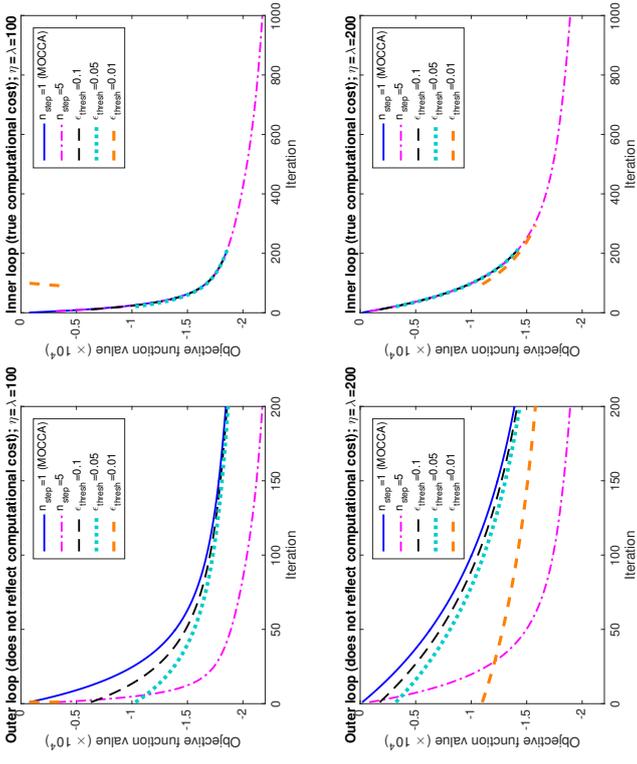


Figure 3: Results from Simulation 2, plotting the value of the objective function (30) against iteration number, counting either iterations of the “outer loop” (left) or of the “inner loop” (right), for various stopping rules for the inner loop in (31). Parameters are set as $\eta = \lambda = 100$ (top) or $\eta = \lambda = 200$ (bottom). Counting the number of passes through the inner loop is an accurate reflection of the true computational cost of (31), and so the right-hand plots give the correct interpretation of the results, where the various versions of the algorithms perform nearly identically in both settings (the lines are indistinguishable) except for the setting $\epsilon_{\text{fresh}} = 0.01$ which diverges for the setting $\eta = \lambda = 100$. Setting $n_{\text{step}} = 1$ yields the MOCCA algorithm as discussed in the text.

plot on the right-hand side of Figure 3 shows the same results plotted against the true number of iterations, i.e. where we count each pass through the inner loop of (31) rather than counting only passes through the outer loop of (31). In this setting, we see that in fact the various versions of the algorithms perform nearly identically—in other words, a one-step approximation to the proximal map performs just as well as a more conservative inner loop that is run for longer—with the exception of setting $\epsilon_{\text{fresh}} = 0.01$ and $\eta = \lambda = 100$, in which case the algorithm diverges immediately. It is interesting to note that this small

choice for ϵ_{fresh} is closest in spirit to proximal gradient descent (that is, the proximal step is the most accurate), although of course there may be some effect of tuning parameters. The choice ϵ_{fresh} does achieve convergence with the more conservative choice $\eta = \lambda = 200$, but convergence is noticeably slower in this case. Thus, we can conclude that when the proximal step does not have a closed form solution, it may be better to use a coarse approximation (which, implicitly, is the strategy taken by MOCCA for this problem) rather than aiming for near-convergence in the proximal step for each iteration.

6. Proofs

6.1 Critical Points (Theorem 1)

For this proof we will use two facts: for a continuous convex function h ,

$$\text{If } a_t \in \partial h(b_t) \text{ and } a_t \rightarrow a, b_t \rightarrow b \text{ then } a \in \partial h(b), \quad (32)$$

and

$$a \in \partial h(b) \text{ if and only if } b \in \partial h^*(a). \quad (33)$$

First, by definition of the w_{t+1} update step,

$$\partial \mathbf{F}_{v_t}^*(w_{t+1}) \ni \Sigma^{-1}(w_{t+1,0} - w_{t+1,1}) + K\bar{x}_{t+1,1},$$

and therefore by (33),

$$w_{t+1,1} \in \partial \mathbf{F}_{v_t}(\Sigma^{-1}(w_{t+1,0} - w_{t+1,1}) + K\bar{x}_{t+1,1}).$$

We can rewrite this as

$$w_{t+1,1} \in \underbrace{\partial \mathbf{F}_{\hat{v}}(\Sigma^{-1}(w_{t+1,0} - w_{t+1,1}) + K\bar{x}_{t+1,1})}_{(\text{Term 1})} + \underbrace{\nabla(\mathbf{F}_{v_t} - \mathbf{F}_{\hat{v}})(\Sigma^{-1}(w_{t+1,0} - w_{t+1,1}) + K\bar{x}_{t+1,1})}_{(\text{Term 2})}. \quad (34)$$

Next, since the solution converges, we see that

$$w_{t+1,1} \rightarrow \hat{w}, v_t \rightarrow \hat{v}, \Sigma^{-1}(w_{t+1,0} - w_{t+1,1}) + K\bar{x}_{t+1,1} \rightarrow K\hat{x}.$$

Our assumption (23) implies that

$$(v, w) \mapsto \nabla(\mathbf{F}_v - \mathbf{F}_{\hat{v}})(w) = -\nabla(\mathbf{F} - \mathbf{F}_{\hat{v}})(w) + \nabla(\mathbf{F} - \mathbf{F}_{\hat{v}})(w)$$

is jointly continuous in (v, w) , and so

$$(\text{Term 2}) = \nabla(\mathbf{F}_{v_t} - \mathbf{F}_{\hat{v}})(\Sigma^{-1}(w_{t+1,0} - w_{t+1,1}) + K\bar{x}_{t+1,1}) \rightarrow \nabla(\mathbf{F}_{\hat{v}} - \mathbf{F}_{\hat{v}})(K\hat{x}) = 0.$$

Therefore, applying the property (32) to the expression in (34), we see that

$$\hat{w} \in \partial \mathbf{F}_{\hat{v}}(K\hat{x}). \quad (35)$$

Next, for each t , by definition,

$$\begin{aligned} v_{t+1} &= \frac{1}{L_{t+1}} \sum_{\ell=1}^{L_{t+1}} (\Sigma^{-1}(w_{t+1,\ell-1} - w_{t+1,\ell}) + K\bar{x}_{t+1,\ell}) \\ &= \frac{1}{L_{t+1}} (\Sigma^{-1}(w_{t+1,0} - w_{t+1,L_{t+1}}) + K(x_{t+1,L_{t+1}} - x_{t+1,0})) + Kx_{t+1}. \end{aligned}$$

Taking limits on each side, we see that $\hat{v} = K\hat{x}$. Returning to (35) above this proves that

$$\hat{w} \in \partial \mathbf{F}_{K\hat{x}}(K\hat{x}).$$

Next, by definition of the x_{t+1} update step,

$$\partial \mathbf{G}_{z_t}(x_{t+1}) = \mathbf{T}^{-1}(x_{t+1,0} - x_{t+1,1}) - K^{\top} w_{t+1,0}.$$

Taking limits on each side as $t \rightarrow \infty$, and applying (32) as before,

$$\partial \mathbf{G}_{\hat{z}}(\hat{x}) \ni -K^{\top} \hat{w}.$$

And, we know that $z_{t+1} = x_{t+1}$ for each t , therefore $\hat{z} = \hat{x}$, and so

$$\partial \mathbf{G}_{\hat{z}}(\hat{x}) \ni -K^{\top} \hat{w}.$$

Combining the work above, then,

$$0 = K^{\top} \hat{w} - K^{\top} \hat{w} \in K^{\top} \partial \mathbf{F}_{K\hat{x}}(K\hat{x}) + \partial \mathbf{G}_{\hat{z}}(\hat{x}),$$

as desired.

6.2 Convergence Guarantee (Theorem 2)

We first introduce some notation and supporting lemmas before turning to the main proof.

6.2.1 NOTATION

Fixing any expansion points $(z, w) \in \mathbb{R}^d \times \mathbb{R}^m$, we define a primal-dual update step:

$$(x', w') = \text{Step}_{z,w}(x, w),$$

given by

$$\begin{cases} x' = \arg \min_{x''} \left\{ Kx'', w \right\} + G_z(x'') + \frac{1}{2} \|x'' - x\|_{\Sigma^{-1}}^2, \\ w' = \arg \min_{w''} \left\{ -(K(2w' - x), w'') + \mathbf{F}_{v'}^*(w'') + \frac{1}{2} \|w'' - w\|_{\Sigma^{-1}}^2 \right\}. \end{cases}$$

This is one step of the CP algorithm applied to the convex objective function

$$\min_x \{ \mathbf{F}_{v'}(Kx) + G_z(x) \}$$

with extrapolation parameter $\theta = 1$.

Next, defining x^* to be any critical point of the original problem as before, let $w^* \in \partial \mathbf{F}_{Kx^*}(Kx^*)$ be an element of the subdifferential such that

$$0 \in K^\top w^* + \partial \mathbf{G}_{x^*}(x^*).$$

Finally, for any expansion points (z, v) , define the (not necessarily unique) solution for the convex optimization problem when the we use approximations $\mathbf{F}_v, \mathbf{G}_z$:

$$x_{z,v}^* = \arg \min_x \{ \mathbf{F}_v(Kx) + \mathbf{G}_z(x) \},$$

and let $w_{z,v}^* \in \partial \mathbf{F}_v(Kx_{z,v}^*)$ be an element of the subdifferential such that

$$0 \in K^\top w^* + \partial \mathbf{G}_z(x_{z,v}^*).$$

6.2.2 LEMMAS

The proof of Theorem 2 can be split into several key results. First we state these lemmas and explain their role, then we will formally prove the theorem. The lemmas are proved in Appendix A.

The first lemma shows that, if we use expansion points (z, v) close to the true solution, i.e. $(z, v) \approx (x^*, Kx^*)$, then the minimizer $x_{z,v}^*$ for the convex approximation will be close to x^* .

Lemma 3 *Suppose that Assumptions 1 and 2 hold. Define*

$$\Theta = \begin{pmatrix} \Theta_G & 0 \\ 0 & \Theta_F \end{pmatrix} + \frac{C_{\text{cvx}}}{(C_{\text{matrix}})^2} \mathbf{I} \succ 0.$$

Then there exist constants $C_{\text{contr}} > 0, C_{\text{excess}} < \infty$, which depend only on $C_{\text{matrix}}, C_{\text{cvx}}, C_{\text{Lip}}, C_{\text{grad}}$, such that for any $(z, v) \in \text{dom}(\mathbf{G}) \times \text{dom}(\mathbf{F})$,

$$\left\| \begin{pmatrix} x_{z,v}^* - x^* \\ Kx_{z,v}^* - Kx^* \end{pmatrix} \right\|_{\Theta} \leq (1 - C_{\text{contr}}) \left\| \begin{pmatrix} z - x^* \\ v - Kx^* \end{pmatrix} \right\|_{\Theta} + C_{\text{excess}} \cdot \tau R$$

and

$$\left\| \begin{pmatrix} x_{z,v}^* - x^* \\ w_{z,v}^* - w^* \end{pmatrix} \right\|_2 \leq C_{\text{Lip}} + C_{\text{excess}} \left(\left\| \begin{pmatrix} z - x^* \\ v - Kx^* \end{pmatrix} \right\|_{\Theta} + \tau R \right).$$

The second lemma shows that, after running an ‘‘inner loop’’, the (x, w) variables are nearly optimal for the current convex approximation, and the next expansion points are also near this optimum.

Lemma 4 *Suppose that Assumptions 1 and 2 hold. For any $L \geq 1$, and any points $(x^{(0)}, w^{(0)}), (z, v) \in \text{dom}(\mathbf{G}) \times \text{dom}(\mathbf{F})$, suppose we iterate $\text{Step}_{z,v}(\cdot)$ for L times,*

$$(x^{(1)}, w^{(1)}) = \text{Step}_{z,v}(x^{(0)}, w^{(0)}), (x^{(2)}, w^{(2)}) = \text{Step}_{z,v}(x^{(1)}, w^{(1)}), \dots, \\ (x^{(L)}, w^{(L)}) = \text{Step}_{z,v}(x^{(L-1)}, w^{(L-1)}),$$

and then define the averages $(\tilde{x}, \tilde{w}) = \frac{1}{L} \sum_{\ell=1}^L (x^{(\ell)}, w^{(\ell)})$ and averaged expansion points (\tilde{z}, \tilde{v}) as

$$\tilde{z} = \frac{1}{L} \sum_{\ell=1}^L x^{(\ell)} \quad \text{and} \quad \tilde{v} = \frac{1}{L} \sum_{\ell=1}^L \left(\sum^{-1} (w^{(\ell-1)} - w^{(\ell)}) + K(2x^{(\ell)} - x^{(\ell-1)}) \right).$$

Then there exists a constant $C_{\text{iter}} < \infty$, depending only on $C_{\text{matrix}}, C_{\text{cvx}}, C_{\text{Lip}}, C_{\text{grad}}$ (and in particular, not dependent on L), such that

$$\left\| \begin{pmatrix} \tilde{z} - x_{z,v}^* \\ \tilde{v} - Kx_{z,v}^* \end{pmatrix} \right\|_{\Theta} \leq C_{\text{iter}} \left\| \begin{pmatrix} 1 \\ \sqrt{L} \end{pmatrix} \left\| \begin{pmatrix} x^{(0)} - x_{z,v}^* \\ w^{(0)} - w_{z,v}^* \end{pmatrix} \right\|_2 + \tau R \right).$$

and

$$\left\| \begin{pmatrix} \tilde{x} - x_{z,v}^* \\ \tilde{w} - w_{z,v}^* \end{pmatrix} \right\|_2 \leq C_{\text{Lip}} + C_{\text{iter}} \left(\frac{1}{\sqrt{L}} \left\| \begin{pmatrix} x^{(0)} - x_{z,v}^* \\ w^{(0)} - w_{z,v}^* \end{pmatrix} \right\|_2 + \tau R \right)$$

6.2.3 PROOF OF THEOREM 2

We will assume for simplicity that the expansion point is initialized with some z_0 satisfying $\|z_0\|_{\text{restrict}} \leq R$; if this is not the case at step $t = 0$, then our results can be easily adjusted since we will have $z_1 = x_1 \in \text{dom}(\mathbf{G}_{z_0})$ and therefore $\|z_1\|_{\text{restrict}} \leq R$, so we can simply shift our calculations by one time point.

First, choose any δ such that $0 < \delta < (1 - C_{\text{contr}})^{-2} - 1$. Define constants

$$C_1 = \max \left\{ \left\| \begin{pmatrix} x_1 - x_{z_1, v_1}^* \\ w_1 - w_{z_1, v_1}^* \end{pmatrix} \right\|_2, \right. \\ \left. 6C_{\text{Lip}} + 4C_{\text{excess}} + 2 \left(C_{\text{iter}} + 2C_{\text{excess}} \left(\frac{C_{\text{iter}} + C_{\text{excess}}}{C_{\text{contr}}} + 1 \right) \right) \tau R \right\}, \\ C_2 = \max \left\{ \sqrt{L_1} \cdot \left\| \begin{pmatrix} z_1 - x^* \\ v_1 - Kx^* \end{pmatrix} \right\|_{\Theta}, \frac{C_{\text{iter}} C_1}{1 - (1 - C_{\text{contr}}) \sqrt{1 + \delta}} \right\}, \\ C_3 = \frac{C_{\text{iter}} + C_{\text{excess}}}{C_{\text{contr}}},$$

and define

$$L_{\min} = \max\{4(C_{\text{iter}})^2, (C_2)^2\}.$$

To prove the desired result, we will prove that

$$\left\| \begin{pmatrix} x_{t+1} - x_{z_{t+1}, v_{t+1}}^* \\ w_{t+1} - w_{z_{t+1}, v_{t+1}}^* \end{pmatrix} \right\|_2 \leq C_1 \quad (36)$$

and

$$\left\| \begin{pmatrix} z_{t+1} - x^* \\ v_{t+1} - Kx^* \end{pmatrix} \right\|_{\Theta} \leq \frac{C_2}{\sqrt{L_{t+1}}} + C_3 \tau R \quad (37)$$

for all $t \geq 0$.

Assuming that these bounds hold, we then have

$$\|x_{t+1} - x^*\|_2 = \|z_{t+1} - x^*\|_2 \leq \|\Theta^{-1}\| \cdot \left(\frac{C_2}{\sqrt{L_{t+1}}} + C_3\tau R \right),$$

where the first step holds by definition of z_{t+1} ; this proves the desired theorem with $C_{\text{converge}} := \|\Theta^{-1}\| \cdot \max\{C_2, C_3\}$.

Now we prove (36) and (37) by induction. For $t = 0$, both statements are true trivially by our definitions of C_1 and C_2 . Now we will assume that the statements are true for all $t = 0, \dots, m-1$ and will prove that they hold for $t = m$. First, for (37), we have

$$\begin{aligned} & \left\| \begin{pmatrix} z_{m+1} - x^* \\ v_{m+1} - Kx^* \end{pmatrix} \right\|_{\Theta} \\ & \leq \left\| \begin{pmatrix} z_{m+1} - x_{z_m, v_m}^* \\ v_{m+1} - Kx_{z_m, v_m}^* \end{pmatrix} \right\|_{\Theta} + \left\| \begin{pmatrix} x_{z_m, v_m}^* - x^* \\ Kx_{z_m, v_m}^* - Kx^* \end{pmatrix} \right\|_{\Theta} \quad \text{by the triangle inequality} \\ & \leq \left\| \begin{pmatrix} z_{m+1} - x_{z_m, v_m}^* \\ v_{m+1} - Kx_{z_m, v_m}^* \end{pmatrix} \right\|_{\Theta} \\ & \quad + (1 - C_{\text{contr}}) \cdot \left\| \begin{pmatrix} z_m - x^* \\ v_m - Kx^* \end{pmatrix} \right\|_{\Theta} + C_{\text{excess}}\tau R \quad \text{by Lemma 3} \\ & \leq C_{\text{lier}} \left(\frac{1}{\sqrt{L_{m+1}}} \left\| \begin{pmatrix} x_m - x_{z_m, v_m}^* \\ v_m - u_{z_m, v_m}^* \end{pmatrix} \right\|_2 + \tau R \right) \\ & \quad + (1 - C_{\text{contr}}) \cdot \left\| \begin{pmatrix} z_m - x^* \\ v_m - Kx^* \end{pmatrix} \right\|_{\Theta} + C_{\text{excess}}\tau R \quad \text{by Lemma 4} \\ & \leq C_{\text{lier}} \left(\frac{1}{\sqrt{L_{m+1}}} C_1 + \tau R \right) + (1 - C_{\text{contr}}) \cdot \left(\frac{C_2}{\sqrt{L_m}} + C_3\tau R \right) \\ & \quad + C_{\text{excess}}\tau R \quad \text{by (36) and (37) applied with } t = m-1 \\ & \leq C_{\text{lier}} \left(\frac{1}{\sqrt{L_{m+1}}} C_1 + \tau R \right) + (1 - C_{\text{contr}}) \cdot \left(C_2\sqrt{1+\delta} + C_3\tau R \right) \\ & \quad + C_{\text{excess}}\tau R \quad \text{since } L'_{m+1} \leq L_{m+1}, (1+\delta)L'_m \\ & \leq \frac{C_2}{\sqrt{L'_{m+1}}} + C_3\tau R, \end{aligned}$$

where the last step holds by our definition of the constants C_2, C_3 . This concludes the proof of (37) for $t = m$.

Next, we turn to the proof of (36). For both $t = m-1$ and $t = m$,

$$\begin{aligned} & \left\| \begin{pmatrix} x_{z_{t+1}, v_{t+1}}^* - x^* \\ u_{z_{t+1}, v_{t+1}}^* - u^* \end{pmatrix} \right\|_2 \leq C_{\text{lip}} + C_{\text{excess}} \left(\left\| \begin{pmatrix} z_{t+1} - x^* \\ v_{t+1} - Kx^* \end{pmatrix} \right\|_{\Theta} + \tau R \right) \quad \text{by Lemma 3} \\ & = C_{\text{lip}} + C_{\text{excess}} \left(\frac{C_2}{\sqrt{L'_{t+1}}} + (C_3 + 1)\tau R \right) \quad \text{by (37) at step } t \\ & \leq C_{\text{lip}} + C_{\text{excess}} \left(\frac{C_2}{\sqrt{L_{\min}}} + (C_3 + 1)\tau R \right) \quad \text{since } L'_{t+1} \geq L_{\min}, \end{aligned}$$

and also, we have

$$\begin{aligned} & \left\| \begin{pmatrix} x_{m+1} - x_{z_m, v_m}^* \\ u_{m+1} - u_{z_m, v_m}^* \end{pmatrix} \right\|_2 \\ & \leq C_{\text{lip}} + C_{\text{lier}} \left(\frac{1}{\sqrt{L_{m+1}}} \left\| \begin{pmatrix} x_m - x_{z_m, v_m}^* \\ u_m - u_{z_m, v_m}^* \end{pmatrix} \right\|_2 + \tau R \right) \quad \text{by Lemma 4} \\ & \leq C_{\text{lip}} + C_{\text{lier}} \left(\frac{1}{\sqrt{L_{m+1}}} C_1 + \tau R \right) \quad \text{by (36) applied with } t = m-1 \\ & \leq C_{\text{lip}} + C_{\text{lier}} \left(\frac{1}{\sqrt{L_{\min}}} C_1 + \tau R \right) \quad \text{since } L_{m+1} \geq L_{\min}. \end{aligned}$$

Therefore, combining these calculations,

$$\begin{aligned} & \left\| \begin{pmatrix} x_{m+1} - x_{z_{m+1}, v_{m+1}}^* \\ u_{m+1} - u_{z_{m+1}, v_{m+1}}^* \end{pmatrix} \right\|_2 \\ & \leq \left\| \begin{pmatrix} x_{m+1} - x_{z_m, v_m}^* \\ u_{m+1} - u_{z_m, v_m}^* \end{pmatrix} \right\|_2 + \left\| \begin{pmatrix} x_{z_m, v_m}^* - x^* \\ u_{z_m, v_m}^* - u^* \end{pmatrix} \right\|_2 + \left\| \begin{pmatrix} x_{z_{m+1}, v_{m+1}}^* - x^* \\ u_{z_{m+1}, v_{m+1}}^* - u^* \end{pmatrix} \right\|_2 \\ & \leq 3C_{\text{lip}} + C_{\text{lier}} \left(\frac{1}{\sqrt{L_{\min}}} C_1 + \tau R \right) + 2C_{\text{excess}} \left(\frac{C_2}{\sqrt{L_{\min}}} + (C_3 + 1)\tau R \right) \\ & \leq 3C_{\text{lip}} + C_{\text{lier}} \left(\frac{1}{\sqrt{4(C_{\text{lier}})^2}} C_1 + \tau R \right) \\ & \quad + 2C_{\text{excess}} \left(\frac{C_2}{\sqrt{(C_2)^2}} + \left(\frac{C_{\text{lier}} + C_{\text{excess}}}{C_{\text{contr}}} + 1 \right) \tau R \right) \quad \text{by definition of } C_3 \text{ and of } L_{\min} \\ & \leq C_1, \end{aligned}$$

by definition of C_1 . This proves the desired bound (36) for $t = m$, and thus we have proved the theorem.

7. Discussion

We have developed a primal/dual algorithm for minimizing composite objective functions of the form $F(Kx) + G(x)$, which is able to handle nondifferentiability and nonconvexity (even strong concavity) in each individual term, beyond what is possible with many existing

approaches based on alternating minimization or proximal gradient methods. The key step of the MOCCA algorithm is the careful choice of local convex approximations to F and G at each step, which respects the mirroring between the primal and dual variables of the algorithm. Our method allows for accurate and efficient optimization for a range of problems arising in high-dimensional statistics, such as nonconvex total variation penalties (which reduce the bias caused by shrinkage, when compared to using a convex total variation norm), as well as inverse problems in computed tomography (CT) imaging.

Our present theoretical results give a convergence guarantee, in the case that the overall objective function is approximately convex, for a more stable form of the MOCCA algorithm. In future work, we hope to better understand the relative performance of the various forms of the algorithm, and to find a tighter characterization of the convergence behavior of the algorithm. It would also be interesting to consider a more general form of objective function, $F(w) + G(x)$ where F, G are nonconvex and nondifferentiable, and where instead of the linear constraint $w = Kx$, the variables w and x are linked via a nonlinear map; such an extension would greatly increase the range of applications of the method.

Acknowledgments

This work was partially supported by NIH grants CA158446, CA182264, and EB018102. The contents of this article are solely the responsibility of the authors and do not necessarily represent the official views of the National Institutes of Health. The authors are grateful to Woosok Ha for discussions on the connections with existing algorithms, and to collaborators Taly Gilat Schmidt and Xiaochuan Pan for discussions on the application to CT imaging.

Appendix A. Proofs of Lemmas

A.1 Proof of Lemma 3

By definition of $x_{z,v}^*$

$$0 \in K^\top \partial F_v(Kx_{z,v}^*) + \partial \mathcal{G}_z(x_{z,v}^*) \quad (38)$$

and since x^* is a critical point of the original objective function,

$$0 \in K^\top \partial F_{Kx^*}(Kx^*) + \partial \mathcal{G}_{x^*}(x^*) . \quad (39)$$

Since $(F_v - F_{Kx^*}) = (F - F_{Kx^*}) - (F - F_v)$ and $(G_z - G_{x^*}) = (G - G_{x^*}) - (G - G_z)$ are differentiable, we can rewrite (39) as

$$0 \in K^\top \partial F_v(Kx^*) + K^\top \nabla(F - F_v)(Kx^*) - K^\top \nabla(F - F_{Kx^*})(Kx^*) \\ + \partial \mathcal{G}_z(x^*) + \nabla(G - G_z)(x^*) - \nabla(G - G_{x^*})(x^*) .$$

By the first-order conditions (4), we know that $\nabla(F - F_{Kx^*})(Kx^*) = 0$ and $\nabla(G - G_{x^*})(x^*) = 0$, so this reduces to

$$0 \in K^\top \partial F_v(Kx^*) + K^\top \nabla(F - F_v)(Kx^*) + \partial \mathcal{G}_z(x^*) + \nabla(G - G_z)(x^*) . \quad (40)$$

We also see that $\|x^*\|_{\text{restrict}}, \|x_{z,v}^*\|_{\text{restrict}} \leq R$, since $x^*, x_{z,v}^*$ must lie in $\text{dom}(G) = \text{dom}(G_z)$.

Then we have

$$\begin{aligned} & \|x_{z,v}^* - x^*\|_{\Theta_G}^2 + \|Kx_{z,v}^* - Kx^*\|_{\Theta_F}^2 + C_{\text{cvx}} \|x_{z,v}^* - x^*\|_2^2 - 2\tau^2 \cdot \|x_{z,v}^* - x^*\|_{\text{restrict}}^2 \\ & \leq \|Kx_{z,v}^* - Kx^*\|_{\Theta_F}^2 + \|x_{z,v}^* - x^*\|_{\Lambda_G}^2 - \tau^2 \cdot \|x_{z,v}^* - x^*\|_{\text{restrict}}^2 \quad \text{by Assumption 2} \\ & \leq \langle Kx_{z,v}^* - Kx^*, \partial F_v(Kx_{z,v}^*) - \partial F_v(Kx^*) \rangle \\ & \quad + \langle x_{z,v}^* - x^*, \partial \mathcal{G}_z(x_{z,v}^*) - \partial \mathcal{G}_z(x^*) \rangle \quad \text{by Assumption 2} \\ & = \langle Kx_{z,v}^* - Kx^*, \nabla(F - F_v)(Kx^*) \rangle + \langle x_{z,v}^* - x^*, \nabla(G - G_z)(x^*) \rangle \quad \text{by (38) and (40)} \\ & \leq \frac{1}{2} \|Kx_{z,v}^* - Kx^*\|_{\Theta_F}^2 + \frac{1}{2} \|v - Kx^*\|_{\Theta_F}^2 + \frac{1}{2} \|x_{z,v}^* - x^*\|_{\Theta_G}^2 + \frac{1}{2} \|z - x^*\|_{\Theta_G}^2 \\ & \quad + \frac{\tau^2}{2} \left(\|x_{z,v}^* - x^*\|_{\text{restrict}}^2 + \|z - x^*\|_{\text{restrict}}^2 \right) \quad \text{by Assumption 2} . \end{aligned}$$

Next, recall that $\|x_{z,v}^*\|_{\text{restrict}}, \|x^*\|_{\text{restrict}} \leq R$ from before and $\|z\|_{\text{restrict}} \leq R$ by assumption. After rearranging terms and multiplying by 2, this gives

$$\begin{aligned} & \|x_{z,v}^* - x^*\|_{\Theta_G}^2 + \|Kx_{z,v}^* - Kx^*\|_{\Theta_F}^2 + 2C_{\text{cvx}} \|x_{z,v}^* - x^*\|_2^2 \\ & \leq \|z - x^*\|_{\Theta_G}^2 + \|v - Kx^*\|_{\Theta_F}^2 + \|v - Kx^*\|_{\Theta_F}^2 + 24\tau^2 R^2 . \quad (41) \end{aligned}$$

Now, using the definition of Θ , we see that

$$\begin{pmatrix} \Theta_G & 0 \\ 0 & \Theta_F \end{pmatrix} \preceq (1 - C_{\text{contr}})^2 \cdot \Theta$$

where

$$C_{\text{contr}} := \frac{C_{\text{cvx}}}{2 \left(1 + \frac{C_{\text{cvx}}}{(C_{\text{matrix}})^3} \right)} > 0 .$$

We then get

$$\begin{aligned} & (1 - C_{\text{contr}})^2 \left\| \begin{pmatrix} z - x^* \\ v - Kx^* \end{pmatrix} \right\|_{\Theta}^2 + 24\tau^2 R^2 \\ & \geq \|z - x^*\|_{\Theta_G}^2 + \|v - Kx^*\|_{\Theta_F}^2 + 24\tau^2 R^2 \\ & \geq \|x_{z,v}^* - x^*\|_{\Theta_G}^2 + \|Kx_{z,v}^* - Kx^*\|_{\Theta_F}^2 + 2C_{\text{cvx}} \|x_{z,v}^* - x^*\|_2^2 \quad \text{from (41)} \\ & \geq \|x_{z,v}^* - x^*\|_{\Theta_G}^2 + \|Kx_{z,v}^* - Kx^*\|_{\Theta_F}^2 \\ & \quad + C_{\text{cvx}} \|x_{z,v}^* - x^*\|_2^2 + \frac{C_{\text{cvx}}}{(C_{\text{matrix}})^2} \|Kx_{z,v}^* - Kx^*\|_2^2 \quad \text{since } \|K\| \leq C_{\text{matrix}} \\ & \geq \|x_{z,v}^* - x^*\|_{\Theta_G + C_{\text{cvx}}/(C_{\text{matrix}})^2}^2 + \|Kx_{z,v}^* - Kx^*\|_{\Theta_F + C_{\text{cvx}}/(C_{\text{matrix}})^2}^2 \quad \text{since } C_{\text{matrix}} \geq 1 \\ & = \left\| \begin{pmatrix} x_{z,v}^* - x^* \\ Kx_{z,v}^* - Kx^* \end{pmatrix} \right\|_{\Theta} . \end{aligned}$$

Using the fact that $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$, we obtain

$$\left\| \begin{pmatrix} x_{z,v}^* - x^* \\ Kx_{z,v}^* - Kx^* \end{pmatrix} \right\|_{\Theta} \leq (1 - C_{\text{contr}}) \left\| \begin{pmatrix} z - x^* \\ v - Kx^* \end{pmatrix} \right\|_{\Theta} + \sqrt{24\tau} R . \quad (42)$$

Next, by definition, we also have

$$u_{z,v}^* \in \partial F_v(Kx_{z,v}^*) \text{ and } w^* \in \partial F_{Kx^*}(Kx^*).$$

This second expression can be rewritten as

$$w^* \in \partial F_v(Kx^*) + \nabla(F - F_v)(Kx^*) - \nabla(F - F_{Kx^*})(Kx^*) = \partial F_v(Kx^*) + \nabla(F - F_v)(Kx^*),$$

where the last step uses the first-order condition (4) to see that $\nabla(F - F_{Kx^*})(Kx^*) = 0$. Therefore,

$$\begin{aligned} \|u_{z,v}^* - w^*\|_2 &\leq \|\partial F_v(Kx^*) - \partial F_v(Kx_{z,v}^*)\|_2 + \|\nabla(F - F_v)(Kx^*)\|_2 \\ &\leq C_{\text{Lip}} + C_{\text{grad}} \|Kx_{z,v}^* - Kx^*\|_2 + \sqrt{\|\Theta_F\|} \|v - Kx^*\|_{\Theta_F} \quad \text{by Assumption 2} \\ &\leq C_{\text{Lip}} + C_{\text{grad}} C_{\text{matrix}} \|x_{z,v}^* - x^*\|_2 + \sqrt{C_{\text{matrix}}} \|v - Kx^*\|_{\Theta_F}, \end{aligned}$$

and so

$$\begin{aligned} \left\| \begin{pmatrix} x_{z,v}^* - x^* \\ u_{z,v}^* - w^* \end{pmatrix} \right\|_2 &\leq C_{\text{Lip}} + (1 + C_{\text{grad}} C_{\text{matrix}}) \|x_{z,v}^* - x^*\|_2 + \sqrt{C_{\text{matrix}}} \|v - Kx^*\|_{\Theta_F} \\ &\leq C_{\text{Lip}} + \frac{1 + C_{\text{grad}} C_{\text{matrix}}}{\sqrt{C_{\text{cov}}}/C_{\text{matrix}}} \left\| \begin{pmatrix} x_{z,v}^* - x^* \\ Kx_{z,v}^* - Kx^* \end{pmatrix} \right\|_{\Theta} \\ &\quad + \sqrt{C_{\text{matrix}}} \|v - Kx^*\|_{\Theta_F} \quad \text{since } \Theta \supseteq \frac{C_{\text{cov}}}{(C_{\text{matrix}})^2} \mathbf{I} \\ &\leq C_{\text{Lip}} + \frac{1 + C_{\text{grad}} C_{\text{matrix}}}{\sqrt{C_{\text{cov}}}/C_{\text{matrix}}} \left((1 - C_{\text{contr}}) \left\| \begin{pmatrix} z - x^* \\ v - Kx^* \end{pmatrix} \right\|_{\Theta} + \sqrt{24}\tau R \right) \\ &\quad + \sqrt{C_{\text{matrix}}} \left\| \begin{pmatrix} z - x^* \\ v - Kx^* \end{pmatrix} \right\|_{\Theta}, \end{aligned}$$

where the last step applies (42) from above. Setting

$$C_{\text{cess}} = \max \left\{ \sqrt{24}, \sqrt{C_{\text{matrix}}} + (1 - C_{\text{contr}}) \cdot \frac{1 + C_{\text{grad}} C_{\text{matrix}}}{\sqrt{C_{\text{cov}}}/C_{\text{matrix}}}, \sqrt{24} \cdot \frac{1 + C_{\text{grad}} C_{\text{matrix}}}{\sqrt{C_{\text{cov}}}/C_{\text{matrix}}} \right\},$$

this is sufficient to prove the lemma.

A.2 Proof of Lemma 4

First we state and prove a supporting lemma which considers only a single step of the ‘‘inner loop’’.

Lemma 5 *There exists a constant $C_{\text{monotone}} > 0$, which depends only on C_{matrix} , C_{cov} , C_{Lip} , C_{grad} , such that, for any x, y, z with $\|x\|_{\text{restrict}} \leq R$, if $(x', w') = \text{Step}_{z,v}(x, w)$, then*

$$\begin{aligned} C_{\text{monotone}} \left(\|x' - x_{z,v}^*\|_2^2 + \left\| \begin{pmatrix} x - x' \\ w - w' \end{pmatrix} \right\|_2^2 \right) \\ \leq \left\| \begin{pmatrix} x - x_{z,v}^* \\ w - w_{z,v}^* \end{pmatrix} \right\|_M^2 - \left\| \begin{pmatrix} x' - x_{z,v}^* \\ w' - w_{z,v}^* \end{pmatrix} \right\|_M^2 + \tau^2 R^2. \end{aligned}$$

Proof By definition of the (x, w) update step, we have

$$\mathbb{T}^{-1}(x - x') - K^{\text{T}} w \in \partial \mathcal{G}_z(x') \quad (43)$$

and

$$\Sigma^{-1}(w - w') + K(2x' - x) \in \partial F_v^*(w'),$$

and from (33) we know that this last expression implies

$$w' \in \partial F_v(\Sigma^{-1}(w - w') + K(2x' - x)). \quad (44)$$

Similarly, by definition of $(x_{z,v}^*, w_{z,v}^*)$, we also have

$$-K^{\text{T}} w_{z,v}^* \in \partial \mathcal{G}_z(x_{z,v}^*) \quad (45)$$

and

$$u_{z,v}^* \in \partial F_v(Kx_{z,v}^*). \quad (46)$$

Therefore, combining these expressions, we have

$$\begin{aligned} &\left\langle \left\langle \Sigma^{-1}(w - w') + K(2x' - x) - Kx_{z,v}^* \right\rangle, \left(\partial F_v(K(2x' - x) + \Sigma^{-1}(w - w')) - \partial F_v(Kx_{z,v}^*) \right) \right\rangle \\ &\quad \ni \left\langle \left\langle \Sigma^{-1}(w - w') + K(2x' - x) - Kx_{z,v}^* \right\rangle, \left(\mathbb{T}^{-1}(x - x') - K^{\text{T}}(w - w_{z,v}^*) \right) \right\rangle \\ &\quad = \left\langle \left\langle \begin{pmatrix} x' - x_{z,v}^* \\ w' - w_{z,v}^* \end{pmatrix}, \left(\mathbb{T}^{-1}(x - x') - K^{\text{T}}(w - w') \right) \right\rangle, \left(\Sigma^{-1}(w - w') - K(x - x') \right) \right\rangle \quad \text{by reorganizing terms} \\ &\quad = \left(\begin{pmatrix} x' - x_{z,v}^* \\ w' - w_{z,v}^* \end{pmatrix} \right)^{\text{T}} M \begin{pmatrix} x - x' \\ w - w' \end{pmatrix}. \end{aligned} \quad (47)$$

As in Chambolle and Pock (2011) we can calculate

$$\begin{aligned} &\left| \left(\begin{pmatrix} x' - x_{z,v}^* \\ w' - w_{z,v}^* \end{pmatrix} \right)^{\text{T}} M \begin{pmatrix} x - x' \\ w - w' \end{pmatrix} \right| \\ &= \frac{1}{2} \left\| \begin{pmatrix} x - x_{z,v}^* \\ w - w_{z,v}^* \end{pmatrix} \right\|_M^2 - \frac{1}{2} \left\| \begin{pmatrix} x' - x_{z,v}^* \\ w' - w_{z,v}^* \end{pmatrix} \right\|_M^2 - \frac{1}{2} \left\| \begin{pmatrix} x - x' \\ w - w' \end{pmatrix} \right\|_M^2. \end{aligned}$$

On the other hand, by the convexity of F_v and G_z as stated in Assumption 2, we have

$$\begin{aligned}
& \left\langle \left(\Sigma^{-1}(w-w') + K(2x' - x) - Kx_{z,v}^* \right), \left(\partial F_v(K(2x' - x) + \Sigma^{-1}(w-w')) - \partial F_v(Kx_{z,v}^*) \right) \right\rangle \\
& \geq \|x' - x_{z,v}^*\|_{\Lambda_C}^2 + \|\Sigma^{-1}(w-w') + K(2x' - x) - Kx_{z,v}^*\|_{\Lambda_F}^2 - \tau^2 \|x' - x_{z,v}^*\|_{\text{restrict}}^2 \\
& \geq \|x' - x_{z,v}^*\|_{\Lambda_C}^2 + \frac{1}{2} \|Kx' - Kx_{z,v}^*\|_{\Lambda_F}^2 - \|\Sigma^{-1}(w-w') - K(x-x')\|_{\Lambda_F}^2 \\
& \quad - \tau^2 \|x' - x_{z,v}^*\|_{\text{restrict}}^2 \quad \text{using the fact that } (a+b)^2 \geq \frac{1}{2}a^2 - b^2 \text{ for all } a, b \\
& \geq \frac{C_{\text{Cvx}}}{2} \|x' - x_{z,v}^*\|_2^2 - \frac{3\tau^2}{2} \|x' - x_{z,v}^*\|_{\text{restrict}}^2 \\
& \quad - \|\Sigma^{-1}(w-w') - K(x-x')\|_{\Lambda_F}^2 \quad \text{by Assumption 2} \\
& \geq \frac{C_{\text{Cvx}}}{2} \|x' - x_{z,v}^*\|_2^2 - \frac{3\tau^2}{2} \|x' - x_{z,v}^*\|_{\text{restrict}}^2 - \left\| \begin{pmatrix} x-x' \\ w-w' \end{pmatrix} \right\|_{\Lambda_F}^2 \cdot \|M^{-1}\| \|\Lambda_F\| (\|\Sigma^{-1}\| + \|K\|) \\
& \geq \frac{C_{\text{Cvx}}}{2} \|x' - x_{z,v}^*\|_2^2 - 6\tau^2 R^2 - \left\| \begin{pmatrix} x-x' \\ w-w' \end{pmatrix} \right\|_{\Lambda_F}^2 \cdot 2(C_{\text{matrix}})^3,
\end{aligned}$$

where the last step holds because $\|x'\|_{\text{restrict}}, \|x_{z,v}^*\|_{\text{restrict}} \leq R$, since x' and $x_{z,v}^*$ must both lie in $\text{dom}(G) = \text{dom}(G_z)$ by their definitions. Now, examining these calculations, we see that the left-hand side must be nonnegative, so we can also write

$$\begin{aligned}
& \left\langle \left(\Sigma^{-1}(w-w') + K(2x' - x) - Kx_{z,v}^* \right), \left(\partial F_v(K(2x' - x) + \Sigma^{-1}(w-w')) - \partial F_v(Kx_{z,v}^*) \right) \right\rangle \\
& \geq c \left(\frac{C_{\text{Cvx}}}{2} \|x' - x_{z,v}^*\|_2^2 - 6\tau^2 R^2 - \left\| \begin{pmatrix} x-x' \\ w-w' \end{pmatrix} \right\|_{\Lambda_F}^2 \cdot 2(C_{\text{matrix}})^3 \right)
\end{aligned}$$

for any $c \in [0, 1]$. Choosing $c = \frac{1}{\max\{12, 8C_{\text{matrix}}\}}$, we obtain

$$\begin{aligned}
& \left\langle \left(\Sigma^{-1}(w-w') + K(2x' - x) - Kx_{z,v}^* \right), \left(\partial F_v(K(2x' - x) + \Sigma^{-1}(w-w')) - \partial F_v(Kx_{z,v}^*) \right) \right\rangle \\
& \geq \frac{C_{\text{Cvx}}}{2 \max\{12, 8C_{\text{matrix}}\}} \|x' - x_{z,v}^*\|_2^2 - \frac{1}{2} \tau^2 R^2 - \frac{1}{4} \left\| \begin{pmatrix} x-x' \\ w-w' \end{pmatrix} \right\|_{\Lambda_F}^2
\end{aligned}$$

Combining all our work, then,

$$\begin{aligned}
& \frac{C_{\text{Cvx}}}{4 \max\{12, 8C_{\text{matrix}}\}} \|x' - x_{z,v}^*\|_2^2 \\
& \leq \left\| \begin{pmatrix} x-x_{z,v}^* \\ w-w_{z,v}^* \end{pmatrix} \right\|_M^2 - \left\| \begin{pmatrix} x'-x_{z,v}^* \\ w'-w_{z,v}^* \end{pmatrix} \right\|_M^2 - \frac{1}{2} \left\| \begin{pmatrix} x-x' \\ w-w' \end{pmatrix} \right\|_M^2 + \tau^2 R^2.
\end{aligned}$$

Setting $C_{\text{monotone}} = \min \left\{ \frac{1}{2C_{\text{matrix}}}, \frac{1}{4 \max\{12, 8C_{\text{matrix}}\}} \right\}$, we have proved the lemma. \blacksquare

Now we turn to the proof of Lemma 4. By Lemma 5, for each $\ell = 1, \dots, L$, we have

$$\begin{aligned}
C_{\text{monotone}} & \left(\|x^{(\ell)} - x_{z,v}^*\|_2^2 + \left\| \begin{pmatrix} x^{(\ell-1)} - x^{(\ell)} \\ w^{(\ell-1)} - w^{(\ell)} \end{pmatrix} \right\|_2^2 \right) \\
& \leq \left\| \begin{pmatrix} x^{(\ell-1)} - x_{z,v}^* \\ w^{(\ell-1)} - w_{z,v}^* \end{pmatrix} \right\|_M^2 - \left\| \begin{pmatrix} x^{(\ell)} - x_{z,v}^* \\ w^{(\ell)} - w_{z,v}^* \end{pmatrix} \right\|_M^2 + \tau^2 R^2.
\end{aligned}$$

Summing this inequality over $\ell = 1, \dots, L$, taking a telescoping sum on the right-hand side, and dividing by L , we have

$$\frac{C_{\text{monotone}}}{L} \sum_{\ell=1}^L \left(\|x^{(\ell)} - x_{z,v}^*\|_2^2 + \left\| \begin{pmatrix} x^{(\ell-1)} - x^{(\ell)} \\ w^{(\ell-1)} - w^{(\ell)} \end{pmatrix} \right\|_2^2 \right) \leq \frac{1}{L} \left\| \begin{pmatrix} x^{(0)} - x_{z,v}^* \\ w^{(0)} - w_{z,v}^* \end{pmatrix} \right\|_M^2 + \tau^2 R^2.$$

Next, by convexity of $w \mapsto \|w\|_2^2$, we have

$$\|\tilde{x} - x_{z,v}^*\|_2^2 \leq \frac{1}{L} \sum_{\ell=1}^L \|x^{(\ell)} - x_{z,v}^*\|_2^2$$

and

$$\left\| \frac{1}{L} \begin{pmatrix} x^{(0)} - x^{(L)} \\ w^{(0)} - w^{(L)} \end{pmatrix} \right\|_2^2 \leq \frac{1}{L} \sum_{\ell=1}^L \left\| \begin{pmatrix} x^{(\ell-1)} - x^{(\ell)} \\ w^{(\ell-1)} - w^{(\ell)} \end{pmatrix} \right\|_2^2.$$

So,

$$C_{\text{monotone}} \left(\|\tilde{x} - x_{z,v}^*\|_2^2 + \left\| \frac{1}{L} \begin{pmatrix} x^{(0)} - x^{(L)} \\ w^{(0)} - w^{(L)} \end{pmatrix} \right\|_2^2 \right) \leq \frac{1}{L} \left\| \begin{pmatrix} x^{(0)} - x_{z,v}^* \\ w^{(0)} - w_{z,v}^* \end{pmatrix} \right\|_M^2 + \tau^2 R^2.$$

Next, by definition of \tilde{z}, \tilde{v} , we can write

$$\begin{aligned}
& \left\| \begin{pmatrix} \tilde{z} - x_{z,v}^* \\ \tilde{v} - Kx_{z,v}^* \end{pmatrix} \right\|_{\Theta} \leq \left\| \begin{pmatrix} \tilde{x} - x_{z,v}^* \\ K\tilde{x} - Kx_{z,v}^* \end{pmatrix} \right\|_{\Theta} \\
& \quad + \frac{1}{L} \left\| \begin{pmatrix} 0 \\ \Sigma^{-1}(w^{(0)} - w^{(L)}) + K(x^{(L)} - x^{(0)}) \end{pmatrix} \right\|_{\Theta}
\end{aligned}$$

and so,

$$\begin{aligned}
& \left\| \begin{pmatrix} \tilde{z} - x_{z,v}^* \\ \tilde{v} - Kx_{z,v}^* \end{pmatrix} \right\|_{\Theta}^2 \leq \|\Theta\| (1 + 2\|K\| + \|\Sigma^{-1}\|)^2 \left(\|\tilde{x} - x_{z,v}^*\|_2^2 + \left\| \frac{1}{L} \begin{pmatrix} x^{(0)} - x^{(L)} \\ w^{(0)} - w^{(L)} \end{pmatrix} \right\|_2^2 \right) \\
& \leq \frac{\|\Theta\| (1 + 2\|K\| + \|\Sigma^{-1}\|)^2}{C_{\text{monotone}}} \left(\frac{1}{L} \left\| \begin{pmatrix} x^{(0)} - x_{z,v}^* \\ w^{(0)} - w_{z,v}^* \end{pmatrix} \right\|_M^2 + \tau^2 R^2 \right) \\
& \leq \frac{C_{\text{matrix}}(1 + 3C_{\text{matrix}})^2}{C_{\text{monotone}}} \left(\frac{1}{L} \left\| \begin{pmatrix} x^{(0)} - x_{z,v}^* \\ w^{(0)} - w_{z,v}^* \end{pmatrix} \right\|_M^2 + \tau^2 R^2 \right).
\end{aligned}$$

Finally, for each ℓ , by definition of the step,

$$w^{(\ell)} \in \partial \mathbf{F}_v \left(\Sigma^{-1}(w^{(\ell-1)} - w^{(\ell)}) + K\bar{x}^{(\ell)} \right)$$

while

$$w_{z,v}^* \in \partial \mathbf{F}_v(Kx_{z,v}^*).$$

Therefore, by Assumption 2,

$$\|w^{(\ell)} - w_{z,v}^*\|_2 \leq C_{\text{Lip}} + C_{\text{grad}} \left\| \Sigma^{-1}(w^{(\ell-1)} - w^{(\ell)}) + K\bar{x}^{(\ell)} - Kx_{z,v}^* \right\|_2.$$

By convexity, then,

$$\begin{aligned} & \|\tilde{w} - w_{z,v}^*\|_2 \\ & \leq \frac{1}{L} \sum_{\ell=1}^L \left(C_{\text{Lip}} + C_{\text{grad}} \left\| \Sigma^{-1}(w^{(\ell-1)} - w^{(\ell)}) + K\bar{x}^{(\ell)} - Kx_{z,v}^* \right\|_2 \right) \\ & \leq C_{\text{Lip}} + C_{\text{grad}} (\|K\| + \|\Sigma^{-1}\|) \sqrt{\frac{1}{L} \sum_{\ell=1}^L \left\| \left(\begin{array}{c} x^{(\ell-1)} - x^{(\ell)} \\ w^{(\ell-1)} - w^{(\ell)} \end{array} \right) \right\|_2^2} \\ & \leq C_{\text{Lip}} + C_{\text{grad}} (\|K\| + \|\Sigma^{-1}\|) \sqrt{\frac{1}{C_{\text{monotone}}} \left(\frac{1}{L} \left\| \left(\begin{array}{c} x^{(0)} - x_{z,v}^* \\ w^{(0)} - w_{z,v}^* \end{array} \right) \right\|_M^2 + \tau R^2 \right)} \\ & \leq C_{\text{Lip}} + \frac{C_{\text{grad}} (\|K\| + \|\Sigma^{-1}\|)}{C_{\text{monotone}}} \left(\frac{1}{\sqrt{L}} \left\| \left(\begin{array}{c} x^{(0)} - x_{z,v}^* \\ w^{(0)} - w_{z,v}^* \end{array} \right) \right\|_M + \tau R \right). \end{aligned}$$

Combining everything, this proves that

$$\begin{aligned} & \left\| \left(\begin{array}{c} \tilde{x} - x_{z,v}^* \\ \tilde{w} - w_{z,v}^* \end{array} \right) \right\|_2 \leq C_{\text{Lip}} + \frac{1 + C_{\text{grad}} (\|K\| + \|\Sigma^{-1}\|)}{C_{\text{monotone}}} \left(\frac{1}{\sqrt{L}} \left\| \left(\begin{array}{c} x^{(0)} - x_{z,v}^* \\ w^{(0)} - w_{z,v}^* \end{array} \right) \right\|_M + \tau R \right) \\ & \leq C_{\text{Lip}} + \max\{1, \sqrt{\|M\|}\} \frac{1 + C_{\text{grad}} (\|K\| + \|\Sigma^{-1}\|)}{C_{\text{monotone}}} \left(\frac{1}{\sqrt{L}} \left\| \left(\begin{array}{c} x^{(0)} - x_{z,v}^* \\ w^{(0)} - w_{z,v}^* \end{array} \right) \right\|_2 + \tau R \right) \\ & \leq C_{\text{Lip}} + \max\{1, \sqrt{C_{\text{matrix}}}\} \frac{1 + 2C_{\text{grad}} C_{\text{matrix}}}{C_{\text{monotone}}} \left(\frac{1}{\sqrt{L}} \left\| \left(\begin{array}{c} x^{(0)} - x_{z,v}^* \\ w^{(0)} - w_{z,v}^* \end{array} \right) \right\|_2 + \tau R \right). \end{aligned}$$

Finally, defining

$$C_{\text{iter}} = \max \left\{ \sqrt{\frac{C_{\text{matrix}}(1 + 3C_{\text{matrix}})^2}{C_{\text{monotone}}}}, \max\{1, \sqrt{C_{\text{matrix}}}\} \frac{1 + 2C_{\text{grad}} C_{\text{matrix}}}{C_{\text{monotone}}} \right\},$$

we have proved the lemma.

References

- Rina Foygel Barber, Emil Y. Sidky, Taly Glat Schmidt, and Xiaochuan Pan. An algorithm for constrained one-step inversion of spectral CT data. *Physics in Medicine and Biology*, 61(10):3784–3818, 2016.
- Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.
- Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2):459–494, 2014.
- Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.
- Emmanuel J Candes, Michael B Wakin, and Stephen P Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications*, 14(5-6):877–905, 2008.
- Antonin Chambolle and Jérôme Darbon. On total variation minimization and surface evolution using parametric maximum flows. *International Journal of Computer Vision*, 84(3):288–307, 2009.
- Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.
- Antonin Chambolle and Thomas Pock. A remark on accelerated block coordinate descent for computing the proximity operators of a sum of convex functions. *SMAI Journal of Computational Mathematics*, 1:29–54, 2015.
- Rick Chartrand. Exact reconstruction of sparse signals via nonconvex minimization. *IEEE Signal Processing Letters*, 14(10):707–710, 2007.
- Aditya Chopra and Heng Lian. Total variation, adaptive total variation and nonconvex smoothly clipped absolute deviation penalty for denoising blocky images. *Pattern Recognition*, 43(8):2609–2619, 2010.
- Ernie Esser, Xiaogun Zhang, and Tony Chan. A general framework for a class of first order primal-dual algorithms for TV minimization. *UCLA CAM Report*, pages 09–67, 2009.
- Jiangtao Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.
- Saeed Ghadimi and Guanghui Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming*, 156(1):59–99, 2016.

- Bingsheng He and Xiaoning Yuan. Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective. *SIAM Journal on Imaging Sciences*, 5(1):119–149, 2012.
- Mingyi Hong, Zhi-Quan Luo, and Meisam Razaviyayn. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization*, 26(1):337–364, 2016.
- David R Hunter and Kenneth Lange. Quantile regression via an MM algorithm. *Journal of Computational and Graphical Statistics*, 9(1):60–77, 2000.
- Nicholas A Johnson. A dynamic programming algorithm for the fused lasso and ℓ_0 -segmentation. *Journal of Computational and Graphical Statistics*, 22(2):246–260, 2013.
- Keith Knight and Wenjiang Fu. Asymptotics for lasso-type estimators. *Annals of Statistics*, 28(5):1356–1378, 2000.
- Guoyin Li and Ting Kei Pong. Global convergence of splitting methods for nonconvex composite optimization. *SIAM Journal on Optimization*, 25(4):2434–2460, 2015.
- Huan Li and Zhouchen Lin. Accelerated proximal gradient methods for nonconvex programming. In *Advances in Neural Information Processing Systems*, pages 379–387, 2015.
- Po-Ling Loh and Martin J Wainwright. High-dimensional regression with noisy and missing data: Provable guarantees with non-convexity. In *Advances in Neural Information Processing Systems*, pages 2726–2734, 2011.
- Po-Ling Loh and Martin J Wainwright. Regularized M-estimators with nonconvexity: Statistical and algorithmic theory for local optima. In *Advances in Neural Information Processing Systems*, pages 476–484, 2013.
- Chengyu Lu and Hua Huang. $TV + TV_2$ regularization with nonconvex sparseness-inducing penalty for image restoration. *Mathematical Problems in Engineering*, 2014:790547, 2014.
- Sindri Magnússon, Pradeep Chathuranga Weeraddana, Michael G Rabbat, and Carlo Fischione. On the convergence of alternating direction Lagrangian methods for nonconvex structured optimization problems. *IEEE Transactions on Control of Network Systems*, 2015.
- MATLAB. *Version 8.6.0 (R2015b)*. The MathWorks Inc., Natick, Massachusetts, 2015.
- Sahand Negahban, Bin Yu, Martin J Wainwright, and Pradeep K Ravikumar. A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers. In *Advances in Neural Information Processing Systems*, pages 1348–1356, 2009.
- Yu Nesterov. Gradient methods for minimizing composite functions, 2013.
- Peter Ochs, Yunjin Chen, Thomas Brox, and Thomas Pock. iPiano: inertial proximal algorithm for nonconvex optimization. *SIAM Journal on Imaging Sciences*, 7(2):1388–1419, 2014.
- Peter Ochs, Alexey Dosovitskiy, Thomas Brox, and Thomas Pock. On iteratively reweighted algorithms for nonsmooth nonconvex optimization in computer vision. *SIAM Journal on Imaging Sciences*, 8:331–372, 2015.
- James M Ortega and Werner C Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. SIAM, 1970.
- Ankit Parekh and Ivan W Selesnick. Convex fused lasso denoising with non-convex regularization and its use for pulse detection. In *2015 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pages 1–6. IEEE, 2015.
- Thomas Pock and Antonin Chambolle. Diagonal preconditioning for first order primal-dual algorithms in convex optimization. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1762–1769. IEEE, 2011.
- R Tyrrell Rockafellar. Convex analysis, 1997.
- Leonid I Rudin, Stanley Osher, and Enad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1):259–268, 1992.
- Ivan W Selesnick, Ankit Parekh, and Ilker Bayram. Convex 1-d total variation denoising with non-convex regularization. *IEEE Signal Processing Letters*, 22(2):141–144, 2015.
- Emil Y Sidky, Rick Chartrand, John M Boone, and Xiaochuan Pan. Constrained minimization for enhanced exploitation of gradient sparsity: application to CT image reconstruction. *IEEE Journal of Translational Engineering in Health and Medicine*, 2:1800418, 2014.
- Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.
- Tuomo Valkonen. A primal-dual hybrid gradient method for nonlinear operators with applications to MRI. *Inverse Problems*, 30(5):055012, 2014.
- Fenghui Wang, Zongben Xu, and Hong-Kun Xu. Convergence of Bregman alternating direction method with multipliers for nonconvex composite problems. *arXiv preprint arXiv:1410.8625*, 2014a.
- Fenghui Wang, Wenfei Cao, and Zongben Xu. Convergence of multi-block Bregman ADMM for nonconvex composite problems. *arXiv preprint arXiv:1505.03063*, 2015a.
- Huahua Wang and Arindam Banerjee. Bregman alternating direction method of multipliers. In *Advances in Neural Information Processing Systems*, pages 2816–2824, 2014.
- Huahua Wang, Arindam Banerjee, and Zhi-Quan Luo. Parallel direction method of multipliers. In *Advances in Neural Information Processing Systems*, pages 181–189, 2014b.
- Yu-Xiang Wang, James Sharpnack, Alexander J Smola, and Ryan J Tibshirani. Trend filtering on graphs. In *Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015b.

Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894–942, 2010.

True Online Temporal-Difference Learning

Harm van Seijen^{†‡}

A. Rupam Mahmood[†]

Patrick M. Pilarski[†]

Marlos C. Machado[†]

Richard S. Sutton[†]

HARM.VANSEIJEN@MALUUBA.COM

ASHIQUE@UALBERTA.CA

PATRICK.PILARSKI@UALBERTA.CA

MACHADO@UALBERTA.CA

SUTTON@CS.UALBERTA.CA

[†]*Reinforcement Learning and Artificial Intelligence Laboratory*

University of Alberta

2-21 Athabasca Hall, Edmonton, AB

Canada, T6G 2E8

[‡]*Maluba Research*

2000 Peel Street, Montreal, QC

Canada, H3A 2W5

Editor: George Konidaris

Abstract

The temporal-difference methods TD(λ) and Sarsa(λ) form a core part of modern reinforcement learning. Their appeal comes from their good performance, low computational cost, and their simple interpretation, given by their forward view. Recently, new versions of these methods were introduced, called true online TD(λ) and true online Sarsa(λ), respectively (van Seijen & Sutton, 2014). Algorithmically, these true online methods only make two small changes to the update rules of the regular methods, and the extra computational cost is negligible in most cases. However, they follow the ideas underlying the forward view much more closely. In particular, they maintain an exact equivalence with the forward view at all times, whereas the traditional versions only approximate it for small step-sizes. We hypothesize that these true online methods not only have better theoretical properties, but also dominate the regular methods empirically. In this article, we put this hypothesis to the test by performing an extensive empirical comparison. Specifically, we compare the performance of true online TD(λ)/Sarsa(λ) with regular TD(λ)/Sarsa(λ) on random MRPs, a real-world myoelectric prosthetic arm, and a domain from the Arcade Learning Environment. We use linear function approximation with tabular, binary, and non-binary features. Our results suggest that the true online methods indeed dominate the regular methods. Across all domains/representations the learning speed of the true online methods are often better, but never worse than that of the regular methods. An additional advantage is that no choice between traces has to be made for the true online methods. Besides the empirical results, we provide an in-depth analysis of the theory behind true online temporal-difference learning. In addition, we show that new true online temporal-difference methods can be derived by making changes to the online forward view and then rewriting the update equations.

Keywords: temporal-difference learning, eligibility traces, forward-view equivalence

1. Introduction

Temporal-difference (TD) learning is a core learning technique in modern reinforcement learning (Sutton, 1988; Kaelbling et al., 1996; Sutton & Barto, 1998; Szepesvári, 2010). One of the main challenges in reinforcement learning is to make predictions, in an initially unknown environment, about the (discounted) sum of future rewards, the return, based on currently observed feature values and a certain behaviour policy. With TD learning it is possible to learn good estimates of the expected return quickly by bootstrapping from other expected-return estimates. TD(λ) (Sutton, 1988) is a popular TD algorithm that combines basic TD learning with eligibility traces to further speed learning. The popularity of TD(λ) can be explained by its simple implementation, its low-computational complexity and its conceptually straightforward interpretation, given by its forward view. The forward view of TD(λ) states that the estimate at each time step is moved towards an update target known as the λ -return, with λ determining the fundamental trade-off between bias and variance of the update target. This trade-off has a large influence on the speed of learning and its optimal setting varies from domain to domain. The ability to improve this trade-off by adjusting the value of λ is what underlies the performance advantage of eligibility traces.

Although the forward view provides a clear intuition, TD(λ) closely approximates the forward view only for appropriately small step-sizes. Until recently, this was considered an unfortunate, but unavoidable part of the theory behind TD(λ). This changed with the introduction of true online TD(λ) (van Seijen & Sutton, 2014), which computes exactly the same weight vectors as the forward view at any step-size. This gives true online TD(λ) full control over the bias-variance trade-off. In particular, true online TD(λ) can achieve fully unbiased updates. Moreover, true online TD(λ) only requires small modifications to the TD(λ) update equations, and the extra computational cost is negligible in most cases.

We hypothesize that true online TD(λ), and its control version true online Sarsa(λ), not only have better theoretical properties than their regular counterparts, but also dominate them empirically. We test this hypothesis by performing an extensive empirical comparison between true online TD(λ), regular TD(λ) (which is based on accumulating traces), and the common variation based on replacing traces. In addition, we perform comparisons between true online Sarsa(λ) and Sarsa(λ) (with accumulating and replacing traces). The domains we use include random Markov reward processes, a real-world myoelectric prosthetic arm, and a domain from the Arcade Learning Environment (Bellemare et al., 2013). The representations we consider range from tabular values to linear function approximation with binary and non-binary features.

Besides the empirical study, we provide an in-depth discussion on the theory behind true online TD(λ). This theory is based on a new online forward view. The traditional forward view, based on the λ -return, is inherently an offline forward view meaning that updates only occur at the end of an episode, because the λ -return requires data up to the end of an episode. We extend this forward view to the online case, where updates occur at every time step, by using a bounded version of the λ -return that grows over time. Whereas TD(λ) approximates the traditional forward view only at the end of an episode, we show that TD(λ) approximates this new online forward view at all time steps. True online TD(λ) is equivalent to this new online forward view at all time steps. We prove this by deriving the true online TD(λ) update equations directly from the online forward

view update equations. This derivation forms a blueprint for the derivation of other true online methods. By making variations to the online forward view and following the same derivation as for true online TD(λ), we derive several other true online methods.

This article is organized as follows. We start by presenting the required background in Section 2. Then, we present the new online forward view in Section 3, followed by the presentation of true online TD(λ) in Section 4. Section 5 presents the empirical study. Furthermore, in Section 6, we present several other true online methods. In Section 7, we discuss in detail related papers. Finally, Section 8 concludes.

2. Background

Here, we present the main learning framework. As a convention, we indicate scalar-valued random variables by capital letters (e.g. S_t, R_t), vectors by bold lowercase letters (e.g. θ, ϕ), functions by non-bold lowercase letters (e.g. v, γ), and sets by calligraphic font (e.g. S, \mathcal{A}).¹

2.1 Markov Decision Processes

Reinforcement learning (RL) problems are often formalized as *Markov decision processes* (MDPs), which can be described as 5-tuples of the form $(S, \mathcal{A}, p, r, \gamma)$, where S indicates the set of all states; \mathcal{A} indicates the set of all actions; $p(s'|s, a)$ indicates the probability of a transition to state $s' \in S$, when action $a \in \mathcal{A}$ is taken in state $s \in S$; $r(s, a, s')$ indicates the expected reward for a transition from state s to state s' under action a ; the discount factor γ specifies how future rewards are weighted with respect to the immediate reward.

Actions are taken at discrete time steps $t = 0, 1, 2, \dots$ according to a *policy* $\pi : S \times \mathcal{A} \rightarrow [0, 1]$, which defines for each action the selection probability conditioned on the state. The *return* at time t is defined as the discounted sum of rewards, observed after t :

$$G_t := R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots = \sum_{i=1}^{\infty} \gamma^{i-1} R_{t+i},$$

where R_{t+1} is the reward received after taking action A_t in state S_t . Some MDPs contain special states called *terminal states*. After reaching a terminal state, no further reward is obtained and no further state transitions occur. Hence, a terminal state can be interpreted as a state where each action returns to itself with a reward of 0. An interaction sequence from the initial state to a terminal state is called an *episode*.

Each policy π has a corresponding state-value function v_π , which maps any state $s \in S$ to the expected value of the return from that state, when following policy π :

$$v_\pi(s) := \mathbb{E}\{G_t | S_t = s, \pi\}.$$

In addition, the action-value function q_π gives the expected return for policy π , given that action $a \in \mathcal{A}$ is taken in state $s \in S$:

$$q_\pi(s, a) := \mathbb{E}\{G_t | S_t = s, A_t = a, \pi\}.$$

¹ An exception to this convention is the TD error, a scalar-valued random variable that we indicate by δ_t .

Because no further rewards can be obtained from a terminal state, the state-value and action-values for a terminal state are always 0.

There are two tasks that are typically associated with an MDP. First, there is the task of determining (an estimate of) the value function v_π for some given policy π . The second, more challenging task is that of determining (an estimate of) the optimal policy π_* , which is defined as the policy whose corresponding value function has the highest value in each state:

$$v_{\pi_*}(s) := \max_{\pi} v_{\pi}(s), \quad \text{for each } s \in S.$$

In RL, these two tasks are considered under the condition that the reward function r and the transition-probability function p are unknown. Hence, the tasks have to be solved using samples obtained from interacting with the environment.

2.2 Temporal-Difference Learning

Let's consider the task of learning an estimate V of the value function v_π from samples, where v_π is being estimated using linear function approximation. That is, V is the inner product between a feature vector $\phi(s) \in \mathbb{R}^n$ of s , and a weight vector $\theta \in \mathbb{R}^n$:

$$V(s, \theta) = \theta^\top \phi(s).$$

If s is a terminal state, then by definition $\phi(s) := \mathbf{0}$, and hence $V(s, \theta) = 0$.

We can formulate the problem of estimating v_π as an error-minimization problem, where the error is a weighted average of the squared difference between the value of a state and its estimate:

$$E(\theta) := \frac{1}{2} \sum_{\delta} d_\pi(s_t) \left(v_\pi(s_t) - \theta^\top \phi(s_t) \right)^2,$$

with d_π the stationary distribution induced by π . The above error function can be minimized by using stochastic gradient descent while sampling from the stationary distribution, resulting in the following update rule:

$$\theta_{t+1} = \theta_t - \alpha \frac{1}{2} \nabla_{\theta} \left(v_\pi(S_t) - \theta^\top \phi_t \right)^2,$$

using ϕ_t as a shorthand for $\phi(S_t)$. The parameter α is called the *step-size*. Using the chain rule, we can rewrite this update as:

$$\begin{aligned} \theta_{t+1} &= \theta_t + \alpha \left(v_\pi(S_t) - \theta^\top \phi_t \right) \nabla_{\theta} \left(\theta^\top \phi_t \right), \\ &= \theta_t + \alpha \left(v_\pi(S_t) - \theta^\top \phi_t \right) \phi_t. \end{aligned}$$

Because v_π is in general unknown, an estimate U_t of $v_\pi(S_t)$ is used, which we call the *update target*, resulting in the following general update rule:

$$\theta_{t+1} = \theta_t + \alpha (U_t - \theta^\top \phi_t) \phi_t. \quad (1)$$

There are many different update targets possible. For an unbiased estimator the full return can be used, that is, $U_t = G_t$. However, the full return has the disadvantage that its

variance is typically very high. Hence, learning with the full return can be slow. Temporal-difference (TD) learning addresses this issue by using update targets based on other value estimates. While the update target is no longer unbiased in this case, the variance is typically much smaller, and learning much faster. TD learning uses the Bellman equations as its mathematical foundation for constructing update targets. These equations relate the value of a state to the values of its successor states:

$$v_\pi(s) = \sum_a \pi(s, a) \sum_{s'} p(s' | s, a) (r(s, a, s') + \gamma v_\pi(s')).$$

Writing this equation in terms of an expectation yields:

$$v_\pi(s) = \mathbb{E}\{R_{t+1} + \gamma v_\pi(S_{t+1}) | S_t = s\}_{\pi, p, r}.$$

Sampling from this expectation, while using linear function approximation to approximate v_π , results in the update target:

$$U_t = R_{t+1} + \gamma \boldsymbol{\theta}^\top \boldsymbol{\phi}_{t+1}.$$

This update target is called a one-step update target, because it is based on information from only one time step ahead. Applying the Bellman equation multiple times results in update targets based on information further ahead. Such update targets are called multi-step update targets.

2.3 TD(λ)

The TD(λ) algorithm implements the following update equations:

$$\delta_t = R_{t+1} + \gamma \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_{t+1} - \boldsymbol{\theta}_t^\top \boldsymbol{\phi}_t, \quad (2)$$

$$\mathbf{e}_t = \gamma \lambda \mathbf{e}_{t-1} + \boldsymbol{\phi}_t, \quad (3)$$

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \delta_t \mathbf{e}_t, \quad (4)$$

for $t \geq 0$, and with $\mathbf{e}_{-1} = \mathbf{0}$. The scalar δ_t is called the *TD error*, and the vector \mathbf{e}_t is called the *eligibility-trace* vector. The update of \mathbf{e}_t shown above is referred to as the *accumulating-trace* update. As a shorthand, we will refer to this version of TD(λ) as ‘accumulate TD(λ)’, to distinguish it from a slightly different version that is discussed below. While these updates appear to deviate from the general, gradient-descent-based update rule given in (1), there is a close connection to this update rule. This connection is formalized through the forward view of TD(λ), which we discuss in detail in the next section. Algorithm 1 shows the pseudocode for accumulate TD(λ).

Accumulate TD(λ) can be very sensitive with respect to the α and λ parameters. Especially, a large value of λ combined with a large value of α can easily cause divergence, even on simple tasks with bounded rewards. For this reason, a variant of TD(λ) is sometimes used that is more robust with respect to these parameters. This variant, which assumes binary features, uses a different trace-update equation:

$$\mathbf{e}_t[i] = \begin{cases} \gamma \lambda \mathbf{e}_{t-1}[i], & \text{if } \boldsymbol{\phi}_t[i] = 0; \\ 1, & \text{if } \boldsymbol{\phi}_t[i] = 1, \end{cases} \quad \text{for all features } i.$$

Algorithm 1 accumulate TD(λ)

INPUT: $\alpha, \lambda, \gamma, \boldsymbol{\theta}_{init}$

$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}_{init}$

Loop (over episodes):

 obtain initial $\boldsymbol{\phi}$

$\mathbf{e} \leftarrow \mathbf{0}$

 While terminal state has not been reached, do:

 obtain next feature vector $\boldsymbol{\phi}'$ and reward R

$\delta \leftarrow R + \gamma \boldsymbol{\theta}^\top \boldsymbol{\phi}' - \boldsymbol{\theta}^\top \boldsymbol{\phi}$

$\mathbf{e} \leftarrow \gamma \lambda \mathbf{e} + \boldsymbol{\phi}$

$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \delta \mathbf{e}$

$\boldsymbol{\phi} \leftarrow \boldsymbol{\phi}'$

where $\mathbf{x}[i]$ indicates the i -th component of vector \mathbf{x} . This update is referred to as the *replacing-trace* update. As a shorthand, we will refer to the version of TD(λ) using the replacing-trace update as ‘replace TD(λ)’.

3. The Online Forward View

The traditional forward view relates the TD(λ) update equations to the general update rule shown in Equation (1). Specifically, for small step-sizes the weight vector at the end of an episode computed by accumulate TD(λ) is approximately the same as the weight vector resulting from a sequence of Equation (1) updates (one for each visited state) using a particular multi-step update target, called the λ -return (Sutton & Barto, 1998; Bertsekas & Tsitsiklis, 1996). The λ -return for state S_t is defined as:

$$G_t^{(\lambda)} := (1 - \lambda) \sum_{n=1}^{T-t-1} \lambda^{n-1} G_t^{(n)} + \lambda^{T-t-1} G_t, \quad (5)$$

where T is the time step the terminal state is reached, and $G_t^{(n)}$ is the n -step return, defined as:

$$G_t^{(n)} := \sum_{k=1}^n \gamma^{k-1} R_{t+k} + \gamma^n V(S_{t+n} | \boldsymbol{\theta}_{t+n-1}).$$

We call a method that updates the value of each visited state at the end of the episode an *offline* method; we call a method that updates the value of each visited state immediately after the visit (i.e., at the time step after the visit) an *online* method. TD(λ) is an online method. The update sequence of the traditional forward view, however, corresponds with an offline method, because the λ -return requires data up to the end of an episode. This leaves open the question of how to interpret the weights of TD(λ) *during* an episode. In this section, we provide an answer to this long-standing open question. We introduce a bounded version of the λ -return that only uses information up to a certain horizon and we use this to construct an online forward view. This online forward view approximates the weight vectors of accumulate TD(λ) at *all* time steps, instead of only at the end of an episode.

3.1 The Online λ -Return Algorithm

The concept of an online forward view contains a paradox. On the one hand, multi-step update targets require data from time steps far beyond the time a state is visited; on the other hand, the online aspect requires that the value of a visited state is updated immediately. The solution to this paradox is to assign a sequence of update targets to each visited state. The first update target in this sequence contains data from only the next time step, the second contains data from the next two time steps, the third from the next three time steps, and so on. Now, given an initial weight vector and a sequence of visited states, a new weight vector can be constructed by updating each visited state with an update target that contains data up to the current time step. Below, we formalize this idea.

We define the *interim λ -return* for state S_k with horizon $h \in \mathbb{N}^+$, $h > k$ as follows:

$$G_k^{\lambda|h} := (1 - \lambda) \sum_{n=1}^{h-k-1} \lambda^{n-1} G_k^{(n)} + \lambda^{h-k-1} G_k^{(h-k)}. \quad (6)$$

Note that this update target does not use data beyond the horizon h . $G_k^{\lambda|h}$ implicitly defines a sequence of update targets for S_k : $\{G_k^{\lambda|k+1}, G_k^{\lambda|k+2}, G_k^{\lambda|k+3}, \dots\}$. As time increases, update targets based on data further away become available for state S_k . At a particular time step t , a new weight vector is computed by performing an Equation (1) update for each visited state using the interim λ -return with horizon t , starting from the initial weight vector θ_{init} . Hence, at time step t , a sequence of t updates occurs. To describe this sequence mathematically, we use weight vectors with two indices: θ_k^t . The superscript indicates the time step at which the updates are performed (this value corresponds with the horizon of the interim λ -returns that are used in the updates). The subscript is the iteration index of the sequence (it corresponds with the number of updates that have been performed at a particular time step). As an example, the update sequences for the first three time steps are:

$$\begin{aligned} t = 1 : \quad & \theta_1^1 = \theta_0^1 + \alpha(G_0^{\lambda|1} - (\theta_0^1)^\top \phi_0) \phi_0, \\ t = 2 : \quad & \theta_1^2 = \theta_0^2 + \alpha(G_0^{\lambda|2} - (\theta_0^2)^\top \phi_0) \phi_0, \\ & \theta_2^2 = \theta_1^2 + \alpha(G_1^{\lambda|2} - (\theta_1^2)^\top \phi_1) \phi_1, \\ t = 3 : \quad & \theta_1^3 = \theta_0^3 + \alpha(G_0^{\lambda|3} - (\theta_0^3)^\top \phi_0) \phi_0, \\ & \theta_2^3 = \theta_1^3 + \alpha(G_1^{\lambda|3} - (\theta_1^3)^\top \phi_1) \phi_1, \\ & \theta_3^3 = \theta_2^3 + \alpha(G_2^{\lambda|3} - (\theta_2^3)^\top \phi_2) \phi_2, \end{aligned}$$

with $\theta_0^t := \theta_{init}$ for all t . More generally, the update sequence at time step t is:

$$\theta_{k+1}^t := \theta_k^t + \alpha \left(G_k^{\lambda|t} - (\theta_k^t)^\top \phi_k \right) \phi_k, \quad \text{for } 0 \leq k < t. \quad (7)$$

We define θ_t (without superscript) as the final weight vector of the update sequence at time t , that is, $\theta_t := \theta_t^t$. We call the algorithm implementing Equation (7) the *online λ -return*

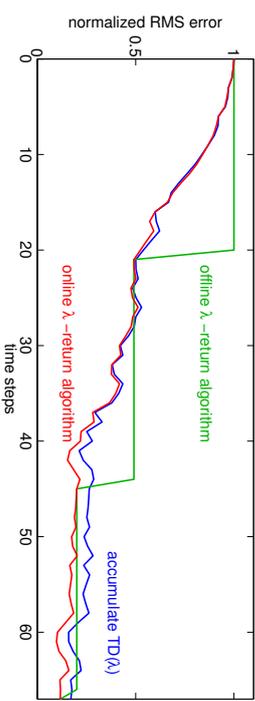


Figure 1: RMS error as function of time for the first 3 episodes of a random walk task, for $\lambda = 1$ and $\alpha = 0.2$. The error shown is the RMS error over all states, normalized by the initial RMS error.

algorithm. By contrast, we call the algorithm that implements the traditional forward view the *offline λ -return algorithm*.

The update sequence performed by the online λ -return algorithm at time step T (the time step that a terminal state is reached) is very similar to the update sequence performed by the offline λ -return algorithm. In particular, note that $G_t^{\lambda|T}$ and G_t^λ are the same, under the assumption that the weights used for the value estimates are the same. Because these weights are in practise not exactly the same, there will typically be a small difference.²

Figure 1 illustrates the difference between the online and offline λ -return algorithm, as well as accumulate TD(λ), by showing the RMS error on a random walk task. The task consists of 10 states laid out in a row plus a terminal state on the left. Each state transitions with 70% probability to its left neighbour and with 30% probability to its right neighbour (or to itself in case of the right-most state). All rewards are 1 and $\gamma = 1$. Furthermore, $\lambda = 1$ and $\alpha = 0.2$. The right-most state is the initial state. Whereas the offline λ -return algorithm only makes updates at the end of an episode, the online λ -return algorithm, as well as accumulate TD(λ), make updates at every time step.

The comparison on the random walk task shows that accumulate TD(λ) behaves similar to the online λ -return algorithm. In fact, the smaller the step-size, the smaller the difference between accumulate TD(λ) and the online λ -return algorithm. This is formalized by Theorem 1. The proof of the theorem can be found in Appendix A. The theorem uses the term Δ_t^f , which is defined as:

$$\Delta_t^f := (G_t^{\lambda|t} - \theta_0^\top \phi_t) \phi_t,$$

with $G_t^{\lambda|t}$ the interim λ -return for state S_t with horizon t that uses θ_0 for all value evaluations. Note that Δ_t^f is independent of the step-size.

² If $\lambda = 1$ there is never a difference because there is no bootstrapping.

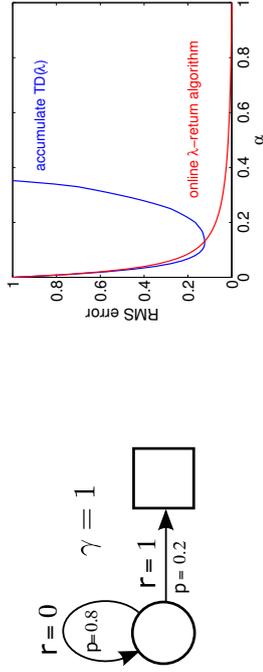


Figure 2: *Left:* One-state example (the square indicates a terminal state). *Right:* The RMS error of the state value at the end of an episode, averaged over the first 10 episodes, for $\lambda = 1$.

Theorem 1 Let θ_0 be the initial weight vector, θ_t^{id} be the weight vector at time t computed by accumulate TD(λ), and θ_t^λ be the weight vector at time t computed by the online λ -return algorithm. Furthermore, assume that $\sum_{i=0}^{t-1} \Delta_i^i \neq \mathbf{0}$. Then, for all time steps t :

$$\frac{\|\theta_t^{id} - \theta_t^\lambda\|}{\|\theta_t^{id} - \theta_0\|} \rightarrow 0, \quad \text{as } \alpha \rightarrow 0.$$

Theorem 1 generalizes the traditional result to arbitrary time steps. The traditional result states that the difference between the weight vector at the end of an episode computed by the offline λ -return algorithm and the weight vector at the end of an episode computed by accumulate TD(λ) goes to 0, if the step-size goes to 0 (Bertsekas & Tsitsiklis, 1996).

3.2 Comparison to Accumulate TD(λ)

While accumulate TD(λ) behaves like the online λ -return algorithm for small step-sizes, small step-sizes often result in slow learning. Hence, higher step-sizes are desirable. For higher step-sizes, however, the behaviour of accumulate TD(λ) can be very different from that of the online λ -return algorithm. And as we show in the empirical section of this article (Section 5), when there is a difference, it is almost exclusively in favour of the online λ -return algorithm. In this section, we analyze why the online λ -return algorithm can outperform accumulate TD(λ), using the one-state example shown in the left of Figure 2.

The right of Figure 2 shows the RMS error over the first 10 episodes of the one-state example for different step-sizes and $\lambda = 1$. While for small step-sizes accumulate TD(λ) behaves indeed like the online λ -return algorithm—as predicted by Theorem 1—, for larger step-sizes the difference becomes huge. To understand the reason for this, we derive an analytical expression for the value at the end of an episode.

First, we consider accumulate TD(λ). Because there is only one state involved, we indicate the value of this state simply by V . The update at the end of an episode is $V_T =$

$V_{T-1} + \alpha e_{T-1} \delta_{T-1}$. In our example, $\delta_t = 0$ for all time steps t , except for $t = T - 1$, where $\delta_{T-1} = 1 - V_{T-1}$. Because δ_t is 0 for all time steps except the last, $V_{T-1} = V_0$. Furthermore, $\phi_t = \mathbf{1}$ for all time steps t , resulting in $e_{T-1} = T$. Substituting all this in the expression for V_T yields:

$$V_T = V_0 + T\alpha(1 - V_0), \quad \text{for accumulate TD}(\lambda). \quad (8)$$

So for accumulate TD(λ), the total value difference is simply a summation of the value difference corresponding to a single update.

Now, consider the online λ -return algorithm. The value at the end of an episode, V_T , is equal to V_T^T , resulting from the update sequence:

$$V_{k+1}^T = V_k^T + \alpha(G_k^{\lambda|T} - V_k^T), \quad \text{for } 0 \leq k < T.$$

By incremental substitution, we can directly express V_T in terms of the initial value, V_0 , and the update targets:

$$V_T = (1 - \alpha)^T V_0 + \alpha(1 - \alpha)^{T-1} G_0^{\lambda|T} + \alpha(1 - \alpha)^{T-2} G_1^{\lambda|T} + \dots + \alpha G_{T-1}^{\lambda|T}.$$

Because $G_k^{\lambda|T} = \mathbf{1}$ for all k in our example, the weights of all update targets can be added together and the expression can be rewritten as a single pseudo-update, yielding:

$$V_T = V_0 + (1 - (1 - \alpha)^T) \cdot (1 - V_0), \quad \text{for the online } \lambda\text{-return algorithm.} \quad (9)$$

The term $1 - (1 - \alpha)^T$ in (9) acts like a pseudo step-size. For larger α or T this pseudo step-size increases in value, but as long as $\alpha \leq 1$ the value will never exceed 1. By contrast, for accumulate TD(λ) the pseudo step-size is $T\alpha$, which can grow much larger than 1 even for $\alpha < 1$, causing divergence of values. This is the reason that accumulate TD(λ) can be very sensitive to the step-size and it explains why the optimal step-size for accumulate TD(λ) is much smaller than the optimal step-size for the online λ -return algorithm in Figure 2 ($\alpha \approx 0.15$ versus $\alpha = 1$, respectively). Moreover, because the variance on the pseudo step-size is higher for accumulate TD(λ) the performance at the optimal step-size for accumulate TD(λ) is worse than the performance at the optimal step-size for the online λ -return algorithm.

3.3 Comparison to Replace TD(λ)

The sensitivity of accumulate TD(λ) to divergence, demonstrated in the previous subsection, has been known for long. In fact, replace TD(λ) was designed to deal with this. But while replace TD(λ) is much more robust with respect to divergence, it also has its limitations. One obvious limitation is that it only applies to binary features, so it is not generally applicable. But even in domains where replace TD(λ) can be applied, it can perform poorly. The reason is that replacing previous trace values, rather than adding to it, reduces the multi-step characteristics of TD(λ).

To illustrate this, consider the two-state example shown in the left of Figure 3. It is easy to see that the value of the left-most state is 2 and of the other state is 0. The state representation consists of only a single, binary feature that is 1 in both states and 0 in the terminal state. Because there is only a single feature, the state values cannot be represented

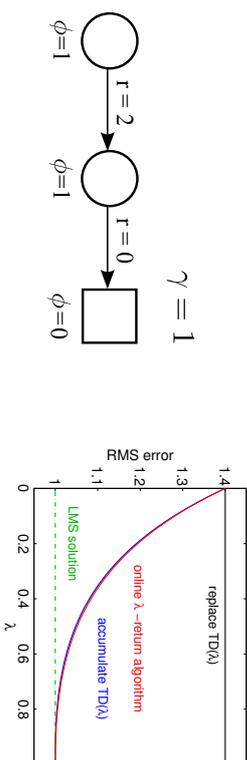


Figure 3: *Left*: Two-state example. *Right*: The RMS error after convergence for different λ (at $\alpha = 0.01$). We consider values to be converged if the error changed less than 1% over the last 100 time steps.

exactly. The weight that minimizes the mean squared error assigns a value of 1 to both states, resulting in an RMS error of 1. Now consider the graph shown in the right of Figure 3, which shows the asymptotic RMS error for different values of λ . The error for accumulate TD(λ) converges to the least mean squares (LMS) error for $\lambda = 1$, as predicted by the theory (Dayan, 1992). The online λ -return algorithm has the same convergence behaviour (due to Theorem 1). By contrast, replace TD(λ) converges to the same value as TD(0) for any value of λ . The reason for this behaviour is that because the single feature is active at all time steps, the multi-step behaviour of TD(λ) is fully removed, no matter the value of λ . Hence, replace TD(λ) behaves exactly the same as TD(0) for any value of λ at all time steps. As a result, it also behaves like TD(0) asymptotically.

The two-state example very clearly demonstrates that there is a price payed by replace TD(λ) to achieve robustness with respect to divergence: a reduction in multi-step behaviour. By contrast, the online λ -return algorithm, which is also robust to divergence, does not have this disadvantage. Of course, the two-state example, as well as the one-state example from the previous section, are extreme examples, merely meant to illustrate what can go wrong. But in practise, a domain will often have some characteristics of the one-state example and some of the two-state example, which negatively impacts the performance of both accumulate and replace TD(λ).

4. True Online TD(λ)

The online λ -return algorithm is impractical on many domains: the memory it uses, as well as the computation required per time step increases linearly with time. Fortunately, it is possible to rewrite the update equations of the online λ -return algorithm to a different set of update equations that can be implemented with a computational complexity that is independent of time. In fact, this alternative set of update equations differs from the update equations of accumulate TD(λ) only by two extra terms, each of which can be computed

Algorithm 2 true online TD(λ)

```

INPUT:  $\alpha, \lambda, \gamma, \theta_{init}$ 
 $\theta \leftarrow \theta_{init}$ 
Loop (over episodes):
  obtain initial  $\phi$ 
   $e \leftarrow 0$ ;  $V_{old} \leftarrow 0$ 
  While terminal state has not been reached, do:
    obtain next feature vector  $\phi'$  and reward  $R$ 
     $V \leftarrow \theta^\top \phi$ 
     $V' \leftarrow \theta'^\top \phi'$ 
     $\delta \leftarrow R + \gamma V' - V$ 
     $e \leftarrow \gamma \lambda e + \phi - \alpha \gamma \lambda (e^\top \phi)$ 
     $\theta \leftarrow \theta + \alpha (\delta + V - V_{old}) e - \alpha (V - V_{old}) \phi$ 
     $V_{old} \leftarrow V$ 
     $\phi \leftarrow \phi'$ 

```

efficiently. The algorithm implementing these equations is called true online TD(λ) and is discussed below.

4.1. The Algorithm

For the online λ -return algorithm, at each time step a sequence of updates is performed. The length of this sequence, and hence the computation per time step, increases over time. However, it is possible to compute the weight vector resulting from the sequence at time step $t + 1$ directly from the weight vector resulting from the sequence at time step t . This results in the following update equations (see Appendix B for the derivation):

$$\delta_t = R_{t+1} + \gamma \theta_t^\top \phi_{t+1} - \theta_t^\top \phi_t, \quad (10)$$

$$e_t = \gamma \lambda e_{t-1} + \phi_t - \alpha \gamma \lambda (e_{t-1}^\top \phi_t) \phi_t, \quad (11)$$

$$\theta_{t+1} = \theta_t + \alpha \delta_t e_t + \alpha (\theta_t^\top \phi_t - \theta_{t-1}^\top \phi_t) (e_t - \phi_t), \quad (12)$$

for $t \geq 0$, and with $e_{-1} = 0$. Compared to accumulate TD(λ), both the trace update and the weight update have an additional term. We call a trace updated in this way a *dutch trace*: we call the term $\alpha (\theta_t^\top \phi_t - \theta_{t-1}^\top \phi_t) (e_t - \phi_t)$ the *TD-error time-step correction*, or simply the δ -correction. Algorithm 2 shows pseudocode that implements these equations.³

In terms of computation time, true online TD(λ) has a (slightly) higher cost due to the two extra terms that have to be accounted for. While the computation-time complexity of true online TD(λ) is the same as that of accumulate/replace TD(λ)— $\mathcal{O}(n)$ per time step with n being the number of features—the actual computation time can be close to twice as much in some cases. In other cases (for example if sparse feature vectors are used), the computation time of true online TD(λ) is only a fraction more than that of

³ When using a time-dependent step-size (e.g., when annealing the step-size) use the pseudocode from Section 6.1. For reasons explained in that section this requires a modified trace update. That pseudocode is the same as the pseudocode from van Seijen & Sutton (2014).

accumulate/replace $\text{TD}(\lambda)$. In terms of memory, true online $\text{TD}(\lambda)$ has the same cost as accumulate/replace $\text{TD}(\lambda)$.

4.2 When Can a Performance Difference be Expected?

In Section 3, a number of examples were shown where the online λ -return algorithm outperforms accumulate/replace $\text{TD}(\lambda)$. Because true online $\text{TD}(\lambda)$ is simply an efficient implementation of the online λ -return algorithm, true online $\text{TD}(\lambda)$ will outperform accumulate/replace $\text{TD}(\lambda)$ on these examples as well. But not in all cases will there be a performance difference. For example, it follows from Theorem 1 that when appropriately small step-sizes are used, the difference between the online λ -return algorithm/true online $\text{TD}(\lambda)$ and accumulate $\text{TD}(\lambda)$ is negligible. In this section, we identify two other factors that affect whether or not there will be a performance difference. While the focus of this section is on performance *difference* rather than performance *advantage*, our experiments will show that true online $\text{TD}(\lambda)$ performs always at least as well as accumulate $\text{TD}(\lambda)$ and replace $\text{TD}(\lambda)$. In other words, our experiments suggest that whenever there is a performance difference, it is in favour of true online $\text{TD}(\lambda)$.

The first factor is the λ parameter and follows straightforwardly from the true online $\text{TD}(\lambda)$ update equations.

Proposition 1 For $\lambda = 0$, accumulate $\text{TD}(\lambda)$, replace $\text{TD}(\lambda)$ and the online λ -return algorithm / true online $\text{TD}(\lambda)$ behave the same.

Proof For $\lambda = 0$, the accumulating-trace update, the replacing-trace update and the dutch-trace update all reduce to $\mathbf{e}_t = \boldsymbol{\phi}_t$. In addition, because $\mathbf{e}_t = \boldsymbol{\phi}_t$, the δ -correction of true online $\text{TD}(\lambda)$ is 0. ■

Because the behaviour of $\text{TD}(\lambda)$ for small λ is close to the behaviour of $\text{TD}(0)$, it follows that significant performance differences will only be observed when λ is large.

The second factor is related to how often a feature has a non-zero value. We start again with a proposition that highlights a condition under which the different $\text{TD}(\lambda)$ versions behave the same. The proposition makes use of an accumulating trace at time step $t-1$, $\mathbf{e}_{t-1}^{\text{acc}}$, whose non-recursive form is:

$$\mathbf{e}_{t-1}^{\text{acc}} = \sum_{k=0}^{t-1} (\gamma\lambda)^{t-1-k} \boldsymbol{\phi}_k. \quad (13)$$

Furthermore, the proposition uses $\boldsymbol{x}[i]$ to denote the i -th element of vector \boldsymbol{x} .

Proposition 2 If for all features i and at all time steps t

$$\mathbf{e}_{t-1}^{\text{acc}}[i] \cdot \boldsymbol{\phi}_t[i] = 0, \quad (14)$$

then accumulate $\text{TD}(\lambda)$, replace $\text{TD}(\lambda)$ and the online λ -return algorithm / true online $\text{TD}(\lambda)$ behave the same (for any λ).

Proof Condition (14) implies that if $\boldsymbol{\phi}_t[i] \neq 0$, then $\mathbf{e}_{t-1}^{\text{acc}}[i] = 0$. From this it follows that for binary features the accumulating-trace update can be written as a replacing-trace

update at every time step:

$$\begin{aligned} \mathbf{e}_t^{\text{acc}}[i] &:= \gamma\lambda\mathbf{e}_{t-1}^{\text{acc}}[i] + \boldsymbol{\phi}_t[i], \\ &= \begin{cases} \gamma\lambda\mathbf{e}_{t-1}^{\text{acc}}[i], & \text{if } \boldsymbol{\phi}_t[i] = 0; \\ 1, & \text{if } \boldsymbol{\phi}_t[i] = 1. \end{cases} \end{aligned}$$

Hence, accumulate $\text{TD}(\lambda)$ and replace $\text{TD}(\lambda)$ perform exactly the same updates.

Furthermore, condition (14) implies that $(\mathbf{e}_{t-1}^{\text{acc}})^\top \boldsymbol{\phi}_t = 0$. Hence, the accumulating-trace update can also be written as a dutch trace update at every time step:

$$\begin{aligned} \mathbf{e}_t^{\text{acc}} &:= \gamma\lambda\mathbf{e}_{t-1}^{\text{acc}} + \boldsymbol{\phi}_t, \\ &= \gamma\lambda\mathbf{e}_{t-1}^{\text{acc}} + \boldsymbol{\phi}_t - \alpha\gamma\lambda((\mathbf{e}_{t-1}^{\text{acc}})^\top \boldsymbol{\phi}_t) \boldsymbol{\phi}_t. \end{aligned}$$

In addition, note that the δ -correction is proportional to $\boldsymbol{\theta}_t^\top \boldsymbol{\phi}_t - \boldsymbol{\theta}_{t-1}^\top \boldsymbol{\phi}_t$, which can be written as $(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1})^\top \boldsymbol{\phi}_t$. The value $(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1})^\top \boldsymbol{\phi}_t$ is proportional to $(\mathbf{e}_{t-1}^{\text{acc}})^\top \boldsymbol{\phi}_t$ for accumulate $\text{TD}(\lambda)$. Because $(\mathbf{e}_{t-1}^{\text{acc}})^\top \boldsymbol{\phi}_t = 0$, accumulate $\text{TD}(\lambda)$ can add a δ -correction at every time step without any consequence. This shows that accumulate $\text{TD}(\lambda)$ makes the same updates as true online $\text{TD}(\lambda)$. ■

An example of a domain where the condition of Proposition 2 holds is a domain with tabular features (each state is represented with a unique standard-basis vector), where a state is never revisited within the same episode.

The condition of Proposition 2 holds approximately when the value $|\mathbf{e}_{t-1}^{\text{acc}}[i] \cdot \boldsymbol{\phi}_t[i]|$ is close to 0 for all features at all time steps. In this case, the different $\text{TD}(\lambda)$ versions will perform very similarly. It follows from Equation (13) that this is the case when there is a long time delay between the time steps that a feature has a non-zero value. Specifically, if there is always at least n time steps between two subsequent times that a feature i has a non-zero value with $\gamma\lambda^n$ being very small, then $|\mathbf{e}_{t-1}^{\text{acc}}[i] \cdot \boldsymbol{\phi}_t[i]|$ will always be close to 0. Therefore, in order to see a large performance difference, the same features should have a non-zero value often and within a small time frame (relative to $\gamma\lambda$).

Summarizing the analysis so far: in order to see a performance difference α and λ should be sufficiently large, and the same features should have a non-zero value often and within a small time frame. Based on this summary, we can address a related question: on what type of domains will there be a performance difference between true online $\text{TD}(\lambda)$ with optimized parameters and accumulate/replace $\text{TD}(\lambda)$ with optimized parameters. The first two conditions suggest that the domain should result in a relatively large optimal α and optimal λ . This is typically the case for domains with a relatively low variance on the return. The last condition can be satisfied in multiple ways. It is for example satisfied by domains that have non-sparse feature vectors (that is, domains for which at any particular time step most features have a non-zero value).

4.3 True Online Sarsa(λ)

$\text{TD}(\lambda)$ and true online $\text{TD}(\lambda)$ are policy evaluation methods. However, they can be turned into control methods in a straightforward way. From a learning perspective, the main difference is that the prediction of the expected return should be conditioned on the state

Algorithm 3 true online Sarsa(λ)

INPUT: $\alpha, \lambda, \gamma, \theta_{init}$
 $\theta \leftarrow \theta_{init}$
Loop (over episodes):
 obtain initial state S
 select action A based on state S (for example ϵ -greedy)
 $\psi \leftarrow$ features corresponding to S, A
 $e \leftarrow \mathbf{0}$; $Q_{old} \leftarrow \mathbf{0}$
 While terminal state has not been reached, do:
 take action A , observe next state S' and reward R
 select action A' based on state S'
 $\psi' \leftarrow$ features corresponding to S', A' (if S' is terminal state, $\psi' \leftarrow \mathbf{0}$)
 $Q \leftarrow \theta^\top \psi$
 $Q' \leftarrow \theta'^\top \psi'$
 $\delta \leftarrow R + \gamma Q' - Q$
 $e \leftarrow \gamma \lambda e + \psi - \alpha \gamma \lambda (e^\top \psi) \psi$
 $\theta \leftarrow \theta + \alpha (\delta + Q - Q_{old}) e - \alpha (Q - Q_{old}) \psi$
 $Q_{old} \leftarrow Q'$
 $\psi \leftarrow \psi'$; $A \leftarrow A'$

and action, rather than only on the state. This means that an estimate of the action-value function q_π is being learned, rather than of the state-value function v_π .

Another difference is that instead of having a fixed policy that generates the behaviour, the policy depends on the action-value estimates. Because these estimates typically improve over time, so does the policy. The (on-policy) control counterpart of TD(λ) is the popular Sarsa(λ) algorithm. The control counterpart of true online TD(λ) is ‘true online Sarsa(λ)’. Algorithm 3 shows pseudocode for true online Sarsa(λ).

To ensure accurate estimates for all state-action values are obtained, typically some exploration strategy has to be used. A simple, but often sufficient strategy is to use an ϵ -greedy behaviour policy. That is, given current state S_t , with probability ϵ a random action is selected, and with probability $1 - \epsilon$ the greedy action is selected:

$$A_t^{greedy} = \arg \max_a \theta_t^\top \psi(S_t, a),$$

with $\psi(s, a)$ an action-feature vector, and $\theta_t^\top \psi(s, a)$ a (linear) estimate of $q_\pi(s, a)$ at time step t . A common way to derive an action-feature vector $\psi(s, a)$ from a state-feature vector $\phi(s)$ involves an action-feature vector of size $n|\mathcal{A}|$, where n is the number of state features and $|\mathcal{A}|$ is the number of actions. Each action corresponds with a block of n features in this action-feature vector. The features in $\psi(s, a)$ that correspond to action a take on the values of the state features; the features corresponding to other actions have a value of 0.

5. Empirical Study

This section contains our main empirical study, comparing TD(λ), as well as Sarsa(λ), with their true online counterparts. For each method and each domain, a scan over the step-size

α and the trace-decay parameter λ is performed such that the optimal performance can be compared. In Section 5.4, we discuss the results.

5.1 Random MRPs

For our first series of experiments we used randomly constructed Markov reward processes (MRPs).⁴ An MRP can be interpreted as an MDP with only a single action per state. Consequently, there is only one policy possible. We represent a random MRP as a 3-tuple (k, b, σ) , consisting of k , the number of states; b , the branching factor (that is, the number of next states with a non-zero transition probability); and σ , the standard deviation of the reward. An MRP is constructed as follows. The b potential next states for a particular state are drawn from the total set of states at random, and without replacement. The transition probabilities to those states are randomized as well (by partitioning the unit interval at $b - 1$ random cut points). The expected value of the reward for a transition is drawn from a normal distribution with zero mean and unit variance. The actual reward is drawn from a normal distribution with a mean equal to this expected reward and standard deviation σ . There are no terminal states.

We compared the performance of TD(λ) on three different MRPs: one with a small number of states, (10, 3, 0.1), one with a larger number of states, (100, 10, 0.1), and one with a larger number of states but a low branching factor and no stochasticity for the reward, (100, 3, 0). The discount factor γ is 0.99 for all three MRPs. Each MRP is evaluated using three different representations. The first representation consists of *tabular* features, that is, each state is represented with a unique standard-basis vector of k dimensions. The second representation is based on *binary* features. This binary representation is constructed by first assigning indices, from 1 to k , to all states. Then, the binary encoding of the state index is used as a feature vector to represent that state. The length of a feature vector is determined by the total number of states: for $k = 10$, the length is 4; for $k = 100$, the length is 7. As an example, for $k = 10$ the binary feature vectors of states 1, 2 and 3 are (0, 0, 0, 1), (0, 0, 1, 0) and (0, 0, 1, 1), respectively. Finally, the third representation uses non-binary features. For this representation each state is mapped to a 5-dimensional feature vector, with the value of each feature drawn from a normal distribution with zero mean and unit variance. After all the feature values for a state are drawn, they are normalized such that the feature vector has unit length. Once generated, the feature vectors are kept fixed for each state. Note that replace TD(λ) cannot be used with this representation, because replacing traces are only defined for binary features (tabular features are a special case of this).

In each experiment, we performed a scan over α and λ . Specifically, between 0 and 0.1, α is varied according to 10^i with i varying from -3 to -1 with steps of 0.2, and from 0.1 to 2.0 (linearly) with steps of 0.1. In addition, λ is varied from 0 to 0.9 with steps of 0.1 and from 0.9 to 1.0 with steps of 0.01. The initial weight vector is the zero vector in all domains. As performance metric we used the mean-squared error (MSE) with respect to the LMS solution during early learning (for $k = 10$, we averaged over the first 100 time steps; for k

⁴ The code for the MRP experiments is published online at: <https://github.com/armahmood/todd-trmdp-experiments>. The process we used to construct the MRPs is based on the process used by Bhatnagar, Sutton, Ghahraman and Lee (2009).

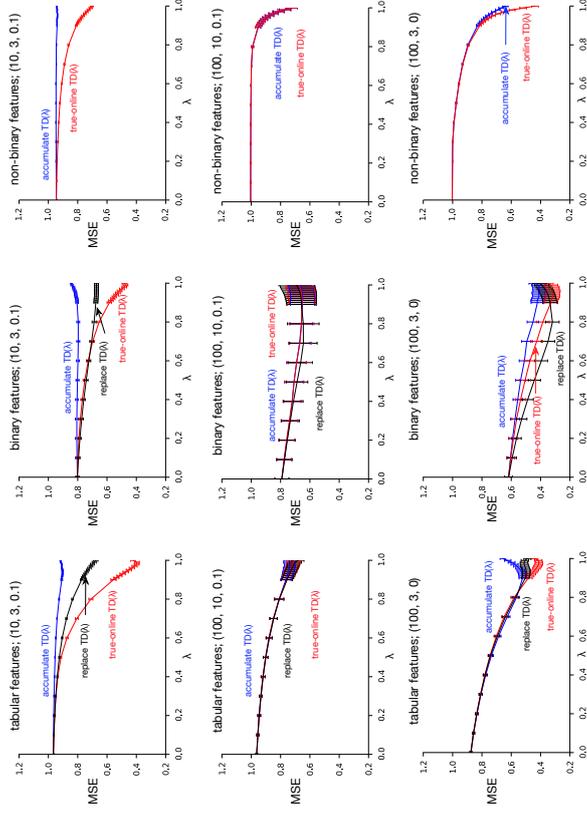


Figure 4: MSE error during early learning for three different MRPs, indicated by (k, b, σ) , and three different representations. The error shown is at optimal α value.

$= 100$, we averaged over the first 1000 time steps). We normalized this error by dividing it by the MSE under the initial weight estimate.

Figure 4 shows the results for different λ at the best value of α . In Appendix C, the results for all α values are shown. The optimal performance of true online TD(λ) is at least as good as the optimal performance of accumulate TD(λ) and replace TD(λ), on all domains and for all representations. A more in-depth discussion of these results is provided in Section 5.4.

5.2 Predicting Signals From a Myoelectric Prosthetic Arm

In this experiment, we compared the performance of true online TD(λ) and TD(λ) on a real-world data-set consisting of sensorimotor signals measured during the human control of an electromechanical robot arm. The source of the data is a series of manipulation tasks performed by a participant with an amputation, as presented by Pilarski et al. (2013). In this study, an amputee participant used signals recorded from the muscles of their residual

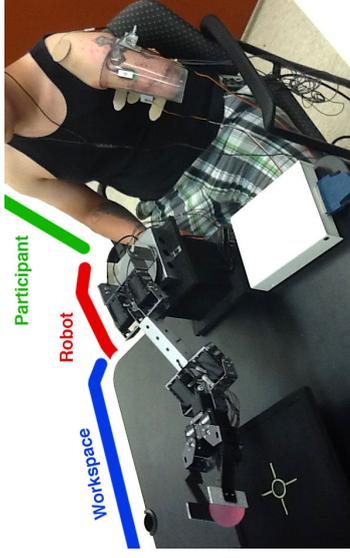


Figure 5: Source of the input data stream and predicted signals used in this experiment: a participant with an amputation performing a simple grasping task using a myoelectrically controlled robot arm, as described in Pilarski et al. (2013). More detail on the subject and experimental setting can be found in Hebert et al. (2014).

limb to control a robot arm with multiple degrees-of-freedom (Figure 5). Interactions of this kind are known as *myoelectric control* (see, for example, Parker et al., 2006).

For consistency and comparison of results, we used the same source data and prediction learning architecture as published in Pilarski et al. (2013). In total, two signals are predicted: grip force and motor angle signals from the robot’s hand. Specifically, the target for the prediction is a discounted sum of each signal over time, similar to return predictions (see general value functions and nexting; Sutton et al., 2011; Modayil et al., 2014). Where possible, we used the same implementation and code base as Pilarski et al. (2013). Data for this experiment consisted of 58,000 time steps of recorded sensorimotor information, sampled at 40 Hz (i.e., approximately 25 minutes of experimental data). The state space consisted of a tile-coded representation of the robot gripper’s position, velocity, recorded gripping force, and two muscle contraction signals from the human user. A standard implementation of tile-coding was used, with ten bins per signal, eight overlapping tilings, and a single active bias unit. This results in a state space with 800,001 features, 9 of which were active at any given time. Hashing was used to reduce this space down to a vector of 200,000 features that are then presented to the learning system. All signals were normalized between 0 and 1 before being provided to the function approximation routine. The discount factor for predictions of both force and angle was $\gamma = 0.97$, as in the results presented by Pilarski et al. (2013). Parameter sweeps over λ and α are conducted for all three methods. The performance metric is the mean absolute return error over all 58,000 time steps of learning, normalized by dividing by the error for $\lambda = 0$.

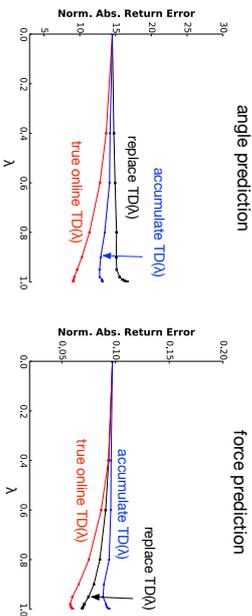


Figure 6: Performance as function of λ at the optimal α value, for the prediction of the servo motor angle (left), as well as the grip force (right).

Figure 6 shows the performance for the angle as well as the force predictions at the best α value for different values of λ . In Appendix D, the results for all α values are shown. The relative performance of replace TD(λ) and accumulate TD(λ) depends on the predictive question being asked. For predicting the robot’s grip force signal—a signal with small magnitude and rapid changes—replace TD(λ) is better than accumulate TD(λ) at all λ values larger than 0. However, for predicting the robot’s hand actuator position, a smoothly changing signal that varies between a range of ~ 300 –500, accumulate TD(λ) dominates replace TD(λ). On both prediction tasks, true online TD(λ) dominates accumulate TD(λ) and replace TD(λ).

5.3 Control in the ALE Domain Asterix

In this final experiment, we compared the performance of true online Sarsa(λ) with that of accumulate Sarsa(λ) and replace Sarsa(λ), on a domain from the Arcade Learning Environment (ALE) (Bellemare et al., 2013; Defazio & Graepel, 2014; Minh et al., 2015), called Asterix. The ALE is a general testbed that provides an interface to hundreds of Atari 2600 games.⁵

In the Asterix domain, the agent controls a yellow avatar, which has to collect ‘potion’ objects, while avoiding ‘harp’ objects (see Figure 7 for a screenshot). Both potions and harps move across the screen horizontally. Every time the agent collects a potion it receives a reward of 50 points, and every time it touches a harp it loses a life (it has three lives in total). The agent can use the actions *up*, *right*, *down*, and *left*, combinations of two directions, and a *no-op* action, resulting in 9 actions in total. The game ends after the agent has lost three lives, or after 5 minutes, whichever comes first.

We use linear function approximation using features derived from the screen pixels. Specifically, we use what Bellemare et al. (2013) call the *Basic* feature set, which ‘encodes

⁵ We used ALE version 0.4.4 for our experiments. The code for the Asterix experiments is published online at: <https://github.com/mcmachado/TrueOnlineSarsa>.

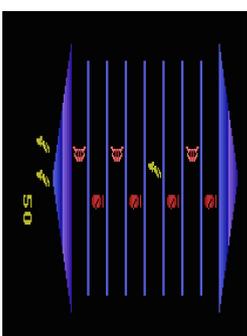


Figure 7: Screenshot of the game ASTERIX.

the presence of colours on the Atari 2600 screen.” It is obtained by first subtracting the game screen background (see Bellemare et al., 2013, sec. 3.1.1) and then dividing the remaining screen in to 16×14 tiles of size 10×15 pixels. Finally, for each tile, one binary feature is generated for each of the 128 available colours, encoding whether a colour is active or not in that tile. This generates 28,672 features (plus a bias term).

Because episode lengths can vary hugely (from about 10 seconds all the way up to 5 minutes), constructing a fair performance metric is non-trivial. For example, comparing the average return on the first N episodes of two methods is only fair if they have seen roughly the same amount of samples in those episodes, which is not guaranteed for this domain. On the other hand, looking at the total reward collected for the first X samples is also not a good metric, because there is no negative reward associated to dying. To resolve this, we look at the return per episode, averaged over the first X samples. More specifically, our metric consists of the average score per episode while learning for 20 hours (4,320,000 frames). In addition, we averaged the resulting number over 400 independent runs.

As with the evaluation experiments, we performed a scan over the step-size α and the trace-decay parameter λ . Specifically, we looked at all combinations of $\alpha \in \{0.20, 0.50, 0.80, 1.10, 1.40, 1.70, 2.00\}$ and $\lambda \in \{0.00, 0.50, 0.80, 0.90, 0.95, 0.99\}$ (these values were determined during a preliminary parameter sweep). We used a discount factor $\gamma = 0.999$ and ϵ -greedy exploration with $\epsilon = 0.01$. The weight vector was initialized to the zero vector. Also, as Bellemare et al. (2013), we take an action at each 5 frames. This decreases the algorithms running time and avoids ‘super-human’ reflexes. The results are shown in Figure 8. On this domain, the optimal performance of all three versions of Sarsa(λ) is similar.

Note that the way we evaluate a domain is computationally very expensive: we perform scans over λ and α , and use a large number of independent runs to get a low standard error. In the case of Asterix, this results in a total of $7 \cdot 6 \cdot 400 = 16,800$ runs per method. This rigorous evaluation prohibits us unfortunately to run experiments on the full suite of ALE domains.

5.4 Discussion

Figure 9 summarizes the performance of the different TD(λ) versions on all evaluation domains. Specifically, it shows the error for each method at its best settings of α and λ . The error is normalized by dividing it by the error at $\lambda = 0$ (remember that all versions of

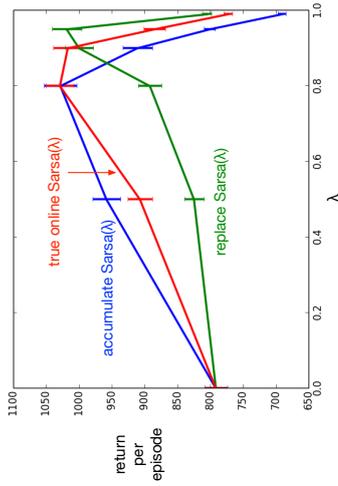


Figure 8: Return per episode, averaged over the first 4,320,000 frames as well as 400 independent runs, as function of λ , at optimal α , on the Asterix domain.

TD(λ) behave the same for $\lambda = 0$). Because $\lambda = 0$ lies in the parameter range that is being optimized over, the normalized error can never be higher than 1. If for a method/domain the normalized error is equal to 1, this means that setting λ higher than 0 either has no effect, or that the error gets worse. In either case, eligibility traces are not effective for that method/domain.

Overall, true online TD(λ) is clearly better than accumulate TD(λ) and replace TD(λ) in terms of optimal performance. Specifically, for each considered domain/representation, the error for true online TD(λ) is either smaller or equal to the error of accumulate/replace TD(λ). This is especially impressive, given the wide variety of domains, and the fact that the computational overhead for true online TD(λ) is small (see Section 4.1 for details).

The observed performance differences correspond well with the analysis from Section 4.2. In particular, note that MRP (10, 3, 0.1) has less states than the other two MRPs, and hence the chance that the same feature has a non-zero value within a small time frame is larger. The analysis correctly predicts that this results in larger performance differences. Furthermore, MRP (100, 3, 0) is less stochastic than MRP (100, 10, 0.1), and hence it has a smaller variance on the return. Also here, the experiments correspond with the analysis, which predicts that this results in a larger performance difference.

On the Asterix domain, the performance of the three Sarsa(λ) versions is similar. This is in accordance with the evaluation results, which showed that the size of the performance difference is domain dependent. In the worst case, the performance of the true online method is similar to that of the regular method.

The optimal performance is not the only factor that determines how good a method is; what also matters is how easy it is to find this performance. The detailed plots in appendices C and D reveal that the parameter sensitivity of accumulate TD(λ) is much higher than

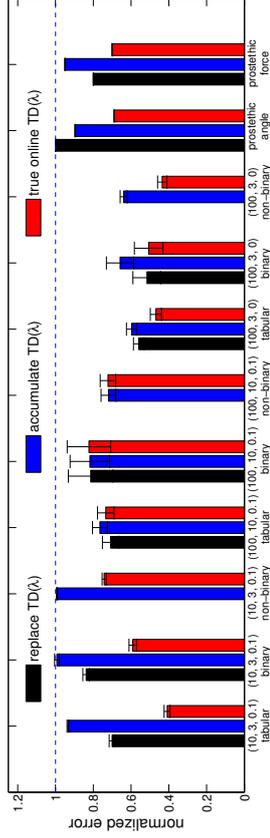


Figure 9: Summary of the evaluation results: error at optimal (α, λ) -settings for all domains/representations, normalized with the TD(0) error.

that of true online TD(λ) or replace TD(λ). This is clearly visible for MRP (10, 3, 0.1) (Figure 10), as well as the experiments with the myoelectric prosthetic arm (Figure 13).

There is one more thing to take away from the experiments. In MRP (10, 3, 0.1) with non-binary features, replace TD(λ) is not applicable and accumulate TD(λ) is ineffective. However, true online TD(λ) was still able to obtain a considerable performance advantage with respect to TD(0). This demonstrates that true online TD(λ) expands the set of domains/representations where eligibility traces are effective. This could potentially have far-reaching consequences. Specifically, using non-binary features becomes a lot more interesting. Replacing traces are not applicable to such representations, while accumulating traces can easily result in divergence of values. For true online TD(λ), however, non-binary features are not necessarily more challenging than binary features. Exploring new, non-binary representations could potentially further improve the performance for true online TD(λ) on domains such as the myoelectric prosthetic arm or the Asterix domain.

6. Other True Online Methods

In Appendix B, it is shown that the true online TD(λ) equations can be derived directly from the online forward view equations. By using different online forward views, new true online methods can be derived. Sometimes, small changes in the forward view, like using a time-dependent step-size, can result in surprising changes in the true online equations. In this section, we look at a number of such variations.

6.1 True Online TD(λ) with Time-Dependent Step-Size

When using a time-dependent step-size in the base equation of the forward view (Equation 7) and deriving the update equations following the procedure from Appendix B, it turns out that a slightly different trace definition appears. We indicate this new trace using a '+' superscript: e_t^+ . For fixed step-size, this new trace definition is equal to:

$$e_t^+ = \alpha e_t, \quad \text{for all } t.$$

Algorithm 4 true online TD(λ) with time-dependent step-size

INPUT: $\lambda, \theta_{init}, \alpha_t$ for $t \geq 0$
 $\theta \leftarrow \theta_{init}$; $t \leftarrow 0$
 Loop (over episodes):
 obtain initial ϕ
 $e^+ \leftarrow \mathbf{0}$; $V_{old} \leftarrow 0$
 While terminal state is not reached, do:
 obtain next feature vector ϕ' and reward R
 $V \leftarrow \theta^\top \phi$
 $V' \leftarrow \theta'^\top \phi'$
 $\delta' \leftarrow R + \gamma V' - V_{old}$
 $e^+ \leftarrow \gamma \lambda e^+ + \alpha_t \phi - \alpha_t \gamma \lambda ((e^+)^T \phi) \phi$
 $\theta \leftarrow \theta + \delta' e^+ - \alpha_t (V - V_{old}) \phi$
 $V_{old} \leftarrow V'$
 $\phi \leftarrow \phi'$
 $t \leftarrow t + 1$

Of course, using e_t^+ instead of e_t also changes the weight vector update slightly. Below, the full set of update equations is shown:

$$\begin{aligned} \delta_t &= R_{t+1} + \gamma \theta_t^\top \phi_{t+1} - \theta_t^\top \phi_t, \\ e_t^+ &= \gamma \lambda e_{t-1}^+ + \alpha_t \phi_t - \alpha_t \gamma \lambda ((e_{t-1}^+)^T \phi_t) \phi_t, \\ \theta_{t+1} &= \theta_t + \delta_t e_t^+ + (\theta_t^\top \phi_t - \theta_{t-1}^\top \phi_t) (e_t^+ - \alpha_t \phi_t). \end{aligned}$$

In addition, $e_{-1}^+ := 0$. We can simplify the weight update equation slightly, by using

$$\delta_t' = \delta_t + \theta_t^\top \phi_t - \theta_{t-1}^\top \phi_t,$$

which changes the update equations to:⁶

$$\delta_t' = R_{t+1} + \gamma \theta_t^\top \phi_{t+1} - \theta_{t-1}^\top \phi_t, \quad (15)$$

$$e_t^+ = \gamma \lambda e_{t-1}^+ + \alpha_t \phi_t - \alpha_t \gamma \lambda ((e_{t-1}^+)^T \phi_t) \phi_t, \quad (16)$$

$$\theta_{t+1} = \theta_t + \delta_t' e_t^+ - \alpha_t (\theta_t^\top \phi_t - \theta_{t-1}^\top \phi_t) \phi_t. \quad (17)$$

Algorithm 2 shows the corresponding pseudocode. Of course, this pseudocode can also be used for constant step-size.

6.2 True Online Version of Watkins's $Q(\lambda)$

So far, we just considered *on-policy* methods, that is, methods that evaluate a policy that is the same as the policy that generates the samples. However, the true online principle can also be applied to *off-policy* methods, for which the evaluation policy is different from the behaviour policy. As a simple example, consider Watkins's $Q(\lambda)$ (Watkins, 1989). This is an off-policy method that evaluates the greedy policy given an arbitrary behaviour policy.

6. These equations are the same as in the original true online paper (van Seijen & Sutton, 2014).

Algorithm 5 true online version of Watkins's $Q(\lambda)$

INPUT: $\alpha, \lambda, \gamma, \theta_{init}, \Psi$
 $\theta \leftarrow \theta_{init}$
 Loop (over episodes):
 obtain initial state S
 select action A based on state S (for example ϵ -greedy)
 $\psi \leftarrow$ features corresponding to S, A
 $e \leftarrow \mathbf{0}$; $Q_{old} \leftarrow 0$
 While terminal state has not been reached, do:
 take action A , observe next state S' and reward R
 select action A' based on state S'
 $A^* \leftarrow \arg \max_a \theta^\top \psi(S', a)$ (if A' tries for the max, then $A^* \leftarrow A'$)
 $\psi' \leftarrow$ features corresponding to S', A^* (if S' is terminal state, $\psi' \leftarrow \mathbf{0}$)
 $Q \leftarrow \theta^\top \psi$
 $Q' \leftarrow \theta'^\top \psi'$
 $\delta \leftarrow R + \gamma Q' - Q$
 $e \leftarrow \gamma \lambda e + \psi - \alpha \gamma \lambda (e^\top \psi) \psi$
 $\theta \leftarrow \theta + \alpha (\delta + Q - Q_{old}) e - \alpha (Q - Q_{old}) \psi$
 if $A' \neq A^*$: $e \leftarrow \mathbf{0}$
 $Q_{old} \leftarrow Q$
 $\psi \leftarrow \psi'$; $A \leftarrow A'$

It does this by combining accumulating traces with a TD error that uses the maximum state-action value of the successor state:

$$\delta_t = R_{t+1} + \max_a Q(S_t, a) - Q(S_t, A_t).$$

In addition, traces are reset to 0 whenever a non-greedy action is taken.

From an online forward-view perspective, the strategy of Watkins's $Q(\lambda)$ method can be interpreted as a growing update target that stops growing once a non-greedy action is taken. Specifically, let τ be the first time step *after* time step t that a non-greedy action is taken, then the interim update target for time step t can be defined as:

$$U_t^h := (1 - \lambda) \sum_{n=1}^{z-t-1} \lambda^{n-1} \tilde{G}_t^{(n)} + \lambda^{z-t-1} \tilde{G}_t^{(z-t)}, \quad z = \min\{h, \tau\},$$

with

$$\tilde{G}_t^{(n)} = \sum_{k=1}^n \gamma^{k-1} R_{t+k} + \gamma^n \max_a \theta_{t+n-1}^\top \psi(S_{t+n}, a).$$

Algorithm 5 shows the pseudocode for the true online method that corresponds with this update target definition. A problem with Watkins's $Q(\lambda)$ is that if the behaviour policy is very different from the greedy policy traces are reset very often, reducing the overall effect of the traces. In Section 7, we discuss more advanced off-policy methods.

Algorithm 6 tabular true online TD(λ)

```

initialize  $v(s)$  for all  $s$ 
Loop (over episodes):
  initialize  $S$ 
   $e(s) \leftarrow 0$  for all  $s$ 
   $V_{old} \leftarrow 0$ 
  While  $S$  is not terminal, do:
    obtain next state  $S'$  and reward  $R$ 
     $\Delta V \leftarrow V(S) - V_{old}$ 
     $V_{old} \leftarrow V(S)$ 
     $\delta \leftarrow R + \gamma V(S') - V(S)$ 
     $e(S) \leftarrow (1 - \alpha)e(S) + 1$ 
    For all  $s$ :
       $V(s) \leftarrow V(s) + \alpha(\delta + \Delta V)e(s)$ 
       $e(s) \leftarrow \gamma\lambda e(s)$ 
     $V(S) \leftarrow V(S) - \alpha\Delta V$ 
   $S \leftarrow S'$ 

```

6.3 Tabular True Online TD(λ)

Tabular features are a special case of linear function approximation. Hence, the update equations for true online TD(λ) that are presented so far also apply to the tabular case. However, we discuss it here separately, because the simplicity of this special case can provide extra insight.

Rewriting the true online update equations (equations 10 – 12) for the special case of tabular features results in:

$$\begin{aligned} \delta_t &= R_{t+1} + \gamma V_t(S_{t+1}) - V_t(S_t), \\ e_t(s) &= \begin{cases} \gamma\lambda e_{t-1}(s), & \text{if } s \neq S_t; \\ (1 - \alpha)\gamma\lambda e_{t-1}(s) + 1, & \text{if } s = S_t, \end{cases} \\ V_{t+1}(s) &= \begin{cases} V_t(s) + \alpha(\delta_t + V_t(S_t) - V_{t-1}(S_t))e_t(s), & \text{if } s \neq S_t; \\ V_t(s) + \alpha(\delta_t + V_t(S_t) - V_{t-1}(S_t))e_t(s) - \alpha(V_t(S_t) - V_{t-1}(S_t)), & \text{if } s = S_t. \end{cases} \end{aligned}$$

What is interesting about the tabular case is that the dutch-trace update reduces to a particularly simple form. In fact, for the tabular case, a dutch-trace update is equal to the weighted average between an accumulating-trace update and a replacing-trace update, with the weight of the former $(1 - \alpha)$ and the latter α . Algorithm 6 shows the corresponding pseudocode.

7. Related Work

Since the first publication on true online TD(λ) (van Seijen & Sutton, 2014), several related papers have been published, extending the underlying concepts and improving the presentation. In sections 7.1, 7.2 and 7.3, we review those papers. In Section 7.4, we discuss other variations of TD(λ).

7.1 True Online Learning and Dutch Traces

As mentioned before, the traditional forward view, which is based on the λ -return, is inherently an offline forward view, because the λ -return is constructed from data up to the end of an episode. As a consequence, the work regarding equivalence between a forward view and a backward view traditionally focused on the final weight vector θ_T . This changed in 2014, when two papers introduced an *online* forward view with a corresponding backward view that has an exact equivalence at each moment in time (van Seijen & Sutton, 2014; Sutton et al., 2014). While both papers introduced an online forward view, the two forward views presented are very different from each other. One difference is that the forward view introduced by van Seijen & Sutton is an on-policy forward view, whereas the forward view by Sutton et al. is an off-policy forward view. However, there is an even more fundamental difference related to how the forward views are constructed. In particular, the forward view by van Seijen & Sutton is constructed in such a way that at each moment in time the weight vector can be interpreted as the result of a sequence of updates of the form:

$$\theta_{k+1} = \theta_k + \alpha(U_k - \theta_k^\top \phi_k)\phi_k, \quad \text{for } 0 \leq k < t. \quad (18)$$

By contrast, the forward view by Sutton et al. gives the following interpretation:

$$\theta_t = \theta_0 + \alpha \sum_{k=0}^{t-1} \delta_k \phi_k, \quad (19)$$

with δ_k some multi-step TD error. Of course, the different forward views also result in different backward views. Whereas the backward view of Sutton et al. uses a generalized version of an accumulating trace, the backward view of van Seijen & Sutton introduced a completely new type of trace.

The advantage of a forward view based on (18) instead of (19) is that it results in much more stable updates. In particular, the sensitivity to divergence of accumulate TD(λ) is a general side-effect of (19), whereas (18) is much more robust with respect to divergence. As a result, true online TD(λ) not only has the property that it has an exact equivalence with an online forward view at all times, it consistently dominates TD(λ) empirically.

The strong performance of true online TD(λ) motivated van Hasselt et al. (2014) to construct an off-policy version of the forward view of true online TD(λ). The corresponding backward view resulted in the algorithm true online GTD(λ), which empirically outperforms GTD(λ). They also introduced the term ‘dutch traces’ for the new eligibility trace.

Van Hasselt & Sutton (2015) showed that dutch traces are not only useful for TD learning. In an offline setting with no bootstrapping using dutch traces can result in certain computational advantages. To understand why, consider the Monte Carlo algorithm (MC), which updates state values at the end of an episode using (18), with the full return as update target. MC requires storing all the feature vectors and rewards during an episode, so the memory complexity is linear in the length of the episode. Moreover, the required computation time is distributed very unevenly: during an episode almost no computation is required, but at the end of an episode there is a huge peak in computation time due to all the updates that need to be performed. With dutch traces an alternative implementation can be made that results in the same final weight vector but that does not require storing all

the feature vectors and where the required computation time is spread out evenly over all the time steps. Van Hasselt & Sutton refer to this appealing property as span-independence: the memory and computation time required per time step is constant and independent of the span of the prediction.⁷

7.2 Backward View Derivation

The task of finding an efficient backward view that corresponds exactly with a particular online forward view is not easy. Moreover, there is no guarantee that there exists an efficient implementation of a particular online forward view. Often, minor changes in the forward view determine whether or not an efficient backward view can be constructed. This created the desire to somehow automate the process of constructing an efficient backward view.

Van Seijen & Sutton (2014) did not provide a direct derivation of the backward view update equations; they simply proved that the forward view and the backward view equations result in the same weight vectors. Sutton et al. (2014) were the first to attempt to come up with a general strategy for deriving a backward view (although for forward views based on Equation 19). Van Hasselt et al. (2014) took the approach of providing a theorem that proves equivalence between a general forward view and a corresponding general backward view. They showed that the forward/backward view of true online TD(λ) is a special case of this general forward/backward view. They showed the same for the off-policy method that they introduced—true online GTD(λ). Recently, Mahmood & Sutton (2015) extended this theorem further by proving equivalence between an even more general forward view and backward view.

Furthermore, van Hasselt & Sutton (2015) derived backward views for a series of increasingly complex forward views. The derivation of the true online TD(λ) equations in Appendix B is similar to those derivations.

7.3 Extension to Non-Linear Function Approximation

The linear update equations of the online forward view presented in Section 3.1 can be easily extended to the case of non-linear function approximation. Unfortunately, it appears to be impossible to construct an efficient backward view with exact equivalence in the case of non-linear function approximation. The reason is that the derivation in Appendix B makes use of the fact that the gradient with respect to the value function is independent of the weight vector; this does not hold for non-linear function approximation.

Fortunately, van Seijen (2016) shows that many of the benefits of true online learning can also be achieved in the case of non-linear function approximation by using an alternative forward view (but still based on Equation 18). While this alternative forward view is not fully online (there is a delay in the updates), it can be implemented efficiently.

7.4 Other Variations on TD(λ)

Several variations on TD(λ) other than those treated in this article have been suggested in the literature. Schapire & Warmuth (1996) introduced a variation of TD(λ) for which

⁷ The span of the prediction refers to the time difference between the first prediction and the moment its target is known (e.g., for episodic tasks it corresponds to the length of an episode).

upper and lower bounds on performance can be derived and proven. Konidaris et al. (2011) introduced TD _{γ} , a parameter-free alternative to TD(λ) based on a multi-step update target called the γ -return. TD _{γ} is an offline algorithm with a computational cost proportional to the episode-length. Furthermore, Thomas et al. (2015) proposed a method based on a multi-step update target, which they call the Ω -return. The Ω -return can account for the correlation of different length returns, something that both the λ -return and the γ -return cannot. However, it is expensive to compute and it is open question whether efficient approximations exist.

8. Conclusions

We tested the hypothesis that true online TD(λ) (and true online Sarsa(λ)) dominates TD(λ) (and Sarsa(λ)) with accumulating as well as with replacing traces by performing experiments over a wide range of domains. Our extensive results support this hypothesis. In terms of learning speed, true online TD(λ) was often better, but never worse than TD(λ) with either accumulating or replacing traces, across all domains/representations that we tried. Our analysis showed that especially on domains with non-sparse features and a relatively low variance on the return a large difference in learning speed can be expected. More generally, true online TD(λ) has the advantage over TD(λ) with replacing traces that it can be used with non-binary features, and it has the advantage over TD(λ) with accumulating traces that it is less sensitive with respect to its parameters. In terms of computation time, TD(λ) has a slight advantage. In the worst case, true online TD(λ) is twice as expensive. In the typical case of sparse features, it is only fractionally more expensive than TD(λ). Memory requirements are the same. Finally, we outlined an approach for deriving new true online methods, based on rewriting the equations of an online forward view. This may lead to new, interesting methods in the future.

Acknowledgments

The authors thank Hado van Hasselt for extensive discussions leading to the refinement of these ideas. Furthermore, the authors thank the anonymous reviewers for their valuable suggestions, resulting in a substantially improved presentation. This work was supported by grants from Alberta Innovates – Technology Futures and the National Science and Engineering Research Council of Canada. Computing resources were provided by Compute Canada through WestGrid.

Appendix A. Proof of Theorem 1

Theorem 1 *Let θ_0 be the initial weight vector, θ_t^{id} be the weight vector at time t computed by accumulate TD(λ), and θ_t^λ be the weight vector at time t computed by the online λ -return algorithm. Furthermore, assume that $\sum_{i=0}^{t-1} \Delta_i^t \neq \mathbf{0}$. Then, for all time steps t :*

$$\frac{\|\theta_t^{id} - \theta_t^\lambda\|}{\|\theta_t^{id} - \theta_0\|} \rightarrow 0, \quad \text{as } \alpha \rightarrow 0.$$

Proof We prove the theorem by showing that $\|\theta_t^{id} - \theta_t^\lambda\| / \|\theta_t^{id} - \theta_0\|$ can be approximated by $\mathcal{O}(\alpha) / (C + \mathcal{O}(\alpha))$ as $\alpha \rightarrow 0$, with $C > 0$. For readability, we will not use the ‘td’ and ‘ λ ’ superscripts; instead, we always use weights with double indices for the online λ -return algorithm and weights with single indices for accumulate TD(λ).

The update equations for accumulate TD(λ) are:

$$\begin{aligned} \delta_t &= R_{t+1} + \gamma \theta_t^\top \phi_{t+1} - \theta_t^\top \phi_t, \\ \mathbf{e}_t &= \gamma \lambda \mathbf{e}_{t-1} + \phi_t, \\ \theta_{t+1} &= \theta_t + \alpha \delta_t \mathbf{e}_t. \end{aligned}$$

By incremental substitution, we can write θ_t directly in terms of θ_0 :

$$\begin{aligned} \theta_t &= \theta_0 + \alpha \sum_{j=0}^{t-1} \delta_j \mathbf{e}_j, \\ &= \theta_0 + \alpha \sum_{j=0}^{t-1} \delta_j \sum_{i=0}^j (\gamma \lambda)^{j-i} \phi_i, \\ &= \theta_0 + \alpha \sum_{j=0}^{t-1} \sum_{i=0}^j (\gamma \lambda)^{j-i} \delta_j \phi_i. \end{aligned} \tag{20}$$

Using the summation rule $\sum_{j=k}^n \sum_{i=k}^j a_{i,j} = \sum_{i=k}^n \sum_{j=i}^n a_{i,j}$ we can rewrite this as:

$$\theta_t = \theta_0 + \alpha \sum_{i=0}^{t-1} \sum_{j=i}^{t-1} (\gamma \lambda)^{j-i} \delta_j \phi_i. \tag{21}$$

As part of the derivation shown in Appendix B, we prove the following (see Equation 26):

$$G_i^{\lambda|h+1} = G_i^{\lambda|h} + (\lambda \gamma)^{h-i} \delta_i^h,$$

with

$$\delta_i^h := R_{h+1} + \gamma \theta_i^\top \phi_{h+1} - \theta_i^\top \phi_h.$$

By applying this sequentially for $i+1 \leq h < t$, we can derive:

$$G_i^{\lambda|h} = G_i^{\lambda|h+1} + \sum_{j=i+1}^{t-1} (\gamma \lambda)^{j-i} \delta_j^h. \tag{22}$$

Furthermore, $G_i^{\lambda|h+1}$ can be written as:

$$\begin{aligned} G_i^{\lambda|h+1} &= R_{i+1} + \gamma \theta_i^\top \phi_{i+1}, \\ &= R_{i+1} + \gamma \theta_i^\top \phi_{i+1} - \theta_{i-1}^\top \phi_i + \theta_{i-1}^\top \phi_i, \\ &= \delta_i^i + \theta_{i-1}^\top \phi_i. \end{aligned}$$

Substituting this in (21) yields:

$$G_i^{\lambda|h} = \theta_{i-1}^\top \phi_i + \sum_{j=i}^{t-1} (\gamma \lambda)^{j-i} \delta_j^i.$$

Using that $\delta_j^i = \delta_j + \theta_{j-1}^\top \phi_j - \theta_{j-1}^\top \phi_j$, it follows that

$$\sum_{j=i}^{t-1} (\gamma \lambda)^{j-i} \delta_j^i = G_i^{\lambda|h} - \theta_{i-1}^\top \phi_i - \sum_{j=i}^{t-1} (\gamma \lambda)^{j-i} (\theta_j - \theta_{j-1})^\top \phi_j.$$

As $\alpha \rightarrow 0$, we can approximate this as:

$$\begin{aligned} \sum_{j=i}^{t-1} (\gamma \lambda)^{j-i} \delta_j^i &= G_i^{\lambda|h} - \theta_{i-1}^\top \phi_i + \mathcal{O}(\alpha), \\ &= \bar{G}_i^{\lambda|h} - \theta_0^\top \phi_i + \mathcal{O}(\alpha), \end{aligned}$$

with $\bar{G}_i^{\lambda|h}$ the interim λ -return that uses θ_0 for all value evaluations. Substituting this in (20) yields:

$$\theta_t = \theta_0 + \alpha \sum_{i=0}^{t-1} (\bar{G}_i^{\lambda|h} - \theta_0^\top \phi_i + \mathcal{O}(\alpha)) \phi_i. \tag{23}$$

For the online λ -return algorithm, we can derive the following by sequential substitution of Equation (7):

$$\theta_t^i = \theta_0 + \alpha \sum_{i=0}^{t-1} (G_i^{\lambda|h} - (\theta_i^i)^\top \phi_i) \phi_i.$$

As $\alpha \rightarrow 0$, we can approximate this as:

$$\theta_t^i = \theta_0 + \alpha \sum_{i=0}^{t-1} (\bar{G}_i^{\lambda|h} - \theta_0^\top \phi_i + \mathcal{O}(\alpha)) \phi_i. \tag{24}$$

Combining (22) and (23), it follows that as $\alpha \rightarrow 0$:

$$\frac{\|\theta_t - \theta_t^i\|}{\|\theta_t - \theta_0\|} = \frac{\|(\theta_t - \theta_t^i)/\alpha\|}{\|(\theta_t - \theta_0)/\alpha\|} = \frac{\mathcal{O}(\alpha)}{C + \mathcal{O}(\alpha)},$$

with

$$C = \left\| \sum_{i=0}^{t-1} (\bar{G}_i^{\lambda|h} - \theta_0^\top \phi_i) \phi_i \right\| = \left\| \sum_{i=0}^{t-1} \Delta_i^t \right\|.$$

From the condition $\sum_{i=0}^{t-1} \Delta_i^t \neq \mathbf{0}$ it follows that $C > 0$. ■

Appendix B. Derivation Update Equations

In this subsection, we derive the update equations of true online TD(λ) directly from the online forward view, defined by equations (6) and (7) (and $\theta_0^i := \theta_{mit}$). The derivation is based on expressing θ_{t+1}^i in terms of θ_t^i .

We start by writing θ_t^i directly in terms of the initial weight vector and the interim λ -returns. First, we rewrite (7) as:

$$\theta_{k+1}^i = (\mathbf{I} - \alpha\phi_k\phi_k^\top)\theta_k^i + \alpha\phi_kG_k^{\lambda|t},$$

with \mathbf{I} the identity matrix. Now, consider θ_k^i for $k = 1$ and $k = 2$:

$$\begin{aligned}\theta_1^i &= (\mathbf{I} - \alpha\phi_0\phi_0^\top)\theta_{mit} + \alpha\phi_0G_0^{\lambda|t}, \\ \theta_2^i &= (\mathbf{I} - \alpha\phi_1\phi_1^\top)\theta_1^i + \alpha\phi_1G_1^{\lambda|t}, \\ &= (\mathbf{I} - \alpha\phi_1\phi_1^\top)(\mathbf{I} - \alpha\phi_0\phi_0^\top)\theta_{mit} + \alpha(\mathbf{I} - \alpha\phi_1\phi_1^\top)\phi_0G_0^{\lambda|t} + \alpha\phi_1G_1^{\lambda|t}.\end{aligned}$$

For general $k \leq t$, we can write:

$$\theta_k^i = \mathbf{A}_0^{k-1}\theta_{mit} + \alpha\sum_{i=0}^{k-1}\mathbf{A}_{t+1}^{k-1}\phi_iG_i^{\lambda|t},$$

where \mathbf{A}_j^i is defined as:

$$\mathbf{A}_j^i := (\mathbf{I} - \alpha\phi_j\phi_j^\top)(\mathbf{I} - \alpha\phi_{j-1}\phi_{j-1}^\top)\dots(\mathbf{I} - \alpha\phi_i\phi_i^\top), \quad \text{for } j \geq i,$$

and $\mathbf{A}_{j+1}^j := \mathbf{I}$. We are now able to express θ_t^i as:

$$\theta_t^i = \mathbf{A}_0^{t-1}\theta_{mit} + \alpha\sum_{i=0}^{t-1}\mathbf{A}_{t+1}^{t-1}\phi_iG_i^{\lambda|t}, \quad (24)$$

Because for the derivation of true online TD(λ), we only need (24) and the definition of $G_i^{\lambda|t}$, we can drop the double indices for the weight vectors and use $\theta_i := \theta_t^i$.

We now derive a compact expression for the difference $G_i^{\lambda|t+1} - G_i^{\lambda|t}$:

$$\begin{aligned}G_i^{\lambda|t+1} - G_i^{\lambda|t} &= (1 - \lambda)\sum_{n=1}^{t-i}\lambda^{n-1}G_i^{(n)} + \lambda^{t-i}G_i^{(t+1-i)}, \\ &\quad - (1 - \lambda)\sum_{n=1}^{t-i-1}\lambda^{n-1}G_i^{(n)} - \lambda^{t-i-1}G_i^{(t-i)}, \\ &= (1 - \lambda)\lambda^{t-i-1}G_i^{(t-i)} + \lambda^{t-i}G_i^{(t+1-i)} - \lambda^{t-i-1}G_i^{(t-i)}, \\ &= \lambda^{t-i}G_i^{(t+1-i)} - \lambda^{t-i}G_i^{(t-i)}, \\ &= \lambda^{t-i}\left(G_i^{(t+1-i)} - G_i^{(t-i)}\right), \\ &= \lambda^{t-i}\left(\sum_{k=1}^{t+1-i}\gamma^{k-1}R_{t+k} + \gamma^{t+1-i}\theta_t^\top\phi_{t+1} - \sum_{k=1}^{t-i}\gamma^{k-1}R_{t+k} - \gamma^{t-i}\theta_{t-1}^\top\phi_t\right), \\ &= \lambda^{t-i}\left(\gamma^{t-i}R_{t+1} + \gamma^{t+1-i}\theta_t^\top\phi_{t+1} - \gamma^{t-i}\theta_{t-1}^\top\phi_t\right), \\ &= (\lambda\gamma)^{t-i}\left(R_{t+1} + \gamma\theta_t^\top\phi_{t+1} - \theta_{t-1}^\top\phi_t\right).\end{aligned}$$

Note that the difference $G_i^{\lambda|t+1} - G_i^{\lambda|t}$ is naturally expressed using a term that looks like a TD error but with a modified time step. We call this the modified TD error, δ_t^i :

$$\delta_t^i := R_{t+1} + \gamma\theta_t^\top\phi_{t+1} - \theta_{t-1}^\top\phi_t.$$

The modified TD error relates to the regular TD error, δ_t , as follows:

$$\begin{aligned}\delta_t^i &= R_{t+1} + \gamma\theta_t^\top\phi_{t+1} - \theta_{t-1}^\top\phi_t, \\ &= R_{t+1} + \gamma\theta_t^\top\phi_{t+1} - \theta_t^\top\phi_t + \theta_t^\top\phi_t - \theta_{t-1}^\top\phi_t, \\ &= \delta_t + \theta_t^\top\phi_t - \theta_{t-1}^\top\phi_t.\end{aligned} \quad (25)$$

Using δ_t^i , the difference $G_i^{\lambda|t+1} - G_i^{\lambda|t}$ can be compactly written as:

$$G_i^{\lambda|t+1} - G_i^{\lambda|t} = (\lambda\gamma)^{t-i}\delta_t^i. \quad (26)$$

To get the update rule, θ_{t+1} has to be expressed in terms of θ_t . This is done below, using (24), (25) and (26):

$$\begin{aligned}
\theta_{t+1} &= \mathbf{A}_0^t \theta_0 + \alpha \sum_{i=0}^t \mathbf{A}_{i+1}^t \phi_i G_i^{\lambda|t+1}, \\
&= \mathbf{A}_0^t \theta_0 + \alpha \sum_{i=0}^{t-1} \mathbf{A}_{i+1}^t \phi_i G_i^{\lambda|t+1} + \alpha \phi_t G_t^{\lambda|t+1}, \\
&= \mathbf{A}_0^t \theta_0 + \alpha \sum_{i=0}^{t-1} \mathbf{A}_{i+1}^t \phi_i G_i^{\lambda|t} + \alpha \sum_{i=0}^{t-1} \mathbf{A}_{i+1}^t \phi_i (G_i^{\lambda|t+1} - G_i^{\lambda|t}) + \alpha \phi_t G_t^{\lambda|t+1}, \\
&= (\mathbf{I} - \alpha \phi_t \phi_t^\top) \left(\mathbf{A}_0^{t-1} \theta_0 + \alpha \sum_{i=0}^{t-1} \mathbf{A}_{i+1}^{t-1} \phi_i G_i^{\lambda|t} \right) \\
&\quad + \alpha \sum_{i=0}^{t-1} \mathbf{A}_{i+1}^t \phi_i (G_i^{\lambda|t+1} - G_i^{\lambda|t}) + \alpha \phi_t G_t^{\lambda|t+1}, \\
&= (\mathbf{I} - \alpha \phi_t \phi_t^\top) \theta_t + \alpha \sum_{i=0}^{t-1} \mathbf{A}_{i+1}^t \phi_i (G_i^{\lambda|t+1} - G_i^{\lambda|t}) + \alpha \phi_t G_t^{\lambda|t+1}, \\
&= (\mathbf{I} - \alpha \phi_t \phi_t^\top) \theta_t + \alpha \sum_{i=0}^{t-1} \mathbf{A}_{i+1}^t \phi_i (\gamma \lambda)^{t-i} \delta_i' + \alpha \phi_t (R_{t+1} + \gamma \theta_t^\top \phi_{t+1}), \\
&= \theta_t + \alpha \sum_{i=0}^{t-1} \mathbf{A}_{i+1}^t \phi_i (\gamma \lambda)^{t-i} \delta_i' + \alpha \phi_t (R_{t+1} + \gamma \theta_t^\top \phi_{t+1} - \theta_t^\top \phi_t), \\
&= \theta_t + \alpha \sum_{i=0}^{t-1} \mathbf{A}_{i+1}^t \phi_i (\gamma \lambda)^{t-i} \delta_i' \\
&\quad + \alpha \phi_t (R_{t+1} + \gamma \theta_t^\top \phi_{t+1} - \theta_t^\top \phi_{t+1} - \theta_t^\top \phi_t + \theta_t^\top \phi_t - \theta_t^\top \phi_t), \\
&= \theta_t + \alpha \sum_{i=0}^{t-1} \mathbf{A}_{i+1}^t \phi_i (\gamma \lambda)^{t-i} \delta_i' + \alpha \phi_t \delta_t' - \alpha (\theta_t^\top \phi_t - \theta_{t-1}^\top \phi_t) \phi_t, \\
&= \theta_t + \alpha \sum_{i=0}^t \mathbf{A}_{i+1}^t \phi_i (\gamma \lambda)^{t-i} \delta_i' - \alpha (\theta_t^\top \phi_t - \theta_{t-1}^\top \phi_t) \phi_t, \\
&= \theta_t + \alpha e_t \delta_t' - \alpha (\theta_t^\top \phi_t - \theta_{t-1}^\top \phi_t) \phi_t, \quad \text{with } e_t := \sum_{i=0}^t \mathbf{A}_{i+1}^t \phi_i (\gamma \lambda)^{t-i}, \\
&= \theta_t + \alpha e_t (\delta_t + \theta_{t-1}^\top \phi_t - \theta_{t-1}^\top \phi_t) - \alpha (\theta_t^\top \phi_t - \theta_{t-1}^\top \phi_t) \phi_t, \\
&= \theta_t + \alpha e_t \delta_t + \alpha (\theta_t^\top \phi_t - \theta_{t-1}^\top \phi_t) (e_t - \phi_t). \tag{27}
\end{aligned}$$

We now have the update rule for θ_t , in addition to an explicit definition of e_t . Next, using this explicit definition, we derive an update rule to compute e_t from e_{t-1} :

$$\begin{aligned}
e_t &= \sum_{i=0}^t \mathbf{A}_{i+1}^t \phi_i (\gamma \lambda)^{t-i}, \\
&= \sum_{i=0}^{t-1} \mathbf{A}_{i+1}^t \phi_i (\gamma \lambda)^{t-i} + \phi_t, \\
&= (\mathbf{I} - \alpha \phi_t \phi_t^\top) \gamma \lambda \sum_{i=0}^{t-1} \mathbf{A}_{i+1}^{t-1} \phi_i (\gamma \lambda)^{t-i-1} + \phi_t, \\
&= (\mathbf{I} - \alpha \phi_t \phi_t^\top) \gamma \lambda e_{t-1} + \phi_t, \\
&= \gamma \lambda e_{t-1} + \phi_t - \alpha \gamma \lambda (e_{t-1}^\top \phi_t) \phi_t. \tag{28}
\end{aligned}$$

Equations (27) and (28), together with the definition of δ_t , form the true online TD(λ) update equations.

Appendix C. Detailed Results Random MRPs

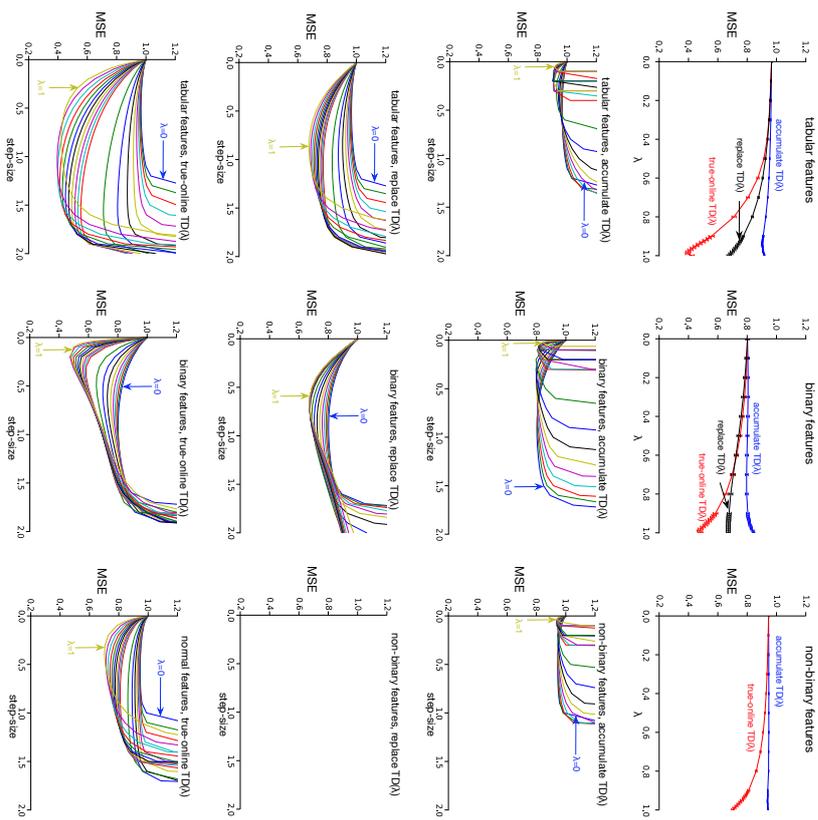


Figure 10: Results on a random MRP with $k = 10$, $b = 3$ and $\sigma = 0.1$. MSE is the mean squared error averaged over the first 100 time steps, as well as 50 runs, and normalized using the initial error. The top graphs summarize the results from the graphs below it; they show the MSE error, for each λ , at the best step-size.

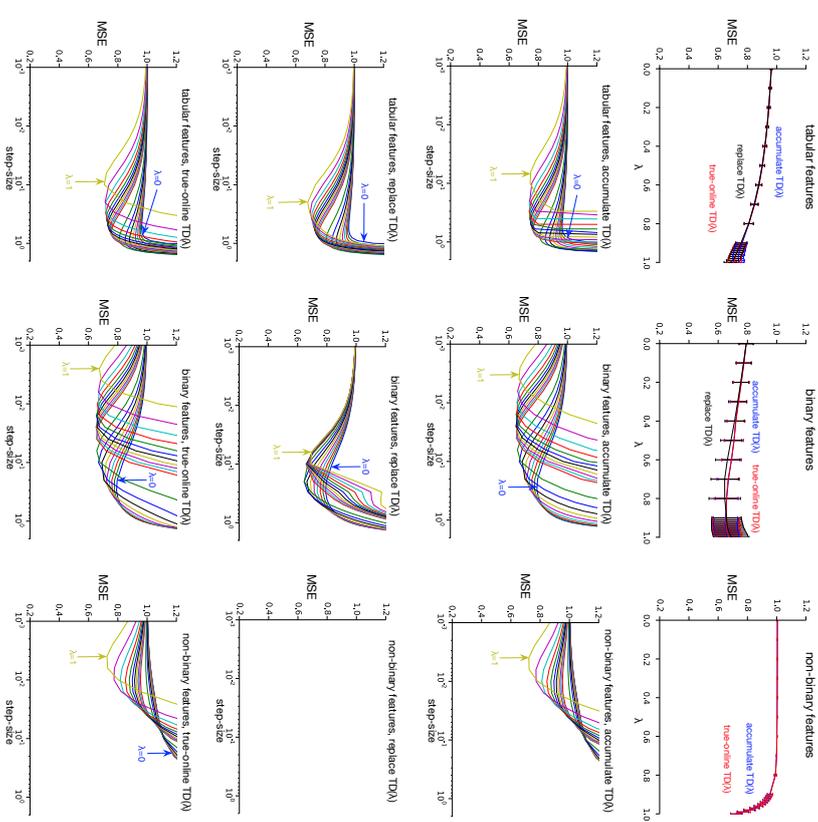


Figure 11: Results on a random MRP with $k = 100$, $b = 10$ and $\sigma = 0.1$. MSE is the mean squared error averaged over the first 1000 time steps, as well as 50 runs, and normalized using the initial error.

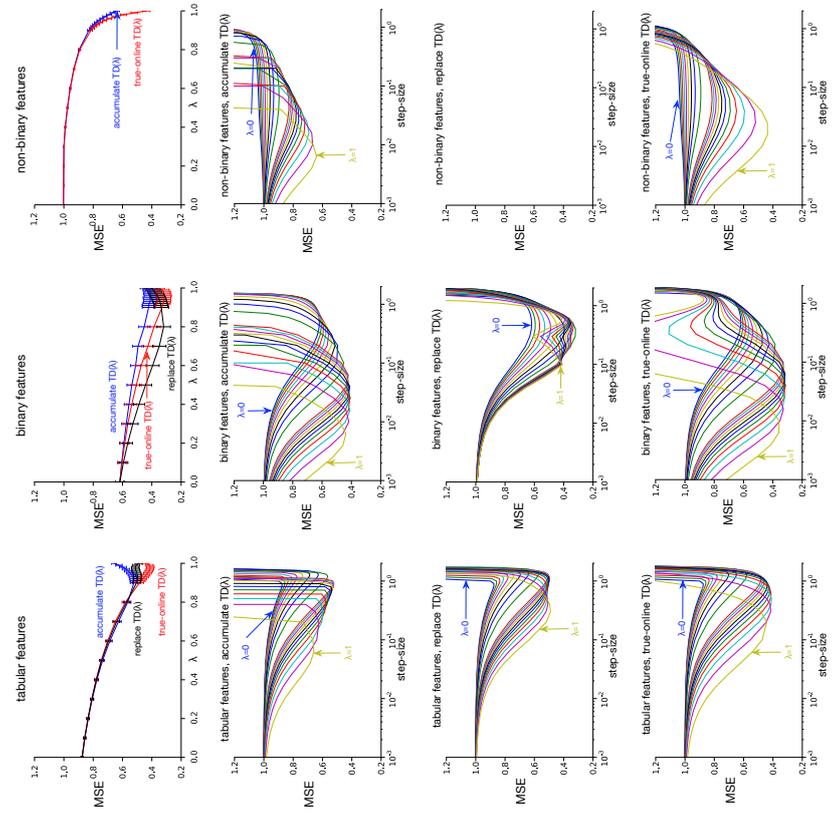


Figure 12: Results on a random MRP with $k = 100$, $b = 3$ and $\sigma = 0$. MSE is the mean squared error averaged over the first 1000 time steps, as well as 50 runs, and normalized using the initial error.

Appendix D. Detailed Results for Myoelectric Prosthetic Arm

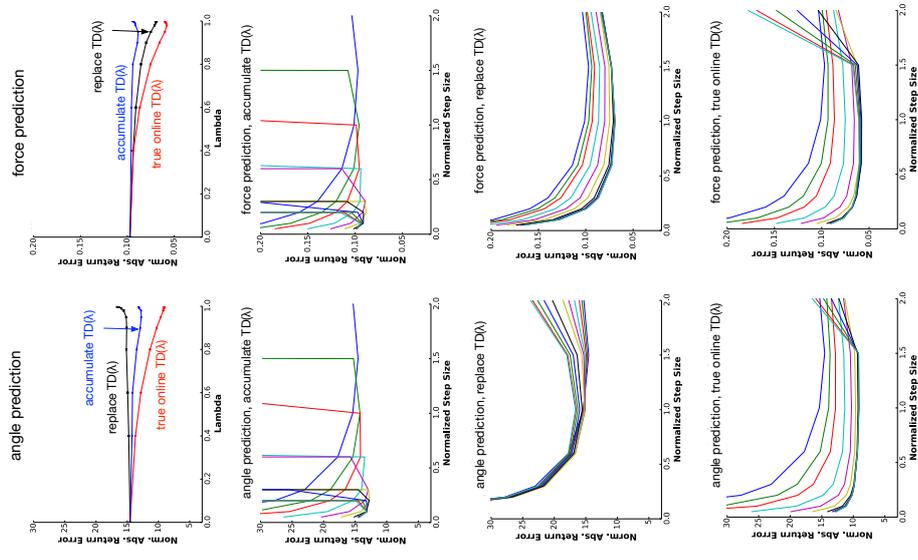


Figure 13: Results on prosthetic data from the single amputee subject described in Pilarski et al. (2013), for the prediction of servo motor angle (*left column*) and grip force (*right column*) as recorded from the amputee’s myoelectrically controlled robot arm during a grasping task.

References

- Bellemare, M. G., Naddaf, Y., Veness, J., and Bowling, M. (2013). The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279.
- Bertsekas, D. P. and Tsitsiklis, J. N. (1996). *Neuro-Dynamic Programming*. Athena Scientific.
- Dayan, P. (1992). The convergence of TD(λ) for general λ . *Machine Learning*, 8(3):341–362.
- Defazio, A. and Graepel, T. (2014). A comparison of learning algorithms on the arcade learning environment. *arXiv:1410.8620*.
- Hebert, J. S., Olson, J. L., Morhart, M. J., Dawson, M. R., Marasco, P. D., Kuitken, T. A., and Chan, K. M. (2014). Novel targeted sensory reinnervation technique to restore functional hand sensation after transhumeral amputation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(4):763–773.
- Kaelbling, L. P., Littman, M. L., and Moore, A. P. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285.
- Konidaris, G., Niekum, S., and Thomas, P. S. (2011). TD 2 : Re-evaluating complex backups in temporal difference learning. In *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, pages 2402–2410.
- Maei, H. R. (2011). *Gradient Temporal-Difference Learning Algorithms*. PhD thesis, University of Alberta, Canada.
- Mahmood, A. R. and Sutton, R. S. (2015). Off-policy learning based on weighted importance sampling with linear computational complexity. In *Proceedings of the 31th Conference on Uncertainty in Artificial Intelligence (UAI)*.
- Mnih, V., Kavukunoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrowski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., Kumaran, H. K. D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518:529–533.
- Modayil, J., White, A., and Sutton, R. S. (2014). Multi-timescale nexting in a reinforcement learning robot. *Adaptive Behavior*, 22(2):146–160.
- Parker, P., Englehart, K. B., and Hudgins, B. (2006). Myoelectric signal processing for control of powered limb prostheses. *Journal of Electromyography and Kinesiology*, 16(6):541–548.
- Pilarski, P. M., Dawson, M. R., Degris, T., Carey, J. P., Chan, K. M., Hebert, J. S., and Sutton, R. S. (2013). Adaptive artificial limbs: A real-time approach to prediction and anticipation. *IEEE Robotics & Automation Magazine*, 20(1):53–64.
- Schapire, R. E. and Warmuth, M. K. (1996). On the worst-case analysis of temporal-difference learning algorithms. *Machine Learning*, 22(1/2/3):95–121.
- Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44.
- Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge.
- Sutton, R. S., Maei, H. R., Precup, D., Bhatnagar, S., Silver, D., Szepesvári, C., and Wiewora, E. (2009a). Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pages 993–1000.
- Sutton, R. S., Maei, H. R., and Szepesvári, C. (2009b). A convergent $\mathcal{O}(n)$ algorithm for off-policy temporal-difference learning with linear function approximation. In *Proceedings of Advances in Neural Information Processing Systems 21 (NIPS)*, pages 1609–1616.
- Sutton, R. S., Mahmood, A. R., Precup, D., and van Hasselt, H. (2014). A new $Q(\lambda)$ with interim forward view and Monte Carlo equivalence. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*.
- Sutton, R. S., Modayil, J., Delp, M., Degris, T., Pilarski, P. M., White, A., and Precup, D. (2011). Horde: A scalable real-time architecture for learning knowledge from unsupervised sensorimotor interaction. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 761–768.
- Szepesvári, C. (2010). *Algorithms for Reinforcement Learning*. Morgan and Claypool.
- Thomas, P. S., Niekum, S., Theodorarous, G., and Konidaris, G. (2015). Policy evaluation using the Ω -return. In *Proceedings of Advances in Neural Information Processing Systems 28 (NIPS)*, pages 334–342.
- van Hasselt, H., Mahmood, A. R. and Sutton, R. S. (2014). Off-policy TD(λ) with a true online equivalence. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence (UAI)*.
- van Hasselt, H. and Sutton, R. S. (2015). Learning to predict independent of span. *arXiv:1508.04582*.
- van Seijen, H. H. (2016). Effective multi-step temporal-difference learning for non-linear function approximation. *arXiv:1608.05151*.
- van Seijen, H. H. and Sutton, R. S. (2014). True online TD(λ). In *Proceedings of the 31th International Conference on Machine Learning (ICML)*.
- Watkins, C. J. C. H. (1989). *Learning from Delayed Rewards*. PhD thesis, Cambridge University, Cambridge, England.

Penalized Maximum Likelihood Estimation of Multi-layered Gaussian Graphical Models

Jiahe Lin*

*Department of Statistics
University of Michigan
Ann Arbor, MI 48109, USA*

JIAHELIN@UMICH.EDU

Sumanta Basu*

*Department of Statistics
University of California, Berkeley
Berkeley, CA 94720, USA*

SUMBOSE@BERKELEY.EDU

Moulinath Banerjee

*Department of Statistics
University of Michigan
Ann Arbor, MI 48109, USA*

MOULIB@UMICH.EDU

George Michailidis†

*Department of Statistics and Computer & Information Science & Engineering
University of Florida
Gainesville, FL 32611, USA*

GMICHAIL@UFL.EDU

Editor: Jie Peng

Abstract

Analyzing multi-layered graphical models provides insight into understanding the conditional relationships among nodes within layers after adjusting for and quantifying the effects of nodes from other layers. We obtain the penalized maximum likelihood estimator for Gaussian multi-layered graphical models, based on a computational approach involving screening of variables, iterative estimation of the directed edges between layers and undirected edges within layers and a final refitting and stability selection step that provides improved performance in finite sample settings. We establish the consistency of the estimator in a high-dimensional setting. To obtain this result, we develop a strategy that leverages the biconvexity of the likelihood function to ensure convergence of the developed iterative algorithm to a stationary point, as well as careful uniform error control of the estimates over iterations. The performance of the maximum likelihood estimator is illustrated on synthetic data.

Keywords: graphical models, penalized likelihood, block coordinate descent, convergence, consistency

1. Introduction.

The estimation of directed and undirected graphs from high-dimensional data has received a lot of attention in the machine learning and statistics literature (e.g., see [Bühlmann and Van De Geer, 2011](#), and references therein), due to their importance in diverse applications including understanding of biological processes and disease mechanisms, financial systems stability and social interactions, just to name a few ([Sachs et al., 2005](#); [Wang et al., 2007](#); [Sobel, 2000](#)). In the case of undirected graphs, the edges capture conditional dependence relationships between the nodes, while for directed graphs they are used to model causal relationships ([Bühlmann and Van De Geer, 2011](#)).

However, in a number of applications the nodes can be *naturally partitioned* into sets that exhibit interactions both between them and amongst them. As an example, consider an experiment where one has collected data for both genes and metabolites for the same set of patient specimens. In this case, we have three types of interactions between genes and metabolites: regulatory interactions between the two of them and co-regulation within the gene and within the metabolic compartments. The latter two types of relationships can be expressed through undirected graphs within the sets of genes and metabolites, respectively, while the regulation of metabolites by genes corresponds to directed edges. Note that in principle there are feedback mechanisms from the metabolic compartment to the gene one, but these are difficult to detect and adequately estimate in the absence of carefully collected time course data. Another example comes from the area of financial economics, where one collects data on returns of financial assets (e.g. stocks, bonds) and also on key macroeconomic indicators (e.g. interest rate, prices indices, various measures of money supply and various unemployment indices). Once again, over short time periods there is influence from the economic variables to the returns (directed edges), while there are co-dependence relationships between the asset returns and the macroeconomic variables, respectively, that can be modeled as undirected edges.

Technically, such *layered* network structures correspond to multi-partite graphs that possess undirected edges and exhibit a directed acyclic graph structure between the layers, as depicted in [Figure 1](#), where we use directed solid edges to denote the dependencies across layers and dashed undirected edges to denote within-layer conditional dependencies.

Selected properties of such so-called *chain graphs* have been studied in the work of [Drton and Perlman \(2008\)](#), with an emphasis on two alternative Markov properties including the LWF Markov property ([Lauritzen and Wermuth, 1989](#); [Frydenberg, 1990](#)) and the AMP Markov property ([Andersson et al., 2001](#)).

While layered networks being interesting from a theoretical perspective and having significant scope for applications, their estimation has received little attention in the literature. Note that for a 2-layered structure, the directed edges can be obtained through a multivariate regression procedure, while the undirected edges in both layers through existing procedures for graphical models (for more technical details see [Section 2.2](#)). This is the strategy leveraged in the work of [Rothman et al. \(2010\)](#), where for a 2-layered network structure they proposed a multivariate regression with covariance estimation (MRCE) method for estimating the undirected edges in the second layer and the directed edges between them. A block coordinate descent algorithm was introduced to estimate the directed edges, while the popular glasso estimator ([Friedman et al., 2008](#)) was used for the undirected edges.

*. Equal Contribution

†. Corresponding Author. Post Address: 205 Griffin Floyd Hall, 1 University Ave, Gainesville, FL, 32611.

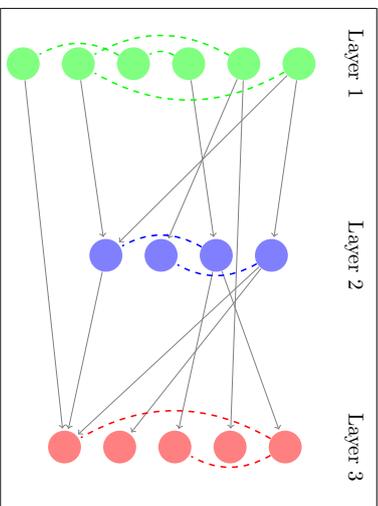


Figure 1: Diagram for a three-layered network

However, this method does not scale well according to the simulation results presented and no theoretical properties of the estimates were provided.

In follow-up work, Yin and Li (2011) used a cyclic block coordinate descent algorithm and claimed convergence to a stationary point leveraging a result in Tseng (2001) (see Proposition 2 in the Supplemental material). Unfortunately, a key assumption in Tseng (2001) –namely, that a corresponding coordinate wise optimization problem that is given by a high-dimensional lasso regression has unique minimum– fails and hence the convergence result does not go through.

In related work, Lee and Lin (2012) proposed the Plug-in Joint Weighted Lasso (PWL) and the Plug-in Joint Graphical Weighted Lasso (PWGL) estimator for estimating the same 2-layered structure, where they use a weighted version of the algorithm in Rothman et al. (2010) and also provide theoretical results for the low dimensional setting, where the number of samples exceeds the number of potential directed and undirected edges to be estimated. Finally, Cai et al. (2012) proposed a method for estimating the same 2-layered structure and provided corresponding theoretical results in the high dimensional setting. The Dantzig-type estimator (Candes and Tao, 2007) was used for the regression coefficients and the corresponding residuals were used as surrogates, for obtaining the precision matrix through the CLIME estimator (Cai et al., 2011). In another line of work (Sohn and Kim, 2012; Yuan and Zhang, 2014; McCarter and Kim, 2014), structured sparsity of directed edges was considered and the edges were estimated with a different parametrization of the objective function. We further elaborate on the connections of our work with these three papers in Section 5.

The above work assumed a Gaussian distribution for the data, in more recent work by Yang et al. (2014), the authors constructed the model under a general *mixed graphical model* framework, which allows each node-conditional distribution to belong to a potentially different univariate exponential family. In particular, with an underlying *mixed MRF* graph structure, instead of maximizing the joint likelihood, the authors proposed to esti-

mate the homogeneous and heterogeneous neighborhood for each node, by obtaining the l_1 regularized M -estimator of the node-conditional distribution parameters, using traditional approaches (e.g. Meinshausen and Bühlmann, 2006) for neighborhood estimation. However, rather than estimating directed edges directly, the directed edges are obtained from a nonlinear transformation of the estimated homogeneous and heterogeneous neighborhood, whose sparsity pattern gets compromised during the process.

In this work, we obtain the regularized maximum likelihood estimator under a sparsity assumption on both directed and undirected parameters for multi-layered Gaussian graphical models and establish its consistency properties in a high-dimensional setting. As discussed in Section 3, the problem is *not jointly convex* on the parameters, but convex on selected subsets of them. Further, it turns out that the problem is *biconvex* if we consider a recursive multi-stage estimation approach that at each stage involves only regression parameters (directed edges) from preceding layers and precision matrix parameters (undirected edges) for the *last layer considered* in that stage. Hence, we decompose the multi-layer network structure estimation into a sequence of 2-layer problems that allows us to establish the desired results. Leveraging the biconvexity of the 2-layer problem, we establish the convergence of the iterates to the maximum-likelihood estimator, which under certain regularity conditions is arbitrarily close to the true parameters. The theoretical guarantees provided require a *uniform control* of the precision of the regression and precision matrix parameters, which poses a number of theoretical challenges resolved in Section 3.

In summary, despite the lack of overall convexity, we are able to provide theoretical guarantees for the MLE in a high dimensional setting. We believe that the proposed strategy is generally applicable to other non-convex statistical estimation problems that can be decomposed to two biconvex problems. Further, to enhance the numerical performance of the MLE in finite (and small) sample settings, we introduce a screening step that selects active nodes for the iterative algorithm used and that leverages recent developments in the high-dimensional regression literature (e.g., Van de Geer et al., 2014; Javanmard and Montanari, 2014; Zhang and Zhang, 2014). We also post-process the final MLE estimate through a stability selection procedure. As mentioned above, the screening and stability selection steps are beneficial to the performance of the MLE in finite samples and hence recommended for similarly structured problems.

The remainder of the paper is organized as follows. In Section 2, we introduce the proposed methodology, with an emphasis on how the multi-layered network estimation problem is decomposed into a sequence of two-layered network estimation problem(s). In Section 3, we provide theoretical guarantees for the estimation procedure posited. In particular, we show consistency of the estimates and convergence of the algorithm, under a number of common assumptions in high-dimensional settings. In Section 4, we show the performance of the proposed algorithm with simulation results under different simulation settings, and introduce several acceleration techniques which speed up the convergence of the algorithm and reduce the computing time in practical settings.

2. Problem Formulation.

Consider an M -layered Gaussian graphical model. Suppose there are p_m nodes in Layer m , denoted by

$$\mathbf{X}^m = (X_1^m, \dots, X_{p_m}^m)', \quad \text{for } m = 1, \dots, M.$$

The structure of the model is given as follows:

- Layer 1. $\mathbf{X}^1 = (X_1^1, \dots, X_{p_1}^1)' \sim \mathcal{N}(0, \Sigma^1)$.
- Layer 2. For $j = 1, \dots, p_2$: $X_j^2 = (B_j^{12})' \mathbf{X}^1 + \epsilon_j^2$, with $B_j^{12} \in \mathbb{R}^{p_1}$, and $\epsilon^2 = (\epsilon_1^2, \dots, \epsilon_{p_2}^2)' \sim \mathcal{N}(0, \Sigma^2)$.
- ⋮
- Layer M . For $j = 1, 2, \dots, p_M$:

$$X_j^M = \sum_{m=1}^{M-1} \{(B_j^{mM})' \mathbf{X}^m\} + \epsilon_j^M, \quad \text{where } B_j^{mM} \in \mathbb{R}^{p_m} \text{ for } m = 1, \dots, M-1,$$

$$\text{and } \epsilon^M = (\epsilon_1^M, \dots, \epsilon_{p_M}^M)' \sim \mathcal{N}(0, \Sigma^M).$$

The parameters of interest are *all directed edges* that encode the dependencies across layers, that is,

$$B^{st} := [B_1^{st} \ \dots \ B_{p_s}^{st}], \quad \text{for } 1 \leq s < t \leq M,$$

and *all undirected edges* that encode the conditional dependencies within layers after adjusting for the effects from directed edges, that is:

$$\Theta^m := (\Sigma^m)^{-1}, \quad \text{for } m = 1, \dots, M.$$

It is assumed that B^{st} and Θ^m are *sparse* for all $1, \dots, M$ and $1 \leq s < t \leq M$.

Given centered data for all M layers, denoted by $X^m = [X_1^m, \dots, X_{p_m}^m] \in \mathbb{R}^{n \times p_m}$ for all $m = 1, \dots, M$, we aim to obtain the MLE for all B^{st} , $1 \leq s < t \leq M$ and all Θ^m , $m = 1, \dots, M$ parameters. Henceforth, we use \mathbf{X}^m to denote random vectors, and X_j^m to denote the j th column in the data matrix $X_n^{m \times p_m}$ whenever there is no ambiguity.

Through Markov factorization (Lauritzen, 1996), the full log-likelihood function can be decomposed as

$$\begin{aligned} \ell(X^m, B^{st}, \Theta^m, 1 \leq s < t \leq M, 1 \leq m \leq M) &= \ell(X^M | X^{M-1}, \dots, X^1, B^{1M}, \dots, B^{M-1,M}, \Theta^M) \\ &\quad + \ell(X^{M-1} | X^{M-2}, \dots, X^1, B^{1,M-1}, \dots, B^{M-2,M-1}, \Theta^{M-1}) \\ &\quad + \dots + \ell(X^2 | X^1, B^{12}, \Theta^2) + \ell(X^1; \Theta^1) \\ &= \ell(X^1; \Theta^1) + \sum_{m=2}^M \ell(X^m | X^1, \dots, X^{m-1}, \Theta^{m-1}; B^{1,m}, \dots, B^{m-1,m}, \Theta^m). \end{aligned}$$

Note that the summands share no common parameters, which enables us to maximize the likelihood with respect to individual parameters in the M terms separately. More importantly, by conditioning Layer m nodes on nodes in its previous $(m-1)$ layers, we can treat Layer m nodes as the “response” layer, and all nodes in the previous $(m-1)$ layer

combined as a super “parent” layer. If we ignore the structure within the bottom layer (X^1) for the moment, the M -layered network can be viewed as $(M-1)$ two-layered networks, each comprising a response layer and a parent layer. Thus, the network structure in Figure 1 can be viewed as a 2 two-layered network: for the first network, Layer 3 is the response layer, while Layers 1 and 2 combined form the “parent” layer; for the second network, Layer 2 is the response layer, and Layer 1 is the “parent” layer. Therefore, the problem for estimating all $\binom{M}{2}$ coefficient matrices and M precision matrices can be translated into estimating $(M-1)$ two-layered network structures with directed edges from the parent layer to the response layer, and undirected edges within the response layer, and finally estimating the undirected edges within the bottom layer separately.

Since all estimation problems boil down to estimating the structure of a 2-layered network, we focus the technical discussion on introducing our proposed methodology for a 2-layered network setting.¹ The theoretical results obtained extend in a straightforward manner to an M -layered Gaussian graphical model.

Remark 1. For the M -layer network structure, we impose certain identifiability-type condition on the largest “parent” layer (encompassing $M-1$ layers), so that the directed edges of the entire network are estimable. The imposed condition translates into a minimum eigenvalue-type condition on the population precision matrix within layers, and conditions on the magnitude of dependencies across layers. Intuitively, consider a three-layered network: if \mathbf{X}^1 and \mathbf{X}^2 are highly correlated, then the proposed (as well as any other) method will exhibit difficulties in distinguishing the effect of \mathbf{X}^1 on \mathbf{X}^3 from that of \mathbf{X}^2 on \mathbf{X}^3 . The (group) identifiability-type condition is thus imposed to obviate such circumstances. An in-depth discussion on this issue is provided in Section 3.4.

2.1 A Two-layered Network Setup.

Consider a two-layered Gaussian graphical model with p_1 nodes in the first layer, denoted by $\mathbf{X} = (X_1, \dots, X_{p_1})'$, and p_2 nodes in the second layers, denoted by $\mathbf{Y} = (Y_1, \dots, Y_{p_2})'$. The model is defined as

- $\mathbf{X} = (X_1, \dots, X_{p_1})' \sim \mathcal{N}(0, \Sigma_X)$.
- For $j = 1, 2, \dots, p_2$: $Y_j = B_j' \mathbf{X} + \epsilon_j$, $B_j \in \mathbb{R}^{p_1}$ and $\epsilon = (\epsilon_1, \dots, \epsilon_{p_2})' \sim \mathcal{N}(0, \Sigma_\epsilon)$.

The parameters of interest are: $\Theta_X := \Sigma_X^{-1}$, $\Theta_\epsilon := \Sigma_\epsilon^{-1}$ and $B = [B_1, \dots, B_{p_2}]$. As with most estimation problems in the high dimensional setting, we assume these parameters to be sparse.

Now given data $X = [X_1, \dots, X_{p_1}] \in \mathbb{R}^{n \times p_1}$ and $Y = [Y_1, \dots, Y_{p_2}] \in \mathbb{R}^{n \times p_2}$, both centered, we would like to use the penalized maximum likelihood approach to obtain estimates for Θ_X , Θ_ϵ and B . Throughout this paper, we use X , Y and E to denote the size- n realizations of the random vectors \mathbf{X} , \mathbf{Y} and ϵ , respectively. Also, with a slight abuse of notation, we use X_i , $i = 1, 2, \dots, p_1$ and Y_j , $j = 1, 2, \dots, p_2$ to denote the columns of the data matrix X and Y , respectively, whenever there is no ambiguity.

1. In Appendix D we give a detail example on how our proposed method works under a 3-layered network setting.

The full log-likelihood can be written as

$$\ell(X, Y; B, \Theta_\epsilon, \Theta_X) = \ell(Y|X; \Theta_\epsilon, B) + \ell(X; \Theta_X) \quad (1)$$

Note that the first term only involves Θ_ϵ and B , and the second term only involves Θ_X . Hence, (1) can be maximized by maximizing $\ell(Y|X)$ w.r.t. (Θ_ϵ, B) , and maximizing $\ell(X)$ w.r.t. Θ_X , respectively. $\widehat{\Theta}_X$ can be obtained using traditional methods for estimating undirected graphs, e.g., the Graphical Lasso (Friedman et al., 2008) or the Node-wise Regression procedure (Meinshausen and Bühlmann, 2006). Therefore, the rest of this paper will mainly focus on obtaining estimates for Θ_ϵ and B . In the next subsection, we introduce our estimation procedure for obtaining the MLE for Θ_ϵ and B .

Remark 2. Our proposed method is targeted towards maximizing $\ell(Y|X; \Theta_\epsilon, B)$ (with proper penalization) in (1) only, which gives the estimates for across-layers dependencies between the response layer and the parent layer, as well as estimates for the conditional dependencies within the response layer each time we solve a 2-layered network estimation problem. For an M -layered estimation problem, the maximization regarding $\ell(X; \Theta_X)$ occurs only when we are estimating the within-layer conditional dependencies for the bottom layer.

2.2 Estimation Algorithm.

The conditional likelihood for response Y given X can be written as

$$L(Y|X) = \left(\frac{1}{\sqrt{2\pi}}\right)^{np_2} |\Sigma_\epsilon \otimes I_n|^{-1/2} \exp\left\{-\frac{1}{2}(\mathcal{Y} - \mathcal{X}\beta)^\top (\Sigma_\epsilon \otimes I_n)^{-1}(\mathcal{Y} - \mathcal{X}\beta)\right\},$$

where $\mathcal{Y} = \text{vec}(Y_1, \dots, Y_{p_2})$, $\mathcal{X} = I_{p_2} \otimes X$ and $\beta = \text{vec}(B_1, \dots, B_{p_2})$. After writing out the Kronecker product, the log-likelihood can be written as

$$\ell(Y|X) = \text{constant} + \frac{n}{2} \log \det \Theta_\epsilon - \frac{1}{2} \sum_{j=1}^{p_2} \sum_{i=1}^{p_2} \sigma_\epsilon^{ij} (Y_i - XB_j)^\top (Y_j - XB_j).$$

Here, σ_ϵ^{ij} denotes the ij -th entry of Θ_ϵ . With ℓ_1 penalization which induces sparsity, we formulate the following optimization problem using penalized log-likelihood, which was initially proposed in Rothman et al. (2010), and has also been examined in Lee and Lin (2012):

$$\min_{\substack{B \in \mathbb{R}^{p_1 \times p_2} \\ \Theta_\epsilon \in \mathbb{S}_{p_2}^{++}}} \left\{ \frac{1}{n} \sum_{j=1}^{p_2} \sum_{i=1}^{p_2} \sigma_\epsilon^{ij} (Y_i - XB_j)^\top (Y_j - XB_j) - \log \det \Theta_\epsilon + \lambda_n \sum_{j=1}^{p_2} \|B_j\|_1 + \rho_n \|\Theta_\epsilon\|_{1,\text{off}} \right\}, \quad (2)$$

and the first term in (2) can be equivalently written as

$$\text{tr} \left\{ \frac{1}{n} \begin{bmatrix} (Y_1 - XB_1)^\top \\ \vdots \\ (Y_{p_2} - XB_{p_2})^\top \end{bmatrix} \begin{bmatrix} (Y_1 - XB_1) & \cdots & (Y_{p_2} - XB_{p_2}) \end{bmatrix} \Theta_\epsilon \right\} =: \text{tr}(\mathcal{S}\Theta_\epsilon).$$

where \mathcal{S} is defined as the sample covariance matrix of $E \equiv Y - XB$. This gives rise to the following optimization problem:

$$\min_{\substack{B \in \mathbb{R}^{p_1 \times p_2} \\ \Theta_\epsilon \in \mathbb{S}_{p_2}^{++}}} \left\{ \text{tr}(\mathcal{S}\Theta_\epsilon) - \log \det \Theta_\epsilon + \lambda_n \sum_{j=1}^{p_2} \|B_j\|_1 + \rho_n \|\Theta_\epsilon\|_{1,\text{off}} \right\} =: f(B, \Theta_\epsilon), \quad (3)$$

where $\|\Theta\|_{1,\text{off}}$ is the absolute sum of the off-diagonal entries in Θ , λ_n and ρ_n are both positive tuning parameters.

Note that the objective function (3) is *not jointly convex* in (B, Θ_ϵ) , but only convex in B for fixed Θ_ϵ and in Θ_ϵ for fixed B ; hence, it is bi-convex, which in turn implies that the proposed algorithm may fail to converge to the global optimum, especially in settings where $p_1 > n$, as pointed out by Lee and Lin (2012). As is the case with most non-convex problems, good initial parameters are beneficial for fast convergence of the algorithm, a fact supported by our numerical work on the present problem. Further, a good initialization is critical in establishing convergence of the algorithm for this problem (see Section 3.1). To that end, we introduce a *screening step* for obtaining a good initial estimate for B . The theoretical justification for employing the screening step is provided in Section 3.3.

An outline of the computational procedure is presented in Algorithm 1, while the details of each step involved are discussed next.

Screening. For each variable Y_j , $j = 1, \dots, p_2$ in the response layer, regress Y_j on X via the de-biased Lasso procedure proposed by Javanmard and Montanari (2014). The output consists of the p -value(s) for each predictor in each regression, denoted by P_j , with $P_j \in [0, 1]^{p_1}$. To control the family-wise error rate of the estimates, we do a Bonferroni correction at level α : define $\alpha^* = \alpha/p_1 p_2$ and set $B_{j,k} = 0$ if the p -value obtained for the k 'th predictor in the j 'th regression $P_{j,k}$ exceeds α^* . Further, let

$$B_j = \{B_j \in \mathbb{R}^{p_1} : B_{j,k} = 0 \text{ if } k \in \widehat{S}_j^c\} \subseteq \mathbb{R}^{p_1}, \quad (4)$$

where \widehat{S}_j is the collection of indices for those predictors deemed ‘‘active’’ for response Y_j :

$$\widehat{S}_j = \{k : P_{j,k} < \alpha^*\}, \quad \text{for } j = 1, \dots, p_2.$$

Therefore, subsequent estimation of the elements of B will be restricted to $B_1 \times \dots \times B_{p_2}$.

Alternating Search. In this step, we utilize the bi-convexity of the problem and estimate B and Θ_ϵ by minimizing in an iterative fashion the objective function with respect to (w.r.t.) one set of parameters, while holding the other set fixed within each iteration.

As with most iterative algorithms, we need an initializer; for $\widehat{B}^{(0)}$ it corresponds to a Lasso/Ridge regression estimate with a small penalty, while for Θ_ϵ we use the Graphical Lasso procedure applied to the residuals obtained from the first stage regression. That is, for each $j = 1, \dots, p_2$,

$$\widehat{B}_j^{(0)} = \arg \min_{B_j \in \mathcal{S}_j} \left\{ \|Y_j - XB_j\|_2^2 + \lambda_n^0 \|B_j\|_1 \right\}, \quad (5)$$

Algorithm 1: Computational procedure for estimating B and Θ_ϵ

Input : Data from the parent layer X and the response layer Y .

- 1 **Screening:**
 - 2 for $j = 1, \dots, p_2$ **do**
 - regress Y_j on X using the de-biased Lasso procedure in [Javanmard and Montanari \(2014\)](#) and obtain the corresponding vector of p -values P_j ;
 - end**
 - obtain adjusted p -values \tilde{P}_j by applying Bonferroni correction to $\text{vec}(P_1, \dots, P_j)$;
 - determine the support set \mathcal{B}_j for each regression using [\(4\)](#).
 - 3 **Initialization:**
 - 4 Initialize column $j = 1, \dots, p_2$ of $\tilde{B}^{(0)}$ by solving [\(5\)](#).
 - Initialize $\hat{\Theta}_\epsilon^{(0)}$ by solving [\(9\)](#) using the graphical lasso ([Friedman et al., 2008](#)).
 - 5 **while** $|f(\tilde{B}^{(k)}, \hat{\Theta}_\epsilon^{(k)}) - f(\tilde{B}^{(k+1)}, \hat{\Theta}_\epsilon^{(k+1)})| \geq \epsilon$ **do**
 - 6 | update \tilde{B} with [\(6\)](#);
 - 7 | update $\hat{\Theta}_\epsilon$ with [\(8\)](#);
 - 8 **end**
 - 9 **Refitting B and Θ_ϵ :**
 - for $j = 1, \dots, p_2$ **do**
 - Obtain the refitted \tilde{B}_j using [\(9\)](#);
 - **end**
 - re-estimate $\tilde{\Theta}_\epsilon$ using [\(10\)](#) with W coming from stability selection.
 - Output:** Final Estimates \tilde{B} and $\tilde{\Theta}_\epsilon$.
-

where λ_n^0 is some small tuning parameter for initialization, and set $\tilde{E}_j^{(0)} := Y_j - X\tilde{B}_j^{(0)}$. An initial estimate for $\hat{\Theta}_\epsilon$ is then given by solving for the following optimization problem with the graphical lasso ([Friedman et al., 2008](#)) procedure:

$$\hat{\Theta}_\epsilon^{(0)} = \underset{\Theta_\epsilon \in \mathbb{S}_{++}^{p_2 \times p_2}}{\text{argmin}} \left\{ \log \det \Theta_\epsilon - \text{tr}(\tilde{S}^{(0)} \Theta_\epsilon) + \rho_n \|\Theta_\epsilon\|_{1,\text{off}} \right\},$$

where $\tilde{S}^{(0)}$ is the sample covariance matrix based on $(\tilde{E}_1^{(0)}, \dots, \tilde{E}_{p_2}^{(0)})$.

Next, we use an alternating block coordinate descent algorithm with ℓ_1 penalization to reach a stationary point of the objective function [\(3\)](#).

- Update B as

$$\tilde{B}^{(k+1)} = \underset{B \in \mathcal{B}_1 \times \dots \times \mathcal{B}_{p_2}}{\text{argmin}} \left\{ \frac{1}{n} \sum_{i=1}^{p_2} \sum_{j=1}^{p_2} (\hat{\sigma}_\epsilon^{ij})^{(k)} (Y_i - XB_i)^\top (Y_j - XB_j) + \lambda_n \sum_{j=1}^{p_2} \|B_j\|_1 \right\}, \quad (6)$$

which can be obtained by cyclic coordinate descent w.r.t each column B_j of B , that is, update each column B_j by:

$$\tilde{B}_j^{(k+1)} = \underset{B_j \in \mathcal{B}_j}{\text{argmin}} \left\{ \frac{(\hat{\sigma}_\epsilon^{jj})^{(k)}}{n} \|Y_j + r_j^{(k+1)} - XB_j\|_2^2 + \lambda_n \|B_j\|_1 \right\}, \quad (7)$$

where

$$r_j^{(k+1)} = \frac{1}{(\hat{\sigma}_\epsilon^{jj})^{(k)}} \left[\sum_{i=1}^{j-1} (\hat{\sigma}_\epsilon^{ij})^{(k)} (Y_i - X\tilde{B}_i^{(k+1)}) + \sum_{i=j+1}^{p_2} (\hat{\sigma}_\epsilon^{ij})^{(k)} (Y_i - X\tilde{B}_i^{(k)}) \right],$$

and iterate over all columns until convergence. Here, we use k to index the outer iteration while minimizing w.r.t. B or Θ_ϵ , and use t to index the inner iteration while cyclically minimizing w.r.t. each column of B .

— Update Θ_ϵ as

$$\hat{\Theta}_\epsilon^{(k+1)} = \underset{\Theta_\epsilon \in \mathbb{S}_{++}^{p_2 \times p_2}}{\text{argmin}} \left\{ \log \det \Theta_\epsilon - \text{tr}(\tilde{S}^{(k+1)} \Theta_\epsilon) + \rho_n \|\Theta_\epsilon\|_{1,\text{off}} \right\}, \quad (8)$$

where $\tilde{S}^{(k+1)}$ is the sample covariance matrix based on $\tilde{E}_j^{(k+1)} = Y_j - X\tilde{B}_j^{(k+1)}$, $j = 1, \dots, p_2$.

Refitting and Stabilizing. As noted in the introduction, this step is beneficial in applications, especially when one deals with large scale multi-layer networks and relatively smaller sample sizes. Denote the solution obtained by the above iterative procedure by B^∞ and Θ_ϵ^∞ . For each $j = 1, \dots, p_2$, set $\tilde{B}_j := \{B_j : B_{j,i} = 0 \text{ if } B_{j,i}^\infty = 0, B_j \in \mathbb{R}^{p_1}\}$ and the final estimate for B_j is given by ordinary least squares:

$$\tilde{B}_j = \underset{B_j \in \mathcal{B}_j}{\text{argmin}} \|Y_j - XB_j\|^2. \quad (9)$$

For Θ_ϵ , we obtain the final estimate by a combination of stability selection ([Meinshausen and Bühlmann, 2010](#)) and graphical lasso ([Friedman et al., 2008](#)). That is, after obtaining the refitted residuals $\tilde{E}_j := Y_j - X\tilde{B}_j$, $j = 1, \dots, p_2$, based on the stability selection procedure with the graphical lasso, we obtain the stability path, or probability matrix W for each edge, which records the proportion of each edge being selected based on bootstrapped samples of \tilde{E}_j 's. Then, using this probability matrix W as a weight matrix, we obtain the final estimate of $\tilde{\Theta}_\epsilon$ as follow:

$$\tilde{\Theta}_\epsilon = \underset{\Theta_\epsilon \in \mathbb{S}_{++}^{p_2 \times p_2}}{\text{argmin}} \left\{ \log \det \Theta_\epsilon - \text{tr}(\tilde{S} \Theta_\epsilon) + \tilde{\rho}_n \|(1 - W) * \Theta_\epsilon\|_{1,\text{off}} \right\}, \quad (10)$$

where we use $*$ to denote the element-wise product of two matrices, and \tilde{S} is the sample covariance matrix based on the refitted residuals \tilde{E} . Again, [\(10\)](#) can be solved by the graphical lasso procedure ([Friedman et al., 2008](#)), with $\tilde{\rho}_n$ properly chosen.

2.3 Tuning Parameter Selection.

To select the tuning parameters (λ_n, ρ_n) , we use the Bayesian Information Criterion (BIC), which is the summation of a goodness-of-fit term (log-likelihood) and a penalty term. The explicit form of BIC (as a function of B and Θ_ϵ) in our setting is given by

$$\text{BIC}(B, \Theta_\epsilon) = -\log \det \Theta_\epsilon + \text{tr}(S\Theta_\epsilon) + \frac{\log n}{n} \left(\frac{\|\Theta_\epsilon\|_0 - p_2}{2} + \|B\|_0 \right)$$

where

$$S := \frac{1}{n} \begin{bmatrix} (Y_1 - XB_1)^\top \\ \vdots \\ (Y_{p_2} - XB_{p_2})^\top \end{bmatrix} [(Y_1 - XB_1) \ \cdots \ (Y_{p_2} - XB_{p_2})],$$

and $\|\Theta_\epsilon\|_0$ is the total number of nonzero entries in Θ_ϵ . Here we penalize the non-zero elements in the upper-triangular part of Θ_ϵ and the non-zero ones in B . We choose the combination (λ_n^*, ρ_n^*) over a grid of (λ, ρ) values, and (λ_n^*, ρ_n^*) should minimize the BIC evaluated at $(\mathcal{B}^\infty, \Theta_\epsilon^\infty)$.

3. Theoretical Results.

In this section, we establish a number of theoretical results for the proposed iterative algorithm. We focus the presentation on the two-layer structure, since as explained in the previous section the multi-layer estimation problem decomposes to a series of two-layer ones. As mentioned in the introduction, one key challenge for establishing the theoretical results comes from the fact that the objective function (3) is not jointly convex in B and Θ_ϵ . Consequently, if we simply used properties of block-coordinate descent algorithms, we would not be able to provide the necessary theoretical guarantees for the estimates we obtain. On the other hand, the biconvex nature of the objective function allows us to establish convergence of the alternating algorithm to a stationary point, provided it is initialized from a point close enough to the true parameters. This can be accomplished using a Lasso-based initializer for B and Θ_ϵ as previously discussed. The details of algorithmic convergence are presented in Section 3.1.

Another technical challenge is that each update in the alternating search step relies on estimated quantities—namely the regression and precision matrix parameters—rather than the raw data, whose estimation precision needs to be controlled *uniformly* across all iterations. The details of establishing consistency of the estimates for both fixed and random realizations are given in Section 3.2.

Next, we outline the structure of this section. In Section 3.1 Theorem 1, we show that for any fixed set of realization of X and E ,² the iterative algorithm is guaranteed to converge to a stationary point if estimates for all iterations lie in a compact ball around the true value of the parameters. In Section 3.2, we show in Theorem 4 that for any random X and E , with high probability, the estimates for all iterations lie in a compact ball around the true value of the parameters. Then in Section 3.3, we show that asymptotically with

² We actually observe X and Y , which is given by a corresponding set of realization in X and E based on the model.

$\log(p_1 p_2)/n \rightarrow 0$, while keeping the family-wise type I error under some pre-specified level, the screening step correctly identifies the true support set for each of the regressions, based upon which the iterative algorithm is provided with an initializer that is close to the true value of the parameters. Finally in Section 3.4, we provide sufficient conditions for both directed and undirected edges to be identifiable (estimable) for multi-layered network.

To aid the readability of the main results, we only present statements of theorems and propositions, while all proofs are relegated to the Appendix (Section A and B).

Throughout this section, to distinguish the estimates from the true values, we use B^* and Θ_ϵ^* to denote the true values.

3.1 Convergence of the Iterative Algorithm.

In this subsection, we prove that the proposed block relaxation algorithm converges to a stationary point for any fixed set of data, provided that the estimates for all iterations lie in a compact ball around the true value of the parameters. This requirement is shown to be satisfied with high probability in the next subsection 3.2.

Decompose the optimization problem in (3) as follows:

$$\min_{\substack{B \in \mathbb{R}^{p_1 \times p_2} \\ \Theta_\epsilon \in \mathbb{S}_{++}^{p_2 \times p_2}}} f(B, \Theta_\epsilon) \equiv f_0(B, \Theta_\epsilon) + f_1(B) + f_2(\Theta_\epsilon),$$

where

$$f_0(B, \Theta_\epsilon) = \frac{1}{n} \sum_{j=1}^{p_2} \sum_{i=1}^{p_2} \sigma_{ij}^2 (Y_i - XB_j)(Y_j - XB_j) - \log \det \Theta_\epsilon = \text{tr}(S\Theta_\epsilon) - \log \det \Theta_\epsilon,$$

$$f_1(B) = \lambda_n \|B\|_1, \quad f_2(\Theta_\epsilon) = \rho_n \|\Theta_\epsilon\|_{1,\text{off}}.$$

and $\mathbb{S}_{++}^{p_2 \times p_2}$ is the collection of $p_2 \times p_2$ symmetric positive definite matrices. Further, denote the limit point (if there is any) of $\{\widehat{B}^{(k)}\}$ and $\{\widehat{\Theta}_\epsilon^{(k)}\}$ by $B^\infty = \lim_{k \rightarrow \infty} \widehat{B}^{(k)}$ and $\Theta_\epsilon^\infty = \lim_{k \rightarrow \infty} \widehat{\Theta}_\epsilon^{(k)}$, respectively.

Definition 1 (stationary point (Tseung, 2001) pp.479). Define z to be a stationary point of f if $z \in \text{dom}(f)$ and $f'(z; d) \geq 0, \forall$ direction $d = (d_1, \dots, d_N)$ where d_t is the t^{th} coordinate block.

Definition 2 (Regularity (Tseung, 2001) pp.479). f is regular at $z \in \text{dom}(f)$ if $f'(z; d) \geq 0$ for all $d = (d_1, \dots, d_N)$ such that

$$f'(z; (0, \dots, d_t, \dots, 0)) \geq 0, \quad t = 1, 2, \dots, N.$$

Definition 3 (Coordinate-wise minimum). Define $(B^\infty, \Theta_\epsilon^\infty)$ to be a coordinate-wise minimum if

$$\begin{aligned} f(B^\infty, \Theta_\epsilon^\infty) &\geq f(B^\infty, \Theta_\epsilon^\infty), & \forall \Theta_\epsilon \in \mathbb{S}_{++}^{p_2 \times p_2}, \\ f(B, \Theta_\epsilon^\infty) &\geq f(B^\infty, \Theta_\epsilon^\infty), & \forall B \in \mathbb{R}^{p_1 \times p_2}. \end{aligned}$$

Note for our iterative algorithm, we only have two blocks, hence with the above notation, $N = 2$.

Remark 3. Tseng (2001) proved that if the level set $\{x : f(x) \leq f(x^0)\}$ is compact and f satisfies certain conditions (Tseng, 2001, see Theorem 4.1 (a), (b) and (c) for details), the limit point given by the general block-coordinate descent algorithm (with $N \geq 2$ blocks) is a stationary point of f . However, the conditions given in Theorem 4.1 (a), (b) and (c) are not satisfied for the objective function at hand. Hence, for the problem under consideration, a different strategy is needed to prove convergence of the 2-block alternating algorithm to a stationary point, and the resulting statements hold true for all problems that use a 2-block coordinate descent algorithm.

Since $\text{dom}(f_0)$ is open and f_0 is Gâteaux-differentiable on the $\text{dom}(f_0)$, by Tseng (2001) Lemma 3.1, f is regular in the $\text{dom}(f)$. From the discussion on Page 479 of (Tseng, 2001), we then have:

Fact 1: Every coordinate-wise minimum is a stationary point of f .

The following theorem shows that any limit point $(B^\infty, \Theta_\epsilon^\infty)$ of the iterative algorithm described in Section 2.2 is a stationary point of f , as long as all the iterates are within a closed ball around the truth.

Theorem 1 (Convergence for fixed design). *Suppose for any fixed realization of X and E , the estimates $\{(\hat{B}^{(k)}, \hat{\Theta}_\epsilon^{(k)})\}_{k=1}^\infty$ obtained by implementing the alternating search step satisfy the following bound for some $\bar{R} > 0$ that only depends on p_1, p_2 and n :*

$$\|(\hat{B}^{(k)}, \hat{\Theta}_\epsilon^{(k)}) - (B^*, \Theta^*)\|_F \leq R(p_1, p_2, n), \quad \forall k \geq 1.$$

Then any limit point $(B^\infty, \Theta_\epsilon^\infty)$ of the iterative algorithm is a stationary point of f .

Remark 4. Recall that in classical parametric statistics, MLE-type asymptotics are derived after establishing that with probability tending to 1 as the sample size n goes to infinity, the likelihood equation has a sequence of roots (hence stationary points of the likelihood function) that converges in probability to the true value. Any such sequence of roots is shown to be asymptotically normal and efficient. Note that such (a sequence of) roots may not be global maximizers since parametric likelihoods are not globally log-concave (see Chapter 6 Lehmann and Casella, 1998). Here we show that the $(B^\infty, \Theta_\epsilon^\infty)$ obtained by the iterative algorithm is a stationary point which satisfies the first-order condition for being a maximizer of the penalized log-likelihood function (which is just the negative of the penalized least-squares function). Moreover, if we let n go to infinity, $(B^\infty, \Theta_\epsilon^\infty)$ converges to the true value in probability (shown in Theorem 4), and therefore behaves the same as the sequence of roots in the classical parametric problem alluded to above. Thus, while $(B^\infty, \Theta_\epsilon^\infty)$ may not be the global maximizer, it can, nevertheless, to all intents and purposes, be deemed as the MLE.

Remark 5. The above convergence result is based upon solving the optimization problem on the “entire” space, that is, we don’t restrict B to live in any subspace. However, when actually implementing the proposed computational procedure, the optimization of the B coordinate is restricted to $\mathcal{B}_1 \times \dots \times \mathcal{B}_{p_2}$ (as defined in eqn.4). It should be noted that the

same convergence property still holds, since for all $k \geq 1$, the following bound holds, for some $R' > 0$:

$$\|(\hat{B}^{(k)}_{\text{restricted}}, \hat{\Theta}_\epsilon^{(k)}) - (B^*, \Theta_\epsilon^*)\|_F \leq R'(p_1, p_2, n). \quad (11)$$

Consequently, the rest of the derivation in Theorem 1 follows, leading to the convergence property. The bound in eqn (11) will be shown at the end of Section 3.2.

3.2 Estimation Consistency.

In this subsection, we show that given a random realization of X and E , with high probability, the sequence $\{(\hat{B}^{(k)}, \hat{\Theta}_\epsilon^{(k)})\}_{k=1}^\infty$ lies in a non-expanding ball around (B^*, Θ_ϵ^*) , thus satisfying the condition of Theorem 1 for convergence of the alternating algorithm.

It should be noted that for the alternating search procedure, we restrict our estimation on a subspace identified by the screening step. However, for the remaining of this subsection, the main propositions and theorems are based on the procedure without such restriction, i.e., we consider “generic” regressions on the entire space of dimension $p_1 \times p_2$. Notwithstanding, it can be easily shown that the theoretical results for the regression parameters on a restricted domain follow easily from the generic case, as explained in Remark 9.

Before providing the details of the main theorem statements and proofs, we first introduce additional notations. Let $\beta = \text{vec}(B)$ be the vectorized version of the regression coefficient matrix. Correspondingly, we have $\tilde{\beta}^{(k)} = \text{vec}(\tilde{B}^{(k)})$ and $\beta^* = \text{vec}(B^*)$. Moreover, we drop the superscripts and use β and Θ_ϵ to denote the generic estimators given by (12) and (13), as opposed to those obtained in any specific iteration:

$$\hat{\beta} \equiv \underset{\beta \in \mathbb{R}^{p_1 p_2}}{\text{argmin}} \left\{ -2\beta' \hat{\gamma} + \beta' \hat{\Gamma} \beta + \lambda_n \|\beta\|_1 \right\}, \quad (12)$$

$$\hat{\Theta}_\epsilon \equiv \underset{\Theta_\epsilon \in \mathbb{S}_+^{p_1 p_2}}{\text{argmin}} \left\{ -\log \det \Theta_\epsilon + \text{tr}(\hat{S} \Theta_\epsilon) + \rho_n \|\Theta_\epsilon\|_{1, \text{off}} \right\}, \quad (13)$$

where

$$\hat{\Gamma} = \left(\hat{\Theta}_\epsilon \otimes \frac{X'X}{n} \right), \quad \hat{\gamma} = \left(\hat{\Theta}_\epsilon \otimes X' \right) \text{vec}(Y)/n, \quad \hat{S} = \frac{1}{n} (Y - X\hat{B})' (Y - X\hat{B}).$$

Remark 6. As opposed to (12) and (13), if $\hat{\gamma}$ and $\hat{\Gamma}$ are replaced by plugging in the true values of the parameters, the two problems in (12) and (13) become

$$\tilde{\beta} \equiv \underset{\beta \in \mathbb{R}^{p_1 p_2}}{\text{argmin}} \left\{ -2\beta' \tilde{\gamma} + \beta' \tilde{\Gamma} \beta + \lambda_n \|\beta\|_1 \right\}, \quad (14)$$

$$\tilde{\Theta}_\epsilon \equiv \underset{\Theta_\epsilon \in \mathbb{S}_+^{p_1 p_2}}{\text{argmin}} \left\{ -\log \det \Theta_\epsilon + \text{tr}(S \Theta_\epsilon) + \rho_n \|\Theta_\epsilon\|_{1, \text{off}} \right\}, \quad (15)$$

where

$$\tilde{\Gamma} = \left(\Theta_\epsilon^* \otimes \frac{X'X}{n} \right), \quad \tilde{\gamma} = \left(\Theta_\epsilon^* \otimes X' \right) \text{vec}(Y)/n, \quad S = \frac{1}{n} (Y - XB^*)' (Y - XB^*) \equiv \tilde{\Sigma}_\epsilon.$$

In (14), we obtain $\tilde{\beta}$ using a penalized maximum likelihood regression estimate, and (15) corresponds to the generic setting for using the graphical Lasso. A key difference between

the estimation problems in (12) and (13) versus those in (14) and (15) is that to obtain $\widehat{\beta}$ and Θ_ϵ we use *estimated quantities* rather than the raw data. This is exactly how we implement our iterative algorithm, namely, we obtain $\widehat{\beta}^{(k)}$ using $\widehat{S}^{(k-1)}$ as a surrogate for the sample covariance of the true error (which is unavailable), then estimate $\widehat{\Theta}_\epsilon^{(k)}$ using the information in $\widehat{\beta}^{(k)}$. This adds complication for establishing the consistency results. Original consistency results for the estimation problem in (14) and (15) are available in Basu and Michailidis (2015) and Ravikumar et al. (2011), respectively. Here we borrow ideas from corresponding theorems in those two papers, but need to tackle concentration bounds of relevant quantities with additional care. This part of the result and its proof are shown in Theorem 4.

As a road map toward our desired result established in Theorem 4, we first show in Theorem 2 that for any fixed realization of X and E , under a number of conditions on (or related to) X and E , when $\|\widehat{\Theta}_\epsilon - \Theta_\epsilon^*\|_\infty$ is small (up to a certain order), the error of $\widehat{\beta}$ is well-bounded. We then verify in Proposition 1 and 2 that for random X and E , the above-mentioned conditions hold with high probability. Similarly in Theorem 3, we show that for fixed realizations in X and E , under certain conditions (verified for random X and E in Proposition 3), the error of $\widehat{\Theta}_\epsilon$ is also well-bounded, given $\|\widehat{\beta} - \beta^*\|_1$ being small. Finally in Theorem 4, we show that for random X and E , with high probability, the iterative algorithm gives $\{(\widehat{\beta}^{(k)}, \Theta_\epsilon^{(k)})\}$ that lies in a small ball centered at $(\beta^*, \Theta_\epsilon^*)$, whose radius depends on p_1, p_2, n and the sparsity levels.

Next, for establishing the main propositions and theorems, we introduce some additional notations.

- Sparsity level of β^* : $s^{**} := \|\beta^*\|_0 = \sum_{j=1}^{p_2} \|\beta_j^*\|_0 = \sum_{j=1}^{p_2} s_j^*$. As a reminder of the previous notation, we have $s^* = \max_{j=1, \dots, p_2} s_j^*$.
- True edge set of Θ_ϵ^* : S_ϵ^* and let $s_\epsilon^* := |S_\epsilon^*|$ be its cardinality.
- Hessian of the log-determinant barrier $\log \det \Theta$ evaluated at Θ_ϵ^* :

$$H^* := \frac{d^2}{d\Theta^2} \log |\Theta|_{\Theta_\epsilon^*} = \Theta_\epsilon^{*-1} \otimes \Theta_\epsilon^{*-1}.$$
- Matrix infinity norm of the true error covariance matrix Σ_ϵ^* :

$$\kappa_{\Sigma_\epsilon^*} := \|\Sigma_\epsilon^*\|_\infty = \max_{i=1, 2, \dots, p_2} \sum_{j=1}^{p_2} |\Sigma_{\epsilon, ij}^*|.$$

- Matrix infinity norm of the Hessian restricted to the true edge set:

$$\kappa_{H^*} := \left\| \left(H_{S_\epsilon^* S_\epsilon^*}^* \right) \right\|_\infty = \max_{i=1, 2, \dots, p_2} \sum_{j=1}^{p_2} \left| H_{S_\epsilon^* S_\epsilon^*}^* \right|.$$

- Maximum degree of Θ_ϵ^* : $d := \max_{i=1, 2, \dots, p_2} \|\Theta_{\epsilon, i}^*\|_0$.
- We write $A \gtrsim B$ if there exists some absolute constant c that is independent of the model parameters such that $A \geq cB$.

Definition 4 (Incoherence condition (Ravikumar et al., 2011)). Θ_ϵ^* satisfies the incoherence condition if:

$$\max_{e \in (S_\epsilon^*)^c} \|H_{e S_\epsilon^*}^* (H_{S_\epsilon^* S_\epsilon^*}^*)^{-1}\|_1 \leq 1 - \xi, \quad \text{for some } \xi \in (0, 1).$$

Definition 5 (Restricted eigenvalue (RE) condition (Loh and Wainwright, 2012)). A symmetric matrix $A \in \mathbb{R}^{m \times m}$ satisfies the RE condition with curvature $\varphi > 0$ and tolerance $\phi > 0$, denoted by $A \sim RE(\varphi, \phi)$ if

$$\theta^* A \theta \geq \varphi \|\theta\|^2 - \phi \|\theta\|_1^2, \quad \forall \theta \in \mathbb{R}^m.$$

Definition 6 (Diagonal dominance). A matrix $A \in \mathbb{R}^{m \times m}$ is strictly diagonally dominant if

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|, \quad \forall i = 1, \dots, m.$$

Based on the model in Section 2.1, since we are assuming $\mathbf{X} = (X_1, \dots, X_{p_1})'$ and $\epsilon = (\epsilon_1, \dots, \epsilon_{p_2})$ come from zero-mean Gaussian distributions, it follows that \mathbf{X} and ϵ are zero-mean sub-Gaussian random vectors with parameters (Σ_X, σ_X^2) and $(\Sigma_\epsilon, \sigma_\epsilon^2)$, respectively. Moreover, throughout this section, all results are based on the assumption that Θ_ϵ^* is diagonally dominant.

Remark 7. Before moving on to the main statements of Theorem 2, we would like to point out that with a slight abuse of notation, for Theorem 2 and its related propositions and corollaries, the statements and analyses are based on equation (12) only, with any deterministic symmetric matrix Θ_ϵ within a small ball around Θ_ϵ^* . Similarly in Theorem 3, Proposition 3 and Corollary 2, the analyses are based on equation (13) only, for any given deterministic $\widehat{\beta}$ within a small ball around β^* . The randomness of $\widehat{\beta}$ and $\widehat{\Theta}_\epsilon$ during the iterative procedure will be taken into consideration comprehensively in Theorem 4.

Theorem 2 (Error bound for $\widehat{\beta}$ with fixed realizations of X and E). Consider $\widehat{\beta}$ given by (12). For any fixed pair of realizations of X and E , assume the following:

A1. $\widehat{\Theta}_\epsilon$ is a deterministic matrix satisfying the bound $\|\widehat{\Theta}_\epsilon - \Theta_\epsilon^*\|_\infty \leq \eta_\Theta$ where $\eta_\Theta = \eta_\Theta \left(\sqrt{\frac{\log p_2}{n}} \right)$ and η_Θ is some constant depending only on Θ_ϵ^* ;

A2. $\widehat{\Gamma} \sim RE(\varphi, \phi)$, with $s^{**} \phi \leq \varphi/32$;

A3. $(\widehat{\Gamma}, \widehat{\gamma})$ satisfies the deviation bound

$$\|\widehat{\gamma} - \widehat{\Gamma} \beta^*\|_\infty \leq \mathbb{Q}(\eta_\Theta) \sqrt{\frac{\log(p_1 p_2)}{n}},$$

where $\mathbb{Q}(\eta_\Theta)$ is some quantity depending on η_Θ .

Then, for any $\lambda_n \geq 4\mathbb{Q}(\eta_\Theta) \sqrt{\frac{\log(p_1 p_2)}{n}}$, the following bound holds:

$$\|\widehat{\beta} - \beta^*\|_1 \leq 64 s^{**} \lambda_n / \varphi.$$

The following two propositions verify the RE condition for $\hat{\Gamma}$ and deviation bound for $(\hat{\Gamma}, \hat{\gamma})$ hold with high probability for a random pair (X, E) , given any symmetric, matrix $\hat{\Theta}_\epsilon$ satisfying (A1).

Proposition 1 (Verification of RE condition for random X and E). *Consider any deterministic matrix $\hat{\Theta}_\epsilon$ satisfying (A1). Let the sample size satisfy $n \gtrsim \max\{s^{**} \log p_1, d^2 \log p_2\}$. With probability at least $1 - 2 \exp(-c_3 n)$ for some constant $c_3 > 0$, $\hat{\Gamma}$ satisfies the following RE condition:*

$$\hat{\Gamma} \equiv \hat{\Theta}_\epsilon \otimes (X'X/n) \sim RE \left(\varphi^* (\min_i \psi^i - d\nu_\Theta), \phi^* \max_i (\psi^i + d\nu_\Theta) \right),$$

where $\varphi^* = \frac{\Lambda_{\min}(\Sigma_X^*)}{2}$, $\phi^* = (\varphi^* \log p_1)/n$, and ψ^i is defined as:

$$\psi^i := \sigma_\epsilon^i - \sum_{j \neq i}^{p_2} \sigma_\epsilon^{ij},$$

where σ_ϵ^{ij} 's denote the entries in Θ_ϵ^* hence ψ^i is the gap between its diagonal entry and the sum of off-diagonal entries for row i .

Proposition 2 (Deviation bound for $(\hat{\Gamma}, \hat{\gamma})$ for random X and E). *Consider any deterministic matrix $\hat{\Theta}_\epsilon$ satisfying (A1). Let sample size n satisfy $n \gtrsim \log(p_1 p_2)$. With probability at least*

$$1 - 12c_1 \exp[-(c_2^2 - 1) \log(p_1 p_2)] \text{ for some } c_1 > 0, c_2 > 1,$$

the following bound holds:

$$\begin{aligned} \|\hat{\Gamma} - \hat{\Gamma}\beta^*\|_\infty &= \frac{1}{n} \|X'E\hat{\Theta}_\epsilon\|_\infty \leq \mathbb{Q}(\nu_\Theta) \sqrt{\frac{\log(p_1 p_2)}{n}}, \\ \mathbb{Q}(\nu_\Theta) &= c_2 \left\{ d\nu_\Theta [\Lambda_{\max}(\Sigma_X^*) \Lambda_{\max}(\Sigma_\epsilon^*)]^{1/2} + \left[\frac{\Lambda_{\max}(\Sigma_X^*)}{\Lambda_{\min}(\Sigma_\epsilon^*)} \right]^{1/2} \right\}. \end{aligned} \quad (16)$$

where

Remark 8. In Proposition 1, the quantity $d^2 \log p_2$ that shows up in the sample size requirement is a result of $\nu_\Theta = O(\sqrt{\log p_2/n})$, which is the common order of error in a generic graphical Lasso problem. Hence here we explicitly list it for the purpose of showing results for the generic graphical Lasso estimation problem. In our iterative algorithm, the order of $\nu_\Theta^{(k)}$ depends on the relative order of p_1 and p_2 , which may potentially make the sample size requirement more stringent. This will be discussed in more detail in the proof of Theorem 4.

Given the results in Theorem 2, Proposition 1 and Proposition 2, next we provide Corollary 1, which gives the error bound for $\hat{\beta}$ for random realizations of X and E .

Corollary 1 (Error Bound for $\hat{\beta}$ for random X and E). *Consider any deterministic $\hat{\Theta}_\epsilon$ satisfying the following element-wise ℓ_∞ -bound:*

$$\|\hat{\Theta}_\epsilon - \Theta_\epsilon^*\|_\infty \leq \nu_\Theta,$$

with $\nu_\Theta = \eta_\Theta \sqrt{\frac{\log p_2}{n}}$. Then for sample size $n \gtrsim \log(p_1 p_2)$ and for any regularization parameter $\lambda_n \geq 4\mathbb{Q}(\nu_\Theta) \sqrt{\frac{\log(p_1 p_2)}{n}}$ with the expression of $\mathbb{Q}(\cdot)$ given in (16), there exists $c_1 > 0$ and $c_2 > 1$ such that with probability at least:

$$1 - 12c_1 \exp[-(c_2^2 - 1) \log(p_1 p_2)] - 2 \exp(-c_3 n), \quad (17)$$

the following bound holds:

$$\|\hat{\beta} - \beta^*\|_1 \leq 64s^{**} \lambda_n / \varphi,$$

where $\varphi = \frac{1}{2} \Lambda_{\min}(\Sigma_\epsilon^*) (\min_i \psi^i - d\nu_\Theta)$.

Next, we move onto analyzing the error bound of the other component, for a fixed given $\hat{\beta}$.

Theorem 3 (Error bound for $\hat{\Theta}_\epsilon$ for fixed realizations of X and E). *Consider $\hat{\Theta}_\epsilon$ given by (13). For any fixed pair of realization (X, E) , assume the following:*

B1. $\hat{\beta}$ is a deterministic vector satisfying $\|\hat{\beta} - \beta^*\|_1 \leq \nu_\beta$, where $\nu_\beta = \eta_\beta \left(\sqrt{\frac{\log(p_1 p_2)}{n}} \right)$, with η_β being some constant depending only on β^* ;

B2. $\|\hat{S} - \Sigma_\epsilon^*\|_\infty \leq g(\nu_\beta)$ where

$$\hat{S} = \frac{1}{n} (Y - X\hat{B})(Y - X\hat{B}),$$

and $g(\nu_\beta)$ is some quantity depending on ν_β ;

B3. Incoherence condition holds for Θ_ϵ^* .

Then, for $\rho_n = (8/\xi)g(\nu_\beta)$ and sample size n satisfying $n \gtrsim \log(p_1 p_2)$, the following error bound for $\hat{\Theta}_\epsilon$ holds:

$$\|\hat{\Theta}_\epsilon - \Theta_\epsilon^*\|_\infty \leq \{2(1 + 8\xi^{-1})\kappa_H\}g(\nu_\beta), \quad (18)$$

where ξ is the incoherence parameter as defined in Definition 4.

Proposition 3 gives an explicit expression for $g(\nu_\beta)$ under condition (B1). Specifically, it shows how well \hat{S} concentrates around Σ_ϵ^* for random X and E , given some \hat{B} exhibiting a small error from its true value (or $\hat{\beta}$, equivalently),

Proposition 3. *Consider any deterministic $\hat{\beta}$ satisfying (B1). Then for sample size n satisfying $n \gtrsim \log(p_1 p_2)$, with probability at least*

$$1 - 1/p_1^{\tau_1 - 2} - 1/p_2^{\tau_2 - 2} - 6c_1 \exp[-(c_2^2 - 1) \log(p_1 p_2)], \text{ for some } c_1 > 0, c_2 > 1, \tau_1, \tau_2 > 2,$$

the following bound holds:

$$\|\hat{S} - \Sigma_\epsilon^*\|_\infty \leq g(\nu_\beta), \quad (19)$$

where

$$g(\nu_\beta) = \sqrt{\frac{\log 4 + \tau_2 \log p_2}{c_\epsilon^* n} + \nu_\beta^2} \left(\sqrt{\frac{\log 4 + \tau_1 \log p_1}{c_X^* n} + \max_i (\Sigma_{X,i}^*)} \right) + 2c_2 \nu_\beta [\Lambda_{\max}(\Sigma_X^*) \Lambda_{\max}(\Sigma_\epsilon^*)]^{1/2} \sqrt{\frac{\log(p_1 p_2)}{n}}, \quad (20)$$

c_ϵ^* and c_X^* are population quantities given in (57) and (62), respectively.

Given Theorem 3 and Proposition 3, we provide Corollary 2, which gives the error bound for $\hat{\Theta}_\epsilon$ for random realizations of X and E :

Corollary 2 (Error bound for $\hat{\Theta}$ for random X and E). Consider any deterministic $\hat{\beta}$ satisfying the following bound

$$\|\hat{\beta} - \beta^*\|_1 \leq \nu_\beta,$$

with $\nu_\beta = \eta_\beta \sqrt{\frac{\log(p_1 p_2)}{n}}$. Also suppose the incoherence condition (B3) is satisfied. Then, for sample size $n \gtrsim \log(p_1 p_2)$ and regularization parameter $\rho_n = (8/\xi)g(\nu_\beta)$ with $g(\nu_\beta)$ given in (20), with probability at least

$$1 - 1/p_1^{\tau_1 - 2} - 1/p_2^{\tau_2 - 2} - 6c_1 \exp[-(c_2^2 - 1) \log(p_1 p_2)], \quad \text{for some } c_1 > 0, c_2 > 1, \tau_1, \tau_2 > 2,$$

the following bound holds:

$$\|\hat{\Theta}_\epsilon - \Theta_\epsilon^*\|_\infty \leq \{2(1 + 8\xi^{-1})\kappa_{H^*}\}g(\nu_\beta).$$

After providing the error bound for (12) and (13), in Theorem 4 we establish that with high probability, the error of the sequence of estimates obtained in the alternating search step of the algorithm described in Section 2.2 is uniformly bounded; that is, the sequence of estimates lie in a non-expanding ball around the true value of the parameters uniformly with a radius that does not depend on the iteration number k .

Theorem 4 (Error bound for $\{\hat{\beta}^{(k)}\}$ and $\{\hat{\Theta}_\epsilon^{(k)}\}$). Consider the iterative algorithm given in Section 2.2 that gives rise to sequences of $\{\hat{\beta}^{(k)}\}$ and $\{\hat{\Theta}_\epsilon^{(k)}\}$ alternately. For random realization of X and E , we assume the following:

C1. The incoherence condition holds for Θ_ϵ^* .

C2. Θ_ϵ^* is diagonally dominant.

C3. The maximum sparsity level for all p_2 regression s^* satisfies $s^* = o(n/\log p_1)$.

(1) For sample size satisfying $n \gtrsim \log(p_1 p_2)$, there exist constants $c_1 > 0, c_2 > 1, c_3 > 0$ such that for any

$$\lambda_n^0 \geq 4c_2 [\Lambda_{\max}(\Sigma_X^*) \Lambda_{\max}(\Sigma_\epsilon^*)]^{1/2} \sqrt{\frac{\log(p_1 p_2)}{n}},$$

with probability at least $1 - 2 \exp(-c_3 n) - 6c_1 \exp[-(c_2^2 - 1) \log(p_1 p_2)]$, the initial estimate $\hat{\beta}^{(0)} \equiv \text{vec}(\hat{B}^{(0)})$ satisfies the following bound

$$\|\hat{\beta}^{(0)} - \beta^*\|_1 \leq 64s^{**} \lambda_n^0 / \varphi^* \equiv \nu_\beta^{(0)}, \quad (21)$$

where $\varphi^* = \Lambda_{\min}(\Sigma_X^*)/2$. Moreover, by choosing $\rho_n^0 = (\frac{\xi}{2})g(\nu_\beta^{(0)})$ where the expression for $g(\cdot)$ is given in (20), with probability at least

$$1 - 1/p_1^{\tau_1 - 2} - 1/p_2^{\tau_2 - 2} - 2 \exp(-c_3 n) - 6c_1 \exp[-(c_2^2 - 1) \log(p_1 p_2)], \quad \text{for some } \tau_1, \tau_2 > 2,$$

the following bound holds:

$$\|\hat{\Theta}_\epsilon^{(0)} - \Theta_\epsilon^*\|_\infty \leq \{2(1 + 8\xi^{-1})\kappa_{H^*}\}g(\nu_\beta^{(0)}) \equiv \tau_\Theta^{(0)}. \quad (22)$$

(II) For sample size satisfying $n \gtrsim d^2 \log(p_1 p_2)$, for any iteration $k \geq 1$, with probability at least

$$1 - 1/p_1^{\tau_1 - 2} - 1/p_2^{\tau_2 - 2} - 12c_1 \exp[-(c_2^2 - 1) \log(p_1 p_2)] - 2 \exp[-c_3 n],$$

the following bounds hold for all $\hat{\beta}^{(k)}$ and $\hat{\Theta}_\epsilon^{(k)}$:

$$\begin{aligned} \|\hat{\beta}^{(k)} - \beta^*\|_1 &\leq C_\beta \left(s^{**} \sqrt{\frac{\log(p_1 p_2)}{n}} \right), \\ \|\hat{\Theta}_\epsilon^{(k)} - \Theta_\epsilon^*\|_\infty &\leq C_\Theta \left(\sqrt{\frac{\log(p_1 p_2)}{n}} \right). \end{aligned}$$

where s^{**} is the sparsity of β^* , C_β and C_Θ are constants depending only on β^* and Θ_ϵ^* , respectively.

As a direct result of Proposition 1 in Basu and Michailidis (2015) and Corollary 3 in Ravikumar et al. (2011), the following bound also holds:

Corollary 3. Under the same set of conditions C1, C2 and C3 in Theorem 4, there exists $\tau_1, \tau_2 > 2, c_1 > 0, c_2 > 1, c_3 > 0$ and constants C'_β and C'_Θ such that for all iterations k , the following bound holds:

$$\begin{aligned} \|\hat{\beta}^{(k)} - \beta^*\|_F &\leq C'_\beta \left(\sqrt{\frac{s^{**} \log(p_1 p_2)}{n}} \right), \\ \|\hat{\Theta}_\epsilon^{(k)} - \Theta_\epsilon^*\|_F &\leq C'_\Theta \sqrt{\frac{(s_\epsilon^* + p_2) \log(p_1 p_2)}{n}}, \end{aligned}$$

with probability at least

$$1 - 1/p_1^{\tau_1 - 2} - 1/p_2^{\tau_2 - 2} - 12c_1 \exp[-(c_2^2 - 1) \log(p_1 p_2)] - 2 \exp[-c_3 n],$$

where s^{**} and s_ϵ^* are the sparsity for β^* and Θ_ϵ^* , respectively.

Remark 9. As mentioned earlier in this subsection, the actual implementation of the alternating search step is restricted to a subspace of $\mathbb{R}^{p_1 \times p_2}$. Next, we outline the corresponding theoretical results for this specific scenario in which for each regression j , some fixed superset of the indices of true covariates is given, and the regressions are restricted to these supersets, respectively. Note that we need to make sure that the restricted subspace contains all the true covariates for the results below to be valid.

Let S_j denote the given *fixed superset* for each regression j , and we consider regressing the response on X_{S_j} . We use $\widehat{\beta}_R^{(k)}$ to denote the corresponding vectorized estimator of iteration k , that is,

$$\widehat{\beta}_R^{(k)} = (\widehat{B}_{1,\text{Restricted}}^{(k)}, \dots, \widehat{B}_{p_2,\text{Restricted}}^{(k)})',$$

where $\widehat{B}_{j,\text{Restricted}}^{(k)}$ is obtained by doing the regression in (7), however with the indices of covariates restricted to S_j . Also, we let β_R^* be the corresponding true value of $\widehat{\beta}_R^{(k)}$. Note that always holds that

$$\|\widehat{\beta}_R^{(k)} - \beta_R^*\| = \|\widehat{\beta}^{(k)} - \beta^*\|.$$

Now let

$$\bar{S} = \bigcup_{j \in \{1, \dots, p_2\}} S_j,$$

and let \bar{s} be its cardinality. It can be shown that the best achievable error bound for $\widehat{\beta}_R^{(k)}$ is identical to $\widehat{\beta}_S^{(k)}$, where $\widehat{\beta}_S^{(k)}$ is obtained by considering covariates $X_{\bar{S}}$ for all p_2 regressions, instead of the entire X . For this specific reason, formally, we state the theoretical results for the case where we consider regressing on $X_{\bar{S}}$, which is almost identical to the generic case.

Suppose conditions C1, C2 and C3 in Theorem 4 hold, then there exists constants $c_1 > 0, c_2 > 1, c_3 > 0, \tau_1 > 2, \tau_2 > 2$ such that: (I) for sample size satisfying $n \gtrsim \log(\bar{s}p_2)$, w.p. at least $1 - 2 \exp(-c_3 n) - 6c_1 \exp[-(c_2^2 - 1) \log(\bar{s}p_2)]$, for any

$$\lambda_n^0 \geq 4c_2 \left[\Lambda_{\max}(\Sigma_{X_{\bar{S}}}^*) \Lambda_{\max}(\Sigma_\epsilon^*) \right]^{1/2} \sqrt{\frac{\log(\bar{s}p_2)}{n}},$$

the initial estimate $\widehat{\beta}_S^{(0)}$ satisfies the following bound:

$$\|\widehat{\beta}_S^{(0)} - \beta_S^*\|_1 \leq 64s^{**} \lambda_n^0 / \varphi_S^* \equiv \nu_{\beta_S}^{(0)},$$

where $\varphi_S^* = \Lambda_{\min}(\Sigma_{X_{\bar{S}}}^*)/2$. Moreover, by choosing $\rho_n^0 = \binom{\bar{s}}{\xi} g(\nu_{\beta_S}^{(0)})$ where the expression for $g(\cdot)$ is given in (20), with probability at least

$$1 - 1/\bar{s}^{\tau_1 - 2} - 1/p_2^{\tau_2 - 2} - 2 \exp(-c_3 n) - 6c_1 \exp[-(c_2^2 - 1) \log(\bar{s}p_2)],$$

the following bound holds:

$$\|\widehat{\Theta}_\epsilon^{(0)} - \Theta_\epsilon^*\|_\infty \leq \{2(1 + 8\xi^{-1})\kappa_{H^*}\} g(\nu_{\beta_S}^{(0)}) \equiv \nu_\Theta^{(0)}.$$

(II) For sample size satisfying $n \gtrsim d^2 \log(\bar{s}p_2)$, for any iteration $k \geq 1$, with probability at least

$$1 - 1/\bar{s}^{\tau_1 - 2} - 1/p_2^{\tau_2 - 2} - 12c_1 \exp[-(c_2^2 > 1) \log(\bar{s}p_2)] - 2 \exp[-c_3 n],$$

the following bound hold for all $\widehat{\beta}_S^{(k)}$ and $\widehat{\Theta}_\epsilon^{(k)}$:

$$\begin{aligned} \|\widehat{\beta}_S^{(k)} - \beta_S^*\|_1 &\leq C_\beta \left(s^{**} \sqrt{\frac{\log(\bar{s}p_2)}{n}} \right), \quad \|\widehat{\beta}_S^{(k)} - \beta_S^*\|_F \leq C'_\beta \left(\sqrt{\frac{s^{**} \log(\bar{s}p_2)}{n}} \right), \\ \|\widehat{\Theta}_\epsilon^{(k)} - \Theta_\epsilon^*\|_\infty &\leq C_\Theta \left(\sqrt{\frac{\log(\bar{s}p_2)}{n}} \right), \quad \|\widehat{\Theta}_\epsilon^{(k)} - \Theta_\epsilon^*\|_F \leq C'_\Theta \sqrt{\frac{(c_\epsilon^* + p_2) \log(\bar{s}p_2)}{n}}, \end{aligned}$$

where s^{**} is the sparsity of β^* , $C_\beta, C_\beta', C_\Theta, C_\Theta'$ and C'_Θ are all constants that do not depend on n, \bar{S}, p_2 .

3.3 Family-wise Error Rate Control of the Screening Step.

As mentioned in the Introduction, for the iterative algorithm to work effectively, it is crucial to initialize from points that are close to the true parameters. Our screening step provides such guarantees *asymptotically*. Based on the screening step described in Section 2.2, initial estimates for each column of the regression matrix are obtained by Lasso or Ridge regression with the support set restricted to the one identified by the screening step. It is desirable for the screening step to correctly identify the true support set. In particular, we would like to retain as many true positive predictor variables as possible without discovering too many false positive ones. The following theorem states that as long as $\log(p_1 p_2)/n = o(1)$ and the sparsity is not beyond a specified level, the screening step will be able to recover all true positive predictors, while keeping the family-wise type I error under control.

Theorem 5. *Let S_j^* denote the true support set of the j th regression and s_j^* be its cardinality. Suppose that $\log(p_1 p_2)/n \rightarrow 0$ and the following condition for sparsity holds:*

$$\max\{s_j^*, j = 1, \dots, p_2\} = o(\sqrt{n}/\log p_1).$$

Then, the screening step described in Section 2.2 will correctly recover S_j^ for all $j = 1, \dots, p_2$ with probability approaching to 1, while keeping the family-wise type I error rate under the pre-specified level α .*

Remark 10. The specified level for sparsity is necessary for the de-biased Lasso procedure in Javanmard and Montanari (2014) to produce unbiased estimates for the regression coefficients. In terms of support recovery for the screening step, with $\log(p_1 p_2)/n = o(1)$, we only require $s^* = o(p_1)$, which is much weaker and easily satisfied.

The following corollary connects the screening step with the alternating search step, under the discussed asymptotic regime :

Corollary 4. *Consider the model set-up given in Section 2.1. Let s^* denote the maximum sparsity for all $B_{j,j}^*, j = 2, \dots, p_2$, and d denote the maximum degree of Θ_ϵ^* . Also, let s^{**} denote the sparsity for β^* and s_ϵ^* denote the sparsity for Θ_ϵ^* . Assume there exist positive constants $c_s^*, c_{s^{**}}, c_d, c_{\bar{s}}, c_{p_2}$ satisfying*

$$0 < c_{s^*} + c_{\bar{s}} < 1/2; \quad 0 < c_{s^{**}} + c_{\bar{s}} < 1; \quad 0 < 2c_d + c_{\bar{s}} < 1; \quad 0 < \max\{c_{s_\epsilon^*}, c_{p_2}\} + c_{\bar{s}} < 1$$

such that

$$s^* = O(n^{c_s}); \quad s^{**} = O(n^{c_{s^{**}}}); \quad s_\epsilon^* = O(n^{c_{s_\epsilon^*}}); \quad d = O(n^{c_d}); \quad \bar{s} = O(e^{n^{c_{\bar{s}}}}); \quad p_2 = O(n^{c_{p_2}}).$$

As $n \rightarrow \infty$,

$\mathbb{P}(\{\text{The screening step correctly recovers the true support set for all } B_{j,j}, j = 1, \dots, p\}) \rightarrow 1,$

and for all iterations k :

$$\max_{k \geq 1} \left\| (\widehat{\beta}_R, \widehat{\Theta}_\epsilon^{(k)}) - (\beta_R^*, \Theta_\epsilon^*) \right\| \xrightarrow{P} 0.$$

The proof of this corollary follows along the same lines as Theorem 4, and we leave the details to the reader.

3.4 Estimation Error and Identifiability.

In this subsection, we discuss in detail the conditions needed for the parameters in our multi-layered network to be identifiable (estimable). We focus the presentation for ease of exposition on a three-layer network and then discuss the general M -layer case.

Consider a 3-layer graphical model. Let $\tilde{X} = [(X^1)^\top, (X^2)^\top]^\top$ be the $(p_1 + p_2)$ dimensional random variable, which represents the ‘‘super’’-layer on which we regress X^3 to estimate B^{13} , B^{23} and Σ^3 . As shown in Theorem 2, the estimation error for β takes the following form:

$$\|\hat{\beta} - \beta^*\|_1 \leq 64s^{**} \lambda_{n_l}/\varphi,$$

where φ is the curvature parameter for RE condition that scales with $\Lambda_{\min}(\Sigma_{\tilde{X}})$ (see Proposition 1). Therefore, the error of estimating these regression parameters is higher when $\Lambda_{\min}(\Sigma_{\tilde{X}})$ is smaller. In this section, we derive a lower bound on this quantity to demonstrate how the estimation error depends on the underlying structure of the graph.

For the undirected subgraph within a layer k , we denote its maximum node capacity by $v(\Theta^k) := \max_{1 \leq i \leq p_k} \sum_{j=1}^{p_k} |\Theta_{ij}^k|$. For the directed bipartite subgraph consisting of Layer $s \rightarrow t$ edges ($s < t$), we similarly define the maximum incoming and outgoing node capacities by $v_{in}(B^{st}) := \max_{1 \leq i \leq p_s} \sum_{j=1}^{p_t} |B_{ij}^{st}|$ and $v_{out}(B^{st}) := \max_{1 \leq i \leq p_s} \sum_{j=1}^{p_t} |B_{ij}^{st}|$. The following proposition establishes the lower bound in terms of these node capacities

Proposition 4.

$$\Lambda_{\min}(\Sigma_{\tilde{X}}) \geq v(\Theta^1)^{-1} v(\Theta^2)^{-1} [1 + (v_{in}(B^{12}) + v_{out}(B^{12})) / 2]^{-2}$$

The three components in the lower bound demonstrate how the structure of Layers 1 and 2 impact the accurate estimation of directed edges to Layer 3. Essentially, the bound suggests that accurate estimation is possible when the total effect (incoming and outgoing edges) at every node of each of the three subgraphs is not very large.

This is inherently related to the identifiability of the multi-layered graphical models and our ability to distinguish between the parents from different layers. For instance, if a node in Layer 2 has high partial correlation with nodes of Layer 1, i.e., a node in Layer 2 has parents from many nodes in Layer 1 and yields a large $v_{in}(B^{12})$; or similarly, a node in Layer 1 is the parent of many nodes in Layer 2, yielding a large $v_{out}(B^{12})$. In either case, we end up with some large lower bound for $\Lambda_{\min}(\Sigma_{\tilde{X}})$ and it can be hard to distinguish Layer 1 \rightarrow 3 edges from Layer 2 \rightarrow 3 edges.

For a general M -layer network, the argument in the proof of Proposition 4 (see Section B for details) can be generalized in a straightforward manner. In the 2-layer network setting, with the notation defined in Section 2, by setting $e^1 = X^1$, we have

$$\begin{bmatrix} e^1 \\ e^2 \end{bmatrix} = P \begin{bmatrix} X^1 \\ X^2 \end{bmatrix}, \quad \text{where } P = \begin{bmatrix} I & 0 \\ -(B^{12})^\top & I \end{bmatrix}.$$

For an M -layer network, a modified P is given in the following form:

$$P = \begin{bmatrix} I & 0 & \cdots & 0 \\ -(B^{12})^\top & I & \cdots & 0 \\ \vdots & \vdots & \vdots & 0 \\ -(B^{1,M-1})^\top & -(B^{2,M-1})^\top & \cdots & I \end{bmatrix}$$

and combines node capacities for different layers. The conclusion is qualitatively similar, i.e., the estimation error of an M -layer graphical model is small as long as the maximum node capacities of different inter-layer and intra-layer subgraphs are not too large.

4. Performance Evaluation and Implementation Issues.

In this section, we present selected simulation results for our proposed method, in two-layer and three-layer network settings. Further, we introduce some acceleration techniques that can speed up the algorithm and reduce computing time.

4.1 Simulation Results.

For the 2-layer network, as mentioned in Section 2.1, since the main target of our proposed algorithm is to provide estimates for B^* and Θ_ε^* (since Θ_X can be estimated separately), we only present evaluation results for B^* and Θ_ε^* estimates. Similarly, for the three-layer network, we only present evaluation results involving Layer 3, using the notation in Section 3.4, that is, B_{XZ}^* , B_{YZ}^* and $\Theta_{\varepsilon Z}^*$ estimates, which is sufficient to show how our proposed algorithm works in the presence of a ‘‘super’’-layer, taking advantage of the separability of the log-likelihood.

2-layered Network. To compare the proposed method with the most recent methodology that also provides estimates for the regression parameters and the precision matrix (CAPME, Cai et al. (2012)), we use the exact same model settings that have been used in that paper. Specifically, we consider the following two models:

- Model A: Each entry in B^* is nonzero with probability $5/p_1$, and off-diagonal entries for Θ_ε^* are nonzero with probability $5/p_2$.
- Model B: Each entry in B^* is nonzero with probability $30/p_1$, and off-diagonal entries for Θ_ε^* are nonzero with probability $5/p_2$.

As in Cai et al. (2012), for both models, nonzero entries of B^* and Θ_ε^* are generated from $\text{Unif}([-1, -0.5] \cup [0.5, 1])$, and diagonals of Θ_ε^* are set identical such that the condition number of Θ_ε^* is p_2 .

	(p_1, p_2, n)
Model A	$p_1 = 30, p_2 = 60, n = 100$
	$p_1 = 60, p_2 = 30, n = 100$
	$p_1 = 200, p_2 = 200, n = 150$
Model B	$p_1 = 300, p_2 = 300, n = 150$
	$p_1 = 200, p_2 = 200, n = 100$
	$p_1 = 200, p_2 = 200, n = 200$

Table 1: Model Dimensions for Model A and B

To evaluate the selection performance of the algorithm, we use sensitivity (SEN), specificity (SPE) and Matthews Correlation Coefficient (MCC) as criteria:

$$\text{SEN} = \frac{\text{TN}}{\text{TN} + \text{FP}}, \quad \text{SPE} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}.$$

Further, to evaluate the accuracy of the magnitude of the estimates, we use the relative error in Frobenius norm:

$$\text{rel-Fnorm} = \frac{\|\tilde{B} - B^*\|_F}{\|B^*\|_F} \quad \text{or} \quad \frac{\|\tilde{\Theta}_\epsilon - \Theta_\epsilon^*\|_F}{\|\Theta_\epsilon^*\|_F}.$$

Tables 2 and 3 show the results for both the regression matrix and the precision matrix. For the precision matrix estimation, we compare our result with those available in Cai et al. (2012), denoted as CAPME.

	(p_1, p_2, n)	SEN	SPE	MCC	rel-Fnorm
Model A	(30,60,100)	0.96(0.018)	0.99(0.004)	0.93(0.014)	0.22(0.029)
	(60,30,100)	0.99(0.009)	0.99(0.003)	0.93(0.017)	0.18(0.021)
	(200,200,150)	0.99(0.001)	0.99(0.001)	0.88(0.009)	0.18(.007)
Model B	(300,300,150)	1.00(0.001)	0.99(0.001)	0.84(0.010)	0.21(0.007)
	(200,200,200)	0.970(0.004)	0.982(0.001)	0.927(0.002)	0.194 (0.009)
	(200,200,100)	0.32(0.010)	0.99(0.001)	0.49(0.009)	0.85(0.006)

Table 2: Performance evaluation for the estimated regression matrix, over 50 replications

	(p_1, p_2, n)	SEN	SPE	MCC	rel-Fnorm	
Model A	(30,60,100)	0.77(0.031)	0.92(0.007)	0.56(0.030)	0.51(0.017)	
		CAPME	0.58(0.03)	0.89(0.01)	0.45(0.03)	
	(60,30,100)	0.76(0.041)	0.89(0.015)	0.59(0.039)	0.49(0.014)	
	(200,200,150)	0.78(0.019)	0.97(0.001)	0.55(0.012)	0.60(0.007)	
	(300,300,150)	0.71(0.017)	0.98(0.001)	0.51(0.011)	0.59(0.005)	
Model B	(200,200,200)	0.73(0.023)	0.94(0.003)	0.39(0.017)	0.62(0.011)	
		CAPME	0.36(0.02)	0.97(0.00)	0.35(0.01)	
	(200,200,100)	0.57(0.027)	0.44(0.007)	0.04(0.008)	0.84(0.002)	
		CAPME	0.19(0.01)	0.87(0.00)	0.04(0.01)	

Table 3: Performance evaluation for estimated precision matrix, over 50 replications

As it can be seen from Tables 2 and 3, the sample size is a key factor that affects the performance. Our proposed algorithm performs extremely well in its selection properties on B and strikes a good balance between sensitivity and specificity in estimating Θ_ϵ .³ For most settings, it provides substantial improvements over the CAPME estimator.

3-layer Network. For a 3-layer network, we consider the following data generation mechanism: for all three models A, B and C, each entry in B_{XY} is nonzero with probability

$5/p_1$, each entry in B_{XZ} and B_{YZ} is nonzero with probability $5/(p_1 + p_2)$, and off-diagonal entries in $\Theta_{\epsilon,Z}$ are nonzero with probability $5/p_3$. Similar to the 2-layered set-up, the nonzero entries in $\Theta_{\epsilon,Z}$ are generated from $\text{Unif}([-1, -0.5] \cup (0.5, 1])$ with its diagonals set identical such that its condition number is p_3 . For the regression matrices in the three models, nonzeros in B_{XY} are generated from $\text{Unif}([-1, -0.5] \cup (0.5, 1])$, and nonzeros in B_{XZ} and B_{YZ} are generated from $\{\text{Unif}([-1, -0.5] \cup (0.5, 1]) * \text{SignalStrength}\}$, where the signal strength in the three models are given by 1, 1.5 and 2, respectively. More specifically, for Model A, B and C, nonzeros in B_{XZ} or B_{YZ} are generated from $\text{Unif}([-1, -0.5] \cup (0.5, 1])$, $\text{Unif}([-1.5, -0.75] \cup (0.75, 1.5])$ and $\text{Unif}([-2, -1] \cup (1, 2])$, respectively.

	Layer 3 SignalStrength	(p_1, p_2, p_3, n)
Model A	1	(50,50,50,200)
Model B	1.5	(50,50,50,200)
Model C	2	(50,50,50,200)
		(20,80,50,200)
		(80,20,50,200)
		(100,100,100,200)

Table 4: Model Dimensions and Signal Strength for Model A, B and C

As mentioned in the beginning of this subsection, we only evaluate the algorithm's performance on B_{XZ}, B_{YZ} and $\Theta_{\epsilon,Z}$.

Based on the results shown in Tables 5, 6 and 7, the signal strength across layers affects the accuracy of the estimation, which is in accordance with what has been discussed regarding identifiability. Overall, the MLE estimator performs satisfactorily across a fairly wide range of settings and in many cases achieving very high values for the MCC criterion.

4.1.1 SIMULATION RESULTS FOR NON-GAUSSIAN DATA

In many applications, the data may not be exactly Gaussian, but approximately Gaussian. Next, we present selected simulation results when the data comes from some distribution that deviates from Gaussian. Specifically, we consider two types of deviations based on the following transformations: (i) a truncated empirical cumulative distribution function and (ii) a shrunken empirical cumulative distribution functions as discussed in Zhao et al. (2015). In both simulation settings, we consider Model A with $(p_1, p_2, n) = (30, 60, 100)$ under the two-layer setting, and the transformation is applied to errors in Layer 2. Table 8 shows the simulation results for these two scenarios over 50 replications.

Based on the results in Table 8, relatively small deviations from the Gaussian distribution do not affect the performance of the MLE estimates under the examined settings that are comparable to those obtained with Gaussian distributed data.

4.2 A comparison with the two-step estimator in Cai et al. (2012)

Next, we present a comparison between the MLE estimator and the two-step estimator of Cai et al. (2012). Specifically, we use the CAPME estimate as an initializer for the

3. In practice, for the debias Lasso procedure, we use the default choice of tuning parameters suggested in the implementation of the code provided in Javanmard and Montanari (2014); for FWER, we suggest using $\alpha = 0.1$ as the thresholding level; for tuning parameter selection, we suggest doing a grid search for (λ_n, ρ_n) on $[0, 0.5\sqrt{\log p_1/n}] \times [0, 0.5\sqrt{\log p_2/n}]$ with BIC.

	(p_1, p_2, p_3, n)	SEN	SPE	MCC	rel-Fnorm
Model A	(50,50,50,200)	0.51(0.065)	0.99(0.001)	0.69(0.049)	0.68(0.050)
Model B	(50,50,50,200)	0.85(0.043)	0.99(0.001)	0.898(0.025)	0.36(0.056)
Model C	(50,50,50,200)	0.97(0.018)	0.99(0.002)	0.96(0.016)	0.16(0.040)
	(20,80,50,200)	0.55(0.078)	0.99(0.001)	0.72(0.059)	0.63(0.066)
	(80,20,50,200)	0.99(0.006)	0.99(0.002)	0.94(0.017)	0.076(0.032)
	(100,100,100,200)	1.00(0.001)	0.99(0.001)	0.87(0.016)	0.07(0.007)

Table 5: Performance evaluation for estimated regression matrix B_{XYZ} over 50 replications

	(p_1, p_2, p_3, n)	SEN	SPE	MCC	rel-Fnorm
Model A	(50,50,50,200)	0.53(0.051)	1.00(0.000)	0.72(0.036)	0.65(0.041)
Model B	(50,50,50,200)	0.90(0.033)	1.00(0.000)	0.95(0.019)	0.25(0.049)
Model C	(50,50,50,200)	0.98(0.013)	1.00(0.000)	0.99(0.007)	0.12(0.042)
	(20,80,50,200)	0.95(0.013)	1.00(0.000)	0.98(0.007)	0.19(0.030)
	(80,20,50,200)	0.96(0.027)	0.99(0.001)	0.97(0.022)	0.14(0.063)
	(100,100,100,200)	1.00(0.000)	1.00(0.000)	0.99(0.002)	0.025(0.002)

Table 6: Performance evaluation for estimated regression matrix B_{YZ} over 50 replications

	(p_1, p_2, p_3, n)	SEN	SPE	MCC	rel-Fnorm
Model A	(50,50,50,200)	0.69(0.044)	0.638(0.032)	0.20(0.036)	0.82(0.017)
Model B	(50,50,50,200)	0.77(0.050)	0.82(0.036)	0.42(0.071)	0.68(0.040)
Model C	(50,50,50,200)	0.88(0.041)	0.91(0.019)	0.63(0.059)	0.56(0.034)
	(20,80,50,200)	0.72(0.041)	0.80(0.028)	0.36(0.050)	0.72(0.021)
	(80,20,50,200)	0.90(0.028)	0.92(0.011)	0.68(0.039)	0.58(0.018)
	(100,100,100,200)	0.96(0.014)	0.96(0.003)	0.68(0.016)	0.049(0.010)

Table 7: Performance evaluation for estimated precision matrix $\Theta_{\epsilon,Z}$ over 50 replications

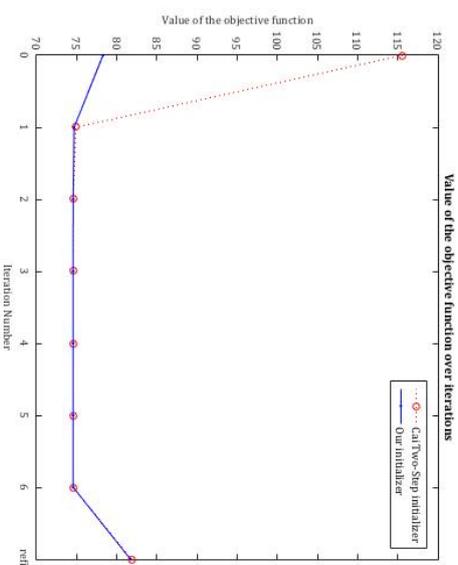
Setting	Parameter	SEN	SPE	MCC	rel-Fnorm
Model A (30,60,100)	B	0.96(0.017)	0.99(0.003)	0.94(0.012)	0.20(0.028)
shrunken	Θ_{ϵ}	0.76(0.031)	0.91(0.008)	0.55(0.030)	0.51(0.019)
Model A (30,60,100)	B	0.96(0.021)	0.98(0.004)	0.93(0.015)	0.21(0.034)
truncation	Θ_{ϵ}	0.76(0.033)	0.92(0.008)	0.56(0.035)	0.52(0.023)

Table 8: Simulation results for B and Θ_{ϵ} over 50 replications under npn transformation

MLE procedure and examine its evolution over successive iterations. We evaluate the value of the objective function at each iteration, and also compare it to the value of the objective function evaluated at our initializer (screening + Lasso/Ridge) and the estimates

afterward. For illustration purposes, we only show the results for a single realization under Model A with $p_1 = 30, p_2 = 60, n = 100$, although similar results were obtained in other simulation settings. Figure 2 shows the value of the objective function as a function of the iteration under both initialization procedures, while Table 9 shows how the cardinality of the estimates changes over iterations for both initializers. It can be seen that the iterative MLE algorithm significantly improves the value of the objective function over the CAPME initialization and also that the set of directed and undirected edges stabilizes after a couple iterations.

Figure 2: Comparison between Cai's estimate and our estimate



	0	1	2	3	4	5	6	refit
Our initializer	$\hat{B}^{(k)}$	275	275	275	275	275	275	275
	$\hat{\Theta}_{\epsilon}^{(k)}$	282	255	247	248	248	248	260
CAPME initializer	$\hat{B}^{(k)}$	433	275	275	275	275	275	275
	$\hat{\Theta}_{\epsilon}^{(k)}$	979	267	250	249	249	248	260

Table 9: Change in cardinality over iterations for B and Θ_{ϵ}

Based on Figure 2 and Table 9, we notice that Cai et. al's two-step estimator yields larger value of the objective function compared with our initializer that is obtained through screening followed by Lasso. However, over subsequent iterations, both initializers yield the same value in the objective function, which keeps decreasing according to the nature of block-coordinate descent.

4.3 Implementation issues

Next, we introduce some acceleration techniques for the MLE algorithm aiming to reduce computing time, yet maintaining estimation accuracy over iterations.

($p_2 + 1$)-block update. In Section 2, we update B and Θ_ϵ by (6) and (8), respectively, and within each iteration, the updated B is obtained by an application of cyclic p_2 -block coordinate descent with respect to each of its columns until convergence. As shown in Section 3.1, the outer 2-block update guarantees the MLE iterative algorithm to converge to a stationary point. However in practice, we can speed up the algorithm by updating B without waiting for it to reach the minimizer for every iteration other than the first one. More precisely, for the alternating search step, we take the following steps when actually implementing the proposed algorithm with initializer $\widehat{B}^{(0)}$ and $\widehat{\Theta}_\epsilon^{(0)}$:

- Iteration 1: update B and Θ_ϵ as follows, respectively:

$$\widehat{B}^{(1)} = \underset{B \in \mathcal{B}_1 \times \dots \times \mathcal{B}_{p_2}}{\operatorname{argmin}} \left\{ \frac{1}{n} \sum_{i=1}^{p_2} \sum_{j=1}^{p_2} (\sigma_\epsilon^{ij})^{(0)} (Y_i - X B_i)^\top (Y_j - X B_j) + \lambda_n \sum_{j=1}^{p_2} \|B_j\|_1 \right\},$$

and

$$\widehat{\Theta}_\epsilon^{(1)} = \underset{\Theta_\epsilon \in \mathbb{S}_{++}^{p_2 \times p_2}}{\operatorname{argmin}} \left\{ \log \det \Theta_\epsilon - \operatorname{tr}(\widehat{S}^{(1)} \Theta_\epsilon) + \rho_n \|\Theta_\epsilon\|_{1, \text{off}} \right\},$$

where $\widehat{S}^{(1)}$ is the sample covariance matrix of $\widehat{E}^{(1)} \equiv Y - X \widehat{B}^{(1)}$.

- For iteration $k \geq 2$, while not converged:

• For $j = 1, \dots, p_2$, update B_j once by

$$\widehat{B}_j^{(k)} = \underset{B_j \in \mathcal{B}_j}{\operatorname{argmin}} \left\{ \frac{(\sigma_\epsilon^{jj})^{(k-1)}}{n} \|Y_j + r_j^{(k)} - X B_j\|_2^2 + \lambda_n \|B_j\|_1 \right\},$$

where

$$r_j^{(k)} = \frac{1}{(\sigma_\epsilon^{jj})^{(k-1)}} \left[\sum_{i=1}^{j-1} (\sigma_\epsilon^{ij})^{(k-1)} (Y_i - X \widehat{B}_i^{(k)}) + \sum_{i=j+1}^{p_2} (\sigma_\epsilon^{ij})^{(k-1)} (Y_i - X \widehat{B}_i^{(k-1)}) \right]. \quad (23)$$

• Update Θ_ϵ by

$$\widehat{\Theta}_\epsilon^{(k)} = \underset{\Theta_\epsilon \in \mathbb{S}_{++}^{p_2 \times p_2}}{\operatorname{argmin}} \left\{ \log \det \Theta_\epsilon - \operatorname{tr}(\widehat{S}^{(k)} \Theta_\epsilon) + \rho_n \|\Theta_\epsilon\|_{1, \text{off}} \right\},$$

where $\widehat{S}^{(k)}$ is defined similarly.

Intuitively, for the first iteration, we wait for the algorithm to complete the whole cyclic p_2 block-coordinate descent step, as the first iteration usually achieves a big improvement in the value of the objective function compared to the initialization values, as depicted in

Figure 2. However, in subsequent iterations, the changes in the objective function become relatively small, so we do $(p_2 + 1)$ successive block-updates in every iteration, and start to update Θ_ϵ once a full p_2 block update in B is completed, instead of waiting for the update in B proceeds cyclically until convergence. In practice, this way of updating B and Θ_ϵ leads to faster convergence in terms of total computing time, yet yields the same estimates compared with the exact 2-block update shown in Section 2.

Parallelization. A number of steps of the MLE algorithm is parallelizable. In the screening step, when applying the de-biased Lasso procedure (Javanmard and Montanari, 2014) to obtain the p -values, we need to implement p_2 separate regressions, which can be distributed to different compute nodes and carried out in parallel. So does the refitting step, in which we refit each column in B in parallel.

Moreover, according to Bradley et al. (2011); Richtárik and Takáč (2012); Scherrer et al. (2012) and a series of similar studies, though the block update in the alternating search step is supposed to be carried out sequentially, we can implement the update in parallel to speed up convergence, yet empirically yield identical estimates. This parallelization can be applied to either the minimization with respect to B within the 2-block update method, or the minimization with respect to each column of B for the $(p_2 + 1)$ -block update method. Either way, $r_j^{(k)}$ in (23) is substituted by

$$r_{j, \text{parallel}}^{(k)} = \frac{1}{(\sigma_\epsilon^{jj})^{(k-1)}} \sum_{i \neq j}^{p_2} (\sigma_\epsilon^{ij})^{(k-1)} (Y_i - X \widehat{B}_i^{(k-1)}),$$

which is not updated until we have updated B_j 's once for all $j = 1, \dots, p_2$ in parallel.

Table 10 shows the elapsed time for carrying out our proposed algorithm using 2-block/ $(p_2 + 1)$ -block update with/without parallelization, under the simulation setting where we have $p_1 = p_2 = 200, n = 150$. The screening step and refitting step are both carried out in parallel for all four different implementations.⁴

elapsed time (sec)	2-block	$(p_2 + 1)$ -block	2-block in parallel	$(p_2 + 1)$ -block in parallel
	5074	2556	848	763

Table 10: Computing time with different update methods

As shown in the table, using $(p_2 + 1)$ -block update and parallelization both can speed up convergence and reduce computing time, which takes only 1/7 of the computing time compared with using 2-block update without parallelization.

Remark 11. The total computing time depends largely on the number of bootstrapped samples we choose for the stability selection step. For the above displayed results, we used 50 bootstrapped samples to obtain the weight matrix. Nevertheless, one can select the number of bootstrap samples judiciously and reduce them if performance would not be seriously impacted.

⁴ For parallelization, we distribute the computation on 8 cores.

5. Discussion.

In this paper, we examined multi-layered Gaussian networks, proposed a provably converging algorithm for obtaining the estimates of the key model parameters and established their theoretical properties in high-dimensional settings. Note that we focused on ℓ_1 penalties for both the directed and undirected edges, since it was assumed that the multi-layer network was sparse both between layers and within layers. In many scientific applications, external information may require imposing group penalties, primarily on the directed edge parameters (B). For example, in a gene-protein 2-layer network, genes can be grouped according to their function in pathways and one may be interested in assessing the pathway's impact on proteins. In that case, a group lasso penalty can be imposed. In general, the proposed framework can easily accommodate other types of penalties in accordance to the underlying data generating procedure. The exact form of the error bounds established would be different, depending on the exact choice of penalty selected. Nevertheless, as long as the penalty is convex, all arguments regarding bi-convexity and convergence follow, and we can use similar strategies to bound the statistical error of the estimators, obtained via the iterative algorithm.

Next, we discuss connections of this work to that in [Sohn and Kim \(2012\)](#); [Yuan and Zhang \(2014\)](#); [McCarter and Kim \(2014\)](#). In these papers, an alternative parameterization of the 2-layer network is adopted. Specifically, all nodes in layers 1 and 2 are considered jointly and assumed to be drawn from the following Gaussian distribution:

$$\begin{pmatrix} \mathbf{X} \\ \mathbf{Y} \end{pmatrix} \sim \mathcal{N} \left(0, \begin{pmatrix} \Omega_X & \Omega_{XY} \\ \Omega_{YX} & \Omega_Y \end{pmatrix}^{-1} \right),$$

and by conditioning \mathbf{Y} on \mathbf{X} , one obtains

$$\mathbf{Y}|\mathbf{X} \sim \mathcal{N}(-\Omega_Y^{-1}\Omega_{XY}\mathbf{X}, \Omega_Y^{-1}). \quad (24)$$

Compare [\(24\)](#) with our model set-up in [Section 2.1](#), the following correspondence holds:

$$B = -\Omega_{XY}\Omega_Y^{-1}, \quad \Omega_Y = \Theta_e. \quad (25)$$

Note that the correspondence in [\(25\)](#) is only guaranteed to hold in selective settings. Specifically: at the population level, the correspondence between (Ω_{XY}, Ω_Y) and (B, Θ_e) holds in the absence of any sparsity penalization. Further, in a low-dimensional data setting without penalty terms in the objective function, the estimates from the two parameterizations would be similar provided that the problem is well-conditioned and the sample size reasonably large.

However, the situation is different in high-dimensional settings and in the presence of sparsity penalties. Specifically, given data X and Y , instead of parameterizing the model in terms of (B, Θ_e) , the authors in [Sohn and Kim \(2012\)](#); [Yuan and Zhang \(2014\)](#); [McCarter and Kim \(2014\)](#) consider the following optimization problem, parameterized in (Ω_{XY}, Ω_Y) :

$$\min_{\Omega_{XY}, \Omega_Y} g(\Omega_{XY}, \Omega_Y) \equiv g_0(\Omega_{XY}, \Omega_Y) + \mathcal{R}(\Omega_{XY}, \Omega_Y) \quad (26)$$

where $g_0(\Omega_{XY}, \Omega_Y) = -\log \det \Omega_Y + \frac{1}{n} \text{tr} \left[(Y + \Omega_{XY}\Omega_Y^{-1}X)' \Omega_Y (Y + \Omega_{XY}\Omega_Y^{-1}X) \right]$ is jointly convex in (Ω_{XY}, Ω_Y) , and $\mathcal{R}(\Omega_{XY}, \Omega_Y)$ is some regularization term. In particular, the

element-wise ℓ_1 norm on Ω_Y , and the element-wise ℓ_1 or column-wise ℓ_1 norm (matrix 2, 1 norm) on Ω_{XY} are the main penalties under consideration in those papers.

Despite the convex formulation in [\(26\)](#), we would like to point out that in general, the sparsity pattern in B and Ω_{XY} are not transferable through the regularization term, which induces a major difference between the formulation in [\(26\)](#) and the one presented in this paper. Given the correspondence in [\(25\)](#), there are two cases where B and Ω_{XY} share the same sparsity pattern: 1) Ω_Y (or Θ_e , equivalently) is diagonal, or 2) both the i^{th} row in B and Ω_{XY} are identically zero, for an arbitrary $i = 1, \dots, p_1$. However, both settings are fairly restrictive and unlike to occur in many applications.

Note that the linear model represents a natural modeling tool for a number of problems and the regression coefficients have a specific scientific interpretation. This is easily accomplished through the (B, Θ_e) -parameterization, by adding proper regularization to B (e.g., penalty which enforces element-wise sparsity or group-Lasso type of sparsity, etc) if necessary. However, with the (Ω_{XY}, Ω_Y) -parameterization, the underlying sparsity in the true data generating procedure, encoded by B , will not be easily incorporated, and to add a regularization term on Ω_{XY} may lose the scientific interpretability, and may also lead to an estimated B whose sparsity pattern is completely mis-specified, obtained from [\(25\)](#) with $\hat{\Omega}_{XY}, \hat{\Omega}_Y$ plugged in.

Another difference we would like to point out is that once we add penalty terms to the objective function in the low dimensional setting, or switch to the high dimensional setting (as considered in [Sohn and Kim \(2012\)](#) and [Yuan and Zhang \(2014\)](#)), the correspondence between the optimizer(s) of [\(1\)](#) and the optimizer(s) of [\(26\)](#) become difficult to connect analytically.

Acknowledgments

George Michaelidis was supported by NSF awards DMS-1228164 and DMS-1545277 and NIH award 7R21GM110171903. Mouninath Banerjee was supported by NSF award DMS-1308890.

Appendix A. Proofs for Main Theorems

Proof of Theorem 1. We initialize the algorithm at $(\widehat{B}^{(0)}, \widehat{\Theta}_\epsilon^{(0)}) \in \text{dom}(f)$. Then for all $k \geq 1$,

$$\widehat{B}^{(k)} = \underset{B}{\text{argmin}} f(B, \widehat{\Theta}_\epsilon^{(k-1)}), \quad (27)$$

$$\widehat{\Theta}_\epsilon^{(k)} = \underset{\Theta_\epsilon}{\text{argmin}} f(\widehat{B}^{(k)}, \Theta_\epsilon). \quad (28)$$

Now, consider a limit point $(B^\infty, \Theta_\epsilon^\infty)$ of the sequence $\{(\widehat{B}^{(k)}, \widehat{\Theta}_\epsilon^{(k)})\}_{k \geq 1}$. Note that such limit point exists by Bolzano-Weierstrass theorem since the sequence $\{(\widehat{B}^{(k)}, \widehat{\Theta}_\epsilon^{(k)})\}_{k \geq 1}$ is bounded. Consider a subsequence $\mathcal{K} \subseteq \{1, 2, \dots\}$ such that $(\widehat{B}^{(k)}, \widehat{\Theta}_\epsilon^{(k)})_{k \in \mathcal{K}}$ converges to $(B^\infty, \Theta_\epsilon^\infty)$. Now for the bounded sequence $\{(\widehat{B}^{(k+1)}, \widehat{\Theta}_\epsilon^{(k)})\}_{k \in \mathcal{K}}$, without loss of generality,⁵ we can say that

$$\{(\widehat{B}^{(k+1)}, \widehat{\Theta}_\epsilon^{(k)})\}_{k \in \mathcal{K}} \rightarrow (\widetilde{B}^\infty, \widetilde{\Theta}_\epsilon^\infty), \quad \text{for some } (\widetilde{B}^\infty, \widetilde{\Theta}_\epsilon^\infty) \in \text{dom}(f).$$

By (27) it follows immediately that $\widetilde{\Theta}_\epsilon^\infty = \Theta_\epsilon^\infty$. Also, the following inequality holds:

$$f(\widehat{B}^{(k+1)}, \widehat{\Theta}_\epsilon^{(k+1)}) \leq f(\widehat{B}^{(k+1)}, \widehat{\Theta}_\epsilon^{(k)}) \leq f(\widehat{B}^{(k)}, \widehat{\Theta}_\epsilon^{(k)}).$$

Thus, by letting $k \rightarrow \infty$ over \mathcal{K} , we have

$$f(B^\infty, \Theta_\epsilon^\infty) \leq f(\widetilde{B}^\infty, \Theta_\epsilon^\infty) \leq f(B^\infty, \Theta_\epsilon^\infty),$$

since f is continuous. This implies that

$$f(\widetilde{B}^\infty, \Theta_\epsilon^\infty) = f(B^\infty, \Theta_\epsilon^\infty). \quad (29)$$

Next, since $f(\widehat{B}^{(k+1)}, \widehat{\Theta}_\epsilon^{(k)}) \leq f(B, \widehat{\Theta}_\epsilon^{(k)})$, for all $B \in \mathbb{R}^{p_1 \times p_2}$, let k grow along \mathcal{K} , and we obtain the following:

$$f(\widetilde{B}^\infty, \Theta_\epsilon^\infty) \leq f(B, \Theta_\epsilon^\infty), \quad \forall B \in \mathbb{R}^{p_1 \times p_2}.$$

It then follows from (29) that

$$f(B^\infty, \Theta_\epsilon^\infty) \leq f(B, \Theta_\epsilon^\infty), \quad \forall B \in \mathbb{R}^{p_1 \times p_2}. \quad (30)$$

Finally, note that $f(\widehat{B}^{(k)}, \widehat{\Theta}_\epsilon^{(k)}) \leq f(\widehat{B}^{(k)}, \Theta_\epsilon)$, for all $\Theta \in \mathbb{S}_{++}^{p_2 \times p_2}$. As before, let k grow along \mathcal{K} and with the continuity of f , we obtain:

$$f(B^\infty, \Theta_\epsilon^\infty) \leq f(B^\infty, \Theta_\epsilon), \quad \forall \Theta_\epsilon \in \mathbb{S}_{++}^{p_2 \times p_2}. \quad (31)$$

Now, (30) and (31) together imply that $(B^\infty, \Theta_\epsilon^\infty)$ is a coordinate-wise minimum of f and by Fact 1, also a stationary point of f . \square

⁵ switching to some further subsequence of \mathcal{K} if necessary.

Proof of Theorem 2. The statement of Theorem 2 is a variation of Proposition 4.1 in Basu and Michailidis (2015), and its proof follows directly from the proof of the proposition in Basu and Michailidis (2015, Appendix B). We only outline how the statement differs. In the original statement of Proposition 4.1 in Basu and Michailidis (2015), the authors provide the error bound for $\widehat{\beta}$, obtained as per (14) whose dimension is qp^2 with q denoting the true lag of the vector-autoregressive process, under an RE condition for $\bar{\Gamma}$ and a deviation bound for $(\widehat{\gamma}, \widehat{\Gamma})$. For our problem, we impose a similar RE condition on $\widehat{\Gamma}$ and deviation bound on $(\widehat{\gamma}, \widehat{\Gamma})$, so as to yield a bound on $\widehat{\beta}$ that lies in a $p_1 p_2$ -dimensional space. \square

Proof of Theorem 3. The statement of this theorem is a variation of Theorem 1 in Ravikumar et al. (2011), so here, instead of providing a complete proof of the theorem, we only outline how the estimation problem differs in our setting, as well as the required changes in its proof.

In Ravikumar et al. (2011), the authors consider the optimization problem in (15), and show that for a random realization, with certain sample size requirement and choice of the regularization parameter, the following bound for $\widehat{\Theta}_\epsilon$ holds with probability at least $1 - 1/p_2^2$ for some $\tau > 2$:

$$\|\widehat{\Theta}_\epsilon - \Theta_\epsilon^*\|_\infty \leq \{2(1 + 8\xi^{-1})\kappa_{H^*}\} \bar{\delta}_f(p_2^*, n), \quad (32)$$

where $\bar{\delta}(r, n)$ is defined as

$$\bar{\delta}(r, n) := 8(1 + 4\sigma^2) \max_i(\Sigma_{\epsilon, ii}^*) \sqrt{\frac{2 \log(4r)}{n}}. \quad (33)$$

The quantity $\bar{\delta}(p_2^*, n)$ that shows up in expression (32) is the bound for $\|S - \Sigma_\epsilon^*\|_\infty \equiv \|\widehat{\Sigma}_\epsilon - \Sigma_\epsilon^*\|_\infty$. In particular, in Lemma 8 (Ravikumar et al., 2011), they show that with probability at least $1 - 1/p_2^2$, $\tau > 2$, the following bound holds:

$$\|S - \Sigma_\epsilon^*\|_\infty \leq \bar{\delta}(p_2^*, n).$$

In our optimization problem (13), we are using \widehat{S} instead of S , hence a bound for $\|\widehat{S} - \Sigma_\epsilon^*\|_\infty$ is necessary, and the remaining argument in the proof of Theorem 1 (Ravikumar et al., 2011) will follow through.

Therefore in our theorem statement, we use $g(\nu/\beta)$ as a bound for $\|\widehat{S} - \Sigma_\epsilon^*\|_\infty$ then yield the bound for $\|\widehat{\Theta}_\epsilon - \Theta_\epsilon^*\|_\infty$, since we are using the surrogate error $\widehat{E} = Y - X\widehat{B}$ in the estimation, instead of the true error E . \square

Proof of Theorem 4. We first consider part (I) of the theorem. Note that by (5), $\widehat{\beta}^{(0)}$ can be equivalently written as

$$\widehat{\beta}^{(0)} \equiv \underset{\beta \in \mathbb{R}^{p_1 \times p_2}}{\text{argmin}} \{-2\beta\gamma^0 + \beta\Gamma^0\beta + \lambda_n^0 \|\beta\|_1\}, \quad (34)$$

where

$$\Gamma^{(0)} = \mathbf{I} \otimes \frac{X'X}{n}, \quad \gamma^{(0)} = (\mathbf{I} \otimes X') \text{vec} Y/n.$$

Consider the following events:

$$\mathbf{E1.} \left\{ \frac{X'X}{n} \sim RE(\varphi^*, \phi^*) \right\},$$

$$\mathbf{E2.} \quad \left\{ \frac{1}{n} \|X'E\|_\infty \leq c_2 [\Lambda_{\max}(\Sigma_X^*) \Lambda_{\max}(\Sigma_\epsilon^*)]^{1/2} \sqrt{\frac{\log(p_1 p_2)}{n}} \right\}.$$

Note that $\mathbf{E1} \cap \mathbf{E2}$ implies the following events:

$$\Gamma^{(0)} \equiv \mathbf{I} \otimes \frac{X'X}{n} \sim RE(\varphi^*, \phi^*), \quad \text{where } \varphi^* = \Lambda_{\min}(\Sigma_X^*),$$

and

$$\|\gamma^{(0)} - \Gamma^{(0)}\beta^*\|_\infty = \frac{1}{n} \|X'E\|_\infty \leq c_2 [\Lambda_{\max}(\Sigma_X^*) \Lambda_{\max}(\Sigma_\epsilon^*)]^{1/2} \sqrt{\frac{\log(p_1 p_2)}{n}}. \quad (35)$$

Hence, by Proposition 4.1 of [Basu and Michaelidis \(2015\)](#), the bound [\(21\)](#) holds on $\mathbf{E1} \cap \mathbf{E2}$.

By Lemmas [1](#) and [2](#), $\mathbb{P}(\mathbf{E1})$ is at least $1 - 2 \exp(-c_3 n)$, for some $c_3 > 0$. By Lemma [3](#), $\mathbb{P}(\mathbf{E2})$ is at least $1 - 6c_1 \exp[-(c_2^2 - 1) \log(p_1 p_2)]$ for some $c_1 > 0$, $c_2 > 1$. Hence, with probability at least

$$\mathbb{P}(\mathbf{E1} \cap \mathbf{E2}) \geq 1 - \mathbb{P}(\mathbf{E1}^c) - \mathbb{P}(\mathbf{E2}^c),$$

the bound in [\(21\)](#) holds, which proves the first part of [\(1\)](#). In particular, we have $\|\hat{\beta}^0 - \beta^*\|_1 \leq \nu_\beta^{(0)} \sim O(\sqrt{\log(p_1 p_2)/n})$ on $\mathbf{E1} \cap \mathbf{E2}$.

To prove the second part of [\(1\)](#), note that by Theorem [3](#) the bound in [\(22\)](#) holds when $\mathbf{B1}$ - $\mathbf{B3}$ are satisfied. Now, from the argument above, $\mathbf{B1}$ holds on the event $\mathbf{E1} \cap \mathbf{E2}$. Also, from the proof of Proposition [3](#), $\mathbf{B2}$ is satisfied, i.e.,

$$\|\widehat{\Sigma}^{(0)} - \Sigma_\epsilon^*\|_\infty \leq g(\nu_\beta^{(0)}), \quad \text{where } \widehat{\Sigma}^{(0)} = \frac{1}{n} (Y - X\widehat{B}^{(0)})'(Y - X\widehat{B}^{(0)}), \quad (36)$$

on $\mathbf{E1} \cap \mathbf{E2} \cap \mathbf{E3} \cap \mathbf{E4}$, where the events $\mathbf{E3}$ and $\mathbf{E4}$ are given by:

$$\mathbf{E3.} \quad \left\{ \left\| \frac{E'E}{n} - \Sigma_\epsilon^* \right\|_\infty \leq \sqrt{\frac{\log(4+\tau) \log p_1}{c_\tau^2 n}} \right\} \text{ for some } \tau_2 > 2 \text{ and } c_\tau^* > 0 \text{ that depends on } \Sigma_\epsilon^*,$$

$$\mathbf{E4.} \quad \left\{ \left\| \frac{X'X}{n} - \Sigma_X^* \right\|_\infty \leq \sqrt{\frac{\log(4+\tau) \log p_1}{c_\tau^2 n}} \right\} \text{ for some } \tau_1 > 2 \text{ and } c_X^* > 0 \text{ that depends on } \Sigma_X^*.$$

Therefore, the probability of the bound for $\widehat{\Theta}_\epsilon^{(0)}$ in [\(22\)](#) to hold is at least

$$\mathbb{P}(\mathbf{E1} \cap \mathbf{E2} \cap \mathbf{E3} \cap \mathbf{E4}), \quad (37)$$

By Lemma [2](#), Lemma [3](#) and the proof of Proposition [3](#), the probability in [\(37\)](#) is lower bounded by:

$$1 - 2 \exp(-c_3 n) - 6c_1 \exp[-(c_2^2 - 1) \log(p_1 p_2)] - 1/p_1^{\tau_1 - 2} - 1/p_2^{\tau_2 - 2}.$$

Consider the following two cases where the relative order of p_1 and p_2 differ. Case 1: $p_1 \prec p_2$, then $\nu_\Theta^{(0)} \sim O(\sqrt{\log p_2/n})$; case 2: $p_1 \gtrsim p_2$, then $\nu_\Theta^{(0)} \sim O(\log(p_1 p_2)/n)$. In either case, since we are assuming $\log(p_1 p_2)/n$ to be a small quantity and it follows that $\sqrt{\log(p_1 p_2)/n} \gtrsim \log(p_1 p_2)/n$, the following bound always holds:

$$\nu_\Theta^{(0)} \leq C\Theta \sqrt{\frac{\log(p_1 p_2)}{n}} \equiv M\Theta,$$

where $C\Theta$ is some large fixed constant that bounds the constant terms in front of $\sqrt{\log(p_1 p_2)/n}$. Now we consider part [\(II\)](#) of the theorem. Note that for each $k \geq 1$, $\widehat{\beta}^{(k)}$ and $\widehat{\Theta}_\epsilon^{(k)}$ are obtained via solving the following two optimizations:

$$\widehat{\beta}^{(k)} = \underset{\beta \in \mathbb{R}^{p_1 \times p_2}}{\operatorname{argmin}} \left\{ -2\beta' \widehat{\gamma}^{(k-1)} + \beta' \widehat{\Gamma}^{(k-1)} \beta + \lambda_n \|\beta\|_1 \right\}, \quad (38)$$

$$\widehat{\Theta}_\epsilon^{(k)} = \underset{\Theta_\epsilon \in \mathbb{S}_+^{p_2 \times p_2}}{\operatorname{argmin}} \left\{ \log \det \Theta_\epsilon - \operatorname{tr}(\widehat{\Sigma}^{(k)} \Theta_\epsilon) + p_n \|\Theta_\epsilon\|_{1,\text{off}} \right\}, \quad (39)$$

where

$$\widehat{\gamma}^{(k)} = \widehat{\Theta}^{(k)} \otimes \frac{X'Y}{n}, \quad \widehat{\Gamma}^{(k)} = \widehat{\Theta}^{(k)} \otimes \frac{X'X}{n}, \quad \widehat{\Sigma}^{(k)} = \frac{1}{n} (Y - X\widehat{B}^{(k)})'(Y - X\widehat{B}^{(k)}).$$

Consider the bound on $\widehat{\beta}^{(k)}$ for $k = 1$. The argument is similar to that of $\widehat{\beta}^{(0)}$, with appropriate modifications to account for the fact that the objective function now involves log likelihood instead of least squares. Formally, we consider the event $\mathbf{E1} \cap \mathbf{E2} \cap \mathbf{E3} \cap \mathbf{E4} \cap \mathbf{E5}$, where

$$\mathbf{E5.} \quad \left\{ \frac{1}{n} \|X'E\Theta_\epsilon^*\|_\infty \leq c_2 \left[\frac{\Lambda_{\max}(\Sigma_X^*)}{\Lambda_{\min}(\Sigma_\epsilon^*)} \right]^{1/2} \sqrt{\frac{\log(p_1 p_2)}{n}} \right\}.$$

Note that $\{\|\widehat{\Theta}_\epsilon^{(0)} - \Theta_\epsilon^*\|_\infty \leq \nu_\Theta^{(0)}\}$ holds on this event. By Lemma [3](#), $\mathbb{P}(\mathbf{E5}) \geq 1 - 6c_1 \exp[-(c_2^2 - 1) \log(p_1 p_2)]$. Combining this with the lower bound on [\(37\)](#) and the sample size requirement (note this sample size requirement can be relaxed to $n \gtrsim \log(p_1 p_2)$ if $p_1 \prec p_2$), we obtain that with probability at least

$$1 - 1/p_1^{\tau_1 - 2} - 1/p_2^{\tau_2 - 2} - 12c_1 \exp[-(c_2^2 - 1) \log(p_1 p_2)] - 2 \exp[-c_3 n],$$

the following three events hold simultaneously:

$$\mathbf{A1}^* \quad \|\widehat{\Theta}_\epsilon^{(0)} - \Theta_\epsilon^*\|_\infty \leq \nu_\Theta^{(0)} \lesssim O(\sqrt{\log(p_1 p_2)/n});$$

$$\mathbf{A2}^* \quad \widehat{\Gamma}^{(0)} \sim RE(\varphi^{(0)}, \phi^{(0)}) \text{ where}$$

$$\varphi^{(0)} \geq \frac{\Lambda_{\min}(\Sigma_X^*)}{2} (\min_i \psi^i - dM\Theta) \quad \text{and} \quad \phi^{(0)} \leq \frac{\log p_1}{n} \frac{\Lambda_{\min}(\Sigma_X^*)}{2} (\max_j \psi^j + dM\Theta);$$

$$\mathbf{A3}^* \quad \|\widehat{\gamma}^{(0)} - \widehat{\Gamma}^{(0)}\beta^*\|_\infty \leq \mathbb{Q}(\nu_\Theta^{(0)}) \sqrt{\frac{\log(p_1 p_2)}{n}} \quad \text{with the expression for } \mathbb{Q}(\cdot) \text{ given in } \text{a href="#">(16)}.$$

By Theorem [2](#), by choosing $\lambda_n \geq 4\mathbb{Q}(M\Theta) \sqrt{\frac{\log(p_1 p_2)}{n}}$, the following bound holds:

$$\|\widehat{\beta}^{(1)} - \beta^*\|_1 \leq 64s^{**} \lambda_n / \varphi^{(0)}. \quad (40)$$

The error bound for $\widehat{\Theta}_\epsilon^{(1)}$ can now be established using the same argument for $\widehat{\Theta}_\epsilon^{(0)}$, with the only difference that now we consider the event $\mathbf{E1} \cap \dots \cap \mathbf{E5}$ instead of $\mathbf{E1} \cap \dots \cap \mathbf{E4}$ and use [\(40\)](#) instead of [\(21\)](#).

Note that an upper bound for the leading term of the right hand side of (40) is at most of the order $O(\sqrt{\log(p_1 p_2)/n})$, and can be written as

$$C_\beta \left(s^{**} \sqrt{\frac{\log(p_1 p_2)}{n}} \right) \equiv M_\beta,$$

with C_β being some potentially large number that bounds the constant term. Notice that M_β is of the same order as $\nu_\beta^{(0)}$; thus, for $\widehat{\Theta}_\epsilon^{(1)}$, we can also achieve the following bound:

$$\|\widehat{\Theta}_\epsilon^{(1)} - \Theta_\epsilon^*\|_\infty \leq M_\Theta,$$

with high probability since we are assuming C_Θ to be some potentially large number.

Note that the events **E1**, ..., **E5** rely only on the parameters and not on the estimated quantities, and on their intersection we have uniform upper bounds on the errors of $\widehat{\beta}^{(k)}$ and $\widehat{\Theta}_\epsilon^k$ for $k = 0, 1$. Hence the error bounds for $k = 1$ can be used to invoke Theorems 2 and 3 inductively on realizations X and E from the set $\mathbf{E1} \cap \dots \cap \mathbf{E5}$ to provide high probability error bounds for all subsequent iterates as well. This leads to the uniform error bounds of part (II) with the desired probability. \square

Proof of Theorem 5. First, we note that with a Bonferroni correction, the family-wise type I error will be automatically controlled at level α . Hence, we will focus on the power of the screening step. Also, from Theorem 7 of Javanmard and Montanari (2014), it is easy to see that all the arguments below hold for a large set of random realizations of X , whose probability approaches 1 under the specified asymptotic regime when the eigenvalues of Σ_X are bounded away from 0 and infinity.

Let $B^* = [B_1^* \dots B_{p_2}^*]$ denote the true value of the regression coefficients and $\widehat{B}_j, j = 1, \dots, p_2$ denote the estimates given by the de-biased Lasso procedure in Javanmard and Montanari (2014). With the given level for sparsity, by Theorem 8 in Javanmard and Montanari (2014), each \widehat{B}_j satisfies the following:

$$\sqrt{n}(\widehat{B}_j - B_j^*) = Z + \Delta,$$

where $Z \sim \mathcal{N}(0, \sigma^2 M_j \widehat{\Sigma}_X M_j')$ and Δ vanishes asymptotically. Here $\widehat{\Sigma}_X$ is the sample covariance matrix of the predictors X , σ is the population noise level of the error term ϵ_j , and M_j is the matrix corresponding to the j th regression, produced by the procedure described in Javanmard and Montanari (2014)⁶. Let $\widehat{B}_{j,i}$ denote the i th coordinate of the j th regression coefficient vector \widehat{B}_j and $\widehat{\Sigma}_j$ be the covariance matrix of the estimator \widehat{B}_j , then

$$\widehat{\Sigma}_j = \frac{\sigma^2}{n} M_j \widehat{\Sigma}_X M_j',$$

and in particular, the variance of $\widehat{B}_{j,i}$ is $\widehat{\Sigma}_{j,ii} := \sigma_{\epsilon_i}^2$. Using these notations, for a prespecified level α , the test statistics for testing $H_0^i : B_{j,i}^* = 0$ vs. $H_A^i : B_{j,i}^* \neq 0$, for all $i = 1, \dots, p_2$ are

6. Details of the procedure is described in p.2871 in Javanmard and Montanari (2014), with M being an intermediate quantity obtained by solving an optimization problem.

$1, \dots, p_1; j = 1, \dots, p_2$ can be equivalently written as

$$\widehat{T}_{j,i} = \begin{cases} 1 & \text{if } |\widehat{B}_{j,i}|/\widehat{\sigma}_{ii}^j > z_{\alpha/(2p_1 p_2)}, \\ 0 & \text{otherwise.} \end{cases}$$

where z_α denotes the upper α quantiles of $\mathcal{N}(0, 1)$.

Define the ‘‘family-wise’’ power as follows:

$$\begin{aligned} \mathbb{P}(\text{all true alternatives are detected}) &= \mathbb{P}\left(\bigcap_{1 \leq j \leq p_2} \bigcap_{k \in S_j^*} \{\widehat{T}_{j,k} = 1\}\right) \\ &= 1 - \mathbb{P}\left(\bigcup_{1 \leq j \leq p_2} \bigcup_{k \in S_j^*} \{\widehat{T}_{j,k} = 0\}\right). \end{aligned}$$

Correspondingly, the family-wise type II error can be written as

$$\mathbb{P}\left(\bigcup_{1 \leq j \leq p_2} \bigcup_{k \in S_j^*} \{\widehat{T}_{j,k} = 0\}\right) \leq \sum_{j=1}^{p_2} \sum_{k \in S_j^*} \mathbb{P}(\widehat{T}_{j,k} = 0). \quad (41)$$

By Theorem 16 in Javanmard and Montanari (2014), asymptotically, $\forall k \in S_j, j = 1, \dots, p_2$,

$$\mathbb{P}(\widehat{T}_{j,k} = 0) \leq 1 - G\left(\frac{\alpha}{p_1 p_2}, \frac{\sqrt{n}\gamma}{\sigma[\widehat{\Sigma}_{k,k}^{-1}]^{1/2}}\right); \quad 0 < \gamma \leq \min |B_{j,k}^*|, \quad \forall k \in S_j, j = 1, \dots, p_2. \quad (42)$$

Here

$$G(\alpha, u) \equiv 2 - \mathbb{P}(\Phi < z_{\alpha/2} + u) - \mathbb{P}(\Phi < z_{\alpha/2} - u),$$

where we use Φ to denote the random variable following a standard Gaussian distribution and the choice of n in (42) doesn't depend on k . Hence, (42) can be rewritten as

$$\begin{aligned} \mathbb{P}(\widehat{T}_{j,k} = 0) &\leq 1 - G\left(\frac{\alpha}{p_1 p_2}, \frac{\sqrt{n}\gamma}{\sigma[\widehat{\Sigma}_{k,k}^{-1}]^{1/2}}\right) \\ &= \mathbb{P}\left(\Phi < z_{\alpha/(2p_1 p_2)} - \frac{\sqrt{n}\gamma}{\sigma[\widehat{\Sigma}_{k,k}^{-1}]^{1/2}}\right) - \mathbb{P}\left(\Phi > z_{\alpha/(2p_1 p_2)} + \frac{\sqrt{n}\gamma}{\sigma[\widehat{\Sigma}_{k,k}^{-1}]^{1/2}}\right) \\ &\leq \mathbb{P}\left(\Phi > \frac{\sqrt{n}\gamma}{\sigma[\widehat{\Sigma}_{k,k}^{-1}]^{1/2}} - z_{\alpha/(2p_1 p_2)}\right), \end{aligned} \quad (43)$$

where we use Φ to denote the random variable following a standard Gaussian distribution. Note that the following inequality holds for standard Normal percentiles:

$$2e^{-t^2} \leq \mathbb{P}(|\Phi| > t) \leq e^{-t^2/2},$$

and by taking the inverse function, the following inequality holds:

$$\sqrt{-\log \frac{y}{2}} \leq z_{y/2} \leq \sqrt{-2 \log y}.$$

Letting $y = \frac{\alpha}{p_1 p_2}$, it follows that

$$\left(-\log \frac{\alpha}{2p_1 p_2}\right)^{1/2} \leq z_{\alpha/(2p_1 p_2)} \leq \left(-2 \log \frac{\alpha}{p_1 p_2}\right)^{1/2},$$

hence

$$\mathbb{P}\left(\Phi > \frac{\sqrt{n}\gamma}{\sigma[\Sigma_{k,k}^{-1}]^{1/2}} - z_{\alpha/(2p_1 p_2)}\right) \leq \mathbb{P}\left(\Phi > \frac{\sqrt{n}\gamma}{\sigma[\Sigma_{k,k}^{-1}]^{1/2}} - \sqrt{-2 \log \frac{\alpha}{p_1 p_2}}\right).$$

Now given

$$\frac{\log(p_1 p_2)}{n} \rightarrow 0,$$

it follows that

$$\frac{\sqrt{2 \log \frac{p_1 p_2}{\alpha}}}{\sqrt{n}/\sigma[\Sigma_{k,k}^{-1}]^{1/2}} \rightarrow 0,$$

indicating that for sufficiently large n , the following lower bound holds for some constant $c_0 > 0$:

$$\left(\frac{\sqrt{n}\gamma}{\sigma[\Sigma_{k,k}^{-1}]^{1/2}} - \sqrt{-2 \log \frac{\alpha}{p_1 p_2}}\right) \geq c_0 \sqrt{n}.$$

Note that c_0 is universal for all choices of k , since this lower bound can be achieved by substituting $\Sigma_{k,k}^{-1}$ by $(1/\Lambda_{\min}(\Sigma_X))$, which is assumed to be bounded away from infinity.

Combined with the fact that $\mathbb{P}(\Phi > t) \leq e^{-t^2/2}$, the last expression in (43) can thus be bounded by

$$\mathbb{P}\left(\Phi > \frac{\sqrt{n}\gamma}{\sigma[\Sigma_{k,k}^{-1}]^{1/2}} - z_{\alpha/(2p_1 p_2)}\right) \leq \exp\left[-\frac{1}{2}\left(\frac{\sqrt{n}\gamma}{\sigma[\Sigma_{k,k}^{-1}]^{1/2}} - \sqrt{-2 \log \frac{\alpha}{p_1 p_2}}\right)^2\right] \leq e^{-c_1 n}, \quad (44)$$

for some universal constant $c_1 > 0$, and the bound in (44) holds uniformly for all $k \in S_j, \forall j$. Combine (41), (42) and (44), it follows that

$$\mathbb{P}\left(\bigcup_{1 \leq j \leq p_2} \bigcup_{k \in S_j^*} \{\hat{T}_{j,k} = 0\}\right) \leq s^* p_2 \exp(-c_1 n). \quad (45)$$

Now with $\log(p_1 p_2)/n = o(1)$ and the given sparsity level, that is, $s^* = o(\sqrt{n}/\log p_1)$, it follows that

$$s^* p_2 \exp(-c_1 n) = o(1),$$

and by (45), we have:

$$\mathbb{P}(\text{family-wise type II error}) \rightarrow 0, \quad \Leftrightarrow \quad \mathbb{P}(\text{family-wise power}) \rightarrow 1.$$

This is equivalent to establishing that, given $\log(p_1 p_2)/n \rightarrow 0$, the screening step recovers the true support sets S_j^* for all $j = 1, 2, \dots, p_2$ with high probability, while keeping the family-wise type I error rate under control. \square

Appendix B. Proofs for Propositions and Auxiliary Lemmas

In this subsection, we provide proofs for the propositions presented in Section 3, which requires several auxiliary lemmas, whose proofs are presented along the context.

To prove Proposition 1, we need the following two lemmas. Lemma 1 was originally provided as Lemma B.1 in Basu and Michailidis (2015), which states that if the sample covariance matrix of X satisfies the RE condition and Θ is diagonally dominant, then $(X'X/n) \otimes \Theta$ also satisfies the RE condition. Here we omit its proof and only state the main result. Lemma 2 verifies that with high probability, the sample covariance matrix of the design matrix X satisfies the RE condition.

Lemma 1. *If $X'X/n \sim RE(\varphi^*, \phi^*)$, and Θ is diagonally dominant, that is, $\psi^i := \sigma^{ii} - \sum_{j \neq i} \sigma^{ij} > 0$ for all $i = 1, 2, \dots, p_2$, where σ^{ij} is the ij th entry in Θ , then*

$$\Theta \otimes X'X/n \sim RE\left(\varphi^* \min_i \psi^i, \phi^* \max_i \psi^i\right).$$

Lemma 2. *With probability at least $1 - 2 \exp(-c_3 n)$, for a zero-mean sub-Gaussian random design matrix $X \in \mathbb{R}^{n \times p_1}$, its sample covariance matrix $\widehat{\Sigma}_X$ satisfies the RE condition with parameter φ^* and ϕ^* , i.e.,*

$$\widehat{\Sigma}_X \sim RE(\varphi^*, \phi^*), \quad (46)$$

where $\widehat{\Sigma}_X = X'X/n$, $\varphi^* = \Lambda_{\min}(\Sigma_X^*)/2$, $\phi^* = \varphi^* \log p_1/n$.

Proof. To prove this lemma, we first use Lemma 15 in Loh and Wainwright (2012), which states that if $X \in \mathbb{R}^{n \times p}$ is zero-mean sub-Gaussian with parameter (Σ, σ^2) , then there exists a universal constant $c > 0$ such that

$$\mathbb{P}\left(\sup_{v \in \mathbb{K}(2s)} \left| \frac{\|Xv\|_2^2}{n} - \mathbb{E}\left[\frac{\|Xv\|_2^2}{n}\right] \right| \geq t\right) \leq 2 \exp\left(-cn \min\left(\frac{t^2}{\sigma^4}, \frac{t}{\sigma^2}\right) + 2s \log p\right), \quad (47)$$

where $\mathbb{K}(2s)$ is a set of $2s$ sparse vectors, defined as

$$\mathbb{K}(2s) := \{v \in \mathbb{R}^p : \|v\| \leq 1, \|v\|_0 \leq 2s\}.$$

By taking $t = \frac{\Lambda_{\min}(\Sigma_X^*)}{54}$, with probability at least $1 - 2 \exp(-c'n + 2s \log p_1)$ for some $c' > 0$, the following bound holds:

$$|v'(\widehat{\Sigma}_X - \Sigma_X^*)v| \leq \frac{\Lambda_{\min}(\Sigma_X^*)}{54}, \quad \forall v \in \mathbb{K}(2s). \quad (48)$$

Then applying supplementary Lemma 13 in [Loh and Wainwright \(2012\)](#), for an estimator $\hat{\Sigma}_X$ of Σ_X^* satisfying the deviation condition in (48), the following RE condition holds:

$$v^* S_{v^*} v \geq \frac{\Lambda_{\min}(\hat{\Sigma}_X^*)}{2} \|v\|_2^2 - \frac{\Lambda_{\min}(\Sigma_X^*)}{2s} \|v\|_1^2.$$

Finally, set $s = c' n_i / 4 \log p_1$, then with probability at least $1 - 2 \exp(-c_3 n)$ ($c_3 > 0$), $\hat{\Sigma}_X \sim RE(\varphi^*, \phi^*)$ with $\varphi^* = \Lambda_{\min}(\Sigma_X^*)/2$, $\phi^* = \varphi^* \log p_1/n$. \square

With the above two lemmas, we are ready to prove [Proposition 1](#).

Proof of [Proposition 1](#). We first show that if Θ_ϵ^* is diagonally dominant, then $\hat{\Theta}_\epsilon$ is also diagonally dominant provided that the error of $\hat{\Theta}_\epsilon$ is of the given order and n is sufficiently large. Define

$$\tilde{\psi}^i = \hat{\sigma}_\epsilon^{ii} - \sum_{j \neq i} \hat{\sigma}_\epsilon^{ij},$$

where $\hat{\sigma}_\epsilon^{ij}$ is the ij th entry of $\hat{\Theta}_\epsilon$, then $\tilde{\psi}^i$ is the gap between the diagonal entry and the off-diagonal entries of row i in matrix $\hat{\Theta}_\epsilon$. We can decompose $\tilde{\psi}^i$ into the following:

$$\tilde{\psi}^i = \left[\sigma_\epsilon^{ii} - \sum_{j \neq i} \sigma_\epsilon^{ij} \right] + \left[(\hat{\sigma}_\epsilon^{ii} - \sigma_\epsilon^{ii}) + \sum_{j \neq i} (\sigma_\epsilon^{ij} - \hat{\sigma}_\epsilon^{ij}) \right]. \quad (49)$$

Recall that we define ψ_i as $\psi^i = \sigma_\epsilon^{ii} - \sum_{j \neq i} \sigma_\epsilon^{ij}$. Hence

$$\begin{aligned} \min_i \tilde{\psi}^i &\geq \min_i \psi^i - \left\| \hat{\Theta}_\epsilon - \Theta_\epsilon^* \right\|_\infty \geq \min_i (\sigma_\epsilon^{ii} - \sum_{j \neq i} \sigma_\epsilon^{ij}) - d\nu_\Theta = \min_i \psi^i - d\nu_\Theta, \\ \max_i \tilde{\psi}^i &\leq \max_i \psi^i + \left\| \hat{\Theta}_\epsilon - \Theta_\epsilon^* \right\|_\infty \leq \max_i (\sigma_\epsilon^{ii} - \sum_{j \neq i} \sigma_\epsilon^{ij}) + d\nu_\Theta = \max_i \psi^i + d\nu_\Theta. \end{aligned}$$

Now given $\nu_\Theta = \eta \epsilon \frac{\log p_2}{n} = O(\sqrt{\log p_2/n})$, with $n \gtrsim d^2 \log p_2$, $d\nu_\Theta = o(1)$, and it follows that

$$\min_i \tilde{\psi}^i - d\nu_\Theta \geq 0.$$

Now by [Lemma 2](#), $X'X/n \sim RE(\varphi^*, \phi^*)$ with high probability. Combine with [Lemma 1](#) and inequality (49), with probability at least $1 - 2 \exp(-c_3 n)$ for some $c_3 > 0$, $\hat{\Gamma}$ satisfies the following RE condition:

$$\hat{\Gamma} = \hat{\Theta}_\epsilon \otimes (X'X/n) \sim RE \left(\varphi^* (\min_i \psi^i - d\nu_\Theta), \phi^* \max_i (\psi^i + d\nu_\Theta) \right), \quad (50)$$

where $\varphi^* = \Lambda_{\min}(\Sigma_X^*)/2$, $\phi^* = \varphi^* \log p_1/n$. \square

To prove [Proposition 2](#), we first prove [Lemma 3](#).

Lemma 3. Let $X \in \mathbb{R}^{n \times p}$ be a zero-mean sub-Gaussian matrix with parameter (Σ_X, σ_X^2) and $E \in \mathbb{R}^{n \times p_2}$ be a zero-mean sub-Gaussian matrix with parameters $(\Sigma_\epsilon, \sigma_\epsilon^2)$. Moreover, X and E are independent. Let $\Theta_\epsilon := \Sigma_\epsilon^{-1}$, then if $n \gtrsim \log(p_1 p_2)$, the following two expressions hold with probability at least $1 - 6c_1 \exp[-(c_2^2 - 1) \log(p_1 p_2)]$ for some $c_1 > 0$, $c_2 > 1$, respectively:

$$\frac{1}{n} \|X'E\|_\infty \leq c_2 [\Lambda_{\max}(\Sigma_X) \Lambda_{\max}(\Sigma_\epsilon)]^{1/2} \sqrt{\frac{\log(p_1 p_2)}{n}}, \quad (51)$$

and

$$\frac{1}{n} \|X'E\Theta_\epsilon\|_\infty \leq c_2 \left[\frac{\Lambda_{\max}(\Sigma_X)}{\Lambda_{\min}(\Sigma_\epsilon)} \right]^{1/2} \sqrt{\frac{\log(p_1 p_2)}{n}}. \quad (52)$$

Proof. The proof of this lemma uses [Lemma 14](#) in [Loh and Wainwright \(2012\)](#), in which they show that if $X \in \mathbb{R}^{n \times p_1}$ is a zero-mean sub-Gaussian matrix with parameters (Σ_X, σ_X^2) and $Y \in \mathbb{R}^{n \times p_2}$ is a zero-mean sub-Gaussian matrix with parameters (Σ_Y, σ_Y^2) , then if $n \gtrsim \log(p_1 p_2)$,

$$\mathbb{P} \left(\left\| \frac{Y'X}{n} - \text{cov}(y_i, x_i) \right\|_\infty \geq t \right) \leq 6p_1 p_2 \exp \left(-cn \min \left\{ \frac{t^2}{(\sigma_X \sigma_Y)^2}, \frac{t}{\sigma_X \sigma_Y} \right\} \right),$$

where X_i and Y_i are the i th row of X and Y , respectively.

Here, we replace Y by E , and since E and X are independent, $\text{cov}(X_i, E_i) = 0$. Let $t = c_2 \sigma_X \sigma_\epsilon \sqrt{\log(p_1 p_2)/n}$, $c_2 > 1$ we get

$$\mathbb{P} \left(\left\| \frac{X'E}{n} \right\|_\infty \geq c_2 \sigma_X \sigma_\epsilon \sqrt{\frac{\log(p_1 p_2)}{n}} \right) \leq 6c_1 (p_1 p_2)^{1-c_2^2} = 6c_1 \exp[-(c_2^2 - 2) \log(p_1 p_2)].$$

Note that the sub-Gaussian parameter satisfies $\sigma_X^2 \leq \max_i (\Sigma_{X,ii}) \leq \Lambda_{\max}(\Sigma_X)$. This directly gives the bound in (51).

To obtain the bound in (52), we note that if E is sub-Gaussian with parameters $(\Sigma_\epsilon, \sigma_\epsilon^2)$, then $E\Theta$ is sub-Gaussian with parameter $(\Theta, \theta_\epsilon^2)$, where

$$\theta_\epsilon^2 \leq \max_i (\Theta_{\epsilon,ii}) \leq \Lambda_{\max}(\Theta_\epsilon) = \frac{1}{\Lambda_{\min}(\Sigma_\epsilon)}.$$

Then we replace Y by $E\Theta$ and yield the bound in (52). \square

As a remark, here we note that the event in (51) and (52) may not be independent. However, the two events hold simultaneously with probability at least $1 - 2c_2 \exp[-c_2 \log(p_1 p_2)]$, with this crude bound for probability hold for sure.

Now we are ready to prove [Proposition 2](#).

Proof of [Proposition 2](#). First we note that

$$X'E\hat{\Theta}_\epsilon = X'E\Theta_\epsilon + X'E(\hat{\Theta}_\epsilon - \Theta_\epsilon^*),$$

which directly gives the following inequality:

$$\|\hat{\gamma} - \hat{\Gamma}\beta^*\|_\infty = \frac{1}{n} \|X'E\hat{\Theta}_\epsilon\|_\infty \leq \frac{1}{n} \|X'E\Theta_\epsilon\|_\infty + \frac{1}{n} \|X'E(\hat{\Theta}_\epsilon - \Theta_\epsilon^*)\|_\infty. \quad (53)$$

Now we would like to bound the two terms separately. The first term can be bounded by (52) in Lemma 3, that is,

$$\frac{1}{n} \|X'E\Theta_\epsilon^*\|_\infty \leq c_2 \left[\frac{\Lambda_{\max}(\Sigma_X)}{\Lambda_{\min}(\Sigma_\epsilon^*)} \right]^{1/2} \sqrt{\frac{\log(p_1 p_2)}{n}}.$$

w.p. at least $1 - 6c_1 \exp[-(c_2^2 - 1) \log(p_1 p_2)]$. For the second term, first we note that

$$\begin{aligned} \frac{1}{n} \|X'E(\widehat{\Theta}_\epsilon - \Theta_\epsilon^*)\|_\infty &= \frac{1}{n} \max_{1 \leq i \leq p_1} \left| e_i' X'E(\widehat{\Theta}_\epsilon - \Theta_\epsilon^*) e_j \right| \\ &\leq \frac{1}{n} \max_i \|e_i' X'E\|_\infty \max_j \|(\widehat{\Theta}_\epsilon - \Theta_\epsilon^*) e_j\|_1, \end{aligned} \quad (54)$$

where we have $e_i \in \mathbb{R}^{p_1}$ and $e_j \in \mathbb{R}^{p_2}$, and the inequality comes from the fact that $|a'b| \leq \|a\|_\infty \|b\|_1$. Note that

$$\max_i \|e_i' X'E\|_\infty = \|X'E\|_\infty,$$

since $\|e_i' X'E\|_\infty$ gives the largest element (in absolute value) of the i th row of $X'E$, and taking the maximum over all i 's gives the largest element of $X'E$ over all entries. And for $\max_j \|(\widehat{\Theta}_\epsilon - \Theta_\epsilon^*) e_j\|_1$, it holds that

$$\max_j \|(\widehat{\Theta}_\epsilon - \Theta_\epsilon^*) e_j\|_1 = \| \widehat{\Theta}_\epsilon - \Theta_\epsilon^* \|_1 = \| \widehat{\Theta}_\epsilon - \Theta_\epsilon^* \|_\infty,$$

where $\|A\|_1 := \max_{|a_i|=1} \|Ax\|_1$ is the ℓ_1 -operator norm, and the last equality follows from the fact that $\|A\|_1 = \|A'\|_\infty$. As a result, (54) can be re-written as:

$$\frac{1}{n} \|X'E(\widehat{\Theta}_\epsilon - \Theta_\epsilon^*)\|_\infty \leq \left(\frac{1}{n} \|X'E\|_\infty \right) \left(\| \widehat{\Theta}_\epsilon - \Theta_\epsilon^* \|_\infty \right). \quad (55)$$

Now, using (51), w.p. at least $1 - 6c_1 \exp[-(c_2^2 - 1) \log(p_1 p_2)]$, we have

$$\frac{1}{n} \|X'E\|_\infty \leq c_2 [\Lambda_{\max}(\Sigma_X) \Lambda_{\max}(\Sigma_\epsilon^*)]^{1/2} \sqrt{\frac{\log(p_1 p_2)}{n}},$$

and since $\| \widehat{\Theta}_\epsilon - \Theta_\epsilon^* \|_\infty \leq \nu_\Theta$, it directly follows that $\| \widehat{\Theta}_\epsilon - \Theta_\epsilon^* \|_\infty \leq d\nu_\Theta$. Therefore, with probability at least $1 - 6c_1 \exp[-(c_2^2 - 1) \log(p_1 p_2)]$,

$$\frac{1}{n} \|X'E(\widehat{\Theta}_\epsilon - \Theta_\epsilon^*)\|_\infty \leq c_2 d\nu_\Theta [\Lambda_{\max}(\Sigma_X) \Lambda_{\max}(\Sigma_\epsilon^*)]^{1/2} \sqrt{\frac{\log(p_1 p_2)}{n}}. \quad (56)$$

Combine the two terms, we obtain the conclusion in Proposition 2. \square

Proof of Proposition 3. First we note the following decomposition:

$$\| \widehat{S} - \Sigma_\epsilon^* \|_\infty \leq \|S - \Sigma_\epsilon\|_\infty + \| \widehat{S} - S \|_\infty := \|W_1\|_\infty + \|W_2\|_\infty,$$

where S is the sample covariance matrix of the true errors E .

For W_1 , by Lemma 8 in Ravikumar et al. (2011), for sample size

$$n \geq 512(1 + 4\sigma_\epsilon^2)^4 \max_i (\Sigma_{\epsilon,ii}^*)^4 \log(4p_2^2),$$

the following bound holds w.p. at least $1 - 1/n_2^{\tau_2 - 2}$ ($\tau_2 > 2$),

$$\|W_1\|_\infty \leq \sqrt{\frac{\log 4 + \tau_2 \log p_2}{c_\epsilon^* n}}, \quad \text{where } c_\epsilon^* = \left[128(1 + 4\sigma_\epsilon^2)^2 \max_i (\Sigma_{\epsilon,ii}^*)^2 \right]^{-1}. \quad (57)$$

For W_2 , rewrite it as:

$$W_2 = \frac{2}{n} E' X(B^* - \widehat{B}) + (B^* - \widehat{B})' \left(\frac{X'X}{n} \right) (B^* - \widehat{B}). \quad (58)$$

The first term in (58) can be bounded as:

$$\left\| \frac{2}{n} E' X(B^* - \widehat{B}) \right\|_\infty \leq 2 \left\| B^* - \widehat{B} \right\|_1 \left\| \frac{1}{n} X'E \right\|_\infty \leq 2 \|\beta^* - \widehat{\beta}\|_1 \cdot \left\| \frac{1}{n} X'E \right\|_\infty. \quad (59)$$

By Lemma 3, with probability at least $1 - 6c_1 \exp[-(c_2^2 - 1) \log(p_1 p_2)]$, the following bound holds:

$$\left\| \frac{2}{n} E' X(B^* - \widehat{B}) \right\|_\infty \leq 2c_2 \nu_\beta [\Lambda_{\max}(\Sigma_X) \Lambda_{\max}(\Sigma_\epsilon^*)]^{1/2} \sqrt{\frac{\log(p_1 p_2)}{n}}, \quad (60)$$

with the sample size requirement being $n \gtrsim \log(p_1 p_2)$.

For the second term in (58), we consider the following bound:

$$\begin{aligned} \left\| (B^* - \widehat{B})' \left(\frac{X'X}{n} \right) (B^* - \widehat{B}) \right\|_\infty &\leq \left\| B^* - \widehat{B} \right\|_1 \left\| \left(\frac{X'X}{n} \right) (B^* - \widehat{B}) \right\|_\infty \\ &\leq \left\| B^* - \widehat{B} \right\|_1^2 \left\| \left(\frac{X'X}{n} \right) \right\|_\infty. \end{aligned} \quad (61)$$

Here, we apply Lemma 8 in Ravikumar et al. (2011) to the design matrix X , for sample size

$$n \geq 512(1 + 4\sigma_x^2)^4 \max_{i,j} (\Sigma_{X,ij})^4 \log(4p_1^2),$$

the following bound holds w.p. at least $1 - 1/p_1^{\tau_1 - 2}$ ($\tau_1 > 2$),

$$\left\| \left(\frac{X'X}{n} \right) - \Sigma_X \right\|_\infty \leq \sqrt{\frac{\log 4 + \tau_1 \log p_1}{c_X^* n}}, \quad \text{where } c_X^* = \left[128(1 + 4\sigma_x^2)^2 \max_{i,j} (\Sigma_{X,ij})^2 \right]^{-1}. \quad (62)$$

This indicates that with this choice of n , the following bound holds with probability at least $1 - 1/p_1^{\tau_1 - 2}$ ($\tau_1 > 2$),

$$\left\| \left(\frac{X'X}{n} \right) \right\|_{\infty} \leq \sqrt{\frac{\log 4 + \tau_1 \log p_1}{c_X^* n}} + \max_i (\Sigma_{X,ii}).$$

Combine with the bound in (61), with probability at least $1 - 1/p_1^{\tau_1 - 2}$ ($\tau_1 > 2$), the following bound holds:

$$\|(B^* - \widehat{B})' \left(\frac{X'X}{n} \right) (B^* - \widehat{B})\|_{\infty} \leq \nu_{\beta}^2 \left(\sqrt{\frac{\log 4 + \tau_1 \log p_1}{c_X^* n}} + \max_i (\Sigma_{X,ii}) \right). \quad (63)$$

Now combine (59), (60) and (63), we reach the conclusion of Proposition 3, with the leading term in the sample size requirement being $n \gtrsim \log(p_1 p_2)$. \square

Proof for Proposition 4. From the structural equations of a multi-layered graph introduced in Section 2.1, and setting $\epsilon^1 := X^1$, we can write

$$\begin{bmatrix} \epsilon^1 \\ \epsilon^2 \end{bmatrix} = \begin{bmatrix} I & 0 \\ -(B^{12})' & I \end{bmatrix} \begin{bmatrix} X^1 \\ X^2 \end{bmatrix}. \quad (64)$$

Define $P = [I, 0; -(B^{12})', I]$. Then, $P\tilde{X}$ is a centered Gaussian random vector with a block diagonal variance-covariance matrix $\text{diag}(\Sigma^1, \Sigma^2)$. Hence, the concentration matrix of \tilde{X} takes the form

$$\Theta_{\tilde{X}} = \Sigma_{\tilde{X}}^{-1} = \begin{bmatrix} I & -B^{12} \\ 0 & I \end{bmatrix} \begin{bmatrix} \Theta^1 & 0 \\ 0 & \Theta^2 \end{bmatrix} \begin{bmatrix} I & 0 \\ -(B^{12})' & 0 \end{bmatrix}.$$

This leads to an upper bound

$$\|\Theta_{\tilde{X}}\| \leq \|\Theta^1\| \|\Theta^2\| \|P\|^2.$$

The result then follows by using the matrix norm inequality $\|A\| \leq \sqrt{\|A\|_1 \|A\|_{\infty}}$ (Golub and Van Loan, 2012), where $\|A\|_1$ and $\|A\|_{\infty}$ denote the maximum absolute row and column sums of A , and the fact that $\Lambda_{\min}(\Sigma_{\tilde{X}}) = \|\Theta_{\tilde{X}}\|^{-1}$. \square

Appendix C. Numerical comparisons between different parametrizations.

In this subsection, we provide some numerical evidence to substantiate the point we made in Section 5, that the two parametrizations are not always equivalent. This is a point also mentioned in the original work on AMP graphs by Andersson et al. (2001), the framework adopted in this paper. The other parametrization which we referred to as the (Ω_{XY}, Ω_Y) -parametrization corresponds to the LWF framework (see Andersson et al., 2001, p.34-35). In the presence of sparsity penalization, a specific sparsity pattern for the (B, Θ_{ϵ}) -parametrization may not be recoverable through the (Ω_{XY}, Ω_Y) -parametrization and vice versa.

Consider the following two simulation settings, in which the data are generated from the AMP framework (B, Θ_{ϵ}) -parametrization and the LWF framework (Ω_{XY}, Ω_Y) -parametrization respectively.

- AMP framework. The data are generated according to the model $Y = XB^* + E$, similar to Model A described in Section 4; that is, each entry in B^* is nonzero with probability $5/p_1$, and off-diagonal entries for Θ_{ϵ}^* are nonzero with probability $5/p_2$. Nonzero entries of B^* and Θ_{ϵ}^* are generated from $\text{Unif}[-1, -0.5) \cup (0.5, 1]$, and diagonals of Θ_{ϵ}^* are set identical, such that the condition number of Θ_{ϵ}^* is p_2 . Table 11 shows the performance of estimated B using different methods that are designed for different parametrizations: the node-conditional method (mixed MRF) and the proposed method in this study (PML).

Table 11: Performance for \widehat{B} using different methods for different parametrizations

(p_1, p_2, n)	Method	SEN	SPC	MCC
(30, 60, 100)	mixed MRF (th)	0.86	0.71	0.45
	PML	0.96	0.99	0.93
(60, 30, 100)	mixed MRF (th)	0.96	0.76	0.70
	PML	0.99	0.99	0.93
(200, 200, 150)	mixed MRF (th)	0.80	0.99	0.70
	PML	0.99	0.99	0.88

- LWF framework. The data are generated based on the multivariate Gaussian specification:

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim \mathcal{N} \left(0, \begin{pmatrix} \Omega_X & \Omega_{XY} \\ \Omega_{YX} & \Omega_Y \end{pmatrix} \right),$$

Specifically, Ω_X is banded with 1 on the diagonal and 0.2 on the upper and lower first diagonal, Ω_Y is also banded with 1 on the diagonal and 0.3 on the upper and lower first diagonal. Each entry in Ω_{XY} is nonzero with probability $5/p_1$, and the nonzero entries are generated from $\text{Unif}[-1, -0.8) \cup (0.8, 1]$. Further, we bump up the diagonal of the joint precision matrix $\begin{bmatrix} \Omega_X & \Omega_{XY} \\ \Omega_{YX} & \Omega_Y \end{bmatrix}$ such that it is positive definite. Table 12 depicts the selection property of the estimated Ω_{XY} using different methods that are designed for different parametrizations.

Table 12: Performance for $\widehat{\Omega}_{XY}$ using different methods for different parametrizations

(p_1, p_2, n)	Method	SEN	SPC	MCC
(30, 60, 100)	mixed MRF	0.84	0.88	0.63
	PML-th	0.99	0.52	0.39
(60, 30, 100)	mixed MRF	0.847	0.95	0.70
	PML-th	1	0.80	0.52
(200, 200, 150)	mixed MRF	0.89	0.93	0.70
	PML-th	1	0.79	0.30

Note that to retrieve stable and meaningful results, for the AMP framework, the estimates using mixed MRF are thresholded at a proper level, and for the LWF framework, the estimates using PML are also thresholded.

It can be seen that the method compatible with the data generation mechanism exhibits superior performance, vis-a-vis its competitor that was designed for another parameterization. Further, the mixed MRF method suffers in terms of both sensitivity and specificity under the AMP parameterization, while the PML method suffers in terms of specificity only under the LWF parameterization.

Appendix D. An example for multi-layered network estimation.

As mentioned at the beginning of Section 2, the proposed methodology is designed for obtaining MLEs for multi-layer Gaussian networks, but the problem breaks down into a sequence of 2-layered estimation problems. Here we give an detailed example to illustrate how our proposed methodology proceeds for a 3-layered network.

Suppose there are p_1, p_2 and p_3 nodes in Layers 1, 2 and 3, respectively. This three-layered network is modeled as follows:

$$\begin{aligned} & - \mathbf{X} \sim \mathcal{N}(0, \Sigma_X), \mathbf{X} \in \mathbb{R}^{p_1}, \\ & - \text{For } j = 1, \dots, p_2: Y_j = \mathbf{X}' B_j^{x^{2y}} + \epsilon_j^y, B_j^{x^{2y}} \in \mathbb{R}^{p_1}, (\epsilon_1^y \dots \epsilon_{p_2}^y)' \sim \mathcal{N}(0, \Sigma_{\epsilon^y}), \\ & - \text{For } l = 1, 2, \dots, p_3: Z_l = \mathbf{X}' B_l^{x^{yz}} + \mathbf{Y}' B_l^{y^z} + \epsilon_l^z, B_l^{x^{yz}} \in \mathbb{R}^{p_1} \text{ and } B_l^{y^z} \in \mathbb{R}^{p_2}, \\ & \quad (\epsilon_1^z \dots \epsilon_{p_3}^z)' \sim \mathcal{N}(0, \Sigma_{\epsilon^z}). \end{aligned}$$

The parameters of interest are: $\Theta_{XY}, \Theta_{\epsilon^y}, \Theta_{\epsilon^z} := \Sigma_{\epsilon^y}^{-1}, \Theta_{\epsilon^z} := \Sigma_{\epsilon^z}^{-1}$, which denote the within-layer conditional dependencies, and

$$B_{XY} = [B_{1^{xy}}^{xy} \dots B_{p_2^{xy}}^{xy}], B_{XZ} = [B_{1^{xz}}^{xz} \dots B_{p_3^{xz}}^{xz}] \text{ and } B_{YZ} = [B_{1^{yz}}^{yz} \dots B_{p_3^{yz}}^{yz}],$$

which encode the across-layer dependencies.

Now given data $X \in \mathbb{R}^{n \times p_1}$, $Y \in \mathbb{R}^{n \times p_2}$ and $Z \in \mathbb{R}^{n \times p_3}$, all centered, the full log-likelihood can be written as:

$$\ell(Z, Y, X) = \ell(Z|Y, X; \Theta_{\epsilon^z}, B_{YZ}, B_{XZ}) + \ell(Y|X; \Theta_{\epsilon^y}, B_{XY}) + \ell(X; \Theta_X). \quad (65)$$

The separability of the log-likelihood enables us to ignore the inner structure of the combined layer $\tilde{X} := (X, Y)$ when trying to estimate the dependencies between Layer 1 and Layer 3, Layer 2 and Layer 3, as well as the conditional dependencies within Layer 3. As a consequence, the optimization problem minimizing the negative log-likelihood can be decomposed into three separate problems, i.e., solving for $\{\Theta_{\epsilon^z}, B_{XZ}, B_{YZ}\}$, $\{\Theta_{\epsilon^y}, B_{XY}\}$ and $\{\Theta_X\}$, respectively.

The estimation procedure described in Section 2.2 can thus be carried out in a recursive way in a sense of what follows. To obtain estimates for $\{B_{XZ}, B_{YZ}, \Theta_{\epsilon^z}\}$, based on the formulation in (2), we solve the following optimization problem:

$$\min_{\substack{\Theta_{\epsilon^z, Z} \in \mathbb{S}_{p_3}^{p_1 \times p_3} \\ B_{XZ}, B_{YZ}}} \left\{ \begin{aligned} & -\log \det \Theta_{\epsilon^z, Z} + \frac{1}{n} \sum_{j=1}^{p_3} \sigma_{Z_j}^2 (Z_j - X B_j^{xz} - Y B_j^{yz})^\top (Z_j - X B_j^{xz} - Y B_j^{yz}) \\ & + \lambda_n (\|B_{XZ}\|_1 + \|B_{YZ}\|_1) + \rho_n \|\Theta_{\epsilon^z, Z}\|_{1, \text{off}} \end{aligned} \right\},$$

which can be solved by treating the combined design matrix $\tilde{X} = (X, Y)$ as a single super layer and Z as the response layer, then apply each step described in Section 2.2. To obtain estimates for B_{XY} and Θ_{ϵ^y} , we can ignore the 3rd layer for now and apply the exact procedure all over again, by treating Y as the response layer and X as the design layer. The estimate for the precision matrix of the bottom layer Θ_X can be obtained by graphical lasso (Friedman et al., 2008) or the nodewise regression (Meinshausen and Bühlmann, 2006).

References

- Steen A Andersson, David Madigan, and Michael D Perlman. Alternative Markov properties for chain graphs. *Scandinavian Journal of Statistics*, 28(1):33–85, 2001.
- Sumantra Basu and George Michailidis. Regularized estimation in sparse high-dimensional time series models. *The Annals of Statistics*, 43(4):1535–1567, 2015.
- Joseph K Bradley, Aapo Kyrola, Danny Bickson, and Carlos Guestrin. Parallel coordinate descent for ℓ_1 -regularized loss minimization. *arXiv preprint arXiv:1105.5379*, 2011.
- Peter Bühlmann and Sara Van De Geer. *Statistics for High-dimensional Data: Methods, Theory and Applications*. Springer Science & Business Media, 2011.
- T Tony Cai, Hongzhe Li, Weidong Liu, and Jichun Xie. Covariate-adjusted precision matrix estimation with an application in genetical genomics. *Biometrika*, 100(1):139–156, 2012.
- Tony Cai, Weidong Liu, and Xi Luo. A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607, 2011.
- Emmanuel Candès and Terence Tao. The Dantzig selector: statistical estimation when p is much larger than n . *The Annals of Statistics*, 35(6):2313–2351, 2007.
- Mathias Drton and Michael D Perlman. A SINFul approach to Gaussian graphical model selection. *Journal of Statistical Planning and Inference*, 138(4):1179–1200, 2008.
- Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.
- Morten Frydenberg. The chain graph Markov property. *Scandinavian Journal of Statistics*, 17(4):333–353, 1990.
- Gene H Golub and Charles F Van Loan. *Matrix Computations*. JHU Press, 2012.
- Adel Javanmard and Andrea Montanari. Confidence intervals and hypothesis testing for high-dimensional regression. *The Journal of Machine Learning Research*, 15(1):2869–2909, 2014.
- Steffen L Lauritzen. *Graphical Models*. Oxford University Press, 1996.
- Steffen L Lauritzen and Nanny Wermuth. Graphical models for associations between variables, some of which are qualitative and some quantitative. *The Annals of Statistics*, 17(1):31–57, 1989.
- Wonyul Lee and Yufeng Liu. Simultaneous multiple response regression and inverse covariance matrix estimation via penalized Gaussian maximum likelihood. *Journal of Multivariate Analysis*, 111:241–255, 2012.
- Erich Leo Lehmann and George Casella. *Theory of Point Estimation*. Springer Science & Business Media, 1998.
- Po-Ling Loh and Martin J Wainwright. High-dimensional regression with noisy and missing data: provable guarantees with nonconvexity. *The Annals of Statistics*, 40(3):1637–1664, 2012.
- Calvin McCarter and Seyoung Kim. On sparse Gaussian chain graph models. In *Advances in Neural Information Processing Systems*, pages 3212–3220, 2014.
- Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the Lasso. *The Annals of Statistics*, 34(3):1436–1462, 2006.
- Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.
- Pradeep Ravikumar, Martin J Wainwright, Garvesh Raskutti, and Bi Yu. High-dimensional covariance estimation by minimizing ℓ_1 -penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.
- Peter Richtárik and Martin Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, pages 1–52, 2012.
- Adam J Rothman, Elizaveta Levina, and Ji Zhu. Sparse multivariate regression with covariance estimation. *Journal of Computational and Graphical Statistics*, 19(4):947–962, 2010.
- Karen Sachs, Omar Perez, Dana Pe’er, Douglas A Lauffenburger, and Garry P Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.
- Chad Scherer, Mahantesh Halappanavar, Ambuj Tewari, and David Haglin. Scaling up coordinate descent algorithms for large ℓ_1 regularization problems. *arXiv preprint arXiv:1206.6409*, 2012.
- Michael E Sobel. Causal inference in the social sciences. *Journal of the American Statistical Association*, 95(450):647–651, 2000.
- Kyung-Ah Sohn and Seyoung Kim. Joint estimation of structured sparsity and output structure in multiple-output regression via inverse-covariance regularization. In *International Conference on Artificial Intelligence and Statistics*, pages 1081–1089, 2012.
- Paul Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109(3):475–494, 2001.
- Sara Van de Geer, Peter Bühlmann, Yaacov Ritov, and Ruben Dezeure. On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics*, 42(3):1166–1202, 2014.
- Chao Wang, Venu Satuluri, and Srinivasan Parthasarathy. Local probabilistic models for link prediction. In *Seventh IEEE International Conference on Data Mining*, pages 322–331. IEEE, 2007.

- Emho Yang, Yulia Baker, Pradeep Ravikumar, Genevera Allen, and Zhaodong Liu. Mixed graphical models via exponential families. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pages 1042–1050, 2014.
- Jianxin Yin and Hongzhe Li. A sparse conditional Gaussian graphical model for analysis of genetical genomics data. *The Annals of Applied Statistics*, 5(4):2630, 2011.
- Xiao-Tong Yuan and Tong Zhang. Partial Gaussian graphical model estimation. *IEEE Transactions on Information Theory*, 60(3):1673–1687, 2014.
- Cun-Hui Zhang and Stephanie S Zhang. Confidence intervals for low dimensional parameters in high dimensional linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(1):217–242, 2014.
- Tuo Zhao, Xingqun Li, Han Liu, Kathryn Roeder, John Lafferty, and Larry Wasserman. *huge: High-dimensional undirected graph estimation*, 2015. URL <http://CRAN.R-project.org/package=huge>. R package version 1.2.7.

Local Network Community Detection with Continuous Optimization of Conductance and Weighted Kernel K-Means

Twan van Laarhoven

Elena Marchiori

*Institute for Computing and Information Sciences
Radboud University Nijmegen
Postbus 9010
6500 GL Nijmegen, The Netherlands*

TVANLAARHOVEN@CS.RU.NL

ELENAM@CS.RU.NL

Editor: Jure Leskovec

Abstract

Local network community detection is the task of finding a single community of nodes concentrated around few given seed nodes in a localized way. Conductance is a popular objective function used in many algorithms for local community detection. This paper studies a continuous relaxation of conductance. We show that continuous optimization of this objective still leads to discrete communities. We investigate the relation of conductance with weighted kernel k-means for a single community, which leads to the introduction of a new objective function, σ -conductance. Conductance is obtained by setting σ to 0. Two algorithms, EMC and PGDC, are proposed to locally optimize σ -conductance and automatically tune the parameter σ . They are based on expectation maximization and projected gradient descent, respectively. We prove locality and give performance guarantees for EMC and PGDC for a class of dense and well separated communities centered around the seeds. Experiments are conducted on networks with ground-truth communities, comparing to state-of-the-art graph diffusion algorithms for conductance optimization. On large graphs, results indicate that EMC and PGDC stay localized and produce communities most similar to the ground, while graph diffusion algorithms generate large communities of lower quality.¹

Keywords: community detection, conductance, k-means

1. Introduction

Imagine that you are trying to find a community of nodes in a network around a given set of nodes. A simple way to approach this problem is to consider this set as seed nodes, and then keep adding nodes in a local neighborhood of the seeds as long as this makes the community better in some sense. In contrast to global clustering, where the overall community structure of a network has to be found, local community detection aims to find only one community around the given seeds by relying on local computations involving only nodes relatively close to the seed. Local community detection by seed expansion is

especially beneficial in large networks, and is commonly used in real-life large scale network analysis (Gargi et al., 2011; Leskovec et al., 2010; Wu et al., 2012).

Several algorithms for local community detection operate by seed expansion. These methods have different expansion strategies, but what they have in common is their use of conductance as the objective to be optimized. Intuitively, conductance measures how strongly a set of nodes is connected to the rest of the graph; sets of nodes that are isolated from the graph have low conductance and make good communities.

The problem of finding a set of minimum conductance in a graph is computationally intractable (Chawla et al., 2005; Šima and Schaeffer, 2006). As a consequence, many heuristic and approximation algorithms for local community detection have been introduced (see references in the related work section). In particular, effective algorithms for this task are based on the local graph diffusion method. A graph diffusion vector \mathbf{f} is an infinite series $\mathbf{f} = \sum_{i=0}^{\infty} \alpha_i \mathbf{P}^i \mathbf{s}$, with diffusion coefficients $\sum_{i=0}^{\infty} \alpha_i = 1$, seed nodes \mathbf{s} , and random walk transition matrix \mathbf{P} . Types of graph diffusion, such as personalized Page Rank (Andersen and Lang, 2006) and Heat Kernel (Chung, 2007), are determined by the choice of the diffusion coefficients. In the diffusion method an approximation of \mathbf{f} is computed. After dividing each vector component by the degree of the corresponding node, the nodes are sorted in descending order by their values in this vector. Next, the conductance of each prefix of the sorted list is computed and either the set of smallest conductance is selected, e.g. in (Andersen and Lang, 2006) or a local optima of conductance along the prefix length dimension (Yang and Leskovec, 2012) is considered.

These algorithms optimize conductance along a single dimension, representing the order in which nodes are added by the algorithm. However this ordering is mainly related to the seed, and not directly to the objective that is being optimized. Algorithms for the direct optimization of conductance mainly operate in the discrete search space of communities, and locally optimize conductance by adding and/or removing one node. This amounts to fixing a specific neighborhood structure over communities where the neighbors of a community are only those communities which differ by the membership of a single node. This is just one possible choice of community neighbor. A natural way to avoid the problem of choosing a specific neighborhood structure is to use continuous rather than discrete optimization. To do this, we need a continuous relaxation of conductance, extending the notion of communities to allow for fractional membership. This paper investigates such a continuous relaxation, which leads to the following findings.

1.0.1 ON LOCAL OPTIMA

Although local optima of a continuous relaxation of conductance might at first glance have nodes with fractional memberships, somewhat surprisingly all strict local optima are discrete. This means that continuous optimization can directly be used to find communities without fractional memberships.

1.0.2 RELATION WITH WEIGHTED KERNEL K-MEANS

We unravel the relation between conductance and weighted kernel k-means objectives using the framework by Dhillion et al. (2007). Since the aim is to find only one community, we consider a slight variation with one mean, that is, with $k = 1$. This relation leads

¹ Source code of the algorithms used in the paper is available at <http://cs.ru.nl/~tvanlaarhoven/conductance2016>.

to the introduction of a new objective function for local community detection, called σ -conductance, which is the sum of conductance and a regularization term whose influence is controlled by a parameter σ . Interestingly, the choice of σ has a direct effect on the number of local optima of the function, where larger values of σ lead to more local optima. In particular, we prove that for $\sigma > 2$ all discrete communities are local optima. As a consequence, due to the seed expansion approach, local optimization of σ -conductance favors smaller communities for larger values of σ .

1.0.3 ALGORITHMS

Local optimization of σ -conductance can be easily performed using the projected gradient descent method. We develop an algorithm based on this method, called PGDC. Motivated by the relation between conductance and k-means clustering, we introduce an Expectation-Maximization (EM) algorithm for σ -conductance optimization, called EMC. We show that for $\sigma = 0$, this algorithm is almost identical to projected gradient descent with an infinite step size in each iteration. We then propose a heuristic procedure for choosing σ automatically in these algorithms.

1.0.4 RETRIEVING COMMUNITIES

We give a theoretic characterization of a class of communities, called dense and isolated communities, for which PGDC and EMC perform optimally. For this class of communities the algorithms exactly recover a community from the seeds. We investigate the relation between this class of communities and the notion of (α, β) -cluster proposed by (Mishra et al., 2008) for social networks analysis. And we show that, while all maximal cliques in a graph are (α, β) -clusters, they are not necessarily dense and isolated communities. We give a simple condition on the degree of the nodes of a community which guarantees that a dense and isolated community satisfying such condition is also an (α, β) -cluster.

1.0.5 EXPERIMENTAL PERFORMANCE

We use publicly available artificial and real-life network data with labeled ground-truth communities to assess the performance of PGDC and EMC. Results of the two methods are very similar, with PGDC performing slightly better, while EMC is slightly faster. These results are compared with those obtained by three state-of-the-art algorithms for conductance optimization based on the local graph diffusion: the popular Personalized Page Rank (PPR) diffusion algorithm by Andersen and Lang (2006), a more recent variant by Yang and Leskovec (2012) (here called YL), and the Heat Kernel (HK) diffusion algorithm by Kloster and Gleich (2014). On large networks PGDC and EMC stay localized and produce communities which are more faithful to the ground truth than those generated by the considered graph diffusion algorithms. PPR and HK produce much larger communities with a low conductance, while the YL strategy outputs very small communities with a higher conductance.

1.1 Related Work

The enormous growth of network data from diverse disciplines such as social and information science and biology has boosted research on network community detection (see for instance the overviews by Schaefer (2007) and Fortunato (2010)). Here we confine ourselves to literature we consider to be relevant to the present work, namely local community detection by seed expansion, and review related work on conductance as objective function and its local optimization. We also briefly review research on other objectives functions, and on properties of communities and of seeds.

1.1.1 CONDUCTANCE AND ITS LOCAL OPTIMIZATION

Conductance has been largely used for network community detection. For instance Leskovec et al. (2008) introduced the notion of network community profile plot to measure the quality of a ‘best’ community as a function of community size in a network. They used conductance to measure the quality of a community and analyze a large number of communities of different size scales in real-world social and information networks.

Direct conductance optimization was shown to favor communities which are quasi-cliques (Kang and Faloutsos, 2011) or communities of large size which include irrelevant subgraphs (Andersen and Lang, 2006; Whang et al., 2013).

Popular algorithms for local community detection employ the local graph diffusion method to find a community with small conductance.

Starting from the seminal work by Spielman and Teng (2004) various algorithms for local community detection by seed expansion based on this approach have been proposed (Andersen et al., 2006; Avron and Horesh, 2015; Chung, 2007; Kloster and Gleich, 2014; Zhu et al., 2013a). The theoretical analysis in these works is largely based on a mixing result which shows that a cut with small conductance can be found by simulating a random walk starting from a single node for sufficiently many steps (Lovász and Simonovits, 1990). This result is used to prove that if the seed is near to a set with small conductance then the result of the procedure is a community with a related conductance, which is returned in time proportional to the volume of the community (up to a logarithmic factor).

Mahoney et al. (2012) performed local community detection by modifying the spectral program used in standard global spectral clustering. Specifically the authors incorporated a bias towards a target region of seed nodes in the form of a constraint to force the solution to be well connected with or to lie near the seeds. The degree of connectedness was specified by setting a so-called correlation parameter. The authors showed that the optimal solution of the resulting constrained optimization problem is a generalization of Personalized PageRank (Andersen and Lang, 2006).

1.1.2 OTHER OBJECTIVES

Conductance is not the only objective function used in local community detection algorithms. Various other objective functions have been considered in the literature. For instance, Chen et al. (2009) proposed to use the ratio of the average internal and external degree of nodes in a community as objective function. Clauset (2005) proposed a local variant of modularity. Wu et al. (2015) modified the classical density objective, equal to the sum of edges in the community divided by its size, by replacing the denominator with the

sum of weights of the community nodes, where the weight of a node quantifies its proximity to the seeds and is computed using a graph diffusion method.

A comparative experimental analysis of objective functions with respect to their experimental and theoretical properties was performed e.g. in (Yang and Leskovec, 2012) and (Wu et al., 2015), respectively.

1.1.3 PROPERTIES OF COMMUNITIES

Instead of focusing on objective functions and methods for local community detection, other researchers investigated properties of communities. Mishra et al. (2008) focused on interesting classes of communities and algorithms for their exact retrieval. They defined the so called (α, β) -communities and developed algorithms capable of retrieving this type of communities starting from a seed connected to a large fraction of the members of the community. Zhu et al. (2013b) considered the class of well-connected communities, which have a better internal connectivity than conductance. Internal connectivity of a community is defined as the inverse of the mixing time for a random walk on the subgraph induced by the community. They showed that for well-connected communities, it is possible to provide an improved performance guarantee, in terms of conductance of the output, for local community detection algorithms based on the diffusion method. Gleich and Seshadhri (2012) investigated the utility of neighbors of the seed; in particular they showed empirically that such neighbors form a 'good' local community around the seed. Yang and Leskovec (2012) investigated properties of ground truth communities in social, information and technological networks.

Lancichinetti et al. (2011) addressed the problem of finding a significant local community from an initial group of nodes. They proposed a method which locally optimizes the statistical significance of a community, defined with respect to a global null model, by iteratively adding external significant nodes and removing internal nodes that are not statistically relevant. The resulting community is not guaranteed to contain the nodes of the initial community.

1.1.4 PROPERTIES OF SEEDS

Properties of seeds in relation to the performance of algorithms were investigated by e.g. Kloumann and Kleinberg (2014). They considered different types of algorithms, in particular a greedy seed expansion algorithm which at each step adds the node that yields the most negative change in conductance (Mislove et al., 2010). Whang et al. (2013) investigated various methods for choosing the seeds for a PageRank based algorithm for community detection. Chen et al. (2013) introduced the notion of local degree central node, whose degree is greater than or equal to the degree of its neighbor nodes. A new local community detection method is introduced based on the local degree central node. In this method, the local community is not discovered from the given starting node, but from the local degree central node that is associated with the given starting node.

1.2 Notation

We start by introducing the notation used in the rest of this paper. We denote by V the set of nodes in a network or graph G . A community, also called a cluster, $C \subseteq V$ will be a

subset of nodes, and its complement $\bar{C} = V \setminus C$ consists of all nodes not in C . Note that we consider any subset of nodes to be a community, and the goal of community detection is to find a *good* community.

Let A be the adjacency matrix of G , where a_{ij} denotes the weight of an edge between nodes i and j . In unweighted graphs a_{ij} is either 0 or 1, and in undirected graphs $a_{ij} = a_{ji}$. In this paper we work only with unweighted undirected graphs. We can generalize this notation to sets of nodes, and write $a_{xy} = \sum_{i \in x, j \in y} a_{ij}$. With this notation in hand we can write conductance as

$$\phi(C) = \frac{a_{C\bar{C}}}{a_{CV}} = 1 - \frac{a_{CC}}{a_{CV}}.$$

A common alternative definition is

$$\phi_{\text{alt}}(C) = \frac{a_{C\bar{C}}}{\min(a_{CV}, a_{\bar{C}V})},$$

which considers the community to be the smallest of C and \bar{C} . For instance Kloster and Gleich (2014) and Andersen and Lang (2006) use this alternative definition, while Yang and Leskovec (2012) use ϕ .

Note that ϕ has a trivial optimum when all nodes belong to the community, while ϕ_{alt} will usually have a global optimum with roughly half of the nodes belonging to the community. Neither of these optima are desirable for finding a single small community.

With a set X we associate an indicator vector $[X]$ of length $|V|$, such that $[X]_i = 1$ if $i \in X$ and $[X]_i = 0$ otherwise. We will usually call this vector \mathbf{x} .

2. Continuous Relaxation of Conductance

If we want to talk about directly optimizing conductance, then we need to define what (local) optima are. The notion of local optima depends on the topology of the input space, that is to say, on what communities we consider to be neighbors of other communities. We could, for instance, define the neighbors of a community to be all communities that can be created by adding or removing a single node. But this is an arbitrary choice, and we could equally well define the neighbors to be all communities reached by adding or removing up to two nodes. An alternative is to move to the continuous world, where we can use our knowledge of calculus to give us a notion of local optima.

To turn community finding into a continuous problem, instead of a set C we need to see the community as a vector \mathbf{c} of real numbers between 0 and 1, where c_i denotes the degree to which node i is a member of the community. Given a discrete community C , we have $\mathbf{c} = [C]$, but the inverse is not always possible, so the vectorial setting is more general.

The edge weight between sets of nodes can be easily generalized to the edge weight of membership vectors,

$$a_{\mathbf{x}\mathbf{y}} = \mathbf{x}^T \mathbf{A} \mathbf{y} = \sum_{i \in V} \sum_{j \in V} x_i a_{ij} y_j.$$

Now we can reinterpret the previous definition of conductance as a function of real vectors, which we could expand as

$$\phi(\mathbf{c}) = 1 - \frac{\sum_{i,j \in V} c_i a_{ij} c_j}{\sum_{i,j \in V} c_i a_{ij}}.$$

With this definition we can apply the vast literature on constrained optimization of differentiable functions. In particular, we can look for local optima of the conductance, subject to the constraint that $0 \leq c_i \leq 1$. These local optima will satisfy the Karush-Kuhn-Tucker conditions, which in this case amounts to, for all $i \in V$,

$$\begin{aligned} 0 &\leq c_i \leq 1 \\ \nabla \phi(\mathbf{c})_i &\geq 0 && \text{if } c_i = 0 \\ \nabla \phi(\mathbf{c})_i &= 0 && \text{if } 0 < c_i < 1, \\ \nabla \phi(\mathbf{c})_i &\leq 0 && \text{if } c_i = 1. \end{aligned}$$

To use the above optimization problem for finding communities from seeds, we add one additional constraint. Given a set S of seeds we require that $c_i \geq s_i$; in other words, that the seed nodes are members of the community. This is the only way in which the seeds are used, and the only way in which we can use the seeds without making extra assumptions.

2.1 A Look at the Local Optima

By allowing community memberships that are real numbers, uncountably many more communities are possible. One might expect that it is overwhelmingly likely that optima of the continuous relaxation of conductance are communities with fractional memberships. But this turns out not to be the case. In fact, the strict local optima will all represent discrete communities.

To see why this is the case, consider the objective in terms of the membership coefficient c_i for some node i . This takes the form of a quadratic rational function,

$$\phi(c_i) = \frac{\alpha_1 + \alpha_2 c_i + \alpha_3 c_i^2}{\alpha_4 + \alpha_5 c_i}.$$

The coefficients in the denominator are positive, which means that the denominator is also positive for $c_i > 0$. At an interior local minimum we must have $\phi'(c_i) = 0$, which implies that $\phi''(c_i) = 2\alpha_3/(\alpha_4 + \alpha_5 c_i)^3$. But $\alpha_3 \leq 0$, since it comes from the $c_i \alpha_i \alpha_i c_i$ term in the numerator of the conductance, so $\phi''(c_i) \leq 0$, and hence there are only local maxima or saddle points, not strict local minima.

It is still possible for there to be plateaus in the objective functions, where $\phi(\mathbf{c})$ is optimal regardless of the value of c_i for a certain node i .

2.2 The Relation to Weighted Kernel K-Means Clustering

Another view on conductance is by the connection to weighted kernel k -means clustering. The connection between weighted kernel k -means and objectives for graph partitioning has been thoroughly investigated in Dhillon et al. (2007). Here we extend that connection to the single cluster case.

Start with weighted k -means clustering, which, given a dataset $\{x_i\}_{i=1}^N$ and weights $\{w_i\}$ minimizes the following objective

$$\sum_{i=1}^N \sum_{j=1}^k w_i c_{ij} \|x_i - \mu_j\|_2^2$$

with respect to μ_j and c_{ij} , where c_{ij} indicates if point i belongs to cluster j , subject to the constraint that exactly one c_{ij} is 1 for every i .

Since our goal is to find a single cluster, a first guess would be to take $k = 2$, and to try to separate a foreground cluster from the background. But when using 2-means, there is no distinction between foreground and background, and so solutions will naturally have two clusters of roughly equal size. Instead, we can consider a one-cluster variant that distinguishes between points in a cluster and background points, which we call 1-mean clustering. This can be formulated as the minimization of

$$\sum_i w_i (c_i \|x_i - \mu\|_2^2 + (1 - c_i) \lambda_i)$$

with respect to a single μ and cluster membership indicators c_i (between 0 and 1). Here λ_i is a cost for node i being a member of the background.

We allow different λ_i for different nodes, as there is no reason to demand a single value. The condition for a node i to be part of the community is $\|x_i - \mu\|_2^2 < \lambda_i$. So different values for λ_i might be useful for two reasons. The first would be to allow incorporating prior knowledge, the second reason would be if the scale (of the clusters) is different, that is, nodes (in different clusters) have different distances from the mean. By adding a diagonal matrix to the kernel, the squared distance from all points to all other points is increased by that same amount. It makes sense to compensate for this in the condition for community membership. And since the diagonal terms we add to the kernel vary per node, the amount that these nodes move away from other points also varies, which is why we use different λ_i per node.

The minimizer for μ is the centroid of the points inside the cluster,

$$\mu = \frac{\sum_i w_i c_i x_i}{\sum_i w_i c_i},$$

while the minimizer for c_i is 1 if and only if $\|x_i - \mu\|_2^2 < \lambda_i$, and 0 otherwise.

The k -means and 1-mean objectives can be kernelized by writing distances in terms of inner products, and using a kernel $K(i, j) = \langle x_i, x_j \rangle$. The cluster mean is then a linear combination of points, $\mu = \sum_i \mu_i x_i$, giving

$$\|x_i - \mu\|_2^2 = K(i, i) - 2 \sum_j \mu_j K(i, j) + \sum_{j,k} \mu_j \mu_k K(j, k).$$

By filling in the optimal μ given above, the 1-mean objective then becomes

$$\begin{aligned} \phi_{W,K,\lambda}(\mathbf{c}) &= \sum_i w_i c_i (K(i, i) - \lambda_i) + \sum_i w_i \lambda_i \\ &\quad - \frac{\sum_{i,j} w_i c_i w_j c_j K(i, j)}{\sum_i w_i c_i}. \end{aligned}$$

The second term is constant, so we can drop it for the purposes of optimization.

We pick $\lambda_i = K(i, i)$. With this choice, the condition for a node i to be a member of the community is $\|x_i - \mu\|_2^2 < \|x_i - 0\|_2^2$. This can be seen as a 2-means cluster assignment

where the background cluster has the origin as the fixed mean. With this choice the first term also drops out.

By converting the graph into a kernel with

$$K = W^{-1}AW^{-1},$$

where W is a diagonal matrix with the weights w_i on the diagonal, we can obtain objectives like conductance and association ratio. However this K is not a legal kernel, because a kernel has to be positive definite. Without a positive definite kernel the distances $\|x_i - \mu\|$ from the original optimization problem can become negative. To make the kernel positive definite, we follow the same route as Dhillon et al., and add a diagonal matrix, obtaining

$$K = \sigma W^{-1} + W^{-1}AW^{-1}.$$

Since we are interested in conductance, we take as weights $w_i = a_{iV}$, the degree of node i , and we take $\lambda_i = K(i, i)$. This results (up to an additive constant) in the following objective which we call σ -conductance,

$$\phi_\sigma(\mathbf{c}) = 1 - \frac{\sum_{i,j} c_i c_j a_{ij}}{\sum_i c_i a_{iV}} - \sigma \frac{\sum_i c_i^2 a_{iV}}{\sum_i c_i a_{iV}}.$$

Observe that if \mathbf{c} is a discrete community, then $c_i^2 = c_i$, and the last term is constant. In that case optimization of this objective is exactly equivalent to optimizing conductance.

For the purposes of continuous optimization however, increasing the σ parameter has the effect of increasing the objective value of non-discrete communities. So different communities become more separated, and in the extreme case, every discrete community becomes a local optimum.

Theorem 1 *When $\sigma > 2$, all discrete communities \mathbf{c} are local minima of $\phi_\sigma(\mathbf{c})$ constrained to $0 \leq c_i \leq 1$.*

Proof The gradient of ϕ_σ is

$$\nabla \phi_\sigma(\mathbf{c})_i = a_{iV} \frac{a_{iC}}{a_{iV}^2} - 2 \frac{a_{iC}}{a_{iV}} + \sigma \left(a_{iV} \frac{c_i^2 a_{iV}}{a_{iV}^2} - 2c_i \frac{a_{iV}}{a_{iV}} \right).$$

When \mathbf{c} is discrete, then $\sum_j c_j^2 a_{jV} = a_{iV}$, so the gradient simplifies to

$$\nabla \phi_\sigma(\mathbf{c})_i = \frac{a_{iV}}{a_{iV}} \left(\frac{a_{iC}}{a_{iV}} + (1 - 2c_i) \sigma - 2 \frac{a_{iC}}{a_{iV}} \right).$$

Because $a_{iC} \leq a_{iV}$ and $a_{iC} \leq a_{iV}$ we can bound this by

$$\frac{a_{iV}}{a_{iV}} \left((1 - 2c_i) \sigma - 2 \right) \leq \nabla \phi_\sigma(\mathbf{c})_i \leq \frac{a_{iV}}{a_{iV}} \left((1 - 2c_i) \sigma + 1 \right).$$

So if $c_i = 0$, we get that $\nabla \phi_\sigma(\mathbf{c})_i > 0$ when $\sigma > 2$. And if $c_i = 1$, we get that $\nabla \phi_\sigma(\mathbf{c})_i < 0$ when $\sigma > 1$.

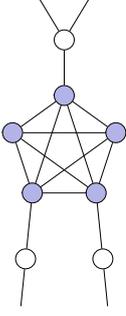


Figure 1: A simple subnetwork consisting of a clique with tails connecting it to the rest of the network. The clique (shaded nodes) is not a local optimum of conductance, but it is a local optimum of σ -conductance when $\sigma > 0.131$.

This means that when $\sigma > 2$ all discrete communities satisfy the KKT conditions, and from the sign of the gradient we can see that they are not local maxima. Furthermore, $\phi_\sigma(\mathbf{c})$ is a concave function, so it has no saddle points (see the proof of Theorem 2). This means that all discrete communities are local minima of ϕ_σ . ■

Conversely, the result from Section 2.1 generalizes to σ -conductance,

Theorem 2 *When $\sigma \geq 0$, all strict local minima \mathbf{c} of $\phi_\sigma(\mathbf{c})$ constrained to $0 \leq c_i \leq 1$ are discrete. Furthermore, if $\sigma > 0$ then all local minima are discrete.*

Proof By the argument from Section 2.1. When $\sigma > 0$ it is always the case that $\alpha_3 < 0$, so there are no saddle points or plateaus, and all local minima are discrete. ■

As an example application of σ -conductance, consider the network in Figure 1. In this network, the clique is not a local optimum of regular conductance. This is because the gradient for the adjacent nodes with degree 2 is always negative, regardless of the conductance of the community. However, for σ -conductance this gradient becomes positive when $\sigma > \phi_\sigma(\mathbf{c})$, in this case when $\sigma > 0.131$. In other words, with higher σ , adjacent nodes with low degree are no longer considered part of otherwise tightly connected communities such as cliques.

3. Algorithms

We now introduce two simple algorithms for the local optimization of conductance and σ -conductance, analyze their computational complexity and provide an exact performance guarantee for a class of communities. Then we look at a procedure for the automatic selection of a value for σ .

3.1 Projected Gradient Descent

Perhaps the simplest possible method for constrained continuous optimization problems is projected gradient descent. This is an iterative algorithm, where in each step the solution is moved in the direction of the negative gradient, and then this solution is projected so as to satisfy the constraints.

In our case, we start from an initial community containing only the seeds,

$$\mathbf{c}^{(0)} = \mathbf{s},$$

where $\mathbf{s} = [S]$ is a sparse vector indicating the seed node(s). Then in each subsequent iteration we get

$$\mathbf{c}^{(t+1)} = p(\mathbf{c}^{(t)} - \gamma^{(t)} \nabla \phi(\mathbf{c}^{(t)})).$$

This process is iterated until convergence. The step size $\gamma^{(t)}$ can be found with line search. The gradient $\nabla \phi$ is given by

$$\nabla \phi(\mathbf{c})_i = \frac{a_{AV} a_{cc}}{a_{cV}^2} - 2 \frac{a_{ic}}{a_{cV}}.$$

And the projection p onto the set of valid communities is defined by

$$p(\mathbf{c}) = \underset{\mathbf{c}', \text{ s.t. } 0 \leq c'_i \leq 1, s_i \leq c'_i}{\text{argmin}} \|\mathbf{c} - \mathbf{c}'\|_2^2,$$

which simply amounts to

$$p(\mathbf{c}) = \max(\mathbf{s}, \min(\mathbf{1}, \mathbf{c})).$$

This function clips values above 1 to 1, and values below s_i to s_i . Since $s_i \geq 0$ this also enforces that $c_i \geq 0$.

The complete algorithm is given in Algorithm 1. If a discrete community is desired, as a final step, we might threshold the vector \mathbf{c} . But as shown in Theorem 2 the found community is usually already discrete.

3.2 Expectation-Maximization

The connection to k -means clustering suggests that it might be possible to optimize conductance using an Expectation-Maximization algorithm similar to Lloyd's algorithm for k -means clustering. Intuitively, the algorithm would work as follows:

- **E step** assign each node i to the community if and only if its squared distance to the mean is less than λ_i .
- **M step** set the community mean to the weighted centroid of all nodes in the community.

These steps are alternated until convergence. Since both these steps do not increase the objective value, the algorithm is guaranteed to converge.

If the community after some iterations is C , then, as in the previous section, we can fill in the optimal mean into the E step, to obtain that a node i should be part of the community if

$$K(i, i) + \frac{a_{cC}/a_{cV} + \sigma}{a_{cV}} - 2 \frac{a_{iC}/a_{iV} + \sigma c_i}{a_{cV}} < \lambda_i.$$

When $\lambda_i = K(i, i)$, this condition is equivalent to

$$\nabla \phi_\sigma(C) < 0.$$

Algorithm 1 Projected Gradient Descent conductance optimization (PGDC)

Input: A set S of seeds of seeds, a graph G , a constant $\sigma \geq 0$.

- 1: $\mathbf{s} \leftarrow [S]$
- 2: $\mathbf{c}^{(0)} \leftarrow \mathbf{s}$
- 3: $t \leftarrow 0$
- 4: **repeat**
- 5: $\gamma^{(t)} \leftarrow \text{LineSearch}(\mathbf{c}^{(t)})$
- 6: $\mathbf{c}^{(t+1)} = p(\mathbf{c}^{(t)} - \gamma^{(t)} \nabla \phi_\sigma(\mathbf{c}^{(t)}))$
- 7: $t \leftarrow t + 1$
- 8: **until** $\mathbf{c}^{(t-1)} = \mathbf{c}^{(t)}$
- 9: $C \leftarrow \{i \in V \mid \mathbf{c}_i^{(t)} \geq 1/2\}$

function LineSearch(\mathbf{c})

- 1: $\gamma^* \leftarrow 0$, $\phi^* \leftarrow \phi_\sigma(\mathbf{c})$
 - 2: $\mathbf{g} \leftarrow \nabla \phi_\sigma(\mathbf{c})$
 - 3: $\gamma \leftarrow 1 / \max(|\mathbf{g}|)$
 - 4: **repeat**
 - 5: $\mathbf{c}' \leftarrow p(\mathbf{c} - \gamma \mathbf{g})$
 - 6: **if** $\phi_\sigma(\mathbf{c}') < \phi^*$ **then**
 - 7: $\gamma^* \leftarrow \gamma$, $\phi^* \leftarrow \phi_\sigma(\mathbf{c}')$
 - 8: **end if**
 - 9: $\gamma \leftarrow 2\gamma$
 - 10: **until** $\mathbf{c}'_i \in \{0, 1\}$ for all i with $g_i \neq 0$
 - 11: **return** γ^*
-

Algorithm 2 EM conductance optimization (EMC)

Input: A set S of seeds, a graph G , a constant $\sigma \geq 0$.

- 1: $C^{(0)} \leftarrow S$
 - 2: $t \leftarrow 0$
 - 3: **repeat**
 - 4: $C^{(t+1)} = \{i \mid \nabla \phi_\sigma(C^{(t)})_i < 0\} \cup S$
 - 5: $t \leftarrow t + 1$
 - 6: **while** $C^{(t)} < C^{(t-1)}$
-

This leads us to the EM community finding algorithm, Algorithm 2.

By taking $\sigma = 0$ we get that nodes are assigned to the community exactly if the gradient $\nabla\phi(C)_i$ is negative. So, this EM algorithm is very similar to projected gradient descent with an infinite step size in each iteration. The only difference is for nodes with $\nabla\phi(C)_i = 0$, which in the EMC algorithm are always assigned to the background, while in PGD their membership of the community is left unchanged compared to the previous iteration.

Of course, we have previously established that $\sigma = 0$ does not lead to a valid kernel (this doesn't preclude us from still using the EM algorithm). In the case that $\sigma > 0$ there is an extra barrier for adding nodes not currently in the community, and an extra barrier for removing nodes that are in the community. This is similar to the effect that increasing σ has on the gradient of ϕ .

3.3 Computational Complexity

Both methods require the computation of the gradient in each iteration. This computation can be done efficiently. The only nodes for which the gradient of the conductance is negative are the neighbors of nodes in the current community, and the only nodes for which a positive gradient can have an effect are those in the community. So the gradient doesn't need to be computed for other nodes. For the other nodes the gradient depends on the number of edges to the community, and on the node's degree. Assuming that the node degree can be queried in constant time, the total time per iteration is proportional to the size of the one-step-neighborhood of the community, which is of the order of the volume of the community. If the node degrees are not known, then the complexity increases to be proportional to the volume of the one-step-neighborhood of the community, though this is a one-time cost, not a per iteration cost.

As seen in Section 3.5, for dense and isolated communities, the number of iterations is bounded by the diameter of the community. In general we can not guarantee such a bound, but in practice the number of iterations is always on the order of the diameter of the recovered community.

For very large datasets, the computation of the gradient can still be expensive, even though it is a local operation. Therefore, we restrict the search to a set of 1000 nodes near the seed. This set N is formed by starting with the seed, and repeatedly adding all neighbors of nodes in N , until the set would contain more than 1000 nodes. In this last step we only add the nodes with the highest a_{iN}/a_{iV} so that the final set contains exactly 1000 nodes.

3.4 Choosing σ

In Section 2.2 we introduced the σ parameter, and we have shown that larger values of σ lead to more local optima. This leaves the question of choosing the value of σ .

One obvious choice is $\sigma = 0$, which means that ϕ_σ is exactly the classical conductance. Another choice would be to pick the smallest σ that leads to a positive definite kernel. But this is a global property of the network, that is furthermore very expensive to compute.

Instead, we try several different values of σ for each seed, and then pick the community with the highest density, that is, the community C with the largest $acc/|C|^2$.

3.5 Exactly Recoverable Communities

We now take a brief look at which kinds of communities can be exactly recovered with gradient descent and expectation-maximization. Suppose that we wish to recover a community C^* from a seeds set S , and assume that this community is connected. Denote by $d(i)$ the shortest path distance from a node $i \in C^*$ to any seed node, in the subnetwork induced by C^* .

First of all, since both algorithms grow the community from the seeds, we need to look at subcommunities $C \subseteq C^*$ centered around the seeds, by which we mean that $d(i) \leq d(j)$ for all nodes $i \in C$ and $j \in C^* \setminus C$.

Secondly, we need the community to be sufficiently densely connected to be considered a community in the first place; but at the same time the community needs to be separated from the rest of the network. Again, because the communities are grown, we require that this holds also for subcommunities that are grown from the seeds,

Definition 3 A community C^* is dense and isolated with threshold σ if for all subsets $C \subseteq C^*$ centered around the seeds S :

- $2a_{iC}/a_{iV} > acc/acv - \sigma$ for all nodes $i \in C$, and
- $2a_{iC}/a_{iV} \leq acc/acv - \sigma$ for all nodes $i \notin C^*$.

Some examples of communities that satisfy this property are cliques and quasi-cliques that are only connected to nodes of high degree.

Now denote by D_n the set of nodes i in C^* with $d(i) \leq n$. Clearly $D_0 = S$, and because the community is connected there is some n^* such that $D_{n^*} = C^*$.

We first look at the expectation-maximization algorithm.

Theorem 4 If C^* is dense and isolated, then the iterates of the EMC algorithm satisfy $C^{(t)} = D_t$.

Proof The proof proceeds by induction. For $t = 0$, the only nodes i with $d(i) = 0$ are the seeds, and $C^{(0)} = S$ by definition.

Now suppose that $C^{(t)} = D_t$. Then for any node i there are three possibilities.

- $i \in D_{t+1}$; then because C^* is dense and D_t is centered around the seeds, $2a_{iC^{(t)}}/a_{iV} > 1 - \phi_\sigma(C^{(t)})$. This implies that $\nabla\phi_\sigma(C^{(t)})_i < 0$.
- $i \in C^* \setminus D_{t+1}$; then there are no edges from D_t to i , since otherwise the shortest path distance from i to a seed would be $t + 1$. So $a_{iC^{(t)}} = 0$, which implies that $\nabla\phi_\sigma(C^{(t)})_i \geq 0$.
- $i \notin C^*$; then because C^* is isolated, $2a_{iC^{(t)}}/a_{iV} \leq 1 - \phi_\sigma(C^{(t)})$, which implies that $\nabla\phi_\sigma(C^{(t)})_i \geq 0$.

hence $\nabla\phi_\sigma(C^{(t)})_i < 0$ if $i \in D_{t+1}$, and $\nabla\phi_\sigma(C^{(t)})_i \geq 0$ otherwise. This means that $C^{(t+1)} = D_{t+1}$. ■

For the projected gradient descent algorithm from Section 3.1 we have an analogous theorem,

Theorem 5 *If C^* is dense and isolated, then the iterates of PGDC satisfy $\mathbf{c}^{(t)} = [D_t]$.*

Proof The proof proceeds by induction, and is analogous to the proof of Theorem 4. For $t = 0$, the only nodes i with $d(i) = 0$ are the seeds, and $\mathbf{c}^{(0)} = \mathbf{s} = [S]$ by definition.

Now suppose that $\mathbf{c}^{(t)} = [D_t]$. We have already shown that $\nabla\phi_\sigma(C^{(t)})_i < 0$ if and only if $i \in D_{t+1}$. This means that after projecting onto the set of valid communities, only the membership of nodes in D_{t+1} can increase. Since nodes in D_t already have membership 1, and nodes not in D_{t+1} already have membership 0, they are not affected.

Let $\gamma_{\max} = \max_{i \in D_{t+1}} -1/\nabla\phi_\sigma(\mathbf{c}^{(t)})_i$. Clearly if $\gamma^{(t)} \geq \gamma_{\max}$, then $c_i^{(t)} - \gamma^{(t)}\nabla\phi_\sigma(\mathbf{c}^{(t)})_i > 1$ for all nodes $i \in D_{t+1}$, and hence $p(\mathbf{c}^{(t)} - \gamma^{(t)}\nabla\phi_\sigma(\mathbf{c}^{(t)})) = [D_{t+1}]$. So to complete the proof, we only need to show that the optimal step size found with line search is indeed (at least) γ_{\max} .

Suppose that $\gamma^{(t)} < \gamma_{\max}$ leads to the optimal conductance. Then there is a node $i \in D_{t+1}$ with fractional membership, $0 < c_i^{(t+1)} < 1$. By repeated application of Theorem 2 we know that there is a discrete community C' with $\phi_\sigma(C') = \phi_\sigma(\mathbf{c}^{(t+1)})_i$, and furthermore $\phi_\sigma(C' \setminus \{i\}) = \phi_\sigma(C' \cup \{i\})$. The latter can only be the case if $\nabla\phi_\sigma(C')_i = 0$. Because the only nodes whose membership has changed compared to $\mathbf{c}^{(t)}$ are those in $D_{t+1} \setminus D_t$, it follows that C' contains all nodes with distance at most t to the seeds, as well as some nodes with distance $t+1$ to the seeds. This means that C' is centered around the seeds, and so $\nabla\phi_\sigma(C')_i > 0$. This is a contradiction, which means that $\gamma^{(t)} \geq \gamma_{\max}$ must be the optimum. ■

As a corollary, since $D_{n^*+1} = D_{n^*}$, both the EMC and the PGDC algorithm will halt, and exactly recover C^* .

The notion of dense and isolated community is weakly related to that of (α, β) -cluster (Mishra et al., 2008) (without the technical assumption that each node has a self-loop): C is an (α, β) -cluster, with $0 \leq \alpha < \beta \leq 1$ if $a_{iC} \geq \beta|C|$ for i in C , $a_{iC} \leq \alpha|C|$ for i outside C .

The definition of dense and isolated community depends on the degree of the nodes while that of (α, β) -cluster does not. As a consequence, not all maximal cliques of a graph are in general dense and isolated communities while they are (α, β) -clusters. For instance, a maximal clique linked to an external isolated node, that is, a node of degree 1, is not dense and isolated.

In general one can easily show that if C is dense and isolated and

$$\min_{i \in C} a_{iV} > \max_{i \notin C, a_{iC} > 0} a_{iV}$$

$$\beta = \frac{1 - \phi(C)}{2|C|} \min_{i \in C} a_{iV}$$

then C is an (α, β) -cluster with

$$\alpha = \frac{1 - \phi(C)}{2|C|} \max_{i \notin C, a_{iC} > 0} a_{iV}.$$

and

Dataset	#node	#edge	clus.c.	#comm	$ C $	$\phi(C)$
LFR (om=1)	5000	25125	0.039	101	49.5	0.302
LFR (om=2)	5000	25123	0.021	146	51.4	0.534
LFR (om=3)	5000	25126	0.016	191	52.4	0.647
LFR (om=4)	5000	25117	0.015	234	53.4	0.717
Karate	34	78	0.103	2	17.0	0.141
Football	115	613	0.186	12	9.6	0.402
Pol.Blogs	1490	16715	0.089	2	745.0	0.094
Pol.Books	105	441	0.151	3	35.0	0.322
Flickr	35313	3017530	0.030	171	4336.1	0.682
Amazon	334863	925872	0.079	151037	19.4	0.554
DBLP	317080	1049866	0.128	13477	53.4	0.622
Youtube	1134890	2987624	0.002	8385	13.5	0.916
LiveJournal	3997962	34681189	0.045	287512	22.3	0.937
Orkut	3072441	117185083	0.014	6288363	14.2	0.977
CYC/Gavin 2006	6230	6531	0.121	408	4.7	0.793
CYC/Krogan 2006	6230	7075	0.075	408	4.7	0.733
CYC/Collins 2007	6230	14401	0.083	408	4.7	0.997
CYC/Costanzo 2010	6230	57772	0.022	408	4.7	0.996
CYC/Hopkins 2011	6230	10093	0.030	408	4.7	0.999
CYC/all	6230	80506	0.017	408	4.7	0.905

Table 1: Overview of the datasets used in the experiments. For each dataset we consider three different sets of communities.

4. Experiments

To test the proposed algorithms, we assess their performance on various networks. We also perform experiments on recent state-of-the-art algorithms based on the diffusion method which also optimize conductance.

4.1 Algorithms

Specifically, we perform a comparative empirical analysis of the following algorithms.

1. PGDC. The projected gradient descent algorithm for optimizing σ -conductance given in Algorithm 1. We show the results for two variants: PGDC-0 with $\sigma = 0$ and PGDC-d where σ is chosen to maximize the community's density as described in Section 3.4.
2. EMC. The Expectation Maximization algorithm for optimizing σ -conductance described in Section 2. We consider the variants EMC-0 with $\sigma = 0$ and EMC-d where σ is chosen automatically.

3. YL. The algorithm by Yang and Leskovec (2012) (with conductance as scoring function), based on the diffusion method. It computes an approximation of the personalized Page Rank graph diffusion vector (Andersen et al., 2006). The values in this

vector are divided by the degree of the corresponding nodes, and the nodes are sorted in descending order by their values. The ranking induces a one dimensional search space of communities C_k , called a sweep, defined by the sequence of prefixes of the sorted list, that is, the k top ranked nodes, for $k = 1, \dots, |V|$. The smallest k whose C_k is a 'local optimum' of conductance is computed and C_k is extracted. Local optima of conductance over the one dimensional space $C_1, C_2, \dots, C_{|V|}$ are computed using a heuristic. For increasing $k = 1, 2, \dots$ $\phi(C_k)$ is measured. When $\phi(C_k)$ stops decreasing at k^* this is a 'candidate point' for a local minimum. It becomes a selected local minimum if $\phi(C_k)$ keeps increasing after k^* and eventually becomes higher than $\alpha\phi(C_k)$, otherwise it is discarded. $\alpha = 1.2$ is shown to give good results and is also used in our experiments. Yang and Leskovec (2012) show that finding the local optima of the sweep curve instead of the global optimum gives a large improvement over previous local spectral clustering methods by Andersen and Lang (2006) and by Spielman and Teng (2004).

4. HK. The algorithm by Kloster and Gleich (2014), also based on the diffusion method. Here, instead of using the Personalized PageRank score, nodes are ranked based on a Heat Kernel diffusion score (Chung, 2007). We use the implementation made available by Kloster and Gleich (2014), which tries different values of the algorithm's parameters t and ϵ , and picks the community with the highest conductance among them. The details are in section 6.2 of (Kloster and Gleich, 2014). Code is available at <https://www.cs.purdue.edu/homes/dgleich/codes/hkgrow>.
5. PPR. The pprpush algorithm by Andersen and Lang (2006) based on the personalized Page Rank graph diffusion. Compared to YL instead of finding a local optimum of the sweep, the method looks for a global optimum, and hence often finds larger communities. We use the implementation included with the HK method.

4.2 Datasets

4.2.1 ARTIFICIAL DATASETS

The first set of experiments we performed is on artificially generated networks with a known community structure. We use the LFR benchmark (Lancichinetti et al., 2008). We used the parameter settings $\eta=5000$ $\mu=0.3$ $k=10$ $\maxk=50$ $t1=2$ $t2=1$ $\minc=20$ $\maxc=100$ $on=2500$, which means that the graph has 5000 nodes, and between 20 and 100 communities, each with between 10 and 50 nodes. Half of the nodes, 2500 are a member of multiple communities. We vary the overlap parameter (om), which determines how many communities these nodes are in. More overlap makes the problem harder.

4.2.2 SOCIAL AND INFORMATION NETWORK DATASETS WITH GROUND TRUTH

We use five social and information network datasets with ground-truth from the SNAP collection (Leskovec and Krevl, 2014). These datasets are summarized in Table 1. For each dataset we list the number of nodes, number of edges and the clustering coefficient. We consider all available ground truth communities with at least 3 nodes.

Yang and Leskovec (2012) also defined a set of top 5000 communities for each dataset. These are communities with a high combined score for several community goodness metrics,

among which is conductance. We therefore believe that communities in this set are biased to be more easy to recover by optimizing conductance, and therefore do not consider them here. Results with these top 5000 ground truth communities are available in tables 1–3 in the supplementary material².

In addition to the SNAP datasets we also include the Flickr social network dataset (Wang et al., 2012).

4.2.3 PROTEIN INTERACTION NETWORK DATASETS

We have also run experiments on protein interaction networks of yeast from the BioGRID database (Stark et al., 2006). This database curates networks from several different studies. We have constructed networks for Gavin et al. (2006), Krogan et al. (2006), Collins et al. (2007), Costanzo et al. (2010), Hoppins et al. (2011), as well as a network that is the union of all interaction networks confirmed by physical experiments.

As ground truth communities we take the CYC2008 catalog of protein complexes for each of the networks (Pu et al., 2009).

4.2.4 OTHER DATASETS

Additionally we used some classical datasets with known communities: Zachary's karate club Zachary (1977); Football: A network of American college football games (Girvan and Newman, 2002); Political books: A network of books about US politics (Krebs, 2004); and Political blogs: Hyperlinks between weblogs on US politics (Adamic and Glance, 2005). These datasets might not be very well suited for this problem, since they have very few communities.

4.3 Results

In all our experiments we use a single seed node, drawn uniformly at random from the community. We have also performed experiments with multiple seeds; the results of those experiments can be found in the supplementary material.

To keep the computation time manageable we have performed all experiments on a random sample of 1000 ground-truth communities. For datasets with fewer than 1000 communities, we include the same community multiple times with different seeds.

Since the datasets here considered have information about ground truth communities, a natural external validation criterion to assess the performance of algorithms on these datasets is to compare the community produced by an algorithm with the ground truth one. In general, that is, when ground truth information is not available, this task is more subtle, because it is not clear what is a good external validation metric to evaluate a community (Yang and Leskovec, 2015).

We measure quality performance with the F_1 score, which for community finding can be defined as

$$F_1(C, C^*) = 2 \frac{|C \cap C^*|}{|C| + |C^*|},$$

² The supplementary material is available from <http://cs.ru.nl/~tvanlaarhoven/conductance2016>

Dataset	PGDC-0	PGDC-d	EMC-0	EMC-d	YL	HK	PPR
LFR (om=1)	0.967	0.185	0.868	0.187	0.203	0.040	0.041
LFR (om=2)	0.483	0.095	0.293	0.092	0.122	0.039	0.041
LFR (om=3)	0.275	0.085	0.158	0.083	0.110	0.037	0.039
LFR (om=4)	0.178	0.074	0.100	0.072	0.092	0.032	0.034
Karate	0.831	0.472	0.816	0.467	0.600	0.811	0.914
Football	0.792	0.816	0.766	0.805	0.816	0.471	0.283
Pol.Blogs	0.646	0.141	0.661	0.149	0.017	0.661	0.535
Pol.Books	0.596	0.187	0.622	0.197	0.225	0.641	0.663
Flickr	0.098	0.027	0.097	0.027	0.013	0.054	0.118
Amazon	0.470	0.522	0.425	0.522	0.493	0.245	0.130
DBLP	0.356	0.369	0.317	0.371	0.341	0.214	0.210
Youtube	0.089	0.251	0.073	0.248	0.228	0.037	0.071
LiveJournal	0.067	0.262	0.059	0.259	0.183	0.035	0.049
Orkut	0.042	0.231	0.033	0.231	0.171	0.057	0.033
CYC/Gavin 2006	0.474	0.543	0.455	0.543	0.526	0.336	0.294
CYC/Krogan 2006	0.410	0.513	0.364	0.511	0.504	0.229	0.169
CYC/Collins 2007	0.346	0.429	0.345	0.429	0.416	0.345	0.345
CYC/Costanzo 2010	0.174	0.355	0.172	0.351	0.314	0.170	0.170
CYC/Hopkins 2011	0.368	0.405	0.368	0.405	0.424	0.368	0.368
CYC/all	0.044	0.459	0.017	0.459	0.425	0.016	0.002

Table 2: Average F_1 score between recovered communities and ground-truth. The best result for each dataset is indicated in bold, as are the results not significantly worse according to a paired T-test (at significance level 0.01).

where C is the recovered community and C^* is the ground truth one. A higher F_1 score is better, with 1 indicating a perfect correspondence between the two communities.

Note that a seed node might be in multiple ground truth communities. In this case we only compare the recovered community to the true community that we started with. If a method finds another ground truth community this is not detected, and so it results in a low F_1 score.

We also analyze the output of these algorithms with respect to the conductance of produced communities and their size. Results on the run time of the algorithms are reported in the supplementary material (Table 4).

Figure 2 shows the F_1 scores as a function of the parameter σ . Table 2 shows the F_1 scores comparing the results of the methods to the true communities. Table 3 shows the mean size of the found communities, and Table 4 their conductance.

In general, results of these experiments indicate that on real-life networks, our methods based on continuous relaxation of conductance, PPR and HK produce communities with good conductance, but all are less faithful to the ground truth when the network contains many small communities. In PGDC, EMC the automatic choice of σ helps to achieve results closer to the ground truth, and the built-in tendency of YL to favor small communities helps

Dataset	PGDC-0	PGDC-d	EMC-0	EMC-d	YL	HK	PPR
LFR (om=1)	52.3	6.2	71.8	6.3	5.9	2410.0	2366.2
LFR (om=2)	93.2	5.6	292.8	6.4	4.9	2404.4	2311.6
LFR (om=3)	104.7	5.6	451.5	6.4	5.0	2399.4	2283.7
LFR (om=4)	108.9	4.9	530.2	5.7	4.9	2389.1	2262.0
Karate	20.0	8.0	24.1	8.0	8.8	16.7	17.1
Football	14.7	9.5	16.2	9.4	8.8	40.5	56.5
Pol.Blogs	515.6	110.1	538.9	118.1	7.2	492.7	1051.1
Pol.Books	37.8	5.2	43.1	5.7	6.7	49.3	53.4
Flickr	639.9	73.0	644.6	73.5	12.9	174.2	1158.1
Amazon	25.2	5.6	45.6	5.7	6.4	88.8	20819.9
DBLP	61.9	5.4	83.5	6.1	6.0	55.0	24495.0
Youtube	340.6	19.4	474.2	21.3	9.3	147.9	20955.5
LiveJournal	243.7	5.5	309.3	5.9	10.8	153.2	3428.7
Orkut	245.1	17.9	344.7	19.1	11.1	212.0	1634.0
CYC/Gavin 2006	19.7	3.4	34.3	3.5	3.1	236.7	621.9
CYC/Krogan 2006	48.4	7.0	138.8	9.4	3.7	723.8	1756.3
CYC/Collins 2007	202.9	19.5	207.2	19.5	2.6	192.0	189.4
CYC/Costanzo 2010	540.2	58.1	564.2	66.5	5.9	1058.8	942.9
CYC/Hopkins 2011	229.9	110.1	235.5	110.4	4.3	295.2	295.2
CYC/all	657.5	16.0	841.9	17.0	9.6	2795.2	5786.0

Table 3: Average size of the recovered communities.

as well. On the other hand, on networks with large communities our methods, PPR and HK work best. On the artificial LFR data continuous relaxation of conductance seems to work best. This result indicates that the LFR model of ‘what is a community’ is somehow in agreement with the notion of local community as local optimum of the continuous relaxation of conductance. However, as observed in recent works like (Jenb et al., 2015), the LFR model does not seem to represent the diverse characteristics of real-life communities.

We have included tables of the standard deviation in the supplementary material. Overall, the standard deviation in cluster size is of the same order of magnitude as the mean. The standard deviation of the conductance is around 0.1 for LFR datasets, 0.2 for the SNAP datasets and 0.3 for the CYC datasets. It is not surprising that the variance is this high, because the communities vary a lot in size and density.

Results on these datasets can be summarized as follows.

4.3.1 ARTIFICIAL LFR DATASETS

On these datasets, HK and PPR tend to find communities that are much too large, with small conductance but also with low F_1 scores. This happens because the LFR networks are small, and the methods are therefore able to consider a large part of the nodes in the network.

On the other hand, YL always starts its search at small communities, and it stops early, so the communities it finds are smaller than the ground truth ones on these networks.

Dataset	PGDC-0	PGDC-d	EMC-0	EMC-d	YL	HK	PPR
LFR (om=1)	0.301	0.750	0.304	0.749	0.755	0.250	0.273
LFR (om=2)	0.532	0.786	0.541	0.787	0.780	0.315	0.338
LFR (om=3)	0.589	0.793	0.587	0.793	0.781	0.333	0.354
LFR (om=4)	0.604	0.791	0.595	0.792	0.775	0.341	0.359
Karate	0.129	0.460	0.081	0.475	0.327	0.222	0.136
Football	0.277	0.356	0.274	0.362	0.385	0.244	0.155
Pol.Blogs	0.228	0.743	0.212	0.737	0.867	0.229	0.137
Pol.Books	0.140	0.622	0.107	0.611	0.571	0.127	0.065
Flickr	0.777	0.937	0.777	0.937	0.951	0.864	0.762
Amazon	0.181	0.464	0.180	0.463	0.402	0.081	0.053
DBLP	0.246	0.571	0.257	0.565	0.498	0.133	0.147
Youtube	0.601	0.765	0.711	0.759	0.700	0.201	0.341
LiveJournal	0.563	0.875	0.589	0.874	0.774	0.336	0.489
Orkut	0.718	0.916	0.731	0.917	0.928	0.750	0.711
CYC/Gavin 2006	0.614	0.734	0.611	0.732	0.735	0.532	0.500
CYC/Krogan 2006	0.466	0.626	0.469	0.620	0.617	0.325	0.265
CYC/Collins 2007	0.716	0.953	0.712	0.953	0.972	0.720	0.730
CYC/Costanzo 2010	0.759	0.931	0.755	0.929	0.934	0.672	0.646
CYC/Hoppins 2011	0.788	0.883	0.785	0.882	0.970	0.763	0.763
CYC/all	0.674	0.872	0.742	0.874	0.840	0.363	0.026

Table 4: Average conductance of the recovered communities.

The best F_1 results are achieved by PGDC with $\sigma = 0$, that is, when the continuous relaxation of conductance is used as the objective function. This method employs a more powerful optimizer than YL, so it is able to find a large community with a better conductance, but it still stops at the first local optimum. In the LFR datasets these optima are very clear, and correspond closely to the ground truth communities.

In all cases EMC shows similar or slightly worse performance compared to PGDC, so the gradient descend algorithm should be preferred.

The automatic choice of σ leads to communities which are of relatively small size. We believe that this happens because the nodes in LFR datasets all have exactly the same fraction of within community edges. Increasing σ suddenly makes the gradient for most of these nodes positive. In real networks there are often hubs that are more central to a community, with more connections to the community's nodes and to the seed. These hubs still can be found at higher values of σ .

4.3.2 SMALL REAL-LIFE SOCIAL NETWORKS WITH FEW COMMUNITIES (KARATE, FOOTBALL, BLOGS, BOOKS)

Our methods based on continuous relaxation of conductance yield the best F_1 results on the Football and Blog networks, while PPR performs best on the other two networks and achieves best overall conductance.

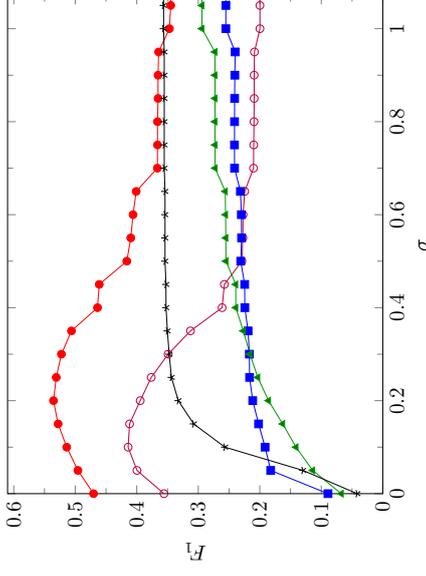


Figure 2: Average F_1 score as a function of the σ parameter on the SNAP datasets with the PGDC method.

4.3.3 LARGE SOCIAL NETWORK WITH BIG COMMUNITIES (FLICKR)

On this network, PPR achieves the best results both in terms of F_1 score as well as conductance. However the produced communities are about four times smaller than the ground truth communities, which have more than 4000 nodes. PGDC and EMC with $\sigma = 0$ yield communities of conductance similar to that of PPR communities, but their size is smaller (about 650 nodes). This happens because the algorithms are restricted to 1000 nodes around the seed, without this restriction larger communities would be found. Somewhat surprisingly HK produces communities of relatively small size (about 175 nodes). The automatic choice of σ yields to even smaller communities (about 64 nodes). The smallest size communities are produced by YL (about 13 nodes).

4.3.4 LARGE REAL-LIFE SNAP NETWORKS WITH MANY SMALL COMMUNITIES

On these networks the automatic choice of σ gives best results, consistently outperforming the other algorithms. In Figure 2, the F_1 score of PGDC and EMC as a function of σ is plotted. For some datasets a small value of σ works well, while for others a larger value of σ is better. Our procedure to choose σ produces results that are close to, but slightly below, the best a posteriori choice of σ . So on these networks the proposed procedure positively affects the performance of our algorithms. YL favors communities of small size less faithful to the ground truth. PGDC-0, EMC-0, PPR and HK 'explode', and produce very large communities. For our methods this 'explosion' is limited only because we limit the search to 1000 nodes near the seed. The ground-truth communities of these datasets have rather high conductance, and the networks have a very low clustering coefficient. In such a case,

communities have many links to nodes outside, hence conductance alone is clearly not suited to finding these type of local communities.

4.3.5 REAL-LIFE PROTEIN INTERACTION NETWORKS WITH VERY SMALL COMMUNITIES (CYC)

Also on these networks the automatic choice of σ gives best results. As expected, due to the very small size of the ground truth communities, YL also achieves very good results. The other algorithms tend to produce less realistic, large communities which have better conductance.

4.3.6 RUNNING TIME

The best performing algorithm with respect to running time is HK. PGDC and EMC are about four times slower with a fixed value of σ , and ten to twenty times slower when automatically determining σ . The running times are included in Table 4 of the supplementary material. All experiments were run on a 2.4GHz Intel XEON E7-4870 machine. Note that the different methods are implemented in different languages (our implementation is written in Octave, while HK and PPR are implemented in C++), so the running times only give an indication of the overall trend, and can not be compared directly.

4.3.7 TOP 5000 COMMUNITIES

Results with only the top 5000 ground truth communities available at the SNAP dataset collection are similar to the results with all communities. As expected, the F_1 score is much higher and the conductance of the recovered community is better. Because these ground truth communities have a better conductance, it is better to optimize conductance, that is to take $\sigma = 0$. As a consequence the performance of PGDC-0 and EMC-0 is better than that of PGDC-d and EMC-d for these communities. The full results are available in Tables 1-3 in the supplementary material.

5. Discussion

This paper investigated conductance as an objective function for local community detection from a set of seeds. By making a continuous relaxation of conductance we show how standard techniques such as projected gradient descent can be used to optimize it. Even though this is a continuous optimization problem, we show that the local optima are almost always discrete communities. We further showed how linking conductance with kernel weighted k -means clustering leads to the new σ -conductance objective function and to simple yet effective algorithms for local community detection by seed expansion.

We provided a formalization of a class of good local communities around a set of seeds and showed that the proposed algorithms can find them. We suspect that these communities can also be exactly retrieved using local community algorithms based on the diffusion method, but do not yet have a proof. The condition that such communities should be centered around the seeds raises the question of how to find such seeds. Although various works have studied seed selection for diffusion based algorithms, such as Kloumann and Kleinberg

(2014), this problem remains to be investigated in the context of local community detection by σ -conductance optimization using PGDC and EMC.

Our experimental results indicate the effectiveness of direct optimization of a continuous relaxation of σ -conductance using gradient descent and expectation maximization. In our algorithms we used community density as a criterion to choose σ . This resulted to be a good choice for the performance of our algorithms on the SNAP networks. It would be interesting to investigate also other criteria to choose σ . Conversely, the fact that maximum density is a good criterion for selecting σ implies that it might also be directly optimized as an objective for finding communities.

On some datasets, when optimizing normal conductance, that is, with $\sigma = 0$, our methods sometimes find very large communities. These communities will have a very good conductance, but they do not correspond well to the ground truth. In some sense the optimizer is ‘too good’, and conductance is not the best criterion to describe these communities. A better objective would perhaps take into account the size of the community more explicitly, but this needs to be investigated further.

In this paper we have used gradient descent, a first order optimization method which utilizes only the objective function’s gradient. More advanced optimization methods also use second derivatives or approximations of those. We believe that such methods will not bring a large advantage compared to gradient descent, because during the optimization many coordinates are at the boundary value 0 or 1, and second derivatives would not help to locate these boundary points. Other constrained optimizers such as interior point methods have the problem that they need to inspect a much larger part of the network, potentially all of it, because intermediate steps have nonzero membership for all nodes.

Acknowledgments

We thank the reviewers for their useful comments. This work has been partially funded by the Netherlands Organization for Scientific Research (NWO) within the EW TOP Comparison 1 project 612.001.352.

References

- Lada Adamic and Natalie Glance. The political blogosphere and the 2004 U.S. election: Divided they blog. In *LinkKDD ’05: Proceedings of the 3rd international workshop on Link discovery*, pages 36–43, 2005.
- Reid Andersen and Kevin J. Lang. Communities from seed sets. In *Proceedings of the 13th International Conference on World Wide Web*, WWW ’06, pages 223–232, New York, NY, USA, 2006. ACM. ISBN 1-59593-323-9.
- Reid Andersen, Fan Chung, and Kevin Lang. Local graph partitioning using pagerank vectors. In *Foundations of Computer Science, 2006. FOCS’06. 47th Annual IEEE Symposium on*, pages 475–486. IEEE, 2006.
- Haim Avron and Lior Horvath. Community detection using time-dependent personalized pagerank. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 1795–1803, 2015.

- Shuchi Chawla, Robert Krauthgamer, Ravi Kumar, Yuval Rabani, and D. Sivakumar. On the hardness of approximating multicut and sparsest-cut. In *Proceedings of the 20th Annual IEEE Conference on Computational Complexity*, CCC '05, pages 144–153, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2364-1.
- Jiyang Chen, Osmar Zaiane, and Randy Goebel. Local community identification in social networks. In *Social Network Analysis and Mining, 2009. ASONAM'09. International Conference on Advances in*, pages 237–242. IEEE, 2009.
- Qiong Chen, Ting-Ting Wu, and Ming Fang. Detecting local community structures in complex networks based on local degree central nodes. *Physica A: Statistical Mechanics and its Applications*, 392(3):529–537, 2013.
- Fan Chung. The heat kernel as the pagerank of a graph. *Proceedings of the National Academy of Sciences*, 104(50):19735–19740, 2007.
- Aaron Clauset. Finding local community structure in networks. *Physical review E*, 72(2):026132, 2005.
- Sean Collins, et al. Towards a comprehensive atlas of the physical interactome of *saccharomyces cerevisiae*. *Molecular Cellular Proteomics*, pages 600381–600200, 2007.
- Michael Costanzo, et al. The genetic landscape of a cell. *Science*, 327(5964):425–431, 2010.
- Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvalues: a multilevel approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(11):1944–1957, November 2007. ISSN 0162-8828.
- Santo Fortunato. Community detection in graphs. *Physics Reports*, 486:75–174, 2010.
- Ullas Gargi, Wenjun Lu, Vahab S Mirrokni, and Sangho Yoon. Large-scale community detection on youtube for topic discovery and exploration. In *ICWSM*, 2011.
- Anne-Claude Gavin, et al. Proteome survey reveals modularity of the yeast cell machinery. *Nature*, 440(7084):631–636, 2006.
- Michelle Girvan and Mark E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99(12):7821–7826, 2002.
- David F. Gleich and C. Seshadhri. Vertex neighborhoods, low conductance cuts, and good seeds for local community methods. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 597–605, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1462-6.
- Suzanne Hoppins, et al. A mitochondrial-focused genetic interaction map reveals a scaffold-like complex required for inner membrane organization in mitochondria. *The Journal of Cell Biology*, 195(2):323–340, 2011.
- Lucas GS Jeub, Prakash Balachandran, Mason A Porter, Peter J Mucha, and Michael W Mahoney. Think locally, act locally: Detection of small, medium-sized, and large communities in large networks. *PHYSICAL REVIEW E Phys Rev E*, 91:012821, 2015.
- U Kang and Christos Faloutsos. Beyond “caveman communities”: Hubs and spokes for graph compression and mining. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 300–309. IEEE, 2011.
- Kyle Kloster and David F. Gleich. Heat kernel based community detection. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 1386–1395, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2956-9.
- Isabel M Kloumann and Jon M Kleinberg. Community membership identification from small seed sets. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1366–1375. ACM, 2014.
- Valdis Krebs. New political patterns. Editorial, 2004.
- Nevan J. Krogan, et al. Global landscape of protein complexes in the yeast *saccharomyces cerevisiae*. *Nature*, 440(7084):637–643, 2006.
- Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78:046110, 2008.
- Andrea Lancichinetti, Filippo Radicchi, José J Ramasco, Santo Fortunato, et al. Finding statistically significant communities in networks. *PLoS one*, 6(4):e18961, 2011.
- Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. <http://snap.stanford.edu/data>, June 2014.
- Jure Leskovec, Kevin J Lang, Anirban Dasgupta, and Michael W Mahoney. Statistical properties of community structure in large social and information networks. In *Proceedings of the 17th international conference on World Wide Web*, pages 695–704. ACM, 2008.
- Jure Leskovec, Kevin J Lang, and Michael Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th international conference on World wide web*, pages 631–640. ACM, 2010.
- László Lovász and Miklós Simonovits. The mixing rate of Markov chains, an isoperimetric inequality, and computing the volume. In *Foundations of Computer Science, 1990. Proceedings., 31st Annual Symposium on*, pages 346–354. IEEE, 1990.
- Michael W Mahoney, Lorenzo Orecchia, and Nisheeth K Vishnoi. A local spectral method for graphs: With applications to improving graph partitions and exploring data graphs locally. *The Journal of Machine Learning Research*, 13(1):2339–2365, 2012.
- Nina Mishra, Robert Schreiber, Isabelle Stanton, and Robert Endre Tarjan. Finding strongly knit clusters in social networks. *Internet Mathematics*, 5(1):155–174, 2008.

- Alan Mislove, Bimal Viswanath, Krishna P. Gummadi, and Peter Druschel. You are who you know: Inferring user profiles in online social networks. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, WSDM '10, pages 251–260. New York, NY, USA, 2010. ACM. ISBN 978-1-60558-889-6.
- Shuye Pu, Jessica Wong, Brian Turner, Emerson Cho, and Shooshana J Wodak. Up-to-date catalogues of yeast protein complexes. *Nucleic Acids Research*, 37(3):825–831, 2009.
- Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007. ISSN 15740137.
- Jiri Šima and Satu Elisa Schaeffer. On the NP-completeness of some graph cluster measures. In *SCOPSEM 2006: Theory and Practice of Computer Science*, pages 530–537. Springer, 2006.
- Daniel A Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 81–90. ACM, 2004.
- Chris Stark, Bobby-Joe Breikreutz, Teresa Reguly, Lorrie Boucher, Ashton Breikreutz, and Mike Tyers. Biogrid: a general repository for interaction datasets. *Nucleic Acids Research*, 34(Database-Issue):535–539, 2006.
- Xufei Wang, Lei Tang, Huan Liu, and Lei Wang. Learning with multi-resolution overlapping communities. *Knowledge and Information Systems (KAMIS)*, 2012.
- Joyce Jiyoung Whang, David F Gleich, and Indejit S Dhillon. Overlapping community detection using seed set expansion. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 2099–2108. ACM, 2013.
- Xiao-Ming Wu, Zhenguo Li, Anthony M So, John Wright, and Shi-Fu Chang. Learning with partially absorbing random walks. In *Advances in Neural Information Processing Systems*, pages 3077–3085, 2012.
- Yuhao Wu, Ruoming Jin, Jing Li, and Xiang Zhang. Robust local community detection: On free rider effect and its elimination. *Proc. VLDB Endow.*, 8(7):798–809, February 2015. ISSN 2150-8097.
- Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics*, MDS '12, pages 3:1–3:8. ACM, 2012. ISBN 978-1-4503-1546-3.
- Jaewon Yang and Jure Leskovec. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, 42(1):181–213, 2015. ISSN 0219-1377.
- Wayne W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.
- Zeyuan A Zhu, Silvio Lattanzi, and Vahab Mirrokni. A local algorithm for finding well-connected clusters. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 396–404, 2013a.
- Zeyuan Allen Zhu, Silvio Lattanzi, and Vahab Mirrokni. Local graph clustering beyond Cheeger’s inequality. *arXiv preprint arXiv:1304.8132*, 2013b.

Megaman: Scalable Manifold Learning in Python

James McQueen

*Department of Statistics
University of Washington
Seattle, WA 98195-4322, USA*

JMCQ@UW.EDU

Marina Meilä

*Department of Statistics
University of Washington
Seattle, WA 98195-4322, USA*

MMP@STAT.WASHINGTON.EDU

Jacob VanderPlas

*e-Science Institute
University of Washington
Seattle, WA 98195-4322, USA*

JAKEVDP@UW.EDU

Zhongyue Zhang

*Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195-4322, USA*

ZHANGZ6@CS.WASHINGTON.EDU

Editor: Alexandre Gramfort

Abstract

Manifold Learning (ML) is a class of algorithms seeking a low-dimensional non-linear representation of high-dimensional data. Thus, ML algorithms are most applicable to high-dimensional data and require large sample sizes to accurately estimate the manifold. Despite this, most existing manifold learning implementations are not particularly scalable. Here we present a Python package that implements a variety of manifold learning algorithms in a modular and scalable fashion, using fast approximate neighbors searches and fast sparse eigendecompositions. The package incorporates theoretical advances in manifold learning, such as the unbiased Laplacian estimator introduced by Coifman and Lafon (2006) and the estimation of the embedding distortion by the Riemannian metric method introduced by Perrault-Joncas and Meilä (2013). In benchmarks, even on a single-core desktop computer, our code embeds millions of data points in minutes, and takes just 200 minutes to embed the main sample of galaxy spectra from the Sloan Digital Sky Survey—consisting of 0.6 million samples in 3750-dimensions—a task which has not previously been possible.

Keywords: manifold learning, dimension reduction, Riemannian metric, graph embedding, scalable methods, python

1. Motivation

We propose *megaman*, a new Python package for scalable manifold learning. This package is designed for performance, while inheriting the functionality of *scikit-learn*'s well-designed API (Buitinck et al., 2013).

2. Downloading and installation

megaman is publicly available at: <https://github.com/mmp2/megaman>. *megaman*'s required dependencies are *numpy*, *scipy*, and *scikit-learn*, but for optimal performance FLANN, *cython*, *pyang* and the C compiler *gcc* are also required. For unit tests and integration *megaman* depends on *nose*. The most recent *megaman* release can be installed along with its dependencies using the cross-platform `conda`¹ package manager:

```
$ conda install megaman --channel=conda-forge
```

Alternatively, *megaman* can be installed from source by downloading the source repository and running:

```
$ python setup.py install
```

With *nose* tests installed, unit tests can be run with:

```
$ make test
```

3. Logical structure and classes overview

embeddings The manifold learning algorithms are implemented in their own classes inheriting from a base class. Included are `SpectralEmbedding`, which implements *Laplacian Eigenmaps* (Belkin and Niyogi, 2002) and *Diffusion Maps* (Nadler et al., 2006), `L TSA` (Zhang and Zha, 2004), `LocallyLinearEmbedding` (Roweis and Saul, 2000), and `Isomap` (Bernstein et al., 2000). Geometric operations common to many or all embedding algorithms (such as computing distances, Laplacians) are implemented by the `Geometry` class. A `Geometry` object is passed or created inside every embedding class. In particular, `RiemannianMetric` produces the estimated Riemannian metric via the method of Perrault-Joncas and Meilä (2013). `eigendecomposition` (module) provides a unified (function) interface to the different eigendecomposition methods provided in *scipy*.

For background of manifold learning, as well as *megaman*'s design philosophy, please see McQueen et al. (2016).

4. Quick start

```
from megaman.geometry import Geometry
from megaman.embedding import SpectralEmbedding
from sklearn.datasets import make_swiss_roll
```

```
X = make_swiss_roll( 10000 ) # generate input data
radius = 1.1 # kernel bandwidth and for graph construction
# a Geometry object encapsulates generic geometric operations
geom = Geometry(
    adjacency_kwds = {'radius':3*radius}, # neighborhood radius
    adjacency_method = 'cyflann', # fast approximate neighbors
```

¹. Conda can be downloaded at <http://conda.pydata.org/miniconda.html>.

```

affinity_method = 'gaussian',          # Gaussian kernel
affinity_kwds = {'radius':radius},     # kernel bandwidth
laplacian_method = 'geometric')       # unbiased Laplacian
SE = SpectralEmbedding( # embedding algorithm & params
    n_components=2,                # embed into 2 dimensions
    eigen_solver='amg',           # eigensolver
    geom=geom)                    # pass the geometric information
Y = SE.fit_transform(X)           # perform embedding

```

The last two instructions are identical to their analogous instructions in `scikit-learn`. Full documentation is available from the `megaman` website at: <http://mmp2.github.io/megaman/>

5. Benchmarks

The one other popular comparable implementation of manifold learning algorithms is the `scikit-learn` package. To make the comparison as fair as possible, we choose the `SpectralEmbedding` method for the comparison, with radius-based neighborhoods and the *Locally-Optimized Block-Preconditioned Conjugate Gradient (lobpcg)* eigensolver. Note, too, that with the default settings, `scikit-learn` would perform slower than in our experiments.

We display total embedding time (including time to compute the graph G , the Laplacian matrix and the embedding 2) for `megaman` versus `scikit-learn`, as the number of samples N varies or the data dimension D varies (Figure 1). All benchmark computations were performed on a single desktop computer running Linux with 24.68GB RAM and a Quad-Core 3.07GHz Intel Xeon CPU. We use a relatively weak machine to demonstrate that our package can be reasonably used without high performance hardware. The experiments show that `megaman` scales considerably better than `scikit-learn`, even in the most favorable conditions for the latter: the memory footprint of `megaman` is smaller, even when `scikit-learn` uses sparse matrices internally. The advantages grow as the data size grows, whether it is w.r.t D or to N .

We also report run times on two real world data sets. The first is the `word2vec` data set ³ which contains feature vectors in 300 dimensions for about 3 million words and phrases, extracted from Google News. The vector representation was obtained via a multilayer neural network by Mikolov et al. (2013). The second data set contains galaxy spectra from the Sloan Digital Sky Survey ⁴ (Abazajian et al., 2009), preprocessed as described in Telford et al. (2016).

Dataset	Size N	Dimensions D	Distances	Embedding	R. metric	Total
Galaxies	0.7M	3750	190.5	8.9	0.1	199.5
Word2Vec	3M	300	107.9	44.8	0.6	153.3

Run time [min]

- For `megaman` we also compute the Riemannian metric estimate at each point; this time is negligible compared to the total time to obtain the embedding.
- The `word2vec` data used were from `GoogleNews-vectors-negative300.bin.gz` which can be downloaded from <https://code.google.com/archive/p/word2vec/>.
- The Sloan Digital Sky Survey data can be downloaded from www.sdss.org.

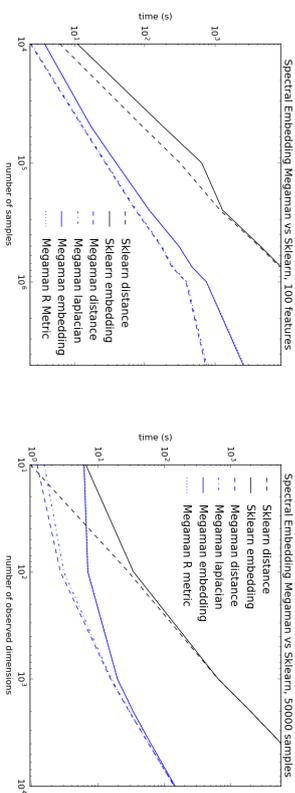


Figure 1: **Run time vs. data set size N** for fixed $D = 100$ (left) and **Run time vs. data set dimension D** for fixed $N = 50,000$ (right). The data is from a Swiss Roll (in 3 dimensions) with additional noise dimensions, embedded into $s = 2$ dimensions by the `SpectralEmbedding` algorithm. By $D = 10,000$ and $N = 1,000,000$ `scikit-learn` was unable to compute an embedding due to insufficient memory. All `megaman` run times (including time between distance and embedding) are faster than `scikit-learn`.

6. Conclusion

`megaman` puts in the hands of scientists and methodologists alike tools that enable them to apply state of the art manifold learning methods to data sets of realistic size. The package is extensible, modular, with an API familiar to `scikit-learn` users. Future development will be mainly in the direction of further scalability (Nystrom extension, parallelization) and expanding the data analytical tools (distance calculations, estimation of dimension, estimation of neighborhood radius, directed graph embedding).

We hope that by providing this package, non-linear dimension reduction will be benefit those who most need it: the practitioners exploring large scientific data sets.

Acknowledgments

We would like to acknowledge support for this project from the National Science Foundation (NSF grant IIS-9988642), the Multidisciplinary Research Program of the Department of Defense (MURI N00014-00-1-0637), the Department of Defense (62-7760 “DOD Unclassified Math”), the Moore/Sloan Data Science Environment grant, and funding from the Washington Research Foundation. This project grew from the Data Science Incubator program ⁵ at the University of Washington eScience Institute.

⁵ For more information on Data Science Incubator program see <http://data.uw.edu/incubator/>.

References

- K. N. Abazajian, J. K. Adelman-McCarthy, M. A. Agüeros, S. S. Allam, C. Allende Prieto, D. An, K. S. J. Anderson, S. F. Anderson, J. Annis, N. A. Bahcall, and et al. The Seventh Data Release of the Sloan Digital Sky Survey. *Astrophysical Journal Supplement Series*, 182:543–558, June 2009. doi: 10.1088/0067-0049/182/2/543.
- M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, Cambridge, MA, 2002. MIT Press.
- Mira Bernstein, Vin de Silva, John C. Langford, and Josh Tenenbaum. Graph approximations to geodesics on embedded manifolds. <http://web.mit.edu/cocosci/isomap/BdSLT.pdf>, December 2000.
- Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, et al. API design for machine learning software: experiences from the scikit-learn project. *arXiv preprint arXiv:1309.0238*, 2013.
- R. R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 30(1):5–30, 2006.
- James McQueen, Marina Meila, Jacob VanderPlas, and Zhongyue Zhang. megaman: Manifold learning with millions of points. *arXiv e-prints: 1603.02763*, March 2016.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, 2013.
- Boaz Nadler, Stephane Lafon, Ronald Coifman, and Ioannis Kevrekidis. Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 955–962, Cambridge, MA, 2006. MIT Press.
- D. Perrault-Joncas and M. Meila. Non-linear dimensionality reduction: Riemannian metric estimation and the problem of geometric discovery. *arXiv e-prints:1305.7255*, May 2013.
- Sam Roweis and Lawrence Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 2000.
- O. Grace Telford, Jacob Vanderplas, James McQueen, and Marina Meila. Metric embedding of Sloan galaxy spectra. (*in preparation*), 2016.
- Zhenyue Zhang and Hongyuan Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM J. Scientific Computing*, 26(1):313–338, 2004.

Kernel Estimation and Model Combination in A Bandit Problem with Covariates

Wei Qian

*School of Mathematical Sciences
Rochester Institute of Technology
Rochester, NY 14623, USA*

WXQSM@RIT.EDU

Yuhong Yang

*School of Statistics
University of Minnesota
Minneapolis, MN 55455, USA*

YYANG@STAT.UMN.EDU

Editor: Gábor Lugosi

Abstract

Multi-armed bandit problem is an important optimization game that requires an exploration-exploitation tradeoff to achieve optimal total reward. Motivated from industrial applications such as online advertising and clinical research, we consider a setting where the rewards of bandit machines are associated with covariates, and the accurate estimation of the corresponding mean reward functions plays an important role in the performance of allocation rules. Under a flexible problem setup, we establish asymptotic strong consistency and perform a finite-time regret analysis for a sequential randomized allocation strategy based on kernel estimation. In addition, since many nonparametric and parametric methods in supervised learning may be applied to estimating the mean reward functions but guidance on how to choose among them is generally unavailable, we propose a model combining allocation strategy for adaptive performance. Simulations and a real data evaluation are conducted to illustrate the performance of the proposed allocation strategy.

Keywords: contextual bandit problem, exploration-exploitation tradeoff, nonparametric regression, regret bound, upper confidence bound

1. Introduction

Following the seminal work by Robbins (1954), multi-armed bandit problems have been studied in multiple fields. The general bandit problem involves the following optimization game: A gambler is given l gambling machines, and each machine has an “arm” that the gambler can pull to receive the reward. The distribution of reward for each arm is unknown and the goal is to maximize the total reward over a given time horizon. If we define the regret to be the reward difference between the optimal arm and the pulled arm, the equivalent goal of the bandit problem is to minimize the total regret. Under a standard setting, it is assumed that the reward of each arm has fixed mean and variance throughout the time horizon of the game. Some of the representative work for standard bandit problem includes Lai and Robbins (1985), Berry and Fristedt (1985), Gittins (1989) and Auer et al. (2002).

See Cesa-Bianchi and Lugosi (2006) and Bubeck and Cesa-Bianchi (2012) for bibliographic remarks and recent overviews on bandit problems.

Different variants of the bandit problem motivated by real applications have been studied extensively in the past decade. One promising setting is to assume that the reward distribution of each bandit arm is associated with some common external covariate. More specifically, for an l -armed bandit problem, the game player is given a d -dimensional external covariate $x \in \mathbb{R}^d$ at each round of the game, and the expected reward of each bandit arm given x has a functional form $f_i(x)$, $i = 1 \dots l$. We call this variant **multi-armed bandit problem with covariates**, or MABC for its abbreviation (MABC is also referred to as CMAB for contextual multi-armed bandit problem in the literature). The consideration of external covariates is potentially important in applications such as personalized medicine. For example, before deciding which treatment arm to be assigned to a patient, we can observe the patient prognostic factors such as age, blood pressure or genetic information, and then use such information for adaptive treatment assignment for best outcome. It is worth noting that the consideration of external covariate is recently further generalized to partial monitoring by Bartók and Szepesvári (2012).

The MABC problems have been studied under both parametric and nonparametric frameworks with various types of algorithms. The first work in a parametric framework appears in Woodroofe (1979) under a somewhat restrictive setting. A linear response bandit problem in more flexible settings is recently studied under a minimax framework (Goldenshluger and Zeevi, 2009; Goldenshluger and Zeevi, 2013). Empirical studies are also reported for parametric UCB-type algorithms (e.g., Li et al., 2010). The regret analysis of a special linear setting is given in e.g., Auer (2002), Chu et al. (2011) and Agrawal and Goyal (2013), in which the linear parameters are assumed to be the same for all arms while the observed covariates can be different across different arms.

MABC problems with the nonparametric framework are first studied by Yang and Zhiu (2002). They show that with histogram or K -nearest neighbor estimation, the function estimation is uniformly strongly consistent, and consequently, the cumulative reward of their randomized allocation rule is asymptotically equivalent to the optimal cumulative reward. Their notion of reward strong consistency has been recently established for a Bayesian sampling method (May et al., 2012). Notably, under the Hölder smoothness condition and a margin condition, the recent work of Perchet and Rigollet (2013) establishes a regret upper bound by arm elimination algorithms with the same order as the minimax lower bound of a two-armed MABC problem (Rigollet and Zeevi, 2010). A different stream of work represented by, e.g., Langford and Zhang (2007) and Dudík et al. (2011) imposes neither linear nor any smoothness assumption on the mean reward function; instead, they consider a class of (finitely many) policies, and the cumulative reward of the proposed algorithms is compared to the best of the policies. Interested readers are also referred to Bubeck and Cesa-Bianchi (2012, Section 4) and its bibliography remarks for studies from numerous different perspectives.

Another important line of development in the bandit problem literature (closely related to, but different from the setting of MABC) is to consider the arm space as opposed to the covariate space in MABC. It is assumed that there are infinitely many arms, and at each round of the game, the player has the freedom to play one arm chosen from the arm space. Like MABC, the setting with the arm space can be studied from both parametric linear and

nonparametric frameworks. Examples of the linear parametric framework include Dani et al. (2008), Rusmevichentong and Tsiatis (2010) and Abbasi-Yadkori et al. (2011). Notable examples of the nonparametric framework (also known as the continuum-armed bandit problem) under the local or global Hölder and Lipschitz smoothness conditions are Kleinberg (2004), Auer et al. (2007), Kleinberg et al. (2007) and Brubeck et al. (2011). Abbasi-Yadkori (2009) studies a forced exploration algorithm over the arm space, which is applied to both parametric and nonparametric frameworks. Interestingly, Lu et al. (2010) and Slivkins (2011) consider both the arm space and the covariate space, and study the problem by imposing Lipschitz conditions on the joint space of arms and covariates.

Our work in this paper follows the nonparametric framework of MABC in Yang and Zhu (2002) and Rigollet and Zeevi (2010) with finitely many arms. One contribution in this work is to show that kernel methods enjoy estimation uniform strong consistency as well, which leads to strongly consistent allocation rules. Note that due to the dependence of the observations for each arm by the nature of the proposed randomized allocation strategy, it is difficult to apply the well-established kernel regression analysis results of i.i.d. or weak dependence settings (e.g., Devroye, 1978; Härdle and Uchmans, 1984; Hansen, 2008). New technical tools and arguments such as “chaining” are developed in this paper.

In addition, with the help of the Hölder smoothness condition, we provide a deeper understanding of the proposed randomized allocation strategy via a finite-time regret analysis. Compared with the result in Rigollet and Zeevi (2010) and Perchet and Rigollet (2013), our finite-time result remains sub-optimal in the minimax sense. Indeed, given Hölder smoothness parameter κ and total time horizon N , our expected cumulative regret upper bound is $O(N^{1-\frac{\kappa}{3+\kappa+d}})$ as compared to $O(N^{1-\frac{\kappa}{3+\kappa+d}})$ of Perchet and Rigollet (2013) (without the extra margin condition). The slightly sub-optimal rate can also be shown to apply to the histogram based randomized allocation strategy proposed in Yang and Zhu (2002). We tend to think that this rate is the best possible for these methods, reflecting to some extent the theoretical limitation of the randomized allocation strategy. In spite of this sub-optimality, our result explicitly shows both the bias-variance tradeoff and the exploration-exploitation tradeoff, which reflects the underlying nature of the proposed algorithm for the MABC problem. With a model combining strategy and dimension reduction technique to be introduced later, the kernel estimation based randomized allocation strategy can be quite flexible with wide potential practical use. Moreover, in Appendix A, we incorporate the kernel estimation into a UCB-type algorithm with randomization and show that its regret rate becomes minimax optimal up to a logarithmic factor.

One natural and interesting issue in the randomized allocation strategy in MABC is how to choose the modeling methods among numerous nonparametric and parametric estimators. The motivation of such a question shares the flavor of model aggregation/combining in statistical learning (see, e.g., Audibert, 2009; Rigollet and Tsybakov, 2012; Wang et al., 2014 and references therein). In the bandit problem literature, model combining is also quite relevant to the adversary bandit problem (e.g., Cesa-Bianchi and Lugosi, 2006; Auer et al., 2003). As a recent example, Mallard and Mimos (2011) study the history-dependent adversarial bandit to target the best among a pool of history class mapping strategies.

As an empirical solution to the difficulty in choosing the best estimation method for each arm in the randomized allocation strategy for MABC, we introduce a fully data-driven model combining technique motivated by the AFTER algorithm, which has shown success

both theoretically (Yang, 2004) and empirically (e.g., Zou and Yang, 2004; Wei and Yang, 2012). We integrate a model combining step by AFTER for reward function estimation into the randomized allocation strategy for MABC. Preliminary simulation results of combining various dimension reduction methods are reported in Qian and Yang (2012). However, no theoretical justification is given there. As another contribution of this paper, we present here new theoretical and numerical results on the proposed combining algorithm. In particular, the strong consistency of the model combining allocation strategy is established.

The rest of this paper is organized as follows. We present a general and flexible problem setup for MABC in Section 2. We describe the algorithm in Section 3 and study the strong consistency and the finite-time regret analysis of kernel estimation methods in Section 4. We also introduce a dimension reduction sub-procedure to handle the situation that the covariate dimension is high. The asymptotic results of the model combining allocation strategy is established in Section 5. We show in Section 6 and Section 7 the numerical performance of the proposed allocation strategy using simulations and a web-based news article recommendation data set, respectively. A brief conclusion is given in Section 8. The kernel estimation based UCB-type algorithm with randomization is described in Appendix A, all technical lemmas and proofs are given in Appendix B, and additional numerical results of the implemented algorithms are left in Appendix C.

2. Problem Setup

Suppose a bandit problem has l ($l \geq 2$) candidate arms to play. At each time point of the game, a d -dimensional covariate x is observed before we decide which arm to pull. Assume that the covariate x takes values in the hypercube $[0, 1]^d$. Also assume the (conditional) mean reward for arm i given x , denoted by $f_i(x)$, is uniformly upper bounded and unknown to game players. The observed reward is modeled as $f_i(x) + \epsilon_i$, where ϵ_i is a random error with mean 0.

Let $\{X_n, n \geq 1\}$ be a sequence of independent covariates generated from an underlying probability distribution P_X supported in $[0, 1]^d$. At each time $n \geq 1$, we need to apply a sequential allocation rule η to decide which arm to pull based on X_n and the previous observations. We denote the chosen arm by I_n and the observed reward of pulling the arm $I_n = i$ at time n by $Y_{i,n}$, $1 \leq i \leq l$. As a result, $Y_{n,n} = f_{I_n}(X_n) + \epsilon_n$, where ϵ_n is the random error with $E(\epsilon_n | X_n) = 0$. Different from Yang and Zhu (2002), the error ϵ_n may be dependent on the covariate X_n . Consider the simple scenario of online advertising where the response is binary (click: $Y = 1$; no click: $Y = 0$). Given an arm i and covariate $x \in [0, 1]$, suppose the mean reward function satisfies e.g., $f_i(x) = x$. Then it is easy to see that the distribution of the random error ϵ depends on x . In case of a continuous response, it is also well-known that heteroscedastic errors commonly occur.

By the previous definitions, we know that at time n , an allocation strategy chooses the arm I_n based on X_n and $(X_j, I_j, Y_{I_j,j})$, $1 \leq j \leq n-1$. To evaluate the performance of the allocation strategy, let $i^*(x) = \operatorname{argmax}_{1 \leq i \leq l} f_i(x)$ and $f^*(x) = f_{i^*(x)}(x)$ (any tie-breaking rule can be applied if there are ties). Without the knowledge of random error ϵ_j , the optimal performance occurs when $I_j = i^*(X_j)$, and the corresponding optimal cumulative reward given X_1, \dots, X_n can be represented as $\sum_{j=1}^n f^*(X_j)$. The cumulative mean reward of the applied allocation rule can be represented as $\sum_{j=1}^n f_{I_n}(X_j)$. Thus we can measure the

performance of an allocation rule η by the cumulative regret

$$R_n(\eta) = \sum_{j=1}^n (f^*(X_j) - f_{\eta_j}(X_j)).$$

We say the allocation rule η is strongly consistent if $R_n(\eta) = o(n)$ with probability one. Also, $R_n(\eta)$ is commonly used for finite-time regret analysis. In addition, define the per-round regret $r_n(\eta)$ by

$$r_n(\eta) = \frac{1}{n} \sum_{j=1}^n (f^*(X_j) - f_{\eta_j}(X_j)).$$

To maintain the readability for the rest of this paper, we use i only for bandit arms, j and n only for time points, r and s only for reward function estimation methods, and t and T only for the total number of times a specific arm is pulled.

3. Algorithm

In this section, we present the model-combining-based randomized allocation strategy. At each time $n \geq 1$, denote the set of past observations $\{(X_j, I_j, Y_{j,i}) : 1 \leq j \leq n-1\}$ by Z^n , and denote the arm i associated subset $\{(X_j, I_j, Y_{j,i}) : I_j = i, 1 \leq j \leq n-1\}$ by $Z^{i,n}$. For estimating the f_i 's, suppose we have m candidate regression estimation procedures (e.g., histogram, kernel estimation, etc.), and we denote the class of these candidate procedures by $\Delta = \{\delta_1, \dots, \delta_m\}$. Let $\hat{f}_{i,n,r}$ denote the regression estimate of procedure δ_r based on $Z^{i,n}$, and let $\hat{f}_{i,n}$ denote the weighted average of $\hat{f}_{i,n,r}$'s, $1 \leq r \leq m$, by the model combining algorithm to be given. Let $\{\pi_n, n \geq 1\}$ be a decreasing sequence of positive numbers approaching 0, and assume that $(l-1)\pi_n < 1$ for all $n \geq 1$. The model combining allocation strategy includes the following steps.

STEP 1. Initialize with forced arm selections. Give each arm a small number of applications. For example, we may pull each arm n_0 times at the beginning by taking $I_1 = 1$, $I_2 = 2, \dots, I_l = l, I_{l+1} = 1, \dots, I_{2l} = l, \dots, I_{(n_0-1)l+1} = 1, \dots, I_{n_0l} = l$.

STEP 2. Initialize the weights and the error variance estimates. For $n = n_0l + 1$, initialize the weights by

$$W_{i,n,r} = \frac{1}{m}, \quad 1 \leq i \leq l, 1 \leq r \leq m,$$

and initialize the error variance estimates by e.g.,

$$\hat{v}_{i,n,r} = 1, \hat{v}_{i,n} = 1, \quad 1 \leq i \leq l, 1 \leq r \leq m.$$

STEP 3. Estimate the individual functions f_i for $1 \leq i \leq l$. For $n = n_0l + 1$, based on the current data $Z^{i,n}$, obtain $\hat{f}_{i,n,r}$ using regression procedure δ_r , $1 \leq r \leq m$.

STEP 4. Combine the regression estimates and obtain the weighted average estimates

$$\hat{f}_{i,n} = \sum_{r=1}^m W_{i,n,r} \hat{f}_{i,n,r}, \quad 1 \leq i \leq l.$$

STEP 5. Estimate the best arm, select and pull. For the covariate X_n , define $\hat{i}_n = \operatorname{argmax}_{1 \leq i \leq l} \hat{f}_{i,n}(X_n)$ (If there is a tie, any tie-breaking rule may apply). Choose an arm, with probability $1 - (l-1)\pi_n$ for arm \hat{i}_n (the currently most promising choice) and with probability π_n for each of the remaining arms. That is,

$$I_n = \begin{cases} \hat{i}_n, & \text{with probability } 1 - (l-1)\pi_n, \\ i, & \text{with probability } \pi_n, i \neq \hat{i}_n, 1 \leq i \leq l. \end{cases}$$

Then pull the arm I_n to receive the reward $Y_{I_n,n}$.

STEP 6. Update the weights and the error variance estimates. For $1 \leq i \leq l$, if $i \neq I_n$, let $W_{i,n+1,r} = W_{i,n,r}$, $1 \leq r \leq m$, $\hat{v}_{i,n+1,r} = \hat{v}_{i,n,r}$, $1 \leq r \leq m$, and $\hat{v}_{i,n+1} = \hat{v}_{i,n}$. If $i = I_n$, update the weights and the error variance estimates by

$$W_{i,n+1,r} = \frac{W_{i,n,r} \exp\left(-\frac{(\hat{f}_{i,n,r}(X_n) - Y_{i,n})^2}{2\hat{v}_{i,n,r}}\right)}{\sum_{k=1}^m W_{i,n,k} \exp\left(-\frac{(\hat{f}_{i,n,k}(X_n) - Y_{i,n})^2}{2\hat{v}_{i,n,k}}\right)}, \quad 1 \leq r \leq m,$$

$$\hat{v}_{i,n+1,r} = \frac{\sum_{k=n_0l+1}^n (Y_{I_n,k} - \hat{f}_{i,n,k}(X_k))^2 I(I_k = i)}{\sum_{k=n_0l+1}^n 1} \vee \underline{v}, \quad 1 \leq r \leq m,$$

and

$$\hat{v}_{i,n+1} = \sum_{r=1}^m W_{i,n+1,r} \hat{v}_{i,n+1,r},$$

where $I(\cdot)$ is the indicator function and \underline{v} is a small positive constant (to ensure that $\hat{v}_{i,n+1,r}$ is nonzero). In practice, we set $\underline{v} = 10^{-16}$.

STEP 7. Repeat steps 3 - 6 for $n = n_0l + 2, n_0l + 3, \dots$, and so on.

In the allocation strategy above, step 1 and step 2 initialize the game and pull each arm the same number of times. Step 3 and step 4 estimate the reward function for each arm using several regression methods, and combine the estimates by a weighted average scheme. Clearly, the importance of these regression methods are differentiated by their corresponding weights. Step 5 performs an enforced randomization algorithm, which gives preference to the arm with the highest reward estimate. This type of arm randomization is also known as the ϵ -greedy algorithm. Step 6 is the key to the model combining algorithm, which updates the weights for the recently played arm. Its weight updating formula implies that if the estimated reward from a regression method turns out to be far away from the observed reward, we penalize this method by decreasing its weight, while if the estimated reward turns out to be accurate, we reward this method by increasing its weight. Note that our combining approach has few tuning parameters except for what is already included in the individual regression procedures.

4. Kernel Regression Procedures

In this section, we consider the special case that kernel estimation is used as the only modeling method. The primary goals include: 1) establishing the uniform strong consistency of kernel estimation under the proposed allocation strategy; 2) performing the finite-time regret analysis. To extend the applicability of kernel methods, a dimension reduction sub-procedure is described in Section 4.3.

4.1 Strong Consistency

We focus on the Nadaraya-Watson regression and study its strong consistency under the proposed allocation strategy. Given a regression method $\hat{y}_n \in \Delta$ and an arm i , we say it is strongly consistent in L_∞ norm for arm i if $\|\hat{f}_{i,n} - f_i\|_\infty \rightarrow 0$ a.s. as $n \rightarrow \infty$.

Assumption 0. *The errors satisfy a (conditional) moment condition that there exist positive constants v and c such that for all integers $k \geq 2$ and $n \geq 1$,*

$$E(|\varepsilon_n|^k | X_n) \leq \frac{k!}{2} v^2 c^{k-2}$$

almost surely.

Assumption 0 means that the error distributions, which could depend on the covariates, satisfy a moment condition known as refined Bernstein condition (e.g., Birgé and Massart, 1998, Lemma 8). Normal distribution, for instance, satisfies the condition. Bounded errors trivially meet the requirement. Therefore, Assumption 0 is met in a wide range of real applications, and will also be used in the next section for understanding strong consistency of model combining procedures. Note that heavy-tailed distributions are also possible for bandit problems (Bubeck et al., 2013).

Given a bandit arm $1 \leq i \leq l$, at each time point n , define $J_{i,n} = \{j : f_j = i, 1 \leq j \leq n-1\}$, the set of past time points at which arm i is pulled. Let $M_{i,n}$ denote the size of the set $J_{i,n}$. For each $u = (u_1, u_2, \dots, u_d) \in R^d$, define $\|u\|_\infty = \max\{|u_1|, |u_2|, \dots, |u_d|\}$. Consider two natural conditions on the mean reward functions and the covariate density as follows.

Assumption 1. *The functions f_i are continuous on $[0, 1]^d$ with $A =: \sup_{1 \leq i \leq l} \sup_{x \in [0, 1]^d} (f^*(x) - f_i(x)) < \infty$.*

Assumption 2. *The design distribution P_X is dominated by the Lebesgue measure with a continuous density $p(x)$ uniformly bounded above and away from 0 on $[0, 1]^d$; that is, $p(x)$ satisfies $\underline{c} \leq p(x) \leq \bar{c}$ for some positive constants $\underline{c} \leq \bar{c}$.*

In addition, consider a multivariate nonnegative kernel function $K(u) : R^d \rightarrow R$ that satisfies Lipschitz, boundedness and bounded support conditions.

Assumption 3. *For some constants $0 < \lambda < \infty$,*

$$|K(u) - K(u')| \leq \lambda \|u - u'\|_\infty$$

for all $u, u' \in R^d$.

Assumption 4. *There exist constants $L_1 \leq L$, $c_3 > 0$ and $c_4 \geq 1$ such that $K(u) = 0$ for $\|u\|_\infty > L$, $K(u) \geq c_3$ for $\|u\|_\infty \leq L_1$, and $K(u) \leq c_4$ for all $u \in R^d$.*

Let h_n denote the bandwidth, where $h_n \rightarrow 0$ as $n \rightarrow \infty$. The Nadaraya-Watson estimator of $f_i(x)$ is

$$\hat{f}_{i,n+1}(x) = \frac{\sum_{j \in J_{i,n+1}} Y_{j,i} K\left(\frac{x-X_j}{h_n}\right)}{\sum_{j \in J_{i,n+1}} K\left(\frac{x-X_j}{h_n}\right)}. \quad (1)$$

Theorem 1. *Suppose Assumptions 0-4 are satisfied. If the bandwidth sequence $\{h_n\}$ and the decreasing sequence $\{\pi_n\}$ are chosen to satisfy $h_n \rightarrow 0$, $\pi_n \rightarrow 0$ and*

$$\frac{nh_n^{2d}\pi_n^d}{\log n} \rightarrow \infty,$$

then the Nadaraya-Watson estimators defined in (1) are strongly consistent in L_∞ norm for the functions f_i .

Note that since checking L_∞ norm strong consistency of kernel methods is more challenging than that of histogram methods, new technical tools are necessarily developed to establish the strong consistency (as seen in the proof of Lemma 3 and Theorem 1 in Appendix B).

4.2 Finite-Time Regret Analysis

Next, we provide a finite-time regret analysis for the Nadaraya-Watson regression based randomized allocation strategy. To understand the regret cumulative rate, define a modulus of continuity $\omega(h; f_i)$ by

$$\omega(h; f_i) = \sup\{|f_i(x_1) - f_i(x_2)| : \|x_1 - x_2\|_\infty \leq h\}.$$

For technical convenience of guarding against the situation that the denominator of (1) is extremely small (which might occur with a non-negligible probability due to arm selection), in this subsection, we replace $K(\cdot)$ in (1) with the uniform kernel $I(\|u\|_\infty \leq L)$ when

$$\sum_{j \in J_{i,n+1}} K\left(\frac{x-X_j}{h_n}\right) < c_5 \sum_{j \in J_{i,n+1}} I(\|x-X_j\|_\infty \leq Lh_n) \quad (2)$$

for some small positive constant $0 < c_5 < 1$. Given $0 < \delta < 1$ and the total time horizon N , we define a special time point n_δ by

$$n_\delta = \min\left\{n > n_0^\delta : \sqrt{\frac{16v^2 \log(8LN^2/\delta)}{cn(2Lh_n)^d \pi_n}} \leq \frac{c_5 v^2}{c} \text{ and } \exp\left(-\frac{3cn(2Lh_n)^d \pi_n}{56}\right) \leq \frac{\delta}{4LN}\right\}. \quad (3)$$

Under the condition that $\lim_{n \rightarrow \infty} nh_n^d \pi_n / \log n = \infty$, we can see from (3) that $n_\delta/N \rightarrow 0$ as $N \rightarrow \infty$. As a result, if the total time horizon is long enough, we have $N > n_\delta$.

Theorem 2. *Suppose Assumptions 0-2 and 4 are satisfied and $\{\pi_n\}$ is a decreasing sequence. Assume $N > n_\delta$ and the kernel function is chosen as described in (2). Then with probability larger than $1 - 2\delta$, the cumulative regret satisfies*

$$R_N(\eta) < An_\delta + \sum_{n=n_\delta}^N \left(2 \max_{1 \leq i \leq l} \omega(Lh_n; f_i) + \frac{C_{N,\delta}}{\sqrt{nb_n^d \pi_n}} + (l-1)\pi_n \right) + A\sqrt{\frac{N}{2}} \log\left(\frac{1}{\delta}\right), \quad (4)$$

where $C_{N,\delta} = \sqrt{16c_d^2 v^2 \log(8lN^2/\delta)/c_d^2 \mathcal{L}(2L)^d}$.

It is interesting to see from the right hand side of (4) that the regret upper bound consists of several terms that make intuitive sense. The first term An_δ comes from the initial rough exploration. The second term has three essential components: $\max_{1 \leq i \leq l} \omega(Lh_n; f_i)$ is associated with the estimation bias, $C_{N,\delta}/\sqrt{nb_n^d \pi_n}$ conforms with the notion of estimation standard error, and $(l-1)\pi_n$ is the randomization error. The third term reflects the fluctuation of the randomization scheme. Such an upper bound explicitly illustrates both the bias-variance tradeoff and the exploration-exploitation tradeoff, which reflects the underlying nature of the proposed algorithm for the MABC problem.

Now we consider a smoothness assumption of the mean reward functions as follows.

Assumption 5. *There exist positive constants ρ and $\kappa \leq 1$ such that for each reward function f_i , the modulus of continuity satisfies*

$$\omega(h; f_i) \leq \rho h^\kappa.$$

Clearly, when $\kappa = 1$, Assumption 5 becomes Lipschitz continuity. As an immediate consequence of Theorem 2 and Assumption 5, we obtain the following result if we choose $h_n = \frac{1}{l} n^{-\frac{1}{3\kappa+d}}$ and $\pi_n = \frac{1}{l-1} n^{-\frac{1}{3\kappa+d/\kappa}}$.

Corollary 1. *Suppose the same conditions as in Theorem 2 are satisfied. Further assume Assumption 5 holds. Let $h_n = \frac{1}{l} n^{-\frac{1}{3\kappa+d}}$, $\pi_n = \frac{1}{l-1} n^{-\frac{1}{3\kappa+d/\kappa}}$ and $N > n_\delta$. Then with probability larger than $1 - 2\delta$, the cumulative regret satisfies*

$$R_N(\eta) < An_\delta + 2(2\rho + C_{N,\delta}^*) N^{1-\frac{1}{3\kappa+d/\kappa}} + A\sqrt{\frac{N}{2}} \log\left(\frac{1}{\delta}\right),$$

where $C_{N,\delta}^* = \sqrt{16c_d^2 v^2 (l-1) \log(8lN^2/\delta)/2^d c_d^2 \mathcal{L}}$.

In Corollary 1, the first term of the regret upper bound is dominated by the second term. Therefore, with high probability, the cumulative regret $R_N(\eta)$ increases at rate no faster than the order of $N^{1-\frac{1}{3\kappa+d/\kappa}} \log^{1/2} N$. This result can be seen more explicitly in Corollary 2, which gives an upper bound for the mean of $R_N(\eta)$. Note that by the definition of n_δ , the condition $N > n_\delta$ in Corollary 2 is satisfied if N is large enough.

Corollary 2. *Suppose the same conditions as in Theorem 2 are satisfied. Further assume Assumption 5 holds. Let $h_n = \frac{1}{l} n^{-\frac{1}{3\kappa+d}}$, $\pi_n = \frac{1}{l-1} n^{-\frac{1}{3\kappa+d/\kappa}}$ and $N > n_\delta$, where $\delta^* = N^{-\frac{1}{3\kappa+d/\kappa}}$. Then there exists a constant $C^* > 0$ (not dependent on N) such that the mean of cumulative regret satisfies*

$$ER_N(\eta) < C^* N^{1-\frac{1}{3\kappa+d/\kappa}} \log^{1/2} N.$$

As mentioned in Section 1, the derived regret cumulative rate in Corollary 2 is slightly slower than the minimax rate $N^{1-\frac{1}{3\kappa+d/\kappa}}$ obtained by Perchet and Rigollet (2013) (without assuming any extra margin condition). We tend to think this shows a limitation of the ϵ -greedy type approach. Nevertheless, with the help of the aforementioned model combining strategy along with the dimension reduction technique to be introduced in the next subsection, the kernel method based allocation can be quite flexible with potential practical use.

4.3 Dimension Reduction

When the covariate dimension is high, the Nadaraya-Watson estimation cannot be applied due to the curse of dimensionality. Next, we describe a dimension reduction sub-procedure to handle this situation, which is also discussed in Qian and Yang (2012). Different from the method there, a sparse dimension reduction technique will be included to handle cases with higher-dimensional covariates.

Recall that Z^n is the set of observations $\{(X_j, I_j, Y_{j,j}), 1 \leq j \leq n-1\}$, and $Z^{i,n}$ is the subset of Z^n where $I_j = i$. Then $M_{i,n}$ is the number of observations in $Z^{i,n}$. Let $X^{i,n}$ be the $M_{i,n} \times d$ design matrix consisting of all covariates in $Z^{i,n}$, and let $Y^{i,n} \in R^{M_{i,n}}$ be the observed reward vector corresponding to $X^{i,n}$. It is known that kernel methods do not perform well when the dimension of covariates is high. We want to apply some dimension reduction methods (see, e.g., Li, 1991; Chen et al., 2010) to $(X^{i,n}, Y^{i,n})$ first to obtain lower dimensional covariates before using kernel estimation.

Specifically, suppose for each arm i , there exists a reduction function $s_i : R^d \rightarrow R^{r_i}$ ($r_i < d$), such that $f_i(x) = g_i(s_i(x))$ for some function $g_i : R^{r_i} \rightarrow R$. Clearly, if the reduction function s_i is known, $s_i(x)$ can be treated like the new lower-dimensional covariate, with which the kernel methods can be applied to find the estimate of g_i , and hence f_i . However, s_i is generally unknown in practice, and it is necessary to first obtain the estimate of s_i . In addition, we assume that s_i is a linear reduction function in the sense that $s_i(x) = B_i^T x$, where $B_i \in R^{d \times r_i}$ is a dimension reduction matrix. It is worth mentioning that s_i is not unique, i.e., $s_i(x) = \tilde{A} B_i^T x$ is a valid reduction function for any full rank matrix $\tilde{A} \in R^{r_i \times r_i}$. Therefore, it suffices to estimate the dimension reduction subspace $\text{span}(B_i)$ spanned by the columns of B_i , and obtain $\hat{s}_{i,n}(x) = \hat{B}_{i,n}^T x$, where $\hat{B}_{i,n} \in R^{d \times r_i}$ is one basis matrix of the estimated subspace at time n , and $\hat{s}_{i,n}$ is the estimate of s_i .

Dimension reduction methods such as sliced inverse regression (also known as SIR, see Li, 1991) can be applied to $(X^{i,n}, Y^{i,n})$ to obtain $\hat{B}_{i,n}$. In practice, it is convenient to have $X^{i,n}$ work on the standardized scale (i.e., the sample mean is zero and the sample covariance matrix is the identity matrix; Li, 1991; Cook, 2007). Suppose the Nadaraya-Watson estimation is used with $K_j(u) : R^{r_i} \rightarrow R$ being a multivariate symmetric kernel function for arm i . Recall $J_{i,n} = \{j : I_j = i, 1 \leq j \leq n-1\}$ is the set of past time points at which arm i is pulled. Then, we can obtain $\hat{f}_{i,n}$ with the following steps.

Step 1. Transform $X^{i,n}$ to the standardized-scale matrix $X_*^{i,n}$: transform the original covariates X_j 's by $X_j^* = \hat{\Sigma}_{i,n}^{-1/2}(X_j - \bar{X}_{i,n})$ for every $j \in J_{i,n}$, where $\bar{X}_{i,n}$ and $\hat{\Sigma}_{i,n}$ are the sample mean vector and the sample covariance matrix of $X^{i,n}$, respectively.

Step 2. Apply a dimension reduction method to $(\hat{X}_{i,n}^{*T}, Y^{i,n})$ to obtain the estimated $d \times r_i$ dimension reduction matrix $B_{i,n}^*$, where $B_{i,n}^{*T} B_{i,n}^* = I_{r_i}$. For example, we can apply SIR (Li, 1991) to obtain $\hat{B}_{i,n}^*$ by using the MATLAB package LDR (Cook et al., 2011, available at <https://sites.google.com/site/11lianarforzani/ldr-package>)

Step 3. Given $x \in R^d$, let $x^* = \hat{\Sigma}_{i,n}^{-1/2}(x - \bar{X}_{i,n})$ be the transformed x at the standardized scale. The Nadaraya-Watson estimator of $f_i(x)$ is

$$\hat{f}_{i,n}(x) = \frac{\sum_{j \in I_{i,n}} Y_{i,j} K_i \left(\frac{\hat{B}_{i,n}^{*T} x^* - \hat{B}_{i,n}^{*T} X_j^*}{h_{n-1}} \right)}{\sum_{j \in I_{i,n}} K_i \left(\frac{\hat{B}_{i,n}^{*T} x^* - \hat{B}_{i,n}^{*T} X_j^*}{h_{n-1}} \right)}. \quad (5)$$

In addition to estimating the reward function for each arm, it is sometimes of interest to know which variables contribute to the reward for each arm, and some sparse dimension reduction techniques can be applied. In particular, Chen et al. (2010) propose the coordinate-independent sparse estimation (CISE) to give sparse dimension reduction matrix such that the estimated coefficients of some predictors are zero for all reduction directions (i.e., some row vectors in $\hat{B}_{i,n}^*$ become 0). When the SIR objective function is used, the corresponding CISE method is denoted by CIS-SIR. To obtain $\hat{B}_{i,n}^*$ in Step 2 above using CIS-SIR, we can apply the MATLAB package CISE (Chen et al., 2010, available at <http://www.stat.nyu.edu/~sgf-stackx/>).

The simulation example in Section 6 and the real data example in Section 7 both use the algorithms described here. The simulation example is implemented in MATLAB and the real data example is implemented in C++. The major source code illustrating the proposed algorithms is available upon request.

5. Strong Consistency in Model Combining Based Allocation

Next, we consider the general case that multiple function estimation methods are used for model combining. In general, it is technically difficult to verify strong consistency in L_∞ norm for a regression method. Also, practically, it is likely that some methods may give good estimation for only a subset of the arms, but performs poorly for the rest. Not knowing which methods work well for which arms, we proposed the combining algorithm in Section 3 to address this issue. We will show that even in the presence of bad-performing regression methods, the strong consistency of our allocation strategy still holds if for any given arm, there is at least one good regression method included for combining.

Given an arm i , let $N_t^{(i)} = \inf\{n : \sum_{j=n_0+1}^n I(U_j = i) \geq t\}$, $t \geq 1$, be the earliest time point where arm i is pulled exactly t times after the forced sampling period. For notation brevity, we use N_t instead of $N_t^{(i)}$ in the rest of this section. Consider the assumptions as follows.

Assumption A. Given any arm $1 \leq i \leq l$, the candidate regression procedures in Δ can be categorized into one of the two subsets denoted by Δ_{i1} (non-empty) and Δ_{i2} . All procedures

in Δ_{i1} are strongly consistent in L_∞ norm for arm i , while procedures in Δ_{i2} are less well-performing in the sense that for each procedure δ_s in Δ_{i2} , there exist a procedure δ_r in Δ_{i1} and some constants $b > 0.5$, $c_1 > 0$ such that

$$\liminf_{T \rightarrow \infty} \frac{\sum_{t=1}^T (\hat{f}_{i,N_t, s}(X_{N_t}) - f_i(X_{N_t}))^2 - \sum_{t=1}^T (\hat{f}_{i,N_t, r}(X_{N_t}) - f_i(X_{N_t}))^2}{\sqrt{T}(\log T)^b} > c_1$$

with probability one.

Assumption B. The mean functions satisfy $A = \sup_{1 \leq i \leq l} \sup_{x \in [0,1]^d} (f_i^*(x) - f_i(x)) < \infty$.

Assumption C. $\|\hat{f}_{i,n,r} - f_i\|_\infty$ is upper bounded by a constant c_2 for all $1 \leq i \leq l$, $n \geq n_0 l + 1$ and $1 \leq r \leq m$.

Assumption D. The variance estimates $\hat{v}_{i,n}$ are upper bounded by a positive constant q with probability one for all $1 \leq i \leq l$ and $n \geq n_0 l + 1$.

Assumption E. The sequence $\{\pi_n, n \geq 1\}$ satisfies that $\sum_{n=1}^\infty \pi_n$ diverges.

Note that Assumption A is automatically satisfied if all the regression methods happen to be strongly consistent (i.e., Δ_{i2} is empty). When a bad-performing method does exist, Assumption A requires that the difference of the mean square errors between a good-performing method and a bad-performing method decreases slower than the order of $(\log T)^b / \sqrt{T}$. If a parametric method δ_s in Δ is based on a wrong model, $\sum_{t=1}^T (\hat{f}_{i,N_t, s}(X_{N_t}) - f_i(X_{N_t}))^2$ is of order T , and then the requirement in Assumption A is met. For an inefficient nonparametric method, the enlargement of the mean square error by the order larger than $(\log T)^b / \sqrt{T}$ is natural to expect. Assumption B is a natural condition in the context of our bandit problem. Assumptions C and D are immediately satisfied if the response is bounded and the estimator is, e.g., a weighted average of some previous observations. Assumption E ensures that N_t is finite as shown in Lemma 5 in the Appendix. As implied in Lemma 5, if we are allowed to play the game infinitely many times, each arm will be pulled beyond any given integer. This guarantees that each ‘‘inferior’’ arm can be pulled reasonably often to ensure enough exploration.

Theorem 3. Under Assumption 0 and Assumptions A-E, the model combining allocation strategy is strongly consistent.

With Theorem 3, one is safe to explore different models or methods in estimating the mean reward functions that may or may not work well for some or all arms. The resulting per-round regret can be much improved if good methods (possibly different for different arms) are added in.

6. Simulations

In this section, we intend to illustrate the dimension reduction function estimation procedures described in Section 4.3 for bandit problem with multivariate covariates. Two

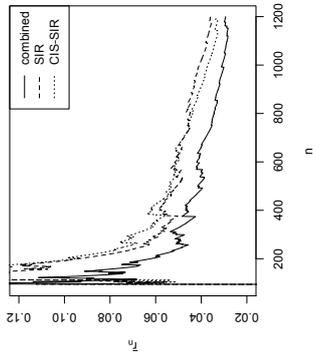


Figure 1: Averaged per-round regret from combining SIR and CIS-SIR.

readily available MATLAB packages for dimension reduction are used: LDR package (Cook et al., 2011) for SIR, and CJSE package (Chen et al., 2010) for CIS-SIR. The kernel used is the Gaussian kernel

$$K(t) = \exp\left(-\frac{\|t\|_2^2}{2}\right).$$

We consider a three-arm bandit model with $d = 10$. Assume that at each time n , the covariate is $X_n = (X_{n1}, X_{n2}, \dots, X_{nd})^T$, and X_{ni} 's ($i = 1, \dots, d$) are i.i.d random variables from $\text{uniform}(0,1)$. Assume the error $\epsilon_n \sim 0.5N(0,1)$. Consider the mean reward functions

$$\begin{aligned} f_1(X_n) &= 0.5(X_{n1} + X_{n2} + X_{n3}), \\ f_2(X_n) &= 0.4(X_{n3} + X_{n4})^2 + 1.5\sin(X_{n1} + 0.25X_{n4}), \\ f_3(X_n) &= \frac{2X_{n3}}{0.5 + (1.5 + X_{n3} + X_{n4})}. \end{aligned}$$

We set the reduction dimensions for the three arms by $r_1 = 1$, $r_2 = 2$ and $r_3 = 2$. Given the time horizon $N = 1200$, the first 90 rounds of the game are the forced sampling period. Let the ‘‘inferior’’ arm sampling probability be $\pi_n = \frac{1}{(\log_2 n)^2}$, and the kernel bandwidth for arm i be $h = n^{-1/(2+r_i)}$, $i = 1, 2, 3$. Dimension reduction methods SIR, CIS-SIR as well as their combining strategy are run separately, and their per-round regret r_n is summarized in Figure 1 (right panel), which shows that the combining strategy performs the best. Since the second arm ($i = 2$) is played the most (for SIR, 1022 times; for CIS-SIR, 1026 times), we show the estimated dimension reduction matrix for the second arm at the last time point $n = N$ in Table 1. As expected, CIS-SIR results in a sparse dimension reduction matrix with rows 1, 3 and 4 being non-zero.

It is worth mentioning that in the simulation above, we assume the reduction dimensions for all arms are already known. In cases where the reduction dimensions are unknown, we may apply model selection procedures to choose them, which will be investigated in the future.

Table 1: Comparing the estimated dimension reduction matrix $\hat{B}_{2,N}^*$ for the second arm between SIR and CIS-SIR.

	SIR	CIS-SIR
1	-0.658	-0.599
2	0.011	-0.091
3	-0.469	0.601
4	-0.582	0.219
5	-0.001	0.075
6	0.071	0.232
7	0.013	-0.340
8	-0.019	0.087
9	-0.029	-0.194
10	0.016	0.030

7. Web-Based Personalized News Article Recommendation

In this section, we use the Yahoo! Front Page Today Module User Click Log data set (Yahoo! Academic Relations, 2011) to evaluate the proposed allocation strategy. The complete data set contains about 46 million web page visit interaction events collected during the first ten days in May 2009. Each of these events has four components: (1) five variables constructed from the Yahoo! front page visitor’s information; (2) a pool of about 10-14 editor-picked news articles; (3) one article actually displayed to the visitor (it is selected uniformly at random from the article pool); (4) the visitor’s response to the selected article (no click: 0, click: 1). Since different visitors may have different preferences for the same article, it is reasonable to believe that the displayed article should be selected based on the visitor associated variables. If we treat the articles in the pool as the bandit arms, and the visitor associated variables as the covariates, this data set provides the necessary platform to test a MABC algorithm.

One remaining issue before algorithm evaluation is that the complete data set is long-term in nature and the pool of articles is dynamic, i.e., some outdated articles are dropped out as people’s interest in these articles fades away, and some breaking-news articles can appear and be added to the pool. Our current problem setup, however, assumes stationary mean reward functions with a fixed set of arms. To avoid introducing biased evaluation results, we focus on short-term performance where people’s interest on a particular article does not change too much and the pool of articles remains stable. Therefore, we consider only one day’s data (May 1, 2009). Also, we choose four articles ($l = 4$) as the candidate bandit arms (article id 109511 - 109514), and keep only the events where the four articles are included in the article pool and one of the four articles is actually displayed to the visitor. A similar screening treatment of the data set is used in May et al. (2012) for MABC algorithm evaluation purposes. With the above, we obtain a reduced data set containing 452,189 interaction events for subsequent use.

Another challenge in evaluating a MABC algorithm comes from the intrinsic nature of bandit problem: for every visitor interaction event, only one article is displayed, and we only have this visitor’s response to the displayed article, while his/her response to other articles is not available, causing a difficulty if the actually displayed article does not match the article selected by a MABC algorithm. To overcome this issue caused by limited feedback, we apply the unbiased offline evaluation method proposed by Li et al. (2010). Briefly, for each encountered event, the MABC algorithm uses the previous “valid” data set (history) to estimate the mean reward functions and propose an arm to pull. If the proposed arm matches the actually displayed arm, this event is kept as a “valid” event, and the “valid” data set (history) is updated by adding this event. On the other hand, if the proposed arm does not match the displayed arm, this event is ignored, and the “valid” data set (history) is unchanged. This process is run sequentially over all the interaction events to generate the final “valid” data set, upon which a MABC algorithm can be evaluated by calculating the click-through rate (CTR, the proportion of times a click is made). Under the fact that in each interaction event, the displayed arm was selected uniformly at random from the pool, it can be argued that the final “valid” data set is like being obtained from running the MABC algorithm over a random sample of the underlying population.

With the reduced data set and the unbiased offline evaluation method, we evaluate the performance of the following algorithms.

random: an arm is selected uniformly at random.

ϵ -greedy: The randomized allocation strategy is run naively without consideration of covariates. A simple average is used to estimate the mean reward for each arm.

SIR-kernel: The randomized allocation strategy is run using SIR-kernel method to estimate the mean reward functions. Three sequences of bandwidth choices are considered: $h_{n1} = n^{-1/6}$, $h_{n2} = n^{-1/8}$ and $h_{n3} = n^{-1/10}$.

model combining: Model combining based randomized allocation strategy described in Section 3 is run with SIR-kernel method ($h_{n3} = n^{-1/10}$) and the naive simple average method (ϵ -greedy) as two candidate modeling methods.

The ϵ -greedy, SIR-kernel and model combining algorithms described above all take the first 1000 time points to be the forced sampling stage and use $\pi_n = n^{-1/4}/6$. Also, for any given arm, the SIR-kernel method limits the history time window for reward estimation to have maximum sample size of 10,000 (larger history sample size does not give us noticeable difference in performance). In addition, we consider the following parametric algorithm:

LinUCB: LinUCB employs Bayesian logistic regression to estimate the mean reward functions. The detailed implementation procedures are described in Algorithm 3 of Chapelle and Li (2011).

Each of the algorithms listed above is run 100 times over the reduced data set with the unbiased offline evaluation method. For each of the 100 runs, the algorithm starts at a position randomly chosen from the first 10,000 events of the reduced data set. The resulting CTRs are divided by the mean CTR of the random algorithm to give the normalized CTRs, and their boxplots are shown in Fig. 2.

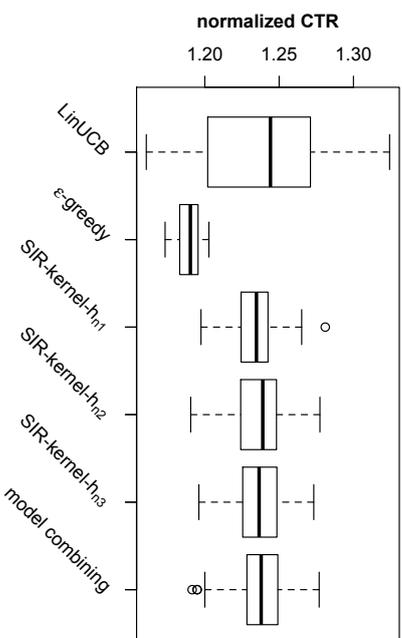


Figure 2: Boxplots of normalized CTRs of various algorithms on the news article recommendation data set. Algorithms include (from left to right): LinUCB, ϵ -greedy, SIR-kernel (h_{n1}), SIR-kernel (h_{n2}), SIR-kernel (h_{n3}), model combining with SIR-kernel (h_{n3}) and ϵ -greedy. CTRs are normalized with respect to the random algorithm.

It appears that the SIR-kernel methods with different candidate bandwidth sequences have very similar performance. The naive ϵ -greedy algorithm, however, clearly underperforms due to its failure to take advantage of the response-covariate association. When the naive simple average estimation (ϵ -greedy) is used together with SIR-kernel method (h_{n3}) in the model combining algorithm, the overall performance does not seem to deteriorate with the existence of this naive estimation method, showing once again that the model combining algorithm allows us to safely explore new modeling methods by automatically selecting the appropriate modeling candidate. Given that the covariates in the news article recommendation data set are constructed with logistic regression related methods (Li et al., 2010), it is satisfactory to observe that SIR-kernel algorithm can have similar performance with relatively small variation when compared with the LinUCB algorithm.

8. Conclusions

In this work, we study the kernel estimation based randomized allocation strategy in a flexible setting for MABC that works for both continuous and discrete response, and establish both the strong consistency and a finite-time regret upper bound. Allowing dependence of the covariate and the error, we rely on new technical tools to add kernel methods to the family of strongly consistent methods, which can potentially improve estimation efficiency for smooth reward functions. Although the finite-time regret upper bound is slightly sub-

optimal for the investigated randomized allocation strategy in the minimax sense (Perchet and Rigollet, 2013), the flexibility in estimation of the mean reward functions can be very useful in applications. In that regard, we integrate a model combination technique into the allocation strategy to share the strengths of different statistical learning methods for reward function estimation. It is shown that with the proposed data-driven combination of estimation methods, our allocation strategy can remain strongly consistent.

In Appendix A, we also show that by resorting to an alternative UCB-type criterion for arm comparison, the regret rate of the modified randomized allocation algorithm is improved to be minimax optimal up to a logarithmic factor. It remains to be seen if the UCB modification can be incorporated to construct a model combination algorithm with adaptive minimax rate. Moreover, as an important open question raised by a reviewer, it would be interesting to see whether the cumulative regret of the model combination strategy is comparable to that of the candidate model with the smallest regret in the sense of an oracle-type inequality similar to that of, e.g., Audibert (2009).

Acknowledgments

We would like to thank the Editor and two anonymous referees for their very constructive comments that help improving this manuscript significantly. This research was supported by the US NSF grant DMS-1106576

Appendix A. A Kernel Estimation Based UCB Algorithm

In this section, we modify the randomized allocation strategy to give a UCB-type algorithm that results in an improved rate of the cumulative regret. Similar to Section 4, we consider the Nadaraya-Watson estimation as the only modeling method, that is,

$$\hat{f}_{i,n}(x) = \frac{\sum_{j \in I_{i,n}} Y_{i,j} K\left(\frac{x-X_j}{h_{n-1}}\right)}{\sum_{j \in I_{i,n}} K\left(\frac{x-X_j}{h_{n-1}}\right)}. \quad (6)$$

We slightly revise step 5 of the proposed randomized allocation strategy:

STEP 5'. Estimate the best arm, select and pull. For the covariate X_n , define

$$\tilde{i}_n = \operatorname{argmax}_{1 \leq i \leq l} \hat{f}_{i,n}(X_n) + U_{i,n}(X_n),$$

where $U_{i,n}(x) = \sqrt{\frac{\tilde{c}(\log N) \sum_{j \in I_{i,n}} K^2\left(\frac{x-X_j}{h_{n-1}}\right)}{\sum_{j \in I_{i,n}} K\left(\frac{x-X_j}{h_{n-1}}\right)}}$ and \tilde{c} is some positive constant (if there is a tie, any tie-breaking rule may be applied). Choose an arm, with probability $1 - (l-1)\pi_n$ for arm \tilde{i}_n (the currently most promising choice) and with probability π_n for each of the remaining arms. That is,

$$I_n = \begin{cases} \tilde{i}_n, & \text{with probability } 1 - (l-1)\pi_n, \\ i, & \text{with probability } \pi_n, i \neq \tilde{i}_n, 1 \leq i \leq l. \end{cases}$$

Clearly, (6) shows a UCB-type algorithm that naturally extends from the UCB1 of Auer et al. (2002) and the UCBogram of Rigollet and Zeevi (2010). Indeed, given the uniform kernel $K(u) = I(\|u\|_\infty \leq 1)$, we have $U_{i,n}(x) = \tilde{c} \sqrt{\frac{\log N}{N_{i,n}(x)}}$, where $N_{i,n}(x)$ is the number of times arm i gets pulled inside the cube that centers at x with bin width $2h_{n-1}$. For presentation clarity, we assume $K(\cdot)$ is the uniform kernel, but the results can be generalized to kernel functions that satisfy Assumption 4. As shown in Theorem 4 below, the finite-time regret upper bound of the UCB-type algorithm achieves the minimax rate up to a logarithmic factor.

Theorem 4. *Suppose Assumptions 0-1 hold and the uniform kernel function is used. Then for the modified algorithm, if $n_0 = lN h^\kappa$, $\tilde{c} > \max\{2\sqrt{3}v, 12c\}$, $h = h_n = 1/\lceil(\frac{N}{\log N})^{\frac{1}{2\kappa+4}}\rceil$ and $\pi_n \leq \frac{1}{l} \wedge \frac{1}{\tilde{c}} \wedge \frac{1}{n} (\frac{\log N}{n})^{\frac{1}{2\kappa+4}}$, the mean of cumulative regret satisfies*

$$ER_N(\eta) < \tilde{C}^* N^{1-\frac{1}{2\kappa+4}} (\log N)^{\frac{1}{2\kappa+4}}. \quad (7)$$

It is worth noting that despite the seemingly minor algorithmic modification, the proof techniques used by Theorem 2 and Theorem 4 are quite different. The key difference is that: the UCB-type criterion enables us to provide upper bounds (with high probability) for the number of times the ‘‘inferior’’ arms are selected, and these bounds are dependent on the reward difference between the ‘‘optimal’’ and the ‘‘inferior’’ arms; for the algorithm before modification, we have to rely on studying the estimation errors of the reward functions and the UCB-type arguments do not apply. It is not settled yet as to whether the suboptimal rate of the ϵ -greedy type algorithm is intrinsic to the method or is the limitation of the proof techniques. But we tend to think that the rate given for the ϵ -greedy type algorithm is intrinsic to the method. Also, although the UCB-type algorithm leads to an improved regret rate, it is not yet clear how it could be used to construct a model combination algorithm.

Appendix B. Lemmas and Proofs

B.1 Proof of Theorem 1

Lemma 1. *Suppose $\{\mathcal{F}_j, j = 1, 2, \dots\}$ is an increasing filtration of σ -fields. For each $j \geq 1$, let ε_j be an \mathcal{F}_{j+1} -measurable random variable that satisfies $E(\varepsilon_j | \mathcal{F}_j) = 0$, and let T_j be an \mathcal{F}_j -measurable random variable that is upper bounded by a constant $C > 0$ in absolute value almost surely. If there exist positive constants v and c such that for all $k \geq 2$ and $j \geq 1$, $E(|\varepsilon_j|^k | \mathcal{F}_j) \leq kv^2 c^{k-2}/2$, then for every $\epsilon > 0$ and every integer $n \geq 1$,*

$$P\left(\sum_{j=1}^n T_j \varepsilon_j \geq n\epsilon\right) \leq \exp\left(-\frac{n\epsilon^2}{2C^2(v^2 + c\epsilon/C)}\right).$$

Proof of Lemma 1. Note that

$$\begin{aligned} P\left(\sum_{j=1}^n T_j \varepsilon_j \geq n\epsilon\right) &\leq e^{-tn\epsilon} E\left[\exp\left(t\sum_{j=1}^n T_j \varepsilon_j\right)\right] \\ &= e^{-tn\epsilon} E\left[E\left(\exp\left(t\sum_{j=1}^n T_j \varepsilon_j\right) \middle| \mathcal{F}_n\right)\right] \\ &= e^{-tn\epsilon} E\left[\exp\left(t\sum_{j=1}^{n-1} T_j \varepsilon_j\right) E(e^{tT_n \varepsilon_n} \middle| \mathcal{F}_n)\right]. \end{aligned}$$

By the moment condition on ε_n and Taylor expansion, we have

$$\begin{aligned} \log E(e^{tT_n \varepsilon_n} \middle| \mathcal{F}_n) &\leq E(e^{tT_n \varepsilon_n} \middle| \mathcal{F}_n) - 1 \\ &\leq tT_n E(\varepsilon_n \middle| \mathcal{F}_n) + \sum_{k=2}^{\infty} \frac{t^k |T_n|^k}{k!} E(|\varepsilon_n|^k \middle| \mathcal{F}_n) \\ &\leq \frac{v^2 C^2 t^2}{2} (1 + cCt + (cCt)^2 + \dots) \\ &= \frac{v^2 C^2 t^2}{2(1 - cCt)} \end{aligned}$$

for $t < 1/cC$. Thus, it follows by induction that

$$\begin{aligned} P\left(\sum_{j=1}^n T_j \varepsilon_j \geq n\epsilon\right) &\leq \exp\left(-tn\epsilon + \frac{nu^2 C^2 t^2}{2(1 - cCt)}\right) \\ &\leq \exp\left(-\frac{n\epsilon^2}{2C^2(v^2 + c\epsilon/C)}\right), \end{aligned}$$

where the last inequality is obtained by minimization over t . This completes the proof of Lemma 1. \square

Lemma 2. *Suppose $\{\mathcal{F}_j; j = 1, 2, \dots\}$ is an increasing filtration of σ -fields. For each $j \geq 1$, let W_j be an \mathcal{F}_j -measurable Bernoulli random variable whose conditional success probability satisfies*

$$P(W_j = 1 \middle| \mathcal{F}_{j-1}) \geq \beta_j$$

for some $0 \leq \beta_j \leq 1$. Then given $n \geq 1$,

$$P\left(\sum_{j=1}^n W_j \leq \sum_{j=1}^n \beta_j / 2\right) \leq \exp\left(-\frac{3\sum_{j=1}^n \beta_j}{28}\right).$$

Lemma 2 is known as an extended Bernstein inequality (see, e.g., Yang and Zhu (2002), Section A.4.). For completeness, we give a brief proof here.

Proof of Lemma 2. Suppose \tilde{W}_j , $1 \leq j \leq n$ are independent Bernoulli random variables with success probability β_j , and are assumed to be independent of \mathcal{F}_n . By Bernstein's inequality (e.g., Cesa-Bianchi and Lugosi, 2006, Corollary A.3),

$$P\left(\sum_{j=1}^n \tilde{W}_j \leq \left(\sum_{j=1}^n \beta_j\right) / 2\right) \leq \exp\left(-\frac{3\sum_{j=1}^n \beta_j}{28}\right).$$

Also, $\sum_{j=1}^n W_j$ is stochastically no smaller than $\sum_{j=1}^n \tilde{W}_j$, that is, for every t , $P(\sum_{j=1}^n W_j > t) \geq P(\sum_{j=1}^n \tilde{W}_j > t)$. Indeed, noting that $P(W_n > t \middle| \mathcal{F}_{n-1}) \geq P(\tilde{W}_n > t)$ for every t , we have

$$P(W_1 + \dots + W_n > t \middle| \mathcal{F}_{n-1}) \geq P(W_1 + \dots + W_{n-1} + \tilde{W}_n > t \middle| \mathcal{F}_{n-1}).$$

Similarly, by $P(W_{n-1} > t \middle| \mathcal{F}_{n-2}) \geq P(\tilde{W}_{n-1} > t)$ for every t and the independence of \tilde{W}_j 's,

$$P(W_1 + \dots + W_{n-1} + \tilde{W}_n > t \middle| \mathcal{F}_{n-2}, \tilde{W}_n) \geq P(W_1 + \dots + W_{n-2} + \tilde{W}_{n-1} + \tilde{W}_n > t \middle| \mathcal{F}_{n-2}, \tilde{W}_n).$$

Continuing the process above, we can see that $P(\sum_{j=1}^n W_j > t) \geq P(\sum_{j=1}^n \tilde{W}_j > t)$ holds. \square

Lemma 3. *Under the settings of the kernel estimation in Section 4.1, given arm i and a cube $A \subset [0, 1]^d$ with side width h , if Assumptions 0, 3 and 4 are satisfied, then for any $\epsilon > 0$,*

$$\begin{aligned} &P\left(\sup_{x \in A} \sum_{j \in J_{i,n+1}} \varepsilon_j K\left(\frac{x - X_j}{h_n}\right) > \frac{n\epsilon}{1 - 1/\sqrt{2}}\right) \\ &\leq \exp\left(-\frac{n\epsilon^2}{4c_2^2 v^2}\right) + \exp\left(-\frac{n\epsilon}{4c_4 c}\right) + \sum_{k=1}^{\infty} 2^{kd} \exp\left(-\frac{2^k n\epsilon^2}{\lambda^2 v^2}\right) + \sum_{k=1}^{\infty} 2^{kd} \exp\left(-\frac{2^{k/2} n\epsilon}{2\lambda c}\right). \end{aligned}$$

Proof of Lemma 3. At each time point j , let $W_j = 1$ if arm i is pulled (i.e., $I_j = i$), and $W_j = 0$ otherwise. Denote $G(x) = \sum_{j=1}^n \varepsilon_j W_j K\left(\frac{x - X_j}{h_n}\right)$. Then, to find an upper bound for $P(\sup_{x \in A} G(x) > n\epsilon/(1 - 1/\sqrt{2}))$, we use a ‘‘chaining’’ argument. For each $k \geq 0$, let $\gamma_k = h_n/2^k$, and we can partition the cube A into 2^{kd} bins with bin width γ_k . Let F_k denote the set consisting of the center points of these 2^{kd} bins. Clearly, $\text{card}(F_k) = 2^{kd}$, and F_k is a $\gamma_k/2$ -net of A in the sense that for every $x \in A$, we can find a $x' \in F_k$ such that $\|x - x'\|_{\infty} \leq \gamma_k/2$. Let $\tau_k(x) = \text{argmin}_{x' \in F_k} \|x - x'\|_{\infty}$ be the closest point to x in the net F_k . With the sequence F_0, F_1, F_2, \dots of $\gamma_0/2, \gamma_1/2, \gamma_2/2, \dots$ nets in A , it is easy to see that for every $x \in A$, $\|\tau_k(x) - \tau_{k-1}(x)\|_{\infty} \leq \gamma_k/2$ and $\lim_{k \rightarrow \infty} \tau_k(x) = x$. Thus, by the continuity of the kernel function, we have $\lim_{k \rightarrow \infty} G(\tau_k(x)) = G(x)$. It follows that

$$G(x) = G(\tau_0(x)) + \sum_{k=1}^{\infty} [G(\tau_k(x)) - G(\tau_{k-1}(x))].$$

Thus,

$$\begin{aligned}
& P\left(\sup_{x \in A} G(x) > \frac{n\epsilon}{1-1/\sqrt{2}}\right) \\
&= P\left(\sup_{x \in A} \{G(\tau_0(x)) + \sum_{k=1}^{\infty} [G(\tau_k(x)) - G(\tau_{k-1}(x))]\} > \sum_{k=0}^{\infty} \frac{n\epsilon}{2^{k/2}}\right) \\
&\leq P\left(\sup_{x \in A} G(\tau_0(x)) > n\epsilon\right) + \sum_{k=1}^{\infty} P\left(\sup_{x \in A} [G(\tau_k(x)) - G(\tau_{k-1}(x))] > \frac{n\epsilon}{2^{k/2}}\right) \\
&\leq P\left(\sup_{x \in F_0} G(x) > n\epsilon\right) + \sum_{k=1}^{\infty} P\left(\sup_{\substack{x_2 \in F_k, x_1 \in F_{k-1} \\ \|x_2 - x_1\|_{\infty} \leq \gamma_k/2}} [G(x_2) - G(x_1)] > \frac{n\epsilon}{2^{k/2}}\right) \\
&\leq \text{card}(F_0) \max_{x \in F_0} P(G(x) > n\epsilon) \\
&\quad + \sum_{k=1}^{\infty} 2^d \text{card}(F_{k-1}) \max_{\substack{x_2 \in F_k, x_1 \in F_{k-1} \\ \|x_2 - x_1\|_{\infty} \leq \gamma_k/2}} P\left(G(x_2) - G(x_1) > \frac{n\epsilon}{2^{k/2}}\right), \tag{8}
\end{aligned}$$

where the last inequality holds because for each $x_1 \in F_{k-1}$, there are only 2^d such points $x_2 \in F_k$ that can satisfy $\|x_2 - x_1\|_{\infty} \leq \gamma_k/2$. Given $x \in F_0$, since $|W_j K(\frac{x-X_j}{h})| \leq c_4$ almost surely for all $j \geq 1$, it follows by Lemma 1 that

$$P(G(x) > n\epsilon) \leq \exp\left(-\frac{n\epsilon^2}{2c_4^2(v^2 + \alpha\epsilon/c_4)}\right). \tag{9}$$

Similarly, given $x_2 \in F_k$, $x_1 \in F_{k-1}$ and $\|x_2 - x_1\|_{\infty} \leq \gamma_k$, since

$$\left|K\left(\frac{x_2 - X_j}{h_n}\right) - K\left(\frac{x_1 - X_j}{h_n}\right)\right| \leq \frac{\lambda \|x_2 - x_1\|_{\infty}}{2h_n} \leq \frac{\lambda \gamma_k}{2h_n} = \frac{\lambda}{2^{k+1}}$$

almost surely, it follows by Lemma 1 that

$$\begin{aligned}
P\left(G(x_2) - G(x_1) > \frac{n\epsilon}{2^{k/2}}\right) &= P\left(\sum_{j=1}^n \epsilon_j W_j \left[K\left(\frac{x_2 - X_j}{h}\right) - K\left(\frac{x_1 - X_j}{h}\right)\right] > \frac{n\epsilon}{2^{k/2}}\right) \\
&\leq \exp\left(-\frac{2^{k+2}n\epsilon^2}{2\lambda^2(v^2 + 2^{k/2+1}\alpha\epsilon/\lambda)}\right). \tag{10}
\end{aligned}$$

Thus, by (8), (9) and (10),

$$\begin{aligned}
& P\left(\sup_{x \in A} G(x) > \frac{n\epsilon}{1-1/\sqrt{2}}\right) \\
&\leq \exp\left(-\frac{n\epsilon^2}{2c_4^2(v^2 + \alpha\epsilon/c_4)}\right) + \sum_{k=1}^{\infty} 2^{kd} \exp\left(-\frac{2^{k+2}n\epsilon^2}{2\lambda^2(v^2 + 2^{k/2+1}\alpha\epsilon/\lambda)}\right) \\
&\leq \exp\left(-\frac{n\epsilon^2}{4c_4^2v^2}\right) + \exp\left(-\frac{n\epsilon}{4c_4\alpha}\right) + \sum_{k=1}^{\infty} 2^{kd} \exp\left(-\frac{2^{k/2}n\epsilon^2}{\lambda^2v^2}\right) + \sum_{k=1}^{\infty} 2^{kd} \exp\left(-\frac{2^{k/2}n\epsilon}{2\lambda\alpha}\right).
\end{aligned}$$

This completes the proof of Lemma 3. \square

Proof of Theorem 1. Recall that $M_{i,n} = |J_{i,n}|$, \underline{c} is the covariate density lower bound, and L, L_1, c_3 are constants defined in Assumption 4 for the kernel function $K(\cdot)$, and. Note that for each $x \in R^d$,

$$\begin{aligned}
& |\hat{f}_{i,n+1}(x) - f_i(x)| = \left| \frac{\sum_{j \in J_{i,n+1}} Y_{ij} K\left(\frac{x - X_j}{h_n}\right)}{\sum_{j \in J_{i,n+1}} K\left(\frac{x - X_j}{h_n}\right)} - f_i(x) \right| \\
&= \left| \frac{\sum_{j \in J_{i,n+1}} (f_i(X_j) + \epsilon_j) K\left(\frac{x - X_j}{h_n}\right)}{\sum_{j \in J_{i,n+1}} K\left(\frac{x - X_j}{h_n}\right)} - f_i(x) \right| \\
&= \left| \frac{\sum_{j \in J_{i,n+1}} (f_i(X_j) - f_i(x)) K\left(\frac{x - X_j}{h_n}\right)}{\sum_{j \in J_{i,n+1}} K\left(\frac{x - X_j}{h_n}\right)} + \frac{\sum_{j \in J_{i,n+1}} \epsilon_j K\left(\frac{x - X_j}{h_n}\right)}{\sum_{j \in J_{i,n+1}} K\left(\frac{x - X_j}{h_n}\right)} \right| \\
&\leq \sup_{\{x,y\}: \|x-y\|_{\infty} \leq Lh_n} |f_i(x) - f_i(y)| + \left| \frac{1}{M_{i,n+1}h_n^d} \frac{\sum_{j \in J_{i,n+1}} \epsilon_j K\left(\frac{x - X_j}{h_n}\right)}{\sum_{j \in J_{i,n+1}} K\left(\frac{x - X_j}{h_n}\right)} \right|, \tag{11}
\end{aligned}$$

where the last inequality follows from the bounded support assumption of kernel function $K(\cdot)$. By uniform continuity of the function f_i ,

$$\lim_{n \rightarrow \infty} \sup_{\{x,y\}: \|x-y\|_{\infty} \leq Lh_n} |f_i(x) - f_i(y)| = 0.$$

To show that $\|\hat{f}_{i,n} - f_i\|_{\infty} \rightarrow 0$ as $n \rightarrow \infty$, we only need

$$\sup_{x \in [0,1]^d} \left| \frac{1}{M_{i,n+1}h_n^d} \frac{\sum_{j \in J_{i,n+1}} \epsilon_j K\left(\frac{x - X_j}{h_n}\right)}{\sum_{j \in J_{i,n+1}} K\left(\frac{x - X_j}{h_n}\right)} \right| \rightarrow 0 \text{ as } n \rightarrow \infty. \tag{12}$$

First, we want to show

$$\inf_{x \in [0,1]^d} \frac{1}{M_{i,n+1}h_n^d} \sum_{j \in J_{i,n+1}} K\left(\frac{x - X_j}{h_n}\right) > \frac{c_3 \epsilon L_1^d n}{2}, \tag{13}$$

almost surely for large enough n . Indeed, for each $n \geq n_0l + 1$, we can partition the unit cube $[0,1]^d$ into \tilde{B} bins with bin width L_1h_n , such that $\tilde{B} \leq 1/(L_1h_n)^d$. We denote these

bins by $\tilde{A}_1, \tilde{A}_2, \dots, \tilde{A}_{\tilde{B}}$. Given an arm i and $1 \leq k \leq \tilde{B}$, for every $x \in \tilde{A}_k$, we have

$$\begin{aligned} \sum_{j \in I_{i,n+1}} K\left(\frac{x - X_j}{h_n}\right) &= \sum_{j=1}^n I(I_j = i) K\left(\frac{x - X_j}{h_n}\right) \\ &\geq \sum_{j=1}^n I(I_j = i, X_j \in \tilde{A}_k) K\left(\frac{x - X_j}{h_n}\right) \\ &\geq c_3 \sum_{j=1}^n I(I_j = i, X_j \in \tilde{A}_k), \end{aligned}$$

where the last inequality follows by Assumption 4. Consequently,

$$\begin{aligned} P\left(\inf_{x \in \tilde{A}_k} \frac{1}{M_{i,n+1} h_n^d} \sum_{j \in I_{i,n+1}} K\left(\frac{x - X_j}{h_n}\right) \leq \frac{c_3 \mathcal{E} L_1^d \pi_n}{2}\right) \\ \leq P\left(\inf_{x \in \tilde{A}_k} \frac{1}{n h_n^d} \sum_{j \in I_{i,n+1}} K\left(\frac{x - X_j}{h_n}\right) \leq \frac{c_3 \mathcal{E} L_1^d \pi_n}{2}\right) \\ \leq P\left(\frac{c_3}{n h_n^d} \sum_{j=1}^n I(I_j = i, X_j \in \tilde{A}_k) \leq \frac{c_3 \mathcal{E} L_1^d \pi_n}{2}\right) \\ = P\left(\sum_{j=1}^n I(I_j = i, X_j \in \tilde{A}_k) \leq \frac{\mathcal{E} n (L_1 h_n)^d \pi_n}{2}\right). \end{aligned} \quad (14)$$

Noting that $P(I_j = i, X_j \in \tilde{A}_k | \mathcal{Z}^j) \geq \mathcal{E}(L_1 h_n)^d \pi_j$ for $1 \leq j \leq n$, we have by Lemma 2 that

$$P\left(\sum_{j=1}^n I(I_j = i, X_j \in \tilde{A}_k) \leq \frac{\mathcal{E} n (L_1 h_n)^d \pi_n}{2}\right) \leq \exp\left(-\frac{3\mathcal{E} n (L_1 h_n)^d \pi_n}{28}\right). \quad (15)$$

Therefore,

$$\begin{aligned} P\left(\inf_{x \in [0,1]^d} \frac{1}{M_{i,n+1} h_n^d} \sum_{j \in I_{i,n+1}} K\left(\frac{x - X_j}{h_n}\right) \leq \frac{c_3 \mathcal{E} L_1^d \pi_n}{2}\right) \\ \leq \sum_{k=1}^{\tilde{B}} P\left(\inf_{x \in \tilde{A}_k} \frac{1}{M_{i,n+1} h_n^d} \sum_{j \in I_{i,n+1}} K\left(\frac{x - X_j}{h_n}\right) \leq \frac{c_3 \mathcal{E} L_1^d \pi_n}{2}\right) \\ \leq \tilde{B} \exp\left(-\frac{3\mathcal{E} n (L_1 h_n)^d \pi_n}{28}\right), \end{aligned}$$

where the last inequality follows by (14) and (15). With the condition $n h_n^{2d} \pi_n^4 / \log n \rightarrow \infty$, we immediately obtain (13) by Borel-Cantelli lemma.

By (13), it follows that (12) holds if

$$\sup_{x \in [0,1]^d} \left| \frac{1}{M_{i,n+1} h_n^d} \sum_{j \in I_{i,n+1}} \varepsilon_j K\left(\frac{x - X_j}{h_n}\right) \right| = o(\pi_n). \quad (16)$$

In the rest of the proof, we want to show that (16) holds. For each $n \geq n_0 l + 1$, we can partition the unit cube $[0,1]^d$ into B bins with bin length h_n such that $B \leq 1/h_n^d$. At each time point j , let $W_j = 1$ if arm i is pulled (i.e., $I_j = i$), and $W_j = 0$ otherwise. Then given $\varepsilon > 0$,

$$\begin{aligned} P\left(\sup_{x \in [0,1]^d} \left| \frac{1}{M_{i,n+1} h_n^d} \sum_{j \in I_{i,n+1}} \varepsilon_j K\left(\frac{x - X_j}{h_n}\right) \right| > \pi_n \varepsilon\right) \\ \leq B \max_{1 \leq k \leq B} P\left(\sup_{x \in \tilde{A}_k} \left| \frac{1}{M_{i,n+1} h_n^d} \sum_{j \in I_{i,n+1}} \varepsilon_j K\left(\frac{x - X_j}{h_n}\right) \right| > \pi_n \varepsilon\right) \\ \leq B P\left(\frac{M_{i,n+1}}{n} \leq \frac{\pi_n}{2}\right) \\ + B \max_{1 \leq k \leq B} P\left(\sup_{x \in \tilde{A}_k} \left| \frac{1}{M_{i,n+1} h_n^d} \sum_{j \in I_{i,n+1}} \varepsilon_j K\left(\frac{x - X_j}{h_n}\right) \right| > \pi_n \varepsilon, \frac{M_{i,n+1}}{n} > \frac{\pi_n}{2}\right) \\ \leq B P\left(\frac{M_{i,n+1}}{n} \leq \frac{\pi_n}{2}\right) + B \max_{1 \leq k \leq B} P\left(\sup_{x \in \tilde{A}_k} \left| \sum_{j \in I_{i,n+1}} \varepsilon_j K\left(\frac{x - X_j}{h_n}\right) \right| > \frac{n \pi_n^2 h_n^d \varepsilon}{2}\right) \\ \leq B P\left(\frac{M_{i,n+1}}{n} \leq \frac{\pi_n}{2}\right) + B \max_{1 \leq k \leq B} P\left(\sup_{x \in \tilde{A}_k} \left| \sum_{j \in I_{i,n+1}} \varepsilon_j K\left(\frac{x - X_j}{h_n}\right) \right| > \frac{n \pi_n^2 h_n^d \varepsilon}{2}\right), \end{aligned} \quad (17)$$

where the last inequality follows by Lemma 2. Note that by Lemma 3,

$$\begin{aligned} P\left(\sup_{x \in \tilde{A}_k} \left| \sum_{j \in I_{i,n+1}} \varepsilon_j K\left(\frac{x - X_j}{h_n}\right) \right| > \frac{n \pi_n^2 h_n^d \varepsilon}{2}\right) \\ \leq P\left(\sup_{x \in \tilde{A}_k} \sum_{j \in I_{i,n+1}} \varepsilon_j K\left(\frac{x - X_j}{h_n}\right) > \frac{n \pi_n^2 h_n^d \varepsilon}{2}\right) \\ + P\left(\sup_{x \in \tilde{A}_k} \sum_{j \in I_{i,n+1}} (-\varepsilon_j) K\left(\frac{x - X_j}{h_n}\right) > \frac{n \pi_n^2 h_n^d \varepsilon}{2}\right) \\ \leq 2 \exp\left(-\frac{(\sqrt{2}-1)^2 n \pi_n^4 h_n^{2d} \varepsilon^2}{32 c_4^2 v^2}\right) + 2 \exp\left(-\frac{(\sqrt{2}-1) n \pi_n^2 h_n^d \varepsilon}{8 \sqrt{2} c_4 v}\right) \\ + 2 \sum_{k=1}^{\infty} 2^{kd} \exp\left(-\frac{(\sqrt{2}-1)^{2k} n \pi_n^4 h_n^{2d} \varepsilon^2}{8 \lambda^2 v^2}\right) + 2 \sum_{k=1}^{\infty} 2^{kd} \exp\left(-\frac{(\sqrt{2}-1) 2^{k/2} n \pi_n^2 h_n^d \varepsilon}{4 \sqrt{2} \lambda v}\right). \end{aligned} \quad (18)$$

Thus, by (17), (18) and the condition that $n h_n^{2d} \pi_n^4 / \log n \rightarrow \infty$, (16) is an immediate consequence of Borel-Cantelli lemma. This completes the proof of Theorem 1. \square

B.2 Proofs of Theorem 2 and Corollary 2

Given $x \in [0,1]^d$, $1 \leq i \leq l$ and $n \geq n_0 l + 1$, define $G_{n+1}(x) = \{j : 1 \leq j \leq n, \|x - X_j\|_{\infty} \leq L h_n\}$ and $G_{i,n+1}(x) = \{j : 1 \leq j \leq n, I_j = i, \|x - X_j\|_{\infty} \leq L h_n\}$. Let $M_{n+1}(x)$ and $M_{i,n+1}(x)$ be the size of the sets $G_{n+1}(x)$ and $G_{i,n+1}(x)$, respectively. Then, the kernel method estimator $\hat{f}_{i,n+1}(x)$ satisfies the following lemma.

Lemma 4. *Suppose Assumptions 0, 1 and 4 are satisfied, and $\{\pi_n\}$ is a decreasing sequence. Given $x \in [0, 1]^d$, $1 \leq i \leq l$ and $n \geq n_{0l} + 1$, for every $\epsilon > \omega(Lh_n; f_i)$,*

$$\begin{aligned} P_{X^n}(|\hat{f}_{i,n+1}(x) - f_i(x)| \geq \epsilon) &\leq \exp\left(-\frac{3M_{n+1}(x)\pi_n}{28}\right) \\ &\quad + 4N \exp\left(-\frac{c_5^2 M_{n+1}(x)\pi_n(\epsilon - \omega(Lh_n; f_i))^2}{4c_4^2 v^2} + 4c_4 c(\epsilon - \omega(Lh_n; f_i))\right), \end{aligned} \quad (19)$$

where $P_{X^n}(\cdot)$ denotes the conditional probability given design points $X^n = (X_1, X_2, \dots, X_n)$.

Proof of Lemma 4. It is clear that if $M_{n+1}(x) = 0$, (19) trivially holds. Without loss of generality, assume $M_{n+1}(x) > 0$. Define the event $B_{i,n} = \left\{\frac{1}{M_{i,n+1}(x)} \sum_{j \in I_{i,n+1}} K\left(\frac{x - X_j}{h_n}\right) \geq c_5\right\}$. Note that

$$\begin{aligned} &P_{X^n}(|\hat{f}_{i,n+1}(x) - f_i(x)| \geq \epsilon) \\ &\leq P_{X^n}\left(\frac{M_{i,n+1}(x)}{M_{n+1}(x)} \leq \frac{\pi_n}{2}\right) + P_{X^n}\left(|\hat{f}_{i,n+1}(x) - f_i(x)| \geq \epsilon, \frac{M_{i,n+1}(x)}{M_{n+1}(x)} > \frac{\pi_n}{2}\right) \\ &\leq \exp\left(-\frac{3M_{n+1}(x)\pi_n}{28}\right) + P_{X^n}\left(|\hat{f}_{i,n+1}(x) - f_i(x)| \geq \epsilon, \frac{M_{i,n+1}(x)}{M_{n+1}(x)} > \frac{\pi_n}{2}, B_{i,n}^c\right) \\ &\quad + P_{X^n}\left(|\hat{f}_{i,n+1}(x) - f_i(x)| \geq \epsilon, \frac{M_{i,n+1}(x)}{M_{n+1}(x)} > \frac{\pi_n}{2}, B_{i,n}^c\right), \\ &=: \exp\left(-\frac{3M_{n+1}(x)\pi_n}{28}\right) + A_1 + A_2, \end{aligned} \quad (20)$$

where the last inequality follows by Lemma 2. Under $B_{i,n}$, by Assumption 4, the definition of the modulus continuity and the same argument as (11), we have

$$\begin{aligned} |\hat{f}_{i,n+1}(x) - f_i(x)| &= \left| \frac{\sum_{j \in I_{i,n+1}} Y_{i,j} K\left(\frac{x - X_j}{h_n}\right)}{\sum_{j \in I_{i,n+1}} K\left(\frac{x - X_j}{h_n}\right)} \right| \\ &\leq \omega(Lh_n; f_i) + \frac{1}{c_5 M_{i,n+1}(x)} \left| \sum_{j \in G_{i,n+1}(x)} \epsilon_j K\left(\frac{x - X_j}{h_n}\right) \right|. \end{aligned}$$

Define $\tilde{\sigma}_t = \inf\{\tilde{n} : \sum_{j=1}^{\tilde{n}} I(I_j = i \text{ and } \|x - X_j\|_\infty \leq Lh_n) \geq t\}$, $t \geq 1$. Then, by the previous display, for every $\epsilon > \omega(Lh_n; f_i)$,

$$\begin{aligned} A_1 &\leq P_{X^n}\left(\left|\sum_{j \in G_{i,n+1}(x)} \epsilon_j K\left(\frac{x - X_j}{h_n}\right)\right| \geq c_5 M_{i,n+1}(x)(\epsilon - \omega(Lh_n; f_i)), \frac{M_{i,n+1}(x)}{M_{n+1}(x)} > \frac{\pi_n}{2}\right) \\ &\leq \sum_{\tilde{n}=0}^n P_{X^n}\left(\left|\sum_{t=1}^{\tilde{n}} \epsilon_t K\left(\frac{x - X_{\tilde{\sigma}_t}}{h_n}\right)\right| \geq c_5 \tilde{n}(\epsilon - \omega(Lh_n; f_i)), \frac{M_{i,n+1}(x)}{M_{n+1}(x)} > \frac{\pi_n}{2}, M_{i,n+1}(x) = \tilde{n}\right) \\ &\leq \sum_{\tilde{n}=\lfloor M_{n+1}(x)\pi_n/2 \rfloor}^n P_{X^n}\left(\left|\sum_{t=1}^{\tilde{n}} \epsilon_t K\left(\frac{x - X_{\tilde{\sigma}_t}}{h_n}\right)\right| \geq c_5 \tilde{n}(\epsilon - \omega(Lh_n; f_i))\right) \\ &\leq \sum_{\tilde{n}=\lfloor M_{n+1}(x)\pi_n/2 \rfloor}^n 2 \exp\left(-\frac{\tilde{n}c_5^2(\epsilon - \omega(Lh_n; f_i))^2}{2c_4^2 v^2} + 2c_4 c(\epsilon - \omega(Lh_n; f_i))\right) \\ &\leq 2N \exp\left(-\frac{c_5^2 M_{n+1}(x)\pi_n(\epsilon - \omega(Lh_n; f_i))^2}{4c_4^2 v^2} + 4c_4 c(\epsilon - \omega(Lh_n; f_i))\right), \end{aligned} \quad (21)$$

where the last to second inequality follows by Lemma 1 and the upper boundedness of the kernel function. By an argument similar to the previous two displays (using the uniform kernel), it is not hard to obtain that

$$A_2 \leq 2N \exp\left(-\frac{M_{n+1}(x)\pi_n(\epsilon - \omega(Lh_n; f_i))^2}{4v^2} + 4c(\epsilon - \omega(Lh_n; f_i))\right). \quad (22)$$

Combining (20), (21), (22) and the fact that $0 < c_5 \leq 1 \leq c_4$, we complete the proof of Lemma 4. \square

Proof of Theorem 2. Since $\hat{f}_{i^*(X_n),n}(X_n) \leq \hat{f}_{i_n,n}(X_n)$, the regret accumulated after the initial forced sampling period satisfies that

$$\begin{aligned} &\sum_{n=n_{0l}+1}^N (f^*(X_n) - f_{i_n}(X_n)) \\ &= \sum_{n=n_{0l}+1}^N (f_{i^*(X_n),n}(X_n) - \hat{f}_{i^*(X_n),n}(X_n) + \hat{f}_{i^*(X_n),n}(X_n) - f_{i_n}(X_n) + f_{i_n}(X_n) - f_{i_n}(X_n)) \\ &\leq \sum_{n=n_{0l}+1}^N (f_{i^*(X_n),n}(X_n) - \hat{f}_{i^*(X_n),n}(X_n) + \hat{f}_{i_n,n}(X_n) - f_{i_n}(X_n) + f_{i_n}(X_n) - f_{i_n}(X_n)) \\ &\leq \sum_{n=n_{0l}+1}^N \left(2 \sup_{1 \leq l \leq l} |f_{i_n}(X_n) - f_l(X_n)| + AI(I_n \neq \hat{i}_n)\right). \end{aligned} \quad (23)$$

It can be seen from (23) that the error upper bound consists of the estimation error regret and randomization error regret.

First, we find the upper bound of the estimation error regret. Given arm i , $n \geq n_{0l}$ and $\epsilon > \omega(Lh_n; f_i)$,

$$\begin{aligned} & P(|\hat{f}_{i,n+1}(X_{n+1}) - f_i(X_{n+1})| \geq \epsilon) \\ & \leq EP_{X_{n+1}}(M_{n+1}(X_{n+1}) \leq \frac{\epsilon n(2Lh_n)^d}{2}) \\ & \quad + EP_{X_{n+1}}(|\hat{f}_{i,n+1}(X_{n+1}) - f_i(X_{n+1})| \geq \epsilon, M_{n+1}(X_{n+1}) > \frac{\epsilon n(2Lh_n)^d}{2}). \end{aligned} \quad (24)$$

Since for every $x \in [0, 1]^d$, $P(\|x - X_j\|_\infty \leq Lh_n) \geq c_l(2Lh_n)^d$, $1 \leq j \leq n$, we have by the extended Bernstein's inequality that

$$P_{X_{n+1}}(M_{n+1}(X_{n+1}) \leq \frac{\epsilon n(2Lh_n)^d}{2}) \leq \exp\left(-\frac{3c_l \epsilon n(2Lh_n)^d}{28}\right). \quad (25)$$

By Lemma 4,

$$\begin{aligned} & P_{X_{n+1}}(|\hat{f}_{i,n+1}(X_{n+1}) - f_i(X_{n+1})| \geq \epsilon, M_{n+1}(X_{n+1}) > \frac{\epsilon n(2Lh_n)^d}{2}) \\ & \leq \exp\left(-\frac{3c_l \epsilon n(2Lh_n)^d \pi_n}{56}\right) + 4N \exp\left(-\frac{c_{\hat{f}_i}^2 \epsilon n(2Lh_n)^d \pi_n (\epsilon - \omega(Lh_n; f_i))^2}{8c_{\hat{f}_i}^2 \gamma^2 + 8c_{\hat{f}_i} c(\epsilon - \omega(Lh_n; f_i))}\right). \end{aligned} \quad (26)$$

Let

$$\tilde{\epsilon}_{i,n} = \omega(Lh_n; f_i) + \sqrt{\frac{16c_{\hat{f}_i}^2 \gamma^2 \log(8LN^2/\delta)}{c_{\hat{f}_i}^2 c(2L)^d n h_n^d \pi_n}}.$$

Then, by (24), (25) and the definition of n_δ in (3), it follows that for every $n \geq n_\delta$,

$$P(|\hat{f}_{i,n+1}(X_{n+1}) - f_i(X_{n+1})| \geq \tilde{\epsilon}_{i,n}) \leq \frac{\delta}{4N} + \frac{\delta}{4N} + \frac{\delta}{2N} = \frac{\delta}{N},$$

which implies that

$$P\left(\sum_{n=n_\delta+1}^N 2 \sup_{1 \leq i \leq l} |\hat{f}_{i,n}(X_n) - f_i(X_n)| \geq \sum_{n=n_\delta+1}^N 2 \max_{1 \leq i \leq l} \tilde{\epsilon}_{i,n-1}\right) \leq \delta. \quad (27)$$

Next, we want to bound the randomization error regret. Given $\epsilon > 0$, since $P(I_n \neq \hat{i}_n) = (l-1)\pi_n$, we have by Hoeffding's inequality that

$$P\left(A \left(\sum_{n=n_\delta+1}^N I(I_n \neq \hat{i}_n) - \sum_{n=n_\delta+1}^N (l-1)\pi_n\right) \geq \epsilon\right) \leq \exp\left(-\frac{2\epsilon^2}{N A^2}\right).$$

Taking $\epsilon = A\sqrt{N/2} \log(1/\delta)$, we immediately get

$$P\left(A \sum_{n=n_\delta+1}^N I(I_n \neq \hat{i}_n) \geq A \sum_{n=n_\delta+1}^N (l-1)\pi_n + A\sqrt{\frac{N}{2}} \log\left(\frac{1}{\delta}\right)\right) \leq \delta. \quad (28)$$

Then, (23), (27) and (28) together complete the proof of Theorem 2. \square

B.3 Proof of Theorem 3

Lemma 5. Under Assumption E and the proposed allocation strategy, for each arm i

$$N_t < \infty \quad \text{a.s. for all } t \geq 1.$$

Proof of Lemma 5. It suffices to check that

$$\sum_{j=n_{0l}+1}^{\infty} I(I_j = i) = \infty \quad \text{a.s.} \quad (29)$$

Indeed, let \mathcal{F}_n , $n \geq 1$ be the σ -field generated by (Z^n, X_n, I_n) . By the proposed allocation strategy, for all $j \geq n_{0l} + 1$,

$$P(I_j = i | \mathcal{F}_{j-1}) \geq \pi_j.$$

By Assumption E, $\sum_{j=n_{0l}+1}^{\infty} P(I_j = i | \mathcal{F}_{j-1}) = \infty$. Therefore, (29) is an immediate result of the Levy's extension of the Borel-Cantelli lemmas (Williams, 1991, pp.124). \square

Proof of Theorem 3. The key to the proof is to show $\|\hat{f}_{i,n} - f_i\|_\infty \rightarrow 0$ almost surely for $1 \leq i \leq l$ (Yang and Zhu, 2002; Theorem 1). Without loss of generality, assume Δ includes only two candidate procedures ($m = 2$). Given $1 \leq i \leq l$, assume that procedure $\delta_1 \in \Delta_1$ and procedure $\delta_2 \in \Delta_2$ (the case of $\delta_1, \delta_2 \in \Delta_1$ is trivial). Since

$$\begin{aligned} \|\hat{f}_{i,n} - f_i\|_\infty &= \|W_{i,n,1}(\hat{f}_{i,n,1} - f_i) + W_{i,n,2}(\hat{f}_{i,n,2} - f_i)\|_\infty \\ &\leq W_{i,n,1} \|\hat{f}_{i,n,1} - f_i\|_\infty + W_{i,n,2} \|\hat{f}_{i,n,2} - f_i\|_\infty, \end{aligned}$$

it suffices to prove that $\frac{W_{i,n,1}}{W_{i,n,2}} \rightarrow \infty$ almost surely as $n \rightarrow \infty$.

As defined before, $N_t = \inf\{n : \sum_{j=n_{0l}+1}^n I(I_j = i) \geq t\}$, and let \mathcal{F}_n be the σ -field generated by (Z^n, X_n, I_n) . Then for any $t \geq 1$, N_t is a stopping time relative to $\{\mathcal{F}_n, n \geq 1\}$. By Lemma 5, $N_t < \infty$ a.s. for all $t \geq 1$. Therefore, the weights $W_{i,N_t,1}$, $W_{i,N_t,2}$ and the variance estimates $\hat{v}_{i,N_t,1}$, $\hat{v}_{i,N_t,2}$ and \hat{v}_{i,N_t} for $t \geq 1$ are well-defined. By the allocation strategy, the weight associated with arm i is updated only after this arm is pulled. Consequently, we only need to show $\frac{W_{i,N_t,1}}{W_{i,N_t,2}} \rightarrow \infty$ almost surely as $t \rightarrow \infty$.

Note that for any $t \geq 1$,

$$\begin{aligned}
\frac{W_{i,N_{t+1,1}}}{W_{i,N_{t+1,2}}} &= \frac{W_{i,N_t,1}}{W_{i,N_t,2}} \times \frac{\hat{v}_{i,N_t,2}^{1/2}}{\hat{v}_{i,N_t,1}^{1/2}} \exp\left(-\frac{(\hat{f}_{i,N_t,1}(X_{N_t}) - Y_{i,N_t})^2 - (\hat{f}_{i,N_t,2}(X_{N_t}) - Y_{i,N_t})^2}{2\hat{v}_{i,N_t}}\right) \\
&= \frac{W_{i,N_t,1}}{W_{i,N_t,2}} \times \frac{\hat{v}_{i,N_t,2}^{1/2}}{\hat{v}_{i,N_t,1}^{1/2}} \times \frac{\hat{v}_{i,N_t,2}}{\hat{v}_{i,N_t,1}} \\
&\quad \times \exp\left(-\frac{(\hat{f}_{i,N_t,1}(X_{N_t}) - f_i(X_{N_t}) - \varepsilon_{N_t})^2 - (\hat{f}_{i,N_t,2}(X_{N_t}) - f_i(X_{N_t}) - \varepsilon_{N_t})^2}{2\hat{v}_{i,N_t}}\right) \\
&= \frac{W_{i,N_t,1}}{W_{i,N_t,2}} \times \frac{\hat{v}_{i,N_t,2}^{1/2}}{\hat{v}_{i,N_t,1}^{1/2}} \exp\left(\frac{(\hat{f}_{i,N_t,2}(X_{N_t}) - f_i(X_{N_t}))^2 - (\hat{f}_{i,N_t,1}(X_{N_t}) - f_i(X_{N_t}))^2}{2\hat{v}_{i,N_t}}\right) \\
&\quad \times \exp\left(\frac{\varepsilon_{N_t}(\hat{f}_{i,N_t,1}(X_{N_t}) - \hat{f}_{i,N_t,2}(X_{N_t}))}{\hat{v}_{i,N_t}}\right) \\
&= \frac{W_{i,N_t,1}}{W_{i,N_t,2}} \times \frac{\hat{v}_{i,N_t,2}^{1/2}}{\hat{v}_{i,N_t,1}^{1/2}} \exp(T_{1t} + T_{2t}),
\end{aligned}$$

where

$$T_{1t} = \frac{(\hat{f}_{i,N_t,2}(X_{N_t}) - f_i(X_{N_t}))^2 - (\hat{f}_{i,N_t,1}(X_{N_t}) - f_i(X_{N_t}))^2}{2\hat{v}_{i,N_t}},$$

and

$$T_{2t} = \frac{\varepsilon_{N_t}(\hat{f}_{i,N_t,1}(X_{N_t}) - \hat{f}_{i,N_t,2}(X_{N_t}))}{\hat{v}_{i,N_t}}.$$

Thus, for each $T \geq 1$,

$$\frac{W_{i,N_{T+1,1}}}{W_{i,N_{T+1,2}}} = \left(\prod_{t=1}^T \frac{\hat{v}_{i,N_t,2}^{1/2}}{\hat{v}_{i,N_t,1}^{1/2}}\right) \exp\left(\sum_{t=1}^T T_{1t} + \sum_{t=1}^T T_{2t}\right). \quad (30)$$

Then define $\xi_t = \varepsilon_{N_t}(\hat{f}_{i,N_t,1}(X_{N_t}) - f_i(X_{N_t}))$ and $\xi_t^* = \varepsilon_{N_t}(\hat{f}_{i,N_t,2}(X_{N_t}) - f_i(X_{N_t}))$. Since $E(\varepsilon_{N_t} | \mathcal{F}_{N_t}) = 0$, it follows by Assumption C, Assumption 0 and Lemma 1 that for every $\tau > 0$ and every $T \geq 1$,

$$P\left(\sum_{t=1}^T \xi_t > T\tau\right) < \exp\left(-\frac{T\tau^2}{2c_2^2(v^2 + c\tau/c_2)}\right).$$

Replacing τ by $\frac{(\log T)^b}{\sqrt{T}}\tau$, we obtain

$$\sum_{t=1}^T \xi_t = o(\sqrt{T}(\log T)^b) \quad (31)$$

almost surely by Borel-Cantelli lemma. By the same argument, $\sum_{t=1}^T \xi_t^* = o(\sqrt{T}(\log T)^b)$ almost surely. Note that for each $T \geq 1$,

$$\begin{aligned}
\hat{v}_{i,N_{T+1,1}} &= \frac{\sum_{t=1}^T (\hat{f}_{i,N_t,1}(X_{N_t}) - Y_{i,N_t})^2}{T} \\
&= \frac{\sum_{t=1}^T (\hat{f}_{i,N_t,1}(X_{N_t}) - f_i(X_{N_t}) - \varepsilon_{N_t})^2}{T} \\
&= \frac{\sum_{t=1}^T (\hat{f}_{i,N_t,1}(X_{N_t}) - f_i(X_{N_t}))^2}{T} + \frac{\sum_{t=1}^T \varepsilon_{N_t}^2}{T} - \frac{2\sum_{t=1}^T \xi_t}{T}.
\end{aligned}$$

Similarly, for each $T \geq 1$

$$\hat{v}_{i,N_{T+1,2}} = \frac{\sum_{t=1}^T (\hat{f}_{i,N_t,2}(X_{N_t}) - f_i(X_{N_t}))^2}{T} + \frac{\sum_{t=1}^T \varepsilon_{N_t}^2}{T} - \frac{2\sum_{t=1}^T \xi_t^*}{T}.$$

By Assumption A and the previous two equations, we obtain that

$$\hat{v}_{i,N_t,1} < \hat{v}_{i,N_t,2} \quad (32)$$

almost surely for large enough t .

The boundedness of $\{\hat{v}_{i,N_t}, t \geq 1\}$ as implied in Assumption D enables us to apply Lemma 1 again to obtain that

$$\sum_{t=1}^T T_{2t} = o(\sqrt{T}(\log T)^b), \quad (33)$$

almost surely. By (31), (32) and Assumption A, we conclude from (30) that

$$\frac{W_{i,N_{T+1,1}}}{W_{i,N_{T+1,2}}} \rightarrow \infty \quad \text{a.s. as } T \rightarrow \infty.$$

This completes the proof of Theorem 3. \square

B.4 Proof of Theorem 4

Proof of Theorem 4. First, note that

$$\begin{aligned}
R_N(\eta) &= \sum_{n=1}^N (f^*(X_n) - f_{I_n}(X_n)) I(I_n = \tilde{i}_n) + \sum_{n=1}^N (f^*(X_n) - f_{I_n}(X_n)) I(I_n \neq \tilde{i}_n) \\
&\leq \sum_{n=1}^N (f^*(X_n) - f_{I_n}(X_n)) I(I_n = \tilde{i}_n, \tilde{i}_n \neq i^*(X_n)) + \sum_{n=1}^N AI(I_n \neq \tilde{i}_n) \\
&= \sum_{i=1}^l \sum_{n=1}^N (f^*(X_n) - f_{I_n}(X_n)) I(I_n = i, \tilde{i}_n \neq i^*(X_n), \tilde{i}_n = i) + \sum_{n=1}^N AI(I_n \neq \tilde{i}_n) \\
&=: \sum_{i=1}^l \sum_{n=1}^N R_{i,n} + R_{N,2}.
\end{aligned} \quad (34)$$

Then we partition the domain into $1/h^d$ bins with bin width h and denote the set of these bins by \mathcal{B} .

Given bin $B \in \mathcal{B}$, define $R_{i,n,B} = \sum_{m=1}^N R_{i,n} I(X_n \in B)$. Let x_B be the center point in B . Define the (nearly) optimal arm $\bar{i} = \bar{i}_B = \operatorname{argmax}_{i \leq i \leq i} f_i(x_B)$ and $\Delta_{i,B} = f_{\bar{i}}(x_B) - f_i(x_B)$. Let $S_{i,B} = \{n : l_{n_0} + 1 \leq n \leq N, I_n = i, \bar{i}_n \neq i^*(X_n), \bar{i}_n = i, X_n \in B\}$. Define $N_{i,B} = \max S_{i,B}$ if $S_{i,B} \neq \emptyset$ and $N_{i,B} = 0$ if $S_{i,B} = \emptyset$. Given $N_{i,B} = \bar{n}$, let $\sigma_i = \sigma_{i,t,\bar{n}} = \min\{n : \sum_{j=1}^n I(I_j = i, K(\frac{X_j - X_{\bar{n}}}{h}) \neq 0) \geq t\}$ be the earliest time point where $\{I_j = i, K(\frac{X_j - X_{\bar{n}}}{h}) \neq 0\}$ happens t times. Define $\tau_i = \tau_{i,\bar{n}} = \max\{t : \sigma_{i,t,\bar{n}} < \bar{n}\}$, which is the number of times $\{I_j = i, K(\frac{X_j - X_{\bar{n}}}{h}) \neq 0\}$ happens before the time point \bar{n} . Similarly, for the (nearly) optimal arm \bar{i} , define $\eta_i = \min\{n : \sum_{j=1}^n I(I_j = \bar{i}, K(\frac{X_j - X_{\bar{n}}}{h}) \neq 0) \geq t\}$ and $\bar{\tau} = \max\{t : \eta_t < \bar{n}\}$. Then, if $N_{i,B} = \bar{n} \neq 0$ and $\tau_i \geq 1$, by the kernel-UCB algorithm, we have $f_{i,\bar{n}}(X_{\bar{n}}) + U_{i,\bar{n}}(X_{\bar{n}}) \geq f_{\bar{i},\bar{n}}(X_{\bar{n}}) + U_{\bar{i},\bar{n}}(X_{\bar{n}})$, that is,

$$\sum_{t=1}^{\tau_i} \frac{Y_{i,\sigma_t}}{\tau_i} + \tilde{c} \sqrt{\frac{\log N}{\log N}} \geq \sum_{t=1}^{\bar{\tau}} \frac{Y_{\bar{i},\eta_t}^2}{\bar{\tau}} + \tilde{c} \sqrt{\frac{\log N}{\bar{\tau}}}, \quad (34)$$

Note that (34) implies at least one of the following three events occurs:

$$\begin{aligned} G_B &=: \left\{ \sum_{t=1}^{\tau_i} \frac{Y_{i,\sigma_t}}{\tau_i} - \sum_{t=1}^{\bar{\tau}} \frac{f_{\bar{i}}(X_{\eta_t})}{\bar{\tau}} > \tilde{c} \sqrt{\frac{\log N}{\tau_i}} \right\}, \\ F_B &=: \left\{ \sum_{t=1}^{\bar{\tau}} \frac{Y_{\bar{i},\eta_t}}{\bar{\tau}} - \sum_{t=1}^{\bar{\tau}} \frac{f_{\bar{i}}(X_{\eta_t})}{\bar{\tau}} < -\tilde{c} \sqrt{\frac{\log N}{\bar{\tau}}} \right\}, \text{ or} \\ H_B &=: \left\{ \sum_{t=1}^{\tau_i} \frac{f_i(X_{\sigma_t})}{\tau_i} + 2\tilde{c} \sqrt{\frac{\log N}{\tau_i}} > \sum_{t=1}^{\bar{\tau}} \frac{f_{\bar{i}}(X_{\eta_t})}{\bar{\tau}} \right\}. \end{aligned}$$

Since $\|f_i(X_{\sigma_t}) - f_i(x_B)\|_\infty \leq \rho h^s$ and $\|f_{\bar{i}}(X_{\eta_t}) - f_{\bar{i}}(x_B)\|_\infty \leq \rho h^s$,

$$\begin{aligned} H_B &\Rightarrow f_i(x_B) + \rho h^s + 2\tilde{c} \sqrt{\frac{\log N}{\tau_i}} > f_{\bar{i}}(x_B) - \rho h^s \\ &\Rightarrow 2\tilde{c} \sqrt{\frac{\log N}{\tau_i}} > \Delta_{i,B} - 2\rho h^s \\ &\Rightarrow \{\Delta_{i,B} \leq 4\rho h^s\} \text{ or } \{\Delta_{i,B} > 4\rho h^s, \tau_i < \frac{16\tilde{c}^2}{\Delta_{i,B}^2} \log N\}. \end{aligned} \quad (35)$$

By Lemma 1,

$$\begin{aligned} P(G_B, N_{i,B} \neq 0, \tau_i > \log N) &\leq N^2 \exp\left(-\frac{\tilde{c}^2 \log N}{2(\rho^2 + \tilde{c}^2)}\right) \\ &\leq \frac{1}{N}, \end{aligned} \quad (36)$$

where the last inequality holds since $\tilde{c} > \max\{2\sqrt{3}\rho, 12\tilde{c}\}$.

Similarly, we can show that

$$P(F_B, N_{i,B} \neq 0, \tau_i > \log N) \leq \frac{1}{N}. \quad (37)$$

Note that

$$\begin{aligned} R_{i,n,B} &\leq R_{i,n,B} I(N_{i,B} = 0) + R_{i,n,B} I(\tau_i \leq \log N, N_{i,B} \neq 0) + \\ &\quad R_{i,n,B} I(\tau_i > \log N, N_{i,B} \neq 0, G_B) + R_{i,n,B} I(\tau_i > \log N, N_{i,B} \neq 0, F_B) + \\ &\quad R_{i,n,B} I(\tau_i > \log N, N_{i,B} \neq 0, H_B) \\ &\leq R_{i,n,B} I(N_{i,B} = 0) + A \log N + AN I(\tau_i > \log N, N_{i,B} \neq 0, G_B) + \\ &\quad AN I(\tau_i > \log N, N_{i,B} \neq 0, F_B) + 6\rho h^s \tau_{i,B} + \left(\frac{3}{2} \Delta_{i,B}\right) \left(\frac{16\tilde{c}^2}{\Delta_{i,B}^2} \log N\right) I(\Delta_{i,B} > 4\rho h^s), \end{aligned}$$

where the last inequality follows by (35), and $\tau_{i,B} = \sum_{n=1}^N I(I_n = i, X_n \in B)$. Then by (36), (37) and the definition of $N_{i,B}$,

$$\begin{aligned} E\left(\sum_{i=1}^L \sum_{n=1}^N R_{i,n}\right) &= \sum_{i=1}^L \sum_{B \in \mathcal{B}} E(R_{i,n,B}) \\ &\leq ALn_0 + ALh^{-d} \log N + 2A + 6\rho N h^s + \frac{6\tilde{c}^2 l}{\rho h^{s+d}} \log N. \end{aligned} \quad (38)$$

Also, taking $\delta = 1/N$, we have by (28) that

$$\begin{aligned} E(R_{N,2}) &\leq AN\delta + A \sum_{n=1}^N \pi_n + A \sqrt{\frac{N}{2}} \log \left(\frac{1}{\delta}\right) \\ &\leq A + A \sum_{n=1}^N \pi_n + A \sqrt{\frac{N}{2}} \log N. \end{aligned} \quad (39)$$

By (33), (38), (39) and our choice of n_0, h and π_n , we obtain (7). \square

Appendix C. Additional Numerical Results

Under the same settings of Section 7 with the Yahoo! data set, we implement additional algorithms as follows.

Simple-SIR-kernel : This algorithm is the same as the SIR-kernel algorithm described in Section 7 except that the dimension reduction matrix is estimated using only the data collected during the forced sampling stage. That is, for every $n > l_{n_0}$, the Nadaraya-Watson estimator of $f_i(x)$ shown in (5) is modified to be

$$\hat{f}_{i,n}(x) = \frac{\sum_{j \in I_{i,n}} Y_{i,j} K_i \left(\frac{\hat{B}_{i,l_{n_0}}^{*T} x^* - \hat{B}_{i,l_{n_0}}^{*T} X_j^*}{h_{n-1}} \right)}{\sum_{j \in I_{i,n}} K_i \left(\frac{\hat{B}_{i,l_{n_0}}^{*T} x^* - \hat{B}_{i,l_{n_0}}^{*T} X_j^*}{h_{n-1}} \right)}, \quad (40)$$

where the forced sampling size for each arm is $n_0 = 1000$ and the bandwidth is $h_n = n^{-1/10}$.

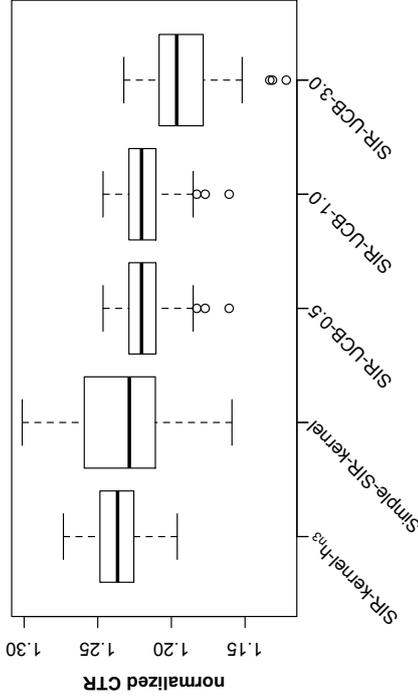


Figure 3: Boxplots of normalized CTRs of various algorithms on the news article recommendation data set. Algorithms include (from left to right): SIR-kernel ($h_{0.3}$), Simple-SIR-kernel, SIR-UCB ($\tilde{c}_0 = 0.5$), SIR-UCB ($\tilde{c}_0 = 1.0$) and SIR-UCB ($\tilde{c}_0 = 3.0$). CTRs are normalized with respect to the random algorithm.

SIR-UCB : This algorithm modifies the kernel estimation based UCB algorithm described in Appendix A to handle covariates with high dimensions. Rather than using the original covariates, we apply the SIR method to estimate the dimension reduction matrices and use them to transform covariates to lower dimensional. These transformed covariates are subsequently applied to compute the kernel estimation based UCB index. That is, at each time point n after the forced sampling stage, we pull the arm with the highest UCB index $\hat{f}_{i,n}(X_n) + U_{i,n}^*(X_n)$, where $\hat{f}_{i,n}(X_n)$ is defined in (5) and

$$U_{i,n}^*(x) = \frac{\tilde{c}_0 \sqrt{\sum_{j \in I_{i,n}} K^2 \left(\frac{\hat{B}_{i,n}^{*T} x^* - \hat{B}_{i,n-1}^{*T} X_n^*}{\sum_{j \in I_{i,n}} \hat{B}_{i,n}^{*T} x^* - \hat{B}_{i,n-1}^{*T} X_n^*} \right)}}{\sum_{j \in I_{i,n}} K \left(\frac{\hat{B}_{i,n}^{*T} x^* - \hat{B}_{i,n-1}^{*T} X_n^*}{h_{n-1}} \right)},$$

with $\hat{B}_{i,n}^*$, x^* and X_n^* defined in Section 4.3. We set $\tilde{c}_0 = 0.5, 1$ or 3 and $h_n = n^{-1/10}$.

The algorithms above are evaluated in the same manner as is described in Section 7, and the resulting normalized CTRs are summarized in the boxplots in Fig. 3. Although the averaged CTRs of the simple-SIR-kernel appears to be similar to SIR-kernel, the variation of the CTRs clearly enlarges as we use only the forced sampling stage to estimate the dimension reduction matrices. The SIR-UCB algorithm does not show significant improvement over the SIR-kernel algorithm either.

Remark 1. Because of the curse of dimensionality, the kernel estimation in Section 4.1 cannot be directly applied to the Yahoo! data set. As described in Section 4.3, one way to address this issue is to assume that for each arm i ($i = 1, 2, \dots, l$), there exists a dimension reduction matrix B_i and a function $g(\cdot)$ such that $\tilde{x}_i = B_i^T x$ becomes lower dimensional covariate and $f_i(x) = g(\tilde{x}_i)$. If B_i (or more precisely, $\text{span}(B_i)$) is known, we can simply work with the lower dimensional covariates (which can be different for different arms), and the kernel estimation algorithm in Section 4.1 still applies. Indeed, we note that if B_i is known, the consistency and finite-time regret analysis (with rate in accordance with the lower dimension) can still be established in a way similar to that of Theorem 1 and Corollary 2.

In practice, since the B_i is unknown, it is natural to estimate it by introducing a dimension reduction method like SIR. The theoretical implications of the dimension reduction procedure is not yet clear to us. To provide some numerical guidance on how to apply SIR, we explored two different ways of estimating B_i using the Yahoo! data. One is the SIR-kernel algorithm, where the estimator for B_i gets updated as more and more data is collected throughout the total time horizon. Alternatively, we considered here the Simple-SIR-kernel algorithm, where only data from the initial forced sampling stage is used to generate a consistent estimator for B_i (Zhu et al., 1996); Subsequently, with the lower-dimensional covariates from a fixed dimension reduction matrix, the kernel estimation can be applied to the remaining time period. Our numerical result favors the former way of applying SIR.

References

- Y. Abbasi-Yadkori. Forced-exploration based algorithms for playing in bandits with large action sets. Master's thesis, Department of Computing Science, University of Alberta, 2009.
- Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In *Proceedings of the 25th Conference on Neural Information Processing Systems*, 2011.
- S. Agrawal and N. Goyal. Thompson sampling for contextual bandits with linear payoffs. In *Proceedings of the 30th International Conference on Machine Learning*, 2013.
- J.-Y. Audibert. Fast learning rates in statistical inference through aggregation. *The Annals of Statistics*, 37:1591–1646, 2009.
- P. Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2002.
- P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256, 2002.
- P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32:48–77, 2003.
- P. Auer, R. Ortner, and C. Szepesvári. Improved rates for the stochastic continuum-armed bandit problem. In *Proceedings of the 20th Annual Conference on Learning Theory*, 2007.

- G. Bartók and C. Szepesvári. Partial monitoring with side information. In *Proceedings of the 23rd International Conference on Algorithmic Learning Theory*, 2012.
- D. A. Berry and B. Fristedt. *Bandit Problems: Sequential Allocation of Experiments*. Chapman and Hall, New York, 1985.
- L. Birgé and Massart. Minimum contrast estimators on sieves: exponential bounds and rates of convergence. *Bernoulli*, 4:329–375, 1998.
- S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5:1–122, 2012.
- S. Bubeck, R. Munos, G. Stoltz, and C. Szepesvári. X-armed bandits. *Journal of Machine Learning Research*, 12:1655–1695, 2011.
- S. Bubeck, N. Cesa-Bianchi, and G. Lugosi. Bandits with heavy tail. *IEEE Transactions on Information Theory*, 59(11):7711–7717, 2013.
- N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning and Games*. Cambridge University Press, Cambridge, UK, 2006.
- O. Chapelle and L. Li. An empirical evaluation of Thompson sampling. In *Proceedings of the 25th Conference on Neural Information Processing Systems*, pages 2249–2257, 2011.
- X. Chen, C. Zou, and R. D. Cook. Coordinate-independent sparse sufficient dimension reduction and variable selection. *The Annals of Statistics*, 38:3696–3723, 2010.
- W. Chu, L. Li, L. Reyzin, and R. E. Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, 2011.
- R. D. Cook. Fisher lecture: Dimension reduction in regression. *Statistical Science*, 22:1–26, 2007.
- R. D. Cook, L. M. Forzani, and D. R. Tomassi. LDR: A package for likelihood-based sufficient dimension reduction. *Journal of Statistical Software*, 39, 2011.
- V. Dani, T. P. Hayes, and S. M. Kakade. Stochastic linear optimization under bandit feedback. In *Proceedings of the 21st Annual Conference on Learning Theory*, pages 355–366, 2008.
- L. P. Devroye. The uniform convergence of the Nadaraya-Watson regression function estimate. *The Canadian Journal of Statistics*, 6:179–191, 1978.
- M. Dudík, D. Hsu, S. Kale, N. Karampatzakis, J. Langford, L. Reyzin, and T. Zhang. Efficient optimal learning for contextual bandits. In *Proceedings of the 27th Annual Conference on Uncertainty in Artificial Intelligence*, 2011.
- J. C. Gittins. *Multi-Armed Bandit Allocation Indices*. Wiley, New York, 1989.
- A. Goldenshluger and A. Zeevi. Woodroofe’s one-armed bandit problem revisited. *The Annals of Applied Probability*, 19:1603–1633, 2009.
- A. Goldenshluger and A. Zeevi. A linear response bandit problem. *Stochastic Systems*, 3: 230–261, 2013.
- B. E. Hansen. Uniform convergence rates for kernel estimation with dependent data. *Econometric Theory*, 24:726–748, 2008.
- W. Härdle and S. Luckhaus. Uniform consistency of a class of regression function estimators. *The Annals of Statistics*, 12:612–623, 1984.
- R. Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *Proceedings of the 18th Conference on Neural Information Processing Systems*, 2004.
- R. Kleinberg, A. Shwartz, and E. Upfal. Multi-armed bandits in metric spaces. In *Proceedings of the 40th Symposium on Theory of Computing*, 2007.
- T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4–22, 1985.
- J. Langford and T. Zhang. The epoch-greedy algorithm for contextual multi-armed bandits. In *Proceedings of the 21th Conference on Neural Information Processing Systems*, 2007.
- K.-C. Li. Sliced inverse regression for dimension reduction, with discussions. *Journal of the American Statistical Association*, 86:316–342, 1991.
- L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International World Wide Web Conference*, 2010.
- T. Lu, D. Pál, and M. Pál. Showing relevant ads via Lipschitz context multi-armed bandits. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, 2010.
- O.-A. Maillard and R. Munos. Adaptive bandits: Towards the best history-dependent strategy. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, 2011.
- B. C. May, N. Korda, A. Lee, and D. S. Leslie. Optimistic Bayesian sampling in contextual bandit problems. *Journal of Machine Learning Research*, 13:2069–2106, 2012.
- V. Perchet and P. Rigollet. The multi-armed bandit problem with covariates. *The Annals of Statistics*, 41:693–721, 2013.
- W. Qian and Y. Yang. Randomized allocation with dimension reduction in a bandit problem with covariates. In *Proceedings of the 9th International Conference on Fuzzy Systems and Knowledge Discovery*, pages 1537–1541, 2012.
- P. Rigollet and A. Tsybakov. Sparse estimation by exponential weighting. *Statistical Science*, 27:558–575, 2012.

- P. Rigollet and A. Zeevi. Nonparametric bandits with covariates. In *Proceedings of the 23rd International Conference on Learning Theory*, pages 54–66. Omnipress, 2010.
- H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58:527–535, 1954.
- P. Rusmevichientong and J. N. Tsitsiklis. Linearly parameterized bandits. *Mathematics of Operations Research*, 35:395–411, 2010.
- A. Slivkins. Contextual bandits with similarity information. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 679–702, 2011.
- Z. Wang, S. Paterlini, F. Gao, and Y. Yang. Adaptive minimax regression estimation over sparse l_q -hulls. *Journal of Machine Learning Research*, 15:1675–1711, 2014.
- X. Wei and Y. Yang. Robust forecast combination. *Journal of Econometrics*, 22:1021–1040, 2012.
- D. Williams. *Probability with Martingales*. Cambridge University Press, Cambridge, UK, 1991.
- M. Woodroofe. A one-armed bandit problem with a concomitant variable. *Journal of the American Statistical Association*, 74:799–806, 1979.
- Yahoo! Academic Relations. Yahoo! front page today module user click log dataset (version 1.0). 2011. Available from <http://webscope.sandbox.yahoo.com>.
- Y. Yang. Combining forecasting procedures: Some theoretical results. *Econometric Theory*, 20:176–222, 2004.
- Y. Yang and D. Zhu. Randomized allocation with nonparametric estimation for a multi-armed bandit problem with covariates. *The Annals of Statistics*, 30:100–121, 2002.
- Li-Xing Zhu, Kai-Tai Fang, et al. Asymptotics for kernel estimate of sliced inverse regression. *The Annals of Statistics*, 24(3):1053–1068, 1996.
- H. Zou and Y. Yang. Combining time series models for forecasting. *International Journal of Forecasting*, 20:69–84, 2004.

A General Framework for Consistency of Principal Component Analysis

Dan Shen

*Interdisciplinary Data Sciences Consortium
Department of Mathematics and Statistics
University of South Florida
Tampa, FL 33620-5700, USA*

DANSHEN@USF.EDU

Haipeng Shen

*School of Business
University of Hong Kong
Pokfulam, Hong Kong*

HAIPENG@HKU.HK

J. S. Marron

*Department of Statistics and Operations Research
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599-3260, USA*

MARRON@EMAIL.UNC.EDU

Editor: massimiliano pontil

Abstract

A general asymptotic framework is developed for studying consistency properties of principal component analysis (PCA). Our framework includes several previously studied domains of asymptotics as special cases and allows one to investigate interesting connections and transitions among the various domains. More importantly, it enables us to investigate asymptotic scenarios that have not been considered before, and gain new insights into the consistency, subspace consistency and strong inconsistency regions of PCA and the boundaries among them. We also establish the corresponding convergence rate within each region. Under general spike covariance models, the dimension (or number of variables) discourages the consistency of PCA, while the sample size and spike information (the relative size of the population eigenvalues) encourage PCA consistency. Our framework nicely illustrates the relationship among these three types of information in terms of dimension, sample size and spike size, and rigorously characterizes how their relationships affect PCA consistency.

Keywords: High dimension low sample size, PCA, Random matrix, Spike model

1. Introduction

Principal Component Analysis (PCA) is an important visualization and dimension reduction tool which finds orthogonal directions reflecting maximal variation in the data. This allows the low dimensional representation of data, by projecting data onto these directions. PCA is usually obtained by an eigen decomposition of the sample variance-covariance matrix of the data. Properties of the sample eigenvalues and eigenvectors have been analyzed under several domains of asymptotics.

In this paper, we develop a *general asymptotic framework* to explore interesting transitions among the various asymptotic domains. The general framework includes the tradi-

tional asymptotic setups as special cases, and furthermore it allows a careful study of the connections among the various setups. More importantly, we investigate a wide range of interesting scenarios that have not been considered before, and offer new insights into the *consistency* (in the sense that the angle between estimated and population eigen directions tends to 0, or the inner product tends to 1) and *strong-inconsistency* (where the angle tends to $\pi/2$, i.e., the inner product tends to 0) properties of PCA, along with some technically challenging convergence rates.

Existing asymptotic studies of PCA roughly fall into four domains:

- (a) the **classical** domain of asymptotics, under which the sample size $n \rightarrow \infty$ and the dimension d is fixed (hence the ratio $n/d \rightarrow \infty$). For example, see Girshick (1939); Lawley (1956); Anderson (1963, 1984); Jackson (1991).
- (b) the **random matrix** theory domain, where both the sample size n and the dimension d increase to infinity, with the ratio $n/d \rightarrow c$, a constant mostly assumed to be within $(0, \infty)$. Representative work includes Bielei and Mietzner (1994); Watkin and Nadal (1994); Reimann et al. (1996); Hoyle and Rattray (2003) from the statistical physics literature, as well as Johnstone (2001); Baik et al. (2005); Baik and Silverstein (2006); Onatski (2012); Paul (2007); Nadler (2008); Johnstone and Lu (2009); Lee et al. (2010); Benaych-Georges and Nadakuditi (2011) from the statistics literature.
- (c) the **high dimension low sample size (HDLSS)** domain of asymptotics, which is based on the limit, as the dimension $d \rightarrow \infty$, with the sample size n being fixed (hence the ratio $n/d \rightarrow 0$). HDLSS asymptotics was originally studied by Casella and Hwang (1982), and rediscovered by Hall et al. (2005). PCA has been studied using the HDLSS asymptotics by Ahn et al. (2007); Jung and Marron (2009).
- (d) the **increasing signal strength** domain of asymptotics, where n, d are fixed and the signal strength tends to infinity. Such a setting is studied in Nadler (2008).

PCA consistency and (strong) inconsistency, defined in terms of angles, are important properties that have been studied before. A common technical device is the spike covariance model, initially introduced by Johnstone (2001). This model has been used in this context by, for example, Nadler (2008); Johnstone and Lu (2009); Jung and Marron (2009). Recently, Ma (2013) formulates sparse PCA (Zou et al., 2006) through iterative thresholding and studies its asymptotic properties under the spike model. An interesting, more general, model has been considered by Benaych-Georges and Nadakuditi (2011).

Under the spike model, the first few eigenvalues are much larger than the others. A *major message of the present paper* is that there are three critical features whose relationships drive the consistency properties of PCA, namely

- (1) the *sample size*: the sample size n *encourages* the consistency of the sample eigenvectors, meaning that more samples tend towards more frequent consistency;
- (2) the *dimension*: the dimension d *discourages* the consistency of the sample eigenvectors, meaning that higher d tends towards less frequent consistency;
- (3) the *spike signal*: the relative sizes of the several leading eigenvalues similarly encourage the consistency.

Our general framework considers increasing sample size n , increasing dimension d , and increasing spike signal. We clearly characterize how their relationships determine the regions of consistency and strong-inconsistency of PCA, along with the boundary in-between.

Note that the classical domain ((a) above) assumes increasing sample size n while fixing dimension d ; the random matrix domain ((b) above) assumes increasing sample size n and increasing dimension d , while fixing the spike signal; the HDLSS domain ((c) above) fixes the sample size, and increases the dimension and the spike signal; the increasing signal strength domain ((d) above) assumes increasing the spike signal, while fixing the sample size and the dimension; thus each of these three domains is a boundary case of our framework. Our theorems, when restricted to these existing domains of asymptotics, are consistent with known results.

In addition, our theorems go beyond these known results to demonstrate the transitions among the existing domains of asymptotics, and for the first time to the best of our knowledge, enable one to understand interesting connections among them. Finally, we also establish novel results on rates of convergence.

Sections 3 and 4 formally state very general theorems for multiple component spike models. For illustration purposes only, in this section we first consider Examples 1 and 2 under some strong assumptions, which provide intuitive insights regarding the much more general theory presented in Sections 3 and 4. In addition, we use Example 3 to show the application of our theoretical study to the factor model considered by Fan et al. (2013).

For Examples 1 and 2, to better demonstrate the connection with existing results, the three types of features (sample size, dimension, and spike signal) and their relationships are mathematically quantified by two indices, namely the *spike index* α and the *sample index* γ . Within the context of these examples, we point out the significant contributions of our results in comparison with existing results. The comparisons and connections are graphically illustrated in Figure 1 and discussed below.

Example 1 Single-component Spike Model Assume that X_1, \dots, X_n are sample vectors from a d -dimensional distribution with zero mean and covariance matrix Σ , where the entries of $\Sigma^{-\frac{1}{2}}X_i$ are i.i.d. random variables with zero mean, unit variance and finite fourth moment. (A special case: X_i is from the d -dimensional normal distribution $N(0, \Sigma)$). In addition, assume that the sample size $n = d^\gamma$ ($\gamma \geq 0$ is defined as the *sample index*), and the covariance matrix Σ has the following eigenvalues:

$$\lambda_1 = c_1 d^\alpha, \lambda_2 = \dots = \lambda_d = 1, \alpha \geq 0,$$

where the constant α is defined as the *spike index*.

Corollary B.2 in the supplementary materials, when applied to this example, shows that the maximal sample eigenvector is consistent when $\alpha + \gamma > 1$ (grey region in Figure 1(A)), and strongly inconsistent when $0 \leq \alpha + \gamma < 1$ (white triangle in Figure 1(A)). These very general new results nicely connect with many existing ones:

- **Previous Results I - the classical domain:**

Under the normal assumption, Theorem 1 of Anderson (1963) implied that for fixed dimension d and finite eigenvalues, when the sample size $n \rightarrow \infty$ (i.e. $\gamma \rightarrow \infty$, the limit on the vertical axis), the maximal sample eigenvector is consistent. This case is the upper left corner of Figure 1(A).

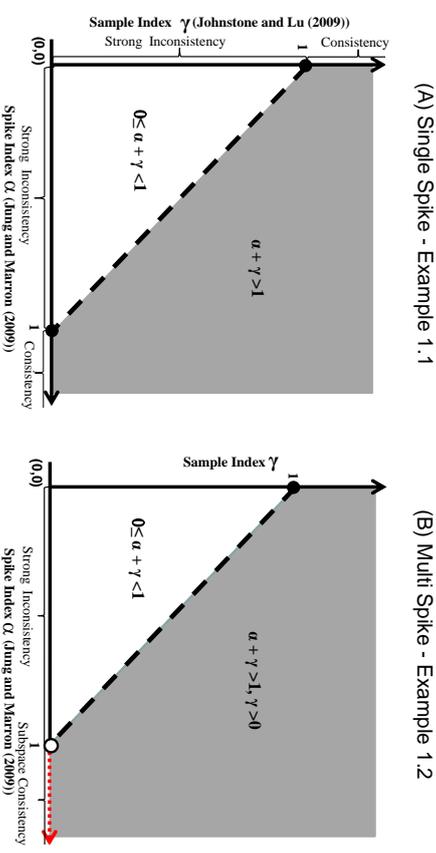


Figure 1: General consistency and strong inconsistency regions for PCA, as a function of the spike index α and the sample index γ . Panel (A) - single spike model in Example 1: PCA is consistent in the grey region ($\alpha + \gamma > 1$), and strongly inconsistent on the white triangle ($0 \leq \alpha + \gamma < 1$). Panel (B) - multiple spike model in Example 2: the first m sample PCs are consistent in the grey region ($\alpha + \gamma > 1, \gamma > 0$), subspace consistent on the dotted line ($\alpha > 1, \gamma = 0$) on the horizontal axis, and strongly inconsistent on the white triangle ($0 \leq \alpha + \gamma < 1$).

- **Previous Results II - the random matrix domain:**

(a) Assuming normality, the results of Johnstone and Lu (2009) appear on the vertical axis in Panel (A) where the spike index $\alpha = 0$ (as they fix the spike information): the first sample eigenvector is consistent when the sample index $\gamma > 1$ and strongly inconsistent when $\gamma < 1$.

(b) Again, under the normal assumption, Nadler (2008) explored the interesting boundary case of $\alpha = 0, \gamma = 1$ (i.e. $\frac{d}{n} \rightarrow c$ for a constant c) and showed that $\langle \hat{u}_1, u_1 \rangle \xrightarrow{a.s.} \frac{((\lambda_1 - 1)^2 - c)^+}{(\lambda_1 - 1)^2 + c(\lambda_1 - 1)^2}$, where \hat{u}_1 and u_1 are the first sample and population eigenvector. This result appears in Panel (A) as the single solid circle $\gamma = 1$ on the vertical axis. Our general framework doesn't cover this boundary case and this boundary result is a complement of our theoretical results.

- **Previous Results III - the HDLSS domain:**

(a) The theorems of Jung and Marron (2009) are represented on the horizontal axis in Panel (A) when the sample index $\gamma = 0$ (as they fix the sample size): the maximal sample eigenvector is consistent with the first population eigenvector when the spike index $\alpha > 1$ and strongly inconsistent when $\alpha < 1$.

(b) Under the normal assumption, Jung et al. (2012) deeply explored limiting behavior at the boundary $\alpha = 1, \gamma = 0$ (i.e. $\frac{d}{\lambda_1} \rightarrow c$ for a constant c) and showed that $\langle \hat{u}_1, u_1 \rangle > 2 \Rightarrow \frac{d}{\lambda_1 c} \rightarrow c$ means convergence in distribution and $A \sim \chi^2$, the chi-squared distribution with n degrees of freedom. This result appears in Panel (A) as the single solid circle $\alpha = 1$ on the horizontal axis. This boundary case is again a complement of our general framework.

- **Our Results** hence nicely connect existing domains of asymptotics, and give a much more complete characterization for the regions of PCA consistency, subspace consistency, and strong inconsistency. We also investigate asymptotic properties of the other sample eigenvectors and all the sample eigenvalues.

Example 2 Multiple-component Spike Model Assume that the covariance matrix Σ in Example 1 has the following eigenvalues:

$$\lambda_j = \begin{cases} c_j d^\alpha & \text{if } j \leq m, \\ 1 & \text{if } j > m, \end{cases} \quad \alpha \geq 0,$$

where m is a finite positive integer, the constants $c_j, j = 1, \dots, m$, are positive and satisfy that $c_j > c_{j+1} > 1, j = 1, \dots, m - 1$.

Corollary B.1 in the supplementary materials, when applied to this example, shows that the first m sample eigenvectors are individually consistent with corresponding population eigenvectors when $\alpha + \gamma > 1, \gamma > 0$ (the grey region in Figure 1(B)), instead of being subspace consistent (Jung and Marron, 2009), and strongly inconsistent when $\alpha + \gamma < 1$ (the white triangle in Panel (B)). This very general new result connects with many others in the existing literature:

- **Previous Results I - the classical domain.**

Assuming normality, Theorem 1 of Anderson (1963) implied that for fixed dimension d and finite eigenvalues, when the sample size $n \rightarrow \infty$ (i.e. $\gamma \rightarrow \infty$, the limit on the vertical axis), the first m sample eigenvectors are consistent, while the other sample eigenvectors are subspace consistent. This case is the upper left corner of Figure 1(B).

- **Previous Results II - the random matrix domain.**

The following results are under the normal assumption. Paul (2007) explored asymptotic properties of the first m eigenvectors and eigenvalues in the interesting boundary case of $\alpha = 0, \gamma = 1$, i.e., $\frac{d}{n} \rightarrow c$ with $c \in (0, 1)$ and showed that $\langle \hat{u}_j, u_j \rangle \xrightarrow{a.s.} \frac{((\lambda_j - 1)^2 - c)}{(\lambda_j - 1)^2 + c(\lambda_j - 1)}$ for $j = 1, \dots, m$. This result appears in Panel (B) as the solid circle $\gamma = 1$ on the vertical axis. This boundary case is a complement of our results for multiple spike models with distinct eigenvalues (Section B.1 of the supplementary materials). Paul and Johnstone (2012) considered a similar framework but from a minimax risk analysis perspective. Nadler (2008); Johnstone and Lu (2009) did not study multiple spike models.

- **Previous Results III - the HDLSS domain:**

The theorems of Jung and Marron (2009) are valid on the horizontal axis in Panel (B) where the sample index $\gamma = 0$. In particular, for this example, their results showed that the first m sample eigenvectors are not respectively consistent with the corresponding population eigenvectors when the spike index $\alpha > 1$ (the horizontal dotted red line segment), instead they are subspace consistent with their corresponding population eigenvectors, and are strongly inconsistent when the spike index $\alpha < 1$ (the horizontal solid line segment). They and Jung et al. (2012) did not study the asymptotic behavior on the boundary - the single open circle ($\alpha = 1, \gamma = 0$) on the horizontal axis.

- **Our Results** cover the classical domain, and are stronger than what Jung and Marron (2009) obtained: the increasing sample size enables us to separate out the first few leading eigenvectors and characterize individual consistency, while only subspace consistency was obtained by Jung and Marron (2009).

Example 3 The Factor Model of Fan et al. (2013) Consider the following model:

$$\mathbf{y}_t = \mathbf{B}\mathbf{f}_t + \mathbf{E}_t,$$

where $\mathbf{y}_t = (y_{t,1}, \dots, y_{t,d})^T$ is the d -dimensional response vector, $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_d)^T$ is the $d \times m$ (m is fixed) loading matrix, \mathbf{f}_t is the $m \times 1$ vector of common factors, and $\mathbf{E}_t = (e_{t,1}, \dots, e_{t,d})^T$ is the d -dimensional noise vector, $t = 1, \dots, T$. The noise vector \mathbf{E}_t is independent of \mathbf{f}_t . Then the population covariance matrix of \mathbf{y}_t is

$$\Sigma = \mathbf{B}\text{cov}(\mathbf{f}_t)\mathbf{B}^T + \Sigma_E,$$

where Σ_E is the covariance matrix of \mathbf{E}_t . Fan et al. (2013) assumes that the first m eigenvalues of $\mathbf{B}\text{cov}(\mathbf{f}_t)\mathbf{B}^T$ increase with d as $d \rightarrow \infty$, whereas all the eigenvalues of Σ_E are bounded. It then follows that $\lambda_m(\Sigma) \asymp \lambda_{m+1}(\Sigma) \asymp \dots \asymp \lambda_d(\Sigma) \asymp 1$, as $d \rightarrow \infty$. Then our theorems are applicable to this factor model when $\mathbf{f}_1, \dots, \mathbf{f}_T$ is i.i.d., and $\mathbf{E}_1, \dots, \mathbf{E}_T$ is i.i.d.

Under the above assumptions of the factor model, we have $d/(\mathbb{T}\lambda_m(\Sigma)) \rightarrow 0$. Then according to our Theorem 1 (together with the third comment after the theorem), the first m sample eigenvalues and eigenvectors are consistent. On the other hand, Fan et al. (2013) proposed the consistent principal orthogonal complement thresholding (POET) estimator for the covariance matrix Σ , which is obtained by keeping the first m sample eigenvalues and eigenvectors, and thresholding the residual sample matrix. Hence, our theorem offers another theoretical support on the consistency of their POET estimator.

The rest of the paper is organized as follows. Section 2 first introduces our notations and relevant consistency concepts. Section 3 studies the PCA asymptotics of spike models with increasing sample size n . We state the main results of our paper - Theorem 1 for multiple-component spike models where the dominating eigenvalues are inseparable. Theorem 2 in Section 4 then is about the HDLSS asymptotics of PCA, where the sample size n is fixed, for spike models with inseparable eigenvalues. Section 5 contains some discussions about the asymptotic properties of PCA when some eigenvalues equal to zero and the challenges to obtain non-asymptotic results. Section 7 contains the technical proofs of Theorem 1 and

the relevant lemmas. The supplementary materials contain the corresponding corollaries of Theorems 1 and 2, for multiple-spike models with distinct eigenvalues and single spike models, along with the proofs of Theorem 2 and all the corollaries.

2. Notations and Concepts

We now introduce some necessary notations, and define consistency concepts relevant for our asymptotic study.

2.1 Notation

Let the population covariance matrix be Σ , whose eigen decomposition is

$$\Sigma = U\Lambda U^T,$$

where Λ is the diagonal matrix of population eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d$, and U is the matrix of the corresponding eigenvectors $U = [u_1, \dots, u_d]$.

As in Jung and Marron (2009), assume that X_1, \dots, X_n are i.i.d. d -dimensional random sample vectors and have the following representation

$$X_i = \sum_{j=1}^d \lambda_j^{\frac{1}{2}} z_{i,j} u_j, \quad (1)$$

where the $z_{i,j}$'s are i.i.d. random variables with zero mean, unit variance, and finite fourth moment. An important special case is that they follow the standard normal distribution.

Assumption 1 X_1, \dots, X_n are a random sample having the distribution of (1).

Jung and Marron (2009) assumes that

$$Z_i = (z_{i,1}, \dots, z_{i,d})^T, \quad i = 1, \dots, n, \quad (2)$$

are independent and the elements $z_{i,j}$ within Z_i are ρ -mixing. This assumption leads to the convergence in probability results under the HDLSS domain in Jung and Marron (2009). Here we assume that the elements $z_{i,j}$ within Z_i are also independent. This helps to get the almost sure convergence results under our general framework, which includes the HDLSS domain. Assumption 1 is necessary to satisfy the conditions of Lemma 1 - the Bai-Yin's law (Bai and Yin, 1993), which is important for our results, for example, Theorem 1.

Denote the sample covariance matrix by $\hat{\Sigma} = n^{-1} X X^T$, where $X = [X_1, \dots, X_n]$. Note that $\hat{\Sigma}$ can also be decomposed as

$$\hat{\Sigma} = \hat{U} \hat{\Lambda} \hat{U}^T, \quad (3)$$

where $\hat{\Lambda}$ is the diagonal matrix of sample eigenvalues $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \dots \geq \hat{\lambda}_d$, and \hat{U} is the matrix of corresponding sample eigenvectors where $\hat{U} = [\hat{u}_1, \dots, \hat{u}_d]$.

Below we introduce asymptotic notations that will be used in our theoretical studies. Let τ stand for either n or d , depending on the context. Assume that $\{\xi_\tau : \tau = 1, \dots, \infty\}$ is a sequence of random variables, and $\{a_\tau : \tau = 1, \dots, \infty\}$ is a sequence of constant values.

- Denote $\xi_\tau = o_{a.s.}(a_\tau)$ if $\lim_{\tau \rightarrow \infty} \frac{\xi_\tau}{a_\tau} = 0$ almost surely.
- Denote $\xi_\tau = O_{a.s.}(a_\tau)$ if $\overline{\lim}_{\tau \rightarrow \infty} \left| \frac{\xi_\tau}{a_\tau} \right| \leq M$, where M is a positive constant.
- Denote almost surely $\xi_\tau \asymp a_\tau$ if $c_2 \leq \underline{\lim}_{\tau \rightarrow \infty} \frac{\xi_\tau}{a_\tau} \leq \overline{\lim}_{\tau \rightarrow \infty} \frac{\xi_\tau}{a_\tau} \leq c_1$ almost surely, for two constants $c_1 \geq c_2 > 0$.

In addition, we introduce the following notions to help understand the assumptions on the population eigenvalues in our theorems and corollaries. Assume that $\{a_\tau : \tau = 1, \dots, \infty\}$ and $\{b_\tau : \tau = 1, \dots, \infty\}$ are two sequences of real valued numbers.

- Denote $a_\tau \gg b_\tau$ if $\lim_{\tau \rightarrow \infty} \frac{b_\tau}{a_\tau} = 0$.
- Denote $a_\tau \asymp b_\tau$ if $c_2 \leq \underline{\lim}_{\tau \rightarrow \infty} \frac{a_\tau}{b_\tau} \leq \overline{\lim}_{\tau \rightarrow \infty} \frac{a_\tau}{b_\tau} \leq c_1$ for two constants $c_1 \geq c_2 > 0$.

2.2 Concepts

We now list several concepts about consistency and strong inconsistency, some of which are modified from the related concepts in Jung and Marron (2009) and Shen et al. (2013).

Let H be an index set, e.g. $H = \{m+1, \dots, d\}$, and then denote $S = \text{span}\{u_k, k \in H\}$ as the linear span generated by $\{u_k, k \in H\}$. Define $\text{angle}(\hat{u}_j, S)$ as the angle between the estimator \hat{u}_j and the subspace S , which is the angle between the estimator and its projection onto the subspace (Jung and Marron, 2009). For further clarification, we provide a graphical illustration of the angle in Section B of the supplement (Shen et al., 2015). As pointed out earlier, let τ stand for either n or d , depending on the context.

- If as $\tau \rightarrow \infty$, $\text{angle}(\hat{u}_j, S) \xrightarrow{a.s.} 0$, then \hat{u}_j is **subspace consistent** with S . If H only includes one index j such that $S = \text{span}\{u_j\}$, then $\text{angle}(\hat{u}_j, S) \xrightarrow{a.s.} 0$ is equivalent to $|\langle \hat{u}_j, u_j \rangle| \xrightarrow{a.s.} 1$, and \hat{u}_j is **consistent** with u_j .
- If as $\tau \rightarrow \infty$, $|\langle \hat{u}_j, u_j \rangle| \xrightarrow{a.s.} 0$, then \hat{u}_j is **strongly inconsistent** with u_j .

3. Cases with increasing sample size n

We study spike models with increasing sample size $n \rightarrow \infty$ in this section. As such, the eigenvalues λ_j and the dimension d depend on the sample size n , and will be denoted as $\lambda_j^{(n)}$ and $d(n)$ throughout this section. They can be viewed as sequences of constant values indexed by n . This section considers multiple-component spike models with inseparable eigenvalues and presents the main theorem of our paper. Section B of the supplementary materials reports the corollaries for multiple component spike models with distinct eigenvalues and single spike models.

We consider multiple spike models with m (a finite integer) dominating eigenvalues. These m eigenvalues can be grouped into r tiers, where the eigenvalues within the same tier have the same limit. To fixed ideas, the first m eigenvalues are grouped into r tiers where there are $q_l (> 0)$ eigenvalues in the l th tier with $\sum_{l=1}^r q_l = m$. Define $q_0 = 0$,

$q_{r+1} = d(n) - \sum_{l=1}^r q_l$, and the index set of the eigenvalues in the l th tier as

$$H_l = \left\{ \sum_{k=0}^{l-1} q_k + 1, \sum_{k=0}^{l-1} q_k + 2, \dots, \sum_{k=0}^{l-1} q_k + q_l \right\}, \quad l = 1, \dots, r+1. \quad (4)$$

Assume the eigenvalues in the l th tier have the same limit $\delta_l^{(n)} (> 0)$, i.e.

Assumption 2 $\lim_{n \rightarrow \infty} \frac{\lambda_j^{(n)}}{\delta_l^{(n)}} = 1, j \in H_l, l = 1, \dots, r$.

According to the above assumption, the eigenvalues that are in the same tier will have the same limit as n goes to infinity. As a result, we can show that the corresponding sample eigenvectors can not be consistently estimated individually. This motivates us to consider subspace consistency. In addition, we assume that the first m population eigenvalues from different tiers are asymptotically different, and dominate the additional population eigenvalues beyond the first r tiers that have the same limit c_λ :

Assumption 3 as $n \rightarrow \infty, \delta_1^{(n)} > \dots > \delta_r^{(n)} > \lambda_{m+1}^{(n)} \rightarrow \dots \rightarrow \lambda_{d(n)}^{(n)} \rightarrow c_\lambda > 0$.

For $i < j, \delta_i^{(n)} > \delta_j^{(n)}$ means that $\lim_{n \rightarrow \infty} \frac{\delta_i^{(n)}}{\delta_j^{(n)}} > 1$. This assumption allows $\delta_i^{(n)} \rightarrow \infty$ and $\delta_i^{(n)} \gg \delta_j^{(n)}$, which is not the case in Paul (2007). Regarding the constant c_λ , the second remark after Theorem 1 discusses what happens when $c_\lambda = 0$.

The above assumptions cover a general class of multiple spike models with tiered eigenvalues. A simple special case is the one where the eigenvalue matrix Λ is block diagonal: for $1 \leq h \leq r$, the h -th block of Λ is $\lambda_h^{(n)} I_{q_h}$ where I_{q_h} is the $q_h \times q_h$ identity matrix, with

$$\lambda_1^{(n)} > \lambda_2^{(n)} > \dots > \lambda_r^{(n)}, \quad q_1 + q_2 + \dots + q_r = m < d;$$

and the last block of Λ is $c_\lambda I_{d(n)-m}$ with $c_\lambda < \lambda_r^{(n)}$.

Under the above setup, Theorem 1 shows that the eigenvector estimates are either subspace consistent with the linear space spanned by the population eigenvectors, or strongly inconsistent. As discussed in the Introduction, Theorem 1 considers the delicate balance among the *sample size* n , the *spike signal* $\delta_l^{(n)}$, and the *dimension* $d(n)$, and characterize the various PCA consistency and strong-inconsistency regions. The three scenarios of Theorem 1 are arranged in the order of a decreasing amount of signal:

- Theorem 1(a): If the amount of signal dominates the amount of noise up to the r th tier, i.e. $\frac{d(n)}{n\delta_r^{(n)}} \rightarrow 0$, then the estimates for the eigenvectors in the first r tiers are subspace consistent, and the estimates for the higher order eigenvectors are also subspace consistent (but) at a different rate;
- Theorem 1(b): Otherwise, if the amount of signal dominates the amount of noise only up to the l th tier ($1 \leq h < r$), i.e. $\frac{d(n)}{n\delta_h^{(n)}} \rightarrow 0$ and $\frac{d(n)}{n\delta_{h+1}^{(n)}} \rightarrow \infty$, then the estimates for the eigenvectors in the first h tiers are subspace consistent, and the estimates for the other eigenvectors are strongly-inconsistent;

- Theorem 1(c): Finally, if the amount of noise always dominates, i.e. $\frac{d(n)}{n\lambda_1^{(n)}} \rightarrow \infty$, then the sample eigenvalues are asymptotically indistinguishable, and the sample eigenvectors are strongly inconsistent.

Before stating Theorem 1, we first introduce several notations. Define the subspace $S_l = \text{span}\{u_k, k \in H_l\}$ for $l = 1, \dots, r+1$ and denote $\delta_0^{(n)} = \infty$ for every n .

Theorem 1 Under Assumptions 1, 2 and 3, as $n \rightarrow \infty$, the following results hold.

(a) If $\frac{d(n)}{n\delta_r^{(n)}} \rightarrow 0$, then $\frac{\lambda_j}{\lambda_1} \xrightarrow{\text{a.s.}} 1, j = 1, \dots, m$, and $\text{angle}(\hat{u}_j, S_l) = \text{O}_{\text{a.s.}} \left(\left\{ \frac{\delta_l^{(n)}}{\delta_{l-1}^{(n)}} \vee \frac{\delta_{l+1}^{(n)}}{\delta_l^{(n)}} \right\}^{\frac{1}{2}} \right)$, $j \in H_l, l = 1, \dots, r-1$. In addition,

- If $\frac{d(n)}{n} \rightarrow 0$, then $\text{angle}(\hat{u}_j, S_l) = \text{O}_{\text{a.s.}} \left(\left\{ \frac{\delta_l^{(n)}}{\delta_{l-1}^{(n)}} \vee \frac{1}{\delta_l^{(n)}} \right\}^{\frac{1}{2}} \right), j \in H_l$ for $l = r$, and $\text{O}_{\text{a.s.}} \left(\left\{ \frac{1}{\delta_l^{(n)}} \right\}^{\frac{1}{2}} \right)$ for $l = r+1$.

- If $\frac{d(n)}{n} \rightarrow c, 0 < c \leq \infty$, then $\text{angle}(\hat{u}_j, S_l) = \text{O}_{\text{a.s.}} \left(\left\{ \frac{\delta_l^{(n)}}{\delta_{l-1}^{(n)}} \right\}^{\frac{1}{2}} \right) \vee \text{O}_{\text{a.s.}} \left(\left\{ \frac{d(n)}{n\delta_l^{(n)}} \right\}^{\frac{1}{2}} \right), j \in H_l$ for $l = r$, and $\text{O}_{\text{a.s.}} \left(\left\{ \frac{d(n)}{n\delta_l^{(n)}} \right\}^{\frac{1}{2}} \right)$ for $l = r+1$.

(b) If $\frac{d(n)}{n\delta_h^{(n)}} \rightarrow 0$ and $\frac{d(n)}{n\delta_{h+1}^{(n)}} \rightarrow \infty$, where $1 \leq h < r$, then $\frac{\lambda_j}{\lambda_1} \xrightarrow{\text{a.s.}} 1, j \in H_l, l = 1, \dots, h$,

and the other non-zero $\frac{n\lambda_j}{d(n)} \xrightarrow{\text{a.s.}} c_\lambda$. In addition, $\text{angle}(\hat{u}_j, S_l) = \text{O}_{\text{a.s.}} \left(\left\{ \frac{\delta_l^{(n)}}{\delta_{l-1}^{(n)}} \vee \frac{\delta_{l+1}^{(n)}}{\delta_l^{(n)}} \right\}^{\frac{1}{2}} \right)$,

$j \in H_l$ for $l = 1, \dots, h-1$, and $\text{O}_{\text{a.s.}} \left(\left\{ \frac{\delta_h^{(n)}}{\delta_{h-1}^{(n)}} \right\}^{\frac{1}{2}} \right) \vee \text{O}_{\text{a.s.}} \left(\left\{ \frac{d(n)}{n\delta_h^{(n)}} \right\}^{\frac{1}{2}} \right)$ for $l = h$. Finally, $|\langle \hat{u}_j, u_j \rangle| = \text{O}_{\text{a.s.}} \left(\left\{ \frac{n\lambda_j^{(n)}}{d(n)} \right\}^{\frac{1}{2}} \right), j \in H_l, l = h+1, \dots, r$, and $\text{O}_{\text{a.s.}} \left(\left\{ \frac{n}{d(n)} \right\}^{\frac{1}{2}} \right), j > m$.

(c) If $\frac{d(n)}{n\delta_1^{(n)}} \rightarrow \infty$, then the non-zero $\frac{n\lambda_j}{d(n)} \xrightarrow{\text{a.s.}} c_\lambda$. In addition, $|\langle \hat{u}_j, u_j \rangle| = \text{O}_{\text{a.s.}} \left(\left\{ \frac{n\lambda_j^{(n)}}{d(n)} \right\}^{\frac{1}{2}} \right), j > m$.

The following comments can be made for the results of Theorem 1.

- Note that, for $j \in H_1$, the subspace consistency rate for \hat{u}_j is $\left\{ \frac{\delta_1^{(n)}}{\delta_1^{(n)}} \right\}^{\frac{1}{2}}$. By defining $\delta_0^{(n)} = \infty$, the consistency rate expression $\left\{ \frac{\delta_l^{(n)}}{\delta_{l-1}^{(n)}} \vee \frac{\delta_{l+1}^{(n)}}{\delta_l^{(n)}} \right\}^{\frac{1}{2}}$ remains valid for $l = 1$.

- If $c_\lambda = 0$ in Assumption 3, then that assumption can be rewritten as

$$\delta_1^{*(n)} > \dots > \delta_r^{*(n)} > \lambda_{m+1}^{*(n)} \rightarrow \dots \rightarrow \lambda_{d(n)}^{*(n)} = 1,$$

where $\delta_j^{*(n)} = \frac{\delta_j^{(n)}}{\lambda_{d(n)}^{(n)}}$ and $\lambda_j^{*(n)} = \frac{\lambda_j^{(n)}}{\lambda_{d(n)}^{(n)}}$. We comment that the asymptotic properties of

\hat{u}_j then depend on the rescaled eigenvalues $\lambda_j^{*(n)}$, instead of the raw eigenvalues $\lambda_j^{(n)}$. In particular, with $c_\lambda = 0$, Theorem 1 can be slightly modified by replacing $\delta_j^{(n)}$ with

$$\delta_j^{*(n)}, \frac{n\hat{\lambda}_j^{(n)}}{d(n)} \xrightarrow{\text{a.s.}} c_\lambda \text{ " with " } \frac{n\hat{\lambda}_j^{(n)}}{d(n)\lambda_{d(n)}^{(n)}} \xrightarrow{\text{a.s.}} 1 \text{ " , and the strongly inconsistency rate } \left\{ \frac{n\lambda_{h-1}^{(n)}}{d(n)} \right\}^{\frac{1}{2}}$$

with $\left\{ \frac{n\lambda_j^{*(n)}}{d(n)} \right\}^{\frac{1}{2}}$, respectively.

- In Assumption 3, if there is a big gap between $\delta_r^{(n)}$ and $\lambda_{m+1}^{(n)}$ such that $\delta_r^{(n)} \gg \lambda_{m+1}^{(n)}$, then $\lambda_{m+1}^{(n)} \rightarrow \dots \rightarrow \lambda_{d(n)}^{(n)} \rightarrow c_\lambda$ can be weakened to $\lambda_{m+1}^{(n)} \asymp \dots \asymp \lambda_{d(n)}^{(n)} \asymp 1$. It follows that the consistency results of the first r tiers of sample eigenvalues in Scenario (a) or the first h tiers in Scenario (b) remain the same, while all other results of the form “ $\xrightarrow{\text{a.s.}}$ ” for the sample eigenvalues should be replaced by almost surely “ \asymp ”. The results for the sample eigenvectors remain the same.

- One needs $\lambda_{m+1}^{(n)} \rightarrow \dots \rightarrow \lambda_{d(n)}^{(n)} \rightarrow c_\lambda$, or $\lambda_{m+1}^{(n)} \asymp \dots \asymp \lambda_{d(n)}^{(n)} \asymp 1$, to obtain general convergence results for the non-spike sample eigenvalues $\hat{\lambda}_j$, $j > m$, under the wide range of scenarios: $\frac{d(n)}{n} \rightarrow 0$, $\frac{d(n)}{n} \rightarrow \infty$, or $\lim_{n \rightarrow \infty} \frac{d(n)}{n} = c$ ($0 < c < \infty$). When one focusses only on the spike eigenvalues, a weaker assumption, such as the slowly decaying non-spike eigenvalues assumed by Bai and Yao (2012), is sufficient. Then, the spike condition $\delta_r^{(n)} \gg \lambda_{m+1}^{(n)}$ is enough to generate the consistency properties of $\hat{\lambda}_j$ and \hat{u}_j , $j \leq m$ in Scenario (a). In that case, the behaviors of the other sample eigenvalues and eigenvectors are scenario specific, depending on whether $\frac{d(n)}{n} \rightarrow 0$, $\frac{d(n)}{n} \rightarrow \infty$, or $\lim_{n \rightarrow \infty} \frac{d(n)}{n} = c$ ($0 < c < \infty$).

- The cases covered by Theorem 1 are not studied in Paul (2007), where the eigenvalues are considered to be individually estimable.

- In Theorem 1, the dimension d can be fixed. In addition, suppose $\infty > \delta_1^{(n)} > \dots > \delta_r^{(n)} > \lambda_{m+1}^{(n)} \rightarrow \dots \rightarrow \lambda_{d(n)}^{(n)} \rightarrow c_\lambda$, and the eigenvalues satisfy Assumption 2. Then, the results of Theorem 1(a) are consistent with the classical asymptotic subspace consistency results implied by Theorem 1 of Anderson (1963).

4. Cases with fixed n

This section studies spike models when the sample size n is fixed. Now the eigenvalues are denoted as $\lambda_j^{(d)}$, a sequence indexed by the dimension d . We first report here the theoretical results for spike models with inseparable eigenvalues. The corresponding results for models with distinct eigenvalues are presented in Section C of the supplementary materials.

Theorem 2 summarizes the results for spike models with tiered eigenvalues. In comparison with Jung and Marron (2009), we make more general assumptions on the population eigenvalues, and obtain the convergence rate results; furthermore, we obtain almost sure convergence, instead of convergence in probability.

Assume that as $d \rightarrow \infty$, the first m eigenvalues fall into r tiers, where the eigenvalues in the same tier are asymptotically equivalent, as stated in the following assumption:

Assumption 4 *for fixed n , as $d \rightarrow \infty$, $\lambda_j^{(d)} \asymp \delta_l^{(d)}$, $j \in H_l$, $l = 1, \dots, r$.*

Different from Assumption 2 for diverging sample size n , now with a fixed n , the eigenvalues within the same tier are assumed to be of the same order, rather than of the same limit when n increases to ∞ . As we will see below in Theorem 2, one can no longer separately estimate the eigenvalues of the same order when n is fixed, which is feasible with an increasing n as long as they do not have the same limit as shown in Theorem 1.

In addition, we assume that the population eigenvalues from different tiers are of different orders and dominate the higher-order eigenvalues which are asymptotically equivalent:

Assumption 5 *for fixed n , as $d \rightarrow \infty$, $\delta_1^{(d)} \gg \dots \gg \delta_r^{(d)} \gg \lambda_{m+1}^{(d)} \asymp \dots \asymp \lambda_{d(n)}^{(d)} \asymp 1$.*

Note that for fixed n and $d \rightarrow \infty$, the assumption $\delta_1^{(d)} > \delta_{h+1}^{(d)}$ can not guarantee asymptotic separation of the corresponding sample eigenvalues $\hat{\lambda}_j$ for $j \in H_l$ and $j \in H_{l+1}$. Thus, we need to replace Assumption 3 with Assumption 5 in order to asymptotically separate the first r subgroups of sample eigenvalues.

Before formally stating Theorem 2, we first introduce several notations. Denote $\delta_0^{(d)} = \infty$ for every d , which is used to describe the subspace consistent rates. Consider the z_{kj} in (1), and let

$$\tilde{Z}_j = (z_{1j}, \dots, z_{nj})^T, \quad j = 1, \dots, d. \quad (5)$$

Define

$$K = \lim_{d \rightarrow \infty} \frac{\sum_{j=m+1}^d \lambda_j^{(d)}}{d} \quad \text{and} \quad A_l^* = \frac{1}{n} \sum_{k \in H_l} \tilde{Z}_k \tilde{Z}_k^T, \quad l = 1, \dots, r, \quad (6)$$

which are used to describe the asymptotic properties of the sample eigenvalues.

Theorem 2 *Under Assumptions 1, 4 and 5, for fixed n , as $d \rightarrow \infty$, the following results hold.*

(a) *If $\frac{d}{n} \rightarrow 0$ and $\frac{d}{n} \rightarrow \infty$, where $1 \leq h \leq r$, then for $j \in H_l$, $l = 1, \dots, h$, almost surely*

$$\lambda_{\min}(A_l^*) \times \min_{k \in H_l} \lambda_k^{(d)} \leq \hat{\lambda}_j \leq \lambda_{\max}(A_l^*) \times \max_{k \in H_l} \lambda_k^{(d)}, \quad (7)$$

and the other non-zero $\hat{\lambda}_j$ satisfy $\frac{n\hat{\lambda}_j}{d} \xrightarrow{\text{a.s.}} K$. In addition, $\text{angle}(\hat{u}_j, S_l) = \text{O}_{\text{a.s.}} \left(\left\{ \frac{\delta_l^{(d)}}{\delta_{l-1}^{(d)}} \vee \frac{\delta_{l+1}^{(d)}}{\delta_l^{(d)}} \right\}^{\frac{1}{2}} \right)$,

$j \in H_l$ for $l = 1, \dots, h-1$, and $\text{O}_{\text{a.s.}} \left(\left\{ \frac{d}{\delta_{h-1}^{(d)}} \right\}^{\frac{1}{2}} \right) \vee \text{O}_{\text{a.s.}} \left(\left\{ \frac{d}{\delta_h^{(d)}} \right\}^{\frac{1}{2}} \right)$ for $l = h$. Finally,

$|\langle \hat{u}_j, u_j \rangle| = \text{O}_{\text{a.s.}} \left(\left\{ \frac{\lambda_j^{(d)}}{d} \right\}^{\frac{1}{2}} \right)$, $j \in H_l$, $l = h+1, \dots, r$, and $\text{O}_{\text{a.s.}} \left(\left\{ \frac{1}{d} \right\}^{\frac{1}{2}} \right)$, $j > m$.

the dual matrix Σ_D in Section 7.3. Finally, we derive the asymptotic properties of the sample eigenvectors of Σ in Section 7.4. Some intuitive ideas are provided in the supplement (Shen et al., 2015) to help understanding the proof.

7.2 Lemmas

We list four lemmas that are used in our proof. Lemma 1 studies asymptotic properties of the largest and smallest non-zero eigenvalues of a random matrix.

Lemma 1 Suppose $B = \frac{1}{q}VV^T$ where V is an $p \times q$ random matrix composed of i.i.d. random variables with zero mean, unit variance and finite fourth moment. As $q \rightarrow \infty$ and $\frac{p}{q} \rightarrow c \in [0, \infty)$, the largest and smallest non-zero eigenvalues of B converge almost surely to $(1 + \sqrt{c})^2$ and $(1 - \sqrt{c})^2$, respectively.

Remark 1 Lemma 1 is known as the Bai-Yin's law (Bai and Yin, 1993). As in Remark 1 of Bai and Yin (1993), the smallest non-zero eigenvalue is the $p - q + 1$ smallest eigenvalue of B for $c > 1$.

Lemma 2 is about the Weyl Inequality and the dual Weyl Inequality (Tao, 2010), which appear below as the right-hand-side inequality and the left-hand-side inequality, respectively.

Lemma 2 If A, B are $p \times p$ real symmetric matrices, then for all $j = 1, \dots, p$,

$$\begin{cases} \lambda_j(A) + \lambda_p(B) \\ \lambda_{j+1}(A) + \lambda_{p-1}(B) \\ \vdots \\ \lambda_p(A) + \lambda_j(B) \end{cases} \leq \lambda_j(A+B) \leq \begin{cases} \lambda_j(A) + \lambda_1(B) \\ \lambda_{j-1}(A) + \lambda_2(B) \\ \vdots \\ \lambda_1(A) + \lambda_j(B) \end{cases},$$

where $\lambda_j(\cdot)$ is the j -th largest eigenvalue of the matrix.

Lemma 3 As $n \rightarrow \infty$, the eigenvalues of the matrix A in (9) satisfy

$$\frac{\lambda_j(A)}{\lambda_j^{(n)}} \xrightarrow{\text{a.s.}} 1, \quad \text{for } j = 1, \dots, m.$$

Proof Define the m -dimensional random vectors $X_i^* = [m, 0_{m \times (d-m)}]$, X_i , $i = 1, \dots, n$. Then, X_i^* has mean zero and the following covariance matrix Σ^* :

$$\Sigma^* = \begin{pmatrix} \lambda_1^{(n)} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_m^{(n)} \end{pmatrix}.$$

Let A^* be the dual matrix of the matrix A . The sample covariance matrix of X_i^* is

$$\begin{aligned} A^* &= \frac{1}{n} \sum_{i=1}^n X_i^* X_i^{*T} \\ &= \lambda_1^{(n)} \times \begin{pmatrix} \frac{1}{n} \sum_{i=1}^n z_{i,1}^2 & \cdots & \left\{ \frac{\lambda_k^{(n)}}{\lambda_1^{(n)}} \right\}^{\frac{1}{2}} \frac{1}{n} \sum_{i=1}^n z_{i,1} z_{i,m} \\ \vdots & \ddots & \vdots \\ \left\{ \frac{\lambda_m^{(n)}}{\lambda_1^{(n)}} \right\}^{\frac{1}{2}} \frac{1}{n} \sum_{i=1}^n z_{i,1} z_{i,m} & \cdots & \frac{\lambda_m^{(n)}}{\lambda_1^{(n)}} \frac{1}{n} \sum_{i=1}^n z_{i,m}^2 \end{pmatrix}, \end{aligned} \quad (10)$$

where the $z_{i,j}$'s are defined in (1).

Since A^* is the dual matrix of A , then A and A^* share the same non-zero eigenvalues. Below we study the eigenvalues of A through the dual matrix A^* .

The i.i.d. and unit variance properties of the $z_{i,j}$'s yield that as $n \rightarrow \infty$,

$$\frac{1}{n} \sum_{i=1}^n z_{i,k} z_{i,l} \xrightarrow{\text{a.s.}} \begin{cases} 1 & 1 \leq k = l \leq m \\ 0 & 1 \leq k \neq l \leq m \end{cases}. \quad (11)$$

Denote $b_k = \lim_{n \rightarrow \infty} \frac{\lambda_k^{(n)}}{\lambda_1^{(n)}} \leq 1$, $k = 1, \dots, m$. Then it follows from (10) and (11) that as $n \rightarrow \infty$,

$$\frac{1}{\lambda_1^{(n)}} A^* \xrightarrow{\text{a.s.}} \begin{pmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & b_m \end{pmatrix},$$

which further yields

$$\frac{\lambda_1(A)}{\lambda_1^{(n)}} = \frac{\lambda_1(A^*)}{\lambda_1^{(n)}} \xrightarrow{\text{a.s.}} 1. \quad (12)$$

Similarly, for $k = 2, \dots, m$, we have that as $n \rightarrow \infty$,

$$\frac{\lambda_1\left(\frac{1}{n} \sum_{j=k}^m \lambda_j^{(n)} \tilde{Z}_j \tilde{Z}_j^T\right)}{\lambda_k^{(n)}} \xrightarrow{\text{a.s.}} 1. \quad (13)$$

Next we derive the upper and lower bounds for $\lambda_k(A)$, $k = 2, \dots, m$. According to Lemma 2, we have the following inequality:

$$\lambda_k(A) = \lambda_k\left(\frac{1}{n} \sum_{j=1}^m \lambda_j^{(n)} \tilde{Z}_j \tilde{Z}_j^T\right) \leq \lambda_1\left(\frac{1}{n} \sum_{j=k}^m \lambda_j^{(n)} \tilde{Z}_j \tilde{Z}_j^T\right) + \lambda_k\left(\frac{1}{n} \sum_{j=1}^{k-1} \lambda_j^{(n)} \tilde{Z}_j \tilde{Z}_j^T\right).$$

Since the rank of $\frac{1}{n} \sum_{j=1}^{k-1} \lambda_j^{(n)} \tilde{Z}_j \tilde{Z}_j^T$ is at most $k-1$, then $\lambda_k\left(\frac{1}{n} \sum_{j=1}^{k-1} \lambda_j^{(n)} \tilde{Z}_j \tilde{Z}_j^T\right) = 0$, which together with (13), yields that

$$\frac{\lambda_k(A)}{\lambda_k^{(n)}} \leq \frac{1}{\lambda_k^{(n)}} \times \lambda_1\left(\frac{1}{n} \sum_{j=k}^m \lambda_j^{(n)} \tilde{Z}_j \tilde{Z}_j^T\right). \quad (14)$$

For the lower bound, it follows from Equation (5.9) in Jung and Marron (2009) that

$$\lambda_1\left(\sum_{j=k+1}^n \frac{\lambda_k}{n} \tilde{Z}_k \tilde{Z}_k^T\right) + \lambda_n\left(\frac{1}{n} \sum_{j=k+1}^m \lambda_j \tilde{Z}_j \tilde{Z}_j^T\right) \leq \lambda_k(A). \quad (15)$$

Given that the rank of $\frac{1}{n} \sum_{j=k+1}^m \lambda_j \tilde{Z}_j \tilde{Z}_j^T$ is at most m with $m < n$, then $\lambda_n(\frac{1}{n} \sum_{j=k+1}^m \lambda_j \tilde{Z}_j \tilde{Z}_j^T) = 0$, which together with (15), yields that

$$\frac{\lambda_k(A)}{\lambda_k^{(n)}} \geq \frac{1}{\lambda_k^{(n)}} \times \lambda_1\left(\frac{\lambda_k}{n} \tilde{Z}_k \tilde{Z}_k^T\right). \quad (16)$$

Note that as $n \rightarrow \infty$,

$$\frac{1}{\lambda_k^{(n)}} \times \lambda_1\left(\frac{\lambda_k}{n} \tilde{Z}_k \tilde{Z}_k^T\right) = \frac{1}{n} \tilde{Z}_k^T \tilde{Z}_k \xrightarrow{\text{a.s.}} 1. \quad (17)$$

It follows from (13), (14), (16) and (17) that, for $k = 2, \dots, m$,

$$\frac{\lambda_k(A)}{\lambda_k^{(n)}} \xrightarrow{\text{a.s.}} 1, \quad \text{as } n \rightarrow \infty. \quad (18)$$

The combination of (12) and (18) proves Lemma 6.1. \blacksquare

Lemma 4 Assume that $\lim_{n \rightarrow \infty} \frac{d(n)}{n} = c$, where $0 \leq c \leq \infty$, and let $\lambda_{\max}(\cdot)$ and $\lambda_{\min}(\cdot)$ be the largest and smallest non-zero eigenvalues of the matrix, respectively. As $n \rightarrow \infty$, $\lambda_{\max}(B)$ and $\lambda_{\min}(B)$, where B in (9), satisfy

$$\lambda_{\max}(B) \text{ and } \lambda_{\min}(B) \xrightarrow{\text{a.s.}} c_\lambda, \quad \text{for } c = 0, \quad (19)$$

$$\frac{n}{d(n)} \lambda_{\max}(B) \text{ and } \frac{n}{d(n)} \lambda_{\min}(B) \xrightarrow{\text{a.s.}} c_\lambda, \quad \text{for } c = \infty, \quad (20)$$

and

$$\lambda_{\max}(B) \xrightarrow{\text{a.s.}} c_\lambda(1 + \sqrt{c}^2) \quad \text{and} \quad \lambda_{\min}(B) \xrightarrow{\text{a.s.}} c_\lambda(1 - \sqrt{c}^2), \quad \text{for } 0 < c < \infty. \quad (21)$$

Remark 2 If $\lambda_{m+1}^{(n)} \rightarrow \dots \rightarrow \lambda_{d(n)}^{(n)}$ is relaxed to $\lambda_{m+1}^{(n)} \asymp \dots \asymp \lambda_{d(n)}^{(n)}$, then “ $\xrightarrow{\text{a.s.}}$ ” is replaced by almost surely “ \asymp ”.

Proof Define $B^* = \frac{1}{n} \sum_{j=m+1}^{d(n)} \tilde{Z}_j \tilde{Z}_j^T$. The proof uses the following inequalities for $k \geq 1$:

$$\lambda_{d(n)}^{(n)} \times \lambda_k(B^*) \leq \lambda_k(B) \leq \lambda_{m+1}^{(n)} \times \lambda_k(B^*). \quad (22)$$

We first prove the right inequality of (22). Note that $\lambda_{m+1}^{(n)} B^*$ can be rewritten as $\lambda_{m+1}^{(n)} B + B^*$, where $B_R^* = \frac{1}{n} \sum_{j=m+1}^{d(n)} (\lambda_m^{(n)} - \lambda_j^{(n)}) \tilde{Z}_j \tilde{Z}_j^T$ and is a non-negative matrix. It then follows from Lemma 2 that for $k \geq 1$,

$$\lambda_{m+1}^{(n)} \times \lambda_k(B^*) = \lambda_k(\lambda_{m+1}^{(n)} B^*) \geq \lambda_k(B) + \lambda_n(B_R^*) \geq \lambda_k(B),$$

which yields the right inequality of (22).

For the left inequality in (22), note that $B = \lambda_{d(n)}^{(n)} B^* + B_L^*$, where $B_L^* = \frac{1}{n} \sum_{j=m+1}^{d(n)} (\lambda_j^{(n)} - \lambda_{d(n)}^{(n)}) \tilde{Z}_j \tilde{Z}_j^T$ and is a non-negative matrix. Lemma 2 implies that for $k \geq 1$,

$$\lambda_k(B) \geq \lambda_k(\lambda_{d(n)}^{(n)} B^*) + \lambda_n(B_L^*) \geq \lambda_k(\lambda_{d(n)}^{(n)} B^*) = \lambda_{d(n)}^{(n)} \times \lambda_k(B^*),$$

which yields the left inequality of (22).

Note that B^* can be rewritten as $B^* = \frac{1}{n} V V^T$, where $V = [\tilde{Z}_{m+1}, \dots, \tilde{Z}_{d(n)}]$ is an $n \times (d(n) - m)$ matrix. If $\lim_{n \rightarrow \infty} \frac{d(n)}{n} = \lim_{n \rightarrow \infty} \frac{d(n) - m}{n} = \infty$, then according to Lemma 1, we have that

$$\frac{1}{d(n) - m} \lambda_{\max}(V V^T) \quad \text{and} \quad \frac{1}{d(n) - m} \lambda_{\min}(V V^T) \xrightarrow{\text{a.s.}} 1.$$

It then follows that $\frac{n}{d(n)} \lambda_{\max}(B^*)$ and $\frac{n}{d(n)} \lambda_{\min}(B^*) \xrightarrow{\text{a.s.}} 1$, which, together with (22) and $\lambda_{m+1}^{(n)} \rightarrow \lambda_{d(n)}^{(n)} \rightarrow c_r$, yields (20).

Now consider the case $\lim_{n \rightarrow \infty} \frac{d(n)}{n} = \lim_{n \rightarrow \infty} \frac{d(n) - m}{n} = c < \infty$. Since $B^* = \frac{1}{n} V V^T$ and $\frac{1}{n} V^T V$ share the non-zero eigenvalues, then we study the eigenvalues of B^* through $\frac{1}{n} V^T V$. Applying Lemma 1 to $\frac{1}{n} V^T V$ yields that

$$\lambda_{\max}\left(\frac{1}{n} V^T V\right) \xrightarrow{\text{a.s.}} (1 + \sqrt{c})^2 \quad \text{and} \quad \lambda_{\min}\left(\frac{1}{n} V^T V\right) \xrightarrow{\text{a.s.}} (1 - \sqrt{c})^2.$$

It then follows that $\lambda_{\max}(B^*) \xrightarrow{\text{a.s.}} (1 + \sqrt{c})^2$ and $\lambda_{\min}(B^*) \xrightarrow{\text{a.s.}} (1 - \sqrt{c})^2$. In addition, given that $\lambda_{m+1}^{(n)} \rightarrow \lambda_{d(n)}^{(n)} \rightarrow c_r$ and (22), then we have $\lambda_{\max}(B) \xrightarrow{\text{a.s.}} c_r(1 + \sqrt{c})^2$ and $\lambda_{\min}(B) \xrightarrow{\text{a.s.}} c_r(1 - \sqrt{c})^2$ for $0 \leq c < \infty$, which yields (19) ($c = 0$) and (21) ($0 < c < \infty$). \blacksquare

7.3 Asymptotic properties of the sample eigenvalues

We now study the asymptotic properties of the sample eigenvalues $\hat{\lambda}_j$ for $j = 1, \dots, [n \wedge d(n)]$, which are the same as those of the dual matrix $\tilde{\Sigma}_D$, denoted as $\lambda_j(\tilde{\Sigma}_D) = \lambda_j(A + B)$.

7.3.1 SCENARIO (a) IN THEOREM 1

Scenario (a) contains three different cases: $\lim_{n \rightarrow \infty} \frac{d(n)}{n} = 0$, ∞ , or c ($0 < c < \infty$). The proofs are different for each case and are provided separately below.

Consider the first case: $\lim_{n \rightarrow \infty} \frac{d(n)}{n} = 0$. According to Lemma 2, we have that

$$\frac{\lambda_j(A)}{\lambda_j^{(n)}} \leq \frac{\hat{\lambda}_j}{\lambda_j^{(n)}} \leq \frac{\lambda_j(A)}{\lambda_j^{(n)}} + \frac{\lambda_1(B)}{\lambda_j^{(n)}}. \quad (23)$$

If $\lambda_m^{(n)} \rightarrow \infty$, it follows from (19) that $\frac{\lambda_1(B)}{\lambda_j^{(n)}} \xrightarrow{a.s.} 0$ for $j = 1, \dots, m$. Then the combination of Lemma 3 and (23) proves that, as $n \rightarrow \infty$,

$$\frac{\hat{\lambda}_j}{\lambda_j^{(n)}} \xrightarrow{a.s.} 1, \quad j = 1, \dots, m. \quad (24)$$

If $\lambda_m^{(n)} < \infty$, according to Theorem 1 ($c = 0$) of Baik and Silverman (2006), we still have (24). In addition, according to Lemma 2, we have

$$\lambda_j(B) \leq \hat{\lambda}_j \leq \lambda_j(A) + \lambda_1(B). \quad (25)$$

Since the rank of A is at most m , then $\lambda_j(A) = 0$ for $j \geq m + 1$, which, together with (25), yields that for $j = m + 1, \dots, [n \wedge d(n) - m]$,

$$\lambda_{\min}(B) \leq \hat{\lambda}_j \leq \lambda_{\max}(B). \quad (26)$$

Thus it follows from (19) and (26) that as $n \rightarrow \infty$,

$$\hat{\lambda}_j \xrightarrow{a.s.} c_j, \quad j = m + 1, \dots, [n \wedge d(n) - m].$$

Now consider the second case: $\lim_{n \rightarrow \infty} \frac{d(n)}{n} = \infty$. Since $\frac{d(n)}{n\lambda_m^{(n)}} \rightarrow 0$, then $\lambda_m^{(n)} \rightarrow \infty$, which, together with (20), (23) and Lemma 3, yields (24). Since $\lim_{n \rightarrow \infty} \frac{d(n)}{n} = \infty$, then $[n \wedge d(n) - m] = [n \wedge d(n)] = n$ as $n \rightarrow \infty$. It follows from (20) and (26) that

$$\frac{n}{d(n)} \hat{\lambda}_j \xrightarrow{a.s.} c_j, \quad j = m + 1, \dots, [n \wedge d(n)].$$

Finally, consider the third case: $\lim_{n \rightarrow \infty} \frac{d(n)}{n} = c$ ($0 < c < \infty$). Similarly, it follows from $\frac{d(n)}{n\lambda_m^{(n)}} \rightarrow 0$ that $\lambda_m^{(n)} \rightarrow \infty$, which, jointly with (21), (23) and Lemma 3, yields (24). In addition, note that (21) and (26), and then almost surely we have

$$c_j(1 - \sqrt{c})^2 \leq \underline{\lim}_{n \rightarrow \infty} \hat{\lambda}_j \leq \overline{\lim}_{n \rightarrow \infty} \hat{\lambda}_j \leq c_j(1 + \sqrt{c})^2, \quad j = m + 1, \dots, [n \wedge d(n) - m].$$

All together, we have proven the consistency of the first m sample eigenvalues under Scenario (a), as stated in (24).

7.3.2 SCENARIO (b) IN THEOREM 1

Given $\frac{d(n)}{n\delta_{h+1}^{(n)}} \rightarrow \infty$ and (8), then $\frac{d(n)}{n} \rightarrow \infty$ and $\frac{d(n)}{n\lambda_j^{(n)}} \rightarrow 0$ for $j \in H_l, l = 1, \dots, h$. Thus, according to (20), we have that $\frac{\lambda_1(B)}{\lambda_j^{(n)}} = \left[\frac{\lambda_1(B)}{d(n)} \lambda_1(B) \right] \left[\frac{d(n)}{n\lambda_j^{(n)}} \right] \xrightarrow{a.s.} 0$ for $j \in H_l, l = 1, \dots, h$. Furthermore, it follows from Lemma 3 and (23) that as $n \rightarrow \infty$,

$$\frac{\hat{\lambda}_j}{\lambda_j^{(n)}} \xrightarrow{a.s.} 1, \quad j \in H_l, l = 1, \dots, h. \quad (27)$$

Note that (25) can be rewritten as

$$\frac{n}{d(n)} \lambda_j(B) \leq \frac{n}{d(n)} \hat{\lambda}_j \leq \frac{n}{d(n)} \lambda_j(A) + \frac{n}{d(n)} \lambda_1(B), \quad (28)$$

which yields that for $j = j_h + 1, \dots, [n \wedge d(n) - m]$,

$$\frac{n}{d(n)} \lambda_{\min}(B) \leq \frac{n}{d(n)} \hat{\lambda}_j \leq \frac{n}{d(n)} \lambda_j(A) + \frac{n}{d(n)} \lambda_{\max}(B). \quad (29)$$

Note that for $j = j_h + 1, \dots, [n \wedge d(n) - m]$, we have

$$\frac{n}{d(n)} \lambda_j(A) \leq \frac{n}{d(n)} \lambda_{j_{h+1}}^{(n)}(A) = \left\{ \frac{n\delta_{h+1}^{(n)}}{d(n)} \right\} \left\{ \frac{\lambda_{j_{h+1}}^{(n)}(A)}{\delta_{h+1}^{(n)}} \right\}.$$

It then follows from $\frac{d(n)}{n\delta_{h+1}^{(n)}} \rightarrow \infty$ and Lemma 3 that $\frac{d(n)}{d(n)} \lambda_j(A) \xrightarrow{a.s.} 0$. Since $\frac{d(n)}{n} \rightarrow \infty$, then $[n \wedge d(n) - m] = [n \wedge d(n)] = n$, as $n \rightarrow \infty$. Then it follows from (20) and (29) that as $n \rightarrow \infty$

$$\frac{n}{d(n)} \hat{\lambda}_j \xrightarrow{a.s.} c_j, \quad j = j_h + 1, \dots, [n \wedge d(n)]. \quad (30)$$

The combination of (27) and (30) yields the asymptotic properties of the non-zero sample eigenvalues in Scenario (b).

7.3.3 SCENARIO (c) IN THEOREM 1

Since $\frac{d(n)}{n\delta_{h+1}^{(n)}} \rightarrow \infty$, then $\frac{d(n)}{n} \rightarrow \infty$. According to (28), we have that for $j = 1, \dots, [n \wedge d(n) - m]$,

$$\frac{n}{d(n)} \lambda_{\min}(B) \leq \frac{n}{d(n)} \hat{\lambda}_j \leq \frac{n}{d(n)} \lambda_1(A) + \frac{n}{d(n)} \lambda_{\max}(B). \quad (31)$$

Since $\frac{d(n)}{n\delta_{h+1}^{(n)}} \rightarrow \infty$, it follows from (8) and Lemma 3 that

$$\frac{n}{d(n)} \lambda_1(A) = \left[\frac{n\delta_1^{(n)}}{d(n)} \right] \times \left[\frac{\lambda_1^{(n)}}{\delta_1^{(n)}} \right] \times \left[\frac{\lambda_1(A)}{\lambda_1^{(n)}} \right] \xrightarrow{a.s.} 0.$$

Again note that $[n \wedge d(n) - m] = [n \wedge d(n)] = n$, as $n \rightarrow \infty$. Then it follows from (20) and (31) that

$$\frac{n}{d(n)} \hat{\lambda}_j \xrightarrow{a.s.} c_j, \quad j = 1, \dots, [n \wedge d(n)].$$

7.4 Asymptotic properties of the sample eigenvectors

We first state two results that simplify the proof. As aforementioned, in light of the invariance property of the angle, we choose the population eigenvectors $u_j, j = 1, \dots, d(n)$, as the basis of the d -dimensional space. It then follows that $u_j = e_j$ where the j th component of e_j equals to 1 and all the other components equal to zero. This suggests that

$$|\langle \hat{u}_j, u_j \rangle|^2 = |\langle \hat{u}_j, e_j \rangle|^2 = \hat{u}_{j,j}^2, \quad (32)$$

and for any index set H ,

$$\cos[\text{angle}(\hat{u}_j, \text{span}\{u_k, k \in H\})] = \sum_{k \in H} \hat{u}_{k,j}^2. \quad (33)$$

As a reminder, the population eigenvalues are grouped into $r+1$ tiers and the index set of the eigenvalues in the l th tier H_l is defined in (4). Define

$$\tilde{U}_{k,l} = (\hat{u}_{i,j})_{i \in H_k, j \in H_l}, \quad 1 \leq k, l \leq r+1.$$

Then, the sample eigenvector matrix \tilde{U} can be rewritten as the following:

$$\tilde{U} = [\hat{u}_1, \hat{u}_2, \dots, \hat{u}_{d(n)}] = \begin{pmatrix} \tilde{U}_{1,1} & \tilde{U}_{1,2} & \dots & \tilde{U}_{1,r+1} \\ \tilde{U}_{2,1} & \tilde{U}_{2,2} & \dots & \tilde{U}_{2,r+1} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{U}_{r+1,1} & \tilde{U}_{r+1,2} & \dots & \tilde{U}_{r+1,r+1} \end{pmatrix}.$$

To derive the asymptotic properties of the sample eigenvectors \hat{u}_j , we consider the three scenarios of Theorem 1 separately.

7.4.1 SCENARIO (b) IN THEOREM 1

Under Scenario (b), there exists a constant $h \in [1, r]$, such that $\frac{d(n)}{n\delta_h^{(n)}} \rightarrow 0$ and $\frac{d(n)}{n\delta_{h+1}^{(n)}} \rightarrow \infty$. In order to obtain the the subspace consistency properties in Scenario (b), according to (33), we only need to show that as $n \rightarrow \infty$,

$$\sum_{k \in H_l} \hat{u}_{k,j}^2 = 1 + o_{\text{a.s.}} \left\{ \frac{\delta_l^{(n)}}{\delta_{l-1}^{(n)}} \vee \frac{\delta_{l+1}^{(n)}}{\delta_l^{(n)}} \right\}, \quad j \in H_l, \quad l = 1, \dots, h-1, \quad (34)$$

$$\sum_{k \in H_h} \hat{u}_{k,j}^2 = 1 + o_{\text{a.s.}} \left\{ \frac{\delta_h^{(n)}}{\delta_{h-1}^{(n)}} \vee O_{\text{a.s.}} \left\{ \frac{d(n)}{n\delta_h^{(n)}} \right\} \right\}, \quad j \in H_h, \quad (35)$$

which are respectively equivalent to

$$\sum_{j \in H_l} \sum_{k \in H_l} \hat{u}_{k,j}^2 = |H_l| + o_{\text{a.s.}} \left\{ \frac{\delta_l^{(n)}}{\delta_{l-1}^{(n)}} \vee \frac{\delta_{l+1}^{(n)}}{\delta_l^{(n)}} \right\}, \quad l = 1, \dots, h-1, \quad (36)$$

$$\sum_{j \in H_h} \sum_{k \in H_h} \hat{u}_{k,j}^2 = |H_h| + o_{\text{a.s.}} \left\{ \frac{\delta_h^{(n)}}{\delta_{h-1}^{(n)}} \vee O_{\text{a.s.}} \left\{ \frac{d(n)}{n\delta_h^{(n)}} \right\} \right\}, \quad (37)$$

where $|H_l|$ is the number of elements in H_l and less than m . Since $\sum_{j \in H_l} \sum_{k \in H_l} = \sum_{k \in H_l} \sum_{j \in H_l}$, then in order to obtain (36) and (37), we just need to prove that as $n \rightarrow \infty$,

$$\sum_{j \in H_l} \hat{u}_{k,j}^2 = 1 + o_{\text{a.s.}} \left\{ \frac{\delta_l^{(n)}}{\delta_{l-1}^{(n)}} \vee \frac{\delta_{l+1}^{(n)}}{\delta_l^{(n)}} \right\}, \quad k \in H_l, \quad l = 1, \dots, h-1, \quad (38)$$

$$\sum_{j \in H_h} \hat{u}_{k,j}^2 = 1 + o_{\text{a.s.}} \left\{ \frac{\delta_h^{(n)}}{\delta_{h-1}^{(n)}} \vee O_{\text{a.s.}} \left\{ \frac{d(n)}{n\delta_h^{(n)}} \right\} \right\}, \quad k \in H_h. \quad (39)$$

Therefore the proof of the subspace consistency contains two steps (38) and (39). Here we first prove (39) and then (38).

The third step is to show the strong inconsistency in Scenario (b). Since $\hat{\lambda}_j = 0$ for $j > [n \wedge d(n)]$, then we only need to show the strong inconsistency of \hat{u}_j , $j < [n \wedge d(n)]$. Here we will prove that as $n \rightarrow \infty$,

$$\max_{j_h+1 \leq l \leq [n \wedge d(n)]} \left\{ \frac{d(n)}{n\lambda_j^{(n)}} \hat{u}_{j,j}^2 \right\} = O_{\text{a.s.}}(1). \quad (40)$$

The First Step: Proof of (39). Since

$$\sum_{j \in H_h} \hat{u}_{k,j}^2 = 1 - \sum_{l=1}^{h-1} \sum_{j \in H_l} \hat{u}_{k,j}^2 - \sum_{j=j_h+1}^{d(n)} \hat{u}_{k,j}^2, \quad (41)$$

then in order to obtain (39), we just need to show that as $n \rightarrow \infty$,

$$\sum_{j=j_h+1}^{d(n)} \hat{u}_{k,j}^2 = O_{\text{a.s.}} \left\{ \frac{d(n)}{n\delta_h^{(n)}} \right\}, \quad k \in H_h, \quad (41)$$

$$\sum_{l=1}^{h-1} \sum_{j \in H_l} \hat{u}_{k,j}^2 = O_{\text{a.s.}} \left\{ \frac{\delta_h^{(n)}}{\delta_{h-1}^{(n)}} \right\}, \quad k \in H_h. \quad (42)$$

We first prove (41). Since $\sum_{j=j_h+1}^{d(n)} \hat{u}_{k,j}^2 \leq \sum_{k=1}^{j_h} \sum_{j=j_h+1}^{d(n)} \hat{u}_{k,j}^2$ for $k \in H_h$, then in order to generate (41), we need to show that as $n \rightarrow \infty$,

$$\sum_{k=1}^{j_h} \sum_{j=j_h+1}^{d(n)} \hat{u}_{k,j}^2 = O_{\text{a.s.}} \left\{ \frac{d(n)}{n\delta_h^{(n)}} \right\}. \quad (43)$$

Since $\sum_{k=1}^{d(n)} \hat{u}_{k,j}^2 = \sum_{j=1}^{d(n)} \hat{u}_{k,j}^2 = 1$, then we have

$$\begin{aligned} d(n) - j_l &= \sum_{j=j_l+1}^{d(n)} \sum_{k=1}^{d(n)} \hat{u}_{k,j}^2 = \sum_{k=1}^{j_l} \sum_{j=j_l+1}^{d(n)} \hat{u}_{k,j}^2 + \sum_{k=j_l+1}^{d(n)} \sum_{j=j_l+1}^{d(n)} \hat{u}_{k,j}^2, \\ d(n) - j_l &= \sum_{k=j_l+1}^{d(n)} \sum_{j=1}^{d(n)} \hat{u}_{k,j}^2 = \sum_{k=j_l+1}^{d(n)} \sum_{j=1}^{j_l} \hat{u}_{k,j}^2 + \sum_{k=j_l+1}^{d(n)} \sum_{j=j_l+1}^{d(n)} \hat{u}_{k,j}^2, \end{aligned}$$

which yields

$$\sum_{k=1}^{j_l} \sum_{j=j_l+1}^{d(n)} \hat{u}_{k,j}^2 = \sum_{k=j_l+1}^{d(n)} \sum_{j=1}^{j_l} \hat{u}_{k,j}^2. \quad (44)$$

Let $l = h$ in (44) and then (43) can be obtained through showing

$$\sum_{k=j_h+1}^{d(n)} \sum_{j=1}^{j_h} \hat{u}_{k,j}^2 = O_{\text{a.s.}} \left\{ \frac{d(n)}{n\delta_h^{(n)}} \right\}. \quad (45)$$

Therefore, in order to show (41), we need to prove (45).

Before proving (45), we need some preparation. Denote $S = \Lambda^{-\frac{1}{2}} \hat{U} \hat{\Lambda}^{\frac{1}{2}}$ where \hat{U} is the sample eigenvector matrix and $\hat{\Lambda}$ is the sample eigenvalue matrix defined in (3). Define

$$Z = (Z_1, \dots, Z_m), \quad (46)$$

where Z_l is in (2). It follows from (1), (2) and (3) that $SS^T = \frac{1}{n}ZZ^T$. Since $s_{k,j} = \lambda_k^{(n)-\frac{1}{2}} \lambda_j^{\frac{1}{2}} \hat{u}_{k,j}$, then considering the k -th diagonal entry of the matrices $SS^T = \frac{1}{n}ZZ^T$ on the two sides leads to

$$\frac{1}{\lambda_k^{(n)}} \sum_{j=1}^d \lambda_j \hat{u}_{k,j}^2 = \sum_{j=1}^d s_{k,j}^2 = \frac{1}{n} \sum_{i=1}^n z_{i,k}^2 \quad k = 1, \dots, d(n). \quad (47)$$

In addition, the j -th diagonal entry of $S^T S$ is less than or equal to its largest eigenvalue, i.e. $\lambda_{\max}(S^T S) = \lambda_{\max}(\frac{1}{n}ZZ^T) = \lambda_{\max}(\frac{1}{n}Z^T Z)$, which yields

$$\hat{\lambda}_j \sum_{k=1}^{d(n)} \frac{1}{\lambda_k^{(n)}} \hat{u}_{k,j}^2 = \sum_{k=1}^{d(n)} s_{k,j}^2 \leq \lambda_{\max}(\frac{1}{n}Z^T Z), \quad j = 1, \dots, d(n). \quad (48)$$

According to (48), we have that for $l = 1, \dots, h$,

$$\begin{aligned} \hat{\lambda}_{j_l} \times \frac{1}{\lambda_{m+1}^{(n)}} \times \sum_{j=1}^{j_l} \hat{u}_{k,j}^2 &\leq \sum_{j=1}^{j_l} \hat{\lambda}_j \sum_{k=m+1}^{d(n)} \frac{1}{\lambda_k^{(n)}} \hat{u}_{k,j}^2 \\ &\leq \sum_{j=1}^{j_l} \hat{\lambda}_j \sum_{k=1}^{d(n)} \frac{1}{\lambda_k^{(n)}} \hat{u}_{k,j}^2 \leq j_l \times \lambda_{\max}(\frac{1}{n}Z^T Z), \end{aligned}$$

which yields

$$\sum_{k=m+1}^{d(n)} \hat{u}_{k,j}^2 = \sum_{j=1}^{j_l} \sum_{k=m+1}^{d(n)} \hat{u}_{k,j}^2 \leq j_l \lambda_{m+1}^{(n)} \times \frac{\delta_{(n)}^{(n)}}{\lambda_{j_l}} \times \lambda_{\max}(\frac{1}{d(n)}Z^T Z) \times \frac{d(n)}{n \delta_l^{(n)}}. \quad (49)$$

Since $\frac{d(n)}{n} = \delta_{k+1}^{(n)} \times \frac{d(n)}{n \delta_{k+1}^{(n)}} \rightarrow \infty$, it follows from Lemma 3 that $\lambda_{\max}(\frac{1}{d(n)}Z^T Z) \xrightarrow{a.s} 1$.

According to (8) and (27), $\frac{\delta_{(n)}^{(n)}}{\lambda_{j_l}} = \frac{\delta_{(n)}^{(n)}}{\lambda_{j_l}} \times \frac{\lambda_{j_l}^{(n)}}{\lambda_{j_l}} \xrightarrow{a.s} 1$, $l = 1, \dots, h$. In addition, note that $j_l (< m)$ is finite and $\lambda_{m+1}^{(n)} \rightarrow c_Y$. Thus it follows from (49) that as $n \rightarrow \infty$,

$$\sum_{k=m+1}^{d(n)} \sum_{j=1}^{j_l} \hat{u}_{k,j}^2 = O_{a.s} \left\{ \frac{d(n)}{n \delta_l^{(n)}} \right\}. \quad (50)$$

From (47), we have that for $l = 1, \dots, h$,

$$\begin{aligned} \frac{1}{\lambda_{j_h+1}^{(n)}} \times \hat{\lambda}_{j_l} \sum_{k=j_h+1}^m \hat{u}_{k,j}^2 &\leq \sum_{k=j_h+1}^m \frac{1}{\lambda_k^{(n)}} \sum_{j=1}^{j_l} \lambda_j \hat{u}_{k,j}^2 \\ &\leq \sum_{k=j_h+1}^m \frac{1}{\lambda_k^{(n)}} \sum_{j=1}^{d(n)} \lambda_j \hat{u}_{k,j}^2 = \sum_{k=j_h+1}^m \frac{1}{n} \sum_{i=1}^n z_{i,k}^2, \end{aligned}$$

which yields

$$\sum_{k=j_h+1}^m \sum_{j=1}^{j_l} \hat{u}_{k,j}^2 \leq \frac{\lambda_{(n)}^{(n)}}{\delta_l^{(n)}} \times \frac{\delta_{(n)}^{(n)}}{\lambda_{j_l}} \times \sum_{k=j_h+1}^m \frac{1}{n} \sum_{i=1}^n z_{i,k}^2. \quad (51)$$

Since $\frac{\lambda_{j_h+1}^{(n)}}{\delta_{j_h+1}^{(n)}} \rightarrow 1$ and $\sum_{k=j_h+1}^m \frac{1}{n} \sum_{i=1}^n z_{i,k}^2 \xrightarrow{a.s} m - j_h$, it follows from (51) that as $n \rightarrow \infty$,

$$\sum_{k=j_h+1}^m \sum_{j=1}^{j_l} \hat{u}_{k,j}^2 = O_{a.s} \left\{ \frac{\delta_{j_h+1}^{(n)}}{\delta_l^{(n)}} \right\}. \quad (52)$$

Since $\delta_{h+1}^{(n)} \ll \frac{d(n)}{n}$, it follows from (50) and (52) that as $n \rightarrow \infty$,

$$\sum_{k=j_h+1}^{d(n)} \sum_{j=1}^{j_l} \hat{u}_{k,j}^2 = O_{a.s} \left\{ \frac{d(n)}{n \delta_l^{(n)}} \right\}, \quad l = 1, \dots, h. \quad (53)$$

Letting $l = h$ in (53) results in (45).

Until now we have proven (41). In order to finish the first step proof, we need to show (42). Since $\frac{1}{n} \sum_{i=1}^n z_{i,k}^2 \xrightarrow{a.s} 1$, it follows from (47) that for $k \in H_n$,

$$\frac{1}{\lambda_k^{(n)}} \sum_{l=1}^{h-1} \sum_{j \in H_l} \hat{\lambda}_j \hat{u}_{k,j}^2 + \frac{1}{\lambda_k^{(n)}} \sum_{j \in H_h} \hat{\lambda}_j \hat{u}_{k,j}^2 + \frac{1}{\lambda_k^{(n)}} \sum_{j=j_h+1}^{d(n)} \hat{\lambda}_j \hat{u}_{k,j}^2 = \frac{1}{\lambda_k^{(n)}} \sum_{j=1}^{d(n)} \lambda_j \hat{u}_{k,j}^2 \xrightarrow{a.s} 1. \quad (54)$$

Since

$$\frac{\hat{\lambda}_j}{\lambda_k^{(n)}} \xrightarrow{a.s} \frac{\delta_{(n)}^{(n)}}{\delta_h^{(n)}}, \quad k \in H_n, j \in H_l, \quad (55)$$

and

$$\frac{1}{\lambda_k^{(n)}} \sum_{j=j_h+1}^{d(n)} \hat{\lambda}_j \hat{u}_{k,j}^2 \leq \frac{\hat{\lambda}_{j_h+1}}{\lambda_k^{(n)}} \xrightarrow{a.s} \frac{\delta_{k+1}^{(n)}}{\delta_h^{(n)}} \rightarrow 0,$$

it follows from (54) that for $k \in H_n$,

$$\sum_{l=1}^{h-1} \frac{\delta_l^{(n)}}{\delta_h^{(n)}} \sum_{j \in H_l} \hat{u}_{k,j}^2 + \sum_{j \in H_h} \hat{u}_{k,j}^2 \xrightarrow{a.s} 1. \quad (56)$$

According to (43), we have $\sum_{j=j_h+1}^{d(n)} \hat{u}_{k,j}^2 \leq \sum_{k=1}^{j_h} \sum_{j=j_h+1}^{d(n)} \hat{u}_{k,j}^2 \xrightarrow{a.s} 0$, which together with

$$\sum_{l=1}^{h-1} \sum_{j \in H_l} \hat{u}_{k,j}^2 + \sum_{j \in H_h} \hat{u}_{k,j}^2 + \sum_{j=j_h+1}^{d(n)} \hat{u}_{k,j}^2 = \sum_{j=1}^d \hat{u}_{k,j}^2 = 1,$$

yields that for $k \in H_n$,

$$\sum_{l=1}^{h-1} \sum_{j \in H_l} \hat{u}_{k,j}^2 + \sum_{j \in H_h} \hat{u}_{k,j}^2 \xrightarrow{a.s} 1. \quad (57)$$

Since $\lim_{n \rightarrow \infty} \frac{\delta_h^{(n)}}{\delta_h^{(n)}} > 1$ for $l < h$, it follows from (56) and (57) that $\sum_{j \in H_h} \hat{u}_{k,j}^2 \xrightarrow{\text{a.s.}} 1$ for $k \in H_h$, which together with (56), yields that for $k \in H_h$,

$$\sum_{l=1}^{h-1} \frac{\delta_l^{(n)}}{\delta_h^{(n)}} \sum_{j \in H_l} \hat{u}_{k,j}^2 \xrightarrow{\text{a.s.}} 0. \quad (58)$$

Since $\lim_{n \rightarrow \infty} \frac{\delta_h^{(n)}}{\delta_h^{(n)}} \geq \lim_{n \rightarrow \infty} \frac{\delta_h^{(n)}}{\delta_h^{(n)}}$ for $l \leq h-1$, it follows from (58) that as $n \rightarrow \infty$,

$$\sum_{l=1}^{h-1} \sum_{j \in H_l} \hat{u}_{k,j}^2 = \text{O}_{\text{a.s.}} \left\{ \begin{matrix} \delta_h^{(n)} \\ \delta_{h-1}^{(n)} \end{matrix} \right\}, \quad k \in H_h,$$

which is (42).

The Second Step: Proof of (38). Below we illustrate how one can use (39) to prove (38) for $l = h-1$. Then through a similar procedure, the result for $l = h-1$ in (38) can be used to prove that (38) holds for $l = h-2$, which is then iterated until finishing the proof of (38).

Since

$$\sum_{j \in H_{h-1}} \hat{u}_{k,j}^2 = 1 - \sum_{l=1}^{h-2} \sum_{j \in H_l} \hat{u}_{k,j}^2 - \sum_{j=j_{h-1}+1}^{d(n)} \hat{u}_{k,j}^2,$$

then in order to obtain (38) for $l = h-1$, we need to prove that as $n \rightarrow \infty$,

$$\sum_{j=j_{h-1}+1}^{d(n)} \hat{u}_{k,j}^2 = \text{O}_{\text{a.s.}} \left\{ \begin{matrix} \delta_h^{(n)} \\ \delta_{h-1}^{(n)} \end{matrix} \right\}, \quad k \in H_{h-1}, \quad (59)$$

$$\sum_{l=1}^{h-2} \sum_{j \in H_l} \hat{u}_{k,j}^2 = \text{O}_{\text{a.s.}} \left\{ \begin{matrix} \delta_{h-1}^{(n)} \\ \delta_{h-2}^{(n)} \end{matrix} \right\}, \quad k \in H_{h-1}. \quad (60)$$

Now we show the proof of (59). Since $j_h < m$ is finite and $\sum_{j=1}^{j_h-1} = \sum_{j=1}^{h-1} \sum_{j \in H_l}$, it follows from (42) that as $n \rightarrow \infty$,

$$\sum_{k=j_{h-1}+1}^{j_h} \sum_{j=1}^{j_h-1} \hat{u}_{k,j}^2 = \sum_{k=j_{h-1}+1}^{j_h} \left(\sum_{l=1}^{h-1} \sum_{j \in H_l} \hat{u}_{k,j}^2 \right) = \text{O}_{\text{a.s.}} \left\{ \begin{matrix} \delta_h^{(n)} \\ \delta_{h-1}^{(n)} \end{matrix} \right\}. \quad (61)$$

Let $l = h-1$ in (53) to obtain that $\sum_{k=j_h+1}^{d(n)} \sum_{j=1}^{j_h-1} \hat{u}_{k,j}^2 = \text{O}_{\text{a.s.}} \left\{ \begin{matrix} d(n) \\ n \delta_{h-1}^{(n)} \end{matrix} \right\}$. Since $\delta_h^{(n)} \gg \frac{d(n)}{n}$, it follows from (61) that as $n \rightarrow \infty$,

$$\begin{aligned} \sum_{k=j_{h-1}+1}^{d(n)} \sum_{j=1}^{j_h-1} \hat{u}_{k,j}^2 &= \sum_{k=j_{h-1}+1}^{j_h} \sum_{j=1}^{j_h-1} \hat{u}_{k,j}^2 + \sum_{k=j_h+1}^{d(n)} \sum_{j=1}^{j_h-1} \hat{u}_{k,j}^2 \\ &= \text{O}_{\text{a.s.}} \left\{ \begin{matrix} \delta_h^{(n)} \\ \delta_{h-1}^{(n)} \end{matrix} \right\} + \text{O}_{\text{a.s.}} \left\{ \begin{matrix} d(n) \\ n \delta_{h-1}^{(n)} \end{matrix} \right\} = \text{O}_{\text{a.s.}} \left\{ \begin{matrix} \delta_h^{(n)} \\ \delta_{h-1}^{(n)} \end{matrix} \right\}. \end{aligned} \quad (62)$$

Let $l = h-1$ in (44), which together with (62), proves that as $n \rightarrow \infty$,

$$\sum_{k=1}^{j_{h-1}} \sum_{j=j_{h-1}+1}^{d(n)} \hat{u}_{k,j}^2 = \sum_{k=j_{h-1}+1}^{j_{h-1}} \sum_{j=1}^{j_{h-1}} \hat{u}_{k,j}^2 = \text{O}_{\text{a.s.}} \left\{ \begin{matrix} \delta_h^{(n)} \\ \delta_{h-1}^{(n)} \end{matrix} \right\}. \quad (63)$$

Since $\sum_{j=j_{h-1}+1}^{d(n)} \hat{u}_{k,j}^2 \leq \sum_{k=1}^{j_{h-1}} \sum_{j=j_{h-1}+1}^{d(n)} \hat{u}_{k,j}^2$ for $k \in H_{h-1}$, then (59) follows from (63).

Now we show the proof of (60) to finish the second step. Since $\frac{1}{n} \sum_{i=1}^n z_{i,k}^2 \xrightarrow{\text{a.s.}} 1$, it follows from (47) that for $k \in H_{h-1}$,

$$\frac{1}{\lambda_k^{(n)}} \sum_{l=1}^{h-2} \sum_{j \in H_l} \hat{\lambda}_j \hat{u}_{k,j}^2 + \frac{1}{\lambda_k^{(n)}} \sum_{j \in H_{h-1}} \hat{\lambda}_j \hat{u}_{k,j}^2 + \frac{1}{\lambda_k^{(n)}} \sum_{j=j_{h-1}+1}^{d(n)} \hat{\lambda}_j \hat{u}_{k,j}^2 = \frac{1}{\lambda_k^{(n)}} \sum_{j=j_{h-1}+1}^{d(n)} \hat{\lambda}_j \hat{u}_{k,j}^2 \xrightarrow{\text{a.s.}} 1. \quad (64)$$

Since $\frac{1}{\lambda_k^{(n)}} \sum_{j=j_{h-1}+1}^{d(n)} \hat{\lambda}_j \hat{u}_{k,j}^2 \leq \frac{1}{\lambda_k^{(n)}} \hat{\lambda}_{j_{h-1}+1} \sum_{j=j_{h-1}+1}^{d(n)} \hat{u}_{k,j}^2$ and $\frac{\hat{\lambda}_{j_{h-1}+1}}{\lambda_k^{(n)}} \xrightarrow{\text{a.s.}} \lim_{n \rightarrow \infty} \frac{\delta_h^{(n)}}{\delta_{h-1}^{(n)}} < 1$ for $k \in H_{h-1}$, it follows from (59) that

$$\frac{1}{\lambda_k^{(n)}} \sum_{j=j_{h-1}+1}^{d(n)} \hat{\lambda}_j \hat{u}_{k,j}^2 \xrightarrow{\text{a.s.}} 0,$$

which together with (55) and (64), yields

$$\sum_{l=1}^{h-2} \frac{\delta_l^{(n)}}{\delta_{h-1}^{(n)}} \sum_{j \in H_l} \hat{u}_{k,j}^2 + \sum_{j \in H_{h-1}} \hat{u}_{k,j}^2 \xrightarrow{\text{a.s.}} 1, \quad k \in H_{h-1}. \quad (65)$$

In addition, since

$$\sum_{l=1}^{h-2} \sum_{j \in H_l} \hat{u}_{k,j}^2 + \sum_{j \in H_{h-1}} \hat{u}_{k,j}^2 = \sum_{j=j_{h-1}+1}^{d(n)} \hat{u}_{k,j}^2 = 1,$$

it follows from (59) that

$$\sum_{l=1}^{h-2} \sum_{j \in H_l} \hat{u}_{k,j}^2 + \sum_{j \in H_{h-1}} \hat{u}_{k,j}^2 \xrightarrow{\text{a.s.}} 1, \quad k \in H_{h-1}. \quad (66)$$

Note that $\lim_{n \rightarrow \infty} \frac{\delta_h^{(n)}}{\delta_{h-1}^{(n)}} > 1$ for $l < h-1$. Then the combination of (65) and (66) gives $\sum_{j \in H_{h-1}} \hat{u}_{k,j}^2 \xrightarrow{\text{a.s.}} 1$ for $k \in H_{h-1}$, which together with (65), yields

$$\sum_{l=1}^{h-2} \frac{\delta_l^{(n)}}{\delta_{h-1}^{(n)}} \sum_{j \in H_l} \hat{u}_{k,j}^2 \xrightarrow{\text{a.s.}} 0, \quad k \in H_{h-1}. \quad (67)$$

Since $\lim_{n \rightarrow \infty} \frac{\delta_h^{(n)}}{\delta_{h-1}^{(n)}} \geq \lim_{n \rightarrow \infty} \frac{\delta_{h-2}^{(n)}}{\delta_{h-1}^{(n)}}$ for $l \leq h-2$, it follows from (67) that as $n \rightarrow \infty$,

$$\sum_{l=1}^{h-2} \sum_{j \in H_l} \hat{u}_{k,j}^2 = \text{O}_{\text{a.s.}} \left\{ \begin{matrix} \delta_h^{(n)} \\ \delta_{h-2}^{(n)} \end{matrix} \right\}, \quad k \in H_h,$$

which is (60).

The Third Step: Proof of (40). According to (47), we have $\frac{1}{\lambda_j^{(n)}} \hat{\lambda}_j \hat{u}_{j,j}^2 \leq \frac{1}{n} \sum_{i=1}^n z_{i,j}^2$ for $j = 1, \dots, d(n)$, which yields

$$\begin{aligned} \hat{\lambda}_{j_{n \wedge d(n)}} \times \max_{j_n+1 \leq j \leq n \wedge d(n)} \left\{ \frac{1}{\lambda_j^{(n)}} \hat{u}_{j,j}^2 \right\} &\leq \max_{j_n+1 \leq j \leq n \wedge d(n)} \left\{ \frac{\hat{\lambda}_j}{\lambda_j^{(n)}} \hat{u}_{j,j}^2 \right\} \\ &\leq \max_{1 \leq j \leq n \wedge d(n)} \left\{ \frac{\hat{\lambda}_j}{\lambda_j^{(n)}} \hat{u}_{j,j}^2 \right\} \leq \max_{1 \leq j \leq n \wedge d(n)} \left\{ \frac{1}{n} \sum_{i=1}^n z_{i,j}^2 \right\}. \end{aligned} \quad (68)$$

Select the first $[n \wedge d(n)]$ rows of Z in (46) and denote the resulting random matrix as Z^* . Since $\frac{1}{n} \sum_{i=1}^n z_{i,j}^2$ is the j -th diagonal entry of $\frac{1}{n} Z^* Z^{*T}$ for $1 \leq j \leq [n \wedge d(n)]$, it follows that

$$\frac{1}{n} \sum_{i=1}^n z_{i,j}^2 \leq \lambda_{\max} \left(\frac{1}{n} Z^* Z^{*T} \right), \quad 1 \leq j \leq [n \wedge d(n)],$$

which yields

$$\max_{1 \leq j \leq [n \wedge d(n)]} \left\{ \frac{1}{n} \sum_{i=1}^n z_{i,j}^2 \right\} \leq \lambda_{\max} \left(\frac{1}{n} Z^* Z^{*T} \right).$$

Then from (68),

$$\left\{ \frac{n}{d(n)} \hat{\lambda}_{j_{n \wedge d(n)}} \right\} \times \max_{j_n+1 \leq j \leq n \wedge d(n)} \left\{ \frac{d(n)}{n \lambda_j^{(n)}} \hat{u}_{j,j}^2 \right\} \leq \lambda_{\max} \left(\frac{1}{n} Z^* Z^{*T} \right). \quad (69)$$

Since $\frac{d(n)}{n} \rightarrow \infty$ here, $[n \wedge d(n)] = n$. According to Lemma 1, we have $\lambda_{\max} \left(\frac{1}{n} Z^* Z^{*T} \right) \xrightarrow{\text{a.s.}} 4$, which together with (30) and (69), yields (40).

7.4.2 SCENARIO (a) IN THEOREM 1

Scenario (a) contains three different cases: $\lim_{n \rightarrow \infty} \frac{d(n)}{n} = 0$, ∞ , or c ($0 < c < \infty$). The proofs are slightly different for each case and are provided separately below.

Consider the case $\lim_{n \rightarrow \infty} \frac{d(n)}{n} = \infty$. Since $\lambda_j^{(n)} \rightarrow c_j$ for $j \in H_{r+1}$, then $\frac{d(n)}{n \delta_r^{(n)}} \rightarrow 0$ and $\frac{d(n)}{n \lambda_j^{(n)}} \rightarrow \infty$ for $j \in H_{r+1}$. Thus h in (34) and (35) becomes r such that as $n \rightarrow \infty$,

$$\sum_{k \in H_l} \hat{u}_{k,j}^2 = 1 + \mathbf{O}_{\text{as}} \left\{ \frac{\delta_l^{(n)}}{\delta_{l-1}^{(n)}} \vee \frac{\delta_{l+1}^{(n)}}{\delta_l^{(n)}} \right\}, \quad j \in H_l, \quad l = 1, \dots, r-1, \quad (70)$$

$$\sum_{k \in H_r} \hat{u}_{k,j}^2 = 1 + \mathbf{O}_{\text{as}} \left\{ \frac{\delta_r^{(n)}}{\delta_{r-1}^{(n)}} \right\} \vee \mathbf{O}_{\text{as}} \left\{ \frac{d(n)}{n \delta_r^{(n)}} \right\}, \quad j \in H_r. \quad (71)$$

Since $j_r = m$, then (50) becomes that as $n \rightarrow \infty$,

$$\sum_{k=m+1, j=1}^{d(n)} \hat{u}_{k,j}^2 = \mathbf{O}_{\text{as}} \left\{ \frac{d(n)}{n \delta_r^{(n)}} \right\},$$

which together with (44), yields that

$$\sum_{k=1, j=m+1}^{d(n)} \hat{u}_{k,j}^2 = \mathbf{O}_{\text{as}} \left\{ \frac{d(n)}{n \delta_r^{(n)}} \right\}. \quad (72)$$

Since

$$1 \geq \sum_{k \in H_{r+1}} \hat{u}_{k,j}^2 = 1 - \sum_{k=1}^m \hat{u}_{k,j}^2 \geq 1 - \sum_{k=1, j=m+1}^m \hat{u}_{k,j}^2, \quad j > m, \quad (73)$$

it follows from (72) that

$$\sum_{k \in H_{r+1}} \hat{u}_{k,j}^2 = 1 + \mathbf{O}_{\text{as}} \left\{ \frac{d(n)}{n \delta_r^{(n)}} \right\}, \quad j = m+1, \dots, [n \wedge d(n)]. \quad (74)$$

Now consider the second case $\lim_{n \rightarrow \infty} \frac{d(n)}{n} = c$ ($0 < c < \infty$). Note that the subspace consistency of the sample eigenvectors in (70) only depends on the asymptotic properties of the sample eigenvalues $\hat{\lambda}_j$, $j = 1, \dots, m$. According to Section 7.3.1, the asymptotic properties of $\hat{\lambda}_j$, $j = 1, \dots, m$, only depends on $\frac{d(n)}{n \delta_r^{(n)}} \rightarrow 0$, and is the same as in the first case $\lim_{n \rightarrow \infty} \frac{d(n)}{n} = \infty$. Thus (70) remains valid here.

However, the subspace consistency of the other eigenvectors also depends on $\hat{\lambda}_j$, $j > m$, whose properties are different from the first case. In fact (71) and (74) respectively become that as $n \rightarrow \infty$,

$$\sum_{k \in H_r} \hat{u}_{k,j}^2 = 1 + \mathbf{O}_{\text{as}} \left\{ \frac{\delta_r^{(n)}}{\delta_{r-1}^{(n)}} \right\} \vee \mathbf{O}_{\text{as}} \left\{ \frac{1}{\delta_r^{(n)}} \right\}, \quad j \in H_r, \quad (75)$$

$$\sum_{k \in H_{r+1}} \hat{u}_{k,j}^2 = 1 + \mathbf{O}_{\text{as}} \left\{ \frac{1}{\delta_r^{(n)}} \right\}, \quad j = m+1, \dots, [n \wedge d(n)]. \quad (76)$$

In order to obtain (75), following the first step proof procedure in Section 7.4.1, we only need to show that as $n \rightarrow \infty$,

$$\sum_{k=1, j=m+1}^m \hat{u}_{k,j}^2 = \mathbf{O}_{\text{as}} \left\{ \frac{1}{\delta_r^{(n)}} \right\}. \quad (77)$$

Since $\lim_{n \rightarrow \infty} \frac{d(n)}{n} = c$ ($0 < c < \infty$), then (72) becomes (77). In addition, it follows from (73) and (77) that (76) is established.

Note that we can combine the first and the second cases together as follows. If $\frac{d(n)}{n} \rightarrow c$, $0 < c \leq \infty$, then the combination of (71) and (75) provides

$$\sum_{k \in H_r} \hat{u}_{k,j}^2 = 1 + \mathbf{O}_{\text{as}} \left\{ \frac{\delta_r^{(n)}}{\delta_{r-1}^{(n)}} \right\} \vee \mathbf{O}_{\text{as}} \left\{ \frac{d(n)}{n \delta_r^{(n)}} \right\}, \quad j \in H_r,$$

and the combination of (74) and (77) yields

$$\sum_{k \in H_{r+1}} \hat{u}_{k,j}^2 = 1 + \mathbf{O}_{\text{a.s.}} \left\{ \frac{d(n)}{n\delta_r} \right\}, \quad j = m+1, \dots, [n \wedge d(n)].$$

In addition, (70) remains valid for both cases. Thus it follows from (33) that we have finished the proof of the second bullet point in Scenario (a).

Finally, consider the last case $\lim_{n \rightarrow \infty} \frac{d(n)}{n} = 0$. It is clear that (70) still holds. According to (33), in order to finish the proof of the first bullet point in Scenario (a), we only need to show that as $n \rightarrow \infty$,

$$\sum_{k \in H_r} \hat{u}_{k,j}^2 = 1 + \mathbf{O}_{\text{a.s.}} \left\{ \frac{\delta_r^{(n)}}{\delta_{r-1}^{(n)}} \vee \frac{1}{\delta_r^{(n)}} \right\}, \quad j \in H_r, \quad (78)$$

$$\sum_{k \in H_{r+1}} \hat{u}_{k,j}^2 = 1 + \mathbf{O}_{\text{a.s.}} \left\{ \frac{1}{\delta_r^{(n)}} \right\}, \quad j = m+1, \dots, [n \wedge d(n)]. \quad (79)$$

In fact, in order to prove (78) and (79), we need to replace (77) by that as $n \rightarrow \infty$,

$$\sum_{k=1}^m \sum_{j=m+1}^{d(n)} \hat{u}_{k,j}^2 = \mathbf{O}_{\text{a.s.}} \left\{ \frac{1}{\delta_r^{(n)}} \right\}. \quad (80)$$

It follows from (44) that

$$\sum_{k=1}^m \sum_{j=m+1}^{d(n)} \hat{u}_{k,j}^2 = \sum_{j=1}^m \sum_{k=m+1}^{d(n)} \hat{u}_{k,j}^2.$$

We also have

$$\sum_{j=1}^m \sum_{k=m+1}^{d(n)} \hat{u}_{k,j}^2 = \sum_{l=1}^r \sum_{j \in H_l} \sum_{k=m+1}^{d(n)} \hat{u}_{k,j}^2.$$

Then in order to obtain (80), we only need to prove that as $n \rightarrow \infty$,

$$\sum_{j \in H_r} \sum_{k=m+1}^{d(n)} \hat{u}_{k,j}^2 = \mathbf{O}_{\text{a.s.}} \left\{ \frac{1}{\delta_r^{(n)}} \right\}, \quad (81)$$

$$\sum_{l=1}^{r-1} \sum_{j \in H_l} \sum_{k=m+1}^{d(n)} \hat{u}_{k,j}^2 = \mathbf{O}_{\text{a.s.}} \left\{ \frac{1}{\delta_r^{(n)}} \right\}. \quad (82)$$

We now prove (81). Since the j -th diagonal entry of $S^T S$ is between its largest and smallest eigenvalue, then (48) becomes

$$\lambda_{\min} \left(\frac{1}{n} Z Z^T \right) \leq \hat{\lambda}_j \sum_{k=1}^{d(n)} \frac{1}{\lambda_k^{(n)}} \hat{u}_{k,j}^2 = \sum_{k=1}^{d(n)} s_{k,j}^2 \leq \lambda_{\max} \left(\frac{1}{n} Z Z^T \right), \quad j = 1, \dots, d(n). \quad (83)$$

Since $\lim_{n \rightarrow \infty} \frac{d(n)}{n} = 0$, it follows from Lemma 1 that $\lambda_{\min} \left(\frac{1}{n} Z Z^T \right)$ and $\lambda_{\max} \left(\frac{1}{n} Z Z^T \right) \xrightarrow{\text{a.s.}} 1$. In addition, since $\frac{\hat{\lambda}_j}{\lambda_j^{(n)}} \xrightarrow{\text{a.s.}} 1$ for $j = 1, \dots, m$ (Section 7.3.1), it follows from (83) that

$$\sum_{k=1}^{d(n)} \lambda_j^{(n)} \frac{1}{\lambda_k^{(n)}} \hat{u}_{k,j}^2 \xrightarrow{\text{a.s.}} 1, \quad j = 1, \dots, m. \quad (84)$$

Note that

$$\sum_{k=1}^{d(n)} \lambda_j^{(n)} \frac{1}{\lambda_k^{(n)}} \hat{u}_{k,j}^2 = \sum_{l=1}^{r-1} \sum_{k \in H_l} \lambda_j^{(n)} \frac{1}{\lambda_k^{(n)}} \hat{u}_{k,j}^2 + \sum_{k \in H_r} \lambda_j^{(n)} \frac{1}{\lambda_k^{(n)}} \hat{u}_{k,j}^2. \quad (85)$$

According to (70), we have that

$$\sum_{j \in H_l, k \in H_l} \hat{u}_{k,j}^2 \xrightarrow{\text{a.s.}} |H_l|, \quad l = 1, \dots, r-1,$$

which leads to

$$\sum_{l=1}^{r-1} |H_l| = \sum_{j=1}^{d(n)} \sum_{l=1}^{r-1} \sum_{k \in H_l} \hat{u}_{k,j}^2 \geq \sum_{l=1}^{r-1} \sum_{j \in H_l^*} \sum_{k \in H_l} \hat{u}_{k,j}^2 \geq \sum_{l=1}^{r-1} \sum_{j \in H_l, k \in H_l} \sum_{l=1}^{r-1} |H_l|.$$

Then it follows that

$$\sum_{j \in H_r} \sum_{l=1}^{r-1} \sum_{k \in H_l} \hat{u}_{k,j}^2 \leq \sum_{j=1}^{d(n)} \sum_{l=1}^{r-1} \sum_{k \in H_l} \hat{u}_{k,j}^2 - \sum_{l=1}^{r-1} \sum_{j \in H_l^*} \sum_{k \in H_l} \hat{u}_{k,j}^2 \xrightarrow{\text{a.s.}} 0. \quad (86)$$

According to (86), we have that

$$\sum_{l=1}^{r-1} \sum_{k \in H_l} \lambda_j^{(n)} \frac{1}{\lambda_k^{(n)}} \hat{u}_{k,j}^2 \leq \sum_{l=1}^{r-1} \sum_{k \in H_l} \hat{u}_{k,j}^2 \leq \sum_{j \in H_r} \sum_{l=1}^{r-1} \sum_{k \in H_l} \hat{u}_{k,j}^2 \xrightarrow{\text{a.s.}} 0, \quad j \in H_r. \quad (87)$$

Since $\frac{\lambda_j^{(n)}}{\lambda_k^{(n)}} \rightarrow 1$ for $k, j \in H_r$, it follows from (85) and (87) that

$$\sum_{k \in H_r} \hat{u}_{k,j}^2 + \sum_{k=m+1}^{d(n)} \lambda_j^{(n)} \frac{1}{\lambda_k^{(n)}} \hat{u}_{k,j}^2 \xrightarrow{\text{a.s.}} 1, \quad j \in H_r. \quad (88)$$

According to (86), we have that

$$\begin{aligned} 1 &\geq \sum_{k \in H_r} \hat{u}_{k,j}^2 + \sum_{k=m+1}^{d(n)} \hat{u}_{k,j}^2 = 1 - \sum_{l=1}^{r-1} \sum_{k \in H_l} \hat{u}_{k,j}^2 \\ &\geq 1 - \sum_{j \in H_r} \sum_{l=1}^{r-1} \sum_{k \in H_l} \hat{u}_{k,j}^2 \xrightarrow{\text{a.s.}} 1, \quad j \in H_r. \end{aligned}$$

Then it follows that

$$\sum_{k \in H_r} \hat{u}_{k,j}^{(n)2} + \sum_{k=m+1}^{d(n)} \hat{u}_{k,j}^{(n)2} \xrightarrow{\text{a.s.}} 1, \quad j \in H_r. \quad (89)$$

Since $\lim_{n \rightarrow \infty} \frac{\lambda_j^{(n)}}{\lambda_k^{(n)}} > 1$ for $j \in H_r$ and $k \geq m+1$, then combining (88) and (89) gives

$$\sum_{k \in H_r} \hat{u}_{k,j}^{(n)2} \xrightarrow{\text{a.s.}} 1, \quad j \in H_r,$$

which together with (88), yields

$$\sum_{k=m+1}^{d(n)} \lambda_j^{(n)} \frac{1}{\lambda_k^{(n)}} \hat{u}_{k,j}^{(n)2} \xrightarrow{\text{a.s.}} 0, \quad j \in H_r. \quad (90)$$

Since $\lambda_k^{(n)} \rightarrow c_k$ for $k \geq m+1$ and $\frac{\lambda_j^{(n)}}{c_k} \rightarrow 1$ for $j \in H_r$, it follows from (90) that as $n \rightarrow \infty$,

$$\sum_{j \in H_r} \sum_{k=m+1}^{d(n)} \hat{u}_{k,j}^{(n)2} = O_{\text{a.s.}} \left\{ \frac{1}{\delta_l^{(n)}} \right\},$$

which is (81).

We now show the proof of (82). According to (84), we have that for $j \in H_l$, $l = 1, \dots, r-1$,

$$\sum_{k \in H_l} \lambda_j^{(n)} \frac{1}{\lambda_k^{(n)}} \hat{u}_{k,j}^{(n)2} + \sum_{k=m+1}^{d(n)} \lambda_j^{(n)} \frac{1}{\lambda_k^{(n)}} \hat{u}_{k,j}^{(n)2} \leq \sum_{k=1}^{d(n)} \lambda_j^{(n)} \frac{1}{\lambda_k^{(n)}} \hat{u}_{k,j}^{(n)2} \xrightarrow{\text{a.s.}} 1. \quad (91)$$

Since $\lambda_j^{(n)} \frac{1}{\lambda_k^{(n)}} \rightarrow 1$ for $k, j \in H_l$, it follows from (70) that

$$\sum_{k \in H_l} \lambda_j^{(n)} \frac{1}{\lambda_k^{(n)}} \hat{u}_{k,j}^{(n)2} \rightarrow \sum_{k \in H_l} \hat{u}_{k,j}^{(n)2} \xrightarrow{\text{a.s.}} 1, \quad k, j \in H_l,$$

which together with (91), yields

$$\sum_{k=m+1}^{d(n)} \lambda_j^{(n)} \frac{1}{\lambda_k^{(n)}} \hat{u}_{k,j}^{(n)2} \xrightarrow{\text{a.s.}} 0, \quad j \in H_l, \quad l = 1, \dots, r-1. \quad (92)$$

Since $\lambda_k^{(n)} \rightarrow c_k$ for $k \geq m+1$ and $\frac{\lambda_j^{(n)}}{c_l} \rightarrow 1$ for $j \in H_l$, it follows from (92) that as $n \rightarrow \infty$,

$$\sum_{j \in H_l} \sum_{k=m+1}^{d(n)} \hat{u}_{k,j}^{(n)2} = O_{\text{a.s.}} \left\{ \frac{1}{\delta_l^{(n)}} \right\}, \quad l = 1, \dots, r-1. \quad (93)$$

Since $\delta_l^{(n)} \leq \delta_l^{(n)}$ for $l = 1, \dots, r-1$, (82) then follows from (93).

7.4.3 SCENARIO (c) IN THEOREM 1

Finally, for Scenario (c) where $\frac{d(n)}{n \delta_1^{(n)}} \rightarrow \infty$, h in (40) equals to 0. Since $j_0 = 0$, then (40) becomes that as $n \rightarrow \infty$,

$$\max_{1 \leq j \leq \lfloor n \nu d(n) \rfloor} \left\{ \frac{d(n)}{n \lambda_j} \hat{u}_{j,j}^{(n)2} \right\} = O_{\text{a.s.}}(1),$$

which yields the strong inconsistency of the sample eigenvectors in Scenario (c).

References

- J. Ahn, J. S. Marron, K. M. Muller, and Y. Y. Chi. The high-dimension, low-sample-size geometric representation holds under mild conditions. *Biometrika*, 94(3):760–766, 2007.
- T. W. Anderson. Asymptotic theory for principal component analysis. *The Annals of Mathematical Statistics*, 34(1):122–148, 1963.
- T. W. Anderson. *An introduction to multivariate statistical analysis*. John Wiley & Sons, New York, 1984.
- Z. D. Bai and J. F. Yao. On sample eigenvalues in a generalized spiked population model. *Journal of Multivariate Analysis*, 106:167–177, 2012.
- Z. D. Bai and Y. Q. Yin. Limit of the smallest eigenvalue of a large dimensional sample covariance matrix. *The Annals of Probability*, pages 1275–1294, 1993.
- J. Baik and J. W. Silverman. Eigenvalues of large sample covariance matrices of spiked population models. *Journal of Multivariate Analysis*, 97(6):1382–1408, 2006.
- J. Baik, G. Ben Arous, and S. P\'ech\'e. Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices. *The Annals of Probability*, 33(5):1643–1697, 2005.
- F. Benaych-Georges and R. R. Nadakuditi. The eigenvalues and eigenvectors of finite, low rank perturbations of large random matrices. *Advances in Mathematics*, 227(1):494–521, 2011.
- M. Biehl and A. Mietzner. Statistical mechanics of unsupervised structure recognition. *Journal of Physics A: Mathematical and General*, 27:1885–1897, 1994.
- G. Casella and J. T. Hwang. Limit expressions for the risk of James-Stein estimators. *Canadian Journal of Statistics*, 10(4):305–309, 1982.
- J. Fan, Y. Liao, and M. Mincheva. Large covariance estimation by thresholding principal orthogonal complements. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(4):603–680, 2013.
- M. A. Girshtick. On the sampling theory of roots of determinantal equations. *The Annals of Mathematical Statistics*, 10(3):203–224, 1939.

- P. Hall, J. S. Marron, and A. Neeman. Geometric representation of high dimension, low sample size data. *Journal of the Royal Statistical Society: Series B*, 67(3):427–444, 2005.
- D. C. Hoyle and M. Rattray. PCA learning for sparse high-dimensional data. *Europhysics Letters*, 62(1):117–123, 2003.
- J.E. Jackson. *A user's guide to principal components*. John Wiley & Sons, New York, 1991. ISBN 0471622672.
- I. M. Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *The Annals of Statistics*, 29(2):295–327, 2001.
- I. M. Johnstone and A. Y. Lu. On consistency and sparsity for principal components analysis in high dimensions. *Journal of the American Statistical Association*, 104(486):682–693, 2009.
- S. Jung and J. S. Marron. PCA consistency in high dimension, low sample size context. *The Annals of Statistics*, 37(6B):4104–4130, 2009.
- S. Jung, A. Sen, and J. S. Marron. Boundary behavior in high dimension, low sample size asymptotics of pca. *Journal of Multivariate Analysis*, 109:190–203, 2012.
- V. Koltchinskii and K. Lounici. Normal approximation and concentration of spectral projectors of sample covariance. *arXiv:1504.07333*, 2015.
- V. Koltchinskii and K. Lounici. Asymptotics and concentration bounds for bilinear forms of spectral projectors of sample covariance. *Annals of Institute Henri Poincaré, to appear*, 2016.
- D. N. Lawley. Tests of significance for the latent roots of covariance and correlation matrices. *Biometrika*, 43(1):128–136, 1956.
- S. Lee, F. Zou, and F. A. Wright. Convergence and prediction of principal component scores in high-dimensional settings. *The Annals of Statistics*, 38(6):3605–3629, 2010.
- Z. M. Ma. Sparse principal component analysis and iterative thresholding. *The Annals of Statistics*, 41(2):772–801, 2013.
- B. Nadler. Finite sample approximation results for principal component analysis: A matrix perturbation approach. *The Annals of Statistics*, 36(6):2791–2817, 2008. ISSN 0090-5364.
- A. Onatski. Asymptotics of the principal components estimator of large factor models with weakly influential factors. *Journal of Econometrics*, 168(2):244–258, 2012.
- D. Paul. Asymptotics of sample eigenstructure for a large dimensional spiked covariance model. *Statistica Sinica*, 17(4):1617–1642, 2007.
- D. Paul and I. Johnstone. Augmented sparse principal component analysis for high dimensional data. *Technical Report, UC Davis*, 2012.
- P. Reimann, C. Broeck, and G. J. Bex. A Gaussian scenario for unsupervised learning. *Journal of Physics A: Mathematical and General*, 29:3521–3535, 1996.
- D. Shen, H. P. Shen, and J. S. Marron. Consistency of sparse PCA in high dimension, low sample size contexts. *Journal of Multivariate Analysis*, 115:317–333, 2013.
- D. Shen, H. P. Shen, and J. S. Marron. A general framework for consistency of principal component analysis: Supplement materials. *Available online at <http://www.unc.edu/~dshen/PCA/PCASupplement.pdf>*, 2015.
- T. Tao. 254A, Notes 3a: Eigenvalues and sums of hermitian matrices. *Available online at <https://terrytao.wordpress.com/2010/01/12/254a-notes-3a-eigenvalues-and-sums-of-hermitian-matrices/>*, 2010.
- TLH Watkin and J-P Nadal. Optimal unsupervised learning. *Journal of Physics A: Mathematical and General*, 27(6):1899–1915, 1994.
- H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006. ISSN 1061-8600.

Conditional Independencies under the Algorithmic Independence of Conditionals.

Jan Lemeire

*Vrije Universiteit Brussel, INDI Dept, ETRO Dept, Pleinlaan 2, B-1050 Brussels, Brussels, Belgium
iMinds, Dept. of Multimedia Technologies, Gaston Crommenlaan 8, B-9050 Ghent, Belgium*

JAN.LEMEIRE@VUB.AC.BE

Editor: Isabelle Guyon and Alexander Statnikov

Abstract

In this paper we analyze the relationship between faithfulness and the more recent condition of algorithmic Independence of Conditionals (IC) with respect to the Conditional Independencies (CIs) they allow. Both conditions have been extensively used for causal inference by refuting factorizations for which the condition does not hold. Violation of faithfulness happens when there are CIs that do not follow from the Markov condition. For those CIs, non-trivial constraints among some parameters of the Conditional Probability Distributions (CPDs) must hold. When such a constraint is defined over parameters of different CPDs, we prove that IC is also violated unless the parameters have a simple description. To understand which non-Markovian CIs are permitted we define a new condition closely related to IC: the Independence from Product Constraints (IPC). The condition reflects that CIs might be the result of specific parameterizations of individual CPDs but not from constraints on parameters of different CPDs. In that sense it is more restrictive than IC: parameters may have a simple description. On the other hand, IC also excludes other forms of algorithmic dependencies between CPDs. Finally, we prove that on top of the CIs permitted by the Markov condition (faithfulness), IPC allows non-minimality, deterministic relations and what we called proportional CPDs. These are the only cases in which a CI follows from a specific parameterization of a single CPD.

Keywords: faithfulness, causality, independence of conditionals, Kolmogorov complexity

1. Introduction

Algorithmic Independence of Conditionals (IC) has been put forward for causal inference and its relation with Faithfulness (FF) was analyzed by Lemeire and Janzing (2013). We showed that both conditions often lead to the same causal conclusions and are motivated by similar grounds: that the CPDs are independently chosen. But we argued that IC is more fundamental than FF: we can trust IC more whenever the conclusions are different from those of FF. Moreover, IC goes beyond FF. IC has led to successful causal inferences in cases that FF cannot decide on the causal orientations, see for instance Janzing and Schölkopf (2010); Janzing and Steudel (2010); Daniusis et al. (2010); Janzing et al. (2012); Chen et al. (2014). In this paper we establish the link between FF and IC regarding the Conditional Independencies (CIs) both conditions admit.

Faithfulness is based on the observed variables entailed by a system. Faithfulness assumes that all CIs come from the system's causal structure, described by a Directed Acyclic Graph (DAG), and hold for all parameterizations of the DAG. These CIs are defined by the Markov condition applied on the DAG and can be identified with the d -separation criterion. Lemeire and Janzing (2013) have shown that some CIs are rejected by causal faithfulness but have to be accepted

by the IC condition. This is true for deterministic relations for example, since a deterministically related variable becomes independent from all other variables when conditioned on its determiner, also those that are not d -separated.

FF is motivated by the Lebesgue measure zero argument, saying that if the system's parameters were randomly chosen, the probability of having a configuration following a specific constraint has Lebesgue measure zero (Meek, 1995). In the case of deterministic relations, all but one probability is zero for each input state. This is very unlikely to occur by chance, hence receives Lebesgue measure zero. IC follows a different reasoning. Its justification is based on Solomonoff's Universal Prior (Solomonoff, 1964) which assigns non-zero probability to those points in parameter space that have a finite description (Lemeire and Janzing, 2013). Points reflecting some regularity (allowing compression of the description) will receive a high probability. The Universal Prior favors simple CPDs, and therefore respects Occam's razor. The reasoning is that patterns or regularities are likely to occur; patterns have to be expected. Which is not the case if parameters are randomly chosen according to for instance a uniform distribution. On the other hand, IC follows FF by excluding 'non-generic' parameter configurations. IC assumes that CPDs correspond to independent mechanisms which were 'chosen' independently. While FF excludes all non-trivial constraints among some parameters of CPDs, IC will in general exclude CIs following from specific parameter matches between *different* conditionals. The latter will be expressed formally by the novel condition Independence from Product Constraints (IPC). The condition is introduced to analyze the relation between IC and FF. In this paper we analyze the relation between the 3 conditions.

We first recall the definitions. Then we discuss CIs that do not follow from the Markov condition and we introduce the new IPC criterion. Then, we analyze when CIs violate the IC condition. Section 5 establishes the link between IC/IPC and faithfulness. Next we prove the link between IC and IPC. Before concluding, we discuss the practical implications of these results.

2. Definitions

A Bayesian network consists of a Directed Acyclic Graph (DAG) and a set of Conditional Probability Distributions (CPDs) defined over variables X_1, \dots, X_n such that the joint probability distribution equals the following factorization:

$$P(X_1, \dots, X_n) = \prod_i P(X_i | \text{Parents}(X_i)) \quad (1)$$

with $\text{Parents}(X_i)$ the parent nodes of node X_i in the DAG. A Bayesian network is edge-minimal (MIN) in the sense that no edge can be removed from the DAG without violating the correctness of the factorization.

In a causal model represented by a Bayesian network, all edges correspond to direct causal relations and each CPD corresponds to an independent and autonomous mechanism of the system (Hausman and Woodward, 1999; Lemeire et al., 2011). Throughout the paper we assume that the CPD of each node X_j is described by a parameter vector $\lambda_j \in \mathbb{R}^{b_j}$. Although the finite dimension of the parameter space restricts the conditionals for continuous variables already, we believe that this is appropriate because inference from finite data requires strong assumptions or approximations anyway.

Conditional Independence (CI) is defined as

$$U \perp\!\!\!\perp W \mid Y \Leftrightarrow \forall v \in \mathcal{V}_{dom}, w \in \mathcal{W}_{dom} : P(U \mid v, w) = P(U \mid v) \text{ whenever } P(v, w) > 0. \quad (2)$$

where \mathcal{X}_{dom} is the domain of variable X . Single random variables are denoted by capital letters and sets of variables by boldface capital letters. Values of variables are denoted by lowercase letters. Note that the conditional distribution $P(U \mid v, w)$ is only defined in points where $P(v, w) > 0$.

The Markov condition gives all conditional independencies following from the above factorization (Hausman and Woodward, 1999, p. 532): Every variable is conditionally independent of its non-descendants (except for itself), given its parents. These Markovian independencies hold for all parameterizations of the CPDs. These independencies can be identified graphically by d -separation. A path¹ is said to be blocked by Z if it contains a collider $\rightarrow \cdot \leftarrow$ whose descendants are not in Z or a non-collider $\rightarrow \cdot \rightarrow$ or $\leftarrow \cdot \leftarrow$ that is in Z . X and Y are d -separated by Z if every path between X and Y is blocked by Z . d -separation is denoted by the ternary operator $\cdot \perp \cdot \mid \cdot$:

$$X \perp Y \mid Z.$$

A Bayesian network is said to be *faithful* if the Markovian CIs are the only independencies present in the joint probability distribution; in other words, there are no CIs not following from Markov. We call them *non-Markovian CIs*. Where the Markovian CIs occur for every parameterization of the CPDs, non-Markovian CIs only occur for specific parameterizations of the model. As we will see, specific parameter constraints should be met.

Next we provide the definition of the IC condition. It is based on Kolmogorov complexity or algorithmic information of a binary string s , denoted by $K(s)$: For a binary string $s \in \{0, 1\}^*$ the *algorithmic information* $K(s)$ (or ‘Kolmogorov complexity’) is defined as the length of the shortest program on a universal prefix-free Turing machine that generates s and then stops (Solomonoff, 1960; Kolmogorov, 1965; Chaitin, 1966, 1975). Prefix-free means that the program has to be given with respect to an encoding where no allowed program code is the prefix of another one. Thus, the program does not require an extra symbol indicating its end. Based on Kolmogorov complexity we can define algorithmic independence.

Definition 1 (Algorithmic Independence) Binary strings $s_1 \dots s_n$ are algorithmically independent if

$$K(s_1, \dots, s_n) \stackrel{\pm}{=} \sum_k^n K(s_k). \quad (3)$$

Note that here and throughout the paper we consider the number n of strings as a constant. Accordingly, in the following the number n of nodes will also be considered as a constant.

As usual in algorithmic information theory, \pm denotes equality up to a constant that is independent of the string s , but does depend on the Turing machine. For fixed strings, we have to interpret \pm in the sense of ‘equality up to a small number’ without further specifying what ‘small’ means. This arbitrariness in setting a threshold is similar to the freedom of choosing the significance level in a statistical dependence test.

1. A path is a set of consecutive edges (independent of the direction) that do not visit a vertex more than once.

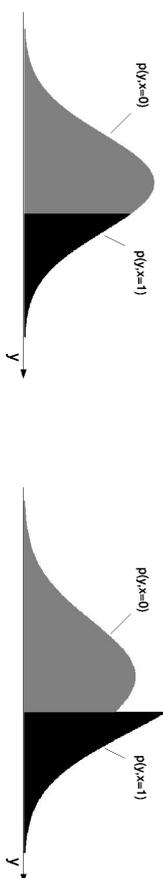


Figure 1: Binary variable X is determined by Gaussian variable Y by a thresholding mechanism, i.e., $X = 1$ for all $y > y_0$, and $X = 0$ otherwise. This is shown on the left. The causal hypothesis $Y \rightarrow X$ is plausible: the conditional $P(X|Y)$ corresponds to setting $X = 1$ for all Y above a certain threshold. On the other hand, $X \rightarrow Y$ is rejected by IC because $P(Y|X)$ and $P(X)$ share algorithmic information: given $P(Y|X)$, only specific choices of $P(X)$ reproduce the Gaussian $P(Y)$, whereas generic choices of $P(X)$ would yield ‘odd’ densities of the type on the right. The figures are taken from (Janzing et al., 2009).

Assumption 1 (CPDs have finite description length) We assume that each parameter vector $\lambda_j \in \mathbb{R}^{k_j}$ has finite description length. To be precise, there is a program that computes the l th component of λ_j up to the precision of d digits if it gets the input (i, j) .² Then, $K(\lambda_j)$ denotes the length of the shortest program of this type.

Now we are ready to define the IC condition (Lemeire and Janzing, 2013):

Definition 2 (Independence of Conditionals)

The conditional probability densities CPD_1, \dots, CPD_n corresponding to a DAG G with n nodes are said to satisfy the Algorithmic Independence of Conditionals, or Independence of Conditionals (IC) for short, if the corresponding parameter vectors λ_j satisfy

$$K(\lambda_1, \dots, \lambda_n) \stackrel{\pm}{=} \sum_{j=1}^n K(\lambda_j), \quad (4)$$

The uncomputability of Kolmogorov complexity hinders the applicability of these concepts. Applications rely on some approximative measure of algorithmic complexity or, as in the following example, on an approximative measure of ‘correlation’ between two distributions.

An example of the usage of the IC condition for causal inference is given by Fig. 1 (Janzing et al., 2009). Consider that Y causes X : $Y \rightarrow X$. Let Y be a Gaussian variable with zero mean and standard deviation 1 (i.e., described by a zero-dimensional parameter space). Let X be a binary variable deterministically determined by Y by a thresholding mechanism, i.e., $X = 1$ for all $y > y_0$ where $y_0 \in \mathbb{R}$ is some threshold, and $X = 0$ otherwise. Here, $P(X|Y)$ is described by 1 parameter, namely y_0 . We now describe the joint distribution $P(X, Y)$ in the wrong causal direction, i.e. with $P(X)$ and $P(Y|X)$. We observe that the set of possible $P(X)$ is not restricted, i.e., we have the one-dimensional parameter $\theta_1 = P(X = 0)$. The set of possible conditionals $P(Y|X)$ obtained by

2. Since the input consists of two strings and the output of k_j strings plus extra information specifying the position of the comma, we use some canonical bijection between $\{0, 1\}^*$ and $\{0, 1, \cdot\}^*$ for appropriate d for input and output.

the above model class is also determined by a one-dimensional parameter $\theta_2 = y_0$ that determines the cutoff. We then observe that θ_1 and θ_2 are related by

$$\theta_1 = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\theta_2} e^{-y^2/2} dy = \text{erf}(\theta_2).$$

Hence, IC is violated whenever $K(\theta_1) \not\geq K(\text{erf})$:

$$\begin{aligned} K(\lambda_X) + K(\lambda_Y) &\geq K(\theta_1) + K(\theta_2) \\ &> K(\text{erf}) + K(\theta_2) \\ &\pm K(\theta_1|\theta_2) + K(\theta_2) \\ &\geq K(\theta_1, \theta_2) \\ &\pm K(\lambda_X + \lambda_Y) \end{aligned}$$

Note that IC is not violated for $y_0 = 0$: then $K(\theta_1) = K(1/2) \pm 0$ and $K(\theta_2) = K(0) \pm 0$.

3. Non-Markovian conditional independencies

We start the investigation by analyzing the CIs that do not follow from the Markov condition. Non-trivial polynomial constraints must be satisfied for non-Markovian conditional independencies. This is shown for discrete Bayesian networks by Meek (1995) and the linear case for distributions over continuous variables by Spirtes et al. (1993).

For a given DAG G and an independence, we define the **Independence Parameter Subspace** as the set of all parameterizations λ of a DAG G for which the independence holds. For Markovian CIs this is the complete space $\mathcal{S} := \times_{j=1}^n \mathcal{S}_j$, where \mathcal{S}_j is the set of possible parameter vectors λ_j . For non-Markovian CIs this is a subspace of \mathcal{S} .

As already pointed out in the introduction and Lemeire and Janzing (2013), deterministic relations between some variables may induce conditional independencies that do not follow from the Markov condition. For $Y \perp\!\!\!\perp X$ in the example $X \rightarrow Y \rightarrow Z$, a sufficient condition is the function $Y = f(X)$. The independence parameter subspace for $Y \perp\!\!\!\perp Z|X$ can be represented by Fig. 2(a) in the case of deterministic relation $Y = f(X)$. If $\lambda_Y \subset \mathcal{R}_Y$ where \mathcal{R}_Y represents all functions, then we have the conditional independence in $P(X, Y, Z)$. λ_Y denotes the parameter vector of the CPD of variable Y , λ_Z that of Z , and so on.

As another example, consider the DAG in Fig. 3 and consider the case where A and D are independent because the influence via B compensates for the influence via C . Assume that all CPDs are given by linear structure equations:

$$\begin{aligned} B &= \alpha A + U_B, \\ C &= \beta A + U_C, \\ D &= \gamma B + \delta C + U_D, \end{aligned} \tag{5}$$

where U_B, U_C and U_D are unobserved disturbances or 'noise' terms that are jointly statistically independent and independent of A . Then the two influences of A on D cancel for

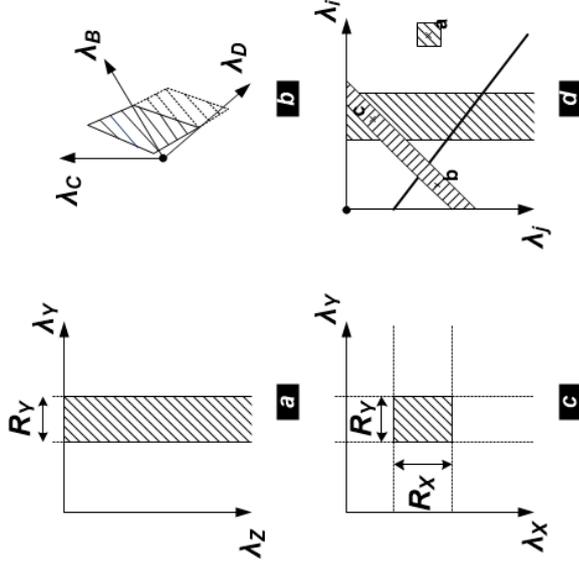


Figure 2: Independence parameter subspaces where each axis represents the possible parameter vectors of a single CPD. In (d) points c and a are permitted by IPC, while b is not.

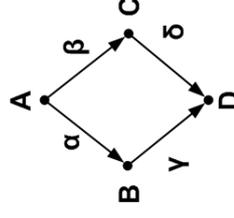


Figure 3: Causal model with linear influences described by parameters $\alpha, \beta, \gamma, \delta$.

such that $A \perp\!\!\!\perp D$. Obviously, FF rejects the causal DAG of Fig. 3 because A and D are not d -separated by the empty set and thus should not be independent. The independence parameter subspace for $A \perp\!\!\!\perp D$ can be represented by Fig. 2(b). Note that γ and δ are represented together by dimension λ_D .

As a third example, consider the causal model $W \rightarrow X \rightarrow Y \rightarrow Z$ with $X = g(W)$ and $Y = f(X)$. It follows that $Y \perp\!\!\!\perp Z|W$. Fig. 2(c) represents the parameter subspace for this CI: both λ_X and λ_Y are restricted for the CI to hold. But both restrictions do not depend on each other: the parameter subspace can be described by the product $\mathcal{R}_X \times \mathcal{R}_Y$. The forthcoming IPC criterion will reject the independence following from Eq. 5 but accepts the last example because the latter parameter subspace is a so-called product subspace.

To formalize the analysis of the independence parameter subspaces, we need the following postulate based on the results of (Meek, 1995) and (Spirtes et al., 1993).

Postulate 1 *There exists a complexity measure $C(\cdot)$ on polynomial equations such that for a given DAG the presence of any conditional independence can be identified by a unique minimal set of undecomposable polynomial constraints on the parameterization of the DAG. With ‘undecomposable’ we mean that a constraint c cannot be written as $(c_1$ or $c_2)$ (one of both constraints should be true) with $C(c) \geq C(c_1) + C(c_2)$. By ‘minimal’ we mean that there is no smaller such set.*

Note that Kolmogorov complexity is not appropriate as complexity measure since it attributes 0 complexity to polynomial functions with simple coefficients. A measure such as AIC or BIC is more appropriate to capture the complexities of polynomial equations. However, it would lead us too far to prove for a complexity measure that it leads to a unique decomposition for non-Markovian CIs. Hence the postulate.

The postulate implies that the parameter subspace of a CI can be described in a unique, ‘canonical’ way as a union of areas, where each of the areas reflects 1 basic polynomial constraint. Fig. 2(d) shows a general example of a parameter subspace. The subspace can be decomposed into 4 basic areas which cannot be further decomposed without increasing its descriptive complexity.

Some areas can be described by a product of parameter constraints and some areas can’t. The latter means that the CI does not follow from a product constraint on the parameters of different CPDs, but from a constraint that is defined over different CPDs. Those parameterizations will be referred: parameter configuration b in Fig. 2(d) is refuted while a and c are permitted. This is expressed by the Independence from Product Constraints (IPC) condition: a parameterization will be rejected if it gives rise to a CI which is part of an area in the parameter subspace which cannot be described by a product of subspaces.

Definition 3 (Independence from Product Constraints (IPC)) *Let a Bayesian network be described by the parameters $\lambda := (\lambda_1, \dots, \lambda_n)$. Then it is said to satisfy the Independence from Product Constraints Condition if for every independence that holds true for λ the parameterization satisfies a constraint of the minimal set of undecomposable constraints of the independence which is a product constraint. A product constraint is a constraint c that can be written as $(c_1$ and $c_2 \dots$ and $c_l)$ where each c_j is a constraint on exactly one λ_j .*

IC, however, will only reject non-product constraints if it results in a compression of the parameters. This is investigated in the next section.

4. Conditional independencies resulting in violations of IC

Although the IPC condition reflects the principle that ‘non-generic’ relations among CPDs are rejected, not all non-Markovian CIs are rejected by IC. To analyze this in detail, we have to recall the following theorem on the violation of IC (Lemeire and Janzing, 2013, Theorem 3):

Theorem 4 *For a given DAG G , let the set of possible CPDs $P(X_j|Parents(X_j))$ be parameterized by some parameter set $\lambda_j := \{\lambda_{j_1}^1, \dots, \lambda_{j_{k_j}}^{k_j}\}$ of parameters. Assume that the parameter values for some specific choice CPD_1, \dots, CPD_n of conditional probability densities satisfy a functional relation in the sense that $\theta_1 = f(\theta_2, \dots, \theta_k)$, where f is some function and $\theta_1, \dots, \theta_k$ are parameters taken from at least two different sets λ_j . Assume furthermore that θ_1 corresponds to CPD_1 (without loss of generality). Then the following condition implies violation of IC:*

$$K(f) \stackrel{\Delta}{\leq} K(\theta_1 | CPD_1^{(\theta_1, *)}), \quad (6)$$

where $CPD_1^{(\theta_1, *)}$ denotes the parameters of CPD_1 without θ_1 (recall that the asterisk denotes the shortest compression).

The theorem states that a constraint results in a violation of IC provided that the parameters are sufficiently complex compared to the complexity of the constraint.

Applied on the example of Fig. 3, the constraint defined by Eq. 5 leads to a violation of IC if α, β, γ and δ have complex values. Describing the JPD by separate descriptions of $P(A)$, $P(B|A)$, $P(C|A)$ and $P(D|B, C)$ is redundant because the parameter γ in $P(D|B, C)$ can be computed from the parameters of the other CPDs via Eq. 5. The constraint is an unlikely coincidence if all real-valued parameters are chosen independently (according to some continuous distribution on \mathbb{R}). Next, consider causal structure $X \rightarrow Y \leftarrow Z$ over binary variables X , Y and Z . $P(Z)$ is parameterized with $P(Z=0) = \alpha$ and $P(Z=1) = 1 - \alpha$, and $P(Y|X, Z)$ with 4 parameters:

$$\begin{array}{c|cc} P(Y=0|X, Z) & Z=0 & Z=1 \\ \hline X=0 & a & b \\ X=1 & c & d \end{array}$$

Then, non-Markovian independence $X \perp\!\!\!\perp Y$ holds when

$$\begin{aligned} P(Y=0|X=0) &= P(Y=0|X=1) \\ \Leftrightarrow P(Z=0).P(Y=0|X=0, Z=0) + P(Z=1).P(Y=0|X=0, Z=1) \\ &= P(Z=0).P(Y=0|X=1, Z=0) + P(Z=1).P(Y=0|X=1, Z=1) \\ \Leftrightarrow \alpha.a + (1-\alpha).b &= \alpha.c + (1-\alpha).d. \end{aligned}$$

Note that this constraint does not depend on the parameterization of $P(X)$. This equation can be rewritten in the following constraint between the parameters of $P(Z)$ and $P(Y|X, Z)$ with T a constant:

$$\frac{\alpha}{1-\alpha} = \frac{d-b}{a-c} = T \quad (7)$$

This equation holds for the following particular parameterization:

- $P(Z=0) = P(Z=1) = 0.5$
- $P(Y|X, Z)$ is a noisy exclusive or (with real-valued $E \in]0, 1[$ representing the noise):

$$\frac{P(Y|X, Z)}{\begin{array}{c|c} X=0 & E \\ X=1 & 1-E \end{array}} \Bigg| \begin{array}{c|c} Z=0 & Z=1 \\ \hline E & 1-E \end{array} = 1$$

It can easily be verified that Eq. 7 holds for all values of E and that $P(Y=1|X=x) = P(Y=1) = 0.5$ for all values of X . Although in this example the parameters of both CPDs are tightened by the constraint, one can hardly say that the parameter of one CPD helps the description of some parameter of the other CPD. $T = 1$ and therefore simple. Also α has constant complexity. Even if E is complex and $K(a)$ therefore too, $K(a|b, c, d)$ has low complexity. Eq. 6 of Theorem 4 does not hold: IC is not violated. It is only violated for complex values of T . Then α will have high Kolmogorov complexity but its description length has constant complexity with the help of a, b, c and d .

Concluding, factorizations having non-Markovian CIs will only be rejected for complex parameter values. The rationale is that we consider simple parameter settings to appear with a much greater probability than when they would be taken randomly from a uniform distribution over the parameter space. We prefer the Universal Prior (Solomonoff, 1964): the probability of a parameter configuration is high for simple parameters, where simple is defined by their Kolmogorov complexity. The probability decreases for increasing parameter complexities. As such we will refute only constraints leading to a compression. Simple parameters will occur and as such make it highly probable that a coincidental CI occurs. In our case, the uniform distribution for $P(Z)$ and xor configuration of $P(Y|X, Z)$ are simple and can be expected. As such, the CI can be expected and must not be excluded.

Finally, note that if a parameter is incompressible with respect to the other CPD parameters, then $K(\theta_1) = K(\theta_1|CPD_1^{\theta_1, \theta_2})$ and the constraint of Theorem 4 becomes simply:

$$K(f) \stackrel{+}{\leq} K(\theta_1). \quad (8)$$

As we will see, the polynomial constraints for non-Markovian CIs have simple coefficients which make that the functions have constant complexity. Thus, complex parameters lead to violations of IC and can be detected by an appropriate approximative measure of Kolmogorov complexity.

5. The IPC and IC conditions and Faithfulness.

Now we infer our most important result: the relation between IC and Faithfulness. We first establish the link between IPC and faithfulness, and then we show that the results also apply for IC when the parameters are sufficiently complex.

Faithfulness implies that variables that are d -connected are (conditionally) dependent: there are unblocked paths. We will prove that under IPC they yield dependence apart from 3 cases permitted by IPC. The theorem is constructed by decomposing the unblocked paths and proving dependency for each component. The basic components of the paths are adjacent variables and v -structures. These are considered first in 2 lemmas together with a lemma on conditioning variables that are not

on one of the paths. All proofs are given in the appendix. We prove it for discrete variables. We believe that a similar approach can be used to prove it for continuous variables.

For using IPC we need the following postulate:

Postulate 2 *The non-trivial polynomial constraints among different CPDs responsible for non-Markovian CIs cannot be decomposed into product constraints without increasing their complexity, i.e. $C(c) < C(c_1) + \dots + C(c_n)$ for any decomposition of non-product constraint c into product constraints c_1, \dots and c_n .*

Since the constraints apply on parameters of different CPDs it seems reasonable to believe that the constraints cannot be decomposed into constraints on individual parameters without increasing the descriptive complexity.

We first define formally what we mean by excluding deterministic relations.

Definition 5 (INDET) *A Bayesian network is said to satisfy INDET if there is no variable X_j that can be written as*

$$X_j = f(X_1, \dots, X_{j-1}, X_{j+1}, \dots, X_n).$$

The first lemma shows that 2 adjacent variables are dependent under IPC and edge-minimality (MIN).

Lemma 6 *Given a Bayesian network defined over discrete variables satisfying MIN and IPC, any two adjacent variables X and Y , where X is a parent of Y , are dependent conditioned on any subset of the other parents of Y .*

The next lemma states that conditionally dependent variables X and Y cannot get conditionally independent via extending the conditioning set by variables that are not on any path between X and Y unless deterministic variables are present.

Lemma 7 *Assume that MIN and IPC holds for some Bayesian network defined over discrete variables. Let $X_{\setminus \{X, Y\}} | \mathbf{Z}'$ where X, Y are arbitrary nodes and \mathbf{Z}' is a set of nodes. Let U denote the set of all variables on the non-blocked paths between X and Y , including X and Y . Then, for every set $\mathbf{Z}'' \subset \bar{\mathbf{Z}} \cup U$, when*

$$X \perp\!\!\!\perp Y | \mathbf{Z}', \mathbf{Z}'',$$

holds, there are deterministic relationships.

In the third lemma we deal with the case in which X and Y are connected via a v -structure $X \rightarrow Z \leftarrow Y$ and one conditions on Z or one of its descendants, such that X and Y are d -connected. It appears that there is a special parameterization in which X and Y are conditionally independent without violating IPC, which we call a *proportional conditional probability distribution*.

Definition 8 (proportional conditional probability distribution (pCPD))

Let Z, Y be variables and \mathbf{X} be a set of variables. The CPD $P(Z|Y, \mathbf{X})$ is said to have a proportional conditional probability distribution if Y can only attain two values y_1 and y_2 and we have

$$\frac{P(Z | y_1, \mathbf{x})}{P(Z | y_2, \mathbf{x})} = \alpha \quad \forall \mathbf{x} \in \mathbf{X}_{dom} \quad (9)$$

with α a constant only depending on Z and \mathbf{X}_{dom} , the domain of X .

The probability distribution of Table 1 shows an example of a PCPD for distributions over discrete variables, for which $\frac{P(z=1|y=1,x_1)}{P(z=1|y=0,x_1)} = 1.5$ and $X \perp\!\!\!\perp Y \mid Z$ holds, which is a non-Markovian CI.

X	$Y = 0$	$Y = 1$
0	0.3	0.45
1	0.6	0.9
2	0.4	0.6

Table 1: Conditional Probability Table of $P(z = 1 \mid X, Y)$ for which Eq. 9 holds and therefore $X \perp\!\!\!\perp Y \mid Z$.

An example for continuous variables in which Eq. 9 is satisfied is the following class of models

$$p(z|y, x_1, \dots, x_n) = e^{-\alpha(z)y - \sum_j \gamma_j(z)x_j},$$

where $c(z)$ and $\gamma_j(z)$ are arbitrary functions.

pCPDs imply non-Markovian CIs in v-structures as shown by the following lemma.

Lemma 9 *If MIN, INDET, IPC holds for a factorization of a joint probability distribution defined over discrete variables, then for any v-structure $X \rightarrow Z \leftarrow Y$ in the DAG and for all \mathbf{W} not containing X, Y or Z :*

$$\begin{aligned} X \perp\!\!\!\perp Y \mid Z, \mathbf{W} &\Leftrightarrow X \perp\!\!\!\perp Y \text{ and } P(Z \mid X, Y) \text{ is a proportional CPD} \\ \forall U \text{ descendants of } Z : X \&\&Y \mid U, \mathbf{W} \end{aligned}$$

The absence of pCPDs is denoted as the NOPCPD condition.

It can easily be verified that violation of MIN or INDET or the presence of a pCPD are permitted by IPC. These 3 cases define constraints on single CPDs. We have to exclude them on top of IPC for achieving faithfulness. The following theorem expresses the relation between IPC and faithfulness:

Theorem 10 *If for a given factorization of a JPD the conditions IPC, MIN, INDET and NOPCPD are met, then faithfulness holds:*

$$\forall \text{ disjoint subsets } X, Y, Z \subset V : X \perp\!\!\!\perp Y \mid Z \Leftrightarrow X \perp\!\!\!\perp Y \mid Z$$

with V the set of all variables under consideration.

In other words, there are only 3 cases in which a non-Markovian CI comes from a specific parameterization of a single CPD: deterministic relationships, non-minimality and proportional CPDs.

To identify the relation between FF and IC we have to define what we mean by sufficiently complex parameters.

Definition 11 (COMPLEX) *A parameterization of a Bayesian network is said to satisfy COMPLEX if for all parameters θ_i of all parameter vectors λ_j :*

$$K(\theta_i | CPD_j^{\setminus \theta_i, *}) \not\geq 0,$$

In the 3 previous lemmas and Theorem 10, the IPC condition can be replaced with IC and COMPLEX.

Theorem 12 *If for a given factorization of a JPD defined over discrete variables the conditions IC, COMPLEX, MIN, INDET and NOPCPD are met, then faithfulness holds:*

$$\forall \text{ disjoint subsets } X, Y, Z \subset V : X \perp\!\!\!\perp Y \mid Z \Leftrightarrow X \perp\!\!\!\perp Y \mid Z$$

with V the set of all variables under consideration.

6. The relation between IC and IPC.

IC and IPC partially overlap. IC rules out all atypical constraints on parameters between different CPDs. IPC only rules out the constraints leading to conditional independencies, while IC allows conditional independencies when matches are to be expected because of low complexity of the parameters.

The theorem on the relation between IC and IPC is based on the fact that non-trivial polynomial constraint must be satisfied for non-Markovian conditional independencies.

Theorem 13 *Given P a distribution over n variables and a DAG G with CPDs parameterized by $\theta_1, \dots, \theta_n$ describing a factorization of P .*

If the parameter vectors satisfy IPC, there exists a parameterization $\theta'_1, \dots, \theta'_n$ of the CPDs which has the same independencies as P and for which IC holds.

*For discrete Bayesian networks and the linear case for continuous distributions, if the factorization satisfies IC and COMPLEX (the parameters θ'_i of the parameter vectors $\theta_1, \dots, \theta_n$ have non-constant complexity: $K(\theta'_i | CPD_i^{\setminus \theta'_i, *}) \not\geq 0$), then IPC holds as well.*

The proof is given in the appendix. We believe that it is provable that under IPC, most parameterizations satisfy IC as well. The proof, however, need some quite technical details to be worked out.

7. Practical application

Despite the uncomputability of Kolmogorov complexity, several novel approaches to causal inference are based on the notion of IC, see for instance Janzing and Schölkopf (2010); Janzing and Stendel (2010); Danusis et al. (2010); Janzing et al. (2012); Chen et al. (2014). As more algorithms will pop up, a philosophical and theoretical underpinning is important. This paper tries to contribute to this endeavour.

Although we advocate that IC is more fundamental than FF (Lemeire and Janzing, 2013), independence-based learning remains a powerful approach. However, we believe that non-Markovian CIs should be taken into account and deserve an in-depth study. Accordingly, independence-based learning algorithms have been adopted in the past to incorporate the presence of deterministic relationships (Lemeire et al., 2012) and pCPDs (violations of orientation faithfulness) (Ramsey et al., 2006).

The theoretical results of this paper might help to understand the nature of the CIs. We showed that the appearance of CIs happens at different levels:

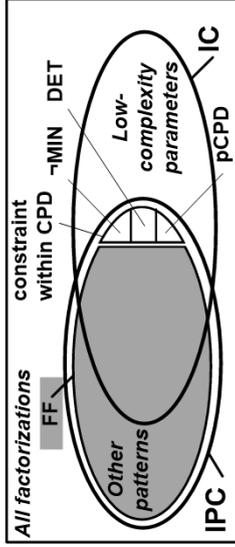


Figure 4: Relation between FF, the IC and IPC condition. It shows the factorizations for which FF, IC and/or IPC holds. The area in which FF holds is shown in gray.

1. some conditional independencies arise from the causal structure (given by the Markov condition),
2. some CIs arise from a specific parameterization of a single CPD,
3. some CIs arise from specific simple parameterizations of different CPDs,
4. some CIs arise from specific complex parameterizations of different CPDs.

Only level 1 is permitted by faithfulness. IPC also permits level 2, while the IC condition allows level 3 as well. Note that some examples of level 4 where given in Lemeire and Janzing (2013) (so-called metamechanisms).

8. Conclusions

Both Faithfulness (FF) and Independence of Conditionals (IC) express a condition on a factorization. They have been extensively used for learning a system’s causal structure from observational data. Non-causal factorizations are refuted when one of both conditions is violated. In this paper we investigated more deeply the relation between both conditions.

Although IC and FF sound like completely different inference principles, the common idea is to reject causal structures for which the CPDs satisfy ‘non-generic’ relations. The relation between the conditions is shown by Fig. 4. The sets represent the factorizations for which the conditions hold. We defined the Independence from Product Constraints (IPC) which allows non-Markovian CIs following from specific parameterizations of individual CPDs but not from constraints among parameterizations of different CPDs. We proved that those CIs come from non-minimality (\neg MIN), deterministic relationships (DET) or so-called proportional CPDs (pCPDs). They are allowed by IPC.

In contrast to IPC, IC is not violated for constraints on low-complexity parameters, since such constraints can be expected to appear by chance. As such, IC can only be applied for sufficiently complex parameters, e.g. when sufficient data is present. As IC goes beyond CIs, IC provides a way to select the causal structure in the Markov equivalence class (the set of DAGs having the same CIs) based on other patterns among the parameters of different CPDs. This has been shown by several recently developed algorithms.

Acknowledgments

I have to thank Dominik Janzing from the MPI for Intelligent Systems of Tübingen, Germany for his insightful comments and discussion. He came up with the IPC condition. He shows me true science. I am also grateful to the anonymous reviewers whose comments helped me to improve the quality of the paper.

Appendix A. Appendix: Proofs

Theorem 12, which is proven here, starts with a factorization and states that if 4 conditions are met, the DAG corresponding to the factorization is faithful to the joint probability distribution. We denote CPD_X as the CPD for X , given its parents, i.e., $P(X | Parents(X))$. To prove faithfulness we will write the down the left-hand side of Eq. 2 as function of the factors of the factorization. The IPC condition rules out independencies following from non-trivial constraints that relate parameters that correspond to different CPDs.

Recall that **conditional independence** is defined as

$$U \perp\!\!\!\perp W | V \Leftrightarrow \forall v \in V_{dom}, w \in W_{dom} : P(U | v, w) = P(U | v) \text{ whenever } P(v, w) > 0.$$

Note that the conditional distribution is only defined in points where $P(v, w) > 0$. Since the right hand side is independent of w , the left hand side must give the same result for all values of W . Hence:

$$U \perp\!\!\!\perp W | V \Leftrightarrow P(U | w_1, V) = P(U | w_2, V) \forall w_1, w_2 \in W_{dom} \quad (10)$$

This is the main equation that will be used throughout the proofs. Conditional dependence is therefore defined as $P(U | V, W) \neq P(U | V)$. It means that there is at least one value for U , V and W for which the negation holds. But since $P(U | V) = \sum_w P(U | V, w)P(w | V)$, there are at least two values of W for which $P(U | V, w) \neq P(U | V)$. This can be understood by noting that $P(U | V)$ is a weighted average of $P(U | V, W)$. If the probability for one value of W is higher than the average $P(U | V)$, there must be at least one value for which the probability is lower.

For completeness we restate the postulate which is necessary to use IPC.

Postulate 2 *The non-trivial polynomial constraints among different CPDs responsible for non-Markovian CIs cannot be decomposed into product constraints without increasing their complexity, i.e. $C(c) < C(c_1) + \dots + C(c_n)$ for any decomposition of non-product constraint c into product constraints c_1, \dots, c_n .*

Whenever we encounter a non-trivial constraint among different CPDs, we may assume that they are represented by a non-product constraint in the minimal decomposition (see definition of IPC). The parameterization could still satisfy a product constraint as well, but we will show that there are only 3 cases of such product constraints leading to CIs. They will be ruled out explicitly.

Lemma 6 *Given a Bayesian network defined over discrete variables satisfying MIN and IPC, any two adjacent variables X and Y , where X is a parent of Y , are dependent conditioned on any subset of the other parents of Y .*

Proof We will prove that $X \perp\!\!\!\perp Y \mid W$ with $W \subset \text{Parents}(Y) \setminus X$. We denote all other parents of Y by $U (= \text{Parents}(Y) \setminus X \setminus W)$. If U is empty, the dependency follows from MIN (otherwise the edge between X and Y can be removed without jeopardizing the factorization). Otherwise:

$$P(Y \mid X, W) = \sum_u P(u \mid X, W) P(Y \mid X, u, W)$$

Independence $X \perp\!\!\!\perp Y \mid W$ would imply (Eq. 10) that $\forall x_1, x_2$:

$$\begin{aligned} &P(u_1 \mid x_1, W) P(Y \mid x_1, u_1, W) + P(u_2 \mid x_1, W) P(Y \mid x_1, u_2, W) + \dots \\ &= P(u_1 \mid x_2, W) P(Y \mid x_2, u_1, W) + P(u_2 \mid x_2, W) P(Y \mid x_2, u_2, W) + \dots \end{aligned}$$

In this equation, $P(Y \mid x_i, u_j, W) \neq P(Y \mid x_i, u_j, W)$ for at least one set of indices i, j, k since, by MIN, $P(Y \mid \text{Parents}(Y)) \neq P(Y \mid \text{Parents}(Y) \setminus X)$. The equation would therefore only hold when there is a constraint satisfied between CPD_Y and the CPDs defining $P(U \mid X, W)$, which is ruled out by IPC. The latter follows from the postulate and the exclusion of non-minimality, the only case in which a product constraint can lead to independence $X \perp\!\!\!\perp Y$. ■

Lemma 7 Assume that MIN and IPC holds for some Bayesian network defined over discrete variables. Let $X \perp\!\!\!\perp Y \mid Z'$ where X, Y are arbitrary nodes and Z' is a set of nodes. Let U denote the set of all variables on the non-blocked paths between X and Y , including X and Y . Then, for every set $Z'' \subset Z' \cup U$, when

$$X \perp\!\!\!\perp Y \mid Z', Z''$$

holds, there are deterministic relationships.

Proof We have to check when $P(Y \mid X, Z', Z'') = P(Y \mid Z', Z'')$. Writing $P(Y \mid X, Z')$ in function of the factors of the factorization results in a function containing the CPD_{U_i} 's for all $U_i \in U$ and factors describing $P(\text{part}(U))$ with $\text{part}(U) \in \text{Parents}(U)$ for each U . Dependency $Y \perp\!\!\!\perp X \mid Z'$ means that the distribution $P(Y \mid X, Z')$ depends on X . Conditioning on $Z'' \in Z'$ results in a new distribution for every value of Z'' . We want to know when this can result in a distribution that becomes independent from X . We have to consider 3 different types of variables of Z'' by looking how they participate in the CPDs containing variables of U .

- (1) Consider that Z'' is a member of a $\text{Parents}(U)$ set but no other parent of that U is in U . Conditioning on Z'' will only change $P(U)$, which cannot affect the dependence between X and Y under IPC unless U is determined by Z'' which is discussed below.
- (2) Consider that Z'' is a member of a $\text{Parents}(U)$ and there is another parent of U in U . Then Z'' participates in a CPD of the form $P(U_1|U_2, Z'')$. Conditioning on Z'' results in a new CPD $P(U_1|U_2)$. This can only affect the dependence between X and Y if it makes U_1 independent from U_2 . This is ruled out by Lemma 6 and MIN.
- (3) If Z'' is a descendant of some U_i, X or Y , then we apply Bayes' theorem:

$$P(Y \mid X, Z', Z'') = P(Y \mid X, Z') \frac{P(Z'' \mid X, Y, Z')}{P(Z'' \mid X, Z')} \quad (11)$$

Z'' is not present in the first factor which depends on X . Therefore, for independence of X , a constraint between the three factors of Eq. 11 must be met. By construction, this will result in a constraint between the parameters of several CPDs, which is excluded by IPC or a deterministic relation must be present. This is proved in the following.

To get independence without a constraint among different CPDs, a factor of the form $P(K \mid L, M)$ must become equal to $P(K \mid M)$. This means that

$$P(K \mid l_1, M) = P(K \mid l_2, M) \quad \forall l_1, l_2 \in L_{\text{atom}}$$

The conditional distribution of the left hand side is undefined whenever $P(l, m) = 0$. If we rule out constraints leading to independence (IPC), independence can only happen when

$$P(l_1, m) = 0 \text{ or } P(l_2, m) = 0 \text{ whenever } P(K \mid l_1, m) \neq P(K \mid l_2, m). \quad (12)$$

The condition can only apply for one CPD (IPC), so it must be a condition on CPD_L or CPD_M which holds for all parameterizations of the other CPDs. Then Eq. 12 only holds if $P(l, m) \neq 0$ for exactly one value of l given m . It follows that $L = f(M)$. ■

Lemma 9 If MIN, INDET, IPC holds for a factorization of a joint probability distribution defined over discrete variables, then for any v-structure $X \rightarrow Z \leftarrow Y$ in the DAG and for all W not containing X, Y or Z :

$$\begin{aligned} X \perp\!\!\!\perp Y \mid Z, W &\Leftrightarrow X \perp\!\!\!\perp Y \text{ and } P(Z \mid X, Y) \text{ is a proportional CPD} \\ &\text{w/ descendants of } Z : X \perp\!\!\!\perp Y \mid U, W \end{aligned} \quad (13)$$

Proof Assume that Y is not an ancestor of X (otherwise swap X and Y in the following). (1) First we prove the first equation for an empty set W and $U \neq Z$.

$$\begin{aligned} P(Y \mid X, Z) &= \frac{P(Z \mid X, Y) \cdot P(Y \mid X)}{P(Z \mid X)} \\ &= \frac{P(Z \mid X, Y) \cdot P(Y \mid X)}{\sum_{y'} P(Z \mid y', X) P(y' \mid X)} \end{aligned}$$

It follows that (using $A/B = 1/(B/A)$)

$$P(y \mid X, Z) = \frac{1}{1 + \frac{\sum_{y' \neq y} P(Z|y', X) P(y'|X)}{P(Z|y, X) P(y|X)}} \quad \forall y$$

Then, for $X \perp\!\!\!\perp Y \mid Z$ Eq. 10 gives:

$$\frac{P(Z \mid y_1, x_1) \cdot P(y_1|z_1)}{\sum_{y' \neq y_1} P(Z \mid y', x_1) P(y'|z_1)} = \frac{P(Z \mid y_1, x_2) \cdot P(y_1|z_2)}{\sum_{y' \neq y_1} P(Z \mid y', x_2) P(y'|z_2)} \quad \forall x_1, x_2$$

This equation defines a constraint between $P(Z \mid X, Y)$ and $P(Y \mid X)$ unless $Y \perp\!\!\!\perp X$ and the domain of Y contains only two values, y_1 and y_2 . Then we have $P(Y \mid X) = P(Y)$ and $P(y_2) =$

$1 - P(y_1)$, which results in a constraint on $P(Z | X, Y)$:

$$\begin{aligned} \frac{P(Z | y_1, x_1).P(y_1)}{P(Z | y_2, x_1)(1 - P(y_1))} &= \frac{P(Z | y_1, x_2).P(y_1)}{P(Z | y_2, x_2)(1 - P(y_1))} \quad \forall x_1, x_2 \\ &\Rightarrow \\ \frac{P(Z | y_1, x_1)}{P(Z | y_2, x_1)} &= \frac{P(Z | y_1, x_2)}{P(Z | y_2, x_2)} = \dots = \frac{P(Z | y_1, x_n)}{P(Z | y_2, x_n)} \end{aligned} \quad (14)$$

Which is a relation only depending on the parameterization of $P(Z | X, Y)$. In function of CPD_Z it gives:

$$P(Z|x, y) = \sum_{\mathbf{o}} P(\mathbf{o}|x, y)P(Z|x, y, \mathbf{o}) \text{ with } \mathbf{o} \in \text{domain of Parents}(Z) \setminus X \setminus Y$$

Since Eq. 14 must hold independent of $P(\mathbf{o}|x, y)$ it follows that CPD_Z must be a *proportional CPT* (Eq. 9).

Next, we consider that the \mathbf{W} is not empty. (2) For the other parents of Z , $\mathbf{W}' \subset \mathbf{W}$ it leads in a similar way to a pCPD:

$$\frac{P(Z | y_1, x_1, \mathbf{W}')}{P(Z | y_2, x_1, \mathbf{W}')} = \frac{P(Z | y_1, x_2, \mathbf{W}')}{P(Z | y_2, x_2, \mathbf{W}')} = \dots = \frac{P(Z | y_1, x_n, \mathbf{W}')}{P(Z | y_2, x_n, \mathbf{W}')}.$$

- (3) Conditioning on some other variables $\mathbf{W}'' \subset \mathbf{W}$ cannot lead to independence by lemma 7.
- (4) For proving the second equation (Eq. 13), we first prove it for an empty set. We write the equation for $P(Y | X, U)$ in function of $P(Z | X, Y)$ with Bayes' theorem:

$$\begin{aligned} P(Y | X, U) &= \frac{P(U | X, Y).P(Y | X)}{P(U | X)} \\ P(Y | X, U) &= \frac{\sum_z P(U | z, X, Y).P(z | X, Y).P(Y | X)}{\sum_z P(U | z, X).P(z | X)} \end{aligned}$$

Since Y is not an ascendant of X , each of the equation's factors depends on different CPDs of the factorization. Independence of X cannot be obtained by a constraint on one CPD unless $Z = f(U)$ which is excluded by DET.

- (5) If $X \perp\!\!\!\perp Y | U$ holds unconditionally, conditioning on some other variables \mathbf{W} cannot lead to independence by lemma 7. ■

Theorem 10 *If for a given factorization of a JPD defined over discrete variables the conditions IPC, MIN, INDET and NOpCPD are met, then faithfulness holds:*

$$\forall \text{ disjoint subsets } \mathbf{X}, \mathbf{Y}, \mathbf{Z} \subset \mathbf{V} : \mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z} \Leftrightarrow \mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z}$$

with \mathbf{V} the set of all variables under consideration.

Proof We prove that $X \perp\!\!\!\perp Y | Z \Rightarrow X \perp\!\!\!\perp Y | Z$. The proof recursively cuts the non-blocked paths between X and Y into subpaths until we end up with a basic connection that follows under lemma 6 or lemma 9. With these lemmas it will be proven that the variables at the outer ends of each subpath

are dependent under the 4 conditions. Secondly, we will prove that combining subpaths results in a dependence. To illustrate this consider $X \rightarrow U \rightarrow Y$. The path between X and Y is cut by U . First, $X \perp\!\!\!\perp U$ and $U \perp\!\!\!\perp Y$ are proven and, secondly, that the concatenation of subpaths $X \rightarrow U$ and $U \rightarrow Y$ leads to a $X \perp\!\!\!\perp Y$.

Without loss of generality, we may consider that X is not a descendant of Y . If X and Y are adjacent, Lemma 6 proves the dependency. If X and Y are part of a v-structure $X \rightarrow Z \leftarrow Y$ of which Z or descendants of Z are in \mathbf{Z} , Lemma 9 proves the dependency. If they are part of several such v-structures, the same approach as in the proof of Lemma 9 can be used to prove dependency. Otherwise, take a minimal ordered cutset $\mathbf{U} \subset \mathbf{V} \setminus \{X, Y\}$ such that $X \perp\!\!\!\perp Y | \mathbf{Z}, \mathbf{U}$. Minimal means that omitting any element of the set would lead to a d -connection. Ordered means that $U_i \in \mathbf{U}$ is not a descendant (in the DAG corresponding to the factorization) of $U_j \in \mathbf{U}$ whenever $i < j$. From $X \perp\!\!\!\perp Y | \mathbf{Z}, \mathbf{U}$ follows $X \perp\!\!\!\perp Y | \mathbf{Z}, \mathbf{U}$, thus:

$$\begin{aligned} P(Y | X, \mathbf{Z}) &= \sum_{u_1} \dots \sum_{u_n} P(Y | \mathbf{Z}, u_1, \dots, u_n).P(u_1 \dots u_n | X, \mathbf{Z}) \\ &= \sum_{u_1} \dots \sum_{u_n} P(Y | \mathbf{Z}, u_1, \dots, u_n).P(u_1 | X, \mathbf{Z}) \prod_{i=2}^n P(u_i | X, \mathbf{Z}, u_1 \dots u_{i-1}) \end{aligned} \quad (15)$$

We will prove that (1) no u_i can be removed from the first factor ($Y \perp\!\!\!\perp U_i | \mathbf{Z}, \mathbf{U} \setminus U_i$) and that X cannot be removed from the following factors ($X \perp\!\!\!\perp U_i | \mathbf{Z}, U_{i+1} \dots U_n$ for all values of i). Under these dependencies we will prove (2) that the right hand side of the equation does not lead to a distribution which is independent of X .

(1) The dependencies are of the form $K \perp\!\!\!\perp L | M$. The d -connection of all dependencies, $K \perp\!\!\!\perp L | M$, follows from the minimality of the set \mathbf{U} . That the dependencies follow from the d -connections is proven in the same way as $X \perp\!\!\!\perp Y | \mathbf{Z}$. The non-blocked paths between K and L are cut into subpaths until the nodes K and L are adjacent or connected by a v-structure. In that case lemmas 6 and 9 apply and prove the dependency.

(2) Because of the dependencies, Eq. 15 cannot be reduced. The first factor depends on CPD_Y , the following on CPD_{U_i} . Therefore, independence of X and Y can only come from a specific parameterization of at least three CPDs. ■

Theorem 12 *If for a given factorization of a JPD defined over discrete variables the conditions IC, COMPLEX, MIN, INDET and NOpCPD are met, then faithfulness holds:*

$$\forall \text{ disjoint subsets } \mathbf{X}, \mathbf{Y}, \mathbf{Z} \subset \mathbf{V} : \mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z} \Leftrightarrow \mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z}$$

with \mathbf{V} the set of all variables under consideration.

Proof In the previous proofs, constraints among parameters from different CPDs where ruled out by IPC and postulate 2. Conditions COMPLEX and IC will rule out the same constraints by Theorem 4. ■

Theorem 13 *Given P a distribution over n variables and a DAG G with CPDs parameterized by $\theta_1, \dots, \theta_n$ describing a factorization of P .*

If the parameter vectors satisfy IPC, there exists a parameterization $\theta_1^*, \dots, \theta_n^*$ of the CPDs which has the same independencies as P and for which IC holds.

For discrete Bayesian networks and the linear case for continuous distributions, if the factorization satisfies IC and COMPLEX (the parameters θ_i^* of the parameter vectors $\theta_1, \dots, \theta_n$ have non-constant complexity: $K(\theta_i^* | \text{CPD}_i^*) > 0$), then IPC holds as well.

Proof

The first statement: if the parameter vectors satisfy IPC, there exists a parameterization $\theta_1, \dots, \theta_n$ of the CPDs which has the same independencies as P and for which IC holds (IPC \rightarrow IC).

By IPC no constraint should hold among the parameters of different CPDs for the independencies of P . It could be that IC is violated for the given parameterization $\theta_1, \dots, \theta_n$, but it is not necessary for any of the conditional independencies in P to hold. By IPC, the parameter subspace of every independence in P can cover the complete subspace or be a product subspace of the form $R_1 \times \dots \times R_n$, where every R_i is an arbitrary subset of S_i , the set of possible parameter vectors θ_j . By Theorem 12 the subsets R_i have a generic form: some values of parameter vector θ_i are zero or similar (violation of MIN or presence of deterministic relationships) or determined by the other parameters (pCPD), but the remaining values of the CPDs are not constrained. Therefore we can choose the parameters independently in the subsets R_i such that the same independencies hold but IC is not violated.

Now the second statement: if the factorization satisfies IC and COMPLEX, then IPC holds as well (IC and COMPLEX \rightarrow IPC).

Non-Markovian CIs require non-trivial polynomial constraints (Spirtes et al., 1993; Meek, 1995) among some parameters, which can be either parameters of a single CPD or among parameters of different CPDs. The first case is not a problem for IPC, while the latter would imply a violation of IC if COMPLEX, since Theorem 4 applies: polynomial constraints correspond to holomorphic functions having zeroes for a finite number of values only. As such, they have a constant description length, while we assumed that all parameters of the CPDs have non-constant complexity. ■

References

- Gregory Chaitin. On the length of programs for computing finite binary sequences. *J. Assoc. Comput. Mach.*, 13:547–569, 1966.
- Gregory Chaitin. A theory of program size formally identical to information theory. *J. Assoc. Comput. Mach.*, 22:329–340, 1975.
- Zhiqiang Chen, Kun Zhang, Laiwan Chan, and Bernhard Schölkopf. Causal discovery via reproducing kernel hilbert space embeddings. *Neural Computations*, 26:1484–1517, 2014.
- Povilas Danušis, Dominik Janzing, Joris Mooij, Jakob Zscheischler, Bastian Steudel, Kun Zhang, and Bernhard Schölkopf. Inferring deterministic causal relations. In *Procs of UAI-2010*, 2010.
- Daniel M. Hausman and James Woodward. Independence, invariance and the causal Markov condition. *British Journal For the Philosophy Of Science*, 50(4):521–583, 1999.

Dominik Janzing and Bernhard Schölkopf. Causal inference using the algorithmic Markov condition. *IEEE Transactions on Information Theory*, 56(10):5168–5194, 2010.

Dominik Janzing and Bastian Steudel. Justifying additive noise model-based causal discovery via algorithmic information theory. *Open Syst. Inform. Dynam.*, pages 189–212, 2010.

Dominik Janzing, Xiaohai Sun, and Bernhard Schölkopf. Distinguishing cause and effect via second order exponential models. <http://arxiv.org/abs/0910.5561>, 2009.

Dominik Janzing, Joris Mooij, Kun Zhang, Jan Lemeire, Jakob Zscheischler, Povilas Danušis, Bastian Steudel, and Bernhard Schölkopf. Information-geometric approach to inferring causal directions. *Artificial Intelligence*, 56(10):5168–5194, 2012.

Andrey Kolmogorov. Three approaches to the quantitative definition of information. *Problems Inform. Transmission*, 1(1):1–7, 1965.

Jan Lemeire and Dominik Janzing. Replacing causal faithfulness with algorithmic independence of conditionals. *Minds and Machines*, 23(2):227–249, 2013. ISSN 0924-6495.

Jan Lemeire, Kris Steenhaut, and Abdellah Touhani. When are graphical causal models not good models? In *Causality in the sciences*, J. Williamson, F. Russo and P. McKay, editors, Oxford University Press, 2011.

Jan Lemeire, Stijn Meganck, Francesco Cartella, and Tingting Liu. Conservative independence-based causal structure learning in absence of adjacency faithfulness. *Int. J. Approx. Reasoning*, 53(9):1305–1325, 2012.

Christopher Meek. Strong completeness and faithfulness in Bayesian networks. In *Procs of UAI-1995*, pages 411–418, 1995.

Joseph Ramsey, Jiji Zhang, and Peter Spirtes. Adjacency-faithfulness and conservative causal inference. In *Procs of UAI-2006*, pages 401–408, 2006.

Ray Solomonoff. A preliminary report on a general theory of inductive inference. *Technical report V-131*, Report ZTB-138 Zator Co., 1960.

Ray Solomonoff. A formal theory of inductive inference. *Information and Control, Part II*, 7(2): 224–254, 1964.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. Springer Verlag, 2nd edition, 1993.

Learning Theory for Distribution Regression

Zoltán Szabó*

ORCID 0000-0001-6183-7603

Gatsby Unit, University College London
Sainsbury Wellcome Centre, 25 Howland Street
London - W1T 4JG, UK

ZOLTAN.SZABO@GATSBY.UCL.AC.UK

Bharath K. Sriperumbudur

Department of Statistics
Pennsylvania State University
University Park, PA 16802, USA

BKS18@PSU.EDU

Barnabás Póczos

Machine Learning Department
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue Pittsburgh, PA 15213 USA

BAPOCZOS@CS.CMU.EDU

Arthur Gretton

ORCID 0000-0003-3169-7624
Gatsby Unit, University College London
Sainsbury Wellcome Centre, 25 Howland Street
London - W1T 4JG, UK

ARTHUR.GRETTON@GMAIL.COM

Editor: Ingo Steinwart

We focus on the distribution regression problem: regressing to vector-valued outputs from probability measures. Many important machine learning and statistical tasks fit into this framework, including multi-instance learning and point estimation problems without analytical solution (such as hyperparameter or entropy estimation). Despite the large number of available heuristics in the literature, the inherent two-stage sampled nature of the problem makes the theoretical analysis quite challenging, since in practice only samples from sampled distributions are observable, and the estimates have to rely on similarities computed between sets of points. To the best of our knowledge, the only existing technique with consistency guarantees for distribution regression requires kernel density estimation as an intermediate step (which often performs poorly in practice), and the domain of the distributions to be compact Euclidean. In this paper, we study a simple, analytically computable, ridge regression-based alternative to distribution regression, where we embed the distributions to a reproducing kernel Hilbert space, and learn the regressor from the embeddings to the outputs. Our main contribution is to prove that this scheme is consistent in the two-stage sampled setup under mild conditions (on separable topological domains enriched with kernels): we present an exact computational-statistical efficiency trade-off analysis showing that our estimator is able to match the *one-stage* sampled minimax op-

*. Now at Applied Mathematics Department, Center for Applied Mathematics, École Polytechnique, University of Paris-Saclay, Route de Saclay, 91128 Palaiseau Cedex, France.

timal rate (Caponnetto and De Vito, 2007; Steinwart et al., 2009). This result answers a 17-year-old open question, establishing the consistency of the classical set kernel (Haussler, 1999; Gärtner et al., 2002) in regression. We also cover consistency for more recent kernels on distributions, including those due to Christmann and Steinwart (2010).

Keywords: Two-Stage Sampled Distribution Regression, Kernel Ridge Regression, Mean Embedding, Multi-Instance Learning, Minimax Optimality

1. Introduction

We address the learning problem of *distribution regression* in the two-stage sampled setting, where we only have bags of samples from the probability distributions: we regress from probability measures to real-valued (Póczos et al., 2013) responses, or more generally to vector-valued outputs (belonging to an arbitrary separable Hilbert space). Many classical problems in machine learning and statistics can be analysed in this framework. On the machine learning side, multiple instance learning (Dietterich et al., 1997; Ray and Page, 2001; Dooly et al., 2002) can be thought of in this way, where each instance in a labeled bag is an i.i.d. (independent identically distributed) sample from a distribution. On the statistical side, tasks might include point estimation of statistics on a distribution without closed form analytical expressions (e.g., its entropy or a hyperparameter).

Intuitive description of our goal: Let us start with a somewhat informal definition of the distribution regression problem and an intuitive phrasing of our goals. Suppose that our data consist of $\mathbf{z} = \{(x_i, y_i)\}_{i=1}^l$, where x_i is a probability distribution, y_i is its label (in the simplest case $y_i \in \mathbb{R}$ or $y_i \in \mathbb{R}^d$) and each (x_i, y_i) pair is i.i.d. sampled from a meta distribution \mathcal{M} . However, we do not observe x_i directly; rather, we observe a sample $x_{i,1}, \dots, x_{i,N_i} \stackrel{i.i.d.}{\sim} x_i$. Thus the observed data are $\hat{\mathbf{z}} = \{(\{x_{i,n}\}_{n=1}^{N_i}, y_i)\}_{i=1}^l$. Since $x_{i,j}$ is sampled from x_i , and x_i is sampled from \mathcal{M} , we call this process two-stage sampling. Our goal is to predict a new y_{l+1} from a new batch of samples $x_{l+1,1}, \dots, x_{l+1,N_{l+1}}$ drawn from a new distribution $x_{l+1} \sim \mathcal{M}$. For example, in a medical application, the l^{th} patient might be identified with a probability distribution (x_i), which can be periodically assessed by blood tests $(\{x_{i,n}\}_{n=1}^{N_i})$. We are also given some health indicator of the patient (y_i), which might be inferred from his/her blood measurements. Based on the observations ($\hat{\mathbf{z}}$), we might wish to learn the mapping from the set of blood tests to the health indicator, and the hope is that by observing more patients (larger l) and performing a larger number of tests (larger N_i) the estimated mapping ($\hat{f} = \hat{f}(\hat{\mathbf{z}})$) becomes more “precise”. Briefly, we consider the following questions:

Can the distribution regression problem be solved consistently under mild conditions? What is the exact computational-statistical efficiency trade-off implied by the two-stage sampling?

In our work the estimated mapping (\hat{f}) is the analytical solution of a kernel ridge regression (KRR) problem.¹ The performance of \hat{f} depends on the assumed function class (\mathcal{H}), the

1. Beyond its simple analytical formula, kernel ridge regression also allows efficient distributed (Zhang et al., 2015; Richiárík and Tokáč, 2016), sketch (Alaoui and Mahoney, 2015; Yang et al., 2016) and Nyström based approximations (Rudi et al., 2015).

family of f candidates used in the ridge formulation. We shall focus on the analysis of two settings:

1. **Well-specified case** ($f_* \in \mathcal{H}$): In this case we assume that the regression function f_* belongs to \mathcal{H} . We focus on bounding the goodness of \hat{f} compared to f_* . In other words, if $\mathcal{R}[f_*]$ denotes the prediction error (expected risk) of f_* , then our goal is to derive a finite-sample bound for the excess risk, $\mathcal{E}(\hat{f}; f_*) = \mathcal{R}[\hat{f}] - \mathcal{R}[f_*]$ that holds with high probability. We make use of this bound to establish the consistency of the estimator (i.e., drive the excess risk to zero) and to derive the exact computational-statistical efficiency trade-off of the estimator as a function of the sample number (l , $N = N_i, V_i$) and the problem difficulty (see Theorem 5 and its corresponding remarks for more details).

2. **Misspecified case** ($f_* \in L^2(\mathcal{H})$): Since in practise it might be hard to check whether $f_* \in \mathcal{H}$, we also study the misspecified setting of $f_* \in L^2$, the relevant case is when $f_* \in L^2 \setminus \mathcal{H}$. In the misspecified setting the ‘richness’ of \mathcal{H} has crucial importance, in other words the size of $D_{\mathcal{H}}^2 = \inf_{f \in \mathcal{H}} \|f_* - f\|_2^2$, the approximation error from \mathcal{H} . Our main contributions consist of proving a finite-sample excess risk bound, using which we show that the proposed estimator can attain the ideal performance, i.e., $\mathcal{E}(\hat{f}; f_*) - D_{\mathcal{H}}^2$ can be driven to zero. Moreover, on smooth classes of f_* -s, we give a simple and explicit description for the computational-statistical efficiency trade-off of our estimator (see Theorem 9 and its corresponding remarks for more details).

There exist a vast number of heuristics to tackle learning problems on distributions; we will review them in Section 5. However, to the best of our knowledge, the only prior work addressing the *consistency* of regression on distributions requires kernel density estimation (Póczos et al., 2013; Oliva et al., 2014; Sutherland et al., 2016), which assumes that the response variable is scalar-valued,² and the covariates are nonparametric continuous distributions on \mathbb{R}^d . As in our setting, the exact forms of these distributions are unknown; they are available only through finite sample sets. Póczos et al. estimated these distributions through a kernel density estimator (assuming these distributions have a density) and then constructed a kernel regressor that acts on these kernel density estimates.³ Using the classical bias-variance decomposition analysis for kernel regressors, they showed the consistency of the constructed kernel regressor, and provided a polynomial upper bound on the rates, assuming the true regressor to be Hölder continuous, and the meta distribution that generates the covariates x_i to have finite doubling dimension (Kpotufe, 2011).⁴

One can define kernel learning algorithms on bags based on set kernels (Gärtner et al., 2002) by computing the similarity of the sets/bags of samples representing the input distributions; set kernels are also called multi-instance kernels or ensemble kernels, and are examples of convolution kernels (Haussler, 1999). In this case, the similarity of two sets

² Oliva et al. (2013, 2015) consider the case where the responses are also distributions or functions.
³ We would like to clarify that the kernels used in their work are classical smoothing kernels—extensively studied in non-parametric statistics (Györfi et al., 2002)—and not the reproducing kernels that appear throughout our paper.

⁴ Using a random kitchen sinks approach, with orthonormal basis projection estimators Oliva et al. (2014); Sutherland et al. (2016) propose distribution regression algorithms that can computationally handle large scale datasets; as with Póczos et al. (2013), these approaches are based on density estimation in \mathbb{R}^d .

is measured by the average pairwise point similarities between the sets. From a theoretical perspective, nothing is known about the consistency of set kernel based learning method since their introduction in 1999 (Haussler, 1999; Gärtner et al., 2002): i.e. in what sense (and with what rates) is the learning algorithm consistent, when the number of items per bag, and the number of bags, are allowed to increase?

It is possible, however, to view set kernels in a distribution setting, as they represent valid kernels between (mean) embeddings of empirical probability measures into a reproducing kernel Hilbert space (RKHS; Berlinet and Thomas-Agnan, 2004). The population limits are well-defined as being dot products between the embeddings of the generating distributions (Altmun and Smola, 2006), and for characteristic kernels the distance between embeddings defines a metric on probability measures (Sriperumbudur et al., 2011; Gretton et al., 2012). When bounded kernels are used, mean embeddings exist for all probability measures (Fukumizu et al., 2004). When we consider the distribution regression setting, however, there is no reason to limit ourselves to set kernels. Embeddings of probability measures to RKHS are used by Christmann and Steinwart (2010) in defining a yet larger class of easily computable kernels on distributions, via operations performed on the embeddings and their distances. Note that the relation between set kernels and kernels on distributions was also applied by MuanDET et al. (2012) for classification on distribution-valued inputs, however consistency was not studied in that work. We also note that motivated by the current paper, Lopez-Paz et al. (2015) have recently presented the first theoretical results about surrogate risk guarantees on a class (relying on uniformly bounded Lipschitz functionals) of soft distribution-classification problems.

Our **contribution** in this paper is to establish the learning theory of a simple, mean embedding based ridge regression (MERR) method for the distribution regression problem. This result applies both to the basic set kernels of Haussler (1999); Gärtner et al. (2002), the distribution kernels of Christmann and Steinwart (2010), and additional related kernels. We provide finite-sample excess risk bounds, prove consistency, and show how the two-stage sampled nature of the problem (bag size) governs the computational-statistical efficiency of the MERR estimator. More specifically, in the

1. **well-specified case:** We

- (a) derive finite-sample bounds on the excess risk: We construct $\mathcal{R}[\hat{f}] - \mathcal{R}[f_*] \leq r(l, N, \lambda)$ bounds holding with high probability, where λ is the regularization parameter in the ridge problem ($\lambda \rightarrow 0$, $l \rightarrow \infty$, $N = N_i \rightarrow \infty$).

- (b) establish consistency and computational-statistical efficiency trade-off of the MERR estimator on a general prior family $\mathcal{P}(b, c)$ as defined by Caporineto and De Vito (2007), where b captures the effective input dimension, and larger c means smoother f_* ($1 < b, c \in (1, 2]$). In particular, when the number of samples per bag is chosen as $N = n \log(l)$ and $a \geq \frac{b(c+1)}{bc+1}$, then the learning rate saturates at $l^{-\frac{1}{bc+1}}$, which is known to be one-stage sampled minimax optimal (Caporineto and De Vito, 2007). In other words, by choosing $a = \frac{b(c+1)}{bc+1} < 2$, we suffer *no loss in statistical performance* compared with the *best possible one-stage sampled estimator*.

Note: the advantage of considering the $\mathcal{P}(b, c)$ family is two-fold. It does not assume parametric distributions, yet certain complexity terms can be explicitly upper bounded in the family. This property will be exploited in our analysis. Moreover, (for special

input distributions) the parameter b can be related to the spectral decay of Gaussian Gram matrices, and existing analysis techniques (Steinwart and Christmann, 2008) may be used in interpreting these decay conditions.

2. misspecified case: We establish consistency and convergence rates even if $f_* \notin \mathcal{H}$. Particularly, by deriving finite-sample bounds on the excess risk we

- (a) prove that the MERR estimator can achieve the best possible approximation accuracy from \mathcal{H} , i.e. the $\mathcal{R}[f] - \mathcal{R}[f_*] - D_{\mathcal{H}}^2$ quantity can be driven to zero (recall that $D_{\mathcal{H}} = \inf_{f \in \mathcal{H}} \|f_* - f\|_2$). Specifically, this result implies that if \mathcal{H} is dense in L^2 ($D_{\mathcal{H}} = 0$), then the excess risk $\mathcal{R}[f] - \mathcal{R}[f_*]$ converges to zero.
- (b) analyse the computational-statistical efficiency trade-off: We show that by choosing the bag size as $N = \ell^{2a} \log(\ell)$ ($a > 0$) one can get rate $L^{-\frac{2sa}{s+1}}$ for $a \leq \frac{s+1}{s+2}$, and the rate saturates for $a \geq \frac{s+1}{s+2}$ at $L^{-\frac{2s}{s+2}}$, where the difficulty of the regression problem is captured by $s \in (0, 1]$ (a larger s means an easier problem). This means that easier tasks give rise to faster convergence (for $s = 1$, the rate is $L^{-\frac{2}{3}}$), the bag size N can again be *sub-quadratic* in ℓ ($2a \leq \frac{2(s+1)}{s+2} \leq \frac{4}{3} < 2$), and the rate at saturation is close to $\bar{r}(\ell) = L^{-\frac{2s}{2s+1}}$, which is the asymptotically optimal rate in the one-stage sampled setup, with real-valued output and stricter eigenvalue decay conditions (Steinwart et al., 2009).

Due to the differences in the assumptions made and the loss function used, a direct comparison of our theoretical result and that of Póczos et al. (2013) remains an open question, however we make three observations. First, our approach is more general, since we may regress from any probability measure defined on separable, topological domains endowed with kernels. Póczos et al.'s work is restricted to compact domains of finite dimensional Euclidean spaces, and requires the distributions to admit probability densities; distributions on strings, graphs, and other structured objects are disallowed. Second, in our analysis we will allow separable Hilbert space valued outputs, in contrast to the real-valued output considered by Póczos et al. (2013). Third, density estimates in high dimensional spaces suffer from slow convergence rates (Wasserman, 2006, Section 6.5). Our approach mitigates this problem, as it works directly on distribution embeddings, and does not make use of density estimation as an intermediate step.

The principal challenge in proving theoretical guarantees arises from the two-stage sampled nature of the inputs. In our analysis of the well-specified case, we make use of Caponnetto and De Vito (2007)'s results, which focus (only) on the one-stage sample setup. These results will make our analysis somewhat shorter (but still rather challenging) by giving upper bounds for some of the objective terms. Even the verification of these conditions requires care since the inputs in the ridge regression are themselves distribution embeddings (i.e., functions in a reproducing kernel Hilbert space).

In the misspecified case, RKHS methods alone are not sufficient to obtain excess risk bounds: one has to take into account the ‘‘richness’’ of the modelling RKHS class (\mathcal{H}) in the embedding L^2 space. The fundamental challenge is whether it is possible to achieve the best possible performance dictated by \mathcal{H} ; or in the special case when further smoothness conditions hold on f_* , what convergence rates can yet be attained, and what computational-statistical efficiency trade-off realized. The second smoothness property could be modelled

for example by range spaces of (fractional) powers of integral operators associated to \mathcal{H} . Indeed, there exist several results along these lines with KRR for the case of real-valued outputs: see for example (Sun and Wu, 2009a, Theorem 1.1), (Sun and Wu, 2009b, Corollary 3.2), (Mendelson and Neeman, 2010, Theorem 3.7 with Assumption 3.2). The question of optimal rates has also been addressed for the semi-supervised KRR setting (Caponnetto, 2006, Theorem 1), and for clipped KRR estimators (Steinwart et al., 2009) with integral operators of rapidly decaying spectrum. Our results apply more generally to the two-stage sampled setting and to vector valued outputs belonging to separable Hilbert spaces. Moreover, we obtain a general consistency result without range space assumptions, showing that the modelling power of \mathcal{H} can be fully exploited, and convergence to the best approximation available from \mathcal{H} can be realized.⁵

There are numerous areas in machine learning and statistics, where estimating vector-valued functions has crucial importance. Often in statistics, one is not only confronted with the estimation of a scalar parameter, but with a vector of parameters. On the machine learning side, multi-task learning (Evgeniou et al., 2005), functional response regression (Kadri et al., 2016), or structured output prediction (Brouard et al., 2011; Kadri et al., 2013) fall under the same umbrella: they can be naturally phrased as learning vector-valued functions (Micchelli and Pontil, 2005). The idea underlying all these tasks is simple and intuitive: if multiple prediction problems have to be solved simultaneously, it might be beneficial to exploit their dependencies. Imagine for example that the task is to predict the motion of a dancer: taking into account the interrelation of the actor's body parts is likely to lead to more accurate estimation, as opposed to predicting the individual parts one by one, independently. Successful real-world applications of a multi-task approach include for example preference modelling of users with similar demographics (Evgeniou et al., 2005), prediction of the daily precipitation profiles of weather stations (Kadri et al., 2010), acoustic-to-articulatory speech inversion (Kadri et al., 2016), identifying biomarkers capable of tracking the progress of Alzheimer's disease (Zhou et al., 2013), personalized human activity recognition based on iPod/iPhone accelerometer data (Sun et al., 2013), finger trajectory prediction in brain-computer interfaces (Kadri et al., 2012) or ecological inference (Flaxman et al., 2015); for a recent review on multi-output prediction methods see (Álvarez et al., 2011; Borchani et al., 2015). A mathematically sound way of encoding prior information about the relation of the outputs can be realized by operator-valued kernels and the associated vector-valued RKHSs (Pedrick, 1957; Micchelli and Pontil, 2005; Carmeli et al., 2006, 2010); this is the tool we use to allow vector-valued learning tasks.

Finally, we note that the current work extends our earlier conference paper (Szabó et al., 2015) in several important respects: we now show that the MERR method can attain the one-stage sampled minimax optimal rate; we generalize the analysis in the well-specified setting to allow outputs belonging to an arbitrary separable Hilbert spaces (in contrast to the original scalar-valued output domain); and we tackle the misspecified setting, obtaining finite sample guarantees, consistency, and computational-statistical efficiency trade-offs.

The paper is structured as follows: The distribution regression problem and the MERR technique are introduced in Section 2. Our assumptions are detailed in Section 3. We present our theoretical guarantees (finite-sample bounds on the excess risk, consistency,

⁵ Specializing our result, we get explicit rates and an exact computational-statistical efficiency description for MERR as a function of sample numbers and problem difficulty, for smooth regression functions.

computational-statistical efficiency trade-offs) in Section 4: the well-specified case is considered in Section 4.1, and the misspecified setting is the focus of Section 4.2. Section 5 is devoted to an overview of existing heuristics for learning on distributions. Conclusions are drawn in Section 6. Section 7 contains proof details. In Section 8 we discuss our assumptions with concrete examples.

2. The Distribution Regression Problem

Below we first introduce our notation (Section 2.1), then formally define the distribution regression task (Section 2.2).

2.1 Notation

We use the following notations throughout the paper:

- **Sets, topology, measure theory:** Let \mathcal{X} be a Hilbert space; $\text{cl}[Y]$ is the closure of a set $Y \subseteq \mathcal{X}$. $X_i \in I_i/S_i$ is the direct product of sets S_i . $f \circ g$ is the composition of function f and g . Let (\mathcal{X}, τ) be a topological space and let $\mathcal{B}(\mathcal{X}) := \mathcal{B}(\tau)$ be the Borel σ -algebra induced by the topology τ . If (\mathcal{X}, d) is a metric space, then $\mathcal{B} = \mathcal{B}(d)$ is the Borel σ -algebra generated by the open sets induced by metric d . $\mathcal{M}_1^+(\mathcal{X})$ denotes the set of Borel probability measures on the $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ measurable space. Given measurable spaces (U_1, \mathcal{S}_1) and (U_2, \mathcal{S}_2) , the $\mathcal{S}_1 \otimes \mathcal{S}_2$ product σ -algebra (Steinwart and Christmann, 2008, page 480) on the product space $U_1 \times U_2$ is the σ -algebra generated by the cylinder sets $U_1 \times S_2$, $S_1 \times U_2$ ($S_1 \in \mathcal{S}_1$, $S_2 \in \mathcal{S}_2$). The weak topology $(\tau_w = \tau_w(\mathcal{X}, \tau))$ on $\mathcal{M}_1^+(\mathcal{X})$ is defined as the weakest topology such that the $L_h : \mathcal{M}_1^+(\mathcal{X}, \tau_w) \rightarrow \mathbb{R}$, $L_h(x) = \int_{\mathcal{X}} h(u) d\tau(u)$ mapping is continuous for all $h \in C_b(\mathcal{X}) = \{\mathcal{X}, \tau\} \rightarrow \mathbb{R}$ bounded, continuous functions⁶.

- **Functional analysis:** Let $(N_1, \|\cdot\|_{N_1})$ and $(N_2, \|\cdot\|_{N_2})$ denote two normed spaces, then $\mathcal{L}(N_1, N_2)$ stands for the space of $N_1 \rightarrow N_2$ bounded linear operators; if $N_1 = N_2$, we will use the $\mathcal{L}(N_1) = \mathcal{L}(N_1, N_2)$ shorthand. For $M \in \mathcal{L}(N_1, N_2)$ the operator norm is defined as $\|M\|_{\mathcal{L}(N_1, N_2)} = \sup_{\|h\|_{N_2} \leq 1} \|Mh\|_{N_1}$. $\text{Im}(M) = \{Mn\}_{n \in N_1}$ denotes the range of M , $\text{Ker}(M) = \{n_1 \in N_1 : Mn_1 = 0\}$ is the null space of M . Let \mathcal{K} be a Hilbert space. The adjoint operator $M^* \in \mathcal{L}(\mathcal{K})$ of an operator $M \in \mathcal{L}(\mathcal{K})$ is the operator such that $\langle Ma, b \rangle_{\mathcal{K}} = \langle a, M^*b \rangle_{\mathcal{K}}$ for all a and b in \mathcal{K} . $M \in \mathcal{L}(\mathcal{K})$ is called positive if $\langle Ma, a \rangle_{\mathcal{K}} \geq 0$ ($\forall a \in \mathcal{K}$), self-adjoint if $M = M^*$, and trace class if $\sum_{j \in J} \langle M|e_j, e_j\rangle_{\mathcal{K}} < \infty$ for an $(e_j)_{j \in J}$ ONB (orthonormal basis) of \mathcal{K} ($|M| := (M^*M)^{\frac{1}{2}}$), in which case $\text{Tr}(M) := \sum_{j \in J} \langle M|e_j, e_j\rangle_{\mathcal{K}} < \infty$; compact if $\text{cl}[|M|a : a \in \mathcal{K}, \|a\|_{\mathcal{K}} \leq 1]$ is a compact set. Let \mathcal{K}_1 and \mathcal{K}_2 be Hilbert spaces. $M \in \mathcal{L}(\mathcal{K}_1, \mathcal{K}_2)$ is called Hilbert-Schmidt if $\|M\|_{\mathcal{L}_2(\mathcal{K}_1, \mathcal{K}_2)}^2 = \text{Tr}(M^*M) = \sum_{j \in J} \langle M|e_j, M|e_j\rangle_{\mathcal{K}_2} < \infty$ for some $(e_j)_{j \in J}$ ONB of \mathcal{K}_1 . The space of Hilbert-Schmidt operators is denoted by $\mathcal{L}_2(\mathcal{K}_1, \mathcal{K}_2) = \{M \in \mathcal{L}(\mathcal{K}_1, \mathcal{K}_2) : \|M\|_{\mathcal{L}_2(\mathcal{K}_1, \mathcal{K}_2)} < \infty\}$. We use the shorthand notation $\mathcal{L}_2(\mathcal{K}) = \mathcal{L}_2(\mathcal{K}, \mathcal{K})$ if $\mathcal{K} := \mathcal{K}_1 = \mathcal{K}_2$. $\mathcal{L}_2(\mathcal{K})$ is separable if and only if \mathcal{K} is separable (Steinwart and Christmann, 2008, page 506). Trace class and Hilbert-Schmidt operators over a \mathcal{K} Hilbert space are compact operators (Steinwart and Christmann, 2008, page 505-506); moreover,

$$\|A\|_{\mathcal{L}(\mathcal{K})} \leq \|A\|_{\mathcal{L}_2(\mathcal{K})}, \quad \forall A \in \mathcal{L}_2(\mathcal{K}), \quad (1)$$

$$\|AB\|_{\mathcal{L}_2(\mathcal{K})} \leq \|A\|_{\mathcal{L}_2(\mathcal{K})} \|B\|_{\mathcal{L}(\mathcal{K})}, \quad \forall A, B \in \mathcal{L}_2(\mathcal{K}). \quad (2)$$

I is the identity operator; $I_I \in \mathbb{R}^{K \times I}$ is the identity matrix.

- **RKHS, mean embedding:** Let $H = H(k)$ be an RKHS (Steinwart and Christmann, 2008) with $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ as the reproducing kernel. Denote by

$$X = \mu(\mathcal{M}_1^+(\mathcal{X})) = \{\mu_x : x \in \mathcal{M}_1^+(\mathcal{X}) \subseteq H, \quad \mu_x = \int_{\mathcal{X}} k(\cdot, u) dx(u) = \mathbb{E}_{u \sim x}[k(\cdot, u)] \in H$$

the set of mean embeddings (Berlinet and Thomas-Agnan, 2004) of the distributions to the space H .⁶ Let Y be a separable Hilbert space, where the inner product is denoted by $\langle \cdot, \cdot \rangle_Y$; the associated norm is $\|\cdot\|_Y$. $\mathcal{Y} = \mathcal{Y}(K)$ is the Y -valued RKHS (Pedrick, 1957; Micchelli and Pontil, 2005; Carmeli et al., 2006, 2010) of $X \rightarrow Y$ functions with $K : X \times X \rightarrow \mathcal{L}(Y)$ as the reproducing kernel (we will present some concrete examples of K in Section 3; see Table 1); $K_{\mu_x} \in \mathcal{L}(Y, \mathcal{Y})$ is defined as

$$K(\mu_x, \mu)(y) = (K_{\mu, y})(\mu_x), \quad (\forall \mu_x, \mu \in X), \quad \text{or } K(\cdot, \mu)(y) = K_{\mu, y}. \quad (3)$$

Further, $f(\mu_x) = K_{\mu_x}^* f$ ($\forall \mu_x \in X, f \in \mathcal{Y}$).

- **Regression function:** Let ρ be the μ -induced probability measure on the $Z = X \times Y$ product space, and let $\rho(\mu_x, y) = \rho(y|\mu_x)\rho_X(\mu_x)$ be the factorization of ρ into conditional and marginal distributions.⁷ The regression function of ρ with respect to the (μ_x, y) pair is denoted by

$$f_{\rho}(\mu_x) = \int_Y y d\rho(y|\mu_x) \quad (\mu_x \in X) \quad (4)$$

and for $f \in L_{\rho_X}^2$ let $\|f\|_{\rho} = \sqrt{\langle f, f \rangle_{\rho}} = \|f\|_{L_{\rho_X}^2} = \left[\int_X \|f(\mu_x)\|_Y^2 d\rho_X(\mu_x) \right]^{\frac{1}{2}}$. Let us assume that the operator-valued kernel $K : X \times X \rightarrow \mathcal{L}(Y)$ is a Mercer kernel (that is $\mathcal{Y} = \mathcal{Y}(K) \subseteq C(X, Y) = \{X \rightarrow Y \text{ continuous functions}\}$), is bounded ($\|BK\| < \infty$ such that $\|K(x, x)\|_{\mathcal{L}(Y)} \leq BK$), and is a compact operator for all $x \in X$. These requirements will be guaranteed by our assumptions; see Section 7.2.6. In this case, the inclusion $S_K^* : \mathcal{Y} \hookrightarrow L_{\rho_X}^2$ is bounded, and its adjoint $S_K : L_{\rho_X}^2 \rightarrow \mathcal{Y}$ is given by

$$(S_K g)(\mu_x) = \int_X K(\mu_x, \mu) g(\mu) d\rho_X(\mu). \quad (5)$$

We further define \tilde{T} as

$$\tilde{T} = S_K^* S_K : L_{\rho_X}^2 \rightarrow L_{\rho_X}^2; \quad (6)$$

in other words, the result of operation (5) belongs to \mathcal{Y} , which is embedded in $L_{\rho_X}^2$. \tilde{T} is a compact, positive, self-adjoint operator (Carmeli et al., 2010, Proposition 3), thus by the spectral theorem \tilde{T}^s exists, where $s \geq 0$.

6. The $x \rightarrow \mu_x$ mapping is defined for $\text{all } x \in \mathcal{M}_1^+(\mathcal{X})$ if k is bounded, i.e., $\sup_{u, v \in \mathcal{X}} k(u, v) < \infty$.

7. Our assumptions will guarantee the existence of ρ (see Section 3). Since Y is a Polish space (because it is separable Hilbert) the $\rho(y|\mu_x)$ conditional distribution ($y \in Y, \mu_x \in X$) is also well-defined (Steinwart and Christmann, 2008, Lemma A.3.16, page 487).

2.2 Distribution Regression

We now formally define the distribution regression task. Let us assume that $\mathcal{M}_1^+(\mathcal{X})$ is endowed with $S_1 = \mathcal{B}(\tau_w)$, the weak-topology generated σ -algebra; thus $(\mathcal{M}_1^+(\mathcal{X}), S_1)$ is a measurable space. In the *distribution regression* problem, we are given samples $\hat{\mathbf{z}} = \{(x_{i,n}, y_i)\}_{i=1}^l$ with $x_{i,1}, \dots, x_{i,N_i} \stackrel{i.i.d.}{\sim} x_i$ where $\mathbf{z} = \{(x_i, y_i)\}_{i=1}^l$ with $x_i \in \mathcal{M}_1^+(\mathcal{X})$ and $y_i \in Y$ drawn i.i.d. from a joint meta distribution \mathcal{M} defined on the measurable space $(\mathcal{M}_1^+(\mathcal{X}) \times Y, S_1 \otimes \mathcal{B}(Y))$, the product space enriched with the product σ -algebra. Unlike in classical supervised learning problems, the problem at hand involves two levels of randomness, wherein first \mathbf{z} is drawn from \mathcal{M} , and then $\hat{\mathbf{z}}$ is generated by sampling points from x_i for all $i = 1, \dots, l$. The goal is to learn the relation between the random distribution x and response y based on the observed $\hat{\mathbf{z}}$. For notational simplicity, we will assume that $N = N_i$ ($\forall i$).

As in the classical regression problem ($\mathbb{R}^d \rightarrow \mathbb{R}$), distribution regression can be tackled via kernel ridge regression (using a squared loss as the discrepancy criterion). The kernel (say $K_{\mathcal{G}}$) is defined on $\mathcal{M}_1^+(\mathcal{X})$, and the regressor is then modelled by an element in the RKHS $\mathcal{G} = \mathcal{G}(K_{\mathcal{G}})$ of functions mapping from $\mathcal{M}_1^+(\mathcal{X})$ to Y . In this paper, we choose $K_{\mathcal{G}}(x, x') = K(\mu_x, \mu_{x'})$ where $x, x' \in \mathcal{M}_1^+(\mathcal{X})$ and so that the function (in 9) to describe the (x, y) random relation is constructed as a composition $f \circ \mu_x$, i.e.

$$\mathcal{M}_1^+(\mathcal{X}) \xrightarrow{\mu} X (\subseteq H = H(k)) \xrightarrow{f \in \mathcal{G} = \mathcal{G}(K)} Y. \quad (7)$$

In other words, the distribution $x \in \mathcal{M}_1^+(\mathcal{X})$ is first mapped to $X \subseteq H$ by the mean embedding μ , and the result is composed with f , an element of the RKHS \mathcal{G} .

Let the expected risk for a $\tilde{f}: X \rightarrow Y$ (measurable) function be defined as

$$\mathcal{R}[\tilde{f}] = \mathbb{E}_{(x,y) \sim \mathcal{M}} \|\tilde{f}(\mu_x) - y\|_Y^2,$$

which is minimized by the f_{ρ} regression function. The classical regularization approach is to optimize

$$f_{\mathbf{z}}^{\lambda} = \arg \min_{f \in \mathcal{G}} \frac{1}{l} \sum_{i=1}^l \|f(\mu_{x_i}) - y_i\|_Y^2 + \lambda \|f\|_{\mathcal{G}}^2 \quad (8)$$

instead of \mathcal{R} , based on samples \mathbf{z} . Since \mathbf{z} is not available, we consider the objective function defined by the observable quantity $\hat{\mathbf{z}}$,

$$f_{\hat{\mathbf{z}}}^{\lambda} = \arg \min_{f \in \mathcal{G}} \frac{1}{l} \sum_{i=1}^l \|f(\mu_{\hat{x}_i}) - y_i\|_Y^2 + \lambda \|f\|_{\mathcal{G}}^2, \quad (9)$$

where $\hat{x}_i = \frac{1}{N} \sum_{n=1}^N \delta_{x_{i,n}}$ is the empirical distribution determined by $\{x_{i,n}\}_{n=1}^N$. The ridge regression objective function has an analytical solution: given training samples $\hat{\mathbf{z}}$, the prediction for a new t test distribution is

$$(\hat{\mathbf{z}}^{\lambda} \circ \mu)(t) = \mathbf{k}(\mathbf{K} + t\lambda Y)^{-1} [y_1; \dots; y_l], \quad (10)$$

where $\mathbf{k} = [K(\mu_{\hat{x}_1}, \mu_t), \dots, K(\mu_{\hat{x}_l}, \mu_t)] \in \mathcal{L}(Y)^{l \times l}$, $\mathbf{K} = [K(\mu_{\hat{x}_i}, \mu_{\hat{x}_j})] \in \mathcal{L}(Y)^{l \times l}$, $[y_1; \dots; y_l] \in Y^l$.

Remark 1

- It is important to note that the algorithm has access to the sample points only via their mean embeddings $\{\mu_{\hat{x}_i}\}_{i=1}^l$ in Eq. (9).
- There is a two-stage sampling difficulty to tackle: The transition from f_{ρ} to $f_{\mathbf{z}}^{\lambda}$ represents the fact that we have only l distribution samples (\mathbf{z}); the transition from $f_{\mathbf{z}}^{\lambda}$ to $f_{\hat{\mathbf{z}}}^{\lambda}$ means that the x_i distributions can be accessed only via samples ($\hat{\mathbf{z}}$).
- While ridge regression can be performed using the kernel $K_{\mathcal{G}}$, the two-stage sampling makes it difficult to work with arbitrary $K_{\mathcal{G}}$. By contrast, our choice of $K_{\mathcal{G}}(x, x') = K(\mu_x, \mu_{x'})$ enables us to handle the two-stage sampling by estimating μ_x with an empirical estimator, and using it in the algorithm as shown above.
- In case of scalar output ($Y = \mathbb{R}$), $\mathcal{L}(Y) = \mathcal{L}(\mathbb{R}) = \mathbb{R}$ and (10) is a standard linear equation with $\mathbf{K} \in \mathbb{R}^{l \times l}$, $\mathbf{k} \in \mathbb{R}^{1 \times l}$. More generally, if $Y = \mathbb{R}^d$, then $\mathcal{L}(Y) = \mathcal{L}(\mathbb{R}^d) = \mathbb{R}^{d \times d}$ and (10) is still a finite-dimensional linear equation with $\mathbf{K} \in \mathbb{R}^{(dl) \times (dl)}$ and $\mathbf{k} \in \mathbb{R}^{d \times (dl)}$.
- One could also formulate the problem (and get guarantees) for more abstract $X \subseteq H \rightarrow Y$ regression tasks [see Eq. (7)] on a convex set X with H and Y being general, separable Hilbert spaces. Since distribution regression is probably the most accessible example where two-stage sampling appears, and in order to keep the presentation simple, we do not consider such extended formulations in this work.

Our main goals in this paper are as follows: first, to analyse the excess risk

$$\mathcal{E}(\hat{\mathbf{z}}^{\lambda}, f_{\rho}) := \mathcal{R}[f_{\hat{\mathbf{z}}}^{\lambda}] - \mathcal{R}[f_{\rho}],$$

both when $f_{\rho} \in \mathcal{H}$ (the well-specified case) and $f_{\rho} \in L_{\rho_X}^2 \setminus \mathcal{H}$ (the misspecified case); second, to establish consistency ($\mathcal{E}(f_{\hat{\mathbf{z}}}^{\lambda}, f_{\rho}) \rightarrow 0$, or in the misspecified case $\mathcal{E}(f_{\hat{\mathbf{z}}}^{\lambda}, f_{\rho}) - D_{\mathcal{H}}^2 \rightarrow 0$, where $D_{\mathcal{H}}^2 := \inf_{g \in \mathcal{H}} \|f_{\rho} - S_K g\|_{\rho}^2$ is the approximation error of f_{ρ} by a function in \mathcal{H}); and third, to derive an exact computational-statistical efficiency trade-off as a function of the (l, N, λ) triplet, and of the difficulty of the problem.

3. Assumptions

In this section, we detail our assumptions on the (\mathcal{X}, Y, k, K) quartet. Our analysis for the well-specified case uses existing ridge regression results (Caponnetto and De Vito, 2007) focusing on problem (8) where only a single-stage sampling is present, hence we have to verify the associated conditions. Though we make use of these results, the analysis still remains challenging: the available bounds can moderately shorten our proof. We must take particular care in verifying that Caponnetto and De Vito (2007)'s conditions are met, since they need to hold for the space of mean embeddings of the distributions ($X = \mu(\mathcal{M}_1^+(\mathcal{X}))$), whose properties as a function of \mathcal{X} and H must themselves be established.

Our **assumptions** are as follows:

1. (\mathcal{X}, τ) is a separable, topological space.

2. Y is a separable Hilbert space.
3. k is bounded, in other words $\exists B_k < \infty$ such that $\sup_{u,v \in X} k(u, v) \leq B_k$, and continuous.
4. The $\{K_{\mu_a}\}_{\mu_a \in X}$ operator family is uniformly bounded in Hilbert-Schmidt norm and Hölder continuous in operator norm. Formally, $\exists B_K < \infty$ such that

$$\|K_{\mu_a}\|_{\mathcal{L}(Y; \mathcal{Y})}^2 = \text{Tr} (K_{\mu_a}^* K_{\mu_a}) \leq B_K, \quad (\forall \mu_a \in X), \quad (11)$$

and $\exists L > 0, h \in (0, 1]$ such that the mapping $K(\cdot) : X \rightarrow \mathcal{L}(Y, \mathcal{Y})$ is Hölder continuous:

$$\|K_{\mu_a} - K_{\mu_b}\|_{\mathcal{L}(Y; \mathcal{Y})} \leq L \|\mu_a - \mu_b\|_H^h, \quad (\forall \mu_a, \mu_b) \in X \times X. \quad (12)$$

5. y is bounded: $\exists C < \infty$ such that $\|y\|_Y \leq C$ almost surely. These requirements hold under mild conditions: in Section 8, we provide insight into the consequences of our assumptions, with several concrete illustrations (e.g. regression with self- and RBF-type kernels).

4. Error Bounds, Consistency & Computational-Statistical Efficiency Trade-off

In this section, we present our analysis of the consistency of the mean embedding based ridge regression (MERR) method.

Given the estimator $(f_{\frac{\lambda}{2}}^\lambda)$ in Eq. (9), we derive finite-sample high probability upper bounds (see Theorems 2 and 7) for the excess risk $\mathcal{E}(f_{\frac{\lambda}{2}}^\lambda, f_\rho)$; and in the misspecified setting, for the excess risk compared to the best attainable value from \mathcal{F}_c , i.e., $\mathcal{E}(f_{\frac{\lambda}{2}}^\lambda, f_\rho) - D_{\mathcal{F}_c}^\lambda$. We illustrate the bounds for particular classes of prior distributions, and work through special cases to obtain consistency conditions and computational-statistical efficiency trade-offs (see Theorems 4, 9 and the 3rd bullet of Remark 8). The main challenge is how to turn the convergence rates of the mean embeddings into those for an error \mathcal{E} of the predictor. Although the main ideas of the proofs can be summarized relatively briefly, the full details are more demanding. High-level ideas with the sketches of the proofs and the obtained results are presented in Section 4.1 (well-specified case) and Section 4.2 (misspecified case). The derivations of some technical details of Theorems 2 and 7 are available in Section 7.

4.1 Results for the Well-specified Case

We first focus on the well-specified case ($f_\rho \in \mathcal{F}_c$) and present our first main result. We derive a high probability upper bound for the excess risk $\mathcal{E}(f_{\frac{\lambda}{2}}^\lambda, f_\rho)$ of the MERR method (Theorem 2). The upper bound is instantiated for a general class of prior distributions (Theorem 4), which leads to a simple computational-statistical efficiency description (Theorem 5); this shows (among others) conditions when the MERR technique is able to achieve the *one-stage* sampled minimax optimal rate. We first give a high-level sketch of our convergence analysis and an intuitive interpretation of the results. An outline of the main proof ideas is given below, with technical details in Section 7.

Let us define $\mathbf{x} = \{x_i\}_{i=1}^N$ and $\mathbf{X} = \{x_{i,n}\}_{i=1}^N$ as the ‘x-part’ of \mathbf{z} and $\hat{\mathbf{z}}$, respectively. One can express $f_{\frac{\lambda}{2}}^\lambda$ [Eq. (8)] (Caponnetto and De Vito, 2007), and similarly $f_{\frac{\lambda}{2}}^\lambda$ [Eq. (9)],

38

$$f_{\frac{\lambda}{2}}^\lambda = (T_{\mathbf{x}} + \lambda)^{-1} g_{\mathbf{x}}, \quad T_{\mathbf{x}} = \frac{1}{l} \sum_{i=1}^l T_{\mu_{x_i}}, \quad g_{\mathbf{x}} = \frac{1}{l} \sum_{i=1}^l K_{\mu_{x_i}} g_{i\mathbf{x}}, \quad (13)$$

$$f_{\frac{\lambda}{2}}^\lambda = (T_{\hat{\mathbf{x}}} + \lambda)^{-1} g_{\hat{\mathbf{x}}}, \quad T_{\hat{\mathbf{x}}} = \frac{1}{l} \sum_{i=1}^l T_{\mu_{\hat{x}_i}}, \quad g_{\hat{\mathbf{x}}} = \frac{1}{l} \sum_{i=1}^l K_{\mu_{\hat{x}_i}} g_{i\hat{\mathbf{x}}}, \quad (14)$$

where $T_{\mu_a} = K_{\mu_a} K_{\mu_a}^* \in \mathcal{L}(\mathcal{Y})$ ($\mu_a \in X$), $T_{\mathbf{x}}, T_{\hat{\mathbf{x}}} : \mathcal{Y} \rightarrow \mathcal{Y}$, $g_{\mathbf{x}}, g_{\hat{\mathbf{x}}} \in \mathcal{Y}$. By these explicit expressions, one can decompose the excess risk into 5 terms (Szabó et al., 2015, Section A.1.8):

$$\mathcal{E}(f_{\frac{\lambda}{2}}^\lambda, f_\rho) = \mathcal{R}[f_{\frac{\lambda}{2}}^\lambda] - \mathcal{R}[f_\rho] \leq 5[S_{-1} + S_0 + A(\lambda) + S_1 + S_2],$$

where

$$S_{-1} = S_{-1}(\lambda, \mathbf{z}, \hat{\mathbf{z}}) = \|\sqrt{T}(T_{\hat{\mathbf{x}}} + \lambda I)^{-1}(g_{\hat{\mathbf{x}}} - g_{\mathbf{x}})\|_{\mathcal{Y}}^2, \quad (15)$$

$$S_0 = S_0(\lambda, \mathbf{z}, \hat{\mathbf{z}}) = \|\sqrt{T}(T_{\hat{\mathbf{x}}} + \lambda I)^{-1}(T_{\hat{\mathbf{x}}} - T_{\mathbf{x}}) f_{\frac{\lambda}{2}}^\lambda\|_{\mathcal{Y}}^2, \quad (16)$$

$$A(\lambda) = \|\sqrt{T}(f_{\frac{\lambda}{2}}^\lambda - f_\rho)\|_{\mathcal{Y}}^2, \quad S_1 = S_1(\lambda, \mathbf{z}) = \|\sqrt{T}(T_{\mathbf{x}} + \lambda I)^{-1}(g_{\mathbf{x}} - T_{\mathbf{x}} f_\rho)\|_{\mathcal{Y}}^2,$$

$$S_2 = S_2(\lambda, \mathbf{z}) = \|\sqrt{T}(T_{\hat{\mathbf{x}}} + \lambda I)^{-1}(T_{\hat{\mathbf{x}}} - T_{\mathbf{x}})(f_{\frac{\lambda}{2}}^\lambda - f_\rho)\|_{\mathcal{Y}}^2,$$

$$f_{\frac{\lambda}{2}}^\lambda = \arg \min_{f \in \mathcal{F}_c} (\mathcal{R}[f] + \lambda \|f\|_{\mathcal{Y}}^2), \quad T = \int_X T_{\mu_a} d\rho_X(\mu_a) = S_K S_K^* : \mathcal{Y} \rightarrow \mathcal{Y}. \quad (17)$$

Three of the terms ($S_1, S_2, A(\lambda)$) are identical to the terms in Caponnetto and De Vito (2007), hence the earlier bounds can be applied. The two new terms (S_{-1}, S_0) resulting from two-stage sampling will be upper bounded by making use of the convergence of the empirical mean embeddings. These bounds will lead to the following results:

Theorem 2 (Finite-sample excess risk bounds; well-specified case) *Let*

$K(\cdot) : X \rightarrow \mathcal{L}(Y, \mathcal{Y})$ *be Hölder continuous with constants* L, h . *Let* $l \in \mathbb{Z}^+, N \in \mathbb{Z}^+, 0 < \lambda, 0 < \eta < 1, 0 < \delta$, $C_\eta = 32 \log^2(6/\eta)$, $\|y\|_Y \leq C$ (a.s.) *and* $A(\lambda)$ *be the residual as defined above. Define* $M = 2(C + \|f_\rho\|_{\mathcal{Y}} \sqrt{B_K})$, $\Sigma = \frac{M}{2}$, T *as in* (17), $\mathfrak{B}(\lambda) = \|f_{\frac{\lambda}{2}}^\lambda - f_\rho\|_{\mathcal{Y}}^2$ *as the reconstruction error, and* $\mathcal{N}(\lambda) = \text{Tr}((T + \lambda I)^{-1} T)$ *as the effective dimension. Then with probability at least* $1 - \eta - e^{-\delta}$, *the excess risk can be upper bounded as*

$$\begin{aligned} \mathcal{E}(f_{\frac{\lambda}{2}}^\lambda, f_\rho) &\leq 5 \left\{ \frac{4L^2 (1 + \sqrt{\log(l) + \delta})^{2h}}{\lambda N^h} (2B_K)^h \left[C^2 + 4B_K \times \right. \right. \\ &\quad \left. \left. \times \left(\log^2 \left(\frac{6}{\eta} \right) \left[\frac{64}{\lambda} \left[\frac{M^2 B_K}{l^2 \lambda} + \frac{\Sigma^2 \mathcal{N}(\lambda)}{l} \right] + \frac{24}{\lambda^2} \left[\frac{4B_K^2 \mathfrak{B}(\lambda)}{l^2} + \frac{B_K A(\lambda)}{l} \right] \right) + \mathfrak{B}(\lambda) + \|f_\rho\|_{\mathcal{Y}}^2 \right] \right. \\ &\quad \left. \left. + A(\lambda) + C_\eta \left[\frac{B_K^2 \mathfrak{B}(\lambda)}{l^2 \lambda} + \frac{B_K A(\lambda)}{4\lambda} + \frac{B_K M^2}{l^2 \lambda} + \frac{\Sigma^2 \mathcal{N}(\lambda)}{l} \right] \right\} \end{aligned}$$

if $l \geq 2C_\eta B_K \mathcal{N}(\lambda) \lambda$, $\lambda \leq \|T\|_{\mathcal{L}(Y; \mathcal{Y})}$ *and* $N \geq (1 + \sqrt{\log(l) + \delta})^2 2^{-\frac{h+\delta}{2}} B_K (B_K)^{\frac{1}{h}} L^{\frac{2}{h}} \lambda^{\frac{2}{h}}$.

Below we specialize our excess risk bound for a general prior class, which captures the difficulty of the regression problem as defined in Caponnetto and De Vito (2007). This $\mathcal{P}(b, c)$ class is described by two parameters b and c : larger b means faster decay of the eigenvalues of the covariance operator T [in Eq. (17)], hence smaller effective input dimension; larger c corresponds to a smoother regression function. Formally:

Definition of the $\mathcal{P}(b, c)$ class: Let us fix the positive constants R, α, β . Then given $1 < b, c \in (1, 2]$, the $\mathcal{P}(b, c)$ class is the set of probability distributions ρ on $Z = X \times Y$ such that

1. a range space assumption is satisfied: $\exists g \in \mathcal{H}$ s.t. $f_\rho = T^{\frac{c-1}{2}} g$ with $\|g\|_{\mathcal{H}}^2 \leq R$,
2. in the spectral decomposition of $T = \sum_{n=1}^{\infty} \lambda_n \langle \cdot, e_n \rangle_{\mathcal{H}} \langle \cdot, e_n \rangle_{\mathcal{H}}$, where $(e_n)_{n=1}^{\infty}$ is a basis of $\text{Ker}(T)^{\perp}$, the eigenvalues of T satisfy $\alpha \leq n^b \lambda_n \leq \beta$ ($\forall n \geq 1$).

Remark 3 We make few remarks about the $\mathcal{P}(b, c)$ class:

- Range space assumption on f_ρ : The smoothness of f_ρ is expressed as a range space assumption, which is slightly different from the standard smoothness conditions appearing in non-parametric function estimation. By the spectral decomposition of T given above $[\lambda_1 \geq \lambda_2 \geq \dots > 0, \lim_{n \rightarrow \infty} \lambda_n = 0]$, $T^r u = \sum_{n=1}^{\infty} (\lambda_n)^r \langle u, e_n \rangle_{\mathcal{H}} e_n$ ($r = \frac{c-1}{2} \geq 0, u \in \mathcal{H}$) and

$$\text{Im}(T^r) = \left\{ \sum_{n=1}^{\infty} c_n e_n : \sum_{n=1}^{\infty} c_n^2 \lambda_n^{2r} < \infty \right\}. \quad (18)$$

Specifically, in the limit as $r \rightarrow 0$, we obtain $f_\rho \in \text{Im}(T^0) = \text{Im}(I) = \mathcal{H}$ (no constraint); larger values of r give rise to faster decay of the $(c_n)_{n=1}^{\infty}$ Fourier coefficients. This is the concrete meaning of $f_\rho \in \text{Im}(T^r)$.

- Spectral decay condition: We can provide a simple illustration of when the spectral decay conditions hold, in the event that the distributions are normal with means m_i and identical variance ($x_i = N(m_i, \sigma^2 I)$). When Gaussian kernels (k) are used with linear K , then $K(\mu_{x_i}, \mu_{x_j}) = e^{-c \|m_i - m_j\|^2}$ (Muandet et al., 2012, Table 1, line 2) (Gaussian, with arguments equal to the difference in means). Thus, this Gram matrix will correspond to the Gram matrix using a Gaussian kernel between points m_i . The spectral decay of the Gram matrix will correspond to that of the Gaussian kernel, with points drawn from the meta-distribution over the m_i . Thus, the source conditions are analysed in the same manner as for Gaussian Gram matrices: see e.g. Steinwart and Christmann (2008) for a discussion of these spectral decay properties.

In the $\mathcal{P}(b, c)$ family, the behaviour of $\mathcal{A}(\lambda)$, $\mathcal{B}(\lambda)$ and $\mathcal{N}(\lambda)$ is known: $\mathcal{A}(\lambda) \leq R\lambda^c$, $\mathcal{B}(\lambda) \leq R\lambda^{c-1}$, $\mathcal{N}(\lambda) \leq \beta \frac{b}{b-1} \lambda^{-\frac{1}{b}}$. Specializing Theorem 2 and retaining its assumptions, we get:

Theorem 4 (Finite-sample excess risk bound for $\rho \in \mathcal{P}(b, c)$)

Suppose the conditions in Theorem 2 hold. Let $\rho \in \mathcal{P}(b, c)$, where $1 < b$ and $c \in (1, 2]$.

Then

$$\begin{aligned} \mathcal{E}(f_{\frac{\lambda}{2}}, f_\rho) &\leq 5 \left\{ \frac{4L^2 (1 + \sqrt{\log(l) + \delta})^{2h} (2B_k)^h}{\lambda N^h} C^2 + 4B_K \times \right. \\ &\quad \times \left. C_\eta \left\{ \frac{2}{\lambda} \left[\frac{M^2 B_K}{l^2 \lambda} + \frac{\Sigma^2 \beta b}{(b-1)\lambda^{\frac{1}{b}}} \right] + \frac{3}{4\lambda^2} \left[\frac{4B_K^2 R \lambda^{c-1}}{l^2} + \frac{B_K R \lambda^c}{l} \right] + R \lambda^{c-1} + \|f_\rho\|_{5\epsilon}^2 \right\} \right. \\ &\quad \left. + R \lambda^c + C_\eta \left[\frac{B_K^2 R \lambda^{c-2}}{l^2} + \frac{B_K R \lambda^{c-1}}{4l} + \frac{B_K M^2}{l^2 \lambda} + \frac{\Sigma^2 \beta b}{(b-1)\lambda^{\frac{1}{b}}} \right] \right\}. \end{aligned}$$

Discarding the constants in Theorem 4, the study of convergence of the excess risk $\mathcal{E}(f_{\frac{\lambda}{2}}, f_\rho)$ to 0 boils down to finding N and λ (as a function of l) where $N \rightarrow \infty$, $\lambda \rightarrow 0$ and

$$r(l, N, \lambda) = \frac{\log^h(l)}{N^h \lambda} \left(\frac{1}{\lambda^2 l^2} + 1 + \frac{1}{\lambda^{1+\frac{1}{b}}} \right) + \lambda^c + \frac{1}{l^2 \lambda} + \frac{1}{\lambda^{\frac{b+1}{b}}} \rightarrow 0, \text{ s.t. } l \lambda^{\frac{b+1}{b}} \geq 1, \frac{\log(l)}{\lambda^{\frac{b}{b-1}}} \leq N \quad (19)$$

as $l \rightarrow \infty$. Let us choose $N = l^{\frac{b}{b-1}} \log(l)$; in this case Eq. (19) reduces to

$$r(l, \lambda) = \frac{1}{l^{2+a} \lambda^3} + \frac{1}{l^a \lambda} + \frac{1}{l^{a+1} \lambda^{2+\frac{1}{b}}} + \lambda^c + \frac{1}{l^2 \lambda} + \frac{1}{\lambda^{\frac{b+1}{b}}} \rightarrow 0, \text{ s.t. } l \lambda^{\frac{b+1}{b}} \geq 1, l^a \lambda^2 \geq 1. \quad (20)$$

One can assume that $a > 0$, otherwise $r(l, \lambda) \rightarrow 0$ fails to hold; in other words, N should grow faster than $\log(l)$. Matching the ‘bias’ (λ^c) and ‘variance’ (other terms in $r(l, \lambda)$) to choose λ , and guaranteeing that the matched terms dominate and the constraints in Eq. (20) hold, one gets the following simple description for the computational-statistical efficiency trade-off:⁸

Theorem 5 (Computational-statistical efficiency trade-off; well-specified case; $\rho \in \mathcal{P}(b, c)$) Suppose the conditions in Theorem 2 hold. Let $\rho \in \mathcal{P}(b, c)$ and $N = l^{\frac{b}{b-1}} \log(l)$, where $0 < a, 1 < b, c \in (1, 2]$. If

- $a \leq \frac{b(c+1)}{bc+1}$, then $\mathcal{E}(f_{\frac{\lambda}{2}}, f_\rho) = \mathcal{O}_p\left(l^{-\frac{ac}{c+1}}\right)$ with $\lambda = l^{-\frac{a}{c+1}}$,
- $a \geq \frac{b(c+1)}{bc+1}$ then $\mathcal{E}(f_{\frac{\lambda}{2}}, f_\rho) = \mathcal{O}_p\left(l^{-\frac{bc}{bc+1}}\right)$ with $\lambda = l^{-\frac{b}{bc+1}}$.

Remark 6 Theorem 5 formulates an exact computational-statistical efficiency trade-off for the choice of the bag size (N) as a function of the number of distributions (l) and problem difficulty (b, c).

- a -dependence: A smaller bag size (smaller a ; $N = l^{\frac{b}{b-1}} \log(l)$) means computational savings, but reduced statistical efficiency. It is not worth increasing a above $\frac{b(c+1)}{bc+1}$ since from that point the rate becomes $r(l) = l^{-\frac{bc}{bc+1}}$; remarkably, this rate is minimax in the one-stage sampled setup (Caponnetto and De Vito, 2007). The sensible choice $a = \frac{b(c+1)}{bc+1} < 2$ means that the one-stage sampled minimax rate can be achieved in the two-stage sampled setting with bag size N sub-quadratic in l .

⁸ The derivations are available in the supplement of <http://arxiv.org/pdf/1411.2066>.

- h -dependence: *In accord with our ‘smoothness’ assumptions it is rewarding to use smoother K kernels (larger $h \in (0, 1]$) since this reduces the bag size $[N = \lfloor h \log(l) \rfloor]$.*
- c -dependence: *The strictly decreasing property of $c \mapsto \frac{h(c+1)}{c+1}$ implies that for ‘smoother’ problems (larger c) fewer samples (N) are sufficient.*

Below we elaborate on the sketched high-level idea and prove Theorem 2.

Proof of Theorem 2 (detailed derivations of each step can be found in Section 7.1)

1. **Decomposition of the excess risk:** We have the following upper bound for the excess risk

$$\mathcal{E}(f_{\mathcal{X}}^{\lambda}, f_D) = \mathcal{R}[f_{\mathcal{X}}^{\lambda}] - \mathcal{R}[f_D] \leq 5[S_{-1} + S_0 + \mathcal{A}(\lambda) + S_1 + S_2]. \quad (21)$$

2. **It is sufficient to upper bound S_{-1} and S_0 :** Caponnetto and De Vito (2007) have shown that for $\forall \eta > 0$ if $l \geq 2C_{\eta} B_K \mathcal{N}(\lambda)/\lambda$, $\lambda \leq \|T\|_{\mathcal{L}(\mathcal{G})}$, then $\mathbb{P}(\Theta(\lambda, \mathbf{z}) \leq 1/2) \geq 1 - \eta/3$, where

$$\Theta(\lambda, \mathbf{z}) = \|(T - T_{\mathcal{X}})(T + \lambda I)^{-1}\|_{\mathcal{L}(\mathcal{G})}, \quad (22)$$

using which upper bounds on S_1 and S_2 that hold with probability $1 - \eta$ are obtained. It is known that $\mathcal{A}(\lambda) \leq R\lambda^c$.

3. **Probabilistic bounds on $\|g_{\mathcal{X}} - g_{\mathcal{X}}\|_{\mathcal{G}}^2$:** $\|T_{\mathbf{x}} - T_{\mathcal{X}}\|_{\mathcal{L}(\mathcal{G})}^2$, $\|\sqrt{T}(T_{\mathbf{x}} + \lambda I)^{-1}\|_{\mathcal{L}(\mathcal{G})}^2$, $\|f_{\mathcal{X}}^{\lambda}\|_{\mathcal{G}}^2$: One can bound S_{-1} and S_0 as

$$S_{-1} \leq \|\sqrt{T}(T_{\mathbf{x}} + \lambda I)^{-1}\|_{\mathcal{L}(\mathcal{G})}^2 \|g_{\mathcal{X}} - g_{\mathcal{X}}\|_{\mathcal{G}}^2$$

and

$$S_0 \leq \|\sqrt{T}(T_{\mathbf{x}} + \lambda I)^{-1}\|_{\mathcal{L}(\mathcal{G})}^2 \|T_{\mathbf{x}} - T_{\mathcal{X}}\|_{\mathcal{L}(\mathcal{G})}^2 \|f_{\mathcal{X}}^{\lambda}\|_{\mathcal{G}}^2$$

For the terms on the r.h.s., we derive upper bounds [for the definition of α , see Eq. (24)]

$$\|g_{\mathcal{X}} - g_{\mathcal{X}}\|_{\mathcal{G}}^2 \leq L^2 C^2 \frac{(1 + \sqrt{\alpha})^{2h} (2B_K)^h}{N^h}, \quad \|\sqrt{T}(T_{\mathbf{x}} + \lambda I)^{-1}\|_{\mathcal{L}(\mathcal{G})} \leq \frac{2}{\sqrt{\lambda}},$$

$$\|T_{\mathbf{x}} - T_{\mathcal{X}}\|_{\mathcal{L}(\mathcal{G})}^2 \leq \frac{(1 + \sqrt{\alpha})^{2h} (2B_K)^h B_K L^2}{N^h},$$

and

$$\|f_{\mathcal{X}}^{\lambda}\|_{\mathcal{G}}^2 \leq 6 \left(\frac{16}{\lambda} \log^2 \left(\frac{6}{\eta} \right) \left[\frac{M^2 B_K}{72\lambda} + \frac{\Sigma^2 \mathcal{N}(\lambda)}{l} \right] + \frac{4}{\lambda^2} \log^2 \left(\frac{6}{\eta} \right) \left[\frac{4B_K^2 \mathcal{B}(\lambda)}{l^2} + \frac{B_K \mathcal{A}(\lambda)}{l} \right] + \mathcal{B}(\lambda) + \|f_D\|_{\mathcal{G}}^2 \right). \quad (23)$$

The bounds hold under the following conditions:

- $\|g_{\mathcal{X}} - g_{\mathcal{X}}\|_{\mathcal{G}}^2$ (see Section 7.1.1): if the empirical mean embeddings are close to their population counterparts, i.e.,

$$\|\mu_{x_i} - \mu_{\mathcal{X}}\|_H \leq \frac{(1 + \sqrt{\alpha})\sqrt{2}B_K}{\sqrt{N}}, \quad (\forall i = 1, \dots, l). \quad (24)$$

This event has probability $1 - le^{-\alpha}$ over all $i = 1, \dots, l$ samples; see (Altmun and Smola, 2006) and (Szabó et al., 2015, Section A.1.10).

- $\|T_{\mathbf{x}} - T_{\mathcal{X}}\|_{\mathcal{L}(\mathcal{G})}^2$ (see Section 7.1.2): (24) is assumed.
- $\|\sqrt{T}(T_{\mathbf{x}} + \lambda I)^{-1}\|_{\mathcal{L}(\mathcal{G})}^2$ (Szabó et al., 2015, Section A.1.11): (24), $\Theta(\lambda, \mathbf{z}) \leq \frac{1}{2}$, and

$$\frac{(1 + \sqrt{\alpha})^2 2^{\frac{h+6}{h}} B_K (B_K)^{\frac{1}{h}} L^{\frac{6}{h}}}{\lambda^{\frac{6}{h}}} \leq N. \quad (25)$$

- $\|f_{\mathcal{X}}^{\lambda}\|_{\mathcal{G}}^2$: The bound is guaranteed to hold under the conditions of the bounds of S_1 and S_2 .⁸

4. **Union bound:** By applying an $\alpha = \log(l) + \delta$ reparameterization, and combining the received upper bounds with Caponnetto and De Vito (2007)’s results for S_1 and S_2 , Theorem 2 follows (Section 7.1.3) with a union bound.

Finally, we note that existing results/ideas were used at two points to simplify our analysis: bounding S_1 , S_2 , $\Theta(\lambda, \mathbf{z})$, $\|f_{\mathcal{X}}^{\lambda}\|_{\mathcal{G}}^2$ (Caponnetto and De Vito, 2007) and $\|\mu_{x_i} - \mu_{\mathcal{X}}\|_H$ (Altmun and Smola, 2006).⁹

4.2 Results for the Misspecified Case

In this section, we focus on the misspecified case ($f_D \in L_{\rho_{\mathcal{X}}}^2 \setminus \mathcal{F}$) and present our second main result, which was inspired by the proof technique of Stripunburd et al. (2014, Theorem 12). We derive a high probability upper bound for $\mathcal{E}(f_{\mathcal{X}}^{\lambda}, f_D)$, i.e., the excess risk of the MERR method (Theorem 7) which gives rise to consistency results (3rd bullet of Remark 8) and precise computational-statistical efficiency trade-off (Theorem 9). Theorem 7 consists of two finite-sample bounds:

1. The first, more general bound [Eq. (27)] will be used to show consistency in the misspecified case (see the 3rd bullet of Remark 8), in other words that $\mathcal{E}(f_{\mathcal{X}}^{\lambda}, f_D)$ can be driven to its smallest possible value determined by the ‘richness’ of \mathcal{F} :

$$D_{\mathcal{X}}^{\lambda} := \inf_{q \in \mathcal{F}} \|f_D - S_K^* q\|_{\rho}^2. \quad (26)$$

The value of $D_{\mathcal{X}}$ equals the approximation error of f_D by a function from \mathcal{F} . Specifically, if $\mathcal{F} = \{S_K^* q : q \in \mathcal{F}\} \subseteq L_{\rho_{\mathcal{X}}}^2$ is dense in $L_{\rho_{\mathcal{X}}}^2$, then $D_{\mathcal{X}} = 0$.

⁸ We also corrected some constants in the previous works (Altmun and Smola, 2006; Caponnetto and De Vito, 2007).

2. The second, specialized result [Eq. (28)] under additional smoothness assumptions on f_ρ will give rise to a precise computational-statistical efficiency trade-off in terms of the problem difficulty (s) and sample numbers (l, N); this result can be seen as the misspecified analogue of Theorem 5.

After stating our results, the main ideas of the proof follow; further technical details are available in Section 7.2. Our main theorem for bounding the excess risk is as follows:

Theorem 7 (Finite-sample excess risk bounds; misspecified case) Let $l \in \mathbb{Z}^+$, $N \in \mathbb{Z}^+$, $0 < \lambda, 0 < \eta < 1$, $0 < \delta$ and $C_\eta = \log\left(\frac{6}{\eta}\right)$. Assume that $\left(\frac{12B_K C_\eta}{\lambda}\right)^2 \leq l$ and $(1 + \sqrt{\log(l) + \delta})^2 2^{\frac{h+\delta}{\eta}} B_k (B_K)^{\frac{1}{2}} L^{\frac{2}{\eta}} / \lambda^{\frac{2}{\eta}} \leq N$.

1. Then for arbitrary $q \in \mathcal{H}$ with probability at least $1 - \eta - e^{-\delta}$

$$\begin{aligned} \sqrt{\mathcal{E}(f_{\frac{1}{2}}^\lambda, f_\rho)} &\leq \frac{2LC \left(1 + \sqrt{\log(l) + \delta}\right)^h (2B_k)^{\frac{1}{2}}}{\sqrt{\lambda N^{\frac{h}{2}}}} \left(1 + \frac{2\sqrt{B_K}}{\sqrt{\lambda}}\right) + \\ &\frac{2C_\eta}{\sqrt{\lambda}} \left\{ \left(\frac{2C\sqrt{B_K}}{l} + \frac{C\sqrt{B_K}}{\sqrt{l}}\right) + \left(\frac{2B_K}{l} + \frac{\sigma}{\sqrt{l}}\right) \frac{1}{\lambda\sqrt{\lambda}} \|f_\rho\|_\rho D_a(\lambda, q) \right\} + D_a(\lambda, q), \end{aligned} \quad (27)$$

where $D_a(\lambda, q) = \|f_\rho - S_K^* q\|_\rho + \max(1, \|\tilde{T}\|_{\mathcal{L}(L^2(\mathcal{H}))}) \lambda^{\frac{1}{2}} \|q\|_{\mathcal{H}}$.

2. In addition, suppose $f_\rho \in \text{Im}(\tilde{T}^s)$ for some $s > 0$, where \tilde{T} is defined in Eq. (6). Then with probability at least $1 - \eta - e^{-\delta}$, we have

$$\begin{aligned} \sqrt{\mathcal{E}(f_{\frac{1}{2}}^\lambda, f_\rho)} &\leq \frac{2LC \left(1 + \sqrt{\log(l) + \delta}\right)^h (2B_k)^{\frac{1}{2}}}{\sqrt{\lambda N^{\frac{h}{2}}}} \left(1 + \frac{2\sqrt{B_K}}{\sqrt{\lambda}}\right) + \\ &\frac{2C_\eta}{\sqrt{\lambda}} \left\{ \left(\frac{2C\sqrt{B_K}}{l} + \frac{C\sqrt{B_K}}{\sqrt{l}}\right) + \left(\frac{2B_K}{l} + \frac{\sigma}{\sqrt{l}}\right) \frac{1}{\lambda} \times \right. \\ &\left. \sqrt{\max\left(1, \|\tilde{T}\|_{\mathcal{L}(L^2(\mathcal{H}))}^s\right) \lambda} \|\tilde{T}^{-s} f_\rho\|_\rho D_b(\lambda, s) \right\} + D_b(\lambda, s), \end{aligned} \quad (28)$$

where $D_b(\lambda, s) = \max(1, \|\tilde{T}\|_{\mathcal{L}(L^2(\mathcal{H}))}^{s-1}) \lambda^{\min(1,s)} \|\tilde{T}^{-s} f_\rho\|_\rho$.

Remark 8 We give a short insight into the assumptions of Theorem 7, followed by consequences of the theorem.

- Range space assumption on f_ρ : The range space assumption for the compact, positive, self-adjoint operator, $\tilde{T} = \tilde{T}(K) : L_{\rho_X}^2 \rightarrow L_{\rho_X}^2$ in the 2nd part of Theorem 7 can be interpreted similarly to that on T ; see Eq. (18). One can also prove alternative descriptions for $\text{Im}(\tilde{T}^s)$ in terms of interpolation spaces (Steinwart and Scovel, 2012, Theorem 4.6, page 387), or the decay of the 2-approximation error function, $A_2(\lambda) = \inf_{f \in \mathcal{H}(K)} (\lambda \|f\|_{\mathcal{H}(K)}^2 + \mathcal{R}[f] - \mathcal{R}[f_\rho])$ (Smale and Zhou, 2003; Steinwart et al., 2009).

- $\sqrt{\mathcal{E}(f_{\frac{1}{2}}^\lambda, f_\rho)}$: Notice that in the bounds [(27), (28)], instead of the excess risk, its square root appears; this has technical reasons, as it is easier to have the $D_a(\lambda, q)$ quantity (without multiplicative constants) appear on the r.h.s. of Eq. (27) with this form.

- Consistency in the misspecified case: The consequence of Theorem 7(1) is as follows. Discarding the constants in Eq. (27), we obtain the upper bound (notice that the constant multiplier of $\|f_\rho - S_K^* q\|_\rho$ in the last term was one):

$$\sqrt{r(l, N, \lambda, q)} = \frac{\log^{\frac{h}{2}}(l)}{N^{\frac{h}{2}} \lambda} + \frac{1}{\sqrt{\lambda}} + \frac{\sqrt{\|f_\rho - S_K^* q\|_\rho + \sqrt{\lambda} \|q\|_{\mathcal{H}}}}{\lambda \sqrt{l}} + \|f_\rho - S_K^* q\|_\rho + \sqrt{\lambda} \|q\|_{\mathcal{H}}.$$

By choosing $N = l^{1/h} \log l$, $\sqrt{r(l, \lambda)}$ is bounded by

$$\inf_{q \in \mathcal{H}} \left\{ \|f_\rho - S_K^* q\|_\rho + \frac{\sqrt{\|f_\rho - S_K^* q\|_\rho + \sqrt{\lambda} \|q\|_{\mathcal{H}}}}{\lambda \sqrt{l}} + \sqrt{\lambda} \|q\|_{\mathcal{H}} \right\} + \mathcal{O}_p\left(\frac{1}{\sqrt{\lambda l}}\right).$$

Our goal is to investigate the behavior of the bound as $l \rightarrow \infty$, $\lambda \rightarrow 0$ and $\lambda \sqrt{l} \rightarrow \infty$. Define $K(\alpha, \beta, \gamma) := \inf_{q \in \mathcal{H}} \left\{ \|f_\rho - S_K^* q\|_\rho + \alpha \sqrt{\|f_\rho - S_K^* q\|_\rho + \beta \sqrt{\|q\|_{\mathcal{H}} + \gamma} \|q\|_{\mathcal{H}}} \right\}$. $K(\alpha, \beta, \gamma)$ is the pointwise infimum of affine functions, therefore it is upper semi-continuous and concave on $\mathbb{R}_{>0}^3$ (Aiprantis and Border, 2006, Lemmas 2.41 and 5.40); it is continuous on $\times_{s=1}^3 \mathbb{R}_{>0}$ (Rockafellar and Wets, 2008, Theorem 2.35). Moreover, by applying (Rockafellar and Wets, 2008, Corollary 2.37) it extends continuously to $\times_{s=1}^3 \mathbb{R}_{\geq 0}$; specifically it is continuous at $(\alpha, \beta, \gamma) = \mathbf{0}$. In other words, as $l \rightarrow \infty$, $\lambda \rightarrow 0$ and $\lambda \sqrt{l} \rightarrow \infty$, $K\left(\frac{1}{\lambda l}, \frac{1}{\lambda l \sqrt{l}}, \sqrt{\lambda}\right) \rightarrow D_{\mathcal{H}}$ and we get consistency in the misspecified case,¹⁰

$$\sqrt{r(N, l, \lambda)} \rightarrow D_{\mathcal{H}}.$$

Discarding the constants in Eq. (28) we get¹⁰

$$\sqrt{r(l, N, \lambda)} = \frac{\log^{\frac{h}{2}}(l)}{N^{\frac{h}{2}} \lambda} + \frac{1}{\sqrt{l\lambda}} + \frac{\sqrt{\lambda^{\min(1,s)}}}{\lambda \sqrt{l}} + \lambda^{\min(1,s)}, \text{ subject to } \frac{1}{\lambda^2} \leq l. \quad (29)$$

Our goal is to drive $r(l, N, \lambda)$ to zero with a suitable choice of the (l, N, λ) triplet under the stronger range space assumption. Since in Eq. (29) $\min(1, s)$ appears, one can assume without loss of generality that $s \in (0, 1]$; consequently $1 - \frac{s}{2} \in [\frac{1}{2}, 1)$ and $\frac{1}{l^{\frac{1}{2} - \frac{s}{2}}} \leq \frac{1}{\lambda^{1 - \frac{s}{2}}}$. Let us choose $N = l^{2s/h} \log(l)$; in this case using the previous dominance note, Eq. (29) reduces to the study of

$$\sqrt{r(l, \lambda)} = \frac{1}{l^{\alpha\lambda}} + \frac{1}{\lambda^{1 - \frac{s}{2} l^{\frac{1}{2}}}} + \lambda^s \rightarrow 0, \text{ s.t. } l\lambda^2 \geq 1. \quad (30)$$

One can assume that $\alpha > 0$, otherwise $r(l, \lambda) \rightarrow 0$ fails to hold: in other words, N should grow faster than $\log(l)$. Matching the ‘bias’ (λ^s) and ‘variance’ (other) terms in $r(l, \lambda)$ to choose λ , guaranteeing that the matched terms dominate and the constraint in Eq. (30) hold, one can arrive at the following computational-statistical efficiency trade-off.⁸

¹⁰ We have discarded the $\log(l)/\lambda^{\frac{s}{2}} \leq N$ constraint implied by the convergence of the first term in \sqrt{r} .

Theorem 9 (Computational-statistical efficiency trade-off; misspecified case, $f_\rho \in \text{Im}(T^s)$) Suppose that $f_\rho \in \text{Im}(T^s)$ and $N = l^{\frac{2a}{s}} \log(l)$, where $s \in (0, 1]$, $a > 0$. If

- $a \leq \frac{s+1}{s+2}$, then $\mathcal{E}(f_\frac{\lambda}{2}, f_\rho) = \mathcal{O}_p\left(l^{-\frac{2as}{s+1}}\right)$ with $\lambda = l^{-\frac{a}{s+1}}$,
- $a \geq \frac{s+1}{s+2}$, then $\mathcal{E}(f_\frac{\lambda}{2}, f_\rho) = \mathcal{O}_p\left(l^{-\frac{2a}{s+2}}\right)$ with $\lambda = l^{-\frac{1}{s+2}}$.

Remark 10 Theorem 9 provides a complete computational-statistical efficiency trade-off description for the choice of the bag size (N) as a number of the distributions (1).

- *a-dependence:* A smaller value of ‘ a ’ (smaller bags $N = \lceil 2a/h \log(l) \rceil$) leads to a computational advantage, but one loses in statistical efficiency. As ‘ a ’ reaches $\frac{s+1}{s+2}$, the rate becomes $r(l) = l^{-\frac{2a}{s+2}}$ and one does not gain from further increasing the value of a . The sensible choice of $a = \frac{s+1}{s+2} \leq \frac{2}{3}$ means that N can again be sub-quadratic ($2a < \frac{4}{3} < 2$) in l .
 - *h-dependence:* By using smoother K kernels (larger $h \in (0, 1]$) one can reduce the size of the bags: $h \mapsto 2a/h$ is decreasing in h . This is compatible with our smoothness requirement on f_ρ .
 - *s-dependence:* ‘Easier’ tasks (larger s) give rise to faster convergence. Indeed, in the $r(l) = l^{-\frac{2a}{s+2}}$ rate the $s \mapsto \frac{2a}{s+2}$ exponent is strictly increasing function of the problem difficulty (s). For example, for extremely non-smooth regression problems ($s \approx 0$) the convergence can be arbitrary slow ($\lim_{s \rightarrow 0} \frac{2a}{s+2} = 0$). In the smooth case ($s = 1$) $\lim_{s \rightarrow 1} \frac{2a}{s+2} = \frac{2}{3}$ and one can achieve the $r(l) = l^{-\frac{2}{3}}$ rate.
 - We may compare our $r(l) = l^{-\frac{2a}{s+2}}$ result with the $r_\rho(l) = l^{-\frac{2a}{2s+1}}$ (one-stage sampled) rate (Steinwart et al., 2009, $\beta/2 := s, q := 2, p := 1$ in Corollary 6), which was shown to be asymptotically optimal on $Y = \mathbb{R}$ for continuous k on compact metric \mathcal{X} . Steinwart et al.’s result is more general in terms of q ($\|f\|_{\mathcal{X}}^q$ based regularization) and p ($\|f\|_\infty \leq C \|f\|_{\mathcal{X}}^p \|f\|_\rho^{1-p}$, $\forall f \in \mathcal{H}$; in our case $p = 1$), although it imposes an additional eigenvalue constraint [(Steinwart et al., 2009, Eq. (6))] as well as $f_\rho \in \text{Im}(T^s)$. Moreover, one can observe that $r_\rho(l) \leq r(l)$ with a small gap, and that for $s \rightarrow 0$ and $s = 1$, $r_\rho(l) = r(l)$; see Fig. 1. We further remind the reader that our MERR analysis also holds for separable Hilbert output spaces Y , separable topological domains \mathcal{X} enriched with a bounded, continuous kernel k , and that we handle the two-stage sampled setting.
- The main steps of the proof of Theorem 7 are as follows:

Proof of Theorem 7 (the details of the derivation are available in Section 7.2) Steps 1-7 will be identical in both proofs,¹¹ and we present them jointly.

1. **Decomposition of the excess risk:** By the triangle inequality, we have

$$\sqrt{\mathcal{E}(f_\frac{\lambda}{2}, f_\rho)} = \|S_K^* f_\frac{\lambda}{2} - f_\rho\|_\rho \leq \|S_K^* (f_\frac{\lambda}{2} - f_\rho)\|_\rho + \|S_K^* f_\rho - f_\rho\|_\rho. \quad (31)$$

11. Importantly, with a slight modification of the more general, first part of Theorem 7, one can get the specialized second setting of the theorem (see Step 8).

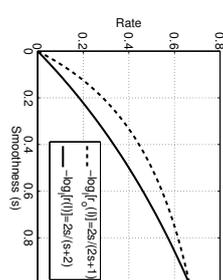


Figure 1: Comparison of the $r_\rho(l) = l^{-\frac{2a}{2s+1}}$ and $r(l) = l^{-\frac{2a}{s}}$ rates as function of the problem difficulty/smoothness (s).

2. **Bound on $\|S_K^*(f_\frac{\lambda}{2} - f_\rho)\|_\rho$:** Using¹² the fact that

$$\|S_K^* h\|_\rho^2 = \|\sqrt{T}h\|_{\mathcal{H}}^2 \quad (\forall h \in \mathcal{H}), \quad (32)$$

and the definitions of S_{-1} and S_0 [see Eqs. (15)-(16)], we obtain

$$\|S_K^*(f_\frac{\lambda}{2} - f_\rho)\|_\rho = \|\sqrt{T}(f_\frac{\lambda}{2} - f_\rho)\|_{\mathcal{H}} \leq \sqrt{S_{-1}} + \sqrt{S_0}, \quad (33)$$

through an application of triangle inequality. One can derive without a $\mathcal{P}(b, c)$ prior assumption (Section 7.2.1) the upper bound¹³

$$\sqrt{S_{-1}} + \sqrt{S_0} \leq \frac{2lC(1 + \sqrt{\alpha})^h (2B_k)^{\frac{h}{2}}}{\sqrt{\lambda} N^{\frac{h}{2}}} \left[1 + \frac{2\sqrt{BK}}{\sqrt{\lambda}} \right]$$

for the r.h.s. of Eq. (33) under the conditions that $\Theta(\lambda, \mathbf{z}) \leq \frac{1}{2}$ (which holds with probability $1 - \eta$ if $[2BK \log(2/\eta)]/\lambda^2 \leq l$), and that Eqs. (24)-(25) hold.

3. **Decomposition of $\|S_K^* f_\frac{\lambda}{2} - f_\rho\|_\rho$:** By the triangle inequality and Eq. (32), we have

$$\begin{aligned} \|S_K^* f_\frac{\lambda}{2} - f_\rho\|_\rho &= \|S_K^*(f_\frac{\lambda}{2} - f^\lambda) + S_K^* f^\lambda - f_\rho\|_\rho \leq \|S_K^*(f_\frac{\lambda}{2} - f^\lambda)\|_\rho + \|S_K^* f^\lambda - f_\rho\|_\rho \\ &= \|\sqrt{T}(f_\frac{\lambda}{2} - f^\lambda)\|_{\mathcal{H}} + \|S_K^* f^\lambda - f_\rho\|_\rho. \end{aligned} \quad (34)$$

4. **Decomposition of $\|\sqrt{T}(f_\frac{\lambda}{2} - f^\lambda)\|_{\mathcal{H}}$:** Making use of the analytical expressions for $f_\frac{\lambda}{2}^\lambda$ and f^λ [see Eq. (13) and Eq. (17)], and the operator Woodbury formula (Ding and Zhou, 2008, Theorem 2.1, page 724) we arrive at the decomposition (see Section 7.2.2)

$$\begin{aligned} \|\sqrt{T}(f_\frac{\lambda}{2} - f^\lambda)\|_{\mathcal{H}} &\leq \|\sqrt{T}(T_{\mathbf{x}} + \lambda I)^{-1}\|_{\mathcal{L}(\mathcal{H})} \left(\|g_{\mathbf{x}} - g_\rho\|_{\mathcal{H}} + \right. \\ &\quad \left. \|T - T_{\mathbf{x}}\|_{\mathcal{L}(\mathcal{H})} \lambda^{-1} \|S_K[f_\rho - (T + \lambda I)^{-1} S_K^* S_K f_\rho]\|_{\mathcal{H}} \right), \end{aligned}$$

where $g_\rho = S_K f_\rho$. As it is known (Caporineto and De Vito, 2007, page 348) $\|\sqrt{T}T_{\mathbf{x}} + \lambda I\|_{\mathcal{L}(\mathcal{H})} \leq 1/\sqrt{\lambda}$ provided that $\Theta(\lambda, \mathbf{z}) \leq \frac{1}{2}$.

12. See for example de Vito et al. (2006) on page 88 with the $(T, S, A, T) := (T, T_\rho^2, S_K^*, T)$ choice.

13. See the remark at the end of Section 7.2.1.

5. **Bound on $\|g_{\mathbf{z}} - g_{\rho}\|_{\mathcal{H}^c}$:** $\|T - T_{\mathbf{x}}\|_{\mathcal{L}(\mathcal{H}^c)}$: By concentration arguments the bounds

$$\|g_{\mathbf{z}} - g_{\rho}\|_{\mathcal{H}^c} \leq \left(\frac{4C\sqrt{B_K}}{l} + \frac{2C\sqrt{B_K}}{\sqrt{l}} \right) \log\left(\frac{2}{\eta}\right), \quad \|T - T_{\mathbf{x}}\|_{\mathcal{L}(\mathcal{H}^c)} \leq \left(\frac{4B_K}{l} + \frac{4\sigma}{\sqrt{l}} \right) \log\left(\frac{2}{\eta}\right)$$

hold with probability at least $1 - \eta$, each (see Section 7.2.3, 7.2.4).

6. **Decomposition of $\|S_K[f_{\rho} - (\tilde{T} + \lambda I)^{-1} S_K^* S_K f_{\rho}]\|_{\mathcal{H}^c}^2$:** Exploiting the analytical formula for f^{λ} , one can construct (Section 7.2.5) the upper bound

$$\|S_K[f_{\rho} - (\tilde{T} + \lambda I)^{-1} S_K^* S_K f_{\rho}]\|_{\mathcal{H}^c}^2 \leq \|\tilde{T}[f_{\rho} - (\tilde{T} + \lambda I)^{-1} S_K^* S_K f_{\rho}]\|_{\rho} \|S_K^* f^{\lambda} - f_{\rho}\|_{\rho}.$$

7. **Bound on $\|\tilde{T}[f_{\rho} - (\tilde{T} + \lambda I)^{-1} S_K^* S_K f_{\rho}]\|_{\rho}$:** Using our assumptions that $f_{\rho} \in \text{Im}(\tilde{T}^s)$ ($s \geq 0$)¹⁴ and exploiting the separability of $L_{\rho_X}^2$, by Lemma 7.3.2 ($\mathcal{K} = L_{\rho_X}^2$, $f = f_{\rho}$, $M = \tilde{T}$, $a = 1$) and $\tilde{T} = S_K^* S_K$ we obtain the upper bound

$$\begin{aligned} \|\tilde{T}[f_{\rho} - (\tilde{T} + \lambda I)^{-1} S_K^* S_K f_{\rho}]\|_{\rho} &= \|\tilde{T}[f_{\rho} - (\tilde{T} + \lambda I)^{-1} \tilde{T} f_{\rho}]\|_{\rho} \\ &\leq \max\left(1, \|\tilde{T}\|_{\mathcal{L}(L_{\rho_X}^2)}^s\right) \lambda^{\min(1, s+1)} \|\tilde{T}^{-s} f_{\rho}\|_{\rho} = \max\left(1, \|\tilde{T}\|_{\mathcal{L}(L_{\rho_X}^2)}^s\right) \lambda \|\tilde{T}^{-s} f_{\rho}\|_{\rho}, \end{aligned}$$

where we used at the last step that $\min(1, s+1) = 1$; this follows from $s \geq 0$.

8. **Bound on $\|S_K^* f^{\lambda} - f_{\rho}\|_{\rho}$:**

(a) **No range space assumption:** One can construct (Section 7.2.6) the bound

$$\|S_K^* f^{\lambda} - f_{\rho}\|_{\rho} \leq \|f_{\rho} - S_K^* q\|_{\rho} + \max\left(1, \|\tilde{T}\|_{\mathcal{L}(\mathcal{H}^c)}\right) \lambda^{\frac{1}{2}} \|q\|_{\mathcal{H}^c},$$

which holds for arbitrary $q \in \mathcal{H}$.

(b) **Range space assumption in $L_{\rho_X}^2$:** Using the $S_K^* f^{\lambda} = (\tilde{T} + \lambda I)^{-1} \tilde{T} f_{\rho}$ identity [see Eq. (43)], and Lemma 7.3.2 ($M = \tilde{T}$, $\mathcal{K} = L_{\rho_X}^2$, $a = 0$), we get

$$\|S_K^* f^{\lambda} - f_{\rho}\|_{\rho} = \|(\tilde{T} + \lambda I)^{-1} \tilde{T} f_{\rho} - f_{\rho}\|_{\rho} \leq \max\left(1, \|\tilde{T}\|_{\mathcal{L}(L_{\rho_X}^2)}^{s-1}\right) \lambda^{\min(1, s)} \|\tilde{T}^{-s} f_{\rho}\|_{\rho}.$$

9. **Union bound:** Applying an $\alpha = \log(l) + \delta$ reparameterization, changing η to $\frac{\eta}{l}$ and combining the derived results (in case of the first statement with $s = 0$) with a union bound, Theorem 7 follows.

Remark 11 *To contrast the derivation of the well- and the misspecified cases, we note that previous results [Section 4.1, or Caponnetto and De Vito (2007)'s bound] were used at two points:*

(a) *In Step 2 by using Eq. (32) and transforming the $L_{\rho_X}^2$ error $\|S_K^*(f_{\frac{\mathbf{z}}{2}} - f^{\lambda})\|_{\rho}$ to \mathcal{H} , we could rely on our previous bounds for S_{-1} and S_0 . However, we were required to use a different concentration argument to guarantee $\Theta(\lambda, \mathbf{z}) \leq \frac{1}{2}$ since we no longer assume the $\mathcal{P}(b, c)$ prior class.*

¹⁴. Note that we choose $s = 0$ and $s > 0$ in the first and second theorem part, respectively.

(b) *In Step 4 the first term could be bounded by Caponnetto and De Vito (2007). Its $\Theta(\lambda, \mathbf{z}) \leq \frac{1}{2}$ condition was guaranteed by Step 2; and see Section 7.2.1.*

We note that our misspecified proof method was inspired by Sriperumbudur et al. (2014, Theorem 12), where the authors focused on the consistency of an infinite-dimensional exponential family estimator.

5. Related Work

In this section we discuss existing approaches and heuristic techniques to tackle learning problems on distributions.

Methods based on parametric assumptions: A number of methods have been proposed to compute the similarity of distributions or bags of samples. As a first approach, one could fit a parametric model to the bags, and estimate the similarity of the bags based on the obtained parameters. It is then possible to define learning algorithms on the basis of these similarities, which often take analytical form. Typical examples with explicit formulas include Gaussians, finite mixtures of Gaussians, and distributions from the exponential family (with known log-normalizer function and zero carrier measure, see Kondor and Jebara, 2003; Jebara et al., 2004; Wang et al., 2009; Nielsen and Nock, 2012). A major limitation of these methods, however, is that they apply quite simple parametric assumptions, which may not be sufficient or verifiable in practise.

Methods based on parametric assumption in a RKHS: A heuristic related to the parametric approach is to assume that the training distributions are Gaussians in a reproducing kernel Hilbert space (see for example Jebara et al., 2004; Zhou and Chellappa, 2006, and references therein). This assumption is algorithmically appealing, as many divergence measures for Gaussians can be computed in closed form using only inner products, making them straightforward to kernelize. A fundamental shortfall of kernelized Gaussian divergences is the lack of their consistency analysis in specific learning algorithms.

Kernels based techniques: A more theoretically grounded approach to learning on distributions has been to define positive definite kernels on the basis of statistical divergence measures on distributions, or by metrics on non-negative numbers; these can then be used in kernel algorithms. This category includes work on semigroup kernels (Cuturi et al., 2005), non-extensive information theoretical kernel constructions (Martins et al., 2009), and kernels based on Hilbertian metrics (Hein and Bousquet, 2005). For example, the intuition of semigroup kernels (Cuturi et al., 2005) is as follows: if two measures or sets of points overlap, then their sum is expected to be more concentrated. The value of dispersion can be measured by entropy or inverse generalized variance. In the second type of approach (Hein and Bousquet, 2005), homogeneous Hilbert metrics on the non-negative real line are used to define the similarity of probability distributions. While these techniques guarantee to provide valid kernels on certain restricted domains of measures, the performance of learning algorithms based on finite-sample estimates of these kernels remains a challenging open question. One might also plug into learning algorithms (based on similarities of distributions) consistent Rényi and Tsallis divergence estimates (Póczos et al., 2011, 2012), but these similarity indices are *not* kernels, and their consistency in specific learning tasks remains an open question.

Multi-instance learning: An alternative paradigm in learning when the inputs are “bags of objects” is to simply treat each input as a *finite set*: this leads to the multi-instance learning task (MIL, see Dietterich et al., 1997; Ray and Page, 2001; Dooly et al., 2002). In MIL one is given a set of labelled bags, and the task of the learner is to find the mapping from the bags to the labels. Many important examples fit into the MIL framework: for example, different configurations of a given molecule can be handled as a bag of shapes, images can be considered as a set of patches or regions of interest, a video can be seen as a collection of images, a document might be described as a bag of words or paragraphs, a web page can be identified by its links, a group of people on a social network can be captured by their friendship graphs, in a biological experiment a subject can be identified by his/her time series trials, or a customer might be characterized by his/her shopping records. The MIL approach has been applied in several domains: see the reviews from Babenko (2004); Zhou (2004); Foulds and Frank (2010); Amores (2013).

“Bag-of-objects” methods (MIL, classification): Despite the large number of MIL applications and the spate of heuristic solution techniques, there exist few *theoretical results* in the area (Auer, 1998; Long and Tan, 1998; Blum and Kalai, 1998; Babenko et al., 2011; Zhang et al., 2013; Sabato and Tishby, 2012) and they focus on the multi-instance *classification* (MIC) task. In particular, let us first consider the standard MIC assumption (Dietterich et al., 1997), where a bag is declared to be positive (labelled with “1”) if at least one of its instances is positive (“1”); otherwise, the bag is negative (“0”).¹⁵ In other words, if the instances $(x_{i,n})$ in the i^{th} bag $\{x_{i,1}, \dots, x_{i,N}\}$ have hidden label $L(x_{i,n}) \in \{0, 1\}$, then the observed label of the bag is $y_i = h(x_{i,1}, \dots, x_{i,N}) = \max\{L(x_{i,1}), \dots, L(x_{i,N})\} \in \{0, 1\}$. In case of the original APR (axis-aligned rectangles; Dietterich et al., 1997) hypothesis class, function L is equal to the indicator of an unknown rectangle R ($L = \mathbb{1}_R$). In other words, a bag is declared to be positive if there exists at least one instance in the bag, which belongs to R .¹⁶ The goal is to learn R with high probability given the bags $(\{x_{i,1}, \dots, x_{i,N}\}$ -s) and their labels $(y_i$ -s). Long and Tan (1998) proved the PAC learnability (probably approximately correct; Valiant, 1984) of the APR hypothesis class, if all instances in each bag are i.i.d. and follow the same product distribution over the instance coordinates. On the other hand, for arbitrary distributions over bags, when the instances within a bag might be statistically dependent, APR learning under MIC is NP-hard (Auer, 1998); the same authors also showed that the product property (Long and Tan, 1998) on the coordinates is not required to obtain PAC results. Blum and Kalai (1998) extended PAC learnability of APRs to hypothesis classes learnable from one-sided classification noise. In contrast to the previous approaches (Long and Tan, 1998; Auer, 1998; Blum and Kalai, 1998), Babenko et al. (2011) modelled the bags as low-dimensional manifolds, and proved PAC bounds. By relaxing the standard MIC assumption, Sabato and Tishby (2012) showed PAC-learnability for general MIC hypothesis classes with extended “max” functions. Zhang et al. (2013) derived high-probability generalization bounds in the MIC setting, when local and global representations are combined. Our work falls outside this setting since the label and bag generation mechanisms we consider are different: we do not assume an exact form of the

labelling mechanism (function L and \max in h). Rather, the labelling is presumed to be stochastically determined by the underlying true distribution, not deterministically by the instance realizations in the bags (these are presumed i.i.d., and may be bag-specific).

“Bag-of-objects” methods (MIL, not classification): Beyond classification, there exist several *heuristics*—without consistency guarantees—for many other multi-instance problems in the literature, including regression (Ray and Page, 2001; Dooly et al., 2002; Zhou et al., 2009; Kwok and Cheung, 2007), clustering (Zhang and Zhou, 2009; Zhang et al., 2009, 2011; Chen and Wu, 2012), ranking (Bergeron et al., 2008; Hu et al., 2008; Bergeron et al., 2012), outlier detection (Wu et al., 2010), transfer learning (Raykar et al., 2008; Zhang and Si, 2009), and feature selection, -weighting and -extraction (also called dimensionality reduction, low-dimensional embedding, manifold learning; see Raykar et al., 2008; Pring et al., 2010; Sun et al., 2010; Carter et al., 2011; Zafra et al., 2013; Chai et al., 2014a,b, and references therein).

Approaches using set metrics: Adapting the bag viewpoint of MIL, one can come up with set metric based learning algorithms.¹⁷ Probably one of the most well-known set metrics is the Hausdorff metric (Edgar, 1995), which is defined for non-empty compact sets of metric spaces, specifically for sets containing finitely many points. There also exist other (semi)metric constructions on points sets (Eiter and Mannila, 1997; Ramon and Brynnoegle, 2001). Unfortunately, the classical Hausdorff metric is highly sensitive to outliers, seriously limiting its practical applicability. In order to mitigate this deficiency, several variants of the Hausdorff metric have been designed in the MIL literature, such as the maximal-, the minimal- and the ranked Hausdorff metrics, with successful applications in MIC (Wang and Zuecker, 2000) and multi-instance outlier detection (Wu et al., 2010); and the average Hausdorff metric (Zhang and Zhou, 2009) and contextual Hausdorff dissimilarity (Chen and Wu, 2012), which have been found useful in multi-instance clustering. Unfortunately, these methods lack any theoretical guarantee when applied in specific learning problems.

Functional data analysis techniques: Finally, the distribution regression task might also be interpreted as a functional data analysis problem (Ramsay and Silverman, 2002, 2005; Müller, 2005), by considering the probability measures π_i as functions. This is a highly non-standard setup, however, since these functions (π_i) are defined on σ -algebras and are non-negative, σ -additive.

6. Conclusion

We have established a learning theory of distribution regression, where the inputs are probability measures on separable, topological domains endowed with reproducing kernels, and the outputs are elements of a separable Hilbert space. We studied a ridge regression scheme defined on embeddings of the input distributions to a reproducing kernel Hilbert space, which has a simple analytical solution, as well as theoretically sound, efficient methods for approximation (Zhang et al., 2015; Richtárik and Takáč, 2016; Alaoui and Mahoney, 2015; Yang et al., 2016; Rudi et al., 2015). We derived explicit bounds on the excess risk as a function of the number of samples and problem difficulty. We tackled both the well-

15. The motivation of this assumption comes from drug discovery: if a molecule has at least one well-binding configuration, then it is considered to bind well.

16. In terms of drug binding prediction, this means that a molecule binds to a target iff at least one of its configurations falls within a fixed, but unknown rectangle.

17. Often these “metrics” are only semi-metrics, as they do not satisfy the triangle inequality.

specified case (when the regression function belongs to the assumed RKHS modelling class), and the more general misspecified setup. As a special case of our results, we proved the consistency of regression for set kernels (Haussler, 1999; Gärtner et al., 2002), which was a 17-year-old open problem, and for a recent kernel family (Christmann and Steinwart, 2010), which we have expanded upon (Table 1). We proved an exact computational-statistical efficiency trade-off for the MERR estimator: in the well-specified setting, we showed how to choose the bag size in the two-stage sampled setup to match the one-stage sampled min-max optimal rate (Caponnetto and De Vito, 2007); and in the misspecified setting, our rates approximate closely an asymptotically optimal estimator imposing stricter eigenvalue decay conditions (Steinwart et al., 2009). Several exciting open questions remain, including whether improved/optimal rates can be derived in the misspecified case, whether we can obtain consistency guarantees for non-point estimates, and how to handle non-ridge extensions.

Finally, we note that although the primary focus of the current paper was theoretical, we have applied the MERR method (Szabó et al., 2015, Section A.2) to supervised entropy learning and aerosol prediction based on multispectral satellite images.¹⁸ In future work, we will address applications with vector-valued outputs.

7. Proofs

We provide proofs for our results detailed in Section 4: Section 7.1 (*resp.* Section 7.2) focuses on the well-specified case (*resp.* misspecified setting). The used lemmas are enlisted in Section 7.3.

7.1 Proofs of the Well-specified Case

We give proof details concerning the excess risk in the well-specified case (Theorem 2).

7.1.1 PROOF OF THE BOUND ON $\|g_{\mathbf{z}} - g_{\mathbf{z}}\|_{\mathcal{Y}}^2$

By (13), (14) we get $g_{\mathbf{z}} - g_{\mathbf{z}} = \frac{1}{l} \sum_{i=1}^l (K_{\mu_{x_i}} - K_{\mu_{x_i}}) y_i$; hence by applying the Hölder property of $K_{(\cdot)}$, the boundedness of y_i ($\|y_i\|_{\mathcal{Y}} \leq C$) and (24), we obtain

$$\begin{aligned} \|g_{\mathbf{z}} - g_{\mathbf{z}}\|_{\mathcal{Y}}^2 &\leq \frac{1}{l^2} \sum_{i=1}^l \|(K_{\mu_{x_i}} - K_{\mu_{x_i}}) y_i\|_{\mathcal{Y}}^2 \leq \frac{1}{l} \sum_{i=1}^l \|K_{\mu_{x_i}} - K_{\mu_{x_i}}\|_{L(\mathcal{Y}; \mathcal{Y})}^2 \|y_i\|_{\mathcal{Y}}^2 \\ &\leq \frac{L^2}{l} \sum_{i=1}^l \|y_i\|_{\mathcal{Y}}^2 \|\mu_{x_i} - \mu_{x_i}\|_H^2 \leq \frac{L^2 C^2}{l} \sum_{i=1}^l \left[\frac{(1 + \sqrt{\alpha}) \sqrt{2B_k}}{\sqrt{N}} \right]^{2h} = L^2 C^2 \frac{(1 + \sqrt{\alpha})^{2h} (2B_k)^h}{N^h} \end{aligned}$$

with probability at least $1 - le^{-\alpha}$, based on a union bound.

7.1.2 PROOF OF THE BOUND ON $\|T_{\mathbf{x}} - T_{\mathbf{x}}\|_{\mathcal{L}(\mathcal{Y})}^2$

Using the definition of $T_{\mathbf{x}}$ and $T_{\mathbf{x}}$, and exploiting (with $\|\cdot\|_{\mathcal{L}(\mathcal{Y})}$) that in a normed space¹⁹ $(N, \|\cdot\|)$, $f_i \in N$, $(i = 1, \dots, n)$

$$\left\| \sum_{i=1}^n f_i \right\|^2 \leq n \sum_{i=1}^n \|f_i\|^2, \quad (35)$$

we get

$$\|T_{\mathbf{x}} - T_{\mathbf{x}}\|_{\mathcal{L}(\mathcal{Y})}^2 \leq \frac{1}{l^2} \sum_{i=1}^l \|T_{\mu_{x_i}} - T_{\mu_{x_i}}\|_{\mathcal{L}(\mathcal{Y})}^2. \quad (36)$$

To upper bound $\|T_{\mu_{x_i}} - T_{\mu_{x_i}}\|_{\mathcal{L}(\mathcal{Y})}^2$, let us see how $T_{\mu_{x_i}} = K_{\mu_{x_i}} K_{\mu_{x_i}}^*$ acts. The existence of an $E \geq 0$ constant satisfying $\|(T_{\mu_{x_i}} - T_{\mu_{x_i}})(f)\|_{\mathcal{Y}} \leq E \|f\|_{\mathcal{Y}}$ implies $\|T_{\mu_{x_i}} - T_{\mu_{x_i}}\|_{\mathcal{L}(\mathcal{Y})} \leq E$. We continue with the l.h.s. of this equation using Eq. (35):

$$\begin{aligned} \|(T_{\mu_{x_i}} - T_{\mu_{x_i}})(f)\|_{\mathcal{Y}}^2 &= \|K_{\mu_{x_i}} K_{\mu_{x_i}}^*(f) - K_{\mu_{x_i}} K_{\mu_{x_i}}^*(f)\|_{\mathcal{Y}}^2 \\ &= \|K_{\mu_{x_i}} [K_{\mu_{x_i}}^*(f) - K_{\mu_{x_i}}^*(f)] + (K_{\mu_{x_i}} - K_{\mu_{x_i}}) K_{\mu_{x_i}}^*(f)\|_{\mathcal{Y}}^2 \\ &\leq 2 \left[\|K_{\mu_{x_i}} [K_{\mu_{x_i}}^*(f) - K_{\mu_{x_i}}^*(f)]\|_{\mathcal{Y}}^2 + \|(K_{\mu_{x_i}} - K_{\mu_{x_i}}) K_{\mu_{x_i}}^*(f)\|_{\mathcal{Y}}^2 \right]. \end{aligned}$$

By Eq. (45) and the Hölder continuity of $K_{(\cdot)}$, one arrives at

$$\begin{aligned} \|K_{\mu_{x_i}} [K_{\mu_{x_i}}^*(f) - K_{\mu_{x_i}}^*(f)]\|_{\mathcal{Y}}^2 &\leq \|K_{\mu_{x_i}}\|_{\mathcal{L}(\mathcal{Y}; \mathcal{Y})}^2 \|K_{\mu_{x_i}}^*(f) - K_{\mu_{x_i}}^*(f)\|_{\mathcal{Y}}^2 \\ &\leq \|K_{\mu_{x_i}}\|_{\mathcal{L}(\mathcal{Y}; \mathcal{Y})}^2 \|K_{\mu_{x_i}}^* - K_{\mu_{x_i}}^*\|_{\mathcal{L}(\mathcal{Y}; \mathcal{Y})}^2 \|f\|_{\mathcal{Y}}^2 = \|K_{\mu_{x_i}}\|_{\mathcal{L}(\mathcal{Y}; \mathcal{Y})}^2 \|(K_{\mu_{x_i}} - K_{\mu_{x_i}})^*\|_{\mathcal{L}(\mathcal{Y}; \mathcal{Y})}^2 \|f\|_{\mathcal{Y}}^2 \\ &= \|K_{\mu_{x_i}}\|_{\mathcal{L}(\mathcal{Y}; \mathcal{Y})}^2 \|K_{\mu_{x_i}} - K_{\mu_{x_i}}\|_{\mathcal{L}(\mathcal{Y}; \mathcal{Y})}^2 \|f\|_{\mathcal{Y}}^2 \leq B_K L^2 \|\mu_{x_i} - \mu_{x_i}\|_H^2 \|f\|_{\mathcal{Y}}^2, \\ \|(K_{\mu_{x_i}} - K_{\mu_{x_i}}) K_{\mu_{x_i}}^*(f)\|_{\mathcal{Y}}^2 &\leq \|K_{\mu_{x_i}} - K_{\mu_{x_i}}\|_{\mathcal{L}(\mathcal{Y}; \mathcal{Y})}^2 \|K_{\mu_{x_i}}^*(f)\|_{\mathcal{Y}}^2 \\ &\leq \|K_{\mu_{x_i}} - K_{\mu_{x_i}}\|_{\mathcal{L}(\mathcal{Y}; \mathcal{Y})}^2 \|K_{\mu_{x_i}}^*\|_{\mathcal{L}(\mathcal{Y}; \mathcal{Y})}^2 \|f\|_{\mathcal{Y}}^2 \leq B_K L^2 \|\mu_{x_i} - \mu_{x_i}\|_H^2 \|f\|_{\mathcal{Y}}^2. \end{aligned}$$

Hence $\|(T_{\mu_{x_i}} - T_{\mu_{x_i}})(f)\|_{\mathcal{Y}}^2 \leq 4B_K L^2 \|\mu_{x_i} - \mu_{x_i}\|_H^2 \|f\|_{\mathcal{Y}}^2 \Rightarrow E^2 = 4B_K L^2 \|\mu_{x_i} - \mu_{x_i}\|_H^2$. Exploiting this property in (36) with Eq. (24) we arrive to the bound

$$\begin{aligned} \|T_{\mathbf{x}} - T_{\mathbf{x}}\|_{\mathcal{L}(\mathcal{Y})}^2 &\leq \frac{4B_K L^2}{l} \sum_{i=1}^l \|\mu_{x_i} - \mu_{x_i}\|_H^2 \leq \frac{4B_K L^2}{l} \sum_{i=1}^l \frac{(1 + \sqrt{\alpha})^{2h} (2B_k)^h}{N^h} \\ &= \frac{(1 + \sqrt{\alpha})^{2h} 2^{h+2} (B_k)^h B_K L^2}{N^h}. \end{aligned} \quad (37)$$

7.1.3 PROOF: FINAL UNION BOUND IN THEOREM 2

Until now, we obtained that if (i) the sample number N satisfies Eq. (25), (ii) (24) holds (which has probability at least $1 - le^{-\alpha} = 1 - e^{-[\alpha - \log(l)]} = 1 - e^{-\alpha}$ applying a union bound

19. Eq. (35) holds since $\|\cdot\|^2$ is convex function, thus $\|\frac{1}{n} \sum_{i=1}^n f_i\|^2 \leq \frac{1}{n} \sum_{i=1}^n \|f_i\|^2$.

18. For code, see <https://bitbucket.org/szozoli/ite/>.

argument; $\alpha = \log(l) + \delta$, and (iii) $\Theta(\lambda, \mathbf{z}) \leq \frac{1}{2}$ is fulfilled [see Eq. (22)], then

$$\begin{aligned} S_{-1} + S_0 &\leq \frac{4}{\lambda} \left[L^2 C^2 (1 + \sqrt{\alpha})^{2h} (2B_K)^h + \frac{(1 + \sqrt{\alpha})^{2h} 2^{h+2} (B_K)^h B_K L^2}{N^h} \right. \\ &\quad \times \left. \left(\log^2 \left(\frac{6}{\eta} \right) \left\{ \frac{64}{\lambda} \left[\frac{M^2 B_K}{l^2 \lambda} + \frac{\Sigma^2 \mathcal{N}(\lambda)}{l} \right] + \frac{24}{\lambda^2} \left[\frac{4B_K^2 \mathcal{B}(\lambda)}{l^2} + \frac{B_K \mathcal{A}(\lambda)}{l} \right] \right\} + \mathcal{B}(\lambda) + \|f_\rho\|_{\mathcal{X}}^2 \right) \right] \\ &= \frac{4L^2 (1 + \sqrt{\alpha})^{2h} (2B_K)^h}{\lambda N^h} \left[C^2 + 4B_K \times \right. \\ &\quad \left. \times \left(\log^2 \left(\frac{6}{\eta} \right) \left\{ \frac{64}{\lambda} \left[\frac{M^2 B_K}{l^2 \lambda} + \frac{\Sigma^2 \mathcal{N}(\lambda)}{l} \right] + \frac{24}{\lambda^2} \left[\frac{4B_K^2 \mathcal{B}(\lambda)}{l^2} + \frac{B_K \mathcal{A}(\lambda)}{l} \right] \right\} + \mathcal{B}(\lambda) + \|f_\rho\|_{\mathcal{X}}^2 \right) \right]. \end{aligned}$$

By taking into account Caporinotto and De Vito (2007)'s bounds for S_1 and S_2 , $S_1 \leq 32 \log^2 \left(\frac{6}{\eta} \right) \left[\frac{B_K \lambda l^2}{l^2 \lambda} + \frac{\Sigma^2 \mathcal{N}(\lambda)}{l} \right]$, $S_2 \leq 8 \log^2 \left(\frac{6}{\eta} \right) \left[\frac{4B_K^2 \mathcal{B}(\lambda)}{l^2 \lambda} + \frac{B_K \mathcal{A}(\lambda)}{l \lambda} \right]$, plugging all the expressions to (21), we obtain Theorem 2 with a union bound.

7.2 Proofs of the Misspecified Case

We present the proof details concerning the excess risk in the misspecified case (Theorem 7).

7.2.1 PROOF OF THE BOUND ON $\sqrt{S_{-1}} + \sqrt{S_0}$ WITHOUT $\mathcal{P}(b, c)$

The upper bounds on S_{-1} and S_0 [which are defined in Eqs. (15), (16)] remain valid without modification provided that (i) $\Theta(\lambda, \mathbf{z}) = \|(T - T_{\mathbf{x}})(T + \lambda)^{-1}\|_{\mathcal{L}(\mathcal{G})} \leq \|(T - T_{\mathbf{x}})(T + \lambda)^{-1}\|_{\mathcal{L}_2(\mathcal{G})} \leq \frac{1}{2}$, where we used Eq. (1), (ii) Eq. (24) is satisfied (which has probability $1 - l e^{-\eta}$) and (iii) Eq. (25) holds. Our goal below is to guarantee the $\Theta(\lambda, \mathbf{z}) \leq \frac{1}{2}$ condition with high probability *without* assuming that the prior belongs to $\mathcal{P}(b, c)$.

Requirement $\Theta(\lambda, \mathbf{z}) \leq \frac{1}{2}$: Let us define $\xi_i = T_{\mu_{\mathbf{x}_i}}(T + \lambda)^{-1} \in \mathcal{L}_2(\mathcal{G})$, $(i = 1, \dots, l)$. With this choice we get $\mathbb{E}[\xi_i] = T(T + \lambda)^{-1}$, $(T - T_{\mathbf{x}})(T + \lambda)^{-1} = \mathbb{E}[\xi_i] - \frac{1}{l} \sum_{i=1}^l \xi_i$ and

$$\|\xi_i\|_{\mathcal{L}_2(\mathcal{G})} \leq \|T_{\mu_{\mathbf{x}_i}}\|_{\mathcal{L}_2(\mathcal{G})} \|(T + \lambda)^{-1}\|_{\mathcal{L}(\mathcal{G})} \leq B_K / \lambda \Rightarrow \mathbb{E}[\|\xi_i\|_{\mathcal{L}_2(\mathcal{G})}^2] \leq (B_K)^2 / \lambda^2,$$

where we made use of (2), the $\|T_{\mu_{\mathbf{x}_i}}\|_{\mathcal{L}_2(\mathcal{G})} \leq B_K$ identity following from the boundedness of K (Caporinotto and De Vito, 2007, page 341, Eq. (13)), and the spectral theorem. Consequently, by the Bernstein's inequality (Lemma 7.3.1 with $\mathcal{X} = \mathcal{L}_2(\mathcal{G})$, $B = 2B_K / \lambda$, $\sigma = B_K / \lambda$) we obtain that for $\forall \eta \in (0, 1)$

$$\mathbb{P} \left(\|(T - T_{\mathbf{x}})(T + \lambda)^{-1}\|_{\mathcal{L}_2(\mathcal{G})} \leq 2 \left(\frac{2B_K}{\lambda l} + \frac{B_K}{\sqrt{l}\lambda} \right) \log \left(\frac{2}{\eta} \right) \right) \geq 1 - \eta.$$

Thus, for $\Theta(\lambda, \mathbf{z}) \leq \frac{1}{2}$ with probability $1 - \eta$ it is sufficient to have

$$2 \left(\frac{2B_K}{\lambda l} + \frac{B_K}{\sqrt{l}\lambda} \right) \log \left(\frac{2}{\eta} \right) \leq \frac{6B_K}{\sqrt{l}\lambda} \log \left(\frac{2}{\eta} \right) \leq \frac{1}{2} \Leftrightarrow \left[\frac{12B_K}{\lambda} \log \left(\frac{2}{\eta} \right) \right]^2 \leq l. \quad (38)$$

Under these conditions, we arrived at the upper bound

$$\begin{aligned} \sqrt{S_{-1}} + \sqrt{S_0} &\leq \sqrt{\frac{4L^2 C^2 (1 + \sqrt{\alpha})^{2h} (2B_K)^h}{\lambda N^h}} \left[\sqrt{1} + \sqrt{\frac{4B_K}{\lambda}} \right] \\ &= \frac{2LC(1 + \sqrt{\alpha})^h (2B_K)^{\frac{h}{2}}}{\sqrt{\lambda N^{\frac{h}{2}}}} \left[1 + \frac{2\sqrt{B_K}}{\sqrt{\lambda}} \right], \end{aligned}$$

where as opposed to Section 7.1.3 and Eq. (23) we used a slightly cruder $\|f_{\mathbf{z}}^{\lambda}\|_{\mathcal{X}}^2 \leq \frac{C^2}{\lambda}$ bound; it holds without the $\mathcal{P}(b, c)$ assumption by the definition of $f_{\mathbf{z}}^{\lambda}$ and the boundedness of y since $\lambda \|f_{\mathbf{z}}^{\lambda}\|_{\mathcal{X}}^2 \leq \frac{1}{l} \sum_{i=1}^l \|y_i\|_{\mathcal{Y}}^2 \leq C^2$.

Remark: Notice that the price we pay for not assuming that the prior belongs to the $\mathcal{P}(b, c)$ class ($b > 1$) is a slightly tighter $\frac{1}{\lambda^2} \leq l$ constraint [Eq. (38)] instead of $\frac{1}{\lambda^{1+\frac{1}{2}}} \leq l$ in Eq. (19), and a somewhat looser $\|f_{\mathbf{z}}^{\lambda}\|_{\mathcal{X}}^2$ bound.

7.2.2 PROOF OF THE DECOMPOSITION OF $\|\sqrt{T}(f_{\mathbf{z}}^{\lambda} - f^{\lambda})\|_{\mathcal{X}}$

Using the analytical formula of $f_{\mathbf{z}}^{\lambda}$ [see Eq. (13)] and that of f^{λ} [see Eq. (17)]

$$f^{\lambda} = (S_K S_K^* + \lambda I)^{-1} S_K f_\rho = (T + \lambda I)^{-1} S_K f_\rho \quad (39)$$

one gets $(T + \lambda I) f^{\lambda} = S_K f_\rho \Rightarrow \lambda f^{\lambda} = S_K f_\rho - T f^{\lambda}$ and

$$\begin{aligned} f_{\mathbf{z}}^{\lambda} - f^{\lambda} &= (T_{\mathbf{x}} + \lambda I)^{-1} g_{\mathbf{z}} - f^{\lambda} = (T_{\mathbf{x}} + \lambda I)^{-1} g_{\mathbf{z}} - (T_{\mathbf{x}} + \lambda I)^{-1} (T_{\mathbf{x}} + \lambda I) f^{\lambda} \\ &= (T_{\mathbf{x}} + \lambda I)^{-1} [g_{\mathbf{z}} - (T_{\mathbf{x}} + \lambda I) f^{\lambda}] = (T_{\mathbf{x}} + \lambda I)^{-1} (g_{\mathbf{z}} - T_{\mathbf{x}} f^{\lambda} - \lambda f^{\lambda}) \\ &= (T_{\mathbf{x}} + \lambda I)^{-1} (g_{\mathbf{z}} - T_{\mathbf{x}} f^{\lambda} - S_K f_\rho + T f^{\lambda}) \\ &= (T_{\mathbf{x}} + \lambda I)^{-1} (g_{\mathbf{z}} - S_K f_\rho) + (T_{\mathbf{x}} + \lambda I)^{-1} (T - T_{\mathbf{x}}) f^{\lambda} \\ &= (T_{\mathbf{x}} + \lambda I)^{-1} (g_{\mathbf{z}} - S_K f_\rho) + (T_{\mathbf{x}} + \lambda I)^{-1} (T - T_{\mathbf{x}}) (T + \lambda I)^{-1} S_K f_\rho. \end{aligned} \quad (40)$$

Let us rewrite $(T + \lambda I)^{-1}$ by the $(A + UV)^{-1} = A^{-1} - A^{-1} U (I + V A^{-1} U)^{-1} V A^{-1}$ operator Woodbury formula (Ding and Zhou, 2008, Theorem 2.1, page 724)

$$\begin{aligned} (T + \lambda I)^{-1} &= (\lambda I + S_K S_K^*)^{-1} = (\lambda^{-1} I) - (\lambda^{-1} I) S_K [I + S_K^* (\lambda^{-1} I) S_K]^{-1} S_K (\lambda^{-1} I) \\ &= (\lambda^{-1} I) - \lambda^{-1} S_K (\lambda I + \tilde{T})^{-1} S_K^*. \end{aligned}$$

By the derived expression for $(T + \lambda I)^{-1}$, we get $(T + \lambda I)^{-1} S_K f_\rho = \lambda^{-1} S_K f_\rho - \lambda^{-1} S_K (\lambda I + \tilde{T})^{-1} S_K^* S_K f_\rho = \lambda^{-1} S_K [f_\rho - (T + \lambda I)^{-1} S_K^* S_K f_\rho]$. Plugging this result to Eq. (40), introducing the $g_\rho = S_K f_\rho$ notation, using the triangle inequality we get

$$\begin{aligned} \|\sqrt{T}(f_{\mathbf{z}}^{\lambda} - f^{\lambda})\|_{\mathcal{X}} &= \\ &= \|\sqrt{T}(T_{\mathbf{x}} + \lambda I)^{-1} \left\{ (g_{\mathbf{z}} - S_K f_\rho) + (T - T_{\mathbf{x}}) \lambda^{-1} S_K [f_\rho - (\tilde{T} + \lambda I)^{-1} S_K^* S_K f_\rho] \right\}\|_{\mathcal{X}} \\ &\leq \|\sqrt{T}(T_{\mathbf{x}} + \lambda I)^{-1}\|_{\mathcal{L}(\mathcal{G})} \left(\|g_{\mathbf{z}} - g_\rho\|_{\mathcal{X}} + \|T - T_{\mathbf{x}}\|_{\mathcal{L}(\mathcal{G})} \lambda^{-1} \|S_K [f_\rho - (\tilde{T} + \lambda I)^{-1} S_K^* S_K f_\rho]\|_{\mathcal{X}} \right). \end{aligned}$$

7.2.3 PROOF OF THE BOUND ON $\|g_{\mathbf{x}} - g_{\rho}\|_{\mathcal{H}}$

As is known $g_{\mathbf{x}} = \frac{1}{l} \sum_{i=1}^l K_{\mu_{x_i}} y_i$ [see Eq. (13)] and $g_{\rho} = \int_{\mathcal{X}} K_{\mu_{\rho}} f_{\rho}(\mu_{\rho}) d\rho_{\mathcal{X}}(\mu_{\rho})$ (Caponnetto and De Vito, 2007, Eq. (23), page 344). Let $\xi_i = K_{\mu_{x_i}} y_i \in \mathcal{H}$ ($i = 1, \dots, l$). In this case $\mathbb{E}[\xi_i] = g_{\rho}$, $g_{\rho} - g_{\mathbf{x}} = \mathbb{E}[\xi_i] - \frac{1}{l} \sum_{i=1}^l \xi_i$, and $\|\xi_i\|_{\mathcal{H}}^2 = \|K_{\mu_{x_i}} y_i\|_{\mathcal{H}}^2 \leq \|K_{\mu_{x_i}}\|_{\mathcal{L}(\mathcal{Y}, \mathcal{H})} \|y_i\|_{\mathcal{Y}}^2 \leq B_K C^2 \Rightarrow \|\xi_i\|_{\mathcal{H}} \leq C\sqrt{B_K} \Rightarrow \mathbb{E}[\|\xi_i\|_{\mathcal{H}}^2] \leq C^2 B_K$ using the boundedness of kernel K ($\|K_{\mu_{x_i}}\|_{\mathcal{L}(\mathcal{Y}, \mathcal{H})} \leq B_K$) and the boundedness of output y ($\|y\|_{\mathcal{Y}} \leq C$). Applying the Bernstein inequality (see Lemma 7.3.1 with $\mathcal{X} = \mathcal{H}$, $B = 2C\sqrt{B_K}$, $\sigma = C\sqrt{B_K}$) one gets that for any $\eta \in (0, 1)$

$$\mathbb{P}\left(\|g_{\mathbf{x}} - g_{\rho}\|_{\mathcal{H}} \leq 2\left(\frac{2C\sqrt{B_K}}{l} + \frac{C\sqrt{B_K}}{\sqrt{l}}\right) \log\left(\frac{2}{\eta}\right)\right) \geq 1 - \eta.$$

7.2.4 PROOF OF THE BOUND ON $\|T - T_{\mathbf{x}}\|_{\mathcal{L}(\mathcal{H})}$

Let $\xi_i = T_{\mu_{x_i}} \in \mathcal{L}_2(\mathcal{H})$ ($i = 1, \dots, l$), then $\mathbb{E}[\xi_i] = T$, $T - T_{\mathbf{x}} = T - \frac{1}{l} \sum_{i=1}^l T_{\mu_{x_i}}$, $\|\xi_i\|_{\mathcal{L}_2(\mathcal{H})} = \|T_{\mu_{x_i}}\|_{\mathcal{L}_2(\mathcal{H})} \leq B_K$, $\mathbb{E}[\|\xi_i\|_{\mathcal{L}_2(\mathcal{H})}^2] \leq B_K^2$. Applying the $\|T - T_{\mathbf{x}}\|_{\mathcal{L}(\mathcal{H})} \leq \|T - T_{\mathbf{x}}\|_{\mathcal{L}_2(\mathcal{H})}$ relation [see Eq. (1)] and the Bernstein inequality (see Lemma 7.3.1 with $\mathcal{X} = \mathcal{L}_2(\mathcal{H})$, $B = 2B_K$, $\sigma = B_K$), we obtain that for any $\eta \in (0, 1)$

$$\mathbb{P}\left(\|T - T_{\mathbf{x}}\|_{\mathcal{L}(\mathcal{H})} \leq 2\left(\frac{2B_K}{l} + \frac{\sigma}{\sqrt{l}}\right) \log\left(\frac{2}{\eta}\right)\right) \geq 1 - \eta.$$

7.2.5 PROOF OF THE DECOMPOSITION OF $\|S_K(f_{\rho} - (\tilde{T} + \lambda I)^{-1} S_K^* S_K f_{\rho})\|_{\mathcal{H}}$

Since $\|S_K a\|_{\mathcal{H}}^2 = \langle S_K a, S_K a \rangle_{\mathcal{H}} = \langle S_K^* S_K a, a \rangle_{\rho} = \langle \tilde{T} a, a \rangle_{\rho}$ ($\forall a \in L_{\rho, \mathcal{X}}^2$) by the definition of the adjoint operator and $\tilde{T} = S_K^* S_K$ [see Eq. (6)], we can rewrite the target term as

$$\begin{aligned} & \|S_K [f_{\rho} - (\tilde{T} + \lambda I)^{-1} S_K^* S_K f_{\rho}]\|_{\mathcal{H}}^2 = \\ &= \langle \tilde{T} [f_{\rho} - (\tilde{T} + \lambda I)^{-1} S_K^* S_K f_{\rho}], f_{\rho} - (\tilde{T} + \lambda I)^{-1} S_K^* S_K f_{\rho} \rangle_{\rho} \\ &\leq \|\tilde{T} [f_{\rho} - (\tilde{T} + \lambda I)^{-1} S_K^* S_K f_{\rho}]\|_{\rho} \|f_{\rho} - (\tilde{T} + \lambda I)^{-1} S_K^* S_K f_{\rho}\|_{\rho}, \end{aligned}$$

where the CBS (Cauchy-Bunyakovsky-Schwarz) inequality was applied. Since

$$\begin{aligned} (S_K^* S_K + \lambda I) S_K^* &= S_K^* (S_K S_K^* + \lambda I) & S_K^* (S_K S_K^* + \lambda I)^{-1} &= (S_K^* S_K + \lambda I)^{-1} S_K^* \\ S_K^* (S_K S_K^* + \lambda I)^{-1} S_K &= (S_K^* S_K + \lambda I)^{-1} S_K^* S_K & S_K^* (S_K S_K^* + \lambda I)^{-1} S_K &= S_K^* f^{\lambda} \\ S_K^* (T + \lambda I)^{-1} S_K &= (\tilde{T} + \lambda I)^{-1} \tilde{T} \end{aligned} \quad (41) \quad (42)$$

using Eq. (41) and the analytical expression for f^{λ} [see Eq. (39)] we have

$$\begin{aligned} (\tilde{T} + \lambda I)^{-1} \tilde{T} f_{\rho} &= (\tilde{T} + \lambda I)^{-1} S_K^* S_K f_{\rho} = (S_K^* S_K + \lambda I)^{-1} S_K^* S_K f_{\rho} \\ &= S_K^* (S_K S_K^* + \lambda I)^{-1} S_K f_{\rho} = S_K^* f^{\lambda} \end{aligned} \quad (43)$$

and $\|S_K [f_{\rho} - (\tilde{T} + \lambda I)^{-1} S_K^* S_K f_{\rho}]\|_{\mathcal{H}}^2 \leq \|\tilde{T} [f_{\rho} - (\tilde{T} + \lambda I)^{-1} S_K^* S_K f_{\rho}]\|_{\rho} \|S_K^* f^{\lambda} - f_{\rho}\|_{\rho}$.

7.2.6 PROOF OF THE BOUND ON $\|S_K^* f^{\lambda} - f_{\rho}\|_{\rho}$

Let us apply (i) the $Af - f = Af - f - q' + q' = (A - I)(f - q') + Aq' - q'$ relation with $A = (\tilde{T} + \lambda I)^{-1} \tilde{T}$, $f = f_{\rho}$ and $q' = S_K^* q$, where $q \in \mathcal{H}$ is an arbitrary element from \mathcal{H} , (ii) Eq. (43) and (iii) the triangle inequality to arrive at

$$\begin{aligned} \|S_K^* f^{\lambda} - f_{\rho}\|_{\rho} &= \|(\tilde{T} + \lambda I)^{-1} \tilde{T} f_{\rho} - f_{\rho}\|_{\rho} \\ &= \|[(\tilde{T} + \lambda I)^{-1} \tilde{T} - I](f_{\rho} - S_K^* q) + (\tilde{T} + \lambda I)^{-1} \tilde{T} S_K^* q - S_K^* q\|_{\rho} \\ &\leq \|[(\tilde{T} + \lambda I)^{-1} \tilde{T} - I](f_{\rho} - S_K^* q)\|_{\rho} + \|(\tilde{T} + \lambda I)^{-1} \tilde{T} S_K^* q - S_K^* q\|_{\rho}. \end{aligned}$$

Below we give upper bounds on these two terms.

First, notice that $\mu_x \in X \mapsto \|K(\mu_x, \mu_x)\|_{\mathcal{L}(\mathcal{Y})} \leq B_K$. This boundedness with the strong continuity of $K(\cdot)$ imply (Carmeli et al., 2006, Proposition 12) that $\mathcal{H} \subseteq C(X, \mathcal{Y})$, i.e., K is a Mercer kernel. Since K_{μ_x} is a Hilbert-Schmidt operator for all $\mu_x \in X$ [see Eq. (11)], it is also a compact operator ($\forall \mu_x \in X$). The compactness of K_{μ_x} -s with the bounded and Mercer property of K guarantees the boundedness of S_K^* and that \tilde{T} is a compact, positive, self-adjoint operator (Carmeli et al., 2010, Proposition 3).

Bound on $\|[(\tilde{T} + \lambda I)^{-1} \tilde{T} - I](f_{\rho} - S_K^* q)\|_{\rho}$: Since \tilde{T} is a compact positive self-adjoint operator, by the spectral theorem (Steinwart and Christmann, 2008, Theorem 4.27, page 127) there exist an $(u_i)_{i \in I}$ countable ONB in $cl[Im(\tilde{T})]$, and $a_1 \geq a_2 \geq \dots > 0$ such that $\tilde{T} f = \sum_{i \in I} a_i \langle f, u_i \rangle_{\rho} u_i$ ($\forall f \in L_{\rho, \mathcal{X}}^2$) and let $(v_j)_{j \in J}$ (J is also countable by the separability²⁰ of $L_{\rho, \mathcal{X}}^2$) an ONB in $Ker(\tilde{T}^*) = Ker(\tilde{T})$; $L_{\rho, \mathcal{X}}^2 = cl[Im(\tilde{T})] \oplus Ker(\tilde{T})$. Thus,

$$\begin{aligned} \|(\tilde{T} + \lambda I)^{-1} \tilde{T} - I\|_{\rho} \|f_{\rho} - S_K^* q\|_{\rho} &= \sum_{i \in I} \left(\frac{a_i}{a_i + \lambda} - 1 \right)^2 \langle f_{\rho} - S_K^* q, u_i \rangle_{\rho}^2 + \sum_{j \in J} \langle f_{\rho} - S_K^* q, v_j \rangle_{\rho}^2 \\ &\leq \sum_{i \in I} \langle f_{\rho} - S_K^* q, u_i \rangle_{\rho}^2 + \sum_{j \in J} \langle f_{\rho} - S_K^* q, v_j \rangle_{\rho}^2 = \|f_{\rho} - S_K^* q\|_{\rho}^2 \end{aligned}$$

exploiting the Parseval's identity and that $\left(\frac{\lambda}{\lambda + \lambda} - 1\right)^2 \leq 1$.

Bound on $\|(\tilde{T} + \lambda I)^{-1} \tilde{T} S_K^* q - S_K^* q\|_{\rho}$: By using Eq. (42), Eq. (32), and Lemma 7.3.2 ($M = T = S_K^* S_K$, $\mathcal{X} = \mathcal{H}$, $f = q$, $a = \frac{1}{2}$), the target quantity can be bounded as

$$\begin{aligned} \|(\tilde{T} + \lambda I)^{-1} \tilde{T} S_K^* q - S_K^* q\|_{\rho} &= \|S_K^* (T + \lambda I)^{-1} S_K S_K^* q - S_K^* q\|_{\rho} \\ &= \|\sqrt{\tilde{T}} [(\tilde{T} + \lambda I)^{-1} S_K S_K^* q - q]\|_{\mathcal{H}} \\ &= \|\sqrt{\tilde{T}} [(T + \lambda I)^{-1} T q - q]\|_{\mathcal{H}} \leq \max(1, \|\tilde{T}\|_{\mathcal{L}(\mathcal{H})}) \lambda^{\frac{1}{2}} \|q\|_{\mathcal{H}}. \end{aligned}$$

Making use of the two derived bounds, we get $\|S_K^* f^{\lambda} - f_{\rho}\|_{\rho} \leq \|f_{\rho} - S_K^* q\|_{\rho} + \max(1, \|\tilde{T}\|_{\mathcal{L}(\mathcal{H})}) \lambda^{\frac{1}{2}} \|q\|_{\mathcal{H}}$.

²⁰ $L_{\rho, \mathcal{X}}^2 = L^2(X, \mathcal{B}(H))_{\mathcal{X}, \rho, \mathcal{X}; \mathbb{R}}$ is isomorphic to $L^2(X, \mathcal{B}(H))_{\mathcal{X}, \rho, \mathcal{X}; \mathbb{R}} \otimes Y$, where \otimes is the tensor product of Hilbert spaces. The separability follows from that of Y and $L^2(X, \mathcal{B}(H))_{\mathcal{X}, \rho, \mathcal{X}; \mathbb{R}}$; the latter holds (Cohn, 2013, Proposition 3.4.5) since $\mathcal{B}(H)_{\mathcal{X}}$ is countably generated since $X \subseteq H$ is separable.

7.3 Supplementary Lemmas

In this section, we list two lemmas used in the proofs.

7.3.1 BERNSTEIN'S INEQUALITY (CAPONNETTO AND DE VITO, 2007, Prop. 2, p. 345)

Let ξ_i ($i = 1, \dots, l$) be i.i.d. realizations of a random variable on a (Ω, \mathcal{A}, P) probability space with values in a separable Hilbert space \mathcal{X} . If there exist $B > 0$, $\sigma > 0$ constants such that $\|\xi_i(\omega)\|_{\mathcal{X}} \leq \frac{B}{\sigma}$ a.s., $\mathbb{E}[\|\xi_i\|_{\mathcal{X}}^2] \leq \sigma^2$, then for all $l \geq 1$ and $\eta \in (0, 1)$ we have

$$\mathbb{P}\left(\left\|\frac{1}{l}\sum_{i=1}^l \xi_i - \mathbb{E}[\xi_1]\right\|_{\mathcal{X}} \leq 2\left(\frac{B}{l} + \frac{\sigma}{\sqrt{l}}\right) \log\left(\frac{2}{\eta}\right)\right) \geq 1 - \eta.$$

7.3.2 LEMMA ON BOUNDED, SELF-ADJOINT COMPACT OPERATORS; SHPERUMBUDUR ET AL. (2014, PROPOSITION A.2, PAGE 39)

Let M be a bounded, self-adjoint compact operator on a separable Hilbert space \mathcal{X} . Let $a \geq 0$, $\lambda > 0$, and $s \geq 0$. Let $f \in \mathcal{X}$ such that $f \in \text{Im}(M^s)$. If $s + a > 0$, then

$$\|M^a [(M + \lambda I)^{-1} M f - f]\|_{\mathcal{X}} \leq \max\left(1, \|M\|_{\mathcal{L}(\mathcal{X})}^{\min(1, s+a)}\right) \|M^{-s} f\|_{\mathcal{X}}.$$

Note: specifically for $s = 0$ we have $\text{Im}(M^s) = \text{Im}(I) = \mathcal{X}$, in other words, there is no additional range space constraint.

8. Discussion of Our Assumptions

We give a short insight into the consequences of our assumptions (detailed in Section 3) and present some concrete examples.

- **Well-definedness of ρ :** The boundedness and continuity of k imply the measurability of $\mu : (\mathcal{M}_1^+(X), \mathcal{B}(\tau_w)) \rightarrow (H, \mathcal{B}(H))$. Let τ denote the open sets on $H = H(k)$, $\tau|_X = \{A \cap X : A \in \tau\}$ the subspace topology on X , and $\mathcal{B}(H)|_X = \{A \cap X : A \in \mathcal{B}(H)\}$ the subspace σ -algebra on X . By noting (Schwartz, 1998, Corollary 5.2.13) that $\mathcal{B}(\tau|_X) = \mathcal{B}(H)|_X = \{A \in \mathcal{B}(H) : A \subseteq X\} \subseteq \mathcal{B}(H)$, the H -measurability of μ guarantees the measurability of $\mu : (\mathcal{M}_1^+(X), \mathcal{B}(\tau_w)) \rightarrow (X, \mathcal{B}(H)|_X)$, and hence the well-definedness of ρ , the measure induced by \mathcal{M} on $X \times Y$; for further details see (Szabó et al., 2015, Section A.1.1).²¹
- **Separability of X :** separability of \mathcal{X} and the continuity of k implies the separability of $H = H(k)$ (Steinwart and Christmann, 2008, Lemma 4.33, page 130). Also, since $X \subseteq H$, X is separable.
- **Finiteness of B_k :** If X is compact, then the continuity of k implies $B_k < \infty$.
- **Finiteness of B_k , compact metrichness of X :** Let \mathcal{X} be a compact metric space. In this case $\mathcal{M}_1^+(X)$ is also compact metric (Parthasarathy, 1967, Theorem 6.4, page 55).

21. Note that the referred proof also holds for separable Hilbert Y , and by the simplified reasoning above the original $X \in \mathcal{B}(H)$ condition could be avoided.

Hence if $\mu : (\mathcal{M}_1^+(X), \tau_w) \rightarrow H(k)$ is continuous²² (not just measurable), then X is compact metric and thus by the Hölder property of $K(\cdot)$, it is continuous implying that $B_K < \infty$.

- **K properties:** It is known (Caponnetto and De Vito, 2007, page 339-340) that

$$K(\mu_a, \mu_b) = K_{\mu_a}^* K_{\mu_b}, \quad (\forall \mu_a, \mu_b \in X) \quad (44)$$

$$\|K_{\mu_a}\|_{\mathcal{L}(Y; Y)} = \|K_{\mu_a}^*\|_{\mathcal{L}(X; X)} \leq \sqrt{B_K}, \quad (\forall \mu_a \in X). \quad (45)$$

Remark: In terms of Eq. (44), the Eq. (11) assumption means that the $\{K(\mu_a, \mu_a)\}_{\mu_a \in X}$ operators are trace class, specifically they are compact operators.

- **Separability of \mathcal{Y} :** The separability of X and the continuity of K imply the separability of \mathcal{Y} . Indeed, since $\mu_a \mapsto K_{\mu_a}$ is Hölder continuous w.r.t. the Hilbert-Schmidt norm it is also continuous. As a result it is continuous w.r.t. the operator norm, and thus also w.r.t. the strong topology. Using this property with the finiteness of B_K the separability of \mathcal{Y} follows (Carmali et al., 2006, Proposition 5.1, Corollary 5.2).

- Our assumptions imply Caponnetto and De Vito (2007)'s conditions (not considering the $\mathcal{P}(b, c)$ prior requirement). Indeed
 1. Y is a separable Hilbert space by assumption; the same property also holds for \mathcal{Y} as we have seen.
 2. The measurability of $(\mu_x, \mu_t) \mapsto \langle K_{\mu_x} w, K_{\mu_t} v \rangle_{\mathcal{Y}}$ for $\forall w, v \in Y$ is guaranteed by the continuity of $K(\cdot)$ w.r.t. the strong topology.
 3. We have $\int_{X \times Y} \|y\|_{\mathcal{Y}}^2 d\rho_X(\mu_x; y) \leq \int_{X \times Y} C^2 d\rho_X(\mu_x; y) = C^2 < \infty$ due to the boundedness of y , and hence $\exists \Sigma > 0, \exists M > 0$ such that for ρ_X -almost $\mu_x \in X$

$$\int_Y \|y - f_{\rho}(\mu_x)\|_{\mathcal{Y}}^m d\rho(y|\mu_x) \leq \frac{m! \Sigma^2 M^{m-2}}{2} \quad (\forall m \geq 2). \quad (46)$$

Indeed, by (Caponnetto and De Vito, 2007, Eq. (33)) the Bernstein condition (46) holds if $\|y - f_{\rho}(\mu_x)\|_{\mathcal{Y}} \leq \frac{M}{\Sigma}$, $\int_Y \|y - f_{\rho}(\mu_x)\|_{\mathcal{Y}}^2 d\rho(y|\mu_x) \leq \Sigma^2$. In our case using the boundedness of y , the regression function is also bounded and the same holds for $\|y - f_{\rho}(\mu_x)\|_{\mathcal{Y}}$ by the triangle inequality: $\|y - f_{\rho}(\mu_x)\|_{\mathcal{Y}} \leq C + \|f_{\rho}\|_{\mathcal{Y}} \sqrt{B_K}$; thus, $M = 2(C + \|f_{\rho}\|_{\mathcal{Y}} \sqrt{B_K})$, $\Sigma = \frac{M}{\Sigma}$ is a suitable choice.

4. The Polishness of $X \times Y$ was used by Caponnetto and De Vito (2007) to assure the existence of $\rho(y|\mu_a)$; we guaranteed this existence under somewhat milder conditions (see footnote 7).

Real-valued outputs: We now consider the specific case of $Y = \mathbb{R}$, when the following simplifications and results hold. By noting that in this case $\text{Tr}(K_{\mu_a}^* K_{\mu_a}) = K(\mu_a, \mu_a)$, Eq. (11) simplifies to the boundedness of kernel K in the traditional sense

$$K(\mu_a, \mu_a) \leq B_K \quad (\forall \mu_a \in X). \quad (47)$$

22. For example, if k is universal, then μ metrizes the weak topology τ_w (Shperumbudur et al., 2010, Theorem 23, page 1552), hence μ is continuous.

K_G	K_e	K_C	K_t	K_i
$e^{-\frac{\ \mu_a - \mu_b\ _H^2}{2\theta^2}}$	$e^{-\frac{\ \mu_a - \mu_b\ _H}{2\theta^2}}$	$\left(1 + \frac{\ \mu_a - \mu_b\ _H^2}{\theta^2}\right)^{-1}$	$\left(1 + \frac{\ \mu_a - \mu_b\ _H}{\theta}\right)^{-1}$	$\left(\frac{\ \mu_a - \mu_b\ _H^2}{\theta^2} + \theta^2\right)^{-\frac{1}{2}}$
$h = 1$	$h = \frac{1}{2}$	$h = 1$	$h = \frac{\theta}{2}$ ($\theta \leq 2$)	$h = 1$

Table 1: Nonlinear kernels on mean embedded distributions: $K = K(\mu_a, \mu_b)$; $\theta > 0$. For the Hölder continuity of Ψ_K , we assume that \mathcal{X} is a compact metric space and μ is continuous.

Eq. (12) reduces to the Hölder continuity of the canonical feature map $\Psi_K(\mu_c) := K(\cdot, \mu_c) : X \rightarrow \mathcal{H}$, in other words $\exists L > 0$, $h \in (0, 1]$ such that $\|\Psi_K(\mu_a) - \Psi_K(\mu_b)\|_{\mathcal{H}} \leq L \|\mu_a - \mu_b\|_H^h$, $\forall (\mu_a, \mu_b) \in X \times X$. In case of a linear kernel, $K(\mu_a, \mu_b) = \langle \mu_a, \mu_b \rangle_H$, ($\mu_a, \mu_b \in X$), the Hölder continuity of Ψ_K holds with $L = 1$, $h = 1$, and $B_K = B_k$ is a suitable choice. Evaluating the kernel K at the empirical embeddings $\mu_{\hat{x}_i} = \int_X k(\cdot, u) d\hat{x}_i(u) = \frac{1}{N} \sum_{n=1}^N k(\cdot, x_{i,n}) \in H$ yields the standard set kernel

$$K(\mu_{\hat{x}_i}, \mu_{\hat{x}_j}) = \langle \mu_{\hat{x}_i}, \mu_{\hat{x}_j} \rangle_H = \left\langle \frac{1}{N} \sum_{n=1}^N k(\cdot, x_{i,n}), \frac{1}{N} \sum_{m=1}^N k(\cdot, x_{j,m}) \right\rangle_H = \frac{1}{N^2} \sum_{n,m=1}^N k(x_{i,n}, x_{j,m})$$

by the bilinearity of $\langle \cdot, \cdot \rangle_H$ and the reproducing property of k .

Remark: One can define many nonlinear kernels (see Table 1) on mean embedded distributions. These kernels are the natural extensions to distributions of the Gaussian (Christmann and Steinwart, 2010), exponential, Cauchy, generalized t-student and inverse multiquadratic kernels. If \mathcal{X} is a compact metric space and μ is continuous, then the Ψ_K canonical feature maps, associated to K -s in Table 1, can be shown to satisfy our Hölder continuity requirement [Eq. (12)]; for details, see (Szabó et al., 2015, Section A.1.5-A.1.6).

Acknowledgments

We would like to thank the anonymous reviewers for their highly valuable, constructive suggestions to improve the manuscript. This work was supported by the Gatsby Charitable Foundation, NSF grant 1247658, and DOE grant DE-SC001114. A part of the work was carried out while Bharath K. Sripunbudur was a research fellow in the Statistical Laboratory, Department of Pure Mathematics and Mathematical Statistics at the University of Cambridge, UK.

References

- Ahmed Alaoui and Michael W. Mahoney. Fast randomized kernel ridge regression with statistical guarantees. In *Advances in Neural Information Processing Systems (NIPS)*, pages 775–783, 2015.
- Charalambos D. Aliprantis and Kim C. Border. *Infinite Dimensional Analysis: A Hitchhiker’s Guide*. Springer, 2006.

Yasemin Altun and Alexander Smola. Unifying divergence minimization and statistical inference via convex duality. In *Conference on Learning Theory (COLT)*, pages 139–153, 2006.

Mauricio A. Álvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4:195–266, 2011.

Jaume Amores. Multiple instance classification: Review, taxonomy and comparative study. *Artificial Intelligence*, 201:81–105, 2013.

Peter Auer. Approximating hyper-rectangles: Learning and pseudorandom sets. *Journal of Computer and System Sciences*, 57:376–388, 1998.

Boris Babenko. Multiple instance learning: Algorithms and applications. Technical report, Department of Computer Science and Engineering, University of California, San Diego, 2004. (http://cms.brookes.ac.uk/research/visiogrroup/talks/rg_dec_11_09/bbenko_re.pdf).

Boris Babenko, Nakul Verma, Piotr Dollár, and Serge Belongie. Multiple instance learning with manifold bags. In *International Conference on Machine Learning (ICML)*, pages 81–88, 2011.

Charles Bergeron, Jed Zaretzki, Curt Breneman, and Kristin P. Bennett. Multiple instance ranking. In *International Conference on Machine Learning (ICML)*, pages 48–55, 2008.

Charles Bergeron, Gregory Moore, Jed Zaretzki, Curt M. Breneman, and Kristin P. Bennett. Fast bundle algorithm for multiple-instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:1068–1079, 2012.

Alain Berlinet and Christine Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer, 2004.

Avrim Blum and Adam Kalai. A note on learning from multiple-instance examples. *Machine Learning*, 30:23–29, 1998.

Hanan Borchani, Gherardo Varando, Concha Bielza, and Pedro Larranaga. A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5:216–233, 2015.

Céline Brouard, Florence d’Alché Buc, and Marie Szafranski. Semi-supervised penalized output kernel regression for link prediction. In *International Conference on Machine Learning (ICML)*, pages 593–600, 2011.

Andrea Caponnetto. Optimal rates for regularization operators in learning theory. Technical report, Massachusetts Institute of Technology, 2006. (http://www6.cityu.edu.hk/ma/doc/peop1e/caponnetto/regop_TR%28TR11%29.pdf).

Andrea Caponnetto and Ernesto De Vito. Optimal rates for regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7:331–368, 2007.

- Claudio Carmeli, Ernesto De Vito, and Alessandro Toigo. Vector valued reproducing kernel Hilbert spaces of integrable functions and Mercer theorem. *Analysis and Applications*, 4: 377–408, 2006.
- Claudio Carmeli, Ernesto De Vito, Alessandro Toigo, and Veronica Umanitá. Vector valued reproducing kernel Hilbert spaces and universality. *Analysis and Applications*, 8:19–61, 2010.
- Kevin M. Carter, Raviv Raich, William G. Finn, and Alfred O. Hero. Information-geometric dimensionality reduction. *IEEE Signal Processing Magazine*, 28:89–99, 2011.
- Jing Chai, Hongtao Chen, Lixia Huang, and Fanhua Shang. Maximum margin multiple-instance feature weighting. *Pattern Recognition*, 47:2091–2103, 2014a.
- Jing Chai, Xinghao Ding, Hongtao Chen, and Tingyu Li. Multiple-instance discriminant analysis. *Pattern Recognition*, 47:2517–2531, 2014b.
- Ying Chen and Ou Wu. Contextual Hausdorff dissimilarity for multi-instance clustering. In *International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pages 870–873, 2012.
- Andreas Christmann and Ingo Steinwart. Universal kernels on non-standard input spaces. In *Advances in Neural Information Processing Systems (NIPS)*, pages 406–414, 2010.
- Donald L. Cohn. *Measure Theory: Second Edition*. Birkhäuser Basel, 2013.
- Marco Cuturi, Kenji Fukumizu, and Jean-Philippe Vert. Semigroup kernels on measures. *Journal of Machine Learning Research*, 6:11691198, 2005.
- Ernesto de Vito, Lorenzo Rosasco, and Andrea Caporinnetto. Discretization error analysis for Tikhonov regularization. *Analysis and Applications*, 4:81–99, 2006.
- Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89:31–71, 1997.
- Jin Ding and Aihui Zhou. A spectrum theorem for perturbed bounded linear operators. *Applied Mathematics and Computation*, 201:723–728, 2008.
- Daniel R. Doody, Qi Zhang, Sally A. Goldman, and Robert A. Amar. Multiple-instance learning of real-valued data. *Journal of Machine Learning Research*, 3:651–678, 2002.
- Gerald Edgar. *Measure, Topology and Fractal Geometry*. Springer-Verlag, 1995.
- Thomas Eifer and Heikki Mannila. Distance measures for point sets and their computation. *Acta Informatica*, 34:109–133, 1997.
- Theodoros Evgeniou, Charles A. Micchelli, and Massimiliano Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.
- Seetha Flaxman, Yu-Xiang Wang, and Alex Smola. Who supported Obama in 2012? ecological inference through distribution regression. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 289–298, 2015.
- James Foulds and Eibe Frank. A review of multi-instance learning assumptions. *The Knowledge Engineering Review*, 25:1–25, 2010.
- Kenji Fukumizu, Francis Bach, and Michael Jordan. Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *Journal of Machine Learning Research*, 5:73–99, 2004.
- Thomas Gärtner, Peter A. Flach, Adam Kowalczyk, and Alexander Smola. Multi-instance kernels. In *International Conference on Machine Learning (ICML)*, pages 179–186, 2002.
- Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.
- László Györfi, Michael Kohler, Adam Krzyżak, and Harro Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer, New-york, 2002.
- David Haussler. Convolution kernels on discrete structures. Technical report, Department of Computer Science, University of California at Santa Cruz, 1999. (<http://cbse.soe.ucsc.edu/sites/default/files/convolutions.pdf>).
- Matthias Hein and Olivier Bousquet. Hilbertian metrics and positive definite kernels on probability measures. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 136–143, 2005.
- Yang Hu, Mingjing Li, and Nenghai Yu. Multiple-instance ranking: Learning to rank images for image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.
- Tony Jebara, Risi Kondor, and Andrew Howard. Probability product kernels. *Journal of Machine Learning Research*, 5:819–844, 2004.
- Hachem Kadri, Emmanuel Dufos, Philippe Preux, Stéphane Cannu, and Manuel Davy. Non-linear functional regression: a functional RKHS approach. *International Conference on Artificial Intelligence and Statistics (AISTATS, JMLR W&CP)*, 9:374–380, 2010.
- Hachem Kadri, Alain Rakotomamonjy, Francis Bach, and Philippe Preux. Multiple operator-valued kernel learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2429–2437, 2012.
- Hachem Kadri, Mohammed Ghannouchi, and Philippe Preux. A generalized kernel approach to structured output learning. *International Conference on Machine Learning (ICML, JMLR W&CP)*, 28:471–479, 2013.
- Hachem Kadri, Emmanuel Dufos, Philippe Preux, Stéphane Cannu, Alain Rakotomamonjy, and Julien Audiffren. Operator-valued kernels for learning from functional response data. *Journal of Machine Learning Research*, 17:1–54, 2016.
- Risi Kondor and Tony Jebara. A kernel between sets of vectors. In *International Conference on Machine Learning (ICML)*, pages 361–368, 2003.

- Samory Kpotufe. k-NN regression adapts to local intrinsic dimension. Technical report, Max Planck Institute for Intelligent Systems, 2011. (<http://arxiv.org/abs/1110.4300>).
- James T. Kwok and Pak-Ming Cheung. Marginalized multi-instance kernels. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 901–906, 2007.
- Philip M. Long and Lei Tan. PAC learning of axis-aligned rectangles with respect to product distributions from multiple-instance examples. *Machine Learning*, 30:7–21, 1998.
- David Lopez-Paz, Krikamol Muandet, Bernhard Schölkopf, and Iliya Tolstikhin. Towards a learning theory of cause-effect inference. *International Conference on Machine Learning (ICML; JMLR W&CP)*, 37:1452–1461, 2015.
- André F. T. Martins, Noah A. Smith, Eric P. Xing, Pedro M. Q. Aguiar, and Mário A. T. Figueiredo. Nonextensive information theoretical kernels on measures. *Journal of Machine Learning Research*, 10:935–975, 2009.
- Shahar Mendelson and Joseph Neeman. Regularization in kernel learning. *The Annals of Statistics*, 38:526–565, 2010.
- Charles A. Micchelli and Massimiliano Pontil. On learning vector-valued functions. *Neural Computation*, 17:177–204, 2005.
- Krikamol Muandet, Keiji Fukumizu, Francesco Dinuzzo, and Bernhard Schölkopf. Learning from distributions via support measure machines. In *Advances in Neural Information Processing Systems (NIPS)*, pages 10–18, 2012.
- Hans-Georg Müller. Functional modelling and classification of longitudinal data. *Scandinavian Journal of Statistics*, 32:223–240, 2005.
- Frank Nielsen and Richard Nock. A closed-form expression for the Sharma-Mittal entropy of exponential families. *Journal of Physics A: Mathematical and Theoretical*, 45:032003, 2012.
- Junier Oliva, Barnabás Póczos, and Jeff Schneider. Distribution to distribution regression. *International Conference on Machine Learning (ICML; JMLR W&CP)*, 28:1049–1057, 2013.
- Junier Oliva, William Neiswanger, Barnabás Póczos, Eric Xing, Hy Trac, Shirley Ho, and Jeff Schneider. Fast function to function regression. *International Conference on Artificial Intelligence and Statistics (AISTATS; JMLR W&CP)*, 38:717–725, 2015.
- Junier B. Oliva, Willie Neiswanger, Barnabás Póczos, Jeff Schneider, and Eric Xing. Fast distribution to real regression. *International Conference on Artificial Intelligence and Statistics (AISTATS; JMLR W&CP)*, 33:706–714, 2014.
- Kalyanapuram R. Parthasarathy. *Probability Measures on Metric Spaces*. Academic Press, 1967.
- George Pedrick. Theory of reproducing kernels for Hilbert spaces of vector valued functions. Technical report, 1957.
- Wei Ping, Ye Xu, Kexin Ren, Chi-Hung Chi, and Shen Furao. Non-I.I.D. multi-instance dimensionality reduction by learning a maximum bag margin subspace. In *AAAI Conference on Artificial Intelligence*, pages 551–556, 2010.
- Barnabás Póczos, Liang Xiong, and Jeff Schneider. Nonparametric divergence estimation with applications to machine learning on distributions. In *Uncertainty in Artificial Intelligence (UAI)*, pages 599–608, 2011.
- Barnabás Póczos, Liang Xiong, Dougal Sutherland, and Jeff Schneider. Support distribution machines. Technical report, Carnegie Mellon University, 2012. (<http://arxiv.org/abs/1202.0302>).
- Barnabás Póczos, Alessandro Rinaldo, Aarti Singh, and Larry Wasserman. Distribution-free distribution regression. *International Conference on Artificial Intelligence and Statistics (AISTATS; JMLR W&CP)*, 31:507–515, 2013.
- Jan Ramon and Maurice Bruynooghe. A polynomial time computable metric between point sets. *Acta Informatica*, 37:765–780, 2001.
- James O. Ramsay and Bernard W. Silverman. *Applied Functional Data Analysis*. Springer Verlag, New York, 2002.
- James O. Ramsay and Bernard W. Silverman. *Functional Data Analysis*. Springer Verlag, New York, 2005.
- Sounya Ray and David Page. Multiple instance regression. In *International Conference on Machine Learning (ICML)*, pages 425–432, 2001.
- Vikas C. Raykar, Balaji Krishnapuram, Jinbo Bi, Murat Dundar, and R. Bharat Rao. Bayesian multiple instance learning: Automatic feature selection and inductive transfer. In *International Conference on Machine Learning (ICML)*, pages 808–815, 2008.
- Peter Richtárik and Martin Takáč. Distributed coordinate descent method for learning with big data. *Journal of Machine Learning Research*, 17:1–25, 2016.
- R. Tyrnell Rockafellar and Roger J-B Wets. *Variational Analysis*. Springer, 2008.
- Alessandro Rudi, Raffaello Camoriano, and Lorenzo Rosasco. Less is more: Nyström computational regularization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1648–1656, 2015.
- Sivan Sabato and Naftali Tishby. Multi-instance learning with any hypothesis class. *Journal of Machine Learning Research*, 13:2999–3039, 2012.
- Laurent Schwartz. *Analyse III, Calcul Intégral*. Hermann, 1998.
- Steve Smale and Ding-Xuan Zhou. Estimating the approximation error in learning theory. *Analysis and Applications*, 1:17–41, 2003.

- Bharath Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Gert Lanckriet, and Bernhard Schölkopf. Hilbert space embeddings and metrics on probability measures. *Journal of Machine Learning Research*, 11:1517–1561, 2010.
- Bharath K. Sriperumbudur, Kenji Fukumizu, and Gert R. G. Lanckriet. Universality, characteristic kernels and RKHS embedding of measures. *Journal of Machine Learning Research*, 12:2389–2410, 2011.
- Bharath K. Sriperumbudur, Kenji Fukumizu, Revant Kumar, Arthur Gretton, and Aapo Hyvärinen. Density estimation in infinite dimensional exponential families. Technical report, 2014. (<http://arxiv.org/pdf/1312.3516>).
- Ingo Steinwart and Andres Christmann. *Support Vector Machines*. Springer, 2008.
- Ingo Steinwart and Clint Scovel. Mercer’s theorem on general domains: On the interaction between measures, kernels, and RKHSs. *Constructive Approximation*, 35:363–417, 2012.
- Ingo Steinwart, Don R. Hush, and Clint Scovel. Optimal rates for regularized least squares regression. In *Conference on Learning Theory (COLT)*, 2009.
- Hongwei Sun and Qiang Wu. Application of integral operator for regularized least-square regression. *Mathematical and Computer Modelling*, 49:276–285, 2009a.
- Hongwei Sun and Qiang Wu. A note on application of integral operator in learning theory. *Applied and Computational Harmonic Analysis*, 26:416–421, 2009b.
- Xu Sun, Hisashi Kashima, and Naonori Ueda. Large-scale personalized human activity recognition using online multitask learning. *IEEE Transactions on Knowledge and Data Engine*, 25:2551–2563, 2013.
- Yu-Yin Sun, Michael K. Ng, and Zhi-Hua Zhou. Multi-instance dimensionality reduction. In *AAAI Conference on Artificial Intelligence*, pages 587–592, 2010.
- Dougal J. Sutherland, Junior B. Oliva, Barnabás Póczos, and Jeff Schneider. Linear-time learning on distributions with approximate kernel embeddings. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 2073–2079, 2016.
- Zoltán Szabó, Arthur Gretton, Barnabás Póczos, and Bharath Sriperumbudur. Two-stage sampled learning theory on distributions. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 948–957, 2015.
- Leslie Valiant. A theory of the learnable. *Communications of the ACM*, 27:1134–1142, 1984.
- Fei Wang, Tanveer Syeda-Mahmood, Baba C. Yenerli, David Beymer, and Anand Rangarajan. Closed-form Jensen-Rényi divergence for mixture of Gaussians and applications to group-wise shape registration. *Medical Image Computing and Computer-Assisted Intervention*, 12:648–655, 2009.
- Jun Wang and Jean-Daniel Zucker. Solving the multiple-instance problem: A lazy learning approach. In *International Conference on Machine Learning (ICML)*, pages 1119–1126, 2000.
- Larry Wasserman. *All of Nonparametric Statistics*. Springer, 2006.
- Ou Wu, Jun Gao, Weiming Hu, Bing Li, and Minglang Zhu. Identifying multi-instance outliers. In *SIAM International Conference on Data Mining (SDM)*, pages 430–441, 2010.
- Yun Yang, Mert Pilanci, and Martin J. Wainwright. Randomized sketches for kernels: Fast and optimal non-parametric regression. *Annals of Statistics*, 2016. (to appear; [arXiv: http://arxiv.org/abs/1501.06195](http://arxiv.org/abs/1501.06195)).
- Amelia Zafra, Mykola Pechenizkiy, and Sebastián Ventura. HyDR-MI: A hybrid algorithm to reduce dimensionality in multiple instance learning. *Information Sciences*, 222:282–301, 2013.
- Dan Zhang and Luo Si. Multiple instance transfer learning. In *IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 406–411, 2009.
- Dan Zhang, Fei Wang, Luo Si, and Tao Li. M3IC: Maximum margin multiple instance clustering. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 1339–1344, 2009.
- Dan Zhang, Fei Wang, Luo Si, and Tao Li. Maximum margin multiple instance clustering with applications to image and text clustering. *IEEE Transactions on Neural Networks*, 22:739–751, 2011.
- Dan Zhang, Jingrui He, Luo Si, and Richard D. Lawrence. MILEAGE: Multiple Instance Learning with Global Embedding. *International Conference on Machine Learning (ICML: JMLR W&CP)*, 28:82–90, 2013.
- Min-Ling Zhang and Zhi-Hua Zhou. Multi-instance clustering with applications to multi-instance prediction. *Applied Intelligence*, 31:47–68, 2009.
- Yuchen Zhang, John C. Duchi, and Martin J. Wainwright. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. 16:3299–3340, 2015.
- Jiayu Zhou, Jun Liu, Vaibhav A. Narayan, and Jieping Ye. Modeling disease progression via multi-task learning. *NeuroImage*, 78:233–248, 2013.
- Shaohua Kevin Zhou and Rama Chellappa. From sample similarity to ensemble similarity: Probabilistic distance measures in reproducing kernel Hilbert space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:917–929, 2006.
- Zhi-Hua Zhou. Multi-instance learning: A survey. Technical report, AI Lab, Department of Computer Science & Technology, Nanjing University, Nanjing, China, 2004. (<http://cs.nju.edu.cn/zhouch/zhouch.files/publication/techrep04.pdf>).
- Zhi-Hua Zhou, Yu-Yin Sun, and Yu-Feng Li. Multi-instance learning by treating instances as non-IID samples. In *International Conference on Machine Learning (ICML)*, pages 1249–1256, 2009.

A Differential Equation for Modeling Nesterov's Accelerated Gradient Method: Theory and Insights

Weijie Su

*Department of Statistics
University of Pennsylvania
Philadelphia, PA 19104, USA*

SUW@WHARTON.UPENN.EDU

Stephen Boyd

*Department of Electrical Engineering
Stanford University
Stanford, CA 94305, USA*

BOYD@STANFORD.EDU

Emmanuel J. Candès

*Departments of Statistics and Mathematics
Stanford University
Stanford, CA 94305, USA*

CANDES@STANFORD.EDU

Editor: Yoram Singer

Abstract

We derive a second-order ordinary differential equation (ODE) which is the limit of Nesterov's accelerated gradient method. This ODE exhibits approximate equivalence to Nesterov's scheme and thus can serve as a tool for analysis. We show that the continuous time ODE allows for a better understanding of Nesterov's scheme. As a byproduct, we obtain a family of schemes with similar convergence rates. The ODE interpretation also suggests restarting Nesterov's scheme leading to an algorithm, which can be rigorously proven to converge at a linear rate whenever the objective is strongly convex.

Keywords: Nesterov's accelerated scheme, convex optimization, first-order methods, differential equation, restarting

1. Introduction

In many fields of machine learning, minimizing a convex function is at the core of efficient model estimation. In the simplest and most standard form, we are interested in solving

$$\text{minimize } f(x),$$

where f is a convex function, smooth or non-smooth, and $x \in \mathbb{R}^n$ is the variable. Since Newton, numerous algorithms and methods have been proposed to solve the minimization problem, notably gradient and subgradient descent, Newton's methods, trust region methods, conjugate gradient methods, and interior point methods (see e.g. Polyak, 1987; Boyd and Vandenberghe, 2004; Nocedal and Wright, 2006; Ruszczyński, 2006; Boyd et al., 2011; Shor, 2012; Beck, 2014, for expositions).

First-order methods have regained popularity as data sets and problems are ever increasing in size and, consequently, there has been much research on the theory and practice

of accelerated first-order schemes. Perhaps the earliest first-order method for minimizing a convex function f is the gradient method, which dates back to Euler and Lagrange. Thirty years ago, however, in a seminal paper Nesterov proposed an accelerated gradient method (Nesterov, 1983), which may take the following form: starting with x_0 and $y_0 = x_0$, inductively define

$$\begin{aligned} x_k &= y_{k-1} - s \nabla f(y_{k-1}) \\ y_k &= x_k + \frac{k-1}{k+2} (x_k - x_{k-1}). \end{aligned} \quad (1)$$

For any fixed step size $s \leq 1/L$, where L is the Lipschitz constant of ∇f , this scheme exhibits the convergence rate

$$f(x_k) - f^* \leq O\left(\frac{\|x_0 - x^*\|^2}{sk^2}\right). \quad (2)$$

Above, x^* is any minimizer of f and $f^* = f(x^*)$. It is well-known that this rate is optimal among all methods having only information about the gradient of f at consecutive iterates (Nesterov, 2004). This is in contrast to vanilla gradient descent methods, which have the same computational complexity but can only achieve a rate of $O(1/k)$. This improvement relies on the introduction of the momentum term $x_k - x_{k-1}$ as well as the particularly tuned coefficient $(k-1)/(k+2) \approx 1-3/k$. Since the introduction of Nesterov's scheme, there has been much work on the development of first-order accelerated methods, see Nesterov (2004, 2005, 2013) for theoretical developments, and Tseng (2008) for a unified analysis of these ideas. Notable applications can be found in sparse linear regression (Beck and Teboulle, 2009; Qin and Goldfarb, 2012), compressed sensing (Becker et al., 2011) and, deep and recurrent neural networks (Sutskever et al., 2013).

In a different direction, there is a long history relating ordinary differential equation (ODEs) to optimization, see Helmke and Moore (1996), Schropp and Singer (2000), and Fiori (2005) for example. The connection between ODEs and numerical optimization is often established via taking step sizes to be very small so that the trajectory or solution path converges to a curve modeled by an ODE. The conciseness and well-established theory of ODEs provide deeper insights into optimization, which has led to many interesting findings. Notable examples include linear regression via solving differential equations induced by linearized Bregman iteration algorithm (Osher et al., 2014), a continuous-time Nesterov-like algorithm in the context of control design (Dürri and Ebenbauer, 2012; Dürri et al., 2012), and modeling design iterative optimization algorithms as nonlinear dynamical systems (Lessard et al., 2014).

In this work, we derive a second-order ODE which is the exact limit of Nesterov's scheme by taking small step sizes in (1); to the best of our knowledge, this work is the first to use ODEs to model Nesterov's scheme or its variants in this limit. One surprising fact in connection with this subject is that a *first-order* scheme is modeled by a *second-order* ODE. This ODE takes the following form:

$$\ddot{X} + \frac{3}{t} \dot{X} + \nabla f(X) = 0 \quad (3)$$

for $t > 0$, with initial conditions $X(0) = x_0, \dot{X}(0) = 0$; here, x_0 is the starting point in Nesterov's scheme, $\dot{X} \equiv dX/dt$ denotes the time derivative or velocity and similarly

$\dot{X} \equiv d^2X/dt^2$ denotes the acceleration. The time parameter in this ODE is related to the step size in (1) via $t \approx k\sqrt{s}$. Expectedly, it also enjoys inverse quadratic convergence rate as its discrete analog,

$$f(X(t)) - f^* \leq O\left(\frac{\|x_0 - x^*\|^2}{t^2}\right).$$

Approximate equivalence between Nesterov's scheme and the ODE is established later in various perspectives, rigorous and intuitive. In the main body of this paper, examples and case studies are provided to demonstrate that the homogeneous and conceptually simpler ODE can serve as a tool for understanding, analyzing and generalizing Nesterov's scheme.

In the following, two insights of Nesterov's scheme are highlighted, the first one on oscillations in the trajectories of this scheme, and the second on the peculiar constant 3 appearing in the ODE.

1.1 From Overdamping to Underdamping

In general, Nesterov's scheme is not monotone in the objective function value due to the introduction of the momentum term. Oscillations or overshoots along the trajectory of iterates approaching the minimizer are often observed when running Nesterov's scheme. Figure 1 presents typical phenomena of this kind, where a two-dimensional convex function is minimized by Nesterov's scheme. Viewing the ODE as a damping system, we obtain interpretations as follows.

Small t . In the beginning, the damping ratio $3/t$ is large. This leads the ODE to be an overdamped system, returning to the equilibrium without oscillating.

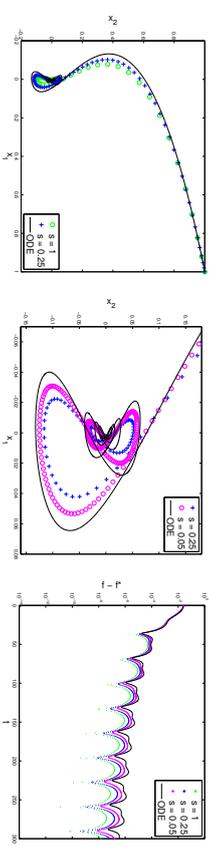
Large t . As t increases, the ODE with a small $3/t$ behaves like an underdamped system, oscillating with the amplitude gradually decreasing to zero.

As depicted in Figure 1a, in the beginning the ODE curve moves smoothly towards the origin, the minimizer x^* . The second interpretation "Large t " provides partial explanation for the oscillations observed in Nesterov's scheme at later stage. Although our analysis extends farther, it is similar in spirit to that carried in O'Donoghue and Candès (2013). In particular, the zoomed Figure 1b presents some butterfly-like oscillations for both the scheme and ODE. There, we see that the trajectory constantly moves away from the origin and returns back later. Each overshoot in Figure 1b causes a bump in the function values, as shown in Figure 1c. We observe also from Figure 1c that the periodicity captured by the bumps are very close to that of the ODE solution. In passing, it is worth mentioning that the solution to the ODE in this case can be expressed via Bessel functions, hence enabling quantitative characterizations of these overshoots and bumps, which are given in full detail in Section 3.

1.2 A Phase Transition

The constant 3, derived from $(k+2) - (k-1)$ in (3), is not haphazard. In fact, it is the smallest constant that guarantees $O(1/t^2)$ convergence rate. Specifically, parameterized by a constant r , the generalized ODE

$$\ddot{X} + \frac{r}{t}\dot{X} + \nabla f(X) = 0$$



(a) Trajectories.

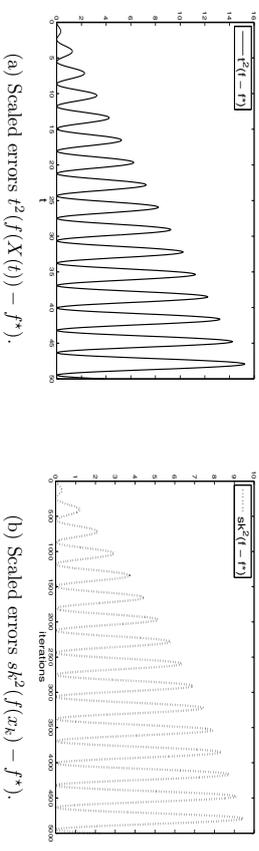
(b) Zoomed trajectories.

(c) Errors $f - f^*$.

Figure 1: Minimizing $f = 2 \times 10^{-2}x_1^2 + 5 \times 10^{-3}x_2^2$, starting from $x_0 = (1, 1)$. The black and solid curves correspond to the solution to the ODE. In (c), for the x-axis we use the identification between time and iterations, $t = k\sqrt{s}$.

can be translated into a generalized Nesterov's scheme that is the same as the original (1) except for $(k-1)/(k+2)$ being replaced by $(k-1)/(k+r-1)$. Surprisingly, for both generalized ODEs and schemes, the inverse quadratic convergence is guaranteed if and only if $r \geq 3$. This phase transition suggests there might be deep causes for acceleration among first-order methods. In particular, for $r \geq 3$, the worst case constant in this inverse quadratic convergence rate is minimized at $r = 3$.

Figure 2 illustrates the growth of $t^2(f(X(t)) - f^*)$ and $sk^2(f(x_k) - f^*)$, respectively, for the generalized ODE and scheme with $r = 1$, where the objective function is simply $f(x) = \frac{1}{2}x^2$. Inverse quadratic convergence fails to be observed in both Figures 2a and 2b, where the scaled errors grow with t or iterations, for both the generalized ODE and scheme.



(a) Scaled errors $t^2(f(X(t)) - f^*)$.

(b) Scaled errors $sk^2(f(x_k) - f^*)$.

Figure 2: Minimizing $f = \frac{1}{2}x^2$ by the generalized ODE and scheme with $r = 1$, starting from $x_0 = 1$. In (b), the step size $s = 10^{-4}$.

1.3 Outline and Notation

The rest of the paper is organized as follows. In Section 2, the ODE is rigorously derived from Nesterov's scheme, and a generalization to composite optimization, where f may be non-smooth, is also obtained. Connections between the ODE and the scheme, in terms of trajectory behaviors and convergence rates, are summarized in Section 3. In Section

4, we discuss the effect of replacing the constant 3 in (3) by an arbitrary constant on the convergence rate. A new restarting scheme is suggested in Section 5, with linear convergence rate established and empirically observed.

Some standard notations used throughout the paper are collected here. We denote by \mathcal{F}_L the class of convex functions f with L -Lipschitz continuous gradients defined on \mathbb{R}^n , i.e., f is convex, continuously differentiable, and satisfies

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$$

for any $x, y \in \mathbb{R}^n$, where $\|\cdot\|$ is the standard Euclidean norm and $L > 0$ is the Lipschitz constant. Next, \mathcal{S}_μ denotes the class of μ -strongly convex functions f on \mathbb{R}^n with continuous gradients, i.e., f is continuously differentiable and $f(x) - \mu\|x\|^2/2$ is convex. We set $\mathcal{S}_{\mu,L} = \mathcal{F}_L \cap \mathcal{S}_\mu$.

2. Derivation

First, we sketch an informal derivation of the ODE (3). Assume $f \in \mathcal{F}_L$ for $L > 0$. Combining the two equations of (1) and applying a rescaling gives

$$\frac{x_{k+1} - x_k}{\sqrt{s}} = \frac{k-1}{k+2} \frac{x_k - x_{k-1}}{\sqrt{s}} - \sqrt{s} \nabla f(y_k). \quad (4)$$

Introduce the Ansatz $x_k \approx X(k\sqrt{s})$ for some smooth curve $X(t)$ defined for $t \geq 0$. Put $k = t/\sqrt{s}$. Then as the step size s goes to zero, $X(t) \approx x_{t/\sqrt{s}} = x_k$ and $X(t + \sqrt{s}) \approx x_{(t+\sqrt{s})/\sqrt{s}} = x_{k+1}$, and Taylor expansion gives

$$(x_{k+1} - x_k)/\sqrt{s} = \dot{X}(t) + \frac{1}{2} \ddot{X}(t)\sqrt{s} + o(\sqrt{s}), \quad (x_k - x_{k-1})/\sqrt{s} = \dot{X}(t) - \frac{1}{2} \ddot{X}(t)\sqrt{s} + o(\sqrt{s})$$

and $\sqrt{s} \nabla f(y_k) = \sqrt{s} \nabla f(X(t)) + o(\sqrt{s})$. Thus (4) can be written as

$$\begin{aligned} \dot{X}(t) + \frac{1}{2} \ddot{X}(t)\sqrt{s} + o(\sqrt{s}) \\ = \left(1 - \frac{3\sqrt{s}}{t}\right) \left(\dot{X}(t) - \frac{1}{2} \ddot{X}(t)\sqrt{s} + o(\sqrt{s})\right) - \sqrt{s} \nabla f(X(t)) + o(\sqrt{s}). \end{aligned} \quad (5)$$

By comparing the coefficients of \sqrt{s} in (5), we obtain

$$\ddot{X} + \frac{3}{t} \dot{X} + \nabla f(X) = 0.$$

The first initial condition is $X(0) = x_0$. Taking $k = 1$ in (4) yields

$$(x_2 - x_1)/\sqrt{s} = -\sqrt{s} \nabla f(y_1) = o(1).$$

Hence, the second initial condition is simply $\dot{X}(0) = 0$ (vanishing initial velocity).

One popular alternative momentum coefficient is $\theta_k(\theta_{k-1}^{-1} - 1)$, where θ_k are iteratively defined as $\theta_{k+1} = \left(\sqrt{\theta_k^4 + 4\theta_k^2 - \theta_k^2}\right)/2$, starting from $\theta_0 = 1$ (Nesterov, 1983; Beck and

Teboulle, 2009). Simple analysis reveals that $\theta_k(\theta_{k-1}^{-1} - 1)$ asymptotically equals $1 - 3/k + O(1/k^2)$, thus leading to the same ODE as (1).

Classical results in ODE theory do not directly imply the existence or uniqueness of the solution to this ODE because the coefficient $3/t$ is singular at $t = 0$. In addition, ∇f is typically not analytic at x_0 , which leads to the inapplicability of the power series method for studying singular ODEs. Nevertheless, the ODE is well posed: the strategy we employ for showing this constructs a series of ODEs approximating (3), and then chooses a convergent subsequence by some compactness arguments such as the Arzelà-Ascoli theorem. Below, $C^2((0, \infty); \mathbb{R}^n)$ denotes the class of twice continuously differentiable maps from $(0, \infty)$ to \mathbb{R}^n ; similarly, $C^1([0, \infty); \mathbb{R}^n)$ denotes the class of continuously differentiable maps from $[0, \infty)$ to \mathbb{R}^n .

Theorem 1 For any $f \in \mathcal{F}_\infty := \cup_{L>0} \mathcal{F}_L$ and any $x_0 \in \mathbb{R}^n$, the ODE (3) with initial conditions $X(0) = x_0, \dot{X}(0) = 0$ has a unique global solution $X \in C^2((0, \infty); \mathbb{R}^n) \cap C^1([0, \infty); \mathbb{R}^n)$.

The next theorem, in a rigorous way, guarantees the validity of the derivation of this ODE. The proofs of both theorems are deferred to the appendices.

Theorem 2 For any $f \in \mathcal{F}_\infty$, as the step size $s \rightarrow 0$, Nesterov's scheme (1) converges to the ODE (3) in the sense that for all fixed $T > 0$,

$$\lim_{s \rightarrow 0} \max_{0 \leq k \leq \frac{T}{\sqrt{s}}} \|x_k - X(k\sqrt{s})\| = 0.$$

2.1 Simple Properties

We collect some elementary properties that are helpful in understanding the ODE.

Time Invariance. If we adopt a linear time transformation, $\tilde{t} = ct$ for some $c > 0$, by the chain rule it follows that

$$\frac{dX}{d\tilde{t}} = \frac{1}{c} \frac{dX}{dt}, \quad \frac{d^2X}{d\tilde{t}^2} = \frac{1}{c^2} \frac{d^2X}{dt^2}.$$

This yields the ODE parameterized by \tilde{t} ,

$$\frac{d^2X}{d\tilde{t}^2} + \frac{3}{\tilde{t}} \frac{dX}{d\tilde{t}} + \nabla f(X)/c^2 = 0.$$

Also note that minimizing f/c^2 is equivalent to minimizing f . Hence, the ODE is invariant under the time change. In fact, it is easy to see that time invariance holds if and only if the coefficient of \dot{X} has the form C/t for some constant C .

Rotational Invariance. Nesterov's scheme and other gradient-based schemes are invariant under rotations. As expected, the ODE is also invariant under orthogonal transformation. To see this, let $Y = QX$ for some orthogonal matrix Q . This leads to $\dot{Y} = Q\dot{X}$, $\ddot{Y} = Q\ddot{X}$ and $\nabla_Y f = Q\nabla_X f$. Hence, denoting by Q^T the transpose of Q , the ODE in the new coordinate system reads $Q^T \ddot{Y} + \frac{3}{\tilde{t}} Q^T \dot{Y} + Q^T \nabla_Y f = 0$, which is of the same form as (3) once multiplying Q on both sides.

Initial Asymptotic. Assume sufficient smoothness of X such that $\lim_{t \rightarrow 0} \ddot{X}(t)$ exists. The mean value theorem guarantees the existence of some $\xi \in (0, t)$ that satisfies $\dot{X}(t)/t = (\dot{X}(t) - \dot{X}(0))/t = \ddot{X}(\xi)$. Hence, from the ODE we deduce $\dot{X}(t) + 3\dot{X}(\xi) + \nabla f(X(t)) = 0$.

Taking the limit $t \rightarrow 0$ gives $\dot{X}(0) = -\nabla f(x_0)/4$. Hence, for small t we have the asymptotic form:

$$X(t) = -\frac{\nabla f(x_0)t^2}{8} + x_0 + o(t^2).$$

This asymptotic expansion is consistent with the empirical observation that Nesterov's scheme moves slowly in the beginning.

2.2 ODE for Composite Optimization

It is interesting and important to generalize the ODE to minimizing f in the composite form $f(x) = g(x) + h(x)$, where the smooth part $g \in \mathcal{F}_L$ and the non-smooth part $h : \mathbb{R}^n \rightarrow (-\infty, \infty]$ is a structured general convex function. Both Nesterov (2013) and Beck and Teboulle (2009) obtain $O(1/k^2)$ convergence rate by employing the proximal structure of h . In analogy to the smooth case, an ODE for composite f is derived in the appendix.

3. Connections and Interpretations

In this section, we explore the approximate equivalence between the ODE and Nesterov's scheme, and provide evidence that the ODE can serve as an amenable tool for interpreting and analyzing Nesterov's scheme. The first subsection exhibits inverse quadratic convergence rate for the ODE solution, the next two address the oscillation phenomenon discussed in Section 1.1, and the last subsection is devoted to comparing Nesterov's scheme with gradient descent from a numerical perspective.

3.1 Analogous Convergence Rate

The original result from Nesterov (1983) states that, for any $f \in \mathcal{F}_L$, the sequence $\{x_k\}$ given by (1) with step size $s \leq 1/L$ satisfies

$$f(x_k) - f^* \leq \frac{2\|x_0 - x^*\|^2}{s(k+1)^2}. \quad (6)$$

Our next result indicates that the trajectory of (3) closely resembles the sequence $\{x_k\}$ in terms of the convergence rate to a minimizer x^* . Compared with the discrete case, this proof is shorter and simpler.

Theorem 3 *For any $f \in \mathcal{F}_\infty$, let $X(t)$ be the unique global solution to (3) with initial conditions $X(0) = x_0, \dot{X}(0) = 0$. Then, for any $t > 0$,*

$$f(X(t)) - f^* \leq \frac{2\|x_0 - x^*\|^2}{t^2}. \quad (7)$$

Our next result indicates that the trajectory of (3) closely resembles the sequence $\{x_k\}$ in terms of the convergence rate to a minimizer x^* . Compared with the discrete case, this proof is shorter and simpler.

$$\mathcal{E} = 2t(f(X) - f^*) + t^2 \langle \nabla f, \dot{X} \rangle + 4 \left\langle X + \frac{t}{2} \dot{X} - x^*, \frac{3}{2} \dot{X} + \frac{t}{2} \ddot{X} \right\rangle.$$

Proof Consider the energy functional¹ defined as $\mathcal{E}(t) = t^2(f(X(t)) - f^*) + 2\|X + t\dot{X}/2 - x^*\|^2$, whose time derivative is

$$\dot{\mathcal{E}} = 2t(f(X) - f^*) + t^2 \langle \nabla f, \dot{X} \rangle + 4 \left\langle X + \frac{t}{2} \dot{X} - x^*, \frac{3}{2} \dot{X} + \frac{t}{2} \ddot{X} \right\rangle.$$

1. We may also view this functional as the negative entropy. Similarly, for the gradient flow $\dot{X} + \nabla f(X) = 0$, an energy function of form $\mathcal{E}_{\text{gradient}}(t) = t(f(X(t)) - f^*) + \|X(t) - x^*\|^2/2$ can be used to derive the bound $f(X(t)) - f^* \leq \frac{\|x_0 - x^*\|^2}{2t}$.

Substituting $3\dot{X}/2 + t\ddot{X}/2$ with $-\nabla f(X)/2$, the above equation gives

$$\dot{\mathcal{E}} = 2t(f(X) - f^*) + 4\langle X - x^*, -\nabla f(X)/2 \rangle = 2t(f(X) - f^*) - 2t\langle X - x^*, \nabla f(X) \rangle \leq 0,$$

where the inequality follows from the convexity of f . Hence by monotonicity of \mathcal{E} and non-negativity of $2\|X + t\dot{X}/2 - x^*\|^2$, the gap satisfies

$$f(X(t)) - f^* \leq \frac{\mathcal{E}(t)}{t^2} \leq \frac{\mathcal{E}(0)}{t^2} = \frac{2\|x_0 - x^*\|^2}{t^2}. \quad \blacksquare$$

Making use of the approximation $t \approx k\sqrt{s}$, we observe that the convergence rate in (6) is essentially a discrete version of that in (7), providing yet another piece of evidence for the approximate equivalence between the ODE and the scheme.

We finish this subsection by showing that the number 2 appearing in the numerator of the error bound in (7) is optimal. Consider an arbitrary $f \in \mathcal{F}_\infty(\mathbb{R})$ such that $f(x) = x$ for $x \geq 0$. Starting from some $x_0 > 0$, the solution to (3) is $X(t) = x_0 - t^2/8$ before hitting the origin. Hence, $t^2(f(X(t)) - f^*) = t^2(x_0 - t^2/8)$ has a maximum $2x_0^2 = 2|x_0 - 0|^2$ achieved at $t = 2\sqrt{x_0}$. Therefore, we cannot replace 2 by any smaller number, and we can expect that this tightness also applies to the discrete analog (6).

3.2 Quadratic f and Bessel Functions

For quadratic f , the ODE (3) admits a solution in closed form. This closed form solution turns out to be very useful in understanding the issues raised in the introduction.

Let $f(x) = \frac{1}{2}\langle x, Ax \rangle + \langle b, x \rangle$, where $A \in \mathbb{R}^{n \times n}$ is a positive semidefinite matrix and b is in the column space of A because otherwise this function can attain $-\infty$. Then a simple translation in x can absorb the linear term $\langle b, x \rangle$ into the quadratic term. Since both the ODE and the scheme move within the affine space perpendicular to the kernel of A , without loss of generality, we assume that A is positive definite, admitting a spectral decomposition $A = Q^T \Lambda Q$, where Λ is a diagonal matrix formed by the eigenvalues. Replacing x with Qx , we assume $f = \frac{1}{2}\langle x, Ax \rangle$ from now on. Now, the ODE for this function admits a simple decomposition of form

$$\ddot{X}_i + \frac{3}{t} \dot{X}_i + \lambda_i X_i = 0, \quad i = 1, \dots, n$$

with $X_i(0) = x_{0,i}, \dot{X}_i(0) = 0$. Introduce $Y_i(u) = uX_i(u/\sqrt{\lambda_i})$, which satisfies

$$u^2 \ddot{Y}_i + u \dot{Y}_i + (u^2 - 1)Y_i = 0.$$

This is Bessel's differential equation of order one. Since Y_i vanishes at $u = 0$, we see that Y_i is a constant multiple of J_1 , the Bessel function of the first kind of order one.² It has an analytic expansion:

$$J_1(u) = \sum_{m=0}^{\infty} \frac{(-1)^m}{(2m)!(2m+2)!} u^{2m+1},$$

2. Up to a constant multiplier, J_1 is the unique solution to the Bessel's differential equation $u^2 \ddot{J}_1 + u \dot{J}_1 + (u^2 - 1)J_1 = 0$ that is finite at the origin. In the analytic expansion of J_1 , $m!!$ denotes the double factorial defined as $m!! = m \times (m-2) \times \dots \times 2$ for even m , or $m!! = m \times (m-2) \times \dots \times 1$ for odd m .

which gives the asymptotic expansion

$$J_1(u) = (1 + o(1)) \frac{u}{2}$$

when $u \rightarrow 0$. Requiring $X_i(0) = x_{0,i}$, hence, we obtain

$$X_i(t) = \frac{2x_{0,i}}{t\sqrt{\lambda_i}} J_1(t\sqrt{\lambda_i}). \quad (8)$$

For large t , the Bessel function has the following asymptotic form (see e.g. Watson, 1995):

$$J_1(t) = \sqrt{\frac{2}{\pi t}} \left(\cos(t - 3\pi/4) + O(1/t) \right). \quad (9)$$

This asymptotic expansion yields (note that $f^* = 0$)

$$f(X(t)) - f^* = f(X(t)) = \sum_{i=1}^n \frac{2x_{0,i}^2}{t^2} J_1^2(t\sqrt{\lambda_i})^2 = O\left(\frac{\|x_0 - x^*\|^2}{t^3 \sqrt{\min \lambda_i}}\right). \quad (10)$$

On the other hand, (9) and (10) give a lower bound:

$$\begin{aligned} \limsup_{t \rightarrow \infty} t^3 (f(X(t)) - f^*) &\geq \lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t u^3 (f(X(u)) - f^*) du \\ &= \lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t \sum_{i=1}^n 2x_{0,i}^2 u J_1^2(u\sqrt{\lambda_i})^2 du \\ &= \sum_{i=1}^n \frac{2x_{0,i}^2}{\pi\sqrt{\lambda_i}} \geq \frac{2\|x_0 - x^*\|^2}{\pi\sqrt{L}}, \end{aligned} \quad (11)$$

where $L = \|A\|_2$ is the spectral norm of A . The first inequality follows by interpreting $\lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t x^3 (f(X(u)) - f^*) du$ as the mean of $x^3 (f(X(u)) - f^*)$ on $(0, \infty)$ in certain sense.

In view of (10), Nesterov's scheme might possibly exhibit $O(1/k^3)$ convergence rate for strongly convex functions. This convergence rate is consistent with the second inequality in Theorem 6. In Section 4.3, we prove the $O(1/t^3)$ rate for a generalized version of (3). However, (11) rules out the possibility of a higher order convergence rate.

Recall that the function considered in Figure 1 is $f(x) = 0.02x_1^2 + 0.005x_2^2$, starting from $x_0 = (1, 1)$. As the step size s becomes smaller, the trajectory of Nesterov's scheme converges to the solid curve represented via the Bessel function. While approaching the minimizer x^* , each trajectory displays the oscillation pattern, as well-captured by the zoomed Figure 1b. This prevents Nesterov's scheme from achieving better convergence rate. The representation (8) offers excellent explanation as follows. Denote by T_1, T_2 , respectively, the approximate periodicities of the first component $|X_1|$ in absolute value and the second $|X_2|$. By (9), we get $T_1 = \pi/\sqrt{\lambda_1} = 5\pi$ and $T_2 = \pi/\sqrt{\lambda_2} = 10\pi$. Hence, as the amplitude gradually decreases to zero, the function $f = 2x_{0,1}^2 J_1(\sqrt{\lambda_1}t)^2/t^2 + 2x_{0,2}^2 J_1(\sqrt{\lambda_2}t)^2/t^2$ has a major cycle of 10π , the least common multiple of T_1 and T_2 . A careful look at Figure 1c reveals that within each major bump, roughly, there are $10\pi/T_1 = 2$ minor peaks.

3.3 Fluctuations of Strongly Convex f

The analysis carried out in the previous subsection only applies to convex quadratic functions. In this subsection, we extend the discussion to one-dimensional strongly convex functions. The Sturm-Picone theory (see e.g. Hinton, 2005) is extensively used all along the analysis.

Let $f \in \mathcal{S}_{\mu,L}(\mathbb{R})$. Without loss of generality, assume f attains minimum at $x^* = 0$. Then, by definition $\mu \leq f'(x)/x \leq L$ for any $x \neq 0$. Denoting by X the solution to the ODE (3), we consider the self-adjoint equation,

$$(t^3 Y')' + \frac{t^3 f'(X(t))}{X(t)} Y = 0, \quad (12)$$

which, apparently, admits a solution $Y(t) = X(t)$. To apply the Sturm-Picone comparison theorem, consider

$$(t^3 Y')' + \mu t^3 Y = 0$$

for a comparison. This equation admits a solution $\tilde{Y}(t) = J_1(\sqrt{\mu}t)/t$. Denote by $\tilde{t}_1 < \tilde{t}_2 < \dots$ all the positive roots of $J_1(t)$, which satisfy (see e.g. Watson, 1995)

$$3.8317 = \tilde{t}_1 - \tilde{t}_0 > \tilde{t}_2 - \tilde{t}_3 > \tilde{t}_3 - \tilde{t}_4 > \dots > \pi,$$

where $\tilde{t}_0 = 0$. Then, it follows that the positive roots of \tilde{Y} are $\tilde{t}_1/\sqrt{\mu}, \tilde{t}_2/\sqrt{\mu}, \dots$. Since $t^3 f'(X(t))/X(t) \geq \mu t^3$, the Sturm-Picone comparison theorem asserts that $X(t)$ has a root in each interval $[\tilde{t}_i/\sqrt{\mu}, \tilde{t}_{i+1}/\sqrt{\mu}]$.

To obtain a similar result in the opposite direction, consider

$$(t^3 Y')' + Lt^3 Y = 0. \quad (13)$$

Applying the Sturm-Picone comparison theorem to (12) and (13), we ensure that between any two consecutive positive roots of X , there is at least one \tilde{t}_i/\sqrt{L} . Now, we summarize our findings in the following. Roughly speaking, this result concludes that the oscillation frequency of the ODE solution is between $O(\sqrt{\mu})$ and $O(\sqrt{L})$.

Theorem 4 *Denote by $0 < t_1 < t_2 < \dots$ all the roots of $X(t) - x^*$. Then these roots satisfy, for all $i \geq 1$,*

$$t_1 < \frac{7.6635}{\sqrt{\mu}}, \quad t_{i+1} - t_i < \frac{7.6635}{\sqrt{\mu}}, \quad t_{i+2} - t_i > \frac{\pi}{\sqrt{L}}.$$

3.4 Nesterov's Scheme Compared with Gradient Descent

The ansatz $t \approx k\sqrt{s}$ in relating the ODE and Nesterov's scheme is formally confirmed in Theorem 2. Consequently, for any constant $t_c > 0$, this implies that x_k does not change much for a range of step sizes s if $k \approx t_c/\sqrt{s}$. To empirically support this claim, we present an example in Figure 3a, where the scheme minimizes $f(x) = \|y - Ax\|_2^2/2 + \|x\|_1$ with $y = (4, 2, 0)$ and $A(:, 1) = (0, 2, 4)$, $A(:, 2) = (1, 1, 1)$ starting from $x_0 = (2, 0)$ (here $A(:, j)$ is the j th column of A). From this figure, we are delighted to observe that x_k with the same t_c are very close to each other.

This interesting square-root scaling has the potential to shed light on the superiority of Nesterov's scheme over gradient descent. Roughly speaking, each iteration in Nesterov's scheme amounts to traveling \sqrt{s} in time along the integral curve of (3), whereas it is known that the simple gradient descent $x_{k+1} = x_k - s\nabla f(x_k)$ moves s along the integral curve of $\dot{X} + \nabla f(X) = 0$. We expect that for small s Nesterov's scheme moves more in each iteration since \sqrt{s} is much larger than s . Figure 3b illustrates and supports this claim, where the function minimized is $f = |x_1|^3 + 5|x_2|^3 + 0.001(x_1 + x_2)^2$ with step size $s = 0.05$ (The coordinates are appropriately rotated to allow x_0 and x^* lie on the same horizontal line). The circles are the iterates for $k = 1, 10, 20, 30, 45, 60, 90, 120, 150, 190, 250, 300$. For Nesterov's scheme, the seventh iterates for $k = 15$, while for gradient descent the last point has merely arrived at $t = 15$.

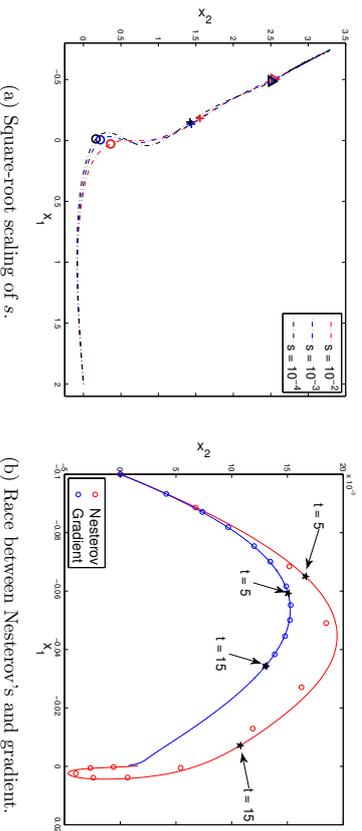


Figure 3: In (a), the circles, crosses and triangles are x_k evaluated at $k = \lceil 1/\sqrt{s} \rceil, \lceil 2/\sqrt{s} \rceil$ and $\lceil 3/\sqrt{s} \rceil$, respectively. In (b), the circles are iterations given by Nesterov's scheme or gradient descent, depending on the color, and the stars are $X(t)$ on the integral curves for $t = 5, 15$.

A second look at Figure 3b suggests that Nesterov's scheme allows a large deviation from its limit curve, as compared with gradient descent. This raises the question of the stable step size allowed for numerically solving the ODE (3) in the presence of accumulated errors. The finite difference approximation by the forward Euler method is

$$\frac{X(t + \Delta t) - 2X(t) + X(t - \Delta t)}{\Delta t^2} + \frac{3}{t} \frac{X(t) - X(t - \Delta t)}{\Delta t} + \nabla f(X(t)) = 0, \quad (14)$$

which is equivalent to

$$X(t + \Delta t) = \left(2 - \frac{3\Delta t}{t}\right)X(t) - \Delta t^2 \nabla f(X(t)) - \left(1 - \frac{3\Delta t}{t}\right)X(t - \Delta t). \quad (15)$$

Assuming f is sufficiently smooth, we have $\nabla f(x + \delta x) \approx \nabla f(x) + \nabla^2 f(x)\delta x$ for small perturbations δx , where $\nabla^2 f(x)$ is the Hessian of f evaluated at x . Identifying $k = t/\Delta t$,

the characteristic equation of this finite difference scheme is approximately

$$\det \left(\lambda^2 - \left(2 - \Delta t^2 \nabla^2 f - \frac{3\Delta t}{t}\right) \lambda + 1 - \frac{3\Delta t}{t} \right) = 0. \quad (16)$$

The numerical stability of (14) with respect to accumulated errors is equivalent to this: all the roots of (16) lie in the unit circle (see e.g. Leader, 2004). When $\nabla^2 f \leq L I_n$ (i.e. $L I_n - \nabla^2 f$ is positive semidefinite), if $\Delta t/t$ small and $\Delta t < 2/\sqrt{L}$, we see that all the roots of (16) lie in the unit circle. On the other hand, if $\Delta t > 2/\sqrt{L}$, (16) can possibly have a root λ outside the unit circle, causing numerical instability. Under our identification $s = \Delta t^2$, a step size of $s = 1/L$ in Nesterov's scheme (1) is approximately equivalent to a step size of $\Delta t = 1/\sqrt{L}$ in the forward Euler method, which is stable for numerically integrating (14). As a comparison, note that the finite difference scheme of the ODE $\dot{X}(t) + \nabla f(X(t)) = 0$, which models gradient descent with updates $x_{k+1} = x_k - s\nabla f(x_k)$, has the characteristic equation $\det(\lambda - (1 - \Delta t \nabla^2 f)) = 0$. Thus, to guarantee $-I_n \leq 1 - \Delta t \nabla^2 f \leq I_n$ in worst case analysis, one can only choose $\Delta t \leq 2/L$ for a fixed step size, which is much smaller than the step size $2/\sqrt{L}$ for (14) when ∇f is very variable, i.e., L is large.

4. The Magic Constant 3

Recall that the constant 3 appearing in the coefficient of \dot{X} in (3) originates from $(k+2) - (k-1) = 3$. This number leads to the momentum coefficient in (1) taking the form $(k-1)/(k+2) = 1 - 3/k + O(1/k^2)$. In this section, we demonstrate that 3 can be replaced by any larger number, while maintaining the $O(1/k^2)$ convergence rate. To begin with, let us consider the following ODE parameterized by a constant r :

$$\ddot{X} + \frac{r}{t} \dot{X} + \nabla f(X) = 0 \quad (17)$$

with initial conditions $X(0) = x_0, \dot{X}(0) = 0$. The proof of Theorem 1, which seamlessly applies here, guarantees the existence and uniqueness of the solution X to this ODE.

Interpreting the damping ratio r/t as a measure of friction³ in the damping system, our results say that more friction does not end the $O(1/t^2)$ and $O(1/k^2)$ convergence rate. On the other hand, in the lower friction setting, where r is smaller than 3, we can no longer expect inverse quadratic convergence rate, unless some additional structures of f are imposed. We believe that this striking phase transition at 3 deserves more attention as an interesting research challenge.

4.1 High Friction

Here, we study the convergence rate of (17) with $r > 3$ and $f \in \mathcal{F}_\infty$. Compared with (3), this new ODE as a damping suffers from higher friction. Following the strategy adopted in the proof of Theorem 3, we consider a new energy functional defined as

$$\mathcal{E}(t) = \frac{2t^2}{r-1} (f(X(t)) - f^*) + (r-1) \left\| X(t) + \frac{t}{r-1} \dot{X}(t) - x^* \right\|^2.$$

3. In physics and engineering, damping may be modeled as a force proportional to velocity but opposite in direction, i.e. resisting motion; for instance, this force may be used as an approximation to the friction caused by drag. In our model, this force would be proportional to $-t\dot{X}$ where \dot{X} is velocity and $\frac{r}{t}$ is the damping coefficient.

By studying the derivative of this functional, we get the following result.

Theorem 5 *The solution X to (17) satisfies*

$$f(X(t)) - f^* \leq \frac{(r-1)^2 \|x_0 - x^*\|^2}{2t^2}, \quad \int_0^\infty t(f(X(t)) - f^*) dt \leq \frac{(r-1)^2 \|x_0 - x^*\|^2}{2(r-3)}.$$

Proof Noting $r\dot{X} + t\ddot{X} = -t\nabla f(X)$, we get $\dot{\mathcal{E}}$ equal to

$$\begin{aligned} \frac{4t}{r-1}(f(X) - f^*) + \frac{2t^2}{r-1}\langle \nabla f, \dot{X} \rangle + 2\langle X + \frac{t}{r-1}\dot{X} - x^*, r\dot{X} + t\ddot{X} \rangle \\ = \frac{4t}{r-1}(f(X) - f^*) - 2t\langle X - x^*, \nabla f(X) \rangle \leq -\frac{2(r-3)t}{r-1}(f(X) - f^*), \end{aligned} \quad (18)$$

where the inequality follows from the convexity of f . Since $f(X) \geq f^*$, the last display implies that \mathcal{E} is non-increasing. Hence

$$\frac{2t^2}{r-1}(f(X(t)) - f^*) \leq \mathcal{E}(t) \leq \mathcal{E}(0) = (r-1)\|x_0 - x^*\|^2,$$

yielding the first inequality of this theorem. To complete the proof, from (18) it follows that

$$\int_0^\infty \frac{2(r-3)t}{r-1}(f(X) - f^*) dt \leq -\int_0^\infty \frac{d\mathcal{E}}{dt} dt = \mathcal{E}(0) - \mathcal{E}(\infty) \leq (r-1)\|x_0 - x^*\|^2,$$

as desired for establishing the second inequality. \blacksquare

The first inequality is the same as (7) for the ODE (3), except for a larger constant $(r-1)^2/2$. The second inequality measures the error $f(X(t)) - f^*$ in an average sense, and cannot be deduced from the first inequality.

Now, it is tempting to obtain such analogs for the discrete Nesterov's scheme as well. Following the formulation of Beck and Teboulle (2009), we wish to minimize f in the composite form $f(x) = g(x) + h(x)$, where $g \in \mathcal{F}_L$ for some $L > 0$ and h is convex on \mathbb{R}^n possibly assuming extended value ∞ . Define the proximal subgradient

$$G_s(x) \triangleq \frac{x - \operatorname{argmin}_z (\|z - (x - s\nabla g(x))\|^2 / (2s) + h(z))}{s}.$$

Parametrizing by a constant r , we propose the generalized Nesterov's scheme,

$$\begin{aligned} x_k &= y_{k-1} - sG_s(y_{k-1}) \\ y_k &= x_k + \frac{k-1}{k+r-1}(x_k - x_{k-1}), \end{aligned} \quad (19)$$

starting from $y_0 = x_0$. The discrete analog of Theorem 5 is below.

Theorem 6 *The sequence $\{x_k\}$ given by (19) with $0 < s \leq 1/L$ satisfies*

$$f(x_k) - f^* \leq \frac{(r-1)^2 \|x_0 - x^*\|^2}{2s(k+r-2)^2}, \quad \sum_{k=1}^\infty (k+r-1)(f(x_k) - f^*) \leq \frac{(r-1)^2 \|x_0 - x^*\|^2}{2s(r-3)}.$$

The first inequality suggests that the generalized Nesterov's schemes still achieve $O(1/k^2)$ convergence rate. However, if the error bound satisfies $f(x_{k'}) - f^* \geq c/k'^2$ for some arbitrarily small $c > 0$ and a dense subsequence $\{k'\}$, i.e., $\{k'\} \cap \{1, \dots, m\} \geq \alpha m$ for all $m \geq 1$ and some $\alpha > 0$, then the second inequality of the theorem would be violated. To see this, note that if it were the case, we would have $(k' + r - 1)(f(x_{k'}) - f^*) \geq \frac{\alpha}{k'}$; the sum of the harmonic series $\frac{1}{k'}$ over a dense subset of $\{1, 2, \dots\}$ is infinite. Hence, the second inequality is not trivial because it implies the error bound is, in some sense, $O(1/k^2)$ suboptimal.

Now we turn to the proof of this theorem. It is worth pointing out that, though based on the same idea, the proof below is much more complicated than that of Theorem 5. **Proof** Consider the discrete energy functional,

$$\mathcal{E}(k) = \frac{2(k+r-2)^2 s}{r-1}(f(x_k) - f^*) + (r-1)\|z_k - x^*\|^2,$$

where $z_k = (k+r-1)y_k/(r-1) - ky_k/(r-1)$. If we have

$$\mathcal{E}(k) + \frac{2s[(r-3)(k+r-2)+1]}{r-1}(f(x_{k-1}) - f^*) \leq \mathcal{E}(k-1), \quad (20)$$

then it would immediately yield the desired results by summing (20) over k . That is, by recursively applying (20), we see

$$\begin{aligned} \mathcal{E}(k) + \sum_{i=1}^k \frac{2s[(r-3)(i+r-2)+1]}{r-1}(f(x_{i-1}) - f^*) \\ \leq \mathcal{E}(0) = \frac{2(r-2)^2 s}{r-1}(f(x_0) - f^*) + (r-1)\|x_0 - x^*\|^2, \end{aligned}$$

which is equivalent to

$$\mathcal{E}(k) + \sum_{i=1}^{k-1} \frac{2s[(r-3)(i+r-1)+1]}{r-1}(f(x_i) - f^*) \leq (r-1)\|x_0 - x^*\|^2. \quad (21)$$

Noting that the left-hand side of (21) is lower bounded by $2s(k+r-2)^2(f(x_k) - f^*)/(r-1)$, we thus obtain the first inequality of the theorem. Since $\mathcal{E}(k) \geq 0$, the second inequality is verified via taking the limit $k \rightarrow \infty$ in (21) and replacing $(r-3)(i+r-1) + 1$ by $(r-3)(i+r-1)$.

We now establish (20). For $s \leq 1/L$, we have the basic inequality,

$$f(y - sG_s(y)) \leq f(x) + G_s(y)^T(y - x) - \frac{s}{2}\|G_s(y)\|^2, \quad (22)$$

for any x and y . Note that $y_{k-1} - sG_s(y_{k-1})$ actually coincides with x_k . Summing of $(k-1)/(k+r-2) \times (22)$ with $x = x_{k-1}, y = y_{k-1}$ and $(r-1)/(k+r-2) \times (22)$ with $x = x^*, y = y_{k-1}$ gives

$$\begin{aligned} f(x_k) &\leq \frac{k-1}{k+r-2}f(x_{k-1}) + \frac{r-1}{k+r-2}f^* \\ &\quad + \frac{r-1}{k+r-2}G_s(y_{k-1})^T \left(\frac{k+r-2}{r-1}y_{k-1} - \frac{k-1}{r-1}x_{k-1} - x^* \right) - \frac{s}{2}\|G_s(y_{k-1})\|^2 \\ &= \frac{k-1}{k+r-2}f(x_{k-1}) + \frac{r-1}{k+r-2}f^* + \frac{(r-1)^2}{2s(k+r-2)^2} (\|z_{k-1} - x^*\|^2 - \|z_k - x^*\|^2), \end{aligned}$$

where we use $z_{k-1} - s(k+r-2)G_s(y_{k-1})/(r-1) = z_k$. Rearranging the above inequality and multiplying by $2s(k+r-2)^2/(r-1)$ gives the desired (20). \blacksquare

In closing, we would like to point out this new scheme is equivalent to setting $\theta_k = (r-1)/(k+r-1)$ and letting $\theta_k(\theta_{k-1}^{-1}-1)$ replace the momentum coefficient $(k-1)/(k+r-1)$. Then, the equal sign “=” in the update $\theta_{k+1} = (\sqrt{\theta_k^4 + 4\theta_k^2} - \theta_k^2)/2$ has to be replaced by an inequality sign “ \geq ”. In examining the proof of Theorem 1(b) in Tseng (2010), we can get an alternative proof of Theorem 6.

4.2 Low Friction

Now we turn to the case $r < 3$. Then, unfortunately, the energy functional approach for proving Theorem 5 is no longer valid, since the left-hand side of (18) is positive in general. In fact, there are counterexamples that fail the desired $O(1/t^2)$ or $O(1/k^2)$ convergence rate. We present such examples in continuous time. Equally, these examples would also violate the $O(1/k^2)$ convergence rate in the discrete schemes, and we forego the details.

Let $f(x) = \frac{1}{2}\|x\|^2$ and X be the solution to (17). Then, $Y = t^{\frac{r-1}{2}}X$ satisfies

$$t^2\dot{Y} + (t^2 - (r-1)^2/4)Y = 0.$$

With the initial condition $Y(t) \approx t^{\frac{r-1}{2}}x_0$ for small t , the solution to the above Bessel equation in a vector form of order $(r-1)/2$ is $Y(t) = 2^{\frac{r-1}{2}}\Gamma((r+1)/2)J_{(r-1)/2}(t)x_0$. Thus,

$$X(t) = \frac{2^{\frac{r-1}{2}}\Gamma((r+1)/2)J_{(r-1)/2}(t)}{t^{\frac{r-1}{2}}}x_0.$$

For large t , the Bessel function $J_{(r-1)/2}(t) = \sqrt{2/(\pi t)}(\cos t - (r-1)\pi/4 - \pi/4) + O(1/t)$. Hence,

$$f(X(t)) - f^* = O(\|x_0 - x^*\|^2/t^r),$$

where the exponent r is tight. This rules out the possibility of inverse quadratic convergence of the generalized ODE and scheme for all $f \in \mathcal{F}_L$ if $r < 2$. An example with $r = 1$ is plotted in Figure 2.

Next, we consider the case $2 \leq r < 3$ and let $f(x) = |x|$ (this also applies to multivariate $f = \|x\|$).⁴ Starting from $x_0 > 0$, we get $X(t) = x_0 - \frac{t^2}{2(1+r)}$ for $t \leq \sqrt{2(1+r)x_0}$. Requiring continuity of X and \dot{X} at the change point 0, we get

$$X(t) = \frac{t^2}{2(1+r)} + \frac{2(2(1+r)x_0)^{\frac{r-1}{2}}}{(r^2-1)t^{r-1}} - \frac{r+3}{r-1}x_0$$

for $\sqrt{2(1+r)x_0} < t \leq \sqrt{2c^*(1+r)x_0}$, where c^* is the positive root other than 1 of $(r-1)c + 4c^{\frac{r-1}{2}} = r+3$. Repeating this process solves for X . Note that t^{1-r} is in the null

4. This function does not have a Lipschitz continuous gradient. However, a similar pattern as in Figure 2 can be also observed if we smooth $|x|$ at an arbitrarily small vicinity of 0.

space of $\dot{X} + r\dot{X}/t$ and satisfies $t^2 \times t^{1-r} \rightarrow \infty$ as $t \rightarrow \infty$. For illustration, Figure 4 plots $t^2(f(X(t)) - f^*)$ and $sk^2(f(x_k) - f^*)$ with $r = 2, 2.5$, and $r = 4$ for comparison⁵. It is clearly that inverse quadratic convergence does not hold for $r = 2, 2.5$, that is, (2) does not hold for $r < 3$. Interestingly, in Figures 4a and 4d, the scaled errors at peaks grow linearly, whereas for $r = 2.5$, the growth rate, though positive as well, seems sublinear.

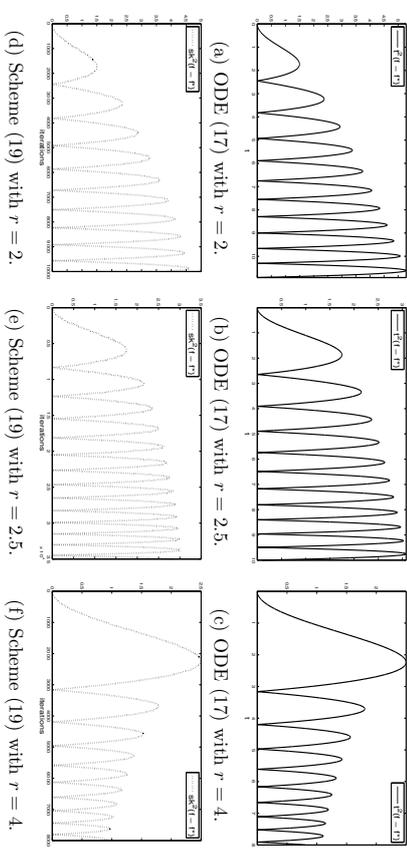


Figure 4: Scaled errors $t^2(f(X(t)) - f^*)$ and $sk^2(f(x_k) - f^*)$ of generalized ODEs and schemes for minimizing $f = |x|$. In (d), the step size $s = 10^{-6}$, in (e), $s = 10^{-7}$, and in (f), $s = 10^{-6}$.

However, if f possesses some additional property, inverse quadratic convergence is still guaranteed, as stated below. In that theorem, f is assumed to be a continuously differentiable convex function.

Theorem 7 Suppose $1 < r < 3$ and let X be a solution to the ODE (17). If $(f - f^*)^{\frac{r-1}{2}}$ is also convex, then

$$f(X(t)) - f^* \leq \frac{(r-1)^2\|x_0 - x^*\|^2}{2t^2}.$$

Proof Since $(f - f^*)^{\frac{r-1}{2}}$ is convex, we obtain

$$(f(X(t)) - f^*)^{\frac{r-1}{2}} \leq \langle X - x^*, \nabla(f(X) - f^*)^{\frac{r-1}{2}} \rangle = \frac{r-1}{2}(f(X) - f^*)^{\frac{r-3}{2}} \langle X - x^*, \nabla f(X) \rangle,$$

which can be simplified to $\frac{r-2}{r-1}(f(X) - f^*) \leq \langle X - x^*, \nabla f(X) \rangle$. This inequality combined with (18) leads to the monotonically decreasing of $\mathcal{E}(t)$ defined for Theorem 5. This completes the proof by noting $f(X) - f^* \leq (r-1)\mathcal{E}(t)/(2t^2) \leq (r-1)\mathcal{E}(0)/(2t^2) = (r-1)^2\|x_0 - x^*\|^2/(2t^2)$. \blacksquare

5. For Figures 4d, 4e and 4f, if running generalized Nesterov's schemes with too many iterations (e.g. 10^5), the deviations from the ODE will grow. Taking a sufficiently small s can solve this issue.

4.3 Strongly Convex f

Strong convexity is a desirable property for optimization. Making use of this property carefully suggests a generalized Nesterov's scheme that achieves optimal linear convergence (Nesterov, 2004). In that case, even vanilla gradient descent has a linear convergence rate. Unfortunately, the example given in the previous subsection simply rules out such possibility for (1) and its generalizations (19). However, from a different perspective, this example suggests that $O(t^{-r})$ convergence rate can be expected for (17). In the next theorem, we prove a slightly weaker statement of this kind, that is, a provable $O(t^{-\frac{2r}{3}})$ convergence rate is established for strongly convex functions. Bridging this gap may require new tools and more careful analysis.

Let $f \in \mathcal{S}_{\mu,L}(\mathbb{R}^n)$ and consider a new energy functional for $\alpha > 2$ defined as

$$\mathcal{E}(t; \alpha) = t^\alpha (f(X(t)) - f^*) + \frac{(2r - \alpha)^2 t^{\alpha-2}}{8} \|X(t) + \frac{2t}{2r - \alpha} \dot{X} - x^*\|^2.$$

When clear from the context, $\mathcal{E}(t; \alpha)$ is simply denoted as $\mathcal{E}(t)$. For $r > 3$, taking $\alpha = 2r/3$ in the theorem stated below gives $f(X(t)) - f^* \lesssim \|x_0 - x^*\|^2/t^{\frac{2r}{3}}$.

Theorem 8 For any $f \in \mathcal{S}_{\mu,L}(\mathbb{R}^n)$, if $2 \leq \alpha \leq 2r/3$ we get

$$f(X(t)) - f^* \leq \frac{C \|x_0 - x^*\|^2}{\mu^{\frac{\alpha-2}{2}} t^\alpha}$$

for any $t > 0$. Above, the constant C only depends on α and r .

Proof Note that $\dot{\mathcal{E}}(t; \alpha)$ equals

$$\begin{aligned} \alpha t^{\alpha-1} (f(X) - f^*) - \frac{(2r - \alpha)t^{\alpha-1}}{2} \langle X - x^*, \nabla f(X) \rangle + \frac{(\alpha - 2)(2r - \alpha)^2 t^{\alpha-3}}{8} \|X - x^*\|^2 \\ + \frac{(\alpha - 2)(2r - \alpha)t^{\alpha-2}}{4} \langle \dot{X}, X - x^* \rangle. \end{aligned} \quad (23)$$

By the strong convexity of f , the second term of the right-hand side of (23) is bounded below as

$$\frac{(2r - \alpha)t^{\alpha-1}}{2} \langle X - x^*, \nabla f(X) \rangle \geq \frac{(2r - \alpha)t^{\alpha-1}}{2} (f(X) - f^*) + \frac{\mu(2r - \alpha)t^{\alpha-1}}{4} \|X - x^*\|^2.$$

Substituting the last display into (23) with the awareness of $r \geq 3\alpha/2$ yields

$$\dot{\mathcal{E}} \leq -\frac{(2\mu(2r - \alpha)t^2 - (\alpha - 2)(2r - \alpha)^2)t^{\alpha-3}}{8} \|X - x^*\|^2 + \frac{(\alpha - 2)(2r - \alpha)t^{\alpha-2}}{8} d \|X - x^*\|^2.$$

Hence, if $t \geq t_\alpha := \sqrt{(\alpha - 2)(2r - \alpha)/(2\mu)}$, we obtain

$$\dot{\mathcal{E}}(t) \leq \frac{(\alpha - 2)(2r - \alpha)t^{\alpha-2}}{8} d \|X - x^*\|^2.$$

Integrating the last inequality on the interval (t_α, t) gives

$$\begin{aligned} \mathcal{E}(t) &\leq \mathcal{E}(t_\alpha) + \frac{(\alpha - 2)(2r - \alpha)t^{\alpha-2}}{8} \|X(t) - x^*\|^2 - \frac{(\alpha - 2)(2r - \alpha)t_\alpha^{\alpha-2}}{8} \|X(t_\alpha) - x^*\|^2 \\ &\quad - \frac{1}{8} \int_{t_\alpha}^t (\alpha - 2)^2 (2r - \alpha) t^{\alpha-3} \|X(u) - x^*\|^2 du \leq \mathcal{E}(t_\alpha) + \frac{(\alpha - 2)(2r - \alpha)t^{\alpha-2}}{8} \|X(t) - x^*\|^2 \\ &\leq \mathcal{E}(t_\alpha) + \frac{(\alpha - 2)(2r - \alpha)t^{\alpha-2}}{4\mu} (f(X(t)) - f^*). \end{aligned} \quad (24)$$

Making use of (24), we apply induction on α to finish the proof. First, consider $2 < \alpha \leq 4$. Applying Theorem 5, from (24) we get that $\mathcal{E}(t)$ is upper bounded by

$$\mathcal{E}(t_\alpha) + \frac{(\alpha - 2)(r - 1)^2 (2r - \alpha) \|x_0 - x^*\|^2}{8\mu t_\alpha^{4-\alpha}} \leq \mathcal{E}(t_\alpha) + \frac{(\alpha - 2)(r - 1)^2 (2r - \alpha) \|x_0 - x^*\|^2}{8\mu t_\alpha^{4-\alpha}}. \quad (25)$$

Then, we bound $\mathcal{E}(t_\alpha)$ as follows.

$$\begin{aligned} \mathcal{E}(t_\alpha) &\leq t_\alpha^\alpha (f(X(t_\alpha)) - f^*) + \frac{(2r - \alpha)^2 t_\alpha^{\alpha-2}}{4} \| \frac{2r-2}{2r-\alpha} X(t_\alpha) + \frac{2t_\alpha}{2r-\alpha} \dot{X}(t_\alpha) - \frac{2r-2}{2r-\alpha} x^* \|^2 \\ &\quad + \frac{(2r - \alpha)^2 t_\alpha^{\alpha-2}}{4} \| \frac{\alpha-2}{2r-\alpha} X(t_\alpha) - \frac{\alpha-2}{2r-\alpha} x^* \|^2 \\ &\leq (r - 1)^2 t_\alpha^{\alpha-2} \|x_0 - x^*\|^2 + \frac{(\alpha - 2)^2 (r - 1)^2 \|x_0 - x^*\|^2}{4\mu t_\alpha^{4-\alpha}}, \end{aligned} \quad (26)$$

where in the second inequality we use the decreasing property of the energy functional defined for Theorem 5. Combining (25) and (26), we have

$$\mathcal{E}(t) \leq (r - 1)^2 t_\alpha^{\alpha-2} \|x_0 - x^*\|^2 + \frac{(\alpha - 2)(r - 1)^2 (2r + \alpha - 4) \|x_0 - x^*\|^2}{8\mu t_\alpha^{4-\alpha}} = O\left(\frac{\|x_0 - x^*\|^2}{\mu^{\frac{\alpha-2}{2}} t^\alpha}\right).$$

For $t \geq t_\alpha$, it suffices to apply $f(X(t)) - f^* \leq \mathcal{E}(t)/t^\alpha$ to the last display. For $t < t_\alpha$, by Theorem 5, $f(X(t)) - f^*$ is upper bounded by

$$\begin{aligned} \frac{(r - 1)^2 \|x_0 - x^*\|^2}{2t^2} &\leq \frac{(r - 1)^2 t^{\frac{\alpha-2}{2}} [(\alpha - 2)(2r - \alpha)/(2\mu)]^{\frac{\alpha-2}{2}} \|x_0 - x^*\|^2}{2} \\ &\quad = O\left(\frac{\|x_0 - x^*\|^2}{\mu^{\frac{\alpha-2}{2}} t^\alpha}\right). \end{aligned} \quad (27)$$

Next, suppose that the theorem is valid for some $\tilde{\alpha} > 2$. We show below that this theorem is still valid for $\alpha := \tilde{\alpha} + 1$ if still $r \geq 3\alpha/2$. By the assumption, (24) further induces

$$\mathcal{E}(t) \leq \mathcal{E}(t_\alpha) + \frac{(\alpha - 2)(2r - \alpha)t^{\alpha-2} \tilde{C} \|x_0 - x^*\|^2}{4\mu} \leq \mathcal{E}(t_\alpha) + \frac{\tilde{C}(\alpha - 2)(2r - \alpha) \|x_0 - x^*\|^2}{4\mu^{\frac{\alpha-1}{2}} t_\alpha}$$

for some constant \tilde{C} only depending on $\bar{\alpha}$ and r . This inequality with (26) implies

$$\begin{aligned} \mathcal{E}(t) &\leq (r-1)^2 t_\alpha^{\alpha-2} \|x_0 - x^*\|^2 + \frac{(\alpha-2)^2 (r-1)^2 \|x_0 - x^*\|^2}{4\mu t_\alpha^4 - \alpha} \tilde{C}(\alpha-2)(2r-\alpha) \|x_0 - x^*\|^2 \\ &\quad + O\left(\|x_0 - x^*\|^2 / \mu^{\frac{\alpha-2}{2}}\right), \end{aligned}$$

which verify the induction for $t \geq t_\alpha$. As for $t < t_\alpha$, the validity of the induction follows from Theorem 5, similarly to (27). Thus, combining the base and induction steps, the proof is completed. \blacksquare

It should be pointed out that the constant C in the statement of Theorem 8 grows with the parameter r . Hence, simply increasing r does not guarantee to give a better error bound. While it is desirable to expect a discrete analogy of Theorem 8, i.e., $O(1/k^s)$ convergence rate for (19), a complete proof can be notoriously complicated. That said, we mimic the proof of Theorem 8 for $\alpha = 3$ and succeed in obtaining a $O(1/k^3)$ convergence rate for the generalized Nesterov's schemes, as summarized in the theorem below.

Theorem 9 *Suppose f is written as $f = g+h$, where $g \in S_{\mu,L}$ and h is convex with possible extended value ∞ . Then, the generalized Nesterov's scheme (19) with $r \geq 9/2$ and $s = 1/L$ satisfies*

$$f(x_k) - f^* \leq \frac{CL\|x_0 - x^*\|^2 \sqrt{L/\mu}}{k^2},$$

where C only depends on r .

This theorem states that the discrete scheme (19) enjoys the error bound $O(1/k^3)$ without any knowledge of the condition number L/μ . In particular, this bound is much better than that given in Theorem 6 if $k \gg \sqrt{L/\mu}$. The strategy of the proof is fully inspired by that of Theorem 8, though it is much more complicated and thus deferred to the Appendix. The relevant energy functional $\mathcal{E}(k)$ for this Theorem 9 is equal to

$$\frac{s(2k+3r-5)(2k+2r-5)(4k+4r-9)}{16} (f(x_k) - f^*) + \frac{2k+3r-5}{16} \|2(k+r-1)yk - (2k+1)x_k - (2r-3)x^*\|^2. \quad (28)$$

4.4 Numerical Examples

We study six synthetic examples to compare (19) with the step sizes are fixed to be $1/L$, as illustrated in Figure 5. The error rates exhibit similar patterns for all r , namely, decreasing while suffering from local bumps. A smaller r introduces less friction, thus allowing x_k moves towards x^* faster in the beginning. However, when sufficiently close to x^* , more friction is preferred in order to reduce overshoot. This point of view explains what we observe in these examples. That is, across these six examples, (19) with a smaller r performs slightly better in the beginning, but a larger r has advantage when k is large. It is an interesting question how to choose a good r for different problems in practice.

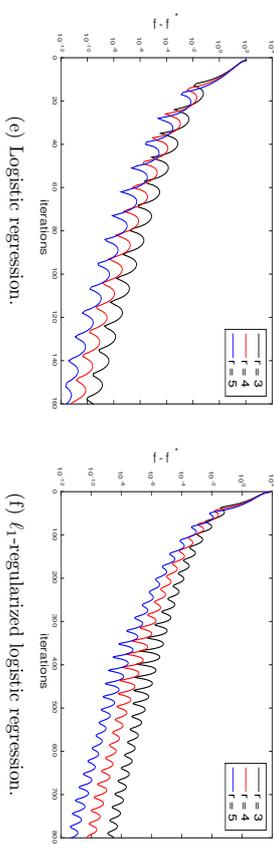
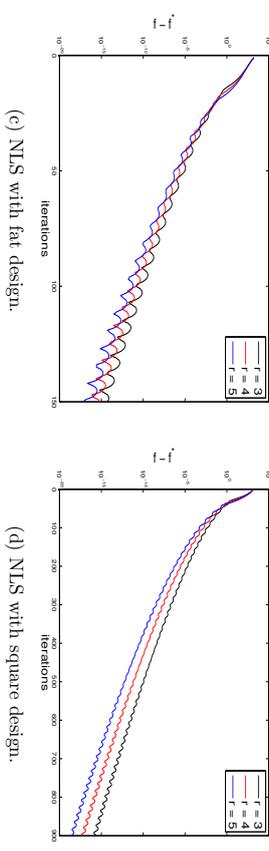
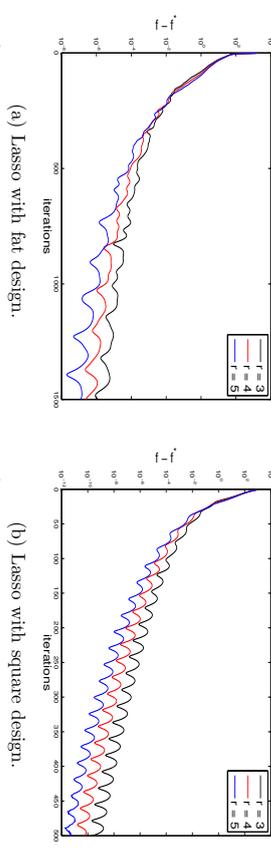


Figure 5: Comparisons of generalized Nesterov's schemes with different r .

Lasso with fat design. Minimize $f(x) = \frac{1}{2}\|Ax - b\|^2 + \lambda\|x\|_1$, in which A a 100×500 random matrix with i.i.d. standard Gaussian $\mathcal{N}(0, 1)$ entries, b generated independently has i.i.d. $\mathcal{N}(0, 25)$ entries, and the penalty $\lambda = 4$. The plot is Figure 5a.

Lasso with square design. Minimize $f(x) = \frac{1}{2}\|Ax - b\|^2 + \lambda\|x\|$, where A a 500×500 random matrix with i.i.d. standard Gaussian entries, b generated independently has i.i.d. $\mathcal{N}(0, 9)$ entries, and the penalty $\lambda = 4$. The plot is Figure 5b.

Nonnegative least squares (NLS) with fat design. Minimize $f(x) = \|Ax - b\|^2$ subject to $x \succeq 0$, with the same design A and b as in Figure 5a. The plot is Figure 5c.

Nonnegative least squares with sparse design. Minimize $f(x) = \|Ax - b\|^2$ subject to $x \geq 0$, in which A is a 1000×10000 sparse matrix with nonzero probability 10% for each entry and b is given as $b = Ax^0 + \mathcal{N}(0, I_{1000})$. The nonzero entries of A are independently Gaussian distributed before column normalization, and x^0 has 100 nonzero entries that are all equal to 4. The plot is Figure 5d.

Logistic regression. Minimize $\sum_{i=1}^n -y_i a_i^T x + \log(1 + e^{a_i^T x})$, in which $A = (a_1, \dots, a_n)^T$ is a 500×100 matrix with i.i.d. $\mathcal{N}(0, 1)$ entries. The labels $y_i \in \{0, 1\}$ are generated by the logistic model: $\mathbb{P}(Y_i = 1) = 1/(1 + e^{-a_i^T x^0})$, where x^0 is a realization of i.i.d. $\mathcal{N}(0, 1/100)$. The plot is Figure 5c.

ℓ_1 -regularized logistic regression. Minimize $\sum_{i=1}^n -y_i a_i^T x + \log(1 + e^{a_i^T x}) + \lambda \|x\|_1$, in which $A = (a_1, \dots, a_n)^T$ is a 200×1000 matrix with i.i.d. $\mathcal{N}(0, 1)$ entries and $\lambda = 5$. The labels y_i are generated similarly as in the previous example, except for the ground truth x^0 here having 10 nonzero components given as i.i.d. $\mathcal{N}(0, 225)$. The plot is Figure 5f.

5. Restarting

The example discussed in Section 4.2 demonstrates that Nesterov's scheme and its generalizations (19) are not capable of fully exploiting strong convexity. That is, this example suggests evidence that $O(1/\text{poly}(k))$ is the best rate achievable under strong convexity. In contrast, the vanilla gradient method achieves linear convergence $O((1 - \mu/L)^k)$. This drawback results from too much momentum introduced when the objective function is strongly convex. The derivative of a strongly convex function is generally more reliable than that of non-strongly convex functions. In the language of ODEs, at later stage a too small $3/t$ in (3) leads to a lack of friction, resulting in unnecessary overshoot along the trajectory. Incorporating the optimal momentum coefficient $\frac{\sqrt{L - \sqrt{\mu}}}{\sqrt{T + \sqrt{\mu}}}$ (This is less than $(k - 1)/(k + 2)$ when k is large), Nesterov's scheme has convergence rate of $O((1 - \sqrt{\mu/L})^k)$ (Nesterov, 2004), which, however, requires knowledge of the condition number μ/L . While it is relatively easy to bound the Lipschitz constant L by the use of backtracking, estimating the strong convexity parameter μ , if not impossible, is very challenging.

Among many approaches to gain acceleration via adaptively estimating μ/L (see Nesterov, 2013; O'Donoghue and Candès (2013) proposes a procedure termed as gradient restarting for Nesterov's scheme in which (1) is restarted with $x_0 = y_0 := x_k$ whenever $f(x_{k+1}) > f(x_k)$). In the language of ODEs, this restarting essentially keeps $\langle \nabla f, \dot{X} \rangle$ negative, and resets $3/t$ each time to prevent this coefficient from steadily decreasing along the trajectory. Although it has been empirically observed that this method significantly boosts convergence, there is no general theory characterizing the convergence rate.

In this section, we propose a new restarting scheme we call the speed restarting scheme. The underlying motivation is to maintain a relatively high velocity \dot{X} along the trajectory, similar in spirit to the gradient restarting. Specifically, our main result, Theorem 10, ensures linear convergence of the continuous version of the speed restarting. More generally, our contribution here is merely to provide a framework for analyzing restarting schemes rather than competing with other schemes; it is beyond the scope of this paper to get optimal constants in these results. Throughout this section, we assume $f \in \mathcal{S}_{\mu,L}$ for some $0 < \mu \leq L$. Recall that function $f \in \mathcal{S}_{\mu,L}$ if $f \in \mathcal{F}_L$ and $f(x) - \mu \|x\|^2/2$ is convex.

5.1 A New Restarting Scheme

We first define the speed restarting time. For the ODE (3), we call

$$T = T(x_0; f) = \sup \left\{ t > 0 : \forall u \in (0, t), \frac{d\|\dot{X}(u)\|^2}{du} > 0 \right\}$$

the speed restarting time. In words, T is the first time the velocity $\|\dot{X}\|$ decreases. Back to the discrete scheme, it is the first time when we observe $\|x_{k+1} - x_k\| < \|x_k - x_{k-1}\|$. This definition itself does not directly imply that $0 < T < \infty$, which is proven later in Lemmas 13 and 25. Indeed, $f(X(t))$ is a decreasing function before time T ; for $t \leq T$,

$$\frac{df(X(t))}{dt} = \langle \nabla f(X), \dot{X} \rangle = -\frac{3}{t} \|\dot{X}\|^2 - \frac{1}{2} \frac{d\|\dot{X}\|^2}{dt} \leq 0.$$

The speed restarted ODE is thus

$$\ddot{X}(t) + \frac{3}{t_{\text{sr}}} \dot{X}(t) + \nabla f(X(t)) = 0, \quad (29)$$

where t_{sr} is set to zero whenever $\langle \dot{X}, \ddot{X} \rangle = 0$ and between two consecutive restarts, t_{sr} grows just as t . That is, $t_{\text{sr}} = t - \tau$, where τ is the latest restart time. In particular, $t_{\text{sr}} = 0$ at $t = 0$. Letting X^{sr} be the solution to (29), we have the following observations.

- $X^{\text{sr}}(t)$ is continuous for $t \geq 0$, with $X^{\text{sr}}(0) = x_0$;
- $X^{\text{sr}}(t)$ satisfies (3) for $0 < t < T_1 := T(x_0; f)$;
- Recursively define $T_{i+1} = T(X^{\text{sr}}(\sum_{j=1}^i T_j); f)$ for $i \geq 1$, and $\tilde{X}(t) := X^{\text{sr}}(\sum_{j=1}^i T_j + t)$ satisfies the ODE (3), with $\tilde{X}(0) = X^{\text{sr}}(\sum_{j=1}^i T_j)$, for $0 < t < T_{i+1}$.

The theorem below guarantees linear convergence of X^{sr} . This is a new result in the literature (O'Donoghue and Candès, 2013; Monteiro et al., 2012). The proof of Theorem 10 is based on Lemmas 12 and 13, where the first guarantees the rate $f(X^{\text{sr}}) - f^*$ decays by a constant factor for each restarting, and the second confirms that restartings are adequate. In these lemmas we all make a convention that the uninteresting case $x_0 = x^*$ is excluded.

Theorem 10 *There exist positive constants c_1 and c_2 , which only depend on the condition number L/μ , such that for any $f \in \mathcal{S}_{\mu,L}$, we have*

$$f(X^{\text{sr}}(t)) - f^* \leq \frac{c_1 L \|x_0 - x^*\|^2}{2} e^{-c_2 t \sqrt{L}}.$$

Before turning to the proof, we make a remark that this linear convergence of X^{sr} remains to hold for the generalized ODE (17) with $r > 3$. Only minor modifications in the proof below are needed, such as replacing u^3 by u^r in the definition of $I(t)$ in Lemma 25.

5.2 Proof of Linear Convergence

First, we collect some useful estimates. Denote by $M(t)$ the supremum of $\|\dot{X}(u)\|/u$ over $u \in (0, t]$ and let

$$I(t) := \int_0^t u^3 (\nabla f(X(u)) - \nabla f(x_0)) du.$$

It is guaranteed that M defined above is finite, for example, see the proof of Lemma 18. The definition of M gives a bound on the gradient of f ,

$$\|\nabla f(X(t)) - \nabla f(x_0)\| \leq L \left\| \int_0^t \dot{X}(u) du \right\| \leq L \int_0^t u \frac{\|\dot{X}(u)\|}{u} du \leq \frac{LM(t)t^2}{2}.$$

Hence, it is easy to see that I can also be bounded via M ,

$$\|I(t)\| \leq \int_0^t u^3 \|\nabla f(X(u)) - \nabla f(x_0)\| du \leq \int_0^t \frac{LM(u)u^5}{2} du \leq \frac{LM(t)t^6}{12}.$$

To fully facilitate these estimates, we need the following lemma that gives an upper bound of M , whose proof is deferred to the appendix.

Lemma 11 For $t < \sqrt{12/L}$, we have

$$M(t) \leq \frac{\|\nabla f(x_0)\|}{4(1 - Lt^2/12)}.$$

Next we give a lemma which claims that the objective function decays by a constant through each speed restarting.

Lemma 12 There is a universal constant $C > 0$ such that

$$f(X(T)) - f^* \leq \left(1 - \frac{C\mu}{L}\right) (f(x_0) - f^*).$$

Proof By Lemma 11, for $t < \sqrt{12/L}$ we have

$$\left\| \dot{X}(t) + \frac{t}{4} \nabla f(x_0) \right\| = \frac{1}{t^3} \|I(t)\| \leq \frac{LM(t)t^3}{12} \leq \frac{L\|\nabla f(x_0)\|t^3}{48(1 - Lt^2/12)},$$

which yields

$$0 \leq \frac{t}{4} \|\nabla f(x_0)\| - \frac{L\|\nabla f(x_0)\|t^3}{48(1 - Lt^2/12)} \leq \|\dot{X}(t)\| \leq \frac{t}{4} \|\nabla f(x_0)\| + \frac{L\|\nabla f(x_0)\|t^3}{48(1 - Lt^2/12)}. \quad (30)$$

Hence, for $0 < t < 4/(5\sqrt{L})$ we get

$$\begin{aligned} \frac{df(X)}{dt} &= -\frac{3}{t} \|\dot{X}\|^2 - \frac{1}{2} \frac{d}{dt} \|\dot{X}\|^2 \leq -\frac{3}{t} \|\dot{X}\|^2 \\ &\leq -\frac{3}{t} \left(\frac{t}{4} \|\nabla f(x_0)\| - \frac{L\|\nabla f(x_0)\|t^3}{48(1 - Lt^2/12)} \right)^2 \leq -C_1 t \|\nabla f(x_0)\|^2, \end{aligned}$$

where $C_1 > 0$ is an absolute constant and the second inequality follows from Lemma 25 in the appendix. Consequently,

$$f\left(X(4/(5\sqrt{L}))\right) - f(x_0) \leq \int_0^{4/(5\sqrt{L})} -C_1 u \|\nabla f(x_0)\|^2 du \leq -\frac{C\mu}{L} (f(x_0) - f^*),$$

where $C = 16C_1/25$ and in the last inequality we use the μ -strong convexity of f . Thus we have

$$f\left(X\left(\frac{4}{5\sqrt{L}}\right)\right) - f^* \leq \left(1 - \frac{C\mu}{L}\right) (f(x_0) - f^*).$$

To complete the proof, note that $f(X(T)) \leq f(X(4/(5\sqrt{L})))$ by Lemma 25. ■

With each restarting reducing the error $f - f^*$ by a constant a factor, we still need the following lemma to ensure sufficiently many restartings.

Lemma 13 There is a universal constant \tilde{C} such that

$$T \leq \frac{4 \exp(\tilde{C}L/\mu)}{5\sqrt{L}}.$$

Proof For $4/(5\sqrt{L}) \leq t \leq T$, we have $\frac{df(X)}{dt} \leq -\frac{3}{t} \|\dot{X}(t)\|^2 \leq -\frac{3}{t} \|\dot{X}(4/(5\sqrt{L}))\|^2$, which implies

$$f(X(T)) - f(x_0) \leq -\int_{4/(5\sqrt{L})}^T \frac{3}{t} \|\dot{X}(4/(5\sqrt{L}))\|^2 dt = -3 \|\dot{X}(4/(5\sqrt{L}))\|^2 \log \frac{5T\sqrt{L}}{4}.$$

Hence, we get an upper bound for T ,

$$T \leq \frac{4}{5\sqrt{L}} \exp\left(\frac{f(x_0) - f(X(T))}{3\|\dot{X}(4/(5\sqrt{L}))\|^2}\right) \leq \frac{4}{5\sqrt{L}} \exp\left(\frac{f(x_0) - f^*}{3\|\dot{X}(4/(5\sqrt{L}))\|^2}\right).$$

Plugging $t = 4/(5\sqrt{L})$ into (30) gives $\|\dot{X}(4/(5\sqrt{L}))\| \geq \frac{C\mu}{2L} \|\nabla f(x_0)\|$ for some universal constant $C_1 > 0$. Hence, from the last display we get

$$T \leq \frac{4}{5\sqrt{L}} \exp\left(\frac{L(f(x_0) - f^*)}{3C_1^2 \|\nabla f(x_0)\|^2}\right) \leq \frac{4}{5\sqrt{L}} \exp\frac{L}{6C_1^2 \mu}.$$

■

Now, we are ready to prove Theorem 10 by applying Lemmas 12 and 13.

Proof Note that Lemma 13 asserts, by time t at least $m := \lfloor 5t\sqrt{L}e^{-C_1L/\mu}/4 \rfloor$ restartings have occurred for X^{sr} . Hence, recursively applying Lemma 12, we have

$$\begin{aligned} f(X^{\text{sr}}(t)) - f^* &\leq f(X^{\text{sr}}(T_1 + \dots + T_m)) - f^* \\ &\leq (1 - C\mu/L) (f(X^{\text{sr}}(T_1 + \dots + T_{m-1})) - f^*) \\ &\leq \dots \leq \dots \\ &\leq (1 - C\mu/L)^m (f(x_0) - f^*) \leq e^{-C\mu m/L} (f(x_0) - f^*) \\ &\leq c_1 e^{-c_2 \sqrt{L}t} (f(x_0) - f^*) \leq \frac{c_1 L \|x_0 - x^*\|^2}{2} e^{-c_2 \sqrt{L}t}, \end{aligned}$$

where $c_1 = \exp(C\mu/L)$ and $c_2 = 5C\mu e^{-C\mu/L}/(4L)$. ■

In closing, we remark that we believe that estimate in Lemma 12 is tight, while not for Lemma 13. Thus we conjecture that for a large class of $f \in \mathcal{S}_{\mu,L}$, if not all, $T = O(\sqrt{L}/\mu)$. If this is true, the exponent constant c_2 in Theorem 10 can be significantly improved.

5.3 Numerical Examples

Below we present a discrete analog to the restarted scheme. There, k_{\min} is introduced to avoid having consecutive restarts that are too close. To compare the performance of the restarted scheme with the original (1), we conduct four simulation studies, including both smooth and non-smooth objective functions. Note that the computational costs of the restarted and non-restarted schemes are the same.

Algorithm 1 Speed Restarting Nesterov's Scheme

input: $x_0 \in \mathbb{R}^n, y_0 = x_0, x_{-1} = x_0, 0 < s \leq 1/L, k_{\max} \in \mathbb{N}^+$ and $k_{\min} \in \mathbb{N}^+$
 $j \leftarrow 1$
for $k = 1$ to k_{\max} **do**
 $x_k \leftarrow \operatorname{argmin}_x (\frac{1}{2s} \|x - y_{k-1} + s\nabla g(y_{k-1})\|^2 + h(x))$
 $y_k \leftarrow x_k + \frac{j-1}{j+2}(x_k - x_{k-1})$
if $\|x_k - x_{k-1}\| < \|x_{k-1} - x_{k-2}\|$ **and** $j \geq k_{\min}$ **then**
 $j \leftarrow 1$
else
 $j \leftarrow j + 1$
end if
end for

Quadratic. $f(x) = \frac{1}{2}x^T Ax + b^T x$ is a strongly convex function, in which A is a 500×500 random positive definite matrix and b a random vector. The eigenvalues of A are between 0.001 and 1. The vector b is generated as i.i.d. Gaussian random variables with mean 0 and variance 25.

Log-sum-exp.

$$f(x) = \rho \log \left[\sum_{i=1}^m \exp((a_i^T x - b_i)/\rho) \right],$$

where $n = 50, m = 200, \rho = 20$. The matrix $A = (a_{ij})$ is a random matrix with i.i.d. standard Gaussian entries, and $b = (b_i)$ has i.i.d. Gaussian entries with mean 0 and variance 2. This function is not strongly convex.

Matrix completion. $f(X) = \frac{1}{2} \|X_{\text{obs}} - M_{\text{obs}}\|_F^2 + \lambda \|X\|_*$, in which the ground truth M is a rank-5 random matrix of size 300×300 . The regularization parameter is set to $\lambda = 0.05$. The 5 singular values of M are $1, \dots, 5$. The observed set is independently sampled among the 300×300 entries so that 10% of the entries are actually observed.

Lasso in ℓ_1 -constrained form with large sparse design. $f(x) = \frac{1}{2} \|Ax - b\|^2$ s.t. $\|x\|_1 \leq \delta$, where A is a 5000×5000 random sparse matrix with nonzero probability 0.5% for each

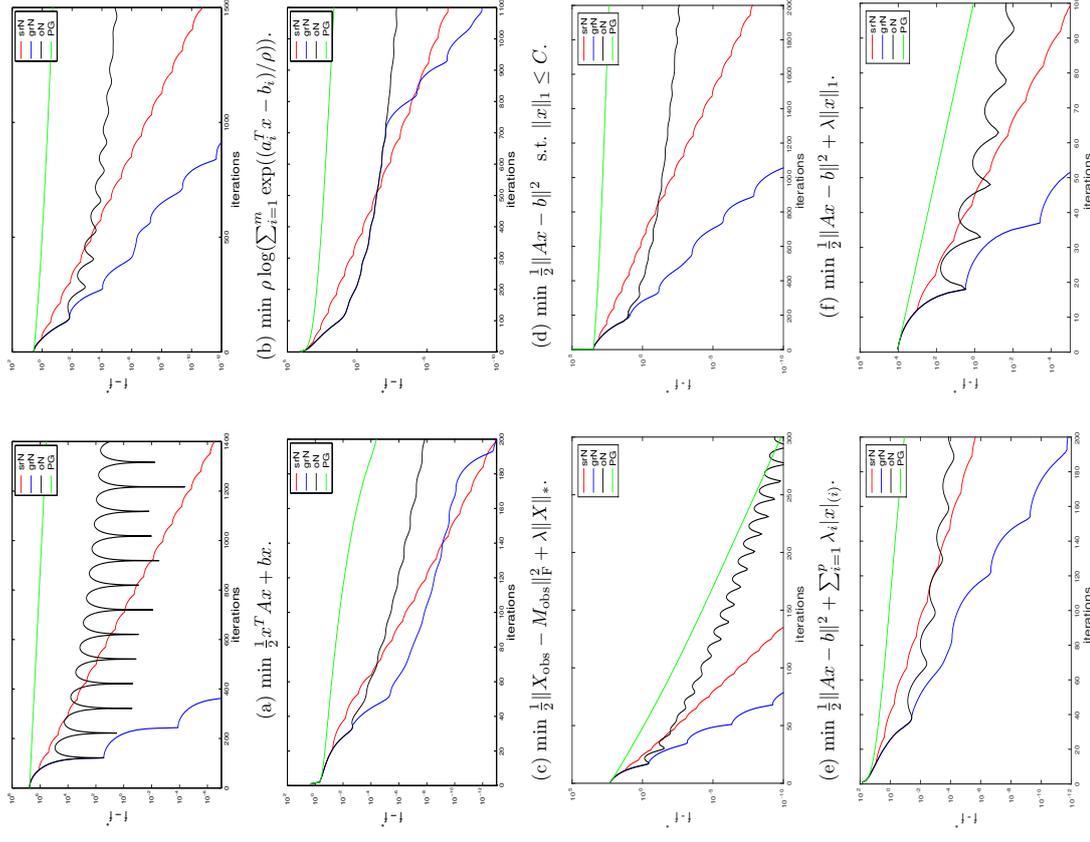


Figure 6: Numerical performance of speed restarting (srN), gradient restarting (grN), the original Nesterov's scheme (oN) and the proximal gradient (PG).

entry and b is generated as $b = Ax^0 + z$. The nonzero entries of A independently follow the Gaussian distribution with mean 0 and variance 0.04. The signal x^0 is a vector with 250 nonzeros and z is i.i.d. standard Gaussian noise. The parameter δ is set to $\|x^0\|_1$.

Sorted l_1 penalized estimation. $f(x) = \frac{1}{2}\|Ax - b\|^2 + \sum_{i=1}^p \lambda_i |x|_{(i)}$, where $|x|_{(i)} \geq \dots \geq |x|_{(p)}$ are the order statistics of $|x|$. This is a recently introduced testing and estimation procedure (Bogdan et al., 2015). The design A is a 1000×10000 Gaussian random matrix, and b is generated as $b = Ax^0 + z$ for 20-sparse x^0 and Gaussian noise z . The penalty sequence is set to $\lambda_i = 1.1\Phi^{-1}(1 - 0.05i/(2p))$.

Lasso. $f(x) = \frac{1}{2}\|Ax - b\|^2 + \lambda\|x\|_1$, where A is a 1000×500 random matrix and b is given as $b = Ax^0 + z$ for 20-sparse x^0 and Gaussian noise z . We set $\lambda = 1.5\sqrt{2\log p}$.

l_1 -regularized logistic regression. $f(x) = \sum_{i=1}^n -y_i a_i^T x + \log(1 + e^{a_i^T x}) + \lambda\|x\|_1$, where the setting is the same as in Figure 5f. The results are presented in Figure 6g.

Logistic regression with large sparse design. $f(x) = \sum_{i=1}^n -y_i a_i^T x + \log(1 + e^{a_i^T x})$, in which $A = (a_1, \dots, a_n)^T$ is a $10^7 \times 20000$ sparse random matrix with nonzero probability 0.1% for each entry, so there are roughly 2×10^8 nonzero entries in total. To generate the labels y_i , we set x^0 to be i.i.d. $\mathcal{N}(0, 1/4)$. The plot is Figure 6h.

In these examples, k_{\min} is set to be 10 and the step sizes are fixed to be $1/L$. If the objective is in composite form, the Lipschitz bound applies to the smooth part. Figure 6 presents the performance of the speed restarting scheme, the gradient restarting scheme, the original Nesterov's scheme and the proximal gradient method. The objective functions include strongly convex, non-strongly convex and non-smooth functions, violating the assumptions in Theorem 10. Among all the examples, it is interesting to note that both speed restarting scheme empirically exhibit linear convergence by significantly reducing bumps in the objective values. This leaves us an open problem of whether there exists provable linear convergence rate for the gradient restarting scheme as in Theorem 10. It is also worth pointing out that compared with gradient restarting, the speed restarting scheme empirically exhibits more stable linear convergence rate.

6. Discussion

This paper introduces a second-order ODE and accompanying tools for characterizing Nesterov's accelerated gradient method. This ODE is applied to study variants of Nesterov's scheme and is capable of interpreting some empirically observed phenomena, such as oscillations along the trajectories. Our approach suggests (1) a large family of generalized Nesterov's schemes that are all guaranteed to converge at the rate $O(1/k^2)$, and (2) a restarting scheme provably achieving a linear convergence rate whenever f is strongly convex.

In this paper, we often utilize ideas from continuous-time ODEs, and then apply these ideas to discrete schemes. The translation, however, involves parameter tuning and tedious calculations. This is the reason why a general theory mapping properties of ODEs into corresponding properties for discrete updates would be a welcome advance. Indeed, this would allow researchers to only study the simpler and more user-friendly ODEs.

As evidenced by many examples, the viewpoint of regarding the ODE as a surrogate for Nesterov's scheme would allow a new perspective for studying accelerated methods in optimization. The discrete scheme and the ODE are closely connected by the exact

mapping between the coefficients of momentum (e.g. $(k-1)/(k+2)$) and velocity (e.g. $3/t$). The derivations of generalized Nesterov's schemes and the speed restarting scheme are both motivated by trying a different velocity coefficient, in which the surprising phase transition at 3 is observed. Clearly, such alternatives are endless, and we expect this will lead to findings of many discrete accelerated schemes. In a different direction, a better understanding of the trajectory of the ODEs, such as curvature, has the potential to be helpful in deriving appropriate stopping criteria for termination, and choosing step size by backtracking.

Acknowledgments

W. S. was partially supported by a General Wang Yaowu Stanford Graduate Fellowship. S. B. was partially supported by DARPA XDATA. E. C. was partially supported by AFOSR under grant FA9550-09-1-0643, by NSF under grant CCF-0963835, and by the Math + X Award from the Simons Foundation. We would like to thank Carlos Sing-Long, Zhon Fan, and Xi Chen for helpful discussions about parts of this paper. We would also like to thank the associate editor and two reviewers for many constructive comments that improved the presentation of the paper.

Appendix A. Proof of Theorem 1

The proof is divided into two parts, namely, existence and uniqueness.

Lemma 14 For any $f \in \mathcal{F}_\infty$ and any $x_0 \in \mathbb{R}^n$, the ODE (3) has at least one solution X in $C^2(0, \infty) \cap C^1[0, \infty)$.

Below, some preparatory lemmas are given before turning to the proof of this lemma. To begin with, for any $\delta > 0$ consider the smoothed ODE

$$\ddot{X} + \frac{3}{\max(\delta, t)} \dot{X} + \nabla f(X) = 0 \quad (31)$$

with $X(0) = x_0, \dot{X}(0) = 0$. Denoting by $Z = \dot{X}$, then (31) is equivalent to

$$\frac{d}{dt} \begin{pmatrix} X \\ Z \end{pmatrix} = \begin{pmatrix} Z \\ -\frac{3}{\max(\delta, t)} Z - \nabla f(X) \end{pmatrix}$$

with $X(0) = x_0, Z(0) = 0$. As functions of (X, Z) , both Z and $-3Z/\max(\delta, t) - \nabla f(X)$ are $\max(1, L) + 3/\delta$ -Lipschitz continuous. Hence by standard ODE theory, (31) has a unique global solution in $C^2(0, \infty)$ denoted by X_δ . Note that X_δ is also well defined at $t = 0$.

Next, introduce $M_\delta(t)$ to be the supremum of $\|\dot{X}_\delta(u)\|/u$ over $u \in (0, t]$. It is easy to see that $M_\delta(t)$ is finite because $\|\dot{X}_\delta(u)\|/u = (\|\dot{X}_\delta(u) - \dot{X}_\delta(0)\|)/u = \|\dot{X}_\delta(0)\| + o(1)$ for small u . We give an upper bound for $M_\delta(t)$ in the following lemma.

Lemma 15 For $\delta < \sqrt{6/L}$, we have

$$M_\delta(t) \leq \frac{\|\nabla f(x_0)\|}{1 - L\delta^2/6}.$$

The proof of Lemma 15 relies on a simple lemma.

Lemma 16 For any $u > 0$, the following inequality holds

$$\|\nabla f(\dot{X}_\delta(u)) - \nabla f(x_0)\| \leq \frac{1}{2} LM_\delta(u) u^2.$$

Proof By Lipschitz continuity,

$$\|\nabla f(\dot{X}_\delta(u)) - \nabla f(x_0)\| \leq L \|\dot{X}_\delta(u) - x_0\| = \left\| \int_0^u \dot{X}_\delta(v) dv \right\| \leq \int_0^u \frac{\|\dot{X}_\delta(v)\|}{v} dv \leq \frac{1}{2} LM_\delta(u) u^2. \quad \blacksquare$$

Next, we prove Lemma 15.

Proof For $0 < t \leq \delta$, the smoothed ODE takes the form

$$\ddot{X}_\delta + \frac{3}{\delta} \dot{X}_\delta + \nabla f(\dot{X}_\delta) = 0,$$

which yields

$$\dot{X}_\delta e^{3t/\delta} = - \int_0^t \nabla f(\dot{X}_\delta(u)) e^{3u/\delta} du = - \nabla f(x_0) \int_0^t e^{3u/\delta} du - \int_0^t (\nabla f(\dot{X}_\delta(u)) - \nabla f(x_0)) e^{3u/\delta} du.$$

Hence, by Lemma 16

$$\begin{aligned} \frac{\|\dot{X}_\delta(t)\|}{t} &\leq \frac{1}{t} e^{-3t/\delta} \|\nabla f(x_0)\| \int_0^t e^{3u/\delta} du + \frac{1}{t} e^{-3t/\delta} \int_0^t \frac{1}{2} LM_\delta(u) u^2 e^{3u/\delta} du \\ &\leq \|\nabla f(x_0)\| + \frac{LM_\delta(\delta) \delta^2}{6}. \end{aligned}$$

Taking the supremum of $\|\dot{X}_\delta(t)\|/t$ over $0 < t \leq \delta$ and rearranging the inequality give the desired result. \blacksquare

Next, we give an upper bound for $M_\delta(t)$ when $t > \delta$.

Lemma 17 For $\delta < \sqrt{6/L}$ and $\delta < t < \sqrt{12/L}$, we have

$$M_\delta(t) \leq \frac{(5 - L\delta^2/6) \|\nabla f(x_0)\|}{4(1 - L\delta^2/6)(1 - Lt^2/12)}.$$

Proof For $t > \delta$, the smoothed ODE takes the form

$$\ddot{X}_\delta + \frac{3}{t} \dot{X}_\delta + \nabla f(\dot{X}_\delta) = 0,$$

which is equivalent to

$$\frac{dt^3 \dot{X}_\delta(t)}{dt} = -t^3 \nabla f(\dot{X}_\delta(t)).$$

Hence, by integration, $t^3 \dot{X}_\delta(t)$ is equal to

$$- \int_\delta^t u^3 \nabla f(\dot{X}_\delta(u)) du + \delta^3 \dot{X}_\delta(\delta) = - \int_\delta^t u^3 \nabla f(x_0) du - \int_\delta^t u^3 (\nabla f(\dot{X}_\delta(u)) - \nabla f(x_0)) du + \delta^3 \dot{X}_\delta(\delta).$$

Therefore by Lemmas 16 and 15, we get

$$\begin{aligned} \frac{\|\dot{X}_\delta(t)\|}{t} &\leq \frac{t^4 - \delta^4}{4t^4} \|\nabla f(x_0)\| + \frac{1}{t^4} \int_\delta^t \frac{1}{2} LM_\delta(u) u^5 du + \frac{\delta^4 \|\dot{X}_\delta(\delta)\|}{t^4} \\ &\leq \frac{1}{4} \|\nabla f(x_0)\| + \frac{1}{12} LM_\delta(t) t^2 + \frac{\|\nabla f(\dot{X}_0)\|}{1 - L\delta^2/6}, \end{aligned}$$

where the last expression is an increasing function of t . So for any $\delta < t' < t$, it follows that

$$\frac{\|\dot{X}_\delta(t')\|}{t'} \leq \frac{1}{4} \|\nabla f(x_0)\| + \frac{1}{12} LM_\delta(t) t^2 + \frac{\|\nabla f(x_0)\|}{1 - L\delta^2/6},$$

which also holds for $t' \leq \delta$. Taking the supremum over $t' \in (0, t)$ gives

$$M_\delta(t) \leq \frac{1}{4} \|\nabla f(x_0)\| + \frac{1}{12} LM_\delta(t) t^2 + \frac{\|\nabla f(\dot{X}_0)\|}{1 - L\delta^2/6}.$$

The desired result follows from rearranging the inequality. \blacksquare

Lemma 18 The function class $\mathcal{F} = \{X_\delta : [0, \sqrt{6/L}] \rightarrow \mathbb{R}^n \mid \delta = \sqrt{3/L}/2^m, m = 0, 1, \dots\}$ is uniformly bounded and equicontinuous.

Proof By Lemmas 15 and 17, for any $t \in [0, \sqrt{6/L}]$, $\delta \in (0, \sqrt{3/L})$ the gradient is uniformly bounded as

$$\|\dot{X}_\delta(t)\| \leq \sqrt{6/L} M_\delta(\sqrt{6/L}) \leq \sqrt{6/L} \max \left\{ \frac{\|\nabla f(x_0)\|}{1 - \frac{1}{2}}, \frac{5 \|\nabla f(x_0)\|}{4(1 - \frac{1}{2})(1 - \frac{1}{2})} \right\} = 5\sqrt{6/L} \|\nabla f(x_0)\|.$$

Thus it immediately implies that \mathcal{F} is equicontinuous. To establish the uniform boundedness, note that

$$\|X_\delta(t)\| \leq \|X_\delta(0)\| + \int_0^t \|\dot{X}_\delta(u)\| du \leq \|x_0\| + 30 \|\nabla f(x_0)\| / L. \quad \blacksquare$$

We are now ready for the proof of Lemma 14.

Proof By the Arzelà-Ascoli theorem and Lemma 18, \mathcal{F} contains a subsequence converging uniformly on $[0, \sqrt{6/L}]$. Denote by $\{X_{\delta_{m_i}}\}_{i \in \mathbb{N}}$ the convergent subsequence and \check{X} the limit. Above, $\delta_{m_i} = \sqrt{3/L}/2^{m_i}$ decreases as i increases. We will prove that \check{X} satisfies (3) and the initial conditions $\check{X}(0) = x_0, \check{X}'(0) = 0$.

Fix an arbitrary $t_0 \in (0, \sqrt{6/L})$. Since $\|\dot{X}_{\delta_{m_i}}(t_0)\|$ is bounded, we can pick a subsequence of $\dot{X}_{\delta_{m_i}}(t_0)$ which converges to a limit, denoted by $\dot{X}_{t_0}^D$. Without loss of generality, assume the subsequence is the original sequence. Denote by \tilde{X} the local solution to (3) with $X(t_0) = \tilde{X}(t_0)$ and $\dot{X}(t_0) = \dot{X}_{t_0}^D$. Now recall that $X_{\delta_{m_i}}$ is the solution to (3) with $X(t_0) = X_{\delta_{m_i}}(t_0)$ and $\dot{X}(t_0) = \dot{X}_{\delta_{m_i}}(t_0)$ when $\delta_{m_i} < t_0$. Since both $X_{\delta_{m_i}}(t_0)$ and $\dot{X}_{\delta_{m_i}}(t_0)$ approach $\tilde{X}(t_0)$ and $\dot{X}_{t_0}^D$, respectively, there exists $\epsilon_0 > 0$ such that

$$\sup_{t_0 - \epsilon_0 < t < t_0 + \epsilon_0} \|\dot{X}_{\delta_{m_i}}(t) - \tilde{X}(t)\| \rightarrow 0$$

as $i \rightarrow \infty$. However, by definition we have

$$\sup_{t_0 - \epsilon_0 < t < t_0 + \epsilon_0} \|\dot{X}_{\delta_{m_i}}(t) - \tilde{X}(t)\| \rightarrow 0.$$

Therefore \tilde{X} and \tilde{X} have to be identical on $(t_0 - \epsilon_0, t_0 + \epsilon_0)$. So \tilde{X} satisfies (3) at t_0 . Since t_0 is arbitrary, we conclude that \tilde{X} is a solution to (3) on $(0, \sqrt{6/L})$. By extension, \tilde{X} can be a global solution to (3) on $(0, \infty)$. It only leaves to verify the initial conditions to complete the proof.

The first condition $\tilde{X}(0) = x_0$ is a direct consequence of $\dot{X}_{\delta_{m_i}}(0) = x_0$. To check the second, pick a small $t > 0$ and note that

$$\begin{aligned} \frac{\|\tilde{X}(t) - \tilde{X}(0)\|}{t} &= \lim_{t \rightarrow \infty} \frac{\|\dot{X}_{\delta_{m_i}}(t) - \dot{X}_{\delta_{m_i}}(0)\|}{t} = \lim_{t \rightarrow \infty} \|\dot{X}_{\delta_{m_i}}(\xi_t)\| \\ &\leq \limsup_{t \rightarrow \infty} t M_{\delta_{m_i}}(t) \leq 5t \sqrt{6/L} \|\nabla f(x_0)\|, \end{aligned}$$

where $\xi_t \in (0, t)$ is given by the mean value theorem. The desired result follows from taking $t \rightarrow 0$. ■

Next, we aim to prove the uniqueness of the solution to (3).

Lemma 19 *For any $f \in \mathcal{F}_\infty$, the ODE (3) has at most one local solution in a neighborhood of $t = 0$.*

Suppose on the contrary that there are two solutions, namely: X and Y , both defined on $(0, \alpha)$ for some $\alpha > 0$. Define $\tilde{M}(t)$ to be the supremum of $\|\dot{X}(u) - \dot{Y}(u)\|$ over $u \in [0, t)$. To proceed, we need a simple auxiliary lemma.

Lemma 20 *For any $t \in (0, \alpha)$, we have*

$$\|\nabla f(X(t)) - \nabla f(Y(t))\| \leq Lt \tilde{M}(t).$$

Proof By Lipschitz continuity of the gradient, one has

$$\begin{aligned} \|\nabla f(X(t)) - \nabla f(Y(t))\| &\leq L \|X(t) - Y(t)\| = L \left\| \int_0^t \dot{X}(u) - \dot{Y}(u) du + X(0) - Y(0) \right\| \\ &\leq L \int_0^t \|\dot{X}(u) - \dot{Y}(u)\| du \leq Lt \tilde{M}(t). \end{aligned}$$

Now we prove Lemma 19.

Proof Similar to the proof of Lemma 17, we get

$$t^3 \|\dot{X}(t) - \dot{Y}(t)\| = - \int_0^t u^3 (\nabla f(X(u)) - \nabla f(Y(u))) du.$$

Applying Lemma 20 gives

$$t^3 \|\dot{X}(t) - \dot{Y}(t)\| \leq \int_0^t L u^4 \tilde{M}(u) du \leq \frac{1}{5} L t^5 \tilde{M}(t),$$

which can be simplified as $\|\dot{X}(t) - \dot{Y}(t)\| \leq L t^2 \tilde{M}(t)/5$. Thus, for any $t' \leq t$ it is true that $\|\dot{X}(t') - \dot{Y}(t')\| \leq L t'^2 \tilde{M}(t)/5$. Taking the supremum of $\|\dot{X}(t') - \dot{Y}(t')\|$ over $t' \in (0, t)$ gives $\tilde{M}(t) \leq L t^2 \tilde{M}(t)/5$. Therefore $\tilde{M}(t) = 0$ for $t < \min(\alpha, \sqrt{5/L})$, which is equivalent to saying $X = Y$ on $[0, \min(\alpha, \sqrt{5/L})]$. With the same initial value $X(0) = Y(0) = x_0$ and the same gradient, we conclude that X and Y are identical on $(0, \min(\alpha, \sqrt{5/L}))$, a contradiction. ■

Given all of the aforementioned lemmas, Theorem 1 follows from a combination of Lemmas 14 and 19.

Appendix B. Proof of Theorem 2

Identifying $\sqrt{s} = \Delta t$, the comparison between (4) and (15) reveals that Nesterov's scheme is a discrete scheme for numerically integrating the ODE (3). However, its singularity of the damping coefficient at $t = 0$ leads to the nonexistence of off-the-shelf ODE theory for proving Theorem 2. To address this difficulty, we use the smoothed ODE (31) to approximate the original one; then bound the difference between Nesterov's scheme and the forward Euler scheme of (31), which may take the following form:

$$\begin{aligned} X_{k+1}^{\delta} &= X_k^{\delta} + \Delta t Z_k^{\delta} \\ Z_{k+1}^{\delta} &= \left(1 - \frac{3\Delta t}{\max\{\delta, k\Delta t\}} \right) Z_k^{\delta} - \Delta t \nabla f(X_k^{\delta}) \end{aligned} \quad (32)$$

with $X_0^{\delta} = x_0$ and $Z_0^{\delta} = 0$.

Lemma 21 *With step size $\Delta t = \sqrt{s}$, for any $T > 0$ we have*

$$\max_{1 \leq k \leq \frac{T}{\sqrt{s}}} \|X_k^{\delta} - x_k\| \leq C \delta^2 + o_{\delta}(1)$$

for some constant C .

Proof Let $z_k = (x_{k+1} - x_k)/\sqrt{s}$. Then Nesterov's scheme is equivalent to

$$\begin{aligned} x_{k+1} &= x_k + \sqrt{s} z_k \\ z_{k+1} &= \left(1 - \frac{3}{k+3} \right) z_k - \sqrt{s} \nabla f \left(x_k + \frac{2k+3}{k+3} \sqrt{s} z_k \right). \end{aligned} \quad (33)$$

Denote by $a_k = \|X_k^\delta - x_k\|$, $b_k = \|Z_k^\delta - z_k\|$, whose initial values are $a_0 = 0$ and $b_0 = \|\nabla f(x_0)\|/\sqrt{s}$. The idea of this proof is to bound a_k via simultaneously estimating a_k and b_k . By comparing (32) and (33), we get the iterative relationship for a_k : $a_{k+1} \leq a_k + \sqrt{s}b_k$. Denoting by $S_k = b_0 + b_1 + \dots + b_k$, this yields

$$a_k \leq \sqrt{s}S_{k-1}. \quad (34)$$

Similarly, for sufficiently small s we get

$$\begin{aligned} b_{k+1} &\leq \left| 1 - \frac{3}{\max\{\delta/\sqrt{s}, k\}} b_k + L\sqrt{s}a_k + \left(\frac{3}{k+3} - \frac{3}{\max\{\delta/\sqrt{s}, k\}} + 2Ls \right) \|z_k\| \right| \\ &\leq b_k + L\sqrt{s}a_k + \left(\frac{3}{k+3} - \frac{3}{\max\{\delta/\sqrt{s}, k\}} + 2Ls \right) \|z_k\|. \end{aligned}$$

To upper bound $\|z_k\|$, denoting by C_1 the supremum of $\sqrt{2L}(f(y_k) - f^*)$ over all k and s , we have

$$\|z_k\| \leq \frac{k-1}{k+2} \|z_{k-1}\| + \sqrt{s} \|\nabla f(y_k)\| \leq \|z_{k-1}\| + C_1\sqrt{s},$$

which gives $\|z_k\| \leq C_1(k+1)\sqrt{s}$. Hence,

$$\begin{aligned} \left(\frac{3}{k+3} - \frac{3}{\max\{\delta/\sqrt{s}, k\}} + 2Ls \right) \|z_k\| &\leq \begin{cases} C_2\sqrt{s}, & k \leq \frac{\delta}{\sqrt{s}} \\ \frac{C_2\sqrt{s}}{k} < \frac{C_2s}{\theta}, & k > \frac{\delta}{\sqrt{s}}. \end{cases} \end{aligned} \quad (35)$$

Making use of (34) gives

$$b_{k+1} \leq \begin{cases} b_k + LsS_{k-1} + C_2\sqrt{s}, & k \leq \delta/\sqrt{s} \\ b_k + LsS_{k-1} + \frac{C_2s}{\theta}, & k > \delta/\sqrt{s}. \end{cases}$$

By induction on k , for $k \leq \delta/\sqrt{s}$ it holds that

$$b_k \leq \frac{C_1Ls + C_2 + (C_1 + C_2)\sqrt{Ls}}{2\sqrt{L}} (1 + \sqrt{Ls})^{k-1} - \frac{C_1Ls + C_2 - (C_1 + C_2)\sqrt{Ls}}{2\sqrt{L}} (1 - \sqrt{Ls})^{k-1}.$$

Hence,

$$S_k \leq \frac{C_1Ls + C_2 + (C_1 + C_2)\sqrt{Ls}}{2L\sqrt{s}} (1 + \sqrt{Ls})^k + \frac{C_1Ls + C_2 - (C_1 + C_2)\sqrt{Ls}}{2L\sqrt{s}} (1 - \sqrt{Ls})^k - \frac{C_2}{L\sqrt{s}}.$$

Letting $k^* = \lceil \delta/\sqrt{s} \rceil$, we get

$$\limsup_{s \rightarrow 0} \sqrt{s}S_{k^*-1} \leq \frac{C_2e^{\delta\sqrt{L}} + C_2e^{-\delta\sqrt{L}} - 2C_2}{2L} = O(\delta^2),$$

which allows us to conclude that

$$a_k \leq \sqrt{s}S_{k-1} = O(\delta^2) + o_s(1) \quad (36)$$

for all $k \leq \delta/\sqrt{s}$.

Next, we bound b_k for $k > k^* = \lceil \delta/\sqrt{s} \rceil$. To this end, we consider the worst case of (35), that is,

$$b_{k+1} = b_k + LsS_{k-1} + \frac{C_2s}{\theta}$$

for $k > k^*$ and $S_{k^*} = S_{k^*+1} = C_3\delta^2/\sqrt{s} + o_s(1/\sqrt{s})$ for some sufficiently large C_3 . In this case, $C_2s/\theta < sS_{k-1}$ for sufficiently small s . Hence, the last display gives

$$b_{k+1} \leq b_k + (L+1)sS_{k-1}.$$

By induction, we get

$$S_k \leq \frac{C_3\delta^2/\sqrt{s} + o_s(1/\sqrt{s})}{2} \left((1 + \sqrt{(L+1)s})^{k-k^*} + (1 - \sqrt{(L+1)s})^{k-k^*} \right).$$

Letting $k^\circ = \lceil T/\sqrt{s} \rceil$, we further get

$$\limsup_{s \rightarrow 0} \sqrt{s}S_{k^\circ} \leq \frac{C_3\delta^2(e^{(T-\delta)\sqrt{L+1}} + e^{-(T-\delta)\sqrt{L+1}})}{2} = O(\delta^2),$$

which yields

$$a_k \leq \sqrt{s}S_{k-1} = O(\delta^2) + o_s(1)$$

for $k^* < k \leq k^\circ$. Last, combining (36) and the last display, we get the desired result. \blacksquare

Now we turn to the proof of Theorem 2.

Proof. Note the triangular inequality

$$\|x_k - X(k\sqrt{s})\| \leq \|x_k - X_k^\delta\| + \|X_k^\delta - X_\delta(k\sqrt{s})\| + \|X_\delta(k\sqrt{s}) - X(k\sqrt{s})\|,$$

where $X_\delta(\cdot)$ is the solution to the smoothed ODE (31). The proof of Lemma 14 implies that, we can choose a sequence $\delta_m \rightarrow 0$ such that

$$\sup_{0 \leq t \leq T} \|X_{\delta_m}(t) - X(t)\| \rightarrow 0.$$

The second term $\|X_k^{\delta_m} - X_{\delta_m}(k\sqrt{s})\|$ will uniformly vanish as $s \rightarrow 0$ and so does the first term $\|x_k - X_k^{\delta_m}\|$ if first $s \rightarrow 0$ and then $\delta_m \rightarrow 0$. This completes the proof. \blacksquare

Appendix C. ODE for Composite Optimization

In analogy to (3) for smooth f in Section 2, we develop an ODE for composite optimization,

$$\text{minimize } f(x) = g(x) + h(x), \quad (37)$$

where $g \in \mathcal{F}_L$ and h is a general convex function possibly taking on the value $+\infty$. Provided it is easy to evaluate the proximal of h , Beck and Teboulle (2009) propose a proximal

gradient version of Nesterov's scheme for solving (37). It is to repeat the following recursion for $k \geq 1$,

$$\begin{aligned} x_k &= y_{k-1} - sG_k(y_{k-1}) \\ y_k &= x_k + \frac{k-1}{k+2}(x_k - x_{k-1}), \end{aligned}$$

where the proximal subgradient G_s has been defined in Section 4.1. If the constant step size $s \leq 1/L$, it is guaranteed that (Beck and Teboulle, 2009)

$$f(x_k) - f^* \leq \frac{2\|x_0 - x^*\|^2}{s(k+1)^2},$$

which in fact is a special case of Theorem 6.

Compared to the smooth case, it is not as clear to define the driving force as ∇f in (3). At first, it might be a good try to define

$$G(x) = \lim_{s \rightarrow 0} G_s(x) = \lim_{s \rightarrow 0} \frac{x - \operatorname{argmin}_z (\|z - (x - s\nabla g(x))\|^2 / (2s) + h(z))}{s},$$

if it exists. However, as implied in the proof of Theorem 24 stated below, this definition fails to capture the *directional* aspect of the subgradient. To this end, we define the subgradients through the following lemma.

Lemma 22 (Rockafellar, 1997) *For any convex function f and any $x, p \in \mathbb{R}^n$, the directional derivative $\lim_{t \rightarrow 0^+} (f(x+sp) - f(x))/s$ exists, and can be evaluated as*

$$\lim_{s \rightarrow 0^+} \frac{f(x+sp) - f(x)}{s} = \sup_{\xi \in \partial f(x)} \langle \xi, p \rangle.$$

Note that the directional derivative is semilinear in p because

$$\sup_{\xi \in \partial f(x)} \langle \xi, cp \rangle = c \sup_{\xi \in \partial f(x)} \langle \xi, p \rangle$$

for any $c > 0$.

Definition 23 *A Borel measurable function $G(x, p; f)$ defined on $\mathbb{R}^n \times \mathbb{R}^n$ is said to be a directional subgradient of f if*

$$\begin{aligned} G(x, p) &\in \partial f(x), \\ \langle G(x, p), p \rangle &= \sup_{\xi \in \partial f(x)} \langle \xi, p \rangle \end{aligned}$$

for all x, p .

Convex functions are naturally locally Lipschitz, so $\partial f(x)$ is compact for any x . Consequently there exists $\xi \in \partial f(x)$ which maximizes $\langle \xi, p \rangle$. So Lemma 22 guarantees the existence of a directional subgradient. The function G is essentially a function defined on $\mathbb{R}^n \times \mathbb{S}^{n-1}$ in that we can define

$$G(x, p) = G(x, p/\|p\|),$$

and $G(x, 0)$ to be any element in $\partial f(x)$. Now we give the main theorem. However, note that we do not guarantee the existence of solution to (38).

Theorem 24 *Given a convex function $f(x)$ with directional subgradient $G(x, p; f)$, assume that the second order ODE*

$$\ddot{X} + \frac{3}{t}\dot{X} + G(X, \dot{X}) = 0, \quad X(0) = x_0, \dot{X}(0) = 0 \quad (38)$$

admits a solution $X(t)$ on $[0, \alpha)$ for some $\alpha > 0$. Then for any $0 < t < \alpha$, we have

$$f(X(t)) - f^* \leq \frac{2\|x_0 - x^*\|_2^2}{t^2}.$$

Proof It suffices to establish that \mathcal{E} , first defined in the proof of Theorem 3, is monotonically decreasing. The difficulty comes from that \mathcal{E} may not be differentiable in this setting. Instead, we study $(\mathcal{E}(t + \Delta t) - \mathcal{E}(t))/\Delta t$ for small $\Delta t > 0$. In \mathcal{E} , the second term $2\|X + t\dot{X}/2 - x^*\|^2$ is differentiable, with derivative $4\langle X + \frac{t}{2}\dot{X} - x^*, \frac{t}{2}\dot{X} + \frac{t}{2}\ddot{X} \rangle$. Hence,

$$\begin{aligned} & 2\|X(t + \Delta t) + \frac{t}{2}\dot{X}(t + \Delta t) - x^*\|^2 - 2\|X(t) + \frac{t}{2}\dot{X}(t) - x^*\|^2 \\ &= 4\langle X + \frac{t}{2}\dot{X} - x^*, \frac{3}{2}\dot{X} + \frac{t}{2}\ddot{X} \rangle \Delta t + o(\Delta t) \\ &= -t^2 \langle \dot{X}, G(X, \dot{X}) \rangle \Delta t - 2t \langle X - x^*, G(X, \dot{X}) \rangle \Delta t + o(\Delta t). \end{aligned} \quad (39)$$

For the first term, note that

$$(t + \Delta t)^2 (f(X(t + \Delta t)) - f^*) - t^2 (f(X(t)) - f^*) = 2t(f(X(t + \Delta t)) - f^*) \Delta t + t^2 (f(X(t + \Delta t)) - f(X(t))) + o(\Delta t).$$

Since f is locally Lipschitz, $o(\Delta t)$ term does not affect the function in the limit,

$$f(X(t + \Delta t)) = f(X + \Delta t \dot{X} + o(\Delta t)) = f(X + \Delta t \dot{X}) + o(\Delta t). \quad (40)$$

By Lemma 22, we have the approximation

$$f(X + \Delta t \dot{X}) = f(X) + \langle \dot{X}, G(X, \dot{X}) \rangle \Delta t + o(\Delta t). \quad (41)$$

Combining all of (39), (40) and (41), we obtain

$$\begin{aligned} \mathcal{E}(t + \Delta t) - \mathcal{E}(t) &= 2t(f(X(t + \Delta t)) - f^*) \Delta t + t^2 \langle \dot{X}, G(X, \dot{X}) \rangle \Delta t - t^2 \langle \dot{X}, G(X, \dot{X}) \rangle \Delta t \\ &\quad - 2t \langle X - x^*, G(X, \dot{X}) \rangle \Delta t + o(\Delta t) \\ &= 2t(f(X) - f^*) \Delta t - 2t \langle X - x^*, G(X, \dot{X}) \rangle \Delta t + o(\Delta t) \leq o(\Delta t), \end{aligned}$$

where the last inequality follows from the convexity of f . Thus,

$$\limsup_{\Delta t \rightarrow 0^+} \frac{\mathcal{E}(t + \Delta t) - \mathcal{E}(t)}{\Delta t} \leq 0,$$

which along with the continuity of \mathcal{E} , concludes that $\mathcal{E}(t)$ is a non-increasing function of t . \blacksquare

We give a simple example as follows. Consider the Lasso problem

$$\text{minimize } \frac{1}{2}\|y - Ax\|^2 + \lambda\|x\|_1.$$

Any directional subgradients admits the form $G(x, p) = -A^T(y - Ax) + \lambda \text{sgn}(x, p)$, where

$$\text{sgn}(x, p)_i = \begin{cases} \text{sgn}(x_i), & x_i \neq 0 \\ \text{sgn}(p_i), & x_i = 0, p_i \neq 0 \\ \in [-1, 1], & x_i = 0, p_i = 0. \end{cases}$$

To encourage sparsity, for any index i with $x_i = 0, p_i = 0$, we let

$$G(x, p)_i = \text{sgn}(A_i^T(Ax - y)) (|A_i^T(Ax - y)| - \lambda)_+.$$

Appendix D. Proof of Theorem 9

Proof Let g be μ -strongly convex and h be convex. For $f = g + h$, we show that (22) can be strengthened to

$$f(y - sG_s(y)) \leq f(x) + G_s(y)^T(y - x) - \frac{s}{2}\|G_s(y)\|^2 - \frac{\mu}{2}\|y - x\|^2. \quad (42)$$

Summing $(4k-3) \times (42)$ with $x = x_{k-1}, y = y_{k-1}$ and $(4r-6) \times (42)$ with $x = x^*, y = y_{k-1}$ yields

$$\begin{aligned} (4k+4r-9)f(x_k) &\leq (4k-3)f(x_{k-1}) + (4r-6)f^* \\ &\quad + G_s(y_{k-1})^T[(4k+4r-9)y_{k-1} - (4k-3)x_{k-1} - (4r-6)x^*] \\ &\quad - \frac{s(4k+4r-9)}{2}\|G_s(y_{k-1})\|^2 - \frac{\mu(4k-3)}{2}\|y_{k-1} - x_{k-1}\|^2 - \mu(2r-3)\|y_{k-1} - x^*\|^2 \\ &\leq (4k-3)f(x_{k-1}) + (4r-6)f^* - \mu(2r-3)\|y_{k-1} - x^*\|^2 \\ &\quad + G_s(y_{k-1})^T[(4k+4r-9)(y_{k-1} - x^*) - (4k-3)(x_{k-1} - x^*)], \quad (43) \end{aligned}$$

which gives a lower bound on $G_s(y_{k-1})^T[(4k+4r-9)y_{k-1} - (4k-3)x_{k-1} - (4r-6)x^*]$. Denote by Δ_k the second term of $\mathcal{E}(k)$ in (28), namely,

$$\Delta_k \triangleq \frac{k+d}{8}\|(2k+2r-2)(y_k - x^*) - (2k+1)(x_k - x^*)\|^2,$$

where $d := 3r/2 - 5/2$. Then by (43), we get

$$\begin{aligned} \Delta_k - \Delta_{k-1} &= -\frac{k+d}{8}\left\langle s(2r+2k-5)G_s(y_{k-1}) + \frac{k-2}{k+r-2}(x_{k-1} - x_{k-2}), (4k+4r-9)(y_{k-1} - x^*) \right. \\ &\quad \left. - (4k-3)(x_{k-1} - x^*) \right\rangle + \frac{1}{8}\|(2k+2r-4)(y_{k-1} - x^*) - (2k-1)(x_{k-1} - x^*)\|^2 \\ &\leq -\frac{s(k+d)(2k+2r-5)}{8}[(4k+4r-9)(f(x_k) - f^*) \\ &\quad - (4k-3)(f(x_{k-1}) - f^*) + \mu(2r-3)\|y_{k-1} - x^*\|^2] \\ &\quad - \frac{(k+d)(k-2)}{8(k+r-2)}\langle x_{k-1} - x_{k-2}, (4k+4r-9)(y_{k-1} - x^*) - (4k-3)(x_{k-1} - x^*) \rangle \\ &\quad + \frac{1}{8}\|2(k+r-2)(y_{k-1} - x^*) - (2k-1)(x_{k-1} - x^*)\|^2. \end{aligned}$$

Hence,

$$\begin{aligned} \Delta_k + \frac{s(k+d)(2k+2r-5)(4k+4r-9)}{8}(f(x_k) - f^*) \\ \leq \Delta_{k-1} + \frac{s(k+d)(2k+2r-5)(4k-3)}{8}(f(x_{k-1}) - f^*) \\ - \frac{s\mu(2r-3)(k+d)(2k+2r-5)}{8}\|y_{k-1} - x^*\|^2 + \Pi_1 + \Pi_2, \quad (44) \end{aligned}$$

where

$$\Pi_1 \triangleq -\frac{(k+d)(k-2)}{8(k+r-2)}\langle x_{k-1} - x_{k-2}, (4k+4r-9)(y_{k-1} - x^*) - (4k-3)(x_{k-1} - x^*) \rangle,$$

$$\Pi_2 \triangleq \frac{1}{8}\|2(k+r-2)(y_{k-1} - x^*) - (2k-1)(x_{k-1} - x^*)\|^2.$$

By the iterations defined in (19), one can show that

$$\begin{aligned} \Pi_1 &= -\frac{(2r-3)(k+d)(k-2)}{8(k+r-2)}\frac{\|x_{k-1} - x^*\|^2 - \|x_{k-2} - x^*\|^2}{8(k+r-2)^2} \\ &\quad - \frac{(k-2)^2(4k+4r-9)(k+d) + (2r-3)(k-2)(k+r-2)(k+d)}{8(k+r-2)^2}\|x_{k-1} - x_{k-2}\|^2, \\ \Pi_2 &= \frac{(2r-3)^2}{8}\|y_{k-1} - x^*\|^2 + \frac{(2r-3)(2k-1)(k-2)}{8(k+r-2)}(\|x_{k-1} - x^*\|^2 - \|x_{k-2} - x^*\|^2) \\ &\quad + \frac{(k-2)^2(2k-1)(2k+4r-7) + (2r-3)(2k-1)(k-2)(k+r-2)}{8(k+r-2)^2}\|x_{k-1} - x_{k-2}\|^2. \end{aligned}$$

Although this is a little tedious, it is straightforward to check that $(k-2)^2(4k+4r-9)(k+d) + (2r-3)(k-2)(k+r-2)(k+d) \geq (k-2)^2(2k-1)(2k+4r-7) + (2r-3)(2k-1)(k-2)(k+r-2)$ for any k . Therefore, $\Pi_1 + \Pi_2$ is bounded as

$$\Pi_1 + \Pi_2 \leq \frac{(2r-3)^2}{8}\|y_{k-1} - x^*\|^2 + \frac{(2r-3)(k-d-1)(k-2)}{8(k+r-2)}(\|x_{k-1} - x^*\|^2 - \|x_{k-2} - x^*\|^2),$$

which, together with the fact that $s\mu(2r-3)(k+d)(2k+2r-5) \geq (2r-3)^2$ when $k \geq \sqrt{(2r-3)/(2s\mu)}$, reduces (44) to

$$\begin{aligned} \Delta_k &+ \frac{s(k+d)(2k+2r-5)(4k+4r-9)}{8} (f(x_k) - f^*) \\ &\leq \Delta_{k-1} + \frac{s(k+d)(2k+2r-5)(4k-3)}{8} (f(x_{k-1}) - f^*) \\ &\quad + \frac{(2r-3)(k-d-1)(k-2)}{8(k+r-2)} (\|x_{k-1} - x^*\|^2 - \|x_{k-2} - x^*\|^2). \end{aligned}$$

This can be further simplified as

$$\tilde{\mathcal{E}}(k) + A_k(f(x_{k-1}) - f^*) \leq \tilde{\mathcal{E}}(k-1) + B_k(\|x_{k-1} - x^*\|^2 - \|x_{k-2} - x^*\|^2) \quad (45)$$

for $k \geq \sqrt{(2r-3)/(2s\mu)}$, where $A_k = (8r-36)k^2 + (20r^2 - 126r + 200)k + 12r^3 - 100r^2 + 288r - 281 > 0$ since $r \geq 9/2$ and $B_k = (2r-3)(k-d-1)(k-2)/(8(k+r-2))$. Denote by $k^* = \lceil \max\{\sqrt{(2r-3)/(2s\mu)}, 3r/2 - 3/2\} \rceil \asymp 1/\sqrt{s\mu}$. Then B_k is a positive increasing sequence if $k > k^*$. Summing (45) from k to k^*+1 , we obtain

$$\begin{aligned} \mathcal{E}(k) &+ \sum_{i=k^*+1}^k A_i(f(x_{i-1}) - f^*) \leq \mathcal{E}(k^*) + \sum_{i=k^*+1}^k B_i(\|x_{i-1} - x^*\|^2 - \|x_{i-2} - x^*\|^2) \\ &= \mathcal{E}(k^*) + B_k\|x_{k-1} - x^*\|^2 - B_{k^*+1}\|x_{k^*-1} - x^*\|^2 + \sum_{i=k^*+1}^{k-1} (B_j - B_{j+1})\|x_{j-1} - x^*\|^2 \\ &\leq \mathcal{E}(k^*) + B_k\|x_{k-1} - x^*\|^2. \end{aligned}$$

Similarly, as in the proof of Theorem 8, we can bound $\mathcal{E}(k^*)$ via another energy functional defined from Theorem 5,

$$\begin{aligned} \mathcal{E}(k^*) &\leq \frac{s(2k^*+3r-5)(k^*+r-2)^2}{2} (f(x_{k^*}) - f^*) \\ &\quad + \frac{2k^*+3r-5}{16} \|2(k^*+r-1)yk^* - 2k^*x_{k^*} - 2(r-1)x^* - (x_{k^*} - x^*)\|^2 \\ &\quad \leq \frac{s(2k^*+3r-5)(k^*+r-2)^2}{2} (f(x_{k^*}) - f^*) \\ &\quad + \frac{2k^*+3r-5}{8} \|2(k^*+r-1)yk^* - 2k^*x_{k^*} - 2(r-1)x^*\|^2 \\ &\quad + \frac{2k^*+3r-5}{8} \|x_{k^*} - x^*\|^2 \leq \frac{(r-1)^2(2k^*+3r-5)}{2} \|x_0 - x^*\|^2 \\ &\quad + \frac{(r-1)^2(2k^*+3r-5)}{8s\mu(k^*+r-2)^2} \|x_0 - x^*\|^2 \lesssim \frac{\|x_0 - x^*\|^2}{\sqrt{s\mu}}. \quad (46) \end{aligned}$$

For the second term, it follows from Theorem 6 that

$$\begin{aligned} B_k\|x_{k-1} - x^*\|^2 &\leq \frac{(2r-3)(2k-3r+3)(k-2)}{8\mu(k+r-2)} (f(x_{k-1}) - x^*) \\ &\leq \frac{(2r-3)(2k-3r+3)(k-2)}{8\mu(k+r-2)} \frac{(r-1)^2\|x_0 - x^*\|^2}{2s(k+r-3)^2} \\ &\leq \frac{(2r-3)(r-1)^2(2k^*-3r+3)(k^*-2)}{16s\mu(k^*+r-2)(k^*+r-3)^2} \|x_0 - x^*\|^2 \lesssim \frac{\|x_0 - x^*\|^2}{\sqrt{s\mu}}. \end{aligned} \quad (47)$$

For $k > k^*$, (46) together with (47) this gives

$$\begin{aligned} f(x_k) - f^* &\leq \frac{16\mathcal{E}(k)}{s(2k+3r-5)(2k+2r-5)(4k+4r-9)} \\ &\leq \frac{16(\mathcal{E}(k^*) + B_k\|x_{k-1} - x^*\|^2)}{s(2k+3r-5)(2k+2r-5)(4k+4r-9)} \lesssim \frac{\|x_0 - x^*\|^2}{s^{\frac{3}{2}}\mu^{\frac{1}{2}}k^3}. \end{aligned}$$

To conclusion, note that by Theorem 6 the gap $f(x_k) - f^*$ for $k \leq k^*$ is bounded by

$$\frac{(r-1)^2\|x_0 - x^*\|^2}{2s(k+r-2)^2} = \frac{(r-1)^2\sqrt{s\mu}k^3\|x_0 - x^*\|^2}{2(k+r-2)^2} \lesssim \sqrt{s\mu}k^* \frac{\|x_0 - x^*\|^2}{s^{\frac{3}{2}}\mu^{\frac{1}{2}}k^3} \lesssim \frac{\|x_0 - x^*\|^2}{s^{\frac{3}{2}}\mu^{\frac{1}{2}}k^3}. \quad \blacksquare$$

Appendix E. Proof of Lemmas in Section 5

First, we prove Lemma 11.

Proof To begin with, note that the ODE (3) is equivalent to $d(t^{\frac{r}{2}}\dot{X}(t))/dt = -\partial^2\nabla f(\dot{X}(t))$, which by integration leads to

$$t^{\frac{3}{2}}\dot{X}(t) = -\frac{t^4}{4}\nabla f(x_0) - \int_0^t v^3(\nabla f(\dot{X}(u)) - \nabla f(x_0))du = -\frac{t^4}{4}\nabla f(x_0) - I(t). \quad (48)$$

Dividing (48) by t^4 and applying the bound on $I(t)$, we obtain

$$\frac{\|\dot{X}(t)\|}{t} \leq \frac{\|\nabla f(x_0)\|}{4} + \frac{\|I(t)\|}{t^4} \leq \frac{\|\nabla f(x_0)\|}{4} + \frac{LM(t)t^2}{12}.$$

Note that the right-hand side of the last display is monotonically increasing in t . Hence, by taking the supremum of the left-hand side over $(0, t]$, we get

$$M(t) \leq \frac{\|\nabla f(x_0)\|}{4} + \frac{LM(t)t^2}{12},$$

which completes the proof by rearrangement. \blacksquare

Next, we prove the lemma used in the proof of Lemma 12.

Lemma 25 *The speed restarting time T satisfies*

$$T(x_0, f) \geq \frac{4}{5\sqrt{L}}.$$

Proof The proof is based on studying $\langle \dot{X}(t), \ddot{X}(t) \rangle$. Dividing (48) by t^3 , we get an expression for \ddot{X} ,

$$\ddot{X}(t) = -\frac{t}{4}\nabla f(x_0) - \frac{1}{t^3} \int_0^t u^3(\nabla f(X(u)) - \nabla f(x_0))du. \quad (49)$$

Differentiating the above, we also obtain an expression for \ddot{X} :

$$\ddot{X}(t) = -\nabla f(\dot{X}(t)) + \frac{3}{4}\nabla f(x_0) + \frac{3}{t^4} \int_0^t u^3(\nabla f(X(u)) - \nabla f(x_0))du. \quad (50)$$

Using the two equations we can show that $d\|\dot{X}\|^2/dt = 2\langle \dot{X}(t), \ddot{X}(t) \rangle > 0$ for $0 < t < 4/(5\sqrt{L})$. Continue by observing that (49) and (50) yield

$$\begin{aligned} \langle \dot{X}(t), \ddot{X}(t) \rangle &= \left\langle -\frac{t}{4}\nabla f(x_0) - \frac{1}{t^3}I(t), -\nabla f(X(t)) + \frac{3}{4}\nabla f(x_0) + \frac{3}{t^4}I(t) \right\rangle \\ &\geq \frac{t}{4}\langle \nabla f(x_0), \nabla f(X(t)) \rangle - \frac{3t}{16}\|\nabla f(x_0)\|^2 - \frac{1}{t^3}\|I(t)\| \left(\|\nabla f(X(t))\| + \frac{3}{2}\|\nabla f(x_0)\| \right) - \frac{3}{t^7}\|I(t)\|^2 \\ &\geq \frac{t}{4}\|\nabla f(x_0)\|^2 - \frac{t}{4}\|\nabla f(x_0)\|\|\nabla f(X(t)) - \nabla f(x_0)\| - \frac{3t}{16}\|\nabla f(x_0)\|^2 \\ &\quad - \frac{LM(t)t^2}{12} \left(\|\nabla f(X(t)) - \nabla f(x_0)\| + \frac{5}{2}\|\nabla f(x_0)\| \right) - \frac{L^2M(t)t^5}{48} \\ &\geq \frac{t}{16}\|\nabla f(x_0)\|^2 - \frac{LM(t)t^2\|\nabla f(x_0)\|}{8} - \frac{LM(t)t^3}{2} \left(\frac{LM(t)t^2}{2} + \frac{5}{2}\|\nabla f(x_0)\| \right) - \frac{L^2M(t)t^5}{48} \\ &= \frac{t}{16}\|\nabla f(x_0)\|^2 - \frac{LM(t)t^3}{3}\|\nabla f(x_0)\| - \frac{L^2M(t)t^5}{16}. \end{aligned}$$

To complete the proof, applying Lemma 11, the last inequality yields

$$\langle \dot{X}(t), \ddot{X}(t) \rangle \geq \left(\frac{1}{16} - \frac{Lt^2}{12(1-Lt^2/12)} - \frac{L^2t^4}{256(1-Lt^2/12)^2} \right) \|\nabla f(x_0)\|^2 t \geq 0$$

for $t < \min\{\sqrt{12/L}, 4/(5\sqrt{L})\} = 4/(5\sqrt{L})$, where the positivity follows from

$$\frac{1}{16} - \frac{Lt^2}{12(1-Lt^2/12)} - \frac{L^2t^4}{256(1-Lt^2/12)^2} > 0,$$

which is valid for $0 < t \leq 4/(5\sqrt{L})$. ■

References

A. Beck. *Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB*. SIAM, 2014.

- A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.
- S. Becker, J. Bobin, and E. J. Candès. NESTA: A fast and accurate first-order method for sparse recovery. *SIAM Journal on Imaging Sciences*, 4(1):1–39, 2011.
- M. Bogdan, E. v. d. Berg, C. Sabatti, W. Su, and E. J. Candès. SLOPE—adaptive variable selection via convex optimization. *The Annals of Applied Statistics*, 9(3):1103–1140, 2015.
- S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.
- H.-B. Dürr and C. Ebenbauer. On a class of smooth optimization algorithms with applications in control. *Nonlinear Model Predictive Control*, 4(1):291–298, 2012.
- H.-B. Dürr, E. Saka, and C. Ebenbauer. A smooth vector field for quadratic programming. In *51st IEEE Conference on Decision and Control*, pages 2515–2520, 2012.
- S. Fiori. Quasi-geodesic neural learning algorithms over the orthogonal group: A tutorial. *Journal of Machine Learning Research*, 6:743–781, 2005.
- U. Helmke and J. Moore. Optimization and dynamical systems. *Proceedings of the IEEE*, 84(6):907, 1996.
- D. Hinton. Sturm's 1836 oscillation results evolution of the theory. In *Sturm-Liouville theory*, pages 1–27. Birkhäuser, Basel, 2005.
- J. J. Leader. *Numerical Analysis and Scientific Computation*. Pearson Addison Wesley, 2004.
- L. Lessard, B. Recht, and A. Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *arXiv preprint arXiv:1408.3595*, 2014.
- R. Monteiro, C. Ortiz, and B. Svaiter. An adaptive accelerated first-order method for convex optimization. Technical report, ISyE, Gatech, 2012.
- Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.
- Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, Boston, MA, 2004.
- Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.
- Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

- J. Nocedal and S. Wright. *Numerical Optimization*. Springer Science & Business Media, 2006.
- B. O'Donoghue and E. J. Candès. Adaptive restart for accelerated gradient schemes. *Found. Comput. Math.*, 2013.
- S. Osher, F. Ruan, J. Xiong, Y. Yao, and W. Yin. Sparse recovery via differential inclusions. *arXiv preprint arXiv:1406.7728*, 2014.
- B. T. Poljak. *Introduction to optimization*. Optimization Software New York, 1987.
- Z. Qin and D. Goldfarb. Structured sparsity via alternating direction methods. *Journal of Machine Learning Research*, 13(1):1435–1468, 2012.
- R. T. Rockafellar. *Convex Analysis*. Princeton Landmarks in Mathematics. Princeton University Press, 1997. Reprint of the 1970 original.
- A. P. Ruszczyński. *Nonlinear Optimization*. Princeton University Press, 2006.
- J. Schropp and I. Singer. A dynamical systems approach to constrained minimization. *Numerical functional analysis and optimization*, 21(3-4):537–551, 2000.
- N. Z. Shor. *Minimization Methods for Non-Differentiable Functions*. Springer Science & Business Media, 2012.
- I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1139–1147, 2013.
- P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. <http://pages.cs.wisc.edu/~brecht/cst726docs/Tseng.APG.pdf>, 2008.
- P. Tseng. Approximation accuracy, gradient methods, and error bound for structured convex optimization. *Mathematical Programming*, 125(2):263–295, 2010.
- G. N. Watson. *A Treatise on the Theory of Bessel Functions*. Cambridge Mathematical Library. Cambridge University Press, 1995. Reprint of the second (1944) edition.

Importance Weighting Without Importation Weights: An Efficient Algorithm for Combinatorial Semi-Bandits

Gergely Neu

Universitat Pompeu Fabra
Roc Boronol 138, 08018, Barcelona, Spain

Gábor Bartók

Google Zürich
Brandschenkestrasse 100, 8002, Zürich, Switzerland

GERGELY.NEU@GMAIL.COM

BARTOK@GOOGLE.COM

Editor: Manfred Warmuth

Abstract

We propose a sample-efficient alternative for importance weighting for situations where one only has sample access to the probability distribution that generates the observations. Our new method, called Geometric Resampling (GR), is described and analyzed in the context of online combinatorial optimization under semi-bandit feedback, where a learner sequentially selects its actions from a combinatorial decision set so as to minimize its cumulative loss. In particular, we show that the well-known Follow-the-Perturbed-Leader (FPL) prediction method coupled with Geometric Resampling yields the first computationally efficient reduction from offline to online optimization in this setting. We provide a thorough theoretical analysis for the resulting algorithm, showing that its performance is on par with previous, inefficient solutions. Our main contribution is showing that, despite the relatively large variance induced by the GR procedure, our performance guarantees hold with high probability rather than only in expectation. As a side result, we also improve the best known regret bounds for FPL in online combinatorial optimization with full feedback, closing the perceived performance gap between FPL and exponential weights in this setting.

Keywords: online learning, combinatorial optimization, bandit problems, semi-bandit feedback, follow the perturbed leader, importance weighting

1. Introduction

Importance weighting is a crucially important tool used in many areas of machine learning, and specifically online learning with partial feedback. While most work assumes that importance weights are readily available or can be computed with little effort during runtime, this is often not the case in many practical settings, even when one has cheap sample access to the distribution generating the observations. Among other cases, such situations may arise when observations are generated by complex hierarchical sampling schemes, probabilistic programs, or, more generally, black-box generative models. In this paper, we propose a simple and efficient sampling scheme called *Geometric Resampling (GR)* to compute reliable estimates of importance weights *using only sample access*.

Our main motivation is studying a specific online learning algorithm whose practical applicability in partial-feedback settings had long been hindered by the problem outlined above. Specifically, we consider the well-known *Follow-the-Perturbed-Leader (FPL)* prediction method that maintains implicit sampling distributions that usually cannot be expressed in closed form. In this paper, we endow FPL with our Geometric Resampling scheme to construct the first known computationally efficient reduction from offline to online combinatorial optimization under an important partial-

A preliminary version of this paper was published as Neu and Bartók (2013). Parts of this work were completed while Gergely Neu was with the Sequel team at INRIA Lille – Nord Europe, France and Gábor Bartók was with the Department of Computer Science at ETH Zürich.

Parameters: set of decision vectors $\mathcal{S} \subseteq \{0, 1\}^d$, number of rounds T ;
For all $t = 1, 2, \dots, T$, **repeat**

1. The learner chooses a probability distribution p_t over \mathcal{S} .
2. The learner draws action V_t randomly according to p_t .
3. The environment chooses loss vector ℓ_t .
4. The learner suffers loss $V_t^\top \ell_t$.
5. The learner observes some feedback based on ℓ_t and V_t .

Figure 1: The protocol of online combinatorial optimization.

information scheme known as *semi-bandit feedback*. In the rest of this section, we describe our precise setting, present related work and outline our main results.

1.1 Online Combinatorial Optimization

We consider a special case of online linear optimization known as online combinatorial optimization (see Figure 1). In every round $t = 1, 2, \dots, T$ of this sequential decision problem, the learner chooses an *action* V_t from the finite action set $\mathcal{S} \subseteq \{0, 1\}^d$, where $\|v\|_1 \leq m$ holds for all $v \in \mathcal{S}$. At the same time, the environment fixes a loss vector $\ell_t \in [0, 1]^d$ and the learner suffers loss $V_t^\top \ell_t$. The goal of the learner is to minimize the cumulative loss $\sum_{t=1}^T V_t^\top \ell_t$. As usual in the literature of online optimization (Cesa-Bianchi and Lugosi, 2006), we measure the performance of the learner in terms of the *regret* defined as

$$R_T = \max_{v \in \mathcal{S}} \sum_{t=1}^T (V_t - v)^\top \ell_t = \sum_{t=1}^T V_t^\top \ell_t - \min_{v \in \mathcal{S}} \sum_{t=1}^T v^\top \ell_t, \quad (1)$$

that is, the gap between the total loss of the learning algorithm and the best fixed decision in hindsight. In the current paper, we focus on the case of *non-oblivious* (or *adaptive*) environments, where we allow the loss vector ℓ_t to depend on the previous decisions V_1, \dots, V_{t-1} in an arbitrary fashion. Since it is well-known that no deterministic algorithm can achieve sublinear regret under such weak assumptions, we will consider learning algorithms that choose their decisions in a randomized way.

For such learners, another performance measure that we will study is the *expected regret* defined as

$$\widehat{R}_T = \max_{v \in \mathcal{S}} \sum_{t=1}^T \mathbb{E} [(V_t - v)^\top \ell_t] = \mathbb{E} \left[\sum_{t=1}^T V_t^\top \ell_t \right] - \min_{v \in \mathcal{S}} \mathbb{E} \left[\sum_{t=1}^T v^\top \ell_t \right].$$

The framework described above is general enough to accommodate a number of interesting problem instances such as path planning, ranking and matching problems, finding minimum-weight spanning trees and cut sets. Accordingly, different versions of this general learning problem have drawn considerable attention in the past few years. These versions differ in the amount of information made available to the learner after each round t . In the simplest setting, called the *full-information* setting, it is assumed that the learner gets to observe the loss vector ℓ_t , regardless of the choice of V_t . As this assumption does not hold for many practical applications, it is more interesting to study the problem under *partial-information* constraints, meaning that the learner only gets some limited feedback based on its own decision. In the current paper, we focus on a more realistic partial-information scheme known as *semi-bandit feedback* (Audibert, Bubeck, and Lugosi, 2014) where the

learner only observes the components $\ell_{t,i}$ of the loss vector for which $V_{t,i} = 1$, that is, the losses associated with the components selected by the learner.¹

1.2 Related Work

The most well-known instance of our problem is the *multi-armed bandit* problem considered in the seminal paper of Auer, Cesa-Bianchi, Freund, and Schapire (2002): in each round of this problem, the learner has to select one of N arms and minimize regret against the best fixed arm while only observing the losses of the chosen arms. In our framework, this setting corresponds to setting $d = N$ and $m = 1$. Among other contributions concerning this problem, Auer et al. propose an algorithm called **Exp3** (Exploration and Exploitation using Exponential weights) based on constructing loss estimates $\hat{\ell}_{t,i}$ for each component of the loss vector and playing arm i with probability proportional to $\exp(-\eta \sum_{s=1}^{t-1} \hat{\ell}_{s,i})$ at time t , where $\eta > 0$ is a parameter of the algorithm, usually called the learning rate². This algorithm is essentially a variant of the Exponentially Weighted Average (EWA) forecaster (a variant of weighted majority algorithm of Littlestone and Warmuth, 1994, and aggregating strategies of Vovk, 1990, also known as Hedge by Freund and Schapire, 1997). Besides proving that the *expected* regret of **Exp3** is $O(\sqrt{NT \log N})$, Auer et al. also provide a general lower bound of $\Omega(\sqrt{NT})$ on the regret of any learning algorithm on this particular problem. This lower bound was later matched by a variant of the Implicitly Normalized Forecaster (INF) of Audibert and Bubeck (2010) by using the same loss estimates in a more refined way. Audibert and Bubeck also show bounds of $O(\sqrt{NT/\log N} \log(N/\delta))$ on the regret that hold with probability at least $1 - \delta$, uniformly for any $\delta > 0$.

The most popular example of online learning problems with actual combinatorial structure is the shortest path problem first considered by Takimoto and Warmuth (2003) in the full information scheme. The same problem was considered by Györfy, Lindner, Lugosi and Ottucsák (2007), who proposed an algorithm that works with semi-bandit information. Since then, we have come a long way in understanding the ‘‘price of information’’ in online combinatorial optimization—see Audibert, Bubeck, and Lugosi (2014) for a complete overview of results concerning all of the information schemes considered in the current paper. The first algorithm directly targeting general online combinatorial optimization problems is due to Koolen, Warmuth, and Kivinen (2010): their method named **Component Hedge** guarantees an optimal regret of $O(m\sqrt{T \log(d/m)})$ in the full information setting. As later shown by Audibert, Bubeck, and Lugosi (2014), this algorithm is an instance of a more general algorithm class known as Online Stochastic Mirror Descent (OSMD). Taking the idea one step further, Audibert, Bubeck, and Lugosi (2014) also show that OSMD-based methods can also be shown to provide expected regret bounds of $O(\sqrt{m d T})$ for the semi-bandit setting, which is also used to coincide with the minimax regret in this setting. For completeness, we note that the EWA forecaster is known to attain an expected regret of $O(m^{3/2} \sqrt{T \log(d/m)})$ in the full information case and $O(m \sqrt{d T \log(d/m)})$ in the semi-bandit case.

While the results outlined above might suggest that there is absolutely no work left to be done in the full information and semi-bandit schemes, we get a different picture if we restrict our attention to *computationally efficient* algorithms. First, note that methods based on exponential weighting of each decision vector can only be efficiently implemented for a handful of decision sets S —see Koolen et al. (2010) and Cesa-Bianchi and Lugosi (2012) for some examples. Furthermore, as noted by Audibert et al. (2014), OSMD-type methods can be efficiently implemented by convex programming if the convex hull of the decision set can be described by a polynomial number of constraints. Details of such an efficient implementation are worked out by Suehiro, Hataano, Kijima, Takimoto, and Nngano (2012), whose algorithm runs in $O(d^{\delta})$ time, which can still be prohibitive in practical applications.

¹ Here, $V_{t,i}$ and $\ell_{t,i}$ are the i th components of the vectors V_t and ℓ_t , respectively.

² In fact, Auer et al. mix the resulting distribution with a uniform distribution over the arms with probability ηN . However, this modification is not needed when one is concerned with the total expected regret, see, e.g., Bubeck and Cesa-Bianchi (2012, Section 3.1).

While Koolen et al. (2010) list some further examples where OSMD can be implemented efficiently, we conclude that there is no general efficient algorithm with near-optimal performance guarantees for learning in combinatorial semi-bandits.

The Follow-the-Perturbed-Leader (FPL) prediction method (first proposed by Hannan, 1957 and later rediscovered by Kalai and Vempala, 2005) offers a computationally efficient solution for the online combinatorial optimization problem given that the *static* combinatorial optimization problem minimizes $v^T \ell$ admits computationally efficient solutions for any $\ell \in \mathbb{R}^d$. The idea underlying FPL is very simple: in every round t , the learner draws some random perturbations $Z_t \in \mathbb{R}^d$ and selects the action that minimizes the perturbed total losses:

$$V_t = \arg \min_{v \in S} v^T \left(\sum_{s=1}^{t-1} \ell_s - Z_t \right).$$

Despite its conceptual simplicity and computational efficiency, FPL have been relatively overlooked until very recently; due to two main reasons:

- The best known bound for FPL in the full information setting is $O(m\sqrt{dT})$, which is worse than the bounds for both EWA and OSMD that scale only logarithmically with d .
- Considering bandit information, no efficient FPL-style algorithm is known to achieve a regret of $O(\sqrt{T})$. On one hand, it is relatively straightforward to prove $O(T^{2/3})$ bounds on the expected regret for an efficient FPL-variant (see, e.g., Averbuch and Kleinberg, 2004 and McMahan and Blum, 2004). Poland (2005) proved bounds of $O(\sqrt{NT \log N})$ in the N -armed bandit setting, however, the proposed algorithm requires $O(T^2)$ numerical operations per round.

The main obstacle for constructing a computationally efficient FPL-variant that works with partial information is precisely the lack of closed-form expressions for importance weights. In the current paper, we address the above two issues and show that an efficient FPL-based algorithm using independent exponentially distributed perturbations can achieve as good performance guarantees as EWA in online combinatorial optimization.

Our work contributes to a new wave of positive results concerning FPL. Besides the reservations towards FPL mentioned above, the reputation of FPL has been also suffering from the fact that the nature of regularization arising from perturbations is not as well-understood as the explicit regularization schemes underlying OSMD or EWA. Very recently, Abernethy et al. (2014) have shown that FPL implements a form of strongly convex regularization over the convex hull of the decision space. Furthermore, Rakhtin et al. (2012) showed that FPL run with a specific perturbation scheme can be regarded as a relaxation of the minimax algorithm. Another recently initiated line of work shows that intuitive *parameter-free* variants of FPL can achieve excellent performance in full-information settings (Devroye et al., 2013 and Van Erven et al., 2014).

1.3 Our Results

In this paper, we propose a loss-estimation scheme called Geometric Resampling to efficiently compute importance weights for the observed components of the loss vector. Building on this technique and the FPL principle, resulting in an *efficient algorithm for regret minimization under semi-bandit feedback*. Besides this contribution, our techniques also enable us to improve the best known regret bounds of FPL in the full information case. We prove the following results concerning variants of our algorithm:

- a bound of $O(m \sqrt{d T \log(d/m)})$ on the expected regret under semi-bandit feedback (Theorem 1),
- a bound of $O(m \sqrt{d T \log(d/m)} + \sqrt{m d T} \log(1/\delta))$ on the regret that holds with probability at least $1 - \delta$, uniformly for all $\delta \in (0, 1)$ under semi-bandit feedback (Theorem 2),

- a bound of $O(m^{3/2}\sqrt{T\log(d/m)})$ on the expected regret under full information (Theorem 13).

We also show that both of our semi-bandit algorithms access the optimization oracle $O(dT)$ times over T rounds with high probability, increasing the running time only by a factor of d compared to the full-information variant. Notably, our results close the gaps between the performance bounds of FPL and EWA under both full information and semi-bandit feedback. Table 1 puts our newly proven regret bounds into context.

	FPL	EWA	OSMD
Full info regret bound	$m^{3/2}\sqrt{T\log\frac{d}{m}}$	$m^{3/2}\sqrt{T\log\frac{d}{m}}$	$m\sqrt{T\log\frac{d}{m}}$
Semi-bandit regret bound	$m\sqrt{dT\log\frac{d}{m}}$	$m\sqrt{dT\log\frac{d}{m}}$	\sqrt{mdT}
Computationally efficient?	always	sometimes	sometimes

Table 1: Upper bounds on the regret of various algorithms for online combinatorial optimization, up to constant factors. The third row roughly describes the computational efficiency of each algorithm—see the text for details. New results are presented in boldface.

2. Geometric Resampling

In this section, we introduce the main idea underlying Geometric Resampling in the specific context of N -armed bandits where $d = N$, $m = 1$ and the learner has access to the basis vectors $\{\mathbf{e}_i\}_{i=1}^d$ as its decision set \mathcal{S} . In this setting, components of the decision vector are referred to as *arms*. For ease of notation, define I_t as the unique arm such that $V_{t,I_t} = 1$ and \mathcal{F}_{t-1} as the sigma-algebra induced by the learner’s actions and observations up to the end of round $t - 1$. Using this notation, we define $P_{t,i} = \mathbb{P}[I_t = i | \mathcal{F}_{t-1}]$.

Most bandit algorithms rely on feeding some loss estimates to a sequential prediction algorithm. It is commonplace to consider *importance-weighted* loss estimates of the form

$$\widehat{\ell}_{t,i}^* = \frac{\mathbb{1}_{\{I_t=i\}} \ell_{t,i}}{P_{t,i}} \quad (2)$$

for all t, i such that $P_{t,i} > 0$. It is straightforward to show that $\widehat{\ell}_{t,i}^*$ is an unbiased estimate of the loss $\ell_{t,i}$ for all such t, i . Otherwise, when $P_{t,i} = 0$, we set $\widehat{\ell}_{t,i}^* = 0$, which gives $\mathbb{E}[\widehat{\ell}_{t,i}^* | \mathcal{F}_{t-1}] = 0 \leq \ell_{t,i}$.

To our knowledge, all existing bandit algorithms operating in the non-stochastic setting utilize some version of the importance-weighted loss estimates described above. This is a very natural choice for algorithms that operate by first computing the probabilities $P_{t,i}$ and then sampling I_t from the resulting distributions. While many algorithms fall into this class (including the Exp3 algorithm of Auer et al. (2002), the Green algorithm of Allenberg et al. (2006) and the INF algorithm of Audibert and Bubeck (2010), one can think of many other algorithms where the distribution \mathbf{p}_t is specified implicitly and thus importance weights are not readily available. Arguably, FPL is the most important online prediction algorithm that operates with implicit distributions that are notoriously difficult to compute in closed form. To overcome this difficulty, we propose a different loss estimate that can be efficiently computed *even when \mathbf{p}_t is not available for the learner*.

Our estimation procedure dubbed Geometric Resampling (GR) is based on the simple observation that, even though $P_{t,i}$ might not be computable in closed form, one can simply generate a geometric random variable with expectation $1/P_{t,i}$ by repeated sampling from \mathbf{p}_t . Specifically, we propose the following procedure to be executed in round t :

Geometric Resampling for multi-armed bandits

1. The learner draws $I_t \sim \mathbf{p}_t$.
2. For $k = 1, 2, \dots$
 - (a) Draw $I'_t(k) \sim \mathbf{p}_t$.
 - (b) If $I'_t(k) = I_t$, break.
3. Let $K_t = k$.

Observe that K_t generated this way is a geometrically distributed random variable given I_t and \mathcal{F}_{t-1} . Consequently, we have $\mathbb{E}[K_t | \mathcal{F}_{t-1}, I_t] = 1/P_{t,I_t}$. We use this property to construct the estimates

$$\widehat{\ell}_{t,i} = K_{t,i} \mathbb{1}_{\{I_t=i\}} \ell_{t,i} \quad (3)$$

for all arms i . We can easily show that the above estimate is unbiased whenever $P_{t,i} > 0$:

$$\begin{aligned} \mathbb{E}[\widehat{\ell}_{t,i} | \mathcal{F}_{t-1}] &= \sum_j P_{t,j} \mathbb{E}[\widehat{\ell}_{t,i} | \mathcal{F}_{t-1}, I_t = j] \\ &= P_{t,i} \mathbb{E}[\ell_{t,i} K_t | \mathcal{F}_{t-1}, I_t = i] \\ &= P_{t,i} \ell_{t,i} \mathbb{E}[K_t | \mathcal{F}_{t-1}, I_t = i] \\ &= \ell_{t,i}. \end{aligned}$$

Notice that the above procedure produces $\widehat{\ell}_{t,i} = 0$ almost surely whenever $P_{t,i} = 0$, giving $\mathbb{E}[\widehat{\ell}_{t,i} | \mathcal{F}_{t-1}] = 0$ for such t, i .

One practical concern with the above sampling procedure is that its worst-case running time is unbounded: while the expected number of necessary samples K_t is clearly N , the actual number of samples might be much larger. In the next section, we offer a remedy to this problem, as well as generalize the approach to work in the combinatorial semi-bandit case.

3. An Efficient Algorithm for Combinatorial Semi-Bandits

In this section, we present our main result: an efficient reduction from offline to online combinatorial optimization under semi-bandit feedback. The most critical element in our technique is extending the Geometric Resampling idea to the case of combinatorial action sets. For defining the procedure, let us assume that we are running a randomized algorithm mapping histories to probability distributions over the action set \mathcal{S} : letting \mathcal{F}_{t-1} denote the sigma-algebra induced by the history of interaction between the learner and the environment, the algorithm picks action $\mathbf{v} \in \mathcal{S}$ with probability $P_t(\mathbf{v}) = \mathbb{P}[V_t = \mathbf{v} | \mathcal{F}_{t-1}]$. Also introducing $q_{t,i} = \mathbb{E}[V_{t,i} | \mathcal{F}_{t-1}]$, we can define the counterpart of the standard importance-weighted loss estimates of Equation 2 as the vector $\widehat{\ell}_t^*$ with components

$$\widehat{\ell}_{t,i}^* = \frac{V_{t,i} \ell_{t,i}}{q_{t,i}} \quad (4)$$

Again, the problem with these estimates is that for many algorithms of practical interest, the importance weights $q_{t,i}$ cannot be computed in closed form. We now extend the Geometric Resampling procedure defined in the previous section to estimate the importance weights in an efficient manner. One adjustment we make to the procedure presented in the previous section is capping off the number of samples at some finite $M > 0$. While this capping obviously introduces some bias, we will show later that for appropriate values of M , this bias does not hurt the performance of

the overall learning algorithm too much. Thus, we define the Geometric Resampling procedure for combinatorial semi-bands as follows:

Geometric Resampling for combinatorial semi-bands

1. The learner draws $V_t \sim p_t$.
2. For $k = 1, 2, \dots, M$, draw $V_t^i(k) \sim p_t$.
3. For $i = 1, 2, \dots, d$,

$$K_{t,i} = \min\{k : V_t^i(k) = 1\} \cup \{M\}.$$

Based on the random variables output by the GR procedure, we construct our loss-estimate vector $\hat{\ell}_t \in \mathbb{R}^d$ with components

$$\hat{\ell}_{t,i} = K_{t,i} V_{t,i} \ell_{t,i} \quad (5)$$

for all $i = 1, 2, \dots, d$. Since $V_{t,i}$ are nonzero only for coordinates for which $\ell_{t,i}$ is observed, these estimates are well-defined. It also follows that the sampling procedure can be terminated once for every i with $V_{t,i} = 1$, there is a copy $V_t^i(k)$ such that $V_t^i(k) = 1$.

Now everything is ready to define our algorithm: FPL+GR, standing for Follow-the-Perturbed-Leader with Geometric Resampling. Defining $\tilde{L}_t = \sum_{s=1}^t \hat{\ell}_s$, at time step t FPL+GR draws the components of the perturbation vector Z_t independently from a standard exponential distribution and selects action³

$$V_t = \arg \min_{v \in S} v^\top (\eta \tilde{L}_{t-1} - Z_t), \quad (6)$$

where $\eta > 0$ is a parameter of the algorithm. As we mentioned earlier, the distribution p_t , while implicitly specified by Z_t and the estimated cumulative losses L_{t-1} , cannot usually be expressed in closed form for FPL.⁴ However, sampling the actions $V_t^i(\cdot)$ can be carried out by drawing additional perturbation vectors $Z_t^i(\cdot)$ independently from the same distribution as Z_t and then solving a linear optimization task. We emphasize that the above additional actions are *never actually played by the algorithm*, but are only necessary for constructing the loss estimates. The power of FPL+GR is that, unlike other algorithms for combinatorial semi-bands, its implementation only requires access to a linear optimization oracle over S . We point the reader to Section 3.2 for a more detailed discussion of the running time of FPL+GR. Pseudocode for FPL+GR is shown on as Algorithm 1.

As we will show shortly, FPL+GR as defined above comes with strong performance guarantees that hold *in expectation*. One can think of several possible ways to robustify FPL+GR so that it provides bounds that hold with high probability. One possible path is to follow Aber et al. (2002) and define the loss-estimate vector $\hat{\ell}_t^*$ with components

$$\hat{\ell}_{t,i}^* = \hat{\ell}_{t,i} - \frac{\beta}{q_{t,i}}$$

for some $\beta > 0$. The obvious problem with this definition is that it requires perfect knowledge of the importance weights $q_{t,i}$ for all i . While it is possible to extend Geometric Resampling developed in the previous sections to construct a reliable proxy to the above loss estimate, there are several downsides to this approach. First, observe that one would need to obtain estimates of $1/q_{t,i}$ for every single i —even for the ones for which $V_{t,i} = 0$. Due to this necessity, there is no hope to terminate

³ By the definition of the perturbation distribution, the minimum is unique almost surely.

⁴ One notable exception is when the perturbations are drawn independently from standard Gumbel distributions, and the decision set is the d -dimensional simplex: in this case, FPL is known to be equivalent with ERM—see, e.g., Abernethy et al. (2014) for further discussion.

Algorithm 1: FPL+GR implemented with a waiting list. The notation $\mathbf{a} \circ \mathbf{b}$ stands for elementwise product of vectors \mathbf{a} and \mathbf{b} : $(\mathbf{a} \circ \mathbf{b})_i = a_i b_i$ for all i .

```

Input:  $S \subseteq \{0, 1\}^d$ ,  $\eta \in \mathbb{R}^+$ ,  $M \in \mathbb{Z}^+$ ;
Initialization:  $\hat{L} = \mathbf{0} \in \mathbb{R}^d$ ;
for  $t = 1, \dots, T$  do
  Draw  $Z_t \in \mathbb{R}^d$  with independent components  $Z_{t,i} \sim \text{Exp}(1)$ ;
  Choose action  $V = \arg \min_{v \in S} \{v^\top (\eta \hat{L} - Z)\}$ ; /* Follow the perturbed leader */
  for  $k = 1, \dots, M$  do /* Initialize waiting list and counters */
     $K = K + \mathbf{r}$ ; /* Geometric Resampling */
    /* Increment counter */
  Draw  $Z_t^i \in \mathbb{R}^d$  with independent components  $Z_{t,i}^i \sim \text{Exp}(1)$ ; /* Sample a copy of  $V$  */
   $V^i = \arg \min_{v \in S} \{v^\top (\eta \hat{L} - Z^i)\}$ ; /* Update waiting list */
   $\mathbf{r} = \mathbf{r} \circ V^i$ ; /* All indices recurred */
  if  $\mathbf{r} = \mathbf{0}$  then break;
   $\hat{L} = \hat{L} + K \circ V \circ \ell$ ; /* Update cumulative loss estimates */
end

```

the sampling procedure in reasonable time. Second, reliable estimation requires multiple samples of $K_{t,i}$, where the sample size has to explicitly depend on the desired confidence level.

Thus, we follow a different path: Motivated by the work of Audibert and Bubeck (2010), we propose to use a loss-estimate vector $\tilde{\ell}_t$ with components of the form

$$\tilde{\ell}_{t,i} = \frac{1}{\beta} \log \left(1 + \beta \hat{\ell}_{t,i} \right) \quad (7)$$

with an appropriately chosen $\beta > 0$. Then, defining $\tilde{L}_{t-1} = \sum_{s=1}^{t-1} \tilde{\ell}_s$, we propose a variant of FPL+GR that simply replaces \tilde{L}_{t-1} by \tilde{L}_{t-1} in the rule (6) for choosing V_t . We refer to this variant of FPL+GR as FPL+GR, P. In the next section, we provide performance guarantees for both algorithms.

3.1 Performance Guarantees

Now we are ready to state our main results. Proofs will be presented in Section 4. First, we present a performance guarantee for FPL+GR in terms of the *expected regret*:

Theorem 1 *The expected regret of FPL+GR satisfies*

$$\hat{R}_T \leq \frac{m(\log(d/m) + 1)}{\eta} + 2mndT + \frac{dT}{cM}$$

under semi-bandit information. In particular, with

$$\eta = \sqrt{\frac{\log(d/m) + 1}{2dT}} \quad \text{and} \quad M = \left\lceil \frac{\sqrt{dT}}{cm\sqrt{2(\log(d/m) + 1)}} \right\rceil,$$

the expected regret of FPL+GR is bounded as

$$\hat{R}_T \leq 3m\sqrt{2dT \left(\log \frac{d}{m} + 1 \right)}.$$

Our second main contribution is the following bound on the regret of FPL+GR . \mathcal{P} .

Theorem 2 Fix an arbitrary $\delta > 0$. With probability at least $1 - \delta$, the regret of FPL+GR . \mathcal{P} satisfies

$$\begin{aligned} R_T \leq & \frac{m(\log(d/m) + 1)}{\eta} + \eta \left(Mm\sqrt{2T \log \frac{5}{\delta}} + 2md\sqrt{T \log \frac{5}{\delta}} + 2mdT \right) + \frac{dT}{eM} \\ & + \beta \left(M\sqrt{2mT \log \frac{5}{\delta}} + 2d\sqrt{T \log \frac{5}{\delta}} + 2dT \right) + \frac{m \log(5d/\delta)}{\beta} \\ & + m\sqrt{2(e-2)T \log \frac{5}{\delta}} + \sqrt{8T \log \frac{5}{\delta}} + \sqrt{2(e-2)T}. \end{aligned}$$

In particular, with

$$M = \left\lceil \sqrt{\frac{dT}{m}} \right\rceil, \quad \beta = \sqrt{\frac{m}{dT}}, \quad \text{and} \quad \eta = \sqrt{\frac{\log(d/m) + 1}{dT}},$$

the regret of FPL+GR . \mathcal{P} is bounded as

$$\begin{aligned} R_T \leq & 3m\sqrt{dT \left(\log \frac{d}{m} + 1 \right)} + \sqrt{mdT} \left(\log \frac{5d}{\delta} + 2 \right) + \sqrt{2mT \log \frac{5}{\delta}} \left(\sqrt{\log \frac{d}{m}} + 1 + 1 \right) \\ & + 1.2m\sqrt{T \log \frac{5}{\delta}} + \sqrt{T} \left(\sqrt{8 \log \frac{5}{\delta}} + 1.2 \right) + 2\sqrt{d \log \frac{5}{\delta}} \left(m\sqrt{\log \frac{d}{m}} + 1 + \sqrt{m} \right) \end{aligned}$$

with probability at least $1 - \delta$.

3.2 Running Time

Let us now turn our attention to computational issues. First, we note that the efficiency of FPL -type algorithms crucially depends on the availability of an efficient oracle that solves the static combinatorial optimization problem of finding $\arg \min_{v \in \mathcal{S}} v^T \ell$. Computing the running time of the full-information variant of FPL is straightforward: assuming that the oracle computes the solution to the static problem in $O(f(S))$ time, FPL returns its prediction in $O(f(S) + d)$ time (with the d overhead coming from the time necessary to generate the perturbations). Naturally, our loss estimation scheme multiplies these computations by the number of samples taken in each round. While terminating the estimation procedure after M samples helps in controlling the running time with high probability, observe that the naive bound of MT on the number of samples becomes way too large when setting M as suggested by Theorems 1 and 2. The next proposition shows that the amortized running time of Geometric Resampling remains as low as $O(d)$ even for large values of M .

Proposition 3 Let S_t denote the number of sample actions taken by GR in round t . Then, $\mathbb{E}[S_t] \leq d$. Also, for any $\delta > 0$,

$$\sum_{t=1}^T S_t \leq (e-1)dT + M \log \frac{1}{\delta}$$

holds with probability at least $1 - \delta$.

Proof For proving the first statement, let us fix a time step t and notice that

$$S_t = \max_{j: V_{t,j}=1} K_{t,j} = \max_{j=1,2,\dots,d} V_{t,j} K_{t,j} \leq \sum_{j=1}^d V_{t,j} K_{t,j}.$$

Now, observe that $\mathbb{E}[K_{t,j} | \mathcal{F}_{t-1}, V_{t,j}] \leq 1/\mathbb{E}[V_{t,j} | \mathcal{F}_{t-1}]$, which gives $\mathbb{E}[S_t] \leq d$, thus proving the first statement. For the second part, notice that $X_t = (S_t - \mathbb{E}[S_t | \mathcal{F}_{t-1}])$ is a martingale-difference sequence with respect to (\mathcal{F}_t) with $X_t \leq M$ and with conditional variance

$$\begin{aligned} \text{Var}[X_t | \mathcal{F}_{t-1}] &= \mathbb{E} \left[(S_t - \mathbb{E}[S_t | \mathcal{F}_{t-1}])^2 | \mathcal{F}_{t-1} \right] \leq \mathbb{E} \left[S_t^2 | \mathcal{F}_{t-1} \right] \\ &= \mathbb{E} \left[\max_j (V_{t,j} K_{t,j})^2 | \mathcal{F}_{t-1} \right] \leq \mathbb{E} \left[\sum_{j=1}^d V_{t,j} K_{t,j}^2 | \mathcal{F}_{t-1} \right] \\ &\leq \sum_{j=1}^d \min \left\{ \frac{2}{q_{t,j}}, M \right\} \leq dM, \end{aligned}$$

where we used $\mathbb{E}[K_{t,i}^2 | \mathcal{F}_{t-1}] = \frac{2-q_{t,i}}{q_{t,i}}$. Then, the second statement follows from applying a version of Freedman's inequality due to Beygelzimer et al. (2011) (stated as Lemma 16 in the appendix) with $B = M$ and $\Sigma_T \leq dMT$. \blacksquare

Notice that choosing $M = O(\sqrt{dT})$ as suggested by Theorems 1 and 2, the above result guarantees that the amortized running time of FPL+GR is $O((d + \sqrt{dT}) \cdot (f(S) + d))$ with high probability.

4. Analysis

This section presents the proofs of Theorems 1 and 2. In a didactic attempt, we present statements concerning the loss-estimation procedure and the learning algorithm separately: Section 4.1 presents various important properties of the loss estimates produced by Geometric Resampling, Section 4.2 presents general tools for analyzing Follow-the-Perturbed-Leader methods. Finally, Sections 4.3 and 4.4 put these results together to prove Theorems 1 and 2, respectively.

4.1 Properties of Geometric Resampling

The basic idea underlying Geometric Resampling is replacing the importance weights $1/q_{t,i}$ by appropriately defined random variables $K_{t,i}$. As we have seen earlier (Section 2), running GR with $M = \infty$ amounts to sampling each $K_{t,i}$ from a geometric distribution with expectation $1/q_{t,i}$, yielding an unbiased loss estimate. In practice, one would want to set M to a finite value to ensure that the running time of the sampling procedure is bounded. Note however that early termination of GR introduces a bias in the loss estimates. This section is mainly concerned with the nature of this bias. We emphasize that the statements presented in this section remain valid no matter what randomized algorithm generates the actions \tilde{V}_t . Our first lemma gives an explicit expression on the expectation of the loss estimates generated by GR .

Lemma 4 For all j and t such that $q_{t,j} > 0$, the loss estimates (5) satisfy

$$\mathbb{E} \left[\hat{\ell}_{t,j} | \mathcal{F}_{t-1} \right] = (1 - (1 - q_{t,j})^M) \ell_{t,j}.$$

Proof Fix any j, t satisfying the condition of the lemma. Setting $q = q_{t,j}$ for simplicity, we write

$$\begin{aligned} \mathbb{E}[K_{t,j} | \mathcal{F}_{t-1}] &= \sum_{k=1}^{\infty} k(1-q)^{k-1} q - \sum_{k=M}^{\infty} (k-M)(1-q)^{k-1} q \\ &= \sum_{k=1}^{\infty} k(1-q)^{k-1} q - (1-q)^M \sum_{k=M}^{\infty} (k-M)(1-q)^{k-M-1} q \\ &= (1 - (1-q)^M) \sum_{k=1}^{\infty} k(1-q)^{k-1} q = \frac{1 - (1-q)^M}{q}. \end{aligned}$$

The proof is concluded by combining the above with $\mathbb{E} \left[\widehat{\ell}_{k,j} \middle| \mathcal{F}_{t-1} \right] = q_{t,j} \ell_{t,j} \mathbb{E} [K_{t,j} \middle| \mathcal{F}_{t-1}]$. \blacksquare

The following lemma shows two important properties of the GR loss estimates (5). Roughly speaking, the first of these properties ensure that any learning algorithm relying on these estimates will be *optimistic* in the sense that the loss of any fixed decision will be underestimated in expectation. The second property ensures that the learner will not be *overly optimistic* concerning its own performance.

Lemma 5 For all $\mathbf{v} \in \mathcal{S}$ and t , the loss estimates (5) satisfy the following two properties:

$$\mathbb{E} \left[\mathbf{v}^\top \widehat{\ell}_t \middle| \mathcal{F}_{t-1} \right] \leq \mathbf{v}^\top \ell_t, \quad (8)$$

$$\mathbb{E} \left[\sum_{\mathbf{u} \in \mathcal{S}} p_t(\mathbf{u}) \left(\mathbf{u}^\top \widehat{\ell}_t \right) \middle| \mathcal{F}_{t-1} \right] \geq \sum_{\mathbf{u} \in \mathcal{S}} p_t(\mathbf{u}) \left(\mathbf{u}^\top \ell_t \right) - \frac{d}{cM}. \quad (9)$$

Proof Fix any $\mathbf{v} \in \mathcal{S}$ and t . The first property is an immediate consequence of Lemma 4: we have that $\mathbb{E} \left[\widehat{\ell}_{t,k} \middle| \mathcal{F}_{t-1} \right] \leq \ell_{t,k}$ for all k , and thus $\mathbb{E} \left[\mathbf{v}^\top \widehat{\ell}_t \middle| \mathcal{F}_{t-1} \right] \leq \mathbf{v}^\top \ell_t$. For the second statement, observe that

$$\mathbb{E} \left[\sum_{\mathbf{u} \in \mathcal{S}} p_t(\mathbf{u}) \left(\mathbf{u}^\top \widehat{\ell}_t \right) \middle| \mathcal{F}_{t-1} \right] = \sum_{i=1}^d q_{t,i} \mathbb{E} \left[\widehat{\ell}_{t,i} \middle| \mathcal{F}_{t-1} \right] = \sum_{i=1}^d q_{t,i} \left(1 - (1 - q_{t,i})^M \right) \ell_{t,i}$$

also holds by Lemma 4. To control the bias term $\sum_i q_{t,i} (1 - q_{t,i})^M$, note that $q_{t,i} (1 - q_{t,i})^M \leq q_{t,i} e^{-Mq_{t,i}}$. By elementary calculations, one can show that $f(q) = qe^{-Mq}$ takes its maximum at $q = \frac{1}{M}$ and thus $\sum_{i=1}^d q_{t,i} (1 - q_{t,i})^M \leq \frac{d}{cM}$. \blacksquare

Our last lemma concerning the loss estimates (5) bounds the conditional variance of the estimated loss of the learner. This term plays a key role in the performance analysis of Exp3-style algorithms (see, e.g., Auer et al. (2002); Uchiya et al. (2010); Audibert et al. (2014)), as well as in the analysis presented in the current paper:

Lemma 6 For all t , the loss estimates (5) satisfy

$$\mathbb{E} \left[\sum_{\mathbf{u} \in \mathcal{S}} p_t(\mathbf{u}) \left(\mathbf{u}^\top \widehat{\ell}_t \right)^2 \middle| \mathcal{F}_{t-1} \right] \leq 2md.$$

Before proving the statement, we remark that the conditional variance can be bounded as md for the standard (although usually infeasible) loss estimates (4). That is, the above lemma shows that, somewhat surprisingly, the variance of our estimates is only twice as large as the variance of the standard estimates.

Proof Fix an arbitrary t . For simplifying notation below, let us introduce $\widetilde{\mathbf{V}}$ as an independent copy of \mathbf{V}_t such that $\mathbb{P} \left[\widetilde{\mathbf{V}} = \mathbf{v} \middle| \mathcal{F}_{t-1} \right] = p_t(\mathbf{v})$ holds for all $\mathbf{v} \in \mathcal{S}$. To begin, observe that for any i

$$\mathbb{E} \left[K_{t,i}^2 \middle| \mathcal{F}_{t-1} \right] \leq \frac{2 - q_{t,i}}{q_{t,i}^2} \leq \frac{2}{q_{t,i}^2} \quad (10)$$

holds, as $K_{t,i}$ has a truncated geometric law. The statement is proven as

$$\begin{aligned} & \mathbb{E} \left[\sum_{\mathbf{u} \in \mathcal{S}} p_t(\mathbf{u}) \left(\mathbf{u}^\top \widehat{\ell}_t \right)^2 \middle| \mathcal{F}_{t-1} \right] = \mathbb{E} \left[\sum_{i=1}^d \sum_{j=1}^d \left(\widetilde{V}_i \widehat{\ell}_{t,i} \right) \left(\widetilde{V}_j \widehat{\ell}_{t,j} \right) \middle| \mathcal{F}_{t-1} \right] \\ &= \mathbb{E} \left[\sum_{i=1}^d \sum_{j=1}^d \left(\widetilde{V}_i K_{t,i} V_{t,i} \ell_{t,i} \right) \left(\widetilde{V}_j K_{t,j} V_{t,j} \ell_{t,j} \right) \middle| \mathcal{F}_{t-1} \right] \\ & \quad \text{(using the definition of } \widehat{\ell}_t \text{)} \\ & \leq \mathbb{E} \left[\sum_{i=1}^d \sum_{j=1}^d \frac{K_{t,i}^2 + K_{t,j}^2}{2} \left(\widetilde{V}_i V_{t,i} \ell_{t,i} \right) \left(\widetilde{V}_j V_{t,j} \ell_{t,j} \right) \middle| \mathcal{F}_{t-1} \right] \\ & \quad \text{(using } 2AB \leq A^2 + B^2 \text{)} \\ & \leq 2\mathbb{E} \left[\sum_{i=1}^d \frac{1}{q_{t,i}^2} \left(\widetilde{V}_i V_{t,i} \ell_{t,i} \right) \sum_{j=1}^d V_{t,j} \ell_{t,j} \middle| \mathcal{F}_{t-1} \right] \\ & \quad \text{(using symmetry, Eq. (10) and } \widetilde{V}_j \leq 1 \text{)} \\ & \leq 2m\mathbb{E} \left[\sum_{j=1}^d \ell_{t,j} \middle| \mathcal{F}_{t-1} \right] \leq 2md, \end{aligned}$$

where the last line follows from using $\|\mathbf{V}_t\|_1 \leq m$, $\|\ell_t\|_\infty \leq 1$, and $\mathbb{E} [V_{t,i} \middle| \mathcal{F}_{t-1}] = \mathbb{E} [\widetilde{V}_i \middle| \mathcal{F}_{t-1}] = q_{t,i}$. \blacksquare

4.2 General Tools for Analyzing FPL

In this section, we present the key tools for analyzing the FPL-component of our learning algorithm. In some respect, our analysis is a synthesis of previous work on FPL-style methods: we borrow several ideas from Poland (2005) and the proof of Corollary 4.5 in Cesa-Bianchi and Lugosi (2006). Nevertheless, our analysis is the first to directly target combinatorial settings and yields the tightest known bounds for FPL in this domain. Indeed, the tools developed in this section also permit an improvement for FPL in the full-information setting, closing the presumed performance gap between FPL and EMA in both the full-information and the semi-bandit settings. The statements we present in this section are not specific to the loss-estimate vectors used by FPL+GR.

Like most other known work, we study the performance of the learning algorithm through a *virtual algorithm* that (i) uses a time-independent perturbation vector and (ii) is allowed to peek one step into the future. Specifically, letting $\widetilde{\mathbf{Z}}$ be a perturbation vector drawn independently from the same distribution as \mathbf{Z}_t , the virtual algorithm picks its t^{th} action as

$$\widetilde{\mathbf{V}}_t = \arg \min_{\mathbf{v} \in \mathcal{S}} \left\{ \mathbf{v}^\top \left(\eta \widetilde{\mathbf{L}}_t - \widetilde{\mathbf{Z}} \right) \right\}. \quad (11)$$

In what follows, we will crucially use that $\widetilde{\mathbf{V}}_t$ and \mathbf{V}_{t+1} are conditionally independent and identically distributed given \mathcal{F}_t . In particular, introducing the notations

$$\begin{aligned} q_{t,i} &= \mathbb{E} [V_{t,i} \middle| \mathcal{F}_{t-1}] & \widetilde{q}_{t,i} &= \mathbb{E} \left[\widetilde{V}_{t,i} \middle| \mathcal{F}_t \right] \\ p_t(\mathbf{v}) &= \mathbb{P} [V_t = \mathbf{v} \middle| \mathcal{F}_{t-1}] & \widetilde{p}_t(\mathbf{v}) &= \mathbb{P} [\widetilde{V}_t = \mathbf{v} \middle| \mathcal{F}_t], \end{aligned}$$

we will exploit the above property by using $q_{t,i} = \tilde{q}_{t-1,i}$ and $p_t(\mathbf{v}) = \tilde{p}_{t-1}(\mathbf{v})$ numerous times in the proofs below.

First, we show a regret bound on the virtual algorithm that plays the action sequence $\tilde{\mathbf{V}}_1, \tilde{\mathbf{V}}_2, \dots, \tilde{\mathbf{V}}_T$.

Lemma 7 For any $\mathbf{v} \in \mathcal{S}$,

$$\sum_{t=1}^T \sum_{\mathbf{u} \in \mathcal{S}} \tilde{p}_t(\mathbf{u}) \left((\mathbf{u} - \mathbf{v})^\top \hat{\boldsymbol{\ell}}_t \right) \leq \frac{m(\log(d/m) + 1)}{\eta}. \quad (12)$$

Although the proof of this statement is rather standard, we include it for completeness. We also note that the lemma slightly improves other known results by replacing the usual $\log d$ term by $\log(d/m)$.

Proof Fix any $\mathbf{v} \in \mathcal{S}$. Using Lemma 3.1 of Cesa-Bianchi and Lugosi (2006) (sometimes referred to as the “follow-the-leader/be-the-leader” lemma) for the sequence $(\eta \hat{\boldsymbol{\ell}}_1 - \tilde{\mathbf{Z}}, \eta \hat{\boldsymbol{\ell}}_2, \dots, \eta \hat{\boldsymbol{\ell}}_T)$, we obtain

$$\sum_{t=1}^T \tilde{\mathbf{V}}_t^\top \hat{\boldsymbol{\ell}}_t - \tilde{\mathbf{V}}_1^\top \tilde{\mathbf{Z}} \leq \eta \sum_{t=1}^T \mathbf{v}^\top \hat{\boldsymbol{\ell}}_t - \mathbf{v}^\top \tilde{\mathbf{Z}}.$$

Reordering and integrating both sides with respect to the distribution of $\tilde{\mathbf{Z}}$ gives

$$\sum_{t=1}^T \sum_{\mathbf{u} \in \mathcal{S}} \tilde{p}_t(\mathbf{u}) \left((\mathbf{u} - \mathbf{v})^\top \hat{\boldsymbol{\ell}}_t \right) \leq \mathbb{E} \left[\left(\tilde{\mathbf{V}}_1 - \mathbf{v} \right)^\top \tilde{\mathbf{Z}} \right]. \quad (13)$$

The statement follows from using $\mathbb{E} \left[\tilde{\mathbf{V}}_1^\top \tilde{\mathbf{Z}} \right] \leq m(\log(d/m) + 1)$, which is proven in Appendix A as Lemma 14, noting that $\tilde{\mathbf{V}}_1^\top \tilde{\mathbf{Z}}$ is upper-bounded by the sum of the m largest components of $\tilde{\mathbf{Z}}$. ■

The next lemma relates the performance of the virtual algorithm to the actual performance of FPL. The lemma relies on a “sparse-loss” trick similar to the trick used in the proof Corollary 4.5 in Cesa-Bianchi and Lugosi (2006), and is also related to the “unit rule” discussed by Koolen et al. (2010).

Lemma 8 For all $t = 1, 2, \dots, T$, assume that $\hat{\boldsymbol{\ell}}_{t,k} \geq 0$ for all $k \in \{1, 2, \dots, d\}$. Then,

$$\sum_{\mathbf{u} \in \mathcal{S}} (p_t(\mathbf{u}) - \tilde{p}_t(\mathbf{u})) \left(\mathbf{u}^\top \hat{\boldsymbol{\ell}}_t \right) \leq \eta \sum_{\mathbf{u} \in \mathcal{S}} p_t(\mathbf{u}) \left(\mathbf{u}^\top \hat{\boldsymbol{\ell}}_t \right)^2.$$

Proof Fix an arbitrary t and $\mathbf{u} \in \mathcal{S}$, and define the “sparse loss vector” $\hat{\boldsymbol{\ell}}_t^-(\mathbf{u})$ with components $\hat{\ell}_{t,k}^-(\mathbf{u}) = u_k \hat{\ell}_{t,k}$ and

$$\mathbf{V}_t^-(\mathbf{u}) = \arg \min_{\mathbf{v} \in \mathcal{S}} \left\{ \mathbf{v}^\top \left(\eta \hat{\boldsymbol{\ell}}_{t-1} + \eta \hat{\boldsymbol{\ell}}_t^-(\mathbf{u}) - \tilde{\mathbf{Z}} \right) \right\}.$$

Using the notation $p_t^-(\mathbf{u}) = \mathbb{P} \left[\mathbf{V}_t^-(\mathbf{u}) = \mathbf{u} \mid \mathcal{F}_t \right]$, we show in Lemma 15 (stated and proved in Appendix A) that $p_t^-(\mathbf{u}) \leq \tilde{p}_t(\mathbf{u})$. Also, define

$$\mathbf{U}(\mathbf{z}) = \arg \min_{\mathbf{v} \in \mathcal{S}} \left\{ \mathbf{v}^\top \left(\eta \hat{\boldsymbol{\ell}}_{t-1} - \mathbf{z} \right) \right\}.$$

Letting $f(\mathbf{z}) = e^{-\|\mathbf{z}\|_1}$, ($\mathbf{z} \in \mathbb{R}_+^d$) be the density of the perturbations, we have

$$\begin{aligned} p_t(\mathbf{u}) &= \int_{\mathbf{z} \in [0, \infty]^d} \mathbb{I}_{\{\mathbf{U}(\mathbf{z}) = \mathbf{u}\}} f(\mathbf{z}) \, d\mathbf{z} \\ &= e^{\eta \|\hat{\boldsymbol{\ell}}_t^-(\mathbf{u})\|_1} \int_{\mathbf{z} \in [0, \infty]^d} \mathbb{I}_{\{\mathbf{U}(\mathbf{z}) = \mathbf{u}\}} f(\mathbf{z} + \eta \hat{\boldsymbol{\ell}}_t^-(\mathbf{u})) \, d\mathbf{z} \\ &= e^{\eta \|\hat{\boldsymbol{\ell}}_t^-(\mathbf{u})\|_1} \int \dots \int_{\mathbf{z}_t \in (\hat{\ell}_{t,1}^-(\mathbf{u}), \infty)} \mathbb{I}_{\{\mathbf{U}(\mathbf{z} - \eta \hat{\boldsymbol{\ell}}_t^-(\mathbf{u})) = \mathbf{u}\}} f(\mathbf{z}) \, d\mathbf{z} \\ &\leq e^{\eta \|\hat{\boldsymbol{\ell}}_t^-(\mathbf{u})\|_1} \int_{\mathbf{z} \in [0, \infty]^d} \mathbb{I}_{\{\mathbf{U}(\mathbf{z} - \eta \hat{\boldsymbol{\ell}}_t^-(\mathbf{u})) = \mathbf{u}\}} f(\mathbf{z}) \, d\mathbf{z} \\ &\leq e^{\eta \|\hat{\boldsymbol{\ell}}_t^-(\mathbf{u})\|_1} p_t^-(\mathbf{u}) \leq e^{\eta \|\hat{\boldsymbol{\ell}}_t^-(\mathbf{u})\|_1} \tilde{p}_t(\mathbf{u}). \end{aligned}$$

Now notice that $\|\hat{\boldsymbol{\ell}}_t^-(\mathbf{u})\|_1 = \mathbf{u}^\top \hat{\boldsymbol{\ell}}_t^-(\mathbf{u}) = \mathbf{u}^\top \hat{\boldsymbol{\ell}}_t$, which gives

$$\tilde{p}_t(\mathbf{u}) \geq p_t(\mathbf{u}) e^{-\eta \mathbf{u}^\top \hat{\boldsymbol{\ell}}_t} \geq p_t(\mathbf{u}) \left(1 - \eta \mathbf{u}^\top \hat{\boldsymbol{\ell}}_t \right).$$

The proof is concluded by repeating the same argument for all $\mathbf{u} \in \mathcal{S}$, reordering and summing the terms as

$$\sum_{\mathbf{u} \in \mathcal{S}} p_t(\mathbf{u}) \left(\mathbf{u}^\top \hat{\boldsymbol{\ell}}_t \right) \leq \sum_{\mathbf{u} \in \mathcal{S}} \tilde{p}_t(\mathbf{u}) \left(\mathbf{u}^\top \hat{\boldsymbol{\ell}}_t \right) + \eta \sum_{\mathbf{u} \in \mathcal{S}} p_t(\mathbf{u}) \left(\mathbf{u}^\top \hat{\boldsymbol{\ell}}_t \right)^2. \quad (14)$$

4.3 Proof of Theorem 1

Now, everything is ready to prove the bound on the expected regret of FPL+GR. Let us fix an arbitrary $\mathbf{v} \in \mathcal{S}$. By putting together Lemmas 6, 7 and 8, we immediately obtain

$$\mathbb{E} \left[\sum_{t=1}^T \sum_{\mathbf{u} \in \mathcal{S}} p_t(\mathbf{u}) \left((\mathbf{u} - \mathbf{v})^\top \hat{\boldsymbol{\ell}}_t \right) \right] \leq \frac{m(\log(d/m) + 1)}{\eta} + 2\eta m d T, \quad (15)$$

leaving us with the problem of upper bounding the expected regret in terms of the left-hand side of the above inequality. This can be done by using the properties of the loss estimates (5) stated in Lemma 5:

$$\mathbb{E} \left[\sum_{t=1}^T (\mathbf{V}_t - \mathbf{v})^\top \boldsymbol{\ell}_t \right] \leq \mathbb{E} \left[\sum_{t=1}^T \sum_{\mathbf{u} \in \mathcal{S}} p_t(\mathbf{u}) \left((\mathbf{u} - \mathbf{v})^\top \hat{\boldsymbol{\ell}}_t \right) \right] + \frac{dT}{\epsilon M}.$$

Putting the two inequalities together proves the theorem.

4.4 Proof of Theorem 2

We now turn to prove a bound on the regret of FPL+GR. P that holds with high probability. We begin by noting that the conditions of Lemmas 7 and 8 continue to hold for the new loss estimates, so we can obtain the central terms in the regret:

$$\sum_{t=1}^T \sum_{\mathbf{u} \in \mathcal{S}} p_t(\mathbf{u}) \left((\mathbf{u} - \mathbf{v})^\top \tilde{\boldsymbol{\ell}}_t \right) \leq \frac{m(\log(d/m) + 1)}{\eta} + \eta \sum_{t=1}^T \sum_{\mathbf{u} \in \mathcal{S}} p_t(\mathbf{u}) \left(\mathbf{u}^\top \tilde{\boldsymbol{\ell}}_t \right)^2.$$

The first challenge posed by the above expression is relating the left-hand side to the true regret with high probability. Once this is done, the remaining challenge is to bound the second term on the right-hand side, as well as the other terms arising from the first step. We first show that the loss estimates used by FPL+GR, \mathbf{P} consistently underestimate the true losses with high probability.

Lemma 9 Fix any $\delta' > 0$. For any $\mathbf{v} \in S$,

$$v^* (\tilde{\mathbf{L}}_T - \mathbf{L}_T) \leq \frac{m \log(d/\delta')}{\beta}$$

holds with probability at least $1 - \delta'$.

The simple proof is directly inspired by Appendix C.9 of Audibert and Bubeck (2010).

Proof Fix any t and i . Then,

$$\mathbb{E} \left[\exp(\beta \tilde{\ell}_{t,i}) \middle| \mathcal{F}_{t-1} \right] = \mathbb{E} \left[\exp \left(\log \left(1 + \beta \tilde{\ell}_{t,i} \right) \right) \middle| \mathcal{F}_{t-1} \right] \leq 1 + \beta \ell_{t,i} \leq \exp(\beta \ell_{t,i}),$$

where we used Lemma 4 in the first inequality and $1 + z \leq e^z$ that holds for all $z \in \mathbb{R}$. As a result, the process $W_t = \exp(\beta(\tilde{\mathbf{L}}_{t,i} - L_{t,i}))$ is a supermartingale with respect to (\mathcal{F}_t) : $\mathbb{E}[W_t | \mathcal{F}_{t-1}] \leq W_{t-1}$. Observe that, since $W_0 = 1$, this implies $\mathbb{E}[W_t] \leq \mathbb{E}[W_{t-1}] \leq \dots \leq 1$. Applying Markov's inequality gives that

$$\begin{aligned} \mathbb{P}[\tilde{L}_{T,i} > L_{T,i} + \epsilon] &= \mathbb{P}[\tilde{L}_{T,i} - L_{T,i} > \epsilon] \\ &\leq \mathbb{E} \left[\exp \left(\beta (\tilde{L}_{T,i} - L_{T,i}) \right) \right] \exp(-\beta \epsilon) \leq \exp(-\beta \epsilon) \end{aligned}$$

holds for any $\epsilon > 0$. The statement of the lemma follows after using $\|\mathbf{v}\|_1 \leq m$, applying the union bound for all i , and solving for ϵ . \blacksquare

The following lemma states another key property of the loss estimates.

Lemma 10 For any t ,

$$\sum_{i=1}^d q_{t,i} \hat{\ell}_{t,i} \leq \sum_{i=1}^d q_{t,i} \tilde{\ell}_{t,i} + \frac{\beta}{2} \sum_{i=1}^d q_{t,i} \tilde{\ell}_{t,i}^2.$$

Proof The statement follows trivially from the inequality $\log(1+z) \geq z - \frac{z^2}{2}$ that holds for all $z \geq 0$. In particular, for any fixed t and i , we have

$$\log \left(1 + \beta \tilde{\ell}_{t,i} \right) \geq \beta \tilde{\ell}_{t,i} - \frac{\beta^2}{2} \tilde{\ell}_{t,i}^2.$$

Multiplying both sides by $q_{t,i}/\beta$ and summing for all i proves the statement. \blacksquare

The next lemma relates the total loss of the learner to its total estimated losses.

Lemma 11 Fix any $\delta' > 0$. With probability at least $1 - 2\delta'$,

$$\sum_{i=1}^T V_t^r \ell_i \leq \sum_{i=1}^T \sum_{\mathbf{w} \in S} p_i(\mathbf{w}) \left(\mathbf{w}^T \hat{\ell}_i \right) + \frac{dT}{cM} + \sqrt{2(e-2)T} \left(m \log \frac{1}{\delta'} + 1 \right) + \sqrt{87T \log \frac{1}{\delta'}}$$

Proof We start by rewriting

$$\sum_{\mathbf{w} \in S} p_i(\mathbf{w}) \left(\mathbf{w}^T \hat{\ell}_i \right) = \sum_{i=1}^d q_{t,i} K_{t,i} V_{t,i} \ell_{t,i}.$$

Now let $k_{t,i} = \mathbb{E}[K_{t,i} | \mathcal{F}_{t-1}]$ for all i and notice that

$$X_t = \sum_{i=1}^d q_{t,i} V_{t,i} \ell_{t,i} (k_{t,i} - K_{t,i})$$

is a martingale-difference sequence with respect to (\mathcal{F}_t) with elements upper-bounded by m (as Lemma 4 implies $k_{t,i}, q_{t,i} \leq 1$ and $\|V_t\|_1 \leq m$). Furthermore, the conditional variance of the increments is bounded as

$$\text{Var}[X_t | \mathcal{F}_{t-1}] \leq \mathbb{E} \left[\left(\sum_{i=1}^d q_{t,i} V_{t,i} K_{t,i} \right)^2 \middle| \mathcal{F}_{t-1} \right] \leq \mathbb{E} \left[\sum_{j=1}^d V_{t,j} \left(\sum_{i=1}^d q_{t,i}^2 K_{t,i}^2 \right) \middle| \mathcal{F}_{t-1} \right] \leq 2m,$$

where the second inequality is Cauchy-Schwarz and the third one follows from $\mathbb{E}[K_{t,i}^2 | \mathcal{F}_{t-1}] \leq 2/q_{t,i}^2$ and $\|V_t\|_1 \leq m$. Thus, applying Lemma 16 with $B = m$ and $\Sigma_T \leq 2mT$ we get that for any $S \geq m \sqrt{\log \frac{1}{\delta'}} / (e-2)$,

$$\sum_{t=1}^T \sum_{i=1}^d q_{t,i} \ell_{t,i} V_{t,i} (k_{t,i} - K_{t,i}) \leq \sqrt{(e-2) \log \frac{1}{\delta'}} \left(\frac{2mT}{S} + S \right)$$

holds with probability at least $1 - \delta'$, where we have used $\|V_t\|_1 \leq m$. After setting $S = m \sqrt{2T \log \frac{1}{\delta'}}$, we obtain that

$$\sum_{t=1}^T \sum_{i=1}^d q_{t,i} \ell_{t,i} V_{t,i} (k_{t,i} - K_{t,i}) \leq \sqrt{2(e-2)T} \left(m \log \frac{1}{\delta'} + 1 \right) \quad (16)$$

holds with probability at least $1 - \delta'$.

To proceed, observe that $q_{t,i} k_{t,i} = 1 - (1 - q_{t,i})^M$ holds by Lemma 4, implying

$$\sum_{i=1}^d q_{t,i} V_{t,i} \ell_{t,i} k_{t,i} \geq V_t^T \ell_t - \sum_{i=1}^d V_{t,i} (1 - q_{t,i})^M.$$

Together with Eq. (16), this gives

$$\sum_{i=1}^T V_t^r \ell_i \leq \sum_{i=1}^T \sum_{\mathbf{w} \in S} p_i(\mathbf{w}) \left(\mathbf{w}^T \hat{\ell}_i \right) + \sqrt{2(e-2)T} \left(m \log \frac{1}{\delta'} + 1 \right) + \sum_{t=1}^T \sum_{i=1}^d V_{t,i} (1 - q_{t,i})^M.$$

Finally, we use that, by Lemma 5, $(1 - q_{t,i})^M \leq 1/(eM)$, and

$$Y_t = \sum_{i=1}^d (V_{t,i} - q_{t,i}) (1 - q_{t,i})^M$$

is a martingale-difference sequence with respect to (\mathcal{F}_t) with increments bounded in $[-1, 1]$. Then, by an application of Hoeffding–Azuma inequality, we have

$$\sum_{i=1}^T \sum_{i=1}^d V_{t,i} (1 - q_{t,i})^M \leq \frac{dT}{cM} + \sqrt{87T \log \frac{1}{\delta'}}$$

with probability at least $1 - \delta'$, thus proving the lemma. \blacksquare

Finally, our last lemma in this section bounds the second-order terms arising from Lemmas 8 and 10.

Lemma 12 Fix any $\delta' > 0$. With probability at least $1 - 2\delta'$, the following hold simultaneously:

$$\begin{aligned} \sum_{t=1}^T \sum_{\mathbf{v} \in \mathcal{S}} p_t(\mathbf{v}) \left(\mathbf{v}^\top \hat{\boldsymbol{\ell}}_t \right)^2 &\leq Mm \sqrt{2T \log \frac{1}{\delta'}} + 2md \sqrt{T \log \frac{1}{\delta'}} + 2mdT \\ \sum_{t=1}^T \sum_{i=1}^d q_{t,i} \hat{\rho}_{t,i}^2 &\leq M \sqrt{2mT \log \frac{1}{\delta'}} + 2d \sqrt{T \log \frac{1}{\delta'}} + 2dT. \end{aligned}$$

Proof First, recall that

$$\mathbb{E} \left[\sum_{\mathbf{v} \in \mathcal{S}} p_t(\mathbf{v}) \left(\mathbf{v}^\top \hat{\boldsymbol{\ell}}_t \right)^2 \middle| \mathcal{F}_{t-1} \right] \leq 2md$$

holds by Lemma 8. Now, observe that

$$X_t = \sum_{\mathbf{v} \in \mathcal{S}} p_t(\mathbf{v}) \left(\left(\mathbf{v}^\top \hat{\boldsymbol{\ell}}_t \right)^2 - \mathbb{E} \left[\left(\mathbf{v}^\top \hat{\boldsymbol{\ell}}_t \right)^2 \middle| \mathcal{F}_{t-1} \right] \right)$$

is a martingale-difference sequence with increments in $[-2md, mM]$. An application of the Hoeffding-Azuma inequality gives that

$$\sum_{t=1}^T \sum_{\mathbf{v} \in \mathcal{S}} p_t(\mathbf{v}) \left(\left(\mathbf{v}^\top \hat{\boldsymbol{\ell}}_t \right)^2 - \mathbb{E} \left[\left(\mathbf{v}^\top \hat{\boldsymbol{\ell}}_t \right)^2 \middle| \mathcal{F}_{t-1} \right] \right) \leq Mm \sqrt{2T \log \frac{1}{\delta'}} + 2md \sqrt{T \log \frac{1}{\delta'}}$$

holds with probability at least $1 - \delta'$. Reordering the terms completes the proof of the first statement. The second statement is proven analogously, building on the bound

$$\mathbb{E} \left[\sum_{t=1}^d \sum_{i=1}^d q_{t,i} \hat{\rho}_{t,i}^2 \middle| \mathcal{F}_{t-1} \right] \leq \mathbb{E} \left[\sum_{t=1}^d \sum_{i=1}^d q_{t,i} V_{t,i} K_{t,i}^2 \middle| \mathcal{F}_{t-1} \right] \leq 2d.$$

Theorem 2 follows from combining Lemmas 9 through 12 and applying the union bound. \blacksquare

5. Improved Bounds for Learning With Full Information

Our proof techniques presented in Section 4.2 also enable us to tighten the guarantees for FPL in the full information setting. In particular, consider the algorithm choosing action

$$\mathbf{V}_t = \arg \min_{\mathbf{v} \in \mathcal{S}} \langle \eta \mathbf{L}_{t-1} - \mathbf{Z}_t, \mathbf{v} \rangle,$$

where $\mathbf{L}_t = \sum_{s=1}^t \boldsymbol{\ell}_s$ and the components of \mathbf{Z}_t are drawn independently from a standard exponential distribution. We state our improved regret bounds concerning this algorithm in the following theorem.

Theorem 13 For any $\mathbf{v} \in \mathcal{S}$, the total expected regret of FPL satisfies

$$\hat{R}_T \leq \frac{m \left(\log(d/m) + 1 \right)}{\eta} + \eta m \sum_{t=1}^T \mathbb{E} [\mathbf{V}_t^\top \boldsymbol{\ell}_t]$$

under full information. In particular, defining $L_T^* = \min_{\mathbf{v} \in \mathcal{S}} \mathbf{v}^\top L_T$ and setting

$$\eta = \min \left\{ \sqrt{\frac{\log(d/m) + 1}{L_T^*}}, \frac{1}{2} \right\},$$

the regret of FPL satisfies

$$R_T \leq 4m \max \left\{ \sqrt{L_T^* \left(\log \left(\frac{d}{m} \right) + 1 \right)}, (m^2 + 1) \left(\log \frac{d}{m} + 1 \right) \right\}.$$

In the worst case, the above bound becomes $2m^{3/2} \sqrt{T \log(d/m) + 1}$, which improves the best known bound for FPL of Kalai and Vempala (2005) by a factor of $\sqrt{d/m}$.

Proof The first statement follows from combining Lemmas 7 and 8, and bounding

$$\sum_{\mathbf{u} \in \mathcal{S}} p_t(\mathbf{u}) \left(\mathbf{u}^\top \boldsymbol{\ell}_t \right)^2 \leq m \sum_{\mathbf{u} \in \mathcal{S}} p_t(\mathbf{u}) \left(\mathbf{u}^\top \boldsymbol{\ell}_t \right),$$

while the second one follows from standard algebraic manipulations. \blacksquare

6. Conclusions and Open Problems

In this paper, we have described the first *general and efficient* algorithm for online combinatorial optimization under semi-bandit feedback. We have proved that the regret of this algorithm is $O(m\sqrt{dT} \log(d/m))$ in this setting, and have also shown that FPL can achieve $O(m^{3/2} \sqrt{T \log(d/m)})$ in the full information case when tuned properly. While these bounds are off by a factor of $\sqrt{m \log(d/m)}$ and \sqrt{m} from the respective minimax results, they exactly match the best known regret bounds for the well-studied Exponentially Weighted Forecaster (EWA). Whether the remaining gaps can be closed for FPL-style algorithms (e.g., by using more intricate perturbation schemes or a more refined analysis) remains an important open question. Nevertheless, we regard our contribution as a significant step towards understanding the inherent trade-offs between computational efficiency and performance guarantees in online combinatorial optimization and, more generally, in online optimization.

The efficiency of our method rests on a novel loss estimation method called Geometric Resampling (GR). This estimation method is not specific to the proposed learning algorithm. While GR has no immediate benefits for OSMD-type algorithms where the ideal importance weights are readily available, it is possible to think about problem instances where EWA can be efficiently implemented while importance weights are difficult to compute.

The most important open problem left is the case of efficient online linear optimization with *full bandit feedback* where the learner only observes the inner product $\mathbf{V}_t^\top \boldsymbol{\ell}_t$ in round t . Learning algorithms for this problem usually require that the (pseudo-)inverse of the covariance matrix $P_t = \mathbb{E}[\mathbf{V}_t \mathbf{V}_t^\top | \mathcal{F}_{t-1}]$ is readily available for the learner at each time step (see, e.g., McMahan and Blum (2004); Dani et al. (2008); Cesa-Bianchi and Lugosi (2012); Buback et al. (2012)). Computing this matrix, however, is at least as challenging as computing the individual importance weights $1/q_{t,i}$. That said, our Geometric Resampling technique can be directly generalized to this setting by observing that the matrix geometric series $\sum_{n=1}^{\infty} (I - P_t)^n$ converges to P_t^{-1} under certain conditions. This sum can then be efficiently estimated by sampling independent copies of \mathbf{V}_t , which paves the path for constructing low-bias estimates of the loss vectors. While it seems straightforward to go ahead and use these estimates in tandem with FPL, we have to note that the analysis presented in this paper does not carry through directly in this case. The main limitation is that our techniques only apply for loss vectors with *non-negative* elements (cf. Lemma 8). Nevertheless, we believe that Geometric Resampling should be a crucial component for constructing truly effective learning algorithms for this important problem.

Acknowledgments

The authors wish to thank Csaba Szepesvári for thought-provoking discussions. The research presented in this paper was supported by the JPPFellows Fellowship (Marie Curie COFUND program n° 600387), the French Ministry of Higher Education and Research and by FUI project Hermès.

Appendix A. Further Proofs and Technical Tools

Lemma 14 Let Z_1, \dots, Z_d be i.i.d. exponentially distributed random variables with unit expectation and let Z_1^*, \dots, Z_d^* be their permutation such that $Z_1^* \geq Z_2^* \geq \dots \geq Z_d^*$. Then, for any $1 \leq m \leq d$,

$$\mathbb{E} \left[\sum_{i=1}^m Z_i^* \right] \leq m \left(\log \left(\frac{d}{m} \right) + 1 \right).$$

Proof Let us define $Y = \sum_{i=1}^m Z_i^*$. Then, as Y is nonnegative, we have for any $A \geq 0$ that

$$\begin{aligned} \mathbb{E}[Y] &= \int_0^\infty \mathbb{P}[Y > y] dy \\ &\leq A + \int_A^\infty \mathbb{P} \left[\sum_{i=1}^m Z_i^* > y \right] dy \\ &\leq A + \int_A^\infty \mathbb{P} \left[Z_1^* > \frac{y}{m} \right] dy \\ &\leq A + d \int_A^\infty \mathbb{P} \left[Z_1 > \frac{y}{m} \right] dy \\ &= A + de^{-A/m} \\ &\leq m \log \left(\frac{d}{m} \right) + m, \end{aligned}$$

where in the last step, we used that $A = \log \left(\frac{d}{m} \right)$ minimizes $A + de^{-A/m}$ over the real line. ■

Lemma 15 Fix any $\mathbf{v} \in S$ and any vectors $\mathbf{L} \in \mathbb{R}^d$ and $\boldsymbol{\ell} \in [0, \infty)^d$. Define the vector $\boldsymbol{\ell}'$ with components $\ell'_k = v_k \ell_k$. Then, for any perturbation vector \mathbf{Z} with independent components,

$$\begin{aligned} \mathbb{P}[\mathbf{v}^\top (\mathbf{L} + \boldsymbol{\ell}' - \mathbf{Z}) \leq \mathbf{u}^\top (\mathbf{L} + \boldsymbol{\ell}' - \mathbf{Z}) \mid \forall \mathbf{u} \in S] \\ \leq \mathbb{P}[\mathbf{v}^\top (\mathbf{L} + \boldsymbol{\ell}' - \mathbf{Z}) \leq \mathbf{u}^\top (\mathbf{L} + \boldsymbol{\ell}' - \mathbf{Z}) \mid \forall \mathbf{u} \in S]. \end{aligned}$$

Proof Fix any $\mathbf{u} \in S \setminus \{\mathbf{v}\}$ and define the vector $\boldsymbol{\ell}'' = \boldsymbol{\ell} - \boldsymbol{\ell}'$. Define the events

$$A(\mathbf{u}) = \{\mathbf{v}^\top (\mathbf{L} + \boldsymbol{\ell}' - \mathbf{Z}) \leq \mathbf{u}^\top (\mathbf{L} + \boldsymbol{\ell}' - \mathbf{Z})\}$$

and

$$A(\mathbf{u}) = \{\mathbf{v}^\top (\mathbf{L} + \boldsymbol{\ell}' - \mathbf{Z}) \leq \mathbf{u}^\top (\mathbf{L} + \boldsymbol{\ell}' - \mathbf{Z})\}.$$

We have

$$\begin{aligned} A(\mathbf{u}) &= \{(\mathbf{v} - \mathbf{u})^\top \mathbf{Z} \geq (\mathbf{v} - \mathbf{u})^\top (\mathbf{L} + \boldsymbol{\ell}')\} \\ &\subseteq \{(\mathbf{v} - \mathbf{u})^\top \mathbf{Z} \geq (\mathbf{v} - \mathbf{u})^\top (\mathbf{L} + \boldsymbol{\ell}') - \mathbf{u}^\top \boldsymbol{\ell}''\} \\ &= \{(\mathbf{v} - \mathbf{u})^\top \mathbf{Z} \geq (\mathbf{v} - \mathbf{u})^\top (\mathbf{L} + \boldsymbol{\ell}') - A(\mathbf{u})\}, \end{aligned}$$

where we used $\mathbf{v}^\top \boldsymbol{\ell}'' = 0$ and $\mathbf{u}^\top \boldsymbol{\ell}'' \geq 0$. Now, since $A(\mathbf{u}) \subseteq A(\mathbf{u})$, we have $\mathbb{P}_{\mathbf{u} \in S} A(\mathbf{u}) \subseteq \mathbb{P}_{\mathbf{u} \in S} A(\mathbf{u})$, thus proving $\mathbb{P}[\mathbb{P}_{\mathbf{u} \in S} A(\mathbf{u})] \leq \mathbb{P}[\mathbb{P}_{\mathbf{u} \in S} A(\mathbf{u})]$ as claimed in the lemma. ■

Lemma 16 (cf. Theorem 1 in Beygelzimer et al. (2011)) Assume X_1, X_2, \dots, X_T is a martingale-difference sequence with respect to the filtration (\mathcal{F}_t) with $X_t \leq B$ for $1 \leq t \leq T$. Let $\sigma_t^2 = \text{Var}[X_t \mid \mathcal{F}_{t-1}]$ and $\Sigma_t^2 = \sum_{s=1}^t \sigma_s^2$. Then, for any $\delta > 0$,

$$\mathbb{P} \left[\sum_{t=1}^T \mathbf{X}_t > B \log \frac{1}{\delta} + (e-2) \frac{\Sigma_T^2}{B} \right] \leq \delta.$$

Furthermore, for any $S > B \sqrt{\log(1/\delta)}(e-2)$,

$$\mathbb{P} \left[\sum_{t=1}^T \mathbf{X}_t > \sqrt{(e-2) \log \frac{1}{\delta}} \left(\frac{\Sigma_T^2}{S} + S \right) \right] \leq \delta.$$

References

- J. Abernethy, C. Lee, A. Sinha, and A. Tewari. Online linear optimization via smoothing. In *Proceedings of The 27th Conference on Learning Theory (COLT)*, pages 807–823, 2014.
- C. Allenberg, P. Auer, L. Györfi, and Gy. Ottucsák. Hannan consistency in on-line learning in case of unbounded losses under partial monitoring. In *Proceedings of the 17th International Conference on Algorithmic Learning Theory (ALT)*, pages 229–243, 2006.
- J.-Y. Audibert and S. Bubeck. Regret bounds and minimax policies under partial monitoring. *Journal of Machine Learning Research*, 11:2635–2686, 2010.
- J.-Y. Audibert, S. Bubeck, and G. Lugosi. Regret in online combinatorial optimization. *Mathematics of Operations Research*, 39:31–45, 2014.
- P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.
- B. Awerbuch and R. D. Kleinberg. Adaptive routing with end-to-end feedback: distributed learning and geometric approaches. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 45–53, 2004.
- A. Beygelzimer, J. Langford, L. Li, L. Reyzin, and R. E. Schapire. Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 19–26, 2011.
- S. Bubeck, N. Cesa-Bianchi, and S. M. Kakade. Towards minimax policies for online linear optimization with bandit feedback. In *Proceedings of The 25th Conference on Learning Theory (COLT)*, pages 1–14, 2012.
- S. Bubeck and N. Cesa-Bianchi. *Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems*. Now Publishers Inc, 2012.
- N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006.
- N. Cesa-Bianchi and G. Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78:1404–1422, 2012.

- V. Dani, T. Hayes, and S. Kakade. The price of bandit information for online optimization. In *Advances in Neural Information Processing Systems (NIPS)*, volume 20, pages 345–352, 2008.
- L. Devroye, G. Lugosi, and G. Neu. Prediction by random-walk perturbation. In *Proceedings of the 26th Conference on Learning Theory*, pages 460–473, 2013.
- Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55:119–139, 1997.
- A. Györfy, T. Linder, G. Lugosi, and Gy. Ottucsák. The on-line shortest path problem under partial monitoring. *Journal of Machine Learning Research*, 8:2369–2403, 2007.
- J. Hannan. Approximation to Bayes risk in repeated play. *Contributions to the Theory of Games*, 3:97–139, 1957.
- A. Kalai and S. Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71:291–307, 2005.
- W. Koolen, M. Warmuth, and J. Kivinen. Hedging structured concepts. In *Proceedings of the 23rd Conference on Learning Theory (COLT)*, pages 93–105, 2010.
- N. Littlestone and M. Warmuth. The weighted majority algorithm. *Information and Computation*, 108:212–261, 1994.
- H. B. McMahan and A. Blum. Online geometric optimization in the bandit setting against an adaptive adversary. In *Proceedings of the 17th Conference on Learning Theory (COLT)*, pages 109–123, 2004.
- G. Neu and G. Bartók. An efficient algorithm for learning with semi-bandit feedback. In *Proceedings of the 24th International Conference on Algorithmic Learning Theory (ALT)*, pages 234–248, 2013.
- J. Poland. FPL analysis for adaptive bandits. In *3rd Symposium on Stochastic Algorithms, Foundations and Applications (SAGA)*, pages 58–69, 2005.
- S. Rakhlin, O. Shamir, and K. Sridharan. Relax and randomize: From value to algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, volume 25, pages 2150–2158, 2012.
- D. Suehiro, K. Hatao, S. Kijima, E. Takimoto, and K. Nagano. Online prediction under submodular constraints. In *Proceedings of the 23rd International Conference on Algorithmic Learning Theory (ALT)*, pages 260–274, 2012.
- E. Takimoto and M. Warmuth. Paths kernels and multiplicative updates. *Journal of Machine Learning Research*, 4:773–818, 2003.
- T. Uchiya, A. Nakamura, and M. Kudo. Algorithms for adversarial bandit problems with multiple plays. In *Proceedings of the 21st International Conference on Algorithmic Learning Theory (ALT)*, pages 375–389, 2010.
- T. Van Erven, M. Warmuth, and W. Kotowski. Follow the leader with dropout perturbations. In *Proceedings of The 27th Conference on Learning Theory (COLT)*, pages 949–974, 2014.
- V. Vovk. Aggregating strategies. In *Proceedings of the 3rd Annual Workshop on Computational Learning Theory (COLT)*, pages 371–386, 1990.

New Perspectives on k -Support and Cluster Norms

Andrew M. McDonald

*Department of Computer Science
University College London
Gower Street, London WC1E 6BT, UK*

A.MCDONALD@CS.UCL.AC.UK

Massimiliano Pontil

*Istituto Italiano di Tecnologia
Via Morego, 30, 16163 Genoa, Italy
Department of Computer Science
University College London
Gower Street, London WC1E 6BT, UK*

M.PONTIL@CS.UCL.AC.UK

Dimitris Stamos

*Department of Computer Science
University College London
Gower Street, London WC1E 6BT, UK*

D.STAMOS@CS.UCL.AC.UK

Editor: Ingo Steinwart

Abstract

We study a regularizer which is defined as a parameterized infimum of quadratics, and which we call the box-norm. We show that the k -support norm, a regularizer proposed by Argyriou et al. (2012) for sparse vector prediction problems, belongs to this family, and the box-norm can be generated as a perturbation of the former. We derive an improved algorithm to compute the proximity operator of the squared box-norm, and we provide a method to compute the norm. We extend the norms to matrices, introducing the spectral k -support norm and spectral box-norm. We note that the spectral box-norm is essentially equivalent to the cluster norm, a multitask learning regularizer introduced by Jacob et al. (2009a), and which in turn can be interpreted as a perturbation of the spectral k -support norm. Centering the norm is important for multitask learning and we also provide a method to use centered versions of the norms as regularizers. Numerical experiments indicate that the spectral k -support and box-norms and their centered variants provide state of the art performance in matrix completion and multitask learning problems respectively.

Keywords: Convex optimization, matrix completion, multitask learning, spectral regularization, structured sparsity.

1. Introduction

We continue the study of a family of norms which are obtained by taking the infimum of a class of quadratic functions. These norms can be used as a regularizer in linear regression learning problems, where the parameter set can be tailored to assumptions on the underlying regression model. This family of norms is sufficiently rich to encompass regularizers such as the ℓ_p norms, the group Lasso with overlap (Jacob et al., 2009b) and the norm of Micchelli et al. (2013). In this paper we focus on a particular norm in this framework—the box-norm—

in which the parameter set involves box constraints and a linear constraint. We study the norm in detail and show that it can be generated as a perturbation of the k -support norm introduced by Argyriou et al. (2012) for sparse vector estimation, which hence can be seen as a special case of the box-norm. Furthermore, our variational framework allows us to study efficient algorithms to compute the norms and the proximity operator of the square of the norms.

Another main goal of this paper is to extend the k -support and box-norms to a matrix setting. We observe that both norms are symmetric gauge functions, hence by applying them to the spectrum of a matrix we obtain two orthogonally invariant matrix norms. In addition, we observe that the spectral box-norm is essentially equivalent to the cluster norm introduced by Jacob et al. (2009a) for multitask clustering, which in turn can be interpreted as a perturbation of the spectral k -support norm.

The characteristic properties of the vector norms translate in a natural manner to matrices. In particular, the unit ball of spectral k -support norm is the convex hull of the set of matrices of rank no greater than k , and Frobenius norm bounded by one. In numerical experiments we present empirical evidence on the strong performance of the spectral k -support norm in low rank matrix completion and multitask learning problems.

Moreover, our computation of the vector box-norm and its proximity operator extends naturally to the spectral case, which allows us to use proximal gradient methods to solve regularization problems using the cluster norm. Finally, we provide a method to use the centered versions of the penalties, which are important in applications (see e.g. Evgeniou et al., 2007; Jacob et al., 2009a).

1.1 Related Work

Our work builds upon a recent line of papers which considered convex regularizers defined as an infimum problem over a parametric family of quadratics, as well as related infimal convolution problems (see Jacob et al., 2009b; Bach et al., 2011; Maurer and Pontil, 2012; Micchelli and Pontil, 2005; Obozinski and Bach, 2012, and references therein). Related variational formulations for the Lasso have also been discussed in (Grandvalet, 1998) and further studied in (Szafranski et al., 2007).

To our knowledge, the box-norm was first suggested by Jacob et al. (2009a) and used as a symmetric gauge function in matrix learning problems. The induced orthogonally invariant matrix norm is named the *cluster norm* in (Jacob et al., 2009a) and was motivated as a convex relaxation of a multitask clustering problem. Here we formally prove that the cluster norm is indeed an orthogonal invariant norm. More importantly, we explicitly compute the norm and its proximity operator.

A key observation of this paper is the link between the box-norm and the k -support norm and in turn the link between the cluster norm and the spectral k -support norm. The k -support norm was proposed in (Argyriou et al., 2012) for sparse vector prediction and was shown to empirically outperform the Lasso (Tibshirani, 1996) and Elastic Net (Zou and Hastie, 2005) penalties. See also Gkirtzou et al. (2013) for further empirical results.

In recent years there has been a great deal of interest in the problem of learning a low rank matrix from a set of linear measurements. A widely studied and successful instance of this problem arises in the context of matrix completion or collaborative filtering, in which

we want to recover a low rank (or approximately low rank) matrix from a small sample of its entries, see e.g. Srebro et al. (2005); Abernethy et al. (2009) and references therein. One prominent method of solving this problem is trace norm regularization: we look for a matrix which closely fits the observed entries and has a small trace norm (sum of singular values) (Jaggi and Sulovsky, 2010; Toh and Yun, 2011; Mazumder et al., 2010). In our numerical experiments we consider the spectral k -support norm and spectral box-norm as alternatives to the trace norm and compare their performance.

Another application of matrix learning is multitask learning. In this framework a number of tasks, such as classifiers or regressors, are learned by taking advantage of commonalities between them. This can improve upon learning the tasks separately, for instance when insufficient data is available to solve each task in isolation (see e.g. Evgeniou et al., 2005; Argyriou et al., 2007, 2008; Jacob et al., 2009a; Cavallanti et al., 2010; Maurer, 2006; Maurer and Pontil, 2008). An approach which has been successful is the use of spectral regularizers such as the trace norm to learn a matrix where the columns represent the individual tasks, and in this paper we compare the performance of the spectral k -support and box-norms as penalties in multitask learning problems.

Finally, we note that this is a longer version of the conference paper (McDonald et al., 2014) and includes new theoretical and experimental results.

1.2 Contributions

We summarise the main contributions of this paper.

- We show that the vector k -support norm is a special case of the more general *box-norm*, which in turn can be seen as a perturbation of the former. The box-norm can be written as a parameterized infimum of quadratics, and this framework is instrumental in deriving a fast algorithm to compute the norm and the proximity operator of the squared norm in $\mathcal{O}(d \log d)$ time. Apart from improving on the $\mathcal{O}(d(k + \log d))$ algorithm for the proximity operator in Argyriou et al. (2012), this method allows one to use optimal first order optimization algorithms (Nesterov, 2007) for the box-norm.¹
- We extend the k -support and box-norms to orthogonally invariant matrix norms. We note that the spectral box-norm is essentially equivalent to the cluster norm, which in turn can be interpreted as a perturbation of the spectral k -support norm in the sense of the Moreau envelope. Our computation of the vector box-norm and its proximity operator also extends naturally to the spectral case. This allows us to use proximal gradient methods for the cluster norm. Furthermore, we provide a method to apply the centered versions of the penalties, which are important in applications.

- We present extensive numerical experiments on both synthetic and real matrix learning data sets. Our findings indicate that regularization with the spectral k -support and box-norms produces state-of-the-art results on a number of popular matrix completion benchmarks and centered variants of the norms show a significant improvement

¹ We note that recently Chatterjee et al. (2014) showed that the proximity operator of the vector k -support norm can be computed in $\mathcal{O}(d \log d)$. Here we directly follow Argyriou et al. (2012) and consider the squared k -support norm.

in performance over the centered trace norm and the matrix elastic net on multitask learning benchmarks.

1.3 Notation

We use \mathbb{N}_n for the set of integers from 1 up to and including n . We let \mathbb{R}^d be the d dimensional real vector space, whose elements are denoted by lower case letters. We let \mathbb{R}_+^d and \mathbb{R}_{++}^d be the subsets of vectors with nonnegative and strictly positive components, respectively. We denote by Δ^d the unit d -simplex, $\Delta^d = \{\lambda \in \mathbb{R}^{d+1} : \sum_{i=1}^{d+1} \lambda_i = 1\}$. For any vector $w \in \mathbb{R}^d$, its *support* is defined as $\text{supp}(w) = \{i : w_i \neq 0\} \subseteq \mathbb{N}_d$. We use 1 to denote either the scalar or a vector of all ones, whose dimension is determined by its context. Given a subset g of \mathbb{N}_d , the d -dimensional vector 1_g has ones on the support g , and zeros elsewhere. We let $\mathbb{R}^{d \times T}$ be the space of $d \times T$ real matrices and write $W = [w_1, \dots, w_T]$ to denote the matrix whose columns are formed by the vectors $w_1, \dots, w_T \in \mathbb{R}^d$. For a vector $\sigma \in \mathbb{R}^d$, we denote by $\text{diag}(\sigma)$ the $d \times d$ diagonal matrix having elements σ_i on the diagonal. We say matrix $W \in \mathbb{R}^{d \times T}$ is diagonal if $W_{ij} = 0$ whenever $i \neq j$. We denote the trace of a matrix W by $\text{tr}(W)$, and its rank by $\text{rank}(W)$. We let $\sigma(W) \in \mathbb{R}_+^T$ be the vector formed by the singular values of W , where $r = \min(d, T)$, and where we assume that the singular values are ordered nonincreasing, i.e. $\sigma_1(W) \geq \dots \geq \sigma_r(W) \geq 0$. We use \mathbf{S}^d to denote the set of real $d \times d$ symmetric matrices, and \mathbf{S}_+^d to denote the subset of positive semidefinite matrices. We use \succeq to denote the positive semidefinite ordering on \mathbf{S}^d . The notation (\cdot, \cdot) denotes the standard inner products on \mathbb{R}^d and $\mathbb{R}^{d \times T}$, that is $(x, y) = \sum_{i=1}^d x_i y_i$ for $x, y \in \mathbb{R}^d$, and $(X, Y) = \text{tr}(X^T Y)$, for $X, Y \in \mathbb{R}^{d \times T}$. Given a norm $\|\cdot\|$ on \mathbb{R}^d or $\mathbb{R}^{d \times T}$, $\|\cdot\|_*$ denotes the corresponding dual norm, given by $\|u\|_* = \sup\{u, w\} : \|w\| \leq 1\}$. On \mathbb{R}^d we denote by $\|\cdot\|_2$ the Euclidean norm, and on $\mathbb{R}^{d \times T}$ we denote by $\|\cdot\|_F$ the Frobenius norm and by $\|\cdot\|_{\text{tr}}$ the trace norm, that is the sum of singular values.

1.4 Organization

The paper is organized as follows. In Section 2, we review a general class of norms and characterize their unit ball. In Section 3, we specialize these norms to the box-norm, which we show is a perturbation of the k -support norm. We study the properties of the norms and we describe the geometry of the unit balls. In Section 4, we compute the box-norm and we provide an efficient method to compute the proximity operator of the squared norm. In Section 5, we extend the norms to orthogonally invariant matrix norms—the spectral k -support and spectral box-norms—and we show that these exhibit a number of properties which relate to the vector properties in a natural manner. In Section 6, we review the clustered multitask learning setting, we recall the cluster norm introduced by Jacob et al. (2009a) and we show that the cluster norm corresponds to the spectral box-norm. We also provide a method for solving the resulting matrix regularization problem using “centered” norms. In Section 7, we apply the norms to matrix learning problems on a number of simulated and real data sets and report on their performance. In Section 8, we discuss extensions to the framework and suggest directions for future research. Finally, in Section 9, we conclude.

2. Preliminaries

In this section we review a family of norms parameterized by a set Θ , and which we call the Θ -norms. They are closely related to the norms considered in Micchelli et al. (2010, 2013). Similar norms are also discussed in Bach et al. (2011, Sect. 1.4.2) where they are called H -norms. We first recall the definition of the norm.

Definition 1 Let Θ be a convex bounded subset of the open positive orthant. For $w \in \mathbb{R}^d$ the Θ -norm is defined as

$$\|w\|_{\Theta} = \sqrt{\inf_{\theta \in \Theta} \sum_{i=1}^d \frac{w_i^2}{\theta_i}}. \quad (1)$$

Note that the function $(w, \theta) \mapsto \sum_{i=1}^d \frac{w_i^2}{\theta_i}$ is strictly convex on $\mathbb{R}^d \times \mathbb{R}_{++}^d$, hence every minimizing sequence converges to the same point. The infimum is, however, not attained in general because a minimizing sequence may converge to a point on the boundary of Θ . For instance, if $\Theta = \{\theta \in \mathbb{R}_{++}^d : \sum_{i=1}^d \theta_i \leq 1\}$, then $\|w\|_{\Theta} = \|w\|_1$ and the minimizing sequence converges to the point $(\frac{w_1}{\|w\|_1}, \dots, \frac{w_d}{\|w\|_1})$, which belongs to Θ only if all the components of w are different from zero.

Proposition 2 The Θ -norm is well defined and the dual norm is given, for $u \in \mathbb{R}^d$, by

$$\|u\|_{*\Theta} = \sqrt{\sup_{\theta \in \Theta} \sum_{i=1}^d \theta_i u_i^2}. \quad (2)$$

Proof Consider the expression for the dual norm. The function $\|\cdot\|_{*\Theta}$ is a norm since it is a supremum of norms. Recall that the Fenchel conjugate h^* of a function $h: \mathbb{R}^d \rightarrow \mathbb{R}$ is defined for every $u \in \mathbb{R}^d$ as $h^*(u) = \sup \{\langle u, w \rangle - h(w) : w \in \mathbb{R}^d\}$. It is a standard result from convex analysis that for any norm $\|\cdot\|$, the Fenchel conjugate of the function $h := \frac{1}{2} \|\cdot\|^2$ satisfies $h^* = \frac{1}{2} \|\cdot\|_*^2$, where $\|\cdot\|_*$ is the corresponding dual norm (see, e.g. Lewis, 1995). By the same result, for any norm the biconjugate is equal to the norm, that is $(\|\cdot\|_*)^* = \|\cdot\|$. Applying this to the dual norm we have, for every $w \in \mathbb{R}^d$, that

$$h(w) = \sup_{u \in \mathbb{R}^d} \{\langle w, u \rangle - h^*(u)\} = \sup_{u \in \mathbb{R}^d} \inf_{\theta \in \Theta} \left\{ \sum_{i=1}^d \left(w_i u_i - \frac{1}{2} \theta_i u_i^2 \right) \right\}.$$

This is a minimax problem in the sense of von Neumann (see e.g. Prop. 2.6.3 in Bertsekas et al., 2003), and we can exchange the order of the inf and the sup, and solve the latter (which is in fact a maximum) componentwise. The gradient with respect to u_i is zero for $u_i = \frac{w_i}{\theta_i}$, and substituting this into the objective function we obtain $h(w) = \frac{1}{2} \|w\|_{\Theta}^2$. It follows that the expression in (1) defines a norm, and its dual norm is defined by (2), as required. ■

The Θ -norm (1) encompasses a number of well known norms. For instance, for $p \in [1, \infty)$ the ℓ_p norm is defined, for every $w \in \mathbb{R}^d$, as $\|w\|_p = (\sum_{i=1}^d |w_i|^p)^{\frac{1}{p}}$, if $p \in [1, \infty)$ and

$\|w\|_{\infty} = \max_{i=1}^d |w_i|$. For $p \in [1, 2)$, one can show (Micchelli and Pontil, 2005, Lemma 26), that $\|w\|_p = \|w\|_{\Theta_p}$, where we have defined $\Theta_p = \{\theta \in \mathbb{R}_{++}^d : \sum_{i=1}^d \theta_i^{\frac{2}{2-p}} \leq 1\}$. For $p = 1$ this confirms the set Θ corresponding to the ℓ_1 norm as claimed above. Similarly, for $p \in (2, \infty]$ we have that $\|w\|_p = \|w\|_{*\Theta_p}$, where $\frac{1}{p} + \frac{1}{q} = 1$. The ℓ_2 -norm is obtained as both a primal and dual Θ -norm in the limit as p tends to 2. See also Aflalo et al. (2011) who considered the case of $p > 2$.

Other norms which belong to the family (1) are presented in (Micchelli et al., 2013) and correspond to choosing $\Theta = \{\theta \in \Lambda : \sum_{i=1}^d \theta_i \leq 1\}$, where $\Lambda \subseteq \mathbb{R}_{++}^d$ is a convex cone. A specific example described therein is the wedge penalty, which corresponds to choosing $\Lambda = \{\theta \in \mathbb{R}_{++}^d, \theta_1 \geq \dots \geq \theta_d\}$.

We now describe the unit ball of the Θ -norm when the set Θ is a polyhedron and we characterize the unit ball of the norm. This setting applies to a number of norms of practical interest, including the group lasso with overlap, the wedge norm mentioned above and, as we shall see, the k -support norm. To describe our observation, for every vector $\gamma \in \mathbb{R}_{++}^d$, we define the seminorm

$$\|w\|_{\gamma} = \sqrt{\sum_{i:\gamma_i>0} \frac{w_i^2}{\gamma_i}}.$$

Proposition 3 Let $\gamma^1, \dots, \gamma^m \in \mathbb{R}_{++}^d$ such that $\sum_{\ell=1}^m \gamma^{\ell} \in \mathbb{R}_{++}^d$ and let $\Theta = \{\theta \in \mathbb{R}_{++}^d : \theta = \sum_{\ell=1}^m \lambda \gamma^{\ell}, \lambda \in \Delta^{m-1}\}$.

Then we have, for every $w \in \mathbb{R}^d$, that

$$\|w\|_{\Theta} = \inf \left\{ \sum_{\ell=1}^m \|v_{\ell}\|_{\gamma^{\ell}} : v_{\ell} \in \mathbb{R}^d, \text{supp}(v_{\ell}) \subseteq \text{supp}(\gamma^{\ell}), \ell \in \mathbb{N}_m, \sum_{\ell=1}^m v_{\ell} = w \right\}. \quad (3)$$

Moreover, the unit ball of the norm is given by the convex hull of the set

$$\bigcup_{\ell=1}^m \left\{ w \in \mathbb{R}^d : \text{supp}(w) \subseteq \text{supp}(\gamma^{\ell}), \|w\|_{\gamma^{\ell}} \leq 1 \right\}. \quad (4)$$

The proof of this result is presented in the appendix. It is based on observing that the Minkowski functional (see e.g. Rudin, 1991) of the convex hull of the set (4) is a norm and it is given by the right hand side of equation (3); we then prove that this norm coincides with $\|\cdot\|_{\Theta}$ by noting that both norms share the same dual norm. To illustrate an application of the proposition, we specialize it to the group Lasso with overlap (Jacob et al., 2009b).

Corollary 4 If \mathcal{G} is a collection of subsets of \mathbb{N}_d such that $\bigcup_{g \in \mathcal{G}} g = \mathbb{N}_d$ and Θ is the interior of the set $\text{co}\{1_g : g \in \mathcal{G}\}$, then we have, for every $w \in \mathbb{R}^d$, that

$$\|w\|_{\Theta} = \inf \left\{ \sum_{g \in \mathcal{G}} \|v_g\|_2 : v_g \in \mathbb{R}^d, \text{supp}(v_g) \subseteq g, \sum_{g \in \mathcal{G}} v_g = w \right\}. \quad (5)$$

Moreover, the unit ball of the norm is given by the convex hull of the set

$$\bigcup_{g \in \mathcal{G}} \left\{ w \in \mathbb{R}^d : \text{supp}(w) \subseteq g, \|w\|_2 \leq 1 \right\}. \quad (6)$$

We do not claim any originality in the above corollary and proposition, although we cannot find a specific reference. The utility of the result is that it links seemingly different norms such as the group Lasso with overlap and the Θ -norms, which provide a more compact representation, involving only d additional variables. This formulation is especially useful whenever the optimization problem (1) can be solved in closed form. One such example is provided by the wedge norm described above. In the next section we discuss one more important case, the box-norm, which plays a central role in this paper.

3. The Box-Norm and the k -Support Norm

We now specialize our analysis to the case that

$$\Theta = \left\{ \theta \in \mathbb{R}^d : a \leq \theta_i \leq b, \sum_{i=1}^d \theta_i \leq c \right\} \quad (7)$$

where $0 < a \leq b$ and $c \in [ad, bd]$. We call the norm defined by (1) the *box-norm* and we denote it by $\|\cdot\|_{\text{box}}$.

The structure of set Θ for the box-norm will be fundamental in computing the norm and deriving the proximity operator in Section 4. Furthermore, we note that the constraints are invariant with respect to permutations of the components of Θ and, as we shall see in Section 5, this property is key to extending the norm to matrices. Finally, while a restriction of the general family, the box-norm nevertheless encompasses a number of norms including the ℓ_1 and ℓ_2 norms, as well as the k -support norm, which we now recall.

For every $k \in \mathbb{N}_d$, the k -support norm $\|\cdot\|_{(k)}$ (Argyriou et al., 2012) is defined as the norm whose unit ball is the convex hull of the set of vectors of cardinality at most k and ℓ_2 -norm no greater than one. The authors show that the k -support norm can be written as the infimal convolution (see Rockafellar, 1970, p. 34)

$$\|w\|_{(k)} = \inf \left\{ \sum_{g \in G_k} \|v_g\|_2 : v_g \in \mathbb{R}^d, \text{supp}(v_g) \subseteq g, \sum_{g \in G_k} v_g = w \right\}, \quad w \in \mathbb{R}^d, \quad (8)$$

where G_k is the collection of all subsets of \mathbb{N}_d containing at most k elements. The k -support norm is a special case of the group lasso with overlap (Jacob et al., 2009b), where the cardinality of the support sets is at most k . When used as a regularizer, the norm encourages vectors w to be a sum of a limited number of vectors with small support. Note that while definition (8) involves a combinatorial number of variables, Argryiou et al. (2012) observed that the norm can be computed in $\mathcal{O}(d \log d)$, a point we return to in Section 4.

Comparing equation (8) with Corollary 4 it is evident that the k -support norm is a Θ -norm where $\Theta = \{\theta \in \mathbb{R}_+^d : \theta = \sum_{g \in G_k} \lambda_g 1_g, \lambda \in \Delta^{(|G_k|-1)}\}$, which by symmetry can be expressed as $\Theta = \{\theta : 0 < \theta_i \leq 1, \sum_{i=1}^d \theta_i \leq k\}$. Hence, we see that the k -support norm is a special case of the box-norm.

Despite the complicated form of (8), Argryiou et al. (2012) observe that the dual norm has a simple formulation, namely the ℓ_2 -norm of the k largest components,

$$\|u\|_{*(k)} = \sqrt{\sum_{i=1}^k (u_i^\dagger)^2}, \quad u \in \mathbb{R}^d, \quad (9)$$

where $|u|^\dagger$ is the vector obtained from u by reordering its components so that they are non-increasing in absolute value. Note from equation (9) that for $k = 1$ and $k = d$, the dual norm is equal to the ℓ_∞ -norm and ℓ_2 -norm, respectively. It follows that the k -support norm includes the ℓ_1 -norm and ℓ_2 -norm as special cases.

We now provide a different argument illustrating that the k -support norm belongs to the family of box-norms using the dual norm. We first derive the dual box-norm.

Proposition 5 *The dual box-norm is given by*

$$\|u\|_{*\text{box}}^2 = a\|u\|_2^2 + (b-a) \left(\|u\|_{*(k)}^2 + (\rho - k)(|u|_{k+1}^\dagger)^2 \right), \quad (10)$$

where $\rho = \frac{c-da}{b-a}$ and k is the largest integer not exceeding ρ .

Proof We need to solve problem (2). We make the change of variable $\phi_i = \frac{\theta_i - a}{b-a}$ and observe that the constraints on θ induce the constraint set $\{\phi \in (0, 1]^d, \sum_{i=1}^d \phi_i \leq \rho\}$, where $\rho = \frac{c-da}{b-a}$. Furthermore $\sum_{i=1}^d \theta_i a_i^2 = a\|u\|_2^2 + (b-a) \sum_{i=1}^d \phi_i u_i^2$. The result then follows by taking the supremum over ϕ . ■

We see from equation (10) that the dual norm decomposes into a weighted combination of the ℓ_2 -norm, the k -support norm and a residual term, which vanishes if $\rho = k \in \mathbb{N}_d$. For the rest of this paper we assume this holds, which loses little generality. This choice is equivalent to requiring that $c = (b-a)k + da$, which is the case considered by Jacob et al. (2009a) in the context of multitask clustering, where $k+1$ is interpreted as the number of clusters and d as the number of tasks. We return to this case in Section 6, where we explain in detail the link between the spectral k -support norm and the cluster norm.

Observe that if $a = 0$, $b = 1$, and $\rho = k$, the dual box-norm (10) coincides with dual k -support norm in equation (9). We conclude that if

$$\Theta = \left\{ \theta \in \mathbb{R}^d : 0 < \theta_i \leq 1, \sum_{i=1}^d \theta_i \leq k \right\}$$

then the Θ -norm coincides with the k -support norm.

3.1 Properties of the Norms

In this section we illustrate a number of properties of the box-norm and the connection to the k -support norm. The first result follows as a special case of Proposition 3.

Corollary 6 *If $0 < a < b$ and $c = (b-a)k + da$, for $k \in \mathbb{N}_d$, then it holds that*

$$\|w\|_{\text{box}} = \inf \left\{ \sum_{g \in G_k} \sqrt{\frac{v_{g,i}^2}{b} + \sum_{i \notin g} \frac{v_{g,i}^2}{a}} : v_g \in \mathbb{R}^d, \sum_{g \in G_k} v_g = w \right\}, \quad w \in \mathbb{R}^d.$$

Furthermore, the unit ball of the norm is given by the convex hull of the set

$$\bigcup_{g \in G_k} \left\{ w \in \mathbb{R}^d : \sum_{i \in g} \frac{w_i^2}{b} + \sum_{i \notin g} \frac{w_i^2}{a} \leq 1 \right\}. \quad (11)$$

Notice in Equation (11) that if $b = 1$, then as a tends to zero, we obtain the expression of the k -support norm (8), recovering in particular the support constraints. If a is small and positive, the support constraints are not imposed, however most of the weight for each v_g tends to be concentrated on $\text{supp}(g)$. Hence, Corollary 6 suggests that if $a \ll b$ then the box-norm regularizer will encourage vectors w whose dominant components are a subset of a union of a small number of groups $g \in \mathcal{G}_k$.

Our next result links two Θ -norms whose parameter sets are related by a linear transformation with positive coefficients.

Lemma 7 *Let Θ be a convex bounded subset of the positive orthant in \mathbb{R}^d , and let $\Phi = \{\phi \in \mathbb{R}^d : \phi_i = \alpha + \beta\theta_i, \theta \in \Theta\}$, where $\alpha, \beta > 0$. Then*

$$\|w\|_{\Phi}^2 = \min_{z \in \mathbb{R}^d} \left\{ \frac{1}{\alpha} \|w - z\|_2^2 + \frac{1}{\beta} \|z\|_{\Theta}^2 \right\}.$$

Proof We consider the definition of the norm $\|\cdot\|_{\Phi}$ in (1). We have

$$\|w\|_{\Phi}^2 = \inf_{\phi \in \Phi} \sum_{i=1}^d \frac{w_i^2}{\phi_i} = \inf_{\theta \in \Theta} \sum_{i=1}^d \frac{w_i^2}{\alpha + \beta\theta_i}, \quad (12)$$

where we have made the change of variable $\phi_i = \alpha + \beta\theta_i$. Next we observe that

$$\min_{z \in \mathbb{R}^d} \left\{ \frac{1}{\alpha} \|w - z\|_2^2 + \frac{1}{\beta} \|z\|_{\Theta}^2 \right\} = \min_{z \in \mathbb{R}^d} \inf_{\theta \in \Theta} \left\{ \sum_{i=1}^d \frac{(w_i - z_i)^2}{\alpha} + \frac{z_i^2}{\beta\theta_i} \right\} = \inf_{\theta \in \Theta} \sum_{i=1}^d \frac{w_i^2}{\alpha + \beta\theta_i}, \quad (13)$$

where we interchanged the order of the minimum and the infimum and solved for z componentwise, setting $z_i = \frac{\beta\theta_i w_i}{\alpha + \beta\theta_i}$. The result now follows by combining equations (12) and (13). ■

In Section 3 we characterized the k -support norm as a special case of the box-norm. Conversely, Lemma 7 allows us to interpret the box-norm as a perturbation of the k -support norm with a quadratic regularization term.

Proposition 8 *Let $\|\cdot\|_{\text{box}}$ be the box-norm on \mathbb{R}^d with parameters $0 < a < b$ and $c = k(b - a) + da$, for $k \in \mathbb{N}_d^+$, then*

$$\|w\|_{\text{box}}^2 = \min_{z \in \mathbb{R}^d} \left\{ \frac{1}{a} \|w - z\|_2^2 + \frac{1}{b - a} \|z\|_{\tau(k)}^2 \right\}. \quad (14)$$

Proof The result directly follows from Lemma 7 for $\Theta = \{\theta \in \mathbb{R}^d : 0 < \theta_i \leq 1, \sum_{i=1}^d \theta_i \leq k\}$, $\alpha = a$ and $\beta = b - a$. ■

Lemma 7 and Proposition 8 can further be interpreted using the Moreau envelope from convex optimization, which we now recall (Rockafellar and Wets, 2009, Ch. 1 §G).

Definition 9 *Let $f : \mathbb{R}^d \rightarrow (-\infty, \infty]$ be proper, lower semi-continuous and let $\rho > 0$. The Moreau envelope of f with parameter ρ is defined as*

$$e_{\rho}f(w) = \inf_{z \in \mathbb{R}^d} \left\{ f(z) + \frac{1}{2\rho} \|w - z\|_2^2 \right\}.$$

Note that $e_{\rho}f$ minorizes f and is convex and smooth (Bauschke and Combettes, 2010, see e.g.). It acts as a parameterized smooth approximation to f from below, which motivates its use in variational analysis (see e.g. Rockafellar and Wets, 2009, for further discussion). Lemma 7 therefore says that $\beta\|\cdot\|_{\Theta}^2$ is a Moreau-envelope of $\|\cdot\|_{\Theta}^2$ with parameter $\rho = \frac{2}{\beta}$ whenever Φ is defined as $\Phi = \alpha + \beta\Theta$, $\alpha, \beta > 0$. In particular we see from (14) that the squared box-norm, scaled by a factor of $(b - a)$, is a Moreau envelope of the squared k -support norm as we have

$$(b - a)\|w\|_{\text{box}}^2 = \min_{z \in \mathbb{R}^d} \left\{ \|z\|_{\tau(k)}^2 + \frac{1}{2\rho} \|w - z\|_2^2 \right\} =: e_{\rho}f(w), \quad (15)$$

where $f(w) = \|w\|_{\tau(k)}^2$ and $\rho = \frac{1}{2(b-a)}$.

Proposition 8 further allows us to decompose the solution to a vector learning problem using the squared box-norm into two components with particular structure. Specifically, consider the regularization problem

$$\min_{w \in \mathbb{R}^d} \|Xw - y\|_2^2 + \lambda\|w\|_{\text{box}}^2 \quad (16)$$

with data $X \in \mathbb{R}^{n \times d}$ and response $y \in \mathbb{R}^n$. Using Proposition 8 and setting $w = u + z$, we see that (16) is equivalent to

$$\min_{u, z \in \mathbb{R}^d} \left\{ \|X(u + z) - y\|_2^2 - \frac{\lambda}{a} \|u\|_2^2 + \frac{\lambda}{b - a} \|z\|_{\tau(k)}^2 \right\}. \quad (17)$$

Furthermore, if (\hat{u}, \hat{z}) solves problem (17) then $\hat{w} = \hat{u} + \hat{z}$ solves problem (16). The solution \hat{w} can therefore be interpreted as the superposition of a vector which has small ℓ_2 norm, and a vector which has small k -support norm, with the parameter a regulating these two components. Specifically, as a tends to zero, in order to prevent the objective from blowing up, \hat{u} must also tend to zero and we recover k -support norm regularization. Similarly, as a tends to b , \hat{z} vanishes and we have a simple ridge regression problem.

A further consequence of Proposition 8 is the differentiability of the squared box-norm.

Proposition 10 *If $a > 0$ the squared box-norm is differentiable on \mathbb{R}^d and its gradient*

$$\nabla(\|\cdot\|_{\text{box}}^2) = \frac{2}{a} \left(\text{Id} - \text{prox}_{\rho\|\cdot\|_{\tau(k)}} \right)$$

is Lipschitz continuous with parameter $\frac{2}{a}$.

Proof Letting $f(w) = \|w\|_{\tau(k)}^2$, $\rho = \frac{1}{2(b-a)}$, by (15) we have $e_{\rho}f(w) = (b - a)\|w\|_{\text{box}}^2$. The result follows directly from Bauschke and Combettes (2010, Prop. 12.29), as f is convex and continuous on \mathbb{R}^d and the gradient is given as $\nabla(e_{\rho}f) = \frac{1}{\rho}(\text{Id} - \text{prox}_{\rho f})$. ■

Proposition 10 establishes that the square of the box-norm is differentiable and its smoothness is controlled by the parameter a . Furthermore, the gradient can be determined from the proximity operator, which we compute in Section 4.

3.2 Geometry of the Norms

In this section, we briefly investigate the geometry of the box-norm. Figure 1 depicts the unit balls for the k -support norm in \mathbb{R}^3 for various parameter values, setting $b = 1$ throughout. For $k = 1$ and $k = 3$ we recognize the ℓ_1 and ℓ_2 balls respectively. For $k = 2$ the unit ball retains characteristics of both norms, and in particular we note the discontinuities along each of x , y and z planes, as in the case of the ℓ_1 norm.

Figure 2 depicts the unit balls for the box-norm for a range of values of a and k , with $c = (b - a)k + da$. We see that in general the balls increase in volume with each of a and k , holding the other parameter fixed. Comparing the k -support norm ($k = 1$), that is the ℓ_1 norm, and the box-norm ($k = 1, a = 0.15$), we see that the parameter a smooths out the sharp edges of the ℓ_1 norm. This is also visible when comparing the k -support ($k = 2$) and the box ($k = 2, a = 0.15$). This illustrates the smoothing effect of the parameter a , as suggested by Proposition 10.

We can gain further insight into the shape of the unit balls of the box-norm from Corollary 6. Equation (11) shows that the primal unit ball is the convex hull of ellipsoids in \mathbb{R}^d , where for each group g the semi-principle axis along dimension i has length \sqrt{b} if $i \in g$, and length \sqrt{a} if $i \notin g$. Similarly, the unit ball of the dual box-norm is the intersection of ellipsoids in \mathbb{R}^d where for each group g the semi-principle axis in dimension i has length $1/\sqrt{b}$ if $i \in g$, and length $1/\sqrt{a}$ if $i \notin g$ (see also Equation 37 in the appendix). It is instructive to further consider the effect of the parameter a on the unit balls for fixed k . To this end, recall that since $c = (b - a)k + da$, when $k = d$ we have $c = bd$. In this case, for all values of a in $(0, b]$, the objective in (1) is attained by setting $\theta_i = b$ for all i , and we recover the ℓ_2 -norm, scaled by $1/\sqrt{b}$, for the primal box-norm. Similarly in (2), the dual norm gives rise to the ℓ_2 -norm, scaled by \sqrt{b} . In the remainder of this section we therefore only consider the cases $k \in \{1, 2\}$ in \mathbb{R}^3 .

For $k = 1$, $\mathcal{G}_k = \{\{1\}, \{2\}, \{3\}\}$. The unit ball of the primal box-norm is the convex hull of the ellipsoids defined by

$$\frac{w_1^2}{b} + \frac{w_2^2}{a} + \frac{w_3^2}{a} = 1, \quad \frac{w_1^2}{a} + \frac{w_2^2}{b} + \frac{w_3^2}{a} = 1, \quad \text{and} \quad \frac{w_1^2}{a} + \frac{w_2^2}{a} + \frac{w_3^2}{b} = 1, \quad (18)$$

and the unit ball of the dual box-norm is the intersection of the ellipsoids defined by

$$\frac{w_1^2}{b-1} + \frac{w_2^2}{a-1} + \frac{w_3^2}{a-1} = 1, \quad \frac{w_1^2}{a-1} + \frac{w_2^2}{b-1} + \frac{w_3^2}{a-1} = 1, \quad \text{and} \quad \frac{w_1^2}{a-1} + \frac{w_2^2}{a-1} + \frac{w_3^2}{b-1} = 1. \quad (19)$$

For $k = 2$, $\mathcal{G}_k = \{\{1\}, \{2\}, \{3\}, \{1, 2\}, \{2, 3\}, \{1, 3\}\}$. The unit ball of the primal box-norm is the convex hull of the ellipsoids defined by (18) in addition to the following

$$\frac{w_1^2}{b} + \frac{w_2^2}{b} + \frac{w_3^2}{a} = 1, \quad \frac{w_1^2}{a} + \frac{w_2^2}{b} + \frac{w_3^2}{b} = 1, \quad \text{and} \quad \frac{w_1^2}{b} + \frac{w_2^2}{a} + \frac{w_3^2}{b} = 1, \quad (20)$$

and the unit ball of the dual box-norm is the intersection of the ellipsoids defined by (19) in addition to the following

$$\frac{w_1^2}{b-1} + \frac{w_2^2}{b-1} + \frac{w_3^2}{a-1} = 1, \quad \frac{w_1^2}{a-1} + \frac{w_2^2}{b-1} + \frac{w_3^2}{a-1} = 1, \quad \text{and} \quad \frac{w_1^2}{b-1} + \frac{w_2^2}{a-1} + \frac{w_3^2}{b-1} = 1. \quad (21)$$

For the primal norm, note that since $b > a$, each of the ellipsoids in (18) is entirely contained within one of those defined by (20), hence when taking the convex hull we need only consider the latter set. Similarly for the dual norm, since $\frac{1}{b} < \frac{1}{a}$, each of the ellipsoids in (19) is contained within one of those defined by (21), hence when taking the intersection we need only consider the latter set.

Figures 3 and 4 depict the constituent ellipses for various parameter values for the primal and dual norms. As a tends to zero the ellipses become degenerate. For $k = 1$, taking the convex hull we recover the ℓ_1 unit ball in the primal norm, and taking the intersection we recover the ℓ_∞ unit ball in the dual norm. As a tends to 1 we recover the ℓ_2 norm in both the primal and the dual.

4. Computation of the Norm and the Proximity Operator

In this section, we compute the norm and the proximity operator of the squared box-norm by explicitly solving the optimization problem (1). We also specialize our results to the k -support norm and comment on the improvement with respect to the method by Argyron et al. (2012). Recall that, for every vector $w \in \mathbb{R}^d$, $|w|^\dagger$ denotes the vector obtained from w by reordering its components so that they are non-increasing in absolute value.

Theorem 11 For every $w \in \mathbb{R}^d$ it holds that

$$\|w\|_{\text{box}}^2 = \frac{1}{b}\|w_Q\|_2^2 + \frac{1}{a}\|w_I\|_2^2 + \frac{1}{a}\|w_L\|_2^2, \quad (22)$$

where $w_Q = (|w|_1^\dagger, \dots, |w|_\ell^\dagger)$, $w_I = (|w|_{q+1}^\dagger, \dots, |w|_{d-\ell}^\dagger)$, $w_L = (|w|_{d-\ell+1}^\dagger, \dots, |w|_d^\dagger)$, q and ℓ are the unique integers in $\{0, \dots, d\}$ that satisfy $q + \ell \leq d$,

$$\frac{|w|_q|}{b} \geq \frac{1}{p} \sum_{i=q+1}^{d-\ell} |w_i| > \frac{|w_{q+1}|}{b}, \quad \frac{|w_{d-\ell}|}{a} \geq \frac{1}{p} \sum_{i=q+1}^{d-\ell} |w_i| > \frac{|w_{d-\ell+1}|}{a}, \quad (23)$$

$p = c - qb - la$ and we have defined $|w_0| = \infty$ and $|w_{d+1}| = 0$. Furthermore, the minimizer θ has the form

$$\theta_i = \begin{cases} b, & \text{if } i \in \{1, \dots, q\}, \\ p \frac{\sum_{j=q+1}^d |w_j|}{\sum_{j=q+1}^d |w_j|}, & \text{if } i \in \{q+1, \dots, d-\ell\}, \\ a, & \text{otherwise.} \end{cases}$$

Proof We solve the constrained optimization problem

$$\inf \left\{ \sum_{i=1}^d \frac{w_i^2}{\theta_i} : a \leq \theta_i \leq b, \sum_{i=1}^d \theta_i \leq c \right\}. \quad (24)$$

To simplify the notation we assume without loss of generality that w_i are positive and ordered nonincreasing, and note that the optimal θ_i are ordered non increasing. To see this, let $\theta^* = \arg\min_{\theta \in \Theta} \sum_{i=1}^d \frac{w_i^2}{\theta_i}$. Now suppose that $\theta_i^* < \theta_j^*$ for some $i < j$ and define $\hat{\theta}$ to

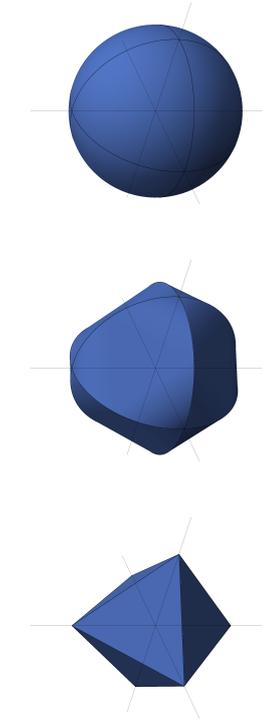


Figure 1: Unit balls of the k -support norm for $k \in \{1, 2, 3\}$.

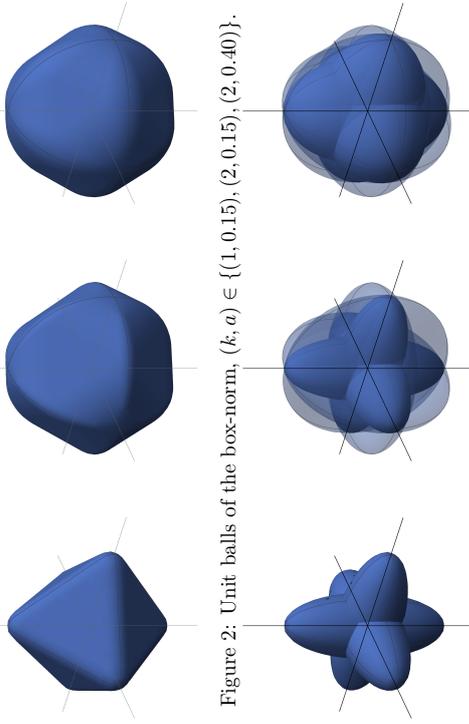


Figure 2: Unit balls of the box-norm, $(k, a) \in \{(1, 0.15), (2, 0.15), (2, 0.40)\}$.

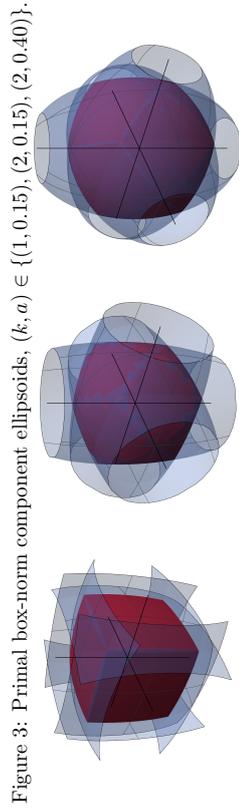


Figure 3: Primal box-norm component ellipsoids, $(k, a) \in \{(1, 0.15), (2, 0.15), (2, 0.40)\}$.

Figure 4: Dual box-norm unit balls and ellipsoids, $(k, a) \in \{(1, 0.15), (2, 0.15), (2, 0.40)\}$. For $k = 2$, only 3 tightest ellipsoids are shown.

be identical to θ^* , except with the i and j elements exchanged. The difference in objective values is

$$\sum_{i=1}^d \frac{w_i^2}{\theta_i^*} - \sum_{i=1}^d \frac{w_i^2}{\theta_i^*} = (w_i^2 - w_j^2) \left(\frac{1}{\theta_j^*} - \frac{1}{\theta_i^*} \right),$$

which is negative so θ^* cannot be a minimizer.

We further assume without loss of generality that $w_i \neq 0$ for all i , and $c \leq db$ (see Remark 12 below). The objective is continuous and we take the infimum over a closed bounded set, so a solution exists and it is unique by strict convexity. Furthermore, since $c \leq db$, the sum constraint will be tight at the optimum. Consider the Lagrangian function

$$L(\theta, \alpha) = \sum_{i=1}^d \frac{w_i^2}{\theta_i} + \frac{1}{\alpha^2} \left(\sum_{i=1}^d \theta_i - c \right), \tag{25}$$

where $1/\alpha^2$ is a strictly positive multiplier, and α is to be chosen to make the sum constraint tight, call this value α^* . Let θ^* be the minimizer of $L(\theta, \alpha^*)$ over θ subject to $a \leq \theta_i \leq b$.

We claim that θ^* solves equation (24). Indeed, for any $\theta \in [a, b]^d$, $L(\theta^*, \alpha^*) \leq L(\theta, \alpha^*)$, which implies that

$$\sum_{i=1}^d \frac{w_i^2}{\theta_i^*} \leq \sum_{i=1}^d \frac{w_i^2}{\theta_i} + \frac{1}{(\alpha^*)^2} \left(\sum_{i=1}^d \theta_i - c \right).$$

If in addition we impose the constraint $\sum_{i=1}^d \theta_i \leq c$, the second term on the right hand side is at most zero, so we have for all such θ that

$$\sum_{i=1}^d \frac{w_i^2}{\theta_i^*} \leq \sum_{i=1}^d \frac{w_i^2}{\theta_i},$$

whence it follows that θ^* is the minimizer of (24).

We can therefore solve the original problem by minimizing the Lagrangian (25) over the box constraint. Due to the coupling effect of the multiplier, the problem is separable, and we can solve the simplified problem componentwise (see Michelli et al., 2013, Theorem 3.1). For completeness we repeat the argument here. For every $w_i \in \mathbb{R}$ and $\alpha > 0$, the unique solution to the problem $\min\{\frac{w_i^2}{\theta} + \frac{\theta}{\alpha^2} : a \leq \theta \leq b\}$ is given by

$$\theta = \begin{cases} b, & \text{if } \alpha|w_i| > b, \\ \alpha|w_i|, & \text{if } b \geq \alpha|w_i| \geq a, \\ a, & \text{if } a > \alpha|w_i|. \end{cases} \tag{26}$$

Indeed, for fixed w_i , the objective function is strictly convex on \mathbb{R}_{++}^d , and has a unique minimum on $(0, \infty)$ (see Figure 1.b in Michelli et al. (2013) for an illustration). The derivative of the objective function is zero for $\theta = \theta^* := \alpha|w_i|$, strictly positive below θ^* and strictly increasing above θ^* . Considering these three cases the result follows and θ is determined by (26) where α satisfies $\sum_{i=1}^d \theta_i(\alpha) = c$.

The minimizer then has the form

$$\theta = (\underbrace{b, \dots, b}_q, \theta_{q+1}, \dots, \theta_{d-\ell}, \underbrace{a, \dots, a}_\ell, d),$$

where $q, \ell \in \{0, \dots, d\}$ are determined by the value of α which satisfies

$$S(\alpha) = \sum_{i=1}^d \theta_i(\alpha) = qb + \sum_{i=q+1}^{d-\ell} \alpha |w_i| + \ell a = c,$$

i.e. $\alpha = p / \left(\sum_{i=q+1}^{d-\ell} |w_i| \right)$, where $p = c - qb - \ell a$.

The value of the norm follows by substituting θ into the objective and we get

$$\|w\|_{\text{box}}^2 = \sum_{i=1}^q \frac{|w_i|^2}{b} + \frac{1}{p} \left(\sum_{i=q+1}^{d-\ell} |w_i| \right)^2 + \sum_{i=d-\ell+1}^d \frac{|w_i|^2}{a} = \frac{1}{b} \|w_Q\|_2^2 + \frac{1}{p} \|w_I\|_1^2 + \frac{1}{a} \|w_A\|_2^2,$$

as required. We can further characterize q and ℓ by considering the form of θ . By construction we have $\theta_q \geq b > \theta_{q+1}$ and $\theta_{d-\ell} > a \geq \theta_{d-\ell+1}$, or equivalently

$$\frac{|w_q|}{b} \geq \frac{1}{p} \sum_{i=q+1}^{d-\ell} |w_i| > \frac{|w_{q+1}|}{b} \quad \text{and} \quad \frac{|w_{d-\ell}|}{a} \geq \frac{1}{p} \sum_{i=q+1}^{d-\ell} |w_i| > \frac{|w_{d-\ell+1}|}{a}.$$

The proof is completed. \blacksquare

Remark 12 *The case where some w_i are zero follows from the case that we have considered in the theorem. If $w_i = 0$ for $n < i \leq d$, then clearly we must have $\theta_i = a$ for all such i . We then consider the n -dimensional problem of finding $(\theta_1, \dots, \theta_n)$ that minimizes $\sum_{i=1}^n \frac{w_i^2}{\theta_i}$, subject to $a \leq \theta_i \leq b$, and $\sum_{i=1}^n \theta_i \leq c'$, where $c' = c - (d - n)a$. As $c \geq da$ by assumption, we also have $c' \geq na$, so a solution exists to the n -dimensional problem. If $c' \geq bn$, then a solution is trivially given by $\theta_i = b$ for all $i = 1, \dots, n$. In general, $c' < bn$, and we proceed as per the proof of the theorem. Finally, a vector that solves the original d -dimensional problem will be given by $(\theta_1, \dots, \theta_n, a, \dots, a)$.*

Theorem 11 suggests two methods for computing the box-norm. First, we can find α such that $S(\alpha) = c$; this value uniquely determines θ in (26), and the norm follows by substitution into the objective in (25). Alternatively, we identify q and ℓ that jointly satisfy (23) and we compute the norm using (22). Taking advantage of the structure of θ in the former method leads to a computation time that is $\mathcal{O}(d \log d)$.

Theorem 13 *The computation of the box-norm can be completed in $\mathcal{O}(d \log d)$ time.*

Proof Following Theorem 11, we need to determine α^* to satisfy the coupling constraint $S(\alpha^*) = c$. Each component θ_i is a piecewise linear function in the form of a step function with a constant positive slope between the values $a/|w_i|$ and $b/|w_i|$. Let $\{\alpha^i\}_{i=1}^{2d}$ be the

set of the $2d$ critical points, where the α^i are ordered nondecreasing. The function $S(\alpha)$ is a nondecreasing piecewise linear function with at most $2d$ critical points. We can find α^* by first sorting the points $\{\alpha^i\}$, finding α^i and α^{i+1} such that

$$S(\alpha^i) \leq c \leq S(\alpha^{i+1})$$

by binary search, and then interpolating α^* between the two points. Sorting takes $\mathcal{O}(d \log d)$. Computing $S(\alpha^i)$ at each step of the binary search is $\mathcal{O}(d)$, so $\mathcal{O}(d \log d)$ overall. Given α^i and α^{i+1} , interpolating α^* is $\mathcal{O}(1)$, so the overall algorithm is $\mathcal{O}(d \log d)$ as claimed. \blacksquare

The k -support norm is a special case of the box-norm, and as a direct corollary of Theorem 11 and Theorem 13, we recover (Argyriou et al., 2012; Proposition 2.1):

Corollary 14 *For $w \in \mathbb{R}^d$, and $k \leq d$,*

$$\|w\|_{(k)} = \left(\sum_{j=1}^q (|w_j|^2)^2 + \frac{1}{k-q} \left(\sum_{j=q+1}^d |w_j|^2 \right)^2 \right)^{\frac{1}{2}},$$

where q is the unique integer in $\{0, k-1\}$ satisfying

$$|w_q| \geq \frac{1}{k-q} \sum_{j=q+1}^d |w_j| > |w_{q+1}|, \quad (27)$$

and we have defined $w_0 = \infty$. Furthermore, the norm can be computed in $\mathcal{O}(d \log d)$ time.

4.1 Proximity Operator

Proximal gradient methods can be used to solve optimization problems of the form

$$\min_w f(w) + \lambda g(w), \quad w \in \mathbb{R}^d,$$

where f is a convex loss function with Lipschitz continuous gradient, $\lambda > 0$ is a regularization parameter, and g is a convex function for which the proximity operator can be computed efficiently, see Nesterov (2007); Combettes and Pesquet (2011); Beck and Teboulle (2009) and references therein. The proximity operator of g with parameter $\rho > 0$ is defined as

$$\text{prox}_{\rho g}(w) = \arg \min \left\{ \frac{1}{2} \|x - w\|^2 + \rho g(x) : x \in \mathbb{R}^d \right\}.$$

We now use the infimum formulation of the box-norm to derive the proximity operator of the squared norm.

Theorem 15 *The proximity operator of the square of the box-norm at point $w \in \mathbb{R}^d$ with parameter $\frac{\lambda}{2}$ is given by $\text{prox}_{\frac{\lambda}{2} \|\cdot\|_{\text{box}}^2}(w) = (\frac{\theta_1 w_1}{\theta_1 + \lambda}, \dots, \frac{\theta_n w_n}{\theta_n + \lambda})$, where*

$$\theta_i = \begin{cases} b, & \text{if } \alpha |w_i| - \lambda > b, \\ \alpha |w_i| - \lambda, & \text{if } b \geq \alpha |w_i| - \lambda \geq a, \\ a, & \text{if } a > \alpha |w_i| - \lambda, \end{cases}$$

and α is chosen such that $S(\alpha) := \sum_{i=1}^d \theta_i(\alpha) = c$. Furthermore, the computation of the proximity operator can be completed in $\mathcal{O}(d \log d)$ time.

Proof Using the infimum formulation of the norm, we solve

$$\min_{x \in \mathbb{R}^d} \inf_{\theta \in \Theta} \left\{ \frac{1}{2} \sum_{i=1}^d (x_i - w_i)^2 + \lambda \sum_{i=1}^d \frac{x_i^2}{\theta_i} \right\}.$$

We can exchange the order of the optimization and solve for x first. The problem is separable and a direct computation yields that $x_i = \frac{\theta_i w_i}{\theta_i + \lambda}$. Discarding a multiplicative factor of $\lambda/2$, and noting that the infimum is attained, the problem in θ becomes

$$\min_{\theta} \left\{ \sum_{i=1}^d \frac{w_i^2}{\theta_i + \lambda} : a \leq \theta_i \leq b, \sum_{i=1}^d \theta_i \leq c \right\}.$$

Note that this is the same as computing a box-norm in accordance with Proposition 8. Specifically, this is exactly like problem (24) after the change of variable $\theta'_i = \theta_i + \lambda$. The remaining part of the proof then follows in a similar manner to the proof of Theorem 11. ■

Algorithm 1 illustrates the computation of the proximity operator for the squared box-norm in $\mathcal{O}(d \log d)$ time. This includes the k -support as a special case, where we let a tend to zero, and set $b = 1$ and $c = k$, which improves upon the complexity of the $\mathcal{O}(d(k + \log d))$ computation provided in Argyriou et al. (2012), and we illustrate the improvement empirically in Table 1. We summarize this in the following corollary.

Corollary 16 *The proximity operator of the square of the k -support norm at point w with parameter $\frac{\lambda}{2}$ is given by $\text{prox}_{\frac{\lambda}{2} \|\cdot\|_k^2}(w) = x$, where $x_i = \frac{\theta_i w_i}{\theta_i + \lambda}$, and*

$$\theta_i = \begin{cases} 1, & \text{if } \alpha |w_i| > \lambda + 1, \\ \alpha |w_i| - \lambda, & \text{if } \lambda + 1 \geq \alpha |w_i| \geq \lambda \\ 0, & \text{if } \lambda > \alpha |w_i|, \end{cases}$$

where α is chosen such that $S(\alpha) = k$. Furthermore, the proximity operator can be computed in $\mathcal{O}(d \log d)$ time.

5. Spectral Norms

We now turn our focus to the matrix norms. For this purpose, we recall that a norm $\|\cdot\|$ on $\mathbb{R}^{d \times T}$ is called orthogonally invariant if $\|W\| = \|UWV\|$, for any orthogonal matrices $U \in \mathbb{R}^{d \times d}$ and $V \in \mathbb{R}^{T \times T}$. A classical result by Von Neumann (1937) establishes that a norm is orthogonally invariant if and only if it is of the form $\|W\| = g(\sigma(W))$, where $\sigma(W)$ is the vector formed by the singular values of W in nonincreasing order, and g is a symmetric gauge function, that is a norm which is invariant under permutations and sign changes of the vector components.

Algorithm 1 Computation of $x = \text{prox}_{\frac{\lambda}{2} \|\cdot\|_{\Theta, \text{box}}}(w)$.

Require: parameters a, b, c, λ .

1. Sort points $\{\alpha^i\}_{i=1}^{2d} = \left\{ \frac{a+\lambda}{|w_j|}, \frac{b+\lambda}{|w_j|} \right\}_{j=1}^d$ such that $\alpha^i \leq \alpha^{i+1}$;
2. Identify points α^i and α^{i+1} such that $S(\alpha^i) \leq c$ and $S(\alpha^{i+1}) \geq c$ by binary search;
3. Find α^* between α^i and α^{i+1} such that $S(\alpha^*) = c$ by linear interpolation;
4. Compute $\theta_i(\alpha^*)$ for $i = 1, \dots, d$;
5. Return $x_i = \frac{\theta_i w_i}{\theta_i + \lambda}$ for $i = 1, \dots, d$.

Lemma 17 *If Θ is a convex bounded subset of the strictly positive orthant in \mathbb{R}^d which is invariant under permutations, then $\|\cdot\|_{\Theta}$ is a symmetric gauge function.*

Proof Let $g(w) = \|w\|_{\Theta}$. We need to show that g is a norm which is invariant under permutations and sign changes. By Proposition 2, g is a norm, so it remains to show that $g(w) = g(Pw)$ for every permutation matrix P , and $g(Jw) = g(w)$ for every diagonal matrix J with entries ± 1 . The former property follows since the set Θ is permutation invariant. The latter property is true because the objective function in (1) involves the squares of the components of w . ■

In particular, this readily applies to both the k -support norm and the box-norm. We can therefore extend both norms to orthogonally invariant norms, which we term the spectral k -support norm and the spectral box-norm respectively, and which we write (with some abuse of notation) as $\|W\|_{(k)} = \|\sigma(W)\|_{(k)}$ and $\|W\|_{\text{box}} = \|\sigma(W)\|_{\text{box}}$. We note that since the k -support norm subsumes the ℓ_1 and ℓ_2 -norms for $k = 1$ and $k = d$ respectively, the corresponding spectral k -support norms are equal to the trace and Frobenius norms respectively.

A number of properties of the vector norms translate in the natural manner to the matrix norms. We first characterize the unit ball of the spectral k -support norm.

Proposition 18 *The unit ball of the spectral k -support norm is the convex hull of the set of matrices of rank at most k and Frobenius norm no greater than one.*

Proof For any $W \in \mathbb{R}^{d \times T}$, define the following sets

$$\bar{T}_k = \{W \in \mathbb{R}^{d \times T} : \text{rank}(W) \leq k, \|W\|_F \leq 1\}, \quad A_k = \text{co}(T_k),$$

and consider the following functional

$$\lambda(W) = \inf\{\lambda > 0 : W \in \lambda A_k\}, \quad W \in \mathbb{R}^{d \times T}. \quad (28)$$

We will apply Lemma 23 in the appendix to the set A_k . To do this, we need to show that the set A_k is bounded, convex, symmetric and absorbing. The first three are clearly satisfied. To see that it is absorbing, let $W \in \mathbb{R}^{d \times T}$ have singular value decomposition $U\Sigma V^T$, and let $r = \min(d, T)$. If W is zero then clearly $W \in A_k$, so assume it is non zero.

For $i \in \mathbb{N}_r$, let $S_i \in \mathbb{R}^{d \times T}$ have entry (i, j) equal to 1, and all remaining entries zero. We then have

$$W = U \Sigma V^T = U \left(\sum_{i=1}^r \sigma_i S_i \right) V^T = \left(\sum_{i=1}^d \sigma_i \right) \sum_{i=1}^r \frac{\sigma_i}{\sum_{j=1}^r \sigma_j} (U S_i V^T) =: \lambda \sum_{i=1}^r \beta_i Z_i.$$

Now for each i , $\|Z_i\|_F = \|S_i\|_F = 1$, and $\text{rank}(Z_i) = \text{rank}(S_i) = 1$, so $Z_i \in T_k$ for any $k \geq 1$. Furthermore $\beta_i \in [0, 1]$ and $\sum_{i=1}^r \beta_i = 1$, that is $(\beta_1, \dots, \beta_r) \in \Delta^{r-1}$, so $\frac{1}{\lambda} W$ is a convex combination of Z_i , in other words $W \in \lambda A_k$, and we have shown that A_k is absorbing. It follows that A_k satisfies the hypotheses of Lemma 23, where we let $C = A_k$, hence λ defines a norm on $\mathbb{R}^{d \times T}$ with unit ball equal to A_k .

Since the constraints in T_k involve spectral functions, the sets T_k and A_k are invariant to left and right multiplication by orthogonal matrices. It follows that λ is a spectral function, that is $\lambda(W)$ is defined in terms of the singular values of W . By von Neumann’s Theorem (Von Neumann, 1937) the norm it defines is orthogonally invariant and we have

$$\lambda(W) = \inf\{\lambda > 0 : W \in \lambda A_k\} = \inf\{\lambda > 0 : \sigma(W) \in \lambda C_k\} = \|\sigma(W)\|_{(k)}$$

where we have used Corollary 24, which states that C_k is the unit ball of the k -support norm. It follows that the norm defined by (28) is the spectral k -support norm with unit ball given by A_k . ■

Referring to the unit ball characterization of the k -support norm, we note that the restriction on the cardinality of the vectors which define the extreme points of the unit ball naturally extends to a restriction on the rank operator in the matrix setting. Furthermore, as noted by Argyrion et al. (2012), regularization using the k -support norm encourages vectors to be sparse, but less so than the l_1 -norm. In matrix regularization problems, Proposition 18 suggests that the spectral k -support norm for $k > 1$ encourages matrices to have low rank, but less so than the trace norm. This is intuitive as the extreme points of the unit ball have rank at most k .

As in the case of the vector norm (Proposition 8), the spectral box-norm (or cluster norm—see below) can be written as a perturbation of the spectral k -support norm with a quadratic term.

Proposition 19 *Let $\|\cdot\|_{\text{box}}$ be a matrix box-norm with parameters a, b, c and let $k = \frac{c-d}{b-a}$. Then*

$$\|W\|_{\text{box}}^2 = \min_{Z \in \mathbb{R}^{d \times T}} \left\{ \frac{1}{a} \|W - Z\|_F^2 + \frac{1}{b-a} \|Z\|_{(k)}^2 \right\}.$$

Proof By von Neumann’s trace inequality (Theorem 25 in the appendix) we have

$$\begin{aligned} \frac{1}{a} \|W - Z\|_F^2 + \frac{1}{b-a} \|Z\|_{(k)}^2 &= \frac{1}{a} (\|W\|_F^2 + \|Z\|_F^2 - 2\langle W, Z \rangle) + \frac{1}{b-a} \|Z\|_{(k)}^2 \\ &\geq \frac{1}{a} (\|\sigma(W)\|_2^2 + \|\sigma(Z)\|_2^2 - 2\langle \sigma(W), \sigma(Z) \rangle) + \frac{1}{b-a} \|\sigma(Z)\|_{(k)}^2 \\ &= \frac{1}{a} \|\sigma(W) - \sigma(Z)\|_2^2 + \frac{1}{b-a} \|\sigma(Z)\|_{(k)}^2. \end{aligned}$$

Furthermore the inequality is tight if W and Z have the same ordered set of singular vectors. Hence

$$\min_{Z \in \mathbb{R}^{d \times T}} \left\{ \frac{1}{a} \|W - Z\|_F^2 + \frac{1}{b-a} \|Z\|_{(k)}^2 \right\} = \min_{z \in \mathbb{R}^d} \left\{ \frac{1}{a} \|\sigma(W) - z\|_2^2 + \frac{1}{b-a} \|z\|_{(k)}^2 \right\} = \|\sigma(W)\|_{\text{box}}^2,$$

where the last equality follows by Proposition 8. ■

In other words, this result shows that the (scaled) squared spectral box-norm can be seen as the Moreau envelope of a squared spectral k -support norm.

5.1 Proximity Operator for Orthogonally Invariant Norms

The computational considerations outlined in Section 4 can be naturally extended to the matrix setting by using von Neumann’s trace inequality stated in the appendix. Here we comment on the computation of the proximity operator, which is important for our numerical experiments in Section 7. The proximity operator of an orthogonally invariant norm $\|\cdot\| = g(\sigma(\cdot))$ is given by

$$\text{prox}_{\|\cdot\|}(W) = U \text{diag}(\text{prox}_g(\sigma(W))) V^T, \quad W \in \mathbb{R}^{m \times T},$$

where U and V are the matrices formed by the left and right singular vectors of W (see e.g. Argyrion et al., 2011, Prop. 3.1). Using this result we can employ proximal gradient methods to solve matrix regularization problems using the square of the spectral k -support and box-norms.

6. Multitask Learning

In this section, we address multitask learning, a framework in which spectral regularizers have successfully been used to learn a set of regression or binary classification tasks. Within this setting each column of the matrix W represents one of the task weight vectors. By leveraging the commonalities between the tasks, learning can often be improved compared to solving each task in isolation (see e.g. Evgeniou et al., 2005; Argyrion et al., 2007, 2008; Jacob et al., 2009a; Cavallanti et al., 2010, and references therein). A natural assumption that arises in applications is that the tasks are clustered. The cluster norm was introduced by Jacob et al. (2009b) as a means to favour this structure. We show that this norm is equivalent to the spectral box-norm and then address the issue of centering the norm.

6.1 Clustering the Tasks

A general approach to multitask learning is based on the regularization problem

$$\min_{W \in \mathbb{R}^{d \times T}} \mathcal{L}(W) + \lambda \Omega(W)$$

where $W = [w_1, \dots, w_T]$ is the $d \times T$ matrix whose columns represent the task vectors, Ω is a regularizer which incorporates prior knowledge of sharing between tasks and \mathcal{L} is the compound empirical error. That is, $\mathcal{L}(W) = \frac{1}{Tn} \sum_{t=1}^T \sum_{i=1}^n \ell(y_i^t, \langle w_t, x_i^t \rangle)$ where

$(x_1^t, y_1^t), \dots, (x_n^t, y_n^t) \in \mathbb{R}^d \times \mathbb{R}$ are the training points for task t (for simplicity we assume that each task has the same number n of training points) and ℓ is a convex loss function.

Jacob et al. (2009a) consider a composite penalty which encourages the tasks to be clustered into $Q < T$ groups. To introduce their setting we require some more notation. Let $\mathcal{J}_q \subseteq \mathbb{N}_T$ be the set of tasks in cluster $q \in \mathbb{N}_Q$ and let $T_q = |\mathcal{J}_q| \geq 0$ be the number of tasks in cluster q , so that $\sum_{q=1}^Q T_q = T$. The clustering uniquely defines the $T \times T$ normalized connectivity matrix M where $M_{st} = \frac{1}{T_q}$ if $s, t \in \mathcal{J}_q$ and $M_{st} = 0$ otherwise. We let $\bar{w} = \frac{1}{T} \sum_{t=1}^T w_t$ be the mean weight vector, $\bar{w}_q = \frac{1}{T_q} \sum_{t \in \mathcal{J}_q} w_t$ be the mean weight vector of tasks in cluster q and define the $T \times T$ orthogonal projection matrices $U = \mathbb{1}^T / T$ and $\Pi = I - U$. Note that $W\Pi = [w_1 - \bar{w}, \dots, w_T - \bar{w}]$. Finally, let $r = \min(d, T)$.

Using this notation, we introduce the three seminorms

$$\begin{aligned} \Omega_m(W) &= T \|\bar{w}\|^2 = \text{tr}(WUW^\top) \\ \Omega_b(W) &= \sum_{q=1}^Q T_q \|\bar{w}_q - \bar{w}\|^2 = \text{tr}(W(M - U)W^\top) \\ \Omega_w(W) &= \sum_{q=1}^Q \sum_{t \in \mathcal{J}_q} \|w_t - \bar{w}_q\|^2 = \text{tr}(W(I - M)W^\top), \end{aligned}$$

each of which captures a different aspect of the clustering: Ω_m penalizes the total *mean* of the weight vectors, Ω_b measures how close to each other the clusters are (*between* cluster variance), and Ω_w measures the compactness of the clusters (*within* cluster variance). Scaling the three penalties by positive parameters ϵ_b , ϵ_b and ϵ_w respectively, we obtain the composite penalty $\epsilon_b \Omega_m + \epsilon_b \Omega_b + \epsilon_w \Omega_w$. The first term Ω_m does not depend on the connectivity matrix M , and it can be included in the error term. The remaining two terms depend on M , which in general may not be known *a-priori*. Jacob et al. (2009a) propose to learn the clustering by minimizing with respect to matrix M , under the assumption that $\epsilon_w \geq \epsilon_b$; this assumption is reasonable as we care more about enforcing a small variance of parameters within the clusters than between them. Using the elementary properties that $M - U = M\Pi = \Pi M\Pi$ and $I - M = (I - M)\Pi = \Pi(I - M)\Pi$ and letting $\tilde{M} = M\Pi$, we rewrite

$$\epsilon_b \Omega_b(W) + \epsilon_w \Omega_w(W) = \text{tr} \left(W\Pi(\epsilon_b \tilde{M} + \epsilon_w(I - \tilde{M}))\Pi W^\top \right) = \text{tr}(W\Pi\Sigma^{-1}\Pi W^\top) \quad (29)$$

where we have defined $\Sigma^{-1} = \epsilon_b \tilde{M} + \epsilon_w(I - \tilde{M})$. Since \tilde{M} is an orthogonal projection, the matrix Σ is well defined and we have

$$\Sigma = (\epsilon_b^{-1} - \epsilon_w^{-1})\tilde{M} + \epsilon_w^{-1}I. \quad (30)$$

The expression in the right hand side of equation (29) is jointly convex in W and Σ (see e.g. Boyd and Vandenberghe, 2004), however the set of matrices Σ defined by equation (30), generated by letting $\tilde{M} = M\Pi$ vary, is nonconvex, because M takes values on a nonconvex set. To address this, Jacob et al. (2009a) relax the constraint on matrix \tilde{M} to the set $\{0 \preceq \tilde{M} \preceq I, \text{tr} \tilde{M} \leq Q - 1\}$. This in turn induces the convex constraint set for Σ

$$\mathcal{S}_{Q,T} = \{\Sigma \in \mathbb{R}^{T \times T} : \Sigma = \Sigma^\top, \epsilon_w^{-1}I \preceq \Sigma \preceq \epsilon_b^{-1}I, \text{tr} \Sigma \leq (\epsilon_b^{-1} - \epsilon_w^{-1})(Q - 1) + \epsilon_w^{-1}T\}.$$

In summary Jacob et al. (2009a) arrive at the optimization problem

$$\min_{W \in \mathbb{R}^{d \times T}} \tilde{\mathcal{L}}(W) + \lambda \|W\Pi\|_c^2 \quad (31)$$

where $\tilde{\mathcal{L}}(W) = \mathcal{L}(W) + \lambda \epsilon_m \text{tr}(WUW^\top)$ and $\|\cdot\|_c$ is the *cluster norm* defined by the equation

$$\|W\|_c = \sqrt{\inf_{\Sigma \in \mathcal{S}_{Q,T}} \text{tr}(\Sigma^{-1}W^\top W)}. \quad (32)$$

6.2 The Cluster Norm and the Spectral Box-Norm

We now discuss the cluster norm in the context of the spectral box-norm. Jacob et al. (2009a) state that the cluster norm of W equals what we in this paper have termed the spectral box-norm, with parameters $a = \epsilon_w^{-1}$, $b = \epsilon_b^{-1}$ and $c = (T - Q + 1)\epsilon_w^{-1} + (Q - 1)\epsilon_b^{-1}$. Here we prove this fact. Denote by $\lambda_i(\cdot)$ the eigenvalues of a matrix which we write in non increasing order $\lambda_1(\cdot) \geq \lambda_2(\cdot) \geq \dots \geq \lambda_d(\cdot)$. Note that if θ_i are the eigenvalues of Σ then $\theta_i = \lambda_{d-i+1}(\Sigma^{-1})$. We have that

$$\text{tr}(\Sigma^{-1}W^\top W) \geq \sum_{i=1}^r \lambda_{d-i+1}(\Sigma^{-1}) \lambda_i(W^\top W) = \sum_{i=1}^r \frac{\sigma_i^2(W)}{\theta_i}$$

where the inequality follows by Lemma 26 (stated in the appendix) for $A = \Sigma^{-1}$ and $B = W^\top W \succeq 0$. Since this inequality is attained whenever $\Sigma = U \text{diag}(\theta) U$, where U are the eigenvectors of $W^\top W$, we see that the cluster norm coincides with the spectral box-norm, that is $\|W\|_c = \|\sigma(W)\|_\Theta$ for $\Theta = \{\theta \in [a, b]^r : \sum_{i=1}^r \theta_i \leq c\}$. In light of our observations in Section 5, we also see that the spectral k -support norm is a special case of the cluster norm, where we let a tend to zero, $b = 1$ and $c = k$, where $k = Q - 1$. More importantly the cluster norm is a perturbation of the spectral k -support norm. Moreover, the methods to compute the norm and its proximity operator (cf. Theorems 11 and 15) can directly be applied to the cluster norm using von Neumann's trace inequality (see Theorem 25 in the appendix).

6.3 Optimization with Centered Spectral Θ -Norms

Centering a matrix has been shown to improve learning in other multitask learning problems, for example Evgeniou et al. (2005) reported improved results using the trace norm. It is therefore valuable to address the problem of how to solve a regularization problem of the type

$$\min_{W \in \mathbb{R}^{d \times T}} \tilde{\mathcal{L}}(W) + \lambda \|W\Pi\|_\Theta^2 \quad (33)$$

in which the regularizer is applied to the matrix $W\Pi = [w_1 - \bar{w}, \dots, w_T - \bar{w}]$. To this end, let Θ be a bounded and convex subset of \mathbb{R}_{++}^r which is invariant under permutation. We have already noted that the function defined, for every $W \in \mathbb{R}^{d \times T}$, as

$$\|W\|_\Theta := \|\sigma(W)\|_\Theta,$$

is an orthogonally invariant norm. In particular, problem (33) includes regularization with the centered cluster norm outlined above.

Note that right multiplication by the centering operator Π , is invariant to a translation of the columns of the matrix by a fixed vector, that is, for every $z \in \mathbb{R}^d$, we have $[w_1 + z, \dots, w_T + z]\Pi = W\Pi$. The quadratic term $\epsilon_m \text{tr}(WUW^T)$, which is included in the error, implements square norm regularization of the mean of the tasks, which can help to prevent overfitting. However, in the remainder of this section this term plays no role in the analysis, which equally applies to the case that $\epsilon_m = 0$.

In order to solve the problem (33) with a centered regularizer the following lemma is key.

Lemma 20 *Let $r = \min(d, T)$ and let Θ be a bounded and convex subset of \mathbb{R}^r_{++} which is invariant under permutation. For every $W = [w_1, \dots, w_T] \in \mathbb{R}^{d \times T}$, it holds that*

$$\|W\Pi\|_{\Theta} = \min_{z \in \mathbb{R}^d} \|[w_1 - z, \dots, w_T - z]\|_{\Theta}.$$

Proof Given the set Θ we define the set $\Theta^{(T)} = \{\Sigma \in \mathbf{S}^T_{++}, \lambda(\Sigma) \in \Theta\}$ and $\Theta^{(d)} = \{D \in \mathbf{S}^{d}_{++} : \lambda(D) \in \Theta\}$. It follows from Lemma 26 that

$$\|W\|_{\Theta}^2 = \|\sigma(W)\|_{\Theta}^2 = \inf_{\Sigma \in \Theta^{(T)}} \text{tr}(\Sigma^{-1}W^TW) = \inf_{D \in \Theta^{(d)}} \text{tr}(D^{-1}W^TW).$$

Using the second identity and recalling that $W\Pi = [w_1 - \bar{w}, \dots, w_T - \bar{w}]$, we have that

$$\begin{aligned} \|W\Pi\|_{\Theta}^2 &= \inf_{D \in \Theta^{(d)}} \text{tr}((W\Pi)^TD^{-1}(W\Pi)) \\ &= \inf_{D \in \Theta^{(d)}} \sum_{t=1}^T (w_t - \bar{w})^TD^{-1}(w_t - \bar{w}) = \inf_{D \in \Theta^{(d)}} \min_{z \in \mathbb{R}^d} \sum_{t=1}^T (w_t - z)^TD^{-1}(w_t - z) \end{aligned}$$

where in the last step we used the fact that the quadratic form $\sum_{t=1}^T (w_t - z)^TD^{-1}(w_t - z)$ is minimized at $z = \bar{w}$. The result now follows by interchanging the infimum and the minimum in the last expression and using the definition of the Θ -norm. ■

Using this lemma, we rewrite problem (31) as

$$\begin{aligned} \min_{W \in \mathbb{R}^{d \times T}} \min_{z \in \mathbb{R}^d} \tilde{\mathcal{L}}(W) + \lambda \|[w_1 - z, \dots, w_T - z]\|_{\Theta}. \\ \min_{(V, z) \in \mathbb{R}^{d \times T} \times \mathbb{R}^d} \tilde{\mathcal{L}}(V + z\mathbf{1}^T) + \lambda \|V\|_{\Theta}. \end{aligned} \quad (34)$$

Letting $v_t = w_t - z$, and $V = [v_1, \dots, v_T]$, we obtain the equivalent problem

This problem is of the form $f(V, z) + \lambda g(V, z)$, where $g(V, z) = \|V\|_{\Theta}$. Using this formulation, we can directly apply the proximal gradient method using the proximity operator computation for the vector norm, since $\text{prox}_{g, \|\cdot\|_{\Theta}}(V_0, z_0) = (\text{prox}_{\|\cdot\|_{\Theta}}(V_0), z_0)$. This observation establishes that, whenever the proximity operator of the spectral Θ -norm is available, we can use proximal gradient methods with minimal additional effort to perform optimization with the corresponding centered spectral Θ -norm. For example, this is the case with the trace norm, the spectral k -support norm and the spectral box-norm or cluster norm.

7. Numerical Experiments

Argyriou et al. (2012) demonstrated the good estimation properties of the vector k -support norm compared to the Lasso and the elastic net. In this section, we investigate the matrix norms and report on their statistical performance in matrix completion and multitask learning experiments on simulated as well as benchmark real data sets. We also offer an interpretation of the role of the parameters in the box-norm and we empirically verify the improved performance of the proximity operator computation of Algorithm 1 (see Table 1).

We compare the spectral k -support norm (k -sup) and the spectral box-norm (box) to the baseline trace norm (*trace*) (see e.g. Argyriou et al., 2007; Mazumder et al., 2010; Srebro et al., 2005; Toh and Yun, 2011), matrix elastic net (*elnet*) (Li et al., 2012) and, in the case of multitask learning, the Frobenius norm (fr), which we recall is equivalent to the spectral k -support norm when $k = d$. As we highlighted in Section 6.3, centering a matrix can lead to improvements in learning. For data sets which we expect to exhibit clustering we therefore also apply centered versions of the norms, c -fr, c -trace, c -elnet, c - k -sup, c -box.²

We report test error and standard deviation, matrix rank (r) and optimal parameter values for k and α , which are determined by validation. We used a t -test to determine the statistical significance of the difference in performance between the regularizers, at a level of $p < 0.001$.

To solve the optimization problem we used an accelerated proximal gradient method (FISTA), (see e.g. Beck and Teboulle, 2009; Nesterov, 2007), using the percentage change in the objective as convergence criterion, with a tolerance of 10^{-5} (10^{-3} for real matrix completion experiments).

As is typical with spectral regularizers such as the trace norm, we found that the spectrum of the learned matrix exhibited a rapid decay to zero. In order to explicitly impose a low rank on the final matrix, we included a thresholding step at the end of the optimization. For the matrix completion experiments, the thresholding level was chosen by validation. Matlab code used in the experiments is available at <http://www0.cs.ucl.ac.uk/staff/M.Pontil/software.html>.

7.1. Simulated Data

Matrix Completion. We applied the norms to matrix completion on noisy observations of low rank matrices. Each $d \times d$ matrix was generated as $W = AB^T + E$, where $A, B \in \mathbb{R}^{d \times r}$, $r \ll d$, and the entries of A, B and E were set to be i.i.d. standard Gaussian. We set $d = 100$, $r \in \{5, 10\}$ and we sampled uniformly a percentage $\rho \in \{10\%, 20\%, 30\%\}$ of the entries for training, and used a fixed 10% for validation. Following Mazumder et al.

2. As we described in Section 6.2, the cluster norm regularization problem from Jacob et al. (2009a) is equivalent to regularization using the box-norm with a squared ℓ_2 norm of the mean column vector included in the loss function. The centering operator is invariant to constant shifts of the columns, which allows the matrix to have unbounded Frobenius norm when using a centered regularizer. The additional quadratic term regulates this effect and can prevent against overfitting. We tested the effect of the quadratic term on the centered norms, however the impact on performance was only incremental, and it introduced a further parameter requiring validation. On the real data sets in particular, the impact was not significant compared to simple centering, so we do not report on the method below.

d	1,000	2,000	4,000	8,000	16,000	32,000
Algorithm 1	0.0011	0.0016	0.0026	0.0046	0.0101	0.0181
Algorithm 2	0.0443	0.1567	0.5907	2.3065	9.0080	35.6199

Table 1: Comparison of proximity operator algorithms for the k -support norm (time in seconds), $k = 0.05d$. Algorithm 1 is our method, Algorithm 2 is the method in Argyriou et al. (2012).

(2010) the error was measured as

$$\text{error} = \frac{\|w_{\text{true}} - w_{\text{predicted}}\|^2}{\|w_{\text{true}}\|^2},$$

and averaged over 100 trials. The results are summarized in Table 2. With thresholding, all methods recovered the rank of the true noiseless matrix. The spectral box-norm generated the lowest test errors in all regimes, with the spectral k -support norm a close second, and both were significantly better than trace and elastic net.

data set	norm	test error	r	k	a	test error	r	k	a
rank 5 $\rho=10\%$	trace	0.8184 (0.03)	20	-	-	0.7799 (0.04)	5	-	-
	el.net	0.8164 (0.03)	20	-	-	0.7794 (0.04)	5	-	-
	k-sup	0.8036 (0.03)	16	3.6	-	0.7728 (0.04)	5	4.23	-
	box	0.7805 (0.03)	87	2.9	1.7e-2	0.7649 (0.04)	5	3.63	8.1e-3
rank 5 $\rho=15\%$	trace	0.5764 (0.04)	22	-	-	0.5209 (0.04)	5	-	-
	el.net	0.5744 (0.04)	21	-	-	0.5203 (0.04)	5	-	-
	k-sup	0.5659 (0.03)	18	3.3	-	0.5099 (0.04)	5	3.25	-
	box	0.5525 (0.04)	100	1.3	9e-3	0.5089 (0.04)	5	3.36	2.7e-3
rank 5 $\rho=20\%$	trace	0.4085 (0.03)	23	-	-	0.3449 (0.02)	5	-	-
	el.net	0.4081 (0.03)	23	-	-	0.3445 (0.02)	5	-	-
	k-sup	0.4031 (0.03)	21	3.1	-	0.3381 (0.02)	5	2.97	-
	box	0.3898 (0.03)	100	1.3	9e-3	0.3380 (0.02)	5	3.28	1.9e-3
rank 10 $\rho=20\%$	trace	0.6356 (0.03)	27	-	-	0.6084 (0.03)	10	-	-
	el.net	0.6359 (0.03)	27	-	-	0.6074 (0.03)	10	-	-
	k-sup	0.6284 (0.03)	24	4.4	-	0.6000 (0.03)	10	5.02	-
	box	0.6243 (0.03)	89	1.8	9e-3	0.6000 (0.03)	10	5.22	1.9e-3
rank 10 $\rho=30\%$	trace	0.3642 (0.02)	36	-	-	0.3086 (0.02)	10	-	-
	el.net	0.3638 (0.02)	36	-	-	0.3082 (0.02)	10	-	-
	k-sup	0.3579 (0.02)	33	5.0	-	0.3025 (0.02)	10	5.13	-
	box	0.3486 (0.02)	100	2.5	9e-3	0.3025 (0.02)	10	5.16	3e-4

Table 2: Matrix completion on simulated data sets, without (left) and with (right) thresholding.

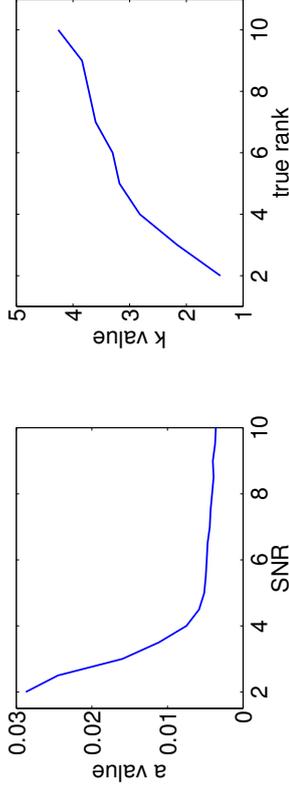


Figure 5: Impact of signal to noise ratio on value of a .

Figure 6: Impact of matrix rank on value of k .

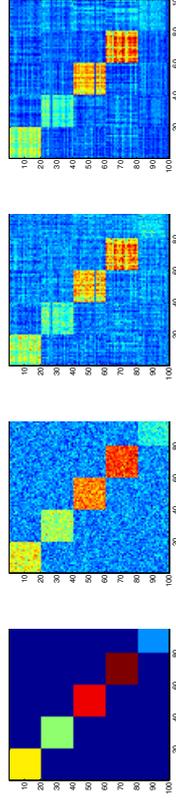


Figure 7: Clustered matrix and recovered solutions. From left to right: true, noisy, trace norm, box-norm.

Role of Parameters. In the same setting we investigated the role of the parameters in the box-norm. As previously discussed, parameter b can be set to 1 without loss of generality. Figure 5 shows the optimal value of parameter a chosen by validation for varying signal to noise ratios (SNR), keeping k fixed. We see that for greater noise levels (smaller SNR), the optimal value for a increases, which further suggests that the noise is filtered out by higher values of the parameter. Figure 6 shows the optimal value of k chosen by validation for matrices with increasing rank, keeping a fixed, and using the relation $k = \frac{a-d}{b-a}$. We note that as the rank of the matrix increases, the optimal k value increases, which is expected since it is an upper bound on the sum of the singular values.

Clustered Learning. We tested the centered norms on a synthetic data set which exhibited a clustered structure. We generated a 100×100 , rank 5, block diagonal matrix, where the entries of each 20×20 block were set to a random integer chosen uniformly in $\{1, \dots, 10\}$, with additive noise. Table 3 illustrates the results averaged over 100 runs. Within each group of norms, the box-norm and the k -support norm outperformed the trace norm and elastic net, and centering improved performance for all norms. Figure 7 illustrates a sample matrix along with the solution found using the box and trace norms.

data set	norm	test error	r	k	a	test error	r	k	a	
$\rho=10\%$	trace	0.6529 (0.10)	20	-	-	0.6065 (0.10)	5	-	-	
	el-net	0.6482 (0.10)	20	-	-	0.6037 (0.10)	5	-	-	
	k-sup	0.6354 (0.10)	19	2.72	-	0.5950 (0.10)	5	2.77	-	
	box	0.6182 (0.09)	100	2.23	1.9e-2	0.5881 (0.10)	5	2.73	4.3e-3	
	c-trace	0.5959 (0.07)	15	-	-	0.5692 (0.07)	5	-	-	
	c-el-net	0.5910 (0.07)	14	-	-	0.5670 (0.07)	5	-	-	
	c-k-sup	0.5837 (0.07)	14	2.03	-	0.5610 (0.07)	5	1.98	-	
	c-box	0.5789 (0.07)	100	1.84	1.9e-3	0.5581 (0.07)	5	1.93	9.7e-3	
	$\rho=15\%$	trace	0.3482 (0.08)	21	-	-	0.3048 (0.07)	5	-	-
		el-net	0.3473 (0.08)	21	-	-	0.3046 (0.07)	5	-	-
k-sup		0.3438 (0.07)	21	2.24	-	0.3007 (0.07)	5	2.89	-	
box		0.3431 (0.07)	100	2.05	8.7e-3	0.3005 (0.07)	5	2.57	1.3e-3	
c-trace		0.3225 (0.07)	19	-	-	0.2932 (0.06)	5	-	-	
c-el-net		0.3215 (0.07)	18	-	-	0.2931 (0.06)	5	-	-	
c-k-sup		0.3179 (0.07)	18	1.89	-	0.2883 (0.06)	5	2.36	-	
c-box		0.3174 (0.07)	100	1.90	2.2e-3	0.2876 (0.06)	5	1.92	3.8e-3	

Table 3: Clustered block diagonal matrix, before (left) and after (right) thresholding.

7.2 Real Data

Matrix Completion (MovieLens and Jester). In this section we report on the performance of the norms on real data sets. We observe a subset of the (user, rating) entries of a matrix and the task is to predict the unobserved ratings, with the assumption that the true matrix is low rank (or approximately low rank). In the first instance we considered the MovieLens data sets³. These consist of user ratings of movies, the ratings are integers from 1 to 5, and all users have rated a minimum number of 20 films. Specifically we considered the following data sets:

- *MovieLens 100k*: 943 users and 1,682 movies, with a total of 100,000 ratings;
- *MovieLens 1M*: 6,040 users and 3,900 movies, with a total of 1,000,209 ratings.

We also considered the Jester⁴ data sets, which consist of user ratings of jokes, where the ratings are real values from -10 to 10 :

- *Jester 1*: 24,983 users and 100 jokes, all users have rated a minimum of 36 jokes;
- *Jester 2*: 23,500 users and 100 jokes, all users have rated a minimum of 36 jokes;
- *Jester 3*: 24,938 users and 100 jokes, all users have rated between 15 and 35 jokes.

Following Toh and Yun (2011), for MovieLens we uniformly sampled $\rho = 50\%$ of the available entries for each user for training, and for Jester 1, Jester 2 and Jester 3 we sampled

3. MovieLens data sets are available at <http://grouplens.org/datasets/movielens/>.

4. Jester data sets are available at <http://goldberg.berkeley.edu/jester-data/>.

data set	norm	test error	r	k	a	test error	r	k	a
MovieLens 100k	trace	0.2034	87	-	-	0.2017	13	-	-
	el-net	0.2034	87	-	-	0.2017	13	-	-
	k-sup	0.2031	102	1.00	-	0.1990	9	1.87	-
$\rho = 50\%$	box	0.2035	943	1.00	1e-5	0.1989	10	2.00	1e-5
	trace	0.1821	325	-	-	0.1790	17	-	-
	el-net	0.1821	319	-	-	0.1789	17	-	-
MovieLens 1M	el-net	0.1820	317	1.00	-	0.1782	17	1.80	-
	k-sup	0.1817	3576	1.09	3e-5	0.1777	19	2.00	1e-6
	box	0.1817	3576	1.09	3e-5	0.1777	19	2.00	1e-6
Jester 1	trace	0.1787	98	-	-	0.1752	11	-	-
	el-net	0.1787	98	-	-	0.1752	11	-	-
	k-sup	0.1764	84	5.00	-	0.1739	11	6.38	-
Jester 2	box	0.1766	100	4.00	1e-6	0.1726	11	6.40	2e-5
	trace	0.1767	98	-	-	0.1758	11	-	-
	el-net	0.1767	98	-	-	0.1758	11	-	-
Jester 3	k-sup	0.1762	94	4.00	-	0.1746	11	4.00	-
	box	0.1762	100	4.00	2e-6	0.1745	11	4.50	5e-5
	trace	0.1988	49	-	-	0.1959	3	-	-
8 per line	el-net	0.1988	49	-	-	0.1959	3	-	-
	k-sup	0.1970	46	3.70	-	0.1942	3	2.13	-
	box	0.1973	100	5.91	1e-3	0.1940	3	4.00	8e-4

Table 4: Matrix completion on real data sets, without (left) and with (right) thresholding.

20, 20 and 8 ratings per user respectively, and we again used 10% for validation. The error was measured as normalized mean absolute error,

$$\text{NMAE} = \frac{\|w_{\text{true}} - w_{\text{predicted}}\|_2}{\#\text{observations}/(r_{\text{max}} - r_{\text{min}})},$$

where r_{max} and r_{min} are upper and lower bounds for the ratings (Toh and Yun, 2011), averaged over 50 runs. The results are outlined in Table 4. In the thresholding case, the spectral box-norm and the spectral k -support norm showed the best performance, and in the absence of thresholding, the spectral k -support norm showed slightly improved performance. Comparing to the synthetic data sets, this suggests that the parameter a did not provide any benefit in the absence of noise. We also note that without thresholding our results for trace norm regularization on MovieLens 100k agreed with those in Jaggi and Sultowsky (2010).

Multitask Learning (Lentk and Animals with Attributes). In our final set of experiments we considered two multitask learning data sets, where we expected the data to exhibit clustering. The *Lentk personal computer* data set (Lentk et al., 1996) consists of 180 ratings of 20 profiles of computers characterized by 14 features (including a bias term). The clustering is suggested by the assumption that users are motivated by similar groups of features. We used the root mean square error of true vs. predicted ratings, normalised over the tasks, averaged over 100 runs. We also report on the Frobenius norm, which in the multitask

norm	test error	k	a
fr	3.7931 (0.07)	-	-
trace	1.9056 (0.04)	-	-
el.net	1.9007 (0.04)	-	-
k-sup	1.8955 (0.04)	1.02	-
box	1.8923 (0.04)	1.01	5.5e-3
c-fr	1.8634 (0.08)	-	-
c-trace	1.7902 (0.03)	-	-
c-el.net	1.7897 (0.03)	-	-
c-k-sup	1.7777 (0.03)	1.89	-
c-box	1.7759 (0.03)	1.12	8.6e-3

Table 5: Multitask learning clustering on Lenk data set.

learning framework corresponds to independent task learning. The results are outlined in Table 5. The centered versions of the spectral k -support norm and spectral box-norm outperformed the other penalties in all regimes. Furthermore, the results clearly indicate the importance of centering, as discussed for the trace norm in Evgeniou et al. (2007).

The *Animals with Attributes* data set (Lampert et al., 2009) consists of 30,475 images of animals from 50 classes. Along with the images, the data set includes pre-extracted features for each image. The data set has been analyzed in the context of multitask learning. We followed the experimental protocol from (Kang et al., 2011), however we used an updated feature set, and we considered all 50 classes. Specifically, we used the DeCAF feature set provided by Lampert et al. (2009) rather than the SIFT bag of word descriptors. These updated features were obtained through a deep convolutional network and represent each image by a 4,096-dimensional vector (Donahue et al., 2014). As the smallest class size is 92 we selected the first $n = 92$ examples of each of the $T = 50$ classes, used PCA (with centering) on the resulting data matrix to reduce dimensionality ($d = 1, 718$) retaining a variance of 95%, and obtained a data set of size $4,600 \times 1, 718$. For each class the examples were split into training, validation and testing data sets, with a split of 50%, 25%, 25% respectively, and we averaged the performance over 50 runs.

We used the logistic loss, yielding the error term

$$\mathcal{L}(W) = \sum_{t=1}^T \sum_{i=1}^{Tn} \log(1 + \exp(-y_{t,i}(w_t, x_i)))$$

where $W = [w_1, \dots, w_T]$, x_1, \dots, x_{Tn} are the inputs and $y_{t,i} = 1$ if x_i is in class t , and $y_{t,i} = -1$ otherwise.

The predicted class for testing example x was $\operatorname{argmax}_{t=1}^T \langle w_t, x \rangle$ and the accuracy was measured as the percentage of correctly classified examples, also known as multi-class error. The results without centering are outlined in Table 6. The corresponding results with centering showed the same relative performance, but worse overall accuracy, which is reasonable as the data is not expected to be clustered, and we omit the results here.

The spectral k -support and box-norms gave the best results, outperforming the Frobenius norm and the matrix elastic net, which in turn outperformed the trace norm. We highlight that in contrast to the Lenk experiments, the Frobenius norm, corresponding to

norm	test error	k	a
fr	38.3428 (0.74)	-	-
tr	37.4285 (0.76)	-	-
el.net	38.2857 (0.73)	-	-
k-sup	38.8571 (0.71)	37.8	-
box	38.9100 (0.65)	32.8	2.1e-2

Table 6: Multitask learning clustering on Animals with Attributes data set, no centering.

independent task learning, was competitive. Furthermore, the optimal values of k for the spectral k -support norm and spectral box-norm were high (38 and 33, respectively) relative to the maximum rank of 50, corresponding to a relatively high rank solution. The spectral k -support norm and spectral box-norm nonetheless outperformed the other regularizers. Notice also that the spectral k -support norm requires the same number of parameters to be tuned as the matrix elastic net, which suggests that it somehow captures the underlying structure of the data in a more appropriate manner.

We finally note as an aside that using the SIFT bag of words descriptors provided by Lampert et al. (2009), which represent the images as a 2,000-dimensional histogram of local features, we replicated the results for independent task learning (Frobenius norm regularization) and single-group learning (trace norm regularization) of Kang et al. (2011) for the subset of 20 classes considered in their paper.

8. Extensions

In this section we outline a number of extensions to topics in this paper.

8.1 k -Support p -Norms

A natural extension of the k -support norm follows by applying a p -norm, rather than the Euclidean norm, in the infimum convolution definition of the k -support norm. In the dual norm, we then obtain the corresponding q -norm, where $\frac{1}{p} + \frac{1}{q} = 1$.

Definition 21 The k -support p -norm is defined for $w \in \mathbb{R}^d$ as

$$\|w\|_{(k,p)} = \inf \left\{ \sum_{g \in \mathcal{G}_k} \|v_g\|_p : \operatorname{supp}(v_g) \subseteq g, \sum_{g \in \mathcal{G}_k} v_g = w \right\}. \quad (35)$$

The following corollary follows along the same lines as the proof of Proposition 3 in the appendix.

Corollary 22 The (p, k) -support norm is well defined and its unit ball is the convex hull of the set $\{w \in \mathbb{R}^d : \|w\|_p \leq 1, \operatorname{card}(w) \leq k\}$. Furthermore, its dual norm is given by

$$\|u\|_{*(k,q)} = \left(\sum_{i=1}^k (|u_i|^q) \right)^{\frac{1}{q}}, \quad u \in \mathbb{R}^d.$$

We discuss the special cases $p \in \{1, 2, \infty\}$. The case $p = 2$ is the k -support norm of Agrignon et al. (2012) discussed above. For $p = 1$ we have $\|u\|_{*(k,q)} = \|u\|_\infty$, hence the $(k, 1)$ -support norm coincides with the ℓ_1 norm for every $k \in \mathbb{N}_d$. The case $p = \infty$ is more interesting; specifically the dual norm is the well-known Ky-Fan norm (see e.g. Bhatia, 1997).

$$\|u\|_{*(k,\infty)} = \sum_{i=1}^k |u_i|.$$

Using the fact that the primal norm is the dual of the dual, we obtain by a direct computation that

$$\|u\|_{(k,\infty)} = \max \left(\|u\|_\infty, \frac{1}{k} \|u\| \right).$$

It is clear that the (p, k) -support norm is a symmetric gauge function. Hence we can define the spectral (p, k) -support norm as $\|W\|_{(k,p)} = \|\sigma(W)\|_{(k,p)}$, for $W \in \mathbb{R}^{d \times T}$. Since the dual of any orthogonally invariant norm is given by $\|\cdot\|_* = \|\sigma(\cdot)\|_*$ (see e.g. Lewis, 1995), we conclude that the dual spectral (k, p) -support norm is given by $\|U\|_{*(k,p)} = \|\sigma(U)\|_{*(k,p)}$, for every $U \in \mathbb{R}^{d \times T}$. Furthermore, the unit ball of the spectral (p, k) -support norm is equal to the convex hull of the set $\{W \in \mathbb{R}^{d \times T} : \text{rank}(W) \leq k, \|\sigma(W)\|_p \leq 1\}$.

8.2 Kernels

The ideas discussed in this paper can be used in the context of multiple kernel learning in a natural way (see e.g. Micchelli and Pontil, 2007, and references therein). Let $K_j, j \in \mathbb{N}_s$, be prescribed reproducing kernels on a set X , and H_j the corresponding reproducing kernel Hilbert spaces with norms $\|\cdot\|_j$. We consider the problem

$$\min \left\{ \sum_{i=1}^n \ell \left(\psi_i, \sum_{\ell=1}^s f_\ell(x_i) \right) + \lambda \left(\|f_1\|_1, \dots, \|f_s\|_s \right) \middle| \Theta : f_1 \in H_1, \dots, f_s \in H_s \right\}.$$

The choice $\Theta = \{\theta \in \mathbb{R}^d : 0 < \theta_i \leq 1, \sum_{i=1}^d \theta_i \leq k\}$, when $k \leq s$, is particularly interesting. It gives rise to a version of multiple kernel learning in which at least k kernels are employed.

8.3 Rademacher Complexity

We briefly comment on the Rademacher complexity of the spectral k -support norm, namely

$$\mathbb{E} \sup_{\|a\|_0 \leq 1} \frac{1}{Tn} \sum_{t=1}^T \sum_{i=1}^n \epsilon_t^i \langle w_t, x_t^i \rangle$$

where the expectations is taken with respect to i.i.d. Rademacher random variables $\epsilon_t^i, i \in \mathbb{N}_n, t \in \mathbb{N}_T$ and the x_t^i are either prescribed or random datapoints associated with the different regression tasks. The Rademacher complexity can be used to derive uniform bounds on the estimation error and excess risk bounds (see Bartlett and Mendelson, 2002; Koltchinskii and Panchenko, 2002, for a discussion). Although a complete analysis is beyond

the scope of the present paper, we remark that the Rademacher complexity of the unit ball of the spectral k -support is a factor of \sqrt{k} larger than the Rademacher complexity bound for the trace norm provided in (Proposition 6 Maurer and Pontil, 2013). This follows from the fact that the dual spectral k -support norm is bounded by \sqrt{k} times the operator norm. Of course the unit ball of the spectral k -support norm contains the unit ball of the trace norm, so the associated excess risk bounds need to be compared with care.

9. Conclusion

We studied the family of box-norms, and showed that the k -support norm belongs to this family. We noted that these can be naturally extended from the vector to the matrix setting. We also provided a connection between the k -support norm and the cluster norm, which essentially coincides with the spectral box-norm. We further observed that the cluster norm is a perturbation of the spectral k -support norm, and we were able to compute the norm and the proximity operator of the squared norm. We also provided a method to solve regularization problems using centered versions of the norms and we considered a number of extensions to the box-norm framework.

Our experiments indicate that the spectral box-norm and k -support norm consistently outperform the trace norm and the matrix elastic net on various matrix completion problems. Furthermore, we studied the application of centering to clustering problems in multi-task learning, and found that this improved performance. With a single parameter, compared to two for the spectral box-norm, and three for the cluster norm, our results suggest that the spectral k -support norm represents a powerful yet straightforward alternative to the trace norm for low rank matrix learning. In future work we would like to complete the analysis of the Rademacher complexity for the norms in this paper, and derive associated statistical oracle inequalities. We would also like to investigate the family of Θ -norms for more general parameter sets.

Acknowledgments

We would like to thank Andreas Maurer, Charles Micchelli and especially Andreas Agrignon for useful discussions. This work was supported in part by EPSRC Grant EP/H027203/1.

Appendix A.

In this appendix, we discuss some auxiliary results which are used in the main body of the paper.

Let X be a finite dimensional vector space. Recall that a subset C of X is called *balanced* if $\alpha C \subseteq C$ whenever $|\alpha| \leq 1$. Furthermore, C is called *absorbing* if for any $x \in X, x \in \lambda C$ for some $\lambda > 0$.

Lemma 23 *Let $C \subseteq X$ be a bounded, convex, balanced, and absorbing set. The Minkowski functional μ_C of C , defined for every $w \in X$, as*

$$\mu_C(w) = \inf \left\{ \lambda : \lambda > 0, \frac{1}{\lambda} w \in C \right\}$$

is a norm on X .

Proof We give a direct proof that μ_C satisfies the properties of a norm. See also e.g. (Rudin, 1991, §1.35) for further details. Clearly $\mu_C(w) \geq 0$ for all w , and $\mu_C(0) = 0$. Moreover, as C is bounded, $\mu_C(w) > 0$ whenever $w \neq 0$.

Next we show that μ_C is one-homogeneous. For every $\alpha \in \mathbb{R}$, $\alpha \neq 0$, let $\sigma = \text{sign}(\alpha)$ and note that

$$\begin{aligned} \mu_C(\alpha w) &= \inf \left\{ \lambda > 0 : \frac{1}{\lambda} \alpha w \in C \right\} \\ &= \inf \left\{ \lambda > 0 : \frac{|\alpha|}{\lambda} \sigma w \in C \right\} \\ &= |\alpha| \inf \left\{ \lambda > 0 : \frac{1}{\lambda} w \in \sigma C \right\} \\ &= |\alpha| \inf \left\{ \lambda > 0 : \frac{1}{\lambda} w \in C \right\} = |\alpha| \mu_C(w), \end{aligned}$$

where we have made a change of variable and used the fact that $\sigma C = C$.

Finally, we prove the triangle inequality. For every $v, w \in X$, if $v/\lambda \in C$ and $w/\mu \in C$ then setting $\gamma = \lambda/(\lambda + \mu)$, we have

$$\frac{v+w}{\lambda+\mu} = \gamma \frac{v}{\lambda} + (1-\gamma) \frac{w}{\mu}$$

and since C is convex, then $\frac{v+w}{\lambda+\mu} \in C$. We conclude that $\mu_C(v+w) \leq \mu_C(v) + \mu_C(w)$. The proof is completed. \blacksquare

Note that for such set C , the unit ball of the induced norm μ_C is C . Furthermore, if $\|\cdot\|$ is a norm then its unit ball satisfies the hypotheses of Lemma 23.

Using this lemma we can prove Proposition 3.

Proof of Proposition 3 Let $A_\ell = \{w \in \mathbb{R}^d : \|w\|_{\gamma^\ell} \leq 1, \text{supp}(w) \subseteq \text{supp}(\gamma^\ell)\}$, and define

$$C = \text{co} \bigcup_{\ell=1}^m A_\ell.$$

Note that C is bounded and balanced, since each set A_ℓ is so. Furthermore, the hypothesis that $\sum_{\ell=1}^m \gamma^\ell > 0$ ensures that C is absorbing. Hence, by Lemma 23 the Minkowski functional μ_C defines a norm. We rewrite $\mu_C(w)$ as

$$\mu_C(w) = \inf \left\{ \lambda : \lambda > 0, w = \lambda \sum_{\ell=1}^m \alpha_\ell z_\ell, z_\ell \in A_\ell, \alpha \in \Delta^{m-1} \right\}$$

where the infimum is over λ , the vectors $z_\ell \in \mathbb{R}^d$ and the vector $\alpha = (\alpha_1, \dots, \alpha_m)$, and recall Δ^{m-1} denotes the unit simplex in \mathbb{R}^m .

The rest of the proof is structured as follows. We first show that $\mu_C(w)$ coincides with the right hand side of equation (3), which we denote by $\nu(w)$. Then we show that $\|w\|_\Theta = \mu_C(w)$ by observing that both norms have the same dual norm.

Choose any vectors $v_1, \dots, v_m \in \mathbb{R}^d$ which satisfies the constraint set in the right hand side of (3) and set $\alpha_\ell = \|v_\ell\|_{\gamma^\ell} / (\sum_{k=1}^m \|v_k\|_{\gamma^k})$ and $z_\ell = v_\ell / \|v_\ell\|_{\gamma^\ell}$. We have

$$w = \sum_{\ell=1}^m v_\ell = \left(\sum_{k=1}^m \|v_k\|_{\gamma^k} \right) \sum_{\ell=1}^m \alpha_\ell z_\ell.$$

This implies that $\mu_C(w) \leq \nu(w)$. Conversely, if $w = \lambda \sum_{\ell=1}^m \alpha_\ell z_\ell$ for some $z_\ell \in A_\ell$ and $\alpha \in \Delta^{m-1}$, then letting $v_\ell = \lambda \alpha_\ell z_\ell$ we have

$$\sum_{\ell=1}^m \|v_\ell\|_{\gamma^\ell} = \sum_{\ell=1}^m \|\lambda \alpha_\ell z_\ell\|_{\gamma^\ell} = \lambda \sum_{\ell=1}^m \alpha_\ell \|z_\ell\|_{\gamma^\ell} \leq \lambda.$$

Next, we show that both norms have the same dual norm. We noted in Proposition 2 that the dual norm of $\|\cdot\|_\Theta$ takes the form (2). When Θ is the interior of $\text{co}\{\gamma^1, \dots, \gamma^m\}$, this can be written as

$$\|u\|_{*\Theta} = \sup_{\theta \in \Theta} \sqrt{\sum_{i=1}^d \theta_i u_i^2} = \max_{\ell=1}^m \sqrt{\sum_{i=1}^d \gamma_i^\ell u_i^2}.$$

We now compute the dual of the norm μ_C ,

$$\max_{w \in C} \langle w, u \rangle = \max \{ \langle w, u \rangle : w \in \bigcup_{\ell=1}^m A_\ell \} = \max_{\ell=1}^m \max_{w \in A_\ell} \langle w, u \rangle = \max_{\ell=1}^m \sqrt{\sum_{i=1}^d \gamma_i^\ell u_i^2}. \quad (36)$$

It follows that the norms share the same dual norm, hence $\mu_C(\cdot)$ coincides with $\|\cdot\|_\Theta$. \blacksquare

The above proof reveals that the unit ball of the dual norm of $\|\cdot\|_\Theta$ is given by an intersection of ellipsoids in \mathbb{R}^d . Indeed equation (36) provides that

$$\begin{aligned} \left\{ u \in \mathbb{R}^d : \|u\|_{*\Theta} \leq 1 \right\} &= \left\{ u \in \mathbb{R}^d : \max_{\ell=1}^m \sum_{i=1}^d \gamma_i^\ell u_i^2 \leq 1 \right\} \\ &= \left\{ u \in \mathbb{R}^d : \sum_{i=1}^d \gamma_i^\ell u_i^2 \leq 1, \forall \ell \in \mathbb{N}_m \right\} \\ &= \bigcap_{\ell \in \mathbb{N}_m} \left\{ u \in \mathbb{R}^d : \sum_{i=1}^d \gamma_i^\ell u_i^2 \leq 1 \right\}. \end{aligned} \quad (37)$$

Notice that for each $\ell \in \mathbb{N}_m$, the set $\left\{ u \in \mathbb{R}^d : \sum_{i=1}^d \gamma_i^\ell u_i^2 \leq 1 \right\}$ defines a (possibly degenerate) ellipsoid in X , where the i -th semi-principal axis has length $1/\sqrt{\gamma_i^\ell}$ (which is infinite if $\gamma_i^\ell = 0$) and the unit ball of the dual Θ -norm is given by the intersection of m such ellipsoids.

The following result, which is discussed in (Argyriou et al., 2012, Section 2) is key for the proof of Proposition 18.

Corollary 24 *The unit ball of the vector k -support norm is equal to the convex hull of the set $\{w \in \mathbb{R}^d : \text{card}(w) \leq k, \|w\|_2 \leq 1\}$.*

Proof The result follows directly by Corollary 4 for $\mathcal{G} = \mathcal{G}_k$ observing that in this case $\bigcup_{g \in \mathcal{G}_k} \{w \in \mathbb{R}^d : \text{supp}(w) \subseteq g, \|w\|_2 \leq 1\} = \{w \in \mathbb{R}^d : \text{card}(w) \leq k, \|w\|_2 \leq 1\}$. ■

The next result is due to Von Neumann (1937); see also Lewis (1995).

Theorem 25 (Von Neumann's trace inequality) *For any $d \times m$ matrices X and Y ,*

$$\text{tr}(XY^T) \leq \langle \sigma(X), \sigma(Y) \rangle$$

and equality holds if and only if X and Y admit a simultaneous singular value decomposition, that is, $X = U \text{diag}(\sigma(X)) V^T$, $Y = U \text{diag}(\sigma(Y)) V^T$, where $U \in \mathbb{R}^{d \times d}$ and $V \in \mathbb{R}^{m \times m}$ are orthogonal matrices.

The following inequality is given in Marshall and Olkin (1979, Sec. 9 H.1.b).

Lemma 26 *If $A, B \in \mathbf{S}_+^d$, then it holds*

$$\text{tr}(AB) = \sum_{i=1}^d \lambda_i(AB) \geq \sum_{i=1}^d \lambda_i(A) \lambda_{d-i+1}(B).$$

References

- J. Abernethy, F. Bach, T. Evgeniou, and J.-P. Vert. A new approach to collaborative filtering. *Journal of Machine Learning Research*, 10:803–826, 2009.
- J. Affalo, A. Ben-Tal, C. Bhattacharyya, J. S. Nath, and S. Raman. Variable sparsity kernel learning. *JMLR*, 12:565–592, 2011.
- A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. *Advances in Neural Information Processing Systems 19*, pages 41–48, 2007.
- A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.
- A. Argyriou, C. A. Micchelli, M. Pontil, L. Shen, and Y. Xu. Efficient first order methods for linear composite regularizers. *CoRR*, abs/1104.1436, 2011.
- A. Argyriou, R. Foygel, and N. Srebro. Sparse prediction with the k -support norm. *Advances in Neural Information Processing Systems 25*, pages 1466–1474, 2012.
- F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Mach. Learn.*, 4(1):1–106, 2011.
- P. L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.
- H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Canadian Mathematical Society, 2010.
- A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, 2(1):183–202, 2009.
- D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, 2003.
- R. Bhatia. *Matrix Analysis*. Springer, 1997.
- S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- G. Cavallanti, N. Cesa-Bianchi, and C. Gentile. Linear algorithms for online multitask classification. *Journal of Machine Learning Research*, 1:2901–2934, 2010.
- S. Chatterjee, S. Chen, and A. Banerjee. Generalized Dantzig selector: application to the k -support norm. In *Advances in Neural Information Processing Systems 28*, 2014.
- P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems*. Springer, 2011.
- J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.
- T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.
- T. Evgeniou, M. Pontil, and O. Toubia. A convex optimization approach to modeling heterogeneity in conjoint estimation. *Marketing Science*, 26:805–818, 2007.
- K. Gkirtzou, J. Honorio, D. Samaras, R. Goldstein, and M. B. Blaschko. fMRI analysis of cocaine addiction using k -support sparsity. In *International Symposium on Biomedical Imaging*, 2013.
- Y. Grandvalet. Least absolute shrinkage is equivalent to quadratic penalization. In *ICANN 98*, pages 201–206. Springer London, 1998.
- L. Jacob, F. Bach, and J.-P. Vert. Clustered multi-task learning: a convex formulation. *Advances in Neural Information Processing Systems 21*, 2009a.
- L. Jacob, G. Obozinski, and J.-P. Vert. Group lasso with overlap and graph lasso. *Proceedings of the 26th International Conference on Machine Learning*, 2009b.
- M. Jaggi and M. Sultovsky. A simple algorithm for nuclear norm regularized problems. *Proceedings of the 27th International Conference on Machine Learning*, 2010.
- Z. Kang, K. Gramann, and F. Sha. Learning with whom to share in multi-task feature learning. In *Proceedings of the 28th International Conference on Machine Learning*, 2011.

- V. Koltchinskii and D. Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *The Annals of Statistics*, 30(1):1–50, 2002.
- C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2009.
- P. J. Lenk, W. S. DeSarbo, P. E. Green, and M. R. Young. Hierarchical bayes conjoint analysis: Recovery of partworth heterogeneity from reduced experimental designs. *Marketing Science*, 15(2):173–191, 1996.
- A. S. Lewis. The convex analysis of unitarily invariant matrix functions. *Journal of Convex Analysis*, 2:173–183, 1995.
- H. Li, N. Chen, and L. Li. Error analysis for matrix elastic-net regularization algorithms. *IEEE Transactions on Neural Networks and Learning Systems*, 23:5:737–748, 2012.
- A. W. Marshall and I. Olkin. *Inequalities: Theory of Majorization and its Applications*. Academic Press, 1979.
- A. Maurer. Bounds for linear multi-task learning. *JMLR*, 2006.
- A. Maurer and M. Pontil. A uniform lower error bound for half-space learning. *ALT*, 2008.
- A. Maurer and M. Pontil. Structured sparsity and generalization. *The Journal of Machine Learning Research*, 13:671–690, 2012.
- A. Maurer and M. Pontil. Excess risk bounds for multitask learning with trace norm regularization. In *Proceedings of The 27th Conference on Learning Theory (COLT)*, 2013.
- R. Mazumder, T. Hastie, and R. Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11:2287–2322, 2010.
- A. M. McDonald, M. Pontil, and D. Stamos. Spectral k -support regularization. In *Advances in Neural Information Processing Systems 28*, 2014.
- C. A. Micchelli and M. Pontil. Learning the kernel function via regularization. *Journal of Machine Learning Research*, 6:1099–1125, 2005.
- C. A. Micchelli and M. Pontil. Feature space perspectives for learning the kernel. *Machine Learning*, 66:297–319, 2007.
- C. A. Micchelli, J. M. Morales, and M. Pontil. A family of penalty functions for structured sparsity. *Advances in Neural Information Processing Systems 23*, 2010.
- C. A. Micchelli, J. M. Morales, and M. Pontil. Regularizers for structured sparsity. *Advances in Comp. Mathematics*, 38:455–489, 2013.
- Y. Nesterov. Gradient methods for minimizing composite objective function. *Center for Operations Research and Econometrics*, 76, 2007.

- G. Obozinski and F. Bach. Convex relaxation for combinatorial penalties. *CoRR*, 2012.
- R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.
- R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis*. Springer, 2009.
- W. Rudin. *Functional Analysis*. McGraw Hill, 1991.
- N. Srebro, J. D. M. Rennie, and T. S. Jaakkola. Maximum-margin matrix factorization. *Advances in Neural Information Processing Systems 17*, 2005.
- M. Szafranski, Y. Grandvalet, and P. Morizet-Mahoudeaux. Hierarchical penalization. In *Advances in Neural Information Processing Systems 21*, 2007.
- R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58:267–288, 1996.
- K.-C. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. *SIAM J. on Img. Sci.*, 4:573–596, 2011.
- J. Von Neumann. *Some matrix-inequalities and metrization of matrix-space*. Tomsk. Univ. Rev. Vol I, 1937.
- H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67(2):301–320, 2005.

Minimum Density Hyperplanes

Nicos G. Pavlidis

*Department of Management Science
Lancaster University
Lancaster, LA1 4YX, UK*

N.PAVLIDIS@LANCASTER.AC.UK

David P. Hofmeyr

*Department of Mathematics and Statistics
Lancaster University
Lancaster, LA1 4YF, UK*

D.HOFMEYR@LANCASTER.AC.UK

Sotiris K. Tasoulis

*Department of Applied Mathematics
Liverpool John Moores University,
Liverpool, L3 3AF, UK*

S.TASOULIS@LJMU.AC.UK

Editor: Andreas Krause

Abstract

Associating distinct groups of objects (clusters) with contiguous regions of high probability density (high-density clusters), is central to many statistical and machine learning approaches to the classification of unlabelled data. We propose a novel hyperplane classifier for clustering and semi-supervised classification which is motivated by this objective. The proposed *minimum density hyperplane* minimises the integral of the empirical probability density function along it, thereby avoiding intersection with high density clusters. We show that the minimum density and the maximum margin hyperplanes are asymptotically equivalent, thus linking this approach to maximum margin clustering and semi-supervised support vector classifiers. We propose a projection pursuit formulation of the associated optimisation problem which allows us to find minimum density hyperplanes efficiently in practice, and evaluate its performance on a range of benchmark data sets. The proposed approach is found to be very competitive with state of the art methods for clustering and semi-supervised classification.

Keywords: low-density separation, high-density clusters, clustering, semi-supervised classification, projection pursuit

1. Introduction

We study the fundamental learning problem: *Given a random sample from an unknown probability distribution with no, or partial label information, identify a separating hyperplane that avoids splitting any of the distinct groups (clusters) present in the sample.* We adopt the cluster definition given by Hartigan (1975, chap. 11), in which a *high-density cluster* is defined as a maximally connected component of the level set of the probability density function, $p(\mathbf{x})$, at level $c \geq 0$,

$$\text{lev}_c p(\mathbf{x}) = \left\{ \mathbf{x} \in \mathbb{R}^d \mid p(\mathbf{x}) > c \right\}.$$

An important advantage of this approach over other methods is that it is well founded from a statistical perspective, in the sense that a well-defined population quantity is being estimated.

However, since $p(\mathbf{x})$ is typically unknown, detecting high-density clusters necessarily involves estimates of this function, and standard approaches to nonparametric density estimation are reliable only in low dimensions. A number of existing *density clustering* algorithms approximate the level sets of the empirical density through a union of spheres around points whose estimated density exceeds a user-defined threshold (Walker, 1997; Cuevas et al., 2000, 2001; Rinaldo and Wasserman, 2010). The choice of this threshold affects both the shape and number of detected clusters, while an appropriate threshold is typically not known in advance. The performance of these methods deteriorates sharply as dimensionality increases, unless the clusters are assumed to be clearly discernible (Rinaldo and Wasserman, 2010). An alternative is to consider the more specific problem of allocating observations to clusters, which shifts the focus to local properties of the density, rather than its global approximation. The central idea underlying such methods is that if a pair of observations belong to the same cluster they must be connected through a path traversing only high-density regions. Graph theory is a natural choice to address this type of problem. Azzalini and Torelli (2007); Stuetzle and Nugent (2010) and Menardi and Azzalini (2014) have recently proposed algorithms based on this approach. Even these approaches however are limited to problems of low dimensionality by the standards of current applications (Menardi and Azzalini, 2014).

An equivalent formulation of the density clustering problem is to assume that clusters are separated through contiguous regions of low probability density, known as the *low-density separation* assumption. In both clustering and semi-supervised classification, identifying the hyperplane with the maximum margin is considered a direct implementation of the low-density separation approach. Motivated by the success of support vector machines (SVMs) in classification, maximum margin clustering (MMC) (Xu et al., 2004), seeks the maximum margin hyperplane to perform a binary partition (bi-partition) of unlabelled data. MMC can be equivalently viewed as seeking the binary labelling of the data sample that will maximise the margin of an SVM estimated using the assigned labels.

In a plethora of applications data can be collected cheaply and automatically, while labelling observations is a manual task that can be performed for a small proportion of the data only. Semi-supervised classifiers attempt to exploit the abundant unlabelled data to improve the generalisation error over using only the scarce labelled examples. Unlabelled data provide additional information about the marginal density, $p(\mathbf{x})$, but this is beneficial only insofar as it improves the inference of the class conditional density, $p(\mathbf{x}|y)$. Semi-supervised classification relies on the assumption that a relationship between $p(\mathbf{x})$ and $p(\mathbf{x}|y)$ exists. The most frequently assumed relationship is that high-density clusters are associated with a single class (cluster assumption), or equivalently that class boundaries pass through low-density regions (low-density separation assumption). The most widely used semi-supervised classifier based on the low-density separation assumption is the semi-supervised support vector machine (S²SVM) (Vapnik and Sterin, 1977; Joachims, 1999; Chapelle and Zien, 2005). S²SVMs implement the low-density separation assumption by partitioning the data according to the maximum margin hyperplane with respect to both labelled and unlabelled data.

Encouraging theoretical results for semi-supervised classification have been obtained under the cluster assumption. If $p(\mathbf{x})$ is a mixture of class conditional distributions, Castelli and Cover (1995, 1996) have shown that the generalisation error will be reduced exponentially in the number of labelled examples if the mixture is identifiable. More recently, Singh et al. (2009) showed that the mixture components can be identified if $p(\mathbf{x})$ is a mixture of a finite number of smooth density functions, and the separation between mixture components is large. Rigollet (2007) considers the cluster assumption in a nonparametric setting, that is in terms of density level sets, and shows that the generalisation error of a semi-supervised classifier decreases exponentially given a sufficiently large number of unlabelled data. However, the cluster assumption is difficult to verify with a limited number of labelled examples. Furthermore, the algorithms proposed by Rigollet (2007) and Singh et al. (2009) are difficult to implement efficiently even if the cluster assumption holds. This renders them impractical for real-world problems (Ji et al., 2012).

Although intuitive, the claim that maximising the margin over (labelled and) unlabelled data is equivalent to identifying the hyperplane that goes through regions with the lowest possible probability density has received surprisingly little attention. The work of Ben-David et al. (2009) is the only attempt we are aware of to theoretically investigate this claim. Ben-David et al. (2009) quantify the notion of a low-density separator by defining the *density on a hyperplane*, as the integral of the probability density function along the hyperplane. They study the existence of universally consistent algorithms to compute the hyperplane with minimum density. The maximum hard margin classifier is shown to be consistent only in one dimensional problems. In higher dimensions only a soft-margin algorithm is a consistent estimator of the minimum density hyperplane. Ben-David et al. (2009) do not provide an algorithm to compute low density hyperplanes.

This paper introduces a novel approach to clustering and semi-supervised classification which directly identifies low-density hyperplanes in the finite sample setting. In this approach the density on a hyperplane criterion proposed by Ben-David et al. (2009) is directly minimised with respect to a kernel density estimator that employs isotropic Gaussian kernels. The density on a hyperplane provides a uniform upper bound on the value of the empirical density at points that belong to the hyperplane. This bound is tight and proportional to the density on the hyperplane. Therefore, the smallest upper bound on the value of the empirical density on a hyperplane is achieved by hyperplanes that minimise the density on a hyperplane criterion. An important feature of the proposed approach is that the density on a hyperplane can be evaluated exactly through a one-dimensional kernel density estimator, constructed from the projections of the data sample onto the vector normal to the hyperplane. This renders the computation of minimum density hyperplanes tractable even in high dimensional applications.

We establish a connection between the minimum density hyperplane and the maximum margin hyperplane in the finite sample setting. In particular, as the bandwidth of the kernel density estimator is reduced towards zero, the minimum density hyperplane converges to the maximum margin hyperplane. An intermediate result establishes that there exists a positive bandwidth such that the partition of the data sample induced by the minimum density hyperplane is identical to that of the maximum margin hyperplane.

The remaining paper is organised as follows: The formulation of the minimum density hyperplane problem as well as basic properties are presented in Section 2. Section 3

establishes the connection between minimum density hyperplanes and maximum margin hyperplanes. Section 4 discusses the estimation of minimum density hyperplanes and the computational complexity of the resulting algorithm. Experimental results are presented in Section 5, followed by concluding remarks and future research directions in Section 6.

2. Problem Formulation

We study the problem of estimating a hyperplane to partition a finite data set, $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^d$, without splitting any of the high-density clusters present. We assume that \mathcal{X} is an i.i.d. sample of a random variable \mathbf{X} on \mathbb{R}^d , with unknown probability density function $p : \mathbb{R}^d \rightarrow \mathbb{R}^+$. A hyperplane is defined as $H(\mathbf{v}, b) := \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{v} \cdot \mathbf{x} = b\}$, where without loss of generality we restrict attention to hyperplanes with unit normal vector, i.e., those parameterised by $(\mathbf{v}, b) \in S^{d-1} \times \mathbb{R}$, where $S^{d-1} = \{\mathbf{v} \in \mathbb{R}^d \mid \|\mathbf{v}\| = 1\}$. Following Ben-David et al. (2009) we define the *density on the hyperplane* $H(\mathbf{v}, b)$ as the integral of the probability density function along the hyperplane,

$$I(\mathbf{v}, b) := \int_{H(\mathbf{v}, b)} p(\mathbf{x}) d\mathbf{x}. \quad (1)$$

We approximate $p(\mathbf{x})$ through a kernel density estimator with isotropic Gaussian kernels,

$$\hat{p}(\mathbf{x} \mid \mathcal{X}, h^2 I) = \frac{1}{n(2\pi h^2)^{d/2}} \sum_{i=1}^n \exp \left\{ -\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2h^2} \right\}. \quad (2)$$

This class of kernel density estimators has the useful property that the integral in Equation (1) can be evaluated exactly by projecting \mathcal{X} onto \mathbf{v} ; constructing a one-dimensional density estimator with Gaussian kernels and bandwidth h ; and evaluating the density at b ,

$$\begin{aligned} \hat{I}(\mathbf{v}, b, \mathcal{X}, h^2 I) &:= \int_{H(\mathbf{v}, b)} \hat{p}(\mathbf{x} \mid \mathcal{X}, h^2 I) d\mathbf{x} \\ &= \frac{1}{n\sqrt{2\pi}h^2} \sum_{i=1}^n \exp \left\{ -\frac{(b - \mathbf{v} \cdot \mathbf{x}_i)^2}{2h^2} \right\} = \hat{p}(b \mid \{\mathbf{v} \cdot \mathbf{x}_i\}_{i=1}^n, h^2). \end{aligned} \quad (3)$$

The univariate kernel estimator $\hat{p}(\cdot \mid \{\mathbf{v} \cdot \mathbf{x}_i\}_{i=1}^n, h^2)$ approximates the *projected density* on \mathbf{v} , that is, the density function of the random variable, $X_{\mathbf{v}} = \mathbf{X} \cdot \mathbf{v}$. Henceforth we use $\hat{I}(\mathbf{v}, b)$ to approximate $I(\mathbf{v}, b)$. To simplify terminology we refer to $\hat{I}(\mathbf{v}, b)$ as the *density on $H(\mathbf{v}, b)$* , or the *density integral on $H(\mathbf{v}, b)$* , rather than the empirical density, or the empirical density integral, respectively. For notational convenience we write $I(\mathbf{v}, b)$ for $\hat{I}(\mathbf{v}, b, \mathcal{X}, h^2 I)$, where \mathcal{X} and h are apparent from context.

The following Lemma, adapted from Tasoulis et al., 2010, Lemma 3, shows that $\hat{I}(\mathbf{v}, b)$ provides an upper bound for the maximum value of the empirical density at any point that belongs to the hyperplane.

Lemma 1 *Let $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^d$, and $\hat{p}(\mathbf{x} \mid \mathcal{X}, h^2 I)$ be a kernel density estimator with isotropic Gaussian kernels. Then, for any $(\mathbf{v}, b) \in S^{d-1} \times \mathbb{R}$,*

$$\max_{\mathbf{x} \in H(\mathbf{v}, b)} \hat{p}(\mathbf{x} \mid \mathcal{X}, h^2 I) \leq (2\pi h^2)^{\frac{1-d}{2}} \hat{I}(\mathbf{v}, b), \quad \text{for all } \mathbf{x} \in H(\mathbf{v}, b). \quad (4)$$

This lemma shows that a hyperplane, $H(\mathbf{v}, b)$, cannot intersect level sets of the empirical density with level higher than $(2\pi h^2)^{\frac{1-d}{2}} \hat{I}(\mathbf{v}, b)$. The proof of the lemma relies on the fact that projection contracts distances, and follows from simple algebra. In Equation (4) equality holds if and only if there exists $\mathbf{x} \in H(\mathbf{v}, b)$ and $\mathbf{c} \in \mathbb{R}^n$ such that all $\mathbf{x}_i \in \mathcal{X}$, can be written as $\mathbf{x}_i = \mathbf{x} + c_i \mathbf{v}$. It is therefore not possible to obtain a uniform upper bound on the value of the empirical density at points that belong to $H(\mathbf{v}, b)$ that is lower than $(2\pi h^2)^{\frac{1-d}{2}} \hat{I}(\mathbf{v}, b)$ using only one-dimensional projections. Since the upper bound of Lemma 1 is tight and proportional to $\hat{I}(\mathbf{v}, b)$, minimising the density on the hyperplane leads to the lowest upper bound on the maximum value of the empirical density along the hyperplane separator.

To obtain hyperplane separators that are meaningful for clustering and semi-supervised classification, it is necessary to constrain the set of feasible solutions, because the density on a hyperplane can be made arbitrarily low by considering a hyperplane that intersects only the tail of the density. In other words, for any \mathbf{v} , $I(\mathbf{v}, b)$ can be made arbitrarily low for sufficiently large $|b|$. In both problems the constraints restrict the feasible set to a subset of the hyperplanes that intersect the interior of the convex hull of \mathcal{X} . In detail, let $\text{conv } \mathcal{X}$ denote the convex hull of \mathcal{X} , and assume $\text{Int}(\text{conv } \mathcal{X}) \neq \emptyset$. Define C to be the set of hyperplanes that intersect $\text{Int}(\text{conv } \mathcal{X})$,

$$C = \left\{ H(\mathbf{v}, b) \mid (\mathbf{v}, b) \in \mathcal{S}^{d-1} \times \mathbb{R}, \exists \mathbf{z} \in \text{Int}(\text{conv } \mathcal{X}) \text{ s.t. } \mathbf{v} \cdot \mathbf{z} = b \right\}. \quad (5)$$

Then denote by F the set of feasible hyperplanes, where $F \subset C$. We define the *minimum density hyperplane* (MDH), $H(\mathbf{v}^*, b^*) \in F$ to satisfy,

$$\hat{I}(\mathbf{v}^*, b^*) = \min_{(\mathbf{v}, b) \in F} \hat{I}(\mathbf{v}, b). \quad (6)$$

In the following subsections we discuss the specific formulations for clustering and semi-supervised classification in turn.

2.1 Clustering

Since high-density clusters are formed around the modes of $p(\mathbf{x})$, the convex hull of these modes would be a natural choice to define the set of feasible hyperplanes. Unfortunately, this convex hull is unknown and difficult to estimate. We instead propose to constrain the distance of hyperplanes to the origin, b . Such a constraint is inevitable as for any $\mathbf{v} \in \mathcal{S}^{d-1}$, $\hat{I}(\mathbf{v}, b)$ can become arbitrarily close to zero for sufficiently large $|b|$. Obviously, such hyperplanes are inappropriate for the purposes of bi-partitioning as they assign all the data to the same partition. Rather than fixing b to a constant, we constrain it in the interval,

$$F(\mathbf{v}) = [\mu_{\mathbf{v}} - \alpha \sigma_{\mathbf{v}}, \mu_{\mathbf{v}} + \alpha \sigma_{\mathbf{v}}], \quad (7)$$

where $\mu_{\mathbf{v}}$ and $\sigma_{\mathbf{v}}$ denote the mean and standard deviation, respectively, of the projections $\{\mathbf{v} \cdot \mathbf{x}_i\}_{i=1}^n$. The parameter $\alpha \geq 0$, controls the width of the interval, and has a probabilistic interpretation from Chebyshev's inequality. Smaller values of α favour more balanced partitions of the data at the risk of excluding low density hyperplanes that separate clusters more effectively. On the other hand, increasing α increases the risk of separating out only a

few outlying observations. We discuss in detail how to set this parameter in the experimental results section. If $\text{Int}(\text{conv } \mathcal{X}) \neq \emptyset$, then there exists $\alpha > 0$ such that the set of feasible hyperplanes for clustering, F_{CL} , satisfies,

$$F_{\text{CL}} = \left\{ H(\mathbf{v}, b) \mid (\mathbf{v}, b) \in \mathcal{S}^{d-1} \times \mathbb{R}, b \in F(\mathbf{v}) \right\} \subset C, \quad (8)$$

where C is the set of hyperplanes that intersect $\text{Int}(\text{conv } \mathcal{X})$, as defined in Equation (5).

The minimum density hyperplane for clustering is the solution to the following constrained optimisation problem,

$$\min_{(\mathbf{v}, b) \in \mathcal{S}^{d-1} \times \mathbb{R}} \hat{I}(\mathbf{v}, b), \quad (9a)$$

$$\text{subject to: } b - \mu_{\mathbf{v}} + \alpha \sigma_{\mathbf{v}} \geq 0, \quad (9b)$$

$$\mu_{\mathbf{v}} + \alpha \sigma_{\mathbf{v}} - b \geq 0. \quad (9c)$$

Since the objective function and the constraints are continuously differentiable, MDHs can be estimated through constrained optimisation methods like sequential quadratic programming (SQP). Unfortunately the problem of local minima due to the nonconvexity of the objective function seriously hinders the effectiveness of this approach.

To mitigate this we propose a parameterised optimisation formulation, which gives rise to a projection pursuit approach. Projection pursuit methods optimise a measure of ‘‘interestingness’’ of a linear projection of a data sample, known as the projection index. For our problem the natural choice of projection index for \mathbf{v} is the minimum value of the projected density within the feasible region, $\min_{b \in F(\mathbf{v})} \hat{I}(\mathbf{v}, b)$. This index gives the minimum density integral of feasible hyperplanes with normal vector \mathbf{v} . To ensure the differentiability of the projection index we incorporate a penalty term into the objective function. We define the penalised density integral as,

$$f_{\text{CL}}(\mathbf{v}, b) = \hat{I}(\mathbf{v}, b) + \frac{L}{\eta} \max\{0, \mu_{\mathbf{v}} - \alpha \sigma_{\mathbf{v}} - b, b - \mu_{\mathbf{v}} - \alpha \sigma_{\mathbf{v}}\}^{1+\epsilon}, \quad (10)$$

where, $L = (e^{1/2} h^2 \sqrt{2\pi})^{-1} \geq \sup_{b \in \mathbb{R}} \left| \frac{\partial \hat{I}(\mathbf{v}, b)}{\partial b} \right|$, $\epsilon \in (0, 1)$ is a constant term that ensures that the penalty function is everywhere continuously differentiable, and $\eta \in (0, 1)$. Other penalty functions are possible, but we only consider the above due to its simplicity, and the fact that its parameters offer a direct interpretation: L in terms of the derivative of the projected density on \mathbf{v} ; and η in terms of the desired accuracy of the minimisers of $f_{\text{CL}}(\mathbf{v}, b)$ relative to the minimisers of Equation (9), as discussed in the following proposition.

Proposition 2 For $\mathbf{v} \in \mathcal{S}^{d-1}$, define, the set of minimisers,

$$B(\mathbf{v}) = \arg \min_{b \in F(\mathbf{v})} \hat{I}(\mathbf{v}, b), \quad (11)$$

$$B_C(\mathbf{v}) = \arg \min_{b \in \mathbb{R}} f_{\text{CL}}(\mathbf{v}, b) \quad (12)$$

For every $b^* \in B(\mathbf{v})$ there exists $b_C^* \in B_C(\mathbf{v})$ such that $|b^* - b_C^*| \leq \eta$. Moreover, there are no minimisers of $f_{\text{CL}}(\mathbf{v}, b)$ outside the interval $[\mu_{\mathbf{v}} - \alpha \sigma_{\mathbf{v}} - \eta, \mu_{\mathbf{v}} + \alpha \sigma_{\mathbf{v}} + \eta]$,

$$B_C(\mathbf{v}) \cap \mathbb{R} \setminus [\mu_{\mathbf{v}} - \alpha \sigma_{\mathbf{v}} - \eta, \mu_{\mathbf{v}} + \alpha \sigma_{\mathbf{v}} + \eta] = \emptyset.$$

Proof

Any minimiser in the interior of the feasible region, $b^* \in B(\mathbf{v}) \cap \text{Int}(F(\mathbf{v}))$, also minimises the penalised function, since $f_{\text{CL}}(\mathbf{v}, b) = \hat{I}(\mathbf{v}, b)$ for all $b \in \text{Int}(F(\mathbf{v}))$, hence $b^* \in B_C(\mathbf{v})$.

Next we consider the case when either or both of the boundary points of $F(\mathbf{v})$, $b^- = \mu_0 - \sigma\sigma_{\mathbf{v}}$ and $b^+ = \mu_0 + \sigma\sigma_{\mathbf{v}}$, are contained in $B(\mathbf{v})$. It suffices to show that, $f_{\text{CL}}(\mathbf{v}, b) > \hat{I}(\mathbf{v}, b^-)$ for all $b < b^- - \eta$, and $f_{\text{CL}}(\mathbf{v}, b) > \hat{I}(\mathbf{v}, b^+)$ for all $b > b^+ + \eta$. We discuss only the case $b > b^+ + \eta$ as the treatment of $b < b^- - \eta$ is identical. Assume that $\hat{I}(\mathbf{v}, b) < \hat{I}(\mathbf{v}, b^+)$ (since in the opposite case the result follows immediately: $f_{\text{CL}}(\mathbf{v}, b) > \hat{I}(\mathbf{v}, b) > \hat{I}(\mathbf{v}, b^+)$). From the mean value theorem there exists $\xi \in (b^+, b)$ such that,

$$\begin{aligned} \hat{I}(\mathbf{v}, b^+) &= \hat{I}(\mathbf{v}, b) - (b - b^+) \frac{\partial \hat{I}(\mathbf{v}, b)}{\partial b} \Big|_{b=\xi} \\ &\leq \hat{I}(\mathbf{v}, b) + (b - b^+)L \\ &< \hat{I}(\mathbf{v}, b) + \frac{L(b - b^+)1+\epsilon}{\eta^\epsilon} = f_{\text{CL}}(\mathbf{v}, b), \end{aligned}$$

In the above we used the following facts: $\frac{\partial \hat{I}(\mathbf{v}, b)}{\partial b} \Big|_{b=\xi} < 0$, $L \geq \sup_{b \in \mathbb{R}} \left| \frac{\partial \hat{I}(\mathbf{v}, b)}{\partial b} \right|$, and $\frac{b - b^+}{\eta} > 1$. ■

We define the projection index for the clustering problem as the minimum of the penalised density integral,

$$\phi_{\text{CL}}(\mathbf{v}) = \min_{b \in \mathbb{R}} f_{\text{CL}}(\mathbf{v}, b). \quad (13)$$

Since the optimisation problem of Equation (13) is one-dimensional it is simple to compute the set of global minimisers $B_C(\mathbf{v})$. As we discuss in Section 4, this is necessary to compute directional derivatives of the projection index, as well as, to determine whether ϕ_{CL} is differentiable. We call the optimisation of ϕ_{CL} , *minimum density projection pursuit* (MDP²). For each \mathbf{v} , MDP² considers only the optimal choice of b . This enables it to avoid local minima of $\hat{I}(\mathbf{v}, \cdot)$. Most importantly MDP² is able to accommodate a discontinuous change in the location of the global minimiser(s), $\arg \min_{b \in \mathbb{R}} f_{\text{CL}}(\mathbf{v}, b)$, as \mathbf{v} changes. Neither of the above can be achieved when the optimisation is jointly over (\mathbf{v}, b) as in the original constrained optimisation problem, Equation (9). The projection index ϕ_{CL} is continuous, but it is not guaranteed to be everywhere differentiable when $B_C(\mathbf{v})$ is not a singleton. The resulting optimisation problem is therefore nonsmooth and nonconvex.

To illustrate the effectiveness of MDP² to estimate MDHs, we compare this approach with a direct optimisation of the constrained problem given in Equation (9) using SQP. To enable visualisation we consider the two-dimensional S1 data set (Franti and Virmajoki, 2006), constructed by sampling from a Gaussian mixture distribution with fifteen components, where each component corresponds to a cluster. Figure 1 depicts the MDHs obtained over 100 random initialisations of SQP and MDP². It is evident that SQP frequently yields hyperplanes that intersect regions with high probability density thus splitting clusters. As SQP always converged in these experiments the poor performance is solely due to convergence to local minima. In contrast, MDP² converges to three different solutions over the 100 experiments, all of which induce high quality partitions, and none intersects a high-density

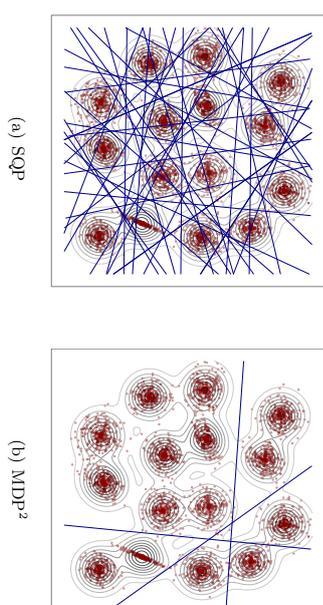


Figure 1: Binary partitions induced by 100 MDHs estimated through SQP and MDP²

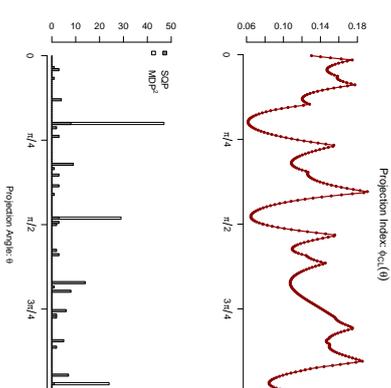


Figure 2: Projection index for S1 data set and solutions obtained through SQP and MDP²

cluster. In polar coordinates any $\mathbf{v} \in S^1$ can be parameterised through a single projection angle. Using this parameterisation, the upper plot of Figure 2 depicts the value of the projection index, $\phi_{\text{CL}}(\mathbf{v}(\theta))$, for $\theta \in [0, \pi]$. The lower plot of the figure provides histograms of the distribution of the solutions (locally optimal projection angles) obtained over the 100 experiments with SQP (grey) and MDP² (white). The figure shows that $\phi_{\text{CL}}(\mathbf{v})$ is continuous but not everywhere differentiable. The solution most frequently obtained through MDP² corresponds to the global optimum, while the only other two solutions identified are the local minimisers with the next two lowest function values. In contrast SQP converges to a much wider range of solutions. Note that this method is not guaranteed to identify the

optimal value of b for any $\mathbf{v}(\theta)$ and this indeed occurs in this example. Therefore the value of $\phi_{\text{CL}}(\mathbf{v})$ is a lower bound for the function values of the minimisers identified through SQP.

2.2 Semi-Supervised Classification

In semi-supervised classification labels are available for a subset of the data sample. The resulting classifier needs to predict as accurately as possible the labelled examples, while avoiding intersection with high-density regions of the empirical density. The MDH formulation can readily accommodate partially labelled data by incorporating the linear constraints associated with the labelled data into the clustering formulation. Without loss of generality assume that the first ℓ examples are labelled by $\mathbf{y} = (y_1, \dots, y_\ell)^\top \in \{-1, 1\}^\ell$. The MDH for semi-supervised classification is the solution to the problem,

$$\min_{(\mathbf{v}, b) \in \mathcal{S}^{d-1} \times \mathbb{R}} \hat{I}(\mathbf{v}, b), \quad (14a)$$

$$\text{subject to: } y_i(\mathbf{v} \cdot \mathbf{x}_i - b) \geq 0, \quad \forall i = 1, \dots, \ell, \quad (14b)$$

$$b - \mu_{\mathbf{v}} + \alpha\sigma_{\mathbf{v}} \geq 0, \quad (14c)$$

$$\mu_{\mathbf{v}} + \alpha\sigma_{\mathbf{v}} - b \geq 0, \quad (14d)$$

where $\hat{I}(\mathbf{v}, b)$, $\mu_{\mathbf{v}}$, and $\sigma_{\mathbf{v}}$ are computed over the entire data set. If the labelled examples are linearly separable the constraints in Equation (14) define a nonempty feasible set of hyperplanes,

$$F_{\text{LB}} = \left\{ H(\mathbf{v}, b) \mid (\mathbf{v}, b) \in \mathcal{S}^{d-1} \times \mathbb{R}, b \in F(\mathbf{v}), y_i(\mathbf{v} \cdot \mathbf{x}_i - b) \geq 0, \forall i \in \{1, \dots, \ell\} \right\} \subset C. \quad (15)$$

Equations (14c) and (14d) act as a *balancing constraint* which discourages MDHs that classify the vast majority of unlabelled data to a single class. Balancing constraints are included in the estimation of S3VMs for the same reason (Joachims, 1999; Chapelle and Zien, 2005).

As in the case of clustering, the direct minimisation of Equation (14) frequently leads to locally optimal solutions. To mitigate this we again propose a projection pursuit formulation. We define the penalised density integral for semi-supervised classification as,

$$f_{\text{SSC}}(\mathbf{v}, b) = f_{\text{CL}}(\mathbf{v}, b) + \gamma \sum_{i=1}^{\ell} \max\{0, -y_i(\mathbf{v} \cdot \mathbf{x}_i - b)\}^{1+\epsilon} \quad (16)$$

where, $\gamma > 0$ is a user-defined constant, which controls the trade-off between reducing the density on the hyperplane, and misclassifying the labelled examples. The projection index is then defined as the minimum of the penalised density integral,

$$\phi_{\text{SSC}}(\mathbf{v}) = \min_{b \in \mathbb{R}} f_{\text{SSC}}(\mathbf{v}, b). \quad (17)$$

3. Connection to Maximum Margin Hyperplanes

In this section we discuss the connection between MDHs and maximum (hard) margin hyperplane separators. The margin of a hyperplane $H(\mathbf{v}, b)$ with respect to a data set \mathcal{X} is

defined as the minimum Euclidean distance between the hyperplane and its nearest datum, margin $H(\mathbf{v}, b) = \min_{\mathbf{x} \in \mathcal{X}} |\mathbf{v} \cdot \mathbf{x} - b|$. (18)

The points whose distance to the hyperplane $H(\mathbf{v}, b)$ is equal to the margin of the hyperplane, that is, $\arg \min_{\mathbf{x} \in \mathcal{X}} |\mathbf{v} \cdot \mathbf{x} - b|$, are called the *support points* of $H(\mathbf{v}, b)$. Let F denote the set of feasible hyperplanes; then the *maximum margin hyperplane* (MMH), $H(\mathbf{v}^m, b^m) \in F$ satisfies,

$$\text{margin } H(\mathbf{v}^m, b^m) = \max_{(\mathbf{v}, b) \mid H(\mathbf{v}, b) \in F} \text{margin } H(\mathbf{v}, b). \quad (19)$$

The main result of this section is Theorem 5, which states that as the bandwidth parameter, h , is reduced to zero the MDH converges to the MMH. An intermediate result, Lemma 4, shows that there exists a positive bandwidth, $h' > 0$ such that, for all $h \in (0, h')$, the partition of the data set induced by the MDH is identical to that of the MMH.

We first discuss some assumptions which allow us to present the theoretical results of this section. As before we assume a fixed and finite data set $\mathcal{X} \subset \mathbb{R}^d$, and approximate its (assumed) underlying probability density function via a kernel density estimator using Gaussian kernels with isotropic bandwidth matrix $h^2 I$. We assume that the interior of the convex hull of the data, $\text{Int}(\text{conv } \mathcal{X})$, is non-empty, and define C as the set of hyperplanes that intersect $\text{Int}(\text{conv } \mathcal{X})$, as in Equation (5). The set of feasible hyperplanes, F , for either clustering or the semi-supervised classification satisfies $F \subset C$. By construction every $H(\mathbf{v}, b) \in F$ defines a hyperplane which partitions \mathcal{X} into two non-empty subsets. Observe that if for each $\mathbf{v} \in \mathcal{S}^{d-1}$ the set $\{b \in \mathbb{R} \mid H(\mathbf{v}, b) \in F\}$ is compact, then by the compactness of \mathcal{S}^{d-1} a maximum margin hyperplane in F exists. For both the clustering and semi-supervised classification problems this compactness holds by construction.

For any $h > 0$, let $(\mathbf{v}_h^*, b_h^*) \in \mathcal{S}^{d-1} \times \mathbb{R}$ parameterise a hyperplane which achieves the minimal density integral over all hyperplanes in F , for bandwidth matrix $h^2 I$. That is,

$$\hat{I}(\mathbf{v}_h^*, b_h^*) = \min_{(\mathbf{v}, b) \mid H(\mathbf{v}, b) \in F} \hat{I}(\mathbf{v}, b). \quad (20)$$

Following the approach of Tong and Koller (2000) we first show that as the bandwidth, h , is reduced towards zero, the density on a hyperplane is dominated by its nearest point. This is achieved by establishing that for all sufficiently small values of h , a hyperplane with non-zero margin has lower density integral than any other hyperplane with smaller margin.

Lemma 3 *Take $H(\mathbf{v}, b) \in F$ with non-zero margin and $0 < \delta < \text{margin } H(\mathbf{v}, b) := M_{\mathbf{v}, b}$. Then $\exists h' > 0$ such that $h \in (0, h')$ and $M_{\mathbf{w}, c} := \text{margin } H(\mathbf{w}, c) \leq M_{\mathbf{v}, b} - \delta$ implies $\hat{I}(\mathbf{v}, b) < \hat{I}(\mathbf{w}, c)$.*

Proof

Using Equation (3) it is easy to see that,

$$\hat{I}(\mathbf{v}, b) \leq \frac{1}{h\sqrt{2\pi}} \exp\left\{-\frac{M_{\mathbf{v}, b}^2}{2h^2}\right\},$$

$$\inf\left\{\hat{I}(\mathbf{w}, c) \mid M_{\mathbf{w}, c} \leq M_{\mathbf{v}, b} - \delta\right\} \geq \frac{1}{nh\sqrt{2\pi}} \exp\left\{-\frac{(M_{\mathbf{v}, b} - \delta)^2}{2h^2}\right\}.$$

Therefore,

$$0 \leq \lim_{h \rightarrow 0^+} \frac{\hat{I}(\mathbf{v}, b)}{\inf_{M_{\mathbf{w},c} \leq M_{\mathbf{v},b} - \delta} \hat{I}(\mathbf{w}, c)} \leq \lim_{h \rightarrow 0^+} \frac{n \exp \left\{ \begin{array}{l} M_{\mathbf{v},b}^2 \\ -\frac{2M_{\mathbf{v},b}^2}{h^2} \end{array} \right\}}{\exp \left\{ \begin{array}{l} -(M_{\mathbf{v},b} - \delta)^2 \\ \frac{2(M_{\mathbf{v},b} - \delta)^2}{h^2} \end{array} \right\}} = 0.$$

Therefore, $\exists h' > 0$ such that $h \in (0, h') \Rightarrow \frac{\hat{I}(\mathbf{v}, b)}{\inf_{M_{\mathbf{w},c} \leq M_{\mathbf{v},b} - \delta} \hat{I}(\mathbf{w}, c)} < 1$. \blacksquare

An immediate corollary of Lemma 3 is that as h tends to zero the margin of the MDH tends to the maximum margin. However, this does not necessarily ensure the stronger result that the sequence of MDHs converges to the MMH. To establish this we require two technical results, which describe some algebraic properties of the MMH, and are provided as part of the proof of Theorem 5 which is given in Appendix A.

The next lemma uses the previous result to show that there exists a positive bandwidth, $h' > 0$, such that an MDH estimated using $h \in (0, h')$ induces the same partition of \mathcal{X} as the MMH. The result assumes that the MMH is unique. Notice that if \mathcal{X} is a sample of realisations of a continuous random variable then this uniqueness holds with probability 1.

Lemma 4 *Suppose there is a unique hyperplane in F with maximum margin, which can be parameterised by $(\mathbf{v}^m, b^m) \in \mathcal{S}^{d-1} \times \mathbb{R}$. Then $\exists h' > 0$ s.t. $h \in (0, h') \Rightarrow H(\mathbf{v}_h^*, b_h^*)$ induces the same partition of \mathcal{X} as $H(\mathbf{v}^m, b^m)$.*

Proof

Let $M = \text{margin } H(\mathbf{v}^m, b^m)$, and let P be the collection of hyperplanes that induce the same partition of \mathcal{X} as that induced by $H(\mathbf{v}^m, b^m)$. Since \mathcal{X} is finite and $H(\mathbf{v}^m, b^m)$ is unique, $\exists \delta > 0$ s.t. $H(w, c) \notin P \Rightarrow \text{margin } H(w, c) \leq M - \delta$. By Lemma 3, $\exists h' > 0$ s.t.,

$$h \in (0, h') \Rightarrow H(\mathbf{v}_h^*, b_h^*) \notin \{H(\mathbf{w}, c) \mid \text{margin } H(\mathbf{w}, c) \leq M - \delta\},$$

therefore $H(\mathbf{v}_h^*, b_h^*) \in P$. \blacksquare

The next theorem is the main result of this section, and states that the MDH converges to the MMH as the bandwidth parameter is reduced to zero. Notice that by the non-unique representation of hyperplanes, the maximum margin hyperplane has two parameterisations in C , namely (\mathbf{v}^m, b^m) and $(-\mathbf{v}^m, -b^m)$. Convergence to the maximum margin hyperplane is therefore equivalent to showing that,

$$\min \{ \|(\mathbf{v}_h^*, b_h^*) - (\mathbf{v}^m, b^m) \|, \| (\mathbf{v}_h^*, b_h^*) + (\mathbf{v}^m, b^m) \| \} \rightarrow 0 \text{ as } h \rightarrow 0^+.$$

Theorem 5 *Suppose there is a unique hyperplane in F with maximum margin, which can be parameterised by $(\mathbf{v}^m, b^m) \in \mathcal{S}^{d-1} \times \mathbb{R}$. Then,*

$$\lim_{h \rightarrow 0^+} \min \{ \|(\mathbf{v}_h^*, b_h^*) - (\mathbf{v}^m, b^m) \|, \| (\mathbf{v}_h^*, b_h^*) + (\mathbf{v}^m, b^m) \| \} = 0.$$

The set F used in Theorem 5 is generic so it can capture the constraints associated with both clustering and semi-supervised classification, Equations (9) and (14) respectively. In the case of semi-supervised classification we must also assume that the labelled data are linearly separable. Theorem 5 is not directly applicable to the MDP² formulations as in this case the function being minimised is not the density on a hyperplane. The next two subsections establish this result for the MDP² formulation of the clustering and semi-supervised classification problem.

3.1 MDP² for Clustering

We have shown that for the constrained optimisation formulation the MDH converges to the MMH within the feasible set, $F_{\text{CL}} \subset C$. In addition, for a fixed \mathbf{v} , Proposition 2 bounds the distance between minimisers of the penalised function f_{CL} , $\arg \min_{\mathbf{v} \in \mathbb{R}} f_{\text{CL}}(\mathbf{v}, b)$, and the optimal b of the constrained problem, $\arg \min_{b \in F(\mathbf{v})} \hat{I}(\mathbf{v}, b)$. Combining these we can show that the optimal solution to the penalised MDP² formulation converges to the maximum margin hyperplane in F_{CL} , provided the parameters within the penalty term suitably depend on the bandwidth parameter, h . While the general case can be shown, for ease of exposition we make the simplifying assumption that the maximum margin hyperplane is strictly feasible, i.e., if (\mathbf{v}^m, b^m) parameterises the maximum margin hyperplane then $b^m \in (h\mathbf{v}^m - \alpha\sigma\mathbf{v}^m, h\mathbf{v}^m + \alpha\sigma\mathbf{v}^m)$.

For $h, \eta, L > 0$ define $(\mathbf{v}_{h,\eta,L}^*, b_{h,\eta,L}^*)$ to be any global minimiser of f_{CL} , i.e.,

$$f_{\text{CL}}(\mathbf{v}_{h,\eta,L}^*, b_{h,\eta,L}^*) = \min_{(\mathbf{v},b) \in \mathcal{S}^{d-1} \times \mathbb{R}} f_{\text{CL}}(\mathbf{v}, b).$$

Lemma 6 *Suppose there is a unique hyperplane in F_{CL} with maximum margin, which can be parameterised by $(\mathbf{v}^m, b^m) \in \mathcal{S}^{d-1} \times \mathbb{R}$. Suppose further that $b^m \in (h\mathbf{v}^m - \alpha\sigma\mathbf{v}^m, h\mathbf{v}^m + \alpha\sigma\mathbf{v}^m)$. For $h > 0$, let $L(h) = (e^{1/2}h^2\sqrt{2\pi})^{-1}$, and $0 < \eta(h) \leq h$. Then,*

$$\lim_{h \rightarrow 0^+} \min \{ \|(\mathbf{v}_{h,\eta(h),L(h)}^*, b_{h,\eta(h),L(h)}^*) - (\mathbf{v}^m, b^m) \|, \| (\mathbf{v}_{h,\eta(h),L(h)}^*, b_{h,\eta(h),L(h)}^*) + (\mathbf{v}^m, b^m) \| \} = 0.$$

Proof

Let $M = \text{margin } H(\mathbf{v}^m, b^m)$ and as in the proof of Lemma 4, let $\delta > 0$ be such that any hyperplane inducing a different partition from $H(\mathbf{v}^m, b^m)$ has margin at most $M - \delta$. Consider the set $F_{\text{CL}}^\delta := \{(\mathbf{v}, b) \in \mathcal{S}^{d-1} \times \mathbb{R} \mid b \in \mathbb{B}_{\delta/2}(F(\mathbf{v}))\}$, where we used the notation $\mathbb{B}_{\delta/2}(F(\mathbf{v}))$ to denote the neighbourhood of $F(\mathbf{v})$ given by $\{r \in \mathbb{R} \mid |d(r, F(\mathbf{v}))| < \delta/2\}$. The set F_{CL}^δ increases the feasible set of hyperplanes by allowing b to range in $b \in \mathbb{B}_{\delta/2}(F(\mathbf{v}))$. For any fixed \mathbf{v} , the maximum margin of all hyperplanes with normal vector \mathbf{v} can increase by at most $\delta/2$. Thus, any hyperplane inducing a different partition compared to $H(\mathbf{v}^m, b^m)$ has a margin at most $M - \delta/2$. Since $H(\mathbf{v}_{h,\eta}^m, b_{h,\eta}^m)$ is strictly feasible it therefore remains the unique maximum margin hyperplane in F_{CL}^δ . Observe now that for $0 < h < \delta/2$ we have $H(\mathbf{v}_{h,\eta(h),L(h)}^*, b_{h,\eta(h),L(h)}^*) \in F_{\text{CL}}^\delta$, by Proposition 2. In addition, by Theorem 5, we know that the minimisers of $\hat{I}(\mathbf{v}, b)$ over F_{CL}^δ , say $H(\mathbf{v}_{h,\eta}^\delta, b_h^\delta)$, satisfy

$$\lim_{h \rightarrow 0^+} \min \{ \|(\mathbf{v}_{h,\eta}^\delta, b_h^\delta) - (\mathbf{v}^m, b^m) \|, \| (\mathbf{v}_{h,\eta}^\delta, b_h^\delta) + (\mathbf{v}^m, b^m) \| \} = 0.$$

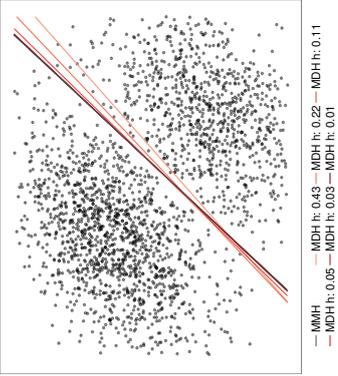


Figure 3: Convergence of the MDH to the maximum margin hyperplane for a decreasing sequence of bandwidth parameters, h .

Now, since $H(\mathbf{v}^m, b^m)$ is strictly feasible $\exists \epsilon' > 0$ s.t. $(\mathbf{v}, b) \in \mathbb{B}_\epsilon(\{(\mathbf{v}^m, b^m), -(\mathbf{v}^m, b^m)\}) \Rightarrow H(\mathbf{v}, b) \in F_{\text{CL}}$. Then for any $0 < \epsilon < \epsilon'$ there exists $h' > 0$ s.t. for $0 < h < h'$ both $(\mathbf{v}_h^*, b_h^*) \in \mathbb{B}_\epsilon(\{(\mathbf{v}^m, b^m), -(\mathbf{v}^m, b^m)\}) \Rightarrow H(\mathbf{v}_h^*, b_h^*) \in F_{\text{CL}}$ and $H(\mathbf{v}_{h, \eta(h), L(h)}^*, b_{h, \eta(h), L(h)}^*) \in F_{\text{CL}}^{\delta}$. Now for $H(\mathbf{v}, b) \in F_{\text{CL}}^{\delta} \setminus F_{\text{CL}}$ we know that $\hat{I}(\mathbf{v}, b) < f_{\text{CL}}(\mathbf{v}, b)$, whereas for $H(\mathbf{v}, b) \in F_{\text{CL}}$, $\hat{I}(\mathbf{v}, b) = f_{\text{CL}}(\mathbf{v}, b)$ and therefore the minimiser of $f_{\text{CL}}(\mathbf{v}, b)$ must lie in the neighbourhood $\mathbb{B}_\epsilon(\{(\mathbf{v}^m, b^m), -(\mathbf{v}^m, b^m)\})$, and the result follows. ■

To illustrate the convergence of the MDH to the MMH we use the two-dimensional data set shown in Figure 3. The data is sampled from a mixture of two Gaussian distributions with equal covariance matrix. The MDH with respect to the true underlying density is $H((1, -1), 0)$. A large margin separator is artificially introduced by removing a few observations in a narrow margin around a hyperplane different from $H((1, -1), 0)$. The margin is intentionally small to ensure that identifying the MMH is non-trivial. Figure 3 illustrates the MDH solutions arising from the MDP2 method for a decreasing sequence of bandwidths, h . Initially the MDH approximately coincides with the optimal MDH with respect to the true density of the Gaussian mixture. As h decreases, the MDH approaches the MMH and for the smallest values of h the two are indistinguishable.

3.2 MDP² for Semi-Supervised Classification

Denote the set of hyperplanes which correctly classify the labelled data by F_{LB} . Under the assumption that $\exists H(\mathbf{v}, b) \in F_{\text{LB}} \cap F_{\text{CL}}$ with non-zero margin, we can show that, provided the parameter γ does not shrink too quickly with h , the hyperplane that minimises f_{SSC} converges to the MMH contained in $F_{\text{LB}} \cap F_{\text{CL}}$, where as before we assume that such an MMH is strictly feasible. To establish this result it is sufficient to show that there exists $h' > 0$ such

that for all $h \in (0, h')$, the optimal hyperplane $H(\mathbf{v}_{h, \eta, L, \gamma}^*, b_{h, \eta, L, \gamma}^*)$ correctly classifies all the labelled examples. If this holds, then $f_{\text{SSC}}(\mathbf{v}_{h, \eta, L, \gamma}^*, b_{h, \eta, L, \gamma}^*) = f_{\text{CL}}(\mathbf{v}_{h, \eta, L, \gamma}^*, b_{h, \eta, L, \gamma}^*)$ for all sufficiently small h , and hence Lemma 6 can be applied to establish the result. The proof relies on the fact that the penalty terms associated with the known labels in Equation (16) are polynomials in b . Provided that γ is bounded below by a polynomial in h , the value of the penalty terms for hyperplanes that do not correctly classify the labelled data dominate the value of the density integral as h approaches zero. Therefore the optimal hyperplane must correctly classify the labelled data for small values of h .

Lemma 7 Define $F_{\text{LB}} = \{H(\mathbf{v}, b) | y_i(\mathbf{v} \cdot \mathbf{x}_i - b) > 0, \forall i = 1, \dots, \ell\}$ and $F_{\text{CL}} = \{H(\mathbf{v}, b) | \mu_{\mathbf{v}} - \alpha \sigma_{\mathbf{v}} \leq b \leq \mu_{\mathbf{v}} + \alpha \sigma_{\mathbf{v}}\}$ and assume that $F_{\text{SSC}} = F_{\text{LB}} \cap F_{\text{CL}} \neq \emptyset$ and that $\exists H(\mathbf{v}, b) \in F_{\text{SSC}}$ with non-zero margin. For $h > 0$, let $L(h) = (e^{1/2} h^2 \sqrt{2\pi})^{-1}$, $0 < \eta(h) \leq h$ and $\gamma(h) \geq h^r$ for some $r > 0$. Then $\exists h' > 0$ s.t. $h \in (0, h') \Rightarrow H(\mathbf{v}_{h, \eta(h), L(h), \gamma(h)}^*, b_{h, \eta(h), L(h), \gamma(h)}^*) \in F_{\text{LB}}$.

Proof

Consider $H(\mathbf{v}, b) \notin F_{\text{LB}}$. Then,

$$f_{\text{SSC}}(\mathbf{v}, b) \geq \frac{1}{n\sqrt{2\pi}h} \exp(-\nu_*^2/2h^2) + \gamma(h)\nu_*^{1+\epsilon} > \gamma(h)\nu_*^{1+\epsilon},$$

where $\nu_* > 0$ minimises $\frac{1}{n\sqrt{2\pi}h} \exp(-\nu^2/2h^2) + \gamma(h)\nu^{1+\epsilon}$. Therefore, ν_* is the unique positive number satisfying,

$$\begin{aligned} \frac{1}{n\sqrt{2\pi}h} \exp\left(-\frac{\nu_*^2}{2h^2}\right) \left(-\frac{\nu_*}{h^2}\right) + (1+\epsilon)\gamma(h)\nu_*^\epsilon &= 0 \\ \Rightarrow \nu_*^{1-\epsilon} &= (1+\epsilon)\gamma(h)n\sqrt{2\pi}h^3 \exp\left(\frac{\nu_*^2}{2h^2}\right) \\ \Rightarrow \nu_* &\geq \left((1+\epsilon)\gamma(h)n\sqrt{2\pi}h^3\right)^{1/1-\epsilon}. \end{aligned}$$

We therefore have,

$$\begin{aligned} f_{\text{SSC}}(\mathbf{v}, b) &> \gamma(h) \left((1+\epsilon)\gamma(h)n\sqrt{2\pi}h^3 \right)^{\frac{1+\epsilon}{1-\epsilon}} \\ &= K\gamma(h)^{\frac{2}{1-\epsilon}} h^{\frac{3(1+\epsilon)}{1-\epsilon}} \\ &\geq K h^{\frac{2+3(1+\epsilon)}{1-\epsilon}}, \end{aligned}$$

where K is a constant which can be chosen independent of (\mathbf{v}, b) . Finally, for any $H(\mathbf{v}', b') \in F_{\text{SSC}}$ with non-zero margin, $\exists h' > 0$ s.t.

$$h \in (0, h') \Rightarrow f_{\text{SSC}}(\mathbf{v}', b') = \hat{I}(\mathbf{v}', b') < K h^{\frac{2+3(1+\epsilon)}{1-\epsilon}} < f_{\text{SSC}}(\mathbf{v}, b).$$

Since K is independent of (\mathbf{v}, b) , the result follows. The final set of inequalities holds since the hyperplane $H(\mathbf{v}', b')$ is assumed to have non-zero margin, say $M_{\mathbf{v}', b'} > 0$, and hence $\hat{I}(\mathbf{v}', b') \leq \frac{1}{n\sqrt{2\pi}} \exp\{-M_{\mathbf{v}', b'}/2h'^2\}$, which tends to zero faster than any polynomial in h . ■

4. Estimation of Minimum Density Hyperplanes

In this section we discuss the computation of MDHs. We first investigate the continuity and differentiability properties required to optimise the projection indices $\phi_{\text{CL}}(\mathbf{v})$ and $\phi_{\text{SSC}}(\mathbf{v})$. Since the domain of both projection indices, $\phi_{\text{CL}}(\mathbf{v})$ and $\phi_{\text{SSC}}(\mathbf{v})$, is the boundary of the unit-sphere in \mathbb{R}^d it is more convenient to express \mathbf{v} in terms of spherical coordinates,

$$v_i(\theta) = \begin{cases} \cos(\theta_i) \prod_{j=1}^{i-1} \sin(\theta_j), & i = 1, \dots, d-1 \\ \prod_{j=1}^{d-1} \sin(\theta_j), & i = d, \end{cases} \quad (21)$$

where $\theta \in \Theta = [0, \pi]^{d-2} \times [0, 2\pi]$ is called the *projection angle*. Using spherical coordinates renders the domain, Θ , convex and compact, and reduces dimensionality by one.

As the following discussion applies to both $\phi_{\text{CL}}(\mathbf{v})$ and $\phi_{\text{SSC}}(\mathbf{v})$ we denote a generic projection index $\phi : \Theta \rightarrow \mathbb{R}$, and the associated set of minimisers, as,

$$\phi(\theta) = \min_{b \in A} f(\mathbf{v}(\theta), b), \quad (22)$$

$$B(\theta) = \{b \in A \mid f(\mathbf{v}(\theta), b) = \phi(\theta)\}, \quad (23)$$

where $f(\mathbf{v}(\theta), b)$ is continuously differentiable, $A \subset \mathbb{R}$ is compact and convex, and the correspondence $B(\theta)$ gives the set of global minimisers of $f(\mathbf{v}(\theta), b)$ for each θ . The definition of A is not critical in our formulation. Setting,

$$A \supset \left[\min_{\mathbf{v} \in S^{d-1}} \{\mu_{\mathbf{v}}\} - \alpha \sigma_{\text{PC}_1} - \eta, \max_{\mathbf{v} \in S^{d-1}} \{\mu_{\mathbf{v}}\} + \alpha \sigma_{\text{PC}_1} + \eta \right], \quad (24)$$

where $\sigma_{\text{PC}_1}^2$ is the variance of the projections along the first principal component, ensures that the set of hyperplanes that satisfy the constraint of Equation (7) will be a subset of A for all \mathbf{v} .

Berge's maximum theorem (Berge, 1963; Polak, 1987), establishes the continuity of $\phi(\theta)$ and the upper-semicontinuity (i.s.c.) of the correspondence $B(\theta)$. Theorem 3.1 in (Polak, 1987) enables us to establish that $\phi(\theta)$ is locally Lipschitz continuous. Using Theorem 4.13 of Bonnans and Shapiro (2000) we can further show that $\phi(\theta)$ is directionally differentiable everywhere. The directional derivative at θ in the direction ν is given by,

$$d\phi(\theta; \nu) = \min_{b \in B(\theta)} D_b f(\mathbf{v}(\theta), b) \cdot \nu, \quad (25)$$

where D_b denotes the derivative with respect to θ . It is clear from Equation (25) that $\phi(\theta)$ is differentiable if $D_b f(\mathbf{v}(\theta), b)$ is the same for all $b \in B(\theta)$. If $B(\theta)$ is a singleton then this condition is trivially satisfied and $\phi(\theta)$ is continuously differentiable at θ .

It is possible to construct examples in which $B(\theta)$ is not a singleton. However, with the exception of contrived examples, our experience with real and simulated data sets indicates that when h is set through standard bandwidth selection rules $B(\theta)$ is almost always a singleton over the optimisation path.

Proposition 8 *Suppose $B(\theta)$ is a singleton for almost all $\theta \in \Theta$. Then $\phi(\theta)$ is continuously differentiable almost everywhere.*

Proof The result follows immediately from the fact that if $B(\theta) = \{b\}$ is a singleton, then the derivative $D\phi(\theta) = D_b f(\mathbf{v}(\theta), b)$, which is continuous. ■

Wolfe (1972) has provided early examples of how standard gradient-based methods can fail to converge to a local optimum when used to minimise nonsmooth functions. In the last decade a new class of nonsmooth optimisation algorithms has been developed based on gradient sampling (Burke et al., 2006). Gradient sampling methods use generalised gradient descent to find local minima. At each iteration points are randomly sampled in a radius ε of the current candidate solution, and the gradient at each point is computed. The convex hull of these gradients serves as an approximation of the ε -Clarke generalised gradient (Burke et al., 2002). The minimum element in the convex hull of these gradients is a descent direction. The gradient sampling algorithm progressively reduces the sampling radius so that the convex hull approximates the Clarke generalised gradient. When the origin is contained in the Clarke generalised gradient there is no direction of descent, and hence the current candidate solution is a local minimum. Gradient sampling achieves almost sure global convergence for functions that are locally Lipschitz continuous and almost everywhere continuously differentiable. It is also well documented that it is an effective optimisation method for functions that are only locally Lipschitz continuous.

4.1 Computational Complexity

In this subsection we analyse the computational complexity of MDP². At each iteration the algorithm projects the data sample onto $\mathbf{v}(\theta)$ which involves $\mathcal{O}(nd)$ operations. To compute the projection index, $\phi(\theta)$, we need to minimise the penalised density integral, $f(\mathbf{v}(\theta), b)$. This can be achieved by first evaluating $f(\mathbf{v}(\theta), b)$ on a grid of m points, to bracket the location of the minimiser, and then applying bisection to compute the minimiser(s) within the desired accuracy. The main computational cost of this procedure is due to the first step which involves m evaluations of a kernel density estimator with n kernels. Using the improved fast Gauss transform (Morariu et al., 2008) this can be performed in $\mathcal{O}(m+n)$ operations, instead of $\mathcal{O}(mn)$. Bisection requires $\mathcal{O}(-\log_2 \varepsilon)$ iterations to locate the minimiser with accuracy ε .

If the minimiser of the penalised density integral $b^* = \arg \min_{b \in A} f(\mathbf{v}(\theta), b)$, is unique the projection index is continuously differentiable at θ . To obtain the derivative of the projection index it is convenient to define the projection function, $F(\mathbf{v}) = (\mathbf{x}_1 \cdot \mathbf{v}, \dots, \mathbf{x}_n \cdot \mathbf{v})^\top$. An application of the chain rule yields,

$$d_b \phi = D_b f(\mathbf{v}(\theta), b^*) = D_P f(\mathbf{v}(\theta), b^*) D_{\mathbf{v}} P D_b \mathbf{v} \quad (26)$$

where the derivative of the projections of the data sample with respect to \mathbf{v} is equal to the data matrix, $D_{\mathbf{v}} P = (\mathbf{x}_1, \dots, \mathbf{x}_n)^\top$; and $D_P \mathbf{v}$ is the derivative of \mathbf{v} with respect to the projection angle, which yields a $d \times (d-1)$ matrix. The computation of the derivative therefore requires $\mathcal{O}(d(n+d))$ operations.

The original GS algorithm requires $\mathcal{O}(d)$ gradient evaluations at each iteration which is costly. Curtis and Que (2013) have developed an adaptive gradient sampling algorithm that requires $\mathcal{O}(1)$ gradient evaluations in each iteration. More recently, Lewis and Overton (2013) have strongly advocated that for the minimisation of nonsmooth, nonconvex, locally

	n	d	c
banknote ^a	1372	4	2
br. cancer ^a	699	9	2
forest ^a	523	27	4
ionosphere ^a	351	33	2
optdigits ^a	5618	64	10
pendigits ^a	10992	16	10
seeds ^a	210	7	3
smartphone ^a	10929	561	12
image seg. ^a	2309	18	7
satellite ^a	6435	36	6
synth ^a	600	60	6
voting ^a	435	16	2
wine ^a	178	13	3
yeast ^b	698	72	5

a. UCI machine learning repository <https://archive.ics.uci.edu/ml/datasets.html>

b. Stanford Yeast Cell Cycle Analysis Project <http://genome-www.stanford.edu/cellcycle/>

Table 1: Details of benchmark data sets: size (n), dimensionality (d), number of clusters (c).

Lipschitz functions, a simple BFGS method using inexact line searches is much more efficient in practice than gradient sampling, although no convergence guarantees have been established for this method. BFGS requires a single gradient evaluation at each iteration and a matrix vector operation to update the Hessian matrix approximation. In our experiments we use the BFGS algorithm.

5. Experimental Results

In this section we assess the empirical performance of MDHs for clustering and semi-supervised classification. We compare performance with existing state-of-the-art methods for both problems on the following 14 benchmark data sets: Banknote authentication (banknote), Breast Cancer Wisconsin original (br. cancer), Forest type mapping (forest), Ionosphere, Optical recognition of handwritten digits (optdigits), Pen-based recognition of hand-written digits (pendigits), Seeds, Smartphone-Based Segmentation of Human Activities and Postural Transitions (smartphone), Statlog Image Segmentation (image seg.), Statlog Landsat Satellite (satellite), Synthetic control chart time series (synth control), Congressional voting records (voting), Wine, and Yeast cell cycle analysis (yeast). Details of these data sets, in terms of their size, n , dimensionality, d and number of clusters, c , can be seen in Table 1.

5.1 Clustering

Since an MDH yields a bi-partition of a data set rather than a complete clustering, we propose two measures to assess the quality of a binary partition of a data set containing an

arbitrary number of clusters. Both take values in $[0, 1]$ with larger values indicating a better partition. These measures are motivated by the fact that a good binary partition should (a) avoid dividing clusters between elements of the partition, and (b) be able to discriminate at least one cluster from the rest of the data. To capture this we modify the cluster labels of the data by assigning each cluster to the element of the binary partition which contains the majority of its members. In the case of a tie the cluster is assigned to the smaller of the two partitions. We thus merge the true clusters into two aggregate clusters, C_1 and C_2 .

The first measure we use is the binary V-measure which is simply the V-measure (Rosenberg and Hirschberg, 2007) computed on C_1, C_2 with respect to the binary partition, which we denote Π_1, Π_2 . The V-measure is the harmonic mean of homogeneity and completeness. For a data set containing clusters C_1, \dots, C_c , partitioned as Π_1, \dots, Π_k , homogeneity is defined as the conditional entropy of the cluster distribution within each partition, Π_i . Completeness is symmetric to homogeneity and measures the conditional entropy of each partition within each cluster, C_j . An important characteristic of the V-measure for evaluating binary partitions is that if the distribution of clusters within each partition is equal to the overall cluster distribution in the data set then the V-measure is equal to zero (Rosenberg and Hirschberg, 2007). This means that if an algorithm fails to distinguish the majority of any of the clusters from the remainder of the data, the binary V-measure returns zero performance. Other evaluation metrics for clustering, such as purity and the Rand index, can assign a high value to such partitions.

To define the second performance measure we first determine the number of correctly and incorrectly classified samples. The error of a binary partition, $E(\Pi_1, \Pi_2)$, given in Equation (27), is defined as the number of elements of each aggregate cluster which are not in the same partition as the majority of their original clusters. In contrast, the success of a partition, $S(\Pi_1, \Pi_2)$, Equation (28), measures the number of samples which are in the same partition as the majority of their original clusters. The Success Ratio, $SR(\Pi_1, \Pi_2)$, Equation (29), captures the extent to which the majority of at least one cluster is well-distinguished from the rest of the data.

$$E(\Pi_1, \Pi_2) = \min \{ |\Pi_1 \cap C_1| + |\Pi_2 \cap C_2|, |\Pi_1 \cap C_2| + |\Pi_2 \cap C_1| \}, \quad (27)$$

$$S(\Pi_1, \Pi_2) = \min \{ \max \{ |\Pi_1 \cap C_1|, |\Pi_1 \cap C_2| \}, \max \{ |\Pi_2 \cap C_1|, |\Pi_2 \cap C_2| \} \}, \quad (28)$$

$$SR(\Pi_1, \Pi_2) = \frac{S(\Pi_1, \Pi_2)}{S(\Pi_1, \Pi_2) + E(\Pi_1, \Pi_2)}. \quad (29)$$

The Success Ratio takes the value zero if an algorithm fails to distinguish the majority of any cluster from the remainder of the data.

5.1.1 PARAMETER SETTINGS FOR MDP²

The two most important settings for the performance of the proposed approach are the initial projection direction, and the choice of α , which controls the width of the interval $F(\mathbf{v})$ within which the optimal hyperplane falls. Despite the ability of the MDP² formulation to mitigate the effect of local minima of the projected density, the problem remains non-convex and local minima in the projection index can still lead to suboptimal performance. We have found that this effect is amplified in general when either or both the number of dimensions, and the number of high density clusters in the data set is large. To better handle the effect

of local optima, we use multiple initialisations and select the MDH that maximises the *relative depth* criterion, defined in Equation (30). The relative depth of an MDH, $H(\mathbf{v}, b)$, is defined as the smaller of the relative differences in the density on the MDH and its two adjacent modes in the projected density,

$$\text{RelativeDepth}(\mathbf{v}, b) = \frac{\min \left\{ \hat{f}(\mathbf{v}, m_l), \hat{f}(\mathbf{v}, m_r) \right\} - \hat{f}(\mathbf{v}, b)}{\hat{f}(\mathbf{v}, b)} \quad (30)$$

where m_l and m_r are the two adjacent modes in the projected density on \mathbf{v} . If an MDH does not separate the modes of the projected density, then its relative depth is set to zero, signalling a failure of MDP2 to identify a meaningful bi-partition. The relative depth is appealing because it captures the fact that a high quality separating hyperplane should have a low density integral, and separate well the modes of the projected density. Note also that the relative depth is equivalent to the inverse of a measure used to define cluster overlap in the context of Gaussian mixtures (Althouiri et al., 2000). In all the reported experiments we initialise MDP2 to the first and second principal component and select the MDH with the largest relative depth. For the data sets listed above it was never the case that both initialisations led to MDHs with zero relative depth.

The choice of α determines the trade-off between a balanced bi-partition and the ability to discover lower density hyperplanes. The difficulties associated with choosing this parameter are illustrated in Figure 4. In each sub-figure the horizontal axis is the candidate projection vector, \mathbf{v} , while the right vertical axis is the direction of maximum variability orthogonal to \mathbf{v} . Points correspond to projections of the data sample onto this two-dimensional space, while colour indicates cluster membership. The solid line depicts the projected density on \mathbf{v} , while the dotted line depicts the penalised function, $f_{CL}(\mathbf{v}, \cdot)$. The scale of both functions is depicted on the left vertical axis. The solid vertical line indicates the MDH along \mathbf{v} . Setting α to a large value can cause MDP2 to focus on hyperplanes that have low density because they partition only a small subset of the data set as shown in Figure 4(a). In contrast smaller values of α may cause the algorithm to disregard valid lower density hyperplane separators (see Figure 4(b)), or for the separating hyperplane to not be a local minimiser of the projected density (see Figure 4(c)).

Rather than selecting a single value for α we recommend solving MDP2 repeatedly for an increasing sequence of values in the range $\{\alpha_{\min}, \alpha_{\max}\}$, where each implementation beyond the first is initialised using the solution to the previous. Setting α_{\min} close to zero forces MDP2 to seek low density hyperplanes that induce a balanced data partition. This tends to find projections which display strong multimodal structure, yet prevents convergence to hyperplanes that have low density because they partition a few observations, as in the case shown in Figure 4(a). Increasing α progressively fine-tunes the location of the MDH. To avoid sensitivity to the value of α_{\max} (set to 0.9) the output of the algorithm is the last hyperplane that corresponds to a minimiser of the projected density. Figure 5 illustrates this approach using the optical recognition of handwritten digits data set from the UCI machine learning repository (Lichman, 2013). Figure 5(a) depicts the projected density on the initial projection direction, which in this case is the second principal component. As shown, the density is unimodal and the clusters are not well separated along this vector. Although not shown, if a large value of α is used from the outset, MDP2 will identify a vector

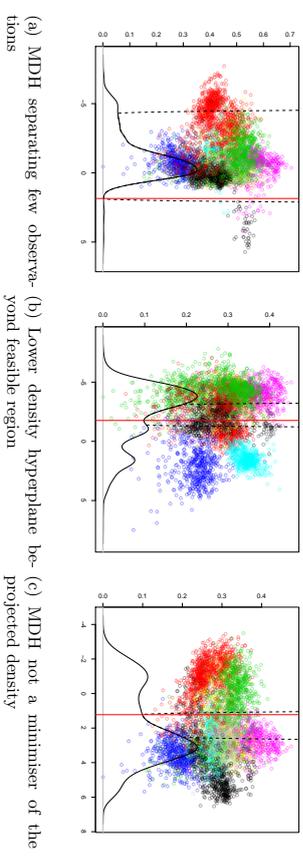


Figure 4: Impact of choice of α on minimum density hyperplane.

along which the projected density is unimodal and skewed. Figure 5(b) shows that after five iterations with $\alpha = 10^{-2}$ MDP2 has identified a projection vector with bimodal density. In subsequent iterations the two modes become more clearly separated, Figure 5(c), while increasing α enables MDP2 to locate an MDH that corresponds to a minimiser of $\hat{f}(\mathbf{v}, b)$, as illustrated in Figure 5(d).

In all experiments we set the bandwidth parameter to $h = 0.9\hat{\sigma}_{p_{C_1}} n^{-1/5}$, where $\hat{\sigma}_{p_{C_1}}$ is the estimated standard deviation of the data projected onto the first principal component. This bandwidth selection rule is recommended when the density being approximated is assumed to be multimodal (Silverman, 1986). The parameter η controls the distance between the minimisers of $\arg \min_{b \in \mathbb{R}} f_{CL}(\mathbf{v}, b)$ and $\arg \min_{b \in F(\mathbf{v})} \hat{f}(\mathbf{v}, b)$, while larger values of ϵ increase the smoothness of the penalised function f_{CL} . Values of η close to zero affect the numerical stability of the one-dimensional optimisation problem, due to the term $\frac{\epsilon}{\eta}$ in f_{CL} becoming very large. We used $\eta = 10^{-2}$ and $\epsilon = 1 - 10^{-6}$ to avoid numerical instability. Beyond these numerical problems the values of η and ϵ do not affect the solutions obtained through MDP2.

5.1.2 PERFORMANCE EVALUATION

We compare the performance of MDP2 for clustering with the following methods:

1. k -means++ (Arthur and Vassilvitskii, 2007), a version of k -means that is guaranteed to be $O(\log k)$ -competitive to the optimal k -means clustering.
2. The adaptive linear discriminant analysis guided k -means (LDA- k m) (Ding and Li, 2007). LDA- k m attempts to discover the most discriminative linear subspace for clustering by iteratively using k -means, to assign labels to observations, and LDA to identify the most discriminative subspace.
3. The principal direction divisive partitioning (PDDP) (Boley, 1998), and the density-enhanced PDDP (dePDDP) (Tasoulis et al., 2010). Both methods project the data onto the first principal component. PDDP splits at the mean of the projections, while dePDDP splits at the lowest local minimum of the one-dimensional density estimator.

space. For the 2-means and LDA-2m algorithm the hyperplane separator bisects the line segment joining the two centroids. iSVR-L directly seeks the maximum margin hyperplane in the original space, while iSVR-G seeks the maximum margin hyperplane in the feature space defined by the Gaussian kernel. PDDP and dePDDP use a hyperplane whose normal vector is the first principal component. PDDP uses a fixed split point while dePDDP uses the hyperplane with minimum density along the fixed projection direction.

Table 2 reports the performance of the considered methods with respect to the success ratio (SR) and the binary V-measure (V-m) on the fourteen data sets. In addition Figures 6(a) and 6(b) provide summaries of the overall performance on all data sets using boxplots of the raw performance measures as well as the associated *regret*. The regret of an algorithm on a given data set is defined as the difference between the best performance attained on this data set and the performance of this algorithm. By comparing against the best performing clustering algorithm regret accommodates for differences in difficulty between clustering problems, while also making use of the magnitude of performance differences between algorithms. The distribution of performance with respect to both SR and V-m is negatively skewed for most methods, and as a result the median is higher than the mean (indicated with a red dot).

It is clear from Table 2 that no single method is consistently superior to all others, although MDP² achieves the highest or tied highest performance on seven data sets (more than any other method). More importantly MDP² is among the best performing methods in almost all cases. This fact is better captured by the regret distributions in Figure 6(b). Here we see that the average, median, and maximum regret of MDP² is substantially lower than any of the competing methods. In addition MDP² achieves the highest mean and median performance with respect to both SR and V-m, while also having much lower variability in performance when compared with most other methods.

Pairwise comparisons between MDP² and other methods reveal some less obvious facts. SCn achieves higher performance than MDP² in more examples (six) than any other competing method, however it is much less consistent in its performance, obtaining very poor performance on five of the data sets. The iSVR maximum margin clustering approach is arguably the closest competitor to MDP². iSVR-L and iSVR-G achieve the second and third highest average performance with respect to V-m and SR respectively. The PDDP algorithm is the second best performing method on average with respect to SR, but performs poorly with respect to V-m. The density enhanced variant, dePDDP, performs on average much worse than MDP². This approach is similarly motivated by obtaining hyperplanes with low density integral, and its low average performance indicates the usefulness of searching for high quality projections as opposed to always using the first principal component. Finally, neither of the k -means variants appears to be competitive with MDP² in general.

5.2 Semi-Supervised Classification

In this section we evaluate MDHs for semi-supervised classification. We compare MDHs against three state-of-the-art semi-supervised classification methods: Laplacian Regularised Support Vector Machines (LapSVM) (Belkin et al., 2006), Simple Semi-Supervised Learning (SSSL) (Ji et al., 2012), and Correlated Nystrom Views (XNV) (McWilliams et al., 2013). For all methods the inner product kernel was used to render the resulting classifiers linear,

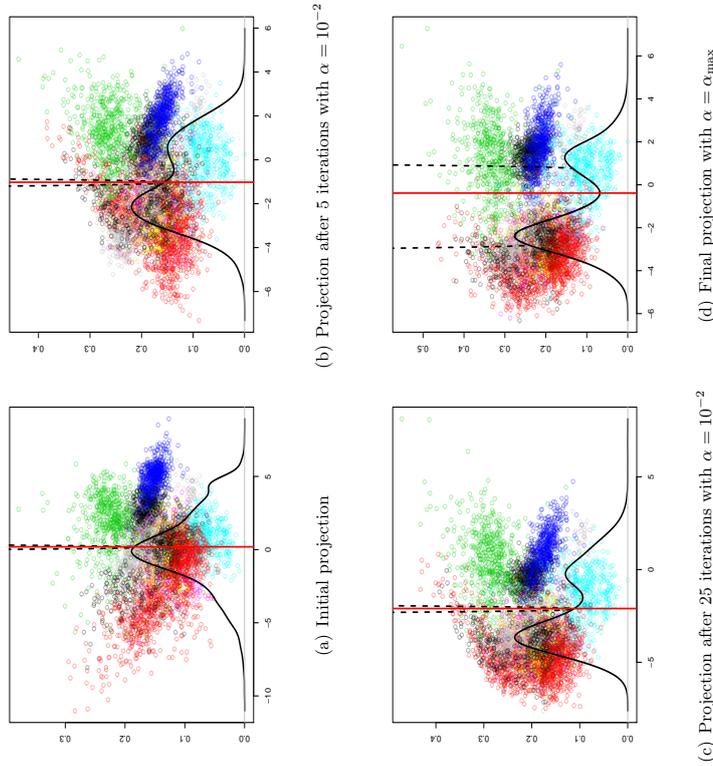


Figure 5: Evolution of the minimum density hyperplane through consecutive iterations.

4. The iterative support vector regression algorithm for MMC (Zhang et al., 2009) using the inner product and Gaussian kernel, iSVR-L and iSVR-G respectively. Both are initialised with the output of 2-means++.
5. Normalised cut spectral clustering (SCn) (Ng et al., 2002) using the Gaussian affinity function, and the automatic bandwidth selection method of Zelnik-Manor and Perona (2004). This choice of kernel and bandwidth produced substantially better performance than alternative choices considered. For data sets that are too large for the eigen decomposition of the Gram matrix to be feasible we employed the Nystrom method (Fowlkes et al., 2004).

We also considered the density-based clustering algorithm PdfCluster (Menardi and Azzalini, 2014), but this algorithm could not be executed on the larger data sets and so its performance is not reported in this paper. With the exception of SCn and iSVR-G, the methods considered bi-partition the data through a hyperplane in the original feature

	MDP ²		iSVR-L		iSVR-G		SC _h		LDA-2m		2-means++		PDDP		dePDDP	
Data set	SR	V-m	SR	V-m	SR	V-m	SR	V-m	SR	V-m	SR	V-m	SR	V-m	SR	V-m
banknote	0.79	0.55	0	0	0.35	0	0.46	0.10	0	0.01	0.37	0.01	0.40	0.03	0	0.03
br. cancer	0.91	0.79	0.73	0.56	0.73	0.56	0	0.13	0.87	0.71	0.87	0.72	0.91	0.78	0.90	0.77
forest	0.78	0.67	0.90	0.72	0.91	0.74	0.56	0.41	0.76	0.63	0.72	0.58	0.64	0.36	0	0
image-seg.	0.89	0.72	0.82	0.59	0.88	0.71	0.92	0.87	0.78	0.58	0.78	0.71	0.87	0.67	1	1
ionosphere	0.48	0.13	0.47	0.13	0.47	0.13	0.55	0.22	0.47	0.12	0.47	0.12	0.47	0.12	0.42	0.09
optdigits	0.93	0.85	0.63	0.29	0.82	0.60	0	0	0.81	0.62	0.92	0.82	0.68	0.30	0	0
pendigits	0.74	0.39	0.79	0.55	0.88	0.68	0.80	0.68	0.79	0.55	0.78	0.57	0.79	0.54	0.61	0.42
satellite	0.89	0.75	0.73	0.40	0.73	0.40	0.92	0.86	0.73	0.40	0.87	0.81	0.71	0.37	0	0
seeds	0.88	0.73	0.71	0.53	0.71	0.53	0.89	0.76	0.96	0.90	0.86	0.70	0.75	0.59	0.73	0.60
smartphone	0.99	0.97	0.99	0.95	0.99	0.96	0.99	0.94	0.99	0.97	0.99	0.94	0.99	0.95	0	0
synth	0.98	0.94	0.91	0.83	0.91	0.83	1	1	0.88	0.76	1	1	0.69	0.51	1	1
voting	0.70	0.43	0.46	0.09	0	0	0.05	0.69	0.41	0	0	0	0.70	0.40	0.68	0.38
wine	0.77	0.61	0.70	0.52	0.69	0.50	0.67	0.48	0.66	0.48	0.68	0.49	0.65	0.46	0.68	0.49
yeast	0.92	0.76	0.89	0.68	0.91	0.72	0.84	0.61	0.86	0.63	0.91	0.73	0.87	0.65	0	0
Average Improvement	0.13	0.18	0.12	0.14	0.22	0.16	0.10	0.11	0.10	0.08	0.11	0.18	0.40	0.32		

Table 2: Performance on the task of binary partitioning. (Ties in best performance were resolved by considering more decimal places)

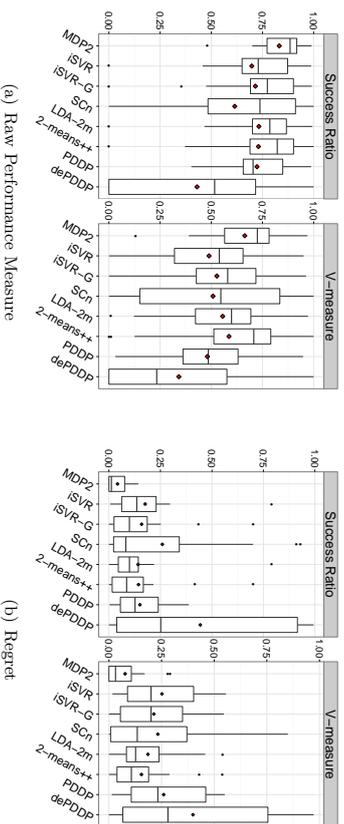


Figure 6: Performance and Regret Distributions for all Methods Considered

and thereby comparable to our method. As the MDH is asymptotically equivalent to a linear S³VM we also considered the continuous formulation for the estimation of a S³VM proposed by Chapelle and Zien (2005). These results are omitted as this method was not competitive on any of the considered data sets.

5.2.1 PARAMETER SETTINGS FOR MDP²

The existence of a few labelled examples enables an informed initialisation of MDP². We consider the first and second principal components as well as the weight vector of a linear SVM trained on the labelled examples only, and initialise MDP² with the vector that minimises the value of the projection index, ϕ_{SVC} . The penalty parameter γ is first set to 0.1 and with this setting α is progressively increased in the same way as for clustering. After this, α is kept at α_{max} and γ is increased to 1 and then 10. Thus the emphasis is initially on finding a low density hyperplane with respect to the marginal density $\hat{p}(\mathbf{x})$. As the algorithm progresses the emphasis on correctly classifying the labelled examples increases, so as to obtain a hyperplane with low training error within the region of low density already determined.

5.2.2 PERFORMANCE EVALUATION

To assess the effect on performance of the number of labelled examples, ℓ , we consider a range of values. We compare the methods using the subset of data sets used in the previous section in which the size of the smallest class exceeds 100. In total eight data sets are used. For each value of ℓ , 30 random partitions into labelled and unlabelled data are considered. As classes are balanced in the data sets considered, performance is measured only in terms of classification error on the unlabelled data. For data sets with more than two classes all pairwise combinations of classes are considered and aggregate performance is reported.

Figure 7 provides plots of the median and interquartile range of the classification error for values of ℓ between 5 and 100 for the four data sets with two classes. Overall MDP² appears to be most competitive when the number of labelled examples is small. In addition, MDP² is comparable with the best performing method in almost every case. The only exception is the ionosphere data set where LapSVM outperforms MDP² for all values of ℓ . Figure 8 provides plots of the median and interquartile range of the aggregate classification error on data sets containing more than two classes. As these data sets are larger we consider up to 300 labelled examples. Note that the interquartile range for XNV is not depicted for the satellite data set. The variability of performance of XNV on this data set was so high that including the interquartile range would obscure all other information in the figure. MDP² exhibits the best performance overall, and obtains the lowest median classification error, or tied lowest, for all data sets and values of ℓ .

5.3 Summary of Experimental Results

We evaluated the performance of the MDP² formulation for finding MDHs for both clustering and semi-supervised classification, on a large collection of benchmark data sets, and in comparison with state-of-the-art methods for both problems.

For clustering, we found that no single method was consistently superior to all others. This is a result of the vastly differing nature of the data sets in terms of size, dimensionality, number and shape of clusters, etc. MDP² achieved the best performance on more data sets than any of the competing methods, and importantly was competitive with the best performing method in almost every data set considered. All other methods performed poorly in at least as many examples. Boxplots of both the raw performance and performance regret, which measures the difference between each method and the best performing method on each

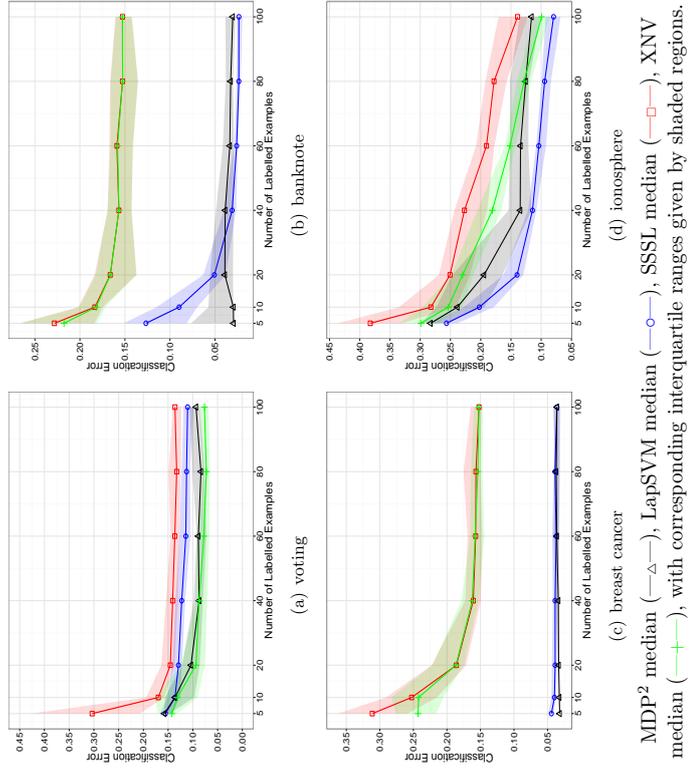


Figure 7: Classification error for different number of labelled examples for data sets with two clusters.

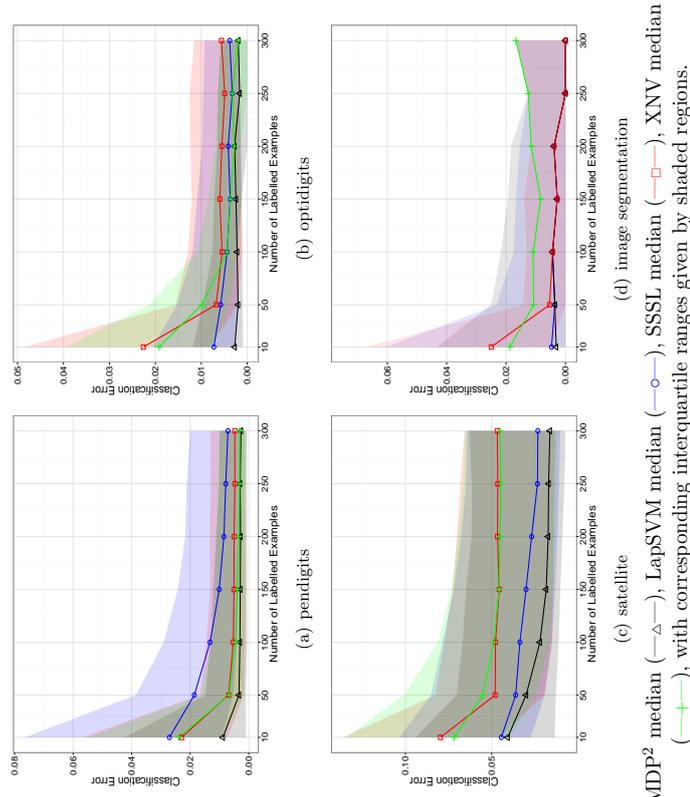


Figure 8: Classification error for different numbers of labelled examples over all pairwise combinations of classes.

data set, allowed us to summarise the comparative performance of the different methods across data sets. The mean and median raw performance of MDP² is substantially higher than the next best performing method, and the regret is also substantially lower.

In the case of semi-supervised classification it was apparent that MDP² is extremely competitive when the number of labelled examples is (very) small, but that in some cases its performance does not improve as much as that of the other methods considered, when the labelled examples become more abundant. Our experiments suggest that overall MDP² is very competitive with the state-of-the-art for semi-supervised classification problems.

6. Conclusions

We proposed a new hyperplane classifier for clustering and semi-supervised classification. The proposed approach is motivated by determining low density linear separators of the high-density clusters within a data set. This is achieved by minimising the integral of the empirical density along the hyperplane, which is computed through kernel density estimation. To the best of our knowledge this is the first direct implementation of the low density separation assumption that underlies high-density clustering and numerous influential semi-supervised classification methods. We show that the minimum density hyperplane is asymptotically connected with maximum margin hyperplane, thereby establishing an important link between the proposed approach, maximum margin clustering, and semi-supervised support vector machines.

The proposed formulation allows us to evaluate the integral of the density on a hyperplane by projecting the data onto the vector normal to the hyperplane, and estimating a univariate kernel density estimator. This enables us to apply our method effectively and efficiently on data sets of much higher dimensionality than is generally possible for density based clustering methods. To mitigate the problem of convergence to locally optimal solutions we proposed a projection pursuit formulation.

We evaluated the minimum density hyperplane approach on a large collection of benchmark data sets. The experimental results obtained indicate that the method is competitive with state-of-the-art methods for clustering and semi-supervised classification. Importantly the performance of the proposed approach displays low variability across a variety of data sets, and is robust to differences in data size, dimensionality, and number of clusters. In the context of semi-supervised classification, the proposed approach shows especially good performance when the number of labelled data is small.

Acknowledgments

We would like to thank the reviewers for their insightful comments which substantially improved this paper. We also thank Prof. David Leslie, and Dr. Teemu Roos for valuable comments and suggestions on this work. Nicos Pavlidis would like to thank the Isaac Newton Institute for Mathematical Sciences, Cambridge, for support and hospitality during the programme ‘Inference for Change-Point and Related Processes’, where part of the work on this paper was undertaken. David Hoemeyer gratefully acknowledges the support of the EPSRC funded EP/H023151/1 STOR-i centre for doctoral training, as well as the Openheimer

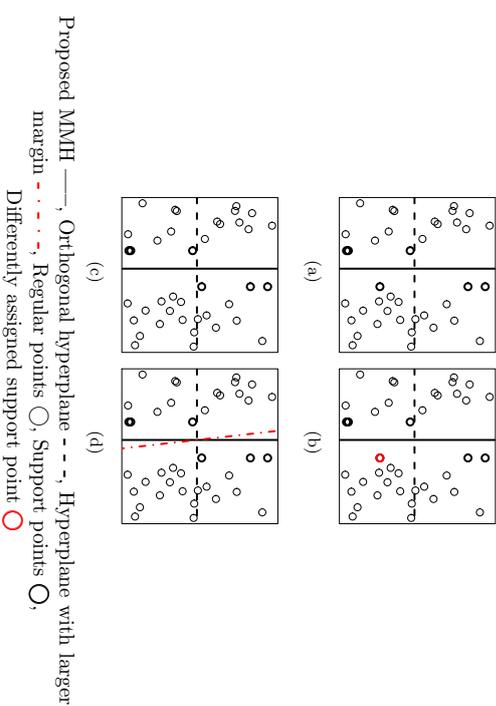


Figure 9: Two dimensional illustration of Lemma 9

Memorial Trust. The underlying code and data are openly available from Lancaster University data repository at <http://dx.doi.org/10.17635/lancaster/researchdata/97>.

Appendix A. Proof of Theorem 5

Before proving Theorem 5 we require the following two technical lemmata, which establish some algebraic properties of the maximum margin hyperplane. The following lemma shows that any hyperplane orthogonal to the maximum margin hyperplane results in a different partition of the support points of the maximum margin hyperplane. The proof relies on the fact that if this statement does not hold then a hyperplane with larger margin exists which is a contradiction. Figure 9 provides an illustration of why this result holds. (a) Any hyperplane orthogonal to MMH generates a different partition of the support points of MMH, e.g., the point highlighted in red in (b) is grouped with the lower three by the dotted line but with the upper two by the solid line, the MMH. If an orthogonal hyperplane can generate the same partition (c), then a larger margin hyperplane than the proposed MMH exists (d).

Lemma 9 *Suppose there is a unique hyperplane in F with maximum margin, which can be parameterised by $(\mathbf{v}^m, b^m) \in \mathcal{S}^{d-1} \times \mathbb{R}$. Let $M = \text{margin } H(\mathbf{v}^m, b^m)$, $C^+ = \{\mathbf{x} \in \mathcal{X} \mid \mathbf{v}^m \cdot \mathbf{x} - b^m = M\}$ and $C^- = \{\mathbf{x} \in \mathcal{X} \mid b^m - \mathbf{v}^m \cdot \mathbf{x} = M\}$. Then, $\forall \mathbf{w} \in \text{Null}(\mathbf{v}^m)$, $c \in \mathbb{R}$ either $\min\{\mathbf{w} \cdot \mathbf{x} - c \mid \mathbf{x} \in C^+\} \leq 0$, or $\max\{\mathbf{w} \cdot \mathbf{x} - c \mid \mathbf{x} \in C^-\} \geq 0$.*

Proof

Suppose the result does not hold, then $\exists(\mathbf{w}, c)$ with $\|\mathbf{w}\| = 1, \mathbf{w} \cdot \mathbf{v}^m = 0$ and $\min\{\mathbf{w} \cdot \mathbf{x} - c | \mathbf{x} \in C^+\} > 0$ and $\max\{\mathbf{w} \cdot \mathbf{x} - c | \mathbf{x} \in C^-\} < 0$. Let $m = \min\{\mathbf{w} \cdot \mathbf{x} - c | \mathbf{x} \in C^+ \cup C^-\}$. Define $\lambda = \frac{m}{2M} < 1$. Define $\mathbf{u} = \frac{1}{\sqrt{\lambda^2 + (1-\lambda)^2}}(\lambda \mathbf{w} + (1-\lambda)\mathbf{v}^m)$ and $d = \frac{\lambda c + (1-\lambda)b^m}{\sqrt{\lambda^2 + (1-\lambda)^2}}$. By construction $\|\mathbf{u}\| = 1$. For any $\mathbf{x}_+ \in C^+$ we have,

$$\begin{aligned} \mathbf{u} \cdot \mathbf{x}_+ - d &= \frac{\lambda(\mathbf{w} \cdot \mathbf{x}_+ - c) + (1-\lambda)(\mathbf{v}^m \cdot \mathbf{x}_+ - b^m)}{\sqrt{\lambda^2 + (1-\lambda)^2}} \\ &\geq \frac{\lambda m + (1-\lambda)M}{\sqrt{\lambda^2 + (1-\lambda)^2}} \\ &= \frac{m^2 + 2M^2 - Mm}{\sqrt{m^2 + (2M - m)^2}} \\ &> M. \end{aligned}$$

Similarly one can show that $d - \mathbf{u} \cdot \mathbf{x}_- > M$ for any $\mathbf{x}_- \in C^-$, meaning that (\mathbf{u}, d) achieves a larger margin on C^+ and C^- than (\mathbf{v}^m, b^m) , a contradiction. \blacksquare

The next lemma uses the above result to provide an upper bound on the distance between pairs of support points projected onto any vector, in terms of the angle between that vector and \mathbf{v}^m .

Lemma 10 *Suppose there is a unique hyperplane in F with maximum margin, which can be parameterised by $(\mathbf{v}^m, b^m) \in \mathcal{S}^{d-1} \times \mathbb{R}$. Define $M = \text{margin } H(\mathbf{v}^m, b^m)$, $C^+ = \{\mathbf{x} \in \mathcal{X} | \mathbf{v}^m \cdot \mathbf{x} - b^m = M\}$, and $C^- = \{\mathbf{x} \in \mathcal{X} | b^m - \mathbf{v}^m \cdot \mathbf{x} = M\}$. There is no vector $\mathbf{w} \in \mathbb{R}^d$ for which $\mathbf{w} \cdot \mathbf{x}_+ - \mathbf{w} \cdot \mathbf{x}_- > 2M\mathbf{v}^m \cdot \mathbf{w}$ for all pairs $\mathbf{x}_+ \in C^+, \mathbf{x}_- \in C^-$.*

Proof

Suppose such a vector exists. Define $\mathbf{w}' = \mathbf{w} - (\mathbf{v}^m \cdot \mathbf{w})\mathbf{v}^m$. By construction $\mathbf{w}' \in \text{Null}(\mathbf{v}^m)$. For any pair $\mathbf{x}_+ \in C^+, \mathbf{x}_- \in C^-$ we have

$$\begin{aligned} \mathbf{w}' \cdot \mathbf{x}_+ - \mathbf{w}' \cdot \mathbf{x}_- &= \mathbf{w} \cdot \mathbf{x}_+ - (\mathbf{v}^m \cdot \mathbf{w})\mathbf{v}^m \cdot \mathbf{x}_+ - \mathbf{w} \cdot \mathbf{x}_- + (\mathbf{v}^m \cdot \mathbf{w})\mathbf{v}^m \cdot \mathbf{x}_- \\ &> \mathbf{v}^m \cdot \mathbf{w}(2M - \mathbf{v}^m \cdot \mathbf{x}_+ + b^m - b^m + \mathbf{v}^m \cdot \mathbf{x}_-) \\ &= 0. \end{aligned}$$

Define $c := \frac{1}{2}(\min\{\mathbf{w}' \cdot \mathbf{x}_+ | \mathbf{x}_+ \in C^+\} + \max\{\mathbf{w}' \cdot \mathbf{x}_- | \mathbf{x}_- \in C^-\})$. Then $\min\{\mathbf{w}' \cdot \mathbf{x}_+ - c | \mathbf{x}_+ \in C^+\} > 0$ and $\max\{\mathbf{w}' \cdot \mathbf{x}_- - c | \mathbf{x}_- \in C^-\} < 0$, a contradiction. \blacksquare

We are now in a position to provide the main proof of this appendix. The theorem states that if the maximum margin hyperplane is unique, and can be parameterised by $(\mathbf{v}^m, b^m) \in \mathcal{S}^{d-1} \times \mathbb{R}$, then

$$\lim_{h \rightarrow 0^+} \min\{\|(\mathbf{v}_h^*, b_h^*) - (\mathbf{v}^m, b^m)\|, \|(\mathbf{v}_h^*, b_h^*) - (\mathbf{v}^m, b^m)\|\} = 0,$$

where $\{H(\mathbf{v}_h^*, b_h^*)\}_h$ is any collection of minimum density hyperplanes indexed by their bandwidth $h > 0$.

Proof of Theorem 5

Define $M = \text{margin } H(\mathbf{v}^m, b^m)$, $C^+ = \{\mathbf{x} \in \mathcal{X} | \mathbf{v}^m \cdot \mathbf{x} - b^m = M\}$ and $C^- = \{\mathbf{x} \in \mathcal{X} | b^m - \mathbf{v}^m \cdot \mathbf{x} = M\}$. Let $B = \max\{\|\mathbf{x}\| | \mathbf{x} \in \mathcal{X}\}$. Take any $\epsilon > 0$ and set $0 < \delta$ to satisfy $\frac{2\delta}{M}(1+B^2) + 2B\delta^{3/2}\sqrt{\frac{2}{M}} + \delta^2 = \epsilon^2$. Now, suppose $(\mathbf{w}, c) \in \mathcal{S}^{d-1} \times \mathbb{R}$ satisfies,

$$\mathbf{w} \cdot \mathbf{x}_+ - c > M - \delta, \forall \mathbf{x}_+ \in C^+ \text{ and } c - \mathbf{w} \cdot \mathbf{x}_- > M - \delta, \forall \mathbf{x}_- \in C^-.$$

By Lemma 10 we know that $\exists \mathbf{x}_+ \in C^+, \mathbf{x}_- \in C^-$ s.t. $\mathbf{w} \cdot \mathbf{x}_+ - \mathbf{w} \cdot \mathbf{x}_- \leq 2M\mathbf{v}^m \cdot \mathbf{w}$. Thus

$$\begin{aligned} \mathbf{v}^m \cdot \mathbf{w} &\geq \frac{\mathbf{w} \cdot \mathbf{x}_+ - \mathbf{w} \cdot \mathbf{x}_-}{2M} \\ &= \frac{2M}{\mathbf{w} \cdot \mathbf{x}_+ - c + c - \mathbf{w} \cdot \mathbf{x}_-} \\ &> \frac{2M - 2\delta}{2M} = 1 - \frac{\delta}{M}. \end{aligned}$$

Thus $\|\mathbf{v}^m - \mathbf{w}\|^2 < \frac{2\delta}{M}$. Now, for each $\mathbf{x}_+ \in C^+, \mathbf{v}^m \cdot \mathbf{x}_+ - b = M$ and for each $\mathbf{x}_- \in C^-, b - \mathbf{v}^m \cdot \mathbf{x}_- = M$. Thus for any such $\mathbf{x}_+, \mathbf{x}_-$ we have,

$$\begin{aligned} M - \delta + \mathbf{w} \cdot \mathbf{x}_- &< c < \mathbf{w} \cdot \mathbf{x}_+ - M + \delta, \\ b^m - \mathbf{v}^m \cdot \mathbf{x}_- - \delta + \mathbf{w} \cdot \mathbf{x}_- &< c < \mathbf{w} \cdot \mathbf{x}_+ - \mathbf{v}^m \cdot \mathbf{x}_+ + b^m + \delta, \\ b^m - \delta - (\mathbf{v}^m - \mathbf{w}) \cdot \mathbf{x}_- &< c < b^m + \delta + (\mathbf{w} - \mathbf{v}^m) \cdot \mathbf{x}_+, \\ b^m - \delta - B\|\mathbf{v}^m - \mathbf{w}\| &< c < b^m + \delta + B\|\mathbf{w} - \mathbf{v}^m\|, \\ |c - b^m| &< |\delta + B\|\mathbf{w} - \mathbf{v}^m\||. \end{aligned}$$

We can now bound the distance between (\mathbf{w}, c) and (\mathbf{v}^m, b^m) ,

$$\begin{aligned} \|(\mathbf{v}^m, b^m) - (\mathbf{w}, c)\|^2 &= \|\mathbf{v}^m - \mathbf{w}\|^2 + |b^m - c|^2 \\ &< \|\mathbf{v}^m - \mathbf{w}\|^2(1+B^2) + 2B\delta\|\mathbf{v}^m - \mathbf{w}\| + \delta^2 \\ &< \frac{2\delta}{M}(1+B^2) + 2B\delta\sqrt{\frac{2\delta}{M}} + \delta^2 \\ &= \epsilon^2. \end{aligned}$$

We have shown that for any hyperplane $H(\mathbf{w}, c)$ that achieves a margin larger than $M - \delta$ on the support points of the maximum margin hyperplane, $\mathbf{x} \in C^+ \cup C^-$, the distance between (\mathbf{w}, c) and (\mathbf{v}^m, b^m) is less than ϵ . Equivalently, any hyperplane $H(\mathbf{w}, c)$ such that $\|(\mathbf{w}, c) - (\mathbf{v}^m, b^m)\| > \epsilon$ has a margin less than $M - \delta$, as $\min\{\mathbf{w} \cdot \mathbf{x} - c | \mathbf{x} \in C^+ \cup C^-\} < M - \delta$. By symmetry, the same holds for any (\mathbf{w}, c) within distance ϵ of $(-\mathbf{v}^m, -b^m)$.

By Lemma 4 $\exists h_1 > 0$ such that for all $h \in (0, h_1)$, the minimum density hyperplane for h , $H(\mathbf{v}_h^*, b_h^*)$, induces the same partition of \mathcal{X} as the maximum margin hyperplane, $H(\mathbf{v}^m, b^m)$. By Lemma 3 $\exists h_2 > 0$ such that for all $h \in (0, h_2)$, $\text{margin } H(\mathbf{v}_h^*, b_h^*) > M - \delta$. Therefore for $h \in (0, \min\{h_1, h_2\})$, $\min\{\|(\mathbf{v}_h^*, b_h^*) - (\mathbf{v}^m, b^m)\|, \|(\mathbf{v}_h^*, b_h^*) - (-\mathbf{v}^m, -b^m)\|\} < \epsilon$. Since $\epsilon > 0$ was arbitrarily chosen, this gives the result. \blacksquare

References

- E. Aïmeur, S. Wang, and D. Ziou. On comparison of clustering techniques for histogram pdf estimation. *Pattern Recognition and Image Analysis*, 10(2):206–217, 2000.
- D. Arthur and S. Vassilvitskii. *k*-means++: The advantages of careful seeding. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1027–1035, 2007.
- A. Azzalini and N. Torelli. Clustering via nonparametric density estimation. *Statistics and Computing*, 17(1):71–80, 2007.
- M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labelled and unlabeled examples. *Journal of Machine Learning Research*, 7:2389–2434, 2006.
- S. Ben-David, T. Lu, D. Pál, and M. Sotáková. Learning low-density separators. In D. van Dyk and M. Welling, editors, *International Conference on Artificial Intelligence and Statistics (AISTATS)*, JMLR Workshop and Conference Proceedings, pages 25–32, 2009.
- C. Berge. *Topological Spaces*. Macmillan, New York, 1963.
- D. Boley. Principal direction divisive partitioning. *Data Mining and Knowledge Discovery*, 2(4):325–344, 1998.
- J. F. Bonnans and A. Shapiro. *Perturbation Analysis of Optimization Problems*. Springer Series in Operations Research. Springer, 2000.
- J. V. Burke, A. S. Lewis, and M. L. Overton. Approximating subdifferentials by random sampling of gradients. *Mathematics of Operations Research*, 27(3):567–584, 2002.
- J. V. Burke, A. S. Lewis, and M. L. Overton. A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. *SIAM Journal on Optimization*, 15(3):751–779, 2006.
- V. Castelli and T. M. Cover. On the exponential value of labeled samples. *Pattern Recognition Letters*, 16(1):105–111, 1995.
- V. Castelli and T. M. Cover. The relative value of labeled and unlabeled samples in pattern recognition with an unknown mixing parameter. *IEEE Transactions on Information Theory*, 42(6):2102–2117, 1996.
- O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In R. G. Cowell and Z. Ghahramani, editors, *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 57–64, 2005.
- A. Cuevas, M. Febrero, and R. Fraiman. Estimating the number of clusters. *Canadian Journal of Statistics*, 28(2):367–382, 2000.
- A. Cuevas, M. Febrero, and R. Fraiman. Cluster analysis: a further approach based on density estimation. *Computational Statistics and Data Analysis*, 36(4):441–459, 2001.
- F. E. Curtis and X. Que. An adaptive gradient sampling algorithm for nonsmooth optimization. *Optimization Methods and Software*, 28(6):1302–1324, 2013.
- C. Ding and T. Li. Adaptive dimension reduction using discriminant analysis and *k*-means clustering. In *International Conference on Machine Learning (ICML)*, pages 521–528, 2007.
- C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the Nystrom method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):214–225, 2004.
- P. Fränti and O. Virmajoki. Iterative shrinking method for clustering problems. *Pattern Recognition*, 39(5):761–775, 2006.
- J. A. Hartigan. *Clustering Algorithms*. Wiley Series in Probability and Mathematical Statistics. Wiley, New York, 1975.
- M. Ji, T. Yang, B. Lin, R. Jin, and J. Han. A simple algorithm for semi-supervised learning with improved generalization error bound. In J. Langford and J. Pheasant, editors, *International Conference on Machine Learning (ICML)*, pages 1223–1230, 2012.
- T. Joachims. Transductive inference for text classification using support vector machines. In *International Conference on Machine Learning (ICML)*, volume 99, pages 200–209, 1999.
- A. Lewis and M. Overton. Nonsmooth optimization via quasi-Newton methods. *Mathematical Programming*, 141:135–163, 2013.
- M. Lichman. UCI machine learning repository, 2013. URL <http://archive.ics.uci.edu/ml>.
- B. McWilliams, D. Balduzzi, and J. M. Buhmann. Correlated random features for fast semi-supervised learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 440–448, 2013.
- G. Menardi and A. Azzalini. An advancement in clustering via nonparametric density estimation. *Statistics and Computing*, 24(5):753–767, 2014.
- V. I. Morariu, B. V. Srinivasan, V. C. Raykar, R. Duraiswami, and L. S. Davis. Automatic online tuning for fast Gaussian summation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1113–1120, 2008.
- A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: analysis and an algorithm. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 14, pages 849–856, 2002.
- E. Polak. On the mathematical foundations of nondifferentiable optimization in engineering design. *SIAM Review*, 29(1):21–89, 1987.
- P. Rigollet. Generalization error bounds in semi-supervised classification under the cluster assumption. *Journal of Machine Learning Research*, 8:1369–1392, 2007.

- A. Rinaldo and L. Wasserman. Generalized density clustering. *The Annals of Statistics*, 38(5):2678–2722, 2010.
- A. Rosenberg and J. Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, volume 7, pages 410–420, 2007.
- B. W. Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.
- A. Singh, R. D. Nowak, and X. Zhu. Unlabeled data: Now it helps, now it doesn't. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems (NIPS)*, pages 1513–1520, 2009.
- W. Stuetzle and R. Nugent. A generalized single linkage method for estimating the cluster tree of a density. *Journal of Computational and Graphical Statistics*, 19(2):397–418, 2010.
- S. K. Tasoulis, D. K. Tasoulis, and V. P. Plagianakos. Enhancing principal direction divisive clustering. *Pattern Recognition*, 43(10):3391–3411, 2010.
- S. Tong and D. Koller. Restricted Bayes optimal classifiers. In *National Conference on Artificial Intelligence (AAAI)*, pages 658–664, 2000.
- V. Vapnik and A. Sterin. On structural risk minimization or overall risk in a problem of pattern recognition. *Automation and Remote Control*, 40(3):1495–1503, 1977.
- G. Walthier. Granulometric smoothing. *The Annals of Statistics*, 25(6):2273–2299, 1997.
- P. Wolfe. On the convergence of gradient methods under constraint. *IBM Journal of Research and Development*, pages 407–411, 1972.
- L. Xu, J. Neufeld, B. Larson, and D. Schuurmans. Maximum margin clustering. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 17, pages 1537–1544, 2004.
- L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1601–1608, 2004.
- K. Zhang, I. W. Tsang, and J. T. Kwok. Maximum margin clustering made practical. *IEEE Transactions on Neural Networks*, 20(4):583–596, 2009.

Theoretical Analysis of the Optimal Free Responses of Graph-Based SFA for the Design of Training Graphs

Alberto N. Escalante-B.

Laurenz Wiskott

Theory of Neural Systems

Institut für Neuroinformatik

Ruhr-University Bochum

Bochum D-44801, Germany

ALBERTO.ESCALANTE@INI.RUB.DE

LAURENZ.WISKOTT@INI.RUB.DE

Editor: Zhuowen Tu

Abstract

Slow feature analysis (SFA) is an unsupervised learning algorithm that extracts slowly varying features from a multi-dimensional time series. Graph-based SFA (GSFA) is an extension to SFA for supervised learning that can be used to successfully solve regression problems if combined with a simple supervised post-processing step on a small number of slow features. The objective function of GSFA minimizes the squared output differences between pairs of samples specified by the edges of a structure called training graph. The edges of current training graphs, however, are derived only from the relative order of the labels. Exploiting the exact numerical value of the labels enables further improvements in label estimation accuracy.

In this article, we propose the exact label learning (ELL) method to create a more precise training graph that encodes the desired labels explicitly and allows GSFA to extract a normalized version of them directly (i.e., without supervised post-processing). The ELL method is used for three tasks: (1) We estimate gender from artificial images of human faces (regression) and show the advantage of coding additional labels, particularly skin color. (2) We analyze two existing graphs for regression. (3) We extract *compact* discriminative features to classify traffic sign images. When the number of output features is limited, such compact features provide a higher classification rate compared to a graph that generates features equivalent to those of nonlinear Fisher discriminant analysis. The method is versatile, directly supports multiple labels, and provides higher accuracy compared to current graphs for the problems considered.

Keywords: slow feature analysis, nonlinear regression, image analysis, pattern recognition, many classes

1. Introduction

The slowness principle is one of the learning paradigms that might explain the self-organization of neurons in the brain to extract invariant representations (e.g. Franzius et al., 2007). This principle operates on an abstract level of information processing and postulates that perceived information relevant to a subject typically changes much slower than individual sensory components—e.g., the position of a moth changes slower than the quickly changing neural activations in the retina of a frog observing it.

The slowness principle was probably first formulated by Hinton (1989), and early online learning rules were developed by Földiák (1991) and Mitchison (1991). The first closed-form algorithm is referred to as slow feature analysis (SFA, Wiskott, 1998; Wiskott and Sejnowski, 2002). Given a multi-dimensional time series (i.e., a sequence of samples), SFA finds an instantaneous mapping from the input samples to output features that change as slowly as possible within a given feature space. The objective function of SFA requires the minimization of the average squared differences of consecutive output values. Thus, SFA is especially useful for extracting slowly changing hidden parameters of the data.

Although SFA is unsupervised, it has also been used to solve supervised learning tasks, where it operates as a dimensionality reduction algorithm that is complemented by a supervised algorithm on a small number of slow features. This is motivated by the idea that samples originating at a similar time (e.g., consecutive samples) are likely to have similar labels due to physical and biological constraints on the subject and the environment. Thus, the temporal arrangement of the samples provides a weak form of supervised information.

Recently, an extension of SFA for classification and regression, called graph-based SFA (GSFA, Escalante-B. and Wiskott, 2013), has been proposed, which explicitly exploits the available labels. The training data of GSFA are organized in a graph structure called training graph, in which the vertices are the samples and the edge weights represent similarities between the corresponding labels, where each sample typically has several connections. The objective function of GSFA is similar to that of SFA, except that the pairs of samples need not be temporally consecutive, are weighted, and are indicated by the edges of the graph.

Typically, GSFA is more effective than SFA at extracting a set of features that tend to concentrate the label information, allowing accurate prediction of the labels from a few features, and implicitly solving the supervised learning problem. The resulting (low-dimensional) output features can then be easily post-processed by standard supervised algorithms, such as a regression method based on a Gaussian classifier (Escalante-B. and Wiskott, 2012) or ordinary least squares, to generate the final label estimation.

The main type of application addressed in this article is the solution of regression problems on high-dimensional data with hierarchical GSFA (HGSFA, see Section 2). Regression problems can be solved with GSFA using pre-defined graphs (e.g., a serial graph explained in Section 5.1). However, the structure of pre-defined graphs only takes into account the rank of the labels and not their exact value, a simplification that might decrease the estimation accuracy.

In this article, we focus on the analysis and design of training graphs. We develop a new approach, called exact label learning (ELL), for solving regression problems with GSFA based on the construction of a special training graph, in which the slowest feature extracted is already a label estimation, up to an affine transformation (Figure 1.c). The resulting system learns a nonlinear mapping from the input data (e.g., the pixels or features) to label estimations, where the features extracted by the layers of the GSFA network increase in complexity, abstraction level, and invariance as the data is propagated from the bottom to the top layer.

To develop the ELL method, we first study the slowest possible features that can be extracted by GSFA from a given graph when the feature space is unrestricted. Such features are called optimal free responses. After we express the optimal free responses of GSFA in a closed form, we develop a theoretical method for the converse operation: from a set of free

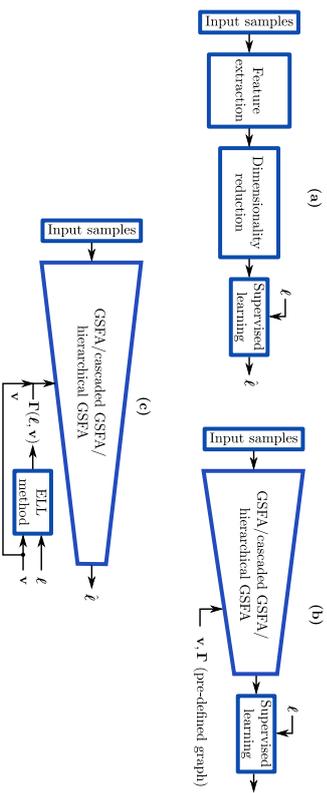


Figure 1: Three approaches for solving supervised learning problems. (a) A classic approach. (b) Previous approach using GSFA with a pre-defined training graph, which is defined by the input samples, node weights \mathbf{v} , and edge weights Γ . The samples are assumed to be ordered by increasing label. (c) Our proposal, which consists of a single GSFA architecture that is trained with a specially constructed graph $\Gamma(\ell, \mathbf{v})$. The first slow feature (with a global sign adjustment) directly provides the label estimation. If the label ℓ does not have weighted zero mean and weighted unit variance, a final affine transformation (scaling and offset) should be included.

responses we design the corresponding training graph. The method allows the creation of a graph in which the slowest possible feature is the label to be learned. Moreover, one can learn *multiple labels* simultaneously (e.g., object position, average color, shape, and size), and balance their importance by setting the value of certain parameters.

We show analytically that the serial graph is similar to the ELL graph in terms of the first optimal free responses, and that when only one label is learned the former may substitute the latter reasonably well with faster training time. In addition, we outline in the discussion a few extensions to the ELL method towards improving its efficiency and allowing the combination of training graphs.

The remainder of the article is organized as follows: In the next section, we describe the context of the ELL method and review previous work. In Section 3, we review GSFA. In Section 4, we propose the ELL method. In Section 5, we provide 3 applications: (1) We solve a regression problem on gender estimation from artificial images, validating the method. (2) We analyze efficient pre-defined training graphs for regression. (3) We use the ELL method in a non-conventional way to design a training graph for the extraction of compact features for classification, yielding improved performance when the number of features preserved is between $\log_2(C)$ and $C - 2$, where C is the number of classes. For applications (1) and (3) the accuracy is evaluated experimentally; the first one uses HGSA and the latter one uses direct (i.e., non-hierarchical) GSFA. Section 6 closes the article with a discussion.

2. Related Work

There are several approaches to solve regression problems on high-dimensional data and reduce the expensive computational requirements typically associated with this type of problems. A classic approach consists of feature extraction, (unsupervised) dimensionality reduction (DR), and an explicit supervised step (Figure 1.a). A different approach first uses GSFA for *supervised* DR. Supervised DR might result in higher accuracy than unsupervised DR. A small number of slow features extracted by GSFA are post-processed with a conventional classification or regression algorithm (Escalante-B. and Wiskott, 2013), see Figure 1.b. In such an approach, the supervised learning problem is mostly solved by GSFA implicitly, because it identifies and concentrates the label-predictive information in a few features.

For high-dimensional data, the direct application of SFA and GSFA is computationally too expensive, but one can resort to hierarchical processing (e.g., Franzini et al., 2011), which is an efficient divide-and-conquer strategy for the extraction of slow features¹ (e.g., Figure 2). For example, if the input dimension I is large, one can divide input data spatially into k lower-dimensional signals $\mathbf{x}^{(1)}(t), \dots, \mathbf{x}^{(k)}(t)$ of dimensionality $I' \stackrel{\text{def}}{=} I/k$. Then, one can extract slow features $\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(k)}$ from each lower-dimensional signal: $\mathbf{y}^{(1)}(t) \stackrel{\text{def}}{=} \text{SFA}^{(1)}(\mathbf{x}^{(1)}(t)), \mathbf{y}^{(2)}(t) \stackrel{\text{def}}{=} \text{SFA}^{(2)}(\mathbf{x}^{(2)}(t)), \dots, \mathbf{y}^{(k)}(t) \stackrel{\text{def}}{=} \text{SFA}^{(k)}(\mathbf{x}^{(k)}(t))$. A concrete instance of SFA trained with a particular subset of the training data is denoted as $\text{SFA}^{(\cdot)}$. Different SFA instances are also referred to as SFA nodes, especially in the context of hierarchical SFA networks structured as directed graphs. The nodes above are called local because their input is only a local subset of the original input. Each of them extracts J' slow features. Afterwards, another SFA node $\text{SFA}^{(\text{top})}$ in an additional layer extracts global slow features from the concatenation of the local slow features computed previously: $\mathbf{y}^{(\text{top})}(t) \stackrel{\text{def}}{=} \text{SFA}^{(\text{top})}(\mathbf{y}^{(1)}(t) \dots \mathbf{y}^{(k)}(t))$, where \cdot is the concatenation operation in space (not in time). A proper choice of J' and k ensures that the computation of $\mathbf{y}^{(\text{top})}(t)$ is feasible.

If the input dimensionality I' of the local nodes $\text{SFA}^{(1)}, \dots, \text{SFA}^{(k)}$ is still too large, one can repeat the strategy above to each of these nodes. Following such an approach recursively results in a multi-layer hierarchical network. Due to information loss before the top node and the change of the feature space, hierarchical SFA does not guarantee globally optimal slow features anymore. However, it has been shown to be effective in many practical experiments, in part because low-level features are spatially localized in most real-world data. Hierarchical GSFA (HGSA) offers an excellent computational complexity compared to direct GSFA that can be as good as linear w.r.t. the number of samples and the input dimensionality, depending on the network architecture (Escalante-B. and Wiskott, 2016).

The ELL method allows the creation of graphs useful to train direct, cascaded, and hierarchical GSFA. Cascaded GSFA refers to the application of several consecutive passes of GSFA (thus, it is equivalent to HGSA with only one GSFA node per layer) and is a useful approach when the features obtained through direct GSFA are not slow enough for a

1. Even linear SFA becomes infeasible if I is sufficiently large. Therefore, the usefulness of hierarchical processing is not limited to nonlinear SFA.

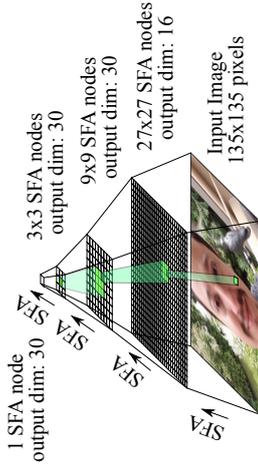


Figure 2: An example hierarchical SFA network for 2D data with 4 layers and no receptive field overlap that has been used for gender and age estimation from artificial face images (Escalante-B. and Wiskott, 2010). The SFA nodes can be easily replaced by GSFA nodes to construct an HG-SFA network. The input to one node in layers 1, 2 and 3 is highlighted.

particular application. The main advantage of cascaded over direct GSFA is that the feature space of the cascade may be more complex without making the individual nodes/layers/steps more complex. The theoretical part of this article concentrates on GSFA for simplicity, but we implicitly assume that it will be implemented in practice as HGSFA and applied to high-dimensional data.

There is a close relation between GSFA, generalized SFA (genSFA, Sprekeler, 2011; also see Rehn and Sprekeler, 2014) and locality preserving projections (LPP, He and Niyogi, 2003), sharing very similar objective functions and constraints, even though they originate from different backgrounds and were developed with different goals and applications in mind. Two differences are that in GSFA the vertex weights are independent of the edge weights and that GSFA is invariant to the scale of the weights, providing a normalized objective function $0 \leq \Delta \leq 4$ (for consistent graphs with non-negative edge weights). There are also differences in the similarity matrices used in practice for these algorithms. LPP originates from the field of manifold learning and its similarity matrices have been frequently computed using nearest neighbors of the input samples, reflecting input similarities. genSFA has mostly been used for classification and its similarity matrices are typically computed using input similarity information (nearest neighbors) with transitions restricted to samples of the same class. One can consider genSFA as LPP applied on the nonlinearly expanded data. GSFA originates from unsupervised learning and learning of invariances but is motivated by supervised learning, and training graphs for classification and regression have been used. Such graphs are computed using only the label information. The goal is to provide sensitivity to the label information and invariance to any other factor. Therefore, GSFA does not intend to preserve the manifold structure of the input data. It is possible to use GSFA to compute LPP features, and vice versa. The results of this article might thus also be relevant for researchers using LPP and genSFA.

Various efficient training graphs for classification (clustered graph) and regression (e.g., serial, mixed, sliding window graphs) have been proposed (Escalante-B. and Wiskott, 2013). These graphs have been pre-defined with efficiency in mind; although the number of edges contained in them is $\mathcal{O}(N^2)$, where N is the number of samples, their structure makes the training complexity linear w.r.t. N . This is possible due to algebraic simplifications in the training method that avoid the explicit representation of the graph as an $N \times N$ matrix.

Wiskott (2003) has studied the optimal free responses of SFA, i.e., the slowest possible features that can be extracted by SFA when there is no restriction regarding the training data or feature space. For SFA, he has computed the optimal free responses in continuous time by using variational calculus (also see Franzius et al., 2007). In this article, we use a different method for GSFA based on linear algebra to cope with the discrete nature of the index n that takes the place of time.

Due to the close connection between GSFA and LPP, the method proposed in Section 4.1 for computing the free responses of GSFA is closely connected to Laplacian eigenmaps (LE, Belkin and Niyogi, 2003). LE can be interpreted as a relaxation of LPP where the output features belong to an unrestricted feature space instead of being linear transformations of the inputs. Equivalently, LPP is a linearization of LE (Zhang et al., 2009).

Most regression methods have the shortcoming of a prohibitive computational cost if the data is high dimensional. Unsupervised DR can be useful to reduce the complexity, but the final accuracy may be suboptimal. Instead of direct regression, one could attempt to do hierarchical regression by training several regression nodes on low-dimensional data chunks and then combining their outputs on higher layers, similarly as HG-SFA is built to create a network of GSFA nodes. However, it is likely that the labels are not extractable at the lowest levels of the network (except with a noticeable error). Therefore, most information would be lost after the first layer, making the next layers unable to recover the labels accurately. GSFA extracts several slow features that do not need to be related to the labels in any simple way, allowing more label information to reach the top layers in HG-SFA, even if the labels cannot be extracted in the first layers.

The features extracted by GSFA nodes in an HG-SFA network have smaller receptive fields in the first layers that increase in size, as well as in complexity and selectivity, as one approaches the top of the network. This property is also present in other neural networks, such as the highly successful convolutional neural networks (CNNs). However, HG-SFA and CNNs differ considerably: the training method of HG-SFA is bottom-up, uses other types of nonlinearities, is not convolutional (although one can make some layers convolutional via weight sharing), and uses neither backpropagation nor max pooling. Moreover, in HG-SFA the features of each node fulfill a local optimality criterion (i.e., slowness).

3. Review of Graph-Based SFA (GSFA)

In this section, we recall the GSFA optimization problem and the GSFA algorithm. For a more detailed presentation of GSFA we refer to Escalante-B. and Wiskott (2013).

3.1 Training Graphs and the GSFA Problem

GSFA is trained with a so-called training graph, in which the vertices are the samples and the edges between two samples may represent or be related to the similarity of their labels.

In mathematical terms, the training data is represented as a training graph $G = (\mathbf{V}, \mathbf{E})$ (illustrated in Figure 3.a) with a set $\mathbf{V} = \{\mathbf{x}(n)\}_n$ of vertices, each vertex being a sample (i.e., a vector of length L), and a set \mathbf{E} of edges $(\mathbf{x}(n), \mathbf{x}(n'))$, which are pairs of samples, with $1 \leq n, n' \leq N$. The index n (or n') replaces the time variable t used by SFA. The edges are directed but typically have symmetric weights $\mathbf{\Gamma}^T = \mathbf{\Gamma} = \{\gamma_{n,n'}\}_{n',n}$; weights $v_n > 0$ are associated with the vertices $\mathbf{x}(n)$ and can be used to reflect their importance, frequency, or reliability. This representation includes the standard time series of SFA as a special case in which the graph has a linear structure (see Figure 3.b).

In order to solve classification problems with GSFA features, one should use training graphs that favor connections between samples from the same class by means of larger edge weights compared to those of different classes. When one is interested in regression problems, the training graphs should favor connections between samples with similar labels.

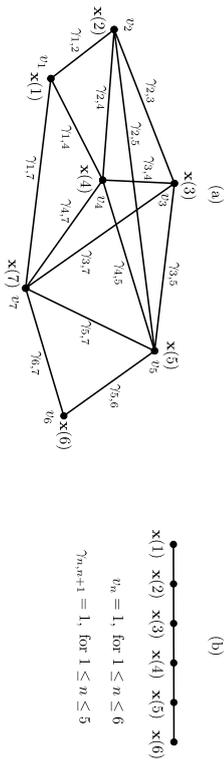


Figure 3: (a) Example of a training graph with $N = 7$ vertices. (b) A regular sample sequence (time series), which can be used to train SFA. This sequence is represented here as a linear graph that can be used with GSFA. If labels are available and the samples have been reordered by increasing/decreasing label (e.g., instead of having been ordered by time), the graph is called *sample reordering* graph. (Figure from Escalante-B. and Wiskott, 2013).

The concept of slowness has been originally defined for sequences of samples, but it has been generalized for GSFA to training graphs. The general goal of GSFA is to extract features that fulfill certain normalization restrictions and minimize the sum of the weighted squared output differences of all connected samples. More formally, the GSFA optimization problem (Escalante-B. and Wiskott, 2013) can be stated as follows: For $1 \leq j \leq J$, where J is the number of output features, find features $y_j(n) \stackrel{\text{def}}{=} g_j(\mathbf{x}(n))$, where $1 \leq n \leq N$, N is the number of samples, and g_j is a function belonging to a feature space \mathcal{F} (frequent choices for \mathcal{F} are all linear or quadratic transformations of the inputs), such that the objective function (weighted data value)

$$\Delta_j \stackrel{\text{def}}{=} \frac{1}{R} \sum_{n,n'} \gamma_{n,n'} (y_j(n') - y_j(n))^2 \text{ is minimal} \quad (1)$$

under the constraints

$$\frac{1}{Q} \sum_n v_n y_j(n) = 0, \quad (2)$$

$$\frac{1}{Q} \sum_n v_n (y_j(n))^2 = 1, \quad \text{and} \quad (3)$$

$$\frac{1}{Q} \sum_n v_n y_j(n) y_{j'}(n) = 0, \quad \text{for } j' < j, \quad (4)$$

$$\text{with } Q \stackrel{\text{def}}{=} \sum_n v_n \quad \text{and} \quad R \stackrel{\text{def}}{=} \sum_{n,n'} \gamma_{n,n'}. \quad (5)$$

The objective function penalizes the squared output differences between arbitrary pairs of samples using the edge weights as weighting factors. The feature $y_1(n)$, for $1 \leq n \leq N$, is the slowest one, $y_2(n)$ is the second slowest, and so on. Constraints (2)–(4) are called weighted zero mean, weighted unit variance, and weighted decorrelation, respectively. They are similar to the normalization constraints of SFA, except for the inclusion of vertex weights. The factors $1/R$ and $1/Q$ are not essential for the optimization problem, but they provide invariance to the scale of the edge weights as well as to the scale of the vertex weights, and serve a normalization purpose.

We write vectors and matrices in bold type. For instance, \mathbf{y}_j is the j -th feature vector of size N , $y_j(n)$ is the j -th feature of sample n , and $\mathbf{x}(n)$ is the n -th input sample of size L .

3.2 Linear GSFA Algorithm

The linear GSFA algorithm is similar to standard SFA (Wiskott and Sejnowski, 2002) and only differs in the computation of the matrices \mathbf{C} and $\hat{\mathbf{C}}$, which in GSFA takes into account the neighborhood structure specified by the training graph (samples, edges, and weights). The sample covariance matrix $\mathbf{C}_{\mathbf{G}}$ is defined as:

$$\mathbf{C}_{\mathbf{G}} \stackrel{\text{def}}{=} \frac{1}{Q} \sum_n v_n (\mathbf{x}(n) - \bar{\mathbf{x}})(\mathbf{x}(n) - \bar{\mathbf{x}})^T,$$

where $\mathbf{x}(n)$ and v_n denote an input sample and its weight, respectively, and

$$\bar{\mathbf{x}} \stackrel{\text{def}}{=} \frac{1}{Q} \sum_n v_n \mathbf{x}(n)$$

is the weighted average of all samples. The derivative second-moment matrix $\hat{\mathbf{C}}_{\mathbf{G}}$ is defined as:

$$\hat{\mathbf{C}}_{\mathbf{G}} \stackrel{\text{def}}{=} \frac{1}{R} \sum_{n,n'} \gamma_{n,n'} (\mathbf{x}(n') - \mathbf{x}(n))(\mathbf{x}(n') - \mathbf{x}(n))^T, \quad (6)$$

where edge weights $\gamma_{n,n'}$ are defined as 0 if the graph does not have an edge $(\mathbf{x}(n), \mathbf{x}(n'))$. Given these matrices, a sphering matrix \mathbf{S} and a rotation matrix \mathbf{R} are computed with

$$\mathbf{S}^T \mathbf{C}_{\mathbf{G}} \mathbf{S} = \mathbf{I}, \quad \text{and} \\ \mathbf{R}^T \mathbf{S}^T \hat{\mathbf{C}}_{\mathbf{G}} \mathbf{S} \mathbf{R} = \mathbf{A},$$

where \mathbf{A} is a diagonal matrix with diagonal elements $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_J$. Finally the algorithm returns $\Delta(y_1, \dots, \Delta(y_J), \mathbf{W}$ and $\mathbf{y}(n)$, where

$$\begin{aligned} \mathbf{W} &= \mathbf{S}\mathbf{R}, \text{ and} \\ \mathbf{y}(n) &= \mathbf{W}^T(\mathbf{x}(n) - \hat{\mathbf{x}}). \end{aligned} \quad (7)$$

It has been shown that the GSFA algorithm presented above indeed solves the optimization problem (1)–(4) in the linear function space. The proof is similar to the corresponding proof of standard linear SFA (Wiskott and Sejnowski, 2002).

The choice of the training graph $\mathbf{\Gamma}$ is important because it defines the types of features to be extracted. Figure 5 shows a serial graph useful for regression, whereas Figure 9 shows a clustered graph useful for classification.

3.3 Probabilistic interpretation of a graph

Interestingly, if the graph is connected and the following *consistency* restriction is fulfilled

$$\forall n : v_n/Q = \sum_{n'} \gamma_{n,n'}/R, \quad (8)$$

then GSFA yields the same features as standard SFA trained on a sequence generated by using the graph as a Markov chain with transition probabilities $\gamma_{n,n'}/R$ (see Klampff and Maass, 2010; Escalante-B. and Wiskott, 2013). Thus, one can use SFA to emulate GSFA. However, depending on the training graph chosen, emulating GSFA with SFA may be more expensive computationally.

3.4 GSFA Optimization Problem in Matrix Notation

In order to apply linear algebra methods to analyze GSFA, we use matrix notation. In what follows we assume that the edge weights are symmetric² ($\mathbf{\Gamma} = \mathbf{\Gamma}^T$) and that the consistency restriction (8) is fulfilled. This restriction can also be written as

$$\mathbf{v} \stackrel{(8)}{=} \frac{Q}{R} \mathbf{\Gamma} \mathbf{1}, \quad (9)$$

where $\mathbf{1}$ is a vector of ones of length N .

If \mathbf{y} is a feasible solution (i.e., satisfying (2) and (3)) and the graph fulfills the consistency restriction (8), the weighted delta value (1) can be simplified as follows,

$$\begin{aligned} \Delta \mathbf{y} &\stackrel{(1)}{=} \frac{1}{R} \sum_{n,n'} \gamma_{n,n'} (y(n') - y(n))^2 \\ &= \frac{1}{R} \left(\sum_{n'} (y(n'))^2 \sum_n \gamma_{n,n'} + \sum_n (y(n))^2 \sum_{n'} \gamma_{n,n'} - 2 \sum_{n,n'} \gamma_{n,n'} y(n') y(n) \right) \\ &\stackrel{(8)}{=} \frac{1}{R} \left(\sum_{n'} (y(n'))^2 \frac{R}{Q} v(n') + \sum_n (y(n))^2 \frac{R}{Q} v(n) - 2 \mathbf{y}^T \mathbf{\Gamma} \mathbf{y} \right) \\ &\stackrel{(3)}{=} 2 - \frac{2}{R} \mathbf{y}^T \mathbf{\Gamma} \mathbf{y}. \end{aligned} \quad (10)$$

2. An asymmetric edge-weight matrix $\mathbf{\Gamma}$ can be converted into a symmetric one $\mathbf{\Gamma}' \stackrel{\text{def}}{=} \frac{\mathbf{\Gamma} + \mathbf{\Gamma}^T}{2}$ without altering the solution to the optimization problem.

The optimization problem can then be stated as follows: For $1 \leq j \leq J$, find vectors \mathbf{y}_j of length N , with $y_j(n) \stackrel{\text{def}}{=} g_j(\mathbf{x}(n))$ and $g_j \in \mathcal{F}$, minimizing

$$\Delta_j \stackrel{(1,3,8)}{=} 2 - \frac{2}{R} \mathbf{y}_j^T \mathbf{\Gamma} \mathbf{y}_j \quad (11)$$

subject to:

$$\mathbf{v}^T \mathbf{y}_j \stackrel{(2)}{=} 0 \quad (12)$$

$$\mathbf{y}_j^T \text{Diag}(\mathbf{v}) \mathbf{y}_j \stackrel{(3)}{=} Q \quad (13)$$

$$\mathbf{y}_j^T \text{Diag}(\mathbf{v}) \mathbf{y}_{j'} \stackrel{(4)}{=} 0, \text{ for } j' < j, \quad (14)$$

where

$$Q \stackrel{(5,a)}{=} \mathbf{1}^T \mathbf{v}, \quad (15)$$

$$R \stackrel{(5,b)}{=} \mathbf{1}^T \mathbf{\Gamma} \mathbf{1}, \quad (16)$$

and $\text{Diag}(\mathbf{v})$ denotes a diagonal matrix with diagonal \mathbf{v} .

The use of matrix notation will facilitate the study of GSFA and the development of the ELL method in the next section.

4. Explicit Label Learning for Regression Problems

In this section, we propose the ELL method. First, we compute the optimal free responses of GSFA given any training graph. Then, we show how to construct a graph useful to learn any particular label or multiple labels. Afterwards, we show how to convert graphs with negative edge weights into graphs with non-negative weights only (such a method is useful to allow the probabilistic interpretation of the graph and to guarantee that the Δ values of all features lie between 0 and 4). Then, we motivate the use of auxiliary labels to improve learning. Finally, we analyze the computational complexity of the ELL method.

4.1 Optimal Free Responses of GSFA

In this section, we calculate the slowest possible solutions (optimal free responses) to the GSFA problem (11)–(14) that one could find if the feature space were unlimited. As we will see, the optimal free responses together with their corresponding Δ values, provide an alternative representation of the training graph and are a useful tool to understand its structure.

We use the Lagrange multiplier method to find critical points \mathbf{y} that are candidates for the optimal free responses. For the moment, we ignore the weighted decorrelation constraint (14) to solve for the first optimal free response, but we consider the remaining responses later. The method of Wiskott (2003) and Franzius et al. (2007) for computing optimal free responses of SFA relies on a continuous time variable t and cannot be applied to GSFA due to the discrete index n . Due to the close relationship between GSFA and LPP, the approach below is strongly related to Laplacian Eigenmaps (Belkin and Niyogi, 2003). Let

$$L \stackrel{\text{def}}{=} \left(2 - \frac{2}{R} \mathbf{y}^T \mathbf{\Gamma} \mathbf{y} \right) + \alpha \mathbf{v}^T \mathbf{y} + \beta \left(\mathbf{y}^T \text{Diag}(\mathbf{v}) \mathbf{y} - Q \right) \quad (17)$$

be a Lagrangian corresponding to the objective function (11), under the constraints (12) and (13). A signal \mathbf{y} is a critical point if the partial derivatives of L with respect to α, β , and $y_i(n)$, for $1 \leq n \leq N$, are simultaneously zero:

$$\partial L / \partial \alpha \stackrel{(17)}{=} \mathbf{v}^T \mathbf{y} \stackrel{!}{=} 0, \quad (18)$$

$$\partial L / \partial \beta \stackrel{(17)}{=} \mathbf{y}^T \text{Diag}(\mathbf{v}) \mathbf{y} - Q \stackrel{!}{=} 0, \quad \text{and} \quad (19)$$

$$\partial L / \partial \mathbf{y} \stackrel{(17)}{=} -\frac{4}{R} \mathbf{\Gamma} \mathbf{y} + \alpha \mathbf{v} + 2\beta \text{Diag}(\mathbf{v}) \mathbf{y} \stackrel{!}{=} \mathbf{0}, \quad (20)$$

where $\mathbf{0}$ is a vector of zeros.

Equations (18) and (19) merely require that the output \mathbf{y} has weighted zero mean and weighted unit variance, respectively. Multiplying (20) with $\mathbf{1}^T$ from the left and taking into account that $\mathbf{1}^T \text{Diag}(\mathbf{v}) = \mathbf{v}^T$, $\mathbf{1}^T \mathbf{v} \stackrel{(15)}{=} Q$, $\mathbf{1}^T \mathbf{\Gamma} \stackrel{(9)}{=} R \mathbf{v}^T$, and $Q > 0$ results in

$$-\frac{4}{R} \left(\frac{R}{Q} \mathbf{v}^T \right) \mathbf{y} + \alpha Q + 2\beta \mathbf{v}^T \mathbf{y} = 0,$$

implying $\alpha = 0$ due to (18). Therefore, (20) can be simplified to:

$$\left(-\frac{4}{R} \mathbf{\Gamma} + 2\beta \text{Diag}(\mathbf{v}) \right) \mathbf{y} = \mathbf{0},$$

$$\Leftrightarrow \text{Diag}(\mathbf{v}^{-1/2}) \left(\frac{4}{R} \mathbf{\Gamma} - 2\beta \text{Diag}(\mathbf{v}) \right) \text{Diag}(\mathbf{v}^{-1/2}) \text{Diag}(\mathbf{v}^{1/2}) \mathbf{y} = \mathbf{0},$$

$$\Leftrightarrow \left(\frac{4}{R} \text{Diag}(\mathbf{v}^{-1/2}) \mathbf{\Gamma} \text{Diag}(\mathbf{v}^{-1/2}) - 2\beta \mathbf{I} \right) \left(\text{Diag}(\mathbf{v}^{1/2}) \mathbf{y} \right) = \mathbf{0},$$

$$\Leftrightarrow \left(\text{Diag}(\mathbf{v}^{-1/2}) \mathbf{\Gamma} \text{Diag}(\mathbf{v}^{-1/2}) - \frac{R\beta}{2} \mathbf{I} \right) \left(\text{Diag}(\mathbf{v}^{1/2}) \mathbf{y} \right) = \mathbf{0}, \quad (21)$$

where $\mathbf{v}^{-1/2}$ is defined as the element-wise square root of the elements of \mathbf{v} , and $\mathbf{v}^{-1/2}$ is defined similarly (as usual, weights v_j are required to be strictly positive).

In a few words, \mathbf{y} is a critical point if it fulfills the weighted normalization constraints and the vector $\text{Diag}(\mathbf{v}^{1/2}) \mathbf{y}$ is an eigenvector of the matrix \mathbf{M} defined as

$$\mathbf{M} \stackrel{\text{def}}{=} \text{Diag}(\mathbf{v}^{-1/2}) \mathbf{\Gamma} \text{Diag}(\mathbf{v}^{-1/2}). \quad (22)$$

The corresponding eigenvalue is denoted

$$\lambda = \frac{R\beta}{2}. \quad (23)$$

We denote the (orthogonal) eigenvectors of matrix \mathbf{M} as \mathbf{u}_j with $\mathbf{u}_j^T \mathbf{u}_j = 1$. Each eigenvector \mathbf{u}_j gives rise to a critical point $\mathbf{y}_j \stackrel{\text{def}}{=} Q^{1/2} \text{Diag}(\mathbf{v}^{-1/2}) \mathbf{u}_j$, as long as also the weighted normalization constraints (12) and (13) are satisfied by \mathbf{y}_j . The slowest possible solution is the critical point \mathbf{y}_j with the smallest Δ -value. As we show below, the Δ -value of a critical point \mathbf{y}_j is directly related to the eigenvalue λ_j of the eigenvector $\mathbf{u}_j = Q^{-1/2} \text{Diag}(\mathbf{v}^{1/2}) \mathbf{y}_j$ of \mathbf{M} and can be computed as follows.

$$\Delta_{\mathbf{y}_j} \stackrel{(11)}{=} 2 - \frac{2}{R} (\mathbf{y}_j)^T \mathbf{\Gamma} \mathbf{y}_j$$

$$\stackrel{(22)}{=} 2 - \frac{2}{R} (\mathbf{y}_j)^T \text{Diag}(\mathbf{v}^{1/2}) \mathbf{M} \left(\text{Diag}(\mathbf{v}^{1/2}) \mathbf{y}_j \right)$$

$$\stackrel{(23)}{=} 2 - \frac{2}{R} (\mathbf{y}_j)^T \text{Diag}(\mathbf{v}^{1/2}) \lambda_j \text{Diag}(\mathbf{v}^{1/2}) \mathbf{y}_j$$

$$\stackrel{(13)}{=} 2 - \frac{2Q}{R} \lambda_j. \quad (24)$$

Thus, the slowest solution is the critical point \mathbf{y}_j with the largest eigenvalue λ_j . The remaining optimal free responses can now be addressed. They are given by the remaining critical points, where their corresponding eigenvalue defines their order, from largest to smallest. The weighted decorrelation condition (14) is fulfilled automatically due to the orthogonality of the eigenvectors: $\mathbf{u}_j^T \mathbf{u}_{j'} = 0 \Leftrightarrow \frac{1}{Q} \mathbf{y}_j^T \text{Diag}(\mathbf{v}) \mathbf{y}_{j'} = 0$ (follows from the definition of \mathbf{y}_j above).

One special case is when an eigenvalue has multiplicities. This means that two or more optimal free responses have the same Δ value, which is in fact the same Δ value of any rotation of such free responses. Therefore, optimal free responses with the same Δ value are not uniquely defined and any rotation of them is equivalent.

4.2 Design of a Training Graph for Learning One or Multiple Labels

Given a set of samples $\{\mathbf{x}(1), \dots, \mathbf{x}(N)\}$ with label $\ell = (\ell_1, \dots, \ell_N)$, we show how to generate a training graph, such that the slowest feature that could be extracted by GSFA is equal to a normalized version of the label. Notice that this problem (determining the structure of a training graph, or more concretely, its edge-weight matrix $\mathbf{\Gamma}$, having a particular optimal solution) differs considerably from the original GSFA problem of finding an optimal solution given a training graph and a feature space. The approach can be extended to multiple labels per sample. To distinguish them, we introduce an index $1 \leq j \leq L$, making ℓ_j denote the j -th label. The L labels can then be expressed as an affine transformation of the first L free responses, as described below.

Vertex-weights v_n indicate *a priori* likelihood information about the samples, and are thus assumed to be given and strictly positive. If this information is absent, one may set the vertex weights constant, e.g. $\mathbf{v} = \frac{1}{N} \mathbf{1}$.

Due to the normalization constraints, the outputs generated by GSFA must have weighted zero mean (12) and weighted unit variance (13). Therefore, to learn a single label ℓ we normalize it as follows: Let $\mu_\ell = \frac{1}{Q} \mathbf{v}^T \ell$ be the weighted label average and $\sigma_\ell^2 = \frac{1}{Q} (\ell - \mu_\ell \mathbf{1})^T \text{Diag}(\mathbf{v}) (\ell - \mu_\ell \mathbf{1})$ be the weighted label variance. Then, the normalized label is computed as

$$\tilde{\ell} = \frac{1}{\sigma_\ell} (\ell - \mu_\ell \mathbf{1}). \quad (25)$$

Hence, it is trivial to convert a normalized label into a non-normalized label and *vice versa*.

In order for the construction to work when samples have multiple labels, we must weight decorrelate them first. To decorrelate two labels $\ell_{j'}$ and ℓ_j , with $j' > j$, one can project ℓ_j out of $\ell_{j'}$: $\ell_{j'}^{\text{dec}}(n) = \ell_{j'}(n) - \frac{1}{Q} (\ell_j^T \text{Diag}(\mathbf{v}) \ell_j) \ell_j(n)$, which is an invertible linear operation.

From now on, we assume that the labels ℓ_1, \dots, ℓ_L have been decorrelated and normalized. We show how to compute edge weights $\gamma_{n,n'}$ such that the j -th optimal free response is equal to ℓ_j (with arbitrary polarity).

Define

$$\mathbf{\Gamma}^{\text{ELL}} \stackrel{\text{def}}{=} \text{Diag}(\mathbf{v}^{1/2}) \mathbf{M}^{\text{ELL}} \text{Diag}(\mathbf{v}^{1/2}), \quad (26)$$

where

$$\mathbf{M}^{\text{ELL}} \stackrel{\text{def}}{=} \sum_{j=0}^{N-1} \lambda_j \mathbf{u}_j^{\text{ELL}} (\mathbf{u}_j^{\text{ELL}})^T. \quad (27)$$

If $L < N - 1$ one can set $\lambda_{j>L} = 0$. The matrix $\mathbf{\Gamma}^{\text{ELL}}$ is symmetric by construction. The eigenvectors and eigenvalues of \mathbf{M}^{ELL} , which are explicit in its eigenvector decomposition (27), directly define the matrix $\mathbf{\Gamma}^{\text{ELL}}$, and determine the optimal free responses of the resulting graph. Concretely, for each $j \geq 1$ one sets $\mathbf{u}_j^{\text{ELL}}$ according to the desired label ℓ_j (ignore $\mathbf{u}_0^{\text{ELL}}$ and λ_0 for the time being).

$$\mathbf{u}_j^{\text{ELL}} = Q^{-1/2} \text{Diag}(\mathbf{v}^{1/2}) \ell_j, \text{ for } j \geq 1 \quad (28)$$

Notice that the weighted decorrelation of the labels translates directly into the orthogonality of the corresponding eigenvectors, that is

$$\frac{1}{Q} (\ell_j)^T \text{Diag}(\mathbf{v}) \ell_{j'} \stackrel{(14)}{=} 0 \quad \Leftrightarrow \quad (\mathbf{u}_j^{\text{ELL}})^T \mathbf{u}_{j'}^{\text{ELL}} = 0 \quad (29)$$

Once the eigenvectors are computed we must decide which eigenvalues we want to give them. Alternatively, we can decide which Δ values we give to the labels, because $\Delta \ell_j$ and λ_j are directly related: $\lambda_j \stackrel{(24)}{=} \frac{R}{2Q} (2 - \Delta \ell_j)$.

Larger eigenvalues (equivalent to smaller Δ values) might result in higher accuracy for the corresponding label. We give some intuition on how to choose the eigenvalues of the eigenvectors. a) In general, important labels should have larger eigenvalues than less important ones. b) The global scale of the eigenvalues $\lambda_{j>0}$ is irrelevant, only their relative scales matter. For convenience one can scale them so that $\sum \lambda_{j>0} = 1$. c) If two labels are similarly important, their eigenvalues should be also similar.

For example, if one only wants to learn a single label ℓ_1 with a delta value $\Delta \ell_1 = 0$, one can set $\mathbf{u}_1^{\text{ELL}} = Q^{-1/2} \text{Diag}(\mathbf{v}^{1/2}) \ell_1$, $\lambda_1 = 1$, and the eigenvalues $\lambda_{j>1}$ to zero. If ℓ_1 takes only two possible values (e.g., -1 and 1), the resulting graph will be disconnected and contain two clusters. Otherwise, the resulting graph will be connected, and the condition $\Delta \ell_1 = 0$ necessarily implies that some of the resulting edge weights will be negative, a condition that we deal with in Section 4.3.

The analysis of Section 4.1, which is used by the ELL method requires that the graph fulfills the consistency restriction (9). We set the remaining eigenvector

$$\mathbf{u}_0^{\text{ELL}} = Q^{-1/2} \mathbf{v}^{1/2}, \quad (30)$$

with eigenvalue $\lambda_0 = R/Q$. This ensures that $(\mathbf{u}_0^{\text{ELL}})^T \mathbf{u}_0^{\text{ELL}} = 1$ and (9) is fulfilled, as follows.

$$\begin{aligned} \mathbf{\Gamma}^{\text{ELL}} \mathbf{1} &\stackrel{(26,27)}{=} \text{Diag}(\mathbf{v}^{1/2}) \left(\sum \lambda_j \mathbf{u}_j^{\text{ELL}} (\mathbf{u}_j^{\text{ELL}})^T \right) \text{Diag}(\mathbf{v}^{1/2}) \mathbf{1} \\ &\stackrel{(30)}{=} \text{Diag}(\mathbf{v}^{1/2}) \left(\sum \lambda_j \mathbf{u}_j^{\text{ELL}} (\mathbf{u}_j^{\text{ELL}})^T \right) \mathbf{u}_0^{\text{ELL}} Q^{1/2} \\ &\stackrel{(30)}{=} \text{Diag}(\mathbf{v}^{1/2}) \lambda_0 \mathbf{u}_0^{\text{ELL}} Q^{1/2} \\ &= (R/Q) \mathbf{v}. \end{aligned} \quad (31)$$

The assignment of $\mathbf{u}_0^{\text{ELL}}$ and λ_0 above also ensures that $\mathbf{1}^T \mathbf{\Gamma}^{\text{ELL}} \mathbf{1} \stackrel{(15,31)}{=} R$. The free pseudo-response $\ell_0 \stackrel{(28)}{=} \mathbf{1}$ corresponding to $\mathbf{u}_0^{\text{ELL}}$ fulfills equations (13) and (14) but not (12). Therefore, ℓ_0 is not a feasible solution, but it has similar properties to the optimal free responses. The introduction of $\mathbf{u}_0^{\text{ELL}}$ does not reduce the generality of the labels $\ell_{j>0}$ that can be learned; orthogonality between $\mathbf{u}_0^{\text{ELL}}$ and $\mathbf{u}_{j>0}^{\text{ELL}}$ is equivalent to (12), i.e., the weighted zero mean of $\ell_{j>0}$, a condition that is required anyway for any feasible solution: $(\mathbf{u}_0^{\text{ELL}})^T \mathbf{u}_{j>0}^{\text{ELL}} = 0 \stackrel{(28)}{\Leftrightarrow} (Q^{-1/2} \mathbf{v}^{1/2})^T Q^{-1/2} \text{Diag}(\mathbf{v}^{1/2}) \ell_{j>0} = Q^{-1} \mathbf{v}^T \ell_{j>0} = 0$.

Although only L free responses are explicitly defined, $N - L - 1$ additional optimal free responses are defined implicitly with an eigenvalue of 0, corresponding to $\Delta = 2.0$. This Δ value has a particular meaning, because as we prove in the next paragraph, it is the Δ value of unit-variance zero-mean i.i.d. noise for certain graphs.

4.2.1 EXPECTED WEIGHTED Δ VALUE OF A NOISE FEATURE

Let \mathbf{y} be a noise feature randomly sampled from a zero-mean unit-variance distribution \mathcal{D} , i.e., $y(n) \leftarrow \mathcal{D}(0, 1)$. On average, \mathbf{y} fulfills the weighted normalization constraints (12) and (13), as can be seen as follows.

$$\langle \mathbf{v}^T \mathbf{y} \rangle_{\mathcal{D}} = \mathbf{v}^T \langle \mathbf{y} \rangle_{\mathcal{D}} = 0, \quad (32)$$

$$\langle \mathbf{y}^T \text{Diag}(\mathbf{v}) \mathbf{y} \rangle_{\mathcal{D}} = \langle \sum_n v_n y(n)^2 \rangle_{\mathcal{D}} = \sum_n v_n \langle y(n)^2 \rangle_{\mathcal{D}} = Q, \quad (33)$$

where $\langle \cdot \rangle_{\mathcal{D}}$ denotes expected value when sampling over \mathcal{D} . The expected delta value can be computed as

$$\begin{aligned} \langle \Delta \rangle_{\mathcal{D}} &\stackrel{(1)}{=} \frac{1}{R} \sum_{n,n'} \gamma_{n,n'} \langle (y(n') - y(n))^2 \rangle_{\mathcal{D}} \\ &= \frac{1}{R} \left(\sum_{n,n',n \neq n'} \gamma_{n,n'} \langle (y(n') - y(n))^2 \rangle_{\mathcal{D}} + \sum_n \gamma_{n,n} \langle (y(n) - y(n))^2 \rangle_{\mathcal{D}} \right) \\ &= \frac{1}{R} \left(\sum_{n,n',n \neq n'} \gamma_{n,n'} \langle (y(n')^2)_{\mathcal{D}} + (y(n)^2)_{\mathcal{D}} - 2 \langle y(n') y(n) \rangle_{\mathcal{D}} \rangle_{\mathcal{D}} + 0 \right) \\ &= \frac{1}{R} \sum_{n,n',n \neq n'} \gamma_{n,n'} (1 + 1 - 0) \\ &= \frac{2}{R} \left(\sum_{n,n',n \neq n'} \gamma_{n,n'} - \sum_n \gamma_{n,n} \right) \stackrel{(5)}{=} \frac{2(R - \sum_n \gamma_{n,n})}{R}. \end{aligned} \quad (34)$$

Therefore, if the graph has no self-loops (i.e., $\forall n : \gamma_{n,n} = 0$), the expected Δ value $(\Delta_y)_p$ of a noise feature \mathbf{y} is 2.0. The self-loops of a graph (e.g., one constructed using the ELL method) can be removed (i.e., their weight be set to zero). This does not change the free responses, only the scale of the Δ values is modified due to the change in R . The consistency restriction might be broken, though.

4.3 Elimination of Negative Edge Weights

From the objective function (1), it is obvious that a positive edge weight connecting two samples expresses that those samples should be mapped close to each other in feature space. In contrast, a negative edge weight expresses that two samples should be mapped as far apart as possible, thus encoding output dissimilarities. Nevertheless, the weighted unit variance constraint still applies, so the solutions are not unbounded.

If the edge weights are non-negative, the smallest possible Δ value is $\Delta = 0$. However, if negative edge weights are allowed, some feasible features might have $\Delta < 0$. A feature with $\Delta < 0$ would appear to be “slower” than the infeasible constant feature $\mathbf{y} = \mathbf{1}$ with $\Delta = 0$, contradicting the intuitive interpretation of slowness. Moreover, negative edge weights hinder the probabilistic interpretation of the graph (see Section 3.2), because some of the transition probabilities $\gamma_{n,n}/R$ of the resulting Markov chain would be negative.

Training graphs constructed using the ELL method might include negative edge weights, which would result in the disadvantages described above. Therefore, in this section, we add an additional step to the ELL method to ensure that the training graph has non-negative edge weights. More concretely, we show how to transform a training graph with strictly positive vertex weights v_n and arbitrary edge weights $\mathbf{\Gamma}$ (positive and negative) into a graph with the same vertex weights and only non-negative edge weights $\mathbf{\Gamma}'$. The optimization problem defined by $\mathbf{\Gamma}'$ is equivalent to the original optimization problem in terms of its solutions and their order. Only the value of the objective function is linearly changed (or, more precisely, changed by an affine function).

Assume that $\forall n : v_n > 0$, and that there is at least one element $\gamma_{n,n'} < 0$. Let

$$c \stackrel{\text{def}}{=} \max_{n,n'} \frac{-\gamma_{n,n'}}{v_n v_{n'}}. \quad (35)$$

The new edge weights $\mathbf{\Gamma}'$ are defined as

$$\mathbf{\Gamma}' \stackrel{\text{def}}{=} \frac{1}{1 + cQ^2/R} (\mathbf{\Gamma} + c\mathbf{v}\mathbf{v}^T). \quad (36)$$

Now, we show the properties of $\mathbf{\Gamma}'$ compared to those of $\mathbf{\Gamma}$:

1. All elements of $\mathbf{\Gamma}'$ are greater or equal to zero, as desired. (Follows from (35), which implies $\gamma_{n,n'} + cv_n v_{n'} \geq 0$.)
2. Symmetry is preserved by (36). Clearly $\mathbf{\Gamma}'$ is symmetric if and only if $\mathbf{\Gamma}$ is symmetric.
3. The sum of edge-weights is preserved:

$$R' \stackrel{(16)}{=} \mathbf{1}^T \mathbf{\Gamma}' \mathbf{1} \stackrel{(36)}{=} \frac{R + cQ^2}{1 + cQ^2/R} = R. \quad (37)$$

4. Fulfillment of the graph consistency restriction (9) is preserved:

$$\begin{aligned} \mathbf{1}^T \mathbf{\Gamma} \stackrel{(9)}{=} R/Q\mathbf{v}^T &\Rightarrow \mathbf{1}^T \mathbf{\Gamma}' \stackrel{(36)}{=} \frac{1}{1 + cQ^2/R} (\mathbf{1}^T \mathbf{\Gamma} + c\mathbf{1}^T \mathbf{v}\mathbf{v}^T) \\ &\stackrel{(9)}{=} \frac{R/Q}{1 + cQ^2/R} (R/Q\mathbf{v}^T + cQ\mathbf{v}^T) \\ &= \frac{R/Q}{1 + cQ^2/R} (1 + cQ^2/R)\mathbf{v}^T \\ &= R/Q\mathbf{v}^T. \end{aligned}$$

5. $\mathbf{\Gamma}$ and $\mathbf{\Gamma}'$ define equivalent optimization problems. Let \mathbf{y} be a feasible solution. The constraints of the optimization problem are independent of $\mathbf{\Gamma}'$, and only the objective function is modified as follows:

$$\begin{aligned} \Delta_y &\stackrel{(10)}{=} 2 - \frac{2}{R} \mathbf{y}^T \mathbf{\Gamma}' \mathbf{y} \\ &\stackrel{(36,37)}{=} 2 - \frac{2}{R(1 + cQ^2/R)} (\mathbf{y}^T \mathbf{\Gamma} \mathbf{y} + c\mathbf{y}^T \mathbf{v}\mathbf{v}^T \mathbf{y}) \\ &\stackrel{(12)}{=} 2 - \frac{2}{R(1 + cQ^2/R)} \mathbf{y}^T \mathbf{\Gamma} \mathbf{y} \\ &\stackrel{(10)}{=} 2 - \frac{R}{R(1 + cQ^2/R)} \frac{2}{2} (2 - \Delta_y) \\ &= \frac{1}{(1 + cQ^2/R)} \left(\Delta_y + \frac{2cQ^2}{R} \right). \end{aligned} \quad (38) \quad (39)$$

Therefore, the objective function is only modified by a positive scaling factor and a constant positive offset, proving that the optimal free solutions to the training graph remain stable, as well as their order.

6. In particular, a feature \mathbf{y} with $\Delta_y = 2$ preserves its delta value, i.e. $\Delta_y = 2 \stackrel{(38)}{\Leftrightarrow} \Delta_y' = 2$.

4.4 Auxiliary Labels for Boosting Estimation Accuracy

It is possible to provide additional *auxiliary* labels derived from the original one ℓ_1 to improve the estimation accuracy when GSFA is applied repeatedly (e.g., cascaded or in a convergent hierarchical GSFA network). Consider two GSFA nodes, one stacked on top of the other. If the first GSFA node is not be able to extract ℓ_1 accurately, it might still be capable of approximating labels $\ell_k = f_k(\ell_1)$, for $2 \leq k \leq K$, where the functions $f_k(\cdot)$ are nonlinear. Since these features are derived from the original label ℓ_1 , they contain a certain amount of information about it. In this case, the output features are likely to contain linear combinations of the labels ℓ_1, \dots, ℓ_k providing a redundant coding of ℓ_1 . These features are likely to be easier to disentangle by the second node to better approximate the original label ℓ_1 . Therefore, we suggest to explicitly promote the appearance of these features by learning also auxiliary labels.

The functions f_k can be defined arbitrarily; we suggest to use

$$\ell_k(n) = \cos \left(\frac{\ell_1(n) - \min(\ell_1)}{\max(\ell_1) - \min(\ell_1)} \pi k \right), \quad \text{for } 2 \leq k \leq K, \quad (40)$$

where $\max(\ell_1)$ is the largest label value, and $\min(\ell_1)$ is the smallest one. As usual, we assume the labels ℓ_1 to ℓ_K are weight decorrelated and normalized before the ELL method is applied.

The eigenvalues corresponding to the auxiliary labels must be set smaller than those of the original label. Otherwise, the slowest features might be more similar to the auxiliary labels than to the original one. From now on, the term *target* labels will be used to refer to the original and auxiliary labels, if present.

The use of auxiliary samples can be justified from information theory. Assume that the samples have been ordered by increasing label ℓ_1 . This implies that for ℓ_k the argument of the cosine function ranges from 0 to $k\pi$. Thus ℓ_2 describes 1 oscillation, ℓ_3 describes 1.5 oscillations, etc. In this sense, the auxiliary labels are “higher-frequency” versions of ℓ_1 . Notice that ℓ_2 contains almost all the information about ℓ_1 except for 1 bit. That is, $I(\ell_1, \ell_2) = H(\ell_1) - 1$, where I is mutual information and H is entropy. Similarly, ℓ_4 loses 2 bits of information about ℓ_1 , ℓ_8 loses 3 bits, and so on. Thus, auxiliary labels contain a large amount of information about ℓ_1 .

Moreover, the use of auxiliary labels supports the goal that samples $\mathbf{x}(n)$ and $\mathbf{x}(n')$ with similar labels $\ell_1(n)$ and $\ell_1(n')$ should have similar output features $y_j(n)$ and $y_j(n')$ on average, for $1 \leq j \leq J$, and not only the slowest features $y_1(n)$ and $y_1(n')$. This is a result of the “smoothness” of the auxiliary labels in terms of ℓ_1 (i.e., how fast they change w.r.t. ℓ_1). Notice that $\ell_1, \ell_2, \dots, \ell_J$ would be ordered by decreasing smoothness.

Interestingly, in regular SFA (or GSFA trained with the reordering graph) the inclusion of auxiliary labels occurs automatically. The slowest free response is a half period of a cosine function, and the subsequent free responses are the higher-frequency harmonics of the first one (see Section 5.2, particularly Figure 7).

4.5 Computational Complexity of Explicit Label Learning

The main drawback of ELL is its computational efficiency compared to efficient pre-defined training graphs, which is more marked for large N . We analyze the efficiency of explicit label learning by considering its two main parts: The construction of the training graph and training GSFA with it.

The graph construction requires $\mathcal{O}(L^2N + LN^2)$ operations. The term L^2N is due to the transformation of L target labels into eigenvectors, which might require a decorrelation step on L N -dimensional vectors. The term LN^2 is due to the computation of \mathbf{M} , which involves L vector multiplications $\mathbf{u}_j \mathbf{u}_j^T$.

When GSFA is trained, three computations are particularly expensive. (1) The computation of \mathbf{C}_G , which takes $\mathcal{O}(NI^2)$ operations. (2) The computation of $\hat{\mathbf{C}}_G$, which can be expressed as $\hat{\mathbf{C}}_G = \frac{2}{\sigma} \mathbf{X} \text{Diag}(\mathbf{v}) \mathbf{X}^T - \frac{2}{\sigma} \mathbf{X} \mathbf{I} \mathbf{X}^T$, where $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$, taking $\mathcal{O}(N^2I + NI^2)$ operations. (3) The solution to the generalized eigenvalue problem, which requires $\mathcal{O}(I^3)$ operations. Therefore, in general, training GSFA requires $\mathcal{O}(N^2 + N^2I + I^3)$ operations. Typically $N > I$ to avoid overfitting, so the computation of $\hat{\mathbf{C}}_G$ is the most expensive part.

However, when an efficient pre-defined graph (e.g., the serial graph) is used instead of an ELL graph, it is possible to avoid the explicit graph construction and compute \mathbf{C}_G with optimized algorithms that take into account the regular structure of the graph. In this way,

efficient pre-defined graphs allow the computation of $\hat{\mathbf{C}}_G$ in $\mathcal{O}(NI^2)$ operations, which is equivalent to the complexity of standard SFA on N I -dimensional samples. Moreover, if the number of edges $N_e \leq N(N+1)/2$ is small, one can use (6) to compute $\hat{\mathbf{C}}_G$ in $\mathcal{O}(N_e I^2)$ operations. Therefore, for these two special cases, training GSFA takes $\mathcal{O}(NI^2 + I^3)$ and $\mathcal{O}(N_e + N)I^2 + I^3$ operations, respectively. In Section 6.4 we further discuss the complexity of the ELL method and in Section 6.5 we propose a few approaches to improve it.

5. Applications of Explicit Label Learning

This section we present three applications of the proposed method. The first one illustrates how to solve a regression problem with GSFA explicitly, learning a direct mapping from images to labels (see Figure 1.c). The second application shows the analysis of two pre-defined graphs by computing their optimal free responses. In the third application, the ELL method is used in a new way to learn compact discriminative labels for classification.

5.1 Explicit Estimation of Gender with GSFA

We consider the problem of gender estimation from artificial face images, which is treated here as a regression problem, because the gender parameter is defined as a real value by the face modeling software (FaceGen SDK, Singular Inversions Inc., 2008).

Input data. The input data are 12,000 64×64 grayscale images. Each image is generated using a new subject identity, where the gender is explicitly specified, and the rest of the parameters of the faces (e.g., age, racial composition) are random. The average pixel intensity of each image is normalized by multiplying the pixel values by an appropriate factor to eliminate skin color as a cue for gender estimation. The resulting images show subjects with a fixed pose, no hair or accessories, and the illumination is fixed, as well as the average pixel intensity and the background color (black). See Figure 4 for some sample images. To specify the gender parameter, 60 different values are used $(-3, -2.9, \dots, 2.9)$.

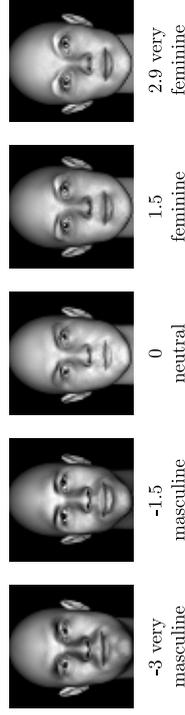


Figure 4: Example of the normalized images used, showing different values of the gender parameter.

The images are randomly split into a training and a test set. The training set consists of 10,800 images, 180 images for each gender value, whereas the test set consists of 1,200 images, 20 images for each gender value.

Besides the gender label, also a second “color” label is considered, which is the average pixel intensity of the image *before* normalization. Due to normalization, this label cannot be

the final clipping step LR was clearly more accurate than affine mapping (experiment not shown), but both methods have similar accuracy if clipping is enabled. For all graphs, the explicitly supervised soft GC method provided better accuracy than the affine mapping, although the difference is smaller than one might have expected.

L	Graph ELL^{g-c-L}			Graph ELL^{g-c-L}		
	scaling (1F)	LR (1F)	soft GC (3F)	scaling (1F)	LR (1F)	soft GC (3F)
1	0.376	0.380	0.364	0.365	0.298	0.289
10	0.364	0.365	0.353	0.356	0.349	0.350
20	0.372	0.374	0.356	0.357	0.423	0.426
30	0.367	0.368	0.350	0.349	0.473	0.478
40	0.368	0.367	0.346	0.345	0.508	0.514
50	0.376	0.375	0.351	0.350	0.535	0.543

Table 2: *Gender* estimation errors (RMSE) using various graphs and either one (1F) or three (3F) features. For the linear regression (LR) mapping, the label is estimated as $\hat{\ell}_1 = ay_1 + b$, with a and b fitted to the training data. Chance level (RMSE) is 1.731 if one uses the constant estimation $\hat{\ell}_1 = -0.05$. All errors computed on test data and averaged over 10 runs. (Left) Estimation errors using training graphs for gender estimation only. (Right) Estimation errors using training graphs for the experiment on simultaneous estimation of gender and color.

For comparison, the serial graph results in RMSEs of 0.351 (soft GC, 1F) and 0.349 (soft GC, 3F), whereas the reordering graph results in RMSEs of 0.353 (soft GC, 1F) and 0.347 (soft GC, 3F). The accuracy of these two graphs appears to be similar; however, in more complex experiments the serial graph has typically been more accurate. The ELL^{g-c-L} graph is, therefore, slightly more accurate than the serial and reordering graphs but 25 times slower, taking about 250 min for training instead of about 10 min (single thread).

Simultaneous learning of gender and color. We construct a graph that encodes gender and color simultaneously, learning labels ℓ_1, \dots, ℓ_L , where ℓ_1 is the gender label, ℓ_2 is the color label, $\ell_3, \dots, \ell_{L-1}$ are derived from ℓ_1 , and $\ell_4, \ell_6, \dots, \ell_L$ are derived from ℓ_2 . Each set of labels is computed using (40) similarly to the auxiliary labels for gender only but starting from either the original gender or color labels. The chosen eigenvalues decrease linearly and add to one. The resulting graphs are denoted ELL^{g-c-L} , where L is the total number of target labels, with $L = 2 \times d$, for $d \in \{1, 5, 10, 15, 20, 25\}$, and $2(d-1)$ is the number of auxiliary labels used for gender and color.

The effect of coding gender and color simultaneously on gender estimation is shown in Table 2, right (compare to Table 2, left). The ELL^{g-c-L} graphs yield significantly higher accuracy than the ELL^{g-L} graphs (an MAE as small as 0.277 vs. 0.345). The results on color estimation using the ELL^{g-c-L} graphs are shown in Table 3, right (compare to Table 3, left). The slowest extracted feature represents mostly gender. However, it must also contain color information since it allows color estimation better than chance level. When 3 features are preserved, the ELL^{g-c-L} graphs yield higher accuracy than the ELL^c-L graphs. Similar

L	Graph ELL^c-L			Graph ELL^{g-c-L}		
	scaling (1F)	LR (1F)	soft GC (3F)	LR (1F)	soft GC (1F)	soft GC (3F)
1	2.000	1.987	1.971	1.979	2 × 1	4.247
10	1.969	1.958	1.905	1.922	2 × 5	3.606
20	2.006	1.999	1.914	1.922	2 × 10	3.214
30	1.991	1.989	1.877	1.889	2 × 15	2.978
40	1.990	1.990	1.864	1.867	2 × 20	2.828
50	1.997	1.997	1.865	1.871	2 × 25	2.718

Table 3: *Color* estimation errors (RMSE) using various graphs and either one (1F) or three (3F) features. Chance level (RMSE) is 7.447. All results computed on test data and averaged over 10 runs. (Left) Error using training graphs that encode only color. (Right) Error using training graphs that simultaneously encode gender and color.

experimental results have been reported, e.g. by Guo and Mu (2014), who have shown that age estimation improves when gender and race labels are also considered.

Learning label transformations. We verify that the method can learn other labels that are implicitly described by the data. More precisely, we use GSFA to learn labels $(\ell_1)^2$ and $(\ell_1)^3$, which are distorted versions of the original gender label ℓ_1 . The graphs constructed for this purpose are denoted $\text{ELL}^{g-40(\ell_1)^2}$ and $\text{ELL}^{g-40(\ell_1)^3}$, respectively. Both of them include 39 auxiliary labels besides the main distorted label. To better approximate the target labels, more complex nonlinearities are used in some of the nodes of the hierarchical networks. The $(\ell_1)^2$ network is identical to the ℓ_1 network, except that in the top node the quadratic expansion is used instead of the 0.8Expo expansion. Similarly, the $(\ell_1)^3$ network uses the quadratic expansion in the 7th layer, and the 6th-degree polynomial expansion in the top node. In both networks, the output dimension of the node in the 7th layer is set to 3 to avoid overfitting due to the expansion in the 8th layer.

The corresponding label estimations are shown in Figure 6. For comparison, also the ELL^{g-40} graph is included. The results prove that the ELL method can also be used to learn distortions of the main label. Admittedly, the accuracy of the estimations (expressed as a fraction of the respective chance levels) decreases even though we increased the complexity of the feature space.

5.2 Analysis of Pre-Defined Training Graphs

In this section, we use the method of Section 4.1 to extract the optimal free responses of three graphs (reordering, serial and ELL^c-L). The optimal free responses and their Δ values (alternatively, the eigenvectors \mathbf{u}_j and eigenvalues λ_j) fully characterize the properties of a training graph, and provide another representation of it that might be more useful in some contexts.

We compute optimal free responses using (21)–(23) and their delta values using (24). Therefore, these results have been obtained analytically. We plot them in Figure 7, which shows an arbitrary label to be learned (top), and three different graphs that can be used

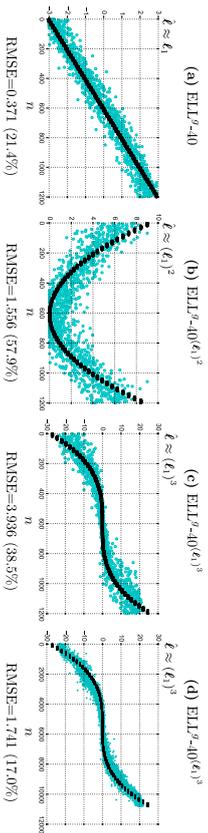


Figure 6: Plots (a) to (c) show the label estimations on test data (a single run) when different distorted versions of ℓ_1 are learned. The affine mapping is used. Therefore, the estimations are only generated from the slowest feature. Ground-truth values are shown in thicker black. The RMSE is expressed in parenthesis as a percentage of the chance level. Plot (d) is analogous to (c) but shows *training* data.

for this purpose. Only $N = 30$ samples (ordered by increasing label) are used to ease visualization, but the plots behave similarly for larger N . The following three graphs are employed. 1) A reordering graph (Figure 3.b) that has been extended with two edge weights $\gamma_{0,0} = 1$ and $\gamma_{N-1,N-1} = 1$ to fulfill the consistency restriction (8), which is required by the method. These weights introduce a constant scaling $N/(N+2)$ of the delta values, without any further consequence. 2) A serial graph (Section 5.1) with $K = 15$ groups of 2 samples each. 3) An ELL-4 graph (Sections 4.2-4.4) that is constructed with the original labels $\ell_1(n) = \ell(n)$, and 3 auxiliary labels computed using (40).

Figure 7 shows also that the most remarkable difference between these graphs is the number of optimal free responses with $\Delta < 2.0$, which is 14 for the reordering graph, 6 for the serial graph, and 4 for the ELL-4 graph, for the parameters above. For arbitrary parameters, the reordering, serial and ELL- L graphs have $\lfloor (N-1)/2 \rfloor$, $\lfloor (K-1)/2 \rfloor$, and, depending on the eigenvalues, up to $L \leq N-1$ optimal free responses with $\Delta < 2.0$, respectively.

Although the graphs differ considerably in their connectivity, their first four to five optimal free responses have a somewhat similar shape. Since in all graphs the slowest free response \mathbf{y}_1 is increasing, a monotonic mapping would be enough to approximate the label for any of these graphs. However, the slowest response of the serial graph is constant within each group, which might lower accuracy due to a discretization error. In contrast, the ELL-4 graph has been tailored to learn a particular label, and therefore \mathbf{y}_1 is exactly ℓ_1 (the original label) except for an offset and scaling.

The analysis makes clear that the serial and ELL-4 graphs are *more selective* than the reordering graph regarding the features that they consider slow. To illustrate why this might be an advantage, consider a scaled and noisy version \mathbf{y}_1 of ℓ_1 . More concretely, $\hat{y}_1(n) = \frac{\sqrt{2}}{2}\ell_1(n) + \frac{\sqrt{2}}{2}e(n)$, where $e(n)$ is an i.i.d. zero-mean unit-variance noise signal. When the reordering graph is used, the feature \mathbf{y}_1 has an average Δ -value of about 1 (i.e. $\langle \Delta_{\mathbf{y}_1} \rangle \approx 1$), and therefore such a feature would appear to be faster than an auxiliary (40) feature $\mathbf{y}_6 = \ell_6$, because $\Delta_{\ell_6} \approx 0.38$. Hence, a GSFA node trained with the reordering

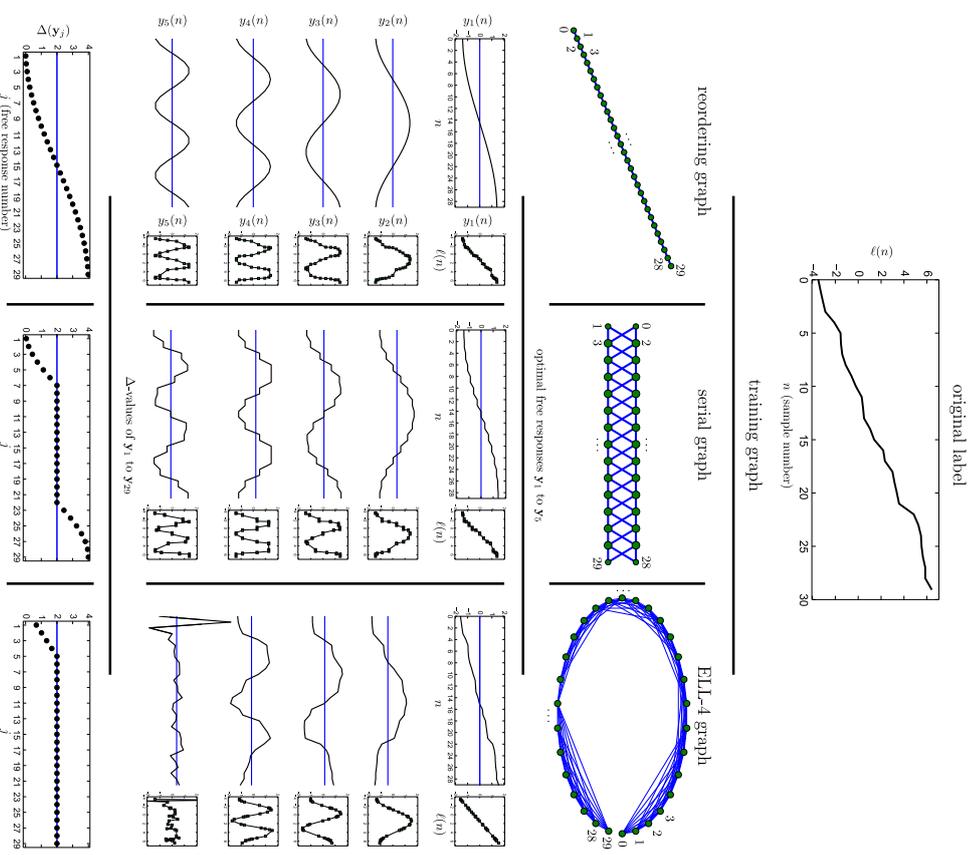


Figure 7: An arbitrary label $\ell(n)$ (top) and three graphs that can be used to learn it. The five slowest optimal free responses \mathbf{y}_1 to \mathbf{y}_5 of each graph are plotted, as well as the delta values of all optimal free responses. The ELL-4 graph is almost fully connected, but here only the strongest 30% of the connections are displayed. Samples have an index n from 0 to 29, and free responses have an index j from 1 to 29. The free responses are also plotted against the original label (smaller square plots). The polarity of the free responses was adjusted once to make them negative for the first sample.

graph would favor the extraction of \mathbf{y}_6 over $\hat{\mathbf{y}}_1$, even though $\hat{\mathbf{y}}_1$ is more similar to the label. In contrast, the serial and ELL-4 graphs might favor the extraction of $\hat{\mathbf{y}}_1$, because for these graphs $\Delta_{\mathcal{L}_6}$ is larger and close to 2.0.

5.3 Compact Discriminative Features for Classification

A well-known algorithm for supervised dimensionality reduction for classification is Fisher discriminant analysis (FDA). According to the theory of FDA, if there are C classes, $C - 1$ features define a $C - 1$ dimensional subspace that best separates the classes. In practice, one typically uses all these $C - 1$ features, because all of them contain discriminative information and contribute to classification accuracy. The same holds for GSFA if the clustered training graph is used (GSFA+clustered), because in this case the features learned are equivalent to those of FDA (see Klampfl and Maass, 2010; Escalante-B. and Wiskott, 2013).

One can take advantage of hierarchical processing to do classification using the clustered graph (HGSFA+clustered). However, when the number of classes C is large (e.g. $C \geq 100$) it might become expensive to preserve $C - 1$ features in each node, because the size of the input to subsequent nodes would be a multiple of $C - 1$. Such a large dimensionality would be further increased by the expansion function, resulting in a large training complexity. For instance, consider a 2-layer nonlinear network for classification with two GSFA nodes in the first layer and one in the top layer. Suppose the first two nodes have output dimensionality $C - 1 = 99$, making the input of the top node 198-dimensional, and suppose that the top node applies a quadratic expansion to its input data before linear GSFA. The expanded data would have dimensionality $I' = 19,701$. The combination of a large sample dimensionality I' and a large number of samples N (with $N \gg I'$ to avoid overfitting) would result in considerable computational and memory costs. Therefore, if we could encode the class information in the first layer more compactly, we could reduce the output dimensionality of the first-layer nodes and reduce overfitting, aiming at increasing classification accuracy.

In this section, we use the theory of explicit learning of multiple labels to compute compact features for classification using GSFA. We classify images of $C = 32$ traffic signs from the German traffic sign recognition benchmark database (Houben et al., 2013).

The images are represented as 48×48 -pixel color (RGB) images (see Figure 8). We use only 32 out of 43 traffic signs with the most samples, so that the number of classes is a power of 2 and the number of samples is maximized. For the training data, we use the same number of samples for each class (traffic sign), namely 2,160 of them, making a total of 69,120 images. To reach 2,160 samples per class, images of some classes are used up to 6 times (since the database is unbalanced). The images used for training are distorted by a random rotation r of $-3.15 \leq r \leq 3.15$ degrees, horizontal and vertical translations Δ_x, Δ_y with $-1.73 \leq \Delta_x, \Delta_y \leq 1.73$ pixels, and a scaling factor s with $0.91 \leq s \leq 1.09$. The purpose of these distortions is to improve generalization and provide invariances to small misalignments. We use the official test data, which ensures that the test images originate from signs physically different from the ones used for training. The test data consists of 9,030 undistorted images.

We used a simple (non-hierarchical) GSFA architecture, in which PCA is applied first to reduce the dimensionality to 120 principal components. Afterwards, quadratic GSFA is



Figure 8: The 32 traffic signs learned, one image per class.

applied using different training graphs, described below. Finally, since this is a classification problem, a nearest centroid classifier is used instead of the affine mapping.

The ELL method is used to construct two training graphs with binary target labels (i.e., label values are either 1 or -1). The first one has 5 labels (compact+5) and the second one has 31 (compact+31). The target labels are defined in Table 4. Notice that the first 5 labels (for both graphs) suffice, in principle, to fully encode the class information, because they can be viewed as a binary representation of the class number.

For the compact+5 graph, identical eigenvalues $(\lambda_1^1 = \lambda_2^1 = \lambda_3^1 = \lambda_4^1 = \lambda_5^1 = 0.2)$ are used to express equal importance of the target labels. The compact+31 graph has been included to show the effect of auxiliary labels $\ell_6, \ell_7, \dots, \ell_{31}$. For this graph, the first five eigenvalues $(\lambda_1^2, \lambda_2^2, \dots, \lambda_5^2) = (0.056, 0.056, \dots, 0.056)$ are identical, but the rest decrease linearly: $(\lambda_6^2, \lambda_7^2, \dots, \lambda_{31}^2) = (0.053, 0.051, \dots, 0.004, 0.002)$, where only three decimal places are shown. Thus, the importance given to the auxiliary labels decreases from ℓ_6 to ℓ_{31} . For both graphs, we scale the eigenvalues to make their sum equal to 1.

We choose $C = 2^5$ classes, because powers of two make it simple to obtain binary labels with a weighted zero mean, weighted unit variance, and weighted decorrelation, as follows. The first five original labels can be computed as $\ell_j(c) = 2 \lfloor \frac{c-1}{2^j} \rfloor \bmod 2 - 1$, where $1 \leq c \leq C$ is the class number, the division is integer division and “mod” is the modulo operation (i.e., an image n of class c is assigned a label $\ell_j(c)$). The auxiliary labels are computed as the product of two or more labels ℓ_1 to ℓ_5 , possibly multiplied by a factor -1 to make the label assigned to the first class negative. More concretely, ℓ_6 is the product of all original labels, ℓ_7 to ℓ_{11} are all products of four of them, ℓ_{12} to ℓ_{21} are all products of three, and ℓ_{22} to ℓ_{31} are all products of two (e.g., $\ell_6 = \ell_1 \ell_2 \ell_3 \ell_4 \ell_5$, $\ell_7 = -\ell_1 \ell_2 \ell_3 \ell_4$, $\ell_8 = -\ell_1 \ell_2 \ell_3 \ell_5$, $\ell_{30} = -\ell_3 \ell_5$, $\ell_{31} = -\ell_4 \ell_5$).

For both graphs, we set $\mathbf{v} = \mathbf{1}$. The corresponding eigenvectors are $\mathbf{u}_j \stackrel{(28)}{=} Q^{-1/2} \ell_j$, where $Q \stackrel{(9)}{=} N \cdot \mathbf{1} = 69,120$ (N is the number of training images). These eigenvectors are also binary and allow for a fast computation of the covariance matrix in $\mathcal{O}(LN I^2 + I^3)$ operations, where L is the number of target labels.

$c \rightarrow$	1	2	3	4	5	6	7	8	9	...	16	17	...	30	31	32
$\ell_1(c)$	-1	-1	-1	-1	-1	-1	-1	-1	-1	...	-1	-1	...	+1	+1	+1
$\ell_2(c)$	-1	-1	-1	-1	-1	-1	-1	-1	-1	...	-1	-1	...	+1	+1	+1
$\ell_3(c)$	-1	-1	-1	-1	-1	-1	-1	-1	-1	...	-1	-1	...	+1	+1	+1
$\ell_4(c)$	-1	-1	-1	-1	-1	-1	-1	-1	-1	...	-1	-1	...	+1	+1	+1
$\ell_5(c)$	-1	-1	-1	-1	-1	-1	-1	-1	-1	...	-1	-1	...	+1	+1	+1
$\ell_6(c)$	-1	+1	+1	+1	+1	+1	+1	+1	+1	...	+1	+1	...	-1	-1	-1
$\ell_7(c)$	-1	-1	+1	+1	+1	+1	+1	+1	+1	...	+1	+1	...	-1	-1	-1
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
$\ell_{30}(c)$	-1	+1	+1	+1	+1	+1	+1	+1	+1	...	+1	+1	...	-1	-1	-1
$\ell_{31}(c)$	-1	+1	+1	+1	+1	+1	+1	+1	+1	...	-1	-1	...	+1	+1	+1

Table 4: Target labels used to encode the class information, compactly expressed as a function of the class number c . The compact+5 graph is constructed with labels ℓ_1 to ℓ_5 , whereas the compact+31 graph with ℓ_1 to ℓ_{31} . The first five labels can be seen as the original ones and the rest as auxiliary.

Clustered training graph. For comparison purposes, we also consider the clustered graph (Escalante-B. and Wiskott, 2013), an efficient pre-defined graph that generates features useful for classification, see Figure 9. The optimization problem associated with this graph explicitly demands that samples from the same class should typically be mapped to similar outputs.

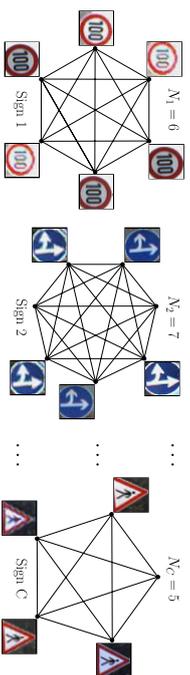


Figure 9: Illustration of a clustered training graph for classification with C classes (traffic signs). Each vertex represents a sample, and edges represent transitions. The N_c samples belonging to a class $c \in \{1, \dots, C\}$ are connected, constituting a fully connected subgraph. Samples of different classes are not connected. Vertex weights are identical and equal to one, whereas edge weights depend on the cluster size as $\gamma_{n,n'} = 1/(N_c - 1)$, where $\mathbf{x}(n)$ and $\mathbf{x}(n')$ belong to class c and $n \neq n'$. For traffic sign recognition, we use $C = 32$ signs and $N_c = 2, 160$ images per sign.

The features learned by GSFA on this graph are equivalent to those learned by Fisher discriminant analysis (FDA, see Klampff and Mass 2010 and also compare Berkes 2005a and Berkes 2005b). This type of problem can be analyzed theoretically when the function space of SFA is unrestricted. Consistent with FDA, the first $C - 1$ slow features extracted (optimal

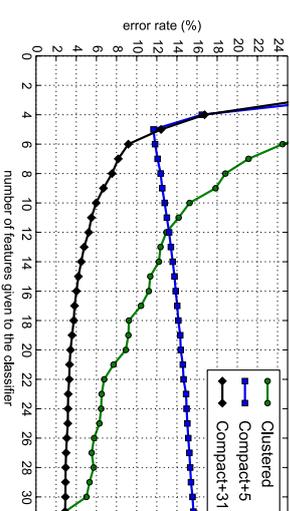


Figure 10: Classification error when GSFA is trained with the compact+5, compact+31 and clustered graph (FDA). This error is a function of the graph and the number of slow features d passed to the classifier. For the clustered graph, dropping even a single feature might increase the error rate significantly. For instance, the error rate of using 30 features computed with the clustered graph is worse than the error rate of 13 features computed with the compact+31 graph. Performance on 9,030 test samples, averaged over 10 runs. For $d \geq 4$ the standard error or the mean is at most 0.38%.

free responses) are orthogonal step functions, and are piece-wise constant for samples from the same class (Berkes, 2005a).

The classification error is plotted in Figure 10, where the number of slow features d given to a nearest centroid classifier ranges from 4 to 31. For comparison, the clustered graph is also evaluated. For $d = 5$ features, the compact+5 graph results in the best accuracy with an error rate of 11.67%, against 12.42% (compact+31) and 29.74% (clustered). However, the error rate of the compact+5 graph increases if one preserves more than 5 features, indicating that additional features contain little or no discriminative information. For $6 \leq d \leq 30$, the compact+31 graph yields clearly better accuracy than the other graphs. Interestingly, for $d = 31 = C - 1$ features, the compact+31 and clustered graph give identical error rates of 2.89%, which is their top performance. In this case, the features extracted are *different* but contain the *same information* since they can be mapped to each other linearly. In other words, the first 31 free responses of both graphs describe the same subspace. Any *single* optimal free response from the compact+31 graph contains 1 bit of discriminative information (which might be redundant to the others). In contrast, the first features extracted by the clustered graph might sacrifice discriminative information to minimize within-class variance (e.g., a feature $\mathbf{Y}(c) = ((\frac{C}{2})^{1/2}, -(\frac{C}{2})^{1/2}, 0, \dots, 0)$ has minimal (zero) within-class variance but provides little discriminative information (less than 1 bit if $C \geq 9$), because most of the time the feature takes the value 0 and otherwise only the first two classes can be identified from it). Using $d > 31$ features does not improve accuracy in any case. For comparison, the highest performance obtained for this database during the official competition is a 0.54% error rate for all 43 signs by Cirisan et al. (2012).

The method of compact discriminative classes provides more accurate label estimations if the feature space is complex enough to allow the extraction of features that approximate the binary labels. If the feature space is poor, the compact graphs might not bring any advantage over the clustered one.

6. Discussion

In this article, we propose exact label learning (ELL) for the construction of training graphs. When GSFA is trained with an ELL graph, the final label estimation is just an affine transformation of the slowest extracted feature. Thus, the method allows the direct solution of regression problems with GSFA, without having to resort to a supervised post-processing step. In other words, given a new input sample (e.g., an input image) the first feature computed using an ELL graph directly provides an approximation of the label (or an affine transformation of it). In practice, even better results may be achieved using more than one feature and supervised post-processing.

Supervised learning problems on high-dimensional data are of great practical importance, but they frequently result in systems with large computational demands. A common approach is to apply feature extraction, dimensionality reduction, and a supervised learning algorithm. A promising alternative approach is hierarchical GSFA (HGSFA), because its complexity scales in some cases even linearly w.r.t. the input dimensionality and the number of samples. In this context, it is especially useful to train HGSFA with an ELL graph since the resulting architecture is simple and homogeneous, as shown in Figure 1.c.

We have proposed a method to compute the optimal free responses of a training graph analytically. This method allows us to understand the type of features that can be extracted from a training graph independently of the feature space. Moreover, it has allowed us to propose the ELL method, where the labels are explicitly considered to create the training graph. In the resulting ELL graph, the optimal free responses are equal to a normalized version of the labels, and if the feature space is complex enough, HGSFA will learn features that approximate (or span) the original labels.

Graphs with negative edge weights would result in negative transition probabilities, violating the probabilistic interpretation of the graph, and might yield features with negative Δ value, contradicting the notion of slowness. Therefore, we also show how to transform a graph to make the edge weights non-negative without altering the extracted features.

We have proved the usefulness of the ELL method by showing three types of applications that are relevant in practice: ELL regression with multiple labels, analysis of training graphs, and classification with compact discriminative features.

It is crucial to emphasize that GSFA optimizes feature slowness, which depends on the particular training graph used, and not label estimation accuracy. However, when the ELL method is used, the training graphs define a slowness objective that requires optimizing an output similarity function where the similarities are intimately related to the desired label similarities. As a consequence, the feature slowness objective and estimation accuracy objective become equivalent when the feature space \mathcal{F} is unlimited. That is, the slowest possible features that can be extracted (i.e. optimal free responses) are equal to a normalized version of the label(s). In practice, \mathcal{F} is finite to allow generalization from training to test data and, if the features extracted are slow enough (i.e. close to the optimal free responses),

they are also good solutions to the original regression problem. If the slowest feature extracted is not sufficiently similar to the label, one can enhance the mapping from features to labels by mapping more than one output feature, and one can boost feature slowness by including auxiliary labels in the graph construction, as explained in Section 6.1.

We would like to underline that the ELL method is not equivalent to linear regression from the data to the (weight-decorrelated and normalized) target labels ℓ_1, \dots, ℓ_L . Any feasible feature vector $\tilde{\mathbf{y}}$ can be decomposed in terms of the optimal free responses $\mathbf{y}_1, \dots, \mathbf{y}_{N-1}$ as $\tilde{\mathbf{y}} = \sum_{j=1}^{N-1} \alpha_j \mathbf{y}_j$, with $\alpha^T \boldsymbol{\alpha} = 1$ to ensure weighted unit variance. The ELL method ensures that the first L optimal free responses $\mathbf{y}_1, \dots, \mathbf{y}_L$ are equal to the target labels ℓ_1, \dots, ℓ_L and have Δ values $\Delta_1, \dots, \Delta_L$. The remaining free responses are defined implicitly and have $\Delta_{L < j < N} = 2$. The Δ value of $\tilde{\mathbf{y}}$ can be expressed as $\Delta_{\tilde{\mathbf{y}}} = \sum_{j=1}^{N-1} (\alpha_j)^2 \Delta_j$. Let $\tilde{\mathbf{y}}_1, \dots, \tilde{\mathbf{y}}_J$ be concrete output features of GSFA for particular data using an ELL graph. We remark that the features $\tilde{\mathbf{y}}_1, \dots, \tilde{\mathbf{y}}_J$ are ordered by slowness, and $\tilde{\mathbf{y}}_j$ does not necessarily approximate \mathbf{y}_j . In particular, $\tilde{\mathbf{y}}_1$ is the slowest possible feature in the feature space, and it may be a linear combination of the free responses that is uncorrelated with $\mathbf{y}_1 = \ell_1$ if \mathbf{y}_1 cannot be approximated in the feature space (although this extreme case is less likely). In contrast, if one used linear regression, each one of the target labels would be approximated separately (i.e., $\tilde{\mathbf{y}}_j$ would approximate \mathbf{y}_j) regardless of the quality of the approximation. This would be mostly disadvantageous when used hierarchically. For instance, if a node in a network has output dimensionality $J < L$ (this scenario is frequent in the lower layers of the network), it is preferable to preserve the J slowest extractable features than the (eventually poor) linear approximations of ℓ_1, \dots, ℓ_J .

6.1 Multiple and Auxiliary Labels

ELL allows learning multiple labels simultaneously, for instance to encode different aspects of the input at once (e.g. object color, size, shape, orientation). The use of multiple labels has been inspired by biological systems, where complementary information channels have been observed and appear to improve feature robustness, for example, under incomplete information (Krüger et al., 2013). Learning gender and color simultaneously yielded clearly smaller estimation errors than when these labels were estimated separately (Section 5.1). This shows that multiple label learning is not only theoretically possible, but that coding complementary information channels might boost accuracy in practice. For instance, an automatic system for face image processing might benefit from the simultaneous extraction of the subject's identity, age, gender, race, pose, and expression.

One application of multiple labels is learning auxiliary labels derived from the original one (e.g. "higher-frequency" transformations of it). The results show that encoding auxiliary labels improves accuracy (Section 5.1). Such a technique is particularly relevant for cascaded or convergent hierarchical GSFA networks, where the outputs of some GSFA nodes feed other nodes. The use of auxiliary labels has been justified based on the fact that these labels contain substantial information about the original label (Section 4.4). For instance, as explained before, the first auxiliary label ℓ_2 only lacks one bit of information about the original label ℓ_1 . Therefore, even if ℓ_1 does not belong to the feature space of a node, the auxiliary labels might be (approximately) extracted, preserving information about ℓ_1 . A GSFA node (or any supervised learning algorithm) higher in the hierarchy may then be

able to approximate ℓ_1 more accurately by making use of the information carried by the auxiliary labels. Additionally, auxiliary labels have also been justified by a smoothness heuristic, where samples n and n' having similar labels $\ell_1(n)$ and $\ell_1(n')$ should have similar output features $y_j(n)$ and $y_j(n')$, for $1 \leq j \leq J$. Without auxiliary labels only the first output feature would have this property, and the remaining features might vary quickly w.r.t. the original label.

6.2 Application of the ELL Method

The experiments on gender (and skin-color) estimation from artificial face images demonstrate that the ELL method also works in practice when used hierarchically.

The experiments of Section 5.1 and the analytical results of Section 5.2 show the strength of the serial graph when only a single label is available. In this case, the ELL graph provided marginally better estimations than the serial graph (an RMSE of 0.345 with the ELL $_{\mathcal{G},40}$ graph vs. 0.349 with the serial graph, in both cases using 3 features, the soft GC post-processing method, and averaging over 10 runs), but the computation time was 25 times longer. We verified that the difference is statistically significant.

Although the shape of the slowest feature extracted with the serial graph may be less similar to the label, a monotonic transformation of the slowest feature learned by a nonlinear supervised step (e.g. soft GC) may suffice to approximate it.

However, the results suggest that if two or more (intrinsically connected) labels are available, the accuracy of using ELL graphs further increases. Efficient pre-defined graphs are not available in this case. In the gender estimation experiment, the RMSE was improved to 0.277 by jointly learning gender and skin (ELL $_{\mathcal{G},\mathcal{C}}(2 \times 5)$ graph, 3 features, soft GC). This is much better than the serial graph. Hence, a particularly promising application for the ELL method is multiple label learning.

Various methods for mapping the slowest feature to a label were tested. The affine mapping method is interesting from a theoretical point of view. However, as one would expect, the soft GC method, which is nonlinear and supervised, provides better accuracy on test data. Therefore, the latter might be preferred in practical applications. Moreover, in this scenario, supervised post-processing methods might be computationally inexpensive, because their input is frequently low-dimensional (e.g., we only used 1 to 3 slow features for gender estimation).

6.3 Classification with ELL

Although ELL was originally designed for regression, we have shown that it can also be useful for classification when particular labels are learned. The experiment on traffic sign classification shows the benefit of using compact discriminative features, implemented here by learning multiple binary labels. The resulting system has a much smaller classification error than the clustered graph (equivalent to nonlinear FDA) when the number of output dimensions is less than $C-1$, where C is the number of classes. The compactness of the feature set can be useful to do classification with many classes. This is particularly beneficial for hierarchical GSFA because less features have to be propagated by the network, which might also reduce overfitting. Although ideally $\log_2(C)$ binary target labels suffice for

perfect classification, the experiments show that additional target labels via auxiliary labels improve classification accuracy in practice.

Interestingly, the clustered graph for C classes (equivalent to FDA) and the compact+ $(C-1)$ graph are equivalent if the latter is constructed with constant positive eigenvalues $\lambda_1 = \dots = \lambda_{C-1} = 1/(C-1)$. The reason for this equivalence is that this compact+ $(C-1)$ graph would only have within-class transitions, because transitions between different classes cancel out each other. Therefore, the clustered graph can be seen as a special case of the compact+ $(C-1)$ graph, with maximum label redundancy ($C-1$ target labels) and giving equal importance (eigenvalues) to all of them.

For simplicity we used binary target labels, but it is also possible to use C -valued labels. For instance, the first label can be the class number, and additional labels can be random permutations of this assignment (label decorrelation and normalization still apply). Ideally, these labels might result in an even more compact representation, because a single optimal free response encodes the class information.

Contrary to many approaches for classification based on LPP, the goal of the ELL method is strictly focused on learning the label information while being invariant to any other aspect of the data. Since we do not intend to learn the input manifold, we do not use nearest neighbors to compute the training graph. However, as shown by the (regression) experiments on simultaneous gender and color estimation, learning specific additional labels can also be useful to better disentangle the discriminative information.

6.4 Efficiency of ELL

The complexity of training a single GSFA node with an ELL graph is $\mathcal{O}(IN^2 + I^2N + I^3)$ operations, where I is the input dimensionality (possibly after a nonlinear expansion), and $N > I$ is the number of samples. For comparison, the serial graph has a complexity of $\mathcal{O}(I^2N + I^3)$. Thus, the main limitation of using ELL graphs is the training complexity when N is large. However, this might not be a big disadvantage for the following reasons: (1) The complexity of the ELL method is comparable to the complexity of LPP. Similarity matrices in LPP are typically computed using nearest neighbors. In practice, the complexity of computing these matrices is similar to $\mathcal{O}(IN^2)$ (He, 2005), and the remaining steps of LPP have complexity $\mathcal{O}(I^2N + I^3)$ if the number of edges is linear w.r.t. N .

(2) The experiment on the estimation of gender shows that it is feasible to apply the ELL method to 10,800 64×64 images in 250 min (*single thread*, Intel Core i7-870 2.93GHz, 16 GByte RAM). This might be fast enough for some real-life applications.

(3) The ELL method is of theoretical interest in any case, allowing the analysis of training graphs and providing insights for the design of better hand-crafted graphs.

In case better efficiency is still necessary, we outline a few extensions to the ELL method in Section 6.5, two of them trading accuracy for speed.

6.5 Extensions of ELL

We have devised the following possible extensions (which may be combined):

(1) **Graph trimming.** One might compute sparse approximations of the ELL graphs with significantly less than $\mathcal{O}(N^2)$ edges. For example, one might delete a fraction of the edges having the smallest weights or a random selection of all the edges.

(2) **Sample grouping.** Another method first groups the input samples according to their labels, resulting in K groups of N/K samples each. The ELL method is then applied to the average labels of the groups to compute a reduced graph with K vertices and $\mathcal{O}(K^2)$ edges. If the largest number of labels L is used, i.e. $L = I$, the reduced graph can be constructed in $\mathcal{O}(IK^2 + I^2K)$ operations. Afterwards, one can derive a specialized method to train GSFA using the reduced graph. Such a method considers the transitions between all pairs of samples of two connected groups, in the same way as the serial graph. This avoids the explicit computation of the full edge-weight matrix of size $N \times N$. The training complexity would then be $\mathcal{O}(I^2N + I^2K^2 + I^3)$ using $\mathcal{O}(I^2K + NI)$ memory. An interesting value for K is $K = \sqrt{N}$, which divides the training data in \sqrt{N} groups of \sqrt{N} samples each, resulting in $\mathcal{O}(I^2N + I^3)$ operations. The term I^2K in the memory complexity might be large, but one can sacrifice some performance to reduce memory usage, resulting in $\mathcal{O}(I^2N + I^2KN + I^3)$ operations and $\mathcal{O}(I^2 + NI)$ memory.

(3) **Combination of graphs.** Under some conditions, we show how to combine training graphs meaningfully. Consider two training graphs that fulfill the consistency restriction (9) and share the same vertices (samples) $\mathbf{x}(n)$ and vertex weights $v(n)$. Let $\mathbf{\Gamma}_1$ and $\mathbf{\Gamma}_2$ be the corresponding edge weight matrices, and $0 < \alpha < 1$ be a weighting factor. The combined graph has the same vertices and node weights, but a combined edge weight matrix $\mathbf{\Gamma}_c \stackrel{\text{def}}{=} \alpha\mathbf{\Gamma}_1 + (1 - \alpha)\mathbf{\Gamma}_2$. Assume that $\mathbf{1}^T\mathbf{\Gamma}_1\mathbf{1} = \mathbf{1}^T\mathbf{\Gamma}_2\mathbf{1} = R$ (otherwise the edge weights of one graph can be scaled). Since the vertices and vertex weights are identical, a feature \mathbf{y} that is feasible for one of the graphs is also feasible for the other two. Let $\Delta_{\mathbf{y}}^{\mathbf{\Gamma}_1}$ and $\Delta_{\mathbf{y}}^{\mathbf{\Gamma}_2}$ be the Δ value of \mathbf{y} for the original graphs. Then, $\Delta_{\mathbf{y}}^{\mathbf{\Gamma}_c} \stackrel{(10)}{=} \alpha\Delta_{\mathbf{y}}^{\mathbf{\Gamma}_1} + (1 - \alpha)\Delta_{\mathbf{y}}^{\mathbf{\Gamma}_2}$. This implies that if a feature \mathbf{y} has a Δ -value smaller than an arbitrary constant β (i.e., $\Delta_{\mathbf{y}}^{\mathbf{\Gamma}_1} < \beta$) and it is not larger than β in the second one (i.e., $\Delta_{\mathbf{y}}^{\mathbf{\Gamma}_2} \leq \beta$), it can be warranted that it will be also smaller than β in the combined graph (i.e., $\Delta_{\mathbf{y}}^{\mathbf{\Gamma}_c} < \beta$) for any $0 < \alpha < 1$. This property may be useful to create graphs with optimal free responses that span various labels.

The third extension can be used to combine ELL and/or pre-defined graphs without distinction. In an upcoming work, Escalante-B. and Wiskott (2016) combine three efficient pre-defined graphs for face image analysis: two clustered graphs for classification of race and gender, and a serial graph for the estimation of age. The accuracy for age estimation on the MORPH-II database using the combined graph (and an improved version of HGSFA) is a mean average error (MAE) of 3.50 years, which is more accurate than the current state-of-the-art systems for this database (Yi et al., 2015 with an MAE of 3.63 years and Guo and Mu, 2014 with an MAE of 3.92).

6.6 Future Work

In future work, we would like to explore the extensions above. For example, it seems reasonable to combine the serial and the reordering graph. The first one has a large number of edges, which provides good generalization, whereas the second one does not incur in the quantization error of the serial graph caused by its grouping of samples. Thus, the combination might improve accuracy.

Although the ELL method supports multiple labels, it might be less effective if the number of target labels L is large. However, in this case the labels are frequently categorical

and sparse. Therefore, we would like to investigate methods that concentrate the label information (e.g., by computing a compressed representation of the labels) to reduce the number of effective labels.

We have proposed a method to set the auxiliary labels, and have explained why it is meaningful to use them. However, one could choose them according to some optimization criterion, e.g., to explicitly maximize estimation accuracy. Also the assignment of the eigenvalues could be optimized. Apparently it is difficult to determine optimal auxiliary labels and their eigenvalues analytically. For classification with $C = 32$ classes, *linearly* decreasing eigenvalues (for the auxiliary labels) provided great results, but other eigenvalues might be better if C is very large.

In the ELL method, several eigenvalues were set to zero and the corresponding eigenvectors remained unspecified. As suggested by a reviewer, one could use these eigenvectors and eigenvalues to construct graphs with special structural constraints, such as a minimum and maximum number of edges per vertex.

6.7 Conclusion

Hierarchical processing and the slowness principle are two powerful brain-inspired learning principles. The strength of SFA originates from its theoretical foundations in the field of learning of invariances and the generality of the slowness principle. For practical supervised learning applications, HGSFA provides good accuracy and efficiency and still profits from strong theoretical foundations. An advantage of relying on such general principles is that the resulting algorithms are application independent and not confined to a particular problem or input feature representation. Of course, fine tuning the network parameters and the integration of problem-specific knowledge are always possible for additional performance. The proposed ELL method explores the limits of HGSFA and is valuable as a theoretical tool for the analysis and design of training graphs. However, the results show that with certain adaptations (e.g., the use of supervised post-processing) it is also sufficiently robust to be applied to practical computer vision and machine learning tasks.

References

- M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6):1373–1396, June 2003.
- P. Berkes. Pattern recognition with Slow Feature Analysis. Cognitive Sciences EPrint Archive (CogPrints), February 2005a. URL <http://cogprints.org/4104/>.
- P. Berkes. Handwritten digit recognition with nonlinear Fisher discriminant analysis. In *ICANN*, volume 3697 of *LNCS*, pages 285–287. Springer Berlin/Heidelberg, 2005b.
- D. Citresan, U. Meier, J. Masci, and J. Schmidhuber. Multi-column deep neural network for traffic sign classification. *Neural Networks*, 32:333 – 338, 2012. Selected Papers from IJCNN 2011.
- A. N. Escalante-B. and L. Wiskott. Gender and age estimation from synthetic face images with hierarchical slow feature analysis. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 240–249, 2010.

- A. N. Escalante-B. and L. Wiskott. Heuristic evaluation of expansions for non-linear hierarchical Slow Feature Analysis. In *Proc. The 10th International Conference on Machine Learning and Applications*, pages 133–138, Los Alamitos, CA, USA, 2011.
- A. N. Escalante-B. and L. Wiskott. How to solve classification and regression problems with an unsupervised learning algorithm based on the slowness principle. (in preparation), 2012.
- A. N. Escalante-B. and L. Wiskott. How to solve classification and regression problems on high-dimensional data with a supervised extension of Slow Feature Analysis. *Journal of Machine Learning Research*, 14:3683–3719, December 2013.
- A. N. Escalante-B. and L. Wiskott. Improved graph-based sfa: Information preservation complements the slowness principle. e-print arXiv:1601.03945, 01 2016.
- P. Földiák. Learning invariance from transformation sequences. *Neural Computation*, 3(2):194–200, 1991.
- M. Franzius, H. Sprekeler, and L. Wiskott. Slowness and sparseness lead to place, head-direction, and spatial-view cells. *PLoS Computational Biology*, 3(8):1605–1622, 2007.
- M. Franzius, N. Wilbert, and L. Wiskott. Invariant object recognition and pose estimation with Slow Feature Analysis. *Neural Computation*, 23(9):2289–2323, 2011.
- G. Gao and G. Mu. A framework for joint estimation of age, gender and ethnicity on a large database. *Image and Vision Computing*, 32(10):761–770, 2014. Best of Automatic Face and Gesture Recognition 2013.
- X. He. *Locality Preserving Projections*. PhD thesis, Computer Science Department, The University of Chicago, Chicago, IL, USA, 2005.
- X. He and P. Niyogi. Locality Preserving Projections. In *Neural Information Processing Systems*, volume 16, pages 153–160, 2003.
- G. E. Hinton. Connectionist learning procedures. *Artificial Intelligence*, 40(1-3):185–234, 1989.
- S. Houben, J. Stallkamp, J. Sahnen, M. Schlipsing, and C. Igel. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *IJCNN*, number 1288, 2013.
- S. Klampfl and W. Maass. Replacing supervised classification learning by Slow Feature Analysis in spiking neural networks. In *Proc. of NIPS 2009: Advances in Neural Information Processing Systems*, volume 22, pages 988–996. MIT Press, 2010.
- N. Krüger, P. Janssen, S. Kalkan, M. Lappe, A. Leonardis, J. Piater, A. Rodriguez-Sanchez, and L. Wiskott. Deep hierarchies in the primate visual cortex: What can we learn for computer vision? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1847–1871, 2013.
- G. Mitchison. Removing time variation with the anti-Hebbian differential synapse. *Neural Computation*, 3(3):312–320, 1991.
- E. M. Rehn and H. Sprekeler. Nonlinear supervised locality preserving projections for visual pattern discrimination. In *2nd International Conference on Pattern Recognition*, pages 1568–1573, 2014.
- Singular Inversions Inc. FaceGen SDK. <http://www.facegen.com>, 2008.
- H. Sprekeler. On the relation of slow feature analysis and laplacian eigenmaps. *Neural Computation*, 23(12):3287–3302, 2011.
- L. Wiskott. Learning invariance manifolds. In *Proc. of 5th Joint Symposium on Neural Computation, San Diego, CA, USA*, volume 8, pages 196–203. Univ. of California, 1998.
- L. Wiskott. Slow Feature Analysis: A theoretical analysis of optimal free responses. *Neural Computation*, 15(9):2147–2177, 2003.
- L. Wiskott and T. Sejnowski. Slow Feature Analysis: Unsupervised learning of invariances. *Neural Computation*, 14(4):715–770, 2002.
- D. Yi, Z. Lei, and S. Li. Age estimation by multi-scale convolutional network. In *Computer Vision – ACCV 2014*, volume 9005 of *Lecture Notes in Computer Science*, pages 144–158, 2015.
- T. Zhang, D. Tao, X. Li, and J. Yang. Patch alignment for dimensionality reduction. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1299–1313, 2009.

Universal Approximation Results for the Temporal Restricted Boltzmann Machine and the Recurrent Temporal Restricted Boltzmann Machine

Simon Odense

Roderick Edwards

Department of Mathematics

University of Victoria

Victoria, BC, 3800 Finnerty Rd, Canada

STODENSE@UVIC.CA

EDWARDS@UVIC.CA

Editor: Yoshua Bengio

Abstract

The Restricted Boltzmann Machine (RBM) has proved to be a powerful tool in machine learning, both on its own and as the building block for Deep Belief Networks (multi-layer generative graphical models). The RBM and Deep Belief Network have been shown to be universal approximators for probability distributions on binary vectors. In this paper we prove several similar universal approximation results for two variations of the Restricted Boltzmann Machine with time dependence, the Temporal Restricted Boltzmann Machine (TRBM) and the Recurrent Temporal Restricted Boltzmann Machine (RTRBM). We show that the TRBM is a universal approximator for Markov chains and generalize the theorem to sequences with longer time dependence. We then prove that the RTRBM is a universal approximator for stochastic processes with finite time dependence. We conclude with a discussion on efficiency and how the constructions developed could explain some previous experimental results.

Keywords: TRBM, RTRBM, machine learning, universal approximation

1. Introduction

Modeling temporal sequences has been an important problem in machine learning because of the natural time dependence in many data sets. The Restricted Boltzmann Machine (RBM), a type of probabilistic neural network, has become popular as a result of the use of an efficient learning algorithm called contrastive divergence (Hinton, 2002). In particular, its use in the construction of Deep Belief Networks (Hinton and Osindero, 2006) has led to widespread use in a number of machine learning tasks. One major drawback of the basic model, however, is the difficulty in using these models to capture temporal dependence in a data set. Several refinements of the model have attempted to combine the efficient statistical modeling of RBMs with the dynamic properties of Recurrent Neural Networks (Hinton and Osindero, 2006)(Taylor et al., 2006)(Le Roux and Bengio, 2008). These include the Temporal Restricted Boltzmann Machine (TRBM), the Recurrent Temporal Restricted Boltzmann Machine (RTRBM), and the Conditional Restricted Boltzmann Machine (CRBM). Boltzmann machines, and RBMs have been shown to be universal approximators for probability distributions on binary vectors (Freund and Haussler, 1991)(Younes, 1996)(Le Roux and

Bengio, 2008). Furthermore the related Deep Belief Networks have also been shown to be universal approximators even when each hidden layer is restricted to a relatively small number of hidden nodes (Sutskever and Hinton, 2010)(Le Roux and Bengio, 2010)(Montufar and Ay, 2011). The universal approximation of CRBMs follows immediately from that of Boltzmann machines (Montufar et al., 2014). The question we wish to address here is the universal approximation of stochastic processes by TRBMs and RTRBMs.

1.1 The Restricted Boltzmann Machine

An RBM defines a probability distribution over a set of binary vectors $x \in \{0, 1\}^n = X$ as follows

$$P(v, h) = \exp(v^\top W h + c^\top v + b^\top h) / Z$$

where the set of binary vectors X is partitioned into visible and hidden units $X = V \times H$ and Z is the normalization factor, in other words $Z = \sum_{v,h} \exp(v^\top W h + c^\top v + b^\top h)$. This distribution is entirely defined by (W, b, c) and is referred to as a Boltzmann Distribution. We are generally concerned with the marginal distribution of the visible units. When we refer to the distribution of an RBM we are referring to the marginal distribution of its visible units. The marginal distribution of a single visible node is given by

$$P(v_i = 1|h) = \sigma \left(\sum_j w_{ij} h_j + c_i \right)$$

where $\sigma(x) = \frac{1}{1 + \exp(-x)}$. A similar equation holds for the hidden units. Variations of the RBM which use real-valued visible and hidden units (or mixes of the two) exist but will not be considered here.

1.2 Approximation

In order to measure how well one distribution approximates another we use the Kullback-Leibler divergence, which for discrete probability distributions is given by

$$KL(R||P) = \sum_v R(v) \log \left(\frac{R(v)}{P(v)} \right),$$

where v ranges over the sample space of R and P . It can be shown that for any $\epsilon > 0$, given a probability distribution R on V there is a Boltzmann Distribution given by an RBM P such that $KL(R||P) < \epsilon$ (Le Roux and Bengio, 2008)(Freund and Haussler, 1991). Our goal now is to prove the same result where R satisfies certain temporal dependency conditions and P is a TRBM.

2. Universal Approximation Results for the TRBM

A TRBM defines a probability distribution on a sequence $x^T = (x^{(0)}, \dots, x^{(T-1)})$, $x^{(t)} \in \{0, 1\}^n$, $x^{(t)} = (v^{(t)}, h^{(t)})$, given by

$$P(v^{(t)}, h^{(t)} | h^{(t-1)}) = \frac{\exp(v^{(t)T} W h^{(t)} + c^T v^{(t)} + b^T h^{(t)} + h^{(t)T} W' h^{(t-1)})}{Z(h^{(t-1)})},$$

$$P(v^T, h^T) = \left(\prod_{k=1}^{T-1} P(v^{(k)}, h^{(k)} | h^{(k-1)}) \right) R_0(v^{(0)}, h^{(0)}).$$

This distribution is defined by the same parameters as the RBM along with the additional parameters W' . The TRBM can be seen as an RBM with a dynamic hidden bias determined by $W' h^{(t-1)}$. The initial distribution, $P_0(v^{(0)}, h^{(0)})$, is the same as $P(v^{(t)}, h^{(t)} | h^{(t-1)})$ with $h^{(0)T} b_{\text{init}}$ replacing $h^{(t)T} W' h^{(t-1)}$ for some initial hidden bias b_{init} . Note that W' is not symmetric in general. We call the connections between $h^{(t-1)}$ and $h^{(t)}$ with weights in W' temporal connections.

2.1 Universal Approximation Results for the Basic TRBM

Our approximation results will deal with distributions which are time-homogeneous and have finite time dependence. These distributions can be written in the form

$$R(v^T) = \left(\prod_{k=m}^{T-1} R_1(v^{(k)} | v^{(k-1)}, \dots, v^{(k-m)}) \right) R_0(v^{(0)}, \dots, v^{(m-1)})$$

where R_1 is the transition probability and R_0 is the initial distribution. We first show that a TRBM can approximate a Markov chain (distributions of the above form with $m = 1$) for a finite number of time steps to arbitrary precision. We begin by proving a lemma. Here P_t is the marginal distribution of P over $(v^{(0)}, \dots, v^{(t)})$. Similarly $R_{0,t}$ is the marginal distribution of R_0 over $(v^{(0)}, \dots, v^{(t)})$.

Lemma 1: *Let R be a distribution on a finite sequence of length T of n -dimensional binary vectors that is time homogeneous with finite time dependence. Given a set of distributions \mathbf{P} on the same sequences, if for every $\epsilon > 0$ we can find a distribution $P \in \mathbf{P}$ such that for every v^T ,*

$$KL(R_1(\cdot | v^{(t-1)}, \dots, v^{(t-m)}) || P_t(\cdot | v^{(t-1)}, \dots, v^{(t-m)})) < \epsilon \text{ for } m \leq t < T - 1,$$

$$KL(R_{0,t}(\cdot | v^{(t-1)}, \dots, v^{(0)}) || P_0(\cdot | v^{(t-1)}, \dots, v^{(0)})) < \epsilon \text{ for } 0 < t < m,$$

and $KL(R_{0,0}(\cdot) || R_0(\cdot)) < \epsilon$, then we can find distributions $P \in \mathbf{P}$ to approximate R to arbitrary precision.

Proof: The proof is given in the appendix.

Now we use this lemma to prove our first universal approximation theorem. In this case \mathbf{P} is the set of distributions given by a TRBM. Note that throughout this paper P will often be used to refer to a TRBM. When we say P is a TRBM we are referring to the distribution associated with a set of parameters defining the TRBM.

Theorem 1: *Let R be a distribution over a sequence of length T of binary vectors of length n that is time homogeneous and satisfies the Markov property. For any $\epsilon > 0$ there exists a TRBM defined on sequences of length T of binary vectors of length n with distribution P such that $KL(R || P) < \epsilon$.*

Proof: By the previous lemma we will be looking for a TRBM that can approximate the transition probabilities of R along with its initial distribution. The proof will rely on the universal approximation properties of RBMs. The idea is that given one of the 2^n configurations of the visible units, v , there is an RBM with distribution P_v approximating $R_1(\cdot | v^{(t-1)} = v)$ to a certain precision. The universal approximation results for RBMs tell us that this approximation can be made arbitrarily precise for an RBM with enough hidden units. Furthermore, this approximation can be done with visible biases set to 0 (Le Roux and Bengio, 2008). We thus set all visible biases of our TRBM to 0 and include each of the approximating RBMs without difficulty. We label these RBMs H_1, \dots, H_{2^n} . Given a specific configuration of the visible nodes v , H_v refers to the RBM chosen to approximate $R_1(\cdot | v^{(t-1)} = v)$.

The challenge then is to signal the TRBM which of the 2^n RBMs should be active at the next time step. To do this we include 2^n additional hidden nodes which we will call *control nodes*, each corresponding to a particular configuration of the visible units. Thus we add 2^n control nodes, $h_{c,1}, \dots, h_{c,2^n}$ corresponding to the hidden nodes H_1, \dots, H_{2^n} . Again, given a particular visible configuration v , we denote the corresponding control node by $h_{c,v}$. The set of all control nodes will be denoted H_c . Note that (c, v) is the label of $h_{c,v}$ and weights involving $h_{c,v}$ will be denoted $w_{c,v}^{(c,v)}$ or $w_{j_i}^{(c,v)}$. The control nodes will signal which of the H_i 's should be active at the next time step. To accomplish this we will choose parameters such that when v is observed at time $t - 1$, $h_{c,v}$ will be on at time $t - 1$ with a probability close to 1 and every other control node will be off at time $t - 1$ with probability close to 0. Each $h_{c,v}$ will have strong negative temporal connections to every $H_{v'}$ with $v' \neq v$, in essence turning off every RBM corresponding to $R_1(\cdot | v^{(t-1)} = v')$, and leaving the RBM corresponding to $R_1(\cdot | v^{(t-1)} = v)$ active (see Fig. 1). We will break the proof down into four parts and we must be able to choose parameters that satisfy all four conditions.

First, we must be able to choose parameters so that given $v^{(t-1)} = v$, the probability that $h_{c,v}^{(t-1)} = 1$ can be made arbitrarily close to 1 and the probability that $h_{c,v'}^{(t-1)} = 1$ can be made arbitrarily close to 0. Second, we must have that the control nodes have no impact on the visible distribution at the same time step so the transition probabilities of the TRBM will still approximate $R_1(\cdot | v^{(t-1)} = v)$. Third, we must be able to choose parameters so that given $h_{c,v}^{(t-1)} = 1$ and $h_{c,v'}^{(t-1)} = 0$, the probability that any nodes in $H_{v'}$ are on at time t can be made arbitrarily close to 0 for $v' \neq v$. Finally, we must be able to approximate the initial distribution R_0 .

Note in the TRBM temporal data cannot flow directly from the visible nodes to the hidden nodes at the next time step. In contrast, by using the visible nodes at time t as input and the visible nodes at time $t + 1$ as output, in the CRBM there is a direct relationship between the visible nodes at time t and the hidden nodes at time $t + 1$. The CRBM is known to be a universal approximator (Montufar et al., 2014). The main challenge for the TRBM is to

encode the visible state in the control unit so that information can be passed to the hidden nodes at the next time step without the encoding changing the visible distribution. This is covered in Steps 1 and 2. Step 3 verifies that the correct distribution can be recovered from the encoding and Step 4 shows we can simulate the initial distribution without changing the rest of the machine.

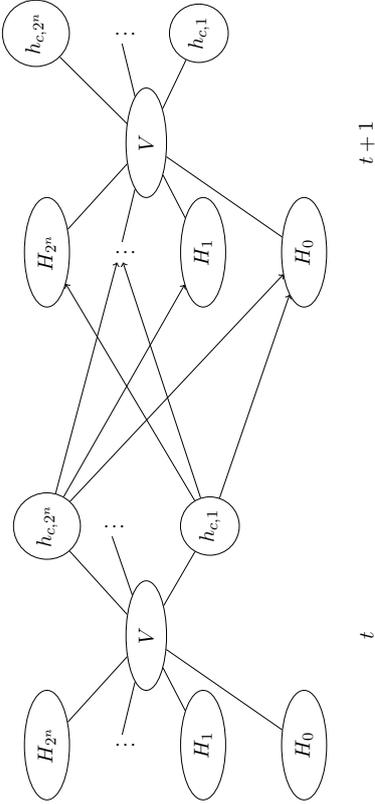


Figure 1: Interactions between sets of nodes within and between time steps: each $h_{c,v}$ turns off every $H_{v'} : v \neq v'$ in the subsequent time step. $h_{c,v}$ will only be on if v is observed at time t and collectively the control nodes H_c have negligible effect on the visible distribution. H_0 is an additional set of hidden units that will be used to model the initial distribution.

To choose temporal connections, define $w'_{j,(c,v)}$ to be $-\alpha$ if $h_j \in H_{v'}$ where $v' \neq v$ and 0 otherwise. Let every other $w'_{i,j} = 0$. In particular, remembering that W' is not necessarily symmetric, we have $w'_{(c,v),j} = 0$ for all h_j . The only parameters left to define are the biases and visible-hidden connections of H_c . Let $b_{(c,v)} = -(k - 0.5)\beta$ where k is the number of visible nodes on in v . Finally, define the connections from the visible units to $h_{c,v}$ by $w_{i,(c,v)} = \beta$ if v_i is on in the configuration v and $-\beta$ otherwise. Here the parameters of the control nodes are completely determined by α and β . We will proceed by showing that all necessary conditions are satisfied when α and β are large enough.

A note on notation. Throughout the proof H will denote the set of hidden nodes and $H^{(t)}$ will denote the set of configurations of hidden nodes at time t . Similarly $H_v^{(t)}$ will denote the set of configurations of the hidden nodes in which $h_i^{(t)} = 0$ if $h_i \notin H_v$. Similar conventions are used for H_c . $(H \setminus H_c)^{(t)}$ then denotes the set of configurations of non-control nodes. This is used in scenarios where we want to sum over a certain subset of hidden nodes and ignore the others, which is equivalent to simply setting all other nodes to 0. $H_{c,v}^{(t)}$ denotes the set of configurations of $h^{(t)}$ with $h_{c,v}^{(t)} = 1$ and $h_{c,v'}^{(t)} = 0$ for $v \neq v'$. $\bar{H}_{c,v}^{(t)}$ denotes the set of configurations $H^{(t)} \setminus H_{c,v}^{(t)}$. See Appendix A for a partial list of relevant notation.

Step 1:

For this step we show that as $\beta \rightarrow \infty$ we have $P(H_{c,v}^{(t)} | v^{(t)}, \dots, v^{(0)}) \rightarrow 1$. Note that given the visible state at time t , the state of the control nodes at time t is conditionally independent of all other previous states. With this in mind, we can write the probability of a control node being on at time t as

$$P(h_{c,v}^{(t)} = 1 | v^{(t)}, \dots, v^{(0)}) = \sigma \left(\sum_i v_i^{(t)} w_{i,(c,v)} + b_{(c,v)} \right),$$

where σ is the logistic function. Note that for all $v^{(t)} \in \{0, 1\}^n$ and $v \in \{0, 1\}^n$, $\sum_i v_i^{(t)} w_{i,(c,v)} = a\beta - b\beta$ where a is the number of nodes on in both v and $v^{(t)}$ and b is the number of nodes on in $v^{(t)}$ but off in v . Since $b_{(c,v)} = -(k - 0.5)\beta$ if $v \neq v^{(t)}$, then either $a < k$, in which case $\sum_i v_i^{(t)} w_{i,(c,v)} + b_{(c,v)} \leq -0.5\beta$, or $a = k$ and $b \geq 1$, which again implies $\sum_i v_i^{(t)} w_{i,(c,v)} + b_{(c,v)} \leq -0.5\beta$. If $v = v^{(t)}$ then $a\beta - b\beta + b_{(c,v)} = k\beta - (k - 0.5)\beta = 0.5\beta$. Thus if $v = v^{(t)}$,

$$\sigma \left(\sum_i v_i^{(t)} w_{i,(c,v)} + b_{(c,v)} \right) = \sigma(0.5\beta).$$

Otherwise

$$\sigma \left(\sum_i v_i^{(t)} w_{i,(c,v)} + b_{(c,v)} \right) \leq \sigma(-0.5\beta).$$

So as $\beta \rightarrow \infty$, $P(H_{c,v}^{(t)} | v^{(t)}, \dots, v^{(0)}) \rightarrow 1$. In other words, for all $v^{(t)}, \dots, v^{(0)}$ and all $\epsilon_0 > 0$ there exists some β_0 such that $\beta > \beta_0$ implies $|1 - P(H_{c,v}^{(t)} | v^{(t)}, \dots, v^{(0)})| = P(\bar{H}_{c,v}^{(t)} | v^{(t)}, \dots, v^{(0)}) < \epsilon_0$.

Step 2:

Here we show that by making β large enough we can make the effect of the control nodes on the visible distribution at the same time step negligible. Take any $v^{(t)}$, for all $h^{(t-1)}$, we have

$$\begin{aligned} P(v^{(t)} | h^{(t-1)}) &= \frac{P(v^{(t)} | h^{(t-1)})}{\sum_{v^{(t)}} P(v^{(t)} | h^{(t-1)})} \\ &= \frac{\sum_{h^{(t)} \in \bar{H}_{c,v}^{(t)}} P(v^{(t)}, h^{(t)} | h^{(t-1)}) + \sum_{h^{(t)} \in H_{c,v}^{(t)}} P(v^{(t)}, h^{(t)} | h^{(t-1)})}{\sum_{v^{(t)}} \sum_{h^{(t)} \in \bar{H}_{c,v}^{(t)}} P(v^{(t)}, h^{(t)} | h^{(t-1)}) + \sum_{v^{(t)}} \sum_{h^{(t)} \in H_{c,v}^{(t)}} P(v^{(t)}, h^{(t)} | h^{(t-1)})}. \end{aligned} \quad (1)$$

We also have that $P(v^{(t)}, h^{(t)} | h^{(t-1)}) = P(h^{(t)} | v^{(t)}, h^{(t-1)}) P(v^{(t)} | h^{(t-1)})$ for all $h^{(t)}, v^{(t)}$ and by definition

$$\sum_{h^{(t)} \in \bar{H}_{c,v}^{(t)}} P(h^{(t)} | v^{(t)}, h^{(t-1)}) P(v^{(t)} | h^{(t-1)}) = P(\bar{H}_{c,v}^{(t)} | v^{(t)}, h^{(t-1)}) P(v^{(t)} | h^{(t-1)}).$$

By Step 1, there exists a β_0 such that for any $\beta > \beta_0$ we have $P(\bar{H}_{c,\alpha^{(t)}}^{(t)} | v^{(t)}, \dots, v^{(0)}) < \epsilon_0$ for all $v^{(t)}, \dots, v^{(t-1)}$. Since the only connections going to a control node are from the visible units, given $v^{(t)}$, the state of the control nodes are conditionally independent of $v^{(t-1)}, \dots, v^{(0)}$ and $h^{(t-1)}$. Since $P(v^{(t)} | h^{(t-1)}) < 1$, we have that $\beta > \beta_0$ implies that $P(\bar{H}_{c,\alpha^{(t)}}^{(t)} | v^{(t)}, h^{(t-1)}) P(v^{(t)} | h^{(t-1)}) < \epsilon_0$, giving us that for $\beta > \beta_0$ and all $v^{(t)}$,

$$\sum_{h^{(t)} \in H^{(t)}} \sum_{c,\alpha^{(t)}} P(v^{(t)}, h^{(t)} | h^{(t-1)}) < \epsilon_0. \quad (2)$$

Note that this inequality is independent of α . Increasing α has no effect on $P(\bar{H}_{c,\alpha^{(t)}}^{(t)} | v^{(t)}, h^{(t-1)})$ and $P(v^{(t)} | h^{(t-1)})$ is bounded above by 1 so even after increasing α arbitrarily the inequality will hold with the same choice of β . Looking back to equation (1), as $\beta \rightarrow \infty$ the right hand terms in both the numerator and denominator go to 0. Consider $\sum_{h^{(t)} \in H^{(t)}} P(v^{(t)}, h^{(t)} | h^{(t-1)})$. For all $v^{(t)}$, this is bounded above by 1 and since we are

summing over $h^{(t)} \in H^{(t)}$, Step 1 tells us this is strictly increasing in β . This tells us that limit of the numerator and denominator of (1) are both finite and non-zero giving us that

$$\lim_{\beta \rightarrow \infty} P(v^{(t)} | h^{(t-1)}) = \lim_{\beta \rightarrow \infty} \frac{\sum_{h^{(t)} \in H^{(t)}} P(v^{(t)}, h^{(t)} | h^{(t-1)})}{\sum_{v^{(t)} \in H^{(t)}} \sum_{c,\alpha^{(t)}} P(v^{(t)}, h^{(t)} | h^{(t-1)})}.$$

Define

$$\begin{aligned} \tilde{P}(v^{(t)} | h^{(t-1)}) &:= \frac{\sum_{h^{(t)} \in H^{(t)}} P(v^{(t)}, h^{(t)} | h^{(t-1)})}{\sum_{v^{(t)} \in H^{(t)}} \sum_{c,\alpha^{(t)}} P(v^{(t)}, h^{(t)} | h^{(t-1)})} \\ &= \sum_{h^{(t)} \in H^{(t)}} \exp \left(\sum_{i,j,h_j \in (H \setminus H_c)} v_i^{(t)} h_j^{(t)} w_{i,j} + \sum_{j,h_j \in (H \setminus H_c)} b_j h_j^{(t)} + 0.5\beta + \sum_{i,j} h_i^{(t)} h_j^{(t-1)} w'_{i,j} \right) \\ &= \sum_{v^{(t)} \in H^{(t)}} \sum_{c,\alpha^{(t)}} \exp \left(\sum_{i,j,h_j \in (H \setminus H_c)} v_i^{(t)} h_j^{(t)} w_{i,j} + \sum_{j,h_j \in (H \setminus H_c)} b_j h_j^{(t)} + \sum_{i,j} h_i^{(t)} h_j^{(t-1)} w'_{i,j} \right) \\ &= \frac{\sum_{h^{(t)} \in H^{(t)}} \exp \left(\sum_{i,j,h_j \in (H \setminus H_c)} v_i^{(t)} h_j^{(t)} w_{i,j} + \sum_{j,h_j \in (H \setminus H_c)} b_j h_j^{(t)} + \sum_{i,j} h_i^{(t)} h_j^{(t-1)} w'_{i,j} \right)}{\sum_{v^{(t)} \in H^{(t)}} \sum_{c,\alpha^{(t)}} \exp \left(\sum_{i,j,h_j \in (H \setminus H_c)} v_i^{(t)} h_j^{(t)} w_{i,j} + \sum_{j,h_j \in (H \setminus H_c)} b_j h_j^{(t)} + \sum_{i,j} h_i^{(t)} h_j^{(t-1)} w'_{i,j} \right)}, \end{aligned}$$

but this is just the probability of $v^{(t)}$ when we remove the control nodes. Thus for any $v^{(t)}$ and $\epsilon_1 > 0$ there exists a β_0 such that $\beta > \beta_0$ implies that for all $h^{(t-1)}$, $|P(v^{(t)} | h^{(t-1)}) - \tilde{P}(v^{(t)} | h^{(t-1)})| < \epsilon_1$. Furthermore, this is unchanged by increasing α .

Step 3:

In this step, remembering that P_v is the distribution of the RBM corresponding to $R_{h_1} \cdot (v^{(t-1)} = v)$, we show that as α and β are increased to infinity, if $h^{(t-1)} \in H_{c,v}^{(t-1)}$ then $P(v^{(t)} | h^{(t-1)}) \rightarrow P_v(v^{(t)})$ for all $v^{(t)}$. First note that since the states of any two hidden nodes at time t are independent, $P(h_j^{(t)} = 1 | v^{(t)}, h^{(t-1)}) = \tilde{P}(h_j^{(t)} = 1 | v^{(t)}, h^{(t-1)})$. Here \tilde{P} is the system without control nodes, defined in the previous step. Take any $v^{(t)}$ and consider some configuration $h^{(t-1)} \in H_{c,v}^{(t-1)}$. We have $h_{c,v}^{(t-1)} = 0$ for all $v' \neq v$ and $w_{j(c,v)} = 0$ for $h_j \in H_v$, giving us

$$\tilde{P}(h_j^{(t)} = 1 | v^{(t)}, h^{(t-1)}) = \sigma \left(\sum_i v_i^{(t)} w_{i,j} + b_j \right).$$

This is $P_v(h_j^{(t)} = 1 | v^{(t)})$. Now take a hidden unit $h_j \in H_{v'}$ with $v' \neq v$. Since $v' \neq v$ and $h^{(t-1)} \in H_{c,v}^{(t-1)}$, then $h_{c,v}^{(t-1)} = 1$ and $w_{j(c,v)} = -\alpha$. This gives us

$$\tilde{P}(h_j^{(t)} = 1 | v^{(t)}, h^{(t-1)}) = \sigma \left(\sum_i v_i^{(t)} w_{i,j} + b_j - \alpha \right).$$

Since h_j is not a control node, $w_{i,j}$ is fixed for all v_i . Thus as $\alpha \rightarrow \infty$, $\tilde{P}(h_j^{(t)} = 1 | v^{(t)}, h^{(t-1)}) \rightarrow 0$. So for any $\epsilon_0 > 0$ there exists α_0 such that $\alpha > \alpha_0$ implies that if $h_j \in H_{v'}$ with $v' \neq v$, $|\tilde{P}(h_j^{(t)} = 1 | v^{(t)}, h^{(t-1)})| < \epsilon_0$. Now we have

$$\begin{aligned} \tilde{P}(v^{(t)} | h^{(t-1)}) &= \sum_{h^{(t)} \in (H \setminus H_c)^{(t)}} \tilde{P}(v^{(t)}, h^{(t)} | h^{(t-1)}) \\ &= \sum_{h^{(t)} \in H_c^{(t)}} \tilde{P}(v^{(t)}, h^{(t)} | h^{(t-1)}) + \sum_{h^{(t)} \in ((H \setminus H_c)^{(t)} \setminus H_c^{(t)})} \tilde{P}(v^{(t)}, h^{(t)} | h^{(t-1)}). \end{aligned} \quad (3)$$

Note that $\tilde{P}(v^{(t)}, h^{(t)} | h^{(t-1)}) = \tilde{P}(h^{(t)} | v^{(t)}, h^{(t-1)}) \tilde{P}(v^{(t)} | h^{(t-1)}, h_j^{(t)})$ and $h_j^{(t)}$ are independent for all i, j , and $\tilde{P}(v^{(t)} | h^{(t-1)}) < 1$. Thus we have that $\alpha > \alpha_0$ implies that if $h^{(t)} \in (H \setminus H_c)^{(t)} \setminus H_c^{(t)}$, then $\tilde{P}(h^{(t)} | v^{(t)}, h^{(t-1)}) \tilde{P}(v^{(t)} | h^{(t-1)}) < \epsilon_0$. So as $\alpha \rightarrow \infty$, the right hand term of (3) goes to 0. So for any ϵ_1 there exists an α_1 such that $\alpha > \alpha_1$ implies $|\tilde{P}(v^{(t)} | h^{(t-1)}) - \sum_{h^{(t)} \in H_c^{(t)}} \tilde{P}(v^{(t)}, h^{(t)} | h^{(t-1)})| < \epsilon_1$. Note that since there are a finite number

of configurations, $v^{(t)}$, we can take α large enough so that this is true for all $v^{(t)}$. So for any ϵ_1 we can choose $\alpha > \alpha_1$ so that

$$\left| \tilde{P}(v^{(t)} | h^{(t-1)}) - \frac{\sum_{h^{(t)} \in H_c^{(t)}} \tilde{P}(v^{(t)}, h^{(t)} | h^{(t-1)})}{\sum_{v^{(t)}, h^{(t)} \in H_c^{(t)}} \tilde{P}(v^{(t)}, h^{(t)} | h^{(t-1)})} \right| < \epsilon_1,$$

but we have

$$\begin{aligned} \frac{\sum_{h^{(t)} \in H_v^{(t)}} \tilde{P}(v^{(t)}, h^{(t)}) |h^{(t-1)}|}{\sum_{v^{(t)}, h^{(t)} \in H_v^{(t)}} \tilde{P}(v^{(t)}, h^{(t)}) |h^{(t-1)}|} &= \frac{\sum_{h^{(t)} \in H_v^{(t)}} \exp\left(\sum_{i,j} v_i^{(t)} h_j^{(t)} w_{i,j} + \sum_j b_j h_j^{(t)}\right)}{\sum_{v^{(t)}, h^{(t)} \in H_v^{(t)}} \exp\left(\sum_{i,j} v_i^{(t)} h_j^{(t)} w_{i,j} + \sum_j b_j h_j^{(t)}\right)} \\ &= P_v(v^{(t)}). \end{aligned}$$

To summarize, Step 3 tells us that for any given $v^{(t)}$ and all $h^{(t-1)} \in H_{c,v}^{(t-1)}$ and any $\epsilon_1 > 0$, there exists α_1 such that $\alpha > \alpha_1$ implies that $|\tilde{P}(v^{(t)}) |h^{(t-1)}| - P_v(v^{(t)})| < \epsilon_1$.

Step 4:

Finally, we must also be able to approximate the initial distribution R_0 to arbitrary precision. We know there is an RBM H_0 with visible biases 0 whose Boltzmann distribution can approximate R_0 to a certain precision. Include this machine in our TRBM. Now we define the initial biases. Let $b_{i,init} = \gamma$ for every $h_i \in H_0$ and $b_{c,v,init} = 0$ for all (c, v) . Set $b_{i,init} = -\gamma$ for all other hidden nodes. Add $-\gamma$ to the biases of H_0 . Call the distribution of this modified machine \tilde{P} . By Step 2, for any $v^{(t)}$, and any $\epsilon_0 > 0$, there exists β_0 such that $\beta > \beta_0$ implies $|\tilde{P}_0(v^{(t)}, h^{(0)}) - \tilde{P}_0(v^{(t)}, h^{(0)})| < \epsilon_0$. If $h_k \in H_v$ for some v , we have

$$\tilde{P}_0(h_k^{(0)}) = 1|v^{(0)}| = \sigma\left(\sum_i v_i^{(0)} w_{i,k} + b_k - \gamma\right),$$

and for $h_j \in H_0$ we have

$$\tilde{P}_0(h_j^{(0)}) = 1|v^{(0)}| = \sigma\left(\sum_i v_i^{(0)} w_{i,j} + b_j\right).$$

Note that \tilde{P}_0 does not depend on α or β . Following the same logic as in Step 3, for any $\epsilon_0 > 0$, there exists γ_0 such that $\gamma > \gamma_0$ implies $\tilde{P}_0(v^{(0)}, h^{(0)}) < \epsilon_0$ if $h^{(0)} \in (H \setminus H_c)^{(t)} \setminus H_0^{(t)}$ for some H_v . So for all $v^{(t)}$, $\tilde{P}_0(v^{(0)})$ can be made arbitrarily close to the probability of $v^{(0)}$ in the Boltzmann distribution of H_0 , which by construction approximates R_0 . At subsequent time steps, for each $h_j \in H_0$ we have $P(h_j^{(t)}) = 1|h^{(t-1)}, v^{(t)}| = \sigma(\sum_i w_{i,j} v_i^{(t)} + b_j - \gamma)$. This can be made arbitrarily close to 0 by making γ arbitrarily large, so $P(h_j^{(t)}) = 1, v^{(t)} |h^{(t-1)}$ can be made arbitrarily close to 0. Thus for any $\epsilon_0 > 0$ there exists γ_1 such that $\gamma > \gamma_1$ implies

$$|\tilde{P}(v^{(t)}) |h^{(t-1)}| - \sum_{h^{(t)} \notin H_0^{(t)}} \tilde{P}(v^{(t)}, h^{(t)}) |h^{(t-1)}| < \epsilon_0. \quad (4)$$

But since γ does not appear anywhere else for $t > 0$, $\sum_{h^{(t)} \notin H_0^{(t)}} \tilde{P}(v^{(t)}, h^{(t)}) |h^{(t-1)}| = P(v^{(t)}) |h^{(t-1)}$.

Note that this construction allows the control nodes to be active in the first time step and to transmit temporal data without disturbing the initial distribution.

Now we put the four steps together. Given an arbitrary $0 < t < T$, we can write each $P(v^{(t)}) |v^{(t-1)}, \dots, v^{(0)}$ as

$$\begin{aligned} &\sum_{h^{(t-1)}} P(v^{(t)}, h^{(t-1)}) |v^{(t-1)}, \dots, v^{(0)} \\ &= \sum_{h^{(t-1)}} P(v^{(t)}) |h^{(t-1)}, v^{(t-1)}, \dots, v^{(0)} P(h^{(t-1)}) |v^{(t-1)}, \dots, v^{(0)} \\ &= \sum_{h^{(t-1)}} P(v^{(t)}) |h^{(t-1)} P(h^{(t-1)}) |v^{(t-1)}, \dots, v^{(0)}. \end{aligned}$$

Step 1 tells us that if $h^{(t-1)} \notin H_{c,v}^{(t-1)}$, then $\lim_{\beta \rightarrow \infty} P(h^{(t-1)}) |v^{(t-1)}, \dots, v^{(0)}| = 0$. Step 2 tells us that $\lim_{\beta \rightarrow \infty} P(v^{(t)}) |h^{(t-1)}| = \tilde{P}(v^{(t)}) |h^{(t-1)}|$. Since P is continuous in terms of β , for any ϵ_1 , there exists β_0 such that $\beta > \beta_0$ implies

$$\begin{aligned} &|\sum_{h^{(t-1)}} P(v^{(t)}) |h^{(t-1)} P(h^{(t-1)}) |v^{(t-1)}, \dots, v^{(0)}| \\ &= \sum_{h^{(t-1)} \in H_{c,v}^{(t-1)}} \tilde{P}(v^{(t)}) |h^{(t-1)} P(h^{(t-1)}) |v^{(t-1)}, \dots, v^{(0)}| < \epsilon_1. \end{aligned} \quad (5)$$

Step 3 tells us that for any $\epsilon_0 > 0$, there exists an α_0 such that for all $h^{(t-1)} \in H_{c,v}^{(t-1)}$, if $\alpha > \alpha_0$ we have $|\tilde{P}(v^{(t)}) |h^{(t-1)}| - P_v(v^{(t)}) |v^{(t-1)}| < \epsilon_0$. So for any ϵ_1 , there exists an α_0 such that $\alpha > \alpha_0$ implies

$$\begin{aligned} &|\sum_{h^{(t-1)} \in H_{c,v}^{(t-1)}} \tilde{P}(v^{(t)}) |h^{(t-1)} P(h^{(t-1)}) |v^{(t-1)}, \dots, v^{(0)}| \\ &= P_v(v^{(t-1)}) |v^{(t)}| \sum_{h^{(t-1)} \in H_{c,v}^{(t-1)}} P(h^{(t-1)}) |v^{(t-1)}, \dots, v^{(0)}| < \epsilon_1. \end{aligned} \quad (6)$$

Again by Step 1, as β goes to infinity, $\sum_{h^{(t-1)} \in H_{c,v}^{(t-1)}} P(h^{(t-1)}) |v^{(t-1)}, \dots, v^{(0)}| \rightarrow 1$, so for any ϵ_1 there exists β_1 such that $\beta > \beta_1$ implies that

$$|P_v(v^{(t-1)}) |v^{(t)}| \sum_{h^{(t-1)} \in H_{c,v}^{(t-1)}} P(h^{(t-1)}) |v^{(t-1)}, \dots, v^{(0)}| - P_v(v^{(t-1)}) |v^{(t)}| < \epsilon_1. \quad (7)$$

Now take any $\epsilon_2 > 0$ and take $\epsilon_1 < \epsilon_2/4$ with corresponding $\beta_0, \beta_1, \alpha_0$ so that the inequalities in (5), (6) and (7) hold. Then from Step 4 there exist γ_0, β_2 such that $\gamma > \gamma_0$ and $\beta > \beta_2$ implies that (4) holds. Then taking $\beta > \max(\beta_0, \beta_1, \beta_2)$, $\alpha > \alpha_0$, $\gamma > \gamma_0$ and applying the triangle inequality to (4), (5), (6), and (7) we have that $|\tilde{P}(v^{(t)}) |v^{(t-1)}, \dots, v^{(0)}| - P_v(v^{(t-1)}) |v^{(t)}| < \epsilon_2$. Since there are a finite number of configurations $v^{(t)}, v^{(t-1)}, \dots, v^{(0)}$, we can choose α, β, γ so that this holds for all $v^{(t)}, v^{(t-1)}, \dots, v^{(0)}$ and by construction, $KL(R(\cdot |v^{(t-1)})) |P_{v^{(t-1)}}| < \epsilon$

for some arbitrarily chosen ϵ . Since the KL -divergence as a function of α , β , and γ is continuous, for any $\epsilon' > 0$ we can find $\alpha_1, \beta_2, \gamma_1$ such that $\alpha > \alpha_1$, $\beta > \beta_2$, $\gamma > \gamma_1$ implies that $|KL(R(\cdot|v^{(t-1)}))|P(\cdot|v^{(t-1)}, \dots, v^{(0)}) - KL(R(\cdot|v^{(t-1)}))|P_{\epsilon'}(\cdot|v^{(t-1)})| < \epsilon'$. And $KL(R(\cdot|v^{(t-1)}))|P_{\epsilon'}(\cdot|v^{(t-1)})| < \epsilon$ for some arbitrarily chosen ϵ . So we can choose parameters such that $KL(R(\cdot|v^{(t-1)}))|P(\cdot|v^{(t-1)}, \dots, v^{(0)})| < \epsilon$. By the same argument, Step 4 tells us that we can choose parameters so that $KL(R_0||P_0) < \epsilon$. Thus by Lemma 1 the result holds. ■

Note that following the remark after the proof of Lemma 1, if we have a TRBM which approximates R over T time steps to a certain precision, it also approximates R over $t < T$ time steps to at least the same precision since the construction satisfies the conditions of Lemma 1.

2.2 The Generalized TRBM

The TRBM used in the previous section is a restricted instance of a more generalized model described by Sutskever et al. (2006). In the full model we allow explicit long-term hidden-hidden connections as well as long-term visible-visible connections. In this paper we will not consider models with visible-visible temporal interaction. From a practical standpoint any learning algorithm operating on a class of models with visible-visible interactions would be able to make those connections arbitrarily small if it helped, so in practice the class of models with visible-visible temporal connections is bigger than the one without any. The generalized TRBM is given by

$$P(v^{(t)}, h^{(t)}|h^{(t-1)}, \dots, h^{(0)}) \\ = \frac{\exp(v^{(t)\top} W h^{(t)} + c^\top v^{(t)} + b^\top h^{(t)} + h^{(t)\top} W^{(1)} h^{(t-1)} + \dots + h^{(t)\top} W^{(m)} h^{(t-m)})}{Z(h^{(t-1)}, \dots, h^{(0)})},$$

$$P(v^{(t)}, h^{(t)}|h^{(t-1)}, \dots, h^{(0)}) \\ \times P_0(v^{(0)}, h^{(0)}),$$

$$P(v^{(t)}, h^{(t)}|h^{(t-1)}, \dots, h^{(t-m)}) \\ = \left(\prod_{i=m}^{t-1} P(v^{(i)}, h^{(i)}|h^{(i-1)}, \dots, h^{(t-m)}) \right) \left(\prod_{i=1}^{m-1} P(v^{(i)}, h^{(i)}|h^{(i-1)}, \dots, h^{(0)}) \right)$$

where we have a finite number of weight matrices $W^{(i)}$ used to determine the bias at time t . We replace $W^{(k)} h^{(t-k)}$ with an initial bias $b_{init}^{(k)}$ if $k > t$. The distribution $P(v^{(t)}, h^{(t)})$ is then given by

$$P(v^{(t)}, h^{(t)}|h^{(t-1)}, \dots, h^{(0)}) \\ \times P_0(v^{(0)}, h^{(0)}),$$

If we drop the restriction that R be a Markov chain we can generalize the previous theorem so that R is any distribution homogeneous in time with a finite time dependence.

Theorem 2: *Let R be a distribution over a sequence of length T of binary vectors of length n that is time homogeneous and has finite time dependence. For any $\epsilon > 0$ there exists a generalized TRBM, P , such that $KL(R||P) < \epsilon$.*

Proof: The initial part of the proof is identical to the proof of Theorem 1. Let m be the time dependence of R . Then for each visible sequence $v^{(t-1)}, \dots, v^{(t-m)}$ we construct a TRBM P by adding sets of hidden units $H_{v^{(t-1)}, \dots, v^{(t-m)}}$ with parameters chosen to approximate $R_1(\cdot|v^{(t-1)}, \dots, v^{(t-m)})$. Note that although the indices here are written as $v^{(t-1)}, \dots, v^{(t-m)}$,

they do not depend on the time step t . Rather, there is one set of hidden nodes added for each configuration of an m -length sequence of visible nodes. The superscripts are added to distinguish different vectors in the sequence as well as emphasize how the connections should be made.

For each visible configuration v we add a control unit $h_{c,v}$ with the same bias and visible-hidden connections (determined by a parameter β) as in the construction for Theorem 1. If $i \leq m$, define the i -step temporal connections as $w_{(c,v)j}^{(i)} = -\alpha$, if $h_j \in H_{v^{(t-1)}, \dots, v^{(t-m)}}$ with $v^{(t-i)} \neq v$ and 0 otherwise. All other temporal connections are set to 0. Then repeating Step 1, Step 2, and Step 3 in Theorem 1, by making α and β sufficiently large we can make $KL(R_1(\cdot|v^{(t-1)}, \dots, v^{(t-m)}))|P(\cdot|v^{(t-1)}, \dots, v^{(0)})|$ arbitrarily small for all $v^{(t)}$.

To finish the proof we must modify the machine to approximate the m initial distributions as well. In practice, one could train an RBM with the first m time steps as input in order to simulate the initial distribution. In this case the remainder of the proof is identical to step 4 of Theorem 1. The proof for the general TRBM as defined above is more intricate. In order to simulate the initial distributions with the general TRBM, first set $b_{init}^{(k)}$ to $-\gamma$ for each node $h \in H_{v^{(t-1)}, \dots, v^{(t-m)}}$ and all k , and set $b_{init}^{(k)}$ to 0 for every control node. Now for each sequence $v^{(t-1)}, \dots, v^{(0)}$ with $i < m$ add a set of hidden units $H_{v^{(t-1)}, \dots, v^{(0)}}$ to approximate $R_{0,i}(\cdot|v^{(t-1)}, \dots, v^{(0)})$ to a certain precision. For each i , call the set of all of these hidden units $H_{(i)}$. Connect each of these sets to the control nodes in the same way as done previously. In other words if $h_j \in H_{v^{(t-1)}, \dots, v^{(0)}}$ then $w_{j(c,v)}^{(l)} = -\alpha$ if $v^{(t-l)} \neq v$ and 0 otherwise. Add $-\gamma$ to the bias of each h_j if $h_j \in H_{(i)}$ for some i . For each $h_j \in H_{(i)}$ let $b_{init}^{(l)} = -\gamma$ for $l \neq i$ and $b_{init}^{(i)} = (m - i + 2)\gamma$.

Start by choosing β so that $|P(v^{(i)}|v^{(i-1)}, \dots, v^{(0)}) - \tilde{P}(v^{(i)}|v^{(i-1)}, \dots, v^{(0)})| < \epsilon_0$. This can be done for any $\epsilon_0 > 0$ by the argument in Theorem 1 Step 2. Now for time $l < m$, for any non-control node $h_j \notin H_{(i)}$, and all $h^{(l-1)}, \dots, h^{(0)}$, $\tilde{P}(h_j^{(l)} = 1|v^{(l-1)}, \dots, h^{(0)}) \leq \sigma \sum w_{i,j} v_i^{(l)} + b_j - \gamma$. This tends to 0 as $\gamma \rightarrow \infty$. So for any $\epsilon_1 > 0$, the argument in Theorem 1 Step 3 tells us we can choose γ large enough so that

$$|P(v^{(l)}|v^{(l-1)}, \dots, v^{(0)}) - \sum_{h^{(0)} \in H_{(i)}^{(0)}} \tilde{P}(v^{(l)}|v^{(l-1)}, \dots, v^{(0)})| < \epsilon_1. \quad (8)$$

Furthermore if $h_j \in H_{(i)}$ then

$$\tilde{P}(h_j^{(l)} = 1|v^{(l-1)}, \dots, h^{(0)})$$

$$= \sigma \left(\sum_i w_{i,j} v_i^{(l)} + b_j + (m - i + 2)\gamma - (m - i + 2)\gamma + \sum_i w_{i,j}^{(1)} h_i^{(l-1)} + \dots + \sum_i w_{i,j}^{(l)} h_i^{(0)} \right) \\ = \sigma \left(\sum_i w_{i,j} v_i^{(l)} + b_j + \sum_i w_{i,j}^{(1)} h_i^{(l-1)} + \dots + \sum_i w_{i,j}^{(l)} h_i^{(0)} \right).$$

So $\sum_{h^{(l)} \in H^{(l)}} \tilde{P}(v^{(l)}, h^{(l)} | v^{(l-1)}, \dots, v^{(0)})$ does not depend on γ . Using the same argument as in Step 3 of Theorem 1, for all $\epsilon_1 > 0$ there exists α_0 so that $\alpha > \alpha_0$ implies that

$$| \sum_{h^{(l)} \in H^{(l)}} \tilde{P}(v^{(l)}, h^{(l)} | v^{(l-1)}, \dots, v^{(0)}) - \sum_{h^{(l)} \in H_{\epsilon} \setminus H_{(0)}} \tilde{P}(v^{(l)}, h^{(l)} | v^{(l-1)}, \dots, v^{(0)}) | < \epsilon_1.$$

But the second term is just the probability of $v^{(l)}$ under the Boltzmann distribution of $H_{v^{(l-1)}, \dots, v^{(0)}}$, so using continuity of the KL-divergence along with the triangle inequality gives us the second and third condition for Lemma 1. Finally note that for $t \geq m$, if $h_j \in H^{(t)}$ for any l and all $h^{(l-1)}, \dots, h^{(l-m)}$, $P(h_j^{(t)} = 1 | v^{(t)}, h^{(t-1)}, \dots, h^{(l-m)}) \leq \sigma(\sum w_{i,j} v_i^{(t)} + b_j - \gamma)$. So for any ϵ_1 , we can take γ large enough such that

$$| \tilde{P}(v^{(t)} | v^{(t-1)}, \dots, v^{(0)}) - \sum_{h^{(l)} \in (H \setminus H_{\epsilon}) \setminus H_{(0)}} \tilde{P}(v^{(t)}, h^{(l)} | v^{(t-1)}, \dots, v^{(0)}) | < \epsilon_1. \quad (9)$$

Since for $t \geq m$, γ does not appear anywhere in the second term, this leaves us with the machine described in the first part of the proof, thus the first condition for Lemma 1 also holds. ■

3. Universal Approximation Results for the Recurrent Temporal Restricted Boltzmann Machines

The TRBM gives a nice way of using a Boltzmann Machine to define a probability distribution that captures time dependence in data, but it turns out to be difficult to train in practice (Sutskever et al., 2008). To fix this, a slight variation of the model, the RTRBM, was introduced. The key difference between the TRBM and the RTRBM is the use of deterministic real values denoted $h^{(t)}$. We will denote the probabilistic binary hidden units at time t by $h^{(t)}$. The distribution defined by an RTRBM, Q , is

$$Q(v^T, h^T) = \left(\prod_{k=1}^{T-1} Q(v^{(k)}, h^{(k)} | h^{(k-1)}) \right) Q_0(v^{(0)}, h^{(0)}).$$

Here $Q(v^{(t)}, h^{(t)} | h^{(t-1)})$ is defined as

$$Q(v^{(t)}, h^{(t)} | h^{(t-1)}) = \frac{\exp(v^{(t)\top} W h^{(t)} + c^\top v^{(t)} + b^\top h^{(t)} + h^{(t)\top} W' h^{(t-1)})}{Z(h^{(t-1)})},$$

and h^T is a sequence of real-valued vectors defined by

$$\begin{aligned} h^{(0)} &= \sigma(W v^{(0)} + W' h^{(0)} + b), \\ h^{(0)} &= \sigma(W v^{(0)} + b_{\text{init}} + b), \end{aligned} \quad (10)$$

where σ is the logistic function and b_{init} is an initial bias. Q_0 is once again an initial distribution defined as a Boltzmann Distribution with bias $b + b_{\text{init}}$. The difference between

the RTRBM and the TRBM is the use of the sequence of real valued vectors h^T for the temporal connections. At each time step each hidden node h_i takes on two values, a deterministic $h_i^{(t)}$ and a probabilistic $h_i^{(t)}$. The fact that the temporal parameters are calculated deterministically makes learning more tractable in these machines (Sutskever et al., 2008).

Theorem 3: *Let R be a distribution over a sequence of length T of binary vectors of length n that is time homogeneous and has finite time dependence. For any ϵ there exists an RTRBM, Q , such that $KL(R||Q) < \epsilon$.*

Proof: As in Theorem 2, for each configuration $v^{(t-1)}, \dots, v^{(l-m)}$, include hidden units $H_{v^{(t-1)}, \dots, v^{(l-m)}}$ with parameters so that the KL distance between the visible distribution of the Boltzmann machine given by $H_{v^{(t-1)}, \dots, v^{(l-m)}}$ with these parameters and the distribution $R_1(\cdot | v^{(t-1)}, \dots, v^{(l-m)})$ is less than ϵ' . Now for each possible visible configuration v add the control node $h_{c,v}$ with the same biases and visible-hidden weights as in Theorems 1 and 2 (determined entirely by parameter β). In Theorem 2, $h_{c,v}$ had i -step temporal connections from $h_{c,v}$ to every $H_{v^{(t-1)}, \dots, v^{(l-m)}}$ with $v^{(l-t)} \neq v$. The proof will proceed by showing that each of these i -step temporal connections can instead be given by a chain of nodes in the RTRBM. We wish to show that we can add i hidden units connecting $h_{c,v}$ to every hidden node in $H_{v^{(t-1)}, \dots, v^{(l-m)}}$ such that if $h_{c,v}$ is on at time $t-i$, it will have the same effect on the distribution of a node in $H_{v^{(t-1)}, \dots, v^{(l-m)}}$ as it does in the general TRBM, and the i additional hidden units do not effect the visible distribution. If we can achieve this then the same proof will hold for the RTRBM. This will be done as follows.

For each $h_{c,v}$ and each $1 \leq k < m$, add an additional hidden unit with 0 bias and no visible-hidden connections. For each $h_{c,v}$, label these $m-1$ hidden units $g_{(v,1)}, \dots, g_{(v,m-1)}$. Since these nodes have no visible-hidden connections they have no effect on the visible distribution at the current time step. For $1 < k < m-1$, let $w'_{(v,k),(v,k+1)} = 1$. Let $w'_{(c,v),(v,1)} = 1$ and $w'_{(v,k),j} = -\alpha$ if $h_j \in H_{v^{(t-1)}, \dots, v^{(t-k-1)}, \dots, v^{(l-m)}}$ with $v^{(t-k-1)} \neq v$, and $w'_{(v,k),j} = 0$ otherwise (see Fig. 2). Given a sequence $v^{(t-1)}, \dots, v^{(l-m)}$, consider the probability that $h_j^{(t)} = 1$ for some hidden unit $h_j \in H_{v^{(t-1)}, \dots, v^{(l-m)}}$ where $v^{(t-k)} \neq v^{(t-k)}$ for some k . Then by construction there is some hidden node g_{k-1} with $w'_{(v,k-1),j} = -\alpha$. The value of $g_{k-1}^{(t-1)}$ is calculated recursively by $g_{k-1}^{(t-1)} = \sigma(g_{k-2}^{(t-2)}) = \sigma^{k-1}(h_{c,v}^{(t-k)}) = \sigma^k(0.5\beta)$. Since k is bounded, by making β arbitrarily large we make g_{k-1} arbitrarily close to 1 and thus make $h_j^{(t)}$ $w'_{(v,k-1),j} g_{k-1}^{(t-1)}$ arbitrarily close to $-\alpha$.

Now suppose we have $h_j^{(t')} = 1$ for some hidden unit in $H_{v^{(t-1)}, \dots, v^{(l-m)}}$. Then every temporal connection is $-\alpha$, and $g_{(v,k)}^{(t-1)} = \sigma^k(-0.5\beta)$ for every $g_{(v,k)}$, so again by making β arbitrarily large we make the temporal terms arbitrarily close to 0. Thus as $\beta \rightarrow \infty$, the temporal terms from $h_{c,v}^{(t')}$ are either $-\alpha$ or 0 as needed.

We know from Theorem 2 that we can construct a TRBM with distribution P such that for $t \geq m$ and all $v^{(t)}$, $|P(v^{(t)} | v^{(t-1)}, \dots, v^{(0)}) - R(v^{(t)} | v^{(t-1)}, \dots, v^{(l-m)})| < \epsilon$ for any $\epsilon > 0$. The above argument shows that we can construct an RTRBM by replacing the connections $w_{(c,v),j}^{(t)}$ in the TRBM with the chain described above so that for any $\epsilon' > 0$

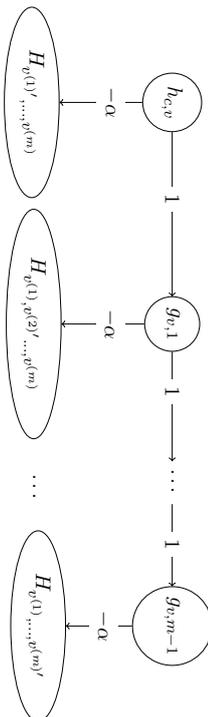


Figure 2: The temporal connections of a control node. Each $g_{c,i}$ connects to every $H_{v^{(i)}, \dots, v^{(m)}}$ with $v^{(i+1)} \neq v$ and $h_{c,v}$ connects to every $H_{v^{(1)}, v^{(2)}, \dots, v^{(m)}}$ with $v^{(1)} \neq v$.

there exist α_0, β_0 such that $\alpha > \alpha_0$ and $\beta > \beta_0$ imply that for all $v^{(t)}$ with $t \geq m$, $|\bar{Q}(v^{(t)}|v^{(t-1)}, \dots, v^{(0)}) - P(v^{(t)}|v^{(t-1)}, \dots, v^{(0)})| < \epsilon'$. Note that since the chains are of length m , Q only depends on the previous m visible configurations. Then, once again applying the triangle inequality and continuity of the KL-divergence we can satisfy the first condition of Lemma 1.

To finish the proof our machine must also be able to approximate the initial m distributions. Again this could be easily done should we choose to use an RBM to approximate the distribution of the first m time step. Instead we provide a construction to simulate the first m distributions in the RTRBM using the definition given above. As before we will use the construction of Theorem 2 and replace the long-term temporal connections with a chain. To begin, add each $H^{(i)}$ described in Theorem 2 with the temporal connections again replaced by the chains described in the above step. Now we just need to replace the m initial biases. First add $-\gamma$ to the bias of each node in $H^{(i)}$. Add hidden units l_0, \dots, l_{m-1} with connections between them $w^{(i,j)}(l_{i+1}) = \delta$ and biases $-\delta$ for l_0, l_1 and -0.5δ for l_2, \dots, l_{m-1} . For every $i > 0$, define the temporal connections to be 2γ from l_i to every node in $H^{(i)}$ and -2γ to every node in $H_{v^{(i-1)}, \dots, v^{(i-m)}}$ for all $v^{(i-1)}, \dots, v^{(i-m)}$. Now set the initial biases for every l_i to be 0 except for l_0 . Set this initial bias to be 2δ . Define the initial bias for every other non-control node to be $-\delta$ with the exception of $H^{(0)}$ whose initial bias is 0 (see Fig 3).

First we calculate the values $l_i^{(t)}$. Since l_0 has bias of $-\delta$ and initial bias of 2δ , we have $l_0^{(0)} = \sigma(\delta)$, and $l_1^{(1)} = \sigma(\delta\sigma(\delta) - \delta)$. Taking the limit as $\delta \rightarrow \infty$ we have $l_0^{(0)} = 1$ and

$$\lim_{\delta \rightarrow \infty} l_1^{(1)} = \lim_{\delta \rightarrow \infty} \sigma(\delta(\sigma(\delta) - 1)) = \lim_{\delta \rightarrow \infty} \sigma\left(\frac{-\delta}{1 + \exp(\delta)}\right) = \sigma(0) = 0.5.$$

Next we calculate the limit of $l_2^{(2)}$ as $\delta \rightarrow \infty$:

$$\lim_{\delta \rightarrow \infty} l_2^{(2)} = \lim_{\delta \rightarrow \infty} \sigma(\delta l_1^{(1)} - 0.5\delta) = \lim_{\delta \rightarrow \infty} \sigma\left(\frac{\delta}{l_1^{(1)} - 0.5}\right) = \lim_{\delta \rightarrow \infty} \sigma\left(\frac{-(l_1^{(1)} - 0.5)^2}{\frac{\delta}{l_1^{(1)}}}\right).$$

Note that $\frac{\delta}{l_1^{(1)}}$ is finite and non-zero, so evaluating the limit we get $\lim_{\delta \rightarrow \infty} l_2^{(2)} = \sigma(0) = 0.5$. Then by induction $l_j^{(j)} = 0.5$ for $i > 1$. Now we look at the case where $j \neq i$. For $j > 0$

we have $l_0^{(j)} = \sigma(-\delta)$, $l_1^{(j+1)} = \sigma(l_0^{(j)} - \delta)$ and $l_k^{(j+k)} = \sigma(l_{k-1}^{(j+k-1)} - 0.5\delta)$ for $k > 1$. So for $j > i$ we have in the limit that $l_j^{(j)} = 0$. For $j < i$, we know $l_{j-i}^{(0)} \leq (-0.5\delta)$ so $l_{j-i+1}^{(1)} = \sigma(l_{j-i}^{(0)} - 0.5\delta)$, etc., so that in the limit we have $l_j^{(j)} = 0$. We conclude that for any $\epsilon > 0$, there exists δ_0 such that $\delta > \delta_0$ implies that for all $i > 0$ and all $j \neq i$, $|l_0^{(j)} - 1| < \epsilon$, $|l_i^{(j)} - 0.5| < \epsilon$, and $|l_k^{(j)}| < \epsilon$.

When $l_0^{(0)} = 0$, $l_i^{(i)} = 0.5$, and $l_j^{(j)} = 0$, we have that for $t < m$, if $h_j \in H_{v^{(t-1)}, \dots, v^{(t-m)}}$ or $h_j \in H^{(i)}$ with $i \neq t$ then the bias is at most $b_j - \gamma$ and if $h_j \in H^{(i)}$ then the temporal connections from l_1, \dots, l_{m-1} are γ , which cancels the γ subtracted initially so the added bias is 0. For $t \geq m$ the temporal connections from l_i, \dots, l_{m-1} to all $H_{v^{(t-1)}, \dots, v^{(t-m)}}$ are 0 and the added bias to each node in $H^{(i)}$ is $-\gamma$. This is exactly the machine described in the second part of Theorem 2.

Putting this together, we first note that since each l_i has no visible connections we can ignore their binary values much in the same way that we can ignore the chains in the first part of the proof. Now for $t \neq 0$ and any $\epsilon > 0$ and any $v^{(t)}$, first we choose $\beta > \beta_0$ so that $|\bar{Q}(v^{(t)}|v^{(t-1)}, \dots, v^{(0)}) - \bar{Q}(v^{(t)}|v^{(t-1)}, \dots, v^{(0)})| < \epsilon'$, then we choose $\delta > \delta_0$ so that $|\bar{Q}(v^{(t)}|v^{(t-1)}, \dots, v^{(0)}) - \bar{Q}(v^{(t)}|v^{(t-1)}, \dots, v^{(0)})| < \epsilon'$ where \bar{Q} is the distribution obtained by replacing $l_0^{(0)}$ with 1, $l_i^{(i)}$ with 0.5 for $i > 0$, and $l_j^{(j)}$ with 0. As noted above this distribution is exactly the construction from the second part of Theorem 2. Finally, by the reasoning of Theorem 1 Step 4, by making δ large we make the initial distribution arbitrarily close to $H^{(0)}$ allowing us to approximate the distribution for the first time step. So the first, second and third conditions of the lemma are satisfied by the same argument used in Theorem 2. ■

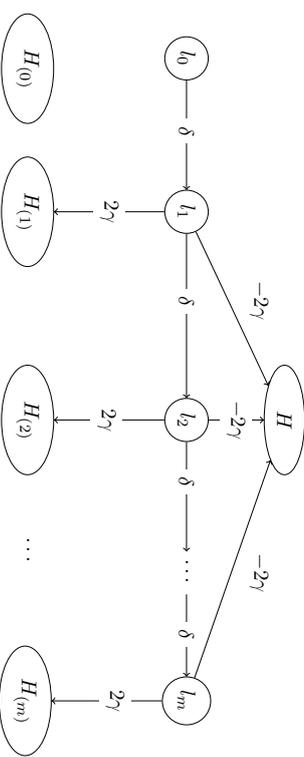


Figure 3: The temporal connections of the initial chain. H is the machine defined in the first part of the proof. Each connection from an l_i to H or $H^{(i)}$ only goes to non-control nodes in the set (control nodes include the chains $g_{v^{(i)}, \dots, v^{(i-m)}}$). The l_i 's have no visible connections.

4. Conclusion

The proofs above have shown that generalized TRBMs and RTRBMs both satisfy the same universal approximation condition. In the proof of universal approximation for the RTRBM we take the weights large enough so that the real-valued hidden sequence becomes approximately binary. This suggests that the same proof could be adapted to the basic TRBM. However, the TRBM seems to have difficulty modeling distributions with long term time dependence. After an RTRBM was trained on videos of bouncing balls the machine was able to model the movement correctly and the hidden units contained chains of length two as described in the above proof (Sutskever et al., 2008). On the other hand the TRBM did not have this structure and modeled the motion of balls as a random walk which is what one might expect for a machine that is unable to use velocity data by modeling two-step time dependencies. Given the likely equivalence in representational power of the two models, this discrepancy of results is best explained by the efficiency of the learning algorithm for the RTRBM in comparison to the TRBM.

At first glance the constructions used here seem quite inefficient. For Theorem 1 we require $2^n(G(n)+1)$ hidden nodes where G is the number of hidden nodes required to approximate an arbitrary distribution on n nodes with a Restricted Boltzmann Machine. It is important to note that the number of nodes required here, although large, depends only on the number of visible units and the process we wish to approximate, not the number of time steps for which we wish to approximate R . If the required number of hidden nodes had depended on the number of time steps then the TRBM and RTRBM would be essentially pointless as the RBM can do the same for any finite number of time steps. In contrast, the CRBM has a comparatively small lower bound on the number of hidden units required to approximate a set of conditional distributions (Montufar et al., 2014). Nonetheless, the above proofs are constructive and give only an upper bound on the required number of hidden units. Furthermore, we made no assumptions about $KL(R_1(\cdot|v^{(t-1)}))\|R_1(\cdot|v^{(t-1)'})$. Even if $v^{(t-1)}$ and $v^{(t-1)'}$ are similar vectors, the resulting distributions may be quite different, so to guarantee the result in full generality we could need a whole new set of hidden units to approximate R_1 for each pattern $v^{(t-1)}$. With this in mind, we might expect $2^n(G(n))$ to be a reasonable lower bound. In practice, similar vectors in the previous time step should produce similar distributions for the current time step. For example, looking at consecutive frames in video data, we expect that two similar frames at a certain time step will lead to similar frames in the next time step. To formalize this we could impose the restriction $KL(R_1(\cdot|v^{(t-1)}))\|R_1(\cdot|v^{(t-1)'}) < f(d(v^{(t-1)}, v^{(t-1)'})$, where f is a bounded function and d is a metric on $\{0, 1\}^n$. With this condition we could hope to find a more efficient TRBM to approximate R than the one given in the proof.

Without making this additional assumption, the most obvious way to increase efficiency is to obtain a better upper bound on G . We know that the bounds given by Le Roux et al. (2008) are not the lowest possible upper bounds for G (Montufar and Ay, 2011). In practice, multiple layers of RBMs are often stacked, leading to a Deep Belief Network. Several papers have investigated the universal approximation properties of Deep Belief Networks (Sutskever and Hinton, 2010)(Le Roux and Bengio, 2010)(Montufar and Ay, 2011). Re-

calling the constructions used in the previous proofs, by replacing the RBMs modeling the transition probabilities with Deep Belief Networks we end up with a column structure in which certain control nodes in a column send negative feedback to the other columns. This structure bears an interesting resemblance to the structure of the visual cortex (Goodhill and Carreira-Perpinán, 2006) suggesting that perhaps the two are computationally similar.

Acknowledgments

The authors thank NSERC and the University of Victoria for partial funding of this work, and anonymous reviewers for helpful comments.

Appendix A.

The following table lists notations used for the labels and states for the nodes in the previous proofs

H_0	a set of hidden nodes whose distribution approximates R_0
$H^{(t)}$	a set of hidden nodes used to approximate the distribution at time t
H_v	Hidden nodes whose distribution approximates $R_1(\cdot v^{(t-1)}) = v$
$h_{c,v}$	the control node corresponding to the configuration v of the visible units
H_c	the set of all control nodes
$H_{c,v}^{(t)}$	the set of configurations of the hidden nodes at time t with $h_{c,v}^{(t)} = 1$ and $h_{c,v'}^{(t)} = 0$
$\bar{H}_{c,v}^{(t)}$	the set of configurations of hidden nodes at time t not in $H_{c,v}^{(t)}$
$g_{c,v^{(t)}}$	the i^{th} node in a chain connecting $h_{c,v}$ to the visible nodes
l_i	the i^{th} node in the initial chain connecting l_0 with the rest of the H

Appendix B.

In this appendix we provide a proof for Lemma 1

Proof: For an arbitrary $\epsilon > 0$, we need to find a $P \in \mathbf{P}$ such that $KL(R\|P) < \epsilon$, where the KL-divergence is

$$KL(R\|P) = \sum_{v^T} R(v^T) \log \left(\frac{R(v^T)}{P(v^T)} \right).$$

We can write $P(v^T)$ as

$$\left(\prod_{t=1}^{T-1} P(v^{(t)}|v^{(t-1)}, \dots, v^{(0)}) \right) P(v^{(0)}),$$

and by assumption

$$R(v^T) = \left(\prod_{t=m}^{T-1} R_1(v^{(t)}|v^{(t-1)}, \dots, v^{(t-m)}) \right) \prod_{i=1}^{m-1} R_0(v^{(i)}|v^{(i-1)}, \dots, v^{(0)}).$$

Then expanding out the log in the KL-divergence gives us

$$\begin{aligned} KL(R\|P) &= \sum_{v^T} \sum_{t=m}^{T-1} R(v^T) \log \left(\frac{R_1(v^{(t)}|v^{(t-1)}, \dots, v^{(t-m)})}{P(v^{(t)}|v^{(t-1)}, \dots, v^{(0)})} \right) \\ &\quad + \sum_{v^T} \sum_{t=1}^{m-1} R(v^T) \log \left(\frac{R_0(v^{(t)}|v^{(t-1)}, \dots, v^{(0)})}{P(v^{(t)}|v^{(t-1)}, \dots, v^{(0)})} \right) + \sum_{v^T} R(v^T) \log \left(\frac{R_0(v^{(0)})}{P(v^{(0)})} \right). \end{aligned}$$

We can decompose $R(v^T)$ into $R(v^{(T-1)}, \dots, v^{(t)}|v^{(t-1)}, \dots, v^{(0)})R(v^{(t-1)}, \dots, v^{(0)})$ so for a given t we can write

$$\sum_{v^T} R(v^T) \log \left(\frac{R_1(v^{(t)}|v^{(t-1)}, \dots, v^{(t-m)})}{P(v^{(t)}|v^{(t-1)}, \dots, v^{(0)})} \right)$$

$$= \sum_{v^{(t-1)}, \dots, v^{(0)}} R(v^{(t-1)}, \dots, v^{(0)})$$

$$\begin{aligned} &\times \left(\sum_{v^{(t)}} \sum_{v^{(t-1)}, \dots, v^{(t+1)}} R(v^{(T-1)}, \dots, v^{(t)}|v^{(t-1)}, \dots, v^{(0)}) \log \left(\frac{R_1(v^{(t)}|v^{(t-1)}, \dots, v^{(t-m)})}{P(v^{(t)}|v^{(t-1)}, \dots, v^{(0)})} \right) \right) \\ &= \sum_{v^{(t-1)}, \dots, v^{(0)}} R(v^{(t-1)}, \dots, v^{(0)}) KL(R_1(\cdot|v^{(t-1)}, \dots, v^{(t-m)}) \| P_1(\cdot|v^{(t-1)}, \dots, v^{(0)})). \end{aligned}$$

Since $R(v^{(t-1)}, \dots, v^{(0)}) < 1$ for all $v^{(t-1)}, \dots, v^{(0)}$ we have

$$\begin{aligned} &\sum_{v^T} R(v^T) \log \left(\frac{R_1(v^{(t)}|v^{(t-1)}, \dots, v^{(t-m)})}{P(v^{(t)}|v^{(t-1)}, \dots, v^{(0)})} \right) \\ &\leq 2^{tn} \sum_{v^t} KL(R_1(\cdot|v^{(t-1)}, \dots, v^{(t-m)}) \| P_1(\cdot|v^{(t-1)}, \dots, v^{(0)})). \end{aligned}$$

The same logic applies for the cases with $t < m$.

By hypothesis there exists $P \in \mathbf{P}$ such that for every v^T and every ϵ' ,

$$\begin{aligned} KL(R_1(\cdot|v^{(t-1)}, \dots, v^{(t-m)}) \| P_1(\cdot|v^{(t-1)}, \dots, v^{(0)})) &< \epsilon' \text{ for } t \geq m, \\ KL(R_0, \delta(\cdot|v^{(t-1)}, \dots, v^{(0)})) \| P_1(\cdot|v^{(t-1)}, \dots, v^{(0)}) &< \epsilon' \text{ for every } 0 < t < m \text{ and} \\ KL(R_0 \| P_0) &< \epsilon'. \end{aligned}$$

This gives us

$$\begin{aligned} KL(R\|P) &\leq \sum_{t=m}^{T-1} 2^{tn} \sum_{v^t} KL(R_1(\cdot|v^{(t-1)}, \dots, v^{(t-m)}) \| P_1(\cdot|v^{(t-1)}, \dots, v^{(0)})) \\ &\quad + \sum_{t=1}^{m-1} 2^{tn} \sum_{v^t} KL(R_0(\cdot|v^{(t-1)}, \dots, v^{(0)})) \| P_1(\cdot|v^{(t-1)}, \dots, v^{(0)}) \\ &\quad + KL(R_0 \| P_0) \\ &< \sum_{t=m}^{T-1} 4^{tn} \epsilon' + \sum_{t=0}^{m-1} 4^{tn} \epsilon'. \end{aligned}$$

Then we merely choose an ϵ' so that this expression is less than ϵ and choose a corresponding $P \in \mathbf{P}$. \blacksquare

Notice in the proof that T was chosen arbitrarily and in the last line of the proof we see that decreasing T provides a tighter bound on the KL-divergence so any distribution which approximates R with a certain upper bound on the KL-divergence for T time steps will approximate R with at most the same upper bound on the KL-divergence for $t < T$ time steps.

References

Y. Freund and D. Haussler. Unsupervised learning of distributions of binary vectors using 2 layer networks. *Advances in Neural Information Processing Systems*, 4:912–919, 1991.

- G. Goodhill and M. Carreira-Perpinán. *Cortical Columns*. Macmillan, first edition, 2006.
- G. Hinton. Training a product of experts by minimizing contrastive divergence. *Neural Computation*, 14:1771–1800, 2002.
- Geoffrey E. Hinton and Simon Osindero. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1553, 2006.
- N. Le Roux and Y. Bengio. Representational power of restricted Boltzmann machines and deep belief networks. *Neural Computation*, 20:1631–1649, 2008.
- N. Le Roux and Y. Bengio. Deep belief networks are compact universal approximators. *Neural Computation*, 22:2192 – 2207, 2010.
- G. Montufar and N. Ay. Refinements of universal approximation results for deep belief networks and restricted Boltzmann machines. *Neural Computation*, 23:1306–1319, 2011.
- G. Montufar, N. Ay, and K. Ghazi-Zahedi. Geometry and expressive power of conditional restricted Boltzmann machines. *Journal of Machine Learning(Preprint)*, 2014. URL <http://arxiv.org/abs/1402.3346>.
- I. Sutskever and G. Hinton. Learning multilevel distributed representations for high-dimensional sequences. *Proceeding of the Eleventh International Conference on Artificial Intelligence and Statistics*, pages 544 – 551, 2006.
- I. Sutskever and G. Hinton. Deep, narrow sigmoid belief networks are universal approximators. *Neural Computation*, 20:2192 – 2207, 2010.
- I. Sutskever, G. Hinton, and G. Taylor. The recurrent temporal restricted Boltzmann machine. *Advances in Neural Information Processing Systems*, 21:1601–1608, 2008.
- G. Taylor, G. Hinton, and S. Roweis. Modeling human motion using binary latent variables. *Advances in Neural Information Processing Systems*, 19:1345–1352, 2006.
- L. Younes. Synchronous Boltzmann machines can be universal approximators. *Applied Mathematics Letters*, 9:109–113, 1996.

Exploration of the (Non-)Asymptotic Bias and Variance of Stochastic Gradient Langevin Dynamics

Sebastian J. Vollmer^{*}, Konstantinos C. Zygalakis[‡] and Yee Whye Teh[†]

¹Department of Statistics, University of Oxford, 24-29 St Giles', Oxford, OX1 3LB, UK

²School of Mathematics, University of Edinburgh, Edinburgh, EH9 3JZ, UK

Editor: Lawrence Carin

Abstract

Applying standard Markov chain Monte Carlo (MCMC) algorithms to large data sets is computationally infeasible. The recently proposed stochastic gradient Langevin dynamics (SGLD) method circumvents this problem in three ways: it generates proposed moves using only a subset of the data, it skips the Metropolis-Hastings accept-reject step, and it uses sequences of decreasing step sizes. In Teh et al. (2014), we provided the mathematical foundations for the decreasing step size SGLD, including consistency and a central limit theorem. However, in practice the SGLD is run for a relatively small number of iterations, and its step size is not decreased to zero. The present article investigates the behaviour of the SGLD with fixed step size. In particular, we characterise the asymptotic bias explicitly, along with its dependence on the step size and the variance of the stochastic gradient. On that basis a modified SGLD which removes the asymptotic bias due to the variance of the stochastic gradients up to first order in the step size is derived. Moreover, we are able to obtain bounds on the finite-time bias, variance and mean squared error (MSE). The theory is illustrated with a Gaussian toy model for which the bias and the MSE for the estimation of moments can be obtained explicitly. For this toy model we study the gain of the SGLD over the standard Euler method in the limit of large data sets.

Keywords: Markov Chain Monte Carlo, Langevin dynamics, big data, fixed step size.

1. Introduction

A standard approach to estimating expectations under a given target density $\pi(\theta)$ is to construct and simulate from Markov chains whose equilibrium distributions are designed to be π Brooks et al. (2011). A well-studied approach, for example in molecular dynamics Leimkuhler and Matthews (2013); Nawaf and Vanden-Eijnden (2010) and throughout Bayesian statistics Millstein and Tretyakov (2007); Neal (2011), is to use Markov chains constructed as numerical schemes which approximate the time dynamics of stochastic differential equations (SDEs). In this paper we will focus on the case of first order Langevin dynamics, which has the form

$$d\theta(t) = \frac{1}{2} \nabla \log \pi(\theta(t)) dt + dW_t, \quad (1)$$

^{*}. vollmer@stats.ox.ac.uk

[†]. kzygalak@ed.ac.uk

[‡]. y.w.teh@stats.ox.ac.uk

where $t \in \mathbb{R}_+$, $\theta \in \mathbb{R}^d$ and W_t is a d -dimensional standard Brownian motion. Under appropriate assumptions on $\pi(\theta)$, it is possible to show that the dynamics generated by Equation (1) are ergodic with respect to $\pi(\theta)$.

The simplest possible numerical scheme for approximating Equation (1) is the Euler-Maruyama method. Let $h > 0$ be a step size. Abusing notation, the diffusion $\theta(k \cdot h)$ at time $k \cdot h$ is approximated by θ_k , which is obtained using the following recursion equation

$$\theta_{k+1} = \theta_k + \frac{h}{2} \nabla \log \pi(\theta_k) + \sqrt{h} \xi_k, \quad (2)$$

where ξ_k is a standard Gaussian random variable on \mathbb{R}^d . One can use the numerical trajectories generated by this scheme for the construction of an empirical measure $\pi_h(\theta)$ either by averaging over one single long trajectory or by averaging over many realisations in order to obtain a finite ensemble average (see for example Milstein and Tretyakov (2007)). However, as discussed in Roberts and Tweedie (1996), one needs to be careful when doing this as it could be the case that the discrete Markov chain generated by Equation (2) is not ergodic. But even if the resulting Markov Chain is ergodic, $\pi_h(\theta)$ will not be equal to $\pi(\theta)$. Mattingly et al. (2010); Abdulle et al. (2014) which thus implies that the resulting sample average is biased. An alternative strategy that avoids this discretization bias and the ergodicity of the numerical procedure, is to use Equation (2) as a proposal for a Metropolis-Hastings (MH) MCMC algorithm Brooks et al. (2011), with an additional accept-reject step which corrects the discretization error.

In this paper we are interested in situations where π arises as the posterior in a Bayesian inference problem with prior density $\pi_0(\theta)$ and a large number $N \gg 1$ of i.i.d. observations X_i with likelihoods $\pi(X_i|\theta)$. In this case, we can write

$$\pi(\theta) \propto \pi_0(\theta) \prod_{i=1}^N \pi(X_i|\theta), \quad (3)$$

and we have the following gradient,

$$\nabla \log \pi(\theta) = \nabla \log \pi_0(\theta) + \sum_{i=1}^N \nabla \log \pi(X_i|\theta). \quad (4)$$

In these situations each update (2) has an impractically high computational cost of $\mathcal{O}(N)$ since it involves computations on all N items in the data set. Likewise, each MH accept-reject step is impractically expensive.

In contrast, the recently proposed stochastic gradient Langevin dynamics (SGLD) algorithm Welling and Teh (2011) circumvents this problem by generating proposals which are only based on a subset of the data, by skipping the accept-reject step and by using a decreasing step-size sequence (h_k) $_{k \geq 0}$. In particular one has

$$\theta_{k+1} = \theta_k + \frac{h_k}{2} \nabla \log \pi(\theta_k) + \sqrt{h_k} \xi_k, \quad (5)$$

$$\widehat{\nabla \log \pi}(\theta_k) = \nabla \log \pi_0(\theta_k) + \frac{1}{n} \sum_{i=1}^n \nabla \log \pi(X_{\tau_{n,i}}|\theta_k) \quad (6)$$

where ξ_k are independent standard Gaussian random variables on \mathbb{R}^d , and $\tau_k = (\tau_{k,1}, \dots, \tau_{k,n})$ is a random subset of $[N] := \{1, \dots, N\}$ of size n , generated, for example, by sampling with or without replacement from $[N]$. The idea behind this algorithm is that, since the stochastic gradient appearing in Equation (5) is an unbiased estimator of the true gradient $\nabla \log \pi(\theta)$, the additional perturbation due to the gradient stochasticity is of order h , smaller than the \sqrt{h} order of the injected noise, and so the limiting dynamics ($k \rightarrow \infty$) of Equation (5) should behave similarly to the case $n = N$. In Teh et al. (2014) it was shown that in this case that the K -step size weighted sample average is consistent and satisfies a CLT with rate depending on the decay of h_k . The optimal rate is limited to $K^{-\frac{1}{3}}$ and achieved by an asymptotic step size decay of $\asymp K^{-\frac{1}{3}}$.

The problem with decaying step sizes is that the efficiency of the algorithm slows the longer it is run for. A common practice for the SGLD and its extensions, the Stochastic Gradient Hamiltonian Monte Carlo Chen et al. (2014) and the Stochastic Gradient Thermostat Monte Carlo algorithm Ding et al. (2014), is to use step sizes that are only decreasing up to a point. The primary aim of this paper is to analyse the behaviour of SGLD with fixed step sizes of $h_k = h$. We provide two complementary analyses in this setting, one asymptotic in nature and one finite time. Let $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$ be a test function whose expectation we are interested in estimating. Using simulations of the dynamics governed by Equation (5), we can estimate the expectation using

$$\mathbb{E}_\pi[\phi(\theta)] \approx \frac{1}{K} \sum_{k=1}^K \phi(\theta_k) \quad (7)$$

for some large number of steps K . Our analyses shed light on the behaviour of this estimator.

In the first analysis, we are interested in the asymptotic bias of the estimator (7) as $K \rightarrow \infty$,

$$\lim_{K \rightarrow \infty} \frac{1}{K} \sum_{k=1}^K \phi(\theta_k) - \mathbb{E}_\pi[\phi(\theta)]. \quad (8)$$

Assuming for the moment that the dynamics governed by Equation (5) is ergodic, with invariant measure $\pi_h(\theta; n)$, the above asymptotic bias simply becomes $\mathbb{E}_{\pi_h(\cdot; n)}[\phi(\theta)] - \mathbb{E}_\pi[\phi(\theta)]$. In the case of Euler-Maruyama, where $n = N$ and the gradient is computed exactly, the asymptotic behaviour of the dynamics is well understood, in particular its asymptotic bias is $\mathcal{O}(h)$ Mattingly et al. (2010). When $n < N$, using the recent generalisations Abdulle et al. (2014); Sato and Nakagawa (2014) of the approach by Talay and Tubaro (1990) reviewed in Section 3, we are able to derive an expansion of the asymptotic bias in powers of the step size h . This allows us to explicitly identify the effect, on the leading order term in the asymptotic bias, of replacing the true gradient (4) with the unbiased estimator (6). In particular, we show in Section 4 that, relative to Euler-Maruyama, the leading term contains an additional factor related to the covariance of the subsampled gradient estimators (6).

Based on this result, in Section 4.2, we propose a modification of the SGLD (referred to simply as mSGLD) which has the same asymptotic bias as the Euler-Maruyama method up to first order in h . The mSGLD is given by

$$\theta_{k+1} = \theta_k + \frac{h}{2} \widehat{\nabla \log \pi}(\theta_k) + \sqrt{h} \left(I - \frac{h}{2} \text{Cov} \left[\widehat{\nabla \log \pi}(\theta_k) \right] \right) \xi; \quad (9)$$

where $\text{Cov} \left[\widehat{\nabla \log \pi}(\theta_k) \right]$ is the covariance of the gradient estimator. When the covariance is unknown, it can in turn be estimated by subsampling as well. This modification is different from the Stochastic Gradient Fisher Scoring S. Ahn and Welling (2012), a modification of the injected noise in order to better match the Bernstein von Mises posterior. In contrast, the mSGLD is a local modification based on the estimated variance of the stochastic gradient.

The second contribution provides a complementary finite time analysis. In the finite time case both the bias and the variance of the estimator are non-negligible, and our analysis accounts for both by focussing on bounding the mean squared error (MSE) of the estimator (7). Our results, presented in Section 5, show that,

$$\mathbb{E} \left[\left(\frac{1}{K} \sum_{k=0}^{K-1} \phi(\theta_k) - \pi(\phi) \right)^2 \right] \leq C(n) \left(h^2 + \frac{1}{Kh} \right), \quad (10)$$

where the RHS only depends on n through the constant $C(n)$, the h^2 term is a contribution of the (square of the) bias while the $1/Kh$ term is a contribution of the variance. We see that there is a bias-variance trade-off, with bias increasing and variance decreasing monotonically with h . Intuitively, with larger h the Markov chain can converge faster (lower variance) with the same number of steps, but this incurs higher discretization error. Our result is achieved by extending the work of Mattingly et al. (2010) from \mathbb{T}^d to \mathbb{R}^d . The main difficulty in achieving this relates to combining the results of Pardoux and Veretennikov (2001) and Teh et al. (2014), in order to establish the existence of nice, well controlled solutions to the corresponding Poisson equation Mattingly et al. (2010). We can minimise Equation (10) over h , finding that the minimizing h is on the order of $K^{-\frac{1}{3}}$, and yields an MSE of order $K^{-\frac{2}{3}}$. This agrees, surprisingly, with the scaling of $K^{-\frac{1}{3}}$ for the central limit theorem established for the case of decreasing step sizes, for the Euler-Maruyama scheme in Lambertson and Pages (2002) and for SGLD in Teh et al. (2014). This unexpected result, that the decreasing step size and fixed step size discretisations have, up to a constant, the same efficiency seems to be missing from the literature.

Our theoretical findings are confirmed by numerical simulations. More precisely, we start by studying a one dimensional Gaussian toy model both in terms of the asymptotic bias and the MSE of time averages in Section 2. The simplicity of this model allows us to obtain explicit expressions for these quantities and thus illustrate in a clear way the connection with the theory. More precisely, we confirm that the scaling of the step size and the number of steps for a prescribed MSE obtained from the upper bound in Equation (10) matches the scaling derived from the analytic expression for the MSE for this toy model. More importantly, this simplicity allows us to make significant analytic progress in the study of the asymptotic bias and MSE of time averages in the limit of large data sets $N \rightarrow \infty$. In particular, we are able to show that the SGLD reduces the computational complexity by one order of magnitude in N , for the estimation of the second moment in comparison with the Euler method if the error is quantified through the MSE.

In summary, this paper is organised as follows. We present our first explorations of the SGLD applied to a one-dimensional Gaussian toy model in Section 2. For this model we obtain an analytic characterisation of its bias and variance. This serves as intuition and benchmark for the two analyses developed in Sections 3 to 5. In Section 3 we review

some known results about the effect of the numerical discretisation of Equation (1) in terms of the finite time weak error as well as in terms of the invariant measure approximation. In Section 4 we apply these results to analyse the finite and long time properties of the SGLD, as well as to construct the modified SGLD algorithm which, asymptotically in h , behaves exactly as the Euler-Maruyama method ($n = N$) while still sub-sampling the data set at each step. Furthermore, in Section 5 we discuss the properties of the finite time sample averages, include its MSE. In Section 6 we revisit the Gaussian toy model to obtain a more precise understanding of the behaviour of SGLD in a large data and high accuracy regime. This is achieved using analytic expressions of the expectations of the sample average which are obtained using the Mathematics[®] notebook that is available upon request from the first author described in detail in Appendix 10. Finally, in Section 7 we demonstrate the observed performance of SGLD for a Bayesian logistic regression model which matches the theory, while, we conclude this paper in Section 8 with a discussion on some possible extensions of this work.

2. Exploring a one-dimensional Gaussian Toy Model

In this section, we develop results for a simple toy model, which will serve as a benchmark for the theory developed in Sections 3 to 5. In particular, we obtain analytic expressions for the bias and the variance of the sample average of the SGLD, allowing us to characterise its performance in detail.

We consider a one-dimensional linear Gaussian model,

$$\begin{aligned} \theta &\sim \mathcal{N}(0, \sigma_\theta^2), \\ X_i | \theta &\stackrel{i.i.d.}{\sim} \mathcal{N}(\theta, \sigma_x^2) \quad \text{for } i = 1, \dots, N. \end{aligned} \quad (11)$$

The posterior is given by

$$\pi = \mathcal{N}(\mu_p, \sigma_p^2) = \mathcal{N}\left(\frac{\sum_{i=1}^N X_i}{\frac{\sigma_x^2}{\sigma_\theta^2} + N}, \left(\frac{1}{\sigma_\theta^2} + \frac{N}{\sigma_x^2}\right)^{-1}\right). \quad (12)$$

For this choice of π , the Langevin diffusion (1) becomes,

$$d\theta(t) = -\frac{1}{2} \left(\frac{\theta(t) - \mu_p}{\sigma_p^2} \right) dt + dW_t, \quad (13)$$

and its numerical discretisation by the SGLD with step size h reads as follows,

$$\theta_{k+1} = (1 - Ah)\theta_k + B_k h + \sqrt{h}\xi_k, \quad (14)$$

where $\xi_k \stackrel{i.i.d.}{\sim} \mathcal{N}(0, 1)$ and

$$\begin{aligned} A &= \frac{1}{2} \left(\frac{1}{\sigma_\theta^2} + \frac{N}{\sigma_x^2} \right), \\ B_k &= \frac{N}{n} \sum_{j=1}^n X_{\tau_{k,j}} - \frac{2\sigma_p^2}{\sigma_x^2}, \end{aligned} \quad (15)$$

where $\tau_k = (\tau_{k,1}, \dots, \tau_{k,n})$ denote a random subset of $[N] = \{1, \dots, N\}$ generated, for example, by sampling with or without replacement from $[N]$, independently for each k . We note that the updates (14) will be stable only if $0 \leq 1 - Ah < 1$, that is, $0 < h < 1/A$. In the following we will also consider parameterising the step size as $h = r/A$ where $0 < r < 1$.

We denote $B = (B_k)_{k \geq 0}$. At the risk of obfuscating the notation, we will denote by $\text{Var}(B)$ the common variance of B_k for all k . For sampling with replacement, we have

$$\text{Var}(B) = \frac{1}{4\sigma_x^4} \frac{N}{n} \sum_{j=1}^n \left(X_j - \frac{1}{N} \sum_{i=1}^N X_i \right)^2 = \frac{1}{4\sigma_x^4} \frac{N(N-1)}{n} \text{Var}(X),$$

where $\text{Var}(X)$ is the typical unbiased empirical estimate of the variance of $\{X_1, \dots, X_N\}$. For sampling without replacement we have,

$$\text{Var}(B) = \frac{1}{4\sigma_x^4} \frac{N(N-n)}{n(N-1)} \sum_{i=1}^N \left(X_i - \frac{1}{N} \sum_{i=1}^N X_i \right)^2 = \frac{1}{4\sigma_x^4} \frac{N(N-n)}{n} \text{Var}(X). \quad (16)$$

2.1 Analysis of the Asymptotic Bias

We start by inspecting the estimate of the posterior mean. In particular, using Equation (14) and taking expectations with respect to ξ_k , we have

$$\mathbb{E}(\theta_{k+1}|B) = (1 - Ah)\mathbb{E}(\theta_k|B) + B_k h \quad (17)$$

which can be solved in order to obtain

$$\mathbb{E}(\theta_M|B) = (1 - Ah)^M \mathbb{E}(\theta_0) + \sum_{k=0}^{M-1} h(1 - Ah)^k B_{M-k-1}.$$

If we now take the expectation with respect to the random subsets B_k , using the fact that $\mathbb{E}(B_k) = \mathbb{E}(B)$ and take the limit of $M \rightarrow \infty$, we have

$$\mathbb{E}(\theta_\infty) = \sum_{k=0}^{\infty} (1 - Ah)^k h \mathbb{E}(B) = h/(1 - (1 - Ah)\mathbb{E}(B)) = \frac{\mathbb{E}(B)}{A} = \frac{\sum_{i=1}^N X_i}{\frac{\sigma_x^2}{\sigma_\theta^2} + N}.$$

We thus see that the SGLD is capturing the correct limiting mean of the posterior independently of the choice of the step size h . In other words, for the test function $\phi(\theta) = \theta$, the asymptotic bias is nil.

We now investigate the behaviour of the limiting variance under the SGLD. Starting with the law of total variance,

$$\text{Var}[\theta_{k+1}] = \mathbb{E}(\text{Var}[\theta_{k+1} | B]) + \text{Var}(\mathbb{E}[\theta_{k+1} | B]),$$

a simple calculation now shows that

$$\text{Var}[\theta_{k+1} | B] = (1 - Ah)^2 \text{Var}[\theta_k | B] + h$$

1. Note that the posterior variance is $\asymp 1/A$, so that steps of size $\asymp 1/A$ are $1/\sqrt{A}$ smaller than the width of the posterior. However the injected noise has variance $\asymp 1/A$ which matches the posterior variance.

and

$$\text{Var}(\mathbb{E}[\theta_{k+1} | B]) = (1 - Ah)^2 \text{Var}(\mathbb{E}[\theta_k | B]) + h^2 \text{Var}(B_k).$$

Combining these two results, we see that

$$\text{Var}(\theta_{k+1}) = (1 - Ah)^2 \text{Var}(\theta_k) + h + h^2 \text{Var}(B_k).$$

If we now take the limit of $k \rightarrow \infty$, we have that

$$\text{Var}(\theta_\infty) = \frac{1}{2A - A^2h} + \frac{h \text{Var}(B)}{2A - A^2h}. \quad (18)$$

where $\text{Var}(B)$ is the common value of $\text{Var}(B_k)$ for all $k \geq 0$. We note here that in the case of the Euler-Maruyama method (from here on we will simply refer to this as the Euler method) where $n = N$ and $\text{Var}(B) = 0$, only the first term remains. In other words, the first term is an (over-)estimate of the posterior variance $\sigma_p^2 = 1/2A$ obtained by the Euler-Maruyama discretisation at step size h . Our result here coincides with Zygalkakis (2011). On the other hand, the second term is an additional bias term due to the variability of the stochastic gradients. Further, using a Taylor expansion in h of the second summand, we see that the SGLD has an excess bias, relative to the Euler method, with first order term equal to

$$h \frac{\text{Var}(B)}{2A}. \quad (19)$$

Using the fact that $\text{Var}(\theta_\infty) = \mathbb{E}[\theta_\infty^2] - \mathbb{E}[\theta_\infty]^2$, and that the asymptotic bias of estimating $\mathbb{E}[\theta]$ is nil in this simple model, we see that the above gives the asymptotic biases of the Euler method and SGLD in the case of the test function $\phi(\theta) = \theta^2$.

We now consider the modified SGLD given in Equation (9) and to be discussed in Section 4.2. In this case the numerical discretisation of Equation (13) becomes

$$\theta_{k+1} = \theta_k - Ah\theta_k + B_k h + \sqrt{h} \left(1 - \frac{h}{2} \text{Var}(B)\right) \xi_k \quad (20)$$

A similar calculation as for the SGLD shows that

$$\begin{aligned} \mathbb{E}(\theta_\infty) &= \frac{\sum_{i=1}^N X_i}{\frac{\sigma_p^2}{\sigma_\theta^2} + N} \\ \text{Var}(\theta_\infty) &= \frac{1}{2A - A^2h} + \frac{h^2 \text{Var}^2(B)}{4(2A - A^2h)}. \end{aligned} \quad (21)$$

with the last term being the excess asymptotic bias. A Taylor expansion of the excess bias term shows that the term of order h vanishes and the leading term has order h^2 . Hence, for small h , the excess bias is negligible compared to the asymptotic bias of the Euler method, and we can say that, up to first order and in this simple example, the mSGLD has the same asymptotic bias as for the Euler method. In Section 4.2, we will show that these results hold more generally.

It is useful to visualise the above analytic results for the asymptotic biases of the Euler method, SGLD and mSGLD. In Figure 1 we show this for a data set of 1000 points drawn

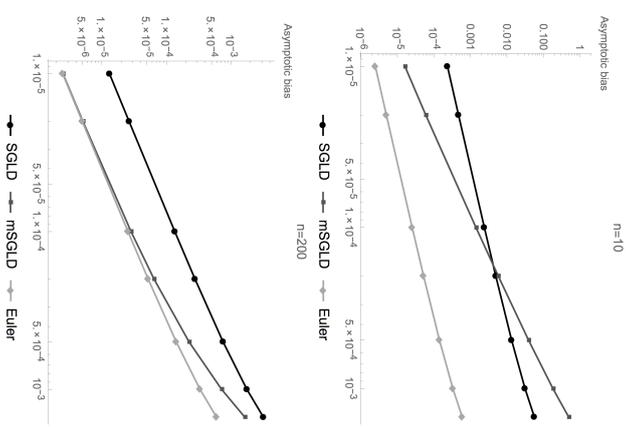


Figure 1: Comparison of the asymptotic biases for the SGLD, the mSGLD and the Euler method for the test function $\phi(\theta) = \theta^2$. For all simulations, we have used $N = 10^3$. We used $n = 10$ on the LHS and $n = 200$ on the RHS.

according to the model. In particular, after choosing the data set, then we calculate analytically $\text{Var}(B)$ for each choice of n using (16), and then use (18), (21) to evaluate the asymptotic bias for each of the methods. The first observation is that the Euler method has lowest asymptotic bias among all three methods (although of course it is also the most computationally expensive; see Section 6). We observe that if we choose $n = 10$ points for each gradient evaluation, for large values of the step size h , the SGLD is superior to the mSGLD. However, as h is reduced, this is no longer the case. Furthermore, if we use a more accurate gradient estimation with $n = 200$ data points, we see that mSGLD outperforms SGLD for all the step sizes used, but more importantly its asymptotic bias is now directly comparable with the Euler method where all the data points are used for evaluating the gradient.

2.2 Finite Time Analysis

In the previous subsection we analysed the behaviours of the three algorithms in terms of their biases in the asymptotic regime. In practice, we can only run our algorithms for a finite number of steps, say K , and it would be interesting to understand the behaviours of the algorithms in this scenario. With a finite number of samples, in addition to bias we also have to account for variance due to the Monte Carlo estimation.

A sensible analysis accounting for both bias and variance is to study the behaviour of the mean squared error (MSE), say in the Monte Carlo estimation for the second moment,

$$\text{MSE}_2 := \mathbb{E} \left(\frac{1}{K} \sum_{k=0}^{K-1} \theta_k^2 - (\mu_p^2 + \sigma_p^2) \right)^2. \quad (22)$$

We can expand the quadratic, and express MSE_2 as a linear combination of terms of the form $\mathbb{E}[\theta_k^p]$ for $p = 1, 2, 3, 4$. Each of terms can be calculated analytically, depending on the data set X , the total number of steps K , the subset size n , as well as the scaled step size parameter $r = hA$. We provide these calculations in Appendix 10.1 and a Mathematica® file in the supplementary materials.

In Figure 2 we visualise the behaviour of the resulting MSE_2 for a fixed data set with $N = 1000$ items, and with scaled step size $r = 1/20$. For the same number of steps M , the left figure shows that SGLD and mSGLD behaves similarly, decreasing initially then asymptoting at their asymptotic biases studied in the previous subsection. At $r = 1/20$ mSGLD has lower asymptotic biases than SGLD. Further, both MSE_2 's decrease with increasing subset size n , and are higher than that for the Euler method at $n = 1000$. Since SGLD and mSGLD computational costs per step are linear in n , the right figure instead plots the same MSE_2 's against the (effective) number of passes through the data set, that is, number of steps times n/N . This quantity is now proportional to the computational budget. Now we see that smaller subset sizes produce initial gains, but asymptote at higher biases.

These analytical results for a simple Gaussian model demonstrate the more general theory which forms the core contributions of this paper. Sections 3 and 4 develop a method to study the asymptotic bias as a Taylor expansion in h , while Section 5 provides a finite time analysis in terms of the mean squared error. Both analyses are based on the behaviour of the algorithms for small step sizes, and in this regime we see that mSGLD has better performance than SGLD. In Section 6 we will return to the simple Gaussian model to study the behaviour of the algorithms using different measures of performance and in different regimes. In particular, we will see that for larger step sizes SGLD has better performance than mSGLD.

3. Review of Weak Order Results

In this section we review some existing results regarding the ergodicity and accuracy of numerical approximations of SDEs. We start in Section 3.1 by introducing the framework and notation, the Fokker-Planck and backward Kolmogorov equations, and with some preliminary results on local weak errors of numerical one-step integrators. Section 3.2 presents assumptions necessary for ergodicity, and extends the results to a global error expansion of

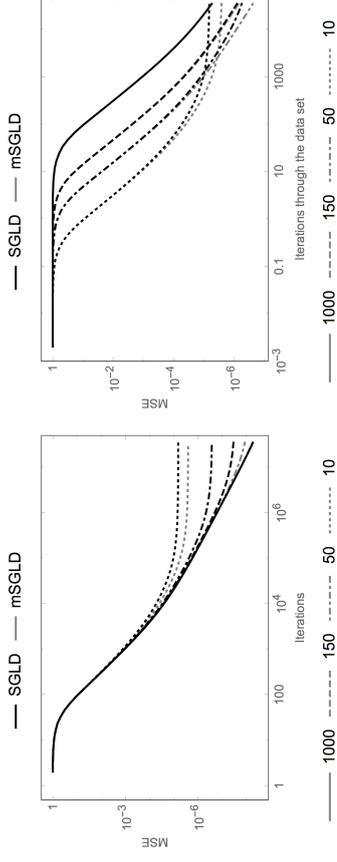


Figure 2: MSE₂ of the sample average for the SGLD and the mSGLD for the second moment of the posterior.

the weak error as well as the error in the approximation of the invariant measure. Finally, in Section 3.3 we apply our results to explicitly calculate the leading order error term of the numerical approximation of an Ornstein-Uhlenbeck solved by the Euler method.

3.1 One-step Numerical Approximations of Langevin Diffusions

Let us denote by $\rho(y, t)$ the probability density of $\theta(t)$ defined by the Langevin diffusion (1) with initial condition $\theta(0) = \theta$ and target density $\pi(y)$. Then $\rho(y, t)$ is the solution of the Fokker-Planck equation,

$$\frac{\partial \rho}{\partial t} = \mathcal{L}^* \rho, \quad (23)$$

with initial condition $\rho(y, 0) = \delta(y - \theta)$, a Dirac mass for the deterministic initial condition, and the operator \mathcal{L}^* given by

$$\mathcal{L}^* \rho = -\frac{1}{2} \nabla_{\theta} \cdot (\nabla \log \pi(\theta) \rho) + \frac{1}{2} \nabla_{\theta} \cdot \nabla_{\theta} \rho. \quad (24)$$

This operator is the L^2 -adjoint of the generator of the Markov process $(\theta(t))_{t \geq 0}$ given by (1),

$$\mathcal{L} = \frac{1}{2} \nabla_{\theta} \log \pi(\theta) \cdot \nabla_{\theta} + \frac{1}{2} \Delta_{\theta}, \quad (25)$$

Given a test function ϕ , define $u(\theta, t)$ to be the expectation,

$$u(\theta, t) = \mathbb{E}(\phi(\theta(t)) | \theta(0) = \theta), \quad (26)$$

with respect to the diffusion at time t when started with initial condition $\theta(0) = \theta$. We note that $u(\theta, t)$ is the solution of the backward Kolmogorov equation

$$\begin{aligned} \frac{\partial u}{\partial t} &= \mathcal{L}u, \\ u(\theta, 0) &= \phi(\theta). \end{aligned} \quad (27)$$

A formal Taylor series expansion for u in terms of the generator \mathcal{L} was derived in Zygalkakis (2011) and made rigorous by Debussche and Raou (2012) for the case where the state space is $\theta \in \mathbb{T}^d$. The Taylor series is of the following form,

$$u(\theta, h) = \phi(\theta) + \sum_{j=1}^l \frac{h^j}{j!} \mathcal{L}^j \phi(\theta) + h^{l+1} r_l(\theta), \quad (28)$$

for all positive integers l , with the remainder satisfying a bound of the form $|r_l(\theta)| \leq c_l(1 + |\theta|^{r_l})$ for some constants c_l, r_l depending on π and ϕ .

Remark 1 Another way to turn $u(\theta, h) = \phi(\theta) + h\mathcal{L}\phi + \frac{h^2}{2}\mathcal{L}^2\phi + \dots$ into a rigorous expansion, see Equation (28), is to follow the approach in (Taday and Tubaro, 1990, Lemma 2) and to assume that $\log \pi$ is C^∞ with bounded derivatives of any order (and this is the approach we follow here). This fact, together with the assumption that

$$|\phi(\theta)| \leq C(1 + |\theta|^s) \quad (29)$$

for some positive integer s is enough to prove that the solution u of Equation (27) has derivatives of any order that have a polynomial growth of the form of Equation (29), with other constants C, s that are independent of $t \in [0, T]$. In turn, these regularity bounds establish that Equation (28) holds. We mention here that the regularity conditions were relaxed in recent work in Kopec (2014) for the elliptic case and in Kopec (2015) for the hypoelliptic case.

Now assume that one solves Equation (1) numerically with a one step integrator, which we shall denote by,

$$\theta_{n+1} = \Psi(\theta_n, h, \xi_n), \quad (30)$$

where $\theta_0 = \theta(0)$, h denotes the step size, ξ_n are iid $\mathcal{N}(0, 1)$, and θ_n denotes the numerical approximation of $\theta(nh)$ for each $n \in \mathbb{N}$. For example in the case of the Euler method for equation (1) one has

$$\Psi(\theta, h, \xi) = \theta + \frac{h}{2} \nabla \log \pi(\theta) + \sqrt{h} \xi$$

Now, using this formulation we can define

$$U(\theta, h) = \mathbb{E}[\phi(\theta_1)|\theta_0 = \theta], \quad (31)$$

for the expectation of the test function after one step of the numerical integrator starting with the initial condition $\theta_0 = \theta$. We will make the following (easily satisfied) regularity and consistency assumptions about the integrator:

Assumption 2 We assume that the following hold:

- $\nabla \log \pi$ is C^∞ with bounded derivatives of all orders.
- For all deterministic initial conditions θ_0 , we have

$$|\mathbb{E}[\theta_1 - \theta_0]| \leq C(1 + |\theta_0|)h, \quad \text{and} \quad |\theta_1 - \theta_0| \leq M(1 + |\theta_0|)\sqrt{h}, \quad (32)$$

where C is a constant independent of h , for h small enough and M is a random variable that has bounded moments of all orders independent of h and θ_0 .

- Equation (31) has a weak Taylor series expansion of the form

$$U(\theta, h) = \phi(\theta) + hA_0(\pi)\phi(\theta) + h^2A_1(\pi)\phi(\theta) + \dots, \quad (33)$$

where $A_i(\pi)$, $i = 0, 1, 2, \dots$ are linear differential operators with coefficients depending smoothly on the drift function $\nabla \log \pi(\theta)$ and its derivatives (depending on the choice of the integrator).

- $A_0(\pi)$ coincides with the generator \mathcal{L} , in other words, the numerical method has weak order at least one.

Remark 3 Equation (33) holds for almost any Taylor based method applied to (1) but also to general SDEs with multiplicative noise. This is discussed further in Abulle et al. (2012), which also contains examples numerical methods, other than the Euler-Maryama method, for which (33) holds.

Assumptions 2 immediately imply the existence of a rigorous expansion

$$U(\theta, h) = \phi(\theta) + \sum_{i=0}^l h^{i+1} A_i(\pi)\phi(\theta) + h^{l+2} R_l(\theta) \quad (34)$$

for all positive integers l , with a remainder satisfying $|R_l(\theta)| \leq C_l(1 + |\theta|^{r_l})$ for some constants C_l, r_l . We say that the numerical solution has local weak order p if the first p terms in the expansion (33) of the numerical approximation agrees with that (28) for the exact diffusion. In this case, it is easy to see that the following local error formula holds,

$$\mathbb{E}[\phi(\theta(h))|\theta(0) = \theta] - \mathbb{E}[\phi(\theta_1)|\theta_0 = \theta] = h^{p+1} \left(\frac{\mathcal{L}^{p+1}}{(p+1)!} - A_p \right) \phi(\theta) + \mathcal{O}(h^{p+2}). \quad (35)$$

3.2 Global Weak Error Expansion

In this subsection, we will extend the local weak error expansion to a global one. Specifically, after M steps of the numerical integrator with step size h , we are interested in the difference between θ_M and the exact diffusion $\theta(T)$ where $T = Mh$, as evaluated by the difference between the corresponding expectations of ϕ ,

$$E(\phi, h, T) = \mathbb{E}[\phi(\theta(T))|\theta(0) = \theta] - \mathbb{E}[\phi(\theta_M)|\theta_0 = \theta], \quad (36)$$

In order for this study to make sense (when considering the limit $T \rightarrow \infty$), we will require that the SDE and its numerical approximation are both ergodic. We make standard assumptions in order for the Langevin diffusion $(\theta(t))_{t \geq 0}$ as given by (1) to be ergodic (see Hasminskii (1980)):

Assumption 4 We assume that the following hold for the Langevin diffusion $(\theta(t))_{t \geq 0}$:

- $\nabla \log \pi$ is C^∞ with bounded derivatives of all orders.
- there exists $\beta > 0$ and a compact set $K \subset \mathbb{R}^d$ such that $\forall \theta \in \mathbb{R}^d \setminus K$,

$$(\theta, \nabla \log \pi(\theta)) \leq -\beta \|\theta\|_2^2.$$

The question of the ergodicity of the numerical approximation (θ_n) is considerably more intricate in general. There exist cases where the underlying Langevin diffusion is ergodic, but its numerical approximation is not ergodic, or does not converge exponentially fast (Roberts and Tweedie (1996)). This relates mainly to the properties of the drift coefficient and its behaviour at infinity. For the Euler-Maruyama and the Milstein scheme this has been investigated in Talay and Tubaro (1990). In what follows we will simply assume that the Markov chain (θ_n) defined by the numerical approximation is indeed ergodic. Under this assumption the following theorem, which combines results derived by Talay and Tubaro (1990) and Milstein (1986), can be shown (see Abdulle et al. (2014) for a proof):

Theorem 5 Suppose that the state space is \mathbb{R}^d , that Assumptions 2 and 4 hold, and that the Markov chain $(\theta_n)_{n \geq 0}$ defined by the one step integrator (30) is ergodic. If the numerical approximation has local weak order p , that is, Equation (35) holds, then we have the following expansion of the global error (36), for all $\phi \in C_P^{2p+4}(\mathbb{R}^d, \mathbb{R})$,

$$E(\phi, h, T) = h^p \int_0^T \mathbb{E}(e(\theta(s), s)) ds + \mathcal{O}(h^{p+1}), \quad (37)$$

where $e(\theta, t)$ is given by

$$e(\theta, t) = \left(\frac{1}{(p+1)!} \mathcal{L}^{p+1} - A_p \right) v(\theta, t), \quad (38)$$

with $v(\theta, t) = \mathbb{E}(\phi(\theta(T)) | \theta(t) = \theta)$ satisfying

$$\begin{aligned} \frac{\partial v}{\partial t} &= -\mathcal{L}v, \\ v(\theta, T) &= \phi(\theta). \end{aligned} \quad (39)$$

The expression (37) was proved by Talay and Tubaro (1990) for specific methods (e.g. the Euler-Maruyama or the Milstein methods), while the general procedure to infer the global weak order from the local weak order is due to Milstein (1986) (see also (Milstein and Tretyakov, 2004, Chapter 2.2)). However, the formulation of the error function (38) here is in terms of the generator \mathcal{L} and the operators A_i in Assumption 2, and does not contain any time derivatives as in Talay and Tubaro (1990); Milstein (1986). This formulation will be particularly useful for obtaining our main results.

Using Theorem 5, one can obtain a similar expansion to that in Equation (37) for the difference between the true and the numerical ergodic averages:

Theorem 6 Suppose that Assumption 4 holds, that our numerical method with deterministic initial condition is ergodic and of weak order p , and that $\phi : \mathbb{R}^d \rightarrow \mathbb{R}$ is a smooth function satisfying Equation (29). Then,

$$\lim_{K \rightarrow \infty} \frac{1}{K} \sum_{n=0}^{K-1} \phi(\theta_n) - \int_{\mathbb{R}^d} \phi(y) \pi(y) dy = -\lambda_p h^p + \mathcal{O}(h^{p+1}) \quad (40)$$

where λ_p is defined as

$$\lambda_p = \int_{\mathbb{R}^d} \int_0^\infty \left(\frac{1}{(p+1)!} \mathcal{L}^{p+1} - A_p \right) u(y, t) \pi(y) dt dy, \quad (41)$$

and $u(y, t)$ satisfies Equation (27).

The proof is given in Abdulle et al. (2014), and is similar to that in (Talay and Tubaro, 1990, Theorem 4) with the main difference being that Equation (37) is used as the starting point, instead of the specific formula for the Euler-Maruyama method used in Talay and Tubaro (1990).

Theorem 6 provides an explicit expression for the leading order term of the asymptotic bias of the numerical method. It will thus be the key result in our analysis of the asymptotic behaviour of SGLD later. Intuitively Equation (41) says that if we want to calculate the error between the numerical and the true ergodic averages, we need to take into account the long time ($t \rightarrow \infty$) discrepancy between the true and the numerical solution given by

$$\left(\frac{1}{(p+1)!} \mathcal{L}^{p+1} - A_p \right) u(y, t),$$

and then average over all possible initial conditions y with respect to invariant measure $\pi(y)$.

3.3 An Illustrative Example

We illustrate the weak order results above in the case of the Euler-Maruyama scheme applied to the Ornstein-Uhlenbeck process. For ease of notation, let $f(x) = \frac{1}{2} \nabla \log \pi(\theta)$. The Euler-Maruyama update steps are,

$$\theta_{n+1} = \theta_n + hf(\theta_n) + \sqrt{h} \xi_n. \quad (42)$$

A straightforward calculation Zygalkakis (2011) yields that the differential operator A_1 in (33) is given by

$$A_1 \psi = \frac{1}{2} f^T \nabla^2 \psi + \frac{1}{2} \sum_{i=1}^d \psi^{(i)}(e_i, e_i, f) + \frac{1}{8} \sum_{i,j=1}^d \psi^{(4)}(e_i, e_i, e_j, e_j) \quad (43)$$

where e_1, \dots, e_d denotes the canonical basis of \mathbb{R}^d and $\psi^{(i)}(\cdot, \cdot, \cdot)$ and $\psi^{(4)}(\cdot, \cdot, \cdot, \cdot)$, are the derivatives of ψ , which are trilinear and quadrilinear forms, respectively. In dimension $d = 1$, it reduces to

$$A_1 \psi = \frac{1}{2} f^2 \psi'' + \frac{1}{2} \psi''' + \frac{1}{8} \psi^{(4)}$$

In the case where $\theta \in \mathbb{R}$ and $\pi(\theta) = e^{-(\theta-\mu)^2/2\sigma^2}$, the Langevin diffusion (1) corresponds to the one dimensional Ornstein-Uhlenbeck process,

$$d\theta(t) = -\frac{1}{2} \left(\frac{\theta(t) - \mu}{\sigma^2} \right) dt + dW_t \quad (44)$$

For the test function $\phi(\theta) = \theta^2$, a simple calculation reveals that the solution of Equation (27) is

$$u(\theta, t) = \sigma^2 \left(1 - e^{-t/\sigma^2} \right) + \theta^2 e^{-t/\sigma^2} + \frac{\mu\theta}{\sigma^2} \left(1 - e^{-t/2\sigma^2} \right) e^{-t/2\sigma^2} + \frac{\mu^2}{4\sigma^4} \left(1 - e^{-t/2\sigma^2} \right)^2. \quad (45)$$

The Euler-Maruyama scheme has weak order $p = 1$, and (see Zygalakis (2011) for more details),

$$\frac{1}{2} \mathcal{L}^2 - A_1 = \frac{1}{8\sigma^4} (\theta - \mu) \frac{d}{d\theta} - \frac{1}{4\sigma^2} \frac{d^2}{d\theta^2}.$$

Using this together with Equation (45) for $u(\theta, t)$, we find

$$\left(\frac{1}{2} \mathcal{L}^2 - A_1 \right) u(\theta, t) = \frac{e^{-t/\sigma^2}}{2\sigma^2} - \frac{e^{-t/2\sigma^2}}{4\sigma^4} (\mu - \theta) \left(\left(1 - e^{-t/2\sigma^2} \right) \mu + e^{-t/2\sigma^2} \theta \right)$$

Formula (41) now gives

$$\begin{aligned} \lambda_1 &= - \int_0^\infty \int_{-\infty}^{+\infty} \left(\frac{e^{-t/\sigma^2}}{2\sigma^2} - \frac{e^{-t/2\sigma^2}}{4\sigma^4} (\mu - \theta) \left(\left(1 - e^{-t/2\sigma^2} \right) \mu + e^{-t/2\sigma^2} \theta \right) \right) \frac{e^{-(\theta-\mu)^2/2\sigma^2}}{\sqrt{2\pi}\sigma} d\theta dt \\ &= \frac{1}{4}. \end{aligned} \quad (46)$$

This is in agreement with known results on the stationary distribution of the Euler-Maruyama approximation to the Ornstein-Uhlenbeck process, see Zygalakis (2011):

$$\pi_h \sim N(\mu, \sigma_h^2) \quad \text{where} \quad \sigma_h^2 = \frac{\sigma^2}{1 - \frac{h}{4}\sigma^{-2}} = \sigma^2 + \frac{1}{4}h + \mathcal{O}(h^2).$$

4. Weak Convergence Analysis

We study the weak convergence properties of the SGLD method in the light of Theorems 5 and 6. The analysis in Section 4.1 implies that at leading order there is a cost associated with not calculating the likelihood over all points. Thus, we introduce in Section 4.2 a modification of the original algorithm which has an error that is asymptotically in h , identical to the error of the Euler method, when all data points are taken into account in the calculation of each likelihood gradient.

4.1 Stochastic Gradient Langevin Dynamics

Theorems 5 and 6 imply that in order to characterise the leading order error term both for the weak convergence and the invariant measure, we need to calculate the corresponding

differential operators A_0, A_1, \dots in Equation (33). To simplify the presentation and to illustrate the main ideas, we present the calculations only in the case where $\theta(t)$ is one dimensional. We start our calculations by rewriting the SGLD method in the following form

$$\theta_{j+1} = \theta_j + h \hat{f}_j(\theta_j) + \sqrt{h} \xi_j, \quad (47)$$

where

$$\hat{f}_j(\theta) = \frac{1}{2} \left(\nabla \log \pi_0(\theta) + \frac{1}{n} \sum_{i=1}^n \nabla \log \pi(X_{\tau_{j+1}}^i | \theta) \right),$$

τ_j is the subset (possibly with repetition) chosen at step j and,

$$\mathbb{E}_{\tau_j} \hat{f}_j(\theta) = f(\theta) := \frac{1}{2} \nabla \log \pi(\theta), \quad \forall n \leq N. \quad (48)$$

Expanding $\phi(\theta_{j+1})$ in powers of h and then taking expectations with respect to the injected random noise ξ_j ,

$$\begin{aligned} \mathbb{E}_{\xi_j} (\phi(\theta_{j+1}) | \theta_j) &= \phi(\theta_j) + h \left(\hat{f}_j(\theta_j) \phi'(\theta_j) + \frac{1}{2} \phi''(\theta_j) \right) \\ &\quad + \frac{h^2}{2} \left(\hat{f}_j^2(\theta_j) \phi''(\theta_j) + \hat{f}_j(\theta_j) \phi^{(3)}(\theta_j) + \frac{1}{4} \phi^{(4)}(\theta_j) \right) + \mathcal{O}(h^3). \end{aligned}$$

If we now take expectations with respect to τ_j ,

$$\mathbb{E} \phi(\theta_{j+1}) | \theta_j = \phi(\theta_j) + h \mathcal{L} \phi(\theta_j) + \frac{h^2}{2} \left(\mathbb{E}_{\tau_j} (\hat{f}_j^2(\theta_j)) \phi''(\theta_j) + f'(\theta_j) \phi^{(3)}(\theta_j) + \frac{1}{4} \phi^{(4)}(\theta_j) \right) + \mathcal{O}(h^3), \quad (49)$$

where \mathcal{L} is the generator (25) of Equation (1). We thus see that the SGLD method is a first order weak method and, dropping the indexing by j for notational convenience from now on,

$$A_1(\pi) \phi = \frac{1}{2} \left(\mathbb{E}_\pi (\hat{f}^2(\theta)) \phi'' + f'(\theta) \phi^{(3)} + \frac{1}{4} \phi^{(4)} \right).$$

The asymptotic bias in Equation (41) has an expansion based on the differential operator,

$$\begin{aligned} \frac{1}{2} \mathcal{L}^2 - A_1 &= \frac{1}{2} \left(f(\theta) f'(\theta) + \frac{1}{2} f''(\theta) \right) \frac{d}{d\theta} + \frac{1}{2} \left(f'(\theta) + f^2(\theta) - \mathbb{E}_\pi (\hat{f}^2(\theta)) \right) \frac{d^2}{d\theta^2} \\ &= \frac{1}{2} \left(f(\theta) f'(\theta) + \frac{1}{2} f''(\theta) \right) \frac{d}{d\theta} + \frac{1}{2} \left(f'(\theta) - \text{Var} \hat{f}(\theta) \right) \frac{d^2}{d\theta^2} \end{aligned} \quad (50)$$

We thus see that in the case of SGLD the leading order error term contains an extra factor of $-\frac{1}{2} \text{Var} \hat{f}(\theta) \nabla^2$ when compared to the Euler method ($n = N$), in which case $\text{Var} \hat{f}(\theta) = 0$. This can be understood as the penalty associated with not using all the available points for calculating the likelihood at every time step. It results in an extra term in the corresponding error expressions given in Theorems 5 and 6 when compared with the Euler method. More precisely, for $n \ll N$ the term $-\frac{1}{2} \text{Var}(\hat{f}(\theta))$ is of size $\mathcal{O}(N^{-2})$ thus making the leading order error term $\mathcal{O}(hN^{-2})$ in Equation (37).

Example 1 We illustrate the above findings on the toy model discussed in Section 2. In particular, using the expression for $u(\theta, t)$ from Section 3.3 (replacing μ and σ by μ_p and σ_p respectively), and that $\text{Var}(\hat{f}(\theta)) = \text{Var}(B)$ for this simple model, the extra term in Equation (50) when compared with the Euler method is now given by

$$-\frac{1}{2} \text{Var}(B) \partial_\theta^2 u(\theta, t) = -\text{Var}(B) e^{-t/\sigma_p}.$$

A simple integration of this term according to the formula (41) gives that the overall contribution of the extra term, which is,

$$\sigma_p \text{Var}(B) = \frac{\text{Var}(B)}{2A},$$

and thus agreeing with Equation (19) derived in Section 2.1.

4.2 Modified SGLD

As we have seen in the previous section, the SGLD method introduces an extra term $-\frac{1}{2} \text{Var}(f(\theta)) \nabla_\theta^2$ in the leading order error term related to the weak error (Theorem 5) and to the ergodic averages (Theorem 6). When $n \ll N$, this term is of order $\mathcal{O}(hN^2)$. In this section we will explore a modification of SGLD (mSGLD) for which this term is removed, so that the leading order term is exactly the same as for the Euler-Maruyama scheme. Specifically, the mSGLD updates are,

$$\theta_{j+1} = \theta_j + h\hat{f}(\theta_j) + \sqrt{h} \left(1 - \frac{h}{2} \text{Var}(\hat{f}(\theta_j)) \right) \xi_j. \quad (51)$$

We can again derive the weak order expansion as in the previous subsection. Our first step is to expand $\phi(\theta_{j+1})$ in powers of h and then take expectations with respect to the random variable ξ_j . In particular, we obtain

$$\begin{aligned} \mathbb{E}_{\xi_j}(\phi(\theta_{j+1})) &= \phi(\theta_j) + h \left(\hat{f}_j(\theta_j) \phi'(\theta_j) + \frac{1}{2} \phi''(\theta_j) \right) \\ &\quad + \frac{h^2}{2} \left(\hat{f}_j^2(\theta_j) - \text{Var}(\hat{f}(\theta_j)) \right) \phi''(\theta_j) + \hat{f}_j(\theta_j) \phi^{(3)}(\theta_j) + \frac{1}{4} \phi^{(4)}(\theta_j) + \mathcal{O}(h^3). \end{aligned}$$

Taking expectations with respect to the random sampling and using Equation (48), we obtain

$$\begin{aligned} \mathbb{E}(\phi(\theta_{j+1})) &= \phi(\theta_j) + h\mathcal{L}\phi(\theta_j) \\ &\quad + \frac{h^2}{2} \left(\mathbb{E}_{\tau_j}(\hat{f}_j^2(\theta_j)) - \text{Var}(\hat{f}(\theta_j)) \right) \phi''(\theta_j) + f(\theta_j) \phi^{(3)}(\theta_j) + \frac{1}{4} \phi^{(4)}(\theta_j) + \mathcal{O}(h^3), \end{aligned} \quad (52)$$

where \mathcal{L} is the generator of Equation (1). We thus see that the mSGLD is a first order weak method and

$$A_1(\pi)\phi = \frac{1}{2} \left(\mathbb{E}_{\tau_j}(\hat{f}_j^2(\theta)) - \text{Var}(\hat{f}(\theta)) \right) \phi'' + f(\theta) \phi^{(3)} + \frac{1}{4} \phi^{(4)}$$

Using the expression for \mathcal{L}^2 as in the case of SGLD, we have that,

$$\begin{aligned} \frac{1}{2} \mathcal{L}^2 - A_1 &= \frac{1}{2} \left(f(\theta) f'(\theta) + \frac{1}{2} f''(\theta) \right) \frac{d}{d\theta} + \frac{1}{2} \left(f'(\theta) + f^2(\theta) - \mathbb{E}_{\tau_j}(f^2(\theta)) + \text{Var}(\hat{f}(\theta)) \right) \frac{d^2}{d\theta^2} \\ &= \frac{1}{2} \left(f(\theta) f'(\theta) + \frac{1}{2} f''(\theta) \right) \frac{d}{d\theta} + \frac{1}{2} f'(\theta) \frac{d^2}{d\theta^2} \end{aligned} \quad (53)$$

We see that the leading order term in the weak error and the error for the ergodic averages is the same as for the Euler method, which uses all data at every step. In higher dimensions, a similar calculation gives the mSGLD updates,

$$\theta_{j+1} = \theta_j + h\hat{f}_j(\theta_j) + \sqrt{h} \left(I - \frac{h}{2} \text{Cov}(\hat{f}(\theta_j)) \right) \xi_j \quad (54)$$

where

$$\text{Cov}(\hat{f}(\theta)) = \mathbb{E} \left[\left(\hat{f}(\theta) - \mathbb{E}(\hat{f}(\theta)) \right) \left(\hat{f}(\theta) - \mathbb{E}(\hat{f}(\theta)) \right)^\top \right]$$

and ξ_j is a d -dimensional standard normal random variable.

Remark 7 Except for special cases, $\text{Var}(\hat{f}(\theta_j))$ does not have a closed form. The simplest possible way to proceed without it is to replace it by an unbiased estimator, for example in case of sampling without replacement,

$$\widehat{\text{Var}}(\hat{f}(\theta)) := \frac{N(N-n)}{n(n-1)} \sum_{i=1}^n \left(\nabla \log \pi(x_{\tau_i} | \theta) - \frac{\hat{f}_j(\theta)}{N} \right)^2.$$

This replacement does not change Equation (53) because the smallest order contribution to Equation (49) is of the form

$$-h^2 \mathbb{E} \left[\widehat{\text{Var}}(\hat{f}(\theta_j)) \xi_j^2 \right] = -h^2 \text{Var}(\hat{f}(\theta_j)).$$

However, estimating the variance of the stochastic gradient will affect higher order terms in h . For fixed h these terms may have larger contribution to the overall error depending on the choice of n and N . In fact, this is true even if we use the exact variance for the toy model in Section 2.1. More precisely, we compare the bias of the mSGLD and the SGLD in Equation (70) notice that h^2 term might be larger depending on the choice of n and N .

5. Finite Time Sample Averages

Having focused on the SGLD in the asymptotic regime, we will now provide non-asymptotic analysis of the mean squared error (MSE) of the finite time sample averages of the SGLD. In particular, we will decompose the MSE into bias and variance. The main result of this section will be of the form

$$\begin{aligned} \text{Bias:} & \quad \left| \mathbb{E} \frac{1}{K} \sum_{i=0}^{K-1} \phi(\theta_i) - \int \phi(x) \pi(x) dx \right| = \mathcal{O} \left(\frac{1}{h + \frac{1}{Kh}} \right) \\ \text{MSE:} & \quad \mathbb{E} \left(\frac{1}{K} \sum_{i=0}^{K-1} \phi(\theta_i) - \int \phi(x) \pi(x) dx \right)^2 = \mathcal{O} \left(h^2 + \frac{1}{Kh} \right) \end{aligned} \quad (55)$$

Remark 8 In Teh et al. (2014), a central limit theorem was provided for the decreasing step size SGLD which shows a convergence rate of $O(K^{-\frac{5}{3}})$. At first sight, the bound in Equation (55) seems better because of the $\frac{1}{\sqrt{K}}$ term in the upper bound. However, due to the bias, an additional term of order $O(h^2)$ appears. In order to compare (55) with the previous result of Teh et al. (2014), we optimise the sum of both terms over the step size h . This results in a bound on the MSE of the SGLD of order $O(K^{-\frac{5}{3}})$ and agrees with the rate achieved by the decreasing step size SGLD. This agreement between decreasing step size discretisation and fixed step size discretisation is, to our knowledge, not a widely-known observation in the literature. In contrast, for standard MCMC algorithms the MSE is bounded by order $O(K^{-1})$ due to the Metropolis-Hastings correction that removes the bias. Nevertheless, experimental results in the literature demonstrate that the SGLD might be advantageous in the initial transient phase of learning, see e.g. Patterson and Teh (2013); Chen et al. (2014)

In Section 5.2 we will focus on establishing the bound in Equation (55) which is an extension of the work by Mattingly et al. (2010). The authors obtained similar results for finite time sample averages of discretisations of diffusions of the form

$$d\theta_t = f(\theta_t) + g(\theta_t) dW_t \quad (56)$$

on the torus which we review subsequently in Section 5.1.

5.1 Preliminaries on the Poisson Equation and Time Averages

In the following a connection between time averages of the diffusion and the corresponding Poisson equation will be presented. For a more elaborate description of this technique we point the reader to Section 4.2 of Mattingly et al. (2010) and references therein.

The Poisson equation is an elliptic PDE on the basis of the generator associated with Equation (56). The generator of Equation (56) is

$$\mathcal{L}\psi = \nabla\psi \cdot \nabla f + \frac{1}{2}g(\theta)^\top \nabla^2\psi g(\theta),$$

while the Poisson equation is given by

$$\mathcal{L}\psi = \phi - \bar{\phi} \quad \text{on } \mathbb{R}^d \quad (57)$$

where ϕ is a test function and $\bar{\phi} := \int \phi(x)\pi(dx)$ with π being the invariant distribution of (56). For applications in Bayesian statistics π represents the posterior and the quantity $\bar{\phi}$ the posterior expectation of interest. The posterior expectation ϕ is estimated by the time average $\frac{1}{T} \int_0^T \phi(\theta(s)) ds$ of the Langevin dynamics. The difference between the two can be expressed explicitly by using Itô's formula on the solution ψ of the Poisson equation

$$\begin{aligned} \psi(\theta(t)) - \psi(\theta(0)) &= \int_0^t \phi(\theta(s)) - \bar{\phi} ds + \int_0^t \nabla\psi(\theta(s)) \cdot g(\theta(s)) dW_s, \\ \frac{1}{T} \int_0^T \phi(\theta(s)) ds - \bar{\phi} &= \frac{1}{T} (\psi(\theta(T)) - \psi(\theta(0))) - \frac{1}{T} \int_0^T \nabla\psi(\theta(s)) \cdot g(\theta(s)) dW_s. \end{aligned}$$

If the first term and the variance of the second term (the martingale term) on the right hand side can be bounded, an error bound for the time average is obtained.

In this article, we are interested in the time average of the Euler discretisation and the SGLD. We can build on the ideas of Section 5 in Mattingly et al. (2010) which considers time discretisations of Equation (56) of the following form

$$\theta_{k+1} = \theta_k + h f(\theta_k, h) + \sqrt{hg}(\theta_k, h) \eta_k, \quad \eta_k \sim \mathcal{N}(0, I).$$

In Mattingly et al. (2010) a Taylor expansion is used to express

$$\Delta\psi(\theta_{k+1}) := \psi(\theta_{k+1}) - \psi(\theta_k) = h(A_0\psi)'(\theta_k) + R_k$$

where R_k is the remainder term. The term A_0 was introduced in Equation (33) in Section 3.1.

Using that $\mathcal{L}\psi = \phi - \bar{\phi}$, summing over k and dividing by hK yields

$$\hat{\phi}_K := \frac{1}{K} \sum_{k=0}^{K-1} \phi(\theta_k) = \bar{\phi} + \frac{1}{Kh} (\psi(\theta_K) - \psi(\theta_0)) - \frac{1}{hK} \sum_{k=0}^{K-1} h(A_0 - \mathcal{L})\psi(\theta_k) - \frac{1}{Kh} \sum_{k=0}^{K-1} R_k.$$

Controlling $A_0 - \mathcal{L}$ and the remainder gives rise to Theorem 5.1 and 5.2 in Mattingly et al. (2010) stating that

$$\begin{aligned} \left| \mathbb{E} \hat{\phi}_K - \bar{\phi} \right| &\leq C \left(h + \frac{1}{h \cdot K} \right) \\ \mathbb{E} \left(\hat{\phi}_K - \bar{\phi} \right)^2 &\leq C \left(h^2 + \frac{1}{h \cdot K} \right). \end{aligned} \quad (58)$$

In particular, these results were derived for discretisations of SDEs on the torus. This simplifies the presentation because the derivatives of ψ are bounded on a compact set. However, the same arguments hold if the following assumption is imposed instead

$$\sup_k \mathbb{E} \left\| \psi^{(i)}(\theta_k) \right\| < \infty \quad \text{for } i = 1, \dots, 4, \quad (59)$$

verifying this condition will allow us to work on \mathbb{R}^d .

5.2 The Bias and the MSE of Finite Time SGLD Averages

We consider the SDE

$$d\theta_t = f(\theta_t) dt + g(\theta_t) dW_t, \quad (60)$$

with $g = I$ being the identity matrix but we keep g in order to make the presentation clearer. Based on this setup the recursion of the corresponding SGLD reads as follows

$$\Delta_{k+1} = \theta_{k+1} - \theta_k = \hat{f}_k h + h^{\frac{1}{2}} g_k \xi_{k+1}$$

where \hat{f}_k is an unbiased estimate of f . The focus of this section is to establish results similar to Equation (58) for the SGLD. They will be formulated in Theorem 9.

For the readability of the subsequent calculation we use the following notations

$$\Delta_{k+1} = \theta_{k+1} - \theta_k, \quad \phi_k = \phi(\theta_k),$$

$\hat{f}_k = \hat{f}(\theta_k, \tau_k, h)$ for the estimate of the drift, $g_k = g(\theta_k, h) = I$, $\psi_k = \psi(\theta_k)$, $V_k = V(\theta_k)$ and $D^k \psi_k = D^k \psi(\theta_k)$. The term A_0 , as introduced in Equation (33) in Section 3.1, satisfies $A_0 = \mathcal{L} = \mathcal{L}$ but we keep A_0 for clarity. Thus, we have

$$A_0 \psi_k = \nabla \psi \cdot \mathbb{E}_\tau \hat{f}(\theta_k, \tau, h) + \frac{1}{2} \text{trace} \left(S(\theta_k)^t \nabla^2 \psi(\theta_k) \right)$$

where $S(\theta) = g(\theta)g(\theta)^T = I$.

We use the following third order Taylor expansion on $\psi(\theta_{k+1}) - \psi(\theta_k)$ in order to obtain a bound on $\frac{1}{K} \sum_{k=0}^{K-1} (\phi_k - \bar{\phi})$

$$\begin{aligned} \psi_{k+1} &= \psi_k + \nabla \psi_k \cdot \Delta_{k+1} + \frac{1}{2} \Delta_{k+1}^T \nabla^2 \psi_k \Delta_{k+1} + \frac{1}{6} \psi_k^{(3)}(\Delta_{k+1}, \Delta_{k+1}, \Delta_{k+1}) + R_{k+1} \\ R_{k+1} &= \frac{1}{6} \int_0^1 s^3 \psi_k^{(4)}(s\theta_k + (1-s)\theta_{k+1})(\Delta_{k+1}, \Delta_{k+1}, \Delta_{k+1}, \Delta_{k+1}) ds. \end{aligned}$$

Here $\psi^{(3)}$ and $\psi^{(4)}$ are the third and fourth order derivative in the form of a trilinear and a quadrilinear form, respectively. In this setting, a third order expansion is required in order to obtain the h^2 term in the $h^2 + \frac{1}{T}$ bound in the MSE (see Equation (58) or Theorem 9). More precisely, the remainder of this expansion is fourth order which together with the term $\sqrt{h} \xi_m$ in Equation (5) contributes to the h^2 error term. In order to make the connection to the Poisson equation, we write the expansion above in terms of A_0 . This yields

$$\begin{aligned} \psi_{k+1} &= \psi_k + h A_0 \psi_k + h^{\frac{1}{2}} \nabla \psi_k \cdot \underbrace{\left(\hat{f}_k - \mathbb{E}_\tau \hat{f}(\theta_k, \tau, h) \right)}_{H_k} + h^{\frac{3}{2}} (g_k \xi_{k+1})^T \nabla^2 \psi_k \hat{f}_k \\ &\quad + \frac{1}{2} \hat{f}_k^T h^2 \nabla^2 \psi_k \hat{f}_k + \frac{1}{6} \psi_k^{(3)}(\Delta_{k+1}, \Delta_{k+1}, \Delta_{k+1}) + r_{k+1} + R_{k+1} \end{aligned}$$

where $r_{k+1} = h^{\frac{1}{2}} \left((g_k \xi_{k+1})^T \nabla^2 \psi_k (g_k \xi_{k+1}) - S(x, h) \right)$.

Notice that $\frac{1}{hK} \sum_{k=0}^{K-1} h A_0 \psi_k = \frac{1}{K} \sum_{k=0}^{K-1} (\phi_k - \bar{\phi})$ is the error of interest. In order to control this error, we sum the expression for ψ_{k+1} for $k = 0, \dots, K-1$ and divide by

$T = hK$. Grouping the terms for subsequent inspection gives

$$\begin{aligned} \frac{\psi_K - \psi_0}{Kh} &= \frac{1}{K} \sum_{k=0}^{K-1} (\phi_k - \bar{\phi}) + \underbrace{\sum_{k=0}^{K-1} (\mathcal{L} - A_0) \psi_k}_0 \\ &\quad + \underbrace{\frac{1}{T} \sum_{k=0}^{K-1} r_{k+1}}_{M_{1,K}} + \underbrace{\frac{1}{T} h^{\frac{1}{2}} \sum_{k=0}^{K-1} \nabla \psi_k (g_k \xi_{k+1})}_{M_{2,K}} + \underbrace{\frac{1}{T} h^{\frac{3}{2}} \sum_{k=0}^{K-1} \hat{f}_k^T \nabla^2 \psi_k (g_k \xi_{k+1})}_{M_{3,K}} \\ &\quad + \underbrace{\frac{1}{T} h \sum_{k=0}^{K-1} \nabla \psi_k \cdot \left(\hat{f}_k - \mathbb{E}_\tau \hat{f}(\theta_k, \tau, h) \right)}_{M_{4,K}} + \underbrace{\frac{1}{T} \frac{1}{2} \sum_{k=0}^{K-1} h^2 \hat{f}_k^T \nabla^2 \psi_k \hat{f}_k}_{S_{1,K}} \\ &\quad + \underbrace{\frac{1}{T} \sum_{k=0}^{K-1} R_{k+1}}_{S_{2,K}} + \underbrace{\frac{1}{T} \frac{1}{6} \sum_{k=0}^{K-1} \psi_k^{(3)}(\Delta_{k+1}, \Delta_{k+1}, \Delta_{k+1})}_{S_{3,K}}. \end{aligned} \quad (61)$$

where the $M_{i,k}$ indicate the martingale terms and the $S_{i,k}$ other remainder terms. We split

$$S_{3,K} = M_{0,K} + \tilde{M}_{0,K} + \tilde{S}_{0,K} + \tilde{S}_{0,K}$$

in terms of

$$\begin{aligned} M_{0,K} &= \frac{1}{6} h^{\frac{3}{2}} \sum_{k=0}^{K-1} \left(\psi_k^{(3)} \left((g_k \eta_{k+1}), (g_k \eta_{k+1}), (g_k \eta_{k+1}) \right) \right) \\ \tilde{M}_{0,K} &= \frac{1}{2} \sum_{k=0}^{K-1} h^{\frac{5}{2}} \psi_k^{(3)} \left(\hat{f}_k, \hat{f}_k, g_k \eta_{k+1} \right) \\ S_{0,K} &= \frac{1}{6} \sum_{k=0}^{K-1} 3h^2 \psi_k^{(3)} \left(g_k \eta_{k+1}, g_k \eta_{k+1}, \hat{f}_k \right) \\ \tilde{S}_{0,K} &= \frac{1}{6} \sum_{k=0}^{K-1} h^3 \psi_k^{(3)} \left(\hat{f}_k, \hat{f}_k, \hat{f}_k \right). \end{aligned}$$

Rearranging Equation (61) for $\frac{1}{K} \sum_{k=0}^{K-1} (\phi_k - \bar{\phi})$ and controlling the resulting right hand side of Equation (61) gives rise to the following theorem.

Theorem 9 *Suppose that there exists a function V such that the following three assumptions hold:*

1. *There are $p_{\psi,1}, \dots, p_{\psi,4} \in (0, \infty)$ such that the derivatives of the solution ψ to the Poisson equation satisfy the following bound*

$$\|\psi^{(k)}\| \lesssim V^{p_{\psi,k}}, \quad \text{for } k = 0, \dots, 4. \quad (62)$$

2. The drift f and the error from the estimate $H := f(\theta, \tau) - f(\theta)$ satisfy

$$\mathbb{E}_x H(\theta, \tau)^{2p} \lesssim V(\theta)^p \quad \forall p \leq p^* \quad (63)$$

$$\|f\|^2 \lesssim V.$$

for $p^* = \max\{2p_{\psi,2} + 2, 2p_{\psi,4} + 4, 2p_{\psi,3} + 1, 2p_{\psi,3} + 3\}$. Moreover, we suppose that the $\mathbb{E}V^p(\theta_k)$ is bounded from above and that this bound is independent of k , that is

$$\sup_k \mathbb{E}V^p(\theta_k) < \infty, \quad \forall p \leq p^*. \quad (64)$$

3. V satisfies

$$\sup_s V(s\theta_1 + (1-s)\theta_2)^p \lesssim V(\theta_1)^p + V(\theta_2)^p, \quad \text{for all } \theta_1, \theta_2, p \leq p^*. \quad (65)$$

Under these assumptions there exists $h_0 > 0$ and constant C such that for all $h < h_0$

$$\text{Bias}(\hat{\phi}_K) = \left| \mathbb{E}\hat{\phi}_K - \bar{\phi} \right| \leq C \left(h + \frac{1}{Kh} \right) \quad (66)$$

$$\mathbb{E} \left(\hat{\phi}_K - \bar{\phi} \right)^2 \leq C \left(h^2 + \frac{1}{Kh} \right) \quad (67)$$

where

$$\hat{\phi}_K = \frac{1}{K} \sum_{k=0}^{K-1} \phi(\theta_k) \quad \text{and} \quad \bar{\phi} = \mathbb{E}\pi\phi.$$

Proof For each term we bound the term inside the sum by a power of V^p and then obtain an overall bound using $\sup_i \mathbb{E}V_i^p < \infty$. For example, $\frac{1}{K}\mathbb{E}S_{1,K}$ can be bounded as follows

$$\begin{aligned} \frac{1}{T} \mathbb{E}S_{1,K} &\lesssim \frac{1}{T} \mathbb{E} \sum_{k=0}^{K-1} h^2 V_k^{p_{\psi,2}} \mathbb{E}\tau_k \|\hat{f}_k\|^2 \\ &\lesssim \frac{1}{T} \sum_{k=0}^{K-1} h^2 \sup_i \mathbb{E}V_i^{p_{\psi,2}+1} \lesssim \frac{1}{T} h^2 K \lesssim h. \end{aligned}$$

The details of this computation are contained in Appendix 9. \blacksquare

Theorem 9 and the results for the decreasing step size SGLD Teh et al. (2014) hold under assumptions formulated in terms of the solution ψ of the Poisson equation. More precisely, the crucial step is to establish a bound of the form

$$\sup_k \mathbb{E} \left\| \psi_{k+1}^{(i)}(\theta_k) \right\| < \infty \quad \text{for } k = 1, \dots, 4.$$

This bound is established using Equations (62) and (64)

$$\sup_k \mathbb{E} \left\| D^{(i)} \psi(\theta_k) \right\| \lesssim \sup_k \mathbb{E} \|V\|^{p_{\psi,i}} < \infty \quad \text{for } i = 1, \dots, 4.$$

Thus, we are left with finding an appropriate Lyapunov function V such that Equations (62) and (64) hold. In Appendix 9.1, we formulate strong sufficient conditions on π that ensure that these assumptions are satisfied and that Theorem 9 is applicable.

6. An Analytic Investigation of the Toy Model

We now extend our analysis of the one-dimensional Gaussian toy model introduced in Section 2 beyond the general results of the previous two sections. More precisely, in Section 6.1, we compare the Euler method, the SGLD and the mSGLD by comparing the computational cost for fixed level accuracy specified in terms of the mean square error in estimating the second moment (MSE₂), optimising over the step size h , the subsample size n and the number of steps M . A numerical solution to the resulting optimisation problem demonstrates that the SGLD is advantageous in the lower accuracy regime while it degenerates to $n = N$ in the high accuracy regime. On the other hand the mSGLD does not degenerate and seems to maintain a constant speed up compared to the Euler method. In Section 6.2 we then consider the MSE₂ and use an analytic expression to study the behaviour of these algorithms for growing N . This allows us to extend the analysis of Sections 2 and 4 (in which we only consider the case limit $h \rightarrow 0$) and study the asymptotic bias of the SGLD and the mSGLD by scaling both n and h in N .

In Section 6.3 we finally adopt a different viewpoint by considering a fixed value of our parameter θ , denoted by θ^* , while we take expectations with respect to the realisation of the data $\{X_i\}$. This enables us to study how $\mathbb{E}_X(\text{MSE}_2)$ behaves in the limit of $N \rightarrow \infty$. In particular, we find that for the case of the SGLD, the computational cost in order for $\mathbb{E}_X(\text{MSE}_2) \rightarrow 0$ is reduced by a factor of N when compared to the Euler method. A similar analysis for the expected relative error in estimating the posterior variance (ERE)

$$\text{ERE} = \mathbb{E}_{\{\theta_i\}} \left[\frac{\frac{1}{K} \sum_{i=0}^{K-1} \theta_i^2 - \left(\frac{1}{K} \sum_{i=0}^{K-1} \theta_i \right)^2}{\sigma_\theta^2} \right] - 1. \quad (68)$$

reveals that under the constraint $\mathbb{E}_X(\text{ERE}) \rightarrow 0$ the Euler method and the SGLD have the same computational cost on the algebraic scale in N .

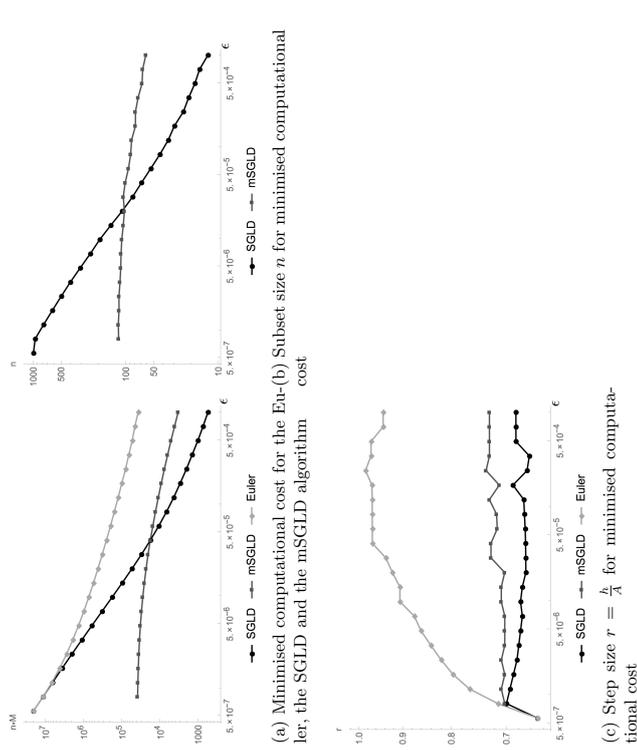
6.1 Minimising Computational Effort for Constrained MSE₂

In Section 2.2 we compared the Euler method, the SGLD and the mSGLD for the same choice of $r = \frac{r}{4}$. In the following we numerically minimise the computational effort with respect to the condition $\text{MSE}_2 \leq \epsilon^2$. We assume that the computational cost is proportional to $M \cdot n$ which leads to the problem of solving

$$\begin{aligned} \min_p \quad & M \cdot n \\ \text{subject to} \quad & \text{MSE}_2(r, M, n) \leq \epsilon^2 \\ \text{w.r.t.} \quad & r < 1, M, n. \end{aligned} \quad (69)$$

Even though we have analytic expressions for the MSE₂, the solution to the optimisation problem does not have a closed form. To conclude our analysis, we illustrate the numerical solution to this problem for $N = 1000$ for the Euler method, the SGLD and the mSGLD. The results, depicted in Figure 3, can be summarised as follows:

1. as ϵ becomes smaller, the gain of the SGLD over the Euler method in terms of computational effort decreases (due to the fact that n increases);

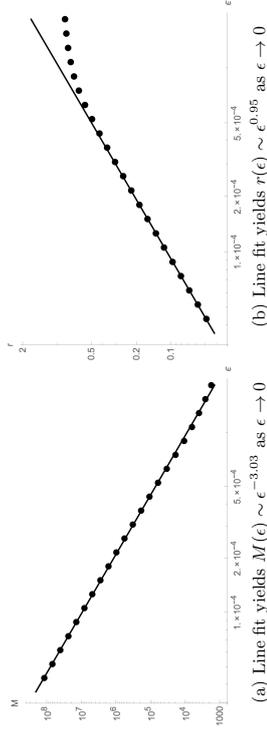
Figure 3: Minimisation of computational cost $\propto M \cdot n$ subject to $MSE \leq \epsilon^2$

- as ϵ becomes smaller, the mSGLD gains efficiency over the SGLD (the reason being that n seems to asymptote as ϵ decreases).

The upper bound obtained in Equation (67) suggests a scaling of $M \sim \epsilon^{-3}$ and $r \sim \epsilon$ to obtain an MSE of order ϵ^2 with minimal computational effort. The numerical minimisation of M with respect to r and M subject to the condition $MSE(r, M) \leq \epsilon^2$ confirms this scaling empirically, see Figure 4.

6.2 The MSE₂ for fixed and increasing N

We now consider the behaviour for growing data size N where a new data set X is generated in each instance. Figure 5 depicts the MSE₂ for $N = 10^i$ with $i = 1, \dots, 4$ for the subset choices $n = N^{0.1}, N^{0.5}$ and $N^{0.9}$; each compared to the Euler method corresponding to $n = N$. In this plot we notice that the SGLD outperforms the mSGLD for $n = N^{0.1}$ and $n = N^{0.5}$. The behaviour in Figure 5 suggests that the mSGLD has a larger bias than the SGLD which seems to contradict the findings of Section 2 and 4. Previously, we have just considered the asymptotic of $h \rightarrow 0$. In contrast, we scale both $h = r \frac{1}{\lambda(N)}$ and $n = N^p$ in

Figure 4: Scaling of $r(\epsilon)$ and $M(\epsilon)$ for minimal computational cost subject to the $MSE_2(r(\epsilon), M(\epsilon)) \leq \epsilon^2$

terms of N in Figure 5. In the following we investigate this relationship further by using the explicit formula for the asymptotic bias which has been made available in Section 2.

For simplicity we consider sampling without replacement, using the expression in Equation (16). Similar conclusions hold for sampling with replacement. First we consider the mSGLD. From Equation (21) and using the parameterisation $h = r/A$ of the step size, the excess asymptotic bias becomes

$$h^2 \frac{\text{Var}(B)^2}{4(2A - A^2h)} = \frac{(r/A)^2 \left(\frac{N-n}{n}\right)^2 N^2 \text{Var}(X)^2}{4(2 - r)A}.$$

Because $A \sim N$, we see that the excess bias stays bounded for large N if and only if $n \gtrsim N^{\frac{1}{2}}$. In contrast, the same consideration for the SGLD shows that the excess bias due to subsampling vanishes so long as $n \rightarrow \infty$ when $N \rightarrow \infty$.

We can also identify the regime in which the mSGLD has a smaller asymptotic bias than the SGLD. From Equations (18) and (21) we see that this is the case when

$$\frac{h^2 \text{Var}^2(B)}{4(2A - A^2h)} \leq \frac{h \text{Var}(B)}{2A - A^2h}. \quad (70)$$

Using Equation (16), the above can be rearranged to

$$1 \geq \frac{h}{4} \frac{1}{16\sigma_x^2} \frac{N(N-n)}{N} \text{Var}(X).$$

Let $c = \frac{N}{N-n}$ be the relative size of the subsampling. Using $h = r/A$ where A is given in Equation (15), we get

$$c \geq \frac{2rN\text{Var}X}{16\sigma_x^2 \left(\frac{1}{\sigma_\theta^2} + \frac{N}{\sigma_x^2}\right) + 2rN\text{Var}X}$$

which in the limit of large data sets $N \rightarrow \infty$ yields

$$c \geq \frac{2r\text{Var}X}{16 + 2r\text{Var}X} > 0.$$

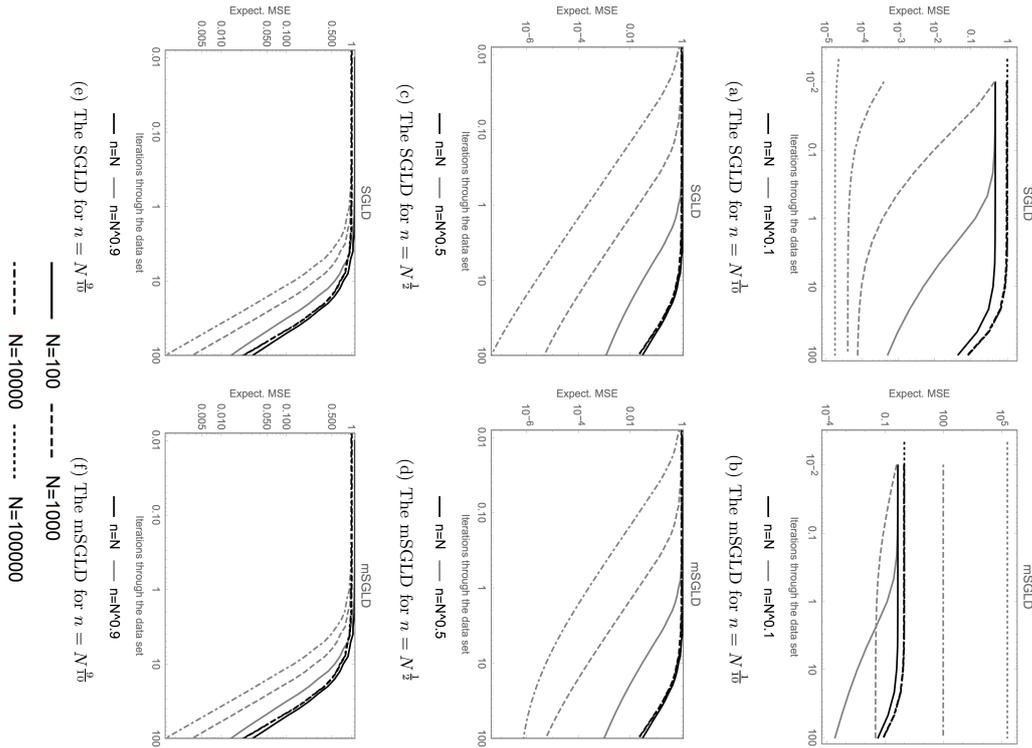


Figure 5: MSE_2 of the time average for the SGLD and the mSGLD for the second moment of the posterior as $N \rightarrow \infty$. Notice that Figures (c) and (d) and (e) and (f) have the same scaling respectively. Moreover, figures (a) and (b) have separate scaling because of the instability of the mSGLD.

In conclusion, for a fixed step size given by r/A , the mSGLD has a smaller bias than the SGLD if the above holds. In other words, the subsampling size has to be linear in the size of the data set for a fixed choice of r .

6.3 Limit of the MSE and ERE for well-specified Data as $N \rightarrow \infty$

In order to investigate the limit of $N \rightarrow \infty$, we need to specify the behaviour of the data as well. We study this in the well-specified case; in other words, we assume that the data is generated by the model for $\theta^* = 1$. Previously, we have obtained an analytic expression for the expectation of the MSE_2 with respect to the realisation of the noise driving the algorithm. In contrast, we take expectations with respect to the realisation of the data X and the noise driving the algorithm in the following results. These results are formulated in analytic expressions² for the MSE_2 and the ERE depending only on M, n, r and N . We then choose M, n and r as functions of N and study the limit $N \rightarrow \infty$ and how this affects the computational cost and the behaviour of the ERE and the MSE_2 as $N \rightarrow \infty$ for the different algorithms.

For the Euler method ($n = N$) we need to take $h < \frac{1}{4} \asymp \frac{1}{N}$ in order to make Equation (17) stable. Moreover, we need the number of steps M to be of order N to approximate the diffusion to a time of order $\mathcal{O}(1)$. Because we evaluate N data points per step, this heuristic argument suggests that the complexity is of order $\mathcal{O}(N^2)$. Furthermore, we verify (using Mathematica[®]), that for the Euler method ($n = N$)

$$\lim_{N \rightarrow \infty} \mathbb{E}_X \text{MSE} = 0, \quad \lim_{N \rightarrow \infty} \mathbb{E}_X \text{ERE} = 0 \quad (71)$$

for the choices $M = N^{1+2\epsilon}$ and $r = N^{-\epsilon}$ for any $\epsilon > 0$. The computational cost for fixed N is $M \cdot n = N^{2+2\epsilon}$. Thus, this confirms the heuristics we used for the Euler method in Section 6.2.

A natural next question to ask in terms of the SGLD is if one can have Equation (71) to hold but for smaller computational complexity than the Euler method. Using Mathematica[®], we obtain the following theorem for the MSE

Theorem 10 For any $\epsilon > 0$ and the choices $h = N^{-1-\epsilon}$, $M = N^{1+2\epsilon}$ and $n = 1$, the SGLD satisfies

$$\lim_{N \rightarrow \infty} \mathbb{E}_{\theta_{1:1}, X} \left(\frac{1}{M} \sum_{k=0}^{M-1} \theta_k^2 - (\mu_p^2 + \sigma_p^2) \right)^2 = 0.$$

This constitutes a substantial gain compared to the Euler method because it reduces the computational complexity in the data size N from being almost quadratic to almost linear.

We now draw our attention to the expected relative error in estimating the posterior variance, abbreviated by

$$\text{ERE} := \mathbb{E}_{(\theta_1)} \frac{\frac{1}{K} \sum_{i=0}^{K-1} \theta_i^2 - \left(\frac{1}{K} \sum_{i=0}^{K-1} \theta_i \right)^2}{\sigma_p^2} - 1. \quad (72)$$

² see Appendix 10.2 for a sketch of the derivation for the MSE_2 (the derivation for ERE is similar)

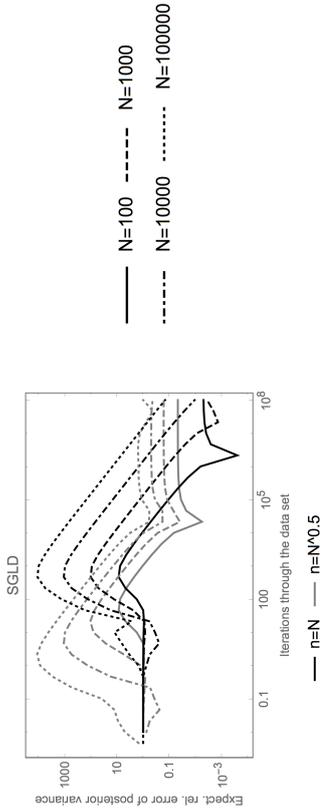


Figure 6: Expected relative error of the estimate of the variance of the posterior based on the SGLD.

Because the posterior variance goes to zero as $N \Rightarrow \infty$, it is conceivable that it requires more computational effort to ensure that $\lim_{N \rightarrow \infty} \mathbb{E}_X \text{ERE} = 0$. In order to illustrate the behaviour of the ERE, we first consider the behaviour for a fixed data set and repeat the experiment of Figure 5 in Figure 6. The latter demonstrates that the asymptotic bias for the choice $n = N^{\frac{1}{2}}$ (the asymptotes for the grey lines) have an increasing value in N . We used $h = \frac{1}{20A} \sim \frac{1}{N}$ which decreases with N . However, we show below that is requirement cancels exactly the gain from $n \ll N$ at least on the algebraic scale in N .

In particular, we now choose $r = N^{-\alpha}$, $n = N^\beta$ and $M = N^\gamma$. Hence the computational cost is $N^{\beta+\gamma}$. The step size $h = \frac{1}{N}$ satisfies $h \sim N^{-1-\alpha}$ because $A \sim N$. Since the algorithm performs $M = N^\gamma$ steps, we expect it to approximate the diffusion on the time interval $h \cdot M = N^{-1-\alpha+\gamma}$. Therefore it is reasonable to require that $\gamma > 1 + \alpha$. Under this assumption and with the help of Mathematica[®], we reduced the limit above to

$$\lim_{N \rightarrow \infty} \mathbb{E}_X \text{ERE} = \lim_{N \rightarrow \infty} \left(\frac{2 \cdot N^{-\alpha-\beta+3}}{(N+1)^2 (N^{-\alpha}-2)^2} - \frac{N^{-2\alpha-\beta+3}}{(N+1)^2 (N^{-\alpha}-2)^2} \right) = \begin{cases} 0 & \text{if } \alpha + \beta > 1 \\ \infty & \text{if } \alpha + \beta < 1. \end{cases}$$

Thus for $\lim_{N \rightarrow \infty} \mathbb{E}_X \text{ERE} = 0$ it is necessary that $\alpha + \beta \geq 1$. This condition in turn implies that the computational complexity satisfies

$$\underbrace{N^\gamma}_{\text{steps}} \times \underbrace{N^\beta}_{\text{cost per step}} = N^{1+\alpha} N^\beta = N^{1+\alpha+\beta} \gtrsim N^2.$$

Thus, there is no computational gain for the ERE in the limit $N \rightarrow \infty$ on the algebraic scale in N . We note that picking $\theta^* = \frac{1}{N}$ instead of $\theta^* = 1$ does not change the results. Thus, a closer initialisation does not change the result for the ERE.

7. Logistic Regression

In the following we present numerical simulations for a Bayesian logistic regression model. The data items are given by covariates $x_i \in \mathbb{R}^d$ that are labeled by $y_i \in \{-1, 1\}$. We assume the data $y_i \in \{-1, 1\}$ is modelled by

$$p(y_i | x_i, \beta) = \sigma(y_i \beta^T x_i) \quad (73)$$

where $\sigma(z) = \frac{1}{1 + \exp(-z)} \in [0, 1]$. The model poses the assumption that y_i depends on x_i through the linear relationship $\beta^T x_i$. Nevertheless, logistic regression is commonly used after a preprocessing has taken place and is therefore used here for numerical illustration.

We put a Gaussian prior $\mathcal{N}(0, C_0)$ on β , for simplicity we use $C_0 = I$ subsequently. By Bayes' rule the posterior π satisfies

$$\pi(\beta) \propto \exp\left(-\frac{1}{2} \|\beta\|_{C_0}^2\right) \prod_{i=1}^N \sigma(y_i \beta^T x_i).$$

We consider $d = 3$ and $N = 1000$ data points and choose the covariate to be

$$x = \begin{pmatrix} x_{1,1} & x_{1,2} & 1 \\ x_{2,1} & x_{2,2} & 1 \\ \vdots & \vdots & \vdots \\ x_{1000,1} & x_{1000,2} & 1 \end{pmatrix}$$

for a fixed sample of $x_{i,j} \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$ for $i = 1, \dots, 1000$ and $j = 1, 2$. We use a long run of the Random-Walk-Metropolis algorithm to estimate the posterior mean.

On that basis we estimate the MSE of the SGLD based mean estimate using 100 runs of the algorithm with step size $h = 0.002$ for various subset sizes. Figure 7 depicts the MSE as function of the iterations and effective iterations through the data set. Notice that for this example the variance of the stochastic gradient $f(\theta)$ depends on both θ and all the data items. For this reason we replace $\text{Var } f(\theta)$ by an estimate $\text{Var } \hat{f}(\theta)$, see also Remark 7. We note that the mSGLD is superior for $n = 150$, inferior for $n = 50$ and for $n = 10$ the MSE of the mSGLD does not drop below 1.

8. Conclusion

This article presents the mathematical foundations that are necessary for posterior sampling for stochastic gradient methods with fixed step sizes. We derived an error expansion of the asymptotic bias in terms of powers of the step size and identified how the constant in the leading order term depends on the unbiased estimator of the gradient. We construct a modified SGLD to match the Euler method in this asymptotic expansion. These asymptotic results are complemented by upper bounds on the bias and the MSE over a finite time horizon. Minimising the MSE with respect to the step size yields a decay of the error at the same rate as the decreasing step size SGLD, see Remark 8. These theoretical findings are completed with extensive analytic investigations of a one dimensional toy model that allows the derivation of analytic expressions for the sample average and its moments. Finally, this

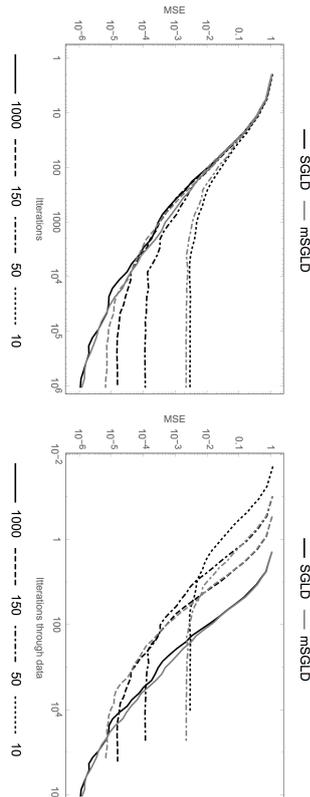


Figure 7: Expected MSE of time average for the SGLD and the mSGLD for the mean of the posterior

yields an exact quantification of expected errors. The results of this investigation can be summarised as follows:

- In the high accuracy regime the SGLD deteriorates to the Euler method while the mSGLD prevails.
- For small data batches the bias of the mSGLD is larger than for the SGLD.
- In the limit as the number of data items goes to infinity the SGLD reduces computational complexity of estimating the second moment with vanishing MSE by one power of the number of data items.

This recommends the construction of new and a study of existing modifications of the SGLD such as the Stochastic Gradient Hamiltonian Monte Carlo Chen et al. (2014) and the Stochastic Gradient Thermostat Monte Carlo Ding et al. (2014) algorithms.

Acknowledgement

STV and YWT acknowledge EPSRC for research funding through grant EP/K009850/1, EP/N000188/1 and EP/K009362/1. The authors thank Rémi Bardenet for fruitful discussions.

9. Appendix A: Finite time Ergodic Average based Poisson Equation

Proof [Proof of Theorem 9] Rearranging Equation (61) for $\frac{1}{K} \sum_{k=0}^{K-1} (\phi_k - \bar{\phi})$, the bias and the MSE can be controlled as follows:

$$\mathbb{E} \frac{1}{K} (\psi_K - \psi_0) \lesssim \frac{1}{T} \sup_i \mathbb{E} V_i^{p_{\psi,0}} \lesssim \frac{1}{T}.$$

Because $A_0 = \mathcal{L}$ for the SGLD it is left to bound to bound $\mathbb{E} \frac{1}{T} S_{i,K}$ for $i = 0 \dots 4$. First we consider $i = 1$ and use Equation (A.6)

$$\begin{aligned} \frac{1}{T} \mathbb{E} S_{1,K} &\lesssim \frac{1}{T} \mathbb{E} \sum_{k=0}^{K-1} h^2 V_k^{p_{\psi,2}} \mathbb{E}_{\tau_k} \|\hat{f}_k\|^2 \\ &\lesssim \frac{1}{T} \sum_{k=0}^{K-1} h^2 \sup_i \mathbb{E} V_i^{p_{\psi,2}+1} \lesssim \frac{1}{T} h^2 K \lesssim h. \end{aligned}$$

This procedure will be used over and over again. It can be summarised as follows:

1. bounding the terms in the sum by a power of V^p , using Equation (A.6) and the assumption on derivatives of ψ ;
2. then derive the bound using $\sup_i \mathbb{E} V_i^p < \infty$.

For $i = 2$ we additionally use that Equation (65) implies

$$\int_0^1 s^3 \psi^{(4)}(s\theta_k + (1-s)\theta_{k+1}) (\Delta_{k+1}, \Delta_{k+1}, \Delta_{k+1}, \Delta_{k+1}) ds \lesssim (V_k^{p_{\psi,4}} + V_{k+1}^{p_{\psi,4}}) \|\Delta_k\|^4,$$

which allows us to follow the general procedure

$$\begin{aligned} \frac{1}{T} \mathbb{E} S_{2,K} &\lesssim \frac{1}{T} \mathbb{E} \sum_{k=0}^{K-1} R_{k+1} \\ &\lesssim \frac{1}{T} \mathbb{E} \sum_{k=0}^{K-1} h^2 (V_k^{p_{\psi,4}} + V_{k+1}^{p_{\psi,4}}) \mathbb{E}_{\tau} \|\Delta_{k+1}\|^4 \\ &\lesssim \frac{1}{T} \mathbb{E} \sum_{k=0}^{K-1} h^2 \sup_i \mathbb{E} V_i^{p_{\psi,4}+2}(\theta_i) \lesssim \frac{1}{T} h^2 \lesssim h. \end{aligned}$$

We apply the general procedure to $\mathbb{E} S_{0,K}$

$$\begin{aligned} \frac{1}{T} \mathbb{E} S_{0,K} &\lesssim \frac{1}{T} \mathbb{E} \sum_{k=0}^{K-1} h^2 V_k^{p_{\psi,3}} \mathbb{E}_{\tau} \|g_k \eta_{k+1}\|^2 \|\hat{f}_k\| \\ &\lesssim \frac{1}{T} \mathbb{E} \sum_{k=0}^{K-1} h^2 \sup_i \mathbb{E} V_i^{p_{\psi,3}+\frac{1}{2}}(\theta_i) \\ &\lesssim \frac{1}{T} h^2 \lesssim h. \\ \frac{1}{T} \mathbb{E} \tilde{S}_{0,K} &\lesssim \frac{1}{T} \mathbb{E} \sum_{k=0}^{K-1} h^3 V_k^{p_{\psi,3}} \mathbb{E}_{\tau} \|\hat{f}_k\|^3 \\ &\lesssim \frac{1}{T} \sum_{k=0}^{K-1} h^3 \sup_i \mathbb{E} V_i^{p_{\psi,3}+\frac{3}{2}}(\theta_i) \lesssim \frac{1}{T} K h^3 \lesssim h^2. \end{aligned}$$

Thus, we have established the bound on the bias given by Equation (A.3).

In order to establish the bound one the MSE in Equation (67) we note that Equation (61) yields

$$\begin{aligned} \mathbb{E} \left(\frac{1}{K} \sum_{k=0}^{K-1} (\phi - \bar{\phi}) \right)^2 &\lesssim \mathbb{E} \frac{(\psi_K - \psi)^2}{T^2} \\ &+ \frac{1}{T^2} \sum_{i=0}^2 \mathbb{E} S_{i,K}^2 + \frac{1}{T^2} \sum_{i=0}^2 \mathbb{E} M_{i,K}^2 \end{aligned}$$

First we note that

$$\mathbb{E} \frac{(\psi_K - \psi)^2}{T^2} \lesssim \frac{1}{T^2} \sup_i \mathbb{E} V_i^{2p_{\psi,0}} \lesssim \frac{1}{T^2}.$$

The $S_{i,K}^2$ terms can be bound in similar way as above with the additional use of Cauchy-Schwartz inequality to break the correlation between V_i^p and V_j^p .

$$\begin{aligned} \frac{1}{T^2} \mathbb{E} S_{1,K}^2 &\lesssim \frac{1}{T^2} \mathbb{E} \sum_{i,j=0}^{K-1} h^4 \nabla^2 \psi_i [f_i, f_i] \nabla^2 \psi_j [f_j, f_j] \\ &\lesssim \frac{1}{T^2} \mathbb{E} \sum_{i,j=0}^{K-1} h^4 \|\nabla^2 \psi_i\| \|\mathbb{E}_\tau [f_i]\|^2 \|\nabla^2 \psi_j\| \|\mathbb{E}_\tau [f_j]\|^2 \\ &\lesssim \frac{1}{T^2} \sum_{i,j=0}^{K-1} h^4 \mathbb{E} V_i^{2p_{\psi,2}+1} V_j^{2p_{\psi,2}+1} \\ &\lesssim \frac{1}{T^2} \sum_{i,j=0}^{K-1} h^4 \left(\mathbb{E} V_i^{2p_{\psi,2}+2} \right)^{\frac{1}{2}} \left(\mathbb{E} V_j^{2p_{\psi,2}+2} \right)^{\frac{1}{2}} \\ &\lesssim \frac{1}{T^2} \sum_{i,j=0}^{K-1} h^4 \left(\sup_i \mathbb{E} V_i^{2p_{\psi,2}+2} \right) \lesssim \frac{K^2 h^4}{T^2} \lesssim h^2 \end{aligned}$$

Similarly, we bound

$$\begin{aligned} \frac{1}{T^2} \mathbb{E} M_{1,K}^2 &= \frac{1}{T^2} \sum_{i=0}^{K-1} \left(\mathbb{E} D_{lm}^2 \psi_i \left(g_i^{l,a} \eta_{i+1}^{m,b} \eta_{i+1}^{k,l} - g_i^{k,l} g_i^{m,b} \right) \right)^2 \\ &\lesssim \frac{1}{T^2} \sum_{i=0}^{K-1} \sup_i \mathbb{E} V_i^{2p_{\psi,2}}. \end{aligned}$$

The following term is the crucial Martingale term as it yields the $\mathcal{O}(\frac{1}{T})$ contribution

$$\begin{aligned} \frac{1}{T^2} \mathbb{E} M_{2,K}^2 &\lesssim \frac{1}{T^2} h \sum_{k=0}^{K-1} \mathbb{E} \|\nabla \psi_k\|^2 \|g_k \xi_{k+1}\|^2 \\ &\lesssim \frac{1}{T^2} h \sum_{k=0}^{K-1} \mathbb{E} V_k^{2p_{\psi,1}} \lesssim \frac{1}{T}. \end{aligned}$$

$$\begin{aligned} \frac{1}{T^2} \mathbb{E} S_{2,K}^2 &\lesssim \frac{1}{T^2} \mathbb{E} \sum_{i,j=0}^{K-1} R_{i+1} R_{j+1} \\ &\lesssim \frac{1}{T^2} \mathbb{E} \sum_{i,j=0}^{K-1} (V_i^{2p_{\psi,4}} + V_{i+1}^{2p_{\psi,4}}) (V_j^{2p_{\psi,4}} + V_{j+1}^{2p_{\psi,4}}) \mathbb{E}_\tau \|\Delta_{i+1}\|^4 \mathbb{E}_\tau \|\Delta_{j+1}\|^4 \\ &\lesssim \frac{h^4}{T^2} \mathbb{E} \sum_{i,j=0}^{K-1} (V_i^{2p_{\psi,4}} + V_{i+1}^{2p_{\psi,4}}) (V_j^{2p_{\psi,4}} + V_{j+1}^{2p_{\psi,4}}) V_i^2 V_j^2 \\ &\lesssim \frac{h^4}{T^2} \mathbb{E} \sum_{i,j=0}^{K-1} (V_i^{2p_{\psi,4}} + V_{i+1}^{2p_{\psi,4}}) (V_j^{2p_{\psi,4}} + V_{j+1}^{2p_{\psi,4}}) V_i^2 V_j^2 \\ &\lesssim \frac{h^4}{T^2} \sum_{i,j=0}^{K-1} \left(\sup_i \mathbb{E} V_i^{2p_{\psi,4}+4} \right) \lesssim \frac{K^2 h^4}{T^2} \lesssim h^2. \end{aligned}$$

Similar bounds can also be obtained for $\frac{1}{T^2} \mathbb{E} S_{0,K}^2$ and $\frac{1}{T^2} \mathbb{E} \tilde{S}_{0,K}^2$

$$\begin{aligned} \frac{1}{T^2} \mathbb{E} S_{0,K}^2 &\lesssim \frac{h^4}{T^2} \sum_{i,j=0}^{K-1} \mathbb{E} V_i^{2p_{\psi,3}} \|f_i\| V_j^{2p_{\psi,3}} \|f_j\| \\ &\lesssim \frac{h^4 K^2}{T^2} \sup_i \mathbb{E} V_i^{2p_{\psi,3}+1} \lesssim h^2 \\ \frac{1}{T^2} \mathbb{E} \tilde{S}_{0,K}^2 &\lesssim \frac{h^6}{T^2} \sum_{i,j=0}^{K-1} \mathbb{E} V_i^{2p_{\psi,3}} \|f_i\|^3 V_j^{2p_{\psi,3}} \|f_j\|^3 \\ &\lesssim \frac{h^6 K^2}{T^2} \sup_i \mathbb{E} V_i^{2p_{\psi,3}+2} \lesssim h^4 \end{aligned}$$

For Martingale terms the cross terms vanish which allows us to obtain the following bounds

Similarly, we estimate

$$\begin{aligned} \frac{1}{T^2} \mathbb{E} M_{3,K}^2 &\lesssim \frac{1}{T^2} h^3 \sum_{k=0}^{K-1} \mathbb{E} \|\nabla^2 \psi_k\|^2 \|f_k\|^2 \|g_k \xi_{k+1}\|^2 \\ &\lesssim \frac{1}{T^2} h^3 \sum_{k=0}^{K-1} \mathbb{E} V_k^{2p_{\psi,2}+1} \lesssim \frac{h^2}{T}. \end{aligned}$$

The terms $\frac{1}{T^2} \mathbb{E} \bar{M}_{0,K}^2$ and $\frac{1}{T^2} \mathbb{E} \bar{M}_{0,K}^2$ can be bounded in the same way

$$\begin{aligned} \frac{1}{T^2} \mathbb{E} M_{0,K}^2 &\lesssim \frac{h^3}{T^2} \sum_{k=0}^{K-1} \mathbb{E} V_k^{2p_{\psi,3}} \lesssim \frac{h^3 K}{T^2} \leq \frac{h^2}{T} \\ \frac{1}{T^2} \mathbb{E} \bar{M}_{0,K}^2 &\lesssim \frac{h^5}{T^2} \sum_{k=0}^{K-1} \mathbb{E} V_k^{2p_{\psi,3}+2} \lesssim \frac{h^5 K}{T^2} \leq \frac{h^4}{T} \end{aligned}$$

The additional part for the SGLD is the term corresponding to the Martingale $M_{4,K}$

$$\begin{aligned} \frac{1}{T^2} M_{4,K}^2 &\lesssim \frac{1}{T^2} \mathbb{E} h^2 \sum_{k=0}^{K-1} (\nabla \psi_k(H_k))^2 \\ &\lesssim \frac{1}{T^2} \mathbb{E} h^2 \sum_{k=0}^{K-1} V_k^{2p_{\psi,1}} \mathbb{E}_\tau \|H_k\|^2 \\ &\lesssim \frac{1}{T^2} h^2 \sum_{k=0}^{K-1} \mathbb{E} V_k^{2p_{\psi,1}+1} \lesssim \frac{h}{T} \end{aligned}$$

For all these calculations to go through need $\sup_i E V_i^{p^*}$ to be bounded. Collecting the orders present, we see that

$$p^* = \max \{2p_{\psi,2} + 2, 2p_{\psi,4} + 4, 2p_{\psi,3} + 1, 2p_{\psi,3} + 3\}$$

is sufficient. \blacksquare

9.1 Sufficient Conditions on π Ensuring Finite Time Bounds on Bias and MSE

We formulate a sufficient condition on π that ensure that Theorem 9 is applicable. This hinges on deriving a sufficient condition for Equations (62), (64) and (65). The aim of this section is to establish and motivate the sufficient condition formulated in the following theorem.

Theorem 11 *Suppose the following condition holds*

$$\langle \theta, \nabla \log \pi_0(\theta) \rangle \leq -\alpha \|\theta\|^2 + \beta \tag{A.1}$$

$$\langle \theta, \nabla \log \pi(X_i | \theta) \rangle \leq -\alpha \|\theta\|^2 + \beta \text{ for } i = 1, \dots, N. \tag{A.2}$$

then Theorem 9 is applicable for polynomially bounded and continuous ϕ , that is

$$\begin{aligned} \text{Bias}(\hat{\phi}_K) &= \left| \mathbb{E} \hat{\phi}_K - \bar{\phi} \right| \leq C \left(h + \frac{1}{Kh} \right) \\ \mathbb{E} \left(\hat{\phi}_K - \bar{\phi} \right)^2 &\leq C \left(h^2 + \frac{1}{Kh} \right) \end{aligned}$$

Proof This result follows from Theorem 5.2 and results quoted below. \blacksquare

First we appeal to a sufficient condition for Equation (64) before summarising a regularity results of Pardoux and Veretennikov (2001) which allows us to establish Equation (62). We believe that the sufficient conditions above can be weakened, but this requires improving the results of Pardoux and Veretennikov (2001) which is out of the scope of this article.

The condition $\sup_k \mathbb{E} V^p(\theta_k) < \infty$ (that is Equation (64)) is established for all $p \leq p^*$ by Lemma 5 in Teh et al. (2014) if p^* satisfies the following assumption.

Assumption 12 *The drift term $\theta \mapsto \frac{1}{2} \nabla \log \pi(\theta)$ is continuous. The function $V : \mathbb{R}^d \rightarrow [1, \infty)$ tends to infinity as $\|\theta\| \rightarrow \infty$, is twice differentiable with bounded second derivatives and satisfies the following conditions:*

1. V is a Lyapunov function for the Langevin dynamics, i.e. there are constants $\alpha, \beta > 0$ such that for every $\theta \in \mathbb{R}^d$ we have

$$\left\langle \nabla V(\theta), \frac{1}{2} \nabla \log \pi(\theta) \right\rangle \leq -\alpha V(\theta) + \beta. \tag{A.3}$$

2. The following bounds hold

- There exists an exponent $p_H \geq 2$ such that
$$\mathbb{E} \|H(\theta, \mathcal{U})\|^{2p_H} \lesssim V^{p_H}(\theta). \tag{A.4}$$

Moreover, this implies that $\mathbb{E} \|H(\theta, \mathcal{U})\|^{2p} \lesssim V^p(\theta)$ for any exponent $0 \leq p \leq p_H$.

- For every $\theta \in \mathbb{R}^d$ we have

$$\|\nabla V(\theta)\|^2 + \|\nabla \log \pi(\theta)\|^2 \lesssim V(\theta). \tag{A.5}$$

Notice that for \hat{f} based on subsampling we obtain that $p_H = \infty$ if

$$\|\nabla \log \pi(x|\theta)\|^2 \leq C(x) V(\theta).$$

Notice that Assumption 12 also implies

$$\begin{aligned} \mathbb{E} \|\theta_{k+1} - \theta_k\|^{2p} &\leq V_k^p \\ \mathbb{E} \|\hat{f}_k\|^{2p} &\leq V_k^p \end{aligned} \tag{A.6}$$

if for $p \leq p_H$. Equation (65) could now simply be formulated as an additional assumption, however currently we need even stronger assumptions to verify Equation (62). Subsequently, we show how the results of Pardoux and Veretennikov (2001) can be used to establish Equation (62) if Equations (A.3) and (A.5) hold for $V = \|\theta\|^2 + 1$.

Theorem 1 and 2 of Pardoux and Veretennikov (2001) characterise the smoothness and growth of the solution to the Poisson equation associated with Equation (60). This is important for our results because the key ingredient for the proof of Theorem 9 is

$$\sup_k \mathbb{E} \left\| \psi^{(i)}(\theta_k) \right\| < \infty \text{ for } k = 1, \dots, 4.$$

Because we have already established in Section 4 that $\sup_i \mathbb{E} V^p(\theta_i) < \infty$ it is left to verify

$$\left\| \psi^{(k)} \right\| \lesssim V^{p_{\psi,k}}, \quad \text{for } k = 0, \dots, 4 \quad (62)$$

The assumptions needed to apply the Theorem 9 results are

$$\left\langle f(\theta), \frac{\theta}{\|\theta\|} \right\rangle \leq -r \|\theta\|, \quad \|\theta\| \geq M_0 \quad (A.7)$$

$$0 < \lambda_- \leq \left\langle g(\theta) g^*(\theta) \frac{\theta}{\|\theta\|}, \frac{\theta}{\|\theta\|} \right\rangle \leq \lambda_+ < \infty. \quad (A.8)$$

This holds if Assumption 12 is satisfied with $V(\theta) = \|\theta\|^2 + 1$.

Theorem 13 *Pardoux and Veretennikov (2001)* Let $\bar{f} = 0$ and Equations (A.7) and (A.8) are satisfied. Then there exists a solution $\psi \in W_{loc}^2$ to the Poisson equation

$$\mathcal{L}\psi = \phi - \bar{\phi}$$

1. If there is a C such that

$$|\phi(\theta)| \leq C(1 + \|\theta\|)^\beta$$

for some $\beta < 0$, then ψ is bounded. Moreover,

$$\sup_{\theta} |\psi(\theta)| \leq C \sup_{\theta} |f|(1 + \|\theta\|)^{-\beta}$$

and

$$\|\nabla\psi\| \leq C$$

2. if there exists a constant C and some $\beta > 0$ such that

$$|\phi(\theta)| \leq C(1 + \|\theta\|)^\beta$$

then there exist a constant such that

$$|\psi(\theta)| \leq C'(1 + \|\theta\|)^\beta.$$

Finally there exists C such that

$$\|\nabla\psi\| \leq C(1 + \|\theta\|^\beta).$$

Remark 14 *We believe that assumption Theorem 13 can be weakened to be of the form of Equation (A.3) but it is out of the scope of this article to explore this direction.*

In order to iterate Theorem 13 we note that the derivatives ψ can be expressed as solution to Poisson equations with different RHSs.

$$\mathcal{L}\psi = \phi - \bar{\phi} \quad (A.9)$$

$$A\partial_i\psi = \partial_i\phi - \frac{1}{2}\nabla\psi \cdot \partial_i\psi \quad (A.10)$$

$$A\partial_{ij}\psi = \partial_{ij}\phi - \frac{1}{2}\nabla\partial_j\psi \cdot \partial_i\psi - \frac{1}{2}\nabla\psi \cdot \partial_{ij}\psi - \frac{1}{2}\nabla\psi \cdot \partial_j\psi \cdot \partial_i\psi \quad (A.11)$$

$$A\partial_{ijk}\psi = \partial_{ijk}\phi - \frac{1}{2}\nabla\partial_k\psi \cdot \partial_{ij}\psi - \frac{1}{2}\nabla\partial_j\psi \cdot \partial_{ik}\psi - \frac{1}{2}\nabla\partial_i\psi \cdot \partial_{jk}\psi - \frac{1}{2}\nabla\psi \cdot \partial_{ijk}\psi - \frac{1}{2}\nabla\psi \cdot \partial_{ij}\psi \cdot \partial_k\psi - \frac{1}{2}\nabla\psi \cdot \partial_{jk}\psi \cdot \partial_i\psi - \frac{1}{2}\nabla\psi \cdot \partial_{ik}\psi \cdot \partial_j\psi \quad (A.12)$$

We will denote by $\beta_{\psi,i}$ numbers that satisfy

$$\sup_{|\alpha|=i} \|\theta^\alpha \psi\| \lesssim (1 + \|\theta\|^{\beta_{\psi,i}}) \quad (A.13)$$

where we used multi-index notation for derivatives. We use a similar notation for the derivatives of f , that is $\beta_{f,i}$ and assume that these bounds are a priori given.

Using Theorem 13 we can obtain $p_{\psi,i}$ to satisfy Equation (62) in terms of the β 's which we formulate as the following lemma.

Lemma 15 *Suppose that ϕ and its derivatives are bounded and Assumption 12 and Equations (A.7) and (A.8) hold. Then the choice*

$$\begin{aligned} p_{\psi,0} &= 0 \\ p_{\psi,1} &= 0 \\ p_{\psi,2} &= \frac{\beta_{f,1}}{2} \\ p_{\psi,3} &= \beta_{f,1} \vee \frac{\beta_{f,2}}{2} \\ p_{\psi,4} &= \frac{1}{2} (3\beta_{f,1} \vee (\beta_{f,1} + \beta_{f,2}) \vee \beta_{f,3}) \end{aligned}$$

satisfies Equation (62).

Proof Assumption 12 yields that $\beta_{f,0} = 1$ is a valid choice. Applying Theorem 13 to Equation (A.9) implies that $\beta_{\psi,0} := \beta_{\psi,1} = 0$ satisfies Equation (A.13). Applying Theorem 13 to Equation (A.10) yields that

$$\beta_{\psi,2} \leq \beta_{f,1}.$$

Applying Theorem 13 to Equation (A.11) yields that

$$\beta_{\psi,3} \leq 2\beta_{f,1} \vee \beta_{f,2}.$$

Applying Theorem 13 to Equation (A.12) yields that

$$\begin{aligned} \beta_{\psi,4} &\leq (\beta_{f,1} + (2\beta_{f,1} \vee \beta_{f,2})) \vee (\beta_{f,1} + \beta_{f,2}) \vee \beta_{f,3} \\ &\leq 3\beta_{f,1} \vee (\beta_{f,1} + \beta_{f,2}) \vee \beta_{f,3}. \end{aligned}$$

Thus we have established Equation (62). \blacksquare

10. Analytic expressions for the Gaussian Toy Model

We sketch the derivation of the analytic expression of the MSE are used for the plots in Section 6.

10.1 Expected MSE for fixed Data Sets

We outline how an analytic expression for the MSE of the sample average can be derived. The following method generalises to any polynomial test function but we concentrate on the sample average for the second moment of the posterior given by $S_2 = \frac{1}{M} \sum_{j=0}^{M-1} \theta_j^2$. Its MSE can be expressed using Equation (12)

$$MSE = \mathbb{E} \left(\frac{1}{M} \sum_{k=0}^{M-1} \theta_j^2 - (\mu_j^2 + \sigma_j^2) \right)^2 = \mathbb{E} S_2^2 - 2\mathbb{E} S_2 (\mu_j^2 + \sigma_j^2) + (\mu_j^2 + \sigma_j^2)^2. \quad (\text{C.1})$$

In order to express $\mathbb{E} S_2^2$ in Equation (C.1) we derive the recurrence equations for $\mathbb{E} \theta_j^i$ for $i = 1, \dots, 4$ by taking the expectations of

$$\theta_{j+1} = (1 - Ah)\theta_j + hB_j + \sqrt{h}\eta_j$$

$$\theta_{j+1}^2 = (1 - Ah)^2 \theta_j^2 + h^2 B_j^2 + h\eta_j^2 + 2(1 - Ah)\theta_j h B_j + 2(1 - Ah)\theta_j \sqrt{h}\eta_j + 2hB_j \sqrt{h}\eta_j \quad (\text{C.2})$$

$$\begin{aligned} \theta_{j+1}^3 &= \left((1 - Ah)\theta_j + hB_j + \sqrt{h}\eta_j \right)^3 \\ &= (1 - Ah)^3 \theta_j^3 + 3(1 - Ah)^2 \theta_j^2 h B_j + 3(1 - Ah)\theta_j h^2 B_j^2 + h^3 B_j^3 \\ &\quad + 3\sqrt{h}\eta_j (\dots) + 3\eta_j^2 h ((1 - Ah)\theta_j + hB_j) + \eta_j^3 h^{\frac{3}{2}} \end{aligned}$$

$$\theta_{j+1}^4 = (1 - Ah)^4 \theta_j^4 + 4(1 - Ah)^3 \theta_j^3 h B_j + 6(1 - Ah)^2 \theta_j^2 h^2 B_j^2 \quad (\text{C.3})$$

$$+ 4(1 - Ah)\theta_j h^3 B_j^3 + h^4 B_j^4 + 4\eta_j (\dots) + 4\eta_j^3 (\dots) \quad (\text{C.4})$$

$$+ 6h\eta_j^2 ((1 - Ah)^2 \theta_j^2 + 2(1 - Ah)\theta_j h B_j + h^2 B_j^2) + \eta_j^4 h^2. \quad (\text{C.5})$$

The recurrent equation for $\mathbb{E} \theta_j$ is linear and first order and can therefore be solved explicitly. Plugging the result into the equation for $\mathbb{E} \theta_j^2$ turns it into a first order linear equation as well. Repeating this processes yields explicit expressions for $\mathbb{E} \theta_j^i$ $i = 1, \dots, 4$. The sums can be carried out explicitly because the terms are of the form of a geometric sum or a geometric term with a polynomial factor. This allows us to obtain an analytic expression for $\mathbb{E} S_2^2$ by reducing it to $\mathbb{E} \theta_j^i$ as follows

$$\mathbb{E} S_2^2 = \frac{1}{M^2} \mathbb{E} \left(\sum_{i=0}^{M-1} \theta_i^4 + 2 \sum_{i=0}^{M-1} \theta_i^2 \sum_{j=i+1}^{M-1} \theta_j^2 \right). \quad (\text{C.6})$$

The cross terms can be removed using Equation (C.2) so that

$$\begin{aligned} \theta_j^2 &= (1 - Ah)^{2(j-i)} \theta_i^2 + \sum_{k=0}^{j-1-i} (1 - Ah)^{2k} \\ &\quad [h^2 B_{j-1-k}^2 + h^2 \eta_{j-1-k}^2 + 2(1 - Ah) B_{j-1-k} h \theta_{j-1-k} + \\ &\quad + 2(1 - Ah) \eta_{j-1-k} \theta_{j-1-k} \sqrt{h} + 2h^{\frac{3}{2}} B_{j-1-k} \eta_{j-1-k}]. \end{aligned} \quad (\text{C.7})$$

Plugging this into Equation (C.6) yields

$$\begin{aligned} \mathbb{E} S_2^2 &= \mathbb{E} \frac{1}{M^2} \sum_{i=0}^{M-1} \theta_i^4 \left(1 + 2 \sum_{j=i+1}^{M-1} (1 - Ah)^{2(j-i)} \right) \\ &\quad + \mathbb{E} \frac{1}{M^2} \sum_{i=0}^{M-1} \theta_i^2 \sum_{j=i+1}^{M-1} \sum_{k=0}^{j-1-i} (1 - Ah)^{2k} \\ &\quad [h^2 B_{j-1-k}^2 + h\eta_{j-1-k}^2 + h\eta_{j-1-k}^2 \sqrt{h} + 2(1 - Ah) B_{j-1-k} h \theta_{j-1-k} \\ &\quad + 2(1 - Ah) \eta_{j-1-k} \sqrt{h} + 2h^{\frac{3}{2}} B_{j-1-k} \eta_{j-1-k}]. \end{aligned}$$

Using the recurrence Equation to express θ_{j-1-k} in terms of θ_i we conclude that $\mathbb{E} S_2^2$ is equal to

$$\begin{aligned} &\frac{1}{M^2} \sum_{i=0}^{M-1} \mathbb{E} \theta_i^4 \left(1 + 2 \sum_{j=i+1}^{M-1} (1 - Ah)^{j-i} \right) \\ &\quad + \frac{1}{M^2} \sum_{i=0}^{M-1} \mathbb{E} \theta_i^2 \sum_{j=i+1}^{M-1} \sum_{k=0}^{j-1-i} (1 - Ah)^{2k} [h^2 \mathbb{E} B^2 + h]. \\ &\quad + \mathbb{E} \frac{1}{M^2} \sum_{i=0}^{M-1} \theta_i^2 \sum_{j=i+1}^{M-1} \sum_{k=0}^{j-1-i} (1 - Ah)^{2k} 2(1 - Ah) B_{j-1-k} h \\ &\quad \left((1 - Ah)^{(j-1-k)-i} \theta_i + \sum_{l=0}^{(j-1-k)-i-1} (1 - Ah)^l (h B_{j-1-k-l-1} + \sqrt{h} \eta_{j-1-k-l-1}) \right) \end{aligned}$$

Taking the expectations into the sum yields

$$\begin{aligned} \mathbb{E}S_2^2 &= \frac{1}{M^2} \sum_{i=0}^{M-1} \mathbb{E}\theta_i^4 \left(1 + 2 \sum_{j=i+1}^{M-1} (1-Ah)^{j-i} \right) \\ &\quad + \frac{1}{M^2} \sum_{i=0}^{M-1} \mathbb{E}\theta_i^2 \sum_{j=i+1}^{M-1} \sum_{k=0}^{j-1-i} (1-Ah)^{2k} [h^2 \mathbb{E}B^2 + h]. \\ &\quad + \mathbb{E} \frac{1}{M^2} \sum_{i=0}^{M-1} \sum_{j=i+1}^{M-1} \sum_{k=0}^{j-1-i} (1-Ah)^{2k} 2(1-Ah) \mathbb{E}Bh(1-Ah)^{j-1-k-i} \\ &\quad + \frac{1}{M^2} \sum_{i=0}^{M-1} \mathbb{E}\theta_i^2 \sum_{j=i+1}^{M-1} \sum_{k=0}^{j-1-i} (1-Ah)^{2k} 2(1-Ah) \mathbb{E}Bh \sum_{l=0}^{j-1-k-i-1} (1-Ah)^l h \mathbb{E}B. \end{aligned}$$

We have an expressions for $\mathbb{E}B$ but in the following we derive the expressions for $\mathbb{E}B^2$ required to express $\mathbb{E}S_2^2$. The terms $\mathbb{E}B^3$ and $\mathbb{E}B^4$ are needed for the derivation of $\mathbb{E}\theta_j^4$. In order to derive expressions for $\mathbb{E}B^p$, we introduce the power sums

$$p_k = \sum_{i=1}^N X_i^k$$

and the elementary symmetric polynomials

$$e_0 = 1, e_1 = \sum_{i=1}^N X_i, e_2 = \sum_{1 \leq i < j \leq N} X_i X_j, \dots, e_N = \prod_{i=1}^N X_i. \quad (\text{C.8})$$

Computing e_i naively has complexity of order $\mathcal{O}(N^i)$ for $i \ll N$. Using Newton's identities

$$e_1 = p_1, e_2 = \frac{1}{2}(e_1 p_1 - p_2), e_3 = \frac{1}{3}(-e_1 p_2 + e_2 p_1 + p_3), e_4 = \frac{1}{4}(e_1 p_3 - e_2 p_2 + e_3 p_1 - p_4)$$

e_i can be expressed in terms of p_k , $k \leq i$ each of which can be computed with complexity of order $\mathcal{O}(N)$.

We consider the term $B = \frac{N}{n} \sum_{\tau_i}^{X_{\tau_i}}$ where τ_i are sampled with replacement from a fixed data set $\{1, \dots, N\}$. The second moment can be calculated as follows

$$\begin{aligned} \mathbb{E}B^2 &= \left(\frac{N}{n^2 \sigma_x^2} \right)^2 \sum_{i,j} \mathbb{E}X_{\tau_i} X_{\tau_j} \\ &= \left(\frac{N}{n^2 \sigma_x^2} \right)^2 \left(n(n-1) \underbrace{\mathbb{E}X_{\tau_1} X_{\tau_2}}_{\text{Mom}_{2,1}} + n \underbrace{\mathbb{E}X_{\tau_1}^2}_{\text{Mom}_{2,2}} \right). \end{aligned}$$

We use Newton's identities to express $\text{Mom}_{2,1}$ and $\text{Mom}_{2,2}$

$$\text{Mom}_{2,1} = \frac{2e_2}{N(N-1)} \quad \text{Mom}_{2,2} = \frac{p_2}{N}.$$

Similarly, we obtain

$$\begin{aligned} \mathbb{E}B^3 &= \left(\frac{N}{n^2 \sigma_x^2} \right)^3 \sum_{i,j,k} \mathbb{E}X_{\tau_i} X_{\tau_j} X_{\tau_k} \\ &= \left(\frac{N}{n^2 \sigma_x^2} \right)^3 \left(n(n-1)(n-2) \underbrace{\mathbb{E}X_{\tau_1} X_{\tau_2} X_{\tau_3}}_{\text{Mom}_{3,1}} + 3n(n-1) \underbrace{\mathbb{E}X_{\tau_1} X_{\tau_2}^2}_{\text{Mom}_{3,2}} + n \underbrace{\mathbb{E}X_{\tau_1}^3}_{\text{Mom}_{3,3}} \right) \\ \text{Mom}_{3,1} &= \frac{\sum_{i \neq j \neq k} X_i X_j X_k}{N(N-1)(N-2)} = \frac{6e_3}{N(N-1)(N-2)} \\ \text{Mom}_{3,2} &= \frac{p_1 p_2 - p_3}{N(N-1)}, \quad \text{Mom}_{3,3} = \frac{p_3}{N}. \end{aligned}$$

A similar calculation yields a representation of $\mathbb{E}B^4$ in terms of p_1, \dots, p_4 .

10.2 Expected MSE for Random Data

We sketch the derivation of the MSE

$$\mathbb{E}_{\theta, X} \left(\frac{1}{M} \sum_{k=0}^{M-1} \theta_j^2 - (\mu_p^2 + \sigma_p^2) \right)^2$$

where we take expectation with respect to $X_i \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}(\theta^i, \sigma_x^2)$ for $i = 1, \dots, N$ and the randomness in the recursion for θ_j . We obtain an analytic expression for the MSE by deriving expressions for $\mathbb{E}_{X, \theta} \theta_j^p$. We illustrate the computation for $p = 2$, noting that we assume $\theta_0 = 0$ a.s.. We know that

$$\begin{aligned} \theta_j^2 &= \sum_{k=0}^{j-1} (1-Ah)^{2k} \left[h^2 B_{j-1-k}^2 + h \eta_{j-1-k}^2 + 2(1-Ah) \eta_{j-1-k} \theta_{j-1-k} \sqrt{h} \right] \\ &\quad + 2h^{\frac{3}{2}} B_{j-1-k} \eta_{j-1-k} + 2(1-Ah) B_{j-1-k} h \theta_{j-1-k} \\ &= \sum_{k=0}^{j-1} (1-Ah)^{2k} \left[h^2 B_{j-1-k}^2 + 2(1-Ah) \eta_{j-1-k} \theta_{j-1-k} \sqrt{h} + 2h^{\frac{3}{2}} B_{j-1-k} \eta_{j-1-k} \right. \\ &\quad \left. + 2h B_{j-1-k-2-l} + \sqrt{h} \eta_{j-k-2-l} \right] \sum_{l=0}^{j-1-k-1} (1-Ah)^l \left(h B_{j-k-2-l} + \sqrt{h} \eta_{j-k-2-l} \right). \end{aligned} \quad (\text{C.9})$$

The expectation $\mathbb{E}_{X, \theta} \theta_j$ therefore boils down to calculating $\mathbb{E}B^2$, $\mathbb{E}B'$ and $\mathbb{E}B$ where B and B' are independent samples of Equation (15). We start by calculating

$$\begin{aligned} \mathbb{E}BB' &= \frac{N^2}{n^2 4\sigma_x^4} \sum_{i=1}^n \sum_{j=1}^n \mathbb{E}X_{\tau_i} X_{\tau_j} \\ &= \frac{N^2}{n^2 4\sigma_x^4} \sum_{i=1}^n \sum_{j=1}^n \left(\frac{1}{N} \mathbb{E}X^2 + \frac{N-1}{N} \mathbb{E}X\bar{X} \right) \\ &= \frac{N^2}{n^2 4\sigma_x^4} \left(\sum_{i=1}^n \sum_{j=1}^n \frac{1}{N} (\theta^{i^2} + \sigma_x^2) + \frac{N-1}{N} \theta^{i^2} \right). \end{aligned}$$

Similarly, we obtain

$$\mathbb{E}B^2 = \frac{N^2}{n^2 4\sigma_x^4} \frac{n(\theta^{i^2} + \sigma_\theta^2) + n(n-1)\theta^{i^2}}{1}.$$

Deriving $\mathbb{E}B_j^p$ for $p = 1, 2, 3, 4$ requires the calculation of $\mathbb{E}B_1^{\alpha_1} B_3^{\alpha_2} B_4^{\alpha_3} B_4^{\alpha_4}$ where B_i are i.i.d. following the distribution of Equation (15) for $\alpha_i \geq 0$ and $\sum_i \alpha_i \leq 4$. The arguments so far allow us to derive T_1 and T_3 in

$$\begin{aligned} MSE &= \mathbb{E} \left(S_2^2 - 2S_2(\mu_p^2 + \sigma_p^2) + (\mu_p^2 + \sigma_p^2)^2 \right) \\ &= \underbrace{\mathbb{E}S_2^2}_{T_1} - 2 \underbrace{\mathbb{E}S_2 \mu_p^2}_{T_2} - 2 \underbrace{\mathbb{E}S_2 \sigma_p^2}_{T_3} + \underbrace{\mathbb{E}(\mu_p^4 + 2\mu_p^2 \sigma_p^2 + \sigma_p^4)}_{T_4}. \end{aligned}$$

Recall that the posterior for this toy model is given by

$$\mathcal{N}(\mu_p, \sigma_p^2) = \mathcal{N} \left(\frac{\sum_{i=1}^N X_i}{\sigma_\theta^2 + N}, \left(\frac{1}{\sigma_\theta^2} + \frac{N}{\sigma_x^2} \right)^{-1} \right)$$

and hence T_1 can be computed explicitly. The summands of T_2 can be derived similarly to Equation (C.9) in terms of the quantities $\mathbb{E}B_j \mu_p^2$, $\mathbb{E}B_j^2 \mu_p^2$ and $\mathbb{E}BB_j \mu_p^2$. The explicit expression can be obtained from the supplemented Mathematica® file.

References

- A. Abdulle, D. Cohen, G. Vilmart, and K. C. Zygalakis. High order weak methods for stochastic differential equations based on modified equations. *SIAM J. Sci. Comput.*, 34(3):1800–1823, 2012.
- A. Abdulle, G. Vilmart, and K. Zygalakis. High order numerical approximation of the invariant measure of ergodic sdes. *SIAM Journal on Numerical Analysis*, 52(4):1600–1622, 2014.
- Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, 2011.
- T. Chen, E. Fox, and C. Guestrin. Stochastic gradient hamiltonian monte carlo. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1683–1691, 2014.
- A. Debussche and E. Faou. Weak backward error analysis for SDEs. *SIAM J. Numer. Anal.*, 50(3):1735–1752, 2012. ISSN 0036-1429.
- N. Ding, Y. Fang, R. Babbush, C. Chen, R. Skeel, and H. Neven. Bayesian sampling using stochastic gradient thermostats. *Advances in Neural Information Processing Systems (NIPS)*, Montreal, Canada, 2014.
- Rafal Hasminskii. *Stochastic stability of differential equations*. Sjthoff and Noordhoff, 1980.
- M. Kopeck. Weak backward error analysis for overdamped langevin processes. *IMA Journal of Numerical Analysis*, 2014.
- M. Kopeck. Weak backward error analysis for langevin process. *BIT Numerical Mathematics*, pages 1–47, 2015.
- D. Lamberton and G. Pages. Recursive computation of the invariant distribution of a diffusion. *Bernoulli*, 8(3):367–405, 2002.
- B. Leimkuhler and C. Matthews. Rational construction of stochastic numerical methods for molecular sampling. *Applied Mathematics Research eXpress*, 2013(1):34–56, 2013.
- J. C. Mattingly, A. M. Stuart, and M. V. Tretyakov. Convergence of Numerical Time-Averaging and Stationary Measures via Poisson Equations. *SIAM J. Numer. Anal.*, 48(2):552–577, 2010. ISSN 0036-1429.
- G. N. Milstein and M. V. Tretyakov. Computing ergodic limits for Langevin equations. *Phys. D*, 229(1):81–95, 2007. ISSN 0167-2789.
- G.N. Milstein. Weak approximation of solutions of systems of stochastic differential equations. *Theory Probab. Appl.*, 30(4):750–766, 1986. ISSN 0040585X.
- G.N. Milstein and M.V. Tretyakov. *Stochastic numerics for mathematical physics*. Scientific Computing. Springer-Verlag, Berlin and New York, 2004.
- B-R Nawaf and E. Vanden-Eijnden. Pathwise accuracy and ergodicity of metropolized integrators for SDEs. *Communications on Pure and Applied Mathematics*, 63(5):655–696, 2010.
- R. M. Neal. Markov chain monte carlo using hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo*. Chapman & Hall/CRC, 2011.
- E. Pardoux and A. Yu. Veretennikov. On the Poisson equation and diffusion approximation. I. *The Annals of Probability*, 29(3):1061–1085, 2001.
- S. Paterson and Y. W. Teh. Stochastic gradient Riemannian Langevin dynamics on the probability simplex. In *Advances in Neural Information Processing Systems*, 2013.
- G. O. Roberts and R. L. Tweedie. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4):pp. 341–363, 1996.

- A. Koratticara S. Ahn and M. Welling. Bayesian posterior sampling via stochastic gradient fisher scoring. In *ICML*, 2012.
- I. Sato and H. Nakagawa. Approximation analysis of stochastic gradient langevin dynamics by using fokker-planck equation and ito process. In Tony Jebara and Eric P. Xing, editors, *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 982–990. JMLR Workshop and Conference Proceedings, 2014.
- D. Talay and L. Tubaro. Expansion of the global error for numerical schemes solving stochastic differential equations. *Stochastic Anal. Appl.*, 8(4):483–509 (1991), 1990. ISSN 0736-2994.
- Y. W. Teh, A. H. Thiéry, and S. J. Vollmer. Consistency and fluctuations for stochastic gradient Langevin dynamics. *ArXiv e-prints*, 2014. Accepted by J. Mach. Learn. Res.
- M. Welling and Y. W. Teh. Bayesian learning via Stochastic Gradient Langevin Dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 681–688, 2011.
- K. C. Zygalakis. On the existence and applications of modified equations for stochastic differential equations. *SIAM J. Sci. Comput.*, 33:102–130, 2011.

A General Framework for Constrained Bayesian Optimization using Information-based Search

José Miguel Hernández-Lobato^{1,*}

Michael A. Gelbart^{3,*}

Ryan P. Adams^{1,4}

Matthew W. Hoffman²

Zoubin Ghahramani²

1. School of Engineering and Applied Sciences

Harvard University, Cambridge, MA 02138, USA

2. Department of Engineering

Cambridge University, Cambridge, CB2 1PZ, UK

3. Department of Computer Science

The University of British Columbia, Vancouver, BC, V6T 1Z4, Canada

4. Twitter

Cambridge, MA 02139, USA

JMH@SEAS.HARVARD.EDU

MGELBART@CS.UBC.CA

RPA@SEAS.HARVARD.EDU

MWH30@CAM.AC.UK

ZOUBIN@ENG.CAM.AC.UK

1. Introduction

Many real-world optimization problems involve finding a global minimizer of a black-box objective function subject to a set of black-box constraints all being simultaneously satisfied. For example, consider the problem of optimizing the performance of a speech recognition system, subject to the requirement that it operates within a specified time limit. The system may be implemented as a neural network with hyper-parameters such as the number of hidden units, learning rates, regularization constants, etc. These hyper-parameters have to be tuned to minimize the recognition error on some validation data under a constraint on the maximum runtime of the resulting system. Another example is the discovery of new materials. Here, we aim to find new molecular compounds with optimal properties such as the power conversion efficiency in photovoltaic devices. Constraints arise from our ability (or inability) to synthesize various molecules. In this case, the estimation of the properties of the molecule and its synthesizability can be achieved by running expensive simulations on a computer.

More formally, we are interested in finding the global minimum \mathbf{x}_* of a scalar objective function $f(\mathbf{x})$ over some bounded domain, typically $\mathcal{X} \subset \mathbb{R}^D$, subject to the non-negativity of a set of constraint functions c_1, \dots, c_K . We write this as

$$\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}) \quad \text{s.t.} \quad c_1(\mathbf{x}) \geq 0, \dots, c_K(\mathbf{x}) \geq 0. \quad (1)$$

However, f and c_1, \dots, c_K are unknown and can only be evaluated pointwise via expensive queries to “black boxes” that may provide noise-corrupted values. Note that we are assuming that f and each of the constraints c_k are defined over the entire space \mathcal{X} . We seek to find a solution to Eq. (1) with as few queries as possible.

For solving unconstrained problems, Bayesian optimization (BO) is a successful approach to the efficient optimization of black-box functions (Mockus et al., 1978). BO methods work by applying a Bayesian model to the previous evaluations of the function, with the aim of reasoning about the global structure of the objective function. The Bayesian model is then used to compute an *acquisition function* (i.e., expected utility function) that represents how promising each possible $\mathbf{x} \in \mathcal{X}$ is if it were to be evaluated next. Maximizing the acquisition function produces a *suggestion* which is then used as the next evaluation location. When the evaluation of the objective at the suggested point is complete, the Bayesian model is updated with the newly collected function observation and the process repeats. The optimization ends after a maximum number of function evaluations is reached, a time threshold is exceeded, or some other stopping criterion is met. When this occurs, a *recommendation* of the solution is given to the user. This is achieved for example by optimizing the predictive mean of the Bayesian model, or by choosing the best observed point among the evaluations. The Bayesian model is typically a Gaussian process (GP); an in-depth treatment of GPs is given by Rasmussen and Williams (2006). A commonly-used acquisition function is the expected improvement (EI) criterion (Jones et al., 1998), which measures the expected amount by which we will improve over some *incumbent* or current best solution, typically given by the expected value of the objective at the current best recommendation. Other acquisition functions aim to approximate the expected information gain or expected reduction in the posterior entropy of the global minimizer of the objective

Abstract

We present an information-theoretic framework for solving global black-box optimization problems that also have black-box constraints. Of particular interest to us is to efficiently solve problems with *decoupled* constraints, in which subsets of the objective and constraint functions may be evaluated independently. For example, when the objective is evaluated on a CPU and the constraints are evaluated independently on a GPU. These problems require an acquisition function that can be separated into the contributions of the individual function evaluations. We develop one such acquisition function and call it Predictive Entropy Search with Constraints (PESC). PESC is an approximation to the expected information gain criterion and it compares favorably to alternative approaches based on improvement in several synthetic and real-world problems. In addition to this, we consider problems with a mix of functions that are fast and slow to evaluate. These problems require balancing the amount of time spent in the meta-computation of PESC and in the actual evaluation of the target objective. We take a bounded rationality approach and develop a partial update for PESC which trades off accuracy against speed. We then propose a method for adaptively switching between the partial and full updates for PESC. This allows us to interpolate between versions of PESC that are efficient in terms of function evaluations and those that are efficient in terms of wall-clock time. Overall, we demonstrate that PESC is an effective algorithm that provides a promising direction towards a unified solution for constrained Bayesian optimization.

Keywords: Bayesian optimization, constraints, predictive entropy search

*. Authors contributed equally.

(Villemonteix et al., 2009; Hennig and Schuler, 2012; Hernández-Lobato et al., 2014). For more information on BO, we refer to the tutorial by Brochu et al. (2010).

There have been several attempts to extend BO methods to address the constrained optimization problem in Eq. (1). The proposed techniques use GPs and variants of the EI heuristic (Schonlau et al., 1998; Parr, 2013; Snoek, 2013; Gelbart et al., 2014; Gardner et al., 2014; Gramacy et al., 2016; Gramacy and Lee, 2011; Picheny, 2014). Some of these methods lack generality since they were designed to work in specific contexts, such as when the constraints are noiseless or the objective is known. Furthermore, because they are based on EI, computing their acquisition function requires the current best feasible solution or incumbent: a location in the search space with low expected objective value and high probability of satisfying the constraints. However, the best feasible solution does not exist when no point in the search space satisfies the constraints with high probability (for example, because of lack of data). Finally and more importantly, these methods run into problems when the objective and the constraint functions are *decoupled*, meaning that the functions f, c_1, \dots, c_r in Eq. (1) can be evaluated independently. In particular, the acquisition functions used by these methods usually consider joint evaluations of the objective and constraints and cannot produce an optimal suggestion when only subsets of these functions are being evaluated.

In this work, we propose a general approach for constrained BO that does not have the problems mentioned above. Our approach to constraints is based on an extension of Predictive Entropy Search (PES) (Hernández-Lobato et al., 2014), an information-theoretic method for unconstrained BO problems. The resulting technique is called Predictive Entropy Search with Constraints (PESC) and its acquisition function approximates the expected information gain with regard to the solution of Eq. (1), which we call \mathbf{x}_* . At each iteration, PESC collects data at the location that is the most informative about \mathbf{x}_* , in expectation. One important property of PESC is that its acquisition function naturally separates the contributions of the individual function evaluations when those functions are modeled independently. That is, the amount of information that we approximately gain by jointly evaluating a set of independent functions is equal to the sum of the gains of information that we approximately obtain by the individual evaluation of each of the functions. This additive property in its acquisition function allows PESC to efficiently solve the general constrained BO problem, including those with decoupled evaluation, something that no other existing technique can achieve, to the best of our knowledge.

An initial description of PESC is given by Hernández-Lobato et al. (2015). That work considers PESC only in the coupled evaluation scenario, where all the functions are jointly evaluated at the same input value. This is the standard setting considered by most prior approaches for constrained BO. Here, we further extend that initial work on PESC as follows:

1. We present a taxonomy of constrained BO problems. We consider problems in which the objective and constraints can be split into subsets of functions or *tasks* that require coupled evaluation, but where different tasks can be evaluated in a decoupled way. These different tasks may or may not compete for a limited set of resources. We propose a general algorithm for solving this type of problems and then show how PESC can be used for the practical implementation of this algorithm.

2. We analyze PESC in the decoupled scenario. We evaluate the accuracy of PESC when the different functions (objective or constraint) are evaluated independently. We show how PESC efficiently solves decoupled problems with an objective and constraints that compete for the same computational resource.

3. We intelligently balance the computational overhead of the Bayesian optimization method relative to the cost of evaluating the black-boxes. To achieve this, we develop a partial update to the PESC approximation that is less accurate but faster to compute. We then automatically switch between partial and full updates so that we can balance the amount of time spent in the Bayesian optimization subroutines and in the actual collection of data. This allows us to efficiently solve problems with a mix of decoupled functions where some are fast and others slow to evaluate.

The rest of the paper is structured as follows. Section 2 reviews prior work on constrained BO and considers these methods in the context of decoupled functions. In Section 3 we present a general framework for describing BO problems with decoupled functions, which contains as special cases the standard coupled framework considered in most prior work as well as the notion of decoupling introduced by Gelbart et al. (2014). This section also describes a general algorithm for BO problems with decoupled functions. In Section 4 we show how to extend Predictive Entropy Search (PES) (Hernández-Lobato et al., 2014) to solve Eq. (1) in the context of decoupling, an approach that we call Predictive Entropy Search with Constraints (PESC). We also show how PESC can be used to implement the general algorithm from Section 3. In Section 5 we modify the PESC algorithm to be more efficient in terms of wall-clock time by adaptively using an approximate but faster version of the method. In Sections 6 and 7 we perform empirical evaluations of PESC on coupled and decoupled optimization problems, respectively. Finally, we conclude in Section 8.

2. Related Work

Below we discuss previous approaches to Bayesian optimization with black-box constraints, many of which are variants of the expected improvement (EI) heuristic (Jones et al., 1998). In the unconstrained setting, EI measures the expected amount by which observing the objective f at \mathbf{x} leads to improvement over the current best recommendation or *incumbent*, the objective value of which is denoted by η (thus, η has the units of f , not \mathbf{x}). The incumbent η is usually defined as the lowest expected value for the objective over the optimization domain. The EI acquisition function is then given by

$$e_{\text{EI}}(\mathbf{x}) = \int \max(0, \eta - f(\mathbf{x}))p(f(\mathbf{x})|\mathcal{D})df(\mathbf{x}) = \sigma f(\mathbf{x})(z_f(\mathbf{x})\Phi(z_f(\mathbf{x})) + \phi(z_f(\mathbf{x}))), \quad (2)$$

where \mathcal{D} represents the collected data (previous function evaluations) and $p(f(\mathbf{x})|\mathcal{D})$ is the predictive distribution for the objective made by a Gaussian process (GP), $\mu_f(\mathbf{x})$ and $\sigma_f^2(\mathbf{x})$ are the GP predictive mean and variance for $f(\mathbf{x})$, $z_f(\mathbf{x}) \equiv (\eta - \mu_f(\mathbf{x}))/\sigma_f(\mathbf{x})$, and Φ and ϕ are the standard Gaussian CDF and PDF, respectively.

2.1 Expected Improvement with Constraints

An intuitive extension of EI in the presence of constraints is to define improvement as only occurring when the constraints are satisfied. Because we are uncertain about the values of the constraints, we must weight the original EI value by the probability of the constraints being satisfied. This results in what we call Expected Improvement with Constraints (EIC):

$$\alpha_{\text{EIC}}(\mathbf{x}) = \alpha_{\text{EI}}(\mathbf{x}) \prod_{k=1}^K \Pr(c_k(\mathbf{x}) \geq 0 | \mathcal{D}) = \alpha_{\text{EI}}(\mathbf{x}) \prod_{k=1}^K \Phi\left(\frac{\mu_k(\mathbf{x})}{\sigma_k(\mathbf{x})}\right), \quad (3)$$

The constraint satisfaction probability factorizes because f and c_1, \dots, c_K are modeled by independent GPs. In this expression μ_k and σ_k^2 are the posterior predictive mean and variance for $c_k(\mathbf{x})$. EIC was initially proposed by Schonlau et al. (1998) and has been revisited by Parr (2013), Snoek (2013), Gardner et al. (2014) and Gelbart et al. (2014).

In the constrained setting, the incumbent η can be defined as the minimum expected objective value subject to all the constraints being satisfied at the corresponding location. However, we can never guarantee that all the constraints will be satisfied when they are only observed through noisy evaluations. To circumvent this problem, Gelbart et al. (2014) define η as the lowest expected objective value subject to all the constraints being satisfied with posterior probability larger than the threshold $1 - \delta$, where δ is a small number such as 0.05. However, this value for η still cannot be computed when there is no point in the search space that satisfies the constraints with posterior probability higher than $1 - \delta$. For example, because of lack of data for the constraints. In this case, Gelbart et al. change the original acquisition function given by Eq. (3) and ignore the factor $\alpha_{\text{EI}}(\mathbf{x})$ in that expression. This allows them to search only for a feasible location, ignoring the objective f entirely and just optimizing the constraint satisfaction probability. However, this can lead to inefficient optimization in practice because the data collected for the objective f is not used to make optimal decisions.

2.2 Integrated Expected Conditional Improvement

Gramacy and Lee (2011) propose an acquisition function called the integrated expected conditional improvement (IECI), defined as

$$\alpha_{\text{IECI}}(\mathbf{x}) = \int_{\mathcal{X}} [\alpha_{\text{EI}}(\mathbf{x}') - \alpha_{\text{EI}}(\mathbf{x} | \mathbf{x})] h(\mathbf{x}') d\mathbf{x}'. \quad (4)$$

Here, $\alpha_{\text{EI}}(\mathbf{x}')$ is the expected improvement at \mathbf{x}' , $\alpha_{\text{EI}}(\mathbf{x} | \mathbf{x})$ is the expected improvement at \mathbf{x}' given that the objective has been observed at \mathbf{x} (but without making any assumptions about the observed value), and $h(\mathbf{x}')$ is an arbitrary density over \mathbf{x}' . The IECI at \mathbf{x} is the expected reduction in EI at \mathbf{x}' , under the density $h(\mathbf{x}')$, caused by observing the objective at \mathbf{x} . Gramacy and Lee use IECI for constrained BO by setting $h(\mathbf{x}')$ to the probability of the constraints being satisfied at \mathbf{x}' . They define the incumbent η as the lowest posterior mean for the objective f over the whole optimization domain, ignoring the fact that the lowest posterior mean for the objective may be achieved in an infeasible location.

The motivation for IECI is that collecting data at an infeasible location may also provide useful information. EIC strongly discourages this, because Eq. (3) always takes very low

values when the constraints are unlikely to be satisfied. This is not the case with IECI because Eq. (4) considers the EI over the whole optimization domain instead of focusing only on the EI at the current evaluation location, which may be infeasible with high probability. Gelbart et al. (2014) compare IECI with EIC for optimizing the hyper-parameters of a topic model with constraints on the entropy of the per-topic word distribution and show that EIC outperforms IECI on this problem.

2.3 Expected Volume Minimization

An alternative approach is given by Picheny (2014), who proposes to sequentially explore the location that most decreases the expected volume (EV) of the feasible region below the best feasible objective value η found so far. This quantity is computed by integrating the product of the probability of improvement and the probability of feasibility. That is,

$$\alpha_{\text{EV}}(\mathbf{x}) = \int p[f(\mathbf{x}') \leq \eta] h(\mathbf{x}') d\mathbf{x}' - \int p[f(\mathbf{x}') \leq \min(\eta, f(\mathbf{x}))] h(\mathbf{x}') d\mathbf{x}', \quad (5)$$

where, as in IECI, $h(\mathbf{x}')$ is the probability that the constraints are satisfied at \mathbf{x}' . Picheny considers noiseless evaluations for the objective and constraint functions and defines η as the best feasible objective value seen so far or $+\infty$ when no feasible location has been found.

A disadvantage of Picheny's method is that it requires the integral in Eq. (5) to be computed over the entire search domain \mathcal{X} , which is done numerically over a grid on \mathbf{x}' . The resulting acquisition function must then be globally optimized. This is often performed by first evaluating the acquisition function on a grid on \mathbf{x} . The best point in this second grid is then used as the starting point of a numerical optimizer for the acquisition function. This nesting of grid operations limits the application of this method to small input dimension D . This is also the case for IECI whose acquisition function in Eq. (4) also includes an integral over \mathcal{X} . Our method PESC requires a similar integral in the form of an expectation with respect to the posterior distribution of the global feasible minimizer \mathbf{x}_* . Nevertheless, this expectation can be efficiently approximated by averaging over samples of \mathbf{x}_* drawn using the approach proposed by Hernández-Lobato et al. (2014). This approach is further described in Appendix B.3. Note that the integrals in Eq. (5) could in principle be also approximated by using Markov chain Monte Carlo (MCMC) to sample from the unnormalized density $h(\mathbf{x}')$. However, this was not proposed by Picheny and he only described the grid based method.

2.4 Modeling an Augmented Lagrangian

Gramacy et al. (2016) propose to use a combination of EI and the augmented Lagrangian (AL) method: an algorithm which turns an optimization problem with constraints into a sequence of unconstrained optimization problems. Gramacy et al. use BO techniques based on EI to solve the unconstrained *inner* loop of the AL problem. When f and c_1, \dots, c_K are known the unconstrained AL objective is defined as

$$L_A(\mathbf{x} | \lambda_1, \dots, \lambda_K, p) = f(\mathbf{x}) + \sum_{k=1}^K \left[\frac{1}{2p} \min(0, c_k(\mathbf{x}))^2 - \lambda_k c_k(\mathbf{x}) \right], \quad (6)$$

where $p > 0$ is a penalty parameter and $\lambda_1 \geq 0, \dots, \lambda_K \geq 0$ serve as Lagrange multipliers. The AL method iteratively minimizes Eq. (6) with different values for p and $\lambda_1, \dots, \lambda_K$ at each iteration. Let $\mathbf{x}_k^{(n)}$ be the minimizer of Eq. (6) at iteration n using parameter values $p^{(n)}$ and $\lambda_1^{(n)}, \dots, \lambda_K^{(n)}$. The next parameter values are $\lambda_k^{(n+1)} = \max(0, \lambda_k^{(n)} - c_k(\mathbf{x}_k^{(n)})/p^{(n)})$ for $k = 1, \dots, K$ and $p^{(n+1)} = p^{(n)}$ if $\mathbf{x}_k^{(n)}$ is feasible and $p^{(n+1)} = p^{(n)}/2$ otherwise. When f and c_1, \dots, c_K are unknown we cannot directly minimize Eq. (6). However, if we have observations for f and c_1, \dots, c_K , we can then map such data into observations for the AL objective. Gramacy et al. fit a GP model to the AL observations and then select the next evaluation location using the EI heuristic. After collecting the data, the AL parameters are updated as above using the new values for the constraints and the whole process repeats.

A disadvantage of this approach is that it assumes that the constraints c_1, \dots, c_K are noiseless to guarantee that that p and $\lambda_1, \dots, \lambda_K$ can be correctly updated. Furthermore, Gramacy et al. (2016) focus only on the case in which the objective f is known, although they provide suggestions for extending their method to unknown f . In section 6.3 we show that PESC and EIC perform better than the AL approach on the synthetic benchmark problem considered by Gramacy et al., even when the AL method has access to the true objective function and PESC and EIC do not.

2.5 Existing Methods for Decoupled Evaluations

The methods described above can be used to solve constrained BO problems with *coupled* evaluations. These are problems in which all the functions (objective and constraints) are always evaluated jointly at the same input. Gelbart et al. (2014) consider extending the EIC method from Section 2.1 to the *decoupled* setting, where the different functions can be independently evaluated at different input locations. However, they identify a problem with EIC in the decoupled scenario. In particular, the EIC utility function requires two conditions to produce positive values. First, the evaluation for the objective must achieve a lower value than the best feasible solution so far and, second, the evaluations for the constraints must produce non-negative values. When we evaluate only one function (objective or constraint), the conjunction of these two conditions cannot be satisfied by a single observation under a myopic search policy. Thus, the new evaluation location can never become the new incumbent and the EIC is zero everywhere. Therefore, standard EIC fails in the decoupled setting.

Gelbart et al. (2014) circumvent the problem mentioned above by treating decoupling as a special case and using a two-stage acquisition function: first, the next evaluation location \mathbf{x} is chosen with EIC, and then, given \mathbf{x} , the task (whether to evaluate the objective or one of the constraints) is chosen according to the expected reduction in the entropy of the global feasible minimizer \mathbf{x}_* , where the entropy computation is approximated using Monte Carlo sampling as proposed by Villemonteix et al. (2009). We call the resulting method EIC-D. Note that the two-stage decision process used by EIC-D is sub-optimal and a joint selection of \mathbf{x} and the task should produce better results. As discussed in the sections that follow, our method, PESC, does not suffer from this disadvantage and furthermore, can be extended to a wider range of decoupled problems than EIC-D can.

3. Decoupled Function Evaluations and Resource Allocation

We present a framework for describing constrained BO problems. We say that a set of functions (objective or constraints) are *coupled* when they always require joint evaluation at the same input location. We say that they are *decoupled* when they can be evaluated independently at different inputs. In practice, a particular problem may exhibit coupled or decoupled functions or a combination of both. An example of a problem with coupled functions is given by a financial simulator that generates many samples from the distribution of possible financial outcomes. If the objective function is the expected profit and the constraint is a maximum tolerable probability of default, then these two functions are computed jointly by the same simulation and are thus coupled to each other. An example of a problem with decoupled functions is the optimization of the predictive accuracy of a neural network speech recognition system subject to prediction time constraints. In this case different neural network architectures may produce different predictive accuracies and different prediction times. Assessing the prediction time may not require training the neural network and could be done using arbitrary network weights. Thus, we can evaluate the timing constraint without evaluating the accuracy objective.

When problems exhibit a combination of coupled and decoupled functions, we can then partition the different functions into subsets of functions that require coupled evaluation. We call these subsets of coupled functions *tasks*. In the financial simulator example, the objective and the constraint form the only task. In the speech recognition system there are two tasks, one for the objective and one for the constraint. Functions within different tasks are decoupled and can be evaluated independently. These tasks may or may not compete for a limited set of *resources*. For example, two tasks that both require the performance of expensive computations may have to compete for using a single available CPU. An example with no competition is given by two tasks, one which performs computations in a GPU and another one which performs computations in a GPU. Finally, two competitive tasks may also have different evaluation costs and this should be taken into account when deciding which one is going to be evaluated next.

In the previous section we showed that most existing methods for constrained BO can only address problems with coupled functions. Furthermore, the extension of these methods to the decoupled setting is difficult because most of them are based on the EI heuristic and, as illustrated in Section 2.5, improvement can be impossible with decoupled evaluations. A decoupled problem can, of course, be coupled artificially and then solved as a coupled problem with existing methods. We examine this approach here with a thought experiment and with empirical evaluations in Section 7. Returning to our time-limited speech recognition system, let us consider the cost of evaluating each of the tasks. Evaluating the objective requires training the neural network, which is a very expensive process. On the other hand, evaluating the constraint (run time) requires only to time the predictions made by the neural network and this could be done without training, using arbitrary network weights. Therefore, evaluating the constraint is in this case much faster than evaluating the objective. In a decoupled framework, one could first measure the run time at several evaluation locations, gaining a sense of the constraint surface. Only then would we incur the significant expense of evaluating the objective task, heavily biasing our search towards locations that are considered to be feasible with high probability. Put another way, arti-

cially coupling the tasks becomes increasingly inefficient as the cost differential is increased; for example, one might spend a week examining one aspect of a design that could have been ruled out within seconds by examining another aspect.

In the following sections we present a formalization of constrained Bayesian optimization problems that encompasses all of the cases described above. We then show that our method, PESC (Section 4), is an effective practical solution to these problems because it naturally separates the contributions of the different function evaluations in its acquisition function.

3.1 Competitive Versus Non-competitive Decoupling and Parallel BO

We divide the class of problems with decoupled functions into two sub-classes, which we call *competitive decoupling* (CD) and *non-competitive decoupling* (NCD). CD is the form of decoupling considered by Gelbart et al. (2014), in which two or more tasks compete for the same resource. This happens when there is only one CPU available and the optimization problem includes two tasks with each of them requiring a CPU to perform some expensive simulations. In contrast, NCD refers to the case in which tasks require the use of different resources and can therefore be evaluated independently, in parallel. This occurs, for example, when one of the two tasks uses a CPU and the other task uses a GPU.

Note that NCD is very related to *parallel* Bayesian optimization (see e.g., Ginsbourger et al., 2011; Snoek et al., 2012). In both parallel BO and NCD we perform multiple task evaluations concurrently, where each new evaluation location is selected optimally according to the available data and the locations of all the currently pending evaluations. The difference between parallel BO and NCD is that in NCD the tasks whose evaluations are currently pending may be different from the task that will be evaluated next, while in parallel BO there is only a single task. Parallel BO conveniently fits into the general framework described below.

3.2 Formalization of Constrained Bayesian Optimization Problems

We now present a framework for describing constrained BO problems of the form given by Eq. (1). Our framework can be used to represent general problems within any of the categories previously described, including coupled and decoupled functions that may or may not compete for a limited number of resources, each of which may be replicated multiple times. Let \mathcal{F} be the set of functions $\{f, c_1, \dots, c_K\}$ and let the set of tasks \mathcal{T} be a partition of \mathcal{F} indicating which functions are coupled and must be jointly evaluated. Let $\mathcal{R} = \{r_1, \dots, r_{|\mathcal{R}|}\}$ be the set of resources available to solve this problem. We encode the relationship between tasks and resources with a bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with vertices $\mathcal{V} = \mathcal{T} \cup \mathcal{R}$ and edges $\{t \sim r\} \in \mathcal{E}$ such that $t \in \mathcal{T}$ and $r \in \mathcal{R}$. The interpretation of an edge $\{t \sim r\}$ is that task t can be performed on resource r . (We do not address the case in which a task requires multiple resources to be executed; we leave this as future work.) We also introduce a *capacity* ω_{\max} for each resource r . The capacity $\omega_{\max}(r) \in \mathbb{N}$ represents how many tasks may be simultaneously executed on resource r ; for example, if r represents a cluster of CPUs, $\omega_{\max}(r)$ would be the number of CPUs in the cluster. Introducing the notion of capacity is simply a matter of convenience since it is equivalent to setting all capacities to one and replicating each resource node in \mathcal{G} according to its capacity.

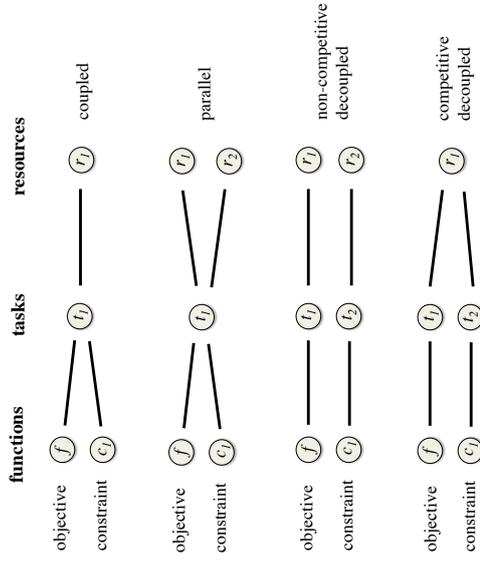


Figure 1: Schematic comparing the coupled, parallel, non-competitive decoupled (NCD), and competitive decoupled (CD) scenarios for a problem with a single constraint c_1 . In each case, the mapping between tasks and resources (the right-hand portion of the figure) is the bipartite graph \mathcal{G} .

We can now formally describe problems with coupled evaluations as well as NCD and CD. In particular, coupling occurs when two functions g_1 and g_2 belong to the same task t . If this task can be evaluated on multiple resources (or one resource with $\omega_{\max} > 1$), then this is parallel Bayesian optimization. NCD occurs when two functions g_1 and g_2 belong to different tasks t_1 and t_2 , which themselves require different resources r_1 and r_2 , (that is, $t_1 \sim r_1, t_2 \sim r_2$ and $r_1 \neq r_2$). CD occurs when two functions g_1 and g_2 belong to *different* tasks t_1 and t_2 (decoupled) that require the *same* resource r (competitive). These definitions are visually illustrated in Fig. 1. The definitions can be trivially extended to cases with more than two functions. The most general case is an arbitrary task-resource graph \mathcal{G} encoding a combination of coupling, NCD, CD and parallel Bayesian optimization.

3.3 A General Algorithm for Constrained Bayesian Optimization

In this section we present a general algorithm for solving constrained Bayesian optimization problems specified according to the formalization from the previous section. Our approach relies on an acquisition function that can measure the expected utility of evaluating any arbitrary subset of functions, that is, of any possible task. When an acquisition function satisfies this requirement we say that it is *separable*. As discussed in Section 4.1, our

Algorithm 1 A general method for constrained Bayesian optimization.

- 1: **Input:** $\mathcal{F}, \mathcal{G} = (\mathcal{T} \cup \mathcal{R}, \mathcal{E}), \alpha_t$ for $t \in \mathcal{T}, \mathcal{K}, \mathcal{M}, \mathcal{D}, \omega, \omega_{\max}$ and δ .
 - 2: **repeat**
 - 3: **for** $r \in \mathcal{R}$ such that $\omega(r) < \omega_{\max}(r)$ **do**
 - 4: Update \mathcal{M} with any new data in \mathcal{D}
 - 5: **for** $t \in \mathcal{T}$ such that $\{t \sim r\} \in \mathcal{E}$ **do**
 - 6: $\mathbf{x}_t^* \leftarrow \arg \max_{\mathbf{x} \in \mathcal{X}} \alpha_t(\mathbf{x} | \mathcal{M})$
 - 7: $\alpha_t^* \leftarrow \alpha_t(\mathbf{x}_t^* | \mathcal{M})$
 - 8: **end for**
 - 9: $t^* \leftarrow \arg \max_r \alpha_t^*$
 - 10: Submit task t^* at input \mathbf{x}_t^* to resource r
 - 11: Update \mathcal{M} with the new pending evaluation
 - 12: **end for**
 - 13: **until** termination condition is met
 - 14: **Output:** $\arg \min_{\mathbf{x} \in \mathcal{X}} \mathbb{E}_{\mathcal{M}}[f(\mathbf{x})]$ s.t. $p(c_1(\mathbf{x}) \geq 0, \dots, c_K(\mathbf{x}) \geq 0 | \mathcal{M}) \geq 1 - \delta$
-

method PESG has this property; when the functions are modeled as independent. This property makes PESG an effective solution for the practical implementation of our general algorithm. By contrast, the EIG-D method of Gelbart et al. (2014) is not separable and cannot be applied in the general case presented here.

Algorithm 1 provides a general procedure for solving constrained Bayesian optimization problems. The inputs to the algorithm are the set of functions \mathcal{F} , the set of tasks \mathcal{T} , the set of resources \mathcal{R} , the task-resource graph $\mathcal{G} = (\mathcal{T} \cup \mathcal{R}, \mathcal{E})$, an acquisition function for each task, that is, α_t for $t \in \mathcal{T}$, the search space \mathcal{X} , a Bayesian model \mathcal{M} , the initial data set \mathcal{D} , the resource query functions ω and ω_{\max} and a confidence level δ for making a final recommendation. Recall that ω_{\max} indicates how many tasks can be simultaneously executed on a particular resource. The function ω is introduced here to indicate how many tasks are currently being evaluated in a resource. The acquisition function α_t measures the utility of evaluating task t at the location \mathbf{x} . This acquisition function depends on the predictions of the Bayesian model \mathcal{M} . The separability property of the original acquisition function guarantees that we can compute an α_t for each $t \in \mathcal{T}$.

Algorithm 1 works as follows. First, in line 3, we iterate over the resources, checking if they are available. Resource r is available if its number of currently running jobs $\omega(r)$ is less than its capacity $\omega_{\max}(r)$. Whenever resource r is available, we check in line 4 if any new function observations have been collected. If this is the case, we then update the Bayesian model \mathcal{M} with the new data (in most cases we will have new data since the resource r probably became available due to the completion of a previous task). Next, we iterate in line 5 over the tasks t that can be evaluated in the new available resource r as dictated by \mathcal{G} . In line 6 we find the evaluation location \mathbf{x}_t^* that maximizes the utility obtained by the evaluation of task t , as indicated by the task-specific acquisition function α_t . In line 7 we obtain the corresponding maximum task utility α_t^* . In line 9, we then maximize over tasks, selecting the task t^* with highest maximum task utility α_t^* (this is the “competition” in CD). Upon doing so, the pair $(t^*, \mathbf{x}_{t^*}^*)$ forms the next *suggestion*. This pair represents the experiment with the highest acquisition function value over all possible

(t, \mathbf{x}) pairs in $\mathcal{T} \times \mathcal{X}$ that can be run on resource r . In line 10, we evaluate the selected task at resource r and in line 11 we update the Bayesian model \mathcal{M} to take into account that we are expecting to collect data for task t^* at input $\mathbf{x}_{t^*}^*$. This can be done for example by drawing virtual data from \mathcal{M} 's predictive distribution and then averaging across the predictions made when each virtual data point is assumed to be the data actually collected by the pending evaluation (Schonlau et al., 1998; Snoek et al., 2012). In line 13 the whole process repeats until a termination condition is met. Finally, in line 14, we give to the user a final *recommendation* of the solution to the optimization problem. This is the input that attains the lowest expected objective value subject to all the constraints being satisfied with posterior probability larger than $1 - \delta$, where δ is maximum allowable probability of the recommendation being infeasible according to \mathcal{M} .

Algorithm 1 can solve problems that exhibit any combination of coupling, parallelism, NCD, and CD.

3.4 Incorporating Cost Information

Algorithm 1 always selects, among a group of competitive tasks, the one whose evaluation produces the highest utility value. However, other cost factors may render the evaluation of one task more desirable than another. The most salient of these costs is the run time or duration of the task’s evaluation, which could depend on the evaluation location \mathbf{x} . For example, in the neural network speech recognition system, one of the variables to be optimized may be the number of hidden units in the neural network. In this case, the run time of an evaluation of the predictive accuracy of the system is a function of \mathbf{x} since the training time for the network scales with its size. Snoek et al. (2012) consider this issue by automatically measuring the duration of function evaluations. They model the duration as a function of \mathbf{x} with an additional Gaussian process (GP). Swersky et al. (2013) extend this concept over multiple optimization tasks so that an independent GP is used to model the unknown duration of each task. This approach can be applied in Algorithm 1 by penalizing the acquisition function for task t with the expected cost of evaluating that task. In particular, we can change lines 6 and 7 in Algorithm 1 to

- 6: $\mathbf{x}_t^* \leftarrow \arg \max_{\mathbf{x} \in \mathcal{X}} \alpha_t(\mathbf{x} | \mathcal{M}) / \zeta_t(\mathbf{x})$
- 7: $\alpha_t^* \leftarrow \alpha_t(\mathbf{x}_t^* | \mathcal{M}) / \zeta_t(\mathbf{x}_t^*)$

where $\zeta_t(\mathbf{x})$ is the expected cost associated with the evaluation of task t at \mathbf{x} : as estimated by a model of the collected cost data. When taking into account task costs modeled by Gaussian processes, the total number of GP models used by Algorithm 1 is equal to the number of functions in the constrained BO problem plus the number of tasks, that is, $|\mathcal{T}| + |\mathcal{T}|$. Alternatively, one could fix the cost functions $\zeta_t(\mathbf{x})$ *a priori* instead of learning them from collected data.

4. Predictive Entropy Search with Constraints (PESG)

To implement Algorithm 1 in practice we need to compute an acquisition function that is separable and can measure the utility of evaluating an arbitrary subset of functions. In this section we describe how to achieve this.

Our acquisition function approximates the expected gain of information about the solution to the constrained optimization problem, which is denoted by \mathbf{x}_* . Importantly, our approximation is additive. For example, let \mathcal{A} be a set of functions and let $I(\mathcal{A})$ be the amount of information that we approximately gain in expectation by jointly evaluating the functions in \mathcal{A} . Then $I(\mathcal{A}) = \sum_{a \in \mathcal{A}} I(\{a\})$. Although our acquisition function is additive, the exact expected gain of information is not. Additivity is the result of a factorization assumption in our approximation (see Section 4.2 for further details). The good results obtained in our experiments seem to support that this is a reasonable assumption. Because of this additive property, we can compute an acquisition function for any possible subset of f, c_1, \dots, c_K using the individual acquisition functions for these functions as building blocks.

We follow MacKay (1992) and measure information about \mathbf{x}_* by the differential entropy of $p(\mathbf{x}_*|\mathcal{D})$, where \mathcal{D} is the data collected so far. The distribution $p(\mathbf{x}_*|\mathcal{D})$ is formally defined in the unconstrained case by Hennig and Schuler (2012). In the constrained case $p(\mathbf{x}_*|\mathcal{D})$ can be understood as the probability distribution determined by the following sampling process. First, we draw f, c_1, \dots, c_K from their posterior distributions given \mathcal{D} and second, we minimize the sampled f subject to the sampled c_1, \dots, c_K being non-negative, that is, we solve Eq. (1) for the sampled functions. The solution to Eq. (1) obtained by this procedure represents then a sample from $p(\mathbf{x}_*|\mathcal{D})$.

We consider first the case in which all the black-box functions f, c_1, \dots, c_K are evaluated at the same time (coupled). Let $\mathbb{H}[\mathbf{x}_*|\mathcal{D}]$ denote the differential entropy of $p(\mathbf{x}_*|\mathcal{D})$ and let $y_f, y_{c_1}, \dots, y_{c_K}$ denote the measurements obtained by querying the black-boxes for f, c_1, \dots, c_K at the input location \mathbf{x} . We encode these measurements in vector form as $\mathbf{y} = (y_f, y_{c_1}, \dots, y_{c_K})^T$. Note that \mathbf{y} contains the result of the evaluation of all the functions at \mathbf{x} , that is, the objective f and the constraints c_1, \dots, c_K . We aim to collect data at the location that maximizes the expected information gain or the expected reduction in the entropy of $p(\mathbf{x}_*|\mathcal{D})$. The corresponding acquisition function is

$$\alpha(\mathbf{x}) = \mathbb{H}[\mathbf{x}_*|\mathcal{D}] - \mathbb{E}_{\mathbf{y}|\mathcal{D}, \mathbf{x}}[\mathbb{H}[\mathbf{x}_*|\mathcal{D} \cup \{(\mathbf{x}, \mathbf{y})\}]]. \quad (7)$$

In this expression, $\mathbb{H}[\mathbf{x}_*|\mathcal{D} \cup \{(\mathbf{x}, \mathbf{y})\}]$ is the amount of information on \mathbf{x}_* that is available once we have collected new data \mathbf{y} at the input location \mathbf{x} . However, this new \mathbf{y} is unknown because it has not been collected yet. To circumvent this problem, we take the expectation with respect to the predictive distribution for \mathbf{y} given \mathbf{x} and \mathcal{D} . This produces an expression that does not depend on \mathbf{y} and could in principle be readily computed.

A direct computation of Eq. (7) is challenging because it requires evaluating the entropy of the intractable distribution $p(\mathbf{x}_*|\mathcal{D})$ when different pairs (\mathbf{x}, \mathbf{y}) are added to the data. To simplify computations, we note that Eq. (7) is the mutual information between \mathbf{x}_* and \mathbf{y} given \mathcal{D} and \mathbf{x} , which we denote by $\text{MI}(\mathbf{x}_*, \mathbf{y})$. The mutual information operator is symmetric, that is, $\text{MI}(\mathbf{x}_*, \mathbf{y}) = \text{MI}(\mathbf{y}, \mathbf{x}_*)$. Therefore, we can follow Houthby et al. (2012) and swap the random variables \mathbf{y} and \mathbf{x}_* in Eq. (7). The result is a reformulation of the original equation that is now expressed in terms of entropies of predictive distributions, which are easier to approximate:

$$\alpha(\mathbf{x}) = \mathbb{H}[\mathbf{y}|\mathcal{D}, \mathbf{x}] - \mathbb{E}_{\mathbf{x}_*|\mathcal{D}}[\mathbb{H}[\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathbf{x}_*]]. \quad (8)$$

This is the same reformulation used by Predictive Entropy Search (PES) (Hernández-Lobato et al., 2014) for unconstrained Bayesian optimization, but extended to the case where \mathbf{y} is a vector rather than a scalar. Since we focus on constrained optimization problems, we call our method Predictive Entropy Search with Constraints (PESC). Eq. (8) is used by PESC to efficiently solve constrained Bayesian optimization problems with decoupled function evaluations. In the following section we describe how to obtain a computationally efficient approximation to Eq. (8). We also show that the resulting approximation is separable.

4.1 The PESC Acquisition Function

We assume that the functions f, c_1, \dots, c_K are independent samples from Gaussian process (GP) priors and that the noisy measurements \mathbf{y} returned by the black-boxes are obtained by adding Gaussian noise to the noise-free function evaluations at \mathbf{x} . Under this Bayesian model for the data, the first term in Eq. (8) can be computed exactly. In particular,

$$\mathbb{H}[\mathbf{y}|\mathcal{D}, \mathbf{x}] = \sum_{i=1}^{K+1} \frac{1}{2} \log \sigma_i^2(\mathbf{x}) + \frac{K+1}{2} \log(2\pi\epsilon), \quad (9)$$

where $\sigma_i^2(\mathbf{x})$ is the predictive variance for y_i at \mathbf{x} and y_i is the i -th entry in \mathbf{y} . To obtain this formula we have used the fact that f, c_1, \dots, c_K are generated independently, so that $\mathbb{H}[\mathbf{y}|\mathcal{D}, \mathbf{x}] = \sum_{i=1}^{K+1} \mathbb{H}[y_i|\mathcal{D}, \mathbf{x}]$, and that $p(y_i|\mathcal{D}, \mathbf{x})$ is Gaussian with variance parameter $\sigma_i^2(\mathbf{x})$ given by the GP predictive variance (Rasmussen and Williams, 2006):

$$\sigma_i^2(\mathbf{x}) = k_i(\mathbf{x}) - \mathbf{k}_i(\mathbf{x})^T \mathbf{K}_i^{-1} \mathbf{k}_i(\mathbf{x}) + \nu_i, \quad i = 1, \dots, K+1, \quad (10)$$

where ν_i is the variance of the additive Gaussian noise in the i -th black-box, with f being the first one and c_K the last one. The scalar $k_i(\mathbf{x})$ is the prior variance of the noise-free black-box evaluations at \mathbf{x} . The vector $\mathbf{k}_i(\mathbf{x})$ contains the prior covariances between the black-box values at \mathbf{x} and at those locations for which data from the black-box is available. Finally, \mathbf{K}_i is a matrix with the prior covariances for the noise-free black-box evaluations at those locations for which data is available.

The second term in Eq. (8), that is, $\mathbb{E}_{\mathbf{x}_*|\mathcal{D}}[\mathbb{H}[\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathbf{x}_*]]$, cannot be computed exactly and needs to be approximated. We do this operation as follows. ①: The expectation with respect to $p(\mathbf{x}_*|\mathcal{D})$ is approximated with an empirical average over M samples drawn from $p(\mathbf{x}_*|\mathcal{D})$. These samples are generated by following the approach proposed by Hernández-Lobato et al. (2014) for sampling \mathbf{x}_* in the unconstrained case. We draw approximate posterior samples of f, c_1, \dots, c_K , as described by Hernández-Lobato et al. (2014, Appendix A), and then solve Eq. (1) to obtain \mathbf{x}_* given the sampled functions. More details can be found in Appendix B.3 of this document. Note that this approach only applies for stationary kernels, but this class includes popular choices such as the squared exponential and Matérn kernels. ②: We assume that the components of \mathbf{y} are independent given \mathcal{D} , \mathbf{x} and \mathbf{x}_* , that is, we assume that the evaluations of f, c_1, \dots, c_K at \mathbf{x} are independent given \mathcal{D} and \mathbf{x}_* . This factorization assumption guarantees that the acquisition function used by PESC is additive across the different functions that are being evaluated. ③: Let \mathbf{x}_j^* be the j -th sample from $p(\mathbf{x}_*|\mathcal{D})$. We then find a Gaussian approximation to each $p(y_i|\mathcal{D}, \mathbf{x}, \mathbf{x}_j^*)$ using expectation propagation (EP) (Minka, 2001a). Let $\sigma_i^2(\mathbf{x}|\mathbf{x}_j^*)$ be the variance of the Gaussian

approximation to $p(y_i | \mathcal{D}, \mathbf{x}, \mathbf{x}_i^j)$ given by EP. Then, we obtain

$$\begin{aligned} \mathbb{E}_{\mathbf{x}_* | \mathcal{D}} [\mathbb{H}[y_i | \mathcal{D}, \mathbf{x}, \mathbf{x}_*]] &\stackrel{\textcircled{1}}{\approx} \frac{1}{M} \sum_{j=1}^M \mathbb{H}[y_i | \mathcal{D}, \mathbf{x}, \mathbf{x}_*^j] \stackrel{\textcircled{2}}{\approx} \frac{1}{M} \sum_{j=1}^M \left[\sum_{i=1}^{K+1} \mathbb{H}[y_i | \mathcal{D}, \mathbf{x}, \mathbf{x}_*^j] \right] \\ &\stackrel{\textcircled{3}}{\approx} \sum_{i=1}^{K+1} \left\{ \frac{1}{M} \sum_{j=1}^M \frac{1}{2} \log \sigma_i^2(\mathbf{x} | \mathbf{x}_*^j) \right\} + \frac{K+1}{2} \log(2\pi e), \end{aligned} \quad (11)$$

where each of the approximations has been numbered with the corresponding step from the description above. Note that in step $\textcircled{3}$ of Eq. (11) we have swapped the sums over i and j .

The acquisition function used by PESC is then given by the difference between Eq. (9) and the approximation shown in the last line of Eq. (11). In particular, we obtain

$$\alpha_{\text{PESC}}(\mathbf{x}) = \sum_{i=1}^{K+1} \tilde{\alpha}_i(\mathbf{x}), \quad (12)$$

where

$$\tilde{\alpha}_i(\mathbf{x}) = \frac{1}{M} \sum_{j=1}^M \underbrace{\frac{1}{2} \log \sigma_i^2(\mathbf{x}) - \frac{1}{2} \log \sigma_i^2(\mathbf{x} | \mathbf{x}_*^j)}_{\tilde{\alpha}_i(\mathbf{x} | \mathbf{x}_*^j)}, \quad i = 1, \dots, K+1. \quad (13)$$

Interestingly, the factorization assumption that we made in step $\textcircled{2}$ of Eq. (11) has produced an acquisition function in Eq. (12) that is the sum of $K+1$ function-specific acquisition functions, given by the $\tilde{\alpha}_i(\mathbf{x})$ in Eq. (13). Each $\tilde{\alpha}_i(\mathbf{x})$ measures how much information we gain on average by only evaluating the i -th black box, where the first black-box evaluates f and the last one evaluates c_{K+1} . Furthermore, $\tilde{\alpha}_i(\mathbf{x})$ is the empirical average of $\tilde{\alpha}_i(\mathbf{x} | \mathbf{x}_*^j)$ across M samples from $p(\mathbf{x}_* | \mathcal{D})$. Therefore, we can interpret each $\tilde{\alpha}_i(\mathbf{x} | \mathbf{x}_*^j)$ in Eq. (13) as a function-specific acquisition function conditioned on \mathbf{x}_* . Crucially, by using bits of information about the minimizer as a common unit of measurement, our acquisition function can make meaningful comparisons between the usefulness of evaluating the objective and constraints.

We now show how PESC can be used to obtain the task-specific acquisition functions required by the general algorithm from Section 3.3. Let us assume that we plan to evaluate only a subset of the functions f, c_1, \dots, c_K and let $t \subseteq \{1, \dots, K+1\}$ contain the indices of the functions to be evaluated, where the first function is f and the last one is c_K . We assume that the functions indexed by t are coupled and require joint evaluation. In this case t encodes a *task* according to the definition from Section 3.2. We can then approximate the expected gain of information that is obtained by evaluating this task at input \mathbf{x} . The process is similar to the one used above when all the black-boxes are evaluated at the same time. However, instead of working with the full vector \mathbf{y} , we now work with the components of \mathbf{y} indexed by t . One can then show that the expected information gain obtained after evaluating task t at input \mathbf{x} can be approximated as

$$\alpha_t(\mathbf{x}) = \sum_{i \in t} \tilde{\alpha}_i(\mathbf{x}), \quad (14)$$

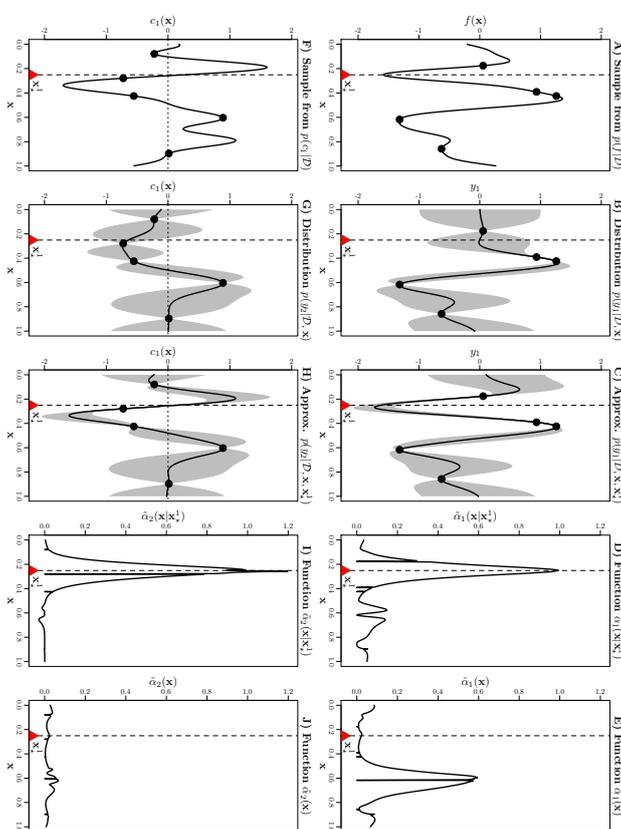


Figure 2: Illustration of the process followed to compute the function-specific acquisition functions given by Eq. (13). See the main text for details.

where the $\tilde{\alpha}_i$ are given by Eq. (13). PESC’s acquisition function is therefore separable since Eq. (14) can be used to obtain an acquisition function for each possible task. The process for constructing these task-specific acquisition functions is also efficient since it requires only to use the individual acquisition functions from Eq. (13) as building blocks. These two properties make PESC an effective solution for the practical implementation of the general algorithm from Section 3.3.

Fig. 2 illustrates with a toy example the process for computing the function-specific acquisition functions from Eq. (13). In this example there is only one constraint function. Therefore, the functions in the optimization problem are only f and c_1 . The search space \mathcal{X} is the unit interval $[0, 1]$ and we have collected four measurements for each function. The data for f are shown as black points in panels A, B and C. The data for c_1 are shown as black points in panels F, G and H. We assume that f and c_1 are independently sampled from a GP with zero mean function and squared exponential covariance function with unit amplitude and length-scale 0.07. The noise variance for the black-boxes that evaluate f and c_1 is zero. Let y_1 and y_2 be the black-box evaluations for f and c_1 at input \mathbf{x} . Under the assumed GP model we can analytically compute the predictive distributions for y_1 and

y_2 , that is, $p(y_1|\mathcal{D}, \mathbf{x})$ and $p(y_2|\mathcal{D}, \mathbf{x})$. Panels B and G show the means of these distributions with confidence bands equal to one standard deviation. The first step to compute the $\tilde{\alpha}_i(\mathbf{x})$ from Eq. (13) is to draw M samples from $p(\mathbf{x}_*|\mathcal{D})$. To generate each of these samples, we first approximately sample f and c_1 from their posterior distributions $p(f|\mathcal{D})$ and $p(c_1|\mathcal{D})$ using the method described by Hernández-Lobato et al. (2014, Appendix A). Panels A and F show one of the samples obtained for f and c_1 , respectively. We then solve the optimization problem given by Eq. (1) when f and c_1 are known and equal to the samples obtained. The solution to this problem is the input that minimizes f subject to c_1 being positive. This produces a sample \mathbf{x}_*^1 from $p(\mathbf{x}_*|\mathcal{D})$ which is shown as a discontinuous vertical line with a red triangle in all the panels. The next step is to find a Gaussian approximation to the predictive distributions when we condition to \mathbf{x}_*^1 , that is, $p(y_1|\mathcal{D}, \mathbf{x}, \mathbf{x}_*^1)$ and $p(y_2|\mathcal{D}, \mathbf{x}, \mathbf{x}_*^1)$. This step is performed using expectation propagation (EP) as described in Section 4.2 and Appendix A. Panels C and H show the approximations produced by EP for $p(y_1|\mathcal{D}, \mathbf{x}, \mathbf{x}_*^1)$ and $p(y_2|\mathcal{D}, \mathbf{x}, \mathbf{x}_*^1)$, respectively. Panel C shows that conditioning to \mathbf{x}_*^1 decreases the posterior mean of y_1 in the neighborhood of \mathbf{x}_*^1 . The reason for this is that \mathbf{x}_*^1 must be the global feasible solution and this means that $f(\mathbf{x}_*^1)$ must be lower than any other feasible point. Panel H shows that conditioning to \mathbf{x}_*^1 increases the posterior mean of y_2 in the neighborhood of \mathbf{x}_*^1 . The reason for this is that $c_1(\mathbf{x}_*^1)$ must be positive because \mathbf{x}_*^1 has to be feasible. In particular, by conditioning to \mathbf{x}_*^1 we are giving zero probability to all c_1 such that $c_1(\mathbf{x}_*^1) < 0$. Let $\sigma_1^2(\mathbf{x}|\mathbf{x}_*^1)$ and $\sigma_2^2(\mathbf{x}|\mathbf{x}_*^1)$ be the variances of the Gaussian approximations to $p(y_1|\mathcal{D}, \mathbf{x}, \mathbf{x}_*^1)$ and $p(y_2|\mathcal{D}, \mathbf{x}, \mathbf{x}_*^1)$ and let $\sigma_1^2(\mathbf{x})$ and $\sigma_2^2(\mathbf{x})$ be the variances of $p(y_1|\mathcal{D}, \mathbf{x})$ and $p(y_2|\mathcal{D}, \mathbf{x})$. We use these quantities to obtain $\hat{\alpha}_1(\mathbf{x}|\mathbf{x}_*^1)$ and $\hat{\alpha}_2(\mathbf{x}|\mathbf{x}_*^1)$ according to Eq. (13). These two functions are shown in panels D and I. The whole process is repeated $M = 50$ times and the resulting $\hat{\alpha}_1(\mathbf{x}|\mathbf{x}_*^j)$ and $\hat{\alpha}_2(\mathbf{x}|\mathbf{x}_*^j)$, $j = 1, \dots, M$, are averaged according to Eq. (13) to obtain the function-specific acquisition functions $\hat{\alpha}_1(\mathbf{x})$ and $\hat{\alpha}_2(\mathbf{x})$, whose plots are shown in panels E and J. These plots indicate that evaluating the objective f is in this case more informative than evaluating the constraint c_1 . But this is certainly not always the case, as will be demonstrated in the experiments later on.

4.2 How to Compute the Gaussian Approximation to $p(y_i|\mathcal{D}, \mathbf{x}, \mathbf{x}_*^j)$

We briefly describe the process followed to find a Gaussian approximation to $p(y_i|\mathcal{D}, \mathbf{x}, \mathbf{x}_*^j)$ using expectation propagation (EP) (Minka, 2001b). Recall that the variance of this approximation, that is, $\sigma_i^2(\mathbf{x}|\mathbf{x}_*^j)$, is used to compute $\hat{\alpha}_i(\mathbf{x}|\mathbf{x}_*^j)$ in Eq. (13). Here we only provide a sketch of the process; full details can be found in Appendix A.

We start by assuming that the search space has finite size, that is, $\mathcal{X} = \{\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_{|\mathcal{X}|}\}$. In this case the functions f , c_1, \dots, c_K are encoded as finite dimensional vectors denoted by \mathbf{f} , $\mathbf{c}_1, \dots, \mathbf{c}_K$. The i -th entries in these vectors are the result of evaluating f , c_1, \dots, c_K at the i -th element of \mathcal{X} , that is, $f(\tilde{\mathbf{x}}_i)$, $c_1(\tilde{\mathbf{x}}_i), \dots, c_K(\tilde{\mathbf{x}}_i)$. Let us assume that \mathbf{x}_*^j and \mathbf{x} are in \mathcal{X} . Then $p(\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathbf{x}_*^j)$ can be defined by the following rejection sampling process. First, we sample \mathbf{f} , $\mathbf{c}_1, \dots, \mathbf{c}_K$ from their posterior distribution given the assumed GP models. We then solve the optimization problem given by Eq. (1). For this, we find the entry of \mathbf{f} with lowest value subject to the corresponding entries of $\mathbf{c}_1, \dots, \mathbf{c}_K$ being positive. Let $i \in \{1, \dots, |\mathcal{X}|\}$ be the index of the selected entry. Then, if $\mathbf{x}_*^j \neq \tilde{\mathbf{x}}_i$, we reject the sampled \mathbf{f} , $\mathbf{c}_1, \dots, \mathbf{c}_K$ and start again. Otherwise, we take the entries of \mathbf{f} , $\mathbf{c}_1, \dots, \mathbf{c}_K$

indexed by \mathbf{x} , that is, $f(\mathbf{x})$, $c_1(\mathbf{x}), \dots, c_K(\mathbf{x})$ and then obtain \mathbf{y} by adding to each of these values a Gaussian random variable with zero mean and variance ν_1, \dots, ν_{K+1} , respectively. The probability distribution implied by this rejection sampling process can be obtained by first multiplying the posterior for \mathbf{f} , $\mathbf{c}_1, \dots, \mathbf{c}_K$ with indicator functions that take value zero when \mathbf{f} , $\mathbf{c}_1, \dots, \mathbf{c}_K$ should be rejected and one otherwise. We can then multiply the resulting quantity by the likelihood for \mathbf{y} given \mathbf{f} , $\mathbf{c}_1, \dots, \mathbf{c}_K$. The desired distribution is finally obtained by marginalizing out \mathbf{f} , $\mathbf{c}_1, \dots, \mathbf{c}_K$.

We introduce several indicator functions to implement the approach described above. The first one $\Gamma(\mathbf{x})$ takes value one when \mathbf{x} is a feasible solution and value zero otherwise, that is,

$$\Gamma(\mathbf{x}) = \prod_{k=1}^K \Theta[c_k(\mathbf{x})], \quad (15)$$

where $\Theta[\cdot]$ is the Heaviside step function which is equal to one if its input is non-negative and zero otherwise. The second indicator function $\Psi(\mathbf{x})$ takes value zero if \mathbf{x} is a better solution than \mathbf{x}_*^j according to the sampled functions. Otherwise $\Psi(\mathbf{x})$ takes value one. In particular,

$$\Psi(\mathbf{x}) = \Gamma(\mathbf{x})[\Theta[f(\mathbf{x}) - f(\mathbf{x}_*^j)] + (1 - \Gamma(\mathbf{x}))]. \quad (16)$$

When \mathbf{x} is infeasible, this expression takes value one. In this case, \mathbf{x} is not a better solution than \mathbf{x}_*^j (because \mathbf{x} is infeasible) and we do not have to reject. When \mathbf{x} is feasible, the factor $\Theta[f(\mathbf{x}) - f(\mathbf{x}_*^j)]$ in Eq. (16) is zero when \mathbf{x} takes lower objective value than \mathbf{x}_*^j . This will allow us to reject \mathbf{f} , $\mathbf{c}_1, \dots, \mathbf{c}_K$ when \mathbf{x} is a better solution than \mathbf{x}_*^j . Using Eq. (15) and Eq. (16), we can then write $p(\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathbf{x}_*^j)$ as

$$p(\mathbf{y}|\mathcal{D}, \mathbf{x}, \mathbf{x}_*^j) \propto \underbrace{p(\mathbf{y}|\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K, \mathbf{x}) p(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K|\mathcal{D}) \Gamma(\mathbf{x}_*^j)}_{f(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K|\mathbf{x}_*^j)} \left\{ \prod_{\mathbf{x}' \in \mathcal{X}} \Psi(\mathbf{x}') \right\} df d\mathbf{c}_1 \dots d\mathbf{c}_K, \quad (17)$$

where $p(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K|\mathcal{D})$ is the GP posterior distribution for the noise-free evaluations of f , c_1, \dots, c_K at \mathcal{X} and $p(\mathbf{y}|\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K, \mathbf{x})$ is the likelihood function, that is, the distribution of the noisy evaluations produced by the black-boxes with input \mathbf{x} given the true function values:

$$p(\mathbf{y}|\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K, \mathbf{x}) = \mathcal{N}(y_1|f(\mathbf{x}), \nu_1) \mathcal{N}(y_2|c_1(\mathbf{x}), \nu_2) \dots \mathcal{N}(y_{K+1}|c_K(\mathbf{x}), \nu_{K+1}). \quad (18)$$

The product of the indicator functions Γ and Ψ in Eq. (17) takes value zero whenever \mathbf{x}_*^j is not the best feasible solution according to $\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K$. The indicator Γ in Eq. (17) guarantees that \mathbf{x}_*^j is a feasible location. The product of all the Ψ in Eq. (17) guarantees that no other point in \mathcal{X} is better than \mathbf{x}_*^j . Therefore, the product of Γ and the Ψ in Eq. (17) rejects any value of $\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K$ for which \mathbf{x}_*^j is not the optimal solution to the constrained optimization problem.

The factors $p(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K|\mathcal{D})$ and $p(\mathbf{y}|\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K, \mathbf{x})$ in Eq. (17) are Gaussian. Thus, their product is also Gaussian and tractable. However, the integral in Eq. (17) does not

have a closed form solution because of the complexity introduced by the the product of indicator functions Γ and Ψ . This means that Eq. (17) cannot be exactly computed and has to be approximated. For this, we use EP to fit a Gaussian approximation to the product of $p(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K | \mathcal{D})$ and the indicator functions Γ and Ψ in Eq. (17), which we have denoted by $f(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K | \mathbf{x}_i^j)$, with a tractable Gaussian distribution given by

$$q(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K | \mathbf{x}_i^j) = \mathcal{N}(\mathbf{f} | \mathbf{m}_1, \mathbf{V}_1) \mathcal{N}(\mathbf{c}_1 | \mathbf{m}_2, \mathbf{V}_2) \dots \mathcal{N}(\mathbf{c}_K | \mathbf{m}_{K+1}, \mathbf{V}_{K+1}), \quad (19)$$

where $\mathbf{m}_1, \dots, \mathbf{m}_{K+1}$ and $\mathbf{V}_1, \dots, \mathbf{V}_{K+1}$ are mean vectors and covariance matrices to be determined by the execution of EP. Let $v_i(\mathbf{x})$ be the diagonal entry of \mathbf{V}_i corresponding to the evaluation location given by \mathbf{x} , where $i = 1, \dots, K+1$. Similarly, let $m_i(\mathbf{x})$ be the entry of \mathbf{m}_i corresponding to the evaluation location \mathbf{x} for $i = 1, \dots, K+1$. Then, by replacing $f(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K | \mathbf{x}_i^j)$ in Eq. (17) with $q(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K | \mathbf{x}_i^j)$, we obtain

$$p(\mathcal{Y} | \mathcal{D}, \mathbf{x}, \mathbf{x}_i^j) \approx \prod_{k=1}^{K+1} \mathcal{N}(y_k | m_k(\mathbf{x}), v_k(\mathbf{x}) + v_k). \quad (20)$$

Consequently, $\sigma_k^2(\mathbf{x} | \mathbf{x}_i^j) = v_k(\mathbf{x}) + v_k$ can be used to compute $\tilde{\alpha}_k(\mathbf{x} | \mathbf{x}_i^j)$ in Eq. (13).

The previous approach does not work when the search space \mathcal{X} has infinite size, for example when $\mathcal{X} = [0, 1]^d$ with d being the dimension of the inputs to f, c_1, \dots, c_K . In this case the product of indicators in Eq. (17) includes an infinite number of factors $\Psi(\mathbf{x}^i)$, one for each possible $\mathbf{x}^i \in \mathcal{X}$. To solve this problem we perform an additional approximation. For the computation of Eq. (17), we consider that \mathcal{X} is well approximated by the finite set \mathcal{Z} , which contains only the locations at which the objective f has been evaluated so far, the value of \mathbf{x}_i^j and \mathbf{x} . Therefore, we approximate the factor $\prod_{\mathbf{x}^i \in \mathcal{X}} \Psi(\mathbf{x}^i)$ in Eq. (17) with the factor $\prod_{\mathbf{x}^i \in \mathcal{Z}} \Psi(\mathbf{x}^i)$, which has now finite size. We expect this approximation to become more and more accurate as we increase the amount of data collected for f . Note that our approximation to \mathcal{X} is finite, but it is also different for each location \mathbf{x} at which we want to evaluate Eq. (17) since \mathcal{Z} is defined to contain \mathbf{x} . A detailed description of the resulting EP algorithm, indicating how to compute the variance functions $v_k(\mathbf{x})$ shown in Eq. (20), is given in Appendix A.

The EP approximation to Eq. (20), performed after replacing \mathcal{X} with \mathcal{Z} , depends on the values of \mathcal{D} , \mathbf{x}_i^j and \mathbf{x} . Having to re-run EP for each value of \mathbf{x} at which we may want to evaluate the acquisition function given by Eq. (12) is a very expensive operation. To avoid this, we split the EP computations between those that depend only on \mathcal{D} and \mathbf{x}_i^j , which are the most expensive ones, and those that depend only on the value of \mathbf{x} . We perform the former computations only once and then reuse them for each different value of \mathbf{x} . This allows us to evaluate the EP approximation to Eq. (17) at different values of \mathbf{x} in a computationally efficient way. See Appendix A for further details.

4.3 Efficient Marginalization of the Model Hyper-parameters

So far we have assumed to know the optimal hyper-parameter values; that is, the amplitude and the length-scales for the GPs and the noise variances for the black-boxes. However, in practice, the hyper-parameter values are unknown and have to be estimated from data. This can be done for example by drawing samples from the posterior distribution of the

hyper-parameters under some non-informative prior. Ideally, we should then average the GP predictive distributions with respect to the generated samples before approximating the information gain. However, this approach is too computationally expensive in practice. Instead, we follow Snoek et al. (2012) and average the PESC acquisition function with respect to the generated hyper-parameter samples. In our case, this involves marginalizing each of the function-specific acquisition functions from Eq. (13). For this, we follow the method proposed by Hernández-Lobato et al. (2014) to average the acquisition function of Predictive Entropy Search in the unconstrained case. Let Θ denote the model hyper-parameters. First, we draw M samples $\Theta^1, \dots, \Theta^M$ from the posterior distribution of Θ given the data \mathcal{D} . Typically, for each of the posterior samples Θ^j of Θ we draw a single corresponding sample \mathbf{x}_i^j from the posterior distribution of \mathbf{x}_* given Θ^j , that is, $p(\mathbf{x}_* | \mathcal{D}, \Theta^j)$. Let $\sigma_k^2(\mathbf{x} | \Theta^j)$ be the variance of the GP predictive distribution for y_i when the hyper-parameter values are fixed to Θ^j , that is, $p(y_i | \mathcal{D}, \mathbf{x}, \Theta^j)$, and let $\sigma_k^2(\mathbf{x} | \mathbf{x}_i^j, \Theta^j)$ be the variance of the Gaussian approximation to the predictive distribution for y_i when we condition to the solution of the optimization problem being \mathbf{x}_i^j and the hyper-parameter values being Θ^j . Then, the version of Eq. (13) that marginalizes out the model hyper-parameters is given by

$$\tilde{\alpha}_k(\mathbf{x}) = \frac{1}{M} \sum_{j=1}^M \left\{ \frac{1}{2} \log \sigma_k^2(\mathbf{x} | \Theta^j) - \frac{1}{2} \log \sigma_k^2(\mathbf{x} | \mathbf{x}_i^j, \Theta^j) \right\}, \quad i = 1, \dots, K+1. \quad (21)$$

Note that j is now an index over joint posterior samples of the model hyper-parameters Θ and the constrained minimizer \mathbf{x}_* . Therefore, we can marginalize out the hyper-parameter values without adding any additional computational complexity to our method because a loop over M samples of \mathbf{x}_* is just replaced with a loop over M joint samples of (Θ, \mathbf{x}_*) . This is a consequence of our reformulation of Eq. (7) into Eq. (8). By contrast, other techniques that work by approximating the original form of the acquisition function used in Eq. (7) do not have this property. An example in the unconstrained setting is Entropy Search (Hennig and Schuler, 2012), which requires re-computing an approximation to the acquisition function for each hyper-parameter sample Θ^j .

4.4 Computational Complexity

In the coupled setting, the complexity of PESC is $\mathcal{O}(MK^2N^3)$, where M is the number of posterior samples of the global constrained minimizer \mathbf{x}_* , K is the number of constraints, and N is the number of collected data points. This cost is determined by the cost of each EP iteration, which requires computing the inverse of the covariance matrices $\mathbf{V}_1, \dots, \mathbf{V}_{K+1}$ in Eq. (20). The dimensionality of each of these matrices grows with the size of \mathcal{Z} , which is determined by the number N of objective evaluations (see the last paragraph of Section 4.2). Therefore each EP iteration has cost $\mathcal{O}(KN^3)$ and we have to run an instance of EP for each of the M samples of \mathbf{x}_* . If M is also the number of posterior samples for the GP hyperparameters, as explained in Section 4.3, this is the same computational complexity as in EIC. However, in practice PESC is slower than EIC because of the cost of running multiple iterations of the EP algorithm.

In the decoupled setting the cost of PESC is $\mathcal{O}(M \sum_{k=2}^{K+1} (N_1 + N_k)^3)$ where N_1 is the number of evaluations of the objective and N_k is the number of evaluations for constraint

$k-1$. The origin of this cost is again the size of the matrices $\mathbf{V}_1, \dots, \mathbf{V}_{k+1}$ in Eq. (20). While \mathbf{V}_1 still scales as a function of $|\mathcal{Z}|$, we have that $\mathbf{V}_2, \dots, \mathbf{V}_{k+1}$ scale now as a function of $|\mathcal{Z}|$ plus the number of observations for the corresponding constraint function. The reason for this is that $\prod_{\mathbf{x}' \in \mathcal{Z}} \Psi(\mathbf{x}')$ is used to approximate $\prod_{\mathbf{x}' \in \mathcal{X}} \Psi(\mathbf{x}')$ in Eq. (17) and each factor in $\prod_{\mathbf{x}' \in \mathcal{Z}} \Psi(\mathbf{x}')$ represents then a virtual data point for each GP. See Appendix A for details.

The cost of sampling the GP hyper-parameters is $\mathcal{O}(MKN^3)$ and therefore, it does not affect the overall computational complexity of PESC.

4.5 Relationship between PESC and PES

PESC can be applied to unconstrained optimization problems. For this we only have to set $K=0$ and ignore the constraints. The resulting technique is very similar to the method PES proposed by Hernández-Lobato et al. (2014) as an information-based approach for unconstrained Bayesian optimization. However, PESC without constraints and PES are not identical. PES approximates $p(y|\mathcal{D}, \mathbf{x}, \mathbf{x}_k^*)$ by multiplying the GP predictive distribution by additional factors that enforce \mathbf{x}_k^* to be the location with lowest objective value. These factors guarantee that 1) the value of the objective at \mathbf{x}_k^* is lower than the minimum of the values for the objective collected so far, 2) the gradient of the objective is zero at \mathbf{x}_k^* and 3) the Hessian of the objective is positive definite at \mathbf{x}_k^* . We do not enforce the last two conditions since the global optimum may be on the boundary of a feasible region and thus conditions 2) and 3) do not necessarily hold (this issue also arises in PES because the optimum may be on the boundary of the search space \mathcal{X}). Condition 1) is implemented in PES by taking the minimum observed value for the objective, denoted by η , and then imposing the soft condition $f(\mathbf{x}_k^*) < \eta + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \nu)$ accounts for the additive Gaussian noise with variance ν in the black-box that evaluates the objective. In PESC this is achieved in a more principled way by using the indicator functions given by Eq. (16).

4.6 Summary of the Approximations Made in PESC

We describe here all the approximations performed in the practical implementation of PESC. PESC approximates the expected reduction in the posterior entropy of \mathbf{x}_* (see Eq. 7) with the acquisition function given by Eq. (12). This involves the following approximations:

1. The expectation over \mathbf{x}_* in Eq. (8) is approximated with Monte Carlo sampling.
2. The Monte Carlo samples of \mathbf{x}_* come from samples of f, c_1, \dots, c_K drawn approximately using a finite basis function approximation to the GP covariance function, as described by Hernández-Lobato et al. (2014, Appendix A).
3. We approximate the factor $\prod_{\mathbf{x}' \in \mathcal{X}} \Psi(\mathbf{x}')$ in Eq. (17) with the factor $\prod_{\mathbf{x}' \in \mathcal{Z}} \Psi(\mathbf{x}')$. Unlike the original search space \mathcal{X} , \mathcal{Z} has now finite size and the corresponding product of Ψ indicators is easier to approximate. The set \mathcal{Z} is formed by the locations of the current observations for the objective f and the current evaluation location \mathbf{x} of the acquisition function.
4. After replacing $\prod_{\mathbf{x}' \in \mathcal{X}} \Psi(\mathbf{x}')$ with $\prod_{\mathbf{x}' \in \mathcal{Z}} \Psi(\mathbf{x}')$ in Eq. (17), we further approximate the factor $f(\mathbf{f}, c_1, \dots, c_K; \mathbf{x}_k^*)$ in this equation with the Gaussian approximation given

by the right-hand-side of Eq. (20). We use the method expectation propagation (EP) for this task, as described in Appendix A. Because the EP approximation in Eq. (19) factorizes across $\mathbf{f}, c_1, \dots, c_K$, the execution of EP implicitly includes the factorization assumption performed in step ② of Eq. (11).

5. As described in the last paragraph of Section 4.2, in the execution of EP we separate the computations that depend on \mathcal{D} and \mathbf{x}_k^* , which are very expensive, from those that depend on the location \mathbf{x} at which the PESC acquisition function will be evaluated. This allows us to evaluate the approximation to Eq. (17) at different values of \mathbf{x} in a computationally efficient way.
6. To deal with unknown hyper-parameter values, we marginalize the acquisition function over posterior samples of the hyper-parameters. Ideally, we should instead marginalize the predictive distributions with respect to the hyper-parameters before computing the entropy, but this is too computationally expensive in practice.

In Section 6.1, we assess the accuracy of these approximations (except the last one) and show that PESC performs on par with a ground-truth method based on rejection sampling.

Note that in addition to the mathematical approximations described above, additional sources of error are introduced by the numerical computations involved. In addition to the usual roundoff error, etc., we draw the reader's attention to the fact that the \mathbf{x}_* samples are the result of numerical global optimization of the approximately drawn samples of f, c_1, \dots, c_K , and then the suggestion is chosen by another numerical global optimization of the acquisition function. At present, we do not have guarantees that the true global optimum is found by our numerical methods in each case.

5. PESC-F: Speeding Up the BO Computations

One disadvantage of PESC is that sampling \mathbf{x}_* and then computing the corresponding EP approximation can be slow. If PESC is slow with respect to the evaluation of the black-box functions f, c_1, \dots, c_K , the entire Bayesian optimization (BO) procedure may be inefficient. For the BO approach to be useful, the time spent doing meta-computations has to be significantly shorter than the time spent actually evaluating the objective and constraints. This issue can be avoided in the coupled case by, for example, switching to a faster acquisition function like EIC or abandoning BO entirely for methods such as the popular CMA-ES (Hansen and Ostermeier, 1996) evolutionary strategy. However, in the decoupling setting, one can encounter problems in which some tasks are fast and others are slow. In this case, a cumbersome BO method might be undesirable because it would be unreasonable to spend minutes making a decision about a task that only takes seconds to complete; and, yet, a method that is fast but inefficient in terms of function evaluations would be ill-suited to making decisions about a task that takes hours complete. This situation calls for an optimization algorithm that can adaptively adjust its own decision-making time. For this reason, we introduce additional approximations in the computations made by PESC to reduce their cost when necessary. The new method that adaptively switches between fast and slow decision-making computations is called PESC-F. The two main challenges are how to speed up the original computations made by PESC and how to

decide when to switch between the slow and the fast versions of those computations. In the following paragraphs we address these issues.

We propose ways to reduce the cost of the computations performed by PESC after collecting each new data point. These computations include

1. Drawing posterior samples of the GP hyper-parameters and then for each sample computing the Cholesky decomposition of the kernel matrix.
2. Drawing approximate posterior samples of \mathbf{x}_* and then running an EP algorithm for each of these samples.
3. Globally maximizing the resulting acquisition functions.

We shorten each of these steps. First, we reduce the cost of step 1 by skipping the sampling of the GP hyper-parameters and instead considering the hyper-parameter samples already used at an earlier iteration. This also allows for additional speedups by using fast ($\mathcal{O}(N^2)$) updates of the Cholesky decomposition of the kernel matrix instead of recomputing it from scratch. Second, we shorten step 2 by skipping the sampling of \mathbf{x}_* and instead considering the samples used at the previous iteration. We also reuse the EP solutions computed at the previous iteration (see Appendix A for further details on how to reuse the EP solutions). Finally, we shorten step 3 by using a coarser termination condition tolerance when maximizing the acquisition function. This allows the optimization process to converge faster but with reduced precision. Furthermore, if the acquisition function is maximized using a local optimizer with random restarts and/or a grid initialization, we can shorten the computation further by reducing the number of restarts and/or grid size.

5.1 Choosing When to Run the Fast or the Slow Version

The motivation for PESC-F is that the time spent in the BO computations should be small compared to the time spent evaluating the black-box functions. Therefore, our approach is to switch between two distinct types of BO computations: the full (slow) and the partial (fast) PESC computations. Our goal is to approximately keep constant the fraction of total wall-clock time consumed by such computations. To achieve this, at each iteration of the BO process, we use the slow version of the computations if and only if

$$\frac{\tau_{\text{now}} - \tau_{\text{last}}}{\tau_{\text{slow}}} > \gamma, \quad (22)$$

where τ_{now} is the current time, τ_{last} is the time at which the last slow BO computations were complete, τ_{slow} is the duration of the last execution of the slow BO computations (this includes the time passed since the actual collection of the data until the maximization of the acquisition function) and $\gamma > 0$ is a constant called the rationality level. The larger the value of γ , the larger the amount of time spent in rational decision making, that is, in performing BO computations. Algorithm 2 shows the steps taken by PESC-F for the decoupled competitive case. In this case each function f, c_1, \dots, c_K represents a different task, that is, the different functions can be evaluated in a decoupled manner and in addition to this, all of them compete for using a single computational resource.

One could replace τ_{slow} with an average over the durations of past slow computations. While this approach is less noisy, we opt for using only the duration of the most recent

Algorithm 2 PESC-F for competitive decoupled functions.

- 1: **Inputs:** $\mathcal{T} = \{f, \{c_1, \dots, c_K\}, \mathcal{D}, \gamma, \delta$.
 - 2: $\tau_{\text{last}} \leftarrow 0$
 - 3: $\tau_{\text{slow}} \leftarrow 0$
 - 4: **repeat**
 - 5: $\tau_{\text{now}} \leftarrow$ current time
 - 6: **if** $(\tau_{\text{now}} - \tau_{\text{last}})/\tau_{\text{slow}} > \gamma$ **then**
 - 7: Sample GP hyper-parameters
 - 8: Fit GP to \mathcal{D}
 - 9: Generate new samples of \mathbf{x}_*
 - 10: Compute the EP solutions from scratch
 - 11: $\tau_{\text{slow}} \leftarrow$ current time $- \tau_{\text{now}}$
 - 12: $\tau_{\text{last}} \leftarrow$ current time
 - 13: $\{\mathbf{x}^*, t^*\} \leftarrow \arg \max_{\mathbf{x} \in \mathcal{X}, t \in \mathcal{T}} \alpha_t(\mathbf{x})$ (expensive optimization)
 - 14: **else**
 - 15: Update fit of GP to \mathcal{D}
 - 16: Reuse previous EP solutions
 - 17: $\{\mathbf{x}^*, t^*\} \leftarrow \arg \max_{\mathbf{x} \in \mathcal{X}, t \in \mathcal{T}} \alpha_t(\mathbf{x})$ (cheap optimization)
 - 18: **end if**
 - 19: Add to \mathcal{D} the evaluation of the function in task t^* at input \mathbf{x}^*
 - 20: **until** termination condition is met
 - 21: **Output:** $\arg \min_{\mathbf{x} \in \mathcal{X}} \text{EGr}[f(\mathbf{x})]$ s.t. $p(c_1(\mathbf{x})) \geq 0, \dots, c_K(\mathbf{x}) \geq 0, \text{GP} \geq 1 - \delta$
-

slow update since these durations may exhibit deterministic trends. For example, the cost of computations tends to increase at each iteration due to the increase in data set size. If indeed the update duration increases monotonically, then the duration of the most recent update would be a more accurate estimate of the duration of the next slow update than the average duration of all past updates.

PESC-F can be used as a generalization of PESC, since it reduces to PESC in the case of sufficiently slow function evaluations. To see this, note that the time spent in a function evaluation will be upper bounded by $\tau_{\text{now}} - \tau_{\text{last}}$ and according to Eq. (22), the slow computations are performed when $\tau_{\text{now}} - \tau_{\text{last}} > \gamma \tau_{\text{slow}}$. When the function evaluation takes a very large amount of time, we have that τ_{slow} will always be smaller than that amount of time and the condition $\tau_{\text{now}} - \tau_{\text{last}} > \gamma \tau_{\text{slow}}$ will always be satisfied for reasonable choices of γ . Thus, PESC-F will always perform slow computations as we would expect. On the other hand, if the evaluation of the black-box function is very fast, PESC-F will mainly perform fast computations but will still occasionally perform slow ones, with a frequency roughly proportional to the function evaluation duration.

5.2 Setting the Rationality Level in PESC-F

PESC-F is designed so that the ratio of time spent in BO computations to time spent in function evaluations is at most γ . This notion is approximate because the time spent in function evaluations includes the time spent doing fast computations. The optimal value of γ may be problem-dependent, but we propose values of γ on the order of 0.1 to 1, which

correspond to spending roughly 50–90% of the total time performing function evaluations. The optimal γ may also change at different stages of the BO process. Selecting the optimal value of γ is a subject for future research. Note that in PESC-F we are making sub-optimal decisions because of time constraints. Therefore, PESC-F is a simple example of *bounded rationality*, which has its roots in the traditional AI literature. For example, Russell (1991) proposes to treat computation as a possible action that consumes time but increases the expected utility of future actions.

5.3 Bridging the Gap Between Fast and Slow Computations

As discussed above, PESC-F can be applied even when function evaluations are very slow, as it automatically reverts to standard PESC when $\tau_{\text{eval}} > \tau_{\text{slow}}$. However, if the function evaluations are extremely fast, that is, faster even than the fast PESC updates, then even PESC-F violates the condition that the decision-making should take less time than the function evaluations. We have already defined τ_{slow} as the duration of the slow BO computations. Let us also define τ_{fast} as the duration of the fast BO computations and τ_{eval} as the duration of the evaluation of the functions. Then, the intuition described above can be put into symbols by saying that PESC-F is most useful when $\tau_{\text{fast}} < \tau_{\text{eval}} < \tau_{\text{slow}}$.

Many aspects of PESC-F are not specific to PESC and could easily be adapted to other acquisition functions like EIC or even unconstrained acquisition functions like PES and EI. In particular, lines 9, 10 and 16 of Algorithm 2 are specific to PESC, whereas others are common to other techniques. For example, when using vanilla unconstrained EI, the computational bottleneck is likely to be the sampling of the GP hyper-parameters (Algorithm 2, line 7) and maximizing the acquisition function (Algorithm 2, line 13). The ideas presented above, namely to skip the hyper-parameter sampling and to optimize the acquisition function with a smaller grid and/or coarser tolerances, are applicable in this situation and might be useful in the case of a fairly fast objective function. However, as mentioned above, in the single-task case one retains the option to abandon BO entirely for a faster method, whereas in the multi-task case considered here, neither a purely slow nor a purely fast method suits the nature of the optimization problem. An interesting direction for future research is to further pursue this notion of optimization algorithms that bridge the gap between those designed for optimizing cheap (fast) functions and those designed for optimizing expensive (slow) functions.

6. Empirical Analyses in the Coupled Case

We first evaluate the performance of PESC in experiments with different types of coupled optimization problems. First, we consider synthetic problems of functions sampled from the GP prior distribution. Second, we consider analytic benchmark problems that were previously used in the literature on Bayesian optimization with unknown constraints. Finally, we address the meta-optimization of machine learning algorithms with unknown constraints.

For the first synthetic case, we follow the experimental setup used by Hennig and Schuler (2012) and Hernández-Lobato et al. (2014). The search space is the unit hypercube of dimension D , and the ground truth objective f is a sample from a zero-mean GP with a squared exponential covariance function of unit amplitude and length scale $\ell = 0.1$ in each dimension. We represent the function f by first sampling from the GP prior on a

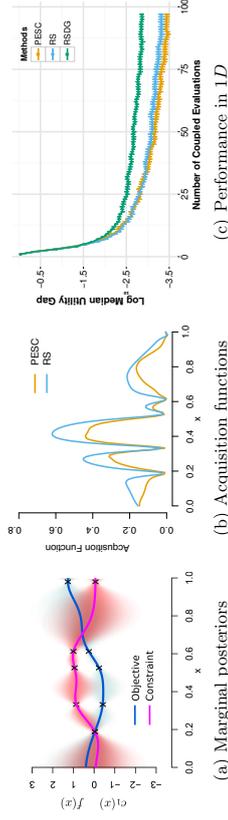


Figure 3: Accuracy of the PESC approximation. (a) Marginal posterior distributions for the objective and constraint given some collected data denoted by \times 's. (b) PESC and RS acquisition functions given the data in (a). (c) Median utility gap for PESC, RS and RSDG in the experiments with synthetic functions sampled from the GP prior with $D = 1$.

grid of 1000 points generated using a Halton sequence (see Leobacher and Pillichshammer, 2014) and then defining f as the resulting GP posterior mean. We use a single constraint function c_1 whose ground truth is sampled in the same way as f . The evaluations for f and c_1 are contaminated with i.i.d. Gaussian noise with variance $\nu_1 = \nu_2 = 0.01$.

6.1 Assessing the Accuracy of the PESC Approximation

We first analyze the accuracy of the PESC approximation to the acquisition function shown in Eq. (8). We compare the PESC approximation with a ground truth for the acquisition function obtained by rejection sampling (RS). The RS method works by discretizing the search space using a fine uniform grid. The expectation with respect to $p(\mathbf{x}_* | \mathcal{D})$ in Eq. (8) is then approximated by Monte Carlo. To achieve this, f, c_1, \dots, c_K are sampled on the grid and the grid cell with non-negative c_1, \dots, c_K (feasibility) and the lowest value of f (optimality) is selected. For each sample of \mathbf{x}_* , $\mathbb{H} [y^f, y^1, \dots, y^K | \mathcal{D}, \mathbf{x}_*, \mathbf{x}_*]$ is approximated by rejection sampling: we sample f, c_1, \dots, c_K on the grid and select those samples whose corresponding feasible optimal solution is the sampled \mathbf{x}_* and reject the other samples. We assume that the selected samples for f, c_1, \dots, c_K have a multivariate Gaussian distribution. Under this assumption, $\mathbb{H} [y^f, y^1, \dots, y^K | \mathcal{D}, \mathbf{x}_*, \mathbf{x}_*]$ can be approximated using the formula for the entropy of a multivariate Gaussian distribution, with the covariance parameter in the formula being equal to the empirical covariance of the selected samples for f and c_1, \dots, c_K at \mathbf{x} plus the corresponding noise variances ν_1 and ν_2, \dots, ν_{K+1} in its diagonal. In our experiments, this approach produces entropy estimates that are very similar, faster to obtain and less noisy than the ones obtained with non-parametric entropy estimators. We compared this implementation of RS with another version that ignores correlations in the samples of f and c_1, \dots, c_K . In practice, both methods produced equivalent results. Therefore, to speed up the method, we ignore correlations in our implementation of RS.

Figure 3(a) shows the posterior distribution for f and c_1 given 5 observations sampled from the GP prior with $D = 1$. The posterior is computed using the optimal GP hyper-

parameters. The corresponding approximations to the acquisition function generated by PESG and RS are shown in Fig. 3(b). In the figure, both PESG and RS use a total of $M = 50$ samples from $p(\mathbf{x}_* | \mathcal{D})$ when approximating the expectation in Eq. (8). The PESG approximation is quite accurate, and importantly its maximum value is very close to the maximum value of the RS approximation. The approximation produced by the version of RS that does not ignore correlations in the samples of f, c_1, \dots, c_K (not shown) cannot be visually distinguished from the one shown in Fig. 3(b).

One disadvantage of the RS method is its high cost, which scales with the size of the grid used. This grid has to be large to guarantee good performance, especially when D is large. An alternative is to use a small dynamic grid that changes as data is collected. Such a grid can be obtained by sampling from $p(\mathbf{x}_* | \mathcal{D})$ using the same approach as in PESG to generate these samples (a similar approach is taken by Hennig and Schuler (2012), in which the dynamic grid is sampled from the EI acquisition function). The samples obtained then form the dynamic grid, with the idea that grid points are more concentrated in areas that we expect $p(\mathbf{x}_* | \mathcal{D})$ to be high. The resulting method is called Rejection Sampling with a Dynamic Grid (RSDG).

We compare the performance of PESG, RS and RSDG in experiments with synthetic data corresponding to 500 pairs of f and c_1 sampled from the GP prior with $D = 1$. RS and RSDG draw the same number of samples of \mathbf{x}_* as PESG. We assume that the GP hyper-parameters are known to each method and fix $\delta = 0.05$, that is, recommendations are made by finding the location with highest posterior mean for f such that c_1 is non-negative with probability at least $1 - \delta$. For reporting purposes, we set the utility $u(\mathbf{x})$ of a recommendation \mathbf{x} to be $f(\mathbf{x})$ if \mathbf{x} satisfies the constraint, and otherwise a penalty value of the worst (largest) objective function value achievable in the search space. For each recommendation \mathbf{x} , we compute the utility gap $|u(\mathbf{x}) - u(\mathbf{x}_*)|$, where \mathbf{x}_* is the true solution to the optimization problem. Each method is initialized with the same three random points drawn with Latin hypercube sampling.

Figure 3(c) shows the median of the utility gap for each method for the 500 realizations of f and c_1 . The x -axis in this plot is the number of joint function evaluations for f and c_1 . We report the median because the empirical distribution of the utility gap is heavy-tailed and in this case the median is more representative of the location of the bulk of the data than the mean. The heavy tails arise because we are averaging over 500 different optimization problems with very different degrees of difficulty. In this and all of the following experiments, unless otherwise specified, error bars are computed using the bootstrap method. The plot shows that PESG and RS are better than RSDG. Furthermore, PESG is very similar to RS, with PESG even performing slightly better, perhaps because PESG is not confined to a grid as RS is. These results seem to indicate that PESG yields an accurate approximation of the information gain.

6.2 Synthetic Functions in 2 and 8 Input Dimensions

We compare the performance of PESG and RSDG with EIG using the same experimental protocol as in the previous section, but with dimensionalities $D = 2$ and $D = 8$. We do not compare with RS here because its use of grids does not scale to higher dimensions. Fig. 4 shows the utility gap for each method across 500 different samples of f and c_1 from the

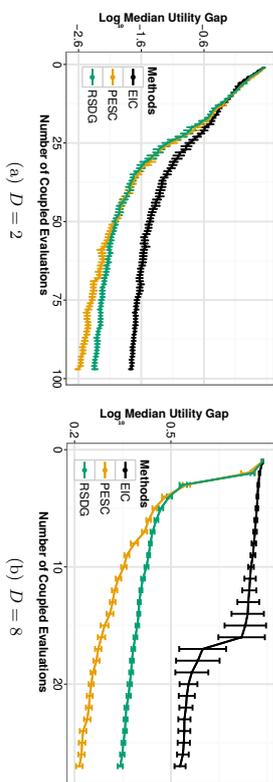


Figure 4: Optimizing samples from the GP prior with (a) $D = 2$ and (b) $D = 8$.

GP prior with (a) $D = 2$ and (b) $D = 8$. Overall, PESG is the best method, followed by RSDG and EIG. RSDG performs similarly to PESG when $D = 2$, but is significantly worse when $D = 8$. This shows that, when D is high, grid based approaches (e.g. RSDG) are at a disadvantage with respect to methods that do not require a grid (e.g. PESG).

6.3 A Toy Problem

Next, we compare PESG with EIG and AL (Gramacy et al. (2016), Section 2.4) on the toy problem described by Gramacy et al. (2016), namely,

$$\begin{aligned} \min_{\mathbf{x} \in [0,1]^2} f(\mathbf{x}) \text{ s.t. } c_1(\mathbf{x}) &\geq 0, c_2(\mathbf{x}) \geq 0, \\ f(\mathbf{x}) &= x_1 + x_2, \\ c_1(\mathbf{x}) &= 0.5 \sin(2\pi(x_1^2 - 2x_2)) + x_1 + 2x_2 - 1.5, \\ c_2(\mathbf{x}) &= -x_1^2 - x_2^2 + 1.5. \end{aligned} \quad (23)$$

This optimization problem has two local minimizers and one global minimizer. At the global solution, which is at $\mathbf{x}_* \approx [0.1954, 0.4404]$, only one of the two constraints (c_1) is active. Since the objective is linear and c_2 is not active at the solution, learning about c_1 is the main challenge of this problem. Fig. 5(a) shows a visualization of the linear objective function and the feasible and infeasible regions, including the location of the global constrained minimizer \mathbf{x}_* .

In this case, the evaluations for f , c_1 and c_2 are noise-free. To produce recommendations in PESG and EIG, we use the confidence value $\delta = 0.05$. We also use a squared exponential GP kernel. PESG uses $M = 10$ samples of \mathbf{x}_* when approximating the expectation in Eq. (8). We use the AL implementation provided by Gramacy et al. (2016) in the R package *laGP*, which is based on the squared exponential kernel and assumes the objective f is known. Thus, in order for this implementation to be used, AL has an advantage over other methods in that it has access to the true objective function. In all three methods, the GP hyperparameters are estimated by maximum likelihood.

Figure 5(b) shows the mean utility gap for each method across 500 repetitions. Each repetition corresponds to a different initialization of the methods with three data points

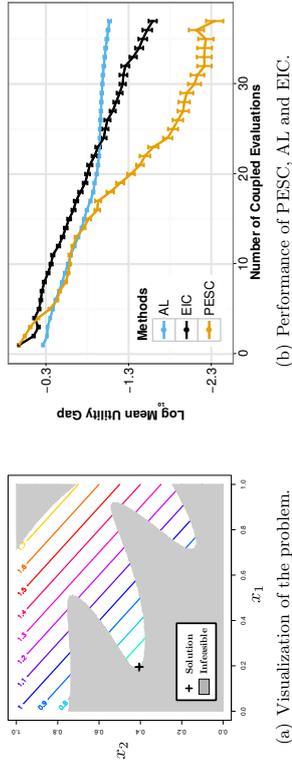


Figure 5: Comparing PESC, AL, and EIC in the toy problem described by Gramacy et al. (2016). (a) Visualization of the linear objective function and the feasible and infeasible regions. (b) Results obtained by PESC, AL and EIC on the toy problem.

selected with Latin hypercube sampling. The results show that PESC is significantly better than EIC and AL for this problem. EIC is superior to AL, which performs slightly better at the beginning, possibly because it has access to the ground truth objective f .

6.4 Finding a Fast Neural Network

In this experiment, we tune the hyper-parameters of a three-hidden-layer neural network subject to the constraint that the prediction time must not exceed 2 ms on an NVIDIA GeForce GTX 580 GPU (also used for training). We use the Matérn 5/2 kernel for the GP priors. The search space consists of 12 parameters: 2 learning rate parameters (initial value and decay rate), 2 momentum parameters (initial and final values, with linear interpolation), 2 dropout parameters (for the input layer and for other layers), 2 additional regularization parameters (weight decay and max weight norm), the number of hidden units in each of the 3 hidden layers, and the type of activation function (RELU or sigmoid). The network is trained using the *deepnet* package¹, and the prediction time is computed as the average time of 1000 predictions for mini-batches of size 128. The network is trained on the MNIST digit classification task with momentum-based stochastic gradient descent for 5000 iterations. The objective is reported as the classification error rate on the standard validation set. For reporting purposes, we treat constraint violations as the worst possible objective value (a classification error of 1.0). This experiment is inspired by a real need for neural networks that can make fast predictions with high accuracy. An example is given by computer vision problems in which the prediction time of the best performing neural network is not fast enough to keep up with the fast rate at which new data is available (e.g., YouTube, connectomics).

Figure 6(a) shows the results of 50 iterations of the Bayesian optimization process. In this experiment and in the next one, the y -axis represents the best objective value observed

1. <https://github.com/nitishsrivastava/deepnet>

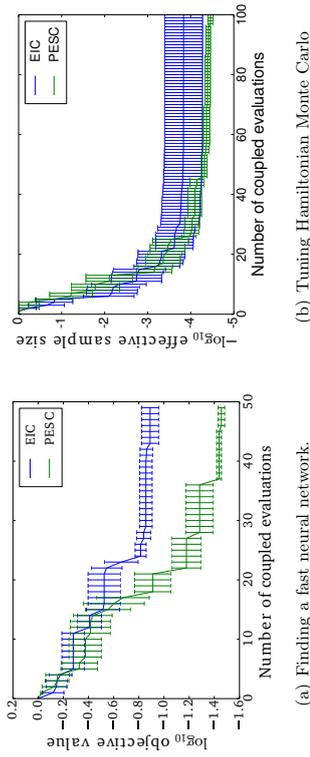


Figure 6: Results for PESC and EIC on the tuning of machine learning methods with coupled constraints. (a) Tuning a neural network subject to the constraint that it makes predictions in under 2 ms. (b) Tuning Hamiltonian Monte Carlo to maximize the number of effective samples within 5 minutes of compute time, subject to the constraints passing the Geweke and Gelman-Rubin convergence diagnostics and integrator stability.

so far, with recommendations produced using $\delta = 0.05$ and observed constraint violations resulting in objective values equal to 1.0. Curves show averages over five independent experiments. In this case, PESC performs significantly better than EIC.

When the constraints are noisy, reporting the best observation is an overly optimistic metric because the best feasible observation might be infeasible in practice. On the other hand, ground-truth is not available. Therefore, to validate our results further, we used the recommendations made at the final iteration of the Bayesian optimization process for each method (EIC and PESC) and evaluated the functions with these recommended parameters. We repeated the evaluation 10 times for each of the 5 repeated experiments. The result is a ground-truth score obtained as the average of 50 function evaluations. This procedure yields a score of $7.0 \pm 0.6\%$ for PESC and $49 \pm 4\%$ for EIC (as in the figure, constraint violations are treated as a classification error of 100%), where the numbers after the \pm symbol denote the empirical standard deviation. This result is consistent with Fig. 6(a) in that PESC performs significantly better than EIC.

6.5 Tuning Markov Chain Monte Carlo

Hamiltonian Monte Carlo (HMC) (Duane et al., 1987) is a popular MCMC technique that takes advantage of gradient information for rapid mixing. HMC contains several parameters that require careful tuning. The two basic parameters are the number of leapfrog steps and the step size. HMC may also include a mass matrix which introduces $\mathcal{O}(D^2)$ additional parameters for problems in D dimensions, although the matrix is often fixed to be diagonal (D parameters) or a multiple of the identity matrix (1 parameter) (Neal, 2011). In this experiment, we optimize the performance of HMC. We use again the Matérn 5/2 kernel for

the GP priors. We tune the following parameters: the number of leapfrog steps, the step size, a mass parameter and the fraction of the allotted computation time spent burning in the chain. Our experiment measures the number of effective samples obtained in a fixed computation time. We impose the constraints that the generated samples must pass the Geweke (Geweke, 1992) and Gelman-Rubin (Gelman and Rubin, 1992) convergence diagnostics. In particular, we require the worst (largest absolute value) Geweke test score across all variables and chains to be at most 2.0, and the worst (largest) Gelman-Rubin score between chains and across all variables to be at most 1.2. We use the *coda* R package (Plummer et al., 2006) to compute the effective sample size and the Geweke convergence diagnostic, and the *PyMCMC* python package (Patil et al., 2010) to compute the Gelman-Rubin diagnostic over two independent traces.

The HMC integration may also diverge for large values of the step size. We treat this as a hidden constraint, and set $\delta = 0.05$. We use HMC to sample from the posterior of a logistic regression binary classification problem using the German credit data set from the UCI repository (Frank and Asuncion, 2010). The data set contains 1000 data points, and is normalized to have zero mean unit variance for each feature. We initialize each chain randomly with $D = 25$ independent draws from a Gaussian distribution with mean zero and standard deviation 10^{-3} . For each set of inputs, we compute two chains, each one with five minutes of computation time on a single core of a compute node.

Figure 6(b) compares EIC and PESCG on this task, averaged over ten realizations of the experiment. As above, we perform a ground-truth assessment of the final recommendations. For each method (EIC and PESCG), we used the recommendations made at the final iteration of the Bayesian optimization process and evaluated the functions with these recommended parameters multiple times. The resulting average effective sample size is 3300 ± 1200 for PESCG and 2300 ± 900 for EIC, where the number after the \pm symbol denotes the empirical standard deviation. Here, the difference between the two methods is within the margin of error. When we compare these results with the ones in Fig. 6(b) we observe that the latter results are overly optimistic, indicating that this experiment is very noisy. The noise presumably comes from the randomness in the initialization and the execution of HMC, which causes the passing or the failure of the convergence diagnostics to be highly stochastic.

7. Empirical Analyses with Decoupled Functions

Section 6 focused on the evaluation of the performance of PESCG in experiments with coupled functions. Here, we evaluate the performance of PESCG in the decoupled case, where the different functions can be evaluated independently.

7.1 Accuracy of the PESCG Approximation

We first evaluate the accuracy of PESCG when approximating the function-specific acquisition functions from Eq. (13). We consider a synthetic problem with input dimension $D = 1$ and including an objective function and a single constraint function, both drawn from the GP prior distribution. Figure 7(a) shows the marginal posterior distributions for f and c_1 given 7 observations for the objective and 3 for the constraint. Figures 7(b) and 7(c) show the PESCG approximations to the acquisition functions for the objective and the constraint,

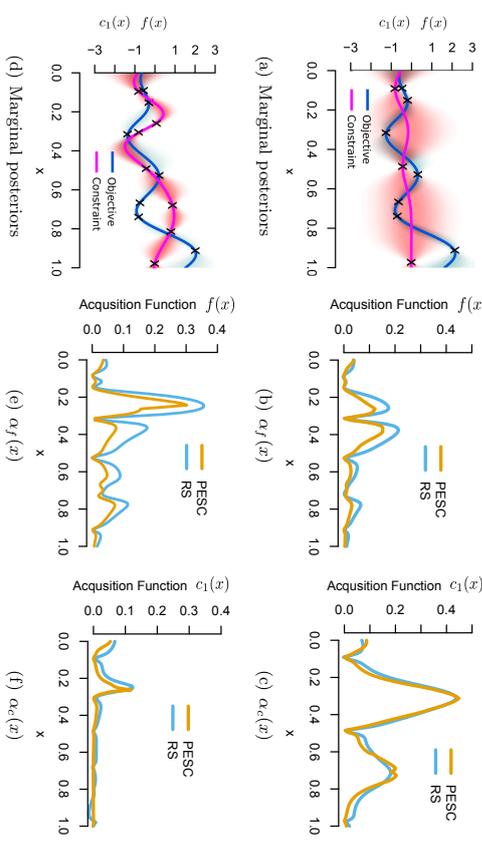


Figure 7: Assessing the accuracy of the decoupled PESCG approximation for the partial acquisition functions for $\alpha_f(x)$ and $\alpha_c(x)$. Between the top and bottom rows, three additional observations of the constraint have been made.

respectively. These functions approximate how much information we would obtain by the individual evaluation of the objective or the constraint at any given location. We also include in Figs. 7(b) and 7(c) the value of a ground truth obtained by rejection sampling (RS). The RS solution is obtained in the same way as in Section 6.1. Both PESCG and RS use a total of $M = 50$ samples from $p(\mathbf{x}_* | \mathcal{D})$. The PESCG approximation is quite accurate, and importantly its maximum value is very close to the maximum value of the RS approximation. Figures 7(b) and 7(c) indicate that the highest expected gain of information is obtained by evaluating the constraint at $x \approx 0.3$. The reason for this is that, as Fig. 7(a) shows, the objective is low near $x \approx 0.3$ but the constraint has not been evaluated at that location yet.

Figure 7(d) shows the marginal posterior distributions for f and c_1 when three more observations have been collected for the constraint. The corresponding approximations given by PESCG and RS to the function-specific acquisition functions are shown in Figs. 7(e) and 7(f). As before, the PESCG approximation is very similar to the RS one. In this case, evaluating the constraint is no longer as informative as before and the highest expected gain of information is obtained by evaluating the objective at $x \approx 0.25$. Intuitively, as we collect more constraint observations the constraint becomes well determined and the optimizer turns its attention to the objective.

7.2 Comparing Coupled and Decoupled PESC

We now compare the performance of coupled and decoupled versions of PESC in the same decoupled optimization problem. This allows us to empirically demonstrate the benefits of treating a decoupled problem as such.

We first consider the toy problem from Section 6.3 given by Eq. (23). We assume that there are three decoupled tasks: one for the objective and another one for each constraint function. We further assume that there is a single resource r with capacity $\omega_{\max}(r) = 3$. Each task requires to use resource r for its evaluation and the evaluation of each task takes always the same amount of time, which is assumed to be much larger than the BO computations. At each iteration resource r is used to evaluate 3 functions in parallel. We compare the performance of four versions of PESC, which differ in how they select the 3 parallel evaluations that will be performed at each iteration. The first method is a coupled approach (Coupled) which, at each iteration, evaluates jointly the three tasks at the same input. The second method is a non-competitive decoupling approach (NCD) which, at each iteration, evaluates all the different tasks once but not necessarily at the same input. This is equivalent to assuming that there are 3 resources with capacity 1 and each task can only be evaluated in one resource: the tasks do not have to compete because each one can only be evaluated in its corresponding resource. The third method is a competitive-decoupling approach (CD) which allows the different tasks to compete such that, at each iteration, three not necessarily unique functions are evaluated at three not necessarily unique locations. We also consider an implementation of CD that is not based on PESC and uses the EIC-D approach, as described in Section 2.5. We call this method EIC-CD. EIC-CD works like CD, with the difference that, at each step, we first determine the next evaluation location \mathbf{x} by maximizing the EIC acquisition function. After this, the next task to be evaluated at \mathbf{x} is chosen according to the expected reduction in the entropy of the global feasible minimizer \mathbf{x}_* . The original description of this method given by Gelbart et al. (2014) approximates the expected reduction in entropy using Monte Carlo sampling. This is in general computationally very expensive. To speed up EIC-CD, we replace the Monte Carlo sampling step by the approximation of the expected reduction in entropy given by PESC.

All the methods have to update the GP model, the posterior samples of \mathbf{x}_* , and the EP solutions just after collecting the data from resource r . However, the method CD and EIC-CD have to do two additional update operations after sending the first and the second evaluations to resource r , respectively. These updates correspond to step 11 in Algorithm 1 and they allow CD and EIC-CD to condition on pending evaluations that are not complete yet. In our experiments we use the Kriging believer approach, in which we pretend that the pending function evaluations have completed and returned the values of the GP predictive mean at those locations. This allows the methods CD and EIC-CD to update the GP model in a fast way, at the cost of ignoring uncertainty in the predictions of the GP model. The samples of \mathbf{x}_* and the EP solutions are, however, recomputed from scratch once the GP model has been updated. This can be expensive in practice. To address this problem we introduce the method CD-F, which works like CD, but replaces the full updates for the samples of \mathbf{x}_* and the EP solutions with the corresponding fast updates used by PESC-F in Section 5. Therefore, by comparing CD and CD-F, we can evaluate the loss in performance that is obtained by using the fast PESC-F updates. Note that CD-F uses the fast updates

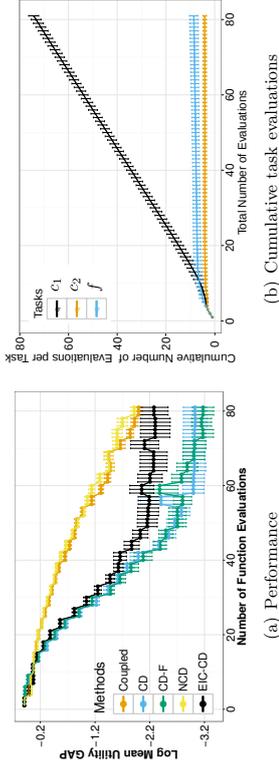


Figure 8: Results for the decoupled toy problem (Eq. (23)) when using a resource r that can evaluate 3 tasks (f , c_1 or c_2) in parallel. (a) Performance comparison of Coupled (orange), NCD (yellow), CD (blue), CD-F (green) and EIC-CD (black) approaches. (b) Cumulative number of evaluations for each task performed by CD-F. The algorithm automatically discovers that the constraint c_1 is much more important than the objective f or the other constraint c_2 .

only after sending the first and the second evaluations to resource r . Once the new data is collected, CD-F uses the original slow updates.

Figure 8(a) shows the results obtained by each method across 500 repetitions of the experiment starting from random initializations. Recommendations are computed with $\delta = 0.01$. The horizontal axis in the plot denotes the number of function evaluations performed so far. Since $\omega_{\max}(r) = 3$, these evaluations are performed in parallel in blocks of three. The vertical axis denotes the average utility gap, computed as in Section 6.1. Overall, CD and CD-F perform the best; the fact that CD and CD-F obtain similar results implies that the fast PESC-F updates incur no significant performance loss in this synthetic optimization problem. EIC-CD is worse than these two methods. This is a result of the sub-optimal two-stage decision process used by EIC-CD to select the next evaluation location and the next task to be evaluated at that location; see Section 2.5 for more details. NCD performs about the same as Coupled which means that, in this problem, the benefits of decoupling come from choosing an unequal distribution of tasks to evaluate, rather than from the additional freedom of evaluating the three tasks at potentially different locations. This hypothesis is corroborated by Fig. 8(b), which shows the average cumulative number of evaluations performed by CD for each task (f , c_1 or c_2) at each iteration. CD chooses to evaluate the constraint c_1 far more often than the objective or the other constraint c_2 . This makes sense since the objective is a linear function and c_1 , which has a complicated form, is the only active constraint at the global solution. Thus, the PESC algorithm has automatically discovered that the constraint c_1 is much more important (both in the sense of being complicated and in the sense of being active at the true solution) than the objective f or the other constraint c_2 . This demonstrates the true power of competitive decoupling,

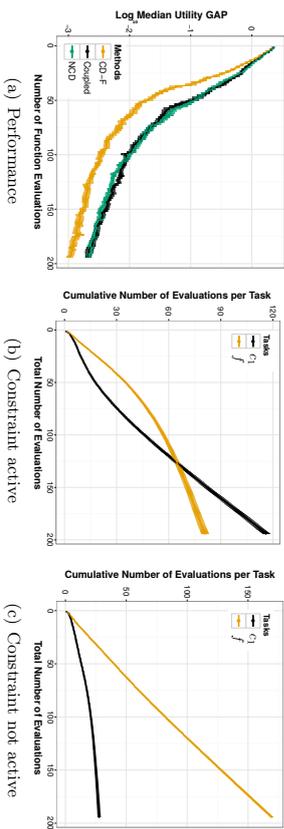


Figure 9: Results on synthetic problems with $D = 2$ sampled from the GP prior as in Section 6.2 when using a resource r that can evaluate 2 tasks (f or c_1) in parallel. (a) Performance comparison of Coupled (black), NCD (green), and CD-F (orange) approaches. (b) Cumulative number of task evaluations performed by CD-F when c_1 is active at the solution. (c) Cumulative number of task evaluations performed by CD-F when c_1 is not active at the solution.

as the algorithm avoids wasting time on uninteresting tasks that might be, in the worst case scenario, even more expensive than the interesting ones.

We perform another comparison of the methods Coupled, NCD and CD-F in synthetic problems in which the objective f and a single constraint function c_1 are drawn from the GP prior with $D = 2$. This is done in the same way as in Section 6.2. We set $\delta = 0.05$ and follow an experimental protocol similar to the one from the previous experiment: we assume that there are two tasks, given by f and c_1 , which can be evaluated at a resource r with capacity $\omega_{\max}(r) = 2$. Therefore, at any iteration we will be evaluating 2 tasks in parallel. Figure 9(a) shows the median utility gap obtained by each method across 500 different realizations of the experiment. As in the previous toy problem, CD-F outperforms Coupled, while Coupled performs similar to NCD. Again, decoupling is useful when we can choose the tasks to evaluate (CD-F) and evaluating the tasks at potentially different locations (NCD) does not seem to produce significant improvements with respect to the coupled approach.

In the previous toy problem, CD-F outperformed NCD and Coupled because it learned that evaluating the constraint c_1 is much more useful than evaluating the objective f or the constraint c_2 . We perform a similar analysis here by plotting the cumulative number of evaluations for each task performed by CD-F. We divide the 500 realizations into those cases in which the constraint c_1 is active at the true solution (Fig. 9(b)) and those in which c_1 is not active at the true solution (Fig. 9(c)). The plots in Figs. 9(b) and 9(c) show that when the constraint c_1 is active at the solution, CD-F chooses to evaluate c_1 much more frequently. By contrast, when the constraint is not active, c_1 is evaluated much less. Presumably, in the latter case c_1 need only be evaluated until it is determined that it is very unlikely to be active at the solution. After this point, further evaluations of c_1 are not

very informative. These results indicate that the task-specific acquisition functions used by PESC and given by Eq. (14) are able to successfully measure the usefulness of evaluating each different task.

7.3 Performance of PESC-F with Respect to Wall-clock Time

We now evaluate the performance of PESC with fast BO computations (PESC-F, Section 5) while considering the wall-clock time of each experiment. Again, we focus on the toy problem from Section 6.3 given by Eq. (23). To highlight what can go wrong in decoupled optimization problems, we will assume that evaluating the objective is instantaneous, evaluating c_1 takes 2 seconds, and evaluating c_2 takes 1 minute. Each of these functions forms a different task so that all of them can be evaluated independently. We also consider that there is a single resource r with $\omega_{\max}(r) = 1$, that is, only one task can be evaluated at any given time with no possible parallelism. This setup corresponds to the competitive decoupling scenario from Fig. 1. We limit each experiment time to 15 minutes and consider the following methods: Coupled, and competitive decoupled (CD) with PESC-F and rationality levels $\gamma = \{\infty, 1, 0.1, 0\}$. According to Eq. (22), setting $\gamma = \infty$ is simply another way of saying that fast BO computations are not used.

Figure 10(a) shows the average utility gap of each method as a function of elapsed time. The coupled approach is the worst performing one, being outperformed by all the versions of PESC-F with different γ . This illustrates the advantages of the decoupled approach. The performance of PESC-F is improved as γ moves from ∞ to 1 and then to 0.1. The reason for this is that, as γ is reduced, less time is spent in the BO computations and more time is spent in the actual collection of data. However, reducing γ too much is detrimental as $\gamma = 0$ performs significantly worse than $\gamma = 0.1$ and $\gamma = 1$. The reason for this is that $\gamma = 0$ performs too many fast BO computations, which produce suboptimal decisions.

Figures 10(b) to 10(f) and Section 7.3 are useful to understand the results obtained by the different methods in Fig. 10(a). These figures show, for each method, the cumulative number of evaluations per task as a function of the elapsed time. The coupled approach performs very few evaluations of the different tasks. The reason for this is that it always evaluates all the tasks the same number of times and this leads to wasting a lot of time by evaluating too often the slowest task, that is, constraint c_2 , which is not very informative about the solution to the problem. The different versions of PESC-F with $\gamma = \{\infty, 1, 0.1, 0\}$ evaluate more often the most informative task, that is, c_1 and less frequently all the other tasks. As the rationality level γ is decreased, less time is spent in the BO computations, and thus more task evaluations are performed. These correspond to increases in performance. However, this trend does not continue indefinitely as γ is decreased. When $\gamma = 0$, performance is significantly diminished. By not performing slow BO computations, the $\gamma = 0$ method is not able to learn that c_2 is uninformative and continues to spend time evaluating it, thus performing many fewer evaluations of the most informative task c_1 . The configuration files for running this experiment are available at <https://github.com/HIPS/Spearmint/tree/PESC/examples/toy-fast-slow>.

Section 7.3 shows the time spent by each method in fast and slow BO computations and in the evaluation of tasks c_1 and c_2 . We do not include the time spent in the evaluation of task f because it is always zero. Note that the total time spent in the BO computations

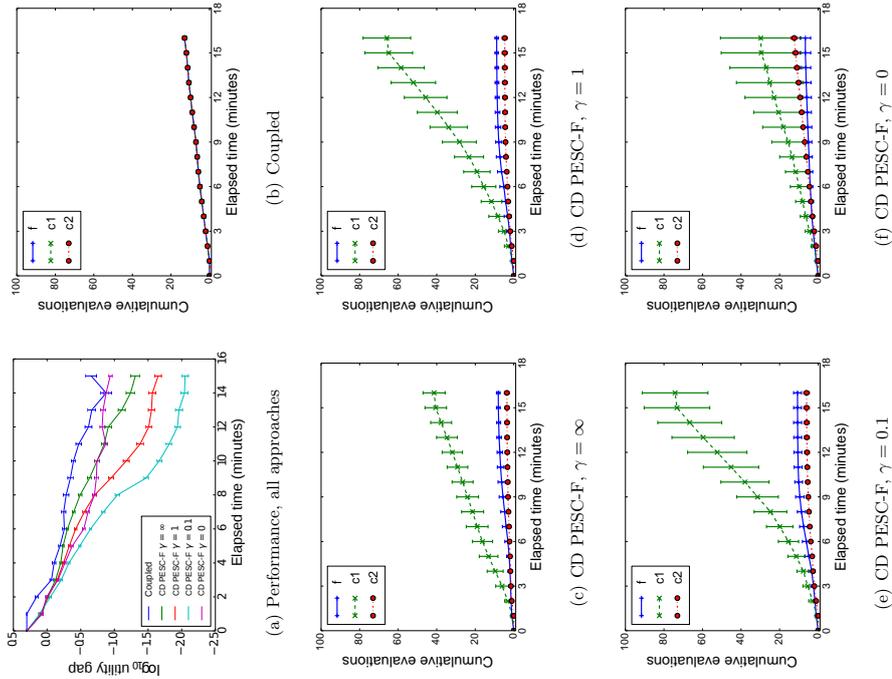


Figure 10: Results for Coupled and CD PESC-F with $\gamma = \{\infty, 1, 0.1, 0\}$ on the toy problem given by Eq. (23). Evaluations of f are instantaneous, evaluations of c_1 take 2 sec. and evaluations of c_2 take 1 min. The maximum experiment time is 15 min. (a) log utility gap versus wall-clock time. (b-f) Cumulative function evaluations for (b) Coupled, (c) CD PESC-F with $\gamma = \infty$ (no fast computations), (d) CD PESC-F with $\gamma = 1.0$, (e) CD PESC-F with $\gamma = 0.1$, and (f) CD PESC-F with $\gamma = 0$ (no slow computations). Curves reflect the mean over 100 trials. Error bars in (b-f) are given by the empirical standard deviations.

Method	Slow BO Comp.	Fast BO Comp.	Total BO Comp.	$c_1(\mathbf{x})$	$c_2(\mathbf{x})$	Total Evaluation
Coupled	2.0	0.0	2.0	0.4	13.0	13.4
CD PESC-F, $\gamma = \infty$	10.2	0.0	10.2	1.4	3.6	5.0
CD PESC-F, $\gamma = 1$	5.2	3.0	8.2	2.2	4.7	6.9
CD PESC-F, $\gamma = 0.1$	1.5	5.1	6.6	2.5	6.0	8.5
CD PESC-F, $\gamma = 0.0$	0.2	1.8	2.0	1.0	12.5	13.5

Table 1: Time spent by each method in BO computations and in task evaluations. For each method, the table reports the mean time in minutes, over 100 independent runs, spent in fast and slow BO computations and in the evaluation of tasks c_1 and c_2 .

and in the evaluation of the different tasks does not add up exactly to 15 minutes because, in our implementation, the current iteration is allowed to finish after the 15-minute mark is reached. As expected, the time spent in the BO computations decreases monotonically as γ is decreased. The coupled approach spends a small amount of time doing BO computations. The reason for this is that this method does not have to perform step 11 in Algorithm 1 and step 4 is performed less frequently than in the PESC-F methods because Coupled spends most of its time in the evaluation of c_2 .

The fifth column in Section 7.3 corresponds to the time spent in the evaluation of c_1 . The entries in this column are indicative of the relative performances of each method, since c_1 is the most important function in this optimization problem. From these entries we may conclude that this problem exhibits an optimal value of γ close to 0.1. This represents an optimal ratio of time spent in the BO computations to time spent in the evaluation of the different tasks. We leave to future work the issue of selecting the optimal value for γ . In a highly sophisticated approach this could be done in an online fashion by using reinforcement learning.

8. Conclusions and Future Work

We have presented a general framework for solving Bayesian optimization (BO) problems with unknown constraint functions. In these problems the objective and the constraints can only be evaluated via expensive queries to black boxes that may provide noisy values. Our framework allows for problems in which the objective and the constraints can be split into subsets of functions that require *coupled* evaluation, meaning that these functions have to be jointly evaluated at the same input. We call these subsets of coupled functions *tasks*. Different tasks may, however, be evaluated independently at different locations, that is, in a *decoupled* way. Furthermore, the tasks may or may not compete for a limited set of resources during their evaluation. Based on this, we have then introduced the notions of *competitive decoupling* (CD), where two or more tasks compete for the same resource, and *non-competitive decoupling* (NCD), where the tasks require to use different resources and can therefore be evaluated in parallel. The notion of *parallel* BO is a special case in which

one task requires a specific resource, of which many instances are available. We have then presented a general procedure, given by Algorithm 1, to solve problems with an arbitrary combination of coupling and decoupling. This algorithm receives as input a bipartite graph \mathcal{G} whose nodes are resources and tasks and whose edges connect each task with the resource at which it can be evaluated. Algorithm 1 relies on an acquisition function that can measure the utility of evaluating any arbitrary subset of functions, that is, of any possible task. An acquisition function that satisfies this requirement is said to be *separable*.

To implement Algorithm 1, we have proposed a new information-based approach called Predictive Entropy Search with Constraints (PESC). At each iteration, PESC collects data at the location that is expected to provide the highest amount of information about the solution to the optimization problem. By introducing a factorization assumption, we obtain an acquisition function that is additive over the subset of functions to be evaluated. That is, the amount of information that we approximately gain by jointly evaluating a set of functions is equal to the sum of the gains of information that we approximately obtain by the individual evaluation of each of the functions. This property means that the acquisition function of PESC is separable. Therefore, PESC can be used to solve general constrained BO problems with decoupled evaluation, something that has not been previously addressed.

We evaluated the performance of PESC in coupled problems, where all the functions (objective and constraints) are always jointly evaluated at the same input location. This is the standard setting considered by most prior approaches to constrained BO. The results of our experiments show that PESC achieves state-of-the-art results in this scenario. We also evaluated the performance of PESC in the decoupled setting, where the different tasks can be evaluated independently at arbitrary input locations. We considered scenarios with competition (CD) and with non-competition (NCD) and compared the performances of two versions of PESC: one with decoupling (decoupled PESC) and another one that always performs coupled evaluations (coupled PESC). Decoupled PESC is significantly better than coupled PESC when there is competition, that is, in the CD setting. The reason for this is that some functions can be more informative than others and decoupled PESC exploits this to make optimal decisions when deciding which function to evaluate next with limited resources. In particular, decoupled PESC avoids wasting time in function evaluations that are unlikely to improve the current estimate of the solution to the optimization problem. However, when there is no competition, that is, in the NCD setting, coupled and decoupled PESC perform similarly. Therefore, in our experiments, the main advantages of considering decoupling seem to come from choosing an unequal distribution of tasks to evaluate, rather than from the additional freedom of evaluating the different tasks at potentially arbitrary locations. In our experiments we have assumed that the evaluation of all the functions takes the same amount of time. However, NCD is expected to perform better than the coupled approach in other settings in which some functions are much faster to evaluate than others. Evaluating the performance of NCD in these settings is left as future work.

For the BO approach to be useful, the time spent performing BO computations (such as computing and globally optimizing the acquisition function) has to be significantly shorter than the time spent collecting data. However, decoupled optimization problems may include some tasks that are fast to evaluate. When these tasks are informative and their evaluation time is comparable to that of the BO computations, the BO approach may be inefficient. To address this issue, we follow a bounded rationality approach and introduce additional

approximations in the computations made by PESC to reduce their cost when necessary. The new method is called PESC-F and it is able to automatically switch between fast, but approximate, and slow, but accurate, operations. A parameter called the *rationality level* is used in PESC-F to balance the amount of time that is spent in the BO computations and in the actual collection of data. Experiments with wall-clock time in a CD scenario show that PESC-F can be significantly better than the original version of PESC.

In summary, PESC is an effective algorithm for BO problems with unknown constraints and the separability of its acquisition function makes it a promising direction towards a unified solution for constrained BO problems. As new acquisition functions are proposed in the future, they will hopefully be developed with separability in mind as an important and desirable property.

The code for PESC, including decoupling and PESC-F, is available in PESC branch of the open-source Bayesian optimization package *Spearmint* at <https://github.com/HIPS/Spearmint/tree/PESC>.

One potential line of future work includes extensions to settings where tasks require more than one resource to run. This could, for example, be formalized using a framework similar to the one presented in Section 3, but where the resource dependencies for each task t are represented as a set of edges $\mathcal{E}_t = \{t \sim r\}$ for the i th potential allocation of resources. This can be interpreted as the statement that all resource nodes $V_t = \{r : (t \sim r) \in \mathcal{E}_t\}$ are required in order to initiate task t using allocation i . Note that the union of these edges now specifies a multigraph with edges $\mathcal{E} = \bigcup_i \mathcal{E}_i$ due to the fact there may be resources that are required across multiple allocations of a particular task. This also modifies the pseudo-code for Algorithm 1 where the loop over resources r becomes a loop over potential allocations such that $\omega(r) < \omega_{\max}(r)$ for $r \in V_t$ for some (t, i) pair. In the case where each task requires only a single resource this reduces to the earlier formulation. Another possibility is for allocations where the resources are time or iteration dependent. This would require some form of temporal planning. To make such a procedure feasible, however, it may be necessary to consider greedy decisions at each point in time.

Another direction for future work is concerned with the use of *bounded rationality* in Bayesian optimization. Here we have used a simple heuristic for selecting between two levels (fast and slow) of computations in PESC-F. However, we could consider a larger number of levels with increasingly more accurate computations (Hay et al., 2012). The Bayesian optimization algorithm would then have to optimally select one of these to determine the next evaluation location. We could also consider different modeling approaches for the collected data, with different trade-offs between accuracy and computational cost. We also leave for future work a theoretical analysis of PESC. This would be in the line of the work of Russo and Van Roy (2014) on information-directed sampling. However, they use simpler models for the data and do not consider problems with constraints.

Finally, we would like to point out that the approach described here can be applied in a straightforward manner to address multi-objective Bayesian optimization problems (Knowles, 2006). In the multi-objective case the different tasks would be given by groups of objective functions that have to be evaluated in a coupled manner. An extension of PESC for working with multiple objectives is given by Hernández-Lobato et al. (2016).

Acknowledgments

José Miguel Hernández-Lobato acknowledges support from the Rafael del Pino Foundation. Zoubin Ghahramani acknowledges support from Google Focused Research Award and EP-SRC grant EP/I036575/1. Matthew W. Hoffman acknowledges support from EPSRC grant EP/J012300/1.

Appendix A. The Expectation Propagation Method Used by PESC

We describe here the expectation propagation (EP) method that is used by PESC to adjust a Gaussian approximation to the non-Gaussian factor $f(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K | \mathbf{x}^j)$ in Eq. (17). This is done after replacing the infinite set \mathcal{X} with the finite set \mathcal{Z} , which contains only the locations at which the objective f has been evaluated so far, the value of \mathbf{x}^j and \mathbf{x} . Recall that \mathbf{x} is the input to the acquisition function, that is, it contains the location at which we are planning to evaluate $f, \mathbf{c}_1, \dots, \mathbf{c}_K$. When \mathcal{X} is replaced with \mathcal{Z} we have that the vectors $\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K$ contain now the result of the noise-free evaluations of $f, \mathbf{c}_1, \dots, \mathbf{c}_K$ at \mathcal{Z} , that is,

$$\mathbf{f} = [f(\mathbf{x}_f^1), \dots, f(\mathbf{x}_f^{N_f}), f(\mathbf{x}^j), f(\mathbf{x})]^T, \quad (24)$$

$$\mathbf{c}_k = [c_k(\mathbf{x}_f^1), \dots, c_k(\mathbf{x}_f^{N_f}), c_k(\mathbf{x}^j), c_k(\mathbf{x})]^T, \quad \text{for } k = 1, \dots, K. \quad (25)$$

where $\mathbf{x}_f^1, \dots, \mathbf{x}_f^{N_f}$ are the locations at which the objective f has been evaluated so far. That is, the first N_f entries in $\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K$ contain the function values at the locations for which there is data for the objective. These entries are then followed by the function values at \mathbf{x}^j and at \mathbf{x} . When we replace \mathcal{X} with \mathcal{Z} we have that

$$f(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K | \mathbf{x}^j) = p(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K | \mathcal{D}) \Gamma(\mathbf{x}^j) \left\{ \prod_{i=1}^{N_f} \Psi(\mathbf{x}_f^i) \right\} \Psi(\mathbf{x}). \quad (26)$$

In this expression we should have included a factor $\Psi(\mathbf{x}^j)$ since $\mathbf{x}^j \in \mathcal{Z}$. We ignore such factor because it is always equal to 1 according to Eq. (16). In Eq. (26) we have separated the non-Gaussian factor that depends on \mathbf{x} , that is, $\Psi(\mathbf{x})$ from those factors that do not depend on \mathbf{x} , that is, $\Gamma(\mathbf{x}^j), \Psi(\mathbf{x}_f^1), \dots, \Psi(\mathbf{x}_f^{N_f})$. All these non-Gaussian factors are approximated with Gaussians using EP.

Finding the next *suggestion* involves maximizing the acquisition function. This requires to evaluate the acquisition function at many different \mathbf{x} and recomputing the complete EP approximation for each value of \mathbf{x} can be excessively expensive. To avoid this, we first compute the EP approximation for the factors that do not depend on \mathbf{x} in isolation, store it and then reuse it as we compute the EP approximation for the remaining factors. Since most of the factors do not depend on \mathbf{x} , this leads to large speedups when we have to evaluate the acquisition function at many different \mathbf{x} . Therefore, we start by finding a Gaussian approximation to the factors that do not depend on \mathbf{x} , that is, $\Gamma(\mathbf{x}^j), \Psi(\mathbf{x}_f^1), \dots, \Psi(\mathbf{x}_f^{N_f})$, while ignoring the other factor that does depend on \mathbf{x} , that is, $\Psi(\mathbf{x})$.

A.1 Approximating the Non-Gaussian Factors that do not Depend on \mathbf{x}

We use EP to find a Gaussian approximation to $\Gamma(\mathbf{x}^j), \Psi(\mathbf{x}_f^1), \dots, \Psi(\mathbf{x}_f^{N_f})$ in Eq. (26) when $\Psi(\mathbf{x}_1), \dots, \Psi(\mathbf{x}_{K+1})$ are assumed to be constant and equal to 1. Because the data is assumed to be generated from independent GPs, we have that $p(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K | \mathcal{D})$ in Eq. (26) is

$$p(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K | \mathcal{D}) = \mathcal{N}(\mathbf{f} | \mathbf{m}_1^{\text{pred}}, \mathbf{V}_1^{\text{pred}}) \prod_{k=1}^K \left\{ \mathcal{N}(\mathbf{c}_k | \mathbf{m}_{k+1}^{\text{pred}}, \mathbf{V}_{k+1}^{\text{pred}}) \right\}, \quad (27)$$

where $\mathbf{m}_1^{\text{pred}}$ and $\mathbf{V}_1^{\text{pred}}$ are the mean and covariance matrix of the posterior distribution of \mathbf{f} given the data for the objective and $\mathbf{m}_{k+1}^{\text{pred}}$ and $\mathbf{V}_{k+1}^{\text{pred}}$ are the mean and covariance matrix of the posterior distribution of \mathbf{c}_k given the data for constraint k . In particular, from Eqs. (2.22) to (2.24) of (Rasmussen and Williams, 2006) we have that

$$\mathbf{m}_i^{\text{pred}} = \mathbf{K}_*^i (\mathbf{K}_*^i + \nu_i \mathbb{I})^{-1} \mathbf{y}^i, \quad (28)$$

$$\mathbf{V}_i^{\text{pred}} = \mathbf{K}_*^i - \mathbf{K}_*^i (\mathbf{K}_*^i + \nu_i \mathbb{I})^{-1} [\mathbf{K}_*^i]^T, \quad \text{for } i = 1, \dots, K+1, \quad (29)$$

where \mathbf{y}^i is an N_i -dimensional vector with the data for the i -th function in $\{f, \mathbf{c}_1, \dots, \mathbf{c}_K\}$, \mathbf{K}_*^i is an $(N_i + 2) \times N_i$ matrix with the prior cross-covariances between the entries of the i -th vector in $\{f, \mathbf{c}_1, \dots, \mathbf{c}_K\}$ and the value of the corresponding function at the locations for which there is data available for that function and $\mathbf{K}_*^{i,*}$ is an $(N_i + 2) \times (N_i + 2)$ matrix with the prior covariances between entries of the i -th vector in $\{f, \mathbf{c}_1, \dots, \mathbf{c}_K\}$ and ν_i is the noise variance at the black-box for the i -th function in $\{f, \mathbf{c}_1, \dots, \mathbf{c}_K\}$.

The exact factors $\Gamma(\mathbf{x}^j), \Psi(\mathbf{x}_f^1), \dots, \Psi(\mathbf{x}_f^{N_f})$ from Eq. (26) are then approximated with the corresponding Gaussian factors $\tilde{\Gamma}(\mathbf{x}^j), \tilde{\Psi}(\mathbf{x}_f^1), \dots, \tilde{\Psi}(\mathbf{x}_f^{N_f})$. Let $\beta_n(\mathbf{f}) = [f(\mathbf{x}_f^1), f(\mathbf{x}_f^2)]^T$, where \mathbf{x}_f^i is the n -th location for which there is data for the objective f . Then, we define

$$\tilde{\Psi}(\mathbf{x}_f^j) \propto \exp \left\{ -\frac{1}{2} \beta_n(\mathbf{f})^T \tilde{\mathbf{A}}_n \beta_n(\mathbf{f}) + \beta_n(\mathbf{f})^T \tilde{\mathbf{b}}_n \right\} \prod_{k=1}^K \exp \left\{ -\frac{1}{2} c_k(\mathbf{x}_f^j)^2 \tilde{d}_n^k + c_k(\mathbf{x}_f^j) \tilde{e}_n^k \right\}, \quad (30)$$

where $\tilde{\mathbf{A}}_n$ and $\tilde{\mathbf{b}}_n$ are the natural parameters of a bivariate Gaussian distribution on $\beta_n(\mathbf{f})$ and \tilde{d}_n^k and \tilde{e}_n^k are the natural parameters of a Gaussian distribution on $c_k(\mathbf{x}_f^j)$, that is, the value of constraint k at the n -th location for which there is data for the objective. We also define

$$\tilde{\Gamma}(\mathbf{x}^j) \propto \prod_{k=1}^K \exp \left\{ -\frac{1}{2} c_k(\mathbf{x}^j) \tilde{g}_k + c_k(\mathbf{x}^j) \tilde{h}_k \right\},$$

where \tilde{g}_k and \tilde{h}_k are the natural parameters of a Gaussian distribution on $c_k(\mathbf{x}^j)$, that is, the value of constraint k at the current posterior sample of \mathbf{x} .

The parameters $\tilde{\mathbf{A}}_n, \tilde{\mathbf{b}}_n, \tilde{d}_n^k, \tilde{e}_n^k, \tilde{g}_k$ and \tilde{h}_k are fixed by running EP. Once the value of these parameters has been fixed, we replace the exact factors $\Gamma(\mathbf{x}^j), \Psi(\mathbf{x}_f^1), \dots, \Psi(\mathbf{x}_f^{N_f})$ in Eq. (26) with their corresponding Gaussian approximations to obtain an approximation to $f(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K | \mathbf{x}^j)$. We denote this approximation by $q(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K)$, where

$$q(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K) \propto p(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K | \mathcal{D}) \tilde{\Gamma}(\mathbf{x}^j) \left\{ \prod_{i=1}^{N_f} \tilde{\Psi}(\mathbf{x}_f^i) \right\} \left\{ \prod_{k=1}^{K+1} \tilde{\Psi}(\mathbf{x}_k) \right\}. \quad (31)$$

Since the approximate factors are Gaussian and $p(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K | \mathcal{D})$ is also Gaussian, we have that $q(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K)$ is also Gaussian:

$$q(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K) = \mathcal{N}(\mathbf{f} | \mathbf{m}_1, \mathbf{V}_1) \prod_{k=1}^K \mathcal{N}(c_k | \mathbf{m}_{k+1}, \mathbf{V}_{k+1}), \quad (32)$$

where, by applying the formula for products of Gaussians, we obtain

$$\mathbf{V}_i = \left[\mathbf{V}_i^{\text{pred}} \right]^{-1} + \tilde{\mathbf{S}}_i^{-1}, \quad (33)$$

$$\mathbf{m}_i = \mathbf{V}_i \left[\mathbf{V}_i^{\text{pred}} \right]^{-1} \mathbf{m}_i^{\text{pred}} + \tilde{\mathbf{t}}_i, \quad \text{for } i = 1, \dots, K+1, \quad (34)$$

with the following definitions for $\tilde{\mathbf{S}}_i$ and $\tilde{\mathbf{t}}_i$:

- $\tilde{\mathbf{S}}_1$ is an $(N_1 + 2) \times (N_1 + 2)$ matrix whose non-zero entries are
 - $[\tilde{\mathbf{S}}_1]_{h,n} = [\mathbf{A}_n]_{1,1}$ for $n = 1, \dots, N_1$,
 - $[\tilde{\mathbf{S}}_1]_{N_1+1,n} = [\tilde{\mathbf{S}}_1]_{n,N_1+1} = [\mathbf{A}_n]_{1,2}$ for $n = 1, \dots, N_1$,
 - $[\tilde{\mathbf{S}}_1]_{N_1+1,N_1+1} = \sum_{n=1}^{N_1} [\mathbf{A}_n]_{2,2}$.
- $\tilde{\mathbf{S}}_{k+1}$, for $k = 1, \dots, K$, is an $(N_1 + 2) \times (N_1 + 2)$ matrix whose non-zero entries are
 - $[\tilde{\mathbf{S}}_{k+1}]_{n,n} = d_n$ for $n = 1, \dots, N_1$,
 - $[\tilde{\mathbf{S}}_{k+1}]_{N_1+1,N_1+1} = g_n$ for $n = 1, \dots, N_1$.
- $\tilde{\mathbf{t}}_1$ is an $(N_1 + 2)$ -dimensional vector whose non-zero entries are
 - $[\tilde{\mathbf{t}}_1]_n = [\tilde{\mathbf{b}}_n]_1$ for $n = 1, \dots, N_1$,
 - $[\tilde{\mathbf{t}}_1]_{N_1+1} = \sum_{n=1}^{N_1} [\tilde{\mathbf{b}}_n]_2$.
- $\tilde{\mathbf{t}}_{k+1}$, for $k = 1, \dots, K$, is an $(N_1 + 2)$ -dimensional vector whose non-zero entries are
 - $[\tilde{\mathbf{t}}_{k+1}]_n = \tilde{c}_n^k$ for $n = 1, \dots, N_1$,
 - $[\tilde{\mathbf{t}}_{k+1}]_{N_1+1} = \tilde{h}_k$.

We now explain how to obtain the values of all the $\tilde{\mathbf{A}}_n$, $\tilde{\mathbf{b}}_n$, \tilde{d}_n^k , \tilde{c}_n^k , \tilde{g}_n , \tilde{h}_k and \tilde{h}_k using EP.

A.1.1 ADJUSTING $\tilde{\Psi}(\mathbf{x}_i^j)$ BY EP

We explain how to adjust the parameters $\tilde{\mathbf{A}}_n$, $\tilde{\mathbf{b}}_n$, \tilde{d}_n^k and \tilde{c}_n^k of the approximate factor $\tilde{\Psi}(\mathbf{x}_i^j)$ using EP. EP performs this operation by minimizing the following Kullback-Leibler divergence:

$$\text{KL}[\Psi(\mathbf{x}_i^j) q^{-n}(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K) \| \tilde{\Psi}(\mathbf{x}_i^j) q^{-n}(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K)], \quad (35)$$

where $q^{-n}(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K)$ is the cavity distribution given by

$$q^{-n}(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K) = q(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K) [\tilde{\Psi}(\mathbf{x}_i^j)]^{-1}, \quad (36)$$

If we marginalize out all variables except those which $\tilde{\Psi}(\mathbf{x}_i^j)$ depends on, namely $\beta_n(\mathbf{f})$ and $c_1(\mathbf{x}_i^j), \dots, c_K(\mathbf{x}_i^j)$, then q^{-n} takes the form

$$q^{-n}[\beta_n(\mathbf{f}), c_1(\mathbf{x}_i^j), \dots, c_K(\mathbf{x}_i^j)] \propto \mathcal{N}(\beta_n(\mathbf{f}) | \mathbf{b}^{-n}, \mathbf{A}^{-n}) \left\{ \prod_{k=1}^K \mathcal{N}(c_k | \mathbf{x}_i^j) \right\} [e_k^{-n}, d_k^{-n}], \quad (37)$$

where the parameters \mathbf{b}^{-n} , \mathbf{A}^{-n} , e_k^{-n} and d_k^{-n} of these Gaussian distributions are obtained from the ratio of q and $\tilde{\Psi}(\mathbf{x}_i^j)$ by using the formula for dividing Gaussians:

$$\mathbf{A}^{-n} = \left\{ \mathbf{V}_{\beta_n(\mathbf{f})}^{-1} - \tilde{\mathbf{A}}_n \right\}^{-1}, \quad \mathbf{b}^{-n} = \mathbf{A}^{-n} \left\{ \mathbf{V}_{\beta_n(\mathbf{f})}^{-1} \mathbf{m}_{\beta_n(\mathbf{f})} - \tilde{\mathbf{b}}_n \right\}, \quad (38)$$

$$d_k^{-n} = \left\{ v_{c_k(\mathbf{x}_i^j)}^{-1} - \tilde{d}_k \right\}^{-1}, \quad e_k^{-n} = d_k^{-n} \left\{ v_{c_k(\mathbf{x}_i^j)}^{-1} m_{c_k(\mathbf{x}_i^j)} - \tilde{e}_k \right\}^{-1}, \quad (39)$$

where $\mathbf{V}_{\beta_n(\mathbf{f})}$ is the 2×2 covariance matrix for $\beta_n(\mathbf{f})$ given by $q(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K)$ in Eq. (32), $\mathbf{m}_{\beta_n(\mathbf{f})}$ is the corresponding 2-dimensional mean vector, $v_{c_k(\mathbf{x}_i^j)}$ is the variance for $c_k(\mathbf{x}_i^j)$ given by $q(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K)$ in Eq. (32) and $m_{c_k(\mathbf{x}_i^j)}$ is the corresponding mean parameter.

To minimize Eq. (35) we match the 1st and 2nd moments of $\Psi(\mathbf{x}_i^j) q^{-n}(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K)$ and $\tilde{\Psi}(\mathbf{x}_i^j) q^{-n}(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K)$. The moments of $\Psi(\mathbf{x}_i^j) q^{-n}(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K)$ can be obtained from the derivatives of the logarithm of its normalization constant Z , which is given by

$$Z = \int \Psi(\mathbf{x}_i^j) q^{-n}(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K) d\mathbf{f} dc_1 \dots dc_K = \Phi(\alpha_n) \prod_{k=1}^K \Phi[\alpha_n^k] + 1 - \prod_{k=1}^K \Phi[\alpha_n^k], \quad (40)$$

where $\alpha_n^k = m_{c_k(\mathbf{x}_i^j)} v_{c_k(\mathbf{x}_i^j)}^{-1/2}$ and $\alpha_n = [1, -1] \mathbf{m}_{\beta_n(\mathbf{f})} ([1, -1] \mathbf{V}_{\beta_n(\mathbf{f})} [1, -1]^\top)^{-1/2}$ and Φ is the standard Gaussian cdf. We follow Eqs. (5.12) and (5.13) in (Minka, 2001b) to update \tilde{d}_n^k and \tilde{c}_n^k in Eq. (30). However, we use the second partial derivative with respect to e_k^{-n} rather than first partial derivative with respect to d_k^{-n} for numerical robustness. These derivatives are given by

$$\frac{\partial \log Z}{\partial e_k^{-n}} = \frac{(Z-1) \phi(\alpha_n^k)}{Z \Phi(\alpha_n^k) \sqrt{d_k^{-n}}}, \quad \frac{\partial^2 \log Z}{\partial [e_k^{-n}]^2} = -\frac{\partial \log Z}{\partial e_k^{-n}} \cdot \frac{\alpha_n^k}{\sqrt{d_k^{-n}}} - \left[\frac{\partial \log Z}{\partial e_k^{-n}} \right]^2, \quad (41)$$

where ϕ is the standard Gaussian pdf. The update equations for the parameters \tilde{d}_n^k and \tilde{c}_n^k of the approximate factor $\tilde{\Psi}(\mathbf{x}_i^j)$ are then

$$[\tilde{d}_n^k]_{\text{new}} = - \left\{ \frac{\partial^2 \log Z}{\partial [e_k^{-n}]^2} + d_k^{-n} \right\}^{-1}, \quad [\tilde{c}_n^k]_{\text{new}} = \left\{ d_k^{-n} - \left[\frac{\partial^2 \log Z}{\partial [e_k^{-n}]^2} \right]^{-1} \frac{\partial \log Z}{\partial e_k^{-n}} \right\} [\tilde{d}_n^k]_{\text{new}}, \quad (42)$$

We now perform the analogous operations to update $\tilde{\mathbf{A}}_n$ and $\tilde{\mathbf{b}}_n$. We need to compute

$$\frac{\partial \log Z}{\partial \mathbf{b}^{-n}} = \frac{\left\{ \prod_{k=1}^K \Phi[\alpha_n^k] \right\} \phi(\alpha_n)}{Z \sqrt{s}} [1, -1], \quad (43)$$

$$\frac{\partial \log Z}{\partial \mathbf{A}^{-n}} = -\frac{1}{2} [1, -1]^\top [1, -1] \frac{\left\{ \prod_{k=1}^K \Phi[\alpha_n^k] \right\} \phi(\alpha_n) \alpha_n}{Z s}, \quad (44)$$

where $s = [-1, 1] \mathbf{A}^{-m} [-1, 1]^\top$. We then compute the mean vector and covariance matrix for $\beta_n(\mathbf{f})$ with respect to $\Psi(\mathbf{x}_k^j) q^{-m}(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K)$:

$$[\mathbf{V}_{\beta_n(\mathbf{f})}]_{\text{new}} = \mathbf{A}^{-m} - \mathbf{A}^{-m} \left[\frac{\partial \log Z}{\partial \mathbf{b}^{-m}} \left(\frac{\partial \log Z}{\partial \mathbf{b}^{-m}} \right)^\top - 2 \frac{\partial \log Z}{\partial \mathbf{A}^{-m}} \right] \mathbf{A}^{-m}, \quad (45)$$

$$[\mathbf{m}_{\beta_n(\mathbf{f})}]_{\text{new}} = \mathbf{b}^{-m} + \mathbf{A}^{-m} \frac{\partial \log Z}{\partial \mathbf{b}^{-m}}. \quad (46)$$

Next, we divide the Gaussian with mean and covariance parameters given by Eqs. (45) and (46) by the marginal for $\beta(\mathbf{f})$ in the cavity distribution $q^{-m}(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K)$. Therefore, the new parameters $\tilde{\mathbf{A}}_n$ and $\tilde{\mathbf{b}}_n$ of the approximate factor $\Psi(\mathbf{x}_k^j)$ are obtained using the formula for the ratio of two Gaussians:

$$\tilde{\mathbf{A}}_n^{\text{new}} = [\mathbf{V}_{\beta_n(\mathbf{f})}]_{\text{new}}^{-1} - [\mathbf{A}^{-m}]^{-1}, \quad (47)$$

$$\tilde{\mathbf{b}}_n^{\text{new}} = [\mathbf{V}_{\beta_n(\mathbf{f})}]_{\text{new}}^{-1} [\mathbf{m}_{\beta_n(\mathbf{f})}]_{\text{new}} - [\mathbf{A}^{-m}]^{-1} \mathbf{b}^{-m}. \quad (48)$$

A.1.2 ADJUSTING $\tilde{\Gamma}(\mathbf{x}_k^j)$ BY EP

We explain how to adjust the parameters \tilde{g}_k and \tilde{h}_k of the approximate factor $\tilde{\Gamma}(\mathbf{x}_k^j)$ using EP. EP performs this operation by minimizing the following Kullback-Leibler divergence:

$$\text{KL}[\tilde{\Gamma}(\mathbf{x}_k^j) q^{-m}(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K) \| \tilde{\Gamma}(\mathbf{x}_k^j) q^{-m}(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K)], \quad (49)$$

where $q^{-m}(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K)$ is the cavity distribution given by

$$q^{-m}(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K) = q(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K) [\tilde{\Gamma}(\mathbf{x}_k^j)]^{-1}, \quad (50)$$

We integrate out in q^{-m} all the variables except those which $\tilde{\Gamma}(\mathbf{x}_k^j)$ does depend on, namely, $c_1(\mathbf{x}_k^j), \dots, c_K(\mathbf{x}_k^j)$. Then q^{-m} takes the form

$$q^{-m}[c_1(\mathbf{x}_k^j), \dots, c_K(\mathbf{x}_k^j)] \propto \prod_{k=1}^K \mathcal{N}(c_k(\mathbf{x}_k^j) | h_k^-, g_k^-), \quad (51)$$

where the parameters h_k^- and g_k^- of these Gaussian distributions are obtained by using the formula for dividing Gaussians:

$$g_k^- = \left\{ v_{c_k(\mathbf{x}_k^j)}^{-1} - \tilde{g}_k^- \right\}^{-1}, \quad h_k^- = g_k^- \left\{ v_{c_k(\mathbf{x}_k^j)}^{-1} m_{c_k}(\mathbf{x}_k^j) - \tilde{e}_k^- \right\}^{-1}, \quad (52)$$

where $v_{c_k(\mathbf{x}_k^j)}$ is the variance for $c_k(\mathbf{x}_k^j)$ given by $q(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K)$ in Eq. (32) and $m_{c_k}(\mathbf{x}_k^j)$ is the corresponding mean parameter.

To minimize Eq. (49) we match the 1st and 2nd moments of $\tilde{\Gamma}(\mathbf{x}_k^j) q^{-m}(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K)$ and $\tilde{\Gamma}(\mathbf{x}_k^j) q^{-m}(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K)$. The moments of $\tilde{\Gamma}(\mathbf{x}_k^j) q^{-m}(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K)$ can be obtained from the derivatives of the logarithm of its normalization constant Z , which is given by

$$Z = \int \Gamma(\mathbf{x}_k^j) q^{-m}(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K) d\mathbf{f} d\mathbf{c}_1 \dots d\mathbf{c}_K = \prod_{k=1}^K \Phi \left[\frac{c_k}{\sigma_k} \right], \quad (53)$$

where $\alpha_n^k = m_{c_k(\mathbf{x}_k^j)} v_{c_k(\mathbf{x}_k^j)}^{-1/2}$. We follow Eqs. (5.12) and (5.13) in (Minka, 2001b) to update \tilde{g}_k and \tilde{h}_k in Eq. (31). However, we use the second partial derivative with respect to g_k^- rather than first partial derivative with respect to h_k^- for numerical robustness. These derivatives are given by

$$\frac{\partial \log Z}{\partial h_k^-} = \frac{(Z-1)\rho(\alpha_n^k)}{Z\Phi(\alpha_n^k)\sqrt{g_k^-}}, \quad \frac{\partial^2 \log Z}{\partial [h_k^-]^2} = -\frac{\partial \log Z}{\partial h_k^-} \cdot \frac{\alpha_n^k}{\sqrt{g_k^-}} - \left[\frac{\partial \log Z}{\partial h_k^-} \right]^2. \quad (54)$$

The update equations for the parameters \tilde{g}_k and \tilde{h}_k of the approximate factor $\tilde{\Gamma}(\mathbf{x}_k^j)$ are

$$[\tilde{g}_k]_{\text{new}} = - \left\{ \left(\frac{\partial^2 \log Z}{\partial [h_k^-]^2} \right)^{-1} + g_k^- \right\}^{-1}, \quad [\tilde{h}_k]_{\text{new}} = \left\{ g_k^- - \left[\frac{\partial^2 \log Z}{\partial [h_k^-]^2} \right] \frac{\partial \log Z}{\partial h_k^-} \right\} [\tilde{g}_k]_{\text{new}}.$$

A.2 Approximating the Non-Gaussian Factor that Depends on \mathbf{x}

Expectation propagation performs the operations described in Appendices A.1.1 and A.1.2 until the Gaussian approximations to $\Gamma(\mathbf{x}_k^j)$, $\Psi(\mathbf{x}_k^j), \dots, \Psi(\mathbf{x}_f^{N_f})$ converge. Importantly the EP operations described in Appendices A.1.1 and A.1.2 can be implemented independently of the value of \mathbf{x} , that is, the location at which we will be evaluating PESC's acquisition function. After EP has converged, the next step is to approximate with Gaussians the other factor in Eq. (26) that does depend on \mathbf{x} , that is, $\Psi(\mathbf{x})$. For this, we first replace the exact factors $\Gamma(\mathbf{x}_k^j)$, $\Psi(\mathbf{x}_k^j), \dots, \Psi(\mathbf{x}_f^{N_f})$ in Eq. (26) with their Gaussian approximations. This results in the following approximation:

$$f(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K | \mathbf{x}_k^j) \approx \tilde{f}(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K | \mathbf{x}_k^j) = q(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K) \Psi(\mathbf{x}), \quad (55)$$

where $q(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K)$, as given by Eq. (32), approximates the product of $p(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K | \mathcal{D})$ and $\Gamma(\mathbf{x}_k^j)$, $\Psi(\mathbf{x}_k^j), \dots, \Psi(\mathbf{x}_f^{N_f})$ in Eq. (26). Next, we find a Gaussian approximation to the right-hand-side of Eq. (55). For this, we first marginalize out in q all the variables except those which $\Psi(\mathbf{x})$ does depend on, that is, $\gamma(\mathbf{f})$ and $c_1(\mathbf{x}), \dots, c_K(\mathbf{x})$, where $\gamma(\mathbf{f}) = [f(\mathbf{x}), f(\mathbf{x}_k^j)]^\top$, we obtain

$$q[\gamma(\mathbf{f}), c_1(\mathbf{x}), \dots, c_K(\mathbf{x})] = \mathcal{N}(\gamma(\mathbf{f}) | \mathbf{m}_{\gamma(\mathbf{f})}, \mathbf{V}_{\gamma(\mathbf{f})}) \left\{ \prod_{k=1}^K \mathcal{N}(c_k(\mathbf{x}) | m_{c_k(\mathbf{x})}, v_{c_k(\mathbf{x})}) \right\}, \quad (56)$$

where $\mathbf{V}_{\gamma(\mathbf{f})}$ is the 2×2 covariance matrix for $\gamma(\mathbf{f})$ given by $q(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K)$ in Eq. (32), $\mathbf{m}_{\gamma(\mathbf{f})}$ is the corresponding 2-dimensional mean vector, $v_{c_k(\mathbf{x})}$ is the variance for $c_k(\mathbf{x})$ given by $q(\mathbf{f}, \mathbf{c}_1, \dots, \mathbf{c}_K)$ in Eq. (32) and $m_{c_k(\mathbf{x})}$ is the corresponding mean parameter.

Let \mathbf{m}_i^j and \mathbf{V}_i^j be the mean vector and covariance matrices for the first $N_1 + 1$ elements of \mathbf{f} in Eq. (24), according to q in Eq. (32). Similarly, \mathbf{m}_{k+1}^j and \mathbf{V}_{k+1}^j be the mean vector and covariance matrices for the first $N_1 + 1$ elements of \mathbf{c}_k in Eq. (25), according to q in Eq. (32), for $k = 1, \dots, K$. After the execution of EP, we compute and store \mathbf{m}_i^j and \mathbf{V}_i^j , for $i = 1, \dots, K$. These parameters can then be used to efficiently compute $\mathbf{V}_{\gamma(\mathbf{f})}$, $\mathbf{m}_{\gamma(\mathbf{f})}$, $v_{c_1(\mathbf{x})}, \dots, v_{c_K(\mathbf{x})}$ and $m_{c_1(\mathbf{x})}, \dots, m_{c_K(\mathbf{x})}$ for any arbitrary value of \mathbf{x} . For this, we use Eqs.

(3.22) and (3.24) in (Rasmussen and Williams, 2006) to obtain

$$\begin{aligned}
 [\mathbf{m}_{\gamma(\mathbf{f})}]_1 &= \mathbf{k}^1(\mathbf{x})^\top [\mathbf{K}_{k_*}^1]^{-1} \mathbf{m}'_1, \\
 [\mathbf{m}_{\gamma(\mathbf{f})}]_2 &= [\mathbf{m}'_1]_{N_1+1}, \\
 [\mathbf{V}_{\gamma(\mathbf{f})}]_{1,1} &= \mathbf{k}^1(\mathbf{x}, \mathbf{x}) - \mathbf{k}^1(\mathbf{x})^\top \left\{ [\mathbf{K}_{k_*}^1]^{-1} + [\mathbf{K}_{k_*}^1]^{-1} \mathbf{V}_1 [\mathbf{K}_{k_*}^1]^{-1} \right\} \mathbf{k}^1(\mathbf{x}), \\
 [\mathbf{V}_{\gamma(\mathbf{f})}]_{2,2} &= [\mathbf{V}'_1]_{N_1+1, N_1+1}, \\
 [\mathbf{V}_{\gamma(\mathbf{f})}]_{1,2} &= \mathbf{k}^1(\mathbf{x}, \mathbf{x}_i^j) - \mathbf{k}^1(\mathbf{x})^\top \left\{ [\mathbf{K}_{k_*}^1]^{-1} + [\mathbf{K}_{k_*}^1]^{-1} \mathbf{V}_1 [\mathbf{K}_{k_*}^1]^{-1} \right\} \mathbf{k}^1(\mathbf{x}_i^j), \\
 m_{\alpha_k(\mathbf{x})} &= \mathbf{k}^{k+1}(\mathbf{x})^\top [\mathbf{K}_{k_*}^{k+1}]^{-1} \mathbf{m}'_{k+1}, \\
 v_{\alpha_k(\mathbf{x})} &= \mathbf{k}^{k+1}(\mathbf{x}, \mathbf{x}) - \mathbf{k}^{k+1}(\mathbf{x})^\top \left\{ [\mathbf{K}_{k_*}^{k+1}]^{-1} + [\mathbf{K}_{k_*}^{k+1}]^{-1} \mathbf{V}'_{k+1} [\mathbf{K}_{k_*}^{k+1}]^{-1} \right\} \mathbf{k}^{k+1}(\mathbf{x}),
 \end{aligned}$$

for $k = 1, \dots, K$, where $\mathbf{k}^k(\mathbf{x})$ is the $(N_1 + 1)$ -dimensional vector with the prior cross-covariances between the value of the i -th function in $\{f, c_1, \dots, c_K\}$ at \mathbf{x} and the values of that function at $\mathbf{x}_f^1, \dots, \mathbf{x}_f^{N_1}, \mathbf{x}_x^1, \dots, \mathbf{x}_x^{N_1}$. $\mathbf{K}_{k_*}^k$ is an $(N_1 + 1) \times (N_1 + 1)$ matrix with the prior covariances between the values of that function at $\mathbf{x}_f^1, \dots, \mathbf{x}_f^{N_1}, \mathbf{x}_x^1, \dots, \mathbf{x}_x^{N_1}$ and $\mathbf{k}^k(\mathbf{x}, \mathbf{x})$ contains the prior variance of the values of that function at \mathbf{x} , for $i = 1, \dots, K + 1$. Finally, $\mathbf{k}^1(\mathbf{x}, \mathbf{x}_i^j)$ contains the prior covariance between $f(\mathbf{x})$ and $f(\mathbf{x}_i^j)$.

Once we have computed the parameters of $q[\gamma(\mathbf{f})|c_1(\mathbf{x}), \dots, c_K(\mathbf{x})]$ in Eq. (56) using the formulas above, we obtain the marginal means and variances for $f(\mathbf{x}), c_1(\mathbf{x}), \dots, c_K(\mathbf{x})$ with respect to $q[\gamma(\mathbf{f}), c_1(\mathbf{x}), \dots, c_K(\mathbf{x})|\Psi(\mathbf{x})]$. Let $m_1(\mathbf{x}), \dots, m_{K+1}(\mathbf{x})$ and $v_1(\mathbf{x}), \dots, v_{K+1}(\mathbf{x})$ be these marginal means and variances. Then, we have the approximation

$$\int q[\gamma(\mathbf{f}), c_1(\mathbf{x}), \dots, c_K(\mathbf{x})|\Psi(\mathbf{x})d\mathbf{f}d\mathbf{x}_i^j] \approx \mathcal{N}(f(\mathbf{x})|m_1(\mathbf{x}), v_1(\mathbf{x})) \prod_{k=1}^K \mathcal{N}(c_k(\mathbf{x})|m_{k+1}(\mathbf{x}), v_{k+1}(\mathbf{x})),$$

where $m_1(\mathbf{x}), \dots, m_{K+1}(\mathbf{x})$ and $v_1(\mathbf{x}), \dots, v_{K+1}(\mathbf{x})$ can be obtained from the normalization constant of $q[\gamma(\mathbf{f}), c_1(\mathbf{x}), \dots, c_K(\mathbf{x})|\Psi(\mathbf{x})]$ using Eqs. (5.12) and (5.13) in (Minka, 2001b). This normalization constant is given by

$$Z = \int q[\gamma(\mathbf{f}), c_1(\mathbf{x}), \dots, c_K(\mathbf{x})|\Psi(\mathbf{x})d\gamma(\mathbf{f})dc_1(\mathbf{x})dc_K(\mathbf{x})] = \Phi(\alpha) \prod_{k=1}^K \Phi(\alpha_k) + 1 - \prod_{k=1}^K \Phi(\alpha_k),$$

where

$$\alpha_k = \frac{m_{\alpha_k(\mathbf{x})}}{\sqrt{v_{\alpha_k(\mathbf{x})}}}, \quad \alpha = \frac{[1, -1]\mathbf{m}_{\gamma(\mathbf{f})}}{\sqrt{s}}, \quad s = [\mathbf{V}_{\gamma(\mathbf{f})}]_{1,1} + [\mathbf{V}_{\gamma(\mathbf{f})}]_{2,2} - 2[\mathbf{V}_{\gamma(\mathbf{f})}]_{1,2}. \quad (57)$$

Given Z , we then compute $m_1(\mathbf{x}), \dots, m_{K+1}(\mathbf{x})$ and $v_1(\mathbf{x}), \dots, v_{K+1}(\mathbf{x})$ using Eqs. (5.12) and (5.13) in (Minka, 2001b):

$$v_1(\mathbf{x}) = [\mathbf{V}_{\gamma(\mathbf{f})}]_{1,1} - \frac{\beta}{s} (\beta + \alpha) \left\{ [\mathbf{V}_{\gamma(\mathbf{f})}]_{1,1} - [\mathbf{V}_{\gamma(\mathbf{f})}]_{1,2} \right\}^2, \quad (58)$$

$$m_1(\mathbf{x}) = [\mathbf{m}_{\gamma(\mathbf{f})}]_1 + \left\{ [\mathbf{V}_{\gamma(\mathbf{f})}]_{1,1} - [\mathbf{V}_{\gamma(\mathbf{f})}]_{1,2} \right\} \frac{\beta}{\sqrt{s}}, \quad (59)$$

$$v_{k+1}(x) = \left\{ v_{\alpha_k(\mathbf{x})}^{-1} + \tilde{\alpha}_k \right\}^{-1}, \quad \text{for } k = 1, \dots, K, \quad (60)$$

$$m_{k+1}(x) = v_{k+1}(x) \left\{ m_{\alpha_k(\mathbf{x})} v_{\alpha_k(\mathbf{x})}^{-1} + \tilde{b}_k \right\}, \quad \text{for } k = 1, \dots, K, \quad (61)$$

where

$$\beta = Z^{-1} \phi(\alpha) \prod_{k=1}^K \Phi(\alpha_k), \quad \tilde{\alpha}_k = - \left\{ \frac{\partial^2 \log Z}{\partial m_{\alpha_k(\mathbf{x})}^2} + v_{\alpha_k(\mathbf{x})} \right\}^{-1}, \quad (62)$$

$$\tilde{b}_k = \tilde{\alpha}_k \left\{ m_{\alpha_k(\mathbf{x})} + \frac{\sqrt{v_{\alpha_k(\mathbf{x})}}}{\alpha_k + \beta_k} \right\}, \quad \frac{\partial^2 \log Z}{\partial m_{\alpha_k(\mathbf{x})}^2} = - \frac{\beta_k \{ \alpha_k + \beta_k \}}{v_{\alpha_k(\mathbf{x})}}, \quad (63)$$

$$\beta_k = \frac{\phi(\alpha_{n_i})}{Z\Phi(\alpha_{n_i})} (Z - 1). \quad (64)$$

Eqs. (58) to (61) are the output of our EP algorithm. These quantities are used in Eq. (20) to evaluate PESOC's acquisition function.

Appendix B. Implementation Considerations

We give details on the practical implementation of PESOC.

B.1 Initialization, Convergence of EP and Parallel EP Updates

We start by fixing the parameters of all the approximate factors $\tilde{\Gamma}(\mathbf{x}_i^j), \tilde{\Psi}(\mathbf{x}_f^1), \dots, \tilde{\Psi}(\mathbf{x}_f^{N_1})$ to be zero. We stop EP when the absolute change in the means and covariance matrices for the first $N_1 + 1$ elements of \mathbf{f} and c_1, \dots, c_K in Eqs. (24) and (25), according to q in Eq. (32), is below 10^{-4} . The approximate factors $\tilde{\Gamma}(\mathbf{x}_i^j), \tilde{\Psi}(\mathbf{x}_f^1), \dots, \tilde{\Psi}(\mathbf{x}_f^{N_1})$ are updated in parallel to speed up convergence (Gerwen et al., 2009). With parallel updates q in Eq. (20) is only updated once per iteration, after all the approximate factors have been refined.

B.2 EP with Damping

To improve the convergence of EP, we use damping (Minka and Laferrière, 2002). If $\tilde{\Psi}(\mathbf{x}_f^j)^{\text{new}}$ is the value of an approximate factor that minimizes the KL-divergence, damping entails using instead $\tilde{\Psi}(\mathbf{x}_f^j)^{\text{damped}}$ as the new factor value, as defined below:

$$\tilde{\Psi}(\mathbf{x}_f^j)^{\text{damped}} = [\tilde{\Psi}(\mathbf{x}_f^j)^{\text{new}}]^\epsilon + [\tilde{\Psi}(\mathbf{x}_f^j)^{\text{old}}]^{1-\epsilon}, \quad (65)$$

where $\tilde{\Psi}(\mathbf{x}_f^j)^{\text{old}}$ is the factor value before performing the update. We do the same for $\tilde{\Gamma}(\mathbf{x}_i^j)$. The parameter ϵ controls the amount of damping, with $\epsilon = 1$ corresponding to no damping. We initialize ϵ to 1 and multiply it by a factor of 0.99 at each iteration.

During the execution of EP, some covariance matrices in q or in the cavity distributions may become non-positive-definite due to an excessively large step size (i.e. large ϵ). If this issue is encountered during an EP iteration, the damping parameter is reduced by half and the iteration is repeated.

B.3 Sampling \mathbf{x}_* in PESC

We sample \mathbf{x}_* from its posterior distribution using an extension of the method described by Hernández-Lobato et al. (2014) to sample \mathbf{x}_* in the unconstrained setting. We perform a finite basis approximation to the GPs used to describe the data for the objective and the constraints. This allows us to sample analytic approximate samples from the GP posterior distribution. We then solve the optimization problem given by Eq. (1), when the functions f, c_1, \dots, c_K are replaced by the generated samples. For this, we use a numerical method for solving constrained optimization problems: the Method of Moving Asymptotes (MMA) (Svanberg, 2002) as implemented in the NLOpt package (Johnson, 2014). We evaluate the sampled functions in a uniform grid of size 10^3 and obtain the best feasible result in that grid. We add to the points in the uniform grid the evaluation locations for which we have already collected data. This is then used as the initial point for the MMA method. The number of basis functions in the approximation to the GP is 10^3 . The NLOpt convergence tolerance is 10^{-6} in the scaled input space units.

The finite basis approximation to the GP is given by a Bayesian Gaussian linear model build on top of a collection of basis functions (Hernández-Lobato et al., 2014). Drawing an approximate sample from the GP posterior distribution involves then sampling from the posterior distribution of that linear model given the observed data. When the number of basis functions is larger than the observed data points, this can be done efficiently as described by Hernández-Lobato et al. (2014). In this case, the covariance matrix of the Gaussian posterior distribution for the linear model is the sum of a low rank matrix and a diagonal matrix, we can then use an efficient method to sample from that Gaussian posterior distribution. This method is outlined in Appendix B.2 of Seeger (2008). The cost is $\mathcal{O}(N^2M)$ where N is the number of collected data points and M is the number of basis functions. Sampling with the naive method takes $\mathcal{O}(M^3)$ operations because we must take the Cholesky decomposition of an $M \times M$ covariance matrix. Given that in our implementation $M = 10^3$ and typically $N < 100$, this method can speed up this sampling procedure by orders of magnitude. A more efficient implementation could also be obtained by using quasi-random numbers to generate the basis functions, thus reducing the number of basis functions needed to attain the same approximation quality (Yang et al., 2014).

B.4 Cholesky Update in PESC-F

In PESC-F, during the fast BO computations, the GP hyperparameters (and in particular the length scales) are not changed from the ones used during last iteration. Because of this, the GP kernel matrix is unchanged except for the addition of a new row and column. Given this, we can compute the Cholesky decomposition of the new kernel matrix with a rank-one update of the Cholesky decomposition of the current kernel matrix. The $\mathcal{O}(N^3)$ computation of the Cholesky decomposition of the kernel matrix is the main bottleneck for GP-based Bayesian optimization. As N gets large, this trick can significantly speed

up the fast BO computations in PESC-F. In fact, this trick applies more generally beyond PESC-F or even any fast-update method: any Bayesian optimization method that does not update the GP hyperparameters at every iteration can take advantage of the rank-one Cholesky update. This update technique is described in more detail by Gill et al. (1974) and is commonly used in the setting of Bayesian optimization as seen in (Osborne, 2010).

References

- Eric Brochu, Vlad M. Cora, and Nando de Freitas. A tutorial on Bayesian optimization of expensive cost functions, 2010. arXiv:1012.2599 [cs.LG].
- Simon Duane, Anthony D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics letters B*, 195(2):216–222, 1987.
- Andrew Frank and Arthur Asuncion. UCI machine learning repository, 2010.
- Jacob R. Gardner, Matt J. Kusner, Zhixiang Eddie Xu, Kilian Q. Weinberger, and John P. Cunningham. Bayesian optimization with inequality constraints. In *Proceedings of the 31th International Conference on Machine Learning (ICML)*, pages 937–945, 2014.
- Michael A. Gelbart, Jasper Snoek, and Ryan P. Adams. Bayesian optimization with unknown constraints. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, (UAI)*, pages 250–259, 2014.
- Andrew Gelman and Donald R. Rubin. A single series from the Gibbs sampler provides a false sense of security. In *Bayesian Statistics 4: Proceedings of the Fourth Valencia International Meeting*, pages 625–32, 1992.
- Marcel V. Gerven, Botond Cséke, Robert Oostenveld, and Tom Heskes. Bayesian source localization with the multivariate Laplace prior. In *Advances in Neural Information Processing Systems 22 (NIPS)*, pages 1901–1909, 2009.
- John Geweke. Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. In *Bayesian Statistics 4: Proceedings of the Fourth Valencia International Meeting*, pages 169–193, 1992.
- Philip E. Gill, Gene H. Golub, Walter Murray, and Michael A. Saunders. Methods for modifying matrix factorizations. *Mathematics of Computation*, 28(126):505–535, 1974.
- David Ginsbourger, Janis Janusevskis, and Rodolphe Le Riche. Dealing with asynchronicity in parallel Gaussian process based global optimization. *hal-00507632*, pages 1–27, 2011. URL <https://hal.archives-ouvertes.fr/hal-00507632>.
- Robert B. Gramacy and Herbert K. H. Lee. Optimization under unknown constraints. In *Bayesian Statistics 9: Proceedings of the Ninth Valencia International Meeting*, pages 229–256, 2011.
- Robert B. Gramacy, Genetha A. Gray, Sébastien Le Digabel, Herbert K. H. Lee, Pritam Ranjan, Garth Wells, and Stefan M. Wild. Modeling an augmented Lagrangian for blackbox constrained optimization. *Technometrics*, 58(1):1–11, 2016.

- Nikolaus Hansen and Andreas Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 312–317, 1996.
- Nicholas Hay, Stuart J. Russell, David Tolpin, and Solomon Eyal Shimony. Selecting computations: Theory and applications. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, (UAI), pages 346–355, 2012.
- Philipp Hennig and Christian J. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(1):1809–1837, 2012.
- Daniel Hernández-Lobato, José Miguel Hernández-Lobato, Amar Shah, and Ryan P. Adams. Predictive entropy search for multi-objective Bayesian optimization. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 1492–1501, 2016.
- José Miguel Hernández-Lobato, Matthew W. Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pages 918–926, 2014.
- José Miguel Hernández-Lobato, Michael A. Gelbart, Matthew W. Hoffman, Ryan P. Adams, and Zoubin Ghahramani. Predictive entropy search for Bayesian optimization with unknown constraints. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 1699–1707, 2015.
- Neil Houlsby, José Miguel Hernández-Lobato, Ferenc Huszar, and Zoubin Ghahramani. Collaborative Gaussian processes for preference learning. In *Advances in Neural Information Processing Systems 25 (NIPS)*, pages 2096–2104, 2012.
- Steven G. Johnson. The NLOPT nonlinear-optimization package, 2014. URL <http://ab-initio.mit.edu/nlopt>.
- Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.
- Joshua Knowles. Parego: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006.
- Günther Leobacher and Friedrich Pillichshammer. *Introduction to quasi-Monte Carlo integration and applications*. Springer, 2014.
- David J. C. MackKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.
- Thomas P. Minka. Expectation propagation for approximate Bayesian inference. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence (UAI)*, pages 362–369, 2001a.
- Thomas P. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001b.
- Thomas P. Minka and John Lafferty. Expectation-propagation for the generative aspect model. In *Proceedings of the 18th Conference in Uncertainty in Artificial Intelligence (UAI)*, pages 352–359, 2002.
- Jonas Mockus, Vytautas Tiesis, and Antanas Zilinskas. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2(117-129):2, 1978.
- Radford M. Neal. MCMC using Hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo*, Chapman & Hall/CRC Handbooks of Modern Statistical Methods. CRC Press, 2011.
- Michael Osborne. *Bayesian Gaussian processes for sequential prediction, optimisation and quadrature*. PhD thesis, University of Oxford, 2010.
- James Parr. *Improvement criteria for constraint handling and multiobjective optimization*. PhD thesis, University of Southampton, 2013.
- Anand Patil, David Huard, and Christopher Fonnnesbeck. PyMC: Bayesian stochastic modeling in Python. *Journal of Statistical Software*, 35(4):1–81, 2010.
- Victor Picheny. A stepwise uncertainty reduction approach to constrained global optimization. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 787–795, 2014.
- Martyn Plummer, Nicky Best, Kate Cowles, and Karen Vines. CODA: Convergence diagnosis and output analysis for MCMC. *R News*, 6(1):7–11, 2006.
- Carl Rasmussen and Christopher Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- Stuart Russell. Principles of metareasoning. *Artificial Intelligence*, 49(1-3):361–395, 1991.
- Dan Russo and Benjamin Van Roy. Learning to optimize via information-directed sampling. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pages 1583–1591, 2014.
- Matthias Schonlau, William J. Welch, and Donald R. Jones. Global versus local search in constrained optimization of computer models. In Nancy Flournoy, William F. Rosenberger, and Weng Kee Wong, editors, *New developments and applications in experimental design*, volume 34 of *Lecture Notes–Monograph Series*, pages 11–25. Institute of Mathematical Statistics, 1998.
- Matthias W. Seeger. Bayesian inference and optimal design for the sparse linear model. *Journal of Machine Learning Research*, 9:759–813, 2008.
- Jasper Snoek. *Bayesian optimization and semiparametric models with applications to assistive technology*. PhD thesis, University of Toronto, 2013.
- Jasper Snoek, Higo Larochelle, and Ryan P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25 (NIPS)*, pages 2951–2959, 2012.

- Krister Svaberg. A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM Journal on Optimization*, 12:555–573, 2002.
- Kevin Swersky, Jasper Snoek, and Ryan P. Adams. Multi-task Bayesian optimization. In *Advances in Neural Information Processing Systems 26 (NIPS)*, pages 2004–2012, 2013.
- Julien Villemonteix, Emmanuel Vazquez, and Eric Walter. An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization*, 44(4):509–534, 2009.
- Jiyan Yang, Vikas Sindhwani, Haim Avron, and Michael W. Mahoney. Quasi-Monte Carlo feature maps for shift-invariant kernels. In *Proceedings of the 31th International Conference on Machine Learning (ICML)*, pages 485–493, 2014.

Optimal Estimation and Completion of Matrices with Biclustering Structures

Chao Gao

Yu Lu

Yale University

CHAO.GAO@YALE.EDU

YU.LU@YALE.EDU

Zongming Ma

University of Pennsylvania

ZONGMING@WHARTON.UPENN.EDU

Harrison H. Zhou

Yale University

HUIBIN.ZHOU@YALE.EDU

Editor: Edo Airoldi

Abstract

Biclustering structures in data matrices were first formalized in a seminal paper by John Hartigan (Hartigan, 1972) where one seeks to cluster cases and variables simultaneously. Such structures are also prevalent in block modeling of networks. In this paper, we develop a theory for the estimation and completion of matrices with biclustering structures, where the data is a partially observed and noise contaminated matrix with a certain underlying biclustering structure. In particular, we show that a constrained least squares estimator achieves minimax rate-optimal performance in several of the most important scenarios. To this end, we derive unified high probability upper bounds for all sub-Gaussian data and also provide matching minimax lower bounds in both Gaussian and binary cases. Due to the close connection of graphon to stochastic block models, an immediate consequence of our general results is a minimax rate-optimal estimator for sparse graphons.

Keywords: Biclustering, graphon, matrix completion, missing data, stochastic block models, sparse network

1. Introduction

In a range of important data analytic scenarios, we encounter matrices with biclustering structures. For instance, in gene expression studies, one can organize the rows of a data matrix to correspond to individual cancer patients and the columns to transcripts. Then the patients are expected to form groups according to different cancer subtypes and the genes are also expected to exhibit clustering effect according to the different pathways they belong to. Therefore, after appropriate reordering of the rows and the columns, the data matrix is expected to have a biclustering structure contaminated by noises (Lee et al., 2010). Here, the observed gene expression levels are real numbers. In a different context, such a biclustering structure can also be present in network data. For example, stochastic block model (SBM for short) (Holland et al., 1983) is a popular model for exchangeable networks. In SBMs, the graph nodes are partitioned into k disjoint communities and the probability that any pair of nodes are connected is determined entirely by the community memberships of the nodes. Consequently, if one rearranges the nodes from the same communities together in

the graph adjacency matrix, then the mean adjacency matrix, where each off-diagonal entry equals the probability of an edge connecting the nodes represented by the corresponding row and column, also has a biclustering structure.

The goal of the present paper is to develop a theory for the estimation (and completion when there are missing entries) of matrices with biclustering structures. To this end, we propose to consider the following general model

$$X_{ij} = \theta_{ij} + \epsilon_{ij}, \quad i \in [n_1], j \in [n_2], \quad (1)$$

where for any positive integer m , we let $[m] = \{1, \dots, m\}$. Here, for each (i, j) , $\theta_{ij} = \mathbb{E}[X_{ij}]$ and ϵ_{ij} is an independent piece of mean zero sub-Gaussian noise. Moreover, we allow entries to be missing completely at random (Rubin, 1976). Thus, let E_{ij} be i.i.d. Bernoulli random variables with success probability $p \in (0, 1]$ indicating whether the (i, j) th entry is observed, and define the set of observed entries

$$\Omega = \{(i, j) : E_{ij} = 1\}. \quad (2)$$

Our final observations are

$$X_{ij}, \quad (i, j) \in \Omega. \quad (3)$$

To model the biclustering structure, we focus on the case where there are k_1 row clusters and k_2 column clusters, and the values of $\{\theta_{ij}\}$ are taken as constant if the rows and the columns belong to the same clusters. The goal is then to recover the signal matrix $\theta \in \mathbb{R}^{n_1 \times n_2}$ from the observations (3). To accommodate most interesting cases, especially the case of undirected networks, we shall also consider the case where the data matrix X is symmetric with zero diagonals. In such cases, we also require $X_{ij} = X_{ji}$ and $E_{ij} = E_{ji}$ for all $i \neq j$.

Main contributions In this paper, we propose a unified estimation procedure for partially observed data matrix generated from model (1) – (3). We establish high probability upper bounds for the mean squared errors of the resulting estimators. In addition, we show that these upper bounds are minimax rate-optimal in both the continuous case and the binary case by providing matching minimax lower bounds. Furthermore, SBM can be viewed as a special case of the symmetric version of (1). Thus, an immediate application of our results is the network completion problem for SBMs. With partially observed network edges, our method gives a rate-optimal estimator for the probability matrix of the whole network in both the dense and the sparse regimes, which further leads to rate-optimal graphon estimation in both regimes.

Connection to the literature If only a low rank constraint is imposed on the mean matrix θ , then (1) – (3) becomes what is known in the literature as the matrix completion problem (Recht et al., 2010). An impressive list of algorithms and theories have been developed for this problem, including but not limited to Candès and Recht (2009); Keshavan et al. (2009); Candès and Tao (2010); Candès and Plan (2010); Cai et al. (2010); Keshavan et al. (2010); Recht (2011); Koltchinskii et al. (2011). In this paper, we investigate an alternative biclustering structural assumption for the matrix completion problem, which was first proposed by John Hartigan (Hartigan, 1972). Note that a biclustering structure

automatically implies low-rankness. However, if one applies a low rank matrix completion algorithm directly in the current setting, the resulting estimator suffers an inferior error bound to the minimax rate-optimal one. Thus, a full exploitation of the biclustering structure is necessary, which is the focus of the current paper.

The results of our paper also imply rate-optimal estimation for sparse graphons. Previous results on graphon estimation include Airolidi et al. (2013); Wolfe and Olhede (2013); Olhede and Wolfe (2014); Borgs et al. (2015); Choi (2015) and the references therein. The minimax rates for dense graphon estimation were derived by Gao et al. (2015a). During the time when this paper was written, we became aware of an independent result on optimal sparse graphon estimation by Klopp et al. (2015).

There are also an interesting line of works on biclustering (Flynn and Perry, 2012; Role et al., 2012; Choi and Wolfe, 2014). While these papers aim to recover the clustering structures of rows and columns, the goal of the current paper is to estimate the underlying mean matrix with optimal rates.

Organization After a brief introduction to notation, the rest of the paper is organized as follows. In Section 2, we introduce the precise formulation of the problem and propose a constrained least squares estimator for the mean matrix θ . In Section 3, we show that the proposed estimator leads to minimax optimal performance for both Gaussian and binary data. Section 4 presents some extensions of our results to sparse graphon estimation and adaptation. Implementation and simulation results are given in Section 5. In Section 6, we discuss the key points of the paper and propose some open problems for future research. The proofs of the main results are laid out in Section 7, with some auxiliary results deferred to the appendix.

Notation For a vector $z \in [k]^n$, define the set $z^{-1}(a) = \{i \in [n] : z(i) = a\}$ for $a \in [k]$. For a set S , $|S|$ denotes its cardinality and $\mathbf{1}_S$ denotes the indicator function. For a matrix $A = (A_{ij}) \in \mathbb{R}^{n_1 \times n_2}$, the ℓ_2 norm and ℓ_∞ norm are defined by $\|A\| = \sqrt{\sum_{ij} A_{ij}^2}$ and $\|A\|_\infty = \max_{ij} |A_{ij}|$, respectively. The inner product for two matrices A and B is $\langle A, B \rangle = \sum_{ij} A_{ij} B_{ij}$. Given a subset $\Omega \in [n_1] \times [n_2]$, we use the notation $\langle A, B \rangle_\Omega = \sum_{(i,j) \in \Omega} A_{ij} B_{ij}$ and $\|A\|_\Omega = \sqrt{\sum_{(i,j) \in \Omega} A_{ij}^2}$. Given two numbers $a, b \in \mathbb{R}$, we use $a \vee b = \max(a, b)$ and $a \wedge b = \min(a, b)$. The floor function $\lfloor a \rfloor$ is the largest integer no greater than a , and the ceiling function $\lceil a \rceil$ is the smallest integer no less than a . For two positive sequences $\{a_n\}, \{b_n\}$, $a_n \lesssim b_n$ means $a_n \leq C b_n$ for some constant $C > 0$ independent of n , and $a_n \gtrsim b_n$ means $a_n \gtrsim b_n$ and $b_n \lesssim a_n$. The symbols \mathbb{P} and \mathbb{E} denote generic probability and expectation operators whose distribution is determined from the context.

2. Constrained least squares estimation

Recall the generative model defined in (1) and also the definition of the set Ω in (2) of the observed entries. As we have mentioned, throughout the paper, we assume that the ϵ_{ij} 's are independent sub-Gaussian noises with sub-Gaussianity parameter uniformly bounded from above by $\sigma > 0$. More precisely, we assume

$$\mathbb{E} e^{\lambda \epsilon_{ij}} \leq e^{\lambda^2 \sigma^2 / 2}, \quad \text{for all } i \in [n_1], j \in [n_2] \text{ and } \lambda \in \mathbb{R}. \quad (4)$$

We consider two types of biclustering structures. One is rectangular and asymmetric, where we assume that the mean matrix belongs to the following parameter space

$$\Theta_{k_1 k_2}(M) = \left\{ \theta = (\theta_{ij}) \in \mathbb{R}^{n_1 \times n_2} : \theta_{ij} = Q_{z_1(i)z_2(j)}, z_1 \in [k_1]^{n_1}, z_2 \in [k_2]^{n_2}, \right. \\ \left. Q \in [-M, M]^{k_1 \times k_2} \right\}. \quad (5)$$

In other words, the mean values within each bicluster is homogenous, i.e., $\theta_{ij} = Q_{ab}$ if the i th row belongs to the a th row cluster and the j th column belong to the b th column cluster. The other type of structures we consider is the square and symmetric case. In this case, we impose symmetry requirement on the data generating process, i.e., $n_1 = n_2 = n$ and

$$X_{ij} = X_{ji}, E_{ij} = E_{ji}, \text{ for all } i \neq j. \quad (6)$$

Since the case is mainly motivated by undirected network data where there is no edge linking any node to itself, we also assume $X_{ii} = 0$ for all $i \in [n]$. Finally, the mean matrix is assumed to belong to the following parameter space

$$\Theta_k^s(M) = \left\{ \theta = (\theta_{ij}) \in \mathbb{R}^{n \times n} : \theta_{ii} = 0, \theta_{ij} = \theta_{ji} = Q_{z(i)z(j)} \text{ for } i > j, z \in [k]^n, \right. \\ \left. Q = Q^T \in [-M, M]^{k \times k} \right\}. \quad (7)$$

We proceed by assuming that we know the parameter space Θ which can be either $\Theta_{k_1 k_2}(M)$ or $\Theta_k^s(M)$ and the rate p of an independent entry being observed. The issues of adaptation to unknown numbers of clusters and unknown observation rate p are addressed later in Section 4.1 and Section 4.2. Given Θ and p , we propose to estimate θ by the following program

$$\min_{\theta \in \Theta} \left\{ \|\theta\|^2 - \frac{2}{p} \langle X, \theta \rangle_\Omega \right\}. \quad (8)$$

If we define

$$Y_{ij} = X_{ij} E_{ij} / p, \quad (9)$$

then (8) is equivalent to the following constrained least squares problem

$$\min_{\theta \in \Theta} \|Y - \theta\|^2, \quad (10)$$

and hence the name of our estimator. When the data is binary, $\Theta = \Theta_k^s(1)$ and $p = 1$, the problem specializes to estimating the mean adjacency matrix in stochastic block models, and the estimator defined as the solution to (10) reduces to the least squares estimator in Gao et al. (2015a).

3. Main results

In this section, we provide theoretical justifications of the constrained least squares estimator defined as the solution to (10). Our first result is the following universal high probability upper bounds.

Theorem 1. For any global optimizer of (10) and any constant $C' > 0$, there exists a constant $C > 0$ only depending on C' such that

$$\|\hat{\theta} - \theta\|^2 \leq C \frac{M^2 \vee \sigma^2}{p} (k_1 k_2 + n_1 \log k_1 + n_2 \log k_2),$$

with probability at least $1 - \exp(-C'(k_1 k_2 + n_1 \log k_1 + n_2 \log k_2))$ uniformly over $\theta \in \Theta_{k_1, k_2}(M)$ and all error distributions satisfying (4). For the symmetric parameter space $\Theta_k^s(M)$, the bound is simplified to

$$\|\hat{\theta} - \theta\|^2 \leq C \frac{M^2 \vee \sigma^2}{p} (k^2 + n \log k),$$

with probability at least $1 - \exp(-C'(k^2 + n \log k))$ uniformly over $\theta \in \Theta_k^s(M)$ and all error distributions satisfying (4).

When $(M^2 \vee \sigma^2)$ is bounded, the rate in Theorem 1 is $(k_1 k_2 + n_1 \log k_1 + n_2 \log k_2)/p$ which can be decomposed into two parts. The part involving $k_1 k_2$ reflects the number of parameters in the biclustering structure, while the part involving $(n_1 \log k_1 + n_2 \log k_2)$ results from the complexity of estimating the clustering structures of rows and columns. It is the price one needs to pay for not knowing the clustering information. In contrast, the minimax rate for matrix completion under low rank assumption would be $(n_1 \vee n_2)(k_1 \wedge k_2)/p$ (Koltchinskii et al., 2011; Ma and Wu, 2015), since without any other constraint the biclustering assumption implies that the rank of the mean matrix θ is at most $k_1 \wedge k_2$. Therefore, we have $(k_1 k_2 + n_1 \log k_1 + n_2 \log k_2)/p \ll (n_1 \vee n_2)(k_1 \wedge k_2)/p$ as long as both $n_1 \vee n_2$ and $k_1 \wedge k_2$ tend to infinity. Thus, by fully exploiting the biclustering structure, we obtain a better convergence rate than only using the low rank assumption.

In the rest of this section, we discuss two most representative cases, namely the Gaussian case and the symmetric Bernoulli case. The latter case is also known in the literature as stochastic block models.

The Gaussian case Specializing Theorem 1 to Gaussian random variables, we obtain the following result.

Corollary 2. Assume $\epsilon_{ij} \stackrel{iid}{\sim} N(0, \sigma^2)$ and $M \leq C_1 \sigma$ for some constant $C_1 > 0$. For any constant $C' > 0$, there exists some constant C only depending on C_1 and C' such that

$$\|\hat{\theta} - \theta\|^2 \leq C \frac{\sigma^2}{p} (k_1 k_2 + n_1 \log k_1 + n_2 \log k_2),$$

with probability at least $1 - \exp(-C'(k_1 k_2 + n_1 \log k_1 + n_2 \log k_2))$ uniformly over $\theta \in \Theta_{k_1, k_2}(M)$. For the symmetric parameter space $\Theta_k^s(M)$, the bound is simplified to

$$\|\hat{\theta} - \theta\|^2 \leq C \frac{\sigma^2}{p} (k^2 + n \log k),$$

with probability at least $1 - \exp(-C'(k^2 + n \log k))$ uniformly over $\theta \in \Theta_k^s(M)$.

We now present a rate matching lower bound in the Gaussian model to show that the result of Corollary 2 is minimax optimal. To this end, we use $\mathbb{P}_{(\theta, \sigma^2, p)}$ to indicate the probability distribution of the model $X_{ij} \stackrel{iid}{\sim} N(\theta_{ij}, \sigma^2)$ with observation rate p .

Theorem 3. Assume $\frac{\sigma^2}{p} \left(\frac{k_1 k_2}{n_1 n_2} + \frac{\log k_1}{n_2} + \frac{\log k_2}{n_1} \right) \lesssim M^2$. There exist some constants $C, c > 0$, such that

$$\inf_{\hat{\theta}} \sup_{\theta \in \Theta_{k_1, k_2}(M)} \mathbb{P}_{(\theta, \sigma^2, p)} \left(\|\hat{\theta} - \theta\|^2 > C \frac{\sigma^2}{p} (k_1 k_2 + n_1 \log k_1 + n_2 \log k_2) \right) > c,$$

when $\log k_1 \asymp \log k_2$, and

$$\inf_{\hat{\theta}} \sup_{\theta \in \Theta_k^s(M)} \mathbb{P}_{(\theta, \sigma^2, p)} \left(\|\hat{\theta} - \theta\|^2 > C \frac{\sigma^2}{p} (k^2 + n \log k) \right) > c.$$

The symmetric Bernoulli case When the observed matrix is symmetric with zero diagonal and Bernoulli random variables as its super-diagonal entries, it can be viewed as the adjacency matrix of an undirected network and the problem of estimating its mean matrix with missing data can be viewed as a network completion problem. Given a partially observed Bernoulli adjacency matrix $\{X_{ij}\}_{(i,j) \in \Omega}$, one can predict the unobserved edges by estimating the whole mean matrix θ . Also, we assume that each edge is observed independently with probability p .

Given a symmetric adjacency matrix $X = X^T \in \{0, 1\}^{n \times n}$ with zero diagonals, the stochastic block model (Holland et al., 1983) assumes $\{X_{ij}\}_{i>j}$ are independent Bernoulli random variables with mean $\theta_{ij} = Q_{z(i)z(j)} \in [0, 1]$ with some matrix $Q \in [0, 1]^{k \times k}$ and some label vector $z \in [k]^n$. In other words, the probability that there is an edge between the i th and the j th nodes only depends on their community labels $z(i)$ and $z(j)$. The following class then includes all possible mean matrices of stochastic block models with n nodes and k clusters and with edge probabilities uniformly bounded by ρ :

$$\Theta_k^+(\rho) = \left\{ \theta \in [0, 1]^{n \times n} : \theta_{ii} = 0, \theta_{ij} = \theta_{ji} = Q_{z(i)z(j)}, Q = Q^T \in [0, \rho]^{k \times k}, z \in [k]^n \right\}. \quad (11)$$

By the definition in (7), $\Theta_k^+(\rho) \subset \Theta_k^s(\rho)$.

Although the tail probability of Bernoulli random variables does not satisfy the sub-Gaussian assumption (4), a slightly modification of the proof of Theorem 1 leads to the following result. The proof of Corollary 4 will be given in Section A in the appendix.

Corollary 4. Consider the optimization problem (10) with $\Theta = \Theta_k^s(\rho)$. For any global optimizer $\hat{\theta}$ and any constant $C' > 0$, there exists a constant $C > 0$ only depending on C' such that

$$\|\hat{\theta} - \theta\|^2 \leq C \frac{\rho}{p} (k^2 + n \log k),$$

with probability at least $1 - \exp(-C'(k^2 + n \log k))$ uniformly over $\theta \in \Theta_k^s(\rho) \supset \Theta_k^+(\rho)$.

When $\rho = p = 1$, Corollary 4 implies Theorem 2.1 in Gao et al. (2015a). A rate matching lower bound is given by the following theorem. We denote the probability distribution of a stochastic block model with mean matrix $\theta \in \Theta_k^+(\rho)$ and observation rate p by $\mathbb{P}_{(\theta, p)}$.

Theorem 5. For stochastic block models, we have

$$\inf_{\hat{\theta}} \sup_{\theta \in \Theta_k^+(\rho)} \mathbb{P}_{(\theta, p)} \left(\|\hat{\theta} - \theta\|^2 > C \left(\frac{\rho(k^2 + n \log k)}{p} \wedge \rho^2 n^2 \right) \right) > c,$$

for some constants $C, c > 0$.

The lower bound is the minimum of two terms. When $\rho \geq \frac{k^2+n \log k}{pn^2}$, the rate becomes $\frac{\rho(k^2+n \log k)}{p} \wedge \rho^2 n^2 \asymp \frac{\rho(k^2+n \log k)}{p}$. It is achieved by the constrained least squares estimator according to Corollary 4. When $\rho < \frac{k^2+n \log k}{pn^2}$, the rate is dominated by $\rho^2 n^2$. In this case, a trivial zero estimator achieves the minimax rate.

In the case of $p = 1$, a comparable result has been found independently by Klopp et al. (2015). However, our result here is more general as it accommodates missing observations. Moreover, the general upper bounds in Theorem 1 even hold for networks with weighted edges.

4. Extensions

In this section, we extend the estimation procedure and the theory in Sections 2 and 3 toward three directions: adaptation to unknown observation rate, adaptation to unknown model parameters, and sparse graphon estimation.

4.1 Adaptation to unknown observation rate

The estimator (10) depends on the knowledge of the observation rate p . When p is not too small, such a knowledge is not necessary for achieving the desired rates. Define

$$\hat{p} = \frac{\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} E_{ij}}{n_1 n_2} \quad (12)$$

for the asymmetric and

$$\hat{p} = \frac{\sum_{1 \leq i < j \leq n} E_{ij}}{\frac{1}{2}n(n-1)} \quad (13)$$

for the symmetric case, and redefine

$$Y_{ij} = X_{ij} E_{ij} / \hat{p} \quad (14)$$

where the actual definition of \hat{p} is chosen between (12) and (13) depending on whether one is dealing with the asymmetric or symmetric parameter space. Then we have the following result for the solution to (10) with Y redefined by (14).

Theorem 6. For $\Theta = \Theta_{k_1 k_2}(M)$, suppose for some absolute constant $C_1 > 0$,

$$p \geq C_1 \frac{[\log(n_1 + n_2)]^2}{k_1 k_2 + n_1 \log k_1 + n_2 \log k_2}.$$

Let $\hat{\theta}$ be the solution to (10) with Y defined as in (14). Then for any constant $C' > 0$, there exists a constant $C > 0$ only depending on C' and C_1 such that

$$\|\hat{\theta} - \theta\|^2 \leq C \frac{M^2 \vee \sigma^2}{p} (k_1 k_2 + n_1 \log k_1 + n_2 \log k_2),$$

with probability at least $1 - (n_1 n_2)^{-C'}$ uniformly over $\theta \in \Theta$ and all error distributions satisfying (4).

For $\Theta = \Theta_k^s(M)$, the same result holds if we replace n_1 and n_2 with n and k_1 and k_2 with k in the foregoing statement.

4.2 Adaptation to unknown model parameters

We now provide an adaptive procedure for estimating θ without assuming the knowledge of the model parameters k_1, k_2 and M . The procedure can be regarded as a variation of a 2-fold cross validation (Wold, 1978). We give details on the procedure for the asymmetric parameter spaces $\Theta_{k_1 k_2}(M)$, and that for the symmetric parameter spaces $\Theta_k^s(M)$ can be obtained similarly.

To adapt to k_1, k_2 and M , we split the data into two halves. Namely, sample i.i.d. T_{ij}^b from $\text{Bernoulli}(\frac{1}{2})$. Define $\Delta = \{(i, j) \in [n_1] \times n_2 : T_{ij} = 1\}$. Define $Y_{ij}^\Delta = 2X_{ij} E_{ij} T_{ij} / p$ and $Y_{ij}^{\Delta^c} = 2X_{ij} E_{ij} (1 - T_{ij}) / p$ for all $(i, j) \in [n_1] \times [n_2]$. Then, for some given (k_1, k_2, M) , the least squares estimators using Y^Δ and Y^{Δ^c} are given by

$$\hat{\theta}_{k_1 k_2 M}^\Delta = \underset{\theta \in \Theta_{k_1 k_2}(M)}{\operatorname{argmin}} \|Y^\Delta - \theta\|^2, \quad \hat{\theta}_{k_1 k_2 M}^{\Delta^c} = \underset{\theta \in \Theta_{k_1 k_2}(M)}{\operatorname{argmin}} \|Y^{\Delta^c} - \theta\|^2.$$

Select the parameters by

$$(\hat{k}_1, \hat{k}_2, \hat{M}) = \underset{(k_1, k_2, M) \in [n_1] \times [n_2] \times \mathcal{M}}{\operatorname{argmin}} \|\hat{\theta}_{k_1 k_2 M}^\Delta - Y^{\Delta^c}\|_{\Delta^c}^2,$$

where $\mathcal{M} = \left\{ \frac{h}{n_1 + n_2} : h \in [(n_1 + n_2)^b] \right\}$, and define $\hat{\theta}^\Delta = \hat{\theta}_{\hat{k}_1 \hat{k}_2 \hat{M}}^\Delta$. Similarly, we can also define $\hat{\theta}^{\Delta^c}$ to validate the parameters using Y^{Δ^c} . The final estimator is given by

$$\hat{\theta}_{ij} = \begin{cases} \hat{\theta}_{ij}^{\Delta^c}, & (i, j) \in \Delta; \\ \hat{\theta}_{ij}^\Delta, & (i, j) \in \Delta^c. \end{cases}$$

Theorem 7. Assume $(n_1 + n_2)^{-1} \leq M \leq (n_1 + n_2)^5 - (n_1 + n_2)^{-1}$. For any constant $C' > 0$, there exists a constant $C > 0$ only depending on C' such that

$$\|\hat{\theta} - \theta\|^2 \leq C \frac{M^2 \vee \sigma^2}{p} \left(k_1 k_2 + n_1 \log k_1 + n_2 \log k_2 + \frac{\log(n_1 + n_2)}{p} \right),$$

with probability at least $1 - \exp(-C'(k_1 k_2 + n_1 \log k_1 + n_2 \log k_2)) - (n_1 n_2)^{-C'}$ uniformly over $\theta \in \Theta_{k_1 k_2}(M)$ and all error distributions satisfying (4).

Compared with Theorem 1, the rate given by Theorem 7 has an extra $p^{-1} \log(n_1 + n_2)$ term. A sufficient condition for this extra term to be inconsequential is $p \gtrsim \frac{\log(n_1 + n_2)}{n_1 \wedge n_2}$. Theorem 7 is adaptive for all $(k_1, k_2) \in [n_1] \times [n_2]$ and for $(n_1 + n_2)^{-1} \leq M \leq (n_1 + n_2)^5 - (n_1 + n_2)^{-1}$. In fact, by choosing a larger M , we can extend the adaptive region for M to $(n_1 + n_2)^{-a} \leq M \leq (n_1 + n_2)^b$ for arbitrary constants $a, b > 0$.

4.3 Sparse graphon estimation

Consider a random graph with adjacency matrix $\{X_{ij}\} \in \{0, 1\}^{n \times n}$, whose sampling procedure is determined by

$$(\xi_1, \dots, \xi_n) \sim \mathbb{P}_\xi, \quad X_{ij} | (\xi_i, \xi_j) \sim \text{Bernoulli}(\theta_{ij}), \quad \text{where } \theta_{ij} = f(\xi_i, \xi_j). \quad (15)$$

For $i \in [n]$, $X_{ii} = \theta_{ii} = 0$. Conditioning on (ξ_1, \dots, ξ_n) , $X_{ij} = X_{ji}$ is independent across $i > j$. The function f on $[0, 1]^2$, which is assumed to be symmetric, is called a graphon. The concept of graphon is originated from graph limit theory (Hoover, 1979; Lovász and Szegedy, 2006; Diaconis and Janson, 2007; Lovász, 2012) and the studies of exchangeable arrays (Aldous, 1981; Kallenberg, 1989). It is the underlying nonparametric object that generates the random graph. Statistical estimation of graphon has been considered by Wolfe and Olhede (2013); Olhede and Wolfe (2014); Gao et al. (2015a,b); Lu and Zhou (2015) for dense networks. Using Corollary 4, we present a result for sparse graphon estimation.

Let us start with specifying the function class of graphons. Define the derivative operator by

$$\nabla_{jk} f(x, y) = \frac{\partial^{j+k}}{(\partial x)^j (\partial y)^k} f(x, y),$$

and we adopt the convention $\nabla_{00} f(x, y) = f(x, y)$. The Hölder norm is defined as

$$\|f\|_{\mathcal{H}_\alpha} = \max_{j+k \leq [\alpha]} \sup_{x, y \in \mathcal{D}} |\nabla_{jk} f(x, y)| + \max_{j+k = [\alpha]} \sup_{(x, y) \neq (x', y') \in \mathcal{D}} \frac{|\nabla_{jk} f(x, y) - \nabla_{jk} f(x', y')|}{\|(x - x', y - y')\|_{\alpha - [\alpha]}},$$

where $\mathcal{D} = \{(x, y) \in [0, 1]^2 : x \geq y\}$. Then, the sparse graphon class with Hölder smoothness α is defined by

$$\mathcal{F}_\alpha(\rho, L) = \{0 \leq f \leq \rho : \|f\|_{\mathcal{H}_\alpha} \leq L\sqrt{\rho}, f(x, y) = f(y, x) \text{ for all } x \in \mathcal{D}\},$$

where $L > 0$ is the radius of the class, which is assumed to be a constant. As argued in Gao et al. (2015a), it is sufficient to approximate a graphon with Hölder smoothness by a piecewise constant function. In the random graph setting, a piecewise constant function is the stochastic block model. Therefore, we can use the estimator defined by (10). Using Corollary 4, a direct bias-variance tradeoff argument leads to the following result. An independent finding of the same result is also made by Klopp et al. (2015).

Corollary 8. *Consider the optimization problem (10) where $Y_{ij} = X_{ij}$ and $\Theta = \Theta_k^*(M)$ with $k = \lceil \frac{1}{n^{1-\alpha}} \rceil$ and $M = \rho$. Given any global optimizer $\hat{\theta}$ of (10), we estimate f by $\hat{f}(\xi_i, \xi_j) = \hat{\theta}_{ij}$. Then, for any constant $C' > 0$, there exists a constant $C > 0$ only depending on C' and L such that*

$$\frac{1}{n^2} \sum_{i, j \in [n]} \left(\hat{f}(\xi_i, \xi_j) - f(\xi_i, \xi_j) \right)^2 \leq C\rho \left(n^{-\frac{2\alpha}{\alpha+1}} + \frac{\log n}{n} \right),$$

with probability at least $1 - \exp(-C'(n^{\frac{1}{\alpha+1}} + n \log n))$ uniformly over $f \in \mathcal{F}_\alpha(\rho, L)$ and \mathbb{P}_ξ .

Corollary 8 implies an interesting phase transition phenomenon. When $\alpha \in (0, 1)$, the rate becomes $\rho(n^{-\frac{2\alpha}{\alpha+1}} + \frac{\log n}{n}) \asymp \rho n^{-\frac{2\alpha}{\alpha+1}}$, which is the typical nonparametric rate times a sparsity index of the network. When $\alpha \geq 1$, the rate becomes $\rho(n^{-\frac{2\alpha}{\alpha+1}} + \frac{\log n}{n}) \asymp \frac{\rho \log n}{n}$, which does not depend on the smoothness α . Corollary 8 extends Theorem 2.3 of Gao et al. (2015a) to the case $\rho < 1$. In Wolfe and Olhede (2013), the graphon f is defined in a different way. Namely, they considered the setting where (ξ_1, \dots, ξ_n) are i.i.d. Unif $[0, 1]$ random variables under \mathbb{P}_ξ . Then, the adjacency matrix is generated with

Bernoulli random variables having means $\theta_{ij} = \rho f(\xi_i, \xi_j)$ for a nonparametric graphon f satisfying $\int_0^1 \int_0^1 f(x, y) dx dy = 1$. For this setting, with appropriate smoothness assumption, we can estimate f by $\hat{f}(\xi_i, \xi_j) = \hat{\theta}_{ij}/\rho$. The rate of convergence would be $\rho^{-1}(n^{-\frac{2\alpha}{\alpha+1}} + \frac{\log n}{n})$.

Using the result of Theorem 7, we present an adaptive version for Corollary 8. The estimator we consider is a symmetric version of the one introduced in Section 4.2. The only difference is that we choose the set \mathcal{M} as $\{m/n : m \in [n+1]\}$. The estimator is fully data driven in the sense that it does not depend on α or ρ .

Corollary 9. *Assume $\rho \geq n^{-1}$. Consider the adaptive estimator $\hat{\theta}$ introduced in Theorem 7, and we set $\hat{f}(\xi_i, \xi_j) = \hat{\theta}_{ij}$. Then, for any constant $C' > 0$, there exists a constant $C > 0$ only depending on C' and L such that*

$$\frac{1}{n^2} \sum_{i, j \in [n]} \left(\hat{f}(\xi_i, \xi_j) - f(\xi_i, \xi_j) \right)^2 \leq C\rho \left(n^{-\frac{2\alpha}{\alpha+1}} + \frac{\log n}{n} \right),$$

with probability at least $1 - n^{-C}$ uniformly over $f \in \mathcal{F}_\alpha(\rho, L)$ and \mathbb{P}_ξ .

5. Numerical Studies

To introduce an algorithm solving (8) or (10), we write (10) in an alternative way,

$$\min_{z_1 \in [k_1]^{n_1}, z_2 \in [k_2]^{n_2}, Q \in [u, l]^{k_1 \times k_2}} L(Q, z_1, z_2),$$

where l and u are the lower and upper constraints of the parameters and

$$L(Q, z_1, z_2) = \sum_{(i, j) \in [n_1] \times [n_2]} (Y_{ij} - Q_{z_1(i)z_2(j)})^2.$$

For biclustering, we set $l = -M$ and $u = M$. For SBM, we set $l = 0$ and $u = \rho$. We do not impose symmetry for SBM to gain computational convenience without losing much statistical accuracy. The simple form of $L(Q, z_1, z_2)$ indicates that we can iteratively optimize over (Q, z_1, z_2) with explicit formulas. This motivates the following algorithm.

The iteration steps (16), (17) and (18) of Algorithm 1 can be equivalently expressed as

$$\begin{aligned} Q &= \operatorname{argmin}_{Q \in [u, l]^{k_1 \times k_2}} L(Q, z_1, z_2); \\ z_1 &= \operatorname{argmin}_{z_1 \in [k_1]^{n_1}} L(Q, z_1, z_2); \\ z_2 &= \operatorname{argmin}_{z_2 \in [k_2]^{n_2}} L(Q, z_1, z_2). \end{aligned}$$

Thus, each iteration will reduce the value of the objective function. It is worth noting that Algorithm 1 can be viewed as a two-way extension for the ordinary k -means algorithm. Since the objective function is non-convex, one cannot guarantee convergence to global optimum. We initialize the algorithm via a spectral clustering step using multiple random starting points, which worked well on simulated datasets we present below.

Now we present some numerical results to demonstrate the accuracy of the error rate behavior suggested by Theorem 1 on simulated data.

Algorithm 1: A Biclustering Algorithm

Input : $\{X_{ij}\}_{(i,j) \in \Omega}$, the numbers of column and row clusters (k_1, k_2) , lower and upper constraints (l, u) and the number of random starting points m .

Output: An $n_1 \times n_2$ matrix θ with $\theta_{ij} = Q_{z_1(i)z_2(j)}$.

Preprocessing: Let $X_{ij} = 0$ for $(i, j) \notin \Omega$, $\hat{p} = |\Omega|/(n_1 n_2)$ and $Y = X/\hat{p}$.

1 Initialization Step

Apply singular value decomposition and obtain $Y = UDV^T$.

Run k -means algorithm on the first k_1 columns of U with m random starting points to get z_1 .

Run k -means algorithm on the first k_2 columns of V with m random starting points to get z_2 .

while not converge do

2 Update Q : for each $(a, b) \in [k_1] \times [k_2]$,

$$Q_{ab} = \frac{1}{|z_1^{-1}(a)||z_2^{-1}(b)|} \sum_{i \in z_1^{-1}(a)} \sum_{j \in z_2^{-1}(b)} Y_{ij}. \quad (16)$$

If $Q_{ab} > u$, let $Q_{ab} = u$. If $Q_{ab} < l$, let $Q_{ab} = l$.

3 Update z_1 : for each $i \in [n_1]$,

$$z_1(i) = \operatorname{argmin}_{a \in [k_1]} \sum_{j=1}^{n_2} (Q_{az_2(j)} - A_{ij})^2. \quad (17)$$

4 Update z_2 : for each $j \in [n_2]$,

$$z_2(j) = \operatorname{argmin}_{b \in [k_2]} \sum_{i=1}^{n_1} (Q_{z_1(i)b} - A_{ij})^2. \quad (18)$$

Bernoulli case. Our theoretical result indicates the rate of recovery is $\sqrt{\frac{p}{n} \left(\frac{k_1^2}{n_1^2} + \frac{\log k_1}{n} \right)}$

for the root mean squared error (RMSE) $\frac{1}{n} \|\hat{\theta} - \theta\|$. When k is not too large, the dominating term is $\sqrt{\frac{p \log k}{pn}}$. We are going to confirm this rate by simulation. We first generate our data from SBM with the number of blocks $k \in \{2, 4, 8, 16\}$. The observation rate $p = 0.5$. For every fixed k , we use four different $Q = 0.5\mathbf{1}_k \mathbf{1}_k^T + 0.1\mathbf{I}_k$ with $t = 1, 2, 3, 4$ and generate the community labels z uniformly on $[k]$. Then we calculate the error $\frac{1}{n} \|\hat{\theta} - \theta\|$. Panel (a) of Figure 1 shows the error versus the sample size n . In Panel (b), we rescale the x-axis to $N = \sqrt{\frac{pn}{\log k}}$. The curves for different k align well with each other and the error decreases at the rate of $1/N$. This confirms our theoretical results in Theorem 1.

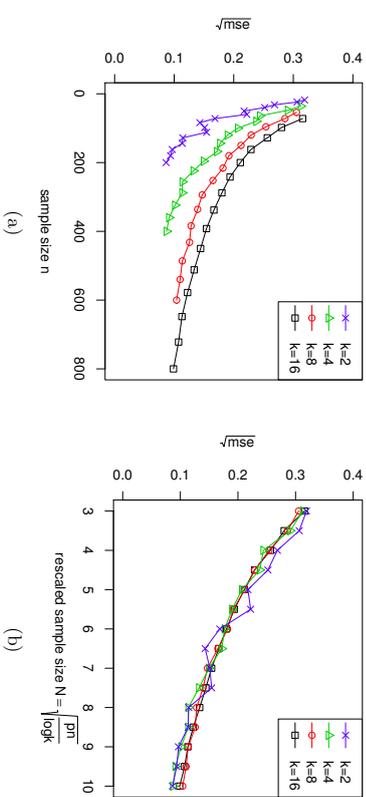


Figure 1: Plots of $\frac{1}{n} \|\hat{\theta} - \theta\|$ when using our algorithm on SBM. Each curve corresponds to a different sample size n with a fixed k . (a) Plots of error against the raw sample size n . (b) Plots of the same error against rescaled sample size $\sqrt{pn/\log k}$.

Gaussian case. We simulate data with Gaussian noise under four different settings of k_1 and k_2 . For each $(k_1, k_2) \in \{(4, 4), (4, 8), (8, 8), (8, 12)\}$, the entries of matrix Q are independently and uniformly generated from $\{1, 2, 3, 4, 5\}$. The cluster labels z_1 and z_2 are uniform on $[k_1]$ and $[k_2]$ respectively. After generating Q , z_1 and z_2 , we add an $N(0, 1)$ noise to the data and observe X_{ij} with probability $p = 0.1$. For each number of rows n_1 , we set the number of columns as $n_2 = n_1 \log k_1 / \log k_2$. Panel (a) of Figure 2 shows the error versus n_1 . In Panel (b), we rescale the x-axis by $N = \sqrt{\frac{pn_1}{\log k_2}}$. Again, the plots for different (k_1, k_2) align fairly well and the error decreases roughly at the rate of $1/N$.

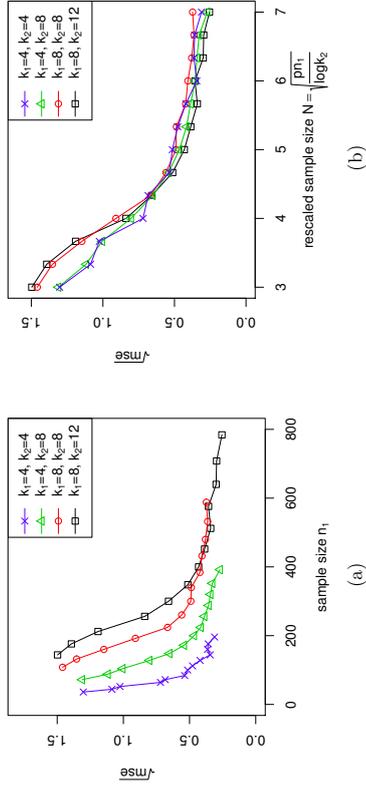


Figure 2: Plots of error $\frac{1}{\sqrt{n_1 n_2}} \|\hat{\theta} - \theta\|$ when using our algorithm on biclustering data with gaussian noise. Each curve corresponds to a fixed (k_1, k_2) . (a) Plots of error against n_1 . (b) Plots of the same error against $\sqrt{pn_1}/\log k_2$.

Sparse Bernoulli case. We also study recovery of sparse SBMs. We do the same simulation as the Bernoulli case except that we choose $Q = 0.02\mathbf{1}_k\mathbf{1}_k^T + 0.05\mathbf{I}_k$ for $t = 1, 2, 3, 4$. The results are shown in Figure 3.

Adaptation to unknown parameters. We use the 2-fold cross validation procedure proposed in Section 4.2 to adaptively choose the unknown number of clusters k and the sparsity level ρ . We use the setting of sparse SBM with the number of block $k \in \{4, 6\}$ and $Q = 0.05\mathbf{1}_k\mathbf{1}_k^T + 0.1\mathbf{I}_k$ for $t = 1, 2, 3, 4$. When running our algorithms, we search over all the (k, ρ) pair for $k \in \{2, 3, \dots, 8\}$ and $\rho \in \{0.2, 0.3, 0.4, 0.5\}$. In Table 1, we report the errors for different configurations of Q . The first row is the error obtained by our adaptive procedure and the second row is the error using the true k and ρ . Consistent with our Theorem 7, the error from the adaptive procedure is almost the same as the oracle error.

rescaled sample size	6	12	18	24
($k=4$) adaptive \sqrt{mse}	0.084	0.066	0.058	0.058
oracle \sqrt{mse}	0.085	0.069	0.060	0.053
($k=6$) adaptive \sqrt{mse}	0.074	0.061	0.051	0.050
oracle \sqrt{mse}	0.078	0.067	0.056	0.048

Table 1: Errors of the adaptive procedure versus the oracle.

The effect of constraints. The optimization (10) and Algorithm 1 involves the constraint $Q \in [l, u]^{k_1 \times k_2}$. It is curious whether this constraint really helps reduce the error or

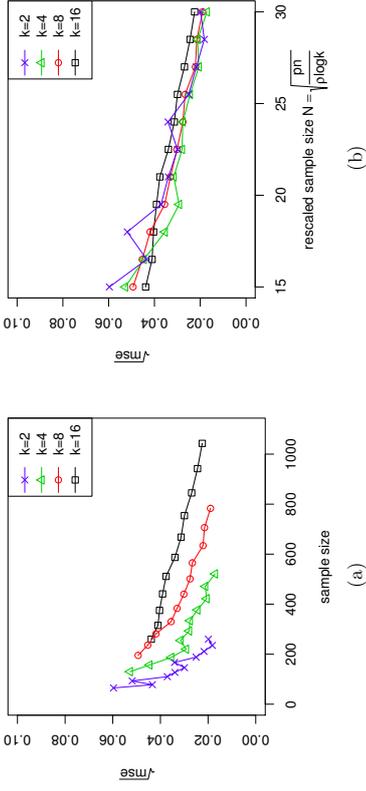


Figure 3: Plots of error $\frac{1}{n} \|\hat{\theta} - \theta\|$ when using our algorithm on sparse SBM. Each curve corresponds to a fixed k . (a) Plots of error against the raw sample size n . (b) Plots of the same error against rescaled sample size $\sqrt{pn}/\log k$.

merely an artifact of the proof. We investigate the effect of this constraint on simulated data by comparing Algorithm 1 with its variation without the constraint for both Gaussian case and sparse Bernoulli case. Panel (a) of Figure 4 shows the plots of sparse SBM with 8 communities. Panel (b) is the plots of Gaussian case with $(k_1, k_2) = (4, 8)$. For both panels, when the rescaled sample size is small, the effect of constraint is significant, while as the rescaled sample size increases, the performance of two estimators become similar.

6. Discussion

This paper studies the optimal rates of recovering a matrix with biclustering structure. While the recent progresses in high-dimensional estimation mainly focus on sparse and low rank structures, the study of biclustering structure does not gain much attention. This paper fills in the gap. In what follows, we discuss some key points of the paper and some possible future directions of research.

Difference from low-rankness. A biclustering structure is implicitly low-rank. Therefore, we show that by exploring the stronger biclustering assumption, one can achieve better rates of convergence in estimation and completion. The minimax rates derived in this paper precisely characterize how much one can gain by taking advantage of this structure.

Relation to other structures. A natural question to investigate is whether there is similarity between the biclustering structure and the well-studied sparsity structure. The paper Gao et al. (2015b) gives a general theory of structured estimation in linear models that puts both sparse and biclustering structures in a unified theoretical framework. According

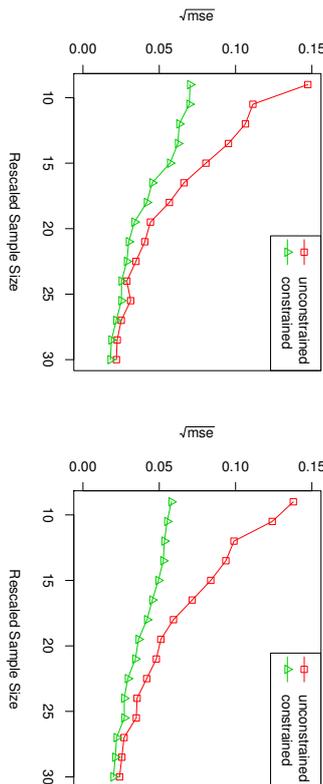


Figure 4: Constrained versus unconstrained least-squares
 (a) Sparse SBM with $k = 8$ (b) Gaussian biclustering with $(k_1, k_2) = (4, 8)$

to this general theory, the $k_1 k_2$ part in the minimax rate is the complexity of parameter estimation and the $n_1 \log k_1 + n_2 \log k_2$ part is the complexity of structure estimation.

Open problems. The optimization problem (10) is not convex, thus causing difficulty in devising a provably optimal polynomial-time algorithm. An open question is whether there is a convex relaxation of (10) that can be solved efficiently without losing much statistical accuracy. Another interesting problem for future research is whether the objective function in (10) can be extended beyond the least squares framework.

7. Proofs

7.1 Proof of Theorem 1

Below, we focus on the proof for the asymmetric parameter space $\Theta_{k_1, k_2}(M)$. The result for the symmetric parameter space $\Theta_k^s(M)$ can be obtained by letting $k_1 = k_2$ and by taking care of the diagonal entries. Since $\hat{\theta} \in \Theta_{k_1, k_2}(M)$, there exists $\hat{z}_1 \in [k_1]^{n_1}$, $\hat{z}_2 \in [k_2]^{n_2}$ and $\hat{Q} \in [-M, M]^{k_1 \times k_2}$ such that $\hat{\theta}_{ij} = \hat{Q}_{\hat{z}_1(i)\hat{z}_2(j)}$. For this (\hat{z}_1, \hat{z}_2) , we define a matrix $\tilde{\theta}$ by

$$\tilde{\theta}_{ij} = \frac{1}{|\hat{z}_1^{-1}(a)| |\hat{z}_2^{-1}(b)|} \sum_{(i,j) \in \hat{z}_1^{-1}(a) \times \hat{z}_2^{-1}(b)} \theta_{ij},$$

for any $(i, j) \in \hat{z}_1^{-1}(a) \times \hat{z}_2^{-1}(b)$ and any $(a, b) \in [k_1] \times [k_2]$. To facilitate the proof, we need to following three lemmas, whose proofs are given in the supplementary material.

Lemma 10. For any constant $C' > 0$, there exists a constant $C_1 > 0$ only depending on C' , such that

$$\|\hat{\theta} - \tilde{\theta}\|^2 \leq C_1 \frac{M^2 \vee \sigma^2}{p} (k_1 k_2 + n_1 \log k_1 + n_2 \log k_2),$$

with probability at least $1 - \exp(-C'(k_1 k_2 + n_1 \log k_1 + n_2 \log k_2))$.

Lemma 11. For any constant $C' > 0$, there exists a constant $C_2 > 0$ only depending on C' , such that the inequality $\|\hat{\theta} - \theta\|^2 \geq C_2 (M^2 \vee \sigma^2) (n_1 \log k_1 + n_2 \log k_2) / p$ implies

$$\left\langle \frac{\hat{\theta} - \theta}{\|\hat{\theta} - \theta\|}, Y - \theta \right\rangle \leq \sqrt{C_2 \frac{M^2 \vee \sigma^2}{p} (k_1 k_2 + n_1 \log k_1 + n_2 \log k_2)},$$

with probability at least $1 - \exp(-C'(k_1 k_2 + n_1 \log k_1 + n_2 \log k_2))$.

Lemma 12. For any constant $C' > 0$, there exists a constant $C_3 > 0$ only depending on C' , such that

$$\left| \langle \hat{\theta} - \tilde{\theta}, Y - \theta \rangle \right| \leq C_3 \frac{M^2 \vee \sigma^2}{p} (k_1 k_2 + n_1 \log k_1 + n_2 \log k_2),$$

with probability at least $1 - \exp(-C'(k_1 k_2 + n_1 \log k_1 + n_2 \log k_2))$.

Proof of Theorem 1. Applying union bound, the results of Lemma 10-12 hold with probability at least $1 - 3 \exp(-C'(k_1 k_2 + n_1 \log k_1 + n_2 \log k_2))$. We consider the following two cases.

Case 1: $\|\tilde{\theta} - \theta\|^2 \leq C_2 (M^2 \vee \sigma^2) (k_1 k_2 + n_1 \log k_1 + n_2 \log k_2) / p$. Then we have

$$\|\hat{\theta} - \theta\|^2 \leq 2\|\hat{\theta} - \tilde{\theta}\|^2 + 2\|\tilde{\theta} - \theta\|^2 \leq 2(C_1 + C_2) \frac{M^2 \vee \sigma^2}{p} (k_1 k_2 + n_1 \log k_1 + n_2 \log k_2)$$

by Lemma 10.

Case 2: $\|\tilde{\theta} - \theta\|^2 > C_2 (M^2 \vee \sigma^2) (k_1 k_2 + n_1 \log k_1 + n_2 \log k_2) / p$.

By the definition of the estimator, we have $\|\hat{\theta} - Y\|^2 \leq \|\hat{\theta} - Y\|^2$. After rearrangement, we have

$$\begin{aligned} \|\hat{\theta} - \theta\|^2 &\leq 2 \langle \hat{\theta} - \theta, Y - \theta \rangle \\ &= 2 \langle \hat{\theta} - \tilde{\theta}, Y - \theta \rangle + 2 \langle \tilde{\theta} - \theta, Y - \theta \rangle \\ &\leq 2 \langle \hat{\theta} - \tilde{\theta}, Y - \theta \rangle + 2\|\tilde{\theta} - \theta\| \left\langle \frac{\tilde{\theta} - \theta}{\|\tilde{\theta} - \theta\|}, Y - \theta \right\rangle \\ &\leq 2 \langle \hat{\theta} - \tilde{\theta}, Y - \theta \rangle + 2(\|\tilde{\theta} - \theta\| + \|\hat{\theta} - \theta\|) \left\langle \frac{\hat{\theta} - \theta}{\|\hat{\theta} - \theta\|}, Y - \theta \right\rangle \\ &\leq 2(C_2 + C_3 + \sqrt{C_1 C_2}) \frac{M^2 \vee \sigma^2}{p} (k_1 k_2 + n_1 \log k_1 + n_2 \log k_2) + \frac{1}{2} \|\hat{\theta} - \theta\|^2, \end{aligned}$$

which leads to the bound

$$\|\hat{\theta} - \theta\|^2 \leq 4(C_2 + C_3 + \sqrt{C_1 C_2}) \frac{M^2 \vee \sigma^2}{p} (k_1 k_2 + n_1 \log k_1 + n_2 \log k_2).$$

Combining the two cases, we have

$$\|\hat{\theta} - \theta\|^2 \leq C \frac{M^2 \vee \sigma^2}{p} (k_1 k_2 + n_1 \log k_1 + n_2 \log k_2),$$

with probability at least $1 - 3 \exp(-C'(k_1 k_2 + n_1 \log k_1 + n_2 \log k_2))$ for $C = 4(C_2 + C_3 + \sqrt{C_1 C_2}) \vee 2(C_1 + C_2)$. \square

7.2 Proof of Theorem 7

We first present a lemma for the tail behavior of sum of independent products of sub-Gaussian and Bernoulli random variables. Its proof is given in the supplementary material.

Lemma 13. *Let $\{X_i\}$ be independent sub-Gaussian random variables with mean $\theta_i \in [-M, M]$ and $\mathbb{E}e^{\lambda(X_i - \theta_i)} \leq e^{\lambda^2 \sigma^2 / 2}$. Let $\{E_i\}$ be independent Bernoulli random variables with mean p . Assume $\{X_i\}$ and $\{E_i\}$ are all independent. Then for $|\lambda| \leq p/(M \vee \sigma)$ and $Y_i = X_i E_i / p$, we have*

$$\mathbb{E}e^{\lambda(Y_i - \theta_i)} \leq 2e^{(M^2 + 2\sigma^2)\lambda^2 / p}.$$

Moreover, for $\sum_{i=1}^n c_i^2 = 1$,

$$\mathbb{P} \left\{ \left| \sum_{i=1}^n c_i (Y_i - \theta_i) \right| \geq t \right\} \leq 4 \exp \left\{ - \min \left(\frac{pt^2}{4(M^2 + 2\sigma^2)}, 2(M \vee \sigma) \|c\|_\infty \right) \right\} \quad (19)$$

for any $t > 0$.

Proof of Theorem 7. Consider the mean matrix θ that belongs to the space $\Theta_{k_1, k_2}(M)$. By the definition of $(\hat{k}_1, \hat{k}_2, \hat{M})$, we have $\|\hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^\Delta - Y^{\Delta^c}\|_{\Delta^c}^2 \leq \|\hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^\Delta - Y^{\Delta^c}\|_{\Delta^c}^2$, where k_1 and k_2 are true numbers of row and column clusters and m is chosen to be the smallest element in \mathcal{M} that is no smaller than M . After rearrangement, we have

$$\begin{aligned} & \|\hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^\Delta - \theta\|_{\Delta^c}^2 \\ & \leq \|\hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^\Delta - \theta\|_{\Delta^c}^2 + 2\|\hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^\Delta - \hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^\Delta\|_{\Delta^c} \left\langle \frac{\hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^\Delta - \hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^\Delta}{\|\hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^\Delta - \hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^\Delta\|_{\Delta^c}}, Y^{\Delta^c} - \theta \right\rangle_{\Delta^c} \\ & \leq \|\hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^\Delta - \theta\|_{\Delta^c}^2 + 2\|\hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^\Delta - \hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^\Delta\|_{\Delta^c} \max_{(l_1, l_2) \in [n_1] \times [n_2]} \left\langle \frac{\hat{\theta}_{l_1 l_2 h}^\Delta - \hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^\Delta}{\|\hat{\theta}_{l_1 l_2 h}^\Delta - \hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^\Delta\|_{\Delta^c}}, Y^{\Delta^c} - \theta \right\rangle_{\Delta^c}. \end{aligned}$$

By Lemma 13 and the independence structure, we have

$$\max_{(l_1, l_2, h) \in [n_1] \times [n_2] \times \mathcal{M}} \left| \left\langle \frac{\hat{\theta}_{l_1 l_2 h}^\Delta - \hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^\Delta}{\|\hat{\theta}_{l_1 l_2 h}^\Delta - \hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^\Delta\|_{\Delta^c}}, Y^{\Delta^c} - \theta \right\rangle_{\Delta^c} \right| \leq C(M \vee \sigma) \frac{\log(n_1 + n_2)}{p},$$

with probability at least $1 - (n_1 n_2)^{-C'}$. Using triangle inequality and Cauchy-Schwarz inequality, we have

$$\|\hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^\Delta - \theta\|_{\Delta^c}^2 \leq \frac{3}{2} \|\hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^\Delta - \theta\|_{\Delta^c}^2 + \frac{1}{2} \|\hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^\Delta - \theta\|_{\Delta^c}^2 + 4C^2(M^2 \vee \sigma^2) \left(\frac{\log(n_1 + n_2)}{p} \right)^2.$$

By rearranging the above inequality, we have

$$\|\hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^\Delta - \theta\|_{\Delta^c}^2 \leq 3\|\hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^\Delta - \theta\|_{\Delta^c}^2 + 8C^2(M^2 \vee \sigma^2) \left(\frac{\log(n_1 + n_2)}{p} \right)^2.$$

A symmetric argument leads to

$$\|\hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^{\Delta^c} - \theta\|_{\Delta}^2 \leq 3\|\hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^{\Delta^c} - \theta\|_{\Delta}^2 + 8C^2(M^2 \vee \sigma^2) \left(\frac{\log(n_1 + n_2)}{p} \right)^2.$$

Summing up the above two inequalities, we have

$$\|\hat{\theta} - \theta\|^2 \leq 3\|\hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^\Delta - \theta\|^2 + 3\|\hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^{\Delta^c} - \theta\|^2 + 16C^2(M^2 \vee \sigma^2) \left(\frac{\log(n_1 + n_2)}{p} \right)^2. \quad (20)$$

Using Theorem 1 to bound $\|\hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^\Delta - \theta\|^2$ and $\|\hat{\theta}_{\hat{k}_1, \hat{k}_2, \hat{M}}^{\Delta^c} - \theta\|^2$ can be bounded by $C \frac{m^2 \vee \sigma^2}{p} (k_1 k_2 + n_1 \log k_1 + n_2 \log k_2)$. Given that $m = M \left(1 + \frac{M}{m}\right) \leq 2M$ by the choice of m , the proof is complete. \square

7.3 Proof of Theorem 6

Recall the augmented data $Y_{ij} = X_{ij} E_{ij} / p$. Define $\mathcal{Y}_{ij} = X_{ij} E_{ij} / \hat{p}$. Let us give two lemmas to facilitate the proof.

Lemma 14. *Assume $p \gtrsim \frac{\log(n_1 + n_2)}{n_1 n_2}$. For any $C' > 0$, there is some constant $C > 0$ such that*

$$\|Y - \mathcal{Y}\|^2 \leq C \left[M^2 + \sigma^2 \log(n_1 + n_2) \right] \frac{\log(n_1 + n_2)}{p^2},$$

with probability at least $1 - (n_1 n_2)^{-C'}$.

Lemma 15. *The inequalities in Lemma 10-12 continue to hold with bounds*

$$\begin{aligned} C_1 \frac{M^2 \vee \sigma^2}{p} (k_1 k_2 + n_1 \log k_1 + n_2 \log k_2) + 2\|Y - \mathcal{Y}\|^2, \\ \sqrt{C_2 \frac{M^2 \vee \sigma^2}{p} (k_1 k_2 + n_1 \log k_1 + n_2 \log k_2) + \|Y - \mathcal{Y}\|}, \\ C_3 \frac{M^2 \vee \sigma^2}{p} (k_1 k_2 + n_1 \log k_1 + n_2 \log k_2) + \|\hat{\theta} - \tilde{\theta}\| \|Y - \mathcal{Y}\|, \end{aligned}$$

and

respectively.

Proof of Theorem 6. The proof is similar to that of Theorem 1. We only need to replace Lemma 10-12 by Lemma 14 and Lemma 15 to get the desired result. \square

7.4 Proofs of Theorem 3 and Theorem 5

This section gives proofs of the minimax lower bounds. We first introduce some notation. For any probability measures \mathbb{P}, \mathbb{Q} , define the Kullback-Leibler divergence by $D(\mathbb{P} \parallel \mathbb{Q}) = \int \left(\log \frac{d\mathbb{P}}{d\mathbb{Q}} \right) d\mathbb{P}$. The chi-squared divergence is defined by $\chi^2(\mathbb{P} \parallel \mathbb{Q}) = \int \left(\frac{d\mathbb{P}}{d\mathbb{Q}} - 1 \right)^2 d\mathbb{P} - 1$. The main tool we will use is the following proposition.

Proposition 16. *Let (Ξ, ℓ) be a metric space and $\{\mathbb{P}_\xi : \xi \in \Xi\}$ be a collection of probability measures. For any totally bounded $T \subseteq \Xi$, define the Kullback-Leibler diameter and the chi-squared diameter of T by*

$$d_{KL}(T) = \sup_{\xi, \xi' \in T} D(\mathbb{P}_\xi \parallel \mathbb{P}_{\xi'}), \quad d_{\chi^2}(T) = \sup_{\xi, \xi' \in T} \chi^2(\mathbb{P}_\xi \parallel \mathbb{P}_{\xi'}).$$

Then

$$\inf_{\xi \in \Xi} \sup_{\mathbb{P}^\xi} \mathbb{P}^\xi \left\{ \rho^2(\hat{\xi}(X), \xi) \geq \frac{\epsilon^2}{4} \right\} \geq 1 - \frac{d_{KL}(T) + \log 2}{\log \mathcal{M}(\epsilon, T, \theta)}, \quad (21)$$

$$\inf_{\xi \in \Xi} \sup_{\mathbb{P}^\xi} \left\{ \rho^2(\hat{\xi}(X), \xi) \geq \frac{\epsilon^2}{4} \right\} \geq 1 - \frac{1}{\mathcal{M}(\epsilon, T, \theta)} - \sqrt{\frac{d_{\chi^2}(T)}{\mathcal{M}(\epsilon, T, \theta)}}, \quad (22)$$

for any $\epsilon > 0$, where the packing number $\mathcal{M}(\epsilon, T, \theta)$ is the largest number of points in T that are at least ϵ away from each other.

The inequality (21) is the classical Fano's inequality. The version we present here is by Yu (1997). The inequality (22) is a generalization of the classical Fano's inequality by using chi-squared divergence instead of KL divergence. It is due to Guntuboyina (2011).

The following proposition bounds the KL divergence and the chi-squared divergence for both Gaussian and Bernoulli models.

Proposition 17. For the Gaussian model, we have

$$D(\mathbb{P}^{(\theta, \sigma^2, \rho)} \| \mathbb{P}^{(\theta', \sigma'^2, \rho)}) \leq \frac{p}{2\sigma^2} \|\theta - \theta'\|^2, \quad \chi^2(\mathbb{P}^{(\theta, \sigma^2, \rho)} \| \mathbb{P}^{(\theta', \sigma'^2, \rho)}) \leq \exp\left(\frac{p}{\sigma^2} \|\theta - \theta'\|^2\right) - 1.$$

For the Bernoulli model with any $\theta, \theta' \in [\rho/2, 3\rho/4]^{n_1 \times n_2}$, we have

$$D(\mathbb{P}^{(\theta, \rho)} \| \mathbb{P}^{(\theta', \rho)}) \leq \frac{8\rho}{\rho} \|\theta - \theta'\|^2, \quad \chi^2(\mathbb{P}^{(\theta, \rho)} \| \mathbb{P}^{(\theta', \rho)}) \leq \exp\left(\frac{8\rho}{\rho} \|\theta - \theta'\|^2\right) - 1.$$

Finally, we need the following Varshamov-Gilbert bound. The version we present here is due to (Massart, 2007, Lemma 4.7).

Lemma 18. There exists a subset $\{\omega_1, \dots, \omega_N\} \subset \{0, 1\}^d$ such that

$$H(\omega_i, \omega_j) \triangleq \|\omega_i - \omega_j\|^2 \geq \frac{d}{4}, \quad \text{for any } i \neq j \in [N], \quad (23)$$

for some $N \geq \exp(d/8)$.

Proof of Theorem 3. We focus on the proof for the asymmetric parameter space $\Theta_{k_1 k_2}(M)$. The result for the symmetric parameter space $\Theta_k^*(M)$ can be obtained by letting $k_1 = k_2 = k$ and by taking care of the diagonal entries. Let us assume n_1/k_1 and n_2/k_2 are integers without loss of generality. We first derive the lower bound for the nonparametric rate $\sigma^2 k_1 k_2 / p$. Let us fix the labels by $z_1(i) = [ik_1/n_1]$ and $z_2(j) = [jk_2/n_2]$. For any $\omega \in \{0, 1\}^{k_1 \times k_2}$, define

$$Q_{ab}^\omega = c \sqrt{\frac{\sigma^2 k_1 k_2}{pn_1 n_2}} \omega_{ab}. \quad (24)$$

By Lemma 18, there exists some $T \subset \{0, 1\}^{k_1 k_2}$ such that $|T| \geq \exp(k_1 k_2 / 8)$ and $H(\omega, \omega') \geq k_1 k_2 / 4$ for any $\omega, \omega' \in T$ and $\omega \neq \omega'$. We construct the subspace

$$\Theta(z_1, z_2, T) = \left\{ \theta \in \mathbb{R}^{n_1 \times n_2} : \theta_{ij} = Q_{z_1(i)z_2(j)}^\omega, \omega \in T \right\}.$$

By Proposition 17, we have

$$\sup_{\theta, \theta' \in \Theta(z_1, z_2, T)} \chi^2(\mathbb{P}^{(\theta, \sigma^2, \rho)} \| \mathbb{P}^{(\theta', \sigma'^2, \rho)}) \leq \exp(c^2 k_1 k_2).$$

For any two different θ and θ' in $\Theta(z_1, z_2, T)$ associated with $\omega, \omega' \in T$, we have

$$\|\theta - \theta'\|^2 \geq \frac{c^2 \sigma^2}{p} H(\omega, \omega') \geq \frac{c^2 \sigma^2}{4p} k_1 k_2.$$

Therefore, $\mathcal{M}\left(\sqrt{\frac{c^2 \sigma^2}{4p}} k_1 k_2, \Theta(z_1, z_2, T), \|\cdot\|\right) \geq \exp(k_1 k_2 / 8)$. Using (22) with an appropriate c , we have obtained the rate $\frac{c^2}{p} k_1 k_2$ in the lower bound.

Now let us derive the clustering rate $\sigma^2 n_2 \log k_2 / p$. Let us pick $\omega_1, \dots, \omega_{k_2} \in \{0, 1\}^{k_1}$ such that $H(\omega_a, \omega_b) \geq \frac{k_1}{4}$ for all $a \neq b$. By Lemma 18, this is possible when $\exp(k_1/8) \geq k_2$. Then, define

$$Q_{*a} = c \sqrt{\frac{\sigma^2 n_2 \log k_2}{pn_1 n_2}} \omega_a. \quad (25)$$

Define $z_1(i) = [ik_1/n_1]$. Fix Q and z_1 and we are going to let z_2 vary. Select a set $\mathcal{Z}_2 \subset [k_2]^{n_2}$ such that $|\mathcal{Z}_2| \geq \exp(Cn_2 \log k_2)$ and $H(z_2, z'_2) \geq \frac{n_2}{6}$ for any $z_2, z'_2 \in \mathcal{Z}_2$ and $z_2 \neq z'_2$. The existence of such \mathcal{Z}_2 is proved by Gao et al. (2015a). Then, the subspace we consider is

$$\Theta(z_1, \mathcal{Z}_2, Q) = \left\{ \theta \in \mathbb{R}^{n_1 \times n_2} : \theta_{ij} = Q_{z_1(i)z_2(j)}, z_2 \in \mathcal{Z}_2 \right\}.$$

By Proposition 17, we have

$$\sup_{\theta, \theta' \in \Theta(z_1, \mathcal{Z}_2, Q)} D(\mathbb{P}^{(\theta, \sigma^2, \rho)} \| \mathbb{P}^{(\theta', \sigma'^2, \rho)}) \leq c^2 n_2 \log k_2.$$

For any two different θ and θ' in $\Theta(z_1, \mathcal{Z}_2, Q)$ associated with $z_2, z'_2 \in \mathcal{Z}_2$, we have

$$\|\theta - \theta'\|^2 = \sum_{j=1}^{n_2} \|\theta_{z_2(j)} - \theta'_{z_2(j)}\|^2 \geq H(z_2, z'_2) \frac{c^2 \sigma^2 n_2 \log k_2 n_1}{pn_1 n_2} \geq \frac{c^2 \sigma^2 n_2 \log k_2}{24p}.$$

Therefore, $\mathcal{M}\left(\sqrt{\frac{c^2 \sigma^2 n_2 \log k_2}{24p}}, \Theta(z_1, \mathcal{Z}_2, Q), \|\cdot\|\right) \geq \exp(Cn_2 \log k_2)$. Using (21) with some appropriate c , we obtain the lower bound $\frac{c^2 n_2 \log k_2}{p}$.

A symmetric argument gives the rate $\frac{c^2 n_1 \log k_1}{p}$. Combining the three parts using the same argument in Gao et al. (2015a), the proof is complete. \square

Proof of Theorem 5. The proof is similar to that of Theorem 3. The only differences are (24) replaced by

$$Q_{ab}^\omega = \frac{1}{2} \rho + \left(c \sqrt{\frac{\rho k^2}{pn^2} \wedge \frac{1}{2} \rho} \right) \omega_{ab}$$

and (25) replaced by

$$Q_{*a} = \frac{1}{2} \rho + \left(c \sqrt{\frac{\rho \log k}{pn} \wedge \frac{1}{2} \rho} \right) \omega_a.$$

It is easy to check that the constructed subspaces are subsets of $\Theta_k^+(\rho)$. Then, a symmetric modification of the proof of Theorem 3 leads to the desired conclusion. \square

7.5 Proofs of Corollary 8 and Corollary 9

The result of Corollary 8 can be derived through a standard bias-variance trade-off argument by combining Corollary 4 and Lemma 2.1 in Gao et al. (2015a). The result of Corollary 9 follows Theorem 7. By studying the proof of Theorem 7, (20) holds for all k . Choosing the best k to trade-off bias and variance gives the result of Corollary 9. We omit the details here.

Acknowledgments

The researches of CG, YL and HHZ are supported in part by NSF grant DMS-1507511. The research of ZM is supported in part by NSF Career grant DMS-1352060.

Appendix A. Proofs of auxiliary results

In this section, we give proofs of Lemma 10-14. We first introduce some notation. Define the set

$$\mathcal{Z}_{k_1, k_2} = \{z = (z_1, z_2) : z_1 \in [k_1]^{n_1}, z_2 \in [k_2]^{n_2}\}.$$

For a matrix $G \in \mathbb{R}^{n_1 \times n_2}$ and some $z = (z_1, z_2) \in \mathcal{Z}_{k_1, k_2}$, define

$$\bar{G}_{ab}(z) = \frac{1}{|z_1^{-1}(a)||z_2^{-1}(b)|} \sum_{(i,j) \in z_1^{-1}(a) \times z_2^{-1}(b)} G_{ij},$$

for all $a \in [k_1]$, $b \in [k_2]$. To facilitate the proof, we need the following two results.

Proposition 19. For the estimator $\hat{\theta}_{ij} = \hat{Q}_{z_1(i)z_2(j)}$, we have

$$\hat{Q}_{ab} = \text{sign}(\bar{Y}_{ab}(\hat{z})) (\bar{Y}_{ab}(\hat{z}) \wedge M),$$

for all $a \in [k_1]$, $b \in [k_2]$.

Lemma 20. Under the setting of Lemma 13, define $S = \frac{1}{\sqrt{n}} \sum_{i=1}^n (Y_i - \theta_i)$ and $\tau = 2(M^2 + 2\sigma^2)/(M \vee \sigma)$. Then we have the following results:

- Let $T = \mathbf{S}\mathbf{1}\{|S| \leq \tau\sqrt{n}\}$, then $\mathbb{E}e^{T^2/(8(M^2+2\sigma^2))} \leq 5$;
- Let $R = \tau\sqrt{n}\mathbf{S}\mathbf{1}\{|S| > \tau\sqrt{n}\}$, then $\mathbb{E}e^{R^2/(8(M^2+2\sigma^2))} \leq 9$.

Proof. By (19),

$$\mathbb{P}(|S| > t) \leq 4 \exp \left\{ -\min \left(\frac{pt^2}{4(M^2 + 2\sigma^2)}, \frac{\sqrt{np}t}{2(M \vee \sigma)} \right) \right\}.$$

Then

$$\begin{aligned} \mathbb{E}e^{M^2} &= \int_0^\infty \mathbb{P}(e^{M^2} > u) du \leq 1 + \int_1^\infty \mathbb{P}(|T| > \sqrt{\frac{\log u}{\lambda}}) du \\ &= 1 + \int_1^{e^{\lambda^2 n}} \mathbb{P}(|S| > \sqrt{\frac{\log u}{\lambda}}) du = 1 + 4 \int_1^{e^{\lambda^2 n}} u^{-p/(4\lambda(M^2+2\sigma^2))} du. \end{aligned}$$

Choosing $\lambda = p/(8(M^2 + 2\sigma^2))$, we get $\mathbb{E}e^{pT^2/(8(M^2+2\sigma^2))} \leq 5$. We proceed to prove the second claim.

$$\begin{aligned} \mathbb{E}e^{\lambda R} &= \mathbb{P}(R=0) + \mathbb{P}(R>0)\mathbb{E}[e^{\lambda R_1} | R>0] \\ &= \mathbb{P}(R=0) + \mathbb{P}(R>0) \int_0^\infty \mathbb{P}(e^{\lambda R} > u | R > 0) du \\ &= \int_0^\infty \mathbb{P}(e^{\lambda R} > u, R > 0) du \\ &\leq \mathbb{P}(R=0) + \mathbb{P}(R>0)e^{\lambda\tau^2 n} + \int_{e^{\lambda\tau^2 n}}^\infty \mathbb{P}(e^{\lambda R} > u) du \\ &\leq 1 + 4e^{-p\tau^2 n/(4(M^2+2\sigma^2)) + \lambda\tau^2 n} + \int_{e^{\lambda\tau^2 n}}^\infty \mathbb{P}(e^{\sqrt{n}\lambda\tau|S|} > u) du \\ &= 1 + 4e^{-p\tau^2 n/(4(M^2+3\sigma^2)) + \lambda\tau^2 n} + 4 \int_{e^{\lambda\tau^2 n}}^\infty u^{-p/(2\lambda\tau(M\vee\sigma))} du \end{aligned}$$

Choosing $\lambda = p/(8(M^2 + 2\sigma^2))$, we get $\mathbb{E}e^{pR/(8(M^2+2\sigma^2))} \leq 9$. \square

Proof of Lemma 10. By the definitions of $\hat{\theta}_{ij}$ and $\tilde{\theta}_{ij}$ and Proposition 19, we have

$$\hat{\theta}_{ij} - \tilde{\theta}_{ij} = \begin{cases} M - \bar{\theta}_{ab}(\hat{z}), & \text{if } \bar{Y}_{ab}(\hat{z}) \geq M; \\ \bar{Y}_{ab}(\hat{z}) - \bar{\theta}_{ab}(\hat{z}), & \text{if } -M \leq \bar{Y}_{ab}(\hat{z}) < M; \\ -M - \bar{\theta}_{ab}(\hat{z}), & \text{if } \bar{Y}_{ab}(\hat{z}) < -M \end{cases}$$

for any $(i, j) \in \hat{z}_1^{-1}(a) \times \hat{z}_2^{-1}(b)$. Define $W = Y - \theta$, and it is easy to check that

$$|\hat{\theta}_{ij} - \tilde{\theta}_{ij}| \leq |\bar{W}_{ab}(\hat{z})| \wedge 2M \leq |\bar{W}_{ab}(\hat{z})| \wedge \tau,$$

where $\hat{z} = (\hat{z}_1, \hat{z}_2)$ and τ is defined in Lemma 20. Then

$$\begin{aligned} \|\hat{\theta} - \tilde{\theta}\|^2 &\leq \sum_{a \in [k_1], b \in [k_2]} |\hat{z}_1^{-1}(a)| |\hat{z}_2^{-1}(b)| \left(|\bar{W}_{ab}(\hat{z})| \wedge \tau \right)^2 \\ &\leq \max_{z \in \mathcal{Z}_{k_1, k_2}} \sum_{a \in [k_1], b \in [k_2]} |z_1^{-1}(a)| |z_2^{-1}(b)| \left(|\bar{W}_{ab}(z)| \wedge \tau \right)^2. \end{aligned} \quad (26)$$

For any $a \in [k_1]$, $b \in [k_2]$ and $z_1 \in [k_1]^{n_1}$, $z_2 \in [k_2]^{n_2}$, define $n_1(a) = |z_1^{-1}(a)|$, $n_2(b) = |z_2^{-1}(b)|$ and

$$\begin{aligned} V_{ab}(z) &= \sqrt{n_1(a)n_2(b)} |\bar{W}_{ab}(z)| \mathbf{1}\{|\bar{W}_{ab}(z)| \leq \tau\}, \\ R_{ab}(z) &= n_1(a)n_2(b)\tau |\bar{W}_{ab}(z)| \mathbf{1}\{|\bar{W}_{ab}(z)| > \tau\}. \end{aligned}$$

Then,

$$\|\hat{\theta} - \tilde{\theta}\|^2 \leq \max_{z \in \mathcal{Z}_{k_1, k_2}} \sum_{a \in [k_1], b \in [k_2]} (V_{ab}^2(z) + R_{ab}(z)). \quad (27)$$

By Markov's inequality and Lemma 20, we have

$$\begin{aligned} \mathbb{P} \left(\sum_{\alpha \in [k_1], \beta \in [k_2]} V_{\alpha\beta}^2(z) > t \right) &\leq e^{-pt/(8(M^2+2\sigma^2))} \prod_{\alpha \in [k_1], \beta \in [k_2]} e^{pV_{\alpha\beta}^2(z)/(8(M^2+2\sigma^2))} \\ &\leq \exp \left\{ -\frac{pt}{8(M^2+2\sigma^2)} + k_1 k_2 \log 5 \right\}, \end{aligned}$$

and

$$\begin{aligned} \mathbb{P} \left(\sum_{\alpha \in [k_1], \beta \in [k_2]} R_{\alpha\beta}(z) > t \right) &\leq e^{-pt/(8(M^2+2\sigma^2))} \prod_{\alpha \in [k_1], \beta \in [k_2]} e^{pR_{\alpha\beta}(z)/(8(M^2+2\sigma^2))} \\ &\leq \exp \left\{ -\frac{pt}{8(M^2+2\sigma^2)} + k_1 k_2 \log 9 \right\}, \end{aligned}$$

Applying union bound and using the fact that $\log |k_1|^{n_1} + \log |[k_2]^{n_2}| = n_1 \log k_1 + n_2 \log k_2$,

$$\mathbb{P} \left(\max_{z \in \mathcal{Z}_{k_1, k_2}} \sum_{\alpha \in [k_1], \beta \in [k_2]} V_{\alpha\beta}^2(z) > t \right) \leq \exp \left\{ -\frac{pt}{8(M^2+2\sigma^2)} + k_1 k_2 \log 5 + n_1 \log k_1 + n_2 \log k_2 \right\}.$$

For any given constant $C' > 0$, we choose $t = C_1 \frac{M^2 \vee \sigma^2}{p} (k_1 k_2 + n_1 \log k_1 + n_2 \log k_2)$ for some sufficiently large $C_1 > 0$ to obtain

$$\max_{z \in \mathcal{Z}_{k_1, k_2}} \sum_{\alpha \in [k_1], \beta \in [k_2]} V_{\alpha\beta}^2(z) \leq C_1 \frac{M^2 \vee \sigma^2}{p} (k_1 k_2 + n_1 \log k_1 + n_2 \log k_2) \quad (28)$$

with probability at least $1 - \exp(-C'(k_1 k_2 + n_1 \log k_1 + n_2 \log k_2))$. Similarly, for some sufficiently large $C_2 > 0$, we have

$$\max_{z \in \mathcal{Z}_{k_1, k_2}} \sum_{\alpha \in [k_1], \beta \in [k_2]} R_{\alpha\beta}(z) \leq C_2 \frac{M^2 \vee \sigma^2}{p} (k_1 k_2 + n_1 \log k_1 + n_2 \log k_2) \quad (29)$$

with probability at least $1 - \exp(-C'(k_1 k_2 + n_1 \log k_1 + n_2 \log k_2))$. Plugging (28) and (29) into (27), we complete the proof. \square

Proof of Lemma 11. Note that

$$\tilde{\theta}_{ij} - \theta_{ij} = \sum_{\alpha \in [k_1], \beta \in [k_2]} \bar{\theta}_{\alpha\beta}(\hat{\mathbf{z}}) \mathbf{1}\{(i, j) \in \hat{\mathbf{z}}_1^{-1}(\alpha) \times \hat{\mathbf{z}}_2^{-1}(\beta)\} - \theta_{ij}$$

is a function of $\hat{\mathbf{z}}_1$ and $\hat{\mathbf{z}}_2$. Then we have

$$\left| \sum_{i,j} \frac{\tilde{\theta}_{ij} - \theta_{ij}}{\sqrt{\sum_{i,j} (\tilde{\theta}_{ij} - \theta_{ij})^2}} (\mathbf{Y}_{ij} - \theta_{ij}) \right| \leq \max_{z \in \mathcal{Z}_{k_1, k_2}} \left| \sum_{i,j} \gamma_{ij}(z) (\mathbf{Y}_{ij} - \theta_{ij}) \right|,$$

where

$$\gamma_{ij}(z) \propto \sum_{\alpha \in [k_1], \beta \in [k_2]} \bar{\theta}_{\alpha\beta}(z) \mathbf{1}\{(i, j) \in z_1^{-1}(\alpha) \times z_2^{-1}(\beta)\} - \theta_{ij}$$

satisfies $\sum_{i,j} \gamma_{ij}(z)^2 = 1$. Consider the event $\|\tilde{\theta} - \theta\|^2 \geq C_2(M^2 \vee \sigma^2)(k_1 k_2 + n_1 \log k_1 + n_2 \log k_2)/p$ for some C_2 to be specified later; we have

$$|\gamma_{ij}(z)| \leq \frac{2M}{\|\tilde{\theta} - \theta\|} \leq \sqrt{C_2(M^2 \vee \sigma^2)(k_1 k_2 + n_1 \log k_1 + n_2 \log k_2)}.$$

By Lemma 13 and union bound, we have

$$\begin{aligned} &\mathbb{P} \left(\max_{z \in \mathcal{Z}_{k_1, k_2}} \left| \sum_{i,j} \gamma_{ij}(z) (\mathbf{Y}_{ij} - \theta_{ij}) \right| > t \right) \\ &\leq \sum_{z_1 \in [k_1]^{n_1}, z_2 \in [k_2]^{n_2}} \mathbb{P} \left(\left| \sum_{i,j} \gamma_{ij}(z) (\mathbf{Y}_{ij} - \theta_{ij}) \right| > t \right) \\ &\leq \exp(-C'(k_1 k_2 + n_1 \log k_1 + n_2 \log k_2)), \end{aligned}$$

by setting $t = \sqrt{C_2(M^2 \vee \sigma^2)(k_1 k_2 + n_1 \log k_1 + n_2 \log k_2)/p}$ for some sufficiently large C_2 depending on C' . Thus, the lemma is proved. \square

Proof of Lemma 12. By definition,

$$\begin{aligned} &\left| \langle \hat{\theta} - \tilde{\theta}, \mathbf{Y} - \theta \rangle \right| \\ &= \left| \sum_{\alpha \in [k_1], \beta \in [k_2]} \left(\text{sign}(\bar{\mathbf{Y}}_{\alpha\beta}(\hat{\mathbf{z}})) (\bar{\mathbf{Y}}_{\alpha\beta}(\hat{\mathbf{z}}) \wedge \mathbf{M}) - \bar{\theta}_{\alpha\beta}(\hat{\mathbf{z}}) \right) \bar{\mathbf{W}}_{\alpha\beta}(\hat{\mathbf{z}}) |z_1^{-1}(\alpha)| |z_2^{-1}(\beta)| \right| \\ &\leq \max_{z \in \mathcal{Z}_{k_1, k_2}} \left| \sum_{\alpha \in [k_1], \beta \in [k_2]} \left(\text{sign}(\bar{\mathbf{Y}}_{\alpha\beta}(z)) (\bar{\mathbf{Y}}_{\alpha\beta}(z) \wedge \mathbf{M}) - \bar{\theta}_{\alpha\beta}(z) \right) \bar{\mathbf{W}}_{\alpha\beta}(z) |z_1^{-1}(\alpha)| |z_2^{-1}(\beta)| \right|. \end{aligned}$$

By definition, we have

$$\left(\text{sign}(\bar{\mathbf{Y}}_{\alpha\beta}(z)) (\bar{\mathbf{Y}}_{\alpha\beta}(z) \wedge \mathbf{M}) - \bar{\theta}_{\alpha\beta}(z) \right) \bar{\mathbf{W}}_{\alpha\beta}(z) \leq |\bar{\mathbf{W}}_{\alpha\beta}(z)|^2 \wedge \tau |\bar{\mathbf{W}}_{\alpha\beta}(z)|.$$

For any fixed $z_1 \in [k_1]^{n_1}$, $z_2 \in [k_2]^{n_2}$, define $n_1(a) = |z_1^{-1}(a)|$ for $a \in [k_1]$, $n_2(b) = |z_2^{-1}(b)|$ for $b \in [k_2]$ and $V_{\alpha\beta}(z) = \sqrt{n_1(a)n_2(b)} |\bar{\mathbf{W}}_{\alpha\beta}(z)| \mathbf{1}\{|\bar{\mathbf{W}}_{\alpha\beta}(z)| \leq \tau\}$, $R_{\alpha\beta}(z) = \tau n_1(a)n_2(b) |\bar{\mathbf{W}}_{\alpha\beta}(z)| \mathbf{1}\{|\bar{\mathbf{W}}_{\alpha\beta}(z)| > \tau\}$. Then

$$\left| \langle \hat{\theta} - \tilde{\theta}, \mathbf{Y} - \theta \rangle \right| \leq \max_{z \in \mathcal{Z}_{k_1, k_2}} \left\{ \sum_{\alpha \in [k_1], \beta \in [k_2]} V_{\alpha\beta}^2(z) + \sum_{\alpha \in [k_1], \beta \in [k_2]} R_{\alpha\beta}(z) \right\}.$$

Following the same argument in the proof of Lemma 10, a choice of $t = C_3(M^2 \vee \sigma^2)(k_1 k_2 + n_1 \log k_1 + n_2 \log k_2)/p$ for some sufficiently large $C_3 > 0$ will complete the proof. \square

Proof of Lemma 13. When $|\lambda| \leq p/(M \vee \sigma)$, $|\lambda\theta_i/p| \leq 1$ and $\lambda^2\sigma^2/p^2 \leq 1$. Then

$$\begin{aligned} \mathbb{E}e^{\lambda(Y_i - \theta_i)} &= p\mathbb{E}e^{\lambda(X_i/p - \theta_i)} + (1-p)\mathbb{E}e^{-\lambda\theta_i} \\ &\leq pe^{\frac{\lambda^2\sigma^2}{2p^2} + \frac{1-\lambda^2}{p}\lambda\theta_i} + (1-p)e^{-\lambda\theta_i} \\ &\leq p \left(1 + \frac{1-p}{p}\lambda\theta_i + \frac{2(1-p)^2}{p^2}\lambda^2\theta_i^2 \right) + (1-p)(1 - \lambda\theta_i + 2\lambda^2\theta_i^2) \\ &\leq 1 + \frac{2(1-p)\theta_i^2 + \sigma^2}{p}\lambda^2 + \frac{1-p}{p^2}\lambda^3\theta_i\sigma^2 + \frac{2(1-p)^2}{p^3}\lambda^4\sigma^2\theta_i^2 + 2(1-p)\lambda^2\theta_i^2 \\ &\leq 2 + (2M^2 + \sigma^2)\lambda^2/p + \lambda^3\theta_i\sigma^2/p^2 + 2\lambda^4\sigma^2\theta_i^2/p^3 \\ &\leq 2 + (2M^2 + \sigma^2)\lambda^2/p + \lambda^2\sigma^2/p + 2\lambda^2\sigma^2/p \\ &\leq 2 + (2M^2 + 4\sigma^2)\lambda^2/p \\ &\leq 2e^{(M^2 + 2\sigma^2)\lambda^2/p}. \end{aligned}$$

The second inequality is due to the fact that $e^x \leq 1 + 2x$ for all $x \geq 0$ and $e^x \leq 1 + x + 2x^2$ for all $|x| \leq 1$. Then for $|\lambda|(M \vee \sigma) \leq p$, Markov inequality implies

$$\mathbb{P} \left(\sum_{i=1}^n e_i(Y_i - \theta_i) \geq t \right) \leq 2 \exp \left\{ -\lambda t + \frac{\lambda^2}{p}(M^2 + 2\sigma^2)t \right\}.$$

By choosing $\lambda = \min \left\{ \frac{pt}{2(M^2 + 2\sigma^2)}, \frac{p}{(M \vee \sigma)\|c\|_\infty} \right\}$, we get (19). \square

Proof of Corollary 4. For independent Bernoulli random variables $X_i \sim \text{Ber}(\theta_i)$ with $\theta_i \in [0, \rho]$ for $i \in [n]$. Let $Y_i = X_i E_i/p$, where $\{E_i\}$ are independent Bernoulli random variables and $\{E_i\}$ and $\{X_i\}$ are independent. Note that $\mathbb{E}Y_i = p_i$, $\mathbb{E}Y_i^2 \leq \rho/p$ and $|Y_i| \leq 1/p$. Then Bernstein's inequality (Massart, 2007, Corollary 2.10) implies

$$\mathbb{P} \left\{ \left| \frac{1}{\sqrt{n}} \sum_{i=1}^n (Y_i - \theta_i) \right| \geq t \right\} \leq 2 \exp \left\{ -\min \left(\frac{pt^2}{4\rho}, \frac{3\sqrt{np}t}{4} \right) \right\} \quad (30)$$

for any $t > 0$. Let $S = \frac{1}{\sqrt{n}} \sum_{i=1}^n (Y_i - \theta_i)$, $T = S\mathbf{1}\{|S| \leq 3\rho\sqrt{n}\}$ and $R = 3\rho\sqrt{n}\mathbf{1}\{|S| > 3\rho\sqrt{n}\}$. Following the same arguments as in the proof of Lemma 20, we have $\mathbb{E}e^{pT/(8\rho)} \leq 5$ and $\mathbb{E}e^{pR/(8\rho)} \leq 9$. Consequently, Lemma 10, Lemma 11 and Lemma 12 hold for the Bernoulli case. Then the rest of the proof follows from the proof of Theorem 1. \square

Proof of Lemma 14. By the definitions of Y and \mathcal{Y} , we have

$$\|Y - \mathcal{Y}\|^2 \leq (\hat{p}^{-1} - p^{-1})^2 \max_{i,j} X_{ij}^2 \sum_{i,j} E_{ij}.$$

Therefore, it is sufficient to bound the three terms. For the first term, we have

$$|\hat{p}^{-1} - p^{-1}| \leq |\hat{p}^{-1} - p^{-1}| \frac{|\hat{p} - p|}{p} + \frac{|\hat{p} - p|}{p^2},$$

which leads to

$$|\hat{p}^{-1} - p^{-1}| \leq \left(1 - \frac{|\hat{p} - p|}{p} \right)^{-1} \frac{|\hat{p} - p|}{p^2}. \quad (31)$$

Bernstein's inequality implies $|\hat{p} - p|^2 \leq C \frac{p \log(n_1 + n_2)}{n_1 n_2}$ with probability at least $1 - (n_1 n_2)^{-C'}$ under the assumption that $p \gtrsim \frac{\log(n_1 + n_2)}{n_1 n_2}$. Plugging the bound into (31), we get

$$(\hat{p}^{-1} - p^{-1})^2 \leq C_1 \frac{\log(n_1 + n_2)}{p^3 n_1 n_2}.$$

The second term can be bounded by a union bound with the sub-Gaussian tail assumption of each X_{ij} . That is,

$$\max_{i,j} X_{ij}^2 \leq C_2(M^2 + \sigma^2 \log(n_1 + n_2)),$$

with probability at least $1 - (n_1 n_2)^{-C'}$. Finally, using Bernstein's inequality again, the third term is bounded as

$$\sum_{i,j} E_{ij} \leq C_3 n_1 n_2 \left(p + \sqrt{\frac{p \log(n_1 + n_2)}{n_1 n_2}} \right) \leq C'_3 n_1 n_2 p,$$

with probability at least $1 - (n_1 n_2)^{-C'}$ under the assumption that $p \gtrsim \frac{\log(n_1 + n_2)}{n_1 n_2}$. Combining the three bounds, we have obtained the desired conclusion. \square

Proof of Lemma 15. For the second and the third bounds, we use

$$\left\langle \frac{\tilde{\theta} - \theta}{\|\tilde{\theta} - \theta\|}, \mathcal{Y} - \theta \right\rangle \leq \left\langle \frac{\tilde{\theta} - \theta}{\|\tilde{\theta} - \theta\|}, Y - \theta \right\rangle + \|\mathcal{Y} - Y\|,$$

and

$$\left| \left\langle \frac{\tilde{\theta} - \theta}{\|\tilde{\theta} - \theta\|}, \mathcal{Y} - \theta \right\rangle \right| \leq \left| \left\langle \tilde{\theta} - \theta, Y - \theta \right\rangle \right| + \|\tilde{\theta} - \theta\| \|Y - \mathcal{Y}\|,$$

followed by the original proofs of Lemma 11 and Lemma 12. To prove the first bound, we introduce the notation $\hat{\theta}_{ij} = \hat{Q}_{\hat{z}_i(t)\hat{z}_j(t)}$ with $\hat{Q}_{ab} = \text{sign}(\hat{Y}_{ab}(\hat{z})) (\hat{Y}_{ab}(\hat{z}) \wedge M)$. Recall the definition of \hat{Q} in Proposition 19 with Y replaced by \mathcal{Y} . Then, we have

$$\|\hat{\theta} - \tilde{\theta}\|^2 \leq 2\|\hat{\theta} - \theta\|^2 + 2\|\hat{\theta} - \tilde{\theta}\|^2.$$

Since $\|\hat{\theta} - \tilde{\theta}\|$ can be bounded by the exact argument in the proof of Lemma 10, it is sufficient to bound $\|\hat{\theta} - \theta\|^2$. By Jensen inequality,

$$\|\hat{\theta} - \tilde{\theta}\|^2 \leq \sum_{ab} |\hat{z}^{-1}(a)| |\hat{z}^{-1}(b)| (\hat{Y}_{ab}(\hat{z}) - \tilde{\mathcal{Y}}_{ab}(\hat{z}))^2 \leq \|Y - \mathcal{Y}\|^2.$$

Thus, the proof is complete. \square

References

- E. M. Airoldi, T. B. Costa, and S. H. Chan. Stochastic blockmodel approximation of a graphon: Theory and consistent estimation. In *Advances in Neural Information Processing Systems*, pages 692–700, 2013.
- D. J. Aldous. Representations for partially exchangeable arrays of random variables. *Journal of Multivariate Analysis*, 11(4):581–598, 1981.
- C. Borgs, J. T. Chayes, H. Cohn, and S. Ganguly. Consistent nonparametric estimation for heavy-tailed sparse graphs. *arXiv preprint arXiv:1508.06675*, 2015.
- J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.
- E. J. Candès and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.
- E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.
- E. J. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *Information Theory, IEEE Transactions on*, 56(5):2053–2080, 2010.
- D. Choi. Co-clustering of nonsmooth graphons. *arXiv preprint arXiv:1507.06352*, 2015.
- D. Choi and P. J. Wolfe. Co-clustering separately exchangeable network data. *The Annals of Statistics*, 42(1):29–63, 2014.
- P. Diaconis and S. Janson. Graph limits and exchangeable random graphs. *arXiv preprint arXiv:0712.2749*, 2007.
- C. J. Flynn and P. O. Perry. Consistent biclustering. *arXiv preprint arXiv:1206.6927*, 2012.
- C. Gao, Y. Lu, and H. H. Zhou. Rate-optimal graphon estimation. *The Annals of Statistics*, 43(6):2624–2652, 2015a.
- C. Gao, A. W. van der Vaart, and H. H. Zhou. A general framework for bays structured linear models. *arXiv preprint arXiv:1506.02174*, 2015b.
- A. Guntuboyina. Lower bounds for the minimax risk using f -divergences, and applications. *Information Theory, IEEE Transactions on*, 57(4):2386–2399, 2011.
- J. A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67(337):123–129, 1972.
- P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.
- D. N. Hoover. Relations on probability spaces and arrays of random variables. *Preprint, Institute for Advanced Study, Princeton, NJ*, 2, 1979.
- O. Kallenberg. On the representation theorem for exchangeable arrays. *Journal of Multivariate Analysis*, 30(1):137–154, 1989.
- R. Keshavan, A. Montanari, and S. Oh. Matrix completion from noisy entries. In *Advances in Neural Information Processing Systems*, pages 952–960, 2009.
- R. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. *Information Theory, IEEE Transactions on*, 56(6):2980–2998, 2010.
- O. Klopp, A. B. Tsybakov, and N. Verzelen. Oracle inequalities for network models and sparse graphon estimation. *arXiv preprint arXiv:1507.04118*, 2015.
- V. Koltchinskii, K. Lounici, and A. B. Tsybakov. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *The Annals of Statistics*, 39(5):2302–2329, 2011.
- M. Lee, H. Shen, J. Z. Huang, and J. S. Marron. Biclustering via sparse singular value decomposition. *Biometrics*, 66(4):1087–1095, 2010.
- L. Lovász. *Large Networks and Graph Limits*, volume 60. American Mathematical Society, 2012.
- L. Lovász and B. Szegedy. Limits of dense graph sequences. *Journal of Combinatorial Theory, Series B*, 96(6):933–957, 2006.
- Y. Lu and H. H. Zhou. Minimax rates for estimating matrix products. *Preprint, Yale University*, 2015.
- Z. Ma and Y. Wu. Volume ratio, sparsity, and minimaxity under unitarily invariant norms. *Information Theory, IEEE Transactions on*, 61(12):6939–6956, 2015.
- P. Massart. *Concentration Inequalities and Model Selection*, volume 1896. Springer, 2007.
- S. C. Olhede and P. J. Wolfe. Network histograms and universality of blockmodel approximation. *Proceedings of the National Academy of Sciences*, 111(41):14722–14727, 2014.
- B. Recht. A simpler approach to matrix completion. *The Journal of Machine Learning Research*, 12:3413–3430, 2011.
- B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.
- K. Rohe, T. Qin, and B. Yu. Co-clustering for directed graphs: the stochastic co-blockmodel and spectral algorithm dis-sim. *arXiv preprint arXiv:1204.2296*, 2012.
- D. B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.
- S. Wold. Cross-validatory estimation of the number of components in factor and principal components models. *Technometrics*, 20(4):397–405, 1978.
- P. J. Wolfe and S. C. Olhede. Nonparametric graphon estimation. *arXiv preprint arXiv:1309.5936*, 2013.

B. Yu. Assouad, Fano, and Le Cam. In *Festschrift for Lucien Le Cam*, pages 423–435. Springer, 1997.

The Teaching Dimension of Linear Learners

Ji Liu

Department of Computer Science
University of Rochester
Rochester, NY 14627, USA

JL.LIU.UWISC@GMAIL.COM

Xiaojin Zhu

Department of Computer Sciences
University of Wisconsin-Madison
Madison, WI 53706, USA

JERRYZHU@CS.WISC.EDU

Editor: Sanjoy Dasgupta

Abstract

Teaching dimension is a learning theoretic quantity that specifies the minimum training set size to teach a target model to a learner. Previous studies on teaching dimension focused on version-space learners which maintain all hypotheses consistent with the training data, and cannot be applied to modern machine learners which select a specific hypothesis via optimization. This paper presents the first known teaching dimension for ridge regression, support vector machines, and logistic regression. We also exhibit optimal training sets that match these teaching dimensions. Our approach generalizes to other linear learners.

Keywords: Optimization based learner, Karush-Kuhn-Tucker conditions, VC-dimension

1. Introduction

Consider a teacher who knows both a target model and the learning algorithm used by a machine learner. The teacher wants to teach the target model to the learner by *constructing* a training set. The training set does not need to contain independent and identically distributed items drawn from some distribution. Furthermore, the teacher can construct any item in the input space. How many training items are needed? This is the question addressed by the *teaching dimension* (Goldman and Kearns, 1995; Shinohara and Miyano, 1991). We give the precise definition in section 2, but first illustrate the intuition with an example.

Consider integers $x \in \{1 \dots 10\}$ and threshold classifiers h_θ on them, so that $h_\theta(x)$ returns -1 if $x < \theta$ and 1 if $x \geq \theta$. Now let the hypothesis space \mathcal{H} consist of eleven classifiers $\mathcal{H} = \{h_\theta \mid \theta \in \{1 \dots 11\}\}$. Let the learner be a version-space learner, namely it maintains a version space $\{h_\theta \in \mathcal{H} \mid h_\theta \text{ consistent with the training set}\}$. Equivalently, the learner is a 0-1 loss empirical risk minimizer (ERM) which finds all models with zero training error. If we want to teach a target model (in this paper we use hypothesis and model interchangeably), say h_θ , to such a learner, we can construct a training set that results in a singleton version space $\{h_\theta\}$. It is easy to see that the training set $D = \{(x_1 = 8, y_1 = -1), (x_2 = 9, y_2 = 1)\}$ is the smallest set for this purpose. We say that the teaching dimension of h_θ with respect

to \mathcal{H} is $TD(h_\theta) = |D| = 2$. Similarly, $TD(h_{11}) = 1$ because $D = \{(x_1 = 10, y_1 = -1)\}$ suffices. In fact, $TD(h_\theta^*) = 1$ for target model $\theta^* = 1$ or 11, and 2 for $\theta^* = 2, 3, \dots, 10$.

The astute reader may notice that this example does not apply to continuous spaces. To see this, let us extend $x \in \mathbb{R}$ and $\mathcal{H} = \{h_\theta \mid \theta \in \mathbb{R}\}$. The learner's version space under any linearly separable training set would now be represented by the interval between the two closest oppositely labeled items. It is impossible for the version-space learner to pick out a unique target model h_{θ^*} with a finite training set. In other words, $TD(h_{\theta^*}) = \infty$ for all target models θ^* . This is counterintuitive because ostensibly we can teach any one of the "modern" machine learning algorithms such as a support vector machine (SVM) with only two training items: $D = \{(x_1 = \theta^* - \epsilon, y_1 = -1), (x_2 = \theta^* + \epsilon, y_2 = 1)\}$ with any $\epsilon > 0$.

The issue here is that a version-space learner is not equipped with the ability to pick the max-margin (or any other specific) hypothesis from the version space. In contrast, an SVM is *not* a version-space learner in our terminology; we have stronger knowledge from optimization on how it picks a specific hypothesis from the hypothesis space. This paper will utilize such knowledge to derive teaching dimensions that are distinct from classic teaching dimension analysis (e.g. Doliwa et al. (2014)). Specifically, we extend teaching dimension to linear learners that learn by regularized surrogate-loss empirical risk minimization:

$$\mathcal{A}_{\text{opt}}(D) := \underset{\theta \in \mathbb{R}^d}{\text{Argmin}} \underbrace{\sum_{i=1}^n \ell(\mathbf{x}_i^\top \theta; y_i) + \frac{\lambda}{2} \|\theta\|_A^2}_{=: f(\theta)}. \quad (1)$$

Here, we identify \mathcal{H} with \mathbb{R}^d , h with θ , the surrogate loss function ℓ is either smooth or convex in the first argument, $\lambda > 0$ is the regularization coefficient, and A is a positive semidefinite matrix. $\|\cdot\|_A$ is the Mahalanobis norm: $\|\theta\|_A := \sqrt{\theta^\top A \theta}$. This covers both homogeneous (e.g. $A = I$) and inhomogeneous (e.g. $A = [I, 0; 0, I]$) learners. We follow the convention in optimization when we use the capitalized Argmin to emphasize that it returns a *set* that achieves the minimum. The teacher can construct a training set with any items in \mathbb{R}^d . The alternative pool-based teaching setting, where the teacher is given a finite pool of candidate training items and must select items from that pool, is not studied in this paper. By linear learners we mean the input \mathbf{x} and the parameter θ interact only via their inner product $\mathbf{x}^\top \theta$. Linear learners include SVMs, logistic regression, and linear regression. Our analysis technique involves a novel application of the Karush-Kuhn-Tucker (KKT) conditions.

	homogeneous			inhomogeneous		
	ridge	SVM	logistic	ridge	SVM	logistic
exact parameter	1	$\lceil \lambda \ \theta^*\ ^2 \rceil$	$\lceil \frac{\lambda \ \theta^*\ ^2}{2\tau_{\max}} \rceil$	2	$2 \lceil \frac{\lambda \ \mathbf{w}^*\ ^2}{2} \rceil$	$2 \lceil \frac{\lambda \ \mathbf{w}^*\ ^2}{2\tau_{\max}} \rceil$
decision boundary	-	1	1	-	2	2

Table 1: The teaching dimension of ridge regression, SVM, and logistic regression. ($\lceil \cdot \rceil$: up to rounding effect, see section 3.3).

To our knowledge, this paper gives the first known values of teaching dimension for ridge regression, SVM, and logistic regression. We summarize our main results in Table 1. The table separately lists homogeneous (without a bias term) and inhomogeneous (with a bias term) versions of the linear learners. The teaching goal refers to the intention of the teacher: is teaching considered successful only when the learner learns the exact target parameter, or when the learner learns the correct decision boundary (which can be achieved by any positive scaling of the target parameter)? See section 3 for definition of the target parameters θ^* , w^* and the constant τ_{\max} . The target parameters are assumed to be nonzero. We will also present the corresponding minimum teaching set construction in section 3.

2. Classic Teaching Dimension and its Limitations

Let \mathcal{X} denote the input space and $\mathcal{Y} \subseteq \mathbb{R}$ the output space. A hypothesis is a function $h : \mathcal{X} \rightarrow \mathcal{Y}$. In this section we identify a hypothesis h_θ with its model parameter θ . The hypothesis space \mathcal{H} is a set of hypotheses. By training item we mean a pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$. A training set is a multiset $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$ where repeated items are allowed. Importantly, for the purpose of teaching we do *not* assume that D be drawn *i.i.d.* from a distribution. Let $\mathbb{D} = \cup_{n=1}^{\infty} (\mathcal{X} \times \mathcal{Y})^n$ denote the set of all training sets of all sizes. A learning algorithm $\mathcal{A} : \mathbb{D} \rightarrow 2^{\mathcal{H}}$ takes in a training set $D \in \mathbb{D}$ and outputs a subset of the hypothesis space \mathcal{H} . That is, \mathcal{A} does not necessarily return a unique hypothesis.

Classic teaching dimension analysis is restricted to the version-space learner \mathcal{A}_{vs} :

$$\mathcal{A}_{vs}(D) = \{h \in \mathcal{H} \mid \forall (x, y) \in D, h(x) = y\}. \quad (2)$$

That is, the learner \mathcal{A}_{vs} keeps track of the version space consisting of all hypotheses h that are consistent with D . Let the target model be $h_{\theta^*} \in \mathcal{H}$. Teaching is successful if the teacher identifies a training set $D \in \mathbb{D}$ such that $\mathcal{A}_{vs}(D) = \{h_{\theta^*}\}$ the singleton set. Such a D is called a **teaching set** of h_{θ^*} with respect to \mathcal{H} . The teaching dimension of the hypothesis h_{θ^*} is the minimum size of the teaching set:

$$TD(h_{\theta^*}) = \begin{cases} \min_{D \in \mathbb{D}} |D|, & \text{for } D \text{ a teaching set of } h_{\theta^*} \\ \infty, & \text{if no teaching set exists} \end{cases}$$

Furthermore, the teaching dimension of the whole hypothesis space \mathcal{H} is defined by the hardest hypothesis: $TD(\mathcal{H}) = \max_{h \in \mathcal{H}} TD(h)$. In this paper we will focus on the fine-grained teaching dimension of individual hypothesis $TD(h)$.

Classic teaching dimension analysis has several limitations: the learner is assumed to be a version-space learner \mathcal{A}_{vs} , and the hypothesis space is typically finite or countably infinite. As the example in section 1 showed, these fail to capture the teaching dimension of “modern” machine learners which has \mathbb{R}^d as input space and picks a unique hypothesis via regularized empirical risk minimization (1). Furthermore, the target model can be ambiguous when the learner is a classifier: should the learner learn the exact target parameter θ^* , or the target decision boundary? In linear models any scaled parameter $c\theta^*$ with $c > 0$ produces the same target decision boundary. These limitations motivate us to generalize the teaching dimension in the next section.

3. Main Results

To make our teaching dimension’s dependency on the learning algorithm explicit, henceforth we write teaching dimension with two arguments as

$$TD(h^*, \mathcal{A})$$

where $h^* \in \mathcal{H}$ is the target model, and $\mathcal{A} : \mathbb{D} \rightarrow 2^{\mathcal{H}}$ is the learning algorithm which given a training set $D \in \mathbb{D}$ returns a set of hypotheses $\mathcal{A}(D)$. We define teaching dimension to be the size of the smallest training set D such that $\mathcal{A}(D) = \{h^*\}$, the singleton set containing the target model. With this notation, the classic teaching dimension is $TD(h^*, \mathcal{A}_{vs})$ where \mathcal{A}_{vs} is the version space learning algorithm (2). In this paper we focus on \mathcal{A}_{opt} in (1) instead, namely linear learners in \mathbb{R}^d . Linear learners include many popular members such as both homogeneous and inhomogeneous versions of linear regression, SVM, and logistic regression. In addition, the linear interaction between \mathbf{x} and θ makes the loss function subgradient easy to compute, though in principle our analysis technique is applicable to other optimization-based learners, too. In this section our goal is to teach the exact parameter θ^* , consequently our teaching dimension of interest is

$$TD(\theta^*, \mathcal{A}_{opt}).$$

Later in section 4 for classification we will teach the decision boundary instead.

How to reason about our teaching dimension $TD(\theta^*, \mathcal{A}_{opt})$? It is the size of the *smallest* training set D with which (1) has a unique solution θ^* . Our strategy is to first establish a number of lower bounds $LB \leq TD(\theta^*, \mathcal{A}_{opt})$ by showing that any training set with which (1) has a unique solution θ^* must have at least LB items. Section 3.1 is devoted to such lower bounds. The actual teaching dimension is learner dependent. In sections 3.2 and 3.3 we construct specific teaching sets for three popular learners: ridge regression, SVM, and logistic regression. These teaching sets uniquely returns θ^* via (1). By definition, the size of these teaching sets is an upper bound on $TD(\theta^*, \mathcal{A}_{opt})$, respectively. If the lower and upper bounds match, we would have identified the teaching dimension $TD(\theta^*, \mathcal{A}_{opt})$.

3.1 Lower Bounds on Teaching Dimension $TD(\theta^*, \mathcal{A}_{opt})$

In this section we provide three general lower bounds on the teaching dimension. These lower bounds capture different aspects of a teaching set, and should be used in conjunction (i.e. taking the maximum) when applicable. We will instantiate these lower bounds for specific learners in section 3.2. In the following let \mathcal{X} and \mathcal{Y} be the feasible region of all \mathbf{x}_i ’s and y_i ’s respectively. We will use the notation $\partial_t \ell(\cdot, \cdot)$ in the following way: if $\ell(\cdot, \cdot)$ is smooth, then it denotes a singleton set only containing the gradient w.r.t. the first argument; if $\ell(\cdot, \cdot)$ is convex, then it denotes the subdifferential w.r.t the first argument.

LBI comes from a degree-of-freedom perspective. It is necessary to have this amount of training items for a unique solution to exist in (1).

Theorem 1 *Given any target model θ^* , there is a degree-of-freedom lower bound on the number of training items to obtain a unique solution θ^* from solving (1):*

$$LBI = \begin{cases} d - \text{Rank}(A) + 1, & \text{if } A\theta^* \neq \mathbf{0} \\ d - \text{Rank}(A), & \text{otherwise.} \end{cases} \quad (3)$$

Proof Let n^* be the minimal number of training items to ensure a unique solution $\boldsymbol{\theta}^*$. First consider the case $n^* = 0$. It happens if and only if $\boldsymbol{\theta}^* = \mathbf{0}$ and $\text{Rank}(A) = d$, which is a special case of $A\boldsymbol{\theta}^* = \mathbf{0}$. Clearly, this case is consistent with LB1. Next consider the case $n^* \geq 1$. Since $\boldsymbol{\theta}^*$ solves (1), the KKT condition holds:

$$-\lambda A\boldsymbol{\theta}^* \in \sum_{i=1}^{n^*} \partial_1 \ell(\mathbf{x}_i^\top \boldsymbol{\theta}^*, y_i) \mathbf{x}_i. \quad (4)$$

We seek all $\boldsymbol{\delta}$ such that $\boldsymbol{\theta}^* + \boldsymbol{\delta}$ satisfies

$$A(\boldsymbol{\theta}^* + \boldsymbol{\delta}) = A\boldsymbol{\theta}^* \quad \text{and} \quad \mathbf{x}_i^\top (\boldsymbol{\theta}^* + \boldsymbol{\delta}) = \mathbf{x}_i^\top \boldsymbol{\theta}^* \quad \forall i = 1, \dots, n^*, \quad (5)$$

For any such $\boldsymbol{\delta}$, simple algebra verifies that $\boldsymbol{\theta}^* + t\boldsymbol{\delta}$ satisfies the KKT condition (4) for any $t \in [0, 1]$. Consequently, $\boldsymbol{\theta}^* + \boldsymbol{\delta}$ also solves the problem in (1). To see this, we consider two situations:

- If the loss function $\ell(\cdot, \cdot)$ is convex in the first argument, the KKT condition is a sufficient optimality condition, which means that $\boldsymbol{\theta}^* + \boldsymbol{\delta}$ solves (1).
- If the loss function $\ell(\cdot, \cdot)$ is smooth (not necessary convex) in the first argument, we have $f(\boldsymbol{\theta}^*) = f(\boldsymbol{\theta}^* + \boldsymbol{\delta})$ by using the Taylor expansion (recall f is defined in equation 1):

$$\begin{aligned} f(\boldsymbol{\theta}^* + \boldsymbol{\delta}) &= f(\boldsymbol{\theta}^*) + \langle \nabla f(\boldsymbol{\theta}^* + t\boldsymbol{\delta}), \boldsymbol{\delta} \rangle \quad (\text{for some } t \in [0, 1]) \\ &= f(\boldsymbol{\theta}^*) + \left\langle \sum_{i=1}^{n^*} \nabla_1 \ell(\mathbf{x}_i^\top (\boldsymbol{\theta}^* + t\boldsymbol{\delta}), y_i) \mathbf{x}_i + \lambda A(\boldsymbol{\theta}^* + t\boldsymbol{\delta}), \boldsymbol{\delta} \right\rangle \\ &= f(\boldsymbol{\theta}^*) + \underbrace{\left\langle \sum_{i=1}^{n^*} \nabla_1 \ell(\mathbf{x}_i^\top \boldsymbol{\theta}^*, y_i) \mathbf{x}_i + \lambda A\boldsymbol{\theta}^*, \boldsymbol{\delta} \right\rangle}_{=0 \text{ due to the KKT condition (4)}} \\ &= f(\boldsymbol{\theta}^*). \end{aligned}$$

Therefore, $\boldsymbol{\theta}^* + \boldsymbol{\delta}$ also solves (1). However, the uniqueness of $\boldsymbol{\theta}^*$ requires $\boldsymbol{\delta} = \mathbf{0}$ to be the only value satisfying (5). This is equivalent to say

$$\text{Null}(A) \cap \text{Null}(\text{Span}\{\mathbf{x}_1, \dots, \mathbf{x}_{n^*}\}) = \{\mathbf{0}\}. \quad (6)$$

It indicates that

$$\text{Rank}(A) + \text{Dim}(\text{Span}\{\mathbf{x}_1, \dots, \mathbf{x}_{n^*}\}) \geq d.$$

From $n^* \geq \text{Dim}(\text{span}\{\mathbf{x}_1, \dots, \mathbf{x}_{n^*}\})$, we have $n^* \geq d - \text{Rank}(A)$. We proved the general case for LB1.

If we have $A\boldsymbol{\theta}^* \neq \mathbf{0}$, we can further improve LB1. Let $\mathbf{g}^* = (g_1^*, \dots, g_{n^*}^*)^\top$ be the vector satisfying

$$-\lambda A\boldsymbol{\theta}^* = \sum_{i=1}^{n^*} g_i^* \mathbf{x}_i \quad \text{and} \quad g_i^* \in \partial_1 \ell(\mathbf{x}_i^\top \boldsymbol{\theta}^*, y_i) \quad \forall i = 1, 2, \dots, n^*. \quad (7)$$

Since $\boldsymbol{\theta}^*$ satisfies the KKT condition, such vector \mathbf{g}^* must exist. Applying $A\boldsymbol{\theta}^* \neq \mathbf{0}$ to (7), we have $\mathbf{g}^* \neq \mathbf{0}$ and

$$\text{Dim}(\text{Span}\{A_{1,1}, A_{2,1}, \dots, A_{d,1}\} \cap \text{Span}\{\mathbf{x}_1, \dots, \mathbf{x}_{n^*}\}) \geq 1. \quad (8)$$

To satisfy (6), we must have

$$d = \text{Dim}(\text{Span}\{A_{1,1}, A_{2,1}, \dots, A_{d,1}, \mathbf{x}_1, \dots, \mathbf{x}_{n^*}\}).$$

Using the fact in linear algebra

$$\begin{aligned} & \text{Dim}(\text{Span}\{A_{1,1}, A_{2,1}, \dots, A_{d,1}, \mathbf{x}_1, \dots, \mathbf{x}_{n^*}\}) \\ &= \underbrace{\text{Dim}(\text{Span}\{A_{1,1}, A_{2,1}, \dots, A_{d,1}\})}_{=\text{Rank}(A)} + \\ & \underbrace{\text{Dim}(\text{Span}\{\mathbf{x}_1, \dots, \mathbf{x}_{n^*}\})}_{\leq n^*} - \\ & \underbrace{\text{Dim}(\text{Span}\{A_{1,1}, A_{2,1}, \dots, A_{d,1}\} \cap \text{Span}\{\mathbf{x}_1, \dots, \mathbf{x}_{n^*}\})}_{\geq 1 \text{ (from (8))}} \end{aligned}$$

We conclude that $n^* \geq d - \text{Rank}(A) + 1$. We completed the proof for LB1. ■

LB2 observes that the regularizer acts as a prior. If λ is large, more items are needed to sway the prior toward the target $\boldsymbol{\theta}^*$.

Theorem 2 Given any target model $\boldsymbol{\theta}^*$, there is a strength-of-regularization lower bound on the required number of training items to obtain a unique solution $\boldsymbol{\theta}^*$ from solving (1):

$$LB2 = \begin{cases} \lambda \left(\sup_{\alpha \in \mathbb{R}, y \in \mathcal{Y}} -\partial_1 \ell(\alpha \|\boldsymbol{\theta}^*\|_A^2, y) \right)^{-1}, & \text{if } A \text{ has full rank and } \boldsymbol{\theta}^* \neq \mathbf{0} \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

Proof When A has full rank we have an equivalent expression for the KKT condition (4):

$$-\lambda A^{\frac{1}{2}} \boldsymbol{\theta}^* \in \sum_{i=1}^{n^*} A^{-\frac{1}{2}} \mathbf{x}_i \partial_1 \ell(\mathbf{x}_i^\top \boldsymbol{\theta}^*, y_i) \quad \forall i = 1, \dots, n^*. \quad (10)$$

Let us decompose $A^{-\frac{1}{2}} \mathbf{x}_i$ for all $i = 1, \dots, n^*$ into $A^{-\frac{1}{2}} \mathbf{x}_i = \alpha_i A^{\frac{1}{2}} \boldsymbol{\theta}^* + \mathbf{u}_i$, where \mathbf{u}_i is orthogonal to $A^{\frac{1}{2}} \boldsymbol{\theta}^*$: $\mathbf{u}_i^\top A^{\frac{1}{2}} \boldsymbol{\theta}^* = 0$. Equivalently $\mathbf{x}_i = \alpha_i A \boldsymbol{\theta}^* + A^{\frac{1}{2}} \mathbf{u}_i$. Applying this decomposition, we have

$$\mathbf{x}_i^\top \boldsymbol{\theta}^* = \alpha_i \|\boldsymbol{\theta}^*\|_A^2 + \mathbf{u}_i^\top A^{\frac{1}{2}} \boldsymbol{\theta}^* = \alpha_i \|\boldsymbol{\theta}^*\|_A^2.$$

Putting it back in (10) we obtain

$$-\lambda A^{\frac{1}{2}} \boldsymbol{\theta}^* \in \sum_{i=1}^{n^*} (\alpha_i A^{\frac{1}{2}} \boldsymbol{\theta}^* + \mathbf{u}_i) \partial_1 \ell(\alpha_i \|\boldsymbol{\theta}^*\|_A^2, y_i) \quad \forall i = 1, \dots, n^*. \quad (11)$$

Since \mathbf{u}_i is orthogonal to $A^{\frac{1}{2}}\boldsymbol{\theta}^*$, (11) can be rewritten as

$$\exists \alpha_i \in \mathbb{R}, \exists y_i \in \mathcal{Y}; \exists g_i \in \partial_1 \ell(\alpha_i \|\boldsymbol{\theta}^*\|_A, y_i) \quad \forall i = 1, \dots, n^*$$

satisfying
$$\sum_{i=1}^{n^*} g_i \mathbf{u}_i = 0$$

$$-\lambda A^{\frac{1}{2}}\boldsymbol{\theta}^* = A^{\frac{1}{2}}\boldsymbol{\theta}^* \sum_{i=1}^{n^*} \alpha_i g_i \quad (12)$$

Since $A\boldsymbol{\theta}^* \neq 0$, we have $A^{\frac{1}{2}}\boldsymbol{\theta}^* \neq 0$ and (12) is equivalent to $-\lambda = \sum_{i=1}^{n^*} \alpha_i g_i$. It follows that

$$\lambda = -\sum_{i=1}^{n^*} \alpha_i g_i \leq n^* \sup_{\alpha \in \mathbb{R}, y \in \mathcal{Y}, g \in \partial_1 \ell(\alpha \|\boldsymbol{\theta}^*\|_A, y)} -\alpha g = n^* \sup_{\alpha \in \mathbb{R}, y \in \mathcal{Y}, g \in \partial_1 \ell(\alpha \|\boldsymbol{\theta}^*\|_A, y)} \alpha g$$

It indicates the lower bound for n^*

$$n^* \geq \left\lfloor \frac{\lambda}{\sup_{\alpha \in \mathbb{R}, y \in \mathcal{Y}, g \in \partial_1 \ell(\alpha \|\boldsymbol{\theta}^*\|_A, y)} \alpha g} \right\rfloor. \quad \blacksquare$$

LB1 and LB2 apply to all generalized linear learners. Due to the popularity of inhomogeneous margin-based linear learners (which include the standard form of SVM and logistic regression), we provide a tighter lower bound LB3 for such learners in Theorem 3. For inhomogeneous margin-based linear learners the learning algorithm A_{opt} solves a special form of (1):

$$A_{opt}(\mathcal{D}) = \text{Argmin}_{\mathbf{w}, b} \sum_{i=1}^n \ell(y_i(\mathbf{x}_i^\top \mathbf{w} + b)) + \frac{\lambda}{2} \|\mathbf{w}\|_A^2. \quad (13)$$

LB3 will prove to be instrumental in computing the teaching dimension for those learners. Following standard notation, we define $\boldsymbol{\theta} = [\mathbf{w}; b]$ where $\mathbf{w} \in \mathbb{R}^d$ is the weight vector and $b \in \mathbb{R}$ the bias (offset) term. Note $\boldsymbol{\theta} \in \mathbb{R}^{d+1}$ now. The $d \times d$ regularization matrix A applies only to \mathbf{w} while b is not regularized. Furthermore, margin-based linear learners have loss functions defined on the margin $y(\mathbf{x}^\top \mathbf{w} + b)$. This loss function structure will play a key role in obtaining LB3.

Theorem 3 *Assume matrix A in (13) has full rank and $\mathbf{w}^* \neq \mathbf{0}$. Given any target model $[\mathbf{w}^*; b^*]$, there is an inhomogeneous-margin lower bound on the required number of training items to obtain a unique solution $[\mathbf{w}^*; b^*]$ from solving (13):*

$$LB3 = \left\lfloor \lambda \left(\sup_{\alpha \in \mathbb{R}, g \in -\partial \ell(\alpha \|\mathbf{w}^*\|_A^2)} \alpha g \right)^{-1} \right\rfloor. \quad (14)$$

Proof Let $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1, \dots, n}$ be a teaching set for $[\mathbf{w}^*; b^*]$. The following KKT condition needs to be satisfied:

$$\mathbf{0} \in \sum_{i=1}^n \partial \ell(y_i(\mathbf{x}_i^\top \mathbf{w}^* + b^*)) y_i \begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix} + \begin{bmatrix} \lambda A \mathbf{w}^* \\ 0 \end{bmatrix}. \quad (15)$$

If we construct a new training set

$$\tilde{\mathcal{D}} = \left\{ \tilde{\mathbf{x}}_i = \mathbf{x}_i + \frac{b^*}{\|\mathbf{w}^*\|_A^2} A \mathbf{w}^*, \tilde{y}_i = y_i \right\}_{i=1, \dots, n}$$

then $[\mathbf{w}^*; 0]$ satisfies the KKT condition defined on $\tilde{\mathcal{D}}$. This can be verified as follows:

$$\begin{aligned} & \sum_{i=1}^n \partial \ell(y_i(\tilde{\mathbf{x}}_i^\top \mathbf{w}^*)) y_i \begin{bmatrix} \tilde{\mathbf{x}}_i \\ 1 \end{bmatrix} + \begin{bmatrix} \lambda A \mathbf{w}^* \\ 0 \end{bmatrix} \\ &= \sum_{i=1}^n \partial \ell(y_i(\mathbf{x}_i^\top \mathbf{w}^* + b^*)) y_i \begin{bmatrix} \mathbf{x}_i + \frac{b^*}{\|\mathbf{w}^*\|_A^2} A \mathbf{w}^* \\ 1 \end{bmatrix} + \begin{bmatrix} \lambda A \mathbf{w}^* \\ 0 \end{bmatrix} \\ &= \underbrace{\sum_{i=1}^n \partial \ell(y_i(\mathbf{x}_i^\top \mathbf{w}^* + b^*)) y_i \begin{bmatrix} \mathbf{x}_i \\ 1 \end{bmatrix} + \begin{bmatrix} \lambda A \mathbf{w}^* \\ 0 \end{bmatrix}}_{\geq 0 \text{ from (15)}} + \underbrace{\sum_{i=1}^n \partial \ell(y_i(\mathbf{x}_i^\top \mathbf{w}^* + b^*)) y_i \begin{bmatrix} \frac{b^*}{\|\mathbf{w}^*\|_A^2} A \mathbf{w}^* \\ 0 \end{bmatrix}}_{\geq 0 \text{ from (15)}} \\ & \geq 0 \end{aligned}$$

where $0 \in \sum_{i=1}^n \partial \ell(y_i(\mathbf{x}_i^\top \mathbf{w}^* + b^*)) y_i$ is from the bias dimension in (15). It follows that

$$\mathbf{0} \in \sum_{i=1}^n \partial \ell(\tilde{y}_i \tilde{\mathbf{x}}_i^\top \mathbf{w}^*) \tilde{y}_i \tilde{\mathbf{x}}_i + \lambda A \mathbf{w}^*$$

which is equivalent to

$$\begin{aligned} \mathbf{0} & \in \sum_{i=1}^n \partial \ell(\tilde{y}_i \tilde{\mathbf{x}}_i^\top \mathbf{w}^*) A^{-\frac{1}{2}} \underbrace{\tilde{y}_i \tilde{\mathbf{x}}_i}_{=: \mathbf{z}_i} + \lambda A^{\frac{1}{2}} \mathbf{w}^* \\ &= \sum_{i=1}^n \partial \ell(\mathbf{z}_i^\top \mathbf{w}^*) A^{-\frac{1}{2}} \mathbf{z}_i + \lambda A^{\frac{1}{2}} \mathbf{w}^*. \end{aligned} \quad (16)$$

We decompose $A^{-\frac{1}{2}} \mathbf{z}_i = \alpha_i A^{\frac{1}{2}} \mathbf{w}^* + \mathbf{u}_i$ where \mathbf{u}_i satisfies $\mathbf{u}_i^\top A^{\frac{1}{2}} \mathbf{w}^* = 0$. Applying this decomposition to (16), we have

$$\lambda A^{\frac{1}{2}} \mathbf{w}^* \in \sum_{i=1}^n -\partial \ell(\alpha_i \|\mathbf{w}^*\|_A^2) (\alpha_i A^{\frac{1}{2}} \mathbf{w}^* + \mathbf{u}_i). \quad (17)$$

Since \mathbf{u}_i is orthogonal to $A^{\frac{1}{2}} \mathbf{w}^*$, (17) implies that

$$\lambda A^{\frac{1}{2}} \mathbf{w}^* \in \sum_{i=1}^n -\partial \ell(\alpha_i \|\mathbf{w}^*\|_A^2) \alpha_i A^{\frac{1}{2}} \mathbf{w}^*.$$

Since $\mathbf{w}^* \neq \mathbf{0}$ we have

$$\lambda \in \sum_{i=1}^n -\partial \ell(\alpha_i \|\mathbf{w}^*\|_A^2) \alpha_i.$$

Together with

$$\sum_{i=1}^n -\partial \ell(\alpha_i \|\mathbf{w}^*\|_A^2) \alpha_i \leq n \sup_{\alpha \in \mathbb{R}, g \in -\partial \ell(\alpha \|\mathbf{w}^*\|_A^2)} \alpha g,$$

we obtain LB3. \blacksquare

3.2 The Teaching Dimension $TD(\boldsymbol{\theta}^*, \mathcal{A}_{opt})$ of Three Homogeneous Learners

We now turn to upper bounding teaching dimension by constructing teaching sets. To prove that we indeed have a teaching set for a target $\boldsymbol{\theta}^*$, we need to show that $\boldsymbol{\theta}^*$ is a solution of (1), and the solution is unique. The size of any such teaching set is an upper bound on the teaching dimension. The teaching dimension itself is determined if such an upper bound matches the corresponding lower bound. We show that this is indeed the case for our constructed teaching sets. For the sake of reference we preview in Table 2 the instantiated lower bounds that we will use in this section; their derivation will be shown below.

lower bound	homogeneous			inhomogeneous		
	ridge	SVM	logistic	ridge	SVM	logistic
LB1	1	1	1	2	2	2
LB2	0	$\lceil \lambda \ \boldsymbol{\theta}^*\ ^2 \rceil$	$\frac{\lambda \ \boldsymbol{\theta}^*\ ^2}{\tau_{\max}}$	0	0	0
LB3	-	-	-	-	$\lceil \lambda \ \mathbf{w}^*\ ^2 \rceil$	$\frac{\lambda \ \mathbf{w}^*\ ^2}{\tau_{\max}}$

Table 2: Lower bounds of teaching dimension $TD(\boldsymbol{\theta}^*, \mathcal{A}_{opt})$ for homogeneous and inhomogeneous versions of ridge regression, SVM, and logistic regression.

Teaching dimension is learner-dependent. We choose three learners to study their teaching dimension due to these learners' popularity in machine learning: ridge regression, SVM, and logistic regression. It turns out that homogeneous and inhomogeneous versions of these learners require different analysis. We devote this section to the homogeneous version where the regularizer matrix $A = I$ the identity matrix, and the next section to the inhomogeneous version. It is possible to extend our analysis to other linear learners of the form (1).

It is easy to see that if the target model $\boldsymbol{\theta}^* = \mathbf{0}$, we do not need any training data to uniquely obtain the target model from (1). In the following, we only consider the nontrivial case $\boldsymbol{\theta}^* \neq \mathbf{0}$.

Homogeneous ridge regression solves the following optimization problem:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \sum_{i=1}^n \frac{1}{2} (\mathbf{x}_i^\top \boldsymbol{\theta} - y_i)^2 + \frac{\lambda}{2} \|\boldsymbol{\theta}\|^2. \quad (18)$$

We only need one training item to uniquely obtain any nonzero target model $\boldsymbol{\theta}^*$, as the following construction shows.

Proposition 1 *Given any target model $\boldsymbol{\theta}^* \neq \mathbf{0}$, the following is a teaching set for homogeneous ridge regression (18):*

$$\mathbf{x}_1 = a\boldsymbol{\theta}^*, \quad y_1 = \frac{\lambda + \|\mathbf{x}_1\|^2}{a} \quad (19)$$

where a can be any nonzero real number.

Proof We simply verify the KKT condition to see that $\boldsymbol{\theta}^*$ is a solution to (18) by applying the construction in (19). The uniqueness of $\boldsymbol{\theta}^*$ is guaranteed by the strong convexity of (18). \blacksquare

It is worth to note that the teaching set is inconsistent with the target model, that is, $\mathbf{x}_1^\top \boldsymbol{\theta}^* = a\|\boldsymbol{\theta}^*\|^2 \neq y_1 = \frac{\lambda}{a} + a\|\boldsymbol{\theta}^*\|^2$, unless the regularization is absent $\lambda = 0$. The teacher intentionally overshoots the target in order to precisely counter the learner's regularizer. This has been observed before for Bayesian learners, too (Zhu, 2013).

We encourage the reader to distinguish two senses of uniqueness. The teaching set itself is not necessarily unique. In the construction (19), any $a \neq 0$ leads to a valid teaching set. Nonetheless, any one of the teaching sets will lead to the unique solution $\boldsymbol{\theta}^*$ in (18).

Corollary 1 *The teaching dimension $TD(\boldsymbol{\theta}^*, \mathcal{A}_{ridge}^{hom}) = 1$ for homogeneous ridge regression and target $\boldsymbol{\theta}^* \neq \mathbf{0}$.*

Proof Substituting A by I in LB1 (3), we obtain the lower bound $d - \text{Rank}(I) + 1 = 1$ which matches the teaching set size in (19). \blacksquare

Homogeneous SVM solves the problem:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \sum_{i=1}^n \max(1 - y_i \mathbf{x}_i^\top \boldsymbol{\theta}, 0) + \frac{\lambda}{2} \|\boldsymbol{\theta}\|^2. \quad (20)$$

To teach this learner one training item is in general not enough: we will show that we need $\lceil \lambda \|\boldsymbol{\theta}^*\|^2 \rceil$ training items. In fact, we will construct such a teaching set consisting of *identical* training items. It is well-known in the teaching literature that a teaching set does not need to consist of *i.i.d.* samples from a distribution, and can look unusual. It is possible to incorporate additional constraints into a teaching problem if one wants the training items to be diverse, but we do not consider that in the present paper.

Proposition 2 *Given any target model $\boldsymbol{\theta}^* \neq \mathbf{0}$, the following is a teaching set for homogeneous SVM (20). There are $n = \lceil \lambda \|\boldsymbol{\theta}^*\|^2 \rceil$ identical training items, each taking the form*

$$\mathbf{x}_i = \frac{\lambda \boldsymbol{\theta}^*}{\lceil \lambda \|\boldsymbol{\theta}^*\|^2 \rceil}, \quad y_i = 1. \quad (21)$$

Proof We only need to verify that the KKT condition holds for $\boldsymbol{\theta}^*$. Due to the strong convexity of (20) uniqueness is guaranteed automatically. We denote the subgradient

$\partial_a \max(1 - a, 0) = -\partial_1 \max(1 - a, 0) = -\mathbf{I}(a)$, where

$$\mathbf{I}(a) = \begin{cases} 1, & \text{if } a < 1 \\ [0, 1], & \text{if } a = 1 \\ 0, & \text{otherwise} \end{cases}. \quad (22)$$

The KKT condition is

$$\begin{aligned} & \sum_{i=1}^n -y_i \mathbf{x}_i \partial_1 \max(1 - y_i \mathbf{x}_i^\top \boldsymbol{\theta}^*, 0) + \lambda \boldsymbol{\theta}^* \\ &= \sum_{i=1}^n -y_i \mathbf{x}_i \mathbf{I}(y_i \mathbf{x}_i^\top \boldsymbol{\theta}^*) + \lambda \boldsymbol{\theta}^* \\ &= -n \frac{\lambda \|\boldsymbol{\theta}^*\|^2}{[\lambda \|\boldsymbol{\theta}^*\|^2]} \mathbf{I} \left(\frac{\lambda \|\boldsymbol{\theta}^*\|^2}{[\lambda \|\boldsymbol{\theta}^*\|^2]} \right) + \lambda \boldsymbol{\theta}^* \\ &= -\lambda \boldsymbol{\theta}^* \mathbf{I} \left(\frac{\lambda \|\boldsymbol{\theta}^*\|^2}{[\lambda \|\boldsymbol{\theta}^*\|^2]} \right) + \lambda \boldsymbol{\theta}^* \\ &\geq \mathbf{0} \end{aligned}$$

where the last line is due to $\mathbf{I} \left(\frac{\lambda \|\boldsymbol{\theta}^*\|^2}{[\lambda \|\boldsymbol{\theta}^*\|^2]} \right)$ giving either the set $[0, 1]$ or the value 1. \blacksquare

Corollary 2 *The teaching dimension $TD(\boldsymbol{\theta}^*, \mathcal{A}_{\text{sym}}^{\text{hom}}) = [\lambda \|\boldsymbol{\theta}^*\|^2]$ for homogeneous SVM and target $\boldsymbol{\theta}^* \neq \mathbf{0}$.*

Proof We show this number matches LB2. Let $A = I$, $\ell(a, b) = \max(1 - ab, 0)$, and consider the denominator of (9):

$$\begin{aligned} \sup_{\alpha \in \mathbb{R}, y \in \mathcal{Y}, g \in -\partial_1 \ell(\alpha \|\boldsymbol{\theta}^*\|^2, y)} \alpha g &= \sup_{\alpha, y \in \{-1, 1\}, g \in \partial_1 \ell(g \alpha \|\boldsymbol{\theta}^*\|^2)} \alpha g \\ &= \sup_{\alpha, g \in \mathbf{I}(\alpha \|\boldsymbol{\theta}^*\|^2)} \alpha g \\ &= \frac{1}{\|\boldsymbol{\theta}^*\|^2} \end{aligned}$$

where the first equality is due to $\partial_1 \ell(a, b) = -b \mathbf{I}(ab)$. Therefore, $LB2 = [\lambda \|\boldsymbol{\theta}^*\|^2]$ which matches the construction in (21). \blacksquare

Homogeneous logistic regression solves the problem:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \sum_{i=1}^n \log(1 + \exp(-y_i \mathbf{x}_i^\top \boldsymbol{\theta})) + \frac{\lambda}{2} \|\boldsymbol{\theta}\|^2 \quad (23)$$

where \log has base e . The situation is similar to homogeneous SVM. However, due to the negative log likelihood term we have a coefficient defined by the Lambert W function (Corless et al., 1996), which we denote by W_{lam} . Recall the defining equation for Lambert W

function is $W_{\text{lam}}(x)e^{W_{\text{lam}}(x)} = x$. We further define

$$\tau_{\max} := \max \frac{t}{1 + e^t} = W_{\text{lam}}(1/e) \approx 0.2785,$$

where the equality can be derived in following: The optimal t^* satisfies

$$1 + e^{t^*} = t^* e^{t^*} \Leftrightarrow (t^* - 1)e^{t^* - 1} = 1/e$$

which suggests $t^* = W_{\text{lam}}(1/e) + 1$. We apply the optimality condition above and the optimal value of t^* to obtain

$$\max_t \frac{t}{1 + e^t} = \frac{t^*}{1 + e^{t^*}} = \frac{1}{e^{t^*}} = \frac{1}{e \cdot e^{W_{\text{lam}}(1/e)}} = W_{\text{lam}}(1/e).$$

For any value $a \leq \tau_{\max}$, we define $\tau^{-1}(a)$ as the solution to $a = \frac{t}{1 + e^t}$. By using the Lambert W function $\tau^{-1}(a)$ can be expressed as $\tau^{-1}(a) \equiv a - W_{\text{lam}}(-ae^a)$, which can be derived from

$$\frac{t}{1 + e^t} = \frac{a - W_{\text{lam}}(-ae^a)}{1 + e^{a - W_{\text{lam}}(-ae^a)}} = \frac{a + ae^a/e^{W_{\text{lam}}(-ae^a)}}{1 + e^a - W_{\text{lam}}(-ae^a)} = a.$$

Proposition 3 *Given any target model $\boldsymbol{\theta}^* \neq \mathbf{0}$, the following is a teaching set for homogeneous logistic regression (23). There are $n = \left\lceil \frac{\lambda \|\boldsymbol{\theta}^*\|^2}{\tau_{\max}} \right\rceil$ identical training items, each taking the form*

$$\mathbf{x}_i = \tau^{-1} \left(\lambda \|\boldsymbol{\theta}^*\|^2 \left\lceil \frac{\lambda \|\boldsymbol{\theta}^*\|^2}{\tau_{\max}} \right\rceil^{-1} \right) \frac{\boldsymbol{\theta}^*}{\|\boldsymbol{\theta}^*\|^2}, \quad y_i = 1. \quad (24)$$

Proof We first verify that $\boldsymbol{\theta}^*$ is a solution to (23) based on the teaching set construction in (24). We only need to verify the gradient of (23) is zero. Computing the gradient of (23), we have

$$\begin{aligned} & \sum_{i=1}^n \frac{-y_i \mathbf{x}_i}{1 + \exp\{y_i \mathbf{x}_i^\top \boldsymbol{\theta}^*\}} + \lambda \boldsymbol{\theta}^* \\ &= -n \frac{\mathbf{x}_i}{1 + \exp \left\{ \tau^{-1} \left(\lambda \|\boldsymbol{\theta}^*\|^2 \left\lceil \frac{\lambda \|\boldsymbol{\theta}^*\|^2}{\tau_{\max}} \right\rceil^{-1} \right) \right\}} + \lambda \boldsymbol{\theta}^* \\ &= -n \frac{\tau^{-1} \left(\lambda \|\boldsymbol{\theta}^*\|^2 \left\lceil \frac{\lambda \|\boldsymbol{\theta}^*\|^2}{\tau_{\max}} \right\rceil^{-1} \right) \boldsymbol{\theta}^*}{\left(\lambda \|\boldsymbol{\theta}^*\|^2 \left\lceil \frac{\lambda \|\boldsymbol{\theta}^*\|^2}{\tau_{\max}} \right\rceil^{-1} \right) \|\boldsymbol{\theta}^*\|^2} + \lambda \boldsymbol{\theta}^* \\ &= -n \lambda \|\boldsymbol{\theta}^*\|^2 \left\lceil \frac{\lambda \|\boldsymbol{\theta}^*\|^2}{\tau_{\max}} \right\rceil^{-1} \frac{\boldsymbol{\theta}^*}{\|\boldsymbol{\theta}^*\|^2} + \lambda \boldsymbol{\theta}^* \\ &= \mathbf{0}, \end{aligned}$$

where the third equality uses the fact $\lambda \|\boldsymbol{\theta}^*\|^2 \left\lceil \frac{\lambda \|\boldsymbol{\theta}^*\|^2}{\tau_{\max}} \right\rceil^{-1} \leq \tau_{\max}$ and the property $a = \frac{\tau^{-1}(a)}{1 + e^{\tau^{-1}(a)}}$. The strong convexity of (23) automatically implies uniqueness. \blacksquare

Corollary 3 The teaching dimension $TD(\boldsymbol{\theta}^*, \mathcal{A}_{\log}^{\text{hom}}) = \left\lceil \frac{\lambda \|\boldsymbol{\theta}^*\|^2}{\tau_{\max}} \right\rceil$ for homogeneous logistic regression and target $\boldsymbol{\theta}^* \neq \mathbf{0}$.

Proof We show that the number matches LB2. In (9) let $A = I$ and $\ell(a, b) = \log(1 + \exp(-ab))$. The denominator of LB2 is:

$$\begin{aligned} \sup_{\alpha \in \mathbb{R}, \beta \in \mathcal{Y}} \sup_{\beta \in \mathcal{Y}} \ell(\alpha \|\boldsymbol{\theta}^*\|^2, \beta) &= \sup_{\alpha, \beta \in \{-1, 1\}, \beta = y(1 + \exp\{\beta \alpha \|\boldsymbol{\theta}^*\|^2\})^{-1}} \alpha \beta \\ &= \sup_{\alpha, \beta = (1 + \exp\{\alpha \|\boldsymbol{\theta}^*\|^2\})^{-1}} \alpha \beta \\ &= \sup_{\alpha} \frac{\alpha}{1 + \exp\{\alpha \|\boldsymbol{\theta}^*\|^2\}} \\ &= \|\boldsymbol{\theta}^*\|^{-2} \sup_t \frac{\alpha}{1 + \exp\{t\}} \\ &= \frac{\tau_{\max}}{\|\boldsymbol{\theta}^*\|^2}, \end{aligned}$$

which implies $LB2 = \left\lceil \frac{\lambda \|\boldsymbol{\theta}^*\|^2}{\tau_{\max}} \right\rceil$. ■

3.3 The Teaching Dimension $TD(\boldsymbol{\theta}^*, \mathcal{A}_{\text{opt}})$ of Three Inhomogeneous Learners

Inhomogeneous learners are defined by $\boldsymbol{\theta} = [w; b]$ where the weight vector $w \in \mathbb{R}^d$ and the scalar offset $b \in \mathbb{R}$. The offset b is not regularized. Similar to the previous section, we need to instantiate the teaching dimension lower bounds and design the teaching sets. We show that the size of our teaching set exactly matches the lower bound for inhomogeneous ridge regression, and differs from the lower bound of inhomogeneous SVM and logistic regression by at most one due to rounding. Therefore, up to rounding we also establish the teaching dimension for these inhomogeneous learners.

Inhomogeneous ridge regression solves the problem:

$$\min_{w \in \mathbb{R}^d, b \in \mathbb{R}} \sum_{i=1}^n \frac{1}{2} (\mathbf{x}_i^\top w + b - y_i)^2 + \frac{\lambda}{2} \|w\|^2 \quad (25)$$

Proposition 4 Given any target model $[w^*; b^*]$, if $w^* = \mathbf{0}$ (b^* can be an arbitrary value), the following is a teaching set for inhomogeneous ridge regression (25) with $n = 1$:

$$\mathbf{x}_1 = \mathbf{0}, \quad y_1 = b^*. \quad (26)$$

If $w^* \neq \mathbf{0}$, any $n = 2$ items satisfying the following are a teaching set for $a \neq 0$:

$$\mathbf{x}_1 - \mathbf{x}_2 = a w^*, \quad y_1 = \mathbf{x}_1^\top w^* + b^* + \frac{\lambda}{a}, \quad y_2 = y_1 - a \|w^*\|^2 - 2 \frac{\lambda}{a}. \quad (27)$$

Proof We first prove the case for $w^* = \mathbf{0}$. We can verify that the KKT condition is satisfied by designing \mathbf{x}_1 and y_1 as in (26):

$$\begin{aligned} (\mathbf{x}_1^\top w^* + b^* - y_1) \mathbf{x}_1 + \lambda w^* &= \mathbf{0} \\ \mathbf{x}_1^\top w^* + b^* - y_1 &= 0. \end{aligned}$$

The uniqueness of $[w^*; b^*]$ is indicated by the strong convexity of (25) when $n = 1$.

We then prove the case for $w^* \neq \mathbf{0}$. With simple algebra, we can verify the KKT condition holds via the construction in (27):

$$\begin{aligned} (\mathbf{x}_1^\top w^* + b^* - y_1) \mathbf{x}_1 + (\mathbf{x}_2^\top w^* + b^* - y_2) \mathbf{x}_2 + \lambda w^* &= \mathbf{0} \\ (\mathbf{x}_1^\top w^* + b^* - y_1) + (\mathbf{x}_2^\top w^* + b^* - y_2) &= 0. \end{aligned}$$

Similarly, the uniqueness is implied by the strong convexity of (25) when $n = 2$. ■

Corollary 4 The teaching dimension for inhomogeneous ridge regression with target $\boldsymbol{\theta}^* = [w^*; b^*]$ is $TD(\boldsymbol{\theta}^*, \mathcal{A}_{\text{ridge}}^{\text{inh}}) = 1$ if target $w^* = \mathbf{0}$, or $TD(\boldsymbol{\theta}^*, \mathcal{A}_{\text{ridge}}^{\text{inh}}) = 2$ if $w^* \neq \mathbf{0}$, regardless of the target offset b^* .

Proof We match the lower bound LB1 in (3). Note $\boldsymbol{\theta}^* = [w^*; b^*] \in \mathbb{R}^{d+1}$, and A in this case is a $(d+1) \times (d+1)$ matrix with the $d \times d$ identity matrix I_d padded with one additional row and column of zeros for the offset. Therefore $\text{Rank}(A) = \text{Rank}(I_d) = d$. When $w^* = \mathbf{0}$, $A\boldsymbol{\theta}^* = \mathbf{0}$ and $LB1 = (d+1) - \text{Rank}(A) = 1$. When $w^* \neq \mathbf{0}$, $A\boldsymbol{\theta}^* \neq \mathbf{0}$ and $LB1 = (d+1) - \text{Rank}(A) + 1 = 2$. These lower bounds match the teaching set sizes in (26) and (27), respectively. ■

Inhomogeneous SVM solves the problem:

$$\min_{w \in \mathbb{R}^d, b \in \mathbb{R}} \max_{i=1}^n \max(1 - y_i (\mathbf{x}_i^\top w + b), 0) + \frac{\lambda}{2} \|w\|^2. \quad (28)$$

Proposition 5 Given any target model $[w^*; b^*]$ with $w^* \neq \mathbf{0}$, the following is a teaching set for inhomogeneous SVM (28). We need $n = 2 \left\lceil \frac{\lambda \|w^*\|^2}{2} \right\rceil$ training items, half of which are identical positive items $\mathbf{x}_i = \mathbf{x}_+$, $y_i = 1$, $\forall i \in \{1, \dots, \frac{n}{2}\}$ and half identical negative items $\mathbf{x}_i = \mathbf{x}_-$, $y_i = -1$, $\forall i \in \{\frac{n}{2} + 1, \dots, n\}$. \mathbf{x}_+ and \mathbf{x}_- can be designed as any vectors satisfying

$$\mathbf{x}_+^\top w^* = 1 - b^*, \quad \mathbf{x}_- = \mathbf{x}_+ - \frac{2w^*}{\|w^*\|^2}. \quad (29)$$

Proof Unlike in previous learners (including homogeneous SVM), we no longer have strong convexity w.r.t. b . In order to prove that (29) is a teaching set, we need to verify the KKT condition and verify solution uniqueness.

We first verify the KKT condition to show that the solution under (29) includes the target model $[w^*; b^*]$. From (29), we have

$$\mathbf{x}_+^\top w^* + b^* = 1, \quad \mathbf{x}_-^\top w^* + b^* = -1. \quad (30)$$

Applying them to the KKT condition and using the notation in (22) we obtain

$$\begin{aligned}
& -\frac{n}{2}\mathbf{I}(\mathbf{x}_+^\top \mathbf{w}^* + b^*) \begin{bmatrix} \mathbf{x}_+ \\ 1 \end{bmatrix} + \frac{n}{2}\mathbf{I}(-\mathbf{x}_-^\top \mathbf{w}^* - b^*) \begin{bmatrix} \mathbf{x}_- \\ 1 \end{bmatrix} + \begin{bmatrix} \lambda \mathbf{w}^* \\ 0 \end{bmatrix} \\
& = -\frac{n}{2}\mathbf{I}(1) \begin{bmatrix} \mathbf{x}_+ \\ 1 \end{bmatrix} + \frac{n}{2}\mathbf{I}(1) \begin{bmatrix} \mathbf{x}_- \\ 1 \end{bmatrix} + \begin{bmatrix} \lambda \mathbf{w}^* \\ 0 \end{bmatrix} \\
& \supseteq \frac{n}{2}\mathbf{I}(1) \begin{bmatrix} \mathbf{x}_- - \mathbf{x}_+ \\ 0 \end{bmatrix} + \begin{bmatrix} \lambda \mathbf{w}^* \\ 0 \end{bmatrix} \quad \text{setting the last dimension to 0} \\
& = \mathbf{I}(1) \begin{bmatrix} -\frac{n}{\|\mathbf{w}^*\|^2} \mathbf{w}^* \\ 0 \end{bmatrix} + \begin{bmatrix} \lambda \mathbf{w}^* \\ 0 \end{bmatrix} \quad \text{applying (29)} \\
& \supseteq \mathbf{I}(1) \begin{bmatrix} -\lambda \mathbf{w}^* \\ 0 \end{bmatrix} + \begin{bmatrix} \lambda \mathbf{w}^* \\ 0 \end{bmatrix} \quad \text{observing } n \geq \lambda \|\mathbf{w}^*\|^2 \\
& \ni \mathbf{0}.
\end{aligned}$$

It proves that $[\mathbf{w}^*; b^*]$ solves (28) by our teaching set construction.

Next we prove uniqueness by contradiction. We use $f(\mathbf{w}, b)$ to denote the objective function in (28) under the teaching set. It is easy to verify that $f(\mathbf{w}^*, b^*) = \frac{\lambda}{2}\|\mathbf{w}^*\|^2$. Assume that there exists another solution $[\bar{\mathbf{w}}; \bar{b}]$ different from $[\mathbf{w}^*; b^*]$. We can obtain $\|\bar{\mathbf{w}}\|^2 \leq \|\mathbf{w}^*\|^2$ due to

$$\frac{\lambda}{2}\|\mathbf{w}^*\|^2 = f(\mathbf{w}^*, b^*) = f(\bar{\mathbf{w}}, \bar{b}) \geq \frac{\lambda}{2}\|\bar{\mathbf{w}}\|^2.$$

The second equality is due to $[\bar{\mathbf{w}}; \bar{b}]$ being a solution; the inequality is due to whole-part relationship. Therefore, there are only two possibilities for the norm of $\bar{\mathbf{w}}$: $\|\bar{\mathbf{w}}\| = \|\mathbf{w}^*\|$ or $\|\bar{\mathbf{w}}\| = t\|\mathbf{w}^*\|$ for some $0 \leq t < 1$. Next we will show that both cases are impossible.

(Case 1) For the case $\|\bar{\mathbf{w}}\| = \|\mathbf{w}^*\|$, we have

$$\begin{aligned}
f(\bar{\mathbf{w}}, \bar{b}) &= \frac{n}{2} \max \left(1 - (\mathbf{x}_+^\top \bar{\mathbf{w}} + \bar{b}), 0 \right) + \frac{n}{2} \max \left(1 + (\mathbf{x}_-^\top \bar{\mathbf{w}} + \bar{b}), 0 \right) + \frac{\lambda}{2} \|\bar{\mathbf{w}}\|^2 \\
&= \frac{n}{2} \max \left(\underbrace{\mathbf{x}_+^\top (\mathbf{w}^* - \bar{\mathbf{w}})}_{=: \Delta_+} + (\bar{b}^* - \bar{b}), 0 \right) + \frac{n}{2} \max \left(\underbrace{-\mathbf{x}_-^\top (\mathbf{w}^* - \bar{\mathbf{w}})}_{=: \Delta_-} - (\bar{b}^* - \bar{b}), 0 \right) \\
&\quad + \frac{\lambda}{2} \|\mathbf{w}^*\|^2 \\
&= \frac{n}{2} \max(\Delta_+, 0) + \frac{n}{2} \max(\Delta_-, 0) + f(\mathbf{w}^*, b^*).
\end{aligned}$$

From $f(\bar{\mathbf{w}}, \bar{b}) = f(\mathbf{w}^*, b^*)$, it follows $\Delta_+ \leq 0$ and $\Delta_- \leq 0$. Since

$$0 \geq \Delta_+ + \Delta_- = (\mathbf{x}_+ - \mathbf{x}_-)^\top (\mathbf{w}^* - \bar{\mathbf{w}}) = \frac{2(\mathbf{w}^*)^\top (\mathbf{w}^* - \bar{\mathbf{w}})}{\|\mathbf{w}^*\|^2} = 2 - 2 \frac{\bar{\mathbf{w}}^\top \mathbf{w}^*}{\|\mathbf{w}^*\|^2},$$

we have $\bar{\mathbf{w}}^\top \mathbf{w}^* \geq \|\mathbf{w}^*\|^2$. But because $\|\bar{\mathbf{w}}\| = \|\mathbf{w}^*\|$, we must have $\bar{\mathbf{w}} = \mathbf{w}^*$. Applying this new observation to $\Delta_+ \leq 0$ and $\Delta_- \leq 0$, we obtain $\bar{b}^* = b$. It means that $[\mathbf{w}^*; b^*] = [\bar{\mathbf{w}}; \bar{b}]$, contradicting our assumption $[\mathbf{w}^*; b^*] \neq [\bar{\mathbf{w}}; \bar{b}]$.

(Case 2) Next we turn to the case $\|\bar{\mathbf{w}}\| = t\|\mathbf{w}^*\|$ for some $t \in [0, 1)$. Recall our assumption that $[\bar{\mathbf{w}}; \bar{b}]$ solves (28). Then it follows that the following specific construction $[\hat{\mathbf{w}}; \hat{b}]$ solves (28) as well:

$$\hat{\mathbf{w}} = t\mathbf{w}^*, \quad \hat{b} = tb^*. \quad (31)$$

To see this, we consider the following optimization problem:

$$\begin{aligned}
& \min_{\mathbf{w}, b} L(\mathbf{w}, b) := \frac{n}{2} \max(1 - (\mathbf{x}_+^\top \mathbf{w} + b), 0) + \frac{n}{2} \max(1 + (\mathbf{x}_-^\top \mathbf{w} + b), 0) \\
& \text{s.t. } \|\mathbf{w}\| \leq t\|\mathbf{w}^*\|.
\end{aligned} \quad (32)$$

Since $[\bar{\mathbf{w}}; \bar{b}]$ solves (28), it is easy to see that $[\bar{\mathbf{w}}; \bar{b}]$ solves (32) too, otherwise there exists a solution for (32) which gives a lower function value on (28). Then we can verify that $[\bar{\mathbf{w}}; \bar{b}]$ solves (32) as well by showing the following geometric optimality condition holds:

$$-\left[\frac{\partial L(\mathbf{w}, b)}{\partial \mathbf{w}} \right]_{[\bar{\mathbf{w}}; \bar{b}]} \cap \underbrace{\mathcal{N}_{\|\mathbf{w}\| \leq t\|\mathbf{w}^*\|}(\hat{\mathbf{w}}, \hat{b})}_{\text{Normal cone to the set } \{\mathbf{w}; \hat{b} : \|\mathbf{w}\| \leq t\|\mathbf{w}^*\| \text{ at } [\bar{\mathbf{w}}; \bar{b}]} \neq \emptyset.$$

Given a convex closed set Ω and a point $\theta \in \Omega$, the normal cone at point θ is defined to be a set

$$\mathcal{N}_\Omega(\theta) = \{\phi : \langle \phi, \psi - \theta \rangle \leq 0 \forall \psi \in \Omega\}.$$

The optimality condition basically suggests that at the optimal point, the negative (sub)gradient direction overlaps with the normal cone. In other words, there does not exist any valid direction to decrease the objective at the optimal point. Readers can refer to Nocedal and Wright (2006) or Bertsekas and Nedic (2003) for more explanations about the geometric optimality condition.

Because of (30) and (31), we have $\mathbf{x}_+^\top \hat{\mathbf{w}} + \hat{b} = t < 1$. Thus at $[\hat{\mathbf{w}}; \hat{b}]$ the subgradient is

$$-\left[\frac{\partial L(\mathbf{w}, b)}{\partial \mathbf{w}} \right]_{[\hat{\mathbf{w}}; \hat{b}]} = \frac{n}{2} \begin{bmatrix} \mathbf{x}_+ - \mathbf{x}_- \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{n\mathbf{w}^*}{\|\mathbf{w}^*\|^2} \\ 0 \end{bmatrix}$$

And the normal cone is

$$\mathcal{N}_{\|\mathbf{w}\| \leq t\|\mathbf{w}^*\|}(\hat{\mathbf{w}}, \hat{b}) = \left\{ s \begin{bmatrix} \mathbf{w}^* \\ 0 \end{bmatrix} \mid s \geq 0 \right\}.$$

The intersection is non-empty by choosing $s = \frac{n}{\|\mathbf{w}^*\|^2}$. Since both $[\hat{\mathbf{w}}; \hat{b}]$ and $[\bar{\mathbf{w}}; \bar{b}]$ solve (32), we have $L(\hat{\mathbf{w}}, \hat{b}) = L(\bar{\mathbf{w}}, \bar{b})$. Together with $\|\hat{\mathbf{w}}\| = \|\bar{\mathbf{w}}\|$, we have

$$f(\hat{\mathbf{w}}, \hat{b}) = L(\hat{\mathbf{w}}, \hat{b}) + \frac{\lambda}{2} \|\hat{\mathbf{w}}\|^2 = f(\bar{\mathbf{w}}, \bar{b}) = f(\mathbf{w}^*, b^*).$$

Therefore, we proved that $(\hat{\mathbf{w}}, \hat{b})$ solves (28) as well. To see the contradiction, let us check the function value of $f(\hat{\mathbf{w}}, \hat{b})$ via a different route:

$$\begin{aligned}
f(\hat{\mathbf{w}}, \hat{b}) &= f(\hat{\mathbf{w}}^*, t\hat{b}^*) \\
&= \sum_{i=1}^{\frac{n}{2}} \max \left(1 - t(\mathbf{x}_+^\top \hat{\mathbf{w}}^* + b^*), 0 \right) + \sum_{i=1}^{\frac{n}{2}} \max \left(1 + t(\mathbf{x}_-^\top \hat{\mathbf{w}}^* + b^*), 0 \right) + \frac{\lambda}{2} \|\hat{\mathbf{w}}^*\|^2 t^2 \\
&= \sum_{i=1}^{\frac{n}{2}} \max(1 - t, 0) + \sum_{i=1}^{\frac{n}{2}} \max(1 + t, 0) + \frac{\lambda}{2} \|\hat{\mathbf{w}}^*\|^2 t^2 \\
&\geq n(1 - t) - \frac{\lambda}{2} \|\hat{\mathbf{w}}^*\|^2 (1 - t^2) + \frac{\lambda}{2} \|\hat{\mathbf{w}}^*\|^2 \\
&\geq n(1 - t) - \frac{n}{2} (1 - t^2) + \frac{\lambda}{2} \|\hat{\mathbf{w}}^*\|^2 \\
&= \frac{n}{2} (1 - t)^2 + f(\hat{\mathbf{w}}^*, b^*) \\
&> f(\hat{\mathbf{w}}^*, b^*),
\end{aligned}$$

where the first inequality uses the fact that $n \geq \lambda \|\hat{\mathbf{w}}^*\|^2$. It contradicts our early assertion $f(\hat{\mathbf{w}}, \hat{b}) = f(\hat{\mathbf{w}}^*, b^*)$. Putting cases 1 and 2 together we prove uniqueness. \blacksquare

Our construction of the teaching set in (29) requires $n = 2 \left\lceil \frac{\lambda \|\mathbf{w}^*\|^2}{2} \right\rceil$ training items. This is an upper bound on the teaching dimension. Meanwhile, we show below that the inhomogeneous SVM lower bound is $LB3 = \lceil \lambda \|\mathbf{w}^*\|^2 \rceil$. There can be a difference of at most one between the lower and upper bounds, which we call the ‘‘rounding effect.’’ We suspect that this small gap is a technicality and not intrinsic. However, at present we do not have a teaching set construction that bridges this gap. Therefore, we state the teaching dimension as an interval in the following corollary and leave the precise value as an open question for future research.

Corollary 5 *The teaching dimension for inhomogeneous SVM and target $\boldsymbol{\theta}^* = [\mathbf{w}^*; b^*]$ where $\mathbf{w}^* \neq \mathbf{0}$ is in the interval $\lceil \lambda \|\mathbf{w}^*\|^2 \rceil \leq TD(\boldsymbol{\theta}^*, \mathcal{A}_{svm}^{inh}) \leq 2 \left\lceil \frac{\lambda \|\mathbf{w}^*\|^2}{2} \right\rceil$.*

Proof The upper bound directly follows Proposition 5. We only need to show the lower bound $LB3 = \lceil \lambda \|\mathbf{w}^*\|^2 \rceil$ in Theorem 3. Let $A = I$, $\ell(a) = \max(1 - a, 0)$, and consider the denominator of (14):

$$\sup_{\alpha \in \mathbb{R}, g \in -\partial \ell(\alpha \|\mathbf{w}^*\|^2)} \alpha g = \sup_{\alpha, g \in \mathbf{I}(\alpha \|\mathbf{w}^*\|^2)} \alpha g = \frac{1}{\|\mathbf{w}^*\|^2}$$

where the first equality is due to $\partial \ell(a) = -\mathbf{I}(a)$. Therefore, $LB3 = \lceil \lambda \|\mathbf{w}^*\|^2 \rceil$ which proves the lower bound. \blacksquare

Inhomogeneous logistic regression solves the problem

$$\min_{\mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R}} \sum_{i=1}^n \log(1 + \exp\{-y_i(\mathbf{x}_i^\top \mathbf{w} + b)\}) + \frac{\lambda}{2} \|\mathbf{w}\|^2. \quad (33)$$

Proposition 6 *To create a teaching set for target model $[\mathbf{w}^*; b^*]$ with nonzero \mathbf{w}^* for inhomogeneous logistic regression (33), we can use $n = 2 \left\lceil \frac{\lambda \|\mathbf{w}^*\|^2}{2\tau_{\max}} \right\rceil$ training items where $\mathbf{x}_i = \mathbf{x}_+$, $y_i = 1$, $\forall i \in \{1, \dots, \frac{n}{2}\}$ and $\mathbf{x}_i = \mathbf{x}_-$, $y_i = -1$, $\forall i \in \{\frac{n}{2} + 1, \dots, n\}$. \mathbf{x}_+ and \mathbf{x}_- can be designed as any vectors satisfying*

$$\mathbf{x}_+^\top \mathbf{w}^* = t - b^*, \quad \mathbf{x}_- = \mathbf{x}_+ - \frac{2t}{\|\mathbf{w}^*\|^2} \mathbf{w}^*, \quad (34)$$

where the constant t is defined by $t := \tau^{-1} \left(\frac{\lambda \|\mathbf{w}^*\|^2}{n} \right)$.

Proof We first point out that for t to be well-defined the argument to $\tau^{-1}(\cdot)$ has to be bounded $\frac{\lambda \|\mathbf{w}^*\|^2}{n} \leq \tau_{\max}$. This implies $n \geq \frac{\lambda \|\mathbf{w}^*\|^2}{\tau_{\max}}$. The size of our proposed teaching set is the smallest among all such symmetric construction that satisfy this constraint.

We verify that the KKT condition to show the construction in (34) includes the solution $[\mathbf{w}^*; b^*]$. From (34), we have

$$\mathbf{x}_+^\top \mathbf{w}^* + b^* = t \quad \mathbf{x}_-^\top \mathbf{w}^* + b^* = -t.$$

We apply them and the teaching set construction to compute the gradient of (33):

$$\begin{aligned}
& -\frac{n}{2} \frac{1}{1 + \exp\{\mathbf{x}_+^\top \mathbf{w}^* + b^*\}} \begin{bmatrix} \mathbf{x}_+ \\ 1 \end{bmatrix} + \frac{n}{2} \frac{1}{1 + \exp\{-\mathbf{x}_-^\top \mathbf{w}^* - b^*\}} \begin{bmatrix} \mathbf{x}_- \\ 1 \end{bmatrix} + \begin{bmatrix} \lambda \mathbf{w}^* \\ 0 \end{bmatrix} \\
&= -\frac{n}{2} \frac{1}{1 + \exp\{t\}} \begin{bmatrix} \mathbf{x}_+ \\ 1 \end{bmatrix} + \frac{n}{2} \frac{1}{1 + \exp\{t\}} \begin{bmatrix} \mathbf{x}_- \\ 1 \end{bmatrix} + \begin{bmatrix} \lambda \mathbf{w}^* \\ 0 \end{bmatrix} \\
&= -\frac{n}{\|\mathbf{w}^*\|^2} \frac{t}{1 + \exp\{t\}} \begin{bmatrix} \mathbf{w}^* \\ 0 \end{bmatrix} + \begin{bmatrix} \lambda \mathbf{w}^* \\ 0 \end{bmatrix} \\
&= -\frac{n}{\|\mathbf{w}^*\|^2} \frac{\lambda \|\mathbf{w}^*\|^2}{n} \begin{bmatrix} \mathbf{w}^* \\ 0 \end{bmatrix} + \begin{bmatrix} \lambda \mathbf{w}^* \\ 0 \end{bmatrix} \\
&= 0.
\end{aligned}$$

This verifies the KKT condition.

Finally we show uniqueness. The Hessian matrix of the objective function (33) under our training set (34) is:

$$\frac{n}{2} \underbrace{\frac{\exp\{t\}}{(1 + \exp\{t\})^2} \begin{bmatrix} \mathbf{x}_+ \mathbf{x}_+^\top + \mathbf{x}_- \mathbf{x}_-^\top & \mathbf{x}_+ + \mathbf{x}_- \\ \mathbf{x}_+ + \mathbf{x}_-^\top & 2 \end{bmatrix}}_{:=A} + \lambda \underbrace{\begin{bmatrix} I & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{bmatrix}}_{:=B}.$$

Note $a > 0$ and $A = \begin{bmatrix} \mathbf{x}_+ & \\ & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_- \\ 1 \end{bmatrix} + \begin{bmatrix} \mathbf{x}_- \\ 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_+ & \\ & 1 \end{bmatrix}$ is positive semi-definite. We show that $aA + \lambda B$ is positive definite. Suppose not. Then there exists $[\mathbf{u}; v] \neq \mathbf{0}$ such that $[\mathbf{u}; v]^\top (aA + \lambda B) [\mathbf{u}; v] = 0$. This implies $[\mathbf{u}; v]^\top (aA) [\mathbf{u}; v] + \lambda \mathbf{u}^\top \mathbf{u} = 0$. Since the first term is non-negative due to A being positive semi-definite, $\mathbf{u} = \mathbf{0}$. But then we have $2av^2 = 0$ which implies $[\mathbf{u}; v] = \mathbf{0}$, a contradiction. Therefore uniqueness is guaranteed. \blacksquare

Corollary 6 *The teaching dimension for inhomogeneous logistic regression and target θ^* = $\lfloor \mathbf{w}^*; b^* \rfloor$ where $\mathbf{w}^* \neq \mathbf{0}$ is in the interval $\left\lfloor \frac{\lambda \|\mathbf{w}^*\|^2}{T_{\max}} \right\rfloor \leq TD(\theta^*, \mathcal{A}_{log}^{inh}) \leq 2 \left\lfloor \frac{\lambda \|\mathbf{w}^*\|^2}{2\tau_{\max}} \right\rfloor$.*

Proof The upper bound directly follows Proposition 6. We only need to show the lower bound $\left\lfloor \frac{\lambda \|\mathbf{w}^*\|^2}{T_{\max}} \right\rfloor$ by applying $LB3$ in Theorem 3. Let $A = I$ and $\ell(a) = \log(1 + \exp\{-a\})$ and consider the denominator of (14):

$$\begin{aligned} \sup_{\alpha \in \mathbb{R}, g \in \mathcal{H}(\alpha \|\mathbf{w}^*\|^2)} \alpha g &= \sup_{\alpha, g = (1 + \exp\{\alpha \|\mathbf{w}^*\|^2\})^{-1}} \alpha g \\ &= \sup_{\alpha} \frac{\alpha}{1 + \exp\{\alpha \|\mathbf{w}^*\|^2\}} \\ &= \|\mathbf{w}^*\|^{-2} \sup_t \frac{t}{1 + \exp\{t\}} \\ &= \frac{\tau_{\max}}{\|\mathbf{w}^*\|^2}, \end{aligned}$$

which implies $LB3 = \left\lfloor \frac{\lambda \|\mathbf{w}^*\|^2}{T_{\max}} \right\rfloor$. ■

4. Teaching a Decision Boundary Instead of a Parameter

In section 3 we considered the teaching goal where the learner is required to learn the exact *target parameter* θ^* . But when the learner is a classifier often a weaker teaching goal is sufficient, namely teaching the learner a *target decision boundary*. In this section we consider this teaching goal. Equivalently, such a goal is defined by the set of parameters that produce the target decision boundary. Teaching is successful if the learner arrives at any one parameter within that set.

In the case of inhomogeneous linear learners, the linear decision boundary $\{\mathbf{x} \mid \mathbf{x}^\top \mathbf{w}^* + b^* = 0\}$ is identified with the parameter set $\{t[\mathbf{w}^*; b^*] : t > 0\}$. Here we assume \mathbf{w}^* is nonzero. The parameter $\theta^* = \lfloor \mathbf{w}^*; b^* \rfloor$ is just a representative member of the set. Inhomogeneous linear learners are similar without b^* . We denote the corresponding “decision boundary” teaching dimension by $TD(\{\theta^*\}, \mathcal{A}_{opt})$. This notation extends our earlier definition of TD by allowing the first argument to be a set, with the understanding that the teaching goal is for the learned model to be an element in the set. It immediately follows that

$$TD(\{\theta^*\}, \mathcal{A}_{opt}) = \min_{t > 0} TD(t\theta^*, \mathcal{A}_{opt}).$$

Since it is sufficient to teach the parameter $t\theta^*$ for some $t > 0$ in order to teach the decision boundary, we can choose the best t that minimizes $TD(t\theta^*, \mathcal{A}_{opt})$. For SVM and logistic regression — either homogeneous or inhomogeneous — the teaching dimension $TD(t\theta^*, \mathcal{A}_{opt})$ depends on $\|\theta^*\|$ (see Table 1). We can choose t sufficiently small to drive down the teaching set size toward its possible minimum indicated by the LBI value in Table 2 (which is nonzero because of the ceiling function). Specifically, for any fixed parameter θ^* representing the target decision boundary:

- (homogeneous SVM): we choose $t \leq \frac{1}{\sqrt{\lambda \|\theta^*\|}}$ so that $TD(\{t\theta^*\}, \mathcal{A}_{svm}^{hom}) = 1$;

- (homogeneous logistic regression): we choose $t \leq \frac{\sqrt{\tau_{\max}}}{\sqrt{\lambda \|\theta^*\|}}$ so that $TD(\{t\theta^*\}, \mathcal{A}_{log}^{hom}) = 1$;
- (inhomogeneous SVM): we choose $t \leq \frac{\sqrt{2}}{\sqrt{\lambda \|\mathbf{w}^*\|}}$ so that $TD(\{t\theta^*\}, \mathcal{A}_{svm}^{inh}) = 2$ (note LBI=2 in Table 2);
- (inhomogeneous logistic regression): we choose $t \leq \frac{\sqrt{2\tau_{\max}}}{\sqrt{\lambda \|\mathbf{w}^*\|}}$ so that $TD(\{t\theta^*\}, \mathcal{A}_{log}^{inh}) = 2$ (note LBI=2 in Table 2).

The resulting teaching dimension $TD(\{t\theta^*\}, \mathcal{A}_{opt})$ is listed in Table 1 on the row marked by “decision boundary.” The teaching set construction is the same as in sections 3.2 and 3.3, respectively, but with θ^* .

5. Related Work

Teaching dimension as a learning-theoretic quantity has attracted a long history of research. It was proposed independently in Goldman and Kearns (1995); Shinohara and Miyano (1991). Subsequent theoretical developments can be found in e.g. Zilles et al. (2011); Balbach and Zeugmann (2009); Angluin (2004); Angluin and Krikis (1997); Goldman and Mathias (1996); Mathias (1997); Balbach and Zeugmann (2006); Balbach (2008); Kobayashi and Shinohara (2009); Angluin and Krikis (2003); Rivest and Yin (1995); Ben-David and Eiron (1998); Doliwa et al. (2014). Many of them assume little extra knowledge on the learner other than that it is consistent with the training data, though Zilles et al. (2011); Balbach (2008) allow the teacher and the learner to cooperate. These theoretically elegant teaching definitions diverge from the practice of modern machine learning where the learner solves an optimization problem to find a single model that is not necessarily the 0-1 loss ERM. Teaching such modern learners is our goal. Section 6 discusses a new view to unify our work and some existing optimal teaching work.

Teaching dimension is distinct from VC dimension. For a finite hypothesis space \mathcal{H} , Goldman and Kearns (1995) proved the relation

$$VC(\mathcal{H}) / \log(|\mathcal{H}|) \leq TD(\mathcal{H}) \leq VC(\mathcal{H}) + |\mathcal{H}| - 2^{VC(\mathcal{H})}.$$

These inequalities are somewhat weak, as Goldman and Kearns had shown both cases where one quantity is much larger than the other. The distinction between TD and VC dimension is also present in our setting. For example, by inspecting the inhomogeneous SVM column in Table 1 we note that TD does not depend on the dimensionality d of the feature space \mathbb{R}^d . To see why this makes intuitive sense, note two d -dimensional points are sufficient to specify any bisecting hyperplane in \mathbb{R}^d . On the other hand, recall that the VC dimension for inhomogeneous hyperplanes in \mathbb{R}^d is $d + 1$. Furthermore, there is an interesting connection to sample compression in Floyd and Warmuth (1995). Our teaching set can be viewed as the compressed sample, but with two generalizations: (i) the original “sample” is the whole input space, and (ii) the labels is allowed to diverge from the target model. Further quantification of these connections remains an open research question.

The teaching setting we considered is also distinct from active learning. In teaching the teacher knows the target model *a priori* and her goal is to *encode* the target model as a

training set, knowing that the decoder is special (namely a specific machine learning algorithm). This communication perspective highlights the difference to active learning, which must explore the hypothesis space to find the target model. Consequently, the teaching dimension can be dramatically smaller than the active learning query complexity for the same learner and hypothesis space. For example, Zhu (2013) demonstrated that to learn a 1D threshold classifier within ϵ error, the teaching dimension is a constant $\text{TD}=2$ regardless of ϵ , while active learning would require $O(\log \frac{1}{\epsilon})$ queries which can be arbitrarily larger than TD .

While the present paper focused on the theory of optimal teaching, there are practical applications, too. One such application is computer-aided personalized education. The human student is modeled by a computational cognitive model, or equivalently the learning algorithm. The educational goal is specified by the target model. The optimal teaching set is then well-defined, and represents the best personalized lesson for the student (Zhu, 2015, 2013; Khan et al., 2011). In one experiment, Patil *et al.* showed that real human students learn statistically significantly better under such optimal teaching set compared to an *i.i.d.* training set (Patil et al., 2014). Because contemporary cognitive models often employ optimization-based machine learners, our teaching dimension study helps to characterize these optimal lessons.

Another application of optimal teaching is in computer security. In particular, optimal teaching is the mathematical formalism to study the so-called data poisoning attacks (Barreno et al., 2010; Mei and Zhu, 2015a,b; Alfeld et al., 2016). Here the “teacher” is an attacker who has a nefarious target model in mind. The “student” is a learning agent (such as a spam filter) which accepts data and adapts itself. The attacker wants to minimally manipulate the input data in order to manipulate the learning agent toward the attacker’s target model. Teaching dimension quantifies the difficulty of data-poisoning attacks, and supports research on defenses.

Teaching dimension also has applications in interactive machine learning to quantify the minimum human interaction necessary (Suh et al., 2016; Cakmak and Thomaz, 2011), and in formal synthesis to generate computer programs satisfying a specification (Jha and Seshia, 2015).

6. A New View on Teaching

The optimal teaching literature has been cautious about the so-called collusion or coding tricks between the teacher and the learner. Nonetheless, what constitutes collusion does not have a fully satisfactory definition. Goldman and Mathias (1996) defined the teacher and the learner as collusion-free if (i) the teaching set is consistent with the target concept; (ii) any superset of the teaching set will make the learner learn the target concept, too. While this definition of collusion-free is useful, it does not capture all interesting learning behaviors. For example, Zilles et al. (2011, section 4) had to introduce a different notion of collusion in order to allow benign cooperation between the teacher and the learner. As another example, standard machine learning algorithms such as ridge regression does not satisfy either of the two properties: the teaching set (19) is inconsistent in that $y_t \neq \mathbf{x}_t^\top \boldsymbol{\theta}^*$, and adding more consistent training items will in general produce a different model due to regularization.

We advance an alternative view on the relation between the teacher and the learner. Under this view, the learner publishes his learning algorithm $\mathcal{A} : \mathbb{D} \rightarrow 2^{\mathcal{H}}$. Recall \mathcal{A} takes in a training set $D \in \mathbb{D}$ and outputs a subset of the hypothesis space \mathcal{H} . The teacher then uses a fixed strategy: she simply solves the training set cardinality minimization problem under the constraint that \mathcal{A} returns the target hypothesis set Θ^* . For example, to teach a specific parameter vector $\boldsymbol{\theta}^*$ the target is the singleton set $\Theta^* = \{\boldsymbol{\theta}^*\}$; to teach a decision boundary the target is the set $\Theta^* = \{\boldsymbol{\theta}^* \mid t > 0\}$. More precisely, the teacher’s strategy is to solve the following optimization problem, whose objective value is the (learner-dependent) teaching dimension $\text{TD}(\Theta^*, \mathcal{A})$:

$$\begin{aligned} \min_{D \in \mathbb{D}} \quad & |D| \\ \text{s.t.} \quad & \Theta^* = \mathcal{A}(D). \end{aligned} \tag{35}$$

Our teaching dimension for linear learners clearly fits this view, with \mathcal{A}_{opt} being a regularized empirical risk minimizer (1). Let us look at a few other interesting learners \mathcal{A} under this view. We will use the following hypothesis space as it is historically used to contrast those learners (Goldman and Kearns, 1995; Zilles et al., 2011). Let $\mathcal{X} = \{x_1, \dots, x_n\}$. Let $h_i(x) = 1$ if $x = x_i$ and 0 otherwise, for $i = 1 \dots n$. In other words, h_i is the indicator concept on x_i . Let the all-negative concept be $h_0(x) = 0$ for all x . Let $\mathcal{H} = \{h_0, h_1, \dots, h_n\}$.

- **The version-space learner \mathcal{A}_{vs} as defined by (2).** This is the learner behind the teaching dimension defined by Goldman and Kearns (1995). We have $\mathcal{A}_{vs}(\{(x_i, 1)\}) = \{h_i\}$ for $i = 1 \dots n$, such that these target concepts have classic teaching dimension $\text{TD}(h_i, \mathcal{A}_{vs}) = 1$. But note that $\mathcal{A}_{vs}(\{(x_i, 0)\}) = \{h_0, \dots, h_{i-1}, h_{i+1}, \dots\}$ which does not reduce the version space to a single element. To specify the all-negative concept we need $\mathcal{A}_{vs}(\{(x_1, 0), \dots, (x_n, 0)\}) = \{h_0\}$. That is, h_0 ’s classic teaching dimension is $\text{TD}(h_0, \mathcal{A}_{vs}) = n$. These teaching dimensions are the objective values in our view (35) when we plug in \mathcal{A}_{vs} .

- **The Balbach learner \mathcal{A}_B (Balbach, 2008).** Balbach noticed that h_1, \dots, h_n can each be taught with one item. The reasoning goes that as soon as the teaching set contains more than one item, it must be a helpful teacher’s hint that the target concept is h_0 . That is, the size of the training set carries useful information about the target concept. In the view of (35), we may define $\mathcal{A}_B(\{(x_i, 1)\}) = \{h_i\}$ for $i = 1 \dots n$, and $\mathcal{A}_B(\{(x_i, 0), (x_j, 0)\}) = \{h_0\}$ for any $i \neq j$. For the sake of completeness, here and below for all other $D \in \mathbb{D}$ not explicitly mentioned we simply define $\mathcal{A}(D) = \{h\}$ consistent with D . When we plug \mathcal{A}_B into (35) we obtain Balbach’s teaching dimension $\text{TD}(h_i, \mathcal{A}_B)$ of 1 for h_1, \dots, h_n , and 2 for h_0 .

- **The subset learner \mathcal{A}_s (Zilles et al., 2011).** Since the teaching sets for h_1, \dots, h_n each contain a positive item, it stands to reason that h_0 is the target concept as soon as a single negative training item is observed. We can define $\mathcal{A}_s(\{(x_i, 1)\}) = \{h_i\}$ and $\mathcal{A}_s(\{(x_i, 0)\}) = \{h_0\}$ for $i = 1 \dots n$. When we plug \mathcal{A}_s into (35) we obtain the subset teaching dimension of $\text{TD}(h_i, \mathcal{A}_s) = 1$ for all $h \in \mathcal{H}$, which is an improvement over the Balbach teaching dimension by a certain benign cooperation.

- **A coding-trick learner \mathcal{A}_{c1} .** This ‘learner’ uses x to encode hypothesis: $\mathcal{A}_{c1}(\{(x_i, y)\}) = \{h_i\}$ for $i = 1 \dots n$ regardless of y , and all non-singleton training set maps to h_0 : $\mathcal{A}_{c1}(D) = \{h_0\}$ if $|D| \neq 1$. \mathcal{A}_{c1} is mathematically well-defined for teaching in (35), but one can argue that it does not seem like a reasonable learner: it ignores y completely and thus is inconsistent (although recall modern regularized empirical risk minimizers (1) can be inconsistent, too).

- **Another coding-trick learner \mathcal{A}_{c2} .** This ‘learner’ uses training set size to encode the hypothesis, while ignoring the content of the training set: $\mathcal{A}_{c2}(D) = \{h_{|D|}\}$ if $|D| \leq n$, and \emptyset if $|D| > n$. Again, \mathcal{A}_{c2} is mathematically well-defined but does not seem like a reasonable learner.

As the examples above show, our alternative view of teaching in (35) does not resolve the issue of what constitutes coding-tricks. All the learners \mathcal{A} are well-defined functions mapping a training set to a subset of hypotheses, so that the optimization problem (35) is also well-defined even for ‘unreasonable’ learners like \mathcal{A}_{c1} and \mathcal{A}_{c2} . However, our alternative view does provide two benefits:

- Because the teacher employs a fixed strategy (35), this view removes the notion of ‘collusion’ altogether. Instead, the question becomes what learning algorithm \mathcal{A} one would consider as admissible. This view point can be more natural when we extend teaching to richer, more complex learners.
- There can be a misconception that the classic teaching dimension defined by Goldman and Kearns (1995) is learner-independent and a property of \mathcal{H} only, in part perhaps fueled by the original notation $TD(\mathcal{H})$. Our view highlights classic teaching dimension’s dependency on the version space learner \mathcal{A}_{vs} . It is true that \mathcal{A}_{vs} is a particularly simple and elegant learner with very nice properties. But, as others have observed (e.g. Balbach (2008); Zilles et al. (2011)), it does not capture all natural teaching and learning behaviors.

7. Conclusion

We have presented a generalization on teaching dimension to optimization-based learners. To the best of our knowledge, our teaching dimension for ridge regression, SVM, and logistic regression is new; so are the lower bounds and our analysis technique in general.

There are many possible extensions to the present work. For example, one may extend our analysis to nonlinear learners. This can potentially be achieved by using the kernel trick on the linear learners. As another example, one may allow ‘approximate teaching’ by relaxing the teaching goal, such that teaching is considered successful if the learner arrives at a model close enough to the target model. Taken together, the present paper and its extensions are expected to enrich our understanding of optimal teaching and enable novel applications.

Acknowledgments

The authors thank the editor and referees for their valuable comments. Special thanks to the production editor Dr. Charles Sutton for his help to prepare the final version of this paper. This work is supported in part by NSF grants CNS-1548078, IIS-0953219, DGE-1545481, and by the University of Wisconsin-Madison Graduate School with funding from the Wisconsin Alumni Research Foundation.

References

- S. Alfeld, X. Zhu, and P. Barford. Data poisoning attacks against autoregressive models. *AAAI*, 2016.
- D. Angluin. Queries revisited. *Theoretical Computer Science*, 313(2):175–194, 2004.
- D. Angluin and M. Krikis. Teachers, learners and black boxes. *COLT*, 1997.
- D. Angluin and M. Krikis. Learning from different teachers. *Machine Learning*, 51(2):137–163, 2003.
- F. J. Balbach. Measuring teachability using variants of the teaching dimension. *Theor. Comput. Sci.*, 397(1-3):94–113, 2008.
- F. J. Balbach and T. Zeugmann. Teaching randomized learners. *COLT*, pages 229–243, 2006.
- F. J. Balbach and T. Zeugmann. Recent developments in algorithmic teaching. In *Proceedings of the 3rd International Conference on Language and Automata Theory and Applications*, pages 1–18, 2009.
- M. Barreno, B. Nelson, A. D. Joseph, and J. D. Tygar. The security of machine learning. *Machine Learning Journal*, 81(2):121–148, 2010.
- S. Ben-David and N. Eiron. Self-directed learning and its relation to the VC-dimension and to teacher-directed learning. *Machine Learning*, 33(1):87–104, 1998.
- D. Bertsekas and A. Nedic. *Conver analysis and optimization (conservative)*. Athena Scientific, 2003.
- M. Cakmak and A. Thoma. Mixed-initiative active learning. *ICML Workshop on Combinating Learning Strategies to Reduce Label Cost*, 2011.
- R. M. Corless, G. H. Gomet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth. On the Lambert W function. *Advances in Computational Mathematics*, 5(1):329–359, 1996.
- T. Doiwa, G. Fan, H. U. Simon, and S. Zilles. Recursive teaching dimension, VC-dimension and sample compression. *Journal of Machine Learning Research*, 15:3107–3131, 2014.
- S. Floyd and M. Warmuth. Sample compression, learnability, and the Vapnik-Chervonenkis dimension. *Machine learning*, 21(3):269–304, 1995.
- S. Goldman and M. Kearns. On the complexity of teaching. *Journal of Computer and Systems Sciences*, 50(1):20–31, 1995.

- S. A. Goldman and H. D. Mathias. Teaching a smarter learner. *Journal of Computer and Systems Sciences*, 52(2):255–267, 1996.
- S. Jha and S. A. Seshia. A theory of formal synthesis via inductive learning. *CoRR*, 2015.
- F. Khan, X. Zhu, and B. Muthu. How do humans teach: On curriculum learning and teaching dimension. *NIPS*, 2011.
- H. Kobayashi and A. Shimohara. Complexity of teaching by a restricted number of examples. *COLT*, pages 293–302, 2009.
- H. David Mathias. A model of interactive teaching. *J. Comput. Syst. Sci.*, 54(3):487–501, 1997.
- S. Mei and X. Zhu. Using machine teaching to identify optimal training-set attacks on machine learners. *AAAI*, 2015a.
- S. Mei and X. Zhu. The security of latent Dirichlet allocation. *AISTATS*, 2015b.
- J. Nocedal and S. J. Wright. *Numerical Optimization (2nd edition)*. Springer, 2006.
- K. Patil, X. Zhu, L. Kopec, and B. C. Love. Optimal teaching for limited-capacity human learners. *NIPS*, 2014.
- R. L. Rivest and Y. L. Yin. Being taught can be faster than asking questions. *COLT*, 1995.
- A. Shinohara and S. Miyano. Teachability in computational learning. *New Generation Computing*, 8(4):337–348, 1991.
- J. Suh, X. Zhu, and S. Amershi. The label complexity of mixed-initiative classifier training. *ICML*, 2016.
- X. Zhu. Machine teaching for Bayesian learners in the exponential family. *NIPS*, 2013.
- X. Zhu. Machine teaching: an inverse problem to machine learning and an approach toward optimal education. *AAAI*, 2015.
- S. Zilles, S. Lange, R. Holte, and M. Zinkevich. Models of cooperative teaching and learning. *Journal of Machine Learning Research*, 12:349–384, 2011.

Augmentable Gamma Belief Networks

Mingyuan Zhou*

Department of Information, Risk, and Operations Management

McCombs School of Business

The University of Texas at Austin

Austin, TX 78712, USA

MINGYUAN.ZHOU@MCCOMBS.UTEXAS.EDU

Yulai Cong

Bo Chen*

National Laboratory of Radar Signal Processing

Collaborative Innovation Center of Information Sensing and Understanding

Xidian University

Xi'an, Shaanxi 710071, China

YULAI.CONG@163.COM

BCHEN@MAIL.XIDIAN.EDU.CN

Editor: Francois Caron

Abstract

To infer multilayer deep representations of high-dimensional discrete and nonnegative real vectors, we propose an augmentable gamma belief network (GBN) that factorizes each of its hidden layers into the product of a sparse connection weight matrix and the nonnegative real hidden units of the next layer. The GBN's hidden layers are jointly trained with an upward-downward Gibbs sampler that solves each layer with the same subroutine. The gamma-negative binomial process combined with a layer-wise training strategy allows inferring the width of each layer given a fixed budget on the width of the first layer. Example results illustrate interesting relationships between the width of the first layer and the inferred network structure, and demonstrate that the GBN can add more layers to improve its performance in both unsupervisedly extracting features and predicting heldout data. For exploratory data analysis, we extract trees and subnetworks from the learned deep network to visualize how the very specific factors discovered at the first hidden layer and the increasingly more general factors discovered at deeper hidden layers are related to each other, and we generate synthetic data by propagating random variables through the deep network from the top hidden layer back to the bottom data layer.

Keywords: Bayesian nonparametrics, deep learning, multilayer representation, Poisson factor analysis, topic modeling, unsupervised learning

1. Introduction

There has been significant recent interest in deep learning. Despite its tremendous success in supervised learning, inferring a multilayer data representation in an unsupervised manner remains a challenging problem (Bengio and LeCun, 2007; Ranzato et al., 2007; Bengio et al., 2015). To generate data with a deep network, it is often unclear how to set the structure of the network, including the depth (number of layers) of the network and the width (number of hidden units) of each layer. In addition, for some commonly used deep

generative models, including the sigmoid belief network (SBN), deep belief network (DBN), and deep Boltzmann machine (DBM), the hidden units are often restricted to be binary. More specifically, the SBN, which connects the binary units of adjacent layers via the sigmoid functions, infers a deep representation of multivariate binary vectors (Neal, 1992; Saul et al., 1996); the DBN (Hinton et al., 2006) is a SBN whose top hidden layer is replaced by the restricted Boltzmann machine (RBM) (Hinton, 2002) that is undirected; and the DBM is an undirected deep network that connects the binary units of adjacent layers using the RBMs (Salakhutdinov and Hinton, 2009). All these three deep networks are designed to model binary observations, without principled ways to set the network structure. Although one may modify the bottom layer to model Gaussian and multinomial observations, the hidden units of these networks are still typically restricted to be binary (Salakhutdinov and Hinton, 2009; Larochelle and Lauly, 2012; Salakhutdinov et al., 2013). To generalize these models, one may consider the exponential family harmoniums (Welling et al., 2004; King et al., 2005) to construct more general networks with non-binary hidden units, but often at the expense of noticeably increased complexity in training and data fitting. To model real-valued data without restricting the hidden units to be binary, one may consider the general framework of nonlinear Gaussian belief networks (Frey and Hinton, 1999) that constructs continuous hidden units by nonlinearly transforming Gaussian distributed latent variables, including as special cases both the continuous SBN of Frey (1997a,b) and the rectified Gaussian nets of Hinton and Ghahramani (1997). More recent scalable generalizations under that framework include variational auto-encoders (Kingma and Welling, 2014) and deep latent Gaussian models (Rezende et al., 2014).

Moving beyond conventional deep generative models using binary or nonlinearly transformed Gaussian hidden units and setting the network structure in a heuristic manner, we construct deep networks using gamma distributed nonnegative real hidden units, and combine the gamma-negative binomial process (Zhou and Carin, 2015; Zhou et al., 2015b) with a greedy-layer wise training strategy to automatically infer the network structure. The proposed model is called the augmentable gamma belief network, referred to hereafter for brevity as the GBN, which factorizes the observed or latent count vectors under the Poisson likelihood into the product of a factor loading matrix and the gamma distributed hidden units (factor scores) of layer one; and further factorizes the shape parameters of the gamma hidden units of each layer into the product of a connection weight matrix and the gamma hidden units of the next layer. The GBN together with Poisson factor analysis can unsupervisedly infer a multilayer representation from multivariate count vectors, with a simple but powerful mechanism to capture the correlations between the visible/hidden features across all layers and handle highly overdispersed counts. With the Bernoulli-Poisson link function (Zhou, 2015), the GBN is further applied to high-dimensional sparse binary vectors by truncating latent counts, and with a Poisson randomized gamma distribution, the GBN is further applied to high-dimensional sparse nonnegative real data by randomizing the gamma shape parameters with latent counts.

For tractable inference of a deep generative model, one often applies either a sampling based procedure (Neal, 1992; Frey, 1997a) or variational inference (Saul et al., 1996; Frey, 1997b; Ranganath et al., 2014b; Kingma and Welling, 2014). However, conjugate priors on the model parameters that connect adjacent layers are often unknown, making it difficult to develop fully Bayesian inference that infers the posterior distributions of these parameters.

*. Correspondence should be addressed to M. Zhou or B. Chen.

It was not until recently that a Gibbs sampling algorithm, imposing priors on the network connection weights and sampling from their conditional posteriors, was developed for the SBN by Gan et al. (2015b), using the Pólya-Gamma data augmentation technique developed for logistic models (Polson et al., 2012). In this paper, we will develop data augmentation technique unique for the augmentable GBN, allowing us to develop a fully Bayesian upward-downward Gibbs sampling algorithm to infer the posterior distributions of not only the hidden units, but also the connection weights between adjacent layers.

Distinct from previous deep networks that often require tuning both the width (number of hidden units) of each layer and the network depth (number of layers), the GBN employs nonnegative real hidden units and automatically infers the widths of subsequent layers given a fixed budget on the width of its first layer. Note that the budget could be infinite and hence the whole network can grow without bound as more data are being observed. Similar to other belief networks that can often be improved by adding more hidden layers (Hinton et al., 2006; Sutskever and Hinton, 2008; Bengio et al., 2015), for the proposed model, when the budget on the first layer is finite and hence the ultimate capacity of the network could be limited, our experimental results also show that a GBN equipped with a narrower first layer could increase its depth to match or even outperform a shallower one with a substantially wider first layer.

The gamma distribution density function has the highly desired strong non-linearity for deep learning, but the existence of neither a conjugate prior nor a closed-form maximum likelihood estimate (Choi and Wette, 1969) for its shape parameter makes a deep network with gamma hidden units appear unattractive. Despite seemingly difficult, we discover that, by generalizing the data augmentation and marginalization techniques for discrete data modeled with the Poisson, gamma, and negative binomial distributions (Zhou and Carin, 2015), one may propagate latent counts one layer at a time from the bottom data layer to the top hidden layer, with which one may derive an efficient upward-downward Gibbs sampler that, one layer at a time in each iteration, upward samples Dirichlet distributed connection weight vectors and then downward samples gamma distributed hidden units, with the latent parameters of each layer solved with the same subroutine.

With extensive experiments in text and image analysis, we demonstrate that the deep GBN with two or more hidden layers clearly outperforms the shallow GBN with a single hidden layer in both unsupervisedly extracting latent features for classification and predicting heldout data. Moreover, we demonstrate the excellent ability of the GBN in exploratory data analysis: by extracting trees and subnetworks from the learned deep network, we can follow the paths of each tree to visualize various aspects of the data, from very general to very specific, and understand how they are related to each other.

In addition to constructing a new deep network that well fits high-dimensional sparse binary, count, and nonnegative real data, developing an efficient upward-downward Gibbs sampler, and applying the learned deep network for exploratory data analysis, other contributions of the paper include: 1) proposing novel link functions, 2) combining the gamma-negative binomial process (Zhou and Carin, 2015; Zhou et al., 2015b) with a layer-wise training strategy to automatically infer the network structure; 3) revealing the relationship between the upper bound imposed on the width of the first layer and the inferred widths of subsequent layers; 4) revealing the relationship between the depth of the network and the model’s ability to model overdispersed counts; and 5) generating multivariate high-

dimensional discrete or nonnegative real vectors, whose distributions are governed by the GBN, by propagating the gamma hidden units of the top hidden layer back to the bottom data layer. We note this paper significantly extends our recent conference publication (Zhou et al., 2015a) that proposes the Poisson GBN.

2. Augmentable Gamma Belief Networks

Denoting $\theta_j^{(t)} \in \mathbb{R}_+^{K_t}$ as the K_t hidden units of sample j at layer t , where $\mathbb{R}_+ = \{x : x \geq 0\}$, the generative model of the augmentable gamma belief network (GBN) with T hidden layers, from top to bottom, is expressed as

$$\begin{aligned} \theta_j^{(T)} &\sim \text{Gam}\left(\mathbf{r}, 1/c_j^{(T+1)}\right), \\ &\vdots \\ \theta_j^{(t)} &\sim \text{Gam}\left(\Phi^{(t+1)}\theta_j^{(t+1)}, 1/c_j^{(t+1)}\right), \\ &\vdots \\ \theta_j^{(1)} &\sim \text{Gam}\left(\Phi^{(2)}\theta_j^{(2)}, p_j^{(2)}/(1-p_j^{(2)})\right), \end{aligned} \quad (1)$$

where $x \sim \text{Gam}(a, 1/c)$ represents a gamma distribution with mean a/c and variance a/c^2 . For $t = 1, 2, \dots, T-1$, the GBN factorizes the shape parameters of the gamma distributed hidden units $\theta_j^{(t)} \in \mathbb{R}_+^{K_t}$ of layer t into the product of the connection weight matrix $\Phi^{(t+1)} \in \mathbb{R}_+^{K_t \times K_{t+1}}$ and the hidden units $\theta_j^{(t+1)} \in \mathbb{R}_+^{K_{t+1}}$ of layer $t+1$; the top layer’s hidden units $\theta_j^{(T)}$ share the same vector $\mathbf{r} = (r_1, \dots, r_{K_T})^\top$ as their gamma shape parameters; and the $p_j^{(2)}$ are probability parameters and $\{1/c^{(t)}\}_{3:T+1}$ are gamma scale parameters, with $c_j^{(2)} := (1-p_j^{(2)})/p_j^{(2)}$. We will discuss later how to measure the connection strengths between the nodes of adjacent layers and the overall popularity of a factor at a particular hidden layer.

For scale identifiability and ease of inference and interpretation, each column of $\Phi^{(t)} \in \mathbb{R}_+^{K_{t-1} \times K_t}$ is restricted to have a unit L_1 norm and hence $0 \leq \Phi^{(t)}(k', k) \leq 1$. To complete the hierarchical model, for $t \in \{1, \dots, T-1\}$, we let

$$\phi_k^{(t)} \sim \text{Dir}(\eta^{(t)}, \dots, \eta^{(t)}), \quad \tau_k \sim \text{Gam}(\gamma_0/K_T, 1/c_0) \quad (2)$$

where $\phi_k^{(t)} \in \mathbb{R}_+^{K_{t-1}}$ is the k th column of $\Phi^{(t)}$; we impose $c_0 \sim \text{Gam}(e_0, 1/f_0)$ and $\gamma_0 \sim \text{Gam}(a_0, 1/b_0)$; and for $t \in \{3, \dots, T+1\}$, we let

$$p_j^{(2)} \sim \text{Beta}(a_0, b_0), \quad c_j^{(t)} \sim \text{Gam}(e_0, 1/f_0). \quad (3)$$

We expect the correlations between the K_t rows (latent features) of $(\theta_1^{(t)}, \dots, \theta_j^{(t)})$ to be captured by the columns of $\Phi^{(t+1)}$. Even if $\Phi^{(t)}$ for $t \geq 2$ are all identity matrices, indicating no correlations between the latent features to be captured, our analysis in Section 3.2 will show that a deep structure with $T \geq 2$ could still benefit data fitting by better modeling the variability of the latent features $\theta_j^{(1)}$. Before further examining the network structure, below we first introduce a set of distributions that will be used to either model different types of data or augment the model for simple inference.

2.1 Distributions for Count, Binary, and Nonnegative Real Data

Below we first describe some useful count distributions that will be used later.

2.1.1 USEFUL COUNT DISTRIBUTIONS AND THEIR RELATIONSHIPS

Let the Chinese restaurant table (CRT) distribution $l \sim \text{CRT}(n, r)$ represent the random number of tables seated by n customers in a Chinese restaurant process (Blackwell and MacQueen, 1973; Antoniak, 1974; Aldous, 1985; Pitman, 2006) with concentration parameter r . Its probability mass function (PMF) can be expressed as

$$P(l | n, r) = \frac{\Gamma(r)^r}{\Gamma(n+r)} |s(n, l)|,$$

where $l \in \mathbb{Z}$, $\mathbb{Z} := \{0, 1, \dots, n\}$, and $|s(n, l)|$ are unsigned Stirling numbers of the first kind. A CRT distributed sample can be generated by taking the summation of n independent Bernoulli random variables as

$$l = \sum_{i=1}^n b_i, \quad b_i \sim \text{Bernoulli}[r/(r+i-1)].$$

Let $u \sim \text{Log}(p)$ denote the logarithmic distribution (Fisher et al., 1943; Anscombe, 1950; Johnson et al., 1997) with PMF

$$P(u | p) = \frac{1}{-\ln(1-p)} \frac{p^u}{u},$$

where $u \in \{1, 2, \dots\}$, and let $n \sim \text{NB}(r, p)$ denote the negative binomial (NB) distribution (Greenwood and Yule, 1920; Bliss and Fisher, 1953) with PMF

$$P(n | r, p) = \frac{\Gamma(n+r)}{n! \Gamma(r)} p^r (1-p)^r,$$

where $n \in \mathbb{Z}$. The NB distribution $n \sim \text{NB}(r, p)$ can be generated as a gamma mixed Poisson distribution as

$$n \sim \text{Pois}(\lambda), \quad \lambda \sim \text{Gam}[r, p/(1-p)],$$

where $p/(1-p)$ is the gamma scale parameter.

As shown in (Zhou and Carin, 2015), the joint distribution of n and l given r and p in

$$l \sim \text{CRT}(n, r), \quad n \sim \text{NB}(r, p),$$

where $l \in \{0, \dots, n\}$ and $n \in \mathbb{Z}$, is the same as that in

$$n = \sum_{t=1}^l u_t, \quad u_t \sim \text{Log}(p), \quad l \sim \text{Pois}[-r \ln(1-p)], \quad (4)$$

which is called the Poisson-logarithmic bivariate distribution, with PMF

$$P(n, l | r, p) = \frac{|s(n, l)| r^l}{n!} p^n (1-p)^r.$$

We will exploit these relationships to derive efficient inference for the proposed models.

2.1.2 BERNOULLI-POISSON LINK AND TRUNCATED POISSON DISTRIBUTION

As in Zhou (2015), the Bernoulli-Poisson (BerPo) link thresholds a random count at one to obtain a binary variable as

$$b = \mathbf{1}(m \geq 1), \quad m \sim \text{Pois}(\lambda), \quad (5)$$

where $b = 1$ if $m \geq 1$ and $b = 0$ if $m = 0$. If m is marginalized out from (5), then given λ , one obtains a Bernoulli random variable as $b \sim \text{Ber}(1 - e^{-\lambda})$. The conditional posterior of the latent count m can be expressed as

$$(m | b, \lambda) \sim b \cdot \text{Pois}_+(\lambda),$$

where $x \sim \text{Pois}_+(\lambda)$ follows a truncated Poisson distribution, with $P(x = k) = (1 - e^{-\lambda})^{-1} \lambda^k e^{-\lambda} / k!$ for $k \in \{1, 2, \dots\}$. Thus if $b = 0$, then $m = 0$ almost surely (a.s.), and if $b = 1$, then $m \sim \text{Pois}_+(\lambda)$, which can be simulated with a rejection sampler that has a minimal acceptance rate of 63.2% at $\lambda = 1$ (Zhou, 2015). Given the latent count m and a gamma prior on λ , one can then update λ using the gamma-Poisson conjugacy. The BerPo link shares some similarities with the probit link that thresholds a normal random variable at zero, and the logistic link that lets $b \sim \text{Ber}[e^x / (1 + e^x)]$. We advocate the BerPo link as an alternative to the probit and logistic links since if $b = 0$, then $m = 0$ a.s., which could lead to significant computational savings if the binary vectors are sparse. In addition, the conjugacy between the gamma and Poisson distributions makes it convenient to construct hierarchical Bayesian models amenable to posterior simulation.

2.1.3 POISSON RANDOMIZED GAMMA AND TRUNCATED BESSEL DISTRIBUTIONS

To model nonnegative data that include both zeros and positive observations, we introduce the Poisson randomized gamma (PRG) distribution as

$$x \sim \text{PRG}(\lambda, c),$$

whose distribution has a point mass at $x = 0$ and is continuous for $x > 0$. The PRG distribution is generated as a Poisson mixed gamma distribution as

$$x \sim \text{Gam}(n, 1/c), \quad n \sim \text{Pois}(\lambda),$$

in which we define $\text{Gam}(0, 1/c) = 0$ a.s. and hence $x = 0$ if and only if $n = 0$. Thus the PMF of $x \sim \text{PRG}(\lambda, c)$ can be expressed as

$$\begin{aligned} f_X(x | \lambda, c) &= \sum_{n=0}^{\infty} \text{Gam}(x; n, 1/c) \text{Pois}(n; \lambda) \\ &= (e^{-\lambda})^{\mathbf{1}(x=0)} \left[e^{-\lambda - cx} \sqrt{\frac{\lambda c}{x}} I_{-1} \left(2\sqrt{\lambda cx} \right) \right]^{\mathbf{1}(x>0)}, \end{aligned} \quad (6)$$

where

$$I_{-1}(\alpha) = \left(\frac{\alpha}{2} \right)^{-1} \sum_{n=1}^{\infty} \frac{\left(\frac{\alpha^2}{4} \right)^n}{n! \Gamma(n)}, \quad \alpha > 0$$

is the modified Bessel function of the first kind $I_\nu(\alpha)$ with ν fixed at -1 . Using the laws of total expectation and total variance, or using the PMF directly, one may show that

$$\mathbb{E}[x | \lambda, c] = \lambda/c, \quad \text{var}[x | \lambda, c] = 2\lambda/c^2.$$

Thus the variance-to-mean ratio of the PRG distribution is $2/c$, as controlled by c .

The conditional posterior of n given x , λ , and c can be expressed as

$$\begin{aligned} f_N(n | x, \lambda, c) &= \frac{\text{Gam}(x; n, 1/c) \text{Pois}(n; \lambda)}{\text{PRG}(x; \lambda, c)} \\ &= \mathbf{1}(x = 0) \delta_0 + \mathbf{1}(x > 0) \sum_{n=1}^{\infty} \frac{1}{L_{-1}(2\sqrt{\lambda cx})} \frac{(\lambda cx)^{n-\frac{1}{2}}}{n! \Gamma(n)} \delta_n \\ &= \mathbf{1}(x = 0) \delta_0 + \mathbf{1}(x > 0) \sum_{n=1}^{\infty} \text{Bessel}_{-1}(n; 2\sqrt{cx\lambda}) \delta_n, \end{aligned} \quad (7)$$

where we define $n \sim \text{Bessel}_{-1}(\alpha)$ as the truncated Bessel distribution, with PMF

$$\text{Bessel}_{-1}(n; \alpha) = \frac{\left(\frac{\alpha}{2}\right)^{2n-1}}{L_{-1}(\alpha) n! \Gamma(n)}, \quad n \in \{1, 2, \dots\}.$$

Thus $n = 0$ if and only if $x = 0$, and n is a positive integer drawn from a truncated Bessel distribution if $x > 0$. In Appendix A, we plot the probability distribution functions of the proposed PRG and truncated Bessel distributions and show how they differ from the randomized gamma and Bessel distributions (Yuan and Kalbfleisch, 2000), respectively.

2.2 Link Functions for Three Different Types of Observations

If the observations are multivariate count vectors $\mathbf{x}_j^{(1)} \in \mathbb{Z}^V$, where $V := K_0$, then we link the integer-valued visible units to the nonnegative real hidden units at layer one using Poisson factor analysis (PFA) as

$$\mathbf{x}_j^{(1)} \sim \text{Pois}\left(\Phi^{(1)} \theta_j^{(1)}\right). \quad (8)$$

Under this construction, the correlations between the K_0 rows (features) of $(\mathbf{x}_1^{(1)}, \dots, \mathbf{x}_J^{(1)})$ are captured by the columns of $\Phi^{(1)}$. Detailed descriptions on how PFA is related to a wide variety of discrete latent variable models, including nonnegative matrix factorization (Lee and Seung, 2001), latent Dirichlet allocation (Blei et al., 2003), the gamma-Poisson model (Canny, 2004), discrete Principal component analysis (Buntine and Jakulin, 2006), and the focused topic model (Williamson et al., 2010), can be found in Zhou et al. (2012) and Zhou and Carin (2015).

We call PFA using the GBN in (1) as the prior on its factor scores as the Poisson gamma belief network (PGBN), as proposed in Zhou et al. (2015a). The PGBN can be naturally applied to factorize the term-document frequency count matrix of a text corpus, not only extracting semantically meaningful topics at multiple layers, but also capturing the relationships between the topics of different layers using the deep network, as discussed below in both Sections 2.3 and 4.

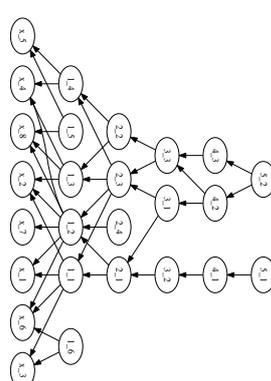


Figure 1: An example directed network of five hidden layers, with $K_0 = 8$ visible units, $[K_1, K_2, K_3, K_4, K_5] = [6, 4, 3, 3, 2]$, and sparse connections between the units of adjacent layers.

If the observations are high-dimensional sparse binary vectors $\mathbf{b}_j^{(1)} \in \{0, 1\}^V$, then we factorize them using Bernoulli-Poisson factor analysis (Ber-PFA) as

$$\mathbf{b}_j^{(1)} = \mathbf{1}(\mathbf{x}_j^{(1)} \geq 0), \quad \mathbf{x}_j^{(1)} \sim \text{Pois}\left(\Phi^{(1)} \theta_j^{(1)}\right). \quad (9)$$

We call Ber-PFA with the augmentable GBN as the prior on its factor scores $\theta_j^{(1)}$ as the Bernoulli-Poisson gamma belief network (BerPo-GBN).

If the observations are high-dimensional sparse nonnegative real-valued vectors $\mathbf{y}_j^{(1)} \in \mathbb{R}_+^V$, then we factorize them using Poisson randomized gamma (PRG) factor analysis as

$$\mathbf{y}_j^{(1)} \sim \text{Gam}(\mathbf{x}_j^{(1)}, 1/q_j), \quad \mathbf{x}_j^{(1)} \sim \text{Pois}\left(\Phi^{(1)} \theta_j^{(1)}\right). \quad (10)$$

We call PRG factor analysis with the augmentable GBN as the prior on its factor scores $\theta_j^{(1)}$ as the PRG gamma belief network (PRG-GBN).

We show in Figure 1 an example directed belief network of five hidden layers, with $K_0 = 8$ visible units, with 6, 4, 3, 3, and 2 hidden units for layers one, two, three, four, and five, respectively, and with sparse connections between the units of adjacent layers.

2.3 Exploratory Data Analysis

To interpret the network structure of the GBN, we notice that

$$\mathbb{E}[\mathbf{x}_j^{(1)} | \theta_j^{(1)}, \{\mathbf{c}_j^{(l)}, c_j^{(l)}\}_{l=1}^t] = \left[\prod_{l=1}^t \Phi^{(l)} \right] \frac{\theta_j^{(1)}}{\prod_{l=2}^t c_j^{(l)}}, \quad (11)$$

$$\mathbb{E}[\theta_j^{(1)} | \{\Phi^{(l)}, c_j^{(l)}\}_{l=1}^T, \mathbf{r}] = \left[\prod_{l=1}^T \Phi^{(l)} \right] \frac{\mathbf{r}}{\prod_{l=1}^{T-1} c_j^{(l)}}. \quad (12)$$

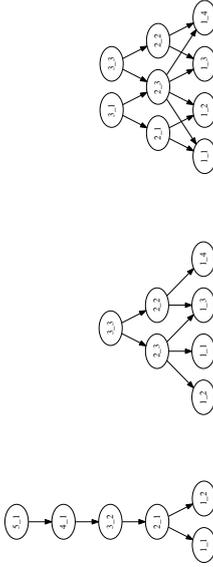


Figure 2: Extracted from the network shown in Figure 1, the left plot is a tree rooted at node 5.1, the middle plot is a tree rooted at node 3.3, and the right plot is a subnetwork consisting of both the tree rooted at node 3.1 and the tree rooted at node 3.3.

Thus for visualization, it is straightforward to project the K_t topics/hidden units/factor loadings/nodes of layer $t \in \{1, \dots, T\}$ to the bottom data layer as the columns of the $V \times K_t$ matrix

$$\prod_{\ell=1}^t \Phi^{(\ell)}, \quad (13)$$

and rank their popularities using the K_t dimensional nonnegative weight vector

$$\mathbf{r}^{(t)} := \left[\prod_{\ell=t+1}^T \Phi^{(\ell)} \right] \mathbf{r}. \quad (14)$$

To measure the connection strength between node k of layer t and node k' of layer $t-1$, we use the value of

$$\Phi^{(t)}(k', k),$$

which is also expressed as $\phi_k^{(t)}(k')$ or $\phi_{k'k}^{(t)}$.

Our intuition is that examining the nodes of the hidden layers, via their projections to the bottom data layer, from the top to bottom layers will gradually reveal less general and more specific aspects of the data. To verify this intuition and further understand the relationships between the general and specific aspects of the data, we consider extracting a tree for each node of layer t , where $t \geq 2$, to help visualize the inferred multilayer deep structure. To be more specific, to construct a tree rooted at a node of layer t , we grow the tree downward by linking the root node (if at layer t) or each leaf node of the tree (if at a layer below layer t) to all the nodes at the layer below that are connected to the root/leaf node with non-negligible weights. Note that a tree in our definition permits a node to have more than one parent, which means that different branches of the tree can overlap with each other. In addition, we also consider extracting subnetworks, each of which consists of multiple related trees from the full deep network. For example, shown in the left of Figure 2 is the tree extracted from the network in Figure 1 using node 5.1 as the root, shown in the middle is the tree using node 3.3 as the root, and shown in the right is a subnetwork consisting of two related trees that are rooted at nodes 3.1 and 3.3, respectively.

2.3.1 VISUALIZING NODES OF DIFFERENT LAYERS

Before presenting the technical details, we first provide some example results obtained with the PGBN on extracting multilayer representations from the 11,269 training documents of the 20newsgroups data set (<http://qwone.com/~jason/20Newsgroups/>). Given a fixed budget of $K_1^{\max} = 800$ on the width of the first layer, with $\eta^{(t)} = 0.1$ for all t , a five-layer deep network inferred by the PGBN has a network structure as $[K_1, K_2, K_3, K_4, K_5] = [386, 63, 58, 54, 51]$, meaning that there are 386, 63, 58, 54, and 51 nodes at layers one to five, respectively.

For visualization, we first relabel the nodes at each layer based on their weights $\{r_k^{(t)}\}_{1, K_t}$, calculated as in (14), with a more popular (larger weight) node assigned with a smaller label. We visualize node k of layer t by displaying its top 12 words ranked according to their probabilities in $(\prod_{\ell=1}^{t-1} \Phi^{(\ell)}) \phi_k^{(t)}$, the k th column of the projected representation calculated as in (13). We set the font size of node k of layer t proportional to $(r_k^{(t)}/r_1^{(t)})^{1/5}$ in each subplot, and color the outside border of a text box as red, green, orange, blue, or black for a node of layer five, four, three, two, or one, respectively. For better interpretation, we also exclude from the vocabulary the top 30 words of node 1 of layer one: “don just like people think know time good make way does writes edu ve want say really article use right did things point going better thing need sure used little,” and the top 20 words of node 2 of layer one: “edu writes article com apr cs ca just know don like think news cc david university john org wrote world.” These 50 words are not in the standard list of stopwords but can be considered as stopwords specific to the 20newsgroups corpus discovered by the PGBN.

For the [386, 63, 58, 54, 51] PGBN learned on the 20newsgroups corpus, we plot 54 example topics of layer one in Figure 3, the top 30 topics of layer three in Figure 4, and the top 30 topics of layer five in Figure 5. Figure 3 clearly shows that the topics of layer one, except for topics 1-3 that mainly consist of common functional words of the corpus, are all very specific. For example, topics 71 and 81 shown in the first row are about “candida yeast symptoms” and “sex,” respectively; topics 53, 73, 83, and 84 shown in the second row are about “printer,” “msg,” “police radar detector,” and “Canadian health care system,” respectively, and topics 46 and 76 shown in third row are about “ice hockey” and “second amendment,” respectively. By contrast, the topics of layers three and five, shown in Figures 4 and 5, respectively, are much less specific and can in general be matched to one or two news groups out of the 20 news groups, including comp.{graphics, os.ms-windows.misc, sys.ibm.pc.hardware, sys.mac.hardware, windows.x}, rec.{autos, motorcycles}, rec.sport.{baseball, hockey}, sci.{crypt, electronics, med, space}, misc.forsale, talk.politics.{misc, guns, mideast}, and {talk.religion.misc, alt.atheism, soc.religion.christian}.

2.3.2 VISUALIZING TREES ROOTED AT THE TOP-LAYER HIDDEN UNITS

While it is interesting to examine the topics of different layers to understand the general and specific aspects of the corpus used to train the PGBN, it would be more informative to further illustrate how the topics of different layers are related to each other. Thus we consider constructing trees to visualize the PGBN. We first pick a node as the root of a tree and grow the tree downward by drawing a line from node k at layer t , the root or a leaf node of the tree, to node k' at layer $t-1$ for all k' in the set $\{k' : \Phi^{(t)}(k', k) > \tau_t/K_{t-1}\}$,

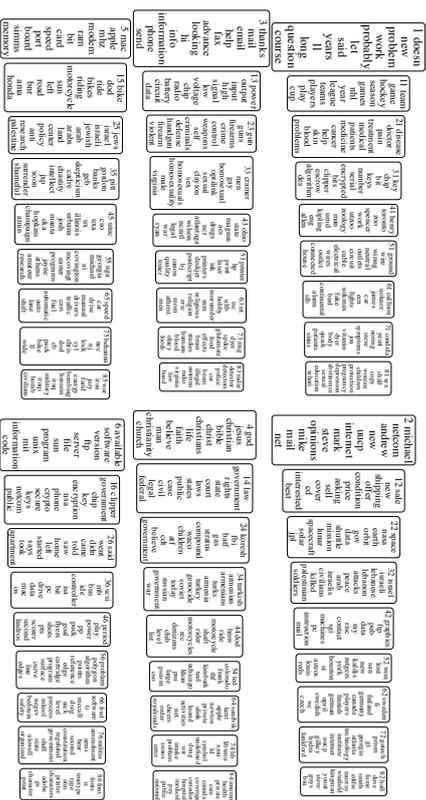


Figure 3: Example topics of layer one of the PGBN trained on the 20newsgrroups corpus.



Figure 4: The top 30 topics of layer three of the PGBN trained on the 20newsgrroups corpus.

Figure 5: The top 30 topics of layer five of the PGBN trained on the 20newsgrroups corpus.

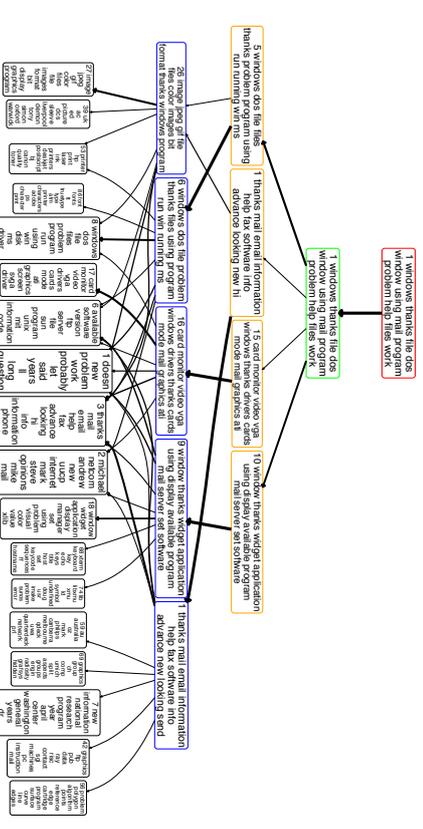


Figure 6: A [18, 5, 4, 1, 1] tree that includes all the lower-layer nodes (directly or indirectly) linked with non-negligible weights to the top ranked node of the top layer, taken from the full [386, 63, 58, 54, 51] network inferred by the PGBN on the 11,269 training documents of the 20newsgrroups corpus, with $\eta^{(t)} = 0.1$ for all t . A line from node k at layer t to node k' at layer $t - 1$ indicates that $\Phi^{(t)}(k', k) > 3/K_{t-1}$, with the width of the line proportional to $\sqrt{\Phi^{(t)}(k', k)}$. For each node, the rank (in terms of popularity) at the corresponding layer and the top 12 words of the corresponding topic are displayed inside the text box, where the text font size monotonically decreases as the popularity of the node decreases, and the outside border of the text box is colored as red, green, orange, blue, or black if the node is at layer five, four, three, two, or one, respectively.

where we set the width of the line connecting node k of layer t to node k' of layer $t - 1$ be proportional to $\sqrt{\Phi^{(t)}(k', k)}$ and use τ_t to adjust the complexity of a tree. In general, increasing τ_t would discard more weak connections and hence make the tree simpler and easier to visualize.

We set $\tau_t = 3$ for all t to visualize both a five-layer tree rooted at the top ranked node of the top hidden layer, as shown in Figure 6, and a five-layer tree rooted at the second ranked node of the top hidden layer, as shown in Figure 7. For the tree in Figure 6, while it is somewhat vague to determine the actual meanings of both node 1 of layer five and node 1 of layer four based on their top words, examining the more specific topics of layers three and two within the tree clearly indicate that this tree is primarily about “windows,” “window system,” “graphics,” “information,” and “software,” which are relatively specific concepts that are all closely related to each other. The similarities and differences between the five nodes of layer two can be further understood by examining the nodes of layer one that are connected to them. For example, while nodes 26 and 16 of layer two share their connections to multiple nodes of layer one, node 27 of layer one on “image” is strongly connected to node 26 of layer two but not to node 16 of layer two, and node 17 of layer one on “video” is strongly connected to node 16 of layer two but not to node 26 of layer two.

The bottom subplot reveals that in layer one, topic 14 on “law & government,” topic 32 on “Israel & Lebanon,” topic 34 on “Turkey, Armenia, Soviet Union, & Russian,” topic 132 on “Greece, Turkey, & Cyprus,” topic 98 on “Bosnia, Serbs, & Muslims,” topic 143 on “Armenia, Azeris, Cyprus, Turkey, & Karabakh,” and several other very specific topics related to Turkey and/or Armenia all tend to co-occur with each other.

We note that capturing the co-occurrence patterns between the topics not only helps exploratory data analysis, but also helps extract better features for classification in an unsupervised manner and improves prediction for held-out data, as will be demonstrated in detail in Section 4.

2.4 Related Models

The structure of the augmentable GBN resembles the sigmoid belief network and recently proposed deep exponential family model (Ranganath et al., 2014b). Such kind of gamma distribution based network and its inference procedure were vaguely hinted in Corollary 2 of Zhou and Carin (2015), and had been exploited by Acharya et al. (2015) to develop a gamma Markov chain to model the temporal evolution of the factor scores of a dynamic count matrix, but have not yet been investigated for extracting multilayer data representations. The proposed augmentable GBN may also be considered as an exponential family harmonium (Welling et al., 2004; Xing et al., 2005).

2.4.1 SIGMOID AND DEEP BELIEF NETWORKS

Under the hierarchical model in (1), given the connection weight matrices, the joint distribution of the observed/latent counts and gamma hidden units of the GBN can be expressed, similar to those of the sigmoid and deep belief networks (Bengio et al., 2015), as

$$P(\mathbf{x}_j^{(1)}, \{\theta_j^{(t)}\}_t | \{\Phi^{(t)}\}_t) = P(\mathbf{x}_j^{(1)} | \Phi^{(1)}, \theta_j^{(1)}) \left[\prod_{t=2}^{T-1} P(\theta_j^{(t)} | \Phi^{(t+1)}, \theta_j^{(t+1)}) \right] P(\theta_j^{(T)}).$$

With ϕ_{vj} representing the v th row Φ , for the gamma hidden units $\theta_{vj}^{(t)}$ we have

$$P(\theta_{vj}^{(t)} | \phi_{vj}^{(t+1)}, \theta_j^{(t+1)}, c_{j+1}^{(t+1)}) = \frac{\binom{c_{j+1}^{(t+1)}}{\theta_{vj}^{(t+1)}} \phi_{vj}^{(t+1)} \theta_j^{(t+1)}}{\Gamma(\phi_{vj}^{(t+1)}) \theta_j^{(t+1)}} \binom{\theta_{vj}^{(t+1)}}{\theta_{vj}^{(t)}} \phi_{vj}^{(t+1)} \theta_j^{(t+1)-1} e^{-\phi_{vj}^{(t+1)} \theta_{vj}^{(t)}}, \quad (19)$$

which are highly nonlinear functions that are strongly desired in deep learning. By contrast, with the sigmoid function $\sigma(x) = 1/(1+e^{-x})$ and bias terms $b_v^{(t+1)}$, a sigmoid/deep belief network would connect the binary hidden units $\theta_{vj}^{(t)} \in \{0, 1\}$ of layer t (for deep belief networks, $t < T-1$) to the product of the connection weights and binary hidden units of the next layer with

$$P(\theta_{vj}^{(t)} = 1 | \phi_{vj}^{(t+1)}, \theta_j^{(t+1)}, b_v^{(t+1)}) = \sigma(b_v^{(t+1)} + \phi_{vj}^{(t+1)} \theta_j^{(t+1)}). \quad (20)$$

Comparing (19) with (20) clearly shows the distinctions between the gamma distributed nonnegative hidden units and the sigmoid link function based binary hidden units. The

limitation of binary units in capturing the approximately linear data structure over small ranges is a key motivation for Frey and Hinton (1999) to investigate nonlinear Gaussian belief networks with real-valued units. As a new alternative to binary units, it would be interesting to further investigate whether the gamma distributed nonnegative real units can in theory carry richer information and model more complex nonlinearities given the same network structure. Note that the rectified linear units have emerged as powerful alternatives of sigmoid units to introduce nonlinearity (Nair and Hinton, 2010). It would be interesting to investigate whether the gamma units can be used to introduce nonlinearity into the positive region of the rectified linear units.

2.4.2 DEEP POISSON FACTOR ANALYSIS

With $T = 1$, the PGBN specified by (1)-(3) and (8) reduces to Poisson factor analysis (PFA) using the (truncated) gamma-negative binomial process (Zhou and Carin, 2015), with a truncation level of K_1 . As discussed in (Zhou et al., 2012; Zhou and Carin, 2015), with priors imposed on neither $\phi_k^{(1)}$ nor $\theta_j^{(1)}$, PFA is related to nonnegative matrix factorization (Lee and Seung, 2001), and with the Dirichlet priors imposed on both $\phi_k^{(1)}$ and $\theta_j^{(1)}$, PFA is related to latent Dirichlet allocation (Blei et al., 2003).

Related to the PGBN and the dynamic model in (Acharya et al., 2015), the deep exponential family model of Ranganath et al. (2014b) also considers a gamma chain under Poisson observations, but it is the gamma scale parameters that are chained and factorized, which allows learning the network parameters using black box variational inference (Ranganath et al., 2014a). In the proposed PGBN, we chain the gamma random variables via the gamma shape parameters. Both strategies worth through investigation. We prefer chaining the shape parameters in this paper, which leads to efficient upward-downward Gibbs sampling via data augmentation and makes it clear how the latent counts are propagated across layers, as discussed in detail in the following sections. The sigmoid belief network has also been recently incorporated into PFA for deep factorization of count data (Gan et al., 2015a), however, that deep structure captures only the correlations between binary factor usage patterns but not the full connection weights. In addition, neither Ranganath et al. (2014b) nor Gan et al. (2015a) provide a principled way to learn the network structure, whereas the proposed GBN uses the gamma-negative binomial process together with a greedy layer-wise training strategy to automatically infer the widths of the hidden layers, which will be described in Section 3.3.

2.4.3 CORRELATED AND TREE-STRUCTURED TOPIC MODELS

The PGBN with $T = 2$ can also be related to correlated topic models (Blei and Lafferty, 2006; Paisley et al., 2012; Chen et al., 2013; Ranganath and Blei, 2015; Linderman et al., 2015), which typically use the logistic normal distributions to replace the topic-proportion Dirichlet distributions used in latent Dirichlet allocation (Blei et al., 2003), capturing the co-occurrence patterns between the topics in the latent Gaussian space using a covariance matrix. By contrast, the PGBN factorizes the topic usage weights (not proportions) under the gamma likelihood, capturing the co-occurrence patterns between the topics of the first layer (i.e., the columns of $\Phi^{(1)}$) in the columns of $\Phi^{(2)}$, the latent weight matrix connecting the hidden units of layers two and one. For the PGBN, the computation does not involve

matrix inversion, which is often necessary for correlated topic models without specially structured covariance matrices, and scales linearly with the number of topics, hence it is suitable to be used to capture the correlations between hundreds of or thousands of topics.

As in Figures 6, 7, and 17-21, trees and subnetworks can be extracted from the inferred deep network to visualize the data. Tree-structured topic models have also been proposed before, such as those in Blei et al. (2010), Adams et al. (2010), and Paisley et al. (2015), but they usually artificially impose the tree structures to be learned, whereas the PGBN learns a directed network, from which trees and subnetworks can be extracted for visualization, without the need to specify the number of nodes per layer, restrict the number of branches per node, and forbid a node to have multiple parents.

3. Model Properties and Inference

Inference for the GBN shown in (1) appears challenging, because not only the conjugate prior is unknown for the shape parameter of a gamma distribution, but also the gradients are difficult to evaluate for the parameters of the (log) gamma probability density function, which, as in (19), includes the parameters inside the (log) gamma function. To address these challenges, we consider data augmentation (van Dyk and Meng, 2001) that introduces auxiliary variables to make it simple to compute the conditional posteriors of model parameters via the joint distribution of the auxiliary and existing random variables. We will first show that each gamma hidden unit can be linked to a Poisson distributed latent count variable, leading to a negative binomial likelihood for the parameters of the gamma hidden unit if it is margined out from the Poisson distribution; we then introduce an auxiliary count variable, which is sampled from the CRT distribution parametrized by the negative binomial latent count and shape parameter, to make the joint likelihood of the auxiliary CRT count and latent negative binomial count given the parameters of the gamma hidden unit amenable to posterior simulation. More specifically, under the proposed augmentation scheme, the gamma shape parameters will be linked to auxiliary counts under the Poisson likelihoods, making it straightforward for posterior simulation, as described below in detail.

3.1 The Upward Propagation of Latent Counts

We break the inference of the GBN of T hidden layers into T related subproblems, each of which is solved with the same subroutine. Thus for implementation, it is straightforward for the GBN to adjust its depth T . Let us denote $\mathbf{x}_j^{(t)} \in \mathbb{Z}^{K_{t-1}}$ as the observed or latent count vector of layer $t \in \{1, \dots, T\}$, and $x_{vj}^{(t)}$ as its v th element, where $v \in \{1, \dots, K_{t-1}\}$.

Lemma 1 (Augment-and-Conquer The Gamma Belief Network) *With $p_j^{(1)} := 1 - e^{-1}$ and*

$$p_j^{(t+1)} := -\ln(1 - p_j^{(t)}) / \left[c_j^{(t+1)} - \ln(1 - p_j^{(t)}) \right] \quad (21)$$

for $t = 1, \dots, T$, one may connect the observed or latent counts $\mathbf{x}_j^{(t)} \in \mathbb{Z}^{K_{t-1}}$ to the product $\Phi^{(t)}(\theta_j^{(t)})$ at layer t under the Poisson likelihood as

$$\mathbf{x}_j^{(t)} \sim \text{Pois} \left[-\Phi^{(t)}(\theta_j^{(t)}) \ln(1 - p_j^{(t)}) \right]. \quad (22)$$

Proof By definition (22) is true for layer $t = 1$. Suppose that (22) is also true for layer $t > 1$, then we can augment each count $x_{vj}^{(t)}$, where $v \in \{1, \dots, K_{t-1}\}$, into the summation of K_t latent counts, which are smaller than or equal to $x_{vj}^{(t)}$ as

$$x_{vj}^{(t)} = \sum_{k=1}^{K_t} x_{vjk}^{(t)}, \quad x_{vjk}^{(t)} \sim \text{Pois} \left[-\phi_{vk}^{(t)} \theta_{kj}^{(t)} \ln(1 - p_j^{(t)}) \right]. \quad (23)$$

Let the \cdot symbol represent summing over the corresponding index and let

$$m_{kj}^{(t)(t+1)} := x_{\cdot jk}^{(t)} := \sum_{v=1}^{K_{t-1}} x_{vjk}^{(t)}$$

represent the number of times that factor $k \in \{1, \dots, K_t\}$ of layer t appears in observation j and $\mathbf{m}_j^{(t)(t+1)} := (x_{\cdot j1}^{(t)}, \dots, x_{\cdot jK_t}^{(t)})'$. Since $\sum_{v=1}^{K_{t-1}} \phi_{vk}^{(t)} = 1$, we can marginalize out $\Phi^{(t)}$ as in (Zhou et al., 2012), leading to

$$\mathbf{m}_j^{(t)(t+1)} \sim \text{Pois} \left[-\theta_j^{(t)} \ln(1 - p_j^{(t)}) \right].$$

Further marginalizing out the gamma distributed $\theta_j^{(t)}$ from the Poisson likelihood leads to

$$m_j^{(t)(t+1)} \sim \text{NB} \left(\Phi^{(t+1)} \theta_j^{(t+1)}, p_j^{(t+1)} \right). \quad (24)$$

Element k of $\mathbf{m}_j^{(t)(t+1)}$ can be augmented under its compound Poisson representation as

$$m_{kj}^{(t)(t+1)} = \sum_{\ell=1}^{x_{\cdot jk}^{(t+1)}} u_\ell, \quad u_\ell \sim \text{Log}(p_j^{(t+1)}), \quad x_{kj}^{(t+1)} \sim \text{Pois} \left[-\phi_{k\cdot}^{(t+1)} \theta_j^{(t+1)} \ln(1 - p_j^{(t+1)}) \right].$$

Thus if (22) is true for layer t , then it is also true for layer $t+1$. ■

Corollary 2 (Propagate the latent counts upward) *Using Lemma 4.1 of (Zhou et al., 2012) on (23) and Theorem 1 of (Zhou and Carin, 2015) on (24), we can propagate the latent counts $x_{vj}^{(t)}$ of layer t upward to layer $t+1$ as*

$$\left\{ \left(x_{vj1}^{(t)}, \dots, x_{vjK_t}^{(t)} \mid x_{vj}^{(t)}, \phi_{v\cdot}^{(t)}, \theta_j^{(t)} \right) \sim \text{Mult} \left(x_{vj}^{(t)}, \frac{\phi_{v1}^{(t)} \theta_{1j}^{(t)}}{\sum_{k=1}^{K_t} \phi_{vk}^{(t)} \theta_{kj}^{(t)}}, \dots, \frac{\phi_{vK_t}^{(t)} \theta_{K_t j}^{(t)}}{\sum_{k=1}^{K_t} \phi_{vk}^{(t)} \theta_{kj}^{(t)}} \right), \quad (25) \right. \\ \left. \left(x_{kj}^{(t+1)} \mid m_{kj}^{(t)(t+1)}, \phi_{k\cdot}^{(t+1)}, \theta_j^{(t+1)} \right) \sim \text{CRT} \left(m_{kj}^{(t)(t+1)}, \phi_{k\cdot}^{(t+1)} \theta_j^{(t+1)} \right). \quad (26) \right.$$

We provide a set of graphical representations in Figure 8 to describe the GBN model and illustrate the augment-and-conquer inference scheme. We provide the upward-downward Gibbs sampler in Appendix B.

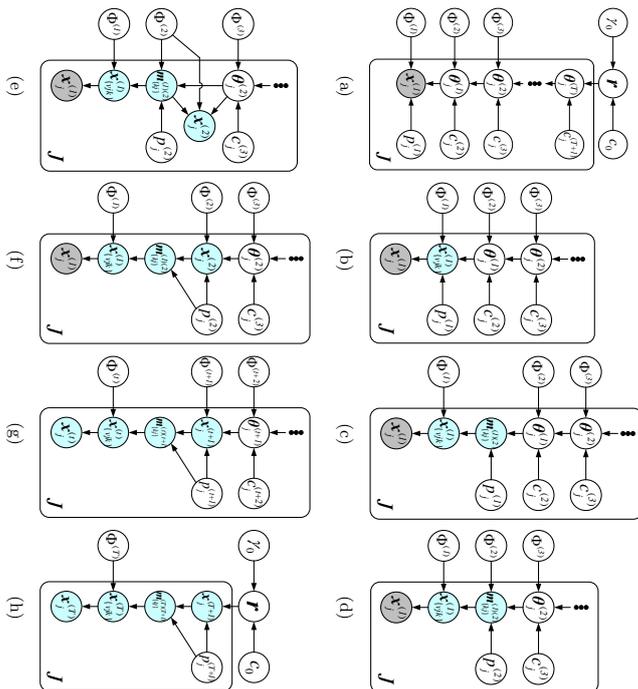


Figure 8: Graphical representations of the model and data augmentation and marginalization based inference scheme. (a) graphical representation of the GBN hierarchical model. (b) an augmented representation of Poisson factor model of layer $t = 1$, corresponding to (23) with $t = 1$. (c) an alternative representation using the relationships between the Poisson and multinomial distributions, obtained by applying Lemma 4.1 of Zhou et al., 2012) on (23) for $t = 1$. (d) a negative binomial distribution based representation that marginalizes out the gamma from the Poisson distributions, corresponding to (24) for $t = 1$. (e) an equivalent representation that introduces CRT distributed auxiliary variables, corresponding to (26) with $t = 1$. (f) an equivalent representation using Theorem 1 of Zhou and Cain, 2015) on (24) and (26) for $t = 1$. (g) An representation obtained by repeating the same augmentation-marginalization steps described in (b)-(f) one layer at a time from layers 1 to t . (h) An representation of the top hidden layer.

Note that $x_{k_j}^{(t)} = m_{k_j}^{(t)(t+1)}$, and as the number of tables occupied by the customers is in the same order as the logarithm of the customer number in a Chinese restaurant process, $x_{k_j}^{(t+1)}$ is in the same order as $\ln(m_{k_j}^{(t)(t+1)})$. Thus the total count of layer $t + 1$ as $\sum_j x_{k_j}^{(t+1)}$ would often be much smaller than that of layer t as $\sum_j x_{k_j}^{(t)}$ (though in general not as small as a count that is in the same order of the logarithm of $\sum_j x_{k_j}^{(t)}$), and hence one may use the total count $\sum_j x_{k_j}^{(T)}$ as a simple criterion to decide whether it is necessary to add more

layers to the GBN. In addition, if the latent count $x_{k',k}^{(t)}$:= $\sum_j x_{k_j}^{(t)k}$ becomes close or equal to zero, then the posterior mean of $\Phi^{(t)}(k', k)$ could become so small that node k' of layer $t - 1$ can be considered to be disconnected from node k of layer t .

3.2 Modeling Data Variability With Distributed Representation

In comparison to a single-layer model with $T = 1$, which assumes that the hidden units of layer one are independent in the prior, the multilayer model with $T \geq 2$ captures the correlations between them. Note that for the extreme case that $\Phi^{(t)} = \mathbf{I}_{K_t}$ for $t \geq 2$ are all identity matrices, which indicates that there are no correlations between the features of $\theta_j^{(t-1)}$ left to be captured, the deep structure could still provide benefits as it helps model latent counts $m_{k_j}^{(1)(2)}$ that may be highly overdispersed. For example, let us assume $\Phi^{(t)} = \mathbf{I}_{K_3}$ for all $t \geq 2$, then from (1) and (24) we have

$$m_{k_j}^{(1)(2)} \sim \text{NB}(\theta_{k_j}^{(2)}, p_j^{(2)}), \dots, \theta_{k_j}^{(t)} \sim \text{Gam}(\theta_{k_j}^{(t+1)}, 1/c_j^{(t+1)}), \dots, \theta_{k_j}^{(T)} \sim \text{Gam}(r_k, 1/c_j^{(T+1)}).$$

Using the laws of total expectation and total variance, we have

$$\mathbb{E}[\theta_{k_j}^{(2)} | r_k] = \frac{r_k}{\prod_{l=3}^{T+1} c_j^{(l)}}, \quad \text{var}[\theta_{k_j}^{(2)} | r_k] = r_k \sum_{l=3}^{T+1} \left[\prod_{l=3}^l (c_j^{(l)})^{-2} \right] \left[\prod_{l=l+1}^{T+1} (c_j^{(l)})^{-1} \right].$$

Further applying the same laws, we have

$$\mathbb{E}[m_{k_j}^{(1)(2)} | r_k] = \frac{r_k p_j^{(2)}}{(1 - p_j^{(2)}) \prod_{l=3}^{T+1} c_j^{(l)}},$$

$$\text{var}[m_{k_j}^{(1)(2)} | r_k] = \frac{r_k p_j^{(2)}}{(1 - p_j^{(2)})^2 \prod_{l=3}^{T+1} c_j^{(l)}} \left\{ 1 + p_j^{(2)} \sum_{l=3}^{T+1} \left[\prod_{l=3}^l (c_j^{(l)})^{-1} \right] \right\}.$$

Thus the variance-to-mean ratio (VMR) of the count $m_{k_j}^{(1)(2)}$ given r_k can be expressed as

$$\text{VMR}[m_{k_j}^{(1)(2)} | r_k] = \frac{1}{(1 - p_j^{(2)})} \left\{ 1 + p_j^{(2)} \sum_{l=3}^{T+1} \left[\prod_{l=3}^l (c_j^{(l)})^{-1} \right] \right\}. \quad (27)$$

In comparison to PFA with $m_{k_j}^{(1)(2)} \sim \text{NB}(r_k, p_j^{(2)})$ given r_k , with a VMR of $1/(1 - p_j^{(2)})$, the GBN with T hidden layers, which mixes the shape of $m_{k_j}^{(1)(2)} \sim \text{NB}(\theta_{k_j}^{(2)}, p_j^{(2)})$ with a chain of gamma random variables, increases $\text{VMR}[m_{k_j}^{(1)(2)} | r_k]$ by a factor of

$$\frac{1 + p_j^{(2)} \sum_{l=3}^{T+1} \left[\prod_{l=3}^l (c_j^{(l)})^{-1} \right]}{1 + (T - 1)p_j^{(2)}}$$

which is equal to

if we further assume $c_j^{(l)} = 1$ for all $l \geq 3$. Therefore, by increasing the depth of the network to distribute the variability into more layers, the multilayer structure could increase its capacity to model data variability.

3.3 Learning The Network Structure With Layer-Wise Training

As jointly training all layers together is often difficult, existing deep networks are typically trained using a greedy layer-wise unsupervised training algorithm, such as the one proposed in (Hinton et al., 2006) to train the deep belief networks. The effectiveness of this training strategy is further analyzed in (Bengio et al., 2007). By contrast, the augmentable GBN has a simple Gibbs sampler to jointly train all its hidden layers, as described in Appendix B, and hence does not necessarily require greedy layer-wise training, but the same as these commonly used deep learning algorithms, it still needs to specify the number of layers and the width of each layer.

In this paper, we adopt the idea of layer-wise training for the GBN, not because of the lack of an effective joint-training algorithm that trains all layers together in each iteration, but for the purpose of learning the width of each hidden layer in a greedy layer-wise manner, given a fixed budget on the width of the first layer. The basic idea is to first train a GBN with a single hidden layer, *i.e.*, $T = 1$, for which we know how to use the gamma-negative binomial process (Zhou and Carin, 2015; Zhou et al., 2015b) to infer the posterior distribution of the number of active factors; we fix the width of the first layer K_1 with the number of active factors inferred at iteration B_1 , prune all inactive factors of the first layer, and continue Gibbs sampling for another C_1 iterations. Now we describe the proposed recursive procedure to build a GBN with $T \geq 2$ layers. With a GBN of $T - 1$ hidden layers that has already been inferred, for which the hidden units of the top layer are distributed as $\theta_j^{(T-1)} \sim \text{Gam}(r, 1/c_j^{(T)})$, where $\mathbf{r} = (r_1, \dots, r_{K_{T-1}})'$, we add another layer by letting $\theta_j^{(T-1)} \sim \text{Gam}(\Phi^{(T)} \theta_j^{(T)}, 1/c_j^{(T)})$, $\theta_j^{(T)} \sim \text{Gam}(r, 1/c_j^{(T+1)})$, where $\Phi^{(T)} \in \mathbb{R}_+^{K_{T-1} \times K_{T \max}^{(T+1)}}$ and \mathbf{r} is redefined as $\mathbf{r} = (r_1, \dots, r_{K_{T \max}^{(T)}})'$. The key idea is with latent counts $m_{kj}^{(T)/(T+1)}$ upward propagated from the bottom data layer, one may marginalize out $\theta_{kj}^{(T)}$, leading to $m_{kj}^{(T)/(T+1)} \sim \text{NB}(r_k, p_j^{(T+1)})$, $r_k \sim \text{Gam}(\gamma_0/K_{T \max}, 1/c_0)$, and hence can again rely on the shrinkage mechanism of a truncated gamma-negative binomial process to prune inactive factors (connection weight vectors, columns of $\Phi^{(T)}$) of layer T , making K_T the inferred layer width for the newly added layer, smaller than $K_{T \max}$ if $K_{T \max}$ is set to be sufficiently large. The newly added layer and all the layers below would be jointly trained, but with the structure below the newly added layer kept unchanged. Note that when $T = 1$, the GBN infers the number of active factors if $K_1 \max$ is set large enough, otherwise, it still assigns the factors with different weights r_k , but may not be able to prune any of them. The details of the proposed layer-wise training strategies are summarized in Algorithm 1 for multivariate count data, and in Algorithm 2 for multivariate binary and nonnegative real data.

4. Experimental Results

In this section, we present experimental results for count, binary, and nonnegative real data.

4.1 Deep Topic Modeling

We first analyze multivariate count data with the Poisson gamma belief network (PGBN). We apply the PGBNs for topic modeling of text corpora, each document of which is represented as a term-frequency count vector. Note that the PGBN with a single hidden layer

is identical to the (truncated) gamma-negative binomial process PFA of Zhou and Carin (2015), which is a nonparametric Bayesian algorithm that performs similarly to the hierarchical Dirichlet process latent Dirichlet allocation of Teh et al. (2006) for text analysis, and is considered as a strong baseline. Thus we will focus on making comparison to the PGBN with a single layer, with its layer width set to be large to approximate the performance of the gamma-negative binomial process PFA. We evaluate the PGBNs' performance by examining both how well they unsupervisedly extract low-dimensional features for document classification, and how well they predict heldout word tokens. Matlab code will be available in <http://mingyuanzhou.github.io/>.

We use Algorithm 1 to learn, in a layer-wise manner, from the training data the connection weight matrices $\Phi^{(1)}, \dots, \Phi^{(T \max)}$ and the top-layer hidden units' gamma shape parameters \mathbf{r} : to add layer T to a previously trained network with $T - 1$ layers, we use B_T iterations to jointly train $\Phi^{(T)}$ and \mathbf{r} together with $\{\Phi^{(t)}\}_{1, T-1}$, prune the inactive factors of layer T , and continue the joint training with another C_T iterations. We set the hyper-parameters as $a_0 = b_0 = 0.01$ and $e_0 = f_0 = 1$. Given the trained network, we apply the upward-downward Gibbs sampler to collect 500 MCMC samples after 500 burnins to estimate the posterior mean of the feature usage proportion vector $\theta_j^{(1)}/\theta_j^{(1)}$ at the first hidden layer, for every document in both the training and testing sets.

4.1.1 FEATURE LEARNING FOR BINARY CLASSIFICATION

We consider the 20newsgroups data set that consists of 18,774 documents from 20 different news groups, with a vocabulary of size $K_0 = 61,188$. It is partitioned into a training set of 11,269 documents and a testing set of 7,505 ones. We first consider two binary classification tasks that distinguish between the *comp.sys.ibm.pc.hardware* and *comp.sys.mac.hardware*, and between the *sci.electronics* and *sci.med* news groups. For each binary classification task, we remove a standard list of stop words and only consider the terms that appear at least five times, and report the classification accuracies based on 12 independent random trials. With the upper bound of the first layer's width set as $K_1 \max \in \{25, 50, 100, 200, 400, 600, 800\}$, and $B_t = C_t = 1000$ and $\eta^{(t)} = 0.01$ for all t , we use Algorithm 1 to train a network with $T \in \{1, 2, \dots, 8\}$ layers. Denote θ_j as the estimated K_1 dimensional feature vector for document j , where $K_1 \leq K_1 \max$ is the inferred number of active factors of the first layer that is bounded by the pre-specified truncation level $K_1 \max$. We use the L_2 regularized logistic regression provided by the LIBLINEAR package (Fan et al., 2008) to train a linear classifier on θ_j in the training set and use it to classify θ_j in the test set, where the regularization parameter is five-fold cross-validated on the training set from $(2^{-10}, 2^{-9}, \dots, 2^{15})$.

As shown in Figure 9, modifying the PGBN from a single-layer shallow network to a multilayer deep one clearly improves the qualities of the unsupervisedly extracted feature vectors. In a random trial, with $K_1 \max = 800$, we infer a network structure of $[K_1, \dots, K_8] = [512, 154, 75, 54, 47, 37, 34, 29]$ for the first binary classification task, and $[K_1, \dots, K_8] = [491, 143, 74, 49, 36, 32, 28, 26]$ for the second one. Figures 9(c)-(d) also show that increasing the network depth in general improves the performance, but the first-layer width clearly plays a critical role in controlling the ultimate network capacity. This insight is further illustrated below.

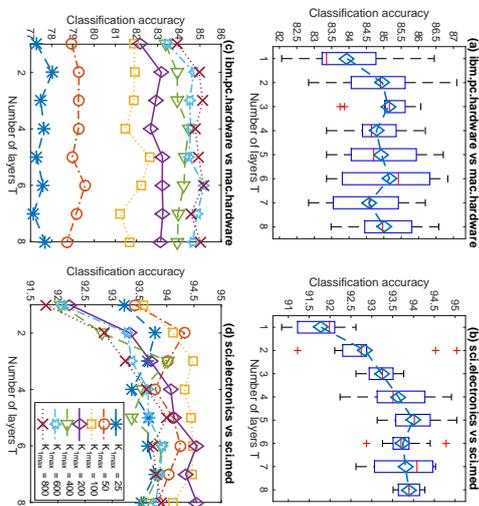


Figure 9: Classification accuracy (%) as a function of the network depth T for two 20news groups binary classification tasks, with $\eta^{(t)} = 0.01$ for all layers. (a)-(b): the boxplots of the accuracies of 12 independent runs with $K_1^{\max} = 800$. (c)-(d): the average accuracies of these 12 runs for various K_1^{\max} and T . Note that $K_1^{\max} = 800$ is large enough to cover all active first-layer topics (inferred to be around 500 for both binary classification tasks), whereas all the first-layer topics would be used if $K_1^{\max} = 25, 50, 100$, or 200.

4.1.2 FEATURE LEARNING FOR MULTI-CLASS CLASSIFICATION

We test the PGBNs for multi-class classification on 20news groups. After removing a standard list of stopwords and the terms that appear less than five times, we obtain a vocabulary with $V = 33,420$. We set $C_t = 500$ and $\eta^{(t)} = 0.05$ for all t ; we set $B_t = 1000$ for all t if $K_1^{\max} \leq 400$, and set $B_t = 1000$ and $B_t = 500$ for $t \geq 2$ if $K_1^{\max} > 400$. We use all 11,269 training documents to infer a set of networks with $T_{\max} \in \{1, \dots, 5\}$ and $K_1^{\max} \in \{50, 100, 200, 400, 600, 800\}$, and mimic the same testing procedure used for binary classification to extract low-dimensional feature vectors, with which each testing document is classified to one of the 20 news groups using the L_2 regularized logistic regression.

Figure 10 shows a clear trend of improvement in classification accuracy by increasing the network depth with a limited first-layer width, or by increasing the upper bound of the width of the first layer with the depth fixed. For example, a single-layer PGBN with $K_1^{\max} = 100$ could add one or more layers to slightly outperform a single-layer PGBN with $K_1^{\max} = 200$, and a single-layer PGBN with $K_1^{\max} = 200$ could add layers to clearly outperform a single-layer PGBN with K_1^{\max} as large as 800.

The proposed Gibbs sampler also exhibits several desirable computational properties. Each iteration of jointly training multiple layers usually only costs moderately more than that of training a single layer, e.g., with $K_1^{\max} = 400$, a training iteration on a single core of an Intel Xeon 2.7 GHz CPU takes about 5.6, 6.7, 7.1 seconds for the PGBN with 1, 3,

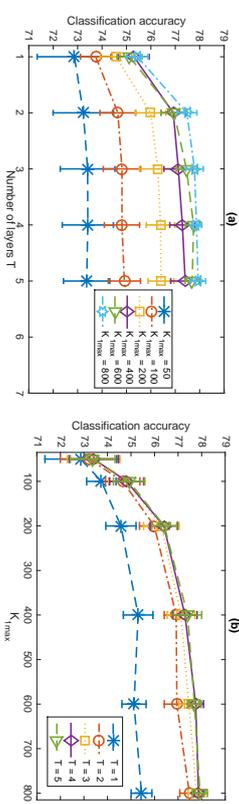


Figure 10: Classification accuracy (%) of the PGBNs with Algorithm 1 for 20news groups multi-class classification (a) as a function of the depth T with various K_1^{\max} and (b) as a function of K_1^{\max} with various depths, with $\eta^{(t)} = 0.05$ for all layers. The widths of the hidden layers are automatically inferred. In a random trial, the inferred network widths $[K_1, \dots, K_5]$ for $K_1^{\max} = 50, 100, 200, 400, 600$, and 800 are $[50, 50, 50, 50, 50]$, $[100, 99, 99, 94, 87]$, $[200, 161, 130, 94, 63]$, $[396, 109, 99, 82, 68]$, $[528, 129, 109, 98, 91]$, and $[608, 100, 99, 96, 89]$, respectively.

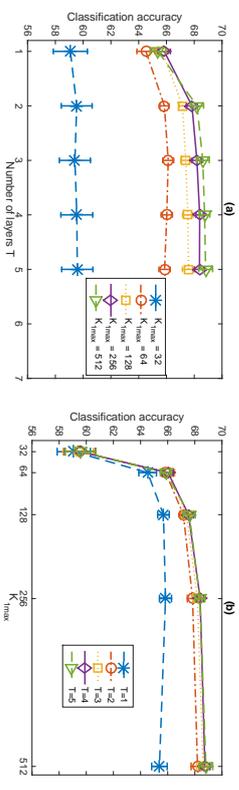


Figure 11: Analogous plots to Figure 10 with the vocabulary size restricted to be 2000, including the most frequent 2000 terms after removing a standard list of stopwords. The widths of the hidden layers are automatically inferred. In a random trial, the inferred network widths $[K_1, \dots, K_5]$ for $K_1^{\max} = 32, 64, 128, 256$, and 512 are $[32, 32, 32, 32, 32]$, $[64, 64, 64, 59, 59]$, $[128, 125, 118, 106, 87]$, $[256, 224, 124, 83, 65]$, and $[512, 187, 89, 78, 62]$, respectively.

and 5 layers, respectively. Since the per iteration cost increases approximately as a linear function of the inferred K_1 and as a linear function of the size of the data set, given a fixed computational budget, one may choose a moderate K_1^{\max} to allow adding a sufficiently large number of hidden layers. In addition, the samplings of $x^{(t)}$, $\phi_k^{(t)}$, and $\theta_{kj}^{(t)}$ in each layer can all be made embarrassingly parallel with blocked Gibbs sampling, and hence can potentially significantly benefit from implementing the algorithm using graphics processing units (GPUs) or other parallel computing architectures.

Examining the inferred network structure also reveals interesting details. For example, in a random trial with Algorithm 1, with $\eta^{(t)} = 0.05$ for all t , the inferred network widths $[K_1, \dots, K_5]$ for $K_1^{\max} = 50, 100, 200, 400, 600$, and 800 are $[50, 50, 50, 50, 50]$, $[100, 99, 99, 94, 87]$, $[200, 161, 130, 94, 63]$, $[396, 109, 99, 82, 68]$, $[528, 129, 109, 98, 91]$, and $[608,$

100, 99, 96, 89], respectively. This indicates that for a network with an insufficient budget on its first-layer width, as the network depth increases, its inferred layer widths decay more slowly than a network with a sufficient or surplus budget on its first-layer width; and a network with a surplus budget on its first-layer width may only need relatively small widths for its higher hidden layers.

In order to make comparison to related algorithms, we also consider restricting the vocabulary to the 2000 most frequent terms of the vocabulary after moving a standard list of stopwords. We repeat the same experiments with the same settings except that we set $K_{1\max} \in \{32, 64, 128, 256, 512\}$, $B_1 = 1000$, $C_1 = 500$, and $B_t = C_t = 500$ for all $t \geq 2$. We show the results in Figure 11. Again, we observe a clear trend of improvement by increasing the network depth with a limited first-layer width, or by increasing the upper bound of the width of the first layer with the depth fixed. In a random trial with Algorithm 1, the inferred network widths $[K_1, \dots, K_5]$ for $K_{1\max} = 32, 64, 128, 256$, and 512 are $[32, 32, 32, 32, 32]$, $[64, 64, 64, 59, 59]$, $[128, 125, 118, 106, 87]$, $[256, 224, 124, 83, 65]$, and $[512, 187, 89, 78, 62]$, respectively.

For comparison, we first consider the same L_2 regularized logistic regression multi-class classifier, trained either on the raw word counts or normalized term-frequencies of the 20newsgroups training documents using five-fold cross-validation. As summarized in Table 1 of Appendix C, when using the raw term-frequency word counts as covariates, the same classifier achieves 69.8% (68.2%) accuracy on the 20newsgroups test documents if using the top 2000 terms that exclude (include) a standard list of stopwords, achieves 75.8% if using all the 61, 188 terms in the vocabulary, and achieves 78.0% if using the 33, 420 terms remained after removing a standard list of stopwords and the terms that appear less than five times; and when using the normalized term-frequencies as covariates, the corresponding accuracies are 70.8% (67.9%) if using the top 2000 terms excluding (including) stopwords, 77.6% with all the 61, 188 terms, and 79.4% with the 33, 420 selected terms.

As summarized in Table 2 of Appendix C, for multi-class classification on the same data set, with a vocabulary size of 2000 that consists of the 2000 most frequent terms after removing stopwords and stemming, the DocNADE (Laroche and Laully, 2012) and the over-replicated softmax (Srivastava et al., 2013) provide the accuracies of 67.0% and 66.8%, respectively, for a feature dimension of $K = 128$, and provide the accuracies of 68.4% and 69.1%, respectively, for a feature dimension of $K = 512$.

As shown in Figure 11 and summarized in Table 3 of Appendix C, with the same vocabulary size of 2000 (but different terms due to different preprocessing), the proposed PGBN provides 65.9% (67.5%) with $T = 1$ ($T = 5$) for $K_{1\max} = 128$, and 65.9% (69.2%) with $T = 1$ ($T = 5$) for $K_{1\max} = 512$, which may be further improved if we also consider the stemming step, as done in these two algorithms, for word preprocessing, or if we set the values of $\eta^{(l)}$ to be smaller than 0.05 to encourage a more complex network structure. We also summarize in Table 3 the classification accuracies shown in Figure 10 for the PGBNs with $V = 33, 420$. Note that the accuracies in Tables 2 and 3 are provided to show that the PGBNs are in the same ballpark as both the DocNADE (Laroche and Laully, 2012) and over-replicated softmax (Srivastava et al., 2013). Note these results are not intended to provide a head-to-head comparison, which is possible if the same data preprocessing and classifier were used and the error bars were shown in Srivastava et al. (2013), or we could obtain the code to replicate the experiments using the same preprocessed data and classifier.

Note that 79.4% achieved using the 33, 420 selected features is the best accuracy reported in the paper, which is unsurprising since all the unsupervisedly extracted latent feature vectors have much lower dimensions and are not optimized for classification (Zhu et al., 2012; Zhou et al., 2015b). In comparison to using appropriately preprocessed high-dimensional features, our experiments show that while text classification performance often clearly deteriorates if one trains a multi-class classifier on the lower-dimensional features extracted using “shallow” unsupervised latent feature models, one could obtain much improved results using appropriate “deep” generalizations. For further improvement, one may consider adding an extra supervised component into the model, which is shown to boost the classification performance for latent Dirichlet allocation (Blei and Mcauliffe, 2008; Zhu et al., 2012), a shallow latent feature model related to the PGBN with a single hidden layer.

4.1.3 PERPLEXITIES FOR HELDOUT WORDS

In addition to examining the performance of the PGBN for unsupervised feature learning, we also consider a more direct approach that we randomly choose 30% of the word tokens in each document as training, and use the remaining ones to calculate per-heldout-word perplexity. We consider both all the 18,774 documents of the 20newsgroups corpus, limiting the vocabulary to the 2000 most frequent terms after removing a standard list of stopwords, and the NIPS12 (<http://www.cs.yyu.edu/~roweis/data.html>) corpus whose stopwords have already been removed, limiting the vocabulary to the 2000 most frequent terms. We set $\eta^{(l)} = 0.05$ and $C_t = 500$ for all t , set $B_1 = 1000$ and $B_t = 500$ for $t \geq 2$, and consider five random trials. Among the $B_t + C_t$ Gibbs sampling iterations used to train layer t , we collect one sample per five iterations during the last 500 iterations, for each of which we draw the topics $\{\phi_k^{(l)}\}_k$ and topics weights $\theta_j^{(l)}$, to compute the per-heldout-word perplexity using Equation (34) of Zhou and Carin (2015). This evaluation method is similar to those used in Newman et al. (2009), Wallach et al. (2009), and Paisley et al. (2012).

As shown in both Figures 12 and 13, we observe a clear trend of improvement by increasing both $K_{1\max}$ and T . We have also examined the topics and network structure learned on the NIPS12 corpus. Similar to the exploratory data analysis performed on the 20newsgroups corpus, as described in detail in Section 2.3, the inferred deep networks also allow us to extract trees and subnetworks to visualize various aspects of the NIPS12 corpus from general to specific and reveal how they are related to each other. We omit these details for brevity and instead provide a brief description: with $K_{1\max} = 200$ and $T = 5$, the PGBN infers a network with $[K_1, \dots, K_5] = [200, 164, 106, 60, 42]$ in one of the five random trials. The ranks, according to the weights $r_k^{(l)}$ calculated in (14), and the top five words of three example topics for layer $T = 5$ are “6 network units input learning training,” “15 data model learning set image,” and “34 network learning model input neural;” while these of five example topics of layer $T = 1$ are “19 likelihood em mixture parameters data,” “37 bayesian posterior prior log evidence,” “62 variables belief networks conditional inference,” “126 boltzmann binary machine energy hinton,” and “127 speech speaker acoustic vowel phonetic.” It is clear that the topics of the bottom hidden layers are very specific whereas these of the top hidden layer are quite general.

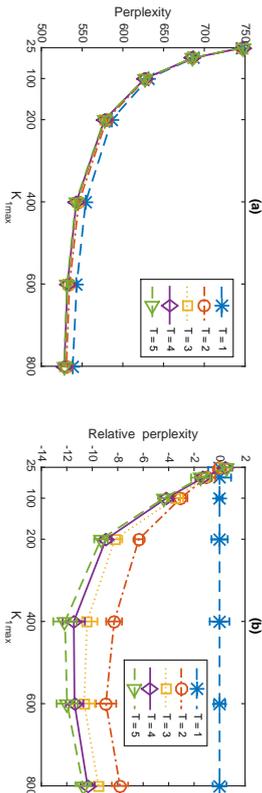


Figure 12: (a) per-held-out-word perplexity (the lower the better) for the NIPS12 corpus (using the 2000 most frequent terms) as a function of the upper bound of the first layer with $K_{1,max}$ and network depth T , with 30% of the word tokens in each document used for training and $\eta^{(l)} = 0.05$ for all l . (b) for visualization, each curve in (a) is reproduced by subtracting its values from the average perplexity of the single-layer network. In a random trial, the inferred network widths $[K_1, \dots, K_5]$ for $K_{1,max} = 25, 50, 100, 200, 400, 600$, and 800 are $[25, 25, 25, 25, 25]$, $[50, 50, 50, 49, 42]$, $[100, 99, 93, 78, 54]$, $[200, 164, 106, 60, 42]$, $[400, 130, 83, 52, 39]$, $[596, 71, 68, 58, 37]$, and $[755, 57, 53, 46, 42]$, respectively.

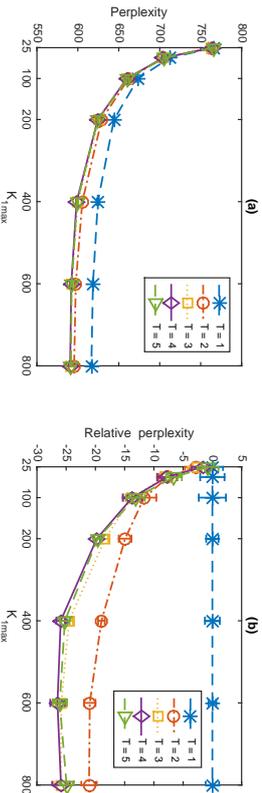


Figure 13: Analogous plots to Figure 12 for the 20newsgroups corpus (using the 2000 most frequent terms after removing a standard list of stopwords). In a random trial, the inferred network widths $[K_1, \dots, K_5]$ for $K_{1,max} = 25, 50, 100, 200, 400, 600$, and 800 are $[25, 25, 25, 25, 25]$, $[50, 50, 50, 50, 50]$, $[100, 99, 99, 97, 97]$, $[200, 194, 177, 152, 123]$, $[398, 199, 140, 116, 105]$, $[557, 156, 133, 118, 103]$, and $[701, 119, 116, 112, 103]$, respectively.

4.1.4 GENERATING SYNTHETIC DOCUMENTS

We have also tried drawing $\theta_j^{(T)} \sim \text{Gann}(\mathbf{r}, 1/c_j^{(T+1)})$ and downward passing it through a T -layer network trained on a text corpus to generate synthetic bag-of-words documents, which are found to be quite interpretable and reflect various general aspects of the corpus used to train the network. We consider the PGBN with $[K_1, \dots, K_5] = [608, 100, 99, 96, 89]$, which is trained on the training set of the 20newsgroups corpus with $K_{1,max} = 800$ and $\eta^{(l)} = 0.05$ for all l . We set $c_j^{(l)}$ as the median of the inferred $\{c_j^{(l)}\}_j$ of the training doc-

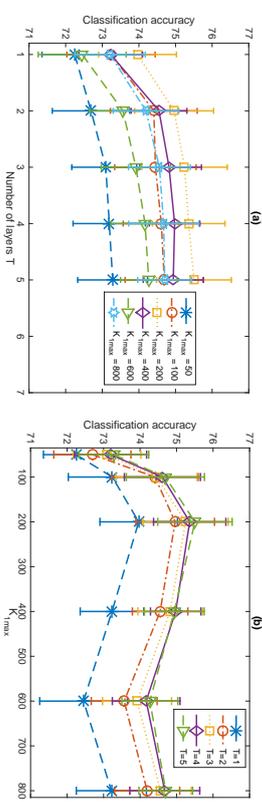


Figure 14: Analogous plots to Figure 10 for the BertPo-GBNs on the binarized 20news-groups term-document count matrix. The widths of the hidden layers are automatically inferred. In a random trial with Algorithm 2, the inferred network widths $[K_1, \dots, K_5]$ for $K_{1,max} = 50, 100, 200, 400, 600$, and 800 are $[50, 50, 50, 50, 50]$, $[100, 97, 95, 90, 82]$, $[178, 145, 122, 97, 72]$, $[184, 139, 119, 101, 75]$, $[172, 165, 158, 138, 110]$, and $[156, 151, 147, 134, 117]$, respectively.

uments for all l . Given $\{\Phi^{(l)}\}_{l,T}$ and \mathbf{r} , we first generate $\theta_j^{(T)} \sim \text{Gann}(\mathbf{r}, 1/c_j^{(T+1)})$ and then downward pass it through the network by drawing nonnegative real random variables, one layer after another, from the gamma distributions as in (1). With the simulated $\theta_j^{(l)}$, we calculate the Poisson rates for all the V words using $\Phi^{(1)}\theta_j^{(1)}$ and display the top 100 words ranked by their Poisson rates. As shown in the text file available at <http://mingyuanzhou.github.io/Results/GBN-BOW.txt>, the synthetic documents generated in this manner are all easy to interpret and reflect various general aspects of the 20newsgroups corpus on which the PGBN is trained.

4.2 Multilayer Representation for Binary Data

We apply the BertPo-GBN to extract multilayer representations for high-dimensional sparse binary vectors. The BertPo link is proposed in Zhou (2015) to construct edge partition models for network analysis, whose computation is mainly spent on pairs of linked nodes and hence is scalable to big sparse relational networks. That link function and its inference procedure have also been adopted by Hu et al. (2015) to analyze big sparse binary tensors.

We consider the same problem of feature learning for multi-class classification studied in detail in Section 4.1.2. We consider the same setting except that the original term-document word count matrix is now binarized into a term-document indicator matrix; the (i, j) element of which is set as one if and only if $n_{ij} \geq 1$ and set as zero otherwise. We test the BertPo-GBNs on the 20newsgroups corpus, with $\eta^{(l)} = 0.05$ for all layers. As shown in Figure 14, given the same upper-bound on the width of the first layer, increasing the depth of the network clearly improves the performance. Whereas given the same number of hidden layers, the performance initially improves and then fluctuates as the upper-bound of the first layer increases. Such kind of fluctuations when $K_{1,max}$ reaches over 200 are expected, since the width of the first layer is inferred to be less than 190 and hence the budget as small as $K_{1,max} = 200$ is already large enough to cover all active factors.

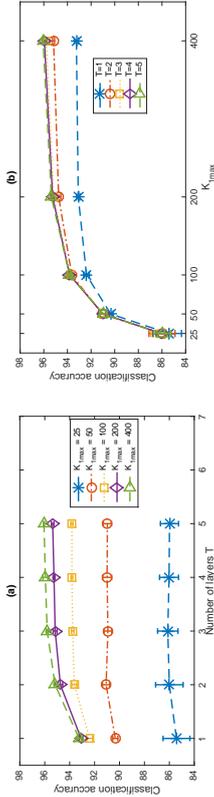


Figure 15: Analogous plots to Figure 10 for the PRG-GBNs on the MNIST data set. In a random trial with Algorithm 2, the inferred network widths $[K_1, \dots, K_5]$ for $K_{1,max} = 50, 100, 200$, and 400 are $[50, 50, 50, 50]$, $[100, 100, 100, 100]$, $[200, 200, 200, 200]$, and $[400, 400, 399, 385, 321]$, respectively.

4.3 Multilayer Representation for Nonnegative Real Data

We use the PRG-GBN to unsupervisedly extract features from nonnegative real data. We consider the MNIST data set (<http://yann.lecun.com/exdb/mnist/>), which consists of 60000 training handwritten digits and 10000 testing ones. We divide the gray-scale pixel values of each 28×28 image by 255 and represent each image as a 784 dimensional nonnegative real vector. We set $\eta^{(1)} = 0.05$ and use all training digits to infer the PRG-GBNs with $T_{max} \in \{1, \dots, 5\}$ and $K_{1,max} \in \{50, 100, 200, 400\}$. We consider the same problem of feature extraction for multi-class classification studied in detail in Section 4.1.2, and we follow the same experimental settings over there. As shown in Figure 15, both increasing the width of the first layer and the depth of the network could clearly improve the performance in terms of unsupervisedly extracting features that are better suited for multi-class classification.

Note that the PRG distribution might not be the best distribution to fit MNIST digits, but nevertheless, displaying the inferred features at various layers as images provides a straightforward way to visualize the latent structures inferred from the data and hence provides an excellent example to understand the properties and working mechanisms of the GBN. We display the projections to the first layer of the factors $\Phi^{(l)}$ at all five hidden layers as images for $K_{1,max} = 100$ and $K_{1,max} = 400$ in Figures 22 and 23, respectively, which clearly show that the inferred latent factors become increasingly more general as the layer increases. In both Figures 22 and 23, the latent factors inferred at the first hidden layer represent filters that are only active at very particular regions of the images, those inferred at the second hidden layer represent larger parts of the hidden-written digits, and those inferred at the third and deeper layers resemble the whole digits.

To visualize the relationships between the factors of different layers, we show in Figure 24 in Appendix C a subset of nodes of each layer and the nodes of the layer below that are connected to them with non-negligible weights.

It is interesting to note that unlike Lee et al. (2009) and many other following works that rely on the convolutional and pooling operations, which are pioneered by LeCun et al. (1989), to extract hierarchical representation for images at different spatial scales, we show that the proposed algorithm, while not breaking the images into spatial patches, is already able to learn the factors that are active on very specific spatial regions of the image in

the bottom hidden layer, and learn these increasingly more general factors covering larger spatial regions of the images as the number of layer increases. However, due to the lack of the ability to discover spatially localized features that can be shared at multiple different spatial regions, our algorithm does not at all exploit the redundancies of the spatially localized features inside a single image and hence may require much more data to train. Therefore, it would be interesting to investigate whether one can introduce convolutional and pooling operations into the GBNs, which may substantially improve their performance on modeling natural images.

5. Conclusions

The augmentable gamma belief network (GBN) is proposed to extract a multilayer representation for high-dimensional count, binary, or nonnegative real vectors, with an efficient upward-downward Gibbs sampler to jointly train all its layers and a layer-wise training strategy to automatically infer the network structure. A GBN of T layers can be broken into T subproblems that are solved by repeating the same subroutine, with the computation mainly spent on training the first hidden layer. When used for deep topic modeling, the GBN extracts very specific topics at the first hidden layer and increasingly more general topics at deeper hidden layers. It provides an excellent way for exploratory data analysis through the visualization of the inferred deep network, whose hidden units of adjacent layers are sparsely connected. Its good performance is further demonstrated in unsupervisedly extracting features for document classification and predicting heldout word tokens. The extracted deep network can also be used to simulate very interpretable synthetic documents, which reflect various general aspects of the corpus that the network is trained on. When applied for image analysis, without using the convolutional and pooling operations, the GBN is already able to extract interpretable factors in the first hidden layer that are active in very specific spatial regions and interpretable factors in deeper hidden layers with increasingly more general spatial patterns covering larger spatial regions. For big data problems, in practice one may rarely have a sufficient budget to allow the first-layer width to grow without bound, thus it is natural to consider a deep network that can use a multilayer deep representation to better allocate its resource and increase its representation power with limited computational power. Our algorithm provides a natural solution to achieve a good compromise between the width of each layer and the depth of the network.

Acknowledgments

The authors would like to thank the editor and two anonymous referees for their insightful and constructive comments and suggestions, which have helped us improve the paper substantially. M. Zhou thanks Texas Advanced Computing Center for computational support. B. Chen thanks the support of the Thousand Young Talent Program of China, NSFC (61372132), NCET-13-0945, and NDRP-9140A07010115DZ01015.

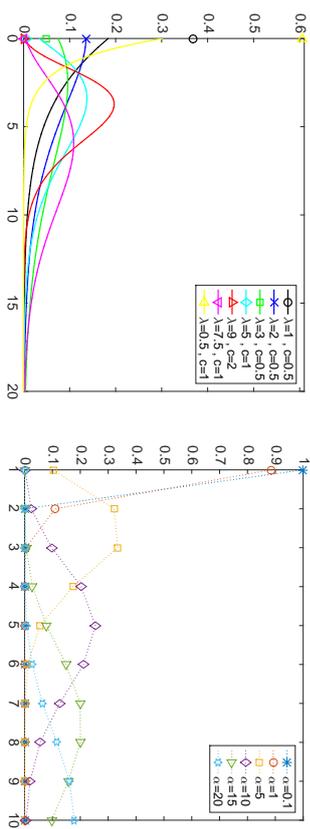


Figure 16: Left: probability distribution functions for the Poisson randomized gamma (PRG) distribution $x \sim \text{PRG}(\lambda, c)$, where the sum of the probability mass at $x = 0$ and the area under the probability density function curve for $x > 0$ is equal to one; Right: probability mass functions for the truncated Bessel distribution $n \sim \text{Bessel}_-(\alpha)$, where $n \in \{1, 2, \dots\}$.

Appendix A. Randomized Gamma and Bessel Distributions

Related to our work, Yuan and Kalbfleisch (2000) proposed the randomized gamma distribution to generate a random positive real number as

$$x | n, \nu \sim \text{Gam}(n + \nu + 1, 1/c), \quad n \sim \text{Pois}(\lambda),$$

where $\nu > -1$ and $c > 0$. As in Yuan and Kalbfleisch (2000), the conditional posterior of n can be expressed as

$$(n | x, \nu, \alpha) \sim \text{Bessel}_\nu(2\sqrt{cx\lambda})$$

where we denote $n \sim \text{Bessel}_\nu(\alpha)$ as the Bessel distribution with parameters $\nu > -1$ and $\alpha > 0$, with PMF

$$\text{Bessel}_\nu(n; \alpha) = \frac{\left(\frac{\alpha}{2}\right)^{2n+\nu}}{\Gamma_\nu(\alpha)n\Gamma(n+\nu+1)}, \quad n \in \{0, 1, 2, \dots\}.$$

Algorithms to draw Bessel random variables can be found in Devroye (2002).

The proposed PRG is different from the randomized gamma distribution of Yuan and Kalbfleisch (2000) in that it models both positive real numbers and exact zeros, and the proposed truncated Bessel distribution $n \sim \text{Bessel}_-(\alpha)$ is different from the Bessel distribution $n \sim \text{Bessel}_\nu(\alpha)$, where $\nu > -1$, in that it is defined only on positive integers. For illustration, we show in Figure 16 the probability distribution functions of both the PRG and truncated Bessel distributions under a variety of parameter settings.

Appendix B. Upward-Downward Gibbs Sampling

Below we first discuss Gibbs sampling for count data and then generalize it for both binary and nonnegative real data.

B.1 Inference for the PGBN

With Lemma 1 and Corollary 2 and the width of the first layer being bounded by $K_{1,\max}$, we first consider multivariate count observations and develop an upward-downward Gibbs sampler for the PGBN, each iteration of which proceeds as follows.

Sample $x_{vj}^{(t)}$. We can sample $x_{vj}^{(t)}$ for all layers using (25). But for the first hidden layer, we may treat each observed count $x_{vj}^{(1)}$ as a sequence of word tokens at the v th term (in a vocabulary of size $V := K_0$) in the j th document, and assign the $x_j^{(1)}$ words $\{v_j^i\}_{i=1, x_j^{(1)}}$ one after another to the latent factors (topics), with both the topics $\Phi^{(1)}$ and topic weights $\theta_j^{(1)}$ marginalized out, as

$$P(z_{ji} = k | -) \propto \frac{\eta_j^{(1)} + x_{vj}^{(1)-ji}}{V\eta_j^{(1)} + x_{\cdot, k}^{(1)-ji}} \left(x_{jk}^{(1)-ji} + \phi_{k\cdot}^{(2)} \theta_j^{(2)} \right), \quad k \in \{1, \dots, K_{1,\max}\}, \quad (28)$$

where z_{ji} is the topic index for v_j^i and $x_{vj}^{(1)} := \sum_i \delta(v_j^i = v, z_{ji} = k)$ counts the number of times that term v appears in document j ; we use x^{-ji} to denote the count x calculated without considering word i in document j . The collapsed Gibbs sampling update equation shown above is related to the one developed in (Zhou, 2014) for PFA using the beta-negative Dirichlet allocation, and the one developed in (Griffiths and Steyvers, 2004) for latent binomial process. When $T = 1$, we would replace the terms $\phi_{k\cdot}^{(2)} \theta_j^{(2)}$ with r_k for PFA built on the gamma-negative binomial process (Zhou and Carin, 2015) (or with $\alpha\pi_k$ for hierarchical Dirichlet process latent Dirichlet allocation, see (Teh et al., 2006) and (Zhou, 2014) for details), and add an additional term to account for the possibility of creating an additional factor (Zhou, 2014). For simplicity, in this paper, we truncate the nonparametric Bayesian model with $K_{1,\max}$ factors and let $r_k \sim \text{Gam}(\gamma_0/K_{1,\max}, 1/c_0)$ if $T = 1$. Note that although we use collapsed Gibbs sampling inference in this paper, if one desires embarrassingly parallel inference and possibly lower computation, then one may consider explicitly sampling $\{\phi_k^{(1)}\}_k$ and $\{\theta_j^{(1)}\}_j$ and sampling $x_{vj}^{(1)}$ with (25).

Sample $\phi_k^{(t)}$. Given these latent counts, we sample the factors/topics $\phi_k^{(t)}$ as

$$(\phi_k^{(t)} | -) \sim \text{Dir} \left(\eta_{1,k}^{(t)} + x_{1,k}^{(t)}, \dots, \eta_{K_{t-1},k}^{(t)} + x_{K_{t-1},k}^{(t)} \right). \quad (29)$$

Sample $x_{vj}^{(t+1)}$. We sample $x_j^{(t+1)}$ using (26), where we replace the term $\phi_{k\cdot}^{(T+1)} \theta_j^{(T+1)}$ with r_v .

Sample \mathbf{r} . Both γ_0 and c_0 are sampled using related equations in (Zhou and Carin, 2015), omitted here for brevity. We sample \mathbf{r} as

$$(r_v | -) \sim \text{Gam} \left(\gamma_0 / K_T + x_{v\cdot}^{(T+1)}, \left[c_0 - \sum_j \ln(1 - p_j^{(T+1)}) \right]^{-1} \right). \quad (30)$$

Sample $\theta_j^{(t)}$. Using (22) and the gamma-Poisson conjugacy, we sample θ_j as

$$\begin{aligned}
 (\theta_j^{(T)} | -) &\sim \text{Gam} \left(\mathbf{r} + \mathbf{m}_j^{(T)(T+1)}, [c_j^{(T)(T+1)} - \ln(1 - p_j^{(T)})]^{-1} \right), \\
 &\vdots \\
 (\theta_j^{(t)} | -) &\sim \text{Gam} \left(\Phi^{(t+1)} \theta_j^{(t+1)} + \mathbf{m}_j^{(t)(t+1)}, [c_j^{(t)(t+1)} - \ln(1 - p_j^{(t)})]^{-1} \right), \\
 &\vdots \\
 (\theta_j^{(1)} | -) &\sim \text{Gam} \left(\Phi^{(2)} \theta_j^{(2)} + \mathbf{m}_j^{(1)(2)}, [c_j^{(1)(2)} - \ln(1 - p_j^{(1)})]^{-1} \right), \tag{31}
 \end{aligned}$$

Sample $c_j^{(t)}$. With $\theta_j^{(t)} := \sum_{k=1}^{K_t} \theta_{kj}^{(t)}$ for $t \leq T$ and $\theta_j^{(T+1)} := \mathbf{r}$, we sample $p_j^{(2)}$ and $\{c_j^{(t)}\}_{t \geq 3}$ as

$$(p_j^{(2)} | -) \sim \text{Beta} \left(a_0 + m_{\cdot j}^{(1)(2)}, b_0 + \theta_j^{(2)} \right), \quad (c_j^{(t)} | -) \sim \text{Gam} \left(\epsilon_0 + \theta_j^{(t)}, [f_0 + \theta_j^{(t-1)}]^{-1} \right), \tag{32}$$

and calculate $c_j^{(2)}$ and $\{p_j^{(t)}\}_{t \geq 3}$ with (21).

B.2 Handling Binary and Nonnegative Real Observations

For binary observations that are linked to the latent counts at layer one as $b_{vj}^{(1)} = \mathbf{1}(x_{vj}^{(1)} \geq 1)$, we first sample the latent counts at layer one from the truncated Poisson distribution as

$$(x_{vj}^{(1)} | -) \sim b_{vj}^{(1)} \cdot \text{Pois}_+ \left(\sum_{k=1}^{K_1} \phi_{ok}^{(1)} \theta_{kj}^{(1)} \right) \tag{33}$$

and then sample $x_{vj}^{(t)}$ for all layers using (25).

For nonnegative real observations $y_{vj}^{(t)}$ that are linked to the latent counts at layer one as

$$y_{vj}^{(1)} \sim \text{Gam}(x_{vj}^{(1)}, 1/a_j),$$

we let $x_{vj}^{(1)} = 0$ if $y_{vj}^{(1)} = 0$ and sample $x_{vj}^{(1)}$ from the truncated Bessel distribution as

$$(x_{vj}^{(1)} | -) \sim \text{Bessel}_{-1} \left(2 \sqrt{a_j y_{vj}^{(1)} \sum_{k=1}^{K_1} \phi_{ok}^{(1)} \theta_{kj}^{(1)}} \right) \tag{34}$$

if $y_{vj}^{(1)} > 0$. We let $a_j \sim \text{Gam}(\epsilon_0, 1/f_0)$ in the prior and sample a_j as

$$(a_j | -) \sim \text{Gam} \left(\epsilon_0 + \sum_v x_{vj}^{(1)}, \frac{1}{f_0 + \sum_v y_{vj}^{(1)}} \right). \tag{35}$$

We then sample $x_{vj}^{(t)}$ for all layers using (25).

Appendix C. Additional Tables and Figures

Algorithm 1 The PGBN upward-downward Gibbs sampler that uses a layer-wise training strategy to train a set of networks, each of which adds an additional hidden layer on top of the previously inferred network, retrains all its layers jointly, and prunes inactive factors from the last layer. **Inputs:** observed counts $\{x_{vj}\}_{v,j}$, upper bound of the width of the first layer $K_{1 \max}$, upper bound of the number of layers T_{\max} , number of iterations $\{B_T, S_T\}_{1:T_{\max}}$, and hyper-parameters.

Outputs: A total of T_{\max} jointly trained PGBNs with depths $T = 1, T = 2, \dots$, and $T = T_{\max}$.

- 1: **for** $T = 1, 2, \dots, T_{\max}$ **do** Jointly train all the T layers of the network
- 2: Set K_{T-1} , the inferred width of layer $T-1$, as $K_{T \max}$, the upper bound of layer T 's width.
- 3: **for** $iter = 1 : B_T + C_T$ **do** Upward-downward Gibbs sampling
- 4: Sample $\{z_{ji}\}_{j,i}$ using collapsed inference; Calculate $\{x_{vjk}^{(1)}\}_{v,k,j}$; Sample $\{x_{vj}^{(2)}\}_{v,j}$;
- 5: **for** $t = 2, 3, \dots, T$ **do**
- 6: Sample $\{x_{vjk}^{(t)}\}_{v,j,k}$; Sample $\{\phi_k^{(t)}\}_k$; Sample $\{x_{vj}^{(t+1)}\}_{v,j}$;
- 7: **end for**
- 8: Sample $p_j^{(2)}$ and Calculate $c_j^{(2)}$; Sample $\{c_j^{(t)}\}_{j,t}$ and Calculate $\{p_j^{(t)}\}_{j,t}$ for $t = 3, \dots, T+1$;
- 9: **for** $t = T, T-1, \dots, 2$ **do**
- 10: Sample \mathbf{r} if $t = T$; Sample $\{\theta_j^{(t)}\}_j$;
- 11: **end for**
- 12: **if** $iter = B_T$ **then**
- 13: Prune layer T 's inactive factors $\{\phi_k^{(T)}\}_{k:z_{\cdot,k}^{(T)}=0}$;
- 14: let $K_T = \sum_k \delta(x_{\cdot,k}^{(T)} > 0)$ and update \mathbf{r} ;
- 15: **end if**
- 16: **end for**
- 17: Output the posterior means (according to the last MCMC sample) of all remaining factors $\{\phi_k^{(t)}\}_{k,t}$ as the inferred network of T layers, and $\{r_k\}_{k=1}^{K_T}$ as the gamma shape parameters of layer T 's hidden units.
- 18: **end for**

Algorithm 2 The upward-downward Gibbs samplers for the Ber-GBN and PRG-GBN are constructed by using Lines 1-8 shown below to substitute Lines 4-11 of the PGBN Gibbs sampler shown in Algorithm 1.

- 1: Sample $\{x_{vj}^{(1)}\}_{v,j}$ using (33) for binary observations; Sample $\{x_{vj}^{(1)}\}_{v,j}$ using (34) and sample a_j using (35) for nonnegative real observations;
- 2: **for** $t = 1, 2, \dots, T$ **do**
- 3: Sample $\{x_{vjk}^{(t)}\}_{v,j,k}$; Sample $\{\phi_k^{(t)}\}_k$; Sample $\{x_{vj}^{(t+1)}\}_{v,j}$;
- 4: **end for**
- 5: Sample $p_j^{(2)}$ and Calculate $c_j^{(2)}$; Sample $\{c_j^{(t)}\}_{j,t}$ and Calculate $\{p_j^{(t)}\}_{j,t}$ for $t = 3, \dots, T+1$;
- 6: **for** $t = T, T-1, \dots, 1$ **do**
- 7: Sample \mathbf{r} if $t = T$; Sample $\{\theta_j^{(t)}\}_j$;
- 8: **end for**

$V = 61, 188$	$V = 61, 188$	$V = 33, 420$	$V = 33, 420$
with stopwords	with stopwords	remove stopwords	remove stopwords
with rare words	with rare words	remove rare words	remove rare words
raw word counts	term frequencies	raw word counts	term frequencies
75.8%	77.6%	78.0%	79.4%
$V = 2000$	$V = 2000$	$V = 2000$	$V = 2000$
with stopwords	with stopwords	remove stopwords	remove stopwords
raw counts	term frequencies	raw counts	term frequencies
68.2%	67.9%	69.8%	70.8%

Table 1: Multi-class classification accuracies of L_2 regularized logistic regression.

DocNADE	$V = 2000, K = 128$	$V = 2000, K = 512$
	remove stopwords, stemming	remove stopwords, stemming
Over-replicated softmax	67.0%	68.4%
	66.8%	69.1%

Table 2: Multi-class classification accuracies of the DocNADE (Larochele and Lamy, 2012) and over-replicated softmax (Srivastava et al., 2013).

PGBN ($T = 1$)	$V = 2000, K_1 \text{ max} = 128$	$V = 2000, K_1 \text{ max} = 256$	$V = 2000, K_1 \text{ max} = 512$
	remove stopwords	remove stopwords	remove stopwords
	65.9% \pm 0.4%	66.3% \pm 0.4%	65.9% \pm 0.4%
	67.1% \pm 0.5%	67.9% \pm 0.4%	68.3% \pm 0.3%
	67.3% \pm 0.3%	68.6% \pm 0.5%	69.0% \pm 0.4%
PGBN ($T = 2$)	$V = 2000, K_1 \text{ max} = 128$	$V = 2000, K_1 \text{ max} = 256$	$V = 2000, K_1 \text{ max} = 512$
	remove stopwords	remove stopwords	remove stopwords
	67.5% \pm 0.4%	68.8% \pm 0.3%	69.2% \pm 0.4%
	$V = 33, 420, K_1 \text{ max} = 200$	$V = 33, 420, K_1 \text{ max} = 400$	$V = 33, 420, K_1 \text{ max} = 800$
	remove stopwords	remove stopwords	remove stopwords
PGBN ($T = 3$)	$V = 33, 420, K_1 \text{ max} = 200$	$V = 33, 420, K_1 \text{ max} = 400$	$V = 33, 420, K_1 \text{ max} = 800$
	remove rare words	remove rare words	remove rare words
	74.6% \pm 0.6%	75.3% \pm 0.6%	75.4% \pm 0.4%
	76.0% \pm 0.6%	76.9% \pm 0.5%	77.5% \pm 0.4%
	76.3% \pm 0.8%	77.1% \pm 0.6%	77.8% \pm 0.4%
PGBN ($T = 5$)	$V = 33, 420, K_1 \text{ max} = 200$	$V = 33, 420, K_1 \text{ max} = 400$	$V = 33, 420, K_1 \text{ max} = 800$
	remove rare words	remove rare words	remove rare words
	76.4% \pm 0.5%	77.4% \pm 0.6%	77.9% \pm 0.3%

Table 3: Multi-class classification accuracies of the PGBN trained with $\eta^l = 0.05$ for all l .

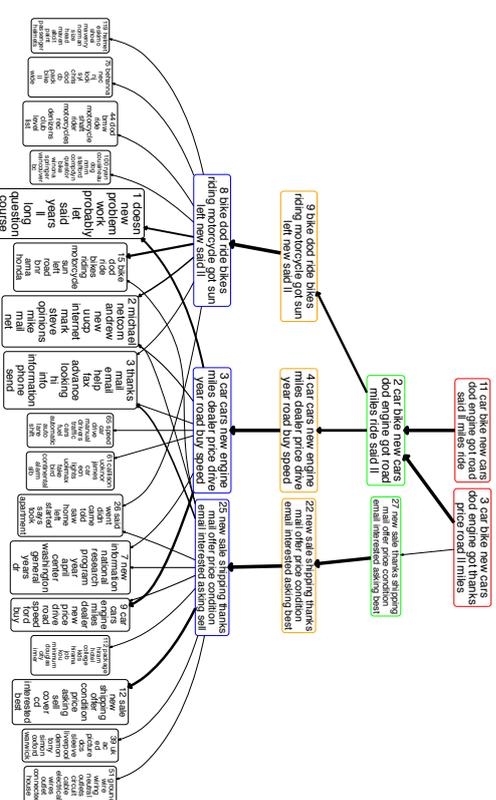


Figure 17: Analogous plots to Figure 6 for a subnetwork on “car & bike”, consisting of two trees rooted at nodes 3 and 11, respectively, of layer one.

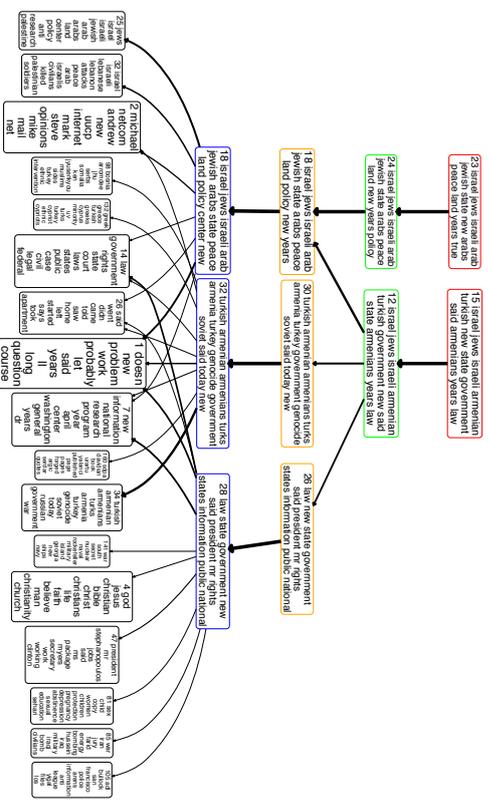


Figure 18: Analogous plot to Figure 6 for a subnetwork on “Middle East”, consisting of two trees rooted at nodes 15 and 23, respectively, of layer one.

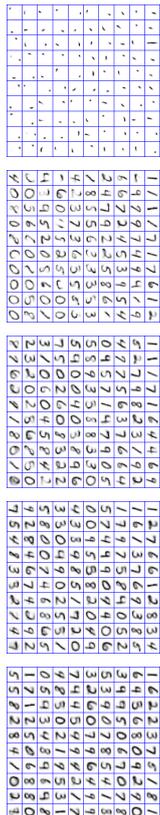


Figure 22: Visualization of the inferred $\{\Phi^{(1)}, \dots, \Phi^{(7)}\}$ on the MNIST data set using the PRG-GBN with $K_{lmax} = 100$ and $\eta^{(l)} = 0.05$ for all l . The latent factors of all layers are projected to the first layer: (a) $\Phi^{(1)}$, (b) $\Phi^{(1)}\Phi^{(2)}$, (c) $\Phi^{(1)}\Phi^{(2)}\Phi^{(3)}$, (d) $\Phi^{(1)}\Phi^{(2)}\Phi^{(3)}\Phi^{(4)}$, and (e) $\Phi^{(1)}\Phi^{(2)}\Phi^{(3)}\Phi^{(4)}\Phi^{(5)}$.



Figure 23: Visualization of the inferred $\{\Phi^{(1)}, \dots, \Phi^{(7)}\}$ on the MNIST data set using the PRG-GBN with $K_{lmax} = 400$ and $\eta^{(l)} = 0.05$ for all l . The latent factors of all layers are projected to the first layer: (a) $\Phi^{(1)}$, (b) $\Phi^{(1)}\Phi^{(2)}$, (c) $\Phi^{(1)}\Phi^{(2)}\Phi^{(3)}$, (d) $\Phi^{(1)}\Phi^{(2)}\Phi^{(3)}\Phi^{(4)}$, and (e) $\Phi^{(1)}\Phi^{(2)}\Phi^{(3)}\Phi^{(4)}\Phi^{(5)}$.

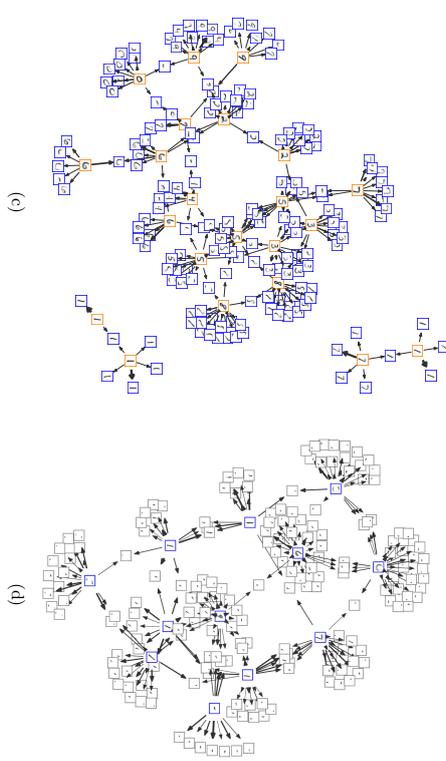
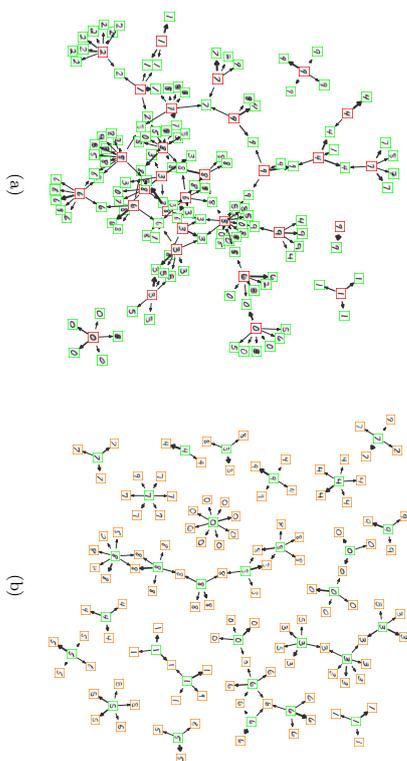


Figure 24: Visualization of the network structures inferred by the PRG-GBN on the MNIST data set with $K_{lmax} = 400$. (a) Visualization of the factors $\{\phi_1^{(5)}, \phi_2^{(5)}, \dots, \phi_{11}^{(5)}\}$ of layer five and those of layer four that are strongly connected to them. (b) Visualization of the factors $\{\phi_1^{(4)}, \phi_6^{(4)}, \dots, \phi_{106}^{(4)}\}$ of layer four and those of layer three that are strongly connected to them. (c) The Visualization of the factors $\{\phi_1^{(3)}, \phi_6^{(3)}, \dots, \phi_{140}^{(3)}\}$ of layer three and those of layer two that are strongly connected to them. (d) Visualization of the factors $\{\phi_1^{(2)}, \phi_6^{(2)}, \dots, \phi_{146}^{(2)}\}$ of layer two and those of layer one that are strongly connected to them.

References

- A. Acharya, J. Ghosh, and M. Zhou. Nonparametric Bayesian factor analysis for dynamic count matrices. In *AISTATS*, pages 1–9, 2015.
- R. P. Adams, Z. Ghahramani, and M. I. Jordan. Tree-structured stick breaking for hierarchical data. In *NIPS*, 2010.
- D. Aldous. Exchangeability and related topics. *École d’été de probabilités de Saint-Flour XIII-1983*, pages 1–198, 1985.
- F. J. Anscombe. Sampling theory of the negative binomial and logarithmic series distributions. *Biometrika*, 37(3-4):358–382, 1950.
- C. E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Ann. Statist.*, 2(6):1152–1174, 1974.
- Y. Bengio and Y. LeCun. Scaling learning algorithms towards AI. In Léon Bottou, Olivier Chapelle, D. DeCoste, and J. Weston, editors, *Large Scale Kernel Machines*. MIT Press, 2007.
- Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *NIPS*, pages 153–160, 2007.
- Y. Bengio, I. J. Goodfellow, and A. Courville. Deep Learning. Book in preparation for MIT Press, 2015.
- D. Blackwell and J. MacQueen. Ferguson distributions via Polya urn schemes. *Ann. Statist.*, 1(2):353–355, 1973.
- D. Blei and J. Lafferty. Correlated topic models. *NIPS*, 2006.
- D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.
- D. M. Blei and J. D. McAuliffe. Supervised topic models. In *NIPS*, 2008.
- D. M. Blei, T. L. Griffiths, and M. I. Jordan. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *Journal of ACM*, 2010.
- C. I. Bliss and R. A. Fisher. Fitting the negative binomial distribution to biological data. *Biometrics*, 9(2):176–200, 1953.
- W. Buntine and A. Jakulin. Discrete component analysis. In *Subspace, Latent Structure and Feature Selection Techniques*. Springer-Verlag, 2006.
- J. Canny. Gap: a factor model for discrete data. In *SIGIR*, 2004.
- J. Chen, J. Zhu, Z. Wang, X. Zheng, and B. Zhang. Scalable inference for logistic-normal topic models. In *NIPS*, 2013.
- S. C. Choi and R. Wette. Maximum likelihood estimation of the parameters of the gamma distribution and their bias. *Technometrics*, pages 683–690, 1969.
- L. Devroye. Simulating Bessel random variables. *Statistics & probability letters*, 57(3):249–257, 2002.
- R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *J. Mach. Learn. Res.*, pages 1871–1874, 2008.
- R. A. Fisher, A. S. Corbet, and C. B. Williams. The relation between the number of species and the number of individuals in a random sample of an animal population. *Journal of Animal Ecology*, 12(1):42–58, 1943.
- B. J. Frey. Continuous sigmoidal belief networks trained using slice sampling. pages 452–458, 1997a.
- B. J. Frey. Variational inference for continuous sigmoidal Bayesian networks. In *Statistical International Workshop on Artificial Intelligence and Statistics. Ft. Lauderdale FL*. Citeseer, 1997b.
- B. J. Frey and G. E. Hinton. Variational learning in nonlinear Gaussian belief networks. *Neural Computation*, 11(1):193–213, 1999.
- Z. Gan, C. Chen, R. Henao, D. Carlson, and L. Carin. Scalable deep Poisson factor analysis for topic modeling. In *ICML*, 2015a.
- Z. Gan, R. Henao, D. E. Carlson, and L. Carin. Learning deep sigmoid belief networks with data augmentation. In *AISTATS*, 2015b.
- M. Greenwood and G. U. Yule. An inquiry into the nature of frequency distributions representative of multiple happenings with particular reference to the occurrence of multiple attacks of disease or of repeated accidents. *J. R. Stat. Soc.*, 1920.
- T. L. Griffiths and M. Steyvers. Finding scientific topics. *PNAS*, 2004.
- G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.
- G. E. Hinton and Z. Ghahramani. Generative models for discovering sparse distributed representations. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 352(1358):1177–1190, 1997.
- G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.
- C. Hu, P. Rai, and L. Carin. Zero-truncated poisson tensor factorization for massive binary tensors. In *UAI*, 2015.
- N. L. Johnson, S. Kotz, and N. Balakrishnan. *Discrete multivariate distributions*, volume 165. Wiley New York, 1997.

- D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *ICLR*, 2014.
- H. Larochelle and S. Lauly. A neural autoregressive topic model. In *NIPS*, 2012.
- Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.
- D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, 2001.
- H. Lee, R. Grosse, R. Ranganath, and A. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *ICML*, 2009.
- S. Linderman, M. Johnson, and R. P. Adams. Dependent multinomial models made easy: Stick-breaking with the Polya-Gamma augmentation. In *NIPS*, pages 3438–3446, 2015.
- V. Nair and G. E. Hinton. Rectified linear units improve restricted Boltzmann machines. In *ICML*, pages 807–814, 2010.
- R. M. Neal. Connectionist learning of belief networks. *Artificial intelligence*, 56(1):71–113, 1992.
- D. Newman, A. Asuncion, P. Smyth, and M. Welling. Distributed algorithms for topic models. *J. Mach. Learn. Res.*, 2009.
- J. Paisley, C. Wang, and D. M. Blei. The discrete infinite logistic normal distribution. *Bayesian Analysis*, 2012.
- J. Paisley, C. Wang, D. M. Blei, and M. I. Jordan. Nested hierarchical dirichlet processes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2015.
- J. Pitman. *Combinatorial stochastic processes*. Lecture Notes in Mathematics. Springer-Verlag, 2006.
- N. G. Polson, J. G. Scott, and J. Windle. Bayesian inference for logistic models using polya-gamma latent variables. *arXiv preprint arXiv:1205.0310*, 2012.
- R. Ranganath and D. Blei. Correlated random measures. *arXiv:1507.00720v1*, 2015.
- R. Ranganath, S. Gerrish, and D. M. Blei. Black box variational inference. In *AISTATS*, 2014a.
- R. Ranganath, L. Tang, L. Charlin, and D. M. Blei. Deep exponential families. *arXiv*, 2014b.
- M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *CVPR*, 2007.
- D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, pages 1278–1286, 2014.
- R. Salakhutdinov and G. E. Hinton. Deep Boltzmann machines. In *AISTATS*, 2009.
- R. Salakhutdinov, J. B. Tenenbaum, and A. Torralba. Learning with hierarchical-deep models. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 1958–1971, 2013.
- L. K. Saul, T. Jaakkola, and M. I. Jordan. Mean field theory for sigmoid belief networks. *Journal of artificial intelligence research*, 4(1):61–76, 1996.
- N. Srivastava, R. Salakhutdinov, and G. E. Hinton. Modeling documents with a deep Boltzmann machine. In *UAI*, 2013.
- I. Sutskever and G. E. Hinton. Deep, narrow sigmoid belief networks are universal approximators. *Neural Computation*, 20(11):2629–2636, 2008.
- Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *J. Amer. Statist. Assoc.*, 2006.
- D.A. van Dyk and X. Meng. The art of data augmentation. *J. Comp. Graph. Stat.*, 2001.
- H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. In *ICML*, 2009.
- M. Welling, M. Rosen-Zvi, and G. E. Hinton. Exponential family harmoniums with an application to information retrieval. In *NIPS*, pages 1481–1488, 2004.
- S. Williamson, C. Wang, K. A. Heller, and D. M. Blei. The IBP compound Dirichlet process and its application to focused topic modeling. In *ICML*, 2010.
- E. P. Xing, R. Yan, and A. G. Hauptmann. Mining associated text and images with dual-wing harmoniums. In *UAI*, 2005.
- L. Yuan and J. D. Kalbfleisch. On the Bessel distribution and related problems. *Annals of the Institute of Statistical Mathematics*, 52(3):438–447, 2000.
- M. Zhou. Beta-negative binomial process and exchangeable random partitions for mixed-membership modeling. In *NIPS*, pages 3455–3463, 2014.
- M. Zhou. Infinite edge partition models for overlapping community detection and link prediction. In *AISTATS*, pages 1135–1143, 2015.
- M. Zhou and L. Carin. Negative binomial process count and mixture modeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(2):307–320, 2015.
- M. Zhou, L. Hannah, D. Dunson, and L. Carin. Beta-negative binomial process and Poisson factor analysis. In *AISTATS*, pages 1462–1471, 2012.
- M. Zhou, Y. Cong, and B. Chen. The Poisson gamma belief network. In *NIPS*, 2015a.
- M. Zhou, O. H. M. Padilla, and J. G. Scott. Priors for random count matrices derived from a family of negative binomial processes. *To appear in J. Amer. Statist. Assoc.*, 2015b.
- J. Zhu, A. Ahmed, and E. P. Xing. MedLDA: maximum margin supervised topic models. *J. Mach. Learn. Res.*, 13(1):2237–2278, 2012.

Optimal Estimation of Derivatives in Nonparametric Regression

Wenlin Dai

CEMSE Division

King Abdullah University of Science and Technology

Saudi Arabia

WENLIN.DAI@KAUST.EDU.SA

Tiejun Tong

Department of Mathematics

Hong Kong Baptist University

Hong Kong

TONGT@HKBU.EDU.HK

Marc G. Genton

CEMSE Division

King Abdullah University of Science and Technology

Saudi Arabia

MARC.GENTON@KAUST.EDU.SA

Editor: Xiaotong Shen

Abstract

We propose a simple framework for estimating derivatives without fitting the regression function in nonparametric regression. Unlike most existing methods that use the symmetric difference quotients, our method is constructed as a linear combination of observations. It is hence very flexible and applicable to both interior and boundary points, including most existing methods as special cases of ours. Within this framework, we define the variance-minimizing estimators for any order derivative of the regression function with a fixed bias-reduction level. For the equidistant design, we derive the asymptotic variance and bias of these estimators. We also show that our new method will, for the first time, achieve the asymptotically optimal convergence rate for difference-based estimators. Finally, we provide an effective criterion for selection of tuning parameters and demonstrate the usefulness of the proposed method through extensive simulation studies of the first- and second-order derivative estimators.

Keywords: Linear combination, Nonparametric derivative estimation, Nonparametric regression, Optimal sequence, Taylor expansion

1. Introduction

Consider the following nonparametric regression model:

$$Y_i = m(x_i) + \varepsilon_i, \quad 1 \leq i \leq n, \quad (1)$$

where x_i are the design points satisfying $0 \leq x_1 < \dots < x_n \leq 1$, $m(x)$ is the regression function, Y_i are the observations, and ε_i are independent and identically distributed random errors with $E(\varepsilon_i) = 0$ and $\text{var}(\varepsilon_i) = \sigma^2 < \infty$. Estimation of $m(x)$ is an important problem in nonparametric regression and has received sustained attention in the literature. Such

methods include, for example, kernel smoothing (Härdle, 1990), spline smoothing (Wahba, 1990), and local polynomial regression (Fan and Gijbels, 1996). It has been noted that the estimation of the first- or higher-order derivatives of $m(x)$ is also important for practical implementations including, but not limited to, the modeling of human growth data (Ramsay and Silverman, 2002), kidney function for a lupus nephritis patient (Ramsay, 2006), and Raman spectra of bulk materials (Charnigo et al., 2011). Derivative estimation is also needed in nonparametric regression to construct confidence intervals for regression functions (Eubank and Speckman, 1993), to select kernel bandwidths (Ruppert et al., 1995), and to compare regression curves (Park and Kang, 2008).

Most existing methods for p th-order derivative estimation can be expressed as a weighted average of the responses,

$$\hat{m}^{(p)}(x) = \sum_{i=1}^n w_i(x) Y_i,$$

where $w_i(x)$ are weights assigned to each observation Y_i . These estimators can be separated into two classes by their ability to directly or indirectly assess the weights, $w_i(x)$. In the indirect methods, the regression function is initially estimated as $\hat{m}(x) = \sum_{i=1}^n c_i(x) Y_i$ by the aforementioned nonparametric smoothing techniques, where $c_i(x)$ are smooth functions. Then, $w_i(x)$ are estimated as $d^p c_i(x)/dx^p$ (Gasser and Müller, 1984; Müller et al., 1987; Fan and Gijbels, 1995; Zhou and Wolfe, 2000; Boente and Rodriguez, 2006; Cao, 2014). We note, however, that the optimal bandwidths may differ for estimating the regression function and for estimating the derivatives, respectively. That is, a good estimate of the regression function may not guarantee the generation of good estimates of the derivatives.

Direct methods lead to the second class, which estimate the derivatives directly without fitting the regression function. The two key steps for such methods are constructing point-wise estimates for the derivatives of each design point and determining the amount of smoothing or the bandwidth. To select the bandwidth, one may refer to some classical methods in Müller et al. (1987), Härdle (1990), Fan and Gijbels (1996), Opsomer et al. (2001), Lahiri (2003), and Kim et al. (2009), among others. In contrast, little attention has been paid to the improvement of the point-wise estimation of the derivatives. One simple point-wise estimator for derivatives uses difference quotients. This method is, however, very noisy. For example, the variance of the first-order difference quotient $(Y_i - Y_{i-1})/(x_i - x_{i-1})$ is of order $O(n^2)$. Charnigo et al. (2011) proposed a variance-reducing linear combination of symmetric difference quotients, called *empirical derivatives*, and applied it to their generalized C_p criterion for tuning parameter selection. De Brabauter et al. (2013) established the L_1 and L_2 convergence rates for the empirical derivatives. Specifically, they defined the empirical derivatives as

$$Y_i^{(L)} = \sum_{j=1}^{k_L} w_{j,L} \left(\frac{Y_{i+j}^{(L-1)} - Y_{i-j}^{(L-1)}}{x_{i+j} - x_{i-j}} \right), \quad L = 1, \dots, p,$$

where $Y_i^{(L)}$ denotes the estimated L th-order derivative at x_i , $Y_i^{(0)} = Y_i$ and $w_{j,L}$ are the associated weights. When $L = 1$, $w_{j,1}$ are chosen as the optimal weights that minimize the estimation variance. For $L \geq 2$, $w_{j,L}$ are determined intuitively instead of by optimizing the estimation variance. As a consequence, their higher-order empirical derivatives may not

be optimally defined. Another attempt was made recently by Wang and Lin (2015). They estimated the derivative as the intercept of a linear regression model through the weighted least squares method. They further showed that their proposed estimators achieve better control of the estimation bias, which makes them superior to empirical derivatives when the signal-to-noise ratio is large. Finally, it is noteworthy that their method only applies to equidistant designs and hence the practical applications are somewhat limited.

In this paper, we propose a simple framework for estimating derivatives in model (1) without fitting the regression function. Our method does not rely on symmetric difference quotients; hence, it is more flexible than existing methods. Within this framework, we define the variance-minimizing estimators for any order derivative of $m(x)$ with a fixed bias-reduction level. For the equidistant design, we derive the asymptotic variance and bias of these estimators. We also show that the proposed estimators perform well on both interior and boundary points and, more importantly, that they achieve the optimal convergence rate for the mean squared error (MSE).

The rest of this paper is organized as follows. In Section 2, we propose a new framework for first-order derivative estimation and show that most existing estimators are special cases of ours. We also investigate the theoretical properties of the proposed estimator, including the optimal sequence, the asymptotic variance and bias, the point-wise consistency, and the boundary behavior. In Section 3, we extend the proposed method to higher-order derivative estimation and provide an effective criterion for the selection of tuning parameters. We then report extensive simulation studies in Section 4 that validate the proposed method. We conclude the paper with a discussion in Section 5. Technical proofs of the theoretical results are given in the Appendix.

2. First-order derivative estimation

In this section, we propose a new framework for estimating derivatives in nonparametric regression. Within this framework, we define the optimal estimator for the first-order derivative by minimizing the estimation variance. Theoretical results including the asymptotic variance and bias, and point-wise consistency are derived for the proposed optimal estimators under the equidistant design. We also investigate the performance of the estimators on the boundaries.

2.1 New framework

Recall that most existing methods are weighted average of symmetric difference quotients, which limits their implementation to some extent. All these estimators can be expressed as a linear combination of the observations for fixed design points. To proceed, we define

$$DY_i = \sum_{k=0}^r d_k Y_{i+k}, \quad 1 \leq i \leq n-r,$$

where (d_0, \dots, d_r) is a sequence of real numbers, and r is referred to as the order of the sequence. Assuming that $m(x)$ is a smooth enough function, we have the following Taylor

expansion at x_{i+l} for each $m(x_{i+k})$,

$$m(x_{i+k}) = m(x_{i+l}) + \sum_{j=1}^{\infty} \frac{(x_{i+k} - x_{i+l})^j}{j!} m^{(j)}(x_{i+l}), \quad 0 \leq l \leq r.$$

Note that x_{i+l} can be any design point within $[x_i, x_{i+r}]$, which frees our method from the symmetric form restriction. If we further assume that x_i are equidistant, then $x_i = i/n$, $i = 1, \dots, n$. Define $C_{j,l} = \sum_{k=0}^r d_k (k-l)^j / (n^j j!)$, $j = 0, 1, \dots$ and $l = 0, \dots, r$. The expectation of DY_i can be expressed as

$$E(DY_i) = \sum_{j=0}^{\infty} C_{j,l} m^{(j)}(x_{i+l}), \quad 1 \leq i \leq n-r. \quad (2)$$

To estimate the first-order derivative at x_{i+l} with DY_i , we let $C_{0,l} = 0$ and $C_{1,l} = 1$ so that

$$E(DY_i) = m'(x_{i+l}) + \sum_{j=2}^{\infty} C_{j,l} m^{(j)}(x_{i+l}),$$

where the second term on the right side is the estimation bias. When the regression function is oscillating around x_{i+l} , we can alter our model by controlling the estimation bias at a higher level. Specifically, if we let

$$C_{1,l} = 1 \text{ and } C_{j,l} = 0, \quad 0 \leq j \neq 1 \leq q-1, \quad (3)$$

then

$$E(DY_i) = m'(x_{i+l}) + \sum_{j=q}^{\infty} C_{j,l} m^{(j)}(x_{i+l}).$$

When $q = 2$, condition (3) reduces to $C_{1,l} = 1$ and $C_{0,l} = 0$. When $q \geq 3$, condition (3) eliminates the estimation bias up to order $q-1$.

2.2 Theoretical results

If we use a sequence with an order $r \geq q$, an infinite number of choices satisfying (3) is available. Among them, we choose the one(s) minimizing the estimation variance, $\text{var}(DY_i) = \sigma^2 \sum_{k=0}^r d_k^2$, which leads to the following optimization problem,

$$(d_0, \dots, d_r)_{1,q} = \underset{(d_0, \dots, d_r) \in \mathbb{R}^{r+1}}{\text{argmin}} \sum_{k=0}^r d_k^2, \quad \text{s.t.} \quad \text{condition (3) holds.}$$

We denote this variance-minimizing sequence as $(d_0, \dots, d_r)_{1,q}$. For simplicity of notation, the dependence of d_k on l is suppressed. In addition, we introduce the following notation:

$$I_i^{(l)} = \sum_{k=0}^r (k-l)^i, \quad l = 0, \dots, r \quad \text{and} \quad i = 0, 1, \dots;$$

$$U^{(l)} \text{ denotes a } q \times q \text{ matrix with } u_{ij}^{(l)} = I_{i+j-2}^{(l)};$$

$V^{(l)} = (U^{(l)})^{-1}$ is the inverse matrix of $U^{(l)}$.

Then, we present the theoretical results for $(d_0, \dots, d_r)_{1,q}$ in the following proposition.

Proposition 1 *Assume that model (1) holds with equidistant design and $m(x)$ has a finite q th-order derivative on $[0, 1]$. For $1 \leq i \leq n - r$ and $0 \leq l \leq r$, the unique variance-minimizing sequence is*

$$(d_k)_{1,q} = n \sum_{j=0}^{q-1} V_{(j+1,2)}^{(l)} (k-l)^j, \quad k = 0, \dots, r,$$

for estimating $m^{(l)}(x_{i+l})$ with an order of accuracy up to $m^{(q)}(x_{i+l})$, $q \geq 2$. Here, $V_{(i,d)}^{(l)}$ denotes the element in the i th row and the j th column of the matrix $V^{(l)}$. *Proof: see Appendix A.*

When q is fixed, the optimal sequence depends only on l , which makes it quite convenient for practical implementation. When r is even and $l = r/2$, we get the symmetric form used in De Brabanter et al. (2013) and Wang and Lin (2015). For this case, it is easy to verify that $d_k = -d_{r-k}$, which eliminates all the even-order derivatives in (2). The sequence is derived for the equidistant design on $[0, 1]$. To extend the result to equidistant designs on an arbitrary interval, $[a, b] \subset \mathbb{R}$, we can simply use $d_k/(b-a)$ instead. We treat the DY_i built on $(d_0, \dots, d_r)_{1,q}$ as the estimator for the first-order derivative with a bias-reduction level of q , denoted by $\hat{m}'_q(x_{i+l})$.

Theorem 1 *Assume that model (1) holds with equidistant design, $m(x)$ has a finite q th-order derivative on $[0, 1]$ and $r = o(n)$, $r \rightarrow \infty$. For $1 \leq i \leq n - r$ and $0 \leq l \leq r$, we have*

$$\begin{aligned} \text{var}[\hat{m}'_q(x_{i+l})] &= n^2 V_{(2,2)}^{(l)} \sigma^2 = O\left(\frac{n^2}{r^3}\right), \\ \text{bias}[\hat{m}'_q(x_{i+l})] &= \frac{1}{q! n^{q-1}} \sum_{j=0}^{q-1} V_{(j+1,2)}^{(l)} m^{(j)}(x_{i+l}) + o\left(\frac{r^{q-1}}{n^{q-1}}\right). \end{aligned}$$

Proof: see Appendix B.

For a larger q , the order of estimation bias is indeed reduced as expected, and the estimation variance surprisingly retains the same order at the same time. Assuming $r = n^\lambda$ and $2/3 < \lambda < 1$, we can establish the point-wise consistency of our estimator, $\hat{m}'_q(x_{i+l}) \xrightarrow{P} m'(x_{i+l})$, where " \xrightarrow{P} " means convergence in probability.

Corollary 1 *Assume that the conditions in Theorem 1 hold. When r is even and $l = r/2$,*

$$\hat{m}'_{2v}(x_{i+r/2}) = \hat{m}'_{2v+1}(x_{i+r/2}), \quad v = 1, 2, \dots, \left\lfloor \frac{q-1}{2} \right\rfloor,$$

where $\lfloor x \rfloor$ denotes the greatest integer less than or equal to x .

This means that, when we employ a symmetric form for our estimator, the optimal sequence is the same for $q = 2v$ and $q = 2v + 1$. In other words, the symmetric form further reduces the order of estimation bias without any increase in the estimation variance. Hence, it is natural to use the symmetric form (r is even and $l = r/2$) for the interior points, $\{x_i : 1 + r/2 \leq i \leq n - r/2\}$, when the design points are equidistant. Also, we can show that the two existing estimators for the first-order derivative (De Brabanter et al., 2013; Wang and Lin, 2015) are special cases of our method.

When $q = 2$ or $q = 3$, we get the same sequence as

$$(d_k)_{1,2} = (d_k)_{1,3} = \frac{6n(2k-r)}{r(r+1)(r+2)}, \quad k = 0, \dots, r.$$

This results in the empirical estimator in De Brabanter et al. (2013), denoted by \hat{m}'_{emp} . Assuming the regression function has a finite third-order derivative on $[0, 1]$, the estimation variance and bias are respectively

$$\text{var}[\hat{m}'_2(x_{i+r/2})] = \frac{12n^2\sigma^2}{r(r+1)(r+2)} \quad \text{and} \quad \text{bias}[\hat{m}'_2(x_{i+r/2})] = \frac{r^2}{40n^2} m^{(3)}(x_{i+r/2}) + o\left(\frac{r^2}{n^2}\right).$$

When $q = 4$ or $q = 5$, we get the same sequence as

$$(d_k)_{1,4} = (d_k)_{1,5} = \frac{n \left[I_6^{(r/2)}(k - \frac{r}{2}) - I_4^{(r/2)}(k - \frac{r}{2}) \right]}{I_2^{(r/2)} I_6^{(r/2)} - I_4^{(r/2)^2}}, \quad k = 0, \dots, r.$$

This results in the least squares estimator in Wang and Lin (2015), denoted by \hat{m}'_{lsq} . Within our framework, it is clear that the least squares estimator can be regarded as a bias-reduction modification of the empirical estimator.

Figure 1 presents an example of $\hat{m}'_q(x_i)$ with different levels of control for the estimation bias ($q = 3, 5$ and 7). We follow the regression function $m(x) = \sqrt{x(1-x)} \cdot \sin[2.1\pi/(x+0.05)]$ for model (1) from De Brabanter et al. (2013) and Wang and Lin (2015). Five hundred design points are equidistant on $[0.25, 1]$ and the random errors are generated from a Gaussian distribution, $N(0, 0.1^2)$. Sequence orders are chosen as $\{50, 100\}$. We observe that the estimation curves are smoother for smaller q , and the bias in oscillating areas decreases significantly for larger q . These results are consistent with our theoretical results. With various levels of bias control, we may achieve a better compromise in the trade-off between the estimation variance and bias.

2.3 Behavior on the boundaries

If we use a sequence with order r , then the boundary region will be $\{x_i : 1 \leq i \leq \lfloor r/2 \rfloor$ or $n - \lfloor r/2 \rfloor + 1 \leq i \leq n\}$. Within our framework, we have two types of estimators for estimating derivatives for the boundary area. One choice is to use a sequence with smaller order, so that we can still use the symmetric estimator as suggested for the interior points. This solution is also suggested by both De Brabanter et al. (2013) and Wang and Lin (2015). The other is to hold the sequence order while using an asymmetric form of the estimator instead.

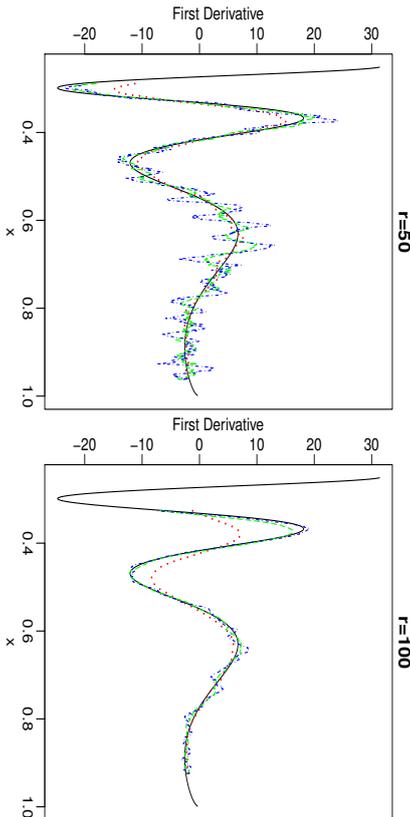


Figure 1: First-order derivative estimates with different levels of bias reduction. Red lines (dotted): $q = 3$; green lines (long dash): $q = 5$; blue lines (dot dash): $q = 7$ and black lines (solid): the true first-order derivative.

For the symmetric estimator, we can choose an even-order t satisfying $1 \leq t/2 \leq \min(i-1, n-i, \lceil r/2 \rceil)$. By Theorem 1, we have

$$\text{var}[\hat{m}'_q(x_i)] = O\left(\frac{n^2}{j^3}\right) \quad \text{and} \quad \text{bias}[\hat{m}'_q(x_i)] = O\left(\frac{t^{q-1}}{n^{q-1}}\right),$$

for $2 \leq i \leq \lceil r/2 \rceil$ or $n - \lceil r/2 \rceil + 1 \leq i \leq n - 1$. The closer x_i locates to the endpoints, the smaller the largest order of the chosen sequence, which means that the information we can incorporate into the estimator becomes very limited. As a consequence, the estimation variance will eventually reach an order of $O(n^2)$, which is rather noisy.

The asymmetric estimator does not require the estimated point to be located at the middle of the interval. We can still use a relatively large sequence order to include as much information as included in the interior points. The theoretical results were provided in Theorem 1:

$$\text{var}[\hat{m}'_q(x_i)] = O\left(\frac{n^2}{r^3}\right) \quad \text{and} \quad \text{bias}[\hat{m}'_q(x_i)] = O\left(\frac{r^{q-1}}{n^{q-1}}\right).$$

With a proper choice of r , we can still get a consistent estimate for the derivatives at the boundary region. Another advantage of this asymmetric form is that it is applicable to all the boundary points including x_1 and x_n , which can never be handled by the symmetric-form estimators.

It is noteworthy that Wang and Lin (2015) also proposed left-side and right-side weighted least squares estimators for the boundary points. Their estimators are, however, two special

cases of our asymmetric estimator with $q = 2$ and $l = 0$ (right-side) or $l = r$ (left-side). The estimation bias for $\hat{m}'_2(x_{i+l})$ is

$$\text{bias}[\hat{m}'_2(x_{i+l})] = \frac{r-2l}{2n} m''(x_{i+l}) + o\left(\frac{r}{n}\right).$$

To minimize the estimation bias on these boundary points, we recommend the following criterion:

$$\hat{m}'_2(x_i) = \begin{cases} DX_1 & 1 \leq i \leq \lceil r/2 \rceil, \\ DY_{n-r} & n - \lceil r/2 \rceil + 1 \leq i \leq n. \end{cases}$$

Then, the smallest absolute estimation bias can be simply derived as

$$|E[\hat{m}'_2(x_i)] - m'(x_i)| = \frac{r - 2\min(i-1, n-i)}{2n} |m''(x_i)| + o\left(\frac{r}{n}\right).$$

In summary, the asymmetric estimator generates a smaller variance, while its estimation bias is of a higher order. Consequently, the sequence order should be selected to achieve the best trade-off between the estimation variance and bias. In view of this, we recommend using the asymmetric estimator when the regression function is flat at the boundary region or when σ^2 is large; otherwise, the symmetric form should be employed.

3. Higher-order derivative estimation

In this section, we extend our method and propose higher-order derivative estimators for model (1). We further demonstrate that the new estimators possess the optimal estimation variance, which is not achieved by the two aforementioned methods (De Brabanter et al., 2013; Wang and Lin, 2015). Our new estimators also achieve the optimal convergence rate for MSE.

3.1. Theoretical results

To define an estimator for $m^{(p)}(x_{i+l})$ with a bias-reduction level up to $m^{(q)}(x_{i+l})$, we construct the new conditions on the coefficients as

$$C_{p,l} = 1 \quad \text{and} \quad C_{j,l} = 0, \quad 0 \leq j \neq p \leq q-1. \quad (4)$$

Then, the optimal sequence can be derived as the solution(s) of the following optimization problem:

$$(d_0, \dots, d_r)_{p,q} = \underset{(d_0, \dots, d_r) \in \mathbb{R}^{r+1}}{\text{argmin}} \sum_{k=0}^r d_k^2, \quad \text{s.t.} \quad \text{condition (4) holds.}$$

We present the result for $(d_0, \dots, d_r)_{p,q}$ in the following proposition.

Proposition 2 Assume that model (1) holds with equidistant design and $m(x)$ has a finite q th-order derivative on $[0, 1]$. For $1 \leq i \leq n-r$ and $0 \leq l \leq r$, the unique variance minimizing sequence is

$$(d_k)_{p,q} = p!r^p \sum_{j=0}^{q-1} V_{(j+1-p+l)}^{(0)} (k-j)^j, \quad k = 0, \dots, r,$$

for estimating $m^{(p)}(x_{i+t})$ with an order of accuracy up to $m^{(q)}(x_{i+t})$, $q \geq p + 1$.
Proof: see Appendix C.

To extend the result to equidistant designs on an arbitrary interval, $[a, b] \subset \mathbb{R}$, we can simply use $(d_k)_{p,q}/(b-a)^p$ instead. We treat the DY_i built on $(d_0, \dots, d_r)_{p,q}$ as the estimator for the p th-order derivative with a bias-reduction level up to $m^{(q)}(x_{i+t})$, denoted as $\hat{m}_q^{(p)}(x_{i+t})$.

Theorem 2 Assume that model (1) holds with equidistant design, $m(x)$ has a finite q th-order derivative on $[0, 1]$ and $r = o(n)$, $\tau \rightarrow \infty$. For $1 \leq i \leq n - r$ and $0 \leq l \leq r$, we have

$$\begin{aligned} \text{var}[\hat{m}_q^{(p)}(x_{i+t})] &= (p!)^2 n^{2p} V_{(p+1, p+1)}^{(l)} \sigma^2 = O\left(\frac{n^{2p}}{r^{2p+1}}\right), \\ \text{bias}[\hat{m}_q^{(p)}(x_{i+t})] &= \frac{p!}{q! n^{q-p}} \sum_{j=0}^{q-1} V_{(j+1, p+1)}^{(l)} I_{j+q}^{(l)} m^{(q)}(x_{i+t}) + o\left(\frac{r^{q-p}}{n^{q-p}}\right). \end{aligned}$$

Proof: see Appendix D.

For a fixed p and an increasing q , we can reduce the estimation bias to a lower order while keeping the order of variance unchanged. Whenever we keep the difference between q and p constant, the convergence rate of the bias is preserved for different p . When r is an even number and $l = r/2$, we can derive that $(d_k)_{p,q} = (-1)^p (d_{n-k})_{p,q}$. Consequently in this case, the optimal sequence remains the same when we increase q from $p + 2\nu - 1$ to $p + 2\nu$, $\nu = 1, 2, \dots$, which means $\hat{m}_{p+2\nu-1}^{(p)}(x_{i+r/2}) = \hat{m}_{p+2\nu}^{(p)}(x_{i+r/2})$. Hence, for this kind of estimator, the symmetric form is also the most favorable choice for the interior points.

The optimal MSE of our estimator is of order $O(n^{-2(q-p)/(2q+1)})$, which achieves the asymptotically optimal rate established by Stone (1980). For comparison, we note that the optimal MSE of the empirical estimator in De Brabanter et al. (2013) is of order $O(n^{-4/(2p+5)})$, that is, their estimator is of the optimal order only when $q = p + 2$. While for the least squares estimator in Wang and Lin (2015), they provided asymptotic results only for the first- and second-order derivative estimator. Their optimal MSE is of order $O(n^{-8/11})$ for $p = 1$ and $O(n^{-8/13})$ for $p = 2$, which corresponds with two special cases, i.e., when $(p, q) = (1, 5)$ or $(2, 6)$. From this point of view, our method has greatly improved the literature in derivative estimation and it achieves the optimal rate of MSE for any (p, q) from Theorem 2.

As mentioned at the beginning of this section, the newly defined estimator is optimal for the estimation variance, which is superior to existing estimators. In what follows, we illustrate this advantage in detail with the second-order derivative estimator, which is usually of greatest interest after the first-order derivative in practice. A similar analysis can be made for other higher-order derivative estimators. For the estimator without bias-control, e.g. \hat{m}_4'' , we derive the following results:

$$\begin{aligned} \text{var}[\hat{m}_4''(x_{i+r/2})] &= \frac{4n^4 \sigma^2 I_0^{(r/2)}}{I_0^{(r/2)} I_4^{(r/2)} - I_2^{(r/2)^2}}, \\ \text{bias}[\hat{m}_4''(x_{i+r/2})] &= \frac{r^2}{14n^2} m^{(4)}(x_{i+r/2}) + o\left(\frac{r^2}{n^2}\right). \end{aligned}$$

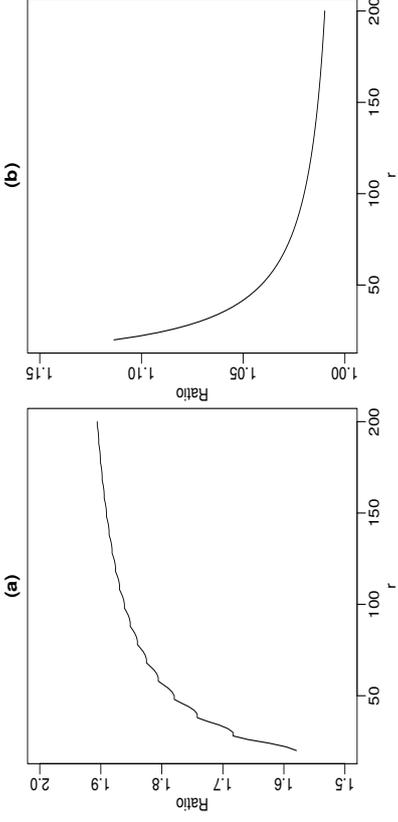


Figure 2: The ratio of estimation variance is plotted against the sequence order, r . Setting: $n = 500$ and r is chosen as an even integer ranging from 20 to 200. (a), $\text{var}(\hat{m}_{\text{emp}}''/\text{var}(\hat{m}_4''))$; (b), $\text{var}(\hat{m}_{\text{isc}}''/\text{var}(\hat{m}_6''))$.

The corresponding method is \hat{m}_{emp}'' in De Brabanter et al. (2013) with regard to the accurate level. Instead of minimizing the estimation variance, they intuitively choose the weight sequences for higher-order derivative estimators, which makes it quite difficult to derive analytical asymptotic results. Hence, we make a finite sample comparison of the variance of the two estimators. We set $n = 500$ and calculate the corresponding sequences for \hat{m}_4'' with an even order r ranging from 20 to 200 and $l = r/2$. For \hat{m}_{emp}'' , we choose (k_1, k_2) , which achieves the smallest estimation variance from $\{(k_1, k_2) : k_1 \leq k_2, k_1 + k_2 = r/2\}$. We do not need a specified form of the regression function, since it is not related with the estimation variance. We illustrate the ratio, $\text{var}(\hat{m}_{\text{emp}}''/\text{var}(\hat{m}_4''))$, in the left panel of Figure 2. Obviously, the new estimator improves the estimation variance significantly, which results in a smaller MSE for smooth regression functions.

A similar comparison is carried out between \hat{m}_{isc}'' and \hat{m}_6'' under the same settings, and the ratio of $\text{var}(\hat{m}_{\text{isc}}''/\text{var}(\hat{m}_6''))$ is presented in the right panel of Figure 2. Wang and Lin (2015) built a linear model with correlated regressors but employed the weighted least squares regression, rather than the generalized least squares technique, to derive the estimator. It can be shown that our method is equivalent with the generalized least squares estimator for their model. As expected, we find that our proposed estimator performs slightly better in terms of the finite sample than the least squares estimator. In addition their asymptotic variances and biases are equivalent for the first-order term. For the boundary points, our second-order estimator also maintains the same advantages over the existing estimators as discussed in Section 2.2 for the first-order estimators.

3.2 Tuning parameter selection

As shown in Figure 1, the order, r , and the bias-reduction level, q , are both critical to the proposed estimators. For practical implementation, (r, q) should be chosen to achieve a better trade-off between the estimation variance and bias.

By Theorem 2, the approximated MSE of $\hat{m}_q^{(p)}(x_{i+h})$ is

$$\text{MSE}[\hat{m}_q^{(p)}(x_{i+h})] \simeq (p!)^2 n^{-2p} V_{(p+1, p+1)}^{(l)} \sigma^2 + \left[\frac{p!}{q! n^{q-p}} \sum_{j=0}^{q-1} V_{(j+1, p+1)}^{(l)} J_{j+q}^{(l)} m^{(q)}(x_{i+h}) \right]^2.$$

We define the averaged mean squared error (AMSE) as a measure of the goodness of fit for all the design points,

$$\text{AMSE}(\hat{m}_q^{(p)}) = \frac{1}{n} \sum_{i=1}^n \text{MSE}[\hat{m}_q^{(p)}(x_i)].$$

A uniform sequence is preferred for the estimate at most points (all the interior points for example) over different sequences for each design point. Hence, we can choose the parameters (r, q) minimizing the AMSE. To achieve this, we replace the unknown quantities, σ^2 and $m^{(q)}(x_{i+h})$, with their consistent estimates. The error variance can be estimated by the method in Tong and Wang (2005) and Tong et al. (2013) and $m^{(q)}(x_i)$ can be estimated by the local polynomial regression of order $q+2$. For the high-order derivatives at the boundary points, we recommend replacing the AMSE for all the points with the following adjusted form:

$$\text{AMSE}_{\text{adj}}(\hat{m}_q^{(p)}) = \frac{1}{n-r} \sum_{i=1+r}^{n-r} \text{MSE}[\hat{m}_q^{(p)}(x_i)] \simeq B_1 \sigma^2 + \frac{B_2}{n-r} \sum_{i=1+r}^{n-r} [m^{(q)}(x_i)]^2, \quad (5)$$

where $B_1 = (p!)^2 n^{2p} V_{(p+1, p+1)}^{(r/2)}$ and $B_2 = \left[\frac{p!}{q! n^{q-p}} \sum_{j=0}^{q-1} V_{(j+1, p+1)}^{(r/2)} J_{j+q}^{(r/2)} \right]^2$. Given all the parameters for a specific problem, B_1 and B_2 are available quantities. The adjusted AMSE includes only derivatives at the interior points that share the identical difference sequence for an even r and $l = r/2$. Another advantage is that we only need $V^{(r/2)}$ and $J_{j+q}^{(r/2)}$ instead of $V^{(l)}$ and $J_{j+q}^{(l)}$ for $l = 0, \dots, r$, which greatly reduces the computation time.

For the tuning parameter space of the sequence order, we recommend $r \in O = \{2l : 1 \leq l \leq k_0\}$, where $k_0 \leq \lfloor n/4 \rfloor$, to keep a symmetric form ($l = r/2$) for the interior points and to make sure that the number of boundary points will be less than that of the interior points. For the bias-reduction level of $\hat{m}_q^{(p)}$, we consider $q \in Q = \{p+2\nu : \nu = 1, 2, \dots, \nu_0\}$, where $p+2\nu_0$ is the highest level chosen by users. Only even differences are considered for $q-p$, since $\hat{m}_{p+2\nu_0-1}^{(p)} = \hat{m}_{p+2\nu_0}^{(p)}$ when we use the recommended symmetric form.

4. Simulation study

In this section, we conduct simulation studies to assess the finite sample performance of the proposed estimators, $\hat{m}_q^{(p)}$, and make comparisons with the empirical estimator, $\hat{m}_{\text{emp}}^{(p)}$.

in De Brabanter et al. (2013) and the least squares estimator, $\hat{m}_{\text{lse}}^{(p)}$, in Wang and Lin (2015). We apply the three methods to both interior (Int) and boundary (Bd) areas, where $\text{Int} = \{x_i : k_0 + 1 \leq i \leq n - k_0\}$ and $\text{Bd} = \{x_i : 1 \leq i \leq k_0 \text{ or } n - k_0 + 1 \leq i \leq n\}$. Throughout the simulation, we set $k_0 = \lfloor n/10 \rfloor$, which means that we treat ten percent of the design points on both sides of the interval as boundary points. We also tried some other proportions and the results were similar. For the interior part, we keep the symmetric form for $\hat{m}_q^{(p)}$ by setting r as an even number and $l = r/2$, as suggested in the theoretical results. For the boundary part, we apply the following criterion for the proposed estimators:

$$\hat{m}_q^{(p)}(x_i) = \begin{cases} DY_1 & 1 \leq i \leq \lfloor r/2 \rfloor, \\ DY_{n-r} & n - \lfloor r/2 \rfloor + 1 \leq i \leq n. \end{cases}$$

The modified version of $\hat{m}_{\text{emp}}^{(p)}$ in De Brabanter et al. (2013) and the one-side weighted least squares estimators in Wang and Lin (2015) are investigated for the empirical and least squares estimators, respectively on the boundary points. We consider estimators for both first- and second-order derivatives, which are of most interest in practice. Similar to De Brabanter et al. (2013) and Wang and Lin (2015), the mean absolute error (MAE) is used as a measure of estimation accuracy. It is defined as follows:

$$\text{MAE} = \frac{1}{\#\mathbb{A}} \sum_{x_i \in \mathbb{A}} |\hat{m}_q^{(p)}(x_i) - m^{(p)}(x_i)|,$$

where $\mathbb{A} = \text{Int}$ or Bd and $\#\mathbb{A}$ denotes the number of elements in set \mathbb{A} .

We consider the following regression function,

$$m(x) = 5 \sin(w\pi x),$$

with $\omega = 1, 2, 4$ corresponding to different levels of oscillations. The $n = 100$ and 500 sample sizes are investigated. We set the design points as $x_i = i/n$ and generate the random errors, ε_i , independently from $N(0, \sigma^2)$. For each regression function, we consider $\sigma = 0.1, 0.5$ and 2 to capture the small, moderate and large variances, respectively. In total, we have 18 combinations of simulation settings. Following the definitions of Int and Bd, we select the sequence order r from $\mathbb{O} = \{2l : 1 \leq l \leq k_0\}$. We choose the bias-reduction level, q , from $\mathbb{Q} = \{p+2, p+4, p+6\}$, with $q = p+2$ and $q = p+4$ corresponding to $\hat{m}_{\text{emp}}^{(p)}$ and $\hat{m}_{\text{lse}}^{(p)}$, respectively, and $q = p+6$ as an even higher level. We denote by $\hat{m}_{\text{opt}}^{(p)}$ the estimator with the selected tuning parameters. For $\hat{m}_{\text{emp}}^{(p)}$ and $\hat{m}_{\text{lse}}^{(p)}$, the parameter k is chosen from $\{l : 1 \leq l \leq k_0\}$. We investigate two scenarios (for the tuning parameters selection criterion): *oracle* and *plug-in* (see below). For each run of the simulation, we compute the MAE of the estimators at both Int and Bd and repeat the procedure 1000 times for each setting. The simulation results for $w = 2$ are reported as box-plot figures. Other results are provided in the supplementary materials.

Oracle parameters

Oracle parameters are selected by assuming that we know the true regression (derivative) function, the purpose of which is to illustrate the possible best performance of each

estimator. Specifically for $\hat{m}_q^{(p)}$, the pair of tuning parameters is chosen as

$$(\tau, q)_{\text{opt}} = \underset{\tau \in \bar{\mathbb{O}}, q \in \mathbb{Q}}{\operatorname{argmin}} \left(\operatorname{MAE}(\hat{m}_q^{(p)}) \right).$$

The bandwidths of $\hat{m}_{\text{emp}}^{(p)}$ and $\hat{m}_{\text{lse}}^{(p)}$ are selected through a similar procedure.

For the first-order derivative, we investigate \hat{m}'_{opt} , \hat{m}'_{emp} and \hat{m}'_{lse} and report the simulation results in Figure 3. On the interior points, \hat{m}'_{opt} always possesses the same MAE as the smaller one of \hat{m}'_{emp} and \hat{m}'_{lse} , due to the fact that \hat{m}'_{emp} and \hat{m}'_{lse} are two special cases of \hat{m}'_q in this area. On the boundary points, \hat{m}'_{opt} is uniformly better than the other two methods. To further explore the reason for the boundary behavior, we use an example from De Brabanter et al. (2013) and Wang and Lin (2015). The fitted results for the three estimators are illustrated in Figure 4, where the red points represent the boundary parts. The empirical estimator suffers a lot from the increasing variance when the estimated points get close to the endpoints of the interval. The least squares estimator simply estimates the boundary parts by shifting the estimates of the interior points nearby, which results in very serious estimation bias. Our estimator fits the boundary points very well, resulting from the flexibility brought by the parameter l , the relative location of the estimated point within the interval $[x_i, x_{i+r}]$.

For the second-order derivative, we include another two estimators, \hat{m}''_4 and \hat{m}''_6 , which have the same bias-reduction level as \hat{m}''_{emp} and \hat{m}''_{lse} , respectively. The sequence order, τ , of the two additional estimators is optimally chosen by minimizing MAE, as well. The simulation results are presented in Figure 5. The relationships between \hat{m}''_{opt} , \hat{m}''_{emp} and \hat{m}''_{lse} remain the same as those observed for the first-order derivative. We also observe that $\operatorname{MAE}(\hat{m}''_4)$ is significantly smaller than $\operatorname{MAE}(\hat{m}''_{\text{emp}})$ and that $\operatorname{MAE}(\hat{m}''_6)$ is almost the same as $\operatorname{MAE}(\hat{m}''_{\text{lse}})$, consistent with our theoretical results in Section 3.

Plug-in parameters

Plug-in parameters are chosen via minimizing the adjusted AMSE in (5) after replacing all the unknown quantities with their consistent estimates. In this simulation, we estimate σ^2 using Tong and Wang's (2005) method with the recommended bandwidth $[n^{1/3}]$. Here, $\hat{m}^{(q)}(x_i)$ ($1 + k_0 \leq i \leq n - k_0$) are calculated with the function *locpol* in the R package *locpol* (Ojeda Cabrera, 2012) with the parameter *deg* = $q + 2$. The bandwidths of $\hat{m}_{\text{emp}}^{(p)}$ and $\hat{m}_{\text{lse}}^{(p)}$ are selected accordingly.

We report the simulation results together with those for the oracle parameters in Figures 6 and 7. From the comparison, we observe that the plug-in parameters lead to quite similar results with those for the oracle parameters, especially on the interior points. Since the tuning parameters are selected based on AMSE of derivative estimates for the interior points, the performance on the boundary is not consistent. Nevertheless, the mutual relationship of the three estimators remains the same for most cases on both interior and boundary points. Overall, the proposed plug-in method is quite effective for choosing the optimal tuning parameters.

In summary, we have demonstrated the superiority of the proposed estimators over the existing estimators through extensive simulation studies. We have further provided an effective criterion for selection of the tuning parameters for the newly defined estimator.

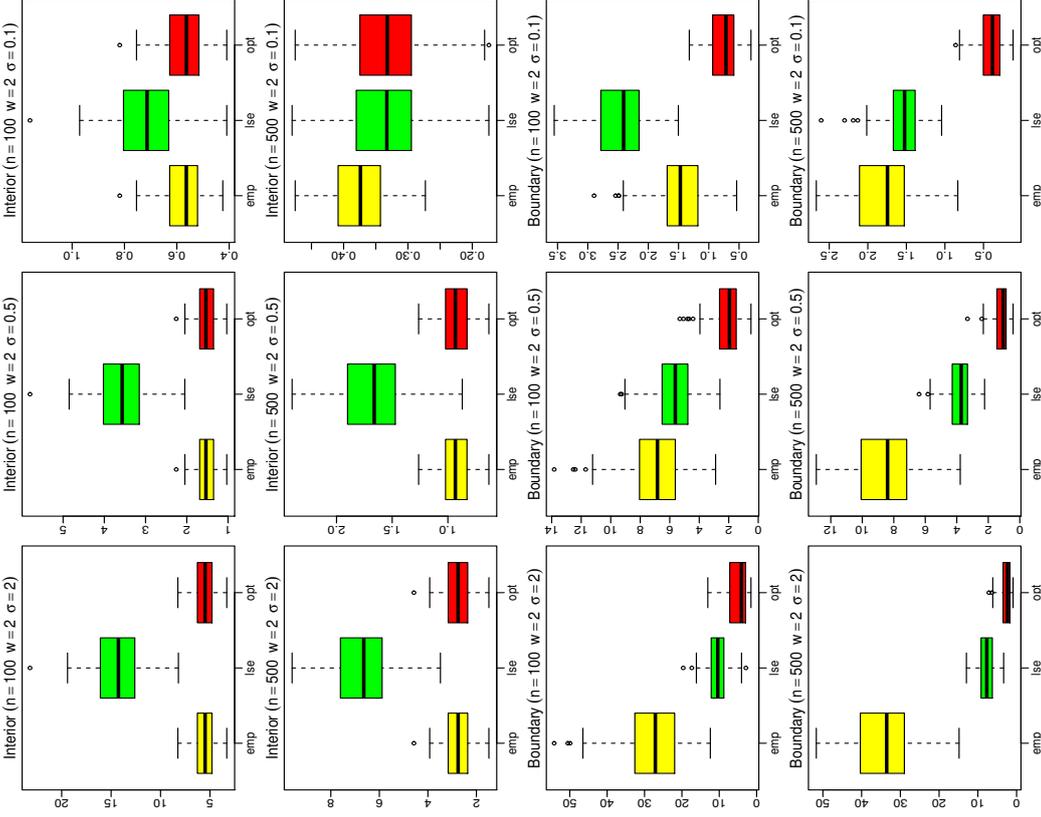


Figure 3: Mean absolute errors of three first-order derivative estimators on both interior (2 top rows) and boundary (2 bottom rows) points for various settings. \hat{m}'_{emp} , yellow box; \hat{m}'_{lse} , green box; \hat{m}'_{opt} , red box. $m(x) = 5 \sin(2\pi x)$ and $\varepsilon \sim N(0, \sigma^2)$.

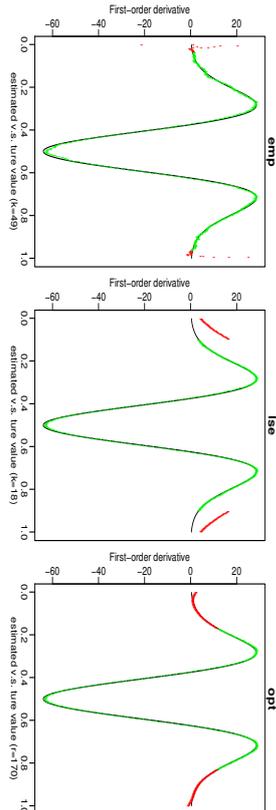


Figure 4: The fitted point-wise derivatives by the three estimators using oracle tuning parameters. $m(x) = 32e^{-8(1-2x)^2(1-2x)}$, ε_i are independent random errors from $N(0, 0.1^2)$ and $n = 500$. Interior points: green points. Boundary points: red points.

5. Conclusion

We proposed a new framework for estimating derivatives without fitting the regression function. Unlike most existing methods using the symmetric difference quotients, our method is constructed as a linear combination of the observations. It is hence very flexible and applicable to both interior and boundary points. We obtained the variance-minimizing estimators for the first- and higher-order derivatives with a fixed bias-reduction level. Under the equidistant design, we derived some theoretical results for the proposed estimators including the optimal sequence, asymptotic variance and bias, point-wise consistency, and boundary behavior. We illustrated that the order of the estimation bias can be reduced while the order of variance remains unchanged. We showed that our method achieves the optimal convergence rate for the MSE. Furthermore, we provided an effective selection procedure for the tuning parameters of the proposed estimators. Simulation studies for the first- and second-order derivative estimators demonstrated the superiority of our proposed method.

The method can be readily extended to unequally spaced designs. In this case, the symmetric form is no longer valid and the choice of l also deserves further consideration. To estimate the point-wise derivatives for unequally spaced designs, we can first find the r nearest neighbors of the estimated point and construct the variance-minimizing estimator with the linear combination of the $r + 1$ points, say $x_i < \dots < x_{i+l} < \dots < x_{i+r}$. Assuming that $m(x)$ is smooth enough and that x_{i+l} is the estimated point, we have the expectation of DY_i as

$$E(DY_i) = m(x_{i+l}) \sum_{k=0}^r d_k + \sum_{j=1}^{\infty} m^{(j)}(x_{i+l}) \sum_{k=0}^r d_k (x_{i+l+k} - x_{i+l})^j / j!$$

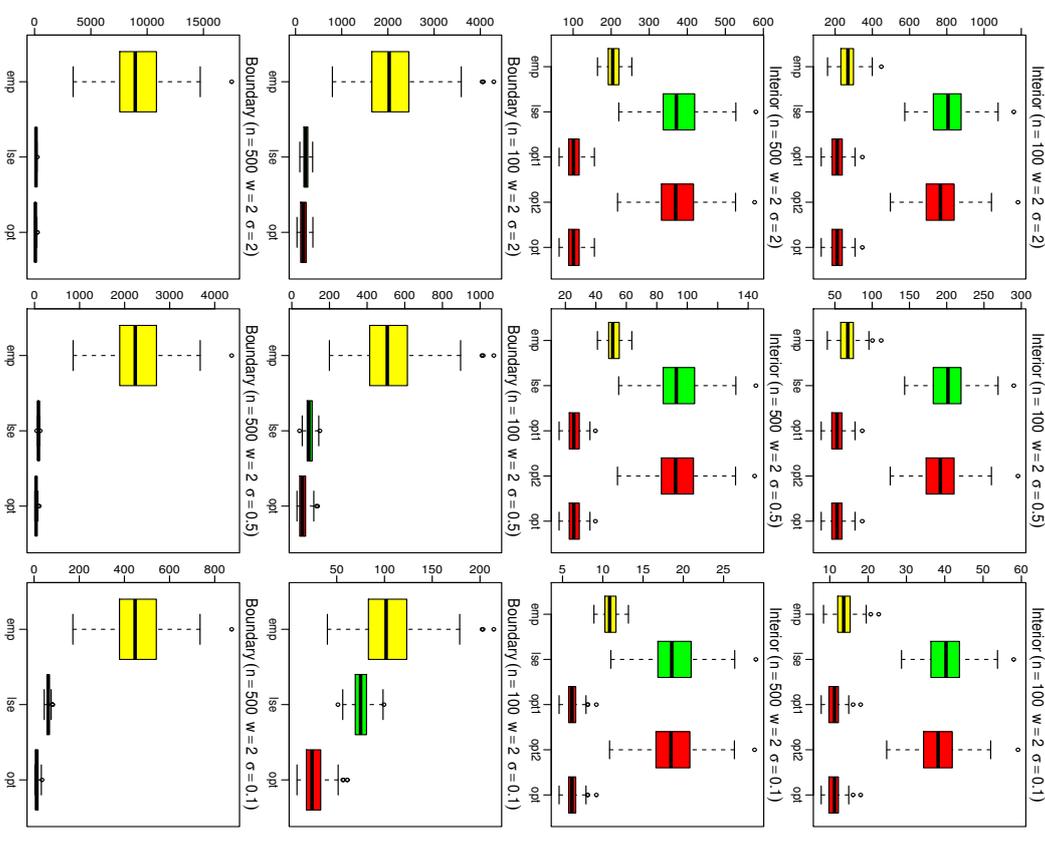


Figure 5: Mean absolute errors of three second-order derivative estimators on both interior (2 top rows) and boundary (2 bottom rows) points for various settings. \hat{m}_{emp}' , yellow box; \hat{m}_{ise}' , green box; \hat{m}_{opt}' , red box. opt1: \hat{m}_{opt}' ; opt2: \hat{m}_{opt}'' ; opt: \hat{m}_{opt}''' . $m(x) = 5 \sin(2\pi x)$ and $\varepsilon \sim N(0, \sigma^2)$.

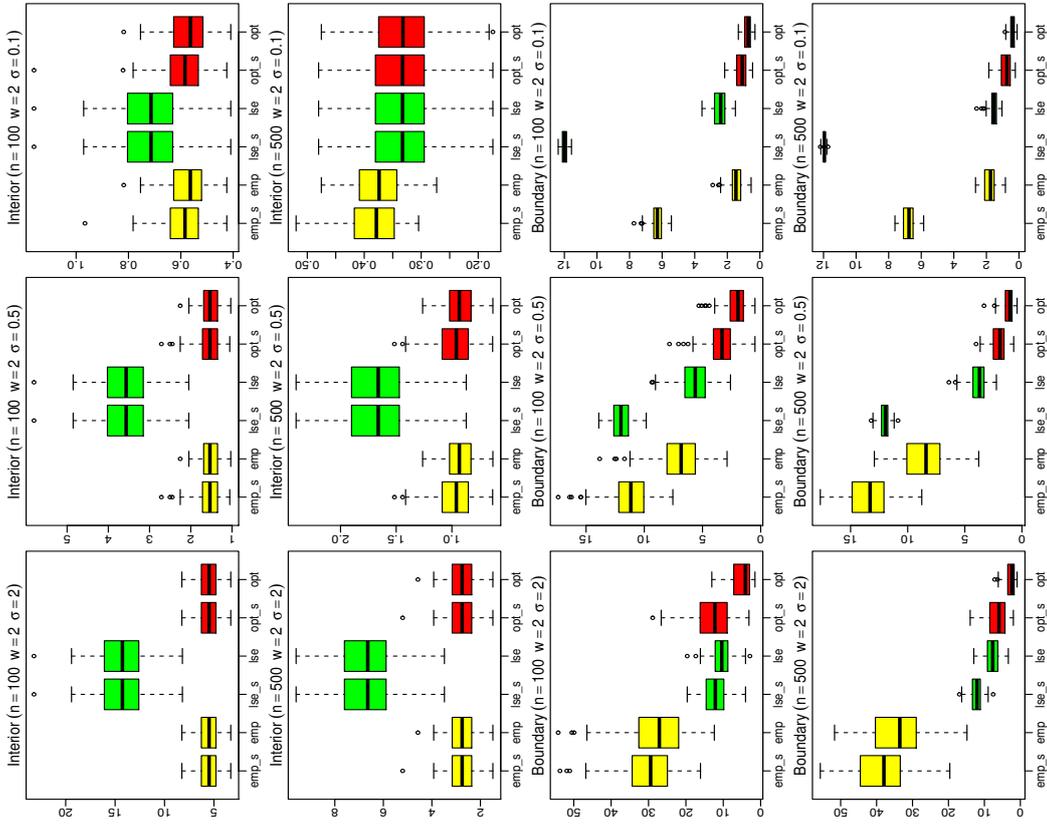


Figure 6: Comparison of the mean absolute errors on both interior (2 top rows) and boundary (2 bottom rows) points between the first-order derivative estimators with oracle tuning parameters and those with plug-in tuning parameters. \hat{m}'_{emp} , yellow box; \hat{m}'_{lse} , green box; \hat{m}'_{opt} , red box. “ s ” denotes estimators using plug-in parameters. $m(x) = 5 \sin(2\pi x)$ and $\varepsilon \sim N(0, \sigma^2)$.

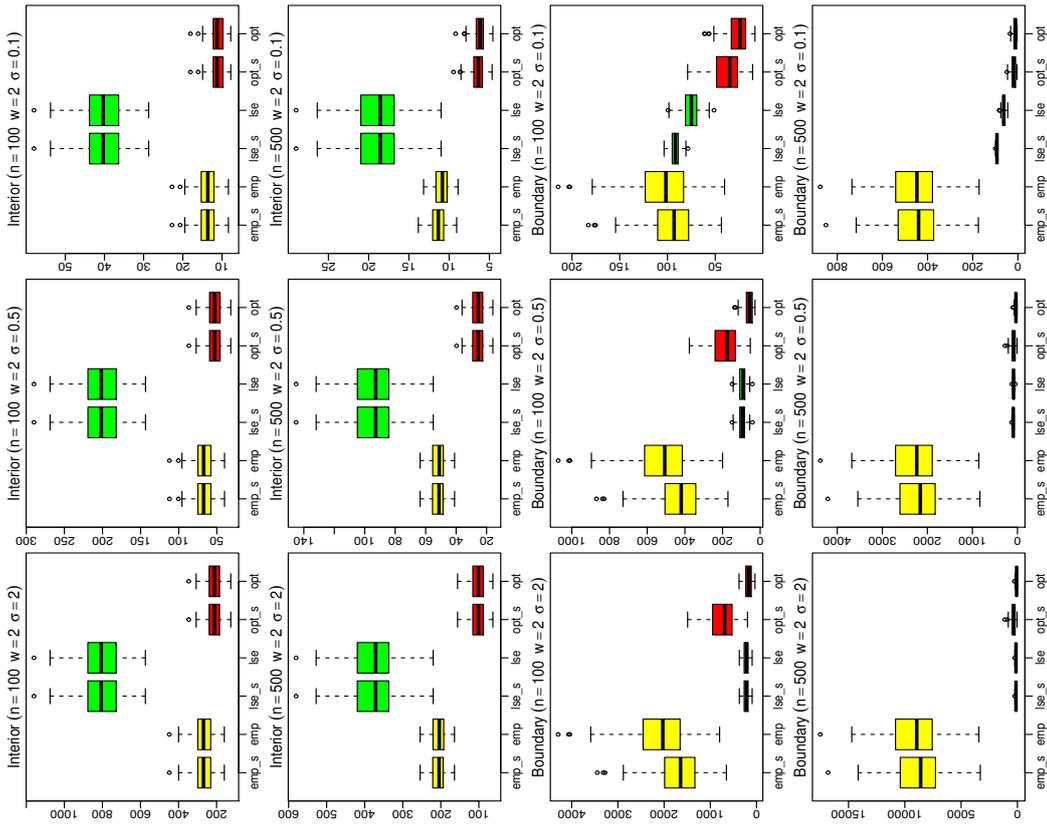


Figure 7: Comparison of the mean absolute errors on both interior (2 top rows) and boundary (2 bottom rows) points between the second-order derivative estimators with oracle tuning parameters and those with plug-in tuning parameters. \hat{m}''_{emp} , yellow box; \hat{m}''_{lse} , green box; \hat{m}''_{opt} , red box. “ s ” denotes estimators using plug-in parameters. $m(x) = 5 \sin(2\pi x)$ and $\varepsilon \sim N(0, \sigma^2)$.

The optimal sequence for estimating $m^{(p)}(x_{i+l})$ with a bias-reduction level q can be decided by solving the following optimization problem:

$$\begin{aligned} (d_0, \dots, d_r)_{pq} = & \underset{(d_0, \dots, d_r) \in \mathbb{R}^{r+1}}{\operatorname{argmin}} \sum_{k=0}^r d_k^2, \\ \text{s.t. } \sum_{k=0}^r d_k \frac{(x_{i+k} - x_{i+l})^j}{j!} = 0, \quad j = 0, \dots, p-1, p+1, \dots, q-1, \quad \sum_{k=0}^r d_k \frac{(x_{i+k} - x_{i+l})^p}{p!} = 1. \end{aligned}$$

The optimal difference sequences are adaptively chosen for each estimated point and they are no longer identical for all the interior design points. As a result, the parameter selection becomes more challenging and we leave this for future research. Finally, other models worthy of investigation include, for example, random design models (De Brabanter and Liu, 2015) and multivariate models (Charrigo et al., 2015; Charrigo and Srinivasan, 2015).

Acknowledgments

The authors thank the editor, the associate editor and the two referees for their constructive comments that led to a substantial improvement of the paper. The work of Wenlin Dai and Marc G. Genton was supported by King Abdullah University of Science and Technology (KAUST). Tiejun Tong's research was supported in part by Hong Kong Baptist University FRG grants FRG1/14-15/044, FRG2/15-16/038, FRG2/15-16/019 and FRG2/14-15/084.

Appendix A. Proof of Proposition 1

To find the optimal sequence for estimating the first-order derivative with q th-order accuracy, we solve the following optimization problem:

$$(d_0, \dots, d_r)_{1q} = \underset{(d_0, \dots, d_r) \in \mathbb{R}^{r+1}}{\operatorname{argmin}} \sum_{k=0}^r d_k^2, \quad \text{s.t.} \quad \text{condition (3) holds.}$$

It is easy to check that condition (3) is equivalent to

$$\sum_{k=0}^r d_k(k-l) = n \quad \text{and} \quad \sum_{k=0}^r d_k(k-l)^j = 0, \quad 0 \leq j \neq 1 \leq q-1.$$

To apply the Lagrange multipliers method to find the optimal sequence, we transform the above problem in the following unconstrained optimization problem:

$$f(d_0, \dots, d_r, \lambda_0, \dots, \lambda_{q-1}) = \sum_{k=0}^r d_k^2 + \lambda_0 C_0 + \sum_{j=2}^{q-1} \lambda_j \sum_{k=0}^r d_k(k-l)^j + \lambda_1 \left[\sum_{k=0}^r d_k(k-l) - n \right].$$

Taking the partial derivative of f with respect to each parameter and setting it to zero, we have

$$\begin{aligned} \frac{\partial f}{\partial d_k} &= 2d_k + \lambda_0 + \sum_{j=1}^{q-1} \lambda_j (k-l)^j = 0, \quad k = 0, \dots, r, \\ \frac{\partial f}{\partial \lambda_j} &= \sum_{k=0}^r d_k (k-l)^j = 0, \quad 0 \leq j \neq 1 \leq q-1, \\ \frac{\partial f}{\partial \lambda_1} &= \sum_{k=0}^r d_k (k-l) = n. \end{aligned} \tag{6}$$

We further make the following transformation:

$$\begin{aligned} \sum_{k=0}^r (k-l)^i \frac{\partial f}{\partial d_k} &= 2 \sum_{k=0}^r d_k (k-l)^i + \lambda_0 \sum_{k=0}^r (k-l)^i + \sum_{j=1}^{q-1} \lambda_j \sum_{k=0}^r (k-l)^{i+j} \\ &= I_i^{(l)} \lambda_0 + \sum_{j=1}^{q-1} I_{i+j}^{(l)} \lambda_j = 0, \quad 0 \leq i \neq 1 \leq q-1. \\ \sum_{k=0}^r (k-l) \frac{\partial f}{\partial d_k} &= 2 \sum_{k=0}^r d_k (k-l) + \lambda_0 \sum_{k=0}^r (k-l) + \sum_{j=1}^{q-1} \lambda_j \sum_{k=0}^r (k-l)^{1+j} \\ &= 2n + I_1^{(l)} \lambda_0 + \sum_{j=1}^{q-1} I_{1+j}^{(l)} \lambda_j = 0, \end{aligned}$$

where $I_i^{(l)} = \sum_{k=0}^r (k-l)^i$ for $i = 1, 2, \dots$.

These results can be expressed as a matrix equation,

$$U^{(l)}(\lambda_0, \dots, \lambda_{q-1})' = -2ne_2,$$

where $U^{(l)}$ is a $q \times q$ matrix with $u_{ij}^{(l)} = I_{i+j-2}^{(l)}$ and e_2 is a $q \times 1$ vector with the second element equal to 1 and the others equal to zero. Noting that $U^{(l)}$ is an invertible matrix, we have

$$(\lambda_0, \dots, \lambda_{q-1})' = -2nV_{(:,2)}^{(l)},$$

where $V^{(l)} = (U^{(l)})^{-1}$ and $V_{(:,2)}^{(l)}$ denotes the second column of $V^{(l)}$. This leads to $\lambda_j = -2pl_n V_{(j+1,2)}^{(l)}$ for $j = 0, \dots, q-1$. Combining this result with (6), we get

$$(d_k)_{1,q} = n \sum_{j=0}^{q-1} V_{(j+1,2)}^{(l)} (k-l)^j, \quad k = 0, \dots, r.$$

This completes the proof of Proposition 1. \square

Appendix B. Proof of Theorem 1

We can easily derive that

$$\begin{aligned} \text{var}[\hat{m}'_q(x_{i+t})] &= \sigma^2 \sum_{k=0}^r d_k^2 = \sigma^2 \sum_{k=0}^r d_k \left[\sum_{j=0}^{q-1} n V_{(j+1,2)}^{(l)}(k-l)^j \right] \\ &= \sigma^2 n \sum_{j=0}^{q-1} V_{(j+1,2)}^{(l)} \sum_{k=0}^r d_k (k-l)^j = \sigma^2 n V_{(2,2)}^{(l)} \sum_{k=0}^r d_k (k-l) \\ &= \sigma^2 n^2 V_{(2,2)}^{(l)}, \\ \text{bias}[\hat{m}'_q(x_{i+t})] &= C_{q,l} m^{(q)}(x_{i+t}) + o(r^{q-1}/n^{q-1}), \end{aligned}$$

where

$$\begin{aligned} C_{q,l} &= \sum_{k=0}^r d_k \frac{(k-l)^q}{n^q q!} = \frac{n}{q! n^q} \sum_{k=0}^r \left[\sum_{j=0}^{q-1} V_{(j+1,2)}^{(l)}(k-l)^j \right] (k-l)^q \\ &= \frac{1}{q! n^{q-1}} \sum_{j=0}^{q-1} V_{(j+1,2)}^{(l)} I_{j+q}^{(l)} = O(r^{q-1}/n^{q-1}). \end{aligned}$$

This completes the proof of Theorem 1. \square

Appendix C. Proof of Proposition 2

To find the optimal sequence for estimating the p th-order derivative with q th-order accuracy, we solve the following optimization problem:

$$(d_0, \dots, d_r)_{p,q} = \underset{(d_0, \dots, d_r) \in \mathbb{R}^{r+1}}{\text{argmin}} \sum_{k=0}^r d_k^2, \quad \text{s.t.} \quad \text{condition (4) holds.}$$

It is easy to check that condition (4) is equivalent to

$$\sum_{k=0}^r d_k (k-l)^p = p! n^p \quad \text{and} \quad \sum_{k=0}^r d_k (k-l)^j = 0, \quad 0 \leq j \neq p \leq q-1.$$

To apply the Lagrange multipliers method to find the optimal sequence, we transform the above problem in the following unconstrained optimization problem:

$$\begin{aligned} f(d_0, \dots, d_r, \lambda_0, \dots, \lambda_{q-1}) &= \sum_{k=0}^r d_k^2 + \lambda_0 C_0 + \left(\sum_{j=1}^{p-1} + \sum_{j=p+1}^{q-1} \right) \lambda_j \sum_{k=0}^r d_k (k-l)^j \\ &\quad + \lambda_p \left[\sum_{k=0}^r d_k (k-l)^p - p! n^p \right]. \end{aligned}$$

Taking the partial derivative of f with respect to each parameter and setting it to zero, we have

$$\begin{aligned} \frac{\partial f}{\partial d_k} &= 2d_k + \lambda_0 + \sum_{j=1}^{q-1} \lambda_j (k-l)^j = 0, \quad k = 0, \dots, r, \\ \frac{\partial f}{\partial \lambda_j} &= \sum_{k=0}^r d_k (k-l)^j = 0, \quad 0 \leq j \neq p \leq q-1, \\ \frac{\partial f}{\partial \lambda_p} &= \sum_{k=0}^r d_k (k-l)^p = p! n^p. \end{aligned} \quad (7)$$

We further make the following transformation:

$$\begin{aligned} \sum_{k=0}^r (k-l)^i \frac{\partial f}{\partial d_k} &= 2 \sum_{k=0}^r d_k (k-l)^i + \lambda_0 \sum_{k=0}^r (k-l)^i + \sum_{j=1}^r \lambda_j \sum_{k=0}^r (k-l)^{i+j} \\ &= I_i^{(l)} \lambda_0 + \sum_{j=1}^{q-1} I_{i+j}^{(l)} \lambda_j = 0, \quad 0 \leq i \neq p \leq q-1. \\ \sum_{k=0}^r (k-l)^p \frac{\partial f}{\partial d_k} &= 2 \sum_{k=0}^r d_k (k-l)^p + \lambda_0 \sum_{k=0}^r (k-l)^p + \sum_{j=1}^r \lambda_j \sum_{k=0}^r (k-l)^{p+j} \\ &= 2p! n^p + I_p^{(l)} \lambda_0 + \sum_{j=1}^{q-1} I_{p+j}^{(l)} \lambda_j = 0, \end{aligned}$$

where $I_i^{(l)} = \sum_{k=0}^r (k-l)^i$ for $i = 1, 2, \dots$.

These results can be expressed as a matrix equation,

$$U^{(l)}(\lambda_0, \dots, \lambda_{q-1})' = -2p! n^p \epsilon_{p+1},$$

where $U^{(l)}$ is a $q \times q$ matrix with $u_{ij}^{(l)} = I_{i+j-2}^{(l)}$ and ϵ_{p+1} is a $q \times 1$ vector with the $(p+1)$ th element equal to 1 and the others equal to zero. Noting that $U^{(l)}$ is an invertible matrix, we have

$$(\lambda_0, \dots, \lambda_{q-1})' = -2p! n^p V_{(-,p+1)}^{(l)},$$

where $V^{(l)} = (U^{(l)})^{-1}$ and $V_{(-,p+1)}^{(l)}$ denotes the $(p+1)$ th column of $V^{(l)}$. This leads to $\lambda_j = -2p! n^p V_{(j+1,p+1)}^{(l)}$ for $j = 0, \dots, q-1$. Combining this result with (7), we get

$$(d_k)_{p,q} = p! n^p \sum_{j=0}^{q-1} V_{(j+1,p+1)}^{(l)} (k-l)^j, \quad k = 0, \dots, r.$$

This completes the proof of Proposition 2. \square

Appendix D. Proof of Theorem 2

We can easily derive that

$$\begin{aligned} \text{var}[\hat{m}_q^{(p)}(x_{i+1})] &= \sigma^2 \sum_{k=0}^r d_k^2 = \sigma^2 \sum_{k=0}^r d_k \left[\sum_{j=0}^{q-1} p!n^p V_{(j+1,p+1)}^{(l)}(k-l)^j \right] \\ &= \sigma^2 p!n^p \sum_{j=0}^{q-1} V_{(j+1,p+1)}^{(l)} \sum_{k=0}^r d_k (k-l)^j = \sigma^2 p!n^p V_{(p+1,p+1)}^{(l)} \sum_{k=0}^r d_k (k-l)^p \\ &= \sigma^2 (p!)^2 n^{2p} V_{(p+1,p+1)}^{(l)}, \\ \text{bias}[\hat{m}_q^{(p)}(x_{i+1})] &= C_{q,l} m^{(q)}(x_{i+1}) + o(r^{q-p}/n^{q-p}), \end{aligned}$$

where

$$\begin{aligned} C_{q,l} &= \sum_{k=0}^r d_k \frac{(k-l)^q}{n^q q!} = \frac{p!n^p}{q!n^q} \sum_{k=0}^r \left[\sum_{j=0}^{q-1} V_{(j+1,p+1)}^{(l)}(k-l)^j \right] (k-l)^q \\ &= \frac{p!}{q!n^{q-p}} \sum_{j=0}^{q-1} V_{(j+1,p+1)}^{(l)} I_{j+q}^{(l)} = O(r^{q-p}/n^{q-p}). \end{aligned}$$

This completes the proof of Theorem 2. \square

References

- Graciela Boente and Daniela Rodriguez. Robust estimators of high order derivatives of regression functions. *Statistics and Probability Letters*, 76(13):1335–1344, 2006.
- Guangun Cao. Simultaneous confidence bands for derivatives of dependent functional data. *Electronic Journal of Statistics*, 8(2):2639–2663, 2014.
- Richard Charrnigo and Cidambi Srinivasan. A multivariate generalized C_p and surface estimation. *Biostatistics*, 16(2):311–325, 2015.
- Richard Charrnigo, Benjamin Hall, and Cidambi Srinivasan. A generalized C_p criterion for derivative estimation. *Technometrics*, 53(3):238–253, 2011.
- Richard Charrnigo, Limin Feng, and Cidambi Srinivasan. Nonparametric and semiparametric compound estimation in multiple covariates. *Journal of Multivariate Analysis*, 141:179–196, 2015.
- Kris De Brabanter and Yu Liu. *Smoothed nonparametric derivative estimation based on weighted difference sequences*. Stochastic Models, Statistics and Their Applications, A. Steland and E. Rafajlowicz and K. Szajowski (Eds.), Chapter 4 (31–38). Springer, 2015.
- Kris De Brabanter, Jos De Brabanter, Bart De Moor, and Irène Gijbels. Derivative estimation with local polynomial fitting. *Journal of Machine Learning Research*, 14(1):281–301, 2013.
- Randall L. Eubank and Paul L. Speckman. Confidence bands in nonparametric regression. *Journal of the American Statistical Association*, 88(424):1287–1301, 1993.
- Jiangping Fan and Irène Gijbels. Data-driven bandwidth selection in local polynomial fitting: variable bandwidth and spatial adaptation. *Journal of the Royal Statistical Society Series B*, 57:371–394, 1995.
- Jiangping Fan and Irène Gijbels. *Local Polynomial Modelling and Its Applications*. CRC Press, 1996.
- Theo Gasser and Hans G. Müller. Estimating regression functions and their derivatives by the kernel method. *Scandinavian Journal of Statistics*, 11:171–185, 1984.
- Wolfgang Härdle. *Applied Nonparametric Regression*. Cambridge University Press, 1990.
- Tae Y. Kim, Byeong U. Park, Myung S. Moon, and Chinho Kim. Using binodal kernel for inference in nonparametric regression with correlated errors. *Journal of Multivariate Analysis*, 100:1487–1497, 2009.
- Soumendra Nath Lahiri. *Resampling Methods for Dependent Data*. Springer, 2003.
- Hans G. Müller, Ulrich Stadtmüller, and Thomas Schmitt. Bandwidth choice and confidence intervals for derivatives of noisy data. *Biometrika*, 74:743–749, 1987.
- Jorge L. Ojeda Cabrera. `locpol`: Kernel local polynomial regression. URL <http://mirrors.usc.edu.cn/CRAN/web/packages/locpol/index.html>, 2012.
- Jean Opsomer, Yuedong Wang, and Yuhong Yang. Nonparametric regression with correlated errors. *Statistical Science*, 16:134–153, 2001.
- Cheolwoo Park and Kee H. Kang. Sizer analysis for the comparison of regression curves. *Computational Statistics and Data Analysis*, 52(8):3954–3970, 2008.
- James O. Ramsay. *Functional Data Analysis*. Wiley, 2006.
- James O. Ramsay and Bernard W Silverman. *Applied Functional Data Analysis: Methods and Case Studies*. Springer, 2002.
- David Ruppert, Simon J. Sheather, and Matthew P. Wand. An effective bandwidth selector for local least squares regression. *Journal of the American Statistical Association*, 90(432):1257–1270, 1995.
- Charles J. Stone. Optimal rates of convergence for nonparametric estimators. *The Annals of Statistics*, 8:1348–1360, 1980.
- Tiejun Tong and Yuedong Wang. Estimating residual variance in nonparametric regression using least squares. *Biometrika*, 92:821–830, 2005.
- Tiejun Tong, Yanyuan Ma, and Yuedong Wang. Optimal variance estimation without estimating the mean function. *Bernoulli*, 19(5A):1839–1854, 2013.

Grace Wahba. *Spline Models for Observational Data*. SIAM, 1990.

Wenwu Wang and Lu Lin. Derivative estimation based on difference sequence via locally weighted least squares regression. *Journal of Machine Learning Research*, 16:2617–2641, 2015.

Shanggang Zhou and Douglas A Wolfe. On derivative estimation in spline regression. *Statistica Sinica*, 10(1):93–108, 2000.

Double or Nothing: Multiplicative Incentive Mechanisms for Crowdsourcing

Nihar B. Shah

*Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, CA 94720 USA*

NIHAR@EECS.BERKELEY.EDU

Dengyong Zhou

*Machine Learning Department
Microsoft Research
One Microsoft Way, Redmond 98052 USA*

DENGYONG.ZHOU@MICROSOFT.COM

Editor: Qiang Liu

Abstract

Crowdsourcing has gained immense popularity in machine learning applications for obtaining large amounts of labeled data. Crowdsourcing is cheap and fast, but suffers from the problem of low-quality data. To address this fundamental challenge in crowdsourcing, we propose a simple payment mechanism to incentivize workers to answer only the questions that they are sure of and skip the rest. We show that surprisingly, under a mild and natural “no-free-lunch” requirement, this mechanism is the one and only incentive-compatible payment mechanism possible. We also show that among all possible incentive-compatible mechanisms (that may or may not satisfy no-free-lunch), our mechanism makes the smallest possible payment to spammers. We further extend our results to a more general setting in which workers are required to provide a quantized confidence for each question. Interestingly, this unique mechanism takes a “multiplicative” form. The simplicity of the mechanism is an added benefit. In preliminary experiments involving over 900 worker-task pairs, we observe a significant drop in the error rates under this unique mechanism for the same or lower monetary expenditure.

Keywords: high-quality labels, supervised learning, crowdsourcing, mechanism design, proper scoring rules

1. Introduction

Complex machine learning tools such as deep learning are gaining increasing popularity and are being applied to a wide variety of problems. These tools require large amounts of labeled data (Hinton et al., 2012; Raykar et al., 2010; Deng et al., 2009; Carlson et al., 2010). These large labeling tasks are being performed by coordinating crowds of semi-skilled workers through the Internet. This is known as crowdsourcing. Generating large labeled data sets through crowdsourcing is inexpensive and fast as compared to employing experts. Furthermore, given the current platforms for crowdsourcing such as Amazon Mechanical Turk and many others, the initial overhead of setting up a crowdsourcing task is minimal. Crowdsourcing as a means of collecting labeled training data has now become indispensable to the engineering of intelligent systems. The crowdsourcing of labels is also often used to supplement automated algorithms, to perform the tasks that are too difficult

to accomplish by machines alone (Khatib et al., 2011; Lang and Rio-Ross, 2011; Bernstein et al., 2010; Von Ahn et al., 2008; Franklin et al., 2011).

Most workers in crowdsourcing are not experts. As a consequence, labels obtained from crowdsourcing typically have a significant amount of error (Kazai et al., 2011; Vuurens et al., 2011; Wais et al., 2010). It is not surprising that there is significant emphasis on having higher quality labeled data for machine learning algorithms, since a higher amount of noise implies requirement of more labels for obtaining the same accuracy in practice. Moreover, several algorithms and settings are not very tolerant of data that is noisy (Long and Servidio, 2010; Hanneke and Yang, 2010; Manwani and Sastry, 2013; Baldrige and Palmer, 2009); for instance, Long and Servidio (2010) conclude that “a range of different types of boosting algorithms that optimize a convex potential function satisfying mild conditions cannot tolerate random classification noise.” Recent efforts have focused on developing statistical techniques to post-process the noisy labels in order to improve its quality (e.g., Raykar et al., 2010; Zhou et al., 2012; Chen et al., 2013; Dawid and Skene, 1979; Karger et al., 2011; Liu et al., 2012; Zhang et al., 2014; Ipeitotis et al., 2014; Zhou et al., 2015; Khetan and Oh, 2016; Shah et al., 2016c). However, when the inputs to these algorithms are highly erroneous, it is difficult to guarantee that the processed labels will be reliable enough for subsequent use by machine learning or other applications. In order to avoid “garbage in, garbage out”, we take a complementary approach to this problem: cleaning the data at the time of collection.

We consider crowdsourcing settings where the workers are paid for their services, such as in the popular crowdsourcing platforms of Amazon Mechanical Turk (mturk.com), Crowdflower (crowdflower.com) and other commercial platforms, as well as internal crowdsourcing platforms of companies such as Google, Facebook and Microsoft. These commercial platforms have gained substantial popularity due to their support for a diverse range of tasks for machine learning labeling, varying from image annotation and text recognition to speech captioning and machine translation. We consider problems that are objective in nature, that is, have a definite answer. Figure 1a depicts an example of such a question where the worker is shown a set of images, and for each image, the worker is required to identify if the image depicts the Golden Gate Bridge.

Our approach builds on the simple insight that in typical crowdsourcing setups, workers are simply paid in proportion to the amount of tasks they complete. As a result, workers attempt to answer questions that they are not sure of, thereby increasing the error rate of the labels. For the questions that a worker is not sure of, her answers could be very unreliable (Wais et al., 2010; Kazai et al., 2011; Vuurens et al., 2011; Jagabathula et al., 2014). To ensure acquisition of only high-quality labels, we wish to encourage the worker to skip the questions about which she is unsure, for instance, by providing an explicit “I’m not sure” option for every question (see Figure 1b). Given this additional option, one must also ensure that the worker is indeed incentivized to skip the questions that she is not confident about. In a more general form, we consider eliciting the confidence of the worker for each question at multiple levels. For instance, in addition to “I’m not sure”, we may also provide options like “absolutely sure”, and “moderately sure” (see Figure 1c). The goal is to design payment mechanisms that incentivize the worker to attempt only those questions for which they are confident enough, or alternatively, report their confidences truthfully. As we will see later, this significantly improves the aggregate quality of the labels that are input to the machine learning algorithms. We will term any payment mechanism that incentivizes the worker to do so as “incentive compatible”.

In addition to incentive compatibility, preventing spammers is another desirable requirement from incentive mechanisms in crowdsourcing. Spammers are workers who answer randomly with-

<p>a Is this the Golden Gate Bridge?</p>  <p><input type="radio"/> Yes <input type="radio"/> No</p>	<p>b Is this the Golden Gate Bridge?</p>  <p><input type="radio"/> Yes <input type="radio"/> No <input type="radio"/> I'm not sure</p>
<p>c Is this the Golden Gate Bridge?</p>  <p><input type="radio"/> Yes <input type="radio"/> Moderately sure <input type="radio"/> Absolutely sure <input type="radio"/> No <input type="radio"/> Moderately sure <input type="radio"/> Absolutely sure <input type="radio"/> I'm not sure</p>	

Figure 1: Different interfaces for a task that requires the worker to answer the question “Is this the Golden Gate Bridge?”: (a) the conventional interface; (b) with an option to skip; (c) with multiple confidence levels.

out regard to the question being asked, in the hope of earning some free money, and are known to exist in large numbers on crowdsourcing platforms (Wais et al., 2010; Bohannon, 2011; Kazai et al., 2011; Vuurens et al., 2011). The presence of spammers can significantly affect the performance of any machine learning algorithm that is trained on this data. It is thus of interest to deter spammers by paying them as low as possible. An intuitive objective, to this end, is to ensure a minimum possible payment to spammers who answer randomly. For instance, in a task with binary-choice questions, a spammer is expected to have half of the attempted answers incorrect; one may thus wish to set the payment to its minimum possible value if half or more of the attempted answers are wrong. In this paper, however, we impose *strictly and significantly weaker requirement*, and then show that there is one and only one incentive-compatible mechanism that can satisfy this weak requirement. Our requirement is referred to as the “no-free-lunch” axiom. In the skip-based setting, it says that if *all* the questions attempted by the worker are answered incorrectly, then the payment must be the minimum possible. The no-free-lunch axiom for the general confidence-based setting is even weaker: if the worker indicates the highest confidence level for *all* the questions she attempts in the gold standard, and furthermore if all these responses are incorrect, then the payment must be the minimum possible. We term this condition the “no-free-lunch” axiom. In the general confidence-based setting, we want to make the minimum possible payment if the worker indicates the *highest confidence level* for *all* the questions she attempts *and* if *all* these responses are incorrect.

In order to test whether our mechanism is practically viable, and to assess the quality of the final labels obtained, we conducted experiments on the Amazon Mechanical Turk crowdsourcing platform. In our preliminary experiments that involved several hundred workers, we found that the quality of data consistently improved by use of our schemes as compared to the standard settings, often by two-fold or higher, with the total monetary expenditure being the same or lower as compared to the conventional baseline.

1.1 Summary of Contributions

We propose a payment mechanism for the aforementioned setting (“incentive compatibility” plus “no-free-lunch”), and show that surprisingly, this is the *only* possible mechanism. We also show that additionally, our mechanism makes the smallest possible payment to spammers among all possible incentive compatible mechanisms that may or may not satisfy the no-free-lunch axiom. Interestingly, our payment mechanism takes a multiplicative form: the evaluation of the worker’s response to each question is a certain score, and the final payment is a *product* of these scores. This mechanism has additional appealing features in that it is simple to compute, and is also simple to explain to the workers. Our mechanism is applicable to any type of objective questions, including multiple choice annotation questions, transcription tasks, etc. In preliminary experiments on Amazon Mechanical Turk involving over 900 worker-task pairs, the quality of data improved significantly under our unique mechanism, with the total monetary expenditure being the same or lower as compared to the conventional baseline.

1.2 Related Literature

The framework of “strictly proper scoring rules” (Brier, 1950; Savage, 1971; Gneiting and Raftery, 2007; Lambert and Shoham, 2009) provides a general theory for eliciting information for settings where this information can subsequently be verified by the mechanism designer, for example, by observing the true value some time in the future. In our work, this verification is performed *via* the presence of some “gold standard” questions in the task. Consequently, our mechanisms can also be called “strictly proper scoring rules”. It is important to note that the framework of strictly proper scoring rules, however, provides a large collection of possible mechanisms and does not guide the choice of a specific mechanism from this collection (Gneiting and Raftery, 2007). In this work, we show that for the crowdsourcing setups considered, under a very mild condition we term the “no-free-lunch” axiom, the mechanism proposed in this paper is the one and only strictly proper scoring rule.

Interestingly, proper scoring rules have another interesting connection with machine learning techniques: quoting Bija et al. (2005), “proper scoring rules comprise most loss functions currently in use: log-loss, squared error loss, boosting loss, and as limiting cases cost-weighted misclassification losses.” The present paper does not investigate this aspect of proper scoring rules, and we refer the reader to Biihmann and Hothorn (2007); Mease et al. (2007); Bija et al. (2005) for more details.

In this paper, we assume the existence of some gold standard questions whose answers are known a priori to the system designer. As a result, the payment to a worker is determined solely by her own work. There are settings where gold standard questions may not be available, for instance, when obtaining gold standard questions is too expensive, or when the questions pertain to subjective preferences (Shah and Wainwright, 2015; Shah et al., 2016b; Chen et al., 2016) instead of labeling data. A parallel line of literature (Miller et al., 2005; Dasgupta and Ghosh, 2013; Prelec, 2004; Wolfers and Zitzewitz, 2004; Conitzer, 2009) addresses such settings without gold standard questions. The idea in the mechanisms designed therein is to reward the agents based on certain criteria that compares certain elicited data from the agents with each other, and typically involves asking agents to predict other agents’ responses. The mechanisms designed often provide weaker guarantees (such as that of truth-telling being a Nash equilibrium) due to the absence of a gold standard answer to compare with. This line of literature includes work on peer-prediction (Miller

et al., 2005; Dasgupta and Ghosh, 2013), the Bayesian truth serum (Prelec, 2004) and prediction markets (Wolfers and Zitewitz, 2004; Conitzer, 2009).

The design of statistical inference algorithms for denoising the data obtained from workers is an active topic of research (Raykar et al., 2010; Zhou et al., 2012; Wauthier and Jordan, 2011; Chen et al., 2013; Khetan and Oh, 2016; Dawid and Skene, 1979; Karger et al., 2011; Liu et al., 2012; Zhang et al., 2014; Vempaty et al., 2014; Ipeirotis et al., 2014; Zhou et al., 2015; Shah et al., 2016c). In addition, several machine learning algorithms accommodating errors in the data have also been designed (Angluin and Laird, 1988; Cano et al., 2001; Lee et al., 2004; Chu et al., 2004). These algorithms are typically oblivious to the elicitation procedure. Our work nicely complements this line of research in that these inference algorithms may now additionally employ the higher quality data and the specific structure of the elicited data for an improved denoising efficiency.

Another relevant problem in crowdsourcing is that of choosing which workers to hire or efficiently matching workers to tasks, and such problems are studied in Yuen et al. (2011); Ho et al. (2013); Zhou et al. (2014); Anari et al. (2014) under different contexts. Our work assumes that a worker is already matched, and focuses on incentivizing that worker to respond in a certain manner. A recent line of work has focused on elicitation of data from multiple agents in order to perform certain specific estimation tasks (Fang et al., 2007; Dekel et al., 2008; Cai et al., 2015). In contrast, our goal is to ensure that workers censor their own low-quality (raw) data, without restricting our attention to any specific downstream algorithm or task.

1.3 Organization

The organization of this paper is as follows. We present the formal problem setting in Section 2. In Section 3 we consider the skip-based setting: We present our proposed mechanism and show that it is the only mechanism which satisfies the requirements discussed above. In Section 4, we then consider the more general setting of eliciting a quantized value of the worker’s confidence. We construct a mechanism for this setting, which also takes a multiplicative form, and prove its uniqueness. In Section 5 we prove that imposing a requirement that is only slightly stronger than our proposed no-free-lunch axiom leads to impossibility results. In Section 6 we present synthetic simulations and real-world experiments on Amazon Mechanical Turk to evaluate the potential of our setting and algorithm to work in practice. We conclude the paper with a discussion on the various modeling choices, future work, and concluding remarks in Section 7.

The paper contains three appendices. In Appendix A we prove all theoretical results whose proofs are not presented in the main text. We provide more details of the experiments in Appendix B. In Appendix C we extend our results to a setting where workers aim to maximize the expected value of some “utility” of their payments.

2. Setting and Notation

In the crowdsourcing setting that we consider, one or more workers perform a *task*, where a task consists of multiple *questions*. The questions are objective, by which we mean, each question has precisely one correct answer. Examples of objective questions include multiple-choice classification questions such as Figure 1, questions on transcribing text from audio or images, etc.

For any possible answer to any question, we define the worker’s *confidence about an answer* as the probability, according to her belief, of this answer being correct. In other words, one can assume that the worker has (in her mind) a probability distribution over all possible answers to a question,

and the confidence for an answer is the probability of that answer being correct. As a shorthand, we also define the *confidence about a question* as the confidence for the answer that the worker is most confident about for that question. We assume that the worker’s confidences for different questions are independent. Our goal is that for every question, the worker should be incentivized to skip if her confidence for that question is below a certain pre-defined threshold, otherwise select the answer that she is most confident about, and if asked, also indicate a correct (quantized) value of her confidence for the answer.

Specifically, we consider two settings:

- **Skip-based.** For each question, the worker can either choose to ‘skip’ the question or provide an answer (Figure 1b).
- **Confidence-based.** For each question, the worker can either ‘skip’ the question or provide an answer, and in the latter case, indicate her confidence for this answer as a number in $\{1, \dots, L\}$ (Figure 1c). We term this indicated confidence as the ‘confidence-level’. Here, L represents the highest confidence-level, and ‘skip’ is considered to be a confidence-level of 0.¹

One can see from the aforementioned definition that the confidence-based setting is a generalization of the skip-based setting (the skip-based setting corresponds to $L = 1$). The goal is to ensure that for a given set of intervals that partition $[0, 1]$, for every question the worker is incentivized to indicate ‘skip’ or choose the appropriate confidence-level when her confidence for that question falls in the corresponding interval. The choice of these intervals will be defined subsequently in the skip-based and confidence-based sections (Section 3 and Section 4) respectively.

Let N denote the total number of questions in the task. Among these questions, we assume the existence of some “gold standard” questions, that is, a set of questions whose answers are known to the requester. Let G ($1 \leq G \leq N$) denote the number of gold standard questions. The G gold standard questions are assumed to be distributed uniformly at random in the pool of N questions (of course, the worker does not know which G of the N questions form the gold standard). The payment to a worker for a task is computed after receiving her responses to all the questions in the task. The payment is based on the worker’s performance on the gold standard questions. Since the payment is based on known answers, the payments to different workers do not depend on each other, thereby allowing us to consider the presence of only one worker without any loss in generality.

We will employ the following standard notation. For any positive integer K , the set $\{1, \dots, K\}$ is denoted by $[K]$. The indicator function is denoted by $\mathbf{1}$, i.e., $\mathbf{1}\{z\} = 1$ if z is true, and 0 otherwise. Let x_1, \dots, x_G denote the evaluations of the answers that the worker gives to the G gold standard questions, and let f denote the scoring rule, i.e., a function that determines the payment to the worker based on these evaluations x_1, \dots, x_G .

In the skip-based setting, $x_i \in \{-1, 0, +1\}$ for all $i \in [G]$. Here, “0” denotes that the worker skipped the question, “-1” denotes that the worker attempted to answer the question and that answer was incorrect, and “+1” denotes that the worker attempted to answer the question and that answer was correct. The payment function is $f : \{-1, 0, +1\}^G \rightarrow \mathbb{R}$.

In the confidence-based setting, $x_i \in \{-L, \dots, +L\}$ for all $i \in [G]$. Here, we set $x_i = 0$ if the worker skipped the question, and for $l \in \{1, \dots, L\}$, we set $x_i = l$ if the question was answered

1. When the task is presented to the workers, the word ‘skip’ or the numbers $\{1, \dots, L\}$ are replaced by more comprehensible phrases such as “I don’t know”, “moderately sure”, “absolutely sure”, etc.

correctly with confidence l and $x_i = -l$ if the question was answered incorrectly with confidence l . The function $f : \{-L, \dots, +L\}^G \rightarrow \mathbb{R}$ specifies the payment to be made to the worker.

The payment is further associated to two parameters, μ_{\max} and μ_{\min} . The parameter μ_{\max} denotes the *budget*, i.e., the maximum amount that is paid to any individual worker for this task:

$$\max_{x_1, \dots, x_G} f(x_1, \dots, x_G) = \mu_{\max}.$$

The amount μ_{\max} is thus the amount of compensation paid to a perfect worker for her work. Further, one may often also have the requirement of paying a certain minimum amount to any worker. The parameter μ_{\min} ($\leq \mu_{\max}$) denotes this minimum payment: the payment function must also satisfy

$$\min_{x_1, \dots, x_G} f(x_1, \dots, x_G) \geq \mu_{\min}.$$

For instance, crowdsourcing platforms today allow payments to workers, but do not allow imposing penalties: this condition gives $\mu_{\min} = 0$.

We assume that the worker attempts to maximize her overall expected payment. In what follows, the expression ‘the worker’s expected payment’ will refer to the expected payment from the worker’s point of view, and the expectation will be taken with respect to the worker’s confidences about her answers and the uniformly random choice of the G gold standard questions among the N questions in the task. A payment function f is called *incentive compatible* if the expected payment of the worker under this payment function is *strictly* maximized when the worker answers in the manner desired.² The specific requirements of the skip-based and the confidence-based settings are discussed subsequently in their respective sections to follow. In the remainder of this section, we formally define the concepts of the worker’s expected payment and incentive compatibility; the reader interested in understanding the paper at a higher level may skip directly to the next section without loss in continuity.

Let Ω denote the set of options for each question. We assume that Ω is a finite set, for instance, the set $\{\text{Yes}, \text{No}\}$ for a task with binary-choice questions, or the set of all strings of at most a certain length for a task with textual responses. Let $Q \in [0, 1]^{|\Omega| \times N}$ denote the beliefs of a worker for the N questions asked. Specifically, for any question $i \in [N]$ and any option $\omega \in \Omega$, let $Q_{\omega, i}$ represent the probability, according to the worker’s belief, that option ω is the correct answer to question i . Then from the law of total probability, any valid Q must have $\sum_{\omega \in \Omega} Q_{\omega, i} = 1$ for every $i \in [N]$. The value of Q is unknown to the mechanism.

Let us first define the notion of the expected payment (from the worker’s point of view) for any given response of the worker to the questions. For any question $i \in [N]$, suppose the worker indicates the confidence-level $\xi_i \in \{0, \dots, L\}$. For every question $i \in [N]$ such that $\xi_i \neq 0$, let $\omega_i \in \Omega$ denote the option selected by the worker; whenever $\xi_i = 0$, indicating a skip, we let ω_i take any arbitrary value in Ω . Furthermore, let $p_i = Q_{\omega_i, i}$ denote the probability, according to the worker’s belief, that the chosen option ω_i is the correct answer to question i . For notational purposes, we also define a vector $E = (\epsilon_1, \dots, \epsilon_G) \in \{-1, 1\}^G$. Then for the given responses, for the worker beliefs Q , and under payment mechanism f , the worker’s expected payment $\Gamma_{Q, f} : (\{0, \dots, L\} \times \Omega)^N \rightarrow \mathbb{R}$

is given by the expression:

$$\begin{aligned} \Gamma_{Q, f}(\xi_1, \omega_1, \dots, \xi_N, \omega_N) &= \frac{1}{\binom{L}{G}} \sum_{\substack{(j_1, \dots, j_G) \\ \subseteq \{1, \dots, N\}}} \sum_{E \in \{-1, 1\}^G} \left(f(\epsilon_1 \xi_{j_1}, \dots, \epsilon_G \xi_{j_G}) \prod_{i=1}^G (p_{j_i})^{\frac{1+\epsilon_i}{2}} (1-p_{j_i})^{\frac{1-\epsilon_i}{2}} \right). \end{aligned} \quad (1)$$

In the expression (1), the outermost summation corresponds to the expectation with respect to the randomness arising from the unknown positions of the gold standard questions. The inner summation corresponds to the expectation with respect to the worker’s beliefs about the correctness of her responses. Note that the right hand side of (1) implicitly depends on $(\omega_1, \dots, \omega_N)$ through the values (p_1, \dots, p_N) . Also note that for every question i such that $\xi_i = 0$, the right hand side of (1) does not depend on the values of ω_i and p_i ; this is because the choice $\xi_i = 0$ of skipping question i implies that the worker did not select any particular option.

We will now use the the definition of the expected payment of the worker to define the notion of incentive compatibility. To this end, for any valid probabilities Q , let $\mathcal{A}(Q) \subseteq \{0, \dots, L\} \times \Omega^N$ denote an associated set of ‘‘desired’’ responses. By this we mean that every $a \in \{0, \dots, L\} \times \Omega^N$ represents a possible response to the set of N questions, and the goal is to incentivize the worker to provide any one response in the set $\mathcal{A}(Q)$. Then a mechanism f is termed *incentive compatible* if

$$\Gamma_{Q, f}(a) > \Gamma_{Q, f}(a') \quad \text{for every } a \in \mathcal{A}(Q), \text{ every } a' \notin \mathcal{A}(Q), \text{ and every valid } Q.$$

The goal is to design mechanisms that are incentive compatible, that is, incentivize the workers to respond in a certain manner. The specific choice of ‘‘desired responses’’ for the skip-based and the confidence-based settings are discussed subsequently in their respective sections. We begin with the skip-based setting.

3. Skip-based Setting

In this section, we consider the setting where for every question, the worker can choose to either answer the question or to skip it; no additional information is asked from the worker. See Figure 1b for an illustration.

3.1 Setting

Let $T \in (0, 1)$ be a predefined value. The goal is to design payment mechanisms that incentivize the worker to skip the questions for which her confidence is lower than T , and answer those for which her confidence is higher than T .³ Moreover, for the questions that she attempts to answer, she must be incentivized to select the answer that she believes is most likely to be correct. The value of T is chosen a priori based on factors such as budget constraints, the targeted quality of labels, and/or the choice of the algorithm used to subsequently aggregate the responses of multiple workers. In this paper, we assume that the value of the threshold T is specified to us.

Now the space of all possible mechanisms for this problem may be rather wide. Thus in order to narrow down our search, we impose the following additional simple and natural requirement:

2. Such a notion of incentive compatibility is often called ‘‘strict incentive compatibility’’, we drop the prefix term ‘‘strict’’ for brevity.

3. In the event that the confidence about a question is exactly equal to T , the worker may choose to answer or skip.

Axiom 1 (No-free-lunch Axiom) *If all the answers attempted by the worker in the gold standard are wrong, then the payment is the minimum possible. More formally, $f(x_1, \dots, x_G) = \mu_{\min}$ for every evaluation (x_1, \dots, x_G) such that $0 < \sum_{i=1}^G \mathbf{1}\{x_i \neq 0\} = \sum_{i=1}^G \mathbf{1}\{x_i = -1\}$.*

One may expect a payment mechanism to impose the restriction of minimum payment to spammers who answer randomly. For instance, in a task with binary-choice questions, a spammer is expected to have 50% of the attempted answers incorrect; one may thus wish to set a the minimum possible payment if 50% or more of the attempted answers were incorrect. The no-free-lunch axiom which we impose is however a *significantly weaker condition*, mandating minimum payment if *all* attempted answers are incorrect.

3.2 Payment Mechanism

We now present our proposed payment mechanism in Algorithm 1.

Algorithm 1: Incentive mechanism for skip-based setting

- Inputs:
 - ▶ Threshold T
 - ▶ Budget parameters μ_{\max} and μ_{\min}
 - ▶ Evaluations $(x_1, \dots, x_G) \in \{-1, 0, +1\}^G$ of the worker's answers to the G gold standard questions
- Set $\alpha_{-1} = 0$, $\alpha_0 = 1$, $\alpha_{+1} = \frac{1}{T}$
- The payment is

$$f(x_1, \dots, x_G) = \kappa \prod_{i=1}^G \alpha_{x_i} + \mu_{\min},$$

where $\kappa = (\mu_{\max} - \mu_{\min})/T^G$.

The proposed mechanism has a *multiplicative* form: each answer in the gold standard is given a score based on whether it was correct (score = $\frac{1}{T}$), incorrect (score = 0) or skipped (score = 1), and the final payment is simply a product of these scores (scaled and shifted by constants). The mechanism is easy to describe to workers: For instance, if $T = \frac{1}{2}$, $G = 3$, $\mu_{\max} = 80$ cents and $\mu_{\min} = 0$ cents, then the description reads:

“The reward starts at 10 cents. For every correct answer in the 3 gold standard questions, the reward will double. However, if any of these questions are answered incorrectly, then the reward will become zero. So please use the ‘I’m not sure’ option wisely.”

Observe how this payment rule is similar to the popular ‘double or nothing’ paradigm (Double or Nothing, 2014).

The algorithm makes a minimum payment if *one or more* attempted answers in the gold standard are wrong. Note that this property is significantly stronger than the no-free-lunch axiom which we originally required, where we wanted a minimum payment only when *all* attempted answers were wrong. Surprisingly, as we prove shortly, Algorithm 1 is the only incentive-compatible mechanism that satisfies no-free-lunch.

The following theorem shows that this mechanism indeed incentivizes a worker to skip the questions for which her confidence is below T , while answering those for which her confidence is greater than T . In the latter case, the worker is incentivized to select the answer which she thinks is most likely to be correct.

Theorem 2 *The mechanism of Algorithm 1 is incentive-compatible and satisfies the no-free-lunch axiom.*

In the remainder of this subsection, we present the proof of Theorem 2. The reader may go directly to subsection 3.3 without loss in continuity.

Proof of Theorem 2. The proposed payment mechanism satisfies no-free-lunch since the payment is μ_{\min} when there are one or more wrong answers in the gold standard. It remains to show that the mechanism is incentive compatible. To this end, observe that the property of incentive-compatibility does not change upon any shift of the mechanism by a constant value or any scaling by a positive constant value. As a result, for the purposes of this proof, we can assume without loss of generality that $\mu_{\min} = 0$.

We will first assume that, for every question that the worker does not skip, she selects the answer which she believes is most likely to be correct. Under this assumption we will show that the worker is incentivized to skip the questions for which her confidence is smaller than T and attempt if it is greater than T . Finally, we will show that the mechanism indeed incentivizes the worker to select the answer which she believes is most likely to be correct for the questions that she doesn't skip. In what follows, we will employ the notation $\kappa = \mu_{\max} T^G$.

Let us first consider the case when $G = N$. Let p_1, \dots, p_N be the confidences of the worker for questions $1, \dots, N$ respectively. Further, let $p_{(1)} \geq \dots \geq p_{(m)} > T > p_{(m+1)} \geq \dots \geq p_{(N)}$ be the ordered permutation of these confidences (for some number m). Let $\{(1), \dots, (N)\}$ denote the corresponding permutation of the N questions. If the mechanism is incentive compatible, then the expected payment received by this worker should be maximized when the worker answers questions $(1), \dots, (m)$ and skips the rest. Under the mechanism proposed in Algorithm 1, this action fetches the worker an expected payment of

$$\frac{p_{(1)}}{\kappa} \dots \frac{p_{(m)}}{T}.$$

Alternatively, if the worker answers the questions $\{i_1, \dots, i_\beta\}$, with $p_{i_1} > \dots > p_{i_\nu} > T > p_{i_{\nu+1}} > \dots > p_{i_\beta}$ for some value ν , then the expected payment is

$$p_{i_1} \dots p_{i_\beta} T^\beta = \kappa \frac{p_{i_1}}{T} \dots \frac{p_{i_\beta}}{T} \tag{2}$$

$$\leq \kappa \frac{p_{i_1}}{T} \dots \frac{p_{i_\nu}}{T} \tag{3}$$

$$\leq \kappa \frac{p_{(1)}}{T} \dots \frac{p_{(m)}}{T}, \tag{4}$$

where inequality (3) holds because $\frac{p_{i_j}}{T} \leq 1 \forall j > \nu$ and holds with equality only when $\beta = \nu$. Inequality (4) is a result of $\frac{p_{i_j}}{T} \geq 1 \forall j \leq m$ and holds with equality only when $\nu = m$. It follows that the expected payment is (strictly) maximized when $i_1 = (1), \dots, i_\beta = (m)$ as required.

The case of $G < N$ is a direct consequence of the result for $G = N$, as follows. When $G < N$, from a worker's point of view, the set of G questions is distributed uniformly at random in

the superset of N questions. However, for every set of G questions, the relations (2), (3), (4) and their associated equality/strict-inequality conditions hold. The expected payment is thus (strictly) maximized when the worker answers the questions for which her confidence is greater than T and skips those for which her confidence is smaller than T .

One can see that for every question that the worker chooses to answer, the expected payment increases with an increase in her confidence. Thus, the worker is incentivized to select the answer that she thinks is most probably correct.

Finally, since $\kappa = \mu_{\max} T^C > 0$ and $T \in (0, 1)$, the payment is always non-negative and satisfies the μ_{\max} -budget constraint.

3.3 Uniqueness of this Mechanism

While we started out with a very weak condition of no-free-lunch that requires a minimum payment when *all* attempted answers are wrong, the mechanism proposed in Algorithm 1 is significantly more strict and pays the minimum amount when *any* of the attempted answers is wrong. A natural question that arises is: can we design an alternative mechanism satisfying incentive compatibility and no-free-lunch that operates somewhere in between? The following theorem answers this question in the negative.

Theorem 3 *The mechanism of Algorithm 1 is the only incentive-compatible mechanism that satisfies the no-free-lunch axiom.*

Theorem 3 gives a strong result despite imposing very weak requirements. To see this, recall our earlier discussion on deterring spammers, that is, making a low payment to workers who answer randomly. For instance, when the task comprises binary-choice questions, one may wish to design mechanisms which make the minimum possible payment when the responses to 50% or more of the questions in the gold standard are incorrect. The no-free-lunch axiom is a much weaker requirement, and the only mechanism that can satisfy this requirement is the mechanism of Algorithm 1.

The proof of Theorem 3 is based on the following key lemma, establishing a condition that any incentive-compatible mechanism must necessarily satisfy. Note that this lemma does *not* require the no-free-lunch axiom.

Lemma 4 *Any incentive-compatible mechanism f must satisfy, for every gold standard question $i \in \{1, \dots, G\}$ and every $(y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_G) \in \{-1, 0, 1\}^{G-1}$,*

$$\begin{aligned} T f(y_1, \dots, y_{i-1}, 1, y_{i+1}, \dots, y_G) + (1 - T) f(y_1, \dots, y_{i-1}, -1, y_{i+1}, \dots, y_G) \\ = f(y_1, \dots, y_{i-1}, 0, y_{i+1}, \dots, y_G). \end{aligned}$$

The proof of Lemma 4 is provided in Appendix A.1. Using this lemma, we will now prove Theorem 3. The reader interested in further results and not the proof may feel free to jump to Subsection 3.4 without any loss in continuity.

Proof of Theorem 3. The property of incentive-compatibility does not change upon any shift of the mechanism by a constant value or any scaling by a positive constant value. As a result, for the purposes of this proof, we can assume without loss of generality that $\mu_{\min} = 0$.

We will first prove that any incentive-compatible mechanism satisfying the no-free-lunch axiom must make a zero payment if one or more answers in the gold standard are incorrect. The proof proceeds by induction on the number of skipped questions S in the gold standard. Let us assume for now that in the G questions in the gold standard, the first question is answered incorrectly, the next $(G - 1 - S)$ questions are answered by the worker and have arbitrary evaluations, and the remaining S questions are skipped. The proof proceeds by an induction on S . Suppose $S = G - 1$. In this case, the only attempted question is the first question and the answer provided by the worker to this question is incorrect. The no-free-lunch axiom necessitates a zero payment in this case, thus satisfying the base case of our induction hypothesis. Now we prove the hypothesis for some S under the assumption of it being true when the number of questions skipped in the gold standard is $(S + 1)$ or more. From Lemma 4 (with $i = G - S - 1$) we have

$$\begin{aligned} T f(-1, y_2, \dots, y_{G-S-2}, 1, 0, \dots, 0) + (1 - T) f(-1, y_2, \dots, y_{G-S-2}, -1, 0, \dots, 0) \\ = f(-1, y_2, \dots, y_{G-S-2}, 0, 0, \dots, 0) \\ = 0, \end{aligned}$$

where the final equation is a consequence of our induction hypothesis: The induction hypothesis is applicable since $f(-1, y_2, \dots, y_{G-S-2}, 0, 0, \dots, 0)$ corresponds to the case when the last $(S + 1)$ questions are skipped and the first question is answered incorrectly. Now, since the payment f must be non-negative and since $T \in (0, 1)$, it must be that

$$f(-1, y_2, \dots, y_{G-S-2}, 1, 0, \dots, 0) = 0,$$

and

$$f(-1, y_2, \dots, y_{G-S-2}, -1, 0, \dots, 0) = 0.$$

This completes the proof of our induction hypothesis. Furthermore, each of the arguments above hold for any permutation of the G questions, thus proving the necessity of zero payment when any one or more answers are incorrect.

We will now prove that when no answers in the gold standard are incorrect, the payment must be of the form described in Algorithm 1. Let κ be the payment when all G questions in the gold standard are skipped. Let C be the number of questions answered correctly in the gold standard. Since there are no incorrect answers, it follows that the remaining $(G - C)$ questions are skipped. Let us assume for now that the first C questions are answered correctly and the remaining $(G - C)$ questions are skipped. We repeatedly apply Lemma 4, and the fact that the payment must be zero when one or more answers are wrong,

$$\begin{aligned} f(\underbrace{1, \dots, 1}_{C-1}, \underbrace{1, 0, \dots, 0}_{G-C}) &= \frac{1}{T} f(\underbrace{1, \dots, 1}_{C-1}, \underbrace{0, 0, \dots, 0}_{G-C}) - \frac{1 - T}{T} f(\underbrace{1, \dots, 1}_{C-1}, \underbrace{-1, 0, \dots, 0}_{G-C}) \\ &= \frac{1}{T} f(\underbrace{1, \dots, 1}_{C-1}, \underbrace{0, 0, \dots, 0}_{G-C}), \end{aligned}$$

and so on to obtain

$$\begin{aligned} f(\underbrace{1, \dots, 1}_{C-1}, \underbrace{1, 0, \dots, 0}_{G-C}) &= \frac{1}{T^C} f(\underbrace{0, \dots, 0}_C) \\ &= \frac{1}{T^C} \kappa. \end{aligned}$$

In order to abide by the budget, we must have the maximum payment as $\mu_{\max} = \kappa \frac{1}{T^G}$. It follows that $\kappa = \mu_{\max} T^G$. Finally, the arguments above hold for any permutation of the G questions, thus proving the uniqueness of the mechanism of Algorithm 1.

3.4 Optimality against Spamming Behavior

As discussed earlier, crowdsourcing tasks, especially those with multiple choice questions, often encounter spammers who answer randomly without heed to the question being asked. For instance, under a binary-choice setup, a spammer will choose one of the two options uniformly at random for every question. A highly desirable objective in crowdsourcing settings is to deter spammers. To this end, one may wish to impose a condition of making the minimum possible payment when the responses to 50% or more of the attempted questions in the gold standard are incorrect. A second desirable metric could be to minimize the expenditure on a worker who simply skips all questions. While the aforementioned requirements were deterministic functions of the worker's responses, one may alternatively wish to impose requirements that depend on the distribution of the worker's answering process. For instance, a third desirable feature would be to minimize the expected payment to a worker who answers all questions uniformly at random. We now show that interestingly, our unique multiplicative payment mechanism *simultaneously* satisfies all these requirements. The result is stated assuming a multiple-choice setup, but extends trivially to non-multiple-choice settings.

Theorem 5.A (Distributional) *Consider any value $A \in \{0, \dots, G\}$. Among all incentive-compatible mechanisms (that may or may not satisfy no-free-lunch), Algorithm 1 pays strictly the smallest amount to a worker who skips some A of the questions in the gold standard, and chooses answers to the remaining $(G - A)$ questions uniformly at random.*

Theorem 5.B (Deterministic) *Consider any value $B \in (0, 1]$. Among all incentive-compatible mechanisms (that may or may not satisfy no-free-lunch), Algorithm 1 pays strictly the smallest amount to a worker who gives incorrect answers to a fraction B or more of the questions attempted in the gold standard.*

We see from this result that the multiplicative payment mechanism of Algorithm 1 thus possesses very useful properties geared to deter spammers, while ensuring that a good worker will be paid a high enough amount. To illustrate this point, let us compare the mechanism of Algorithm 1 with the popular additive class of payment mechanisms.

Example 1 *Consider the popular class of “additive” mechanisms, where the payments to a worker are added across the gold standard questions. This additive payment mechanism offers a reward of $\frac{\mu_{\max}}{G}$ for every correct answer in the gold standard, $\frac{\mu_{\max} T}{G}$ for every question skipped, and 0 for every incorrect answer. Importantly, the final payment to the worker is the sum of the rewards across the G gold standard questions. One can verify that this additive mechanism is incentive compatible. One can also see that that as guaranteed by our theory, this additive payment mechanism does not satisfy the no-free-lunch axiom.*

Suppose each question involves choosing from two options. Let us compute the payment that these two mechanisms make under a spamming behavior of choosing the answer randomly to each question. Given the 50% likelihood of each question being correct, one can compute that the additive

mechanism makes a payment of $\frac{\mu_{\max}}{2}$ in expectation. On the other hand, our mechanism pays an expected amount of only $\mu_{\max} 2^{-G}$. The payment to spammers thus reduces exponentially with the number of gold standard questions under our mechanism, whereas it does not reduce at all in the additive mechanism.

Now, consider a different means of exploiting the mechanism(s) where the worker simply skips all questions. To this end, observe that if a worker skips all the questions then the additive payment mechanism will make a payment of $\mu_{\max} T$. On the other hand, the proposed payment mechanism of Algorithm 1 pays an exponentially smaller amount of $\mu_{\max} T^G$ (recall that $T < 1$).

We prove Theorem 5 in the rest of this subsection. The reader may feel free to jump directly to Section 4 without any loss in continuity.

Proof of Theorem 5. The property of incentive-compatibility does not change upon any shift of the mechanism by a constant value or any scaling by a positive constant value. As a result, for the purposes of this proof, we can assume without loss of generality that $\mu_{\min} = 0$.

Part A (Distributional). Let m denote the number of options in each question. One can verify that under the mechanism of Algorithm 1, a worker who skips A questions and answers the rest uniformly at random will get a payment of $\frac{\mu_{\max} T^A}{m^{G-A}}$ in expectation. This expression arises due to the fact that Algorithm 1 makes a zero payment if any of the attempted answers are incorrect, and a payment of $\mu_{\max} T^A$ if the worker skips A questions and answers the rest correctly. Under uniformly random answers, the probability of the latter event is $\frac{1}{m^{G-A}}$.

Now consider any other mechanism, and denote it as f' . Let us suppose without loss of generality that the worker attempts the first $(G - A)$ questions. Since the payment must be non-negative, a repeated application of Lemma 4 gives

$$f'(1, \dots, 1, 0, \dots, 0) \geq T f'(1, \dots, 1, 0, \dots, 0) \quad (5)$$

$$\begin{aligned} & \vdots \\ & \geq T^A f'(1, \dots, 1) \\ & = T^A \mu_{\max}, \end{aligned} \quad (6)$$

where (6) is a result of the μ_{\max} -budget constraint. Since there is a $\frac{1}{m^{G-A}}$ chance of the $(G - A)$ attempted answers being correct, the expected payment under any other mechanism f' must be at least $\frac{\mu_{\max} T^A}{m^{G-A}}$.

We will now show that if any mechanism f' that makes an expected payment of $\frac{\mu_{\max} T^A}{m^{G-A}}$ to such a spammer, then the mechanism must be identical to Algorithm 1. We split the proof of this part into two cases, depending on the value of the parameter A .

Case I ($A < G$): In order to make an expected payment of $\frac{\mu_{\max} T^A}{m^{G-A}}$, the mechanism must achieve the bound (6) with equality, and furthermore, the mechanism must have zero payment if any of the $(G - A)$ attempted questions are answered incorrectly. In other words, the mechanism f' under consideration must satisfy

$$f'(y_1, \dots, y_{G-A}, 0, \dots, 0) = 0 \quad \forall (y_1, \dots, y_{G-A}) \in \{-1, 1\}^{G-A}.$$

A repeated application of Lemma 4 then implies

$$f'(0, 0, \dots, 0, \dots, -1) = 0. \quad (7)$$

Note that so far we considered the case when the worker attempts the first $(G - A)$ questions. The arguments above hold for any choice of the $(G - A)$ attempted questions, and consequently the results shown so far in this proof hold for all permutations of the arguments to f' . In particular, the mechanism f' must make a zero payment when any $(G - 1)$ questions in the gold standard are skipped and the remaining question is answered incorrectly. Another repeated application of Lemma 4 to this result gives

$$f'(y_1, \dots, y_G) = 0 \quad \forall (y_1, \dots, y_G) \in \{0, -1\}^G \setminus \{0\}^G.$$

This condition is precisely the no-free-lunch axiom, and in Theorem 3 we had shown that Algorithm 1 is the only incentive-compatible mechanism that satisfies this axiom. It follows that our mechanism, Algorithm 1 strictly minimizes the expected payment in the setting under consideration.

Case II ($A = G$): In order to achieve the bound (6) with equality, the mechanism f' must also achieve the bound (5) with equality. Noting that we have $A = G$ in this case, it follows that the mechanism f' must satisfy

$$f'(-1, 0, \dots, 0) = 0.$$

This condition is identical to (7) established for Case I earlier, and the rest of the argument now proceeds in a manner identical to the subsequent arguments in Case I.

Part B (Deterministic). Given our result of Theorem 3, the proof for the deterministic part is straightforward. Algorithm 1 makes a payment of zero when one or more of the answers to questions in the gold standard are incorrect. Consequently, for every value of parameter $B \in (0, 1]$, Algorithm 1 makes a zero payment when a fraction B or more of the attempted answers are incorrect. Any other mechanism doing so must satisfy the no-free-lunch axiom. In Theorem 3 we had shown that Algorithm 1 is the only incentive-compatible mechanism that satisfies this axiom. It follows that our mechanism, Algorithm 1, strictly minimizes the payment in the event under consideration.

4. Confidence-based Setting

In this section, we will discuss incentive mechanisms when the worker is asked to select from more than one confidence-level for every question (Figure 1c). In particular, for some $L \geq 1$, the worker is asked to indicate a confidence-level in the range $\{0, \dots, L\}$ for every answer. Level 0 is the ‘skip’ level, and level L denotes the highest confidence. Note that we do not solicit an answer if the worker indicates a confidence-level of 0 (skip), but the worker must provide an answer if she indicates a confidence-level of 1 or higher. This makes the case of having only a ‘skip’ as considered in Section 3 a special case of this setting, and corresponds to $L = 1$.

We generalize the requirement of no-free-lunch to the confidence-based setting as follows.

Axiom 6 (Generalized-no-free-lunch axiom) *If all the answers attempted by the worker in the gold standard are selected as the highest confidence-level (level L), and all of them turn out to be wrong, then the payment is μ_{\min} . More formally, we require the mechanism f to satisfy $f(x_1, \dots, x_G) = \mu_{\min}$ for every evaluation (x_1, \dots, x_G) that satisfies $0 < \sum_{i=1}^G \mathbf{1}\{x_i \neq 0\} = \sum_{i=1}^G \mathbf{1}\{x_i = -L\}$.*

In the confidence-based setting, we require specification of a set of thresholds $\{S_l, T_l\}_{l=1}^L$ that determine the confidence-levels that the workers should indicate. These thresholds are used to choose the payment mechanism in a principled manner. In particular, we will require specification of two reference points for each confidence level, and this specification generalizes the skip-based setting.

- The first set of thresholds specifies a comparison of any confidence level with the skipping option as a fixed reference. To this end, recall that in the skip-based setting, the threshold T specified when the worker should skip a question and when she should attempt to answer. This is generalized to the confidence-based setting where for every level $l \in [L]$, a fixed threshold S_l specifies the ‘strength’ of confidence-level l . If restricted to only the two options of skipping or selecting confidence-level l for any question, the worker should be incentivized to select confidence-level l if her confidence is higher than S_l and skip if her confidence is lower than S_l .

- The second set of thresholds specifies a comparison of any confidence level with its neighbors. If a worker decides to not skip a question, she must choose one of multiple confidence-levels. A set $\{T_l\}_{l=1}^L$ of thresholds specify the boundaries between different confidence-levels. In particular, when the confidence of the worker for a question lies in (T_{l-1}, T_{l+1}) , then the worker must be incentivized to indicate confidence-level $(l - 1)$ if her confidence is lower than T_l and to indicate confidence-level l if her confidence is higher than T_l . This includes selecting level L if her confidence is higher than T_L and selecting level 0 if her confidence is lower than T_1 .

We will call a payment mechanism as incentive-compatible if it satisfies the two requirements listed above, and also incentivizes the worker to select the answer that she believes is most likely to be correct for every question for which her confidence is higher than T_1 .

The problem setting inherently necessitates certain restrictions in the choice of the thresholds. Since we require the worker to choose a higher level when her confidence is higher, the thresholds must necessarily be monotonic and satisfy $0 < S_1 < S_2 < \dots < S_L < 1$ and $0 < T_1 < T_2 < \dots < T_L < 1$. Also observe that the definitions of S_l and T_l coincide, and hence $S_1 = T_1$. Additionally, we can show (Proposition 18 in Appendix A.5) that for incentive-compatible mechanisms to exist, it must be that $T_l > S_l \forall l \in \{2, \dots, L\}$. As a result, the thresholds must also satisfy $T_1 = S_1, T_2 > S_2, \dots, T_L > S_L$. These thresholds may be chosen based on various factors of the problem at hand, for example, on the post-processing algorithms, any statistics on the distribution of worker abilities, budget constraints, etc. In this paper, we will assume that these values are given to us.

4.1 Payment Mechanism

In this section, we present our proposed payment mechanism, and prove that it is guaranteed to satisfy our requirements. We begin with a description of the mechanism in Algorithm 2.

Algorithm 2: Incentive mechanism for the confidence-based setting

- Inputs:
 - ▶ Thresholds S_1, \dots, S_L and T_1, \dots, T_L
 - ▶ Budget parameters μ_{\max} and μ_{\min}
 - ▶ Evaluations $(x_1, \dots, x_G) \in \{-L, \dots, +L\}^G$ of the worker's answers to the G gold standard questions
- Set $\alpha_{-L}, \dots, \alpha_L$ as
 - ▶ $\alpha_L = \frac{1}{S_L}, \alpha_{-L} = 0$
 - ▶ For $l \in \{L-1, \dots, 1\}$,

$$\alpha_l = \frac{(1-S_l)T_{l+1}\alpha_{l+1} + (1-S_l)(1-T_{l+1})\alpha_{-(l+1)} - (1-T_{l+1})}{T_{l+1} - S_l} \quad \text{and} \quad \alpha_{-l} = \frac{1 - S_l \alpha_l}{1 - S_l}$$

- ▶ $\alpha_0 = 1$

- The payment is

$$f(x_1, \dots, x_G) = \kappa \prod_{i=1}^G \alpha_{x_i} + \mu_{\min}$$

where $\kappa = (\mu_{\max} - \mu_{\min}) \left(\frac{1}{\alpha_L}\right)^G$.

The following theorem shows that this mechanism indeed incentivizes a worker to select answers and confidence-levels as desired.

Theorem 7 *The mechanism of Algorithm 2 is incentive-compatible and satisfies the generalized-no-free-lunch axiom.*

The proof of Theorem 7 follows in a manner similar to that of the proof of Theorem 2, and is provided in Appendix A.2.

Remark 8 *The mechanism of Algorithm 2 also ensures a condition stronger than the 'boundary-based' definition of the thresholds $\{T_l\}_{l \in [L]}$ given earlier. Under this mechanism, for every $l \in [L-1]$ the worker is incentivized to select confidence-level 1 (over all else) whenever her confidence lies in the interval (T_l, T_{l+1}) , select confidence-level 0 (over all else) whenever her confidence is lower than T_l and select confidence-level L (over all else) whenever her confidence is higher than T_L .*

4.2 Uniqueness of this Mechanism

We prove that the mechanism of Algorithm 2 is unique, that is, no other incentive-compatible mechanism can satisfy the generalized-no-free-lunch axiom.

Theorem 9 *The payment mechanism of Algorithm 2 is the only incentive-compatible mechanism that satisfies the generalized-no-free-lunch axiom.*

The proof of Theorem 9 is provided in Appendix A.3. The proof is conceptually similar to that of Theorem 9 but involves resolving several additional complexities that arise due to elicitation from multiple confidence levels.

5. A Stronger No-free-lunch Axiom: Impossibility Results

Recall that the no-free-lunch axiom under the skip-based mechanism of Section 3 requires the payment to be the minimum possible if all attempted answers in the gold standard are incorrect. However, a worker who skips all the questions may still receive a payment. The generalization under the confidence-based mechanism of Section 4 requires the payment to be the minimum possible if all attempted answers in the gold standard were selected with the highest confidence-level and were incorrect. However, a worker who marked all questions with a lower confidence level may be paid even if her answers to all the questions in the gold standard turn out to be incorrect. One may thus wish to impose a stronger requirement instead, where the minimum payment is made to workers who make no useful contribution. This is the primary focus of this section.

Consider the skip-based setting. Define the following axiom which is slightly stronger than the no-free-lunch axiom defined previously.

Strong-no-free-lunch: If none of the answers in the gold standard are correct, then the payment is μ_{\min} . More formally, strong-no-free-lunch imposes the condition $f(x_1, \dots, x_G) = \mu_{\min}$ for every evaluation (x_1, \dots, x_G) that satisfies $\sum_{i=1}^G \mathbf{1}\{x_i > 0\} = 0$.

The strong-no-free-lunch axiom is only slightly stronger than the no-free-lunch axiom proposed in Section 3 for the skip-based setting. The strong-no-free-lunch axiom can equivalently be written as imposing requiring the payment to be the minimum possible for every evaluation that satisfies $\sum_{i=1}^G \mathbf{1}\{x_i \neq 0\} = \sum_{i=1}^G \mathbf{1}\{x_i = -1\}$. From this interpretation, one can see that to the set of events necessitating the minimum payment under the no-free-lunch axiom, the strong-no-free-lunch axiom adds only one extra event, the event of the worker skipping all questions. Unfortunately, it turns out that this minimal strengthening of the requirements is associated to impossibility results.

In this section we show that no mechanism satisfying the strong-no-free-lunch axiom can be incentive compatible in general. The only exception is the case when (a) all questions are in the gold standard ($G = N$), and (b) it is guaranteed that the worker has a confidence greater than T for at least one of the N questions. These conditions are, however, impractical for the crowdsourcing setup under consideration in this paper. We will first prove the impossibility results under the strong-no-free-lunch axiom. For the sake of completeness (and also to satisfy mathematical curiosity), we will then provide a (unique) mechanism that is incentive-compatible and satisfies the strong-no-free-lunch axiom for the skip-based setting under the two conditions listed above. The proofs of each of the claims made in this section are provided in Appendix A.6.

Let us continue to discuss the skip-based setting. In this section, we will call any worker whose confidences for all of the N questions is lower than T as an *unknowledgeable worker*, and call the worker a *knowledgeable worker* otherwise.

Proposition 10 *No payment mechanism satisfying the strong-no-free-lunch axiom can incentivize an unknowledgeable worker to skip all questions. As a result, no mechanism satisfying the strong-no-free-lunch axiom can be incentive-compatible.*

The proof of this proposition, and that of all other theoretical claims made in this section, are presented in Appendix A.6.

The impossibility result of Proposition 10 relies on trying to incentivize an unknowledgeable worker to act as desired. Since no mechanism can be incentive compatible for unknowledgeable workers, we will now consider only workers who are knowledgeable. The following proposition shows that the strong-no-free-lunch axiom is too strong even for this relaxed setting.

Proposition 11 *When $G < N$, there exists no mechanism that is incentive-compatible for knowledgeable workers and satisfies the strong-no-free-lunch axiom.*

Given this impossibility result for $G < N$, we are left with $G = N$ which means that the true answers to all the questions are known a priori. This condition is clearly not applicable to a crowdsourcing setup; nevertheless, it is mathematically interesting and may be applicable to other scenarios such as testing and elicitation of beliefs about future events.

Proposition 12 below presents a mechanism for this case and proves its uniqueness. We previously saw that an unknowledgeable worker cannot be incentivized to skip all the questions (even when $G = N$). Thus, in our payment mechanism, we do the next best thing: Incentivize the unknowledgeable worker to answer only one question, that which she is most confident about, while incentivizing the knowledgeable worker to answer questions for which her confidence is greater than T and skip those for which her confidence is smaller than T .

Proposition 12 *Let C be the number of correct answers and W be the number of wrong answers (in the gold standard). Let the payment be μ_{\min} if $W > 0$ or $C = 0$, and be $(\mu_{\max} - \mu_{\min})T^{G-C} + \mu_{\min}$ otherwise. Under this mechanism, when $G = N$, an unknowledgeable worker is incentivized to answer only one question, that for which her confidence is the maximum, and a knowledgeable worker is incentivized to answer the questions for which her confidence is greater than T and skip those for which her confidence is smaller than T . Furthermore, when $G = N$, this mechanism is the one and only mechanism that obeys the strong-no-free-lunch axiom and is incentive-compatible for knowledgeable workers.*

The following proposition shows that the strong-no-free-lunch axiom leads to negative results in the confidence-based setting ($L > 1$) as well. The strong-no-free-lunch axiom is still defined as in the beginning of Section 5, i.e., the payment is zero if none of the answers are correct.

Proposition 13 *When $L > 1$, for any values of N and G ($\leq N$), it is impossible for any mechanism to satisfy the strong-no-free-lunch axiom and be incentive-compatible even when the worker is knowledgeable.*

6. Simulations and Experiments

In this section, we present synthetic simulations and real-world experiments to evaluate the effects of our setting and our mechanism on the final label quality.

6.1 Synthetic Simulations

We employ synthetic simulations to understand the effects of various distributions of the confidences and labeling errors. We consider binary-choice questions in this set of simulations. Whenever a worker answers a question, her confidence for the correct answer is drawn from a distribution \mathcal{P} independent of all else. We investigate the effects of the following five choices of the distribution \mathcal{P} :

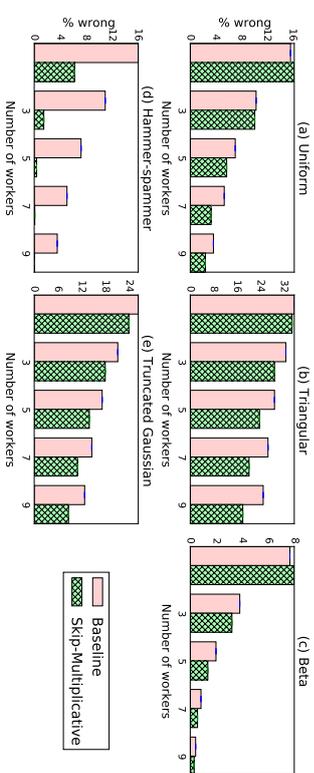


Figure 2: Error under different interfaces for synthetic simulations of five distributions of the workers' error probabilities.

- The uniform distribution on the support $[0.5, 1]$.
- A triangular distribution with lower end-point 0.2, upper end-point 1 and a mode of 0.6.
- A beta distribution with parameter values $\alpha = 5$ and $\beta = 1$.
- The hammer-spammer distribution (Karger et al., 2011): uniform on the discrete set $\{0.5, 1\}$.
- A truncated Gaussian distribution: a truncation of $\mathcal{N}(0.75, 0.5)$ to the interval $[0, 1]$.

We compare (a) the setting where workers attempt every question, with (b) the setting where workers skip questions for which their confidence is below a certain threshold T . In this set of simulations, we set $T = 0.75$. In either setting, we aggregate the labels obtained from the workers for each question via a majority vote on the two classes. Ties are broken by choosing one of the two options uniformly at random.

Figure 2 depicts the results from these simulations. Each bar represents the fraction of questions that are labeled incorrectly, and is an average across 50,000 trials. (The standard error of the mean is too small to be visible.) We see that the skip-based setting consistently outperforms the conventional setting, and the gains obtained are moderate to high depending on the underlying distribution of the workers' errors. In particular, the gains are quite striking under the hammer-spammer model: this result is not surprising since the mechanism (ideally) screens the spammers out and leaves only the hammers who answer perfectly.

The setup of the simulations described above assumes that the workers' confidences equal the true error probabilities. In practice, however, the workers may have incorrect beliefs. The setup also assumes that ties are broken randomly; however in practice, ties may be broken in a more systematic manner by eliciting additional labels for only these hard questions. We now present a second set of simulations that mitigates these biases. In particular, when a worker has a confidence of p , the actual probability of error is assumed to be drawn from a Gaussian distribution with mean p and standard deviation 0.1, truncated to $[0, 1]$. In addition, when evaluating the performance of the

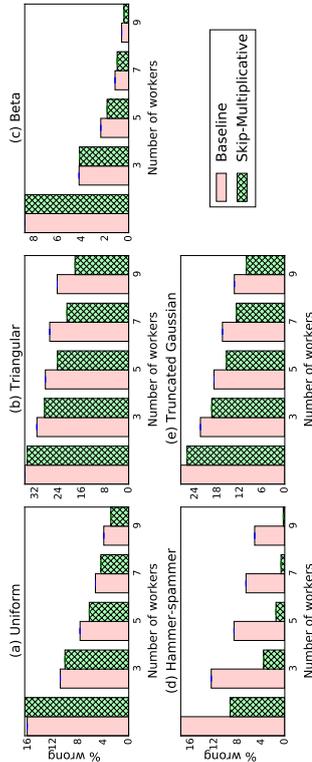


Figure 3: Errors under a model that is a perturbation of the first experiment, where the worker’s confidence is a noisy version of the true error probability and where ties are considered different from random decisions.

majority voting procedure, we consider a tie as having an error of 0.4. Figure 3 depicts the results of these simulations. We observe that the results from these simulations are very similar to those obtained in the earlier simulation setup of Figure 2.

6.2 Experiments on Amazon Mechanical Turk

We conducted preliminary experiments on the Amazon Mechanical Turk commercial crowdsourcing platform (`mturk.com`) to evaluate our proposed scheme in real-world scenarios. The complete data, including the interface presented to the workers in each of the tasks, the results obtained from the workers, and the ground truth solutions, are available on the website of the first author.

6.2.1 GOAL

Before delving into details, we first note certain caveats relating to such a study of mechanism design on crowdsourcing platforms. When a worker encounters a mechanism for only a small amount of time (a handful of tasks in typical research experiments) and for a small amount of money (at most a few dollars in typical crowdsourcing tasks), we cannot expect the worker to completely understand the mechanism and act precisely as required. For instance, we wouldn’t expect our experimental results to change significantly even upon moderate modifications in the promised amounts, and furthermore, we do expect the outcomes to be noisy. Incentive compatibility kicks in when the worker encounters a mechanism across a longer term, for example, when a proposed mechanism is adopted as a standard for a platform, or when higher amounts are involved. This is when we would expect workers or others (e.g., bloggers or researchers) to design strategies that can game the mechanism. The theoretical guarantee of incentive compatibility then prevents such gaming in the long run.

We thus regard these experiments as preliminary. Our intentions towards this experimental exercise were (a) to evaluate the potential of our algorithms to work in practice, (b) to investigate

the effect of the proposed algorithms on the net error in the collected labeled data, and (c) to identify if there is any major issue of dissatisfaction among the workers.

6.2.2 EXPERIMENTAL SETUP

We conducted our experiments on the “Amazon Mechanical Turk” commercial crowdsourcing platform (`mturk.com`). On this platform, individuals or businesses (called ‘requesters’) can post tasks, and any individual (called a ‘worker’) may complete the task over the Internet in exchange for a pre-specified payment. The payment may comprise of two parts: a fixed component which is identical for all workers performing that task, and a ‘bonus’ which may be different for different workers and is paid at the discretion of the requester.

We designed nine experiments (tasks) ranging from image annotation to text and speech recognition. The individual experiments are described in more detail in Appendix B. All experiments involved objective questions, and the responses elicited were multiple choice in five of the experiments and free-form text in the rest. For each experiment, we tested three settings: (i) the baseline conventional setting (Figure 1a) with a mechanism of paying a fixed amount per correct answer, (ii) our skip-based setting (Figure 1b) with our multiplicative mechanism, and (iii) our confidence-based setting (Figure 1c) with our confidence-based mechanism. For each mechanism in each experiment, we specified the requirement of 35 workers independently performing the task. This amounts to a total of 945 worker-tasks (315 worker-tasks for each mechanism). We also set the following constraints for a worker to attempt our tasks: the worker must have completed at least 100 tasks previously, and must have a history of having at least 95% of her prior work approved by the respective requesters. In each experiment, we offered a certain small fixed payment (in order to attract the workers in the first place) and executed the variable part of our mechanisms via a bonus payment.

6.2.3 RESULTS: RAW DATA

Figure 4 plots, for the baseline, skip-based and confidence-based mechanisms for all nine experiments, the (i) fraction of questions that were answered incorrectly, (ii) fraction of questions that were answered incorrectly among those that were attempted, (iii) the average payment to a worker (in cents), and (iv) break up of the answers in terms of the fraction of answers in each confidence level. The payment for various tasks plotted in Figure 4 is computed as the average of the payments across 100 (random) selections of the gold standard questions, in order to prevent any distortion of the results due to the randomness in the choice of the gold standard questions.

The figure shows that the amount of errors among the attempted questions is much lower in the skip and the confidence-based settings than the baseline setting. Also observe that in the confidence-based setting, as expected, the answers selected with higher confidence-levels are more correct. The total money spent under each of these settings is similar, with the skip and the confidence-based settings faring better in most cases. We also elicited feedback from the workers, in which we received several positive comments (and no negative comments). Examples of comments that we received: “I was wondering if it would be possible to increase the maximum number of HTTs I may complete for you. As I said before, they were fun to complete. I think I did a good job completing them, and it would be great to complete some more for you.”; “I am eagerly waiting for your bonus.”; “Enjoyable. Thanks.”

6.2.4. RESULTS: AGGREGATED DATA

We saw in the previous section that under the skip-based setting, the amount of error among the attempted questions was significantly lower than the amount of error in the baseline setting. However, the skip-based setting was also associated, by design, to lesser amount of data by virtue of questions being skipped by the workers. A natural question that arises is how the baseline and the skip-based mechanisms will compare in terms of the final data quality, i.e., the amount of error once data from multiple workers is aggregated.

To this end, we considered the five experiments that consisted of multiple-choice questions. We let a parameter `num_workers` take values in {3, 5, 7, 9, 11}. For each of the five experiments and for each of the five values of `num_workers`, we perform the following actions 1,000 times: for each question, we choose `num_workers` workers and perform a majority vote on their responses. If the correct answer for that question does not lie in the set of options given by the majority, we consider it as an accuracy of zero. Otherwise, if there are m options tied in the majority vote, and the correct answer is one of these m , then we consider it as an accuracy of $\frac{100\%}{m}$ (hence, 100% if the correct answer is the only answer picked by the majority vote). We average the accuracy across all questions and across all iterations.

We choose majority voting as the means of aggregation since (a) it is the simplest and still most popular aggregation method, and (b) to enable an apples-to-apples comparison design since while more advanced aggregation algorithms have been developed for the baseline setting without the skip (Raykar et al., 2010; Zhou et al., 2012; Wauthier and Jordan, 2011; Chen et al., 2013; Khetani and Oh, 2016; Dawid and Skene, 1979; Karger et al., 2011; Liu et al., 2012; Vempaty et al., 2014; Zhang et al., 2014; Ipeirotis et al., 2014; Zhou et al., 2015; Shah et al., 2016c), but design of analogous algorithms for the new skip-based setting is still open.

The results are presented in Figure 5. We see that in most cases, our skip-based mechanism induces a lower labeling error at the aggregate level than the baseline. Furthermore, in many of the instances, the reduction is two-fold or higher.

All in all, in the experiments, we observe a substantial reduction in the error-rates while expending the same or lower amounts and receiving no negative comments from the workers, suggesting that these mechanisms can work; the fundamental theory underlying the mechanisms ensures that the system cannot be gamed in the long run. Our proposed settings and mechanisms thus have the potential to provide much higher quality labeled data as input to machine learning algorithms.

7. Discussion and Conclusions

In this concluding section, we first discuss the modeling assumptions that we made in this paper, followed by a discussion on future work and concluding remarks.

7.1 Modeling Assumptions

When forming the model for our problem, as in any other field of theoretical research, we had to make certain assumptions and choices. In what follows, we discuss the reasons for the modeling choices we made.

- *Use of gold standard questions:* We assume the existence of gold standard questions in the task, i.e., a subset of questions to which the answers are known to the system designer. The existence of gold standard is commonplace in crowdsourcing platforms (Le et al., 2010; Chen et al., 2011).

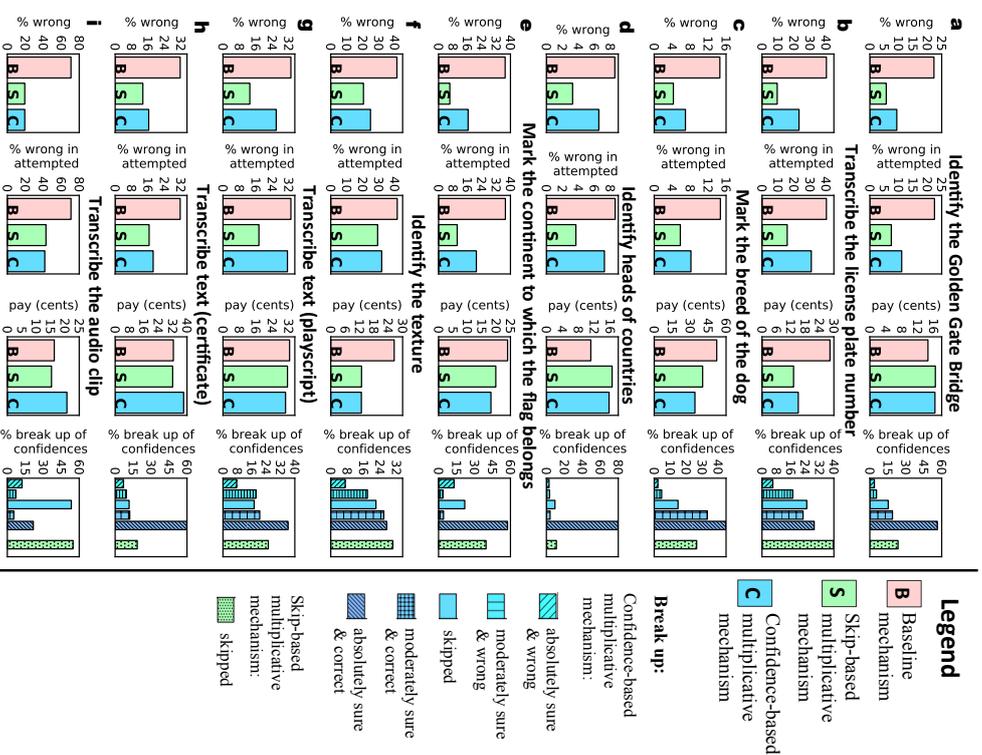


Figure 4: The error-rates in the raw data and payments in the nine experiments. Each individual bar in the plots corresponds to one mechanism in one experiment and is generated from 35 distinct workers (this totals to 945 worker-tasks).

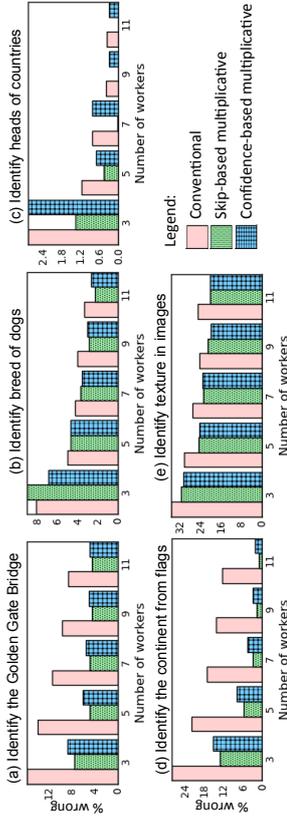


Figure 5: Error-rates in the aggregated data in the five experiments involving multiple-choice tasks.

- Workers aiming to maximize their expected payments:** We assume that the workers aim to maximize their expected payments. In many other problems in game theory, one often makes the assumption that people are “risk-averse”, and aim to maximize the expected value of some “utility function” of their payments. While we extend our results to general utility functions in Appendix C in order to accommodate such requirements, we also think that the assumption of workers maximizing their expected payments is a perfectly reasonable assumption for the crowdsourcing settings considered here. The reason is that each such task lasts for a handful of minutes and is worth a few tens of cents. Workers typically perform tens to hundreds of tasks per day, and consequently their empirical hourly wages very quickly converge to their expectation.
- Workers knowing their confidences:** We understand that in practice the workers will have noisy or granular estimates of their own beliefs. The mathematical assumption of workers knowing their precise confidences is an idealization intended for mathematical tractability. This is one of the reasons why we only elicit a quantized value of the workers’ beliefs (in terms of skipping or choosing one of a finite number of confidence levels), and not try to ask for a precise value.
- Eliciting a quantized version of the beliefs:** We do not directly attempt to elicit the values of the beliefs of the workers, but instead ask them to indicate only a quantization (e.g., “I’m not sure” or “moderately confident”, etc.). We prefer this quantization to direct assessment to real-valued probability, motivated by the extensive literature in psychology on the coarseness of human perception and processing (e.g., (Miller, 1956; Shiffrin and Nosofsky, 1994; Jones and Loe, 2013; Shah et al., 2016a)) establishing that humans are more comfortable at providing quantized responses. This notion is verified by experiments on Amazon Mechanical Turk in Shah et al. (2016a) where it is observed that people are more consistent when giving ordinal answers (comparing pairs of items) as opposed to when they are asked for numeric evaluations.
- Choosing the number of confidence levels L :** In the paper we assume that the number of confidence levels L is specified to us, and we provide mechanisms for any given choice of L . It is an interesting and challenging open problem to choose L for any given application in a principled manner. Up on increasing L , on one hand, we obtain additional nuanced information about the workers’ beliefs, while on the other hand, workers require a greater time and effort to provide

select the confidence level and their answers also tend to get noisier. In other words, both the “signal” and the “noise” in the data increase with an increase in the value of L , and lead to an interesting trade-off.

7.2 Open problems

We discuss two sets of open problems, one from the practical perspective and another on the theoretical front.

First, in the paper, we assumed that the number of total questions N in a task, the number of gold standard questions G , and the threshold T for skipping (or the number and thresholds of the different confidence levels) were provided to the mechanism. While these parameters may be chosen by hand by a system designer based on her own experience, a more principled design of these parameters is an important question. The choices for these parameters may have to be made based on certain tradeoffs. For instance, a higher value of G reduces the variance in the payments but uses more resources in terms of gold standard questions. Or for instance, more number of threshold levels L would increase the amount of information obtained about the workers’ beliefs, but also increase the noise in the workers’ estimates of her own beliefs.

A second open problem is the design of inference algorithms that can exploit the specific structure of the skip-based setting. There are several algorithms and theoretical analyses in the literature for aggregating data from multiple workers in the baseline setting (Raykar et al., 2010; Zhou et al., 2012; Wauthier and Jordan, 2011; Chen et al., 2013; Khetian and Oh, 2016; Dawid and Skene, 1979; Karger et al., 2011; Liu et al., 2012; Vempaty et al., 2014; Zhang et al., 2014; Ipeirotis et al., 2014; Zhou et al., 2015; Shah et al., 2016c). A useful direction of research in the future is to develop algorithms and the theoretical guarantees that incorporate information about the workers’ confidences. For instance, for the skip-based setting, the missing labels are not missing “at random” but are correlated with the difficulty of the task; in the confidence-based setting, we elicit information about the workers’ perceived confidence levels. Designing algorithms that can exploit this information judiciously (e.g., via confidence-weighted worker/item constraints in the minimax entropy method of Zhou et al. (2015)) is a useful direction of future research.

7.3 Conclusions

Despite remarkable progress in machine learning and artificial intelligence, many problems are still not solvable by either humans or machines alone. In recent years, crowdsourcing has emerged as a powerful tool to combine both human and machine intelligence. Crowdsourcing is also a standard means of collecting labeled data for machine learning algorithms. However, crowdsourcing is often plagued with the problem of poor-quality output from workers.

We designed a reward mechanism for crowdsourcing to ensure collection of high-quality data. Under a very natural “no-free-lunch” axiom, we mathematically prove that surprisingly, our mechanism is the only feasible reward mechanism. We further show that among all possible incentive-compatible mechanisms, our “multiplicative” mechanism makes the strictly smallest expenditure on spammers. In preliminary experiments, we observe a significant drop in the error rates under this unique mechanism as compared to basic baseline mechanisms, suggesting that our mechanism has the potential to work well in practice. Our mechanisms offer some additional benefits. The pattern of skips or confidence levels of the workers provide a reasonable estimate of the difficulty of each question. In practice, the questions that are estimated to be more difficult may now be delegated

to an expert or to more non-expert workers. Secondly, the theoretical guarantees of the mechanism may allow for better post-processing of the data, incorporating the confidence information and improving the overall accuracy. The simplicity of the rules of our mechanisms may facilitate an easier adoption among the workers.

In conclusion, given the uniqueness in theory, simplicity, and good performance observed in practice, we envisage our ‘multiplicative’ mechanisms to be of interest to machine learning researchers and practitioners who use crowdsourcing to collect labeled data.

Acknowledgments

The work of Nihar B. Shah was funded in part by a Microsoft Research PhD fellowship. We thank John C. Platt, Christopher J. C. Burges and Christopher Meek for many inspiring discussions. We also thank John C. Platt and Martin J. Wainwright for helping in proof-reading and polishing parts of the manuscript. This work was done when Nihar B. Shah was an intern at Microsoft Research.

Appendix A. Proofs

In this section, we prove the claimed theoretical results whose proofs are not included in the main text of the paper.

The property of incentive-compatibility does not change upon any shift of the mechanism by a constant value or any scaling by a positive constant value. As a result, for the purposes of these proofs, we can assume without loss of generality that $\mu_{\min} = 0$.

A.1 Proof of Lemma 4: The Workhorse Lemma

First we consider the case of $G = N$. In the set $\{y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_G\}$, for some $(\eta, \gamma) \in \{0, \dots, G-1\}^2$ such that $\eta + \gamma + 1 \leq G$, suppose there are η elements with a value $1, \gamma$ elements with a value -1 , and $(G-1-\eta-\gamma)$ elements with a value 0 . Let us assume for now that $i = \eta + \gamma + 1, y_1 = 1, \dots, y_\eta = 1, y_{\eta+1} = -1, \dots, y_{\eta+\gamma} = -1, y_{\eta+\gamma+2} = 0, \dots, y_G = 0$.

Suppose the worker has confidences $(p_1, \dots, p_{\eta+\gamma}) \in (T, 1]^{\eta+\gamma}$ for the first $(\eta + \gamma)$ questions, a confidence of $q \in (0, 1]$ for the next question, and confidences smaller than T for the remaining $(G - \eta - \gamma - 1)$ questions. The mechanism must incentivize the worker to answer the first $(\eta + \gamma)$ questions and skip the last $(G - \eta - \gamma - 1)$ questions; for question $(\eta + \gamma + 1)$, it must incentivize the worker to answer if $q > T$ and skip if $q < T$. Supposing the worker indeed attempts the first $(\eta + \gamma)$ questions and skips the last $(G - \eta - \gamma - 1)$ questions, let $x = \{x_1, \dots, x_{\eta+\gamma}\} \in \{-1, 1\}^{\eta+\gamma}$ denote the evaluation of the worker’s answers to the first $(\eta + \gamma)$ questions. Define quantities $\{r_j\}_{j \in [\eta+\gamma]}$ as $r_j = 1 - p_j$ for $j \in \{1, \dots, \eta\}$, and $r_j = p_j$ for $j \in \{\eta + 1, \eta + \gamma\}$. The requirement

of incentive compatibility necessitates

$$\begin{aligned} & q \sum_{x \in \{-1, 1\}^{\eta+\gamma}} \left(f(x_1, \dots, x_\eta, -x_{\eta+1}, \dots, -x_{\eta+\gamma}, 1, 0, \dots, 0) \prod_{j \in [\eta+\gamma]} \frac{1-x_j}{r_j} (1-r_j)^{\frac{1+x_j}{2}} \right) \\ & + (1-q) \sum_{x \in \{-1, 1\}^{\eta+\gamma}} \left(f(x_1, \dots, x_\eta, -x_{\eta+1}, \dots, -x_{\eta+\gamma}, -1, 0, \dots, 0) \prod_{j \in [\eta+\gamma]} \frac{1-x_j}{r_j} (1-r_j)^{\frac{1+x_j}{2}} \right) \\ & \stackrel{q < T}{\leq} \sum_{q > T} \sum_{x \in \{-1, 1\}^{\eta+\gamma}} \left(f(x_1, \dots, x_\eta, -x_{\eta+1}, \dots, -x_{\eta+\gamma}, 0, 0, \dots, 0) \prod_{j \in [\eta+\gamma]} \frac{1-x_j}{r_j} (1-r_j)^{\frac{1+x_j}{2}} \right). \end{aligned}$$

The left hand side of this expression is the expected payment if the worker chooses to answer question $(\eta + \gamma + 1)$, while the right hand side is the expected payment if she chooses to skip it. For any real-valued variable q , and for any real-valued constants a, b and c ,

$$aq \stackrel{q < c}{\leq} b \stackrel{q > c}{\Rightarrow} ac = b.$$

As a result,

$$\begin{aligned} & T \sum_{x \in \{-1, 1\}^{\eta+\gamma}} \left(f(x_1, \dots, x_\eta, -x_{\eta+1}, \dots, -x_{\eta+\gamma}, 1, 0, \dots, 0) \prod_{j \in [\eta+\gamma]} \frac{1-x_j}{r_j} (1-r_j)^{\frac{1+x_j}{2}} \right) \\ & + (1-T) \sum_{x \in \{-1, 1\}^{\eta+\gamma}} \left(f(x_1, \dots, x_\eta, -x_{\eta+1}, \dots, -x_{\eta+\gamma}, -1, 0, \dots, 0) \prod_{j \in [\eta+\gamma]} \frac{1-x_j}{r_j} (1-r_j)^{\frac{1+x_j}{2}} \right) \\ & - \sum_{x \in \{-1, 1\}^{\eta+\gamma}} \left(f(x_1, \dots, x_\eta, -x_{\eta+1}, \dots, -x_{\eta+\gamma}, 0, 0, \dots, 0) \prod_{j \in [\eta+\gamma]} \frac{1-x_j}{r_j} (1-r_j)^{\frac{1+x_j}{2}} \right) = 0. \end{aligned} \tag{8}$$

The left hand side of (8) represents a polynomial in $(\eta + \gamma)$ variables $\{r_j\}_{j=1}^{\eta+\gamma}$ which evaluates to zero for all values of the variables within a $(\eta + \gamma)$ -dimensional solid Euclidean ball. Thus, the coefficients of the monomials in this polynomial must be zero. In particular, the constant term must be zero. The constant term appears when $x_j = 1 \forall j$ in the summations in (8). Setting the constant term to zero gives

$$\begin{aligned} & T f(x_1 = 1, \dots, x_\eta = 1, -x_{\eta+1} = -1, \dots, -x_{\eta+\gamma} = -1, 1, 0, \dots, 0) \\ & + (1-T) f(x_1 = 1, \dots, x_\eta = 1, -x_{\eta+1} = -1, \dots, -x_{\eta+\gamma} = -1, -1, 0, \dots, 0) \\ & - f(x_1 = 1, \dots, x_\eta = 1, -x_{\eta+1} = -1, \dots, -x_{\eta+\gamma} = -1, 0, 0, \dots, 0) = 0, \end{aligned}$$

as desired. Since the arguments above hold for any permutation of the G questions, this completes the proof for the case of $G = N$.

Now consider the case $G < N$. Let $g : \{-1, 0, 1\}^N \rightarrow \mathbb{R}_+$ represent the expected payment given an evaluation of all the N answers, when the identities of the gold standard questions are

unknown. Here, the expectation is with respect to the (uniformly random) choice of the G gold standard questions. If $(x_1, \dots, x_N) \in \{-1, 0, 1\}^N$ are the evaluations of the worker's answers to the N questions then the expected payment is

$$g(x_1, \dots, x_N) = \frac{1}{\binom{N}{G}} \sum_{\{i_1, \dots, i_G\} \subseteq \{1, \dots, N\}} f(x_{i_1}, \dots, x_{i_G}). \quad (9)$$

Notice that when $G = N$, the functions f and g are identical.

In the set $\{y_1, \dots, y_{\eta-1}, y_{\eta+1}, \dots, y_G\}$, for some $(\eta, \gamma) \in \{0, \dots, G-1\}^2$ with $\eta + \gamma < G$, suppose there are η elements with a value 1, γ elements with a value -1 , and $(G-1-\eta-\gamma)$ elements with a value 0. Let us assume for now that $i = \eta + \gamma + 1$, $y_i = 1, \dots, y_\eta = 1, y_{\eta+1} = -1, \dots, y_{\eta+\gamma} = -1, y_{\eta+\gamma+2} = 0, \dots, y_G = 0$.

Suppose the worker has confidences $\{p_1, \dots, p_{\eta+\gamma}\} \in (T, 1]^{\eta+\gamma}$ for the first $(\eta + \gamma)$ of the N questions, a confidence of $q \in (0, 1]$ for the next question, and confidences smaller than T for the remaining $(N - \eta - \gamma - 1)$ questions. The mechanism must incentivize the worker to answer the first $(\eta + \gamma)$ questions and skip the last $(N - \eta - \gamma - 1)$ questions; for the $(\eta + \gamma + 1)^{\text{th}}$ question, the mechanism must incentivize the worker to answer if $q > T$ and skip if $q < T$. Supposing the worker indeed attempts the first $(\eta + \gamma)$ questions and skips the last $(N - \eta - \gamma - 1)$ questions, let $x = \{x_1, \dots, x_{\eta+\gamma}\} \in \{-1, 1\}^{\eta+\gamma}$ denote the evaluations of the worker's answers to the first $(\eta + \gamma)$ questions. Define quantities $\{r_j\}_{j \in [\eta+\gamma]}$ as $r_j = 1 - p_j$ for $j \in \{1, \dots, \eta\}$, and $r_j = p_j$ for $j \in \{\eta + 1, \eta + \gamma\}$. The requirement of incentive compatibility necessitates

$$\begin{aligned} q & \sum_{x \in \{-1, 1\}^{\eta+\gamma}} \left(g(x_1, \dots, x_\eta, -x_{\eta+1}, \dots, -x_{\eta+\gamma}, 1, 0, \dots, 0) \prod_{j \in [\eta+\gamma]} r_j^{\frac{1-x_j}{2}} (1-r_j)^{\frac{1+x_j}{2}} \right) \\ & + (1-q) \sum_{x \in \{-1, 1\}^{\eta+\gamma}} \left(g(x_1, \dots, x_\eta, -x_{\eta+1}, \dots, -x_{\eta+\gamma}, -1, 0, \dots, 0) \prod_{j \in [\eta+\gamma]} r_j^{\frac{1-x_j}{2}} (1-r_j)^{\frac{1+x_j}{2}} \right) \\ & \sum_{q < T} \sum_{q > T} \left(g(x_1, \dots, x_\eta, -x_{\eta+1}, \dots, -x_{\eta+\gamma}, 0, 0, \dots, 0) \prod_{j \in [\eta+\gamma]} r_j^{\frac{1-x_j}{2}} (1-r_j)^{\frac{1+x_j}{2}} \right). \end{aligned} \quad (10)$$

Again, applying the fact that for any real-valued variable q and for any real-valued constants a, b and $c, aq \sum_{q < c} b \sum_{q > c} b \Rightarrow ac = b$, we get that

$$\begin{aligned} Tg(x_1 = 1, \dots, x_\eta = 1, -x_{\eta+1} = -1, \dots, -x_{\eta+\gamma} = -1, 1, 0, \dots, 0) \\ + (1-T)g(x_1 = 1, \dots, x_\eta = 1, -x_{\eta+1} = -1, \dots, -x_{\eta+\gamma} = -1, -1, 0, \dots, 0) \\ - g(x_1 = 1, \dots, x_\eta = 1, -x_{\eta+1} = -1, \dots, -x_{\eta+\gamma} = -1, 0, 0, \dots, 0) = 0. \end{aligned} \quad (11)$$

The proof now proceeds via induction on the quantity $(G - \eta - \gamma - 1)$, i.e., on the number of skipped questions in $\{y_1, \dots, y_{\eta-1}, y_{\eta+1}, \dots, y_G\}$. We begin with the case of $(G - \eta - \gamma - 1) = G - 1$ which implies $\eta = \gamma = 0$. In this case (11) simplifies to

$$Tg(1, 0, \dots, 0) + (1-T)g(-1, 0, \dots, 0) = g(0, 0, \dots, 0).$$

Applying the expansion of function g in terms of function f from (9) gives

$$\begin{aligned} T(c_1 f(1, 0, \dots, 0) + c_2 f(0, 0, \dots, 0)) + (1-T)(c_1 f(-1, 0, \dots, 0) + c_2 f(0, 0, \dots, 0)) \\ = (c_1 f(0, 0, \dots, 0) + c_2 f(0, 0, \dots, 0)) \end{aligned}$$

for constants $c_1 > 0$ and $c_2 > 0$ that respectively denote the probabilities that the first question is picked and not picked in the set of G gold standard questions. Canceling out the common terms on both sides of the equation, we get the desired result

$$Tf(1, 0, \dots, 0) + (1-T)f(-1, 0, \dots, 0) = f(0, 0, \dots, 0).$$

Next, we consider the case when $(G - \eta - \gamma - 1)$ questions are skipped in the gold standard, and assume that the result is true when more than $(G - \eta - \gamma - 1)$ questions are skipped in the gold standard. In (11), the functions g decompose into a sum of the constituent f functions. These constituent functions f are of two types: the first where all of the first $(\eta + \gamma + 1)$ questions are included in the gold standard, and the second where one or more of the first $(\eta + \gamma + 1)$ questions are not included in the gold standard. The second case corresponds to situations where there are more than $(G - \eta - \gamma - 1)$ questions skipped in the gold standard and hence satisfies our induction hypothesis. The terms corresponding to these functions thus cancel out in the expansion of (11). The remainder comprises only evaluations of function f for arguments in which the first $(\eta + \gamma + 1)$ questions are included in the gold standard: since the last $(N - \eta - \gamma - 1)$ questions are skipped by the worker, the remainder evaluates to

$$\begin{aligned} Tc_3 f(y_1, \dots, y_{\eta+\gamma}, 1, 0, \dots, 0) + (1-T)c_3 f(y_1, \dots, y_{\eta+\gamma}, -1, 0, \dots, 0) \\ = c_3 f(y_1, \dots, y_{\eta+\gamma}, 0, 0, \dots, 0) \end{aligned}$$

for some constant $c_3 > 0$. Dividing throughout by c_3 gives the desired result.

Finally, the arguments above hold for any permutation of the first G questions, thus completing the proof.

A.2 Proof of Theorem 7: Working of Algorithm 2

We first state three properties that the constants $\{\alpha_l\}_{l=-L}^L$ defined in Algorithm 2 must satisfy. We will use these properties subsequently in the proof of Theorem 7.

Lemma 14 For every $l \in \{0, \dots, L-1\}$

$$T_{l+1}\alpha_{l+1} + (1-T_{l+1})\alpha_{-(l+1)} = T_{l+1}\alpha_l + (1-T_{l+1})\alpha_{-l}, \quad (12a)$$

and

$$S_{l+1}\alpha_{l+1} + (1-S_{l+1})\alpha_{-(l+1)} = \alpha_0 = 1. \quad (12b)$$

Lemma 15 $\alpha_L > \alpha_{L-1} > \dots > \alpha_{-L} = 0$.

Lemma 16 For any $m \in \{1, \dots, L\}$, any $p > T_m$ and any $z < m$,

$$p\alpha_m + (1-p)\alpha_{-m} > p\alpha_z + (1-p)\alpha_{-z}, \quad (13a)$$

and for any $m \in \{0, \dots, L-1\}$, any $p < T_{m+1}$ and any $z > m$,

$$p\alpha_m + (1-p)\alpha_{-m} > p\alpha_z + (1-p)\alpha_{-z}. \quad (13b)$$

The proof of these results are available at the end of this subsection. Assuming these lemmas hold, we will now complete the proof of Theorem 7.

The choice of $\alpha_{-L} = 0$ made in Algorithm 2 ensures that the payment is zero whenever any answer in the gold standard evaluates to $-L$. This choice ensures that the no-free-lunch axiom is satisfied. One can easily verify that the payment lies in the interval $[0, \mu_{\max}]$. It remains to prove that the proposed mechanism is incentive-compatible.

Define $E = (\epsilon_1, \dots, \epsilon_G) \in \{-1, 1\}^G$ and $E_{\setminus 1} = (\epsilon_2, \dots, \epsilon_G)$. Suppose the worker has confidences p_1, \dots, p_N for her N answers. For some $(s(1), \dots, s(N)) \in \{0, \dots, L\}^N$ suppose $p_i \in (T_{s(i)}; T_{s(i+1)}) \forall i \in \{1, \dots, N\}$, i.e., $s(1), \dots, s(N)$ are the correct confidence-levels for her answers. Consider any other set of confidence-levels $s'(1), \dots, s'(N)$. When the mechanism of Algorithm 2 is employed, the expected payment (from the point of view of the worker) on selecting confidence-levels $s(1), \dots, s(N)$ is

$$\mathbb{E}[\text{Pay}] = \frac{1}{\binom{N}{G}} \sum_{\substack{(j_1, \dots, j_G) \\ \subseteq \{1, \dots, N\}}} \sum_{E \in \{-1, 1\}^G} \prod_{i=1}^G \alpha_{\epsilon_i s(j_i)} (p_{j_i})^{\frac{1+\epsilon_i}{2}} (1-p_{j_i})^{\frac{1-\epsilon_i}{2}} \quad (14a)$$

$$= \frac{1}{\binom{N}{G}} \sum_{\substack{(j_1, \dots, j_G) \\ \subseteq \{1, \dots, N\}}} \sum_{E_{\setminus 1} \in \{-1, 1\}^{G-1}} (p_{j_1} \alpha_{s(j_1)} + (1-p_{j_1}) \alpha_{-s(j_1)}) \prod_{i=2}^G \alpha_{\epsilon_i s(j_i)} (p_{j_i})^{\frac{1+\epsilon_i}{2}} (1-p_{j_i})^{\frac{1-\epsilon_i}{2}} \quad (14b)$$

⋮

$$= \frac{1}{\binom{N}{G}} \sum_{\substack{(j_1, \dots, j_G) \\ \subseteq \{1, \dots, N\}}} \prod_{i=1}^G (p_{j_i} \alpha_{s(j_i)} + (1-p_{j_i}) \alpha_{-s(j_i)}) \quad (14c)$$

$$> \frac{1}{\binom{N}{G}} \sum_{\substack{(j_1, \dots, j_G) \\ \subseteq \{1, \dots, N\}}} \prod_{i=1}^G (p_{j_i} \alpha_{s'(j_i)} + (1-p_{j_i}) \alpha_{-s'(j_i)}) \quad (14d)$$

which is the expected payment under any other set of confidence-levels $s'(1), \dots, s'(N)$. The last inequality is a consequence of Lemma 16.

An argument similar to the above also proves that for any $m \in \{1, \dots, L\}$, if allowed to choose level m over skip if her confidence is greater S_{m_m} and choose skip over level m if her confidence is smaller than S_{m_m} . Finally, from Lemma 15 we have $\alpha_{-L} > \dots > \alpha_{-L} = 0$. It follows that the expected payment (14c) is strictly increasing in each of the values p_1, \dots, p_N . Thus the worker is incentivized to report the answer that she thinks is most likely to be correct.

A.2.1. PROOF OF LEMMA 14

Algorithm 2 states that $\alpha_{-l} = \frac{1-\alpha_l S_l}{1-S_l}$ for all $l \in [L]$. A simple rearrangement of the terms in this expression gives (12b).

Towards the goal of proving (12a), we will first prove an intermediate result:

$$\alpha_l > 1 > \alpha_{-l} \forall l \in \{L, \dots, 1\}. \quad (15)$$

The proof proceeds via an induction on $l \in \{L, \dots, 2\}$. The case of $l = 1$ will be proved separately. The induction hypothesis involves two claims: $\alpha_l > 1 > \alpha_{-l}$ and $T_l \alpha_l + (1-T_l) \alpha_{-l} > 1$. The base case is $l = L$ for which we know that $\alpha_L = \frac{1}{S_L} > 1 > 0 = \alpha_{-L}$ and $T_l \alpha_l + (1-T_l) \alpha_{-l} = \frac{T_l}{S_l} > 1$. Now suppose that the induction hypothesis is true for $(l+1)$. Rearranging the terms in the expression defining α_l in Algorithm 2 and noting that $1 > T_{l+1} > S_l$, we get

$$\begin{aligned} \alpha_l &= \frac{(1-S_l)(T_{l+1} \alpha_{l+1} + (1-T_{l+1}) \alpha_{-(l+1)}) - (1-T_{l+1})}{(1-S_l) - (1-T_{l+1})} \\ &> \frac{(1-S_l) - (1-T_{l+1})}{(1-S_l) - (1-T_{l+1})} \\ &= 1. \end{aligned} \quad (16)$$

From (12b) we see that the value 1 is a convex combination of α_l and α_{-l} . Since $\alpha_l > 1$ and $S_l \in (0, 1)$, it must be that $\alpha_{-l} < 1$. Furthermore, since $T_l > S_l$ we get

$$\begin{aligned} T_l \alpha_l + (1-T_l) \alpha_{-l} &> S_l \alpha_l + (1-S_l) \alpha_{-l} \\ &= 1. \end{aligned}$$

This proves the induction hypothesis. Let us now consider $l = 1$. If $L = 1$ then we have $\alpha_L = \frac{1}{S_L} > 1 > 0 = \alpha_{-L}$ and we are done. If $L > 1$ then we have already proved that $\alpha_2 > 1 > \alpha_{-2}$ and $T_2 \alpha_2 + (1-T_2) \alpha_{-2} > 1$. An argument identical to (16) onwards proves that $\alpha_1 > 1 > \alpha_{-1}$.

Now that we have proved $\alpha_l > \alpha_{-l} \forall l \in [L]$, we can rewrite the expression defining α_{-l} in Algorithm 2 as

$$S_l = \frac{1-\alpha_{-l}}{\alpha_l - \alpha_{-l}}.$$

Substituting this expression for S_l in the definition of α_l in Algorithm 2 and making some simple rearrangements gives the desired result (12a).

A.2.2. PROOF OF LEMMA 15

We have already shown (15) in the proof of Lemma 14 above that $\alpha_l > 1 > \alpha_{-l} \forall l \in [L]$.

Next we will show that $\alpha_{l+1} > \alpha_l$ and $\alpha_{-(l+1)} < \alpha_{-l} \forall l \geq 0$. First consider $l = 0$, for which Algorithm 2 sets $\alpha_0 = 1$, and we have already proved that $\alpha_1 > 1 > \alpha_{-1}$.

Now consider some $l > 0$. Observe that since $S_l \alpha_l + (1-S_l) \alpha_{-l} = 1$ (Lemma 14), $S_{l+1} > S_l$ and $\alpha_l > \alpha_{-l}$, it must be that

$$S_{l+1} \alpha_l + (1-S_{l+1}) \alpha_{-l} > 1. \quad (17)$$

From Lemma 14, we also have

$$S_{l+1} \alpha_{l+1} + (1-S_{l+1}) \alpha_{-(l+1)} = 1. \quad (18)$$

Subtracting (17) from (18) we get

$$S_{l+1} (\alpha_{l+1} - \alpha_l) + (1-S_{l+1}) (\alpha_{-(l+1)} - \alpha_{-l}) < 0. \quad (19)$$

From Lemma 14 we also have

$$T_{l+1}\alpha_{l+1} + (1 - T_{l+1})\alpha_{-(l+1)} = T_{l+1}\alpha_l + (1 - T_{l+1})\alpha_{-l} \quad (20)$$

$$\Rightarrow T_{l+1}(\alpha_{l+1} - \alpha_l) + (1 - T_{l+1})(\alpha_{-(l+1)} - \alpha_{-l}) = 0. \quad (21)$$

Subtracting (19) from (21) gives

$$(T_{l+1} - S_{l+1})(\alpha_{l+1} - \alpha_l) + (\alpha_{-l} - \alpha_{-(l+1)}) > 0. \quad (22)$$

Since $T_{l+1} > S_{l+1}$ by definition, it must be that

$$\alpha_{l+1} - \alpha_l > \alpha_{-(l+1)} - \alpha_{-l}. \quad (23)$$

Now, rearranging the terms in (20) gives

$$(\alpha_{l+1} - \alpha_l)T_{l+1} = -(\alpha_{-(l+1)} - \alpha_{-l})(1 - T_{l+1}). \quad (24)$$

Since $T_{l+1} \in (0, 1)$, it follows that the terms $(\alpha_{l+1} - \alpha_l)$ and $(\alpha_{-(l+1)} - \alpha_{-l})$ have opposite signs. Using (23) we conclude that $\alpha_{l+1} - \alpha_l > 0$ and $\alpha_{-(l+1)} - \alpha_{-l} < 0$.

A.2.3 PROOF OF LEMMA 16

Let us first prove (13a). First consider the case $z = m - 1$. From Lemma 14 we know that

$$T_m\alpha_{m-1} + (1 - T_m)\alpha_{-(m-1)} = T_m\alpha_m + (1 - T_m)\alpha_{-m},$$

and hence

$$\begin{aligned} 0 &= T_m(\alpha_m - \alpha_{m-1}) + T_m(\alpha_{-(m-1)} - \alpha_{-m}) - (\alpha_{-(m-1)} - \alpha_{-m}) \\ &< p(\alpha_m - \alpha_{m-1}) + p(\alpha_{-(m-1)} - \alpha_{-m}) - (\alpha_{-(m-1)} - \alpha_{-m}), \end{aligned} \quad (25)$$

where (25) is a consequence of $p > T_m$ and Lemma 15. A simple rearrangement of the terms in (25) gives (13a). Now, for any $z < m$, recursively apply this result to get

$$\begin{aligned} p\alpha_m + (1 - p)\alpha_{-m} &> p\alpha_{m-1} + (1 - p)\alpha_{-(m-1)} \\ &> p\alpha_{m-2} + (1 - p)\alpha_{-(m-2)} \\ &\vdots \\ &> p\alpha_z + (1 - p)\alpha_{-z}. \end{aligned}$$

Let us now prove (13b). We first consider the case $z = m + 1$. From Lemma 14 we know that

$$T_{m+1}\alpha_m + (1 - T_{m+1})\alpha_{-m} = T_{m+1}\alpha_{m+1} + (1 - T_{m+1})\alpha_{-(m+1)},$$

and hence

$$\begin{aligned} 0 &= T_{m+1}(\alpha_{m+1} - \alpha_m) + T_{m+1}(\alpha_{-m} - \alpha_{-(m+1)}) - (\alpha_{-m} - \alpha_{-(m+1)}) \\ &> p(\alpha_{m+1} - \alpha_m) + p(\alpha_{-m} - \alpha_{-(m+1)}) - (\alpha_{-m} - \alpha_{-(m+1)}), \end{aligned} \quad (26)$$

where (26) is a consequence of $p < T_{m+1}$ and Lemma 15. A simple rearrangement of the terms in (26) gives (13b). For any $z > m$, applying this result recursively gives

$$\begin{aligned} p\alpha_m + (1 - p)\alpha_{-m} &> p\alpha_{m+1} + (1 - p)\alpha_{-(m+1)} \\ &> p\alpha_{m+2} + (1 - p)\alpha_{-(m+2)} \\ &\vdots \\ &> p\alpha_z + (1 - p)\alpha_{-z}. \end{aligned}$$

A.3 Proof of Theorem 9: Uniqueness of Algorithm 2

We will first define one additional piece of notation. Let $g : \{-L, \dots, L\}^N \rightarrow \mathbb{R}_+$ denote the expected payment given an evaluation of the N answers, where the expectation is with respect to the (uniformly random) choice of the G gold standard questions. If $(x_1, \dots, x_N) \in \{-L, \dots, L\}^N$ are the evaluations of the worker's answers to the N questions then the expected payment is

$$g(x_1, \dots, x_N) = \frac{1}{\binom{N}{G}} \sum_{\{i_1, \dots, i_G\} \subseteq \{1, \dots, N\}} f(x_{i_1}, \dots, x_{i_G}). \quad (27)$$

Notice that when $G = N$, the functions f and g are identical.

The proof of uniqueness is based on a certain condition necessitated by incentive-compatibility stated in the form of Lemma 17 below. Note that this lemma does *not* require the generalized-no-free-lunch axiom, and may be of independent interest.

Lemma 17 *Any incentive-compatible mechanism must satisfy, for every question $i \in \{1, \dots, G\}$, every*

$$(y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_G) \in \{-L, \dots, L\}^{G-1}, \text{ and every } m \in \{1, \dots, L\},$$

$$\begin{aligned} T_m f(y_1, \dots, y_{i-1}, m, y_{i+1}, \dots, y_G) + (1 - T_m) f(y_1, \dots, y_{i-1}, -m, y_{i+1}, \dots, y_G) \\ = T_m f(y_1, \dots, y_{i-1}, m - 1, y_{i+1}, \dots, y_G) + (1 - T_m) f(y_1, \dots, y_{i-1}, -(m - 1), y_{i+1}, \dots, y_G) \end{aligned} \quad (28a)$$

and

$$\begin{aligned} S_m f(y_1, \dots, y_{i-1}, m, y_{i+1}, \dots, y_G) + (1 - S_m) f(y_1, \dots, y_{i-1}, -m, y_{i+1}, \dots, y_G) \\ = f(y_1, \dots, y_{i-1}, 0, y_{i+1}, \dots, y_G). \end{aligned} \quad (28b)$$

Note that (28a) and (28b) coincide when $m = 1$, since $T_1 = S_1$ by definition.

We first prove that any incentive compatible mechanism that satisfies the no-free-lunch axiom must give a zero payment when one or more questions are selected with a confidence L and turn out to be incorrect. Let us assume for now that in the G questions in the gold standard, the first question is answered incorrectly with a confidence of L , the next $(G - 1 - S)$ questions are answered by the worker and have arbitrary evaluations, and the remaining S questions are skipped. The proof proceeds by an induction on S . If $S = G - 1$, the only attempted question is the first question and this is incorrect with confidence L . The no-free-lunch axiom necessitates a zero payment in this

case, thus satisfying the base case of our induction hypothesis. Now we prove the hypothesis for some S under the assumption that the hypothesis is true for every $S' > S$. From Lemma 4 with $m = 1$, we have

$$\begin{aligned} T_1 f(-L, y_2, \dots, y_{G-S-1}, 1, 0, \dots, 0) + (1 - T_1) f(-L, y_2, \dots, y_{G-S-1}, -1, 0, \dots, 0) \\ = T_1 f(-L, y_2, \dots, y_{G-S-1}, 0, 0, \dots, 0) + (1 - T_1) f(-L, y_2, \dots, y_{G-S-1}, 0, 0, \dots, 0) \\ = f(-L, y_2, \dots, y_{G-S-1}, 0, 0, \dots, 0) \\ = 0, \end{aligned} \quad (29)$$

where the final equation (29) is a consequence of our induction hypothesis given the fact that $f(-L, y_2, \dots, y_{G-S-1}, 0, 0, \dots, 0)$ corresponds to the case when the last $(S + 1)$ questions are skipped and the first question is answered incorrectly with confidence L . Now, since the payment f must be non-negative and since $T \in (0, 1)$, it must be that

$$f(-L, y_2, \dots, y_{G-S-1}, 1, 0, \dots, 0) = 0 \quad (30a)$$

and

$$f(-L, y_2, \dots, y_{G-S-1}, -1, 0, \dots, 0) = 0. \quad (30b)$$

Repeatedly applying the same argument to $m = 2, \dots, L$ gives that for every value of m , it must be that $f(-L, y_2, \dots, y_{G-S-1}, m, 0, \dots, 0) = f(-L, y_2, \dots, y_{G-S-1}, -m, 0, \dots, 0) = 0$. This completes the proof of our induction hypothesis. Observe that each of the aforementioned arguments hold for any permutation of the G questions, thus proving the necessity of zero payment when any one or more answers are incorrect.

We will now prove that when no answers in the gold standard are incorrect with confidence L , the payment must be of the form described in Algorithm 1. Let κ denote the payment when all G questions in the gold standard are skipped, i.e.,

$$\kappa = f(0, \dots, 0).$$

Now consider any $S \in \{0, \dots, G - 1\}$ and any $(y_1, \dots, y_{G-S-1}, m) \in \{-L, \dots, L\}^{G-S}$. The payments $\{f(y_1, \dots, y_{G-S-1}, m, 0, \dots, 0)\}_{m=-L}^L$ must satisfy the $(2L - 1)$ linear constraints arising out of Lemma 17 and must also satisfy $f(y_1, \dots, y_{G-S-1}, -L, 0, \dots, 0) = 0$. This set of conditions comprises a total of $2L$ linearly independent constraints on the set of $(2L + 1)$ values $\{f(y_1, \dots, y_{G-S-1}, m, 0, \dots, 0)\}_{m=-L}^L$. The only set of solutions that meet these constraints are

$$f(y_1, \dots, y_{G-S-1}, m, 0, \dots, 0) = \alpha_m f(y_1, \dots, y_{G-S-1}, 0, 0, \dots, 0),$$

where the constants $\{\alpha_m\}_{m=-L}^L$ are as specified in Algorithm 2. Applying this argument G times, starting from $S = 0$ to $S = G - 1$, gives

$$f(y_1, \dots, y_G) = \kappa \prod_{j=1}^G \alpha_{y_j}.$$

Finally, the budget requirement necessitates $\mu_{\max} = \kappa (\alpha_L)^G$, which mandates the value of κ to be $\mu_{\max} \left(\frac{1}{\alpha_L}\right)^G$. This is precisely the mechanism described in Algorithm 2.

A.4 Proof of Lemma 17: Necessary condition for any incentive-compatible mechanism

First consider the case of $G = N$. For every $j \in \{1, \dots, i - 1, i + 1, \dots, G\}$, define

$$r_j = \begin{cases} 1 - p_j & \text{if } y_j \geq 0 \\ p_j & \text{if } y_j < 0. \end{cases}$$

Define $E_{y_i} = \{\epsilon_1, \dots, \epsilon_{i-1}, \epsilon_{i+1}, \dots, \epsilon_G\}$. For any $l \in \{-L, \dots, L\}$ let $\Lambda_l \in \mathbb{R}_+$ denote the expected payment (from the worker's point of view) when her answer to the i^{th} question evaluates to l :

$$\Lambda_l = \sum_{E_{y_i} \in \{-1, 1\}^{G-1}} \left(f(y_1 \epsilon_1, \dots, y_{i-1} \epsilon_{i-1}, l, y_{i+1} \epsilon_{i+1}, \dots, y_G \epsilon_G) \prod_{j \in [G] \setminus \{i\}} r_j^{\frac{1-\epsilon_j}{2}} (1 - r_j)^{\frac{1+\epsilon_j}{2}} \right). \quad (31)$$

Consider a worker who has confidences $\{p_1, \dots, p_{i-1}, p_{i+1}, \dots, p_G\} \in (0, 1)^{G-1}$ for questions $\{1, \dots, i - 1, i + 1, \dots, G\}$ respectively, and for question i suppose she has a confidence of $q \in (T_{m-1}, T_{m+1})$. For question i , we must incentivize the worker to select confidence-level m if $q > T_m$, and to select $(m - 1)$ if $q < T_m$. This necessitates

$$q \Lambda_m + (1 - q) \Lambda_{-m} \stackrel{q < T_m}{\leq} \sum_{q > T_m} q \Lambda_{m-1} + (1 - q) \Lambda_{-(m-1)}. \quad (32)$$

Also, for question i , the requirement of level m having a higher incentive as compared to skipping when $q > S_m$ and vice versa when $q < S_m$ necessitates

$$q \Lambda_m + (1 - q) \Lambda_{-m} \stackrel{q < S_m}{\leq} \sum_{q > S_m} \Lambda_0. \quad (33)$$

Now, note that for any real-valued variable q , and for any real-valued constants a, b and c ,

$$aq \stackrel{q < c}{\leq} b \Rightarrow ac = b, \quad aq \stackrel{q > c}{\geq} b$$

Applying this fact to (32) and (33) gives

$$(T_m \Lambda_m + (1 - T_m) \Lambda_{-m}) - (T_m \Lambda_{m-1} + (1 - T_m) \Lambda_{-(m-1)}) = 0, \quad (34a)$$

$$(S_m \Lambda_m + (1 - S_m) \Lambda_{-m}) - \Lambda_0 = 0. \quad (34b)$$

From the definition of Λ_l in (31), we see that the left hand sides of (34a) and (34b) are both polynomials in $(G - 1)$ variables $\{r_j\}_{j \in [G] \setminus \{i\}}$ and take a value of zero for all values of the variables in a $(G - 1)$ -dimensional solid ball. Thus, each of the coefficients (of the monomials) in both polynomials must be zero, and in particular, the constant terms must also be zero. Observe that in both these polynomials, the constant term arises only when $\epsilon_j = 1 \forall j \in [G] \setminus \{i\}$ (which makes the

exponent of r_j to be 0 and that of $(1 - r_j)$ to be 1). Thus, setting the constant term to zero in the two polynomials results in

$$\begin{aligned} & T_m f(y_1, \dots, y_{i-1}, m, y_{i+1}, \dots, y_G) + (1 - T_m) f(y_1, \dots, y_{i-1}, -m, y_{i+1}, \dots, y_G) \\ &= T_m f(y_1, \dots, y_{i-1}, m-1, y_{i+1}, \dots, y_G) + (1 - T_m) f(y_1, \dots, y_{i-1}, -(m-1), y_{i+1}, \dots, y_G) \end{aligned} \quad (35a)$$

and

$$\begin{aligned} & S_m f(y_1, \dots, y_{i-1}, m, y_{i+1}, \dots, y_G) + (1 - S_m) f(y_1, \dots, y_{i-1}, -m, y_{i+1}, \dots, y_G) \\ &= f(y_1, \dots, y_{i-1}, 0, y_{i+1}, \dots, y_G) \end{aligned} \quad (35b)$$

thus proving the claim for the case of $G = N$.

Now consider the case when $G < N$. In order to simplify notation, let us assume $i = 1$ without loss of generality (since the arguments presented hold for any permutation of the questions). Suppose a worker's answers to questions $\{2, \dots, G\}$ evaluate to $(y_2, \dots, y_G) \in \{-L, \dots, L\}^{G-1}$, and further suppose that the worker skips the remaining $(N - G)$ questions. By going through arguments identical to those for $G = N$, but with f replaced by g , we get the necessity of

$$\begin{aligned} & T_m g(m, y_2, \dots, y_G, 0, \dots, 0) + (1 - T_m) g(-m, y_2, \dots, y_G, 0, \dots, 0) \\ &= T_m g(m-1, y_2, \dots, y_G, 0, \dots, 0) + (1 - T_m) g(-(m-1), y_2, \dots, y_G, 0, \dots, 0) \end{aligned} \quad (36a)$$

and

$$\begin{aligned} & S_m g(m, y_2, \dots, y_G, 0, \dots, 0) + (1 - S_m) g(-m, y_2, \dots, y_G, 0, \dots, 0) = g(0, y_2, \dots, y_G, 0, \dots, 0). \end{aligned} \quad (36b)$$

We now use this result in terms of function g to get an equivalent result in terms of function f . For some $S \in \{0, \dots, G-1\}$, suppose $y_{G-S+1} = 0, \dots, y_G = 0$. The remaining proof proceeds via an induction on S . We begin with $S = G-1$. In this case, (36a) and (36b) simplify to

$$\begin{aligned} & T_m g(m, 0, \dots, 0) + (1 - T_m) g(-m, 0, 0, \dots, 0) \\ &= T_m g(m-1, 0, \dots, 0) + (1 - T_m) g(-(m-1), 0, \dots, 0) \end{aligned} \quad (37a)$$

and

$$S_m g(m, 0, \dots, 0) + (1 - S_m) g(-m, 0, \dots, 0) = g(0, 0, \dots, 0). \quad (37b)$$

Applying the definition of function g from (27) leads to

$$\begin{aligned} & T_m (c_1 f(m, 0, \dots, 0) + c_2 f(0, 0, \dots, 0)) + (1 - T_m) (c_1 f(-m, 0, \dots, 0) + c_2 f(0, 0, \dots, 0)) \\ &= T_m (c_1 f(m-1, 0, \dots, 0) + c_2 f(0, 0, \dots, 0)) \\ &\quad + (1 - T_m) (c_1 f(-(m-1), 0, \dots, 0) + c_2 f(0, 0, \dots, 0)), \end{aligned} \quad (38a)$$

and

$$\begin{aligned} & S_m (c_1 f(m, 0, \dots, 0) + c_2 f(0, 0, \dots, 0)) + (1 - S_m) (c_1 f(-m, 0, \dots, 0) + c_2 f(0, 0, \dots, 0)) \\ &= (c_1 f(0, 0, \dots, 0) + c_2 f(0, 0, \dots, 0)) \end{aligned} \quad (38b)$$

for constants $c_1 > 0$ and $c_2 > 0$ that respectively denote the probabilities that the first question is picked and not picked in the set of G gold standard questions. Can calling out the common terms on both sides of the equation, we get the desired results

$$\begin{aligned} & T_m f(m, 0, \dots, 0) + (1 - T_m) f(-m, 0, \dots, 0) \\ &= T_m f(m-1, 0, \dots, 0) + (1 - T_m) f(-(m-1), 0, \dots, 0) \end{aligned}$$

and

$$S_m f(m, 0, \dots, 0) + (1 - S_m) f(-m, 0, \dots, 0) = f(0, 0, \dots, 0).$$

Next, we consider the case of a general $S \in \{0, \dots, G-2\}$ and assume that the result is true when $y_{G-S} = 0, \dots, y_G = 0$. In (36a) and (36b), the functions g decompose into a sum of the constituent f functions. These constituent functions f are of two types: the first where all of the first $(G-S)$ questions are included in the gold standard, and the second where one or more of the first $(G-S)$ questions are not included in the gold standard. The second case corresponds to situations where there are more than S questions skipped in the gold standard, i.e., when $y_{G-S} = 0, \dots, y_G = 0$, and hence satisfies our induction hypothesis. The terms corresponding to these functions thus cancel out in the expansion of (36a) and (36b). The remainder comprises only evaluations of function f for arguments in which the first $(G-S)$ questions are included in the gold standard: since the last $(N-G+S)$ questions are skipped by the worker, the remainder evaluates to

$$\begin{aligned} & T_m c_3 f(y_1, \dots, y_{i-1}, m, y_{i+1}, \dots, y_G) + (1 - T_m) c_3 f(y_1, \dots, y_{i-1}, -m, y_{i+1}, \dots, y_G) \\ &= T_m c_3 f(y_1, \dots, y_{i-1}, m-1, y_{i+1}, \dots, y_G) \\ &\quad + (1 - T_m) c_3 f(y_1, \dots, y_{i-1}, -(m-1), y_{i+1}, \dots, y_G), \end{aligned}$$

and

$$\begin{aligned} & S_m c_3 f(y_1, \dots, y_{i-1}, m, y_{i+1}, \dots, y_G) + (1 - S_m) c_3 f(y_1, \dots, y_{i-1}, -m, y_{i+1}, \dots, y_G) \\ &= c_3 f(y_1, \dots, y_{i-1}, 0, y_{i+1}, \dots, y_G), \end{aligned}$$

for some constant $c_3 > 0$. Dividing throughout by c_3 gives the desired result.

Finally, the arguments above hold for any permutation of the first G questions, thus completing the proof.

A.5 Necessity of $T_1 > S_l$ for the Problem to be Well Defined

We now show that the restriction $T_1 > S_l$ was necessary when defining the thresholds in Section 4.

Proposition 18 *Incentive-compatibility necessitates $T_1 > S_l \forall l \in \{2, \dots, L\}$, even in the absence of the generalized-no-free-lunch axiom.*

First observe that the proof of Lemma 17 did not employ the generalized-no-free-lunch axiom, neither did it assume $T_1 > S_l$. We will thus use the result of Lemma 17 to prove our claim.

Suppose the confidence of the worker for all but the first question is lower than T_1 and that the worker decides to skip all these questions. Suppose the worker attempts the first question. In order to ensure that the worker selects the answer that she believes is most likely to be true, it must be that

$$f(l, 0, \dots, 0) > f(-l, 0, \dots, 0) \quad \forall l \in [L].$$

We now call upon Lemma 17 where we set $i = 1$, $m = l$, $y_2 = \dots$, $y_G = 0$. Using the fact that $T_1 > T_{l-1} \forall l \in \{2, \dots, L\}$, we get

$$\begin{aligned} T_1 f(l, 0, \dots, 0) &+ (1 - T_1) f(-l, 0, \dots, 0) \\ &= T_1 f(l - 1, 0, \dots, 0) + (1 - T_1) f(-(l - 1), 0, \dots, 0) \\ &> T_{l-1} f(l - 1, 0, \dots, 0) + (1 - T_{l-1}) f(-(l - 1), 0, \dots, 0) \\ &= T_{l-1} f(l - 2, 0, \dots, 0) + (1 - T_{l-1}) f(-(l - 2), 0, \dots, 0) \\ &> T_{l-2} f(l - 2, 0, \dots, 0) + (1 - T_{l-2}) f(-(l - 2), 0, \dots, 0) \\ &\vdots \\ &> T_1 f(1, 0, \dots, 0) + (1 - T_1) f(-1, 0, \dots, 0) \\ &= f(0, \dots, 0) \\ &= S_l f(l, 0, \dots, 0) + (1 - S_l) f(-l, 0, \dots, 0). \end{aligned}$$

Since $f(l, 0, \dots, 0) > f(-l, 0, \dots, 0)$, we have the claimed result.

A.6 A Stronger No-free-lunch Axiom: Impossibility Results

In this section, we prove the various claims regarding the strong no-free-lunch axiom studied in Section 5.

A.6.1 PROOF OF PROPOSITION 10

If the worker skips all questions, then the expected payment is zero under the strong-no-free-lunch axiom. On the other hand, in order to incentivize knowledgeable workers to select answers whenever their confidences are greater than T , there must exist some situation in which the payment is strictly larger than zero. Suppose the payment is strictly positive when questions $\{1, \dots, z\}$ are answered correctly, questions $\{z + 1, \dots, z'\}$ are answered incorrectly, and the remaining questions are skipped. If the confidence of the unknowledgeable worker is in the interval $(0, T)$ for every question, then attempting to answer questions $\{1, \dots, z'\}$ and skipping the rest fetches her a payment that is strictly positive in expectation. Thus, this unknowledgeable worker is incentivized to answer at least one question.

A.6.2 PROOF OF PROPOSITION 11

Consider a (knowledgeable) worker who has a confidence of $p \in (T, 1]$ for the first question, $q \in (0, 1)$ for the second question, and confidences in the interval $(0, T)$ for the remaining questions. Suppose the worker attempts to answer the first question (and selects the answer she believes is most likely to be correct) and skips the last $(N - 2)$ questions as desired. Now, in order to incentivize her to answer the second question if $q > T$ and skip the second question if $q < T$, the payment

mechanism must satisfy

$$\begin{aligned} pqg(1, 1, 0, \dots, 0) &+ (1 - p)qg(-1, 1, 0, \dots, 0) + p(1 - q)g(1, -1, 0, \dots, 0) \\ &+ (1 - p)(1 - q)g(-1, -1, 0, \dots, 0) \stackrel{q < T}{\leq} pq(1, 0, 0, \dots, 0) + (1 - p)g(-1, 0, 0, \dots, 0). \end{aligned}$$

For any real-valued variable g , and for any real-valued constants a, b and c ,

$$aq \sum_{q > T}^{q < c} b \Rightarrow ac = b.$$

As a result,

$$\begin{aligned} pTg(1, 1, 0, \dots, 0) &+ (1 - p)Tg(-1, 1, 0, \dots, 0) + p(1 - T)g(1, -1, 0, \dots, 0) \\ &+ (1 - p)(1 - T)g(-1, -1, 0, \dots, 0) - pqg(1, 0, 0, \dots, 0) - (1 - p)g(-1, 0, 0, \dots, 0) = 0. \end{aligned}$$

The left hand side of this equation is a polynomial in variable p and takes a value of zero for all values of p in a one-dimensional box $(T, 1]$. It follows that the monomials of this polynomial must be zero, and in particular the constant term must be zero:

$$Tg(-1, 1, 0, \dots, 0) + (1 - T)g(-1, -1, 0, \dots, 0) - g(-1, 0, 0, \dots, 0) = 0.$$

The strong-no-free-lunch axiom implies $f(-1, -1, 0, \dots, 0) = f(-1, 0, \dots, 0) = f(0, \dots, 0) = 0$, and hence $g(-1, -1, 0, \dots, 0) = g(-1, 0, 0, \dots, 0) = 0$. Since $T \in (0, 1]$, we have

$$\begin{aligned} 0 &= g(-1, 1, 0, \dots, 0) \\ &= c_1 f(-1, 1, 0, \dots, 0) + c_2 f(-1, 0, \dots, 0) + c_2 f(1, 0, \dots, 0), \end{aligned} \quad (39)$$

for some constants $c_1 > 0$ and $c_2 > 0$ that represent the probability that the first two questions are included in the gold standard, and the probability that the first (or, second) but not the second (or, first) questions are included in the gold standard. Since f is a non-negative function, it must be that

$$f(1, 0, \dots, 0) = 0.$$

Now suppose a (knowledgeable) worker has a confidence of $p \in (T, 1]$ for the first question and confidences lower than T for the remaining $(N - 1)$ questions. Suppose the worker chooses to skip the last $(N - 1)$ questions as desired. In order to incentivize the worker to answer the first question, the mechanism must satisfy for all $p \in (T, 1]$,

$$\begin{aligned} 0 &< pg(1, 0, \dots, 0) + (1 - p)g(-1, 0, \dots, 0) - g(0, 0, \dots, 0) \\ &= pc_3 f(1, 0, \dots, 0) + pc_4 f(0, 0, \dots, 0) + (1 - p)c_3 f(-1, 0, \dots, 0) \\ &\quad + (1 - p)c_4 f(0, 0, \dots, 0) - f(0, 0, \dots, 0) \\ &= 0, \end{aligned}$$

where $c_3 > 0$ and $c_4 > 0$ are some constants. The final equation is a result of the strong-no-free-lunch axiom and the fact that $f(1, 0, \dots, 0) = 0$ as proved above. This yields a contradiction, and hence no incentive-compatible mechanism f can satisfy the strong-no-free-lunch axiom when $G < N$ even when allowed to address only knowledgeable workers.

Finally, as a sanity check, note that if $G = N$ then $c_2 = 0$ in (39). The proof above thus doesn't hold when $G = N$.

A.6.3. PROOF OF PROPOSITION 12

We will first show that the mechanism works as desired.

First consider the case when the worker is unknowledgeable and her confidences are of the form $T > p_{(1)} \geq p_{(2)} \geq p_{(3)} \geq \dots \geq p_{(G)}$. If she answers only the first question, then her expected payment is

$$\frac{p_{(1)}}{\kappa - T}.$$

Let us now see her expected payment if she doesn't follow this answer pattern. The strong-no-free-lunch axiom implies that if the worker doesn't answer any question then her expected payment is zero. Suppose the worker chooses to answer questions $\{i_1, \dots, i_z\}$. In that case, her expected payment is

$$\kappa \frac{p_{i_1} \dots p_{i_z}}{T^z} = \kappa \frac{p_{i_1}}{T} \dots \frac{p_{i_z}}{T} \quad (40)$$

$$\leq \kappa \left(\frac{p_{(1)}}{T} \right)^z \quad (41)$$

$$\leq \kappa \frac{p_{(1)}}{T}, \quad (42)$$

where (42) uses the fact that $p_{(1)} < T$. The inequality in (42) becomes an equality only when $z = 1$. Now when $z = 1$, the inequality in (41) becomes an equality only when $i_1 = (1)$. Thus the unknowledgeable worker is incentivized to answer only one question – the one that she has the highest confidence in.

Now consider a knowledgeable worker and suppose her confidences are of the form $p_{(1)} \geq \dots \geq p_{(m)} > T > p_{(m+1)} \geq \dots \geq p_{(G)}$ for some $m \geq 1$. If the worker answers questions $(1), \dots, (m)$ as desired, her expected payment is

$$\kappa \frac{p_{(1)}}{T} \dots \frac{p_{(m)}}{T}.$$

Now let us see what happens if the worker does not follow this answer pattern. The strong-no-free-lunch axiom implies that if the worker doesn't answer any question then her expected payment is zero. Now, if she answers some other set of questions, say questions $\{i_1, \dots, i_z\}$ with $p_{(1)} \leq p_{i_1} < \dots < p_{i_y} \leq p_{(m)} < p_{i_{y+1}} < \dots < p_{i_z} \leq p_{(G)}$. The expected payment in that case is

$$\kappa \frac{p_{i_1} \dots p_{i_z}}{T^z} = \kappa \frac{p_{i_1}}{T} \dots \frac{p_{i_z}}{T} \quad (43)$$

$$\leq \kappa \frac{p_{i_1}}{T} \dots \frac{p_{i_y}}{T} \quad (44)$$

$$\leq \kappa \frac{p_{(1)}}{T} \dots \frac{p_{(m)}}{T}$$

where inequality (43) is a result of $\frac{p_{i_j}}{T} \leq 1 \ \forall j > y$ and holds with equality only when $y = z$. Inequality (44) is a result of $\frac{p_{i_j}}{T} \geq 1 \ \forall j \leq m$ and holds with equality only when $y = m$. Thus the expected payment is maximized when $i_1 = (1), \dots, i_z = (m)$ as desired. Finally, the payment strictly increases with an increase in the confidences, and hence the worker is incentivized to always consider the answer that she believes is most likely to be correct.

We now show that this mechanism is unique.

The necessary conditions derived in Lemma 4, when restricted to the case of $G = N$ and $(y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_G) \neq \{0\}^{N-1}$, is also applicable to the present setting. This is because the strong-no-free-lunch axiom assumed here is a stronger condition than the no-free-lunch axiom considered in Lemma 4, and moreover, $(y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_G) \neq \{0\}^{N-1}$ avoids the use of unknowledgeable workers in the proof of Lemma 4. It follows that for every question $i \in \{1, \dots, G\}$ and every $(y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_G) \in \{-1, 0, 1\}^{G-1} \setminus \{0\}^{G-1}$, it must be that

$$\begin{aligned} Tf(y_1, \dots, y_{i-1}, 1, y_{i+1}, \dots, y_G) + (1 - T)f(y_1, \dots, y_{i-1}, -1, y_{i+1}, \dots, y_G) \\ = f(y_1, \dots, y_{i-1}, 0, y_{i+1}, \dots, y_G). \end{aligned} \quad (45)$$

We claim that the payment must be zero whenever the number of incorrect answers $W > 0$. The proof proceeds by induction on the number of correct answers C . First suppose $C = 0$ (and $W > 0$). Then all questions are either wrong or skipped, and hence by the strong-no-free-lunch axiom, the payment must be zero. Now suppose the payment is necessarily zero whenever $W > 0$ and the total number of correct answers is $(C - 1)$ or lower, for some $C \in [G - 1]$. Consider any evaluation $(y_1, \dots, y_G) \in \{-1, 0, 1\}^G$ in which the number of incorrect answers is more than zero and the number of correct answers is C . Suppose $y_i = 1$ for some $i \in [G]$, and $y_j = -1$ for some $j \in [G] \setminus \{i\}$. Then from the induction hypothesis, we have $f(y_1, \dots, y_{i-1}, -1, y_{i+1}, \dots, y_G) = f(y_1, \dots, y_{i-1}, 0, y_{i+1}, \dots, y_G) = 0$. Applying (45) and noting that $T \in (0, 1)$, we get that $f(y_1, \dots, y_{i-1}, 1, y_{i+1}, \dots, y_G) = 0$ as claimed. This result also allows us to simplify (45) to: For every question $i \in \{1, \dots, G\}$ and every $(y_1, \dots, y_{i-1}, y_{i+1}, \dots, y_G) \in \{-1, 0, 1\}^{G-1} \setminus \{0\}^{G-1}$,

$$f(y_1, \dots, y_{i-1}, 1, y_{i+1}, \dots, y_G) = \frac{1}{T} f(y_1, \dots, y_{i-1}, 0, y_{i+1}, \dots, y_G). \quad (46)$$

We now show that when $C > 0$ and $W = 0$, the payment must necessarily be of the form described in the statement of Proposition 12. The proof again proceeds via an induction on the number of correct answers $C (\geq 1)$. Define a quantity $\kappa > 0$ as

$$\kappa = Tf(1, 0, \dots, 0). \quad (47)$$

Now consider the payment $f(1, y_2, \dots, y_G)$ for some $(y_2, \dots, y_G) \in \{0, 1\}^{G-1} \setminus \{0\}^{G-1}$ with C correct answers. Applying (46) repeatedly (once for every i such that $y_i = 1$), we get

$$f(1, y_2, \dots, y_G) = \frac{\kappa}{T^C}.$$

Unlike other results in this paper, at this point we cannot claim the result to hold for all permutations of the questions. This is because we have defined the quantity κ in an asymmetric manner (47), in terms of the payment function when the *first* question is correct and the rest are skipped. In what follows, we will prove that the result claimed in the statement of Proposition 12 indeed holds for all permutations of the questions.

From (46) we have

$$\begin{aligned} f(0, 1, 0, \dots, 0) &= Tf(1, 1, 0, \dots, 0) \\ &= f(1, 0, 0, \dots, 0) \\ &= \kappa. \end{aligned}$$

Thus the payment must be κ even if the second answer in the gold standard is correct and the rest are skipped. In fact, the argument holds when any one answer in the gold standard is correct and the rest are skipped. Thus the definition of κ is not restricted to the first question alone as originally defined in (47), but holds for all permutations of the questions. This allows the other arguments above to be applicable to any permutation of the questions. Finally, the budget constraint of μ_{\max} fixes the value of κ to that claimed, thereby completing the proof.

A.6.4 PROOF OF PROPOSITION 13

Proposition 12 proved that under the skip-based setting with the strong-no-free-lunch axiom, the payment must be zero when one or more answers are incorrect. This part of the proof of Proposition 12 holds even when $L > 1$. It follows that for any question, the penalty for an incorrect answer is the same for any confidence-level in $\{1, \dots, L\}$. Thus the worker is incentivized to always select that confidence-level for which the payment is the maximum when the answer is correct, irrespective of her own confidence about the question. This contradicts our requirements.

Appendix B. Details of Experiments

In this section, we provide further details about the experiments described earlier in Section 6.2. The experiments were carried out on the Amazon Mechanical Turk (mturk.com) online crowdsourcing platform in the time period June to October 2013. Figure 6 illustrates the interface shown to the workers for each of the experiments described in Section 6.2, while Figure 7 depicts the instructions given to the workers. The following are more details of each individual experiment. In the description, the notation κ is as defined in Algorithm 1 and Algorithm 2, namely, $\kappa = (\mu_{\max} - \mu_{\min})T^G$ for the skip-based setting and $\kappa = (\mu_{\max} - \mu_{\min}) \left(\frac{1}{\alpha_L}\right)^C$ for the confidence-based setting.

B.1 Recognizing the Golden Gate Bridge

A set of 21 photographs of bridges were shown to the workers, and for each photograph, they had to identify if it depicted the Golden Gate Bridge or not. An example of this task is depicted in Figure 6a, and the instructions provided to the worker under the three mechanisms are depicted in Figure 7. The fixed amount offered to workers was $\mu_{\min} = 3$ cents for the task, and the bonus was based on 3 gold standard questions. We compared (a) the baseline mechanism with 5 cents for each correct answer in the gold standard, (b) the skip-based mechanism with $\kappa = 5.9$ and $\frac{1}{\alpha} = 1.5$, and (c) the confidence-based mechanism with $\kappa = 5.9$ cents, $L = 2$, $\alpha_2 = 1.5$, $\alpha_1 = 1.4$, $\alpha_0 = 1$, $\alpha_{-1} = 0.5$, $\alpha_{-2} = 0$. The results of this experiment are presented in Figure 4a.

B.2 Transcribing Vehicles' License Plate Numbers from Photographs

This task presented the workers with 18 photographs of cars and asked them to transcribe the license plate numbers from each of them (source of photographs: <http://www.coolpic8z.com>). An example of this task is depicted in Figure 6b. The fixed amount offered to workers was $\mu_{\min} = 4$ cents for the task, and the bonus was based on 4 gold standard questions. We compared (a) the baseline mechanism with 10 cents for each correct answer in the gold standard, (b) the skip-based mechanism with $\kappa = 0.62$ and $\frac{1}{\alpha} = 3$, and (c) the confidence-based mechanism with $\kappa = 3.1$ cents, $L = 2$, $\alpha_2 = 2$, $\alpha_1 = 1.95$, $\alpha_0 = 1$, $\alpha_{-1} = 0.5$, $\alpha_{-2} = 0$. The results of this experiment are presented in

Figure 4b. When evaluating, in the worker's answers as well as in the true solutions, we converted all text to upper case, and removed all spaces and punctuations. We then declared a worker's answer to be in error if it did not have an exact match with the true solution.

B.3 Classifying Breeds of Dogs

This task required workers to identify the breeds of dogs shown in 85 images (source of images: Khosla et al. (2011); Deng et al. (2009)). For each image, the worker was given ten breeds to choose from. An example of this task is depicted in Figure 6c. The fixed amount offered to workers was $\mu_{\min} = 5$ cents for the task, and the bonus was based on 7 gold standard questions. We compared (a) the baseline mechanism with 8 cents for each correct answer in the gold standard, (b) the skip-based mechanism with $\kappa = 0.78$ and $\frac{1}{\alpha} = 2$, and (c) the confidence-based mechanism with $\kappa = 0.78$ cents, $L = 2$, $\alpha_2 = 2$, $\alpha_1 = 1.66$, $\alpha_0 = 1$, $\alpha_{-1} = 0.67$, $\alpha_{-2} = 0$. The results of this experiment are presented in Figure 4c.

B.4 Identifying Heads of Countries

Names of 20 personalities were provided and had to be classified as to whether they were ever the (a) President of the USA, (b) President of India, (c) Prime Minister of Canada, or (d) neither of these. An example of this task is depicted in Figure 6d. The fixed amount offered to workers was $\mu_{\min} = 2$ cents for the task, and the bonus was based on 4 gold standard questions. While the ground truth in most other multiple-choice experiments had approximately an equal representation from all classes, this experiment was heavily biased with one of the classes never being correct and another being correct for just 3 of the 20 questions. We compared (a) the baseline mechanism with 2.5 cents for each correct answer in the gold standard, (b) the skip-based mechanism with $\kappa = 0.25$ and $\frac{1}{\alpha} = 3$, and (c) the confidence-based mechanism with $\kappa = 1.3$ cents, $L = 2$, $\alpha_2 = 2$, $\alpha_1 = 1.95$, $\alpha_0 = 1$, $\alpha_{-1} = 0.5$, $\alpha_{-2} = 0$. The results of this experiment are presented in Figure 4d.

B.5 Identifying Flags

This was a relatively long task, with 126 questions. Each question required the workers to identify if a displayed flag belonged to a place in (a) Africa, (b) Asia/Oceania, (c) Europe, or (d) neither of these. An example of this task is depicted in Figure 6e. The fixed amount offered to workers was $\mu_{\min} = 4$ cents for the task, and the bonus was based on 8 gold standard questions. We compared (a) the baseline mechanism with 4 cents for each correct answer in the gold standard, (b) the skip-based mechanism with $\kappa = 0.2$ and $\frac{1}{\alpha} = 2$, and (c) the confidence-based mechanism with $\kappa = 0.2$ cents, $L = 2$, $\alpha_2 = 2$, $\alpha_1 = 1.66$, $\alpha_0 = 1$, $\alpha_{-1} = 0.67$, $\alpha_{-2} = 0$. The results of this experiment are presented in Figure 4e.

B.6 Distinguishing Textures

This task required the workers to identify the textures shown in 24 gray-scale images (source of images: Lazebnik et al. (2005; Data set 1: Textured surfaces)). For each image, the worker had to choose from 8 different options. Such a task has applications in computer vision, where it aids in recognition of objects or their surroundings. An example of this task is depicted in Figure 6f. The fixed amount offered to workers was $\mu_{\min} = 3$ cents for the task, and the bonus was based on 4 gold

standard questions. We compared (a) the baseline mechanism with 10 cents for each correct answer in the gold standard, (b) the skip-based mechanism with $\kappa = 3.1$ and $\frac{1}{\kappa} = 2$, and (c) the confidence-based mechanism with $\kappa = 3.1$ cents, $L = 2$, $\alpha_2 = 2$, $\alpha_1 = 1.66$, $\alpha_0 = 1$, $\alpha_{-1} = 0.67$, $\alpha_{-2} = 0$. The results of this experiment are presented in Figure 4f.

B.7 Transcribing Text from an Image: Film Certificate

The task showed an image containing 11 (short) lines of blurry text which the workers had to decipher. We used text from a certain certificate which movies releasing in India are provided. We slightly modified its text in order to prevent workers from searching a part of it online and obtaining the entire text by searching the first few transcribed lines on the Internet. An example of this task is depicted in Figure 6g. The fixed amount offered to workers was $\mu_{\min} = 5$ cents for the task, and the bonus was based on 2 gold standard questions. We compared (a) the baseline mechanism with 20 cents for each correct answer in the gold standard, (b) the skip-based mechanism with $\kappa = 5.5$ and $\frac{1}{\kappa} = 3$, and (c) the confidence-based mechanism with $\kappa = 12.5$ cents, $L = 2$, $\alpha_2 = 2$, $\alpha_1 = 1.95$, $\alpha_0 = 1$, $\alpha_{-1} = 0.5$, $\alpha_{-2} = 0$. The results of this experiment are presented in Figure 4g. When evaluating, in the worker’s answers as well as in the true solutions, we converted all text to upper case, and removed all spaces and punctuations. We then declared a worker’s answer to be in error if it did not have an exact match with the true solution.

B.8 Transcribing Text from an Image: Script of a Play

The task showed an image containing 12 (short) lines of blurry text which the workers had to decipher. We borrowed a paragraph from Shakespeare’s play ‘As You Like It.’ We slightly modified the text of the play in order to prevent workers from searching a part of it online and obtaining the entire text by searching the first few transcribed lines on the internet. An example of this task is depicted in Figure 6h. The fixed amount offered to workers was 5 cents for the task, and the bonus was based on 2 gold standard questions. We compared (a) the baseline mechanism with $\mu_{\min} = 20$ cents for each correct answer in the gold standard, (b) the skip-based mechanism with $\kappa = 5.5$ and $\frac{1}{\kappa} = 3$, and (c) the confidence-based mechanism with $\kappa = 12.5$ cents, $L = 2$, $\alpha_2 = 2$, $\alpha_1 = 1.95$, $\alpha_0 = 1$, $\alpha_{-1} = 0.5$, $\alpha_{-2} = 0$. The results of this experiment are presented in Figure 4h. When evaluating, in the worker’s answers as well as in the true solutions, we converted all text to upper case, and removed all spaces and punctuations. We then declared a worker’s answer to be in error if it did not have an exact match with the true solution.

B.9 Transcribing Text from Audio Clips

The workers were given 10 audio clips which they had to transcribe to text. Each audio clip was 3 to 6 seconds long, and comprised of a short sentence, e.g., “my favorite topics of conversation are sports, politics, and movies.” Each of the clips were recorded in different accents using a text-to-speech converter. An example of this task is depicted in Figure 6i. The fixed amount offered to workers was $\mu_{\min} = 5$ cents for the task, and the bonus was based on 2 gold standard questions. We compared (a) the baseline mechanism with 20 cents for each correct answer in the gold standard, (b) the skip-based mechanism with $\kappa = 5.5$ and $\frac{1}{\kappa} = 3$, and (c) the confidence-based mechanism with $\kappa = 12.5$ cents, $L = 2$, $\alpha_2 = 2$, $\alpha_1 = 1.95$, $\alpha_0 = 1$, $\alpha_{-1} = 0.5$, $\alpha_{-2} = 0$. The results of this experiment are presented in Figure 4i.

a Recognize the Golden Gate Bridge
 Golden Gate
 NOT Golden Gate
 Answer:

b Transcribe the license plate number
 Answer:

c Mark the breed of the dog
 Afghan Hound
 Doberman
 French Bulldog
 Tibetan Terrier

d Identify heads of countries
 Mohandas Gandhi
 President of the USA
 President of India
 Prime Minister of Canada
 None of these

e Mark the continent to which the flag belongs
 Africa
 Asia/Oceania
 Europe
 None of these

f Identify the texture
 Granite
 Carpet
 Fur
 Glass
 Corduroy
 Wood
 None of these

g Transcribe text (playscript)
 Line 1:
 Line 2:

h Transcribe text (certificate)
 Line 1:
 Line 2:

i Transcribe the audio clip
 Answer:

Figure 6: Various tasks on which the payment mechanisms were tested. The interfaces shown are that of the baseline mechanism, i.e., without the skipping or confidence choices.

a Baseline Mechanism

- *** Instructions for BONUS (Read Carefully) ***
- There are three questions whose answers are known to us, based on which the bonus is calculated
- BONUS (cents) = 5 * number of questions out of these that you correctly answer

b Skip-based multiplicative mechanism

- If you are not sure about any answer, then mark "I'm not sure"
 You need to mark at least something for every question, otherwise your work will be rejected
- *** Instructions for BONUS (Read Carefully) ***
 - You start with 5.9 cents of bonus for this HIT
 - There are three questions whose answers are known to us, based on which the bonus is calculated
 - For each of these questions you answer CORRECTLY, your bonus will INCREASE BY 50% (every 1 cent will become 1.5 cents)
 - If you answer any of these questions WRONG, your bonus will become ZERO
 - So for questions you are not sure of, mark the "I'm not sure" option; this does not affect the bonus

c Confidence-based multiplicative mechanism

- For each answer, you also need to indicate how sure you are about that answer
 If you are not sure about any answer, then mark "I don't know"
 You need to mark at least something for every question, otherwise your work will be rejected
- *** Instructions for BONUS (Read Carefully) ***
 - If an answer marked "absolutely sure" is wrong, your bonus will become ZERO for this entire HIT (you do not get any bonus for this HIT)
 - For every answer marked "absolutely sure" that is correct, your bonus will INCREASE BY 50% (every 1 cent will become 1.5 cents)
 - For every answer marked "moderately sure" that is correct, your bonus will be HALVED (every 1 cent will become half a cent)
 - For every answer marked "moderately sure" that is wrong, your bonus will be INCREASE BY 40% (every 1 cent will become 1.4 cents)
 - Marking "I don't know" for any answer does not change your bonus

Figure 7: An example of the instructions displayed to the worker under the three mechanisms.

Appendix C. General Utility Functions

In this section, we consider a setting where the worker, instead of maximizing her expected payment, aims to maximize the expected value of some *utility function* of her payment. Consider any function $U : \mathbb{R}_+ \rightarrow \mathcal{I}$, where \mathcal{I} is any interval on the real number line. We will require the function U to be strictly increasing and to have an inverse. Examples of such functions include $U(x) = \log(1+x)$ with $\mathcal{I} = \mathbb{R}_+$, $U(x) = \sqrt{x}$ with $\mathcal{I} = \mathbb{R}_+$, and $U(x) = 1 - e^{-x}$ with $\mathcal{I} = [0, 1]$. For any payment f made to the worker (based on the evaluation of her answers to the gold standard questions), her utility for this payment is $U(f)$. The worker aims to maximize the expected value of $U(f)$, where the expectation is with respect to her beliefs regarding correctness of her answers and the uniformly random distribution of the G gold standard questions among the set of N questions. The function U is assumed to be known to the worker as well as the system designer.

Consider the confidence-based setting of Section 4 (of which, the skip-based setting of Section 3 is a special case). Recall the notation $\{x_i\}_{i=1}^G$, $\{\alpha_j\}_{j=-L}^L$ and κ from Algorithm 2. Also recall the (generalized)-no-free-lunch axiom which mandates a zero payment if, in the gold standard, (all attempted questions are marked as the highest confidence L and) the answers to all the attempted questions are incorrect. The following proposition extends the results of the main text in the paper to this setting with utility functions.

Proposition 19 *For a worker who aims to maximize function U of the payment, the one and only mechanism that is incentive-compatible and satisfies the (generalized)-no-free-lunch axiom is*

$$\text{Payment}(x_1, \dots, x_G) = U^{-1} \left(\kappa \prod_{i=1}^G \alpha_{x_i} + U(\mu_{\min}) \right),$$

where the constants $\{\alpha_j\}_{j=-L}^L$ are as defined in Algorithm 2 and $\kappa = (U(\mu_{\max}) - U(\mu_{\min}))\alpha_L^{-G}$.

Note that for the problem to be well defined, the interval $[\mu_{\min}, \mu_{\max}]$ should be contained in the interval \mathcal{I} . The proof of Proposition 19 follows easily from the results proved earlier in the paper, and is provided below for completeness.

Proof of Proposition 19. We will first verify that the proposed payment is always non-negative and satisfies the (generalized)-no-free-lunch axiom. Recall from Theorem 7 that for every evaluation $\{x_1, \dots, x_G\}$ for which the (generalized)-no-free-lunch axiom mandates a zero payment, the value of $\kappa \prod_{i=1}^G \alpha_{x_i}$ is zero. It follows that the payment $U^{-1} \left(\kappa \prod_{i=1}^G \alpha_{x_i} + U(\mu_{\min}) \right) = U^{-1}(0 + U(\mu_{\min})) = \mu_{\min}$, where the final equation is a consequence of the invertibility of U . Further, recall that the value of $\kappa \prod_{i=1}^G \alpha_{x_i}$ in Algorithm 2 is never smaller than zero. Since the function U is increasing, so is U^{-1} , and hence the payment is always non-negative.

We will now prove that the proposed payment is incentive-compatible. To this end, observe that the utility of the proposed payment is

$$\begin{aligned} U(\text{Payment}) &= U \left(U^{-1} \left(\kappa \prod_{i=1}^G \alpha_{x_i} + U(\mu_{\min}) \right) \right) \\ &= \kappa \prod_{i=1}^G \alpha_{x_i} + U(\mu_{\min}). \end{aligned}$$

Noting that $U(0)$ is a constant independent of the worker's answers, the result of Theorem 7 implies that the expectation of $U(\text{Payment})$ behaves exactly as required for incentive-compatibility.

We will now prove uniqueness of this mechanism. Replacing $f(\cdot)$ by $U(\text{Payment}(\cdot))$ in the proof of Theorem 9, we get that the function $U(\text{Payment})$ must be of the form

$$U(\text{Payment}(x_1, \dots, x_G)) = c_1 \prod_{i=1}^G \alpha_{x_i} + c_2,$$

for some constants c_1 and c_2 , where $\{\alpha_{x_j}\}_{j=-L}^L$ are as defined in Algorithm 2. In other words, the payment must be of the form

$$\text{Payment}(x_1, \dots, x_G) = U^{-1} \left(c_1 \prod_{i=1}^G \alpha_{x_i} + c_2 \right).$$

One can evaluate that the maximum value of this payment is $c_1 + c_2$. From our μ_{\max} -budget constraint, we then have $c_1 + c_2 = \mu_{\max}$. Furthermore, when the evaluations x_1, \dots, x_G are such that the (generalized)-no-free-lunch applies, we need $\text{Payment} = \mu_{\min}$. It follows that $c_2 = U(\mu_{\min})$, and consequently $c_1 = U(\mu_{\max}) - U(\mu_{\min})$, thereby completing the proof.

Bibliography

- Nima Anari, Gagan Goel, and Afshin Nikzad. Mechanism design for crowdsourcing: An optimal 1-1/e competitive budget-feasible mechanism for large markets. In *Foundations of Computer Science (FOCS)*, pages 266–275, 2014.
- Dana Anglun and Philip Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.
- Jason Baldridge and Alexis Palmer. How well does active learning actually work?: Time-based evaluation of cost-reduction strategies for language documentation. In *Conference on Empirical Methods in Natural Language Processing*, pages 296–305, 2009.
- Michael S Bernstein, Greg Little, Robert C Miller, Björn Hartmann, Mark S Ackerman, David R Karger, David Crowell, and Katrina Panovich. Soylent: a word processor with a crowd inside. In *ACM symposium on User interface software and technology (UIST)*, pages 313–322, 2010.
- John Bohannon. Social science for pennies. *Science*, 334(6054):307–307, 2011.
- Glenn W Briar. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.
- Peter Bühlmann and Torsten Hothorn. Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science*, pages 477–505, 2007.
- Andreas Buja, Werner Stuetzle, and Yi Shen. Loss functions for binary class probability estimation and classification: Structure and applications. *Working draft, November*, 2005.
- Yang Cai, Constantinos Daskalakis, and Christos H Papadimitriou. Optimum statistical estimation with strategic data sources. In *Conference on Learning Theory (COLT)*, 2015.

- Izquierdo JM Cano, Yannis A Dimitriadis, Sánchez E Gómez, and Coronado J López. Learning from noisy information in fasnart and fasback neuro-fuzzy systems. *Neural networks: the official journal of the International Neural Network Society*, 14(4-5):407–425, 2001.
- Andrew Carlson, Justin Betteridge, Richard C Wang, Estevam R Hruschka Jr, and Tom M Mitchell. Coupled semi-supervised learning for information extraction. In *ACM international conference on Web search and data mining*, pages 101–110, 2010.
- Jenny J Chen, Natalia J Menezes, Adam D Bradley, and TA North. Opportunities for crowdsourcing research on Amazon mechanical turk. *Interfaces*, 5(3), 2011.
- Xi Chen, Paul N Bennett, Keyvn Collins-Thompson, and Eric Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *ACM international conference on Web search and data mining*, pages 193–202, 2013.
- Xi Chen, Sivakamth Gopi, Jieming Mao, and Jon Schneider. Competitive analysis of the top-K ranking problem. *arXiv preprint arXiv:1605.03933*, 2016.
- Fang Chu, Yizhou Wang, and Carlo Zaniolo. An adaptive learning approach for noisy data streams. In *IEEE International Conference on Data Mining (ICDM)*, pages 351–354, 2004.
- Vincent Conitzer. Prediction markets, mechanism design, and cooperative game theory. In *Uncertainty in Artificial Intelligence (UAI)*, pages 101–108, 2009.
- Anirban Dasgupta and Arpita Ghosh. Crowdsourced judgement elicitation with endogenous proficiency. In *International conference on World Wide Web (WWW)*, pages 319–330, 2013.
- Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied statistics*, pages 20–28, 1979.
- Ofer Dekel, Felix Fischer, and Ariel D Procaccia. Incentive compatible regression learning. In *ACM-SIAM symposium on Discrete algorithms (SODA)*, pages 884–893, 2008.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.
- Double or Nothing. http://wikipedia.org/wiki/Double_or_nothing, 2014. Last accessed: July 31, 2014.
- Fang Fang, Maxwell Stinchcombe, and Andrew Whinston. Putting your money where your mouth is: A betting platform for better prediction. *Review of Network Economics*, 6(2), 2007.
- Michael J Franklin, Donald Kossmann, Tim Kraska, Sukriti Ramesh, and Reynold Xin. CrowdDB: answering queries with crowdsourcing. In *ACM SIGMOD International Conference on Management of Data*, pages 61–72, 2011.
- Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

- Steve Hanneke and Liu Yang. Negative results for active learning with convex losses. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 321–325, 2010.
- Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.
- Chien-Ju Ho, Shahin Jabbari, and Jennifer W Vaughan. Adaptive task assignment for crowdsourced classification. In *International Conference on Machine Learning (ICML)*, pages 534–542, 2013.
- Panagiotis G Ipeirotis, Foster Provost, Victor S Sheng, and Jing Wang. Repeated labeling using multiple noisy labelers. *Data Mining and Knowledge Discovery*, 28(2):402–441, 2014.
- Srikanth Jagabathula, Lakshminarayanan Subramanian, and Ashwin Venkataraman. Reputation-based worker filtering in crowdsourcing. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2492–2500, 2014.
- W Paul Jones and Scott A Loe. Optimal number of questionnaire response categories more may not be better. *SAGE Open*, 3(2):2158244013489691, 2013.
- David R Karger, Sewoong Oh, and Devavrat Shah. Iterative learning for reliable crowdsourcing systems. In *Advances in neural information processing systems (NIPS)*, 2011.
- Gabriella Kazai, Jaap Kamps, Marijn Koolen, and Natasa Milic-Frayling. Crowdsourcing for book search evaluation: impact of HIT design on comparative system ranking. In *ACM SIGIR conference on Research and development in Information Retrieval*, pages 205–214, 2011.
- Firas Khatib, Frank DiMaio, Seth Cooper, Maciej Kazmierczyk, Mirosław Gilski, Szymon Krzywda, Helena Zabranska, Iva Pichova, James Thompson, Zoran Popović, Mariusz Jaskolski, and David Baker. Crystal structure of a monomeric retroviral protease solved by protein folding game players. *Nature structural & molecular biology*, 18(10):1175–1177, 2011.
- Ashish Khetan and Sewoong Oh. Reliable crowdsourcing under the generalized dawid-skene model. *arXiv preprint arXiv:1602.03481*, 2016.
- Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-fei Li. L.: Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, CVPR*, 2011.
- Nicolas Lambert and Yoav Shoham. Eliciting truthful answers to multiple-choice questions. In *ACM conference on Electronic commerce*, pages 109–118, 2009.
- ASID Lang and Joshua Rio-Ross. Using Amazon Mechanical Turk to transcribe historical handwritten documents. *The Code4Lib Journal*, 2011.
- Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1265–1278, 2005.

- John Le, Andy Edmonds, Vaughn Hester, and Lukas Biewald. Ensuring quality in crowdsourced search relevance evaluation: The effects of training question distribution. In *SIGIR 2010 workshop on crowdsourcing for search evaluation*, pages 21–26, 2010.
- Eric WM Lee, Chee Peng Lim, Richard KK Yuen, and SM Lo. A hybrid neural network model for noisy data regression. *IEEE Transactions on Systems, Man, and Cybernetics*, 34(2):951–960, 2004.
- Qiang Liu, Jian Peng, and Alexander T Ihler. Variational inference for crowdsourcing. In *Advances in Neural Information Processing Systems (NIPS)*, pages 701–709, 2012.
- Philip M Long and Rocco A Servedio. Random classification noise defeats all convex potential boosters. *Machine Learning*, 78(3):287–304, 2010.
- Naresh Manwani and PS Sasstry. Noise tolerance under risk minimization. *IEEE Transactions on Cybernetics*, 43(3):1146–1151, 2013.
- David Mease, Abraham J Wynner, and Andreas Buja. Boosted classification trees and class probability/quantile estimation. *The Journal of Machine Learning Research*, 8:409–439, 2007.
- George A Miller. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.
- Nolan Miller, Paul Resnick, and Richard Zeckhauser. Eliciting informative feedback: The peer-prediction method. *Management Science*, 51(9):1359–1373, 2005.
- Drazen Prelec. A Bayesian truth serum for subjective data. *Science*, 306(5695):462–466, 2004.
- Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *The Journal of Machine Learning Research (JMLR)*, 11:1297–1322, 2010.
- Leonard J Savage. Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association*, 66(336):783–801, 1971.
- Nihar B Shah and Martin J Wainwright. Simple, robust and optimal ranking from pairwise comparisons. *arXiv preprint arXiv:1512.08949*, 2015.
- Nihar B Shah, Sivaraman Balakrishnan, Joseph Bradley, Abhay Parekh, Kannan Ramchandran, and Martin J Wainwright. Estimation from pairwise comparisons: Sharp minimax bounds with topology dependence. In *Journal of Machine Learning Research (JMLR)*, 2016a.
- Nihar B Shah, Sivaraman Balakrishnan, Adityanand Guntuboyina, and Martin J Wainwright. Stochastically transitive models for pairwise comparisons: Statistical and computational issues. *International Conference on Machine Learning (ICML)*, 2016b.
- Nihar B Shah, Sivaraman Balakrishnan, and Martin J Wainwright. A permutation-based model for crowd labeling: Optimal estimation and robustness. *arXiv preprint arXiv:1606.09632*, 2016c.
- Richard M Shiffrin and Robert M Nosofsky. Seven plus or minus two: A commentary on capacity limitations. *Psychological Review*, 101(2):357–61, 1994.
- Aditya Vempaty, Law R Varshney, and Pramod K Varshney. Reliable crowdsourcing for multi-class labeling using coding theory. *IEEE Journal of Selected Topics in Signal Processing*, 8(4):667–679, 2014.
- Luis Von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. reCAPTCHA: Human-based character recognition via web security measures. *Science*, 321(5895):1465–1468, 2008.
- Jeroen Vaurens, Arjen P de Vries, and Carsten Eickhoff. How much spam can you take? An analysis of crowdsourcing results to increase accuracy. In *ACM SIGIR Workshop on Crowdsourcing for Information Retrieval*, pages 21–26, 2011.
- Paul Wais, Shivaram Lingamneni, Duncan Cook, Jason Fennell, Benjamin Goldenberg, Daniel Lubratov, David Marin, and Hari Simons. Towards building a high-quality workforce with Mechanical Turk. *NIPS workshop on computational social science and the wisdom of crowds*, 2010.
- Fabian L Wauthier and Michael Jordan. Bayesian bias mitigation for crowdsourcing. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.
- Justin Wolfers and Eric Zitzewitz. Prediction markets. Technical report, National Bureau of Economic Research, 2004.
- Man-Ching Yuen, Irwin King, and K'wong-Sak Leung. Task matching in crowdsourcing. In *IEEE International Conference on Cyber, Physical and Social Computing*, pages 409–412, 2011.
- Yuehen Zhang, Xi Chen, Dengyong Zhou, and Michael I Jordan. Spectral methods meet EM: A provably optimal algorithm for crowdsourcing. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.
- Dengyong Zhou, John Platt, Sumit Basu, and Yi Mao. Learning from the wisdom of crowds by minimax entropy. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2204–2212, 2012.
- Dengyong Zhou, Qiang Liu, John C Platt, Christopher Meek, and Nihar B Shah. Regularized minimax conditional entropy for crowdsourcing. *arXiv preprint arXiv:1503.07240*, 2015.
- Yuan Zhou, Xi Chen, and Jian Li. Optimal PAC multiple arm identification with applications to crowdsourcing. In *International Conference on Machine Learning (ICML)*, pages 217–225, 2014.

Joint Structural Estimation of Multiple Graphical Models

Jing Ma

Department of Biostatistics and Epidemiology
Perelman School of Medicine
University of Pennsylvania
211 Blockley Hall, 423 Guardian Drive
Philadelphia, PA 19104, USA

JINMA@UPENN.EDU

George Michailidis

Department of Statistics
University of Florida
205 Griffin-Floyd Hall, P.O. Box 118545
Gainesville, FL 32611, USA

GMICHAIL@UFL.EDU

Editor: Nicolai Meinshausen

Abstract

Gaussian graphical models capture dependence relationships between random variables through the pattern of nonzero elements in the corresponding inverse covariance matrices. To date, there has been a large body of literature on both computational methods and analytical results on the estimation of a *single* graphical model. However, in many application domains, one has to estimate several *related* graphical models, a problem that has also received attention in the literature. The available approaches usually assume that all graphical models are *globally* related. On the other hand, in many settings different relationships between subsets of the node sets exist between different graphical models. We develop methodology that *jointly* estimates multiple Gaussian graphical models, assuming that there exists prior information on how they are structurally related. For many applications, such information is available from external data sources. The proposed method consists of first applying neighborhood selection with a group lasso penalty to obtain edge sets of the graphs, and a maximum likelihood refit for estimating the nonzero entries in the inverse covariance matrices. We establish consistency of the proposed method for sparse high-dimensional Gaussian graphical models and examine its performance using simulation experiments. Applications to a climate data set and a breast cancer data set are also discussed.

Keywords: Gaussian graphical model, structured sparsity, group lasso penalty, consistency, edge set recovery

1. Introduction

There has been a large amount of work over the last few years on estimating Gaussian graphical models from high-dimensional data. In this family of models, jointly normally distributed random variables are represented by the nodes of a graph, while its edges reflect conditional dependence relationships amongst nodes that are captured through the nonzero entries of the inverse covariance matrix (or precision matrix) (Lauritzen, 1996; Edwards, 2000). Formally, let X be a p -dimensional multivariate normal random vector where

$$X = (X_1, \dots, X_p) \sim \mathcal{N}(\mu, \Sigma).$$

For $1 \leq i \neq j \leq p$, X_i and X_j are said to be conditionally independent given all the remaining variables, if the corresponding entry in the precision matrix $\Omega = \Sigma^{-1}$ is zero. An edge between the nodes X_i and X_j in the graph implies that they are conditionally dependent and corresponds to a nonzero entry in the precision matrix. To identify the graph, one only needs to select the corresponding precision matrix.

Bühlmann and van de Geer (2011, chap. 13) gave an overview of statistical methods developed for estimating a Gaussian graphical model subject to sparsity constraints, an attractive feature that reduces the number of parameters to be estimated and also enhances interpretability of the results. These models have found applications in diverse fields including analysis of omics data (Perraud et al., 2006; Pujana et al., 2007; Putluri et al., 2011), reconstruction of gene regulatory networks (Dehmer and Emmert-Streib, 2008, chap. 6), as well as study of climate networks (Zerener et al., 2014).

More recently, the focus has shifted from estimating a single graphical model to joint estimation of multiple graphs due to the availability of heterogeneous data (see discussion in Guo et al., 2011). For example, climate models capturing relationships between climate defining variables over a large area share common patterns; i.e. there are *shared common links* and also sharing of *absence of links* between the models (networks at different spatial locations). While separate estimation of individual models without taking the known pattern into consideration ignores the common structure, estimating one single model could mask the differences that could prove critical in understanding local climate features.

Several authors have studied the problem of *jointly* estimating multiple graphical models under different assumptions on how the models are related. Guo et al. (2011) introduced a procedure using a hierarchical penalty on the log-likelihood, whose objective is to estimate the common zeros (absence of edges) in the precision matrix across all graphical models under consideration. Thus, the procedure borrows strength across models through the non-connected nodes, but does not impose any structure on the connected ones. Danaher et al. (2014) proposed a joint graphical lasso by maximizing the log-likelihood subject to a generalized fused lasso or group lasso penalty, which can be solved efficiently by a standard alternating directions method of multipliers algorithm (Boyd et al., 2011). When employing a group lasso penalty, the underlying assumption is that the various observed graphical models are *perturbations* of a *single* common connectivity pattern across all graphical models, while when using a fused lasso across all models a similar outcome occurs, although more heterogeneity between estimated graphical models can be obtained depending on the tuning of the penalties. The work by Zhu et al. (2014) investigates the joint estimation problem by introducing a truncated ℓ_1 penalty on the pairwise differences between the precision matrices to achieve entry-wise clustering of the network structure over multiple graphs. Peterson et al. (2015) introduced a Bayesian approach that links the estimation of the graphs via a Markov random field prior for common structures. Further, a spike-and-slab prior is placed on the parameters that measure the similarity between graphs, thus relaxing the assumption on sharing a common structure across all graphical models.

Despite recent advances in joint estimation algorithms, theoretical properties of the resulting estimators have not been fully investigated. Guo et al. (2011) represents an exception, wherein asymptotic properties of the resulting estimator are established for consistent recovery of the common zeros across multiple precision matrices, which is the focus of that procedure. Zhu et al. (2014) focused mainly on efficient computational algorithms when the graphs have disjoint subgraphs, with a brief mention of consistency of precision matrices in a special temporal setting; however, no the-

oretical guarantees are provided for more general settings. Finally, many papers only present algorithms for joint estimation of the Gaussian graphical models under consideration, but no theoretical properties of the estimates (Honorio and Samaras, 2010; Chiquet et al., 2011; Danaher et al., 2014; Mohan et al., 2014).

In this paper, we investigate estimation of multiple graphical models under *complex structural relationships*, assuming that there exists *prior information* on their specification. In many applications, such information is available and may come from prior knowledge in the literature of relationships among different node subsets of the graphical models under consideration, or from clustering of all graphs. The approach allows sharing common sub-graph components between different models and does not require sharing of values for the same element across multiple precision matrices. The proposed method, called the *Joint Structural Estimation Method* (JSEM), leverages structured sparsity patterns as illustrated in Section 2 and is a two-step procedure. In the first step, we infer the sparse graphical models by incorporating the available structure through a group lasso penalty. In the second step, we maximize the Gaussian log-likelihood subject to the edge set constraints obtained from the previous step. Numerically, JSEM demonstrates superior performance in controlling both the number of false positive and false negative edges compared to available methods. When applied to joint modeling of climate networks, our results highlight the different roles climate defining factors play at different regions of the United States. In another application to breast cancer gene expression data, the JSEM methodology reveals interesting differences in the molecular network rewiring between the ER+ and ER- classes (see extensive discussion in Section 5.2). Understanding the rewiring of biological networks under different conditions provides deeper insights into biological mechanisms of disease, especially when combined with topology-based pathway enrichment methods as discussed and illustrated in Ma et al. (2016) and Kaushik et al. (2016).

The contributions of this work are three-fold. First, we develop a general framework for the problem of joint estimation of multiple Gaussian graphical models. The method can incorporate detailed structural information regarding relationships between subsets of the graphical models, while in the absence of such information reduces to the group graphical lasso procedure of Danaher et al. (2014). Further, we establish that the JSEM estimator is consistent with a fast rate of convergence in terms of the Frobenius norm for the estimated precision matrices. We also establish rigorously the consistent recovery of the edge sets for JSEM under suitable regularity conditions. Finally, when the externally provided structured sparsity pattern is moderately misspecified, we provide a modified estimator that reduces the number of false positive edges identified due to prior information misspecification, thus further enhancing the applicability of JSEM.

The paper is organized as follows. Section 2 discusses the structural relationships model used in this work and presents the estimation procedure. Section 3 presents the theoretical properties of the proposed method, followed by simulation studies in Section 4 and two real data applications—climate modeling and genomics of breast cancer—are presented in Section 5. We conclude with a discussion in Section 6. Most details of the theoretical analysis and proofs, additional simulation results as well as additional analyses on the applications are relegated to the Appendix.

2. The Joint Structural Estimation Method

Suppose we are interested in estimating K Gaussian graphical models from their corresponding K independent data sets, assuming that the models exhibit complex relationships between their edge sets. The data in the k -th model are organized in an $n_k \times p$ matrix $\mathbf{X}^k = (\mathbf{X}_1^k, \dots, \mathbf{X}_p^k)$, where each

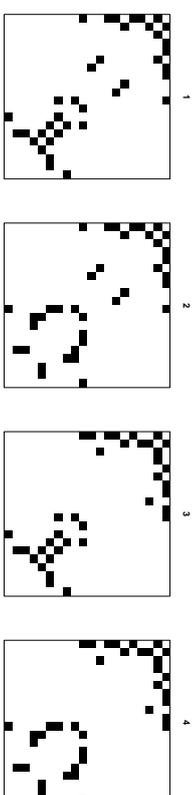


Figure 1: Image plots of the adjacency matrices for four graphical models with vertex set $\{1, \dots, p\}$. The black color represents presence of an edge. The structured sparsity pattern is encoded in $\mathcal{G} = \cup_{1 \leq i < j \leq p} \mathcal{G}^{ij}$, where $\mathcal{G}^{ij} = \{[1, 3], [2, 4]\}$ for $(i, j) \in \{[p/2] + 1, \dots, p\}$ and $\mathcal{G}^{ij} = \{[1, 2], [3, 4]\}$ for all other pairs of (i, j) .

row represents one observation from $\mathcal{N}(\mathbf{0}, \Sigma_0^k)$, $k = 1, \dots, K$. Throughout the remaining sections, we reserve the notations $\Sigma_0, \Omega_0, \dots$ to denote the population parameters in the true model and use Σ, Ω, \dots to denote generic parameters. Without loss of generality, we assume the columns of \mathbf{X}^k are centered and standardized to have mean zero and unit variance. For ease of presentation, it is assumed that the sample size $n_k = n$ for all $k = 1, \dots, K$, but the modeling framework can easily accommodate unequal sample sizes. Our goal is to estimate jointly $\Omega_0^k = (\Sigma_0^k)^{-1}$ for all k , under the assumption that the K corresponding graphs are related via a structured sparsity pattern \mathcal{G} . For example, consider climate models capturing relationships between climate forcing variables defined over a pre-specified spatial domain. Models that belong to the same climate zone may exhibit greater similarity in their graph structures than those from different zones. Thus, one can define \mathcal{G} based on their spatial locations. Figure 1 gives an illustration of the structured sparsity among four graphical models in terms of their adjacency matrices. This pattern indicates that sharing of structures may occur at different subsets of the edge set, which motivates us to develop a joint estimation method that can incorporate such rich and complex structural information.

2.1 Neighborhood Selection

Neighborhood selection was introduced by Meinshausen and Bühlmann (2006) as an efficient method to construct Gaussian graphical models from high-dimensional data. For each node $i = 1, \dots, p$ in the graphical model, consider the optimal prediction of the random variable X_i as a linear combination of the remaining variables:

$$X_i = \sum_{j \neq i} \theta_{ij} X_j + \varepsilon_i$$

where θ_{ij} ($j \neq i$) are the regression coefficients and $\varepsilon_i \perp \{X_j : j \neq i\}$. The matrix $(\theta_{ij})_{1 \leq i, j \leq p}$ is determined by the inverse covariance matrix $\Omega = (\omega_{ij})_{1 \leq i, j \leq p}$. Specifically, it holds that $\theta_{ij} = -\omega_{ij}/\omega_{ii}$, for all $j \neq i$. The set of nonzero coefficients of θ_{ij} ($j \neq i$) is thus the same as the set of nonzero entries in the row vector of ω_{ij} ($j \neq i$), which defines the set of neighbors of node i . Using an l_1 -penalized regression, Meinshausen and Bühlmann (2006) estimated the neighborhood for each node and combined the estimates to obtain the underlying graph.

2.2 An Illustrative Example

We first illustrate how to extend the idea of neighborhood selection to multiple graphical models using the example in Figure 1. For $k = 1, \dots, K$, let $(\theta_{ij}^k)_{p \times p}$ be the matrix of regression coefficients in graph k and θ_i^k the vector of all θ_{ij}^k ($j \neq i$) for node $i = 1, \dots, p$. Unless otherwise stated, all vectors are assumed to be column vectors. For node i in a single graph k , neighborhood selection suggests estimating the coefficients θ_i^k by

$$\min_{\theta_i^k} \frac{1}{n} \|\mathbf{X}_i^k - \mathbf{X}_{-i}^k \theta_i^k\|^2 + 2\lambda \sum_{j \neq i} |\theta_{ij}^k|, \quad (1)$$

where \mathbf{X}_{-i}^k is \mathbf{X}^k with the i -th column removed, $\|\cdot\|$ represents the standard Euclidean norm and λ is the regularization parameter. To achieve joint estimation, consider the following regularized regression problem

$$\min_{\Theta_i} \frac{1}{n} \sum_{k=1}^K \|\mathbf{X}_i^k - \mathbf{X}_{-i}^k \theta_i^k\|^2 + 2P_\lambda(\Theta_i), \quad (1)$$

where $K = 4$, $\Theta_i = (\theta_{i1}^1, \dots, \theta_{i4}^4)$ and $P_\lambda(\Theta_i)$ is a regularization term to be determined next. Note that each column of Θ_i represents the regression coefficients from one graphical model and each row of Θ_i corresponds to the four coefficients at the same (i, j) pair.

The penalty $P_\lambda(\Theta_i)$ is chosen based on information from the structured sparsity pattern \mathcal{G} in Figure 1. For example, for $i = 1$ with grouping $\{[1, 2], [3, 4]\}$,

$$\Theta_1 = \begin{pmatrix} \theta_{12}^1 & \theta_{12}^2 & \theta_{12}^3 & \theta_{12}^4 \\ \vdots & \vdots & \vdots & \vdots \\ \theta_{1p}^1 & \theta_{1p}^2 & \theta_{1p}^3 & \theta_{1p}^4 \end{pmatrix}.$$

As indicated by the colors, we can then group the coefficients in the j -th row of Θ_1 ($j = 2, \dots, p$) as

$$\underbrace{(\theta_{1j}^1, \theta_{1j}^2, \theta_{1j}^3, \theta_{1j}^4)}_{\theta_{1j}^{\{1,2\}}} \underbrace{(\theta_{1j}^3, \theta_{1j}^4)}_{\theta_{1j}^{\{3,4\}}}$$

and set $P_\lambda(\Theta_1)$ to be the group lasso penalty

$$\sum_{j=2, \dots, p} \sum_{g=\{1,2\}, \{3,4\}} \lambda_{ij}^g \|\theta_{1j}^g\|.$$

The group lasso penalty forces the two coefficients in each group to be zero or nonzero at the same time, leading to the same structure for graphical models belonging to the same group.

The solution Θ_1 to (1) for $i = 1, \dots, p$ can then be used for graph selection.

2.3 The General Case

Denote the structured sparsity pattern by $\mathcal{G} = \cup_{1 \leq i < j \leq p} \mathcal{G}^{ij}$, where the union is over all $p(p-1)/2$ pairs of potential edges. Each \mathcal{G}^{ij} is a partition of the set $\{1, 2, \dots, K\}$ and consists of prior

knowledge on the structural similarity for the (i, j) -th pair across models. For example in Figure 1, $\mathcal{G}^{12} = \{[1, 2], [3, 4]\}$ means that the graphs 1 and 2 exhibit the same structure at (i, j) , whereas 3 and 4 behave the same at (i, j) . It is possible for all four graphs to have the edge (i, j) or not have the edge (i, j) at the same time, but we do not impose this restriction. Taking the union over all pairs, $\mathcal{G} = \{[1, 2], [3, 4], [1, 3], [2, 4]\}$ in Figure 1. Therefore the pattern \mathcal{G} allows a more flexible structural relationships among multiple graphical models. Further, the sparsity pattern in \mathcal{G} is symmetric as we require $\mathcal{G}^{ji} = \mathcal{G}^{ij}$ for $i < j$.

For $1 \leq i < j \leq p$ and a group $g \in \mathcal{G}^{ij}$, denote by θ_{ij}^g the vector $(\theta_{ij}^k)_{k \in g}$, a concatenation of all regression coefficients from graphs in g . The grouping for the regression coefficients $(\theta_{ij}^1, \dots, \theta_{ij}^K)$ is determined by \mathcal{G}^{ij} . Under correctly specified \mathcal{G} , all coefficients in the same group should be zero or nonzero simultaneously. For $k = 1, \dots, K$, let $E^k = \{(i, j) : \theta_{ij}^k \neq 0\}$ be the set of undirected edges in graph k and $E_{E^k}^+ = \{\Omega : \Omega \succ 0 \text{ and } \omega_{ij} = 0 \text{ for all } (i, j) \notin E^k \text{ where } i \neq j\}$.

The *Joint Structural Estimation Method* (JSEM) proceeds with the following two steps.

- (I) For $k = 1, \dots, K$, we infer the sparse graphs \hat{E}^k through the following group lasso estimator.
For $i = 1, \dots, p$,

$$\min_{\Theta_i} \left\{ \frac{1}{n} \sum_{k=1}^K \|\mathbf{X}_i^k - \mathbf{X}_{-i}^k \theta_i^k\|^2 + 2 \sum_{j \neq i, g \in \mathcal{G}^{ij}} \lambda_{ij}^g \|\theta_{ij}^g\| \right\}. \quad (2)$$

\hat{E}^k is estimated to be the set

$$\{(i, j) : 1 \leq i < j \leq p, \hat{\theta}_{ij}^k \neq 0 \text{ OR } \hat{\theta}_{ji}^k \neq 0\}. \quad (3)$$

- (II) We refit the model by

$$\min_{\Omega^k \in S_{E^k}^+} \left\{ \text{tr}(\hat{\Sigma}^k \Omega^k) - \log \det(\Omega^k) \right\}, \quad k = 1, \dots, K. \quad (4)$$

Note the grouped variables in (2) are non-overlapping because \mathcal{G}^{ij} partitions the set $\{1, \dots, K\}$ into disjoint subsets. The ‘OR’ rule defined in (3) can be replaced by the ‘AND’ rule. The problems in (2) and (4) are both convex and can thus be solved by available convex optimization algorithms. In this work, we use the R-package `gprreg` (Breheny and Huang, 2009) for implementation of the group lasso penalized optimization (2) and the `glasso` (Friedman et al., 2008) one for solving (4). The computational complexity for step (II) is $O(Kp^3)$ using the standard graphical lasso algorithm. Since `gprreg` uses a coordinate descent algorithm, the computational complexity for step (I) can be as fast as $O(nKp^2)$ if the number of graphs K does not exceed the sample size n , or $O(K^2p^2)$ otherwise. Thus, the overall computational complexity of JSEM is $O(Kp^3)$ if $p > K$, and $O(K^2p^2)$ otherwise.

2.4 Choice of Tuning Parameters

Like any other penalty-based method, JSEM requires selection of the tuning parameters λ_{ij}^g for all p regressions in (2). One can customize λ_{ij}^g for each 3-tuple (i, j, g) based on prior knowledge on graph similarity or simply use the same λ for all 3-tuples (i, j, g) . In the sequel, we present results

based on the latter approach. We recommend choosing the tuning parameters via the Bayesian information criterion (BIC). Specifically, for a given λ , we define BIC for the proposed method as

$$\text{BIC}(\lambda) = \sum_{k=1}^K \left\{ \text{tr}(\hat{\Sigma}^k \hat{Q}_i^k) - \log \det(\hat{Q}_i^k) + \frac{\log(n_k)}{n_k} |\hat{E}^k| \right\},$$

where \hat{Q}_i^k ($k = 1, \dots, K$) are the estimated precision matrices from the data. The optimal tuning parameter is thus $\lambda^* = \arg \min_{\lambda \in \mathcal{D}_n} \text{BIC}(\lambda)$, where the set of values \mathcal{D}_n is chosen such that for every $\lambda_j \in \mathcal{D}_n$ ($n_k = n$):

$$\lambda_j = c_j \left(|g_{\max}| + \sqrt{\log G_0} \right) / \sqrt{n}, \quad c_j = 0.02 * j, \quad j = 1, \dots, 20.$$

Here $|g_{\max}|$ and G_0 refer, respectively, to the maximum size of groups in \mathcal{G} and maximum total number of groups in all regressions. They can be conveniently defined by the input sparsity pattern. In practice, it is also recommended to apply the stability selection procedure (Meinshausen and Bühlmann, 2010; Shah and Samworth, 2013) to select graphical models that are both stable and interpretable.

3. Theoretical Results

The JSEM estimator enjoys nice theoretical properties under certain regularity conditions. Specifically, we establish the norm consistency of the estimated precision matrices, as well as the consistent recovery of the edge sets of the various graphical models under consideration based on the structured sparsity pattern \mathcal{G} .

3.1 Estimation Consistency

Let $\mathbb{N}_{(p-1)K}^i = \{(j, k) : j \neq i, k = 1, \dots, K\}$ be the variable index set for equation (2) with a fixed node i . Given the structural information \mathcal{G} , the grouped variable index set $\{(j, g) : j \neq i, g \in \mathcal{G}^{ij}\}$ defines a partition of $\mathbb{N}_{(p-1)K}^i$. Denote by G_i the cardinality of the set $\{(j, g) : j \neq i, g \in \mathcal{G}^{ij}\}$. Then $1 \leq G_i \leq (p-1)K$. Let $J(\Theta_{0,i}) = \{(j, g) : j \neq i, g \in \mathcal{G}^{ij}, \theta_{0,i}^{[jg]} \neq 0\}$ be the set of nonzero groups in the i -th regression. We assume an overall sparsity at the group level, that is, the size of $J(\Theta_{0,i})$ is $s_i \ll G_i$. Let

$$G_0 = \max_{i=1, \dots, p} G_i, \quad s_0 = \max_{i=1, \dots, p} s_i, \quad S_0 = \sum_{i=1}^p s_i,$$

and also let $|g|$ be the size of the group g with $|g_{\max}| = \max_{g \in \mathcal{G}} |g|$.

Let $\mathbb{M}(p, K)$ represent the set of all $p \times K$ matrices. For $\Delta = (\delta^1, \dots, \delta^K) \in \mathbb{M}(p, K)$ and a group $g \subset \{1, \dots, K\}$, denote by $\delta_j^{[g]}$ the vector composed of all δ_j^k for which $k \in g$. Write $\mathcal{J} = \{J(\Theta_{0,1}), \dots, J(\Theta_{0,p})\}$, the collection of sets of nonzero groups in all p regressions. For any $J \in \mathcal{J}$, denote Δ_J the nonzero matrix in $\mathbb{M}(p, K)$, which has the same coordinates as Δ on J and zero elsewhere. Let J^c denote the complement of the index set J . Write $\underline{0}$ as the zero matrix in $\mathbb{M}(p, K)$. We make the following assumptions.

$$(A1) \quad \text{For } 0 < s < G_0, \text{ there exists } \kappa = \kappa(s) > 0, \text{ such that}$$

$$\min_{J \in \mathcal{J}, |J| \leq s} \min_{\Delta \in \mathcal{F}_J} \frac{\sum_{k=1}^K \|\mathbf{X}^k \delta^k\|^2 / n}{\|\Delta_J\|^2} \geq \kappa^2(s),$$

where for i satisfying $J(\Theta_{0,i}) = J$, \mathcal{F}_J is defined as

$$\mathcal{F}_J = \{\Delta : \Delta \in \mathbb{M}(p, K) \setminus \{0\}, \sum_{(j,g) \in J^c} \lambda_{ij}^g \|\delta_j^{[g]}\| \leq 3 \sum_{(j,g) \in J} \lambda_{ij}^g \|\delta_j^{[g]}\|\}.$$

$$(A2) \quad \text{For every } k = 1, \dots, K \text{ and } i = 1, \dots, p, \text{Var}(X_i^k) = 1. \text{ Further, there exist constants } c_0 \text{ and } d_0 \text{ such that for every } k,$$

$$0 < 1/c_0 \leq \phi_{\min}(\Sigma_0^k) \leq \phi_{\max}(\Sigma_0^k) \leq 1/d_0 < \infty,$$

where $\phi_{\min}(\Sigma_0^k)$ and $\phi_{\max}(\Sigma_0^k)$ are the minimum and maximum eigenvalues of the matrix Σ_0^k , respectively.

Assumption (A1) is a generalization of the Restricted Eigenvalue assumption for the Lasso in Bickel et al. (2009) to the group lasso setting in our problem and requires the super design matrix $\text{diag}(\mathbf{X}^1, \dots, \mathbf{X}^K)$ to be well conditioned over the restricted set of vectors under consideration. One sufficient condition is that the eigenvalues of the Gram matrix of $\text{diag}(\mathbf{X}^1, \dots, \mathbf{X}^K)$ is positive when restricted to the subset of sparse vectors with cardinality no greater than $2s$.

The equal variance requirement in assumption (A2) can be easily achieved by appropriate scaling of the data. The second part of the assumption explicitly excludes singular or nearly singular covariance matrices and guarantees that Ω_0^k exists for every model $k = 1, \dots, K$. We are now ready to state our first result.

Theorem 1 Consider $\hat{\Omega}^k$ ($k = 1, \dots, K$) defined in (4). Let Assumption (A1) with $s = 2s_0$ and Assumption (A2) be satisfied. For every regression defined in (2), choose

$$\chi_{ij}^g = \frac{2}{\sqrt{nd_0}} \left(\sqrt{|g_{\max}|} + \frac{\pi}{\sqrt{2}} \sqrt{q \log G_0} \right),$$

with $q > 1$. Then, with probability at least $1 - 2pG_0^{1-q}$, we have

$$\frac{1}{K} \sum_{k=1}^K \|\hat{\Omega}^k - \Omega_0^k\|_F \leq O \left(\sqrt{\frac{S_0}{nK}} \left\{ \sqrt{|g_{\max}|} + \frac{\pi}{\sqrt{2}} \sqrt{q \log G_0} \right\} \right), \quad (5)$$

where G_0 is the maximum number of groups in all regressions, S_0 is the total number of relevant groups and $|g_{\max}|$ is the maximum group size.

Proof of Theorem 1 is available in Appendix A. Note the rate in (5) improves over estimating each precision matrix separately, as long as the sparsity pattern \mathcal{G} is appropriately specified and nontrivial, i.e. there exists structural similarity among the considered graphical models. Further, the proposed procedure obtains a faster convergence rate than that of Guo et al. (2011) in some scenarios.

For example, if all K graphs share the same structure, then $|g_{\max}| = K$ and $G_0 = p - 1$. Thus, JSEM achieves a convergence rate of the order of

$$O\left(\sqrt{\frac{S_0}{n}} \left\{ 1 + \frac{\pi}{\sqrt{2}} \sqrt{\frac{q \log(p-1)}{K}} \right\}\right). \quad (6)$$

In contrast, separate estimation of Ω^k is known to be of the order of

$$O\left(\sqrt{\sum_k \|\Omega_0^{k-}\|_0 \frac{\log p}{nK}}\right),$$

where $\|\Omega_0^{k-}\|_0$ denotes the number of nonzero off-diagonal entries in Ω_0^k and \sum_k is short-hand notation for $\sum_{k=1}^K$. The joint estimation method by Guo et al. (2011) has the following convergence rate

$$O\left(\sqrt{(p+m) \frac{\log p}{nK}}\right),$$

where $m = |\cup \{k = 1, \dots, K : \omega_{0,ij}^k \neq 0\}|$. Under correctly specified \mathcal{G} , we have $S_0 = m$. Thus, JSEM has a lower estimation error rate than the joint estimation method of Guo et al. (2011). JSEM also outperforms separate estimation if $S_0 \asymp \|\Omega_0^{k-}\|_0$, where \asymp means that the expressions on both sides are of the same order. On the other hand, the rate in (6) could be worse if the sparsity pattern \mathcal{G} is highly misspecified such that the number of nonzero parameters $S_0 > \sum_k \|\Omega_0^{k-}\|_0 \geq m$. The issue of sparsity pattern misspecification is addressed in the next section.

3.2 Graph Selection Consistency

To understand how JSEM performs in selecting the edge sets of the graphical models, it suffices to focus on each of the group lasso estimation problems (2), as consistent graph selection relies on consistent variable selection in all p regressions. Unlike the sign consistency in the lasso setting (Zhao and Yu, 2006), variable selection properties with a group lasso penalty are much more complicated because the latter selects whole groups rather than individual variables (see Basu et al., 2015, and the discussion therein). The Basu et al. (2015) paper offers a generalization and introduces the notion of direction consistency for the group lasso. Specifically, for a nonzero vector ξ , its direction vector is defined as $D(\xi) = \xi/\|\xi\|$ and $D(\mathbf{0}) = \mathbf{0}$. An estimator $\hat{\Theta}_i$ of (2) is *direction consistent* at rate α_n if for a sequence of positive real numbers $\alpha_n \rightarrow 0$,

$$\mathbb{P}(\|D(\hat{\Theta}_{0,ij}^{[g]}) - D(\theta_{0,ij}^{[g]})\| < \alpha_n, \forall (j,g) \in J(\Theta_{0,i}); \hat{\theta}_{ij}^{[g]} = \mathbf{0}, \forall (j,g) \notin J(\Theta_{0,i})) \rightarrow 1,$$

as $n, p \rightarrow \infty$. In general, direction consistency does not guarantee sign consistency, especially when there are multiple members within one group. However, if the group is selected, all the members within the group are selected, which is sufficient for joint neighborhood selection for each node and subsequent selection of graphs. Motivated by the above idea, we establish the graph selection consistency property of JSEM in Theorem 2, which can be conveniently modified to adjust for the misspecification in the prior information \mathcal{G} . Before we present the main result, we need more notations.

Consider the group lasso estimation problem (2) for node i . For simplicity, we discuss the estimation consistency properties with a common tuning parameter λ for all (j,g) . For $k = 1, \dots, K$,

denote \mathbf{X}_k^k the $n \times |I_k|$ sub-matrix consisting of all relevant variables from the k -th model. In other words, for all $j \in I_k$, there exists a group $g \ni k$ such that $(j,g) \in J(\Theta_{0,i})$. Note the dependency of each index set I_k on i is made implicit here for notational convenience. Further, let $\xi^k \in \mathbb{R}^{|I_k|}$ be a vector indexed by I_k . The following assumption adapts the *Uniform Irrepresentability Condition (IC)* in Basu et al. (2015) to our setting:

(A3) There exists a positive constant η such that for all $\xi = ((\xi^1)^T, \dots, (\xi^K)^T)^T \in \mathbb{R}^{\sum_k |I_k|}$ with $\max_{(j,g)} \|\xi_j^{[g]}\| \leq 1$ and all $(j,g) \notin J(\Theta_{0,i})$,

$$\left(\sum_{k \in G} [\mathbf{X}_j^k]^T \mathbf{X}_{I_k}^k \{(\mathbf{X}_{I_k}^k)^T \mathbf{X}_k^k\}^{-1} \xi^k \right]^{1/2} \leq 1 - \eta. \quad (7)$$

Note the group level constraint (7) is required to hold for all p regressions and is less stringent than the IC for the selection consistency of lasso. In general, it is not easy to verify Assumption (A3). One sufficient condition, as suggested in Zhao and Yu (2006), is that the regression coefficients of \mathbf{X}_j^k on \mathbf{X}_k^k ($k = 1, \dots, K$) have Euclidean norm less than 1 for all $(j,g) \notin J(\Theta_{0,i})$.

Theorem 2 Let Assumption (A1) with $s = s_0$, (A2) and (A3) be satisfied. Assume further that the sparsity pattern \mathcal{G} is correctly specified. For every regression defined in (2), choose

$$\lambda \geq \max_{i,(j,g) \notin J(\Theta_{0,i})} \frac{1}{\eta} \frac{1}{\sqrt{nd_0}} \left(\sqrt{|g|} + \frac{\pi}{\sqrt{2}} \sqrt{q \log G_0} \right), \quad (8)$$

$$\alpha_n \geq \max_{i,(j,g) \in J(\Theta_{0,i})} \frac{1}{\kappa(s_0)} \frac{1}{\|\theta_{0,ij}^{[g]}\|} \left\{ \lambda \frac{\sqrt{s_0}}{\kappa(s_0)} + \frac{1}{\sqrt{nd_0}} \left(\sqrt{|g|} + \frac{\pi}{\sqrt{2}} \sqrt{q \log G_0} \right) \right\}, \quad (9)$$

with $q > 1$. Then with probability at least $1 - 4pG_0^{1-q}$, we have simultaneously for all i

1. $\hat{\theta}_{ij}^{[g]} = \mathbf{0}$, for all $(j,g) \notin J(\Theta_{0,i})$,
2. $\|\hat{\theta}_{ij}^{[g]} - \theta_{0,ij}^{[g]}\| < \alpha_n \|\theta_{0,ij}^{[g]}\|$, and hence $\|D(\hat{\Theta}_{0,ij}^{[g]}) - D(\theta_{0,ij}^{[g]})\| < 2\alpha_n$, for all $(j,g) \in J(\Theta_{0,i})$.

Further, if $\alpha_n < 1$, then

$$\mathbb{P}(\hat{E}^k = E_0^k, \forall k = 1, \dots, K) \geq 1 - 4pG_0^{1-q},$$

where \hat{E}^k is defined in (3).

Note the choice of λ in (8) is of the same order as the tuning parameter required for estimation consistency in Theorem 1. With the above choice of λ , α_n can be chosen to be of the order of $O(\sqrt{s_0}(\sqrt{|g_{\max}|} + \sqrt{\log G_0})/\sqrt{\eta})$. A proof of Theorem 2 can be found in Appendix B.

Bach (2008) using a strong irrepresentability assumption also establishes group support recovery. In this work, we take a different route, where a similar strong irrepresentability assumption leads to direction consistency. Then, we leverage the notion of direction consistency to propose *within group thresholding* which allows us to handle successfully moderate misspecification of the group structures, as discussed next. Further, from a technical perspective, we build on the Karush-Kuhn-Tucker (KKT) conditions inversion scheme introduced in Zhao and Yu (2006), and noting

that the $\text{sign}(\cdot)$ function in standard lasso KKT conditions is replaced by the $D(\cdot)$ function in the group lasso KKT conditions. Therefore, sign consistency has a natural generalized counterpart when considering optimization over groups.

When \mathcal{G} is misspecified, it is possible that not all the members within a group have nonzero effects. However, the group lasso penalty may fail to exclude members with actual zero effect within the misspecified group, leading to the recovery of spurious edges. The following result implies that the property of direction consistency helps identify influential members within a group, that is, those with noticeable nonzero effects.

Corollary 3 *Let Assumption (A1) with $s = s_0$, (A2) and (A3) be satisfied. For every regression defined in (2), choose λ and α_n as in Theorem 2. Define*

$$\hat{\theta}_{ij}^{k,thr} = \hat{\theta}_{ij}^k \mathbf{1}\{\|\hat{\theta}_{ij}^k\| / \|\hat{\theta}_{ij}^{[0]}\| > 2\alpha_n\}, \quad \forall k \in g, \forall (i, j) \in J(\Theta_{0,i}),$$

and

$$\hat{E}^{k,thr} = \{(i, j) : 1 \leq i < j \leq p, \hat{\theta}_{ij}^{k,thr} \neq 0 \text{ OR } \hat{\theta}_{ji}^{k,thr} \neq 0\},$$

If for all $g \in \mathcal{G}$, $\min_{k \in g} \hat{\theta}_{i,j}^k / \|\hat{\theta}_{0,i,j}^{[0]}\| > 2\alpha_n$, then

$$\mathbb{P}(\hat{E}^{k,thr} = E_0^k, \forall k = 1, \dots, K) \geq 1 - 4pG_0^{1-q}.$$

The result in Corollary 3 implies immediately that JSEM with an additional thresholding step on the estimated direction vectors $D(\|\hat{\theta}_{ij}^{[0]}\|)$ can be applied to reduce false discoveries and thus improve selection of the edge sets when the structured pattern \mathcal{G} is moderately misspecified (that is, most of the structural relationships specified in \mathcal{G} are reliable). This is illustrated in the third simulation study of Section 4.

4. Performance Evaluation

We present three simulation studies to evaluate the performance of JSEM. Other methods compared include the separate estimation method Glasso, where the *Graphical Lasso* by Friedman et al. (2008) is applied to each graphical model separately, joint estimation by Guo et al. (2011), denoted by JEM-G, the Group Graphical Lasso denoted by GGL by Danaher et al. (2014), and the structural pursuit method MGGM by Zhu et al. (2014). Note we choose MGGM over the Fused Graphical Lasso method (Danaher et al., 2014), as the former has been consistently shown to exhibit better performance.

The first study considers a *single common structure* across all graphical models, while the second one features a *more complex structured sparsity pattern*. Our comparisons are based on the overall performance of different methods in terms of their ROC curves, as well as their finite sample performance in identifying the corresponding graphical models. For the latter, we use BIC to select the tuning parameters for all methods; in addition, the maximum likelihood refitting step (4) is added to all joint estimation methods to ensure fair comparison. We point out that the first study is favorable to existing joint estimation methods due to high degree of structural similarity, while the second one with varying degrees of structural similarity is more favorable to the JSEM procedure. Nevertheless, the results show that JSEM outperforms these competing methods in both settings, even when the structured pattern is moderately misspecified.

The third simulation compares JSEM with its thresholded version under misspecified \mathcal{G} using the experimental settings of the first two studies. In this setting, one also needs to select the within group thresholding α_n besides λ . As in previous simulations, we first select λ via BIC without any thresholding. At the optimal λ , we select α_n from the grid of values

$$\alpha_n(c) = c \left(|g_{\max}| + \sqrt{\log G_0} \right) / \sqrt{\pi}, \quad c \in \{0.1, 0.2, \dots, 1\},$$

where $|g_{\max}|$ and G_0 are defined by the input sparsity pattern. The optimal α_n^* is selected as the one that minimizes the corresponding BIC.

We refer readers to Appendix C for additional simulation results, including comparison of all joint estimation methods with and without maximum likelihood refitting step (4), and large p settings.

4.1 Simulation Study 1

In our first simulation, we set $K = 5$, with each graphical model being of size $p = 100$. The structured pattern is constructed as follows: we first generate a scale-free network with edge set E_0 as the common structure shared across all graphs, shown in the left panel of Figure 2. To generate the edge set E^k , we randomly select a pair of (i, j) , $i < j$ such that $(i, j) \notin E_0$ and add it to E^k . This procedure was repeated $\rho|E_0|$ times for each k , where ρ is a positive number corresponding to the ratio of individual edges to common ones. In this example, we set $\rho = 0.1$ to allow high structural similarity across graphs. Thus, all graphical models have the same degree of sparsity, with 108 or 2.2% of all possible edges present. Note that due to the sparse structure of each graph, the proportion of shared non-edges (common zeros in the adjacency matrices) among all models is 98%.

Given the edge set E^k , we then constructed the inverse covariance matrix with the nonzero off-diagonal entries in Ω^k being uniformly generated from the $[-1, -0.5] \cup [0.5, 1]$ interval. The positive definiteness of Ω^k is guaranteed by setting the diagonal elements to be $[\phi_{\min}(\Omega^k)] + 0.1$. The covariance matrix Σ^k is then determined by

$$\Sigma_{ij}^k = (\Omega^k)_{ij}^{-1} / \sqrt{(\Omega^k)_{ii}^{-1} (\Omega^k)_{jj}^{-1}}.$$

By construction, each Σ^k corresponds to the correlation matrix for the k -th graphical model. The sparsity pattern supplied for JSEM is $\mathcal{G} = \{1, \dots, K\}$, that is assuming all graphical models share the same structure. Note by setting the parameter $\rho = 0.1$, we have created a situation where about 10% of the information in \mathcal{G} is misspecified for JSEM. This is of interest for us to see whether JSEM is robust to pattern misspecification.

To compare the overall performance of all methods, we generated $n_k = 50$ samples from each $k = 1, \dots, K$ and computed the average false positive and true positive rates of the estimated precision matrices over a fine grid of tuning parameters from 20 replications. The resulting ROC curves are shown in the right panel of Figure 2. Since both GGL and MGGM require two tuning parameters, one for controlling the *sparsity* of individual graph and the other for controlling the *similarity* across all graphs, we computed the ROC curves over a fine grid of the sparsity parameter while fixing the similarity regularization at four different levels (from low to high similarity), and plotted the one that has the largest value of area under the curve (AUC). The graph \mathcal{U} supplied for MGGM is a complete graph such that each pair of graphical models is included in the fused lasso

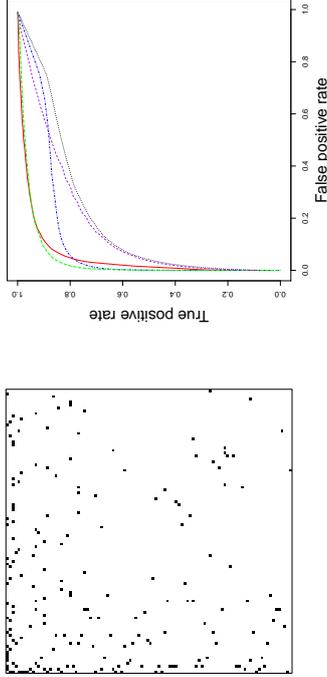


Figure 2: Simulation study 1: left panel shows the image plot of the adjacency matrix corresponding to the shared structure across all graphs. Each black cell indicates presence of an edge. The right panel shows the ROC curves for sample size $n_k = 50$: GGLasso (dotted in black), JEM-G (dotted in blue), GGL (dashed in red), MGGM (dashed in purple), JSEM (long-dash in green).

In this example, it turns out that GGL performs the best when there is only regularization on the similarity, i.e. a *group lasso* penalty on the same entry across all K precision matrices, which we expect to exhibit a similar performance to the proposed JSEM. In the right panel of Figure 2, the ROC curve of GGL falls slightly below that of JSEM. In comparison, MGGM does not perform as well despite the flexible penalty. The best curve we got from MGGM shows some advantage over the separate estimation GGLasso, but mostly falls below curves from other joint estimation methods. JEM-G performs well and is very competitive compared to GGL and JSEM for very low false positive and high true positive rates, but starts falling behind when the false positive rate is greater than 5%. In this example, JSEM performs the best with the highest ROC curve throughout the domain.

Next, we computed the estimators from different methods with $n_k = 50$ samples for each $k = 1, \dots, K$, using the tuning parameters selected by BIC. Results are summarized in Table 1, which compares the estimated precision matrices with the population version in the true model based on 50 replications under falsely discovered edges (FP), falsely deleted edges (FN), structural hamming distance (SHD), F_1 score (F1) and Frobenius norm loss (FL). The F_1 score (based on the effectiveness measure in Rijsbergen, 1979) measures the accuracy of a test by summarizing information from both FP and FN, where it reaches its best value at 1 and worst at 0. The results indicate that although GGL is good at controlling false positives, it tends to produce a high number of false negatives. The performance of MGGM is quite the opposite, with relatively small false negatives, but a huge number of false positive edges. In comparison, the proposed method JSEM achieves

Method	FP	FN	SHD	F1	FL
Glasso	35(6)	81(2)	116(5)	0.32(0.02)	0.73(< 0.01)
JEM-G	22(4)	40(4)	62(6)	0.69(0.03)	0.28(0.02)
GGL	17(6)	73(2)	90(6)	0.44(0.03)	0.29(0.02)
MGGM	286(13)	49(3)	335(13)	0.26(0.01)	0.64(0.02)
JSEM	19(4)	35(3)	54(6)	0.73(0.03)	0.25(0.02)

Table 1: Performance of different regularization methods for estimating graphical models in Simulation Study 1: average FP, FN, SHD, F1 and FL (SE) for sample size $n_k = 50$. The best cases are highlighted in bold.

a balance and obtains the highest F_1 score, as well as the lowest Frobenius norm loss. JEM-G performs slightly worse, but still well above the other three methods.

4.2 Simulation Study 2

In our second study, we consider a more structured pattern with $K = 10$ graphs. Each graphical model consists of $p = 50$ variables. Figure 3 shows the heat maps of the 10 adjacency matrices. This structured pattern is constructed as follows: we first generate the adjacency matrices corresponding to five distinct p -dimensional scale-free networks, so that the adjacency matrices in each column of the plot are the same. Next, we replace the connectivity structure of the bottom right diagonal block of size $p/2$ by $p/2$ in each adjacency matrix with that of another two distinct $p/2$ -dimensional scale-free networks, so that graphical models in each column exhibit the same connectivity pattern except in the bottom right diagonal block of their adjacency matrices. Note that by replacing the connectivity structure among the second half of the nodes, the relationships between the first half and the second half of the nodes are also altered. In summary, this structured pattern illustrates how different subsets of the edge sets across multiple graphical models can be similar, as well as exhibit differences in their topologies. To the best of our knowledge, such complex relationships have not been studied in the literature. In this setting, the proportion of shared non-edges (common zeros in the precision matrices) among all graphical models is about 60%.

Given the adjacency matrix or equivalently the edge set E^k , we generate the covariance and inverse covariance matrices in the same way as in the first simulation study. The input sparsity pattern \mathcal{S} supplied for JSEM and the graph \mathcal{U} required in MGGM are defined according to the pattern in Figure 3. We also study the effect of misspecification in \mathcal{S} by varying $\rho = 0, 0.2, 0.4, 0.6$, each corresponding to having only $(1 - \rho) * 100\%$ of the information in \mathcal{S} being correct for JSEM.

At each level of pattern misspecification, we generated $n_k = 100$ independent samples for each $k = 1, \dots, K$ and compared the ROC curves from different methods based on 20 replications in Figure 4. Again, the ROC curves for GGL and MGGM were optimized first with respect to the similarity regularization in terms of AUC. When $\rho = 0$, the results show a superior performance of JSEM, since it effectively incorporates available prior information across the various graphical models. JEM-G also yields a reasonably high ROC curve by taking advantage of the shared non-edges among all models. The performance of MGGM is comparable to that of JEM-G and much better than that of GGL. This is not surprising since MGGM benefits from knowing which pairs of

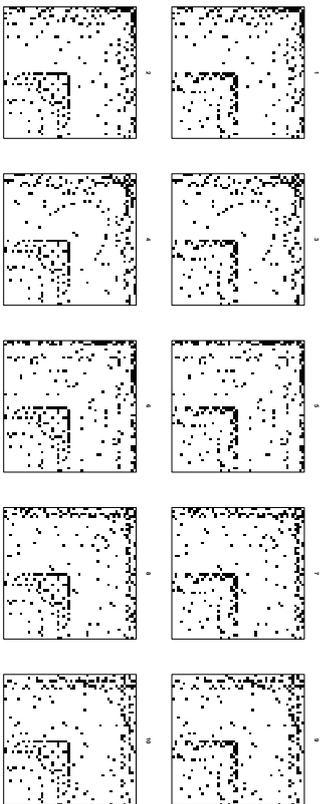


Figure 3: Simulation study 2: image plots of the adjacency matrices from all graphical models. Graphs in the same row share the same connectivity pattern at the bottom right block, whereas graphs in the same column share the same pattern at remaining locations.

graphical models to group. As ρ increases ($0 < \rho \leq 0.4$), JSEM still performs the best despite the incorrectly specified \mathcal{G} , while other methods perform not much better than the separate estimation method *Giaco*. When $\rho = 0.6$, JSEM starts suffering from the large amount of pattern misspecification as well and performing not much better than separate estimation. Note at such high ρ values, the assumption of the presence of any related structures across graphical models becomes tenuous and therefore one is better off employing a separate estimation method for each graph.

Next, we examined the finite sample performance of different methods in identifying the true graphs and estimating the precision matrices at the optimal choice of tuning parameters. Table 2 shows the deviance measures between the estimated and the true precision matrices based on 50 replications for varying levels of pattern misspecification. For $\rho \leq 0.4$, JSEM achieves a good balance between FP and FN, and yields the highest F_1 score and lowest Frobenius norm loss. JEM-G is also very competitive in controlling false positive edges and comes next in overall performance. MGGM benefits from knowing the grouping structures and has comparable performance to JEM-G. In all cases, GGL achieves low FN, but very high FP, thus resulting in low F_1 scores. When $\rho = 0.6$, the advantage of using a joint estimation method begins to diminish due to the high heterogeneity and separate estimation is recommended.

4.3 Simulation Study 3

Finally, we illustrate how direction consistency helps improve the estimation of graphical models using the previous two experimental settings. Table 3 presents the performance of thresholded JSEM when \mathcal{G} is moderately misspecified with individual to common ratio $\rho = 0.3$, based on 50 replications. Note that we used a larger sample size $n_k = 200$ in both settings to ensure that the Uniform IC required for direction consistency holds. The advantage of thresholding within groups is obvious in both settings, where the thresholded JSEM significantly reduces the number of false

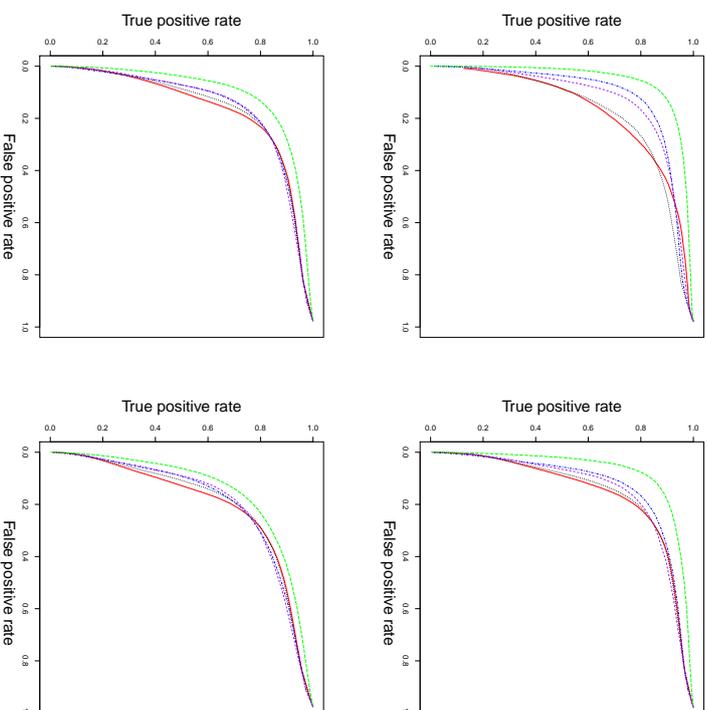


Figure 4: Simulation study 2: ROC curves for sample size $n_k = 100$: *Giaco* (dotted in black), JEM-G (dotted in blue), GGL (solid in red), MGGM (dashed in purple), JSEM (long-dash in green). The misspecification ratio ρ varies from (left to right): 0, 0.2 (top row) and 0.4, 0.6 (bottom row).

positive edges with only a small loss in the presence of false negative edges. One may notice the slight increase in Frobenius norm loss for thresholded JSEM, which is likely due to the increased presence of false negative edges. Nevertheless, the thresholded version of JSEM obtains higher F_1 scores, indicating an overall improvement in the structural estimation of all graphs.

We point out that the JSEM with thresholding procedure is most effective when ρ is moderate to small, such as $\rho < 0.5$ in this example. In other words, one believes most of the structural relationships are fairly reliable. If this is not the case, the numerical work presented strongly suggests that no joint estimation method works well, since the fundamental assumption of structural similarity

ρ	Method	FP	FN	SHD	F1	FL
0	Glasso	154(4)	38(1)	192(4)	0.51(0.01)	0.60(0.005)
	JEM-G	86(3)	36(2)	122(3)	0.62(0.01)	0.31(0.01)
	GGL	144(3)	39(1)	184(4)	0.52(0.01)	0.37(0.01)
0.2	MGGM	30(2)	67(1)	97(2)	0.59(0.01)	0.36(0.01)
	JSEM	21(2)	42(2)	63(3)	0.75(0.01)	0.28(0.01)
	Glasso	164(3)	47(1)	211(4)	0.53(0.01)	0.59(0.005)
0.4	JEM-G	92(3)	57(2)	149(3)	0.59(0.01)	0.35(0.01)
	GGL	155(3)	48(1)	203(3)	0.53(0.01)	0.37(0.01)
	MGGM	94(3)	64(1)	158(4)	0.56(0.01)	0.37(0.01)
0.6	JSEM	32(3)	64(2)	96(3)	0.67(0.01)	0.32(0.01)
	Glasso	159(3)	59(1)	218(4)	0.55(0.01)	0.57(0.005)
	JEM-G	100(3)	77(2)	177(3)	0.56(0.01)	0.37(0.01)
0.8	GGL	149(3)	61(2)	210(4)	0.55(0.01)	0.37(0.01)
	MGGM	119(3)	65(1)	184(3)	0.58(0.01)	0.37(0.01)
	JSEM	49(3)	84(2)	132(3)	0.62(0.01)	0.36(0.01)
1.0	Glasso	176(4)	73(2)	249(4)	0.54(0.01)	0.55(0.01)
	JEM-G	94(3)	109(2)	203(3)	0.52(0.01)	0.39(0.01)
	GGL	165(4)	76(2)	241(4)	0.54(0.01)	0.39(0.01)
1.2	MGGM	109(3)	95(2)	204(4)	0.55(0.01)	0.39(0.01)
	JSEM	50(3)	123(2)	173(4)	0.52(0.01)	0.38(0.01)

Table 2: Performance of different regularization methods for estimating graphical models in Simulation Study 2: average FP, FN, SHD, F1 and FL (SE) for sample size $n_k = 100$. The best cases are highlighted in bold.

Design	Method	FP	FN	SHD	F1	FL
$K = 5, p = 100$	JSEM	84(6)	12(1)	96(6)	0.71(0.01)	0.16(0.01)
	ThJSEM	29(4)	17(1)	46(4)	0.83(0.01)	0.16(0.01)
$K = 10, p = 40$	JSEM	32(2)	5(0.7)	37(2)	0.78(0.01)	0.17(0.01)
	ThJSEM	20(2)	8(0.7)	28(2)	0.82(0.01)	0.19(0.01)

Table 3: Performance of JSEM and thresholded JSEM with misspecified groups ($\rho = 0.3$): average FP, FN, SHD, F1 and FL (SE) for sample size $n_k = 200$. The better cases are highlighted in bold.

among multiple models is violated. Instead, separate estimation is recommended for handling high heterogeneity among multiple graphical models.

5. Applications

To illustrate the proposed joint estimation method in inferring real-world networks, we applied JSEM to a climate data set to study relationships between climate defining variables at multiple locations in North America, as well as a breast cancer gene expression data extracted from The Cancer Genome Atlas project (TCGA, 2012).

5.1 Application to Climate Modeling

Recent assessments from the Intergovernmental Panel on Climate Change (IPCC, Stocker et al., 2013) indicate multiple lines of evidence for climate change in the past century and these changes have caused significant impacts on natural and human systems. One common approach towards understanding the climate system has been attribution studies of detected changes to internal and external forcing mechanisms (such as solar radiation, greenhouse gases, etc.) using simulated climate models. Lozano et al. (2009) used spatial-temporal modeling to study the attribution of climate defining mechanisms from observed data. In this work, we provide an alternative to learning the complex interactions among climate defining factors exhibited across different climate zones based on observed data.

The data used in this study are monthly measurements from January 2001 to June 2005 on 16 variables including mean temperature (TMP), diurnal temperature range (DTR), maximum and minimum temperature (TMX, TMN), precipitation (PRE), vapor pressure (VAP), cloud cover (CLD), rain days (WET), potential evapotranspiration (PET), frost days (FRS), greenhouse gases (carbon dioxide (CO₂), carbon monoxide (CO), methane (CH₄), hydrogen (H₂), aerosols (AER) and solar radiation (SOL) from CRU (<http://www.cru.uea.ac.uk/cru/data>), NOAA (<http://www.esrl.noaa.gov/gmd/dv/ftpdata.html>), NASA (<http://disc.sci.gsfc.nasa.gov/aerosols>) and NCDC (<http://ftp.ncdc.noaa.gov/pub/data/nsrdb-solar/>). The data are organized as a 2.5 degree latitude by 2.5 degree longitude grid across North America. To avoid complications from any seasonality or autocorrelation of the data, we aggregated the monthly time series into bins of 3-month intervals and took first differences of the quarterly data. The data after differencing were further normalized. Details on the pre-processing steps are included in Appendix D. Next, we randomly selected $K = 27$ locations spanning all types of climate from the 2.5 by 2.5 degree grid of North America (see Figure 5). This gives us an $n \times p$ matrix at each of the 27 locations, corresponding to $n = 17$ observations for the $p = 16$ climate defining variables. At each location, the conditional dependency network is of dimension $p \times p$, which has $16 \times 15/2 = 120$ edges to be inferred.

Our goal is to infer the conditional dependency networks for all locations simultaneously based on available spatial information, obtained from the classification of climate zones in Peel et al. (2007). Specifically, we assume that AER and SOL have one common connectivity pattern with other variables in the geographical south of North America and another common pattern in the north. The definition of the south and north is given in Figure 5. Variables on greenhouse gases (CO₂, CO, CH₄ and H₂) are assumed to interact with other variables (except AER and SOL) in the same fashion within each of the four climate groups, that is Mid-latitude Desert, Semiarid Steppe, Humid Subtropical and Humid Continental. The connectivity patterns among all remaining variables are assumed to be the same within each of the six distinct climate zones in Figure 5.

We used BIC on the normalized data to select the tuning parameter λ for the proposed JSEM. At the optimal λ , we applied our method coupled with complementary pairs stability selection (Shah

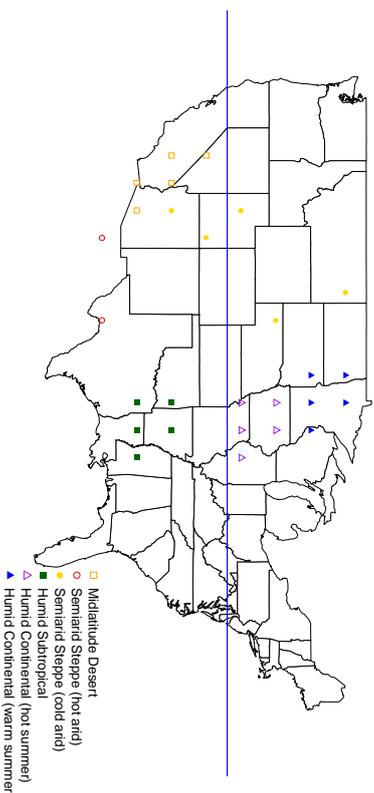


Figure 5: The selected 27 locations based on climate classification. The solid line separates the south and north of North America and corresponds to latitude 39° N.

and Samworth, 2013) to identify the interaction networks at the 27 locations. To perform stability selection, we ran our method 50 times on two randomly drawn complementary pairs of sizes 8 and 9, and kept only edges that are selected over 70% of the time.

Due to space limitation, we present in Figure 6 the estimated networks at the six distinct climate zones. Readers are referred to Appendix D for the complete picture of the 27 networks from all the 27 locations under study, as well as more detailed comparisons. Although we do not impose the assumption on sharing of a single common structure across all locations, there are common edges (solid) identified for all climate zones, reflecting key features of climate defining regardless of geographical location. Such relationships are consistent with how the corresponding climate defining variables are defined, as well as how the data are collected (Harris et al., 2014). The Mid-latitude Desert and Semiarid Steppe climate zones share the edge between DTR and CLD, indicating that they are correlated conditional on all other variables. Similar relationships have also been found over drier regions in Zhou et al. (2009). In addition, one can see that the variable FRS interacts mainly with PET at Mid-latitude Desert and Semiarid Steppe climates, whereas it is partially correlated with both PRE and TMN (or TMX) at Continental climates. This can be explained from the distinction between these climate zones. At Humid Continental climate, precipitation is relatively well distributed year-round in most areas and snowfall occurs in all areas. It is thus not difficult to see why precipitation (PRE) and temperature related variables correlate with the number of frost days (FRS). Further, a primary criterion of an area characterized as Mid-latitude Desert or Semiarid Steppe is that it receives precipitation below potential evapotranspiration (PET), which possibly explains why FRS is partially correlated only with PET for Mid-latitude Desert and Semiarid Steppe climate. Finally, we point out that the inferred networks at neighboring climate zones are more similar, such as Semiarid Steppe (hot arid and cold arid), or Humid Continental (hot summer and

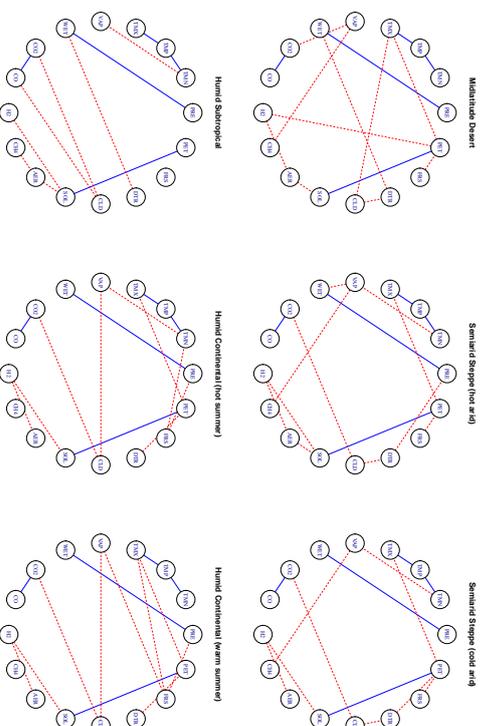


Figure 6: Estimated climate networks at the six distinct climate zones using JSEM, with edges shared across all locations blue solid and differential edges red dashed.

warm summer), whereas those with dramatically different climate show significantly different connectivity patterns. These common and individual interactions can prove critical in understanding the mechanisms of climate defining, and facilitate decision making in maintaining the best environmental results.

5.2 Application to Breast Cancer

Breast cancer is the most common cancer in women worldwide, with nearly 1.7 million new cases diagnosed in 2012 (second most common cancer overall). This represents about 12% of all new cancer cases and 25% of all cancers in women (Ferlay et al., 2013). Breast cancer is hormone related and this leads to a basic classification of cancer cells. Specifically, a cancer is called estrogen-receptor-positive (or ER+) if it has receptors for estrogen, and hence the cells receive signals from estrogen that could promote their growth. It is estimated that about 80% of all breast cancer cases are ER+ and they are more likely to respond to hormone therapy. Further, ER+ status is associated with better survival rates, especially if the cancer is diagnosed early. On the other hand, the ER-status lacks the estrogen receptor and in general exhibits poorer survival rates. Note that the presence/absence of other hormone receptors (progesterone and HER2) also play an important role in breast cancer tumor classification, therapeutic strategies and survival rates.

The breast cancer data set (TCGA, 2012) contains RNA-seq measurements for 17296 genes from 1033 breast cancer specimens, including ER+, ER- and other unevaluated cases. Due to the

overall small sample size, we first reduced the number of variables by focusing on a subset of the genes that are present in the 44 KEGG pathways in Table 4. These pathways correspond to the major signaling and biochemical ones that have been reported in the literature of playing a significant role in all cancer types. This leaves for further consideration 800 genes with 403 samples from the ER+ and 117 from the ER- classes.

The structural similarity between the networks for ER+ and ER- status was defined based on the third column in Table 4, which indicates whether the pathway is significantly enriched when testing ER+ vs ER- status via NetGSA (Ma et al., 2016), and complemented through literature searches. If one pathway is not significantly enriched, then the genes belonging to the pathway are considered to share a common structure under both ER+ and ER- status. However, due to overlaps amongst pathways (since some of their members are assigned to multiple ones in the KEGG database), only genes that did not belong to any of the differential pathways were used to define the common structure. The remaining genes are assumed to have distinct structures under the two conditions.

We then used BIC on the normalized data to select the tuning parameter λ for the proposed JSEM. At the optimal λ , we applied our method coupled with complementary pairs stability selection (Shah and Samworth, 2013) to identify the interaction networks for the ER+ and ER- classes, respectively. Due to the large number of variables, visualization of the estimated networks at the individual gene level is challenging. Instead, we examine the interactions among pathways in Figure 7 to gain insight into their co-regulation behavior. The weighted pathway level network is defined as follows. Let each node in the network represent one pathway, with size proportional to the size of the corresponding pathway. A weighted edge between two pathways P_1 and P_2 is defined as the number of nonzero partial correlations between genes in P_1 and those in P_2 (normalized by the sizes of the two pathways). Links visualized in Figure 7 are the top 5% of the weighted edges, where ranking is based on edge weights. Note pathways that are isolated from all others were removed.

The first thing to note is that structural information provided enables us to estimate a much more dense graph than either separate estimation or an agnostic method like JEM-G (see Figure 12 in Appendix D), which in turn aids biological interpretation. We focus next on the interactions between pathways, as shown in Figure 7. The central role of known cancer related pathways—TGF- β , p53, MAPK and hedgehog—is apparent. Further, we see high degree of interconnections between signaling and biochemical pathways including glycolysis gluconeogenesis, pyrimidine, cysteine and methionine, and tryptophan, which is expected due to the impact of energy metabolism in tumor growth and progression. One surprising finding is that the p53 pathway is connected only in the ER+ class, but we suspect that this may be the case due to the big discrepancy in terms of available samples for the ER+ and ER- classes and the large number of genes present. In summary, the proposed method captures established cross-talk patterns between various signaling and biochemical pathways, which is not the case with competing methods or with separate estimation.

6. Discussion

This work introduces a flexible joint structural estimation method (JSEM) that incorporates *a priori* known structural relationships between multiple graphical models. The proposed method works well in situations where there is a large number of graphical models, but external similarity information is available only for sub-components of the models. In practice, if not all entry-wise structural relationships across multiple graphical models are available, it is recommended to add constraints at mainly edge pairs that are likely to share the same structures instead of providing a highly mis-

Vertex id	Vertex names	KEGG names	Status
1	glycolysis_gluconeogenesis	glycolysis_gluconeogenesis	TRUE
2	citrate_cycle_tca_cycle	citrate_cycle_tca_cycle	FALSE
3	pentose_phosphate	pentose_phosphate_pathway	TRUE
4	fructose_and_mannose	fructose_and_mannose_metabolism	TRUE
5	galactose	galactose_metabolism	TRUE
6	fatty_acid	fatty_acid_metabolism	FALSE
7	oxidative_phosphorylation	oxidative_phosphorylation	FALSE
8	purine	purine_metabolism	TRUE
9	pyrimidine	pyrimidine_metabolism	TRUE
10	glycine_serine_and_threonine	glycine_serine_and_threonine_metabolism	FALSE
11	cysteine_and_methionine	cysteine_and_methionine_metabolism	TRUE
12	valine_leucine_and_isoleucine	valine_leucine_and_isoleucine_degradation	TRUE
13	lysine	lysine_degradation	FALSE
14	arginine_and_proline	arginine_and_proline_metabolism	FALSE
15	tryptophan	tryptophan_metabolism	FALSE
16	beta_alanine	beta_alanine_metabolism	TRUE
17	glutathione	glutathione_metabolism	TRUE
18	starch_and_sucrose	starch_and_sucrose_metabolism	TRUE
19	amino_sugar_and_nucleotide_sugar	amino_sugar_and_nucleotide_sugar_metabolism	FALSE
20	ppar	ppar_signaling_pathway	TRUE
21	mapk	mapk_signaling_pathway	FALSE
22	erbB	erbB_signaling_pathway	TRUE
23	calcium	calcium_signaling_pathway	FALSE
24	chemokine	chemokine_signaling_pathway	TRUE
25	phosphatidylinositol	phosphatidylinositol_signaling_system	FALSE
26	cell_cycle	cell_cycle	TRUE
27	p53	p53_signaling_pathway	TRUE
28	mtor	mtor_signaling_pathway	FALSE
29	wnt	wnt_signaling_pathway	FALSE
30	notch	notch_signaling_pathway	FALSE
31	hedgehog	hedgehog_signaling_pathway	TRUE
32	tgf_beta	tgf_beta_signaling_pathway	TRUE
33	vegf	vegf_signaling_pathway	FALSE
34	tolL	tolL_receptor_signaling_pathway	TRUE
35	nod_like	nod_like_receptor_signaling_pathway	TRUE
36	rig_i_like	rig_i_like_receptor_signaling_pathway	FALSE
37	jak_stat	jak_stat_signaling_pathway	TRUE
38	l_cell	l_cell_receptor_signaling_pathway	FALSE
39	b_cell	b_cell_receptor_signaling_pathway	FALSE
40	fc_epsilon_ri	fc_epsilon_ri_signaling_pathway	TRUE
41	neurotrophin	neurotrophin_signaling_pathway	FALSE
42	insulin	insulin_signaling_pathway	FALSE
43	gmh	gmh_signaling_pathway	TRUE
44	adipocytokine	adipocytokine_signaling_pathway	TRUE

Table 4: List of simplified vertex (pathway) names, their matching names in KEGG and whether the corresponding pathway is used to define structural similarity

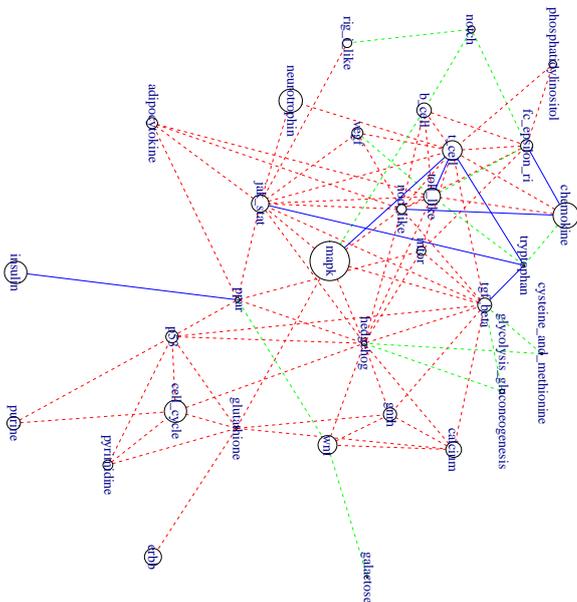


Figure 7: Estimated pathway networks for the ER+ and ER- classes using JSEM, with edges shared across all locations blue solid and differential edges red dashed (ER+ / green dashed (ER-)).

specified structured sparsity pattern. On the other hand, if more structural information is available, one may generalize the group lasso penalty to incorporate additional structural constraints.

The theoretical guarantees of JSEM rely on two important, but standard in the literature, assumptions: the restricted eigenvalue assumption (A1) and the uniform IC assumption in (A3). In practice, it might be difficult to verify whether these assumptions are fulfilled, especially the more stringent assumption (A3). For the latter condition, Meinshausen and Yu (2009) observe that the irrepresentability condition (a variant of A3) may be violated in practical settings in the presence of highly correlated variables; nevertheless, the lasso estimates are still ℓ_2 consistent, under (A1).

Acknowledgments

The authors would like to thank two anonymous reviewers for helpful comments and suggestions. The work of GM was supported in part by NSF awards DMS-1228164 and DMS-1545277 and NIH award 7R21GM10171903.

Appendix A. Proof of Theorem 1

To prove the rate of convergence in Theorem 1, we look at three key steps: nodewise regression in subsection A.1, selecting the edge set in A.2 and maximum likelihood refitting in A.3. More information can be found in Appendix E. When it is clear, we shall use \sum_k as a short notation for $\sum_{k=1}^K$.

A.1 Regression

For $j \neq i, g \in \mathcal{G}^{ij}, k \in g$, let $\epsilon_j^k = \mathbf{X}_j^k - \sum_{j' \neq i} \theta_{0,j'}^k \mathbf{X}_j^k$. Let $\langle a, b \rangle$ represent the inner product between two vectors a and b . Denote $\zeta_{ij}^k = \langle \epsilon_j^k, \mathbf{X}_j^k \rangle / n$ and $\zeta_{ij}^{|\beta|} = \langle \epsilon_j^{|\beta|}, \mathbf{X}_j^k \rangle_{k \in g} \in \mathbb{R}^{|\beta|}$. Consider the random event $\mathcal{A} = \bigcap_{i,j \neq i,g} \mathcal{A}_{ij}^g$, where $\mathcal{A}_{ij}^g = \{2\|\zeta_{ij}^{|\beta|}\| \leq \lambda_{ij}^g\}$. By Lemma E.2, if we choose λ_{ij}^g as

$$\lambda_{ij}^g \geq \max_{k \in g} \frac{2}{\sqrt{n\epsilon_{0,iz}^k}} \left(\sqrt{|\beta|} + \frac{\pi}{\sqrt{2}} \sqrt{q \log G_0} \right) \quad (10)$$

with $q > 1$, then $\mathbb{P}(\mathcal{A}) \geq 1 - 2pG_0^{1-q}$. We first present the following proposition that establishes oracle bounds for $\hat{\Theta}_i - \Theta_{0,i}$ under the chosen λ_{ij}^g .

Proposition A.1 For $i = 1, \dots, p$, consider the problem (2) and choose λ_{ij}^g as in (10). Let $\hat{\Theta}_i$ be the solution to problem (2). If Assumption (A1) holds with $\kappa^2 = \kappa^2(s_0)$, then for any solution $\hat{\Theta}_i$ of problem (2), we have on the event \mathcal{A}

$$\sum_{j \neq i, g \in \mathcal{G}^{ij}} \|\hat{\theta}_{ij}^{|\beta|} - \theta_{0,ij}^{|\beta|}\| \leq \frac{16}{\kappa^2 \lambda_{\min}} \sum_{(j,g) \in J(\Theta_{0,i})} (\lambda_{ij}^g)^2, \quad (11)$$

$$\mathcal{M}(\hat{\Theta}_i) \leq \frac{64\phi_{\max}}{\kappa^2 \lambda_{\min}^2} \sum_{(j,g) \in J(\Theta_{0,i})} (\lambda_{ij}^g)^2, \quad (12)$$

where $\lambda_{\min} = \min_{i,j \neq i, g \in \mathcal{G}^{ij}} \lambda_{ij}^g$, $\mathcal{M}(\hat{\Theta}_i) = |J(\hat{\Theta}_i)|$ and ϕ_{\max} is the maximal eigenvalue of $(\mathbf{X}^k)^T \mathbf{X}^k / n$ for all $k = 1, \dots, K$. If, in addition, Assumption (A1) holds with $\kappa^2(2s_0)$, then for any solution $\hat{\Theta}_i$ of problem (2) we have that

$$\|\hat{\Theta}_i - \Theta_{0,i}\|_F \leq \frac{4\sqrt{10}}{\kappa^2(2s_0)} \sum_{(j,g) \in J(\Theta_{0,i})} (\lambda_{ij}^g)^2 \lambda_{\min} \sqrt{s_i} \quad (13)$$

By Assumption (A2), $\epsilon_{0,iz}^k \geq \phi_{\min}(\Sigma_0^k) = \phi_{\max}^{-1}(\Sigma_0^k) \geq d_0$ for all i, k . Thus, (10) implies that we can choose $\lambda_{ij}^g = \lambda_{\max}$ as

$$\lambda_{\max} = \frac{2}{\sqrt{nd_0}} \left(\sqrt{|\beta|_{\max}} + \frac{\pi}{\sqrt{2}} \sqrt{q \log G_0} \right), \quad (14)$$

with $q > 1$ for all 3-tuples (i, j, g) . Then we can rewrite the oracle inequalities in (12) and (13) as

$$\mathcal{M}(\hat{\Theta}_i) \leq \frac{64\phi_{\max}}{\kappa^2} s_i, \quad (15)$$

$$\|\hat{\Theta}_i - \Theta_{0,i}\|_F \leq \frac{8\sqrt{10}}{\kappa^2(2s_0)\sqrt{d_0}} \left(\sqrt{|g_{\max}|} + \frac{\pi}{\sqrt{2}} \sqrt{q \log G_0} \right) \sqrt{\frac{s_i}{n}}. \quad (16)$$

Detailed proof of Proposition A.1 follows similarly to that of Theorem 3.1 in Lounici et al. (2011) and can be found in Appendix E.

A.2 Selecting Edge Set

Given the estimates $\hat{\Theta}_i$ ($i = 1, \dots, p$), define \hat{E}^k as in (3) the estimated set of edges in graph $k = 1, \dots, K$. For every k , let $\hat{\Omega}^k = \text{diag}(\Omega_0^k) + \Omega_{0, E_0^k \cap \hat{E}^k}$ and $\hat{\Sigma}^k = (\hat{\Omega}^k)^{-1}$. Let

$$C_{\text{bias}} = \frac{8\sqrt{10}c_0}{\kappa^2(2s_0)\sqrt{d_0}}.$$

The following corollary is an immediate result of (15) and (16).

Corollary A.1 Consider \hat{E}^k ($k = 1, \dots, K$) selected in (3). Suppose all conditions in Theorem 1 are satisfied. Choose $\lambda_{ij}^k = \lambda_{\max}$ as defined in (14) with $q > 1$. Then we have on the event \mathcal{A}

$$|\hat{E}^k| \leq \frac{64\phi_{\max}}{\kappa^2(s_0)} S_0, \quad k = 1, \dots, K, \quad (17)$$

and

$$\frac{1}{K} \sum_k \|\hat{\Omega}^k - \Omega_0^k\|_F \leq \frac{1}{\sqrt{K}} \left\{ \sum_k \|\hat{\Omega}^k - \Omega_0^k\|_F^2 \right\}^{1/2} \leq C_{\text{bias}} \sqrt{\frac{S_0}{nK}} \left(\sqrt{|g_{\max}|} + \frac{\pi}{\sqrt{2}} \sqrt{q \log G_0} \right), \quad (18)$$

where G_0 is the maximum number of groups in all p regressions, S_0 is the total number of relevant groups, and $|g_{\max}|$ is the maximum group size.

The bound in (17) says that the cardinality of the estimated set of edges is at most of the order of S_0 and proves essential in controlling the error rate of the maximum likelihood estimate $\hat{\Omega}^k$ in the refitting step. Further, the second inequality in (18) implies

$$\left\{ \sum_k \|\hat{\Omega}^k - \Omega_0^k\|_F^2 \right\}^{1/2} \leq \tau_1 d_0,$$

provided the sample size n satisfies for $0 < \tau_1 < 1$,

$$n \geq S_0 \left(\sqrt{|g_{\max}|} + \frac{\pi}{\sqrt{2}} \sqrt{q \log G_0} \right)^2 \left(\frac{C_{\text{bias}}}{\tau_1 d_0} \right)^2.$$

It follows immediately that on the event \mathcal{A} , we can bound the spectrum of $\hat{\Omega}^k$ ($k = 1, \dots, K$) as follows. For a symmetric matrix A , let $\|A\|$ represent the spectral norm of A , which is equal to $\phi_{\max}(A)$. By definition,

$$\phi_{\min}(\hat{\Omega}^k) = \min_{v: v^T v = 1} v^T \hat{\Omega}^k v = \min_{v: v^T v = 1} \{v^T \hat{\Omega}^k v + v^T (\hat{\Omega}^k - \Omega_0^k) v\} \geq \phi_{\min}(\Omega_0^k) - \|\hat{\Omega}^k - \Omega_0^k\|.$$

Since $\phi_{\min}(\Omega_0^k) \geq d_0$ by Assumption (A2), we have

$$\begin{aligned} \phi_{\min}(\hat{\Omega}^k) &\geq \phi_{\min}(\Omega_0^k) - \|\hat{\Omega}^k - \Omega_0^k\| \geq \phi_{\min}(\Omega_0^k) - \|\hat{\Omega}^k - \Omega_0^k\|_F \\ &\geq \phi_{\min}(\Omega_0^k) - \left\{ \sum_k \|\hat{\Omega}^k - \Omega_0^k\|_F^2 \right\}^{1/2} \geq (1 - \tau_1) d_0 > 0, \end{aligned} \quad (19)$$

In addition, we have an upper bound for the maximum eigenvalue of $\hat{\Omega}^k$,

$$\begin{aligned} \phi_{\max}(\hat{\Omega}^k) &\leq \phi_{\max}(\Omega_0^k) + \|\hat{\Omega}^k - \Omega_0^k\| \leq \phi_{\max}(\Omega_0^k) + \|\hat{\Omega}^k - \Omega_0^k\|_F \\ &\leq \phi_{\max}(\Omega_0^k) + \left\{ \sum_k \|\hat{\Omega}^k - \Omega_0^k\|_F^2 \right\}^{1/2} \leq c_0 + \tau_1 d_0 < \infty. \end{aligned} \quad (20)$$

A.3 Refitting

Let $\hat{\Omega}^k$ ($k = 1, \dots, K$) be defined in (4) and

$$r_n = C_{\text{bias}} \sqrt{\frac{S_0}{n}} \left(\sqrt{|g_{\max}|} + \frac{\pi}{\sqrt{2}} \sqrt{q \log G_0} \right). \quad (21)$$

Proof [of Theorem 1.] In view of Corollary A.1, it suffices to show that

$$\sum_k \|\hat{\Omega}^k - \tilde{\Omega}^k\|_F^2 \leq O(r_n^2),$$

since by Cauchy-Schwarz inequality,

$$\frac{1}{K} \sum_k \|\hat{\Omega}^k - \tilde{\Omega}^k\|_F \leq \frac{1}{\sqrt{K}} \left\{ \sum_k \|\hat{\Omega}^k - \tilde{\Omega}^k\|_F^2 \right\}^{1/2},$$

and by triangle inequality,

$$\frac{1}{K} \sum_k \|\hat{\Omega}^k - \Omega_0^k\|_F \leq \frac{1}{K} \sum_k \|\hat{\Omega}^k - \tilde{\Omega}^k\|_F + \frac{1}{K} \sum_k \|\tilde{\Omega}^k - \Omega_0^k\|_F.$$

For $k = 1, \dots, K$, let $\Delta^k = \Omega^k - \tilde{\Omega}^k \in \mathbb{M}(p, p)$ and $\hat{\Delta}^k = \hat{\Omega}^k - \tilde{\Omega}^k$. Let

$$Q(\Omega) = \sum_k \left\{ \text{tr}(\hat{\Sigma}^k \Omega^k) - \log \det(\Omega^k) - \text{tr}(\hat{\Sigma}^k \tilde{\Omega}^k) + \log \det(\tilde{\Omega}^k) \right\}.$$

Since $(\hat{\Omega}^k)_{k=1}^K$ minimizes $Q(\Omega)$, $(\hat{\Delta}^k)_{k=1}^K$ minimizes $G(\Delta) = Q(\hat{\Omega} + \Delta)$. Recall the definition $S_E^+ = \{\Gamma \in \mathbb{R}^{p \times p} : \Gamma > 0 \text{ and } \Gamma_{ij} = 0, \text{ for all } (i, j) \notin E \text{ where } i \neq j\}$. For $k = 1, \dots, K$, define a sequence of convex sets

$$\mathcal{U}_k(\tilde{\Omega}^k) = \{\Gamma - \tilde{\Omega}^k | \Gamma \in S_E^+\}.$$

The main idea of the proof is as follows. For a sufficiently large $M > 0$, consider the set

$$\mathcal{T}_n = \{(\Delta^1, \dots, \Delta^K) : \Delta^k \in \mathcal{U}_k(\tilde{\Omega}^k), \sum_k \|\Delta^k\|_F^2 = M r_n^2\}.$$

Write $0_{p \times p}$ the zero matrix in $\mathbb{M}(n, p)$. It is clear that $G(\Delta)$ is a convex function and $G(\hat{\Delta}) \leq G(0_{p \times p}) = 0$. Thus if we can show $\inf_{\Delta \in \mathcal{T}_n} G(\Delta) > 0$, the minimizer $\hat{\Delta}$ must be inside the ball defined by \mathcal{T}_n . That is $\sum_k \|\Delta^k\|_F^2 \leq M r_n^2$. To see this, note that the convexity of $Q(\hat{\Omega})$ implies that $\inf_{\Delta \in \mathcal{T}_n} Q(\hat{\Omega} + \Delta) > Q(\hat{\Omega}) = 0$. There exists therefore a local minimizer in the ball $\{\tilde{\Omega}^k + \Delta^k : \sum_k \|\Delta^k\|_F^2 \leq M r_n^2\}$, or equivalently, $\sum_k \|\hat{\Delta}^k\|_F^2 \leq M r_n^2$.

In the remainder of the proof, we focus on

$$G(\Delta) = \sum_k \left\{ \text{tr}(\tilde{\Sigma}^k \Delta^k) - \log \det(\tilde{\Omega}^k + \Delta^k) + \log \det(\tilde{\Omega}^k) \right\}.$$

Applying Taylor expansion to the logarithm terms in the above equation, we have

$$\begin{aligned} & \log \det(\tilde{\Omega}^k + \Delta^k) - \log \det(\tilde{\Omega}^k) \\ &= \text{tr}(\tilde{\Sigma}^k \Delta^k) - \text{vec}(\Delta^k)^T \left\{ \int_0^1 (1-t)(\tilde{\Omega}^k + t\Delta^k)^{-1} \otimes (\tilde{\Omega}^k + t\Delta^k)^{-1} dt \right\} \text{vec}(\Delta^k), \end{aligned}$$

where \otimes is the Kronecker product, and $\text{vec}(\Delta^k)$ is Δ^k vectorized to match the dimensions of the Kronecker product. Therefore, we can rewrite $G(\Delta) = L_1 - L_2 + L_3$, with

$$\begin{aligned} L_1 &= \sum_k \text{tr} \{ (\tilde{\Sigma}^k - \Sigma_0^k) \Delta^k \}, \\ L_2 &= \sum_k \text{tr} \{ (\tilde{\Sigma}^k - \Sigma_0^k) \Delta^k \}, \\ L_3 &= \sum_k \text{vec}(\Delta^k)^T \left\{ \int_0^1 (1-t)(\tilde{\Omega}^k + t\Delta^k)^{-1} \otimes (\tilde{\Omega}^k + t\Delta^k)^{-1} dt \right\} \text{vec}(\Delta^k). \end{aligned}$$

Next we bound each term separately.

Recall for every k , Σ_0^k and $\tilde{\Sigma}^k$ represent the correlation and the sample correlation matrix, respectively. By Lemma 14 of Zhou et al. (2011) [see details on page 3003],

$$\mathbb{P} \left\{ |\hat{\sigma}_{0,ij}^k - \sigma_{0,ij}^k| \geq t \right\} \leq \exp \left(- \frac{3nr t^2}{10t + (\sigma_{0,ij}^k)^2} \right) \leq \exp \left(- \frac{3nr t^2}{20} \right), \quad (22)$$

for $0 \leq t \leq \{1 + (\sigma_{0,ij}^k)^2\}^{1/2}$. Then the union sum inequality and (22) imply that, with probability tending to 1,

$$\begin{aligned} \max_{k,i,j} |\hat{\sigma}_{ij}^k - \sigma_{0,ij}^k| &\leq c_1 \sqrt{\frac{1}{nK}} \left(\sqrt{|g_{\max}|} + \frac{\pi}{\sqrt{2}} \sqrt{q \log G_0} \right), \\ n &\geq \frac{4c_1^2}{K} \left(\sqrt{|g_{\max}|} + \frac{\pi}{\sqrt{2}} \sqrt{q \log G_0} \right)^2, \end{aligned}$$

provided that the sample size satisfies

where $c_1 > 0$ is a constant. Write $\Delta^k = \Delta^{k,+} + \Delta^{k,-}$ such that $\Delta^{k,+} = \text{diag}(\Delta^k)$ and $\Delta^{k,-}$ consists of the off-diagonal entries of Δ^k . Then

$$|L_1| \leq \sum_k \sum_{i \neq j} |\hat{\sigma}_{ij}^k - \sigma_{0,ij}^k| |\Delta_{ij}^k| \leq c_1 \sqrt{\frac{1}{nK}} \left(\sqrt{|g_{\max}|} + \frac{\pi}{\sqrt{2}} \sqrt{q \log G_0} \right) \sum_k \|\Delta^{k,-}\|_1.$$

By Cauchy-Schwarz inequality and the definition of $\Delta^k \in \mathcal{U}_n(\tilde{\Omega}^k)$,

$$\sum_k \|\Delta^{k,-}\|_1 \leq \sum_k (2|\hat{E}^k|)^{1/2} \|\Delta^{k,-}\|_F \leq \max_k (2|\hat{E}^k|)^{1/2} \sqrt{K} \left(\sum_k \|\Delta^k\|_F^2 \right)^{1/2}.$$

Using the bound of \hat{E}^k in (17) and the definition of r_n , we obtain

$$\begin{aligned} |L_1| &\leq c_1 \sqrt{\frac{1}{n}} \left(\sqrt{|g_{\max}|} + \frac{\pi}{\sqrt{2}} \sqrt{q \log G_0} \right) \frac{8\sqrt{2}\phi_{\max}^{\mathcal{S}_0}}{K^{1/2}(s_0)} \left(\sum_k \|\Delta^k\|_F^2 \right)^{1/2} \\ &= \frac{8\sqrt{2}c_1 \sqrt{\phi_{\max}^{\mathcal{S}_0}} r_n (M r_n^2)^{1/2}}{C_{\text{bias}} \kappa(s_0)} = \frac{8\sqrt{2}c_1 \sqrt{\phi_{\max}^{\mathcal{S}_0}} \sqrt{M} r_n^2}{C_{\text{bias}} \kappa(s_0)}, \end{aligned} \quad (23)$$

where the first equality in (23) follows from the definition of r_n in (21).

Using results from (19) and (18) together with Cauchy-Schwarz inequality, the second term L_2 can be bounded by

$$\begin{aligned} |L_2| &\leq \sum_k \langle \tilde{\Sigma}^k - \Sigma_0^k, \Delta^k \rangle \leq \sum_k \|\tilde{\Sigma}^k - \Sigma_0^k\|_F \|\Delta^k\|_F \leq \sum_k \|\Delta^k\|_F \frac{\|\tilde{\Omega}^k - \Omega_0^k\|_F}{\phi_{\min}(\tilde{\Omega}^k) \phi_{\min}(\Omega_0^k)} \\ &\leq \frac{1}{(1-\tau_1)d_0^2} \left(\sum_k \|\Delta^k\|_F^2 \right)^{1/2} \left(\sum_k \|\tilde{\Omega}^k - \Omega_0^k\|_F^2 \right)^{1/2} \leq \frac{\sqrt{M} r_n^2}{(1-\tau_1)d_0^2}, \end{aligned} \quad (24)$$

where the last inequality in (24) comes from the rotation invariant property of the Frobenius norm.

Finally we bound L_3 . Suppose for a small constant $0 < \tau_2 < 1$ such that $\tau_1 + \tau_2 < 1$, the sample size n satisfies

$$n \geq M S_0 \left(\sqrt{|g_{\max}|} + \frac{\pi}{\sqrt{2}} \sqrt{q \log G_0} \right)^2 \left(\frac{C_{\text{bias}}}{\tau_2 d_0} \right)^2,$$

then $\sqrt{M} r_n \leq \tau_2 d_0$. By (20), $\phi_{\max}(\tilde{\Omega}^k)$ is bounded above by $c_0 + \tau_1 d_0$. Therefore for $\Delta \in \mathcal{T}_n$,

$$\begin{aligned} \phi_{\max}(\tilde{\Omega}^k + \Delta^k) &\leq c_0 + \tau_1 d_0 + \|\Delta^k\| \leq c_0 + \tau_1 d_0 + \|\Delta^k\|_F \\ &\leq c_0 + \tau_1 d_0 + \left(\sum_k \|\Delta^k\|_F^2 \right)^{1/2} \leq c_0 + (\tau_1 + \tau_2) d_0, \\ \phi_{\min}(\tilde{\Omega}^k + \Delta^k) &\geq (1 - \tau_1) d_0 - \|\Delta^k\| \geq (1 - \tau_1) d_0 - \|\Delta^k\|_F \\ &\geq (1 - \tau_1) d_0 - \left(\sum_k \|\Delta^k\|_F^2 \right)^{1/2} \geq (1 - \tau_1 - \tau_2) d_0 > 0. \end{aligned}$$

For $\tilde{\Omega}^k$ and Δ^k defined above, Zhou et al. (2011) showed that $\tilde{\Omega}^k + t\Delta^k \succ 0$, $t \in [0, 1]$, for all $k = 1, \dots, K$ on the event \mathcal{A} . Thus, following similar arguments as in Rothman et al. (2008, page 502), we have

$$\begin{aligned} |L_3| &\geq \frac{1}{2} \sum_k \phi_{\min}^2(\tilde{\Omega}^k + \Delta^k)^{-1} \|\Delta^k\|_F^2 = \frac{1}{2} \sum_k \phi_{\max}^{-2}(\tilde{\Omega}^k + \Delta^k) \|\Delta^k\|_F^2 \\ &\geq \frac{M r_n^2}{2(c_0 + \tau_1 d_0 + \tau_2 d_0)^2}. \end{aligned}$$

Combining the above three bounds, we thus have

$$\begin{aligned} G(\Delta) &\geq |L_3| - |L_4| - |L_2| \\ &\geq \frac{Mr_n^2}{2(c_0 + \tau_1 d_0 + \tau_2 d_0)^2} - \frac{8\sqrt{2}c_1\sqrt{\phi_{\max}}}{C_{\text{bias}\kappa}(s_0)}\sqrt{Mr_n^2} - \frac{\sqrt{Mr_n^2}}{(1 - \tau_1)d_0^2} \\ &\geq Mr_n^2 \left\{ \frac{1}{2(c_0 + \tau_1 d_0 + \tau_2 d_0)^2} - \frac{8c_1\sqrt{2\phi_{\max}}}{C_{\text{bias}\kappa}(s_0)}\sqrt{M} - \frac{1}{(1 - \tau_1)d_0^2\sqrt{M}} \right\} > 0, \end{aligned}$$

for M sufficiently large. \blacksquare

Appendix B. Proof of Theorem 2

Consider the group lasso estimator $\hat{\Theta}_i$ defined in (2). Since the problem (2) is a special case of the generic group lasso in Basu et al. (2015), we adapt their results in Theorem 4.1 to our design.

Proof Let \mathcal{X}_i be the block diagonal matrix composed of all variables but \mathbf{X}_i^k ($k = 1, \dots, K$), that is

$$\mathcal{X}_i = \begin{pmatrix} \mathbf{X}_{-i}^1 & & \\ & \ddots & \\ & & \mathbf{X}_{-i}^K \end{pmatrix}.$$

After rearranging the columns of \mathcal{X}_i , we assume without loss of generality $\mathcal{X}_i = (\mathcal{X}_{i(1)}, \mathcal{X}_{i(2)})$ such that

$$\mathcal{X}_{i(1)} = \text{diag}(\mathbf{X}_{I_1}^1, \dots, \mathbf{X}_{I_K}^K)$$

is the sub-matrix consisting of all relevant variables. Denote the Gram matrix

$$C = \frac{1}{n} \mathcal{X}_i^T \mathcal{X}_i = \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}$$

with $C_{11} = \mathcal{X}_{i(1)}^T \mathcal{X}_{i(1)}/n$ and $C_{22} = \mathcal{X}_{i(2)}^T \mathcal{X}_{i(2)}/n$. C_{12} and C_{21} are also defined accordingly. Note due to the block diagonal structure of $\mathcal{X}_{i(1)}$, C_{11} is also block diagonal.

Now consider interchanging the columns of \mathcal{X}_i such that

$$\tilde{\mathcal{X}}_i = \mathcal{X}_i \text{diag}(R_1, R_2) = (\mathcal{X}_{i(1)} R_1, \mathcal{X}_{i(2)} R_2) = (\tilde{\mathcal{X}}_{i(1)}, \tilde{\mathcal{X}}_{i(2)}),$$

where the columns of $\tilde{\mathcal{X}}_{i(1)}$ and $\tilde{\mathcal{X}}_{i(2)}$ are ordered in groups of variables. Here R_l is the product of elementary column switching matrices and satisfies $R_l^{-1} = R_l^T$ ($l = 1, 2$). Note $R_1 \in \mathbb{M}(\sum_k |I_k|, \sum_k |I_k|)$. Based on $\tilde{\mathcal{X}}_i$, we can define \tilde{C}_{11} , \tilde{C}_{21} and \tilde{C}_{22} similarly as above. The advantage of using $\tilde{\mathcal{X}}_i$ as the design matrix is that it orders the variables based on the grouping structures, and is in the form of the generic group lasso design in Basu et al. (2015). It is thus more straightforward to adapt their results using $\tilde{\mathcal{X}}_i$. Moreover, since each group of variables (j, g) corresponds to regression coefficients at the same (i, j) position across different models in g , the matrix \tilde{C}_{11} is in fact a block matrix, whose diagonal blocks are all identity matrices. To see this, consider

$g = \{k_1, k_2\}$, the columns of $\tilde{\mathcal{X}}_{i(1)}$ that correspond to the group (j, g) is

$$\mathbf{X}_j^g = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \vdots & \vdots \\ \mathbf{X}_j^{k_1} & \mathbf{0} \\ \mathbf{0} & \mathbf{X}_j^{k_2} \\ \vdots & \vdots \\ \mathbf{0} & \mathbf{0} \end{pmatrix}.$$

Hence the (j, g) -th diagonal block $(\tilde{C}_{11})_{[j,g]} = (\mathbf{X}_j^g)^T \mathbf{X}_j^g / n = I_2$.

With the above notations, the Uniform IC in Assumption (A3) is equivalent to saying for all $\xi = ((\xi^1)^T, \dots, (\xi^K)^T)^T \in \mathbb{R}^{\sum_k |I_k|}$ with $\max_{(j,g) \in J(\Theta_{0,i})} \|\xi_{[j,g]}\| \leq 1$ and all $(j, g) \notin J(\Theta_{0,i})$

$$\|[\tilde{C}_{21}(\tilde{C}_{11})^{-1}\tilde{\xi}]_{[j,g]}\| \leq 1 - \eta,$$

where $\tilde{\xi} = R_1^T \xi$.

It remains to select λ and α_n to ensure that the direction consistency results hold simultaneously for all i with probability tending to 1. For any $(j, g) \in J(\Theta_{0,i})$, denote $(\tilde{C}_{11})_{[j,g]}^{-1}$ the diagonal block in \tilde{C}_{11}^{-1} corresponding to the group (j, g) . By Theorem 4.1 of Basu et al. (2015), it suffices to find the upper bounds for $\|\tilde{C}_{11}^{-1}\|$, $\|(\tilde{C}_{11})_{[j,g]}^{-1}\|$, $\|(\tilde{C}_{22})_{[j,g]}\|$ and substitute the constant variance σ with the appropriate bound for $\text{Var}(\mathbf{X}_i^k | \mathbf{X}_{-i}^k) = 1/\omega_{0,ii}^k$ ($k = 1, \dots, K$).

By definition and the fact that the columns of \mathbf{X}^k are centered and standardized to have mean zero and unit variance, $(\tilde{C}_{11})_{[j,g]}$ is the identity matrix of size $|g| \times |g|$. It follows that

$$1 = \phi_{\min}^{-1}((\tilde{C}_{11})_{[j,g]}) \leq \phi_{\max}((\tilde{C}_{11})_{[j,g]}^{-1}) = \|(\tilde{C}_{11})_{[j,g]}^{-1}\| \leq \|(\tilde{C}_{11})^{-1}\|, \quad (25)$$

where the last step is obtained by applying Courant minimax principle since $0 \prec (\tilde{C}_{11})_{[j,g]}^{-1} \preceq (\tilde{C}_{11})^{-1}$. Similarly, for any $(j, g) \notin J(\Theta_{0,i})$, $(\tilde{C}_{22})_{[j,g]}$ is the identity matrix and

$$\|(\tilde{C}_{22})_{[j,g]}\| = 1. \quad (26)$$

Moreover, the variance for the random design in our problem

$$\text{Var}(\mathbf{X}_i^k | \mathbf{X}_{-i}^k) = 1/\omega_{0,ii}^k \leq 1/d_0, \quad \forall k, \quad (27)$$

by Assumption (A2).

It remains to find an upper bound for $\|\tilde{C}_{11}^{-1}\|$. Under Assumption (A1) with $s = s_0$, if we set $\Delta \in \mathcal{F}$ such that $\delta_j^{[g]} = \mathbf{0}$ for any $(j, g) \notin J(\Theta_{0,i})$, then

$$\frac{\sum_k \|\mathbf{X}^k \delta^{[g]}\|^2 / n}{\|\Delta_{J(\Theta_{0,i})}\|_{\mathcal{F}}^2} = \frac{\xi^T C_{11} \xi}{\xi^T \xi},$$

where $\xi = ((\xi^1)^T, \dots, (\xi^k)^T)^T \in \mathbb{R}^{\sum_k |E_i|}$ such that each ξ^k corresponds to the nonzero part of δ^k . If we choose Δ such that ξ is the eigenvector corresponding to the smallest eigenvalue of C_{11} , then

$$\kappa^2(s_0) \leq \frac{\sum_k \|\mathbf{X}^k \delta^k\|^2/n}{\|\Delta_{J(\Theta_{0,j})}\|_F^2} = \frac{\xi^T C_{11} \xi}{\xi^T \xi} = \phi_{\min}(C_{11}).$$

Since $R_1^{-1} = R_1^T$, C_{11} and \tilde{C}_{11} are similar (there exists a non-singular matrix P such that $P^{-1}C_{11}P = \tilde{C}_{11}$) and thus share the same set of eigenvalues. Therefore $\phi_{\min}(\tilde{C}_{11}) \geq \kappa^2(s_0)$ and

$$\|\tilde{C}_{11}^{-1}\| \leq \kappa^{-2}(s_0). \quad (28)$$

Combining the upper bounds in (25), (26), (27) and (28), Theorem 4.1 of Basu et al. (2015) implies that if we select λ and α_n as in (8) and (9), respectively, the direction consistency results follow by considering the union bound on all probabilities made across $i = 1, \dots, p$.

Further, if $\alpha_n < 1$, the direction consistency property of Θ_j implies exact recovery of all nonzero entries in the inverse covariance matrices, provided that the sparsity pattern \mathcal{G} is correctly specified. In other words, the set in (3) estimates correctly the true edge set E_0^k for all k .

The probability statement $1 - 4pC_0^{-1}e^{-q}$ follows from considering the union bound of the above result over all p regressions.

This completes the proof. ■

Appendix C. Additional Simulation Results

C.1 Performance with and without maximum likelihood refitting

In the main paper, we have compared the performance of different methods in estimating multiple Gaussian graphical models under optimally chosen tuning parameters with the results shown in Tables 1 and 2. All joint estimation methods were evaluated by adding the maximum likelihood refitting Step (II) for fair comparisons. To confirm that this is indeed the case, we present in the following additional simulation results for cases evaluated without the maximum likelihood refitting step.

Table 5 presents the complete table of deviance measures for various methods considered in simulation study 1. These methods include the separate estimation method Glasso, where the *Graphical Lasso* by Friedman et al. (2008) is applied to each graphical model separately, joint estimation by Guo et al. (2011), denoted by JEM-G, the Group Graphical Lasso denoted by GGL by Danaher et al. (2014), and the structural pursuit method MGGM by Zhu et al. (2014). For the latter three methods, we also present deviance measures for which the maximum likelihood refitting Step (II) is not included, denoted respectively by JEM-GI, GGL I and MGGM I. For the proposed two-step method JSEM, deviance measures based on Step I only is presented under the name JSEM I. It is clear from Table 5 that the refitting step generally does not introduce more errors in terms of structural estimation, but can significantly reduce the estimation errors in Frobenius norm.

Table 6 presents the performance of different regularization methods in estimating multiple inverse covariance matrices in simulation study 2. Here we observe similar pattern as that in Table 5, which confirms again that contribution from the maximum likelihood refitting step is mainly in reducing the Frobenius norm loss.

Method	FP	FN	SHD	F1	FL
Glasso	35(6)	81(2)	116(5)	0.32(0.02)	0.73(< 0.01)
JEM-GI	22(4)	40(4)	62(6)	0.69(0.03)	0.28(0.03)
JEM-G	22(4)	40(4)	62(6)	0.69(0.03)	0.28(0.02)
GGL I	18(7)	73(2)	91(7)	0.44(0.03)	0.70(0.01)
GGL	17(6)	73(2)	90(6)	0.44(0.03)	0.29(0.02)
MGGM I	291(14)	47(3)	339(14)	0.26(0.01)	0.69(0.02)
MGGM	286(13)	49(3)	335(13)	0.26(0.01)	0.64(0.02)
JSEM I	20(4)	34(3)	54(6)	0.73(0.03)	0.71(0.04)
JSEM	19(4)	35(3)	54(6)	0.73(0.03)	0.25(0.02)

Table 5: Performance of different regularization methods for estimating graphical models in Simulation Study 1: average FP, FN, SHD, F1 and FL (SE) for sample size $n_k = 50$. JEM-GI, GGL I, MGGM I and JSEM I correspond to respective method without the maximum likelihood refitting step. The best cases are highlighted in bold.

C.2 Performance as a function of p and n

Table 7 presents the performance of JSEM for $p = 500$ and $p = 1000$ with sample sizes n varying from 100, 200 to 500. The simulation setup is similar to that in Simulation Study 1: at each p , there are $K = 5$ graphical models sharing a *single* common structure. Individual structures with $p = 0.1$ are added to each graph separately such that about 10% of the edges in each graph are unique to themselves. It is clear that as the sample size n increases, the performance of JSEM also improves with smaller structural hamming distances (SHD), higher F1 score (F1) and smaller Frobenius norm loss (FL). In particular, the number of falsely rejected edges (FN) has decreased significantly.

Appendix D. Real Data Analysis

D.1 Climate data sources and pre-processing

The data we use in this study come from multiple sources and are collected under different resolutions for varying lengths of time periods. Specifically, the sources we consider include:

- (1) CRU: Climate Research Unit provides monthly climatology data (<http://www.cru.uea.ac.uk/cru/data/>) for 10 surface variables including mean temperature (TMP), diurnal temperature range (DTR), maximum and minimum temperature (TMX, TMN), precipitation (PRE), vapor pressure (VAP), cloud cover (CLD), rainy days counts (WET), potential evapotranspiration (PET) and frost days (FRS) from 1901 to 2013 at the 0.5 degree latitude and longitude resolution. Note these high-resolution gridded data sets are constructed using not only directly observed data, but also derived and estimated values with well-known formulae whenever the observed data are not available (see details in Harris et al., 2014).
- (2) NASA: The Goddard Earth Sciences Data and Information Services Center (GES DISC) from the National Aeronautics and Space Administration (NASA) has collected aerosol measurements using Moderate Resolution Imaging Spectroradiometer (MODIS) on satellites. The

ρ	Method	FP	FN	SHD	F1	FL
0	Glasso	154(4)	38(1)	192(4)	0.51(0.01)	0.60(0.005)
	JEM-G1	87(2)	36(2)	123(3)	0.62(0.01)	0.41(0.01)
	JEM-G	86(3)	36(2)	122(3)	0.62(0.01)	0.31(0.01)
	GGL1	152(3)	38(1)	191(4)	0.51(0.01)	0.60(0.01)
	GGL	144(3)	39(1)	184(4)	0.52(0.01)	0.37(0.01)
	MGGM1	30(2)	67(1)	97(2)	0.59(0.01)	0.37(0.01)
	MGM	30(2)	67(1)	97(2)	0.59(0.01)	0.36(0.01)
	JSEM1	22(2)	42(2)	64(3)	0.75(0.01)	0.68(0.01)
	JSEM	21(2)	42(2)	63(3)	0.75(0.01)	0.28(0.01)
	0.2	Glasso	164(3)	47(1)	211(4)	0.53(0.01)
JEM-G1		94(3)	57(2)	151(3)	0.59(0.01)	0.44(0.01)
JEM-G		92(3)	57(2)	149(3)	0.59(0.01)	0.35(0.01)
GGL1		163(3)	47(1)	210(4)	0.53(0.01)	0.59(0.005)
GGL		155(3)	48(1)	203(3)	0.53(0.01)	0.37(0.01)
MGGM1		98(4)	63(1)	161(4)	0.56(0.01)	0.38(0.01)
MGM		94(3)	64(1)	158(4)	0.56(0.01)	0.37(0.01)
JSEM1		33(3)	64(2)	97(3)	0.67(0.01)	0.77(0.01)
JSEM		32(3)	64(2)	96(3)	0.67(0.01)	0.32(0.01)
0.4		Glasso	159(3)	59(1)	218(4)	0.55(0.01)
	JEM-G1	101(3)	77(2)	178(3)	0.56(0.01)	0.45(0.01)
	JEM-G	100(3)	77(2)	177(3)	0.56(0.01)	0.37(0.01)
	GGL1	158(3)	60(1)	218(4)	0.55(0.01)	0.57(0.005)
	GGL	149(3)	61(2)	210(4)	0.55(0.01)	0.37(0.01)
	MGGM1	122(3)	65(1)	187(3)	0.57(0.01)	0.38(0.01)
	MGM	119(3)	65(1)	184(3)	0.58(0.01)	0.37(0.01)
	JSEM1	50(3)	83(2)	133(3)	0.62(0.01)	0.84(0.01)
	JSEM	49(3)	84(2)	132(3)	0.62(0.01)	0.36(0.01)
	0.6	Glasso	176(4)	73(2)	249(4)	0.54(0.01)
JEM-G1		95(3)	109(2)	204(4)	0.52(0.01)	0.45(0.01)
JEM-G		94(3)	109(2)	203(3)	0.52(0.01)	0.39(0.01)
GGL1		174(4)	74(2)	248(4)	0.54(0.01)	0.55(0.01)
GGL		165(4)	76(2)	241(4)	0.54(0.01)	0.39(0.01)
MGGM1		113(3)	94(2)	207(4)	0.55(0.01)	0.39(0.01)
MGM		109(3)	95(2)	204(4)	0.55(0.01)	0.39(0.01)
JSEM1		52(3)	122(2)	174(4)	0.53(0.01)	0.89(0.01)
JSEM		50(3)	123(2)	173(4)	0.52(0.01)	0.38(0.01)

Table 6: Performance of different regularization methods for estimating graphical models in Simulation Study 2: average FP, FN, SHD, F1 and FL (SE) for sample size $n_k = 100$. JEM-G1, GGL1, MGGM1 and JSEM1 correspond to respective method without the maximum likelihood refitting step. The best cases are highlighted in bold.

p	n	FP	FN	SHD	F1	FL
500	100	20(4)	179(8)	200(9)	0.79(0.01)	0.21(0.01)
	200	40(5)	50(3)	90(6)	0.92(0.005)	0.14(0.01)
	500	49(3)	36(1)	85(3)	0.92(0.002)	0.09(0.003)
1000	100	17(4)	643(14)	661(15)	0.58(0.01)	0.23(0.005)
	200	34(5)	187(9)	221(10)	0.89(0.005)	0.14(0.005)
	500	77(6)	80(2)	157(5)	0.93(0.002)	0.10(0.003)

Table 7: Performance of JSEM as a function of p and n : average FP, FN, SHD, F1 and FL (SE). The setup is similar to that in Simulation Study 1.

data set obtained from Terra satellite consists of monthly average aerosol optical depth (AER) at the 1 degree latitude by 1 degree longitude resolution from March 2000 to August 2014.

- (3) NCDC: The National Solar Radiation Database (NSRDB) 1991–2010 (a collaborative project between The National Renewable Energy Laboratory (NREL) and the National Climatic Data Center (NCDC)) provides statistical summaries for solar data (<ftp://ftp.ncdc.noaa.gov/pub/data/nsrdb-solar/>) from 860 different locations across the United States. The locations are recorded using their latitude, longitude and altitude. We used measurements for global horizontal radiation (SOL) at 242 class I stations that have high-quality data.
 - (4) NOAA: The climate data center of National Oceanic and Atmospheric Administration (NOAA) has archived the trace gases data, including carbon dioxide (CO₂), carbon monoxide (CO), methane (CH₄) and hydrogen (H₂), from 170 worldwide stations (<http://www.esrl.noaa.gov/gmd/dv/ftpdata.html>). These data sets consist of measurements spanning different time periods, with CO₂ ranging from 1968 to 2013 (the longest) and H₂ from 1992 to 2005 (the shortest). In addition, they come with relatively low resolution compared to other variables due to the limited number of stations.
- To ensure compatibility and consistency among multiple data sources, we performed the following pre-processing:
- (1) Normalization: We first transformed each data set into monthly observations in a standard format including longitude, latitude, altitude (when available), date, variable, value, unit, and source. We focus on a 54-month time period from January 2001 to June 2005 where data for all variables are available.
 - (2) Interpolation and smoothing: We interpolated the monthly data from NCDC and NOAA onto a common 2.5 by 2.5 degree grid for North America using thin plate splines. Since the data from CRU and NASA were provided for a finer resolution grid, thin plate splines were used to first interpolate the data onto a grid of the same resolution as the source data. Then we performed spatial averaging to get data on the common 2.5 by 2.5 degree grid.
 - (3) Seasonality and autocorrelation: We reduced the short-term autocorrelation by aggregating the time series for each variable at each location into bins of 3-month intervals and taking

first differences on the quarterly data. The resulting data, consisting of 17 measurements, are assumed to be independent samples for the corresponding variable at the specified location.

The final data are organized as an $n \times p$ matrix at each of the 27 locations considered, where $n = 17$ and $p = 16$.

D.2 Additional results in climate modeling

The inferred networks at the six distinct climate zones using JSEM are presented in Section 5. The estimated networks at the 27 distinct locations are also presented in Figure 8 for reference. For notational convenience, we have renamed the climate zones such that BW for Midlatitude Desert, Cfa for Humid Subtropical, Bsh for Semiarid Steppe (hot arid), Bsk for Semiarid Steppe (cold arid), Dfa for Humid Continental (hot summer), and Dfb for Humid Continental (warm summer).

The networks in Figure 8 are ordered such that those in the same row belong to the same climate zone; further, networks in the third and fourth rows represent those from the Semiarid Steppe group, whereas networks in the last two rows are all from the Humid Continental group. Such an ordering respects how the structural information is defined and helps visualize similarities across networks. Indeed, by comparing networks at locations from geographical south, that is networks entitled ‘South’, we notice that the interactions between AER, SOL, and the remaining variables are very similar. For example, almost all of them share the edges AER—SOL and SOL—H2, except at two locations ‘BW South desert 3’ and ‘BW South desert 7’. In contrast, networks from the geographical north all share the edges AER—H2 and SOL—H2. Further, the interactions between variables on greenhouse gases (CO2, CO, CH4 and H2) and others have four distinct patterns at the four distinct climate groups. For example, greenhouse gases interact with VAP for the desert group, whereas they interact with CLD in the subtropical group. The partial correlation between CLD and greenhouse gases at subtropical climate makes sense because such humid areas are more likely to be cloudy, thereby influencing the concentration of CO2 (Graham et al., 2003). Finally, variables excluding AER, SOL, and those on greenhouse gases show distinct interaction patterns with others at the six distinct climate zones. In particular, one can see that the variable FRS interacts mainly with PET at Desert and Steppe climate, whereas it is partially correlated with both PRE and TMN (or TMX) at Continental climate. This can be explained from the distinction between these climate zones: At Humid Continental climate, precipitation is relatively well distributed year-round in most areas and snowfall occurs in all areas. It is thus not difficult to see why precipitation (PRE) and temperature related variables correlate with the number of frost days (FRS). Further, a primary criterion of an area being Midlatitude Desert or Semiarid Steppe is that it receives precipitation below potential evapotranspiration (PET), which possibly explains why FRS is partially correlated only with PET for Desert and Steppe climate. We also point out that networks at adjacent climate zones are very similar. For example, networks at ‘Bsh Steppe’ and ‘Bsk Steppe’ share similar topologies. As a comparison, we also applied other joint estimation methods JEM-G, GGL and MGGM on the same data set. For each of the three methods considered here, we used BIC on the normalized data to select the optimal tuning parameters and coupled each method with complementary pairs stability selection (Shah and Samworth, 2013) to infer the related climate networks. As in the case of JSEM, we run each method 50 times on two randomly drawn complementary pairs of size 8 and 9 and kept only edges that are selected above a certain threshold. The selection probability used for JSEM is 70%. However, as the second simulation study indicates JEM-G and GGL tend to produce higher false positives, especially GGL, we increased the probability threshold for JEM-G and GGL

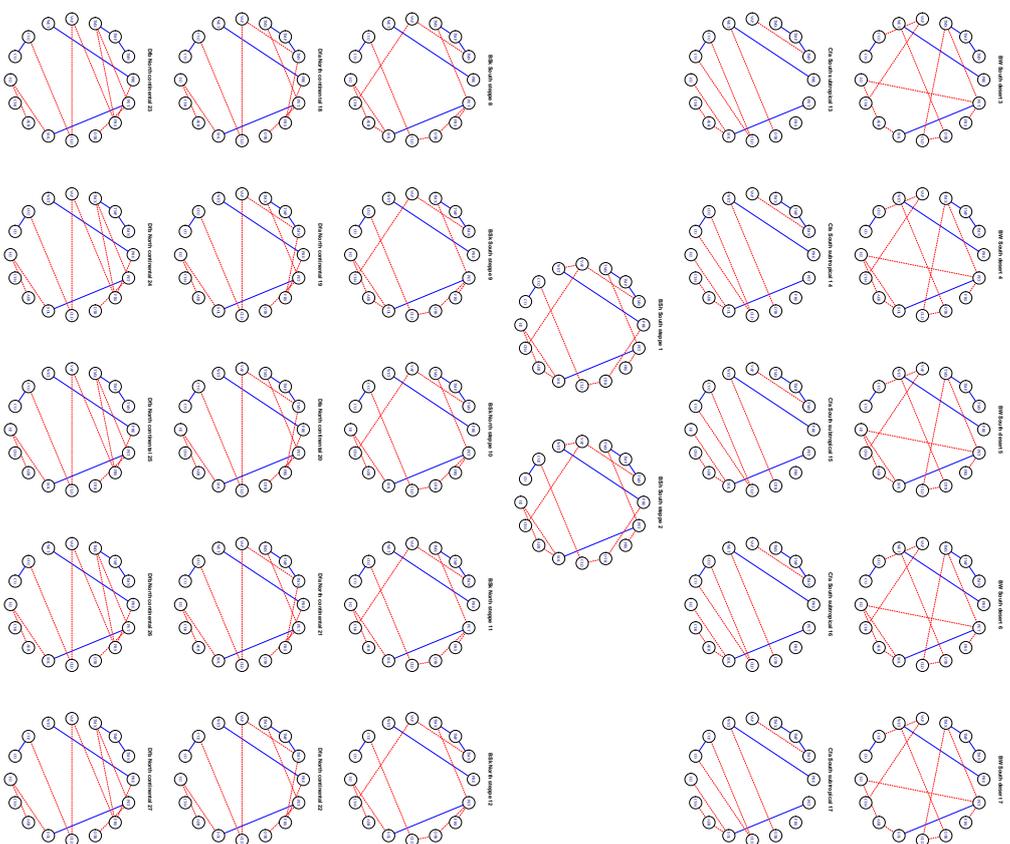


Figure 8: Estimated climate networks at the 27 locations using JSEM, with edges shared across all locations solid and differential edges dashed.

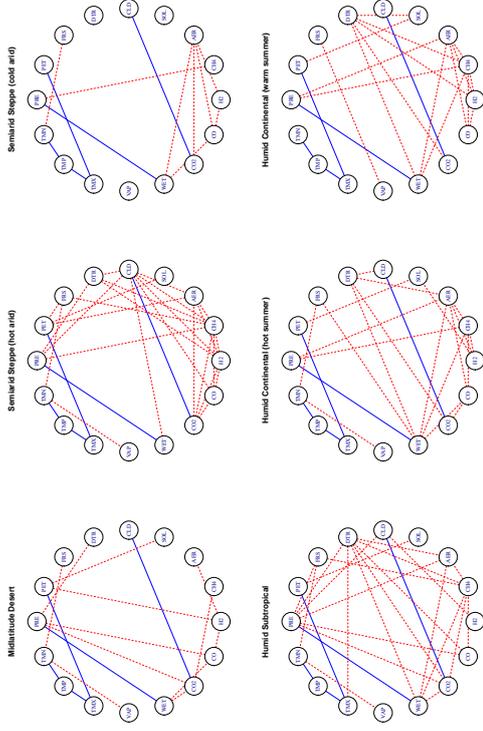


Figure 9: Estimated climate networks at the six distinct climate zones using JEM-G, with edges shared across all locations solid and differential edges dashed.

to 90% and 100%, respectively. On the other hand, we reduced the threshold for MGGM to 50% due to the relatively few edges recovered. The results are shown in Figure 9, 10 and 11.

One can see clearly that the estimated networks using the three methods exhibit quite different connectivity patterns from those inferred from JSEM. In particular, the results from GGL seem to suggest strong conditional dependence among a subset of variables, which distinguishes itself from JEM-G and JSEM. The estimated networks using MGGM, though sparse, bear certain similarity to those recovered using GGL. On the other hand, the results from JEM-G and JSEM are more similar. For example, common edges identified using JEM-G, such as TMN—TMP, TMP—TMX, PRE—WET, also show up under JSEM. The common edge between CLD and CO2 is found at all locations except Midlatitude Desert under JSEM, whereas the edge between PET and SOL identified using JSEM exists everywhere except at Semiarid Steppe (cold arid) under JEM-G. Note although JEM-G does not require external information on the structural relationships across graphs, the inferred networks respect roughly the spatial pattern of all climate zones. For instance, Humid Continental (hot summer and cool summer) are more similar.

D.3 Additional results in analysis of breast cancer

We have presented the inferred pathway level networks under both ER+ and ER- status using JSEM in the main paper. As a comparison, we applied JEM-G with tuning parameters selected via BIC to the same normalized and processed data set. To ensure a stable estimation, we further

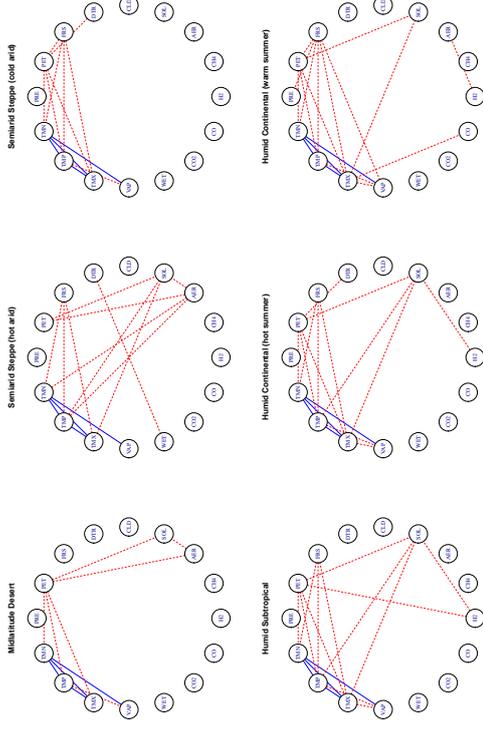


Figure 10: Estimated climate networks at the six distinct climate zones using GGL, with edges shared across all locations solid and differential edges dashed.

coupled JEM-G with complementary pairs stability selection (Shah and Samworth, 2013). Figure 12 shows the pathway level interactions estimated from JEM-G at selection frequency 70%, after removing isolated pathways. One striking difference between Figure 12 and Figure 7 is that Figure 12 sees more edges shared across the two classes (in blue). This is partly due to how JEM-G is implemented directly via the sample covariance matrices and partly to JEM-G being an agnostic method.

We also present estimated gene-level networks (with isolated genes removed) using both JEM-G and JSEM in Figure 13. Similar to what we observe in the pathway level network comparison, the JEM-G recovered gene networks show more edges shared between the two classes. In comparison, JSEM recovers more differential edges for the ER+ class. Apart from the differences, we also observe some similarities between the estimated gene networks. For example, both methods identify a small hub around the gene SFRP1, indicating their potential in regulating the underlying biological process.

Appendix E. Additional Technical Details

We include here some additional lemmas and proofs necessary for establishing the theoretical results in Section 3.

The first lemma is borrowed from Basu et al. (2015, Lemma A.2). We state the result here for completeness. Please refer to their paper for proof of the lemma.

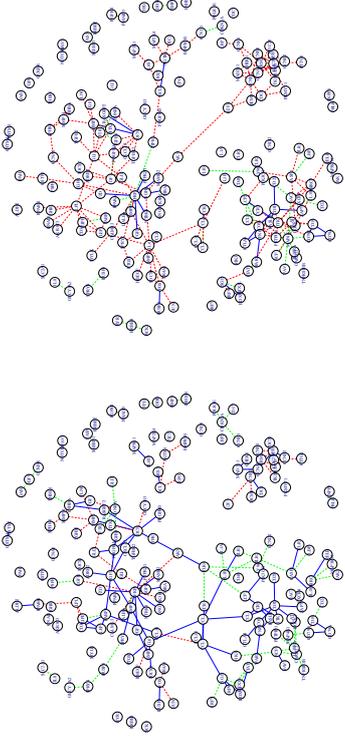


Figure 13: Estimated gene networks combined using JEM-G (left) and JSEM (right), with edges shared between ER+ and ER- blue solid and differential edges red dashed (ER+)/ green dashed (ER-).

Applying Lemma E.1,

$$\begin{aligned} \mathbb{P}(\{\mathcal{A}_{ij}^g\}^c) &\leq \mathbb{P}(\|Z^{[g]}\| - E\|Z^{[g]}\| > \sqrt{n}\lambda_{ij}^g/2 - E\|Z^{[g]}\|) \\ &\leq 2 \exp\left\{-\frac{2}{\pi^2\|\text{Var}(Z^{[g]})\|}\left(\frac{\sqrt{n}\lambda_{ij}^g}{2} - E\|Z^{[g]}\|\right)^2\right\}. \end{aligned}$$

Choose λ_{ij}^g such that the right-hand side of above inequality is less than $2G_0^{-q}$ for some positive parameter q . Then

$$\lambda_{ij}^g \geq \frac{2}{\sqrt{n}} \left(E\|Z^{[g]}\| + \frac{\pi}{\sqrt{2}} \sqrt{q \log G_0} \sqrt{\|\text{Var}(Z^{[g]})\|} \right),$$

and is satisfied if

$$\lambda_{ij}^g \geq \max_{k \in g} \frac{2}{\sqrt{n\omega_{0,ii}^k}} \left(\sqrt{|g|} + \frac{\pi}{\sqrt{2}} \sqrt{q \log G_0} \right),$$

by Lemma E.1. With the above choice of λ_{ij}^g ,

$$\mathbb{P}(\mathcal{A}^c) \leq \sum_{i=1}^p \sum_{j \neq i} \sum_{g \in \mathcal{G}^{ij}} \mathbb{P}(\{\mathcal{A}_{ij}^g\}^c) \leq 2pG_0^{1-q},$$

or equivalently, $\mathbb{P}(\mathcal{A}) \geq 1 - 2pG_0^{1-q}$. \blacksquare

Proof of Proposition A.1

Proof For all $\Theta_i \in \mathbb{M}(p-1, K)$, using a similar argument to that in Lemma 3.1 of Lounici et al. (2011), it is straightforward to verify the following:

$$\begin{aligned} &\sum_{k=1}^K \frac{1}{n} \|\mathbf{X}_{-i}^k(\hat{\theta}_i^k - \theta_{0,i}^k)\|^2 + \sum_{j \neq i, g \in \mathcal{G}^{ij}} \lambda_{ij}^g \|\hat{\theta}_{ij}^{[g]} - \theta_{ij}^{[g]}\| \\ &\leq \sum_{k=1}^K \frac{1}{n} \|\mathbf{X}_{-i}^k(\theta_i^k - \theta_{0,i}^k)\|^2 + 4 \sum_{(j,g) \in \mathcal{I}(\Theta_i)} \lambda_{ij}^g \min\left(\|\theta_{ij}^{[g]}\|, \|\hat{\theta}_{ij}^{[g]} - \theta_{ij}^{[g]}\|\right), \end{aligned} \quad (30)$$

$$\left\{ \sum_{k \in \mathcal{G}^g} (n^{-1} \mathbf{X}_j^k, \mathbf{X}_{-i}^k(\theta_i^k - \theta_{0,i}^k))^2 \right\}^{1/2} \leq \frac{3\lambda_{ij}^g}{2}, \quad (31)$$

$$\mathcal{M}(\hat{\Theta}_i) \leq \frac{4\phi_{\max}}{\lambda_{\min}^2} \sum_{k=1}^K \frac{1}{n} \|\mathbf{X}_{-i}^k(\hat{\theta}_i^k - \theta_{0,i}^k)\|^2, \quad (32)$$

where λ_{\min} and ϕ_{\max} are defined in Proposition A.1.

Let $\Delta = (\delta^1, \dots, \delta^K)$ be a matrix in $\mathbb{M}(p, K)$ such that $\delta_j^k = \hat{\theta}_{ij}^k - \theta_{0,ij}^k$ for $j \neq i$ and $\delta_i^k = 0$ for all k . We would like to first find an upper bound for B^2 , where

$$B^2 := \sum_{k=1}^K \frac{1}{n} \|\mathbf{X}_{-i}^k(\hat{\theta}_i^k - \theta_{0,i}^k)\|^2 = \sum_{k=1}^K \frac{1}{n} \|\mathbf{X}^k \delta^k\|^2.$$

On the event \mathcal{A} , we have

$$\sum_{j \neq i, g \in \mathcal{G}^{ij}} \lambda_{ij}^g \|\delta_j^{[g]}\| \leq B^2 + \sum_{j \neq i, g \in \mathcal{G}^{ij}} \lambda_{ij}^g \|\delta_j^{[g]}\| \leq 4 \sum_{(j,g) \in \mathcal{I}(\Theta_{0,i})} \lambda_{ij}^g \|\delta_j^{[g]}\|, \quad (33)$$

where the second inequality follows from setting $\Theta_i = \Theta_{0,i}$ in (30). Therefore

$$\sum_{(j,g) \in \mathcal{I}(\Theta_{0,i})^c} \lambda_{ij}^g \|\delta_j^{[g]}\| \leq 3 \sum_{(j,g) \in \mathcal{I}(\Theta_{0,i})} \lambda_{ij}^g \|\delta_j^{[g]}\|,$$

which implies that $\Delta \in \mathcal{F}$, the restricted set defined in Assumption (A1). Under Assumption (A1) with $\kappa = \kappa(s_0)$, one has

$$B^2 \geq \kappa^2 \|\Delta_J\|_{\mathcal{F}}^2 = \kappa^2 \sum_{(j,g) \in \mathcal{I}(\Theta_{0,i})} \|\delta_j^{[g]}\|^2. \quad (34)$$

Combing (33) and the Cauchy-Schwarz inequality, we obtain

$$B^2 \leq 4 \sum_{(j,g) \in \mathcal{I}(\Theta_{0,i})} \lambda_{ij}^g \|\delta_j^{[g]}\| \leq 4 \left\{ \sum_{(j,g) \in \mathcal{I}(\Theta_{0,i})} (\lambda_{ij}^g)^2 \right\}^{1/2} \left\{ \sum_{(j,g) \in \mathcal{I}(\Theta_{0,i})} \|\delta_j^{[g]}\|^2 \right\}^{1/2} \quad (35)$$

$$\leq 4 \left\{ \sum_{(j,g) \in \mathcal{I}(\Theta_{0,i})} (\lambda_{ij}^g)^2 \right\}^{1/2} \frac{B}{\kappa}, \quad (36)$$

where the last inequality in (36) comes from (34). Canceling out the extra B in (36), we get

$$B^2 = \sum_k \frac{1}{n} \|\mathbf{X}_{0,i}^k(\hat{\theta}_i^k - \theta_{0,i}^k)\|^2 \leq \frac{16}{\kappa^2} \sum_{(j,g) \in I(\Theta_{0,i})} (\lambda_{ij}^g)^2. \quad (37)$$

To show the inequality in (11), we note by (33), the Cauchy-Schwarz inequality, (34) and (37),

$$\begin{aligned} \sum_{j \neq i} \sum_{g \in \mathcal{G}^{ij}} \|\delta_j^{[g]}\| &\leq \frac{1}{\lambda_{\min}} \sum_{j \neq i} \sum_{g \in \mathcal{G}^{ij}} \lambda_{ij}^g \|\delta_j^{[g]}\| \leq \frac{4}{\lambda_{\min}} \sum_{(j,g) \in I(\Theta_{0,i})} \lambda_{ij}^g \|\delta_j^{[g]}\| \\ &\leq \frac{4}{\lambda_{\min}} \left\{ \sum_{(j,g) \in I(\Theta_{0,i})} \|\delta_j^{[g]}\|^2 \right\}^{1/2} \left\{ \sum_{(j,g) \in I(\Theta_{0,i})} (\lambda_{ij}^g)^2 \right\}^{1/2} \\ &\leq \frac{4}{\lambda_{\min}} \frac{B}{\kappa} \left\{ \sum_{(j,g) \in I(\Theta_{0,i})} (\lambda_{ij}^g)^2 \right\}^{1/2} \\ &\leq \frac{16}{\kappa^2 \lambda_{\min}} \sum_{(j,g) \in I(\Theta_{0,i})} (\lambda_{ij}^g)^2. \end{aligned}$$

(12) follows readily from (32) and (37)

$$\mathcal{M}(\hat{\Theta}_i) \leq \frac{4\phi_{\max}}{\lambda_{\min}^2} B^2 \leq \frac{64\phi_{\max}}{\kappa^2 \lambda_{\min}^2} \sum_{(j,g) \in I(\Theta_{0,i})} (\lambda_{ij}^g)^2.$$

Finally, we prove (13). Let $J_0 = J(\Theta_{0,i})$ and J_1 denote the set of indices in J_0^c corresponding to the s_i largest values of $\lambda_{ij}^g \|\delta_j^{[g]}\|$. The dependence of J_0 and J_1 on i is made implicit here for clarity. Let $J_{01} = J_0 \cup J_1$. So $|J_{01}| \leq 2s_i$. Let (j_ℓ, g_ℓ) be the index of the ℓ th largest element of the set $\{\lambda_{ij}^g \|\delta_j^{[g]}\| : (j, g) \in J_0^c\}$. Then

$$\lambda_{j_\ell g_\ell}^{g_\ell} \|\Delta_{j_\ell}^{[g_\ell]}\| \leq \sum_{(j,g) \in J_0^c} \frac{\lambda_{ij}^g \|\delta_j^{[g]}\|}{\ell}.$$

Combining with the fact that $\Delta \in \mathcal{F}$, we have on the event \mathcal{A}_1 ,

$$\begin{aligned} \sum_{(j,g) \in I_0^c} \left(\lambda_{ij}^g \|\delta_j^{[g]}\| \right)^2 &\leq \sum_{(j,g) \in I_0^c} \left(\lambda_{j_\ell g_\ell}^{g_\ell} \|\delta_{j_\ell}^{[g_\ell]}\| \right)^2 \leq \sum_{\ell=s_i+1}^{\infty} \frac{\left(\sum_{(j,g) \in I_0^c} \lambda_{ij}^g \|\delta_j^{[g]}\| \right)^2}{\ell^2} \\ &\leq \frac{1}{s_i} \left(\sum_{(j,g) \in I_0^c} \lambda_{ij}^g \|\delta_j^{[g]}\| \right)^2 \\ &\leq \frac{9}{s_i} \left(\sum_{(j,g) \in I_0} \lambda_{ij}^g \|\delta_j^{[g]}\| \right)^2 \\ &\leq \frac{9}{s_i} \sum_{(j,g) \in I_0} (\lambda_{ij}^g)^2 \|\Delta_{j_0}\|_F^2 \\ &\leq \frac{9}{s_i} \sum_{(j,g) \in J_0} (\lambda_{ij}^g)^2 \|\Delta_{j_0}\|_F^2, \end{aligned} \quad (38)$$

where (38) comes from the Cauchy-Schwarz inequality. It follows immediately that

$$\lambda_{\min}^2 \sum_{(j,g) \in I_0^c} \|\delta_j^{[g]}\|^2 \leq \frac{9}{s_i} \sum_{(j,g) \in J_0} (\lambda_{ij}^g)^2 \|\Delta_{j_0}\|_F^2.$$

Hence

$$\begin{aligned} \|\hat{\Theta}_i - \Theta_{0,i}\|_F^2 &= \sum_{j \neq i} \sum_{g \in \mathcal{G}^{ij}} \|\delta_j^{[g]}\|^2 = \|\Delta_{j_0}\|_F^2 + \|\Delta_{j_0^c}\|_F^2 \\ &\leq \|\Delta_{j_0}\|_F^2 + \frac{9}{s_i \lambda_{\min}^2} \sum_{(j,g) \in J_0} (\lambda_{ij}^g)^2 \|\Delta_{j_0}\|_F^2 \\ &\leq \frac{10}{s_i \lambda_{\min}^2} \sum_{(j,g) \in J_0} (\lambda_{ij}^g)^2 \|\Delta_{j_0}\|_F^2. \end{aligned} \quad (39)$$

Now we bound $\|\Delta_{j_0}\|_F$. Note (35) implies that

$$B^2 \leq 4 \left\{ \sum_{(j,g) \in J_0} (\lambda_{ij}^g)^2 \right\}^{1/2} \|\Delta_{j_0}\|_F \leq 4 \left\{ \sum_{(j,g) \in J_0} (\lambda_{ij}^g)^2 \right\}^{1/2} \|\Delta_{j_0}\|_F.$$

Further we have $B^2 \geq \kappa^2 (2s_0) \|\Delta_{j_0}\|_F^2$ under Assumption (A1) with $s = 2s_0$. So

$$\|\Delta_{j_0}\|_F^2 \leq \frac{B^2}{\kappa^2 (2s_0)} \leq \frac{4}{\kappa^2 (2s_0)} \left\{ \sum_{(j,g) \in J_0} (\lambda_{ij}^g)^2 \right\}^{1/2} \|\Delta_{j_0}\|_F,$$

which implies

$$\|\Delta_{j_0}\|_F \leq \frac{4}{\kappa^2 (2s_0)} \left\{ \sum_{(j,g) \in J_0} (\lambda_{ij}^g)^2 \right\}^{1/2}. \quad (40)$$

Plugging the bound in (40) into (39), we obtain

$$\|\hat{\Theta}_i - \Theta_{0,i}\|_F^2 \leq \left\{ \frac{4\sqrt{10}}{\kappa^2 (2s_0)} \right\}^2 \left\{ \sum_{(j,g) \in J_0} (\lambda_{ij}^g)^2 \right\}^2,$$

or equivalently

$$\|\hat{\Theta}_i - \Theta_{0,i}\|_F \leq \frac{4\sqrt{10}}{\kappa^2 (2s_0)} \sum_{(j,g) \in I(\Theta_{0,i})} (\lambda_{ij}^g)^2 \lambda_{\min} \sqrt{s_i}.$$

■

Proof of Corollary A.1

Proof By definition, $\omega_{0,ij}^k = -\theta_{0,ij}^k \omega_{0,ii}^k$ for all $j \neq i$ and $k = 1, \dots, K$. Further, under Assumption (A2), $\omega_{0,ii}^k \leq \phi_{\max}(\Omega_0^k) = \phi_{\min}^{-1}(\Sigma_0^k) \leq c_0$ for all i, k . Therefore

$$\begin{aligned} \sum_k \|\tilde{\Omega}^k - \Omega_0^k\|_F^2 &= \sum_k \sum_{i=1}^p \sum_{j \in J(\Theta_{0,i}) \cap J(\hat{\Theta}_i)^c} (\theta_{0,ij}^k \omega_{0,ii}^k)^2 \\ &= \sum_{i=1}^p \sum_{j \in J(\Theta_{0,i}) \cap J(\hat{\Theta}_i)^c} \sum_{g \in \mathcal{G}^{ij}} (\theta_{0,ij}^k \omega_{0,ii}^k)^2 \\ &\leq c_0^2 \sum_{i=1}^p \sum_{j \in J(\Theta_{0,i}) \cap J(\hat{\Theta}_i)^c} \sum_{g \in \mathcal{G}^{ij}} \|\theta_{0,ij}^g\|^2 \\ &\leq c_0^2 \sum_{i=1}^p \sum_{j \neq i} \sum_{g \in \mathcal{G}^{ij}} \|\theta_{0,ij}^g\|^2. \end{aligned}$$

Under Assumption (A1) with $s = 2s_0$, applying Proposition A.1 with $\lambda_{ij}^g = \lambda_{\max}$ in (14),

$$\sum_{\substack{j \neq i \\ g \in \mathcal{G}^{ij}}} \|\theta_{0,ij}^g - \hat{\theta}_{0,ij}^g\|^2 \leq \left\{ \frac{4\sqrt{10}}{r^2(2s_0)} \lambda_{\max} \right\}^2 s_i.$$

Therefore,

$$\sum_k \|\tilde{\Omega}^k - \Omega_0^k\|_F^2 \leq \left\{ \frac{4\sqrt{10}c_0}{r^2(2s_0)} \lambda_{\max} \right\}^2 \sum_{i=1}^p s_i = \left\{ \frac{4\sqrt{10}c_0}{r^2(2s_0)} \lambda_{\max} \right\}^2 S_0.$$

It follows immediately that

$$\begin{aligned} \frac{1}{K} \sum_k \|\tilde{\Omega}^k - \Omega_0^k\|_F &\leq \frac{1}{\sqrt{K}} \left\{ \sum_k \|\tilde{\Omega}^k - \Omega_0^k\|_F^2 \right\}^{1/2} \leq \frac{4\sqrt{10}c_0}{r^2(2s_0)} \lambda_{\max} \sqrt{\frac{S_0}{K}} \\ &\leq C_{\text{bias}} \sqrt{\frac{S_0}{nK}} \left(\sqrt{g_{\max}} + \frac{\pi}{\sqrt{2}} \sqrt{q \log C_0} \right). \end{aligned}$$

To bound the size of the estimated edge set \hat{E}^k , we notice if there exists (i, j, k) such that $\hat{\theta}_{ij}^k \neq 0$, then $\hat{\theta}_{ij}^{[g]} \neq 0$, where $g \ni k$. Hence $\mathcal{M}(\hat{\theta}_i^k) \subseteq \mathcal{M}(\hat{\Theta}_i)$ for all k . By (12), the upper bound for \hat{E}^k is thus

$$|\hat{E}^k| \leq \sum_{i=1}^p \mathcal{M}(\hat{\theta}_i^k) \leq \sum_{i=1}^p \frac{64\phi_{\max}}{r^2(s_0)} \lambda_{\min}^2 \sum_{(j,g) \in J(\Theta_{0,i})} s_i \leq \frac{64\phi_{\max}}{r^2(s_0)} S_0. \quad \blacksquare$$

References

- Francis Bach. Consistency of the group lasso and multiple kernel learning. *Journal of Machine Learning Research*, 9:1179–1225, 2008.
- Sumanta Basu, Ali Shojaie, and George Michailidis. Network granger causality with inherent grouping structure. *Journal of Machine Learning Research*, 16:417–453, 2015.
- Peter J. Bickel, Ya'Acov Ritov, and Alexandre B. Tsybakov. Simultaneous analysis of lasso and dantzig selector. *Annals of Statistics*, 37(4):1705–1732, 2009.
- Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.
- Patrick Breheny and Jian Huang. Penalized methods for bi-level variable selection. *Statistics and Its Interface*, 2(3):369–380, 2009.
- Peter Bühlmann and Sara van de Geer. *Statistics for High-Dimensional Data*. Springer-Verlag Berlin Heidelberg, 2011.
- Julien Chiquet, Yves Grandvalet, and Christophe Ambroise. Inferring multiple graphical structures. *Statistics and Computing*, 21(4):537–553, 2011.
- Patrick Danaher, Pei Wang, and Daniela M Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(2):373–397, 2014.
- Matthias Dehmer and Frank Emmert-Streib. *Analysis of Microarray Data: A Network-Based Approach*. John Wiley & Sons, 2008.
- David Edwards. *Introduction to Graphical Modelling*. Springer New York, 2000.
- Jacques Ferlay, Isabelle Soerjomataram, M Ervik, Rajesh Dikshit, Sultan Eser, Colin Mathers, Marise Rebelo, Donald Maxwell Parkin, David Forman, and Freddie Bray. *GLOBOCAN 2012 v1.0. Cancer Incidence and Mortality Worldwide: IARC CancerBase No. 11 [Internet]*. Lyon, France: International Agency for Research on Cancer, 2013.
- Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.
- Eric A. Graham, Stephen S. Mulkey, Kaoru Kitajima, Nathan G. Phillips, and S. Joseph Wright. Cloud cover limits net co2 uptake and growth of a rainforest tree during tropical rainy seasons. *Proceedings of the National Academy of Sciences*, 100(2):572–576, 2003.
- Jian Guo, Elizaveta Levina, George Michailidis, and Ji Zhu. Joint estimation of multiple graphical models. *Biometrika*, 98(1):1–15, 2011.
- I. Harris, P.D. Jones, T.J. Osborn, and D.H. Lister. Updated high-resolution grids of monthly climatic observations—the cru ts3.10 dataset. *International Journal of Climatology*, 34(3):623–642, 2014.

- Jean Honorio and Dimitris Samaras. Multi-task learning of gaussian graphical models. In *International Conference on Machine Learning (ICML)*, pages 447–454, 2010.
- Akash K. Kaushik, Ali Shojate, Katrin Panzitz, Rajni Sonavane, Harene Venghatakishnan, Mohan Manikkam, Alexander Zaslavsky, Vasanta Puturi, Vihass T. Vasu, Yiqing Zhang, et al. Inhibition of the hexamine biosynthetic pathway promotes castration-resistant prostate cancer. *Nature Communications*, 7, 2016.
- Stefien L. Lauritzen. *Graphical Models*. Oxford University Press, New York, 1996.
- Karim Lounici, Massimiliano Pontil, Sara van de Geer, and Alexandre B. Tsybakov. Oracle inequalities and optimal inference under group sparsity. *Annals of Statistics*, 39(4):2164–2204, 2011.
- Aurelie C Lozano, Hongfei Li, Alexandru Niculescu-Mizil, Yan Liu, Claudia Perlich, Jonathan Hosking, and Naoki Abe. Spatial-temporal causal modeling for climate change attribution. In *SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 587–596. ACM, 2009.
- Jing Ma, Ali Shojate, and George Michailidis. Network-based pathway enrichment analysis with incomplete network information. *Bioinformatics*, page btw410, 2016.
- Nicolai Meinshausen and Peter Bühlmann. High dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34(3):1436–1462, 2006.
- Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.
- Nicolai Meinshausen and Bin Yu. Lasso-type recovery of sparse representations for high-dimensional data. *Annals of Statistics*, pages 246–270, 2009.
- Karthik Mohan, Palma London, Maryam Fazel, Daniela Witten, and Su-In Lee. Node-based learning of multiple gaussian graphical models. *Journal of Machine Learning Research*, 15(1):445–488, 2014.
- Murray C. Peel, Brian L. Finlayson, and Thomas A. McMahon. Updated world map of the köppen-geiger climate classification. *Hydrology and Earth System Sciences Discussions*, 4(2):439–473, 2007.
- Bertrand Perroud, Jinoo Lee, Nelly Valkova, Amy Dirapong, Pei-Yin Lin, Oliver Fiehn, Dietmar Kiltz, and Robert H. Weiss. Pathway analysis of kidney cancer using proteomics and metabolic profiling. *Molecular Cancer*, 5:64, 2006.
- Christine Peterson, Francesco Stingo, and Marina Vannucci. Bayesian inference of multiple gaussian graphical models. *Journal of the American Statistical Association*, 110(509):159–174, 2015.
- Miguel Angel Pujana, Jing-Dong J Han, Lea M Startina, Kristen N Stevens, Munesh Tewari, Jin Sook Ahn, Gad Remmert, Víctor Moreno, Tomas Kirchhoff, Bert Gold, et al. Network modeling links breast cancer susceptibility and centrosome dysfunction. *Nature Genetics*, 39(11):1338–1349, 2007.
- Nagireddy Puturi, Ali Shojate, Vihass T Vasu, Shaiju K Vareed, Sriatha Nalluri, Vasanta Puturi, Gagan Singh Thangjam, Karim Panzitz, Christopher T Tallman, Charles Butler, et al. Metabonomic profiling reveals potential markers and bioprocesses altered in bladder cancer progression. *Cancer Research*, 71(24):7376–7386, 2011.
- C. J. Van Ripsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979.
- Adam J. Rothman, Peter J. Bickel, Elizaveta Levina, and Ji Zhu. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515, 2008.
- Rajen D. Shah and Richard J. Samworth. Variable selection with error control: Another look at stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(1):55–80, 2013.
- Thomas F. Stocker, Dabe Qin, Gian-Kasper Plattner, Melinda Tignor, Simon K. Allen, Judith Boshung, Alexander Nauels, Yu Xia, Vincent Bex, Pauline M. Midgley, et al. Climate change 2013: The physical science basis. contribution of working group I to the fifth assessment report of the intergovernmental panel on climate change. Technical report, Intergovernmental Panel on Climate Change, 2013.
- TCGA. Comprehensive molecular portraits of human breast tumours. *Nature*, 490(7418):61–70, 2012.
- Tanja Zerenner, Petra Friederichs, Klaus Lehnertz, and Andreas Hense. A gaussian graphical model approach to climate networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 24(2):023103, 2014.
- Peng Zhao and Bin Yu. On model selection consistency of lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006.
- Liming Zhou, Aiguo Dai, Yongjin Dai, Russell S Vose, Cheng-Zhi Zou, Yuhong Tian, and Haishan Chen. Spatial dependence of diurnal temperature range trends on precipitation from 1950 to 2004. *Climatic Dynamics*, 32(2):429–440, 2009.
- Shuheng Zhou, Philipp Rütimann, Min Xu, and Peter Bühlmann. High-dimensional covariance estimation based on gaussian graphical models. *Journal of Machine Learning Research*, 12:2975–3026, 2011.
- Xunzhang Zhu, Xiaotong Shen, and Wei Pan. Structural pursuit over multiple undirected graphs. *Journal of the American Statistical Association*, 109(508):1683–1696, 2014.

Support Vector Hazards Machine: A Counting Process Framework for Learning Risk Scores for Censored Outcomes

Yuanjia Wang

*Department of Biostatistics
Mailman School of Public Health
Columbia University
New York, NY 10032, USA*

YW2016@COLUMBIA.EDU

Tianle Chen

*Biogen
300 Binney Street
Cambridge, MA 02142, USA*

TIANLE.CHEN@BIOGEN.COM

Donglin Zeng

*Department of Biostatistics
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599, USA*

DZENG@EMAIL.UNC.EDU

Editor: Karsten Borgwardt

Abstract

Learning risk scores to predict dichotomous or continuous outcomes using machine learning approaches has been studied extensively. However, how to learn risk scores for time-to-event outcomes subject to right censoring has received little attention until recently. Existing approaches rely on inverse probability weighting or rank-based regression, which may be inefficient. In this paper, we develop a new support vector hazards machine (SVHM) approach to predict censored outcomes. Our method is based on predicting the counting process associated with the time-to-event outcomes among subjects at risk via a series of support vector machines. Introducing counting processes to represent time-to-event data leads to a connection between support vector machines in supervised learning and hazards regression in standard survival analysis. To account for different at risk populations at observed event times, a time-varying offset is used in estimating risk scores. The resulting optimization is a convex quadratic programming problem that can easily incorporate non-linearity using kernel trick. We demonstrate an interesting link from the profiled empirical risk function of SVHM to the Cox partial likelihood. We then formally show that SVHM is optimal in discriminating covariate-specific hazard function from population average hazard function, and establish the consistency and learning rate of the predicted risk using the estimated risk scores. Simulation studies show improved prediction accuracy of the event times using SVHM compared to existing machine learning methods and standard conventional approaches. Finally, we analyze two real world biomedical study data where we use clinical markers and neuroimaging biomarkers to predict age-at-onset of a disease, and demonstrate superiority of SVHM in distinguishing high risk versus low risk subjects.

Keywords: support vector machine, survival analysis, risk bound, risk prediction, neuroimaging biomarkers, early disease detection

1. Introduction

Time-to-event outcome is of interest in many scientific studies in which right censoring occurs when subjects' event times are longer than the duration of studies or subjects drop out of the study prematurely. One important goal in these studies is to use baseline covariates collected on a newly recruited subject to construct an effective risk score to predict likelihood an event of interest. For example, in one of our motivating studies analyzed in Section 4.2 (PREDICT-HD, Paulsen et al. 2008a), the research aim is to combine neuroimaging biomarkers with clinical markers measured at the baseline to provide risk stratification for time-to-onset of Huntington's disease (HD) to facilitate early diagnosis, where subjects who did not experience HD during the study had censored HD onset time. This critical goal of identifying prognostic markers predictive of disease onset is shared by research communities on other neurological disorders such as Alzheimer's disease and Parkinson's disease, and recognized as one of the primary aims in research initiatives such as Alzheimer's Disease Neuroimaging Initiative (Mueller et al., 2005) and Parkinson's Progression Markers Initiative (Marek et al., 2011).

Learning risk scores for binary or continuous outcomes are examined extensively in statistical learning literature (Hastie et al., 2009). However, learning risk scores for occurrence of an event subject to censoring is much less explored. Existing work on survival analysis focuses on estimating population-level quantities such as survival function or association parameters through hazard function. For example, the most popular model for the time-to-event analysis is the Cox proportional hazards model (Cox, 1972), which assumes the hazard ratio between two subjects with different covariate values stays as a constant as time progresses. A Cox partial likelihood function is maximized for estimation. When the proportional hazards assumption is violated, several alternative models have been proposed in statistics literature, including the proportional odds model (Bennett, 1983), the accelerated failure time model (Buckley and James, 1979), the linear transformation models (Dabrowska and Doksum, 1988; Cheng et al., 1995; Chen et al., 2002), and more recently general transformation models (Zeng and Lin, 2006, 2007). The above models are all likelihood-based which impose certain parametric or semiparametric relationship between the underlying hazard function and the covariates. In addition, they are designed to estimate the population-level parameters for the association between covariates and the time-to-event outcomes (and thus uses likelihood as the optimization function), but do not directly focus on individual risk scores for predicting an event time.

For non-censored outcomes, supervised learning plays an important role for risk prediction. In many applications, a large number of input variables with known output values are used to learn an unknown functional relationship between the inputs and outputs through a suitable algorithm, and the learned functional is used to predict the output value for future subjects from their input variables (Steinwart and Christmann, 2008). Many learning approaches have been developed for standard classification and regression problems, such as kernel smoothing, support vector machines (SVM), projection pursuit regression, neural network, and decision trees (Hastie et al., 2009). In particular, support vector machine is among one of the most popular and successful learning methods in practice (Moguerza and Munoz, 2006; Orru et al., 2012). From the training data, support vector machine finds a hyperplane that separates the data into two classes as accurately as possible and has a simple

geometric interpretation. In addition, the algorithm can be written as a strictly convex optimization problem, which leads to a unique global optimum and incorporates non-linearity in an automatic manner using various kernel machines. By reformulating the algorithm into a minimization of regularized empirical risk, Steinwart (2002) established the universal consistency and learning rate on some functional space. Support vector machines have also been applied to continuous outcomes through regression (Shioda and Schölkopf, 2004), multiclass category discrete outcomes (Lee et al., 2004), and structured classification problems (Wang et al., 2011).

For time-to-event outcomes, right censoring makes developing supervised learning techniques challenging due to missing event times for censored subjects and a lack of standard prediction loss function. Ripley and Ripley (2001) and Ripley et al. (2004) discussed models for survival analysis based on neural network. Bou-Hamad et al. (2011) reviewed survival tree approaches in the recent work as non-parametric alternatives to semiparametric models. Compared to survival trees, effectively extending the support vector-based methods to censored data is still an on-going research. Shivswamy et al. (2007) and Khan and Zubek (2008) proposed asymmetric modifications to the ϵ -insensitive loss function of support vector regression (SVR) to handle censoring. Specifically, they penalized the censored and non-censored subjects using different loss functions to extract incomplete information due to censoring. Van Belle et al. (2010) proposed a least-squares support vector machine, where they adopted the concept of concordance index and added rank constraints to handle censored data. In their method, the empirical risk of miss-ranking two data points with respect to their event times was minimized. Furthermore, Van Belle et al. (2011) conducted numerical experiments to compare some recent machine learning methods for censored data and proposed a modified procedure to adjust for censoring based on both rank and regression constraints. Their results indicate that including two types of constraints performs the best regarding the prediction accuracy. None of the above methods has theoretical justification and the relationship between their objective loss functions to be minimized and the goal of predicting survival time remains unclear. The rank-based methods only use feasible pairs of observations whose ranks are comparable so that it may result in potential selection bias when constructing prediction rules, especially when the censoring mechanism is not completely at random (e.g., censoring time depends/correlates with a subject's covariates). Recently, Goldberg and Kosorok (2013) used inverse-probability-of-censoring weighting to adapt standard support vector methods for complete data to censored data. However, inverse weighting is known to be inefficient (Robins et al., 1995) due to the fact that it discards useful information for some subjects known to survive longer than observed times, and in addition, this method may exhibit severe bias when the censoring distribution is misspecified. Additionally, the weights used in the inverse weighting can be large in some situations, and computation of Goldberg and Kosorok (2013) becomes numerically unstable and even infeasible.

In this work, we propose a new support vector hazards machine (SVHM) framework to learn risk scores for survival outcomes using the concept of counting process. We aim to maximally separate event and no-event subjects among all subjects at risk, and allow censoring times to depend on covariates without modeling the censoring distribution. One major challenge in predicting censored event times is the difficulty of defining a sensible loss function for prediction. Because of the equivalence of an event time to its counting process,

if a prediction rule can adequately predict the event time, the same rule should also predict the counting process at any given time that a subject is still at risk. We propose a flexible nonparametric decision function with an additive structure for the counting process, which gives the desirable risk scores but also includes a time-varying offset to account for different at-risk population as time progresses. Empirically, we transform the prediction of an event time to predicting a sequence of binary outcomes for which algorithm such as support vector machine (SVM) is standard and commonly used. This transformation allows for the successful statistical learning tools designed for classification and prediction of binary outcomes to be used for censored outcomes without modeling the censoring distribution. The developed algorithm formulation is similar to the standard support vector machines and can be solved conveniently using any convex quadratic programming packages. In addition, theoretical analysis shows that the optimal rule obtained from SVHM is equivalent to maximizing the difference between the instantaneous subject-specific hazards and population-average hazard, which intuitively links SVHM to the commonly used hazards regression models in traditional survival analysis. The profile loss shares similarity with Cox partial likelihood. Under some regularity conditions, we show the universal consistency of SVHM and derive corresponding finite sample bounds on the deviation from the optimal risk. Numeric simulations and applications to real world studies show superior performance in distinguishing high risk versus low risk subjects.

2. Learning Risk Scores with SVHM

In this section, we first introduce the population loss function that SVHM aims to optimize with infinite sample and its corresponding Bayes risk. Next, we lay out the algorithm to empirically learn the risk scores and assess the empirical risk.

2.1. Review of Survival Analysis and Introduction of Counting Process

Framework for SVHM

We begin by briefly introducing basic concepts and notation of classical survival analysis (c.f. Fleming and Harrington, 1991). Survival analysis focuses on using covariates to predict time to event outcomes. The events of interest can be death, diagnosis of a disease, onset of cancer metastasis, or failure of a machine component. An event time of interest (i.e., age at onset of a disease) is usually denoted by T , and a vector of baseline covariates (e.g., genomic risk factors) is denoted by \mathbf{X} . The main goals of survival analysis are to understand association between \mathbf{X} and T or predicting T from \mathbf{X} . A fundamental problem of survival analysis is to deal with incomplete observation of T due to that the event may not occur in some of the subjects due to study termination or subjects dropping out of the study. For example, in a study on predicting time to cancer metastasis, some subjects may not develop metastasis by the end of study period, and thus their T is not observed. These subjects are termed as being censored and their time to study termination is termed as censoring time, usually denoted by C . For each subject in the study, we observe either their event time T or censoring time C , whichever is smaller. This observation is usually denoted by $T_i \wedge C_i$, where the operator $(a \wedge b)$ denotes taking minimum of a and b . A usual assumption in survival analysis is that the censoring time C is independent of T given covariates \mathbf{X} . From a random sample of n subjects, the observed data consist of $\{T_i \wedge C_i, \Delta_i = I(T_i \leq C_i), \mathbf{X}_i\}$

for $i = 1, \dots, n$, where $I(\cdot)$ is an indicator function and $I(T_i \leq C_i)$ is thus the event indicator. The central quantity of interest in a survival analysis is occurrence of an event over time. Such occurrences are equivalent to point processes described by counting the number of events as they occur by certain time point, termed as counting processes. That is, a counting process of the event on subject i counts the number of events that have occurred up to, and including t , and is denoted as $N_i(t) = I[(T_i \wedge C_i) \leq t]$. Corresponding to the counting process for the events, the at-risk process counts subjects who have not yet had an event by time t and thus who are still "at risk" of experiencing an event. Such process is denoted by $Y_i(t) = I[(T_i \wedge C_i) \geq t]$.

The fundamental idea to learn risk scores for T to distinguish high risk versus low risk subjects is to equivalently learn risk scores for the counting process associated with T at each time point. Since the latter can be treated as a sequence of binary outcomes (event vs. no event) over time, it motivates one to reformulate the problem as deriving the risk score for predicting the jumps of the counting process over a sequence of time points among subjects still at risk at those times. This amounts to developing a classification rule to predict whether a subject will experience an event in the next immediate time point given that the subject has not yet experienced an event. To account for different risk sets as time progresses (i.e., risk set at time t is the subset of subjects with $Y_i(t) = 1$), it is necessary to include a time-varying offset for the nonparametric risk score. Thus, consider the following general form at time t for a subject with $\mathbf{X} = \mathbf{x}$,

$$f(t, \mathbf{x}) = \alpha(t) + g(\mathbf{x}), \quad (1)$$

where both $\alpha(\cdot)$ and $g(\cdot)$ are unknown nonparametric functions, $g(\mathbf{x})$ is the risk score, and $\alpha(t)$ is the time-varying offset. To understand (1), consider a risk score function at time t for a subject with $\mathbf{X} = \mathbf{x}$: if this subject is still at risk at time t , we predict the subject to experience the event at the next immediate time point if $f(t, \mathbf{x}) > 0$, and predict as event-free if $f(t, \mathbf{x}) \leq 0$. Thus, within a small time interval $[t, t + dt)$, where dt denotes a positive infinitesimal unit, a natural prediction loss counting rate of risk-misclassification is given by

$$Y(t)dN(t)I(f(t, \mathbf{X}) < 0) + Y(t)(1 - dN(t))I(f(t, \mathbf{X}) \geq 0),$$

where $Y(t)$ and \mathbf{X} are the at risk process and covariates for a subject drawn from the population, respectively, and $dN(t)$ denotes the number of jumps of the counting process $N(t)$ in a small time interval $[t, t + dt)$. Equivalently, $dN(t) = 1$ if $T \in [t, t + dt)$ and $dN(t) = 0$ otherwise, so it is a binary variable taking value one if an event occurs in the interval $[t, t + dt)$ for subjects who are still at risk for experiencing an event. Thus, summing above loss function over subjects counts the number of at-risk subjects miss-classified by the prediction rule $f(t, \mathbf{X})$. The above prediction loss can be viewed as a natural extension of the 0-1 loss for binary case to capture the same information for an at-risk subject in a survival analysis: if the prediction function and the observed counting process at time t are inconsistent, a loss is incurred. However, at any time t , the probability of $dN(t) = 1$ is almost zero as compared to the probability of $dN(t) = 0$, which implies that the above prediction loss is completely dominated by non-event subjects in the risk set. In order to balance the contribution from subjects with and without events at any given time, borrowing from the weighted SVM for unbalanced classes, a sensible prediction loss is the following

weighted loss, where the ratio of weights for two unbalanced classes is proportional to $E[dN(t)]/E[Y(t)]$:

$$Y(t)dN(t)I(f(t, \mathbf{X}) \leq 0) + \frac{E[dN(t)]}{E[Y(t)]}Y(t)(1 - dN(t))I(f(t, \mathbf{X}) \geq 0). \quad (2)$$

This weighting scheme can also be understood in the context of nested case-control design. That is, select one subject from the event class, $\{i : dN_i(t) = 1\}$, at this interval and another subject from the non-event class, $\{i : dN_i(t) = 0\}$, using $E[dN(t)]/E[Y(t)]$ as the sampling weights for the latter. Consequently, an overall weighted prediction loss for the proposed SVHM, which is the expectation of (2) and ignores infinitesimal terms, is

$$\mathcal{R}_0(f) = E \left(\int Y(t)I(f(t, \mathbf{X}) \leq 0]dN(t) \right) + \int \frac{E(Y(t)I(f(t, \mathbf{X}) \geq 0])}{E(Y(t))}E(dN(t)),$$

where the expectation is with respect to random variables $Y(t)$ and $dN(t)$. Our goal of learning a prediction rule for T , or equivalently, $N(t)$, based on the censored data is to minimize the population loss $\mathcal{R}_0(f)$.

To define the empirical loss, suppose there are m distinct ordered event times, $t_1 < t_2 < \dots < t_m$. We let

$$\delta N_i(t_j) \equiv 2(N_i(t_j) - N_i(t_{j-1})) - 1$$

so $\delta N_i(t_j)$ takes values 1 or -1 depending on whether the i th subject experiences an event at t_j or not. Learning $f(t, \mathbf{x})$ becomes a sequence of binary classification problems over t_j 's. Furthermore, at each t_j and for subject i at risk at t_j , we use the following weight associated with the risk set size at t_j :

$$u_i(t_j) = I \left\{ \delta N_i(t_j) = 1 \right\} \left\{ 1 - \frac{1}{\sum_{i=1}^n Y_i(t_j)} \right\} + I \left\{ \delta N_i(t_j) = -1 \right\} \left\{ \frac{1}{\sum_{i=1}^n Y_i(t_j)} \right\}.$$

Note that the weights $u_i(t_j)$ are the empirical version of the weights used in (2) with similar interpretation as the reciprocal of the empirical probability of remaining event free or experiencing an event at the observed event time. Such weights balance the differential size of event class and non-event class at time t_j . Then an optimal decision function that minimizes the empirical version of $\mathcal{R}_0(f)$ is to minimize the following weighted total misclassification error:

$$\mathcal{R}_{0n}(f) = n^{-1} \sum_{i=1}^n \sum_{j=1}^m u_i(t_j) Y_i(t_j) I(\delta N_i(t_j) f(t_j, \mathbf{X}_i) < 0), \quad (3)$$

where the term $Y_i(t_j)$ reflects that only subjects still at risk will contribute towards prediction.

Directly minimizing (3) is difficult due to non-smoothness of the 0-1 loss in the indicator function. Furthermore, no restriction on the complexity of f leads to potential overfitting. To handle these issues, we adopt the same idea as SVM for supervised learning to replace the 0-1 loss in (1) by the hinge loss, and impose regularization to estimate f . Specifically, we propose to minimize the following regularized SVHM loss:

$$\mathcal{R}_n(f) + \lambda_n \|f\|^2,$$

$$\text{with } \mathcal{R}_n(f) \equiv n^{-1} \sum_{i=1}^n \sum_{j=1}^m w_i(t_j) Y_i(t_j) [1 - f(t_j, \mathbf{X}_i) \delta N_i(t_j)]_+, \quad (4)$$

where $[1 - x]_+ = \max(1 - x, 0)$ is the hinge loss, $\|\cdot\|$ is a suitable norm or semi-norm for f to be discussed in the following sections, and λ_n is the regularization parameter. This minimization is equivalent to maximizing the margin between subjects in the event and non-event classes subject to an upper bound on the misclassification rate. Since this learning method is a weighted version of the standard support vector machines and learning $f(t, \mathbf{x})$ is essentially learning the hazard rate function, we refer our proposed method as ‘‘support vector hazards machine’’.

2.2 Learning Algorithm

Next, we describe the computational algorithm to solve the optimization in (4). We do not impose any restriction on $\alpha(t)$, and assume $g(\mathbf{X})$ lies in a reproducing kernel Hilbert space \mathcal{H}_n with a kernel function $K(\mathbf{x}, \mathbf{x}')$. Commonly used kernels include linear kernel, where $K(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle$; radial basis kernel where $K(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\|^2/\sigma)$; and l th-degree polynomial kernel, where $K(\mathbf{x}, \mathbf{x}') = (1 + \langle \mathbf{x}, \mathbf{x}' \rangle)^l$. Furthermore, let $\|f\| = \|g\|_{\mathcal{H}_n}$ which is the norm in the reproducing kernel Hilbert space \mathcal{H}_n . The minimization in (3),

$$\min_{\alpha, g} \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^m w_i(t_j) Y_i(t_j) [1 - (\alpha(t_j) + g(\mathbf{X}_i)) \delta N_i(t_j)]_+ + \lambda_n \|g\|_{\mathcal{H}_n}^2, \quad (5)$$

is equivalent to

$$\min_{\alpha, g} \frac{1}{2} \|g\|_{\mathcal{H}_n}^2 + C_n \sum_{i=1}^n \sum_{j=1}^m w_i(t_j) Y_i(t_j) \zeta_i(t_j), \quad (6)$$

subject to $Y_i(t_j) \zeta_i(t_j) \geq 0, i = 1, \dots, n, j = 1, \dots, m$,

$$Y_i(t_j) \delta N_i(t_j) \{\alpha(t_j) + g(\mathbf{X}_i)\} \geq Y_i(t_j) \{1 - \zeta_i(t_j)\}, i = 1, \dots, n, j = 1, \dots, m,$$

where the value $\zeta_i(t_j)$ is the proportional amount by which the prediction is on the wrong side of its margin at time t_j , and C_n is the cost parameter.

The constrained optimization in (6) is usually solved by turning it into its dual form (through including Lagrange multipliers of the constraints into the objective function). We convert the above problem to its dual form by using the corresponding Lagrangian function

$$\begin{aligned} L_p &= \frac{1}{2} \|g\|_{\mathcal{H}_n}^2 + C_n \sum_{i=1}^n \sum_{j=1}^m w_i(t_j) Y_i(t_j) \zeta_i(t_j) - \sum_{i=1}^n \sum_{j=1}^m \mu_{ij} Y_i(t_j) \zeta_i(t_j) \\ &\quad - \sum_{i=1}^n \sum_{j=1}^m \gamma_{ij} [Y_i(t_j) \delta N_i(t_j) \{\alpha(t_j) + g(\mathbf{X}_i)\} - Y_i(t_j) \{1 - \zeta_i(t_j)\}], \end{aligned}$$

where $\mu_{ij} \geq 0$ and $\gamma_{ij} \geq 0$ are the corresponding Lagrange multipliers. Let $\{\phi_1, \phi_2, \dots\}$ be the orthonormal basis system in \mathcal{H}_n and $g(\mathbf{X}) = \sum_{k=1}^{\infty} \beta_k \phi_k(\mathbf{X})$. Then after differentiating

the Lagrangian function with respect to β 's, $\alpha(t_j)$'s and $\zeta_i(t_j)$'s, we obtain

$$\begin{aligned} \beta_k &= \sum_{i=1}^n \sum_{j=1}^m \gamma_{ij} Y_i(t_j) \delta N_i(t_j) \phi_k(\mathbf{X}_i), \quad k = 1, 2, \dots, \\ &\quad \sum_{i=1}^n \gamma_{ij} Y_i(t_j) \delta N_i(t_j) = 0, \end{aligned}$$

$$C_n w_i(t_j) Y_i(t_j) - \mu_{ij} Y_i(t_j) = \gamma_{ij} Y_i(t_j), \quad i = 1, \dots, n, j = 1, \dots, m,$$

as well as the positivity constraints $\gamma_{ij}, \mu_{ij}, \zeta_i(t_j) \geq 0$ for all i and j . By substituting these back to L_p and noting that $\sum_{k=1}^{\infty} \phi_k(\mathbf{X}_i) \phi_k(\mathbf{X}) = K(\mathbf{X}_i, \mathbf{X})$ (Theorem 4.2, Steinwart (2002)), we obtain the dual objective function to be

$$L_D = \sum_{i=1}^n \sum_{j=1}^m \gamma_{ij} Y_i(t_j) - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^m \sum_{i'=1}^n \sum_{j'=1}^m \gamma_{ij} \gamma_{i'j'} Y_i(t_j) Y_{i'}(t_{j'}) \delta N_i(t_j) \delta N_{i'}(t_{j'}) K(\mathbf{X}_i, \mathbf{X}_{i'}), \quad (7)$$

and the optimization is carried out by maximizing L_D with respect to γ_{ij} subject to $0 \leq \gamma_{ij} \leq w_i(t_j) C_n$ and $\sum_{i=1}^n \sum_{j=1}^m \gamma_{ij} Y_i(t_j) \delta N_i(t_j) = 0$ for $i = 1, \dots, n$ and $j = 1, \dots, m$. This optimization can be solved using quadratic programming packages available in many softwares (for example, MOSEK toolbox in Matlab). The tuning parameter C_n is chosen by cross-validation searching over a grid of values. Denote $\hat{\gamma}_{ij}$ as the solutions for γ_{ij} obtained from the optimization procedure in (7). Comparing (7) with existing standard support vector machine algorithms, we see that the objective function sums across all at-risk subjects and across all time points for which they are at risk. Constraints are placed on those subjects and time points.

Next, from the equalities between β_k 's and γ_{ij} 's in the above duality derivation, the solutions for β_k (denoted as $\hat{\beta}_k$) are given by

$$\hat{\beta}_k = \sum_{i=1}^n \sum_{j=1}^m \hat{\gamma}_{ij} Y_i(t_j) \delta N_i(t_j) \phi_k(\mathbf{X}_i), \quad k = 1, 2, \dots.$$

Thus, the solution for g that minimizes (5), which is the risk score for a future subject with baseline covariates \mathbf{x} , is

$$\begin{aligned} \hat{g}(\mathbf{x}) &= \sum_{k=1}^{\infty} \hat{\beta}_k \phi_k(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^m \hat{\gamma}_{ij} Y_i(t_j) \delta N_i(t_j) \sum_{k=1}^{\infty} \phi_k(\mathbf{X}_i) \phi_k(\mathbf{x}) \\ &= \sum_{i=1}^n \sum_{j=1}^m \hat{\gamma}_{ij} \delta N_i(t_j) K(\mathbf{x}, \mathbf{X}_i). \end{aligned} \quad (7)$$

It follows that those data points with $\hat{\gamma}_{ij} > 0$ form support vectors and determine $g(\mathbf{X})$.

Furthermore, to determine the solution to $\alpha(t_j)$ at each t_j , denoted by $\hat{\alpha}(t_j)$, we solve the Karush-Kuhn-Tucker (KKT) conditions

$$\gamma_{ij} [Y_i(t_j) \delta N_i(t_j) \{\alpha(t_j) + g(\mathbf{X}_i)\} - Y_i(t_j) \{1 - \zeta_i(t_j)\}] = 0,$$

$$Y_i(t_j)\zeta_i(t_j) \geq 0,$$

$$Y_i(t_j)\delta N_i(t_j)\{\alpha(t_j) + g(\mathbf{X}_i)\} - Y_i(t_j)\{1 - \zeta_i(t_j)\} \geq 0.$$

Specifically, if there are some support vectors lying on the edge of the margin which are characterized by $0 < \hat{\gamma}_{ij} < w_i(t_j)C_n$, $\hat{\alpha}(t_j) = 1/\delta N_i(t_j) - \hat{g}(\mathbf{X}_i)$ for these points, and we take the average of all the solutions for numerical stability. If all the support vectors at t_j are $\hat{\gamma}_{ij} = C_n w_i(t_j)$, $\hat{\alpha}(t_j)$ is not unique and falls into a range

$$\min_{\substack{Y_i(t_j)=1, \hat{\gamma}_{ij}=C_n w_i(t_j), \\ \delta N_i(t_j)=1}} \{1 - \hat{g}(\mathbf{X}_i)\} \geq \hat{\alpha}(t_j) \geq \max_{\substack{Y_i(t_j)=1, \hat{\gamma}_{ij}=C_n w_i(t_j), \\ \delta N_i(t_j)=-1}} \{-1 - \hat{g}(\mathbf{X}_i)\}.$$

In this case, we take $\hat{\alpha}(t_j) = 1 - \hat{g}(\mathbf{X}_i)$ where $\delta N_i(t_j) = 1$ for some i with $Y_i(t_j) = 1$.

Since a higher value of the prediction function $\hat{\alpha}(t) + \hat{g}(x)$ leads to a greater likelihood of having an event at an earlier time, the magnitude of $\hat{g}(x)$ induces a natural ordering of the risks. Lastly, the learned risk scores can be used to predict the event time for any future subjects using their baseline covariates \mathbf{x} . To this end, consider the nearest-neighbor prediction: for a future subject with $\mathbf{X} = \mathbf{x}$, find k ($k=1$ or 3 in our applications) non-censored subjects in the training data whose predictive scores are closest to $\hat{g}(\mathbf{x})$, denoted as $\hat{g}(\mathbf{X}_j)$. To maintain the monotone relationship between the event times and predictive scores, sort these scores of non-censored subjects in the training data in descending order and identify the rank of $\hat{g}(\mathbf{X}_j)$. Next, sort the event times of the derived scores in the training data in ascending order and find the event times with the same rank as the rank of $\hat{g}(\mathbf{X}_j)$, denoted as $T_{j'}$. The event time for this subject is predicted as $T_{j'}$ (or the average of $T_{j'}$ for $k=3$). We provide a detailed description of SVHM algorithm in Appendix A.

2.3 Connection with Existing Support Vector-Based Approaches

Support vector-based approaches in machine learning literature are motivated by the fact that they are easy to compute and enable estimation under weak or no assumptions on the distribution. Most of these approaches (Shivaswamy et al., 2007; Van Belle et al., 2010, 2011) adapt the ϵ -insensitive loss for SVR to account for incomplete observations in time-to-event data. To improve performance, modified SVR (Van Belle et al., 2011) further places ranking constraints under the ϵ -insensitive loss. The formulation of the problem is

$$\min_{\mathbf{w}, \boldsymbol{\xi}, \boldsymbol{\xi}^*} \frac{1}{2} \mathbf{w}^T \mathbf{w} + \lambda_1 \sum_i \epsilon_i + \lambda_2 \sum_i (\xi_i + \xi_i^*), \quad (8)$$

$$\text{subject to } \mathbf{w}^T(\varphi(\mathbf{X}_i) - \varphi(\mathbf{X}_{j(i)})) \geq Y_i - Y_{j(i)} - \epsilon_i, i = 1, \dots, n,$$

$$\mathbf{w}^T \varphi(\mathbf{X}_i) + b \geq Y_i - \xi_i, i = 1, \dots, n,$$

$$\Delta_i(\mathbf{w}^T \varphi(\mathbf{X}_i) + b) \geq -\Delta_i Y_i - \xi_i^*, i = 1, \dots, n,$$

$$\epsilon_i \geq 0, \xi_i \geq 0, \xi_i^* \geq 0, i = 1, \dots, n,$$

where $Y_i = T_i \wedge C_i$, $\varphi(\cdot)$ is the feature map that does not need to be specified explicitly in a kernel-based method, and $j(i)$ indicates the subject with the largest event time smaller than Z_i . The first set of constraints above aims at ensuring rank consistency to maximize C-index for predicting survival outcomes, and the second and third sets of constraints are the

same as the regression constraints in Shivaswamy et al. (2007) for the modified ϵ -insensitive loss for survival outcomes. One potential problem with the above optimization is that the observations contributing to these three sets of constraints may consist of a selected (non-censored) sample from the full data; thus, the derived prediction rule will likely favor those observations which contribute most.

Furthermore, comparing the modified SVR in (8) with SVHM in (6), we see that the loss function for the former is the ϵ -insensitivity loss plus the loss resulting from violating rank consistency, while for the latter it is sum of a sequence of hinge losses. The objective function and the slack variables (i.e., $\epsilon_i, \xi_i, \xi_i^*$) for the modified SVR, however, are time-invariant, while the slack variables for SVHM (i.e., $\zeta_i(t_j)$ in (8)) are time-sensitive. Thus, we expect better control of the prediction error by SVHM. Note that this advantage stems from the counting process formulation of SVHM transforming prediction of time-to-event outcomes (or survival outcomes) as a sequence of binary prediction problems over time.

2.4 Connection with the Cox Partial Likelihood

In classical survival analysis using Cox regression model (Cox, 1972), partial likelihood plays a central role since it only involves association parameter of interest (i.e., hazard ratios as regression coefficients) but not the nuisance parameter (i.e., baseline hazard function), and maximizing the partial likelihood directly estimates the hazard ratios. The partial likelihood is constructed by multiplying together the conditional probabilities of observing an event for individual i at time t , given the past and given that an event is observed at that time, over all observed event times. This conditional probability formulation shares some similarity with our hazard formulation for SVHM. Since maximizing Cox partial likelihood leads to regression estimators that enjoy optimal statistical property (i.e., being semiparametric efficient, Bickel et al. (1998)), it is worth to draw connection between SVHM and partial likelihood to shed lights on the theoretical properties of SVHM.

To this end, we further explore the optimization in (5) to compare the SVHM objective function and the Cox partial likelihood. First note that the function $\alpha(t)$ in (5) is analogous to the baseline hazard function in the Cox model (Cox, 1972), which is treated as a nuisance parameter and profiled out for inference. Thus, we also profile out $\alpha(t)$ to investigate the profile risk function for SVHM (e.g., substitute fitted $\alpha(t)$ in the original risk function). For a fixed $g(\mathbf{x})$, from the derivation similar to Hastie et al. (2009) (p421) and Abe (2010) (p77), we can show that at each t_j , if there are some support vectors lying on the edge of the margin which are characterized by $0 < \hat{\gamma}_{ij} < w_i(t_j)C_n$, these margin points can be used to solve for $\alpha(t_j)$. This yields

$$\hat{\alpha}(t_j) = 1 - g(\mathbf{X}_i), \quad \delta N_i(t_j) = 1.$$

Note that \mathbf{X}_i is the covariate value for the subject who has an event at t_j . However, if $\hat{\gamma}_{ij}$ is not within $(0, w_i(t_j)C_n)$, $\hat{\alpha}(t_j)$ can be any value satisfying

$$\min_{\substack{\hat{\gamma}_{ij}=C_n w_i(t_j), \\ \delta N_i(t_j)=-1}} \{1 - g(\mathbf{X}_i)\} \geq \alpha(t_j) \geq \max_{\substack{\hat{\gamma}_{ij}=C_n w_i(t_j), \\ \delta N_i(t_j)=-1}} \{-1 - g(\mathbf{X}_i)\}.$$

In this case, taking $\hat{\alpha}(t_j) = 1 - g(\mathbf{X}_i)$ where $\delta N_i(t_j) = 1$ satisfies these constraints. Further note that optimizing (5) is equivalent to minimizing

$$\begin{aligned} & \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^d Y_i(t_j) w_i(t_j) [1 - (\alpha(t) + g(\mathbf{X}_i)) \delta N_i(t_j)]_+ + \lambda_n \|g\|_{\mathcal{H}_n} \\ &= \frac{1}{n} \sum_{i=1}^n \int [1 - (\alpha(t) + g(\mathbf{X}_i))]_+ dN_i(t) + \frac{1}{n} \int \frac{\sum_{i=1}^n Y_i(t) [1 + (\alpha(t) + g(\mathbf{X}_i))]_+ d \left\{ \sum_{i=1}^n N_i(t) \right\}}{\sum_{i=1}^n Y_i(t)} \\ & \quad - \frac{1}{n} \sum_{i=1}^n \int \frac{1}{\sum_{i=1}^n Y_i(t)} ([1 - (\alpha(t) + g(\mathbf{X}_i))]_+ + [1 + (\alpha(t) + g(\mathbf{X}_i))]_+) dN_i(t) + \lambda_n \|g\|_{\mathcal{H}_n}. \end{aligned}$$

After we plug the expression of $\hat{\alpha}(t) = \sum_{i=1}^n (1 - g(\mathbf{X}_i)) I(\delta N_i(t) = 1)$ into the above expression, we obtain

$$\frac{1}{n} \sum_{i=1}^n \int [1 - (\hat{\alpha}(t) + g(\mathbf{X}_i))]_+ dN_i(t) = \frac{1}{n} \sum_{i=1}^n \Delta_i [1 - (1 - g(\mathbf{X}_i) + g(\mathbf{X}_i))]_+ = 0,$$

and similarly,

$$\frac{1}{n} \sum_{i=1}^n \int \frac{1}{\sum_{i=1}^n Y_i(t)} ([1 - (\hat{\alpha}(t) + g(\mathbf{X}_i))]_+ + [1 + (\hat{\alpha}(t) + g(\mathbf{X}_i))]_+) dN_i(t) = \frac{2}{n} \sum_{i=1}^n \int \frac{dN_i(t)}{\sum_{i=1}^n Y_i(t)}.$$

Additionally,

$$\begin{aligned} & \frac{1}{n} \int \frac{\sum_{i=1}^n Y_i(t) [1 + (\hat{\alpha}(t) + g(\mathbf{X}_i))]_+ d \left\{ \sum_{i=1}^n N_i(t) \right\}}{\sum_{i=1}^n Y_i(t)} \\ &= \frac{1}{n} \sum_{k=1}^n \Delta_k \frac{\sum_{i=1}^n I(Y_i \geq Y_k) [1 + (\hat{\alpha}(\mathbf{X}_k) + g(\mathbf{X}_k))]_+}{\sum_{i=1}^n I(Y_i \geq Y_k)} \\ &= \frac{1}{n} \sum_{k=1}^n \frac{\sum_{i=1}^n I(Y_i \geq Y_k) [2 - g(\mathbf{X}_k) + g(\mathbf{X}_i)]_+ \Delta_k}{\sum_{i=1}^n I(Y_i \geq Y_k)}. \end{aligned}$$

The objective function (5) can be written as $\mathcal{P}\mathcal{R}_n(g) + \lambda_n \|g\|_{\mathcal{H}_n}^2$, where

$$\begin{aligned} \mathcal{P}\mathcal{R}_n(g) &= \frac{1}{n} \sum_{i=1}^n \int \frac{\sum_{k=1}^n Y_k(t) [2 - g(\mathbf{X}_i) + g(\mathbf{X}_k)]_+ dN_i(t)}{\sum_{k=1}^n Y_k(t)} - \frac{2}{n} \sum_{i=1}^n \int \frac{dN_i(t)}{\sum_{k=1}^n Y_k(t)} \\ &= \frac{1}{n} \sum_{i=1}^n \Delta_i \frac{\sum_{k=1}^n I(Y_k \geq Y_i) [2 - g(\mathbf{X}_i) + g(\mathbf{X}_k)]_+}{\sum_{k=1}^n I(Y_k \geq Y_i)} - \frac{2}{n} \sum_{i=1}^n \frac{\Delta_i}{\sum_{k=1}^n I(Y_k \geq Y_i)} \\ &= \mathbf{P}_n \left(\Delta \frac{\tilde{\mathbf{P}}_n \{I(\tilde{Y} \geq Y) [2 + g(\tilde{\mathbf{X}}) - g(\mathbf{X})]_+\}}{\tilde{\mathbf{P}}_n [I(\tilde{Y} \geq Y)]} - \frac{2}{n} \mathbf{P}_n \left\{ \frac{\Delta}{\tilde{\mathbf{P}}_n [I(\tilde{Y} \geq Y)]} \right\} \right). \end{aligned}$$

Here, \mathbf{P}_n denotes the empirical measure from n observations and $\tilde{\mathbf{P}}_n$ is the empirical measure applied to $(\tilde{Y}, \tilde{\mathbf{X}}, \tilde{\Delta})$, an i.i.d copy of (Y, \mathbf{X}, Δ) . Thus, $\hat{g}(\mathbf{x})$ minimizes $\mathcal{P}\mathcal{R}_n(g) + \lambda_n \|g\|_{\mathcal{H}_n}^2$.

If we let $\hat{f}(g, t) = \hat{\alpha}(t) + \hat{g}(\mathbf{x})$ be the function minimizing (5) over $g \in \mathcal{H}_n$, then $\mathcal{R}_n(\hat{f}) = \mathcal{P}\mathcal{R}_n(\hat{g})$.

In a Cox partial likelihood function, $g(\mathbf{X})$ is estimated by minimizing

$$\mathbf{P}_n \left(\Delta \log \frac{\tilde{\mathbf{P}}_n \{I(\tilde{Y} \geq Y) \exp\{g(\tilde{\mathbf{X}}) - g(\mathbf{X})\}\}}{\tilde{\mathbf{P}}_n [I(\tilde{Y} \geq Y)]} \right).$$

Therefore, it is worthy to point out one interesting observation: both $\mathcal{P}\mathcal{R}_n(g)$ and the Cox partial likelihood take a similar form which essentially evaluates a loss comparing the risk scores from the subjects at risk versus the one from the subject who has an event at the same time. SVHM uses a hinge loss while Cox partial likelihood uses an exponential loss and a logarithm transformation, which is similar to the contrast between SVM and logistic regression. The robustness of hinge loss compared to exponential loss suggests SVHM will be less sensitive to extreme observations. In addition, this connection sheds lights on the theoretical optimality of SVHM which we prove in the next section.

3. Theoretical Properties

In this section, we study the asymptotic properties of SVHM and the predicted risk. We first derive the population risk function for the proposed SVHM. Next, we derive the optimal fully nonparametric decision rule for this risk function and show that it also optimizes the 0-1 loss corresponding to (3). We highlight important differences in the theoretical proof that distinguish this work from the standard proofs in the statistical learning theories.

3.1 Risk Function and Optimal Risk Classification Rule

To derive the population risk function for SVHM, we first examine the population version (the expectation) of $\mathcal{R}_n(f)$. Recall the definition of $\mathcal{R}_n(f)$ is given in (4) as

$$\begin{aligned} \mathcal{R}_n(f) &= n^{-1} \sum_{i=1}^n \sum_{j=1}^n \frac{I\{\delta N_i(t_j) = 1\} (\sum_{i=1}^n Y_i(t_j) - 1)}{\sum_{i=1}^n Y_i(t_j)} [1 - f(t_j, \mathbf{X}_i)]_+ \\ & \quad + n^{-1} \sum_{i=1}^n \sum_{j=1}^n \frac{I\{\delta N_i(t_j) = 1\}}{\sum_{i=1}^n Y_i(t_j)} [1 + f(t_j, \mathbf{X}_i)]_+, \end{aligned}$$

After re-arranging the terms and adopting counting process notation, we can rewrite $\mathcal{R}_n(f)$ as

$$\begin{aligned} \mathcal{R}_n(f) &= \frac{1}{n} \sum_{i=1}^n \int Y_i(t) [1 - f(t, \mathbf{X}_i)]_+ dN_i(t) + \frac{1}{n} \int \frac{\sum_{i=1}^n Y_i(t) [1 + f(t, \mathbf{X}_i)]_+ d \left\{ \sum_{i=1}^n N_i(t) \right\}}{\sum_{i=1}^n Y_i(t)} \\ & \quad - \frac{1}{n} \sum_{i=1}^n \int \frac{1}{\sum_{i=1}^n Y_i(t)} ([1 - f(t, \mathbf{X}_i)]_+ + [1 + f(t, \mathbf{X}_i)]_+) dN_i(t). \end{aligned}$$

Note that the last term in $\mathcal{R}_n(f)$ is on the order of $O(1/n)$, so it vanishes as n goes to infinity. By the central limit theorem, we obtain the asymptotic limit of $\mathcal{R}_n(f)$, denoted as

$\mathcal{R}(f)$, to be

$$\mathcal{R}(f) = E \left(\int Y(t)[1 - f(t, \mathbf{X})]_+ dN(t) \right) + \int \frac{E(Y(t)[1 + f(t, \mathbf{X})]_+)}{E(Y(t))} E(dN(t)).$$

Likewise, similar arguments show that the empirical risk based on the prediction error in (1), i.e., $\mathcal{R}_n(f)$, converges to $\mathcal{R}_0(f)$.

Let $f^*(t, \mathbf{x})$ denote the limit of the risk function estimated by SVHM (i.e., the optimal function minimizing $\mathcal{R}(f)$). Since the difference between $\mathcal{R}(f)$ and $\mathcal{R}_0(f)$ is the hinge loss versus the zero-one loss, one question is whether $f^*(t, \mathbf{x})$ also minimizes $\mathcal{R}_0(f)$. The following theorem gives such a result for $f^*(t, \mathbf{x})$.

Theorem 3.1 *Let $h(t, \mathbf{x})$ denote the conditional hazard rate function of $T = t$ given $\mathbf{X} = \mathbf{x}$ and let $\bar{h}(t) = E[dN(t)/dt]/E[Y(t)] = E[h(t, \mathbf{X})|Y(t) = 1]$ be the average hazard rate at time t . Then $f^*(t, \mathbf{x}) = \text{sign}(h(t, \mathbf{x}) - \bar{h}(t))$ minimizes $\mathcal{R}(f)$. Furthermore, $f^*(t, \mathbf{x})$ also minimizes $\mathcal{R}_0(f)$ and*

$$\mathcal{R}_0(f^*) = P(T \leq C) - \frac{1}{2} E \left[\int E(Y(t)|\mathbf{X} = \mathbf{x})h(t, \mathbf{x}) - \bar{h}(t) dt \right].$$

In addition, for any $f(t, \mathbf{x}) \in [-1, 1]$,

$$\mathcal{R}_0(f) - \mathcal{R}_0(f^*) \leq \mathcal{R}(f) - \mathcal{R}(f^*),$$

where $h(t, \mathbf{x})$ denotes the conditional hazard rate of $T = t$ given $\mathbf{X} = \mathbf{x}$ and $\bar{h}(t)$ is the population average hazard at time t ,

$$\bar{h}(t) = \frac{E[dN(t)]/dt}{E[Y(t)]} = E[h(t, \mathbf{X})|Y(t) = 1].$$

The proof of Theorem 3.1 is provided in the Appendix B. Theorem 3.1 resembles the excess risk in most learning theories (Bartlett et al., 2006); however, the loss function in our case is some composite expectation, $\mathcal{R}_0(f)$, which is not covered by Bartlett et al. (2006). From Theorem 3.1, we see that the optimal rule is essentially to predict whether an at-risk subject will experience an event by comparing the subject-specific hazard rate depending on the covariate to the population-average hazard rate obtained from all at-risk subjects at a given time point. Since the minimizer of $\mathcal{R}(f)$ also minimizes $\mathcal{R}_0(f)$, this theory justifies the use of hinge-loss in SVHM to minimize the weighted prediction error in $\mathcal{R}_0(f)$. The last inequality in Theorem 3.1 proves that a decision function with a small excess hinge-loss-based risk will lead to a small excess 0-1 loss-based risk.

3.2 Asymptotic Properties

Here, we study the asymptotic properties of SVHM when the decision function takes the form in (1). Specifically, we examine a stochastic bound for the excess risk when using \hat{g} , the estimator from n observations. This bound will be given in terms of the sample size n , the tuning parameter λ_n and the bandwidth of the kernel function σ_n . Denote \mathcal{H}_n as

a reproducing kernel Hilbert space from a Gaussian kernel $k(x, x') = \exp\{-\|x - x'\|^2/\sigma_n\}$. Instead of considering the risk for $\mathcal{R}(f)$, we consider

$$\mathcal{PR}(g) = \min_{\alpha(t)} \mathcal{R}(\alpha(t) + g(\mathbf{x}))$$

and refer it as ‘‘profile risk’’, since $\alpha(t)$ is profiled out from the original risk function. In other words, $\mathcal{PR}(g)$ is the best expected risk for a given score $g(\mathbf{x})$ after accounting for $\alpha(t)$.

To obtain an explicit expression of $\mathcal{PR}(g)$, we first note that

$$\begin{aligned} \mathcal{R}(\alpha(t) + g(\mathbf{x})) &= E \left(\int Y(t)[1 - f(t, \mathbf{X})]_+ dN(t) \right) + \int \frac{E(Y(t)[1 + f(t, \mathbf{X})]_+)}{E(Y(t))} E(dN(t)) \\ &= \int E[Y(t)h(t, \mathbf{X})] \left[\frac{E[Y(t)h(t, \mathbf{X})] - E[Y(t)g(\mathbf{X})h(t, \mathbf{X})]}{E[Y(t)h(t, \mathbf{X})]} - \alpha(t) \right]_+ dt \\ &\quad + \int \bar{h}(t)E[Y(t)] \left[\frac{E[Y(t)] + E[Y(t)g(\mathbf{X})]}{E[Y(t)]} + \alpha(t) \right]_+ dt. \end{aligned}$$

Since $\alpha(t)$ is arbitrary and the integrand in the above expression is a piecewise linear function of $\alpha(t)$, simple algebra gives that

$$\alpha(t) = - \frac{E[Y(t)] + E[Y(t)g(\mathbf{X})]}{E[Y(t)]}$$

minimizes $\mathcal{R}(f)$. Therefore, after replacing $\alpha(t)$ by this minimizer in $\mathcal{R}(\alpha(t) + g(\mathbf{x}))$, we obtain

$$\mathcal{PR}(g) = E \left[\Delta \frac{\tilde{\mathbf{P}}I(\tilde{Y} \geq Y)}{g} [2 - g(\tilde{\mathbf{X}}) + g(\mathbf{X})]_+ \right].$$

Clearly, $\mathcal{PR}(g)$ is the asymptotic limit of $\mathcal{PR}_n(g)$. The following theorem holds for the risk $\mathcal{PR}(\hat{g})$.

Theorem 3.2 *Assume that \mathbf{X} 's support is compact and $E[Y(\tau)|\mathbf{X}]$ is bounded from zero where τ is the study duration. Furthermore, assume λ_n and σ_n satisfies $\lambda_n, \sigma_n \rightarrow 0$, and $n\lambda_n\sigma_n^{(2/p-1/2)d} \rightarrow \infty$ for some $p \in (0, 2)$. Then it holds*

$$\lambda_n \|\hat{g}\|_{\mathcal{H}_n}^2 + \mathcal{PR}(\hat{g}) \leq \inf_g \mathcal{PR}(g) + O_p \left\{ \lambda_n + \frac{\sigma_n^{d/2}}{\sqrt{n}} + \frac{\lambda_n^{-1/2} \sigma_n^{-(1/p-1/4)d}}{\sqrt{n}} \right\}.$$

The proof of Theorem 3.2 mostly follows the machinery for support vector machines. It mainly uses empirical process theories to control the stochastic error of the empirical risk functions and the approximation properties of the reproducing kernel Hilbert space based on the Gaussian kernel function. However, one major difference from the classical proof is that the empirical loss function we study here is some composite statistics instead of the summation of n i.i.d terms. This poses additional challenges to control stochastic variability. The constants in Theorem 3.2 imply that the bandwidth for the Gaussian

kernel and regularization parameter should converge to zero in certain rates depending on \mathbf{X} 's dimension, but not too fast to ensure stochastic variability is under control. Finally, we state two useful observations as remarks below.

Remark 1. From Theorem 3.2, if we choose $\sigma_n = (n\lambda_n)^{-1/2d(1/p+1/4)}$, it gives

$$\mathcal{PR}(\hat{g}) - \mathcal{PR}(g^*) = O_p \{ \lambda_n + (n\lambda_n)^{-q} \},$$

where $q = 1/(4/p + 1)$ and g^* is the function minimizing $\mathcal{PR}(g)$.

Remark 2. Furthermore, if we choose $\lambda_n = n^{-q/(q+1)}$, then the optimal rate obtained from Theorem 3.2 becomes

$$\mathcal{PR}(\hat{g}) - \mathcal{PR}(g^*) = O(n^{-q/(q+1)}).$$

4. Numeric Examples

In this section, we first present simulation results comparing SVHM to existing machine learning approaches and semiparametric approaches based on the Cox proportional hazards regression. Next, we provide applications to two real world empirical studies.

4.1 Simulation Studies

In all scenarios, we generated both event times and censoring times to be dependent on the covariates. First we simulated five covariates $\mathbf{Z} = (Z_1, \dots, Z_5)$ which are marginally normal $N(0, 0.5^2)$ with pairwise correlation $\text{corr}(Z_j, Z_k) = \rho^{|j-k|}$, and $\rho = 0.5$. The event times were generated from the Cox proportional hazards model with true $\beta = (2, -1.6, 1.2, -0.8, 0.4)^T$ and the exponential distribution with $\lambda = 0.25$ was assumed to compute the baseline cumulative hazard function $\Lambda(t) = \int_0^t \lambda_0(s)ds$, where $\lambda_0(s)$ is the baseline hazard function. We simulated two types of censoring distributions. In the first type, the censoring times were generated from an accelerated failure time model following the log-normal distribution, that is, $\log C \sim N(\mathbf{Z}^T \beta_c + a, 0.5^2)$, with true $\beta_c = (1, 1, 1, 1, 1)^T$. In the second type, the distribution of the censoring times follows the Cox proportional hazards model with true $\beta_c = (1, 1, 1, -2, -2)^T$ and the baseline cumulative hazard function $\Lambda_c(t) = bt$ ($b > 0$). The parameters a and b were chosen to obtain the desired censoring ratio. We considered the censoring ratios 40% and 60%. Any event times or censoring times greater than u_0 were truncated at u_0 , where u_0 is the 90th percentile of the event times. Moreover, we explored some generalizations of the above scenarios to include more covariates in the regression models and include additional noise variables. Besides these training data sets (with a sample size of 100 or 200), we use a randomly generated testing data set with 10,000 subjects in each scenario with no censoring to evaluate prediction performance of various methods.

For all scenarios, we compared SVHM with the modified support vector regression for right censored data based on the ranking constraints (modified SVR) (Van Belle et al., 2011) and the inverse-probability-of-censoring weighting with censoring distribution estimated using Kaplan-Meier (IPCW-KM) or estimated under a Cox model (IPCW-Cox) (Goldberg and Kosorok, 2013), whose objective function is defined as

$$n^{-1} \sum_{i=1}^n \frac{\Delta_i}{\tilde{S}(Y_i)} (\log Y_i - \mathbf{x}^T \phi(\mathbf{X}_i))^2$$

with $\tilde{S}(t)$ is the estimated survival probability for the censoring time. We used linear kernel $K(x, x') = \mathbf{x}^T \mathbf{x}'$ in all four methods, and used 5-fold cross-validation to choose the tuning parameters from the grid of $\{2^{-16}, 2^{-15}, \dots, 2^{15}, 2^{16}\}$. As the model comparison criterion, we adapted mean squared error to censored data, which sums up the mean squared differences between the fitted event times and observed event times for uncensored subjects. For censored subjects, we sum up the squared differences between fitted times and censoring times if the former is smaller than the latter. Essentially, for these censored subjects, if their predicted event times were less than the observed censoring times, we imposed a penalty to measure how much under-estimation there is. The mean squared differences were assumed to be zero for censored subjects if their predicted values were greater than the observed censoring times. We divided the total sum of squares by the total number of observations. We repeated the simulation 500 times, since our results show that 500 repetitions are sufficient to obtain stable simulation results to draw conclusions on comparing performance of different methods while still achieving computational efficiency.

Table 1 and 2 give the average Pearson correlations and root mean square errors (RMSE) $\{\sum(\hat{T} - T)^2\}^{1/2}$ based on the fitted event times and observed event times T on the testing data set. Larger correlation and smaller root mean squared error indicate better performance. The results show that SVHM outperforms the other methods for all the simulation cases and sample sizes. The advantages are not affected by including 5 or 15 noise variables, and the improvements become more evident when the censoring rate is 60% or the censoring distribution follows the accelerated failure time model. The columns of the average correlations show that the modified SVR has similar capability to capture the rank information as SVHM. However, it gives less accurate prediction of the exact event times as measured by the higher RMSEs. The IPCW methods have the worst performance, no matter using the Kaplan-Meier estimator or Cox model to estimate the censoring distribution, even when the censoring distribution follows the Cox model. The performances of all the methods are improved as the sample size increases from 100 to 200, and the proposed SVHM has the largest improvement with respect to the ratios of the average RMSEs. The RMSE of SVHM is significantly lower than the best competing method in all simulation settings in Table 1 and 2. Correlation between the risk scores and event times for SVHR is not significantly different from modified SVR, but in the first simulation setting it is significantly higher than two IPW-based methods except when there are 95 noise variables (Table 1). In the second simulation setting, difference between SVHR and IPW is smaller, with the former significantly greater for most cases with $n = 200$ (Table 2).

In conclusion, Table 1 shows that SVHM performs much better than Cox regression when the model assumption does not hold, and Table 2 shows that SVHM still maintains its advantage when the Cox proportional hazards assumption holds. This advantage may be due to that Cox model aims at maximizing the likelihood while SVHM directly aims at discriminating individual's risk and prediction.

We also explored SVHM with a Gaussian kernel for the sample size 100 and the computation is more intensive. The resulting average correlations and RMSEs are similar to those for linear kernel. For example, under the setting in Table 1 with 60% censoring rate, no noise variable and $n = 100$, using Gaussian kernel yields almost similar correlation of 0.48, 0.10, 0.15, and 0.53 for four competing methods (modified SVR, IPCW-KM, IPCW-Cox, SVHM), respectively. The corresponding RMSEs are 6.03, 6.62, 6.75, and 5.26, respectively

Censoring Noises	# of Noises	Method	$n = 100$			$n = 200$				
			CORR ^a	RMSE ^b (SD) ^c	Ratio ^d	CORR	RMSE (SD)	Ratio		
40%	0	Modified SVR	0.59	5.59 (0.60)	1.19	0.62	5.58 (0.58)	1.24		
		IPCW-KM ^e	0.40	5.60 (0.52)	1.20	0.45	5.45 (0.41)	1.21		
		IPCW-Cox ^f	0.43	5.80 (0.64)	1.24	0.50	5.62 (0.57)	1.25		
	5	SVHM	0.61^g	4.68 (0.27)	1.00	0.64	4.49 (0.17)	1.00		
		Modified SVR	0.55	5.64 (0.60)	1.15	0.61	5.63 (0.57)	1.22		
		IPCW-KM	0.32	5.93 (0.47)	1.21	0.42	5.63 (0.44)	1.22		
		IPCW-Cox	0.33	6.17 (0.54)	1.26	0.44	5.87 (0.57)	1.27		
		SVHM	0.58	4.90 (0.35)	1.00	0.63	4.62 (0.20)	1.00		
		Modified SVR	0.46	5.73 (0.47)	1.10	0.54	5.55 (0.50)	1.15		
		IPCW-KM	0.21	6.12 (0.32)	1.18	0.31	5.86 (0.34)	1.22		
95 ^h	15	IPCW-Cox	0.20	6.47 (0.46)	1.24	0.32	6.09 (0.47)	1.26		
		SVHM	0.48	5.20 (0.36)	1.00	0.57	4.82 (0.23)	1.00		
		Modified SVR	0.21	6.65 (0.89)	1.10	0.30	6.29 (0.47)	1.09		
	95 ^h	IPCW-KM	0.06	6.33 (0.21)	1.05	0.10	6.28 (0.14)	1.09		
		IPCW-Cox	0.08	6.59 (0.23)	1.09	0.11	6.61 (0.39)	1.15		
		SVHM	0.22	6.04 (0.32)	1.00	0.32	5.76 (0.25)	1.00		
		60%	0	Modified SVR	0.55	6.00 (0.54)	1.16	0.60	6.07 (0.42)	1.24
				IPCW-KM	0.15	6.45 (0.41)	1.25	0.18	6.42 (0.37)	1.32
				IPCW-Cox	0.21	6.56 (0.47)	1.27	0.26	6.47 (0.48)	1.33
			5	SVHM	0.57	5.18 (0.43)	1.00	0.61	4.88 (0.33)	1.00
Modified SVR	0.50			6.06 (0.53)	1.12	0.57	6.07 (0.50)	1.21		
IPCW-KM	0.11			6.61 (0.34)	1.22	0.15	6.56 (0.32)	1.31		
IPCW-Cox	0.15			6.77 (0.39)	1.25	0.21	6.66 (0.39)	1.33		
SVHM	0.51			5.40 (0.48)	1.00	0.58	5.02 (0.33)	1.00		
Modified SVR	0.39			6.14 (0.45)	1.10	0.49	5.97 (0.45)	1.15		
IPCW-KM	0.07			6.56 (0.30)	1.17	0.10	6.54 (0.24)	1.26		
95	15	IPCW-Cox	0.10	6.82 (0.30)	1.22	0.13	6.70 (0.27)	1.29		
		SVHM	0.40	5.60 (0.44)	1.00	0.51	5.20 (0.36)	1.00		
		Modified SVR	0.17	6.90 (1.08)	1.11	0.25	7.20 (1.52)	1.21		
	95	IPCW-KM	0.01	6.53 (0.26)	1.05	0.03	6.54 (0.20)	1.10		
		IPCW-Cox	0.02	6.87 (0.20)	1.10	0.04	6.86 (0.21)	1.15		
		SVHM	0.17	6.22 (0.24)	1.00	0.26	5.94 (0.25)	1.00		

^a CORR, average value of correlations.^b RMSE, average value of root mean square errors.^c Empirical standard deviation of the RMSE across 500 repetitions^d Ratio, ratio of average root mean square errors between the method used and our method.^e IPCW-KM, IPCW using the Kaplan-Meier estimator for the censoring distribution.^f IPCW-Cox, IPCW using the Cox model for the censoring distribution.^g Entries in boldface highlight the best performance method.^h For the cases of 95 noises, the calculation of inverse weights in the IPCW-Cox method uses only five signal variables to fit the Cox model for the censoring times.

Table 1: Comparison of four support vector learning methods for right censored data using a linear kernel, with censoring times following the accelerated failure time (AFT) model

Censoring Noises	# of Noises	Method	$n = 100$			$n = 200$				
			CORR ^a	RMSE ^b (SD) ^c	Ratio ^d	CORR	RMSE (SD)	Ratio		
40%	0	Modified SVR	0.59	5.15 (0.59)	1.11	0.62	5.09 (0.54)	1.12		
		IPCW-KM ^e	0.53	5.16 (0.42)	1.11	0.55	5.08 (0.31)	1.12		
		IPCW-Cox ^f	0.52	5.31 (0.57)	1.14	0.56	5.09 (0.46)	1.12		
	5	SVHM	0.61^g	4.66 (0.25)	1.00	0.63	4.53 (0.16)	1.00		
		Modified SVR	0.56	5.28 (0.51)	1.08	0.61	5.09 (0.50)	1.12		
		IPCW-KM	0.46	5.58 (0.42)	1.14	0.52	5.27 (0.34)	1.13		
		IPCW-Cox	0.44	5.73 (0.52)	1.17	0.51	5.41 (0.51)	1.16		
		SVHM	0.58	4.89 (0.29)	1.00	0.62	4.65 (0.18)	1.00		
		Modified SVR	0.47	5.43 (0.40)	1.04	0.55	5.14 (0.38)	1.06		
		IPCW-KM	0.36	5.79 (0.34)	1.11	0.44	5.49 (0.30)	1.13		
95 ^h	15	IPCW-Cox	0.34	6.00 (0.40)	1.15	0.42	5.70 (0.43)	1.18		
		SVHM	0.49	5.21 (0.33)	1.00	0.57	4.84 (0.20)	1.00		
		Modified SVR	0.21	6.43 (0.92)	1.04	0.33	6.03 (0.54)	1.04		
	95 ^h	IPCW-KM	0.17	6.16 (0.21)	1.00	0.24	6.06 (0.18)	1.05		
		IPCW-Cox	0.16	6.32 (0.23)	1.02	0.22	6.21 (0.22)	1.07		
		SVHM	0.23	6.18 (0.40)	1.00	0.34	5.78 (0.24)	1.00		
		60%	0	Modified SVR	0.56	5.43 (0.56)	1.08	0.59	5.43 (0.47)	1.12
				IPCW-KM	0.44	5.68 (0.43)	1.13	0.46	5.62 (0.33)	1.16
				IPCW-Cox	0.42	5.83 (0.56)	1.16	0.47	5.67 (0.48)	1.17
			5	SVHM	0.57	5.01 (0.37)	1.00	0.60	4.85 (0.25)	1.00
Modified SVR	0.50			5.61 (0.48)	1.07	0.57	5.40 (0.46)	1.09		
IPCW-KM	0.36			6.02 (0.38)	1.15	0.43	5.79 (0.35)	1.17		
IPCW-Cox	0.34			6.25 (0.44)	1.20	0.41	5.96 (0.47)	1.20		
SVHM	0.53			5.23 (0.37)	1.00	0.59	4.96 (0.27)	1.00		
Modified SVR	0.40			5.77 (0.42)	1.05	0.50	5.44 (0.38)	1.06		
IPCW-KM	0.27			6.07 (0.31)	1.10	0.35	5.94 (0.26)	1.16		
95	15	IPCW-Cox	0.25	6.39 (0.40)	1.16	0.32	6.16 (0.33)	1.20		
		SVHM	0.42	5.51 (0.40)	1.00	0.52	5.13 (0.29)	1.00		
		Modified SVR	0.18	6.47 (0.87)	1.05	0.27	6.31 (0.80)	1.05		
	95	IPCW-KM	0.12	6.22 (0.29)	1.01	0.18	6.19 (0.21)	1.03		
		IPCW-Cox	0.12	6.54 (0.26)	1.07	0.16	6.50 (0.23)	1.08		
		SVHM	0.20	6.14 (0.38)	1.00	0.28	6.00 (0.35)	1.00		

^a CORR, average value of correlations.^b RMSE, average value of root mean square errors.^c Empirical standard deviation of the RMSE across 500 repetitions^d Ratio, ratio of average root mean square errors between the method used and our method.^e IPCW-KM, IPCW using the Kaplan-Meier estimator for the censoring distribution.^f IPCW-Cox, IPCW using the Cox model for the censoring distribution.^g Entries in boldface highlight the best performance method.^h For the cases of 95 noises, the calculation of inverse weights in the IPCW-Cox method uses only five signal variables to fit the Cox model for the censoring times.

Table 2: Comparison of four support vector learning methods for right censored data using a linear kernel, with censoring times following the Cox proportional hazards model

for each method. Under the setting in Table 2 with 60% censoring rate, no noise variable and $n = 100$, the correlations for the four methods are 0.52, 0.42, 0.40, and 0.55, respectively, and the RMSEs are 5.52, 5.76, 5.94, and 5.06.

In our next simulation experiment, we compare SVHM with Cox model based analysis and explore 1-nearest-neighbor (1-NN) prediction and the average of 3-nearest-neighbors (3-NN) prediction. In the first setting we generate five discrete covariates $\mathbf{Z} = (Z_1, \dots, Z_5)$ with equal probability of taking each value: Z_1 takes values $-5, -4, -2, -1$ or 0 ; Z_2 takes values $-1, 0$ or 1 ; Z_3 takes integer values 1 to 10 ; Z_4 has a correlation of 0.5 with Z_1 and is also correlated with a random normal noise variable $N(0, 0.5)$, and Z_5 has a correlation of 0.3 with Z_1 and is also correlated with a random uniform noise variable $U(0, 0.5)$. Similar to the previous simulations, the event times were generated from Cox proportional hazards model with true $\beta = (-2, -1.6, 1.2, -0.8, 0.4)^T$ and the exponential distribution with $\lambda = 0.25$ was assumed for the baseline cumulative hazard function $\Lambda(t)$. The distribution of the censoring times followed Cox proportional hazards model with true $\beta_c = (1, 1, 1, -2, -2)^T$ and the baseline hazard rate was a constant. In the second setting, we generated Z_1, \dots, Z_3 independently from $U(0, 1)$ and Z_4 from a binary distribution with $P(Z_4 = 1) = P(Z_4 = -1) = 0.5$. Furthermore, both the event times and censoring times were generated from accelerated failure time models with both main effects and interactions:

$$\log T = -0.2 - 0.5Z_1 + 0.5Z_2 + 0.3Z_3 + 0.5Z_4 - 0.1Z_1Z_4 - 0.6Z_2Z_4 + 0.1Z_3Z_4 + N(0, 1),$$

$$\log C = 0.5 - 0.8Z_1 + 0.4Z_2 + 0.4Z_3 + 0.5Z_4 - 0.1Z_1Z_4 - 0.6Z_2Z_4 + 0.3Z_3Z_4 + N(0, 1).$$

The censoring ratio was around 30% in both settings. We experimented two sample sizes, 100 or 200, and two numbers of noise variables, 10 or 30.

The simulation results are summarized in Table 3. The same 1-NN or 3-NN method was applied to predict event times using the fitted scores derived from SVHM or Cox model. We can see that 1-NN performs slightly better than 3-NN in terms of a higher correlation and lower RMSE for both methods. In addition, when the event times were simulated from the Cox model, SVHM with 1-NN or 3-NN performs similarly to Cox model-based analysis. This is expected since proportional hazards assumption was satisfied for the Cox model based method. We also compared using 1-NN and 3-NN for prediction with using median survival times under a Cox model. We see 1-NN with SVHM or 1-NN with Cox model leads to superior performance than using median survival time. When the true model for the event times was accelerated failure time model (AFT), SVHM outperforms Cox model based analysis in terms of a higher correlation and lower RMSE. In the AFT model case, using the median survival time from the Cox model for prediction tends to be less accurate since the model assumption does not hold. Lastly, when the number of noise variables was 95, Cox model analysis did not converge in most simulations and thus the results were not included. In summary, results in Table 3 show that nearest neighbor based prediction rule performs better than using median survival time, and SVHM performs better than Cox model based methods when the model assumption does not hold.

4.2 PREDICT-HD Study

In the first real data analysis, we apply various methods to a study on Huntington's disease (HD). HD is a severe dominant genetic disorder for which at risk subjects can be identified

Model	n	Index	Cox Model			SVHM	
			1-NN	3-NN ^a	Median ^b	1-NN	3-NN
Cox1 ^c	100	CORR	0.871	0.859	0.866	0.863	0.851
		RMSE	6.068	6.485	6.487	6.099	6.503
	200	CORR	0.896	0.890	0.871	0.885	0.879
		RMSE	5.755	6.168	6.226	5.781	6.186
Cox2 ^d	100	CORR	0.841	0.831	0.839	0.854	0.844
		RMSE	6.146	6.548	6.546	6.139	6.546
	200	CORR	0.887	0.884	0.855	0.883	0.879
		RMSE	5.760	6.209	6.273	5.761	6.214
AFT1 ^e	100	CORR	0.210	0.211	0.192	0.224	0.224
		RMSE	0.766	0.756	0.950	0.739	0.731
	200	CORR	0.275	0.275	0.262	0.277	0.277
		RMSE	0.720	0.717	0.879	0.709	0.706
AFT2 ^f	100	CORR	0.129	0.129	0.110	0.174	0.175
		RMSE	0.859	0.841	1.050	0.753	0.745
	200	CORR	0.197	0.197	0.175	0.221	0.222
		RMSE	0.778	0.774	0.999	0.732	0.729

^a Using mean of 3 nearest neighbors as predicted event time

^b Using median survival time fitted from a Cox model as predicted event time.

^c T and C simulated from Cox model with 10 noise variables.

^d T and C simulated from Cox model with 30 noise variables.

^e T and C simulated from AFT model with 10 noise variables.

^f T and C simulated from AFT model with 10 noise variables.

Table 3: Comparison of SVHM with Cox model based methods

through a genetic testing of C-A-G expansion status at the IT15 gene (MacDonald et al., 1993). The availability of genetic testing and virtually complete penetrance of gene provides opportunity for early intervention. Currently a major research interest in HD is to combine salient clinical markers and biological markers sensitive enough to detect early indicators of patient disease diagnosis before evident clinical signs of HD emerge, and thus inform early interventions long before the clinical diagnosis. The hope of such early detection and intervention is to alter the disease course before substantial damage has occurred. The most promising markers thus far are brain imaging biomarkers and some cognitive markers which correlate with future clinical diagnosis (Paulsen, 2011; Paulsen et al., 2014).

We perform analysis using data collected in the PREDICT-HD study (Paulsen et al. 2008b; data available through dbGap: http://www.ncbi.nlm.nih.gov/projects/gap/cgi-bin/study.cgi?study_id=phs000222.v3.p2). PREDICT-HD is by far the largest and most comprehensive study of prodromal HD subjects that collects clinical, cognitive and structural MRI imaging biomarkers predictive of HD onset. Pre-manifest HD subjects in the absence of experimental treatment were recruited and followed to monitor HD symptom progression and assess HD onset. In our analyses, there were 647 subjects and 118 of them developed HD during the course of study. For each subject, a wide range of measures on motor, psychiatric, cognitive signs as well as MRI imaging markers were collected at the baseline visit. The covariates cover important clinical, cognitive, functional, psychiatric and imaging domains of HD including CAP score (a combination of age and C-A-G repeats length, Zhang et al. (2011)), symbol digital modality test (SDMT), STROOP color, word and interference tests, total functional capacity scores, UHDRS total motor scores, various SCL-90 psychiatric scores, demographic variables such as gender and education in years, and imaging measures based on regional brain volumes. The structural MRI T1-weighted imaging analysis of subcortical and cortical segmentations and cortical parcellations were based on a customized FreeSurfer 5.2 pipeline developed at The University of Iowa. The details of imaging preprocessing and analysis are available in the online Supplementary Material of Paulsen et al. (2014). The subcortical volumetric measures of interest include nucleus accumbens, caudate, putamen, hippocampus, and thalamus (Paulsen et al., 2014).

We study the combined prediction capability of 31 baseline markers predicting the age-at-onset of HD diagnosis during the study period, and evaluate the usefulness of the fitted prediction score on performing risk stratification. The covariates are normalized to the same scale to achieve numeric stability and allow for comparing their relative importance. The predicted values of HD onset ages are obtained via three-fold cross validation, and the cost tuning parameter is chosen from the grid $2^{-16}, 2^{-15}, \dots, 2^{16}$. We consider both linear kernel and Gaussian kernel. For the Gaussian kernel $K(x, x') = \exp(-\gamma \|x - x'\|^2)$, the parameter γ is fixed to be 0.005. To compare the prediction capability, we compute several quantities using the predicted values of onset ages and the observed onset ages or censoring ages. Specifically, we report the concordance index defined as the percentage of correctly ordered pairs among all feasible pairs (C-index) when including imaging markers. To evaluate the ability of the fitted scores on performing risk stratification, we separated subjects into two groups (high risk versus low risk group) based on whether their predicted scores are higher or lower than given percentiles computed from all subjects' fitted scores. We then calculate the Chi-square statistics from the logrank test and the hazard ratios comparing the hazard rate of developing HD between two groups.

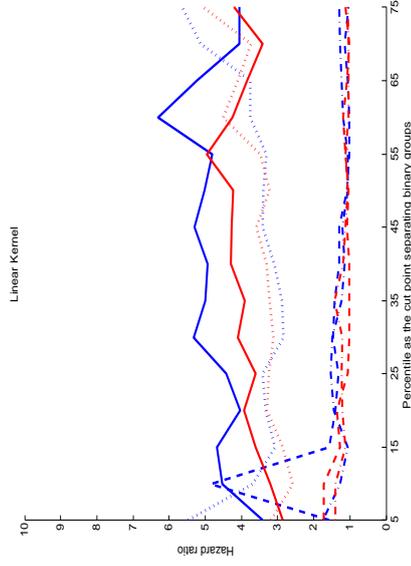


Figure 1: Hazard ratios comparing two groups separated using percentiles of predicted scores as cut points for PREDICT-HD data with linear kernel. Blue curves obtained from analyses with MRI imaging biomarkers and red curves without imaging biomarkers. Solid curve: SVHM; Dotted curve: Modified SVR; Dashed curve: IPCW-KM; Dashed-dotted curve: IPCW-Cox.

The analysis results are given in Table 4. SVHM improves over the other methods with respect to all the quantities for both linear kernel and Gaussian kernel, and the performances are similar using different kernels. For example, the logrank Chi-square statistics and hazard ratios of SVHM are much larger than the competing methods at most quantiles except at the right tail (e.g., over 65th percentile). A higher value of logrank Chi-square and a larger hazard ratio indicate greater difference between high risk and low risk subjects using a given percentile as a cut off value, and thus better discriminant ability of a risk score distinguishing high/low risk subjects. In addition, the predictions from IPCW cannot capture the trend of the original disease onset ages. Figure 1 complements the results in the table by plotting the hazard ratios comparing two groups separated using a series of percentiles of the predicted scores as cut points, and SVHM consistently has the largest hazard ratio across all percentiles among all methods. The improvement of SVHM increases at the higher percentiles, indicating that it is particularly effective in discriminating high risk subjects. This observation is consistent with our theoretical results which reveal that SVHM is optimal in separating the individual covariate-specific hazard function, $h(t, \mathbf{x})$ given \mathbf{x} , from the population average hazard function, $h(t)$.

Additionally, we show the fitted coefficients from SVHM and other competing methods in Table 5 and compare with coefficients obtained from a Cox proportional hazards model.

Imaging ^a	Method	C-index	25th percentile		50th percentile		75th percentile	
			Logrank χ^2 ^b	HR ^c	Logrank χ^2	HR	Logrank χ^2	HR
No	Modified SVR	0.70	43.55	3.24	25.85	3.23	15.53	5.07
	IPCW-KM	0.47	0.04	1.05	0.04	1.04	0.31	1.12
	IPCW-Cox	0.48	1.05	1.23	0.05	1.05	0.08	1.06
	SVHM (Linear) ^d	0.73	53.41	3.61	32.72	4.22	9.02	4.21
Yes	Modified SVR	0.70	47.85	3.41	38.36	3.40	27.44	5.65
	IPCW-KM	0.47	0.76	1.31	0.14	1.08	0.04	1.04
	IPCW-Cox	0.47	1.47	1.57	0.07	1.05	0.90	1.49
	SVHM (Linear)	0.74	71.26	4.42	46.73	5.02	14.75	4.06
	SVHM (Gaussian) ^e	0.79	105.99	5.86	67.66	7.44	25.31	7.18

^a Whether structural MRI imaging biomarkers were included in the analysis.

^b Logrank χ^2 : Chi-square statistics from Logrank tests for two groups separated using the 25th percentile, 50th percentile, and 75th percentile of predicted values.

^c HR: Hazard Ratios comparing two groups separated using the 25th percentile, 50th percentile, and 75th percentile of predicted values.

^d SVHM with linear kernel.

^e SVHM with Gaussian kernel.

Table 4: Comparison of prediction capability for different methods using PREDICT-HD data with and without structural MRI imaging measures ($n = 647$)

Modified SVR yields coefficients in the same direction as SVHM, while two IPW methods give several coefficients in the opposite direction of other methods. SVHM suggests the top ranking markers with largest standardized effects to be the baseline total motor score and CAP score, which is consistent with the clinical literature on the importance of these markers on the diagnosis of HD (Zhang et al., 2011; Paulsen et al., 2014; Chen et al., 2014). The baseline total motor score as a measure of motor impairment appears to be more informative than CAP score in terms of predicting future HD diagnosis during the study. Several neuropsychological markers (Stroop color, Stroop word, SDMT) are predictive except for Stroop interference score. The coefficients from Cox model however, suggest that SDMT is not important, which is not consistent with the clinical literature (Paulsen, 2011; Paulsen et al., 2014). Note that SVHM gives psychiatric markers (SCL 90 depression, GSI, PST and PSDI) low weights which is consistent with clinical observations that the psychiatric markers are considered as noisy for predicting HD diagnosis due to reasons such as subjects may seek treatment for their psychiatric symptoms. In contrast, Cox model yields high weights for these markers which are deemed to be less informative.

In terms of neuroimaging markers, we see that pallidum, putamen, caudate, and thalamus show relatively strong predictive ability of HD onset, while accumbens and hippocampus show low predictive ability. Comparing SVHM and Cox model analysis, note that SVHM provides coefficients with similar magnitude for imaging measures on the left and right side of the same brain region, but Cox model sometimes produces substantially different results for left and right side. For example, left pallidum area is significant but not right pallidum area in Cox model. This observation suggests that SVHM may lead to more interpretable results especially for correlated variables. Another biomarker, cerebral spinal fluid, appears to be promising for predicting HD onset with a coefficient with moderate magnitude. To assess the added value of MRI imaging measures in terms of risk stratification, in Figure 1 we show the hazard ratio comparing high risk versus low risk group based on percentile split of the fitted scores obtained with and without imaging biomarkers. For SVHM with linear kernel, adding imaging measures leads to a larger hazard ratio and a greater difference between high and low risk groups at all percentiles, which demonstrates the ability of SVHM to extract information from imaging biomarkers and corroborates other findings suggesting their added values in predicting HD onset (Paulsen et al., 2014). When using Gaussian kernel for SVHM, we see further improvement of C-index and logrank chi-square statistics. Other methods such as modified SVR or IPCW do not show an advantage from including imaging measures, which may suggest their limitations in handling correlated biomarkers.

4.3 ARIC Study

As a second real world numeric example, we analyze data from the Atherosclerosis Risk in Communities Study, a prospective investigation of the aetiology of atherosclerosis and its clinical sequelae, as well as the variation in cardiovascular risk factors, medical care and disease by race, gender, location and date (The ARIC investigators, 1989; Lubin et al., 2016). We assess the prediction capability of some common cardiovascular risk factors for incident heart failure until 2005. Specifically, these risk factors include age, diabetes status, body mass index, systolic blood pressure, fasting glucose, serum albumin, serum creatinine, heart

rate, left ventricular hypertrophy, bundle branch block, prevalent coronary heart disease, valvular heart disease, high-density lipoprotein, pack-years of smoking, and current and former smoking status. The analysis sample consists of 624 participants who are African-American males living in Jackson, Mississippi. Incident heart failure occurred in 133 men through 2005, with a median follow-up time 16.2 years. Among those participants who did not develop heart failure, 324 were administratively censored on December 31st, 2005. The analysis follows the same procedure as in Section 4.2. The results for prediction capability of different methods are given in Table 6. SVHM provides more accurate prediction than other methods using the linear kernel or Gaussian kernel. It also has higher logrank test statistic and hazard ratio comparing high risk versus low risk group using various percentiles of the predictive scores as cut off points (Figure 2).

In Table 7, we can see that all the risk factors have positive effects on the incident heart failure except high-density lipoprotein, serum albumin and former smoking status. Risk factors for incident heart failure with largest standardized effects include HDL, age, prevalent CHD, and serum albumin level. We also present estimated coefficients from a Cox proportional hazards model as comparison in Table 7. Most coefficients are comparable in terms of size. However, note that higher fasting glucose level appears to be protective of heart rate failure using Cox model, which is the opposite of the expected direction.

5. Concluding Remarks

In this paper, we propose a new statistical framework to learn risk scores for event times using right-censored data by support vector hazards machine. We propose to view the prediction of time-to-event outcomes from a counting process point of view to avoid complications from specifying a censoring distribution. Asymptotically, we justify the associated universal consistency and learning rate through the structural risk minimization and show a natural link between the fitted decision function and the true hazard function: the fitted decision rule asymptotically minimizes the integrated difference between the covariate-specific hazard function and population average hazard function. Our theory shows that SVHM essentially compares events and non-events among the subjects at risk at each follow-up time; therefore, SVHM is sensitive to temporal difference between events and non-events which may not be reflected in either SVR or inverse weighted approaches. We also reveal a theoretical connection between SVHM and Cox partial likelihood function; the proposed method uses a hinge loss which should be robust to extreme observations in contrast to the exponential loss used in Cox partial likelihood. The simulation studies and real data applications demonstrate satisfactory results in finite samples with improved overall risk prediction accuracy in the presence of noise variables compared to other methods, especially when the censoring rate is high and the distribution of censoring times is unknown. The improved performance of our method is due to introducing counting processes to represent the time-to-event data, which leads to an intuitive connection of the method with both support vector machines in standard supervised learning and hazard regression models in standard survival analysis.

Since SVHM essentially learns hazard functions across subjects conditional on each risk set, the intercept term, $\alpha(t)$, is a non-informative nuisance parameter and allowed to be discontinuous over time. This feature is analogous to the estimation in Cox regression

Variable	Modified SVR ($\times 10^{-1}$)	IPCW-KM ($\times 10^{-2}$)	IPCW-Cox ($\times 10^{-3}$)	SVHM	Cox model ^a
CAP	0.051	-0.936	0.202	0.255	0.058
TOTAL MOTOR SCORE	0.280	-1.083	0.529	0.519	0.308*
SDMT	-0.096	-0.411	0.076	-0.119	-0.190
STROOP COLOR	-0.042	0.412	-0.038	-0.153	-0.160
STROOP WORD	-0.227	0.488	0.188	-0.191	-0.217
STROOP INTERFERENCE	0.254	-0.432	-0.239	-0.000	0.328
TOTAL FUNCTIONAL CAPACITY	-0.062	0.175	0.142	-0.072	0.007
UHDRS PSYCH	0.168	0.137	-0.280	0.155	0.228
SCL90 DEPRESS	-0.285	-0.255	-0.132	0.064	-0.618*
SCL90 GSI	0.316	-0.184	-0.182	0.007	0.618
SCL90 PST	-0.108	-0.265	-0.246	-0.057	-0.268
SCL90 PSDI	0.099	-0.379	-0.249	0.103	0.035
FRSBE TOTAL	-0.088	0.108	-0.136	0.112	0.115
Education Years	0.019	-1.057	0.349	-0.016	-0.053
Gender (Male)	0.178	-0.394	-0.202	0.376	0.344*
Right Putamen	-0.009	-0.395	-0.376	-0.134	-0.038
Left Putamen	-0.590	-0.165	-0.210	-0.116	-0.369
Right Pallidum	-0.015	-0.490	-0.151	-0.225	-0.049
Left Pallidum	-0.329	0.100	-0.189	-0.261	-0.626*
Right Caudate	-0.830	0.655	-0.160	-0.147	-0.943*
Left Caudate	0.397	0.738	-0.265	-0.079	0.306
Right Accumbens	0.282	-0.214	-0.470	0.051	0.220
Left Accumbens	-0.256	-0.568	-0.487	-0.057	-0.467*
Right Thalamus	0.099	-0.295	-0.710	0.172	0.260
Left Thalamus	0.258	-0.404	-0.636	0.219	0.138
Right Hippocampus	0.103	-1.152	-0.821	0.010	0.095
Left Hippocampus	-0.130	-1.087	-0.847	-0.082	-0.128
Third Ventricle	-0.101	1.071	-0.841	-0.042	-0.046
Right Lateral Ventricle	0.140	2.794	1.409	-0.119	-0.016
Subcortical Gray Area	0.932	-0.868	-0.691	0.307	1.473*
Cerebral Spinal Fluid	-0.268	0.116	0.954	-0.113	-0.104

^a The estimates from Cox model with significant p -values ($p < 0.05$) are marked with *.

Table 5: Normalized coefficients estimated from PREDICT-HD data (including imaging biomarkers) using Modified SVR, IPCW-KM, IPCW-Cox, SVHM with linear kernel and Cox model

Kernel	Method	C-index	25th percentile		50th percentile		75th percentile	
			Logrank χ^2 ^a	HR ^b	Logrank χ^2	HR	Logrank χ^2	HR
Linear	Modified SVR	0.74	90.52	4.63	59.11	4.16	31.85	5.01
	IPCW-KM	0.69	54.90	3.48	29.53	2.64	22.92	3.45
	IPCW-Cox	0.71	48.34	3.24	39.70	3.12	27.63	4.32
	SVHM	0.76	95.09	4.78	67.06	4.63	34.93	5.36
Gaussian	Modified SVR	0.76	105.10	5.12	70.41	4.87	37.66	6.39
	IPCW-KM	0.70	58.15	3.61	33.49	2.81	19.61	3.00
	IPCW-Cox	0.72	52.77	3.39	47.10	3.50	27.99	4.37
	SVHM	0.77	111.10	5.31	64.79	4.53	35.60	5.76

^a Logrank χ^2 , Chi-square statistics from Logrank tests for two groups separated using the 25th percentile, 50th percentile, and 75th percentile of predicted values.

^b HR, Hazard Ratios comparing two groups separated using the 25th percentile, 50th percentile, and 75th percentile of predicted values.

Table 6: Comparison of prediction capability for different methods using Atherosclerosis Risk in Communities data

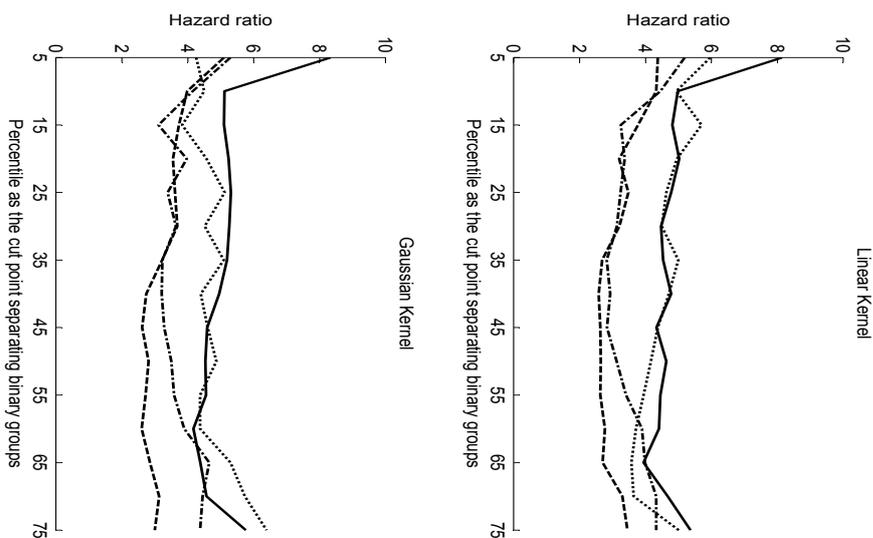


Figure 2: Hazard Ratios comparing two groups separated using percentiles of predicted values as cut points for Atherosclerosis Risk in Communities data. Solid curve: SVHM; Dotted curve: Modified SVR; Dashed curve: IPCW-KM; Dash-dot-dotted curve: IPCW-Cox.

through maximizing the Cox partial likelihood function, where the baseline hazard function is estimated to be non-continuous. Furthermore, due to the martingale property of the counting process, data from each time point can be viewed as independent in the learning method, despite that they may be from the same individual. Thus, we expect little efficiency loss even though some weighing scheme can be adopted to weight distinct risk sets differently over time.

In the current framework, the time-specific risk score $f(t, \mathbf{X})$ being considered includes a class of additive rules. They can be generalized to be fully nonparametric to learn dynamic risk profiles using a subject's time-varying covariates under the current set up. However, this generalization may lose the similarity of formulation to the standard support vector machines and cause numerical instability in the optimization algorithm. These challenging issues will be further investigated in future work. One limitation of the current nonparametric framework not specifying the event distribution is that no straightforward prediction formulae using distribution exist. We used nearest neighbors to perform prediction and simulation studies show that using less closer neighbors (3-NN instead of 1-NN) has little influence on the results. In our simulation studies, we found that a training sample size of $n = 100$ or $n = 200$ both yield stable estimation of correlation and RMSE (not sensitive to the choice of 1-NN or 3-NN). However, further work is needed to examine alternative prediction methods. Lastly, this work opens possibilities to use other powerful learning algorithms for binary and continuous outcomes to handle censored outcomes. For example, instead of using series of SVM to predict counting process as demonstrated here, other effective tools such as AdaBoosting and random forest can also be used. Gaussian process approaches (Barrett and Coolen, 2013) have been recently applied for survival data with competing risks so it will be interesting to compare SVHM with their approaches in terms of prediction performance and robustness.

Acknowledgments

We wish to acknowledge Dr. XiaoXi Liu's contribution to this work. We would like to acknowledge the National Institute of Health Grants NS073671 and NS082062, the NIH dbGap data repository (phs000222.v3.p2) and the PREDICT-HD investigators.

Covariate	Normalized β	Cox model ^a
Age (in years)	0.363	0.328 *
Diabetes	0.288	0.221 *
BMI (kg/m ²)	0.150	0.136
SBP (mm of Hg)	0.172	0.178
Fasting glucose (mg/dL)	0.173	-0.093
Serum albumin (g/dL)	-0.363	-0.273 *
Serum creatinine (mg/dl)	0.007	0.029
Heart rate (beats/minute)	0.124	0.125
Left ventricular hypertrophy	0.250	0.158 *
Bundle branch block	0.341	0.242 *
Prevalent CHD	0.330	0.216 *
Valvular heart disease	0.200	0.169 *
HDL (mg/dl)	-0.287	-0.436 *
LDL (mg/dl)	0.016	0.051
Pack years of smoking	0.289	0.230 *
Current smoking status	0.210	0.022
Former smoking status	-0.133	-0.232 *

^a The estimates from Cox model with significant p-value (p-value < 0.05) are marked with *.

Table 7: Normalized coefficient estimates using linear kernel for Atherosclerosis Risk in Communities data

Appendix A. SVHM Algorithm

In this section, we provide a detailed description of the SVHM algorithm:

Algorithm: SVHM for Censored Outcomes

Input: Training data $(\mathbf{X}_i, T_i \wedge C_i, Y_i(t_j), \delta N_i(t_j))$ for $i = 1, \dots, n, j = 1, \dots, m$.

Step 1. Solve the quadratic programming problem:

$$\max_{\gamma_{ij}} \sum_{i=1}^n \sum_{j=1}^m \gamma_{ij} Y_i(t_j) - \frac{1}{2} \sum_{i=1}^n \sum_{i'=1}^n \sum_{j=1}^m \sum_{j'=1}^m \gamma_{ij} \gamma_{i'j'} Y_i(t_j) Y_{i'}(t_{j'}) \delta N_i(t_j) \delta N_{i'}(t_{j'}) K(\mathbf{X}_i, \mathbf{X}_{i'})$$

subject to: $0 \leq \gamma_{ij} \leq w_i(t_j) C_{ni}, \sum_{i=1}^n \gamma_{ij} Y_i(t_j) \delta N_i(t_j) = 0, i = 1, \dots, n, j = 1, \dots, m$.

Denote the solutions as $\hat{\gamma}_{ij}$.

Step 2. Compute the risk scores for non-censored subjects in the training data as

$$\hat{g}(\mathbf{X}_s) = \sum_{i=1}^n \sum_{j=1}^m \hat{\gamma}_{ij} \delta N_i(t_j) K(\mathbf{X}_s, \mathbf{X}_i).$$

Step 3. Predicting event time of a future subject with covariates \mathbf{x} by k -nearest-neighbor:

- Compute the risk score for this subject as $\hat{g}(\mathbf{x}) = \sum_{i=1}^n \sum_{j=1}^m \hat{\gamma}_{ij} \delta N_i(t_j) K(\mathbf{x}, \mathbf{X}_i)$.
 - Find k non-censored subjects in the training data whose risk scores are closest to $\hat{g}(\mathbf{x})$ and denote them as $\hat{g}(\mathbf{X}_l)$ for $l = 1, \dots, k$.
 - Sort all $\hat{g}(\mathbf{X}_s)$ in descending order and denote the rank of $\hat{g}(\mathbf{X}_l)$ as r_l .
 - Sort event times T_s of all non-censored subjects in ascending order. Find the r_l -th event time and denote as T_l for $l = 1, \dots, k$.
 - The event time for this subject is predicted as $\hat{T} = \frac{1}{k} \sum_{l=1}^k T_l$.
- Output:** For a subject with covariates \mathbf{x} , predict risk score as $\hat{g}(\mathbf{x})$, and predict event time as \hat{T} .
-

Appendix B. Proof of Theorems

In this section, we prove Theorem 3.1 and Theorem 3.2.

Proof (Theorem 3.1)

Since $f^*(t, \mathbf{x})$ minimizes $\mathcal{R}(f)$, conditional $\mathbf{X} = \mathbf{x}$, $f^*(t, \mathbf{x})$ also minimizes

$$E \left(\int Y(t) [1 - f(t, \mathbf{X})]_+ dN(t) | \mathbf{X} = \mathbf{x} \right) + \int \frac{E(Y(t) [1 + f(t, \mathbf{X})]_+ | \mathbf{X} = \mathbf{x})}{E(Y(t))} E(dN(t)). \quad (A.1)$$

Clearly, the value $f^*(t, \mathbf{x})$ should belong to the interval $[-1, 1]$, because otherwise truncation of f at -1 or 1 gives a lower value. Assuming $-1 \leq f(t, \mathbf{x}) \leq 1$, (A.1) becomes

$$\int E(Y(t) | \mathbf{X} = \mathbf{x}) \{h(t, \mathbf{x}) + \bar{h}(t)\} dt - \int f(t, \mathbf{x}) E(Y(t) | \mathbf{X} = \mathbf{x}) \{h(t, \mathbf{x}) - \bar{h}(t)\} dt,$$

where recall that $h(t, \mathbf{x})$ is the conditional hazard rate of $T = t$ given $\mathbf{X} = \mathbf{x}$ and $\bar{h}(t)$ is the population average hazard at time t ,

$$\bar{h}(t) = \frac{E[dN(t)]/dt}{E[Y(t)]} = E[h(t, \mathbf{X}) | Y(t) = 1].$$

Therefore, one optimal decision function minimizing $\mathcal{R}_L(f)$ is

$$f^*(t, \mathbf{x}) = \text{sign}\{h(t, \mathbf{x}) - \bar{h}(t)\}.$$

On other hand, we note

$$\mathcal{R}_0(f) = \int I[f(t, \mathbf{x}) \leq 0] E(Y(t) | \mathbf{X} = \mathbf{x}) h(t, \mathbf{x}) dt + \int I[f(t, \mathbf{x}) \geq 0] E(Y(t) | \mathbf{X} = \mathbf{x}) \bar{h}(t) dt.$$

Thus, any decision function has the same sign as $(h(t, \mathbf{x}) - \bar{h}(t))$ minimizes $\mathcal{R}_0(f)$ so $f^*(t, \mathbf{x})$ minimizes $\mathcal{R}_0(f)$. Finally, under the optimal rule $f^*(t, \mathbf{x})$, the minimal value of the weighted 0-1 risk is given as

$$\begin{aligned} \mathcal{R}_0(f^*) &= E \left[\int E(Y(t) | \mathbf{X} = \mathbf{x}) \min\{h(t, \mathbf{x}), \bar{h}(t)\} dt \right] \\ &= \frac{1}{2} E \left[\int E(Y(t) | \mathbf{X} = \mathbf{x}) \{h(t, \mathbf{x}) + \bar{h}(t) - |h(t, \mathbf{x}) - \bar{h}(t)|\} dt \right] \\ &= P(T \leq C) - \frac{1}{2} E \left[\int E(Y(t) | \mathbf{X} = \mathbf{x}) |h(t, \mathbf{x}) - \bar{h}(t)| dt \right]. \end{aligned}$$

To show the last inequality in Theorem 3.1, we note that for $-1 \leq f(t, \mathbf{x}) \leq 1$,

$$\begin{aligned} \mathcal{R}(f) &= E \left[\int E(Y(t) | \mathbf{X} = \mathbf{x}) \{h(t, \mathbf{x}) + \bar{h}(t)\} dt - \int f(t, \mathbf{x}) E(Y(t) | \mathbf{X} = \mathbf{x}) \{h(t, \mathbf{x}) - \bar{h}(t)\} dt \right] \\ &= 2P(T \leq C) - E \left[\int f(t, \mathbf{x}) E(Y(t) | \mathbf{X} = \mathbf{x}) \{h(t, \mathbf{x}) - \bar{h}(t)\} dt \right], \end{aligned}$$

and

$$\mathcal{R}(f^*) = 2P(T \leq C) - E \left[\int \text{sign}\{h(t, \mathbf{x}) - \bar{h}(t)\} E(Y(t) | \mathbf{X} = \mathbf{x}) \{h(t, \mathbf{x}) - \bar{h}(t)\} dt \right].$$

Thus,

$$\begin{aligned} \mathcal{R}(f) - \mathcal{R}(f^*) &= E \left[\int E(Y(t) | \mathbf{X} = \mathbf{x}) \{ \text{sign}\{h(t, \mathbf{x}) - \bar{h}(t)\} - f(t, \mathbf{x}) \} \times \{h(t, \mathbf{x}) - \bar{h}(t)\} dt \right] \\ &= E \left[\int E(Y(t) | \mathbf{X} = \mathbf{x}) |f(t, \mathbf{x}) - \text{sign}\{h(t, \mathbf{x}) - \bar{h}(t)\}| \times |h(t, \mathbf{x}) - \bar{h}(t)| dt \right] \end{aligned}$$

On the other hand, for the risk function based on the 0-1 loss, we have

$$\begin{aligned} \mathcal{R}_0(f) - \mathcal{R}_0(f^*) &= E \left[\int E(Y(t) | \mathbf{X} = \mathbf{x}) (I[f(t, \mathbf{x}) \leq 0] h(t, \mathbf{x}) + I[f(t, \mathbf{x}) \geq 0] \bar{h}(t) - \min\{h(t, \mathbf{x}), \bar{h}(t)\}) dt \right] \\ &= E \left[\int E(Y(t) | \mathbf{X} = \mathbf{x}) |h(t, \mathbf{x}) - \bar{h}(t)| \times I\{|h(t, \mathbf{x}) - \bar{h}(t)\} \text{sign}\{f(t, \mathbf{x})\} < 0\} dt \right]. \end{aligned}$$

Note that

$$I(\{h(t, \mathbf{x}) - \bar{h}(t)\} \text{sign}\{f(t, \mathbf{x})\} < 0) \leq |f(t, \mathbf{x}) - \text{sign}\{h(t, \mathbf{x}) - \bar{h}(t)\}|.$$

We then obtain $\mathcal{R}_0(f) - \mathcal{R}_0(f^*) \leq \mathcal{R}(f) - \mathcal{R}(f^*)$. \blacksquare

Proof (Theorem 3.2)

The proof of Theorem 3.2 follows a similar procedure to the standard support vector machine theory. However, the main difference is that the proof handles $\mathcal{P}\mathcal{R}_n(f)$ instead of the simple empirical mean of the hinge-loss in the standard theory. Let g_n be the function in \mathcal{H}_n which minimizes $\lambda_n \|g\|_{\mathcal{H}_n}^2 + \mathcal{P}\mathcal{R}(g)$. The proof consists of the following steps.

First, we derive a preliminary bound for some norms of \hat{g} . Clearly,

$$\lambda_n \|g_{\lambda_n}\|_{\mathcal{H}_n}^2 + \mathcal{P}\mathcal{R}(g_{\lambda_n}) \leq \mathcal{P}\mathcal{R}(0).$$

This gives $\|g_{\lambda_n}\|_{\mathcal{H}_n} \leq \sqrt{c/\lambda}$ for some constant λ_n so by Lemma 4.23 (Steinwart and Christmann, 2008, p124), we obtain $\|g_{\lambda_n}\|_{\infty} \leq \sqrt{c/\lambda_n}$. Furthermore, using the fact

$$\lambda_n \|\hat{g}\|_{\mathcal{H}_n}^2 + \mathcal{P}\mathcal{R}_n(\hat{g}) \leq \lambda_n \|g_{\lambda_n}\|_{\mathcal{H}_n}^2 + \mathcal{P}\mathcal{R}_n(g_{\lambda_n}),$$

we conclude $\|\hat{g}\|_{\mathcal{H}_n} \leq \sqrt{c/\lambda_n}$ so $\|\hat{g}\|_{\infty} \leq \sqrt{c/\lambda_n}$, where c may be another different constant (without confusion, we always use c to denote some constant). Therefore, we can restrict g in the minimization of (2) to be in $\sqrt{c/\lambda_n} \mathcal{B}_{\mathcal{H}_n}$, where $\mathcal{B}_{\mathcal{H}_n}$ be the unit ball in \mathcal{H}_n .

Second, we obtain a key inequality for comparing the risks of \hat{g} and g_{λ_n} . By the definition of \hat{g} , the following fact holds:

$$\begin{aligned} & \lambda_n \|\hat{g}\|_{\mathcal{H}}^2 + \mathcal{P}\mathcal{R}(\hat{g}) - (\lambda_n \|g_{\lambda_n}\| + \mathcal{P}\mathcal{R}(g_{\lambda_n})) \\ & \leq \lambda_n \|\hat{g}\|_{\mathcal{H}}^2 + \mathcal{P}\mathcal{R}(\hat{g}) - (\lambda_n \|g_{\lambda_n}\| + \mathcal{P}\mathcal{R}(g_{\lambda_n})) \\ & \quad - [\lambda_n \|\hat{g}\|_{\mathcal{H}}^2 + \mathcal{P}\mathcal{R}_n(\hat{g}) - (\lambda_n \|g_{\lambda_n}\| + \mathcal{P}\mathcal{R}_n(g_{\lambda_n}))] \\ & = \mathcal{P}\mathcal{R}(\hat{g}) - \mathcal{P}\mathcal{R}_n(\hat{g}) - \{\mathcal{P}\mathcal{R}(g_{\lambda_n}) - \mathcal{P}\mathcal{R}_n(g_{\lambda_n})\}. \end{aligned}$$

From Step 1, we conclude

$$\lambda_n \|\hat{g}\|_{\mathcal{H}}^2 + \mathcal{P}\mathcal{R}(\hat{g}) - (\lambda_n \|g_{\lambda_n}\| + \mathcal{P}\mathcal{R}(g_{\lambda_n})) \leq 2 \sup_{\|g\|_{\mathcal{H}_n} \leq \sqrt{c/\lambda_n}} |\mathcal{P}\mathcal{R}_n(g) - \mathcal{P}\mathcal{R}(g)|. \quad (\text{A.2})$$

We derive a bound for the right-hand side of (A.2). First,

$$\mathcal{P}\mathcal{R}_n(g) - \mathcal{P}\mathcal{R}(g) = (\mathbf{P}_n - \mathbf{P})f_g(Y, \mathbf{X}, \Delta) - \frac{2}{n} \mathbf{P}_n \left\{ \frac{\Delta}{\tilde{\mathbf{P}}_n[I(\tilde{Y} \geq Y)]} \right\},$$

where

$$\begin{aligned} f_g(Y, \mathbf{X}, \Delta) &= \Delta \frac{\tilde{\mathbf{P}}_n\{I(\tilde{Y} \geq Y)[2 + g(\tilde{\mathbf{X}}) - g(\mathbf{X})]_+\}}{\tilde{\mathbf{P}}_n[I(\tilde{Y} \geq Y)]} + \tilde{\mathbf{P}} \left(\frac{\Delta I(Y \geq \tilde{Y})[2 + g(\mathbf{X}) - g(\tilde{\mathbf{X}})]_+}{\tilde{\mathbf{P}}_n[I(\tilde{Y} \geq Y)]} \right) \\ &= \tilde{\mathbf{P}} \left(\frac{\Delta I(Y \geq \tilde{Y}) \mathbf{P}^*\{I(Y^* \geq \tilde{Y})[2 + g(\mathbf{X}^*) - g(\tilde{\mathbf{X}})]_+\}}{\mathbf{P}_n^*\{I(Y^* \geq \tilde{Y})\} \mathbf{P}^*\{I(Y^* \geq \tilde{Y})\}} \right). \end{aligned}$$

Therefore,

$$\sup_{\|g\|_{\mathcal{H}_n} \leq \sqrt{c/\lambda_n}} |\mathcal{P}\mathcal{R}_n(g) - \mathcal{P}\mathcal{R}(g)| \leq \sup_{\|g\|_{\mathcal{H}_n} \leq \sqrt{c/\lambda_n}} |(\mathbf{P}_n - \mathbf{P})f_g| + c/n.$$

On the other hand, from Theorem 3.1 in Steinwart and Scovel (2007), we have

$$\log N(\epsilon, \sqrt{c/\lambda_n} \mathcal{B}_{\mathcal{H}_n}, l_{\infty}) \leq c_{p,d} \sigma_n^{(p/4-1)d} \left(\frac{\epsilon}{\sqrt{c/\lambda_n}} \right)^{-p} \leq c_{p,d} \sigma_n^{(p/4-1)d} \lambda_n^{-p/2} \epsilon^{-p},$$

where $N(\epsilon, \mathcal{F}, l_{\infty})$ is the ϵ -covering number of \mathcal{F} under l_{∞} -norm, d is the dimension of \mathbf{X} , p is any number in $(0, 2)$ and $c_{p,d}$ is a constant only depending on (p, d) . Moreover, we note that by the property of the hinge-loss, f_g is the Lipschitz continuous in g and satisfies

$$|f_{g_1} - f_{g_2}| \leq c|g_1 - g_2|.$$

This implies

$$\log N(\epsilon, \{f_g/a_n : g \in \sqrt{c/\lambda_n} \mathcal{B}_{\mathcal{H}_n}\}, l_{\infty}) \leq c_{p,d} \sigma_n^{(p/4-1)d} \epsilon^{-p},$$

where $a_n = \sqrt{c/\lambda_n} \sigma_n^{-(1-p/4)d/p}$. Therefore, according to Theorem 2.14.10 in van der Vaart and Wellner (1996), we obtain

$$P \left(\sqrt{n} \sup_{\|g\|_{\mathcal{H}_n} \leq \sqrt{c/\lambda_n}} |(\mathbf{P}_n - \mathbf{P})(f_g/a_n)| > x \right) \leq e^{-cx^2}$$

for some constant c only depending on (p, d) . Consequently, (A.2) gives

$$P(\lambda_n \|\hat{g}\|_{\mathcal{H}}^2 + \mathcal{P}\mathcal{R}(\hat{g}) - (\lambda_n \|g_{\lambda_n}\| + \mathcal{P}\mathcal{R}(g_{\lambda_n})) > cn^{-1} + a_n n^{-1/2} x) \leq e^{-cx^2}. \quad (\text{A.3})$$

Hence, we have proved

$$\lambda_n \|\hat{g}\|_{\mathcal{H}_n}^2 + \mathcal{P}\mathcal{R}(\hat{g}) \leq \inf_{g \in \mathcal{H}_n} \{\lambda_n \|g\|_{\mathcal{H}_n} + \mathcal{P}\mathcal{R}(g)\} + O_p \left(\frac{\lambda_n^{-1/2} \sigma_n^{-(1/p-1/4)d}}{\sqrt{n}} \right). \quad (\text{A.3})$$

Let $g^* = \text{argmin} \mathcal{P}\mathcal{R}(g)$. From the expression of $\mathcal{P}\mathcal{R}(g)$, we note

$$|\mathcal{P}\mathcal{R}(g) - \mathcal{P}\mathcal{R}(g^*)| \leq c \|g - g^*\|_{L_1(\mathcal{P})}.$$

Thus, if we define

$$\tilde{g}(\mathbf{x}) = \frac{2\sigma_n^{-d/2}}{\pi^{d/4}} \int e^{-\|x-y\|^2/(2\sigma_n^2)} g^*(y) dy,$$

then $\tilde{g} \in \mathcal{H}_n$ and

$$\|g - g^*\|_{\mathcal{H}_n} \leq \|g - \tilde{g}\|_{L_2(\mathcal{P})} \leq c\sigma_n^{d/2}.$$

Therefore,

$$\inf_{g \in \mathcal{H}_n} \{\lambda_n \|g\|_{\mathcal{H}_n} + \mathcal{P}\mathcal{R}(g)\} \leq \{\lambda_n \|\tilde{g}\|_{\mathcal{H}_n} + \mathcal{P}\mathcal{R}(\tilde{g})\} \leq \mathcal{P}\mathcal{R}(g^*) + c\sigma_n^{d/2} + c\lambda_n,$$

and the result in Theorem 3.2 holds. \blacksquare

References

- S. Abe. *Support Vector Machines for Pattern Classification, Second Edition*. Springer, London, 2010.
- J. E. Barrett and A. C. C. Coolen. Gaussian process regression for survival data with competing risks. *arXiv preprint*, 1312.1591, 2013.
- P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe. Convexity, classification and risk bounds. *The Journal of American Statistical Associations*, 101(473):138–156, 2006.
- S. Bennett. Analysis of survival data by the proportional odds model. *Statistics in Medicine*, 2:273–277, 1983.
- P. J. Bickel, C. A. J. Klaassen, Y. Ritov, J. A. Wellner, et al. *Efficient and Adaptive Estimation for Semiparametric Models*. Springer-Verlag, 1998.
- I. Bou-Hamad, D. Larocque, and H. Ben-Amour. A review of survival trees. *Statistics Surveys*, 5:44–71, 2011.
- J. Buckley and I. James. Linear regression with censored data. *Biometrika*, 66:429–436, 1979.
- K. Chen, Z. Jin, and Z. Ying. Semiparametric analysis of transformation models with censored data. *Biometrika*, 89:659–668, 2002.
- T. Chen, Y. Wang, H. Chen, K. Marder, and D. Zeng. Targeted local support vector machine for age-dependent classification. *Journal of the American Statistical Association*, 109(507):1174–1187, 2014.
- S. C. Cheng, L. J. Wei, and Z. Ying. Analysis of transformation models with censored data. *Biometrika*, 82:835–845, 1995.
- D. R. Cox. Regression models and life-tables (with discussion). *Journal of the Royal Statistical Society: Series B*, 34:187–220, 1972.
- D. M. Dabrowska and K. A. Doksum. Partial likelihood in transformation models with censored data. *Scandinavian Journal of Statistics*, 15:1–23, 1988.
- Y. Goldberg and M. R. Kosorok. Support vector regression for right censored data. *Unpublished manuscript*, 2013.
- T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer, New York, 2009.
- F. M. Khan and V. B. Zubeck. Support vector regression for censored data (SVRC): a novel tool for survival analysis. *In Eighth IEEE International Conference on Data Mining*, pages 863–868, 2008.
- Y. Lee, Y. Lin, and G. Wahba. Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radance data. *Journal of the American Statistical Association*, 99:67–81, 2004.
- J. H. Lubin, D. Couper, P. L. Lutsey, M. Woodward, H. Yatsuya, and R. R. Huxley. Risk of cardiovascular disease from cumulative cigarette use and the impact of smoking intensity. *Epidemiology*, 27(3):395–404, 2016.
- M. E. MacDonald, C. M. Ambrose, M. P. Duryao, R. H. Myers, C. Lin, L. Srinidhi, G. Barnes, S. A. Taylor, M. James, N. Groof, et al. A novel gene containing a trinucleotide repeat that is expanded and unstable on Huntington’s disease chromosomes. *Cell*, 72(6):971–983, 1993.
- K. Marek, D. Jennings, S. Lasch, A. Siderowf, C. Tanner, T. Simuni, C. Coffey, K. Kieburtz, E. Flagg, S. Chowdhury, et al. The parkinson progression marker initiative (PPMI). *Progress in neurobiology*, 95(4):629–635, 2011.
- J. Mugueraza and A. Munoz. Support vector machines with applications. *Statistical Science*, 21(3):322–336, 2006.
- S. G. Mueller, M. W. Weiner, L. J. Thal, R. C. Petersen, C. R. Jack, W. Jagust, J. Q. Trojanowski, A. W. Toga, and L. Beckett. Ways toward an early diagnosis in Alzheimer’s disease: the Alzheimer’s Disease Neuroimaging Initiative (ADNI). *Alzheimer’s & Dementia*, 1(1):55–66, 2005.
- G. Orru, W. Pettersson-Yeo, A.F. Margandi, and et al. Using support vector machine to identify imaging biomarkers of neurological and psychiatric disease: a critical review. *Neurosci Biobehav Rev*, 36(4):1140–1152, 2012.
- J. S. Paulsen. Cognitive impairment in Huntington disease: diagnosis and treatment. *Current neurology and neuroscience reports*, 11(5):474–483, 2011.
- J. S. Paulsen, D. R. Langbehn, J. C. Stout, E. Aylward, C. A. Ross, M. Nance, and et al. Detection of Huntington’s disease decades before diagnosis: the Predict-HD study. *Journal of Neurology, Neurosurgery and Psychiatry*, 79:874–880, 2008a.
- J. S. Paulsen, D. R. Langbehn, J. C. Stout, E. Aylward, C. A. Ross, M. Nance, M. Guttman, S. Johnson, M. MacDonald, L. J. Bejlinger, K. Duff, E. Kayson, K. Biglan, I. Shoulson, D. Oakes, and M. Hayden. Detection of Huntington’s disease decades before diagnosis: the Predict-HD study. *Journal of Neurology, Neurosurgery & Psychiatry*, 79(8):874–880, 2008b.
- J. S. Paulsen, J. D. Long, H. J. Johnson, E. H. Aylward, C. A. Ross, J. K. Williams, M. A. Nance, C. J. Erwin, H. J. Westervelt, D. L. Harrington, et al. Clinical and biomarker changes in premanifest Huntington disease show trial feasibility: a decade of the PREDICT-HD study. *Frontiers in aging neuroscience*, 6:78:1–11, 2014.
- B. D. Ripley and R. M. Ripley. Neural networks as statistical methods in survival analysis. *Clinical Application of Artificial Neural Network*, pages 237–255, 2001.
- R. M. Ripley, A. L. Harris, and L. Tarasenko. Non-linear survival analysis using neural networks. *Statistics in Medicine*, 23(5):825–842, 2004.

- J. M. Robins, A. Rotnitzky, and L. P. Zhao. Analysis of semiparametric regression models for repeated outcomes in the presence of missing data. *Journal of the American Statistical Association*, 90(429):106–121, 1995.
- P. K. Shivaswamy, W. Chu, and M. Jansche. A support vector approach to censored targets. *In Seventh IEEE International Conference on Data Mining*, pages 655–660, 2007.
- A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Journal of Statistics and Computing*, 14:199–222, 2004.
- I. Steinwart. Support vector machines are universally consistent. *Journal of Complexity*, 18:768–791, 2002.
- I. Steinwart and A. Christmann. *Support Vector Machines, First Edition*. Springer, New York, 2008.
- I. Steinwart and C. Scovel. Fast rates for support vector machines using Gaussian kernels. *The Annals of Statistics*, 35(2):575–607, 2007.
- The ARIC investigators. The Atherosclerosis Risk in Communities (ARIC) study: design and objectives. *American Journal of Epidemiology*, 129(4):687–702, 1989.
- V. Van Belle, K. Pelckmans, J. A. K. Suykens, and S. Van Huffel. Additive survival least-squares support vector machines. *Statistics in Medicine*, 29(2):296–308, 2010.
- V. Van Belle, K. Pelckmans, S. Van Huffel, and J. A. K. Suykens. Support vector methods for survival analysis: a comparison between ranking and regression approaches. *Artificial Intelligence in Medicine*, 53(2):107–118, 2011.
- A. W. van der Vaart and J. A. Wellner. *Weak Convergence and Empirical Processes*. Springer, New York, 1996.
- H. Wang, X. Shen, and W. Pan. Large margin hierarchical classification with mutually exclusive class membership. *The Journal of Machine Learning Research*, 12:2721–2748, 2011.
- D. Zeng and D. Y. Lin. Efficient estimation of semiparametric transformation models for counting processes. *Biometrika*, 93(3):627–640, 2006.
- D. Zeng and D. Y. Lin. Maximum likelihood estimation in semiparametric models with censored data (with discussion). *Journal of the Royal Statistical Society, B*, 69(4):507–564, 2007.
- Y. Zhang, J. D. Long, J. A. Mills, J. H. Warner, W. Lu, J. S. Paulsen, and et al. Indexing disease progression at study entry with individuals at-risk for Huntington disease. *American Journal of Medical Genetics Part B: Neuropsychiatric Genetics*, 156B(7):751–763, 2011.

Stable Graphical Models

Navodit Misra

*Max Planck Institute for Molecular Genetics
Innsstr. 63-73, 14195 Berlin, Germany*

NAVODITMISRA@GMAIL.COM

Ercan E. Kuruoglu

*ISTI-CNR
Via G. Moruzzi 1, 56124 Pisa, Italy*
and

ERCAN.KURUOGLU@ISTI-CNR.IT

*Max Planck Institute for Molecular Genetics
Innsstr. 63-73, 14195 Berlin, Germany*

Editor: Jeff Bilmes

Abstract

Stable random variables are motivated by the central limit theorem for densities with (potentially) unbounded variance and can be thought of as natural generalizations of the Gaussian distribution to skewed and heavy-tailed phenomenon. In this paper, we introduce α -stable graphical (α -SG) models, a class of multivariate stable densities that can also be represented as Bayesian networks whose edges encode linear dependencies between random variables. One major hurdle to the extensive use of stable distributions is the lack of a closed-form analytical expression for their densities. This makes penalized maximum-likelihood based learning computationally demanding. We establish theoretically that the *Bayesian information criterion* (BIC) can asymptotically be reduced to the computationally more tractable *minimum dispersion criterion* (MDC) and develop **Stable**, a structure learning algorithm based on MDC. We use simulated datasets for five benchmark network topologies to empirically demonstrate how **Stable** improves upon ordinary least squares (OLS) regression. We also apply **Stable** to microarray gene expression data for lymphoblastoid cells from 727 individuals belonging to eight global population groups. We establish that **Stable** improves test set performance relative to OLS via ten-fold cross-validation. Finally, we develop **SGEX**, a method for quantifying differential expression of genes between different population groups.

Keywords: Bayesian networks, stable distributions, linear regression, structure learning, gene expression, differential expression

1. Introduction

Stable distributions have found applications in modeling several real-life phenomena (Berger and Mandelbrot, 1963; Mandelbrot, 1963; Nikias and Shao, 1995; Gallardo et al., 2000; Achim et al., 2001) and have robust theoretical justification in the form of the generalized central limit theorem (Feller, 1968; Nikias and Shao, 1995; Nolan, 2013). Several special instances of multivariate generalization of stable distributions have also been described in literature (Samorodnitsky and Taqqu, 1994; Nolan and Rajput, 1995). Multivariate stable densities have previously been applied to modeling wavelet coefficients with bivariate α -

stable distributions (Achim and Kuruoglu, 2005), inferring parameters for linear models of network flows (Bickson and Guestrin, 2011) and stock market fluctuations (Bonato, 2012).

In this paper, we describe α -stable graphical (α -SG) models, a new class of multivariate stable densities that can be represented as directed acyclic graphs (DAG) with arbitrary network topologies. We prove that these multivariate densities also correspond to linear regression-based Bayesian networks and establish a model selection criterion that is asymptotically equivalent to the *Bayesian information criterion* (BIC). Using simulated data for five benchmark network topologies, we empirically show how α -SG models improve structure and parameter learning performance for linear regression networks with additive heavy-tailed noise.

One motivation for the present work comes from potential applications to computational biology, especially in genomics, where Bayesian network models of gene expression profiles are a popular tool (Friedman et al., 2000; Ben-Dor et al., 2000; Friedman, 2004). A common approach to network models of gene expression involves learning linear regression-based Gaussian graphical models. However, the distribution of experimental microarray intensities shows a clear skew and may not necessarily be best described by a Gaussian density (Section 3.2). Another aspect of microarray intensities is that they represent the average mRNA concentration in a population of cells. Assuming the number of mRNA transcripts within each cell to be independent and identically distributed, the generalized central limit theorem suggests that the observed shape should asymptotically (for large population size) approach a stable density (Feller, 1968; Nikias and Shao, 1995; Nolan, 2013). Univariate stable distributions have previously been used to model gene expression data (Salas-Gonzalez et al., 2009a,b) and it is therefore natural to consider multivariate α -stable densities as models for mRNA expression for larger sets of genes. In Section 3.2 we provide empirical evidence to support this reasoning. We further develop α -stable graphical (α -SG) models for quantifying differential expression of genes from microarray data belonging to phase III of the HapMap project (International HapMap 3 Consortium and others, 2010; Montgomery et al., 2010; Stranger et al., 2012).

The rest of the paper is structured as follows : Section 2.1 describes the basic notation and background concepts for Bayesian networks and stable densities. Section 2.2 introduces α -SG models and establishes that these models are Bayesian networks that also represent multivariate stable distributions with discrete spectral measures. Section 2.3 establishes the equivalence of the popular but (in this case) computationally challenging *Bayesian information criterion* (BIC) for structure learning and the computationally more tractable *minimum dispersion criterion* (MDC), for all α -SG models that represent symmetric densities. Furthermore, we establish how data samples from any α -SG model can be combined to generate samples from a partner symmetric α -SG model with identical network topology and regression coefficients. Using these theoretical results we design **Stable**, an efficient algorithm that combines ordering-based search (OBS) (Teyssier and Koller, 2005) for structure learning with the iteratively re-weighted least squares (IRLS) algorithm (Byrd and Payne, 1979) for learning the regression parameters via least l_p norm estimation. Finally, in Section 3 we implement the structure and parameter learning algorithm on simulated and expression microarray data sets.

2. Methods

In this section we develop the theory and algorithms for learning α -SG models from data. First, we discuss some well-established results for Bayesian networks and α -stable densities.

2.1 Background

We begin with an introduction to Bayesian network models (Pearl, 1988) for the joint probability distribution of a finite set of random variables $\mathcal{X} = \{X_1, \dots, X_N\}$. A Bayesian network $B(G, \Theta)$ is specified by a directed acyclic graph (DAG) G , whose vertices represent random variables in \mathcal{X} and a set of parameters $\Theta = \{\theta_i | X_i \in \mathcal{X}\}$, that determine the conditional probability distribution $p(X_i | Pa(X_i), \theta_i)$ for each variable $X_i \in \mathcal{X}$ given the state of its parents $Pa(X_i) \subseteq \mathcal{X} \setminus \{X_i\}$ in G (Koller and Friedman, 2009). We will overload the symbols X_i and $Pa(X_i)$ to represent both sets of random variables and their realizations. The directed acyclic graph G implies a factorization of the joint probability density into terms representing each variable X_i and its parents $Pa(X_i)$ (called a *family*) such that :

$$P_B(\mathcal{X}) = \prod_{i=1}^{|\mathcal{X}|} p(X_i | Pa(X_i), \theta_i) \quad (1)$$

The dependence of $p(X_i | Pa(X_i), \theta_i)$ on θ_i is usually specified by an appropriately chosen family of parameterized probability densities for the random variables, such as Gaussian or log-Normal. In this paper, we will use multivariate stable densities to model the random variables in \mathcal{X} . The primary motivation for modeling continuous random variables using stable distributions comes from the generalization of the central limit theorem to distributions with unbounded variance (Feller, 1968; Nikias and Shao, 1995). Stable distributions are parameterized to allow varying degrees of impulsiveness and skewness. The generalized central limit theorem requires that the sums of stable random variables are stable and more generally in the limit of large N , all sums of N independent, identically distributed random variables approach a stable density. A formal definition for stable random variables can be provided in terms of the characteristic function (Fourier transform of the density function)

Definition 1 A stable random variable $X \sim S_{\alpha}(\beta, \gamma, \mu)$, is defined for each $\alpha \in (0, 2]$, $\beta \in [-1, 1]$, $\gamma \in (0, \infty)$ and $\mu \in (-\infty, \infty)$. The probability density $f(X | \alpha, \beta, \gamma, \mu)$ is implicitly specified by a characteristic function $\phi(q | \alpha, \beta, \gamma, \mu)$:

$$\begin{aligned} \phi(q | \alpha, \beta, \gamma, \mu) &\equiv \mathbb{E}[\exp(iqX)] \\ &= \int_{-\infty}^{\infty} f(X | \alpha, \beta, \gamma, \mu) \exp(iqX) dX \\ &= \exp(i\mu q - \gamma |q|^\alpha [1 - i\beta \operatorname{sign}(q)^\gamma r(q, \alpha)]) \end{aligned}$$

where, $r(q, \alpha) = \begin{cases} \tan \frac{\alpha\pi}{2} & \alpha \neq 1 \\ -\frac{2}{\pi} \log |q| & \alpha = 1 \end{cases}$

The parameters α, β, γ and μ will be called the characteristic exponent, skew, dispersion and location respectively. Unfortunately, the density $f(X | \alpha, \beta, \gamma, \mu)$ does not have a closed-form analytical expression except for the three well-known stable distributions (Figure 1 and Table 1).

Distribution	$S_{\alpha}(\beta, \gamma, \mu)$	$f(X \alpha, \beta, \gamma, \mu)$	Support
Lévy(γ, μ)	$S_{0.5}(1, \gamma, \mu)$	$\frac{\sqrt{\gamma}}{\sqrt{2\pi}} \frac{1}{(x-\mu)^{3/2}} \exp\left(-\frac{\gamma^2}{2(x-\mu)}\right)$	$\mu < x < \infty$
Cauchy(γ, μ)	$S_{1.0}(0, \gamma, \mu)$	$\frac{1}{\pi} \frac{\gamma}{\gamma^2 + (x-\mu)^2}$	$-\infty < x < \infty$
Normal(μ, σ)	$S_{2.0}(0, \gamma = \frac{\sigma^2}{2}, \mu)$	$\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$	$-\infty < x < \infty$

Table 1: Closed-form analytical expressions for Lévy, Cauchy and Normal densities and the corresponding α -stable parameters.

Except for the Gaussian case, the asymptotic (large x) behavior of univariate α -stable densities shows Pareto or power law tails (Lévy, 1925). The following lemma formalizes this observation (Samorodnitsky and Taqqu, 1994; Nolan, 2013)

Lemma 1 $Hf X \sim S_{\alpha}(\beta, \gamma, 0)$ with $0 < \alpha < 2$, then as $x \rightarrow \infty$

$$Pr(X > x) \sim (1 + \beta)\gamma C_{\alpha} x^{-\alpha}$$

$$C_{\alpha} = (2 \int_0^{\infty} x^{-\alpha} \sin x dx)^{-1} = \frac{1}{\pi} \Gamma(\alpha) \sin\left(\frac{\alpha\pi}{2}\right)$$

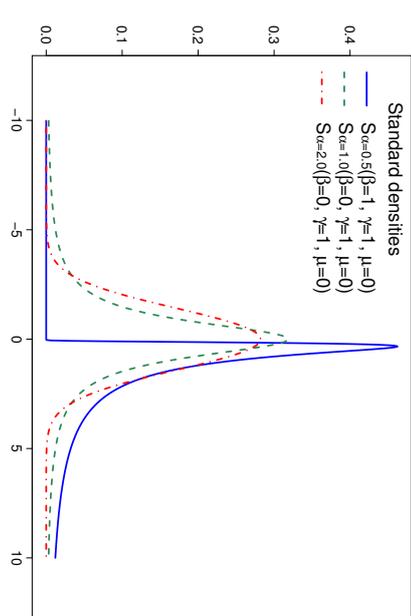


Figure 1: The three instances of analytically known univariate α -stable densities $S_{\alpha}(\beta, \gamma, \mu)$. Lévy(γ, μ) $\sim S_{0.5}(1, \gamma, \mu)$ (solid blue curves), Cauchy(γ, μ) $\sim S_{1.0}(0, \gamma, \mu)$ (dashed green curves) and Normal(μ, σ) $\sim S_{2.0}(0, \frac{\sigma^2}{2}, \mu)$ (dot-dashed red curves).

It is straight forward to use the characterization of stable random variables in Definition 1 to verify the following well-known properties (Samorodnitsky and Taqqu, 1994),

Property 1 If $X_1 \sim S_\alpha(\beta_1, \gamma_1, \mu_1)$ and $X_2 \sim S_\alpha(\beta_2, \gamma_2, \mu_2)$ are independent stable random variables, then $Y = X_1 + X_2 \sim S_\alpha(\beta, \gamma, \mu)$, with

$$\beta = \frac{\beta_1 \gamma_1 + \beta_2 \gamma_2}{\gamma_1 + \gamma_2}, \quad \gamma = (\gamma_1 + \gamma_2), \quad \mu = \mu_1 + \mu_2$$

Property 2 If $X \sim S_\alpha(\beta, \gamma, \mu)$ and $c, d \in \mathbb{R}$, then

$$cX + d \sim \begin{cases} S_\alpha(\text{sign}(c)\beta, |c|^\alpha \gamma, c\mu + d), & \alpha \neq 1 \\ S_\alpha(\text{sign}(c)\beta, |c|\gamma, c(\mu - \frac{2\gamma \beta \ln|c|}{\pi}) + d), & \alpha = 1 \end{cases}$$

A word on the notation used throughout this paper. We will use the symbol $\|Y\|_p = (\sum_\lambda |Y_\lambda|^p)^{1/p}$ to represent the l_p norm of a vector. The l_p norm of a vector representing N realizations of a random variable Z is related to the p^{th} moment $E(|Z|^p) = \|Z\|_p^p/N$. For heavy-tailed α -stable densities, one convenient method for parameter estimation is via fractional lower order moments (FLOM) for $p < \alpha$ (Hardin Jr, 1984; Nikias and Shao, 1995). Later, we will discuss FLOM-based parameter learning in greater detail (Section 2.4.1).

2.2 α -Stable Graphical Models

We can now introduce Bayesian network models reconstructed from stable densities that have compact representations for the characteristic function. Univariate α -stable densities can be generalized to represent multivariate stable distributions that are defined as follows (Samorodnitsky and Taqqu, 1994),

Definition 2 A d -dimensional multivariate stable distribution over $\mathcal{X} = \{X_1, \dots, X_d\}$ is defined by an $\alpha \in (0, 2]$, $\mu \in \mathbb{R}^d$ and a spectral measure Λ over the d -dimensional unit sphere S_d , such that the characteristic function

$$\begin{aligned} \Phi(q|\alpha, \mu, \Lambda) &\equiv \mathbb{E}[\exp(iq^T \mathcal{X})] \\ &= \exp\left(-\int_{S_d} \psi(s^T q|\alpha)\Lambda(ds) + i\mu^T q\right) \end{aligned}$$

where, $\psi(u|\alpha) = |u|^\alpha (1 - i \text{sign}(u)r(u, \alpha))$

Definition 3 An α -stable graphical (α -SG) model $B(G, \Theta)$ is a probability distribution over \mathcal{X} such that

1. $Z_j \equiv X_j - \sum_{X_k \in Pa(X_j)} w_{jk} X_k \sim S_\alpha(\beta_j, \gamma_j, \mu_j)$
2. Z_j is independent of Z_k , if $j \neq k$, $\forall X_j \in \mathcal{X}$

where $Pa(X_i) \subseteq \mathcal{X} \setminus \{X_i\}$ are the parent nodes of X_j in the directed acyclic graph G and Θ describes the distribution parameters

$$\begin{aligned} w_{jk} &\in \mathbb{R}, & W_j &= \{w_{jk} | X_k \in Pa(X_j)\}, \\ \theta_j &= \{\alpha, \beta_j, \gamma_j, \mu_j\} \cup W_j, & \Theta &= \{\theta_i | X_i \in \mathcal{X}\}. \end{aligned}$$

A symmetric α -stable graphical (So-SG) model is a α -SG model with $\beta = 0$.

It is straightforward to see that $B(G, \Theta)$ is indeed a Bayesian network. Note also that the fact that Z_j are stable follows directly from Property 1.

Lemma 2 $B(G, \Theta)$ in Definition 3 represents a Bayesian network

Proof Let $d = |\mathcal{X}|$. First note that every directed acyclic graph can be used to infer an ordering (not necessarily unique) on the variables in \mathcal{X} such that all parents of each variable have a lower order than the variable itself. Suppose we index each variable with its order in an ordering compatible with the DAG, such that X_i has order i . The proof rests on the fact that the transformation matrix from $\{Z_i\}$ to $\{X_i\}$ for such a graph is lower triangular, with each diagonal entry equal to 1. Since the determinant of a triangular matrix equals the product of its diagonal entries, the Jacobian for the transformation (or the determinant of the transformation matrix), $|\frac{\partial(Z_1, \dots, Z_d)}{\partial(X_1, \dots, X_d)}| = 1$. Furthermore, since the noise variables Z_j are independent of each other

$$\begin{aligned} P_B(Z_1, \dots, Z_d) &= \prod_{j=1}^d f(Z_j | \alpha, \beta_j, \gamma_j, \mu_j) \\ \text{also, } p(X_j | Pa(X_j), \theta_j) &= f(Z_j | \alpha, \beta_j, \gamma_j, \mu_j) \\ &\implies P_B(\mathcal{X}) = P_B(Z_1, \dots, Z_d) \frac{\partial(Z_1, \dots, Z_d)}{\partial(X_1, \dots, X_d)} \\ &\implies P_B(\mathcal{X}) = \prod_{j=1}^d p(X_j | Pa(X_j), \theta_j) \frac{\partial(Z_1, \dots, Z_d)}{\partial(X_1, \dots, X_d)} \\ &\implies P_B(\mathcal{X}) = \prod_{j=1}^d p(X_j | Pa(X_j), \theta_j) \end{aligned}$$

Hence, $B(G, \Theta)$ is a Bayesian network. ■

Before establishing the fact that an α -SG model is a multivariate stable density in the sense of Definition 2, we prove the following result (proof is provided in Appendix A) :

Lemma 3 Every d -dimensional distribution with a characteristic function of the form

$$\Phi(q|\alpha, \tilde{\mu}, \Lambda) = \prod_{k=1}^d \phi(c_k^T q | \alpha, \beta_k, \gamma_k, \mu_k) \quad \text{where, } c_k, q \in \mathbb{R}^d$$

represents a multivariate stable distribution with a discrete spectral measure Λ .

We are now in a position to establish that α -SG models imply a multivariate stable density with a spectral measure concentrated on a finite number of points over the unit sphere.

Lemma 4 Every α -SG model represents a multivariate stable distribution with a discrete spectral measure of the form in Lemma 3.

Proof We will prove the lemma by induction. First, observe that every Bayesian network can be used to assign an ordering (not unique) such that $Pa(X_j) \subseteq \{X_1, \dots, X_j - 1\}$. As before, we will use such an ordering to index each random variable in \mathcal{X} , such that $X_{|\mathcal{X}|}$ has no descendants. The base case of the lemma, where $|\mathcal{X}| = 1$ is clearly true. Assume that the lemma is true for all Bayesian networks with $|\mathcal{X}| = m - 1$. Then for any Bayesian network B with $|\mathcal{X}| = m$ random variables

$$\begin{aligned} \Phi_B(q) &\equiv \mathbb{E}[\exp(iq^T \mathcal{X})] \\ &= \int \prod_{j=1}^{|\mathcal{X}|} dX_j f(Z_j | \alpha, \beta_j, \gamma_j, \mu_j) \exp(iq_j X_j) \\ &= \int \left[\prod_{j=1}^{m-1} dX_j f(Z_j | \alpha, \beta_j, \gamma_j, \mu_j) \exp(iq_j X_j) \right] \int dX_m f(Z_m | \alpha, \beta_m, \gamma_m, \mu_m) \exp(iq_m X_m) \\ &= \int \left[\prod_{j=1}^{m-1} dX_j f(Z_j | \alpha, \beta_j, \gamma_j, \mu_j) \exp(iq_j X_j) \right] \int dZ_m f(Z_m | \alpha, \beta_m, \gamma_m, \mu_m) \exp(iq_m Z_m) \\ &= \Phi_{\tilde{B}}(\tilde{q}) \phi(q_m | \alpha, \beta_m, \gamma_m, \mu_m) \\ &\quad \text{where } \tilde{B} \text{ is the Bayes net on } \tilde{\mathcal{X}} = \mathcal{X} \setminus \{X_m\}, \\ &\quad \text{and } \tilde{q}_j = q_j + u_{mj} q_m | Pa(X_m) \cap \{X_j\} \vee X_j \in \tilde{\mathcal{X}} \end{aligned}$$

Since by assumption,

$$\begin{aligned} \Phi_{\tilde{B}}(\tilde{q}) &= \prod_{k=1}^{m-1} \phi(s_k^T \tilde{q} | \alpha, \beta_k, \gamma_k, \mu_k) \\ \implies \Phi_B(q) &= \phi_{\tilde{B}}(\tilde{q}) \phi(q_m | \alpha, \beta_m, \gamma_m, \mu_m) \\ &= \prod_{k=1}^m \phi(s_k^T q | \alpha, \beta_k, \gamma_k, \mu_k), \text{ where :} \\ s_k^T q &= \begin{cases} \sum_{j=1}^{m-1} s_{k,j} q_j + u_{mj} q_m | Pa(X_m) \cap \{X_j\} & k < m \\ q_m & k = m \end{cases} \end{aligned}$$

Therefore, $\Phi_B(q)$ represents a m -dimensional multivariate stable distribution with a discrete spectral measure (Lemma 3). Therefore, by induction, every α -SG model represents a multivariate stable distribution with a discrete spectral measure of the form in Lemma 3. ■

2.3 Learning α -SG Models

A popular method for structure learning in Bayesian network models is based on the *Bayesian information criterion* (BIC) which is also equivalent to the minimum description length (MDL) principle (Schwarz, 1978; Heckerman et al., 2000).

Definition 4 Given a data set $D = \{D_1, \dots, D_N\}$, the *Bayesian Information Score* $S_{BIC}(B|D)$ for a Bayesian network $B(G; \Theta)$ is defined as,

$$S_{BIC}(B|D) = \sum_{D_j \in D} \log [P_B(D_j)] - \sum_{X_i \in \mathcal{X}} \frac{|Pa(X_i)|}{2} \log N$$

The *Bayesian information criterion* (BIC) selects the Bayesian network that maximizes this score over the space of all directed acyclic graphs G and parameters Θ .

The major stumbling block in using stable densities is due to the fact that there is no known closed-form analytical expression for them (apart from special cases representing Gaussian, Cauchy and Levy distributions). This makes BIC based inference computationally demanding due to the marginal likelihood term $P_B[D_\lambda]$. One main contribution of this paper is an efficient method of learning the network structure and parameters for α -SG models. The next lemma establishes a new result that is useful in efficiently solving the learning problem.

Lemma 5 Given a data set $D_Y = \{Y_1, \dots, Y_N\}$ generated from a stable random variable $Y \sim S_\alpha(\beta, \gamma, \mu)$

$$\begin{aligned} \sum_{j=1}^N \log [f(Y_j | \alpha, \beta, \gamma, \mu)] &= -N \left(\log \gamma + h(Y | \alpha, \beta) \right) \\ \text{where, } \lim_{N \rightarrow \infty} h(Y | \alpha, \beta) &= - \int dY f(Y | \alpha, \beta, 1, 0) \log f(Y | \alpha, \beta, 1, 0) \\ &= H[S_\alpha(\beta, 1, 0)] \end{aligned}$$

where, $H[\cdot]$ is the entropy of the corresponding random variable.

Proof Since Y includes samples from a stable distribution, $Y \sim S_\alpha(\beta, \gamma, \mu)$ by definition, performing a change of variable to

$$\begin{aligned} Y \rightarrow \tilde{Y} &= \frac{Y}{\gamma^{1/\alpha}} - \tilde{\mu} \\ \text{where, } \tilde{\mu} &= \begin{cases} \frac{\mu}{\gamma^{1/\alpha}} & \alpha \neq 1 \\ \frac{\mu}{\gamma} + \frac{2\beta \ln 2}{\pi} & \alpha = 1 \end{cases} \end{aligned} \quad (2)$$

we get, the *standard* form density $\tilde{Y} \sim S_\alpha(\beta, 1, 0)$ using Property 2. Furthermore, samples from the transformed data set $\tilde{Y} = \{\tilde{Y}_1, \dots, \tilde{Y}_N\}$ are also distributed according to the following standard density :

$$f(Y | \alpha, \beta, \gamma, \mu) = f(\tilde{Y} | \alpha, \beta, 1, 0) \frac{d\tilde{Y}}{dY} = f(\tilde{Y} | \alpha, \beta, 1, 0) \frac{1}{\gamma^{1/\alpha}}$$

This implies that if we know the parameters α, β, γ and μ for the density generating D_Y

$$\begin{aligned} \log [f(Y|\alpha, \beta, \gamma, \mu)] &= \sum_{j=1}^N \log f(Y_j|\alpha, \beta, \gamma, \mu) \\ &= \sum_{j=1}^N \left\{ -\frac{\log \gamma}{\alpha} + \log f(\tilde{Y}_j|\alpha, \beta, 1, 0) \right\} \\ &= -N \left(\frac{\log \gamma}{\alpha} + h(Y|\alpha, \beta) \right) \end{aligned}$$

where, $h(Y|\alpha, \beta)$ is defined by

$$h(Y|\alpha, \beta) \equiv -\frac{1}{N} \sum_{j=1}^N \log f(\tilde{Y}_j|\alpha, \beta, 1, 0) \quad (3)$$

Here \tilde{Y}_j and Y_j are related via Equation 2 for all $1 \leq j \leq N$. Note that since the transformed variables \tilde{Y}_j are samples from $f(\tilde{Y}|\alpha, \beta, 1, 0)$, we have the following asymptotic result for large N

$$\begin{aligned} \lim_{N \rightarrow \infty} h(Y, \alpha, \beta) &= -\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{j=1}^N \log f(\tilde{Y}_j|\alpha, \beta, 1, 0) \\ &= -\int_{-\infty}^{\infty} f(\tilde{Y}|\alpha, \beta, 1, 0) \log f(\tilde{Y}|\alpha, \beta, 1, 0) dY \\ &= H[S_\alpha(\beta, 1, 0)] \end{aligned}$$

where, $H[\cdot]$ is the entropy of the corresponding random variable. ■

As things stand, the entropy $H[\cdot]$ of stable random variables in the standard form is just as difficult to compute as the original log-likelihood and the previous lemma has just transformed one intractable quantity into another. However, there is an important class of models where we can ignore the entropy term during structure learning; this class of multivariate distributions have a special property that every linear combination of random variables is distributed as a stable distribution $S_\alpha(\beta, \cdot, \cdot)$ with the same α and β . One scenario when this is true is when the noise term is symmetric *i.e.* $\beta_i = 0 \forall X_i \in \mathcal{X}$. This special case is important since we later show (Lemma 8) that every α -SG model can be easily transformed into a partner symmetric α -SG model with identical network topology and regression coefficients. For all practical purposes, learning the structure of symmetric α -SG models is effectively the same as learning structure of arbitrary α -SG models.

Lemma 6 Given a symmetric α -stable graphical model for variables in \mathcal{X} ,

$$Z \equiv w^T \mathcal{X} = \sum_{X_j \in \mathcal{X}} w_j X_j \sim S(\alpha, \beta(w) = 0, \gamma(w), \mu(w)), \quad \forall w \in \mathbb{R}^{|\mathcal{X}|}$$

if, $\beta_i = 0, \forall X_i \in \mathcal{X}$

Proof The dispersion $\gamma(w)$ and skewness $\beta(w)$ for the projection $w^T \mathcal{X}$ of any d -dimensional stable random density are given by (Samorodnitsky and Taqqu, 1994)

$$\begin{aligned} \gamma(w) &= \int_{S_d} |w^T s|^\alpha \Lambda(ds) \\ \beta(w) &= \gamma(w)^{-1} \int_{S_d} \text{sign}(w^T s) |w^T s|^\alpha \Lambda(ds) \end{aligned}$$

Since, \mathcal{X} represents a symmetric α -stable graphical model, Lemma 4 and Lemma 3 imply (substituting the characteristic function in the expression for $\beta(w)$ with the expansion in Lemma 3 :

$$\begin{aligned} \beta(w) &= \sum_{k=1}^d \frac{|w^T c_k|^{\frac{\alpha}{2} \gamma_k}}{2^\gamma(w)} \int_{S_d} \left\{ \delta(s - \frac{c_k}{|c_k|_2}) + \delta(s + \frac{c_k}{|c_k|_2}) \right\} |w^T s|^\alpha \text{sign}(w^T s) ds \\ &= 0 \end{aligned}$$

For a recent reference on multiple regression with stable errors, see also Nolan (2013b).

We are now in a position to present the main contribution of this paper : an alternative criterion for model selection that is both computationally efficient and comes with robust theoretical guarantees (Lemma 7). The criterion is called *minimum dispersion criterion (MDC)* and is a penalized version of a technique previously used in signal processing literature for designing filters for heavy-tailed noise (Stuck, 1978).

Definition 5 Given a data set $D = \{D_1, \dots, D_N\}$, the penalized dispersion score $S_{MDC}(B|D)$ for a Bayesian network $B(G, \Theta)$ is defined as,

$$S_{MDC}(B|D) = - \sum_{X_i \in \mathcal{X}} \left\{ N \frac{\log \gamma_i}{\alpha} + \frac{|Pa(X_i)|}{2} \log N \right\}$$

The minimum dispersion criterion (MDC) selects the Bayesian network that maximizes this score over the space of all directed acyclic graphs G and parameters Θ .

Lemma 7 Given a data set $D = \{D_1, \dots, D_N\}$ generated by a symmetric α -stable graphical model, $B^*(G^*, \Theta^*)$, the minimum dispersion criterion is asymptotically equivalent to the Bayesian information criterion over the search space of all symmetric α -stable graphical models

Proof First consider the contribution to BIC score from each family (ie., each random variable and its parents) separately. Let $Z_j = X_j - \sum_{X_k \in Pa(X_j)} w_{jk} X_k$ be any arbitrary set of regression coefficients for a candidate network $B(G, \Theta)$. Note that the coefficients $W_j = \{w_{jk} | X_k \in Pa(X_j)\}$ need not be the true regression coefficients W_j^* and B need not be the true network B^* . We will use the notation $Z_{i,\lambda}$ for the realization of Z_i in sample $D_\lambda \in D$. Since D includes samples from a symmetric α -stable graphical model, Lemma 6

implies $Z_j \sim S_\alpha(\beta = 0, \gamma_j, \mu_j)$. Therefore, using Lemma 5

$$\begin{aligned} Fam(X_j, Pa(X_j)|D) &\equiv \sum_{\lambda=1}^N \log \left[f(Z_{j,\lambda}|\alpha, \beta = 0, \gamma_j, \mu_j) \right] - \frac{|Pa(X_j)|}{2} \log N \\ &= -N \left(\frac{\log \gamma_j}{\alpha} + h(\tilde{Z}_j|\alpha, \beta = 0) \right) - \frac{|Pa(X_j)|}{2} \log N \end{aligned}$$

where, as in Equation 3, Z_j and \tilde{Z}_j are related by the transformation in Equation 2 and $Fam(X_j, Pa(X_j)|D)$ represents the contribution to BIC score from each family (ie., each random variable and its parents).

$$\begin{aligned} \implies \frac{SBIC(B|D)}{N} &= \sum_{X_j \in \mathcal{X}} \frac{Fam(X_j, Pa(X_j)|D)}{N} \\ &= - \sum_{X_j \in \mathcal{X}} \left(\frac{\log \gamma_j}{\alpha} + h(Z_j|\alpha, \beta = 0) + \frac{|Pa(X_j)|}{2N} \log N \right) \\ \implies \lim_{N \rightarrow \infty} \frac{SBIC(B|D)}{N} &= \lim_{N \rightarrow \infty} \frac{SMDC(B|D)}{N} - |\mathcal{X}|H[S_\alpha(\beta = 0, 1, 0)] \end{aligned}$$

Since, $|\mathcal{X}|H[S_\alpha(\beta = 0, 1, 0)]$ is independent of the candidate network structure and regression parameters $\{W_j|X_j \in \mathcal{X}\}$, we get the result that for any pair of networks B and B'

$$\implies \lim_{N \rightarrow \infty} \frac{1}{N} (SBIC(B|D) - SBIC(B'|D)) = \lim_{N \rightarrow \infty} \frac{1}{N} (SMDC(B|D) - SMDC(B'|D))$$

Therefore, asymptotically, BIC is equivalent to MDC when data is generated by a symmetric α -SG graphical model. ■

We now show how samples from any stable graphical model can be combined to yield samples from a partner symmetric stable graphical model with identical parameters and network topology. This transformation was earlier used by Kurugolu (2001) in order to estimate parameters from skewed univariate stable densities. We should point out that the procedure described above has the drawback that symmetrized data set has half the sample size.

Lemma 8 *Every α -SG model can be associated with a symmetric α -SG model with identical skeleton (graph structure) and regression parameters.*

Proof Given a data set $D = \{D_1, \dots, D_N\}$ representing any α -SG model $B(G, \Theta)$, consider a resampled data set $\tilde{D} = \{\tilde{D}_1, \dots, \tilde{D}_{N_s}\}$ with variable realizations

$$\widehat{X}_{i,\lambda} = X_{i,2\lambda} - X_{i,2\lambda-1}, \quad \forall \lambda \in \{1, \dots, N_s = \lfloor N/2 \rfloor\}$$

These 'bootstrapped' data samples $\tilde{D}_\lambda = \{\widehat{X}_{i,\lambda}|X_i \in \mathcal{X}\}$ represent independent realizations of random variables $\widehat{\mathcal{X}} \equiv \{\widehat{X}_i|X_i \in \mathcal{X}\}$. Similarly, we may use the regression parameters W to define resampled noise variables :

$$\tilde{Z}_j \equiv \widehat{X}_j - \sum_{\widehat{X}_k \in Pa(\widehat{X}_j)} w_{jk} \widehat{X}_k$$

We now make two observations :

1. If $Z_j = X_j - \sum_{X_k \in Pa(X_j)} w_{jk} X_k \sim S_\alpha(\beta_j, \gamma_j, \mu_j)$, then using Property 1

$$\tilde{Z}_j \equiv \widehat{X}_j - \sum_{\widehat{X}_k \in Pa(\widehat{X}_j)} w_{jk} \widehat{X}_k \sim S_\alpha(\beta = 0, 2\gamma_j, 0)$$

2. The transformed noise variables \tilde{Z}_j are independent of each other.

But these conditions define an α -SG model (Definition 3). Therefore, by Lemma 2, the resampled data is distributed according to a Bayesian network $B(G, \Theta)$ such that

$$\begin{aligned} \tilde{Z}_j &\equiv \widehat{X}_j - \sum_{\widehat{X}_k \in Pa(\widehat{X}_j)} w_{jk} \widehat{X}_k \\ P_{\tilde{B}}(\widehat{\mathcal{X}}) &= \prod_{j=1}^{|\mathcal{X}|} f(\tilde{Z}_j|\alpha, 0, 2\gamma_j, 0) \\ \tilde{\theta}_j &= \{\alpha, \beta = 0, 2\gamma_j, 0\} \cup W_j, \quad \widehat{\Theta} = \{\tilde{\theta}_j|X_j \in \mathcal{X}\} \end{aligned}$$

■

The expression for MDC in Definition 5 does not involve the stable pdf and hence one may wonder how the variables of the distribution could be estimated. The answer is given by the following property of stable distributions Kurugolu (2001).

Lemma 9 *If $Z \sim S_\alpha(0, \gamma, 0)$, then*

$$E(|Z|^p) = C(p, \alpha) \gamma^{p/\alpha} \quad \forall -1 < p < \alpha$$

where,

$$C(p, \alpha) = \frac{\Gamma(1 - \frac{p}{\alpha})}{\Gamma(1 - p) \cos(p \frac{\pi}{\alpha})}.$$

2.4 The Stable Algorithm

In this section we describe **Stable**, an algorithm for learning the structure and parameters of α -SG models (Algorithm 1). The first step of **Stable** is to center and symmetrize the entire data matrix D_j in terms of the variables \mathcal{X} , as described in Lemma 8. This is followed by estimating the global parameter α using the method of log statistics (Kurugolu, 2001). Finally, structure learning is performed by a modified version of the *ordering-based search* (OBS) algorithm (Section 2.4.2). The details of parameter estimation and structure learning algorithms are discussed next.

2.4.1 PARAMETER LEARNING

First, we describe the algorithms **Stable** uses to estimate the characteristic exponent α from the data matrix D , as well as the parameters $\Gamma = \{\gamma_j|X_j \in \mathcal{X}\}$ and $W_j = \{w_{jk}|X_k \in Pa(X_j)\}$ for any given directed acyclic graph G .

Algorithm 1 Stable

Input: Input data matrix D_I , number of random restarts N_{reps}
Output: α -SG model $B(G, \Theta)$ over \mathcal{X}
 $D \leftarrow \text{Symmetrized}(D_I)$ // Symmetrize the data as per Lemma 8
Estimate α from D // Use log-statistics, Equation 4
Initialize $B(G, \Theta) = \emptyset$
for $i = 1$ **to** N_{reps} **do**
 Initialize a random ordering σ
 $B_\sigma(G, \Theta) = \text{OBS}(D, \alpha, \sigma)$ // Ordering-based search, Algorithm 4
 if $S_{\text{MDC}}(B_\sigma | D) > S_{\text{MDC}}(B | D)$ **then**
 $B = B_\sigma$
 end if
end for

Estimating the global parameter α : Log statistics can be used to estimate the characteristic exponent α from the centered and symmetrized variables in \mathcal{X} (Kuruoglu, 2001).

Algorithm: Since every linear combination of variables in $\hat{\mathcal{X}}$ has the same α , if we define

$$\begin{aligned} \hat{\mathcal{X}} &= \sum_{i=1}^{|\hat{\mathcal{X}}|} \hat{X}_i, \text{ then} \\ \alpha &= \left(\frac{L_2}{\psi_1} - \frac{1}{2} \right)^{-1/2} \\ L_2 &\equiv \mathbb{E} \left[\left(\log |\hat{\mathcal{X}}| - \mathbb{E}[\log |\hat{\mathcal{X}}|] \right)^2 \right] \\ \psi_1 &\equiv \left. \frac{d^2}{dy^2} \Gamma(y) \right|_{y=1} = \frac{\pi^2}{6} \end{aligned} \quad (4)$$

Estimating the dispersion γ_j , and regression parameters $W_j = \{w_{jk} | X_k \in P_a(X_j)\}$
If $\gamma_j(W_j)$ is the dispersion parameter for the distribution of $Z_j = X_j - \sum_{X_k \in P_a(X_j)} w_{jk} X_k$, then the minimum dispersion criterion selects regression parameters

$$W_j^* = \arg \min_{\alpha} \frac{1}{\alpha} \log \gamma_j(W_j)$$

Minimum dispersion regression coefficients are estimated using a connection between the l_p -norm of a stable random variable and the dispersion parameter γ (Zolotarev, 1957; Kuruoglu, 2001) given above in Lemma 9.

This lemma tells us that within a constant term $\log C(p, \alpha)$, minimizing $\frac{1}{\alpha} \log \gamma_j$ is identical to minimizing the l_p -norm $\|Z_j\|_p \equiv \left(\sum_{\lambda=1}^N |Z_{j,\lambda}|^p \right)^{1/p}$ for $-1 < p < \alpha$.

$$W_j^* = \arg \min \log \left(\|Z_j\|_p \right) \equiv \arg \min \log \left(\sum_{\lambda=1}^N |Z_{j,\lambda}|^p \right)^{1/p}$$

Algorithm 2 IRLS // Find the least l_p norm regression coefficients

Input: N dimensional vector for realizations of the child node Y , $N \times M$ matrix X of realizations of the parent set $P_a(Y)$, tolerance ϵ and $p \in (0, 2]$
Output: M dimensional vector of regression co-efficients $W^* = \arg \min_W \|Y - XW\|_p$
Initialize W with OLS co-efficients $W = (X^T X)^{-1} X^T Y$
repeat
 Initialize buffer for current regression coefficients $\beta = W$
 Initialize a diagonal $N \times N$ matrix Ω from β for weighted least squares regression
 $\Omega_{ij} = \delta_{ij} (Y_i - (XW)_i)^{p-2} \forall i, j \in \{1, \dots, N\}$
 Update regression coefficients vector $W = (X^T \Omega X)^{-1} (X^T \Omega Y)$
until $\|\beta - W\|_2 < \epsilon$ // Change in regression coefficients is within tolerance

Algorithm: Minimization of the l_p norm is performed by the *iteratively least squares* (IRLS) algorithm (Byrd and Payne, 1979). Briefly, the IRLS algorithm repeatedly solves an instance of the weighted least squares problem to achieve successive estimates for the least l_p norm coefficients (Algorithm 2). IRLS is attractive since rigorous convergence guarantees can be given (Daubechies et al., 2010) and the method is easy to implement since several software packages are available for the weighted least squares problem. Even though the IRLS objective is no longer convex for $p < 1.0$, Daubechies et al. (2010) show that under certain sparsity conditions, the algorithm can recover the true solution. Simulations described in Section 3.1 tend to support this observation.

For experiments described in this manuscript, **Stable** used two values of p for l_p -norm estimation. For learning regression coefficients during structure learning, IRLS was implemented with $p = \alpha/1.01$, since lower values tended to give noisier estimates (possibly due to numerical errors). However, we also found that estimating the term $\log C(p, \alpha)$ is prone to numerical errors for small values of $|\alpha - p|$. Therefore, we ignore this constant term during structure learning since it is common to all candidate structures. **Stable** estimates the dispersion parameters γ_j after structure learning, by computing the l_p -norm for p sufficiently smaller than α (e.g. $\alpha/10 \leq p \leq \alpha/2$ and applying Lemma 9).

2.4.2 STRUCTURE LEARNING

Searching the space of all network structures can be performed through any of the popular hill-climbing algorithms. In this paper we used the *ordering-based search* (OBS) algorithm (Teyssier and Koller, 2005) to search for a local optimum in the space of all directed acyclic graphs. The algorithm starts with an initial ordering σ and then learns a DAG consistent with σ (i.e., all parents of each node must have a lower order). This part of structure learning is performed via a subroutine K2Search (Algorithm 3), which is a modified version of the hill-climbing based K2Search algorithm Cooper and Herskovits (1992). K2Search starts with an empty parent set for each node $X_i \in \mathcal{X}$ and greedily adds edges until the MDC based score $FS(X_i, P_a(X_i) | D, \alpha) = -\frac{N}{\alpha} \log \gamma_i - \frac{|P_a(X_i)|}{2} \log N$ reaches a local maximum. The main difference from Gaussian graphical models (Heckerman et al., 2000; Schmidt et al., 2007) is that K2Search scores each family based on least l_p norm instead

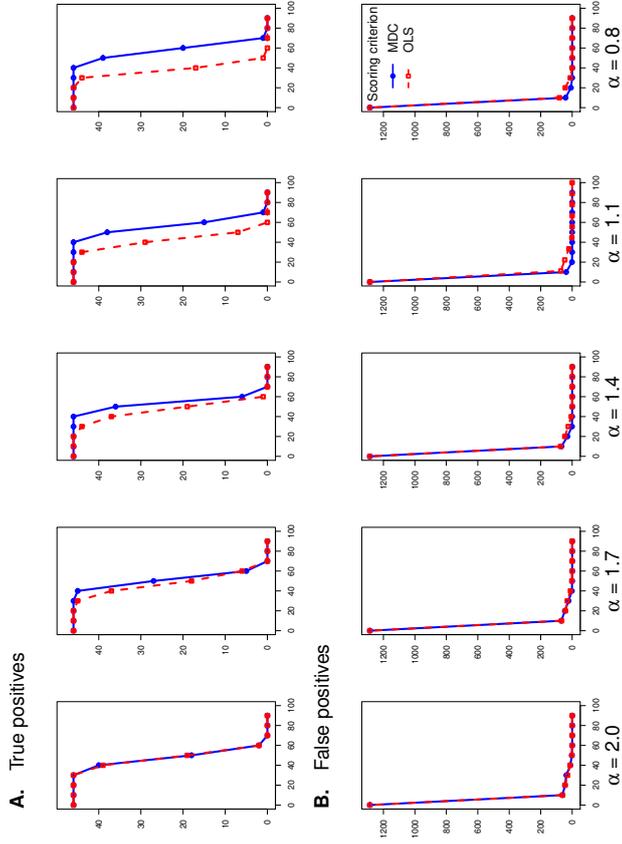


Figure 2: The ALARM network - Inferred structure. Comparative performance of MDC based Stable algorithm (solid blue curves) versus an identical algorithm based on OLS score (dashed red curves). Vertical axes show true positives in **A** and false positives in **B**, for directed edges present in the input network. Horizontal axes show respective confidence (percentage of simulated data sets with the feature)

goal was to assess **Stable** in terms of its performance at structure learning and estimation of stable noise parameters for a variety of regression coefficients.

We performed five sets of experiments for each network, corresponding to different values of $\alpha = 0.8, 1.1, 1.4, 1.7, 2.0$. For each set of experiments, we chose $\rho = 1.0$, $\beta = 0.9$ and $\gamma = 1.0$. We chose such a high skew ($\beta = 0.9$) in the input data to test our algorithm on its ability to symmetrize and correctly learn (possibly) difficult problem instances. Instead of β however, we report a related parameter $\theta = \arctan(\beta \tan \alpha \frac{\rho}{\gamma})$ which can be inferred more robustly in practice since it avoids the singularity near $\alpha = 2$ (Kuruoglu, 2001). We used the zeroth order signed moments based method for estimating θ (Kuruoglu, 2001).

$$\theta_i = \frac{\alpha\pi}{2N} \sum_{\lambda=1}^N \text{sign}(X_{i,\lambda}), \quad \forall X_i \in \mathcal{X} \quad (6)$$

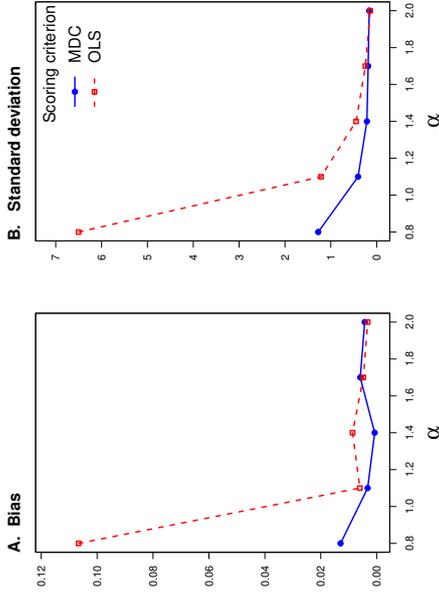


Figure 3: The ALARM network - Estimated regression parameters.

We report two set of results for each network : structure learning and parameter estimation. For convenience, we describe the results for the ALARM network first (results for other data sets are provided in Appendix B).

3.1.1 INFERRED STRUCTURE

Figure 2 shows the comparative performance of MDC and OLS based approaches. Each curve shows the number of inferred directed edges. Figure 2A, B show the number of true positives and true negatives at a given confidence level (percentage of simulated data sets where the directed edge was learnt). Solid (blue) curves show the performance of MDC and dashed (red) curves show OLS based method. The results are along expected lines with the difference between the two getting larger as α is varied away from 2.0. One clear trend is that while the sensitivity to true positive detection degrades for OLS (Type II errors) as α decreases, the MDC based method remain robust to changes in α . Both methods are however quite reliable at not inferring incorrect edges (false positives or Type I errors). Similar behavior is observed for other data sets as well (Appendix B).

3.1.2 ESTIMATED PARAMETERS

Figure 3 shows the comparative performance of MDC and OLS scores in estimating regression coefficients. Figure 3A shows the bias in mean estimates (in absolute magnitude) and Figure 3B, the standard deviation around the mean in estimated coefficients and are averaged over all true positives and all simulated data sets. Note that each of the 100 simulated data set had regression coefficients sampled independently from $[-1/2, 1/2]$. OLS had a much higher standard deviation and bias at low α . As with structure learning, this pattern was consistently observed for other network topologies as well (Appendix B).

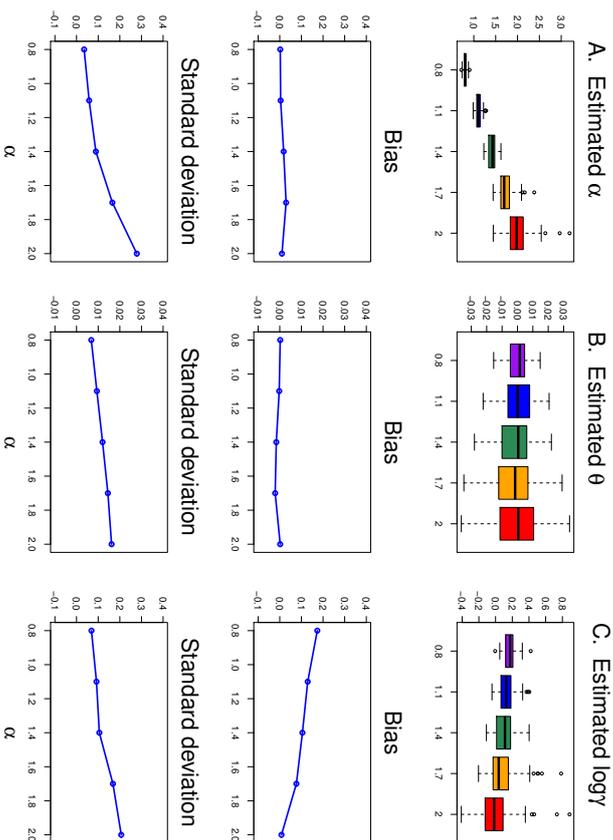


Figure 4: The ALARM network - Estimated noise parameters.

We also assessed the ability of `Stable` to infer α -stable noise parameters accurately and robustly. However, we could not show a comparative performance since OLS scores assume Gaussian noise. Figure 4 shows the box plot and basic statistics for the estimates for α , θ and $\log \gamma$ from the symmetrized data set (node specific parameters θ and $\log \gamma$ are reported as averages).

$$\alpha, \quad \theta \equiv \frac{1}{|X|} \sum_i \arctan(\beta_i \tan \alpha_i^{\frac{\pi}{2}}), \quad \log \gamma = \frac{1}{|X|} \sum_i \log \gamma_i$$

Both α and θ estimates have low bias and standard deviation for each of the five data sets. But, $\log \gamma$ estimates show a clear tendency to overestimate the dispersion in noise at very low α . This is however a difficult parameter domain for most existing methods for parameter estimation, even for univariate α stable densities (Kurugolu, 2001). As with other inferences, the performance of `Stable` is again robust to changes in network topology (Appendix B).

ID	Ethnicity	Location	# Samples	# Genes/Probes
CEU	Caucasians	Utah, USA	109	21800
CHB	Han Chinese	Beijing, China	80	21800
GHI	Gujarati Indians	Houston, USA	82	21800
JPT	Japanese	Tokyo, Japan	82	21800
LWK	Luhya	Webuye, Kenya	83	21800
MEX	Mexican	Los Angeles, USA	45	21800
MKK	Maasai	Kituyawa, Kenya	138	21800
YRI	Yoruba	Ibadan, Nigeria	108	21800

Table 2: The HapMap III population groups and selected microarray probes as reported by Stranger et al. (2012).

3.2 Gene Expression Microarray Data

In this section, we describe two sets of analyses for gene expression microarray data from phase III of the HapMap project³. Our approach models the set of gene expression profiles as a multivariate stable distribution that can be represented by an α -SG model. The first set of experiments aimed at comparing the prediction accuracy of MDC with OLS-based structure learning via ten-fold cross-validation (Section 3.2.2). The results of these experiments establish the utility of heavy-tailed models for gene expression profiles.

Next, we apply α -SG models to the problem of quantifying differential expression (DE) of a gene between samples belonging to different conditions. This is a common task in gene expression-based analyses in contemporary genomics. However, popular methods for detecting differentially expressed genes usually assume the expression profile for each gene to be independent of others. Based on this assumption, DE quantification is performed by testing the null hypothesis that the log-expression of each gene is identical across the observed conditions and using the corresponding p-value as a measure of DE. In Section 3.2.3, we develop `SGEX`, a new technique for quantifying differential expression of each gene that is based on α -SG models. We apply `SGEX` to quantify the DE for a gene in each population group within the HapMap data. Contrary to most existing methods, `SGEX` takes into account both the heavy-tailed behavior of gene expression densities, as well as linear dependencies between mRNA expression of different genes.

3.2.1 DATA NORMALIZATION

We downloaded pre-processed data for 727 individuals from eight global population groups as reported in Stranger et al. (2012). Details about the eight population groups are provided in Table 2. For each individual, the input data represented log-intensities for 21800 microarray probes⁴ that were quantile and median normalized, as described in the original paper (Stranger et al., 2012). These microarray intensities provide a measure for mRNA concen-

3. Data sets can be downloaded from the Array Express database <http://www.ebi.ac.uk/arrayexpress/> using Series Accession Numbers E-MTAB-198 and E-MTAB-264.

4. Each selected probe mapped to a unique, autosomal Ensembl gene. Ensembl gene IDs are available at <http://www.ensembl.org>.

tration within a sample of lymphoblastoid cells from each individual. Before performing structure learning, we further processed each probe intensity as follows :

1. The log-intensity $l(i)$ for each probe i was median-centered to obtain transformed log-intensities $ml(i)$, ie., the number of samples with positive log-intensity was half (or 0.5 less than) the total ($= 363 = \lfloor 727/2 \rfloor$). This is a standard technique for learning Gaussian graphical models from gene expression data and does not affect the network structure.
2. The median-centered log-intensities were used to assign a rank $R(i)$ to each probe i , in decreasing order of variance. Even for α -stable distributions, variance of log transformed data is finite (Kuruoglu, 2001). This is also a standard technique for restricting computing time by selecting a subset of genes with most variation.
3. The median-centered log-intensities $\{ml(i) | R(i) \leq 21800\}$ were exponentiated to $\mathcal{I} = \{2^{ml(i)} | R(i) \leq 21800\}$.
4. The exponentiated-median-centered log-intensities $\mathcal{I}_k = \{2^{ml(i)} | R(i) \leq k \leq 21800\}$ for the top k ranked probes were provided as input to **Stable** (for cross-validation) and **SGEX** (for DE quantification, as described in Section 3.2.3). In the experiments reported here $k = 100$.

We estimated α over 1000 resampled bootstrap replicates of the data. This was meant as a diagnostic to assess the heavy-tailed nature of the intensities. As shown in Figure 5A, the data suggests a clear departure from a Gaussian profile.

3.2.2 CROSS-VALIDATION ANALYSIS

We performed a ten-fold cross-validation for the top 100 ranked probes from the HapMap data. Since we wanted to compare MDC with OLS-based learning, we report goodness of fit of the graphical model B on the test set $T = \{T_1, \dots, T_N\}$ in terms of log fractional lower order moments :

$$LFLOM(T|B, p) = \sum_{X_i \in \mathcal{X}} \left[\frac{1}{p} \log E(|Z_i|^p) \right] = \sum_{X_i \in \mathcal{X}} \left[\frac{1}{p} \log E(|X_i - \sum_{X_j \in \mathcal{P}_A(X_i)} w_{ij} X_j|^p) \right]$$

where, w_{ij} represents the regression co-efficient for the edge (X_j, X_i) . Clearly, if most of the variation in X_i can be explained by the parent set $\mathcal{P}_A(X_i)$, the corresponding $LFLOM$ will be small. For $p = 2$, $LFLOM$ is identical to the negative log-likelihood for Gaussian graphical models⁵. However, the second order moment diverges for $\alpha < 2$ (Lemma 9). Therefore, $LFLOM$ provide a more robust estimate for evaluating the model on test set for heavy-tailed noise ($\alpha < 2$).

Figure 5B shows the average (over the ten-folds) of $LFLOM$ for MDC (blue) and OLS-based (red) models. In each case, the curves show the difference in $LFLOM$ between optimal (MDC or OLS) network and an empty network (NULL). This allows us to also assess the deterioration in test set performance by treating each gene as an independent

5. Note that the noise term Z_i has zero mean, since the data is centro-symmetrized before cross-validation.

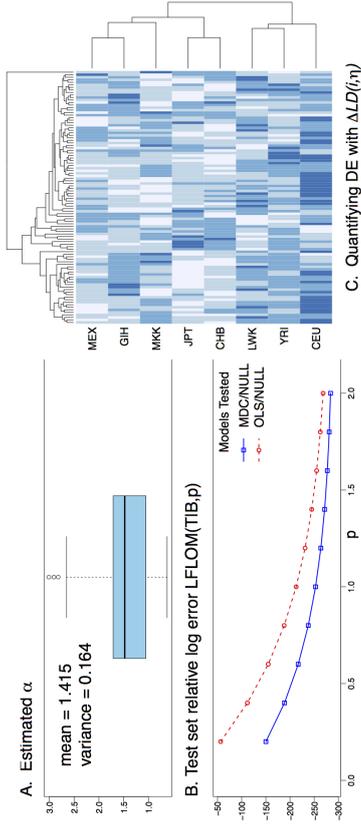


Figure 5: Test set performance and differential expression quantification with **SGEX**. **A** shows a box plot of estimated α over 1000 bootstrap replicates. **B** shows comparative Test set performance for MDC and OLS based networks relative to an empty network (no edges). **C** shows a heat map of ΔLD that quantifies differential expression of a gene. The color for each column is normalized by scaling and centering.

random variable (a common assumption in DE quantification). Although the data set contains only 727 samples, we see a clear improvement in test set performance of α -SG models (MDC curve) relative to Gaussian graphical models (OLS curve).

3.2.3 QUANTIFYING DIFFERENTIAL EXPRESSION WITH SGEX

Finally, we discuss **SGEX**, a new technique for quantifying differential expression using α -SG models. **SGEX** is based on cross-validation for assessing DE of a gene across different conditions. For the HapMap data, we chose each of the eight population groups in turn as the test set and learnt the optimal α -SG model for the rest of the samples. We then estimated $\Delta LD(i, \eta)$, the change in negative log-likelihood per sample between the test set η and the training set as a measure of DE for each probe i

$$\Delta LD(i, \eta) = \frac{1}{p} \left[\log E_{\eta}(|Z_i|^p) - \log E_{\bar{\eta}}(|Z_i|^p) \right], \quad p \in (-1, \alpha)$$

Here, $E_{\eta}(\cdot)$ is the expectation value for population η (test set) and $E_{\bar{\eta}}(\cdot)$ for the rest (training set). Note that Lemma 9 guarantees that RHS of the previous equation is indeed independent of p . For the calculation reported here $p = \alpha/1.01$, just as it was during structure learning. Thus, $\Delta LD(i, \eta)$ measures the average increase (or decrease) in log-dispersion for the noise variable Z_i corresponding to probe i within population η . This density is represented as a heat map in Figure 5C. We should point out that a higher (or lower) dispersion

for the noise variable associated with a gene in the test set does not necessarily imply over (or under) expression of a gene in the test set population. The change in dispersion could also be due to a change in network topology or regression coefficients for the test set population.

4. Discussion

In this paper we have introduced and developed the theory for efficiently learning α -SG models from data. In particular, one of the main contributions of this paper is to show how the BIC can be asymptotically reduced to the MDC for α -SG models. This result makes it feasible to efficiently learn the structure of these models, since the log-likelihood term does not have a closed form expression in general. We have also empirically validated the resultant algorithm `Stable` on both simulated and microarray data. In both cases, the presence of heavy-tailed noise has a clear effect on learning performance of OLS based methods. Based on these results, we recommend a bootstrapped estimation of α as an effective and computationally efficient diagnostic to assess the applicability of OLS based Gaussian graphical models.

We have also described `SGEX`, a new technique for quantifying differential expression from microarray data. α -SG models may also have wider applicability to other aspects of computational biology, especially to data from next-generation sequencing technologies. In addition to mRNA expression measurements (RNA-seq experiments), α -SG models may prove helpful for other experiments, such as protein-DNA binding (ChIP-seq experiments) and DNA accessibility measurements (DNase-seq and FAIRE-seq experiments).

Finally, we should mention that there are several potential applications of α -SG models beyond computational biology. In particular, image processing provides several problem instances where there is a need to relate different regions of the image. For example, functional magnetic resonance imaging (fMRI) experiments generate a series of images highlighting activity sites in the brain in response to stimuli. Bayesian networks are an effective way of modeling statistical relations between different areas of the brain and the stimuli (Li et al., 2011). Stable distributions may provide a better model for such applications. Another image processing application with potentials for α -SG models is remote sensing images of the earth (Mustrafai et al., 2012) where image histograms demonstrate clearly skewed and heavy tailed characteristics (Kuruoglu and Zerbiba, 2004). Traffic modeling (Castillo et al., 2012) and financial data analysis (Bonato, 2012) are also promising application areas.

5. Software Availability

Source code for `Stable` and data sets used here are available at <https://sourceforge.net/projects/sgmodels/>. `SGEX` is available upon request from the first author.

Acknowledgments

E.E. Kuruoglu was funded by the Alexander von Humboldt Foundation with an Experienced Research Fellowship.

Appendix A

In this section we provide the proof for Lemma 3

Lemma 3 Every d -dimensional distribution with a characteristic function of the form

$$\Phi(q|\alpha, \tilde{\mu}, \Lambda) = \prod_{k=1}^d \phi(c_k^T q|\alpha, \beta_k, \gamma_k, \mu_k) \quad \text{where, } c_k, q \in \mathbb{R}^d$$

represents a multivariate stable distribution with a discrete spectral measure Λ .

Proof Assume the following ansatz for the spectral measure Λ ,

$$\begin{aligned} \Lambda_k &= \frac{\|c_k\|_2^{\alpha} \gamma_k}{2} \left((1 + \beta_k) \delta(s - \frac{c_k}{\|c_k\|_2}) + (1 - \beta_k) \delta(s + \frac{c_k}{\|c_k\|_2}) \right) \\ \Lambda(ds) &= \sum_k \Lambda_k ds \end{aligned}$$

and location vector $\tilde{\mu}$,

$$\begin{aligned} \eta_k(c_k|\alpha, \beta_k, \gamma_k, \mu_k) &= \begin{cases} \mu_k & \alpha \neq 1 \\ \mu_k - \frac{2\beta_k c_k}{\pi} \log \|c_k\|_2 & \alpha = 1 \end{cases} \\ \tilde{\mu} &= \sum_{k=1}^d \eta_k(c_k|\alpha, \beta_k, \gamma_k, \mu_k) c_k \in \mathbb{R}^d \end{aligned}$$

Upon substitution into the parametrization in Definition 2 we get

$$\begin{aligned} \int_{S_d} \psi(s^T q|\alpha) \Lambda_k ds &= \frac{\|c_k\|_2^{\alpha} \gamma_k}{2} \left((1 + \beta_k) \psi\left(\frac{c_k^T q}{\|c_k\|_2}|\alpha\right) + (1 - \beta_k) \psi\left(-\frac{c_k^T q}{\|c_k\|_2}|\alpha\right) \right) \\ &= \frac{\|c_k\|_2^{\alpha} \gamma_k}{2} \frac{\|c_k^T q\|_2^{\alpha}}{\|c_k\|_2^{\alpha}} \left((1 + \beta_k) (1 - \text{sign}(c_k^T q) r\left(\frac{c_k^T q}{\|c_k\|_2}, \alpha\right)) \right. \\ &\quad \left. + (1 - \beta_k) (1 + \text{sign}(c_k^T q) r\left(\frac{c_k^T q}{\|c_k\|_2}, \alpha\right)) \right) \\ \implies \int_{S_d} \psi(s^T q|\alpha) \Lambda_k ds &= \gamma_k |c_k^T q|^{\alpha} \left(1 - \nu \beta_k \text{sign}(c_k^T q) r\left(\frac{c_k^T q}{\|c_k\|_2}, \alpha\right) \right) \\ &= \gamma_k |c_k^T q|^{\alpha} \left(1 - \nu \beta_k \text{sign}(c_k^T q) r\left(\frac{c_k^T q}{\|c_k\|_2}, \alpha\right) \right) \\ &\quad - \nu \beta_k \gamma_k |c_k^T q|^{\alpha} \text{sign}(c_k^T q) \left(r\left(\frac{c_k^T q}{\|c_k\|_2}, \alpha\right) - r\left(-\frac{c_k^T q}{\|c_k\|_2}, \alpha\right) \right) \end{aligned}$$

$$\text{Since, } r\left(\frac{c_k^T q}{\|c_k\|_2}, \alpha\right) - r\left(-\frac{c_k^T q}{\|c_k\|_2}, \alpha\right) = \begin{cases} 0 & \alpha \neq 1 \\ \frac{2}{\pi} \log \|c_k\|_2 & \alpha = 1 \end{cases}$$

$$\begin{aligned} \nu \beta_k \gamma_k |c_k^T q|^{\alpha} \text{sign}(c_k^T q) \left(r\left(\frac{c_k^T q}{\|c_k\|_2}, \alpha\right) - r\left(-\frac{c_k^T q}{\|c_k\|_2}, \alpha\right) \right) &= \begin{cases} 0 & \alpha \neq 1 \\ \nu \beta_k \gamma_k \frac{\|c_k^T q\|_2^{\alpha}}{\|c_k\|_2^{\alpha}} \log \|c_k\|_2 & \alpha = 1 \end{cases} \\ &= \begin{cases} \nu \beta_k \gamma_k \frac{\|c_k^T q\|_2^{\alpha}}{\|c_k\|_2^{\alpha}} \log \|c_k\|_2 & \alpha \neq 1 \\ \nu \beta_k \gamma_k \frac{\|c_k^T q\|_2^{\alpha}}{\|c_k\|_2^{\alpha}} \log \|c_k\|_2 & \alpha = 1 \end{cases} \\ &= \begin{cases} \nu \beta_k \gamma_k \frac{\|c_k^T q\|_2^{\alpha}}{\|c_k\|_2^{\alpha}} \log \|c_k\|_2 & \alpha \neq 1 \\ \nu \beta_k \gamma_k \frac{\|c_k^T q\|_2^{\alpha}}{\|c_k\|_2^{\alpha}} \log \|c_k\|_2 & \alpha = 1 \end{cases} \\ &= \nu \beta_k \gamma_k \left(\mu_k - \eta_k(c_k|\alpha, \beta_k, \gamma_k, \mu_k) \right) \end{aligned}$$

$$\begin{aligned}
 \implies \int_{S_{it}} \psi(s^T q | \alpha) \Lambda_k ds &= -\log \phi(c_k^T q | \alpha, \beta_k, \gamma_k, \mu_k) + \eta_k(c_k | \alpha, \beta_k, \gamma_k, \mu_k) c_k^T q \\
 \implies \log(\Phi(q | \alpha, \tilde{\mu}, \Lambda)) &= -\int_{S_{it}} \psi(s^T q | \alpha) \Lambda(ds) + i \tilde{\mu} q \\
 &= -\sum_{k=1}^d \int_{S_{it}} \psi(s^T q | \alpha) \Lambda_k ds + i \sum_{k=1}^d \eta_k(c_k | \alpha, \beta_k, \gamma_k, \mu_k) c_k^T q \\
 \implies \log(\Phi(q | \alpha, \tilde{\mu}, \Lambda)) &= \sum_{k=1}^d \log \phi(c_k^T q | \alpha, \beta_k, \gamma_k, \mu_k) \\
 \implies \Phi(q | \alpha, \tilde{\mu}, \Lambda) &= \prod_{k=1}^d \phi(c_k^T q | \alpha, \beta_k, \gamma_k, \mu_k)
 \end{aligned}$$

■

Appendix B The BARLEY network

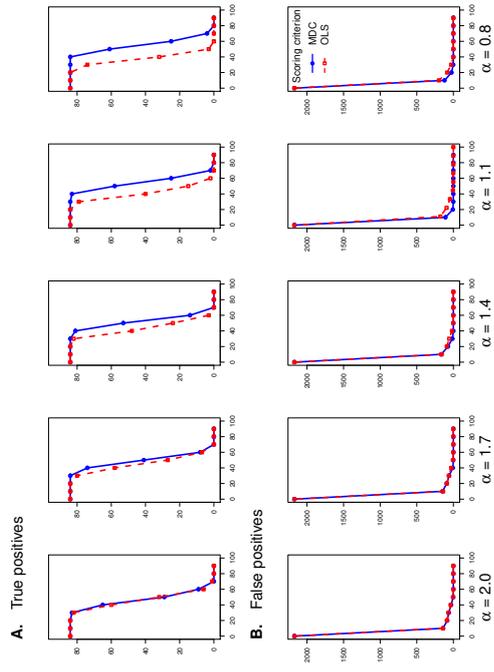


Figure 6: The BARLEY network - Inferred structure

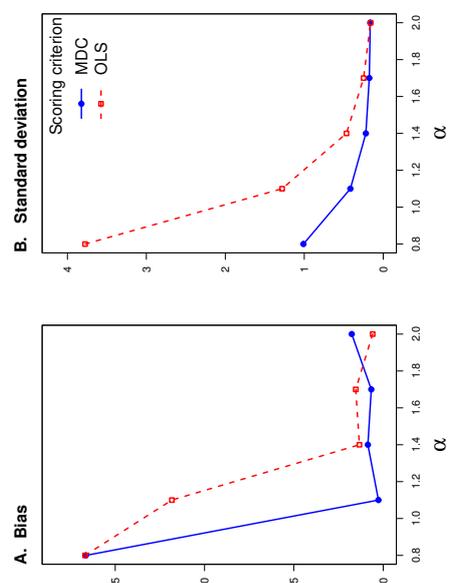


Figure 7: The BARLEY network - Estimated regression parameters.

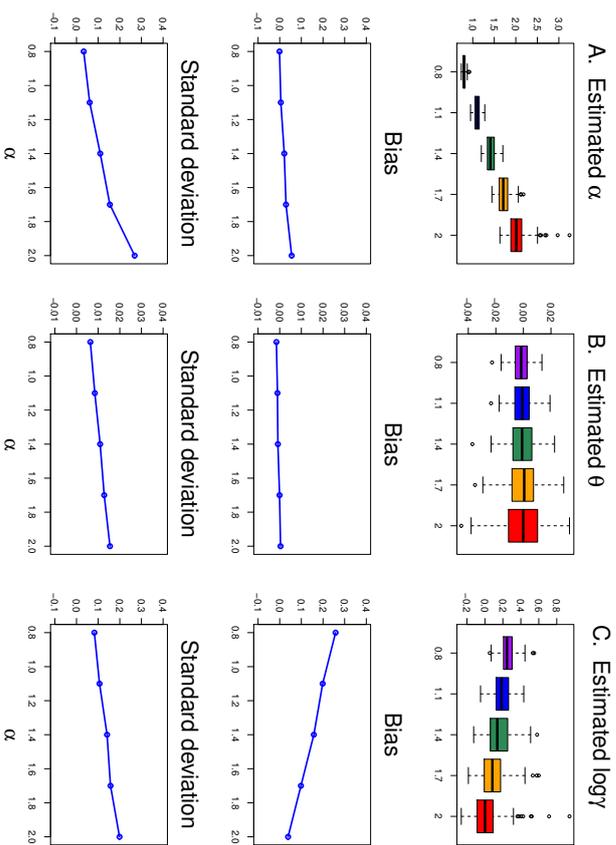


Figure 8: The BARLEY network - Estimated noise parameters

The CHILD network

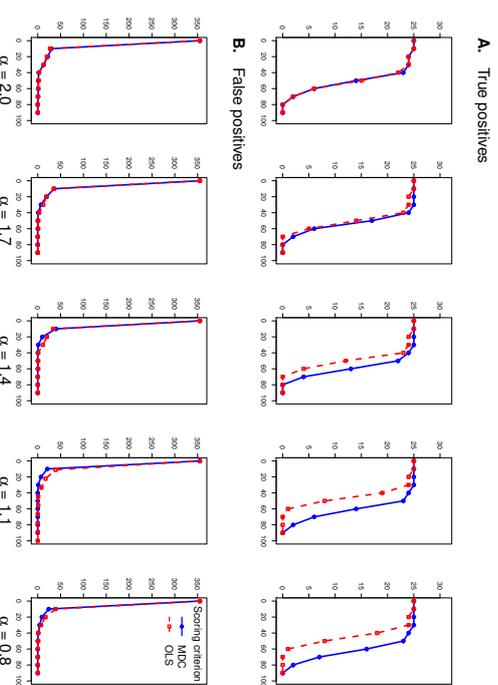


Figure 9: The CHILD network - Inferred structure

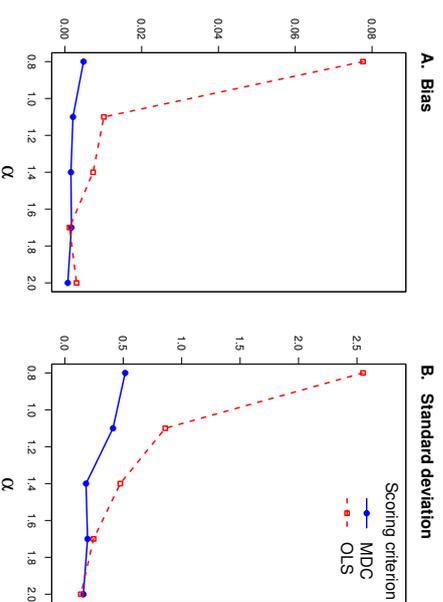


Figure 10: The CHILD network - Estimated regression parameters.

The INSURANCE network

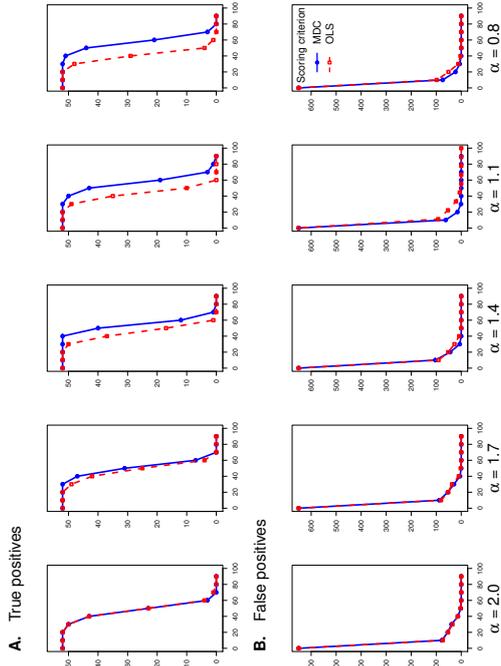


Figure 12: The INSURANCE network - Inferred structure

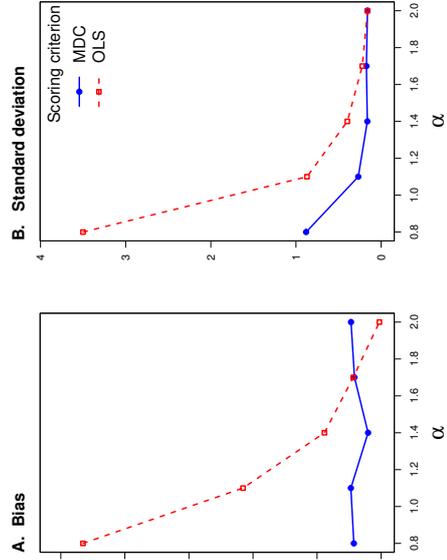


Figure 13: The INSURANCE network - Estimated regression parameters.

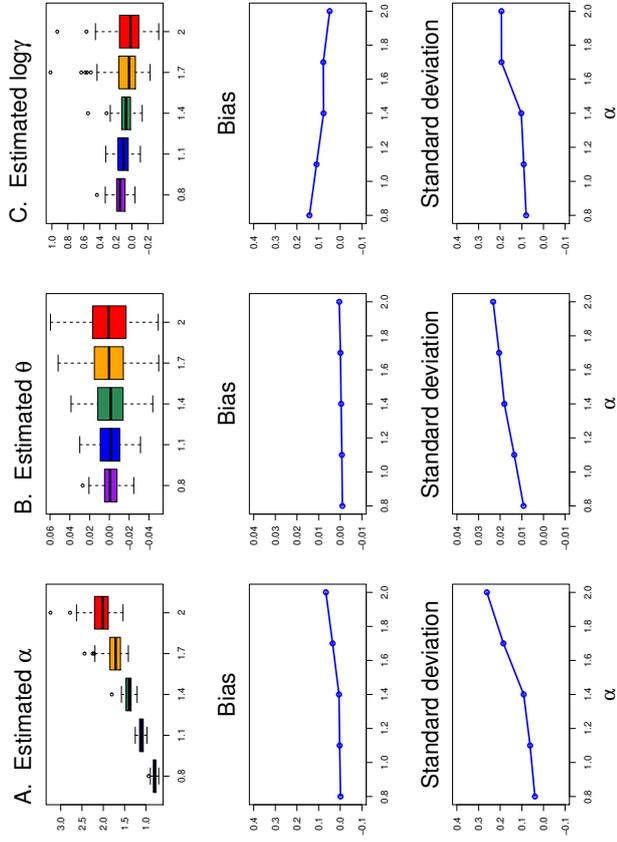


Figure 11: The CHILD network - Estimated noise parameters

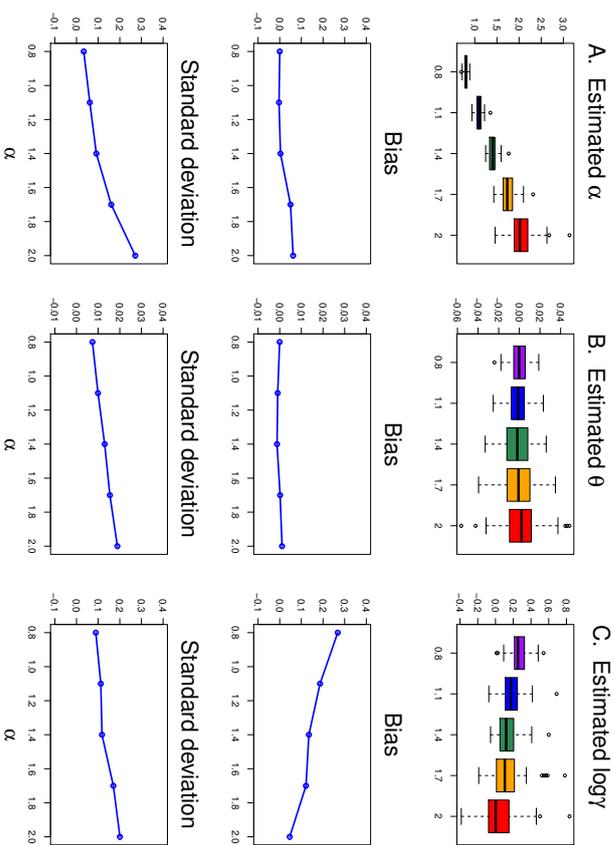


Figure 14: The INSURANCE network - Estimated noise parameters

The MIDDEW network

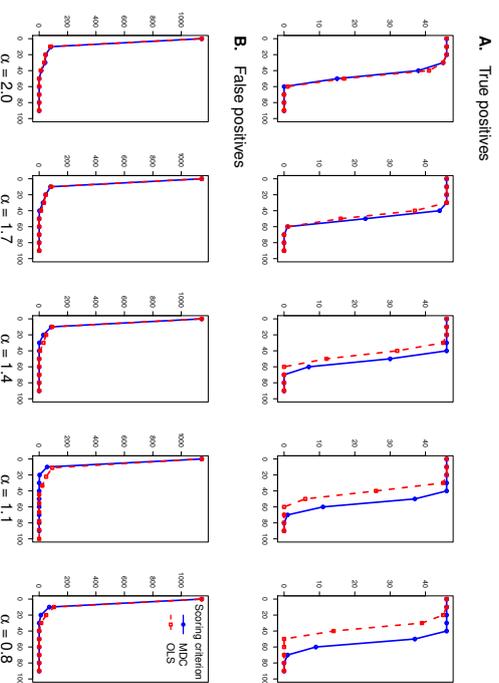


Figure 15: The MIDDEW network - Inferred structure

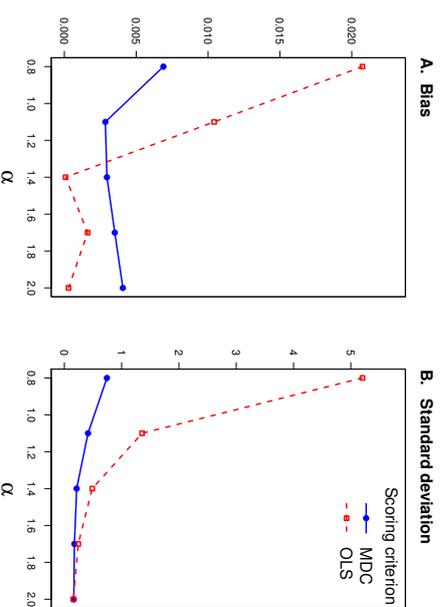


Figure 16: The MIDDEW network - Estimated regression parameters.

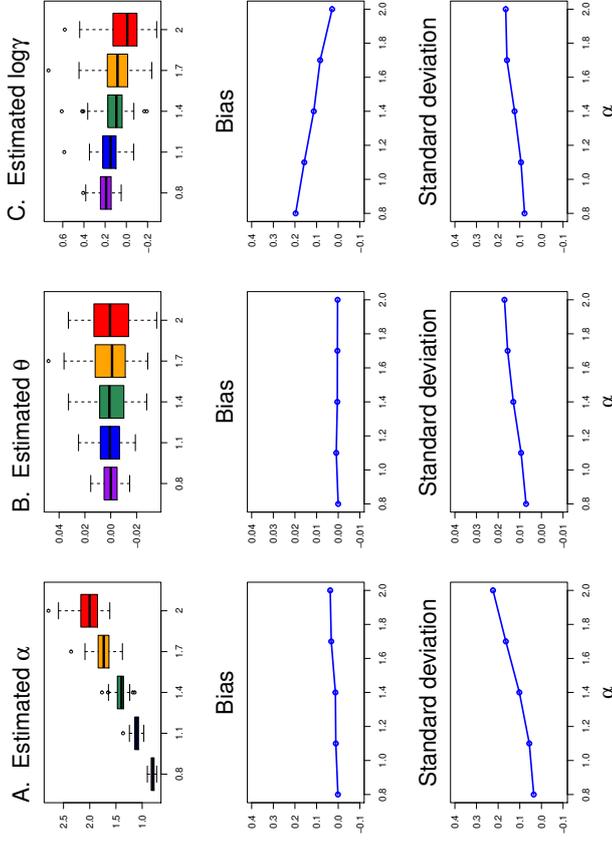


Figure 17: The MILDEW network - Estimated noise parameters

References

A. Achim and E. E. Kuruoglu. Image denoising using bivariate α -stable distributions in the complex wavelet domain. *IEEE Signal Processing Letters*, 12(1):17–20, 2005.

A. Achim, A. Bezerianos, and P. Tsakalides. Novel Bayesian multiscale method for speckle removal in medical ultrasound images. *IEEE Transactions on Medical Imaging*, 20(8):772–783, 2001.

A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini. Tissue classification with gene expression profiles. *Journal of Computational Biology*, 7(3-4):559–583, 2000.

J. Berger and B. Mandelbrot. A new model for error clustering in telephone circuits. *IBM Journal of Research and Development*, pages 224–236, 1963.

D. Bickson and C. Guestrin. Inference with multivariate heavy-tails in linear models. In *Proceedings of NIPS*, 2011.

M. Bonato. Modeling fat tails in stock returns: a multivariate stable-GARCH approach. *Computational Statistics*, 27(3):499–521, 2012.

R. H. Byrd and D. A. Payne. Convergence of the iteratively reweighted least squares algorithm for robust regression. Technical Report 313, The Johns Hopkins University, Baltimore, MD, 1979.

E. Castillo, M. Nogal, M. Menéndez, J., S. Sánchez-Cambrotero, and P. Jiménez. Stochastic demand dynamic traffic models using generalized beta-Gaussian Bayesian networks. *IEEE Transactions on Intelligent Transportation Systems*, 13(2):565–581, 2012.

J. Chambers, C. Mallows, and B. Stuck. A method for simulating stable random variables. *Journal of the American Statistical Association*, 71(354):340–344, 1976.

G. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics*, LXIII:1–38, 2010.

W. Feller. *An Introduction to Probability Theory, vol. I, vol. II*. John Wiley, New York, 1968.

N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659):799–805, 2004.

N. Friedman, M. Linial, I. Nachman, and D. Pe’er. Using Bayesian networks to analyze expression data. *Journal of computational biology*, 7(3-4):601–620, 2000.

J. R. Gallardo, D. Makrakis, and L. Orozco-Barbosa. Use of α -stable self-similar stochastic processes for modeling traffic in broadband networks. *Performance Evaluation*, 40(1):71–98, 2000.

C. D. Hardin Jr. Skewed stable variables and processes. Technical Report 79, Univ. North Carolina, Chapel Hill, 1984.

D. Heckerman, D. Chickering, C. Meek, R. Rounthwaite, and C. Kadie. Dependency networks for density estimation, collaborative filtering, and data visualization. *Journal of Machine Learning Research*, 1:49–75, 2000.

International HapMap 3 Consortium and others. Integrating common and rare genetic variation in diverse human populations. *Nature*, 467(7311):52–58, 2010.

D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, Cambridge, MA, 2009.

E. E. Kuruoglu. Density parameter estimation of skewed α -stable distributions. *IEEE Transactions on Signal Processing*, 49(10):2192–2201, 2001.

- E. E. Kurunoglu and J. Zerrubia. Modeling SAR images with a generalization of the Rayleigh distribution. *IEEE Transactions on Image Processing*, 13(4):527–533, 2004.
- P. Lévy. *Calcul des probabilités*. Gauthier-Villars Paris, 1925.
- R. Li, K. Chen, A. S. Fleisher, E. M. Reinman, L. Yao, and X. Wu. Large-scale directional connections among multi resting-state neural networks in human brain: A functional MRI and bayesian network modeling study. *NeuroImage*, 56(3):1035–1042, 2011.
- B. Mandelbrot. The variation of certain speculative prices. *Journal of Business*, 26:394–419, 1963.
- S. B. Montgomery et al. Transcriptome genetics using second generation sequencing in a caucasian population. *Nature*, 464(7289):773–777, 2010.
- Y. T. Mustafa, V. A. Tolpekin, and A. Stein. Application of the expectation maximization algorithm to estimate missing values in gaussian bayesian network modeling for forest growth. *IEEE Transactions on Geoscience and Remote Sensing*, 50(5):1821–1831, 2012.
- C. L. Nikias and M. Shao. *Signal Processing with Alpha-Stable Distributions*. Wiley, New York, 1995.
- J. P. Nolan. *Stable Distributions - Models for Heavy Tailed Data*. Birkhäuser, Boston, Chapter 1 online at academic2.american.edu/~jpnolan edition, 2013.
- J. P. Nolan. Linear and nonlinear regression with stable errors. *Journal of Econometrics*, 172(2): 86–194, 2013.
- J. P. Nolan and B. Rajput. Calculation of multi-dimensional stable densities. *Communications in Statistics - Simulation and Computation*, 24(3):551–566, 1995.
- J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Mateo, CA, 1988.
- D. Salas-Gonzalez, E. E. Kurunoglu, and D. P. Ruiz. Modelling and assessing differential gene expression using the alpha stable distribution. *The International Journal of Biostatistics*, 5(1):1–24, 2009a.
- D. Salas-Gonzalez, E. E. Kurunoglu, and D. P. Ruiz. A heavy-tailed empirical bayes method for replicated microarray data. *Computational Statistics & Data Analysis*, 53(5):1535–1546, 2009b.
- G. Samorodnitsky and M. S. Taqqu. *Stable Non-Gaussian Random Processes*. Chapman and Hall, New York, 1994.
- M. Schmidt, A. Niculescu-Mizil, and K. Murphy. Learning graphical model structure using L1-regularization paths. In *Proceedings of AAAI*, 2007.
- G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.
- B. E. Stranger et al. Patterns of cis regulatory variation in diverse human populations. *PLoS genetics*, 8(4):e1002639, 2012.
- B. W. Stuck. Minimum error dispersion linear filtering of scalar symmetric stable processes. *IEEE Transactions on Automatic Control*, 23:507–509, 1978.
- M. Teyssier and D. Koller. Ordering-based search: A simple and effective algorithm for learning Bayesian networks. In *Proceedings of Uncertainty in Artificial Intelligence (UAI)*, 2005.
- I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006.
- V. M. Zolotarev. Mellin-Stieltjes transforms in probability theory. *Theory Probability Appl*, 2:433–460, 1957.

Bounding the Search Space for Global Optimization of Neural Networks Learning Error: An Interval Analysis Approach

Stavros P. Adam

*Computational Intelligence Laboratory
Department of Mathematics
University of Patras
GR-26110 Patras, Greece*

ADAMSP@UPATRAS.GR

George D. Magoulas

*Department of Computer Science and Information Systems
Birkbeck College, University of London
Malet Street, London WC1E 7HX, UK*

GMAGOULAS@DCS.BBK.AC.UK

Dimitrios A. Karras

*Department of Automation
Technological Educational Institute of Sterea Hellas
34400 Psachna, Evia, Greece*

DAKARRAS@TEIESTE.GR

Michael N. Vrahatis

*Computational Intelligence Laboratory
Department of Mathematics
University of Patras
GR-26110 Patras, Greece*

VRAHATIS@MATH.UPATRAS.GR

Editor: Kevin Murphy

Abstract

Training a multilayer perceptron (MLP) with algorithms employing global search strategies has been an important research direction in the field of neural networks. Despite a number of significant results, an important matter concerning the bounds of the search region—typically defined as a box—where a global optimization method has to search for a potential global minimizer seems to be unresolved. The approach presented in this paper builds on interval analysis and attempts to define guaranteed bounds in the search space prior to applying a global search algorithm for training an MLP. These bounds depend on the machine precision and the term “guaranteed” denotes that the region defined surely encloses weight sets that are global minimizers of the neural network’s error function. Although the solution set to the bounding problem for an MLP is in general non-convex, the paper presents the theoretical results that help deriving a box which is a convex set. This box is an outer approximation of the algebraic solutions to the interval equations resulting from the function implemented by the network nodes. An experimental study using well known benchmarks is presented in accordance with the theoretical results.

Keywords: neural network training, bound constrained global optimization, interval analysis, interval linear equations, algebraic solution

1. Introduction

Multi-layer perceptrons are feed forward neural networks featuring universal approximation properties used both in regression problems and in complex pattern classification tasks. Actually, an MLP is a means used to encode empirical knowledge about a physical phenomenon, i.e., a real world process. This encoding is done in terms of realizing a function F that is close enough to a target function $d = f(x)$ representing the underlying process. The target function is rarely formulated in analytical terms but it is defined as a set of input-output values $\{x_l, d_l\}$, $1 \leq l \leq p$, which is the training set resulting from observations of the underlying process.

More formally, an MLP is used to implement a model $F(x; w)$ about a physical process which is a mapping $F : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^k$ whose values $F(x_l; w)$ have minimum distance from their corresponding d_l , where $x_l = (x_{1,l}, x_{2,l}, \dots, x_{n,l})^T$ and $w = (w_1, w_2, \dots, w_m)^T$. Given that x_l are known, this minimum distance depends on the values of the weights w . This formulates a problem of parametric model identification on the basis of the training set composed of the inputs and their corresponding outputs. Determining the values of the weights is done by an iterative process commonly known as network training, which consists in minimizing some cost function $E : \mathbb{R}^k \rightarrow \mathbb{R}$ of the network output.

The dominant approach when training feedforward neural networks has been to adopt local search in the weight space using some gradient descent technique in order to minimize the output error function. Adaptation of the weights is done with the well known error back-propagation (BP) algorithm. A detailed review of BP and other training algorithms based on gradient descent can be found in Haykin (1999). However, local search techniques are, in the general case, unable to detect with certainty a global or a “good” local minimum, and consequently the corresponding minimizer, as they may stuck in “bad” local minima. Researchers in the area of neural networks have tried to overcome this defect by proposing efficient weight initialization techniques (Adam et al., 2014; Nguyen and Widrow, 1990). Moreover, a number of effective search strategies have been proposed (Magoulas et al., 1997, 1999) which concern definition or adaptation of relevant parameters of the search procedure such as the step-size, the momentum term etc.

The counter approach to local search is to use global search of the free parameter space. The techniques based on global optimization can be either exact or heuristic (Horst and Pardalos, 1995). Exact global optimization methods can be roughly defined as deterministic or stochastic (Pál, 2010). Deterministic global search techniques perform exhaustive search of the weight space and they guarantee to locate a global minimizer (Tang and Koehler, 1994). However, often the time needed for this task is unacceptable, especially for real life applications, and usually it is achieved with excessive use of computational resources. This is mainly due to the size of the parameter space which grows with the number of free parameters (weights) of the neural networks. An alternative approach to deterministic search has been the use of stochastic global search methods which are based on random sampling of the search space adopting random or adaptive random search strategies. Such methods include Monte Carlo techniques (Brooks, 1958; Caffisch, 1998) and multistart methods (Boender et al., 1982). On the other hand, heuristic methods include simulated annealing (Engel, 1988; Kirkpatrick et al., 1983) and techniques based on evolutionary computation such as the genetic algorithm (GA) (Castillo et al., 2000; Montana and Davis, 1989), Par-

title Swarm Optimization (PSO) (Gudise and Venayagamoorthy, 2003; van den Bergh and Engelbrecht, 2000) and differential evolution algorithms (Iloen et al., 2003).

Although these methods possess several desirable characteristics and have proven to be effective in several applications, they cannot guarantee—in the general case—that they are able to detect a global minimizer in the first run. Moreover, their performance, in terms of the overall search time, depends on the scale of the problem. To a large extent, this is due to inappropriate initial conditions because the search space is only heuristically known (Parsopoulos and Vrahatis, 2010). A common technique to reduce the uncertainty about the search space and enhance the efficiency of these methods is to define the region of the search space to be “as large as possible”. A survey of global optimization techniques used for neural network training can be found in Duch and Korczak (1998) as well as in Paganakes et al. (2001, 2006). Finally, it is worth noting that a number of research work has focused on combining local and global search in order to alleviate the disadvantages of local search methods or in order to decrease the overall search time in the case of global search of the weight space (Bagirov et al., 2009; Caprani et al., 1993; Ng et al., 2010; Shiang and Wall, 1996; Voglis and Lagaris, 2009; Xu, 2002).

A common characteristic to all global search procedures used for training MLPs is that the bounds of the search region are intuitively defined. This approach, often, proves to be inadequate to support training MLPs with global optimization methods. Hence, defining effective bounds of the search space is a cornerstone for using global search strategies in real life applications of feedforward neural networks. The approach presented in this paper is based on concepts of interval analysis and attempts to define guaranteed bounds of the search space prior to applying a global search procedure for neural network training. The term “guaranteed” denotes that the bounds of the region defined surely enclose weights that are global minimizers of the error function. Once the box is defined, it is up to the global optimization method to explore the region identified by the proposed approach in order to find the set of weights that minimize the error function of the multilayer perceptron.

The rest of the paper is organized as follows. Section 2 is devoted to a detailed description of the problem background and the motivation for the proposed approach. Section 3 presents the solution proposed to tackle this problem. In Section 4 we derive the theoretical results for bounding the feasible solution set of the problem and discuss these results. In Section 5 we validate the proposed method on real world problems and discuss the experimental results. Finally, Section 6 summarizes the paper with some concluding remarks.

2. Problem Formulation in a Global Optimization Context

Hereafter, we consider feed forward neural networks having 3 layers with n , h , and o nodes in the input, the hidden and the output layers, respectively. It has been proved that standard feedforward networks with only a single hidden layer can approximate any continuous function uniformly on any compact set and any measurable function to any desired degree of accuracy (Cybenko, 1989; Hornik et al., 1989; Kreinovich and Shtinsengtaksin, 1993; White, 1990). Hence, it has general importance to study 3-layer neural networks. On the other hand, as it will be shown, the assumption regarding the number of layers is not restrictive, given that the weights of each layer are treated with respect to the output range of the previous layer and the input domain of the next one. Hence, it is straightforward to extend

this analysis to networks with multiple hidden layers. Moreover, despite the variety of activation functions found in the literature (Duch and Jankowski, 1999), in this paper we have retained the most common assumptions adopted for MLPs. This means that the nodes in the hidden layer are supposed to have a sigmoid activation function which may be one of the following:

$$\text{logistic sigmoid} : \sigma_1(\text{net}) = \frac{1}{1 + e^{-\alpha \text{net}}}, \quad (1a)$$

$$\text{hyperbolic tangent} : \sigma_2(\text{net}) = \tanh(\beta \text{net}) = \frac{e^{\beta \text{net}} - e^{-\beta \text{net}}}{e^{\beta \text{net}} + e^{-\beta \text{net}}}, \text{ or} \quad (1b)$$

$$\sigma_2(\text{net}) = \frac{2}{1 + e^{-2\beta \text{net}}} - 1,$$

where net denotes the input to a node and α , β are the slope parameters of the sigmoids. Hereafter, let us assume that $\alpha = \beta = 1$ and note that this assumption has no effect on the theoretical analysis presented in this paper. Nevertheless, for application purposes using different values for the slope parameters has an effect on the results obtained, as it will be explained later in Subsection 3.2. In addition to the previous assumptions, let us suppose that the nodes in the output layer may have either one of the above sigmoid activation functions or the linear one. Finally, we assume that all nodes in the hidden layer use the same activation function and the same stands for the output layer nodes.

With respect to the above assumptions the function computed by a feed forward neural network can be defined as $F : \mathbb{R}^n \times \mathbb{R}^{(n+1)h} \times \mathbb{R}^{(h+1)o} \rightarrow \mathbb{R}^o$, where $F(X, W_1, W_2)$ denotes the actual network output, $X \in \mathbb{R}^n$ refers to the input patterns, $W_1 \in \mathbb{R}^{(n+1)h}$ is the matrix of the input-to-hidden layer connection weights and biases, and $W_2 \in \mathbb{R}^{(h+1)o}$ denotes the matrix of the hidden-to-output layer connection weights and biases. Moreover, let us denote $T \in \mathbb{R}^o$ the target output of the function approximated by the network. Obviously, if any of the parameters, X, W_1, W_2 , and T is interval valued then the network is an interval neural network (Hu et al., 1998).

The process of neural networks training aims to solve the following optimization problem:

$$\text{arg min}_{W_1 \in \mathbb{R}^{(n+1)h}, W_2 \in \mathbb{R}^{(h+1)o}} \|F(X, W_1, W_2) - T\|_q \quad (2)$$

for some appropriate norm $\|\cdot\|_q$, considering that the ideal minimum of $\|F(X, W_1, W_2) - T\|_q$ is 0, which means that $F(X, W_1, W_2) = T$.

As said above, training the network is defining the values of the elements of the matrices W_1 and W_2 that solve the previous unconstrained minimization problem. The domain of these weights is not known in advance and so, in practice, the global optimization procedures used to solve this problem search for the global minimum in a box $\mathbb{D}_w \subset \mathbb{R}^{(n+1)h} \times \mathbb{R}^{(h+1)o}$. This means that the initial unconstrained global optimization problem is arbitrarily converted to an ad-hoc bound constrained optimization one. The bounds of this box and therefore its size are defined intuitively. Such a choice may lead to unsuccessful training or give a computationally expensive solution that is impractical to use in real life problems. The argument that the box \mathbb{D}_w is intuitively defined is supported by a number of examples found in the literature.

For instance, Sarcev (2012) gives an example of interval global optimization for training a network that approximates the function $z = 0.5 \sin(\pi x^2) \sin(2\pi y)$ and defines the initial

box to be $[w]_0 = [-10, 10]^s$, where s is the number of unknown weights. The results reported in the paper indicate that the global optimization technique succeeded in training the network using this initial box. However, the author does not provide any justification on how the initial search box was defined.

Another example is given by the approach formulated by Jamett and Acuña (2006) to solve the problem of weight initialization for an MLP based on interval analysis. These researchers argue that their approach “solves the network weight initialization problem performing an exhaustive global search for minimums in the weight space by means of interval arithmetic. The global minimum is obtained once the search has been limited to the estimated region of convergence.” For the experimental evaluation proposed in Jamett and Acuña (2006) the interval weights are initially defined as wide as necessary, with widths up to 10^6 .

In Hu et al. (1998) the authors consider a 3-layer interval neural network and they propose to adjust the interval weights with the use of the Newton interval method solving a system of nonlinear equations of the $h(n+o)$ unknown weights. This is possible under the hypothesis that the network is, what they call, a Type 1 interval neural network, i.e., a network for which the number l of nonlinear equations, formed using all available patterns x^k , $1 \leq k \leq l$, satisfy the inequality $l \leq h(n+o)$. When this inequality is not true they propose to use an interval branch-and-bound algorithm to solve the minimization problem related to the network training. However, the authors do not provide any kind of concrete experiment where interval global optimization is actually used for training the network.

In addition to the above, one should mention the problems related to the initialization and the bounds of the search region reported with heuristic and population based approaches when solving global optimization problems and more specifically when training MLPs. For instance, Helwig and Wanka (2008) showed that when PSO is used to solve problems with boundary constraints in high-dimensional search spaces, many particles leave the search space at the beginning of the optimization process. This commonly known problem of “particle explosion” (Clerc and Kennedy, 2002) for PSO in high-dimensional bounded spaces highlights the importance of defining bound handling procedures.

Moreover, when dealing with neural network training, one should take into account the remarks reported by Gudise and Venayagamoorthy (2003) regarding the selection of parameters for PSO. The authors report that during their experiments in order to determine the optimal PSO parameters they found that “A small search space range of $[-1, 1]$ did not provide enough freedom for the particles to explore the space and hence they failed to find the best position”. So, they gradually increased the size of the search space from $[-1, 1]$ to $[-200, 200]$ and they observed that a search space range of $[-100, 100]$ was the best range for having optimal performance of the PSO algorithm. Finally, when no limits were set on the search space range, the convergence rate of PSO decreased and there appeared to be even cases of no convergence.

According to the above examples, it is obvious that the outcome of training an MLP with global optimization is unpredictable as there is no information concerning the region where a global minimum of the error function is located. To the best of our knowledge this statement is true no matter the type of the global optimization procedure used. So, researchers and practitioners in the field proceed using random (intuitively defined) bounds which result in trial and error training strategies. This paper argues that it is possible to

derive the bounds of the search region for this type of global optimization task. It will be shown that the box defined by these bounds is guaranteed to enclose some global minimizers of the network output error function. This permits to obtain a value for the global minimum which is at least equal to the “machine ideal global minimum”, that is the best value for the global minimum that can be computed within the limits set by the machine precision. The results obtained in this paper apply to any type of global optimization procedure used to train an MLP.

3. Bounding the Weight Space Using Interval Analysis

The idea underlying our analysis is to consider the bounds of the range of each node in the output, the hidden and the input layer respectively. These bounds are implicit constraints imposed by the type of the activation function for the nodes in the output and the hidden layers. Moreover, explicit constraints are imposed by the values of the input patterns for the input layer. Performing local inversion of the activation function at the level of each node and using the constraints on its inputs imposed by the output of the previous layer results in formulating suitable linear interval equations. The subsequent analysis of the solution sets of these equations permits to derive the bounds of a box enclosing the region where some global minimizers of the network output error function are located. The approach proposed leads to defining a suitable approximation \mathbb{D}_w of the feasible set S for the error function of the network.

3.1 The Interval Analysis Formalism

Interval arithmetic has been defined as a means of setting lower and upper bounds on the effect produced on a computed quantity by different types of mathematical errors. The efficiency of methods built on interval arithmetic is related to these bounds, which need to be as narrow as possible. Thus, the major focus of interval analysis is to develop interval algorithms producing sharp solutions when computing numerical problems. A real interval, or interval number, is a closed interval $[a, b] \subset \mathbb{R}$ of all real numbers between (and including) the endpoints a and b , with $a \leq b$. The terms interval and interval number are used interchangeably. Whenever $a = b$ the interval is said to be degenerate, thin or even point interval. An interval $[a, b]$ where $a = -b$ is called a symmetric interval. Regarding notation, an interval x may be also denoted $[x]$, or $[\underline{x}, \bar{x}]$, or even $[x_L, x_U]$ where subscripts L and U stand for lower and upper bounds respectively. Notation for interval variables may be uppercase or lowercase (Alefeld and Mayer, 2000). Throughout this paper we will use the notation $[x]$ whenever we refer to an interval number, while notation $[\underline{x}, \bar{x}]$ will be used when explicit reference to the interval bounds is required. Moreover, an n -dimensional interval vector, that is a vector having n real interval components $([v_1], [v_2], \dots, [v_n])^T$, will be denoted $[v]$ or even $[\underline{v}, \bar{v}]$. Finally, uppercase letters will be used for matrices with interval elements as for an $n \times m$ interval matrix $[A] = ((a_{ij}))_{\substack{i=1,2,\dots,n \\ j=1,2,\dots,m}}$.

Generally, if $[z] = [\underline{z}, \bar{z}]$ is a real interval then the following notation is used:

$\text{rad}([x]) = (\bar{x} - \underline{x})/2$, is the radius of the interval $[x]$,

$\text{mid}([x]) = (\bar{x} + \underline{x})/2$, is the midpoint (mean value) of the interval $[x]$,

$|[x]| = \max\{|\underline{x}|, |\bar{x}|\}$, is the absolute value (magnitude) of the interval $[x]$,

$d([x]) = \bar{x} - \underline{x}$, is the diameter (width) of the interval $[x]$,

\mathbb{IR} , denotes the set of closed real intervals,

\mathbb{IR}^n , denotes the set of n -dimensional vectors of closed real intervals.

Let \diamond denote one of the elementary arithmetic operators $\{+, -, \times, \div\}$ for the simple arithmetic of real numbers x, y . If $[x] = [\underline{x}, \bar{x}]$ and $[y] = [\underline{y}, \bar{y}]$ denote real intervals then the four elementary arithmetic operations are defined by the rule

$$[x] \diamond [y] = \{x \diamond y \mid x \in [x], y \in [y]\}.$$

This definition guarantees that $x \diamond y \in [x] \diamond [y]$ for any arithmetic operator and any values x and y . In practical calculations each interval arithmetic operation is reduced to operations between real machine numbers. For the intervals $[x]$ and $[y]$ it can be shown that the above definition produces the following intervals for each arithmetic operation:

$$\begin{aligned} [x] + [y] &= [\underline{x} + \underline{y}, \bar{x} + \bar{y}], \\ [x] - [y] &= [\underline{x} - \bar{y}, \bar{x} - \underline{y}], \\ [x] \times [y] &= \{\min(\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y}), \max(\underline{x}\underline{y}, \underline{x}\bar{y}, \bar{x}\underline{y}, \bar{x}\bar{y})\}, \\ [x] \div [y] &= [x] \times \frac{1}{[y]}, \text{ with} \\ \frac{1}{[y]} &= \left[\frac{1}{\bar{y}}, \frac{1}{\underline{y}} \right], \text{ provided that } 0 \notin [\underline{y}, \bar{y}]. \end{aligned}$$

The usual algebraic laws of arithmetic operations applied to real numbers need to be re-considered regarding finite arithmetic on intervals. For instance, a non-degenerate (thick) interval has no inverse with respect to addition and multiplication. So, if $[x], [y]$, and $[z]$ are non-degenerate intervals then,

$$\begin{aligned} [x] + [y] &= [z] \not\Rightarrow [x] = [z] - [y], \\ [x] \times [y] &= [z] \not\Rightarrow [x] = [z] \times \frac{1}{[y]}. \end{aligned}$$

The following sub-distributive law holds for non-degenerate intervals $[x], [y]$, and $[z]$,

$$[x] \times ([y] + [z]) \subseteq [x] \times [y] + [x] \times [z].$$

Note that the usual distributive law holds in some particular cases: if $[x]$ is a point interval, if both $[y]$ and $[z]$ are point intervals, or if $[y]$ and $[z]$ lie on the same side of 0. Hereafter, the multiplication operator \times will be omitted as in usual algebraic expressions with

real numbers. Interval arithmetic operations are said to be *inclusion isotonic* or *inclusion monotonic* or even *inclusion monotone* given that the following relations hold,

$$\begin{aligned} \text{if } [a], [b], [c], [d] \in \mathbb{IR} \text{ and } [a] \subseteq [b], [c] \subseteq [d] \\ \text{then } [a] \diamond [c] \subseteq [b] \diamond [d], \text{ for } \diamond \in \{+, -, \times, \div\}. \end{aligned}$$

The property of *inclusion isotony* for interval arithmetic operations is considered to be the fundamental principle of interval analysis. More details on interval arithmetic and its extensions can be found in Alefeld and Mayer (2000); Hansen and Walster (2004), and Neumaier (1990).

Standard interval functions φ are extensions of the corresponding real functions $F = \{\sin(), \cos(), \tan(), \arctan(), \exp(), \ln(), \text{abs}(), \text{sqrt}(), \text{sqrt}()\}$ and they are defined via their range, i.e., $\varphi([x]) = \{\varphi(x) \mid x \in [x]\}$, for $\varphi \in F$, (Alefeld and Mayer, 2000). Given that, these real functions are continuous and piecewise monotone on any compact interval in their domain of definition, the values $\varphi([x])$ can be computed directly from the values at the bounds of $[x]$. For non-monotonic functions such as the trigonometric ones the computation is more complicated. For instance, $\sin(0, \pi) = [0, 1]$ which differs from the interval $[\sin(0), \sin(\pi)]$. In these cases, the computation is carried out using an algorithm (Jaulin et al., 2001). Finally, it must be noted that the standard interval functions are inclusion monotonic.

Let $f : D \subset \mathbb{R} \rightarrow \mathbb{R}$ be a real function and $[x] \subseteq D$ an interval in its domain of definition. The range of values of f over $[x]$ may be denoted by $R(f; [x])$ (Alefeld and Mayer, 2000) or simply $f([x])$. Computing the range $f([x])$ of a real function by interval analysis tools practically comes to enclosing the range $f([x])$ by an interval which is as narrow as possible. This is an important task in interval analysis which can be used for various reasons, such as localizing and enclosing global minimizers and global minima of f on $[x]$, verifying that $f([x]) \subseteq [y]$ for some given interval $[y]$, verifying the nonexistence of a zero of f in $[x]$ etc.

Enclosing the range of f over an interval $[x]$ is achieved by defining a suitable interval function $[f] : \mathbb{IR} \rightarrow \mathbb{IR}$ such that $\forall [x] \in \mathbb{IR}, f([x]) \subset [f]([x])$. This interval function is called an inclusion function of f . What is important when considering an inclusion function $[f]$ is that it permits to compute a box $[f]([x])$ which is guaranteed to contain $f([x])$, whatever the shape of $f([x])$ (Jaulin et al., 2001). Note that the so-called natural inclusion function is defined if $f(x), x \in D$ is computed as a finite composition of elementary arithmetic operators $\{+, -, \times, \div\}$ and standard functions $\varphi \in F$ as above. The natural inclusion function of f is obtained by replacing the real variable x by an interval variable $[x] \subseteq D$, each operator or function by its interval counterpart and evaluating the resulting interval expression using the rules in the previous paragraphs. The natural inclusion function has important properties such as being inclusion monotonic and if f involves only continuous operators and continuous standard functions it is convergent (see Jaulin et al., 2001).

3.2 Weights of Hidden-to-output Layer Connections

Let us compute the bounds of the intervals of the weights for any output node. As we will see the procedure described herein does not depend on the number of inputs to the node but merely on the type of its activation function.

The output z_k of the k -th node in the output layer is given by

$$z_k = \sigma \left(\sum_{j=1}^h w_{kj} y_j + w_{kb} \right), \quad (3)$$

where y_j is the output of the j -th node in the hidden layer. During training the output z_k may be any value in the range of the sigmoid activation function σ used by the output layer nodes. So, $z_k \in [0, 1]$ for the logistic sigmoid and $z_k \in [-1, 1]$ for the hyperbolic tangent.

The activation functions defined in (1) are bijective and so they are invertible. Using the inverse σ^{-1} of the activation function σ on both sides of (3) gives:

$$\sigma^{-1}(z_k) = \sigma^{-1} \left(\sigma \left(\sum_{j=1}^h w_{kj} y_j + w_{kb} \right) \right)$$

and finally the equation,

$$\sum_{j=1}^h w_{kj} y_j + w_{kb} = \sigma^{-1}(z_k). \quad (4)$$

The inverse of the activation functions are

$$\text{for the logistic sigmoid : } \sigma_1^{-1} = -\ln \frac{1-x}{x}, \text{ and}$$

$$\text{for the hyperbolic tangent : } \sigma_2^{-1} = \operatorname{atanh}(x) = \frac{1}{2} \ln \left(\frac{1+x}{1-x} \right).$$

The output of each activation function is an interval. So, the values of the inverse of the activation functions are,

$$\text{for the logistic sigmoid : } \sigma_1^{-1}([0, 1]) = [-\infty, +\infty],$$

$$\text{for the hyperbolic tangent : } \sigma_2^{-1}([-1, 1]) = [-\infty, +\infty].$$

For nodes with sigmoid activation functions when the output value is one of the exact bounds of the intervals $[0, 1]$ or $[-1, 1]$, then the corresponding node is saturated. This implies that connection weights take very large values. However, in practice this is rarely achieved and as Rumelhart et al. (1986) note: "The system cannot actually reach its extreme values of 1 or 0 without infinitely large weights. Therefore, in a practical learning situation in which the desired outputs are binary $\{0, 1\}$, the system can never actually achieve these values. Therefore, we typically use the values of 0.1 and 0.9 as the targets, even though we will talk as if values of $\{0, 1\}$ are sought." This consideration is adopted for the values of the output layer nodes in various practical pattern recognition applications of MLPs as well as in research papers such as Sprinkhuizen-Kuyper and Boers (1999); Yam and Chow (2001).

This means that a node with a sigmoid activation function becomes saturated with input values much greater than $-\infty$ and drastically smaller than $+\infty$, and its output is still considered to be 0 (or -1), and 1, respectively. Hence, as far as saturation of the node is concerned, for practical use instead of the interval $[-\infty, +\infty]$ one may consider a substitute $[-\mathbf{b}, \mathbf{b}]$, where $\mathbf{b} > 0$. In consequence, the node output is not any of the "ideal" intervals

$[0, 1]$ or $[-1, 1]$ but merely some approximation such as $[0 + \varepsilon, 1 - \varepsilon]$ or $[-1 + \varepsilon, 1 - \varepsilon]$ where ε depends on the value of \mathbf{b} and, obviously, it should satisfy the precision required for the problem.

Henceforth, the interval $[-\mathbf{b}, \mathbf{b}]$, for some $\mathbf{b} > 0$, will be used instead of the $[-\infty, +\infty]$ for the input values of the sigmoid activation function of any node in the output, or in the hidden, layer. With this assumption the intervals resulting for the inverse of the activation functions of the output layer nodes are:

$$\text{for the logistic sigmoid : } \sigma_1^{-1}([0 + \varepsilon, 1 - \varepsilon]) = [-\mathbf{b}_1, \mathbf{b}_1], \quad (5a)$$

$$\text{for the hyperbolic tangent : } \sigma_2^{-1}([-1 + \varepsilon, 1 - \varepsilon]) = [-\mathbf{b}_2, \mathbf{b}_2]. \quad (5b)$$

So, \mathbf{b}_1 and \mathbf{b}_2 are such that $\sigma_1([-1 + \varepsilon, 1 - \varepsilon]) = [0 + \varepsilon, 1 - \varepsilon] \triangleq [0, 1]_\varepsilon$ and $\sigma_2([-1 + \varepsilon, 1 - \varepsilon]) \triangleq [-1, 1]_\varepsilon$.

With these results (4) implies that

$$\sum_{j=1}^h w_{kj} y_j + w_{kb} \in [-\mathbf{b}, \mathbf{b}], \quad (6)$$

where $\mathbf{b} = \mathbf{b}_1$ or $\mathbf{b} = \mathbf{b}_2$, depending on the type of the activation function used. Recall that in Section 2 we assumed that the slope parameters α and β of the sigmoids are considered to be equal to 1. If this assumption does not hold then it is easy to see that the bounds \mathbf{b}_1 and \mathbf{b}_2 , defined here, need to be divided by α and β , respectively.

Taking into account the type of the activation function of the hidden layer nodes we have that $y_j \in [0, 1]$ for the logistic sigmoid and $y_j \in [-1, 1]$ for the hyperbolic tangent. However, we note that the above assumptions, regarding the approximations of the interval $[-\infty, +\infty]$, also hold for the sigmoid activation functions of the hidden layer nodes. So, in practical situations, we consider that $y_j \in [0, 1]_\varepsilon$ and $y_j \in [-1, 1]_\varepsilon$, where $[0, 1]_\varepsilon \subset [0, 1]$ and $[-1, 1]_\varepsilon \subset [-1, 1]$. Then, because of the inclusion monotonicity property of the interval arithmetic operators and the activation functions, in the analysis presented in this paper, for the interval equations we use the intervals $[0, 1]$ and $[-1, 1]$ instead of $[0, 1]_\varepsilon$ and $[-1, 1]_\varepsilon$, respectively. This means that any solution set or enclosure of a solution set determined, herein, for an interval equation using the wider intervals $[0, 1]$ and $[-1, 1]$ also constitute enclosures of the solution set of the same equation using the narrower ones, $[0, 1]_\varepsilon$ and $[-1, 1]_\varepsilon$.

Using all these results we may formulate the following two linear interval inequalities for the unknown weights of the hidden-to-output layer connections:

$$\sum_{j=1}^h [w_{kj}] [0, 1] + [w_{kb}] \leq [-\mathbf{b}, \mathbf{b}],$$

$$\sum_{j=1}^h [w_{kj}] [-1, 1] + [w_{kb}] \leq [-\mathbf{b}, \mathbf{b}].$$

Hence, the interval $[-\mathbf{b}, \mathbf{b}]$ determines an enclosure of the intervals $[w_{kj}]$, $1 \leq j \leq h$. However, as the aim of this paper is to bound the box containing the values of w_{kj} which solve

Equation (4), instead of solving the above inequalities we may, equivalently, consider the following equations and define approximations of the corresponding solution sets:

$$\sum_{j=1}^h [w_{kj}] [0, 1] + [w_{kh}] = [-b, b], \quad (7a)$$

$$\sum_{j=1}^h [w_{kj}] [-1, 1] + [w_{kh}] = [-b, b]. \quad (7b)$$

Before, examining solvability issues and the relation of the solutions to these equations with the problem at hand, let us discuss, hereafter, some issues concerning the parameter \mathbf{b} and its relation to ε .

3.2.1 ON THE DEFINITION OF THE PARAMETERS \mathbf{b} AND ε

The previous analysis introduced the interval $[-b, b]$ for the input values of a sigmoid activation function. This interval may also be defined for other types of activation functions and so its bounds depend on the type of the activation function. In the following sections we will show that these bounds, actually, define the volume of the area in the network weight space where global minimizers of the network's error function are located. The analysis presented in this paper does not rely on some specific values of the bounds of the interval $[-b, b]$. Instead, the results obtained depend on the activation functions of the network nodes and the calculation precision set for the problem. So, they have \mathbf{b} 's as parameters.

As noted above, for sigmoid activation functions, the interval $[-b, b]$ implicitly defines the intervals $[0, 1]_\varepsilon$ or $[-1, 1]_\varepsilon$ which approximate the intervals $[0, 1]$ or $[-1, 1]$, respectively. Conversely, by setting some specific value to ε and using the inverse of the activation functions, one is able to define a value for \mathbf{b} , and in consequence the width of the interval $[-b, b]$. As this interval determines enclosures for the intervals $[w_{kj}]$, $1 \leq j \leq h$, it is obvious that its width determines the magnitudes of the corresponding dimensions of the minimizers' search area. Here, this search area is considered to be a hyper-box in the weight space, determined by the intervals $[-b, b]$, for all network nodes. On the other hand, selecting some specific value for ε determines the numerical precision used by the network output for solving the problem at hand. Therefore, an interval $[-b, b]$ defines some ε inducing some numerical precision for the problem, and vice-versa. The smaller the area defined by small values of \mathbf{b} the bigger the values of ε and so the lower the precision. Conversely, the highest the precision for computing the network outputs the smaller the values of ε and so the larger the search area defined by larger \mathbf{b} 's.

This problem can be formulated in the following terms: 'How large can ε be in order to minimize \mathbf{b} and in consequence the search area of global minimizers without compromising the numerical precision for solving the problem'. Hence, some tradeoff between \mathbf{b} (volume of the search area) and ε (numerical precision) seems to be required here. The impact of selecting a value for \mathbf{b} , or equivalently for ε , on the numerical precision of the problem mainly concerns the accuracy required for the network output in order to match the problem requirements. In a classification context and for practical applications, this issue is probably insignificant given that, typically, for such problems an approximate response of the order of

$\varepsilon = 0.1$ is sufficient for the classification task. On the other hand, for function approximation problems where accuracy of the network output is really a prerequisite, the output nodes use linear activation functions as for these kind of problems networks perform regression tasks (Hornik et al., 1989; White, 1990). As we will see, hereafter, such approximation assumptions do not apply to linear activation functions. Moreover, for hidden nodes using sigmoid activation functions, as we will see, approximating the intervals $[0, 1]$ and $[-1, 1]$ by $[0, 1]_\varepsilon$ and $[-1, 1]_\varepsilon$, respectively, where ε is of the order of the machine precision, is sufficient for carrying out the neural computation. Actually, such an approximation is within the numerical precision adopted for neural computations in various contexts.

The problem of numerical precision pertaining neural computations has been addressed by several researchers. For instance, some of the earlier research include the work of Hoelzfeld and Fahlman (1992) who studied the possibility of training a network using the cascade-correlation algorithm in a limited numerical precision learning context. Moreover, Hoi and Hwang (1993) addressed the question of the precision required in order to implement neural network algorithms on low precision hardware. The authors performed a theoretical analysis of the error produced by finite precision computation and investigated the precision that is necessary in order to perform successful neural computations both for training a multilayer perceptron with back-propagation as well as for forward activation and retrieving. A more recent work (Draghici, 2002) deals with the capabilities of neural networks using weights with limited precision and the work of Vassiliadis et al. (2000) states that a representation error of 2^{-10} provides a limit for a safe approximation error in neural functions computations. Such an error is within the limits of machine precision and double precision arithmetic. The most recent research focuses on the numerical precision required for deep neural networks and their hardware implementation (Courbariaux et al., 2014; Gupta et al., 2015; Judd et al., 2016).

The precision level used for a neural network and the problem at hand is a matter that needs to be addressed prior to any implementation, either software or hardware. Following the above literature, in this paper, we may state that double precision floating point arithmetic proposed by the IEEE (754–2008) standard (Higham, 2002; IEEE, 2008) provides a very good precision level as required by modern neural computation applications and hardware implementations. Hence, for the experimental evaluation of the proposed approach the value of the machine epsilon, commonly known as double precision epsilon ($\varepsilon = 2.2204 \times 10^{-16}$) is retained in this paper. This value of ε is the precision (eps) used for the computing environment MATLAB (MATLAB-Documentation, 2014) as well as for the majority of the programming languages according to IEEE (754–2008) standard. As a consequence we have the following values for \mathbf{b} :

$$\text{for the logistic sigmoid : } \mathbf{b}_1 = 36.0437,$$

$$\text{for the hyperbolic tangent : } \mathbf{b}_2 = 18.3684.$$

Note that, under the specific assumptions of Rummelhart et al. (1986) the intervals resulting for the inverse of the activation functions of the output layer nodes are,

$$\text{for the logistic sigmoid : } \sigma_1^{-1}(0.1, 0.9) = [-2.1972, 2.1972],$$

$$\text{for the hyperbolic tangent : } \sigma_2^{-1}(-0.9, 0.9) = [-1.4722, 1.4722].$$

The analysis presented in this subsection relies on the assumption that the activation function of the output nodes is the logistic sigmoid or the hyperbolic tangent. However, when an MLP is used for function approximation or regression then usually linear activations are used in the output nodes. In order to establish a unique formalism for our analysis we need to define an interval of the type $[-\mathbf{b}, \mathbf{b}]$ for a pure linear activation function of the network's output nodes.

The sample of the input data, by default, defines an interval $[\underline{z}_k, \bar{z}_k]$ for the range of values of the k -th output node. A pure linear activation function f_{lin} is invertible in its domain. So, using inversion we obtain an interval of the type $[a_k, b_k] = f_{lin}^{-1}([\underline{z}_k, \bar{z}_k])$ for the input values of the linear activation function. Seemingly, such an interval is not symmetric and so, in order to keep pace with the previous assumptions, we extend it to obtain a symmetric interval of the type $[-\mathbf{b}, \mathbf{b}]$. This is achieved by setting $\mathbf{b}_{3,k} = \max\{a_k, |b_k|\}$ and considering the interval $[-\mathbf{b}_{3,k}, \mathbf{b}_{3,k}]$ instead of $[a_k, b_k]$ which actually permits to adopt the above assumptions and the ensuing conclusions. It is obvious that, during training it depends on the optimization procedure to narrow the search space back to the initial interval $[a_k, b_k]$.

Note that, if sampling of the input space is performed with a distribution which is close to the underlying distribution of the input space then the sample data are representative of the real data and hence the output of any node is reliably set by the values derived from the patterns. If sampling of the input space is not correct then the input data will provide erroneous dimensions of the search area. However, this unfortunate outcome will happen whatever the network or the training algorithm used.

The above considerations are necessary in order to tackle the hidden-to-output connection weights using Equations (7a) and (7b) as the unique formalism for the problem. Hereafter, whatever the activation function of the output nodes its range of values will be considered a symmetric interval of the form $[-\mathbf{b}, \mathbf{b}]$.

3.2.2 ON THE SOLUTION OF THE INTERVAL EQUATIONS

In interval analysis, the formal notation $[A][x] = [b]$ (or $[A]x = [b]$) of an interval linear system, where $[A] \in \mathbb{IR}^{m \times n}$ is an m -by- n matrix of real intervals and $[b] \in \mathbb{IR}^m$ is an m -dimensional vector of real intervals, denotes different interval problems (Shary, 1995). Respectively, the solution or solution set to this interval linear system has been defined in a number of possible ways (Shary, 1997). The paper uses the so-called *algebraic solution*, which is an interval vector $[x^e]$ such that, when substituting into $[A][x] = [b]$ and executing all interval arithmetic operations, results in the valid equality $[A][x^e] = [b]$. Later, in Section 4, we will show the implication of an *algebraic solution* in deriving an outer approximation of the solutions of Equations (7a) and (7b). For a theoretical justification of this choice see Shary (1996, 2002) and references therein. An algebraic solution for the bounding of the weight space is an interval vector of the form $[w_k^a] = ([w_{k1}^a], [w_{k2}^a], \dots, [w_{kb}^a], [w_{kb}^a])^\top$, such that each of the relations,

$$\begin{aligned} [w_{k1}^a][0, 1] + [w_{k2}^a][0, 1] + \dots + [w_{kb}^a][0, 1] + [w_{kb}^a][1, 1] &= [-\mathbf{b}, \mathbf{b}], \\ [w_{k1}^a][-1, 1] + [w_{k2}^a][-1, 1] + \dots + [w_{kb}^a][-1, 1] + [w_{kb}^a][1, 1] &= [-\mathbf{b}, \mathbf{b}], \end{aligned}$$

corresponding to (7a) and (7b), respectively, is a valid interval equality.

A suitable approximation of the feasible solution set to the training problem (2) can be

derived considering the bounds of the interval components of the algebraic solution to each of the Equations (7a) and (7b). In the following section we are defining the bounds of an algebraic solution to each of the aforementioned equations. This is accomplished by taking into account the specific type of these equations and considering the algebraic solutions (Shary, 1997) to these equations. Before proceeding to defining these algebraic solutions let us consider two issues; one concerning solvability of Equations (7a) and (7b) and the other dealing with effectively solving these equations.

Solvability of the above Equations (7a) and (7b) can be studied either considering each equation as an $1 \times (h + 1)$ system and using the results in Rohn (2003), or applying the conclusions of Ratschek and Sauer (1982). Using the latter in our case is more appropriate given that Ratschek and Sauer (1982) have considered the concept of the algebraic solution to an interval linear equation (see also Shary, 1996). Hence, the conclusions concerning the solvability of a single interval equation of the general form $\mathbf{A}_1 \mathbf{x}_1 + \mathbf{A}_2 \mathbf{x}_2 + \dots + \mathbf{A}_n \mathbf{x}_n = \mathbf{B}$, as denoted and studied in Ratschek and Sauer (1982), apply also in our case. The hypotheses of Ratschek and Sauer (1982, Theorem 1) are satisfied and so the above Equations (7a) and (7b) are solvable having eventually multiple solutions.

Solving a linear interval equation or defining an optimal enclosure of the solution set is a matter addressed by several researchers; see for example Beaumont (1998); Hansen (1969); Hansen and Sengupta (1981); Jansson (1997); Kreinovich et al. (1993); Neumaier (1986); Rohn (1989, 1995); Shary (1995) and extensive references therein. However, there is quite a little work done on solving an underdetermined linear interval equation. To the best of our knowledge, for this matter Neumaier (1986) refers to the work of Rump (1983) while the work of Popova (2006) proposes an algorithm which improves the approach used in Hölbig and Krämer (2003) and Krämer et al. (2006). In addition, INTLAB (Rump, 1999) provides an algorithm for solving underdetermined linear systems of interval equations which, however, does not seem to converge in all cases.

Finally, note that Ratschek and Sauer (1982) examined in detail the convexity of the solution set of the equation $\mathbf{A}_1 \mathbf{x}_1 + \mathbf{A}_2 \mathbf{x}_2 + \dots + \mathbf{A}_n \mathbf{x}_n = \mathbf{B}$ if this equation is solvable. According to Ratschek and Sauer (1982, Theorem 6) the solution set to each of the Equations (7a) and (7b) is not convex. However, the approach proposed in the next section results in defining an outer approximation which is a convex set.

3.3 Weights of Input-to-hidden Layer Connections

The output of the j -th node in the hidden layer used in the previous subsection is defined by

$$y_j = \sigma \left(\sum_{i=1}^n w_{ji} x_i + w_{j0} \right), \quad (8)$$

where x_i is the i -th input, i.e., the output of the i -th node in the input layer. Here, let us assume that scaling is applied (Bishop, 1995) to the neural network inputs x_i in order to facilitate training (LeCun, 1993), and so $x_i \in [0, 1]$ or $x_i \in [-1, 1]$. Moreover, the output y_j may be any value in the range of the sigmoid activation function σ used by the hidden layer nodes. So, $y_j \in [0, 1]$ for the logistic sigmoid and $y_j \in [-1, 1]$ for the hyperbolic tangent.

All these hypotheses suggest that defining the intervals of the weights of the input-to-hidden layer connections is a problem formulated exactly as the problem of defining the

weights of the hidden-to-output layer connections of the previous subsection. In consequence, we formulate the following equations:

$$\sum_{i=1}^n [w_{ji}] [0, 1] + [w_j b] = [-b, b], \quad (9a)$$

$$\sum_{i=1}^k [w_{ji}] [-1, 1] + [w_j b] = [-b, b]. \quad (9b)$$

The same arguments as those in the previous subsection apply concerning the solvability of Equations (9a) and (9b). In addition, solving these equations and defining outer approximations of the corresponding solution sets is covered by the same analysis as in the previous subsection.

4. An Algebraic Solution to Bounding the Weight Space

For the neural network training problem defined in (2), the solution set of each of the resulting Equations (7a), (7b) and (9a), (9b) is extremely difficult to be described analytically. Hence, in this section, we derive an outer approximation (an enclosure) of an algebraic solution, that is more practical to define and use. This enclosure is a convex polytope, depending on the MLP architecture, and more specifically on the type of the activation function of the network nodes, and is derived under the assumption that the neural network inputs x_i are scaled in the interval $[0, 1]$ or $[-1, 1]$. First, the theoretical results are derived and then the conditions for their applicability are discussed.

4.1 Theoretical Results

As seen in Subsection 3.2, above, an algebraic solution exists for each of the Equations (7a), (7b) and (9a), (9b). Here, given an algebraic solution $[w^a]$ we proceed with computing limit values for the bounds of the interval components of $[w^a]$. In the sequel, given these limit values we derive outer approximations of the solutions of Equations (7a), (7b) and (9a), (9b).

In order to simplify their representation, we define the outer approximations of the solutions as convex sets or convex polytopes. However, what really matters with these outer approximations for each of the equations is that, they enclose the solutions containing the connection weights which are the global minimizers of the problem defined by (2).

Theorem 1 Consider the equation,

$$[w_1] [0, 1] + [w_2] [0, 1] + \dots + [w_n] [0, 1] + [w_b] [1, 1] = [-b, b]. \quad (10)$$

Then, whenever an algebraic solution is regarded for this equation, the following statements are valid:

- (i) For any variable $[w_i] = [\underline{w}_i, \bar{w}_i]$, $1 \leq i \leq n$ the maximum possible value of \bar{w}_i is $2b$ or the minimum possible value of \underline{w}_i is $-2b$.
- (ii) For the variable $[w_b] = [\underline{w}_b, \bar{w}_b]$ the maximum possible value of \bar{w}_b is b and the minimum possible value of \underline{w}_b is $-b$.

Proof First, let us note that for the product $[x, y] [0, 1]$ the following equalities are valid depending on the values of x and y :

$$[x, y] [0, 1] = [0, y], \quad \text{if } 0 \leq x \leq y, \quad (11a)$$

$$[x, y] [0, 1] = [x, 0], \quad \text{if } x \leq y \leq 0, \quad (11b)$$

$$[x, y] [0, 1] = [x, y], \quad \text{if } x \leq 0 \leq y. \quad (11c)$$

In addition let $[t_j]$ denote the product $[w_j] [0, 1]$, for $1 \leq j \leq n$. Then, with respect to the above Equalities (11) the sum of all terms $[t_j]$ of the left hand side of Equation (10), that is, $[t_1] + [t_2] + \dots + [t_n]$ may be written as the sum of three intervals of the form $[0, s_k] + [-s_l, 0] + [-s_m^L, s_m^H]$ defined as follows:

$$[t_1] + [t_2] + \dots + [t_k] = [0, s_k], \quad 0 \leq k \leq n, \quad (12a)$$

$$[t_1] + [t_2] + \dots + [t_l] = [-s_l, 0], \quad 0 \leq l \leq n, \quad (12b)$$

$$[t_1] + [t_2] + \dots + [t_m] = [-s_m^L, s_m^H], \quad 0 \leq m \leq n, \quad (12c)$$

with $k + l + m = n$, $s_k \geq 0$, $s_l \geq 0$ and $s_m^L, s_m^H \geq 0$.

Statement (i)

Let i be an index such that for the interval $[\underline{w}_i, \bar{w}_i]$ we have $\bar{w}_i > 0$. We will show that $\bar{w}_i \leq 2b$. Excluding the term $[t_i]$ let us consider that the above integers k, l , and m are defined for the sum $[t_1] + [t_2] + \dots + [t_{i-1}] + [t_{i+1}] + \dots + [t_n]$. This means that, $0 \leq k \leq n-1$, $0 \leq l \leq n-1$, $0 \leq m \leq n-1$, and $k + l + m = n-1$. In addition, $[t_j] = [t_j, \bar{t}_j]$ with $\bar{t}_j = \bar{w}_j$ and

$$t_j = \begin{cases} 0 & \text{if } 0 \leq \underline{w}_j \\ \underline{w}_j & \text{if } \underline{w}_j < 0. \end{cases}$$

Then Equation (10) becomes,

$$[0, s_k] + [-s_l, 0] + [-s_m^L, s_m^H] + [t_i, \bar{w}_i] + [\underline{w}_b, \bar{w}_b] = [-b, b],$$

or even,

$$[-s_l - s_m^L - |t_i| + \underline{w}_b, s_k + s_m^H + \bar{w}_i + \bar{w}_b] = [-b, b].$$

This interval equation translates to:

$$s_k + s_m^H + \bar{w}_i + \bar{w}_b = b, \quad (13a)$$

$$-s_l - s_m^L - |t_i| + \underline{w}_b = -b. \quad (13b)$$

Furthermore, let us suppose that $\bar{w}_i > b$. Given that $s_k + s_m^H \geq 0$ then (13a) holds true if $\bar{w}_b < 0$ and $\bar{w}_b \leq b - (s_k + s_m^H + \bar{w}_i)$. Setting $\bar{w}_i = b + x$, where $x > 0$, then the previous inequality gives that $\bar{w}_b \leq b - (s_k + s_m^H + b + x)$, or else, $\bar{w}_b \leq -(s_k + s_m^H) - x < 0$.

On the other hand $[\underline{w}_b, \bar{w}_b]$ is a valid interval if $\underline{w}_b \leq \bar{w}_b$. It follows that $\underline{w}_b < 0$. Using (13b) we have that $\underline{w}_b = -b + s_l + s_m^L + |t_i|$. From this equation we deduce that the minimum possible value for \underline{w}_b is $-b$ attained when $s_l + s_m^L + |t_i| = 0$. In consequence we have $-b \leq \underline{w}_b \leq \bar{w}_b \leq -(s_k + s_m^H) - x$. So, $-b \leq -(s_k + s_m^H) - x$. The last inequality gives that $-b + (s_k + s_m^H) \leq -x$, and finally, $x \leq b - (s_k + s_m^H)$. This suggests that $x \leq b$ and the

maximum value is attained when $s_k + s_m^H = 0$. Hence, $\bar{w}_i \leq 2\mathbf{b}$. Note that $\bar{w}_i = 2\mathbf{b}$ when $[w_1] = [w_2] = \dots = [w_{i-1}] = [w_{i+1}] = \dots = [w_n] = [0]$, $\underline{w}_i = 0$, and $[w_b] = [-\mathbf{b}, -\mathbf{b}]$.

Now, suppose that $\underline{w}_i < -\mathbf{b}$ and $[w_i] = [\underline{w}_i, \bar{w}_i]$. Then following the same reasoning as above we may infer that $-2\mathbf{b} \leq \underline{w}_i$. In this case \underline{w}_b is a positive number that diminishes the excess on the lower bound of the left hand side interval of the Equation (10) produced by \underline{w}_i . Note that $\underline{w}_i = -2\mathbf{b}$ when $[w_1] = [w_2] = \dots = [w_{i-1}] = [w_{i+1}] = \dots = [w_n] = [0]$, $\bar{w}_i = 0$, and $[w_b] = [\mathbf{b}, \mathbf{b}]$.

Statement (ii)

In order to prove this statement let us suppose that $[\underline{w}_b, \bar{w}_b]$ is such that $\bar{w}_b > \mathbf{b}$. In this case $\bar{w}_b = \mathbf{b} + x$ with $x > 0$. Considering Equalities (12) we rewrite Equation (10) as,

$$[0, s_k] + [-s_l, 0] + [-s_m^L, s_m^H] + [\underline{w}_b, \bar{w}_b] = [-\mathbf{b}, \mathbf{b}],$$

which gives,

$$\begin{aligned} s_k + s_m^H + \bar{w}_b &= \mathbf{b}, \\ -s_l - s_m^L + \underline{w}_b &= -\mathbf{b}. \end{aligned}$$

The hypothesis $\bar{w}_b = \mathbf{b} + x$ with $x > 0$ implies that $s_k + s_m^H = \mathbf{b} - \bar{w}_b = -x$. This is impossible given that, by definition, $s_k + s_m^H \geq 0$. Hence, the only possibility for x is to have $x = 0$ which means that $\bar{w}_b \leq \mathbf{b}$. Using the same reasoning we may infer that $-\mathbf{b} \leq \underline{w}_b$. ■

The above theorem defines maximum and minimum values for the bounds of any interval parameter in Equations (7a) and (9a). Moreover, it suggests that in Equation (10) the interval $[w_b]$ acts as a term that cuts off the excess in the upper or the lower bound of the sum of the weight intervals. However, this is true only for either the upper or the lower bound but not for both bounds at the same time. The following Theorem 2 constitutes a generalization of *statement (i)* in the proof of Theorem 1.

Theorem 2 Suppose that the interval vector $([w_1^*], [w_2^*], \dots, [w_n^*], [w_b^*])^\top$ is an algebraic solution of Equation (10) of Theorem 1. If $S \subseteq I_n = \{1, 2, \dots, n\}$ denotes a set of indices $\{i_1, i_2, \dots, i_q\}$ such that all $[w_{i_j}^*] \neq 0$, $1 \leq j \leq q$, then only one of the following two relations can be true:

$$\begin{aligned} (i) \quad -2\mathbf{b} &\leq \sum_{j=1}^q \underline{w}_{i_j}^* < -\mathbf{b}, \text{ or} \\ (ii) \quad \mathbf{b} &< \sum_{j=1}^q \bar{w}_{i_j}^* \leq 2\mathbf{b}. \end{aligned}$$

■ **Proof** The proof is obtained using the same reasoning as for Theorem 1.

Corollary 3 Suppose that the interval vector $([w_1^*], [w_2^*], \dots, [w_n^*], [w_b^*])^\top$ is an algebraic solution of the equation $[w_1][0, 1] + [w_2][0, 1] + \dots + [w_n][0, 1] + [w_b][1, 1] = [-\mathbf{b}, \mathbf{b}]$. Then for any interval $[w_i^*]$ both the upper and the lower bounds of this interval cannot exceed the bounds of the interval $[-\mathbf{b}, \mathbf{b}]$.

Proof Suppose that there exists an index i such that for the interval $[\underline{w}_i^*, \bar{w}_i^*]$ both inequalities $\underline{w}_i^* < -\mathbf{b}$ and $\mathbf{b} < \bar{w}_i^*$ hold true. That is $[-\mathbf{b}, \mathbf{b}] < [\underline{w}_i^*, \bar{w}_i^*]$. Using the reasoning of Theorem 1 in order to reduce the excessive effect produced by $[\underline{w}_i^*, \bar{w}_i^*]$ on the left hand side of the interval of the equation we need to define $[\underline{w}_b, \bar{w}_b]$ with $\underline{w}_b > 0$ and $\bar{w}_b < 0$ which obviously is not a valid interval. Hence, it is impossible for some variable $[\underline{w}_i, \bar{w}_i]$ to have a solution such that $[-\mathbf{b}, \mathbf{b}] < [\underline{w}_i^*, \bar{w}_i^*]$. ■

Theorem 2 and Corollary 3, derived above, show the complexity of the set containing the algebraic solutions of Equations (7a) and (9a). However, the following Corollary defines an outer approximation of this solution set which is a convex set (polytope).

Corollary 4 The algebraic solutions of Equation (10) of Theorem 1 are enclosed in the polytope defined by $[-2\mathbf{b}, 2\mathbf{b}]^n \times [-\mathbf{b}, \mathbf{b}]$.

Proof Let the interval vector $([w_1^*], [w_2^*], \dots, [w_n^*], [w_b^*])^\top$ be a solution of Equation (10). Then for each $[w_j^*]$, $1 \leq j \leq n$ we have that $[w_j^*] < [-2\mathbf{b}, 2\mathbf{b}]$. Moreover $[w_b^*] < [-\mathbf{b}, \mathbf{b}]$. This proves the Corollary. ■

Theorem 5 Consider the equation,

$$[w_1][-1, 1] + [w_2][-1, 1] + \dots + [w_n][-1, 1] + [w_b][1, 1] = [-\mathbf{b}, \mathbf{b}]. \quad (14)$$

Then, whenever an algebraic solution is regarded for this equation, for any variable $[w_i] = [\underline{w}_i, \bar{w}_i]$, $1 \leq i \leq n$ the maximum possible value of \bar{w}_i is \mathbf{b} and the minimum possible value of \underline{w}_i is $-\mathbf{b}$. The same holds for the maximum possible value of \bar{w}_b and the minimum possible value of \underline{w}_b .

Proof First note that any interval multiplied by $[-1, 1]$ gives a symmetric interval. Actually, $[x, y][-1, 1] = [-m, m]$, where $m = \max\{|x|, |y|\}$, for any interval $[x, y]$. If $[t_j]$ denotes $[w_j][-1, 1]$, for $1 \leq j \leq n$, then Equation (14) becomes $[t_1] + [t_2] + \dots + [t_n] + [\underline{w}_b, \bar{w}_b] = [-\mathbf{b}, \mathbf{b}]$, with all $[t_j]$ being symmetric intervals. So, the following two equalities must hold:

$$\underline{t}_1 + \underline{t}_2 + \dots + \underline{t}_n + \underline{w}_b = -\mathbf{b}, \quad (15a)$$

$$\bar{t}_1 + \bar{t}_2 + \dots + \bar{t}_n + \bar{w}_b = \mathbf{b}, \quad (15b)$$

with $\underline{t}_j \leq 0$ and $0 \leq \bar{t}_j$ for $j = 1, 2, \dots, n$.

Let i be an index such that for the interval $[\underline{w}_i, \bar{w}_i]$ we have $\underline{w}_i < -\mathbf{b}$ and $\mathbf{b} < \bar{w}_i$. Thus, the term $[\underline{t}_i, \bar{t}_i] > [-\mathbf{b}, \mathbf{b}]$.

Then (15a) is true if $\underline{t}_1 + \underline{t}_2 + \dots + \underline{t}_{i-1} + \underline{t}_{i+1} + \dots + \underline{t}_n = 0$ and $\underline{w}_b > 0$. In addition (15b) is true if $\bar{t}_1 + \bar{t}_2 + \dots + \bar{t}_{i-1} + \bar{t}_{i+1} + \dots + \bar{t}_n = 0$ and $\bar{w}_b < 0$. The above suggest that $[\underline{w}_b, \bar{w}_b]$ defined so it is not a valid interval. In consequence the interval $[\underline{t}_i, \bar{t}_i] \leq [-\mathbf{b}, \mathbf{b}]$ and so neither $\mathbf{b} < \bar{w}_i$ nor $\underline{w}_i < -\mathbf{b}$. Hence, $[\underline{w}_i, \bar{w}_i] \leq [-\mathbf{b}, \mathbf{b}]$.

For the interval $[\underline{w}_b, \bar{w}_b]$ suppose that $\bar{w}_b < -\mathbf{b}$. This means that $\underline{w}_b = -\mathbf{b} - x$ for some $x > 0$. Under these assumptions (15a) holds true if $\underline{t}_1 + \underline{t}_2 + \dots + \underline{t}_n = x$, which is impossible given that $\underline{t}_1 + \underline{t}_2 + \dots + \underline{t}_n < 0$. The reasoning is the same if we suppose that $\bar{w}_b > \mathbf{b}$. Hence the minimum possible value of \underline{w}_b is $-\mathbf{b}$ and the maximum possible value of \bar{w}_b is \mathbf{b} . ■

The following corollary is a direct conclusion of Theorem 5.

Corollary 6 *The algebraic solutions of Equation (14) of Theorem 5 are enclosed in the hypercube $[-b, b]^{n+1}$.*

Proof Let the interval vector $([w_1^*], [w_2^*], \dots, [w_n^*], [w_b^*])^T$ be a solution of Equation (14). Then for each $[w_j^*]$, $1 \leq j \leq n$ we have that $[w_j^*] \subseteq [-b, b]$. The same stands for $[w_b^*]$. This proves the Corollary. ■

Theorem 5 and Corollary 6 help defining a convex outer approximation of the set of algebraic solutions to Equations (7b) and (9b).

4.2 Evaluation of the Theoretical Results

In the context of our minimization problem, solving the linear interval Equations (7a), (7b), (9a) and (9b) is equivalent to defining a solution in the tolerance solution set of each of these equations. This is achieved considering algebraic solutions for the following reasons. First, such an interval solution is proven to exist (Ratschek and Sauer, 1982) for each of these equations. Moreover, the algebraic solution of a linear interval system is a solution to the corresponding linear interval tolerance problem (Shary, 1996, Proposition 1). Finally, our aim is not to define exact bounds of the algebraic solution interval vector, but to identify the bounds of the algebraic solution.

Recall that Shary (2002) defined the tolerance (or tolerable) solution set of an interval equation $F([a], x) = [b]$ as the set formed by all point vectors $x \in \mathbb{R}^n$ such that for any $\tilde{a} \in [a]$ the image $F(\tilde{a}, x) \in [b]$. This can be written in one of the following two forms:

$$\begin{aligned} \sum_{\text{tol}} (F, [a], [b]) &= \{x \in \mathbb{R}^n \mid (\forall \tilde{a} \in [a]) (\exists \tilde{b} \in [b]) (F(\tilde{a}, x) = \tilde{b})\}, \\ \sum_{\subseteq} (F, [a], [b]) &= \{x \in \mathbb{R}^n \mid F([a], x) \subseteq [b]\}. \end{aligned}$$

The theoretical results of the previous subsection permit to define suitable convex polytopes containing the algebraic solutions of Equations (7a), (9a) and hypercubes for the algebraic solutions of Equations (7b), (9b). The algebraic solutions of the linear interval Equations (7a), (7b) and (9a), (9b), enclosed in their respective convex sets, are also solutions in the corresponding tolerance solution sets of these linear equations which may well be the activation functions of any output node. We are now interested in examining the relation of these convex sets with Equations (3) and (8). Let us consider the following non-linear interval equations:

$$\sigma_1 \left(\sum_{j=1}^h [w_{kj}] [y_j] + [w_{kb}] \right) = [0, 1], \quad (16a)$$

$$\sigma_2 \left(\sum_{j=1}^h [w_{kj}] [y_j] + [w_{kb}] \right) = [-1, 1], \quad (16b)$$

for any node k in the output layer using a sigmoid activation function, as well as the equations:

$$\sigma_1 \left(\sum_{i=1}^n [w_{ji}] [x_i] + [w_{jb}] \right) = [0, 1], \quad (17a)$$

$$\sigma_2 \left(\sum_{i=1}^n [w_{ji}] [x_i] + [w_{jb}] \right) = [-1, 1], \quad (17b)$$

defined for any node j in the hidden layer using a sigmoid activation function. Note that, $[y_j]$ denotes the interval of the output values of the j -th node in the hidden layer and $[x_i]$ denotes the interval of the values of the i -th component of the input patterns. The above equations are the interval counterparts of Equations (3) and (8).

We will show that the convex polytopes, derived in the previous subsection, constitute inner approximations of the tolerance solution sets for the interval Equations (16) and (17), namely the convex polytope for the Equations (16) and (17) having $[y_j] = [0, 1]$ and $[x_i] = [0, 1]$ while the hypercube does it, for the same equations, when $[y_j] = [-1, 1]$ and $[x_i] = [-1, 1]$.

First, let us formulate the following Proposition regarding Equations (16a), (16b) and (17a), (17b).

Proposition 7 *The tolerance solution set corresponding to each of the Equations (16a), (16b) and (17a), (17b) is unbounded.*

Proof Let us consider Equation (16a) and a real vector $w^1 \in \mathbb{R}^{h+1}$ such that

$$\sigma_1 \left(\sum_{j=1}^h w_j^1 [y_j] + w_{h+1}^1 \right) \subseteq [0, 1].$$

Then, there exists a real vector $w^2 \in \mathbb{R}^{h+1}$ for which $w^1 \leq w^2$, where the operator “ \leq ” denotes the component-wise operator “ \leq ”. If instead of the real vectors we consider their interval counterparts, that is, the degenerate interval vectors $[w^1]$ and $[w^2]$, it is obvious that $[w^1] \subseteq [w^2]$. Then, given that the sigmoid σ_1 is inclusion monotonic we have that

$$\sigma_1 \left(\sum_{j=1}^h w_j^1 [y_j] + w_{h+1}^1 \right) \subseteq \sigma_1 \left(\sum_{j=1}^h w_j^2 [y_j] + w_{h+1}^2 \right) \subseteq [0, 1].$$

Hence, for any real vector $[w^1]$ in the tolerance solution set there will always exist a “larger” vector $[w^2]$ for which the output interval produced will enclose the output interval targeted for $[w^1]$. So, the tolerance solution set of Equation (16a) is unbounded. The proof is similar for the other equations. ■

It is obvious that the outcome of this Proposition is due to the asymptotic convergence of each of the sigmoids to its extreme values, i.e., to 0, 1 or to $-1, 1$. Hence, if one aims in finding a suitable inner approximation of each tolerance solution set, then one should define an upper and a lower limit for the output of the corresponding sigmoid. Actually, this is

how we proceeded in Subsection 3.2. Subsequently, the direct consequence of this approach is to consider the solution set of some linear interval equation suitably defined to take into account the input values of a node. So, for each of the non-linear Equations (16a) and (16b) of some node in the output layer two linear interval equations, namely (7a) and (7b), are defined depending on the type of the sigmoid used. In addition, for each of the non-linear interval Equations (17a) and (17b) of any node in the hidden layer the two linear interval equations derived are (9a) and (9b).

For any node with a sigmoid activation function there are two possible types of boxes, as defined by Corollaries 4 and 6, depending on the intervals of the input values. Here, let us examine the range of application of these boxes in terms of their usage in (16a), (16b) and (17a), (17b). The convex polytope $[-2b, 2b]^h \times [-b, b]$ is used whenever the input signals are in the interval $[0, 1]$. Then, from (16a) and (16b) we derive the following relations:

$$\begin{aligned} \sigma_1 \left(\sum_{j=1}^h [-2b_1, 2b_1][0, 1] + [-b_1, b_1] \right) &\subseteq [0, 1], \\ \sigma_2 \left(\sum_{j=1}^h [-2b_2, 2b_2][0, 1] + [-b_2, b_2] \right) &\subseteq [-1, 1]. \end{aligned}$$

A direct conclusion of these relations is that the convex polytope $[-2b_1, 2b_1]^h \times [-b_1, b_1]$ constitutes an inner approximation of the tolerance solution set of the non-linear interval equation (16a). The same stands for the box $[-2b_2, 2b_2]^h \times [-b_2, b_2]$ and the non-linear interval equation (16b). Obviously, the hypercubes $[-b_1, b_1]^{(h+1)}$ and $[-b_2, b_2]^{(h+1)}$ are used instead of the corresponding convex polytopes when the input signals to a node are in the interval $[-1, 1]$. The same reasoning applies to Equations (17a) and (17b).

Besides these conclusions other interesting results are derived, hereafter, for the above convex polytopes. First, it is easy to verify that the following two equations hold true:

$$\begin{aligned} \sum_{j=1}^h [-2b_1, 2b_1][0, 1] + [-b_1, b_1] &= (2h+1)[-b_1, b_1], \\ \sum_{j=1}^h [-2b_2, 2b_2] [-1, 1] + [-b_2, b_2] &= (2h+1)[-b_2, b_2]. \end{aligned}$$

So, we have the relations:

$$\begin{aligned} \sigma_1((2h+1)[-b_1, b_1]) &\subseteq [0, 1], \\ \sigma_2((2h+1)[-b_2, b_2]) &\subseteq [-1, 1]. \end{aligned}$$

If we set $\lambda = \lambda(h)$, a small number, $0 < \lambda \ll 1$, depending on the number of inputs h of the node then we may write that

$$\begin{aligned} \sigma_1((2h+1)[-b_1, b_1]) &= [0 + \lambda, 1 - \lambda] = [0, 1]_\lambda \subseteq [0, 1], \text{ and} \\ \sigma_2((2h+1)[-b_2, b_2]) &= [-1 + \lambda, 1 - \lambda] = [-1, 1]_\lambda \subseteq [-1, 1]. \end{aligned}$$

In consequence, recalling the definition of the intervals $[-b_1, b_1]$ and $[-b_2, b_2]$, in Subsection 3.2, it is obvious that the following two relations hold true:

$$[0, 1]_\varepsilon = [0 + \varepsilon, 1 - \varepsilon] \subset [0 + \lambda, 1 - \lambda] = [0, 1]_\lambda, \quad (18a)$$

$$[-1, 1]_\varepsilon = [-1 + \varepsilon, 1 - \varepsilon] \subset [-1 + \lambda, 1 - \lambda] = [-1, 1]_\lambda. \quad (18b)$$

In consequence we may state that, for any node in the output layer using a sigmoid activation function, having input signals in the interval $[0, 1]$ and connection weights in the convex polytope $[-2b, 2b]^h \times [-b, b]$ then the range of output values are in the interval $[0, 1]_\lambda$ or $[-1, 1]_\lambda$ for some λ smaller than the precision ε used for the problem.

When the inputs to a node are in the interval $[-1, 1]$ then using the box $[-b, b]^{(h+1)}$ we arrive to similar results and formulate a similar statement. Finally, note that, using the same reasoning we get exactly the same results for any node in the hidden layer bringing out the validity of the theoretical results when used in (17a) and (17b) corresponding to Equations (9a) and (9b).

To further advance this reasoning suppose that the real matrices $W_i^s \in \mathbb{R}^{(n+1)h}$ and $W_2^s \in \mathbb{R}^{(h+1)o}$ constitute a global minimizer of the minimization problem (2), i.e., the distance $\|F(X, W_1^s, W_2^s) - T\|_q \leq \varepsilon$, where ε is the machine precision or the best precision set for the problem. Also, suppose that p patterns are available for the problem, which means that $X = \{x_1, x_2, \dots, x_p\}$ and the corresponding target outputs are $T = \{t_1, t_2, \dots, t_p\}$. The input patterns are n -dimensional vectors while the target outputs are o -dimensional. For any input pattern x_l , where $1 \leq l \leq p$, let z_l be the o -dimensional network output for this pattern, that is, $z_l = F(x_l, W_1^s, W_2^s)$.

Now, if the norm q is such that $1 \leq q$ and considering that the total network output error is averaged over all patterns then we can set the network output error for the l -th pattern to be $(\sum_{k=1}^o |z_{k,l} - t_{k,l}|^q)^{1/q} \leq \varepsilon$. From this, for the k -th output node we may consider that $|z_{k,l} - t_{k,l}|^q \leq \sum_{k=1}^o |z_{k,l} - t_{k,l}|^q \leq \varepsilon^q$. Hence, $|z_{k,l} - t_{k,l}|^q \leq \varepsilon^q$, which gives that $|z_{k,l} - t_{k,l}| \leq \varepsilon$. In consequence, $t_{k,l} - \varepsilon \leq z_{k,l} \leq t_{k,l} + \varepsilon$. So, if $t_{k,l} = 0$ or $t_{k,l} = -1$ then $z_{k,l} \in [0, 0 + \varepsilon]$ or $z_{k,l} \in [-1, -1 + \varepsilon]$. Similarly, if $t_{k,l} = 1$ then $z_{k,l} \in [1 - \varepsilon, 1]$. Hence, we may suppose that there exists some small positive constant, say λ , such that $0 + \lambda \leq z_{k,l} \leq 0 + \varepsilon$, or, $-1 + \lambda \leq z_{k,l} \leq -1 + \varepsilon$, and $1 - \varepsilon \leq z_{k,l} \leq 1 - \lambda$. Then, we may deduce that $[0 + \varepsilon, 1 - \varepsilon] \subset [0 + \lambda, 1 - \lambda]$ and $[-1 + \varepsilon, 1 - \varepsilon] \subset [-1 + \lambda, 1 - \lambda]$. This is exactly what has been derived above with relations (18). So, we deduce that the real matrices $W_1^s \in \mathbb{R}^{(n+1)h}$ and $W_2^s \in \mathbb{R}^{(h+1)o}$ supposed to constitute a global minimizer of the minimization problem (2) are, indeed, located in the corresponding convex polytope, as defined above. The reasoning is similar if q is the infinity norm.

Following the above discussion we consider that it is legitimate to argue that the convex polytopes identified by the proposed method are guaranteed, within the machine precision accuracy, to enclose some global minimizers of the network output error function.

4.3 Discussion on the Theoretical Results

The proposed approach operates under the assumption that the input data to a destination node are either in the interval $[0, 1]$ or in $[-1, 1]$. Typically, this happens in pattern recognition or classification applications; for example, this is the case for the interval used for coding the corresponding components of the input patterns if the weights concern input-to-

hidden layer connections. For a weight of a hidden-to-output layer connection this interval is the range of values of the sigmoid activation function of the corresponding hidden node. So, for an input data point in $[0, 1]$, the initial interval for the value of the corresponding weight is $[-2b, 2b]$; otherwise, if the value of the input data point is in $[-1, 1]$ then the initial interval for the corresponding weight is considered to be $[-b, b]$. In all cases, the interval for the value of any bias weight is $[-b, b]$.

Depending on the type of the activation function parameter b is denoted by b_1 for the logistic sigmoid, b_2 for the hyperbolic tangent, and b_3 for the pure linear activation function. Moreover, recall that the specific value in each case is implementation dependent being a function of the precision ε set for the problem. In this paper, we consider that ε is the machine epsilon corresponding to double precision and so the specific values for b should be $b = b_1 \geq 36.05$ and $b = b_2 \geq 18.4$, while the specific value for $b = b_3$ is defined by the problem data. Table 1 summarizes these results for a 3-layer perceptron having n , h and o nodes in the input, the hidden and the output layers respectively, and two possible ranges, $[0, 1]$ and $[-1, 1]$, for the input data coding.

The convex sets in Table 1 are defined according to the above theoretical results and more specifically the conclusions of Corollaries 4 and 6. However, if one adopts a ‘‘proud interpretation’’ of Corollary 3 it would be possible to consider narrower intervals for nodes with inputs in the interval $[0, 1]$. So, instead of the interval $[-2b, 2b]$ one would rather consider the interval $[-b, b]$. Obviously with this argument the convex set defined, in this case, is no longer a polytope, in general terms, but rather a hypercube. However, we should note that this result is based on heuristic considerations, as the complexity of the structure of the solution sets of Equations (7a) and (9a) leaves no space for such a theoretical conclusion. The validity of this heuristic conjecture is tested in the following Section 5.

The values defined, previously, for b and the subsequent volumes of the convex polytopes, though formally proven, are relatively large. So, they may be not convincing regarding the advantage offered to the global optimization based training by the proposed approach. Here, let us recall the assumption that, often, in practical applications, the output values of a node are expected to be in the interval $[0.1, 0.9]$ for the sigmoid activation function or in the interval $[-0.9, 0.9]$ for the hyperbolic tangent. Then according to Paragraph 3.2.1 the corresponding values for b , denoted here by b^* , should be $b^* \geq 2.2 \approx 2.1972$ and $b^* \geq 1.5 \approx 1.4722$. Despite the apparent disagreement of the intervals $[0.1, 0.9]$ and $[-0.9, 0.9]$ with the linear inverse equations (10) and (14), respectively, the previous theoretical results are still valid. As an example, let us compute the bounds of the weights for the k -th output node having a sigmoid activation function with output in the interval $[0.1, 0.9]$ and inputs y_j in the interval $[0.1, 0.9]$ supposing that any node in the hidden layer has the logistic sigmoid activation function, as well. With these hypotheses Equation (7a) becomes

$$\sum_{j=1}^h [w_{kj}] [0.1, 0.9] + [w_{k0}] = [-b^*, b^*], \quad (19)$$

where b^* is the specific b as noted previously. In addition, due to inclusion monotonicity $\sum_{j=1}^h [w_{kj}] [0.1, 0.9] + [w_{k0}] \subseteq \sum_{j=1}^h [w_{kj}] [0, 1] + [w_{k0}]$. On the other hand, at the expense of lower precision, we may have the equation $\sum_{j=1}^h [w_{kj}] [0, 1] + [w_{k0}] = [-b^*, b^*]$. Then, the solution set defined for this specific form of Equation (7a) is a tolerance solution set of

the linear interval equation (19). In consequence, the convex polytope $[-2b, 2b]^h \times [-b, b]$, for any $b = b^* \geq 2.2$, is a tolerance solution set of the linear interval equation (19), and constitutes an inner approximation of the tolerance solution set of the non-linear interval equation

$$\sigma_1 \left(\sum_{j=1}^h [w_{kj}] [y_j] + [w_{k0}] \right) = [0.1, 0.9], \quad (20)$$

where $[y_j] = [0.1, 0.9]$ for $1 \leq j \leq h$. It is obvious that this reasoning applies to any linear interval equation, such as (7a), (7b), (9a) and (9b).

If the scaling assumption of the neural network inputs x_i does not hold (this may happen with function approximation or regression problems), it is still possible to solve the problem; however, the solution does not comply with the algebraic formalism adopted in the previous section. In this case, let us suppose that $x_i \in [x_i^-, x_i^+]$, for $i = 1, 2, \dots, n$ where the value of the functional $\chi([x_i^-, x_i^+]) \in [-1, 1]$ (see Ratschek and Sauer, 1982), which means that the bounds of the interval $[x_i^-, x_i^+]$ may have any value provided that $x_i^- \leq x_i^+$. For the bias we have that the input is $\chi(1, 1) = 1$, so the hypotheses of Ratschek and Sauer (1982, Theorem 1) are satisfied and in consequence even in this case the Equations (9a) and (9b) have a solution. However, it is clear that the resulting equation does not comply with the formalism adopted in the previous subsection for the Equations (7a), (7b) and (9a), (9b). Although a detailed coverage of this issue is considered out-of-the-scope of this paper, it can briefly be stated that this situation can be dealt with by considering the work of Popova (2006) and the algorithm proposed therein.

5. Testing

In this section we give some concrete examples of the approach presented above. Our objective is to perform some tests, using well known benchmarks, in order to show that the convex sets, derived following the analysis of this paper, indeed enclose global minimizers of the cost function associated with the output of the MLP used for each problem. To this end, the interval global optimization software (GOP), provided by Pál and Csendes (2009), is used.

The reason for adopting an interval global optimization procedure instead of some version of PSO or a GA-based approach is easily understood if one takes into account the fact that interval global optimization is a deterministic approach which is guaranteed to locate the best available minimizers and the interval for the corresponding global minimum. On the other hand, convergence capabilities of population based and heuristic approaches depend on a number of heuristically defined random parameters. Hence, a proof of convergence for these methods is either probabilistic (Bertsimas and Tsitsiklis, 1993; Elben et al., 1991), or it relies on empirical results (Pedersen, 2010). This means that, by using a global search method of this class, it is practically impossible to verify that the boxes derived by the proposed approach indeed contain the best available minimizers.

Table 1: Approximation of the initial weight set

Activation function	Input Data Interval	
	[0,1]	[-1,1]
HL: logistic sigmoid	$[-2b_1, 2b_1]^{*sh} \times [-b_1, b_1]^h$	$[-b_1, b_1]^{(n+1)*h}$
OL: logistic sigmoid	$[-2b_1, 2b_1]^{*so} \times [-b_1, b_1]^o$	$[-b_1, b_1]^{(h+1)*o}$
HL: logistic sigmoid	$[-2b_1, 2b_1]^{*sh} \times [-b_1, b_1]^h$	$[-b_1, b_1]^{(n+1)*h}$
OL: hyperbolic tangent	$[-2b_2, 2b_2]^{*so} \times [-b_2, b_2]^o$	$[-b_2, b_2]^{(h+1)*o}$
HL: logistic sigmoid	$[-2b_1, 2b_1]^{*sh} \times [-b_1, b_1]^h$	$[-b_1, b_1]^{(n+1)*h}$
OL: linear	$[-2b_3, 2b_3]^{*so} \times [-b_3, b_3]^o$	$[-b_3, b_3]^{(h+1)*o}$
HL: hyperbolic tangent	$[-2b_2, 2b_2]^{*sh} \times [-b_2, b_2]^h$	$[-b_2, b_2]^{(n+1)*h}$
OL: logistic sigmoid	$[-2b_1, 2b_1]^{*so} \times [-b_1, b_1]^o$	$[-b_1, b_1]^{(h+1)*o}$
HL: hyperbolic tangent	$[-2b_2, 2b_2]^{*sh} \times [-b_2, b_2]^h$	$[-b_2, b_2]^{(n+1)*h}$
OL: hyperbolic tangent	$[-2b_2, 2b_2]^{*so} \times [-b_2, b_2]^o$	$[-b_2, b_2]^{(h+1)*o}$
HL: hyperbolic tangent	$[-2b_2, 2b_2]^{*sh} \times [-b_2, b_2]^h$	$[-b_2, b_2]^{(n+1)*h}$
OL: linear	$[-2b_3, 2b_3]^{*so} \times [-b_3, b_3]^o$	$[-b_3, b_3]^{(h+1)*o}$

- HL : Hidden Layer

- OL : Output Layer

- n, h and o are the number of nodes in the input, hidden and output layer respectively

- b_1 denotes the value of \mathbf{b} in the case of the logistic sigmoid activation function

- b_2 denotes the value of \mathbf{b} in the case of the hyperbolic tangent activation function

- b_3 denotes the value of \mathbf{b} in the case of the linear activation function

5.1 Experimental Setup

In order to perform this experimental verification, we follow the next three steps:

1. Consider an MLP which is typically used for solving the corresponding problem.
2. Write the MATLAB code implementing the natural inclusion function of the MLP mapping, (an example is given in Appendix A).
3. Use the GOP procedure to solve the minimization problem defined for this natural inclusion function, where the bounds for the weights are set by the convex polytope computed by the proposed approach.

Typically, GOP is used for solving a global optimization problem with bound constraints and performs an exhaustive search of the initially defined set of boxes. The criterion used to stop the search process is a tolerance level of the width of the interval computed as the value of the inclusion function. So, given that in this paper we are interested only in verifying the existence of global minimizers and not in effectively locating a global minimum, as this happens when training the MLP, we specify a large value for the tolerance level of the width of the inclusion function interval. Hence, we use GOP to obtain rough estimates of the interval enclosing the global minimum of the network's output error function.

Moreover, GOP uses gradient information of the objective function to perform branch-and-bound computations. In order to comply with this requirement we use the Mean-

Squared-Error (MSE) for computing the distance between the actual network output and the target output. However, using a differentiable distance function such as MSE is not restrictive. Actually, the analysis provided throughout Section 4 makes no assumption regarding the type of the cost function of the network output. What really matters is that the set of boxes defined by the proposed method surely encloses weight vectors that are global minimizers, which means that the range of the error function computed on this set of boxes is an interval enclosing its global minimum.

Note that for all examples the boxes are computed using for \mathbf{b}_1 and \mathbf{b}_2 the values proposed in Paragraph 3.2.1. Obviously, one may set these values according to the precision required for the problem at hand. However, in order to comply with the theoretical considerations of the paper the values chosen for \mathbf{b}_1 and \mathbf{b}_2 should be such that the corresponding sigmoid is driven to the saturation region.

5.2 Experiments and Results

5.2.1 A SIMPLE PROBLEM

The Single Neuron problem: Here, we consider the simplest possible case consisting of a single neuron having one input, a bias and using the logistic sigmoid activation. The advantage of proposing such a rudimentary example is the facility it offers in visualizing its error function. In the following figures one may observe the form of the error functions for two simple artificial mappings with weights in the intervals $[-5.0, 5.0]$ and $[-5.0, 5.0]$. Using GOP it is easy to locate the global minima for each of these functions for various

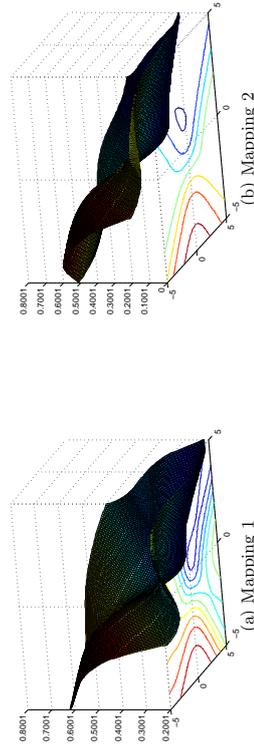


Figure 1: A simple neuron and the error functions for two artificial mappings

boxes of the weights. The results obtained for various trials are given hereafter.

Initial box for the weights: $[-72.1, 72.1][[-36.05, 36.05]$, tolerance = 0.001. Intervals computed by GOP:

Function name: SINGLE NEURON MAPPING 1

The set of global minimizers is located in the union of the following boxes:

c1: [21.75680336914063, 37.45827910156252][20.27820312500000, 36.05010000000000]

c2: [21.54557275390626, 21.68639316406251][19.99656210937500, 20.27820312500000]

c3: [21.12311152343751, 21.40475234375001][19.71492109375000, 19.99656210937500]

The global minimum is enclosed in:
 $[0.176989707623682269, 0.177214552453163560]$
 Function name: SINGLE NEURON MAPPING 2
 The set of global minimizers is located in the union of the following boxes:
 c1: $[-1.44335920410156, -1.05610307617187][1.09140893554687, 1.40825507812500]$
 The global minimum is enclosed in:
 $[0.063025775813289955, 0.093289721540457949]$

In these two cases, it is obvious that the location of the minimizers greatly differs from one error function to the other. Moreover, in the case of the first error function the global minimum is defined exclusively for positive values of the weights. This shows that the input data greatly affect the location of the global minimum.

Let us now consider a narrower interval. Initial box for the weights: $[-5.0, 5.0][-5.0, 5.0]$, tolerance = 0.01. Intervals computed by GOP:

Function name: SINGLE NEURON MAPPING 1
 The set of global minimizers is located in the union of the following boxes:
 c1: $[4.68750000000000, 5.00000000000000][4.21875000000000, 4.68750000000000]$
 The global minimum is enclosed in:
 $[0.215127653743816594, 0.219438417879982417]$
 Function name: SINGLE NEURON MAPPING 2
 The set of global minimizers is located in the union of the following boxes:
 c1: $[-2.03125000000000, -0.62500000000000][0.85837500000000, 1.87500000000000]$
 c2: $[-2.34375000000000, -2.18750000000000][2.03125000000000, 2.18750000000000]$
 The global minimum is enclosed in:
 $[0.089012518962891946, 0.093326904683555062]$

In this trial, narrower intervals were used for the bounds of the weights. As far as the first mapping is concerned, it's easy to notice that, despite the larger value for the tolerance, the intervals used do not enclose the previously detected global minimum. For the intervals specified the so-called global minimum is a local one, despite being the best that GOP was able to locate. For the second mapping, the intervals estimated for both the location of the global minimizers as well as for the global minimum are larger, due to the higher tolerance value, and enclose the intervals computed by GOP in the previous trial.

The main outcome of these examples is that they are consistent with the analysis provided in the paper. These trials provide "good" examples of boxes with global minimizers, as well as "counter-examples" of boxes without global minimizers. The results confirm that heuristic arguments, concerning the weight intervals, are inefficient, whenever global optimization based training is considered for a neural network. Moreover, the convex set defined by the proposed approach provides reliable estimation of these weight intervals.

For bound constrained problems the global optimizer GOP detects the best available minimizers within the box defined for the error function and hence the interval estimate of the corresponding global minimum. Given the size of the search space, in the following experiments it has not been possible to perform exhaustive search of the global minimum and so we tried to obtain rough estimations of the interval enclosing it.

5.2.2 PROBLEMS – NETWORKS WITH LOGISTIC SIGMOID ACTIVATION FUNCTIONS

The XOR problem: Let us consider a 2-2-1 network used for the XOR problem. The purpose of this experiment is to allow us to verify the results of the proposed approach against previous literature (see Haney, 1995, 1998; Sprinkhuizen-Kuyper and Boers, 1999), which has investigated analytically the local and global minima in the XOR problem. All network nodes use the logistic sigmoid activation function, inputs to all nodes are in the interval $[0, 1]$ and the desired outputs are the bounds of the interval $[0, 1]$. The network is fully connected and so there are $2 * 2 + 2 * 1 = 6$ unknown weights as well as $2 + 1 = 3$ bias weights. The interval equation of the weights to any node in the network is one of (7a) or (9a). Thus, if we set $\mathbf{b} = \mathbf{b}_1 = 36.05$ we obtain,

$$[w_1][0, 1] + [w_2][0, 1] + \dots + [w_n][0, 1] + [w_b][1, 1] = [-36.05, 36.05]$$

and so, the initial box where global minimizers are expected to be located is

$$\underbrace{[-72.1, 72.1]^{2*2}}_{\text{Hidden layer}} \times \underbrace{[-36.05, 36.05]^2}_{\text{Output layer}} \times \underbrace{[-72.1, 72.1]^{2*1}}_{\text{Output layer}} \times \underbrace{[-36.05, 36.05]^1}_{\text{Hidden layer}}.$$

The output provided by GOP for the above network is:

Function name: XOR
 The set of global minimizers is located in the union of the following boxes:
 c1: $[-72.09999999999999, 72.09999999999999] \dots$
 $[-36.05000000000000, 36.05000000000000]$
 The global minimum is enclosed in:
 $[0.0000000000000000, 0.250000000000000000]$.

The results of this experiment confirm the theoretical conclusions regarding the location of the global minimizers.

In addition to the previous experiment, it is tentative to examine, here, if the narrower box, defined using a loose interpretation of Corollary 3, is likely to enclose the global minimizers of the network output error. So, if the box $[-36.05, 36.05]^{2*2} \times [-36.05, 36.05]^2$ \times $\underbrace{[-36.05, 36.05]^{2*1}}_{\text{Output layer}} \times \underbrace{[-36.05, 36.05]^1}_{\text{Hidden layer}} = [-36.05, 36.05]^9$ is used as the initial box then the output provided by GOP is:

Function name: XOR
 The set of global minimizers is located in the union of the following boxes:
 c1: $[-36.05000000000000, 36.05000000000000] \dots$
 $[-36.05000000000000, 36.05000000000000]$
 The global minimum is enclosed in:
 $[0.0000000000000000, 0.250000000000000000]$.

The result of this last experiment shows that a narrower box can be effectively used for searching the global minimizers. Nevertheless, we should re-iterate that, as mentioned in

Subsection 4.3, using such a narrower box lacks the necessary theoretical justification.

Finally, if we make the assumption that the outputs of the sigmoids are in the interval $[0.1, 0.9]$ the initial box is considered to be $[-4.4, 4.4]^6 \times [-2.2, 2.2]^3$ which, according to Table 1, is $[-4.4, 4.4]^{2+2} \times [-2.2, 2.2]^2 \times [-4.4, 4.4]^{2+1} \times [-2.2, 2.2]^1$. Then the output provided by GOP is:

Output layer

Function name: XOR

The set of global minimizers is located in the union of the following boxes:

c1: $[-4.4000000000000, 4.4000000000000] \dots$

$[-2.2000000000000, 2.2000000000000]$

The global minimum is enclosed in:

$[0.0000000000000000, 0.160000000000000031]$.

Overall, the outcomes of the three experiments are in line with previous research (Hamey, 1995, 1998; Sprinkhuizen-Kuyper and Boers, 1999). Actually, Hamey (1998) applied to the XOR problem a new methodology for the analysis of the error surface and showed that starting from any point with finite weight values, there exists a finite non-ascending trajectory to a point with error equal to zero. Moreover, Sprinkhuizen-Kuyper and Boers (1999) proved that “the error surface of the two-layer XOR network with two hidden units has a number of regions with local minima” and concluded that “from each finite point in weight space, a strictly decreasing path exists to a point with error zero”. The conclusions of these papers resulted following an exhaustive analysis of the error surface of the network output. Finally, both papers conclude that for a 2-2-1 network using finite weights a global minimum is reachable in the error surface of the XOR problem. This conclusion constitutes a qualitative validation of the results obtained in this paper.

Finally, let us consider the case where a set of boxes does not contain a global minimizer. An example of such a set of boxes is reported by GOP hereafter.

Function name: XOR

The set of global minimizers is located in the union of the following boxes:

c1:

$[-72.09999999999999, 102.09999999999999][[-72.09999999999999, 102.09999999999999]$

$[-72.09999999999999, 102.09999999999999][[-72.09999999999999, 102.09999999999999]$

$[36.05000000000000, 136.05000000000000][36.05000000000000, 136.05000000000000]$

$[72.09999999999999, 102.09999999999999][72.09999999999999, 102.09999999999999]$

$[36.05000000000000, 136.05000000000000]$

The global minimum is enclosed in:

$[0.49882522561647045, 0.5000000000000000111]$.

The IRIS classification problem: Let us consider a 4-5-3 network used for the IRIS classification problem. This benchmark is known as Fisher’s IRIS problem. Based on the values of sepal length and width, petal length and width, the class of IRIS plant needs to be predicted. Inputs are scaled in the interval $[0, 1]$, all network nodes use the logistic sigmoid activation function and the desired outputs are binary $\{0, 1\}$. The network is fully connected and so there are $4 * 5 + 5 * 3 = 35$ unknown weights along with $5 + 3 = 8$ bias

weights. The interval equation of the weights to any node in the network is one of (7a) or (9a) and so if we set $\mathbf{b} = \mathbf{b}_1 = 36.05$ we have again,

$$[w_1][0, 1] + [w_2][0, 1] + \dots + [w_n][0, 1] + [w_b][1, 1] = [-36.05, 36.05].$$

So, one may consider the following initial box for searching the global minimizers,

$$\underbrace{[-72.1, 72.1]^{4*5} \times [-36.05, 36.05]^5}_{\text{Hidden layer}} \times \underbrace{[-72.1, 72.1]^{5*3} \times [-36.05, 36.05]^3}_{\text{Output layer}}.$$

The output provided by GOP for the above network is:

Function name: IRIS

The set of global minimizers is located in the union of the following boxes:

c1: $[-72.09999999999999, 72.09999999999999] \dots$

$[-36.050000000000000, 36.050000000000000]$

The global minimum is enclosed in:

$[0.0000000000000000, 0.75000000000000000]$.

Moreover, under the assumption that the output of the sigmoids are in the interval $[0.1, 0.9]$ the initial box is considered to be $[-4.4, 4.4]^{35} \times [-2.2, 2.2]^8$ which, according to Table 1, is $[-4.4, 4.4]^{4+5} \times [-2.2, 2.2]^5 \times [-4.4, 4.4]^{5+3} \times [-2.2, 2.2]^3$. Then the output provided by GOP is:

Hidden layer

Output layer

Function name: IRIS

The set of global minimizers is located in the union of the following boxes:

c1: $[-4.40000000000000, 4.40000000000000] \dots$

$[-2.20000000000000, 2.20000000000000]$

The global minimum is enclosed in:

$[0.0000000000000000, 0.48000000000000002480]$.

The results of these IRIS experiments are encouraging. Again, the results of the first experiment roughly confirm the theoretical conclusions regarding the location of the global minimizers, while the other experiment indicates that narrower boxes can be effectively used for searching the global minimizers, eventually providing a narrower interval estimate for the global minimum.

5.2.3 A PROBLEM – NETWORK WITH HYPERBOLIC TANGENT ACTIVATION FUNCTIONS

The British vowels recognition problem: Consider the 3-layer 10-20-11 network used for the British vowels data recognition benchmark. This benchmark, known as Deterding Data (Deterding, 1989), is a speaker independent recognition problem of the eleven steady state vowels of British English using a specified training set of 10 lpc (linear prediction coefficients) derived log area ratios. Inputs for this problem are scaled in the interval $[-1, 1]$ and all network nodes use the hyperbolic tangent sigmoid activation function. The network is fully connected and so there are $10 * 20 + 20 * 11 = 420$ unknown weights as well as $20 + 11 = 31$ bias weights. In this case the interval equation of the weights to any node

in the network is one of (7b) or (9b). Setting $\mathbf{b} = \mathbf{b}_2 = 18.04$ one obtains the following equation,

$$[w_1][-1, 1] + [w_2][-1, 1] + \dots + [w_n][-1, 1] + [w_b][1, 1] = [-18.4, 18.4].$$

So, the initial box where global minimizers are potentially located is the hypercube

$$\underbrace{[-18.4, 18.4]^{(10+1)*20}}_{\text{Hidden layer}} \times \underbrace{[-18.4, 18.4]^{(20+1)*11}}_{\text{Output layer}} = [-18.4, 18.4]^{451}.$$

The output provided by GOP for the above network is:

Function name: BRITISH VOWELS

The set of global minimizers is located in the union of the following boxes:

c1: [-18.400000000000000, 18.400000000000000]...

[-18.400000000000000, 18.400000000000000]

The global minimum is enclosed in:
[0.00000000000000000, 11.0000000000000000].

Under the assumption that the output of the sigmoids are in the interval $[-0.9, 0.9]$ the initial box is expected to be $\underbrace{[-1.5, 1.5]^{(10+1)*20}}_{\text{Hidden layer}} \times \underbrace{[-1.5, 1.5]^{(20+1)*11}}_{\text{Output layer}} = [-1.5, 1.5]^{451}$. The

output provided by GOP for the above network is:

Function name: BRITISH VOWELS

The set of global minimizers is located in the union of the following boxes:

c1: [-1.500000000000000, 1.500000000000000]...

[-1.500000000000000, 1.500000000000000]

The global minimum is enclosed in:
[0.00000000000000000, 8.910000000000142251].

The results of the first experiment confirm the theoretical conclusions regarding the location of the global minimizers. In addition, the second experiment confirms the hypothesis that the narrower box derived using the interval $[-0.9, 0.9]$ for the range of the hyperbolic activation functions, can be effectively used for searching the global minimizers of the network learning error.

5.2.4 A PROBLEM – NETWORK WITH HYPERBOLIC TANGENT AND A PURE LINEAR ACTIVATION FUNCTION

A sinusoidal function approximation problem: Consider the 3-layer 2-21-1 network used for approximating the function $y = 0.5 \sin(\pi x_1^2) \sin(2\pi x_2)$ defined in the original paper of Nguyen and Widrow (Nguyen and Widrow, 1990) where they introduce their weight initialization method. Inputs for this problem are in the interval $[-1, 1]$, all nodes in the hidden layer use the hyperbolic tangent sigmoid activation function and nodes in the output layer are linear. The network is fully connected and so there are $2 * 21 + 21 * 1 = 63$ unknown weights as well as $21 + 1 = 22$ bias weights. The interval equation of the weights

to any node in the hidden layer is (7b). Retaining for $\mathbf{b} = \mathbf{b}_2 = 18.04$, as in the previous benchmark, we have

$$[w_1][-1, 1] + [w_2][-1, 1] + \dots + [w_n][-1, 1] + [w_b][1, 1] = [-18.4, 18.4].$$

Moreover, the interval equation of the weights to any node in the output layer is,

$$[w_1][-1, 1] + [w_2][-1, 1] + \dots + [w_n][-1, 1] + [w_b][1, 1] = [-0.5, 0.5].$$

So, the initial box used for searching the global minimizers is

$$\underbrace{[-18.4, 18.4]^{(2+1)*21}}_{\text{Hidden layer}} \times \underbrace{[-0.5, 0.5]^{(21+1)*1}}_{\text{Output layer}}.$$

The output provided by GOP for the above network is:

Function name: FUNCTION APPROXIMATION

The set of global minimizers is located in the union of the following boxes:

c1: [-18.400000000000000, 18.400000000000000]...

[-0.500000000000000, 0.500000000000000]

The global minimum is enclosed in:
[0.00000000000000000, 0.048430679703413922].

5.3 Discussion and Open Problems

The above experiments provide substantial evidence regarding the validity of the theoretical results, showing that the convex set computed by the proposed approach for each problem contains some global minimizers of the network's error function. In addition, it is easy to see that there exist areas in the weight space which do not contain global minimizers.

In all cases the approach for evaluating the theoretical results was based on an interval global optimization procedure. However, the theoretical analysis presented in Section 4 made no assumption regarding the global optimization procedure used to minimize the error function of the MLP. Therefore, the set of boxes enclosing the global minimizers is determined independently of the global optimization method used for training, and so it is also valid for population-based or stochastic global search methods. It is worth noting here that the results reported by Gudise and Venayagamoorthy (2003) constitute an experimental verification of this statement when PSO is used to train an MLP. Moreover, the experimental analysis provided in that paper, concerning the computational cost induced by the training procedure in relation with the width of the search space, may be seen as an evidence of the advantage offered by our approach.

An interesting point of the above experiments concerns the results obtained when the boxes are defined under the assumption that the outputs of the sigmoid and the hyperbolic tangent activation functions are $[0, 1, 0, 9]$ and $[-0.9, 0, 9]$ instead of $[0, 1]$ and $[-1, 1]$. These results indicate that the weight boxes defined so, besides being thinner, also result in a narrower interval enclosing the global minimum. We note here, without further details, that the derivation of these boxes and the entailing results, may be justified using the following theorem by Moore (1966) as formulated in Alefeld and Mayer (2000, Theorem 1).

Theorem 8 *Let $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}$ be continuous and let $[x] \subseteq [x]^0 \subseteq D$. Then (under mild additional assumptions) $q(R(f; [x]), [f]([x])) \leq \gamma \|d([x])\|_\infty$, $\gamma \geq 0$, and $d([f]([x])) \leq \delta \|d([x])\|_\infty$, $\delta \geq 0$.*

Here, $q(\cdot, \cdot)$ is the Hausdorff distance between intervals which measures the quality of the interval inclusion function $[f]$ as an enclosure of the range of f over an interval $[x]$. This Theorem states that if the inclusion function exists then the Hausdorff distance between $R(f; [x])$ and $[f]([x])$ goes linearly to zero with the diameter $d([x])$ (Alefeld and Mayer, 2000). In addition the diameter of the inclusion function goes linearly to zero if $d([x])$ is approaching zero. The values of parameters γ and δ are very important for defining thinner boxes.

Regarding the previous comment it is important to note that considering narrower intervals for the outputs of the nonlinear activation functions of the nodes cannot be done arbitrarily. Actually, narrowing the intervals of the activation functions' output values causes a decrease in their level of saturation which translates to less sharp values for the network outputs. Obviously, this introduces a degree of uncertainty to the network outputs and thereby to the network function itself. The importance of this outcome seems to be problem dependent. If the underlying problem deals with pattern classification this outcome may have no effect on the classification task and in the best case even increase the generalization ability of the network. On the other hand when dealing with a function approximation or regression problem defining less sharp values for the network outputs is very likely to cause large deviations from the target outputs. To the best of our knowledge there is no research report attempting to define any proper saturation level for the activation functions in order to fit any unknown function with infinite accuracy.

Finally, an issue that needs to be commented concerns the impact of the proposed method on the global optimization procedure, in terms of computational complexity and cost. As the results of the experiments underline, this issue has not been addressed in this paper mainly for two reasons; first the objective of our paper is not to detect the global minimizers of the network error function but merely to delimit the area where a global optimization procedure should search for them. Secondly, evaluating the computational cost of our approach requires either a suitable mathematical formulation or significant computational effort in order to effectively measure the impact of our approach on specific global optimization procedures. Definitely, this issue constitutes our main concern for continuing this research.

6. Conclusion

The approach presented in this paper deals with computing guaranteed bounds for the region where global optimization procedures should search for global minimizers when training an MLP. In contrast to current practice, which defines these bounds heuristically, the proposed approach relies on interval analysis and exploits both the network architecture and information of the input patterns for shaping the search region. The resulting feasible domain of the network's output error function is rather complicated and so a suitable approximation is derived in the form of a convex polytope or a hypercube depending on the network architecture and the problem at hand.

The analysis presented deals with 3-layer feed forward neural networks but the results

can easily be extended to networks with more than one hidden layer. Moreover, this analysis covers the widely used type of feed forward neural networks using some kind of sigmoid activation function for the nodes in the hidden layer. Nodes in the output layer may use either a sigmoid or a linear activation function. The conclusions derived are applicable to both standard MLPs and interval feed forward neural networks.

The examples and the experiments given on well known benchmarks highlight the application of the theoretical results formally derived in the paper. The results of the experiments provide significant evidence that the global minimizers of the considered network error functions fall within the bounds of the polytope defined by our method. Moreover, the proposed approach is not restrictive in terms of the hidden layers considered for the MLP or the bounds of the interval for node's output values.

A thorough performance analysis of global optimization procedures with and without using the proposed approach was not part of the objectives of this paper — this will be considered in future work. Also, we are planning to investigate the possibility of dealing with MLPs that use other types of activation functions as well as explore the applicability of the results to other types of networks such as the radial basis function and the recurrent networks.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable suggestions and comments on earlier version of the manuscript, that helped to significantly improve the paper at hand.

Appendix A. Sample Code for an MLP Natural Inclusion Function

Here, we give an example of the MATLAB code used for the natural inclusion function of the function implemented by a 3-layer MLP for the XOR benchmark. It is straightforward to obtain the natural inclusion function by simply replacing real variables by their interval counterparts. So, all variables corresponding to the weights of the MLP, as well as the quantities computed with these variables, are intervals. Interval computations are automatically carried out using INTLAB (Rump, 1999).

```
% the following define the activation functions
avfHL = @(x)[1./(1+exp(-x))]; % for the hidden layer
avfOL = @(x)[1./(1+exp(-x))]; % for the output layer

% the code for the inclusion function
function y = evxor(WO)
% WO is the vector of the weight intervals
% global network parameters and training data
global p t np n h o avfHL avfOL

W1 = reshape(WO(1:h*n,1),h,n);
```

```

b1 = repmat(creshape(W0((h*n)+1:(h*n)+h*1,1),h,1),1,np);
W2 = reshape(W0((h*(n+1))+1:(h*(n+1))+o+h,1),o,h);
b2 = repmat(creshape(W0(h*(n+1)+o+h+1:end,1),o,1),1,np);
y = sum((-avf0L(W2*avfHL(W1*p+b1)+b2)).^-2) ./np;
% end function

```

References

- S.P. Adam, D.A. Karras, G.D. Magoulas, and M.N. Vrahatis. Solving the linear interval tolerance problem for weight initialization of neural networks. *Neural Networks*, 54:17–37, 2014.
- G. Alefeld and G. Mayer. Interval analysis: theory and applications. *Journal of Computational and Applied Mathematics*, 121:421–464, 2000.
- A.M. Bagirov, A.M. Rubinov, and J. Zhang. A multidimensional descent method for global optimization. *Optimization*, 58(5):611–625, 2009.
- O. Beaumont. Solving interval linear systems with linear programming techniques. *Linear Algebra and its Applications*, 281:293–309, 1998.
- D. Bertsimas and J. Tsitsiklis. Simulated annealing. *Statistical Science*, 8(1):10–15, 1993.
- C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, New York, 1995.
- C.G.E. Boender, A.H.G. Rinnooy Kan, G.T. Timmer, and L. Stongie. A stochastic method for global optimization. *Mathematical Programming*, 22:125–140, 1982.
- S.H. Brooks. A discussion of random methods for seeking maxima. *Operations Research*, 6(2):244–251, 1958.
- R.E. Caflisch. Monte Carlo and quasi-Monte Carlo methods. *Acta Numerica*, 7:1–49, 1998.
- O. Caprani, B. Godthaab, and K. Madsen. Use of a real-valued local minimum in parallel interval global optimization. *Interval Computations*, 2:71–82, 1993.
- P.A. Castillo, J.J. Merelo, A. Prieto, V. Rivas, and G. Romero. G-prop: Global optimization of multilayer perceptrons using GAs. *Neurocomputing*, 35:149–163, 2000.
- M. Clerc and J. Kennedy. The particle swarm – explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1):58–73, 2002.
- M. Courbariaux, Y. Bengio, and J.-P. David. Training deep neural networks with low precision multiplications. *ArXiv e-prints*, abs/1412.7024, 2014. Available electronically via <http://arxiv.org/abs/1412.7024>.
- G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control Signals and Systems*, 2:303–314, 1989.
- D.H. Deterding. *Speaker Normalisation for Automatic Speech Recognition*. PhD thesis, University of Cambridge, 1989.
- S. Draghici. On the capabilities of neural networks using limited precision weights. *Neural Networks*, 15(3):395–414, 2002.
- W. Duch and N. Jankowski. Survey of neural transfer functions. *Neural Computing Surveys*, 2:163–212, 1999. Available electronically at <http://ftp.icsi.berkeley.edu/pub/ai/jagota/vol2.6.pdf>.
- W. Duch and J. Korczak. Optimization and global minimization methods suitable for neural networks. *Neural Computing Surveys*, 2:163–212, 1998. Available electronically via <http://www.fizyka.umk.pl/publications/kmk/99globmin.html>.
- A.E. Eiben, E.H.L. Aarts, and K.M. Van Hee. Global convergence of genetic algorithms: A Markov chain analysis. In Hans-Paul Schwefel and Reinhard Manner, editors, *Parallel Problem Solving from Nature*, volume 496 of *Lecture Notes in Computer Science*, pages 3–12. Springer, Berlin, 1991.
- J. Engel. Teaching feed-forward neural networks by simulated annealing. *Complex Systems*, 2(6):641–648, 1988.
- V.G. Gudiş and G.K. Venayagamoorthy. Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks. In *Proceedings of the IEEE Swarm Intelligence Symposium, SIS '03*, pages 110–117, 2003.
- S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan. Deep learning with limited numerical precision. In David Blei and Francis Bach, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 1737–1746. JMLR Workshop and Conference Proceedings, 2015.
- L.G.C. Haney. Analysis of the error surface of the XOR network with two hidden nodes. Computing Report 95/167C, Department of Computing, Macquarie University, NSW 2109, Australia, 1995.
- L.G.C. Haney. XOR has no local minima: A case study in neural network error surface analysis. *Neural Networks*, 1(4):669–681, 1998.
- E.R. Hansen. On the solution of linear algebraic equations with interval coefficients. *Linear Algebra and its Applications*, 2(2):153–165, 1969.
- E.R. Hansen and S. Sengupta. Bounding solutions of systems of equations using interval analysis. *BIT Numerical Mathematics*, 21:203–211, 1981.
- E.R. Hansen and G.W. Walster. *Global Optimization Using Interval Analysis*. Marcel Dekker, New York, 2004.

- S. Haykin. *Neural Networks A Comprehensive Foundation*. Prentice-Hall, Upper Saddle River, New Jersey, 1999.
- S. Helwig and R. Wanka. Theoretical analysis of initial particle swarm behavior. In G. Rudolph, T. Jansen, S. Lucas, C. Poloni, and N. Beume, editors, *Parallel Problem Solving from Nature – PPSN X*, volume 5199 of *Lecture Notes in Computer Science*, pages 889–898. Springer, Berlin, 2008.
- N. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, 2002.
- M. Hoehfeld and S. E. Fahlman. Learning with limited numerical precision using the cascade-correlation algorithm. *IEEE Transactions on Neural Networks*, 3(4):602–611, 1992.
- C. Hölbig and W. Krämer. Self-verifying solvers for dense systems of linear equations realized in C-XSC. Universität Wuppertal, Preprint BUGHW-WRSWT 2003/1, 2003. Available electronically via <http://www.math.uni-wuppertal.de/wrswt/literatur.html>.
- J. L. Holí and J. N. Hwang. Finite precision error analysis of neural network hardware implementations. *IEEE Transactions on Computers*, 42(3):281–290, 1993.
- K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.
- R. Horst and P.M. Pardalos. *Handbook of Global Optimization*. Kluwer Academic Publishers, Dordrecht, 1995.
- C. Hu, M. Beheshti, A. Berrached, A. de Korvin, and O. Sirisaengtaksin. On interval weighted three-layer neural networks. In *Proceedings of the 31st Annual Simulation Symposium*, pages 188–194. IEEE Computer Society Press, 1998.
- IEEE. Standard for floating-point arithmetic. *IEEE Std 754-2008*, pages 1–70, Aug 2008.
- J. Ilonen, J.K. Kamarainen, and J. Lampinen. Differential evolution training algorithm for feed-forward neural networks. *Neural Processing Letters*, 17(1):93–105, 2003.
- M. Jamett and G. Acuña. An interval approach for weight's initialization of feedforward neural networks. In *Proceedings of the 5th Mexican International Conference on Artificial Intelligence, MICAI 2006*, volume 4293 of *LNCS*, pages 305–315. Springer-Verlag, Berlin, 2006.
- C. Jansson. Calculation of exact bounds for the solution set of linear interval systems. *Linear Algebra and its Applications*, 251:321–340, 1997.
- L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis. With Examples in Parameter and State Estimation, Robust Control and Robotics*. Springer-Verlag, London, 2001.
- P. Judd, J. Albericio, T. Hetherington, T. M. Aamodt, N. E. Jeger, and A. Moshovos. Proteus: Exploiting numerical precision variability in deep neural networks. In *Proceedings of the 2016 International Conference on Supercomputing, ICS '16*, pages 23:1–23:12, New York, 2016. ACM.
- S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- W. Krämer, U. Kulisch, and R. Lohner. *Numerical Toolbox for Verified Computing II: Advanced Numerical Problems*. Springer-Verlag, Berlin, 2006.
- V. Kreinovich and O. Sirisaengtaksin. 3-layer neural networks are universal approximators for functionals and for control strategies. *Neural Parallel & Scientific Computations*, 1: 325–346, 1993.
- V. Kreinovich, A.V. Lakeyev, and S.I. Noskov. Optimal solution of interval linear systems is intractable (NP-hard). *Interval Computations*, 1:6–14, 1993.
- Y. LeCun. Efficient learning and second-order methods. Tutorial at Neural Information Processing Systems Conference, NIPS, 1993.
- G.D. Magoulas, M.N. Vrahatis, and G.S. Androulakis. Effective back-propagation training with variable stepsize. *Neural Networks*, 10(1):69–82, 1997.
- G.D. Magoulas, M.N. Vrahatis, and G.S. Androulakis. Improving the convergence of the backpropagation algorithm using learning rate adaptation methods. *Neural Computation*, 11(7):1769–1796, 1999.
- MATLAB-Dokumentation. Floating-point relative accuracy – MATLAB eps. Online documentation, 2014. Retrieved 25 July 2014 from <http://www.mathworks.com/help/matlab/ref/eps.html>.
- D.J. Montana and L. Davis. Training feedforward neural networks using genetic algorithms. In N.S. Sridharan, editor, *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pages 762–767. Morgan Kaufmann, 1989.
- R.E. Moore. *Interval Analysis*. Prentice-Hall, Englewood Cliffs, New Jersey, 1966.
- A. Neumaier. Linear interval equations. In K. Nickel, editor, *Interval Mathematics 1985*, volume 212 of *Lecture Notes in Computer Science*, pages 109–120. Springer, Berlin, 1986.
- A. Neumaier. *Interval Methods for Systems of Equations*. Cambridge University Press, New York, 1990.
- C. Ng, D. Li, and L. Zhang. Global descent method for global optimization. *SIAM Journal on Optimization*, 20(6):3161–3184, 2010.
- D. Nguyen and B. Widrow. Improving the learning speed of two-layer neural networks by choosing initial values of the adaptive weights. In *Proceedings of the International Joint Conference on Neural Networks, IJCNN '90*, volume 3, pages 21–26, Ann Arbor, Michigan, 1990.

- L. Pál. *Global Optimization Algorithms for Bounded Constrained Problems*. PhD Dissertation, PhD School in Computer Science, University of Szeged, Szeged, Hungary, 2010.
- L. Pál and T. Csendes. INTLAB implementation of an interval global optimization algorithm. *Optimization Methods and Software*, 24(4-5):749–759, 2009. Available electronically via <http://www.inf.u-szeged.hu/~csendes/Reg/regform.php>.
- K.E. Parsopoulos and M.N. Vrahatis. *Particle Swarm Optimization and Intelligence: Advances and Applications*. Information Science Publishing (IGI Global), Hershey, Pennsylvania, 2010.
- M.E.H. Pedersen. *Tuning & Simplifying Heuristical Optimization*. PhD thesis, University of Southampton, School of Engineering Sciences, Computational Engineering and Design Group, 2010.
- V.P. Plagianakos, G.D. Magoulas, and M.N. Vrahatis. Learning in multilayer perceptrons using global optimization strategies. *Nonlinear Analysis: Theory, Methods & Applications*, 47(5):3431–3436, 2001.
- V.P. Plagianakos, G.D. Magoulas, and M.N. Vrahatis. Improved learning of neural nets through global search. In Janos D. Pinter, editor, *Global Optimization*, volume 85 of *Nonconvex Optimization and Its Applications*, pages 361–388. Springer, 2006.
- E.D. Popova. Improved solution enclosures for over- and underdetermined interval linear systems. In Ivan Litkov, Svetozar Margenov, and Jerzy Wasniowski, editors, *Large-Scale Scientific Computing*, volume 3743 of *Lecture Notes in Computer Science*, pages 305–312. Springer, Berlin, 2006.
- H. Ratschek and W. Sauer. Linear interval equations. *Computing*, 28:105–115, 1982.
- J. Rohn. Systems of linear interval equations. *Linear Algebra and its Applications*, 126: 39–78, 1989.
- J. Rohn. Checking bounds on solutions of linear interval equations is NP-hard. *Linear Algebra and its Applications*, 223–224:589–596, 1995.
- J. Rohn. Solvability of systems of linear equations. *SIAM Journal on Matrix Analysis and Applications*, 25:237–245, 2003.
- D.E. Rummelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. In D.E. Rummelhart and J.L. McClelland, editors, *Parallel Distributed Processing*. MIT Press, Cambridge, Massachusetts, 1986.
- S.M. Rump. Solving algebraic problems with high accuracy. In U.W. Kulisch and W.L. Miranker, editors, *A New Approach to Scientific Computation*, pages 51–120. Academic Press, New York, 1983.
- S.M. Rump. INTLAB – INTERVAL LABORATORY. In Tibor Csendes, editor, *Developments in Reliable Computing*, pages 77–104. Kluwer Academic, Dordrecht, Netherlands, 1999.
- P.V. Saraev. Numerical methods of interval analysis in learning neural network. *Automation and Remote Control*, 73(11):1865–1876, 2012.
- Y. Shang and B.W. Wah. Global optimization for neural network training. *IEEE Computer*, 29(3):45–54, 1996.
- S.P. Shary. On optimal solution of interval linear equations. *SIAM Journal on Numerical Analysis*, 32(2):610–630, 1995.
- S.P. Shary. Algebraic approach to the interval linear static identification, tolerance, and control problems, or one more application of Kantor arithmetic. *Reliable Computing*, 2(1):3–33, 1996.
- S.P. Shary. Algebraic approach in the “outer problem” for interval linear equations. *Reliable Computing*, 3(2):103–135, 1997.
- S.P. Shary. A new technique in systems analysis under interval uncertainty and ambiguity. *Reliable Computing*, 8:321–418, 2002.
- I.G. Sprinkhuizen-Kuyper and E.J.W. Boers. The local minima of the error surface of the 2-2-1 XOR network. *Annals of Mathematics and Artificial Intelligence*, 25(1–2):107–136, 1999.
- Z. Tang and G.J. Koehler. Deterministic global optimal FNN training algorithms. *Neural Networks*, 7(2):301–311, 1994.
- F. van den Bergh and A.P. Engelbrecht. Cooperative learning in neural networks using particle swarm optimizers. *South African Computer Journal*, 26:84–90, 2000.
- S. Vassiliadis, M. Zhang, and J. G. Delgado-Frias. Elementary function generators for neural-network emulators. *IEEE Transactions on Neural Networks*, 11(6):1438–1449, 2000.
- C. Voglis and I.E. Lagaris. Towards “ideal multistart”. A stochastic approach for locating the minima of a continuous function inside a bounded domain. *Applied Mathematics and Computation*, 213(1):216–229, 2009.
- H. White. Connectionist nonparametric regression: multilayer feedforward networks can learn arbitrary mappings. *Neural Networks*, 3:535–549, 1990.
- P. Xu. A hybrid global optimization method: the one-dimensional case. *Journal of Computational and Applied Mathematics*, 147(12):301–314, 2002.
- J.Y.F. Yam and T.W.S. Chow. Feedforward networks training speed enhancement by optimal initialization of the synaptic coefficients. *IEEE Transactions on Neural Networks*, 12:430–434, 2001.

mlr: Machine Learning in R

Bernd Bischl

Michel Lang

Lars Kotthoff

Julia Schiffner

Jakob Richter

Erich Studerus

Giuseppe Casalicchio

Zachary M. Jones

Department of Statistics

Ludwig-Maximilians-University Munich

Ludwigstrasse 33, 80539 Munich, Germany

BERND.BISCHL@STAT.UNI-MUENCHEN.DE

LANG@STATISTIK.TU-DORTMUND.DE

LARSKO@CS.UBC.CA

SCHIFFNER@MATH.UNI-DUESSELDORF.DE

JAKOB.RICHTER@TU-DORTMUND.DE

ERICH.STUDERUS@UPKBS.CH

GIUSEPPE.CASALICCHIO@STAT.UNI-MUENCHEN.DE

ZMJ@ZMJONES.COM

Editor: Antti Honkela

Abstract

The MLR package provides a generic, object-oriented, and extensible framework for classification, regression, survival analysis and clustering for the R language. It provides a unified interface to more than 160 basic learners and includes meta-algorithms and model selection techniques to improve and extend the functionality of basic learners with, e.g., hyperparameter tuning, feature selection, and ensemble construction. Parallel high-performance computing is natively supported. The package targets practitioners who want to quickly apply machine learning algorithms, as well as researchers who want to implement, benchmark, and compare their new methods in a structured environment.

Keywords: machine learning, hyperparameter tuning, model selection, feature selection, benchmarking, R, visualization, data mining

1. Introduction

R is one of the most popular and widely-used software systems for statistics, data mining, and machine learning. However, it does not define a standardized interface to, e.g., supervised predictive modelling. For any non-trivial experiment one needs to write lengthy, tedious, and error-prone code to unify calling methods and handling output. The MLR package offers a clean, easy-to-use, and flexible domain-specific language for machine learning experiments in R. It supports classification, regression, clustering, and survival analysis with more than 160 modelling techniques. Defining learning tasks, training models, making predictions, and evaluating their performance abstracts from the implementation of the underlying learner through an object-oriented interface. Replacing one learning algorithm with another becomes as easy as changing a string. MLR goes far beyond simply providing a unified interface. It implements a generic architecture that allows the assessment of generalization performance, comparison of different algorithms in a scientifically rigorous way, feature selection, and hyperparameter tuning for any method, as well as extending

the functionality of learners through a wrapper mechanism. Queryable properties provide a reflection mechanism for machine learning objects. Finally, MLR provides sophisticated visualization methods that allow to show effects of partial dependence of models. MLR's long term goal is to provide a high-level domain-specific language to express as many aspects of machine learning experiments as possible.

2. Implemented Functionality

MLR uses R's S3 object system and follows a clear structure. Everything is an object and the classes are as reusable and extensible as possible. This permits to extend the package; e.g., connect a new model from a third-party package or write a custom performance measure.

Tasks and Learners. Tasks encapsulate the data and further relevant information like the name of the target variable for supervised learning problems. They are organized hierarchically, with an abstract Task at the top and specific subclasses. MLR supports regular, multilabel and cost-sensitive classification, regression, survival analysis, and clustering. The integrated learners specialize to these task types. Currently 82 classification learners, 61 regression learners, 13 survival learners, and 9 cluster learners are integrated. Cost-sensitive classification with observation-dependent costs is supported through a cost-sensitive one-versus-one approach, which delegates to ordinary weighted binary classification.

Evaluation and Resampling. MLR provides 46 different performance measures and implements the resampling methods subsampling (including simple holdout), bootstrapping (OOB, B632, B632+), and cross-validation (normal, leave-one-out, repeated). All resampling strategies may be stratified on both target classes and categorical input features. Observations may be partitioned into inseparable blocks (e.g., when observations come from the same image, sound file, or clinic). Moreover, nested resampling is supported and the resampling strategies used in the outer and inner loops can be combined arbitrarily.

Tuning. In practice, successful modelling often depends on a number of choices like the applied learner, its hyperparameter settings, or the data preprocessing. MLR implements joint optimization of hyperparameters of any learning algorithm and any pre- and postprocessing methods for any task, any resampling strategy, and any performance measure, including categorical and conditional hyperparameters. Random search, grid search, evolutionary algorithms, iterated F-racing, and sequential model-based optimization are available.

Feature Selection. Feature selection can improve the interpretability and performance of a learned predictive model. MLR supports *filter* and *wrapper* approaches, while *embedded* techniques like L₁-penalization are included directly in the learners. Supported selection techniques include information gain, MRM, and RELIEF, with forward and backward search. Filter scores and sequential wrapper search results can be visualized.

Wrapper Extensions. MLR's wrapper mechanism allows to extend learners through pre-train, post-train, pre-predict, and post-predict hooks. We provide wrappers for missing value imputation, user-defined preprocessing, class imbalance correction, feature selection, tuning, bagging, and stacking. Wrappers can be nested to combine functionalities. Wrapped learners behave like base learners, with added functionality and expanded hyperparameter set. During resampling, all added steps are carried out in each iteration. During tuning,

the joint parameter space can be optimized. For example thresholds for feature filtering can be tuned jointly with other hyperparameters (Lang et al., 2015).

Benchmarking and Parallelization. The benchmark function evaluates the performance of multiple learners on multiple tasks. As benchmark studies can quickly become very resource-demanding, MLR natively supports parallelization through the PARALLELMAP package (Bischl and Lang, 2015) that can use local multicore, socket, and MPI computation modes. BATCHOBS (Bischl et al., 2015) provides distribution on compute clusters. Operations to be parallelized can be selected explicitly.

Properties and Parameters. Many of the MLR objects have properties that allow them to be used programmatically, e.g., check whether a task has missing values, whether a learner can handle categorical variables, or list all learners suitable for a given task. Every learner includes a description object that defines all hyperparameters, including type, default value, and feasible range. This information is usually not readily available from the implementation of an integrated learning method and may only be listed in its documentation.

3. Example

The following example demonstrates the use of MLR. After loading required packages and the “Sonar” data set (Line 1), we create a classification task and a support vector machine learner (Lines 2–3). The resample description tells MLR to use a 5-fold cross-validation (Line 4). Hyperparameters and box-constraints for tuning are specified in Lines 5–11. We optimize over the choice of a polynomial versus a Gaussian kernel by making their individual parameters dependent on the kernel via the `requires` setting (Lines 9 and 11). We use random search with at most 50 evaluations (Line 12). The values for `C` and `sigma` are sampled on a log-scale through the transformation functions given as the `trafo` argument (Lines 7–8). Line 13 binds everything together and optimizes for mean misclassification error (`mnce`). `res` holds the best configuration and information on the evaluated parameters.

```

1 library(mlr); library(mlbench); data(Sonar)
2 task = makeClassifTask(data=Sonar, target="Class")
3 lrn = makeLearner("Classif.ksvm")
4 rdesc = makeResampleDesc(method="CV", iters=5)
5 ps = makeParamSet(
6   makeDiscretParam("kernel", values=c("polydot", "rbfdot")),
7   makeNumericParam("C", lower=-15, upper=15, trafo=function(x) 2^x),
8   makeNumericParam("sigma", lower=-15, upper=15, trafo=function(x) 2^x,
9     requires = quote(kernel == "rbfdot")),
10  makeIntegerParam("degree", lower = 1, upper = 5,
11    requires = quote(kernel == "polydot"))
12 ctrl = makeTuneControlRandom(maxit=50)
13 res = tuneParams(lrn, task, rdesc, par.set=ps, control=ctrl, measures=mnce)

```

4. Availability, Documentation, Maintenance, and Code Quality Control

The MLR source code is available under the BSD 2-clause license and hosted on GitHub (<https://github.com/mlr-org/mlr>). Stable releases are frequently published on the Contributed R Archive Network (CRAN), which lists MLR in Task View ‘Machine Learning & Statistical Learning’. We provide extensive API documentation through R’s internal help

BISCHL, LANG, KOTTHOFF, SCHIFFNER, RICHTER, STURDERUS, CASALICCHIO AND JONES

system and a very detailed tutorial (Schiffer et al., 2016) that guides the user from very basic tasks to complex applications with worked examples and is continuously extended. An issue tracker, the test framework TESTTHAT (with more than 10,000 lines of tests and more than 1,200 assertions), and the CI systems Travis and Jenkins support the correctness of the code base. In addition, we provide documentation and coding guidelines for developers and contributors.

5. Comparison to Similar Toolkits/Frameworks

Several other R packages provide frameworks for handling prediction models, including CARET (Kuhn, 2008), DMWR (Torgo, 2010), CORELEARN (Robnik-Sikonja and with contributions from John Adeyariju Alao, 2016), RATTLE (Williams, 2011), RMINER (Cortez, 2010), CMA (Slawski et al., 2008), and RPRED (Peters and Hothorn, 2015). The first 5 only support classification and regression, CMA only classification. MLR’s generic wrapper mechanism is not provided by any other package in this form. Although CARET and CMA can fuse a learner with a preprocessing or variable selection method, only MLR can seamlessly tune these methods simultaneously (Koch et al., 2012). Only MLR, RMINER, and CMA support nested cross-validation. A similar degree of flexibility can be achieved in CARET, but requires custom implementations. Only MLR supports ensemble learning through stacking natively. MLR and CARET support bagging natively. Bagging is also available in RPRED and CARETENSEMBLE provides stacking for CARET. Only MLR and CARET have native support for parallel computations. Similar toolkits exist for other languages, e.g., WEKA for Java (Hall et al., 2009) and SCKITR-LEARN for Python (Pedregosa et al., 2011).

6. Conclusions and Outlook

We presented the MLR package, which provides a unified interface to machine learning in R. It implements a generic architecture for a range of common machine learning tasks. MLR is alive and under active development. It has a growing user community and is used for teaching and research.

Major directions for future extensions include better support for large-scale data, a closer connection to the OpenML project (Van Schooren et al., 2013) for open machine learning experiments,¹ and better integration of sequential model-based optimization.²

Acknowledgments

This work was supported by the Deutsche Forschungsgemeinschaft [SCHW 1508/3-1 to J.S.] and Collaborative Research Center SFB 876, project A3.

1. An OpenML-R connector package is available at <https://github.com/openml/r>.

2. MLR supports an experimental integration via mlrMBO (<https://github.com/mlr-org/mlrMBO>).

References

- B. Bischl and M. Lang. *parallelMap: Unified interface to some popular parallelization backends for interactive usage and package development*, 2015. URL <https://github.com/berndbischl/parallelMap>. R package version 1.3.
- B. Bischl, M. Lang, O. Mersmann, J. Rahnenführer, and C. Weils. BatchJobs and BatchExperiments: Abstraction mechanisms for using R in batch environments. *Journal of Statistical Software*, 64(11), 2015.
- P. Cortez. Data Mining with Neural Networks and Support Vector Machines using the R/miner Tool. In P. Perner, editor, *Advances in Data Mining. Applications and Theoretical Aspects*, volume 6171 of *LNC3*, pages 572–583, Berlin, Germany, 2010. Springer.
- M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: An update. *SIGKDD Explorations*, 11(1), 2009.
- P. Koch, B. Bischl, O. Flasch, T. Bartz-Beielstein, C. Weils, and W. Konen. Tuning and evolution of support vector kernels. *Evolutionary Intelligence*, 5(3):153–170, 2012.
- M. Kuhn. Building predictive models in R using the caret package. *Journal of Statistical Software*, 28(5):1–26, 2008.
- M. Lang, H. Kotthaus, P. Marwedel, C. Weils, J. Rahnenführer, and B. Bischl. Automatic model selection for high-dimensional survival analysis. *Journal of Statistical Computation and Simulation*, 85(1):62–76, 2015.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- A. Peters and T. Hothorn. *ipred: Improved Predictors*, 2015. URL <http://CRAN.R-project.org/package=ipred>. R package version 0.9-5.
- M. Robnik-Sikonja and P. S. with contributions from John Adeyanju Alao. *CORE-learn: Classification, Regression and Feature Evaluation*, 2016. URL <https://CRAN.R-project.org/package=CORElearn>. R package version 1.48.0.
- J. Schiffler, B. Bischl, M. Lang, J. Richter, Z. M. Jones, P. Probst, F. Pfisterer, M. Gallo, D. Kirchhoff, T. Kühn, J. Thomas, and L. Kotthoff. mlr tutorial, 2016.
- M. Slawski, M. Daumer, and A.-L. Boulesteix. CMA – a comprehensive Bioconductor package for supervised classification with high dimensional data. *BMC Bioinformatics*, 9(1):439, 2008.
- L. Torgo. *Data Mining with R: Learning with Case Studies*. Data Mining and Knowledge Discovery Series. Chapman and Hall/CRC, Boca Raton, FL, 2010.
- J. Vanschoren, J. N. van Rijn, B. Bischl, and L. Torgo. OpenML: Networked science in machine learning. *SIGKDD Explorations*, 15(2):49–60, 2013.
- G. J. Williams. *Data Mining with Rattle and R: The Art of Excavating Data for Knowledge Discovery*. Use R! Springer, New York, NY, 2011.

Feature-Level Domain Adaptation

Wouter M. Kouw

Laurens J.P. van der Maaten

*Department of Intelligent Systems
Delft University of Technology
Mekelweg 4, 2628 CD, the Netherlands*

W.M.KOUW@TUDELFT.NL

L.J.P.VANDERMAATEN@TUDELFT.NL

Jesse H. Krijthe

*Department of Intelligent Systems
Delft University of Technology
Mekelweg 4, 2628 CD Delft, the Netherlands
Department of Molecular Epidemiology
Leiden University Medical Center
Eindhovenvweg 20, 2333 ZC Leiden, the Netherlands*

JKRIJTHE@GMAIL.COM

Marco Loog

*Department of Intelligent Systems
Delft University of Technology
Mekelweg 4, 2628 CD Delft, the Netherlands
The Image Group,
University of Copenhagen
Universitetsparken 5, DK-2100, Copenhagen, Denmark*

M.LOOG@TUDELFT.NL

Editor: Urun Dogan, Marius Kloft, Francesco Orabona, and Tatiana Tonmazi

Abstract

Domain adaptation is the supervised learning setting in which the training and test data are sampled from different distributions: training data is sampled from a source domain, whilst test data is sampled from a target domain. This paper proposes and studies an approach, called feature-level domain adaptation (FLDA), that models the dependence between the two domains by means of a feature-level transfer model that is trained to describe the transfer from source to target domain. Subsequently, we train a domain-adapted classifier by minimizing the expected loss under the resulting transfer model. For linear classifiers and a large family of loss functions and transfer models, this expected loss can be computed or approximated analytically, and minimized efficiently. Our empirical evaluation of FLDA focuses on problems comprising binary and count data in which the transfer can be naturally modeled via a dropout distribution, which allows the classifier to adapt to differences in the marginal probability of features in the source and the target domain. Our experiments on several real-world problems show that FLDA performs on par with state-of-the-art domain-adaptation techniques.

Keywords: Domain adaptation, transfer learning, covariate shift, risk minimization

1. Introduction

Domain adaptation is an important research topic in machine learning and pattern recognition that has applications in, among others, speech recognition (Leggetter and Woodland, 1995), medical image processing (Van Oprooek et al., 2013), computer vision (Saenko et al., 2010), social signal processing (Zen et al., 2014), natural language processing (Peddinti and Chintalapoodi, 2011), and bioinformatics (Borgwardt et al., 2006). Domain adaptation deals with supervised-learning settings in which the common assumption that the training and the test observations stem from the same distribution is dropped. This learning setting may arise, for instance, when the training data is collected with a different measurement device than the test data, or when a model that is trained on one data source is deployed on data that comes from another data source. This creates a learning setting in which the training set contains samples from one distribution (the so-called source domain), whilst the test set constitutes samples from another distribution (the target domain). In domain adaptation, one generally assumes a transductive learning setting: that is, it is assumed that the unlabeled test data are available to us at training time and that the main goal is to predict their labels as well as possible.

The goal of domain-adaptation approaches is to exploit information on the dissimilarity between the source and target domains that can be extracted from the available data in order to make more accurate predictions on samples from the target domain. To this end, many domain adaptation approaches construct a *sample-level transfer model* that assigns weights to observations from the source domain in order to make the source distribution more similar to the target distribution (Shimodaira, 2000; Huang et al., 2006; Cortes et al., 2008; Gretton et al., 2009; Cortes and Mohri, 2011). In contrast to such sample-level reweighing approaches, in this work, we develop a *feature-level transfer model* that describes the shift between the target and the source domain for each feature individually. Such a feature-level approach may have advantages in certain problems: for instance, when one trains a natural language processing model on news articles (the source domain) and applies it to Twitter data (the target domain), the marginal distribution of some of the words or n-grams (the features) is likely to vary between target and source domain. This shift in the marginal distribution of the features cannot be modeled well by sample-level transfer models, but it can be modeled very naturally by a feature-level transfer model.

Our feature-level transfer model takes the form of a conditional distribution that, conditioned on the training data, produces a probability density of the target data. In other words, our model of the target domain thus comprises a convolution of the empirical source distribution and the transfer model. The parameters of the transfer model are estimated by maximizing the likelihood of the target data under the model of the target domain. Subsequently, our classifier is trained as to minimize the expected value of the classification loss under the target-domain model. We show empirically that when the true domain shift can be modeled by the transfer model, under certain assumptions, our domain-adapted classifier converges to a classifier trained on the true target distribution. Our feature-level approach to domain adaptation is general in that it allows the user to choose a transfer model from a relatively large family of probability distributions. This allows practitioners to incorporate domain knowledge on the type of domain shift in their models. In the experimental section of this paper, we focus on a particular type of transfer distribution that is well-suited for

problems in which the features are binary or count data (as often encountered in natural language processing), but the approach we describe is more generally applicable. In addition to experiments on artificial data, we present experiments on several real-world domain adaptation problems, which show that our feature-level approach performs on par with the current state-of-the-art in domain adaptation.

The outline of the remainder of this paper is as follows. In Section 2, we give an overview of related prior work on domain adaptation. Section 3 presents our feature-level domain adaptation (FLDA) approach. In Section 4, we present our empirical evaluation of feature-level domain adaptation and Section 5 concludes the paper with a discussion of our results.

2. Related Work

Current approaches to domain adaptation can be divided into one of three main types. The first type constitutes *importance weighting* approaches that aim to reweigh samples from the source distribution in an attempt to match the target distribution as well as possible. The second type are *sample transformation* approaches that aim to transform samples from the source distribution in order to make them more similar to samples from the target distribution. The third type are *feature augmentation* approaches that aim to extract features that are shared across domains. Our feature-level domain adaptation (FLDA) approach is an example of a sample-transformation approach.

2.1 Importance Weighting

Importance-weighting approaches assign a weight to each source sample in such a way as to make the reweighted version of the source distribution as similar to the target distribution as possible (Shimodaira, 2000; Hwang et al., 2006; Cortes et al., 2008; Gretton et al., 2009; Cortes and Mohri, 2011; Gong et al., 2013; Baktashmoghlagh et al., 2014). If the class posteriors are identical in both domains (that is, the covariate-shift assumption holds) and the importance weights are unbiased estimates of the ratio of the target density to the source density, then the importance-weighted classifier converges to the classifier that would have been learned on the target data if labels for that data were available (Shimodaira, 2000). Despite their theoretic appeal, importance-weighting approaches generally do not to perform very well when the data set is small, or when there is little “overlap” between the source and target domain. In such scenarios, only a very small set of samples from the source domain is assigned a large weight. As a result, the effective size of the training set on which the classifier is trained is very small, which leads to a poor classification model. In contrast to importance-weighting approaches, our approach performs a *feature-level* reweighting. Specifically, FLDA assigns a data-dependent weight to each of the features that represents how informative this feature is in the target domain. This approach effectively uses all the data in the source domain and therefore suffers less from the small sample size problem.

2.2 Sample Transformation

Sample-transformation approaches learn functions that make the source distribution more similar to the target distribution (Blitzer et al., 2006, 2011; Pan et al., 2011; Gopalan et al.,

2011; Gong et al., 2012; Baktashmoghlagh et al., 2013; Dinh et al., 2013; Fernando et al., 2013; Shao et al., 2014). Most sample-transformation approaches learn global (non)linear transformations that map source and target data points into the same, shared feature space in such a way as to maximize the overlap between the transformed source data and the transformed target data (Gopalan et al., 2011; Pan et al., 2011; Gong et al., 2012; Fernando et al., 2013; Baktashmoghlagh et al., 2013). Approaches that learn a shared subspace in which both the source and the target data are embedded often minimize the maximum mean discrepancy (MMD) between the transformed source data and the transformed target data (Pan et al., 2011; Baktashmoghlagh et al., 2013). If used in combination with a universal kernel, the MMD criterion is zero when all the moments of the (transformed) source and target distribution are identical. Most methods minimize the MMD subject to constraints that help to avoid trivial solutions (such as collapsing all data onto the same point) via some kind of spectral analysis. An alternative to the MMD is the subspace disagreement measure (SDM) of Gong et al. (2012), which measures the discrepancy of the angles between the principal components of the transformed source data and the transformed target data. Most current sample-transformation approaches work well for “global” domain shifts such as translations or rotations in the feature space, but they are less effective when the domain shift is “local” in the sense that it strongly nonlinear. Similar limitations apply to the FLDA approach we explore, but it differs in that (1) our transfer model does not learn a subspace but operates in the original feature space and (2) the measure it minimizes to model the transfer is different, namely, the negative log-likelihood of the target data under the transferred source distribution.

2.3 Feature Augmentation

Several domain-adaptation approaches extend the source data and the target data with additional features that are similar in both domains (Blitzer et al., 2006; Li et al., 2014). Specifically, the approach by Blitzer et al. (2006) tries to induce correspondences between the features in both domains by identifying so-called pivot features that appear frequently in both domains but that behave differently in each domain; SVD is applied on the resulting pivot features to obtain a low-dimensional, real-valued feature representation that is used to augment the original features. This approach works well for natural language processing problems due to the natural presence of correspondences between features, e.g. words that signal each other. The approach of Blitzer et al. (2006) is related to many of the instantiations of FLDA that we consider in this paper, but it is different in the sense that we only use information on differences in feature presence between the source and the target domain to reweigh those features (that is, we do not explicitly augment the feature representation). Moreover, the formulation of FLDA is more general, and can be extended through a relatively large family of transfer models.

3. Feature-Level Domain Adaptation

Suppose we wish to train a sentiment classifier for reviews, and we have a data set with book reviews and associated sentiment labels (positive or negative review) available. After having trained a linear classifier on word-count representations of the book reviews, we wish to deploy it to predict the sentiment of kitchen appliance reviews. This leaves us

with a domain-adaptation problem on which the classifier trained on book reviews will likely not work very well: the classifier will assign large positive weights to, for instance, words such as “interesting” and “insightful” as these suggest positive book reviews and will be assigned large positive weights by a linear classifier. But these words hardly ever appear in reviews of kitchen appliances. As a result, a classifier trained naively on the book reviews may perform poorly on kitchen appliance reviews. Since the target domain data (the kitchen appliance reviews) are available at training time, a natural approach to resolving this problem may be to down-weight features corresponding to words that do not appear in the target reviews, for instance, by applying a high level of dropout (Hinton et al., 2012) to the corresponding features in the source data when training the classifier. The use of dropout mimics the target domain scenario in which the “interesting” and “insightful” features are hardly ever observed during the training of the classifier, and prevents that these features are assigned large positive weights during training. Feature-level domain adaptation FLDA aims to formalize this idea in a two-stage approach that (1) fits a probabilistic sample transformation model that aims to model the transfer between source and target domain and (2) trains a classifier by minimizing the risk of the source data under the transfer model.

In the first stage, FLDA models the transfer between the source and the target domain: the transfer model is a data-dependent distribution that models the likelihood of target data conditioned on observed source data. Examples of such transfer models may be a dropout distribution that assigns a likelihood of $1 - \theta$ to the observed feature value in the source data and a likelihood of θ to a feature value of 0, or a Parzen density estimator in which the mean of each kernel is shifted by a particular value. The parameters of the transfer distribution are learned by maximizing the likelihood of target data under the transfer distribution (conditioned on the source data). In the second stage, we train a linear classifier to minimize the expected value of a classification loss under the transfer distribution. For quadratic and exponential loss functions, this expected value and its gradient can be analytically derived whenever the transfer distribution factorizes over features and is in the natural exponential family; for logistic and hinge losses, practical upper bounds and approximations can be derived (Van der Maaten et al., 2013; Wager et al., 2013; Chen et al., 2014).

In the experimental evaluation of FLDA, we focus on applying dropout transfer models to domain-adaptation problems involving binary and count features. These features frequently appear in, for instance, bag-of-words features in natural language processing (Blei et al., 2003) or bag-of-visual-words features in computer vision (Jégou et al., 2012). However, we note that FLDA can be used in combination with a larger family of transfer models; in particular, the expected loss that is minimized in the second stage of FLDA can be computed or approximated efficiently for any transfer model that factorizes over variables and that is in the natural exponential family.

3.1 Notation

We assume a domain adaptation setting in which we receive pairs of samples and labels from the source domain, $S = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \sim p_{\mathcal{X}}, y_i \sim p_{\mathcal{Y}} \mid \mathcal{X}, \mathbf{x}_i \in \mathbb{R}^m, y_i \in Y\}_{i=1}^{|S|}$, at training time. Herein, the set Y is assumed to be a set of discrete classes, $|\cdot|$ denotes the cardinality of the sets and p refers to the probability distribution of its subscripted variable (\mathcal{X} for the source domain variable, \mathcal{Z} for the target domain variable and \mathcal{Y} for the class variable). At test

time, we receive samples from the target domain, $T = \{\mathbf{z}_j \mid \mathbf{z}_j \sim p_{\mathcal{Z}}, \mathbf{z}_j \in \mathbb{R}^m\}_{j=1}^{|T|}$ that need to be classified. Note that we assume samples \mathbf{x}_i and \mathbf{z}_j to lie in the same m -dimensional feature space \mathbb{R}^m , hence, we assume that $p_{\mathcal{X}}$ and $p_{\mathcal{Z}}$ are distributions over the same space. For brevity, we occasionally adopt the notation $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_{|S|}]$, $\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_{|T|}]$, and $\mathbf{y} = [y_1, \dots, y_{|S|}]$.

3.2 Target Risk

We adopt an empirical risk minimization (ERM) framework for constructing our domain-adapted classifier. The ERM framework proposes a classification function $h: \mathbb{R}^m \rightarrow \mathbb{R}$ and assesses the quality of the hypothesis by comparing its predictions with the true labels on the empirical data using a loss function $L: Y \times \mathbb{R} \rightarrow \mathbb{R}_0^+$. The empirical loss is an estimate of the risk, which is defined as the expected value of the loss function under the data distribution. Below, we show that if the target domain carries no additional information about the label distribution, the risk of a model on the target domain is equivalent to the risk on the source domain under a particular transfer distribution.

We first note that the joint source data, target data and label distribution can be decomposed into two conditional distributions and one marginal source distribution; $p_{\mathcal{Y}, \mathcal{Z}, \mathcal{X}} = p_{\mathcal{Y} \mid \mathcal{Z}, \mathcal{X}} p_{\mathcal{Z} \mid \mathcal{X}} p_{\mathcal{X}}$. The first conditional $p_{\mathcal{Y} \mid \mathcal{Z}, \mathcal{X}}$ describes the full class-posterior distribution given both source and target distribution. Next, we introduce our main assumption: the labels are conditionally independent of the target domain given the source domain ($\mathcal{Y} \perp\!\!\!\perp \mathcal{Z} \mid \mathcal{X}$), which implies: $p_{\mathcal{Y} \mid \mathcal{Z}, \mathcal{X}} = p_{\mathcal{Y} \mid \mathcal{X}}$. In other words, we assume that we can construct an optimal target classifier if (1) we have access to infinitely many labeled source samples—we know $p_{\mathcal{Y} \mid \mathcal{X}}$ —and (2) we know the true domain transfer distribution $p_{\mathcal{Z} \mid \mathcal{X}}$. In this scenario, observing target labels $y_j \sim p_{\mathcal{Y} \mid \mathcal{Z}}$ does not provide us with any new information.

To illustrate our assumption, imagine a sentiment classification problem. If people frequently use the word “nice” in positive reviews about electronics products (the source domain) and we know that electronics and kitchen products (the target domain) are very similar, then we assume that the word “nice” is not predictive of negative reviews of kitchen appliances. In other words, knowing that “nice” is predictive of a positive review and knowing that the domains are similar, it cannot be the case that “nice” is suddenly predictive of a negative review. Under this assumption, learning a good model for the target domain amounts to transferring the source domain to the target domain (that is, altering the marginal probability of observing the word “nice”) and learning a good predictive model on the resulting transferred source domain. Admittedly, there are scenarios in which our assumption is invalid: if people like “small” electronics but dislike “small” cars, the assumption is violated and our domain-adaptation approach will likely work less well. We do note, however, that our assumption is less stringent than the covariate-shift assumption, which assumes that the posterior distribution over classes is identical in the source and the target domain (i.e. that $p_{\mathcal{Y} \mid \mathcal{X}} = p_{\mathcal{Y} \mid \mathcal{Z}}$): the covariate-shift assumption does not facilitate the use of a transfer distribution $p_{\mathcal{Z} \mid \mathcal{X}}$.

Formally, we start by rewriting the risk R_Z on the target domain as follows:

$$\begin{aligned} R_Z(h) &= \int_Z \sum_{y \in \mathcal{Y}} L(y, h(\mathbf{z})) p_{\mathcal{Y}, \mathcal{Z}}(y, \mathbf{z}) \, d\mathbf{z} \\ &= \int_Z \sum_{y \in \mathcal{Y}} \int_{\mathcal{X}} L(y, h(\mathbf{z})) p_{\mathcal{Y}, \mathcal{Z}, \mathcal{X}}(y, \mathbf{z}, \mathbf{x}) \, d\mathbf{x} \, d\mathbf{z} \\ &= \int_Z \sum_{y \in \mathcal{Y}} \int_{\mathcal{X}} L(y, h(\mathbf{z})) p_{\mathcal{Y} | \mathcal{Z}, \mathcal{X}}(y | \mathbf{z}, \mathbf{x}) p_{\mathcal{Z} | \mathcal{X}}(\mathbf{z} | \mathbf{x}) p_{\mathcal{X}}(\mathbf{x}) \, d\mathbf{x} \, d\mathbf{z}. \end{aligned}$$

Using the assumption $p_{\mathcal{Y} | \mathcal{Z}, \mathcal{X}} = p_{\mathcal{Y} | \mathcal{X}}$ (or equivalently, $\mathcal{Y} \perp\!\!\!\perp \mathcal{Z} | \mathcal{X}$) introduced above, we can rewrite this expression as:

$$\begin{aligned} R_Z(h) &= \int_Z \sum_{y \in \mathcal{Y}} \int_{\mathcal{X}} L(y, h(\mathbf{z})) p_{\mathcal{Y} | \mathcal{X}}(y | \mathbf{x}) p_{\mathcal{Z} | \mathcal{X}}(\mathbf{z} | \mathbf{x}) p_{\mathcal{X}}(\mathbf{x}) \, d\mathbf{x} \, d\mathbf{z} \\ &= \int_Z \mathbb{E}_{\mathcal{Y}, \mathcal{X}}[L(y, h(\mathbf{z})) p_{\mathcal{Z} | \mathcal{X}}(\mathbf{z} | \mathbf{x})] \, d\mathbf{z}. \end{aligned}$$

Next, we replace the target risk $R_Z(h)$ with its empirical estimate $\hat{R}_Z(h | S)$ by plugging in source data S for the source joint distribution $p_{\mathcal{Y}, \mathcal{X}}$:

$$\begin{aligned} \hat{R}_Z(h | S) &= \frac{1}{|S|} \int_Z \sum_{(\mathbf{x}_i, y_i) \in S} L(y_i, h(\mathbf{z})) p_{\mathcal{Z} | \mathcal{X}}(\mathbf{z} | \mathbf{x} = \mathbf{x}_i) \, d\mathbf{z} \\ &= \frac{1}{|S|} \sum_{(\mathbf{x}_i, y_i) \in S} \mathbb{E}_{\mathcal{Z} | \mathcal{X} = \mathbf{x}_i}[L(y_i, h(\mathbf{z}))]. \end{aligned} \quad (1)$$

Feature-level domain adaptation (FLDA) trains classifiers by constructing a parametric model of the transfer distribution $p_{\mathcal{Z} | \mathcal{X}}$ and, subsequently, minimizing the expected loss in Equation 1 on the source data with respect to the parameters of the classifier. For linear classifiers, the expected loss in Equation 1 can be computed analytically for quadratic and exponential losses if the transfer distribution factorizes over dimensions and is in the natural exponential family: for the logistic and hinge losses, it can be upper-bounded or approximated efficiently under the same assumptions (Van der Maaten et al., 2013; Wager et al., 2013; Chen et al., 2014). Note that no observed target samples \mathbf{z}_j are involved in Equation 1: the expectation is over the transfer model $p_{\mathcal{Z} | \mathcal{X}}$, conditioned on a particular sample \mathbf{x}_i . The target data is only used to estimate the parameters of the transfer model.

3.3 Transfer Model

The transfer distribution $p_{\mathcal{Z} | \mathcal{X}}$ describes the relation between the source and the target domain: given a particular source sample, it produces a distribution of which target samples are likely to be observed (with the same label). The transfer distribution is modeled by selecting a parametric distribution and learning the parameters of this distribution from the source and target data (without looking at the source labels). Prior knowledge on the relation between source and target domain may be incorporated in the model via the choice

for a particular family of distributions. For instance, if we know that the main variation between two domains consists of particular words that are frequently used in one domain (say, news articles) but infrequently in another domain (say, tweets), then we choose a distribution that alters the relative frequency of words.

Given a model of the transfer distribution $p_{\mathcal{Z} | \mathcal{X}}$ and a model of the source distribution $p_{\mathcal{X}}$, we can work out the marginal distribution over the target domain as

$$q_{\mathcal{Z}}(\mathbf{z} | \theta, \eta) = \int_{\mathcal{X}} p_{\mathcal{Z} | \mathcal{X}}(\mathbf{z} | \mathbf{x}, \theta) p_{\mathcal{X}}(\mathbf{x} | \eta) \, d\mathbf{x}, \quad (2)$$

where θ represents the parameters of the transfer model, and η the parameters of the source model. We learn these parameters separately: first, we learn η by maximizing the likelihood of the source data under the model $p_{\mathcal{X}}(\mathbf{x} | \eta)$ and, subsequently, we learn θ by maximizing the likelihood of the target data under the compound model $q_{\mathcal{Z}}(\mathbf{z} | \theta, \eta)$. Hence, we first estimate the value of η by solving:

$$\hat{\eta} = \arg \max_{\eta} \sum_{\mathbf{x}_i \in T} \log p_{\mathcal{X}}(\mathbf{x}_i | \eta).$$

Subsequently, we estimate the value of θ by solving:

$$\hat{\theta} = \arg \max_{\theta} \sum_{\mathbf{z}_j \in T} \log q_{\mathcal{Z}}(\mathbf{z}_j | \theta, \hat{\eta}). \quad (3)$$

In this paper, we focus primarily on domain-adaptation problems involving binary and count features. In such problems, we wish to encode changes in the marginal likelihood of observing non-zero values in the transfer model. To this end, we employ a dropout distribution as transfer model that can model domain-shifts in which a feature occurs less often in the target domain than in the source domain. Learning a FLDA model with a dropout transfer model has the effect of strongly regularizing weights on features that occur infrequently in the target domain.

3.3.1 DROPOUT TRANSFER

To define our transfer model for binary or count features, we first set up a model that describes the likelihood of observing non-zero features in the source data. This model comprises a product of independent Bernoulli distributions:

$$p_{\mathcal{X}}(\mathbf{x}_i | \eta) = \prod_{d=1}^m \mathbb{1}_{x_{i,d} \neq 0} \eta_d (1 - \eta_d)^{1 - \mathbb{1}_{x_{i,d} \neq 0}}, \quad (4)$$

where $\mathbb{1}$ is the indicator function and η_d is the success probability (probability of non-zero values) of feature d . For this model, the maximum likelihood estimate of η_d is simply the sample average: $\hat{\eta}_d = |S|^{-1} \sum_{\mathbf{x}_i \in S} \mathbb{1}_{x_{i,d} \neq 0}$.

Next, we define a transfer model that describes how often a feature has a value of zero in the target domain when it has a non-zero value in the source domain. We assume an

unbiased dropout distribution (Wager et al., 2013; Rostamizadeh et al., 2011) that sets an observed feature in the source domain to zero in the target domain with probability θ_d :

$$p_{Z|X}(z_{-d} | x = x_{id}, \theta_d) = \begin{cases} \theta_d & \text{if } z_{-d} = 0 \\ 1 - \theta_d & \text{if } z_{-d} = x_{id} / (1 - \theta_d) \end{cases}, \quad (5)$$

where $\forall d : 0 \leq \theta_d \leq 1$, the subscript of z_{-d} denotes the d -th feature for any target sample, and where the outcome of not *dropping out* is scaled by a factor $1/(1 - \theta_d)$ in order to center the dropout distribution on the particular source sample. We assume the transfer distribution factorizes over features to obtain: $p_{Z|X}(\mathbf{z} | \mathbf{x} = \mathbf{x}_i, \theta) = \prod_d^m p_{Z|X}(z_{-d} | x_{-d} = x_{id}, \theta_d)$. The equation above defines a transfer distribution for every source sample. We obtain our final transfer model by sharing the parameters θ between all transfer distributions and averaging over all source samples.

To compute the maximum likelihood estimate of θ , the dropout transfer model from Equation 5 and the source model from Equation 4 are plugged into Equation 2 to obtain (see Appendix A for details):

$$\begin{aligned} q_Z(\boldsymbol{\theta} | \eta) &= \prod_{d=1}^m \int_{\mathcal{X}} p_{Z|X}(z_{-d} | x_{-d}, \theta_d) p_X(x_{-d} | \eta_d) dx_{-d} \\ &= \prod_{d=1}^m \left((1 - \theta_d) \eta_d \right)^{\mathbb{1}_{z_{-d} \neq 0}} \left(1 - (1 - \theta_d) \eta_d \right)^{1 - \mathbb{1}_{z_{-d} \neq 0}}. \end{aligned} \quad (6)$$

Plugging this expression into Equation 3 and maximizing with respect to θ , we obtain:

$$\hat{\theta}_d = \max\{0, 1 - \frac{\hat{\zeta}_d}{\hat{\eta}_d}\},$$

where $\hat{\zeta}_d$ is the sample average of the dichotomized target samples, $|T|^{-1} \sum_{z_d \in T} \mathbb{1}_{z_d \neq 0}$, and where $\hat{\eta}_d$ is the sample average of the dichotomized source samples, $|S|^{-1} \sum_{\mathbf{x}_i \in S} \mathbb{1}_{x_{id} \neq 0}$. We note that our particular choice for the transfer model cannot represent rate changes in the values of non-zero count features, such as whether a word is used on average 10 times in a document versus used on average only 3 times. The only variation that our dropout distribution captures is the variation in whether or not a feature occurs ($z_{-d} \neq 0$).

Because our dropout transfer model factorizes over features and is in the natural exponential family, the expectation in Equation 1 can be analytically computed. In particular, for a transfer distribution conditioned on source sample \mathbf{x}_i , its mean and variance are:

$$\begin{aligned} \mathbb{E}_{Z|X}[\mathbf{z}] &= \mathbf{x}_i \\ \mathbb{V}_{Z|X}[\mathbf{z}] &= \text{diag}\left(\frac{\theta}{1 - \theta}\right) \circ \mathbf{x}_i \mathbf{x}_i^\top, \end{aligned}$$

where \circ denotes the element-wise product of two matrices and we use the shorthand notation $\mathbb{E}_{Z|X=\mathbf{x}_i} = \mathbb{E}_{Z|X}$. The variance is a diagonal matrix due to our assumption of independent transfer distributions per feature. We will use these expressions below in our description of how to learn the parameters of the domain-adapted classifiers.

3.4 Classification

In order to perform classification with the risk formulation in Equation 1, we need to select a loss function L . Popular choices for the loss function include the quadratic loss (used in least-squares classification), the exponential loss (used in boosting), the hinge loss (used in support vector machines) and the logistic loss (used in logistic regression). The formulation in (1) has been studied before in the context of *dropout training* for the quadratic, exponential, and logistic loss by Wäger et al. (2013); Van der Maaten et al. (2013), and for hinge loss by Chen et al. (2014). In this paper, we focus on the quadratic and logistic loss functions, but we note that the FLDA approach can also be used in combination with exponential and hinge losses.

3.4.1 QUADRATIC LOSS

Assuming binary labels $Y = \{-1, +1\}$, a linear classifier $h(\mathbf{z}) = \mathbf{w}^\top \mathbf{z}$ parametrized by \mathbf{w} , and a quadratic loss function $L = (\mathbf{y} - \mathbf{w}^\top \mathbf{z})^2$, the expectation in Equation 1 can be expressed as:

$$\begin{aligned} \hat{R}_Z(\mathbf{w} | S) &= \sum_{(\mathbf{x}_i, y_i) \in S} \mathbb{E}_{Z|X} [y_i - \mathbf{w}^\top \mathbf{z}]^2 \\ &= \mathbf{y}^\top \mathbf{y} - 2 \mathbf{w}^\top \mathbb{E}_{Z|X}[\mathbf{z}] \mathbf{y} + \mathbf{w}^\top (\mathbb{E}_{Z|X}[\mathbf{z}] \mathbb{E}_{Z|X}[\mathbf{z}]^\top + \mathbb{V}_{Z|X}[\mathbf{z}]) \mathbf{w}, \end{aligned}$$

in which the data is augmented with $\mathbf{1}$ to capture the intercept, and in which we denote the $(m+1) \times |S|$ matrix of expectations as $\mathbb{E}_{Z|X=\mathbf{x}}[\mathbf{Z}] = [\mathbb{E}_{Z|X=\mathbf{x}_1}[\mathbf{z}], \dots, \mathbb{E}_{Z|X=\mathbf{x}_{|S|}}[\mathbf{z}]]$ and the $(m+1) \times (m+1)$ diagonal matrix of variances as $\mathbb{V}_{Z|X=\mathbf{x}}[\mathbf{Z}] = \sum_{\mathbf{x}_i \in S} \mathbb{V}_{Z|X=\mathbf{x}_i}[\mathbf{z}]$. Deriving the gradient for this loss function and setting it to zero yields the following closed-form solution for the classifier weights:

$$\mathbf{w} = \left(\mathbb{E}_{Z|X}[\mathbf{z}] \mathbb{E}_{Z|X}[\mathbf{z}]^\top + \mathbb{V}_{Z|X}[\mathbf{z}] \right)^{-1} \mathbb{E}_{Z|X}[\mathbf{z}] \mathbf{y}^\top. \quad (7)$$

In the case of a multi-class problem, i.e. $Y = \{1, \dots, K\}$, K predictors can be built in an one-vs-all fashion or $K(K-1)/2$ predictors in an one-vs-one fashion.

The solution in Equation 7 is very similar to the solution of a standard ridge regression model: $\mathbf{w} = (\mathbf{X}\mathbf{X}^\top + \lambda \mathbf{I})^{-1} \mathbf{X}\mathbf{y}^\top$. The main difference is that, in a standard ridge regressor, the regularization is independent of the data. By contrast, the regularization on the weights of the FLDA solution is determined by the variance of the transfer model: hence, it is different for each dimension and it depends on the transfer from source to target domain. Algorithm 1 summarizes the training of a binary quadratic-loss FLDA classifier with dropout transfer.

Algorithm 1 Binary FLDA with dropout transfer model and quadratic loss function.

```

procedure FLDA-Q( $S, T$ )
  for  $d=1, \dots, m$  do
     $\hat{\eta}_d = |S|^{-1} \sum_{\mathbf{x}_i \in S} \mathbb{1}_{x_i, d \neq 0}$ 
     $\hat{\zeta}_d = |T|^{-1} \sum_{\mathbf{x}_j \in T} \mathbb{1}_{x_j, d \neq 0}$ 
     $\theta_d = \max \{0, 1 - \hat{\zeta}_d / \hat{\eta}_d\}$ 
  end for
   $\mathbf{w} = (\mathbf{X}\mathbf{X}^\top + \text{diag}(\frac{\theta}{1-\theta})) \circ \mathbf{X}\mathbf{X}^\top)^{-1} \mathbf{X}\mathbf{Y}^\top$ 
  return  $\text{sign}(\mathbf{w}^\top \mathbf{Z})$ 
end procedure

```

▷ ◦ Element-wise product

3.4.2 LOGISTIC LOSS

Following the same choice of labels and hypothesis class as for the quadratic loss, the logistic version can be expressed as:

$$\begin{aligned}
 \hat{R}_Z(\mathbf{w} | S) &= \frac{1}{|S|} \sum_{(\mathbf{x}_i, y_i) \in S} \mathbb{E}_Z | \mathbf{x}_i | [-y_i \mathbf{w}^\top \mathbf{z} + \log \sum_{y' \in Y} \exp(y' \mathbf{w}^\top \mathbf{z})] \\
 &= \frac{1}{|S|} \sum_{(\mathbf{x}_i, y_i) \in S} -y_i \mathbf{w}^\top \mathbb{E}_Z | \mathbf{x}_i | \mathbf{z} + \mathbb{E}_Z | \mathbf{x}_i | \left[\log \sum_{y' \in Y} \exp(y' \mathbf{w}^\top \mathbf{z}) \right].
 \end{aligned} \tag{8}$$

This is a convex function in \mathbf{w} because the log-sum-exp of an affine function is convex, and because the expected value of a convex function is convex. However, the expectation cannot be computed analytically. Following Wäger et al. (2013), we approximate the expectation of the log-partition function, $A(\mathbf{w}^\top \mathbf{z}) = \log \sum_{y' \in Y} \exp(y' \mathbf{w}^\top \mathbf{z})$, using a Taylor expansion around the value $a_i = \mathbf{w}^\top \mathbf{x}_i$:

$$\begin{aligned}
 \mathbb{E}_Z | \mathbf{x}_i | [A(\mathbf{w}^\top \mathbf{z})] &\approx A(a_i) + A'(a_i) (\mathbb{E}_Z | \mathbf{x}_i | [\mathbf{w}^\top \mathbf{z}] - a_i) + \frac{1}{2} A''(a_i) (\mathbb{E}_Z | \mathbf{x}_i | [\mathbf{w}^\top \mathbf{z}] - a_i)^2 \\
 &= \text{const} + \sigma(-2\mathbf{w}^\top \mathbf{x}_i) \sigma(2\mathbf{w}^\top \mathbf{x}_i) \mathbf{w}^\top \mathbb{V}_Z | \mathbf{x}_i | \mathbf{z} | \mathbf{w},
 \end{aligned} \tag{9}$$

where $\sigma(x) = 1/(1 + \exp(-x))$ is the sigmoid function. In the Taylor approximation, the first-order term disappears because we chose an unbiased transfer model: $\mathbb{E}_Z | \mathbf{x}_i | [\mathbf{w}^\top \mathbf{z}] = \mathbf{w}^\top \mathbf{x}_i$. The approximation cannot be minimized in closed-form: we repeatedly take steps in the direction of its gradient in order to minimize the approximation of the risk in Equation 8, as described in Algorithm 2 (see Appendix B for the gradient derivation). The algorithm can be readily extended to multi-class problems by replacing \mathbf{w} by a $(m+1) \times K$ matrix and using an one-hot encoding for the labels (see Appendix C).

4. Experiments

In our experiments, we first study the empirical behavior of FLDA on artificial data for which we know the true transfer distribution. Following that, we measure the performance of our method in a “missing data at test time” scenario, as well as on two image data sets and three text data sets with varying amounts of domain transfer.

Algorithm 2 Binary FLDA with dropout transfer model and logistic loss function.

```

procedure FLDA-L( $S, T$ )
  for  $d=1, \dots, m$  do
     $\hat{\eta}_d = |S|^{-1} \sum_{\mathbf{x}_i \in S} \mathbb{1}_{x_i, d \neq 0}$ 
     $\hat{\zeta}_d = |T|^{-1} \sum_{\mathbf{x}_j \in T} \mathbb{1}_{x_j, d \neq 0}$ 
     $\theta_d = \max \{0, 1 - \hat{\zeta}_d / \hat{\eta}_d\}$ 
  end for
   $\mathbf{w} = \arg \min_{\mathbf{w}} - \sum_{(\mathbf{x}_i, y_i) \in S} [y_i \mathbf{w}^\top \mathbf{x}_i]$ 
    +  $\mathbf{w}'^\top (\sum_{(\mathbf{x}_i, y_i) \in S} [\sigma(-2\mathbf{w}^\top \mathbf{x}_i) \sigma(2\mathbf{w}^\top \mathbf{x}_i) \text{diag}(\frac{\theta}{1-\theta}) \mathbf{x}_i \mathbf{x}_i^\top]) \mathbf{w}'$ 
  return  $\text{sign}(\mathbf{w}^\top \mathbf{Z})$ 
end procedure

```

4.1 Artificial Data

We first investigate the behavior of FLDA on a problem in which the model assumptions are satisfied. We create such a problem setting by first sampling a source domain data set from known class-conditional distributions. Subsequently, we construct a target domain data set by sampling additional source data and transforming it using a pre-defined (dropout) transfer model.

4.1.1 ADAPTATION UNDER CORRECT MODEL ASSUMPTIONS

We perform experiments in which the domain-adapted classifier estimates the transfer model and trains on the source data; we evaluate the quality of the resulting classifier by comparing it to an oracle classifier that was trained on the target data (that is, the classifier one would train if labels for the target data were available at training time).

In the first experiment, we generate binary features by drawing 100,000 samples from two bivariate Bernoulli distributions. The marginal distributions are $[0.7 \ 0.7]$ for class one and $[0.3 \ 0.3]$ for class two. The source data is transformed to the target data using a dropout transfer model with parameters $\theta = [0.5 \ 0]$. This means that 50% of the values for feature 1 are set to 0 and the other values are scaled by $1/(1 - 0.5)$. For reference, two naive least-squares classifiers are trained, one on the labeled source data (S-LS) and one on the labeled target data (T-LS), and compared to FLDA-Q. S-LS achieves a misclassification error of 0.40 while T-LS and FLDA-Q achieve an error of 0.30. This experiment is repeated for the same classifiers but with logistic losses: a source logistic classifier (S-LR), a target logistic classifier (T-LR) and FLDA-L. In this experiment, S-LR again achieves an error of 0.40 and T-LR and FLDA-L an error of 0.30. Figure 1 shows the decision boundaries for the quadratic loss classifiers on the left and the logistic loss classifiers on the right. The figure shows that for both loss functions, FLDA has completely adapted to be equivalent to the target classifier in this artificial problem.

In the second experiment, we generate count features by sampling from bivariate Poisson distributions. Herein, we used rate parameters $\lambda = [2 \ 2]$ for the first class and $\lambda = [6 \ 6]$ for the second class. Again, we construct the target domain data by generating new samples

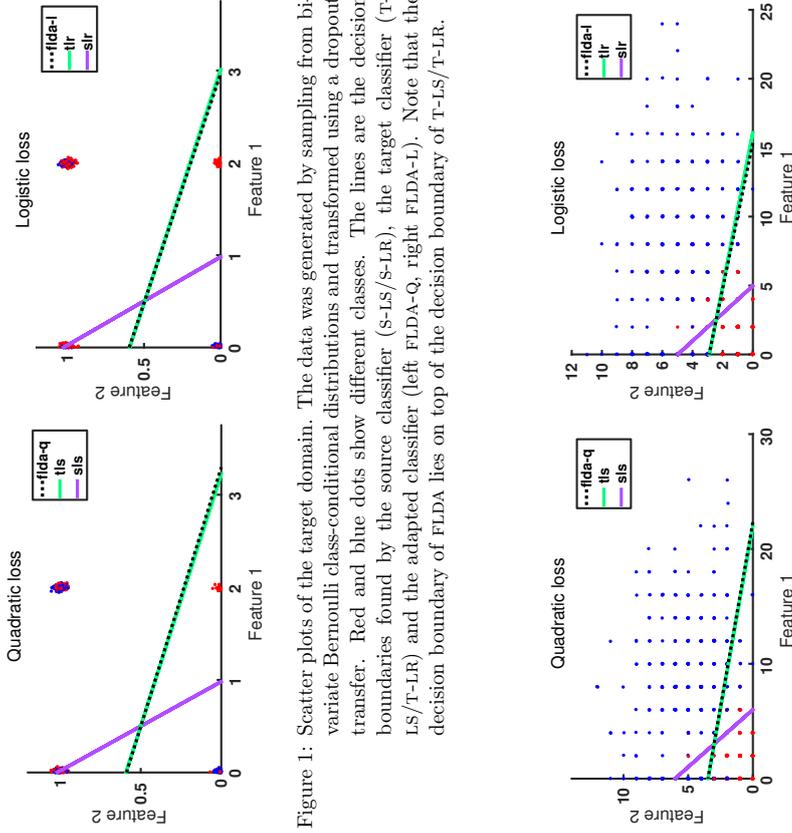


Figure 1: Scatter plots of the target domain. The data was generated by sampling from bivariate Bernoulli class-conditional distributions and transformed using a dropout transfer. Red and blue dots show different classes. The lines are the decision boundaries found by the source classifier (S-LS/S-LR), the target classifier (T-LS/T-LR) and the adapted classifier (left FLDA-Q, right FLDA-L). Note that the decision boundary of FLDA lies on top of the decision boundary of T-LS/T-LR.

Figure 2: Scatter plots of the target domain with decision boundaries of classifiers. The data was generated by sampling from bivariate Poisson class-conditional distributions and the decision boundaries were constructed using the source classifier (S-LS/S-LR), the target classifier (T-LS/T-LR), and the adapted classifiers (left FLDA-Q, right FLDA-L). Note that the decision boundary of FLDA lies on top of the decision boundary of T-LS / T-LR.

and dropping out the values of feature 1 with a probability of 0.5. In this experiment S-LS achieves an error of 0.181 and T-LS / FLDA-Q achieve an error of 0.099, while S-LR achieves an error of 0.170 and T-LR / FLDA-L achieve an error of 0.084. Figure 2 shows the decision boundaries of each of these classifiers and that FLDA has fully adapted to the domain shift.

4.1.2 LEARNING CURVES

One question that arises from the previous experiments is how many samples FLDA needs to estimate the transfer parameters and adapt to be (nearly) identical to the target classifier. To answer it, we performed an experiment in which we computed the classification error rate as a function of the number of training samples. The source training and validation data was generated from the same bivariate Poisson distributions as in Figure 2. The target data was constructed by generating additional source data and dropping out the first feature with a probability of 0.5. Each of the four data sets contained 10,000 samples. First, we trained a naive least-squares classifier on the source data (S-LS) and tested its performance on both the source validation and the target sets as a function of the number of source training samples. Second, we trained a naive least-squares classifier on the labeled target data (T-LS) and tested it on the source validation and another target validation set as a function of the number target training samples. Third, we trained an adapted classifier (FLDA-Q) on equal amounts of labeled source training data and unlabeled target training data and tested it on both the source validation and target validation sets. The experiment was repeated 50 times for every sample size to calculate the standard error of the mean.

The learning curves are plotted in Figure 3, which shows the classification error on the source validation set (top) and the classification error on the target validation (bottom). As expected, the source classifier (S-LS) outperforms the target (T-LS) and adapted (FLDA-Q) classifiers on the source domain (dotted lines), while FLDA-Q and T-LS outperform S-LS on the target domain (solid lines). In this problem, it appears that roughly 20 labeled source samples and 20 unlabeled target samples are sufficient for FLDA to adapt to the domain shift. Interestingly, FLDA-Q is outperforming S-LS and T-LS for small sample sizes. This is most likely due to the fact that the application of the transfer model is acting as a kind of regularization. In particular, when the learning curves are computed with ℓ_2 -regularized classifiers, then the difference in performance disappears.

4.1.3 ROBUSTNESS TO PARAMETER ESTIMATION ERRORS

Another question that arises is how sensitive the approach is to estimation errors in the transfer parameters. To answer this question, we performed an experiment in which we artificially introduce an error in the transfer parameters by perturbing them. As before, we generate 100,000 samples for both domains by sampling from bivariate Poisson distributions with $\lambda = [2 \ 2]$ for class 1 and $\lambda = [6 \ 6]$ for class 2. Again, the target domain is constructed by dropping out feature 1 with a probability of 0.5. We trained a naive classifier on the source data (S-LS), a naive classifier on the target data (T-LS), and an adapted classifier FLDA-Q with four different sets of parameters: the maximum likelihood estimate of the first transfer parameter $\hat{\theta}_1$ with an addition of 0, 0.1, 0.2, and 0.3. Table 1 shows the resulting classification errors, which reveal a relatively small effect of perturbing the estimated transfer parameters: the errors only increase by a few percent in this experiment.

To further illustrate the effect of the transfer parameters, Figure 4 shows the decision boundaries for the perturbed adapted classifiers. The figures show that the linear boundaries start to angle upwards when the error in the transfer parameter estimate increases. Overall, one could describe the effect of a dropout transfer model as steering the direction of the linear classifier. This experiment shows the importance of an accurate estimation of the

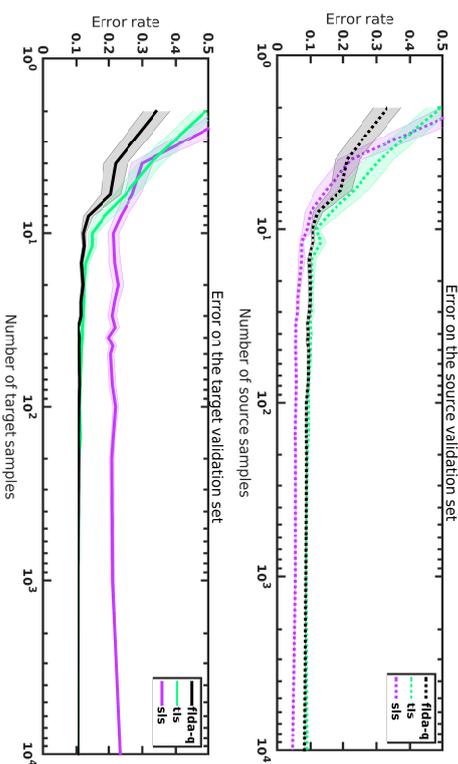


Figure 3: Learning curves of the source classifier (S-SL), the target classifier (T-TL), and adapted classifier (FLDA-Q). The top figure shows the error on a validation set generated from two bivariate Poisson distributions. The bottom figure shows the error on a validation set generated from two bivariate Poisson distributions with the first feature dropped out with a probability of 0.5.

	SL	TL	$\hat{\theta}_1 + 0$	$\hat{\theta}_1 + 0.1$	$\hat{\theta}_1 + 0.2$	$\hat{\theta}_1 + 0.3$
Quadratic loss	0.245	0.137	0.138	0.145	0.149	0.150
Logistic loss	0.264	0.139	0.139	0.140	0.142	0.146

Table 1: Classification errors for a naive source classifier, a naive target classifier, and the adapted classifier with a value of 0, 0.1, 0.2, and 0.3 added to the estimate of the first transfer parameter θ_1 .

transfer parameters to obtain high-quality adaptation. Nonetheless, our results do suggest that FLDA is robust to relatively small perturbations.

4.2 Natural Data

In a second set of experiments, we evaluate FLDA on a series of real-world data sets and compare it with several state-of-the-art methods. The evaluations are performed in the transductive learning setting: we measure the performance of the classifier on the already given, but unlabeled target samples.

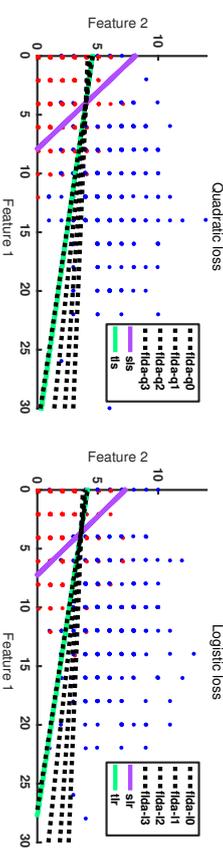


Figure 4: Scatter plots of the target data and decision boundaries of two naive and four adapted classifiers with transfer parameter estimate errors of 0, 0.1, 0.2, and 0.3. Results are shown for both the quadratic loss classifier (FLDA-Q; left) and the logistic loss classifier (FLDA-L; right).

As baselines, we consider eight alternative methods for domain adaptation. All of these employ a two-stage procedure. In the first step, sample weights, domain-invariant features, or a transformation of the feature space is estimated. In the second step, a classifier is trained using the results of the first stage. In all experiments, we estimate the hyperparameters, such as ℓ_2 -regularization parameters, via cross-validation on held-out source data. It should be noted that these values are optimal for generalizing to new source data but not necessarily for generalizing to the target domain (Sugiyama et al., 2007). Each of the eight baseline methods is described briefly below.

4.2.1 NAIVE SUPPORT VECTOR MACHINE (S-SVM)

Our first baseline method is a support vector machine trained on only the source samples and applied on the target samples. We made use of the libsvm package by Chang and Lin (2011) with a radial basis function kernel and we performed cross-validation to estimate the kernel bandwidth and the ℓ_2 -regularization parameter. All multi-class classification is done through an one-vs-one scheme. This method can be readily compared to subspace alignment (SA) and transfer component analysis (TCA) to evaluate the effects of the respective adaptation approaches.

4.2.2 NAIVE LOGISTIC REGRESSION (S-LR)

Our second baseline method is an ℓ_2 -regularized logistic regressor trained on only the source samples. Its main difference with the support vector machine is that it uses a linear model, a logistic loss instead of a hinge loss, and that it has a natural extension to multi-class as opposed to one-vs-one. The value of the regularization parameter was set via cross-validation. This method can be readily compared to kernel mean matching (KMM), structural correspondence learning (SCL), as well as to the logistic loss version of feature-level domain adaptation (FLDA-L).

4.2.3 KERNEL MEAN MATCHING (KMM)

Kernel mean matching (Huang et al., 2006; Gretton et al., 2009) finds importance weights by minimizing the maximum mean discrepancy (MMD) between the reweighted source samples and the target samples. To evaluate the empirical MMD, we used the radial basis function kernel. The weights are then incorporated in an importance weighted ℓ_2 -regularized logistic regressor.

4.2.4 STRUCTURAL CORRESPONDENCE LEARNING (SCL)

In order to build the domain-invariant subspace (Blitzer et al., 2006), the 20 features with the largest proportion of non-zero values in both domains are selected as the pivot features. Their values were dichotomized (1 if $x \neq 0$, 0 if $x = 0$) and predicted using a modified Huber loss (Ando and Zhang, 2005). The resulting classifier weight matrix was subjected to an eigenvalue decomposition and the eigenvectors with the 15 largest eigenvalues are retained. The source and target samples are both projected onto this basis and the resulting subspaces are added as features to the original source and target feature spaces, respectively. Consequently, classification is done by training an ℓ_2 -regularized logistic regressor on the augmented source samples and testing on the augmented target samples.

4.2.5 TRANSFER COMPONENT ANALYSIS (TCA)

For transfer component analysis, the closed-form solution to the parametric kernel map described in Pan et al. (2011) is computed using a radial basis function kernel. Its hyperparameters (kernel bandwidth, number of retained components and trade-off parameter μ) are estimated through cross-validation. After mapping the data onto the transfer components, we trained a support vector machine with an RBF kernel, cross-validating over its kernel bandwidth and the regularization parameter.

4.2.6 GEODESIC FLOW KERNEL (GFK)

The geodesic flow kernel is extracted based on the difference in angles between the principal components of the source and target samples (Gong et al., 2012). The basis functions of this kernel implicitly map the data onto all possible subspaces on the geodesic path between domains. Classification is performed using a kernel 1-nearest neighbor classifier. We used the subspace disagreement measure (SDM) to select an optimal value for the subspace dimensionality.

4.2.7 SUBSPACE ALIGNMENT (SA)

For subspace alignment (Fernando et al., 2013), all samples are normalized by their sum and all features are z-scored before extracting principal components. Subsequently, the Frobenius norm between the transformed source components and target components is minimized with respect to an affine transformation matrix. After projecting the source samples onto the transformed source components, a support vector machine with a radial basis function kernel is trained with cross-validated hyperparameters and tested on the target samples mapped onto the target components.

	hepat.	ozone	heart	mam.	auto.	arry.
Features	19	72	13	4	24	279
Samples	155	2534	704	961	205	452
Classes	2	2	2	2	6	13
Missing	75	685	615	130	72	384

Table 2: Summary statistics of the UCI repository data sets with missing data.

4.2.8 TARGET LOGISTIC REGRESSION (T-LR)

Finally, we trained a ℓ_2 -regularized logistic regressor using the normally unknown target labels as the oracle solution. This classifier is included to obtain an upper bound on the performance of our classifiers.

4.2.9 MISSING DATA AT TEST TIME

In this set of experiments, we study "missing data at test time" problems in which we argue that dropout transfer occurs naturally. Suppose that for the purposes of building a classifier, a data set is neatly collected with all features measured for all samples. At test time, however, some features could not be measured, due to for instance sensor failure, and the missing values are replaced by 0. This setting can be interpreted as two distributions over the same space with their transfer characterized by a relative increase in the number of 0 values, which our FLDA with dropout transfer is perfectly suited for. We have collected six data sets from the UCI machine learning repository (Lichman, 2013) with missing data: Hepatitis (hepat.), Ozone (ozone; Zhang and Fan, 2008), Heart Disease (heart; Detraano et al., 1989), Mammographic masses (mam.; Elter et al., 2007), Automobile (auto), and Arrhythmia (arry); Guvenir et al., 1997). Table 2 shows summary statistics for these sets. In the experiments, we construct the training set (source domain) by selecting all samples with no missing data, with the remainder as the test set (target domain). We note that instead of doing 0-imputation, we also could have used methods such as mean-imputation (Rubin, 1976; Little and Rubin, 2014). It is worth noting that the FLDA framework can adapt to such a setting by simply defining a different transfer model (one that replaces a feature value by its mean instead of a 0).

Table 3 reports the classification error rate of all domain-adaptation methods on the before-mentioned data sets. The lowest error rates for a particular data set are bold-faced. From the results presented in the table, we observe that whilst there appears to be little difference between the domains in the Hepatitis and Ozone data sets, there is substantial domain shift in the other data sets: the naive classifiers even perform at chance level on the Arrhythmia and Automobile data sets. On almost all data sets, both FLDA-Q and FLDA-L improve substantially over the S-LR, which suggests that they are successfully adapting to the missing data at test time. By contrast, most of the other domain-adaptation techniques do not consistently improve although, admittedly, sample transformation methods appear to work reasonable well on the Ozone, Mammography, and Arrhythmia data sets.

	S-SVM	S-LR	KMM	SCL	SA	CFK	TCA	FLDA-Q	FLDA-L	T-LR
hepat.	.213	.493	.347	.480	.253	.227	.213	.227	.200	.150
ozone	.060	.124	.126	.136	.047	.093	.140	.047	.079	.069
heart	.409	.338	.390	.319	.596	.362	.391	.203	.203	.177
mann.	.331	.462	.446	.462	.323	.423	.423	.462	.431	.194
auto.	.848	.935	.913	.935	.587	.323	.848	.848	.371	.371
arthy.	.930	.854	.620	.818	.414	.651	.930	.456	.889	.353

Table 3: Classification error rates on 6 UCI data sets with missing data. The data sets were partitioned into a training set (source domain), containing all samples with no missing features, and a test set (target domain), containing all samples with missing features.

4.2.10 HANDWRITTEN DIGITS

Handwritten digit data sets have been popular in machine learning due to the large sample size and the interpretability of the images. Generally, the data is acquired by assigning an integer value between 0 and 255 proportional to the amount of pressure that is applied at a particular spatial location on an electronic writing pad. Therefore, the probability of a non-zero value of a pixel informs us how often a pixel is part of a particular digit. For instance, the middle pixel in the digit 8 is a very important part of the digit because it nearly always corresponds to a high-pressure location, but the upper-left corner pixel is not used that often and is less important. Domain shift may be present between digit data sets due to differences in recording conditions. As a result, we may observe pixels that are discriminative in one data set (the source domain) that are hardly ever observed in another data set (the target domain). As a result, these pixels cannot be used to classify digits in the target domain, and we would like to inform the classifier that it should not assign a large weight to such pixels. We created a domain adaptation setting by considering two handwritten digit sets, namely MNIST (LeCun et al., 1998) and USPS (Hull, 1994). In order to create a common feature space, images from both data sets are resized to 16 by 16 pixels. To reduce the discrepancy between the size of MNIST data set (which contains 70,000 examples) and the USPS data set (which contains 9,298 examples), we only use 14,000 samples from the MNIST data set. The classes are balanced in both data sets.

Figure 5 shows a visualization of the probability that each pixel is non-zero for both data sets. The visualization shows that while the digits in the MNIST data set occupy mostly the center region, the USPS digits tend to occupy a substantially larger part of the image, specifically a center column. Figure 6 (left) visualizes the weights of the naive linear classifier (S-LR), (middle) the dropout probabilities θ , and (right) the adapted classifiers weights (FLDA-L). The middle image shows that dropout probabilities are large in regions where USPS pixels are frequent (the white pixels in Figure 5 right) but MNIST pixels are infrequent (the black pixels in Figure 5, left). The weights of the naive classifier appear to be shaped in a somewhat noisy circular pattern in the periphery, with the center containing negative weights (if these center pixels have a low intensity in a new sample, then the image is more likely to be a 0 digit). By contrast, the adapted weights of the FLDA classifier are

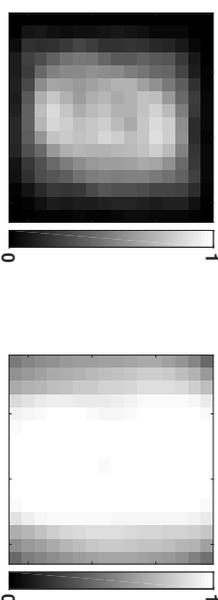


Figure 5: Visualization of the probability of non-zero values for each pixel on the MNIST data set (left) and the USPS data set (right).

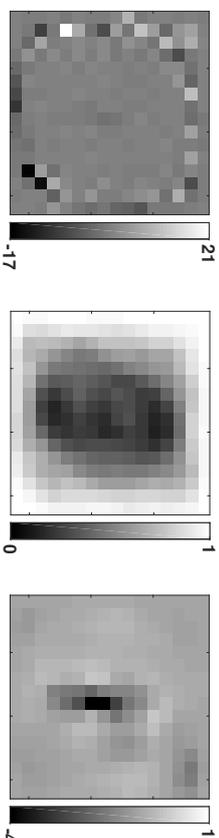


Figure 6: Weights assigned by the naive source classifier to the 0-digit predictor (left), the transfer parameters of the dropout transfer model (middle), and the weights assigned by the adapted classifier to the 0-digit predictor for training on USPS images and testing on MNIST (right; $U \rightarrow M$).

smoothed in the periphery, which indicates that the classifier is placing more value on the center pixels and is essentially ignoring the peripheral ones.

Table 4 shows the classification error rates where the rows correspond to the combinations of treating one data set as the source domain and the other as the target. The results show that there is a large difference between the domain-specific classifiers (S-LR and T-LR), which indicates that the domains are highly dissimilar. We note that the error rates of the target classifier on the MNIST data set are higher than usual for this data set (T-LR has an error rate of 0.234), which is because of the down-sampling of the images to 16×16 pixels and the smaller sample size. The results presented in the table highlight an interesting property of FLDA with dropout transfer: while FLDA performs well in settings in which the domain transfer can be appropriately modeled by the transfer distribution ($U \rightarrow M$ setting where pixels that appear in the USPS do not appear in MNIST), it does not perform well the other way around. The dropout transfer model does not capture pixels appearing *more often* instead of less often in the target domain. To work well in that setting, it is presumably necessary to use a richer transfer model.

	S-SVM	S-LR	KMM	SCL	SA	GFK	TCA	FLDA-Q	FLDA-L	T-LR
M→U	.522	.747	.748	.747	.890	.497	.808	.811	.678	.055
U→M	.766	.770	.769	.808	.757	.660	.857	.640	.684	.234

Table 4: Classification error rates obtained on both combinations of treating one domain as the source and the other as the target. M='MNIST' and U='USPS'.

	S-SVM	S-LR	KMM	SCL	SA	GFK	TCA	FLDA-Q	FLDA-L	T-LR
A→D	.599	.618	.616	.621	.627	.624	.624	.599	.624	.303
A→W	.688	.675	.668	.686	.606	.631	.712	.648	.678	.181
A→C	.557	.553	.563	.555	.594	.614	.579	.565	.550	.427
D→W	.312	.312	.346	.317	.167	.153	.295	.322	.312	.181
D→C	.744	.712	.734	.712	.655	.706	.680	.712	.710	.427
W→C	.721	.698	.709	.705	.677	.697	.688	.675	.701	.427
D→A	.876	.719	.727	.724	.616	.680	.650	.700	.722	.258
W→A	.676	.695	.706	.707	.631	.665	.668	.671	.691	.258
C→A	.493	.523	.515	.496	.538	.592	.504	.490	.475	.258
W→D	.198	.191	.178	.198	.214	.121	.166	.191	.185	.303
C→D	.612	.616	.631	.583	.575	.599	.612	.510	.599	.303
C→W	.712	.725	.729	.724	.600	.603	.695	.654	.702	.181

Table 5: Classification error rates obtained by ten (domain-adapted) classifiers for all pairwise combinations of domains on the Office-Caltech data set with SURF features (A='Amazon', D='DSLR', W='Webcam', and C='Caltech').

4.2.11 OFFICE-CALTECH

The Office-Caltech data set (Hoffman et al., 2013) consists of images of objects gathered using four different methods: one from images found through a web image search (referred to as 'C'), one from images of products on Amazon (A), one taken with a digital SLR camera (D) and one taken with a webcam (W). Overall, the set contains 10 classes, with 1123 samples from Caltech, 958 samples from Amazon, 157 samples from the DSLR camera, and 295 samples from the webcam. Our first experiment with the Office-Caltech data set is based on features extracted through SURF features (Bay et al., 2006). These descriptors determine a set of interest points by finding local maxima in the determinant of the image Hessian. Weighted sums of Haar features are computed in multiple sub-windows at various scales around each of the interest points. The resulting descriptors are vector-quantized to produce a bag-of-visual-words histogram of the image that is both scale and rotation-invariant. We perform domain-adaptation experiments by training on one domain and testing on another.

Table 5 shows the results of the classification experiments, where compared to competing methods, SA is performing well for a number of domain pairs, which may indicate that the SURF descriptor representation leads to domain dissimilarities that can be accurately

	S-SVM	S-LR	KMM	SCL	SA	GFK	TCA	FLDA-Q	FLDA-L	T-LR
A→D	.406	.388	.402	.422	.460	.424	.351	.428	.388	.104
A→W	.434	.468	.455	.474	.499	.477	.426	.491	.468	.064
D→W	.086	.079	.083	.074	.103	.073	.087	.088	.079	.064
D→A	.516	.496	.502	.497	.520	.569	.489	.589	.487	.216
W→A	.520	.496	.514	.506	.541	.584	.510	.645	.501	.216
W→D	.034	.030	.032	.034	.062	.052	.042	.024	.044	.104

Table 6: Classification error rates obtained by ten (domain-adapted) classifiers for all pairwise combinations of domains on the Office data set with DeCAF₈ features (A='Amazon', D='DSLR', and W='Webcam').

captured by subspace transformations. This result is further supported by the fact that the transformations found by GFK and TCA are also outperforming s-svm. FLDA-Q and FLDA-L are among the best performers on certain domain pairs. In general, FLDA does appear to perform at least as good or better than a naive s-LR classifier. The results on the Office-Caltech data set depend on the type of information the SURF descriptors are extracting from the images. We also studied the performance of domain-adaptation methods on a richer visual representation, produced by a pre-trained convolutional neural network. Specifically, we used a data set provided by Donahue et al., 2014, who extracted 1000-dimensional feature-layer activations (so-called DeCAF₈ features) in the upper layers of the a convolutional network that was pre-trained on the Imagenet data set. Donahue et al. (2014) used a larger superset of the Office-Caltech data set that contains 31 classes with 2817 images from Amazon, 498 from the DSLR camera, and 795 from the webcam. The results of our experiments with the DeCAF₈ features are presented in Table 6. The results show substantially lower error rates overall, but they also show that domain transfer in the DeCAF₈ feature representation is not amenable to effective modeling by subspace transformations. KMM and SCL obtain performances that are similar to the of the naive s-LR classifier but in one experiment, the naive classifier is actually the best-performing model. Whilst achieving the best performance on 2 out of 6 domain pairs, the FLDA-Q and FLDA-L models are not as effective as on other data sets, presumably, because dropout is not a good model for the transfer in a continuous feature space such as the DeCAF₈ feature space.

4.2.12 IMDB

The Internet Movie Database (IMDb) (Pang and Lee, 2004) contains written reviews of movies labeled with a 1-10 star rating, which we dichotomize by setting values > 5 as +1 and values ≤ 5 as -1. Using this dichotomy, both classes are roughly balanced. From the original bag-of-words representation, we selected only the features with more than 100 non-zero values in the entire data set, resulting in 4180 features. To obtain the domains, we split the data set by genre and obtained 3402 reviews of action movies, 1249 reviews of family movies, and 3697 reviews of war movies. We assume that people tend to use different words to review different genres of movies, and we are interested in predicting viewer sentiment after adapting to changes in the word frequencies. To visualize whether this assumption is

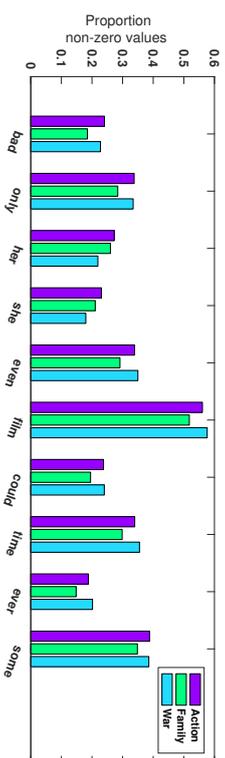


Figure 7: Proportion of non-zero values for a subset of words per domain on the IMDB data set.

	S-SVM	S-LR	KMM	SCL	SA	GFK	TCA	FLDA-Q	FLDA-L	T-LR
A→F	.145	.136	.133	.133	.184	.276	.230	.135	.136	.196
A→W	.158	.155	.155	.165	.163	.249	.266	.158	.154	.163
F→W	.256	.206	.208	.206	.182	.289	.355	.205	.202	.163
F→A	.201	.195	.193	.198	.193	.296	.363	.194	.194	.169
W→A	.168	.160	.159	.159	.167	.238	.222	.155	.157	.169
W→F	.340	.167	.163	.163	.232	.292	.203	.172	.159	.196

Table 7: Classification error rates obtained by ten (domain-adapted) classifiers for all pairwise combinations of domains on the IMDB data set. (A=‘Action’, F=‘Family’, and W=‘War’).

valid, we plot the proportion of non-zero values of 10 randomly chosen words per domain in Figure 7. The figure suggests that action movie and war movie reviews are quite similar, but the word use in family movie reviews does appear to be different.

Table 7 reports the results of the classification experiments on the IMDB database. The first thing to note is that the performances of S-LR and T-LR are quite similar, which suggests that the frequencies of discriminative words do not vary too much between genres. The results also show that GFK and TCA are not as effective on this data set as they were on the handwritten digits and Office-Caltech data sets, which suggests that finding a joint subspace that is still discriminative is hard, presumably, because only a small number of the 4180 words actually carry discriminative information. FLDA-Q and FLDA-L are better suited for such a scenario, which is reflected by their competitive performance on all domain pairs.

4.2.13 SPAM

Domain adaptation settings also arise in spam detection systems. For this experiment, we concatenated two data sets from the UCI machine learning repository: one containing 4205 emails from the Enron spam database (Kilint and Yang, 2004) and one containing 5338 text messages from the SMS-spam data set (Almeida et al., 2011). Both were represented using

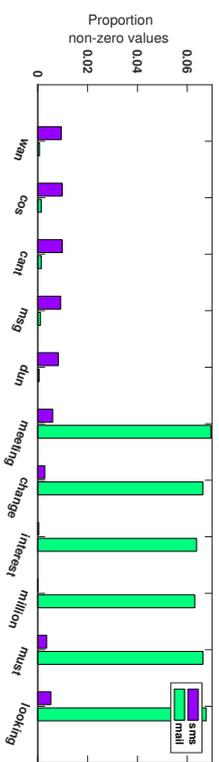


Figure 8: Proportion of non-zero values for a subset of words per domain on the spam data set.

	S-SVM	S-LR	KMM	SCL	SA	GFK	TCA	FLDA-Q	FLDA-L	T-LR
S→M	.460	.522	.521	.524	.445	.491	.508	.511	.521	.073
M→S	.830	.804	.799	.804	.408	.696	.863	.636	.727	.133

Table 8: Classification error rates obtained by ten (domain-adapted) classifiers for both domain pairs on the spam data set. (S=‘SMS’ and M=‘E-Mail’).

bag-of-words vectors over 4272 words that occurred in both data sets. Figure 8 shows the proportions of non-zero values for some example words, and shows that there exist large differences in word frequencies between the two domains. In particular, much of the domain differences are due to text messages using shortened words, whereas email messages tend to be more formal.

Table 8 shows results from our classification experiments on the spam data set. As can be seen from the results of T-LR, fairly good accuracies can be obtained on the spam detection task. However, the domains are so different that the naive classifiers S-SVM and S-LR are performing according to chance or worse. Most of the domain-adaptation models do not appear to improve much over the naive models. For KMM this makes sense, as the importance weight estimator will assign equal values to each sample when the empirical supports of the two domains are disjoint. There might be some features that are shared between domains, *i.e.*, words that are span in both emails and text messages, but considering the performance of SCL these might not be corresponding well with the other features. FLDA-Q and FLDA-L are showing slight improvements over the naive classifiers, but the transfer model we used is too poor as the domains contain a large amount of increased word frequencies.

4.2.14 AMAZON

We performed a similar experiment on the Amazon sentiment analysis data set of product reviews (Blitzer et al., 2007). The data consists of a 30,000 dimensional bag-of-words representations of 27,677 reviews with the labels derived from the dichotomized 5-star rating (ratings > 3 are +1 and ratings ≤ 3 as -1). Each review describes a product from one of four categories: books (6465 reviews), DVDs (5586 reviews), electronics (7681 reviews) and

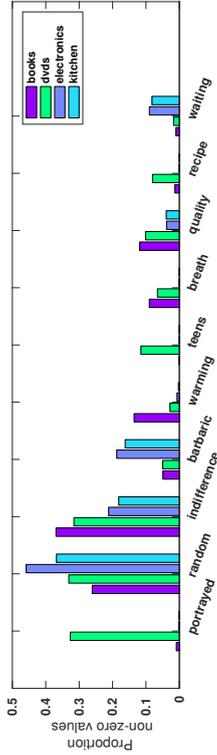


Figure 9: Proportion of non-zero values for a subset of words per domain on the Amazon data set.

kitchen appliances (7945 reviews). Figure 9 shows the probability of a non-zero value for some example words in each category. Some words, such as ‘portrayed’ or ‘barbaric’, are very specific to one or two domains, but the frequencies of many other words do not vary much between domains. We performed experiments on the Amazon data set using the same experimental setup as before.

In Table 9, we report the classification error rates on all pairwise combinations of domains. The difference in classification errors between S-LR and T-LR is up to 10%, which suggests there is potential for success with domain adaptation. However, the domain transfer is not captured well by SA, GFK, TCA: on average, these methods are performing worse than the naive classifiers. We presume this happens because only a small number of words are actually discriminative, and these words carry little weight in the sample transformation measures used. Furthermore, there are significantly less samples than features in each domain which means models with large amounts of parameters are likely to experience estimation errors. By contrast, FLDA-L performs strongly on the Amazon data set, achieving the best performance on many of the domain pairs. FLDA-Q performs substantially worse than FLDA-L, presumably, because of the singular covariance matrix and the fact that least-squares classifiers are very sensitive to outliers.

5. Discussion and Conclusions

We have presented an approach to domain adaptation, called FLDA, that fits a probabilistic model to capture the transfer between the source and the target data and, subsequently, trains a classifier by minimizing the expected loss on the source data under this transfer model. Whilst the FLDA approach is very general, in this paper, we have focused on one particular transfer model, namely, a dropout model. Our extensive experimental evaluation with this transfer model shows that FLDA performs on par with the current state-of-the-art methods for domain adaptation.

An interesting interpretation of our formulation is that the expected loss under the transfer model performs a kind of data-dependent regularization (Wager et al., 2013). For instance, if a quadratic loss function is employed in combination with a Gaussian transfer model, FLDA reduces to a transfer-dependent variant of ridge regression (Bishop, 1995). This transfer-dependent regularizer increases the amount of regularization on features when it

	S-SVM	S-LR	KMM	SCL	SA	GFK	TCA	FLDA-Q	FLDA-L	T-LR
B→D	.180	.168	.166	.167	.414	.392	.413	.303	.166	.153
B→E	.217	.221	.222	.220	.372	.429	.369	.343	.210	.116
B→K	.188	.188	.189	.184	.371	.443	.338	.384	.185	.095
D→E	.201	.202	.205	.207	.403	.480	.385	.369	.196	.116
D→K	.182	.182	.185	.190	.330	.494	.360	.379	.185	.095
E→K	.108	.110	.106	.112	.311	.416	.261	.308	.104	.095
D→B	.192	.190	.191	.202	.351	.388	.420	.368	.186	.145
E→B	.257	.262	.253	.260	.372	.445	.481	.406	.261	.145
K→B	.261	.277	.268	.273	.414	.418	.426	.399	.271	.145
E→D	.245	.240	.238	.242	.398	.441	.427	.384	.238	.153
K→D	.230	.230	.230	.231	.383	.410	.400	.370	.228	.153
K→E	.123	.131	.126	.126	.290	.353	.296	.292	.119	.116

Table 9: Classification error rates obtained by ten (domain-adapted) classifiers for all pairwise combinations of domains on the Amazon data set. (B=‘Books’, D=‘DVD’, E=‘Electronics’, and K=‘Kitchen’).

is undesired for the classifier to assign a large weight to that feature. In other words, the regularizer forces the classifier to ignore features that are frequently present in the source domain but very infrequently present in the target domain.

In some of our experiments, the adaptation strategies are producing classifiers that perform worse than a naive classifier trained on the source data. A potential reason for this is that many domain-adaptation models make strong assumptions on the data that are invalid in many real-world scenarios. In particular, it is unclear to what extent the relation between source data and classes ($\mathcal{X} \rightarrow \mathcal{Y}$) truly is informative about the target labels ($\mathcal{Z} \rightarrow \mathcal{Y}$). This issue arises in every domain-adaptation problem: without target labels, there is no way of knowing whether matching the source distribution $p_{\mathcal{X}}$ to the target distribution $p_{\mathcal{Z}}$ will improve the match between $p_{\mathcal{Y}|\mathcal{X}}$ and $p_{\mathcal{Y}|\mathcal{Z}}$.

Acknowledgments

This work was supported by the Netherlands Organization for Scientific Research (NWO; grant 612.001.301). The authors thank Simo Pan and Boqing Gong for insightful discussions.

Appendix A

For some combinations of source and target models, the source domain can be integrated out. For others, we would have to resort to Markov Chain Monte Carlo sampling and subsequently averaging the samples drawn from the *transferred* source distribution q_Z . For the Bernoulli and dropout distributions defined in Equations 4 and 5, the integration as in Equation 6 can be performed by plugging in the specified probabilities and performing the summation:

$$\begin{aligned}
q_Z(\mathbf{z} | \eta, \theta) &= \prod_{d=1}^m \int_{\mathcal{X}} p_Z | \mathcal{X}(z_{-d} | x_{-d}, \theta_d) p_{\mathcal{X}}(x_{-d} | \eta_d) dx_{-d} \\
&= \prod_{d=1}^m \sum_{\mathbb{1}_{x_{-d} \neq 0}} p_Z | \mathcal{X}(z_{-d} | \mathbb{1}_{x_{-d} \neq 0}, \theta_d) p_{\mathcal{X}}(\mathbb{1}_{x_{-d} \neq 0}; \eta_d) \\
&= \prod_{d=1}^m \begin{cases} \sum_{\mathbb{1}_{x_{-d} \neq 0}=0} p_Z | \mathcal{X}(z_{-d} = 0 | \mathbb{1}_{x_{-d} \neq 0}, \theta_d) p_{\mathcal{X}}(\mathbb{1}_{x_{-d} \neq 0}; \eta_d) & \text{if } z_{-d} = 0 \\ \sum_{\mathbb{1}_{x_{-d} \neq 0}=0} p_Z | \mathcal{X}(z_{-d} \neq 0 | \mathbb{1}_{x_{-d} \neq 0}, \theta_d) p_{\mathcal{X}}(\mathbb{1}_{x_{-d} \neq 0}; \eta_d) & \text{if } z_{-d} \neq 0 \end{cases} \\
&= \prod_{d=1}^m \begin{cases} 1(1 - \eta_d) + \theta_d \eta_d & \text{if } z_{-d} = 0 \\ 0(1 - \eta_d) + (1 - \theta_d) \eta_d & \text{if } z_{-d} \neq 0 \end{cases} \\
&= \prod_{d=1}^m (1 - \theta_d) \eta_d^{\mathbb{1}_{z_{-d} \neq 0}} (1 - (1 - \theta_d) \eta_d)^{1 - \mathbb{1}_{z_{-d} \neq 0}},
\end{aligned}$$

where the subscript of x_{-d} refers to the d -th feature of any sample x_{id} . Note that we chose our transfer model such that the probability is 0 for a non-zero target sample value given a zero source sample value; $p_Z | \mathcal{X}(z_{-d} \neq 0 | \mathbb{1}_{x_{-d} \neq 0} = 0, \theta_d) = 0$. In other words, if a word is not used in the source domain, then we expect that it is also not used in the target domain. By setting different values for these probabilities, one models different types of transfer.

Appendix B

The gradient to the second-order Taylor approximation of binary FLDA-L for a general transfer model is:

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{w}} \hat{R}_Z(h | S) &= \frac{1}{|S|} \sum_{(\mathbf{x}_i, y_i) \in S} -y_i \mathbb{E}_{Z | \mathbf{x}_i}[\mathbf{z}] + \sum_{y' \in \mathcal{Y}} y' \frac{\exp(y' \mathbf{w}^\top \mathbf{x}_i)}{\exp(y' \mathbf{w}^\top \mathbf{x}_i)} \mathbf{x}_i + \\
&\left[\left(1 - \left[\frac{\sum_{y' \in \mathcal{Y}} y' \exp(y' \mathbf{w}^\top \mathbf{x}_i)}{\sum_{y' \in \mathcal{Y}} \exp(y' \mathbf{w}^\top \mathbf{x}_i)} \right]^2 \right) \mathbf{w}^\top \mathbf{x}_i + \frac{\sum_{y' \in \mathcal{Y}} y' \exp(y' \mathbf{w}^\top \mathbf{x}_i)}{\sum_{y' \in \mathcal{Y}} \exp(y' \mathbf{w}^\top \mathbf{x}_i)} \left(\mathbb{E}_{Z | \mathbf{x}_i}[\mathbf{z}] - \mathbf{x}_i \right) \right. \\
&\quad \left. + 4\sigma \left(-2 \mathbf{w}^\top \mathbf{x}_i \right) \sigma \left(2 \mathbf{w}^\top \mathbf{x}_i \right) \left[\left(\sigma \left(-2 \mathbf{w}^\top \mathbf{x}_i \right) - \sigma \left(2 \mathbf{w}^\top \mathbf{x}_i \right) \right) \mathbf{w}^\top \mathbf{x}_i + 1 \right] \right) \\
&\mathbf{w}^\top \left(\mathbb{V}_{Z | \mathbf{x}_i}[\mathbf{z}] + \left(\mathbb{E}_{Z | \mathbf{x}_i}[\mathbf{z}] - \mathbf{x}_i \right) \left(\mathbb{E}_{Z | \mathbf{x}_i}[\mathbf{z}] - \mathbf{x}_i \right)^\top \right) \mathbf{w}.
\end{aligned}$$

Appendix C

The second-order Taylor approximation to the expectation over the log-partition function for a multi-class classifier weight matrix \mathbf{W} of size $(m+1) \times K$ around the point $a_i = \mathbf{W}^\top \mathbf{x}_i$ is:

$$\begin{aligned}
\mathbb{E}_{Z | \mathbf{x}_i} [A(\mathbf{W}^\top \mathbf{z})] &\approx A(a_i) + A'(a_i) (\mathbb{E}_{Z | \mathbf{x}_i}[\mathbf{W}^\top \mathbf{z}] - a_i) + \frac{1}{2} A''(a_i) (\mathbb{E}_{Z | \mathbf{x}_i}[\mathbf{W}^\top \mathbf{z}] - a_i)^2 \\
&= \log \sum_{k=1}^K \exp(\mathbf{W}_k^\top \mathbf{x}_i) + \sum_{k=1}^K \frac{\exp(\mathbf{W}_k^\top \mathbf{x}_i)}{\sum_{k=1}^K \exp(\mathbf{W}_k^\top \mathbf{x}_i)} \mathbf{W}_k^\top (\mathbb{E}_{Z | \mathbf{x}_i}[\mathbf{z}] - \mathbf{x}_i) \\
&\quad + \frac{1}{2} \sum_{k=1}^K \left(\frac{\exp(\mathbf{W}_k^\top \mathbf{x}_i)}{\sum_{k=1}^K \exp(\mathbf{W}_k^\top \mathbf{x}_i)} - \frac{\exp(2\mathbf{W}_k^\top \mathbf{x}_i)}{\left(\sum_{k=1}^K \exp(\mathbf{W}_k^\top \mathbf{x}_i) \right)^2} \right) \\
&\quad \mathbf{W}_k^\top \left(\mathbb{V}_{Z | \mathbf{x}_i}[\mathbf{z}] + \left(\mathbb{E}_{Z | \mathbf{x}_i}[\mathbf{z}] - \mathbf{x}_i \right) \left(\mathbb{E}_{Z | \mathbf{x}_i}[\mathbf{z}] - \mathbf{x}_i \right)^\top \right) \mathbf{W}_k.
\end{aligned}$$

The results contains a number of recurring terms which means it can be efficiently implemented. Incorporating the multi-class approximation into the loss, we can derive the following gradient:

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{W}_k} \hat{R}_Z(h | S) &= \frac{1}{|S|} \sum_{(\mathbf{x}_i, y_i) \in S} -y_i \mathbb{E}_{Z | \mathbf{x}_i}[\mathbf{z}] \\
&+ \left[\frac{\exp(\mathbf{W}_k^\top \mathbf{x}_i)}{\sum_{k=1}^K \exp(\mathbf{W}_k^\top \mathbf{x}_i)} - \frac{\exp(2\mathbf{W}_k^\top \mathbf{x}_i)}{\left(\sum_{k=1}^K \exp(\mathbf{W}_k^\top \mathbf{x}_i) \right)^2} \right] \mathbf{x}_i \mathbf{W}_k^\top (\mathbb{E}_{Z | \mathbf{x}_i}[\mathbf{z}] - \mathbf{x}_i) \\
&+ \frac{\exp(\mathbf{W}_k^\top \mathbf{x}_i) \mathbf{x}_i}{\sum_{k=1}^K \exp(\mathbf{W}_k^\top \mathbf{x}_i)} (\mathbb{E}_{Z | \mathbf{x}_i}[\mathbf{z}] - \mathbf{x}_i) \\
&+ \left[\frac{\exp(\mathbf{W}_k^\top \mathbf{x}_i) \mathbf{x}_i}{\sum_{k=1}^K \exp(\mathbf{W}_k^\top \mathbf{x}_i)} - 3 \frac{\exp(2\mathbf{W}_k^\top \mathbf{x}_i) \mathbf{x}_i}{\left(\sum_{k=1}^K \exp(\mathbf{W}_k^\top \mathbf{x}_i) \right)^2} + 2 \frac{\exp(3\mathbf{W}_k^\top \mathbf{x}_i) \mathbf{x}_i}{\left(\sum_{k=1}^K \exp(\mathbf{W}_k^\top \mathbf{x}_i) \right)^3} \right] \\
&\mathbf{W}_k^\top \left(\mathbb{V}_{Z | \mathbf{x}_i}[\mathbf{z}] + \left(\mathbb{E}_{Z | \mathbf{x}_i}[\mathbf{z}] - \mathbf{x}_i \right) \left(\mathbb{E}_{Z | \mathbf{x}_i}[\mathbf{z}] - \mathbf{x}_i \right)^\top \right) \mathbf{W}_k \\
&+ \left[\frac{2 \exp(\mathbf{W}_k^\top \mathbf{x}_i)}{\sum_{k=1}^K \exp(\mathbf{W}_k^\top \mathbf{x}_i)} - \frac{2 \exp(2\mathbf{W}_k^\top \mathbf{x}_i)}{\sum_{k=1}^K \exp(\mathbf{W}_k^\top \mathbf{x}_i)^2} \right] (\mathbb{V}_{Z | \mathbf{x}_i}[\mathbf{z}] + \left(\mathbb{E}_{Z | \mathbf{x}_i}[\mathbf{z}] - \mathbf{x}_i \right) (\mathbb{E}_{Z | \mathbf{x}_i}[\mathbf{z}] - \mathbf{x}_i)^\top) \mathbf{W}_k \\
&- \frac{\exp(\mathbf{W}_k^\top \mathbf{x}_i) \mathbf{x}_i}{\left(\sum_{k=1}^K \exp(\mathbf{W}_k^\top \mathbf{x}_i) \right)^2} \sum_{j \neq k}^K \exp(\mathbf{W}_j^\top \mathbf{x}_i) \mathbf{W}_j^\top (\mathbb{V}_{Z | \mathbf{x}_i}[\mathbf{z}] + \left(\mathbb{E}_{Z | \mathbf{x}_i}[\mathbf{z}] - \mathbf{x}_i \right) (\mathbb{E}_{Z | \mathbf{x}_i}[\mathbf{z}] - \mathbf{x}_i)^\top) \mathbf{W}_j \\
&+ \frac{2 \exp(\mathbf{W}_k^\top \mathbf{x}_i) \mathbf{x}_i}{\left(\sum_{k=1}^K \exp(\mathbf{W}_k^\top \mathbf{x}_i) \right)^3} \sum_{j \neq k}^K \exp(2\mathbf{W}_j^\top \mathbf{x}_i) \mathbf{W}_j^\top (\mathbb{V}_{Z | \mathbf{x}_i}[\mathbf{z}] + \left(\mathbb{E}_{Z | \mathbf{x}_i}[\mathbf{z}] - \mathbf{x}_i \right) (\mathbb{E}_{Z | \mathbf{x}_i}[\mathbf{z}] - \mathbf{x}_i)^\top) \mathbf{W}_j.
\end{aligned}$$

References

- TA Almeida, JMG Hidalgo, and A Yamakami. Contributions to the study of sms spam filtering: new collection and results. In *ACM Symposium on Document Engineering*, pages 259–262, 2011.
- RK Ando and T Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853, 2005.
- M Baktashmotlagh, MT Harandi, BC Lovell, and M Salzmann. Unsupervised domain adaptation by domain invariant projection. In *International Conference on Computer Vision (ICCV)*, pages 769–776, 2013.
- M Baktashmotlagh, MT Harandi, BC Lovell, and M Salzmann. Domain adaptation on the statistical manifold. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2481–2488, 2014.
- H Bay, T Tuytelaars, and L Van Gool. Surf: speeded up robust features. In *European Conference on Computer Vision (ECCV)*, pages 404–417, 2006.
- CM Bishop. Training with noise is equivalent to tilkhonov regularization. *Neural Computation*, 7(1):108–116, 1995.
- DM Blei, AY Ng, and MI Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.
- J Blitzer, R McDonald, and F Pereira. Domain adaptation with structural correspondence learning. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 120–128, 2006.
- J Blitzer, M Dredze, and F Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Association for Computational Linguistics (ACL)*, volume 7, pages 440–447, 2007.
- J Blitzer, S Kakade, and DP Foster. Domain adaptation with coupled subspaces. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 173–181, 2011.
- KM Borgwardt, A Gretton, MJ Rasch, HP Kriegel, B Schölkopf, and AJ Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006.
- CC Chang and CJ Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- N Chen, J Zhu, J Chen, and B Zhang. Dropout training for support vector machines. In *AAAI Conference on Artificial Intelligence*, 2014.
- C Cortes and M Mohri. Domain adaptation in regression. In *Algorithmic Learning Theory (ALT)*, pages 308–323, 2011.
- C Cortes, M Mohri, M Riley, and A Rostamizadeh. Sample selection bias correction theory. In *Algorithmic Learning Theory (ALT)*, pages 38–53, 2008.
- R Detrano, A Janosi, W Steinbrunn, M Pfisterer, JJ Schmid, S Sandhu, KH Guppy, S Lee, and V Froelicher. International application of a new probability algorithm for the diagnosis of coronary artery disease. *American Journal of Cardiology*, 64(5):304–310, 1989.
- CV Dinh, RPW Duin, I Piqueiras-Salazar, and M Loog. Fidos: A generalized fisher based feature extraction method for domain shift. *Pattern Recognition*, 46(9):2510–2518, 2013.
- J Donahue, Y Jia, O Vinyals, J Hoffman, N Zhang, E Tzeng, and T Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning (ICML)*, pages 647–655, 2014.
- M Elter, R Schulz-Wendland, and T Wittenberg. The prediction of breast cancer biopsy outcomes using two cad approaches that both emphasize an intelligible decision process. *Medical Physics*, 34(11):4164–4172, 2007.
- B Fernando, A Habrard, M Sebban, and T Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *International Conference on Computer Vision (ICCV)*, pages 2960–2967, 2013.
- B Gong, Y Shi, F Sha, and K Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2066–2073, 2012.
- B Gong, K Grauman, and F Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *International Conference on Machine Learning (ICML)*, pages 222–230, 2013.
- R Gopalan, R Li, and R Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *International Conference on Computer Vision (ICCV)*, pages 999–1006, 2011.
- A Gretton, AJ Smola, J Huang, M Schmittfull, KM Borgwardt, and B Schölkopf. Covariate shift by kernel mean matching. In Quinero Candela, M Sugiyama, A Schwaighofer, and ND Lawrence, editors, *Dataset Shift in Machine Learning*, pages 131–160. MIT Press, 2009.
- HA Guvenir, B Acar, G Demiroz, and A Cekin. Supervised machine learning algorithm for arrhythmia analysis. *Computers in Cardiology*, pages 433–436, 1997.
- GE Hinton, N Srivastava, A Krizhevsky, I Sutskever, and RR Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.
- J Hoffman, E Rodner, J Donahue, T Darrell, and K Saenko. Efficient learning of domain-invariant image representations. *International Conference on Learning Representations (ICLR)*, 2013.

- J Huang, A Gretton, KM Borgwardt, B Schölkopf, and AJ Smola. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems*, pages 601–608, 2006.
- JJ Hull. A database for handwritten text recognition research. *Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994.
- H Jégou, F Perronnin, M Douze, J Sanchez, P Perez, and C Schmid. Aggregating local image descriptors into compact codes. *Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, 2012.
- B Klimt and Y Yang. Introducing the enron corpus. 2004.
- Y LeCun, L Bottou, Y Bengio, and P Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- CJ Leggetter and PC Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer Speech & Language*, 9(2):171–185, 1995.
- W Li, L Duan, D Xu, and IW Tsang. Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. *Pattern Analysis and Machine Intelligence*, 36(6):1134–1148, 2014.
- M Lichman. UCI machine learning repository. 2013. URL <http://archive.ics.uci.edu/ml>.
- RJA Little and DB Rubin. *Statistical Analysis with Missing Data*. John Wiley & Sons, 2014.
- SI Pan, IW Tsang, JT Kwok, and Q Yang. Domain adaptation via transfer component analysis. *Neural Networks*, 22(2):199–210, 2011.
- B Pang and L Lee. A sentimental education: Sentiment analysis using subjectivity summation based on minimum cuts. In *Association for Computational Linguistics (ACL)*, page 271, 2004.
- VMK Peddinti and P Chintalapudi. Domain adaptation in sentiment analysis of twitter. In *AAAI Conference on Artificial Intelligence*, 2011.
- A Rostamizadeh, A Agarwal, and PL Bartlett. Learning with missing features. In *Uncertainty in Artificial Intelligence (UAI)*, pages 635–642, 2011.
- DB Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.
- K Saenko, B Kulis, M Fritz, and T Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision (ECCV)*, pages 213–226, 2010.
- M Shao, D Kit, and Y Fu. Generalized transfer subspace learning through low-rank constraint. *International Journal of Computer Vision*, 109(1-2):74–93, 2014.
- H Shinodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244, 2000.
- M Sugiyama, M Krauledat, and KR Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8:985–1005, 2007.
- LJP Van der Maaten, Mi Chen, S Tyree, and KQ Weinberger. Learning with marginalized corrupted features. In *International Conference on Machine Learning (ICML)*, pages 410–418, 2013.
- A Van Ophroek, MA Ikram, MW Vernooij, and M De Bruijne. A transfer-learning approach to image segmentation across scanners by maximizing distribution similarity. In *International Workshop on Machine Learning in Medical Imaging*, pages 49–56, 2013.
- S Wäger, S Wang, and P Liang. Dropout training as adaptive regularization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 351–359, 2013.
- G Zen, E Saungineto, E Ricci, and N Sebe. Unsupervised domain adaptation for personalized facial emotion recognition. In *International Conference on Multimodal Interaction (ICMI)*, pages 128–135, 2014.
- K Zhang and W Fan. Forecasting skewed biased stochastic ozone days: Analyses, solutions and beyond. *Knowledge and Information Systems*, 14(3):299–326, 2008.

Semiparametric Mean Field Variational Bayes: General Principles and Numerical Issues

David Rohde

*School of Mathematical and Physical Sciences
University of Technology Sydney
P.O. Box 123, Broadway, 2007, Australia*

DAVID.ROHDE@GMAIL.COM

Matt P. Wand

*School of Mathematical and Physical Sciences
University of Technology Sydney
P.O. Box 123, Broadway, 2007, Australia*

MATT.WAND@UTS.EDU.AU

Editor: Xiaotong Shen

Abstract

We introduce the term *semiparametric mean field variational Bayes* to describe the relaxation of mean field variational Bayes in which some density functions in the product density restriction are pre-specified to be members of convenient parametric families. This notion has appeared in various guises in the mean field variational Bayes literature during its history and we endeavor to unify this important topic. We lay down a general framework and explain how previous relevant methodologies fall within this framework. A major contribution is elucidation of numerical issues that impact semiparametric mean field variational Bayes in practice.

Keywords: Bayesian Computing, Factor Graph, Fixed-form Variational Bayes, Fixed-point Iteration, Non-conjugate Variational Message Passing, Nonlinear Conjugate Gradient Method

1. Introduction

We expound *semiparametric mean field variational Bayes*, a powerful combination of the notions of minimum Kullback-Leibler divergence and mean field restriction, that allows fast and often accurate approximate Bayesian inference for a wide range of scenarios. Most of its foundational literature and applications are in Machine Learning. However, semiparametric mean field variational Bayes is also an important paradigm for Statistics in the age of very big sample sizes and models.

Several articles concerned with deterministic approximate Bayesian inference, such as Barber and Bishop (1997), Honkela et al. (2010), Knowles and Minka (2011), Tan and Nott (2013), Wand (2014) and Menictas and Wand (2015), have demonstrated that modification of mean field variational Bayes (e.g. Wainwright and Jordan, 2008) to include pre-specified parametric families in the product density posterior approximation can have great practical benefits. For example, Barber and Bishop (1997) used a pre-specified Multivariate Normal distribution for the posterior density of the vector of adaptive parameters in multilayer neural networks while Tan and Nott (2013) derived a closed form variational approximate

algorithm for Bayesian Poisson mixed models by pre-specifying the fixed and random effects parameters to have Multivariate Normal distributions. Knowles and Minka (2011) took a message passing approach to mean field variational Bayes and explain how their approach to inclusion of pre-specified parametric families allows modular inference algorithms for arbitrary factors. Some recent articles on this topic have used the terms *fixed-form variational Bayes* (Honkela et al., 2010) and *non-conjugate variational message passing* (Knowles & Minka, 2011), to describe this modification of mean field variational Bayes. However, in this article we argue for adoption of the term *semiparametric mean field variational Bayes*.

Although we give a precise mathematical description of semiparametric mean field variational Bayes in Section 2, it simply refers to the relaxation of ordinary mean field variational Bayes in which some of the density functions in the postulated product density form are pre-specified to be particular parametric density functions, often chosen for reasons of tractability. This is a ‘halfway house’ between fully parametric approximation of the joint posterior density function of the model parameters with minimum Kullback-Leibler divergence used for parameter choice and pure mean field variational Bayes in which there is no parametric specification at all – only the product restriction. The following comments apply to our general framework:

- Semiparametric mean field variational Bayes is a modification of mean field variational Bayes that could be carried out via a message passing approach, as done by Knowles and Minka (2011), or by using the more common q -density approach used in, for example, Bishop (2006) and Ormerod and Wand (2010).
- The notion of conjugacy is not intrinsic to semiparametric mean field variational Bayes. The principle may be applied regardless of conjugacy relationships amongst the messages and/or q -densities. Therefore, the ‘non-conjugacy’ label used in recent articles for semiparametric relaxations of mean field variational Bayes is somewhat misleading.
- Contributions such as Knowles and Minka (2011) and Tan and Nott (2013) restrict attention to pre-specification of parametric families that are of exponential family form (e.g. Wainwright and Jordan, 2008). Whilst exponential family density functions have tractability advantages when used in semiparametric mean field variational Bayes, there is no intrinsic reason for only such densities to be used. In Section 5 we illustrate this point using pre-specified Skew-Normal density functions, which are not within the exponential family.
- Recent articles on non-conjugate variational message passing, such as Knowles and Minka (2011), Tan and Nott (2013) and Menictas and Wand (2015) used fixed-point iteration to minimize the Kullback-Leibler divergence or, equivalently, maximize the lower bound on the marginal log-likelihood. Theorem 1 of Knowles and Minka (2011) constitutes such an approach. However, any optimization approach could be used for obtaining the Kullback-Leibler optimal parameters such as Newton-Raphson iteration, quasi-Newton iteration, stochastic gradient descent, the Nelder-Mead simplex method and various hybrids and modifications of such methods.
- Some articles, such as the recent Challis and Barber (2013), are concerned solely with approximate inference via minimum divergence according to the pre-specification that

the posterior is within a parametric family such as Multivariate Normal with banded Cholesky covariance matrix factors. These contributions represent special cases of semiparametric mean field variational Bayes and their findings have relevance to the more general situation.

The main purposes of this article are:

- (1) Bring together the literature on semiparametric mean field variational Bayes and identify its core tenets.
- (2) Lay out and discuss numerical issues that arise in semiparametric mean field variational Bayes, which have a significant practical implications for this body of methodology.

The resulting exposition constitutes the first compendium on semiparametric mean field variational Bayes at its fullest level of generality. It can also be used as a basis for enhancements of semiparametric mean field variational Bayes methodology.

We use two examples to elucidate the general principles and numerical issues. The first, Example 1, involves a Bayesian model with a single parameter and, hence, is such that mean field approximation is not required. The simplicity of Example 1 allows a deep appreciation of the various issues with minimal notational overhead. Example 2 is the Bayesian Poisson mixed model treated in Wand (2014) and benefits from semiparametric mean field variational Bayes methodology. It demonstrates issues with high-dimensional optimization problems that are intrinsic to practical implementation.

One of the main outcomes of our numerical investigations is that fitting exponential family density functions via *natural* fixed-point iteration has some attractive properties. By ‘natural’, we mean a simple version of fixed-point iteration that arises when natural parametrizations are used. As we explain in Section 4, natural fixed-point iterations use *Riemannian* gradients to step through the parameter space, which is generally more efficient than ordinary gradients. The benefits of Riemannian gradient-based algorithms for Machine Learning problems goes back at least to Amari (1998). Such algorithms are the basis of the semiparametric mean field variational Bayes approach of Honkela et al. (2010).

In Section 2 we describe semiparametric mean field variational Bayes in full generality. A general overview of optimization strategies, pertinent to semiparametric mean field variational Bayes, is given in Section 3. The important special case of pre-specified exponential family density functions is treated in Section 4. Section 5 deals with the more difficult non-exponential family case via an illustrative example. Some closing remarks are given in Section 6.

2. General Principles

Semiparametric mean field variational Bayes is an approximate Bayesian inference method based on the principle of minimum Kullback-Leibler divergence. For arbitrary density functions p_1 and p_2 on \mathbb{R}^d ,

$$\text{KL}(p_1 \parallel p_2) \equiv \int_{\mathbb{R}^d} p_1(\mathbf{x}) \log \left\{ \frac{p_1(\mathbf{x})}{p_2(\mathbf{x})} \right\} d\mathbf{x}$$

denotes the *Kullback-Leibler divergence* of p_2 from p_1 . Note that

$$\text{KL}(p_1 \parallel p_2) \geq 0 \quad \text{for any } p_1 \text{ and } p_2. \quad (1)$$

Consider a generic Bayesian model with observed data \mathcal{D} and parameter vector (θ, ϕ) . The reason for this notational decomposition of the parameter vector will soon become apparent. Throughout this section we assume that (θ, ϕ) and \mathcal{D} are continuous random vectors with density functions $p(\theta, \phi)$ and $p(\mathcal{D})$. The situation where some components are discrete has similar treatment with summations replacing integrals. Bayesian inference for θ and ϕ is based on the posterior density function

$$p(\theta, \phi | \mathcal{D}) = \frac{p(\mathcal{D}, \theta, \phi)}{p(\mathcal{D})}.$$

The denominator, $p(\mathcal{D})$, is usually referred to as the *marginal likelihood* or the *model evidence*.

Let $q(\theta, \phi)$ be an arbitrary density function over the parameter space of (θ, ϕ) . The essence of variational approximate inference is to restrict $q(\theta, \phi)$ to some class of density functions \mathcal{Q} and then use the optimal q -density function, given by

$$q^*(\theta, \phi) = \underset{q \in \mathcal{Q}}{\text{argmin}} \text{KL} \left\{ q(\theta, \phi) \parallel p(\theta, \phi | \mathcal{D}) \right\}, \quad (2)$$

as an approximation to the posterior density function $p(\theta, \phi | \mathcal{D})$.

Simple algebraic arguments (e.g. Section 2.1 of Ormerod and Wand, 2010) lead to

$$\log p(\mathcal{D}) = \text{KL} \left\{ q(\theta, \phi) \parallel p(\theta, \phi | \mathcal{D}) \right\} + \log \underline{p}(\mathcal{D}; q) \quad (3)$$

where

$$\underline{p}(\mathcal{D}; q) \equiv \exp \left[\int \int q(\theta, \phi) \log \left\{ \frac{p(q, \theta, \phi)}{q(\theta, \phi)} \right\} d\theta d\phi \right]. \quad (4)$$

From (1) we have

$$\underline{p}(\mathcal{D}; q) \leq p(\mathcal{D}) \quad \text{for any } q(\theta, \phi)$$

showing that $\underline{p}(\mathcal{D}; q)$ is a lower bound on the marginal likelihood. The non-negativity condition (1) means that an equivalent form for the optimal q -density function is

$$q^*(\theta, \phi) = \underset{q \in \mathcal{Q}}{\text{argmax}} \underline{p}(\mathcal{D}; q). \quad (5)$$

This alternative optimization problem has the attractive interpretation of $q^*(\theta, \phi)$ being chosen to maximize a lower bound on the marginal likelihood. For the remainder of this article we work with (5) rather than (2).

Parametric variational approximate inference involves setting

$$\mathcal{Q} = \{q(\theta, \phi; \xi) : \xi \in \Xi\},$$

corresponding to a parametric family of density functions with parameter vector ξ ranging over Ξ . In this case (5) reduces to

$$q^*(\theta, \phi) = \underset{\xi \in \Xi}{\text{argmax}} \underline{p}(\mathcal{D}; q, \xi), \quad (6)$$

where $\underline{p}(\mathcal{D}; q, \xi)$ is the marginal likelihood lower bound defined by (4), but with the dependence on ξ reflected in the notation.

An early contribution of this type is Hinton and van Camp (1993) who used minimum Kullback-Leibler divergence for Gaussian approximation of posterior density functions in neural networks models. Gaussian \mathcal{Q} families have also been used by Lappalainen and Honkela (2000), Archambeau et al. (2007), Raiko et al. (2007), Nickisch and Rasmussen (2008), Honkela and Valpola (2005), Honkela et al. (2007), Honkela et al. (2008) and Opper and Archambeau (2009). The recent contribution by Challis and Barber (2013) is an in-depth coverage of Gaussian minimum Kullback-Leibler approximate inference. Salimans and Knowles (2013) devised a stochastic approximation algorithm for solving (6) when \mathcal{Q} is a parametric family of exponential family form. Gershman et al. (2012) and Zobay (2014) investigated Gaussian-mixture extensions.

In what one may label a *nonparametric* variational approximation approach, ordinary mean field variational Bayes uses restricted q -density spaces such as

$$\mathcal{Q} = \{q(\theta, \phi) : q(\theta, \phi) = q(\theta_1) \cdots q(\theta_M) q(\phi)\} \text{ for some partition } \{\theta_1, \dots, \theta_M\} \text{ of } \theta. \quad (7)$$

The word ‘nonparametric’ is justified by the fact that there is no pre-specification that the q -density, or any of its factors, belong to a particular parametric family. Restriction of $q(\theta, \phi)$ to a product density form is the only pre-specification being made. An iterative scheme for solving (5) under (7) follows from the last displayed equation given in Section 10.1.1 of Bishop (2006). The scheme is listed explicitly as Algorithm 1 of Ormerod and Wand (2010). Note that a simple adjustment that caters for (θ, ϕ) , rather than θ , is required for notation being used here. Gershman et al. (2012) also use the word ‘nonparametric’ to describe a variational approximation approach. However, their methodology is parametric in the sense of the terminology that we are using here.

We propose that the term *semiparametric mean field variational Bayes* be used for restrictions of the form:

$$\mathcal{Q} = \{q(\theta, \phi) : q(\theta, \phi) = q(\theta_1) \cdots q(\theta_M) q(\phi; \xi), \xi \in \Xi\} \quad (8)$$

where $\{q(\phi; \xi) : \xi \in \Xi\}$ is a pre-specified parametric family of density functions in ϕ . Under (8) there is no insistence on the $q(\theta_i)$ having a particular parametric form. For models possessing particular conjugacy properties the optimal q -densities, $q^*(\theta_i)$, will belong to relevant conjugate families. However, in general, the optimal q -densities of the θ_i can assume arbitrary forms; see, for example, Figure 6 of Pham et al. (2013). The quality of a variational approximation is limited by the restrictions imposed by the particular choice of \mathcal{Q} . Semiparametric mean field variational Bayes imposes a product density restriction and then a parametric constraint on one of the factors. The overall quality of the approximation is determined by the combination of these two restrictions. While the estimated nonparametric factors are optimal given the product restriction, a parametric restriction with fewer product assumptions may be more accurate.

We now turn to the practical problem of solving the optimization problem (5) when the q -density restriction is of the form (8). Appropriate strategies for solving (5) depend on the nature of $q(\phi; \xi)$ as a function of ξ and the set Ξ . Some possibilities are:

(A) Ξ is a finite set.

(B) Ξ is an open subset of \mathbb{R}^d for some $d \in \mathbb{N}$ and $q(\phi; \xi)$ is smooth function of ξ over $\xi \in \Xi$.

(C) Ξ is an open subset of \mathbb{R}^d for some $d \in \mathbb{N}$ and $q(\phi; \xi)$ is a non-smooth function of ξ over $\xi \in \Xi$.

(D) Ξ is a complicated set that does not satisfy any of the descriptions given in (A)–(C).

For the vast majority of models in common use and $q(\phi; \xi)$ families (B) applies and most of the remainder of this article is devoted to that case. However, we will first briefly deal with (A) in Section 2.1, since it aids understanding of the semiparametric extension of mean field variational Bayes. To date, we are unaware of any semiparametric mean field variational Bayes contributions where (C) or (D) apply, so these cases are left aside.

2.1 Finite Parameter Space Case

Suppose that Ξ is a finite set. Then Algorithm 1 is the natural extension of the mean field variational Bayes coordinate ascent algorithm given, for example, in Section 10.1.1 of Bishop (2006) and Algorithm 1 of Ormerod and Wand (2010). In Algorithm 1, and elsewhere, the notation $\theta \setminus \theta_i$ denotes the vector θ with the entries of θ_i excluded.

For each $\xi \in \Xi$:

Initialize: $q(\theta_1), \dots, q(\theta_M)$.

Cycle:

$$q(\theta_1) \leftarrow \frac{\exp [E_{q(\theta \setminus \theta_1), q(\phi; \xi)} \{\log p(\mathbf{y}, \theta, \phi)\}]}{\int \exp [E_{q(\theta \setminus \theta_1), q(\phi; \xi)} \{\log p(\mathbf{y}, \theta, \phi)\}] d\theta_1}$$

⋮

$$q(\theta_M) \leftarrow \frac{\exp [E_{q(\theta \setminus \theta_M), q(\phi; \xi)} \{\log p(\mathbf{y}, \theta, \phi)\}]}{\int \exp [E_{q(\theta \setminus \theta_M), q(\phi; \xi)} \{\log p(\mathbf{y}, \theta, \phi)\}] d\theta_M}$$

until the increase in $\underline{p}(\mathcal{D}; q, \xi)$ is negligible.

$$q^*(\theta_i; \xi) \leftarrow q(\theta_i), 1 \leq i \leq M \quad ; \quad \underline{p}(\mathcal{D}; q^*, \xi) \leftarrow \underline{p}(\mathcal{D}; q, \xi).$$

$$\xi^* \leftarrow \operatorname{argmax}_{\xi \in \Xi} \underline{p}(\mathcal{D}; q^*, \xi) \quad ; \quad q^*(\theta_i) \leftarrow q^*(\theta_i; \xi^*), 1 \leq i \leq M.$$

Algorithm 1: *Coordinate ascent algorithm for semiparametric mean field variational Bayes when Ξ is a finite parameter space.*

For each value of ξ in Ξ , Algorithm 1 is essentially the ordinary mean field variational Bayes iterative algorithm — but with the density function of ϕ set to the parametric density function $q(\phi; \xi)$. The optimal ξ is then found by maximizing over the approximate marginal likelihood values that are recorded for each element of Ξ .

2.2 Infinite Parameter Space Case

Algorithm 1 shows how to obtain the Kullback-Leibler-optimal $q(\theta_i)$ and $q(\phi; \xi)$ densities in the case where Ξ is finite. However, for common parametric families such as the Normal and Gamma, Ξ is infinite and the solution of (5) under (8) is more delicate. The coordinate ascent idea used to obtain the $q^*(\theta_i)$ in Algorithm 1 can still be entertained. However, it needs to be combined with an optimization scheme that searches for the optimal ξ over the infinite space Ξ .

For the remainder of this article we focus on the problem of solving (5) under restriction (B) on the q -density parameter space Ξ . We start by studying the criterion function $\underline{p}(\mathcal{G}; q; \xi)$ and special forms that it takes under the mean field restriction. The notions of entropy and factor graphs are shown to be very relevant and useful. We then introduce two running examples, Example 1 and Example 2, to illustrate the issues involved. Since Example 1 has only one parameter requiring inference, this is not a fully-fledged semiparametric mean field variational Bayes problem and the optimization problem is of the form (6). Additionally, (6) for Example 1 is a bivariate optimization problem which allows deeper probing of the numerical analytic issues. Example 2 uses the Poisson mixed model, treated in Section 5.1 of Wand (2014), as our main semiparametric mean field variational Bayes example. It is substantial enough to convey various practical issues but also has a closed form $\log \underline{p}(q; \xi)$ expression that allows purely algebraic exposition.

2.2.1 ENTROPY, FACTOR GRAPHS AND THE MARGINAL LOG-LIKELIHOOD LOWER BOUND

If \mathbf{x} is a random vector having density function p then the corresponding *entropy* is given by

$$\text{Entropy}(p) \equiv E_p\{-\log p(\mathbf{x})\}.$$

For many common distributional families, the entropy admits an algebraic expression in terms of the distribution's parameters. For example, if

$$p(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-d/2} |\boldsymbol{\Sigma}|^{-1/2} \exp\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\}$$

is the Multivariate Normal density function of dimension d with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ then

$$\text{Entropy}(p; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{2} d \{1 + \log(2\pi)\} + \frac{1}{2} \log |\boldsymbol{\Sigma}|. \quad (9)$$

Note that the entropy is independent of the mean vector $\boldsymbol{\mu}$.

Another entropy expression, which arises in many Bayesian models and the example of Section 2.2.3, is that for the Inverse Gamma family of density functions. Let

$$x \sim \text{Inverse-Gamma}(\kappa, \lambda)$$

denote the random variable x having density function

$$p(x; \kappa, \lambda) = \frac{\kappa^\lambda}{\Gamma(\kappa)} x^{-\kappa-1} \exp(-\lambda/x), \quad x > 0,$$

with parameters $\kappa, \lambda > 0$. In this case

$$\text{Entropy}(p; \kappa, \lambda) = \log(\lambda) + \kappa + \log\{\Gamma(\kappa)\} - (\kappa + 1)\text{digamma}(\kappa) \quad (10)$$

where $\text{digamma}(x) \equiv (d/dx) \log \Gamma(x)$ is the digamma function.

The next relevant concept is that of a *factor graph*, which we first explain via an example. Consider the approximate Bayesian inference problem according to the semiparametric mean field variational Bayes restriction (8). Figure 1 is the factor graph for an $M = 9$ example of (8) with the joint density function of all random vectors in the model factorizing as follows:

$$p(\mathbf{x}, \boldsymbol{\theta}, \phi) = f_1(\boldsymbol{\theta}_1) f_2(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \phi) f_3(\boldsymbol{\theta}_2) f_4(\boldsymbol{\theta}_1) f_5(\boldsymbol{\theta}_5, \phi) f_6(\boldsymbol{\theta}_2) f_7(\boldsymbol{\theta}_6, \phi) f_8(\boldsymbol{\theta}_6) \\ \times f_9(\boldsymbol{\theta}_7, \boldsymbol{\theta}_8, \boldsymbol{\theta}_9) f_{10}(\boldsymbol{\theta}_2, \boldsymbol{\theta}_3, \boldsymbol{\theta}_4) \quad (11)$$

for *factors* f_1, \dots, f_{10} . Note that some of these factors depend on the data vector \mathbf{x} , but the dependence is suppressed in the f_j notation. Specific examples are given in Sections 2.2.2 and 2.2.3. In Figure 1 the circles correspond to the components of the mean field product restriction

$$q(\phi) \prod_{i=1}^9 q(\theta_i) \quad (12)$$

and are called *stochastic nodes*. The solid squares correspond to the factors f_j , $1 \leq j \leq 10$, and are called *factor nodes*. Edges join each factor node f_j to those stochastic nodes that are included in the f_j function.

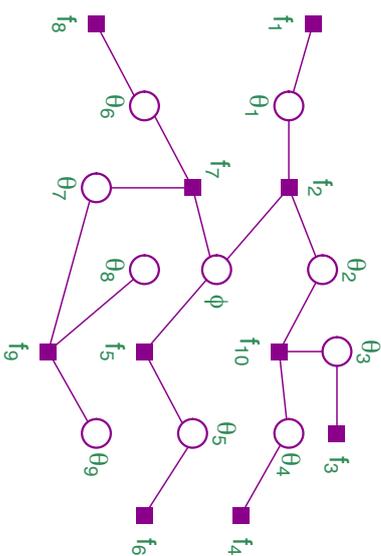


Figure 1: Factor graph corresponding to the model (11) and q -density product restriction (12).

Now consider the general case with semiparametric mean field restriction (8) and suppose that $p(\mathbf{x}, \boldsymbol{\theta}, \phi)$ has N factors f_j , $1 \leq j \leq N$. Then the marginal log-likelihood lower bound admits the following expression in terms of the components of the corresponding factor graph:

$$\log \underline{p}(\boldsymbol{\mathcal{D}}; q, \boldsymbol{\xi}) = \text{Entropy}\{q(\boldsymbol{\theta}_1; \boldsymbol{\xi})\} + \sum_{i=1}^M \text{Entropy}\{q(\boldsymbol{\theta}_i)\} + \sum_{j=1}^N E_q\{\log(f_j)\}. \quad (13)$$

The ϕ -localized component of $\log \underline{p}(\boldsymbol{\mathcal{D}}; q, \boldsymbol{\xi})$, which we denote by $\log \underline{p}(\boldsymbol{\mathcal{D}}; q, \boldsymbol{\xi})^{|\phi|}$, is

$$\log \underline{p}(\boldsymbol{\mathcal{D}}; q, \boldsymbol{\xi})^{|\phi|} \equiv \text{Entropy}\{q(\boldsymbol{\phi}; \boldsymbol{\xi})\} + \sum_{j \in \text{neighbors}(\phi)} E_q\{\log(f_j)\} \quad (14)$$

where

$$\begin{aligned} \text{neighbors}(\phi) &\equiv \{1 \leq j \leq N : f_j \text{ is a neighbor of } \phi \text{ on the factor graph}\} \\ &= \{1 \leq j \leq N : f_j \text{ involves } \phi\}. \end{aligned}$$

For the factor graph shown in Figure 1 $\text{neighbors}(\phi) = \{2, 5, 7\}$ and so have

$$\log \underline{p}(\boldsymbol{\mathcal{D}}; q, \boldsymbol{\xi})^{|\phi|} = \text{Entropy}\{q(\boldsymbol{\phi}; \boldsymbol{\xi})\} + E_q\{\log(f_2)\} + E_q\{\log(f_5)\} + E_q\{\log(f_7)\}.$$

as the ϕ -localized component of $\log \underline{p}(\boldsymbol{\mathcal{D}}; q, \boldsymbol{\xi})$.

For large Bayesian models it is prudent to maximize this localized approximate log-likelihood as part of a coordinate ascent scheme involving all q -density parameters. Such an approach, combined with the *locality property* of mean field variational Bayes (e.g. Wand et al., 2011, Section 3), allows for streamlined handling of arbitrarily large models. We formalize this approach to semiparametric mean field variational Bayes in the shape of Algorithm 2 in the upcoming Section 2.2.4. However, we first give some concrete examples of mean field variational Bayes with pre-specified parametric family q -density functions.

2.2.2 EXAMPLE 1: GUMBEL RANDOM SAMPLE

A Bayesian model for a random sample x_1, \dots, x_n from a Gumbel distribution with location parameter ϕ and unit scale is

$$p(x_1, \dots, x_n | \phi) = \prod_{i=1}^n \exp\{-(x_i - \phi) - e^{-(x_i - \phi)}\}, \quad \phi \sim N(\mu_\phi, \sigma_\phi^2). \quad (15)$$

The posterior density function of ϕ is

$$p(\phi | \mathbf{x}) = \frac{\exp\left\{n\phi - e^\phi \sum_{i=1}^n e^{-x_i} - \frac{1}{2\sigma_\phi^2}(\phi - \mu_\phi)^2\right\}}{\int_{-\infty}^{\infty} \exp\left\{n\phi' - e^{\phi'} \sum_{i=1}^n e^{-x_i} - \frac{1}{2\sigma_\phi^2}(\phi' - \mu_\phi)^2\right\} d\phi'}. \quad (16)$$

The denominator on the right-hand side of (15) is not available in closed form. This implies that numerical methods such as quadrature are required to obtain the Bayes estimate of ϕ

and corresponding credible sets. Instead we consider minimum Kullback-Leibler divergence approximation of $p(\phi | \mathbf{x})$ over a parametric pre-specified family. Let

$$\mathcal{Q} = \{q(\phi; \boldsymbol{\xi}) : \boldsymbol{\xi} \in \Xi\}$$

be such a parametric family. Then the optimal q -density is $q(\phi; \boldsymbol{\xi}^*)$ where

$$\boldsymbol{\xi}^* = \underset{\boldsymbol{\xi} \in \Xi}{\text{argmax}} \underline{p}(q; \boldsymbol{\xi}). \quad (17)$$

Figure 2 shows the factor graph of the model, with factors $p(\phi)$ and $p(\mathbf{x} | \phi)$ neighboring the stochastic node ϕ .

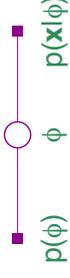


Figure 2: Factor graph for the Example 1 model.

The marginal log-likelihood lower bound is

$$\log \underline{p}(\mathbf{x}; \boldsymbol{\xi}) = \text{Entropy}\{q(\phi; \boldsymbol{\xi})\} + E_q\{\log p(\phi)\} + E_q\{\log p(\mathbf{x} | \phi)\} \quad (18)$$

and the contributions to $\log \underline{p}(\mathbf{x}; \boldsymbol{\xi})$ from the factors are

$$\begin{aligned} E_q\{\log p(\phi)\} &= -\frac{1}{2\sigma_\phi^2} \left\{ E_{q(\phi; \boldsymbol{\xi})}(\phi) - \mu_\phi \right\}^2 + \text{Var}_{q(\phi; \boldsymbol{\xi})}(\phi) \\ \text{and } E_q\{\log p(\mathbf{x} | \phi)\} &= n E_{q(\phi; \boldsymbol{\xi})}(\phi) - M_{q(\phi; \boldsymbol{\xi})}(1) \sum_{i=1}^n e^{-x_i} - n\bar{x} \end{aligned} \quad (19)$$

where $M_{q(\phi; \boldsymbol{\xi})}$ is the moment generating function corresponding to $q(\phi; \boldsymbol{\xi})$.

Now suppose that

$$\mathcal{Q} = \left\{ q(\phi; \mu_{q(\phi)}, \sigma_{q(\phi)}^2) = \frac{1}{\sqrt{2\pi\sigma_{q(\phi)}^2}} \exp\left\{ -\frac{(\phi - \mu_{q(\phi)})^2}{2\sigma_{q(\phi)}^2} \right\} : \mu_{q(\phi)} \in \mathbb{R}, \sigma_{q(\phi)}^2 > 0 \right\}$$

corresponding to the Normal family with q -density parameter vector $\boldsymbol{\xi} = (\mu_{q(\phi)}, \sigma_{q(\phi)}^2)$ and parameter space $\Xi = \mathbb{R} \times \mathbb{R}_+$ where $\mathbb{R}_+ \equiv (0, \infty)$ is the positive half-line. Then, from the entropy result (9) and the well-known expression for the moment generating function of the Normal distribution we obtain

$$\begin{aligned} \log \underline{p}(\mathbf{x}; \mu_{q(\phi)}, \sigma_{q(\phi)}^2) &= \frac{1}{2} \{1 + \log(2\pi)\} + \frac{1}{2} \log(\sigma_{q(\phi)}^2) + n \mu_{q(\phi)} \\ &\quad - \exp(\mu_{q(\phi)} + \frac{1}{2} \sigma_{q(\phi)}^2) \sum_{i=1}^n e^{-x_i} - \frac{1}{2\sigma_\phi^2} \{(\mu_{q(\phi)} - \mu_\phi)^2 + \sigma_{q(\phi)}^2\} - n\bar{x}. \end{aligned}$$

It follows that the Kullback-Leibler optimal Normal q -density function is $q(\phi; \mu_{q(\phi)}^*, \sigma_{q(\phi)}^{2*})$ where

$$\left[\begin{array}{c} \mu_{q(\phi)}^* \\ \sigma_{q(\phi)}^{2*} \end{array} \right] = \underset{\mu_{q(\phi)} \in \mathbb{R}, \sigma_{q(\phi)}^2 > 0}{\text{argmax}} \left\{ f_{\text{ExI}}^N \left(\mu_{q(\phi)}, \sigma_{q(\phi)}^2; n, \sum_{i=1}^n e^{-x_i}, \mu_\phi, \sigma_\phi^2 \right) \right\} \quad (20)$$

and

$$f_{\text{EVI}}^N(x, y; a, b, c, d) = \frac{1}{2} \log(y) + ax - b \exp(x + \frac{1}{2}y) - \frac{1}{2d} \{(x - c)^2 + y\}. \quad (20)$$

The main arguments satisfy $x \in \mathbb{R}$, $y > 0$ and the auxiliary arguments are such that $a, b, d > 0$ and $c \in \mathbb{R}$. From (19) we see that the minimum Kullback-Leibler divergence problem (16), where \mathcal{Q} is the Normal family, reduces to a non-linear bivariate optimization problem. Theory given in Challis and Barber (2013) applies to this example. For example, results given in their Section 3.2 can be used to establish that $f_{\text{EVI}}^N(x, y; a, b, c, d)$ is jointly concave in x and \sqrt{y} .

In Section 3 we study strategies for solving such problems and apply them to this example in Section 4.2.

2.2.3 EXAMPLE 2: POISSON MIXED MODEL

A single variance component *Poisson mixed model* is

$$\begin{aligned} y_i | \beta, \mathbf{u} &\stackrel{\text{ind.}}{\sim} \text{Poisson}[\exp\{(\mathbf{X}\beta + \mathbf{Z}\mathbf{u})_i\}], \quad 1 \leq i \leq n, \\ \mathbf{u} | \sigma^2 &\sim N(0, \sigma^2 \mathbf{I}_K), \quad \sigma^2 | a \sim \text{Inverse-Gamma}(\frac{1}{2}, 1/a), \\ \beta &\sim N(\mathbf{0}, \sigma_\beta^2 \mathbf{I}_p), \quad a \sim \text{Inverse-Gamma}(\frac{1}{2}, 1/A^2) \end{aligned} \quad (21)$$

where \mathbf{X} is an $n \times p$ fixed effects design matrix, \mathbf{Z} is an $n \times K$ random effects design matrix. Note that the prior on σ in (21) is the Half Cauchy distribution with scale parameter A :

$$p(\sigma) = \frac{(2/\pi)}{A[1 + (\sigma/A)^2]}, \quad \sigma > 0.$$

In (21) $\sigma\beta > 0$ and $A > 0$ are hyperparameters to be chosen by the analyst.

A mean field approximation to the joint posterior density function of the model parameters is

$$p(\beta, \mathbf{u}, \sigma^2, a | \mathbf{y}) \approx q(\beta, \mathbf{u}) q(\sigma^2) q(a). \quad (22)$$

As detailed in Appendix A.3, the optimal q -density functions satisfy

$$\begin{aligned} q^*(\sigma^2) \text{ and } q^*(a) &\text{ are both Inverse-Gamma density functions, and} \\ q^*(\beta, \mathbf{u}) &\propto \exp\{\mathbf{y}^T(\mathbf{X}\beta + \mathbf{Z}\mathbf{u}) - \mathbf{1}^T \exp(\mathbf{X}\beta + \mathbf{Z}\mathbf{u}) \\ &\quad - \frac{1}{2}\sigma_\beta^2 \|\beta\|^2 - \frac{1}{2} E_{q(\sigma^2)}(1/\sigma^2) \|\mathbf{u}\|^2\}. \end{aligned} \quad (23)$$

Since $q^*(\beta, \mathbf{u})$ is not a standard form, numerical methods are required to obtain the variational approximate Bayes estimates and credible sets. *Semiparametric* mean field variational Bayes alternatives take the form

$$p(\beta, \mathbf{u}, \sigma^2, a | \mathbf{y}) \approx q(\beta, \mathbf{u}; \xi) q(\sigma^2) q(a) \quad (24)$$

where $\{q(\beta, \mathbf{u}; \xi) : \xi \in \Xi\}$ is a pre-specified parametric family of density functions. The optimal density functions $q(\beta, \mathbf{u}; \xi^*)$, $q^*(\sigma^2)$ and $q^*(a)$ are found by minimizing

$$\text{KL}\{q(\beta, \mathbf{u}; \xi) q(\sigma^2) q(a) \parallel p(\beta, \mathbf{u}, \sigma^2, a | \mathbf{y})\}. \quad (25)$$

We now focus on solving (25).

In Appendix C of Wand (2014) it is shown that

$q^*(\sigma^2)$ is an Inverse-Gamma($\frac{1}{2}(K+1)$, $B_{q(\sigma^2)}$) density function, and $q^*(a)$ is an Inverse-Gamma(1, $B_{q(a)}$) density function

where

$$B_{q(\sigma^2)} = \frac{1}{2} \|\| E_{q(\beta, \mathbf{u}; \xi)}(\mathbf{u}) \|^2 + \text{tr}\{\text{Cov}_{q(\beta, \mathbf{u}; \xi)}(\mathbf{u})\} + \mu_{q(1/a)} \quad (26)$$

and $B_{q(a)} = \mu_{q(1/\sigma^2)} + A^{-2}$ with the definition

$$\mu_{q(1/v)} \equiv E_{q(v)}(1/v)$$

for a generic random variable v .

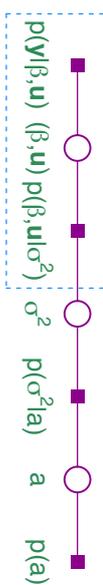


Figure 3: Factor graph for the Example 2 model with stochastic nodes corresponding the mean field restriction (24). The dashed line box contains the stochastic node (β, \mathbf{u}) and its neighboring factors.

It remains to obtain the optimal value of ξ in $q(\beta, \mathbf{u}; \xi)$. Figure 3 shows the factor graph of the current model under mean field restriction (24). The lower bound on the marginal log-likelihood, in terms of the stochastic nodes and factors of Figure 3, is

$$\begin{aligned} \log \underline{p}(\mathbf{y}; q, \xi) &= \text{Entropy}\{q(\beta, \mathbf{u}; \xi)\} + \text{Entropy}\{q(\sigma^2)\} + \text{Entropy}\{q(a)\} \\ &\quad + E_{q\{\log p(\mathbf{y}|\beta, \mathbf{u})\}} + E_q\{\log p(\beta, \mathbf{u} | \sigma^2)\} \\ &\quad + E_{q\{\log p(\sigma^2 | a)\}} + E_q\{\log p(a)\}. \end{aligned} \quad (27)$$

One could substitute (26) into the $\log \underline{p}(\mathbf{y}; q, \xi)$ expression. This resulting marginal log-likelihood lower bound would depend on the q -densities only through ξ and could then be maximized over $\xi \in \Xi$.

An alternative strategy, that scales better to larger models, is to use a coordinate ascent scheme that maximizes $\log \underline{p}(\mathbf{y}; q, \xi)^{(\beta, \mathbf{u})}$, the (β, \mathbf{u}) -localized component of $\log \underline{p}(\mathbf{y}; q, \xi)$, over $\xi \in \Xi$. The relevant factors are those neighboring (β, \mathbf{u}) in Figure 3, corresponding to the dashed line box. The quantity requiring maximization is then

$$\begin{aligned} \log \underline{p}(\mathbf{y}; q, \xi)^{(\beta, \mathbf{u})} &= \text{Entropy}\{q(\beta, \mathbf{u}; \xi)\} + E_{q\{\log p(\mathbf{y}|\beta, \mathbf{u})\}} + E_q\{\log p(\beta, \mathbf{u} | \sigma^2)\} \\ &= \text{Entropy}\{q(\beta, \mathbf{u}; \xi)\} + \mathbf{y}^T \{\mathbf{X} E_{q(\beta, \mathbf{u}; \xi)}(\beta) + \mathbf{Z} E_{q(\beta, \mathbf{u}; \xi)}(\mathbf{u})\} \\ &\quad - \mathbf{1}^T E_{q(\beta, \mathbf{u}; \xi)}\{\exp(\mathbf{X}\beta + \mathbf{Z}\mathbf{u})\} \\ &\quad - \frac{1}{2\sigma_\beta^2} \|\| E_{q(\beta, \mathbf{u}; \xi)}(\beta) \|^2 + \text{tr}\{\text{Cov}_{q(\beta, \mathbf{u}; \xi)}(\beta)\} \\ &\quad - \frac{1}{2} \mu_{q(1/\sigma^2)} \|\| E_{q(\beta, \mathbf{u}; \xi)}(\mathbf{u}) \|^2 + \text{tr}\{\text{Cov}_{q(\beta, \mathbf{u}; \xi)}(\mathbf{u})\} + \text{const} \end{aligned} \quad (28)$$

where ‘const’ denotes terms not depending on ξ .

Next, suppose that \mathcal{Q} corresponds to the family of Multivariate Normal density functions in (β, \mathbf{u}) :

$$\begin{aligned} q(\beta, \mathbf{u}; \boldsymbol{\mu}_{q(\beta, \mathbf{u})}, \boldsymbol{\Sigma}_{q(\beta, \mathbf{u})}) &= (2\pi)^{-(p+K)/2} |\boldsymbol{\Sigma}_{q(\beta, \mathbf{u})}|^{-1/2} \\ &\times \exp \left\{ -\frac{1}{2} \left(\begin{bmatrix} \beta \\ \mathbf{u} \end{bmatrix} - \boldsymbol{\mu}_{q(\beta, \mathbf{u})} \right)^T \boldsymbol{\Sigma}_{q(\beta, \mathbf{u})}^{-1} \left(\begin{bmatrix} \beta \\ \mathbf{u} \end{bmatrix} - \boldsymbol{\mu}_{q(\beta, \mathbf{u})} \right) \right\}. \end{aligned} \quad (29)$$

Then the (β, \mathbf{u}) -localized approximate marginal log-likelihood reduces to

$$\begin{aligned} \log \underline{p}(\mathbf{y}; q, \boldsymbol{\mu}_{q(\beta, \mathbf{u})}, \boldsymbol{\Sigma}_{q(\beta, \mathbf{u})}) &= \frac{1}{2} \log |\boldsymbol{\Sigma}_{q(\beta, \mathbf{u})}| + \mathbf{y}^T \mathbf{C} \boldsymbol{\mu}_{q(\beta, \mathbf{u})} \\ &- \mathbf{1}^T \exp \left\{ \mathbf{C} \boldsymbol{\mu}_{q(\beta)} + \frac{1}{2} \text{diagonal}(\mathbf{C} \boldsymbol{\Sigma}_{q(\beta, \mathbf{u})} \mathbf{C}^T) \right\} \\ &- \frac{1}{2\sigma_\beta^2} \left\{ \|\boldsymbol{\mu}_{q(\beta)}\|^2 + \text{tr}(\boldsymbol{\Sigma}_{q(\beta)}) \right\} \\ &- \frac{1}{2} \mu_{q(1/\sigma^2)} \left\{ \|\boldsymbol{\mu}_{q(\mathbf{u})}\|^2 + \text{tr}(\boldsymbol{\Sigma}_{q(\mathbf{u})}) \right\} + \text{const} \end{aligned}$$

where $\mathbf{C} \equiv [\mathbf{X} \ \mathbf{Z}]$, $\text{diagonal}(\mathbf{M})$ is the vector containing the diagonal entries of the square matrix \mathbf{M} , and $\boldsymbol{\mu}_{q(\beta)}$ is the sub-vector of $\boldsymbol{\mu}_{q(\beta, \mathbf{u})}$ corresponding to β . Analogous definitions apply to $\boldsymbol{\mu}_{q(\mathbf{u})}$, $\boldsymbol{\Sigma}_{q(\beta)}$ and $\boldsymbol{\Sigma}_{q(\mathbf{u})}$. Appendix A.3 provides derivational details for (29).

For this example, the full coordinate ascent scheme has updates as follows:

perform one or more updates of $(\boldsymbol{\mu}_{q(\beta, \mathbf{u})}, \boldsymbol{\Sigma}_{q(\beta, \mathbf{u})})$ within an iterative scheme for the optimization problem:

$$\underset{\boldsymbol{\mu}'_{q(\beta, \mathbf{u})}, \boldsymbol{\Sigma}'_{q(\beta, \mathbf{u})}}{\text{argmax}} \left\{ \log \underline{p}(\mathbf{y}; q, \boldsymbol{\mu}'_{q(\beta, \mathbf{u})}, \boldsymbol{\Sigma}'_{q(\beta, \mathbf{u})})^{|\beta, \mathbf{u}|} \right\}$$

$$B_{q(\alpha)} \leftarrow \mu_{q(1/\sigma^2)} + A^{-2} \quad ; \quad \mu_{q(1/\alpha)} \leftarrow 1/B_{q(\alpha)}$$

$$B_{q(\sigma^2)} \leftarrow \frac{1}{2} \left\{ \|\boldsymbol{\mu}_{q(\mathbf{u})}\|^2 + \text{tr}(\boldsymbol{\Sigma}_{q(\mathbf{u})}) \right\} + \mu_{q(1/\alpha)}$$

$$\mu_{q(1/\sigma^2)} \leftarrow \frac{1}{2} (K + 1) / B_{q(\sigma^2)}.$$

For now, we deliberately leave the form of the $(\boldsymbol{\mu}_{q(\beta, \mathbf{u})}, \boldsymbol{\Sigma}_{q(\beta, \mathbf{u})})$ maximization strategy unspecified. We also allow for one or more updates of an iterative scheme aimed at maximizing $\log \underline{p}(\mathbf{y}; q, \boldsymbol{\mu}'_{q(\beta, \mathbf{u})}, \boldsymbol{\Sigma}'_{q(\beta, \mathbf{u})})^{|\beta, \mathbf{u}|}$, rather than iterating to convergence at every iteration of the full coordinate ascent scheme. Section 3 describes various optimization strategies that could be used for updating $(\boldsymbol{\mu}_{q(\beta, \mathbf{u})}, \boldsymbol{\Sigma}_{q(\beta, \mathbf{u})})$.

Negligible absolute change in $\log \underline{p}(\mathbf{y}; q, \boldsymbol{\mu}_{q(\beta, \mathbf{u})}, \boldsymbol{\Sigma}_{q(\beta, \mathbf{u})})$ can be used as a stopping criterion for the iterations and an algebraic expression for this quantity is given in Appendix A.3.

We return to Example 2 in Section 4.3.

2.2.4 GENERAL SEMIPARAMETRIC MEAN FIELD VARIATIONAL BAYES ALGORITHM

We now treat semiparametric mean field variational Bayes in general, with the set-up laid out in Section 2 and restriction (8). Let $\log \underline{p}(\mathcal{D}; q, \xi)^{|\phi|}$ be defined with respect to the

factor graph of $p(\mathbf{x}, \phi, \boldsymbol{\theta})$ with stochastic nodes $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_M$ and ϕ . Algorithm 2 is a general coordinate ascent algorithm for approximate inference that builds on the standard mean field variational Bayes algorithm.

Initialize: $q(\boldsymbol{\theta}_2), \dots, q(\boldsymbol{\theta}_M)$.

Cycle:

perform one or more updates of ξ within an iterative scheme for the optimization problem:

$$\begin{aligned} &\underset{\xi \in \Xi}{\text{argmax}} \left\{ \log \underline{p}(\mathcal{D}; q, \xi)^{|\phi|} \right\} \\ q(\boldsymbol{\theta}_1) &\leftarrow \frac{\exp \left[E_{q(\boldsymbol{\theta} \setminus \boldsymbol{\theta}_1)} q(\phi; \xi) \left\{ \log p(\mathbf{y}, \boldsymbol{\theta}, \phi) \right\} \right]}{\int \exp \left[E_{q(\boldsymbol{\theta} \setminus \boldsymbol{\theta}_1)} q(\phi; \xi) \left\{ \log p(\mathbf{y}, \boldsymbol{\theta}, \phi) \right\} \right] d\boldsymbol{\theta}_1} \\ &\quad \vdots \\ q(\boldsymbol{\theta}_M) &\leftarrow \frac{\exp \left[E_{q(\boldsymbol{\theta} \setminus \boldsymbol{\theta}_M)} q(\phi; \xi) \left\{ \log p(\mathbf{y}, \boldsymbol{\theta}, \phi) \right\} \right]}{\int \exp \left[E_{q(\boldsymbol{\theta} \setminus \boldsymbol{\theta}_M)} q(\phi; \xi) \left\{ \log p(\mathbf{y}, \boldsymbol{\theta}, \phi) \right\} \right] d\boldsymbol{\theta}_M} \end{aligned}$$

until the absolute change in $\log \underline{p}(\mathcal{D}; q, \xi)$ is negligible.

Algorithm 2: *The general semiparametric mean field variational Bayes algorithm for restriction (8) with $\log \underline{p}(\mathcal{D}; q, \xi)^{|\phi|}$ defined with respect to factor graph of $p(\mathbf{x}, \boldsymbol{\theta}, \phi)$ with stochastic nodes $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_M$ and ϕ .*

For each of these approaches, there remains the practical problem of devising an optimization scheme for $\log \underline{p}(\mathcal{D}; q, \xi)^{|\phi|}$ and ensuring that it leads to the optimal parameters being chosen. Section 3 deals with this problem.

2.3 Relationship to Existing Literature

The general principle of semiparametric mean field variational Bayes that we have laid out in this section is not brand new and, in fact, instances of this principle have made appearances in the literature since the late 1990s — although they are few in number. We now briefly survey articles known to us that have a semiparametric mean field variational Bayes component. As we will see, the terminology varies quite considerably.

Barber and Bishop (1997) uses the terms *ensemble learning* and *hyperparameter adaptation* for what essentially is a semiparametric mean field variational Bayes approach to fitting multi-layer neural networks. They pre-specify Multivariate Normal distributions for the coefficient vector but, in their Section 2.1, allow covariance matrix parameters to be unspecified except for mean field assumptions. However, they do not include numerical details for minimizing Kullback-Leibler divergence over the Multivariate Normal parameters.

Honkela et al. (2010) adopt the phrase *fixed-form variational Bayes* in what is a quite general approach to semiparametric mean field variational Bayes as summarized in their Algorithm 1. Optimization of the pre-specified parametric component parameters is achieved using the *nonlinear conjugate gradient method*, which we describe in Section 3.2. However, Honkela et al. (2010) work with *Riemannian* gradients which, they argue, are more efficient than Euclidean gradients.

In Knowles and Minka (2011) the focus is incorporation of pre-specified exponential family distributions whilst preserving the message passing aspect of a modular approach to mean field variational Bayes, known as *variational message passing*. They arrive at an extension which they label *non-conjugate variational message passing*. The exponential family distribution parameters are chosen via fixed-point iteration, which we describe in detail in Section 3.1.

Tan and Nott (2013) take a semiparametric mean field variational Bayes approach to approximate inference in Bayesian generalized linear mixed models for grouped data. They use pre-specified Multivariate Normal density functions for the random effects of each group with mean field product restrictions and achieve good approximation accuracy via so-called partially noncentered parameterizations.

In the case of pre-specified Multivariate Normal density functions, Wand (2014) obtains an explicit form for the fixed-point iteration scheme of Knowles and Minka (2011) and illustrated its use for the Poisson mixed model described in Section 2.2.3. In Luts and Wand (2015) and Menzies and Wand (2015), semiparametric mean field variational Bayes with Multivariate Normal pre-specification is applied, respectively, to count response semiparametric regression and heteroscedastic semiparametric regression.

2.4 Advantages and Limitations of Algorithm 2

As just described in Section 2.3, Algorithm 2 is a very useful generalization of the ordinary mean field variational Bayes algorithm and allows for tractable handling of a wider class of models. For example, the heteroscedastic nonparametric regression model of Menzies and Wand (2015) is such that ordinary mean field variational Bayes is numerically challenging if one uses the same product restrictions as used in homoscedastic nonparametric regression. The semiparametric mean field variational Bayes extension, based on Multivariate Normal pre-specification of basis function coefficients, leads to an iterative scheme with closed form updates. Simulation studies show very good accuracy compared with Markov chain Monte Carlo-based inference. However Algorithm 2 is not guaranteed to converge and, when it does converge, may result in mediocre approximate Bayesian inference. We close this section by briefly discussing such limitations of Algorithm 2.

In cases where a generic iterative procedure is used to solve $\operatorname{argmax}_{\xi \in \Xi} \{\log p(\mathcal{D}; \eta, \xi')\}$ there is no guarantee that the lower bound is increased in a particular iteration or of convergence in general. As a consequence, the convergence guarantees enjoyed by ordinary mean field variational Bayes algorithms are not shared by their semiparametric extension. As we demonstrate in Section 4.2, convergence does not occur for particular numerical optimization strategies.

Lastly, there is the limitation imposed by the mean field restriction. Even though mean field variational Bayes can lead to very accurate approximate inference, there are

circumstances where its accuracy is quite poor. Some examples, with explanations for the inaccuracy, are given in Wang and Titterton (2005) and Neville et al. (2014). Semiparametric mean field variational Bayes shares this limitation since parametric pre-specification imposes a degradation in accuracy compared with ordinary mean field variational Bayes.

3. Numerical Optimization Strategies

Ordinary mean field variational Bayes, parameter optimization is achieved using a convex optimization algorithm that converges under reasonable assumptions (e.g. Uentberger and Ye, 2008). In the semiparametric extension there is no such convex optimization theory and general numerical optimization has to be called upon to optimize the parameters in the pre-specified parametric density function.

Numerical optimization is a major area of mathematical study with an enormous literature. Recent summaries are given in, for example, Givens and Hoeting (2005) and Ackleh et al. (2010), with the former being geared towards optimization problems arising in Statistics. The choice of optimization method typically is driven by factors such as the smoothness of the function requiring optimization and availability of expressions for low-order derivatives. Optimization methods with derivative information invariably take the form of iterative schemes. Semiparametric mean field variational Bayes often has the luxuries of smoothness and derivative expressions. It is also beneficial to have relatively simple iterative updates given the overarching goal of fast approximate inference. Therefore, we gear our summary of numerical optimization strategies towards simple derivative-based schemes. This is in keeping with the existing literature on parametric and semiparametric variational inference.

Let $f : D \subseteq \mathbb{R}^d \rightarrow \mathbb{R}$ be a function and consider the problem of finding its maximum or minimum value over a set $D_0 \subseteq D$. If all partial derivatives of f exist then a necessary condition for a maximum or minimum in the interior of D_0 is the *stationary point condition*

$$\frac{\partial}{\partial x_j} f(\mathbf{x}) = 0, \quad 1 \leq j \leq d. \quad (30)$$

This converts the optimization problem to a multivariate root-finding problem. However (30) is not a sufficient condition for global optima since local optima and saddle points of f inside D_0 also satisfy (30). Properties of f , such as regions where it is concave or convex, can aid the determination of global optima.

Throughout this section we make use of the derivative matrix and Hessian matrix notation defined in Appendix A.2.

3.1 Fixed-Point Iteration

Assuming that the derivative vector $\mathbf{D}_x f(\mathbf{x})$ (defined formally in Appendix A.1) exists, the stationary point condition (30) can be written

$$\mathbf{D}_x f(\mathbf{x})^T = \mathbf{0} \quad (31)$$

where $\mathbf{0}$ is the vector of zeroes. Fixed-point iteration aims to find points that satisfy (31), which we denote generically by \mathbf{x}^* . Such points are then candidates as maxima or minima

of f . Firstly, (31) is rewritten in the form

$$\mathbf{x} = \mathbf{g}(\mathbf{x}) \quad (32)$$

for some function $\mathbf{g} : D \subseteq \mathbb{R}^d \rightarrow \mathbb{R}^d$. Given this \mathbf{g} , fixed-point iteration simply involves repeated evaluation of \mathbf{g} , as given in Algorithm 3

Initialize: $\mathbf{x} \leftarrow \mathbf{x}_{\text{init}}$ for some $\mathbf{x}_{\text{init}} \in D$.

Cycle:

$$\mathbf{x} \leftarrow \mathbf{g}(\mathbf{x})$$

until convergence.

Algorithm 3: *The fixed-point iteration algorithm in generic form.*

Note, however, the following issues regarding fixed point iteration:

- For a given stationary point condition (31) there are numerous functions \mathbf{g} for which (32) holds. In other words, there are many possible fixed point algorithms available to solve (31).
- Once \mathbf{g} and \mathbf{x}_{init} are chosen then the above algorithm is *not necessarily guaranteed* to converge to a stationary point \mathbf{x}^* . There is a large literature on convergence of fixed-point iterative algorithms and good references on the topic include Section 8.1 of Ortega (1990) and Section 8.2 of Ackleh et al. (2010). For example Theorem 8.1.7 of Ortega (1990) asserts that convergence of Algorithm 3 is guaranteed when \mathbf{x}_{init} is sufficiently close to \mathbf{x}^* , the components of \mathbf{g} are differentiable at \mathbf{x}^* and

$$\rho(D_{\mathbf{x}}\mathbf{g}(\mathbf{x}^*)) < 1.$$

Here $\rho(\mathbf{A})$ denotes the *spectral radius* of the square matrix \mathbf{A} , defined to be

$$\rho(\mathbf{A}) \equiv \text{maximum of the absolute values of the eigenvalues of } \mathbf{A}.$$

Theorem 8.4 of Ackleh et al. (2010) provides a similar condition in terms of the *spectral norm* $\|D_{\mathbf{x}}\mathbf{g}(\mathbf{x}^*)\|_{\text{spec}}$ where

$$\|\mathbf{A}\|_{\text{spec}} \equiv \sqrt{\text{largest eigenvalue of } \mathbf{A}^T \mathbf{A}}.$$

- There are also theorems that guarantee convergence of Algorithm 3 for particular choices of \mathbf{x}_{init} . If D_0 is a closed convex subset of D such that $\mathbf{g} : D_0 \rightarrow D_0$, the entries of $D_{\mathbf{x}}\mathbf{g}(\mathbf{x})$ are each bounded and continuous on D_0 and

$$\sup_{\mathbf{x} \in D_0} \|D_{\mathbf{x}}\mathbf{g}(\mathbf{x})\|_{\text{spec}} \leq \alpha < 1$$

then Algorithm 3 will converge from any initial point $\mathbf{x}_{\text{init}} \in D_0$ (Theorems 8.2 and 8.3 of Ackleh et al., 2010).

Despite this elegant theory, it is difficult to apply in practice with regards to choosing \mathbf{g} and \mathbf{x}_{init} so that Algorithm 3 converges. This is exemplified in Section 4.2 when we return to the Example 1 optimization problem. We also note that $\|D_{\mathbf{x}}\mathbf{g}(\mathbf{x})\|_{\text{spec}} < 1$ near \mathbf{x}^* is a *sufficient* but not *necessary* condition for convergence of fixed point iteration. Nevertheless, the function

$$\rho(D_{\mathbf{x}}\mathbf{g}(\mathbf{x}))$$

is a useful convergence diagnostic for fixed-point iteration. For instance, if $\rho(D_{\mathbf{x}}\mathbf{g}(\mathbf{x})) \gg 1$ during the iterations then this would indicate convergence problems and the possibility of non-existence of a fixed point \mathbf{x}^* .

Various adjustments to fixed-point iteration have been proposed to enhance convergence. For example, in the context of semiparametric mean field variational Bayes, Section 7 of Minka & Knowles (2011) describes a *damping* adjustment.

3.1.1 NEWTON-RAPHSON ITERATION

Newton-Raphson iteration is a special case of fixed-point iteration for optimizing \mathbf{f} with the \mathbf{g} function taking the form

$$\mathbf{g}_{\text{NR}}(\mathbf{x}) = \mathbf{x} - \{H_{\mathbf{x}}\mathbf{f}(\mathbf{x})\}^{-1}D_{\mathbf{x}}\mathbf{f}(\mathbf{x})^T \quad (33)$$

where $H_{\mathbf{x}}\mathbf{f}(\mathbf{x})$ denotes the Hessian matrix of $\mathbf{f}(\mathbf{x})$ as formally defined in Appendix A.1. Assuming existence of $\{H_{\mathbf{x}}\mathbf{f}(\mathbf{x})\}^{-1}$, it is easily shown that $\mathbf{x} = \mathbf{g}_{\text{NR}}(\mathbf{x})$ if and only if $D_{\mathbf{x}}\mathbf{f}(\mathbf{x})^T = \mathbf{0}$. This leads to Algorithm 4, which conveys the generic form of Newton-Raphson iteration.

Initialize: $\mathbf{x} \leftarrow \mathbf{x}_{\text{init}}$ for some $\mathbf{x}_{\text{init}} \in D$.

Cycle:

$$\mathbf{x} \leftarrow \mathbf{x} - \{H_{\mathbf{x}}\mathbf{f}(\mathbf{x})\}^{-1}D_{\mathbf{x}}\mathbf{f}(\mathbf{x})^T$$

until convergence.

Algorithm 4: *The Newton-Raphson algorithm in generic form.*

Some pertinent features of Algorithm 4 are:

- The function \mathbf{g}_{NR} in (33) has the property

$$\rho(D_{\mathbf{x}}\mathbf{g}_{\text{NR}}(\mathbf{x}^*)) = 0 \quad (34)$$

for stationary points \mathbf{x}^* . A proof is given in Appendix A.4. Therefore, via Theorem 8.4 of Ackleh et al. (2010), convergence to \mathbf{x}^* is guaranteed from a sufficiently close \mathbf{x}_{init} .

- If \mathbf{x}_{init} is such that Algorithm 4 is convergent to \mathbf{x}^* then, under certain regularity conditions, convergence is *quadratic*, in that the number of significant figures doubles on each iteration.

- Locating \mathbf{x}_{init} values sufficiently close to \mathbf{x}^* for convergence to occur can be difficult in practice and it is common to combine Newton-Raphson iteration with more robust optimization strategies, such as the Nelder-Mead simplex method.
- A disadvantage of Newton-Raphson iteration compared with other fixed-point iterative schemes is the requirement for second order partial derivatives, corresponding to the entries of the Hessian matrix $\mathbf{H}_x f(\mathbf{x})$. A feeling for the type of additional calculus needed is given in Appendix A.7.
- A variant of Newton-Raphson optimization known as *damped* Newton-Raphson employs line searches (or backtracking) in order to achieve much improved convergence behavior. See, e.g., Section 9.5.2 of Boyd and Vandenberghe (2004).

3.2 Nonlinear Conjugate Gradient Method

The *nonlinear conjugate gradient method* is based on the *conjugate gradient method*, an established iterative approach to solving large systems of linear equations (Hestenes and Stiefel, 1952). The former arises from applying the latter to the linear system that arises from optimization of a multivariate quadratic function. Details of the nonlinear conjugate gradient method are given in Section 10.8 of Press et al. (2007). Algorithm 5 lists the *Polak-Ribière* version of the nonlinear conjugate gradient method for *maximization* of f over D . Since minimization of f is equivalent to maximization of $-f$ it is straightforward to adapt Algorithm 5 to the minimization problem. We choose the Polak-Ribière form here, but another one is the *Fletcher-Reeves* form given by $\beta \leftarrow (\mathbf{v}_{\text{cur}}^T \mathbf{v}_{\text{arr}}) / (\mathbf{v}_{\text{prev}}^T \mathbf{v}_{\text{prev}})$.

Initialize: $\mathbf{x} \leftarrow \mathbf{x}_{\text{init}}$ for some $\mathbf{x}_{\text{init}} \in D$.

$$\mathbf{v}_{\text{prev}} \leftarrow \mathbf{D}_x f(\mathbf{x})^T ; \alpha \leftarrow \underset{\alpha > 0}{\operatorname{argmax}} f(\mathbf{x} + \alpha \mathbf{v}_{\text{prev}})$$

$$\mathbf{x} \leftarrow \mathbf{x} + \alpha \mathbf{v}_{\text{prev}} ; \mathbf{s} \leftarrow \mathbf{v}_{\text{prev}}$$

Cycle:

$$\mathbf{v}_{\text{cur}} \leftarrow \mathbf{D}_x f(\mathbf{x})^T ; \beta \leftarrow \mathbf{v}_{\text{cur}}^T (\mathbf{v}_{\text{cur}} - \mathbf{v}_{\text{prev}}) / (\mathbf{v}_{\text{prev}}^T \mathbf{v}_{\text{prev}})$$

$$\mathbf{s} \leftarrow \beta \mathbf{s} + \mathbf{v}_{\text{cur}} ; \alpha \leftarrow \underset{\alpha > 0}{\operatorname{argmax}} f(\mathbf{x} + \alpha \mathbf{s})$$

$$\mathbf{x} \leftarrow \mathbf{x} + \alpha \mathbf{s} ; \mathbf{v}_{\text{prev}} \leftarrow \mathbf{v}_{\text{cur}}$$

until convergence.

Algorithm 5: *The nonlinear conjugate gradient method for maximization of the function f with the Polak-Ribière form of the β parameter.*

A key aspect of the nonlinear conjugate gradient method is that, on each iteration, it takes a step in the direction $\mathbf{D}f(\mathbf{x})^T$ from the current position at \mathbf{x} . This is the steepest instantaneous direction on the f surface. The parameter denoted by β has several alter-

native forms. Nonlinear conjugate gradient methods have been shown to have good global convergence properties (Dai and Yuan, 1999).

3.3 Other Optimization Strategies

Other popular optimization strategies include *ascent* (or *descent*) *algorithms* (e.g. Boyd and Vandenberghe, 2004, Section 9.3), *quasi-Newton methods* (e.g. Givens and Hoeting, 2005, Section 2.2.2.3), the *Gauss-Newton method* (e.g. Givens and Hoeting, 2005, Section 2.2.3), *stochastic gradient descent* (e.g. Botton, 2004) and the *Nelder-Mead simplex method* (Nelder and Mead, 1965). The last of these has the attraction of not requiring derivatives of f and is generally more robust than derivative-based methods.

3.4 Application to Semiparametric Mean Field Variational Bayes

We now focus on the optimization component of Algorithm 2

$$\underset{\xi \in \Xi}{\operatorname{argmax}} \left\{ \log \underline{p}(\mathcal{Q}; q, \xi)^{|\phi|} \right\} \quad (35)$$

and discuss ways in which numerical optimization strategies described in Sections 3.1–3.3 are applicable.

The stationary condition for the maximizer in (35) is

$$\mathbf{D}_\xi \log \underline{p}(\mathcal{Q}; q, \xi)^{|\phi|} = \mathbf{0}$$

and this may be manipulated in any of a number of ways to produce an equation of the form $\xi = \mathbf{g}(\xi)$ for some function \mathbf{g} . Fixed-point iteration Algorithm 3 can then be entertained but, as discussed in Section 3.1, convergence is not guaranteed for arbitrary \mathbf{g} . We study this issue in the context of Examples 1 and 2 in Sections 4.2 and 4.3.

Newton-Raphson iteration involves iterative updates:

$$\xi \leftarrow \xi - \{\mathbf{H}_\xi \log \underline{p}(\mathcal{Q}; q, \xi)^{|\phi|}\}^{-1} \mathbf{D}_\xi \log \underline{p}(\mathcal{Q}; q, \xi)^{|\phi|}$$

and so is a candidate for insertion into Algorithm 2 for updating the pre-specified parametric q -density parameters.

Another alternative is, of course, updating ξ according to one or more iterations of the nonlinear conjugate gradient method given by Algorithm 5, or any other iterative optimization scheme. However, convergence needs to be monitored. For high-dimensional ξ , speed of convergence may be also be an important factor. Next we discuss an adjustment aimed at improving the convergence speed of gradient-based algorithms.

3.4.1 RIEMANNIAN GEOMETRY ADJUSTMENT

As explained in, for example, Section 6.2 of Murray and Rice (1993) the density function family $\{q(\phi; \xi) : \xi \in \Xi\}$ can be viewed as a *submanifold* of a *Riemannian manifold*. Riemannian manifolds do not necessarily have a *flat* Euclidean geometry. For example, the Riemannian manifold corresponding to the Univariate Normal family:

$$\left\{ \frac{1}{\sqrt{2\pi\sigma_q^2(\phi)}} \exp \left\{ -\frac{(\phi - \mu_{q(\phi)})^2}{2\sigma_q^2(\phi)} \right\} : \mu_{q(\phi)} \in \mathbb{R}, \sigma_q^2(\phi) > 0 \right\} \quad (36)$$

has *hyperbolic geometry* (Murray and Rice, 1993, Example 6.6.2) which is *curved*. Therefore notions such as closeness of two members of (36) and steepness of gradients when searching over the parameter space $\Xi = \mathbb{R} \times \mathbb{R}_+$ are not properly captured by the Euclidean geometry notions of distance and slope. Adjustments for the Riemannian geometry of the family often improve convergence of optimization algorithms for solving problems such as (35). More detailed discussion on this issue is given in Section 2.2 of Honkela et al. (2010) and Section 2.3 of Hoffman et al. (2013).

Consider an optimization method that uses gradients of the form

$$D_{\xi} \log \underline{p}(\mathcal{D}; q, \xi)^{|\phi|^T}$$

to find the direction of steepest descent of the objective function $\log \underline{p}(\mathcal{D}; q, \xi)^{|\phi|}$. The Riemannian geometry adjustment is to instead use

$$[-E\{\mathbf{H}_{\xi} \log q(\phi; \xi)\}]^{-1} D_{\xi} \log \underline{p}(\mathcal{D}; q, \xi)^{|\phi|^T} \quad (37)$$

where the premultiplying matrix is the inverse *Fisher information* of $q(\phi; \xi)$. In the Machine Learning literature (e.g. Amari, 1998) (37) is often labeled the *natural* or *Riemannian gradient* of $\log \underline{p}(\mathcal{D}; q, \xi)^{|\phi|}$ with respect to ξ and the corresponding geometry is called *information geometry*. If $q(\phi; \xi)$ corresponds to the Univariate Normal family (36) then the Fisher information matrix is $\text{diag}(\sigma_{q(\phi)}^{-2}, \frac{1}{2}\sigma_{q(\phi)}^{-4})$. Therefore, from (37), the natural gradient of $\log \underline{p}(q; \mu_{q(\phi)}, \sigma_{q(\phi)}^2)^{|\phi|}$ with respect to $(\mu_{q(\phi)}, \sigma_{q(\phi)}^2)$ is given by

$$\begin{bmatrix} \sigma_{q(\phi)}^2 & \frac{\partial \underline{p}(q; \mu_{q(\phi)}, \sigma_{q(\phi)}^2)^{|\phi|}}{\partial \mu_{q(\phi)}} \\ 2\sigma_{q(\phi)}^4 & \frac{\partial \underline{p}(q; \mu_{q(\phi)}, \sigma_{q(\phi)}^2)^{|\phi|}}{\partial \sigma_{q(\phi)}^2} \end{bmatrix}^T. \quad (38)$$

Honkela et al. (2010) is a major contribution to semiparametric mean field variational Bayes methodology and their Algorithm 1 uses the nonlinear conjugate gradient method (Algorithm 5) with natural gradients rather than ordinary gradients. Via both simple examples and numerical studies, they make a compelling case for the use of natural gradients for optimization of the parameters of the pre-specified parametric q -density function.

3.5 Summary of Semiparametric Mean Field Variational Bayes Ramifications

In this section we have discussed several iterative numerical optimization strategies. Any of these are candidates for the updating ξ in the Algorithm 2 cycle. Special mention has been given to the well-known Newton-Raphson iteration since it can achieve very rapid convergence and the nonlinear conjugate gradient method which has been shown to be effective in semiparametric mean field variational Bayes contexts when the Riemannian geometry adjustment of Section 3.4.1 is employed (Honkela et al., 2010).

General fixed-point iteration has been discussed in detail. It has the advantage of yielding particularly simple iterative updates for ξ in Algorithm 2. Established theory shows that the spectral radius of the derivative matrix of the fixed-point function can be

used to assess the quality of the scheme. In Section 4 we will explain how a particular fixed-point iteration scheme, which we call *natural* fixed-point iteration, has attractive properties when $q(\cdot; \xi)$ is an exponential family density function. We will also revisit Examples 1 and 2 in Section 4 and make some comparisons among various numerical optimization strategies. Natural fixed-point iteration is seen to perform particularly well.

4. Exponential Family Special Case

We now focus on the important special case where the parametric density function family $\{q(\phi; \xi) : \xi \in \Xi\}$ can be expressed in exponential family form:

$$q(\phi; \eta) = \exp\{\mathbf{T}(\phi)^T \eta - A(\eta)\} h(\phi), \quad \eta \in H, \quad (39)$$

where η is a one-to-one transformation of ξ and H is the image of Ξ under this transformation. In (39) $A(\eta)$ is called the *log-partition function* and $h(\phi)$ is called the *base measure*. For example, the Univariate Normal density function family used in Example 1:

$$q(\phi; \mu_{q(\phi)}, \sigma_{q(\phi)}^2) = \frac{1}{\sqrt{2\pi\sigma_{q(\phi)}^2}} \exp\left\{-\frac{(\phi - \mu_{q(\phi)})^2}{2\sigma_{q(\phi)}^2}\right\}, \quad \mu_{q(\phi)} \in \mathbb{R}, \sigma_{q(\phi)}^2 > 0,$$

can be expressed as (39) with

$$\mathbf{T}(\phi) = \begin{bmatrix} \phi \\ \phi^2 \end{bmatrix}, \quad \eta \equiv \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = \begin{bmatrix} \mu_{q(\phi)}/\sigma_{q(\phi)}^2 \\ -1/(2\sigma_{q(\phi)}^2) \end{bmatrix}, \quad A(\eta) = -\frac{1}{4}\eta_1^2/\eta_2 - \frac{1}{2}\log(-2\eta_2)$$

and $h(\phi) = (2\pi)^{-1/2}$. The natural parameter space is $H = \{(\eta_1, \eta_2) : \eta_1 \in \mathbb{R}, \eta_2 < 0\}$. Even though semiparametric mean field variational Bayes can involve pre-specification of an arbitrary parametric family, virtually all methodology and examples in the existing literature involves pre-specification within an exponential family. Exponential family distributions also play an important role in the theory of mean field variational Bayes (e.g. Sato, 2001; Beal and Ghahramani, 2006; Wainwright and Jordan, 2008).

Now consider the general factor graph set-up described in Section 2.2.1 with the approximate marginal log-likelihood $\log \underline{p}(\mathcal{D}; q, \eta)^{|\phi|}$ given by (13) but as a function of the natural parameter vector η . Define

$$\text{NonEntropy}\{q(\phi; \eta)\} \equiv \sum_{j \in \text{neighbors}(\phi)} E_{q(\phi; \eta)}\{\log(f_j)\}$$

so that

$$\log \underline{p}(\mathcal{D}; q, \eta)^{|\phi|} = \text{Entropy}\{q(\phi; \eta)\} + \text{NonEntropy}\{q(\phi; \eta)\}.$$

An advantage of working with exponential family density functions is that the entropy takes the simple form:

$$\text{Entropy}\{q(\phi; \eta)\} = A(\eta) - D_{\eta} A(\eta) \eta + E[\exp\{h(\phi)\}]$$

where the notation of Appendix A.2 is being used. Moreover, as shown in Lemma 1 of Appendix A.5, the derivative vector of $\text{Entropy}\{q(\phi; \eta)\}$ is simply

$$D_{\eta} \text{Entropy}\{q(\phi; \eta)\} = -\eta^T \mathbf{H}_{\eta} A(\eta). \quad (40)$$

This implies that the stationary point condition

$$\{D_{\eta} \log \underline{p}(g; \eta)\}^{|\phi|} \}^T = \mathbf{0} \quad (41)$$

is equivalent to

$$\eta = \{H_{\eta} A(\eta)\}^{-1} D_{\eta} \text{NonEntropy}\{q(\phi; \eta)\}^T. \quad (42)$$

Algorithm 1 of Knowles and Minka (2011) is a fixed-point iteration scheme based on (42).

One further interesting and useful connection concerns the *mean parameter* vector

$$\tau \equiv E\{T(\phi)\}$$

which is related to the natural parameter vector via

$$\tau = D_{\eta} A(\eta)^T.$$

Under suitable technical conditions τ is a one-to-one transformation of η . Also the chain rule for differentiation of a smooth function s , listed as Lemma 2 in Appendix A.5, is

$$D_{\eta} s = (D_{\tau} s)(D_{\eta} \tau) = (D_{\tau} s) D_{\eta} \{D_{\eta} A(\eta)^T\} = (D_{\tau} s) H_{\eta} A(\eta).$$

which leads to

$$D_{\tau} s = \{H_{\eta} A(\eta)\}^{-1} D_{\eta} s. \quad (43)$$

Putting all of these relationships together we have:

Result 1 *Let ξ be an arbitrary differentiable one-to-one transformation of η . Then the stationary point condition (41) is equivalent to each the following conditions:*

- (a) $\eta = \{H_{\eta} A(\eta)\}^{-1} D_{\eta} \text{NonEntropy}\{q(\phi; \eta)\}^T$,
- (b) $\eta = \{H_{\eta} A(\eta)\}^{-1} (D_{\eta} \xi)^T D_{\xi} \text{NonEntropy}\{q(\phi; \xi)\}^T$,
- (c) $\eta = D_{\tau} \text{NonEntropy}\{q(\phi; \tau)\}^T$ and
- (d) $\eta = (D_{\tau} \xi)^T D_{\xi} \text{NonEntropy}\{q(\phi; \xi)\}^T$.

We make the following remarks concerning Result 1:

- Result 1(a) immediately gives rise to the following fixed-point iteration scheme in the natural parameter space $\eta \in H$:

$$\eta \leftarrow \{H_{\eta} A(\eta)\}^{-1} D_{\eta} \text{NonEntropy}\{q(\phi; \eta)\}^T. \quad (44)$$

We refer to (44) as the *natural fixed-point iteration scheme* and denote the corresponding fixed-point function by

$$\mathbf{g}_{\text{nat}}(\eta) \equiv \{H_{\eta} A(\eta)\}^{-1} D_{\eta} \text{NonEntropy}\{q(\phi; \eta)\}^T.$$

According to the theory of fixed-point iteration discussed in Section 3.1, convergence of (44) is implied by

$$\rho(D_{\eta} \mathbf{g}_{\text{nat}}(\eta)) < 1$$

in a neighborhood of the maximizer η^* .

- Result 1(b)–(d) offer the possibility of more convenient forms for the fixed-point updates in terms of derivatives of the common parameters or mean parameters. Particularly noteworthy is the fact that Result 1(c)–(d) do not require computation of $\{H_{\eta} A(\eta)\}^{-1}$. We make use of this situation for the Multivariate Normal family in Section 4.1.

- The Fisher information of $q(\phi; \eta)$ is

$$-E\{H_{\eta} \log q(\phi; \eta)\} = H_{\eta} A(\eta)$$

which implies that the natural fixed-point iteration scheme (44) involves updating η according to natural Riemannian gradients of $\text{NonEntropy}(g; \tau)$. From Result 1(c), an equivalent updating scheme is

$$\eta \leftarrow D_{\tau} \text{NonEntropy}\{q(\phi; \tau)\}^T$$

which simply involves updating η according to the direction of maximum slope on the $\text{NonEntropy}(g; \tau)$ surface in the τ space.

- The forms for the stationary point in Result 1 can also be used to derive iterative Newton-Raphson schemes for maximizing $\log \underline{p}(g; \eta)^{|\phi|}$. An example, corresponding to Result 1(a) and optimization within the η space, is

$$\eta \leftarrow \eta - [H_{\eta} \text{NonEntropy}\{q(\phi; \eta)\} - H_{\eta} A(\eta) - (\eta^T \otimes \mathbf{I}) D_{\eta} \text{vec}\{H_{\eta} A(\eta)\}]^{-1} \times [D_{\eta} \text{NonEntropy}\{q(\phi; \eta)\}^T - H_{\eta} A(\eta) \eta].$$

The vec operator flattens a square matrix into a column vector and is defined formally in Appendix A.1.

Any of the other optimization methods mentioned in Section 3 can also be applied to the problem of obtaining

$$\eta^* \equiv \underset{\eta \in H}{\text{argmax}} \{\log \underline{p}(\mathcal{D}; q, \xi)^{|\phi|}\} = \underset{\eta \in H}{\text{argmax}} [A(\eta) - D_{\eta} A(\eta) \eta + \text{NonEntropy}\{q(\phi; \eta)\}]$$

and those involving gradients benefit from the entropy derivative result (40). Additionally, relationship (43) implies that natural (Riemannian) gradients of the objective function in the natural parameter space are equivalent to ordinary Euclidean gradients in the mean parameter space.

4.1 Multivariate Normal Special Case

We now focus on the important special case of $q(\phi; \xi)$ being a d -variate Multivariate Normal density function:

$$q(\phi; \mu_{q(\phi)}, \Sigma_{q(\phi)}) = (2\pi)^{-d/2} |\Sigma_{q(\phi)}|^{-1/2} \exp\{-\frac{1}{2}(\phi - \mu_{q(\phi)})^T \Sigma_{q(\phi)}^{-1} (\phi - \mu_{q(\phi)})\}.$$

Let

$$\xi \equiv \begin{bmatrix} \mu_{q(\phi)} \\ \text{vec}(\Sigma_{q(\phi)}) \end{bmatrix}$$

be the vector of common parameters. An explicit form for the natural fixed point iteration updates in terms of $\mu_{q(\phi)}$ and $\Sigma_{q(\phi)}$ was derived by Wand (2014) and appears as equation (7) there. However Result 1 affords a more direct derivation of the same result, that benefits from (43) and the cancellation of the $H_\eta A(\eta)$ matrix. We can also obtain a neater alternative explicit form by using a differentiation identity, given as Lemma 4 in Appendix A.5. The essence of Lemma 4 is given in the appendix of Oppner and Archambeau (2009).

Result 2 *Natural fixed-point iteration for $q(\phi; \xi)$ corresponding to the $N(\mu_{q(\phi)}, \Sigma_{q(\phi)})$ density function is equivalent to the following updating scheme:*

$$\begin{cases} \mathbf{v}_{q(\phi)} \leftarrow \mathbf{D}_{\mu_{q(\phi)}} \text{NonEntropy}(q; \mu_{q(\phi)}, \Sigma_{q(\phi)})^T \\ \Sigma_{q(\phi)} \leftarrow -\{H_{\mu_{q(\phi)}} \text{NonEntropy}(q; \mu_{q(\phi)}, \Sigma_{q(\phi)})\}^{-1} \\ \mu_{q(\phi)} \leftarrow \mu_{q(\phi)} + \Sigma_{q(\phi)} \mathbf{v}_{q(\phi)}. \end{cases}$$

Appendix A.6 provides details on how Result 2 follows from Result 1.

Result 2 facilitates a semiparametric mean field variational Bayes algorithm that requires only the first and second order derivatives of $\text{NonEntropy}(q; \mu_{q(\phi)}, \Sigma_{q(\phi)})$ with respect to $\mu_{q(\phi)}$. Concrete illustrations are given in Section 4.3 and Appendix A of Menictas and Wand (2015).

In the case where $q(\phi; \xi)$ is the Univariate Normal density function with mean $\mu_{q(\phi)}$ and variance $\sigma_{q(\phi)}^2$ Result 2 leads to the following common parameter updates for the natural fixed point iterative scheme:

$$\begin{cases} \mathbf{v}_{q(\phi)} \leftarrow \frac{\partial \text{NonEntropy}(q; \mu_{q(\phi)}, \sigma_{q(\phi)}^2)}{\partial \mu_{q(\phi)}} \\ \sigma_{q(\phi)}^2 \leftarrow -1 / \left\{ \frac{\partial^2 \text{NonEntropy}(q; \mu_{q(\phi)}, \sigma_{q(\phi)}^2)}{\partial \mu_{q(\phi)}^2} \right\} \\ \mu_{q(\phi)} \leftarrow \mu_{q(\phi)} + \sigma_{q(\phi)}^2 \mathbf{v}_{q(\phi)}. \end{cases} \quad (45)$$

Despite its use of natural gradients, there is no automatic guarantee that iteration of (45) leads to convergence to the maximum of $\log p(\mathcal{D}; q, \xi)^{|\phi|}$ on any given cycle of Algorithm 2. However, the fixed-point iteration theory summarized in Section 3.1 provides some guidance. We now use Example 1 to illustrate this point using the natural fixed-point iteration scheme (45), an alternative simpler fixed-point scheme and a Newton-Raphson scheme.

4.2 Application to Example 1

Consider again the Gumbel random sample example introduced in Section 2.2.2 and the problem of minimum Kullback-Leibler approximation of $p(\phi|\mathbf{x})$ within the Univariate Normal family. As shown there, the optimization problem is encapsulated in (19) and (20). The Newton-Raphson scheme that arises directly from (19) is

$$\begin{bmatrix} \mu_{q(\phi)} \\ \sigma_{q(\phi)}^2 \end{bmatrix} \leftarrow \begin{bmatrix} \mu_{q(\phi)} \\ \sigma_{q(\phi)}^2 \end{bmatrix} - \left\{ H_{\text{Ex1}}^N(\mu_{q(\phi)}, \sigma_{q(\phi)}^2; n, \sum_{i=1}^n e^{-x_i}, \mu_{\phi}, \sigma_{\phi}^2) \right\}^{-1} \\ \times \mathbf{D} f_{\text{Ex1}}^N(\mu_{q(\phi)}, \sigma_{q(\phi)}^2; n, \sum_{i=1}^n e^{-x_i}, \mu_{\phi}, \sigma_{\phi}^2)^T. \quad (46)$$

Differentiation with respect to $(\mu_{\phi}, \sigma_{\phi}^2)$ is suppressed in the \mathbf{D} and \mathbf{H} on the right-hand side of (46). Simple calculus leads to (46) being equivalent to the fixed-point iterative scheme

$$\begin{bmatrix} \mu_{q(\phi)} \\ \sigma_{q(\phi)}^2 \end{bmatrix} \leftarrow \mathbf{g}_{\text{NR}} \left(\begin{bmatrix} \mu_{q(\phi)} \\ \sigma_{q(\phi)}^2 \end{bmatrix}; n, \sum_{i=1}^n e^{-x_i}, \mu_{\phi}, \sigma_{\phi}^2 \right) \quad (47)$$

where

$$\mathbf{g}_{\text{NR}} \left(\begin{bmatrix} x \\ y \end{bmatrix}; a, b, c, d \right) \equiv \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} -b e^{x+\frac{1}{2}y} - 1/d & -\frac{1}{2} b e^{x+\frac{1}{2}y} \\ -\frac{1}{2} b e^{x+\frac{1}{2}y} & -\frac{1}{2y^2} - \frac{1}{4} b e^{x+\frac{1}{2}y} \end{bmatrix}^{-1} \\ \times \begin{bmatrix} a - b e^{x+\frac{1}{2}y} - (x-c)/d \\ \frac{1}{2y} - \frac{1}{2} b e^{x+\frac{1}{2}y} - 1/(2d) \end{bmatrix}.$$

According to (45), the natural fixed-point iteration scheme (44) takes the form (47), but with \mathbf{g}_{NR} replaced by \mathbf{g}_{nat} where

$$\mathbf{g}_{\text{nat}} \left(\begin{bmatrix} x \\ y \end{bmatrix}; a, b, c, d \right) \equiv \begin{bmatrix} x + \{a - b e^{x+\frac{1}{2}y} - (x-c)/d\} / (b e^{x+\frac{1}{2}y} + d^{-1}) \\ y / (b e^{x+\frac{1}{2}y} + d^{-1}) \end{bmatrix}.$$

Lastly, there is the very simple fixed-point iteration scheme that arises from full simplification of $\mathbf{D} f_{\text{Ex1}}^N(\mu_{q(\phi)}, \sigma_{q(\phi)}^2; n, \sum_{i=1}^n e^{-x_i}, \mu_{\phi}, \sigma_{\phi}^2)^T = \mathbf{0}$, and corresponds to fixed points of

$$\mathbf{g}_{\text{simp}} \left(\begin{bmatrix} x \\ y \end{bmatrix}; a, b, c, d \right) \equiv \begin{bmatrix} c + d(a - b e^{x+\frac{1}{2}y}) \\ 1 / (b e^{x+\frac{1}{2}y} + d^{-1}) \end{bmatrix}.$$

In Figure 4 we compare \mathbf{g}_{NR} , \mathbf{g}_{nat} and \mathbf{g}_{simp} in terms of the behavior of the spectral norm function $\rho(\mathbf{D} \mathbf{g}(x, y))$ and convergence of the fixed-point iterative scheme. We simulated data from the $n = 20$ version of the Gumbel random sample model (14) with the value of ϕ set to 0. The sufficient statistic $\sum_{i=1}^{20} \exp(-x_i)$ fully determines f_{Ex1}^N and has a mean of 20. In an effort to exhibit typical behavior, we selected a sample that produced a sufficient statistic value close to this mean. The actual value is $\sum_{i=1}^{20} \exp(-x_i) \approx 19.94$. The hyperparameters were set to $\mu_{\phi} = 0$ and $\sigma_{\phi}^2 = 10^{10}$. The optimal parameters in the minimum Kullback-Leibler Univariate Normal approximation to $p(\phi|\mathbf{x})$ are $(\mu_{q(\phi)}^*, (\sigma_{q(\phi)}^2)^*) \approx (0.2260, 0.0500)$. We set up a 101×101 pixel mesh of $(\mu_{q(\phi)}, \log(\sigma_{q(\phi)}^2))$ values around this optimum with limits $(\mu_{q(\phi)}^* - 5, \mu_{q(\phi)}^* + 5)$ and $(\log\{(\sigma_{q(\phi)}^2)^*/5\}^2, \log\{(5\sigma_{q(\phi)}^2)^*\}^2)$. The upper panels of Figure 4 show the

indicator of $\rho(\mathbf{D} \mathbf{g}(\mu_{q(\phi)}, \sigma_{q(\phi)}^2)) < 1$ for $\mathbf{g} \in \{\mathbf{g}_{\text{NR}}, \mathbf{g}_{\text{nat}}, \mathbf{g}_{\text{simp}}\}$.

The lower panels show the

indicator of fixed-point iteration converging when starting from $(\mu_{q(\phi)}, \sigma_{q(\phi)}^2)$.

The top half of Figure 4 shows, via dark grey shading, that both $\rho(\mathbf{D} \mathbf{g}_{\text{NR}}(\mu_{q(\phi)}, \sigma_{q(\phi)}^2))$ and $\rho(\mathbf{D} \mathbf{g}_{\text{nat}}(\mu_{q(\phi)}, \sigma_{q(\phi)}^2))$ are below 1 in regions around the root. The dark grey region for

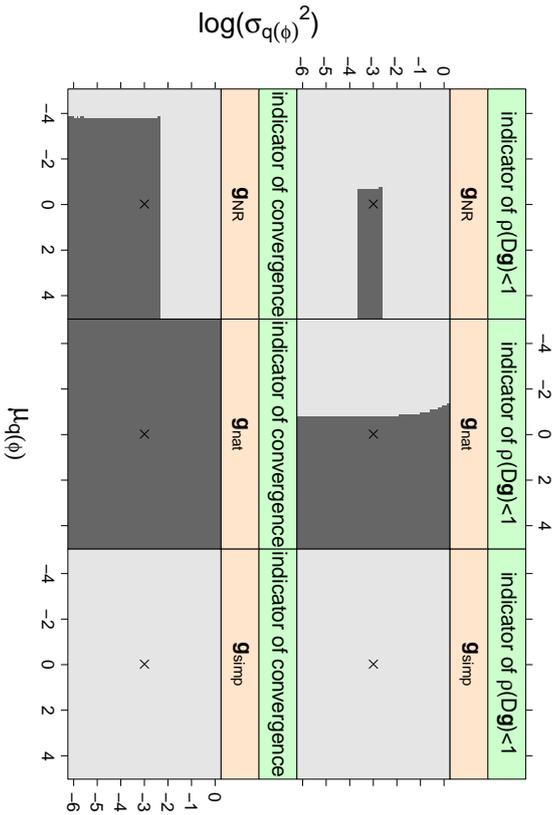


Figure 4: *Upper panels: Dark grey shading shows points where $\rho(\mathbf{Dg}(\mu_{q(\phi)}, \sigma_q^2)) < 1$ for $\mathbf{g} \in \{\mathbf{g}_{\text{NR}}, \mathbf{g}_{\text{nat}}, \mathbf{g}_{\text{simp}}\}$. Lower panels: Dark grey shading shows points from which fixed-point iteration, based on $\mathbf{g} \in \{\mathbf{g}_{\text{NR}}, \mathbf{g}_{\text{nat}}, \mathbf{g}_{\text{simp}}\}$, converges if initialized from that point. The optimum is shown by a cross in each panel and corresponds to minimum Kullback-Leibler divergence for a single $n = 20$ sample of the Gumbel random sample model with hyperparameters set to $\mu_\phi = 0$ and $\sigma_\phi^2 = 10^{10}$.*

\mathbf{g}_{nat} is much larger than that of \mathbf{g}_{NR} , suggesting that the former has better convergence properties according to the theory described in Section 3.1. The lower panels confirm this, with \mathbf{g}_{nat} -based fixed-point iteration converging from every initial value on the pixel mesh, but Newton-Raphson iteration not converging from the sub-region on the top and left-hand side of the mesh. Also note that $\rho(\mathbf{Dg}_{\text{simp}}(\mu_{q(\phi)}, \sigma_q^2)) \geq 1$ over the whole pixel mesh and \mathbf{g}_{simp} -based fixed-point iteration does not converge, regardless of initial point.

Figure 5 shows the iteration trajectories from four different starting values and four iterative schemes based on the same data and hyperparameter values as used in Figure 4. Also shown in each panel is an image plot of the surface being maximized, with a logarithmic scale used for σ_q^2 . In addition to the fixed-point iteration schemes based on \mathbf{g}_{NR} and \mathbf{g}_{nat} we include the nonlinear conjugate gradient method given in Algorithm 5 and

the Riemannian geometry adjustment involving the natural gradients given by (38). In most cases the iterations led to convergence to $(\mu_{q(\phi)}^*, (\sigma_{q(\phi)}^2)^*)$ and the first three iterates are plotted. However, Newton-Raphson failed to converge from the starting values in each of the upper panels and the subsequent iterates are outside of the image plot boundaries.

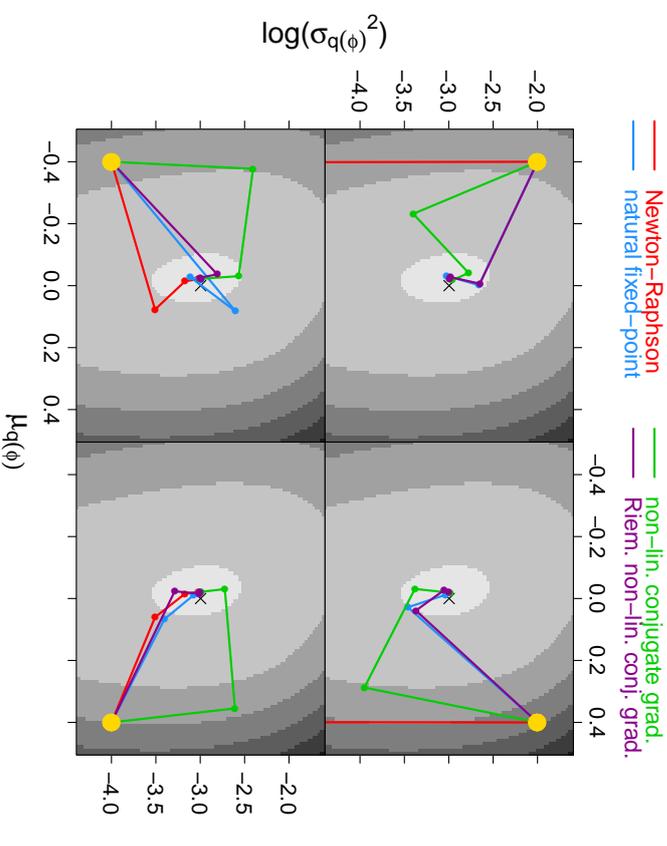


Figure 5: *Iteration trajectories of four iterative algorithms aimed at solving the minimum Kullback-Leibler problem for a Gumbel random sample of size $n = 20$ with hyperparameters $\mu_{q(\phi)} = 0$ and $\sigma_{q(\phi)}^2 = 10^{10}$. The initial value differs for each panel and is shown by the yellow dot. The iterative algorithms are: (1) Newton-Raphson fixed-point iteration based on \mathbf{g}_{NR} , (2) natural fixed-point iteration based on \mathbf{g}_{nat} , (3) ordinary non-linear conjugate gradient method and (4) Riemannian non-linear conjugate gradient method.*

The most striking feature of Figure 5 is the directness with which natural fixed-point iteration and the Riemannian non-linear conjugate gradient method converge from all four starting points and the similarity of their trajectories. This behavior is in keeping with the fact that both work with the more appropriate Riemannian gradients. The ordinary non-linear conjugate gradient trajectories are not as direct. Similar observations are made in Honkela et al. (2010). As demonstrated there, the payoffs from using Riemannian gradients in non-linear conjugate gradient updating are greater in higher-dimensional versions of semiparametric mean field variational Bayes. Based on Figure 5, we anticipate that natural fixed-point iteration is also very good in higher dimensions, and this is corroborated by experiments for Example 2 described in Section 4.3. Newton-Raphson fixed-point iteration is seen to be unreliable for this optimization problem and nowhere near as robust as natural fixed-point iteration. Lastly, we note that the behavior represented in Figures 4 and 5 persists across each of several other samples that we generated.

4.3 Application to Example 2

From (29) and some simple matrix algebra

$$\begin{aligned} \text{NonEntropy}(g; \boldsymbol{\mu}_{q(\beta, \mathbf{u})}, \boldsymbol{\Sigma}_{q(\beta, \mathbf{u})}) &= \mathbf{y}^T \mathbf{C} \boldsymbol{\mu}_{q(\beta, \mathbf{u})} - \mathbf{1}^T \exp \left\{ \mathbf{C} \boldsymbol{\mu}_{q(\beta, \mathbf{u})} + \frac{1}{2} \text{diagonal}(\mathbf{C} \boldsymbol{\Sigma}_{q(\beta, \mathbf{u})} \mathbf{C}^T) \right\} \\ &\quad - \frac{1}{2} \text{tr} \left(\begin{bmatrix} \sigma_\beta^{-2} \mathbf{I}_p & \mathbf{0} \\ \mathbf{0} & \mu_{q(1/\sigma^2)} \mathbf{I}_K \end{bmatrix} \{ \boldsymbol{\mu}_{q(\beta, \mathbf{u})} \boldsymbol{\mu}_{q(\beta, \mathbf{u})}^T + \boldsymbol{\Sigma}_{q(\beta, \mathbf{u})} \} \right) \\ &\quad - \frac{1}{2} (p + K) \log(2\pi) - \frac{1}{2} p \log(\sigma_\beta^2) - \frac{1}{2} K E_q \{ \log(\sigma^2) \} - \mathbf{1}^T \log(\mathbf{y}) \end{aligned}$$

where

$$\mu_{q(1/\sigma^2)} = E_{q(1/\sigma^2)}(1/\sigma^2).$$

The derivatives appearing in Result 2 are

$$\begin{aligned} \mathbf{D}_{\boldsymbol{\mu}_{q(\beta, \mathbf{u})}} \text{NonEntropy}(g; \boldsymbol{\mu}_{q(\beta, \mathbf{u})})^T &= \mathbf{C}^T \left[\mathbf{y} - \exp \left\{ \mathbf{C} \boldsymbol{\mu}_{q(\beta, \mathbf{u})} + \frac{1}{2} \text{diagonal}(\mathbf{C} \boldsymbol{\Sigma}_{q(\beta, \mathbf{u})} \mathbf{C}^T) \right\} \right] \\ &\quad - \begin{bmatrix} \sigma_\beta^{-2} \mathbf{I}_p & \mathbf{0} \\ \mathbf{0} & \mu_{q(1/\sigma^2)} \mathbf{I}_K \end{bmatrix} \boldsymbol{\mu}_{q(\beta, \mathbf{u})} \end{aligned}$$

and

$$\begin{aligned} \mathbf{H}_{\boldsymbol{\mu}_{q(\beta, \mathbf{u})}} \text{NonEntropy}(g; \boldsymbol{\mu}_{q(\beta, \mathbf{u})}) &= \\ - \left(\mathbf{C}^T \text{diag} \left[\exp \left\{ \mathbf{C} \boldsymbol{\mu}_{q(\beta, \mathbf{u})} + \frac{1}{2} \text{diagonal}(\mathbf{C} \boldsymbol{\Sigma}_{q(\beta, \mathbf{u})} \mathbf{C}^T) \right\} \right] \mathbf{C} + \begin{bmatrix} \sigma_\beta^{-2} \mathbf{I}_p & \mathbf{0} \\ \mathbf{0} & \mu_{q(1/\sigma^2)} \mathbf{I}_K \end{bmatrix} \right). \end{aligned}$$

It follows that the updates take the explicit form

$$\begin{cases} \mathbf{w}_{q(\beta, \mathbf{u})} \leftarrow \exp \left\{ \mathbf{C} \boldsymbol{\mu}_{q(\beta, \mathbf{u})} + \frac{1}{2} \text{diagonal}(\mathbf{C} \boldsymbol{\Sigma}_{q(\beta, \mathbf{u})} \mathbf{C}^T) \right\} \\ \boldsymbol{\Sigma}_{q(\beta, \mathbf{u})} \leftarrow \left(\mathbf{C}^T \text{diag} \{ \mathbf{w}_{q(\beta, \mathbf{u})} \} \mathbf{C} + \begin{bmatrix} \sigma_\beta^{-2} \mathbf{I}_p & \mathbf{0} \\ \mathbf{0} & \mu_{q(1/\sigma^2)} \mathbf{I}_K \end{bmatrix} \right)^{-1} \\ \boldsymbol{\mu}_{q(\beta, \mathbf{u})} \leftarrow \boldsymbol{\mu}_{q(\beta, \mathbf{u})} + \boldsymbol{\Sigma}_{q(\beta, \mathbf{u})} \left\{ \mathbf{C}^T (\mathbf{y} - \mathbf{w}_{q(\beta, \mathbf{u})}) - \begin{bmatrix} \sigma_\beta^{-2} \mathbf{I}_p & \mathbf{0} \\ \mathbf{0} & \mu_{q(1/\sigma^2)} \mathbf{I}_K \end{bmatrix} \boldsymbol{\mu}_{q(\beta, \mathbf{u})} \right\}. \end{cases}$$

This is equivalent to the fixed-point iteration scheme

$$\begin{bmatrix} \boldsymbol{\mu}_{q(\beta, \mathbf{u})} \\ \text{vech}(\boldsymbol{\Sigma}_{q(\beta, \mathbf{u})}) \end{bmatrix} \leftarrow \mathbf{g}_{\text{Ex2}} \left(\begin{bmatrix} \boldsymbol{\mu}_{q(\beta, \mathbf{u})} \\ \text{vech}(\boldsymbol{\Sigma}_{q(\beta, \mathbf{u})}) \end{bmatrix} ; \mathbf{y}, \mathbf{C}, \begin{bmatrix} \sigma_\beta^{-2} \mathbf{I}_p & \mathbf{0} \\ \mathbf{0} & \mu_{q(1/\sigma^2)} \mathbf{I}_K \end{bmatrix} \right)$$

where

$$\mathbf{g}_{\text{Ex2}} \left(\begin{bmatrix} \boldsymbol{\mu} \\ \text{vech}(\boldsymbol{\Sigma}) \end{bmatrix} ; \mathbf{y}, \mathbf{C}, \mathbf{M} \right) \equiv \begin{bmatrix} \boldsymbol{\mu} + \left[\mathbf{C}^T \text{diag} \{ \boldsymbol{\omega}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \} \mathbf{C} + \mathbf{M} \right]^{-1} \\ \times \left[\mathbf{C}^T \{ \mathbf{y} - \boldsymbol{\omega}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \} - \mathbf{M} \boldsymbol{\mu} \right] \\ \text{vech} \left(\left[\mathbf{C}^T \text{diag} \{ \boldsymbol{\omega}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \} \mathbf{C} + \mathbf{M} \right]^{-1} \right) \end{bmatrix}$$

and

$$\boldsymbol{\omega}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \equiv \exp \{ \mathbf{C} \boldsymbol{\mu} + \frac{1}{2} \text{diagonal}(\mathbf{C} \boldsymbol{\Sigma} \mathbf{C}^T) \}.$$

Note that the vech operator stores the unique entries of a symmetric matrix in a column vector. A formal definition of vech is given in Appendix A.1.

We simulated data according to the following special case of the Poisson mixed model:

$$\begin{aligned} y_{ij} | U_i &\sim \text{Poisson} \{ \exp(\beta_0 + \beta_1 x_{ij} + U_i) \}, \quad U_i | \sigma^2 \sim N(0, \sigma^2), \\ 1 \leq i \leq m, \quad 1 \leq j \leq n, \quad \beta &\sim N(\mathbf{0}, \sigma_\beta^2 \mathbf{I}), \\ \sigma^2 | a &\sim \text{Inverse-Gamma}(\tfrac{1}{2}, 1/a), \quad a \sim \text{Inverse-Gamma}(\tfrac{1}{2}, A^{-2}). \end{aligned} \quad (48)$$

The hyperparameters were set at $\sigma_\beta = A = 10^5$ and the sample sizes were $m = 30$, $n = 5$. Note that (48) is a special case of (21) with $\mathbf{Z} = \mathbf{I}_m \otimes \mathbf{I}_n$, where \mathbf{I}_n is the $n \times 1$ vector with all entries equal to one. We then ran Algorithm 2 with $q(\beta, \mathbf{u})$ pre-specified to be the $N(\boldsymbol{\mu}_{q(\beta, \mathbf{u})}, \boldsymbol{\Sigma}_{q(\beta, \mathbf{u})})$ density function and a single natural fixed-point iteration in each cycle based on \mathbf{g}_{Ex2} . The fixed-point iteration search over values of $[\boldsymbol{\mu}_{q(\beta, \mathbf{u})}^T \text{vech}(\boldsymbol{\Sigma}_{q(\beta, \mathbf{u})})]^T$ is within an open subset of \mathbb{R}^{560} .

Figure 6 shows trace plots of $\log \bar{p}(\mathbf{y}; q, \boldsymbol{\mu}_{q(\beta, \mathbf{u})}, \boldsymbol{\Sigma}_{q(\beta, \mathbf{u})})$ and $\rho(\mathbf{D} \mathbf{g}_{\text{Ex2}})$, based on the explicit expressions for $\mathbf{D} \mathbf{g}_{\text{Ex2}}$ given in Appendix A.7. The upper panel indicates that the algorithm becomes close to convergence after 6–10 iterations. After the same number of iterations the values of $\rho(\mathbf{D} \mathbf{g}_{\text{Ex2}})$ fall below 1 and settle at about 0.15.

Before leaving this example we note that numerical checks indicate that the optimal $\boldsymbol{\Sigma}_{q(\beta, \mathbf{u})}$ matrix is approximately sparse, with dominant diagonal entries. This implies the possibility of low-rank approximations to the above semiparametric mean field variational Bayes algorithm given, as described in Section 4.1.3 of Challinor and Barber (2013).

5. A Non-Exponential Family Example

In the previous section it was seen that semiparametric mean field variational Bayes with $q(\phi; \boldsymbol{\xi})$ having an exponential family form (39) leads to simplifications of the Kullback-Leibler minimization problem. However, $q(\phi; \boldsymbol{\xi})$ does not have to be restricted in this way. In this section we illustrate semiparametric mean field variational Bayes with the q -density of ϕ specified to be in a non-exponential family: the family of Skew-Normal density functions (Azzalini and Dalla Valle, 1996). Even though this family has a multivariate extension,

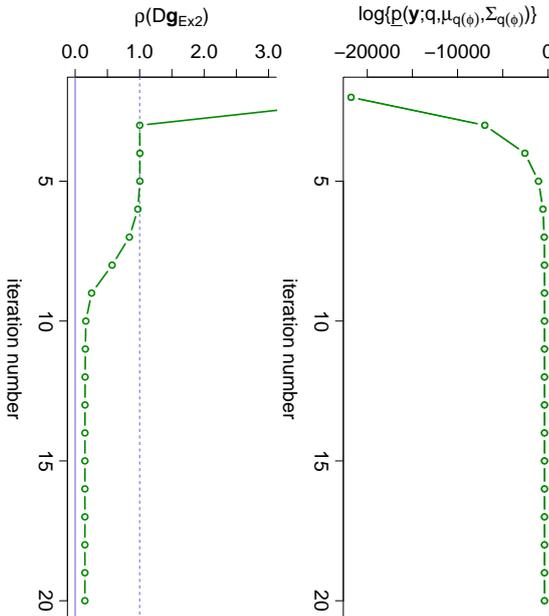


Figure 6: Trace plots of $\log p(\mathbf{y}; q, \boldsymbol{\xi})^{(\beta; u)}$ and $\rho(\mathbf{Dg}_{\text{Ex2}})$ for the version of the Poisson mixed model given by (48) with sample sizes $m = 30$ and $n = 5$.

we restrict attention to the univariate case and, in particular, its use within the context of Example 1.

Specification of $q(\phi; \boldsymbol{\xi})$ being within the family of univariate Skew-Normal density functions entails having

$$q(\phi; \boldsymbol{\xi}) = \sqrt{\frac{2}{\pi\sigma_{q(\phi)}^2}} \exp \left\{ -\frac{(\phi - \mu_{q(\phi)})^2}{2\sigma_{q(\phi)}^2} \right\} \Phi \left\{ \frac{\lambda_{q(\phi)}(\phi - \mu_{q(\phi)})}{\sigma_{q(\phi)}} \right\} \quad (49)$$

where $\Phi(x) \equiv (2\pi)^{-1/2} \int_{-\infty}^x e^{-t^2/2} dt$ is the $N(0, 1)$ cumulative distribution function. The q -density parameter vector is $\boldsymbol{\xi} = (\mu_{q(\phi)}, \sigma_{q(\phi)}^2, \lambda_{q(\phi)})$ and the corresponding parameter space is $\boldsymbol{\Xi} = \mathbb{R} \times \mathbb{R}_+ \times \mathbb{R}$. Now consider the Example 1 setting with $q(\phi; \boldsymbol{\xi})$ restricted to the Skew-Normal family (49). The marginal log-likelihood lower bound, given by (17) and (18),

depends on the explicit expressions

$$E_{q(\phi; \boldsymbol{\xi})}(\phi) = \mu_{q(\phi)} + \frac{\sigma_{q(\phi)} \lambda_{q(\phi)}}{\sqrt{(\pi/2)(1 + \lambda_{q(\phi)}^2)}}, \quad \text{Var}_{q(\phi; \boldsymbol{\xi})}(\phi) = \sigma_{q(\phi)}^2 \left\{ 1 - \frac{2\lambda_{q(\phi)}^2}{\pi(1 + \lambda_{q(\phi)}^2)} \right\}$$

$$\text{and } M_{q(\phi; \boldsymbol{\xi})}(t) = 2 \exp \left(\mu_{q(\phi)} + \frac{1}{2} \sigma_{q(\phi)}^2 t^2 \right) \Phi \left(\frac{\lambda_{q(\phi)} \sigma_{q(\phi)} t}{\sqrt{1 + \lambda_{q(\phi)}^2}} \right)$$

where, as defined in Section 2.2.2, $M_{q(\phi; \boldsymbol{\xi})}$ is the moment generating function corresponding to $q(\phi; \boldsymbol{\xi})$. It also depends on

$$\text{Entropy}\{q(\phi; \boldsymbol{\xi})\} = \frac{1}{2} \{1 + \log(\pi/2) + \log(\sigma_{q(\phi)}^2)\} - \sqrt{\frac{2}{\pi}} \int_{-\infty}^{\infty} \log \Phi(\lambda_{q(\phi)} t) \Phi(\lambda_{q(\phi)} t) e^{-t^2/2} dt$$

which does not simplify any further. Plugging these expressions into (17) and (18) we get the Kullback-Leibler optimal Skew-Normal q -density function is $q(\phi; \mu_{q(\phi)}^*, (\sigma_{q(\phi)}^2)^*, \lambda_{q(\phi)}^*)$ where

$$\begin{bmatrix} \mu_{q(\phi)}^* \\ (\sigma_{q(\phi)}^2)^* \\ \lambda_{q(\phi)}^* \end{bmatrix} = \underset{\substack{\mu_{q(\phi)} \in \mathbb{R}, \sigma_{q(\phi)}^2 \in \mathbb{R} \\ \lambda_{q(\phi)} \in \mathbb{R}}} {\text{argmax}} \left\{ f_{\text{Ex1}}^{\text{SN}} \left(\mu_{q(\phi)}, \sigma_{q(\phi)}^2, \lambda_{q(\phi)}; n, \sum_{i=1}^n e^{-x_i}, \mu_{\phi}, \sigma_{\phi}^2 \right) \right\} \quad (50)$$

and

$$\begin{aligned} f_{\text{Ex1}}^{\text{SN}}(x, y, z; a, b, c, d) &= \frac{1}{2} \log(y) - \sqrt{\frac{2}{\pi}} \int_{-\infty}^{\infty} \log\{\Phi(zt)\} \Phi(zt) e^{-t^2/2} dt \\ &\quad + a \left\{ x + z \sqrt{\frac{2y}{\pi(1+z^2)}} \right\} - 2b \exp\left(x + \frac{1}{2}y\right) \Phi\left(\frac{z\sqrt{y}}{\sqrt{z^2+1}}\right) \\ &\quad - \frac{1}{2d} \left\{ x + z \sqrt{\frac{2y}{\pi(1+z^2)}} - c \right\}^2 + y \left\{ 1 - \frac{2z^2}{\pi(1+z^2)} \right\}. \end{aligned}$$

Optimization problem (50) is considerably more challenging than its Normal counterpart. In particular, evaluations of the objective function and its derivatives require numerical integration.

We solved (50) for three Gumbel random samples of size $n = 5, 10$ and 20 and with $\sum_{i=1}^n e^{-x_i} \approx n$, corresponding to the mean of this sufficient statistic. The intractable integral in $f_{\text{Ex1}}^{\text{SN}}$ was approximated using a trapezoidal quadrature scheme similar to that described in Appendix B.2 of Wand et al. (2011). The limits of the trapezoidal grid were increased until the ratio of the global maximum and minimum absolute values of the integrand fell below 10^{-20} . The number of grid points was then doubled until the relative difference between two successive iterations was less than 10^{-20} . Multiple start locations and simulated annealing were used to locate global optima. Natural fixed-point iteration no

longer applies in this non-exponential family example and optimization of $f_{\text{Ex1}}^{\text{SN}}$ was accomplished using the *Broyden-Fletcher-Goldfarb-Shanno* quasi-Newton method via the `opt.im()` function in the R computing environment (R Development Core Team, 2016).

Figure 7 shows the optimal Skew-Normal q -density functions, together with the exact posterior density functions and those based on the Normal q -density restriction. We see that the Normal approximation is inferior for very low sample sizes, but that the approximations are about the same for moderate to large sample sizes.

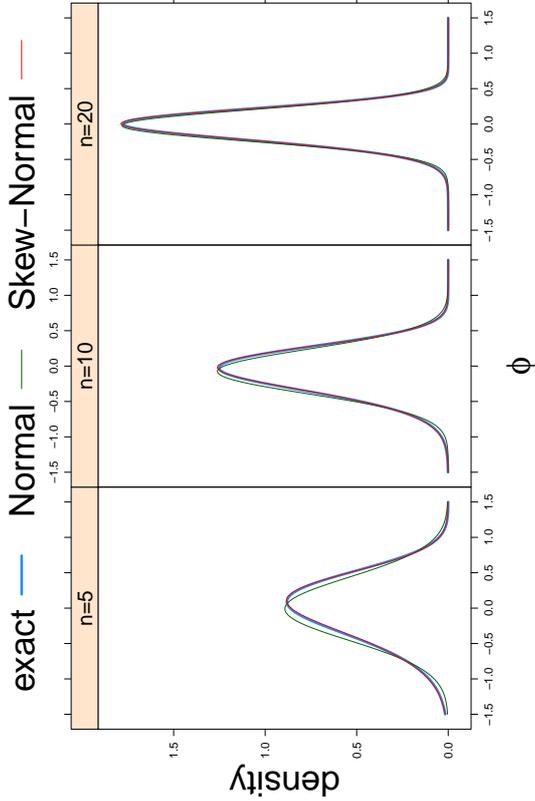


Figure 7: *Skew-Normal minimum Kullback-Leibler approximate posterior density functions for samples of size $n = 5, 10$ and 20 for the Example 1 Gumbel random sample setting. The exact posterior density functions and those based on restriction to the Normal family are also shown.*

6. Closing Remarks

We have taken a broad view of mean field variational Bayes with parametric pre-specification of one of the q -density components and coined the term ‘semiparametric mean field variational Bayes’ for this general approach. As well as laying out the general principles of

semiparametric mean field variational Bayes, we have provided an overview of the numerical issues attached to this methodology. Natural fixed-point iteration has been identified as a promising general approach to dealing with the Kullback-Leibler optimization problem and its attractive Riemannian gradient properties have been elucidated. Proof of convergence of a particular semiparametric mean field variational Bayes strategy appears to be too difficult a goal. However, for fixed-point iteration strategies, the spectral radius of the derivative matrix of the fixed-point update function is a reasonable diagnostic measure for checking convergence.

Acknowledgments

This research was partially supported by Australian Research Council Discovery Project DP110100061. We are grateful to Frances Kuo, Christian Wolff and Rob Womersley for their advice on aspects of this research. We also thank the editor and referees for their helpful comments.

Appendix A. Definitions and Derivations

A.1 Matrix definitions and identity

If \mathbf{A} is a $d \times d$ matrix then $\text{vec}(\mathbf{A})$ is the $d^2 \times 1$ vector obtained by stacking the columns of \mathbf{A} underneath each other in order from left to right. The inverse vec operator is denoted by vec^{-1} . In addition we let $\text{vech}(\mathbf{A})$ denote the $\frac{1}{2}d(d+1) \times 1$ vector obtained from $\text{vec}(\mathbf{A})$ by eliminating the above-diagonal entries of \mathbf{A} . If \mathbf{A} is symmetric then $\text{vech}(\mathbf{A})$ contains all of the unique entries of \mathbf{A} .

The derivations also require the commutation and duplication matrix notation of Magnus and Neudecker (1999). If \mathbf{A} is an arbitrary $d \times d$ matrix then the *commutation matrix* of order d , denoted by \mathbf{K}_d , is the the $d^2 \times d^2$ matrix of zeroes and ones for which

$$\mathbf{K}_d \text{vec}(\mathbf{A}) = \text{vec}(\mathbf{A}^T).$$

If \mathbf{B} is a symmetric but otherwise arbitrary $d \times d$ matrix then the *duplication matrix* of order d is the $d^2 \times \frac{1}{2}d(d+1)$ matrix of zeroes and ones for which

$$\mathbf{D}_d \text{vech}(\mathbf{B}) = \text{vec}(\mathbf{B}).$$

The Moore-Penrose inverse of \mathbf{D}_d is

$$\mathbf{D}_d^+ \equiv (\mathbf{D}_d^T \mathbf{D}_d)^{-1} \mathbf{D}_d^T.$$

Note that

$$\mathbf{D}_d^+ \text{vec}(\mathbf{B}) = \text{vech}(\mathbf{B}). \quad (51)$$

Another useful notation is

$$\mathbf{Q}(\mathbf{A}) \equiv (\mathbf{A} \otimes \mathbf{1}^T) \odot (\mathbf{1}^T \otimes \mathbf{A})$$

for a general $m \times n$ matrix \mathbf{A} and $\mathbf{1}$ a $n \times 1$ vector of ones. The symbol \odot denotes element-wise product.

The following well-known matrix identity is used several times in the derivations:

$$\text{vec}(\mathbf{ABC}) = (\mathbf{C}^T \otimes \mathbf{A}) \text{vec}(\mathbf{B}). \quad (52)$$

A.2 Derivative Matrix and Hessian Matrix Notation

Our summary of derivative-based optimization, and subsequent discussion, benefits from derivative vector and Hessian matrix notation. Such notation is not universal, and throughout this article we follow the conventions of Magnus and Neudecker (1999).

If \mathbf{h} is a \mathbb{R}^d -valued with argument $\mathbf{x} \in \mathbb{R}^d$ then the *derivative matrix* of \mathbf{h} with respect to \mathbf{x} , denoted by $\mathbf{D}_{\mathbf{x}}\mathbf{h}(\mathbf{x})$, is the $p \times d$ matrix with (i, j) entry

$$\frac{\partial \mathbf{h}(\mathbf{x})}{\partial x_j}$$

A concrete derivative vector example is given in Section 2.3 of Wand (2014).

In the case $p = 1$, the *Hessian matrix* of \mathbf{h} with respect to \mathbf{x} is the $d \times d$ matrix

$$\mathbf{H}_{\mathbf{x}}\mathbf{h}(\mathbf{x}) \equiv \mathbf{D}_{\mathbf{x}}\{\{\mathbf{D}_{\mathbf{x}}\mathbf{h}(\mathbf{x})\}^T\}.$$

A.3 Example 2 Derivational Details

Here provide derivational details pertaining to Example 2 discussed in Section 2.2.3.

According to product restriction (22), the optimal q -density functions satisfy

$$\begin{aligned} q^*(\boldsymbol{\beta}, \mathbf{u}) &\propto \exp[E_{q(\sigma^2, a)}] \log\{p(\mathbf{y}, \boldsymbol{\beta}, \mathbf{u}, \sigma^2, a)\}, \\ q^*(\sigma^2) &\propto \exp[E_{q(\boldsymbol{\beta}, \mathbf{u}, a)}] \log\{p(\mathbf{y}, \boldsymbol{\beta}, \mathbf{u}, \sigma^2, a)\} \\ \text{and } q^*(a) &\propto \exp[E_{q(\boldsymbol{\beta}, \mathbf{u}, \sigma^2)}] \log\{p(\mathbf{y}, \boldsymbol{\beta}, \mathbf{u}, \sigma^2, a)\} \end{aligned}$$

(e.g. Bishop, 2006, Section 10.1.1). Simple algebraic steps lead to the forms given in (23).

First we consider general pre-specified q -density families of the form $q(\boldsymbol{\beta}, \mathbf{u}; \boldsymbol{\xi})$, $\boldsymbol{\xi} \in \Xi$. With the help of (10) each of the terms in (27), can be expressed as follows:

$$\begin{aligned} \text{Entropy}\{q(\boldsymbol{\beta}, \mathbf{u}; \boldsymbol{\xi})\} &= -\int_{\mathbb{R}^{K+2}} \log\{q(\boldsymbol{\beta}, \mathbf{u}; \boldsymbol{\xi})\} q(\boldsymbol{\beta}, \mathbf{u}; \boldsymbol{\xi}) d\boldsymbol{\beta} d\mathbf{u}, \\ \text{Entropy}\{q(\sigma^2)\} &= \log(B_{q(\sigma^2)}) + \frac{1}{2}(K+1) + \log\{\Gamma(\frac{1}{2}(K+1))\} \\ &\quad - \frac{1}{2}(K+3) \text{digamma}\{\frac{1}{2}(K+1)\}, \\ \text{Entropy}\{q(a)\} &= \log(B_{q(a)}) + 1 - 2 \text{digamma}(1), \\ E_{q\{\log p(\mathbf{y}|\boldsymbol{\beta}, \mathbf{u})\}} &= \mathbf{y}^T \{\mathbf{X} E_{q(\boldsymbol{\beta}, \mathbf{u}; \boldsymbol{\xi})}(\boldsymbol{\beta}) + \mathbf{Z} E_{q(\boldsymbol{\beta}, \mathbf{u}; \boldsymbol{\xi})}(\mathbf{u})\} \\ &\quad - \mathbf{1}^T E_{q(\boldsymbol{\beta}, \mathbf{u}; \boldsymbol{\xi})} \{\exp(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\mathbf{u})\} - \mathbf{1}^T \log(\mathbf{y}!), \\ E_{q\{\log p(\boldsymbol{\beta}, \mathbf{u}|\sigma^2)\}} &= -\frac{1}{2}(p+K) \log(2\pi) - \frac{1}{2} p \log(\sigma_{\boldsymbol{\beta}}^2) \\ &\quad - \frac{1}{2} K \{\log\{B_{q(\sigma^2)}\} - \text{digamma}\{\frac{1}{2}(K+1)\}\} \\ &\quad - \frac{1}{2\sigma_{\boldsymbol{\beta}}^2} \|\| E_{q(\boldsymbol{\beta}, \mathbf{u}; \boldsymbol{\xi})}(\boldsymbol{\beta})\|^2 + \text{tr}\{\text{Cov}_{q(\boldsymbol{\beta}, \mathbf{u}; \boldsymbol{\xi})}(\boldsymbol{\beta})\} \\ &\quad - \frac{1}{2} \mu_{q(1/\sigma^2)} \|\| E_{q(\boldsymbol{\beta}, \mathbf{u}; \boldsymbol{\xi})}(\mathbf{u})\|^2 + \text{tr}\{\text{Cov}_{q(\boldsymbol{\beta}, \mathbf{u}; \boldsymbol{\xi})}(\mathbf{u})\}, \\ E_{q\{\log p(\sigma^2|a)\}} &= -\frac{1}{2} \log(\pi) - \frac{1}{2} \{\log\{B_{q(a)}\} - \text{digamma}(1)\} \\ &\quad - \frac{1}{2} \{\log\{B_{q(\sigma^2)}\} - \text{digamma}\{\frac{1}{2}(K+1)\}\} \\ &\quad - \mu_{q(1/a)} \mu_{q(1/\sigma^2)} \\ &\quad - \mu_{q(1/a)}^2 / A^2, \\ \text{and } E_{q\{\log p(a)\}} &= -\frac{1}{2} \log(\pi) - \log(A) - \frac{3}{2} \{\log\{B_{q(a)}\} - \text{digamma}(1)\} \\ &\quad - \mu_{q(1/a)} / A^2. \end{aligned} \quad (53)$$

The $(\boldsymbol{\beta}, \mathbf{u})$ -localized approximate marginal log-likelihood expression given by (28) follows immediately from the relevant terms in (53).

If $q(\boldsymbol{\beta}, \mathbf{u}; \boldsymbol{\xi})$ is specified to be the $N(\boldsymbol{\mu}_{q(\boldsymbol{\beta}, \mathbf{u})}, \boldsymbol{\Sigma}_{q(\boldsymbol{\beta}, \mathbf{u})})$ density function then the terms in (27) that depend on $\boldsymbol{\xi} = (\boldsymbol{\mu}_{q(\boldsymbol{\beta}, \mathbf{u})}, \boldsymbol{\Sigma}_{q(\boldsymbol{\beta}, \mathbf{u})})$ are

$$\begin{aligned} \text{Entropy}\{q(\boldsymbol{\beta}, \mathbf{u}; \boldsymbol{\mu}_{q(\boldsymbol{\beta}, \mathbf{u})}, \boldsymbol{\Sigma}_{q(\boldsymbol{\beta}, \mathbf{u})})\} &= \frac{1}{2}(p+K) \{1 + \log(2\pi)\} + \frac{1}{2} \log\|\boldsymbol{\Sigma}_{q(\boldsymbol{\beta}, \mathbf{u})}\|, \\ E_{q\{\log p(\mathbf{y}|\boldsymbol{\beta}, \mathbf{u})\}} &= \mathbf{y}^T \mathbf{C} \boldsymbol{\mu}_{q(\boldsymbol{\beta}, \mathbf{u})} - \mathbf{1}^T \log(\mathbf{y}!) \\ &\quad - \mathbf{1}^T \exp\{\mathbf{C} \boldsymbol{\mu}_{q(\boldsymbol{\beta})} + \frac{1}{2} \text{diagonal}(\mathbf{C} \boldsymbol{\Sigma}_{q(\boldsymbol{\beta}, \mathbf{u})} \mathbf{C}^T)\} \\ \text{and } E_{q\{\log p(\boldsymbol{\beta}, \mathbf{u}|\sigma^2)\}} &= -\frac{1}{2}(p+K) \log(2\pi) - \frac{1}{2} p \log(\sigma_{\boldsymbol{\beta}}^2) \\ &\quad - \frac{1}{2} K \{\log\{B_{q(\sigma^2)}\} - \text{digamma}\{\frac{1}{2}(K+1)\}\} \\ &\quad - \frac{1}{2\sigma_{\boldsymbol{\beta}}^2} \left\{ \|\boldsymbol{\mu}_{q(\boldsymbol{\beta})}\|^2 + \text{tr}(\boldsymbol{\Sigma}_{q(\boldsymbol{\beta})}) \right\} \\ &\quad - \frac{1}{2} \mu_{q(1/\sigma^2)} \left\{ \|\boldsymbol{\mu}_{q(\boldsymbol{\mu})}\|^2 + \text{tr}(\boldsymbol{\Sigma}_{q(\boldsymbol{\mu})}) \right\} \end{aligned} \quad (54)$$

where $\mathbf{C} \equiv [\mathbf{X} \ \mathbf{Z}]$. The $(\boldsymbol{\beta}, \mathbf{u})$ -localized approximate marginal log-likelihood expression given by (29) follows immediately. An explicit expression for $\log p(q; \boldsymbol{\mu}_{q(\boldsymbol{\beta}, \mathbf{u})}, \boldsymbol{\Sigma}_{q(\boldsymbol{\beta}, \mathbf{u})})$, for use as a stopping criterion, can be formed by combining the relevant terms from (53) with those in (54).

A.4 Proof of (34)

In this proof, all appearances of D and H are assumed to be with respect to \mathbf{x} . Let

$$\mathbf{g}_{\text{NR}}(\mathbf{x}) \equiv \mathbf{x} - \{Hf(\mathbf{x})\}^{-1} Df(\mathbf{x})^T.$$

Then, using (52),

$$\begin{aligned} d\mathbf{g}_{\text{NR}}(\mathbf{x}) &= d\mathbf{x} + \{Hf(\mathbf{x})\}^{-1} \{dHf(\mathbf{x})\} \{Hf(\mathbf{x})\}^{-1} Df(\mathbf{x})^T \\ &\quad - \{Hf(\mathbf{x})\}^{-1} dDf(\mathbf{x})^T \\ &= d\mathbf{x} + \{Hf(\mathbf{x})\}^{-1} \text{vec}^{-1} [D\text{vec}\{Hf(\mathbf{x})\} d\mathbf{x}] \{Hf(\mathbf{x})\}^{-1} Df(\mathbf{x})^T \\ &\quad - \{Hf(\mathbf{x})\}^{-1} Hf(\mathbf{x}) d\mathbf{x} \\ &= \{Hf(\mathbf{x})\}^{-1} \text{vec} \left(\mathbf{I} \text{vec}^{-1} [D\text{vec}\{Hf(\mathbf{x})\} d\mathbf{x}] \{Hf(\mathbf{x})\}^{-1} Df(\mathbf{x})^T \right) \\ &= \{Hf(\mathbf{x})\}^{-1} \left([Df(\mathbf{x})\{Hf(\mathbf{x})\}^{-1}] \otimes \mathbf{I} \right) D\text{vec}\{Hf(\mathbf{x})\} d\mathbf{x}. \end{aligned}$$

Therefore

$$D\mathbf{g}_{\text{NR}}(\mathbf{x}) = \{Hf(\mathbf{x})\}^{-1} \left([Df(\mathbf{x})\{Hf(\mathbf{x})\}^{-1}] \otimes \mathbf{I} \right) D\text{vec}\{Hf(\mathbf{x})\}.$$

Since $Df(\mathbf{x}^*) = \mathbf{0}$, we get

$$D\mathbf{g}_{\text{NR}}(\mathbf{x}^*) = \mathbf{O},$$

where \mathbf{O} is the $d \times d$ matrix with all entries equal to zero, and (34) follows immediately.

A.5 Lemmas and Proofs Required for Results 1 and 2

Lemma 1 If

$$q(\mathbf{x}; \boldsymbol{\eta}) = \exp\{\mathbf{T}(\mathbf{x})^T \boldsymbol{\eta} - A(\boldsymbol{\eta})\} h(\mathbf{x})$$

is an exponential family density function then

$$D_{\boldsymbol{\eta}} \text{Entropy}\{q(\mathbf{x}; \boldsymbol{\eta})\} = -\boldsymbol{\eta}^T H_{\boldsymbol{\eta}} A(\boldsymbol{\eta}).$$

Proof of Lemma 1

Since

$$\text{Entropy}\{q(\mathbf{x}; \boldsymbol{\eta})\} = A(\boldsymbol{\eta}) - D_{\boldsymbol{\eta}} A(\boldsymbol{\eta}) \boldsymbol{\eta} - E[\log\{h(\mathbf{x})\}]$$

we then have

$$D_{\boldsymbol{\eta}} \text{Entropy}\{q(\mathbf{x}; \boldsymbol{\eta})\} = D_{\boldsymbol{\eta}} A(\boldsymbol{\eta}) - D_{\boldsymbol{\eta}} \{D_{\boldsymbol{\eta}} A(\boldsymbol{\eta}) \boldsymbol{\eta}\} = D_{\boldsymbol{\eta}} A(\boldsymbol{\eta}) - D_{\boldsymbol{\eta}} \{\boldsymbol{\eta}^T D_{\boldsymbol{\eta}} A(\boldsymbol{\eta})^T\}.$$

Next,

$$\begin{aligned} d\{\boldsymbol{\eta}^T D_{\boldsymbol{\eta}} A(\boldsymbol{\eta})^T\} &= (d\boldsymbol{\eta})^T D_{\boldsymbol{\eta}} A(\boldsymbol{\eta})^T + \boldsymbol{\eta}^T d\{D_{\boldsymbol{\eta}} A(\boldsymbol{\eta})^T\} \\ &= D_{\boldsymbol{\eta}} A(\boldsymbol{\eta}) d\boldsymbol{\eta} + \boldsymbol{\eta}^T D_{\boldsymbol{\eta}} \{D_{\boldsymbol{\eta}} A(\boldsymbol{\eta})^T\} d\boldsymbol{\eta} \\ &= \{D_{\boldsymbol{\eta}} A(\boldsymbol{\eta}) + \boldsymbol{\eta}^T H_{\boldsymbol{\eta}} A(\boldsymbol{\eta})\} d\boldsymbol{\eta} \end{aligned}$$

and so

$$D_{\boldsymbol{\eta}} \{\boldsymbol{\eta}^T D_{\boldsymbol{\eta}} A(\boldsymbol{\eta})^T\} = D_{\boldsymbol{\eta}} A(\boldsymbol{\eta}) + \boldsymbol{\eta}^T H_{\boldsymbol{\eta}} A(\boldsymbol{\eta}).$$

Hence,

$$D_{\boldsymbol{\eta}} \text{Entropy}\{q(\mathbf{x}; \boldsymbol{\eta})\} = D_{\boldsymbol{\eta}} A(\boldsymbol{\eta}) - \{D_{\boldsymbol{\eta}} A(\boldsymbol{\eta}) + \boldsymbol{\eta}^T H_{\boldsymbol{\eta}} A(\boldsymbol{\eta})\} = -\boldsymbol{\eta}^T H_{\boldsymbol{\eta}} A(\boldsymbol{\eta}).$$

Lemma 2 Let s be a differentiable scalar-valued function of $\mathbf{x} \in \mathbb{R}^d$ and let $\mathbf{u} \in \mathbb{R}^k$ be one-to-one transformation of \mathbf{x} . Then

$$D_{\mathbf{x}} s = (D_{\mathbf{u}} s) (D_{\mathbf{x}} \mathbf{u}).$$

Proof of Lemma 2

Lemma 2 is a restatement of Theorem 8, Chapter 5, of Magnus and Neudecker (1999).

Lemma 3 Let $\mathbf{x} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ have a d -dimensional Multivariate Normal distribution. The natural statistic is

$$\mathbf{T}(\mathbf{x}) = \begin{bmatrix} \mathbf{x} \\ \text{vech}(\mathbf{x}\mathbf{x}^T) \end{bmatrix}$$

and corresponding mean parameter is $\boldsymbol{\tau} \equiv E\{\mathbf{T}(\mathbf{x})\}$. Then

$$D_{\boldsymbol{\tau}} \begin{bmatrix} \boldsymbol{\mu} \\ \text{vec}(\boldsymbol{\Sigma}) \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -(\mathbf{I} + \mathbf{K}_d)(\boldsymbol{\mu} \otimes \mathbf{I}) & D_d \end{bmatrix}.$$

Proof of Lemma 3

The transformation from the common parameters to the mean parameters is

$$\boldsymbol{\tau} \equiv \begin{bmatrix} \boldsymbol{\tau}_1 \\ \boldsymbol{\tau}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{\mu} \\ \text{vech}(\boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^T) \end{bmatrix}$$

and the inverse transformation is easily shown to be

$$\begin{bmatrix} \boldsymbol{\mu} \\ \text{vec}(\boldsymbol{\Sigma}) \end{bmatrix} = \begin{bmatrix} D_d \boldsymbol{\tau}_2 - \text{vec}(\boldsymbol{\tau}_1 \boldsymbol{\tau}_1^T) \\ \boldsymbol{\tau}_1 \end{bmatrix}.$$

Hence

$$D_{\boldsymbol{\tau}} \begin{bmatrix} \boldsymbol{\mu} \\ \text{vec}(\boldsymbol{\Sigma}) \end{bmatrix} = \begin{bmatrix} D_{\boldsymbol{\tau}_1} \boldsymbol{\mu} & D_{\boldsymbol{\tau}_2} \boldsymbol{\mu} \\ D_{\boldsymbol{\tau}_1, \text{vec}(\boldsymbol{\Sigma})} & D_{\boldsymbol{\tau}_2, \text{vec}(\boldsymbol{\Sigma})} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -D_{\boldsymbol{\tau}_1, \text{vec}(\boldsymbol{\tau}_1 \boldsymbol{\tau}_1^T)} & D_d \end{bmatrix}.$$

To obtain an explicit expression for the bottom left-hand block we note that

$$d\text{vec}(\boldsymbol{\tau}_1 \boldsymbol{\tau}_1^T) = (\mathbf{I} + \mathbf{K}_d) \text{vec}\{d(\boldsymbol{\tau}_1) \boldsymbol{\tau}_1^T\}.$$

Then, with the help of (52),

$$\text{vec}\{d(\boldsymbol{\tau}_1) \boldsymbol{\tau}_1^T\} = \text{vec}\{\mathbf{I}(d\boldsymbol{\tau}_1) \boldsymbol{\tau}_1^T\} = (\boldsymbol{\tau}_1 \otimes \mathbf{I}) d\boldsymbol{\tau}_1.$$

Hence,

$$D_{\tau_1} \text{vec}(\Sigma) = (\mathbf{I} + \mathbf{K}_d)(\tau_1 \otimes \mathbf{I}) = (\mathbf{I} + \mathbf{K}_d)(\mu \otimes \mathbf{I})$$
and the lemma follows immediately.

Lemma 4 *Let*

$$\phi(\mathbf{x}; \mu, \Sigma) \equiv (2\pi)^{-d/2} |\Sigma|^{-1/2} \exp\left\{-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)\right\}$$

denote the d -variate $N(\mu, \Sigma)$ density function. Then

$$\text{vec}^{-1}\left\{D_{\text{vec}(\Sigma)} \phi(\mathbf{x}; \mu, \Sigma)^T\right\} = \frac{1}{2} \mathbf{H}_\mu \phi(\mathbf{x}; \mu, \Sigma).$$

Proof of Lemma 4

First note that

$$(2\pi)^{d/2} \phi(\mathbf{x}; \mu, \Sigma) = |\Sigma|^{-1/2} \exp\left[-\frac{1}{2} \text{tr}\{(\mathbf{x} - \mu)(\mathbf{x} - \mu)^T \Sigma^{-1}\}\right].$$

Then, using the identity $\text{tr}(\mathbf{A}^T \mathbf{B}) = \text{vec}(\mathbf{A})^T \text{vec}(\mathbf{B})$,

$$\begin{aligned} (2\pi)^{d/2} d\Sigma \phi(\mathbf{x}; \mu, \Sigma) &= (d\Sigma |\Sigma|^{-1/2}) \exp\left[-\frac{1}{2} \text{tr}\{(\mathbf{x} - \mu)(\mathbf{x} - \mu)^T \Sigma^{-1}\}\right] \\ &\quad + |\Sigma|^{-1/2} \left(d\Sigma \exp\left[-\frac{1}{2} \text{tr}\{(\mathbf{x} - \mu)(\mathbf{x} - \mu)^T \Sigma^{-1}\}\right]\right) \\ &= -\frac{1}{2} |\Sigma|^{-3/2} \Sigma \left[\text{tr}(\Sigma^{-1} d\Sigma \Sigma) \exp\left[-\frac{1}{2} \text{tr}\{(\mathbf{x} - \mu)(\mathbf{x} - \mu)^T \Sigma^{-1}\}\right]\right] \\ &\quad + |\Sigma|^{-1/2} \exp\left[-\frac{1}{2} \text{tr}\{(\mathbf{x} - \mu)(\mathbf{x} - \mu)^T \Sigma^{-1}\}\right] \\ &\quad \times \left[-\frac{1}{2} \text{tr}\{(\mathbf{x} - \mu)(\mathbf{x} - \mu)^T d\Sigma \Sigma^{-1}\}\right] \\ &= -\frac{1}{2} (2\pi)^{d/2} \phi(\mathbf{x}; \mu, \Sigma) \text{vec}(\Sigma^{-1})^T d\text{vec}(\Sigma) \\ &\quad - \frac{1}{2} (2\pi)^{d/2} \phi(\mathbf{x}; \mu, \Sigma) \text{tr}\{(\mathbf{x} - \mu)(\mathbf{x} - \mu)^T \Sigma^{-1} (d\Sigma) \Sigma^{-1}\} \\ &= -\frac{1}{2} (2\pi)^{d/2} \phi(\mathbf{x}; \mu, \Sigma) \text{vec}(\Sigma^{-1})^T d\text{vec}(\Sigma) \\ &\quad + \frac{1}{2} (2\pi)^{d/2} \phi(\mathbf{x}; \mu, \Sigma) \text{vec}(\Sigma^{-1} (\mathbf{x} - \mu)(\mathbf{x} - \mu)^T \Sigma^{-1})^T d\text{vec}(\Sigma). \end{aligned}$$

Therefore, by Theorem 6, Chapter 5, of Magnus and Neudecker (1999),

$$D_{\text{vec}(\Sigma)} \phi(\mathbf{x}; \mu, \Sigma) = \frac{1}{2} \phi(\mathbf{x}; \mu, \Sigma) \text{vec}[\Sigma^{-1} \{(\mathbf{x} - \mu)(\mathbf{x} - \mu)^T \Sigma^{-1} - \mathbf{I}\}]^T.$$

Also,

$$\begin{aligned} |\Sigma|^{1/2} (2\pi)^{d/2} d\mu \phi(\mathbf{x}; \mu, \Sigma) &= \exp\left[-\frac{1}{2} \text{tr}\{(\mathbf{x} - \mu)(\mathbf{x} - \mu)^T \Sigma^{-1}\}\right] \\ &\quad \times \left[-\frac{1}{2} \text{tr}\{d\mu \{(\mathbf{x} - \mu)(\mathbf{x} - \mu)^T \Sigma^{-1}\}\right] \\ &= |\Sigma|^{1/2} (2\pi)^{d/2} \phi(\mathbf{x}; \mu, \Sigma) \text{tr}\{(\mathbf{x} - \mu)(d\mu)^T \Sigma^{-1}\} \\ &= |\Sigma|^{1/2} (2\pi)^{d/2} \phi(\mathbf{x}; \mu, \Sigma) \Sigma^{-1} (\mathbf{x} - \mu)^T d\mu \end{aligned}$$

which simplifies to

$$d\mu \phi(\mathbf{x}; \mu, \Sigma) = \phi(\mathbf{x}; \mu, \Sigma) \Sigma^{-1} (\mathbf{x} - \mu)^T d\mu.$$

The second differential with respect to μ is then

$$\begin{aligned} d_\mu^2 \phi(\mathbf{x}; \mu, \Sigma) &= \{d_\mu \phi(\mathbf{x}; \mu, \Sigma)\} \Sigma^{-1} (\mathbf{x} - \mu)^T d\mu + \phi(\mathbf{x}; \mu, \Sigma) \Sigma^{-1} (-d\mu)^T d\mu \\ &= \{\phi(\mathbf{x}; \mu, \Sigma) \Sigma^{-1} (\mathbf{x} - \mu)^T d\mu\} \Sigma^{-1} (\mathbf{x} - \mu)^T d\mu + \phi(\mathbf{x}; \mu, \Sigma) \Sigma^{-1} (-d\mu)^T d\mu \\ &= (d\mu)^T \left(\phi(\mathbf{x}; \mu, \Sigma) \Sigma^{-1} \{(\mathbf{x} - \mu)(\mathbf{x} - \mu)^T \Sigma^{-1} - \mathbf{I}\} \right) d\mu. \end{aligned}$$

Hence, using Theorem 6, Chapter 6, of Magnus and Neudecker (1999)

$$\mathbf{H}_\mu \phi(\mathbf{x}; \mu, \Sigma) = \phi(\mathbf{x}; \mu, \Sigma) \Sigma^{-1} \{(\mathbf{x} - \mu)(\mathbf{x} - \mu)^T \Sigma^{-1} - \mathbf{I}\} = 2 \text{vec}^{-1}\left\{D_{\text{vec}(\Sigma)} \phi(\mathbf{x}; \mu, \Sigma)^T\right\}.$$

A.6 Derivation of Result 2

To make the derivation less cumbersome we will suppress the subscripts on the mean μ and covariance matrix Σ . As in Wand (2014) we work with the natural statistic and natural parameter pair

$$\mathcal{T}(\phi) = \begin{bmatrix} \phi \\ \text{vech}(\phi \phi^T) \end{bmatrix} \quad \text{and} \quad \eta = \begin{bmatrix} \Sigma^{-1} \mu \\ -\frac{1}{2} D_d \text{vec}(\Sigma^{-1}) \end{bmatrix}.$$

The mean parameter vector is

$$\tau = E\{\mathcal{T}(\phi)\} = \begin{bmatrix} \mu \\ \text{vech}(\Sigma + \mu \mu^T) \end{bmatrix}.$$

In Lemma 3 in Appendix A.5 we show that

$$D_\tau \begin{bmatrix} \mu \\ \text{vec}(\Sigma) \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -(\mathbf{I} + \mathbf{K}_d)(\mu \otimes \mathbf{I}) & D_d \end{bmatrix}$$

and so Result 1(d) becomes

$$\begin{aligned} &\begin{bmatrix} \Sigma^{-1} \mu \\ -\frac{1}{2} D_d^T \text{vec}(\Sigma^{-1}) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I} & -(\mu^T \otimes \mathbf{I})(\mathbf{I} + \mathbf{K}_d) \\ \mathbf{0} & D_d^T \end{bmatrix} \begin{bmatrix} D_\mu \text{NonEntropy}\{q(\phi; \mu, \Sigma)\}^T \\ D_{\text{vec}(\Sigma)} \text{NonEntropy}\{q(\phi; \mu, \Sigma)\}^T \end{bmatrix} \end{aligned} \quad (55)$$

where we have used the fact that $\mathbf{K}_d^T = \mathbf{K}_d$. Using (52), and the fact that $\text{vec}^{-1}[D_{\text{vec}(\Sigma)} \text{NonEntropy}\{q(\phi; \mu, \Sigma)\}^T]$

is symmetric, the first component of (55) is equivalent to

$$\begin{aligned}
& \Sigma^{-1} \mu = D_\mu \text{NonEntropy}\{q(\phi; \mu, \Sigma)\}^T \\
& \quad - (\mu^T \otimes \mathbf{I})(\mathbf{I} + \mathbf{K}_a) D_{\text{vec}(\Sigma)} \text{NonEntropy}\{q(\phi; \mu, \Sigma)\}^T \\
& = D_\mu \text{NonEntropy}\{q(\phi; \mu, \Sigma)\}^T \\
& \quad - 2(\mu^T \otimes \mathbf{I})(\mathbf{I} + \mathbf{K}_a) \text{vec}\left(\text{vec}^{-1}[D_{\text{vec}(\Sigma)} \text{NonEntropy}\{q(\phi; \mu, \Sigma)\}^T]\right) \\
& = D_\mu \text{NonEntropy}\{q(\phi; \mu, \Sigma)\}^T \\
& \quad - 2(\mu^T \otimes \mathbf{I}) \text{vec}\left(\text{vec}^{-1}[D_{\text{vec}(\Sigma)} \text{NonEntropy}\{q(\phi; \mu, \Sigma)\}^T]\right) \\
& = D_\mu \text{NonEntropy}\{q(\phi; \mu, \Sigma)\}^T \\
& \quad - 2 \text{vec}\left(\text{vec}^{-1}[D_{\text{vec}(\Sigma)} \text{NonEntropy}\{q(\phi; \mu, \Sigma)\}^T]\right) \mu \\
& = D_\mu \text{NonEntropy}\{q(\phi; \mu, \Sigma)\}^T \\
& \quad - 2 \text{vec}^{-1}[D_{\text{vec}(\Sigma)} \text{NonEntropy}\{q(\phi; \mu, \Sigma)\}^T] \mu.
\end{aligned} \tag{56}$$

The second component (55) is equivalent to

$$-\frac{1}{2} D_d^T \text{vec}(\Sigma) = D_d^T [D_{\text{vec}(\Sigma)} \text{NonEntropy}\{q(\phi; \mu, \Sigma)\}^T]$$

which, under the constraint that Σ is symmetric, is equivalent to

$$\Sigma^{-1} = -2 \text{vec}^{-1}[D_{\text{vec}(\Sigma)} \text{NonEntropy}\{q(\phi; \mu, \Sigma)\}^T]. \tag{57}$$

In view of relationships (56) and (57), the natural fixed-point iteration scheme becomes

$$\begin{cases} \Sigma_{\text{new}}^{-1} \mu_{\text{new}} = [D_\mu \text{NonEntropy}\{q(\phi; \mu, \Sigma)\}]^T_{\mu=\mu_{\text{old}}, \Sigma=\Sigma_{\text{old}}} \\ \quad - 2 \text{vec}^{-1}\left([D_{\text{vec}(\Sigma)} \text{NonEntropy}\{q(\phi; \mu, \Sigma)\}]^T_{\mu=\mu_{\text{old}}, \Sigma=\Sigma_{\text{old}}}\right) \mu_{\text{old}} \\ \Sigma_{\text{new}} = \left\{ -2 \text{vec}^{-1}\left([D_{\text{vec}(\Sigma)} \text{NonEntropy}\{q(\phi; \mu, \Sigma)\}]^T_{\mu=\mu_{\text{old}}, \Sigma=\Sigma_{\text{old}}}\right) \right\}^{-1} \end{cases}$$

where $(\mu_{\text{old}}, \Sigma_{\text{old}})$ and $(\mu_{\text{new}}, \Sigma_{\text{new}})$, respectively, denote the old and new values of (μ, Σ) . The following simplification ensues:

$$\begin{cases} \mu_{\text{new}} = \mu_{\text{old}} + \Sigma_{\text{new}} [D_\mu \text{NonEntropy}\{q(\phi; \mu, \Sigma)\}]^T_{\mu=\mu_{\text{old}}, \Sigma=\Sigma_{\text{old}}} \\ \Sigma_{\text{new}} = \left\{ -2 \text{vec}^{-1}\left([D_{\text{vec}(\Sigma)} \text{NonEntropy}\{q(\phi; \mu, \Sigma)\}]^T_{\mu=\mu_{\text{old}}, \Sigma=\Sigma_{\text{old}}}\right) \right\}^{-1} \end{cases}$$

which is equivalent to the following updating scheme:

$$\begin{cases} \mathbf{v} \leftarrow D_\mu \text{NonEntropy}\{q(\phi; \mu, \Sigma)\}^T \\ \Sigma \leftarrow \left(-2 \text{vec}^{-1}[D_{\text{vec}(\Sigma)} \text{NonEntropy}\{q(\phi; \mu, \Sigma)\}^T] \right)^{-1} \\ \mu \leftarrow \mu + \Sigma \mathbf{v} \end{cases} \tag{58}$$

as given in Wand (2014). However, from Lemma 4 in Appendix A.5 (see also Appendix A of Oppé and Archambeau, 2009) and the fact that $\text{NonEntropy}\{q(\phi; \mu, \Sigma)\}$ is an expectation with respect to the $N(\mu, \Sigma)$ density function we have

$$\text{vec}^{-1}[D_{\text{vec}(\Sigma)} \text{NonEntropy}\{q(\phi; \mu, \Sigma)\}^T] = \frac{1}{2} H_\mu \text{NonEntropy}\{q(\phi; \mu, \Sigma)\}$$

which leads to a somewhat more elegant alternative to (58) given in Result 2.

A.7 Derivation of the Derivative Matrix of g_{Ex2}

From Section 4.3, the fixed-point iteration updating function is of the form

$$g_{\text{Ex2}} \left(\begin{bmatrix} \mu \\ \text{vech}(\Sigma) \end{bmatrix} \right) \equiv \begin{bmatrix} g_{\text{Ex2}, \mu} \\ \text{vech}(\mathbf{G}_{\text{Ex2}, \Sigma}) \end{bmatrix}$$

where

$$\mathbf{G}_{\text{Ex2}, \Sigma} \equiv [\mathbf{C}^T \text{diag}\{\omega(\mu, \Sigma)\} \mathbf{C} + \mathbf{M}]^{-1},$$

$$g_{\text{Ex2}, \mu} \equiv \mu + \mathbf{G}_{\text{Ex2}, \Sigma} [\mathbf{C}^T \{y - \omega(\mu, \Sigma)\} - \mathbf{M} \mu]$$

$$\text{and } \omega(\mu, \Sigma) \equiv \exp\{\mathbf{C} \mu + \frac{1}{2} \text{diagonal}(\mathbf{C} \Sigma \mathbf{C}^T)\}.$$

Note that the dependence of $\mathbf{G}_{\text{Ex2}, \Sigma}$ and g_{Ex2} on y , \mathbf{C} and \mathbf{M} is suppressed here.

The derivative matrix of g_{Ex2} with respect to $[\mu, \text{vech}(\Sigma)]^T$ is

$$D \begin{bmatrix} \mu \\ \text{vech}(\Sigma) \end{bmatrix} g_{\text{Ex2}} = \begin{bmatrix} D_\mu g_{\text{Ex2}, \mu} & D_{\text{vech}(\Sigma)} g_{\text{Ex2}, \mu} \\ D_\mu \text{vech}(\mathbf{G}_{\text{Ex2}, \Sigma}) & D_{\text{vech}(\Sigma)} \text{vech}(\mathbf{G}_{\text{Ex2}, \Sigma}) \end{bmatrix}.$$

We now give explicit expressions for each of these four components of the derivative matrix. It is more efficient, notationally, to first obtain expressions for the derivatives of $\text{vech}(\mathbf{G}_{\text{Ex2}, \Sigma})$.

A.7.1 EXPRESSION FOR $D_\mu \text{vech}(\mathbf{G}_{\text{Ex2}, \Sigma})$

$$D_\mu \text{vech}(\mathbf{G}_{\text{Ex2}, \Sigma}) = -D_{p+K}^+ (\mathbf{G}_{\text{Ex2}, \Sigma} \otimes \mathbf{G}_{\text{Ex2}, \Sigma}) \mathbf{Q}(\mathbf{C})^T \text{diag}\{\omega(\mu, \Sigma)\} \mathbf{C} \tag{59}$$

Derivation:

Using the second rule in Section 3.3.5 of Wand (2002), (52) and (51),

$$\begin{aligned}
D_\mu \text{vech}(\mathbf{G}_{\text{Ex2}, \Sigma}) &= D_{p+K}^+ d_\mu \text{vec}\left([\mathbf{C}^T \text{diag}\{\omega(\mu, \Sigma)\} \mathbf{C} + \mathbf{M}\right]^{-1}) \\
&= -D_{p+K}^+ \text{vec}\left\{\mathbf{G}_{\text{Ex2}, \Sigma} \left(d_\mu [\mathbf{C}^T \text{diag}\{\omega(\mu, \Sigma)\} \mathbf{C} + \mathbf{M}]\right) \mathbf{G}_{\text{Ex2}, \Sigma}\right\} \\
&= -D_{p+K}^+ \left(\mathbf{G}_{\text{Ex2}, \Sigma} \otimes \mathbf{G}_{\text{Ex2}, \Sigma}\right) \text{vec}\left[\mathbf{C}^T \text{diag}\{d_\mu \omega(\mu, \Sigma)\} \mathbf{C}\right].
\end{aligned}$$

From Theorem 2(b) of Wand (2014),

$$\text{vec}[\mathbf{C}^T \text{diag}\{d_\mu \omega(\mu, \Sigma)\} \mathbf{C}] = \mathbf{Q}(\mathbf{C})^T d_\mu \omega(\mu, \Sigma).$$

Lastly, we use the chain rule in Section 3.3.2 of Wand (2002) to get

$$d_{\mu}\omega(\mu, \Sigma) = d_{\mu} \exp\{C\mu + \frac{1}{2} \text{diagonal}(C\Sigma C^T)\} = \text{diag}\{\omega(\mu, \Sigma)\} C d_{\mu}.$$

Combining, we then obtain

$$d_{\mu} \text{vec}(\mathbf{G}_{\text{Ex}2, \Sigma}) = -D_{p+K}^+ \left(\mathbf{G}_{\text{Ex}2, \Sigma} \otimes \mathbf{G}_{\text{Ex}2, \Sigma} \right) \mathbf{Q}(C)^T \text{diag}\{\omega(\mu, \Sigma)\} C d_{\mu}$$

and the stated expression then follow from Theorem 6, Chapter 5, of Magnus and Neudecker (1999).

A.7.2 EXPRESSION FOR $D_{\text{vech}(\Sigma)} \text{vech}(\mathbf{G}_{\text{Ex}2, \Sigma})$

$$\begin{aligned} D_{\text{vech}(\Sigma)} \text{vech}(\mathbf{G}_{\text{Ex}2, \Sigma}) &= -\frac{1}{2} D_{p+K}^+ (\mathbf{G}_{\text{Ex}2, \Sigma} \otimes \mathbf{G}_{\text{Ex}2, \Sigma}) \\ &\quad \times \mathbf{Q}(C)^T \text{diag}\{\omega(\mu, \Sigma)\} \mathbf{Q}(C) D_{p+K}. \end{aligned} \quad (60)$$

Derivation:

The derivation is similar to that for $D_{\mu} \mathbf{G}_{\text{Ex}2, \Sigma}$. It differs in that it requires $d_{\Sigma} \omega(\mu, \Sigma)$ rather than $d_{\mu} \omega(\mu, \Sigma)$. This entails

$$d_{\Sigma} \omega(\mu, \Sigma) = d_{\Sigma} \exp\{C\mu + \frac{1}{2} \text{diagonal}(C\Sigma C^T)\} = \frac{1}{2} \text{diag}\{\omega(\mu, \Sigma)\} d_{\Sigma} \text{diagonal}(C\Sigma C^T).$$

But Theorem 2(a) of Wand (2014) gives

$$d_{\Sigma} \text{diagonal}(C\Sigma C^T) = \mathbf{Q}(C) d_{\Sigma} \text{vec}(\Sigma) = \mathbf{Q}(C) D_{p+K} d_{\Sigma} \text{vech}(\Sigma)$$

which leads to the stated result.

A.7.3 EXPRESSION FOR $D_{\mu} \mathbf{g}_{\text{Ex}2, \mu}$

$$D_{\mu} \mathbf{g}_{\text{Ex}2, \mu} = (C^T \{y - \omega(\mu, \Sigma)\} - M\mu)^T \otimes I) D_{p+K} D_{\mu} \text{vech}(\mathbf{G}_{\text{Ex}2, \Sigma})$$

where $D_{\mu} \text{vech}(\mathbf{G}_{\text{Ex}2, \Sigma})$ is given by (59).

Derivation:

Using the second rule in Section 3.3.4 of Wand (2002) for differentiation of matrix products,

$$\begin{aligned} d_{\mu} \mathbf{g}_{\text{Ex}2, \mu} &= d_{\mu} + d_{\mu} \left(\mathbf{G}_{\text{Ex}2, \Sigma} [C^T \{y - \omega(\mu, \Sigma)\} - M\mu] \right) \\ &= d_{\mu} + (d_{\mu} \mathbf{G}_{\text{Ex}2, \Sigma}) [C^T \{y - \omega(\mu, \Sigma)\} - M\mu] \\ &\quad - \mathbf{G}_{\text{Ex}2, \Sigma} [C^T d_{\mu} \omega(\mu, \Sigma) + M d_{\mu}] \\ &= d_{\mu} + \text{vec}\{I (d_{\mu} \mathbf{G}_{\text{Ex}2, \Sigma}) [C^T \{y - \omega(\mu, \Sigma)\} - M\mu]\} \\ &\quad - \mathbf{G}_{\text{Ex}2, \Sigma} [C^T \text{diag}\{\omega(\mu, \Sigma)\} C + M] d_{\mu}. \end{aligned}$$

where we have used the fact that $(d_{\mu} \mathbf{G}_{\text{Ex}2, \Sigma}) [C^T \{y - \omega(\mu, \Sigma)\} - M\mu]$ is a column vector. Application of (52) leads to

$$\begin{aligned} d_{\mu} \mathbf{g}_{\text{Ex}2, \mu} &= d_{\mu} + ([C^T \{y - \omega(\mu, \Sigma)\} - M\mu]^T \otimes I) d_{\mu} \text{vec}(\mathbf{G}_{\text{Ex}2, \Sigma}) \\ &\quad - \mathbf{G}_{\text{Ex}2, \Sigma} [C^T \text{diag}\{\omega(\mu, \Sigma)\} C + M] d_{\mu} \\ &= \left\{ I + ([C^T \{y - \omega(\mu, \Sigma)\} - M\mu]^T \otimes I) D_{\mu} \text{vec}(\mathbf{G}_{\text{Ex}2, \Sigma}) \right. \\ &\quad \left. - \mathbf{G}_{\text{Ex}2, \Sigma} \mathbf{G}_{\text{Ex}2, \Sigma}^{-1} \right\} d_{\mu} \\ &= ([C^T \{y - \omega(\mu, \Sigma)\} - M\mu]^T \otimes I) D_{p+K} D_{\mu} \text{vech}(\mathbf{G}_{\text{Ex}2, \Sigma}) d_{\mu}. \end{aligned}$$

The given expression follows from Theorem 6, Chapter 5, of Magnus and Neudecker (1999).

A.7.4 EXPRESSION FOR $D_{\text{vech}(\Sigma)} \mathbf{g}_{\text{Ex}2, \mu}$

$$\begin{aligned} D_{\text{vech}(\Sigma)} \mathbf{g}_{\text{Ex}2, \mu} &= \left([C^T \{y - \omega(\mu, \Sigma)\} - M\mu]^T \otimes I \right) D_{p+K} D_{\text{vech}(\Sigma)} \text{vech}(\mathbf{G}_{\text{Ex}2, \Sigma}) \\ &\quad - \frac{1}{2} \mathbf{G}_{\text{Ex}2, \Sigma} C^T \text{diag}\{\omega(\mu, \Sigma)\} \mathbf{Q}(C) D_{p+K} \end{aligned}$$

where $D_{\text{vech}(\Sigma)} \text{vech}(\mathbf{G}_{\text{Ex}2, \Sigma})$ is given by (60).

Derivation:

Dealing with matrix products via the second rule in Section 3.3.4 of Wand (2002) we obtain

$$\begin{aligned} d_{\Sigma} \mathbf{g}_{\text{Ex}2, \mu} &= d_{\Sigma} \left(\mathbf{G}_{\text{Ex}2, \Sigma} [C^T \{y - \omega(\mu, \Sigma)\} - M\mu] \right) \\ &= (d_{\Sigma} \mathbf{G}_{\text{Ex}2, \Sigma}) [C^T \{y - \omega(\mu, \Sigma)\} - M\mu] \\ &\quad - \mathbf{G}_{\text{Ex}2, \Sigma} C^T d_{\Sigma} \omega(\mu, \Sigma) \\ &= \text{vec} \left(I (d_{\Sigma} \mathbf{G}_{\text{Ex}2, \Sigma}) [C^T \{y - \omega(\mu, \Sigma)\} - M\mu] \right) \\ &\quad - \frac{1}{2} \mathbf{G}_{\text{Ex}2, \Sigma} C^T \text{diag}\{\omega(\mu, \Sigma)\} \mathbf{Q}(C) d_{\text{vec}}(\Sigma) \\ &= ([C^T \{y - \omega(\mu, \Sigma)\} - M\mu]^T \otimes I) D_{p+K} d_{\text{vec}}(\mathbf{G}_{\text{Ex}2, \Sigma}) \\ &\quad - \frac{1}{2} \mathbf{G}_{\text{Ex}2, \Sigma} C^T \text{diag}\{\omega(\mu, \Sigma)\} \mathbf{Q}(C) D_{p+K} d_{\text{vec}}(\Sigma) \\ &= ([C^T \{y - \omega(\mu, \Sigma)\} - M\mu]^T \otimes I) \\ &\quad \times D_{p+K} D_{\text{vech}(\Sigma)} \text{vech}(\mathbf{G}_{\text{Ex}2, \Sigma}) d_{\text{vech}}(\Sigma) \\ &\quad - \frac{1}{2} \mathbf{G}_{\text{Ex}2, \Sigma} C^T \text{diag}\{\omega(\mu, \Sigma)\} \mathbf{Q}(C) D_{p+K} d_{\text{vec}}(\Sigma). \end{aligned}$$

Once again we call upon Theorem 6, Chapter 5, of Magnus and Neudecker (1999) to complete the derivation.

References

- A.S. Ackleh, E.J. Allen, R.B. Hearfott, and P. Seshaiyer. *Classical and Modern Numerical Analysis: Theory, Methods and Practice*. Chapman & Hall, CRC Press, Boca Raton, Florida, 2010.
- S. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10:251–276, 1998.
- C. Archambeau, D. Cornford, M. Opper, and J. Shawe-Taylor. Gaussian process approximations of stochastic differential equations. *Journal of Machine Learning Research: Workshop and Conference Proceedings*, 1:1–16, 2007.
- A. Azzalini and A. Dalla Valle. The Multivariate Skew-Normal distribution. *Biometrika*, 83:715–726, 1996.
- D. Barber and C.M. Bishop. Ensemble learning for multi-layer networks. In M.I. Jordan, K.J. Kearns, and S.A. Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 395–401, 1997.
- M.J. Beal and Z. Ghahramani. Variational Bayesian learning of directed graphical models with hidden variables. *Bayesian Analysis*, 4:793–832, 2006.
- C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.
- L. Bottou. Stochastic learning. In O. Bousquet, U. von Luxburg, and G. Rätsch, editors, *Advanced Lectures on Machine Learning*, pages 146–168, 2004.
- S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.
- E. Challis and D. Barber. Gaussian Kullback-Leibler approximate inference. *Journal of Machine Learning Research*, 14:2239–2286, 2013.
- Y.H. Dai and Y. Yuan. A nonlinear conjugate gradient method with a strong global convergence property. *SIAM Journal on Optimization*, 10:177–182, 1999.
- S.J. Gershman, M.D. Hoffman, and D.M. Blei. Nonparametric variational inference. In *Proceedings of the Twenty Ninth International Conference on Machine Learning*, pages 633–670, 2012.
- G.H. Givens and J.A. Hoeting. *Computational Statistics*. Wiley, Hoboken, New Jersey, 2005.
- M.R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49:409–436, 1952.
- G.E. Hinton and D. van Camp. Keeping neural networks simple by minimizing description length of the weights. In L. Pitt, editor, *Proceedings of the Sixth Annual Conference on Computational Learning Theory*, pages 5–13, 1993.
- M.D. Hoffman, D.M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013.
- A. Honkela and H. Valpola. Unsupervised variational Bayesian learning of nonlinear models. *Advances in Neural Information Processing Systems*, 17:593–600, 2005.
- A. Honkela, H. Valpola, A. Ilin, and J. Karhunen. Blind separation of nonlinear mixtures by variational Bayesian learning. *Digital Signal Processing*, 17:914–934, 2007.
- A. Honkela, M. Tornio, T. Raiko, and J. Karhunen. Natural conjugate gradient in variational inference. In M. Ishikawa, K. Doya, H. Miyamoto, and T. Yamakawa, editors, *Proceedings of the Fourteenth International Conference on Neural Information Processing*, pages 305–314, 2008.
- A. Honkela, T. Raiko, M. Kuusela, M. Tornio, and J. Karhunen. Approximate Riemannian conjugate gradient learning for fixed-form variational Bayes. *Journal of Machine Learning Research*, 11:3235–3268, 2010.
- D.A. Knowles and T.P. Minka. Non-conjugate message passing for multinomial and binary regression. In J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, 2011.
- H. Lappalainen and A. Honkela. Bayesian nonlinear independent component analysis by multilayer perceptrons. In M. Girolami, editor, *Advances in Independent Component Analysis*, pages 93–121, 2000.
- D.G. Luenberger and Y. Ye. *Linear and Nonlinear Programming, Third Edition*. Springer, New York, 2008.
- J. Luts and M.P. Wand. Variational inference for count response semiparametric regression. *Bayesian Analysis*, 10:991–1023, 2015.
- J.R. Magnus and H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics, Revised Edition*. Wiley, Chichester, UK, 1999.
- M. Menictas and M.P. Wand. Variational inference for heteroscedastic semiparametric regression. *Australian and New Zealand Journal of Statistics*, 57:119–138, 2015.
- M.K. Murray and J.W. Rice. *Differential Geometry and Statistics*. Chapman & Hall, London, 1993.
- J.A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.
- S.E. Neville, J.T. Ormerod, and M.P. Wand. Mean field variational Bayes for continuous sparse signal shrinkage: pitfalls and remedies. *Electronic Journal of Statistics*, 8:1113–1151, 2014.
- H. Nickisch and C.E. Rasmussen. Approximations for binary Gaussian process classification. *Journal of Machine Learning Research*, 9:2035–2078, 2008.

- M. Opper and C. Archambeau. The variational Gaussian approximation revisited. *Neural Computation*, 21:786–792, 2009.
- J.T. Ormerod and M.P. Wand. Explaining variational approximations. *The American Statistician*, 64:140–153, 2010.
- J.M. Ortega. *Numerical Analysis: A Second Course*. SIAM, Philadelphia, 1990.
- T. Pham, J.T. Ormerod, and M.P. Wand. Mean field variational Bayesian inference for nonparametric regression with measurement error. *Computational Statistics and Data Analysis*, 68:375–387, 2013.
- W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes: The Art of Scientific Computing, Third Edition*. Cambridge University Press, New York, 2007.
- T. Raiko, H. Valpola, M. Harva, and J. Karhunen. Building blocks for variational Bayesian learning of latent variable models. *Journal of Machine Learning Research*, 8:155–201, 2007.
- R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016. URL <http://www.R-project.org/>.
- T. Salimans and D.A. Knowles. Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Analysis*, 8:837–882, 2013.
- M. Sato. Online model selection based on the variational Bayes. *Neural Computation*, 13:1649–1681, 2001.
- L.S.L. Tan and D.J. Nott. Variational inference for generalized linear mixed models using partially noncentered parametrizations. *Statistical Science*, 28:168–188, 2013.
- M.J. Wainwright and M.I. Jordan. Graphical models, exponential families and variational inference. *Foundations and Trends in Machine Learning*, 1:1–305, 2008.
- M.P. Wand. Vector differential calculus in statistics. *The American Statistician*, 56:55–62, 2002.
- M.P. Wand. Fully simplified Multivariate Normal updates in non-conjugate variational message passing. *Journal of Machine Learning Research*, 15:1351–1369, 2014.
- M.P. Wand, J.T. Ormerod, S.A. Padoan, and R. Prithwirth. Mean field variational Bayes for elaborate distributions. *Bayesian Analysis*, 6:847–900, 2011.
- B. Wang and D.M. Titterton. Inadequacy of interval estimates corresponding to variational Bayes approximations. In Z. Ghahramani and R. Cowell, editors, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pages 373–380, 2005.
- O. Zobyay. Variational Bayesian inference with Gaussian-mixture approximations. *Electronic Journal of Statistics*, 8:355–389, 2014.

Online PCA with Optimal Regret*

Jiazhong Nie

Department of Computer Science, University of California, Santa Cruz

NIJIAZHONG@CSE.UCSB.EDU

Wojciech Kotłowski

Institute of Computing Science, Poznań University of Technology, Poland

WKOTLOWSKI@CS.PUT.POZNAN.PL

Manfred K. Warmuth

Department of Computer Science, University of California, Santa Cruz

MANFRED@CSE.UCSB.EDU

Editor: Nathan Srebro

Abstract

We investigate the online version of Principle Component Analysis (PCA), where in each trial t the learning algorithm chooses a k -dimensional subspace, and upon receiving the next instance vector \mathbf{x}_t , suffers the “compression loss”, which is the squared Euclidean distance between this instance and its projection into the chosen subspace. When viewed in the right parameterization, this compression loss is linear, i.e. it can be rewritten as $\text{tr}(\mathbf{W}_t \mathbf{x}_t \mathbf{x}_t^\top)$, where \mathbf{W}_t is the parameter of the algorithm and the outer product $\mathbf{x}_t \mathbf{x}_t^\top$ (with $\|\mathbf{x}_t\| \leq 1$) is the instance matrix. In this paper we generalize PCA to arbitrary positive definite instance matrices \mathbf{X}_t with the linear loss $\text{tr}(\mathbf{W}_t \mathbf{X}_t)$.

We evaluate online algorithms in terms of their worst-case regret, which is a bound on the additional total loss of the online algorithm on all instances matrices over the compression loss of the best k -dimensional subspace (chosen in hindsight). We focus on two popular online algorithms for generalized PCA: the Gradient Descent (GD) and Matrix Exponentiated Gradient (MEG) algorithms. We show that if the regret is expressed as a function of the number of trials, then both algorithms are optimal to within a constant factor on worst-case sequences of positive definite instance matrices with trace norm at most one (which subsumes the original PCA problem with outer products). This is surprising because MEG is believed to be suboptimal in this case. We also show that when considering regret bounds as a function of a loss budget, then MEG remains optimal and strictly outperforms GD when the instance matrices are trace norm bounded.

Next, we consider online PCA when the adversary is allowed to present the algorithm with positive semidefinite instance matrices whose largest eigenvalue is bounded (rather than their trace which is the sum of their eigenvalues). Again we can show that MEG is optimal and strictly better than GD in this setting.

Keywords: online learning, regret bounds, expert setting, k -sets, PCA, Gradient Descent, Matrix Exponentiated Gradient algorithm

1. Introduction

In Principle Component Analysis (PCA), the data points $\mathbf{x}_t \in \mathbb{R}^n$ are projected / compressed onto a k -dimensional subspace. Such a subspace can be represented by its projection matrix \mathbf{P} which is a symmetric matrix in $\mathbb{R}^{n \times n}$ with k eigenvalues equal 1 and $n - k$

eigenvalues equal 0. The goal of *uncentered PCA* is to find the rank k projection matrix that minimizes the total *compression loss* $\sum_t \|\mathbf{P} \mathbf{x}_t - \mathbf{x}_t\|^2$, i.e. the sum of the squared Euclidean distances between the original and the projected data points. In *centered PCA* the goal is to minimize $\sum_t \|\mathbf{P}(\mathbf{x}_t - \boldsymbol{\mu}) - (\mathbf{x}_t - \boldsymbol{\mu})\|^2$ where \mathbf{P} is a projection matrix of rank k and $\boldsymbol{\mu} \in \mathbb{R}^n$ is a second mean parameter. For the sake of simplicity we focus on the optimal algorithms for uncentered PCA. However we believe that our results will essentially carry over to the centered case as was already partially done in Warmuth and Kuzmin (2008). Surprisingly, this loss can be written as a linear loss (Warmuth and Kuzmin, 2008):

$$\sum_t \|\mathbf{P} \mathbf{x}_t - \mathbf{x}_t\|^2 = \sum_t \|(\mathbf{P} - \mathbf{I}) \mathbf{x}_t\|^2 = \sum_t \mathbf{x}_t^\top (\mathbf{I} - \mathbf{P})^\top \mathbf{x}_t = \underbrace{\text{tr}((\mathbf{I} - \mathbf{P}) \sum_t \mathbf{x}_t \mathbf{x}_t^\top)}_{\mathbf{C}},$$

where in the 3rd equality we used the fact that $\mathbf{I} - \mathbf{P}$ is a projection matrix and therefore $(\mathbf{I} - \mathbf{P})^2 = \mathbf{I} - \mathbf{P}$. The final expression of the compression loss is linear in the projection matrix $\mathbf{P} - \mathbf{I}$ as well as the covariance matrix $\mathbf{C} = \sum_t \mathbf{x}_t \mathbf{x}_t^\top$. The projection matrix $\mathbf{P} - \mathbf{I}$ is a sum of $n - k$ outer products: $\mathbf{P} - \mathbf{I} = \sum_{i=1}^{n-k} \mathbf{u}_i \mathbf{u}_i^\top$, where the \mathbf{u}_i are unit length and orthogonal. The crucial point to note here is that the compression loss is *linear* in the projection matrix $\mathbf{P} - \mathbf{I}$ but not in the direction vectors \mathbf{u}_i .

The batch version of uncentered PCA is equivalent to finding the eigenvectors $\mathbf{u}_1, \dots, \mathbf{u}_k$ belonging to the k largest eigenvalues of the covariance matrix \mathbf{C} : if $\mathbf{P} = \sum_{i=1}^k \mathbf{u}_i \mathbf{u}_i^\top$ is the k dimensional projection matrix formed from these k eigenvectors, then $\mathbf{I} - \mathbf{P}$ is the complementary $n - k$ dimensional projection matrix minimizing the linear loss $\text{tr}((\mathbf{I} - \mathbf{P})\mathbf{C})$.

In this paper we consider the online version of uncentered PCA (Warmuth and Kuzmin, 2008), where in each trial $t = 1, \dots, T$, the algorithm chooses (based on the previously observed points $\mathbf{x}_1, \dots, \mathbf{x}_{t-1}$) a subspace of dimension k described by its projection matrix \mathbf{P}_t of rank k . Then a next point \mathbf{x}_t (or instance matrix $\mathbf{x}_t \mathbf{x}_t^\top$) is revealed and the algorithm suffers the *compression loss*:

$$\|\mathbf{x}_t - \mathbf{P}_t \mathbf{x}_t\|^2 = \text{tr}((\mathbf{I} - \mathbf{P}_t) \mathbf{x}_t \mathbf{x}_t^\top). \quad (1.1)$$

The goal here is to obtain an online algorithm whose cumulative loss over trials $t = 1, \dots, T$ is close to the cumulative loss of the best rank k projection matrix chosen in hindsight after seeing all T instances. The maximum difference between the cumulative loss of the algorithm and the best off-line comparator is called the (worst-case) *regret*. This regret naturally scales with the maximum square L_2 -norm of the data points \mathbf{x}_t . For the sake of simplicity we assume that all points have L_2 -norm bounded by one, i.e. $\|\mathbf{x}_t\| \leq 1$ for all t . In the paper we find the optimal algorithm for online PCA (and some generalizations), where optimal here means that the upper bounds we prove for the regret of the algorithm is at most a constant factor larger than the lower bound we can prove for the learning problem.

There are two main families of algorithms in online learning, which differ in how the parameter vector/matrix is updated: the Gradient Descent (GD) family (Cesa-Bianchi et al., 1996; Kivinen and Warmuth, 1997; Zinkevich, 2003) and the Exponentiated Gradient (EG) family (Kivinen and Warmuth, 1997). The updated parameters of both families of algorithms are solutions to certain minimization problems which trade off a divergence to

*. A preliminary version of this paper appeared in the 24th International Conference on Algorithmic Learning Theory (2013) (Nie et al., 2013).

the last parameter against the loss on the current instance. The GD family uses the squared Euclidean distance divergence in the trade-off, whereas the Exponentiated Gradient (EG) family is motivated by the relative entropy divergence (Kivinen and Warmuth, 1997). The first family leads to *additive updates* of the parameter vector/matrix. When there are no constraints on the parameter space, then the parameter vector/matrix of the GD family is a linear combination of the instances. However when there are constraints, then after the update the parameter is projected onto the constraints (by a Bregman projection with respect to the squared Euclidean distance). The second family leads to *multiplicative update* algorithms. For that family, the components of the parameter are non-negative and if the parameter space consists of probability vectors, then the non-negativity is already enforced by the relative entropy divergence and less projections are needed.

What is the best parameter space for uncentered PCA? The compression loss (1.1) is linear in the projection matrix $\mathbf{I} - \mathbf{P}_t$ which is of rank $n - k$. An online algorithm has uncertainty over the best projection matrix. Therefore the parameter matrix \mathbf{W}_t of the algorithm is a mixture of such matrices (Warmuth and Kuzmin, 2008) which must be a positive semi-definite matrix of trace $n - k$ whose eigenvalues are capped at 1. The algorithm chooses its projection matrix $\mathbf{I} - \mathbf{P}_t$ by sampling from this mixture \mathbf{W}_t , i.e. $\mathbb{E}[\mathbf{I} - \mathbf{P}_t] = \mathbf{W}_t$. The loss of the algorithm is $\text{tr}((\mathbf{I} - \mathbf{P}_t) \mathbf{x}_t \mathbf{x}_t^\top)$ and its expected loss $\text{tr}(\mathbf{W}_t \mathbf{x}_t \mathbf{x}_t^\top)$.

In Warmuth and Kuzmin (2008), a matrix version of the multiplicative update was applied to PCA, whose regret bound is logarithmic in the dimension n . This algorithm uses the quantum relative entropy in its motivation and is called the *Matrix Exponentiated Gradient* (MEG) algorithm (Tsuda et al., 2005). It does a matrix version of a multiplicative update and then projects onto the “trace equal $n - k$ ” and the “capping” constraints (Here the projections are with respect to the quantum relative entropy).

For the PCA problem, the (expected) loss of the algorithm at trial t is $\text{tr}(\mathbf{W}_t \mathbf{x}_t \mathbf{x}_t^\top)$. Consider the generalization to the loss $\text{tr}(\mathbf{W}_t \mathbf{X}_t)$ where now \mathbf{X}_t is any positive semi-definite symmetric instance matrix and the parameter \mathbf{W}_t is still a convex combination of rank $n - k$ dimensional projection matrices, i.e. $\mathbf{W}_t = \mathbb{E}[\mathbf{I} - \mathbf{P}_t]$ where \mathbf{P}_t is the rank k projection matrix chosen by the algorithm at trial t . The linear loss $\text{tr}(\mathbb{E}[\mathbf{I} - \mathbf{P}_t] \mathbf{X}_t)$ still has a meaning in terms of a compression loss: For any decomposition of \mathbf{X}_t into a linear combination of outer products, i.e. $\mathbf{X}_t = \sum_{q=1}^n \lambda_q \mathbf{z}_q \mathbf{z}_q^\top$ (where the λ_q may be positive or negative and the $\mathbf{z}_q \in \mathbf{R}^n$ don’t have to be orthogonal) we have

$$\text{tr}((\mathbf{I} - \mathbf{P}_t) \mathbf{X}_t) = \sum_q \lambda_q \text{tr}((\mathbf{I} - \mathbf{P}_t) \mathbf{z}_q \mathbf{z}_q^\top) = \sum_q \lambda_q \|\mathbf{z}_q - \mathbf{P}_t \mathbf{z}_q\|^2. \quad (1.2)$$

In this paper we analyze our algorithm for two classes of positive definite instance matrices. Recall that in the vanilla PCA problem the instance matrices are the outer products, i.e. $\mathbf{X}_t = \mathbf{x}_t \mathbf{x}_t^\top$, where $\|\mathbf{x}_t\| \leq 1$. Such instance matrices have a “sparse spectrum” in the sense that they have at most one non-zero eigenvalue. Our first class consists of the convex hull of outer products of length at most one or equivalently all positive semidefinite matrices of trace norm at most one. We call this class L_1 -*bounded* instance matrices. The most important fact to remember is that the case of L_1 -bounded instances contains vanilla PCA with outer product instances as a special case.

Beginning with some of the early work on linear regression (Kivinen and Warmuth, 1997), it is known that multiplicative updates are especially useful when the non-negative

instance vectors are allowed to be “dense”, i.e. their maximum component is bounded by say one but it could contain many components of size up to one. In the matrix context this means that the symmetric positive semi-definite instance matrices \mathbf{X}_t have maximum eigenvalue (or spectral norm) at most one and are thus “spectrally dense”. We call this second class L_∞ -*bounded* instance matrices.

We will show that MEG is optimal for L_∞ -bounded instance matrices and GD is sub-optimal in this case. However for L_1 -bounded instances one might suspect that MEG is not able to fully exploit the spectral sparsity. For example, in the case of linear regression GD is known to have the advantage when the instance vectors¹ have bounded L_2 norm (Kivinen and Warmuth, 1997) and consistently with that, when GD is used for PCA with L_1 -bounded instance matrices, then its regret is bounded by a term that is *independent* of the dimension of the instances. The advantage of GD in the spectrally sparse case is also supported by a general survey of Mirror Descent algorithms (to which GD and MEG belong) for the case when the gradient vectors of the convex loss functions (which may have negative components) lie in certain symmetric norm balls (Srebro et al., 2011). Again when the gradient vectors of the losses are sparse, then GD has the advantage.

Surprisingly, the situation is quite different for PCA: We show that MEG achieves the same regret bound as GD for online PCA with L_1 -bounded instances matrices (despite the spectral sparseness) and the regret bounds for both algorithms are within a constant factor of a new lower bound proved in this paper that holds for any algorithm for PCA with L_1 -bounded instance matrices. This surprising performance of MEG seems to come from the fact that gradients \mathbf{X}_t of the linear loss $\text{tr}(\mathbf{W}_t \mathbf{X}_t)$ of our generalized online PCA problem are restricted to be non-negative. Therefore our results are qualitatively different from the cases studied in Srebro et al. (2011) where the gradients of the loss functions are within a p -norm ball, i.e. symmetric around zero.

Actually, there are two kinds of regret bounds in the literature: bounds expressed as a function of the time horizon T and bounds that depend on an *upper bound* on the loss of the best comparator (which we call a *loss budget* following Aberkane et al. (2008)). In typical applications for PCA, there exists a low dimensional subspace which captures most of the variance in the data and the compression loss is small. Therefore, guarding against the worst-case loss that grows with the number of trials T is overly pessimistic. We can show that when considering regret bounds as a function of a loss budget, MEG is optimal and strictly better than GD by a factor of \sqrt{n} . This suggests that the multiplicative updates algorithm is the best choice for prediction problems in which the parameters are mixtures of projection matrices and the gradients of the losses are non-negative. Note that in this paper we call an algorithm *optimal* for a particular problem if we can prove an upper bound on its worst-case regret that is within a constant factor of the lower bound for the problem (which must hold for any algorithm).

1.1 Related Work and Our Contribution:

The comparison of the GD and MEG algorithms has an extensive history (see, e.g. Kivinen and Warmuth (1997); Warmuth and Vishwanathan (2005); Sridharan and Tewari (2010); Srebro et al. (2011)). It is simplest to compare algorithms in the case when the loss is

¹ Note that for $\mathbf{x} \in \mathbf{R}^n$, $\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1$.

linear. Linear losses are the least convex losses and in the regret bounds, convex losses are often approximated by first-order Taylor approximations which are linear, and the gradient of the loss function serves as the linear “loss/gain vector” (Kivinen and Warmuth, 1997; Zinkevich, 2003). In this case it is often assumed that the gradient of the loss lies in an L_p ball (which is a symmetric constraint) and the results are as expected: EG is optimal when the parameter space is L_1 -bounded and the gradient vectors are L_∞ -bounded, and GD is optimal when the both spaces are L_2 -bounded (Sridharan and Tewari, 2010; Strebrot et al., 2011).

In contrast for PCA, the gradient of the loss $\text{tr}(\mathbf{W}_t \mathbf{X}_t)$ is the instance matrix \mathbf{X}_t which is assumed to be positive semi-definite. None of the previous work exploits this special property of the PCA setup, where the gradient of the loss satisfies some non-negativity property. In this paper we carefully study this case and show that MEG is optimal.

We also made significant technical progress on the lower bounds for online PCA. The previous lower bounds (Warmuth and Kuzmin (2008) and Koolen et al. (2010)) were incomplete in the following three ways: First, the lower bounds only apply to the case of L_∞ -bounded instances and not to the more restricted case of L_1 -bounded instances. Second, the previous lower bounds assume that the dimension k of target subspace is at least $\frac{n}{2}$ and in common PCA problems, k is much smaller than $\frac{n}{2}$. Third, the proofs rely on the Central Limit Theorem and therefore the resulting lower bounds only hold in the limit as T and n go to infinity (See Cesa-Bianchi et al. (1997); Cesa-Bianchi and Lugosi (2006); Abernethy et al. (2009) for details). In this paper, we circumvent all three weak aspects of the previous proofs: We give lower bounds for all four combinations of L_1 or L_∞ -bounded instance matrices versus $k \leq \frac{n}{2}$ or $k \geq \frac{n}{2}$, respectively. All our lower bounds are non-asymptotic, i.e. they hold for all values of the variables T and n . The new lower bounds use a novel probabilistic bounding argument for the minimum of n random variables. Alternate methods for obtaining non-asymptotic lower bound for label efficient learning problems in the expert setting were given in (Audibert and Bubeck, 2010). However those techniques are more complicated and it is not clear how to adapt them to the online PCA problem.

In summary, our contribution consists of proving tight upper bounds on the regret of the two main online PCA algorithms, as well as proving lower bounds on the regret of any algorithm for solving online PCA. For the case when the regret is expressed as a function of the number of trials T , we show that MEG’s and GD’s regret bounds are independent of the dimension n of the problem and are within a constant factor of the lower bound on the regret of any online PCA algorithm. This means the both algorithms are optimal in this case. For the case when the regret is a function of the loss budget, we prove that MEG remains optimal, while we show that the regret of GD is suboptimal by a \sqrt{k} factor.

Furthermore, for the generalization of the PCA with L_∞ -bounded instance matrices, we improve the known regret bound significantly by switching from a loss version to a gain version of MEG depending on the dimension k of the subspace. If $k \geq \frac{n}{2}$, then the gain version of MEG is optimal for L_∞ -bounded instances, and when $k \leq \frac{n}{2}$, then the loss version is optimal. On the other hand, GD is non-optimal for both ranges of k .

A much shorter preliminary version of this manuscript appeared in the 24th International Conference on Algorithmic Learning Theory (2013) (Nie et al., 2013). In this more detailed journal version we give more background and complete proofs of all of our results (mostly omitted or only sketched in the conference version). This paper also has the following

additional material: A proof of the budget bound (3.5) for the gain version of MEG; an extension of the lower bound on the regret of GD (Theorem 4.1) to the case of small budgets; the analysis of the Follow the Regularized Leader variant of GD (Section 4.2) and a discussion of its final parameter matrix (Appendix E); lower bounds on the regret when the number of trials is small (Appendix G).

1.2 Outline of the Paper:

In Section 2, we start with describing the MEG and GD algorithms for online PCA. In particular, we present two versions of the MEG algorithm: the Loss MEG algorithm introduced in (Warmuth and Kuzmin, 2008), and the Gain MEG algorithm, which is the same as Loss MEG except for a sign change in the exponential. Following the description of each algorithm, we then derive in Section 3 their regret bounds expressed as functions of the number of trials T . These bounds are compared in Section 3.2 for all four combinations of L_1 or L_∞ -bounded instance matrices versus $k \leq \frac{n}{2}$ or $k \geq \frac{n}{2}$, respectively (see Table 3.2). Next we consider regret bounds expressed as functions of the loss budget. In Section 4, we prove a lower bound on GD’s regret which shows that the regret of GD is at least \sqrt{k} times larger than the regret of Loss EG. A similar lower bound is proved for the Follow the Regularized Leader variant of GD in Section 4.2. In Section 5 we prove lower bounds for online PCA with L_1 and L_∞ -bounded instances that hold for any online algorithm, and in Section 6 we conclude with a summary of which algorithms are optimal.

2. The Online Algorithms

Online uncentered PCA uses the following protocol in each trial $t = 1, \dots, T$: the algorithm probabilistically chooses a projection matrix $\mathbf{P}_t \in \mathbb{R}^{n \times n}$ of rank k . Then a point $\mathbf{x}_t \in \mathbb{R}^n$ is received and the algorithm suffers the loss $\text{tr}((\mathbf{I} - \mathbf{P}_t)\mathbf{x}_t\mathbf{x}_t^T)$.

We also consider the generalization where the instance matrix is any positive definite matrix \mathbf{X}_t instead of an outer product $\mathbf{x}_t\mathbf{x}_t^T$. In that case the loss of the algorithm is $\text{tr}((\mathbf{I} - \mathbf{P}_t)\mathbf{X}_t)$. As discussed in the introduction (e.g. Equation (1.2)), this linear loss has a compression loss interpretation. It is “complementary” to the gain $\text{tr}(\mathbf{P}_t\mathbf{X}_t)$, i.e.

$$\underbrace{\text{tr}((\mathbf{I} - \mathbf{P}_t)\mathbf{X}_t)}_{\text{loss}} = \underbrace{\text{tr}(\mathbf{X}_t)}_{\text{constant}} - \underbrace{\text{tr}(\mathbf{P}_t\mathbf{X}_t)}_{\text{gain}}$$

and the $n - k$ dimensional projection matrix $\mathbf{I} - \mathbf{P}_t$ is “complementary” to the k dimensional projection matrix \mathbf{P}_t . These two complementations are inherent to our problem and will be present throughout the paper.

In the above protocol, the algorithm is allowed to choose its k dimensional subspace \mathbf{P}_t probabilistically. Therefore we use the expected compression loss $\mathbb{E}[\text{tr}((\mathbf{I} - \mathbf{P}_t)\mathbf{X}_t)]$ as the loss of the algorithm. The regret of the algorithm is then the difference between its cumulative loss and the loss of the best k subspace:

$$\mathcal{R} = \sum_{t=1}^T \mathbb{E}[\text{tr}((\mathbf{I} - \mathbf{P}_t)\mathbf{X}_t)] - \min_{\mathbf{P} \text{ projection matrix of rank } k} \sum_{t=1}^T \text{tr}((\mathbf{I} - \mathbf{P})\mathbf{X}_t).$$

The regret can also be rewritten in terms of gain, but this gives the same value of the regret. Therefore, throughout the paper we use (expected) losses and “loss” regrets (as defined above) to evaluate the algorithms.

Now we rewrite the loss of the algorithm as $\text{tr}(\mathbb{E}[\mathbf{I} - \mathbf{P}_t] \mathbf{X}_t)$ which shows that for any random prediction \mathbf{P}_t of rank k , this loss is fully determined by $\mathbb{E}[\mathbf{I} - \mathbf{P}_t]$, a convex combination of rank $m = n - k$ projection matrices. Hence it is natural to choose the set \mathcal{W}_m of convex combinations of rank m projection matrices as the parameter set of the algorithm. By the definition of projection matrices, \mathcal{W}_m is the set of positive semi-definite matrices of trace m and eigenvalues not larger than 1. The current parameter $\mathbf{W}_t \in \mathcal{W}_m$ of the online algorithm expresses its “uncertainty” about which subspace of rank m is best for the online data stream seen so far and the (expected) loss in trial t becomes $\text{tr}(\mathbf{W}_t \mathbf{X}_t)$. Alternatively, the complementary set \mathcal{W}_k of rank k projection matrices can be used as the parameter set (in that case the loss is $\text{tr}((\mathbf{I} - \mathbf{W}_t) \mathbf{X}_t)$). As discussed, there is a one-to-one correspondence between the two parameter sets: Given $\mathbf{W} \in \mathcal{W}_k$, then $\mathbf{I} - \mathbf{W}$ is the corresponding convex combination in \mathcal{W}_m .

The second reason why convex combinations are natural parameter spaces is that since the loss is linear, the convex combination with the minimum loss occurs at a “pure” projection matrix, i.e.

$$\begin{aligned} \min_{\mathbf{W} \in \mathcal{W}_m} \sum_{t=1}^T \text{tr}(\mathbf{W} \mathbf{X}_t) &= \min_{\substack{\mathbf{P} \\ \text{matrix of rank } k}} \sum_{t=1}^T \text{tr}((\mathbf{I} - \mathbf{P}) \mathbf{X}_t) \quad \text{and} \\ \min_{\mathbf{W} \in \mathcal{W}_k} \sum_{t=1}^T \text{tr}((\mathbf{I} - \mathbf{W}) \mathbf{X}_t) &= \min_{\substack{\mathbf{P} \\ \text{matrix of rank } k}} \sum_{t=1}^T \text{tr}(\mathbf{I} - \mathbf{P}) \mathbf{X}_t. \end{aligned} \quad (2.1)$$

Our protocol requires the algorithm to predict with a rank k projection matrix. Therefore, given a parameter matrix \mathbf{W}_t in say \mathcal{W}_m , the online algorithm still needs to produce a random projection matrix \mathbf{P}_t of rank k at the beginning of trial t such that $\mathbb{E}[\mathbf{I} - \mathbf{P}_t] = \mathbf{W}_t$. A simple greedy algorithm for achieving this is given in (Warmuth and Kuzmin, 2008) (Algorithm 2) which efficiently decomposes \mathbf{W}_t into a convex combination of up to n projection matrices of rank m in general, followed by a *mixture decomposition* of \mathbf{W}_t , which has $O(n^3)$ time complexity in general, followed by a *mixture decomposition* of the eigenvalues which runs in $O(n^2)$ time). Using the mixture coefficients it is now easy to sample a projection matrix $\mathbf{I} - \mathbf{P}_t$ from parameter matrix \mathbf{W}_t .

We now motivate the two main online algorithms used in this paper: the GD and MEG algorithms. The GD algorithm is straightforward and the MEG algorithm was introduced in Tsuda et al. (2005). Both are examples of the *Mirror Descent* family of algorithms developed much earlier in the area of convex optimization (Nemirovski and Yudin, 1978). The Mirror Descent algorithms update their parameter by minimizing a trade-off function of a divergence between the new and old parameter and the loss of the new parameter on the current instance, while constraining the new parameter to lie in the parameter set.

For the problem of online PCA, the update specializes into the following two versions depending on the choice of the parameter set:

Loss update on parameter set \mathcal{W}_m (i.e., $\mathbf{W}_{t+1}, \mathbf{W}, \mathbf{W}_t \in \mathcal{W}_m$):

$$\mathbf{W}_{t+1} = \underset{\mathbf{W} \in \mathcal{W}_m}{\text{argmin}} (\Delta(\mathbf{W}, \mathbf{W}_t) + \eta \text{tr}(\mathbf{W} \mathbf{X}_t)). \quad (2.2)$$

Gain update on parameter set \mathcal{W}_k (i.e., $\mathbf{W}_{t+1}, \mathbf{W}, \mathbf{W}_t \in \mathcal{W}_k$):

$$\begin{aligned} \mathbf{W}_{t+1} &= \underset{\mathbf{W} \in \mathcal{W}_k}{\text{argmin}} (\Delta(\mathbf{W}, \mathbf{W}_t) + \eta \text{tr}((\mathbf{I} - \mathbf{W}) \mathbf{X}_t)) \\ &= \underset{\mathbf{W} \in \mathcal{W}_k}{\text{argmin}} (\Delta(\mathbf{W}, \mathbf{W}_t) - \eta \text{tr}(\mathbf{W} \mathbf{X}_t)). \end{aligned} \quad (2.3)$$

Here $\Delta(\mathbf{W}, \mathbf{W}_t)$ is the motivating Bregman divergence that will be different for the MEG and GD algorithms. The *Loss update* minimizes a trade-off with the expected loss $\text{tr}(\mathbf{W} \mathbf{X}_t)$ which is a matrix version of the dot loss used for motivating the Hedge algorithm (Freund and Schapire, 1995). Note that in the *gain* version, minimizing the loss $-\text{tr}(\mathbf{W} \mathbf{X}_t)$ is the same as maximizing the gain $\text{tr}(\mathbf{W} \mathbf{X}_t)$. Recall that there is a one-to-one correspondence between \mathcal{W}_m and \mathcal{W}_k , i.e. \mathbf{I} minus a parameter in \mathcal{W}_m gives the corresponding parameter in \mathcal{W}_k and vice versa. Therefore, one can for example rewrite the Gain update (2.3) with the parameter set \mathcal{W}_m as well:

$$\widetilde{\mathbf{W}}_{t+1} = \underset{\mathbf{W} \in \mathcal{W}_m}{\text{argmin}} (\Delta(\mathbf{I} - \mathbf{W}, \mathbf{I} - \widetilde{\mathbf{W}}_t) + \eta \text{tr}(\mathbf{W} \mathbf{X}_t)), \quad (2.4)$$

where the above solution $\widetilde{\mathbf{W}}_{t+1} \in \mathcal{W}_m$ of the Gain update is related to the solution $\mathbf{W}_{t+1} \in \mathcal{W}_k$ of (2.3) by the same complementary relationship, i.e. $\widetilde{\mathbf{W}}_{t+1} = \mathbf{I} - \mathbf{W}_{t+1}$, for $t = 1, \dots, T$. Notice that the Loss update is motivated by the divergence $\Delta(\mathbf{W}, \mathbf{W}_t)$ on parameter space \mathcal{W}_m (2.2). On the other hand, when the Gain update is formulated with parameter \mathcal{W}_m , then it is motivated by the divergence $\Delta(\mathbf{I} - \mathbf{W}, \mathbf{I} - \widetilde{\mathbf{W}}_t)$ (2.4).

Now we define the GD and MEG algorithms for online PCA. For the GD algorithm, the motivating Bregman divergence is the squared Frobenius norm between the old and new parameters: $\Delta(\mathbf{W}, \mathbf{W}_t) = \frac{1}{2} \|\mathbf{W} - \mathbf{W}_t\|_F^2$ (Kivinen and Warmuth, 1997; Zinkevich, 2003). With this divergence, the Loss update is solved in the following two steps:

$$\begin{aligned} \text{GD update:} \quad & \text{Descent step:} \quad \widehat{\mathbf{W}}_{t+1} = \mathbf{W}_t - \eta \mathbf{X}_t, \\ & \text{Projection step:} \quad \mathbf{W}_{t+1} = \underset{\mathbf{W} \in \mathcal{W}_m}{\text{argmin}} \|\mathbf{W} - \widehat{\mathbf{W}}_{t+1}\|_F^2. \end{aligned} \quad (2.5)$$

Note, that the split into two steps happens whenever a Bregman divergence is traded off with a linear loss and domain is convex (See Helmbold and Warmuth (2009), Section 5.2, for a discussion). For the squared Frobenius norm, the Gain update is equivalent to the Loss update, since when formulating both updates on parameter set \mathcal{W}_m , then the divergence $\|\mathbf{W} - \mathbf{W}_t\|_F^2$ of the Loss update (2.2) and the divergence $\|(\mathbf{I} - \mathbf{W}) - (\mathbf{I} - \mathbf{W}_t)\|_F^2$ of the Gain update (2.4) are the same. A procedure for projecting $\widehat{\mathbf{W}}_{t+1}$ into \mathcal{W}_m with respect to the squared Frobenius norm is given in Algorithm 2 of Arora et al. (2013). The expensive part of this procedure is obtaining the eigendecomposition of $\widehat{\mathbf{W}}_{t+1}$.

The MEG algorithm uses the (un-normalized) quantum relative entropy $\Delta(\mathbf{W}, \mathbf{W}_t) = \text{tr}(\mathbf{W} (\log \mathbf{W} - \log \mathbf{W}_t) + \mathbf{W}_t - \mathbf{W})$ as its motivating Bregman divergence (Tsuda et al.,

T	Number of trials
n	Dimension of data points $\mathbf{x}_t \in \mathbb{R}^n$ and instance matrices $\mathbf{X}_t \in \mathbb{R}^{n \times n}$
k	Rank of the subspace of PCA into which the data is projected
m	Complement of k , $m = n - k$ (used for the rank of subspace of Loss MEG)
L_1 -bounded instances	positive semi-definite matrices \mathbf{X}_t s.t. $\text{tr}(\mathbf{X}_t) \leq 1$ (subsumes the special case when the \mathbf{X}_t are of the form $\mathbf{x}_t \mathbf{x}_t^\top$, w. $\ \mathbf{x}_t\ \leq 1$)
L_∞ -bounded instances	positive semi-definite matrices \mathbf{X}_t with spectral norm at most one, that is $\lambda_{\max}(\mathbf{X}_t) \leq 1$
B_L	Upper bound on loss of best subspace of rank $n - k$, c.f. (3.1)
B_G	Upper bound on gain of best subspace of rank k , c.f. (3.2).

Table 3.1: Summary of various symbols and terms used in Section 3.

2005) which is based on the matrix logarithm \log . With this divergence the solutions to the Loss update (2.2) and Gain update (2.3) are the following expressions which make use of the matrix exponential \exp (the inverse of \log):

$$\begin{aligned} \text{Loss MEG update:} \quad & \widehat{\mathbf{W}}_{t+1} = \exp(\log \mathbf{W}_t - \eta \mathbf{X}_t), & (2.6) \\ \text{Projection step:} \quad & \mathbf{W}_{t+1} = \underset{\mathbf{W} \in \mathbf{W}_m}{\text{argmin}} \Delta(\mathbf{W}, \widehat{\mathbf{W}}_{t+1}). \end{aligned}$$

$$\begin{aligned} \text{Gain MEG update:} \quad & \widehat{\mathbf{W}}_{t+1} = \exp(\log \mathbf{W}_t + \eta \mathbf{X}_t), & (2.7) \\ \text{Projection step:} \quad & \mathbf{W}_{t+1} = \underset{\mathbf{W} \in \mathbf{W}_k}{\text{argmin}} \Delta(\mathbf{W}, \widehat{\mathbf{W}}_{t+1}). \end{aligned}$$

Note that the only difference between the gain and loss versions of MEG is a sign flip in the exponential. The projection steps in the algorithms are with respect to the quantum relative entropy. An efficient procedure for solving such projections is given in Algorithm 4 of Warmuth and Kuzmin (2008); it does a projection with respect to the standard relative entropy on the vector of eigenvalues of the parameter matrix. Finally note that the computational complexity of all described updates (GD, Loss MEG, Gain MEG) is dominated by the time required for obtaining the eigendecomposition of the parameter matrix \mathbf{W}_{t+1} (or $\widehat{\mathbf{W}}_{t+1}$), which is $O(n^3)$ in general.

3. Upper Bounds on the Regret

Recall that the instance matrices \mathbf{X}_t are always assumed to be positive semi-definite matrices. We call such instance matrices L_1 -bounded, if the trace norm of the instance matrices is at most one, i.e. $\text{tr}(\mathbf{X}_t) \leq 1$ always holds. In particular, this happens for the vanilla PCA setting where the data received at trial is a point $\mathbf{x}_t \in \mathbb{R}^n$ s.t. $\|\mathbf{x}_t\| \leq 1$. In this case the instance matrices have the form $\mathbf{X}_t = \mathbf{x}_t \mathbf{x}_t^\top$ and $\text{tr}(\mathbf{x}_t \mathbf{x}_t^\top) = \mathbf{x}_t^\top \mathbf{x}_t = \|\mathbf{x}_t\|^2 \leq 1$. Note that in the L_1 -bounded case, the sums of the eigenvalues of the \mathbf{X}_t are at most one. We also study the case when the maximum eigenvalue of the instance matrices \mathbf{X}_t is at most one and call the latter the L_∞ -bounded case.

In this section, we present regret upper bounds for the three online algorithms introduced in the previous section, which are Loss MEG, Gain MEG and GD. All three algorithms are examples from the Mirror Descent family of algorithms. Our proof techniques require us to use different restrictions on the worst-case sequences that the adversary can produce. For the Loss MEG algorithm, we give the adversary a *loss budget*, i.e. the adversary must produce a sequence of instances $\mathbf{X}_1 \dots \mathbf{X}_T$ for which the loss of the best subspace is upper bounded by the loss budget B_L :

$$\min_{\substack{\mathbf{P} \text{ projection} \\ \text{matrix of rank } k}} \sum_{t=1}^T \text{tr}((\mathbf{I} - \mathbf{P}) \mathbf{X}_t) \leq B_L. \quad (3.1)$$

We call a regret bound that depends on this parameter a *loss budget dependent* bound. A bound of this type was first proved for Loss MEG in Warmuth and Kuzmin (2008). The latter paper is the precursor of this paper in which the analysis of online algorithms for PCA was started.

For the algorithm of Gain MEG, we give the adversary a *gain budget* B_G , i.e. an upper bound on the gain of the best subspace:

$$\max_{\substack{\mathbf{P} \text{ projection} \\ \text{matrix of rank } k}} \sum_{t=1}^T \text{tr}(\mathbf{P} \mathbf{X}_t) \leq B_G. \quad (3.2)$$

Now the adversary can only produce sequences for which all subspaces have gain at most B_G . We call this type of bound a *gain budget dependent* bound.

Finally we prove regret bounds of a third type for the GD algorithm. For this type the regret is a function of the number of trials T , and we call such a regret bound a *time dependent* regret bound.

We present the three regret bounds in the next subsection and compare them in the following subsection. As we shall see, upper bounds of the regret in terms of a budget imply time dependent bounds, and for lower bounds the implication is reversed. The main symbols and terms used throughout this section are summarized in Table 3.1.

3.1 Upper Bounds on the Regret of Loss MEG, Gain MEG, and GD

The Loss MEG algorithm (2.6) is the original MEG algorithm developed in the precursor paper of Warmuth and Kuzmin (2008) for online PCA. This paper proves a loss budget dependent upper bound on the regret of Loss MEG. This is done by exploiting the fact that PCA learning has the so called expert setting as a special case (See extensive discussion at the beginning of Section 4). More precisely the following bound is proven by lifting a regret bound developed for learning well compared to the best subset of $m = n - k$ experts to the matrix case, where subsets of size m generalize to projection matrices of rank m .

Loss budget dependent bound of Loss MEG:

$$\mathcal{R}_{\text{Loss MEG}} \leq \sqrt{2B_L m \log \frac{n}{m}} + m \log \frac{n}{m}. \quad (3.3)$$

This bound follows from Theorem 6 of Warmuth and Kuzmin (2008), and holds for any sequence of instance matrices (L_∞ as well as L_1 -bounded) for which the total compression loss of the best rank m subspace does not exceed the loss budget B_L (Condition (3.1)).

We begin by showing that the right-hand side of (3.3) is bounded above by an expression that does not depend on the dimension n of the data points:

$$\mathcal{R}_{\text{Loss MEG}} \leq \sqrt{2B_L k} + k. \quad (3.4)$$

This follows immediately from the following inequality and the relationship $m = n - k$ ($n = m + k$):

$$m \log \frac{n}{m} = m \log \left(\frac{k+m}{m} \right) = m \log \left(1 + \frac{k}{m} \right) \leq m \frac{k}{m} = k.$$

As mentioned at the beginning of this subsection (and discussed in more detail later in Section 4), online PCA specializes to the problem of learning well compared to the best set of $m = n - k$ experts. Regret bounds for the expert setting typically depend logarithmically on the number of experts n . Therefore the above dimension free regret bound might seem puzzling at first. However there is no contradiction. In the current setup we have $m = 1$ and $k = n - m = n - 1$ for the vanilla single expert case, and the above dimension free bound (3.4) becomes $\sqrt{2B_L(n-1)}$. This bound is not close to the optimum loss budget dependent regret bound for the single expert case which is $O(\sqrt{B_L \log n} + \log n)$. This latter bound is obtained by plugging $m = 1$ into the *original* regret bound (3.3). Thus for $m = 1$, the above dimension free approximation (3.4) of the original bound is loose. However, when $k \leq \frac{n}{2}$, then as we shall see in Section 5, the dimension free approximation actually is tight. In the precursor paper (Warmuth and Kuzmin, 2008), a different but weaker approximation of the original bound was proved that still has an additional logarithmic dependence when $k \leq \frac{n}{2}$: $O(\sqrt{B_L k \log \frac{n}{k}} + k \log \frac{n}{k})$.

We next develop a regret bound for Gain MEG (2.7). The proof technique is a variation of the original regret bound for Loss MEG (and is given for the sake of completeness in Appendix A).

Gain budget dependent bound of Gain MEG:

$$\mathcal{R}_{\text{Gain MEG}} \leq \sqrt{2B_G k \log \frac{n}{k}}. \quad (3.5)$$

This bound holds for any sequence of instance matrices (L_1 as well as L_∞ -bounded) for which the total gain of the best rank k subspace does not exceed the gain budget B_G (Condition (3.2)).

Finally, we give a simple regret bound for the GD algorithm. This bound (also observed in Arora et al. (2013) and proved for the sake of completeness in Appendix B) is based on two standard techniques: the use of the squared Frobenius norm (Kivinen and Warmuth, 1997) as a measure of progress and the use of the Pythagorean Theorem for handling the projection step (Herbst and Warmuth, 2001).

Time dependent regret bound of GD:

$$\mathcal{R}_{\text{GD}} \leq \begin{cases} \sqrt{\frac{T}{n}} & \text{for } L_1\text{-bounded instances} \\ \sqrt{T k m} & \text{for } L_\infty\text{-bounded instances} \end{cases}. \quad (3.6)$$

Note that each regret bound is expressed as a function of a loss budget, a gain budget or a time bound. They are obtained by setting the fixed learning rate of the algorithm as a function of one of these three parameters. The resulting basic algorithms can be used as sub-modules: For example the algorithm can be stopped as soon as the loss budget is reached and restarted with twice the budget and the corresponding re-tuned learning rate. This heuristic is known as the ‘‘doubling trick’’ (Cesa-Bianchi et al., 1997). Much fancier tuning schemes are explored in (van Erven et al., 2011; de Rooij et al., 2014) and are not the focus of this paper.

3.2 Comparison of the Regret Upper Bounds

Our goal is to find algorithms that achieve the optimal loss budget dependent and time dependent regret bounds where optimal means that the bound is within a constant factor of optimum. We are not interested in *gain dependent* regret bounds per se, i.e. bounds in terms of a gain budget B_G , because the maximal gain is typically much larger than the minimal loss. However when the gain budget restricted regret bounds are converted to time bounds, then for some setting (discussed below) the resulting algorithm becomes the only optimal algorithm we are aware of.

The only known *loss budget dependent* regret bound is bound (3.3) for Loss MEG obtained in the original paper for online learning of PCA (Warmuth and Kuzmin, 2008). We will show later in Section 5 that this upper bound on the regret is optimal. There are no known loss budget dependent upper bounds on the regret of GD. However in Section 4, we prove a lower bound on GD’s regret in terms of the loss budget which shows that GD’s regret is suboptimal by at least a factor of \sqrt{k} when the regret is expressed as a function of the loss budget. The discussion of the *time dependent* regret upper bounds is more involved. We first convert the budget dependent regret bounds of the MEG algorithms into time dependent bounds. We shall see later, for lower bounds on the regret, time dependent bounds lead to budget dependent bounds (see Corollary 5.7). Before we do this, recall that the instance matrices \mathbf{X}_t are L_1 -bounded if their trace is at most one, and for L_∞ -bounded instance matrices, their maximum eigenvalue is at most one. Note that for any vector \mathbf{x}_t of length at most one, $\text{tr}(\mathbf{x}_t \mathbf{x}_t^\top) \leq 1$, and therefore vanilla PCA belongs to the case of L_1 -bounded instance matrices.

Theorem 3.1 *When the instances are L_1 -bounded, then for the online PCA with T trials, the following regret bounds hold for the Loss MEG and Gain MEG algorithms, respectively:*

$$\mathcal{R}_{\text{Loss MEG}} \leq m \sqrt{\frac{2T}{n}} \log \frac{n}{m} + m \log \frac{n}{m}, \quad \mathcal{R}_{\text{Gain MEG}} \leq \sqrt{2T k \log \frac{n}{k}}. \quad (3.7)$$

Similarly, when the instances are L_∞ -bounded, then the following regret bounds hold:

$$\mathcal{R}_{\text{Loss MEG}} \leq m \sqrt{2T \log \frac{n}{m}} + m \log \frac{n}{m}, \quad \mathcal{R}_{\text{Gain MEG}} \leq k \sqrt{2T \log \frac{n}{k}}. \quad (3.8)$$

Proof The theorem will be proved by developing simple upper bounds on the loss/gain of the best rank k subspace that depend on the sequence length T . These upper bounds are then used as budgets in the previously obtained budget dependent bounds.

The best rank k subspace picks k eigenvectors of the covariance matrix $\mathbf{C} = \sum_{t=1}^T \mathbf{X}_t$ with the largest eigenvalues. Hence the total compression loss equals the sum of the smallest m eigenvalues of \mathbf{C} . If $\omega_1, \dots, \omega_n$ denote all the eigenvalues of \mathbf{C} , then:

$$\sum_{i=1}^n \omega_i = \text{tr}(\mathbf{C}) = \sum_{t=1}^T \text{tr}(\mathbf{X}_t) \leq \begin{cases} T & \text{for } L_1\text{-bounded instances} \\ Tn & \text{for } L_\infty\text{-bounded instances} \end{cases},$$

where the inequality follows from our definition of L_1 -bounded and L_∞ -bounded instance matrices. This implies that the sum of the m smallest eigenvalues is upper bounded by $\frac{Tm}{n}$ and Tm , respectively. By using these two bounds as the loss budget B_L in (3.3), we get the time dependent bound for Loss MEG for L_1 -bounded and L_∞ -bounded instances, respectively.

For the regret bounds of Gain MEG, we use the fact that B_G is upper bounded by T when instances are L_1 -bounded and upper bounded by kT when the instances are L_∞ -bounded, and plug these values for B_G into (3.5). ■

Table 3.2 compares time dependent upper bounds for each of the three algorithms (Loss MEG, Gain MEG, GD) where we consider each of the 4 variants of the problem: L_1 -bounded or L_∞ -bounded instance matrices versus $k \leq \frac{n}{2}$ or $k \geq \frac{n}{2}$.

As far as time dependent bounds are concerned, no single algorithm is optimal in all cases. In Table 3.2, the optimum bounds are shown in bold. The lower bounds matching these bold bounds within a constant factor will be proved in Section 5. Note that one version of MEG (either the loss or gain version) is optimal in each case, while GD is optimal only in first case (This is the most important case in practice: vanilla online PCA with $k \ll n$). For the remaining three cases, consider the ratio between the GD's bound and the better of the two MEG bounds, which is

- $\sqrt{\frac{n}{m} / (\log \frac{n}{m})}$, when the instances are L_1 -bounded and $k \geq \frac{n}{2}$,
- $\sqrt{\frac{n}{k} / (\log \frac{n}{k})}$, when the instances are L_∞ -bounded and $k \leq \frac{n}{2}$ and
- $\sqrt{\frac{n}{m} / (\log \frac{n}{m})}$, when the instances are L_∞ -bounded and $k \geq \frac{n}{2}$.

Since none of these three ratios can be upper bounded by a constant, GD is clearly suboptimal in each of the remaining three cases.

	L_1 -bounded instances		L_∞ -bounded instances	
	$k \leq \frac{n}{2}$	$k \geq \frac{n}{2}$	$k \leq \frac{n}{2}$	$k \geq \frac{n}{2}$
Loss MEG	\sqrt{Tk}	$\sqrt{Tm} (\log \frac{n}{m}) / \frac{n}{m}$	\sqrt{Tkm}	$\sqrt{Tm^2 \log \frac{n}{m}}$
Gain MEG	$\sqrt{Tk \log \frac{n}{k}}$	\sqrt{Tm}	$\sqrt{Tk^2 \ln \frac{n}{k}}$	\sqrt{Tkm}
GD	\sqrt{Tk}	\sqrt{Tm}	\sqrt{Tkm}	\sqrt{Tkm}

Table 3.2: Comparison of the time dependent upper bounds on the regret of the Loss MEG, Gain MEG, and GD algorithms. Each column corresponds to one of the four combinations of L_1 -bounded or L_∞ -bounded instance matrices versus $k \leq \frac{n}{2}$ or $k \geq \frac{n}{2}$, respectively. All bounds were given in Section 3.1 and Section 3.2: constants are omitted, we only show the leading term of each bound, and when we compare Loss and Gain MEG bounds, we use $m \ln \frac{n}{m} = \Theta(k)$ when $k \leq \frac{n}{2}$ and $k \ln \frac{n}{k} = \Theta(m)$ when $k \geq \frac{n}{2}$. Recall that m is shorthand for $n - k$. The best (smallest) bound for each case (column) is shown in bold. In Section 5, all bold bounds will be shown to be optimal (within constant factors).

4. Lower Bounds on the Regret of GD

Recall that vanilla online PCA uses L_1 -bounded instance matrices and the subspace dimension k is typically at most $\frac{n}{2}$. In this case Loss MEG has regret $O(\sqrt{Tk})$ and the regret of GD is $O(\sqrt{Tk})$ as well. As for loss budget dependent regret bounds, Loss MEG has regret $O(\sqrt{B_L k} + k)$ and we initially conjectured that GD has the same bound. However, this is not true: we will now show in this section an $\Omega(\max\{B_L, k\sqrt{B_L}, k\})$ lower bound on the regret of GD for L_1 -bounded instance sequences when $k \leq \frac{n}{2}$. In contrast, Loss MEG's regret bound of $O(\sqrt{B_L k} + k)$ will be shown to be optimal in Section 5 for this case. It follows that GD is suboptimal by at least a factor of \sqrt{k} when $B_L = \Omega(k^2)$. A detailed comparison of the lower bound for GD and the optimum upper bound is given in Table 4.1.

It suffices to prove lower bounds on GD's regret on a restricted class of instance matrices: We assume that all instance matrices are in the same eigensystem, i.e. they are diagonal matrices $\mathbf{X} = \text{diag}(\boldsymbol{\ell})$ with $\boldsymbol{\ell} \in \mathbb{R}_{\geq 0}^n$. We call the diagonals $\boldsymbol{\ell}$ the *loss vectors*. All loss vectors in our lower bounds are restricted to be bit vectors in $\{0, 1\}^n$. In the L_1 -bounded instance case, the loss vectors are further restricted to be one of the n unit bit vectors \mathbf{e}_i , i.e. $\mathbf{X} = \text{diag}(\mathbf{e}_i) = \mathbf{e}_i \mathbf{e}_i^\top$. In the L_∞ -bounded instance case, the loss vectors $\boldsymbol{\ell}$ are arbitrary n -dimensional bit vectors.

When all instance matrices are diagonal then the off-diagonal elements in a parameter matrix \mathbf{W} are irrelevant and therefore the algorithm's loss and regret is determined by the diagonals of the parameter matrices \mathbf{W} which is of trace m . Therefore without loss of generality we can assume that the parameter matrices are diagonal as well, i.e. $\mathbf{W} = \text{diag}(\mathbf{w})$ where \mathbf{w} is a *weight vector* in $[0, 1]^n$ with total weight m . Note that the loss becomes

Regret bounds for L_1 -bounded instances, $k \leq \frac{n}{2}$	$B_L \leq k$	$k \leq B_L \leq k^2$	$k^2 \leq B_L$
Upper bound on regret of Loss MEG (see (3.3))	$O(k)$	$O(\sqrt{B_L k})$	$O(\sqrt{B_L k})$
Lower bound on regret of GD (see Theorem 4.1)	$\Omega(k)$	$\Omega(B_L)$	$\Omega(k\sqrt{B_L})$

Table 4.1: Comparison of the loss budget dependent regret bounds for online PCA with $k \leq \frac{n}{2}$. Given dimension k of the subspace, each column shows the values of the two bounds for a specific range of the loss budget B_L . The first row gives the upper bound on the regret of Loss MEG in bold, which will be shown to be optimal in Section 5. The second row gives the lower bound on the regret of GD, which is suboptimal whenever $B_L \geq k$.

a dot product between the weight vector and the loss vector:

$$\text{tr}(\mathbf{W}\mathbf{X}) = \text{tr}(\text{diag}(\mathbf{w}) \text{diag}(\boldsymbol{\ell})) = \mathbf{w} \cdot \boldsymbol{\ell}.$$

What is the prediction of the algorithm with a diagonal parameter matrix $\mathbf{W} = \text{diag}(\mathbf{w})$? It probabilistically predicts with an m dimensional projection matrix \mathbf{P} s.t. $\mathbb{E}[\mathbf{P}] = \text{diag}(\mathbf{w})$. This means \mathbf{P} is a subset of size m from $\{\mathbf{e}_1\mathbf{e}_1^\top, \mathbf{e}_2\mathbf{e}_2^\top, \dots, \mathbf{e}_n\mathbf{e}_n^\top\}$. The diagonals of such projection matrices consists of exactly m ones and $n - m = k$ zeros. In other words the diagonals are indicator vectors of the chosen *subsets* of size m and the expected indicator vector equals the weight vector \mathbf{w} .

We just outlined one of the main insights of (Warmuth and Kuzmin, 2008): The restriction of the PCA problem to diagonal matrices corresponds to learning a subset of size m . The n components of the vectors are usually called *experts*. At trial t the algorithm chooses a subset of m experts. It then receives a loss vector $\boldsymbol{\ell} \in \mathbb{R}_{\geq 0}^n$ for the experts and incurs the total loss of the chosen m experts. The algorithm maintains its uncertainty over the m -sets by means of a parameter vector $\mathbf{w} \in [0, 1]^n$ with total weight m , and it chooses the subset of size m probabilistically so that the expected indicator vector equals \mathbf{w} . We denote the set of such parameter vectors as \mathcal{S}_m . In the L_1 -bounded instance case, the loss vector is a unit bit vector (only one expert incurs a unit of loss). In the L_∞ -bounded instance case, the loss vectors are restricted to be n -dimensional bit vectors.

4.1 Lower Bound on the Regret of the GD Algorithm

The GD algorithm for online PCA (2.5) specializes to the following update of the parameter vector for learning sets:

$$\begin{aligned} \text{Descent step:} \quad \hat{\mathbf{w}}_{t+1} &= \mathbf{w}_t - \eta \boldsymbol{\ell}_t, \\ \text{Projection step:} \quad \mathbf{w}_{t+1} &= \text{argmin}_{\mathbf{w} \in \mathcal{S}_m} \|\mathbf{w} - \hat{\mathbf{w}}_{t+1}\|^2. \end{aligned} \quad (4.1)$$

We now give a lower bound on the regret of the GD algorithm for the m -set problem. This lower bound is expressed as a function of the loss budget.

Theorem 4.1 *Consider the $m = n - k$ set problem with $k \leq n/2$ and unit bit vectors as loss vectors. Then for any fixed learning rate $\eta \geq 0$, the GD algorithm (4.1) can be forced to have regret $\Omega(\max\{\min\{B_L, k\sqrt{B_L}\}, k\})$.*

We prove this theorem in Appendix C. From the fact that m -set problem is a special case of PCA problem, we get the following corollary, which shows that the GD algorithm is suboptimal (see Table 4.1 for an overview):

Corollary 4.2 *Consider the PCA problem with $k \leq n/2$ and L_1 -bounded instance matrices. Then for any fixed learning rate $\eta \geq 0$, the GD algorithm (2.5) can be forced to have regret $\Omega(\max\{\min\{B_L, k\sqrt{B_L}\}, k\})$.*

4.2 Lower Bound on the Regret of the Follow the Regularized Leader GD Algorithm (FRL-GD)

In the previous section, we showed that for online PCA with L_1 -bounded instance matrices and $k \leq \frac{n}{2}$, the GD algorithm is suboptimal for loss budget dependent regret bounds. However, our lower bounds are only for the Mirror Descent version of GD given in (2.5). This algorithm is prone to “forgetting” lots of information about the past losses when projections with respect to inequality constraints are involved. Recall that at the end of each trial t , the mirror descent algorithm uses the last parameter \mathbf{W}_t as a summary of the knowledge attained so far, and minimizes a trade-off between a divergence to the \mathbf{W}_t and the loss on the last data point \mathbf{x}_t to determine the next parameter \mathbf{W}_{t+1} . When the parameter resulting from the trade-off lies outside the parameter set, then it is projected back into the parameter set (see update (2.5)). In the case when the projection enforces inequality constraints on the parameters, information about the past losses may be lost. This issue was first discussed in Section 5.5 of Helmbold and Warmuth (2009). Curiously enough, Bregman projections with respect to only equality constraints do not lose information.

We now demonstrate in more detail the “forgetting” issue for the Mirror Descent GD algorithm when applied to online PCA. First recall that the batch PCA solution consists of the subspace spanned by the k eigenvectors belonging to the k largest values of the covariance matrix $\mathbf{C} = \sum_{i=1}^T \mathbf{x}_i \mathbf{x}_i^\top$. The complementary space is the $m = n - k$ dimensional subspace formed by the m eigenvectors of m largest eigenvalues of $-\mathbf{C}$. Hence, the final parameter \mathbf{W}_{T+1} of the on-line algorithm should have the same eigenvectors as $-\mathbf{C}$, as well as the order of their corresponding eigenvalues. The descent step of (2.5) accumulates the scaled negated instance matrices $\mathbf{X}_t = \mathbf{x}_t \mathbf{x}_t^\top$, i.e. $\widehat{\mathbf{W}}_{t+1} = \mathbf{W}_t - \eta \mathbf{X}_t$. In the projection step of (2.5), the parameter matrix $\widehat{\mathbf{W}}_{t+1}$ is projected back to the parameter set \mathcal{W}_m by enforcing an equality constraint $\text{tr}(\mathbf{W}_{t+1}) = m$ and inequality constraints that keep all the eigenvalues of \mathbf{W}_{t+1} are in the range $[0, 1]$. The equality constraint on $\widehat{\mathbf{W}}_{t+1}$ results in adding to $\widehat{\mathbf{W}}_{t+1}$ a scaled version of the identity matrix \mathbf{I} (See Appendix C). These iterated shifts do not affect either the eigenvectors or the order of their corresponding eigenvalues. However, when the inequality constraints are enforced, then at trial t the eigenvalues of $\widehat{\mathbf{W}}_{t+1}$ that are larger than 1 or less than 0 are capped at 1 and 0, respectively. Performing such a non-uniform capping of $\widehat{\mathbf{W}}_{t+1}$ ’s eigenvalues in each trial will result in a final parameter \mathbf{W}_{T+1} with an eigensystem that is typically different from $-\mathbf{C}$. Therefore the PCA solution extracted from \mathbf{W}_{T+1} and the covariance matrix \mathbf{C} will not be the same.

There is another version of the GD algorithm that does not “forget”: The Follow the Regularized Leader GD (FRL-GD) algorithm (see, e.g., Shalev-Schwartz and Singer (2007)?)

2. This algorithm is also called as the Incremental Off-line Algorithm in (Azoury and Warmuth, 2001).

trades off the total loss on all data points against the Frobenius norm of the parameter matrix:

$$\widehat{\mathbf{W}}_{t+1} = \operatorname{argmin} \left(\|\mathbf{W}\|_F^2 + \eta \sum_{q=1}^t \operatorname{tr}(\mathbf{W}\mathbf{X}_q) \right) = -\eta \sum_{q=1}^t \mathbf{X}_q, \quad (4.2)$$

Projection step:

$$\mathbf{W}_{t+1} = \operatorname{argmin}_{\mathbf{W} \in \mathbf{W}_m} \|\mathbf{W} - \widehat{\mathbf{W}}_{t+1}\|_F^2 = \operatorname{argmin}_{\substack{\text{Eigenvalues of } \mathbf{W} \text{ in} \\ [0, 1] \text{ and } \operatorname{tr}(\mathbf{W}) = m}} \|\mathbf{W} - \widehat{\mathbf{W}}_{t+1}\|_F^2.$$

Note that in each trial, the update (4.2) projects a parameter $\widehat{\mathbf{W}}_{t+1}$ that accumulates all the past scaled negated instance matrices $(-\eta\mathbf{X}_t)$ back to trial one. In contrast, the Mirror Descent update in (2.5) performs projection iteratively, i.e. it projects parameter matrices of previous trials that are projections themselves. Therefore, the FRL-GD algorithm circumvents the forgetting issue introduced by iterative projections with respect to inequality constraints. In fact the final parameter \mathbf{W}_{T+1} of the FRL-GD is the projection of the scaled negated covariance matrix $\widehat{\mathbf{W}}_{T+1} = -\eta \sum_{t=1}^T \mathbf{x}_t \mathbf{x}_t^\top = -\eta \mathbf{C}$. We will show essentially in Appendix E that a single projection operation does not change the set of eigenvectors belonging to the m largest eigenvalues. This means that the eigenvectors belonging to the k smallest eigenvalues of \mathbf{W}_{T+1} agree with the eigenvectors of \mathbf{C} belonging to the k largest eigenvalues of \mathbf{C} .

Encouraged by this observation, we initially conjectured that the FRL-GD is strictly better than the commonly studied Mirror Descent version. More concretely, we conjectured that the FRL-GD has the optimal loss budget dependent regret bound for vanilla online PCA (as Mirror Descent MEG does which enforces the non-negativity constraints with its divergence). Unfortunately, we are able to show the opposite: The $\Omega(\max\{\min\{B_L, k\sqrt{B_L}\}, k\})$ lower bound we showed for (Mirror Descent) GD in Theorem 4.1 also holds for FRL-GD. To be precise, we have the following theorem and corollary:

Theorem 4.3 *Consider the $m = n - k$ set problem with $k \leq n/2$ and unit bit vectors as loss vectors. Then for any fixed learning rate $\eta \geq 0$, the vector version of the FRL-GD algorithm (4.2) can be forced to have regret $\Omega(\max\{\min\{B_L, k\sqrt{B_L}\}, k\})$.*

The proof is given in Appendix D. Theorem 4.3 immediately gives the lower bound on the regret of FRL-GD algorithm for the online PCA:

Corollary 4.4 *Consider the PCA problem with $k \leq n/2$ and L_1 -bounded instance matrices. Then for any fixed learning rate $\eta \geq 0$, the FRL-GD algorithm (4.2) can be forced to have regret $\Omega(\max\{\min\{B_L, k\sqrt{B_L}\}, k\})$.*

This shows that the worst case regret of the FRL-GD algorithm is the same as that of (Mirror Descent) GD, and hence suboptimal.

5. General Lower Bounds and Optimal Algorithms

In the previous section, we presented lower bounds on the regret of the GD algorithms. In this section we present lower bounds on the regret of *any* algorithm that solves the online PCA problem with L_1 and L_∞ -bounded instance matrices. More importantly, these lower bounds match all our upper bounds on the regret of the MEG algorithms within a constant factor (See bold entries in Table 3.2 and Table 4.1). To be precise, we will prove in this section a series of regret lower bounds that match our loss budget dependent upper bound (3.3) on the regret of Loss MEG, and our time dependent upper bounds (Theorem 3.1) on the regret of Loss MEG and Gain MEG, respectively. For the time dependent bounds, our lower bounds will match the lower of the two MEG bounds in each of the four sub-cases of the problem, i.e. L_1 and L_∞ -bounded instance matrices versus $k \leq \frac{n}{2}$ or $k \geq \frac{n}{2}$ (See Table 3.2 for a summary). Note that in one case the GD algorithm is also optimal: time dependent regret bounds for PCA with L_1 -bound instances when $k \leq \frac{n}{2}$.

We begin with an overview of our proof techniques for proving lower bounds that hold for any algorithm. When proving *upper bounds* on the regret (in Section 3), we first proved upper bounds as a function of the loss budget B_L and then converted them into time dependent upper bounds. For *lower* bounds on the regret, the order is reversed: we first will show time dependent lower bounds and then convert them into loss budget dependent lower bounds. As discussed in Section 4, it suffices to prove lower bounds for the m -set problem, which is the hard special case when all instances are diagonal.

Let \mathcal{A} be the set of all online algorithms for the m -set problem. Such algorithms maintain a weight vector in \mathcal{S}_m (consisting of all vectors in $[0, 1]^m$ of total weight m). For an algorithm $A \in \mathcal{A}$, we denote its regret by $\mathcal{R}(A, \ell_1, \dots, \ell_T)$ where ℓ_1, \dots, ℓ_T is a sequence of T loss vectors. The loss vectors ℓ_t lie in a constraint set \mathcal{L} . The constraint set \mathcal{L} either consists of all n dimensional unit bit vectors (the restriction of the L_1 -bounded case we use in the lower bounds), or $\mathcal{L} = \{0, 1\}^n$ (the restriction used for the L_∞ -bounded case). We use the standard method of lower bounding the regret for worst case loss sequences from \mathcal{L} by the expected regret when the loss vectors are generated i.i.d. with respect to a distribution \mathcal{P} on \mathcal{L} :

$$\min_{\text{alg. } A \in \mathcal{A}} \left\{ \max_{\text{over loss vectors } \ell_1, \dots, \ell_T \in \mathcal{L}} \mathcal{R}(A, \ell_1, \dots, \ell_T) \right\} \geq \min_{\text{over any alg. } A \in \mathcal{A}} \left\{ \mathbb{E}_{\ell_1, \dots, \ell_T \sim \mathcal{P}^T} [\mathcal{R}(A, \ell_1, \dots, \ell_T)] \right\}.$$

Each lower bound is proved as follows: Choose a distribution \mathcal{P} on \mathcal{L} , and then show a lower bound on the expected regret of any algorithm $A \in \mathcal{A}$. Note that this expectation becomes the expected loss of A minus the expected loss of the best comparator (i.e. the best m -set). We first prove time dependent regret lower bounds with L_1 and L_∞ -bounded instance vectors in sections 5.1 and 5.2, respectively. Finally we convert these lower bounds into loss budget dependent lower bounds (in Section 5.3).

5.1 Time Dependent Lower Bounds for Online PCA

Recall that $m = n - k$. First, we give a lower bound on the regret of any algorithm for the m -set problem, when $k \leq \frac{n}{2}$:

Theorem 5.1 *Consider the m -set problem with unit bit vectors as loss vectors. Then for $k \leq \frac{n}{2}$ and $T \geq k$, any online algorithm suffers worst case regret at least $\Omega(\sqrt{Tk})$.*

The proof is given in Appendix F. We lower bound the expected loss w.r.t. the distribution \mathcal{P} which is uniform on the first $2k$ unit bit vectors. Note that Theorem 5.1 requires the condition $T \geq k$. For the case $T < k$, there is a lower bounds of $\Omega(T)$ (See Theorem G.1 in Appendix G). When the loss vectors are bit vectors, then any algorithm has loss (and regret) $O(T)$. Therefore when $T < k$, any algorithm achieves the minimax regret up to a constant factor.

We now consider the uncommon case when $k \geq \frac{n}{2}$:

Theorem 5.2 *Consider the m -set problem with unit bit vectors as loss vectors. Then for $k \geq \frac{n}{2}$ and $T \geq n \log_2(n/m)$, any online algorithm suffers worst case regret of at least $\Omega(n\sqrt{\frac{T}{n} \ln \frac{n}{m}})$.*

We now set \mathcal{P} to the uniform distribution on all n unit bit vectors (See Appendix F). The small T case (here $T < n \log_2(n/m)$) is slightly more involved. There is a lower bound of $\Omega(\frac{m}{n}T)$ regret for any algorithm (see Theorem G.3 in Appendix G). Also the algorithm which predicts with the uniform weight $\frac{n}{m}$ on all experts achieves the matching regret of $O(\frac{m}{n}T)$.

Recall that the m -set problem with unit bit vectors as loss vectors is a special case of the online PCA problem with L_1 -bounded instance matrices. Combining the above two lower bounds for different ranges of k with our upper bound (Theorem 3.1) on the regret of Loss MEG for online PCA with L_1 -bounded instances gives the following corollary:

Corollary 5.3 *Consider the problem of online PCA with L_1 -bounded instance matrices.*

Then for $T \geq n \log_2(n/m)$, the $\Theta(m\sqrt{\frac{T}{n} \ln \frac{n}{m}})$ regret of Loss MEG is within a constant factor of the minimax regret.

Note that we do not use the condition $T \geq k$ of Theorem 5.1, since when $k \leq \frac{n}{2}$, $k = \Theta(n \log_2(n/m))$.

5.2 Time Dependent Lower Bound for the Generalization with L_∞ -Bounded Instance Matrices

We first give the time dependent lower bound for the m -set problem with bit vectors.

Theorem 5.4 *Consider the m -set problem with bit vectors as loss vectors. Then for $T \geq \log_2 \frac{n}{\min\{k,m\}}$, any online algorithm suffers worst case regret of at least*

$$\Omega(k\sqrt{T \ln \frac{n}{k}}) \text{ when } k \leq \frac{n}{2} \quad \text{or} \quad \Omega(m\sqrt{T \ln \frac{n}{m}}) \text{ when } k \geq \frac{n}{2}.$$

The proof is given in Appendix F. The distribution \mathcal{P} is such that each expert incurs a unit of loss with probability $1/2$ independently from the other experts. For the small T case ($T < \log_2 \frac{n}{\min\{k,m\}}$), there is a lower bound of $\Omega(\min\{Tm, Tk\})$ (See Theorem G.4 and Theorem G.5 in Appendix G). A matching upper bound of $O(\min\{Tm, Tk\})$ on the

regret of any algorithm can be reasoned as follows: At each trial, the algorithm plays with $\mathbf{W}_t \in \mathcal{W}_m$ and suffers loss $\text{tr}(\mathbf{W}_t \mathbf{X}_t)$. Since $\text{tr}(\mathbf{W}_t) = m$, the algorithm suffers loss at most m per trial and for T trials, and the cumulative loss (and thus regret) is at most Tm . The Tk upper bound can be showed similarly by considering the “gain” of the best rank k projector \mathbf{P}^* , which is $\sum_{j=1}^T \text{tr}(\mathbf{P}^* \mathbf{X}_j) \leq Tk$. Combining the lower bounds of Theorem 5.4 with the upper bounds on the regret of Loss MEG and Gain MEG when the instance matrices are L_∞ -bounded (Inequality (3.8)), results in the following corollary, which states that the Gain MEG is optimal for $k \leq \frac{n}{2}$ while the Loss MEG is optimal for $k \geq \frac{n}{2}$.

Corollary 5.5 *Consider the generalization of online PCA where the instance matrices are L_∞ -bounded.*

- When $k \leq \frac{n}{2}$ and $T \geq \log_2 \frac{n}{k}$, then the regret $\Theta(k\sqrt{T \log \frac{n}{k}})$ of Gain MEG is within a constant factor of the minimax regret.
- When $k \geq \frac{n}{2}$ and $T \geq \log_2 \frac{n}{m}$, then the regret $\Theta(m\sqrt{T \log \frac{n}{m}})$ of Loss MEG is within a constant factor of the minimax regret.

5.3 Loss Budget Dependent Lower Bounds

In this subsection, we give regret lower bounds that are functions of the loss budget B_t (defined in (3.1)). Similar to our loss budget dependent upper bound (3.3) on the regret of Loss MEG, the loss dependent lower bounds are the same for both unit and arbitrary bit vectors:

Theorem 5.6 *For the m -set problem with either unit or arbitrary bit vectors as loss vectors, any online algorithm suffers worst case regret at least $\Omega(\sqrt{B_t m \ln \frac{n}{m}}) + m \ln \frac{n}{m}$.*

The proof of the theorem is given in Appendix H. We convert the time dependent lower bounds given in Theorem 5.1 and Theorem 5.2 into loss budget dependent ones. Note that unlike our time dependent lower bounds, Theorem 5.6 is stated for the full range of the loss budget parameter B_t . The proof also distinguishes between a small and a large budget case depending on whether $B_t \leq m \ln \frac{n}{m}$. The lower bound of $\Theta(m \ln \frac{n}{m})$ follows from a conversion. However the upper bound of $O(m \ln \frac{n}{m})$ for the small budget case is non-trivial. Incidentally, this upper bound is achieved by Loss MEG.

Finally, combining this lower bound with the upper bounds (3.3) on the regret of Loss MEG, gives the following corollary, which establishes the optimality of Loss MEG no matter if the instance matrices are L_1 or L_∞ -bounded.

Corollary 5.7 *Consider the problem of online PCA with L_1 or L_∞ -bounded instance matrices. Then the regret $\Theta(\sqrt{B_t m \ln \frac{n}{m}} + m \ln \frac{n}{m})$ of Loss MEG is within a constant factor of the minimax regret.*

6. Conclusion

In this paper, we carefully studied two popular online algorithms for PCA: the Gradient Descent (GD) and Matrix Exponentiated Gradient (MEG) algorithms. For the case when the instance matrices are L_1 -bounded, we showed that both algorithms are optimal to within

a constant factor when the worst-case regret is expressed as a function of the number of trials. Furthermore, when considering regret bounds as a function of a loss budget, then MEG remains optimal and strictly outperforms GD for L_1 -bounded instances. We also studied the case when the instance matrices are L_∞ -bounded. Again we show MEG to be optimal and strictly better than GD in this case. It follows that MEG is the algorithm of choice for both cases. Note that that vanilla PCA (where the instances are outer products of vectors of length at most one) is subsumed by the case of L_1 -bounded instance matrices.

In this paper we focused on obtaining online algorithms with optimal regret and we ignored efficiency concerns. Straightforward implementations of both the GD and MEG online PCA updates required $O(n^3)$ computation per trial (because they require an eigen-decomposition of the parameter matrices). This leads to a major open problem for online PCA (Hazan et al., 2010): Is there any algorithm that can achieve optimal regret with $O(n^2)$ computation per trial. To this end, Arora et al. (2013) considers the Gain version of GD (Equation (2.3)), with the squared Euclidean distance as the divergence) where the projection enforces the additional constraint that the parameter matrix \mathbf{W}_t has rank \hat{k} . Encouraging experimental results are provided for the choice $\hat{k} = k + 1$. However, as we shall see immediately, in the most basic case when the instance matrices are outer products of unit length vectors \mathbf{x}_t that are chosen by an adversary, then any algorithm that uses parameter matrices of rank \hat{k} less than n can be forced to suffer worst case regret linear in T . Recall that the parameter matrix \mathbf{W}_t at trial t is simply the expected projection matrix of rank k chosen by the algorithm and this matrix is defined for any (deterministic or randomized) algorithm. We give an adversary argument for any algorithm for which the rank of the parameter matrix \mathbf{W}_t at any trial t is at most \hat{k} . The parameter matrices are known to the adversary. Also the initial parameter matrix \mathbf{W}_1 must have rank \hat{k} and be known to the adversary. For any algorithm following this setup the adversary argument proceeds as follows: At the beginning of the game the adversary fixes any subspace \mathcal{Q} of dimension $\hat{k} + 1$. In each trial, the adversary picks a unit length vector $\mathbf{x}_t \in \mathcal{Q}$, which is in the null space of the parameter matrix \mathbf{W}_t of the algorithm (This is always possible, because the dimension of \mathcal{Q} is larger than the rank of \mathbf{W}_t). After T trials, the algorithm has zero gain, while the total gain T is accumulated within subspace \mathcal{Q} . This means that there are k orthogonal directions within \mathcal{Q} with the total gain at least $\frac{k}{k+1}T$ and therefore, the algorithm suffers regret at least $\frac{k}{k+1}T$.

Besides restricting the rank of the parameter matrix, a second approach is to add perturbations to the current covariance matrix and then find the eigenvectors of the k -largest eigenvalues (Hazan et al., 2010). So far this approach has not led to algorithms with optimal regret bounds and $O(n^2)$ update time. Some partial results recently appeared in Garber et al. (2015) and Kotowski and Warmuth (2015).

Acknowledgments. Jiazhong Nie and Manfred K. Warmuth were supported by NSF grant IIS-1118028. Wojciech Kotowski was supported by the Polish National Science Centre grant 2013/11/D/ST6/03050 and by the Foundation for Polish Science grant Homing Plus 2012-5/5.

Appendix A. Proof of Upper Bound (3.5) on the Regret of Gain MEG

Proof The proof is based on the by now standard proof techniques of Tsuda et al. (2005). Let $\mathbf{W}_t \in \mathcal{W}_k$ be the parameter of the Gain MEG algorithm at trial t and \mathbf{X}_t be the instance matrix at this trial. Now plugging the (un-normalized) relative entropy $\Delta(\mathbf{W}, \mathbf{W}_t) = \text{tr}(\mathbf{W}(\log \mathbf{W} - \log \mathbf{W}_t) + \mathbf{W}_t - \mathbf{W})$ into the descent step of the Gain MEG algorithm (2.7) gives:

$$\widehat{\mathbf{W}}_{t+1} = \exp(\log \mathbf{W}_t + \eta \mathbf{X}_t) \quad \text{where } \eta \geq 0 \text{ is the learning rate.}$$

Take any projection matrix $\mathbf{W} \in \mathcal{W}_k$ as a comparator and use $\Delta(\mathbf{W}, \mathbf{W}_t) - \Delta(\mathbf{W}, \mathbf{W}_{t+1})$ as a measure of progress towards \mathbf{W} :

$$\begin{aligned} \Delta(\mathbf{W}, \mathbf{W}_t) - \Delta(\mathbf{W}, \mathbf{W}_{t+1}) &\geq \Delta(\mathbf{W}, \mathbf{W}_t) - \Delta(\mathbf{W}, \widehat{\mathbf{W}}_{t+1}) \\ &= \text{tr}(\mathbf{W}(\log \widehat{\mathbf{W}}_{t+1} - \log \mathbf{W}_t) + \mathbf{W}_t - \widehat{\mathbf{W}}_{t+1}) \\ &= \text{tr}(\eta \mathbf{W} \mathbf{X}_t) + \text{tr}(\mathbf{W}_t - \exp(\log \mathbf{W}_t + \eta \mathbf{X}_t)) \\ &\geq \text{tr}(\eta \mathbf{W} \mathbf{X}_t) + \text{tr}(\mathbf{W}_t - \mathbf{W}_t \exp(\eta \mathbf{X}_t)) \\ &= \text{tr}(\eta \mathbf{W} \mathbf{X}_t) + \text{tr}(\mathbf{W}_t(\mathbf{I} - \exp(\eta \mathbf{X}_t))), \end{aligned} \quad (\text{A.1})$$

where the first inequality follows from the Pythagorean Theorem and the second from the Golden-Thompson inequality: $\text{tr}(\exp(\log \mathbf{W}_t + \eta \mathbf{X}_t)) \leq \text{tr}(\mathbf{W}_t \exp(\eta \mathbf{X}_t))$. By Lemma 2.1 of Tsuda et al. (2005),

$$\text{tr}(\mathbf{W}_t(\mathbf{I} - \exp(\eta \mathbf{X}_t))) \geq (1 - e^\eta) \text{tr}(\mathbf{W}_t \mathbf{X}_t),$$

and therefore

$$\Delta(\mathbf{W}, \mathbf{W}_t) - \Delta(\mathbf{W}, \mathbf{W}_{t+1}) \geq \underbrace{\text{tr}(\mathbf{W} \mathbf{X}_t)}_{\text{gain of the comparator}} + (1 - e^\eta) \underbrace{\text{tr}(\mathbf{W}_t \mathbf{X}_t)}_{\text{gain of the algorithm}}.$$

Summing over trials gives:

$$\eta \underbrace{\sum_{t=1}^T \text{tr}(\mathbf{W} \mathbf{X}_t)}_{\text{total gain } G_{\mathbf{W}} \text{ of the comparator } \mathbf{W}} + (1 - e^\eta) \underbrace{\sum_{t=1}^T \text{tr}(\mathbf{W}_t \mathbf{X}_t)}_{\text{total gain } G_A \text{ of Gain MEG}} \leq \underbrace{\Delta(\mathbf{W}, \mathbf{W}_1)}_{\substack{\leq k \log \frac{n}{k} \\ \text{with initialization} \\ \mathbf{W}_1 = \frac{\mathbf{I}}{n}}} - \underbrace{\Delta(\mathbf{W}, \mathbf{W}_{T+1})}_{\geq 0}.$$

We now rearrange the terms to bound the regret of Gain MEG:

$$G_{\mathbf{W}} - G_A \leq \frac{1}{e^\eta - 1} k \log \frac{n}{k} + \left(1 - \frac{\eta}{e^\eta - 1}\right) G_{\mathbf{W}}. \quad (\text{A.2})$$

Since $e^\eta \geq 1 + \eta$, the coefficient $\frac{1}{e^\eta - 1}$ of the first term on the RHS is upper bounded by $\frac{1}{\eta}$. Next we upper bound the coefficient of the second term by η :

$$1 - \frac{\eta}{e^\eta - 1} = 1 - \frac{\eta e^{-\eta}}{1 - e^{-\eta}} \leq 1 - \frac{\eta e^{-\eta}}{1 - e^{-\eta}} = 1 - e^{-\eta} \leq \eta.$$

The inequality (3.5) on the regret of Gain MEG now follows from these two upper bounds, the budget inequality $G_{\mathcal{W}} \leq B_G$ and from tuning the learning rate as a function of B_G :

$$\mathcal{R}_{\text{Gain EG}} \leq \frac{k \log \frac{k}{\eta}}{\eta} + \eta B_G \stackrel{\eta = \sqrt{\frac{\log \frac{k}{B_G}}{B_G}}}{=} \sqrt{2B_G k \log \frac{k}{\eta}} \frac{n}{n}.$$

Appendix B. Proof of Upper Bound (3.6) on the Regret of GD

Proof This proof is also standard (Herbster and Warmuth, 2001). Minor alterations are needed because we have matrix parameters. Let $\mathbf{W}_t \in \mathcal{W}_m$ be the parameter of the GD algorithm at trial t and \mathbf{X}_t be the instance matrix at this trial. Then for the best comparator $\mathbf{W} \in \mathcal{W}_m$ and any learning rate $\eta \geq 0$, the following holds

$$\|\mathbf{W}_{t+1} - \mathbf{W}\|_F^2 \leq \|\widehat{\mathbf{W}}_{t+1} - \mathbf{W}\|_F^2 = \|\mathbf{W}_t - \mathbf{W}\|_F^2 - 2\eta \text{tr}((\mathbf{W}_t - \mathbf{W})\mathbf{X}_t^\top) + \eta^2 \|\mathbf{X}_t\|_F^2,$$

where the inequality follows from the Pythagorean Theorem (Herbster and Warmuth, 2001) and the equality follows from the descent step of the GD algorithm (see (2.5)). By rearranging terms, we have

$$\text{tr}(\mathbf{W}_t \mathbf{X}_t^\top) - \text{tr}(\mathbf{W} \mathbf{X}_t^\top) \leq \frac{\|\mathbf{W}_t - \mathbf{W}\|_F^2 - \|\mathbf{W}_{t+1} - \mathbf{W}\|_F^2}{2\eta} + \frac{\eta \|\mathbf{X}_t\|_F^2}{2}.$$

Note that the LHS is the regret in trial t w.r.t. \mathbf{W} . By summing all trials, we have that the (total) regret $\mathcal{R}_{GD} = \sum_{t=1}^T \text{tr}(\mathbf{W}_t \mathbf{X}_t^\top)$ is upper bounded by

$$\frac{\|\mathbf{W}_1 - \mathbf{W}\|_F^2 - \|\mathbf{W}_{T+1} - \mathbf{W}\|_F^2}{2\eta} + \frac{\eta \sum_{t=1}^T \|\mathbf{X}_t\|_F^2}{2} \leq \frac{k(n-k)}{2n\eta} + \frac{\eta \sum_{t=1}^T \|\mathbf{X}_t\|_F^2}{2}, \quad (\text{B.1})$$

where we used $\|\mathbf{W}_1 - \mathbf{W}\|_F^2 \leq \frac{k(n-k)}{n}$ since $\mathbf{W} \in \mathcal{W}_m$ and $\mathbf{W}_1 = \frac{n-k}{n} \mathbf{I}$. In the L_1 -bounded instance matrix case (when $\|\mathbf{X}_t\|_F^2 \leq 1$), (B.1) can be further simplified as

$$\mathcal{R}_{GD} \leq \frac{k(n-k)}{2n\eta} + \frac{\eta T}{2}.$$

By setting $\eta = \frac{k(n-k)}{n\sqrt{T}}$, we obtain the $\sqrt{\frac{k(n-k)}{n}T}$ regret bound for the L_1 -bounded instance case. When the instance matrices are L_∞ -bounded, then $\|\mathbf{X}_t\|_F^2 \leq n$ and hence, $\mathcal{R}_{GD} \leq \sqrt{k(n-k)T}$ with $\eta = \frac{k(n-k)}{\sqrt{T}}$. \blacksquare

Appendix C. Proof of Theorem 4.1

Theorem 4.1 gives a lower bound on the regret of the GD algorithm for the m -set problem with unit bit vectors as loss vectors. At each trial of the m -set problem, the online algorithm

first predicts with a weight vector $\mathbf{w}_t \in [0, 1]^n$, the coordinates of which sum to m . Then the algorithm receives a unit bit vector ℓ_t and suffers loss $\mathbf{w}_t \cdot \ell_t$. The GD algorithm for online PCA (2.5) specializes to the following updates of the parameter vector for learning m -sets:

$$\begin{aligned} \text{Descent step:} \quad & \hat{\mathbf{w}}_{t+1} = \mathbf{w}_t - \eta \ell_t, \\ \text{Projection step:} \quad & \mathbf{w}_{t+1} = \text{argmin}_{\mathbf{w} \in \mathcal{S}_m} \|\mathbf{w} - \hat{\mathbf{w}}_{t+1}\|^2, \end{aligned} \quad (\text{C.1})$$

where $\eta > 0$ is the learning rate and $\mathcal{S}_m = \{\mathbf{w} \in [0, 1]^n : \sum_{i=1}^n w_i = m\}$.

Since our lower bound for GD must hold no matter what the *fixed* learning rate η is, we construct two adversarial loss sequences: The first causes the GD algorithm to suffer large regret when η is small and the second causes large regret when η is large. Specifically, we will show that the GD algorithm suffers regret at least $\Omega(k/\eta)$ on the first sequence, and at least $\Omega(\min\{B_T, kB_T\eta\})$ on the second sequence. We will then show that the lower bound of the theorem follows by taking the maximum of these two bounds and by solving for the learning rate that minimizes this maximum. The first sequence consists of unit losses assigned to the first k experts. At each trial, the adversary gives a unit of loss to the expert (out of the first k) with the largest current weight. If the learning rate η is small, then the weights assigned to the first k experts decrease too slowly (Lemma C.2). This causes the algorithm to suffer a substantial amount of loss on the first sequence, while the loss of the remaining m experts remains zero. The second sequence consists of unit losses assigned to the first $k+1$ experts. As before, the adversary always gives the expert with the largest weight (now out of the first $k+1$) a unit of loss. Intuitively, the GD algorithm will give high weight to the $m-1 = n-(k+1)$ loss free experts and the best out of the first $k+1$ experts. As the η gets larger, the algorithm puts more and more weight on the *current* best out of the $k+1$ experts instead of hedging its bets over all $k+1$ experts. So the algorithm becomes more and more deterministic and the adversary strategy of hitting the expert with the largest weight (out of the first $k+1$) causes the algorithm to suffer a substantial loss (Lemma C.3). Formalizing these findings is not simple as the projection step of the GD algorithm does not have a closed form. Hence, we need to resort to the Karush-Kuhn-Tucker optimality conditions and prove a sequence of lemmas before assembling all the pieces for proving Theorem 4.1.

Let α_i be a dual variable for the constraint $w_{t+1,i} \geq 0$ ($i = 1, \dots, n$), β_i be a dual variable for the constraint $w_{t+1,i} \leq 1$ ($i = 1, \dots, n$), and γ be a dual variable for the constraint $\sum_{i=1}^n w_{t+1,i} = m$. Then the KKT conditions on the projection step of (C.1) have the following form: For $i = 1, \dots, n$,

$$\begin{aligned} \text{Stationarity:} \quad & w_{t+1,i} = -w_{t,i} - \eta \ell_{t,i} + \gamma + \alpha_i - \beta_i, \\ \text{Complementary slackness:} \quad & w_{t+1,i} \alpha_i = 0, \quad (w_{t+1,i} - 1) \beta_i = 0, \\ \text{Primal feasibility:} \quad & \sum_{i=1}^n w_{t+1,i} = m, \quad 0 \leq w_{t+1,i} \leq 1, \\ \text{Dual feasibility:} \quad & \alpha_i \geq 0, \quad \beta_i \geq 0. \end{aligned} \quad (\text{C.2})$$

Note that since the projection step of (C.1) is a convex optimization problem, these conditions are necessary and sufficient for the optimality of a solution. Hence, for any intermediate weight vector $\hat{\mathbf{w}}_{t+1} = \mathbf{w}_t - \eta \ell_t$, if a set of primal and dual variables $\mathbf{w}_{t+1}, \boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n), \boldsymbol{\beta} = (\beta_1, \dots, \beta_n), \gamma$ satisfy all the conditions (C.2), then they are the unique primal and dual solutions of the projection step.

We start with a special case where the GD update (C.1) actually has a closed form solution:

Lemma C.1 *Consider a trial of the m -set problem with n experts, when only one expert incurs a unit of loss. If this expert has weight w and all remaining experts have weight at most $1 - \min\{\frac{\eta}{n}, \frac{w}{n-1}\}$, then the GD algorithm with learning rate $\eta > 0$ will decrease w by $\min\{\frac{(n-1)\eta}{n}, w\}$ and increase all the other weights by $\min\{\frac{\eta}{n}, \frac{w}{n-1}\}$.*

Proof W.l.o.g., the first expert incurs a unit of loss in trial t , i.e. $w_{t,1} = w$ and $\ell_t = \mathbf{e}_1$, where \mathbf{e}_1 is the unit bit vector with first coordinate equal to 1 (and all other coordinates equal to 0). To solve the projection step of the GD update (C.1), we distinguish two cases based on the value of $w_{t,1}$. In each case we propose a solution to the projection step and show that it is a valid solution by verifying the KKT conditions (C.2).

Case $w_{t,1} = w \geq \frac{n-1}{n}\eta$: The proposed solution is $\gamma = \frac{\eta}{n}$ and for $1 \leq i \leq n$, $\alpha_i = \beta_i = 0$,

$$w_{t+1,i} = \begin{cases} w_{t,1} - \frac{n-1}{n}\eta & \text{for } i = 1 \\ w_{t,i} + \frac{\eta}{n} & \text{for } i \geq 2 \end{cases}.$$

All KKT conditions are easy to check, except for the primal feasibility condition: $w_{t+1,i} \leq 1$, for $i \geq 2$. By the assumption of the lemma, $w_{t,i} \leq 1 - \min\{\frac{\eta}{n}, \frac{w_{t,1}}{n-1}\}$. Since we are in the case $w_{t,1} \geq \frac{n-1}{n}\eta$, we have $w_{t,i} \leq 1 - \frac{\eta}{n}$ and therefore

$$w_{t+1,i} = w_{t,i} + \frac{\eta}{n} \leq 1 - \frac{\eta}{n} + \frac{\eta}{n} = 1.$$

We conclude that in this case, the first weight decreases by $\frac{n-1}{n}\eta$ and all the other weights increase by $\frac{\eta}{n}$.

Case $w_{t,1} = w < \frac{n-1}{n}\eta$: The proposed solution is $\gamma = \frac{w_{t,1}}{n-1}$ and for $1 \leq i \leq n$, $\beta_i = 0$,

$$\alpha_i = \begin{cases} \eta - \frac{\eta}{n-1}w_{t,1} & \text{for } i = 1 \\ 0 & \text{for } i \geq 2 \end{cases}, \quad w_{t+1,i} = \begin{cases} 0 & \text{for } i = 1 \\ w_{t,i} + \frac{w_{t,1}}{n-1} & \text{for } i \geq 2 \end{cases}.$$

Again, all KKT conditions are easy to check, except for the primal feasibility condition $w_{t+1,i} \leq 1$ for $i \geq 2$. By the assumption of the lemma $w_{t,i} \leq 1 - \min\{\frac{\eta}{n}, \frac{w_{t,1}}{n-1}\}$. Since we are in the case $w_{t,1} < \frac{n-1}{n}\eta$, we have $w_{t,i} \leq 1 - \frac{w_{t,1}}{n-1}$ and therefore

$$w_{t+1,i} = w_{t,i} + \frac{w_{t,1}}{n-1} \leq 1 - \frac{w_{t,1}}{n-1} + \frac{w_{t,1}}{n-1} = 1.$$

We conclude that in this case, the first weight decreases by $w_{t,1}$ and all the other weights increase by $\frac{w_{t,1}}{n-1}$. Combining the above two cases proves the lemma. \blacksquare

Our next lemma considers the general case when the weight vector before the update does not necessarily satisfy the assumption in Lemma C.1, i.e. the weights of the experts not incurring loss may be larger than $1 - \min\{\frac{\eta}{n}, \frac{w}{n-1}\}$ (where w is the weight of the only expert incurring loss).

Lemma C.2 *Consider a trial of the m -set problem with n experts, when only one expert incurs a unit of loss. If this expert has weight w , then the GD algorithm with learning rate $\eta > 0$ will decrease w by at most η and will not decrease the weights of any other experts. Furthermore, if any expert not incurring loss has weight at least $1 - \min\{\frac{\eta}{n}, \frac{w}{n-1}\}$, then its weight will be set to 1 by the capping constraint.*

Proof Let w_t be the weight vector at the beginning of the trial and assume w.l.o.g. that the first expert incurs one unit of loss, i.e. $\ell_t = \mathbf{e}_1$. Let w_{t+1} , α , β and γ denote the variables satisfying the KKT conditions (C.2). The lemma now states that:

$$w_{t+1,1} \geq w_{t,1} - \eta \quad \text{and} \quad w_{t+1,i} \geq w_{t,i}, \quad \text{for } 2 \leq i \leq n, \quad (\text{C.3})$$

and furthermore

$$w_{t+1,i} = 1, \quad \text{for any } 2 \leq i \leq n \text{ such that } w_{t,i} \geq 1 - \min\left\{\frac{\eta}{n}, \frac{w_{t,1}}{n-1}\right\}. \quad (\text{C.4})$$

We first prove (C.3). By the stationarity condition of (C.2) and the assumption $\ell_t = \mathbf{e}_1$, we have that

$$w_{t+1,1} - w_{t,1} = \cancel{w_{t,1}} - \eta + \alpha_1 - \beta_1 + \gamma - \cancel{w_{t,1}} = -\eta + \alpha_1 - \beta_1 + \gamma,$$

$$\text{and for } 2 \leq i \leq n: \quad w_{t+1,i} - w_{t,i} = \cancel{w_{t,i}} + \alpha_i - \beta_i + \gamma - \cancel{w_{t,i}} = \alpha_i - \beta_i + \gamma.$$

Therefore, to prove (C.3), it suffices to show $\alpha_i - \beta_i + \gamma \geq 0$ for $1 \leq i \leq n$. By the dual feasibility condition of (C.2), $\alpha_i \geq 0$ but $-\beta_i \leq 0$. However, when $-\beta_i < 0$, we have $w_{t+1,i} = 1$ by the complementary slackness condition, and therefore (C.3) holds trivially in this case (noting that $w_{t,i} \leq 1$). Now we only need to show $\gamma \geq 0$. We do this by summing $w_{t,i} - \eta\ell_{t,i} + \gamma$ over indices i such that $w_{t+1,i} > 0$:

$$\begin{aligned} \sum_{i:1 \leq i \leq n, w_{t+1,i} > 0} (w_{t,i} - \eta\ell_{t,i} + \gamma) & \stackrel{\alpha_i = 0 \text{ since } w_{t+1,i} > 0}{=} \sum_{i:1 \leq i \leq n, w_{t+1,i} > 0} (w_{t,i} - \eta\ell_{t,i} + \gamma + \alpha_i) \\ & \geq \sum_{i:1 \leq i \leq n, w_{t+1,i} > 0} (w_{t,i} - \eta\ell_{t,i} + \gamma + \alpha_i - \beta_i) \\ & \geq \sum_{i:1 \leq i \leq n, w_{t+1,i} > 0} (w_{t,i}) = m. \end{aligned} \quad (\text{C.5})$$

Furthermore, since both the learning rate η and the loss vector ℓ_t are non-negative, we have that for all $1 \leq i \leq n$,

$$\sum_{i:1 \leq i \leq n, w_{t+1,i} > 0} (w_{t,i} - \eta\ell_{t,i}) \leq \sum_{i:1 \leq i \leq n, w_{t+1,i} > 0} (w_{t,i}) \leq m.$$

Combining the above inequality with (C.5) implies that $\gamma \geq 0$, which completes our proof of (C.3).

Next we prove (C.4). By the stationarity condition of (C.2) and the assumption $\ell_t = \mathbf{e}_1$, we have that for $2 \leq i \leq n$,

$$w_{t+1,i} = w_{t,i} - \eta\ell_{t,i} + \alpha_i - \beta_i + \gamma = w_{t,i} + \alpha_i - \beta_i + \gamma. \quad (\text{C.6})$$

Now if we further assume that $w_{t,i} \geq 1 - \min\{\frac{\eta}{n}, \frac{w_{t,1}}{n-1}\}$, then (C.6) is lower bounded by

$$w_{t+1,i} = w_{t,i} + \alpha_i - \beta_i + \gamma \geq 1 - \min\left\{\frac{\eta}{n}, \frac{w_{t,1}}{n-1}\right\} + \alpha_i - \beta_i + \gamma.$$

Thus to prove (C.4), it suffices to show that $-\min\{\frac{\eta}{n}, \frac{w_{t,1}}{n-1}\} + \alpha_i - \beta_i + \gamma \geq 0$. By the dual feasibility condition of (C.2), $\alpha_i \geq 0$ but $-\beta_i \leq 0$. However, when $-\beta_i < 0$, then $w_{t+1,i} = 1$ follows directly from the complementary slackness condition. Therefore w.l.o.g., we assume $\beta_i = 0$. Now all that remains is to show $\gamma \geq \min\{\frac{\eta}{n}, \frac{w_{t,1}}{n-1}\}$, for which we distinguish the following 2 cases.

Case $w_{t+1,1} > 0$: We will show $\gamma \geq \frac{\eta}{n}$ for this case. First note that

$$m \stackrel{(C.5)}{\leq} \sum_{i:1 \leq i \leq n, w_{t+1,i} > 0} (w_{t,i} - \eta \ell_{t,i} + \gamma) \stackrel{\gamma \geq 0}{\leq} \sum_{i:1 \leq i \leq n, w_{t+1,i} > 0} (w_{t,i} - \eta \ell_{t,i}) + n\gamma. \quad (C.7)$$

Now since we assume $w_{t+1,1} > 0$ and $\ell_t = e_1$, the first term on RHS of (C.7) is upper bounded by:

$$\sum_{i:1 \leq i \leq n, w_{t+1,i} > 0} (w_{t,i} - \eta \ell_{t,i}) = \sum_{i:1 \leq i \leq n, w_{t+1,i} > 0} (w_{t,i}) - \eta \leq m - \eta.$$

Together, we get $m \leq n\gamma - \eta + m$, and this gives $\gamma \geq \frac{\eta}{n}$.

Case $w_{t+1,1} = 0$: We will show $\gamma \geq \frac{w_{t,1}}{n-1}$ for this case. Since $w_{t+1,1} = 0$, the summation

$$\sum_{i:1 \leq i \leq n, w_{t+1,i} > 0} (w_{t,i} - \eta \ell_{t,i} + \gamma) \text{ does not include the case } i = 1, \text{ i.e.}$$

$$\sum_{i:1 \leq i \leq n, w_{t+1,i} > 0} (\hat{w}_{t,i} - \eta \ell_{t,i} + \gamma) = \sum_{i:2 \leq i \leq n, w_{t+1,i} > 0} (\hat{w}_{t,i} - \eta \ell_{t,i} + \gamma).$$

Therefore, (C.7) can be tightened as follows:

$$m \stackrel{(C.5)}{\leq} \sum_{i:2 \leq i \leq n, w_{t+1,i} > 0} (w_{t,i} - \eta \ell_{t,i} + \gamma) \stackrel{\gamma \geq 0}{\leq} \sum_{i:1 \leq i \leq n, w_{t+1,i} > 0} (w_{t,i} - \eta \ell_{t,i}) + (n-1)\gamma.$$

Again, by the assumption $\ell_t = e_1$, we have

$$\sum_{i:2 \leq i \leq n, w_{t+1,i} > 0} (w_{t,i} - \eta \ell_{t,i}) = \sum_{i:2 \leq i \leq n, w_{t+1,i} > 0} (w_{t,i}) \leq m - w_{t,1}.$$

Together, we get $m \leq (n-1)\gamma + m - w_{t,1}$, which gives $\gamma \geq \frac{w_{t,1}}{n-1}$ and completes the proof. ■

Our third lemma lower bounds the loss of the GD algorithm with respect to a particular adversarial loss sequence of n trials (instead of the above lower bounds for single trials). We argue this lower bound for the special case of the m -set problem when $m = 1$, i.e. the vanilla expert setting. As we shall see shortly in the main proof of Theorem 4.1, the lower bound of the general m -set problem degenerates into this special case for a certain loss sequence. Note that the assumptions of Lemma C.1 are always met when $m = 1$, because in this case any expert not incurring loss has weight at most $1 - w$, where w is the weight of the expert incurring loss.

Lemma C.3 *Consider the m -set problem with n experts, and $m = 1$. If at each trial, only the expert with the largest weight incurs a unit of loss, then after n consecutive such trials, the GD algorithm with learning rate $\eta > 0$ suffers loss at least $1 + \frac{1}{32} \min\{\eta, 1\}$.*

Proof First notice that when $m = 1$, the largest of the n expert weights at each trial is at least $\frac{1}{n}$. Therefore, any algorithm suffers total loss at least 1 in n trials. To show the extra loss of $\frac{1}{32} \min\{\eta, 1\}$, we claim that in at least $\frac{n}{4}$ of these n trials, the largest expert weight assigned by the GD algorithm is at least $\frac{1}{n} + \frac{1}{8} \min\{\eta, \frac{1}{n}\}$. This claim is proved as follows.

Let $\eta' = \min\{\eta, \frac{1}{n}\}$ and t_0 be the first trial that the largest expert weight of the trial is less than $\frac{1}{n} + \frac{1}{8}\eta'$. If $t_0 > \frac{n}{4}$, the claim holds trivially. Hence, we assume $t_0 \leq \frac{n}{4}$. Now call any expert with weight at least $\frac{1}{n} - \frac{1}{8}\eta'$ at trial t_0 a *candidate*. We will show that the number of candidates s is at least $\frac{n}{4}$. To show this we first upper bound the expert weights at trial t_0 as follows:

$$\begin{aligned} \text{sum of non-candidates' weights} &\leq (n-s) \left(\frac{1}{n} - \frac{1}{8}\eta'\right), \\ \text{sum of candidates' weights} &\leq s \left(\frac{1}{n} + \frac{1}{8}\eta'\right). \end{aligned}$$

The first inequality follows from the fact that non-candidates have weight at most $\frac{1}{n} - \frac{1}{8}\eta'$ and the second inequality follows from the definition of t_0 , i.e. the maximum weight at that trial is less than $\frac{1}{n} + \frac{1}{8}\eta'$. Now, since all the expert weights at a trial sum to 1, we have

$$1 \leq s \left(\frac{1}{n} + \frac{1}{8}\eta'\right) + (n-s) \left(\frac{1}{n} - \frac{1}{8}\eta'\right) = 1 + \frac{s}{4} \eta' - \frac{n}{8} \eta',$$

which gives $s \geq \frac{n}{2}$ since $\eta' \geq \eta > 0$.

Next, we show that at trial $t_0 + \frac{n}{4}$, there will be a subset of at least $\frac{n}{4}$ candidates whose weight will be at least $\frac{1}{n} + \frac{1}{8}\eta'$. First recall that at each trial, only one expert incurs a unit of loss. Therefore, in the $\frac{n}{4}$ trials from t_0 to $t_0 + \frac{n}{4} - 1$, there will be at least $\frac{n}{2} - \frac{n}{4} = \frac{n}{4}$ candidates that do not incur any loss. By Lemma C.1, the weight of an expert not incurring loss is increased at each trial by $\min\{\frac{\eta}{n}, \frac{w_{t,i}}{n-1}\}$, where w is the weight of the expert not incurring loss at that trial. Note that $w \geq \frac{1}{n}$ always hold since the expert incurring loss has the largest weight among the n experts. Therefore, at trial $t_0 + \frac{n}{4}$, each of the $\frac{n}{4}$ candidates that do not incur any loss from trial t_0 to trial $t_0 + \frac{n}{4} - 1$ has weight at least:

$$\underbrace{\frac{1}{n} - \frac{1}{8}\eta'}_{\text{lower bound on the weight at trial } t_0} + \underbrace{\frac{n}{4} \min\left\{\frac{\eta}{n}, \frac{w_t}{n-1}\right\}}_{\text{lower bound on the increase from trial } t_0 \text{ to trial } t_0 + \frac{n}{4} - 1} \stackrel{w_t \geq \frac{1}{n}}{\geq} \frac{1}{n} - \frac{1}{8}\eta' + \frac{n}{4} \min\left\{\frac{\eta}{n}, \frac{1}{n^2}\right\} = \frac{1}{n} + \frac{\eta'}{8}.$$

Finally, consider the next $\frac{n}{4}$ trials from $t_0 + \frac{n}{4}$ to $t_0 + \frac{n}{2} - 1$. (The game must have more than $t_0 + \frac{n}{2}$ trials, since we assume $t_0 \leq \frac{n}{4}$.) The maximum weights at these trials are always at least $\frac{1}{n} + \frac{1}{8}\eta'$, because only one expert incurs loss at a time, and the weights of the remaining experts are never decreased. This completes the proof of the claim and the lemma. ■

Now we are ready to give the lower bound on the regret of the GD algorithm for the m -set problem. For the sake of readability, we repeat the statement of Theorem 4.1 below:

Theorem 4.1 *Consider the m -set problem with $k \leq n/2$ and unit bit vectors as loss vectors. Then for any fixed learning rate η , the GD algorithm (C.1) can be forced to have regret at least $\Omega(\max\{\min\{B_L, k\sqrt{B_L}\}, k\})$.*

Proof The lower bound $\Omega(k)$ directly follows from Lemma G.2 proven later: If we set the variable i in the statement of the lemma to k , then this results in a lower bound of $\Omega(m \log \frac{m}{k})$ for any algorithm. Now, $m \log \frac{m}{k} = m \log(\frac{k}{m} + 1) \geq k$. Hence to prove this theorem, we only need to show a lower bound of $\Omega(\min\{B_L, k\sqrt{B_L}\})$, where B_L is the loss budget (defined in (3.1)). Also, w.l.o.g., assume $B_L \geq 4k$ since when $B_L \leq 4k$, the claimed bound is in fact $\Omega(k)$, which always holds as we just argued.

The hard part (deferred to later) in proving the $\Omega(\min\{B_L, k\sqrt{B_L}\})$ lower bound for GD is to show that the algorithm suffers regret at least $\Omega(k/\eta)$ and $\Omega(\min\{B_L, kB_L\eta\})$ on two different loss sequences, respectively. Clearly, it follows that the regret of GD is then at least the maximum of these two bounds. By a case analysis, one can show that $\max\{a, \min\{b, c\}\} \geq \min\{b, c\}$ for any $a, b, c \in \mathbb{R}$. (We prove this as Lemma I.1 in Appendix I.) Therefore we get the lower bound of $\Omega(\min\{B_L, \max\{k/\eta, kB_L\eta\}\})$. The lower bound for GD with any fixed learning rate now follows from fact that $\max\{k/\eta, kB_L\eta\}$ is minimized at $\eta = \Theta(1/\sqrt{B_L})$. The value of the lower bound with this choice of η is the target lower bound of $\Omega(k\sqrt{B_L})$.

We still need to describe the two loss sequences and prove the claimed lower bounds on the regret. The first loss sequence forces GD to suffer regret $\Omega(k/\eta)$. It consists of $\lfloor \frac{km}{m\eta} \rfloor + 1$ trials in which only the first k experts incur losses. More precisely, at each trial, the expert with the largest weight (within the first k experts) incurs one unit of loss (In the case of tied weights, only the expert with the smallest index incurs loss). The last m experts have loss 0. Therefore the regret is simply the total loss of the GD algorithm. The loss of the algorithm at each trial is equal to the largest weight of the first k experts. Therefore the loss is lower bounded by the average of the first k weights. With a uniform initial weight vector, this average is $\frac{m}{n}$ at the beginning of the first trial, and by Lemma C.2, it is decreased by at most $\frac{\eta}{k}$ after each of the following $\lfloor \frac{km}{m\eta} \rfloor + 1$ trials. Therefore, at the beginning of trial t , the average is at least $\frac{m}{n} - (t-1)\frac{\eta}{k}$. Summing up the arithmetic series from trial 1 to trial $\lfloor \frac{km}{m\eta} \rfloor + 1$ gives the following lower bound on the total loss of GD:

$$\frac{1}{2} \left(\left\lfloor \frac{km}{m\eta} \right\rfloor + 1 \right) \left(\frac{m}{n} - \left(\left\lfloor \frac{km}{m\eta} \right\rfloor + 1 - 1 \right) \frac{\eta}{k} \right) \stackrel{\frac{m}{n} \geq \frac{1}{2}}{\geq} \frac{1}{4} \left(\left\lfloor \frac{k}{2\eta} \right\rfloor + 1 \right) = \Omega\left(\frac{k}{\eta}\right).$$

Now we describe the second loss sequence which forces the GD algorithm to suffer regret $\Omega(\min\{B_L, kB_L\eta\})$. For the sake of clarity, we assume that B_L is integer (otherwise replace B_L by $\lfloor B_L \rfloor$ in the proof). The sequence consists of $(k+1)B_L$ trials, where the expert with the largest weight among first $k+1$ experts incurs a unit of loss. The best comparator of this sequence consists of the last $m-1$ experts that have 0 total loss and the best of the first $k+1$ experts which has total loss at most B_L .

Next we lower bound the loss of GD with respect to this loss sequence. First observe, that the last $m-1$ experts do not incur any loss in the $(k+1)B_L$ trials. Therefore their weight may increase (from their initial value of $\frac{m}{n}$), but at any trial the weights of these

experts always have the same value. The value of this block of equal weights is always the maximum weight of any expert, since the weight value of the block is never decreased by the algorithm. More precisely, at each trial the block's value is increased as given in Lemma C.1, until it becomes 1 at trial t_{cap} and stay at 1 till the end of the game. If no such trial t_{cap} exists (i.e. the value of the block remains less than 1 at the end of the game), then let $t_{cap} = \infty$. In the degenerate case when $m = 1$ (i.e. the block has size $m-1 = 0$), we simply set $t_{cap} = 1$ from the beginning.

Depending on the value of t_{cap} , we distinguish two cases in which GD suffers loss at least $B_L + \Omega(B_L)$ and $B_L + \Omega(\min\{B_L, kB_L\eta\})$, respectively.

Case $t_{cap} > (k+1)B_L/2$: We will show that GD suffers loss at least $B_L + \Omega(B_L)$ in this case. First recall that at the beginning of the proof we assumed $B_L \geq 4k$. Therefore in the case $t_{cap} > (k+1)B_L/2$ we have $t_{cap} > 4$. From our definition of t_{cap} this means that $m \geq 2$. Next we argue that since $t_{cap} > (k+1)B_L/2$, we have $\eta \leq \frac{1}{k+1}$. Let W_t denote the sum of the first $k+1$ weights at trial t and let w_t be their maximum. By Lemma C.1, we know that in each trial prior to t_{cap} , the weight w_t of the expert incurring loss is decreased by $\min\{\frac{(n-1)\eta}{n}, w_t\}$ and all other weights are increased by $\min\{\frac{\eta}{n}, \frac{w_t}{n-1}\}$. Since the expert incurring loss is always one of the first $k+1$ experts, we have that in each trial prior to t_{cap} , the total weight W_t is decreased by at least

$$\min \left\{ \frac{(n-1)\eta}{n}, w_t \right\} - k \min \left\{ \frac{\eta}{n}, \frac{w_t}{n-1} \right\} \geq \frac{m-1}{n} \min\{\eta, w_t\} \geq \frac{m-1}{n} \min \left\{ \eta, \frac{1}{k+1} \right\}.$$

The second inequality follows from the fact that since w_t is the largest of the first $k+1$ expert weights, it must be at least $\frac{1}{k+1}$. Together with the fact that $W_1 = \frac{(k+1)m}{n}$, we have

$$W_{(k+1)B_L/2} \leq \frac{(k+1)m}{n} - \frac{(k+1)B_L m - 1}{2n} \min \left\{ \eta, \frac{1}{k+1} \right\}. \quad (\text{C.8})$$

Now if $\eta \geq \frac{1}{k+1}$, the upper bound (C.8) becomes $\frac{(k+1)m}{n} - \frac{(m-1)B_L}{2n}$, which can be further upper bounded by $\frac{m}{n}$ using the fact $m \geq 2$ and the assumption $B_L \geq 4k$. However, the upper bound of $W_{(k+1)B_L/2} \leq \frac{m}{n}$ is less than 1 and all W_t are at least 1 since $m - W_t$ is the total weight of the last $m-1$ experts which is at most $m-1$. Therefore we have $\eta < \frac{1}{k+1}$ in this case.

Now we lower bound the loss of GD by lower bounding the average weight $W_t/(k+1)$. We have $\eta < \frac{1}{k+1}$ and $t_{cap} > (k+1)B_L/2$. Also by Lemma C.1, W_t decreases by exactly $\frac{(m-1)\eta}{n}$ at each trial for $1 \leq t \leq (k+1)B_L/2$. Therefore the total average weight in trials 1 through $(k+1)B_L/2$ is at least

$$\frac{1}{2k+1} \left(\frac{(k+1)m}{n} + 1 \right) \frac{(k+1)B_L}{2} = \left(\frac{(k+1)m}{n} + 1 \right) \frac{B_L}{4}. \quad (\text{C.9})$$

Now with $m \geq 2$, $k \geq 1$ and $n = m+k$, it is easy to verify that $\frac{(k+1)m}{n}$ is at least $1 + \Omega(1)$, which along with (C.9) results in a $\frac{B_L}{4} + \Omega(B_L)$ lower bound on the loss of GD for $1 \leq t \leq (k+1)B_L/2$. In trials $(k+1)B_L/2 < t \leq (k+1)B_L$, GD suffers loss at least $\frac{(k+1)B_L}{2} = \frac{B_L}{2}$ since the weight of the expert incurring loss is at least $\frac{1}{k+1}$. Thus in trial

1 through $(k+1)B_L$ the loss of GD is at least $B_L + \Omega(B_L)$ which concludes the proof of the case $t_{\text{cap}} \geq (k+1)B_L/2$.

Case $t_{\text{cap}} \leq (k+1)B_L/2$: We will show that GD suffers total loss at least $B_L + \Omega(\min\{B_L, kB_L\eta\})$ in this case. First note that GD suffers loss at least $B_L/2$ in the first $(k+1)B_L/2$ trials. This follows from the fact that in each trial, the expert with the largest weight among first $k+1$ experts incurs a unit of loss. Since the sum of all weights is equal to m , and none of the remaining $m-1$ weights can exceed 1, the sum of weights of the first $k+1$ experts must be at least 1, and hence the largest weight among the first $k+1$ experts is at least $\frac{1}{k+1}$. This means that in a sequence of $(k+1)B_L/2$ trials, the loss of the GD algorithm is at least $B_L/2$.

Thus, it suffices to show that GD suffers loss at least $B_L/2 + \Omega(\min\{B_L, kB_L\eta\})$ in trials $(k+1)B_L/2 + 1$ through $(k+1)B_L$. First note that since $t_{\text{cap}} \leq (k+1)B_L/2$, in each of these trials the weights of the $m-1$ loss free experts have reached the cap 1. This means that GD updates the weights of the first $k+1$ experts as in the vanilla expert setting (i.e. $m=1$). Therefore by Lemma C.3, the loss of GD in the second $(k+1)B_L/2$ trials is at least $\frac{B_L}{2}(1 + \frac{1}{32} \min\{(k+1)\eta, 1\}) = \frac{B_L}{2} + \Omega(\min\{B_L, kB_L\eta\})$.

We conclude that for the second loss sequence, the loss of the best comparator is at most B_L and the loss of GD is at least $B_L + \Omega(\min\{B_L, kB_L\eta\})$. Therefore, the regret of GD is at least $\Omega(\min\{B_L, kB_L\eta\})$ for the second loss sequence and this completes our proof of the theorem. \blacksquare

Appendix D. Proof of Theorem 4.3

Theorem 4.3 gives a lower bound on the regret of the FRL-GD algorithm for the m -set problem with unit bit vectors as loss vectors. In this case, the FRL-GD algorithm (4.2) specializes to the following:

$$\begin{aligned} \text{Follow the regularized leader: } \hat{\mathbf{w}}_{t+1} &= -\eta \sum_{g=1}^t \ell_{g_t} \\ \text{Projection step: } \mathbf{w}_{t+1} &= \underset{\mathbf{w} \in \mathcal{S}_m}{\text{argmin}} \|\mathbf{w} - \hat{\mathbf{w}}_{t+1}\|^2. \end{aligned} \quad (\text{D.1})$$

The proof has the same structure as the lower bound for the GD algorithm (Appendix C). Again we use two adversarial loss sequences (one for low and high learning rates) and give three technical lemmas that reason with the KKT conditions. The details are different because the intermediate weight vector \mathbf{w}_{t+1} has a different form than for vanilla GD. The KKT conditions are the same as the KKT condition for GD (C.1) except for a slight change in the stationarity condition. For $i=1, \dots, n$,

$$\begin{aligned} \text{Stationarity:} & \quad w_{t+1,i} = -\eta \ell_{\leq t,i} + \gamma + \alpha_i - \beta_i, \\ \text{Complementary slackness: } & \quad w_{t+1,i} \alpha_i = 0, \quad (w_{t+1,i} - 1) \beta_i = 0, \\ \text{Primal feasibility:} & \quad \sum_{i=1}^n w_{t+1,i} = m, \quad 0 \leq w_{t+1,i} \leq 1, \\ \text{Dual feasibility:} & \quad \alpha_i \geq 0, \quad \beta_i \geq 0, \end{aligned} \quad (\text{D.2})$$

where $\ell_{\leq t,i} = \sum_{g=1}^t \ell_{g_t,i}$ is the cumulative loss of expert i up to trial t . Again we prove three technical lemmas before assembling them into the main proof.

Lemma D.1 Consider the m -set problem with n experts, where at the beginning of trial $t+1$, each of the first $k+1$ experts (where $k = n-m$) have incurred the same cumulative loss ℓ , and all the remaining experts are loss free, i.e.

$$\ell_{\leq t,i} = \begin{cases} \ell & \text{for } i \leq k+1 \\ 0 & \text{for } i > k+1 \end{cases}.$$

Now the FRL-GD algorithm predicts at trial $t+1$ with the weight vector \mathbf{w}_{t+1} given by:

$$w_{t+1,i} = \begin{cases} \begin{cases} \text{if } \eta \ell < \frac{k}{k+1} \text{ then } \begin{cases} \frac{m-\eta \ell(m-1)}{n} & \text{for } i \leq k+1 \\ \frac{m+\eta \ell(k+1)}{n} & \text{for } i > k+1 \end{cases} \\ \text{if } \eta \ell \geq \frac{k}{k+1} \text{ then } \begin{cases} \frac{1}{k+1} & \text{for } i \leq k+1 \\ 1 & \text{for } i > k+1 \end{cases} \end{cases} \end{cases}.$$

Proof We prove this lemma by verifying the KKT conditions (D.2). If $\eta \ell < \frac{k}{k+1}$, we have:

$$1 > \frac{m - \eta \ell(m-1)}{n} > 0, \quad \text{and} \quad 0 < \frac{m + \eta \ell(k+1)}{n} < 1.$$

Therefore $0 < w_{t+1,i} < 1$, for all i . By taking $\boldsymbol{\alpha} = \boldsymbol{\beta} = \mathbf{0}$, and $\gamma = \frac{m+\eta \ell(k+1)}{n}$, the KKT conditions can easily be verified to hold. If $\eta \ell \geq \frac{k}{k+1}$, the KKT conditions are satisfied by taking $\alpha_i = 0$ for $i \leq k+1$ and $\alpha_i = \frac{k}{k+1} - \eta \ell$ for $i > k+1$, $\boldsymbol{\beta} = \mathbf{0}$ and $\gamma = \frac{1}{k+1} + \eta \ell$. \blacksquare

Lemma D.2 Consider a trial of the m -set problem with n experts, when only one expert incurs a unit of loss. Then the FRL-GD algorithm with learning rate $\eta > 0$ decreases the weight of this expert by at most η and none of the other weights are decreased in this trial.

Proof Let $\ell_{\leq t-1}$ be the cumulative loss vector at the beginning of the trial, and let $\mathbf{w}_t, \boldsymbol{\alpha}_t, \boldsymbol{\beta}_t$ and γ_t be the corresponding primal and dual variables satisfying KKT conditions (D.2) with respect to $\ell_{\leq t-1}$. W.l.o.g., we assume the first expert incurs a unit of loss, i.e. $\ell_{\leq t} = \ell_{\leq t-1} + \mathbf{e}_1$. Let $\mathbf{w}_{t+1}, \boldsymbol{\alpha}_{t+1}, \boldsymbol{\beta}_{t+1}$ and γ_{t+1} denote the variables satisfying the KKT conditions with respect to the updated loss vector $\ell_{\leq t}$. The lemma now states that $w_{t+1,1} - w_{t,1} \geq -\eta$.

The lemma holds trivially when $\mathbf{w}_{t+1} = \mathbf{w}_t$. When $\mathbf{w}_{t+1} \neq \mathbf{w}_t$, we first show that $\gamma_{t+1} \geq \gamma_t$. Since both \mathbf{w}_t and \mathbf{w}_{t+1} sum to m , there must be an expert j , such that $w_{t+1,j} < w_{t,j}$. By the stationarity condition of (D.2), we have:

$$\begin{aligned} 0 < w_{t+1,j} - w_{t,j} &= (-\eta \ell_{\leq t,j} + \alpha_{t+1,j} - \beta_{t+1,j} + \gamma_{t+1}) - (-\eta \ell_{\leq t-1,j} + \alpha_{t,j} - \beta_{t,j} + \gamma_t), \\ \text{or, equivalently,} & \quad \gamma_{t+1} - \gamma_t > \eta(\ell_{\leq t,j} - \ell_{\leq t-1,j}) + (\alpha_{t,j} - \alpha_{t+1,j}) + (\beta_{t+1,j} - \beta_{t,j}). \end{aligned} \quad (\text{D.3})$$

Since $w_{t+1,j} > w_{t,j}$, and the weights must be non-negative, we have $w_{t+1,j} > 0$, and thus $\alpha_{t+1,j} = 0$ due to the complementary slackness condition of (D.2). Since $\alpha_{t,j}$ must be

non-negative due to the dual feasibility condition of (D.2), we have $\alpha_{t,j} \geq \alpha_{t+1,j}$. A similar argument gives $\beta_{t+1,j} \geq \beta_{t,j}$. Moreover, since $\ell_{\leq t,j} - \ell_{\leq t-1,j} \geq 0$ (due to $\ell_{\leq t} = \ell_{\leq t-1} + \mathbf{e}_t$), the RHS of (D.3) is non-negative, and thus $\gamma_{t+1} \geq \gamma_t$.

By the stationary condition of (D.2), we have:

$$\begin{aligned} w_{t+1,1} - w_{t,1} &= (-\eta \ell_{\leq t,1} + \alpha_{t+1,1} - \beta_{t+1,1} + \gamma_{t+1}) - (-\eta \ell_{\leq t-1,1} + \alpha_{t,1} - \beta_{t,1} + \gamma_t) \\ &= -\eta + (\gamma_{t+1} - \gamma_t) + (\alpha_{t+1,1} - \alpha_{t,1}) + (\beta_{t,1} - \beta_{t+1,1}), \end{aligned} \quad (\text{D.4})$$

where we used $\ell_{\leq t,1} = \ell_{\leq t-1,1} + 1$. If $\alpha_{t,1} \neq 0$, then $w_{t,1} = 0$ due to complementary slackness, and the lemma trivially holds. Similarly if $\beta_{t+1,1} \neq 0$, then $w_{t+1,1} = 1$, and again the lemma holds trivially. Thus, we may assume that $\alpha_{t,1} = \beta_{t+1,1} = 0$. However then (D.4) becomes

$$w_{t+1,1} - w_{t,1} = -\eta + (\gamma_{t+1} - \gamma_t) + \alpha_{t+1,1} + \beta_{t,1} \geq -\eta.$$

We now show the second statement of the lemma, that $w_{t+1,i} \geq w_{t,i}$ for all $i > 1$. First note that if $\alpha_{t,i} > 0$, then by the complementary slackness condition of (D.2), $w_{t,i} = 0$, and the statement trivially holds. Similarly, if $\beta_{t+1,i} > 0$, then by the complementary slackness condition, $w_{t+1,i} = 1$, and, again the statement trivially holds. Therefore we prove the statement assuming that $\alpha_{t,i} = 0$ and $\beta_{t+1,i} = 0$. Since $\ell_{\leq t,i} = \ell_{\leq t-1,i}$, the complementary slackness condition of (D.2) implies:

$$\begin{aligned} w_{t+1,i} - w_{t,i} &= (-\eta \ell_{\leq t,i} + \alpha_{t+1,i} - \beta_{t+1,i} + \gamma_{t+1}) - (-\eta \ell_{\leq t-1,i} + \alpha_{t,i} - \beta_{t,i} + \gamma_t) \\ &= (\underbrace{\alpha_{t+1,i} - \alpha_{t,i}}_{=0}) + (\underbrace{\beta_{t,i} - \beta_{t+1,i}}_{=0}) + \underbrace{(\gamma_{t+1} - \gamma_t)}_{\geq 0} \\ &\geq \alpha_{t+1,i} + \beta_{t,i} \geq 0, \end{aligned}$$

where the last inequality is by the dual feasibility condition of (D.2). This finishes the proof. \blacksquare

Lemma D.3 Consider the m -set problem with n experts, and $m = 1$. Assume at the end of trial t , the cumulative losses of all experts are the same. Assume further that the loss sequence in trials $t+1, \dots, n$ is $\ell_{t+1} = \mathbf{e}_1, \ell_{t+2} = \mathbf{e}_2, \dots, \ell_{t+n} = \mathbf{e}_n$, i.e. each expert subsequently incurs a unit of loss. Then the cumulative loss incurred by the FRL-GD algorithm in iterations $t+1, \dots, n$ is at least $1 + \frac{1}{4} \min\{\eta\eta, 1\}$.

Proof The proof goes by providing primal and dual variables satisfying the KKT conditions (D.2). Since the solution w_{t+1} to (D.2) does not change if we shift all cumulative losses $\ell_{\leq t,i}$ by a constant we can assume w.l.o.g. that the cumulative loss of all experts at the end of trial t is 0.

Take trial $t+j+1$ ($j \geq 0$), at the beginning of which each of the first j experts have already incurred a unit of loss and the remaining $n-j$ experts are loss free. If $\eta \leq \frac{1}{n-j}$, then the KKT conditions (D.2) are satisfied by taking $\alpha_i = \beta_i = 0$ for all $i = 1, \dots, n$, $\gamma = \frac{j}{n}\eta + \frac{1}{n}$, and

$$w_{t+j+1,i} = \begin{cases} \frac{1}{n} - \frac{n-j}{n}\eta & \text{for } i \leq j \\ \frac{1}{n} + \frac{j}{n}\eta & \text{for } i > j \end{cases}.$$

In this trial, expert $j+1$ incurs a unit of loss, and hence the algorithm's loss is $\frac{1}{n} + \frac{j}{n}\eta$. If $\eta > \frac{1}{n-j}$, then the KKT conditions (D.2) are satisfied by taking $\gamma = \frac{1}{n-j}$ and for $1 \leq i \leq n$, $\beta_i = 0$,

$$w_{t+j+1,i} = \begin{cases} 0 & \text{for } i \leq j \\ \frac{1}{n-j} & \text{for } i > j \end{cases}, \quad \alpha_i = \begin{cases} \eta - \frac{1}{n-j} & \text{for } i \leq j \\ 0 & \text{for } i > j \end{cases}.$$

The loss of the algorithm in such a case is $\frac{1}{n-j}$.

Thus depending on η , the algorithm's loss at trial $t+j+1$ is equal to

$$\begin{cases} \frac{1}{n} + \frac{j}{n}\eta & \text{if } \eta \leq \frac{1}{n-j} \\ \frac{1}{n-j} + \frac{1}{n} + \frac{j}{n} \frac{1}{n-k} & \text{if } \eta > \frac{1}{n-j} \end{cases},$$

which can be concisely written as: $\frac{1}{n} + \frac{j}{n} \min\left\{\eta, \frac{1}{n-j}\right\}$. Summing the above over $j = 0, \dots, n$ gives the cumulative loss of the algorithm incurred at trials $t+1, \dots, t+n$:

$$\begin{aligned} \sum_{j=0}^{n-1} \frac{1}{n} + \frac{j}{n} \min\left\{\eta, \frac{1}{n-j}\right\} &\geq \sum_{j=0}^{n-1} \frac{1}{n} + \frac{j}{n} \min\left\{\eta, \frac{1}{n}\right\} \\ &= 1 + \frac{n-1}{2} \min\left\{\eta, \frac{1}{n}\right\} \\ &\geq 1 + \frac{1}{4} \min\{\eta\eta, 1\}, \end{aligned}$$

where the last inequality is due to $n-1 > \frac{n}{2}$ for $n \geq 2$. \blacksquare

We are now ready to give the proof of Theorem 4.3:

Theorem 4.3 Consider the m -set problem with $k \leq n/2$ and unit bit vectors as loss vectors. Then for any fixed learning rate η , the FRL-GD algorithm (D.1) can be forced to have regret at least $\Omega(\max\{\min\{B_L, k\sqrt{B_L}\}, k\})$.

Proof Theorem 5.6 gives a lower bound of $\Omega(\sqrt{B_L m \log \frac{n}{m}} + m \log \frac{n}{m})$ that holds for any algorithm. This lower bound is at least $\Omega(k)$ since $m \log \frac{n}{m} = m \log(\frac{n}{m} + 1) \geq k$. Hence to prove this theorem, we only need to show a lower bound of $\Omega(\{\min\{B_L, k\sqrt{B_L}\}\})$. Similarly as in the proof of Theorem 4.1, we show this in two steps: First, we give two loss sequences that force FRL-GD to have regret at least $\Omega(k/\eta)$ and $\Omega(\min\{B_L, kB_L\eta\})$, respectively. Then, the lower bound follows by taking the maximum between the two lower bounds.

The first loss sequence is exactly the same as in the proof of Theorem 4.1, i.e. the sequence consists of $\lfloor \frac{k\eta}{m} \rfloor + 1$ trials and in each trial, the expert with the largest weight (within the first k experts) incurs one unit of loss. With Lemma D.2 in place of Lemma C.2, one can easily show an $\Omega(k/\eta)$ regret lower bound for FRL-GD by repeating the argument from the proof of Theorem 4.1.

Now we describe the second loss sequence which forces the FRL-GD algorithm to suffer regret $\Omega(\min\{B_L, kB_L\eta\})$. For the sake of clarity, we assume that B_L is integer (otherwise

replace B_L by $[B_L]$ in the proof). The sequence consists of B_L “rounds”, and each round consist of $k + 1$ trials (so that there are $(k + 1)B_L$ trials in total). In each round, one unit of loss is given alternately to each of the first $k + 1$ experts, one at a time. In other words, in trial t , the loss vector ℓ_t equals to e_r where $r = t \bmod (k + 1)$. The best comparator of this sequence consists of the last $m - 1$ loss free experts and any of the first $k + 1$ experts, which incurs cumulative loss B_L .

To lower bound the loss of the algorithm, first notice that in each round, each of the first $k + 1$ experts incurs exactly one unit of loss. The sum of weights of these experts at the beginning of a round lower bounds the algorithm’s loss in this round. This is because the weight of a given expert cannot decrease if the expert does not incur any loss (Lemma D.2); hence, the weight of a given expert at a trial, in which that expert receives a unit of loss, will be at least as large as the weight of that expert at the beginning of a round. Since the weights are initialized uniformly, this sum is $m(k + 1)/n$ before round 1, and by Lemma D.1, each of the following rounds decreases it by $(m - 1)(k + 1)/n$ until it is lower capped at 1 (Since the total sum of the weights is m , and none of the remaining $m - 1$ weights can exceed 1, the sum of weights of the first $k + 1$ experts must be at least 1).

We first assume that after $B_L/2$ rounds, this sum is strictly larger than 1 which means the sum decreases as an arithmetic series for all the first $B_L/2$ rounds and the algorithm’s loss can be lower bounded by

$$\frac{1}{2}(m(k + 1)/n + 1) \frac{B_L}{2} \stackrel{\text{Use the same argument as in (C.9)}}{=} B_L/2 + \Omega(B_L).$$

Since the sum of the first $k + 1$ weights at the beginning of any trial is at least 1, the algorithm incurs loss at least $B_L/2$ in the remaining $B_L/2$ rounds. Summing up the algorithm’s loss on both halves of the sequence, we get a regret lower bound of $\Omega(B_L)$.

Now consider the case, when after the first $B_L/2$ rounds, the sum of the first $k + 1$ weights is 1. This implies that the weights of $m - 1$ remaining experts are all equal to 1, and will stay at this value, since only the first $k + 1$ experts incur any loss (and, by Lemma D.2, the weight of an expert cannot decrease if that expert does not incur any loss). Thus, we can disregard the loss free $m - 1$ experts, and in the remaining $B_L/2$ rounds, the first $k + 1$ expert weights are updated as in the m -set problem with $m = 1$. Notice that the algorithm suffers loss at least $B_L/2$ in the first $B_L/2$ rounds and by Lemma D.3, suffers loss at least $B_L/2 + B_L \min\{(k + 1)/n, 1\}/8$ in the second $B_L/2$ rounds. Summing up the algorithm’s loss on both halves of the sequence, we get a regret lower bound of $\Omega(\min\{B_L, kB_L/n\})$. ■

Appendix E. A Discussion on the Final Parameter of FRL-GD

In this appendix, we show that the final parameter matrix of the FRL-GD algorithm essentially contains the solution to the batch PCA problem. First recall that given n dimensional data points $\mathbf{x}_1, \dots, \mathbf{x}_T$, the batch version of the k -PCA problem is solved by the eigenvectors of the k largest eigenvalues of the covariance matrix $C = \sum_{T=1}^T \mathbf{x}_T \mathbf{x}_T^\top$. Let \mathbf{W}_{T+1} be the final parameter matrix of the FRL-GD algorithm when the instance matrices are $\mathbf{X}_1 = \mathbf{x}_1 \mathbf{x}_1^\top, \dots, \mathbf{X}_T = \mathbf{x}_T \mathbf{x}_T^\top$. We will show that the eigenvectors of the $m = n - k$ largest eigenvalues of \mathbf{W}_{T+1} are the same as the eigenvectors of the m largest eigenvalues of the

negated covariance matrix $-C$. Thus, by computing the complementary subspace of rank k , one finds the solution of the batch PCA problem with respect to data points $\mathbf{x}_1, \dots, \mathbf{x}_T$.

Recall that the final parameter \mathbf{W}_{T+1} of FRL-GD is the projection of the $-C$ into the parameter set \mathcal{W}_m :

$$\mathbf{W}_{T+1} = \operatorname{argmin}_{\mathbf{W} \in \mathcal{W}_m} \|\mathbf{W} - \mathbf{W}_{T+1}^*\|_F^2.$$

Let $-C$ have eigendecomposition $-C = U \operatorname{diag}(\boldsymbol{\lambda}) U^\top$, where $\boldsymbol{\lambda}$ is the vector of the eigenvalues of $-C$. Agra et al. (2013, Lemma 3.2) shows that the projection of $-C$ is solved by projecting the eigenvalues $\boldsymbol{\lambda}$ into \mathcal{S}_m while keeping its eigensystem unchanged:

$$\mathbf{W}_{T+1} = U \operatorname{diag}(\boldsymbol{\lambda}') U^\top \quad \text{and} \quad \boldsymbol{\lambda}' = \operatorname{argmin}_{\boldsymbol{\lambda}' \in \mathcal{S}_m} \|\boldsymbol{\lambda} - \boldsymbol{\lambda}'\|_2^2.$$

W.l.o.g., assume the elements of $\boldsymbol{\lambda}$ are in descending order, i.e. $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m$. To prove that the eigenvectors of the m largest eigenvalues are the same in \mathbf{W}_{T+1} and $-C$, we only need to show the following: for any integers pair i and j such that $1 \leq i \leq m < j \leq n$, if $\lambda_i > \lambda_j$, then $\lambda'_i > \lambda'_j$. First note that by the KKT analysis of the problem of projecting into \mathcal{S}_m (see (D.2)), it is easy to see that if $\lambda_i > \lambda_j$, then exactly one of the following three cases holds.

$$\lambda'_i > \lambda'_j \quad \text{or} \quad \lambda'_i = \lambda'_j = 0 \quad \text{or} \quad \lambda'_i = \lambda'_j = 1.$$

Now we show that when i and j further satisfy $i \leq m < j$, the latter two cases can never happen. Suppose $\lambda'_i = \lambda'_j = 1$ for some $i \leq m < j$. In this case for any $i' \leq m$, $\lambda'_{i'} = \lambda'_j = 1$ also holds. Therefore, the sum of all the coordinates of $\boldsymbol{\lambda}'$ will be at least $m + 1$ which contradicts $\boldsymbol{\lambda}' \in \mathcal{S}_m$. Now assume $\lambda'_i = \lambda'_j = 0$ for some $i \leq m < j$. In this case for any $m < j'$, $\lambda'_{j'} = \lambda'_j = 0$ also holds. This implies that the sum of all the coordinates of $\boldsymbol{\lambda}'$ will be at most $m - 1$ which again contradicts $\boldsymbol{\lambda}' \in \mathcal{S}_m$.

Appendix F. Regret Lower Bounds When the Number of Trials Is Large

This appendix proves lower bounds on the regret of any online algorithm for the m -set problem: Theorem 5.1 and Theorem 5.2 prove lower bounds for unit bit vectors as loss vectors and Theorem 5.4 proves lower bounds for arbitrary bit vectors as loss vectors. In all of these lower bounds, we assume that the number of the trials T is larger than either the number of experts n or some function of n , m and k (see details of the assumptions in individual theorems). The regret lower bounds for small number of trials are given in the next Appendix G.

All the lower bounds given in this appendix are proved with the probabilistic bounding technique described in Section 5, i.e. in each case, we first choose a probability distribution \mathcal{P} and then show a lower bound on the expected regret of any algorithm when the loss vectors are generated i.i.d. from \mathcal{P} . Our lower bounds on the expected regret make use of the following lemma which gives an upper bound on the expected loss of the best comparator in a two expert game.

Lemma F.1 Consider a two expert game in which the random loss pairs of both experts are i.i.d. between trials, and at each trial the random pair follows the distribution:

$$\begin{array}{c|ccc} \text{value of the loss pair} & (0, 1) & (1, 0) & (1, 1) & (0, 0) \\ \hline \text{probability} & p & p & q & 1 - 2p - q \end{array} \quad (\text{F.1})$$

where non-negative parameters p and q satisfy $2p+q \leq 1$. Let M be the minimum total loss of the two experts in T such trials. If T and p satisfy $Tp \geq 1/2$, then

$$\mathbb{E}[M] \leq T(p+q) - c\sqrt{Tp}$$

for some constant $c > 0$ independent of T , p and q .

Later we will use the case $q = 0$ of the two expert distribution (F.1) as a submodule for building distributions over the n unit bit vectors and $p = q = 1/4$ for building distributions over $\{0, 1\}^n$. To prove Lemma F.1, we need the following lemma from (Koolen, 2011, Theorem 2.5.3):

Lemma F.2 Let a_t and b_t be two binary random variables following the distribution

$$\frac{\text{value of } (a_t, b_t)}{\text{probability}} \mid \begin{array}{l} (0, 1) \\ 0.5 \end{array} \mid \begin{array}{l} (1, 0) \\ 0.5 \end{array}.$$

For T independent such pairs, we have

$$\frac{T}{2} - \sqrt{\frac{T-1}{2\pi}} \leq \mathbb{E} \left[\min \left\{ \sum_{t=1}^T a_t, \sum_{t=1}^T b_t \right\} \right] \leq \frac{T}{2} - \sqrt{\frac{T+1}{2\pi}}.$$

Proof of Lemma F.1 Denote the experts' losses at trials $1 \leq t \leq T$ by \tilde{a}_t and \tilde{b}_t . In this notation, the statement of Lemma F.1 is equivalent to:

$$\mathbb{E} \left[\min \left\{ \sum_t \tilde{a}_t, \sum_t \tilde{b}_t \right\} \right] \leq T(p+q) - c\sqrt{Tp}.$$

At each trial, the random variable pair $(\tilde{a}_t, \tilde{b}_t)$ has four possible values: $(1, 0)$, $(0, 1)$, $(1, 1)$ or $(0, 0)$. If $\tilde{a}_t \neq \tilde{b}_t$, then this trial is "covered by" Lemma F.2. If $\tilde{a}_t = \tilde{b}_t$, then this trial affects $\sum_t \tilde{a}_t$ and $\sum_t \tilde{b}_t$ the same way and therefore can be excluded from the minimization. We formalize this observation as follows:

$$\begin{aligned} \mathbb{E} \left[\min \left\{ \sum_t \tilde{a}_t, \sum_t \tilde{b}_t \right\} \right] &= \mathbb{E} \left[\min \left\{ \sum_{t: \tilde{a}_t \neq \tilde{b}_t} \tilde{a}_t, \sum_{t: \tilde{a}_t \neq \tilde{b}_t} \tilde{b}_t \right\} \right] + \mathbb{E} \left[\sum_{t: \tilde{a}_t = \tilde{b}_t} \tilde{a}_t \right] \\ &\stackrel{\text{Lemma F.2}}{\leq} \mathbb{E} \left[\frac{R-1}{2} - \sqrt{\frac{R-1}{2\pi}} \right] + Tq, \end{aligned}$$

where R is a binomial random variable with T draws and success probability $2p$. Clearly $\mathbb{E}[R] = 2Tp$ and therefore $\mathbb{E}[\frac{R-1}{2}] = Tp$. Moreover under the assumption that $Tp \geq 1/2$, we will show in Lemma I.2 of Appendix I (using an application of the Chernoff bound) that $\mathbb{E} \left[\sqrt{\frac{R-1}{2\pi}} \right] \geq c\sqrt{Tp}$ for some constant c that does not depend on T , p and q . ■

We now use Lemma F.1 to prove the following theorem which addresses the m -set problem with unit bit vectors for the case $k \leq \frac{n}{2}$.

Theorem 5.1 Consider the m -set problem with unit bit vectors as loss vectors, where $m = n - k$. Then for $k \leq \frac{n}{2}$ and $T \geq k$, any online algorithm suffers worst case regret at least $\Omega(\sqrt{Tk})$.

Proof In this proof, each loss vector is uniformly sampled from the first $2k$ unit bit vectors, i.e. at each trial, one of the first $2k$ experts is uniformly chosen to incur a unit of loss. To show an upper bound on the loss of the comparator, we group these $2k$ experts into k pairs and note that the loss of each expert pair follows the joint distribution described Lemma F.1 with $p = \frac{1}{2k}$ and $q = 0$. Furthermore, the condition $Tp \geq 1/2$ of Lemma F.1 is also satisfied because of the assumption $T \geq k$. Hence, by applying Lemma F.1 we know that the expected loss of the winner in each pair is at most $T/2k - c\sqrt{T/2k}$, and the total expected loss for k winners from all k pairs is upper bounded by $T/2 - c\sqrt{kT/2}$. Now recalling that the comparator consists of the $m = n - k$ best experts, its total expected loss is upper bounded by the expected loss of the k winners, which is at most $T/2 - c\sqrt{kT/2}$, plus the expected loss of the remaining $n - 2k$ experts, which is zero. Therefore, we have an upper bound of $T/2 - c\sqrt{kT/2}$ on the expected loss of the comparator. On the other hand, since losses are generated independently between trials, any online algorithm suffers loss at least $T/2$. The difference between the lower bound on the expected loss of the algorithm and the upper bound on the expected loss of the best m -set gives the regret lower bound of the theorem. ■

The case $k \geq \frac{n}{2}$ is more complicated. Recall that $k = n - 1$ reproduces the vanilla single expert case. Therefore additional $\log n$ factor must appear in the square root of the lower bound. We need the following lemma, which is a generalization of Lemma F.1 to n experts. In the proof, we upper bound the minimum loss of the experts by the loss of the winner of a tournament among the n experts. The tournament winner does not necessarily have the lowest loss. However as we shall see, its expected loss is close enough to the expected loss of the best expert so that this bounding technique is still useful for obtaining lower bounds on the regret.

Lemma F.3 Choose any n, S and T , such that $n = 2^S$ and S divides T . If the loss sequence of length T is generated from a distribution \mathcal{P} , such that:

- at each trial t , the distribution of losses on n experts is exchangeable, i.e. for any permutation π on a set $\{1, \dots, n\}$, and for any t , $\ell_t = (\ell_{t,1}, \ell_{t,2}, \dots, \ell_{t,n})$ and $\ell_t^\pi = (\ell_{t,\pi(1)}, \ell_{t,\pi(2)}, \dots, \ell_{t,\pi(n)})$ have the same distribution,
- and the distribution of losses is i.i.d. between trials,

then,

$$\mathbb{E}[\text{minimum loss among } n \text{ experts in } T \text{ trials}] \leq S \mathbb{E}[\text{minimum loss among experts } 1 \text{ and } 2 \text{ in } T/S \text{ trials}].$$

Proof The key idea is to upper bound the minimum loss of any expert by the loss of the expert that wins an S round tournament. In the first round, we start with n experts and pair each expert with a random partner. The round lasts for T/S trials. For each pair, the expert with the smaller loss wins in this round (tie always broken randomly). The

	first round			second round		
$t = 1$	$t = 2$	$t = 3$	$t = 4$	$t = 5$	$t = 6$	
[expert 1	1	0	1	0	0	
expert 4	0	1	0	1	1	
[expert 2	0	0	1	1	0	
expert 3	1	1	0	1	1	

Table F.1: A tournament with $T = 6$ trials, $S = 2$ rounds, and $n = 4$ experts. The bits in the table are the binary losses of the experts in each trial. The brackets show the pairing in each round. The losses of the winners are in bold

$n/2$ winners continue to the second round. At round s , the remaining $n/2^{s-1}$ experts are randomly paired and the winners are determined based on the losses in another set of T/S trials. After S rounds and $T = ST/S$ trials we are left with 1 overall winner.

For example for $S = 2$ rounds, $n = 2^2 = 4$ experts and $T = 6$ trials, consider the sequence of losses shown in Table F. Each of the two tournament consists of $6/2 = 3$ trials. In the first round, expert 1 is paired with expert 4 and expert 2 with expert 3. The cumulative losses of experts 1, 2, 3, 4 in this round are 2, 1, 2, 1, respectively. So expert 4 is the winner of the first pair and expert 2 is the winner of the second pair. In the second round, the two winners (experts 2 and 4) are paired, and they incur cumulative loss 2 and 3, respectively. Hence, expert 2 wins the tournament. The total loss of the tournament winner in all 6 trials is 3. Note that this is larger than the minimum total loss of the 4 experts since expert 1 incurred total loss 2. Nevertheless we shall see that for our probabilistic lower bound proof, the total loss of the tournament winner is close enough to the total loss of the best expert.

To complete the proof it suffices to show that

$$\begin{aligned} & \mathbb{E} [\text{total loss of tournament winner in } T \text{ trials}] \\ &= S \mathbb{E} [\text{minimum loss among experts 1 and 2 in } T/S \text{ trials}]. \end{aligned}$$

Due to linearity of expectation:

$$\begin{aligned} & \mathbb{E} [\text{total loss of tournament winner in } T \text{ trials}] \\ &= \sum_{i=1}^S \mathbb{E} [\text{total loss of tournament winner in } i\text{-th round}]. \end{aligned}$$

The exchangeability of the losses and the symmetry of the tournament guarantees that each expert is equally likely to be the overall winner. Therefore w.l.o.g., expert 1 is the overall winner. Consider i -th round of the tournament ($1 \leq i \leq S$), and let (w.l.o.g.) expert 2 be

the partner of expert 1 in this round. We have:

$$\begin{aligned} & \mathbb{E} [\text{total loss of tournament winner in } i\text{-th round}] \\ &= \mathbb{E} \left[\begin{array}{c} \text{total loss of exp. 1 in } i\text{-th round} \\ \text{total loss of exp. 2 won all past competitions} \\ \text{at rounds } 1, \dots, i-1. \end{array} \right] \\ &= \mathbb{E} [\text{total loss of exp. 1 in } i\text{-th round} \mid \text{exp. 1 wins over exp. 2 in } i\text{-th round}]. \end{aligned}$$

The second equality is due to the fact that the distribution of losses is i.i.d. between trials, and therefore the future and past rounds are independent of the current round. Since the last expression is the same for each of the S rounds we have:

$$\begin{aligned} & \mathbb{E} [\text{total loss of tournament winner in } T \text{ trials}] \\ &= S \mathbb{E} [\text{expected loss of expert 1 in } T/S \text{ trials} \mid \text{expert 1 wins over expert 2}]. \end{aligned}$$

Remains to be shown that the latter expectation is simple the expectation of the minimum of the two experts losses in a single round. Let L_1 and L_2 be the total losses of both experts in the T/S trials and let " $L_1 > L_2$ " denote the event that 1 wins over 2 (ties broken uniformly; so that, e.g., $\Pr(L_1 > L_2) + \Pr(L_2 > L_1) = 1$). Then

$$\begin{aligned} \mathbb{E} [L_1 | L_2 > L_1] &= \left(\Pr(L_2 > L_1) + \Pr(L_1 > L_2) \right) \mathbb{E} [L_1 | L_2 > L_1], \\ &= \Pr(L_2 > L_1) \mathbb{E} [L_1 | L_2 > L_1] + \Pr(L_1 > L_2) \mathbb{E} [L_1 | L_2 > L_1] \\ \text{(exchangeability)} &= \Pr(L_2 > L_1) \mathbb{E} [L_1 | L_2 > L_1] + \Pr(L_1 > L_2) \mathbb{E} [L_2 | L_1 > L_2] \\ &= \Pr(L_2 > L_1) \mathbb{E} [\min\{L_1, L_2\} | L_2 > L_1] \\ &\quad + \Pr(L_1 > L_2) \mathbb{E} [\min\{L_1, L_2\} | L_1 > L_2] \\ &= \mathbb{E} [\min\{L_1, L_2\}]. \quad \blacksquare \end{aligned}$$

Now, we use this lemma to prove a lower bound for the m -set problem with $k \geq \frac{n}{2}$:

Theorem 5.2 *Consider the m -set problem with unit bit vectors as loss vectors, where $m = n - k$. Then for $k \geq \frac{n}{2}$ and $T \geq n \log_2(n/m)$, any online algorithm suffers worst case regret at least $\Omega(m \sqrt{T \ln(n/m)/n})$.*

Proof Let us first assume that $n = 2^j m$ for some integer $j > 0$, i.e. $\log_2(n/m)$ is a positive integer, and that $\frac{T}{\log_2(n/m)}$ is an integer value as well.

At each trial, a randomly chosen expert out of n experts incurs a unit of loss. To show an upper bound on the loss of the comparator, we partition the n experts into m groups (n divides m from the assumption), and notice that the losses of the n/m experts in each group are exchangeable. Applying Lemma F.3 to each group of n/m experts with $S = \log_2(n/m)$ rounds and T/S trials per round, we obtain:

$$\begin{aligned} & \mathbb{E} [\text{Loss of the winner in a given group in } T \text{ trials}] \\ &\leq \log_2 \binom{n}{m} \mathbb{E} \left[\text{Loss of the winner of two experts in } \frac{T}{\log_2(n/m)} \text{ trials} \right]. \quad (\text{F.2}) \end{aligned}$$

The expectation on the RHS is the two expert game considered in Lemma F.1 with parameters $p = 1/n$ and $q = 0$. Note that $q = 0$ because only one expert suffers loss in each trial. Applying this lemma bounds the expectation on the RHS as

$$\frac{T}{\log_2(n/m)n} - c\sqrt{\frac{T}{\log_2(n/m)n}}.$$

Plugging this into (F.2) gives $T/n - c\sqrt{T \log_2(n/m)/n}$ upper bound on the expected loss of the winner in a given group. We upper bound the expected loss of the comparator by the total loss of m winners from the m groups, which in expectation is at most $Tm/n - cm\sqrt{T \log_2(n/m)/n}$.

Finally the loss of the algorithm is lower bounded as follows: Every expert incurs loss $1/n$ in expectation at each trial and losses are i.i.d. between trials. Therefore any online algorithm suffers loss at least mT/n , and the expected regret is lower bounded by $cm\sqrt{T \log_2(n/m)/n}$. This concludes the proof when $n = 2^j m$ and $\log_2(n/m)$ divides T .

If n is not of this form, we take the largest $n_0 < n$, such that $n_0 = 2^j m$ for some integer j , i.e. $n_0 = \max_{j \in \mathbb{N}} \{2^j m : 2^j m \leq n\}$. We then apply the reasoning above to n_0 experts, while the remaining $n - n_0$ will incur loss 1 all the time, which can only increase the loss of the algorithm, but this will not affect the loss of the comparator (comparator will never pick these experts). Since $n_0 \geq n/2$ (otherwise n_0 would not be the largest integer of the form $2^j n$, smaller than n), this does not change the rate under $\Omega(\cdot)$ for the lower bound in the statement of the theorem. Finally, if $\frac{T}{\log_2(n/m)}$ is not an integer value, we can choose the largest $T_0 < T$, such $\frac{T_0}{\log_2(n/m)}$ is integer, and use the proof with T_0 rounds, while in the remaining $T - T_0$ rounds all losses are zero. Since $T_0 \geq T/2$, this, again, does not change the rate under $\Omega(\cdot)$. ■

Finally, we consider the m -set problems with L_∞ -bounded loss vectors. The following theorem proves lower bounds for such problems when $k \leq \frac{n}{2}$ and when $k \geq \frac{n}{2}$.

Theorem 5.4 Consider the m -set problem with loss vectors in $\{0, 1\}^n$, where $m = n - k$. Then for $T \geq \log_2 \frac{n}{\min\{k, m\}}$, any online algorithm suffers worst case regret of at least

$$\Omega\left(k\sqrt{T \ln \frac{n}{k}}\right) \text{ when } k \leq \frac{n}{2} \quad \text{or} \quad \Omega\left(m\sqrt{T \ln \frac{n}{m}}\right) \text{ when } k \geq \frac{n}{2}.$$

Proof The proof is similar to the proof of Theorem 5.2, except that at each trial, the losses of all n experts are i.i.d. Bernoulli random variable with probability $p = 1/2$. For such a distribution over losses, any algorithm suffers expected cumulative loss at least $mT/2$ for the m -set problem.

For the sake of simplicity, we make some assumptions about n , k and T that avoid rounding issues. When $k \leq n/2$, we assume that $n = 2^j k$ for some integer $j \geq 1$ and that $\frac{T}{\log_2(n/k)}$ is an integer. When $k \geq n/2$, i.e. $m = n - k \leq n/2$, we assume that $n = 2^j m$ for some integer $j \geq 1$ and that $\frac{T}{\log_2(n/m)}$ is an integer. As in the proof of Theorem 5.2, it is easy to generalize the theorem to arbitrary n , k and T satisfying $T \geq \log_2 \frac{n}{\min\{k, m\}}$.

Now, we prove our regret lower bounds for each of the two cases: When $m \leq n/2$, we group the experts into m groups of size n/m and upper bound the loss of the comparator using the m group winners. As before, the loss of each winner can be upper bounded by the lemmas F.1 (with $p = q = 1/4$) and F.3:

$$\begin{aligned} \mathbb{E}[\text{Loss of the winner in a given group in } T \text{ trials}] & \\ \stackrel{\text{Lemma F.3}}{\leq} \log_2 \frac{n}{m} & \mathbb{E}\left[\text{Loss of two experts in } \frac{T}{\log_2(n/m)} \text{ trials}\right] \\ \stackrel{\text{Lemma F.1}}{\leq} \frac{T}{2} - c\sqrt{\frac{T}{4} \log_2 \frac{n}{m}}. & \end{aligned}$$

Note that since the experts here incur i.i.d. *Bernoulli*($\frac{1}{2}$) losses, the above application of Lemma F.1 requires $p = q = 1/4$. Next, summing up m winners, we have the expected loss of the comparator upper bounded by $Tm/2 - cm\sqrt{T \log_2(n/m)/4}$. Taking the difference between this upper bound and the $Tm/2$ lower bound on loss of any algorithm results in the claimed $\Omega(m\sqrt{T \ln(n/m)})$ lower bound on the regret.

When $k \leq n/2$, we group the experts into k groups and consider a *loser* out of each group which is the expert which incurs the *largest* loss in each group. One can flip around the content of Lemma F.1 and F.3 to show that the loser in a group of n/k experts incurs loss in expectation at least $T/2 + c\sqrt{T \log_2(n/k)/4}$. Therefore, the expected loss of all k losers is lower bounded by $Tk/2 + ck\sqrt{T \log_2(n/k)/4}$. Now note that the expected loss of the comparator is upper bounded by the expected total loss of all the experts, which is $Tn/2$, minus the expected loss of the k losers, and hence upper bounded by

$$\frac{Tn}{2} - \left(\frac{Tk}{2} + ck\sqrt{\frac{T}{4} \log_2 \frac{n}{k}}\right) = \frac{Tm}{2} - ck\sqrt{\frac{T}{4} \log_2 \frac{n}{k}}.$$

Finally, the claimed regret bound follows from taking the difference between this upper bound and the $Tm/2$ lower bound on the loss of any online algorithm. ■

Appendix G. Regret Lower Bounds When the Number of Trials Is Small

This appendix gives general regret lower bounds for the m -set problem when the number of trials T is small: Theorem G.1 and Theorem G.3 show lower bounds when the loss vectors are unit bit vectors; Theorem G.4 and Theorem G.5 show lower bounds when the loss vectors are bit vectors. Unlike the lower bounds for large T that are proved with probabilistic arguments (see previous Appendix F) all of the lower bounds in this appendix are proved by showing explicit adversary strategies that force large regret to any online algorithm. The matching upper bounds for small T are trivial and can be found in Section 5.

Theorem G.1 Consider the m -set problem with unit bit vectors as loss vectors, where $m = n - k$. Then for $k \leq \frac{n}{2}$ and $T \leq k$, any online algorithm suffers worst case regret at least $\Omega(T)$.

Proof Consider an adversary that at each trial gives a unit of loss to the expert with the largest weight assigned by the algorithm. Recall that $m = n - k$ and $k \leq \frac{n}{2}$. Therefore

all the weights assigned by the algorithm sum to $m \geq \frac{n}{2}$ and the largest weight out of n experts is at least $\frac{n}{2}$. Hence, after T trials, any algorithm suffers total loss at least $\frac{n}{2}$. On the other hand, since there are at least $n - T \geq m$ (because $T \leq k$) experts that are loss free, the loss of the best m -set of experts is zero. Therefore, the regret of any algorithm is at least $\frac{n}{2}$. ■

Now we consider the case when $k \geq \frac{n}{2}$. We start with a lemma which is parameterized by an integer $1 \leq i \leq k$ instead of the number of the trials T .

Lemma G.2 *Consider the m -set problem with unit bit vectors as loss vectors, where $m = n - k$. For any integer $1 \leq i \leq k$, an adversary can force any algorithm to suffer loss $\Omega(m \log_2 \frac{n}{n-i})$ in $O(n \log_2 \frac{n}{n-i})$ trials, and at the same time, keep a set of m experts with loss zero.*

Proof The adversary's strategy has i rounds, where the j -th round ($1 \leq j \leq i$) has at most $\lceil \frac{n}{n-j+1} \rceil$ trials and after it finishes, there will be at least $n - j$ experts that still have loss zero. The first round has only one trial, in which a unit of loss is given to the expert with the largest weight. Since all the weights assigned by the algorithm sum to m , the algorithm suffers loss at least $\frac{m}{n}$ in the first round.

Each of the following rounds may contain multiple trials and at the end of round $j - 1$ ($2 \leq j \leq i$), there are still at least $n - j + 1$ loss free experts. In round j , the adversary uses a strategy with two subcases as follows: The adversary first considers the experts that are still loss free. If any of the first $n - j + 1$ of them has weight at least $\frac{2m}{2(n-j+1)}$, then we are in case 1, where a unit of loss is given to this expert. Otherwise, we are in case 2, in which the adversary considers the remaining $j - 1$ experts (which may or may not be loss free) and gives a unit of loss to the one with the largest weight among them. The j -th round ends when the algorithm has suffered total loss at least $\frac{m}{2(n-j+1)}$ in that round. Note that whenever case 1 is reached, a round ends immediately. Our strategy guarantees that after round j , there are at least $n - j$ experts that are loss free. Next we upper bound the number of case 2 trials in a round by showing a lower bound on the loss of the algorithm in case 2 trials. Recall that in case 2, $n - j + 1$ experts have weight no more than $\frac{2m}{2(n-j+1)}$ each, and the expert that has the largest weight in the remaining $j - 1$ experts incurs a unit of loss. Using these facts as well as the fact that all the weights sum to m , we can lower bound the weight of the expert that incurs loss (which is also the loss of the algorithm) as follows:

$$\frac{\left(m - \frac{m}{2(n-j+1)}(n-j+1)\right)}{j-1} \geq \frac{m}{2(j-1)} \geq \frac{m}{2n}.$$

Recalling that the j -th round ends when the algorithm suffers total loss $\frac{m}{2(n-j+1)}$ in that round, we conclude that the j -th round can have at most $\lceil \frac{n}{n-j+1} \rceil$ trials.

Summing up over i rounds, the algorithm suffers total loss at least $\sum_{j=1}^i \frac{m}{2(n-j+1)} = \Omega(m \log_2 \frac{n}{n-i})$ in at most $\sum_{j=1}^i \lceil \frac{n}{2(n-j+1)} \rceil = O(n \log_2 \frac{n}{n-i})$ trials. On the other hand, the loss of the best m -set of experts is zero due to assumption $i \leq k$ and the fact that after $j = i$ rounds, there are at least $n - i$ loss free experts. Hence the lemma follows. ■

Theorem G.3 *Consider the m -set problem with unit bit vectors as loss vectors, where $m = n - k$. Then for $k \geq \frac{n}{2}$ and $T \leq n \log_2 \frac{n}{m}$, any algorithm suffers worst case regret at least $\Omega(\frac{n}{m}T)$.*

Proof Lemma G.2 states that there exist two positive constants c_1 and c_2 , such that for any integer $1 \leq i \leq k$, the adversary can force any algorithm to suffer regret at least $c_1 m \log_2 \frac{n}{n-i}$ in at most $c_2 n \log_2 \frac{n}{n-i}$ trials. The proof splits into two cases, depending on the number of the trials T :

- When $T < c_2 n \log_2 \frac{n}{n-1}$, T is upper bounded by a constant as follows:

$$T < c_2 n \log_2 \frac{n}{n-1} = \frac{c_2 n}{\log 2} \log \left(1 + \frac{1}{n-1}\right) \leq \frac{c_2 n}{(n-1) \log 2} \stackrel{n \geq 2}{\leq} \frac{2c_2}{\log 2}.$$

Since the adversary can always force any algorithm to suffer constant regret, the theorem holds trivially.

- When $T \geq c_2 n \log_2 \frac{n}{n-1}$, we set $i = \min\{\lceil i' \rceil, k\}$, where $i' = n(1 - 2^{-T/c_2 n})$ is the solution of $c_2 n \log_2 \frac{n}{n-i'} = T$. We note that the function $c_2 n \log_2 \frac{n}{n-i'}$ is monotonically increasing in i' , which results in two facts: first, $i' \geq 1$, since we assumed $T \geq c_2 n \log_2 \frac{n}{n-1}$; second, $c_2 n \log_2 \frac{n}{n-i'} \leq T$, since $i \leq \lceil i' \rceil$. We further show that $c_2 n \log_2 \frac{n}{n-i'} \geq \min\{c_2, \frac{1}{3}\}T$ as follows:

$$\begin{aligned} & \text{— When } i = \lceil i' \rceil, \text{ first note that } \left(\frac{n}{n-i'}\right)^3 \geq \frac{n}{n-T}, \text{ since:} \\ & (n-i')n^2 - (n-i')^3 \geq (n-i-1)n^2 - (n-i)^3 = 2ni^2 + 3ni^2 - i^3 - n^2 \stackrel{1 \leq i < n}{\geq} 0. \end{aligned}$$

$$\text{Plugging } c_2 n \log_2 \frac{n}{n-i'} = T, \text{ we have } c_2 n \log_2 \frac{n}{n-i'} \geq \frac{1}{3}T.$$

- When $i = k$, $c_2 n \log_2 \frac{n}{n-i} = c_2 n \log_2 \frac{n}{n-i} \geq c_2 T$, since $T \leq n \log_2 \frac{n}{m}$ is assumed in the theorem.

Now, using Lemma G.2 with i set as $i = \min\{\lceil i' \rceil, k\}$, results in an adversary that forces the algorithm to suffer regret at least $c_1 m \log_2 \frac{n}{n-i} \geq \frac{m \log 2}{n c_2} \min\{c_2, \frac{1}{3}\}T = \Omega(\frac{n}{m}T)$ in at most T trials. When the adversary uses less than T trials, then the game can be extended to last exactly T trials by playing zero loss vectors for the remaining trials. ■

Theorem G.4 *Consider the m -set problem with loss vectors in $\{0, 1\}^n$, where $m = n - k$. Then for $k \geq \frac{n}{2}$ and $T \leq \log_2 \frac{n}{m}$, the worst case regret of any algorithm is at least $\Omega(Tm)$.*

Proof The proof uses an adversary which forces any algorithm to suffer loss $\Omega(Tm)$, and still keeps the best m -set of experts to be loss free. Note that at each trial, the adversary decides on the loss vector after the algorithm makes its prediction w_t , where $w_t \in \{0, 1\}^n$ with $\sum_i w_{t,i} = m$.

At trial one, the adversary first sorts the n experts by their weights assigned by the algorithm, and then gives a unit of loss to each of the experts in the first half, i.e. the experts with larger weights. Since the weights sum to m , the total weight assigned to the experts in the first half is at least $\frac{m}{2}$. Hence in the first trial, the algorithm suffers loss at least $\frac{m}{2}$.

At each of the following trials, the adversary only sorts those experts that have not incur any loss so far and gives unit losses to the first half (the half with larger weights) of these experts, as well as all the experts that have already incurred losses before this trial. It is easy to see that in this way the algorithm suffers loss at least $\frac{m}{2}$ at each trial.

Since the number of the experts that are loss free halves at each trial, after $T \leq \log_2 \frac{m}{2}$ trials, there will still be at least m loss free experts. Now since the algorithm suffers loss at least $\frac{mT}{2}$ in T trials, the theorem follows. ■

Theorem G.5 Consider the m -set problem with loss vectors in $\{0, 1\}^n$, where $m = n - k$. Then for $k \leq \frac{n}{2}$ and $T \leq \log_2 \frac{n}{k}$, any algorithm suffers worst case regret at least $\Omega(Tk)$.

Proof The proof becomes conceptually simpler if we use the notion of gain defined as the follows: if \mathbf{w}_t is the parameter of the algorithm, we define its complement $\bar{\mathbf{w}}_t$ as $\bar{w}_{t,i} = 1 - w_{t,i}$. The gain of the algorithm at trial t is the inner product between the “gain” vector $\boldsymbol{\ell}_t$ and the complement $\bar{\mathbf{w}}_t$, i.e. $\bar{\mathbf{w}}_t \cdot \boldsymbol{\ell}_t$. Similarly, for any comparator $\mathbf{w} \in \mathcal{S}_m$, we define its gain as $\bar{\mathbf{w}} \cdot \boldsymbol{\ell}_t = \sum_{i=1}^n (1 - w_i) \ell_{t,i}$. It is easy to verify that the regret of the algorithm can be written as the difference between the largest gain of any subset of k experts and the gain of the algorithm:

$$\mathcal{R} = \max_{\mathbf{w} \in \mathcal{S}_k} \sum_{t=1}^T \bar{\mathbf{w}} \cdot \boldsymbol{\ell}_t - \sum_{t=1}^T \bar{\mathbf{w}}_t \cdot \boldsymbol{\ell}_t,$$

where $\mathcal{S}_k = \{\mathbf{w} \in [0, 1]^n : \sum_i w_i = k\}$. At trial one, the adversary first sorts the n experts by their complementary weights and then gives a unit of gain to each of the experts in the second half, i.e. the experts with smaller complementary weights. Since the complementary weights sum to k , the gain of the algorithm is at most $\frac{k}{2}$ in the first trial.

At each of the following trials, the adversary only sorts the experts that received gains in all of the previous trials by their complementary weights. It then gives unit gains to the second half (the half with smaller complementary weights) of these experts. It is easy to see that in this way the gain of the algorithm is at most $\frac{k}{2}$ at each trial.

Note that half of the experts that always receive gain prior to a trial t will receive gain again in trial t . Hence, after $T \leq \log_2 \frac{k}{2}$ trials, there will be at least k experts that received gains in all of the T trials, which means that the total gain of the best k experts is Tk . Now, since the algorithm receives total gain at most $\frac{kT}{2}$ in T trials, the theorem follows. ■

Appendix H. Proof of Theorem 5.6

The following theorem gives a regret lower bound that is expressed as a function of the loss budget B_L . This lower bound holds for any online algorithm that solves the m -set problem

with either unit bit vectors or arbitrary bit vectors as loss vectors. The proof is based on the time dependent regret lower bounds proven in the previous appendices.

Theorem 5.6 For the m set problem with either unit bit vectors or arbitrary bit vectors, any online algorithm suffers worst case regret of at least $\Omega(\max\{\sqrt{B_L m \ln(n/m)}, m \ln(n/m)\})$.

Proof It suffices to prove the lemma for unit bit vectors. The lower bound $\Omega(m \ln(n/m))$ follows directly from Lemma G.2 by setting the variable i of the lemma to k .

What is left to show is the lower bound $\Omega(\sqrt{B_L m \ln(n/m)})$ when it dominates the bound $\Omega(m \ln(n/m))$, i.e. when $B_L = \Omega(m \ln \frac{n}{m})$. Thus, we assume $B_L \geq m \log_2 \frac{n}{m} + 1$ and we construct an instance sequence of loss budget B_L incurring regret at least $\Omega(\sqrt{B_L m \ln(n/m)})$ to any algorithm. This instance sequence is constructed via Theorem 5.1 and Theorem 5.2: For any algorithm, these theorems provide a sequence of T unit bit vectors that incurs regret at least $\Omega(m \sqrt{\frac{T \ln(n/m)}{n}})$. We apply these theorems with $T = \lfloor \frac{n}{m} B_L \rfloor \geq n \log_2 \frac{n}{m}$. Since the produced sequence consists of unit bit vectors and has length $\lfloor \frac{n}{m} B_L \rfloor$, the total loss of the m best experts is at most B_L . Finally plugging $T = \lfloor \frac{n}{m} B_L \rfloor$ into the regret bounds guaranteed by the theorems results in the regret $\Omega(\sqrt{B_L m \ln(n/m)})$. ■

Appendix I. Auxiliary Lemmas

Lemma I.1 Inequality $\max\{\min\{a, b\}, c\} \geq \min\{\max\{a, c\}, b\}$ holds for any real number a, b and c .

Proof If $c \geq \max\{a, b\}$, LHS is c and RHS is b . Hence, the inequality holds. If $a \geq c \geq b$ or $b \geq c \geq a$, LHS is c while RHS is at most c . If $c \leq a$ and $c \leq b$, both sides are $\min\{a, b\}$. ■

Lemma I.2 Let $X \sim \text{Binomial}(T, p)$. If $Tp \geq 8c$ for any positive constant c , then $\mathbb{E}[\sqrt{X}] \geq \frac{\sqrt{c}}{\sqrt{2(1+c)}} \sqrt{Tp}$.

Proof We use the following form of the Chernoff bound (DeGroot and Schervish, 2002):

$$\Pr(X \leq Tp - \delta) \leq e^{-\frac{\delta^2}{2Tp}}.$$

Setting $\delta = \frac{1}{2}Tp$, we have $\Pr(X \leq \frac{1}{2}Tp) \leq e^{-Tp/8} \leq e^{-c}$. Since for $c > 0$, $\log(c) \leq c - 1$, this implies $e^{-c} \leq \frac{1}{1+c}$, so that we further have $\Pr(X \leq \frac{1}{2}Tp) \leq \frac{1}{1+c} = 1 - \frac{c}{1+c}$. Now we

calculate $\mathbb{E}[\sqrt{X}]$ from its definition,

$$\begin{aligned} \mathbb{E}[\sqrt{X}] &= \sum_{x=0}^T \Pr(X=x)\sqrt{x} \geq \sum_{x=\lfloor \frac{Tp}{2} \rfloor+1}^T \Pr(X=x)\sqrt{x} \\ &\geq \sum_{x=\lfloor \frac{Tp}{2} \rfloor+1}^T \Pr(X=x)\sqrt{\left\lfloor \frac{Tp}{2} \right\rfloor+1} \\ &= \Pr(X > \tfrac{1}{2}Tp) \sqrt{\left\lfloor \frac{Tp}{2} \right\rfloor+1} \\ &\geq \frac{c}{\sqrt{2(1+c)}} \sqrt{Tp}. \end{aligned}$$

■

References

- Jacob Abernethy, Manfred K. Warmuth, and Joel Yellin. When random play is optimal against an adversary. In *COLT*, pages 437–446, 2008.
- Jacob Abernethy, Alekh Agarwal, Peter L. Bartlett, and Alexander Rakhlin. A stochastic view of optimal regret through minimax duality. In *COLT*, pages 56–64, 2009.
- Raman Arora, Andrew Cotter, and Nati Srebro. Stochastic optimization of PCA with capped MSG. In *NIPS*, pages 1815–1823, 2013.
- Jean-Yves Audibert and Sébastien Bubeck. Regret bounds and minimax policies under partial monitoring. *Journal of Machine Learning Research*, 11:2785–2836, 2010.
- Katy S. Azoury and Manfred K. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211–246, 2001.
- Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006. ISBN 978-0-521-84108-5.
- Nicolò Cesa-Bianchi, Philip M. Long, and Manfred K. Warmuth. Worst-case quadratic loss bounds for prediction using linear functions and gradient descent. *IEEE Trans. Neural Netw. Learning Syst.*, 7(3):604–619, 1996.
- Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *Journal of the ACM*, 44(3):427–485, 1997.
- Steven de Rooij, Tim van Erven, Peter D. Grünwald, and Wouter M. Koolen. Follow the leader if you can, hedge if you must. *Journal of Machine Learning Research*, 15: 1281–1316, 2014.
- Morris H. DeGroot and Mark J. Schervish. *Probability and Statistics*. Addison-Wesley series in statistics. Addison-Wesley, 2002. ISBN 9780201524888.
- Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *EuroCOLT*, pages 23–37, 1995.
- Dan Garber, Elad Hazan, and Tengyu Ma. Online learning of eigenvectors. In *ICML*, pages 560–568, 2015.
- Elad Hazan, Satyen Kale, and Manfred K. Warmuth. On-line variance minimization in $o(n^2)$ per trial? In *COLT*, pages 314–315, 2010.
- David P. Helmbold and Manfred K. Warmuth. Learning permutations with exponential weights. *Journal of Machine Learning Research*, 10:1705–1736, 2009.
- Mark Herbster and Manfred K. Warmuth. Tracking the best linear predictor. *Journal of Machine Learning Research*, 1:281–309, 2001.
- Jyrki Kivinen and Manfred K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Inf. Comput.*, 132(1):1–63, 1997.
- Wouter M. Koolen. *Combining Strategies Efficiently: High-quality Decisions from Conflicting Advice*. PhD thesis, Institute of Logic, Language and Computation (ILLC), University of Amsterdam, 2011.
- Wouter M. Koolen, Manfred K. Warmuth, and Jyrki Kivinen. Hedging structured concepts. In *COLT*, pages 93–105, 2010.
- Wojciech Kotłowski and Manfred K. Warmuth. PCA with Gaussian perturbation. Private communication, 2015.
- Arkadi Nemirovski and D. Yudin. On Cesaro’s convergence of the gradient descent method for finding saddle points of convex-concave functions. *Doklady Akademii Nauk*, 4(249): 249, 1978.
- Jiazhong Nie, Wojciech Kotłowski, and Manfred K. Warmuth. Online PCA with optimal regrets. In *ALT*, pages 98–112, 2013.
- Shai Shalev-Shwartz and Yoram Singer. A primal-dual perspective of online learning algorithms. *Machine Learning*, 69(2-3):115–142, 2007.
- Nathan Srebro, Karthik Sridharan, and Ambuj Tewari. On the universality of online mirror descent. In *NIPS*, pages 2645–2653, 2011.
- Karthik Sridharan and Ambuj Tewari. Convex games in Banach spaces. In *COLT*, pages 1–13, 2010.
- Koji Tsuda, Gunnar Rätsch, and Manfred K. Warmuth. Matrix Exponential Gradient updates for on-line learning and Bregman projection. *Journal of Machine Learning Research*, 6:995–1018, 2005.

- Tim van Erven, Peter Grünwald, Wouter M. Koolen, and Steven de Rooij. Adaptive Hedge. In *NIPS*, pages 1656–1664, 2011.
- Maufred K. Warmuth and Dima Kuzmin. Randomized online PCA algorithms with regret bounds that are logarithmic in the dimension. *Journal of Machine Learning Research*, 9: 2287–2320, 2008.
- Maufred K. Warmuth and S. V. N. Vishwanathan. Leaving the span. In *COLT*, pages 366–381, 2005.
- Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.

Efficient Computation of Gaussian Process Regression for Large Spatial Data Sets by Patching Local Gaussian Processes

Chiwoo Park

*Department of Industrial and Manufacturing Engineering
Florida State University
2525 Pottsdamer St., Tallahassee, FL 32310-6046, USA*

CPARK5@FSU.EDU

Jianhua Z. Huang

*Department of Statistics
Texas A&M University
3143 TAMU, College Station, TX 77843-3143, USA*

JIANHUA@STAT.TAMU.EDU

Editor: Manfred Opper

Abstract

This paper develops an efficient computational method for solving a Gaussian process (GP) regression for large spatial data sets using a collection of suitably defined local GP regressions. The conventional local GP approach first partitions a domain into multiple non-overlapping local regions, and then fits an independent GP regression for each local region using the training data belonging to the region. Two key issues with the local GP are (1) the prediction around the boundary of a local region is not as accurate as the prediction at interior of the local region, and (2) two local GP regressions for two neighboring local regions produce different predictions at the boundary of the two regions, creating undesirable discontinuity in the prediction. We address these issues by constraining the predictions of local GP regressions sharing a common boundary to satisfy the same boundary constraints, which in turn are estimated by the data. The boundary constrained local GP regressions are solved by a finite element method. Our approach shows competitive performance when compared with several state-of-the-art methods using two synthetic data sets and three real data sets.

Keywords: constrained Gaussian process regression, kriging, local regression, boundary value problem, spatial prediction, variational problem

1. Introduction

Within its origin in geostatistics and known as kriging, the Gaussian process regression (hereafter abbreviated as GP regression) has been developed to be a useful tool in machine learning (Rasmussen and Williams, 2005). It provides the best linear unbiased prediction computable by a simple closed-form expression, which also has a nice probabilistic interpretation (MacKay, 1998). However, computing the exact solution of a GP regression requires $O(N^3)$ operations when the number of data points is N , which is more than 10,000 or 100,000 in a typical geospatial data set. Such a computational complexity is prohibitively

high for data sets of large size. The purpose of this paper is to develop a new computational method to expedite the computation of GP regression for large data sets.

The computation issue for GP regression has received much attention in machine learning and spatial statistics. Since a major computation bottleneck for a GP regression is the inversion of a big sample covariance matrix of size $N \times N$, many approaches proposed to approximate the sample covariance matrix with a more easily invertible one. Covariance tapering (Furrer et al., 2006; Kaufman et al., 2008) tapers the original covariance function to make a sample covariance matrix sparser and applies the sparse matrix computation algorithms for faster inversion of the matrix. Low-rank approximation (Seeger et al., 2003; Snelson and Ghahramani, 2006; Cressie and Johannesson, 2008; Banerjee et al., 2008; Sang and Huang, 2012) introduces M latent variables and assumes a certain conditional independence given the latent variables, which reduces the rank of the resulting sample covariance matrix to M . The approximation of a Gaussian random field by a Gaussian Markov random field has also been proposed (Lindgren et al., 2011). When a covariance matrix of the approximated Gaussian random field is a Matérn covariance function, the sparse precision matrix for the Gaussian Markov random field can be explicitly constructed, and the approximation can be efficiently computed.

On the other hand, local GP regression partitions a regression domain into local regions, and an independent GP regression model is learned for each local region. Since the number of observations belonging to a local region is much smaller than the total number of observations, the resulting sample covariance matrix for the local GP regression becomes much smaller. However, because of the independence of the local GP regressions, two local GP regressions for two neighboring local regions produce different predictions at the boundary of the two regions. This discontinuity in prediction is not acceptable in applications. Many proposed methods have combined local GP regressions into a global model. A popular approach is to take a mixture of local GP regressions through a Dirichlet mixture (Rasmussen and Ghahramani, 2002), a tree mixture (Gramacy and Lee, 2008), Bayesian model averaging (Tresp, 2000; Chen and Ren, 2009; Deisenroth and Ng, 2015), or a locally weighted projection (Nguyen-Tuong et al., 2009). Another approach is to use multiple additive covariance functions of a global covariance and a local covariance (Snelson and Ghahramani, 2007; Vanhatalo and Vehtari, 2008), to simply construct a new local model for each testing location (Gramacy and Apley, 2015).

Domain decomposition method (DDM, Park et al., 2011) is a specific local GP regression method that attempts to constrain the prediction of local GP regressions to be equal at their shared domain boundaries. DDM was shown in the original paper to numerically outperform several existing local GP methods in terms of computational cost and prediction accuracy. Our proposed approach in this paper follows and advances DDM in several aspects. In particular, we improve the way of constraining the prediction at boundaries of local regions. In the original DDM paper, the predictions of two local GP regressions for two neighboring local regions were constrained to be equal only at a finite number of locations on the boundary of the two regions, so there is no guarantee that the predictions are the same at other boundary locations. Our new approach considers a variational formulation for a collection of boundary constrained local GP regressions to ensure that the predictions of the two local GP regressions for neighboring regions are the same for all points on the shared boundary. The variational formulation allows us to solve the boundary

constrained local GP regressions using the finite element method. This is mathematically more elegant and conceptually simpler than the previously somewhat ad hoc treatment. In addition, we significantly improve the DDM by proposing two approaches for estimating the boundary constraints (i.e., boundary values of local regions). The improved accuracy of estimating these constraints leads to better prediction accuracy of our constrained local GP regressions. Last, our new approach has better numerical stability than the DDM. It was previously reported that the predictive variance estimate of the DDM can be negative for some numerical examples (Ponhahitb et al., 2014). Since the expression of the DDM’s predictive variance estimate cannot be theoretically negative, the negative estimate is due to numerical issues. Our new approach provides positive predictive variances for all the examples. The computation speed of the new approach is comparable to the DDM. When the domain in \mathbb{R}^d is partitioned into S local regions and each local region has N_S training data points, the computational cost of the proposed method is $O(NN_S^2 + dS)$, where $N_S \ll N$. Since our method can be viewed as “patching” a collection of local GP regressions, we refer to our method as *patched Gaussian Process regression* or *patched GP* for short.

The proposed patched GP method is theoretically applicable for an arbitrary input dimension d , but the implementation of the approach may be practically difficult for $d > 2$ mainly due to hardness in generating finite element meshes for high dimensions. Therefore, the practical application of the proposed approach would be a GP regression with a spatial data set of large volume (i.e., the data fall in a domain in \mathbb{R}^2), which finds broad applications in spatial statistics and remote sensing (Stein, 2012; Curran and Atkinson, 1998). We will still describe and derive our approach for a general dimension to ease a possible future extension of the approach to high dimensional problems. As a byproduct of this work, we develop a finite element method for the boundary constrained GP regression problem, which may have potential applications in GP solutions for partial differential equations (Graepel, 2003) or for linear stochastic differential equations having boundary conditions (Steinke and Schölkopf, 2008).

The rest of the paper is organized as follows. Section 2 reformulates the GP regression as an optimization problem and also provides an equivalent variational formulation. Section 3 considers the boundary constrained GP regression problem through optimization and develops a finite element method for solving the equivalent variational problem. Section 4 presents the core methodology of the proposed patched GP method for efficient computation of GP regressions, including a finite element method for solving a boundary constrained local GP regression, estimation of the boundary constraints, and estimation of the parameters of the Gaussian process. Section 5 shows the numerical performance of the patched GP method for different tuning parameters, and compares it to the full GP regression (i.e., full implementation of GP regression without approximation) and its precursor, the DDM, using both synthetic and real data sets. Section 6 provides additional numerical comparisons of the patched GP with several state-of-the-art approaches using real data sets. Finally Section 7 concludes the paper.

2. Reformulation of Gaussian Process Regression as An Optimization Problem

A GP regression is formulated as follows: given a training set $\mathcal{D} = \{(x_n, y_n), n = 1, \dots, N\}$ of N pairs of inputs x_n and noisy outputs y_n of a latent function f , obtain the predictive distribution of f at a test location x_* , denoted by $f_* = f(x_*)$. We assume that the latent function comes from a zero-mean Gaussian process with a covariance function $k(\cdot, \cdot)$ and the noisy observations y_i are given by

$$y_i = f(x_i) + \epsilon_i, \quad i = 1, \dots, N,$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ are white noises independent of $f(x_i)$. Denote $\mathbf{x} = [x_1, x_2, \dots, x_N]^T$ and $\mathbf{y} = [y_1, y_2, \dots, y_N]^T$. The joint distribution of (f_*, \mathbf{y}) is

$$P(f_*, \mathbf{y}) = \mathcal{N} \left(\mathbf{0}, \begin{bmatrix} k_{**} & \mathbf{k}'_{**} \\ \mathbf{k}_{**} & \sigma^2 \mathbf{I} + \mathbf{K}_{\mathbf{xx}} \end{bmatrix} \right),$$

where $k_{**} = k(x_*, x_*)$, $\mathbf{k}_{**} = (k(x_1, x_*), \dots, k(x_N, x_*))'$ and $\mathbf{K}_{\mathbf{xx}}$ is an $N \times N$ matrix with $(i, j)^{th}$ entry $k(x_i, x_j)$. The subscripts of k_{**} , \mathbf{k}_{**} , and $\mathbf{K}_{\mathbf{xx}}$ represent two sets of locations between which the covariance is computed, and x_* is abbreviated as $*$. The predictive distribution of f_* given \mathbf{y} is

$$P(f_* | \mathbf{y}) = \mathcal{N}(\mathbf{k}'_{**}(\sigma^2 \mathbf{I} + \mathbf{K}_{\mathbf{xx}})^{-1} \mathbf{y}, k_{**} - \mathbf{k}'_{**}(\sigma^2 \mathbf{I} + \mathbf{K}_{\mathbf{xx}})^{-1} \mathbf{k}_{**}). \quad (1)$$

The predictive mean $\mathbf{k}'_{**}(\sigma^2 \mathbf{I} + \mathbf{K}_{\mathbf{xx}})^{-1} \mathbf{y}$ gives the point prediction of $f(x)$ at location x_* , whose uncertainty is measured by the predictive variance $k_{**} - \mathbf{k}'_{**}(\sigma^2 \mathbf{I} + \mathbf{K}_{\mathbf{xx}})^{-1} \mathbf{k}_{**}$. Efficient calculation of the predictive mean and variance has been the focus of much research.

The predictive mean and variance can be derived using the viewpoint of the best linear unbiased predictor (BLUP) as follows. Consider all linear predictors

$$\mu(x_*) = \mathbf{u}(x_*)' \mathbf{y}, \quad (2)$$

which automatically satisfy the unbiasedness requirement $E[\mu(x_*)] = 0$ since all random variables y_i have zero mean. We seek an N -dimensional vector $\mathbf{u}(x_*)$ such that the mean squared prediction error $E[\mu(x_*) - f(x_*)]^2$ is minimized. Since $E[\mu(x_*)] = 0$ and $E[f(x_*)] = 0$, the mean squared prediction error equals the error variance $\text{var}[\mu(x_*) - f(x_*)]$ and can be expressed as

$$\begin{aligned} \sigma(x_*) &= \mathbf{u}(x_*)' E(\mathbf{y} \mathbf{y}') \mathbf{u}(x_*) - 2 \mathbf{u}(x_*)' E(\mathbf{y} f_*) + E(f_*^2) \\ &= \mathbf{u}(x_*)' (\sigma^2 \mathbf{I} + \mathbf{K}_{\mathbf{xx}}) \mathbf{u}(x_*) - 2 \mathbf{u}(x_*)' \mathbf{k}_{**} + k_{**}, \end{aligned} \quad (3)$$

which is a quadratic form in $\mathbf{u}(x_*)$. It is easy to see $\sigma(x_*)$ is minimized if and only if $\mathbf{u}(x_*)$ is chosen to be $(\sigma^2 \mathbf{I} + \mathbf{K}_{\mathbf{xx}})^{-1} \mathbf{k}_{**}$; moreover, the minimal value of $\sigma(x_*)$ equals the predictive variance given in (1).

The BLUP view of GP regression suggests a reformulation of GP regression as the following optimization problem:

$$\begin{aligned} \text{Minimize}_{\mathbf{u}(x_*) \in \mathbb{R}^N} \quad & z[\mathbf{u}(x_*)] = \frac{1}{2} \mathbf{u}(x_*)' \mathbf{A} \mathbf{u}(x_*) - \mathbf{f}(x_*)' \mathbf{u}(x_*), \end{aligned} \quad (4)$$

where $\mathbf{A} = (\sigma^2 \mathbf{I} + \mathbf{K}_{\mathbf{xx}})$ is an $N \times N$ positive definite matrix, and $\mathbf{f}(x_*) = \mathbf{k}_{\mathbf{x}*}$ is a $N \times 1$ vectorial function. Note that the objective function in (4) equals half of the error variance of a linear predictor given in (3) subtracting the constant term $k_{**}/2$. The one half factor is introduced here to make subsequent formulas neat. The solution of (4) is $\mathbf{u}^l(x_*) = \mathbf{A}^{-1} \mathbf{k}_{\mathbf{x}*} = (\sigma^2 \mathbf{I} + \mathbf{K}_{\mathbf{xx}})^{-1} \mathbf{k}_{\mathbf{x}*}$. Back to the GP regression problem, the predictive mean is given by $\mathbf{u}^l(x_*)^T \mathbf{y}$ and the predictive variance is $2z[\mathbf{u}^l(x_*) + k_{**}] + k_{**}$, twice the optimal objective value plus the variance of f_* at the location x_* .

Usually we are interested in obtaining the predictive mean and variance at multiple locations in a domain $\Omega \subset \mathbb{R}^N$, where all training locations x_i 's also belong to. To consider the prediction at all locations in Ω , we consider the following optimization problem:

$$\underset{\mathbf{u}(\cdot) \in [L^2(\Omega)]^N}{\text{Minimize}} \quad J(\mathbf{u}) = \int_{\Omega} \left\{ \frac{1}{2} \mathbf{u}(x)^T \mathbf{A} \mathbf{u}(x) - \mathbf{f}(x)^T \mathbf{u}(x) \right\} dx, \quad (5)$$

where $[L^2(\Omega)]^N$ is the Cartesian product of N Hilbert spaces $L^2(\Omega)$. The objective function here is simply the integration of the objective function in the single location problem (4). Since the optimal solution $\mathbf{u}^l(x_*)$ obtains the minimum objective value of (4) at every location $x_* \in \Omega$, $\mathbf{u}^l(x_*)$ as a function of x_* also solves the global problem (5).

According to a standard result from functional analysis (Ern and Guermond, 2004), the problem (5) has an equivalent variational formulation, as follows.

Proposition 1 *The vector $\mathbf{u} \in [L^2(\Omega)]^N$ minimizes $J(\mathbf{u})$ if and only if it solves the integral equation*

$$\int_{\Omega} \mathbf{u}(x)^T \mathbf{A} \mathbf{v}(x) dx = \int_{\Omega} \mathbf{f}(x)^T \mathbf{v}(x) dx \quad \text{for each } \mathbf{v} \in [L^2(\Omega)]^N. \quad (6)$$

The proof is given in Appendix A.

Throughout the rest of the paper, whenever no confusion may arise and to alleviate the notation, we omit the Lebesgue measure under the integral sign. For example, we shall write $\int_{\Omega} \mathbf{u}^T \mathbf{A} \mathbf{v}$ and $\int_{\Omega} \mathbf{f}^T \mathbf{v}$ instead of $\int_{\Omega} \mathbf{u}(x)^T \mathbf{A} \mathbf{v}(x) dx$ and $\int_{\Omega} \mathbf{f}(x)^T \mathbf{v}(x) dx$.

3. Gaussian Process Regression with Boundary Constraints

The optimization problem (5) was introduced in previous section as a reformulation of the GP regression. Now we introduce boundary constraints to the GP regression through this optimization problem and develop a finite element solution for the corresponding variational formulation. This finite element solution will serve as a building block in the next section for the patched GP regression, which consists of a collection of boundary constrained GP regressions.

3.1 Constrained Optimization Problem and Its Variational Formulation

We require that the domain Ω is a Lipschitz bounded open set. Let $\partial\Omega$ denote the boundary of the domain Ω . We need to restrict our attention to smooth functions and specifically consider only solution vectors in the Sobolev space $[H^1(\Omega)]^N$ instead of $[L^2(\Omega)]^N$, where $H^1(\Omega) := \{u \in L^2(\Omega); \partial_i u \in L^2(\Omega), 1 \leq i \leq d\}$ with $\partial_i u$ denoting the partial derivative of u with respect to the i th dimension of input. We cannot use the bigger $L^2(\Omega)$ space because

the value of a function in this space can be unbounded on $\partial\Omega$ and so not well-defined. For example, take $\Omega = (0, 1)$ and $u(x) = x^{-1/3}$. It is clear that $u(x) \in L^2(\Omega)$ but $u(0) = \infty$. On the other hand, the values of a function in $H^1(\Omega)$ are bounded at its domain boundary. In fact, if $u \in H^1(\Omega)$, then u restricted on $\partial\Omega$, which we denote by $u|_{\partial\Omega}$, is a member of $H^{1/2}(\partial\Omega) := \{u \in L^2(\Omega) : \frac{u(x) - u(y)}{\|x - y\|^{(d+1)/2}} \in L^2(\Omega \times \Omega)\}$ (Ern and Guermond, 2004, Theorem B.52).

Consider the boundary constraints of the form $\mathbf{y}^T \mathbf{u}(x) = b(x)$ for some known function $b \in H^{1/2}(\partial\Omega)$. Since $\mathbf{y}^T \mathbf{u}(x)$ can be interpreted as a linear predictor at location x , the constraints simply require the predictions at the domain boundary to have certain specified functional form. Denote

$$H_b = \left\{ \mathbf{u} \in [H^1(\Omega)]^N : \int_{\partial\Omega} \mathbf{y}^T \mathbf{u}|_{\partial\Omega} v = \int_{\partial\Omega} b v \quad \text{for each } v \in H^{1/2}(\partial\Omega) \right\}.$$

A constrained version of the optimization problem (5) can be written as

$$\underset{\mathbf{u} \in H_b}{\text{Minimize}} \quad J(\mathbf{u}) = \int_{\Omega} \left\{ \frac{1}{2} \mathbf{u}(x)^T \mathbf{A} \mathbf{u}(x) - \mathbf{f}(x)^T \mathbf{u}(x) \right\} dx. \quad (7)$$

Here, for mathematical convenience, we have replaced the strict boundary constraints $\mathbf{y}^T \mathbf{u}(x) = b(x)$ by a weaker form for $\mathbf{u} \in H_b$.

Similar to Proposition 1, we can show that the optimization problem (7) is equivalent to a variational formulation:

Proposition 2 *The vector $\mathbf{u} \in H_b$ minimizes $J(\mathbf{u})$ if and only if it solves the integral equation*

$$\int_{\Omega} \mathbf{u}(x)^T \mathbf{A} \mathbf{v}(x) dx = \int_{\Omega} \mathbf{f}(x)^T \mathbf{v}(x) dx \quad \text{for each } \mathbf{v} \in H_b. \quad (8)$$

The proof is given in Appendix B.

3.2 Finite Element Approximation

A finite element method approximates the space H_b of vector-valued functions on a domain Ω by a finite dimensional vector space. With the approximation, the integral equation (8) is converted to a finite-dimensional linear system of equations.

The finite dimensional approximation scheme requires a mesh and a set of finite elements. A mesh $\mathcal{K}_h = \{K_1, \dots, K_M\}$ is a set of a finite number of compact, connected and Lipschitz subsets of Ω with non-empty interior which partitions Ω , where each K_m is called a mesh cell, and h parameterizes the size of K_m ; e.g. if K_m is a polygon, h is the length of the polygon's side. For each $K_m \in \mathcal{K}_h$, a finite element is defined as a triplet $\{K_m, P_m, \mathcal{A}_m\}$, where P_m is a vector space of functions $q : K_m \rightarrow \mathbb{R}$ with $\dim(P_m) = p$, and \mathcal{A}_m is a set of p linear forms $\alpha_{mj} : P_m \rightarrow \mathbb{R}^N$ spanning the dual vector space of P_m , which is called the local degrees of freedom. There exists a basis $\{\phi_{m1}, \dots, \phi_{mp}\}$ of the vector space P_m satisfying $\alpha_{mj}(\phi_{mk}) = \mathbf{I}_N \delta_{jk}$, where \mathbf{I}_N is a N -vector of 1's. In our implementation, we use the Lagrange finite element for $\{K_m, P_m, \mathcal{A}_m\}$, where K_m is a simplex in \mathbb{R}^d , P_m is the polynomial of order k and $\alpha_{mj} = \mathbf{u}(x_{mj})$ with $x_{mj} \in K_m$; more details can be found in Ern and Guermond (2004, Section 1.2.3).

For any vector-valued function $\mathbf{u} \in [H^1(\Omega)]^N$, let $\mathbf{u}|_{K_m}$ denote its restriction to K_m , i.e., $\mathbf{u}|_{K_m}(x) = \mathbf{1}_{K_m}(x)\mathbf{u}(x)$. The finite element approximation of \mathbf{u} on K_m is given by

$$\mathbf{u}|_{K_m} \approx \mathbf{u}_{mh} := \sum_{j=1}^p \beta_{mj} \phi_{mj},$$

where β_{mj} is an N -vector of coefficients in the basis expansion. The combination of \mathbf{u}_{mh} 's over K_m , $m \in \mathcal{M}$, provides a global approximation of \mathbf{u} over the whole domain Ω , i.e.,

$$\mathbf{u}_h = \sum_{m=1}^M \mathbf{u}_{mh} = \sum_{m=1}^M \sum_{j=1}^p \beta_{mj} \phi_{mj}, \quad (9)$$

which has a matrix-vector representation

$$\mathbf{u}_h = \mathbf{U}^T \phi, \quad (10)$$

where \mathbf{U} is the $(Mp) \times N$ matrix formed by concatenating row vectors β_{mj} in row-wise and ϕ is the (Mp) -dimensional column vector of ϕ_{mj} 's. In (9) and (10) we explicitly extended ϕ_{mj} to be zero outside K_m . The collection of vector-valued functions with expression (10) is a finite-dimensional linear space. We call this space the finite element space and denote it as G_h , where the subscript denotes the mesh size h .

The finite element method for solving the unconstrained integral equation (6) seeks $\mathbf{u}_h \in G_h$ such that

$$a(\mathbf{u}_h, \mathbf{v}_h) = c(\mathbf{v}_h) \quad \text{for each } \mathbf{v}_h \in G_h, \quad (11)$$

where $a(\mathbf{u}, \mathbf{v}) = \int \mathbf{u}' \mathbf{A} \mathbf{v}$ and $c(\mathbf{v}) = \int \mathbf{f}' \mathbf{v}$. Following (10), we can write $\mathbf{u}_h = \mathbf{U}^T \phi$ and similarly $\mathbf{v}_h = \mathbf{V}^T \phi$. Both of the lhs and rhs of (11) can be simply represented as algebraic forms as follows. First, since

$$\mathbf{u}_h' \mathbf{A} \mathbf{v}_h = \text{trace}(\phi' \mathbf{U} \mathbf{A} \mathbf{V}^T \phi) = \text{trace}(\phi \phi' \mathbf{U} \mathbf{A} \mathbf{V}^T),$$

we have

$$a(\mathbf{u}_h, \mathbf{v}_h) = \int_{\Omega} \mathbf{u}_h' \mathbf{A} \mathbf{v}_h = \text{trace} \left(\int_{\Omega} \phi \phi' \mathbf{U} \mathbf{A} \mathbf{V}^T \right) = \text{trace}(\Phi \mathbf{U} \mathbf{A} \mathbf{V}^T),$$

where $\Phi = \int_{\Omega} \phi \phi'$. Second, since

$$\mathbf{f}' \mathbf{v}_h = \text{trace}(\mathbf{f}' \mathbf{V}^T \phi) = \text{trace}(\phi \mathbf{f}' \mathbf{V}^T),$$

we have that

$$c(\mathbf{v}_h) = \int_{\Omega} \mathbf{f}' \mathbf{v}_h = \text{trace} \left(\int_{\Omega} \phi \mathbf{f}' \mathbf{V}^T \right) = \text{trace}(\mathbf{F} \mathbf{V}^T),$$

where $\mathbf{F} = \int_{\Omega} \phi \mathbf{f}'$. Therefore, the integral equation (11) is equivalent to the following linear system

$$\text{trace}((\Phi \mathbf{U} \mathbf{A} - \mathbf{F}) \mathbf{V}^T) = 0 \quad \text{for each } \mathbf{V} \in \mathbb{R}^{N \times (Mp)},$$

which is equivalent to

$$\Phi \mathbf{U} = \mathbf{F} \mathbf{A}^{-1}. \quad (12)$$

We now introduce the boundary constraints. We partition (after reordering the elements) the basis function vector ϕ used in $\mathbf{u}_h = \mathbf{U}^T \phi$ into two vectors ϕ_0 and ϕ_b in column-wise such that, ϕ_0 is a column vector of the $\phi_{mj}(x)$'s satisfying $\phi_{mj}|_{\partial\Omega} = 0$ and ϕ_b is a column vector of the $\phi_{mj}(x)$'s satisfying $\phi_{mj}|_{\partial\Omega} \neq 0$. Suppose that ϕ_0 has Q elements and ϕ_b has R elements. Since the number of columns of ϕ is Mp , $Q + R = Mp$. With the partition, we have that

$$\mathbf{u}_h = \mathbf{U}'_0 \phi_0 + \mathbf{U}'_b \phi_b, \quad (13)$$

where \mathbf{U}'_0 and \mathbf{U}'_b are submatrices consisting of \mathbf{U} 's rows corresponding to ϕ_0 and ϕ_b respectively. Substituting (13) into equation (12), we have that

$$\Phi'_0 \mathbf{U}'_0 + \Phi'_b \mathbf{U}'_b = \mathbf{F}' \mathbf{A}^{-1}, \quad (14)$$

where $\Phi'_0 = \int_{\Omega} \phi'_0 \phi'_0$ and $\Phi'_b = \int_{\Omega} \phi'_b \phi'_b$.

The boundary constraints restricted to G_h can be written as

$$\int_{\partial\Omega} \mathbf{y}' \mathbf{u}_h|_{\partial\Omega} v = \int_{\partial\Omega} b v \quad \text{for each } v \in G_h.$$

Using (14) and dropping the basis functions whose values are zero at the boundary, we obtain

$$\int_{\partial\Omega} \mathbf{y}' \mathbf{U}'_b \phi_b|_{\partial\Omega} \phi_{mj}|_{\partial\Omega} = \int_{\partial\Omega} b \phi_{mj}|_{\partial\Omega} \quad \text{for each } \phi_{mj}|_{\partial\Omega} \neq 0,$$

which implies

$$\int_{\partial\Omega} \mathbf{y}' \mathbf{U}'_b \phi_b|_{\partial\Omega} \phi'_b|_{\partial\Omega} = \int_{\partial\Omega} b \phi'_b|_{\partial\Omega}.$$

Letting $\mathbf{B} = \int_{\partial\Omega} \phi_b|_{\partial\Omega} \phi'_b|_{\partial\Omega}$ and $\mathbf{b} = \int_{\partial\Omega} b \phi'_b|_{\partial\Omega}$, we have

$$\mathbf{y}' \mathbf{U}'_b \mathbf{B} = \mathbf{b}', \quad (15)$$

We decompose \mathbf{U}'_b into two components: one orthogonal to \mathbf{y} and the residual. Let \mathbf{O}_g be $N \times (N-1)$ matrix of $N-1$ column vectors orthogonal to \mathbf{y} , which can be obtained by the Gram-Schmidt process. There exist $\mathbf{Z} \in \mathbb{R}^{R \times (N-1)}$ and $\mathbf{z} \in \mathbb{R}^{R \times 1}$ satisfying

$$\mathbf{U}'_b = \mathbf{Z} \mathbf{O}'_g + \mathbf{z} \mathbf{y}'. \quad (16)$$

Therefore, (14) becomes

$$\Phi'_0 \mathbf{U}'_0 + \Phi'_b \mathbf{Z} \mathbf{O}'_g + \Phi'_b \mathbf{z} \mathbf{y}' = \mathbf{F}' \mathbf{A}^{-1} \quad (17)$$

Since

$$\mathbf{y}' \mathbf{U}'_b = \mathbf{y}' (\mathbf{O}_g \mathbf{Z}' + \mathbf{y} \mathbf{z}') = \mathbf{y}' \mathbf{y} \mathbf{z}',$$

equation (15) gives us

$$\mathbf{y}' \mathbf{y} \mathbf{z}' \mathbf{B} = \mathbf{b}',$$

therefore,

$$(\mathbf{y}' \mathbf{y}) \mathbf{z} = \mathbf{B}^{-1} \mathbf{b}'. \quad (18)$$

In summary, we first solve (18) for \mathbf{z} , then solve (17) for \mathbf{U}'_0 and \mathbf{Z} , and then calculate \mathbf{U}'_b using (16). The finite element solution of the variational problem (8) is given by $\mathbf{u}_h = \mathbf{U}'_0 \phi_0 + \mathbf{U}'_b \phi_b$. By the theory of finite element method (Ern and Guermond, 2004), the approximate solution \mathbf{u}_h converges to the solution of the original problem as the mesh size h tends to zero.

4. Patching Constrained Local GPs for Efficient Computation of Global GP regression

This section presents our patched GP regression method. We start from some notations. We partition the domain Ω of the data into small local regions $\{\Omega_s, s = 1, \dots, S\}$ and partition the training data set $\mathcal{D} = \{(x_n, y_n) : n = 1, \dots, N\}$ accordingly into S data sets $\mathcal{D}_s := \{(x_n, y_n) \in \mathcal{D} : x_n \in \Omega_s\}$. We then calculate the local prediction function f_s for the local region Ω_s using the data set \mathcal{D}_s . There are some issues with this localized solution. First, the prediction at around $\partial\Omega_s$ is not as accurate as the prediction at the interior of Ω_s , mainly because of the less number of observations available around $\partial\Omega_s$. In particular, when S becomes large, the boundary regions also increase. Therefore, the inaccuracy at the boundaries $\partial\Omega_s$ can have significant negative effects on the overall prediction accuracy. Second, two local GP regressions from two neighboring local regions Ω_s and Ω_t produce different predictions f_s and f_t at the shared boundary, making the prediction discontinuous on the boundary. This discontinuity is unacceptable since continuity of the prediction is often desired. We propose to impose boundary constraints such that the two neighboring local GP regressions give the same predictions on the shared boundary.

Section 4.1 applies the finite element method of Section 3 to solve a boundary constrained local GP problem for each local region when the boundary constraints are given. Section 4.2 gives two methods for estimating the boundary constraints. Section 4.3 discusses some implementation details, including calculation of the integrations involving the finite elements, and learning parameters of the covariance function of the GP regression.

4.1 Boundary-constrained Local GP

For two neighboring local regions Ω_s and Ω_t , let $\Gamma_{st} = \bar{\Omega}_s \cap \bar{\Omega}_t$ denote the shared boundary, where $\bar{\Omega}_s$ is the closure of Ω_s . The prediction function f specialized on Γ_{st} is denoted by a boundary function $b_{st}(x)$ for $x \in \Gamma_{st}$. For the time being, we assume that $b_{st}(x)$ is known and shall discuss in next section how to estimate $b_{st}(x)$. Fix a domain Ω_s , suppose all its boundary functions $\{b_{st} : \forall t, \Gamma_{st} \neq \emptyset\}$ are known. Consider the following local GP problem on Ω_s

$$\begin{aligned} \text{Minimize}_{\mathbf{u}_s} \quad & J(\mathbf{u}_s) = \int_{\Omega_s} \left\{ \frac{1}{2} \mathbf{u}_s(x)' \mathbf{A}_s \mathbf{u}_s(x) - \mathbf{f}_s(x)' \mathbf{u}_s(x) \right\} dx \\ \text{subject to} \quad & \mathbf{y}'_s \mathbf{u}_s(x) = b_{st}(x) \text{ on } x \in \Gamma_{st}, \quad \forall t, \Gamma_{st} \neq \emptyset \end{aligned} \quad (19)$$

where \mathbf{A}_s , $\mathbf{f}_s(x)$ and \mathbf{y}_s are the localized versions of \mathbf{A} , $\mathbf{f}(x)$ and \mathbf{y} , which are all computed using the data in Ω_s . The constraints used in (19) restrict two local GP prediction functions f_s and f_t to have the same prediction b_{st} on the shared boundary Γ_{st} .

As in Section 3, we replace the boundary constraints by the weak form, approximate \mathbf{u}_s by its finite element approximation $\mathbf{u}_{s,h} = \mathbf{U}'_{s,0} \phi_{s,0} + \mathbf{U}'_{s,b} \phi_{s,b}$, where $\phi_{s,0}$ is a vector of local basis functions ϕ_{mj} 's satisfying $\phi_{mj}(x)|_{\Gamma_{st}} = 0$ for all t 's, and $\phi_{s,b}$ is a vector of local basis functions ϕ_{mj} 's satisfying $\phi_{mj}(x)|_{\Gamma_{st}} \neq 0$ for all t 's, and $\phi_s = (\phi'_{s,0}, \phi'_{s,b})'$. Let N_s be the length of \mathbf{y}_s and \mathbf{O}_{y_s} be $N_s \times (N_s - 1)$ matrix of $N_s - 1$ column vectors orthogonal to \mathbf{y}_s . We can decompose $\mathbf{U}_{s,b}$ into $\mathbf{U}_{s,b} = \mathbf{Z}_s \mathbf{O}'_{y_s} + \mathbf{z}_s \mathbf{y}'_s$. As we derived in Section 3, the finite element solution of the local problem (19) is obtained by solving the following linear

$$\begin{aligned} \text{system of equations for } \mathbf{U}_{s,0}, \mathbf{Z}_s \text{ and } \mathbf{z}_s, \\ \Phi_{s,0} \mathbf{U}_{s,0} + \Phi_{s,b} \mathbf{Z}_s \mathbf{O}'_{y_s} + \Phi_{s,b} \mathbf{z}_s \mathbf{y}'_s = \mathbf{F}_s \mathbf{A}_s^{-1}, \end{aligned} \quad (20)$$

$$\text{where } \Phi_{s,0} = \int_{\Omega_s} \phi_s \phi'_{s,0}, \quad \Phi_{s,b} = \int_{\Omega_s} \phi_s \phi'_{s,b}, \quad \mathbf{F}_s = \int_{\Omega_s} \phi_s \mathbf{f}'_s, \text{ and}$$

$$\mathbf{B}_s = \sum_{t: \Gamma_{st} \neq \emptyset} \int_{\Gamma_{st}} \phi_{s,b} |_{\Gamma_{st}} \phi_{s,b} |_{\Gamma_{st}},$$

$$\mathbf{b}_s = \sum_{t: \Gamma_{st} \neq \emptyset} \int_{\Gamma_{st}} b_{st} |_{\Gamma_{st}} \phi_{s,b} |_{\Gamma_{st}}.$$

Since $b_{st}(x)$ is unknown, we need to estimate \mathbf{b}_s using the procedure to be described in Section 4.2. Using the second equation of (20), we obtain $\mathbf{z}_s = \mathbf{B}_s^{-1} \mathbf{b}_s / \mathbf{y}'_s \mathbf{y}_s$. Substituting this expression of \mathbf{z}_s in the first equation of (20), we obtain

$$\begin{aligned} \Phi_{s,0} \mathbf{U}_{s,0} + \Phi_{s,b} \mathbf{Z}_s \mathbf{O}'_{y_s} &= \mathbf{F}_s \mathbf{A}_s^{-1} - \Phi_{s,b} \mathbf{z}_s \mathbf{y}'_s, \\ &= \mathbf{F}_s \mathbf{A}_s^{-1} - \Phi_{s,b} \frac{\mathbf{B}_s^{-1} \mathbf{b}_s \mathbf{y}'_s}{\mathbf{y}'_s \mathbf{y}_s}. \end{aligned} \quad (21)$$

We then solve this equation for $\mathbf{U}_{s,0}$ and \mathbf{Z}_s , and compute $\mathbf{U}_{s,b} = \mathbf{Z}_s \mathbf{O}'_{y_s} + \mathbf{z}_s \mathbf{y}'_s$. Finally the finite element solution of the constrained local GP problem (19) is $\mathbf{u}_{s,h} = \mathbf{U}'_{s,0} \phi_{s,0} + \mathbf{U}'_{s,b} \phi_{s,b}$.

When the number of mesh cells in each local region is M on average and the number of training data in each domain is N_s , the computation complexity of solving the constrained local GP regression (21) for one local region is $O(N_s^3 + M^2)$. The first part N_s^3 is for inverting \mathbf{A}_j , and the second part is for inverting $\Phi_{s,0}$ and $\Phi_{s,b}$, which is proportional to M^2 because $\Phi_{s,0}$ and $\Phi_{s,b}$ are sparse banded matrices; note that the complexity of solving a linear system with a banded coefficient matrix is proportional to the square of the size of the linear system (Mahmood et al., 1991). The cost per local region is mostly bounded by the cubic term $O(N_s^3)$. The total computational cost for S local regions is thus $O(SN_s^3)$, which also equals to $O(NN_s^2)$.

4.1.1 ILLUSTRATIVE OUTPUT OF THE CONSTRAINED GP FORMULATION

Our constrained local GP regression provides a good approximate to a full GP regression when the value of boundary function b_{st} is close to the mean prediction of the full GP regression at Γ . To show this, we performed a simple simulation study. In the study, we generated a data set of 6,000 noisy observations from a zero-mean Gaussian process with an exponential covariance function,

$$y_i = f(x_i) + \epsilon_i \quad \text{for } i = 1, \dots, 6000,$$

where $x_i \sim \text{Uniform}(0, 10)$ and $\epsilon_i \sim \mathcal{N}(0, 1)$ are independently sampled, and $f(x_i)$ is simulated by the R package `RandomField`. Three hundreds of the observations were randomly sampled to learn a Gaussian process regression (full GP) and the covariance hyperparameters, while the remaining 5,700 were kept for test data. The domain $[0, 10]$ was partitioned into ten local regions of size 1 delineated by boundary points $\{0.0, 0.1, 0.2, 0.3, \dots, 1.0\}$,

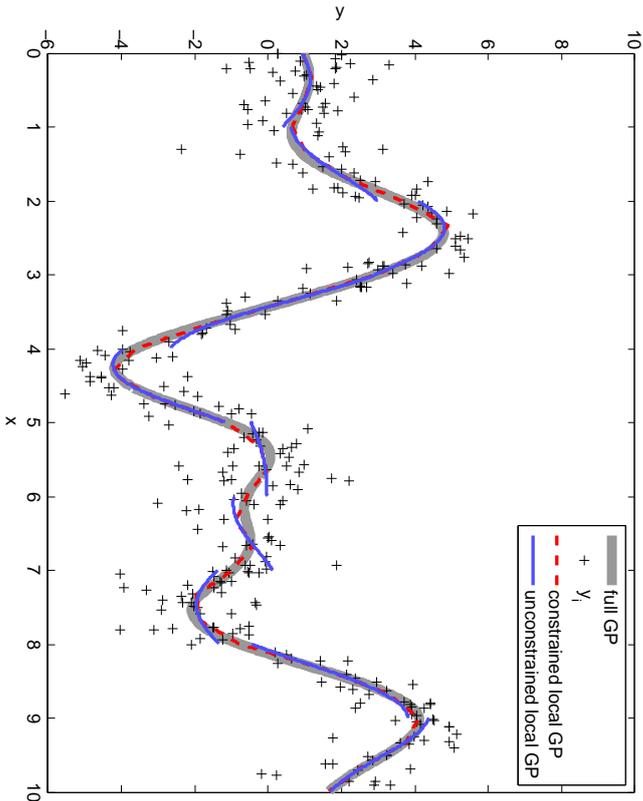


Figure 1: Effect of the boundary constraint on Gaussian process regression.

and the 300 observations were distributed into the local regions accordingly. For each domain, an unconstrained GP regression (unconstrained local GP) and a constrained GP regression (constrained local GP) were learned. When the constrained local GP was learned, the values of the regression outcome at the boundary points were constrained to be equal to the mean prediction of a full GP regression at the points. Note this is not a fair comparison since the full GP prediction was used. This example was just used to show the room for improvement if constraints are used.

Figure 1 shows the comparison of the mean predictions from a full GP, the constrained local GP and the unconstrained local GP regressions. Compared to the unconstrained local GP regression, the constrained model is much closer to a full GP regression especially at the boundary points. The maximum difference in the mean predictions of the constrained model and a full GP over the test data was 0.2803, while that of the unconstrained version was 0.6411. The advantage of placing the boundary constraints on the local GP in improving the prediction accuracy is clear. However, placing the boundary constraints requires knowing the value of the ground truth f at boundary points, i.e., boundary function b_{st} . Estimating the values of f at boundary points is the subject of the next section.

4.2 Estimation of Boundary Values

The prediction function f at Γ_{st} , i.e., the boundary function b_{st} is unknown and needs to be estimated before the constrained local GP regression is solved. We propose two approaches for the estimation boundary values for the local GP regressions. The first approach is to train a separate local GP regression using a subset of training data located around the boundary Γ_{st} —this together with the constrained local GP regressions, leads to a two-step procedure. The second approach is to iteratively solve the boundary value estimation and the constrained local GP problems.

4.2.1 LOCALIZED ESTIMATION

This approach is motivated by our observation that a GP regression for a local domain gives accurate prediction at the center of the domain. We propose to estimate the prediction function f at Γ_{st} by learning a local GP regression with a subset of the training data that belong to a neighborhood of Γ_{st} . When $x \in \mathcal{R}$, Γ_{st} is a point coordinate in \mathcal{R} , and its neighborhood is defined by an interval $[\Gamma_{st}-r, \Gamma_{st}+r]$ around it with half width $r > 0$. When $x \in \mathcal{R}^d$ in general, Γ_{st} is a $d-1$ dimensional hyperplane within \mathcal{R}^d , and its neighborhood is defined by $Nh_r(\Gamma_{st})$,

$$Nh_r(\Gamma_{st}) = \{x' \in \mathcal{R}^d; \min_{x \in \Gamma_{st}} \|x' - x\|_2 \leq r\}.$$

The value of the prediction function f at $x \in \Gamma_{st}$, i.e. $b_{st}(x)$, is estimated by the mean prediction of the local GP regression built from a subset of training data, $\mathbf{x}_{st} = \{x_n \in \mathcal{D}; x_n \in Nh_r(\Gamma_{st})\}$ and the corresponding observed outputs \mathbf{y}_{st} ,

$$\hat{b}_{st}(x) = \mathbf{K}_{\mathbf{x}_{st}, x}^*(\sigma^2 \mathbf{I} + \mathbf{K}_{\mathbf{x}_{st}, \mathbf{x}_{st}})^{-1} \mathbf{y}_{st}. \quad (22)$$

When the average number of observations in the local neighborhood $Nh_r(\Gamma_{st})$ is N_B , the complexity of this boundary estimation per boundary is $O(N_B^3)$. When the dimension of the domain Ω is d and the domain is decomposed into S local regions of d -simplices, the total number of the boundaries in between the local regions is proportional to dS . So, the complexity of this boundary estimation procedure is $O(dSN_B^3)$.

4.2.2 BLOCK GAUSS-SEIDEL ITERATION

The system of equations for the constrained local GP regression given by (20) is converted into the following equation for three unknown variables $\mathbf{U}_{s,0}$, \mathbf{Z}_s and \mathbf{b}_{s1} ,

$$\Phi_{s,0} \mathbf{U}_{s,0} + \Phi_{s,\delta} \mathbf{Z}_s \mathbf{O}'_s + \Phi_{s,\delta} \mathbf{B}_s^{-1} \mathbf{b}_{s1} \mathbf{g}'_s / (\mathbf{g}'_s \mathbf{g}_s) = \mathbf{F}_s \mathbf{A}_s^{-1}, \quad (23)$$

where we used $(\mathbf{g}'_s \mathbf{g}_s) \mathbf{z}_s = \mathbf{B}_s^{-1} \mathbf{b}_s$ to replace the \mathbf{z}_s in the first line of (20) with $\mathbf{B}_s^{-1} \mathbf{b}_s / (\mathbf{g}'_s \mathbf{g}_s)$. Note that the above equation depends on an unknown boundary function b_{st} only through \mathbf{b}_{s1} . We will estimate the vector quantity \mathbf{b}_{s1} instead of estimating the boundary function b_{st} directly. The equation for $\mathbf{U}_{s,0}$, \mathbf{Z}_s and \mathbf{b}_{s1} can be solved iteratively by the block Gauss-Seidel method (Saad, 2003). The block Gauss-Seidel method is an iterative solver for a linear system that partitions a number of unknowns into multiple blocks and solves the linear system for one block at a time while keeping the other blocks fixed. In our problem,

we have two block of unknowns, one block for $\mathbf{U}_{s,0}$ and \mathbf{Z}_s and the other block for \mathbf{b}_s . The corresponding block Gauss-Seidel iteration is as follows. Start with an initial guess $\mathbf{b}_s^{(0)}$. We used a zero vector for the initial guess. At iteration k , we perform the following two steps sequentially:

Step 1. With $\mathbf{b}_s^{(k-1)}$ fixed from the previous iteration, obtain $\mathbf{U}_{s,0}^{(k)}$ and $\mathbf{Z}_s^{(k)}$ by solving

$$\Phi_{s,0} \mathbf{U}_{s,0}^{(k)} + \Phi_{s,b} \mathbf{Z}_s^{(k)} \mathbf{O}'_{y_s} = \mathbf{F}_s \mathbf{A}_s^{-1} - \Phi_{s,b} \mathbf{B}_s^{-1} \mathbf{b}_s^{(k-1)} \mathbf{y}'_s / (\mathbf{y}'_s \mathbf{y}_s), \quad s = 1, \dots, S. \quad (24)$$

Step 2. Obtain $\mathbf{b}_s^{(k)}$ by solving

$$\Phi_{s,b} \mathbf{B}_s^{-1} \mathbf{b}_s^{(k)} \mathbf{y}'_s / (\mathbf{y}'_s \mathbf{y}_s) = \mathbf{F}_s \mathbf{A}_s^{-1} - \Phi_{s,0} \mathbf{U}_{s,0}^{(k)} - \Phi_{s,b} \mathbf{Z}_s^{(k)} \mathbf{O}'_{y_s}, \quad s = 1, \dots, S. \quad (25)$$

Note that the equations in the system (24) appeared in Step 1 can be solved in parallel for $s = 1, \dots, S$. But the equations in the system (25) appeared in Step 2 should be solved collectively for all $s = 1, \dots, S$, since \mathbf{b}_s is shared by multiple local regions and thus appears in multiple equations. The block Gauss-Seidel method converges very fast. When the dimension of the domain Ω is d and the domain is decomposed into S local regions of d -simplices, the total number of boundaries in between the local regions is proportional to dS . On the other hand, the size of the linear system to be solved in Step 2 is proportional to the number of boundaries. Since the coefficient matrix of the linear system is a banded matrix, the complexity of solving such a linear system is proportional to the square of the size of the linear system (Mahmood et al., 1991), that is, $O(d^2 S^2)$.

4.2.3 NUMERICAL COMPARISON

This section numerically compares the two aforementioned solutions for boundary value estimation. We used the same data set used in Section 4.1.1 and applied the same partitioning scheme for splitting the entire domain into 10 local regions, in between which there are nine boundary locations. We applied the localized estimation method and the iterative block Gauss-Seidel approach for estimating $f(x)$ at the nine locations, and compared them with the estimated values from a full GP regression. Figure 2-(a) shows the comparison results. The root mean squared difference of the localized estimation to a full GP regression was 0.0775, while that of the iterative approach was 0.1369. Both of the errors are far below the noise parameter $\sigma = 1$. The computation time for the estimation was comparable, 0.041328 seconds for the localized method and 0.038071 seconds for the iterative approach.

For another comparison, we generated a synthetic data set in 2-d of 8,000 noisy observations from a zero-mean Gaussian process with an exponential covariance function of scale one and variance 10,

$$y_i = f(x_i) + \epsilon_i \quad \text{for } i = 1, \dots, 8000,$$

where $x_i \sim \text{Uniform}([0, 6] \times [0, 6])$ and $\epsilon_i \sim \mathcal{N}(0, 1)$ were independently sampled, and the Gaussian process realization $f(x_i)$ was simulated by the R package `RandomField`. We split the input domain $[0, 6] \times [0, 6]$ into sixteen local regions of equal size, and 1,881 test locations were chosen uniformly surrounding the local region boundaries. For each test location, we obtained the prediction based on a full GP regression and computed the differences of the

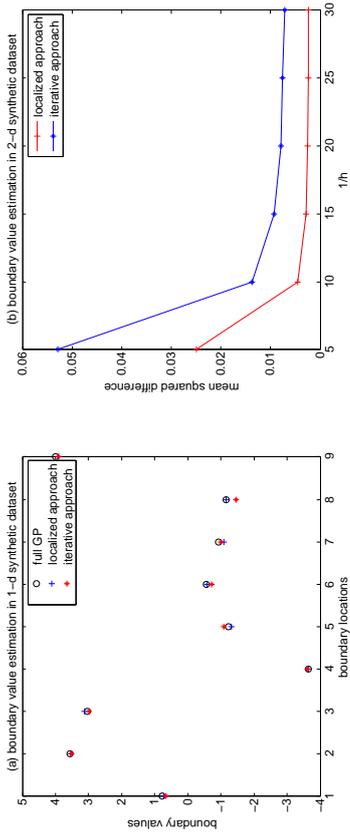


Figure 2: Comparison of the two proposed methods of boundary value estimation with a full GP regression.

boundary estimation by the localized approach or the iterative approach and the full GP regression prediction. The mean squared differences versus mesh size h are plotted in Figure 2-(b). Again, the localized approach works better.

4.3 Computation Complexity and Implementation Details

The total computation complexity of our proposed approach is the summation of two complexities, one for the constrained GP regressions and the other for the boundary value estimation. It is $O(SN_S^3 + dSN_B^2)$ when the localized estimation approach is used for boundary value estimation, and it is $O(SN_S^3 + d^2 S^2)$ when the block Gauss-Seidel iteration is used. Since $SN_S = N$ and N_B is a constant, the complexity is $O(NN_S^2 + dS)$ or $O(NN_S^2 + d^2 S^2)$ respectively.

4.3.1 EVALUATION OF INTEGRALS FOR QUANTITIES IN EQUATION (20)

The integrals for defining several quantities in equation (20) can be computed effectively using well-established finite element computations. Suppose that $\Omega \subset \mathbb{R}^d$ and we use the Lagrange finite elements of polynomial degree k , where the s th local region Ω_s is partitioned into M mesh cells $\{\mathcal{K}_m; m = 1, \dots, M\}$ for the finite element approximation. The ϕ_s is a column vector of the Lagrange basis functions for the mesh cells,

$$\{\phi_{m,j}; m = 1, \dots, M, j = 0, 1, \dots, J\}, \quad \text{and } J = \binom{d+k}{k}.$$

It is well known that each $\phi_{m,j}$ is a polynomial function of barycentric coordinates λ_j 's with respect to the d -simplex \mathcal{K}_m (Ern and Guermond, 2004, pages 22-23). One can use the integral formula for barycentric coordinates (Voitovich and Vandewalle, 2008) to compute

$\Phi_{s,0}$ and $\Phi_{s,b}$. For example, when $d = 2$,

$$\int_{\mathcal{K}_m} \lambda_{j_1}^a \lambda_{j_2}^b \lambda_{j_3}^c = 2! |\mathcal{K}_m| \frac{a!b!c!}{(2+a+b+c)!},$$

and when $d = 3$,

$$\int_{\mathcal{K}_m} \lambda_{j_1}^a \lambda_{j_2}^b \lambda_{j_3}^c \lambda_{j_4}^d = 6! |\mathcal{K}_m| \frac{a!b!c!d!}{(3+a+b+c+d)!},$$

where $|\mathcal{K}_m|$ is the volume of \mathcal{K}_m . Since $\phi_{m,j_1} \phi_{m,j_2}$ is also a polynomial functions of λ_j 's, one can use the previous integration formulas to evaluate $\int_{\mathcal{K}_m} \phi_{m,j_1} \phi_{m,j_2}$ and

$$\int_{\Omega_s} \phi_{m,j_1} \phi_{m,j_2} = \sum_m \int_{\mathcal{K}_m} \phi_{m,j_1} \phi_{m,j_2}. \quad (26)$$

For the values of $\mathbf{F}_{s,i}$, one can take the finite element approximation of $\mathbf{f}_{s,i}$ where each function f_i in \mathbf{f}_s is approximated by

$$\sum_j \alpha_{m,j}^{(i)} \phi_{m,j} \text{ on } \mathcal{K}_m.$$

With this approximation, \mathbf{F}_s becomes

$$\begin{aligned} \int_{\Omega_s} \phi_s f_i &= \sum_m \int_{\mathcal{K}_m} \phi_s \sum_j \alpha_{m,j}^{(i)} \phi_{m,j} \\ &= \sum_m \sum_j \alpha_{m,j}^{(i)} \int_{\mathcal{K}_m} \phi_s \phi_{m,j}. \end{aligned}$$

The last integral can be computed using (26).

Since $\phi_{m,j} \Gamma_{st}$ is a polynomial function of barycentric coordinates with respect to Γ_{st} ,

$$\int_{\Gamma_{st}} \phi_{m,j_1} \Gamma_{st} \phi_{m,j_2} \Gamma_{st}$$

can be computed using the integral formulas in barycentric coordinates, facilitating the evaluation of \mathbf{B}_s and \mathbf{b}_s .

4.3.2 LEARNING COVARIANCE PARAMETERS

By far, our discussions have been made when using fixed parameters (often referred to as hyperparameters in the literature) for the covariance function $k(\cdot, \cdot)$. In this subsection, we discuss how to choose the hyperparameters. Basically, we follow the approach in Park et al. (2011), which has two options, namely choosing different hyperparameters for each local region or choosing the same hyperparameters for all local regions. When different hyperparameters are chosen for each local region, the hyperparameters are estimated by maximizing the local marginal likelihood functions. Specifically, the local hyperparameters, denoted by θ_s associated with each Ω_s , are selected such that they minimize the negative log marginal likelihood:

$$ML_s(\theta_s) := -\log p(\mathbf{y}_{s,i}; \theta_s) = \frac{n_s}{2} \log(2\pi) + \frac{1}{2} \log |\mathbf{A}_s| + \frac{1}{2} \mathbf{y}_{s,i}^T \mathbf{A}_s^{-1} \mathbf{y}_{s,i}. \quad (27)$$

where \mathbf{A}_s depends on θ_s . Note that (27) is the marginal likelihood of the standard local kriging model typically seen in geostatistics.

When we want to choose the same hyperparameters applied for all local regions, we choose the hyperparameter θ such that it minimizes

$$ML(\theta) = \sum_{s=1}^S ML_s(\theta), \quad (28)$$

where the summation of the negative log local marginal likelihoods is over all local regions. The above treatment implicitly assumes that the data from each local region are mutually independent. We used the criterion (28) for all numerical comparisons presented below.

5. Numerical Study of Patched GP and Comparison with DDM

In this section, we present the numerical performance of our patched GP method for different tuning parameters, compared to the full GP regression. We also compare our patched GP method with its precursor, the DDM (Park et al., 2011).

5.1 Data Sets and Evaluation Criteria

We considered four data sets: one synthetic data set in 1-d, one synthetic data set in 2-d, and three real spatial data sets both in 2-d. The two synthetic data sets were generated by the R package `RandomField`. The first data set in 1-d (hereafter denoted by `synthetic-1d`) consists of 6,000 noisy observations from a zero-mean Gaussian process with an exponential covariance function of scale one and variance 10,

$$y_i = f(x_i) + \epsilon_i \quad \text{for } i = 1, \dots, 6000,$$

where $x_i \sim \text{Uniform}(0, 10)$ and $\epsilon_i \sim \mathcal{N}(0, 1)$ were independently sampled, and the Gaussian process realization $f(x_i)$ was simulated by the R package `RandomField`. The two (hereafter denoted by `synthetic-2d`) consists of 8,000 noisy observations from a zero-mean Gaussian process with an exponential covariance function of scale one and variance 10,

$$y_i = f(x_i) + \epsilon_i \quad \text{for } i = 1, \dots, 8000,$$

where $x_i \sim \text{Uniform}([0, 6] \times [0, 6])$ and $\epsilon_i \sim \mathcal{N}(0, 1)$ were independently sampled, and the Gaussian process realization $f(x_i)$ was simulated by the R package `RandomField`. The two synthetic data sets were used to show how our proposed method performs, compared to the full GP regression.

The first real data set, TCO, contains data collected by NIMBUS-7/TOMS satellite to measure the total column of ozone over the globe on Oct 1 1988. This set consists of 48,331 measurements. The second real data set, TCO-L2, also contains the total column of ozone measured by the same satellite on the same date at much more locations (182,591 locations). The third real data set, ICETHICK, is the ice thickness profile for a portion of the western Antarctic ice sheet, which is available at <http://nsidc.org/>. The data set has 32,481 measurements. As shown in Figure 3, the ICETHICK data set has some sparse regions with very few training points, while the TCO data set has a very dense distribution of the training points; TCO-L2 data set has even denser distribution.

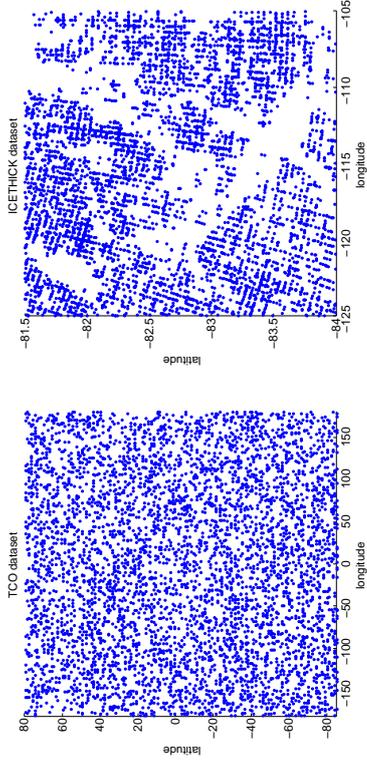


Figure 3: Spatial distribution of the measurements for two real data sets. A dot represent one measurement.

Using the three real spatial data sets, we can compare the computation time and prediction accuracy of the patched GP with other methods. We randomly split each data set into a training set containing 90% of the total observations and a test set containing the remaining 10% of the observations. To compare the computational efficiency of methods, we measure two computation times, the training time (including the time for hyperparameter learning) and the prediction (or test) time. For comparison of accuracy, we use two measures on the set of the test data, denoted as $\{(x_t, y_t); t = 1, \dots, T\}$, where T is the size of the test set. The first measure is the mean squared error (MSE)

$$\text{MSE} = \frac{1}{T} \sum_{t=1}^T (y_t - \mu_t)^2, \quad (29)$$

which measures the accuracy of the mean prediction μ_t at location x_t . The second one is the negative log predictive density (NLPD)

$$\text{NLPD} = \frac{1}{T} \sum_{t=1}^T \left[\frac{(y_t - \mu_t)^2}{2\sigma_t^2} + \frac{1}{2} \log(2\pi\sigma_t^2) \right], \quad (30)$$

which considers the accuracy of the predictive variance σ_t as well as the mean prediction μ_t . These two criteria were used broadly in the GP regression literature. A smaller value of MSE or NLPD indicates a better performance.

When applying the patched GP, one issue is how to partition the whole domain into local regions, also known as *meshing* in the finite element analysis literature (Ern and Guermond, 2004). The patched GP works with any shapes of meshing. For this paper, we used a uniform triangular mesh, where each local region is a triangular shaped region of the same size. The implementation of the meshing was performed using the `DistMesh` MATLAB

software (Persson and Strang, 2004). We used the localized estimation presented in Section 4.2.1 for boundary value estimation, and the hyperparameters of a covariance function was obtained by minimizing (28) and was applied for all local regions. All numerical studies were performed on a computer with Intel Xeon Processor W3520 and 6GB memory.

5.2 Performance of Patched GP with Different Tuning Parameters

The patched GP has two tuning parameters, number of local regions S and mesh size h in finite element approximation. If the number of local regions (S) is one (i.e. there is no split to local regions), the patched GP should converge to a full GP as the mesh size of the patched GP's finite element approximation goes to zero. In this section we illustrate how the patched GP works for $S > 1$ and different mesh sizes using the data sets described in the previous section.

For `synthetic-1d`, we uniformly partitioned the domain $[0, 10]$ into S local regions of equal size where S varies over $\{2, 4, 6, 8, 10\}$. The number of meshes per local region is denoted by M , and it is related to mesh size h , which is the length of an interval mesh. We randomly split 6,000 observations in `synthetic-1d` into a training data set of 4,500 observations and a test data set of 1,500 observations. For each S and h , we used the training data set to learn the patched GP and a full GP, and compared the mean squared difference of the patched GP and a full GP over the test data set. Figure 4-(a) shows the mean squared difference versus S and h . Regardless of S , the difference converges to almost zero (about e^{-6}) as h goes to zero, which implies that the mean prediction of the patched GP becomes very close to that of a full GP even with a large S ; this can be qualitatively seen in Figure 4-(b), -(c) and -(d). In other words, the performance of the patched GP does not vary much with the choice of S although the method with a larger S typically converges faster.

For `synthetic-2d`, we uniformly partitioned the domain into S local regions of equal size where S varies over $\{8, 47, 32, 17, 10, 4\}$. The number of meshes per local region is denoted by M , and it is related to mesh size h , which is the side length of a triangular mesh for `synthetic-2d`. We randomly split 8,000 observations in `synthetic-2d` into a training data set of size 6,500 and a test data set of size 1,500. For each S and h , we used the training data set to learn the patched GP and a full GP, and compared the mean squared difference between the patched GP and the full GP over the test data set. Figure 5 shows the mean squared difference versus S and h . Similar to the 1-d case, the performance of patched GP does not vary much with different S values for the two synthetic data sets.

We also evaluated the performance of the patched GP on the real data sets `TCO` and `ICETHICK` for different values of S and M . As described in Section 5.1, 90% of each data set was randomly chosen and used as a training data set, and the remaining 10% was used for computing the MSE. Figure 6 summarizes the results. The performance of the patched GP did not vary much for different S . If S is large, the overall computation complexity decreases significantly as M decreases. Therefore, in general, a larger S is preferred.

5.3 Comparison with DDM

Our proposed patched GP method is the direct enhancement of DDM. In this section, we compare the numerical performance of DDM and patched GP.

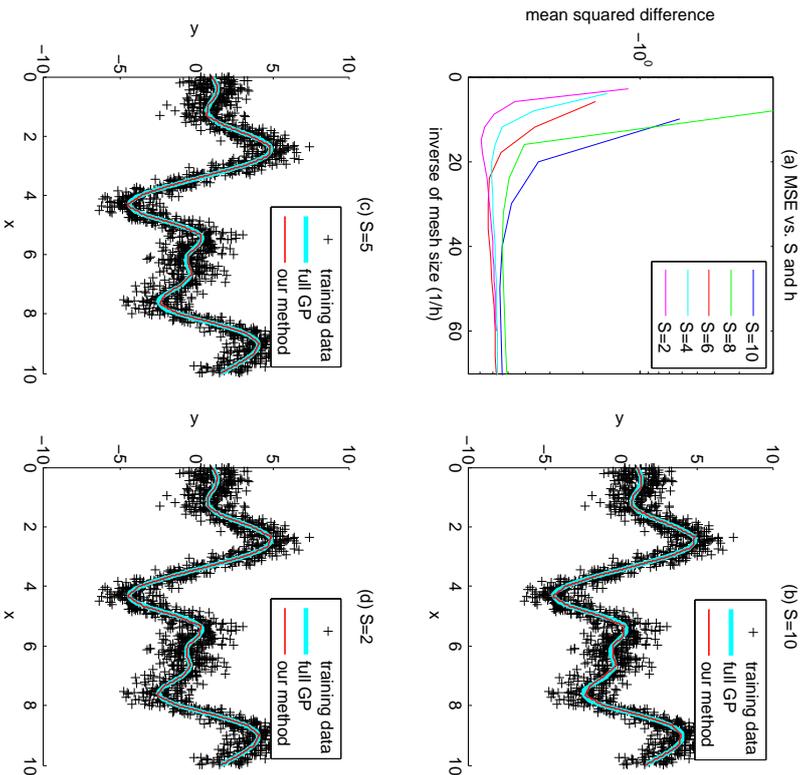


Figure 4: Mean squared difference of the patched GP and a full GP over the test data set for `synthetic-1d`: (a) shows the mean squared differences for different S and h parameter values, and (b)-(d) illustrate the mean predictions of the patched GP and a full GP for different S 's with fixed $1/h = 3$.

5.3.1 OVERALL PERFORMANCE

We used three real data sets in 2-d to compare the MSEs and NLPDs of patched GP and DDM versus the total computation time (training and test time), which includes the time for hyperparameter learning, model learning, and prediction.

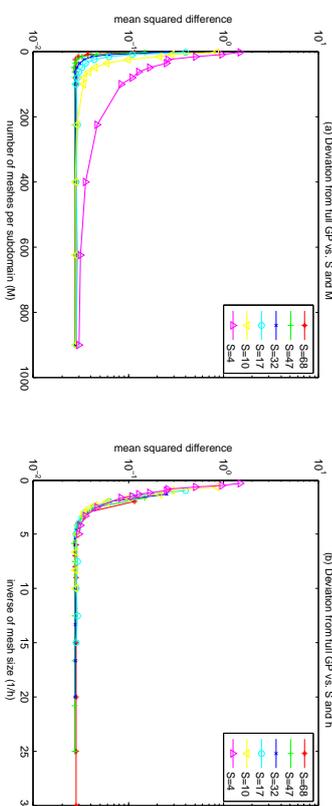


Figure 5: Mean squared difference of the patched GP and a full GP versus S and h for `synthetic-2d`

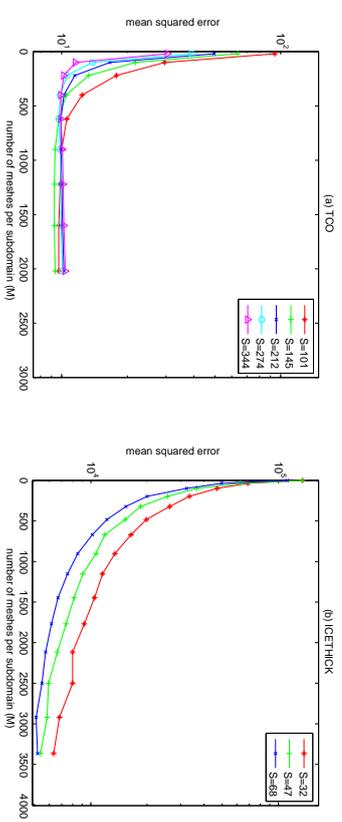


Figure 6: MSE of the patched GP versus S and h for TCO and ICETHICK.

For both methods, we fixed S , $S = 145$ for TCO, $S = 623$ for TCO.L2, and $S = 47$ for ICETHICK, because the performance did not vary much with the choice of S . We applied the squared exponential covariance function and the same covariance hyperparameter learning method for both DDM and patched GP, which is described in Park et al. (2011, Section 5). In the hyperparameter learning, we used a subset of the training data. The fractions of the training data set used for the hyperparameter learning varied over $\{0.3, 0.5, 0.8, 1.0\}$. As the fraction increases, we expect the training time increases, and the accuracy of the hyperparameter estimation and the final prediction improves. For patched GP, the number of meshes per local region for the finite element approximation (h) varied from 5 to 30 with step size 5. As seen in Section 5.2, the increase of h implies the increase of the

overall prediction accuracy and the total computation time. For DDM, we varied the number of degree of freedoms on a local region boundary from 5 to 30 with step size 5. For each experimental setting, we performed 20 replicated experiments with new random splits of training and test data sets. We randomly split each data set into a training set containing 90% of the total observations and a test set containing the remaining 10% of the observations. The MSEs, NLPDs and the total computation times were averaged to reduce the variation caused by random splits.

Figure 7 shows the MSE and NLPD versus the total computation time for both DDM and patched GP. For shorter computation time, DDM performed better in terms of MSE but patched GP obtained lower MSEs with longer computation time. In terms of NLPD, the DDM was better for TCO and TCO.L2. The NLPD is roughly the squared bias of the predictive mean divided by the predictive variance. Since the patched GP is better than the DDM in MSE, the better NLPD performance of the DDM can be attributed to difference in variance estimation. For TCO and TCO.L2, the DDM's variance estimation is sufficiently large to cover most observations. However, the DDM's variance estimation is sometimes too small, being negative. For example, the DDM produced the negative predictive variances for ICETHICK, so the resulting NLPDs are imaginary numbers. The issue with DDM regarding possible negative predictive variances has been reported in Pourhabib et al. (2014). The same problem occurred in this numerical example.

5.3.2 COMPARISON IN BOUNDARY VALUE ESTIMATION

We compared DDM and patched GP for boundary value estimation. For this comparison, we used the localized estimation approach described in Section 4.2.1 with $N_B = 50$. The comparison was primarily focused on (1) how the boundary estimation of each method on a boundary location is close to the mean prediction of a full GP regression on the same location, and (2) how the boundary estimations of two neighboring local regions on a boundary point are close to each other. We fixed $S = 16$ and tried different mesh sizes h for patched GP and different numbers of the control points placed on each boundary (p) for DDM, which are directly relevant to the performance of boundary estimation. The h varied over one fifth of a local region size through one thirtieth of a local region size, while p comparably varied over five through thirty.

We used the whole `synthetic-2d` data set as a training data set to train both of the methods, and 1,881 test locations were chosen uniformly from local region boundaries. For each test location, we obtained the mean prediction of a full GP regression. The squared differences of the mean prediction of DDM or patched GP to that of a full GP at the test locations are averaged to obtain the mean squared difference. This difference versus h or p is plotted in Figure 8-(a). As h decreases, the boundary estimation of patched GP at the test locations converges to the mean prediction of a full GP regression at the same locations, while the boundary estimation of DDM keeps deviating from a full GP result.

We also compared how consistent the mean predictions from two neighboring local regions at their shared boundary are. We simply took the two mean predictions from two neighboring local regions at some of the 1,881 test locations on their shared boundary. The squared differences were taken and averaged over all shared boundaries. The mean squared differences versus h or p are plotted in Figure 8-(b). The mean squared differences for

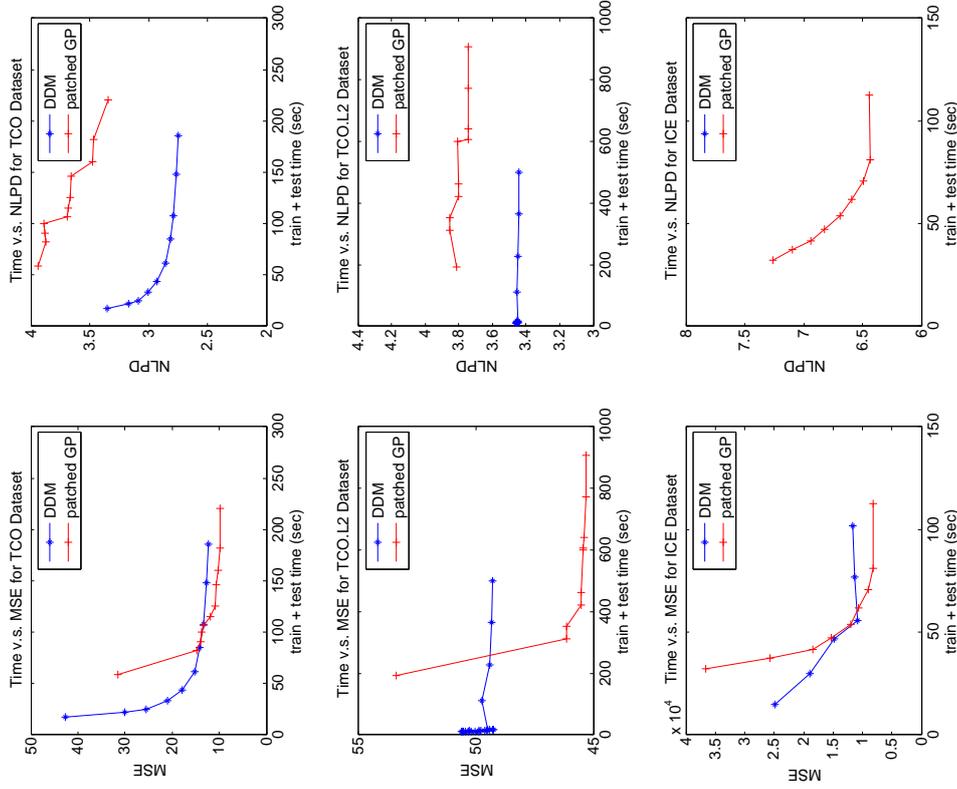


Figure 7: Prediction accuracy versus total computation time. For ICETHICK (ICE) data set, the NLPDs of DDM are imaginary numbers since the predictive variance estimates were all negative.

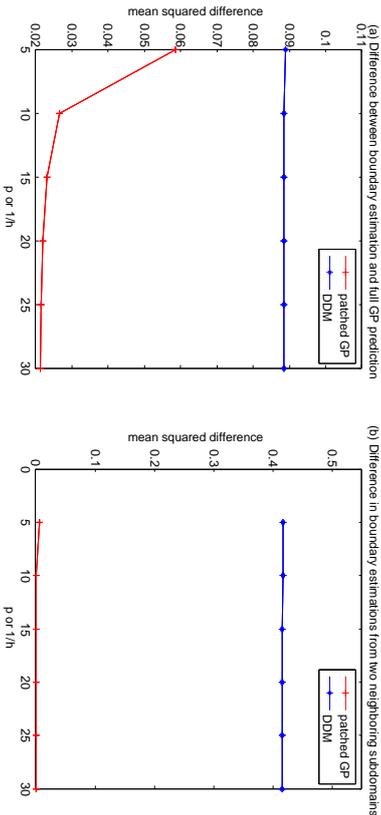


Figure 8: Accuracy of Boundary Estimation.

patched GP are almost zero, while those for DDM are significantly non-zero relative to those of patched GP.

6. Numerical Comparison with Other State-of-The-Art methods

This section compares patched GP with other state-of-the-art methods. Section 6.1 contains the comparison to several localized approaches for GP regression, while Section 6.2 contains the comparison to the Gaussian Markov random field approach to the GP regression (Lindgren et al., 2011).

6.1 Comparison with Other Local GP Methods

In this section, we compare patched GP with other localized approaches for GP regression, including BCM (Tresp, 2000), PIC (Snelson and Ghahramani, 2007), and RBCM (Deisenroth and Ng, 2015); we used the author’s implementation of BCM and implemented RBCM and PIC with matlab by ourselves. We used three real data sets TCO, TCO_L2 and ICETHICK to compare the MSEs and NLPDs of the approaches versus the total computation time, which includes the time for hyperparameter learning, model learning and prediction. We used the squared exponential covariance function and used the whole training data set to choose hyperparameters for all of the compared methods. For patched GP, we fixed $S = 145$ for TCO, $S = 623$ for TCO_L2, and $S = 47$ for ICETHICK, and the number of meshes per local region was varied from 5 to 40 with step size 5. For BCM and RBCM, we varied the number of local experts $M \in \{100, 150, 200, 250, 300, 600\}$ for TCO, $M \in \{50, 100, 150, 200, 250, 300\}$ for TCO_L2, and $M \in \{50, 100, 150, 200, 250, 300\}$ for ICETHICK. For PIC, we varied the total number of local regions $m \in \{100, 150, 200, 250, 350\}$ for TCO, $m \in \{100, 200, 300, 400, 600\}$ for TCO_L2, and $m \in \{50, 100, 150, 200\}$ for ICETHICK, and also varied the number of in-

cluding inputs ($M \in \{100, 150, 200, 250, 300, 400, 500, 600, 700\}$) for all of the data sets. We used the k-means clustering for splitting training data for both of BCM, RBCM, and PIC.

Figure 9 shows the logarithms of MSEs and NLPDs versus total computation times for the three data sets. For TCO data set, the BCM, RBCM, and patched GP obtained more accurate prediction than the PIC, and the patched GP was computationally more efficient than the BCM and RBCM. For TCO_L2 data set, the patched GP uniformly outperformed other competing methods; scaling better than BCM and achieving better MSE than the PIC. It is interesting to see that BCM is almost identically performing as the RBCM when N is so large like in TCO_L2 data set. For ICETHICK data set, the PIC and patched GP obtained more accurate prediction than the BCM and the RBCM. Please note that the training data are quite densely spread over the whole domain for TCO while the training data are sparse for some local regions in ICETHICK. The patched GP worked well for both of the cases, while the PIC worked better for the sparse case and the BCM worked better for the dense case. The RBCM has shown much better results than the BCM for the sparse case but it is not better than the patched GP and PIC. The PIC combines a global model with local models, which may help to improve the performance for the sparse case.

6.2 Comparison with GMRF

In this section, we compare patched GP with the Gaussian Markov random field approach to the GP regression (Lindgren et al., 2011, GMRF), which was reported to scale great with massive data set; we implemented the GMRF with matlab. The major checkpoints of this comparison are the scalability and prediction accuracy. We used three real data sets of different sizes, ICETHICK ($N = 32, 813$), TCO ($N = 48, 311$) and TCO_L2 ($N = 182, 591$) to compare the MSEs, NLPDs, and computation time of the approaches. In this comparison, we used the exponential covariance function, since the GMRF does not work with the squared exponential covariance function used for the other comparissons; at least, the construction of the precision matrix for Gaussian Markov random field is not straightforward. The GMRF does not have any tuning parameters, and the hyperparameter learning of the GMRF was performed using 5% of the training data; the MSE performance did not change much as the percentage increases, so we chose the smallest percentage to obtain the smallest computation time. For patched GP, we presented the results with the combinations of tuning parameters that obtain the best RMSE and the worst RMSE. To be specific, we fixed $S = 145$ for TCO, $S = 47$ for ICETHICK, and $S = 623$ for TCO_L2, while the number of meshes per local region was varied from 5 to 40 with step size 5.

Table 1 summarizes the comparison results. The computation time of patched GP increases linearly in data size N , while the GMRF’s computation time increases in $O(N^2)$. This is not surprising because the GMRF’s computation depends on n_z , the number of nonzero elements in the precision matrix, proportionally in n_z^2 or $n_z^{3/2}$ and the n_z increases at least linearly in N . For prediction performance, the best RMSE of the patched GP was at least comparable or better than that of GMRF. The patched GP uniformly outperformed the GMRF in terms of NLPD, which means that the posterior distribution of the patched GP was better fitted to test data sets.

Datasets	patched GP (best, worst)			GMRF		
	Time	RMSE	NLPD	Time	RMSE	NLPD
TCO	(250.4, 48.4)	(2.85, 5.75)	(3.20, 3.91)	651.4	2.78	5.01
TCO.L2	(807.4, 192.2)	(6.74, 7.31)	(3.70, 3.82)	5702.7	9.24	5.26
ICETHICK	(212.1, 31.2)	(89.30, 199.33)	(6.12, 6.80)	385.0	89.80	7.08

Table 1: Performance comparison of the patched GP with GMRF: the time unit used is second.

7. Conclusion

We developed a method for solving a Gaussian process regression with constraints on a domain boundary and also developed a solution approach based on a finite element method. The method is then applied to local GP regressions as a building block to develop the patched GP method as a computationally efficient solver of a large-scale Gaussian process regression or spatial kriging problem. The patched GP solves two issues of the simple local GP approaches, namely the inaccuracy and inconsistency of prediction on the boundaries of neighboring local regions. Comparing with its precursor DDM, the patched GP has an improved way of considering the constraints related to the boundary regions. Both methods reformulate the GP regression as an optimization problem, the patched GP method improves DDM by rewriting the optimization problem in a function space and using the finite element methods to solve the required integrals arising from the solution of the minimization problem. The patched GP method is mathematically more elegant and its competitiveness to existing methods is demonstrated through numerical studies.

Acknowledgments

The authors thank the reviewers for constructive comments. The authors also thank Anton Schwaighofer at Microsoft for sharing the BCM implementation. Chiwoo Park was supported by grants from the National Science Foundation (CMMI-1334012) and the Air Force Office of Scientific Research (FA9550-13-1-0075). Jianhua Z. Huang was supported by grants from the National Science Foundation (DMS-1208952) and the Air Force Office of Scientific Research (FA9550-13-1-0075).

Appendix A. Proof of Proposition 1

Note that $[L^2(\Omega)]^N$ is a Hilbert space with the following inner product

$$(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \mathbf{u}' \mathbf{v}.$$

We define a bi-linear form on $[L^2(\Omega)]^N$ by $a : [L^2(\Omega)]^N \times [L^2(\Omega)]^N \rightarrow \mathbb{R}$,

$$a(\mathbf{u}, \mathbf{v}) = (\mathbf{A}\mathbf{u}, \mathbf{v}) = \int_{\Omega} \mathbf{u}' \mathbf{A} \mathbf{v} \quad \text{for } \mathbf{u}, \mathbf{v} \in [L^2(\Omega)]^N,$$

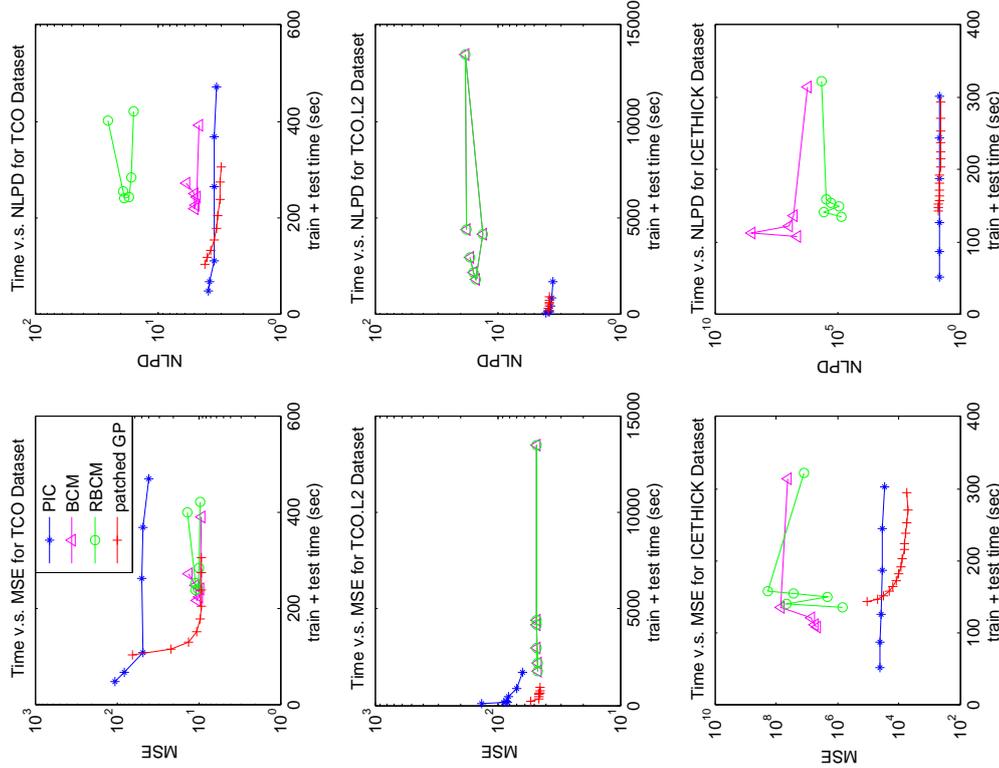


Figure 9: Prediction accuracy versus total computation time. The legend in the upper left panel applies to all other panels.

a linear functional $c : [L^2(\Omega)]^N \rightarrow \mathbb{R}$ as

$$c(\mathbf{u}) = \int_{\Omega} \mathbf{f}^T \mathbf{u},$$

and define $J(\mathbf{u}) = \frac{1}{2}a(\mathbf{u}, \mathbf{u}) - c(\mathbf{u})$. Since \mathbf{A} is a $N \times N$ (real) positive definite matrix, the bilinear form $a(\mathbf{u}, \mathbf{v})$ is symmetric and positive. Let α be the smallest eigenvalue of \mathbf{A} . We have that

$$a(\mathbf{u}, \mathbf{u}) \geq \alpha \|\mathbf{u}\|^2, \quad \forall \mathbf{u} \in [L^2(\Omega)]^N.$$

Therefore, the bilinear form a is coercive. It follows from Ern and Guermond (2004, Proposition 2.4) that, \mathbf{u} satisfies $a(\mathbf{u}, \mathbf{v}) - c(\mathbf{v}) = 0$ for every $\mathbf{v} \in [L^2(\Omega)]^N$ if and only if it minimizes $J(\mathbf{u})$ over $\mathbf{u} \in [L^2(\Omega)]^N$. Note that the coercivity of the bilinear form a can be interpreted as a strong convexity property of the functional $J(\mathbf{u})$, which makes the problem have a unique optimal solution (Ern and Guermond, 2004, Lemma 2.2). ■

Appendix B. Proof of Proposition 2

We have already proven that $a(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \mathbf{u}^T \mathbf{A} \mathbf{v}$ is coercive, symmetric and positive for $\mathbf{u}, \mathbf{v} \in [L^2(\Omega)]^N$ in the proof of Proposition 1. The same result holds for $\mathbf{u}, \mathbf{v} \in H_0$ because $H_0 \subset [L^2(\Omega)]^N$. Since H_0 is a Hilbert space, solving the minimization problem is equivalent to solving the integral equation (8) by Ern and Guermond (2004, Proposition 2.4). ■

References

- Sudipto Banerjee, Alan E Gelfand, Andrew O Finley, and Huiyan Sang. Gaussian predictive process models for large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(4):825–848, 2008.
- Tao Chen and Jiangehong Ren. Bagging for Gaussian process regression. *Neurocomputing*, 72(7):1605–1610, 2009.
- Noel Cressie and Gardar Johannesson. Fixed rank kriging for very large spatial data sets. *Journal of Royal Statistical Society, Series B*, 70:209–226, 2008.
- Paul J Curran and Peter M Atkinson. Geostatistics and remote sensing. *Progress in Physical Geography*, 22(1):61–78, 1998.
- Marc Peter Deisenroth and Jun Wei Ng. Distributed Gaussian processes. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1–10, 2015.
- A. Ern and J.L. Guermond. *Theory and practice of finite elements*. Springer Verlag, 2004. ISBN 0387205748.
- Reinhard Furrer, Marc G Genton, and Douglas Nychka. Covariance tapering for interpolation of large spatial datasets. *Journal of Computational and Graphical Statistics*, 15(3):2006.
- Thore Graepel. Solving noisy linear operator equations by Gaussian processes: Application to ordinary and partial differential equations. In *International Workshop on Machine Learning*, volume 20, page 234, 2003.
- Robert B Gramacy and Daniel W Apley. Local Gaussian process approximation for large computer experiments. *Journal of Computational and Graphical Statistics*, pages 561–578, 2015.
- Robert B Gramacy and Herbert KH Lee. Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103(483), 2008.
- Cari G Kaufman, Mark J Schervish, and Douglas W Nychka. Covariance tapering for likelihood-based estimation in large spatial data sets. *Journal of the American Statistical Association*, 103(484):1545–1555, 2008.
- Finn Lindgren, Håvard Rue, and Johan Lindström. An explicit link between Gaussian fields and Gaussian markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498, 2011.
- David JC MacKay. Introduction to Gaussian processes. In C. M. Bishop, editor, *Neural networks and machine learning*, volume 168 of *NATO ASI Series F Computer and Systems Sciences*, pages 133–166. Springer Verlag, 1998.
- A Mahmood, DJ Lynch, and LID Philipp. A fast banded matrix inversion using connectivity of schur’s complements. In *IEEE International Conference on Systems Engineering*, pages 303–306. IEEE, 1991.
- Duy Nguyen-Thong, Jan R Peters, and Matthias Seeger. Local Gaussian process regression for real time online model learning. In *Advances in Neural Information Processing Systems*, pages 1193–1200, 2009.
- Chiwoo Park, Jianhua Z. Huang, and Yu Ding. Domain decomposition approach for fast Gaussian process regression of large spatial datasets. *Journal of Machine Learning Research*, 12:1697–1728, May 2011.
- Per-Olof Persson and Gilbert Strang. A simple mesh generator in matlab. *SIAM review*, 46(2):329–345, 2004.
- Arash Pourhabib, Fanning Liang, and Yu Ding. Bayesian site selection for fast Gaussian process regression. *IIE Transactions*, 46(5):543–555, 2014.
- C. E. Rasmussen and Z. Ghahramani. Infinite mixtures of Gaussian process experts. In *Advances in Neural Information Processing Systems 14*, pages 881–888. MIT Press, 2002.
- C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005.
- Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

- Huiyan Sang and Jianhua Z. Huang. A full-scale approximation of covariance functions for large spatial data sets. *Journal of Royal Statistical Society, Series B*, 74:111–132, 2012.
- Matthias Seeger, Christopher K. I. Williams, and Neil D. Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In *International Workshop on Artificial Intelligence and Statistics 9*. Society for Artificial Intelligence and Statistics, 2003.
- Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems 18*, pages 1257–1264. MIT Press, 2006.
- Edward Snelson and Zoubin Ghahramani. Local and global sparse Gaussian process approximations. In *International Conference on Artificial Intelligence and Statistics 11*, pages 524–531. Society for Artificial Intelligence and Statistics, 2007.
- Michael L Stein. *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media, 2012.
- Florian Steinke and Bernhard Schölkopf. Kernels, regularization and differential equations. *Pattern Recognition*, 41(11):3271–3286, 2008.
- Volker Tresp. A bayesian committee machine. *Neural Computation*, 12(11):2719–2741, 2000.
- Jarno Vanhatalo and Aki Vehtari. Modelling local and global phenomena with sparse Gaussian processes. In *the 24th Conference on Uncertainty in Artificial Intelligence*, page 571578. Association for Uncertainty in Artificial Intelligence, 2008.
- Tatiana V Voitovich and Stefan Vandewalle. Barycentric interpolation and exact integration formulas for the finite volume element method. In *International Conference on Numerical Analysis and Applied Mathematics*, volume 1048, pages 575–579. AIP Publishing, 2008.

bandicoot: a Python Toolbox for Mobile Phone Metadata

Yves-Alexandre de Montjoye

MIT Media Lab, 02139 Cambridge MA, USA

Imperial College London, Dept. of Computing and Data Science Institute, London SW7 2AZ, UK

DEMONTJOYE@IMPERIAL.AC.UK

Luc Rocher

Université catholique de Louvain, ICTEAM, 1348 Louvain-la-Neuve, Belgium

LUC.ROCHER@UCLOUVAIN.BE

Alex ‘Sandy’ Pentland

MIT Media Lab, Cambridge, 02139 Cambridge MA, USA

PENTLAND@MIT.EDU

Editor: Alexandre Gramfort

Abstract

bandicoot is an open-source Python toolbox to extract more than 1442 features from standard mobile phone metadata. bandicoot makes it easy for machine learning researchers and practitioners to load mobile phone data, to analyze and visualize them, and to extract robust features which can be used for various classification and clustering tasks. Emphasis is put on ease of use, consistency, and documentation. bandicoot has no dependencies and is distributed under MIT license.

Keywords: Python, feature engineering, mobile phone metadata, CDR, visualization

1. Introduction

The metadata generated at large-scale by mobile phones and collected by every carrier around the world have the potential to fundamentally transform the way we fight diseases and design transportation systems. Scientists have compared the recent availability of these large-scale behavioral data sets to the invention of the microscope (Giles, 2012) and their business value is considerable (Kaye, 2015). In machine learning, mobile phone metadata have been used to predict people’s gender (Sarraute et al., 2014), age (Sarraute et al., 2014), personality (de Montjoye et al., 2013), literacy rates (Sundsoy, 2016); as well as their likelihood to repay loans (Bjorktegren and Grissen, 2015), subscribe to services (Sundsoy et al., 2014), and commit crimes (Bogomolov et al., 2014). In disaster relief operations for instance, knowing the demographics of a person along with his or her mobility data is extremely valuable (Wilson et al., 2016). As most phones in low and middle income countries are prepaid (ITU, 2013), we often lack these information about the users. Being able to predict million of people’s demographic information with a small training set is therefore tremendously valuable and cost-efficient.

Despite a great potential for impact and close to 10 years of academic and industry research in using mobile phone metadata (Blondel et al., 2015), there were so far no open-source software to process and extract robust features from them. All prediction works were consequently based on a limited number of custom indicators. This has prevented research from progressing: features had to be redeveloped every time and numerous implementation

choices (e.g. reconciling data, choices of thresholds and time periods, edge cases) were lost from one paper to another. This made it hard to replicate results, quantify the impact of new methods, and transfer learnings.

bandicoot solves this problem by providing researchers and practitioners with an efficient open-source Python feature extractor for mobile phone metadata. bandicoot is a complete, easy-to-use and extensible environment: with a few lines of code, a user can load mobile phone data, extract features, and export them. bandicoot’s modular structure makes it easy for users to add new features and leverage existing pre- and post-processing functions.

2. Usage and features

bandicoot provides users with more than 1442 individual, spatial, and social network features:

Individual features (e.g. percent of nocturnal interactions, time it takes someone to answer text message, inter-event time between two phones calls) describe an individual’s phone usage and interactions with his or her contacts.

Spatial features (e.g. entropy of visited antennas, radius of gyration) describe an individual’s mobility patterns. Note that to avoid sampling biases, bandicoot groups location data per 30 min slots.

Social network (e.g. clustering coefficient, assortativity) describe an individual’s social network and compare his or her behavior with the one of their contacts.

bandicoot computes these indicators or, when they are a distribution their mean and standard deviation, on a weekly basis. It then returns the weekly mean and standard deviation to the user (see Fig. 1 (a)). The user can also compute indicators only on call/texts, weekdays/weekends, or days/nights and an extended set of weekly summary statistics (median, min, max, kurtosis, and skewness).

bandicoot also standardizes from the mobile phone research literature (Blondel et al., 2015) the definition of conversations between individuals, the conversion of directed to undirected matrices, the assortativity of attributes within ego-networks, as well as a the binning scheme to avoid sampling biases in location data.

3. Project focus

Code quality Correctness and consistency is absolutely crucial for us. The metrics implemented need to be correct and the values returned stable over time. To ensure this, we implemented functions to generate synthetic mobile phone data, regression tests, and more than 50 function-level unit tests covering 88% of the source code.

Community-driven development The choice of the Python language, as well as a specific focus on readability, helps contributors understand and modify bandicoot. We are actively building a community of users around bandicoot to foster changes and improvements in the source code, develop new features, and to report and help correct faulty behaviors. bandicoot is hosted on GitHub and has already been developed by 9 contributors over

the last three years with many others helping with bug reports, features requests, and discussions.

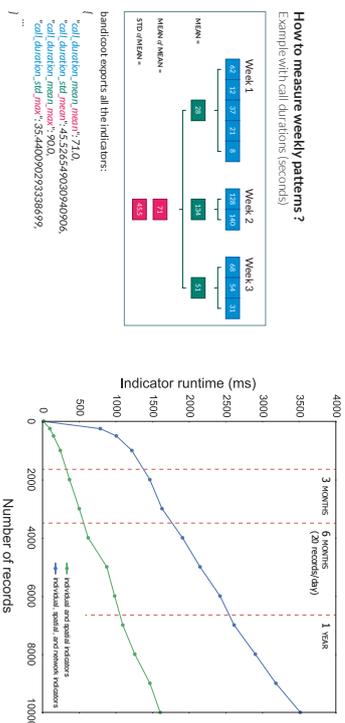


Figure 1: (a) Weekly aggregation of bandicoot's behavioral indicators. (b) bandicoot's run-time as a function of the number of records (single core) with and without network indicators

Documentation An extensive documentation has been written and is maintained. This includes an exhaustive description of the input and output of functions, a quick start guide, and a demonstration notebook. A specific part of the documentation is written to help users test and contribute new features.

Easy-to-install and without dependencies bandicoot runs on Python 2 and 3; we support (and test bandicoot with) GNU/Linux, Mac OS, and Windows. bandicoot is meant to (and already is) used in highly secured and heterogeneous telco environments where packages have to be approved and verified. To make its use (incl. in Hadoop environments) and installation easier, we developed bandicoot to be free of any dependencies such as pandas or compilers.

Detecting data issues Numerous data issues can happen in mobile phone data sets: incorrect locations, missing incoming records, duplicates, etc. We have built 38 reporting variables that are automatically added when exporting features. These include details about the underlying data (start and end date, number of records, percentage of antenna missing location information, etc.) and about the data processing (bandicoot version number, type of aggregation used, records that have been removed and why, whether a home location has been detected, etc.).

4. Visualization

bandicoot includes an interactive tool to visualize the data of a specific user. The visualization allows researchers to inspect the data, detect issues, and spot potentially important patterns. Fig. 2 shows the visualization tool, with options to display only specific weeks

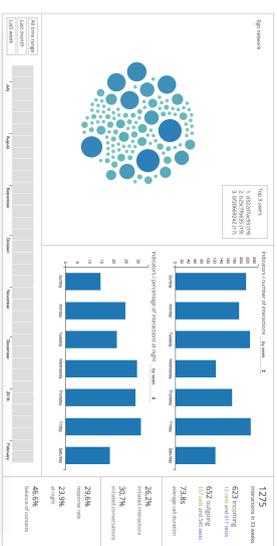


Figure 2: bandicoot visualization tool for an individual's data

and to plot indicators (or their weekly mean/sum) over the time period considered. It can be generated and served on-the-fly over HTTP, or exported to disk and shared.

5. Performances

While our priority developing bandicoot is to produce correct and verifiable code that can be easily extended, a certain number of implementation choices and optimizations were made to ensure that a standard large-scale dataset can be processed overnight. For instance, bandicoot caches groups of records by week, weekend, call or text to limit both the computing time and the memory footprint. All together these optimizations have sped up computation by a factor 3 compared to a naive implementation.

We tested bandicoot on a computer with an Intel i7 CPU (2.6GHz) and 8GB of memory for users with an average of 20 records per days over 3 months. It takes on average 250ms to compute all 1442 behavioral and mobility indicators using the standard Python Interpreter, CPython, and 160ms using pypy, a fast just-in-time compiler. For all indicators, including network features, the total time is 1.11s using CPython (740ms with pypy). Network indicators take significantly more time as data for all the neighbors of one node have to be loaded and their behavioral indicators computed. All indicators run in linear time with the number of records (Fig. 1(b)) and a standard large-scale data set of one million mobile phone users is processed in less than 9 hours (resp. 39h for the network indicators) using the multiprocessing code we provide.

6. Conclusion

The application of machine learning algorithms to mobile phone metadata has a great potential for good and businesses. Until now, there were no toolbox to efficiently process and extract robust features from them. This was a major issue for both researchers and practitioners. bandicoot implements a full data pipeline for mobile phone data including more than 1442 features. It has been used in research papers as well as by carriers (e.g. Orange, Telenor) and international organizations (e.g. WFP).

References

- Daniel Bjorkgren and Darrell Grissen. Behavior revealed in mobile phone usage predicts loan repayment. *Available at SSRN 2611775*, 2015.
- Vincent D Blondel, Adeline Decuyper, and Gautier Krings. A survey of results on mobile phone datasets analysis. *EPI Data Science*, 4(1):1, 2015.
- Andrey Bogomolov, Bruno Lepri, Jacopo Staiano, Nuria Oliver, Fabio Pianesi, and Alex Pentland. Once upon a crime: Towards crime prediction from demographics and mobile data. In *Proceedings of the 16th International Conference on Multimodal Interaction*, pages 427–434. ACM, 2014.
- Yves-Alexandre de Montjoye, Jordi Quidbach, Florent Robic, and Alex Sandy Pentland. Predicting personality using novel mobile phone-based metrics. In *Social Computing, Behavioral-Cultural Modeling and Prediction*, pages 48–55. Springer, 2013.
- Jim Giles. Making the links. *Nature*, 488(7412):448–450, 2012.
- International Telecommunication Union. Itu releases latest global technology development figures. http://www.itu.int/net/pressoffice/press_releases/2013/05.aspx, 2013. Accessed: 2015-10-17.
- Kate Kaye. The \$24 Billion Data Business That Telcos Don't Want to Talk About, 2015. URL <http://adage.com/article/datadriven-marketing/24-billion-data-business-telcos-discuss/301058>.
- Carlos Sarraute, Pablo Blanc, and Javier Burroni. A study of age and gender seen through mobile phone usage patterns in mexico. In *Advances in Social Networks Analysis and Mining, 2014 IEEE/ACM International Conference on*, pages 836–843. IEEE, 2014.
- Pål Sundsoy. Can mobile usage predict illiteracy in a developing country? *arXiv preprint arXiv:1607.01337*, 2016.
- Pål Sundsoy, Johannes Bjelland, Asif M Iqbal, Yves-Alexandre de Montjoye, et al. Big data-driven marketing: How machine learning outperforms marketers' gut-feeling. In *Social Computing, Behavioral-Cultural Modeling and Prediction*, pages 367–374. Springer, 2014.
- Robin Wilson, Elisabeth zu Erbach-Schoenberg, Maximilian Albert, Daniel Power, Simon Tudge, Miguel Gonzalez, Sam Guthrie, Heather Chamberlain, Christopher Brooks, Christopher Hughes, et al. Rapid and near real-time assessments of population displacement using mobile phone data following disasters: the 2015 nepal earthquake. *PLoS currents*, 8, 2016.

Input Output Kernel Regression: Supervised and Semi-Supervised Structured Output Prediction with Operator-Valued Kernels

Céline Brouard

*Helsinki Institute for Information Technology HIIT
Department of Computer Science, Aalto University
02150 Espoo, Finland
IBISC, Université d'Évry Val d'Essonne
91037 Évry cedex, France*

CELINE.BROUARD@AALTO.FI

Marie Szafranski

*ENSILE & LAMME, Université d'Évry Val d'Essonne, CNRS, INRA
91037 Évry cedex, France
IBISC, Université d'Évry Val d'Essonne
91037 Évry cedex, France*

MARIE.SZAFRAŃSKI@MATH.CNRS.FR

Florence d'Alché-Buc

*LTCI, CNRS, Télécom ParisTech
Université Paris-Saclay
46, rue Barrault 75013 Paris, France
IBISC, Université d'Évry Val d'Essonne
91037 Évry cedex, France*

FLORENCE.DALCHE@TELECOM-PARISTECH.FR

Editor: Koji Tsuda

Abstract

In this paper, we introduce a novel approach, called Input Output Kernel Regression (IOKR), for learning mappings between structured inputs and structured outputs. The approach belongs to the family of Output Kernel Regression methods devoted to regression in feature space endowed with some output kernel. In order to take into account structure in input data and benefit from kernels in the input space as well, we use the Reproducing Kernel Hilbert Space theory for vector-valued functions. We first recall the ridge solution for supervised learning and then study the regularized hinge loss-based solution used in Maximum Margin Regression. Both models are also developed in the context of semi-supervised setting. In addition we derive an extension of Generalized Cross Validation for model selection in the case of the least-square model. Finally we show the versatility of the IOKR framework on two different problems: link prediction seen as a structured output problem and multi-task regression seen as a multiple and interdependent output problem. Eventually, we present a set of detailed numerical results that shows the relevance of the method on these two tasks.

Keywords: structured output prediction, output kernel regression, vector-valued RKHS, operator-valued kernel, semi-supervised learning

1. Introduction

Many real world applications involve objects with an explicit or implicit discrete structure. Texts, images and videos in document processing and retrieval as well as genes and proteins in computational biology are all examples of implicit structured data that we may want to use as inputs or outputs in a prediction system. Besides these structured objects, structured output prediction can also concern multiple outputs linked by some relationship that is relevant to take into account. Surprisingly, although a lot of attention has been paid to learning from structured inputs for now two decades, this problem, often referred as *structured output learning*, has emerged relatively recently as a field of interest in statistical learning. In the literature, structured output prediction has been addressed from two main angles. A first angle consists in *discriminative learning algorithms* that provide predictions by maximizing a scoring function over the output space. Conditional Random Fields (Lafferty et al., 2001) and their extension to kernels (Lafferty et al., 2004) were first proposed for discriminative modeling of graph-structured data and sequence labeling. Other discriminative learning algorithms based on maximum margin such as structured SVM (Tsochantaridis et al., 2004, 2005), Maximum Margin Markov Networks (M^3N) (Taskar et al., 2004) or Maximum Margin Regression (Szedmak et al., 2005) have then been developed and thoroughly studied. A common approach to those methods consists in defining a linear scoring function based on the image of an input-output pair by a joint feature map. Both methods, either based on Conditional Random Fields or maximum-margin techniques, are costly to train and generally assume that the output set \mathcal{Y} is discrete. Keeping the idea of a joint feature map over inputs and outputs, a generative method called Joint Kernel Support Estimation has been recently proposed (Lampert and Blaschko, 2009). In this approach, a one-class SVM is used to learn the support of the joint-probability density $p(x, y)$. More recently, another angle to structured output prediction, that we called *Output Kernel Regression (OKR)*, has emerged around the idea of using the kernel trick in the output space and making predictions in a feature space associated to the output kernel. As a first example, the seminal work of Kernel Dependency Estimation (KDE) was based on the definition of an input kernel as well as an output kernel. After a first version using kernel PCA to define a finite-dimensional output feature space (Weston et al., 2003), a more general KDE framework consisting in learning a linear function from the input feature space to the output feature space was proposed by Cortes et al. (2005). In this setting, predictions in the original output space are retrieved by solving a pre-image problem. Interestingly, the idea of Output Kernel Regression can be implemented without defining an input kernel as it is shown with Output Kernel Tree-based methods (Geurts et al., 2006, 2007a,b). In these approaches, a regression tree whose outputs are linear combinations of the training outputs in the output feature space is built using the kernel trick in the output space: the loss function which is locally minimized during the construction only involves inner products between training outputs. These methods are not limited to discrete output sets and they do not require expensive computations to make a prediction nor to train the model. Combined in ensembles such as random forests and boosting, they exhibit excellent performances. However these tree-based approaches suffer from two drawbacks: trees do not take into account structured input data except by using a flat description of them and the associated (greedy) building algorithm cannot be easily extended to semi-supervised learning.

In this work, we therefore propose to extend the methodology of Output Kernel Regression to another large family of nonparametric regression tools that allows to tackle structured data in the input space as well as in the output space. Moreover we will show that this new family of tools is useful in a semi-supervised context. Called Input Output Kernel Regression, this novel family for structured output prediction from structured inputs relies on Reproducing Kernel Hilbert Spaces (RKHS) for vector-valued functions with the following specification: the output vector belongs to some output feature space associated to a chosen output kernel, as introduced in the works of Brouard et al. (2011) and Brouard (2013). Let us recall that in the case of scalar-valued functions, the RKHS theory offers a flexible framework for penalized regression as witnessed by the abundant literature on the subject (Wahba, 1990; Pearce and Wand, 2006). A penalized regression problem is seen as a minimization problem in a functional space built on an input scalar-valued kernel. Depending the nature of the prediction problem, appropriate penalties can be defined and representer theorem can be proven, facilitating the minimization problem to be further solved. In the RKHS theory, regularization constraint on the geometry of the probability distribution of labeled and unlabeled data can also be added to perform semi-supervised regression (Belkin et al., 2006). When functions are vector-valued, the adequate RKHS theory makes use of operator-valued kernels (Pedrick, 1957; Senkane and Tempelman, 1973; Mitchell and Pontil, 2005). Operator-valued kernels have already been proposed to solve problems of multi-task regression (Evgeniou et al., 2005; Baldassarre et al., 2012), structured classification (Dinuzzo et al., 2011), vector autoregression (Lim et al., 2013) as well as functional regression (Kadri et al., 2010). The originality of this work is to consider that the output space is a feature space associated to a chosen output kernel. This new approach not only enhances setting of pattern recognition tasks by requiring to pay attention on both input and output sets but also opens new perspectives in machine learning. It encompasses in a unique framework kernel-based regression tools devoted to structured inputs as well as structured outputs.

1.1 Related Works

In Brouard et al. (2011), the vector-valued RKHS theory was used to address the output kernel regression problem in the semi-supervised setting. This approach was applied to the link prediction problem. By working in the framework of RKHS theory for vector-valued functions, we extended the manifold regularization framework introduced by Belkin et al. (2006) to functions with values in a Hilbert space. We have also shown that the first step of KDE (Cortes et al., 2005) is a special case of IOKR using a particular operator-valued kernel.

Kadri et al. (2013) studied a formulation of KDE using operator-valued kernels. The first step of this approach is identical to the IOKR framework developed in Brouard et al. (2011) and Brouard (2013). The second step consists in extending the pre-image step of KDE using the vector-valued RKHS theory. They also proposed two covariance-based operator-valued kernels and showed that using these operator-valued kernels allow to express the pre-image problem using only input and output Gram matrices.

In parallel of Brouard et al. (2011), Minh and Shindhwani (2011) generalized the manifold regularization framework proposed by Belkin et al. (2006) for semi-supervised learning to vector-valued functions.

1.2 Contributions

We introduce Input Output Kernel Regression (IOKR), a novel class of penalized regression problems based on the definition of an output scalar-valued kernel and an input operator-valued kernel. This article is an extended version of Brouard et al. (2011), that addresses more generally the problem of structured output prediction. In this work, we present several novel contributions regarding the RKHS theory for functions with values in a Hilbert space. We present the representer theorem for vector-valued functions in the semi-supervised setting. Based on this representer theorem, we study two particular models obtained using two different loss functions: the *IOKR-ridge* model introduced in Brouard et al. (2011) and a new model called *IOKR-margin*. This model extends the Maximum Margin Regression (MMR) framework introduced by Seidman et al. (2005) to operator-valued kernels and to the semi-supervised setting. In this paper, we also put the reformulation of Kernel Dependency Estimation proposed by Cortes et al. (2005) into perspective in the Output Kernel Regression framework. We present the solutions corresponding to decomposable kernels. In the case of the least-squared loss function, we describe a new tool for model selection, which was first introduced in Brouard (2013). The selection of the hyperparameters is done by estimating the averaged error obtained with leave-one-out cross-validation as a closed-form solution. We show the versatility of the IOKR framework on two different problems: link prediction and multi-task regression. Finally, we present numerical results obtained with IOKR on these two tasks.

1.3 Organization of the Paper

This paper is organized as follows. In Section 2, we introduce the Input Output Kernel Regression approach and show how it can be used to solve structured output prediction problems. In Section 3 we describe the RKHS theory devoted to vector-valued function and present our contributions to this theory in the supervised and semi-supervised settings. We also present in this section models based on decomposable operator-valued kernels. We then show in Section 4 that, in the case of the least-squares loss function, the leave-one-out criterion can be estimated by a closed-form solution. In Section 5, we describe how Input Output Kernel Regression (IOKR) can be used to solve two structured prediction problems, which are link prediction and multi-task learning. In Section 6, we present the results obtained with IOKR on these two problems.

The notations used in this paper are summarized in Table 1.

2. From Output Kernel Regression to Input Output Kernel Regression

We consider the general regression task consisting in learning a mapping between an input set \mathcal{X} and an output set \mathcal{Y} . We assume that both \mathcal{X} and \mathcal{Y} are sample spaces and that $S_n = \{(x_i, y_i), i = 1 \dots n\}$ is an i.i.d. sample drawn from the joint probability law \mathcal{P} defined on $\mathcal{X} \times \mathcal{Y}$. Outputs are supposed to be structured, for example objects such as sequences,

Meaning	Symbol
number of labeled examples	ℓ
number of unlabeled examples	n
input set	\mathcal{X}
set of labeled examples	\mathcal{X}_ℓ
union of the labeled and unlabeled sets	$\mathcal{X}_{\ell+n}$
output set	\mathcal{Y}
input scalar kernel	$\kappa_x : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$
output scalar kernel	$\kappa_y : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$
input feature space	\mathcal{F}_x
output feature space	\mathcal{F}_y
input feature map	$\varphi_x : \mathcal{X} \rightarrow \mathcal{F}_x$
output feature map	$\varphi_y : \mathcal{Y} \rightarrow \mathcal{F}_y$
set of bounded operators from an Hilbert space \mathcal{F} to itself	$\mathcal{B}(\mathcal{F})$
set of bounded operators from \mathcal{F} to an Hilbert space \mathcal{G}	$\mathcal{B}(\mathcal{F}, \mathcal{G})$
operator-valued kernel	$\mathcal{K}_x : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{B}(\mathcal{F}_y)$
reproducing kernel Hilbert space of \mathcal{K}_x	$\mathcal{H}, \mathcal{HK}_x$
canonical feature map of \mathcal{K}_x	$\phi_x : \mathcal{X} \rightarrow \mathcal{B}(\mathcal{F}_y, \mathcal{H})$
gram matrix of \mathcal{K}_x on \mathcal{X}_ℓ and $\mathcal{X}_{\ell+n}$	$\mathbf{K}_{\mathcal{X}_\ell}, \mathbf{K}_{\mathcal{X}_{\ell+n}}$
gram matrix of κ_x on \mathcal{X}_ℓ and $\mathcal{X}_{\ell+n}$	$K_{\mathcal{X}_\ell}, K_{\mathcal{X}_{\ell+n}}$
gram matrix of κ_y on \mathcal{Y}_ℓ	$K_{\mathcal{Y}_\ell}$
graph Laplacian	L
matrix vectorization	vec
Kronecker product	\otimes
Hadamard product (element-wise product)	\circ

Table 1: Notations used in this paper

graphs, nodes in a graph, or simply vectors of interdependent variables. It is realistic to assume that one can build a similarity $\kappa_y : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ between the elements of the output set \mathcal{Y} , such that κ_y takes into account the inherent structure of the elements of \mathcal{Y} and has the properties of a positive definite kernel. Then, due to the Moore-Aronszajn theorem (Aronszajn, 1950), there exists a Hilbert space \mathcal{F}_y , called a *feature space*, and a corresponding function $\varphi_y : \mathcal{Y} \rightarrow \mathcal{F}_y$, called a *feature map* such that:

$$\forall (y, y') \in \mathcal{Y} \times \mathcal{Y}, \kappa_y(y, y') = \langle \varphi_y(y), \varphi_y(y') \rangle_{\mathcal{F}_y}.$$

The regression problem between \mathcal{X} and \mathcal{Y} can be decomposed into two tasks (see Figure 1):

- the first task is to learn a function h from the set \mathcal{X} to the Hilbert space \mathcal{F}_y
- the second one is to define or learn a function f from \mathcal{F}_y to \mathcal{Y} to provide an output in the set \mathcal{Y} .

We call the first task, *Output Kernel Regression* (OKR), referring to previous works based on Output Kernel Trees (OK3) (Geurts et al., 2006, 2007a) and the second task, a

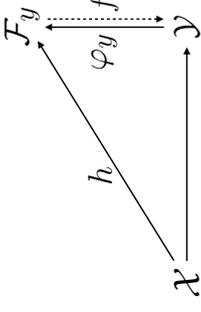


Figure 1: Schema of the Output Kernel Regression approach.

pre-image problem. In this paper, we develop a general theoretical and practical framework for the OKR task, allowing to deal with structured inputs as well as structured outputs. To illustrate our approach, we have chosen two structured output learning tasks which do not require to solve a pre-image problem. One is *multi-task regression* for which the dimension of the output feature space is finite, and the other one is *link prediction* for which prediction in the original set \mathcal{Y} is not required. However, the approach we propose can be combined with pre-image solvers now available on the shelves. The interested reader may want to refer to Honeine and Richard (2011) or Kadri et al. (2013) to benefit from existing pre-image algorithms to solve structured output learning tasks.

In this work, we propose to build a family of models and learning algorithms devoted to *Output Kernel Regression* that present two additional properties compared to OK3-based methods: namely, models are able to take into account structure in input data and can be learned within the framework of penalized regression, enjoying various penalties including smoothness penalties for semi-supervised learning. To achieve this goal, we choose to use kernels both in the input and output spaces. As the models have values in a feature space and not in \mathbb{R} , we turn to the vector-valued reproducing kernel Hilbert spaces theory (Pedrick, 1957; Senkane and Tempelman, 1973; Burbea and Masani, 1984) to provide a general framework for penalized regression of nonparametric vector-valued functions. In that theory, the values of kernels are operators on the output vectors which belong to some Hilbert space. Introduced in machine learning by the seminal work of Micchelli and Pontil (2005) to solve multi-task regression problems, operator-valued kernels (OVK) have then been studied under the angle of their universality (Caponnetto et al. (2008); Carmeli et al. (2010)) and developed in different contexts such as structured classification (Dinuazzo et al., 2011), functional regression (Kadri et al., 2010), link prediction (Brouard et al., 2011) or semi-supervised learning (Minh and Sindhvani, 2011; Brouard et al., 2011). With operator-valued kernels, models of the following form can be constructed:

$$\forall x \in \mathcal{X}, h(x) = \sum_{i=1}^n \mathcal{K}_x(x, x_i) \mathbf{c}_i, \mathbf{c}_i \in \mathcal{F}_y, x_i \in \mathcal{X}, \quad (1)$$

extending nicely the usual kernel-based models devoted to real-valued functions.

In the case of IOKR, the output Hilbert space \mathcal{F}_y is defined as a feature space related to a given output kernel. Note that there exists different pairs (feature space, feature map) associated with a given kernel κ_y . Let us take for instance the polynomial kernel

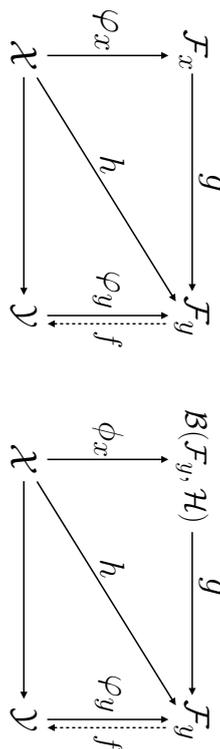


Figure 2: Diagrams describing Kernel Dependency Estimation (KDE) on the left and Input Output Kernel Regression (IOKR) on the right.

$\kappa_{y_g}(y, y') = (y^T y' + c)^p$: we can choose the finite feature space defined by the different monomies of the coordinates of a vector \mathbf{y} or we can choose the RKHS associated with the polynomial kernel. This choice will open doors to different output feature spaces \mathcal{F}_{y_g} , leading to different definitions of the input operator-valued kernel \mathcal{K}_x and thus to different learning problems. Omitting the choice of the feature map associated to \mathcal{F}_{y_g} , we therefore need to define a triplet $(\kappa_{y_g}, \mathcal{F}_{y_g}, \mathcal{K}_x)$ as a pre-requisite to solve the structured output learning task. By explicitly requiring to define an output kernel we emphasize the fact that an input operator-valued kernel cannot be defined without calling into question the output space, \mathcal{F}_{y_g} , and therefore, the output kernel κ_{y_g} . We will show in Section 6 that the same structured output prediction problem can be solved in different ways using different values for the triplet $(\kappa_{y_g}, \mathcal{F}_{y_g}, \mathcal{K}_x)$.

Interestingly, IOKR generalizes Kernel Dependency Estimation (KDE), a problem that was introduced in Weston et al. (2003) and was reformulated in a more general way by Cortes et al. (2005). If we call \mathcal{F}_x a feature space associated to a scalar input kernel $\kappa_x : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ and $\varphi_x : \mathcal{X} \rightarrow \mathcal{F}_x$ a corresponding feature map, KDE uses Kernel Ridge regression to learn a function h from \mathcal{X} to \mathcal{F}_{y_g} by building a function g from \mathcal{F}_x to \mathcal{F}_{y_g} and composing it with the feature map φ_x (see Figure 2). The function h is modeled as a linear function: $h(x) = W\varphi_x(x)$, where W is a linear operator from \mathcal{F}_x to \mathcal{F}_{y_g} . The second phase consists in computing the pre-image of the obtained prediction.

In the case of IOKR, we build models of the general form introduced in Equation (1). Denoting ϕ_x the canonical feature map associated to the OVK \mathcal{K}_x , which is defined as: $\phi_x(x) = \mathcal{K}_x(\cdot, x)$, we can draw the chart depicted in Figure 2 on the right. The function ϕ_x maps inputs from \mathcal{X} to $\mathcal{B}(\mathcal{F}_{y_g}, \mathcal{H})$. Indeed the value $\phi_x(x)y = \mathcal{K}_x(\cdot, x)y$ is a function of the RKHS \mathcal{H} for all y in \mathcal{F}_{y_g} . The model h is seen as the composition of a function g from $\mathcal{B}(\mathcal{F}_{y_g}, \mathcal{H})$ to the output feature space \mathcal{F}_{y_g} with the input feature map ϕ_x .

We can therefore see on Figure 2 how IOKR extends KDE. In Brouard et al. (2011), we have shown that we retrieve the model used in KDE when considering the following operator-valued kernel:

$$\mathcal{K}_x(x, x') = \kappa_x(x, x')I, \quad (2)$$

where I is the identity operator from \mathcal{F}_{y_g} to \mathcal{F}_{y_g} . Unlike KDE, that learns independently each component of the vectors $\varphi_y(y)$, IOKR takes into account the structure existing between these components.

The next section is devoted to the RKHS theory for vector-valued functions and to our contributions to this theory in the supervised and semi-supervised settings.

3. Operator-Valued Kernel Regression

In the following, we briefly recall the main elements of the RKHS theory devoted to vector-valued functions (Senkane and Tempelman, 1973; Mitchell and Pontil, 2005) and then present our contributions to this theory.

Let \mathcal{X} be a set and \mathcal{F}_{y_g} a Hilbert space. In this section, no assumption is needed about the existence of an output kernel κ_{y_g} . We note $\tilde{\mathbf{y}}$ the vectors in \mathcal{F}_{y_g} . Given two Hilbert spaces \mathcal{F} and \mathcal{G} , we note $\mathcal{B}(\mathcal{F}, \mathcal{G})$ the set of bounded operators from \mathcal{F} to \mathcal{G} and $\mathcal{B}(\mathcal{F})$ the set of bounded operators from \mathcal{F} to itself. Given an operator A , A^* denotes the adjoint of A .

Definition 1 *An operator-valued kernel on $\mathcal{X} \times \mathcal{X}$ is a function $\mathcal{K}_x : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{B}(\mathcal{F}_{y_g})$ that verifies the two following conditions:*

- $\forall (x, x') \in \mathcal{X} \times \mathcal{X}, \mathcal{K}_x(x, x') = \mathcal{K}_x(x', x)^*$,
- $\forall m \in \mathbb{N}, \forall \mathcal{S}_m = \{(x_i, \tilde{\mathbf{y}}_i)\}_{i=1}^m \subseteq \mathcal{X} \times \mathcal{F}_{y_g}, \sum_{i,j=1}^m \langle \tilde{\mathbf{y}}_i, \mathcal{K}_x(x_i, x_j)\tilde{\mathbf{y}}_j \rangle_{\mathcal{F}_{y_g}} \geq 0$.

The following theorem shows that given any operator-valued kernel, it is possible to build a reproducing kernel Hilbert space associated to this kernel.

Theorem 2 (Senkane and Tempelman (1973); Micchelli and Pontil (2005))

Given an operator-valued kernel $\mathcal{K}_x : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{B}(\mathcal{F}_{y_g})$, there is a unique Hilbert space $\mathcal{H}_{\mathcal{K}_x}$ of functions $h : \mathcal{X} \rightarrow \mathcal{F}_{y_g}$ which satisfies the following reproducing property:

$$\forall h \in \mathcal{H}_{\mathcal{K}_x}, \forall x \in \mathcal{X}, h(x) = \mathcal{K}_x(x, \cdot)h,$$

where $\mathcal{K}_x(x, \cdot)$ is an operator in $\mathcal{B}(\mathcal{H}_{\mathcal{K}_x}, \mathcal{F}_{y_g})$.

As a consequence, $\forall x \in \mathcal{X}, \forall \tilde{\mathbf{y}} \in \mathcal{F}_{y_g}, \forall h \in \mathcal{H}_{\mathcal{K}_x}, \langle \mathcal{K}_x(\cdot, x)\tilde{\mathbf{y}}, h \rangle_{\mathcal{H}_{\mathcal{K}_x}} = \langle \tilde{\mathbf{y}}, h(x) \rangle_{\mathcal{F}_{y_g}}$.

The Hilbert space $\mathcal{H}_{\mathcal{K}_x}$ is called the reproducing kernel Hilbert space associated to the kernel \mathcal{K}_x . This RKHS can be built by taking the closure of $\text{span}\{\mathcal{K}_x(\cdot, x)\alpha \mid x \in \mathcal{X}, \alpha \in \mathcal{F}_{y_g}\}$. The scalar product on $\mathcal{H}_{\mathcal{K}_x}$ between two functions $f = \sum_{i=1}^n \mathcal{K}_x(\cdot, x_i)\alpha_i$ and $g = \sum_{j=1}^m \mathcal{K}_x(\cdot, t_j)\beta_j, x_i, t_j \in \mathcal{X}, \alpha_i, \beta_j \in \mathcal{F}_{y_g}$, is defined as:

$$\langle f, g \rangle_{\mathcal{H}_{\mathcal{K}_x}} = \sum_{i=1}^n \sum_{j=1}^m \langle \alpha_i, \mathcal{K}_x(x_i, t_j)\beta_j \rangle_{\mathcal{F}_{y_g}}.$$

The corresponding norm $\|\cdot\|_{\mathcal{H}_{\mathcal{K}_x}}$ is defined by $\|f\|_{\mathcal{H}_{\mathcal{K}_x}}^2 = \langle f, f \rangle_{\mathcal{H}_{\mathcal{K}_x}}$. For sake of simplicity we replace the notation $\mathcal{H}_{\mathcal{K}_x}$ by \mathcal{H} in the rest of the paper.

As for scalar-valued functions, one of the most appealing feature of RKHS is to provide a theoretical framework for regularization with the representer theorems.

3.1 Regularization in Vector-Valued RKHS

Based on the RKHS theory for vector-valued functions, Micchelli and Pontil (2005) have proved a representer theorem for convex loss functions in the supervised case.

We note $S_\ell = \{(x_i, \tilde{\mathbf{y}}_i)\}_{i=1}^\ell \subseteq \mathcal{X} \times \mathcal{F}_y$ the set of labeled examples and \mathcal{H} the RKHS with reproducing kernel $\mathcal{K}_x : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{B}(\mathcal{F}_y)$.

Theorem 3 (Micchelli and Pontil (2005)) *Let \mathcal{L} be a convex loss function, and $\lambda_1 > 0$ a regularization parameter. The minimizer of the following optimization problem:*

$$\operatorname{argmin}_{h \in \mathcal{H}} \mathcal{J}(h) = \sum_{i=1}^{\ell} \mathcal{L}(h(x_i), \tilde{\mathbf{y}}_i) + \lambda_1 \|h\|_{\mathcal{H}}^2,$$

admits an expansion:

$$\hat{h}(\cdot) = \sum_{j=1}^{\ell} \mathcal{K}_x(\cdot, x_j) \mathbf{c}_j,$$

where the coefficients $\mathbf{c}_j, j = 1, \dots, \ell$ are vectors in the Hilbert space \mathcal{F}_y .

In the following, we plug the expansion form of the minimizer into the optimization problem and consider the problem of finding the coefficients \mathbf{c}_j for two different loss functions: the least-squares loss and the hinge loss.

3.1.1 PENALIZED LEAST SQUARES

Considering the least-squares loss function for regularization of vector-valued functions, the minimization problem becomes:

$$\operatorname{argmin}_{h \in \mathcal{H}} \mathcal{J}(h) = \sum_{i=1}^{\ell} \|h(x_i) - \tilde{\mathbf{y}}_i\|_{\mathcal{F}_y}^2 + \lambda_1 \|h\|_{\mathcal{H}}^2. \quad (3)$$

Theorem 4 (Micchelli and Pontil (2005)) *Let $\mathbf{c}_j \in \mathcal{F}_y, j = 1, \dots, \ell$, be the coefficients of the expansion admitted by the minimizer \hat{h} of the optimization problem in Equation (3). The vectors $\mathbf{c}_j \in \mathcal{F}_y$ satisfy the equations:*

$$\sum_{i=1}^{\ell} (\mathcal{K}_x(x_j, x_i) + \lambda_1 \delta_{ij}) \mathbf{c}_i = \tilde{\mathbf{y}}_j,$$

where δ is the Kronecker symbol: $\delta_{ii} = 1$ and $\forall j \neq i, \delta_{ij} = 0$.

Let $\mathbf{c} = (\mathbf{c}_j)_{j=1}^{\ell} \in \mathcal{F}_y^{\ell}$ and $\tilde{\mathbf{y}} = (\tilde{\mathbf{y}}_j)_{j=1}^{\ell} \in \mathcal{F}_y^{\ell}$. This system of equations can be equivalently written (Micchelli and Pontil, 2005):

$$(S_{X_\ell} S_{X_\ell}^* + \lambda_1 I) \mathbf{c} = \tilde{\mathbf{y}},$$

where I denotes the identity operator from \mathcal{F}_y^{ℓ} to \mathcal{F}_y^{ℓ} and $S_{X_\ell} : \mathcal{H} \rightarrow \mathcal{F}_y^{\ell}$ is the sampling operator defined for every $h \in \mathcal{H}$ by: $S_{X_\ell} h = (h(x_i))_{i=1}^{\ell}$. The expression of its adjoint $S_{X_\ell}^* : \mathcal{F}_y^{\ell} \rightarrow \mathcal{H}$ of S_{X_ℓ} for every $\mathbf{c} \in \mathcal{F}_y^{\ell}$ is given by: $S_{X_\ell}^* \mathbf{c} = \sum_{i=1}^{\ell} \mathcal{K}_x(\cdot, x_i) \mathbf{c}_i$. Therefore the solution of the optimization problem in Equation (3) writes as:

$$h_{\text{ridge}}(x) = \mathcal{K}_x(x, \cdot) S_{X_\ell}^* (S_{X_\ell} S_{X_\ell}^* + \lambda_1 I)^{-1} \tilde{\mathbf{y}}.$$

3.1.2 MAXIMUM MARGIN REGRESSION

Szedmak et al. (2005) formulated a Support Vector Machine algorithm with vector output, called Maximum Margin Regression (MMR). The optimization problem of MMR in the supervised setting is the following:

$$\operatorname{argmin}_h \mathcal{J}(h) = \sum_{i=1}^{\ell} \max(0, 1 - \langle \tilde{\mathbf{y}}_i, h(x_i) \rangle_{\mathcal{F}_y}) + \lambda_1 \|h\|_{\mathcal{H}}^2. \quad (4)$$

In Szedmak et al. (2005), the function h was modeled as: $h(x) = W \varphi_x(x) + b$, where φ_x is a feature map associated to a scalar-valued kernel. In this subsection, we extend this maximum margin based regression framework to the context of the vector-valued RKHS theory by searching h in the RKHS \mathcal{H} associated to \mathcal{K}_x .

Similarly to SVM, the MMR problem (4) can be expressed according to a primal formulation that involves the optimization of $h \in \mathcal{H}$ and slack variables $\xi_i \in \mathbb{R}, i = 1, \dots, \ell$, as well as its dual formulation which is expressed according to the Lagrangian parameters $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_\ell]^T \in \mathbb{R}^\ell$. The latter leads to solve a quadratic program, for which efficient solvers exist. Both formulations are given below.

The primal form of the MMR optimization problem can be written as:

$$\begin{aligned} \min_{h \in \mathcal{H}, \{\xi_i \in \mathbb{R}\}_{i=1}^{\ell}} \quad & \lambda_1 \|h\|_{\mathcal{H}}^2 + \sum_{i=1}^{\ell} \xi_i \\ \text{s.t.} \quad & \langle \tilde{\mathbf{y}}_i, h(x_i) \rangle_{\mathcal{F}_y} \geq 1 - \xi_i, i = 1, \dots, \ell \\ & \xi_i \geq 0, i = 1, \dots, \ell. \end{aligned}$$

The Lagrangian of the above problem is given by:

$$\mathcal{L}_\alpha(h, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\eta}) = \lambda_1 \|h\|_{\mathcal{H}}^2 + \sum_{i=1}^{\ell} \xi_i - \sum_{i=1}^{\ell} \alpha_i (\langle \mathcal{K}_x(\cdot, x_i) \tilde{\mathbf{y}}_i, h \rangle_{\mathcal{H}} - 1 + \xi_i) - \sum_{i=1}^{\ell} \eta_i \xi_i,$$

with α_i and η_i being Lagrange multipliers. By differentiating the Lagrangian with respect to ξ_i and h and setting the derivatives to zero, the dual form of the optimization problem can be expressed as:

$$\begin{aligned} \min_{\boldsymbol{\alpha} \in \mathbb{R}^\ell} \quad & \frac{1}{4\lambda_1} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j \langle \tilde{\mathbf{y}}_i, \mathcal{K}_x(x_i, x_j) \tilde{\mathbf{y}}_j \rangle_{\mathcal{F}_y} - \sum_{i=1}^{\ell} \alpha_i \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq 1, i = 1, \dots, \ell \end{aligned} \quad (5)$$

and the solution \hat{h} can be written as: $\hat{h}(\cdot) = \frac{1}{2\lambda_1} \sum_{j=1}^{\ell} \alpha_j \mathcal{K}_x(\cdot, x_j) \tilde{\mathbf{y}}_j$. Note that, similarly to KDE, we retrieve the original MMR solution when using the following operator-valued kernel: $\mathcal{K}_x(x, x') = \kappa_x(x, x') I$.

In Appendix B, we derive the dual optimization problem for a general convex loss function using the Fenchel duality.

3.2 Extension to Semi-Supervised Learning

In the case of real-valued functions, Belkin et al. (2006) have introduced a novel framework, called *manifold regularization*. This approach is based on the assumption that the data lie in a low-dimensional manifold. Belkin et al. (2006) have proved a representer theorem devoted to semi-supervised learning by adding a new regularization term which exploits the information of the geometric structure. This regularization term forces the target function h to be smooth with respect to the underlying manifold. In general, the geometry of this manifold is not known but it can be approximated by a graph. In this graph, nodes correspond to labeled and unlabeled data and edges reflect the local similarities between data in the input space. For example, this graph can be built using k -nearest neighbors. The representer theorem of Belkin et al. (2006) has been extended to the case of vector-valued functions in Brouard et al. (2011) and Minh and Sindhwani (2011). In the following, we present this theorem and derive the solutions for the least-squares loss function and maximum margin regression.

Let \mathcal{L} be a convex loss function. Given a set of ℓ labeled examples $\{(x_i, \tilde{y}_i)\}_{i=1}^{\ell} \subseteq \mathcal{X} \times \mathcal{F}_y$ and an additional set of n unlabeled examples $\{x_j\}_{j=\ell+1}^{\ell+n} \subseteq \mathcal{X}$, we consider the following optimization problem:

$$\operatorname{argmin}_{h \in \mathcal{H}} \mathcal{J}(h) = \sum_{i=1}^{\ell} \mathcal{L}(h(x_i), \tilde{y}_i) + \lambda_1 \|h\|_{\mathcal{H}}^2 + \lambda_2 \sum_{i,j=1}^{\ell+n} W_{ij} \|h(x_i) - h(x_j)\|_{\mathcal{F}_y}^2, \quad (6)$$

where $\lambda_1, \lambda_2 > 0$ are two regularization hyperparameters and W is the adjacency matrix of a graph built from labeled and unlabeled data. This matrix measures the similarity between objects in the input space. We assume that the values of W are non-negative. This optimization problem can be rewritten as:

$$\operatorname{argmin}_{h \in \mathcal{H}} \mathcal{J}(h) = \sum_{i=1}^{\ell} \mathcal{L}(h(x_i), \tilde{y}_i) + \lambda_1 \|h\|_{\mathcal{H}}^2 + 2\lambda_2 \sum_{i,j=1}^{\ell+n} L_{ij} \langle h(x_i), h(x_j) \rangle_{\mathcal{F}_y},$$

where L is the graph Laplacian given by $L = D - W$ and D is the diagonal matrix of general term $D_{ii} = \sum_{j=1}^{\ell+n} W_{ij}$. Instead of the graph Laplacian, other matrices, such as iterated Laplacians or diffusion kernels (Kondor and Lafferty, 2002), can also be used.

Theorem 5 (Brouard et al. (2011); Minh and Sindhwani (2011)) *The minimizer of the optimization problem in Equation (6) admits an expansion:*

$$\hat{h}(\cdot) = \sum_{j=1}^{\ell+n} \mathcal{K}_x(\cdot, x_j) \mathbf{c}_j,$$

for some vectors $\mathbf{c}_j \in \mathcal{F}_y$, $j = 1, \dots, \ell + n$.

This theorem extends the representer theorem proposed by Belkin et al. (2006) to vector-valued functions. Besides, it also extends Theorem 3 to the semi-supervised framework.

3.2.1 SEMI-SUPERVISED PENALIZED LEAST-SQUARES

Considering the least-squares cost, the optimization problem becomes:

$$\operatorname{argmin}_{h \in \mathcal{H}} \mathcal{J}(h) = \sum_{i=1}^{\ell} \|h(x_i) - \tilde{y}_i\|_{\mathcal{F}_y}^2 + \lambda_1 \|h\|_{\mathcal{H}}^2 + 2\lambda_2 \sum_{i,j=1}^{\ell+n} L_{ij} \langle h(x_i), h(x_j) \rangle_{\mathcal{F}_y}. \quad (7)$$

Theorem 6 (Brouard et al. (2011); Minh and Sindhwani (2011)) *The coefficients $\mathbf{c}_j \in \mathcal{F}_y$, $j = 1, \dots, \ell + n$ of the expansion admitted by the minimizer \hat{h} of the optimization problem (7) satisfy this equation:*

$$J_j \sum_{i=1}^{\ell+n} \mathcal{K}_x(x_j, x_i) \mathbf{c}_i + \lambda_1 \mathbf{c}_j + 2\lambda_2 \sum_{i=1}^{\ell+n} L_{ij} \sum_{m=1}^{\ell+n} \mathcal{K}_x(x_i, x_m) \mathbf{c}_m = J_j \tilde{y}_j,$$

where $J_j \in \mathcal{B}(\mathcal{F}_y)$ is the identity operator if $j \leq \ell$ and the null operator if $\ell < j \leq (\ell + n)$.

For the proofs of Theorems 5 and 6, the reader can refer to the proofs given in the supplementary materials of Brouard et al. (2011) or Minh and Sindhwani (2011).

As in the supervised setting, the solution of the optimization problem (7) can be expressed using the sampling operator:

$$h_{\text{ridge}}(x) = \mathcal{K}_x(x, \cdot) S_{X_{\ell+n}}^* (J J^* S_{X_{\ell+n}} S_{X_{\ell+n}}^* + \lambda_1 I + 2\lambda_2 M S_{X_{\ell+n}} S_{X_{\ell+n}}^*)^{-1} J \tilde{y},$$

where the operator $J \in \mathcal{B}(\mathcal{F}_y^{\ell}, \mathcal{F}^{\ell+n})$ is defined for every $\mathbf{c} = (\mathbf{c}_j)_{j=1}^{\ell} \in \mathcal{F}_y^{\ell}$ as: $J\mathbf{c} = (\mathbf{c}_1, \dots, \mathbf{c}_{\ell}, \mathbf{0}, \dots, \mathbf{0})$. Its adjoint is defined as: $J^*(\mathbf{c}_j)_{j=1}^{\ell+n} = (\mathbf{c}_j)_{j=1}^{\ell}$. M is an operator in $\mathcal{B}(\mathcal{F}^{\ell+n})$ and each M_{ij} , $i, j \in \mathbb{N}^{\ell+n}$ is an operator in $\mathcal{B}(\mathcal{F}_y)$ equal to $L_{ij}I$.

3.2.2 SEMI-SUPERVISED MAXIMUM MARGIN REGRESSION

The optimization problem in the semi-supervised case using the hinge loss is the following:

$$\operatorname{argmin}_{h \in \mathcal{H}} \mathcal{J}(h) = \sum_{i=1}^{\ell} \max(0, 1 - \langle \tilde{y}_i, h(x_i) \rangle_{\mathcal{F}_y}) + \lambda_1 \|h\|_{\mathcal{H}}^2 + 2\lambda_2 \sum_{i,j=1}^{\ell+n} L_{ij} \langle h(x_i), h(x_j) \rangle_{\mathcal{F}_y}. \quad (8)$$

Theorem 7 *The solution of the optimization problem (8) is given by*

$$h(\cdot) = \frac{1}{2} B^{-1} \left(\sum_{i=1}^{\ell} \alpha_i \mathcal{K}_x(\cdot, x_i) \tilde{y}_i \right),$$

where $B = \lambda_1 I + 2\lambda_2 \sum_{i,j=1}^{\ell+n} L_{ij} \mathcal{K}_x(\cdot, x_i) \mathcal{K}_x(x_j, \cdot)$ is an operator from \mathcal{H} to \mathcal{H} , and α is the solution of

$$\begin{aligned} \min_{\alpha \in \mathbb{R}^{\ell}} \quad & \frac{1}{4} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j \langle \mathcal{K}_x(\cdot, x_i) \tilde{y}_i, B^{-1} \mathcal{K}_x(\cdot, x_j) \tilde{y}_j \rangle_{\mathcal{H}} - \sum_{i=1}^{\ell} \alpha_i \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq 1, i = 1, \dots, \ell. \end{aligned} \quad (9)$$

The proof of this theorem is detailed in Appendix A.

3.3 Solutions when $\mathcal{F}_y = \mathbb{R}^d$

In this subsection we consider that the dimension of \mathcal{F}_y is finite and equal to d . We first introduce the following notations:

- $\tilde{Y}_\ell = (\tilde{y}_1, \dots, \tilde{y}_\ell)$ is a matrix of size $d \times \ell$,
- $C_\ell = (\mathbf{c}_1, \dots, \mathbf{c}_\ell)$, $C_{\ell+n} = (\mathbf{c}_1, \dots, \mathbf{c}_{\ell+n})$,
- $\mathcal{K}_{X_\ell}^x = (\mathcal{K}_x(x_1, x), \dots, \mathcal{K}_x(x_\ell, x))^T$, $\mathcal{K}_{X_{\ell+n}}^x = (\mathcal{K}_x(x_1, x), \dots, \mathcal{K}_x(x_{\ell+n}, x))^T$,
- \mathbf{K}_{X_ℓ} is a $\ell \times \ell$ block matrix, where each block is a $d \times d$ matrix. The (j, k) -th block of \mathbf{K}_{X_ℓ} is equal to $\mathcal{K}_x(x_j, x_k)$,
- $\mathbf{K}_{X_{\ell+n}}$ is a $(\ell + n) \times (\ell + n)$ block matrix such that the (j, k) -th block of $\mathbf{K}_{X_{\ell+n}}$ is equal to $\mathcal{K}_x(x_j, x_k)$,
- $I_{\ell d}$ and $I_{(\ell+n)d}$ are identity matrices of size $(\ell d) \times (\ell d)$ and $(\ell + n)d \times (\ell + n)d$,
- $J = (I_\ell, 0)$ is a $\ell \times (\ell + n)$ matrix that contains an identity matrix of size $\ell \times \ell$ on the left hand side and a zero matrix of size $\ell \times n$ on the right hand side,
- \otimes denotes the Kronecker product and $\text{vec}(A)$ denotes the vectorization of a matrix A , formed by stacking the columns of A into a single column vector.

In the supervised setting, the solutions for the least-squares loss and MMR can be rewritten as $h(x) = (\mathcal{K}_{X_\ell}^x)^T C_\ell$, where C_ℓ is given by:

$$\begin{aligned} C_{\ell, \text{lsqr}} &= (\lambda_1 I_{\ell d} + \mathbf{K}_{X_\ell})^{-1} \text{vec}(\tilde{Y}_\ell), \\ C_{\ell, \text{mmr}} &= \frac{1}{2\lambda_1} \text{vec}(\tilde{Y}_\ell \text{diag}(\boldsymbol{\alpha})). \end{aligned}$$

In the semi-supervised setting, these solutions can be written as $h(x) = (\mathcal{K}_{X_{\ell+n}}^x)^T C_{\ell+n}$ where:

$$\begin{aligned} C_{\ell+n, \text{lsqr}} &= (\lambda_1 I_{(\ell+n)d} + ((J^T J + 2\lambda_2 L) \otimes I_d) \mathbf{K}_{X_{\ell+n}})^{-1} \text{vec}(\tilde{Y}_\ell J), \\ C_{\ell+n, \text{mmr}} &= (2\lambda_1 I_{(\ell+n)d} + 4\lambda_2 (L \otimes I_d) \mathbf{K}_{X_{\ell+n}})^{-1} \text{vec}(\tilde{Y}_\ell \text{diag}(\boldsymbol{\alpha}) J). \end{aligned}$$

For MMR, the vector $\boldsymbol{\alpha}$ is obtained by solving the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\alpha} \in \mathbb{R}^\ell} & \frac{1}{4} \text{vec}(\tilde{Y}_\ell \text{diag}(\boldsymbol{\alpha}) J)^T \\ & (\lambda_1 I_{(\ell+n)d} + 2\lambda_2 \mathbf{K}_{X_{\ell+n}} (L \otimes I_d))^{-1} \mathbf{K}_{X_{\ell+n}} \text{vec}(\tilde{Y}_\ell \text{diag}(\boldsymbol{\alpha}) J) - \boldsymbol{\alpha}^T \mathbf{1} \end{aligned} \quad (10)$$

s.t. $0 \leq \alpha_i \leq 1, i = 1, \dots, \ell$.

In the case of IOKR, \mathcal{F}_y is the feature space of some output kernel. Its dimension may therefore be infinite depending of which kernel is used. In this case, explicit feature vectors can be defined using the eigendecomposition of the output kernel matrix on the labeled data.

3.4 Models for General Decomposable Kernels

In the remainder of this section we propose to derive models based on a simple but powerful family of operator-valued kernels (OVK) based on scalar-valued kernels, called *decomposable kernels* or *separable kernels* (Alvarez et al., 2012; Baldassarre et al., 2012). They correspond to the simplest generalization of scalar kernels to operator-valued kernels. Decomposable kernels were first defined to deal with multi-task regression (Evgeniou et al., 2005; Michelli and Pontil, 2005) and later, with structured multi-class classification (Dinuzzo et al., 2011). Other kernels (Caponnetto et al., 2008; Álvarez et al., 2012) have also been proposed: for instance, Lim et al. (2013) introduced a Hadamard kernel based on the Hadamard product of decomposable kernels and transformable kernels to deal with nonlinear vector autoregressive models. Caponnetto et al. (2008) proved that they are universal, meaning that an operator-valued regressor built on them is a universal approximator in \mathcal{F}_y .

Proposition 8 *The class of decomposable operator-valued kernels is composed of kernels of the form:*

$$\mathcal{K}_x : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{B}(\mathcal{F}_y)$$

$$(x, x') \mapsto \kappa_x(x, x') A$$

where $\kappa_x : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a scalar-valued input kernel and $A \in \mathcal{B}(\mathcal{F}_y)$ is a positive semidefinite operator.

In the multi-task learning framework, $\mathcal{F}_y = \mathbb{R}^d$ is a finite dimensional output space and the matrix A encodes the existing relations among the d different tasks. This matrix can be estimated from labeled data or being learned simultaneously with the matrix C (Dinuzzo et al., 2011).

In the following we assume that the dimension of \mathcal{F}_y is finite and equal to d : $\mathcal{F}_y = \mathbb{R}^d$.

3.4.1 PENALIZED LEAST-SQUARES REGRESSION

In this section, we will use the following notations: \mathcal{F}_x and the function $\varphi_x : \mathcal{X} \rightarrow \mathcal{F}_x$ correspond respectively to the feature space and the feature map associated to the input scalar kernel κ_x . We note $\kappa_{X_\ell}^x = (\kappa_x(x_1, x), \dots, \kappa_x(x_\ell, x))^T$ the vector of length ℓ containing the kernel values between the labeled examples and x and $\kappa_{X_{\ell+n}}^x = (\kappa_x(x_1, x), \dots, \kappa_x(x_{\ell+n}, x))^T$. Let K_{X_ℓ} and $K_{X_{\ell+n}}$ be respectively the Gram matrices of κ_x over the sets X_ℓ and $X_{\ell+n}$. I_ℓ denotes the identity matrix of size ℓ .

The minimizer h of the optimization problem for the penalized least-squares cost in the supervised setting (3) using a decomposable OVK can be expressed as:

$$\begin{aligned} \forall x \in \mathcal{X}, h(x) &= A \sum_{i=1}^{\ell} \kappa_x(x, x_i) \mathbf{c}_i = AC_\ell \kappa_{X_\ell}^x = ((\kappa_{X_\ell}^x)^T \otimes A) \text{vec}(C_\ell) \\ &= ((\kappa_{X_\ell}^x)^T \otimes A) (\lambda_1 I_{\ell d} + K_{X_\ell} \otimes A)^{-1} \text{vec}(\tilde{Y}_\ell). \end{aligned} \quad (11)$$

Therefore, the computation of the solution h requires to compute the inverse of a matrix of size $\ell d \times \ell d$. A being a real symmetric matrix, we can write the eigendecomposition of A :

$$A = E \Gamma E^T = \sum_{i=1}^d \gamma_i \mathbf{e}_i \mathbf{e}_i^T,$$

where $E = (\mathbf{e}_1, \dots, \mathbf{e}_d)$ is a $d \times d$ matrix and Γ is a diagonal matrix containing the eigenvalues of A : $\Gamma = \text{diag}(\gamma_1, \dots, \gamma_d)$. Using the eigendecomposition of A , we can prove that the solution $\hat{h}(x)$ can be obtained by solving d independent problems.

Proposition 9 *The minimizer of the optimization problem for the supervised penalized least squares cost (3) in the case of a decomposable operator-valued kernel can be expressed as:*

$$\forall x \in \mathcal{X}, h_{\text{ridge}}(x) = \sum_{j=1}^d \gamma_j \mathbf{e}_j \mathbf{e}_j^T \tilde{Y}_\ell (\lambda_1 I + \gamma_j K_{X_j})^{-1} \kappa_{X_{\ell+1}}^x \quad (12)$$

and in the semi-supervised setting (7), it writes as

$$\forall x \in \mathcal{X}, h_{\text{ridge}}(x) = \sum_{j=1}^d \gamma_j \mathbf{e}_j \mathbf{e}_j^T \tilde{Y}_\ell J (\lambda_1 I_{\ell+n} + \gamma_j K_{X_{\ell+n}} (J^T J + 2\lambda_2 L))^{-1} \kappa_{X_{\ell+n}}^x.$$

We observe that, in the supervised setting, the complexity to solve Equation (11) is equal to $O((d\ell)^3)$, while the complexity for solving Equation (12) is $O(d^3 + \ell^3)$.

3.4.2 MAXIMUM MARGIN REGRESSION

Proposition 10 *Given $K_x(x, x') = \kappa_x(x, x')$, A , the dual formulation of the MMR optimization problem (4) in the supervised setting becomes:*

$$\begin{aligned} \min_{\alpha \in \mathbb{R}^\ell} \quad & \frac{1}{4\lambda_1} \alpha^T (\tilde{Y}_\ell^T A \tilde{Y}_\ell \circ K_{X_\ell}) \alpha - \alpha^T \mathbf{1} \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq 1, i = 1, \dots, \ell \end{aligned}$$

where \circ denotes the Hadamard product, and the solution is given by:

$$h_{\text{mmr}}(x) = \frac{1}{2\lambda_1} A \tilde{Y}_\ell \text{diag}(\alpha) \kappa_{X_\ell}^x.$$

In the semi-supervised MMR minimization problem (8), it writes as:

$$\begin{aligned} \min_{\alpha \in \mathbb{R}^\ell} \quad & \frac{1}{2} \alpha^T \left(\sum_{i=1}^d \gamma_i \tilde{Y}_\ell^T \mathbf{e}_i \mathbf{e}_i^T \tilde{Y}_\ell \circ J (2\lambda_1 I_{\ell+n} + 4\lambda_2 \gamma_i K_{X_{\ell+n}} L)^{-1} K_{X_{\ell+n}} J^T \right) \alpha - \alpha^T \mathbf{1} \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq 1, i = 1, \dots, \ell. \end{aligned}$$

The corresponding solution is:

$$h_{\text{mmr}}(x) = \frac{1}{2} \sum_{j=1}^d \gamma_j \mathbf{e}_j \mathbf{e}_j^T \tilde{Y}_\ell \text{diag}(\alpha) J (\lambda_1 I_{\ell+n} + 2\gamma_j \lambda_2 K_{X_{\ell+n}} L)^{-1} \kappa_{X_{\ell+n}}^x.$$

Proofs of Propositions 9 and 10 are given in Appendix A.

4. Model Selection

Real-valued kernel-based models enjoy a closed-form solution for the estimate of the leave-one-out criterion in the case of kernel ridge regression (Golub et al., 1979; Rifkin and Lippert, 2007). In order to select the hyperparameters of OVK-based models with a least-squares loss presented below, we develop a closed-form solution for the leave-one-out estimate of the sum of square errors. This solution extends Allen's predicted residual sum of squares (PRESS) statistics (Allen, 1974) to vector-valued functions. This result was first presented in french in the PhD thesis of Brouard (2013) in the case of decomposable kernels. In the following, we will use the notations used by Rifkin and Lippert (2007). We assume in this section that the dimension of \mathcal{F}_y is finite.

Let $\mathcal{S} = \{(x_1, \tilde{\mathbf{y}}_1), \dots, (x_\ell, \tilde{\mathbf{y}}_\ell)\}$ be the training set composed of ℓ labeled points. We define \mathcal{S}^i , $1 \leq i \leq \ell$, as the labeled data set with the i^{th} point removed:

$$\mathcal{S}^i = \{(x_1, \tilde{\mathbf{y}}_1), \dots, (x_{i-1}, \tilde{\mathbf{y}}_{i-1}), (x_{i+1}, \tilde{\mathbf{y}}_{i+1}), \dots, (x_\ell, \tilde{\mathbf{y}}_\ell)\}.$$

In this section, h_S denotes the function obtained when the regression problem is trained on the entire training set \mathcal{S} and we note $h_{\mathcal{S}^i}(x_i)$ the i^{th} leave-one-out value, that is the value at the point x_i of the function obtained when the training set is \mathcal{S}^i . The PRESS criterion corresponds to the sum of the ℓ leave-one-out square errors:

$$\text{PRESS} = \sum_{i=1}^{\ell} \|\tilde{\mathbf{y}}_i - h_{\mathcal{S}^i}(x_i)\|_{\mathcal{F}_y}^2.$$

As for scalar-valued functions, we show that it is possible to compute this criterion without evaluating explicitly $h_{\mathcal{S}^i}(x_i)$ for $i = 1, \dots, \ell$ and for each value of the grid of parameters.

Assuming we know $h_{\mathcal{S}^i}$, we define the matrix $\tilde{Y}_\ell^i = (\tilde{\mathbf{y}}_1^i, \dots, \tilde{\mathbf{y}}_\ell^i)$, where the vector $\tilde{\mathbf{y}}_j^i$ is given by:

$$\tilde{\mathbf{y}}_j^i = \begin{cases} \tilde{\mathbf{y}}_j & \text{if } j \neq i, \\ h_{\mathcal{S}^i}(x_i) & \text{if } j = i. \end{cases}$$

In the following, we show that when using \tilde{Y}_ℓ^i instead of \tilde{Y}_ℓ , the optimal solution corresponds to $h_{\mathcal{S}^i}$:

$$\begin{aligned} & \sum_{j=1}^{\ell} \|\tilde{\mathbf{y}}_j^i - h_{\mathcal{S}}(x_j)\|_{\mathcal{F}_y}^2 + \lambda_1 \|h_{\mathcal{S}}\|_{\mathcal{H}}^2 + \lambda_2 \sum_{j,k=1}^{\ell+n} W_{j,k} \|h_{\mathcal{S}}(x_j) - h_{\mathcal{S}}(x_k)\|_{\mathcal{F}_y}^2 \\ & \geq \sum_{j \neq i} \|\tilde{\mathbf{y}}_j^i - h_{\mathcal{S}}(x_j)\|_{\mathcal{F}_y}^2 + \lambda_1 \|h_{\mathcal{S}}\|_{\mathcal{H}}^2 + \lambda_2 \sum_{j,k=1}^{\ell+n} W_{j,k} \|h_{\mathcal{S}}(x_j) - h_{\mathcal{S}}(x_k)\|_{\mathcal{F}_y}^2 \\ & \geq \sum_{j \neq i} \|\tilde{\mathbf{y}}_j^i - h_{\mathcal{S}^i}(x_j)\|_{\mathcal{F}_y}^2 + \lambda_1 \|h_{\mathcal{S}^i}\|_{\mathcal{H}}^2 + \lambda_2 \sum_{j,k=1}^{\ell+n} W_{j,k} \|h_{\mathcal{S}^i}(x_j) - h_{\mathcal{S}^i}(x_k)\|_{\mathcal{F}_y}^2 \\ & \geq \sum_{j=1}^{\ell} \|\tilde{\mathbf{y}}_j^i - h_{\mathcal{S}^i}(x_j)\|_{\mathcal{F}_y}^2 + \lambda_1 \|h_{\mathcal{S}^i}\|_{\mathcal{H}}^2 + \lambda_2 \sum_{j,k=1}^{\ell+n} W_{j,k} \|h_{\mathcal{S}^i}(x_j) - h_{\mathcal{S}^i}(x_k)\|_{\mathcal{F}_y}^2. \end{aligned}$$

The second inequality comes from the fact that h_{S^i} is defined as the minimizer of the optimization problem when the i^{th} point is removed from the training set. As h_{S^i} is the optimal solution when \tilde{Y}_ℓ is replaced with \tilde{Y}_ℓ^i , it can be written as:

$$\forall i = 1, \dots, \ell, \quad h_{S^i}(x_i) = (\mathbf{K}_{X_{\ell+n}}^{x_i})^T B \text{vec}(\tilde{Y}_\ell^i) = (KB)_i \cdot \text{vec}(\tilde{Y}_\ell^i),$$

where $K = \mathbf{K}_{X_{\ell+n}(\ell+n)}$ is the input gram matrix between the sets X_ℓ and $X_{\ell+n}$ and $B = (\lambda_1 I_{(\ell+n)d} + ((J^T J + 2\lambda_2 L) \otimes I_d) \mathbf{K}_{X_{\ell+n}})^{-1} (J^T \otimes I_d)$. $(KB)_i$ corresponds to the i^{th} row of the matrix KB and $(KB)_{i,j}$ is the value of the matrix corresponding to the row i and the column j .

We can then derive an expression of h_{S^i} by computing the difference between $h_{S^i}(x_i)$ and $h_S(x_i)$:

$$\begin{aligned} h_S(x_i) - h_{S^i}(x_i) &= (KB)_i \cdot \text{vec}(\tilde{Y}_\ell^i - \tilde{Y}_\ell) \\ &= \sum_{k=1}^{\ell} (KB)_{i,k} (\tilde{Y}_k^i - \tilde{Y}_k) \\ &= (KB)_{i,i} (h_{S^i}(x_i) - \tilde{y}_i), \end{aligned}$$

which leads to

$$\begin{aligned} (I_d - (KB)_{i,i}) h_{S^i}(x_i) &= h_S(x_i) - (KB)_{i,i} \tilde{y}_i \\ \Rightarrow (I_d - (KB)_{i,i}) h_{S^i}(x_i) &= (KB)_i \cdot \text{vec}(\tilde{Y}_\ell) - (KB)_{i,i} \tilde{y}_i \\ \Rightarrow h_{S^i}(x_i) &= (I_d - (KB)_{i,i})^{-1} \left((KB)_i \cdot \text{vec}(\tilde{Y}_\ell) - (KB)_{i,i} \tilde{y}_i \right). \end{aligned}$$

Let $L_{oo} = (h_{S^1}(x_1), \dots, h_{S^i}(x_\ell))$ be the matrix containing the leave-one-out vector values over the training set. The equation above can be rewritten as:

$$\text{vec}(L_{oo}) = (I_d - \text{diag}_b(KB))^{-1} (KB - \text{diag}_b(KB)) \text{vec}(\tilde{Y}_\ell),$$

where diag_b corresponds to the block diagonal of a matrix.

The Allen's PRESS statistic can be expressed as:

$$\begin{aligned} \text{PRESS} &= \|\text{vec}(\tilde{Y}_\ell) - \text{vec}(L_{oo})\|^2 \\ &= \|(I_d - \text{diag}_b(KB))^{-1} (I_d - \text{diag}_b(KB) - KB + \text{diag}_b(KB)) \text{vec}(\tilde{Y}_\ell)\|^2 \\ &= \|(I_d - \text{diag}_b(KB))^{-1} (I_d - KB) \text{vec}(\tilde{Y}_\ell)\|^2. \end{aligned}$$

This closed-form expression allows to evaluate the PRESS criterion without having to solve ℓ problems involving the inversion of a matrix of size $(\ell + n - 1)d$.

5. Input Output Kernel Regression

We now have all the needed tools to approximate vector-valued functions. In this section, we go back to IOKR and consider that \mathcal{F}_y is the feature space associated to some output kernel $\kappa_y : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$, and that vectors \tilde{y}_i now correspond to output feature vectors $\varphi_y(y_i)$. Several feature spaces can be defined, including the unique RKHS associated to the kernel

κ_y . This choice has direct consequences on the choice of the input operator-valued kernel \mathcal{K}_x . Depending on the application, we might be interested for instance on choosing \mathcal{F}_y as a functional space to get integral operators or as the finite-dimensional Euclidean space \mathbb{R}^d to get matrices. It is important to notice that this reflects a radically new approach in machine learning where we usually focus on the choice of the input feature space and do not discuss a lot the output space. Moreover, the choice of a given triplet $(\kappa_y, \mathcal{F}_y, \mathcal{K}_x)$ has a great impact of the learning task both in terms of complexity in time and potentially of performance. In the following, we explain how Input Output Kernel Regression can be used to solve link prediction and multi-task problems.

5.1 Link Prediction

Link prediction is a challenging machine learning problem that has been defined recently in social networks as well as biological networks. Let us formulate this problem using the previous notations: $\mathcal{X} = \mathcal{Y} = \mathcal{U}$ is the set of candidate nodes we are interested in. We want to estimate some relation between these nodes, for example a social relationship between persons or some physical interaction between molecules. During the training phase we are given $\mathcal{G}_\ell = (\mathcal{U}_\ell, A_\ell)$, a non oriented graph defined by the subset $\mathcal{U}_\ell \subseteq \mathcal{U}$ and the adjacency matrix A_ℓ of size $\ell \times \ell$. Supervised link prediction is usually addressed by learning a binary pairwise classifier $f : \mathcal{U} \times \mathcal{U} \rightarrow \{0, 1\}$ that predicts if there exists a link between two objects or not, from the training information \mathcal{G}_ℓ . One way to solve this learning task is to build a pairwise classifier. However, the link prediction problem can also be formalized as an output kernel regression task (Geurts et al., 2007a; Brouard et al., 2011).

The OKR framework for link prediction is based on the assumption that an approximation of the output kernel κ_y will provide valuable information about the proximity of the objects of \mathcal{U} as nodes in the unknown graph defined on \mathcal{U} . Given that assumption, a classifier f_θ is defined from the approximation $\widehat{\kappa}_y$ by thresholding its output values:

$$f_\theta(u, u') = \text{sgn}(\widehat{\kappa}_y(u, u') - \theta).$$

An approximation of the target output kernel κ_y is built from the scalar product between the outputs of a single variable function $h : \mathcal{U} \rightarrow \mathcal{F}_y$: $\widehat{\kappa}_y(u, u') = \langle h(u), h(u') \rangle_{\mathcal{F}_y}$. Using the kernel trick in the output space therefore allows to reduce the problem of learning a pairwise classifier to the problem of learning a single variable function with output values in a Hilbert space (the output feature space \mathcal{F}_y).

In the case of IOKR, the function h is learnt in an appropriate RKHS by using the operator-valued kernel regression approach presented in Section 3. In the following, we describe the output kernel and the input operator-valued kernel that we propose to use for solving the link prediction problem with IOKR.

Regarding the output kernel, we do not have a kernel κ_y defined on $\mathcal{U} \times \mathcal{U}$ in the link prediction problem but we can define a Gram matrix K_{Y_ℓ} on the training set \mathcal{U}_ℓ . Here, we define K_{Y_ℓ} from the known adjacency matrix A_ℓ of the training graph such that it encodes the proximities in the graph between the labeled nodes. For instance, we can choose the diffusion kernel matrix (Kondor and Lafferty, 2002), which is defined as:

$$K_{Y_\ell} = \exp(-\beta L_{y_\ell}),$$

$B =$	Supervised learning	Semi-supervised learning
Ridge	$(\lambda I_\ell + K_X)^{-1}$	$J(\lambda I_{\ell+n} + K_{X_{\ell+n}})^{-1}$
MMR	$\frac{1}{2\lambda} \text{diag}(\alpha)$	$\frac{1}{2} \text{diag}(\alpha) J(\lambda I_{\ell+n} + 2\lambda_2 K_{X_{\ell+n}} L)^{-1}$

Table 2: Matrix B of the models obtained using the identity decomposable kernel in the case of different settings and loss functions.

where $L_{y_\ell} = D_\ell - A_\ell$ is the graph Laplacian, with D_ℓ the diagonal matrix of degrees. The kernel trick allows to work as if we have chosen the subspace spanned by $\{\mathbf{e}_1, \dots, \mathbf{e}_\ell\}$, the eigenvectors of the matrix K_Y as a feature space with an associated feature map that verifies:

$$\forall i \in \{1, \dots, \ell\}, \varphi_{y_\ell}(u_i) = [\sqrt{\gamma_1} \mathbf{e}_1^i, \dots, \sqrt{\gamma_\ell} \mathbf{e}_\ell^i]^T.$$

The kernel $\kappa_{y_\ell} : \mathcal{U} \times \mathcal{U} \rightarrow \mathbb{R}$, also verifies:

$$\forall i, j \in \{1, \dots, \ell\}, \kappa_{y_\ell}(u_i, u_j) = (K_Y)_{i,j}.$$

In practise, we only need to know K_Y . Regarding the operator-valued kernel, we consider here the identity decomposable kernel:

$$\forall (u, u') \in \mathcal{U} \times \mathcal{U}, \mathcal{K}_x(u, u') = \kappa_x(u, u') I.$$

We underline that even if this kernel may seem simple, we must be aware that in this task, we do not have the explicit expressions of outputs $\varphi_y(u)$ and prediction in \mathcal{F}_y is not the final target. Therefore this operator-valued kernel allows us to work properly with output Gram matrix values.

Of particular interest for us is the expression of the scalar product which is the only one we need for link prediction. When using the identity decomposable kernel, the approximation of the output kernel can be written as follows:

$$\widehat{\kappa}_{y_\ell}(u, u') = \langle \hat{h}(u), \hat{h}(u') \rangle_{\mathcal{F}_y} = (\kappa_{x_\ell}^u)^T B^T K_Y B \kappa_{x_\ell}^{u'} \quad (13)$$

where B is a matrix of size $\ell \times \ell$ that depends of the loss function used (see Table 2). In the semi-supervised setting, the approximated output kernel has a similar expression, where $\kappa_{x_\ell}^u, \kappa_{x_\ell}^{u'}$ are replaced by $\kappa_{x_\ell}^u, \kappa_{x_\ell}^{u'}$ and B is a matrix of size $(\ell + n) \times \ell$. We can notice that we do not need to know the explicit expressions of outputs $\varphi_y(u)$ to compute this scalar product. Besides, the approximation of the scalar product $\langle \varphi_y(u), \varphi_y(u') \rangle_{\mathcal{F}_y}$ can be interpreted as a modified scalar product between the inputs $\varphi_x(u)$ and $\varphi_x(u')$.

5.2 Multi-Task Learning

Multi-task learning has been developed based on the observation that it may happen that several learning tasks are not disjoint and are characterized by a relationship such as inclusion or similarity. Learning simultaneously such related tasks has been shown to improve the performance comparing to learning the different tasks independently from each other

(Carrara, 1997; Eygenion et al., 2005). Examples of multi-task learning problems can be found in document categorization as well as in protein functional annotation prediction. Dependencies among target variables can also be encountered in the case of multiple regression.

We consider here the case of learning d tasks having the same input and output domains. Eygenion et al. (2005) have shown that this problem is equivalent to learning a vector-valued function $h : \mathcal{X} \rightarrow \mathcal{Y}$ with d components $h^i : \mathcal{X} \rightarrow \mathcal{Y}_i$ using the vector-valued RKHS theory. A natural way to integrate the task relatedness with operator-valued kernels is to use the decomposable kernels introduced in Subsection 3.4: $\mathcal{K}_x(x, x') = \kappa_x(x, x') A$. Several values for the matrix A have been proposed (Eygenion et al., 2005; Sheldon, 2008; Balassare et al., 2012) based on the fact that the regularization term in the RKHS associated to a decomposable OVK can be expressed in function of A :

$$\|h\|_{\mathcal{H}}^2 = \sum_{i,j=1}^d A_{i,j}^1 \langle h^i, h^j \rangle_{\mathcal{H}_{\kappa_x}},$$

where \dagger denotes the pseudoinverse and \mathcal{H}_{κ_x} the RKHS associated to the scalar kernel κ_x .

In the IOKR framework, the task structure can be encoded in two different ways. We can use a decomposable OVK in input as described previously and define a regularization term that will penalize the d components of the function h according to the task structure. Another way is to modify the output representation by defining an output kernel that will integrate the task structure. We propose to compare the three following models to solve multi-task learning with our framework:

- Model 0: $\kappa_y(\mathbf{y}, \mathbf{y}') = \mathbf{y}^T \mathbf{y}'$, with the identity kernel $\mathcal{K}_x(x, x') = \kappa_x(x, x') I$,
- Model 1: $\kappa_y(\mathbf{y}, \mathbf{y}') = \mathbf{y}^T A_1 \mathbf{y}'$, with the identity kernel $\mathcal{K}_x(x, x') = \kappa_x(x, x') I$,
- Model 2: $\kappa_y(\mathbf{y}, \mathbf{y}') = \mathbf{y}^T \mathbf{y}'$, with the decomposable kernel $\mathcal{K}_x(x, x') = \kappa_x(x, x') A_2$.

In the first case, the different tasks are learned independently :

$$\forall x \in \mathcal{X}, \hat{h}_0(x) = Y_\ell J (\lambda_1 I_{\ell+n} + K_{X_{\ell+n}})^{-1} \kappa_{X_{\ell+n}}^x,$$

while in the other cases, the tasks relatedness is taken into account :

$$\begin{aligned} \forall x \in \mathcal{X}, \hat{h}_1(x) &= \sqrt{\lambda_1} Y_\ell J (\lambda_1 I_{\ell+n} + K_{X_{\ell+n}})^{-1} \kappa_{X_{\ell+n}}^x, \\ \forall x \in \mathcal{X}, \hat{h}_2(x) &= \sum_{j=1}^d \gamma_j \mathbf{e}_j \mathbf{e}_j^T Y_\ell J (\lambda_1 I_{\ell+n} + \gamma_j K_{X_{\ell+n}})^{-1} \kappa_{X_{\ell+n}}^x, \end{aligned}$$

where γ_j and \mathbf{e}_j are the eigenvalues and eigenvectors of A_2 .

We consider a matrix M of size $d \times d$ that encodes the relations existing between the different tasks. This matrix can be considered as the adjacency matrix of a graph between tasks. We note L_M the graph Laplacian associated to this matrix. The matrices A_1 and A_2 are defined as follow:

$$\begin{aligned} A_1 &= \mu M + (1 - \mu) I_d, \\ A_2 &= (\mu L_M + (1 - \mu) I_d)^{-1}, \end{aligned}$$

where μ is a parameter in $[0, 1]$.

The matrix A_2 was proposed by Evgeniou et al. (2005) and Sheldon (2008) for multi-task learning from the following regularizer:

$$\|h_2\|_{\mathcal{H}}^2 = \frac{\mu}{2} \sum_{i,j=1}^d M_{ij} \|h_2^i - h_2^j\|^2 + (1 - \mu) \sum_{i=1}^d \|h_2^i\|^2.$$

This regularization term forces two tasks h_2^i and h_2^j to be close to each other when the similarity value M_{ij} is high and conversely.

6. Numerical Experiments

In this section, we present the performances obtained with the IOKR approach on two different problems: link prediction and multi-task regression. In these experiments, we examine the effect of the smoothness constraint through the variation of its related hyperparameter λ_2 , using supervised method as a baseline. We evaluate the method in the transductive setting, in which the goal is to predict the correct outputs for the unlabeled examples, as well as in the semi-supervised setting.

6.1 Link Prediction

For the link prediction problem, we considered experiments on three datasets: a collection of synthetic networks, a co-authorship network and a protein-protein interaction (PPI) network.

6.1.1 PROTOCOL

For different percentages of labeled nodes, we randomly selected a subsample of nodes as labeled nodes. We split the remaining nodes in two subsets: one containing the unlabeled nodes and another containing the test nodes. Labeled interactions correspond to interactions between two labeled nodes. This means that when 10% of labeled nodes are selected, it corresponds to only 1% of labeled interactions. The performances were evaluated by averaging the areas under the ROC curve and the precision-recall curve (denoted AUC-ROC and AUC-PR) over ten random choices of the labeled set. A Gaussian kernel was used for the scalar input kernel κ_x . Its corresponding bandwidth σ was selected by a leave-one-out cross-validation procedure on the training set to maximize the AUC-ROC, jointly with the hyperparameter λ_1 . In the case of the least-squares loss function, we used the leave-one-out estimates approach introduced in Section 4. The output kernel used is a diffusion kernel of parameter β . Another diffusion kernel of parameter β_2 was also used for the smoothing penalty: $\exp(-\beta_2 L) = \sum_{i=0}^{\infty} \frac{(-\beta_2 L)^i}{i!}$, where L is the Laplacian of W . Preliminary runs have shown that the values of β and β_2 have a limited influence on the performances, we then have set both parameters to 1. Finally we set W to $K_{X_{t+n}}$.

6.1.2 SYNTHETIC NETWORKS

We first illustrate our method on synthetic networks where the input kernel was chosen as a very good approximation of the output kernel. In these experiments we wanted to measure

the improvement brought by the semi-supervised method in extreme cases, i.e. when the percentage of labeled nodes is very low.

The output networks were obtained by sampling random graphs containing 700 nodes from an Erdős-Rényi law with different graph densities. The graph density corresponds to the probability of presence of edges in the graph. In this experiment we chose three densities that are representative of real network densities: 0.007, 0.01 and 0.02. For each network, we used the diffusion kernel on the full graph as output kernel and chose the diffusion parameter such that it maximizes an information criterion. To built an input kernel corresponding to a good approximation of the output kernel, we applied kernel PCA on the output kernel and derived input vectors from the truncated basis of the first components. We can control the quality of the input representation by varying the relative inertia captured by the first components. We then build a Gaussian kernel based on these inputs.

Figures 3 and 4 report respectively the averaged values and standard deviations for the AUC-ROC and AUC-PR obtained for different network densities and different percentages of labeled nodes in the transductive setting. IOKR-ridge corresponds to IOKR with a least-square loss and IOKR-margin to the hinge loss used in MMR. For these results, we used the components capturing 95% of the variance for defining the input vectors. We observe that IOKR-ridge outperforms IOKR-margin in the supervised and in the semi-supervised cases. This improvement is particularly significant for AUC-PR, especially when the network density is strong and the percentage of labeled data is high. It is thus very significant for 10% and 20% of labeled data. In the supervised case, this observation can be explained by the difference between the complexities of the models. As shown in Equation (13), the solution obtained in the supervised case writes as: $\kappa_{\tilde{y}}(u, u') = (\kappa_{x, U_{\ell}}^u)^T B^T K_Y B \kappa_{x, U_{\ell}}^{u'}$. In Table 2, we can see that the matrix B is only a diagonal matrix in the case of IOKR-margin while B is a full matrix for IOKR-ridge. This can also be seen in the dual optimization problem for a general loss function (see Appendix B), where we observe that the dual variables α_i are simply collinear to the vector \tilde{y}_i for IOKR-margin. The synthetic networks may therefore require a more complex predictor.

We observe an improvement of the performances in terms of AUC-ROC and AUC-PR for both approaches in the semi-supervised setting compared to the supervised setting. This improvement is more significant for IOKR-margin. This can be explained by the fact that the IOKR-margin models obtained in the supervised and in the semi-supervised cases do not have the same complexity. As shown in Table 2, the matrix B of the IOKR-margin model is a much richer matrix in the semi-supervised setting than in the supervised setting where it corresponds to a diagonal matrix. For IOKR-ridge, the improvement of the performance is only observed for low percentages of labeled data. We can therefore make the assumption that for this model, using unlabeled data increases the AUCs for low percentages of labeled data. But when enough information can be found in the labeled data, semi-supervised learning does not improve the performance. Based on these results, we can also formulate the assumption that link prediction is harder in the case of dense networks.

In Appendix C we experimented how the method behaves with perfect to noisy input features. We chose different levels of inertia (75%, 85%, 95% and 100%) for defining the input features. The results obtained with IOKR-ridge and IOKR-margin are shown in Table 8. We also include results on synthetic networks generated using mixtures of Erdős-Rényi random graphs in Table 9.

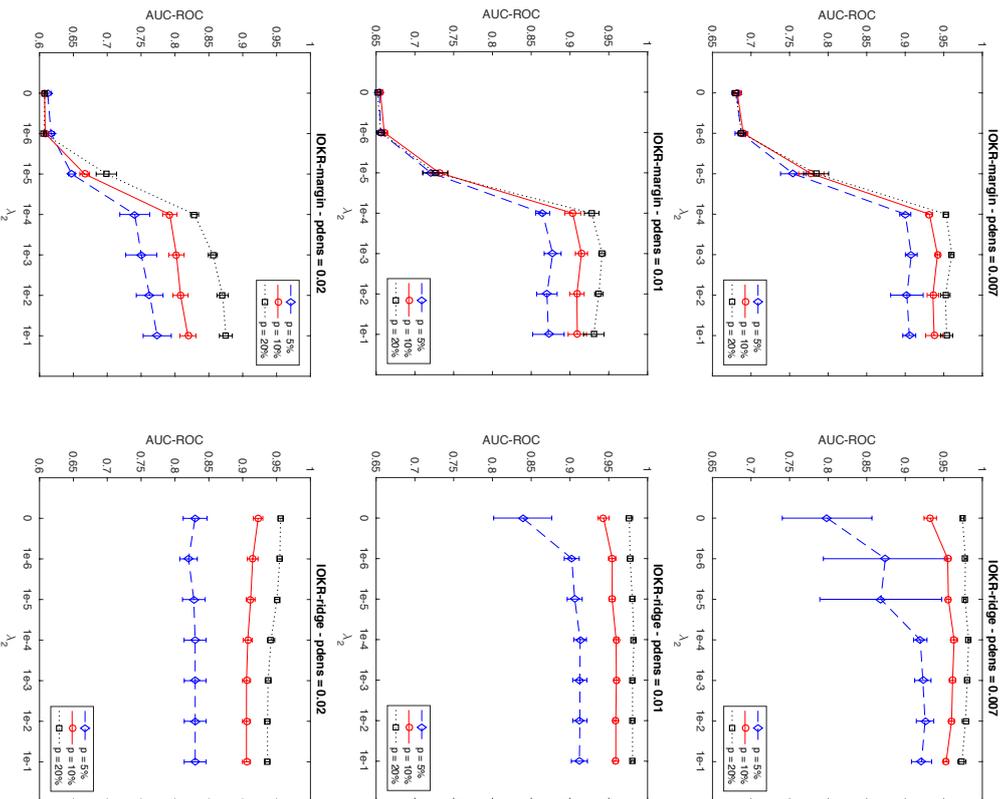


Figure 3: Averaged AUC-ROC for the reconstruction of three synthetic networks with IOKR-margin (left) and IOKR-ridge (right) in the transductive setting. The rows correspond to different graph densities (denoted pdens), which are 0.007, 0.01 and 0.02 respectively.

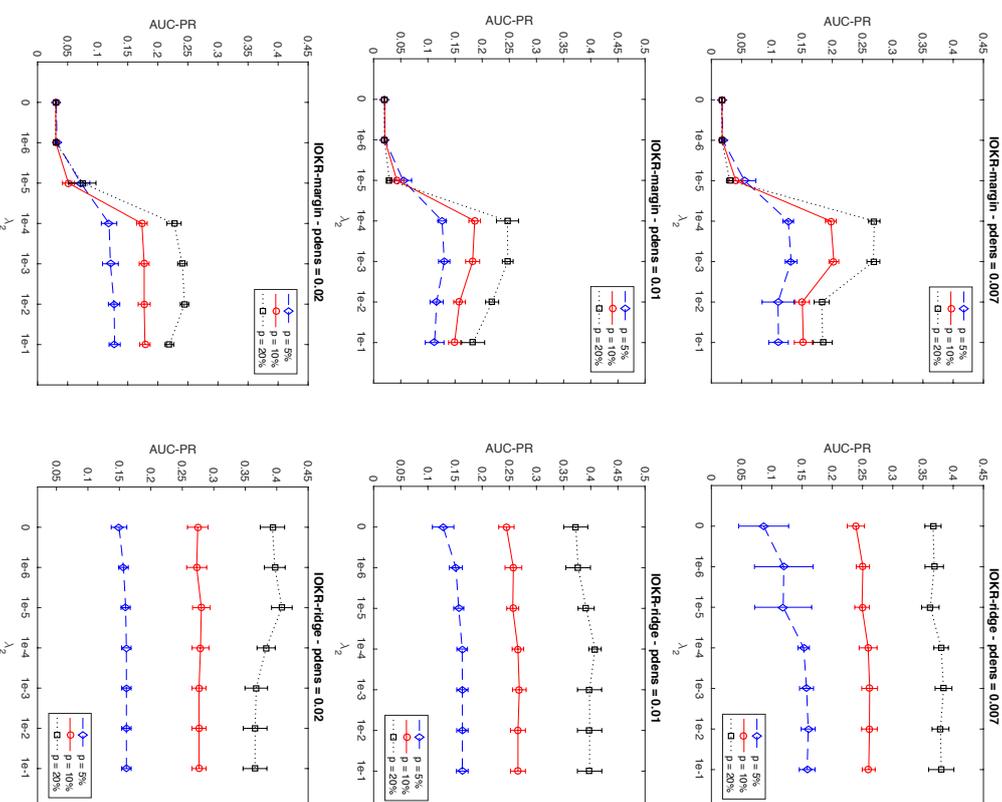


Figure 4: Averaged AUC-PR for the reconstruction of three synthetic networks with IOKR-margin (left) and IOKR-ridge (right) in the transductive setting. The rows correspond to different graph densities (denoted pdens), which are 0.007, 0.01 and 0.02 respectively.

6.1.3 NIPS CO-AUTHORSHIP NETWORK

We applied our method on a co-authorship network containing information on publications of the NIPS conferences between 1988 to 2003 (Globerson et al., 2007). In this network, vertices represent authors and an edge connects two authors if they have at least one NIPS publication in common. Among the 2865 authors, we considered the ones with at least two links in the co-authorship network in order to have a significant density and try to keep close to the original data. We therefore focused on a network containing 2026 authors with an empirical link density of 0.002. Each author was described by a vector of 14036 values, corresponding to the frequency with which he uses each given word in his papers.

Figure 5 reports the averaged AUC-ROC and AUC-PR obtained on the NIPS co-authorship network in the transductive setting for different values of λ_2 and different percentages of labeled nodes. As previously, we observe that the semi-supervised approach

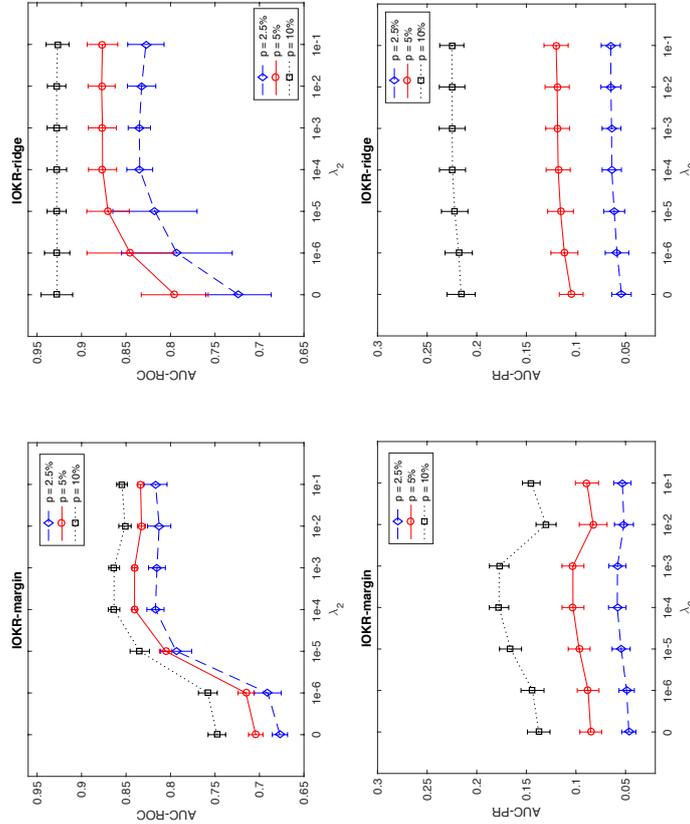


Figure 5: AUC-ROC and AUC-PR obtained for the NIPS co-authorship network inference with IOKR-margin (left) and IOKR-ridge (right) in the transductive setting.

p	AUC-ROC		AUC-PR			
	5%	20%	5%	20%		
Transductive setting						
EM	87.3 ± 2.4	92.9 ± 1.7	96.4 ± 0.8	13.8 ± 4.5	22.5 ± 6.6	41.1 ± 2.5
PKMR	85.7 ± 4.1	92.4 ± 1.6	96.4 ± 0.4	9.7 ± 2.8	20.0 ± 4.8	38.8 ± 2.0
IOKR	83.6 ± 5.9	93.6 ± 1.0	96.5 ± 0.4	12.0 ± 3.0	24.5 ± 2.9	43.7 ± 1.9
Semi-supervised setting						
IOKR	86.0 ± 2.7	93.3 ± 0.7	95.7 ± 1.4	7.6 ± 2.3	13.8 ± 1.7	25.3 ± 3.0

Table 3: AUC-ROC and AUC-PR obtained for the NIPS co-authorship network inference with EM, PKMR, IOKR in the transductive setting, and with IOKR in the semi-supervised setting. p indicates the percentage of labeled examples.

improves the performances compared to the supervised one for both models. For AUC-ROC values, this improvement is especially important when the percentage of labeled nodes is low. Indeed, with 2.5% of labeled nodes, the improvement can reach in average up to 0.14 points of AUC-ROC for IOKR-margin and up to 0.11 points for IOKR-ridge. As for the synthetic networks, the IOKR-ridge model outperforms IOKR-margin model in terms of AUC-ROC and AUC-PR, especially when the proportion of labeled examples is large. The explanation provided for the synthetic networks regarding the complexity of the solutions for IOKR-margin and IOKR-ridge holds here also. In the following, we will focus on IOKR-ridge only.

We compared IOKR-ridge with two transductive approaches: the EM-based approach (Tsuda et al., 2003; Kato et al., 2005) and Penalized Kernel Matrix Regression (PKMR) (Yamanishi and Vert, 2007). These two methods regard the link prediction problem as a kernel matrix completion problem. The EM method fills the missing entries of the output Gram matrix K_Y by minimizing the information geometry, as measured by the Kullback-Leibler divergence, with the input Gram matrix K_X . The PKMR approach considers the kernel matrix completion problem as a regression problem between the labeled input Gram matrix K_{Y_L} and the labeled output Gram matrix K_{Y_U} . We did not compare our method with the *Link Propagation* framework (Kashima et al., 2009) because this framework assumes that arbitrary interactions may be considered as labeled while IOKR requires a known subgraph. 500 examples were used for the test set and the remaining 1526 examples for the training set, which corresponds to the union of the labeled and unlabeled sets. For the labeled set we used 5%, 10% and 20% of the training examples and the left examples for the unlabeled set. We averaged the AUC over ten random partitions of the examples in labeled/unlabeled/test sets. The hyperparameters were selected by a 5-CV experiment on the labeled set for the three methods. The hyperparameters were selected separately for the AUC-ROC and the AUC-PR. For IOKR, we selected also the λ_2 parameter in this experiment, and we sparsified the matrix W in the semi-supervised constraint using 10% of the k -nearest-neighbors.

The results obtained for the comparison in the transductive and semi-supervised setting are reported in Table 3. Only the results for IOKR-ridge are reported in the semi-supervised

setting as the other methods are transductive. We observe that IOKR obtains better AUC-ROC and AUC-PR than EMI and PKMR when the percentage of labeled data is greater or equal than 10%. For 5% the best performing method is the EMI approach. Regarding the results of IOKR in the semi-supervised setting, we observe that the AUC-ROC results stay relatively similar compared to the transductive setting. However the AUC-PR values decrease significantly between the transductive and the semi-supervised settings. Considering the proteins for which we want to predict the interactions in the continuity constraint seems to help a lot the performances in term of AUC-PR.

6.1.4 PROTEIN-PROTEIN INTERACTION NETWORK

We also performed experiments on a protein-protein interaction (PPI) network of the yeast *Saccharomyces Cerevisiae*. This network was built using the DIP database (Salwinski et al., 2004), which contains protein-protein interactions that have been experimentally determined and manually curated. We used more specifically the high confidence DIP core subset of interactions (Deane et al., 2002). For the input kernels, we used the annotations provided by Gene Ontology (GO) (Ashburner et al., 2000) in terms of biological processes, cellular components and molecular functions. These annotations are organized in three different ontologies. Each ontology is represented by a directed acyclic graph, where each node is a GO annotation and edges correspond to relationships between the annotations, like sub-class relationships for example. A protein can be annotated to several terms in an ontology. We chose to represent each protein u_i by a vector s_i , whose dimension is equal to the total number of terms in the considered ontology. If a protein u_i is annotated by the term t , then :

$$s_i^{(t)} = -\ln \left(\frac{\text{number of proteins annotated by } t}{\text{total number of proteins}} \right).$$

This encoding allows to take into account the specificity of a term in the ontology. We then used these representations to build a Gaussian kernel for each GO ontology. By considering the set of proteins being annotated for each input kernel and being involved in at least one physical interaction, we obtained a PPI network containing 1242 proteins.

Based on the previous numerical results, we chose to consider only IOKR-ridge in the following experiments. We compared our approach to several supervised methods proposed for biological network inference:

- *Naive* (Yamanishi et al., 2004): this approach predicts an interaction between two proteins u and u' if $\kappa_{x_i}(u, u')$ is greater than a threshold θ .
- *kCCA* (Yamanishi et al., 2004): kernel CCA is used to detect correlations existing between the input kernel and a diffusion kernel derived from the adjacency matrix of the labeled PPI network.
- *kML* (Vert and Yamanishi, 2005): kernel Metric Learning consists in learning a new metric such that interacting proteins are close to each other, and conversely for non interacting proteins.
- *Local* (Bleakley et al., 2007): a local model is built for each protein in order to learn the subnetwork associated to each protein and these models are then combined together.

- *OK3+ET* (Gauts et al., 2006, 2007a): Output Kernel Tree with extra-trees is a tree-based method where the output is kernelized and is combined with ensemble methods.

The pairwise kernel method (Ben-Hur and Noble, 2005) was not considered here because this method requires to define a Gram matrix between pairs of nodes, which raises some practical issues in terms of computation time and storage. However we could have used an online implementation of the pairwise kernel method like the one used in Kashima et al. (2009) and thus avoid to store the large Gram matrix.

Each method was evaluated through a 5-fold cross-validation (5-CV) experiment and the hyperparameters were tuned on the training fold using a 4-CV experiment. As the local method can not be used for predicting interactions between two proteins of the test set, AUC-ROC and AUC-PR were only computed for the prediction of interactions between proteins in the test set and proteins in the training set. Input kernel matrices were defined for GO ontology and an integrated kernel, which was obtained by averaging the three input kernels, was also considered. Table 4 reports the results obtained for the comparison of the different methods in the supervised setting. We can see that output kernel regression based methods work better on this dataset than the other methods. In terms of AUC-ROC, the IOKR-ridge method obtains the best results for the four different input kernels, while for AUC-PR, OK3 with extra-trees presents better performances. We also compared the aver-

a) AUC-ROC:

Method	GO-BP	GO-CC	GO-MF	int
Naive	60.8 ± 0.8	64.4 ± 2.5	64.2 ± 0.8	67.7 ± 1.5
kCCA	82.4 ± 3.6	77.0 ± 1.7	75.0 ± 0.6	85.7 ± 1.6
kML	83.2 ± 2.4	77.8 ± 1.1	76.6 ± 1.9	84.5 ± 1.5
Local	79.5 ± 1.6	73.1 ± 1.3	66.8 ± 1.2	83.0 ± 0.5
OK3+ET	84.3 ± 2.4	81.5 ± 1.6	79.3 ± 1.8	86.9 ± 1.6
IOKR-ridge	88.8 ± 1.9	87.1 ± 1.3	84.0 ± 0.6	91.2 ± 1.2

b) AUC-PR:

Method	GO-BP	GO-CC	GO-MF	int
Naive	4.8 ± 1.0	2.1 ± 0.6	2.4 ± 0.4	8.0 ± 1.7
kCCA	7.1 ± 1.5	7.7 ± 1.4	4.2 ± 0.5	9.9 ± 0.4
kML	7.1 ± 1.3	3.1 ± 0.6	3.5 ± 0.4	7.8 ± 1.6
Local	6.0 ± 1.1	1.1 ± 0.3	0.7 ± 0.0	22.6 ± 6.6
OK3+ET	19.0 ± 1.8	21.8 ± 2.5	10.5 ± 2.0	26.8 ± 2.4
IOKR-ridge	15.3 ± 1.2	20.9 ± 2.1	8.6 ± 0.3	22.2 ± 1.6

Table 4: AUC-ROC and AUC-PR estimated by 5-CV for the yeast PPI network reconstruction in the supervised setting with different input kernels (*GO-BP*: GO biological processes; *GO-CC*: GO cellular components; *GO-MF*: GO molecular functions; *int*: average of the different kernels).

Method	Running time (s)
Naive	0.05 ± 0.01
kCCA	144.60 ± 70.70
kML	18.28 ± 0.78
Local	141.64 ± 14.91
OK3+ET	638.53 ± 69.09
IOKR-ridge	0.49 ± 0.02

Table 5: Averaged running time in seconds for one fold of the 5-CV for the yeast PPI network reconstruction in the supervised setting.

aged running time for one fold of the 5-CV for the different methods. These running times are shown in Table 5 and correspond to the times needed to perform both the training and the prediction steps. All the algorithms were implemented in Matlab and run on a MacBook Pro 2.4 GHz dual-core. For this computation, we fixed the values of the parameters and we did not take into account the computation of the input kernel. In Table 5 we observe that the running time of IOKR-ridge is relatively small compared to the other methods. The fastest method is the naive method. It can be explained by the fact that this method does not have a training step like the other methods. Interactions between two proteins are simply predicted if the proteins are similar according to the input kernel function.

As for the NIPS co-authorship network, we compared IOKR-ridge with the EM and PKMR approaches in the transductive setting. For this network, we used 300 examples for the test set and the remaining 942 examples for the training set. We used as input kernel the integrated kernel introduced in the supervised experiments. The results obtained for this comparison as well as the results for IOKR-ridge in the semi-supervised setting are reported in Table 6. Regarding the AUC-ROC, the EM approach obtains better results when the percentage of labeled data is equal to 5%. For 10% and 20% of labeled data, the difference between EM and IOKR-ridge is not significant. In terms of AUC-PR, EM achieves rather good performances compared to the others, especially for 5% and 10% of labeled data.

p	AUC-ROC			AUC-PR		
	5%	10%	20%	5%	10%	20%
Transductive setting						
EM	82.2 ± 0.6	82.9 ± 0.6	84.6 ± 0.6	15.7 ± 1.4	16.5 ± 2.7	19.7 ± 0.7
PKMR	77.5 ± 2.3	80.8 ± 1.1	83.9 ± 1.2	6.1 ± 1.5	9.8 ± 1.8	13.8 ± 1.2
IOKR	80.6 ± 0.7	83.1 ± 0.5	83.9 ± 0.5	7.1 ± 1.1	11.7 ± 1.1	17.8 ± 1.5
Semi-supervised setting						
IOKR	81.0 ± 1.1	82.9 ± 1.2	83.8 ± 1.0	6.6 ± 1.6	10.5 ± 1.3	16.0 ± 2.3

Table 6: AUC-ROC and AUC-PR obtained for the PPI network inference with EM, PKMR, IOKR in the transductive setting, and with IOKR in the semi-supervised setting.

p	5%	10%	20%
EM	0.19 ± 0.01	0.19 ± 0.00	0.22 ± 0.04
PKMR	0.16 ± 0.01	0.18 ± 0.00	0.19 ± 0.01
IOKR	1.61 ± 0.01	1.63 ± 0.01	1.76 ± 0.22

Table 7: Averaged running time in seconds of one repetition for the yeast PPI network reconstruction in the transductive setting.

However we can notice that the EM-based approach is purely transductive while IOKR-ridge learns a function and can therefore be used in the semi-supervised learning, which is more general. Regarding the IOKR-ridge results in the semi-supervised setting, we observe that the performances are very similar to the ones obtained in the transductive setting. We also compared in Table 7 the averaged running time for different percentages of labeled data. We can first observe that the three methods have a small running time (less than 2 seconds). EM and PKMR are a bit faster than IOKR-ridge as these methods require to inverse a matrix of size $\ell \times \ell$ while IOKR-ridge needs to inverse a matrix of size $(\ell+n) \times (\ell+n)$.

6.2 Application to Multi-Task Regression

In the following, we compare the behavior of the three models proposed for multi-task learning with IOKR in Section 5 on a drug activity prediction problem. The goal of this problem is to predict the activities of molecules in different cancer cell lines (cancer types). This problem has potential applications in cancer drug discovery. In this application, \mathcal{X} corresponds to the set of molecules and $\mathcal{Y} = \mathcal{F}_y = \mathbb{R}^d$, where d is the number of cell lines.

6.2.1 DATASET

We used the dataset of Su et al. (2010) that contains the biological activities of molecules against a set of 59 human cancer cell lines. These data have been extracted from the NCI-Cancer dataset. We used the "No-Zero-Active" version of the dataset which contains the 2303 molecules that are all active against at least one cell line. Each molecule is represented by a graph, where nodes correspond to atoms and edges to bonds between atoms. The Tanimoto kernel (Ralaivola et al., 2005) is used for the scalar input kernel:

$$\kappa_x(x, x') = \frac{k_m(x, x')}{k_m(x, x) + k_m(x', x') - k_m(x, x')}.$$

In this application, k_m is chosen as a linear path kernel. The corresponding input feature vectors $\varphi_{x_m}(x)$ are binary vectors that indicate the presences and absences in the molecules of all existing paths containing a maximum of m bonds. In this experiment, the value of m was set to 6.

6.2.2 PROTOCOL

We evaluated the behavior of the IOKR-ridge model in the transductive setting. The performances were measured by computing the mean squared error (MSE) on the unlabeled

set:

$$MSE = \frac{1}{n} \sum_{i=\ell+1}^{\ell+n} \|h(x_i) - \varphi_j(\mathbf{Y}^i)\|_2^2.$$

We estimated the task structure between the cancer types by comparing the molecular activities associated to each cancer type on the training set:

$$M_{ij} = \exp\left(-\gamma \|Y_\ell^i - Y_\ell^j\|^2\right), \quad i, j = 1, \dots, d,$$

where $Y_\ell^i = (\mathbf{Y}_1^i, \mathbf{Y}_2^i, \dots, \mathbf{Y}_\ell^i)$.

The parameter γ of the matrix M was chosen to maximize an information criterion and the regularization parameter λ_1 was set to 1. Regarding the matrix W used in the semi-supervised term, we sparsified the Gram matrix $K_{\ell+\ell+n}$ of the scalar input kernel κ_x using a k -nearest neighbors procedure with $k = 50$. We then computed the graph Laplacian of the obtained graph and considered the Laplacian iterated to degree 5.

6.2.3. RESULTS

The results presented in Figure 6 were obtained from ten random choices of the training set. The performances obtained with model 1 and model 2 for different percentages of labeled data are represented as a function of the parameters μ and λ_2 .

We observe on this figure that for both models, using unlabeled data helps to improve the performances. We also observe that when μ is increased from 0 to 0.8 or 1, the mean squared errors are decreased. The obtained results therefore show the benefit of taking into account the relationships existing between the outputs for both models and both settings (supervised and semi-supervised).

We reported on Figure 7 the MSE obtained with models 1 and 2 for the best parameter μ and added the results obtained with the model 0, which corresponds to the case where $A = I$. We observe on this figure that the model 2 obtains better results than the model 1 when the percentage of labeled data is small ($p = 5\%$). For $p = 10\%$, the two models behave similarly, while for 20% of labeled data, the model 1 improves significantly the performances, compared to model 2. Therefore, we observe that using the output structure information either in the input operator-valued kernel or in the output kernel leads to different results. And depending on the amount of labeled data, one of the two models can be more interesting to use.

7. Conclusion and Perspectives

Operator-valued kernels and the associated RKHS theory provide a general framework to address approximation of functions with values in some Hilbert space. When characterizing the output Hilbert space as a feature space related to some real-valued scalar kernel, we get an original framework to deal with structured outputs. Extending our previous work (Brouard et al., 2011) which introduced a new representer theorem for semi-supervised learning with vector-valued functions, we presented solutions of semi-supervised penalized regression developed for two empirical loss functions, the square loss and the hinge loss in the general case and in the special case of decomposable kernels using tensors. We also showed

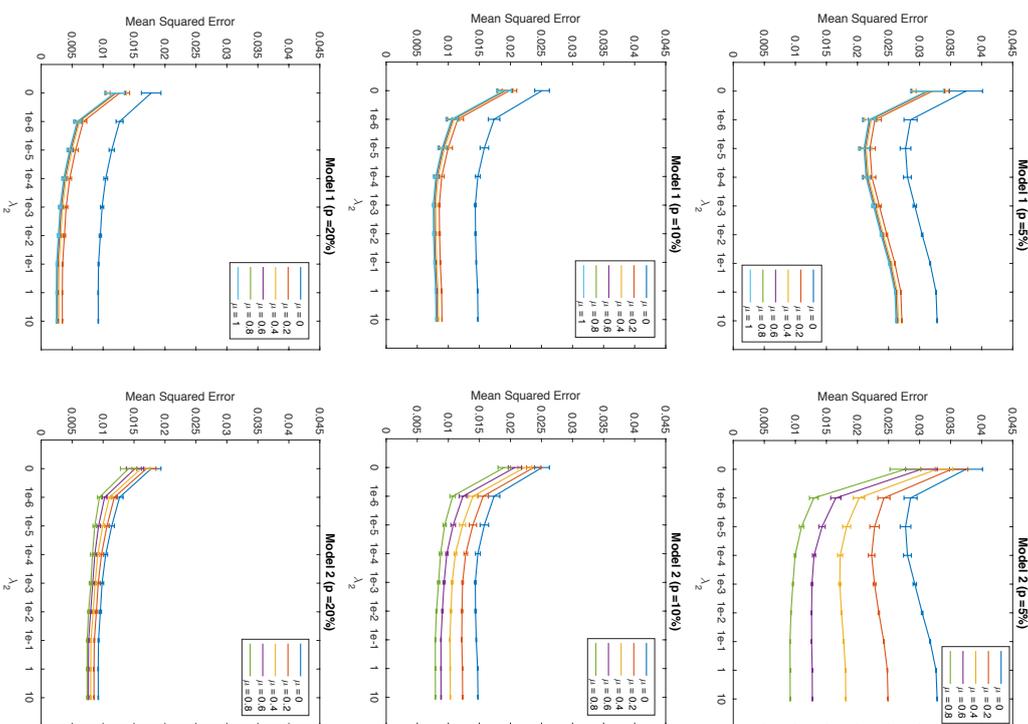


Figure 6: Mean squared errors obtained with the two models for the prediction of molecular activities. The results are averaged over ten random choices of the training set and are given for different percentages of labeled data (5%, 10% and 20%).

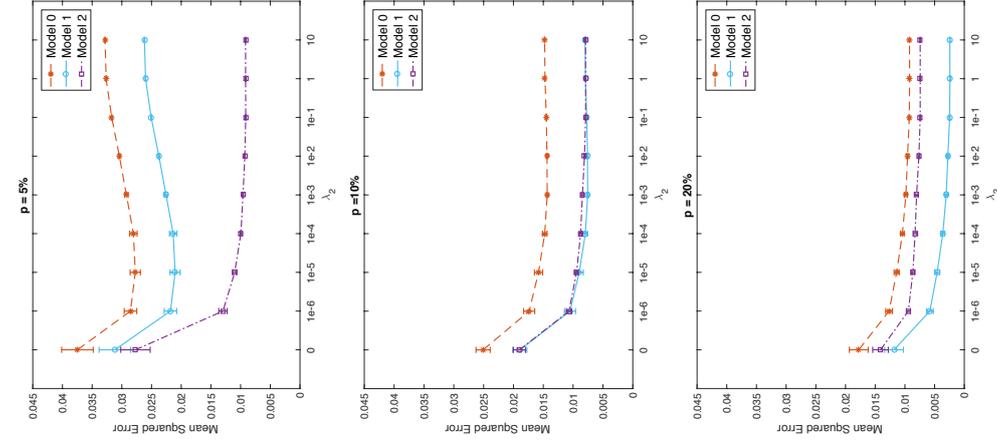


Figure 7: Mean squared errors obtained for the prediction of molecular activities for the model 0 (corresponding to $A = I$), model 1 ($\mu = 1$) and model 2 ($\mu = 0.8$). The results are averaged over ten random training sets and are given for different percentages of labeled data (5%, 10% and 20%).

that Generalized Cross-Validation extends in the case of the closed-form solution of IOKR-ridge, providing an efficient tool for model selection. Perspectives to this work concern the construction of new models by minimizing loss functions with different penalties, for instance, penalties that enforce the parsimony of the model. For these non-smooth penalties, proximal gradient descent methods can be applied such as in Lim et al. (2013). A more general research direction is related to the design of new kernels and appropriate kernel learning algorithms. Finally, although the pre-image problem has received a lot of attention in the literature, there is still room for improvement in order to apply IOKR in other tasks than link prediction or multiple output structured regression.

Acknowledgments

We thank the reviewers for their valuable comments. We are also grateful to Maxime Sangnier for his relevant comments about the paper. We would like to acknowledge support for this project from ANR (grant ANR-009-SYSC-009-02) and University of Evry (PhD grant).

Appendix A. Technical Proofs

In this appendix section, we provide the proofs for some theorems and propositions presented in the paper.

A.1 Proof of Theorem 7

We derive here the solution obtained for the Maximum Margin Regression optimization problem in the semi-supervised setting (Equation 8). We begin by writing the primal formulation of this optimization problem:

$$\begin{aligned} \min_{h \in \mathcal{H}, \xi \in \mathbb{R}^\ell} \quad & \lambda_1 \|h\|_{\mathcal{H}}^2 + 2\lambda_2 \sum_{i,j=1}^{\ell+n} L_{ij} \langle h(x_i), h(x_j) \rangle_{\mathcal{F}_y} + \sum_{i=1}^{\ell} \xi_i \\ \text{s.t.} \quad & \langle \tilde{y}_i, h(x_i) \rangle_{\mathcal{F}_y} \geq 1 - \xi_i, i = 1, \dots, \ell \\ & \xi_i \geq 0, i = 1, \dots, \ell. \end{aligned}$$

We write the corresponding Lagrangian:

$$\begin{aligned} \mathcal{L}_a(h, \xi, \alpha, \eta) = & \lambda_1 \|h\|_{\mathcal{H}}^2 + 2\lambda_2 \sum_{i,j=1}^{\ell+n} L_{ij} \langle h(x_i), h(x_j) \rangle_{\mathcal{F}_y} + \sum_{i=1}^{\ell} \xi_i \\ & - \sum_{i=1}^{\ell} \alpha_i (\langle \tilde{y}_i, h(x_i) \rangle_{\mathcal{F}_y} - 1 + \xi_i) - \sum_{i=1}^{\ell} \eta_i \xi_i. \end{aligned}$$

In the following we note $K_x = K_x(\cdot, \cdot)$ and $K_x^* = K_x(x, \cdot)$. By using the reproducing property the expression of the Lagrangian becomes:

$$\begin{aligned}
 \mathcal{L}_\alpha &= \lambda_1 \|h\|_{\mathcal{H}}^2 + 2\lambda_2 \sum_{i,j=1}^{\ell+n} L_{ij} \langle K_{x_i}^* h, K_{x_j}^* h \rangle_{\mathcal{H}} - \sum_{i=1}^{\ell} \alpha_i \langle \tilde{y}_i, K_{x_i}^* h \rangle_{\mathcal{H}} - 1 + \sum_{i=1}^{\ell} (1 - \alpha_i - \eta_i) \xi_i \\
 &= \langle (\lambda_1 I + 2\lambda_2) \sum_{i,j=1}^{\ell+n} L_{ij} K_{x_j} K_{x_i}^* \rangle h, h \rangle_{\mathcal{H}} - \sum_{i=1}^{\ell} \alpha_i \langle K_{x_i} \tilde{y}_i, h \rangle_{\mathcal{H}} + \sum_{i=1}^{\ell} \alpha_i + \sum_{i=1}^{\ell} (1 - \alpha_i - \eta_i) \xi_i \\
 &= \langle Bh, h \rangle_{\mathcal{H}} - \sum_{i=1}^{\ell} \alpha_i \langle K_{x_i} \tilde{y}_i, h \rangle_{\mathcal{H}} + \sum_{i=1}^{\ell} \alpha_i + \sum_{i=1}^{\ell} (1 - \alpha_i - \eta_i) \xi_i,
 \end{aligned}$$

where $B \in \mathcal{B}(h)$ is the operator defined as: $B = \lambda_1 I + 2\lambda_2 \sum_{i,j=1}^{\ell+n} L_{ij} K_{x_j} K_{x_i}^*$. Due to the symmetry of the Laplacian L , this operator is self-adjoint:

$$B^* = \lambda_1 I + 2\lambda_2 \sum_{i,j=1}^{\ell+n} L_{ij} K_{x_j} K_{x_i}^* = \lambda_1 I + 2\lambda_2 \sum_{i,j=1}^{\ell+n} L_{ij} K_{x_i} K_{x_j}^* = B.$$

Differentiating the Lagrangian with respect to ξ_i and h gives:

$$\begin{aligned}
 \frac{\partial \mathcal{L}_\alpha}{\partial \xi_i} &= 0 \Rightarrow 1 - \alpha_i - \eta_i = 0 \\
 \frac{\partial \mathcal{L}_\alpha}{\partial h} &= 0 \Rightarrow 2Bh - \sum_{i=1}^{\ell} \alpha_i K_{x_i} \tilde{y}_i = 0 \Rightarrow h = \frac{1}{2} B^{-1} \left(\sum_{i=1}^{\ell} \alpha_i K_{x_i} \tilde{y}_i \right).
 \end{aligned}$$

B is invertible as it is a positive definite operator:

$$\begin{aligned}
 \forall h \in \mathcal{H}, \langle h, Bh \rangle_{\mathcal{H}} &= \lambda_1 \|h\|_{\mathcal{H}}^2 + 2\lambda_2 \sum_{i,j=1}^{\ell+n} L_{ij} \langle h, K_{x_j} K_{x_i}^* h \rangle_{\mathcal{H}} \\
 &= \lambda_1 \|h\|_{\mathcal{H}}^2 + 2\lambda_2 \sum_{i,j=1}^{\ell+n} L_{ij} \langle h(x_j), h(x_i) \rangle_{\mathcal{F}_n} \\
 &= \lambda_1 \|h\|_{\mathcal{H}}^2 + \lambda_2 \sum_{i,j=1}^{\ell+n} W_{ij} \|h(x_j) - h(x_i)\|_{\mathcal{F}_n}^2 \\
 &> 0 \text{ for all non-zero function } h.
 \end{aligned}$$

The last inequality is deduced from the assumption that the values of W are non-negative. We formulate a reduced Lagrangian :

$$\begin{aligned}
 \mathcal{L}_\alpha(\alpha) &= \frac{1}{4} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j \langle BB^{-1} K_{x_i} \tilde{y}_i, B^{-1} K_{x_j} \tilde{y}_j \rangle_{\mathcal{H}} - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j \langle K_{x_i} \tilde{y}_i, B^{-1} K_{x_j} \tilde{y}_j \rangle_{\mathcal{H}} + \sum_{i=1}^{\ell} \alpha_i \\
 &= -\frac{1}{4} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j \langle K_{x_i} \tilde{y}_i, B^{-1} K_{x_j} \tilde{y}_j \rangle_{\mathcal{H}} + \sum_{i=1}^{\ell} \alpha_i.
 \end{aligned}$$

The dual formulation of the optimization problem (8) can thus be expressed as:

$$\begin{aligned}
 \min_{\alpha \in \mathbb{R}^{\ell}} \quad & \frac{1}{4} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j \langle K_{x_i} \tilde{y}_i, B^{-1} K_{x_j} \tilde{y}_j \rangle_{\mathcal{H}} - \sum_{i=1}^{\ell} \alpha_i \\
 \text{s.t.} \quad & 0 \leq \alpha_i \leq 1, i = 1, \dots, \ell.
 \end{aligned}$$

This concludes the proof.

A.2 Proof of Proposition 9

We start from the expression of $\text{vec}(C_{\ell+n})$ given in Section 3.3 for the least-squares loss and replace A by its eigendecomposition:

$$\text{vec}(C_{\ell+n}) = (\lambda_1 I_{(\ell+n)d} + M \otimes A)^{-1} \text{vec}(\tilde{Y} \tilde{Y}^T),$$

where $M = (J^T J + 2\lambda_2 L) K X_{\ell+n}$.

We introduce the vec-permutation matrices P_{nm} and P_{mn} defined as:

$$\forall A \in \mathbb{R}^{m \times n}, \text{vec}(A^T) = P_{nm} \text{vec}(A) \text{ and } \text{vec}(A) = P_{mn} \text{vec}(A^T).$$

For any $m \times n$ matrix A and $p \times q$ matrix B ,

$$B \otimes A = P_{pm} (A \otimes B) P_{nq}.$$

Using these properties, we can write:

$$\begin{aligned}
 \text{vec}(C_{\ell+n}^T) &= P_{d(\ell+n)} \text{vec}(C_{\ell+n}) \\
 &= P_{d(\ell+n)} (\lambda_1 I_{(\ell+n)d} + P_{(\ell+n)d} (A \otimes M) P_{d(\ell+n)})^{-1} \text{vec}(\tilde{Y} \tilde{Y}^T) \\
 &= (\lambda_1 I_{(\ell+n)d} + P_{d(\ell+n)} P_{(\ell+n)d} (A \otimes M))^{-1} P_{d(\ell+n)} \text{vec}(\tilde{Y} \tilde{Y}^T) \\
 &= (\lambda_1 I_{(\ell+n)d} + A \otimes M)^{-1} \text{vec}(J^T \tilde{Y}^T) \\
 &= (\lambda_1 I_{(\ell+n)d} + ETE^T \otimes M)^{-1} \text{vec}(J^T \tilde{Y}^T).
 \end{aligned}$$

We multiply each side by $(E^T \otimes I_{\ell+n})$

$$\begin{aligned}
 (E^T \otimes I_{\ell+n}) \text{vec}(C_{\ell+n}^T) &= \\
 (E^T \otimes I_{\ell+n}) (\lambda_1 I_{(\ell+n)d} + (E \otimes I_{\ell+n})(\Gamma \otimes M)(E^T \otimes I_{\ell+n}))^{-1} \text{vec}(J^T \tilde{Y}^T).
 \end{aligned}$$

We use the facts that $\text{vec}(AXB) = (B^T \otimes I) \text{vec}(X)$ and that $E^T E = I_d$ to obtain the following equation:

$$\text{vec}(C_{\ell+n}^T E) = (\lambda_1 I_{(\ell+n)d} + \Gamma \otimes M)^{-1} \text{vec}(J^T \tilde{Y}^T E).$$

The matrix $(\lambda_1 I_{(\ell+n)d} + \Gamma \otimes M)$ being block-diagonal, we have

$$C_{\ell+n}^T \mathbf{e}_i = (\lambda_1 I_{\ell+n} + \gamma_i M)^{-1} J^T \tilde{Y}^T \mathbf{e}_i, \text{ for } i = 1, \dots, \ell + n.$$

Then, we can express the model h as:

$$\begin{aligned} \forall x \in \mathcal{X}, h(x) &= AC_{\ell+n} \mathbf{K}_{x_{\ell+n}}^{\mathbf{w}} = \sum_{j=1}^d \gamma_j \mathbf{e}_j \mathbf{e}_j^T C_{\ell+n} \mathbf{K}_{x_{\ell+n}}^{\mathbf{w}} \\ &= \sum_{j=1}^d \gamma_j \mathbf{e}_j \mathbf{e}_j^T \tilde{Y}_\ell J (\lambda_1 I_{\ell+n} + \gamma_j K_{X_{\ell+n}} (J^T J + 2\lambda_2 L))^{-1} \mathbf{K}_{X_{\ell+n}}^{\mathbf{w}}. \end{aligned}$$

In the supervised setting ($\lambda_2 = 0$), the model h writes as:

$$\forall x \in \mathcal{X}, h(x) = \sum_{j=1}^d \gamma_j \mathbf{e}_j \mathbf{e}_j^T \tilde{Y}_\ell (\lambda_1 I_\ell + \gamma_j K_{X_\ell})^{-1} \mathbf{K}_{X_\ell}^{\mathbf{w}}.$$

This completes the proof.

A.3 Proof of Proposition 10

Let $Z_\ell = \tilde{Y}_\ell \text{diag}(\boldsymbol{\alpha}) J$. We start from the expression of the Lagrangian in the case of a general operator-valued kernel (Equation 10) and replace A by its eigendecomposition:

$$\begin{aligned} \mathcal{L}_\alpha(\boldsymbol{\alpha}) &= -\frac{1}{4} \text{vec}(Z_\ell)^T (\lambda_1 I_{(\ell+n)d} + 2\lambda_2 K_{X_{\ell+n}} L \otimes A)^{-1} (K_{X_{\ell+n}} \otimes A) \text{vec}(Z_\ell) + \boldsymbol{\alpha}^T \mathbf{1} \\ &= -\frac{1}{4} \text{vec}(Z_\ell)^T (\lambda_1 I_{(\ell+n)d} + 2\lambda_2 (I_{\ell+n} \otimes E)(K_{X_{\ell+n}} L \otimes \Gamma)(I_{\ell+n} \otimes E^T))^{-1} \\ &\quad (I_{\ell+n} \otimes E)(K_{X_{\ell+n}} \otimes \Gamma)(I_{\ell+n} \otimes E^T) \text{vec}(Z_\ell) + \boldsymbol{\alpha}^T \mathbf{1}. \\ &= -\frac{1}{4} \text{vec}(E^T Z_\ell)^T (\lambda_1 I_{(\ell+n)d} + 2\lambda_2 K_{X_{\ell+n}} L \otimes \Gamma)^{-1} (K_{X_{\ell+n}} \otimes \Gamma) \text{vec}(E^T Z_\ell) + \boldsymbol{\alpha}^T \mathbf{1}. \end{aligned}$$

Using the vec-permutation matrices, we can show that:

$$\mathcal{L}_\alpha(\boldsymbol{\alpha}) = -\frac{1}{4} \text{vec}(Z_\ell^T E)^T (\lambda_1 I_{(\ell+n)d} + 2\lambda_2 \Gamma \otimes K_{X_{\ell+n}} L)^{-1} (\Gamma \otimes K_{X_{\ell+n}}) \text{vec}(Z_\ell^T E) + \boldsymbol{\alpha}^T \mathbf{1}.$$

As $(\lambda_1 I_{(\ell+n)d} + 2\lambda_2 \Gamma \otimes K_{X_{\ell+n}} L)$ is a block diagonal matrix, we can write:

$$\begin{aligned} \mathcal{L}_\alpha(\boldsymbol{\alpha}) &= -\frac{1}{4} \sum_{i=1}^d \mathbf{e}_i^T Z_\ell (\lambda_1 I_{\ell+n} + 2\lambda_2 \gamma_i K_{X_{\ell+n}} L)^{-1} \gamma_i K_{X_{\ell+n}} Z_\ell^T \mathbf{e}_i + \boldsymbol{\alpha}^T \mathbf{1} \\ &= -\frac{1}{4} \sum_{i=1}^d \gamma_i \text{trace}(\tilde{Y}_\ell \mathbf{e}_i \mathbf{e}_i^T \tilde{Y}_\ell \text{diag}(\boldsymbol{\alpha}) J (\lambda_1 I_{\ell+n} + 2\lambda_2 \gamma_i K_{X_{\ell+n}} L)^{-1} K_{X_{\ell+n}} J^T \text{diag}(\boldsymbol{\alpha})) \\ &\quad + \boldsymbol{\alpha}^T \mathbf{1}. \end{aligned}$$

Using the fact that $\mathbf{y}^T (A \circ B) \mathbf{x} = \text{trace}(\text{diag}(\mathbf{y})^T A \text{diag}(\mathbf{x}) B^T)$, the Lagrangian can be written as:

$$\mathcal{L}_\alpha(\boldsymbol{\alpha}) = -\frac{1}{4} \sum_{i=1}^d \gamma_i \boldsymbol{\alpha}^T (\tilde{Y}_\ell^T \mathbf{e}_i \mathbf{e}_i^T \tilde{Y}_\ell \circ J (\lambda_1 I_{\ell+n} + 2\lambda_2 \gamma_i K_{X_{\ell+n}} L)^{-1} K_{X_{\ell+n}} J^T) \boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathbf{1}.$$

In the supervised setting ($\lambda_2 = 0$), the Lagrangian becomes:

$$\begin{aligned} \mathcal{L}_\alpha(\boldsymbol{\alpha}) &= -\frac{1}{4\lambda_1} \sum_{i=1}^d \gamma_i \boldsymbol{\alpha}^T (\tilde{Y}_\ell^T \mathbf{e}_i \mathbf{e}_i^T \tilde{Y}_\ell \circ K_{X_\ell}) \boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathbf{1} \\ &= -\frac{1}{4\lambda_1} \boldsymbol{\alpha}^T (\tilde{Y}_\ell^T A \tilde{Y}_\ell \circ K_{X_\ell}) \boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathbf{1}, \end{aligned}$$

which concludes the proof.

Appendix B. Dual Optimization Problem for a General Convex Loss Function

In this appendix we derive the dual optimization problem for a general convex loss function in the supervised and semi-supervised settings using the Fenchel duality.

B.1 Supervised Setting

We consider the following optimization problem where the cost function $\mathcal{L} : \mathcal{F}_y \times \mathcal{F}_y \rightarrow \mathbb{R}$ is convex in its first variable:

$$\min_{h \in \mathcal{H}} \sum_{i=1}^{\ell} \mathcal{L}(h(x_i), \tilde{\mathbf{y}}_i) + \lambda \|h\|_{\mathcal{H}}^2.$$

It can be rewritten by introducing the constraint $\mathbf{u}_i = h(x_i)$ and the function $\mathcal{L}_i : \mathcal{F}_y \rightarrow \mathbb{R}$ defined as $\mathcal{L}_i(\mathbf{u}_i) = \mathcal{L}(\mathbf{u}_i, \tilde{\mathbf{y}}_i)$ for $i \in [1, \ell]$:

$$\begin{aligned} \min_{h \in \mathcal{H}, \{\mathbf{u}_i \in \mathcal{F}_y\}_{i=1}^{\ell}} \quad & \sum_{i=1}^{\ell} \mathcal{L}_i(\mathbf{u}_i) + \lambda \|h\|_{\mathcal{H}}^2 \\ \text{s.t.} \quad & \mathbf{u}_i = h(x_i), \quad i = 1, \dots, \ell. \end{aligned}$$

We write the expression of the Lagrangian:

$$\begin{aligned} \mathcal{L}_\alpha(h, \mathbf{u}_i, \boldsymbol{\alpha}_i) &= \sum_{i=1}^{\ell} \mathcal{L}_i(\mathbf{u}_i) + \lambda \|h\|_{\mathcal{H}}^2 + \sum_{i=1}^{\ell} \langle \boldsymbol{\alpha}_i, \mathbf{u}_i - h(x_i) \rangle_{\mathcal{F}_y} \\ &= \sum_{i=1}^{\ell} \mathcal{L}_i(\mathbf{u}_i) + \lambda \|h\|_{\mathcal{H}}^2 + \sum_{i=1}^{\ell} \langle \boldsymbol{\alpha}_i, \mathbf{u}_i \rangle_{\mathcal{F}_y} - \sum_{i=1}^{\ell} \langle \mathcal{K}_x(\cdot, x_i) \boldsymbol{\alpha}_i, h \rangle_{\mathcal{H}}. \end{aligned}$$

The dual function can be written:

$$\begin{aligned} g(\alpha) &= \inf_{h \in \mathcal{H}, \{\mathbf{u}_i \in \mathcal{F}_y\}_{i=1}^{\ell}} \left(\sum_{i=1}^{\ell} \mathcal{L}_i(\mathbf{u}_i) + \lambda \|h\|_{\mathcal{H}}^2 + \sum_{i=1}^{\ell} \langle \alpha_i, \mathbf{u}_i \rangle_{\mathcal{F}_y} - \sum_{i=1}^{\ell} \langle \mathcal{K}_x(\cdot, x_i) \alpha_i, h \rangle_{\mathcal{H}} \right) \\ &= \sum_{i=1}^{\ell} \inf_{\mathbf{u}_i \in \mathcal{F}_y} \left(\mathcal{L}_i(\mathbf{u}_i) + \langle \alpha_i, \mathbf{u}_i \rangle_{\mathcal{F}_y} \right) + \inf_h \left(\lambda \|h\|_{\mathcal{H}}^2 - \sum_{i=1}^{\ell} \langle \mathcal{K}_x(\cdot, x_i) \alpha_i, h \rangle_{\mathcal{H}} \right) \\ &= - \sum_{i=1}^{\ell} \sup_{\mathbf{u}_i \in \mathcal{F}_y} \left(-\mathcal{L}_i(\mathbf{u}_i) + \langle -\alpha_i, \mathbf{u}_i \rangle_{\mathcal{F}_y} \right) + \inf_h \left(\lambda \|h\|_{\mathcal{H}}^2 - \sum_{i=1}^{\ell} \langle \mathcal{K}_x(\cdot, x_i) \alpha_i, h \rangle_{\mathcal{H}} \right) \\ g(\alpha) &= - \sum_{i=1}^{\ell} \mathcal{L}_i^*(-\alpha_i) - \frac{1}{4\lambda} \sum_{i,j=1}^{\ell} \langle \alpha_i, \mathcal{K}_x(x_i, x_j) \alpha_j \rangle_{\mathcal{F}_y}, \end{aligned}$$

where \mathcal{L}_i^* denotes the convex conjugate, also called Fenchel conjugate, of the function \mathcal{L}_i :

$$\mathcal{L}_i^*(\alpha_i) = \sup_{\mathbf{u}_i \in \mathcal{F}_y} \langle \alpha_i, \mathbf{u}_i \rangle_{\mathcal{F}_y} - \mathcal{L}_i(\mathbf{u}_i)$$

and $h = \frac{1}{2\lambda} \sum_{i=1}^{\ell} \mathcal{K}_x(\cdot, x_i) \alpha_i$.

The dual optimization problem for a general convex loss function writes as follows:

$$\max_{\{\alpha_i \in \mathcal{F}_y\}_{i=1}^{\ell}} - \sum_{i=1}^{\ell} \mathcal{L}_i^*(-\alpha_i) - \frac{1}{4\lambda} \sum_{i,j=1}^{\ell} \langle \alpha_i, \mathcal{K}_x(x_i, x_j) \alpha_j \rangle_{\mathcal{F}_y}. \quad (14)$$

In the following, we derive it for the least-squares and the MMR loss functions.

B.1.1 LEAST-SQUARES LOSS

We compute the convex conjugate of the least-squares loss:

$$\mathcal{L}_i^*(-\alpha_i) = \sup_{\mathbf{u}_i \in \mathcal{F}_y} -\langle \alpha_i, \mathbf{u}_i \rangle_{\mathcal{F}_y} - \|\mathbf{u}_i - \tilde{\mathbf{y}}_i\|_{\mathcal{F}_y}^2.$$

By setting the derivative $\frac{\partial \mathcal{L}_i^*}{\partial \mathbf{u}_i}$ to 0 we find that $\mathbf{u}_i = \tilde{\mathbf{y}}_i - \frac{1}{2} \alpha_i$. By substituting we see that:

$$\mathcal{L}_i^*(-\alpha_i) = \frac{1}{4} \|\alpha_i\|_{\mathcal{F}_y}^2 - \langle \alpha_i, \tilde{\mathbf{y}}_i \rangle_{\mathcal{F}_y}.$$

We replace the expression of $\mathcal{L}_i^*(-\alpha_i)$ in the dual problem:

$$\max_{\{\alpha_i \in \mathcal{F}_y\}_{i=1}^{\ell}} - \frac{1}{4} \sum_{i=1}^{\ell} \|\alpha_i\|_{\mathcal{F}_y}^2 + \sum_{i=1}^{\ell} \langle \alpha_i, \tilde{\mathbf{y}}_i \rangle_{\mathcal{F}_y} - \frac{1}{4\lambda} \sum_{i,j=1}^{\ell} \langle \alpha_i, \mathcal{K}_x(x_i, x_j) \alpha_j \rangle_{\mathcal{F}_y}.$$

We derive with respect to α_i , $i = 1, \dots, \ell$ and find that the solution of the dual optimization problem satisfy the following equations:

$$\sum_{j=1}^{\ell} (\mathcal{K}_x(x_i, x_j) + \lambda \delta_{ij}) \alpha_j = 2\lambda \tilde{\mathbf{y}}_i, \quad i = 1, \dots, \ell.$$

B.1.2 MAXIMUM MARGIN REGRESSION

We compute the convex conjugate function of the MMR loss using the Lagrange technique:

$$\begin{aligned} -\mathcal{L}_i^*(-\alpha_i) &= - \sup_{\mathbf{u}_i \in \mathcal{F}_y} \{ -\langle \alpha_i, \mathbf{u}_i \rangle_{\mathcal{F}_y} - \max(0, 1 - \langle \tilde{\mathbf{y}}_i, \mathbf{u}_i \rangle_{\mathcal{F}_y}) \} \\ &= \inf_{\mathbf{u}_i \in \mathcal{F}_y, \xi_i \in \mathbb{R}} \{ \langle \alpha_i, \mathbf{u}_i \rangle_{\mathcal{F}_y} + \xi_i \} \\ &= \inf_{\xi_i \geq 0, \xi_i \geq 1 - \langle \tilde{\mathbf{y}}_i, \mathbf{u}_i \rangle_{\mathcal{F}_y}} \{ \langle \alpha_i, \mathbf{u}_i \rangle_{\mathcal{F}_y} + \xi_i \} \\ &= \sup_{\beta_i, \eta_i \geq 0} \left\{ \inf_{\mathbf{u}_i \in \mathcal{F}_y, \xi_i \in \mathbb{R}} \{ \langle \alpha_i, \mathbf{u}_i \rangle_{\mathcal{F}_y} + \xi_i + \beta_i (1 - \langle \tilde{\mathbf{y}}_i, \mathbf{u}_i \rangle_{\mathcal{F}_y} - \xi_i) - \eta_i \xi_i \} \right\} \\ &= \sup_{\beta_i, \eta_i \geq 0} \left\{ \inf_{\mathbf{u}_i \in \mathcal{F}_y} \{ \langle \alpha_i, \mathbf{u}_i \rangle_{\mathcal{F}_y} - \beta_i \langle \tilde{\mathbf{y}}_i, \mathbf{u}_i \rangle_{\mathcal{F}_y} \} + \inf_{\xi_i \in \mathbb{R}} \{ \xi_i - \beta_i \xi_i - \eta_i \xi_i \} + \beta_i \right\} \\ &= \sup_{\substack{0 \leq \beta_i \leq 1 \\ \alpha_i = \beta_i \tilde{\mathbf{y}}_i}} \beta_i. \end{aligned}$$

This means that $-\mathcal{L}_i^*(-\alpha_i) = \beta_i$ at the condition that $\alpha_i = \beta_i \tilde{\mathbf{y}}_i$ and $0 \leq \beta_i \leq 1$. Otherwise it is unbounded. We replace in the dual problem in Equation (14):

$$\max_{\beta_i \in \mathbb{R}^{\ell}} \sum_{i=1}^{\ell} \beta_i - \frac{1}{4\lambda} \sum_{i,j=1}^{\ell} \beta_i \beta_j \langle \tilde{\mathbf{y}}_i, \mathcal{K}_x(x_i, x_j) \tilde{\mathbf{y}}_j \rangle_{\mathcal{F}_y} \quad \text{s.t.} \quad 0 \leq \beta_i \leq 1, \quad i = 1, \dots, \ell.$$

B.2 Semi-supervised Setting

In the semi-supervised setting, the optimization problem can be written as:

$$\min_{h \in \mathcal{H}, \{\mathbf{u}_i \in \mathcal{F}_y\}_{i=1}^{\ell}} \sum_{i=1}^{\ell} \mathcal{L}(\mathbf{u}_i, \tilde{\mathbf{y}}_i) + \lambda_1 \|h\|_{\mathcal{H}}^2 + 2\lambda_2 \sum_{i,j=1}^{\ell+n} L_{ij} \langle h(x_i), h(x_j) \rangle_{\mathcal{H}} \quad \text{s.t.} \quad \mathbf{u}_i = h(x_i), \quad i = 1, \dots, \ell.$$

We write the expression of the dual function:

$$\begin{aligned} g(\alpha) &= \inf_{h \in \mathcal{H}} \sum_{i=1}^{\ell} \mathcal{L}(\mathbf{u}_i, \tilde{\mathbf{y}}_i) + \lambda_1 \|h\|_{\mathcal{H}}^2 + 2\lambda_2 \sum_{i,j=1}^{\ell+n} L_{ij} \langle h(x_i), h(x_j) \rangle_{\mathcal{H}} + \sum_{i=1}^{\ell} \langle \alpha_i, \mathbf{u}_i - h(x_i) \rangle_{\mathcal{F}_y} \\ &= - \sum_{i=1}^{\ell} \mathcal{L}_i^*(-\alpha_i) + \inf_{h \in \mathcal{H}} \left(\langle B \mathbf{1}_\ell, h \rangle_{\mathcal{H}} - \sum_{i=1}^{\ell} \langle \mathcal{K}_x \alpha_i, h \rangle_{\mathcal{H}} \right), \end{aligned}$$

where $B \in \mathcal{B}(h)$ is the operator defined as: $B = \lambda_1 I + 2\lambda_2 \sum_{i,j=1}^{\ell+n} L_{ij} \mathcal{K}_x(\cdot, x_j) \mathcal{K}_x(x_i, \cdot)$.

By setting the derivative of the second term with respect to h to zero we find that: $h = \frac{1}{2} B^{-1} \left(\sum_{i=1}^{\ell} \mathcal{K}_x(\cdot, x_i) \alpha_i \right)$. The proof that B can be inverted was already given in Appendix A.1.

We replace in the dual function and obtain the following dual optimization problem:

$$\max_{\{\alpha_i \in \mathcal{F}_i\}_{i=1}^{\ell}} - \sum_{i=1}^{\ell} \mathcal{L}_i^*(-\alpha_i) - \frac{1}{4} \sum_{i,j=1}^{\ell} \langle K_{x_i} \alpha_i, B^{-1} K_{x_j} \alpha_j \rangle_{\mathcal{H}}.$$

Appendix C. Additional Results on Synthetic Networks

This appendix contains additional results on synthetic networks.

C.1 Influence of the Level of Inertia

We experimented how IOKR behaves with perfect to noisy input features on the synthetic networks. We modified the quality of the input representation by varying the relative inertia captured by the first components. We chose four different levels of inertia: 75%, 85%, 95% and 100%. The results obtained with IOKR-ridge and IOKR-margin are shown in Table 8.

For both methods we observe small differences in term of AUC-ROC when the inertia varies between 75% and 100%. On the other hand, there is more variation in the AUC-PR results, especially for a low graph density. The difference between the AUC-PRs for 75% and 100% of inertia increases when the percentage of labeled nodes is increased. Overall IOKR is robust to the noise level of the input data in all the cases for the AUC-ROC, and in the networks of density 0.01 and 0.02 for the AUC-PR.

C.2 Mixture of Erdős-Renyi Random Graphs

We generated synthetic networks using mixtures of Erdős-Renyi random graphs. The 700 nodes of the graphs were divided equally in three classes. We considered that the connection probability between a node belonging to the class i and a node in the class j can take two values:

$$\forall i, j \in \{1, \dots, 3\}, p_{i,j} = \begin{cases} p_{intra} & \text{if } i = j, \\ p_{inter} & \text{if } i \neq j. \end{cases}$$

We evaluated the performances of IOKR-ridge and IOKR-margin on these random networks for $p_{intra} \in \{0.02, 0.03\}$ and $p_{inter} \in \{5 * 10^{-4}, 10^{-3}\}$. The input vectors were derived from the diffusion kernel applied on the network as described in Section 6.1.2. These results are reported in Table 9. For IOKR-margin, we observe that the AUC values stay relatively similar for the different networks and also for the different percentage of labeled data. On the opposite, IOKR-ridge presents better performances when the inter-class connection probability is higher. As in Section 6.1.2, in which we noted that denser networks are more difficult to predict, we observe here that the AUC values decrease when the intra-class connection probability increases.

In Figure 8, we illustrate the fact that IOKR is able to recover the clusters present in a synthetic network ($p_{intra} = 0.02$, $p_{inter}=5e-4$). The true network is shown on the left and the network predicted with IOKR-ridge is shown on the right. The predicted network was obtained by thresholding the values in the predicted output kernel. The value of the threshold was selected with the other parameters on the training set such that it maximizes the F1-score value.

a) IOKR-ridge:

pdens	var %	nc	AUC-ROC			AUC-PR		
			p=5%	p=10%	p=20%	p=5%	p=10%	p=20%
0.007	75	69	92.3 ± 0.4	95.9 ± 0.2	97.7 ± 0.2	12.9 ± 0.9	21.9 ± 0.8	30.0 ± 1.8
	85	100	92.1 ± 0.9	95.6 ± 0.3	97.9 ± 0.2	14.2 ± 0.9	22.7 ± 1.0	35.3 ± 1.5
	95	159	92.2 ± 1.2	95.6 ± 0.3	97.8 ± 0.2	15.4 ± 1.5	24.7 ± 1.7	36.1 ± 1.5
0.01	75	104	90.3 ± 1.3	94.8 ± 0.9	97.1 ± 0.9	14.1 ± 1.4	22.8 ± 2.3	32.5 ± 3.6
	85	145	90.5 ± 1.3	94.9 ± 0.5	97.6 ± 0.1	15.0 ± 1.5	23.6 ± 1.1	35.1 ± 0.9
	95	227	90.6 ± 1.0	95.4 ± 0.4	98.0 ± 0.3	15.7 ± 1.0	25.6 ± 1.2	39.2 ± 1.6
0.02	75	201	82.6 ± 1.4	91.2 ± 0.8	95.4 ± 0.4	15.3 ± 0.8	26.7 ± 1.9	38.2 ± 2.0
	85	274	83.0 ± 1.7	90.6 ± 0.6	94.6 ± 0.5	16.1 ± 0.8	26.3 ± 1.3	36.3 ± 1.8
	95	411	82.8 ± 1.8	91.2 ± 0.7	95.1 ± 0.4	16.0 ± 0.8	28.0 ± 1.5	40.8 ± 1.7
100	700	82.8 ± 1.8	91.1 ± 0.7	95.0 ± 0.4	16.0 ± 0.8	27.9 ± 1.5	40.5 ± 1.7	

b) IOKR-margin:

pdens	var %	nc	AUC-ROC			AUC-PR		
			p=5%	p=10%	p=20%	p=5%	p=10%	p=20%
0.007	75	69	91.3 ± 0.7	93.4 ± 0.7	94.7 ± 0.6	10.5 ± 1.1	12.5 ± 1.2	14.4 ± 1.5
	85	100	91.0 ± 0.8	93.5 ± 0.8	95.2 ± 0.7	12.0 ± 0.8	15.2 ± 2.6	19.6 ± 4.4
	95	159	90.5 ± 0.9	93.1 ± 0.5	95.3 ± 0.3	12.5 ± 0.9	18.9 ± 2.5	26.5 ± 1.0
0.01	75	104	87.9 ± 0.9	91.0 ± 0.8	93.5 ± 0.4	12.1 ± 1.2	16.5 ± 1.0	22.8 ± 1.0
	85	145	87.7 ± 1.0	91.1 ± 1.2	92.9 ± 0.5	12.5 ± 1.0	16.9 ± 0.9	23.3 ± 1.1
	95	227	87.3 ± 1.6	91.3 ± 0.8	94.1 ± 0.5	12.5 ± 1.6	17.9 ± 1.3	24.7 ± 1.0
0.02	75	201	87.2 ± 1.6	91.2 ± 0.8	94.1 ± 0.5	12.4 ± 1.6	17.9 ± 1.3	24.9 ± 1.1
	85	274	78.4 ± 2.1	83.2 ± 1.2	88.4 ± 0.7	12.0 ± 1.0	17.3 ± 0.7	24.0 ± 0.9
	95	411	77.6 ± 2.1	81.9 ± 1.6	87.5 ± 0.8	12.5 ± 1.0	17.3 ± 1.0	24.3 ± 0.8
100	700	77.3 ± 2.3	82.0 ± 1.6	87.1 ± 0.9	12.8 ± 1.0	17.8 ± 1.2	24.4 ± 0.7	

Table 8: Averaged AUCs obtained with IOKR for the reconstruction of three synthetic networks. The first column indicates the link probability between two nodes, var corresponds to the percentage of variance, or inertia, used to truncate the principal components and nc indicates the corresponding number of principal components.

a) IOKR-ridge:

P_{intra}	P_{inter}	var	AUC-ROC			AUC-PR		
			p=5%	p=10%	p=20%	p=5%	p=10%	p=20%
0.02	5e-4	75	85.0 ± 0.7	86.8 ± 0.9	93.1 ± 2.2	2.8 ± 0.4	3.5 ± 0.5	10.3 ± 2.7
		85	85.7 ± 0.7	90.4 ± 3.5	95.5 ± 0.3	3.0 ± 0.4	7.3 ± 3.8	15.9 ± 1.2
		95	86.6 ± 1.8	93.2 ± 2.8	96.6 ± 0.3	3.8 ± 1.9	12.7 ± 4.7	21.2 ± 1.8
1e-3	100	75	87.0 ± 2.3	93.3 ± 2.8	96.6 ± 0.3	4.5 ± 2.7	13.1 ± 4.8	22.1 ± 1.9
		85	87.1 ± 2.2	93.4 ± 2.7	96.9 ± 0.2	5.3 ± 2.5	15.1 ± 5.2	23.9 ± 1.3
		95	88.4 ± 2.1	93.2 ± 2.6	97.2 ± 0.2	7.7 ± 3.7	16.2 ± 5.7	27.1 ± 1.4
0.03	5e-4	75	88.8 ± 1.7	93.1 ± 2.6	97.4 ± 0.5	8.9 ± 3.7	17.0 ± 5.8	31.3 ± 4.0
		85	89.2 ± 1.6	93.1 ± 2.5	97.9 ± 0.4	9.8 ± 3.6	17.1 ± 5.8	34.5 ± 3.1
		100	83.5 ± 0.1	83.6 ± 0.1	84.5 ± 1.2	3.2 ± 0.0	3.2 ± 0.0	3.7 ± 0.5
1e-3	100	75	83.7 ± 0.1	83.8 ± 0.1	86.4 ± 1.4	3.3 ± 0.0	3.3 ± 0.0	4.6 ± 0.7
		85	84.2 ± 1.3	84.5 ± 1.4	88.8 ± 1.0	3.6 ± 0.6	3.8 ± 0.9	6.1 ± 1.0
		95	84.2 ± 1.3	84.7 ± 1.7	89.7 ± 1.6	3.6 ± 0.6	3.9 ± 1.0	7.2 ± 1.9
1e-3	100	75	86.2 ± 1.9	91.1 ± 1.7	93.7 ± 0.3	5.9 ± 2.2	12.1 ± 2.4	16.5 ± 1.3
		85	86.3 ± 1.8	91.8 ± 0.6	94.3 ± 0.3	6.3 ± 2.3	14.3 ± 1.3	19.4 ± 1.6
		95	86.8 ± 2.2	91.7 ± 0.6	94.8 ± 0.7	6.9 ± 3.1	14.8 ± 1.4	22.3 ± 3.1
1e-3	100	75	86.8 ± 2.2	92.1 ± 1.1	95.2 ± 1.0	7.0 ± 3.1	15.7 ± 2.5	23.9 ± 4.2

b) IOKR-margin:

P_{intra}	P_{inter}	inertia	AUC-ROC			AUC-PR		
			p=5%	p=10%	p=20%	p=5%	p=10%	p=20%
0.02	5e-4	0.75	84.5 ± 0.6	84.6 ± 0.4	84.5 ± 0.3	2.6 ± 0.1	2.5 ± 0.1	2.5 ± 0.1
		0.85	84.8 ± 0.7	84.9 ± 0.5	84.9 ± 0.4	2.7 ± 0.1	2.6 ± 0.1	2.6 ± 0.2
		0.95	85.1 ± 0.9	85.2 ± 0.5	85.2 ± 0.5	2.8 ± 0.1	2.8 ± 0.1	2.7 ± 0.2
1e-3	1	0.75	85.2 ± 0.9	85.3 ± 0.5	85.3 ± 0.5	2.9 ± 0.2	2.8 ± 0.1	2.8 ± 0.2
		0.85	84.8 ± 0.6	85.0 ± 0.3	84.7 ± 0.6	3.4 ± 0.4	3.3 ± 0.3	3.0 ± 0.2
		0.95	85.1 ± 0.6	85.3 ± 0.3	85.3 ± 0.4	3.6 ± 0.4	3.7 ± 0.3	3.4 ± 0.3
1e-3	1	0.75	85.2 ± 0.7	85.5 ± 0.4	85.6 ± 0.2	3.8 ± 0.4	3.9 ± 0.3	3.9 ± 0.1
		0.85	85.5 ± 0.4	85.6 ± 0.4	85.7 ± 0.2	3.9 ± 0.4	3.9 ± 0.3	3.9 ± 0.1
		0.95	83.4 ± 0.2	83.5 ± 0.2	83.5 ± 0.2	3.2 ± 0.0	3.2 ± 0.0	3.2 ± 0.0
0.03	5e-4	0.75	83.6 ± 0.2	83.7 ± 0.2	83.6 ± 0.2	3.3 ± 0.0	3.3 ± 0.0	3.3 ± 0.0
		0.85	83.6 ± 0.2	83.7 ± 0.2	83.7 ± 0.2	3.3 ± 0.0	3.4 ± 0.0	3.3 ± 0.0
		0.95	83.6 ± 0.2	83.8 ± 0.2	83.7 ± 0.2	3.3 ± 0.0	3.4 ± 0.0	3.3 ± 0.0
1e-3	1	0.75	84.0 ± 0.6	83.9 ± 0.4	84.0 ± 0.1	3.8 ± 0.2	3.7 ± 0.2	3.7 ± 0.1
		0.85	84.2 ± 0.6	84.1 ± 0.4	84.1 ± 0.1	4.0 ± 0.3	3.9 ± 0.2	3.9 ± 0.1
		0.95	84.3 ± 0.7	84.2 ± 0.4	84.2 ± 0.1	4.1 ± 0.3	4.0 ± 0.2	4.0 ± 0.1
1e-3	1	0.75	84.3 ± 0.7	84.2 ± 0.4	84.2 ± 0.1	4.2 ± 0.3	4.0 ± 0.2	4.0 ± 0.1

Table 9: Averaged AUCs obtained with IOKR on different mixtures of Erdős-Renyi random graphs. P_{intra} and P_{inter} denote respectively the intra- and inter-class connection probabilities. The third column indicates the percentage of variance used to define the input vectors from the principal components.

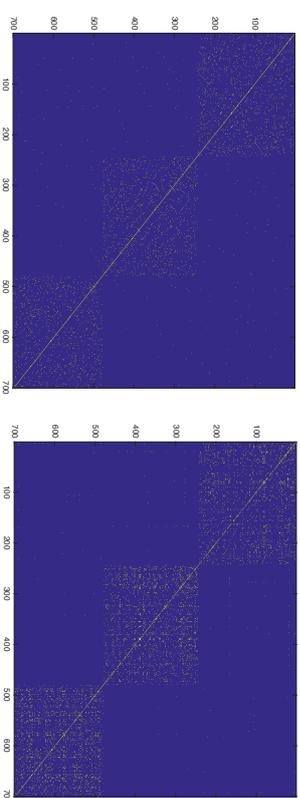
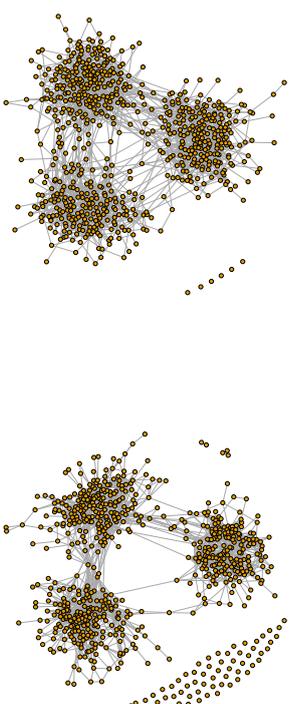


Figure 8: Prediction of a mixture of Erdős-Renyi random graphs with IOKR. The true network is shown on the left and the network predicted with IOKR-ridge on the right. The respective adjacency matrices are displayed under the two networks.

References

D. M. Allen. The relationship between variable selection and data augmentation and a method for prediction. *Technometrics*, 16(1):125–127, 1974.

M. A. Álvarez, L. Rosasco, and N. D. Lawrence. Kernels for vector-valued functions: a review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012.

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, pages 337–404, 1950.

- M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.
- L. Baldassarre, L. Rosasco, A. Barla, and A. Verri. Multi-output learning via spectral filtering. *Machine Learning*, 87(3):259–301, 2012.
- M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.
- A. Ben-Hur and W. S. Noble. Kernel methods for predicting protein–protein interactions. *Bioinformatics*, 21(1):38–46, 2005. ISSN 1367-4803.
- K. Bleakley, G. Biau, and J.-P. Vert. Supervised reconstruction of biological networks with local models. *Bioinformatics*, 23(13):i57–i65, 2007.
- C. Brouard. *Inférence de réseaux d'interaction protéine-protéine par apprentissage statistique*. PhD thesis, University of Evry, France, feb 2013.
- C. Brouard, F. d'Alché-Buc, and M. Szafranski. Semi-supervised penalized output kernel regression for link prediction. In *International Conference on Machine Learning (ICML)*, pages 593–600, 2011.
- J. Burbea and P. Masani. Banach and Hilbert spaces of vector-valued functions. *Pitman Research Notes in Mathematics*, 90, 1984.
- A. Caponnetto, C. A. Micchelli, M. , and Y. Ying. Universal multitask kernels. *Journal of Machine Learning Research*, 9:1615–1646, 2008.
- C. Carmeli, E. De Vito, A. Toigo, and V. Umaita. Vector valued reproducing kernel Hilbert spaces and universality. *Analysis and Applications*, 8:19–61, 2010.
- R. Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.
- C. Cortes, M. Mohri, and J. Weston. A general regression technique for learning transductions. In *International Conference on Machine Learning (ICML)*, pages 153–160, 2005.
- C. M. Deane, L. Salwinski, I. Xenarios, and D. Eisenberg. Protein interactions: two methods for assessment of the reliability of high throughput observations. *Molecular & Cellular Proteomics*, 1(5):349–356, 2002.
- F. Dimuzzo, C.S. Ong, P. Gehler, and G. Pillonetto. Learning output kernels with block coordinate descent. In *International Conference on Machine Learning (ICML)*, pages 49–56, 2011.
- T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.
- BROUARD, SZAFRANSKI AND D'ALCHÉ-BUC
- P. Geurts, L. Wehenkel, and F. d'Alché-Buc. Kernelizing the output of tree-based methods. In *International Conference on Machine Learning (ICML)*, pages 345–352, 2006.
- P. Geurts, N. Touleimat, M. Dutreix, and F. d'Alché-Buc. Inferring biological networks with output kernel trees. *BMC Bioinformatics (PMSB06 special issue)*, 8(Suppl 2):S4, 2007a.
- P. Geurts, L. Wehenkel, and F. d'Alché-Buc. Gradient boosting for kernelized output spaces. In *International Conference on Machine Learning (ICML)*, volume 227, pages 289–296, 2007b.
- A. Globerson, G. Chechik, F. Pereira, and N. Tishby. Euclidean embedding of co-occurrence data. *Journal of Machine Learning Research*, 8:2265–2295, 2007.
- G. H. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.
- P. Honeine and C. Richard. Preimage problem in kernel-based machine learning. *IEEE Signal Processing Magazine*, 28(2):77–88, 2011.
- H. Kadri, E. Duflos, P. Preux, S. Canu, and M. Davy. Nonlinear functional regression: a functional RKHS approach. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 374–380, 2010.
- H. Kadri, M. Ghavamzadeh, and P. Preux. A generalized kernel approach to structured output learning. In *International Conference on Machine Learning (ICML)*, pages 471–479, 2013.
- H. Kashima, T. Kato, Y. Yamashita, M. Sugiyama, and K. Tsuda. Link propagation: a fast semi-supervised learning algorithm for link prediction. In *SIAM International Conference on Data Mining*, pages 1099–1110, 2009.
- T. Kato, K. Tsuda, and K. Asai. Selective integration of multiple biological data for supervised network inference. *Bioinformatics*, 21(10):2488–2495, 2005.
- R. I. Kondor and J. D. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *International Conference on Machine Learning (ICML)*, pages 315–322, 2002.
- J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning (ICML)*, pages 282–289, 2001.
- J. Lafferty, X. Zhu, and Y. Liu. Kernel conditional random fields: representation and clique selection. In *International Conference on Machine Learning (ICML)*, pages 504–511, 2004.
- C. H. Lampert and M. B. Blaschko. Structured prediction by joint kernel support estimation. *Machine Learning*, 77(2-3):249–269, 2009.

- N. Lim, Y. Senbabaoğlu, G. Michailidis, and F. d'Alché-Buc. Okvar-boost: a novel boosting algorithm to infer nonlinear dynamics and interactions in gene regulatory networks. *Bioinformatics*, 29(11):1416–1423, 2013.
- C. A. Micchelli and M. A. Pottli. On learning vector-valued functions. *Neural Computation*, 17:177–204, 2005.
- H. Q. Minh and V. Sindhwani. Vector-valued manifold regularization. In *International Conference on Machine Learning (ICML)*, pages 57–64, 2011.
- N.D. Pearce and M.P. Ward. Penalized splines and reproducing kernel methods. *The American Statistician*, 60(3):233–240, august 2006.
- G. Pedrick. Theory of reproducing kernels for Hilbert spaces of vector-valued functions. Technical report, University of Kansas, Department of Mathematics, 1957.
- L. Ralavivola, S. J. Swamidass, H. Saigo, and P. Baldi. Graph kernels for chemical information. *Neural Network*, 18(8):1093–1110, 2005.
- R. M. Rifkin and R. A. Lippert. Notes on regularized least-squares. Technical report, MIT, Computer Science and Artificial Intelligence Laboratory, 2007.
- L. Salwinski, C. S. Miller, A. J. Smith, F.K. Pettit, J. U. Bowie, and D. Eisenberg. The database of interacting proteins: 2004 update. *Nucleic Acids Research*, 32 (Database Issue):D449–D451, 2004.
- E. Senkne and A. Tempelman. Hilbert spaces of operator-valued functions. *Lithuanian Mathematical Journal*, 13(4):665–670, 1973.
- D. Sheldon. Graphical multi-task learning. Technical report, Cornell University, 2008. URL <http://web.engr.oregonstate.edu/~sheldon/>.
- H. Su, M. Heinonen, and J. Rousu. Structured output prediction of anti-cancer drug activity. In *International Conference on Pattern Recognition In Bioinformatics (PRIB)*, pages 38–49. Springer-Verlag, 2010.
- S. Szedmak, J. Shawe-Taylor, and E. Parado-Hernandez. Learning via linear operators: Maximum margin regression. Technical report, University of Southampton, UK, 2005.
- B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In *Advances in Neural Information Processing Systems (NIPS)*, volume 16, page 25, 2004.
- I. Tsochanaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *International Conference on Machine Learning (ICML)*, page 104, 2004.
- I. Tsochanaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.
- K. Tsuda, S. Akaho, and K. Asai. The em algorithm for kernel matrix completion with auxiliary data. *Journal of Machine Learning Research*, 4:67–81, 2003.
- J.-P. Vert and Y. Yamaniishi. Supervised graph inference. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1433–1440, 2005.
- G. Walther. *Spline Model for Observational Data*. Philadelphia, Society for Industrial and Applied Mathematics, 1990.
- J. Weston, O. Chapelle, A. Elisseeff, B. Schölkopf, and V. Vapnik. Kernel dependency estimation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 873–880, 2003.
- Y. Yamaniishi and J.-P. Vert. Kernel matrix regression. In *International Conference on Applied Stochastic Models and Data Analysis (ASMDA)*, 2007.
- Y. Yamaniishi, J.-P. Vert, and M. Kanehisa. Protein network inference from multiple genomic data: a supervised approach. *Bioinformatics*, 20:i363–i370, 2004.

A Note on the Sample Complexity of the Er-SpUD Algorithm by Spielman, Wang and Wright for Exact Recovery of Sparsely Used Dictionaries

Radosław Adamczak

Institute of Mathematics

University of Warsaw

Banacha 2

02-097 Warszawa

Poland

R.ADAMCZAK@MIMUW.EDU.PL

Editor: Sara van de Geer

Abstract

We consider the problem of recovering an invertible $n \times n$ matrix A and a sparse $n \times p$ random matrix X based on the observation of $Y = AX$ (up to a scaling and permutation of columns of A and rows of X). Using only elementary tools from the theory of empirical processes we show that a version of the Er-SpUD algorithm by Spielman, Wang and Wright with high probability recovers A and X exactly, provided that $p \geq Cn \log n$, which is optimal up to the constant C .

Keywords: sparse dictionaries, Er-SpUD algorithm, ℓ_1 minimization, exact recovery, sample complexity

1. Introduction

Recovery of sparsely-used dictionaries has recently attracted considerable attention in connection to applications in machine learning, signal processing or computational neuroscience. In particular, two important fields of applications are *dictionary learning* (Olshausen and Field, 1996; Kreutz-Delgado et al., 2003; Bruckstein et al., 2009; Rubinfeld et al., 2010; Yang et al., 2010) and *blind source separation* (Zibulevsky and Pearlmutter, 2001; Georgiev et al., 2005). We do not discuss these applications and refer the Reader to the aforesaid articles for details.

Among many approaches to this problem a particularly successful one has been presented by Spielman, Wang, and Wright (2012a,b), who considered the noiseless-invertible case:

The main problem:

Consider an invertible $n \times n$ matrix A and a random $n \times p$ sparse matrix X . Denote $Y = AX$. The objective is to reconstruct A and X (up to scaling and permutation of columns of A and rows of X) based on the observable data Y .

Spielman, Wang, and Wright (2012a,b) provide an algorithm which with high probability successfully recovers the matrices A and X up to rescaling and permutation of the columns

of A and rows of X , provided that X is a sparse random matrix satisfying the following probabilistic assumptions.

Probabilistic model specification

$$X = [X_{ij}]_{1 \leq i \leq n, 1 \leq j \leq p},$$

where

$$X_{ij} = \chi_{ij} R_{ij}$$

and

- χ_{ij}, R_{ij} are independent random variables,
- χ_{ij} are Bernoulli distributed: $\mathbb{P}(\chi_{ij} = 1) = 1 - \mathbb{P}(\chi_{ij} = 0) = \theta$,
- R_{ij} are i.i.d., with mean zero and satisfy

$$\mu := \mathbb{E}|R_{ij}| \geq 1/10,$$

$$\forall t > 0 \quad \mathbb{P}(|R_{ij}| \geq t) \leq 2e^{-t^2/2}.$$

Following Spielman, Wang, and Wright (2012a) we will say that matrices satisfying the above assumptions follow the Bernoulli-Subgaussian model with parameter θ .

We remark that the constant $1/10$ above is of no importance and has been chosen following Spielman, Wang, and Wright (2012a) and Luh and Vu (2016).

The approach of Spielman, Wang and Wright consists of two steps. At the first step (given by the Er-SpUD algorithm we describe below) one gathers $p/2$ candidates for the rows of X . The second, greedy step (Greedy algorithm, also described below) selects from the candidates the set of n sparse vectors, which form a matrix of rank n .

The algorithms work as follows:

ER-SpUD(DC): Exact Recovery of Sparsely-Used Dictionaries using the sum of two columns of Y as constraint vectors.

1. Randomly pair the columns of Y into $p/2$ groups $g_j = \{Y_{e_{j_1}}, Y_{e_{j_2}}\}$.
2. For $j = 1, \dots, p/2$
 Let $r_j = Y_{e_{j_1}} + Y_{e_{j_2}}$, where $g_j = \{Y_{e_{j_1}}, Y_{e_{j_2}}\}$.
 Solve $\min_w \|w^T Y\|_1$ subject to $r_j^T w = 1$, and set $s_j = w^T Y$.

Above we use the convention that if $r_j = 0$ (which happens with nonzero probability), and as a consequence the minimization problem has no solution, then we skip the corresponding step of the algorithm.

The second stage, described below, is run on the set S of vectors s_i returned at the first stage (for notational simplicity we relabel them if $r_j = 0$ for some j). We use the standard notation that $\|x\|_0$ denotes the number of nonzero coordinates of a vector x .

Greedy: A Greedy Algorithm to Reconstruct X and A .

1. **REQUIRE:** $S = \{s_1, \dots, s_T\} \subseteq \mathbb{R}^p$.
2. **For** $i = 1, \dots, n$
REPEAT
 $l \leftarrow \operatorname{argmin}_{s_i \in S} \|s_i\|_0$, **breaking ties arbitrarily**
 $x_i = s_l$, $S = S \setminus \{s_l\}$
UNTIL $\operatorname{rank}([x_1, \dots, x_i]) = i$
3. **Set** $X = [x_1, \dots, x_n]^T$ and $A = YY^T(XY^T)^{-1}$.

Spielman, Wang, and Wright (2012a) have proved that there exist positive constants C, α , such that if

$$\frac{2}{n} \leq \theta \leq \frac{\alpha}{\sqrt{n}}$$

and $p \geq Cn^2 \log^2 n$, then the ER-SpUD algorithm successfully recovers the matrices A, X with probability at least $1 - \frac{1}{Cnp}$. Note that the equation $Y = AX'$ still holds if we set $A' = A\Pi A$ and $X' = A^{-1}\Pi^T X$ for some permutation matrix Π and a nonsingular diagonal matrix A . Therefore, by recovery we mean that nonzero multiples of all the rows of X are among the set $\{s_1, \dots, s_{n/2}\}$ produced by the ER-SpUD(DC) algorithm. In (Spielman, Wang, and Wright, 2012a) it is also proved that if $\mathbb{P}(R_{ij} = 0) = 0$, then for $p > Cn \log n$, with probability $1 - C'n \exp(-c\theta p)$ for any matrices A', X' such that $Y = A'X'$ and $\max_i \|e_i^T X'\|_0 \leq \max_i \|e_i^T X\|_0$ there exists a permutation matrix Π and a nonsingular diagonal matrix A such that $A' = A\Pi A$, $X' = A^{-1}\Pi^T X$. In fact, Spielman, Wang, and Wright (2012a) prove that with the above probability any row of X is nonzero and has at most $(10/9)\theta p$ nonzero entries, whereas any linear combination of two or more rows of X has at least $(11/9)\theta p$ nonzero entries.

In particular it follows that the Greedy algorithm will extract from the set $\{s_1, \dots, s_T\}$ multiples of all n rows of X (note that all s_j 's are in the row space of Y and thus also in the row space of X). Since X is with high probability of rank n , one easily shows that one can recover A by the formula used in the 3rd step of the algorithm. We remark that Luh and Vu (2016) obtained the same results concerning sparsity of linear combinations of rows of X without the assumptions about the symmetry of the variables R_{ij} .

Note also that for θ of the order n^{-1} , $p = Cn \log n$ is necessary for uniqueness of the solution in the sense described above, otherwise with significant probability some of the rows of X may be zero, which means that some columns of A do not influence the matrix Y .

In (Spielman, Wang, and Wright, 2012a) it is also proved that if $p > Cn \log n$, $\theta > C\sqrt{\frac{\log n}{n}}$, then with high probability the ER-SpUD algorithm does not recover any of the rows of X .

Spielman, Wang and Wright have conjectured that their algorithm works with high probability provided that $p > Cn \log n$ (which, as mentioned above is required for well-posedness of the problem).

In this note we will consider a modified version of the algorithm with a slightly different first stage. Namely, instead of using only $p/2$ pairs of columns of Y , we will use all $\binom{p}{2}$ pairs. For fixed p it clearly increases the time complexity of the algorithm (which however remains polynomial), but the advantage of this modification is the possibility of proving that it requires only $p = Cn \log n$ to recover X and A with high probability, which as explained above is optimal. More specifically, we will consider the following algorithm.

Modified ER-SpUD(DC): Exact Recovery of Sparsely-Used Dictionaries using the sum of two columns of Y as constraint vectors.

- For** $i = 1, \dots, p - 1$
For $j = i + 1, \dots, p$
 Let $r_{ij} = Y e_j + Y e_i$
 Solve $\min_w \|w^T Y\|_1$ subject to $r_{ij}^T w = 1$, and set $s_{ij} = w^T Y$.

The final step of the recovery algorithm is again a greedy selection of the sparsest vectors among the candidates collected at the first step. As before, under the assumption $\mathbb{P}(R_{ij} = 0) = 0$, the greedy procedure successfully recovers X and A , provided that multiples of all the rows of X are present among the input set S .

The main result of this note is

Theorem 1 *There exist absolute constants $C, \alpha \in (0, \infty)$ such that if*

$$\frac{2}{n} \leq \theta \leq \frac{\alpha}{\sqrt{n}}$$

and X follows the Bernoulli-Subgaussian model with parameter θ , then for $p \geq Cn \log n$, with probability at least $1 - 1/p$ the modified ER-SpUD(DC) algorithm successfully recovers all the rows of X , i.e., multiples of all the rows of X are present among the vectors s_{ij} returned by the algorithm.

Remark on the single column algorithm. Spielman, Wang, and Wright (2012a,b) proposed also a version of the ER-SpUD algorithm, which instead of sums of two columns of Y as the vectors r_j in the constraints $r_j^T w = 1$ of the optimization problem, chooses simply consecutive columns of Y . They prove that such a version of the algorithm performs well under the assumption that the random variables X_{ij} are i.i.d. standard Gaussian variables, $2/n \leq \theta \leq \alpha/\sqrt{n \log n}$ and $p > Cn^2 \log^2 n$ (α, C are again universal constants). We remark that by using our approach in combination with the original arguments of Spielman, Wang, and Wright (2012a) one can prove that this algorithm works for $p > Cn \log^3 n$. To this end

one needs to prove a counterpart of our Lemma 3 (see below) with the vectors b_{ij} replaced just by the columns of the matrix X and combine it with Lemma 12 of Spielman, Wang, and Wright (2012a) (Lemma 6 below) in exactly the same way as in Section B.3. of (Spielman, Wang, and Wright, 2012a) (with $\gamma \simeq 1/\log n$). The needed counterpart of Lemma 3 can be obtained just by formal changes from the proof we present below. The factor $\log^3 n$ (instead of $\log n$) is related to the use of Lemma 12 and is a consequence of the fact that one takes γ depending on n).

Remarks on recent developments Very recently Sun, Qing, and Wright (2015) proposed an algorithm with polynomial sample complexity, which recovers well conditioned dictionaries under the assumption that the variables R_{ij} are i.i.d. standard Gaussian and $\theta \leq 1/2$, thus allowing for the first time for a linear number of nonzero entries per column of the matrix X . Their novel approach is based on non-convex optimization. The sample complexity of the algorithms in (Sun, Qing, and Wright, 2015) is however higher than for the Er-SpUD algorithm; as mentioned by the Authors, numerical simulations suggest that it is at least $p = \Omega(n^2 \log n)$ even in the case of orthogonal matrix A . Sun, Qing, and Wright (2015) conjecture that algorithms with sample complexity $p = O(n \log n)$ should be possible also for large θ .

As for the complexity of the Er-SpUD algorithm (in its original version), the recent article (Luh and Vu, 2016) contains a claim that it works for $p > Cn \log^4 n$, which differs from the number of samples conjectured by Spielman, Wang, and Wright (2012a) just by a polylogarithmic factor. However, as pointed out very recently (after the submission of the first version of this article) by Blasiok and Nelson (2016), the argument of Luh and Vu (2016) contains certain inaccuracies. Moreover, Blasiok and Nelson have proved that if the variables R_{ij} are Rademachers, than for the original version of the Er-SpUD algorithm to work one needs $p \geq n^{1.99}$, which shows that the result of Luh and Vu (2016) and in fact the original conjecture do not hold. Blasiok and Nelson also propose a modified version of the algorithm (in the same spirit as in this article) and prove that it works with probability $1 - \varepsilon$ for $p > Cn \log(Cn/\varepsilon)$, thus obtaining an independent proof of our main result. We remark that while certain aspects of the analysis are common for (Blasiok and Nelson, 2016) and the present article, the main technical ingredient (i.e., bounding the empirical process involved in the estimates) is approached differently. While Proposition 2 below is based on the contraction principle, Blasiok and Nelson (2016) rely on the generic chaining (majorizing measure) method, see (Talagrand, 2014). Let us also remark that it seems that by combining the inequality for empirical processes obtained by Luh and Vu (2016) with the approach of this paper or of (Blasiok and Nelson, 2016) one can prove a weaker result, namely that the modified version of the algorithm works for $p > Cn \log^4 n$.

2. Proof of Theorem 1

We will follow the general approach proposed by Spielman, Wang, and Wright (2012a). The main new part of the argument is an improved bound on the sample complexity for empirical approximation of first moments of arbitrary marginals of the columns of the matrix X , given in Proposition 2 below. So as not to reproduce technical and lengthy parts of the original proof, we organize this section as follows. First, we present the crucial Proposition 2 and provide a brief discussion of its mathematical content. Next, we present an overview of the

main steps in the proof scheme of Spielman, Wang, and Wright (2012a). For parts of the proof not related to Proposition 2 or to the modification of the algorithm considered here, we only indicate the relevant statements from (Spielman, Wang, and Wright, 2012a), while for parts involving the use of Proposition 2 and for the conclusion of the proof we provide the full argument. Proposition 2 is proved in Section 3.

Below by $\epsilon_1, \dots, \epsilon_N$ we will denote the standard basis in \mathbb{R}^N for various choices of N (in particular for $N = n$ and $N = p$). The value of N will be clear from the context and so this should not lead to ambiguity.

By B_1^n we will denote the unit ball in the space ℓ_1^n , i.e., $B_1^n = \{x \in \mathbb{R}^n : \|x\|_1 \leq 1\}$, where for $x = (x(1), \dots, x(n))$, $\|x\|_1 = \sum_{i=1}^n |x(i)|$. The coordinates of a vector x will be denoted by $x(i)$ or if it does not interfere with other notation (e.g. for indexed families of vectors) simply by x_i . Again, the meaning of the notation will be clear from the context. If Y is a random variable and $q > 0$, we denote $\|Y\|_q = (\mathbb{E}|Y|^q)^{1/q}$.

Proposition 2 *Let $U_1, U_2, \dots, U_p, \chi_1, \dots, \chi_p$ be independent random vectors in \mathbb{R}^n . Assume that for some constant M and all $1 \leq i \leq p, 1 \leq j \leq n$,*

$$\mathbb{E}e^{U_i(j)/M} \leq 2 \tag{1}$$

and

$$\mathbb{P}(\chi_i(j) = 1) = 1 - \mathbb{P}(\chi_i(j) = 0) = \theta.$$

Define the random vectors Z_1, \dots, Z_p with the equality $Z_i(j) = U_i(j)\chi_i(j)$ for $1 \leq i \leq p, 1 \leq j \leq n$ and consider the random variable

$$W := \sup_{x \in B_1^p} \left| \frac{1}{p} \sum_{i=1}^p (|x^T Z_i| - \mathbb{E}|x^T Z_i|) \right|. \tag{2}$$

Then, for some universal constant C and every $q \geq \max(2, \log n)$,

$$\|W\|_q \leq \frac{C}{p} (\sqrt{p\theta q} + q)M \tag{3}$$

and as a consequence

$$\mathbb{P}\left(W \geq \frac{C}{p} (\sqrt{p\theta q} + q)M\right) \leq e^{-q}. \tag{4}$$

The above proposition can be considered a quantitative version of the uniform law of large numbers for linear functionals $x^T Z$ indexed by the unit sphere in the space ℓ_1^p . As such it is a classical object of study in the theory of empirical processes. The proof we give uses only Bernstein's inequality, see e.g., (van der Vaart and Wellner, 1996), and Talagrand's contraction principle (Ledoux and Talagrand, 1991), which in a somewhat similar context was applied e.g., by Mendelson (2008); Adamczak et al. (2010).

Let us also remark that in the above proposition we do not require independence between components of the random vectors U_i or χ_i for fixed i , but just independence between the random vectors $U_i, \chi_i, i = 1, \dots, p$.

2.1 Main Steps of the Proof of Theorem 1

As announced, we will now present an outline of the proof of Theorem 1, indicating which steps differ from the original argument in (Spielman, Wang, and Wright, 2012a).

Step 1. A change of variables.

Recall that r_{ij} are sums of two columns of the matrix Y . At the first step of the proof, instead of looking at the original optimization problem

$$\text{minimize } \|w^T Y\|_1 \text{ subject to } r_{ij}^T w = 1 \quad (5)$$

one performs a change of variables $z = A^T w$, $b_{ij} = A^{-1} r_{ij}$, arriving at the optimization problem

$$\text{minimize } \|z^T X\|_1 \text{ subject to } b_{ij}^T z = 1. \quad (6)$$

Note that one cannot solve (6) since it involves the unknown matrices X and A . The goal of the subsequent steps is to prove that with probability sufficiently separated from zero the solution z_* of (6) is a multiple of one of the basis vectors e_1, \dots, e_n , say $z_* = \lambda e_k$. This means that $w_*^T Y = z_*^T X = \lambda e_k^T X$, i.e., (5) recovers the k -th row of X up to scaling. In combination with a coupon collector phenomenon this will allow to conclude that if p is sufficiently large, then all the rows will be recovered (this is the content of Step 4).

Step 2. If $0 < |(\text{supp } X_{e_j}) \cup (\text{supp } X_{e_j})| < 1/(8\theta)$, then $\text{supp } (z_*) \subseteq (\text{supp } X_{e_j}) \cup (\text{supp } X_{e_j})$.

At this step we prove the following lemma, which is a counterpart of Lemma 11 in (Spielman et al., 2012a). It is weaker in that we do not consider arbitrary vectors b_{ij} , but only sums of two distinct columns of X (which is enough for the application in the proof of Theorem 1). On the other hand it works already for $p > Cn \log n$ and not for $p > Cr^2 \log n$ as the original lemma from (Spielman, Wang, and Wright, 2012a).

Lemma 3 For $1 \leq i < j \leq p$, define $b_{ij} = X_{e_i} + X_{e_j}$, $I_{ij} = (\text{supp } X_{e_i}) \cup (\text{supp } X_{e_j})$.

There exist numerical constants $C, \alpha > 0$ such that if $2/n \leq \theta \leq \alpha/\sqrt{n}$ and $p > Cn \log n$, then with probability at least $1 - p^{-2}$ the random matrix X has the following property:

(P1) For every $1 \leq i < j \leq p$, if $0 < |I_{ij}| \leq 1/(8\theta)$ then every solution z_* to the optimization problem (6) satisfies $\text{supp } z_* \subseteq I_{ij}$.

Before we pass to the presentation of auxiliary facts needed in the proof of the above lemma, let us indicate briefly the two main observations behind the lemma, not present in (Spielman, Wang, and Wright, 2012a). The first one is Proposition 2, which allows to prove the technical Lemma 5 below. The second one is the fact that due to independence of the entries of the matrix we do not need to use the union bound over all possible locations of nonzero coefficients of X_{e_i} and X_{e_j} , instead we can condition on the disjoint events that $(\text{supp } X_{e_i}) \cup (\text{supp } X_{e_j}) = I$ (where I ranges over nonempty subsets of $[n]$ with $|I| \leq 1/(8\theta)$), estimate appropriate conditional probabilities and integrate the result over I .

To prove Lemma 3, one first shows a counterpart of Lemma 16 in (Spielman, Wang, and Wright, 2012a).

Lemma 4 For any $1 \leq j \leq p$, if $Z = (X_{1j} R_{1j}, \dots, X_{nj} R_{nj})$, then for all $v \in \mathbb{R}^n$,

$$\mathbb{E} \|v^T Z\| \geq \frac{\mu}{8} \sqrt{\frac{\theta}{n}} \|v\|_1.$$

Proof Let $\varepsilon_1, \dots, \varepsilon_n$ be a sequence of i.i.d. Rademacher variables, independent of $\{X_{ij}, R_{ij}\}$. By standard symmetrization inequalities, see e.g., Lemma 6.3. in (Ledoux and Talagrand, 1991),

$$\mathbb{E} \|v^T R\| = \mathbb{E} \left| \sum_{i=1}^n v_i X_{ij} R_{ij} \right| \geq \frac{1}{2} \mathbb{E} \left| \sum_{i=1}^n v_i \varepsilon_i X_{ij} R_{ij} \right|.$$

The random variables $\varepsilon_i R_{ij}$ are symmetric and $\mathbb{E} |\varepsilon_i R_{ij}| = \mu$, so by Lemma 16 from (Spielman, Wang, and Wright, 2012a), the right-hand side above is bounded from below by $\frac{\mu}{8} \sqrt{\frac{\theta}{n}} \|v\|_1$. ■

The next lemma is an improvement of Lemma 17 in (Spielman, Wang, and Wright, 2012a), which is crucial for obtaining Lemma 3.

Lemma 5 There exists an absolute constant C , such that the following holds for $p > Cn \log n$. Let $J \subseteq \{1, \dots, p\}$ be a fixed subset of size $|J| \leq \frac{p}{4}$. Let X_J be the submatrix of X , obtained by a restriction of X to the columns indexed by J . With probability at least $1 - p^{-8}$, for any $v \in \mathbb{R}^n$,

$$\|v^T X\|_1 - 2\|v^T X_J\|_1 > \frac{p\mu}{32} \sqrt{\frac{\theta}{n}} \|v\|_1.$$

Proof Note first that by increasing the set J , we increase $\|v^T X_J\|_1$, so without loss of generality we can assume that $|J| = \lfloor p/4 \rfloor$. Apply Proposition 2 with the vectors $U_j = (R_{1j}, \dots, R_{nj})$ and $X_j = (X_{1j}, \dots, X_{nj})$ and $q = 8 \log p$. Note that our integrability assumptions on R_{ij} imply (1) with M being a universal constant. Therefore, for some absolute constant C and $p \geq Cn \log n$, with probability at least $1 - p^{-8}$ we have

$$\begin{aligned} \sup_{v \in B_1^n} \left| \|v^T X\|_1 - \mathbb{E} \|v^T X\|_1 \right| &\leq C \sqrt{p\theta \log p} + \log p \leq 2C \sqrt{p\theta \log p}, \\ \sup_{v \in B_1^n} \left| \|v^T X_J\|_1 - \mathbb{E} \|v^T X_J\|_1 \right| &\leq 2C \sqrt{p\theta \log p}, \end{aligned}$$

where we used that for C sufficiently large, $p/\log p \geq n \geq 1/\theta$.

Thus, by homogeneity, with probability at least $1 - p^{-8}$, for all $v \in \mathbb{R}^n$,

$$\begin{aligned} \left| \|v^T X\|_1 - \mathbb{E} \|v^T X\|_1 \right| &\leq 2C \sqrt{p\theta \log p} \|v\|_1, \\ \left| \|v^T X_J\|_1 - \mathbb{E} \|v^T X_J\|_1 \right| &\leq 2C \sqrt{p\theta \log p} \|v\|_1. \end{aligned}$$

In particular this means that (using the notation of Proposition 2)

$$\begin{aligned} \|v^T X\|_1 &\geq \mathbb{E} \|v^T X\|_1 - 2C \sqrt{p\theta \log p} \|v\|_1 = p \mathbb{E} \|v^T Z\|_1 - 2C \sqrt{p\theta \log p} \|v\|_1, \\ 2\|v^T X_J\|_1 &\leq 2\mathbb{E} \|v^T X_J\|_1 + 4C \sqrt{p\theta \log p} \|v\|_1 = 2|J| \mathbb{E} \|v^T Z\|_1 + 4C \sqrt{p\theta \log p} \|v\|_1, \end{aligned}$$

and so

$$\begin{aligned} \|v^T X\|_1 - 2\|v^T X_J\|_1 &\geq (p-2|J|)\mathbb{E}\|v^T Z_1\| - 6C\sqrt{\theta p \log p}\|v\|_1. \\ \|v^T X\|_1 - 2\|v^T X_J\|_1 &\geq \left(\frac{p\mu}{16}\sqrt{\frac{\theta}{n}} - 6C\sqrt{\theta p \log p}\right)\|v\|_1 > \frac{p\mu}{32}\sqrt{\frac{\theta}{n}}\|v\|_1 \end{aligned}$$

Now, by Lemma 4 and the assumed bound on the cardinality of J , we get

■

for $p > C'n \log n$, where C' is another absolute constant.

We are now in position to prove Lemma 3.

Proof of Lemma 3 We will show that for each $1 \leq i < j \leq p$ the probability that $0 < |I_{ij}| \leq 1/(8\theta)$ and there exists a solution to (6) not supported on I_{ij} is bounded from above by $1/p^4$. This will imply the lemma, since by the union bound over all $i < j$,

$$\begin{aligned} \mathbb{P}(\text{Property P1 does not hold}) & \quad (7) \\ \leq \sum_{1 \leq i < j \leq p} \mathbb{P}(0 < |I_{ij}| \leq 1/(8\theta) \ \& \ \text{there exists a solution } z_* \text{ to (6) not supported on } I_{ij}). \end{aligned}$$

Let us thus fix i, j and let

$$S = \{I \in [p] : \exists_{k \in I_{ij}} X_{ki} \neq 0\}.$$

Moreover, for any set $I \subseteq [n]$, define the event

$$\mathcal{A}_I = \{I_{ij} = I\}.$$

By independence of the random variables R_{ij}, χ_{ij} , for each $k \notin \{i, j\}$, if $0 < |I| \leq 1/(8\theta)$, then

$$\mathbb{P}(k \in S | \mathcal{A}_I) \leq 1 - (1 - \theta)^{|I|} \leq 1 - e^{-2\theta|I|} \leq 1 - e^{-\frac{1}{4}} < \frac{1}{4},$$

where the second inequality holds if α is sufficiently small.

Thus, by independence of columns of X and Hoeffding's inequality, if $0 < |I| \leq 1/(8\theta)$, then

$$\mathbb{P}\left(|S \setminus \{i, j\}| \leq \frac{p-2}{4} \mid \mathcal{A}_I\right) \geq 1 - 2e^{-c\theta} \quad (8)$$

for some universal constant $c > 0$.

Let z_* be any solution of (6) and denote by z_0 its orthogonal projection on $\mathbb{R}^{I_{ij}} = \{x \in \mathbb{R}^n : x_k = 0 \text{ for } k \notin I_{ij}\}$. Set also $z_1 = z_* - z_0$ and let X_S, X_{S^c} be the matrices obtained from X by selecting the columns labeled by S and $S^c = [p] \setminus S$ respectively. By the triangle inequality, and the fact that $z_0^T X_{S^c} = 0$, we get

$$\begin{aligned} \|z_*^T X\|_1 &= \|(z_0^T + z_1^T)X_S\|_1 + \|(z_0^T + z_1^T)X_{S^c}\|_1 \\ &\geq \|z_0^T X_S\|_1 - \|z_1^T X_S\|_1 + \|z_1^T X\|_1 - \|z_1^T X_{S^c}\|_1 \\ &= \|z_0^T X\|_1 + (\|z_1^T X\|_1 - 2\|z_1^T X_S\|_1). \end{aligned} \quad (9)$$

For $J \subseteq [p] \setminus \{i, j\}$ define the events

$$\mathcal{S}_J = \{S \setminus \{i, j\} = J\}.$$

For the moment let us restrict our attention to the event $\mathcal{A}_I \cap \mathcal{S}_J$, for a fixed (but arbitrary) $I \subseteq [n]$, satisfying $0 < |I| \leq 1/(8\theta)$ and $J \subseteq [p] \setminus \{i, j\}$, satisfying $|J| \leq (p-2)/4$.

Denote by X' the $|I^c| \times (p-2)$ matrix obtained by restricting X to the rows from I^c and columns from $[p] \setminus \{i, j\}$. If, slightly abusing the notation, we identify z_1 with a vector from $\mathbb{R}^{|I^c|}$, on the event $\mathcal{A}_I \cap \mathcal{S}_J$ we have

$$\|z_1^T X\|_1 - 2\|z_1^T X_S\|_1 = \|z_1^T X'\|_1 - 2\|z_1^T X'_S\|_1 = \|z_1^T X'\|_1 - 2\|z_1^T X'_J\|_1, \quad (10)$$

where in the first equality we used the fact that $z_1^T X e_i = z_1^T X e_j = 0$ and the second one follows from the definition of the event \mathcal{S}_J .

Due to independence and identical distribution of the entries of X , conditionally on the event $\mathcal{A}_I \cap \mathcal{S}_J$ the matrix X' still follows the Bernoulli-Subgaussian model with parameter θ . This matrix is of size $|I^c| \times (p-2)$, therefore if the constant C' is large enough and $p > C'n \log n$, it satisfies the assumptions of Lemma 5 (with $p-2$ instead of p and $|I^c|$ instead of n). Since $|J| \leq (p-2)/4$, a conditional application of Lemma 5 gives

$$\begin{aligned} \mathbb{P}\left(\text{for all } v \in \mathbb{R}^{|I^c|} : \|v^T X'\|_1 - 2\|v^T X'_J\|_1 \geq \frac{(p-2)\mu}{32} \sqrt{\frac{\theta}{|I^c|}} \|v\|_1 \mid \mathcal{A}_I \cap \mathcal{S}_J\right) & \quad (11) \\ \geq 1 - (p-2)^{-8} \geq 1 - 2p^{-8}, \end{aligned}$$

where the last inequality holds for $p > C'$ and C' sufficiently large.

Note that by the definition of z_0 , we have $b_{ij}^T z_0 = b_{ij}^T z_* = 1$, therefore z_0 is a feasible candidate for the solution of the optimization problem (6). Thus, by (9) and (10), we have $\|z_1^T X'\|_1 - 2\|z_1^T X'_J\|_1 \leq 0$ and as a consequence, if $z_1 \neq 0$ then the event of inequality (11) does not hold. Thus, for $0 < |I| \leq 1/(8\theta)$ and $|J| \leq (p-2)/4$, we get

$$\mathbb{P}(\text{for some solution } z_* \text{ to (6), } z_1 \neq 0 \mid \mathcal{A}_I \cap \mathcal{S}_J) \leq 2p^{-8}. \quad (12)$$

We are now ready to finish the proof. To shorten the notation, let us denote

$$\mathcal{B} = \{\text{for some solution } z_* \text{ to (6), } z_1 \neq 0 \text{ and } 0 < |I_{ij}| \leq 1/(8\theta)\}.$$

By (8) we get

$$\begin{aligned} \mathbb{P}\left(\mathcal{B} \cap \{|S \setminus \{i, j\}| > (p-2)/4\}\right) &= \sum_{I \subseteq [n] : 0 < |I| \leq 1/(8\theta)} \mathbb{P}(\mathcal{B} \cap \mathcal{A}_I \cap \{|S'| > (p-2)/4\}) \\ &\leq \sum_{I \subseteq [n] : 0 < |I| \leq 1/(8\theta)} \mathbb{P}(\mathcal{A}_I \cap \{|S'| > (p-2)/4\}) \\ &= \sum_{I \subseteq [n] : 0 < |I| \leq 1/(8\theta)} \mathbb{P}(|S'| > (p-2)/4 \mid \mathcal{A}_I) \\ &\leq 2e^{-c\theta} \sum_{I \subseteq [n] : 0 < |I| \leq 1/(8\theta)} \mathbb{P}(\mathcal{A}_I) \leq 2e^{-c\theta}, \end{aligned}$$

where the second to last inequality follows from (8) and the last one from the pairwise disjointness of the events \mathcal{A}_I .

Similarly,

$$\begin{aligned} \mathbb{P}(\mathcal{B} \cap \{S \setminus \{i, j\}\}) &\leq (p-2)/4) = \sum_{\substack{J \subseteq [n]: \\ 0 < |J| \leq 1/(8\theta)}} \sum_{\substack{J \subseteq [p] \setminus \{i, j\}: \\ |J| \leq (p-2)/4}} \mathbb{P}(\mathcal{B} \cap \mathcal{A}_I \cap S_J) \\ &\leq \sum_{\substack{J \subseteq [n]: \\ 0 < |J| \leq 1/(8\theta)}} \sum_{\substack{J \subseteq [p] \setminus \{i, j\}: \\ |J| \leq (p-2)/4}} \mathbb{P}(\mathcal{B}_I \mathcal{A}_I \cap S_J) \mathbb{P}(\mathcal{A}_I \cap S_J) \\ &\leq 2p^{-8} \sum_{\substack{J \subseteq [n]: \\ 0 < |J| \leq 1/(8\theta)}} \sum_{\substack{J \subseteq [p] \setminus \{i, j\}: \\ |J| \leq (p-2)/4}} \mathbb{P}(\mathcal{A}_I \cap S_J) \leq 2p^{-8}, \end{aligned}$$

where we used (12) and disjointness of the events $\mathcal{A}_I \cap S_J$. Combining the two last inequalities, we get

$$\mathbb{P}(\mathcal{B}) \leq 2e^{-cp} + 2p^{-8} \leq p^{-4}$$

for $p > Cn \log n$ with a sufficiently large absolute constant C . By (7) this ends the proof of the lemma. \blacksquare

Step 3. If $(\text{supp } X_{e_j}) \cup (\text{supp } X_{e_j})$ is small, then $z_k = \lambda e_k$ where $k = \text{argmax}_{1 \leq i \leq n} |b_{ij}(t)|$.

At this step one proves the following lemma (Lemma 12 in Spielman, Wang, and Wright 2012a). Since no changes with respect to the original argument are required (we do not use Proposition 2 here), we do not reproduce the proof and refer the Reader to (Spielman, Wang, and Wright, 2012a) for details. We remark that although the lemma is formulated therein for symmetric variables, the symmetry assumption is not used in its proof.

Below, by $|b_1^\dagger| \geq |b_2^\dagger| \geq \dots \geq |b_n^\dagger|$ we denote the nonincreasing rearrangement of the sequence $|b_1|, \dots, |b_n|$, while for $J \subseteq [n]$, X^J denotes the matrix obtained from X by selecting the rows indexed by the set J .

Lemma 6 *There exist two positive constants c_1, c_2 such that the following holds. For any $\gamma > 0$ and $s \in \mathbb{Z}_+$, such that $\theta s < \gamma/8$ and p such that*

$$p \geq \max \left\{ \frac{c_1 s \log n}{\theta \gamma^2}, n \right\}, \quad \text{and} \quad \frac{p}{\log p} \geq \frac{c_2}{\theta \gamma^2},$$

with probability at least $1 - 4p^{-10}$, the random matrix X has the following property:

(P2) For every $J \subseteq [n]$ with $|J| = s$ and every $b \in \mathbb{R}^s$, satisfying $\frac{|b_i^\dagger|}{|b_i|} \leq 1 - \gamma$, the solution to the restricted problem

$$\text{minimize } \|z^T X^J\|_1 \text{ subject to } b^T z = 1, \quad (13)$$

is unique, 1-sparse, and is supported on the index of the largest entry of b .

Step 4. Conclusion of the proof.

Set $s = 12\theta n + 1$. Our first goal is to prove that with probability at least $1 - 1/p^2$, for all $k \in [n]$, there exist $i, j \in [p]$, $i \neq j$ such that the vector $b = X_{e_i} + X_{e_j}$ satisfies the assumptions of Lemma 6, $|b_1^\dagger| = |b_k|$ and $I_{ij} := (\text{supp } X_{e_i}) \cup (\text{supp } X_{e_j})$ satisfies $0 < |I_{ij}| \leq 1/(8\theta)$, which will allow us to take advantage of Lemma 3. This will already imply that the solution to the problem (6) for such i, j produces a multiple of the k -th row of X .

Note that we have

$$\mathbb{E} R_{ij}^2 \leq 4 \int_0^\infty t e^{-t^2/2} dt = 4.$$

Since $\mathbb{E}|R_{ij}| = \mu \geq \frac{1}{10}$, by the Paley-Zygmund inequality, see e.g., Corollary 3.3.2. in (de la Peña and Giné, 1999), we have

$$\mathbb{P}(|R_{ij}| \geq \frac{1}{20}) \geq \frac{3}{4} \frac{(\mathbb{E}|R_{ij}|)^2}{\mathbb{E} R_{ij}^2} \geq c_0$$

for some universal constant $c_0 > 0$. In particular $\mathbb{P}(|R_{ij}| = 0) < 1 - \frac{c_0}{2}$. Let q be any $(1 - c_0/(2s))$ -quantile of $|R_{ij}|$, i.e., $\mathbb{P}(|R_{ij}| \leq q) \geq (1 - c_0/(2s))$ and $\mathbb{P}(|R_{ij}| \geq q) \geq c_0/(2s)$. In particular, since $s \geq 1$, we get $q > 0$. We have $\mathbb{P}(R_{ij} \geq q) \geq c_0/(4s)$ or $\mathbb{P}(R_{ij} \leq -q) \geq c_0/(4s)$. Let us assume that $\mathbb{P}(R_{ij} \geq q) \geq c_0/(4s)$, the other case is analogous.

Before we proceed with the formal proof, which due to many events under consideration may appear technical, let us provide its informal description. Let us focus on a single value of k (at the end of the argument we will take a union bound over all $k \leq n$). We will first prove that among the first $p/2$ columns of the matrix X there is one (say X_{e_i}) which has few nonzero entries, the k -th entry exceeds the quantile q and all the other entries are smaller than q in the absolute value. This corresponds to the events \mathcal{E}_{ki} and \mathcal{A}_k considered below. Once we establish that this holds with high probability (equation (14)), we will fix a single column with this property (say with the smallest index) and will prove that conditionally on this event among the $p/2$ last columns of X we can find a column (say X_{e_j}) with the same properties and such that the only entry which is nonzero in both X_{e_i} and X_{e_j} is the k -th one (which corresponds to the event \mathcal{B}_k below and is the content of equation (17)). This will imply that

- the set I_{ij} satisfies the premises of Lemma 3 (it is nonempty and not too large),
- the k -th entry of $b_{ij} = X_{e_i} + X_{e_j}$ exceeds $2q$ while all the other entries are smaller than q in absolute value, which allows to use Lemma 6 with $\gamma = 1/2$.

Combining the two lemmas will allow us to conclude that the solution to (6) produces a nonzero multiple of e_k , i.e., the solution to (5) produces a nonzero multiple of the k -th row of X .

Establishing the aforesaid properties is not difficult and relies just on the independence of entries. In essence it can be reduced to saying that in a sequence of Bernoulli trials with probability of success equal to ρ , it is highly unlikely that we will have to wait much longer than $1/\rho$ for the first success. Specifically, if $\rho > c/n$, then under our assumptions on p , the probability that no success occurs in $p/2$ steps is smaller than $1/p^4$ (see e.g.,

equation (16) below). In the proof the trials correspond to columns of X and success to the conjunction of the properties stated above. Both parts of the proof rely on estimating the probability of success from below (in the second part it is the conditional probability, since the event in question depends on the first part). The main reason behind technical (notational) difficulties is that one needs to explore independence of the variables χ_{ij} and R_{ij} in the right order to be able to take advantage of the already established bounds in consecutive steps.

Define thus the event \mathcal{E}_{ki} as

$$\mathcal{E}_{ki} = \left\{ \chi_{ki} = 1, |\{r \in [n] \setminus \{k\} : \chi_{ri} = 1\}| \leq (s-1)/2, R_{ki} \geq q, \forall_{r \neq k} \chi_{ri} = 1 \implies |R_{ri}| \leq q \right\}$$

(see the description above for the motivation of this and subsequent definitions).

We will assume that $p \geq 2Cn \log n$ for some numerical constant C to be fixed later on. For $k \in [n]$, consider the events

$$\mathcal{A}_k = \bigcup_{1 \leq i \leq \lfloor p/2 \rfloor} \mathcal{E}_{ki}$$

and

$$\mathcal{B}_k = \bigcup_{1 \leq i \leq \lfloor p/2 \rfloor} \bigcup_{\lfloor p/2 \rfloor < j \leq p} (\mathcal{E}_{ki} \cap \mathcal{E}_{kj} \cap \{l \in [n] : \chi_{li} = \chi_{lj} = 1\} = \{k\}).$$

We will first show that for all $k \in [n]$,

$$\mathbb{P}(\mathcal{A}_k) \geq 1 - \frac{1}{p^4}, \tag{14}$$

which we will use to prove that

$$\mathbb{P}(\mathcal{B}_k) \geq 1 - \frac{1}{p^3}. \tag{15}$$

Let us start with the proof of (14). Set $\mathcal{B}_{ki} = \{|\{r \in [n] \setminus \{k\} : \chi_{rk} = 1\}| \leq (s-1)/2\}$. By independence we have

$$\begin{aligned} \mathbb{P}(\mathcal{E}_{ki}) &= \mathbb{P}(\chi_{ki} = 1) \mathbb{P}(R_{ki} \geq q) \mathbb{P}(\mathcal{B}_{ki}) \mathbb{P}(\forall_{r \neq k} \chi_{ri} = 1 \implies |R_{ri}| \leq q | \mathcal{B}_{ki}) \\ &\geq \theta \frac{c_0}{4s} \left(1 - \frac{c_0}{2s}\right) \left(\frac{c_0}{4s}\right)^{(s-1)/2}, \end{aligned}$$

where to estimate $\mathbb{P}(\mathcal{B}_{ki})$ we used Markov's inequality. The last factor comes from the definition of q as the $(1 - c_0/(2s))$ -quantile of R_{ij} . The right hand side above is bounded from below by c_1/n for some universal constant c_1 . Therefore if the constant C is large enough, we obtain

$$\mathbb{P}\left(\bigcap_{1 \leq i \leq \lfloor p/2 \rfloor} \mathcal{E}_{ki}^c\right) \leq \left(1 - \frac{c_1}{n}\right)^{\lfloor p/2 \rfloor} \leq \exp(-c_1 p / (4n)) \leq \exp(-4 \log p) = \frac{1}{p^4}, \tag{16}$$

where we used the inequality $p / \log p \geq 16c_1^{-1}n$ for $p \geq Cn \log n$. We have thus established (14).

Let us now pass to (15). Denote by \mathcal{F}_1 the σ -field generated by $\chi_{ki}, R_{ki}, k \in [n], 1 \leq i \leq \lfloor p/2 \rfloor$. Note that $\mathcal{A}_k \in \mathcal{F}_1$.

For $\omega \in \mathcal{A}_k$ define $i_{\min}(\omega) = \min\{1 \leq i \leq \lfloor p/2 \rfloor : \omega \in \mathcal{E}_{ki}\}$. Note that on \mathcal{A}_k ,

$$\mathbb{P}(\mathcal{B}_k | \mathcal{F}_1) \geq \mathbb{P}\left(\bigcup_{\lfloor p/2 \rfloor < j \leq p} (\mathcal{E}_{kj} \cap \{l \in [n] : \chi_{li_{\min}} = \chi_{lj} = 1\} = \{k\}) \mid \mathcal{F}_1\right)$$

Define

$$\mathcal{C}_{kj} = \{|\{r \in [n] \setminus \{k\} : \chi_{rj} = 1\}| \leq (s-1)/2\} \cap \{l \in [n] : \chi_{li_{\min}} = \chi_{lj} = 1\} = \{k\}.$$

Similarly as in the argument leading to (14), for fixed j , using the independence of the variables χ_{lm}, R_{lm} and properties of the conditional probability, we obtain on the event \mathcal{A}_k ,

$$\begin{aligned} \mathbb{P}(\mathcal{E}_{kj} \cap \{l \in [n] : \chi_{li_{\min}} = \chi_{lj} = 1\} = \{k\} \mid \mathcal{F}_1) &= \mathbb{P}(R_{kj} \geq q) \mathbb{E}\left(\mathbf{1}_{\mathcal{C}_{kj}} \mathbb{P}(\forall_{r \neq k} \chi_{rj} = 1 \implies |R_{rj}| \leq q | \mathcal{C}_{kj}, \mathcal{F}_1) \mid \mathcal{F}_1\right) \\ &\geq \mathbb{P}(R_{kj} \geq q) \mathbb{E}\left(\mathbf{1}_{\mathcal{C}_{kj}} \left(1 - \frac{c_0}{2s}\right)^{\frac{s-1}{2}} \mid \mathcal{F}_1\right) \\ &= \mathbb{P}(R_{kj} \geq q) \left(1 - \frac{c_0}{2s}\right)^{\frac{s-1}{2}} \mathbb{P}(\mathcal{C}_{kj} | \mathcal{F}_1) \\ &\geq \frac{c_0}{4s} \left(1 - \frac{c_0}{2s}\right)^{\frac{s-1}{2}} \times \\ &\quad \left(\mathbb{P}\left(\{l \in [n] : \chi_{li_{\min}} = \chi_{lj} = 1\} = \{k\} \mid \mathcal{F}_1\right) - \mathbb{P}\left(\chi_{kj} = 1, |\{r \in [n] \setminus \{k\} : \chi_{rj} = 1\}| > \frac{s-1}{2} \mid \mathcal{F}_1\right)\right) \\ &\geq \frac{c_0}{4s} \left(1 - \frac{c_0}{4s}\right)^{\frac{s-1}{2}} \left(\theta(1-\theta)^{(s-1)/2} - \theta \frac{2\theta(n-1)}{s-1}\right), \end{aligned}$$

where in the last line we again used Markov's inequality.

Now recall that $\theta \leq \frac{\alpha}{\sqrt{n}}$ for some universal constant α . If α is small enough then $1 - \theta \geq e^{-2\theta}$ and

$$(1 - \theta)^{(s-1)/2} \geq e^{-\theta(s-1)} = e^{-12\theta^2 n} \geq e^{-12\alpha^2} \geq \frac{1}{3}.$$

Since $\frac{2\theta(n-1)}{s-1} \leq \frac{1}{6}$, this implies that

$$\mathbb{P}(\mathcal{E}_{kj} \cap \{l \in [n] : \chi_{li_{\min}} = \chi_{lj} = 1\} = \{k\} \mid \mathcal{F}_1) \geq \frac{c_2}{n}$$

for some positive universal constant c_2 . Since the events $\mathcal{E}_{kj} \cap \{l \in [n] : \chi_{li_{\min}} = \chi_{lk} = 1\} = \{k\}$, $\lfloor p/2 \rfloor < k \leq p$ are conditionally independent, given \mathcal{F}_1 , we obtain that on \mathcal{A}_k ,

$$\mathbb{P}(\mathcal{B}_k^c | \mathcal{F}_1) \leq \left(1 - \frac{c_2}{n}\right)^{\lfloor p/2 \rfloor} \leq \frac{1}{p^4}, \tag{17}$$

provided C is a sufficiently large universal constant. Now, using (14), we get

$$\mathbb{P}(\mathcal{B}_k) \geq \mathbb{E} \mathbf{1}_{\mathcal{A}_k} \mathbb{P}(\mathcal{B}_k | \mathcal{A}_k) \geq \mathbb{P}(\mathcal{A}_k) \left(1 - \frac{1}{p}\right) \geq \left(1 - \frac{1}{p}\right)^2 \geq 1 - \frac{1}{p^3},$$

proving (15).

Taking the union bound over $k \in [n]$, we get

$$\mathbb{P}\left(\bigcap_{1 \leq k \leq n} \mathcal{B}_k\right) \geq 1 - \frac{1}{p^2}.$$

Set $\gamma = 1/2$ and observe that if C is large enough and α small enough, then the assumptions of Lemma 3 and Lemma 6 are satisfied. In particular $s = 12\theta n + 1 \leq \frac{\gamma}{s\theta} \leq \frac{1}{s\theta}$. Recall the properties P1 and P2 considered in the said lemmas. Consider the event $\mathcal{A} = \bigcap_{1 \leq k \leq n} \mathcal{B}_k \cap \{\text{properties P1 and P2 hold}\}$ and note that $\mathbb{P}(\mathcal{A}) \geq 1 - \frac{1}{p}$. On the event \mathcal{A} , for every k , there exist $1 \leq i < j \leq p$, such that

- $1 \leq |I_{ij}| \leq s \leq \frac{\gamma}{s\theta} \leq \frac{1}{s\theta}$,
- the largest entry of b_{ij} (in absolute value) equals $b_{ij}(k) \geq 2q > 0$ whereas the remaining entries do not exceed q ,

In particular, by property P1 we obtain that any solution z_* to the problem (6) satisfies $\text{supp } z_* \subseteq I_{ij}$. Therefore for some (any) $J \supseteq I_{ij}$ with $|J| = |s|$, we obtain (identifying vectors supported on J with their restrictions to J), that z_* is in fact a solution to the restricted problem (13) with $b = b_{ij}$, which by property P2 implies that $z_* = \lambda e_k$ for some $\lambda \neq 0$.

According to the discussion at the beginning of Step 1, this means that the solution w_* to (5) satisfies $w_*^T Y = \lambda e_k^T X$, i.e., the algorithm, when analyzing the vector b_{ij} , will add a multiple of the k -th row of X to the collection S .

This ends the proof of Theorem 1. \blacksquare

3. Proof of Proposition 2

The first tool we will need is the classical Bernstein's inequality, see e.g., Lemma 2.2.11 in (van der Vaart and Wellner, 1996).

Lemma 7 (Bernstein's inequality) *Let Y_1, \dots, Y_p be independent mean zero random variables such that for some constants M, v and every integer $k \geq 2$, $\mathbb{E}|Y_i|^k \leq k!M^{k-2}v/2$. Then, for every $t > 0$,*

$$\mathbb{P}\left(\left|\sum_{i=1}^p Y_i\right| \geq t\right) \leq 2 \exp\left(-\frac{t^2}{2(pv + Mt)}\right).$$

As a consequence, for every $q \geq 2$,

$$\left\|\sum_{i=1}^p Y_i\right\|_q \leq C(\sqrt{qp} + qM), \quad (18)$$

where C is a universal constant.

Another (also quite standard) tool we will rely on is the contraction principle for empirical processes due to Talagrand, see Theorem 4.12. in (Ledoux and Talagrand, 1991).

Lemma 8 (Talagrand's contraction principle) *Let $F: \mathbb{R}_+ \rightarrow \mathbb{R}_+$ be convex and increasing. Let further $\varphi: \mathbb{R} \rightarrow \mathbb{R}$ be a 1-Lipschitz function such that $\varphi(0) = 0$. For every bounded subset T of \mathbb{R}^n , if $\varepsilon_1, \dots, \varepsilon_n$ are i.i.d. Rademacher variables, then*

$$\mathbb{E}F\left(\sup_{t \in T} \frac{1}{2} \left|\sum_{i=1}^n \varphi(t_i) \varepsilon_i\right|\right) \leq \mathbb{E}F\left(\sup_{t \in T} \left|\sum_{i=1}^n t_i \varepsilon_i\right|\right)$$

We can now present the proof of Proposition 2.

Proof of Proposition 2 Let $\varepsilon_1, \dots, \varepsilon_p$ be i.i.d. Rademacher variables, independent of the sequences (U_i) , (X_i) . By the symmetrization inequality, see e.g., (Ledoux and Talagrand, 1991, Lemma 6.3.) or (van der Vaart and Wellner, 1996, Lemma 2.31), we have

$$\mathbb{E}W^q \leq 2^q \mathbb{E} \sup_{x \in \mathcal{B}_1^p} \left| \sum_{i=1}^p \varepsilon_i x^T Z_i \right|^q.$$

Now, since the function $t \mapsto |t|$ is a contraction, an application of Lemma 8 with $F(x) = |x|^q$, conditionally on Z_i , gives

$$\begin{aligned} \mathbb{E}W^q &\leq 2^{2q} \mathbb{E} \sup_{x \in \mathcal{B}_1^p} \left| \sum_{i=1}^p \varepsilon_i x^T Z_i \right|^q = \frac{2^{2q}}{p^q} \mathbb{E} \sup_{x \in \mathcal{B}_1^p} \left| x^T \sum_{i=1}^p \varepsilon_i Z_i \right|^q \\ &= \frac{2^{2q}}{p^q} \mathbb{E} \left\| \sum_{i=1}^p \varepsilon_i Z_i \right\|_\infty^q = \frac{2^{2q}}{p^q} \mathbb{E} \max_{1 \leq j \leq n} \left| \sum_{i=1}^p \varepsilon_i Z_i(j) \right|^q \\ &\leq \frac{2^{2q}}{p^q} \sum_{j=1}^n \mathbb{E} \left| \sum_{i=1}^p \varepsilon_i Z_i(j) \right|^q. \end{aligned} \quad (19)$$

Now, for every i, j and every integer $k \geq 2$ we have

$$\mathbb{E}|Z_i(j)|^k = \theta \mathbb{E}|U_i(j)|^k \leq \theta M^k k! \mathbb{E}e^{|U_i(j)|/M} \leq 2k! \theta M^k = k! v M^{k-2} / 2$$

with $v = 4\theta M^2$. Thus by the moment version (18) of Bernstein's inequality for some universal constant C we get

$$\mathbb{E} \left| \sum_{i=1}^p \varepsilon_i X_i(j) \right|^q \leq C^q \left(\sqrt{qp\theta} M + qM \right)^q,$$

which, when combined with (19), yields for $q \geq \log n$,

$$\|W\|_q \leq \frac{4C^q}{p} (\sqrt{p\theta}q + q)M.$$

The first part of the proposition follows by adjusting the constant C . The tail bound is a direct consequence of the Chebyshev inequality for the q -th moment. \blacksquare

References

- R. Adamczak, A. E. Litvak, A. Pajor, and N. Tomczak-Jaegermann. Quantitative estimates of the convergence of the empirical covariance matrix in log-concave ensembles. *J. Amer. Math. Soc.*, 23(2):535–561, 2010.
- J. Blasiok and J. Nelson. An improved analysis of the ER-SPUD dictionary learning algorithm. *43rd International Colloquium on Automata, Languages and Programming*, 2016.
- A. M. Bruckstein, D. L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Rev.*, 51(1):34–81, 2009.
- V. H. de la Peña and E. Giné. *Decoupling*. Probability and its Applications. Springer-Verlag, New York, 1999.
- P. Georgiev, F. Theis, and A. Cichocki. Sparse component analysis and blind source separation of underdetermined mixtures. *IEEE Transactions on Neural Networks*, 16(4), 2005.
- K. Kreutz-Delgado, J. Murray, B. Rao, K. Engan, T. Lee, and T. Sejnowski. Dictionary learning algorithms for sparse representation. *Neural Computation*, 15(20):349–396, 2003.
- M. Ledoux and M. Talagrand. *Probability in Banach Spaces*, volume 23 of *Ergebnisse der Mathematik und ihrer Grenzgebiete (3)*. Springer-Verlag, Berlin, 1991.
- K. Luh and V. Vu. Dictionary learning with few samples and matrix concentration. *IEEE Transactions on Information Theory*, 62(3):1516–1527, 2016.
- S. Mendelson. On weakly bounded empirical processes. *Math. Ann.*, 340(2):293–314, 2008.
- B. Olshausen and D. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6538):607–609, 1996.
- R. Rubinfeld, A. Bruckstein, and M. Elad. Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6):1045–1057, 2010.
- D. Spielman, H. Wang, and J. Wright. Exact recovery of sparsely-used dictionaries (long version). Preprint, 2012a. URL <http://www.columbia.edu/~jw2966/papers/SWW12-pp.pdf>.
- D. Spielman, H. Wang, and J. Wright. Exact recovery of sparsely-used dictionaries. *Journal of Machine Learning Research: Workshop and Conference Proceedings*, 23, 2012b. 25th Annual Conference on Learning Theory (COLT).
- J. Sun, Q. Qing, and J. Wright. Complete dictionary recovery over the sphere II: recovery by riemannian trust-region method. Preprint, 2015. URL <http://arxiv.org/abs/1511.04777>.
- M. Talagrand. *Upper and Lower Bounds for Stochastic Processes*. Springer, Heidelberg, 2014.
- A. W. van der Vaart and J. A. Wellner. *Weak Convergence and Empirical Processes*. Springer-Verlag, New York, 1996.
- J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Trans. Image Process.*, 19(11):2861–2873, 2010.
- M. Zibulevsky and B. Pearlmutter. Blind source separation by sparse decomposition. *Neural Computation*, 13(4), 2001.

The Asymptotic Performance of Linear Echo State Neural Networks

Romain Couillet

CentralesSupélec – LSS – Université ParisSud (Gif-sur-Yvette, France).

ROMAIN.COUILLET@CENTRALESUPELEC.FR

Gilles Wainrib

Département Informatique, team DATA, Ecole Normale Supérieure de Lyon (Lyon, France).

GILLES.WAINRIB@ENS.FR

Harry Sevi

Laboratoire de Physique, Ecole Normale Supérieure de Lyon (Lyon, France).

HARRY.SEVI@ENS-LYON.FR

Hafiz Tiomoko Ali

CentralesSupélec – LSS – Université ParisSud (Gif-sur-Yvette, France).

HAFIZ.TIOMOKALI@CENTRALESUPELEC.FR

Editor: Yoshua Bengio

Abstract

In this article, a study of the mean-square error (MSE) performance of linear echo-state neural networks is performed, both for training and testing tasks. Considering the realistic setting of noise present at the network nodes, we derive deterministic equivalents for the aforementioned MSE in the limit where the number of input data T and network size n both grow large. Specializing then the network connectivity matrix to specific random settings, we further obtain simple formulas that provide new insights on the performance of such networks.

Keywords: recurrent neural networks; echo state networks; random matrix theory; mean square error; linear networks

1. Introduction

Echo State Networks (ESN) are a class of recurrent neural networks (RNN) designed for performing supervised learning tasks, such as time-series prediction (Jaeger, 2001; Jaeger and Haas, 2004) or more generally any supervised learning task involving sequences. The ESN architecture is a special case of the general framework of reservoir computing (Lukoševičius and Jaeger, 2009). The ESN reservoir is a fixed (generally randomly designed) recurrent neural network, driven by a (usually time dependent) input. Since the internal connectivity matrix is not modified during learning, the number of parameters to learn is much smaller than in a classical RNN setting and the system is as such less prone to overfitting. However, the prediction performance of ESN often depends significantly on several hyper-parameters controlling the law of the internal connectivity matrix.

It has in particular been understood that the spectral radius and spectral norm of the connectivity matrix play a key role on the stability of the network (Jaeger, 2001) and that the structure of the connectivity matrix may be adapted to trade memory capacities versus task complexity (Verstraeten et al., 2010; Rodan and Tino, 2011; Strauss et al., 2012; Ozturk et al., 2007). Nonetheless, to date, and to the best of the authors' knowledge, the understanding of echo-state networks has progressed mostly through extensive empirical studies and lacks solid theoretical foundations.

In the present article, we consider linear ESN's with a general connectivity matrix and internal network noise. By leveraging tools from the field of random matrix theory, we shall attempt to provide a first *theoretical* study of the performance of ESN's. Beyond the obvious advantage of exploiting theoretical formulas to select the optimal hyper-parameters, this mathematical study reveals key quantities that intimately relate the internal network memory to the input-target relationship, therefore contributing to a better understanding of short-term memory properties of RNNs.

More specifically, we shall consider an n -node ESN trained with an input of size T and shall show that, assuming the internal noise variance η^2 remains large compared to $1/\sqrt{n}$, the training and testing performances of the network can be well approximated by a deterministic quantity which is a function of the training and test data as well as the connectivity matrix. Under the further assumption that the connectivity matrix is random, we shall then obtain closed-form formulas of the aforementioned performances for a large class of connectivity structures. The reach of our study so far only addresses ESN's with linear activation functions, a limitation which we anticipate to work around with more elaborate methods in the future, as discussed in Section 4.

At this point, we wish to highlight the specificity of our approach regarding (i) the introduction of noise perturbing the internal dynamics of the reservoir and (ii) the restriction to linear networks.

The introduction of additive noise in the reservoir is inspired on the one hand by it being a natural assumption in modelling biological neural networks (Ganguli et al., 2008; Toyozumi and Abbott, 2011) and on the other hand by the observation in (Jaeger, 2001) that ESN's are very sensitive to low variance noise and thus likely unstable in this regime, a problem successfully cured by internal noise addition (Jaeger, 2005).¹ From the neuro-computational perspective, we shall observe tight connections between the ESN performance and the reservoir information processing capacities discussed in (Ganguli et al., 2008). As for the artificial neural network viewpoint, it shall be noticed that the internal noise regularizes the network in a way sharing interesting similarities with the well-known connection between noise at the network output and Tikhonov regularization (Bishop, 1995). It is, as a matter of fact, already mentioned in (Lukoševičius and Jaeger, 2009, Section 8.2) that internal noise behaves as a natural regularization option (similar to what input or output noise would) although this aspect was not deeply investigated. More importantly, while internal noise necessarily leads to random outputs (a not necessarily desirable feature on the onset), we shall show that all these outputs (almost surely) asymptotically have the same performance, thus inducing random but equally useful innovation; this we believe is a more desirable feature than deterministic arbitrary biases in the innovation (see the comments around Remark 12 in Section 2.2).

As for the choice of studying linear activation functions, rarely considered in the practical side of RNNs, it obviously follows first from a mathematical tractability of the problem under study. Nonetheless, while being clearly a strong limitation of our study (recall that the non-linearity is the main driver for the network to perform complex tasks), we believe it brings sufficient insights (at least as far as memory capabilities and minimal performance are concerned) and exploitable results when it comes to parametrizing non-linear network counterparts.

Among other results, the main findings of our study are as follows:

1. According to Jaeger in (Jaeger, 2005) (specifically in the context of reservoirs with output feedback): "When you are training an ESN with output feedback from accurate (mathematical, noise-free) training data, stability of the trained network is often difficult to achieve. A method that works wonders is to inject noise into the dynamical reservoir update during sampling [...]. It is not clearly understood why this works."⁹

1. we retrieve a deterministic implicit expression for the mean-square error (MSE) performance of training and testing for any fixed connectivity matrix $W \in \mathbb{R}^{n \times n}$ which, for every given internal noise variance $\eta^2 > 0$, is all the more accurate that the network size n is large
2. the aforementioned expression reveals fundamental quantities which generalize several known qualitative notions of ESN's, such as the *memory capacity* and the *Fisher memory curve* (Jaeger, 2001; Ganguli et al., 2008);
3. we obtain more tractable closed-form expressions for the same quantities for simple classes of random normal and non-normal matrices: these two classes exhibit a strikingly different asymptotic performance behavior;
4. from the previous analysis, we shall also introduce a novel multi-modal connectivity matrix that adapts to a wider scope of memory ranges and that is reminiscent to the long short-term memory ESNs designed in (Xue et al., 2007);
5. an important interplay between memory and internal noise will be shed light on, by which the questions of noise-induced stability are better understood.

The remainder of the article is organized as follows. In Section 2, we introduce the ESN model and the associated supervised learning problem and we give our main theoretical results in Theorems 2 and 9 (technical proofs are deferred to the Appendix). Then, in Section 3, we apply our theoretical results for various choices of specific connectivity matrices and discuss their consequences in terms of prediction performance. Finally, in Section 4, we discuss our findings and their limitations.

Notations: In the remainder of the article, uppercase characters will stand for matrices, lowercase for scalars or vectors. The transpose operation will be denoted $(\cdot)^T$. The multivariate Gaussian distribution of mean μ and covariance C will be denoted $\mathcal{N}(\mu, C)$. The notation $V = \{V_{ij}\}_{i,j=1}^T$ denotes the matrix with (i, j) -entry V_{ij} (scalar or matrix), $1 \leq i \leq n$, $1 \leq j \leq T$, while $\{V_{ij}\}_{i=1}^n$ is the row-wise concatenation of the V_i 's and $\{V_{ij}\}_{j=1}^T$ the column-wise concatenation of the V_j 's. We further introduce the notation $(x)^+ \equiv \max(x, 0)$. For random or deterministic matrices X_n and $Y_n \in \mathbb{R}^{n \times n}$, the notation $X_n \leftrightarrow Y_n$ stands for $\frac{1}{n} \text{tr} A_n(X_n - Y_n) \rightarrow 0$ and $a_n(X_n - Y_n) b_n \rightarrow 0$, almost surely, for every deterministic matrix A_n and vectors a_n, b_n having bounded norm (spectral norm for matrices and Euclidean norm for vectors); for $X_n, Y_n \in \mathbb{R}$ scalar, the notation will simply mean that $X_n - Y_n \rightarrow 0$ almost surely. The notation $\rho(X)$ will denote the spectral radius of matrix X , while $\|X\|$ will denote its operator norm (and for vectors, $\|x\|$ is the Euclidean norm). The symbol δ_x shall stand for Kronecker's delta function, i.e., $\delta_x(y) = 1$ if $y = x$ (or x is true) and zero otherwise.

2. Main Results

We consider here an echo-state neural network constituted of n nodes, with state $x_t \in \mathbb{R}^n$ at time instant t , connectivity matrix $W \neq 0$, and input source sequence $\dots, u_{-1}, u_0, u_1, \dots \in \mathbb{R}$. The state evolution is given by the linear recurrent equation

$$x_{t+1} = Wx_t + mu_{t+1} + \eta \epsilon_{t+1}$$

for all $t \in \mathbb{Z}$, in which $\eta > 0$ and $\epsilon_t \sim \mathcal{N}(0, I_n)$, while $m \in \mathbb{R}^n$ is the input-to-network connectivity.

Our first objective is to understand the training performance of such a network. To this end, we shall focus on a (training) time window $\{0, \dots, T-1\}$ and will denote $X =$

$\{x_j\}_{j=0}^{T-1} \in \mathbb{R}^{n \times T}$ as well as $A = MU$, $M \in \mathbb{R}^{n \times T}$, $U \in \mathbb{R}^{T \times T}$, where

$$\begin{aligned} M &\equiv \{W^j m\}_{j=0}^{T-1} \\ U &\equiv \frac{1}{\sqrt{T}} \{u_j - \bar{u}\}_{i,j=1}^T. \end{aligned}$$

With these notations, we especially have $X = \sqrt{T}(A+Z)$, where $Z = \frac{1}{\sqrt{T}} \{\sum_{k=0}^{\infty} W^k \epsilon_{j-k}\}_{j=0}^{T-1}$. The matrix A can be seen here as the matrix carrying the information about the input vector u while Z serves the purpose of regularization noise.

For X to be properly defined (at least almost surely so), we shall impose the following hypothesis.

Assumption 1 (Spectral Norm) *The spectral norm $\|W\|$ of W satisfies $\|W\| < 1$.*

Note that this constraint is in general quite strong and it is believed (following the insights of previous works (Jaeger, 2001)) that for many model choices of W , it can be lightened to merely requiring that the spectral radius $\rho(W)$ be smaller than one. Nonetheless, in the course of the article, we shall often take W to be such that both its spectral norm and spectral radius coincide.

2.1. Training Performance

The training step consists in teaching the network to obtain a specific output sequence $r = \{r_j\}_{j=0}^{T-1}$ out of the network, when fed by a corresponding input vector $u = \{u_j\}_{j=0}^{T-1}$ over the time window. To this end, unlike conventional neural networks, where W is adapted to u and r , ESN's adopt the strategy to solely enforce an output link from the network to a sink (or readout). Letting $\omega = \{\omega_i\}_{i=1}^n$ be the network-to-sink connectivity vector, we shall consider here that ω is obtained as the (least-square) minimizer of $\|X^T \omega - r\|^2$. When $T > n$, we have

$$\omega \equiv (X X^T)^{-1} X r \quad (1)$$

which is almost surely well-defined (since $\eta > 0$) or, when $T \leq n$,

$$\omega \equiv X (X^T X)^{-1} r. \quad (2)$$

The per-input mean-square error in training associated with the couple (u, r) for the ESN under study is then defined as

$$E_\eta(u, r) \equiv \frac{1}{T} \|r - X^T \omega\|^2 \quad (3)$$

which is identically zero when $T \leq n$.

Our first objective is to study precisely the random quantity $E_\eta(u, r)$ for every given W and noise variance η^2 in the limit where $n \rightarrow \infty$. Our scaling hypotheses are as follows.

Assumption 2 (Random Matrix Regime) *The following conditions hold:*

1. $\limsup_n \eta/T < \infty$
2. $\limsup_n \|AA^T\| < \infty$.

That is, according to Item 1, we allow n to grow with T . Also, from Item 2, we essentially allow u_t to be of order $O(1)$ (unless u is sparse and then u_t may be as large as $O(\sqrt{T})$) when m remains of bounded Euclidean norm. Under this setting, and along with Assumption 1, we shall thus essentially require all neural connections to be of order $O(n^{-\frac{1}{2}})$ while all input and output data (constituents of u and r) shall be in general of order $O(1)$.

For every square symmetric matrix $B \in \mathbb{R}^{n \times n}$, a central quantity in random matrix theory is the resolvent $(B - zI_n)^{-1}$ defined for every $z \in \mathbb{C} \setminus \mathcal{S}_B$, with $\mathcal{S}_B \subset \mathbb{R}$ the support of the eigenvalues of B . Here, letting $z = -\gamma$ for some $\gamma > 0$, it is particularly convenient to make the following observation.

Lemma 1 (Training MSE and resolvent) For $\gamma > 0$, let $\bar{Q}_\gamma \equiv (\frac{1}{T}X^T X + \gamma I_T)^{-1}$. Then we have, for $E_\gamma(u, r)$ defined as in (3),

$$E_\gamma(u, r) = \lim_{\gamma \downarrow 0} \frac{1}{T} r^T \bar{Q}_\gamma r.$$

Our first technical result provides an asymptotically tight approximation for \bar{Q}_γ for every $\gamma > 0$. Recall that, for $X_n, Y_n \in \mathbb{R}^{n \times n}$, the notation $X_n \leftrightarrow Y_n$ means that, for every deterministic and bounded norm matrix A_n or vector $a_n, b_n, \frac{1}{n} \text{tr} A_n(X_n - Y_n) \rightarrow 0$ and $a_n^T(X_n - Y_n)b_n \rightarrow 0$, almost surely.

Theorem 2 (Deterministic Equivalent) Let Assumptions 1–2 hold. For $\gamma > 0$, let also $Q_\gamma \equiv (\frac{1}{T}X X^T + \gamma I_n)^{-1}$ and $\hat{Q}_\gamma \equiv (\frac{1}{T}X^T X + \gamma I_T)^{-1}$. Then, as $n \rightarrow \infty$, the following approximations hold:

$$\begin{aligned} Q_\gamma \leftrightarrow \hat{Q}_\gamma &\equiv \frac{1}{\gamma} \left(I_n + \eta^2 \bar{R}_\gamma + \frac{1}{\gamma} A (I_T + \eta^2 R_\gamma)^{-1} A^T \right)^{-1} \\ \hat{Q}_\gamma \leftrightarrow \bar{Q}_\gamma &\equiv \frac{1}{\gamma} \left(I_T + \eta^2 R_\gamma + \frac{1}{\gamma} A^T (I_n + \eta^2 \bar{R}_\gamma)^{-1} A \right)^{-1} \end{aligned}$$

where $R_\gamma \in \mathbb{R}^{T \times T}$ and $\bar{R}_\gamma \in \mathbb{R}^{n \times n}$ are solutions to the system of equations

$$\begin{aligned} R_\gamma &= \left\{ \frac{1}{T} \text{tr} (S_{t-j} \bar{Q}_\gamma) \right\}_{i,j=1}^T \\ \bar{R}_\gamma &= \sum_{q=-\infty}^{\infty} \frac{1}{T} \text{tr} (J^q \bar{Q}_\gamma) S_q \end{aligned}$$

with $[J^q]_{ij} \equiv \delta_{i+q,j}$ and $S_q \equiv \sum_{k \geq 0} W^{k+(-q)^+} (W^{k+q^+})^T$.

Remark 3 (On Theorem 2) Theorem 2 is in fact valid under more general assumptions than in the present setting. In particular, A may be any deterministic matrix satisfying Assumption 2. However, when $A = MU$, an important phenomenon arises, which is that A behaves similar to a low-rank matrix, since, by Assumption 1, only $o(n)$ columns of M have non-vanishing norm. As such, by a low-rank perturbation argument, it can be shown that the term \bar{Q}_γ in the expression of R_γ and the term \hat{Q}_γ in the expression of \bar{R}_γ can be replaced by $\gamma^{-1} (I_n + \eta^2 \bar{R}_\gamma)^{-1}$ and $\gamma^{-1} (I_T + \eta^2 R_\gamma)^{-1}$, respectively. As such, R_γ and \bar{R}_γ only depend on the matrix W and the parameter η^2 , and are thus asymptotically independent of the input data matrix U . Note also in passing that, while \bar{R}_γ is defined with a sum over $q = -\infty$ to ∞ , this summation is empty for all $|q| \geq T$.

2. Note that $\text{tr}(J^q B)$ is merely $\text{tr}(J^q B) = \sum_{i=1+q^+}^{T-q^+} [B]_{i+i|q}$.

In order to evaluate the training mean-square error $E_\gamma(u, r)$ from Lemma 1, one must extend Theorem 2 uniformly over γ approaching zero. This can be guaranteed under the following additional assumption.

Assumption 3 (Network size versus training time) As $n \rightarrow \infty$, $n/T \rightarrow c \in [0, 1) \cup (1, \infty)$.

Under Assumption 3, two scenarios must be considered. Either $c < 1$ or $c > 1$. In the former case, we can show that, as $\gamma \downarrow 0$, R_γ and $\gamma \bar{R}_\gamma$ have well defined limits. Besides, it appears that the limit of $\eta^2 R_\gamma$ does not depend on η^2 , so that we shall denote \mathcal{R} and $\bar{\mathcal{R}}$ the limits of $\eta^2 R_\gamma$ and $\gamma \bar{R}_\gamma$, as $\gamma \downarrow 0$, respectively. Similarly, $\eta^2 \bar{Q}_\gamma$ and $\gamma \bar{Q}_\gamma$ converge to well defined limits, denoted respectively \mathcal{Q} and $\bar{\mathcal{Q}}$. Symmetrically, for $c > 1$, as $\gamma \downarrow 0$, γR_γ and $\eta^2 \bar{R}_\gamma$ have well-behaved limits which we shall also refer to as \mathcal{R} and $\bar{\mathcal{R}}$; similarly, γQ_γ and $\eta^2 \bar{Q}_\gamma$ converge to non trivial limits again denoted \mathcal{Q} and $\bar{\mathcal{Q}}$. These results are gathered in the following proposition.

Proposition 4 (Small γ limit of Theorem 2) Let Assumptions 1–3 hold. For all large n , define \mathcal{R} and $\bar{\mathcal{R}}$ a pair of solutions of the system

$$\begin{aligned} \mathcal{R} &= c \left\{ \frac{1}{n} \text{tr} \left(S_{t-j} \left(\delta_{c>1} I_n + \bar{\mathcal{R}} \right)^{-1} \right) \right\}_{i,j=1}^T \\ \bar{\mathcal{R}} &= \sum_{q=-\infty}^{\infty} \frac{1}{T} \text{tr} (J^q (\delta_{c<1} I_T + \mathcal{R})^{-1}) S_q. \end{aligned}$$

Subsequently define

$$\begin{aligned} \bar{\mathcal{Q}} &\equiv \left(\delta_{c<1} I_T + \mathcal{R} + \frac{1}{\eta^2} A^T \left(\delta_{c>1} I_n + \bar{\mathcal{R}} \right)^{-1} A \right)^{-1} \\ \mathcal{Q} &\equiv \left(\delta_{c>1} I_n + \bar{\mathcal{R}} + \frac{1}{\eta^2} A \left(\delta_{c<1} I_T + \mathcal{R} \right)^{-1} A^T \right)^{-1}. \end{aligned}$$

Then, with the definitions of Theorem 2, we have the following results.

1. If $c < 1$, then in the limit $\gamma \downarrow 0$, $\eta^2 R_\gamma \rightarrow \mathcal{R}$, $\gamma \bar{R}_\gamma \rightarrow \bar{\mathcal{R}}$, $\eta^2 \bar{Q}_\gamma \rightarrow \bar{\mathcal{Q}}$, and $\gamma \bar{Q}_\gamma \rightarrow \bar{\mathcal{Q}}$.
2. If $c > 1$, then in the limit $\gamma \downarrow 0$, $\gamma R_\gamma \rightarrow \mathcal{R}$, $\eta^2 \bar{R}_\gamma \rightarrow \bar{\mathcal{R}}$, $\gamma \bar{Q}_\gamma \rightarrow \bar{\mathcal{Q}}$, and $\eta^2 \bar{Q}_\gamma \rightarrow \bar{\mathcal{Q}}$.

With these notations, we now have the following result.

Corollary 5 (Training MSE for $n < T$) Let Assumptions 1–3 hold and let $r \in \mathbb{R}^T$ be of $O(\sqrt{T})$ Euclidean norm. Then, with $E_\gamma(u, r)$ defined in (3), as $n \rightarrow \infty$,

$$E_\gamma(u, r) \leftrightarrow \begin{cases} (1/T)^T \bar{\mathcal{Q}}^r & , c < 1 \\ 0 & , c > 1. \end{cases}$$

It is interesting at this point to discuss the a priori involved expression of Proposition 4 and Corollary 5. Let us concentrate on the interesting $c < 1$ case. To start with, observe that \mathcal{R} and $\bar{\mathcal{R}}$ are deterministic matrices which only depend on W through the S_q matrices so that the only dependence of $E_\gamma(u, r)$ on the noise variance η^2 lies explicitly in the expression of $\bar{\mathcal{Q}}$. Now, making $A^T \bar{\mathcal{R}}^{-1} A$ explicit, we have the following telling limiting expression for $E_\gamma(u, r)$

$$E_\gamma(u, r) \leftrightarrow \frac{1}{T} r^T \left(I_T + \mathcal{R} + \frac{1}{\eta^2} U^T \left\{ m^T (W^t)^T \bar{\mathcal{R}}^{-1} W^t m \right\}_{i,j=0}^{T-1} U \right)^{-1} r. \quad (4)$$

Recalling that $\tilde{\mathcal{R}}$ is a linear combination of the matrices $S_q = W^{(-q)} S_0 W^{(q)}$, with $S_0 = \sum_{k \geq 0} W^k (W^k)^T$, the expression $\frac{1}{\eta} m^T (W^i)^T \tilde{\mathcal{R}}^{-1} W^i m$ is strongly reminiscent of the Fisher memory curve $f : \mathbb{N} \rightarrow \mathbb{R}$ of the ESN, introduced in (Ganguli et al., 2008) and defined by $f(k) = \frac{1}{\eta} m^T (W^k)^T S_0^{-1} W^k m$. The Fisher memory curve $f(k)$ qualifies the ability of a k -step behind past input to influence the ESN at present time. Correspondingly, it appears here that the ability of the ESN to retrieve the desired expression of r from input u is importantly related to the matrix $\{\frac{1}{\eta} m^T (W^i)^T \tilde{\mathcal{R}}^{-1} W^i m\}_{i,j=0}^{T-1}$. As a matter of fact, for $c = 0$ (thus for a long training period), note that $\mathcal{R} = 0$ while $\tilde{\mathcal{R}} = S_0$ and we then find in particular

$$E_{\eta}(u, r) \leftrightarrow \frac{1}{\tau} r^T \left(I_T + \frac{1}{\eta^2} U^T \{m^T (W^i)^T S_0^{-1} W^i m\}_{i,j=0}^{T-1} U \right)^{-1} r.$$

Pushing further our discussion on \mathcal{R} and $\tilde{\mathcal{R}}$, it is interesting to intuit their respective structures. Observe in particular that \mathcal{R}_{ij} depends only on $i - j$ and thus \mathcal{R} is a Toeplitz matrix. Besides, since $\text{tr } B = \text{tr } B^T$ for square matrices B , from $S_{i-j}^T = S_{j-i}$ it comes that $\mathcal{R}_{ij} = \mathcal{R}_{ji}$. Also note that, since $\rho(W^q) = \rho(W)^q$ decays exponentially as $q \rightarrow \infty$, it is expected that $\mathcal{R}_{i,i+q}$ decays exponentially fast for large q . As a consequence, \mathcal{R} is merely defined by $o(n)$ first entries of its first row.

From the results of (Gray, 2006) on Toeplitz versus circulant matrices, it then appears that, for every deterministic matrix B , $\frac{1}{\tau} \text{tr } B \mathcal{R}^{-1}$ is well approximated by $\frac{1}{\tau} \text{tr } B \mathcal{R}_c^{-1}$ for \mathcal{R}_c a circulant matrix approximation of \mathcal{R} . Since circulant matrices are diagonalizable in a Fourier basis, so are their inverses and then, as far as normalized traces are concerned, $(I_T + \mathcal{R})^{-1}$ can be seen as approximately Toeplitz with again decaying behavior away from the main diagonals. Although slightly ambiguous, this approximation still makes it that the trace $\frac{1}{\tau} \text{tr } \eta^2 (I_T + \mathcal{R})^{-1}$ appearing in the expression of $\tilde{\mathcal{R}}$, is well approximated by any value $\lfloor (I_T + \mathcal{R})^{-1} \rfloor_{i,i+q}$ for i sufficiently far from 1 and T , and decays to zero as q grows large. This, and the fact that S_q also decays exponentially fast in norm allows us to conclude that \mathcal{R} can be seen as a decaying weighted sum of $o(n)$ matrices S_{q_r} .

As shall be shown in Section 3, for W taken random with sufficient invariance properties, fundamental differences appear in the structure of \mathcal{R} and $\tilde{\mathcal{R}}$ depending on whether W is taken normal or not. In particular, for W non-normal with left and right independent isotropic eigenvectors and m deterministic or random independent of W , \mathcal{R} is well approximated by a scaled identity matrix and $\{m^T (W^i)^T \tilde{\mathcal{R}}^{-1} W^i m\}_{i,j=0}^{T-1}$ well approximated by a diagonal matrix with exponential decay along the diagonal.

Having a clearer understanding of Corollary 5, a few key remarks are in order.

Remark 6 (On the ESN stability to low noise levels) It is easily seen by differentiation along η^2 that $r^T \mathcal{Q} r$ is an increasing function of η^2 , thus having a minimum as $\eta^2 \downarrow 0$. It is thus tempting to suppose that $E_{\eta}(u, r)$ converges to this limit in the noiseless case (*i.e.*, for $\eta^2 = 0$). Such a reasoning is however hazardous and incorrect in most cases. Indeed, Corollary 5 only ensures an appropriate approximation of $E_{\eta}(u, r)$ for given $\eta > 0$ in the limit where $n \rightarrow \infty$. Classical random matrix considerations allow one to assert slightly stronger results. In particular, for the approximation of $E_{\eta}(u, r)$ to hold, one may allow η^2 to depend on n in such a way that $\eta^2 \gg n^{-\frac{1}{2}}$. This indicates that n must be quite large for the ESN behavior at moderate noise levels to be understood through the random matrix method. What seems like a defect of the tool on the onset in fact sheds some light on a deeper feature of ESN's. When η^2 is of the same order of magnitude or smaller than $n^{-\frac{1}{2}}$, Corollary 5 may become invalid due to the resurgence of randomness from $\dots, \epsilon_{-1}, \epsilon_0, \epsilon_1, \dots$. Precisely, when η^2 gets small and thus the training MSE variance should decay, an opposite effect makes

the MSE more random and thus possibly no longer tractable; this means in particular that, for any two independent runs of the ESN (with different noise realizations), all other parameters being fixed, the resulting MSE's might be strikingly different, making the network quite unstable. In practice, the opposition of the reduced noise variance η^2 and the resurgence of noise effects lead to various behaviors depending on the task and input data under considerations, ranging from largely increased MSE fluctuations at low η^2 to reduced fluctuations, through stabilisation of the fluctuations. In some specific cases discussed later, it might nonetheless be accepted to let $\eta^2 \rightarrow 0$ irrespective of n while keeping the random matrix approximation valid.

Remark 7 (Memory capacity revisited) For $c < 1$, letting $u_k = \sqrt{T} \delta_k$ and $r_k = \sqrt{T} \delta_{k-\tau}$ (that is, all input energy is gathered in a single entry), for some $t, \tau \in \mathbb{N}$, makes the ESN fill a pure delay task of τ time-steps. In this case, we find that

$$E_{\eta}(u, r) \leftrightarrow \left[\left(I_T + \mathcal{R} + \frac{1}{\eta^2} \{m^T (W^i)^T \tilde{\mathcal{R}}^{-1} W^i m\}_{i,j=0}^{T-1} \right)^{-1} \right]_{t+\tau, t+\tau}^{-1}.$$

In the particular case where, for all $i \neq j$, $\mathcal{R}_{ij} = o(1)$ and $m^T (W^i)^T \tilde{\mathcal{R}}^{-1} W^i m = o(1)$ (see Section 3.1 for a practical application with random non-normal W), by a uniform control argument due to the fast decaying for off-diagonal elements of \mathcal{R} and $\{m^T (W^i)^T \tilde{\mathcal{R}}^{-1} W^i m\}$, the training MSE is further (almost surely) well approximated as

$$E_{\eta}(u, r) \leftrightarrow \frac{\eta^2}{\eta^2(1 + \mathcal{R}_{11}) + m^T (W^t)^T \tilde{\mathcal{R}}^{-1} W^t m}.$$

If the quantity $m^T (W^t)^T \tilde{\mathcal{R}}^{-1} W^t m$ remains away from zero as $n \rightarrow \infty$, then it is allowed here to say (as opposed to the general case discussed in Remark 6) that $E_{\eta}(u, r) \rightarrow 0$ as $\eta \rightarrow 0$ and that $\eta^2/E_{\eta}(u, r) \sim m^T (W^t)^T \tilde{\mathcal{R}}^{-1} W^t m$, where we recover again a generalized form of the Fisher information curve at delay τ . From this discussion and Remark 6, we propose to define a novel network memory capacity metric $\text{MC}(\tau)$, representing the inverse slope of decay of $E_{\eta}(\sqrt{T} \delta_t, \sqrt{T} \delta_{t-\tau})$ for small η^2 :

$$\text{MC}(\tau) \equiv \lim_{\eta \downarrow 0} \liminf_{\tau} \left[\left(\eta^2 (I_T + \mathcal{R}) + \{m^T (W^i)^T \tilde{\mathcal{R}}^{-1} W^i m\}_{i,j=0}^{T-1} \right)^{-1} \right]_{\tau+1, \tau+1}^{-1}.$$

Remark 7 follows up on recent works, here from an MSE performance perspective, that establish links between memory capacity metrics and the Fisher memory curve, as in e.g., (Tño and Rodan, 2013). Practical applications of Corollary 5 to specific matrix models for W shall be derived in Section 3. Beforehand, we will study the more involved question of the test MSE performance.

2.2 Test Performance

In this section, we assume $\omega \equiv \omega(X; u, r)$ has been obtained as per (1) or (2), depending on whether $c < 1$ or $c > 1$. We now consider the test performance of the ESN that corresponds to its ability to map an input vector $\hat{u} \in \mathbb{R}^T$ to an expected output vector $\hat{r} \in \mathbb{R}^T$ of duration T in such a way to fulfill the same task that links u to r . For notational convenience, all test data will be denoted with a hat mark on top.

As opposed to the training mean square error, the testing MSE, defined as

$$\hat{E}_{\eta}(u, r; \hat{u}, \hat{r}) \equiv \frac{1}{T} \|\hat{r} - \hat{X}^T \omega\|^2 \quad (5)$$

where $\hat{X} = \{\hat{x}_t\}_{t=0}^{T-1} \in \mathbb{R}^{n \times T}$ is defined by the recurrent equation $\hat{x}_{t+1} = W\hat{x}_t + m\hat{u}_{t+1} + \eta\hat{\varepsilon}_{t+1}$, with $\hat{\varepsilon}_t \sim \mathcal{N}(0, I_n)$ independent of the ε_t 's, does not assume a similar simple form as the training MSE. We importantly assume here that a sufficiently long washout period between training and testing is present in the sense that \hat{x}_0 is assumed independent of the x_t described in the previous section (see Remark 22 in Appendix A for a discussion on the results generalization when no washout period is assumed). Under these assumptions, we merely have the following result.

Lemma 8 (Testing MSE) For $\gamma > 0$, $Q_\gamma = (\frac{1}{T}XX^T + \gamma I_n)^{-1}$, and $\hat{Q}_\gamma = (\frac{1}{T}X^T X + \gamma I_T)^{-1}$, we have

$$\begin{aligned} \hat{E}_\eta(u, r; \hat{u}, \hat{r}) &= \lim_{T \rightarrow \infty} \frac{1}{T} \|\hat{r}\|^2 + \frac{1}{T^2 T} r^T X^T Q_\gamma \hat{X} \hat{X}^T Q_\gamma X r - \frac{2}{T T} r^T \hat{X}^T Q_\gamma X r \\ &= \lim_{T \rightarrow \infty} \frac{1}{T} \|\hat{r}\|^2 + \frac{1}{T^2 T} r^T \hat{Q}_\gamma X^T \hat{X} \hat{X}^T X \hat{Q}_\gamma r - \frac{2}{T T} r^T \hat{X}^T X \hat{Q}_\gamma r \end{aligned}$$

with $\hat{E}_\eta(u, r; \hat{u}, \hat{r})$ defined in (5).

If $n < T$, Q_γ is well-defined in the limit $\gamma \downarrow 0$, while if instead $n \geq T$, then one may observe that $X^T Q_\gamma = \hat{Q}_\gamma X^T$ with \hat{Q}_γ having well defined limit as $\gamma \downarrow 0$.

Technically, estimating \hat{E} requires to retrieve, in a similar fashion as for Theorem 2, a deterministic approximation of quantities of the type $Q_\gamma X$ and $X^T Q_\gamma B Q_\gamma X = \hat{Q}_\gamma X^T B X \hat{Q}_\gamma$ for B a matrix independent of X . We precisely obtain the following result.

Theorem 9 (Second order deterministic equivalent) Let Assumptions 1–2 hold and let $B \in \mathbb{R}^{n \times n}$ be a deterministic symmetric matrix of bounded spectral norm. Then, recalling the notations of Theorem 2, for every $\gamma > 0$,

$$\begin{aligned} Q_\gamma \frac{1}{\sqrt{T}} X \leftrightarrow Q_\gamma A (I_n + \eta^2 R_\gamma)^{-1} \\ \frac{1}{T} X^T Q_\gamma B Q_\gamma X \leftrightarrow \eta^2 \gamma^2 \hat{Q}_\gamma C^{[B]} \hat{Q}_\gamma + (I_n + \eta^2 R_\gamma)^{-1} A^T \hat{Q}_\gamma [B + \hat{G}_\gamma^{[B]}] \hat{Q}_\gamma A (I_n + \eta^2 R_\gamma)^{-1} \end{aligned}$$

where $G_\gamma^{[B]} \in \mathbb{R}^{T \times T}$ and $\hat{G}_\gamma^{[B]} \in \mathbb{R}^{n \times n}$ are solutions to the system of equations

$$\begin{aligned} G_\gamma^{[B]} &= \left\{ \frac{1}{T} \text{tr} \left(S_{i-j} \hat{Q}_\gamma [B + \hat{G}_\gamma^{[B]}] \hat{Q}_\gamma \right) \right\}_{i,j=1}^T \\ \hat{G}_\gamma^{[B]} &= \sum_{q=-\infty}^{\infty} \eta \gamma^2 \frac{1}{T} \text{tr} \left(J^q \hat{Q}_\gamma C_\gamma^{[B]} \hat{Q}_\gamma \right) S_q. \end{aligned}$$

With these results at hand, we may then determine limiting approximations of the test mean-square error under both $n < T$ and $n > T$ regimes. As in Section 2.1, one may observe here that, under Assumption 3 with, say $c < 1$, $\eta^4 G_\gamma^{[B]}$ and $\hat{G}_\gamma^{[B]}$ both have well defined limits as $\gamma \downarrow 0$ which we shall subsequently refer to as $\mathcal{G}^{[B]}$ and $\hat{\mathcal{G}}^{[B]}$, respectively, and the symmetrical result holds for $c > 1$. Precisely, we have the following result.

Proposition 10 (Small γ limit of Theorem 9) Let Assumptions 1–3 hold and let $B \in \mathbb{R}^{n \times n}$ be a deterministic symmetric matrix of bounded spectral norm. For all large n , define $\mathcal{G}^{[B]}$ and $\hat{\mathcal{G}}^{[B]}$ a pair of solutions of the system

$$\begin{aligned} \mathcal{G}^{[B]} &= c \left\{ \frac{1}{n} \text{tr} \left(S_{i-j} (\delta_{>1} I_n + \hat{\mathcal{R}})^{-1} [B + \hat{\mathcal{G}}^{[B]}] (\delta_{>1} I_n + \hat{\mathcal{R}})^{-1} \right) \right\}_{i,j=1}^T \\ \hat{\mathcal{G}}^{[B]} &= \sum_{q=-\infty}^{\infty} \frac{1}{T} \text{tr} \left(J^q (\delta_{<1} I_T + \mathcal{R})^{-1} \mathcal{G}^{[B]} (\delta_{<1} I_T + \mathcal{R})^{-1} \right) S_q. \end{aligned}$$

Then, with the definitions of Theorem 9, we have the following results.

1. If $c < 1$, then in the limit $\gamma \downarrow 0$, $\eta^4 G_\gamma^{[B]} \rightarrow \mathcal{G}^{[B]}$ and $\hat{G}_\gamma^{[B]} \rightarrow \hat{\mathcal{G}}^{[B]}$.
2. If $c > 1$, then in the limit $\gamma \downarrow 0$, $\gamma^2 G_\gamma^{[B]} \rightarrow \mathcal{G}^{[B]}$ and $\hat{G}_\gamma^{[B]} \rightarrow \hat{\mathcal{G}}^{[B]}$.

Proposition 10 will be exploited on the deterministic matrix $\frac{1}{T} E[\hat{X} \hat{X}^T] = \eta^2 S_0 + \hat{A} \hat{A}^T$. Rather than taking $B = \eta^2 S_0 + \hat{A} \hat{A}^T$, which would induce an implicit dependence of $\mathcal{G}^{[B]}$ and $\hat{\mathcal{G}}^{[B]}$ on η^2 , we shall instead split $\eta^2 S_0 + \hat{A} \hat{A}^T$ into η^2 times S_0 and $\hat{A} \hat{A}^T$. Noticing then that $\mathcal{G}^{[A \hat{A}^T]}$ is asymptotically the same as $\mathcal{G}^{[0]}$, with 0 the all zero matrix, we may then obtain an approximation for the test mean square error. Prior to this, we need the following growth control assumptions.

Assumption 4 (Random Matrix Regime for Test Data) The following conditions hold:

1. $\limsup_n n/T < \infty$
2. $\limsup_n \|\hat{A} \hat{A}^T\| < \infty$.

Note in passing here that the $\min(T, \hat{T})$ first columns of $\hat{M} \in \mathbb{R}^{n \times \hat{T}}$ in the definition of \hat{A} and $M \in \mathbb{R}^{n \times T}$ in the definition of A are identical. As such, only \hat{U} actually particularizes the data matrix \hat{A} .

With this condition, we have the following corollary of Theorem 9.

Corollary 11 (Test MSE) Let Assumptions 1–4 hold and let $\hat{r} \in \mathbb{R}^{\hat{T}}$ be a vector of Euclidean norm $O(\sqrt{\hat{T}})$. Then, as $n \rightarrow \infty$, both for $c < 1$ and $c > 1$, we have, with the notations of Propositions 4–10,

$$\begin{aligned} \hat{E}_\eta(u, r; \hat{u}, \hat{r}) \leftrightarrow & \left\| \frac{1}{\eta^2 \sqrt{T}} A^T Q_A (\delta_{<1} I_T + \mathcal{R})^{-1} r - \frac{1}{\sqrt{T}} \hat{r} \right\|^2 + \frac{1}{T} r^T \hat{\mathcal{G}} \hat{\mathcal{G}} \hat{Q} r \\ & + \frac{1}{\eta^2 T} r^T (\delta_{<1} I_T + \mathcal{R})^{-1} A^T \mathcal{Q} [S_0 + \hat{\mathcal{G}}] Q_A (\delta_{<1} I_T + \mathcal{R})^{-1} r \end{aligned} \quad (6)$$

where $\mathcal{G} \equiv \mathcal{G}^{[S_0]}$ and $\hat{\mathcal{G}} \equiv \hat{\mathcal{G}}^{[S_0]}$.

The form of Corollary 11 is more involved than that of Corollary 5 but is nonetheless quite interpretable. To start with, observe that \mathcal{G} and $\hat{\mathcal{G}}$ are again only function of W and therefore quantify the network connectivity only. Then, note that only the first right-hand side term of the approximation of $E_\eta(u, r; \hat{u}, \hat{r})$ depends on \hat{u} and \hat{r} . As such, the quality of the learned task relies mostly on this term.

If $c = 0$, for all B , $\mathcal{G}^{[B]} = 0$ and $\hat{\mathcal{G}}^{[B]} = 0$, so we have here the simplified expression

$$\hat{E}_\eta(u, r; \hat{u}, \hat{r}) \leftrightarrow \left\| \frac{1}{\sqrt{T}} A^T (\eta^2 S_0 + A A^T)^{-1} A r - \frac{1}{\sqrt{T}} r \right\|^2 + \frac{1}{T} r^T A^T (\eta^2 S_0 + A A^T)^{-2} A r.$$

Some remarks are in order to appreciate these results.

Remark 12 (Noiseless case) As a follow-up on Remark 6, note that some alternative approaches to ESN normalization assume instead that $\eta = 0$ but that ω is taken to be the regularized least-square (or ridge-regression) estimator $\omega = X(X^T X + \gamma I_T)^{-1} r$ with $\gamma > 0$. In this case, it is easily seen that the corresponding mean-square error performance in training is given by $E^T(u, r) \equiv \gamma \frac{1}{T} r^T \hat{Q}_\gamma^T r$, which is precisely

$$E^T(u, r) = \frac{1}{T} r^T \left(I_T + \frac{1}{\gamma} U^T \{ m^T (W^T W^T m) \}_{i,j=0}^{T-1} U \right)^{-2} r.$$

It is interesting to parallel this (exact) expression to the approximation (4) in which the noise variance η^2 plays the role of the regularization γ , but (i) where the two additional quantities \mathcal{R} and \mathcal{R} are present, and (ii) where the power factor of the matrix inverse is 1 in place of 2. As for the testing performance, we are here comparing Corollary 11 to the noiseless regularized MSE

$$E^r(u, r; \hat{u}, \hat{r}) = \left\| \frac{1}{\sqrt{T}} \hat{A}^T (\gamma I_n + A A^T)^{-1} A r - \frac{1}{\sqrt{T}} \hat{r} \right\|^2.$$

This is again easily paralleled with the first right-hand side term in (6) which, for say $c < 1$, reads

$$\left\| \frac{1}{\sqrt{T}} \hat{A}^T \left(\eta^2 \tilde{\mathcal{R}} + A(I_T + \mathcal{R})^{-1} A^T \right)^{-1} A(I_T + \mathcal{R})^{-1} r - \frac{1}{\sqrt{T}} \hat{r} \right\|^2.$$

Again, it is clear that η^2 plays a similar role as that of γ , and that the matrices \mathcal{R} and $\tilde{\mathcal{R}}$ capture the behavior of the in-network noise.

Remark 12 suggests that internal noise plays a similar role to ridge normalization and that both lead to similar MSE performances. This being said, both regularizations behave strikingly differently in practice. While ridge-regularization provides a deterministic network output for given input vector u , internal noise instead induces random independent outputs for any two feeds of the network by the same vector u . Since all such random outputs have similar MSE performance (for sufficiently large network sizes), this may be a preferable choice in practice to avoid deterministic arbitrary mappings.

3. Applications

In this section, we shall further estimate the results of Corollary 5 and Corollary 11 in specific settings for the network connectivity matrix W and the input weights m . By leveraging specific properties of certain stochastic models for W (such as invariance by orthogonal matrix product or by normality), the results of Section 2 will be greatly simplified, by then providing further insights on the network performance.

3.1 Bi-orthogonally invariant W

We first consider the scenario where W is random with distribution invariant to left- and right-multiplication by orthogonal matrices, which we refer to as *bi-orthogonal invariance*. Precisely, in singular-value decomposition form, we shall write $W = UGV^T$, where U , V , and Ω are independent and U , V are real Haar distributed (that is, orthogonal with bi-orthogonally invariant distribution) and shall impose that the eigenvalues of W remain bounded by $\sigma < 1$ for all large n . Two classical examples of such a scenario are (i) W is itself a scaled Haar matrix, in which case $\Omega = \sqrt{\sigma} I_n$, and the eigenvalues of W all have modulus σ , or (ii) W has independent $\mathcal{N}(0, \sigma^2)$ entries, in which case, according to standard random matrix results, for any $\varepsilon > 0$, the eigenvalues of W have modulus less than $\sigma + \varepsilon$ for all large n almost surely and W is clearly orthogonally invariant by orthogonal invariance of the real multivariate Gaussian distribution.

In this scenario, one can exploit the fact (arising for instance from free probability considerations (Biane, 2003)) that, for all $i \neq j$ fixed, the moments $\frac{1}{n} \text{tr}(W^i (W^j)^T)$ vanish as $n \rightarrow \infty$. In our setting, i and j may however be growing with n , but then in all, in the large n setting, only the first few moments $\frac{1}{n} \text{tr}(W^i (W^j)^T)$, $i = 1, 2, \dots$, do not vanish. Although the implication is not immediate, this remark leads naturally to the intuition that

the Toeplitz matrix \mathcal{R} defined in Proposition 4 should be diagonal and thus proportional to the identity matrix.

At this point, we need to differentiate the cases where $c < 1$ and $c > 1$.

3.1.1 CASE $c < 1$

Based on the remarks above, we may explicitly solve for \mathcal{R} and $\tilde{\mathcal{R}}$ to find that, in the large n limit

$$\begin{aligned} \mathcal{R} &\leftrightarrow \frac{c}{1-c} I_r \\ \tilde{\mathcal{R}} &\leftrightarrow (1-c) S_0 \\ \hat{g}^{[B]} &\leftrightarrow \frac{c}{(1-c)^3} \frac{1}{n} \text{tr}(S_0^{-1} B) I_r \\ \hat{g}^{[B]} &\leftrightarrow \frac{c}{1-c} \frac{1}{n} \text{tr}(S_0^{-1} B) S_0. \end{aligned}$$

Replacing in the expressions of both Corollaries 5–11, we obtain the further corollary

Corollary 13 (Orthogonally invariant case, $c < 1$) *Let W be random and left and right independently orthogonally invariant. Then, under Assumptions 1–4 and with $c < 1$, the following hold*

$$\begin{aligned} E_\eta(u, r; \hat{u}, \hat{r}) &\leftrightarrow (1-c) \frac{1}{T} r^T \left(I_T + \frac{1}{\eta^2} U^T D U \right)^{-1} r \\ E_\eta(u, r; \hat{u}, \hat{r}) &\leftrightarrow \left\| \frac{1}{\eta^2 \sqrt{T}} \hat{U}^T D U \left(I_T + \frac{1}{\eta^2} U^T D U \right)^{-1} r - \frac{1}{\sqrt{T}} \hat{r} \right\|^2 \\ &\quad + \frac{1}{1-c} \frac{1}{T} r^T \left(I_T + \frac{1}{\eta^2} U^T D U \right)^{-1} r - \frac{1}{T} r^T \left(I_T + \frac{1}{\eta^2} U^T D U \right)^{-2} r \end{aligned}$$

where we defined $D \equiv \{m^T (W^i)^T S_0^{-1} W^j m\}_{i,j=0}^{T-1}$ and $\hat{D} \equiv \{m^T (W^i)^T S_0^{-1} W^j m\}_{i,j=0}^{\hat{T}-1, T-1}$.

We see here that the matrix D plays a crucial role in the ESN performance. First, from its Gram structure and the positive definiteness of S_0 , D is symmetric and nonnegative definite. This matrix has an exponential decaying profile down its rows and columns. As such, the dominating coefficients of the matrix $U^T D U$ lie in its upper-left corner. Recalling that the j -th column of $\sqrt{T} U^T$ is $\{u_{i-j}\}_{i=1}^T$, $U^T D U$ is essentially a linear combination of the outer products $\{u_{i-j}\}_{i=1}^T \{u_{i-j}\}_{i=1}^T$ for small j ,³ that is of combinations of (outer-products of) short-time delayed versions of the input vector u .

Note that, although we do not provide a rigorous proof of this fact, by the standard universality property of random matrix results, Corollary 13 is equally valid if W is chosen to be a matrix with i.i.d. zero mean and variance σ^2 non-necessarily Gaussian entries. In particular, it extends to the case of (properly recentered) matrices W with Bernoulli entries of Bernoulli parameter not scaling with n . In the regime under consideration, the asymptotic performance equivalence suggests that (here non sparse) Bernoulli random matrices have no particular advantage when compared to Gaussian random matrices, which is somewhat opposed to what is sometimes suggested in the ESN literature.³

Now, it is interesting to particularize the vector m and study its impact on D . It may be thought that taking m to be one of the dominant eigenvectors of W could drive the

3. However, sparse connectivity matrices prevail over non sparse ones in the literature, in which case our claim no longer holds.

inputs towards interesting memory-capacity levels of W ; this aspect is discussed in (Ganguli et al., 2008) where it is found that such an m maximizes the integrated Fisher-memory curve. If such a real eigenvector having eigenvalue close to σ exists, then we would find that $D_{ij} \simeq \sigma^{i+j} m^{-1} S_0^{-1} m$ and thus D would essentially be a rank-one matrix. As we shall discuss below, this would lead to extremely bad MSE performance in general.

If instead m is chosen deterministic or random independent of W with say $\|m\| = 1$ (or tending to one) for simplicity, then by the trace lemma (Bai and Silverman, 2009, Lemma B.26), one can show that $m^T (W^T)^T S_0^{-1} W^j m \leftrightarrow \frac{1}{n} \text{tr} W^j (W^T)^T S_0^{-1}$. According to our earlier discussion, this quantity vanishes for all $i \neq j$ as $n \rightarrow \infty$, and thus D would now essentially be diagonal. Besides, it is clear that $\text{tr} D \leftrightarrow 1$ and thus D here plays the role of affecting a short-term memorization ability, that can be seen as a total load 1, to the successive delayed versions of u . In particular, from our definition in Remark 7, we have precisely here

$$\text{MC}(\tau) = \frac{1}{1-c} \liminf_n \frac{1}{n} \text{tr} (W^T (W^T)^T S_0^{-1})$$

which, for the chosen m , is precisely the Fisher memory curve (Ganguli et al., 2008), up to the factor $1-c$.

Remark 14 (Haar W and independent m) For $W = \sigma Z$ with Z Haar distributed (orthogonal and orthogonally invariant) and m independent of Z and of unit norm, D is asymptotically diagonal and we find precisely

$$D_{ii} \leftrightarrow (1 - \sigma^2) \sigma^{2(i-1)}$$

and in particular

$$\text{MC}(\tau) = \frac{1 - \sigma^2}{1 - c} \sigma^{2r}.$$

Remark 14 can be extended to design an interesting multiple memory-mode network as follows.

Remark 15 (Multiple memory modes) Take W to be the block diagonal matrix $W = \text{diag}(W_1, \dots, W_k)$ where, for $j = 1, \dots, k$, $W_j = \sigma_j Z_j$, $\sigma_j > 0$, and $Z_j \in \mathbb{R}^{n_j \times n_j}$ is Haar distributed, independent across j . Take then m independent of W with unit norm. Also assume that $n_j/n \rightarrow c_j > 0$ as $n \rightarrow \infty$ and $\sum_j n_j = n$. Then we find that

$$D_{ii} \leftrightarrow \sum_{j=1}^k c_j \sigma_j^{2(i-1)} \frac{1}{\sum_{j=1}^k c_j (1 - \sigma_j^2)^{-1}}$$

and in particular, with $\text{MC}(\tau)$ defined in Remark 7,

$$\text{MC}(\tau) = \frac{1}{1-c} \frac{\sum_{j=1}^k c_j \sigma_j^{2r}}{\sum_{j=1}^k c_j (1 - \sigma_j^2)^{-1}}.$$

A graph of $\text{MC}(\tau)$ for $k=3$ is depicted in Figure 1, where it clearly appears that the memory curve follows successively each one of the three modes, giving in particular more weight to short-term past inputs at first, and then smoothly providing increasingly more importance to longer-term past inputs. This is reminiscent of the long short-term memory framework devised in (Xue et al., 2007).

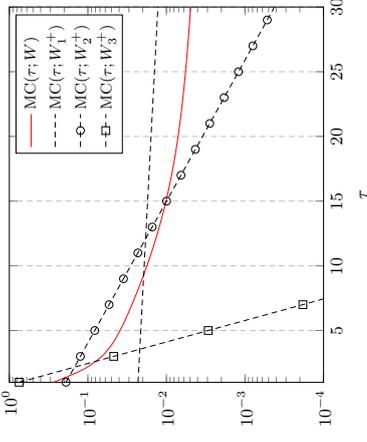


Figure 1: Memory curve for $W = \text{diag}(W_1, W_2, W_3)$, $W_j = \sigma_j Z_j$, $Z_j \in \mathbb{R}^{n_j \times n_j}$ Haar distributed, $\sigma_1 = .99$, $n_1/n = .01$, $\sigma_2 = .9$, $n_2/n = .1$, and $\sigma_3 = .5$, $n_3/n = .89$. The matrices W_i^+ are defined by $W_i^+ = \sigma_i Z_i^+$, with $Z_i^+ \in \mathbb{R}^{n_i \times n_i}$ Haar distributed.

It is next interesting to study Corollary 13 more deeply. Let us first assume that the task to be performed, both in training and testing, consists in retrieving a mere linear combination of latest past inputs $u_k, u_{k-1}, \dots, u_{k-(k-1)}$ for k fixed. Then we may write $r = \sqrt{T} U^T b$ for some vector $b \in \mathbb{R}^T$ with $b_j = 0$ for all $j \geq k$. We then have

$$E_\eta(u, r) \leftrightarrow (1-c)b^T U \left(I_T + \frac{1}{\eta^2} U^T D U \right)^{-1} U^T b.$$

For D positive diagonal with exponential decaying profile, D^{-1} is extremely ill-conditioned and may only be used with extreme care. However, for k fixed, $D^{-\frac{1}{2}} b$ is well behaved as its norm is bounded by $\|b\| D_{k-1, k-1}^{-\frac{1}{2}}$. We may thus write $b = D^{\frac{1}{2}} (D^{-\frac{1}{2}} b)$ to obtain, after basic algebraic manipulations

$$E_\eta(u, r) \leftrightarrow \eta^2 (1-c) (D^{-\frac{1}{2}} b)^T \frac{1}{\eta^2} D^{\frac{1}{2}} U U^T D^{\frac{1}{2}} \left(I_T + \frac{1}{\eta^2} D^{\frac{1}{2}} U U^T D^{\frac{1}{2}} \right)^{-1} (D^{-\frac{1}{2}} b).$$

Since $\|A(I+A)^{-1}\| \leq 1$ for any symmetric nonnegative definite matrix A , we thus conclude that, for every $\eta, \varepsilon > 0$, $E_\eta(u, r) \leq (1-c)\eta^2 b^T D^{-1} b + \varepsilon$ for all large n almost surely. Thus, for sufficiently large n , $E_\eta(u, r)$ can be made arbitrarily small in the limit where $\eta \rightarrow 0$ and thus the task can be performed accurately. As for E_η , note that, since D and D are essentially zero away from the upper left corner and otherwise equal, if $\hat{r} = \sqrt{T} \hat{U} \hat{b}$, for $\hat{b} \in \mathbb{R}^T$ having the same first k entries as b and zeroes next, then we find

$$\begin{aligned} \hat{E}_\eta(u, r; \hat{u}, \hat{r}) &\leftrightarrow \frac{\eta^2}{1-c} (D^{-\frac{1}{2}} b)^T \frac{1}{\eta^2} D^{\frac{1}{2}} U U^T D^{\frac{1}{2}} \left(I_T + \frac{1}{\eta^2} D^{\frac{1}{2}} U U^T D^{\frac{1}{2}} \right)^{-1} (D^{-\frac{1}{2}} b) \\ &\quad + (D^{-\frac{1}{2}} b)^T \left(I_T + \frac{1}{\eta^2} D^{\frac{1}{2}} U U^T D^{\frac{1}{2}} \right)^{-1} D^{\frac{1}{2}} \Delta D^{\frac{1}{2}} \left(I_T + \frac{1}{\eta^2} D^{\frac{1}{2}} U U^T D^{\frac{1}{2}} \right)^{-1} (D^{-\frac{1}{2}} b) \end{aligned} \quad (7)$$

where $\Delta \equiv [\hat{U}^T I_{r \times r} - U U^T]$, with the operator $[X]_{I_{r \times r}}$ extending (or reducing) X to a $T \times T$ matrix by filling it with zeroes (or discarding fast rows and columns). Note here that, for $U = \hat{U}$, $\Delta = 0$ and we find that

$$\hat{E}_\eta(u, r; u, r) \leftrightarrow \frac{1}{(1-c)^2} E_\eta(u, r). \quad (8)$$

When $\Delta \neq 0$, observe first that $\|(I_r + \eta^{-2} D^{\frac{1}{2}} U U^T D^{\frac{1}{2}})^{-1}\| \leq 1$ and thus $\hat{E}_\eta(u, r; \hat{u}, \hat{r})$ remains bounded. Now, with a more subtle analysis, note that, since the product $B D^{-\frac{1}{2}} b$ for any matrix B only concerns the first k columns of B , the behavior of E_η as $\eta \rightarrow 0$ merely depends on the behavior of the top-left $k \times k$ submatrix of $(I_r + \eta^{-2} D^{\frac{1}{2}} U U^T D^{\frac{1}{2}})^{-1}$. A block matrix inverse then reveals that the second right-hand side term of (7) goes to zero as $\eta \rightarrow 0$ provided that the k -th largest eigenvalue of $D^{\frac{1}{2}} U U^T D^{\frac{1}{2}}$ remains away from zero as $T \rightarrow \infty$. From the structure of U , we thus conclude that, for E_η to vanish as $\eta \rightarrow 0$, it is sufficient for the vector u to be sufficiently ‘‘diverse’’ in its constituents (that is, so that the first columns of U remain linearly independent). An obvious counter-example is when the sequence $\dots, u_{-1}, u_0, u_1, \dots \in \mathbb{R}$ is periodic of period less than k . Note that the specific choice of \hat{u} does not alter this behavior.

The discussion above leads to interesting practical considerations that may help improve the design of an ESN.

Remark 16 (Selecting W based on delayed correlations) Note that, in the aforementioned formulas, the quantity $b^T D^{-1} b$ with b defined by $r = U^T b$ appears as a fundamental quantity bounding the training and testing MSE. In practical settings where r is not a pure linear combination of delayed versions of u , it may nonetheless be useful to obtain an estimate \hat{b} of the closest approximation of r by delays of u , in such a way that $b^T D^{-1} b$ be small. One may for instance let

$$\hat{b} = (U U^T + \gamma I_r)^{-1} U r$$

for some regularization parameter $\gamma \geq 0$ (if needed), and parameterize W so that $b^T D^{-1} \hat{b}$ is minimal. For instance, if $b_i = \alpha^{i-1}$ for some $\alpha \in (-1, 1)$, it is easily shown that an optimal choice for $W = \sigma Z$ with Z Haar is to take $\sigma^2 = |\alpha|$. This scenario is illustrated in Figure 2, where the theoretical approximations for the testing and training normalized MSE are depicted for various choices of σ^2 . For less obvious values of \hat{b} , a more elaborate multi-memory matrix W , as introduced in Remark 15, can be used, with proper setting of the parameters n_i and σ_i .

Remark 17 (Memory Capacity for Stationary Inputs) Let W be orthogonally invariant and m random, so that D is diagonal in the limit. Further assume the sequence u is an anti-regressive Gaussian process, so that we may write $u = C^{\frac{1}{2}} \tilde{u}$ with \tilde{u} having independent zero mean unit variance Gaussian entries and C a Toeplitz covariance matrix with $C_{ab} = q^{|b-a|}$ for some $q \in [0, 1)$. Then, for the r -delay memory task, i.e., $r_i = u_{i-r}$ with τ fixed, we find that

$$E_\eta(u, r) \leftrightarrow \eta^2 \frac{1-c}{D_{\tau+1, \tau+1}} \left[1 - \left[\left(I_r + \frac{1}{\eta^2} \left\{ \sqrt{D_{i_i q}^{|i-j|}} \sqrt{D_{j_j}} \right\}_{i,j=0}^{T-1} \right)^{-1} \right]_{\tau+1, \tau+1} \right].$$

Since $q < 1$, the matrix $\{q^{|i-j|}\}_{i,j}$ has its smallest eigenvalue asymptotically far from zero (see e.g., (Gruy, 2006) for arguments) so that the right-hand side inner bracket vanishes as $\eta^2 \rightarrow 0$ and we thus have, for small η^2

$$\eta^{-2} E_\eta(u, r) \leftrightarrow \frac{1-c}{D_{\tau+1, \tau+1}} + o(\eta^2).$$

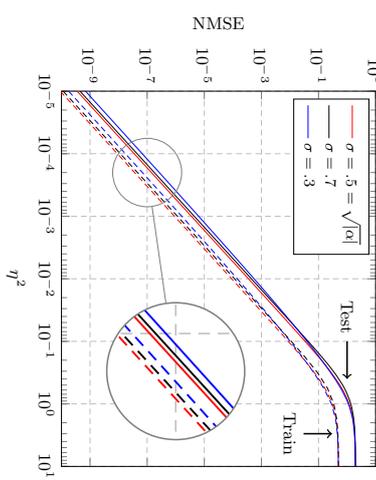


Figure 2: Optimal σ choice for $r_i = \sum_{i \geq 0} u_{-i} b_i$, $b_i = \alpha^{i-1}$ with $\alpha = -0.25$, u i.i.d. zero mean Gaussian, W Haar distributed, $n = 200$, $T = \hat{T} = 400$.

As a consequence, the memory task is performed irrespectively of the smoothness of u , so that u can be assumed composed of i.i.d. elements (i.e., $q = 0$). Observe that this leads to the same performance as the memory task considering $u = \sqrt{T} \delta_0$, defining the memory capacity in Remark 7. Of course, if instead $q = 1$, then the matrix in curly brackets would have unit rank and the previous conclusions would fail (in this case u is a constant vector).

Expression (8) also provides us an opportunity to open a short parenthesis on the effect of c on the training and testing MSE. From Corollary 13, it appears that, while E_η is minimal for $c = 1$, \hat{E}_η is minimal for $c = 0$. The former observation is clear from the fact that ω is a least-square regressor, but the latter observation is less trivial. As a matter of fact, note that, even if $U = I$ and $\hat{r} = r$, in the limit of $\eta > 0$ fixed and $c \rightarrow 1$, E_η becomes arbitrarily large. The reason for this seemingly counter-intuitive effect (after all, we merely ask the ESN to reproduce the exact learned sequence) lies in the fact that ω is built upon the network noise realization during training, while during testing a new noise realization is produced. As such, training an ESN of size almost equal to T produces dramatic effects on testing. However, this has the positive effect of strongly reducing over fitting. Of course, in practical settings, there exists an interplay between η^2 that drives both MSEs to zero as $\eta \rightarrow 0$ and c that reduces overfitting as it tends to 1.

Coming back to the approximations of E_η and \hat{E}_η , note now that if D is a rank-one matrix, then we may write $D = d d^T$ for some vector $d \in \mathbb{R}^T$ having exponentially vanishing entries. In this case, we find, again after standard algebraic calculus, that

$$E_\eta(u, r) \leftrightarrow (1-c) \left(\frac{1}{T} \|r\|^2 - \frac{1}{\eta^2} \frac{d^T U r r^2}{|d^T U r|^2} \right).$$

Taking as above $r = \sqrt{T} U b$, this is $E_\eta(u, r) \leftrightarrow (1-c) \left(b^T U U^T b - \frac{|d^T U U^T b|^2}{\eta^2 + d^T U U^T d} \right)$. By Cauchy-Schwarz inequality, this quantity, even in the limit $\eta^2 \rightarrow 0$, cannot vanish unless $b = d$.

As such, the ESN will only adequately fulfill a single task, which depends on the network configuration itself through d . A similar reasoning can be made on E_η , revealing the same shortcomings.

As a practical example, we provide in Figure 3 Monte Carlo simulations versus theory curves of the training and testing performances of networks of $n = 200$ and $n = 400$ nodes, for training and testing times $T = T = 2n$, on the Mackey Glass one-step ahead anticipation task (Glass and Mackey, 1979). The network is chosen to be the multi-memory model introduced in Remark 15 and following the description of Figure 1. The NMSE is defined here as the ratio between the MSE and the output vector squared norm $\|r\|^2/T$ or $\|s\|^2/\hat{T}$. Simulations are run for a single W but different noise realizations and comparison is made against theory for either this W or its approximated asymptotic limit. Observe the extremely accurate match between theory and practice, with increasing precision as n, T grow large.

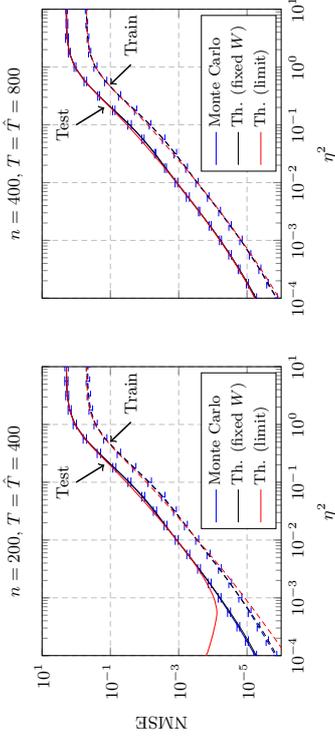


Figure 3: Training and testing (normalized) MSE for the Mackey Glass one-step ahead task, W fixed and defined as in Figure 1, $n = 200$, $T = \hat{T} = 400$ (left) and $n = 400$, $T = \hat{T} = 800$ (right). Comparison between Monte Carlo simulations (Monte Carlo), deterministic approximation assuming W fixed (Th. (fixed W)) as per Corollaries 5 and 11, and assuming W random in the large n limit (Th. (limit)) as per Corollary 13. Error bars indicate one standard deviation of the Monte Carlo simulations.

3.1.2 CASE $c > 1$

The case $c > 1$ is slightly more involved as it does not lend itself to a purely explicit expression. Precisely, following the same steps as for $c < 1$, we find that in the large n

limit

$$\begin{aligned} \tilde{R} &\leftrightarrow \alpha I_T \\ \tilde{R} &\leftrightarrow \frac{1}{\alpha} S_0 \\ \hat{G}^{[B]} &\leftrightarrow \alpha \alpha^{\frac{1}{n}} \frac{\frac{1}{n} \text{tr} S_0 (\alpha I_n + S_0)^{-1} B (\alpha I_n + S_0)^{-1} I_T}{1 - c \frac{1}{n} \text{tr} S_0^2 (\alpha I_n + S_0)^{-2}} \\ \hat{G}^{[B]} &\leftrightarrow c \frac{1}{n} \frac{\frac{1}{n} \text{tr} S_0 (\alpha I_n + S_0)^{-1} B (\alpha I_n + S_0)^{-1} S_0}{1 - c \frac{1}{n} \text{tr} S_0^2 (\alpha I_n + S_0)^{-2}} \end{aligned}$$

where $\alpha > 0$ is the unique solution to the equation

$$1 = c \frac{1}{n} \text{tr} S_0 (\alpha I_n + S_0)^{-1}.$$

With these notations, we have the following counterpart to Corollary 13.

Corollary 18 (Orthogonally invariant case, $c > 1$) *Let W be random and left and right independently orthogonally invariant and let $\alpha > 0$ be the unique solution to $1 = c \frac{1}{n} \text{tr} S_0 (\alpha I_n + S_0)^{-1}$. Then, under Assumptions 1–4 and with $c > 1$, the following holds*

$$\begin{aligned} \hat{E}_\eta(u, r; \hat{u}, \hat{r}) &\leftrightarrow \left\| \frac{\eta^{-2} \hat{U}^T \hat{D} U (I_T + \eta^{-2} U^T D U)^{-1} \frac{r}{\sqrt{T}} - \frac{1}{\sqrt{T}} r^T (I_T + \eta^{-2} U^T D U)^{-1} r}{\frac{1}{T} r^T (I_T + \eta^{-2} U^T D U)^{-1} [I_T + \eta^{-2} U^T D_2 U] (I_T + \eta^{-2} U^T D U)^{-1} r} \right\|^2 \\ &\quad + \frac{\frac{1}{T} r^T (I_T + \eta^{-2} U^T D U)^{-1} [I_T + \eta^{-2} U^T D_2 U] (I_T + \eta^{-2} U^T D U)^{-1} r}{1 - c \frac{1}{n} \text{tr} S_0^2 (\alpha I_T + S_0)^{-2}} \end{aligned}$$

where $D \equiv \{m^T (W^i)^T (\alpha I_n + S_0)^{-1} W^j m\}_{i,j=0}^{T-1}$, $\hat{D} \equiv \{m^T (W^i)^T (\alpha I_n + S_0)^{-1} W^j m\}_{i,j=0}^{T-1, T-1}$, and $D_2 \equiv \{m^T (W^i)^T (\alpha I_n + S_0)^{-1} S_0 (\alpha I_n + S_0)^{-1} W^j m\}_{i,j=0}^{T-1}$.

Of course here $E_\eta(u, r) = 0$.

Remark 19 (Haar W , random m for $c > 1$) *Although seemingly less tractable, for W following a Haar model, Corollary 18 takes a much simpler form. Indeed, for W and m as defined in Remark 14, we find that $\alpha = (c-1)(1-\sigma^2)^{-1}$ and $S_0 = (1-\sigma^2)^{-1} I_n$ which then leads to*

$$\begin{aligned} \hat{E}_\eta(u, r; \hat{u}, \hat{r}) &\leftrightarrow \left\| (c\eta^2)^{-1} \hat{U}^T \hat{D} U (I_T + (c\eta^2)^{-1} U^T D U)^{-1} \frac{r}{\sqrt{T}} - \frac{1}{\sqrt{T}} r^T \right\|^2 \\ &\quad + \frac{1}{c-1} \frac{1}{T} r^T (I_T + (c\eta^2)^{-1} U^T D U)^{-1} r \end{aligned}$$

where D is diagonal with $D_{ii} \equiv (1-\sigma^2)\sigma^{2(i-1)}$.

Aside from obtaining a shorter form expression for D and D_2 , the multi-memory model of Remark 15 does not lead to an explicit formulation as in Remark 19, but it is nonetheless instructive to observe the performance achieved on the Mackey Glass model from Figure 3, now in the setting where $c > 1$. This is depicted here in Figure 4.

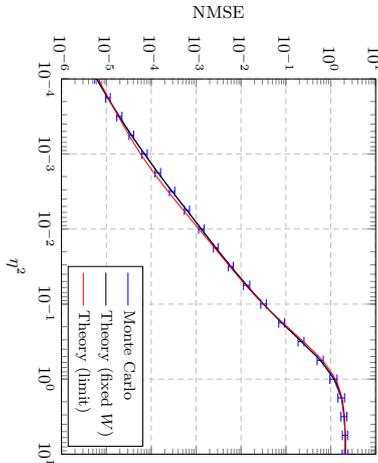


Figure 4: Testing (normalized) MSE for the Mackey Glass one-step ahead task. W fixed and defined as in Figure 1, $n = 400$, $T = \tilde{T} = 200$. Error bars indicate one standard deviation of the Monte Carlo simulations.

3.2 Normal W

We now turn to the case of normal matrices. Let then W be normal (i.e., diagonalizable in orthogonal basis) and having an eigenvalue decomposition of the type $W = V\Lambda V^T$ with V orthogonal and Λ diagonal with largest absolute entry less than one. For simplicity, we shall further assume that, as $n \rightarrow \infty$, the normalized counting measure of the diagonal elements of Λ ($n^{-1}\sum_i \delta_{\Lambda_i}$) converges in law to a probability measure μ . We do not make any assumption here on V .

For instance, real Gaussian Wigner matrices W , that is with i.i.d. zero mean variance $\frac{1}{4}\sigma^2$ Gaussian entries on and above the diagonal, and symmetrized below the diagonal, is an example of such a matrix. In this case, μ corresponds (almost surely) to the well-known semi-circular distribution, with density $\mu(d\lambda) = 2(\pi\sigma^2)^{-1}\sqrt{(\sigma^2 - \lambda^2)^+}d\lambda$. Another example is when $\mu(d\lambda) = \frac{1}{2}[\delta_c + \delta_{-c}]d\lambda$, so that W is the sum of two (σ^2 -scaled) projection matrices on orthogonal subspaces. In particular here, $W^2 = \sigma^2 I_n$, so that $W^{2k} = \sigma^{2k} I_n$ and $W^{2k+1} = \sigma^{2k} W$, for all $k \geq 0$.

Because of the symmetry property, it is no longer true that $\frac{1}{n} \text{tr } W^t (W^j)^T = \frac{1}{n} \text{tr } W^{t+j}$ vanished for $t \neq j$, and we then obtain more involved results. To keep this discussion short and since the results take here more involved forms, we shall only deal here with the case $c < 1$ and focus on the training performance. In this case, solving Proposition 4 for \mathcal{R} and \mathcal{R} , we have the following result. As $n \rightarrow \infty$, \mathcal{R} has a limit (which for simplicity we keep calling \mathcal{R}) which is solution to

$$\mathcal{R}_{ab} = c \int \frac{t^{a-b} \mu(dt)}{\sum_{q=-\infty}^{\infty} \frac{1}{t^q} \text{tr}(J^q (I_T + \mathcal{R})^{-1}) |t|^q} \quad (9)$$

for all $a, b \in \{1, \dots, T\}$. Remember that \mathcal{R} is Toeplitz with fast decaying values off the diagonal, so that (9) is computationally easy to solve. Similar conclusions can be drawn on the matrices $\mathcal{G}^{|\beta|}$ and $\tilde{\mathcal{G}}^{|\beta|}$, that however do not lead to simple expressions.

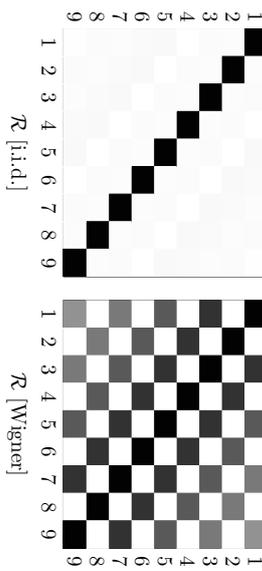


Figure 5: Upper 9×9 part of \mathcal{R} for $c = 1/2$ and $\sigma = 0.9$ for W with i.i.d. zero mean Gaussian entries (left) and W Gaussian Wigner (right). Linear grayscale representation with black being 1 and white being 0.

Remark 20 (Symmetric μ) An interesting scenario is when μ is symmetric, i.e., $\mu(-t) = \mu(t)$, which is the case of both aforementioned (Wigner and projection matrix) examples. From (9), we find in this case that $[\mathcal{R}]_{ab}$ is zero if $a - b$ is odd and positive if $a - b$ is even. As such, \mathcal{R} takes the form of a checkerboard matrix. Figure 5 provides a representation of \mathcal{R} in both normal and non-normal Gaussian W cases.

Remark 21 (Projection W) Let $W = V\Lambda V^T$ with the normalized counting measure of Λ converging to $\mu(d\lambda) = \frac{1}{2}[\delta_c + \delta_{-c}]d\lambda$ and $c < 1$. Then, $\mathcal{R}_{ab} \leftrightarrow \sigma^{|b-a|} r_0 \delta_{|b-a| \in 2\mathbb{N}}$ and $\mathcal{R} \leftrightarrow -(1-\sigma^2)^{-1} \sigma r_0^{-1} I_n$ where

$$r_0 = c \left(\sum_{j=-\infty}^{\infty} \frac{1}{T^j} \text{tr } J^j (I_T + \mathcal{R})^{-1} \right)^{-1}.$$

As a consequence, letting m be random, we find that

$$E_{r_0}(u, r) \leftrightarrow \frac{1}{T} \text{tr} \left(I_T + r_0 \left\{ \sigma^{|j-i|} \delta_{|j-i| \in 2\mathbb{N}} \right\}^{T-1} + r_0 (1-\sigma^2) U^T \left\{ \sigma^{j+i} \delta_{|j-i| \in 2\mathbb{N}} \right\}^{T-1} U \right)^{-1} r.$$

Note in particular that the matrix $\{\sigma^{|j+i|} \delta_{|j-i| \in 2\mathbb{N}}\}_{i,j=0}^{T-1}$ can be decomposed as the sum of two matrices: (i) the rank-one matrix $v v^T$ with $v = (1, 0, \sigma^2, 0, \sigma^4, \dots)^T$ and the diagonal matrix $\text{diag}(0, \sigma^2, 0, \sigma^4, \dots)$. Recalling that rank-one matrices in this position do not allow for efficient training (see the final discussions in Section 3.1, $c < 1$ case), only the diagonal component $\text{diag}(0, \sigma^2, 0, \sigma^4, \dots)$ really matters here. This diagonal misses half its entries and thus intuitively does not allow for efficient retrieval of odd past steps. This remark generally prefigures a weaker performance of normal matrices with symmetric spectrum than their non-normal counterparts.

The final discussion in Remark 21 motivates a deeper comparative study of the performance of non-normal versus normal connectivity matrices. From (9), we may in particular evaluate the memory curve (as defined here in Remark 7) for W a Wigner random matrix. The performance figures are displayed in Table 1, which show a dramatic decay of the memory curve for the Wigner connectivity matrix as compared to an i.i.d. Gaussian non-normal

matrix. In Figure 6, a practical scenario of a τ -delay task is depicted comparatively for Haar versus Wigner matrices (the input data being extracted from Mackey-Glass processes but the general results hold for any non-trivial input dataset); there we confirm that, for increasing values of the delay τ , the ESN performance strongly decays for Wigner matrices as compared to Haar matrices, as predicted by the theoretical results of Table 1.

τ	i.i.d.	Wigner
0	$5.2 \cdot 10^{-1}$	$4.8 \cdot 10^{-1}$
1	$2.0 \cdot 10^{-1}$	$1.6 \cdot 10^{-2}$
2	$1.0 \cdot 10^{-1}$	$1.3 \cdot 10^{-3}$
3	$6.0 \cdot 10^{-2}$	$2.0 \cdot 10^{-4}$
4	$3.9 \cdot 10^{-2}$	$5.7 \cdot 10^{-5}$

Table 1: Memory curve $MC(\tau)$ for i.i.d. versus Wigner matrices, $c = .5$.

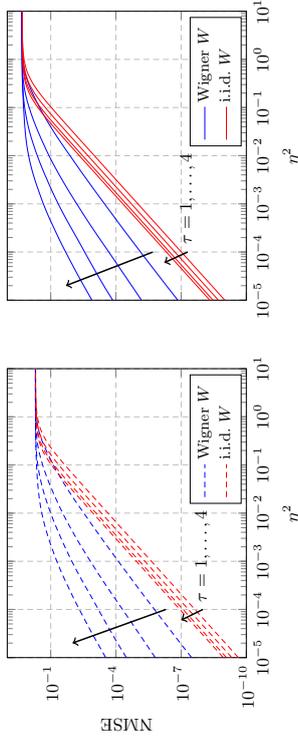


Figure 6: Training (left) and testing (right) performance of a τ -delay task for $\tau \in \{1, \dots, 4\}$ compared for i.i.d. W versus Wigner W , $\sigma = .9$ and $n = 200$, $T = \hat{T} = 400$ in both cases (here on the Mackey-Glass dataset).

An application example in a less artificial context is devised in Figure 7, where, on a real dataset of daily pollution (PM10) records, we provide the one-day ahead interpolation performance of neural networks assuming m random i.i.d. and either (i) W with i.i.d. Gaussian entries or (ii) W Gaussian Wigner. We observe again a better performance achieved by the ESN with non-normal matrix W which, accordingly with the fact that ESN's rely heavily on past input retrieval, is coherent with the previous remark.

3.3 Further Experiments

In this section, we provide further noticeable results of interest to neural network optimization.

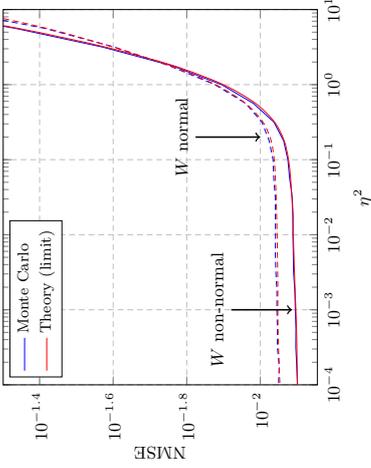


Figure 7: Testing (normalized) MSE for the PM10 one-step ahead task, W i.i.d. Gaussian or Gaussian Wigner ($\sigma = .9$), $n = 200$, $T = \hat{T} = 400$.

To start with, we consider a scenario where the testing dataset is polluted by an additional impulsive white Gaussian noise arising independently with probability p . This is depicted in Figure 8 for the Mackey-Glass one-step ahead task. It is observed here that the in-network noise is valuable in bringing the normalized MSE down to acceptable values. It is in particular seen that the more the noise impulsion probability the larger the variance η^2 should be chosen. A particular realization of the noisy Mackey-Glass output is provided in Figure 9, where it is observed that a visually small noise impulsion in the input vector drives a large fluctuation of the output for a too small- η^2 ESN.

This phenomenon can be theoretically anticipated in simple settings. Let us consider the scenario of Section 3.1 with W orthogonally invariant, where $r = U^T b$ for a vector $b \in \mathbb{R}^T$ having only its last $T - k$ entries identically zero for some fixed k ; let us now assume that $\hat{u} = \hat{u}_0 + \hat{\epsilon}$ for some noise vector $\hat{\epsilon}$ made of i.i.d. zero mean and variance s^2 entries, and suppose that $\hat{r} = \hat{U}_0$ for $\{\hat{U}_0\}_{ij} = [\hat{u}_0]_{i=j}$. Then, an application of Corollary 13 leads to $\hat{E}_\eta(u, r; \hat{u}, \hat{r})$ asymptotically equal to (7) plus an additional term given by (after calculus)

$$s^2 \left\| \left(\eta^2 I_T + D^{\frac{1}{2}} U U^T D^{\frac{1}{2}} \right)^{-1} D^{\frac{1}{2}} U U^T D^{\frac{1}{2}} (D^{-\frac{1}{2}} b) \right\|^2. \quad (10)$$

From the inequality $\|(\eta^2 I_T + D^{\frac{1}{2}} U U^T D^{\frac{1}{2}})^{-1}\| \leq \eta^{-2}$ and the fact that $\|D^{-\frac{1}{2}} b\|$ remains bounded, we get that the term (10) can be made arbitrarily small by letting $\eta^2 \rightarrow \infty$. Therefore, η^2 induces robustness in this scenario. Since $\eta^2 \rightarrow 0$ was shown to be optimal in the scenario where $s^2 = 0$, there must exist an MSE minimizing choice of $\eta^2 \in (0, \infty)$.

In a second experiment, we shall illustrate the ‘‘noise resurgence’’ effect discussed earlier in Remark 6. In Figure 10, we specifically draw the curves of the testing MSE variances for various experiments conducted earlier in the article. It is observed, as discussed in Remark 6 that, somewhat counter-intuitively, smaller η^2 values may lead to increased variances solely due to the in-network noise realization itself (recall that in all our experiments,

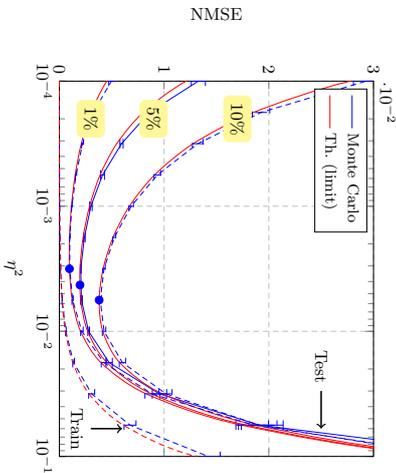


Figure 8: Testing (normalized) MSE for the Mackey-Glass one-step ahead task with 1% or 10% impulsive $\mathcal{N}(0, .01)$ noise pollution in test data inputs, W Haar with $\sigma = .9$, $n = 400$, $T = \hat{T} = 1000$. Circles indicate the NMSE theoretical minima. Error bars indicate one standard deviation of the Monte Carlo simulations.

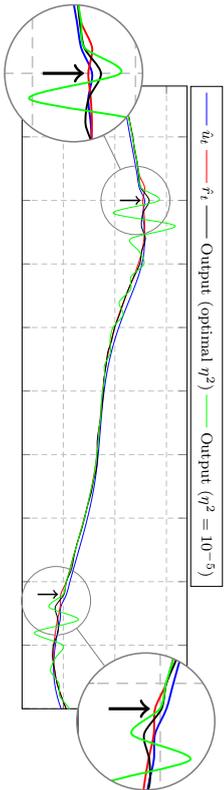


Figure 9: Realization of a 1% $\mathcal{N}(0, .01)$ -noisy Mackey-Glass sequence versus network output, W Haar with $\sigma = .9$, $n = 400$, $T = \hat{T} = 1000$. In magnifying lenses, points of added impulsive noise.

the connectivity matrix W and the input-output pairs (u_i, r_i) and (\hat{u}_i, \hat{r}_i) are fixed across all Monte Carlo realizations). It is even more interesting to observe here each of the three possible behaviors: a “natural” MSE variance decay as $\gamma^2 \rightarrow 0$, a surprising MSE increase, and even an MSE stabilization. Further theoretical analysis to understand those strikingly different behaviors would be appreciable, which would demand more advanced technical considerations.

We complete this section by a last comparative experiment of the performance of the multi-memory matrix W defined in Remark 15 specialized to the setting of Figure 1 (that is, with three rates $\sigma_1 = .99$, $\sigma_2 = .9$, and $\sigma_3 = .5$) versus Haar matrices for the different σ_i

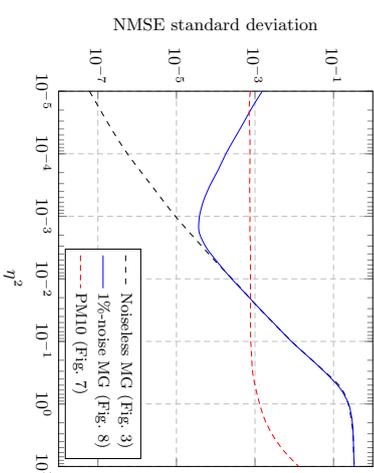


Figure 10: Standard deviation of testing NMSE for different testbeds (exemplifying the resurgence of noise effect). MG in legend stands for Mackey-Glass. In all scenarios, $n = 200$, $T = \hat{T} = 400$.

values, for the Mackey-Glass model. This is depicted in Figure 11, which shows a valuable performance gain versus ill-chosen individual hypotheses of σ and a rather fair match to the best individual σ value.

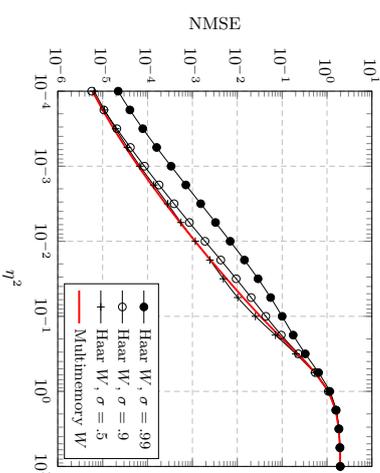


Figure 11: Testing (normalized) MSE for the Mackey-Glass one-step ahead task, W (multi-memory) versus $W_1^+ = .99Z_1^+$, $W_2^+ = .9Z_2^+$, $W_3^+ = .5Z_3^+$ (with Z_i^+ Haar distributed) all defined as in Figure 1, $n = 400$, $T = \hat{T} = 800$.

4. Concluding Remarks

One of the main outcomes of the present study is a better understanding of the ESN instability to low internal noise variance described by Jaeger in (Jaeger, 2001). We made it clear here that, when the noise variance is sufficiently large compared to the inverse square root of the network size, the ESN tends to have a deterministic behavior (that is, independent of the noise realization) as both time and network size grow large. This deterministic behavior was characterized here through new results from random matrix theory, with the main consequences to ESN's being encapsulated in Corollary 5 and Corollary 11. When the noise variance is however too small, random matrix theory cannot guarantee in general the aforementioned deterministic network behavior in the large system asymptotic. Although difficult to read, the asymptotic performances revolve around a critical matrix that contains the exponential memory decay information and may be used to generalize Ganguli's notion of memory curve (see Remark 7). This generalized memory curve draws improved conclusions on the ESN performance (with sometimes opposite outcomes as compared to the conclusions drawn upon the former memory curve notion).

In the particular case of some standard random matrix models for the neural connectivity matrix, we further simplified the rather involved generic expressions from Corollary 5 and Corollary 11. Of particular interest is the case of bi-orthogonally invariant random connectivity matrices for which the mean square error performances of learning and testing take on the explicit expressions of Corollary 13 or Corollary 18 from which much can be inferred. Among other results, we understood the importance of random input weights for the network performance as compared to input weights that match the leading eigenvectors of the connectivity matrix and we made it clear that the ESN testing performance is asymptotically optimal for arbitrary low noise variances *when* the task to fulfill is a mere linear combination of the last few past inputs. In additional experiments, we also understood the role of a non-trivial noise level as a robustness-to-outliers enhancer.

Beyond their theoretical value, note also that the results of this article may be used in practice to anticipate the behavior of ESN's on real-life datasets, thereby saving one from the painstaking task of running long Monte Carlo simulations. For instance, one may consider retrieving the theoretical MSE outputs corresponding to successive sequences of training and testing inputs so to better tune the ESN parameters. This is all the more precious that the network size and time windows are large since then the formulas of, say Corollary 13, can be retrieved extremely fast. In practice, for random networks, results such as Corollary 13 can be evaluated at a computational cost of $O(T^2)$ operations with minimal optimization (one may exploit the Toeplitz structure in U to further improve computations), when each run of a Monte Carlo simulation requires a prior evaluation of the successive products Wx_t , $t = 1, \dots, T-1$, to evaluate X , prior to evaluating $(XX^T)^{-1}X$; the latter amounts to a total minimum cost of $O(RTn^2)$ with R the number of Monte Carlo iterations.

One frustrating aspect of the work nonetheless remains that, for low noise variances (typically of practical interest), our analysis leads to large mismatches when the network size is kept moderate. This is observed in Figure 3 in particular. There is as such no theoretical control of this regime. This being said, in some scenarios where the limiting singularity at zero noise can be avoided, we showed an accurate fit of our theoretical findings at all noise levels. But the main limitation of the analysis so far lies in its dealing with linear activation functions only. In a currently on-going study, using more advanced notions of random matrix theory, the authors have managed to overcome the non-linearity limitation in retrieving deterministic approximations for the mean-square error performance of a single-layer feedforward neural network with random input layer (sometimes referred

to as extreme learning machines (Huang et al., 2006)). Since the key to this result lies in the independence of the random connectivity matrix entries when seen from each neuron, the extension to multi-layer networks and eventually to recurrent network is naturally envisioned in a future investigation.

Acknowledgments

The work of Couillet and Tiomoko Ali is supported by the ANR RMT4GRAPH Project (ANR-14-CE28-0006).

Appendix A. Proof of Theorem 2

The present and next sections are dedicated to the proofs of the main results Theorems 2-9 of the article. The proofs rely on now well-established tools from random matrix theory, with an additional specificity due to the "infinitely long" time dependence between the columns of the random matrices involved; however, as the time dependence is *effectively* short (of order $o(T^\alpha)$ for any $\alpha > 0$), these matrices can be handled as if dependence was among only a few next and previous columns. We shall not deeply elaborate on all technical arguments for the sake of readability and concision. The reader more interested in the proof techniques and in more advanced time dependence considerations may refer to (Pastur and Šerbina, 2011; Hachem et al., 2008) on the Gaussian methods and (Baum and Merlevède, 2013) for stationary processes in random matrix theory.

Before delving into the proof of Theorem 2, let us first prove Lemma 1 which provides the expression of interest for $E_\eta(u, r)$ exploited in Theorem 2. The result is clearly valid when $n/T > 1$ as $X^T X$ is almost surely non singular. Thus, only the scenario where $n/T < 1$ is of interest. Expanding the expression of ω , first observe that $E_\eta(u, r) = \frac{1}{2} \|r - X^T \omega\|^2 = \frac{1}{2} r^T (I_T - X^T (X X^T)^{-1} X) r$ and that $(I_T - X^T (X X^T)^{-1} X)^2 = I_T - \frac{1}{2} r^T (X X^T)^{-1} X$. Introducing $\gamma > 0$, $E_\eta(u, r) = \lim_{\gamma \rightarrow 0} \frac{1}{2} r^T (I_T - X^T (X X^T + \gamma I_n)^{-1} X) r$. Now, using the identities $(AB+I)^{-1} A = A(BA+I)^{-1}$ and $A(A+bI)^{-1} = I - b(A+bI)^{-1}$ for matrices A, B and scalar b , this is $E_\eta(u, r) = \lim_{\gamma \rightarrow 0} \frac{1}{2} r^T (I_T - X^T X (X^T X + \gamma T I_T)^{-1}) r = \lim_{\gamma \rightarrow 0} \frac{1}{2} \gamma T r^T (X^T X + \gamma T I_T)^{-1} r$, from which Lemma 1 follows.

With the result at hand, we are ready to tackle the proof of Theorem 2. In the present section, W is considered a deterministic matrix with operator norm less than unity. We recall that $X = \{x_j\}_{j=0}^{T-1} \in \mathbb{R}^{n \times T}$, for the infinite time series $\dots, x_{-1}, x_0, x_1, \dots \in \mathbb{R}^n$, defined recursively through

$$x_{t+1} = Wx_t + mu_{t+1} + \eta \varepsilon_{t+1}$$

with m of bounded norm and $\dots, u_{-1}, u_0, u_1, \dots \in \mathbb{R}$ some time series. We additionally denote $A = MU$ where $M = \{W^k/m\}_{k=0}^{T-1}$ and $U = T^{-\frac{1}{2}} \{u_j\}_{j=0}^{T-1}$. Also, let $Z = \eta T^{-\frac{1}{2}} \{\sum_{k \geq 0} W^k \varepsilon_{j-k}\}_{j=0}^{T-1}$ the concatenated noise vectors, with $\varepsilon_t \sim \mathcal{M}(0, I_n)$. With these notations and normalization, we have $X = \sqrt{T}(A+Z)$, where A and Z are expected to have operator norm of order $O(1)$ with respect to $n, T \rightarrow \infty$ as per Assumption 3 and thus should $\frac{1}{2} X X^T$.

For $\gamma > 0$, denoting $Q_\gamma = (\frac{1}{2} X X^T + \gamma I_n)^{-1}$, our objective is to obtain an approximation of Q_γ in the sense of the equivalence \Leftrightarrow using the so-called *Gaussian method* introduced by Pastur in (Pastur and Šerbina, 2011). This method consists in two ingredients: (i) an integration by parts formula for Gaussian random variables (also called Stein's lemma) that stipulates that, for $x \sim \mathcal{N}(0, 1)$ and a polynomially bounded differentiable f , $E[xf(x)] =$

$E[f'(x)]$, and (ii) concentration inequalities or moment based bounds (such as the Nash-Poincaré inequality) to control small terms. The idea here is to expand terms of the type $E[|\varepsilon_{it}|_k |Q_{it}|_k]$ using the Gaussian integration by parts formula in order to retrieve an implicit *but deterministic* expression for Q_{it} , up to small random terms. Then, thanks to concentration or moment bounds, the aforementioned small terms are shown to vanish at a sufficient speed to ensure almost sure convergence of Q_{it} to the deterministic solution of the implicit equation in the sense of the equivalence \leftrightarrow .

We start by noticing that $Q_{it} = \frac{1}{\gamma} I_n - \frac{1}{\gamma} X^T X^T Q_{it}$, a relation often referred to as the *resolvent identity*. This allows one to write $E[Q_{it}]$ as a function of $E[XX^T Q_{it}]$ which lends itself to the integration by parts approach since X is a linear function of the Gaussian variables $\{\varepsilon_{it}\}$.

In what follows, for readability, we shall denote $Q = Q_{it}$ (and thus $Q_{it} = [Q_{it}]_{ij}$) and $\varepsilon_{it} = \{\varepsilon_{it}\}$. Then we have

$$E[Q_{it}] = \frac{1}{\gamma} \delta_{ij} - \frac{1}{\gamma} \underbrace{\left(E[[ZZ^T Q]_{ij}] \right)}_{(I)} + \underbrace{E[[ZA^T Q]_{ij}]}_{(II)} + \underbrace{E[[AZ^T Q]_{ij}]}_{(III)} + \underbrace{E[[AA^T Q]_{ij}]}_{(IV)}. \quad (11)$$

Each of the four bracketed terms needs be treated independently. Note first that term (IV) is simply $\sum_k [AA^T]_{kk} E[Q_{it}]$ and is thus treated similar to $E[Q_{it}]$ itself. It then remains to handle terms (I)–(III). Before handling each term, let us first introduce a few elementary results of constant use in what follows. First, by a mere development, we have

$$Z_{ab} = \frac{\eta}{\sqrt{T}} \sum_{k \geq 0} \sum_{p=1}^n [W^k]_{\sigma^p \varepsilon_{i,b-k}}$$

from which

$$\frac{\partial Z_{ab}}{\partial \varepsilon_{it}} = \frac{\eta}{\sqrt{T}} \sum_{k \geq 0} \sum_{p=1}^n \delta_{\sigma^p \varepsilon_{i,b-k}} [W^k]_{\sigma^p}. \quad (12)$$

Expanding X in the expression of Q and using $\partial Q = -Q(\partial Q^{-1})Q$, we then find

$$\frac{\partial Q_{mi}}{\partial \varepsilon_{it}} = -\frac{\eta}{\sqrt{T}} \sum_{p=1}^n \delta_{i \leq p} \left([Q(Z+A)]_{mp} [W^{p-1}]^T Q \right]_{ij} + \left[(Z+A)^T Q \right]_{ji} [QW^{p-1}]_{mi}. \quad (13)$$

It is important at this point to bring some insight from random matrix theory. If ε_{it} were a *complex* rather than real standard Gaussian random variable, the second term in the right-hand side parenthesis would not have appeared. Since first order deterministic equivalents (which is what we are proceeding to here) are usually valid irrespective of the i.i.d. distribution (real or complex) of the ε_{it} 's, it is expected that this second term will lead to vanishing terms in what follows.

With these preliminary results and this remark in mind, we can tackle the calculus of terms (I)–(III) from (11). Let us first focus on term (I). Developing $E[[ZZ^T Q]_{ij}]$ as a function of the ε_{it} 's and applying the Gaussian integration-by-parts formula, we find

$$\begin{aligned} E[[ZZ^T Q]_{ij}] &= \eta \sum_{l=1}^T \sum_{m=1}^n \sum_{o=1}^n \sum_{k \geq 0} E[\varepsilon_{o,l-k} Z_{m,l} Q_{mj}] [W^k]_{io} \\ &= \eta \sum_{l=1}^T \sum_{m=1}^n \sum_{o=1}^n \sum_{k \geq 0} \left(E \left[\frac{\partial Z_{m,k}}{\partial \varepsilon_{o,l-k}} Q_{mj} \right] + E \left[\frac{\partial Q_{mj}}{\partial \varepsilon_{o,l-k}} Z_{ml} \right] \right) [W^k]_{io}. \end{aligned}$$

Substituting the derivatives by the forms (12) and (13), we obtain after full development and simplifications

$$\begin{aligned} E[[ZZ^T Q]_{ij}] &= \eta^2 \sum_{k \geq 0} E \left[[W^k (W^k)^T Q]_{ij} \right] \\ &\quad - \eta^2 \sum_{k \geq 0} \sum_{q=-k}^{T-1} E \left[[W^k (W^{k+q})^T Q]_{ij} \frac{1}{T} [Z^T (Z+A) \tilde{Q}]_{i,q+1} \right] + E[\zeta_{ij}^{[1]}] \end{aligned}$$

where we defined $\tilde{Q} = \tilde{Q}_{it} = (\frac{1}{T} X^T X + \gamma I_n)^{-1}$ and where the term $\zeta_{ij}^{[1]}$ arises from the development of the aforementioned second term in the parentheses of (13) and can be shown to satisfy $\zeta^{[1]} \leftrightarrow 0$. Similarly, addressing the term (II) in (11), we find

$$E[[ZA^T Q]_{ij}] = -\eta^2 \sum_{k \geq 0} \sum_{q=-k}^{T-1} E \left[\frac{1}{T} [A^T (Z+A) \tilde{Q}]_{i,q+1} [W^k (W^{k+q})^T Q]_{ij} \right] + E[\zeta_{ij}^{[2]}]$$

where again we can show that $\zeta^{[2]} \leftrightarrow 0$. Summing the approximations for (I) and (II), from the resolvent identity $(Z+A)^T (Z+A) \tilde{Q} = I_n - \gamma \tilde{Q}$, we find

$$\begin{aligned} E[[Z(Z+A)^T Q]_{ij}] &= \eta^2 \sum_{k \geq 0} E \left[[W^k (W^k)^T Q]_{ij} \right] \\ &\quad - \eta^2 \sum_{k \geq 0} \sum_{q=-k}^{T-1} E \left[[W^k (W^{k+q})^T Q]_{ij} \frac{1}{T} [I_n - \gamma \tilde{Q}]_{i,q+1} \right] + E[\zeta_{ij}^{[1]}] + \zeta_{ij}^{[2]}. \end{aligned}$$

Since $[I_n]_{i,q+1} = \delta_{q=0}$, the first right-hand side term cancels with the part of the second term involved with matrix $\frac{1}{T} I_n$, and we find

$$E[[Z(Z+A)^T Q]_{ij}] = \eta^2 \sum_{k \geq 0} \sum_{q=-k}^{T-1} \sum_{l=1}^n E \left[[W^k (W^{k+q})^T Q]_{ij} \frac{1}{T} \tilde{Q}_{l,q+1} \right] + E[\zeta_{ij}^{[1]}] + \zeta_{ij}^{[2]}. \quad (14)$$

Moving to term (III) in (11), since A is deterministic, we first find the interesting expression

$$E[[AZ^T Q]_{ij}] = -\eta^2 \sum_{k \geq 0} \sum_{q=-k}^{T-1} \sum_{l=1}^n E \left[\frac{1}{T} \text{tr}(W^k (W^{k+q})^T Q) (Z+A)^T Q]_{l,i+1} \right] + E[\zeta_{ij}^{[3]}] \quad (15)$$

with $\zeta^{[3]} \leftrightarrow 0$ from which immediately we get

$$E[[AZ^T Q]_{ij}] = -\eta^2 \sum_{k \geq 0} \sum_{q=-k}^{T-1} \sum_{l=1}^n E \left[\frac{1}{T} \text{tr}(W^k (W^{k+q})^T Q) A_{il} ((Z+A)^T Q]_{l,i+1} \right] + E[A \zeta_{ij}^{[3]}]$$

and we of course still have $A \zeta^{[3]} \leftrightarrow 0$.

We must discuss at this point the next key idea of the Gaussian method. In term (III), the right-hand side expectation is taken over the product of the trace $\frac{1}{T} \text{tr}(W^k (W^{k+q})^T Q)$ and of the quantity $A_{il} ((Z+A)^T Q]_{l,i+1}$. Writing $\frac{1}{T} \text{tr}(W^k (W^{k+q})^T Q) = E[\frac{1}{T} \text{tr}(W^k (W^{k+q})^T Q)] + (\frac{1}{T} \text{tr}(W^k (W^{k+q})^T Q) - E[\frac{1}{T} \text{tr}(W^k (W^{k+q})^T Q)])$, it can be shown, using Cauchy-Schwarz and the Nash-Poincaré inequalities (Pastur and Seppälä, 2011), along with the Borel-Cantelli lemma (Billingsley, 1995), that

$$\sum_{k \geq 0} \sum_{q=-k}^{T-1} \sum_{l=1}^n \left(\frac{1}{T} \text{tr}(W^k (W^{k+q})^T Q) - E \left[\frac{1}{T} \text{tr}(W^k (W^{k+q})^T Q) \right] \right) A_{il} ((Z+A)^T Q]_{l,i+1} \leftrightarrow 0$$

which unfolds from $\frac{1}{T} \text{tr}(W^k(W^{k+q})^\top Q)$ concentrating around its mean in the large n, T regime, a standard result of random matrix theory. The main non-classical difficulty in showing this result lies here in the fact that the summation over up to T values of the dummy variable q involves both terms in and outside the bracket. Nonetheless, since $\rho(W) < 1$, $\|W^q\|$ vanishes at exponential speed and thus only $O(\log(T))$ values of q are effectively playing a role. The aforementioned Nash–Poincaré inequality ensures a control of the residual terms with a $O(1/T^2)$ variance for each q -summand which can then be summed over the non-trivial values of q to bring a total variance bounded by $O(\log(T)/T^2)$, which is summable, and then allows for Borel–Cantelli to be applied.

The same reasoning applies to the main expectation in the expression of $(I) + (II)$, where here the term that concentrates around its mean is $\frac{1}{T} \sum_{j=1}^{T-\eta} \bar{Q}_{i, q+\eta}$, which is more easily seen as $\frac{1}{T} \text{tr}(J^\eta \bar{Q})$.

The relation (15) in itself is quite instructive. Indeed, with the previous remark on the concentration of $\frac{1}{T} \text{tr}(W^k(W^{k+q})^\top Q)$, we may break the right-hand expectation as well as the term $(Z+A)^\top Q$ into $Z^\top Q + A^\top Q$ to retrieve a connection between left- and right-hand sides. Precisely, we find that

$$\begin{aligned} & \left[\left(I_T + \eta^2 \sum_{k \geq 0} \sum_{q=-k}^{T-1} \mathbb{E} \left[\frac{1}{T} \text{tr}(W^k W^{k+q})^\top Q \right] \right) \mathbb{E}[X^\top Q] \right]_{ij} \\ &= -\eta^2 \sum_{k \geq 0} \sum_{q=-k}^{T-1} \mathbb{E} \left[\frac{1}{T} \text{tr}(W^k(W^{k+q})^\top Q) \right] \mathbb{E}[J^\eta A^\top Q]_{i,j} + o(1) \end{aligned}$$

where we used $[B]_{|q|+i,j} = [J^\eta B]_{i,j}$. Remark now that

$$\begin{aligned} \sum_{k \geq 0} \sum_{q=-k}^{T-1} \frac{1}{T} \text{tr}(W^k(W^{k+q})^\top Q) J^q &= \sum_{k \geq 0} \left\{ \frac{1}{T} \text{tr}(W^{k+(b-a)^+} (W^{k+(a-b)^+})^\top Q) \right\}_{a,b=1}^T \\ &= \left\{ \frac{1}{T} \text{tr}(S_{a-b} Q) \right\}_{a,b=1}^T. \end{aligned}$$

Denoting $\bar{R} = \mathbb{E}[\{\frac{1}{T} \text{tr}(S_{a-b} Q)\}_{a,b=1}^T]$ and using concentration arguments (Nash–Poincaré inequality in particular) entails

$$Z^\top Q \leftrightarrow -\eta^2 (I_T + \eta^2 \bar{R})^{-1} \bar{R} A^\top Q. \quad (16)$$

From the definition of the equivalence relation \leftrightarrow , this entails

$$AZ^\top Q \leftrightarrow -\eta^2 A (I_T + \eta^2 \bar{R})^{-1} \bar{R} A^\top Q. \quad (17)$$

Similarly, recalling (14), we have

$$\begin{aligned} Z(Z+A)^\top Q &\leftrightarrow \eta^2 \gamma \sum_{k \geq 0} \sum_{q=-k}^{T-1} \text{tr}(J^\eta \bar{Q}) W^k (W^{k+q})^\top Q \\ &= \eta^2 \gamma \sum_{q=-\infty}^{\infty} \frac{1}{T} \text{tr}(J^\eta \bar{Q}) S_q Q. \end{aligned}$$

We may then define $\bar{R} = \sum_{q=-\infty}^{\infty} \mathbb{E}[\frac{1}{T} \text{tr}(J^\eta \bar{Q}) S_q]$. Added to (17) and $AA^\top Q$, this is

$$(Z+A)(Z+A)^\top Q \leftrightarrow -\eta^2 \gamma \bar{R} - \eta^2 A (I_T + \eta^2 \bar{R})^{-1} \bar{R} A^\top Q + AA^\top Q.$$

With $AA^\top = A(I_T + \eta^2 \bar{R})^{-1} (I_T + \eta \bar{R}) A^\top$ and $(Z+A)(Z+A)^\top Q = I_n - \gamma Q$, this further reads

$$Q \leftrightarrow \frac{1}{\gamma} I_n - \eta^2 \bar{R} Q - \frac{1}{\gamma} A (I_T + \eta^2 \bar{R})^{-1} A^\top Q.$$

which, after gathering the factors of Q together, finally gives the first identity

$$Q \leftrightarrow \frac{1}{\gamma} \left(I_n + \eta^2 \bar{R} + \frac{1}{\gamma} A (I_T + \eta^2 \bar{R})^{-1} A^\top \right)^{-1}. \quad (18)$$

To pursue our investigation, we need to proceed to the same development for the matrix \bar{Q} which appears in the definition of \bar{R} . The idea is to express \bar{Q} under a form involving Q itself, then closing the loop. The analysis is extremely similar to that of Q and it is not surprising (from the symmetry between Q and \bar{Q}) to finally obtain

$$\bar{Q} \leftrightarrow \frac{1}{\gamma} \left(I_T + \eta^2 \bar{R} + \frac{1}{\gamma} A^\top (I_n + \eta^2 \bar{R})^{-1} A \right)^{-1}. \quad (19)$$

At this point, however, both \bar{R} and \bar{Q} are non explicit quantities that depend on the statistics of Q and \bar{Q} . From (18), we get that, for each a, b ,

$$\frac{1}{T} \text{tr}(S_{a-b} Q) \leftrightarrow \frac{1}{\gamma} \frac{1}{T} \text{tr} S_{a-b} \left(I_n + \eta^2 \bar{R} + \frac{1}{\gamma} A (I_T + \eta^2 \bar{R})^{-1} A^\top \right)^{-1}$$

and this relation is shown to be uniform across a, b , as it involves only $O(\log(T))$ non-trivial coefficients. To freely identify \bar{R} with $\{\frac{1}{\gamma} \text{tr} S_{a-b} (I_n + \eta^2 \bar{R} + \frac{1}{\gamma} A (I_T + \eta^2 \bar{R})^{-1} A^\top)^{-1}\}_{a,b=1}^n$, one may ensure that the difference between both matrices vanishes in spectral norm almost surely (here the relation \leftrightarrow may not be enough).⁴ Here the result holds true because both $\{\frac{1}{T} \text{tr}(S_{a-b} Q)\}_{a,b=1}^n$ and \bar{R} are Toeplitz matrices with exponentially decaying coefficients away from the main diagonal. Hence, we may essentially see each matrix as the sum of a circulant matrix and of a matrix with $O(\log(T))$ non-vanishing upper-right and lower-left entries (see (Gray, 2006) for such a construction). Circulant matrices being diagonalizable in the Fourier basis with eigenvalues equal to the Fourier transform of the concatenated first column and row, that the difference in spectral norm vanishes boils down to the convergence of the difference between these Fourier transforms, which is easily obtained through the joint entry convergence and exponential decrease. As for the remaining corner entries, being of $\log(T)$ number, we deal here with the difference in spectral norm of small rank matrices, which is obtained by direct uniform convergence. As such, generally speaking, if the entries of a Toeplitz matrix with exponentially vanishing profile converge jointly to given limits, then the limiting Toeplitz matrix is equivalent in the spectral norm sense.

Similarly, to identify \bar{R} with $\sum_{q \neq 0} \frac{1}{\gamma} \text{tr}(J^q (I_T + \eta^2 \bar{R} + \frac{1}{\gamma} A^\top (I_n + \eta^2 \bar{R})^{-1} A)^{-1}) S_q$, we need to show the spectral norm difference of these matrices vanishes almost surely. This is here obtained from the uniform convergence across the $O(\log(T))$ first trace coefficients (say for all $|q| \leq C \log(T)$) and from the corresponding exponentially vanishing spectral norm of S_q .

All said, we may then define $\bar{R}_\gamma, \bar{R}_{\bar{\gamma}}, \bar{Q}_\gamma,$ and $\bar{Q}_{\bar{\gamma}}$ as in Theorem 2 and the results above ensure that $Q_\gamma \leftrightarrow \bar{Q}_\gamma$ and $\bar{Q}_{\bar{\gamma}} \leftrightarrow \bar{Q}_{\bar{\gamma}}$.

4. A typical counter-example is the case of $Z \in \mathbb{R}^{n \times T}$ with i.i.d. zero mean and unit variance entries for which $\{\frac{1}{T} Z Z^\top\}_{ab} \rightarrow \delta_{a-b}$ uniformly over a, b while clearly $(\frac{1}{T} Z Z^\top + \gamma I_n)^{-1} \not\rightarrow (1 + \gamma)^{-1} I_T$.

Remark 22 (Result without washout period) *Theorem 2 assumes an infinite noise time series $(\dots, \varepsilon_{-1}, \varepsilon_0, \varepsilon_1, \dots)$. One might have alternatively considered a scenario without washout period, that is, with $x_{-1} = 0$ and first time instant being $t = 0$. In this case, Theorem 2 remains valid but for the following updated expressions of R_γ and \bar{R}_γ*

$$R_\gamma = \left\{ \sum_{k=0}^{\max(i,j)-1} \frac{1}{T} \text{tr } W^{k+(j-i)^+} (W^{k+(i-j)^+})^T Q_\gamma \right\}_{i,j=1}^T$$

$$\bar{R}_\gamma = \sum_{q=-(T-1)}^{T-1} \frac{1}{T} \text{tr} \left(J^q \bar{Q}_\gamma \right) \sum_{k=0}^{T-1-|q|} W^{k+(-q)^+} (W^{k+q^+})^T.$$

In particular, R_γ is no longer Toeplitz. Nonetheless the non-Toeplitz behavior is essentially concentrated in the top-left corner of size $O(\log(T))$ since the remainder of the matrix behaves essentially as Toeplitz (for $i, j \geq C \log(T)$ for some large enough constant C). This modification may alter the behavior of the associated train and test MSE, especially if r and \hat{r} concentrate their energy in their first entries.

Appendix B. Proof of Theorem 9

The first part of Theorem 9 is directly obtained from (16) along with $Q_\gamma \leftrightarrow \bar{Q}_\gamma$. Indeed, from these relations, we have

$$Q_\gamma \frac{1}{\sqrt{T}} X = Q_\gamma Z + Q_\gamma A \leftrightarrow -\eta^2 Q_\gamma A \bar{R}_\gamma (I_T + \eta^2 R_\gamma)^{-1} + Q_\gamma A$$

$$= \bar{Q}_\gamma A (I_T + \eta^2 R_\gamma)^{-1}.$$

The proof of the second part of Theorem 9 is not as straightforward as it involves twice the matrix Q_γ and thus results from Theorem 2 cannot be immediately applied. To handle this term, first write

$$\frac{1}{T} X^T Q_\gamma B Q_\gamma X = \underbrace{Z^T Q_\gamma B Q_\gamma Z}_{(I)} + \underbrace{Z^T Q_\gamma B Q_\gamma A}_{(II)} + \underbrace{A^T Q_\gamma B Q_\gamma Z}_{(III)} + \underbrace{A^T Q_\gamma B Q_\gamma A}_{(IV)}. \quad (20)$$

Since B is assumed symmetric, (III) is the transposed version of (II), so that only one of the two needs be studied.

Similar to Appendix A, we shall from now on simply write Q_γ as \bar{Q}_γ , \bar{Q}_γ as \bar{Q} , etc. We start by addressing term (I). We use again the Gaussian tools centered around the Gaussian integration by parts formula. We shall also benefit from the results of Theorem 2. Since B is deterministic, it needs not be included early in calculations so we merely start by evaluating, for given indices i, j, k, l ,

$$\begin{aligned} \mathbb{E} \left[Z^T Q_{ij} [QZ]_{kl} \right] &= \sum_{m, m', p, p' = 1}^n \sum_{q, q' \geq 0} \eta^2 \mathbb{E} \left[\varepsilon_{p,i} \varepsilon_{p',l} \varepsilon_{q,j} Q_{km} Q_{km'} \right] [W^q]_{mp} [W^{q'}]_{m'p'} \\ &= \sum_{m, m', p, p' = 1}^n \sum_{q, q' \geq 0} \eta^2 \mathbb{E} \left[\frac{\partial (\varepsilon_{p,i} \varepsilon_{p',l} Q_{km} Q_{km'})}{\partial \varepsilon_{p',l} \varepsilon_{q'}} \right] [W^q]_{mp} [W^{q'}]_{m'p'} \end{aligned}$$

where the second line follows from the Gaussian integration-by-parts formula. Developing the derivative based on (13) and on the fact that $\partial \varepsilon_{ab} / \partial \varepsilon_{cd} = \delta_{ac} \delta_{bd}$, we get after

$$\begin{aligned} \mathbb{E} \left[Z^T Q_{ij} [QZ]_{kl} \right] &= \eta^2 \sum_{q, q' \geq 0} \mathbb{E} \left[\frac{1}{T} Q W^q (W^{q'})^T Q_{ij} \delta_{i-q,l-q'} \right] \\ &= \eta^3 \sum_{q, q' \geq 0} \sum_{s=1}^T \mathbb{E} \left[\frac{1}{\sqrt{T}} \varepsilon^T (W^q)^T Q (Z + A)_{i-q,s} \frac{1}{T} [Q W^{q'} (W^{q'+s-1})^T Q]_{kj} \right] \\ &\quad - \eta^3 \sum_{q, q' \geq 0} \sum_{s=1}^T \mathbb{E} \left[\frac{1}{\sqrt{T}} \varepsilon^T (W^q)^T Q_{i-q,j} [Q (Z + A)]_{k,s} \frac{1}{T} \text{tr} (W^{q'} (W^{q'+s-1})^T Q) \right] \\ &\quad + \mathbb{E} [\zeta_{ij,kl}^{(1)}] \end{aligned}$$

simplification

$$\begin{aligned} &= \eta^2 \bar{G}_{ij} - \eta^2 \mathbb{E} \left[Z^T Q (Z + A) \bar{G} (I_T + \eta^2 \bar{R})^{-1} - \eta^2 Z^T Q B Q A \bar{R} (I_T + \eta^2 \bar{R})^{-1} \right] \\ &= \eta^2 \bar{G}_{ij} - \eta^2 \mathbb{E} \left[\frac{1}{T} \text{tr} \left(B Q W^{k+(j-i)^+} (W^{k+(i-j)^+})^T Q \right) \right]. \end{aligned} \quad (21)$$

for some $\zeta_{ij,kl}^{(1)} \leftrightarrow 0$ (arising from terms consistent with the remark following (13) in Appendix A) and where $\varepsilon = \{\varepsilon_{ij}\}_{j=1}^{n \times T}$. Inserting $B_{j,k}$, summing over j and k , we obtain after simplifications

$$\mathbb{E} \left[Z^T Q B Q Z_{ij} \right] = \eta^2 \bar{G}_{ii} - \eta^2 \mathbb{E} \left[Z^T Q (Z + A) \bar{G} [i] \right] - \eta^2 \mathbb{E} \left[Z^T Q B Q (Z + A) \bar{R} [i] \right] + o(1)$$

where \bar{R} was introduced in Appendix A and we defined \bar{G} the matrix with

$$\bar{G}_{ij} = \sum_{k \geq 0} \mathbb{E} \left[\frac{1}{T} \text{tr} \left(B Q W^{k+(j-i)^+} (W^{k+(i-j)^+})^T Q \right) \right].$$

Gathering the terms in $Z^T Q B Q Z$ together along with concentration arguments, we finally obtain

$$Z^T Q B Q Z \leftrightarrow \eta^2 \bar{G} (I_T + \eta^2 \bar{R})^{-1} - \eta^2 Z^T Q (Z + A) \bar{G} (I_T + \eta^2 \bar{R})^{-1} - \eta^2 Z^T Q B Q A \bar{R} (I_T + \eta^2 \bar{R})^{-1}.$$

In the right-hand side formulation, the second term can be approximated from the results of Theorem 2 as well as the first part of Theorem 9; indeed, note from $(Z + A)^T Q (Z + A) = Q (Z + A)^T (Z + A) = I_T - \gamma \bar{Q}$ that $Z^T Q (Z + A) = I_T - \gamma \bar{Q} - A^T Q (Z + A)$, so that

$$\begin{aligned} Z^T Q B Q Z &\leftrightarrow \eta^2 \gamma \bar{Q} \bar{G} (I_T + \eta^2 \bar{R})^{-1} + \eta^2 A^T \bar{Q} A (I_T + \eta^2 \bar{R})^{-1} \bar{G} (I_T + \eta^2 \bar{R})^{-1} \\ &\quad - \eta^2 Z^T Q B Q A \bar{R} (I_T + \eta^2 \bar{R})^{-1}. \end{aligned} \quad (22)$$

In this expression, the last right-hand side term still involves $Z^T Q B Q A$, yet to be characterized. This is the objective of the next step, which coincides with the study of the term (II) in (20).

Following the derivation of term (I), terms (II) and (III) are easily obtained (indeed, they somewhat boil down to (21) without the first right-hand side term and without the components $\varepsilon^T (W^q)^T$ in the subsequent terms). Precisely, all calculus made, we find that

$$Q B Q Z \leftrightarrow -\eta^2 Q (Z + A) \bar{G} - \eta^2 Q B Q (Z + A) \bar{R}$$

from which

$$Q B Q Z \leftrightarrow -\eta^2 Q (Z + A) \bar{G} (I_T + \eta^2 \bar{R})^{-1} - \eta^2 Q B Q A \bar{R} (I_T + \eta^2 \bar{R})^{-1}.$$

Again, the first right-hand side term is easily expressed by Theorem 2 and the first result of Theorem 9, from which

$$Q B Q Z \leftrightarrow -\eta^2 \bar{Q} A (I_T + \eta^2 \bar{R})^{-1} \bar{G} (I_T + \eta^2 \bar{R})^{-1} - \eta^2 Q B Q A \bar{R} (I_T + \eta^2 \bar{R})^{-1}. \quad (23)$$

but the second term now involves the quantity QBQ which is our next target. Since studying QBQ entails studying A^TQBQA , this shall provide us with the term (IV) in (20). To address QBQ , it suffices to estimate $E[Q_{ij}Q_{kl}]$; from the resolvent identity $Q_{ij} = \frac{1}{\gamma}\delta_{ij} - \frac{1}{\gamma}[\frac{1}{\gamma}XX^TQ]_{ij}$, this is developed as

$$E[Q_{ij}Q_{kl}] = -\frac{1}{\gamma} \left(E[[Z^T Z]Q_{ij}Q_{kl}] + E[[ZA^T Q]_{ij}Q_{kl}] + E[[AZ^T Q]_{ij}Q_{kl}] + E[[AA^T Q]_{ij}Q_{kl}] \right) + \frac{1}{\gamma}\delta_{ij}E[Q_{kl}].$$

The deterministic equivalent for $E[Q_{kl}]$ is already known, and we are then left to evaluate the first four terms, some of which can be retrieved from previous calculus. Developing each term, integrating the previously developed equivalents, while introducing the matrix B and summing, after some tedious calculus, we finally obtain

$$QBQ \leftrightarrow \frac{1}{\gamma}B\bar{Q} + \frac{\eta^2}{\gamma}A(I_T + \eta^2\bar{R})^{-1}\bar{G}(I_T + \eta^2\bar{R})^{-1}A^T\bar{Q} - \eta^2\bar{R}QBQ + \frac{1}{\gamma}\bar{G}\bar{Q} - \frac{1}{\gamma}A(I_T + \eta^2\bar{R})^{-1}A^TQBQ$$

where we introduced the notation

$$\bar{G} = \sum_{q=-\infty}^{\infty} \eta^2 E \left[\frac{1}{T} \text{tr} \left(J^q(A+Z)^T QBQ(Z+A) \right) \sum_{k \geq 0} W^{k+(-q)^+} (W^{k+q^+})^T \right].$$

Gathering all terms proportional to QBQ , we finally obtain

$$QBQ \leftrightarrow \bar{Q}(B + \bar{G})\bar{Q} + \eta^2\bar{Q}A(I_T + \eta^2\bar{R})^{-1}\bar{G}(I_T + \eta^2\bar{R})^{-1}A^T\bar{Q}. \quad (24)$$

Substituting (24) in (23), then substituting the result in (22), we may now completely characterize $\frac{1}{T}X^TQBQX$ (after simplification) as

$$\frac{1}{T}X^TQBQX \leftrightarrow \eta^2\gamma^2\bar{Q}\bar{G}\bar{Q} + (I_T + \eta^2\bar{R})^{-1}A^T\bar{Q}[\bar{Q}(B + \bar{G})\bar{Q}A(I_T + \eta^2\bar{R})^{-1}].$$

It remains to evaluate $E[\frac{1}{T} \text{tr}(J^q(A+Z)^T QBQ(Z+A))]$ in the expression of \bar{G} . For this, we shall exploit the fact that $A = MU$ which, since M has columns of exponentially decreasing norm, can be considered as a matrix of rank “essentially of order $O(\log(T))$ ”; that is, while being full rank, A can be well approximated in spectral norm by the product $\bar{M}\bar{U}$ of the first $O(\log(T))$ columns \bar{M} of M and first $O(\log(T))$ rows \bar{U} of U . This entails that, in the deterministic approximation for $(A+Z)^TQBQ(Z+A)$, only the terms not involving a product with A or A^T will remain after taking the normalized trace. And thus we get, after development and simplification

$$\left\| \bar{G} - \sum_{q=-\infty}^{\infty} \gamma^2 \eta^2 \frac{1}{T} \text{tr} \left(J^q \bar{Q} \bar{G} \bar{Q} \right) \sum_{k \geq 0} W^{k+(-q)^+} (W^{k+q^+})^T \right\| \rightarrow 0.$$

It then suffices to use concentration identities and the results of Appendix A to finally substitute \bar{R} with \bar{R} , \bar{R} with \bar{R} , and \bar{G} , \bar{G} with G and \bar{G} , respectively. This concludes the proof of Theorem 9.

Remark 23 (On the speed of convergence) To better appreciate the interplay between η^2 and n, T , note that all convergences discussed in Appendices A–B involve either quadratic forms of the type $a^T Q a$ for $Q \in \mathbb{R}^{n \times n}$ a random matrix based on some $\varepsilon \in \mathbb{R}^{n \times T}$, matrix with independent entries, or normalized traces $\frac{1}{n} \text{tr} Q$. It is a standard central limit result in random matrix theory that the former quadratic form $a^T Q a$ fluctuates at speed $O(n^{-\frac{1}{2}})$, that is, $\text{var}(a^T Q a) = O(n^{-1})$, and that normalized traces fluctuate at the faster speed $O(n^{-1})$. As such, the results of Theorems 2–9 and Proposition 4–10 can be trusted with high probability within a $O(n^{-\frac{1}{2}})$ error bound.

With respect to η^2 , the bounds between random quantities and deterministic equivalents, say Q and \bar{Q} , are proportional to $1/\eta^2$. This is why η^2 is assumed fixed and not decaying in our results. Nonetheless, as both bounds in n and η^2 multiply, it is expected that convergence is maintained in general so long that $n^{-\frac{1}{2}}/\eta^2 \rightarrow 0$, i.e., when $\eta^2 \gg n^{-\frac{1}{2}}$.

References

- Z. D. Bai and J. W. Silverman. *Spectral analysis of large dimensional random matrices*. Springer Series in Statistics, New York, NY, USA, second edition, 2009.
- Marwa Banna and Florence Merlevede. Limiting spectral distribution of large sample covariance matrices associated with a class of stationary processes. *Journal of Theoretical Probability*, pages 1–39, 2013.
- P. Biane. Free probability for probabilists. *Quantum Probability Communications*, 11:55–71, 2003.
- P. Billingsley. *Probability and Measure*. John Wiley and Sons, Inc., Hoboken, NJ, third edition, 1995.
- C. M. Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.
- Surya Ganguli, Dongsung Huh, and Haim Sompolinsky. Memory traces in dynamical systems. *Proceedings of the National Academy of Sciences*, 105(48):18970–18975, 2008.
- Leon Glass and Michael C. Mackey. A simple model for phase locking of biological oscillators. *Journal of Mathematical Biology*, 7(4):339–352, 1979.
- R. M. Gray. Toeplitz and circulant matrices: a review. *Foundations and Trends in Communications and Information Theory*, 2(3), 2006.
- W. Hachem, O. Khorunzhy, P. Loubaton, J. Najim, and L. A. Pastur. A new approach for capacity analysis of large dimensional multi-antenna channels. *IEEE Transactions on Information Theory*, 54(9):3987–4004, 2008.
- Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Stew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1):489–501, 2006.
- H. Jaeger. *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the “echo state network” approach*. GMD-Forschungszentrum Informationstechnik, 2005.
- H. Jaeger and H. Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80, 2004.
- Herbert Jaeger. *Short term memory in echo state networks*. GMD-Forschungszentrum Informationstechnik, 2001.

- M. Lukševičius and H. Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.
- M. Oztuik, C. Mustafa, D. Xu, and J. C. Principe. Analysis and design of echo state networks. *Neural Computation*, 19(1):111–138, 2007.
- L. Pastur and M. Šerbina. *Eigenvalue distribution of large random matrices*. American Mathematical Society, 2011.
- A. Rodan and P. Tiño. Minimum complexity echo state network. *Neural Networks, IEEE Transactions on*, 22(1):131–144, 2011.
- Tobias Strauss, Wolf Wüstlich, and Roger Labahn. Design strategies for weight matrices of echo state networks. *Neural computation*, 24(12):3246–3276, 2012.
- Peter Tiño and Ali Rodan. Short term memory in input-driven linear dynamical systems. *Neurocomputing*, 112:58–63, 2013.
- T. Toyozumi and L. F. Abbott. Beyond the edge of chaos: Amplification and temporal integration by recurrent networks in the chaotic regime. *Physical Review E*, 84(5):051908, 2011.
- D. Venstraeten, J. Dambre, X. Dutoit, and B. Schrauwen. Memory versus non-linearity in reservoirs. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–8. IEEE, 2010.
- Y. Xue, L. Yang, and S. Haykin. Decoupled echo state networks with lateral inhibition. *Neural Networks*, 20(3):365–376, 2007.

On the Consistency of the Likelihood Maximization Vertex Nomination Scheme: Bridging the Gap Between Maximum Likelihood Estimation and Graph Matching

Vince Lyzinski

*Department of Applied Math and Statistics
Johns Hopkins University
Baltimore, MD 21218-2608, USA*

VLYZINS1@JHU.EDU

Keith Levin

*Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218-2608, USA*

KLEVIN@JHU.EDU

Donniell E. Fishkind

*Department of Applied Math and Statistics
Johns Hopkins University
Baltimore, MD 21218-2608, USA*

DEF@JHU.EDU

CEP@JHU.EDU

Carey E. Priebe

*Department of Applied Math and Statistics
Johns Hopkins University
Baltimore, MD 21218-2608, USA*

Editor: Edo Airoldi

Abstract

Given a graph in which a few vertices are deemed interesting a priori, the vertex nomination task is to order the remaining vertices into a nomination list such that there is a concentration of interesting vertices at the top of the list. Previous work has yielded several approaches to this problem, with theoretical results in the setting where the graph is drawn from a stochastic block model (SBM), including a vertex nomination analogue of the Bayes optimal classifier. In this paper, we prove that maximum likelihood (ML)-based vertex nomination is consistent, in the sense that the performance of the ML-based scheme asymptotically matches that of the Bayes optimal scheme. We prove theorems of this form both when model parameters are known and unknown. Additionally, we introduce and prove consistency of a related, more scalable restricted-focus ML vertex nomination scheme. Finally, we incorporate vertex and edge features into ML-based vertex nomination and briefly explore the empirical effectiveness of this approach.

Keywords: vertex nomination, graph matching, graph inference, stochastic block model, graph mining

1. Introduction and Background

Graphs are a common data modality, useful for modeling complex relationships between objects, with applications spanning fields as varied as biology (Jeong et al., 2001; Bullmore and Sporns, 2009), sociology (Wasserman and Faust, 1994), and computer vision (Foggia et al., 2014; Kandel et al., 2007), to name a few. For example, in neuroscience, vertices may be neurons and edges adjoin pairs of neurons that share a synapse (Bullmore and Sporns, 2009);

in social networks, vertices may correspond to people and edges to friendships between them (Carrington et al., 2005; Yang and Leskovec, 2015); in computer vision, vertices may represent pixels in an image and edges may represent spatial proximity or multi-resolution mappings (Kandel et al., 2007). In many useful networks, vertices with similar attributes form densely-connected communities compared to vertices with highly disparate attributes, and uncovering these communities is an important step in understanding the structure of the network. There is an extensive literature devoted to uncovering this community structure in network data, including methods based on maximum modularity (Newman and Girvan, 2004; Newman, 2006b), spectral partitioning algorithms (Luxburg, 2007; Rohe et al., 2011; Sussman et al., 2012; Lyzinski et al., 2014b), and likelihood-based methods (Bickel and Chen, 2009), among others.

In the setting of *vertex nomination*, one community in the network is of particular interest, and the inference task is to order the vertices into a nomination list with those vertices from the community of interest concentrating at the top of the list. See Marchette et al. (2011); Coppersmith and Priebe (2012); Coppersmith (2014); Fishkind et al. (2015) and the references contained therein for a review of the relevant vertex nomination literature. Vertex nomination is a semi-supervised inference task, with example vertices from the community of interest—and, ideally, also examples not from the community of interest—being leveraged in order to create a nomination list. In this way, the vertex nomination problem is similar to the problem faced by personalized recommender systems (see, for example, Resnick and Varian, 1997; Ricci et al., 2011), where, given a training list of objects of interest, the goal is to arrange the remaining objects into a recommendation list with “interesting” objects concentrated at the top of the list. The main difference between the two inference tasks is that in vertex nomination the features of the data are encoded into the topology of a network, rather than being observed directly as features (though see Section 5 for the case where vertices are annotated with additional information in the form of features).

In this paper, we develop the notion of a consistent vertex nomination scheme (Definition 2). We then proceed to prove that the maximum-likelihood vertex nomination scheme of Fishkind et al. (2015) is consistent under mild model assumptions on the underlying stochastic block model (Theorem 6). In the process, we propose a new, efficiently executable solvable likelihood-based nomination scheme, the restricted-focus maximum-likelihood vertex-nomination scheme, $\mathcal{L}_R^{\text{ML}}$, and prove the analogous consistency result (Theorem 8). In addition, under mild model assumptions, we prove that both schemes maintain their consistency when the stochastic block model parameters are unknown and are estimated using the seed vertices (Theorems 9 and 10). In both cases, we show that consistency is possible even when the seeds are an asymptotically vanishing portion of the graph. Lastly, we show how both schemes can be easily modified to incorporate edge weights and vertex features (Section 5), before demonstrating the practical effect of our theoretical results on real and synthetic data (Section 6) and closing with a brief discussion (Section 7).

1.1 Notation

We say that a sequence of random variables $(X_n)_{n=1}^\infty$ converges almost surely to random variable X , written $X_n \rightarrow X$ a.s., if $\mathbb{P}[\lim_{n \rightarrow \infty} X_n = X] = 1$. We say a sequence of events $(A_n)_{n=1}^\infty$ occurs almost always almost surely (abbreviated a.a.s.) if with probability 1,

A_n^c occurs for at most finitely many n . By the Borel-Cantelli lemma, $\sum_{n=1}^{\infty} \mathbb{P}[A_n^c] < \infty$ implies $(A_n)_{n=1}^{\infty}$ a.a.s. We write G_n to denote the set of all (possibly weighted) graphs on n vertices. Throughout, without loss of generality, we will assume that the vertex set is given by $V = \{1, 2, \dots, n\}$. For a positive integer K , we will often use $[K]$ to denote the set $\{1, 2, \dots, K\}$. For a set V , we will use $\binom{V}{2}$ to denote the set of all pairs of distinct elements of V . That is, $\binom{V}{2} = \{u, v\} : u, v \in V, u \neq v\}$. For a function f with domain V , we write $f|_U$ to denote the restriction of f to the set $U \subset V$.

1.2 Background

Stochastic block model random graphs offer a theoretically tractable model for graphs with latent community structure (Rohle et al., 2011; Sussman et al., 2012; Bickel and Chen, 2009), and have been widely used in the literature to model community structure in real networks (Airoldi et al., 2008; Karrer and Newman, 2011). While stochastic block models can be too simplistic to capture the eccentricities of many real graphs, they have proven to be a useful tractable surrogate for more complicated networks (Airoldi et al., 2013; Olhede and Wolfe, 2014).

Definition 1 Let K and n be positive integers and let $\vec{n} = (n_1, n_2, \dots, n_K)^T \in \mathbb{R}^K$ be a vector of positive integers with $\sum_k n_k = n$. Let $b : [n] \rightarrow [K]$ and let $\Lambda \in [0, 1]^{K \times K}$ be symmetric. A G_n -valued random graph G is an instantiation of a (K, \vec{n}, b, Λ) conditional Stochastic Block Model, written $G \sim \text{SBM}(K, \vec{n}, b, \Lambda)$, if

- i. The vertex set V is partitioned into K blocks, V_1, V_2, \dots, V_K of cardinalities $|V_k| = n_k$ for $k = 1, 2, \dots, K$;
- ii. The block membership function $b : V \rightarrow [K]$ is such that for each $v \in V$, $v \in V_{b(v)}$;
- iii. The symmetric block communication matrix $\Lambda \in [0, 1]^{K \times K}$ is such that for each $\{v, u\} \in \binom{V}{2}$, there is an edge between vertices u and v with probability $\Lambda_{b(u), b(v)}$, independently of all other edges.

Without loss of generality, let V_1 be the block of interest for vertex nomination. For each $k \in [K]$, we further decompose V_k into $V_k = S_k \cup U_k$ (with $|S_k| = m_k$), where the vertices in $S := \cup_k S_k$ have their block membership observed *a priori*. We call the vertices in S *seed vertices*, and let $m = |S|$. We will denote the set of nonseed vertices by $U = \cup_k U_k$, and for all $k \in [K]$, let $u_k := n_k - m_k = |U_k|$ and $n - m = u = |U|$. Throughout this paper, we assume that the seed vertices S are chosen uniformly at random from all possible subsets of V of size m . The task in vertex nomination is to leverage the information contained in the seed vertices to produce a *nomination list* $\mathcal{L} : U \rightarrow [u]$ (i.e., an ordering of the vertices in U) such that the vertices in U_1 concentrate at the top of the list. We note that, strictly speaking, a nomination list \mathcal{L} is also a function of the observed graph G , a fact that we suppress for ease of notation. We measure the efficacy of a nomination scheme via *average precision*

$$\text{AP}(\mathcal{L}) = \frac{1}{u_1} \sum_{i=1}^u \frac{\sum_{j=1}^i \mathbb{1}\{\mathcal{L}^{-1}(j) \in U_1\}}{i}. \quad (1)$$

AP ranges from 0 to 1, with a higher value indicating a more effective nomination scheme: indeed, $\text{AP}(\mathcal{L}) = 1$ indicates that the first u_1 vertices in the nomination list are all from the block of interest, and $\text{AP}(\mathcal{L}) = 0$ indicates that none of the u_1 top-ranked vertices are from the block of interest. Letting $H_k = \sum_{j=1}^k 1/j$ denote the k -th harmonic number, with the convention that $H_0 = 0$, we can rearrange (1) as

$$\text{AP}(\mathcal{L}) = \sum_{i=1}^{u_1} \frac{H_{u_1} - H_{i-1}}{u_1} \mathbb{1}\{\mathcal{L}^{-1}(i) \in U_1\},$$

from which we see that the average precision is simply a convex combination of the indicators of correctness in the rank list, in which correctly placing an interesting vertex higher in the nomination list (i.e., with rank close to 1) is rewarded more than correctly placing an interesting vertex lower in the nomination list.

In Fishkind et al. (2015), three vertex nomination schemes are presented in the context of stochastic block model random graphs: the canonical vertex nomination scheme, \mathcal{L}^C , which is suitable for small graphs (tens of vertices); the maximum-likelihood vertex-nomination scheme, \mathcal{L}^{ML} , which is suitable for small to medium graphs (up to thousands of vertices); and the spectral partitioning vertex nomination scheme, \mathcal{L}^{SP} , which is suitable for medium to very large graphs (up to tens of millions of vertices). In the stochastic block model setting, the canonical vertex nomination scheme is provably optimal: under mild model assumptions, $\mathbb{E} \text{AP}(\mathcal{L}^C) \geq \mathbb{E} \text{AP}(\mathcal{L})$ for any vertex nomination scheme \mathcal{L} (Fishkind et al., 2015), where the expectation is with respect to a G_{n+n} -valued random graph G and the selection of the seed vertices. Thus, the canonical method is the vertex nomination analogue of the Bayes classifier, and this motivates the following definition:

Definition 2 Let $G \sim \text{SBM}(K, \vec{n}, b, \Lambda)$. With notation as above, a vertex nomination scheme \mathcal{L} is consistent if

$$\lim_{n \rightarrow \infty} |\mathbb{E} \text{AP}(\mathcal{L}^C) - \mathbb{E} \text{AP}(\mathcal{L})| = 0.$$

In our proofs below, where we establish the consistency of two nomination schemes, we prove a stronger fact, namely that $\text{AP}(\mathcal{L}) = 1$ a.a.s. We prefer the definition of consistency given in Definition 2 since it allows us to speak about the best possible nomination scheme even when the model is such that $\lim_{n \rightarrow \infty} \mathbb{E} \text{AP}(\mathcal{L}^C) < 1$.

In Fishkind et al. (2015), it was proven that under mild assumptions on the stochastic block model underlying G , we have

$$\lim_{n \rightarrow \infty} \mathbb{E} \text{AP}(\mathcal{L}^{\text{SP}}) = 1,$$

from which the consistency of \mathcal{L}^{SP} follows immediately. The spectral nomination scheme \mathcal{L}^{SP} proceeds by first K -means clustering the adjacency spectral embedding (Sussman et al., 2012) of G , and then nominating vertices based on their distance to the cluster of interest. Consistency of \mathcal{L}^{SP} is an immediate consequence of the fact that, under mild model assumptions on the underlying stochastic block model, K -means clustering of the adjacency spectral embedding of G perfectly clusters the vertices of G a.a.s. (Lyzinski et al., 2014b). Bickel and Chen (2009) proved that maximum-likelihood estimation provides consistent estimates of the model parameters in a more common variant of the conditional stochastic

is equivalent to finding

$$\hat{P} = \arg \min_{P \in \Pi_u} -\frac{1}{2} \operatorname{tr} \left(A^{(2,2)} P (B^{(2,2)})^T P^T \right) - \operatorname{tr} \left((A^{(1,2)})^T B^{(1,2)} P^T \right), \quad (5)$$

as we shall explain below.

With B defined as in (4), we define

$$\mathcal{Q} = \left\{ Q \in \Pi_u \text{ s.t. } (I_m \oplus Q) B (I_m \oplus Q)^T = B \right\}.$$

Define an equivalence relation \sim on Π_u via $P_1 \sim P_2$ iff there exists a $Q \in \mathcal{Q}$ such that $P_1 = P_2 Q$; i.e.,

$$(I_m \oplus P_1) B (I_m \oplus P_1)^T = (I_m \oplus P_2 Q) B (I_m \oplus P_2 Q)^T = (I_m \oplus P_2) B (I_m \oplus P_2)^T.$$

Let \hat{P}/\sim denote the set of equivalence classes of \hat{P} under equivalence relation \sim . Solving (2) is equivalent to solving (5) in that there is a one-to-one correspondence between \hat{b} and \hat{P}/\sim : for each $\phi \in \hat{b}$ there is a unique $P \in \hat{P}/\sim$ (with associated permutation σ) such that $\phi|_U = b|_U \circ \sigma$; and for each $P \in \hat{P}/\sim$ (with the permutation associated with $I_m \oplus P$ given by σ), it holds that $b \circ \sigma \in \hat{b}$.

2.1 The \mathcal{L}^{ML} Vertex Nomination Scheme

The maximum-likelihood vertex-nomination scheme proceeds as follows. First, the SGM algorithm (Fishkind et al., 2012; Lyzinski et al., 2014a) is used to approximately find an element of \hat{P} , which we shall denote by P . Let the corresponding element of \hat{b} be denoted by ϕ . For any $i, j \in V$ such that $\phi(i) \neq \phi(j)$, define $\phi_{i \leftrightarrow j} \in \mathcal{B}$ as

$$\phi_{i \leftrightarrow j}(v) = \begin{cases} \phi(i) & \text{if } v = j, \\ \phi(j) & \text{if } v = i, \\ \phi(v) & \text{if } v \neq i, j; \end{cases}$$

i.e., $\phi_{i \leftrightarrow j}$ agrees with ϕ except that i and j have their block memberships from ϕ switched in $\phi_{i \leftrightarrow j}$. For $i \in U$ such that $\phi(i) = 1$, define

$$\eta(i) := \left(\prod_{\substack{j \in U \text{ s.t.} \\ \phi(j) \neq 1}} \frac{\ell(\phi_{i \leftrightarrow j}, G)}{\ell(\phi, G)} \right)^{\frac{1}{n-1}},$$

where, for each $\psi \in \mathcal{B}$, the likelihood ℓ is given by

$$\ell(\psi, G) = \prod_{\{i,j\} \in \binom{V}{2}} A_{\psi(i), \psi(j)}^{A_{i,j}} (1 - A_{\phi(i), \phi(j)})^{1-A_{i,j}} \prod_{(i,j) \in S \times U} A_{\psi(i), \psi(j)}^{A_{i,j}} (1 - A_{b(i), b(j)})^{1-A_{i,j}}.$$

A low/high value of $\eta(i)$ is a measure of our confidence that i is/is not in the block of interest. For $i \in U$ such that $\phi(i) \neq 1$, define

$$\xi(i) := \left(\prod_{\substack{j \in U \text{ s.t.} \\ \phi(j)=1}} \frac{\ell(\phi_{i \leftrightarrow j}, G)}{\ell(\phi, G)} \right)^{\frac{1}{n-1}}.$$

A low/high value of $\xi(i)$ is a measure of our confidence that i is/is not in the block of interest. We are now ready to define the maximum-likelihood nomination scheme \mathcal{L}^{ML} :

$$\begin{aligned} (\mathcal{L}^{\text{ML}})^{-1}(1) &\in \arg \min \{ \eta(v) : \phi(v) = 1 \} \\ (\mathcal{L}^{\text{ML}})^{-1}(2) &\in \arg \min \{ \eta(v) : v \in U \setminus \{ (\mathcal{L}^{\text{ML}})^{-1}(1) \}, \phi(v) = 1 \} \\ &\vdots \\ (\mathcal{L}^{\text{ML}})^{-1}(u_1) &\in \arg \min \{ \eta(v) : v \in U \setminus \{ (\mathcal{L}^{\text{ML}})^{-1}(i) \}_{i=1}^{u_1-1}, \phi(v) = 1 \} \\ (\mathcal{L}^{\text{ML}})^{-1}(u_1+1) &\in \arg \max \{ \xi(v) : \phi(v) \neq 1 \} \\ (\mathcal{L}^{\text{ML}})^{-1}(u_1+2) &\in \arg \max \{ \xi(v) : v \in U \setminus \{ (\mathcal{L}^{\text{ML}})^{-1}(u_1+1) \}, \phi(v) \neq 1 \} \\ &\vdots \\ (\mathcal{L}^{\text{ML}})^{-1}(u) &\in \arg \max \{ \xi(v) : v \in U \setminus \{ (\mathcal{L}^{\text{ML}})^{-1}(i) \}_{i=u_1+1}^{u-1}, \phi(v) \neq 1 \} \end{aligned}$$

Note that in the event that an argmin (or argmax) above contains more than one element, the order in which these elements is nominated should be taken to be uniformly random.

Remark 5 In the event that Λ is unknown *a priori*, we can use the block memberships of the seeds S (assumed to be chosen uniformly at random from V) to estimate the edge probability matrix Λ as

$$\hat{\Lambda}_{k,\ell} = \frac{|\{i,j\} \in E \text{ s.t. } i \in S_k, j \in S_\ell\}|}{m_k m_\ell} \quad \text{for } k \neq \ell,$$

and

$$\hat{\Lambda}_{k,k} = \frac{|\{i,j\} \in E \text{ s.t. } i \in S_k, j \in S_k\}|}{\binom{m_k}{2}}.$$

The plug-in estimate \hat{B} of B , given by

$$\hat{B}_{i,j} := \log \left(\frac{\hat{\Lambda}_{b(i), b(j)}}{1 - \hat{\Lambda}_{b(i), b(j)}} \right),$$

can then be used in place of B in Eq. (5). If, in addition, \bar{n} is unknown, we can estimate the block sizes n_k as

$$\hat{n}_k = \frac{m_k \bar{n}}{m},$$

for each $k \in [K]$, and these estimates can be used to determine the block sizes in \hat{B} .

2.2 The $\mathcal{L}_R^{\text{ML}}$ Vertex Nomination Scheme

Graph matching is a computationally difficult problem, and there are no known polynomial time algorithms for solving the general graph matching problem for simple graphs. Furthermore, if the graphs are allowed to be weighted, directed, and loopy, then graph matching is equivalent to the NP-hard quadratic assignment problem. While there are numerous

efficient, approximate graph matching algorithms (see, for example, Vogelstein et al., 2014; Fishkind et al., 2012; Zaslavskiy et al., 2009; Fiori et al., 2013, and the references therein), these algorithms often lack performance guarantees.

Inspired by the restricted-focus seeded graph matching problem considered in Lyzinski et al. (2014a), we now define the computationally tractable restricted-focus maximum-likelihood nomination scheme $\mathcal{L}_R^{\text{ML}}$. Rather than attempting to quickly approximate a solution to the full graph matching problem as in Vogelstein et al. (2014); Fishkind et al. (2012); Zaslavskiy et al. (2009); Fiori et al. (2013), this approach simplifies the problem by ignoring the edges between unseeded vertices. An analogous restriction for matching simple graphs was introduced in Lyzinski et al. (2014a). We begin by considering the graph matching problem in Eq. (5). The objective function

$$-\frac{1}{2} \text{tr} \left(A^{(2,2)} P (B^{(2,2)})^\top P^\top \right) - \text{tr} \left((A^{(1,2)})^\top B^{(1,2)} P^\top \right)$$

consists of two terms: $-\frac{1}{2} \text{tr} (A^{(2,2)} P (B^{(2,2)})^\top P^\top)$, which seeks to align the induced sub-graphs of the nonseed vertices; and $-\text{tr} (A^{(1,2)} B^{(1,2)} P^\top)$, which seeks to align the induced bipartite subgraphs between the seed and nonseed vertices. While the graph matching objective function, Eq. (5), is quadratic in P , restricting our focus to the second term in Eq. (5) yields the following *linear assignment problem*

$$\tilde{P} = \arg \min_{P \in \Pi_u} - \text{tr} \left((A^{(1,2)})^\top B^{(1,2)} P^\top \right), \quad (6)$$

which can be efficiently and exactly solved in $O(u^3)$ time with the Hungarian algorithm (Kuhn, 1955; Jonker and Volgenant, 1987). We note that, exactly as was the case of \tilde{P} and \tilde{b} , finding \tilde{P} is equivalent to finding

$$\tilde{b} = \arg \max_{\phi \in \mathcal{B}} \sum_{(i,j) \in \mathcal{S} \times \mathcal{U}} A_{i,j} \log \left(\frac{\Lambda_{b(i),\phi(j)}}{1 - \Lambda_{b(i),\phi(j)}} \right),$$

in that there is a one-to-one correspondence between \tilde{b} and \tilde{P} / \sim .

The $\mathcal{L}_R^{\text{ML}}$ scheme proceeds as follows. First, the linear assignment problem, Eq. (6), is exactly solved using, for example, the Hungarian algorithm (Kuhn, 1955) or the path augmenting algorithm of Jonker and Volgenant (1987), yielding $P \in \tilde{P}$. Let the corresponding element of \tilde{b} be denoted by ϕ . For $i \in U$ such that $\phi(i) = 1$, define

$$\tilde{\eta}(i) := \left(\prod_{\substack{j \in U \text{ s.t.} \\ \phi(j) \neq 1}} \frac{\ell_R(\phi_{k+i,j}, G)}{\ell_R(\phi, G)} \right)^{\frac{1}{u-1}},$$

where, for each $\psi \in \mathcal{B}$, the *restricted likelihood* ℓ_R is defined via

$$\ell_R(\psi, G) = \prod_{(i,j) \in \mathcal{S} \times \mathcal{U}} \Lambda_{b(i),\psi(j)}^{A_{i,j}} (1 - \Lambda_{b(i),\psi(j)})^{1-A_{i,j}}.$$

As with \mathcal{L}^{ML} , a low/high value of $\tilde{\eta}(i)$ is a measure of our confidence that i is/is not in the block of interest. For $i \in U$ such that $\phi(i) \neq 1$, define

$$\tilde{\xi}(i) := \left(\prod_{\substack{j \in U \text{ s.t.} \\ \phi(j) = 1}} \frac{\ell_R(\phi_{k+i,j}, G)}{\ell_R(\phi, G)} \right)^{\frac{1}{u-1}}.$$

As before, a low/high value of $\tilde{\xi}(i)$ is a measure of our confidence that i is/is not in the block of interest. We are now ready to define $\mathcal{L}_R^{\text{ML}}$:

$$\begin{aligned} (\mathcal{L}_R^{\text{ML}})^{-1}(1) &\in \arg \min \{ \tilde{\eta}(v) : \phi(v) = 1 \} \\ (\mathcal{L}_R^{\text{ML}})^{-1}(2) &\in \arg \min \{ \tilde{\eta}(v) : v \in U \setminus \{ (\mathcal{L}_R^{\text{ML}})^{-1}(1) \}, \phi(v) = 1 \} \\ &\vdots \\ (\mathcal{L}_R^{\text{ML}})^{-1}(u_1) &\in \arg \min \{ \tilde{\eta}(v) : v \in U \setminus \{ (\mathcal{L}_R^{\text{ML}})^{-1}(i) \}_{i=1}^{u_1-1}, \phi(v) = 1 \} \\ (\mathcal{L}_R^{\text{ML}})^{-1}(u_1 + 1) &\in \arg \max \{ \tilde{\xi}(v) : \phi(v) \neq 1 \} \\ (\mathcal{L}_R^{\text{ML}})^{-1}(u_1 + 2) &\in \arg \max \{ \tilde{\xi}(v) : v \in U \setminus \{ (\mathcal{L}_R^{\text{ML}})^{-1}(u_1 + 1) \}, \phi(v) \neq 1 \} \\ &\vdots \\ (\mathcal{L}_R^{\text{ML}})^{-1}(u) &\in \arg \max \{ \tilde{\xi}(v) : v \in U \setminus \{ (\mathcal{L}_R^{\text{ML}})^{-1}(i) \}_{i=u_1+1}^{u-1}, \phi(v) \neq 1 \} \end{aligned}$$

Note that, as before, in the event that the argmin (or argmax) in the definition of $\mathcal{L}_R^{\text{ML}}$ contains more than one element above, the order in which these elements are nominated should be taken to be uniformly random.

Unlike \mathcal{L}^{ML} , the restricted focus scheme $\mathcal{L}_R^{\text{ML}}$ is feasible even for comparatively large graphs (up to thousands of nodes, in our experience). However, we will see in Section 6 that the extra information available to $\mathcal{L}_R^{\text{ML}}$ —the adjacency structure among the nonseed vertices—leads to superior precision in the \mathcal{L}^{ML} nomination lists as compared to $\mathcal{L}_R^{\text{ML}}$. We next turn our attention to proving the consistency of the \mathcal{L}^{ML} and $\mathcal{L}_R^{\text{ML}}$ schemes.

3. Consistency of \mathcal{L}^{ML} and $\mathcal{L}_R^{\text{ML}}$

In this section, we state theorems ensuring the consistency of the vertex nomination schemes \mathcal{L}^{ML} (Theorem 6) and $\mathcal{L}_R^{\text{ML}}$ (Theorem 8). For the sake of expository continuity, proofs are given in the Appendix. We note here that in these Theorems, the parameters of the underlying block model are assumed to be known *a priori*. In Section 4, we prove the consistency of \mathcal{L}^{ML} and $\mathcal{L}_R^{\text{ML}}$ in the setting where the model parameters are unknown and must be estimated, as in Remark 5.

Let $G \sim \text{SBM}(K, \tilde{\eta}, b, \Lambda)$ with associated adjacency matrix A , and let \mathcal{B} be defined as in (4). For each $P \in \Pi_u$ (with associated permutation σ) and $k, \ell \in [K]$, define

$$\epsilon_{k,\ell} = \epsilon_{k,\ell}(P) = |\{v \in U_k \text{ s.t. } \sigma(v) \in U_\ell\}|$$

to be the number of vertices in U_k mapped to U_ℓ by $I_m \oplus P$, and for each $k \in [K]$ define

$$\epsilon_{k,\bullet}(P) := \epsilon_{k,\bullet} = \sum_{\ell \neq k} \epsilon_{k,\ell}.$$

Before stating and proving the consistency of \mathcal{L}^{ML} , we first establish some necessary notation. Note that in the definitions and theorems presented next, all values implicitly depend on n , as $\Lambda = \Lambda_n$ is allowed to vary in n . Let L be the set of distinct entries of Λ , and define

$$\begin{aligned} \alpha &= \min_{\{k,\ell\} \text{ s.t. } k \neq \ell} |\Lambda_{k,k} - \Lambda_{k,\ell}| & \beta &= \min_{\{k,\ell\} \text{ s.t. } k \neq \ell} |B_{k,k} - B_{k,\ell}| & c &= \max_{i,j,k,\ell} |B_{i,j} - B_{k,\ell}|, & (7) \\ \gamma &= \min_{x,y \in L} |x - y|, & \kappa &= \min_{x,y \in L} \left| \log \left(\frac{x}{1-x} \right) - \log \left(\frac{y}{1-y} \right) \right|. & & (8) \end{aligned}$$

Theorem 6 *Let $G \sim \text{SBM}(K, \bar{n}, b, \Lambda)$ and assume that*

- i. $K = o(\sqrt{n})$;
- ii. $\Lambda \in [0, 1]^{K \times K}$ is such that for all $k, \ell \in [K]$ with $k \neq \ell$, $\Lambda_{k,k} \neq \Lambda_{k,\ell}$;
- iii. For each $k \in [K]$, $u_k = \omega(\sqrt{n})$, and $m_k = \omega(\log u_k)$;
- iv. $\frac{c^2}{\alpha \beta \kappa \gamma} = \Theta(1)$.

Then it holds that $\lim_{n \rightarrow \infty} \mathbb{E} \text{AP}(\mathcal{L}^{\text{ML}}) = 1$, and \mathcal{L}^{ML} is a consistent nomination scheme.

A proof of Theorem 6 is given in the Appendix.

Remark 7 There are numerous assumptions akin to those in Theorem 6 under which we can show that \mathcal{L}^{ML} is consistent. Essentially, we need to ensure that if we define $\mathcal{P} = \{P \in \Pi_n : \epsilon_{1,\bullet}(P) = \Theta(u_1)\}$, then $\mathbb{P}(\exists P \in \mathcal{P} \text{ s.t. } X_P \leq 0)$ is summably small, from which it follows that $\epsilon_{1,\bullet} = o(u_1)$ with high probability, which is enough to ensure the desired consistency of \mathcal{L}^{ML} .

Consistency of $\mathcal{L}_R^{\text{ML}}$ holds under similar assumptions.

Theorem 8 *Let $G \sim \text{SBM}(K, \bar{n}, b, \Lambda)$. Under the following assumptions*

- i. $K = \Theta(1)$;
- ii. $\Lambda \in [0, 1]^{K \times K}$ is such that for all $k, \ell \in [K]$ with $k \neq \ell$, $\Lambda_{k,k} \neq \Lambda_{k,\ell}$;
- iii. For each $k \in [K]$, $u_k = \omega(\sqrt{n})$, and $m_k = \omega(\log u_k)$;
- iv. $\frac{c^2}{\alpha \beta \kappa \gamma} = \Theta(1)$;

it holds that $\lim_{n \rightarrow \infty} \mathbb{E} \text{AP}(\mathcal{L}^{\text{ML}}) = 1$, and \mathcal{L}^{ML} is a consistent nomination scheme.

A proof of this Theorem can be found in the Appendix.

4. Consistency of \mathcal{L}^{ML} and $\mathcal{L}_R^{\text{ML}}$ When the Model Parameters are Unknown

If Λ is unknown a priori, then the seeds can be used to estimate Λ as $\hat{\Lambda}$, and n_i as \hat{n} for each $i \in [K]$. In this section, we will prove analogues of the consistency Theorems 6 and 8 in the case where Λ and \bar{n} are estimated using seeds. In Theorems 9 and 10 below, we prove that under mild model assumptions, both \mathcal{L}^{ML} and $\mathcal{L}_R^{\text{ML}}$ are consistent vertex nomination schemes, even when the seed vertices form a vanishing fraction of the graph.

We now state the consistency result analogous to Theorem 6, this time for the case where we estimate Λ and \bar{n} . The proof can be found in the Appendix.

Theorem 9 *Let $\Lambda \in \mathbb{R}^{K \times K}$ be a fixed, symmetric, block probability matrix satisfying*

- i. K is fixed in n ;
- ii. $\Lambda \in [0, 1]^{K \times K}$ is such that for all $k, \ell \in [K]$ with $k \neq \ell$, $\Lambda_{k,k} \neq \Lambda_{k,\ell}$;
- iii. For each $k \in [K]$, $n_k = \Theta(n)$ and $m_k = \omega(n^{2/3} \log(n))$;
- iv. α and γ defined as in (7) and (8) are fixed in n .

Suppose that the model parameters of $G \sim (K, \bar{n}, b, \Lambda)$ are estimated as in Remark 5 yielding log-odds matrix estimate B and estimated block sizes $\hat{n} = (\hat{n}_1, \hat{n}_2, \dots, \hat{n}_K)^T$. If \mathcal{L}^{ML} is run on A and \hat{B} using the block sizes given by \hat{n} , then under the above assumptions it holds that $\lim_{n \rightarrow \infty} \mathbb{E} \text{AP}(\mathcal{L}^{\text{ML}}) = 1$, and \mathcal{L}^{ML} is a consistent nomination scheme.

We now state the analogous consistency result to Theorem 8 when we estimate Λ and \bar{n} . The proof is given in the Appendix.

Theorem 10 *Let $\Lambda \in \mathbb{R}^{K \times K}$ be a fixed, symmetric, block probability matrix satisfying*

- i. K is fixed in n ;
- ii. $\Lambda \in [0, 1]^{K \times K}$ is such that for all $k, \ell \in [K]$ with $k \neq \ell$, $\Lambda_{k,k} \neq \Lambda_{k,\ell}$;
- iii. For each $k \in [K]$ s.t. $k \neq 1$, $n_k = \Theta(n)$ and $m_k = \omega(n^{2/3} \log(n))$;
- iv. $n_1 = \Theta(n)$ and $m_1 = \omega(n^{4/5})$;
- v. α and γ defined at (7) and (8) are fixed in n .

Suppose that the model parameters of $G \sim (K, \bar{n}, b, \Lambda)$ are estimated as in Remark 5 yielding B and estimated block sizes $\hat{n} = (\hat{n}_1, \hat{n}_2, \dots, \hat{n}_K)^T$. If \mathcal{L}^{ML} is run on A and \hat{B} using block sizes given by \hat{n} , then under the above assumptions it holds that $\lim_{n \rightarrow \infty} \mathbb{E} \text{AP}(\mathcal{L}^{\text{ML}}) = 1$ and \mathcal{L}^{ML} is a consistent nomination scheme.

The two preceding theorems imply that vertex nomination is possible even when the number of seeds is a vanishing fraction of the vertices in the graph. Indeed, we find that in practice, accurate nomination is possible even with just a handful of seed vertices. See the experiments presented in Section 6.

where, for $k \in [K]$, $\hat{f}_k(\cdot)$ is the estimated density of the k -th block features. Note that here we assume that the feature densities must be estimated, even when the matrix Λ is known. A low/high value of $\eta_P(i)$ is a measure of our confidence that i is/is not in the block of interest. For $i \in U$ such that $\phi(i) \neq 1$, define

$$\xi_P(i) := \left(\prod_{\substack{j \in U \text{ s.t.} \\ \phi(j)=1}} \frac{f_P(\phi_{k+j}, G)}{f_P(\phi, G)} \right)^{\frac{1}{u_1}}$$

A low/high value of $\xi_P(i)$ is a measure of our confidence that i is/is not in the block of interest. The nomination list produced by $\mathcal{L}_P^{\text{ML}}$ is then realized via:

$$\begin{aligned} (\mathcal{L}_P^{\text{ML}})^{-1}(1) &\in \arg \min \{ \eta_P(v) : \phi(v) = 1 \} \\ (\mathcal{L}_P^{\text{ML}})^{-1}(2) &\in \arg \min \{ \eta_P(v) : v \in U \setminus \{ (\mathcal{L}_P^{\text{ML}})^{-1}(1) \}, \phi(v) = 1 \} \\ &\vdots \\ (\mathcal{L}_P^{\text{ML}})^{-1}(u_1) &\in \arg \min \left\{ \eta_P(v) : v \in U \setminus \{ (\mathcal{L}_P^{\text{ML}})^{-1}(i) \}_{i=1}^{u_1-1}, \phi(v) = 1 \right\} \\ (\mathcal{L}_P^{\text{ML}})^{-1}(u_1 + 1) &\in \arg \max \{ \xi_P(v) : \phi(v) \neq 1 \} \\ (\mathcal{L}_P^{\text{ML}})^{-1}(u_1 + 2) &\in \arg \max \{ \xi_P(v) : v \in U \setminus \{ (\mathcal{L}_P^{\text{ML}})^{-1}(u_1 + 1) \}, \phi(v) \neq 1 \} \\ &\vdots \\ (\mathcal{L}_P^{\text{ML}})^{-1}(u) &\in \arg \max \left\{ \xi_P(v) : v \in U \setminus \{ (\mathcal{L}_P^{\text{ML}})^{-1}(i) \}_{i=u_1+1}^{u-1}, \phi(v) \neq 1 \right\} \end{aligned}$$

Note that, once again, in the event that the argmin (or argmax) contains more than one element above, the order in which these elements is nominated should be taken to be uniformly random.

We leave for future work a more thorough investigation of how best to choose the parameter λ . We found that choosing λ approximately equal to the number of nonseed vertices yielded reliably good results, but in general the best choice of λ is likely to be dependent on both the structure of the graph and the available features (e.g., how well the features actually predict block membership). We note also that in the case where the feature densities are not easily estimated or where we would like to relax our distributional assumptions, we might consider other terms to use in lieu of $\text{tr } FP^T$. For example, let $\hat{\mu}_k = \frac{1}{m_k} \sum_{v \in S_k} X_v$ be the empirical estimate of μ_k , the average feature vector for the seeds in block k , and create let Y be defined via

$$Y = \begin{bmatrix} u_1 & \hat{\mu}_1 \otimes \bar{\mathbf{1}} \\ u_2 & \hat{\mu}_2 \otimes \bar{\mathbf{1}} \\ \vdots & \vdots \\ u_k & \hat{\mu}_k \otimes \bar{\mathbf{1}} \end{bmatrix}^d.$$

Incorporating these features into the seeded graph matching problem similarly to (9), we have

$$\hat{P} = \arg \min_{P \in \Pi_u} -\frac{1}{2} \text{tr} \left(A^{(2,2)} P (B^{(2,2)})^T P^T \right) - \text{tr} \left((A^{(1,2)})^T B^{(1,2)} P^T \right) - \lambda \text{tr} (X^{(u)} Y^T P^T). \quad (10)$$

We leave further exploration of this and related approaches, as well as how to deal with categorical data (e.g., as in Newman and Clauset (2016)), for future work.

6. Experiments

To compare the performance of maximum-likelihood vertex nomination against other methods, we performed experiments on five data sets, one synthetic, the others from linguistics, sociology, political science and ecology.

In all our data sets, we consider vertex nomination both when the edge probability matrix Λ is known and when it must be estimated. When model parameters are unknown, $m < n$ seed vertices are selected at random and the edge probability matrix is estimated based on the subgraph induced by the seeds, with entries of the edge probability matrix estimated via add-one smoothing. In the case of synthetic data, the known-parameter case simply corresponds to the algorithm having access to the parameters used to generate the data. In this paper, we consider a 3-block stochastic block model (see below), so the known-parameter case corresponds to the true edge probability matrix being given. In the case of our real-world data sets, the notion of a ‘‘true’’ Λ is more hazy. Here, knowing the model parameters corresponds to using the entire graph, along with the true block memberships, to estimate Λ , again using add-one smoothing. This is, in some sense, the best access we can hope to have to the model parameters, to the extent that such parameters even exist in the first place.

6.1 Simulations

We consider graphs generated from stochastic block models at two different scales. Following the experiments in Fishkind et al. (2015), we consider 3-block models, where block sizes are given by $\bar{n} = q \cdot (4, 3, 3)^T$ for $q = 1, 50$, which we term the small and medium cases, respectively. In Fishkind et al. (2015), a third case, with $q = 1000$, was also considered, but since ML vertex nomination is not practical at this scale, we do not include such experiments here, though we note that $\mathcal{L}_R^{\text{ML}}$ can be run successfully on such a graph. We use an edge probability matrix given by

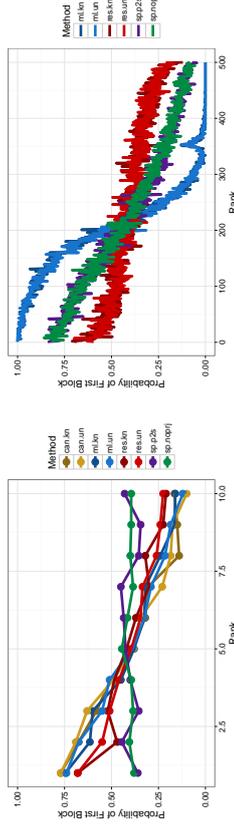
$$\Lambda(t) = t \begin{bmatrix} 0.5 & 0.3 & 0.4 \\ 0.3 & 0.8 & 0.6 \\ 0.4 & 0.6 & 0.3 \end{bmatrix} + (1-t) \begin{bmatrix} 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 \\ 0.5 & 0.5 & 0.5 \end{bmatrix} \quad (11)$$

for $t = 1, 0.3$ respectively in the small and medium cases, so that the amount of signal present in the graph is smaller as the number of vertices increases. We consider $m = 4, 20$ seeds in the small and medium scales, respectively. For a given choice of \bar{n}, m, t , we generate a single draw of an SBM with edge probability matrix $\Lambda(t)$ and block sizes given by \bar{n} . A set

of m vertices is chosen uniformly at random from the first block to be seeds. Note that this means that the only model parameter that can be estimated is the intra-block probability for the first block. For all model parameter estimation in the ML methods (i.e., for the unknown case of \mathcal{L}^{ML} and $\mathcal{L}_R^{\text{ML}}$), we use add-1 smoothing to prevent inaccurate estimates. We note that in all conditions, the block of interest (the first block) is not the densest block of the graph.

Recall that all of the methods under consideration return a list of the nonseed vertices, which we call a *nomination list*, with the vertices sorted according to how likely they are to be in the block of interest. Thus, vertices appearing early in the nomination list are the best candidates to be vertices of interest. Figure 1 compares the performance of canonical, spectral, maximum-likelihood and restricted-focus ML vertex nomination by looking at (estimates of) their average nomination lists. The plot shows, for each of the methods under consideration, an estimate (each based on 200 Monte Carlo replicates) of the average nomination list. Each curve describes the empirical probability that the k th-ranked vertex was indeed a vertex of interest. A perfect method, which on every input correctly places the n_1 vertices of interest in the first n_1 entries of the nomination list, would produce a curve in Figure 1 resembling a step function, with a step from 1 to 0 at the $(n_1 + 1)$ th rank. Conversely, a method operating purely at random would yield an average nomination list that is constant n_1/n . Canonical vertex nomination is shown in gold, ML in blue, restricted-focus ML in red, and spectral vertex nomination is shown in purple and green. These two colors correspond, respectively, to spectral VN in which vertex embeddings are projected to the unit sphere prior to nomination and in which the embeddings are used as-is. In sparse networks, the adjacency spectral embedding places all vertices near to the origin. In such settings, projection to the sphere often makes cluster structure in the embeddings more easily recoverable. Dark colors correspond to the known-parameter case, and light colors correspond to unknown parameters. Note that spectral VN does not make such a distinction.

Examining the plots, we see that in the small case, maximum-likelihood nomination is quite competitive with the canonical method, and restricted-focus ML nomination is not much worse. Somewhat surprising is that these methods perform well seemingly irrespective of whether or not the model parameters are known, though this phenomenon is accounted for by the fact that the smoothed estimates are automatically close to the truth, since A is approximately equal to the matrix with all entries $1/2$. Meanwhile, the small number of nodes is such that there is little signal available to spectral vertex nomination. We see that spectral vertex nomination performs approximately at-chance regardless of whether or not we project the spectral embeddings to the sphere. 10 nodes are not enough to reveal eigenvalue structure that spectral methods attempt to recover. In the medium case, where there are 500 vertices, enough signal is present that reasonable performance is obtained by spectral vertex nomination, with performance with (purple) and without (green) projection to the sphere again indistinguishable. The comparative density of the SBM in question ensures that projection to the sphere is not necessary, and that doing so does no appreciable harm to nomination. However, in the medium case, ML-based vertex nomination still appears to best spectral methods, with the known and unknown cases being nearly indistinguishable. We note that in both the small and medium cases all of the methods appear to intersect at an empirical probability of 0.4. These intersection points correspond



(a) Small scale simulation results

(b) Medium scale simulation results

Figure 1: The mean nomination lists for the (a) small and (b) medium stochastic block model experiments for the different vertex nomination techniques in both the known (dark colors) and unknown (light colors). Plot (a) shows performance for the canonical (gold), maximum likelihood (blue), restricted-focus maximum likelihood (red) and spectral (green and purple) methods. Spectral VN both with and without projection to the sphere is shown in purple and green, respectively. Plot (b) does not include canonical vertex nomination due to runtime constraints.

to the transition from the block of interest to the non-interesting vertices: these vertices, about which we are least confident, tend to be nominated correctly at or near chance, which is 40% in both the small and large cases.

A more quantitative assessment of the vertex nomination methods is contained in Tables 1 and 2, which compare the performance of the methods as assessed by, respectively, average precision (AP) and adjusted Rand index (ARI). As defined in Equation (1), AP is a value between 0 and 1, where a value of 1 indicates perfect performance. ARI Hubert and Arabie (1985) measures how well a given partition of a set recovers some ground truth partition. Here a value of 1 indicates perfect recovery, while randomly partitioning a data set yields ARI approximately 0 (note that negative ARI is possible). We include ARI as an evaluation to highlight the fact that spectral and maximum-likelihood nomination do not merely classify vertices as interesting or not. Rather, they return a partition of the vertices into clusters. Canonical vertex nomination, on the other hand, makes no attempt to recover the full cluster structure of the graph, instead only attempting to classify vertices according to whether or not they are of interest. As such, we do not include ARI numbers for canonical vertex nomination. Turning first to performance in the small graph condition in Table 1, we see that \mathcal{L}^{C} is the best method, so long as the graph in question is small enough that the canonical method is tractable, but \mathcal{L}^{ML} , regardless of whether or not model parameters are known, nearly matches canonical VN, and, unlike its canonical counterpart, scales to graphs with more than a few nodes. The numbers for \mathcal{L}^{SP} bear out our observation above, that the small graphs contain too little information for spectral VN to act upon, and \mathcal{L}^{SP} performs approximately at chance, as a result. It is worth noting that while $\mathcal{L}_R^{\text{ML}}$ does not match the performance of \mathcal{L}^{ML} , presumably owing to the fact that the restricted-focus algorithm does not use all of the information present in the graph, it still outperforms spectral nomination, and lags \mathcal{L}^{ML} by less than 0.1 AP.

	Known				Unknown			
	ML	RES	SP	CAN	ML	RES	SP	CAN
small	0.670	0.588	0.388	0.700	0.680	0.606	0.415	0.710
medium	0.954	0.545	0.738	—	0.954	0.537	0.735	—

Table 1: Empirical estimates of mean average precision on the two stochastic block model data sets for the four methods under consideration. Each data point is the mean of 200 independent trials.

	Known				Unknown			
	ML	RES	SP	CAN	ML	RES	SP	CAN
small	0.338	0.259	0.011	—	0.338	0.259	0.011	—
medium	0.572	0.039	0.268	—	0.572	0.037	0.271	—

Table 2: ARI on the different sized data sets for the ML, restricted ML, and spectral methods. Each data point is the mean of 200 independent trials. Performance of canonical vertex nomination is knot included, since canonical vertex nomination makes no attempt to recover all three blocks, and thus ARI is not a sensible measure.

Turning our attention to the medium case, we see again that $\mathcal{L}_{ML}^{\text{ML}}$ and $\mathcal{L}_{R}^{\text{ML}}$ remain largely impervious to whether model parameters are known or not, presumably a consequence of the use of smoothing—we’ll see in the sequel that estimation can be the difference between near-perfect performance and near-chance. With more vertices, we see that spectral improves above chance, leaving restricted ML slightly worse, but spectral still fails to match the performance of ML, VN, even when model parameters are unknown.

In sum, these results suggest that different size graphs (and different modeling assumptions) call for different vertex nomination methods. In small graphs, regardless of whether or not model parameters are known, canonical vertex nomination is both tractable and quite effective. In medium graphs, maximum-likelihood vertex nomination remains tractable and achieves impressively good nomination. Of course, for graphs with thousands of vertices, $\mathcal{L}_{ML}^{\text{ML}}$ becomes computationally expensive, leaving only $\mathcal{L}_{R}^{\text{SP}}$ and $\mathcal{L}_{R}^{\text{ML}}$ as options. We have observed that $\mathcal{L}_{R}^{\text{ML}}$ tends to lag $\mathcal{L}_{R}^{\text{SP}}$ in such large graphs, though increasing the number of seeds (and hence the amount of information available to $\mathcal{L}_{R}^{\text{ML}}$) closes this gap considerably. We leave for future work a more thorough exploration of under what circumstances we might expect $\mathcal{L}_{R}^{\text{ML}}$ to be competitive with $\mathcal{L}_{R}^{\text{SP}}$ in graphs on thousands of vertices.

6.2 Word Co-occurrences

We consider a linguistic data set consisting of co-occurrences of 54 nouns and 58 adjectives in Charles Dickens’ novel *David Copperfield* Newman (2006a). We construct a graph in which each node corresponds to a word, and an edge connects two nodes if the two corresponding words occurred adjacent to one another in the text. The adjacency matrix of this graph is shown in Figure 2. Visual inspection reveals a clear block structure, and that this block structure is clearly not assortative (i.e., inter-block edges are more frequent than intra-block

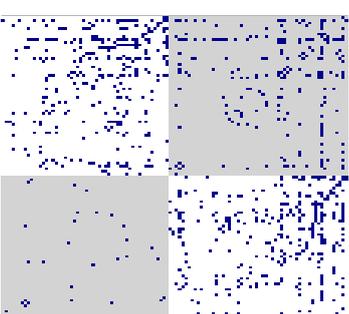


Figure 2: Adjacency matrix of the linguistic data set, arranged to highlight the graph’s structure. The grey shading indicates the two blocks, with adjectives in the upper left and nouns in the lower right. Note the disassortative block structure.

edges). This runs contrary to many commonly-studied data sets and model assumptions. Figure 3 shows the performance of spectral and maximum-likelihood vertex nomination, measured by (a) average precision and adjusted Rand index (ARI) at various numbers of seeds. Each data point is the average over 1000 trials. In each trial, a set of m seeds was chosen uniformly at random from the 112 nodes, with the restriction that at least one noun and one adjective be included in the seed set. Performance was then measured as the mean average precision in identifying the adjective block.

Figure 3 shows the performance of the VN schemes under consideration, as a function of the number of seed vertices, using both known (dark colors) and estimated (light colors) model parameters. Looking first at AP in Figure 3 (a), we see that ML in the known-parameter case (dark blue) does consistently well, even with only a handful of seeds, and attains near-perfect performance for $m \geq 20$. When model parameters must be estimated (light blue), ML is less dominant, though it still performs nearly perfectly for $m \geq 20$. We note the dip in unknown-parameters ML as m increases from 2 to 5 to 10, a phenomenon we attribute to the bias-variance tradeoff. Namely, with more seeds available, variance in the estimated model parameters increases, but for $m < 20$, this increase in variance is not offset by an appreciable improvement in estimation, possibly attributable to our use of add-one smoothing. Somewhat surprisingly, restricted-focus ML performs quite well, consistently improving on spectral VN in the known parameter case for $m > 2$, and in the unknown parameter case once $m > 10$. Finally, we turn our attention to spectral VN, shown in green for the variant in which we project embeddings to the sphere and in purple for the variant in which we do not. In contrast to our simulations, the sparsity of this network makes projection to the sphere a critical requirement for successful retrieval of the first block. Without projection to the sphere, spectral VN fails to rise appreciably above chance performance.

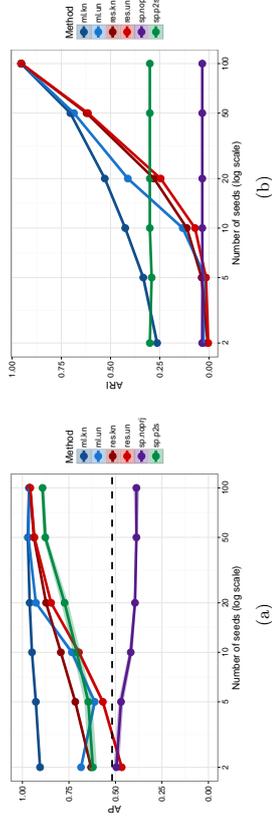


Figure 3: Performance on the linguistic data set as measured by (a) AP and (b) ARI as a function of the number of seeds for the ML vertex nomination (blue), restricted-focus ML (red), and spectral vertex nomination with (green) and without projection to the sphere (violet), when model parameters are known (light colors) and unknown (dark colors). Each data point is the mean of 1000 Monte Carlo trials, and shaded regions indicate two standard deviations of the mean.

6.3 Zachary’s Karate Club

We consider the classic sociological data set, Zachary’s karate club network Zachary (1977). The graph, visualized in Figure 4, consists of 34 nodes, each corresponding to a member of a college karate club, with edges joining pairs of club members according to whether or not those members were observed to interact consistently outside of the club. Over the course of Zachary’s observation of the group, a conflict emerged that led to the formation of two factions, led by the individuals numbered 1 and 34 in Figure 4, and these two factions constitute the two blocks in this experiment. Zachary’s karate data set is particularly well-suited for spectral methods. Indeed, the flow-based model originally proposed by Zachary recovers factions nearly perfectly, and visual inspection of the graph (Figure 4) suggests a natural cut separating the two factions. As such, we expect ML-based vertex nomination to lose out against the spectral-based method. Figure 5 shows performance of the two algorithms as measured by ARI and average precision. We see, as expected, that spectral performance performs nearly perfectly, irrespective of the number of seeds. Surprisingly, maximum-likelihood nomination is largely competitive with spectral VN, but only provided that the model parameters are already known. Interesting to note that here again we see the phenomenon discussed previously in which ML performance with an unknown edge probability matrix degrades when going from $s = 2$ seeds to $s = 5$ before improving again, with AP comparable to the known case for $s \geq 20$.

6.4 Political Blogs

We consider a network of American political blogs in the lead-up to the 2004 election Adamic and Glance (2005), where an edge joins two blogs if one links to the other, with blogs classified according to political leaning (liberal vs conservative). From an initial 1490 vertices, we

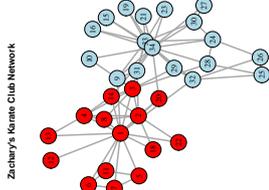


Figure 4: Visualization of the graph corresponding to Zachary’s karate club data set. The vertices are colored according to which of the two clubs each member chose to join after the schism. Our block of interest is in red.

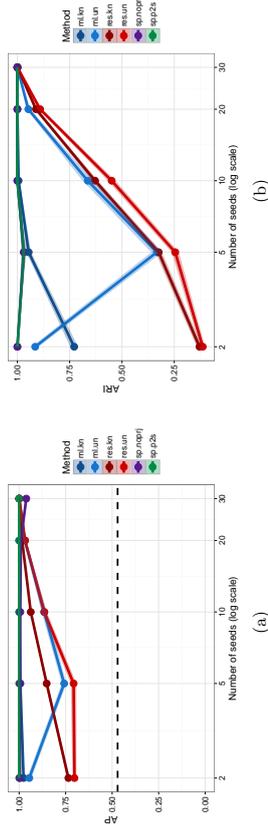


Figure 5: Performance on the karate data set as a function of the number of seeds for the ML vertex nomination (blue), restricted-focus ML nomination (red), and spectral vertex nomination with (green) and without projection to the sphere (violet), when model parameters are known (light colors) and unknown (dark colors) as measured by (a) AP and (b) ARI. The black dashed line indicates chance performance. Each observation is the mean of 1000 independent trials, with the shaded bars indicating two standard errors of the mean in either direction.

removed all isolated vertices to obtain a network of 1224 vertices and 16718 edges. Figure 6 shows the performance of the spectral- and ML-based methods in recovering the liberal block. We observe first and foremost that the sparsity of this network results in exceptionally poor performance in both AP and ARI for spectral VN unless the embeddings are projected to the sphere, but that spectral vertex nomination is otherwise quite effective at recovering the liberal block, with performance nearly perfect for $m > 10$. Unsurprisingly, ML and its restricted counterpart both perform approximately at-chance when $m < 10$. We see that in both the known and unknown cases, ML VN is competitive with spectral VN for

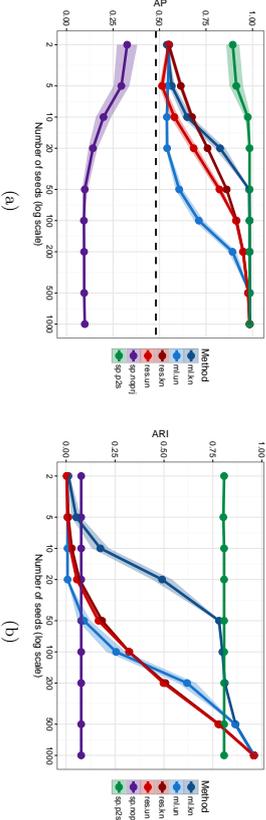


Figure 6: Performance on the political blogs data set as a function of the number of seeds for the ML vertex nomination (blue), restricted-focus ML (red), and spectral vertex nomination with (green) and without projection to the sphere (violet), when model parameters are known (light colors) and unknown (dark colors), as measured by (a) AP and (b) ARI.

suitably large m ($m \geq 50$ for known, $m \geq 500$ for unknown). As expected in such a sparse network, restricted-focus ML lags ML in the known-parameter case, but surprisingly, in the unknown-parameter case, restricted ML achieves remarkably better AP than does ML, a fact we are unable to account for, though it is worth noting that looking at ARI in Figure 6 (b), no such gap appears between ML and its restricted-focus counterpart in the unknown-parameter case.

6.5 Ecological Network

We consider a trophic network, consisting of 125 nodes and 1907 edges, in which nodes correspond to (groups of) organisms in the Florida Bay ecosystem Ulanowicz et al. (1997): Noy et al. (2011), and an edge joins a pair of organisms if one feeds on the other. Our features are the (log) mass of organisms. We take our community of interest to be the 16 different types of birds in the ecosystem. This choice makes for an interesting task for several reasons. Firstly, unlike the other data sets we consider, our community of interest is a comparatively small fraction of the network—it consists of a mere 16 nodes of 125 in total. Further, our block of interest is comparatively heterogeneous in the sense that the roles of the different types of birds in the Florida Bay ecosystem is quite diverse. For example, the block of interest includes both raptors and shorebirds, which feed on quite different collections of organisms. Finally, it stands to reason that the mass of the organisms in question might be a crucial piece of information for disambiguating, say, a raptor from a shark. Thus, we expect that using node features will be crucial for retrieving the block of interest.

The topology of the Florida Bay network is shown in Figure 7 (a). Note that the block of interest, indicated in red, has a strongly disassortative structure. Indeed, all intra-block edges in the red block are incident to the node corresponding to raptors. Figure 7 (b) summarizes vertex nomination performance for several methods. The plot shows performance,

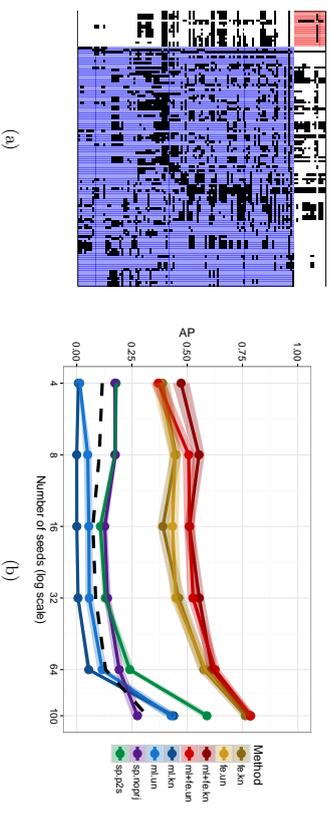


Figure 7: (a) The adjacency matrix of the Florida Bay trophic network. Nodes correspond to classes of plants and animals (e.g., sharks, rays, shorebirds, zooplankton, phytoplankton). An edge joins two nodes if the corresponding organisms are in a predator-prey relation. The sixteen types of birds in the network are highlighted in the red block. Note the disassortative structure of the bird block (the edges within the red block are all incident to the node that corresponds to raptors). (b) Average precision in identifying the bird nodes as a function of the number of seed vertices for ML vertex nomination (blue), restricted-focus ML (red), and spectral vertex nomination with (green) and without projection to the sphere (violet), when model parameters are known (light colors) and unknown (dark colors). The black dashed line indicates chance performance.

as measured by mean average precision (AP), as a function of the number of seeds for several different nomination schemes. As in earlier plots, dark colors correspond to model parameters being known, while light colors correspond to model parameters being estimated using the seed vertices. We see immediately that spectral nomination (green and purple) and ML VN (blue) fail to improve appreciably upon chance performance except when the vast majority of the vertices’ labels are observed. Like in the linguistic data set presented above, the disassortative structure of the data appears to cause problems for spectral nomination. The failure of ML suggests that no useful information is encoded in the graph itself, but turning our attention to the curves corresponding to \mathcal{L}_{PL}^{ML} (red) and using only features (gold), we see that this is not the case. Indeed, we see that while using features alone achieves a marked improvement over both spectral and ML-based nomination, using both features and graph matching in the form of \mathcal{L}_{PL}^{ML} yields an additional improvement of some 0.1 AP in the range of $m = 8, 16, 32$. This result suggests that there may be cases where the only reliable way to retrieve vertices of interest is to leverage both features and graph topology jointly.

7. Discussion and Future Work

Network data has become ubiquitous in the sciences, giving rise to a vast array of computational and statistical problems that are only beginning to be explored. In this paper, we have explored one such problem that arises when working with network data, namely the task of performing vertex nomination. This task, in some sense the graph analogue of the classic information retrieval problem, is fundamental to exploratory data analysis on graphs as well as to machine learning applications. Above, we established the consistency of two methods of vertex nomination: a maximum-likelihood scheme \mathcal{L}^{ML} and its restricted-focus variant $\mathcal{L}_R^{\text{ML}}$, in which we obtain a feasibly exactly-solvable optimization problem at the expense of using less than the full information available in the graph. Additionally, we have introduced a maximum-likelihood nomination scheme for the case where vertices are endowed with features and when (possibly weighted) edges are drawn from a canonical exponential family. The key to all of these methods is the ability to quickly approximate a solution to the seeded graph matching problem.

We have presented experimental comparisons of these methods against each other and against several other benchmark methods, where we see that the best choice of method depends highly on graph size and structure. The major tradeoff appears to be that large graphs (tens of thousands of vertices) are not tractable for \mathcal{L}^{ML} , but in smaller and medium-sized graphs, \mathcal{L}^{ML} can detect signal where spectral methods fail to do so. It is worth noting that \mathcal{L}^{ML} , and, to a lesser extent, $\mathcal{L}_R^{\text{ML}}$, is quite competitive with \mathcal{L}^{SP} , and even manages to best \mathcal{L}^{SP} when the structure of the graph is ill-suited to the typical assumptions of spectral methods, as in the case of our linguistic data set. All told, our experimental results mirror those in Fishkind et al. (2015) and point toward a theory of which methods are best-suited to which graphs, a direction that warrants further exploration.

Acknowledgments

The authors would like to thank the reviewers for their feedback, which was both prompt and helpful. This work is supported in part by Johns Hopkins University Human Language Technology Center of Excellence (JHU HLTCOE), the XDATA program of the Defense Advanced Research Projects Agency (DARPA) administered through Air Force Research Laboratory contract FA8750-12-2-0303, and the DARPA SIMPLEX program SPAWAR contract N66001-15-C-4041

Appendix A.

Before proving Theorem 6, we first state a useful initial proposition.

Proposition 12 *Let $\vec{x} = (x_1, x_2, \dots, x_k)$ be a vector with distinct entries in \mathbb{R}^k . Let $f(\cdot)$ be a strictly increasing real valued function (with the abuse of notation, $f(\vec{x})$, denoting $f(\cdot)$ applied entrywise to \vec{x}). Let the order statistics of \vec{x} be denoted*

$$x_{(1)} < x_{(2)} < \dots < x_{(k)},$$

and define $\alpha = \min_{i \in \{2, 3, \dots, k\}} |x_{(i)} - x_{(i-1)}|$, and $\beta = \min_{i \in \{2, 3, \dots, k\}} |f(x_{(i)}) - f(x_{(i-1)})|$. If σ is the cyclic permutation

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & \dots & k \\ 2 & 3 & 4 & \dots & 1 \end{pmatrix},$$

then

$$\langle \vec{x}, f(\vec{x}) \rangle - \langle \vec{x}, f(\sigma(\vec{x})) \rangle \geq (k-1)\alpha\beta.$$

Proof We will induct on k . To establish the base case, $k = 2$, let $x_1 = x_{(1)}$ without loss of generality and observe that

$$\begin{aligned} \langle \vec{x}, f(\vec{x}) \rangle - \langle \vec{x}, f(\sigma(\vec{x})) \rangle &= (x_2 - x_1)(f(x_2) - f(x_1)) \\ &= (x_{(2)} - x_{(1)})(f(x_{(2)}) - f(x_{(1)})) \geq \alpha\beta. \end{aligned}$$

For general k , again, without loss of generality let $x_1 = x_{(1)}$, and define the permutation

$$\tau = \begin{pmatrix} 2 & 3 & \dots & k \\ 3 & 4 & \dots & 2 \end{pmatrix}.$$

Then

$$\begin{aligned} \langle \vec{x}, f(\vec{x}) \rangle - \langle \vec{x}, f(\sigma(\vec{x})) \rangle &= \langle \vec{x}, f(\vec{x}) \rangle - \langle \vec{x}, f(\tau(\vec{x})) \rangle + \langle \vec{x}, f(\tau(\vec{x})) \rangle - \langle \vec{x}, f(\sigma(\vec{x})) \rangle \\ &= \langle \vec{x}, f(\vec{x}) \rangle - \langle \vec{x}, f(\tau(\vec{x})) \rangle + (x_k - x_1)(f(x_2) - f(x_1)) \\ &\geq \langle \vec{x}, f(\vec{x}) \rangle - \langle \vec{x}, f(\tau(\vec{x})) \rangle + \alpha\beta, \end{aligned}$$

and the result follows from the inductive hypothesis. \blacksquare

Remark 13 It follows immediately that in Proposition 12, if there exists an index $i \in [k]$ such that $\alpha_i = \min_{j \neq i} |x_{(i)} - x_{(j)}| > 0$, and $\beta_i = \min_{j \neq i} |f(x_{(i)}) - f(x_{(j)})| > 0$, then $\langle \vec{x}, f(\vec{x}) \rangle - \langle \vec{x}, f(\sigma(\vec{x})) \rangle \geq \alpha_i \beta_i$.

We are now ready to prove Theorem 6.

Proof [Proof of Theorem 6] Define

$$X_P := \text{tr}(AB^\top) - \text{tr}(A(I_m \oplus P)B(I_m \oplus P)^\top)$$

and define $\mathcal{P} = \{P \in \Pi_u : \epsilon_{1,\bullet}(P) > 0\}$. We will show that

$$\mathbb{P}(\exists P \in \mathcal{P} \text{ s.t. } X_P \leq 0) = O(1/n^2),$$

from which the desired consistency of \mathcal{L}^{ML} follows by the Borel-Cantelli Lemma, since this probability is summable in n . Fix $P \in \mathcal{P}$, and let $\sigma_P \in S_n$ be the permutation associated with $I_m \oplus P$. The action of shuffling B via $I_m \oplus P$ is equivalent to permuting the $[n^2]$ elements of $\text{vec}(B)$ via a permutation τ_P , in that

$$\text{tr}(A(I_m \oplus P)B(I_m \oplus P)^\top) = \langle \text{vec}(A), \tau_P(\text{vec}(B)) \rangle.$$

Moreover, τ_P can be chosen so that, in the cyclic decomposition of $\tau_P = \tau_P^{(1)} \tau_P^{(2)} \cdots \tau_P^{(l)}$, each (disjoint) cycle is acting on a set of distinct real numbers. Note that Proposition 12 implies that the contribution of each cycle $\tau_P^{(i)}$ to $\mathbb{E}(X_P)$ is nonnegative, and the assumptions of Theorem 6 imply that for each $i, j \in [K]$ such that $i \neq j$, the contribution of each (nontrivial) cycle permuting a $\Lambda_{i,j}$ entry to a $\Lambda_{i,j}$ entry contributes at least $\alpha\beta$ to $\mathbb{E}(X_P)$. It follows immediately that

$$\begin{aligned} \mathbb{E}(X_P) &= \mathbb{E}\left(\operatorname{tr}(AB) - \operatorname{tr}(APBP^\top)\right) \\ &= \mathbb{E}\left(\operatorname{vec}(A), \operatorname{vec}(B)\right) - \left(\operatorname{vec}(A), \tau_P(\operatorname{vec}(B))\right) \\ &\geq 2\alpha\beta \sum_i \left(\frac{1}{2} \sum_{j \neq i} \sum_{k \neq j} \epsilon_{i,j} \epsilon_{i,k} + m_i \epsilon_{i,\bullet}\right) \\ &\geq 2\alpha\beta \sum_i \left(\frac{u_i - \epsilon_{i,\bullet}}{2} \epsilon_{i,\bullet} + m_i \epsilon_{i,\bullet}\right). \end{aligned}$$

Let $\mathfrak{n}(P)$ be the total number of distinct entries of $\operatorname{vec}(B)$ permuted by τ_P , and note that an application of Proposition 12 yields

$$\begin{aligned} \mathbb{E}(X_P) &= \mathbb{E}\left(\operatorname{tr}(AB) - \operatorname{tr}(APBP^\top)\right) \\ &= \mathbb{E}\left(\operatorname{vec}(A), \operatorname{vec}(B)\right) - \left(\operatorname{vec}(A), \tau_P(\operatorname{vec}(B))\right) \\ &\geq \frac{1}{2} \mathfrak{n}(P) \gamma \kappa. \end{aligned}$$

The assumptions in the Theorem also immediately yield that

$$\mathfrak{n}(P) \geq \sum_k \left(\frac{u_k - \epsilon_{k,\bullet}}{2} \epsilon_{k,\bullet} + m_k \epsilon_{k,\bullet}\right).$$

We next note that X_P is a sum of $\mathfrak{n}(P)$ independent random variables, each bounded in $[-c, c]$. An application of Hoeffding's inequality then yields

$$\begin{aligned} \mathbb{P}(X_P \leq 0) &\leq \mathbb{P}\left(|X_P - \mathbb{E}X_P| \geq \mathbb{E}X_P\right) \leq 2 \exp\left\{-\frac{2\mathbb{E}^2 X_P}{4c^2 \mathfrak{n}(P)}\right\} \\ &\leq 2 \exp\left\{-\frac{\mathbb{E}X_P |\kappa \gamma}{2c^2}\right\} \leq 2 \exp\left\{-\frac{\alpha\beta \kappa \gamma}{4c^2} \sum_k \left(\frac{u_k - \epsilon_{k,\bullet}}{2} \epsilon_{k,\bullet} + m_k \epsilon_{k,\bullet}\right)\right\}. \end{aligned}$$

Next, note that

$$|\{P \in \mathcal{P} \text{ s.t. } X_P \leq 0\}| = 0 \text{ iff } |\{P \in \mathcal{P}' \sim \text{s.t. } X_P \leq 0\}| = 0.$$

Given $\{\epsilon_{k,\ell}\}_{k,\ell=1}^K$ satisfying $u_k = \sum_{\ell} \epsilon_{k,\ell} = \sum_{\ell} \epsilon_{\ell,k}$ for all $k \in [K]$, the number of elements $P \in \mathcal{P}' \sim$ with $\epsilon_{k,\ell}(P) = \epsilon_{k,\ell}$ for all $k, \ell \in [K]$ is at most

$$\begin{aligned} &\prod_{\substack{i_1, \ell_1, \ell_2 \\ i_2}} \sum_{\substack{\ell_1, \ell_2 \\ k \neq \ell_1}} \sum_{\substack{\ell_1, \ell_2 \\ k \neq \ell_1}} \epsilon_{k,\ell_1} \epsilon_{k,\ell_2} = u_1^{u_1 - \epsilon_{1,1}} u_2^{u_2 - \epsilon_{2,2}} \cdots u_K^{u_K - \epsilon_{K,K}} \\ &= e^{\sum_k (u_k - \epsilon_{k,k}) \log(u_k)}. \end{aligned} \tag{12}$$

The number of ways to choose such a set (i.e. the $\{\epsilon_{k,\ell}\}_{k,\ell}$) is bounded above by

$$\prod_{k \text{ s.t. } \epsilon_{k,\bullet} \neq 0} (u_k + K)^{K} = e^{\sum_k \text{s.t. } \epsilon_{k,\bullet} \neq 0 K \log(u_k + K)}. \tag{13}$$

Applying the union bound over all $P \in \mathcal{P}' \sim$, we then have

$$\begin{aligned} \mathbb{P}(\exists P \in \mathcal{P} \text{ s.t. } X_P \leq 0) &= \mathbb{P}(\exists P \in \mathcal{P}' \sim \text{ s.t. } X_P \leq 0) \\ &\leq \exp\left\{-\frac{\alpha\beta \kappa \gamma}{2c^2} \sum_k \left(\frac{u_k - \epsilon_{k,\bullet}}{2} \epsilon_{k,\bullet} + m_k \epsilon_{k,\bullet}\right)\right\} \\ &\quad + \sum_k (u_k - \epsilon_{k,k}) \log u_k + \sum_{k \text{ s.t. } \epsilon_{k,\bullet} \neq 0} K \log(u_k + K). \end{aligned} \tag{14}$$

It remains for us to establish that the expression inside the exponent goes to $-\infty$ fast enough to ensure our desired bound. For each k , the contribution to the exponent in (14) is

$$\begin{aligned} &-\frac{\alpha\beta \kappa \gamma}{2c^2} \left(\frac{u_k - \epsilon_{k,\bullet}}{2} \epsilon_{k,\bullet} + m_k \epsilon_{k,\bullet}\right) + (u_k - \epsilon_{k,k}) \log u_k + \mathbb{I}\{\epsilon_{k,\bullet} \neq 0\} K \log(u_k + K) \\ &= -\frac{\alpha\beta \kappa \gamma}{2c^2} \left(\frac{\epsilon_{k,k} \epsilon_{k,\bullet}}{2} + m_k \epsilon_{k,\bullet}\right) + \epsilon_{k,\bullet} \log u_k + \mathbb{I}\{\epsilon_{k,\bullet} \neq 0\} K \log(u_k + K) \end{aligned} \tag{16}$$

If $u_k/2 \leq \epsilon_{k,k} < u_k$, then

$$\epsilon_{k,k} \epsilon_{k,\bullet} \geq \frac{u_k \epsilon_{k,\bullet}}{2} = \omega(\epsilon_{k,\bullet} \log u_k), \text{ and } \epsilon_{k,k} \epsilon_{k,\bullet} \geq \frac{u_k \epsilon_{k,\bullet}}{2} = \omega(K \log(u_k + K)),$$

and the contribution to the exponent in (14) from k , Eq. (16), is clearly bounded above by $-2 \log(n)$ for sufficiently large n . If $\epsilon_{k,k} \leq u_k/2$ then $\epsilon_{k,\bullet} > u_k/2$, and

$$m_k \epsilon_{k,\bullet} = \omega(\epsilon_{k,\bullet} \log u_k), \text{ and } m_k \epsilon_{k,\bullet} \geq \frac{m_k u_k}{2} = \omega(K \log(u_k + K)),$$

and the contribution to the exponent in (14) from k , Eq. (16), is clearly bounded above by $-2 \log(n)$ for sufficiently large n . If $\epsilon_{k,k} = u_k$, then all terms in the exponent (16) are equal to 0. For sufficiently large n , Eq. (14) is then bounded above by

$$\exp\left\{-\sum_{k \text{ s.t. } \epsilon_{k,\bullet} \neq 0} 2 \log(n)\right\} \leq \exp\{-2 \log(n)\},$$

and the result follows. \blacksquare

Consistency of $\mathcal{L}_K^{\text{ML}}$ as claimed in Theorem 8 follows similarly to that of \mathcal{L}^{ML} , and we next briefly sketch the details of the proof.

Proof [Proof of Theorem 8 (Sketch)] Analogously to the proof of Theorem 6, define

$$X_P := \operatorname{tr}\left((A^{(1,2)})^\top B^{(1,2)}\right) - \operatorname{tr}\left((A^{(1,2)})^\top B^{(1,2)} P^\top\right).$$

The proof follows *mutatis mutandis* to the proof of Theorem 6, with the key difference being that in this case,

$$\begin{aligned} \mathbb{E}(X_P) &= \mathbb{E} \left(\operatorname{tr} \left((A^{(1,2)})^\top B^{(1,2)} \right) - \operatorname{tr} \left((A^{(1,2)})^\top B^{(1,2)} P^\top \right) \right) \\ &\geq 2\alpha\beta \sum_k m_k \epsilon_{k,\bullet}. \end{aligned}$$

Details are omitted for brevity. \blacksquare

Before proving Theorem 9 we establish some preliminary concentration results for our estimates $\hat{\Lambda}$, and \hat{n}_k , $k \in [K]$. An application of Hoeffding's inequality yields that for $k, \ell \in [K]$ such that $k \neq \ell$,

$$\mathbb{P} \left(\left| \hat{\Lambda}_{k,\ell} - \Lambda_{k,\ell} \right| \geq \frac{\sqrt{n \log n}}{m_k m_\ell} \right) \leq 2 \exp \{-2n \log n\}, \quad (17)$$

and for $k \in [K]$,

$$\mathbb{P} \left(\left| \hat{\Lambda}_{k,k} - \Lambda_{k,k} \right| \geq \frac{\sqrt{n \log n}}{\binom{m_k}{2}} \right) \leq 2 \exp \{-2n \log n\}, \quad (18)$$

and

$$\mathbb{P} \left(|\hat{n}_k - n_k| \geq t \right) \leq 2 \exp \left\{ -\frac{2nt^2}{n^2} \right\}, \quad (19)$$

With γ defined as in (8), define the events $\mathcal{E}_n^{(1)}$ and $\mathcal{E}_n^{(2)}$ via

$$\mathcal{E}_n^{(1)} = \left\{ \forall \{k, \ell\} \in \binom{[K]}{2}, \text{ s.t. } |\Lambda_{k,k} - \Lambda_{k,\ell}| > \gamma, \text{ it holds that } \left| \hat{\Lambda}_{k,k} - \hat{\Lambda}_{k,\ell} \right| > \frac{\gamma}{2} \right\};$$

$$\mathcal{E}_n^{(2)} = \left\{ \forall k \in [K], |\hat{n}_k - n_k| \leq n_k^{2/3} \right\}.$$

Combining (17)–(19), we see that if for each $k \in [K]$, $n_k = \Theta(n)$, $\min_k m_k = \omega(\sqrt{n_k} \log(n_k))$, then for sufficiently large n ,

$$\mathbb{P} \left((\mathcal{E}_n^{(1)} \cup \mathcal{E}_n^{(2)})^c \right) \leq e^{-2 \log n}. \quad (20)$$

We are now ready to prove Theorem 9, proving the consistency of \mathcal{L}^{ML} when the model parameters are unknown.

Proof [Proof of Theorem 9] Let \tilde{B} be our estimate of B using the seed vertices; i.e., there are \hat{n}_k vertices from block k for each $k \in [K]$, and for each $k, \ell \in [K]$, the entry of \tilde{B} between a block k vertex and a block ℓ vertex is

$$\log \left(\frac{\hat{\Lambda}_{k,\ell}}{1 - \hat{\Lambda}_{k,\ell}} \right).$$

Let \tilde{L} be the set of distinct entries of $\hat{\Lambda}$, and define

$$\hat{\alpha} = \min_{\{k,\ell\} \text{ s.t. } k \neq \ell} \left| \hat{\Lambda}_{k,k} - \hat{\Lambda}_{k,\ell} \right| \quad \hat{\beta} = \min_{\{k,\ell\} \text{ s.t. } k \neq \ell} \left| \hat{B}_{k,k} - B_{k,\ell} \right| \quad \hat{c} = \max_{i,j,k,\ell} \left| \hat{B}_{i,j} - \hat{B}_{k,\ell} \right|, \quad (21)$$

$$\hat{\gamma} = \min_{x,y \in \tilde{L}} |x - y|, \quad \hat{\kappa} = \min_{x,y \in \tilde{L}} \left| \log \left(\frac{x}{1-x} \right) - \log \left(\frac{y}{1-y} \right) \right|. \quad (22)$$

Note that conditioning on $\mathcal{E}_n^{(1)} \cup \mathcal{E}_n^{(2)}$ and assumption *iv*, ensures that each of $\hat{\alpha}$, $\hat{\beta}$, \hat{c} , $\hat{\gamma}$, and $\hat{\kappa}$ is bounded away from 0 by an absolute constant for sufficiently large n . For each $k \in [K]$, define

$$\epsilon_k := |\hat{n}_k - n_k| = |\hat{u}_k - u_k|, \quad \epsilon = \sum_k \epsilon_k, \quad \eta_k := \min(n_k, \hat{n}_k), \quad \eta = \sum_k \eta_k, \quad (23)$$

and note that conditioning on $\mathcal{E}_n^{(1)} \cup \mathcal{E}_n^{(2)}$ ensures that $\epsilon_k = O(n_k^{2/3})$ for all $k \in [K]$. An immediate result of this is that, conditioning on $\mathcal{E}_n^{(1)} \cup \mathcal{E}_n^{(2)}$, we have that $\eta_k = \Theta(n_k) = \Theta(n)$ for all $k \in [K]$.

Define $\mathcal{P} := \{P \in \Pi_u : \epsilon_{1,\bullet}(P) > n^{2/3} \log n\}$, and for $P \in \Pi_u$, define

$$X_P := \operatorname{tr}(A\tilde{B}^\top) - \operatorname{tr}(A(I_m \oplus P)\tilde{B}(I_m \oplus P)^\top).$$

We will show that

$$\mathbb{P}(\exists P \in \mathcal{P} \text{ s.t. } X_P \leq 0) = O(1/n^2),$$

and the desired consistency of \mathcal{L}^{ML} follows immediately. To this end, decompose A and B as

$$A = \begin{bmatrix} \eta & \epsilon \\ A^{(c,c)} & A^{(c,e)} \end{bmatrix} \quad B = \begin{bmatrix} \eta & \epsilon \\ B^{(c,c)} & B^{(c,e)} \end{bmatrix},$$

where $A^{(c,c)}$ (resp., $B^{(c,c)}$) is an $\eta \times \eta$ submatrix of A (resp., B)—which contains the seed vertices in A —with exactly η_k vertices (resp., labels) from block k for each $k \in [K]$. We view $A^{(c,c)}$ as the “core” matrix of A (with $A^{(c,e)}$ and $A^{(e,c)}$ being the “errorful” part of A), as $A^{(c,c)}$ is a submatrix of A that we could potentially cluster perfectly along block assignments. Note that similarly decomposing P as

$$P = \begin{bmatrix} \eta & \epsilon \\ P^{(c,c)} & P^{(c,e)} \end{bmatrix},$$

we see that there exists a principal permutation submatrix of size $(\eta - 2\epsilon) \times (\eta - 2\epsilon)$, which we denote \tilde{P} (with associated permutation $\tilde{\sigma}$). This matrix represents a subgraph of the core vertices of A mapped to a subgraph of the core vertices in B . We can then write $P = \tilde{P} \oplus Q$, where $Q \in \Pi_{3\epsilon}$. For each $k, \ell \in [K]$, let

$$\tilde{\epsilon}_{k,\ell} = \tilde{\epsilon}_{k,\ell}(\tilde{P}) = |\{v \in U_k \text{ s.t. } \tilde{\sigma}(v) \in U_k\}|$$

Consider now

$$X_P = \operatorname{tr}(A(I_{\eta-3\epsilon} \oplus Q)B(I_{\eta-3\epsilon} \oplus Q)^\top) - \operatorname{tr}(A(\tilde{P} \oplus Q)B(\tilde{P} \oplus Q)^\top). \quad (24)$$

Letting \tilde{u}_k denote the number of vertices from the k -th block acted on by \tilde{P} , our assumptions yield

$$\mathbb{E}(X_P) \geq 2\hat{\alpha}\hat{\beta} \sum_k \left(\frac{(\tilde{u}_k - \bar{\epsilon}_{k,\bullet})\bar{\epsilon}_{k,\bullet}}{2} + m_k \bar{\epsilon}_{k,\bullet} \right) - \Theta(n\epsilon) - \Theta(\epsilon^2).$$

Let $\tilde{n}(P)$ be the total number of distinct entries of $\text{vec}(B^{(c,\epsilon)})$ permuted by \tilde{P} , and note that another application of Proposition 12 yields

$$\mathbb{E}(X_P) \geq \frac{1}{2} \tilde{n}(P) \bar{\gamma} \bar{\kappa} - \Theta(n\epsilon) - \Theta(\epsilon^2).$$

The assumptions in the Theorem also immediately yield that

$$\tilde{n}(P) \geq \sum_k \left(\frac{(\tilde{u}_k - \bar{\epsilon}_{k,\bullet})\bar{\epsilon}_{k,\bullet}}{2} + m_k \bar{\epsilon}_{k,\bullet} \right).$$

We then have that there exists a constants $c_1 > 0$ and $c_2 > 0$ such that

$$\begin{aligned} \mathbb{P}(\exists P \in \mathcal{P} \text{ s.t. } X_P \leq 0 \mid \mathcal{E}_n^{(1)} \cup \mathcal{E}_n^{(2)}) &= \mathbb{P}(\exists P \in \mathcal{P} / \sim \text{ s.t. } X_P \leq 0 \mid \mathcal{E}_n^{(1)} \cup \mathcal{E}_n^{(2)}) \\ &\leq \exp \left\{ -\frac{\hat{\alpha}\hat{\beta}\bar{\kappa}\bar{\gamma}}{2c_2^2} \sum_k \left(\frac{(\tilde{u}_k - \bar{\epsilon}_{k,\bullet})\bar{\epsilon}_{k,\bullet}}{2} + m_k \bar{\epsilon}_{k,\bullet} \right) + \Theta(n\epsilon) + \Theta(\epsilon^2) \right\} \\ &\quad + \sum_k (\tilde{u}_k - \bar{\epsilon}_{k,k}) \log \tilde{u}_k + \sum_{k \text{ s.t. } \bar{\epsilon}_{k,\bullet} \neq 0} K \log(\tilde{u}_k + K) + O(\epsilon \log \epsilon) \\ &= \exp \left\{ -c_1 \sum_k \left(\frac{(\tilde{u}_k - \bar{\epsilon}_{k,\bullet})\bar{\epsilon}_{k,\bullet}}{2} + m_k \bar{\epsilon}_{k,\bullet} \right) \right. \\ &\quad \left. + \sum_k \bar{\epsilon}_{k,\bullet} \log \tilde{u}_k + \sum_{k \text{ s.t. } \bar{\epsilon}_{k,\bullet} \neq 0} K \log(\tilde{u}_k + K) + \Theta(n\epsilon) \right\} \\ &\leq \exp\{-c_2 n^{7/4} \log n\}. \end{aligned} \tag{25}$$

Unconditioning Equation (25) combined with Equation (20) yields the desired result. ■

Proof [Proof of Theorem 10 (Sketch)] The proof of Theorem 10 is a straightforward combination of the proofs of Theorems 8 and 9 once we have defined

$$P := \{P \in \Pi_n : \epsilon_{1,\bullet}(P) > n^{8/9} \log n\}.$$

Details are omitted for the sake of brevity. ■

References

L. A. Adamic and N. Glance. The political blogosphere and the 2004 US election. In *Proc. WWW-2005 Workshop on the Weblogging Ecosystem*, 2005.

E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *The Journal of Machine Learning Research*, 9:1981–2014, 2008.

E. M. Airoldi, T. B. Costa, and S. H. Chan. Stochastic blockmodel approximation of a graphon: Theory and consistent estimation. *Advances in Neural Information Processing Systems*, 26:692–700, 2013.

P. J. Bickel and A. Chen. A nonparametric view of network models and Newman-Girvan and other modularities. *Proc. National Academy of Sciences*, 106:21068–21073, 2009.

E. Bullmore and O. Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*, 10(3):186–198, 2009.

P. J. Carrington, J. Scott, and S. Wasserman. *Models and Methods in Social Network Analysis*. Cambridge University Press, 2005.

G. A. Coppersmith. Vertex nomination. *Wiley Interdisciplinary Reviews: Computational Statistics*, 6(2):144–153, 2014.

G. A. Coppersmith and C. E. Priebe. Vertex nomination via content and context. *arXiv preprint arXiv:1201.4118*, 2012.

M. Fiori, P. Sprechmann, J. T. Vogelstein, P. Mus, and G. Sapiro. Robust multimodal graph matching: Sparse coding meets graph matching. *Advances in Neural Information Processing Systems*, pages 127–135, 2013.

D. E. Fishkind, S. Adali, and C. E. Priebe. Seeded graph matching. *arXiv preprint arXiv:1209.0367*, 2012.

D. E. Fishkind, V. Lyzinski, H. Pao, L. Chen, and C. E. Priebe. Vertex nomination schemes for membership prediction. *The Annals of Applied Statistics*, 9(3):1510–1532, 2015.

P. Foglia, G. Percannella, and M. Vento. Graph matching and learning in pattern recognition in the last 10 years. *International Journal of Pattern Recognition and Artificial Intelligence*, 28(01):1450001, 2014.

B. Franke and P. J. Wolfe. Network modularity in the presence of covariates. *arXiv preprint arXiv:1603.01214*, 2016.

L. Hubert and P. Arabie. Comparing partitions. *J. Classification*, 2:193–218, 1985.

H. Jeong, S. P. Mason, A.-L. Barabási, and Z. N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41–42, 2001.

R. Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340, 1987.

A. Kandel, H. Bunke, and M. Last. *Applied Graph Theory in Computer Vision and Pattern Recognition*, volume 1. Springer, 2007.

- B. Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Physical Review E*, 83, 2011.
- H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97, 1955.
- U. Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.
- V. Lyzinski, D. E. Fishkind, and C. E. Priebe. Seeded graph matching for correlated Erdős-Rényi graphs. *Journal of Machine Learning Research*, 15:3513–3540, 2014a.
- V. Lyzinski, D. L. Sussman, M. Tang, A. Athreya, and C. E. Priebe. Perfect clustering for stochastic blockmodel graphs via adjacency spectral embedding. *Electronic Journal of Statistics*, 8:2905–2922, 2014b.
- D. Marchette, C. E. Priebe, and G. A. Coppersmith. Vertex nomination via attributed random dot product graphs. In *Proceedings of the 57th ISI World Statistics Congress*, volume 6, page 16, 2011.
- M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E*, 74(3):036104, 2006a.
- M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006b.
- M. E. J. Newman. Community detection in networks: Modularity optimization and maximum likelihood are equivalent. *arXiv preprint arXiv:1606.02319*, 2016.
- M. E. J. Newman and A. Clauset. Structure and inference in annotated networks. *Nature Communications*, 7(11863), 2016.
- M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):1–15, February 2004. ISSN 1539-3755.
- W. De Nooy, A. Mrvar, and V. Batagelj. *Exploratory social network analysis with Pajek*. Cambridge University Press, 2011.
- S. C. Olhede and P. J. Wolfe. Network histograms and universality of block model approximation. *Proceedings of the National Academy of Sciences*, 111:14722–14727, 2014.
- P. Resnick and H. R. Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.
- F. Ricci, L. Rokach, and B. Shapira. *Introduction to recommender systems handbook*. Springer, 2011.
- K. Rohe, S. Chatterjee, and B. Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *Annals of Statistics*, 39:1878–1915, 2011.

- D. L. Sussman, M. Tang, D. E. Fishkind, and C. E. Priebe. A consistent adjacency spectral embedding for stochastic blockmodel graphs. *Journal of the American Statistical Association*, 107(499):1119–1128, 2012.
- R. E. Ulanowicz, C. Bondavalli, and M. S. Egnotovich. Network analysis of trophic dynamics in South Florida ecosystems, FY 97: The Florida Bay ecosystem. Annual Report to the U.S. Geological Survey, Biological Resources Division. Ref. No. [UMCES]CBL 98-123, 1997.
- J. T. Vogelstein, J. M. Conroy, V. Lyzinski, L. J. Podrazik, S. G. Kratzer, E. T. Harley, D. E. Fishkind, R. J. Vogelstein, and C. E. Priebe. Fast Approximate Quadratic Programming for Graph Matching. *PLoS ONE*, 10(04), 2014.
- S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.
- J. Yang and J. Leskovec. Defining and evaluating network communities based on ground truth. *Knowledge and Information Systems*, 42(1):181–213, 2015.
- J. Yang, J. McAuley, and J. Leskovec. Community detection in networks with node attributes. In *Proc. IEEE 13th International Conference on Data Mining*, pages 1151–1156, 2013.
- W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):452–473, 1977.
- M. Zaslavskiy, F. Bach, and J.P. Vert. A path following algorithm for the graph matching problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12):2227–2242, 2009.
- Y. Zhang, E. Levina, and J. Zhu. Community detection in networks with node features. *arXiv preprint arXiv:1509.01173*, 2015.

Characteristic Kernels and Infinitely Divisible Distributions

Yu Nishiyama

The University of Electro-Communications

1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan

YNISHIYAM@GMAIL.COM

Kenji Fukumizu

The Institute of Statistical Mathematics

10-3 Midori-cho, Tachikawa, Tokyo 190-8562, Japan

FUKUMIZU@ISM.AC.JP

Editor: Ingo Steinwart

Abstract

We connect shift-invariant characteristic kernels to infinitely divisible distributions on \mathbb{R}^d . Characteristic kernels play an important role in machine learning applications with their kernel means to distinguish any two probability measures. The contribution of this paper is twofold. First, we show, using the Lévy–Khintchine formula, that any shift-invariant kernel given by a bounded, continuous, and symmetric probability density function (pdf) of an infinitely divisible distribution on \mathbb{R}^d is characteristic. We mention some closure properties of such characteristic kernels under addition, pointwise product, and convolution. Second, in developing various kernel mean algorithms, it is fundamental to compute the following values: (i) kernel mean values $m_P(x)$, $x \in \mathcal{X}$, and (ii) kernel mean RKHS inner products $\langle m_P, m_Q \rangle_{\mathcal{H}}$, for probability measures P, Q . If P, Q , and kernel k are Gaussians, then the computation of (i) and (ii) results in Gaussian pdfs that are tractable. We generalize this Gaussian combination to more general cases in the class of infinitely divisible distributions. We then introduce a *conjugate* kernel and a *convolution trick* so that the above (i) and (ii) have the same pdf form, expecting tractable computation at least in some cases. As specific instances, we explore α -stable distributions and a rich class of generalized hyperbolic distributions, where the Laplace, Cauchy, and Student's t distributions are included.

Keywords: Characteristic Kernel, Kernel Mean, Infinitely Divisible Distribution, Conjugate Kernel, Convolution Trick

1. Introduction

Let $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$ be a measurable space and $\mathcal{M}_1(\mathcal{X})$ be the set of probability measures. Let \mathcal{H} be the real-valued reproducing kernel Hilbert space (RKHS) associated with a bounded and measurable positive-definite (p.d.) kernel $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$. In machine learning, kernel methods provide a technique for developing nonlinear algorithms, by mapping data X_1, \dots, X_n in \mathcal{X} to higher- or infinite-dimensional RKHS functions $k(\cdot, X_1), \dots, k(\cdot, X_n)$ in \mathcal{H} (Schölkopf and Smola, 2002; Steinwart and Christmann, 2008).

Recently, an RKHS representation of a probability measure $P \in \mathcal{M}_1(\mathcal{X})$, called kernel mean, $m_P := \mathbb{E}_{X \sim P}[k(\cdot, X)] \in \mathcal{H}$ (Smola et al., 2007; Fukumizu et al., 2013), or equivalently,

$$m_P(x) = \int k(x, y) dP(y), \quad x \in \mathcal{X} \quad (1)$$

has been used to handle probability measures in RKHSs. The kernel mean enables us to introduce a similarity and distance between two probability measures $P, Q \in \mathcal{M}_1(\mathcal{X})$, via the RKHS inner product $\langle m_P, m_Q \rangle_{\mathcal{H}}$ and the norm $\|m_P - m_Q\|_{\mathcal{H}}$, respectively. Using these quantities, different authors have proposed many algorithms, including density estimations (Smola et al., 2007; Song et al., 2008; McCalman et al., 2013), hypothesis tests (Gretton et al., 2012; Gretton et al., 2008; Fukumizu et al., 2008), kernel Bayesian inference (Song et al., 2009; Song et al., 2010; Song et al., 2011; Fukumizu et al., 2013; Song et al., 2013; Kanagawa et al., 2016; Nishiyama et al., 2016), classification (Muandet et al., 2012), dimension reduction (Fukumizu and Leng, 2012), and reinforcement learning (Grünewälder et al., 2012; Nishiyama et al., 2012, Rawlik et al., 2013, Boots et al., 2013).

In these applications, the characteristic property of a p.d. kernel k is important: a p.d. kernel is said to be *characteristic* if any two probability measures $P, Q \in \mathcal{M}_1(\mathcal{X})$ can be distinguished by their kernel means $m_P, m_Q \in \mathcal{H}$ (Fukumizu et al., 2004; Sriperumbudur et al., 2010, 2011). For a continuous, bounded, and shift-invariant p.d. kernel on \mathbb{R}^d with $k(x, y) = \kappa(x - y)$, a necessary and sufficient condition for the kernel to be characteristic is known via the Bochner theorem (Sriperumbudur et al., 2010, Theorem 9).

As the first contribution of this paper, we show, using the Lévy–Khintchine formula (Sato, 1999; F. W. Steutel, 2004; Applebaum, 2009), that if κ is a continuous, bounded, and symmetric pdf of an infinitely divisible distribution P on \mathbb{R}^d , then k is a characteristic p.d. kernel. We call such kernels *convolutionally infinitely divisible* (CID) kernels. Examples of CID kernels are given in Example 3.4. In addition, we note some closure properties of the CID kernels with respect to addition, pointwise product, and convolution.

To describe the second contribution, we briefly explain what is essentially computed in kernel mean algorithms. In general kernel methods, the following computations are fundamental:

- (i) RKHS function values: $f(x)$ for $f \in \mathcal{H}$, $x \in \mathcal{X}$,
- (ii) RKHS inner products: $\langle f, g \rangle_{\mathcal{H}}$, $f, g \in \mathcal{H}$.

If $f \in \mathcal{H}$ is represented by $f := \sum_{i=1}^n w_i k(\cdot, X_i)$, $w \in \mathbb{R}^n$, then the function value (i) $f(x) = \sum_{i=1}^n w_i k(x, X_i)$ reduces to the evaluation of the kernel $k(x, y)$. Similarly, if two RKHS functions $f, g \in \mathcal{H}$ are both represented by $f := \sum_{i=1}^n w_i k(\cdot, X_i)$ and $g := \sum_{j=1}^l \tilde{w}_j k(\cdot, \tilde{X}_j)$, respectively, then the inner product (ii) $\langle f, g \rangle_{\mathcal{H}} = \sum_{i=1}^n \sum_{j=1}^l w_i \tilde{w}_j k(X_i, \tilde{X}_j)$ reduces to the evaluation of the kernel $k(x, y)$, which is so-called the *kernel trick* $\langle k(\cdot, x), k(\cdot, y) \rangle_{\mathcal{H}} = k(x, y)$.

We consider a more general case in which $f, g \in \mathcal{H}$ are represented by $f := \sum_{i=1}^n w_i m_{P_i}$ and $g := \sum_{j=1}^l \tilde{w}_j m_{Q_j}$, respectively, where $\{m_{P_i}\}, \{m_{Q_j}\} \subset \mathcal{H}$ are kernel means of probability measures $\{P_i\}, \{Q_j\} \subset \mathcal{M}_1(\mathcal{X})$. Kernel algorithms involving kernel means use this type of RKHS functions explicitly or implicitly. If $\{P_i\}, \{Q_j\}$ are delta measures $\{\delta_{x_i}\}, \{\delta_{\tilde{x}_j}\}$, then these functions are specialized to the above kernel trick case, where $m_{\delta_x} = k(\cdot, x)$. The computation of (i) $f(x) = \sum_{i=1}^n w_i m_{P_i}(x)$ and (ii) $\langle f, g \rangle_{\mathcal{H}} = \sum_{i=1}^n \sum_{j=1}^l w_i \tilde{w}_j \langle m_{P_i}, m_{Q_j} \rangle_{\mathcal{H}}$ requires the following kernel mean evaluations:²

1. A probability measure $\delta_x(\cdot)$, $x \in \mathcal{X}$ is a delta measure; if $x \in B$, then $\delta_x(B) = 1$; otherwise, $\delta_x(B) = 0$ for $B \in \mathcal{B}(\mathcal{X})$.
2. If kernel means m_P, m_Q are also both expressed by a weighted sum, $m_P := \sum_{i=1}^n \eta_i k(\cdot, \tilde{X}_i)$ and $m_Q := \sum_{j=1}^l \tilde{\eta}_j k(\cdot, \tilde{X}_j)$, $\{\tilde{X}_i\}, \{\tilde{X}_j\} \subset \mathcal{X}$, then the computation also reduces to the above kernel trick case.

- (iii) kernel mean values: $m_P(x)$ for $P \in \mathcal{M}_1(\mathcal{X})$, $x \in \mathcal{X}$,
- (iv) kernel mean inner products: $\langle m_P, m_Q \rangle_{\mathcal{H}}$, $P, Q \in \mathcal{M}_1(\mathcal{X})$.

Note that the kernel mean value (1) and the kernel mean inner product $\langle m_P, m_Q \rangle_{\mathcal{H}} = \int k(x, y) dP(x) dQ(y)$ involve an integral, and their rigorous computation is not tractable in general.

The second contribution of this paper is to provide some classes of p.d. kernels and parametric models $P, Q \in \mathcal{P}_{\Theta} := \{P_{\theta} | \theta \in \Theta\}$ such that the kernel computation of (iii) and (iv) can be reduced to a kernel evaluation, where tractable computation is considered. For a shift-invariant kernel $k(x, y) = \kappa(x - y)$, $x, y \in \mathbb{R}^d$ on \mathbb{R}^d , as shown in Lemma 2.5, the computation of (iii) and (iv) reduces to the following convolution:

- (iii)' kernel mean values: $m_P(x) = (\kappa * P)(x)$,
- (iv)' kernel mean inner products: $\langle m_P, m_Q \rangle_{\mathcal{H}} = (\kappa * \tilde{P} * \tilde{Q})(0) = (\kappa * P * \tilde{Q})(0)$,

where \tilde{P} and \tilde{Q} are the dual of P and Q , respectively.³ This convolution representation motivates us to explore a set of parametric distributions \mathcal{P}_{Θ} that is closed under convolution, namely, a convolution semigroup $(\mathcal{P}_{\Theta}, *) \subset \mathcal{M}_1(\mathbb{R}^d)$, where κ is a density function in \mathcal{P}_{Θ} .

To illustrate the basic idea, let us consider Gaussian distributions \mathcal{P}_{Θ} as a parametric class, which is closed under convolution, and a Gaussian kernel. For simplicity, we consider the case of scalar variance matrices $\sigma^2 I_d$. Let $N_d(\mu, \sigma^2 I_d)$ and $f_d(x | \mu, \sigma^2 I_d)$ denote the d -dimensional Gaussian distribution with mean μ and variance-covariance matrix $\sigma^2 I_d$, and its pdf, respectively. If P and Q are Gaussian distributions $N_d(\mu_P, \sigma_P^2 I_d)$ and $N_d(\mu_Q, \sigma_Q^2 I_d)$, respectively, and k is given by the pdf $f_d(x - y | 0, \sigma^2 I_d)$, it is easy to see that $m_P(x) = f_d(x | \mu_P, (\sigma_P^2 + \sigma^2) I_d)$ and $\langle m_P, m_Q \rangle_{\mathcal{H}} = f_d(\mu_P - \mu_Q, (\sigma_P^2 + \sigma_Q^2 + \sigma^2) I_d)$. The kernel mean value and inner product are thus reduced to simply evaluating Gaussian pdfs, which are given by a parameter update following a specific rule. This type of computation appears in various applications: to list a few, Muanudet et al. (2012) proposed a support measure classification by considering kernels $k(P, Q)$ between two input probability measures P, Q , including Gaussian models; Song et al. (2008) and McCalmann et al. (2013) considered an approximation of a (target) probability measure P with a Gaussian mixture P_{θ} , via an optimization problem $\theta = \text{argmin}_{\theta} \|m_P - m_{P_{\theta}}\|_{\mathcal{H}}^2$. The parametric expression of (iii) and (iv) is especially useful for the optimization of θ in the class of distributions. Other such applications are given in Section 5.

We generalize this closedness or ‘‘conjugacy’’⁴ of Gaussians with respect to kernel means and explore other cases in CID kernels. We then introduce a *conjugate* kernel k to parametric models \mathcal{P}_{θ} and a *convolution trick*, so that (iii) and (iv) have the same density form, i.e., there is some parameter update in the class. If P, Q are delta measures δ_a, δ_b , then the convolution trick simplifies to the kernel trick. See Proposition 4.2 for a description.

While a general perspective is obtained from the convolution semigroup $(\mathbb{I}(\mathbb{R}^d), *)$ of infinitely divisible distributions, the pdfs of $\mathbb{I}(\mathbb{R}^d)$ are not tractable in general. We then

3. A probability measure $\tilde{P} \in \mathcal{M}_1(\mathbb{R}^d)$ is called a *dual* of $P \in \mathcal{M}_1(\mathbb{R}^d)$ if $\tilde{P}(B) = P(-B)$ for every $B \in \mathcal{B}(\mathbb{R}^d)$, where $-B := \{-x : x \in B\}$ (Sato, 1999, p.8)

4. Here, the term ‘‘conjugacy’’ is an analogy of the conjugate prior in the Bayes’ theorem, where the prior and posterior have the same pdf form in a probabilistic model.

explore smaller convolution sub-semigroups $(\mathcal{P}_{\Theta}, *) \subset (\mathbb{I}(\mathbb{R}^d), *)$ having a small number of parameters. In particular, we focus on the well-known α -stable distributions $S_{\alpha}(\mathbb{R}^d)$ for each $\alpha \in (0, 2]$ and generalized hyperbolic (GH) distributions $\mathbb{G}\mathbb{H}(\mathbb{R}^d)$, which include Laplace, Cauchy, and Student’s t distributions. For each $\alpha \in (0, 2]$, the class $S_{\alpha}(\mathbb{R}^d)$ is closed under convolution. The GH class has various convolutional properties, as given in Proposition 4.5. As in the Gaussian cases, the computation of (iii) and (iv) is realized by the evaluation of pdfs, i.e., evaluation of conjugate kernels, after a parameter update.

Unfortunately, these conjugate kernels are not generally tractable. However, we can find some subclasses of tractable conjugate kernels. See Section 6 for a discussion on the computation of conjugate kernels. Note that α -stable and GH distribution classes have many applications: applications of $S_{\alpha}(\mathbb{R}^d)$ are listed in Nolan (2013a), and the GH distributions have been applied, e.g., to mathematical finance with the Lévy processes (Schoutens, 2003; Cont and Tankov, 2004; Barndorff-Nielsen and Halgreen, 1990; Madan et al., 1998; Barndorff-Nielsen, 1998; Barndorff-Nielsen and Prause, 2001; Carr et al., 2002). Note also that the Matérn kernel (Rasmussen and Williams, 2006, Section 4.2.1), often used in machine learning, is included in this GH class.

The rest of this paper is organized as follows. In Section 2, we review the notions of kernel means, characteristic kernels, and related matters. In Section 3, we show that the CID kernels are characteristic p.d. kernels on \mathbb{R}^d . In addition, we present the closedness property with respect to addition, pointwise product, and convolution. In Section 4, we introduce the absorbing, conjugate kernel and convolution trick for convolution semigroups of infinitely divisible distributions. Section 5 lists some motivating examples of kernel machine algorithms involving kernel means and parametric models. Section 6 notes the computation of the pdfs of conjugate kernels to realize the convolution trick.

2. Preliminaries: Kernel Means and Characteristic Kernels

In this section, we review kernel means and characteristic kernels restricted to \mathbb{R}^d .

Let \mathbb{R}^d be the set of $d \times d$ p.d. matrices. Let $\|x\|_{\Sigma} = \sqrt{x^{\top} \Sigma x}$, $x \in \mathbb{R}^d$, and $\Sigma \in \mathbb{P}_d$. Let $L^1(\mathbb{R}^d)$ be the absolutely integrable function space on \mathbb{R}^d . Let $C_0(\mathbb{R}^d)$ be the continuous and bounded function space on \mathbb{R}^d .

A symmetric function $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is called a *p.d. kernel* on \mathbb{R}^d if, for any $n \in \mathbb{N}$, $x_1, \dots, x_n \in \mathbb{R}^d$, the matrix $G_{ij} = k(x_i, x_j)$, $i, j \in \{1, \dots, n\}$ is positive-semidefinite. Throughout this paper, we assume a p.d. kernel k is on \mathbb{R}^d . It is known (Aronszajn, 1950) that every p.d. kernel k has the unique RKHS \mathcal{H} , which is a Hilbert space of functions $f : \mathbb{R}^d \rightarrow \mathbb{R}$, satisfying the following: (i) $k(\cdot, x) \in \mathcal{H}$, $\forall x \in \mathbb{R}^d$, (ii) $\text{Span}\{k(\cdot, x) | x \in \mathbb{R}^d\}$ is dense in \mathcal{H} ; and (iii) the *reproducing property* holds:

$$f(x) = \langle f, k(\cdot, x) \rangle_{\mathcal{H}}, \quad \forall f \in \mathcal{H}, \quad \forall x \in \mathbb{R}^d,$$

where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ denotes the inner product of \mathcal{H} . The map $\Phi : \mathbb{R}^d \rightarrow \mathcal{H}; x \mapsto k(\cdot, x)$ is called a *feature map*.

A p.d. kernel k is called *bounded* if $\sup_{x \in \mathbb{R}^d} k(x, x) < \infty$. A p.d. kernel k is bounded if and only if every $f \in \mathcal{H}$ is bounded (Steinwart and Christmann, 2008, Lemma 4.23). A p.d. kernel k is called *separately continuous* if $k(\cdot, x) : \mathbb{R}^d \rightarrow \mathbb{R}$ is continuous for all $x \in \mathbb{R}^d$. A p.d. kernel k is bounded and separately continuous if and only if every $f \in \mathcal{H}$ is a bounded

and continuous function, i.e., $\mathcal{H} \subset C_b(\mathbb{R}^d)$, (Steinwart and Christmann, 2008, Lemma 4.28). A p.d. kernel k is called *continuous* if k is separately continuous and $x \mapsto k(x, x)$, $x \in \mathbb{R}^d$ is continuous (Steinwart and Christmann, 2008, Lemma 4.29). If a p.d. kernel k is continuous, the RKHS \mathcal{H} is separable (Steinwart and Christmann, 2008, Lemma 4.33).

A p.d. kernel k is called *shift-invariant* if there exists a function $\kappa : \mathbb{R}^d \rightarrow \mathbb{R}$ such that $k(x, y) = \kappa(x - y)$, $x, y \in \mathbb{R}^d$. The function κ is called a *p.d. function*. A p.d. function κ on \mathbb{R}^d is characterized by the Bochner theorem:

Theorem 2.1 (Bochner, 1959) (Wendland, 2005, Theorem 6.6) *A continuous function $\kappa : \mathbb{R}^d \rightarrow \mathbb{C}$ is positive definite if and only if it is the Fourier transform $\mathcal{F}(\Lambda)$ of a finite nonnegative Borel measure Λ on \mathbb{R}^d :*

$$\kappa(x) = \int_{\mathbb{R}^d} e^{\sqrt{-1}w^\top x} d\Lambda(w), \quad x \in \mathbb{R}^d.$$

Let $\mathcal{K}_{cb}(\mathbb{R}^d) \subset C_b(\mathbb{R}^d)$ denote the set of continuous bounded p.d. functions.

A p.d. kernel k is called *radial* if there exists a function $\tilde{\kappa} : [0, \infty) \rightarrow \mathbb{R}$ such that $k(x, y) = \tilde{\kappa}(\|x - y\|)$, $x, y \in \mathbb{R}^d$. A radial kernel k is given by

$$k(x, y) = \tilde{\kappa}(\|x - y\|) = \int_{[0, \infty)} e^{-t\|x-y\|} d\nu(t), \quad x, y \in \mathbb{R}^d, \quad (2)$$

where $\nu(t)$ is a finite nonnegative Borel measure on the Borel sets $\mathcal{B}([0, \infty))$. A p.d. kernel k is called *elliptical* if $k(x, y) = \tilde{\kappa}(\|x - y\|_\Sigma)$, $x, y \in \mathbb{R}^d$, $\Sigma \in \mathbb{P}^d$.

Let $\mathcal{M}_1(\mathbb{R}^d)$ be the set of Borel probability measures on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$. An RKHS element $m_P \in \mathcal{H}$ with a p.d. kernel k is called a *kernel mean* of a probability measure $P \in \mathcal{M}_1(\mathbb{R}^d)$ if there exists an expectation of the feature map:

$$m_P := \mathbb{E}_{X \sim P}[\Phi(X)] = \mathbb{E}_{X \sim P}[k(\cdot, X)] \in \mathcal{H}, \quad P \in \mathcal{M}_1(\mathbb{R}^d).$$

If k is a bounded and continuous p.d. kernel, then the feature map $\Phi : \mathbb{R}^d \rightarrow \mathcal{H}$ is Bochner P -integrable for all $P \in \mathcal{M}_1(\mathbb{R}^d)$, since $\mathbb{E}_{X \sim P}[\|k(\cdot, X)\|_{\mathcal{H}}] = \mathbb{E}_{X \sim P}[\sqrt{k(X, X)}] < \infty$ for all $P \in \mathcal{M}_1(\mathbb{R}^d)$ (Steinwart and Christmann, 2008, p. 510). Throughout this paper, we assume a bounded and continuous p.d. kernel k . We write $m_P := \{m_P\} \in \mathcal{P} \subset \mathcal{M}_1(\mathbb{R}^d)$.

As mentioned in the Introduction, there are many applications using m_P , since m_P enables us to introduce a similarity and distance between probability measures $P, Q \in \mathcal{M}_1(\mathbb{R}^d)$, via the Hilbert space inner product $\langle m_P, m_Q \rangle_{\mathcal{H}}$ and norm $\|m_P - m_Q\|_{\mathcal{H}}$, respectively, where the reproducing property is also exploited. In these applications, the characteristic kernel is important to distinguish any probability measures $P, Q \in \mathcal{M}_1(\mathbb{R}^d)$ by their kernel means $m_P, m_Q \in \mathcal{H}$. The following is the definition restricted to \mathbb{R}^d :

Definition 2.2 (Fukumizu et al., 2004)(Sriperumbudur et al., 2010, Definition 6) *A bounded and continuous p.d. kernel $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is called characteristic on \mathbb{R}^d if the kernel mean map $\mathcal{M}_1(\mathbb{R}^d) \rightarrow \mathcal{H}; P \mapsto m_P$ is injective, i.e., $m_P = m_Q$ implies $P = Q$ for any $P, Q \in \mathcal{M}_1(\mathbb{R}^d)$.*

Sriperumbudur et al. (2010) showed a necessary and sufficient condition for a shift-invariant p.d. kernel $k(x, y) = \kappa(x - y)$, $x, y \in \mathbb{R}^d$, $\kappa \in \mathcal{K}_{cb}(\mathbb{R}^d)$, to be characteristic via the Bochner theorem:

Theorem 2.3 (Sriperumbudur et al., 2010, Theorem 9) *A shift-invariant p.d. kernel k with $\kappa \in \mathcal{K}_{cb}(\mathbb{R}^d)$ is characteristic if and only if the finite nonnegative measure Λ in Theorem 2.1 has the entire support, $\text{supp}(\Lambda) = \mathbb{R}^d$.*

Let $\mathcal{K}_{cb}^{ch}(\mathbb{R}^d) \subset \mathcal{K}_{cb}(\mathbb{R}^d)$ denote the set of such characteristic p.d. functions on \mathbb{R}^d .

The convolution $f * g$ of two functions f and g is defined by $f * g := \int_{\mathbb{R}^d} f(\cdot - y)g(y)dy$. The convolution $f * Q$ of a function f and a probability measure $Q \in \mathcal{M}_1(\mathbb{R}^d)$ is defined by $f * Q := \int_{\mathbb{R}^d} f(\cdot - y)dQ(y)$. The convolution $P * Q$ of two probability measures $P, Q \in \mathcal{M}_1(\mathbb{R}^d)$ is defined by the probability measure $(P * Q)(B) := \int_{\mathbb{R}^d} P(B - x)dQ(x)$, where $B - x := \{z - x : z \in B\}$, $B \in \mathcal{B}(\mathbb{R}^d)$.

Given a function $f(x)$, $x \in \mathbb{R}^d$, the function \tilde{f} denotes $\tilde{f}(x) = f(-x)$, $x \in \mathbb{R}^d$. Given a probability measure $P \in \mathcal{M}_1(\mathbb{R}^d)$, a probability measure $\tilde{P} \in \mathcal{M}_1(\mathbb{R}^d)$ is called *dual* if $\tilde{P}(B) = P(-B)$, $B \in \mathcal{B}(\mathbb{R}^d)$, where $-B := \{-x : x \in B\}$ (Sato, 1999, p.8). A probability measure P is symmetric if $P = \tilde{P}$.

We have the following simple equalities:

Proposition 2.4 $\widetilde{f * g} = \tilde{f} * \tilde{g}$, $\widetilde{f * P} = \tilde{f} * \tilde{P}$, and $\widetilde{P * Q} = \tilde{P} * \tilde{Q}$.

Kernel mean m_P and RKHS inner product $\langle m_P, m_Q \rangle_{\mathcal{H}}$ have the following convolution representation:

Lemma 2.5 *Let k be a shift-invariant p.d. kernel with $\kappa \in C_b(\mathbb{R}^d)$. Then, we have the following:*

1. Kernel mean m_P is given by the convolution

$$m_P = \kappa * P \in \mathcal{H} \subset C_b(\mathbb{R}^d), \quad P \in \mathcal{M}_1(\mathbb{R}^d).$$
2. The RKHS inner product $\langle m_P, m_Q \rangle_{\mathcal{H}}$ is given by the convolution

$$\langle m_P, m_Q \rangle_{\mathcal{H}} = (\kappa * \tilde{P} * Q)(0) = (\kappa * P * \tilde{Q})(0), \quad P, Q \in \mathcal{M}_1(\mathbb{R}^d),$$
 where \tilde{P} and \tilde{Q} are the dual of P and Q , respectively.

Proof 1. Kernel mean m_P has the following convolution representation:

$$m_P = \int_{\mathbb{R}^d} k(x, \cdot) dP(x) = \int_{\mathbb{R}^d} \kappa(\cdot - x) dP(x) = \kappa * P, \quad P \in \mathcal{M}_1(\mathbb{R}^d).$$

Kernel mean $m_P \in \mathcal{H} \subset C_b(\mathbb{R}^d)$ exists for all $P \in \mathcal{M}_1(\mathbb{R}^d)$ because, for $\kappa \in C_b(\mathbb{R}^d)$, the feature map $\Phi : x \mapsto k(x, \cdot)$ is Bochner P -integrable for all $P \in \mathcal{M}_1(\mathbb{R}^d)$, as given in the definition of m_P .

2. RKHS inner product $\langle m_P, m_Q \rangle_{\mathcal{H}}$ has the following convolution representation:

$$\langle m_P, m_Q \rangle_{\mathcal{H}} = \int_{\mathbb{R}^d} m_P(y)dQ(y) = \int_{\mathbb{R}^d} \tilde{m}_P(-y)dQ(y) = (\tilde{m}_P * Q)(0) = (\kappa * \tilde{P} * Q)(0),$$

where we have used Proposition 2.4 and $\tilde{\kappa} = \kappa$ in the last equality. Since $\langle m_P, m_Q \rangle_{\mathcal{H}}$ is symmetric with respect to P and Q , then $(\kappa * \tilde{P} * Q)(0) = (\kappa * P * \tilde{Q})(0)$. This is also

obtained by $(\kappa * \tilde{P} * \tilde{Q})(0) = (\kappa * \widetilde{\tilde{P} * \tilde{Q}})(0) = (\kappa * P * \tilde{Q})(0)$. ■

In this paper, we simply consider that κ is a pdf of a probability distribution.⁵ Then, Lemma 2.5 motivates us to explore the set of probability distributions $\mathcal{P}_\Theta \subset \mathcal{M}_1(\mathbb{R}^d)$ that is closed under convolution, i.e., convolution semigroup $(\mathcal{P}_\Theta, *)$.

3. Characteristic kernels and infinitely divisible distributions

In this section, we introduce CID kernels, which are defined by infinitely divisible distributions, and show that they are characteristic (Section 3.1). In addition, we examine some closure properties of CID kernels with respect to addition, pointwise product, and convolution (Section 3.2).

3.1 Convolutionally Infinitely Divisible kernels

We review the infinite divisibility of a probability measure (Sato, 1999; F. W. Steutel, 2004; Applebaum, 2009).

Definition 3.1 (Sato, 1999, Definition 7.1, p. 31) *A probability measure $P \in \mathcal{M}_1(\mathbb{R}^d)$ is called infinitely divisible if, for any integer $n \in \mathbb{N}$, there exists a probability measure $P_n \in \mathcal{M}_1(\mathbb{R}^d)$ such that $P = P_n^{*n}$.*

The support of every infinitely divisible distribution P is unbounded except for delta measures $\{\delta_x(\cdot) \mid x \in \mathbb{R}^d\}$ (Sato, 1999, Examples 7.2, p. 31). Let $\mathbb{I}(\mathbb{R}^d)$ denote the set of infinitely divisible distributions on \mathbb{R}^d , $\mathbb{I}(\mathbb{R}^d)$ is closed under convolution. Every infinitely divisible distribution $P \in \mathbb{I}(\mathbb{R}^d)$ has the following unique Lévy–Khintchine representation for the characteristic function. Let $x \wedge y = \min\{x, y\}$; $x, y \in \mathbb{R}$. Let 1_B denote the indicator function on \mathbb{R}^d with $B \subset \mathbb{R}^d$.

Theorem 3.2 (Sato, 1999, Theorem 8.1, p. 37) *The characteristic function $\hat{P}(w)$ of an infinitely divisible distribution $P \in \mathbb{I}(\mathbb{R}^d)$ has the following unique representation:*

$$\hat{P}(w) = \exp \left(iw^\top \gamma - \frac{1}{2} w^\top A w + \int_{\mathbb{R}^d} \left(e^{i w^\top x} - 1 - i w^\top x 1_{\{|x| \leq 1\}}(x) \right) \nu(dx) \right), \quad w \in \mathbb{R}^d, \quad (3)$$

where $\gamma \in \mathbb{R}^d$, $A \in \mathbb{R}^{d \times d}$, is a symmetric nonnegative-definite matrix and ν is a measure on \mathbb{R}^d satisfying

$$\nu(\{0\}) = 0 \quad \text{and} \quad \int_{\mathbb{R}^d} (|x|^2 \wedge 1) \nu(dx) < \infty. \quad (4)$$

5. In machine learning, normalized kernels $\bar{k}(x, y) := \frac{k(x, y)}{\sqrt{k(x, x)k(y, y)}}$ are often used (e.g., Gaussian kernels $\bar{k}(x, y) := \exp(-\frac{\|x-y\|_2^2}{2\sigma^2})$) (Steinwart and Christmann, 2008, Lemma 4.55). However, we consider here pdf kernels (e.g., Gaussian kernels $k(x, y) := \frac{1}{\sqrt{(2\pi\sigma^2)^d}} \exp(-\frac{\|x-y\|_2^2}{2\sigma^2})$) for the closedness of the pdfs of P and $m \cdot P$. A scalar multiplication ($c > 0$) changes as follows: $\bar{m} \cdot P := \mathbb{E}^{X \sim P}[\bar{k}(\cdot, X)] = c \mathbb{E}^{X \sim P}[k(\cdot, X)] = cm \cdot P$ and $(\bar{m} \cdot P, \bar{m} \cdot Q)_{\mathcal{H}} = c(m \cdot P, m \cdot Q)_{\mathcal{H}}$, where $(f, g)_{\mathcal{H}} = \int_{\mathcal{X}} f(x)g(x) \nu(x)$ (Berlinet and Thomas-Agnan, 2004, p.37).

Conversely, for any $\gamma \in \mathbb{R}^d$, symmetric nonnegative-definite matrix $A \in \mathbb{R}^{d \times d}$, and measure ν satisfying (4), there exists an infinitely divisible distribution $P \in \mathbb{I}(\mathbb{R}^d)$.

(A, ν, γ) is called the generating triplet of $P \in \mathbb{I}(\mathbb{R}^d)$. A is called the covariance matrix of the Gaussian factor of $P \in \mathbb{I}(\mathbb{R}^d)$, and ν is called the Lévy measure of $P \in \mathbb{I}(\mathbb{R}^d)$. Gaussians correspond to the generating triplet $(A, 0, \gamma)$. α -Stable distributions, including Cauchy distributions, correspond to generating triplet $(0, \nu, \gamma)$, where ν is the corresponding nonzero Lévy measure. The Lévy measure of the α -stable distributions is shown in Appendix A.

An infinitely divisible distribution $P \in \mathbb{I}(\mathbb{R}^d)$ is symmetric if and only if $(A, \nu, \gamma) = (A, \nu_s, 0)$, where ν_s is a symmetric Lévy measure⁶ (Sato, 1999, p.114). Let $\mathbb{IS}(\mathbb{R}^d)$ denote the set of symmetric and infinitely divisible distributions on \mathbb{R}^d . $\mathbb{IS}(\mathbb{R}^d)$ is closed under convolution. Let $\mathcal{K}_{\text{cid}}^{\text{id}}(\mathbb{R}^d) \subset C_b(\mathbb{R}^d) \cap L^1(\mathbb{R}^d)$ denote the set of continuous and bounded pdfs⁷ of symmetric infinitely divisible distributions $\mathbb{IS}(\mathbb{R}^d)$:

$$\mathcal{K}_{\text{cid}}^{\text{id}}(\mathbb{R}^d) := \{\Xi(P_S) \in C_b(\mathbb{R}^d) \mid P_S \in \mathbb{IS}(\mathbb{R}^d)\},$$

where $\Xi : \mathcal{M}_1(\mathbb{R}^d) \rightarrow L^1(\mathbb{R}^d)$ is a function that maps a probability measure P to its pdf f if it exists.

The infinitely divisible pdf $\kappa \in \mathcal{K}_{\text{cid}}^{\text{id}}(\mathbb{R}^d)$ can be used for a characteristic kernel as follows.

Theorem 3.3 *The function $k(x, y) = \kappa(x - y)$, $x, y \in \mathbb{R}^d$, $\kappa \in \mathcal{K}_{\text{cid}}^{\text{id}}(\mathbb{R}^d)$ is a p.d. and characteristic kernel, i.e., $\mathcal{K}_{\text{cid}}^{\text{id}}(\mathbb{R}^d) \subset \mathcal{K}_{\text{cb}}^{\text{id}}(\mathbb{R}^d)$.*

Proof A probability measure P on \mathbb{R}^d is symmetric if and only if the characteristic function $\hat{P}(w)$, $w \in \mathbb{R}^d$ is real valued (Sato, 1999, p.67). If P is symmetric and infinitely divisible, $\hat{P}(w) > 0$ for every $w \in \mathbb{R}^d$ from the Lévy–Khintchine formula (3). Since $\hat{P}(w)$ is positive and has the entire support, $\text{supp}(\hat{P}(w)) = \mathbb{R}^d$, then k is a p.d. and characteristic kernel from Theorem 2.3. ■

We call a p.d. kernel k in Theorem 3.3 a convolutionally infinitely divisible (CID) kernel⁸. CID kernels include the following examples:

Example 3.4 (CID p.d. kernels) *CID kernels include Gaussian kernels, Laplace kernels, Cauchy kernels, α -stable kernels for each $\alpha \in (0, 2]$ ($\alpha = 2$ corresponds to Gaussian kernels (Grosswald, 1976), GH kernels, normalized inverse Gaussian α -stable kernels, Student's t kernels (Grosswald, 1976), GH kernels, normalized inverse Gaussian (NIG) kernels, variance gamma (VG) kernels (Matérn kernel is a special case of this), tempered α -stable (ToS) kernels (Rauchen et al., 2011; Rosinski, 2007; Bianchi et al., 2010), etc.*

6. A symmetric Lévy measure is a Lévy measure such that $\nu_s(B) = \nu_s(-B)$ for $\forall B \in \mathcal{B}(\mathbb{R}^d)$.
7. A necessary and sufficient condition for $P \in \mathbb{IS}(\mathbb{R}^d)$ to have a pdf is not known (Sato, 1999, p.177). If the Gaussian factor $A \in \mathbb{R}^{d \times d}$ is full rank, then $P \in \mathbb{I}(\mathbb{R}^d)$ has the pdf. If $A = 0$, see some sufficient conditions (Sato, 1999, Theorem 27.7, 27.10). Every nondegenerate self-decomposable distribution on \mathbb{R}^d has the pdf (Sato, 1999, Theorem 27.13).
8. The term “infinite divisibility” of a p.d. kernel is used in the pointwise product sense (Berg et al., 1984, Definition 2.6, p. 76), i.e., a p.d. kernel $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{C}$ on a nonempty set \mathcal{X} is called infinitely divisible if, for every $n \in \mathbb{N}$, there exists a p.d. kernel $k_n : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{C}$ such that $k = (k_n)^{*n}$. The CID kernel considered here is the convolution sense $\kappa = (\kappa_n)^{*n}$.

3.2 Closure Property

In this subsection, we note some closure properties of CID and characteristic kernels with respect to addition, pointwise product, and convolution. The closure property is used, e.g., to generate a new CID and characteristic kernel. Example 3.8 shows such an example.

It is known that the set of continuous and bounded p.d. kernels $\mathcal{K}_{cb}(\mathbb{R}^d)$ is closed under addition and pointwise product as follows (Steinwart and Christmann, 2008, p. 114):

Proposition 3.5 *If $\kappa_1, \kappa_2 \in \mathcal{K}_{cb}(\mathbb{R}^d)$, then $\kappa_1 + \kappa_2 \in \mathcal{K}_{cb}(\mathbb{R}^d)$ and $\kappa_1 \kappa_2 \in \mathcal{K}_{cb}(\mathbb{R}^d)$.*

Similarly, the set of characteristic kernels $\mathcal{K}_{cb}^{ch}(\mathbb{R}^d)$ is closed under addition and pointwise product as follows (Striperumbudur et al., 2010, Corollary 11):

Proposition 3.6 *If $\kappa \in \mathcal{K}_{cb}^{ch}(\mathbb{R}^d)$, $\kappa_1, \kappa_2 \in \mathcal{K}_{cb}(\mathbb{R}^d)$, and $\kappa_2 \neq 0$, then $\kappa + \kappa_1, \kappa \kappa_2 \in \mathcal{K}_{cb}^{ch}(\mathbb{R}^d)$.*

The set of CID kernels $\mathcal{K}_{cb}^{cid}(\mathbb{R}^d)$ is closed under convolution but not closed under addition or pointwise product.

Proposition 3.7 *Let $\kappa_1, \kappa_2 \in \mathcal{K}_{cb}^{cid}(\mathbb{R}^d)$. Then, we have the following:*

1. *Convolution $\kappa_1 * \kappa_2 \in \mathcal{K}_{cb}^{cid}(\mathbb{R}^d)$.*
2. *Addition $\kappa_1 + \kappa_2$ and product $\kappa_1 \kappa_2$ do not necessarily belong to $\mathcal{K}_{cb}^{cid}(\mathbb{R}^d)$, although they are characteristic, $\kappa_1 + \kappa_2, \kappa_1 \kappa_2 \in \mathcal{K}_{cb}^{ch}(\mathbb{R}^d)$.*

Proof 1. Let $\kappa_1 = \Xi(P_1)$ and $\kappa_2 = \Xi(P_2)$. Then, $\kappa_1 * \kappa_2 = \Xi(P_1 * P_2)$. If $P_1, P_2 \in \mathbb{IS}(\mathbb{R}^d)$ are absolutely continuous and symmetric infinitely divisible measures, so is $P_1 * P_2 \in \mathbb{IS}(\mathbb{R}^d)$.

2. A mixture of two infinitely divisible distributions is not necessarily infinitely divisible. A product of two infinitely divisible distributions is not necessarily infinitely divisible. The counter-examples are as follows. Let $\kappa_1(x) = e^{-|x|}$ and $\kappa_2(x) = e^{-x^2}$, $x \in \mathbb{R}$, be p.d. functions of Laplace and Gaussian kernels, respectively. Then, the product $k(x) \propto e^{-|x|} e^{-x^2}$ is not infinitely divisible (F. W. Steutel, 2004, Example 11.1.3), although it is characteristic (Proposition 3.6). Let $\kappa_1(x) = \frac{1}{4\sqrt{\pi}} e^{-\frac{1}{4}x^2}$ and $\kappa_2(x) = \frac{1}{4\sqrt{2\pi}} e^{-\frac{1}{8}x^2}$, $x \in \mathbb{R}$, be Gaussian kernels; then, the addition $\kappa_1 + \kappa_2$ is not infinitely divisible (F. W. Steutel, 2004, Example 11.15), although it is characteristic (Proposition 3.6). Many examples can be found in F. W. Steutel (2004). ■

As given in Proposition 3.7, the infinite divisibility is not closed under mixing in general, although some special mixing cases preserve it (F. W. Steutel, 2004, Chapter 7). The *normal mean-variance mixture* with an infinitely divisible mixing distribution, given in Lemma 4.4, is one of them.

New CID kernels and characteristic kernels may be generated by using these closure properties. If $\kappa = \mathcal{F}(\tilde{\kappa})$ is an infinitely divisible pdf with the characteristic function $\tilde{\kappa}$, then symmetrization $\kappa^* := \kappa * \tilde{\kappa} = \mathcal{F}(|\tilde{\kappa}|^2)$ and positive powers $(\kappa^*)^{*\lambda} = \mathcal{F}(|\tilde{\kappa}|^{2\lambda})$ ($\lambda > 0$) are also infinitely divisible pdfs. The following example shows that the Laplace and symmetric Gamma kernels are CID kernels generated from an exponential distribution.

Example 3.8 (F. W. Steutel, 2004, Example 2.9) *An exponential distribution P with the pdf $\kappa(x) = \alpha \exp(-\alpha x) 1_{[0, \infty)}(x)$, $\alpha > 0$ is infinitely divisible. The dual is $\tilde{\kappa}(x) = \alpha \exp(\alpha x) 1_{(-\infty, 0)}(x)$.*

1. *The symmetrization $\kappa^* = \kappa * \tilde{\kappa}$ has the characteristic function $\tilde{\kappa}^*(w) = \tilde{\kappa}(w) \tilde{\kappa}(w) = \frac{\alpha}{\alpha^2 + w^2}$. This is a Laplace pdf $\kappa^*(x) = \frac{\alpha}{2} \exp(-\alpha|x|)$.*

2. *Positive powers $(\kappa^*)^{*\lambda}$ ($\lambda > 0$) have the characteristic functions $(\tilde{\kappa}^*)^\lambda(w) = (\frac{\alpha^2}{\alpha^2 + w^2})^\lambda$. If $\lambda = 1$, the pdf is the above Laplace case. If $\lambda = 2$, the pdf is given by $(\kappa^*)^{*2}(x) = \frac{\alpha}{4}(1 + \alpha|x|) \exp(-\alpha|x|)$. For general $\lambda > 0$, the pdf is given by*

$$f(x) = \frac{\alpha^{2\lambda}}{\sqrt{\pi}(2\alpha)^{\lambda - \frac{1}{2}} \Gamma(\lambda)} |x - \mu|^{\lambda - \frac{1}{2}} K_{\lambda - \frac{1}{2}}(\alpha|x - \mu|), \quad x \in \mathbb{R}$$

where $\Gamma(\lambda)$ is the Gamma function and $K_\lambda(x)$ is the modified Bessel function of the third kind with index λ . This is the pdf of the zero-skewed VG distribution $VG_1(\lambda, \alpha, \beta = 0, \mu, 1)$ on \mathbb{R} , as given in Section 4.3.

The additions $(\kappa^*)^{*\lambda} + \tilde{\kappa}$, $\tilde{\kappa} \in \mathcal{K}_{cb}(\mathbb{R}^d)$, and products $(\kappa^*)^{*\lambda} \tilde{\kappa}$, $\tilde{\kappa} \in \mathcal{K}_{cb}^{ch}(\mathbb{R}^d)$, are characteristic kernels based on the closure properties.

4. Kernel Means and Infinitely Divisible Distributions

In this section, we examine the kernel means of a parametric class of distributions $\mathcal{P}_\Theta \subset \mathbb{I}(\mathbb{R}^d)$. As mentioned in the Introduction, we wish to compute (iii) kernel mean values $m_P(x)$, $x \in \mathbb{R}^d$ and (iv) RKHS inner products $\langle m_P, m_Q \rangle_{\mathcal{H}}$ for parametric models $P, Q \in \mathcal{P}_\Theta$. These form a basic computation for establishing kernel machine algorithms combining kernel means and parametric models. In Section 4.1, we introduce absorbing, conjugate kernels, and convolution trick in the set of infinitely divisible distributions $\mathbb{I}(\mathbb{R}^d)$. In Sections 4.2 and 4.3, we focus on well-known subclasses of α -stable distributions and GH distributions, which include Laplace, Cauchy, and Student's t distributions.

4.1 Absorbing, Conjugate Kernels, and Convolution Trick

We begin by introducing the notion of *absorbing* and *conjugate* p.d. kernels to particular sets of parametric models \mathcal{P}_Θ as follows:

Proposition 4.1 (absorbing & conjugate kernel) *Let $\mathcal{P}_\Theta, \mathcal{Q}_{\Theta'} \subset \mathcal{M}_1(\mathbb{R}^d)$ be two sets of parametric models such that $\mathcal{P}_\Theta * \mathcal{Q}_{\Theta'} \subseteq \mathcal{P}_\Theta$, where Θ and Θ' are finite or infinite index p.d. kernel. We have the following statements:*

1. *If $\kappa \in \Xi(\mathcal{P}_\Theta)$, then $m_{\mathcal{Q}_{\Theta'}} \in \Xi(\mathcal{P}_\Theta)$ holds. RKHS inner products $\langle m_P, m_Q \rangle_{\mathcal{H}}$, $P, Q \in \mathcal{P}_\Theta$ are values of pdfs in $\Xi(\mathcal{P}_\Theta)$.*
2. *If $\kappa \in \Xi(\mathcal{Q}_{\Theta'})$, then $m_{\mathcal{P}_\Theta} \in \Xi(\mathcal{P}_\Theta)$ holds. RKHS inner products $\langle m_P, m_Q \rangle_{\mathcal{H}}$, $P, Q \in \mathcal{P}_\Theta$ are not necessarily values of pdfs in $\Xi(\mathcal{P}_\Theta)$.*

Proof These statements are straightforward from Lemma 2.5 and assumptions. ■

Statements 1 and 2 indicate an *absorbing property* of k with respect to parametric models. If $\mathcal{P}_\Theta = \mathcal{Q}_{\Theta'}$ in Proposition 4.1, we call k (and, hence, its RKHS \mathcal{H}) a *conjugate* to \mathcal{P}_Θ . A general perspective may be given by the CID kernels, where these kernels are conjugate to $\mathbb{I}(\mathbb{R}^d)$ as follows:

Proposition 4.2 Let $k_{A,\nu_s}(x, y) = \kappa_{A,\nu_s}(x - y)$, $x, y \in \mathbb{R}^d$ be a CID kernel, where $\kappa_{A,\nu_s} \in \mathcal{K}_{\text{fid}}^d(\mathbb{R}^d)$ has a generating triplet $(A, \nu_s, 0)$, and let \mathcal{H}_{A,ν_s} be the RKHS given by κ_{A,ν_s} . Let $P, Q \in \mathbb{I}(\mathbb{R}^d)$ be infinitely divisible distributions with the generating triplets (A_P, ν_P, γ_P) and (A_Q, ν_Q, γ_Q) , respectively. Then, we have the following:

1. Kernel mean m_P is given by an infinitely divisible pdf:

$$\begin{aligned} m_P(\cdot) &= f(\cdot; A + A_P, \nu_s + \nu_P, \gamma_P), \quad f \in \Xi(\mathbb{I}(\mathbb{R}^d)) \\ &= k_{A+A_P, \nu_s + \nu_P}(\gamma_P, \cdot). \end{aligned}$$

2. The RKHS inner product $\langle m_P, m_Q \rangle_{\mathcal{H}_{A,\nu_s}}$ is given by

$$\begin{aligned} \langle m_P, m_Q \rangle_{\mathcal{H}_{A,\nu_s}} &= f(0; A + A_P + A_Q, \nu_s + \tilde{\nu}_P + \mu_Q, \gamma_Q - \gamma_P) \\ &= f(0; A + A_P + A_Q, \nu_s + \nu_P + \tilde{\nu}_Q, \gamma_P - \gamma_Q), \\ &= k_{A+A_P+A_Q, \nu_s + \nu_P + \tilde{\nu}_Q}(\gamma_P, \gamma_Q), \end{aligned}$$

where $\tilde{\nu}_P$ (respectively, $\tilde{\nu}_Q$) is the dual of the Lévy measure ν_P (respectively, ν_Q).

Proposition 4.2 indicates a general convolution trick. The computation of $\langle m_P, m_Q \rangle_{\mathcal{H}_{A,\nu_s}}$ is reduced to the computation of the same kernel $k_{A+A_P+A_Q, \nu_s + \nu_P + \tilde{\nu}_Q}$ with the updated parameters of the generating triplets. If Q is a delta measure δ_y (i.e., $A_Q = 0$, $\nu_Q = 0$, $\gamma_Q = y$), then statement 2 is specialized to statement 1. If P, Q are both delta measures δ_{x_P} , δ_{y_P} (i.e., $A_P = A_Q = 0$, $\nu_P = \nu_Q = 0$, $\gamma_P = x$, $\gamma_Q = y$), then statement 2 is specialized to the kernel trick $\langle k_{A,\nu_s}(\cdot, x), k_{A,\nu_s}(\cdot, y) \rangle_{\mathcal{H}_{A,\nu_s}} = k_{A,\nu_s}(x, y)$. If P, Q and k are all Gaussians (i.e., $\nu_P = \nu_Q = \nu_s = 0$), then statement 2 results in the computation of the same Gaussian kernel with increased variance $A + A_P + A_Q$, where the computation of Gaussian pdfs is tractable.

Although Proposition 4.2 gives us a theory that kernel means m_P and RKHS inner products $\langle m_P, m_Q \rangle$ are expressed with generating triplets (A, ν, γ) , the computation of the general infinitely divisible pdfs may be intractable. We then systematically examine smaller subsemigroups of parametric models $(\mathcal{P}_{\Theta}, *) \subset (\mathbb{I}(\mathbb{R}^d), *)$ such that the computation of pdfs may be possible. We specifically examine well-known parametric classes of α -stable distributions and GH distributions on \mathbb{R}^d in Sections 4.2 and 4.3, respectively.

4.2 α -stable distributions

α -Stable distributions $S_\alpha(\mathbb{R}^d)$, $\alpha \in (0, 2]$, on \mathbb{R}^d are a well-known convolution subsemigroup of infinitely divisible distributions (Zolotarev, 1986; Samorodnitsky and Taqqu, 1994).

$\alpha = 2$ implies Gaussian distributions $S_2(\mathbb{R}^d) = \mathcal{G}(\mathbb{R}^d)$, which are closed under convolution; if P and Q are $N(\mu_P, R_P)$ and $N(\mu_Q, R_Q)$ with mean vectors μ_P, μ_Q and covariance matrices R_P, R_Q , respectively, then convolution $P * Q$ is $N(\mu_P + \mu_Q, R_P + R_Q)$.

For $\alpha \in (0, 2)$, α -stable distributions are heavy tailed, where there are many applications, as listed in Nolan (2013a). For each $\alpha \in (0, 2)$, a one-dimensional α -stable distribution $S_\alpha(\sigma, \beta, \mu)$ is specified by a scale parameter $\sigma > 0$, a skewness parameter $\beta \in [-1, 1]$, and a location parameter $\mu \in \mathbb{R}$. For each $\alpha \in (0, 2)$, the set $S_\alpha(\mathbb{R})$ is closed under convolution:

if P and Q are two stable laws $S_\alpha(\sigma_P, \beta_P, \mu_P)$ and $S_\alpha(\sigma_Q, \beta_Q, \mu_Q)$, respectively, then $P * Q$ is $S_\alpha(\sigma, \beta, \mu) = S_\alpha((\sigma_P^\alpha + \sigma_Q^\alpha)^{1/\alpha}, \frac{\beta_P \sigma_P^\alpha + \beta_Q \sigma_Q^\alpha}{\sigma_P^\alpha + \sigma_Q^\alpha}, \mu_P + \mu_Q)$ (Samorodnitsky and Taqqu, 1994, Property 1.2.1). See Appendix A.2 for more details.

For each $\alpha \in (0, 2)$, a d -dimensional α -stable distribution $S_\alpha(\mu, \Gamma)$ is specified by a location parameter $\mu \in \mathbb{R}^d$ and a spectral measure Γ on the unit sphere $S_{d-1} := \{s \in \mathbb{R}^d : \|s\| = 1\}$ (Samorodnitsky and Taqqu, 1994, Theorem 2.3.1, p.65). For each $\alpha \in (0, 2)$, the set $S_\alpha(\mathbb{R}^d)$ is closed under convolution; if P and Q are two stable laws $S_\alpha(\mu_P, \Gamma_P)$ and $S_\alpha(\mu_Q, \Gamma_Q)$, respectively, then $P * Q$ is $S_\alpha(\mu_P + \mu_Q, \Gamma_P + \Gamma_Q)$. See Appendix A.1 for more details. α -Stable pdfs on \mathbb{R}^d are intractable in general.

Sub-Gaussian α -stable distributions (equivalently, elliptically contoured α -stable distributions) $\text{SG}_\alpha(\mathbb{R}^d)$ are a well-known subclass of $S_\alpha(\mathbb{R}^d)$ (Samorodnitsky and Taqqu, 1994; Nolan, 2013b). For each $\alpha \in (0, 2)$, a sub-Gaussian α -stable distribution is specified by a location parameter $\mu \in \mathbb{R}^d$ and a p.d. matrix $R \in \mathbb{R}^{d \times d}$ (Samorodnitsky and Taqqu, 1994, Theorem 2.5.2, p.78). See Appendix A.4 for more details. Sub-Gaussian 1-stable distributions imply d -dimensional Cauchy distributions $\text{CAU}(\mathbb{R}^d)$ (Samorodnitsky and Taqqu, 1994, Example 2.5.3, p.79). If $d = 1$, for each $\alpha \in (0, 2)$, sub-Gaussians $\text{SG}_\alpha(\mathbb{R})$ are closed under convolution. If $d > 1$, for each $\alpha \in (0, 2)$, sub-Gaussians $\text{SG}_\alpha(\mathbb{R}^d)$ are not closed under convolution. Let us decompose $\text{SG}_\alpha(\mathbb{R}^d)$ into an equivalent class $\text{SG}_\alpha(\mathbb{R}^d) = \bigcup_R \text{SG}_\alpha(\mathbb{R}^d)[R]$, where

$$\text{SG}_\alpha(\mathbb{R}^d)[R] := \{P \in \text{SG}_\alpha(\mathbb{R}^d) \mid P = \text{SG}_\alpha(\mu, cR), \mu \in \mathbb{R}^d, c > 0\}.$$

For each $\alpha \in (0, 2)$ and a p.d. matrix $R \in \mathbb{P}^d$, the set $\text{SG}_\alpha(\mathbb{R}^d)[R]$ is closed under convolution; if P and Q are $\text{SG}_\alpha(\mu_P, c_P R)$ and $\text{SG}_\alpha(\mu_Q, c_Q R)$, respectively, then $P * Q$ is $\text{SG}_\alpha(\mu_P + \mu_Q, (c_P^2 + c_Q^2)^{\frac{\alpha}{2}} R)$. Note that when $\alpha = 2$, the whole set $\text{SG}_2(\mathbb{R}^d)$ is closed.

These convolution properties of α -stable distributions lead to the following conjugate pairs of α -stable kernels k and α -stable distributions \mathcal{P}_Θ .

Example 4.3 Conjugate pairs of α -stable kernels k and α -stable distributions on \mathbb{R}^d .

1. For $\alpha = 2$, let $k_R(x, y) = \frac{1}{\sqrt{(2\pi)^d |R|}} \exp(-\frac{1}{2}(x - y)^T R^{-1}(x - y))$ be a Gaussian kernel and \mathcal{H}_R be its RKHS. Let P, Q be two Gaussians $N(\mu_P, R_P)$ and $N(\mu_Q, R_Q)$, respectively. Then, the kernel mean is given by the Gaussian pdf $m_P = f_\alpha(\cdot; \mu_P, R + R_P)$ and the RKHS inner product is given by the Gaussian pdf $\langle m_P, m_Q \rangle_{\mathcal{H}_R} = f(\mu_P | \mu_Q, R + R_P + R_Q)$.
2. For each $\alpha \in (0, 2)$, let $k_{\alpha, \sigma}(x, y) = \kappa_{\alpha, \sigma}(x - y)$, $x, y \in \mathbb{R}$, be an α -stable kernel on \mathbb{R} and $\mathcal{H}_{\alpha, \sigma}$ be its RKHS. Let P, Q be two α -stable laws $S_\alpha(\sigma_P, \beta_P, \mu_P)$ and $S_\alpha(\sigma_Q, \beta_Q, \mu_Q)$, respectively, on \mathbb{R} . Then, the kernel mean is given by the stable pdf $m_P = f_\alpha(\cdot; (\sigma_P^\alpha + \sigma^\alpha)^{1/\alpha}, \frac{\beta_P \sigma_P^\alpha}{\sigma_P^\alpha + \sigma^\alpha}, \mu_P)$ and the RKHS inner product is given by the stable pdf $\langle m_P, m_Q \rangle_{\mathcal{H}_{\alpha, \sigma}} = f_\alpha(\mu_P | (\sigma_P^\alpha + \sigma^\alpha)^{1/\alpha}, \frac{\beta_Q \sigma_Q^\alpha - \beta_P \sigma_P^\alpha}{\sigma_Q^\alpha + \sigma_P^\alpha + \sigma^\alpha}, \mu_Q)$. If $\alpha = 1$ and $\beta = 0$, then $S_1(\sigma, 0, \mu)$ corresponds to the Cauchy distribution.
3. For each $\alpha \in (0, 2)$, let $k_{\alpha, \Gamma}(x, y) = \kappa_{\alpha, \Gamma}(x - y)$, $x, y \in \mathbb{R}^d$, be an α -stable kernel on \mathbb{R}^d , where Γ is a symmetric spectral measure, and let $\mathcal{H}_{\alpha, \Gamma}$ be its RKHS. Let P, Q be

two α -stable laws $S_\alpha(\mu_P, \Gamma_P)$ and $S_\alpha(\mu_Q, \Gamma_Q)$, respectively, on \mathbb{R}^d . Then, the kernel mean is given by the stable pdf $m_P = f_\alpha(\cdot | \mu_P, \Gamma_P + \Gamma_s)$ and the RKHS inner product is given by the stable pdf $\langle m_P, m_Q \rangle_{\mathcal{H}_{\alpha, \sigma}} = f_\alpha(\mu_P | \mu_Q, \Gamma_Q + \tilde{\Gamma}_P + \Gamma_s)$.

4. For each $\alpha \in (0, 2)$, let $k_{\alpha, R}(x, y) = k_{\alpha, R}(x - y)$, $x, y \in \mathbb{R}^d$ be a sub-Gaussian α -stable kernel on \mathbb{R}^d and let $\mathcal{H}_{\alpha, R}$ be its RKHS. Let $P, Q \in \text{SG}_\alpha(\mathbb{R}^d)[R]$ be two sub-Gaussian α -stable laws $S_\alpha(\mu_P, c_P R)$ and $S_\alpha(\mu_Q, c_Q R)$, respectively, on \mathbb{R}^d . Then, the kernel mean is given by the sub-Gaussian pdf $m_P = f_\alpha(\cdot | \mu_P, (c_P^{\frac{\alpha}{2}} + 1)^{\frac{\alpha}{2}} R)$ and the RKHS inner product is given by the sub-Gaussian pdf $\langle m_P, m_Q \rangle_{\mathcal{H}_{\alpha, R}} = f_\alpha(\mu_P | \mu_Q, (c_P^{\frac{\alpha}{2}} + c_Q^{\frac{\alpha}{2}} + 1)^{\frac{\alpha}{2}} R)$. If $\alpha = 1$, then $S_1(\mu, R)$ corresponds to multivariate Cauchy distributions with pdf $f(x) \propto (1 + \|x - \mu\|_{R^{-1}}^2)^{-\frac{d+1}{2}}$.

5. Tempered stable distributions can also be considered as examples (Rachev et al., 2011, Table 3.2, p. 77).

4.3 Generalized Hyperbolic Distributions

GH distributions on \mathbb{R}^d are a rich model class that includes, e.g., NIGs, hyperbolic distributions, VG distributions, Laplace distributions, Cauchy distributions, and Student's t distributions, as special cases and limiting cases (Barndorff-Nielsen and Halgreen, 1977; Prause, 1999; v. Hammerstein, 2010). A list of parametric models is found in, e.g., Prause (1999, Table 1.1 p.4). The GH and related models are applied, e.g., to mathematical finance (Schoutens, 2003; Cont and Tankov, 2004; Barndorff-Nielsen and Halgreen, 1990; Madan et al., 1998; Barndorff-Nielsen, 1998; Barndorff-Nielsen and Prause, 2001; Carr et al., 2002). The Matérn kernel, often used in machine learning, is a special case of the VG distributions. A GH distribution is obtained by a *normal mean-variance mixture* of a generalized inverse Gaussian (GIG) distribution, which is a special case of the normal mean-variance mixture of the generalized Γ -convolution (Thorin, 1978). The pdfs of GIG, GH, NIG, and VG distributions are presented in Appendix B.

We start by introducing a normal mean-variance mixture distribution. Let $N_d(\mu, \Delta)$ be a Gaussian distribution with mean vector $\mu \in \mathbb{R}^d$ and covariance matrix $\Delta \in \mathbb{P}^d$. A *normal mean-variance mixture* distribution P on \mathbb{R}^d is given by

$$P(dx) = \int_{\mathbb{R}^+} N_d(\mu + y\beta, y\Delta)(dx)G(dy), \quad \beta \in \mathbb{R}^d,$$

where G is a mixing probability measure on \mathbb{R}^+ (v. Hammerstein, 2010, Definition 2.4, p. 78). $P = N_d(\mu + y\beta, y\Delta) \circ G$ denotes a simple notation. The closure properties of the convolution and the infinite divisibility of G are preserved as follows:

Lemma 4.4 (v. Hammerstein, 2010, Lemma 2.5, p. 68) *Let \mathcal{G} be a class of probability distributions on $(\mathbb{R}^+, \mathcal{B}^+)$ and $G, G_1, G_2 \in \mathcal{G}$.*

1. If $G = G_1 * G_2 \in \mathcal{G}$, then

$$(N_d(\mu_1 + y\beta, y\Delta) \circ G_1) * (N_d(\mu_2 + y\beta, y\Delta) \circ G_2) = N_d(\mu_1 + \mu_2 + y\beta, y\Delta) \circ G.$$
2. If G is infinitely divisible, then so is $N_d(\mu + y\beta, y\Delta) \circ G$.

A GH distribution on \mathbb{R}^d is given by a normal mean-variance mixture with the GIG distribution:

$$GH_d(\lambda, \alpha, \beta, \delta, \mu, \Delta) := N_d(\mu + y\Delta\beta, y\Delta) \circ GIG(\lambda, \delta, \sqrt{\alpha^2 - \|\beta\|_\Delta^2}),$$

where the parameters imply $\lambda \in \mathbb{R}$, shape parameter $\alpha > 0$, skewness parameter β , scaling parameter δ , location parameter μ , and p.d. matrix $\Delta \in \mathbb{P}^d$ (see Appendices B.1 and B.2 for more details). A univariate GH distribution on \mathbb{R} is given by letting $d = 1$ and $\Delta = 1$.

The GH distribution contains the following subclasses and limiting cases. Their pdfs are found in Appendices B.3, B.4, and v. Hammerstein (2010):

1. If $\lambda = -\frac{1}{2}$, then $GH_d(-\frac{1}{2}, \alpha, \beta, \delta, \mu, \Delta)$ corresponds to the NIG distribution:

$$NIG_d(\alpha, \beta, \delta, \mu, \Delta) := N_d(\mu + y\Delta\beta, y\Delta) \circ GIG(-\frac{1}{2}, \delta, \sqrt{\alpha^2 - \|\beta\|_\Delta^2}).$$

2. If $\lambda = \frac{d+1}{2}$, then $GH_d(\frac{d+1}{2}, \alpha, \beta, \delta, \mu, \Delta)$ corresponds to the hyperbolic distribution $HYPd(\alpha, \beta, \delta, \mu, \Delta)$.

3. If $\lambda > 0$ and $\delta \rightarrow 0$, then $GH_d(\lambda > 0, \alpha, \beta, 0, \mu, \Delta)$ corresponds to the VG distribution

$$VG_d(\lambda, \alpha, \beta, \mu, \Delta) := N_d(\mu + y\Delta\beta, y\Delta) \circ \text{Gamma}(\lambda, \frac{\alpha^2 - \|\beta\|_\Delta^2}{2}),$$

where $\text{Gamma}(\lambda, \gamma)$ is the Gamma distribution with the pdf $f(x) = \frac{\gamma^\lambda}{\Gamma(\lambda)} x^{\lambda-1} e^{-\gamma x}$. Furthermore, if $\lambda = \frac{d+1}{2}$ (i.e., the above hyperbolic case), then $VG_d(\frac{d+1}{2}, \alpha, \beta, \mu, \Delta)$ corresponds to the skewed Laplace distribution

$$LAP_d(\alpha, \beta, \mu, \Delta) := N_d(\mu + y\Delta\beta, y\Delta) \circ \text{Gamma}(\frac{d+1}{2}, \frac{\alpha^2 - \|\beta\|_\Delta^2}{2}),$$

with the pdf $f(x) \propto e^{-\alpha\|x-\mu\|_{\Delta^{-1}} + (\beta, x-\mu)}$. We have seen the case of $d = 1$ in Example 3.8.

4. If $\lambda < 0$, $\alpha \rightarrow 0$, and $\beta \rightarrow 0$, then $GH_d(\lambda < 0, 0, \delta, \mu, \Delta)$ corresponds to the scaled and shifted t distribution with $f = -2\lambda$ degrees of freedom:

$$t_d(\lambda, \delta, \mu, \Delta) := N_d(\mu, y\Delta) \circ i\text{Gamma}(\lambda, \frac{\delta^2}{2}),$$

where $i\text{Gamma}(\lambda, \delta)$ is the inverse Gamma distribution with the pdf $f(x) = \frac{\delta^{\lambda-1}}{\delta\Gamma(-\lambda)} e^{-\frac{\delta}{x}}$. Furthermore, if $\lambda = -\frac{1}{2}$ (i.e., the above NIG case), then $t_d(-\frac{1}{2}, \delta, \mu, \Delta)$ corresponds to the multivariate Cauchy distribution

$$CAU(\delta, \mu, \Delta) := N_d(\mu, y\Delta) \circ i\text{Gamma}(-\frac{1}{2}, \frac{\delta^2}{2}),$$

with the pdf $f(x) \propto (1 + \frac{\|x-\mu\|_\Delta^2}{\delta^2})^{-\frac{d+1}{2}}$, which is also shown in Example 4.3.

These classes have the following convolution properties, by using Lemma 4.4 and Proposition B.1, which are the multivariate extensions of the univariate case (v. Hammerstein, 2010, eq. (1.9), p. 14).

Proposition 4.5 For each $d \geq 1$, there are the following convolution properties in the d -dimensional GH distributions:

1. $NIG_d(\alpha, \beta, \delta_1, \mu_1, \Delta) * NIG_d(\alpha, \beta, \delta_2, \mu_2, \Delta) = NIG_d(\alpha, \beta, \delta_1 + \delta_2, \mu_1 + \mu_2, \Delta)$,
2. $VG_d(\lambda_1, \alpha, \beta, \mu_1, \Delta) * VG_d(\lambda_2, \alpha, \beta, \mu_2, \Delta) = VG_d(\lambda_1 + \lambda_2, \alpha, \beta, \mu_1 + \mu_2, \Delta)$,
3. $NIG_d(\alpha, \beta, \delta_1, \mu_1, \Delta) * GH_d(1/2, \alpha, \beta, \delta_2, \mu_2, \Delta) = GH_d(1/2, \alpha, \beta, \delta_1 + \delta_2, \mu_1 + \mu_2, \Delta)$,
4. $GH_d(-\lambda, \alpha, \beta, \delta, \mu_1, \Delta) * GH_d(\lambda, \alpha, \beta, 0, \mu_2, \Delta) = GH_d(\lambda, \alpha, \beta, \delta, \mu_1 + \mu_2, \Delta)$,

where $\lambda, \lambda_1, \lambda_2 > 0$.

These convolution properties can also be obtained by looking up their characteristic functions and Lévy measures in v. Hammerstein (2010, Section 1.6.4, p. 46, Section 2.3, p. 79). Properties 1 and 2 imply a convolution semigroup. Property 3 implies an absorbing property. Property 4 implies another convolution property. By observing Proposition 4.5, we obtain the following conjugate, absorbing, and related pairs in GH kernels and GH distributions. The parametric models in Proposition 4.5 contain p.d. kernels κ if and only if $\beta = \mathbf{0}$. Each example (1–4) in the following corresponds to each property (1–4) in Proposition 4.5.

Example 4.6 Conjugate, absorbing, and related pairs in the GH class.

1. Let $k_{\alpha, \delta, \Delta}(x, y)$ be a shift invariant NIG p.d. kernel and $\mathcal{H}_{\alpha, \delta, \Delta}$ be the RKHS. Let P, Q be two NIG distributions $NIG(\alpha, \mathbf{0}, \delta_P, \mu_P, \Delta)$ and $NIG(\alpha, \mathbf{0}, \delta_Q, \mu_Q, \Delta)$, respectively. Then, the kernel mean is the NIG pdf $m_P = f(\cdot | \alpha, \mathbf{0}, \delta_P + \delta, \mu_P, \Delta)$ and the RKHS inner product is the NIG pdf $\langle m_P, m_Q \rangle_{\mathcal{H}_{\alpha, \delta, \Delta}} = f(\mu_P | \alpha, \mathbf{0}, \delta_P + \delta_Q + \delta, \mu_Q, \Delta)$. If $\alpha \rightarrow 0$, then these correspond to the Cauchy case.
2. Let $k_{\lambda, \alpha, \Delta}(x, y)$ be a shift invariant VG p.d. kernel⁸ and $\mathcal{H}_{\lambda, \alpha, \Delta}$ be the RKHS. Let P, Q be two VG distributions $VG(\lambda_P, \alpha, \mathbf{0}, \mu_P, \Delta)$ and $VG(\lambda_Q, \alpha, \mathbf{0}, \mu_Q, \Delta)$, respectively. Then, the kernel mean is the VG pdf $m_P = f(\cdot | \lambda_P + \lambda, \alpha, \mathbf{0}, \mu_P, \Delta)$ and the RKHS inner product is the VG pdf $\langle m_P, m_Q \rangle_{\mathcal{H}_{\lambda, \alpha, \Delta}} = f(\mu_P | \lambda_P + \lambda_Q + \lambda, \alpha, \mathbf{0}, \mu_Q, \Delta)$. If $\lambda = \frac{d+1}{2}$, $\lambda_P = \frac{d+1}{2}$, or $\lambda_Q = \frac{d+1}{2}$, then these correspond to the Laplace case.
3. Let $k_{\alpha, \delta, \Delta}(x, y)$ be a NIG kernel and $\mathcal{H}_{\alpha, \delta, \Delta}$ be the RKHS. Let P be a GH distribution $GH(1/2, \alpha, \mathbf{0}, \delta_P, \mu_P, \Delta)$. Then, the kernel mean is the GH pdf $m_P = f(\cdot | 1/2, \alpha, \mathbf{0}, \delta_P + \delta, \mu_P, \Delta)$. If $\alpha \rightarrow 0$, then the NIG kernel $k_{\alpha, \delta, \Delta}(x, y)$ corresponds to the Cauchy kernel.

Let $k_{1, 2, \alpha, \delta, \Delta}(x, y)$ be a GH kernel and $\mathcal{H}_{1, 2, \alpha, \delta, \Delta}$ be the RKHS. Let P, Q be two NIG distributions $NIG(\alpha, 0, \delta_P, \mu_P, \Delta)$ and $NIG(\alpha, \mathbf{0}, \delta_Q, \mu_Q, \Delta)$, respectively. Then, the

9. The Matérn kernel corresponds to $\Delta = I$, and $\alpha = \frac{\sqrt{2\nu}}{\sigma}$ (Rasmussen and Williams, 2006, Section 4.2.1) (Sriperumbudur et al., 2010, p. 1533)

kernel mean is the GH pdf $m_P = f(\cdot | 1/2, \alpha, \mathbf{0}, \delta_P + \delta, \mu_P, \Delta)$ and the RKHS inner product is the GH pdf $\langle m_P, m_Q \rangle_{\mathcal{H}_{1, 2, \alpha, \delta, \Delta}} = f(\mu_P | 1/2, \alpha, \mathbf{0}, \delta_P + \delta_Q + \delta, \mu_Q, \Delta)$. If $\alpha \rightarrow 0$, then the NIG distributions, P and Q , correspond to the Cauchy distributions.

4. For $\lambda > 0$, let $k_{-\lambda, \alpha, \delta, \Delta}(x, y)$ be a GH kernel and $\mathcal{H}_{-\lambda, \alpha, \delta, \Delta}$ be the RKHS. Let P be a GH distribution $GH(\lambda, \alpha, \mathbf{0}, 0, \mu_P, \Delta)$. Then, the kernel mean is the GH pdf $m_P = f(\cdot | \lambda, \alpha, \delta, \mu_P, \Delta)$. If $\alpha \rightarrow 0$, then $k_{-\lambda, \alpha, \delta, \Delta}(x, y)$ corresponds to the Student's t kernel. Furthermore, if $\lambda = \frac{1}{2}$, then $k_{-\frac{1}{2}, \alpha, \delta, \Delta}(x, y)$ corresponds to the Cauchy kernel. For $\lambda > 0$, let $k_{\lambda, \alpha, \Delta}(x, y)$ be a GH kernel and $\mathcal{H}_{\lambda, \alpha, \Delta}$ be the RKHS. Let P be a GH distribution $GH(-\lambda, \alpha, \mathbf{0}, \delta_P, \mu_P, \Delta)$. Then, the kernel mean is the GH pdf $m_P = f(\cdot | \lambda, \alpha, \mathbf{0}, \delta_P, \mu_P, \Delta)$. If $\alpha \rightarrow 0$, then P is the Student's t distribution. Furthermore, if $\lambda = -\frac{1}{2}$, then P is the Cauchy distribution.

5. Connection to Machine Learning

As mentioned in the Introduction, absorbing and conjugate kernels (Examples 4.3 and 4.6) provide a way to compute the RKHS values (i) $f(x)$, $x \in \mathbb{R}^d$, and the RKHS inner products (ii) $\langle f, g \rangle_{\mathcal{H}}$ when $f, g \in \mathcal{H}$ are expressed by the weighted sums of parametric kernel means, $f = \sum_{i=1}^n w_i m_{P_i}$ and $g = \sum_{j=1}^n \tilde{w}_j m_{Q_j}$ for $\{P_i\}, \{Q_j\} \subset \mathcal{P}_{\Theta}$. Many algorithms aim to use the convolution trick. Examples include as follows:

- The difference between a probability measure $P \in \mathcal{M}_1(\mathbb{R}^d)$ and a model $P_{\theta} \in \mathcal{P}_{\Theta}$ in the RKHS norm $\|m_P - m_{P_{\theta}}\|_{\mathcal{H}}$ needs to be computed, e.g., for the purpose of a goodness-of-fit test and model criticism (Lloyd and Ghaltrayami, 2015), based on the maximum mean discrepancy (MMD) (Gretton et al., 2012).
- Various kernels $k(P, P_{\theta})$ between a probabilistic measure P and a model P_{θ} , e.g., $k(P, P_{\theta}) = \exp(-\frac{\|m_P - m_{P_{\theta}}\|_{\mathcal{H}}^2}{2\sigma^2})$ need to be computed, as in the support measure machine (Mhandet et al., 2012).
- Song et al. (2008) and McCauman et al. (2013) studied an approximation of a target probability measure $P \in \mathcal{M}_1(\mathbb{R}^d)$ with a Gaussian mixture model $P_{\theta} = \sum_{i=1}^n \theta_i P_i$ via solving the following optimization problem:

$$\hat{\theta} = \operatorname{argmin}_{\theta} \|m_P - m_{P_{\theta}}\|_{\mathcal{H}}^2 + \Omega(\theta) = \operatorname{argmin}_{\theta} \|m_P - \sum_{i=1}^n \theta_i m_{P_i}\|_{\mathcal{H}}^2 + \Omega(\theta),$$

where $\Omega(\theta)$ is a regularization term, $\frac{\lambda}{2} \|\theta\|^2$ ($\lambda > 0$). This optimization is solved by a constrained quadratic program: $\min_{\theta} \frac{1}{2} \theta^T (A + \lambda I_n) \theta - b^T \theta$ subject to $\sum_{i=1}^n \theta_i = 1$ and $\theta \geq 0$, where we then need the computation of matrix $A \in \mathbb{R}^{n \times n}$ and vector $b \in \mathbb{R}^n$:

$$A_{ij} = \langle m_{P_i}, m_{P_j} \rangle_{\mathcal{H}}, \quad b_j = \langle \hat{m}_P, m_{P_j} \rangle_{\mathcal{H}}, \quad 1 \leq i, j \leq n,$$

for parametric kernel means $\{m_{P_i}\}$.

- As mentioned in the Introduction, the kernel Bayesian inference (KBI), which employs Bayesian inference in kernel mean form, has been proposed (Fukumizu et al. 2013,

Song et al. 2013). KBI is applied to, e.g., filtering and smoothing algorithms on state space models (Fukumizu et al. 2013 Kanagawa et al. 2016, Nishiyama et al. 2016) and policy learning in reinforcement learning (Grünewälder et al. 2012, Nishiyama et al. 2012, Rawlik et al. 2013, Boots et al. 2013). When we extend it to *semiparametric* KBI, which combines nonparametric inference and parametric inference, we may want to use the RKHS functions $f = \sum_{i=1}^n w_i m_{P_{\theta_i}} \in \mathcal{H}$ expressed by parametric kernel means $\{P_{\theta_i}\} \in \mathcal{P}_{\Theta}$, as is used in the model-based kernel sum rule (Mb-KSR) (Nishiyama et al., 2014).

- Preimage algorithms (Mika et al., 1999; Fukumizu et al., 2013) and kernel herding algorithms (Chen et al., 2010) can also be extended to estimators $f = \sum_{i=1}^n w_i m_{P_{\theta_i}}$ with parametric kernel means $\{P_{\theta_i}\}$.

6. Computation of Conjugate Kernels (Convolution Trick)

In Section 4, we mathematically investigated that several convolution tricks hold within a general convolution trick (Proposition 4.2): the computation of kernel mean values and RKHS inner products is the same as the computation of p.d. kernels having different parameters, if conjugate kernels are used. However, conjugate kernels do not provide a tractable computation in general. We then discuss the computation of the conjugate kernels: α -stable kernels and GH kernels.

- It is known that α -stable pdfs do not generally have a closed-form expression except for some special cases, Gaussians ($\alpha = 2$) and Cauchy ($\alpha = 1$), as given in Appendix A.3. Gaussian and Cauchy kernels may be used as tractable conjugate kernels. For other α -stable kernels ($\alpha \neq 2$ and $\alpha \neq 1$), some numerical elaborations or approximations may be needed for the computation of the pdfs. The STABLE 5.1¹⁰ software allows the computation of α -stable pdfs when they are independent, isotropic, elliptical, or have discrete spectral measures Γ_d under some settings. More information can be found in the STABLE 5.1 software manual. For elliptically contoured α -stable sub-Gaussian kernels on any dimension \mathbb{R}^d , the computation of pdfs is sufficient only to compute a one-dimensional amplitude function $\tilde{\kappa}(r)$ in equation (2), which can be computed by, e.g., a one-dimensional numerical integration. The STABLE 5.1 software supports the computation of sub-Gaussian pdfs in dimension $d < 100$.

- GH kernels and their subclasses are also elliptical pdfs, and the computation of the kernels is sufficient only to compute a one-dimensional amplitude function $\tilde{\kappa}(r)$. VG kernels or Matérn kernels, which are a generalization of Laplace kernels, are used for covariance kernels in Gaussian processes. GH and NIG kernels are variants of Matérn kernels, all of which are expressed by the Bessel function of the third kind. For example, there is an R package software called ‘ghyp’ on the GH distributions (Breyer and Lüthi, 2013).

In addition, random Fourier features (Rahimi and Recht, 2007) may be an approach to approximately compute conjugate kernels. From Proposition 4.2, we have an equality

$$\langle m_P, m_Q \rangle_{\mathcal{H}_{A, \nu_s}} = k_{A+A_P+A_Q, \nu_s + \nu_P + \nu_Q}(\gamma_P, \gamma_Q) = \mathbb{E}_{\omega}[\zeta_{\omega}(\gamma_P)\zeta_{\omega}(\gamma_Q)^*].$$

An RKHS inner product (l.h.s.) may be computed by approximating the expectation of $\zeta_{\omega}(\gamma_P)\zeta_{\omega}(\gamma_Q)^*$ (r.h.s.) with sampling ω from the characteristic function having the generating triplet $(A + A_P + A_Q, \nu_s + \nu_P + \nu_Q)$.

7. Conclusion

In this paper, we introduced a class of CID kernels that constitutes a large subclass in the set of shift-invariant characteristic kernels on \mathbb{R}^d , where CID kernels are closed under convolution but not closed under addition and pointwise product. We introduced absorbing, conjugate kernels, and convolution trick with respect to parametric models, where the basic computation of kernel mean values and RKHS inner products results in the computation of the same p.d. kernels with different parameters, which is an extension of kernel trick. Although the convolution trick may offer a mathematical view, the computation of conjugate kernels is not tractable in general. We then restrict convolution trick only to tractable cases or approximately compute intractable conjugate kernels. Future works include investigating the effectiveness of convolution trick in practice and developing approximation algorithms to efficiently compute intractable conjugate kernels.

Acknowledgments

We thank anonymous reviewers and the action editor for helpful comments. Y.N. thanks Prof. Tatsuhiro Saigo and Prof. Takaaki Shimura for a helpful discussion on infinitely divisible distributions. This work was supported in part by JSPS KAKENHI (grant nos. 26870821 and 22300098), the MEXT Grant-in-Aid for Scientific Research on Innovative Areas (no. 25120012), and by the Program to Disseminate Tenure Tracking System, MEXT, Japan.

Appendix A. α -Stable Distributions

We briefly review the α -stable distributions on \mathbb{R}^d .

A.1 α -Stable Distributions on \mathbb{R}^d

The α -stable distribution on \mathbb{R}^d has the following characteristic function:

Theorem A.1 (Samorodnitsky and Taqqu, 1994, Theorem 2.3.1, p. 65) *Let $\alpha \in (0, 2)$. Then, $X = (X_1, \dots, X_d)$ is an α -stable random vector in \mathbb{R}^d if and only if there exists a finite measure Γ on the unit sphere $S_{d-1} = \{s \in \mathbb{R}^d : \|s\| = 1\}$ and a vector $\mu^0 \in \mathbb{R}^d$ such that*

$$\hat{P}(\theta) = \begin{cases} \exp\left(-\int_{S_{d-1}} |\theta^\top s|^\alpha (1 - i \operatorname{sgn}(\theta^\top s) \tan \frac{\pi\alpha}{2}) \Gamma(ds) + i\theta^\top \mu^0\right), & (\alpha \neq 1), \\ \exp\left(-\int_{S_{d-1}} |\theta^\top s|^\alpha (1 + i \frac{2}{\pi} \operatorname{sgn}(\theta^\top s) \ln |\theta^\top s|) \Gamma(ds) + i\theta^\top \mu^0\right), & (\alpha = 1). \end{cases}$$

10. John Nolan’s Page. <http://academic2.american.edu/~jpnolan/stable/stable.html>

The pair (Γ, μ^0) is unique.

The measure Γ is called the *spectral measure*. See Samorodnitsky and Taqqu (1994, Section 2.3) for some examples of spectral measures. The radial sub-Gaussian distribution has a uniform spectral measure. An α -stable random vector $X = (X_1, \dots, X_d)$ has independent components if and only if its spectral measure Γ is discrete and concentrated on the intersection of the axes with the sphere S_{d-1} . It is known that any nondegenerate stable distribution on \mathbb{R}^d has the C^∞ pdf (Sato, 1999, Example 28.2, p. 190). An α -stable distribution on \mathbb{R}^d is symmetric if and only if $\mu^0 = 0$ and Γ is a symmetric measure on S_{d-1} (i.e., it satisfies $\Gamma(A) = \Gamma(-A)$ for any $A \in \mathcal{B}(S_{d-1})$) (Samorodnitsky and Taqqu, 1994, p.73). For each $\alpha \in (0, 2)$, α -stable distributions on \mathbb{R}^d have the generating triplet $(0, \nu, \gamma)$ with

$$\nu(B) = \int_{S_{d-1}} \Gamma(ds) \int_0^\infty 1_B(rs) \frac{dr}{r^{1+\alpha}}, \quad B \in \mathcal{B}(\mathbb{R}^d), \quad (5)$$

where Γ is the spectral measure on S_{d-1} (Sato, 1999, Theorem 14.3, p. 77). The sum of Lévy measures $\nu_1 + \nu_2$ implies the sum of spectral measures $\Gamma_1 + \Gamma_2$.

A.2 α -Stable Distributions on \mathbb{R}

As a special case, an α -stable distribution on \mathbb{R} has the following characteristic function:

Theorem A.2 (Samorodnitsky and Taqqu, 1994, Definition 1.1.6, p. 5) *A random variable X is α -stable ($\alpha \in (0, 2]$) in \mathbb{R} if and only if the parameters satisfy the conditions $\sigma \geq 0$, $\beta \in [-1, 1]$, and $\mu \in \mathbb{R}$ such that its characteristic function has the form*

$$\hat{F}(\theta) = \begin{cases} \exp(-\sigma^\alpha |\theta|^\alpha (1 - i\beta(\operatorname{sgn}\theta) \tan(\frac{\pi}{2}) + i\mu\theta)) & (\alpha \neq 1), \\ \exp(-\sigma |\theta| (1 + i\beta \frac{\pi}{2} (\operatorname{sgn}\theta) \ln|\theta| + i\mu\theta)) & (\alpha = 1), \end{cases}$$

where $\operatorname{sgn}\theta$ is a sign function

$$\operatorname{sgn}\theta = \begin{cases} 1 & \theta > 0, \\ 0 & \theta = 0, \\ -1 & \theta < 0. \end{cases}$$

When $\alpha \in (0, 2)$, the parameters σ , β , and μ are unique. When $\alpha = 2$, β is irrelevant, and σ and μ are unique.

An α -stable distribution on \mathbb{R} is specified by the parameters (α, β, μ) , where σ is a scale parameter, β is a skewness parameter, and μ is a location parameter. $\sigma = 0$ implies a delta measure. For $\alpha \in (0, 2)$, an α -stable distribution is symmetric if and only if $\beta = \mu = 0$ (Samorodnitsky and Taqqu, 1994, Property 1.2.5, p. 11). A 2-stable distribution is symmetric if and only if $\mu = 0$. An α -stable density does not generally have a closed-form expression, except for some special cases. However, it is known that every nondegenerate stable distribution has the C^∞ pdf (Sato, 1999, Example 28.2, p. 190). Some known univariate α -stable pdfs, expressed by elementary functions and special functions, are given in Appendix A.3.

The Lévy measure ν of a univariate stable distribution is obtained by letting $d = 1$ in the Lévy measure (5). If $d = 1$, then $S_0 = \{-1, 1\}$ and $\Gamma = \Gamma(\{-1\})\delta_{-1} + \Gamma(\{1\})\delta_1$, where

$\Gamma(\{-1\}), \Gamma(\{1\}) \geq 0$ and $\Gamma(\{-1\}) + \Gamma(\{1\}) > 0$ (Samorodnitsky and Taqqu, 1994, Example 2.3.3, p. 67). By substituting this into equation (5), we can obtain the Lévy measure ν of a univariate stable distribution as

$$\nu(dx) = \Gamma(\{1\}) \frac{1}{x^{1+\alpha}} 1_{(0,\infty)}(x) dx + \Gamma(\{-1\}) \frac{1}{|x|^{1+\alpha}} 1_{(-\infty,0)}(x) dx.$$

A stable distribution $S_\alpha(\alpha, \beta, \mu)$ is given with the spectral measure as

$$\sigma = (\Gamma(\{1\}) + \Gamma(\{-1\}))^{\frac{1}{\alpha}} > 0, \quad \beta = \frac{(\Gamma(\{1\}) - \Gamma(\{-1\}))}{\Gamma(\{1\}) + \Gamma(\{-1\})} \in [-1, 1].$$

The sum of Lévy measures $\nu_1 + \nu_2$ implies the sum of mass functions $\Gamma_1(\{-1\}) + \Gamma_2(\{-1\})$ and $\Gamma_1(\{1\}) + \Gamma_2(\{1\})$. We can see the convolution property $S_\alpha(\alpha_1, \beta_1, \mu_1) * S_\alpha(\alpha_2, \beta_2, \mu_2) = S_\alpha((\sigma_1^\alpha + \sigma_2^\alpha)^{\frac{1}{\alpha}}, \frac{\sigma_1^\alpha \beta_1 + \sigma_2^\alpha \beta_2}{\sigma_1^\alpha + \sigma_2^\alpha}, \mu_1 + \mu_2)$ of the univariate stable distribution from the viewpoint of the spectral measure.

A.3 Closed-Form and Special Function Form of α -Stable PDFs on \mathbb{R}

There are three cases where the α -stable pdf on \mathbb{R} is expressed by elementary functions:

1. The 2-stable distribution $S_2(\alpha, \beta, \mu)$ is the Gaussian $N(\mu, 2\sigma^2)$, where β has no effect, with the pdf

$$f_{\text{Gauss}}(x) = \frac{1}{2\sigma\sqrt{\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad x \in \mathbb{R}.$$

2. The 1-stable distribution $S_1(\alpha, \beta = 0, \mu)$ is the Cauchy distribution with the pdf

$$f_{\text{Cauchy}}(x) = \frac{\sigma}{\pi((x-\mu)^2 + \sigma^2)}, \quad x \in \mathbb{R}.$$

3. The 1/2-stable distribution $S_{1/2}(\alpha, \beta = \pm 1, \mu)$ is the Lévy distribution with the pdf

$$f_{\text{Lévy}}(x) = \frac{\sqrt{\sigma}}{\sqrt{2\pi}(x-\mu)^{3/2}} e^{-\frac{\sigma(x-\mu)}{2(x-\mu)^2}}, \quad \mu < x < \infty.$$

There are some cases where the α -stable pdf is expressed by special functions. The following expression is found in Lee (2010). Note that kernel means m_P and RKHS inner products also take these expressions. For simplicity, we only show standardized stable pdfs $d_{\text{stable}}(x; \alpha, \sigma = 1, \beta, \mu = 0)$.

Fresnel integrals:

If $(\alpha, \sigma, \beta, \mu) = (1/2, 1, 0, 0)$,

$$\begin{aligned} d_{\text{stable}}(x; 1/2, 1, 0, 0) \\ = \frac{|x|^{-\frac{3}{2}}}{\sqrt{2\pi}} \left(\sin\left(\frac{1}{4|x|}\right) \left(\frac{1}{2} - S\left(\sqrt{\frac{1}{2\pi|x|}}\right) \right) + \cos\left(\frac{1}{4|x|}\right) \left(\frac{1}{2} - C\left(\sqrt{\frac{1}{2\pi|x|}}\right) \right) \right), \end{aligned}$$

where $C(z)$ and $S(z)$ are the Fresnel integrals

$$C(z) = \int_0^z \cos\left(\frac{\pi t^2}{2}\right) dt, \quad S(z) = \int_0^z \sin\left(\frac{\pi t^2}{2}\right) dt.$$

This is a symmetric stable pdf. $k(x, y) = d_{stable}(x - y; 1/2, 1, 0, 0)$, $x, y \in \mathbb{R}$, gives a characteristic p.d. kernel.

Modified Bessel function:

If $(\alpha, \sigma, \beta, \mu) = (1/3, 1, 1, 0)$, the one-sided continuous density is

$$d_{stable}(x; 1/3, 1, 1, 0) = \frac{1}{\pi} \frac{2^{3/2}}{3^{7/4}} x^{-3/2} K_{1/3}\left(\frac{2^{5/2}}{3^{9/4}} x^{-1/2}\right), \quad x \geq 0,$$

where $K_\nu(x)$ is a modified Bessel function of the third kind.

Hypergeometric function:

If $(\alpha, \sigma, \beta, \mu) = (4/3, 1, 0, 0)$,

$$d_{stable}(x; \frac{4}{3}, 1, 0, 0) = \frac{3^{5/4} \Gamma(7/12) \Gamma(11/12)}{2^{5/2} \sqrt{\pi} \Gamma(6/12) \Gamma(8/12)} {}_2F_2\left(\frac{7}{12}, \frac{11}{12}; \frac{1}{2}, \frac{1}{2}; -\frac{3^3 x^4}{2^8}\right) - \frac{3^{11/4} |x|^3 \Gamma(13/12) \Gamma(17/12)}{2^{13/2} \sqrt{\pi} \Gamma(18/12) \Gamma(15/12)} {}_2F_2\left(\frac{13}{12}, \frac{17}{12}; \frac{1}{2}, \frac{1}{2}; -\frac{3^3 x^4}{2^8}\right), \quad x \in \mathbb{R},$$

where ${}_pF_q$ is the (generalized) hypergeometric function

$${}_pF_q(a_1, \dots, a_p; b_1, \dots, b_q; z) = \sum_{n=0}^{\infty} \frac{(a_1)_n \cdots (a_p)_n z^n}{(b_1)_n \cdots (b_q)_n n!}$$

with the Pochhammer symbol $(a)_0 = 1$, $(a)_n = a(a+1)\cdots(a+n-1)$ for $n \in \mathbb{N}^+$. This is a symmetric stable pdf. $k(x, y) = d_{stable}(x - y; \frac{4}{3}, 1, 0, 0)$, $x, y \in \mathbb{R}$, gives a characteristic p.d. kernel.

If $(\alpha, \sigma, \beta, \mu) = (3/2, 1, 0, 0)$ (the Holtsmark distribution),

$$d_{stable}(x; \frac{3}{2}, 1, 0, 0) = \frac{1}{\pi} \Gamma(5/3) {}_2F_3\left(\frac{5}{12}, \frac{11}{12}; \frac{1}{3}, \frac{2}{6}; -\frac{2^2 x^6}{3^6}\right) - \frac{x^2}{3\pi} {}_3F_4\left(\frac{3}{4}, 1, \frac{5}{4}; \frac{3}{6}, \frac{5}{6}, \frac{7}{6}; -\frac{2^2 x^6}{3^6}\right) + \frac{7x^4}{3^4 \pi} \Gamma(4/3) {}_2F_3\left(\frac{13}{12}, \frac{19}{12}; \frac{7}{6}, \frac{5}{2}; -\frac{2^2 x^6}{3^6}\right), \quad x \in \mathbb{R}.$$

This is a symmetric stable pdf. The Holtsmark kernel $k(x, y) = d_{stable}(x - y; 3/2, 1, 0, 0)$, $x, y \in \mathbb{R}$, gives a characteristic p.d. kernel.

Whittaker function:

If $(\alpha, \sigma, \beta, \mu) = (2/3, 1, 0, 0)$,

$$d_{stable}(x; 2/3, 1, 0, 0) = \frac{1}{2\sqrt{3\pi} |x|} \exp\left(\frac{2}{27x^2}\right) W_{-1/2, 1/6}\left(\frac{4}{27x^2}\right), \quad x \in \mathbb{R},$$

where $W_{\lambda, \mu}(z)$ is the Whittaker function defined as

$$W_{\lambda, \mu}(z) = \frac{z^\lambda e^{-z/2}}{\Gamma(\mu - \lambda + 1/2)} \int_0^\infty e^{-t} t^{\mu - \lambda - 1/2} \left(1 + \frac{t}{z}\right)^{\mu - \lambda - 1/2} dt,$$

$$\operatorname{Re}(\mu - \lambda) > -\frac{1}{2}, |\arg(z)| < \pi.$$

This is a symmetric stable pdf. $k(x, y) = d_{stable}(x - y; 2/3, 1, 0, 0)$, $x, y \in \mathbb{R}$, gives a characteristic p.d. kernel.

If $(\alpha, \sigma, \beta, \mu) = (2/3, 1, 1, 0)$, the one-sided density is

$$d_{stable}(x; 2/3, 1, 1, 0) = \sqrt{\frac{3}{\pi}} \frac{1}{|x|} \exp\left(-\frac{16}{27|x|^2}\right) W_{1/2, 1/6}\left(\frac{32}{27|x|^2}\right), \quad x \geq 0.$$

If $(\alpha, \sigma, \beta, \mu) = (3/2, 1, 1, 0)$, the α -stable density is

$$d_{stable}(x; 2/3, 1, 1, 0) = \begin{cases} \sqrt{\frac{3}{\pi}} \frac{1}{|x|} \exp\left(\frac{x^3}{27}\right) W_{1/2, 1/6}\left(-\frac{2}{27}x^3\right), & x < 0 \\ \frac{1}{2\sqrt{3\pi}|x|} \exp\left(\frac{x^3}{27}\right) W_{-1/2, 1/6}\left(\frac{2}{27}x^3\right), & x > 0 \end{cases}$$

Lommel function:

If $(\alpha, \sigma, \beta, \mu) = (1/3, 1, 0, 0)$,

$$d_{stable}(x; 1/3, 1, 0, 0) = \operatorname{Re}\left(\frac{2 \exp(-i\pi/4)}{3\sqrt{3\pi}|x|^{3/2}} S_{0,1/3}\left(\frac{2 \exp(i\pi/4)}{3\sqrt{3}|x|^{1/2}}\right)\right).$$

Here, the Lommel functions $s_{\mu, \nu}(z)$ and $S_{\mu, \nu}(z)$ are defined by

$$s_{\mu, \nu}(z) = \frac{\pi}{2} \left(Y_\nu(z) \int_0^z z^\mu J_\nu(z) dz - J_\nu(z) \int_0^z z^\mu Y_\nu(z) dz \right),$$

$$S_{\mu, \nu}(z) = s_{\mu, \nu}(z) - \frac{2^{\mu-1} \Gamma((1+\mu+\nu)/2)}{\pi \Gamma((\nu-\mu)/2)} \left(J_\nu(z) - \cos\left(\frac{\mu-\nu}{2}\pi\right) Y_\nu(z) \right),$$

where $J_\nu(z)$ and $Y_\nu(z)$ are Bessel functions of the first and second kind, respectively. This is a symmetric stable pdf. $k(x, y) = d_{stable}(x - y; 1/3, 1, 0, 0)$, $x, y \in \mathbb{R}$, gives a characteristic p.d. kernel.

Landau distribution:

If $(\alpha, \sigma, \beta, \mu) = (1, 1, 1, 0)$ (the Landau distribution),

$$d_{stable}(x; 1, 1, 1, 0) = \frac{1}{\pi} \int_0^\infty e^{-t \log t - xt} \sin(\pi t) dt.$$

A.4 Sub-Gaussian (Elliptically Contoured) α -Stable Distributions on \mathbb{R}^d

The sub-Gaussian α -stable distribution has the following characteristic function:

Proposition A.3 (Samorodnitsky and Taqqu, 1994, Proposition 2.5.2, p. 78) *Let $\alpha \in (0, 2)$. The sub-Gaussian α -stable random vector X in \mathbb{R}^d has the characteristic function*

$$E \exp \left[i \sum_{k=1}^d \theta_k X_k \right] = \exp \left(- \left| \frac{1}{2} \sum_{i,j=1}^d \theta_i \theta_j R_{ij} \right|^{\frac{\alpha}{2}} + i(\theta, \mu^0) \right),$$

where R is a p.d. matrix and $\mu^0 \in \mathbb{R}^d$ is a shift vector.

$\alpha = 2$ and $\alpha = 1$ imply the multivariate Gaussian and Cauchy distribution, respectively.

For $\alpha \in (0, 2)$, the radial sub-Gaussian $\text{SG}_\alpha(\mathbb{R}^d)[I]$ (with identity matrix $R = I$) has the uniform spectral measure $\Gamma(B) = c|B|$, $\forall B \in \mathcal{B}(S_{d-1})$ in the Lévy measure (5) (Samorodnitsky and Taqqu, 1994, Proposition 2.5.5, p. 79), Sub-Gaussian $\text{SG}_\alpha(\mathbb{R}^d)[R]$ with a p.d. matrix R is the elliptical version of the radial sub-Gaussians. Its spectral measure is given in Samorodnitsky and Taqqu (1994, Proposition 2.5.8, p. 82).

Appendix B. GH Classes on \mathbb{R}^d

A GH distribution on \mathbb{R}^d is given by the normal mean-variance mixture with the GIG mixing distribution. See, e.g., v. Hammerstein (2010) for more information. We here reproduce some of them.

B.1 GIG Distributions on \mathbb{R}^+

A generalized inverse Gaussian (GIG) distribution $GIG(\lambda, \delta, \gamma)$ on \mathbb{R}^+ is given by the following pdf:

$$d_{GIG(\lambda, \delta, \gamma)}(x) = \left(\frac{\gamma}{\delta}\right)^\lambda \frac{1}{2K_\lambda(\delta\gamma)} x^{\lambda-1} \exp\left(-\frac{1}{2}\left(\frac{\delta^2}{x} + \gamma^2 x\right)\right) 1_{(0, \infty)}(x),$$

where $K_\lambda(x)$ is the modified Bessel function of the third kind with index λ . The parameters $(\lambda, \delta, \gamma)$ take the following values:

$$\begin{cases} \delta \geq 0, \gamma > 0, & \text{if } \lambda > 0, \\ \delta > 0, \gamma > 0, & \text{if } \lambda = 0, \\ \delta > 0, \gamma \geq 0, & \text{if } \lambda < 0, \end{cases}$$

where $\delta = 0$ and $\gamma = 0$ correspond to limiting cases,¹¹ which are the Gamma distribution and the inverse Gamma distribution, respectively. The GIG distributions have the following convolution properties:

Proposition B.1 (v. Hammerstein, 2010, Proposition 1.11, p. 11) *Within the class of GIG distributions, the following convolution properties hold:*

- $GIG(-\frac{1}{2}, \delta_1, \gamma) * GIG(-\frac{1}{2}, \delta_2, \gamma) = GIG(-\frac{1}{2}, \delta_1 + \delta_2, \gamma)$,
 - $GIG(-\frac{1}{2}, \delta_1, \gamma) * GIG(\frac{1}{2}, \delta_2, \gamma) = GIG(\frac{1}{2}, \delta_1 + \delta_2, \gamma)$,
 - $GIG(-\lambda, \delta, \gamma) * GIG(\lambda, 0, \gamma) = GIG(\lambda, \delta, \gamma)$, $\lambda > 0$,
 - $GIG(\lambda_1, 0, \gamma) * GIG(\lambda_2, 0, \gamma) = GIG(\lambda_1 + \lambda_2, 0, \gamma)$, $\lambda_1, \lambda_2 > 0$.
11. If $\lambda \neq 0$, then $K_\lambda(x) \sim \frac{1}{2}\Gamma(\lambda)(\frac{x}{2})^{-\lambda}$ ($x \downarrow 0$).

B.2 GH Distributions on \mathbb{R}^d

A GH distribution has the following pdf:

$$d_{GH_d(\lambda, \alpha, \beta, \delta, \mu, \Delta)}(x) = a(\lambda, \alpha, \beta, \delta, \mu, \Delta) \left(\sqrt{\delta^2 + \|x - \mu\|_{\Delta}^2} \right)^{\lambda - \frac{d}{2}} K_{\lambda - \frac{d}{2}} \left(\alpha \sqrt{\delta^2 + \|x - \mu\|_{\Delta}^2} \right) e^{(\beta, x - \mu)},$$

where $a(\lambda, \alpha, \beta, \delta, \mu, \Delta)$ is the normalization constant:

$$a(\lambda, \alpha, \beta, \delta, \mu, \Delta) = \frac{(\alpha^2 - \|\beta\|_{\Delta}^2)^{\lambda \nu/2}}{(2\pi)^{d/2} |\Delta|^{\frac{1}{2}} \alpha^{\lambda - d/2} \delta^{\lambda} K_{\lambda}(\delta \sqrt{\alpha^2 - \|\beta\|_{\Delta}^2})}.$$

The GH parameters $(\lambda, \alpha, \beta, \delta, \mu, \Delta)$ take the following values:

$$\begin{aligned} \lambda &\in \mathbb{R}, \quad \alpha, \delta \in \mathbb{R}_+, \quad \beta, \mu \in \mathbb{R}^d, \quad \Delta \in \mathbb{P}_d, \\ \delta &\geq 0, 0 \leq \|\beta\|_{\Delta} < \alpha, & \text{if } \lambda > 0, \\ \delta &> 0, 0 \leq \|\beta\|_{\Delta} < \alpha, & \text{if } \lambda = 0, \\ \delta &> 0, 0 \leq \|\beta\|_{\Delta} \leq \alpha, & \text{if } \lambda < 0, \end{aligned}$$

where $\delta = 0$ or $\alpha = \|\beta\|_{\Delta}$ is a limiting case. The GH distribution is symmetric if and only if $\beta = \mathbf{0}$ and $\mu = 0$. The symmetric GH has the following elliptical pdf:

$$d_{SGH_d(\lambda, \alpha, \delta, \Delta)}(x) = \frac{\alpha^{\frac{d}{2}}}{(2\pi)^{\frac{d}{2}} |\Delta|^{\frac{1}{2}} \delta^{\lambda} K_{\lambda}(\delta \alpha)} \left(\sqrt{\delta^2 + \|x\|_{\Delta}^2} \right)^{\lambda - \frac{d}{2}} K_{\lambda - \frac{d}{2}} \left(\alpha \sqrt{\delta^2 + \|x\|_{\Delta}^2} \right),$$

where $\nu(t)$ in equation (2) is given by a GIG distribution.

B.3 NIG Distributions on \mathbb{R}^d

The NIG distribution $NIG_d(\alpha, \beta, \delta, \mu, \Delta)$ has the following pdf (v. Hammerstein, 2010, p.74):

$$d_{NIG_d(\alpha, \beta, \delta, \mu, \Delta)}(x) \propto \left(\sqrt{\delta^2 + \|x - \mu\|_{\Delta}^2} \right)^{-\frac{d+1}{2}} K_{\frac{d+1}{2}} \left(\alpha \sqrt{\delta^2 + \|x - \mu\|_{\Delta}^2} \right) e^{(\beta, x - \mu)}.$$

B.4 VG Distributions on \mathbb{R}^d

The VG distribution $VG_d(\lambda, \alpha, \beta, \mu, \Delta)$ has the following pdf (v. Hammerstein, 2010, p.74):¹²

$$d_{VG_d(\lambda, \alpha, \beta, \mu, \Delta)}(x) \propto \|x - \mu\|_{\Delta}^{-\lambda} \lambda^{-\frac{\lambda}{2}} K_{\lambda - \frac{\lambda}{2}}(\alpha \|x - \mu\|_{\Delta}^{-1}) e^{(\beta, x - \mu)}.$$

References

- D. Applebaum. *Lévy processes and stochastic calculus*, second edition, Cambridge University Press, 2009.
12. The VG pdf is bounded at $x = \mu$ if and only if $\lambda > \frac{d}{2}$.

- N. Aronszajn. Theory of Reproducing Kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.
- E. O. Barndorff-Nielsen. Processes of normal inverse gaussian type. *Finance and Stochastics*, 2:41–68, 1998.
- E. O. Barndorff-Nielsen and K. Prause. Apparent scaling. *Finance and Stochastics*, 5:103–113, 2001.
- O. E. Barndorff-Nielsen and C. Halgreen. Infinite divisibility of the hyperbolic and generalized inverse gaussian distributions. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 38:309–312, 1977.
- O. E. Barndorff-Nielsen and C. Halgreen. The variance gamma (v.g.) model for share market returns. *Journal of Business*, 63:511–524, 1990.
- C. Berg, J. P. R. Christensen, and P. Ressel. *Harmonic Analysis on Semigroups*. Springer, 1984.
- A. Berlinet and C. Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Kluwer Academic Publisher, 2004.
- M. L. Bianchi, S.T. Rachev, Y.S. Kim, and F.J. Fabozzi. Tempered infinitely divisible distributions and processes. *Theory of Probability and Its Applications (TVP)*, Society for Industrial and Applied Mathematics (SIAM), 55(1):59–86, 2010.
- S. Bochner. Lectures on fourier integrals. with an author’s supplement on monotonic functions, stieljes integrals, and harmonic analysis. In *Princeton University Press, Princeton, N.J.* 1959.
- B. Boots, G. Gordon, and A. Gretton. Hilbert space embeddings of predictive state representations. *Uncertainty in Artificial Intelligence (UAI)*, 2013.
- W. Breyermann and D. Lüthi. ghypp: A package on generalized hyperbolic distributions. 2013.
- P. Carr, H. Geman, D. B. Madan, and M. Yor. The fine structure of asset returns: an empirical investigation. *Journal of Business*, 75:305–332, 2002.
- Y. Chen, M. Welling, and A. Smola. Super-Samples from Kernel Herding. In *Uncertainty in Artificial Intelligence (UAI)*. 2010.
- R. Cont and P. Tankov. *Financial Modelling with Jump Processes*. Boca Raton: Chapman & Hall CRC Press, 2004.
- K. v. Harn F. W. Steutel. *Infinite Divisibility of Probability Distributions on the Real Line*. Monogr. Textb. Pure Appl. Math., vol. 259, Marcel Dekker Inc., 2004.
- K. Fukumizu and C. Leng. Gradient-based kernel method for feature extraction and variable selection. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 2123–2131. 2012.

- K. Fukumizu, F. R. Bach, and M. I. Jordan. Dimensionality Reduction for Supervised Learning with Reproducing Kernel Hilbert Spaces. *Journal of Machine Learning Research*, 5:73–99, 2004.
- K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf. Kernel Measures of Conditional Dependence. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 489–496. 2008.
- K. Fukumizu, L. Song, and A. Gretton. Kernel bayes’ rule: Bayesian inference with positive definite kernels. *Journal of Machine Learning Research*, pages 3753–3783, 2013.
- A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Schölkopf, and A. Smola. A kernel statistical test of independence. In *Annual Conference on Neural Information Processing Systems (NIPS)*. 2008.
- A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola. A Kernel Two-Sample Test. *Journal of Machine Learning Research*, 13:723–773, 2012.
- E. Grosswald. The student t-distribution of any degree of freedom is infinitely divisible. *Zeit. Wahrsch. Verw. Gebiete*, 36:103–109, 1976.
- S. Grünewälder, G. Lever, L. Baldassarre, M. Pontil, and A. Gretton. Modelling transition dynamics in MDPs with RKHS embeddings. In *International Conference on Machine Learning (ICML)*, pages 535–542, 2012.
- M. Kanagawa, Y. Nishiyama, A. Gretton, and K. Fukumizu. Filtering with State-Observation Examples via Kernel Monte Carlo Filter. *Neural Computation*, 28:382–444, 2016.
- W. H. Lee. Continuous and discrete properties of stochastic processes. *PhD thesis, The University of Nottingham*, 2010.
- J. R. Lloyd and Z. Ghahramani. Statistical Model Criticism using Kernel Two Sample Test. In *Annual Conference on Neural Information Processing Systems (NIPS)*. 2015.
- B. D. Madan, P. Carr, and E. C. Chang. The variance gamma process and option pricing. *European Finance Review*, 2:79–105, 1998.
- L. McCalman, S. O’Callaghan, and F. Ramos. Multi-modal estimation with kernel embeddings for learning motion models. In *IEEE International Conference on Robots and Automation (ICRA)*, 2013.
- S. Mika, B. Schölkopf, A. Smola, K. Müller, M. Scholz, and G. Rätsch. Kernel PCA and de-noising in feature spaces. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 536–542, 1999.
- K. Muandet, K. Fukumizu, F. Dinuzzo, and B. Schölkopf. Learning from Distributions via Support Measure Machines. In *Annual Conference on Neural Information Processing Systems (NIPS)*, pages 10–18. 2012.

- Y. Nishiyama, A. Boularias, A. Gretton, and K. Fukumizu. Hilbert Space Embeddings of POMDPs. In *Uncertainty in Artificial Intelligence (UAI)*, pages 644–653, 2012.
- Y. Nishiyama, M. Kanagawa, A. Gretton, and K. Fukumizu. Model-based Kernel Sum Rule. In *arXiv: 1409.5178*, 2014.
- Y. Nishiyama, A. H. Afsharinejad, S. Naruse, B. Boots, and L. Song. The Nonparametric Kernel Bayes’ Smoother. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.
- J. Nolan. Bibliography on stable distributions, processes and related topics. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.295.9970&rep=rep1&type=pdf>, 2013a.
- J. Nolan. Multivariate elliptically contoured stable distributions: theory and estimation. *Computational Statistics*, 28(5):2067–2089, 2013b.
- K. Prause. *The generalized hyperbolic model: estimation, financial derivatives, and risk measures*. Ph.D. thesis University of Freiburg, 1999.
- S. T. Rachev, Y. S. Kim, M. L. Bianchi, and F. J. Fabozzi. *Financial Models with Levy Processes and Volatility Clustering*. Wiley & Sons, 2011.
- A. Rahimi and B. Recht. Random Features for Large-Scale Kernel Machines. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2007.
- C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.
- K. Rawlik, M. Toussaint, and S. Vijayakumar. Path Integral Control by Reproducing Kernel Hilbert Space Embedding. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2013.
- J. Rosinski. Tempering stable processes. *Stochastic Processes and Their Applications*, 117(6):677–707, 2007.
- G. Samorodnitsky and M. S. Taqqu. *Stable non-Gaussian random processes : stochastic models with infinite variance*. Chapman & Hall, 1994.
- K. Sato. Lévy processes and infinitely divisible distributions. *Cambridge University Press*, 1999.
- B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, 2002.
- W. Schoutens. *Lévy Processes in Finance: Pricing Financial Derivatives*. Chichester: Wiley, 2003.
- A. Smola, A. Gretton, L. Song, and B. Schölkopf. A Hilbert space embedding for distributions. In *International Conference on Algorithmic Learning Theory (ALT)*, pages 13–31, 2007.
- L. Song, X. Zhang, A. Smola, A. Gretton, and B. Schölkopf. Tailoring Density Estimation via Reproducing Kernel Moment Matching. *International Conference on Machine Learning (ICML)*, pages 992–999, 2008.
- L. Song, J. Hwang, A. Smola, and K. Fukumizu. Hilbert Space Embeddings of Conditional Distributions with Applications to Dynamical Systems. In *International Conference on Machine Learning (ICML)*, pages 961–968, 2009.
- L. Song, B. Boots, S. M. Siddiqi, G. J. Gordon, and A. J. Smola. Hilbert Space Embeddings of Hidden Markov Models. In *International Conference on Machine Learning (ICML)*, pages 991–998, 2010.
- L. Song, A. Gretton, D. Bleckson, Y. Low, and C. Guestin. Kernel Belief Propagation. *Journal of Machine Learning Research - Proceedings Track*, 15:707–715, 2011.
- L. Song, K. Fukumizu, and A. Gretton. Kernel embedding of conditional distributions. *IEEE Signal Processing Magazine*, 30(4):98–111, 2013.
- B. Sriperumbudur, A. Gretton, K. Fukumizu, G. Lanckriet, and B. Schölkopf. Hilbert Space Embeddings and Metrics on Probability Measures. *Journal of Machine Learning Research*, 11:1517–1561, 2010.
- B. Sriperumbudur, K. Fukumizu, and G. Lanckriet. Universality, Characteristic Kernels and RKHS Embedding of Measures. *Journal of Machine Learning Research*, 12:2389–2410, 2011.
- I. Steinwart and A. Christmann. *Support Vector Machines*. Information Science and Statistics. Springer, 2008.
- O. Thorin. An extension of the notion of a generalized T-convolution. *Scandinavian Actuarial Journal*, pages 141–149, 1978.
- E. A. F. v. Hammerstein. *Generalized hyperbolic distributions: Theory and applications to GDO pricing*. Ph.D. thesis University of Freiburg, 2010.
- H. Wendland. *Scattered Data Approximation*. Cambridge University Press, Cambridge, UK, 2005.
- V.M. Zolotarev. *One-dimensional stable distributions*. Translations of mathematical monographs, American Mathematical Society, 1986.

Consistency of Cheeger and Ratio Graph Cuts

Nicolás García Trillos

Dejan Slepčev

*Department of Mathematical Sciences
Carnegie Mellon University
Pittsburgh, PA 15213, USA*

NGARCIA@ANDREW.CMU.EDU
SLEPCEV@MATH.CMU.EDU

James von Brecht

*Department of Mathematics and Statistics
California State University, Long Beach
Long Beach, CA 90840, USA*

JAMES.VONBRECHT@CSULB.EDU

Thomas Laurent

*Department of Mathematics
Loyola Marymount University
1 LMU Dr
Los Angeles, CA 90045, USA*

THOMAS.LAURENT@LMU.EDU

Xavier Bresson

*Institute of Electrical Engineering
Swiss Federal Institute of Technology (EPFL)
1015 Lausanne, Switzerland*

XAVIER.BRESSON@EPFL.CH

Editor: Matthias Hein

Abstract

This paper establishes the consistency of a family of graph-cut-based algorithms for clustering of data clouds. We consider point clouds obtained as samples of a ground-truth measure. We investigate approaches to clustering based on minimizing objective functionals defined on proximity graphs of the given sample. Our focus is on functionals based on graph cuts like the Cheeger and ratio cuts. We show that minimizers of these cuts converge as the sample size increases to a minimizer of a corresponding continuum cut (which partitions the ground truth measure). Moreover, we obtain sharp conditions on how the connectivity radius can be scaled with respect to the number of sample points for the consistency to hold. We provide results for two-way and for multiway cuts. Furthermore we provide numerical experiments that illustrate the results and explore the optimality of scaling in dimension two.

Keywords: data clustering, balanced cut, consistency, graph partitioning

1. Introduction

Partitioning data clouds in meaningful clusters is one of the fundamental tasks in data analysis and machine learning. A large class of the approaches, relevant to high-dimensional data, relies on creating a graph out of the data cloud by connecting nearby points. This allows one to leverage the geometry of the data set and obtain high quality clustering. Many of the graph-clustering approaches are based on optimizing an objective function

which measures the quality of the partition. The basic desire to obtain clusters which are well separated leads to the introduction of objective functionals which penalize the size of cuts between clusters. The desire to have clusters of meaningful size and for the approaches to be robust to outliers lead to the introduction of "balance" terms and objective functionals such as Cheeger cut and closely related edge expansion (Arora et al., 2009; Bresson and Laurent, 2012; Bresson et al., 2012; Kannan et al., 2004; Szlam and Bresson, 2010), ratio cut (Hagen and Kahng, 1992; Hein and Setzer, 2011; von Luxburg, 2007; Wei and Cheng, 1989), normalized cut (Arias-Castro et al., 2012; Shi and Malik, 2000; von Luxburg, 2007), and conductance (sparsest cut) (Arora et al., 2009; Kannan et al., 2004; Spielman and Teng, 2004). Such functionals have been extended by Bresson et al. (2013); Yu and Shi (2003) to treat multiclass partitioning. The balanced cuts above have been widely studied theoretically and used computationally. The algorithms of Andersen et al. (2006); Spielman and Teng (2004, 2013) use local clustering algorithms to compute balanced cuts of large graphs. Total variation based algorithms (Bresson et al., 2012, 2013; Hein and Bühler, 2010; Hein and Setzer, 2011; Szlam and Bresson, 2010) are also used to optimize either the conductance or the edge expansion of a graph. Closely related are the spectral approaches to clustering (Shi and Malik, 2000; von Luxburg, 2007) which can be seen as a relaxation of the normalized cuts.

In this paper we consider data clouds, $X_n = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, which have been obtained as i.i.d. samples of a measure ν with density ρ on a bounded domain D . The measure ν represents the ground truth that X_n is a sample of. In the large sample limit, $n \rightarrow \infty$, clustering methods should exhibit *consistency*. That is, the clustering of the data X_n should converge as $n \rightarrow \infty$ towards a specific clustering of the underlying ground-truth domain. In this paper we characterize in a precise manner when and how the minimizers of a ratio, Cheeger, sparsest, and normalized graph cuts, converge towards a suitable partition of the domain. We define the discrete and continuum objective functionals considered in Subsections 1.1 and 1.2 respectively, and informally state our result in Subsection 1.3.

An important consideration when investigating consistency of algorithms is how graphs on X_n are constructed. In simple terms, when building a graph on X_n , one sets a length scale ε_n such that edges between vertices in X_n are given significant weights if the distance of points they connect is ε_n or less. In some way this sets the length scale over which the geometric information is averaged when setting up the graph. Taking smaller ε_n is desirable because it is computationally less expensive and gives better resolution, but there is a price. Taking ε_n small increases the error due to randomness and in fact, if ε_n is too small, the resulting graph may not represent the geometry of D well, and consequently the discrete graph cut may be very far from the desired one. In our work we determine precisely how small ε_n can be taken for the consistency to hold. We obtain consistency results both for two-way and multi-way cuts.

To prove our results we use the variational notion of convergence known as the Γ -convergence. It is one of the standard tools of modern applied analysis that allows one to consider a limit of a family of variational problems (Braides, 2002; Dal Maso, 1993). In the recent work of García Trillos and Slepčev (2016), this notion was developed in the random discrete setting designed for the study of consistency of minimization problems on random point clouds. In particular the proof of Γ -convergence of total variation on graphs proved there, provides the technical backbone of this paper. The approach we take is general and

flexible and we believe suitable for the study of many problems involving large sample limits of minimization problems on graphs.

Background on consistency of clustering algorithms and related problems.

Consistency of clustering algorithms has been considered for a number of approaches. Pol-land (1981) has proved the consistency of k -means clustering.

Consistency of k -means clustering for paths with regularization was recently studied by Thorpe et al. (2015), using a similar viewpoint to those of this paper. Consistency for a class of single linkage clustering algorithms was shown by Hartigan (1981). Arias-Castro and Pelletier (2013) have proved the consistency of low-dimensional embeddings via the maximum variance unfolding. Consistency of spectral clustering was rigorously considered by von Luxburg, Belkin, and Bousquet (2004, 2008). These works show the convergence of all eigenfunctions of the graph Laplacian for fixed length scale $\epsilon_n = \epsilon$ which results in the limiting (as $n \rightarrow \infty$) continuum problem being a nonlocal one. Belkin and Niyogi (2006) consider the spectral problem (Laplacian eigenmaps) and show that there exists a sequence $\epsilon_n \rightarrow 0$ such that in the limit the (manifold) Laplacian is recovered, however no rate at which ϵ_n can go to zero is provided. Consistency of normalized cuts was considered by Arias-Castro, Pelletier, and Pudlo (2012) who provide a rate on $\epsilon_n \rightarrow 0$ under which the minimizers of the discrete cut functionals minimized over a specific family of subsets of X_n converge to the continuum Cheeger set. Our work improves on (Arias-Castro et al., 2012) in several ways. We minimize the discrete functionals over all discrete partitions on X_n as it is considered in practice and prove the result for the optimal, in terms of scaling, range of rates at which ϵ_n can go to zero as $n \rightarrow \infty$ for consistency to hold.

There are also a number of works which investigate how well the discrete functionals approximate the continuum ones for a particular function. Among them are works by Belkin and Niyogi (2008), Giné and Koltchinskii (2006), Hein, Audibert, and Von Luxburg (2005), Narayanan, Belkin, and Niyogi (2006), Singer (2006) and Ting, Huang, and Jordan (2010). Maier, von Luxburg, and Hein (2013) considered pointwise convergence for Cheeger and normalized cuts, both for the geometric and kNN graphs and obtained a range of scalings of graph construction on n for the convergence to hold. While these results are quite valuable, we point out that they do not imply that the minimizers of discrete objective functionals are close to minimizers of continuum functionals.

A notion of convergence suitable for showing the convergence of minimizers of approximating objective functionals converge towards a minimizer of the limit functional is the notion of Γ -convergence, which was introduced by De Giorgi in the 70's and represents a standard notion of variational convergence. For detailed exposition of the properties of Γ -convergence see the books by Braides (2002) and Dal Maso (1993). Particularly relevant to our investigation are works considering nonlocal functionals converging to the perimeter or to total variation which include works by Alberti and Bellotti (1998), Savin and Valdinoci (2012), and Esedoğlu and Otto (2015). Also related are works of Ponce (2004), who showed the Γ -convergence of nonlocal functionals related to characterization of Sobolev spaces and of Gobbino (1998) and Gobbino and Mora (2001) who investigated nonlocal approximations of the Mumford-Shah functional. In the discrete deterministic setting, works related to the Γ -convergence of functionals to continuous functionals involving perimeter include works of

Braides and Yip (2012), Chambolle, Giacomini, and Lussardi (2010), and van Gennip and Bertozzi (2012).

1.1 Graph partitioning

The balanced cut objective functionals we consider are relevant to general graphs (not just the ones obtained from point clouds). We introduce them here.

Given a weighted graph $G = (X, W)$ with vertex set $X = \{x_1, \dots, x_n\}$ and weight matrix $W = \{w_{ij}\}_{1 \leq i, j \leq n}$, the balanced graph cut problems we consider take the form

$$\text{Minimize } \frac{\text{Cut}(Y, Y^c)}{\text{Bal}(Y, Y^c)} := \frac{\sum_{x \in Y} \sum_{x' \in Y^c} w_{ij}}{\text{Bal}(Y, Y^c)} \quad \text{over all nonempty } Y \subsetneq X. \quad (1.1)$$

That is, we consider the class of problems with $\text{Cut}(Y, Y^c)$ as the numerator together with different balance terms. For $Y \subset X$ let $|Y|$ be the ratio between the number of vertices in Y and the number of vertices in X , that is $|Y| = \frac{|X|}{n}$. Well-known balance terms include

$$\text{Bal}_R(Y, Y^c) = 2|Y||Y^c| \quad \text{and} \quad \text{Bal}_C(Y, Y^c) = \min(|Y|, |Y^c|), \quad (1.2)$$

which correspond to ratio cut (Hagen and Kalmg, 1992; Hein and Setzer, 2011; von Luxburg, 2007; Wei and Cheng, 1989) and Cheeger cut (Arya et al., 2009; Cheeger, 1970; Chung, 1997; Kannan et al., 2004) respectively¹, as well as

$$\text{Bal}_S(Y, Y^c) = 2 \frac{\deg(Y) \deg(Y^c)}{\deg(X)^2} \quad \text{and} \quad \text{Bal}_N(Y, Y^c) = \frac{\min(\deg(Y), \deg(Y^c))}{\deg(X)}, \quad (1.3)$$

where $\deg(Y) = \sum_{i=1}^n \sum_{j \neq i} w_{ij}$ is the sum of weighted degrees of all vertices in Y , which correspond to sparsest cut (Arya et al., 2009; Kannan et al., 2004; Spielman and Teng, 2004) and normalized cut (Arias-Castro et al., 2012; Shi and Malik, 2000; von Luxburg, 2007) respectively. We refer to a pair $\{Y, Y^c\}$ that solves (1.1) as an *optimal balanced cut of the graph*. Note that a given graph $G = (X, W)$ may have several optimal balanced cuts (although one expects that generically the optimal cut is unique, since a small perturbation of the weights of a graph with a non-unique minimal balanced cut, is almost sure to lead to only one of them having the least energy).

We are also interested in multi-class balanced cuts. Specifically, in order to partition the set X into $R \geq 3$ clusters, we consider the following ratio cut functional:

$$\text{Minimize } \sum_{r=1}^R \frac{\text{Cut}(Y_r, Y_r^c)}{|Y_r|}, \quad Y_r \cap Y_s = \emptyset \quad \text{if } r \neq s, \quad \bigcup_{r=1}^R Y_r = X. \quad (1.4)$$

1.2 Continuum partitioning

Given a bounded and connected open domain $D \subset \mathbb{R}^d$ and a probability measure ν on D , with positive density $\rho > 0$, we define the class of balanced domain cut problems in an analogous way. A balanced domain-cut problem takes the form

$$\text{Minimize } \frac{\text{Cut}_\rho(A, A^c)}{\text{Bal}_\rho(A, A^c)}, \quad A \subset D \quad \text{with } 0 < \nu(A) < 1. \quad (1.5)$$

¹ The factor of 2 in the definition of $\text{Bal}_R(Y, Y^c)$ is introduced to simplify the computations in the remainder. We remark that when using Bal_R problem (1.1) is equivalent to the usual ratio cut problem.

where $A^c = D \setminus A$. Just as the graph cut term $\text{Cut}(Y, Y^c)$ in (1.1) provides a weighted (by W) measure of the boundary between Y and Y^c , the cut term $\text{Cut}_\rho(A, A^c)$ for a domain denotes a ρ^2 -weighted area of the boundary between the sets A and A^c . Assuming that $\partial_D A := \partial A \cap D$ (the boundary between A and A^c) is a smooth curve (in 2d), surface (in 3d) or manifold (in 4d+), we can define

$$\text{Cut}_\rho(A, A^c) := \int_{\partial_D A} \rho^2(x) \, dS(x). \quad (1.6)$$

We only consider cuts with weight ρ^2 , since they appear as the limit of the discrete cuts we consider in this paper, as indicated in subsection 1.3.

For our results and analysis we need the notion of continuum cut which is defined for sets with less regular boundary. We present the required notions of geometric measure theory and the rigorous and mathematically precise formulation of problem (1.5) in Subsection 3.1.

If $\rho(x) = 1$ then $\text{Cut}_\rho(A, A^c)$ simply corresponds to arc-length (in 2d) or surface area (in 3d). In the general case, the presence of $\rho^2(x)$ in (1.6) indicates that the regions of low density are easier to cut, so $\partial_D A$ has a tendency to pass through regions in D of low density. As in the graph case, we consider balance terms

$$\text{Bal}_\rho(A, A^c) = 2|A| |A^c| \quad \text{and} \quad \text{Bal}_\rho(A, A^c) = \min(|A|, |A^c|), \quad (1.7)$$

which correspond to weighted continuous equivalents of the ratio cut and the Cheeger cut. In the continuum setting $|A|$ stands for the total ν -content of the set A , that is,

$$|A| = \nu(A) = \int_A \rho(x) \, dx. \quad (1.8)$$

We also consider balance terms

$$\text{Bal}_\rho(A, A^c) = 2|A|_{\rho^2} |A^c|_{\rho^2} \quad \text{and} \quad \text{Bal}_\rho(A, A^c) = \min(|A|_{\rho^2}, |A^c|_{\rho^2}), \quad (1.9)$$

which correspond to weighted continuous equivalents of the sparsest cut and the normalized cut. Here $|A|_{\rho^2}$ stands for

$$|A|_{\rho^2} = \frac{1}{\int_D \rho^2(x) \, dx} \int_A \rho^2(x) \, dx. \quad (1.10)$$

We refer to a pair $\{A, A^c\}$ that solves (1.5) as an *optimal balanced cut of the domain*.

The continuum equivalent of the multiway cut problem (1.4) reads

$$\underset{(A_1, \dots, A_R)}{\text{Minimize}} \sum_{r=1}^R \frac{\text{Cut}_\rho(A_r, A_r^c)}{|A_r|}, \quad (1.11)$$

where (A_1, \dots, A_R) is an R -tuple of measurable subsets of D such that $\nu(A_r \cap A_s) = 0$ if $r \neq s$, and $\nu\left(D \setminus \bigcup_{r=1}^R A_r\right) = 0$.

1.3 Consistency of partitioning of data clouds

Let $\mathbf{x}_1, \dots, \mathbf{x}_n, \dots$ be a sequence of i.i.d random points drawn from an underlying ground-truth measure ν . Throughout the paper ν is a probability measure supported on a bounded, open set with Lipschitz boundary D . Furthermore we assume that ν has continuous density $\rho : D \rightarrow \mathbb{R}$ and that $0 < \lambda \leq \rho \leq \Lambda$ on D ; in other words, ρ is bounded below and above by positive constants. We denote by $X_n = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, the set consisting of the first n data points.

To extract the desired information from the point cloud X_n , one builds a graph by connecting nearby points. More precisely, let $\eta : \mathbb{R}^d \rightarrow [0, \infty)$ be a radially symmetric kernel, radially decreasing, and decaying to zero sufficiently fast. We introduce a parameter ε which basically describes over which length scale the data points are connected. For $i, j \in \{1, \dots, n\}$, we consider the weight

$$w_{ij} = \eta\left(\frac{\mathbf{x}_i - \mathbf{x}_j}{\varepsilon}\right). \quad (1.12)$$

As more data points are available one takes smaller ε to obtain increased resolution. That is, one sets the length scale ε based on the number of available data points: $\varepsilon = \varepsilon_n$. We investigate under what scaling of ε_n on n the optimal balanced cuts (that is, minimizers of (1.1)) of the graph $\mathcal{G}_n = (X_n, W_n)$ converge towards optimal balanced cuts in the continuum setting (minimizers of (1.5)). On Figure 1, we illustrate the partitioning of a data cloud sampled from the uniform distribution on the given domain D .

Informal statement of (a part of) the main results. Consider $d \geq 2$ and assume the continuum balanced cut (1.5) has a unique minimizer $\{A, A^c\}$. Consider $\varepsilon_n > 0$ such that $\lim_{n \rightarrow \infty} \varepsilon_n = 0$ and

$$\lim_{n \rightarrow \infty} \frac{(\log n)^{pd}}{n^{1/d}} \frac{1}{\varepsilon_n} = 0, \quad (1.13)$$

where $p_d = 1/d$ for $d \geq 3$ and $p_2 = 3/4$. Then almost surely the minimizers, $\{Y_n, Y_n^c\}$, of the balanced cut (1.1) of the graph \mathcal{G}_n , converge to $\{A, A^c\}$. Moreover, after appropriate rescaling, almost surely the minimum of problem (1.1) converges to the minimum of (1.5). The result also holds for multiway cuts. That is, the minimizers of (1.4) converge towards minimizers of (1.11).

Let us make the notion of convergence of discrete partitions $\{Y_n, Y_n^c\}$ to continuum partitions $\{A, A^c\}$ precise.

To be able to easily account for the invariance $\{Y_n, Y_n^c\} = \{Y_n^c, Y_n\}$, let $Y_{n,1} = Y_n$ and $Y_{n,2} = Y_n^c$. Let $\mathbf{1}_{Y_{n,i}} : X_n \rightarrow \{0, 1\}$ for $i = 1, 2$ be the characteristic function of $Y_{n,i}$ (on the set X_n). We say that $\{Y_n, Y_n^c\}$ converge towards $\{A, A^c\}$ as $n \rightarrow \infty$ if there is a sequence of indices $I : \mathbb{N} \rightarrow \{1, 2\}$ such that

$$\frac{1}{n} \sum_{i=1}^n \mathbf{1}_{Y_{n,I(n)}}(\mathbf{x}_i) \delta_{\mathbf{x}_i} \xrightarrow{w} \mathbf{1}_A \nu \quad (1.14)$$

where \xrightarrow{w} denotes the weak convergence of measures (see Dudley (2002)). Since by assumption on the points \mathbf{x}_i it holds that $\frac{1}{n} \sum_{i=1}^n \delta_{\mathbf{x}_i} \xrightarrow{w} \nu$, the property (1.14) is equivalent

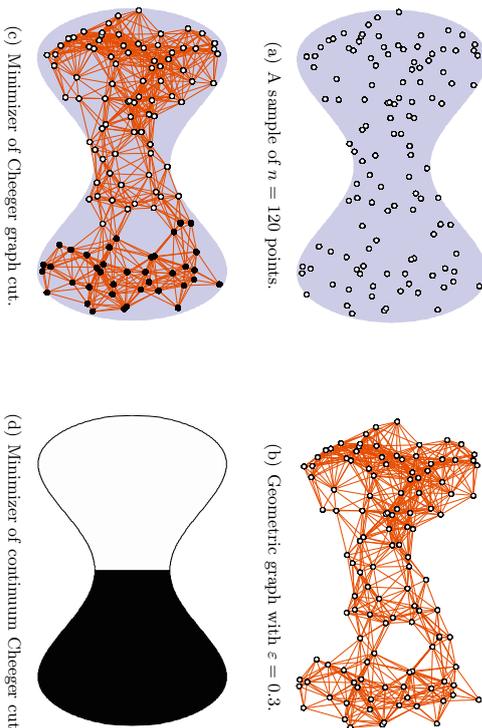


Figure 1: Given the sample of Figure (a), graph is constructed using $\eta(z) = \mathbf{1}_{\{|z| \leq 1\}}$ and $\varepsilon = 0.3$, as illustrated on Figure (b). On Figure (c) we present the solution to the Cheeger graph-cut problem obtained using algorithm of Bresson et al. (2012). A solution to the continuum Cheeger-cut problem is illustrated in Figure (d).

$$\text{to} \quad \frac{1}{n} \sum_{k=1}^n \mathbf{1}_{Y_{n,3-(\alpha)}(\mathbf{x}_k)} \delta_{\mathbf{x}_k} \stackrel{w}{\sim} \mathbf{1}_{A^c \nu}$$

In Section 2 we discuss this topology in more detail and present a conceptually clearer framework, which applies to general functions (not just characteristic functions of sets).

Let us also indicate briefly why the weight ρ^2 present in the weighted perimeter (1.6) can be expected to appear in the limit of balanced graph cuts (1.1). Let $\nu_n = \frac{1}{n} \sum_{i=1}^n \delta_{\mathbf{x}_i}$ be the empirical measure of the sample $X_n = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$. Let $A \subset D$ be a set with smooth boundary and let $A_n = A \cap X_n$. Then, using $\eta_\varepsilon(z) = \eta(z/\varepsilon)/\varepsilon^d$ we get

$$\begin{aligned} \frac{1}{n^2 \varepsilon^d} \text{Cut}(A_n, A_n^c) &= \frac{1}{n^2} \sum_{\mathbf{x}_i \in A_n} \sum_{\mathbf{x}_j \in A_n^c} \frac{1}{\varepsilon^d} \eta \left(\frac{\mathbf{x}_i - \mathbf{x}_j}{\varepsilon} \right) \\ &= \int_D \int_D \mathbf{1}_{A_n}(x) \mathbf{1}_{A_n^c}(y) \eta_\varepsilon(x-y) d\nu_n(x) d\nu_n(y) \\ &\sim \int_D \int_D \mathbf{1}_A(x) \mathbf{1}_{A^c}(y) \eta_\varepsilon(x-y) \rho(x) \rho(y) d\nu dx \\ &\sim C \int_{D \cap \partial A} \rho^2(x) dS(x). \end{aligned}$$

The factor $1/(n^2 \varepsilon^d)$ in front of the cut above is accounted for in the way we scale the cuts, see (5.6). We remark that the above just provides a rough heuristic idea as to what weight should be expected. It does not serve as a basis for our proof, since the optimal balanced graph cuts $\{Y_n^c, Y_n^c\}$ (minimizer of (1.1)) could be rather different from $\{A \cap X_n, A^c \cap X_n\}$ where $\{A, A^c\}$ is the optimal balanced domain cut (minimizer of (1.5)).

The reason for the presence of ρ in (1.8) is clear since the particles are drawn from the measure, ν , with density ρ , and thus the empirical measures of the sample, ν_n , converge to ν . Let us now indicate the reason for the presence of ρ^2 in (1.10). Namely for the graph weights given by (1.12) and A, X_n and A_n as above

$$\begin{aligned} \frac{1}{n^2 \varepsilon^d} \text{deg}(A_n) &= \frac{1}{n^2} \sum_{\mathbf{x}_i \in A_n} \sum_{\mathbf{x}_j \in X_n} \frac{1}{\varepsilon^d} \eta \left(\frac{\mathbf{x}_i - \mathbf{x}_j}{\varepsilon} \right) = \int_D \int_D \mathbf{1}_{A_n}(x) \eta_\varepsilon(x-y) d\nu_n(x) d\nu_n(y) \\ &\sim \int_D \int_D \mathbf{1}_A(x) \eta_\varepsilon(x-y) \rho(x) \rho(y) dy dx \\ &\sim C_\eta \int_D \mathbf{1}_A(x) \rho^2(x) dx. \end{aligned}$$

Therefore,

$$\frac{\text{deg}(A_n)}{\text{deg}(X_n)} \sim \frac{1}{\int_D \rho^2(x) dx} \int_D \mathbf{1}_A(x) \rho^2(x) dx = |A| \rho^2.$$

Since the proofs are analogous in most of the paper, we only consider the ratio and Cheeger cuts in detail and only comment briefly on sparsest and normalized cuts.

Remark 1 (Optimality of scaling of ε_n for $d \geq 3$) *If $d \geq 3$ then the rate presented in (1.13) is sharp in terms of scaling. Namely for $D = (0, 1)^d$, and ν the Lebesgue measure on D and η compactly supported, it is known from graph theory (see (Goel et al., 2004; Gupta and Kumar, 1999; Penrose, 1999)) that there exists a constant $c > 0$ such that if $\varepsilon_n < \frac{c \log(n)^{1/d}}{n^{1/d}}$ then the weighted graph associated to (X_n, W_n) is disconnected with high probability. The resulting optimal discrete cuts have zero energy, but may be very far from the optimal continuum cuts.*

While the above example demonstrates the optimality of our results, we remark that the convergence fails because the lack of connectedness of random geometric graphs (with connectivity radius below the before mentioned threshold) leads to undesirable partitions. Considering different objective functionals which are still based on perimeter, but more strongly penalize existence of small connected components, or considering different graph constructions (for example by restricting attention to the giant component) could lead to convergence even for some scaling ε_n below the connectivity threshold $\frac{1}{n^{1/d}} \ll \varepsilon_n < \frac{(\log n)^{1/d}}{n^{1/d}}$.

Remark 2 *In case $d = 2$ the connectivity threshold for a random geometric graph is $\varepsilon_n = \frac{\log(n)^{1/2}}{n^{1/2}}$, which is below the rate for which we can establish the consistency of balanced cuts. Thus, an interesting open problem is to determine if the consistency results we present in*

this paper are still valid when the parameter ε_n is taken below the rate $\frac{\log(n)^{3/4}}{n^{1/2}}$ we obtained the proof for, but above the connectivity rate. In particular we are interested in determining if connectivity is the determining factor in order to obtain consistency of balance graph cuts. We numerically explore this problem in Section 8.

1.4 Outline

In Section 2 we introduce the notion of convergence we use to bridge between discrete and continuum partitions. In particular this notion of convergence allows us to consider the discrete and continuum objective functionals in a common metric space, which we denote by TL^1 . This notion of convergence relies on some of the notions of the theory of optimal transportation which we recall. We also recall results on optimal min-max matching between the random sample and the underlying measure (Proposition 5), which are needed in the proof of the convergence. They represent the main estimates which account for randomness. The rest of the arguments in the paper are not probabilistic.

In Section 3 we study more carefully the notion of continuum partitioning (1.5). We introduce the notion of total variation of functions on D in Subsection 3.1 and recall some of its basic properties. This enables us to introduce, in Subsection 3.2, the general setting for problem (1.5) where desirable properties such as lower semicontinuity and existence of minimizers hold. In Section 4 we give the precise statement of the consistency result, both for the two-way cuts (Theorem 9) and the multi-way cuts (Theorem 12). Proving that minimizers of discrete balanced cuts converge to optimal continuum balanced cuts is reduced to proving that the discrete balanced-cut objective functionals converge (in the sense of the notion of variational convergence known as Γ -convergence) to continuum balanced-cut objective functionals. In Section 5 we recall the definition of Γ -convergence and its basic properties. In Subsection 5.1 we recall the results on Γ -convergence of graph total variation which provide the backbone for our results. Section 6 contains the proof of the Theorem 9 and Section 7 the proof of Theorem 12. Finally, in Section 8 we present numerical experiments which illustrate our results; we also investigate the issues related to Remark 2.

2. From Discrete to Continuum

Let $\mathbf{x}_1, \dots, \mathbf{x}_n, \dots$ be a sequence of i.i.d random points drawn from an underlying ground-truth measure ν . For the two-class case, our main result shows that a sequence of partitions $\{Y_n, Y_n^c\}$ of the point clouds $X_n = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subset D$ converges toward a continuum partition $\{A, A^c\}$ of the domain D . In this section we expand on the notion of convergence introduced in Subsection 1.3 to compare the discrete and continuum partitions. We give an equivalent definition for such type of convergence which turns out to be more useful for the computations in the remainder.

Associated to the partitions $\{Y_n, Y_n^c\}$ are the characteristic functions of Y_n and Y_n^c , namely $\mathbf{1}_{Y_n} : X_n \rightarrow \{0, 1\}$ and $\mathbf{1}_{Y_n^c} : X_n \rightarrow \{0, 1\}$. Let $\nu_n = \frac{1}{n} \sum_{i=1}^n \delta_{\mathbf{x}_i}$ be the empirical measures associated to X_n . Note that $\mathbf{1}_{Y_n}, \mathbf{1}_{Y_n^c} \in L^1(\nu_n)$. Likewise a continuum partition of D by measurable sets A and $A^c = D \setminus A$ can be described via the characteristic functions $\mathbf{1}_A : D \rightarrow \{0, 1\}$ and $\mathbf{1}_{A^c} : D \rightarrow \{0, 1\}$. These too can be considered as L^1 functions, but with respect to the measure ν rather than ν_n .

We compare the partitions $\{Y_n, Y_n^c\}$ and $\{A, A^c\}$ by comparing the associated characteristic functions. To do so, we need a way of comparing L^1 functions with respect to different measures. We follow the approach of García Trillos and Slepčev (2016). We denote by $\mathcal{B}(D)$ the Borel σ -algebra on D and by $\mathcal{P}(D)$ the set of Borel probability measures on D . The set of objects of our interest is

$$TL^1(D) := \{(\mu, f) : \mu \in \mathcal{P}(D), f \in L^1(\mu)\}.$$

Note that $(\nu_n, \mathbf{1}_{Y_n})$ and $(\nu, \mathbf{1}_A)$ both belong to TL^1 . To compare functions defined with respect to different measures, say (μ, f) and (θ, g) in TL^1 , we need a way to say for which $(x, y) \in \text{supp}(\mu) \times \text{supp}(\theta)$ should we compare $f(x)$ and $g(y)$. The notion of *coupling* (or *transportation plan*) between μ and θ , provides a way to do that. A coupling between $\mu, \theta \in \mathcal{P}(D)$ is a probability measure π on the product space $D \times D$, such that the marginal on the first variable is μ and the marginal on the second variable is θ . The set of couplings $\Gamma(\mu, \theta)$ is thus

$$\Gamma(\mu, \theta) = \{\pi \in \mathcal{P}(D \times D) : (\forall U \in \mathcal{B}(D)) \pi(U \times D) = \mu(U) \text{ and } \pi(D \times U) = \theta(U)\}.$$

For (μ, f) and (θ, g) in $TL^1(D)$ we define the distance

$$d_{TL^1}((\mu, f), (\theta, g)) = \inf_{\pi \in \Gamma(\mu, \theta)} \iint_{D \times D} |x - y| + |f(x) - g(y)| d\pi(x, y). \quad (2.1)$$

This is the distance that we use to compare L^1 functions with respect to different measures.

It is motivated by optimal transportation distances (such as the Wasserstein distance and the earth-mover distance, see (García Trillos and Slepčev, 2016, 2015; Villani, 2003) and references therein). Indeed, the distance (2.1) can be seen as an optimal transportation distance between measures supported on the graphs of the functions f and g , as discussed in García Trillos and Slepčev (2016). To better understand it here, we focus on the case that one of the measures, say μ , is absolutely continuous with respect to the Lebesgue measure, as this case is relevant for us when passing from discrete to continuum. In this case, the convergence in TL^1 space can be formulated in simpler ways using transportation maps instead of couplings to match the measures. Given a Borel map $T : D \rightarrow D$ and $\mu \in \mathcal{P}(D)$, the *push-forward* of μ by T , denoted by $T\#\mu \in \mathcal{P}(D)$ is given by:

$$T\#\mu(A) := \mu(T^{-1}(A)), \quad A \in \mathfrak{B}(D).$$

A Borel map $T : D \rightarrow D$ is a *transportation map* between the measures $\mu \in \mathcal{P}(D)$ and $\theta \in \mathcal{P}(D)$ if $\theta = T\#\mu$. Associated to a transportation map T , there is a plan $\pi_T \in \Gamma(\mu, \theta)$ given by $\pi_T := (\text{Id} \times T)\#\mu$, where $(\text{Id} \times T)(x) = (x, T(x))$.

We note that if $\theta = T\#\mu$, then the following change of variables formula holds for any $f \in L^1(\theta)$

$$\int_D f(y) d\theta(y) = \int_D f(T(x)) d\mu(x). \quad (2.2)$$

In order to give the desired interpretation of convergence in TL^1 we also need the notion of a stagnating sequence of transportation maps. Given $\mu_n \in \mathcal{P}(D)$, for $n = 1, \dots$

and $\mu \in \mathcal{P}(D)$, a sequence $\{T_n\}_{n \in \mathbb{N}}$ of transportation maps between μ and μ_n (meaning that $T_n\#\mu = \mu_n$) is *stagnating* if

$$\int_D |x - T_n(x)| d\mu(x) \rightarrow 0 \quad \text{as } n \rightarrow \infty. \quad (2.3)$$

This notion is relevant to our considerations because for the measure ν and its empirical measures ν_n there exists (with probability one) a sequence of stagnating transportation maps $T_n\#\nu = \nu_n$. The idea is that as $n \rightarrow \infty$ the mass from ν needs to be moved only very little to be matched with the mass of ν_n . We make this precise in Proposition 5

We now provide the desired interpretation of the convergence in TL^1 , which is a part of Proposition 3.12 of García Trillos and Slepčev (2016).

Proposition 3 *Consider a measure $\mu \in \mathcal{P}(D)$ which is absolutely continuous with respect to the Lebesgue measure. Let $(\mu, f) \in TL^1(D)$ and let $\{(\mu_n, f_n)\}_{n \in \mathbb{N}}$ be a sequence in $TL^1(D)$. The following statements are equivalent:*

- (i) $(\mu_n, f_n) \xrightarrow{TL^1} (\mu, f)$ as $n \rightarrow \infty$.
- (ii) $\mu_n \xrightarrow{w} \mu$ and there exists a stagnating sequence of transportation maps $T_n\#\mu = \mu_n$ such that:

$$\int_D |f(x) - f_n(T_n(x))| d\mu(x) \rightarrow 0, \text{ as } n \rightarrow \infty. \quad (2.4)$$
- (iii) $\mu_n \xrightarrow{w} \mu$ and for any stagnating sequence of transportation maps $T_n\#\mu = \mu_n$ convergence (2.4) holds.

The previous proposition implies that in order to show that (μ_n, f_n) converges to (μ, f) in the TL^1 -sense, it is enough to find a sequence of stagnating transportation maps $T_n\#\mu = \mu_n$ and then show the L^1 convergence of $f_n \circ T_n$ to f in $L^1(\mu)$. An important feature of Proposition 3 is that there is complete freedom on what sequence of transportation maps $\{T_n\}_{n \in \mathbb{N}}$ to take, as long as it is stagnating. In particular this shows that if $\mu_n = \mu$ for all n then the convergence in TL^1 is equivalent to convergence in $L^1(\mu)$.

Remark 4 *Suppose that the sequence of probability measures $\{\mu_n\}_{n \in \mathbb{N}}$ is such that $\mu_n \xrightarrow{w} \mu$. Let $f_n \in L^1(\mu_n)$ and let $f \in L^1(\mu)$. With a slight abuse of notation we say that $f_n \xrightarrow{TL^1} f$ whenever $(\mu_n, f_n) \xrightarrow{TL^1} (\mu, f)$. In particular when we write $f_n \xrightarrow{TL^1} f$ it should be clear what the corresponding measures μ_n, μ are.*

To obtain the scaling of (1.13) we need a stagnating sequence of transportation maps between ν and $\{\nu_n\}_{n \in \mathbb{N}}$ with precise information on the rate at which convergence (2.3) occurs. More precisely for some of our considerations we need the control of $T_n(x) - x$ in the stronger $L^\infty(\nu)$ -norm, rather than in the $L^1(\nu)$ -norm. Since the typical distance between nearby points is of order $n^{-1/d}$ the typical transportation distance, $T_n(x) - x$, must be at least of that order. The optimal upper bound on the $L^\infty(\nu)$ -norm of $T_n - I$ however has an extra logarithmic correction. In particular in García Trillos and Slepčev (2015) it was shown that:

Proposition 5 *Let D be an open, connected and bounded subset of \mathbb{R}^d which has Lipschitz boundary. Let ν be a probability measure on D with density ρ which is bounded from below and from above by positive constants. Let $\mathbf{x}_1, \dots, \mathbf{x}_n, \dots$ be a sequence of independent random points distributed on D according to measure ν and let $\nu_n = \frac{1}{n} \sum_{i=1}^n \delta_{\mathbf{x}_i}$. Then there is a constant $C > 0$ (that depends on D and ρ) such that with probability one there exists a sequence of transportation maps $\{T_n\}_{n \in \mathbb{N}}$ from ν to ν_n ($T_n\#\nu = \nu_n$) and such that:*

$$\limsup_{n \rightarrow \infty} \frac{n^{1/d} \|\text{Id} - T_n\|_{L^\infty(\nu)}}{(\log n)^{p_d}} \leq C, \quad (2.5)$$

where the power p_d is equal to $1/d$ if $d \geq 3$ and equal to $3/4$ if $d = 2$.

The optimality of the upper bound is discussed in García Trillos and Slepčev (2015). If $d \geq 3$ it follows from the fact that for n large, with large probability there exists a ball of radius comparable to $((\ln n)/n)^{1/d}$ which contains none of the points $\mathbf{x}_1, \dots, \mathbf{x}_n$.

Having defined the TL^1 -convergence for functions, we turn to the TL^1 -convergence for partitions. When defining a notion of convergence for sequences of partitions $\{Y_1^n, \dots, Y_R^n\}$, we need to address the inherent ambiguity that arises from the fact that both $\{Y_1^n, \dots, Y_R^n\}$ and $\{Y_{(1)}^n, \dots, Y_{(R)}^n\}$ refer to the same partition for any permutation P of $\{1, \dots, R\}$. Having the previous observation in mind, the convergence of partitions is defined in a natural way.

Definition 6 *The sequence $\{Y_1^n, \dots, Y_R^n\}_{n \in \mathbb{N}}$ where $\{Y_1^n, \dots, Y_R^n\}$ is a partition of X_n , converges in the TL^1 -sense to the partition $\{A_1, \dots, A_R\}$ of D , if there exists a sequence of permutations $\{P_n\}_{n \in \mathbb{N}}$ of the set $\{1, \dots, R\}$, such that for every $r \in \{1, \dots, R\}$,*

$$\left(\nu_n, \mathbf{1}_{Y_{P_n(r)}^n} \right) \xrightarrow{TL^1} (\nu, \mathbf{1}_{A_r}) \quad \text{as } n \rightarrow \infty.$$

We note that the definition above is equivalent to

$$\frac{1}{n} \sum_{i=1}^n \mathbf{1}_{Y_{P_n(r)}^n}(x_i) \delta_{x_i} \xrightarrow{w} \mathbf{1}_{A_r} \nu \quad (2.6)$$

for all $r = 1, \dots, R$ which is analogous to the definition in (1.14) which we gave in Subsection (1.3) when discussing the main result. The equivalence follows from the fact that the TL^1 metric (2.1) can be seen as the distance between the graphs of functions, considered as measures. Namely given $(\mu, f), (\theta, g) \in TL^1(D)$, let $\Gamma_f = (\text{Id} \times f)\#\mu$ and $\Gamma_g = (\text{Id} \times g)\#\theta$ be the measures representing the graphs. Consider $d(\Gamma_f, \Gamma_g) := d_{\Gamma L}(\mu, f), (\theta, g)$. Proposition 3.3 in García Trillos and Slepčev (2016) (also see the paragraph right after Remark 3.1) implies that this distance metrizes the weak convergence of measures on the family of graph measures. Therefore the convergence of partitions of Definition 6 is equivalent to one given in (2.6).

We end this section by making some remarks about why the TL^1 -metric is a suitable metric for considering consistency problems. On one hand if one considers a sequence of minimizers $\{Y_n, Y_n^c\}$ of the graph balanced cut (1.1) the topology needs to be weak enough

for the sequence of minimizers to be guaranteed to converge (at least along a subsequence). Mathematically speaking the topology needs to be weak enough for the sequence to be pre-compact. On the other hand the topology has to be strong enough for one to be able to conclude that the limit of a sequence of minimizers is a minimizer of the continuum balanced cut energy. In Proposition 21 and Lemma 23 we establish that the TL -metric satisfies both of the desired properties.

Finally we point out that our approach from discrete to continuum can be interpreted as an extrapolation or extension approach, as opposed to restriction viewpoint. Namely when comparing (μ_n, f_n) and (μ, f) where μ_n is discrete and μ is absolutely continuous with respect to the Lebesgue measure we end up comparing two L^1 functions with respect to the Lebesgue measure, namely $f_n \circ T_n$ and f , in (2.4). Therefore $f_n \circ T_n$ used in Proposition 3 can be seen as a continuum representative (extrapolation) of the discrete f_n . We think that this approach is more flexible and suitable for the task than the, perhaps more common, approach of comparing the discrete and continuum by restricting the continuum object to the discrete setting (this would correspond to considering $f|_{\text{supp}(f_n)}$ and comparing it to f_n).

3. Continuum partitioning: rigorous setting

We first recall the general notion of (weighted) total variation and some notions of analysis and geometric measure theory.

3.1 Total Variation

Let D be an open and bounded domain in \mathbb{R}^d with Lipschitz boundary and let $\rho : D \rightarrow (0, \infty)$ be a continuous density function. We let ν be the measure with density ρ . We assume that ρ is bounded above and below by positive constants, that is, $\lambda \leq \rho \leq \Lambda$ on D for some $\Lambda \geq \lambda > 0$.

Given a function $u \in L^1(\nu)$, we define the weighted (by weight ρ^2) total variation of u by:

$$TV(u; \nu) := \sup \left\{ \int_D u(x) \text{div}(\Phi(x)) \, dx : \Phi(x) \in C_c^1(D; \mathbb{R}^d), \quad |\Phi(x)| \leq \rho^2(x) \right\}, \quad (3.1)$$

where in the above $C_c^1(D; \mathbb{R}^d)$ denotes the set of C^1 -functions from D to \mathbb{R}^d , whose support is compactly contained in D . If u is regular enough then the weighted total variation can be written as

$$TV(u; \nu) = \int_D |\nabla u| \rho^2(x) \, dx. \quad (3.2)$$

Also, given that $\rho : D \rightarrow \mathbb{R}$ is continuous, if $u \in \mathbf{1}_A$ is the characteristic function of a set $A \subseteq \mathbb{R}^d$ with C^1 boundary, then

$$TV(\mathbf{1}_A; \nu) = \int_{\partial A \cap D} \rho^2(x) \, d\mathcal{H}^{d-1}(x), \quad (3.3)$$

where \mathcal{H}^{d-1} represents the $(d-1)$ -dimensional Hausdorff measure in \mathbb{R}^d . In case ρ is a constant (ν is the uniform distribution), the functional $TV(\cdot; \nu)$ reduces to a multiple of

the classical total variation and in particular (3.3) reduces to a multiple of the surface area of the portion of ∂A contained in D .

Since ρ is bounded above and below by positive constants, a function $u \in L^1(\nu)$ has finite weighted total variation if and only if it has finite classical total variation. Therefore, if $u \in L^1(\nu)$ with $TV(u; \nu) < \infty$, then u is a BV function and hence it has a distributional derivative Du which is a Radon measure (see Chapter 13 of Leoni (2009)). We denote by $|Du|$ the total variation of the measure Du and denote by $|Du|_{\rho^2}$ the measure determined by

$$d|Du|_{\rho^2} = \rho^2(x) d|Du|. \quad (3.4)$$

By Theorem 4.1 of Baldi (2001)

$$TV(u; \nu) = |Du|_{\rho^2}(D) = \int_D \rho^2(x) \, d|Du|(x). \quad (3.5)$$

A simple consequence of the definition of the weighted TV is its lower semicontinuity with respect to L^1 -convergence. More precisely, if $u_k \xrightarrow{L^1(\nu)} u$ then

$$TV(u; \nu) \leq \liminf_{k \rightarrow \infty} TV(u_k; \nu). \quad (3.6)$$

Finally, for $u \in BV(D)$, the co-area formula

$$TV(u; \nu) = \int_{\mathbb{R}} TV(\mathbf{1}_{\{u>t\}}; \nu) \, dt,$$

relates the weighted total variation of u with the weighted total variation of its level sets. A proof of this formula can be found in Bellettini, Bouchitté, and Fragalà (1999). For a proof of the formula in the case that ρ is constant see Leoni (2009).

In the remainder of the paper, we write $TV(u)$ instead of $TV(u; \nu)$ when the context is clear.

3.2 Continuum partitioning

We use the total variation to rigorously formulate the continuum partitioning problem (1.5). The precise definition of the $\text{Cut}_{\rho}(A, A^c)$ functional in (1.5) is

$$\text{Cut}_{\rho}(A, A^c) = TV(\mathbf{1}_A; \nu),$$

where $TV(\mathbf{1}_A; \nu)$ is defined in (3.1). We note that $TV(\mathbf{1}_A; \nu)$ is equal to $TV(\mathbf{1}_{A^c}; \nu)$, and is the perimeter of the set A in D weighted by ρ^2 .

Recall that $|D| = \nu(D)$, as defined in (1.8). Given that ν is a probability measure supported on D we have $|D| = 1$. We now formulate the balance terms defined by (1.7) and (1.8) using characteristic functions. We start by extending the balance term to arbitrary functions $u \in L^1(\nu)$:

$$B_R(u) = \int_D |u(x) - \text{mean}_{\rho}(u)| \rho(x) \, dx \quad \text{and} \quad B_C(u) = \min_{c \in \mathbb{R}} \int_D |u(x) - c| \rho(x) \, dx, \quad (3.7)$$

where $\text{mean}_{\rho}(u)$ denotes the mean/expectation of $u(x)$ with respect to the measure $dv = \rho dx$.

Analogously, using the symbol $\int_D f(x)\rho^2(x)dx := \frac{1}{\int_D \rho^2(x)dx} \int_D f(x)\rho^2(x)dx$, we define

$$B_S(u) = \int_D |u(x) - \text{mean}_{\rho^2}(u)|\rho^2(x) dx \quad \text{and} \quad B_N(u) = \min_{c \in \mathbb{R}} \int_D |u(x) - c|\rho^2(x) dx, \quad (3.8)$$

where $\text{mean}_{\rho^2}(u) = \int_D u(x)\rho^2(x)dx$.

We have the desired relation with balance terms defined in (1.2) and (1.3)

$$B(\mathbf{1}_A) = \text{Bal}(A, A^c) \quad \text{for } I = R, C, S, \text{ and } N \quad (3.9)$$

for every measurable subset A of D . From here on, we use B to represent B_R, B_C, B_S , or B_N , depending on the context. We also consider *normalized indicator functions* $\tilde{\mathbf{1}}_A$ given by

$$\tilde{\mathbf{1}}_A := \frac{\mathbf{1}_A}{B(\mathbf{1}_A)}, \quad A \subseteq D,$$

and consider the set

$$\text{Ind}(D) := \{u \in L^1(\nu) : u = \tilde{\mathbf{1}}_A \text{ for some measurable set } A \subseteq D \text{ with } B(\mathbf{1}_A) \neq 0\}. \quad (3.10)$$

Then for $u = \tilde{\mathbf{1}}_A \in \text{Ind}(D)$

$$TV(u) = TV(\tilde{\mathbf{1}}_A) = TV\left(\frac{\mathbf{1}_A}{B(\mathbf{1}_A)}\right) = \frac{TV(\mathbf{1}_A)}{B(\mathbf{1}_A)} = \text{Cut}_{\rho}(A, A^c). \quad (3.11)$$

Consequently the problem (1.5) is equivalent to minimizing $E : TL^1(D) \rightarrow (-\infty, \infty]$, given by

$$E(\mu, u) := \begin{cases} TV(u) & \text{if } \mu = \nu \text{ and } u \in \text{Ind}(D) \\ +\infty & \text{otherwise.} \end{cases} \quad (3.12)$$

where μ is a probability measure on D , $u \in L^1(\mu)$, $TV(u) = TV(u; \nu)$, is given by (3.5) and $\text{Ind}(D)$ is defined by (3.10). Since the functional E is only non-trivial when $\mu = \nu$, from now on we write $E(u)$ instead of $E(u, u)$.

Before we show that both the continuum ratio cut and Cheeger cut indeed have a minimizer, we need the following lemma:

Lemma 7 (i) *The balance functions B_I are continuous on $L^1(\nu)$.*

(ii) *The set $\text{Ind}(D)$ is closed in $L^1(\nu)$.*

Proof Let us start by proving (i). We first consider the balance term $B_C(u)$ that corresponds to the Cheeger cut. Let $u_1, u_2 \in L^1(\nu)$. Let c_i be the median of u_i for $i = 1, 2$, that is let c_i be a minimizer of $c \mapsto \int_D |u_i(x) - c|\rho(x) dx$. Then, by (3.7),

$$\begin{aligned} B_C(u_1) - B_C(u_2) &\leq \int |u_1 - c_2|\rho(x) dx - \int |u_2 - c_2|\rho(x) dx \\ &\leq \int |u_1 - u_2|\rho(x) dx = \|u_1 - u_2\|_{L^1(\nu)}. \end{aligned}$$

Exchanging the role of u_1 and u_2 in this argument implies that $|B_C(u_1) - B_C(u_2)| \leq \|u_1 - u_2\|_{L^1(\nu)}$, which implies Lipschitz continuity of B_C .

Now consider the balance term $B_R(u)$ that corresponds to the ratio cut. Let $\{u_k\}_{k=1}^\infty$ be a sequence in $L^1(\nu)$ converging to u . The inequality $||a| - |b|| \leq |a - b|$ implies that

$$\begin{aligned} &\left| \int |u_k - \text{mean}_{\rho}(u_k)|\rho(x) dx - \int |u - \text{mean}_{\rho}(u)|\rho(x) dx \right| \\ &\leq \int |u_k - u|\rho(x) dx + \int |\text{mean}_{\rho}(u_k) - \text{mean}_{\rho}(u)|\rho(x) dx \\ &\leq \int |u_k - u|\rho(x) dx + |\text{mean}_{\rho}(u_k) - \text{mean}_{\rho}(u)|. \end{aligned}$$

Since $u_k \rightarrow u$ in $L^1(\nu)$ we have that $\text{mean}_{\rho}(u_k) \rightarrow \text{mean}_{\rho}(u)$ and therefore $|B_R(u_k) - B_R(u)| \leq \|u_k - u\|_{L^1(\nu)} + |\text{mean}_{\rho}(u_k) - \text{mean}_{\rho}(u)| \rightarrow 0$ as desired.

In order to prove (ii) suppose that $\{u_k\}_{k \in \mathbb{N}}$ is a sequence in $\text{Ind}(D)$ converging in $L^1(\nu)$ to some $u \in L^1(\nu)$; we need to show that $u \in \text{Ind}(D)$. By (i) we know that $B(u_k) \rightarrow B(u)$ as $k \rightarrow \infty$. Since $u_k \in \text{Ind}(D)$, in particular $B(u_k) = 1$. Thus, $B(u) = 1$. On the other hand, $u_k \in \text{Ind}(D)$ implies that u_k has the form $u_k = \alpha_k \mathbf{1}_{A_k}$. Since this is true for every k , in particular we must have that u has the form $u = \alpha \mathbf{1}_A$ for some real number α and some measurable subset A of D . Finally, the fact that B is 1-homogeneous implies that $1 = B(u) = \alpha B(\mathbf{1}_A)$. In particular $B(\mathbf{1}_A) \neq 0$ and $\alpha = \frac{1}{B(\mathbf{1}_A)}$. Thus $u = \tilde{\mathbf{1}}_A$ with $B(\mathbf{1}_A) \neq 0$ and hence $u \in \text{Ind}(D)$. ■

Lemma 8 *Let D and ν be as stated at the beginning of this section. There exists a measurable set $A \subseteq D$ with $0 < \nu(A) < 1$ such that $\tilde{\mathbf{1}}_A$ minimizes (3.12).*

Proof The statement follows by the direct method of the calculus of variations. Since the functional is bounded from below it suffices to show that it is lower semicontinuous with respect to the $L^1(\nu)$ norm and that a minimizing sequence is precompact in $L^1(\nu)$. To show lower semi-continuity it is enough to consider a sequence $u_n = \tilde{\mathbf{1}}_{A_n} \in \text{Ind}(D)$ converging in $L^1(\nu)$ to $u \in L^1(\nu)$. From Lemma 7 it follows that $u \in \text{Ind}(D)$ and hence $u = \tilde{\mathbf{1}}_A$ for some A with $B(\mathbf{1}_A) > 0$. Therefore $\mathbf{1}_{A_n} \rightarrow \mathbf{1}_A$ as $n \rightarrow \infty$ in $L^1(\nu)$. The lower semi-continuity then follows from the lower semi-continuity of the total variation (3.6), the continuity of B and the fact that since $B(\mathbf{1}_A) > 0$, $1/B(\mathbf{1}_{A_n}) \rightarrow 1/B(\mathbf{1}_A)$ as $n \rightarrow \infty$.

The pre-compactness of any minimizing sequence of (3.12) follows directly from Theorem 5.1 of Baldi (2001), which completes the proof. ■

4. Assumptions and statements of main results.

Here we present the precise hypotheses we use and state precisely the main results of this paper. Let D be an open, bounded, connected subset of \mathbb{R}^d with Lipschitz boundary, and let $\rho : D \rightarrow \mathbb{R}$ be a continuous density which is bounded below and above by positive constants, that is, for all $x \in D$

$$\lambda \leq \rho(x) \leq \Lambda \quad (4.1)$$

for some $\Lambda \geq \lambda > 0$. We let ν be the measure $d\nu = \rho dx$. Let $\boldsymbol{\eta} : [0, \infty) \rightarrow [0, \infty)$ be the radial profile of the similarity kernel, namely the function satisfying $\eta(x) = \boldsymbol{\eta}(|x|)$. We assume

(K1) $\boldsymbol{\eta}(0) > 0$ and $\boldsymbol{\eta}$ is continuous at 0.

(K2) $\boldsymbol{\eta}$ is non-increasing.

(K3) $\sigma_\boldsymbol{\eta} := \int_{\mathbb{R}^d} \boldsymbol{\eta}(|x|) | \langle x, e_1 \rangle | dx < \infty$.

We refer to the quantity $\sigma_\boldsymbol{\eta}$ as the *surface tension* associated to $\boldsymbol{\eta}$. In the above, $\langle x, e_1 \rangle$ denotes the inner product of the vector x with the vector whose first entry is 1 and whose other entries are equal to zero. We remark that due to radial symmetry, the vector e_1 can be replaced by any unit vector in \mathbb{R}^d without changing the value of $\sigma_\boldsymbol{\eta}$.

The kernel $\eta : \mathbb{R}^d \rightarrow [0, \infty)$ is now given by $\eta(x) = \boldsymbol{\eta}(|x|)$.

These hypotheses on $\boldsymbol{\eta}$ hold for the standard similarity functions used in clustering contexts, such as the Gaussian similarity function $\boldsymbol{\eta}(r) = \exp(-r^2)$ and the proximity similarity kernel $\boldsymbol{\eta}(r) = 1$ if $r \leq 1$ and $\boldsymbol{\eta}(r) = 0$ otherwise). For a sample $\mathbf{x}_1, \dots, \mathbf{x}_n$ from the measure ν , we denote by ν_n the empirical measure associated to the sample.

The main result of our paper is:

Theorem 9 (Consistency of cuts) *Let domain D , probability measure ν , with density ρ , and kernel η satisfy the conditions above. Let ε_n denote any sequence of positive numbers converging to zero that satisfy*

$$\lim_{n \rightarrow 0} \frac{(\log n)^{3/4}}{n^{1/2}} \frac{1}{\varepsilon_n} = 0 \quad (d = 2), \quad \lim_{n \rightarrow 0} \frac{(\log n)^{1/d}}{n^{1/d}} \frac{1}{\varepsilon_n} = 0 \quad (d \geq 3).$$

Let $\{\mathbf{x}_j\}_{j \in \mathbb{N}}$ be an i.i.d. sequence of random points in D drawn from the measure ν and let $X_n = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$. Let $\mathcal{G}_n = (X_n, W_n)$ denote the graph whose edge weights are

$$w_{ij}^n := \boldsymbol{\eta} \left(\frac{|\mathbf{x}_i - \mathbf{x}_j|}{\varepsilon_n} \right) \quad 1 \leq i, j \leq n$$

where $\boldsymbol{\eta}$ satisfies assumptions (K1)-(K3). Finally, let $\{Y_n^, Y_n^{*c}\}$ denote any optimal balanced cut of \mathcal{G}_n (solution of problem (1.1)). If problem (3.12) has a unique solution $\{A^*, A^{*c}\}$, then with probability one the sequence $\{Y_n^*, Y_n^{*c}\}$ converges to $\{A^*, A^{*c}\}$ in the TL^1 -sense. If there is more than one optimal continuum balanced cut (3.12) then with probability one, $\{Y_n^*, Y_n^{*c}\}$ converges along a subsequence to an optimal continuum balanced cut.*

Additionally, with probability one, \mathcal{C}_n the minimum balanced cut of the graph \mathcal{G}_n (the minimum of (1.1)), satisfies

$$\lim_{n \rightarrow \infty} \frac{2\mathcal{C}_n}{n^2 \varepsilon_n^{d+1}} = \sigma_\boldsymbol{\eta} \mathcal{C}, \tag{4.2}$$

where $\sigma_\boldsymbol{\eta}$ is the surface tension associated to the kernel η and \mathcal{C} is the minimum of (1.5).

As the proofs for sparsest and normalized cuts are analogous, in the remainder of the paper we only treat the ratio and Cheeger cuts in detail.

Remark 10 *For simplicity of notation, from now on we make the assumption that problem (3.12) has a unique solution $\{A^*, A^{*c}\}$. In the general case, Theorem 9 follows using the same approach; the only difference is that the convergence of minimizers happens along subsequences (see Proposition 17 below).*

As we discussed in Remark 1 for $d \geq 3$ the scaling of $\varepsilon = \varepsilon_n$ on n is essentially the best possible. The proof of Theorem 9 relies on establishing a variational convergence of discrete balanced cuts to continuum balanced cuts called the Γ -convergence which we recall in Subsection 5. The proof uses the results obtained of García Trillos and Slepčev (2016), where the notion of Γ -convergence was introduced in the context of objective functionals on random data samples, and in particular the Γ -convergence of the graph total variation is considered. The Γ -convergence, together with a compactness result, provides sufficient conditions for the convergence of minimizers of a given family of functionals to the minimizers of a limiting functional.

Remark 11 *A few remarks help clarify the hypotheses and conclusions of our main result. The scaling condition $\varepsilon_n \gg (\log n)^{p_n} n^{-1/d}$ comes directly from the existence of transportation maps from Proposition 5. This means that ε_n must decay more slowly than the maximal distance a point in D has to travel to match its corresponding data point in X_n . In other words, the similarity graph \mathcal{G}_n must contain information on a larger scale than that on which the intrinsic randomness operates. Lastly, the conclusion of the theorem still holds if the partitions $\{Y_n^*, Y_n^{*c}\}$ only approximate an optimal balanced cut, that is if the energies of $\{Y_n^*, Y_n^{*c}\}$ satisfy*

$$\lim_{n \rightarrow \infty} \left(\frac{\text{Cut}(Y_n^*, Y_n^{*c})}{\text{Bal}(Y_n^*, Y_n^{*c})} - \min_{Y \subseteq X_n} \frac{\text{Cut}(Y, Y^c)}{\text{Bal}(Y, Y^c)} \right) = 0.$$

This important property follows from a general result on Γ -convergence which we recall in Proposition 17.

We also establish the following multi-class equivalent to Theorem 9.

Theorem 12 *Let domain D , measure ν , kernel η , sequence $\{\varepsilon_n\}_{n \in \mathbb{N}}$, sample points $\{\mathbf{x}_i\}_{i \in \mathbb{N}}$, and graph \mathcal{G}_n satisfy the assumptions of Theorem 9. Let $(Y_{1,n}^*, \dots, Y_{R,n}^*)$ denote any optimal balanced cut of \mathcal{G}_n , that is a minimizer of (1.4). If (A_1^*, \dots, A_R^*) is the unique optimal balanced cut of D (that is minimizer of (1.11)) then with probability one the sequence $(Y_{1,n}^*, \dots, Y_{R,n}^*)$ converges to (A_1^*, \dots, A_R^*) in the TL^1 -sense. If the optimal continuum balanced cut is not unique then the convergence to a minimizer holds along subsequences. Additionally, \mathcal{C}_n , the minimum of (1.4), satisfies*

$$\lim_{n \rightarrow \infty} \frac{2\mathcal{C}_n}{n^2 \varepsilon_n^{d+1}} = \sigma_\boldsymbol{\eta} \mathcal{C},$$

where $\sigma_\boldsymbol{\eta}$ is the surface tension associated to the kernel η and \mathcal{C} is the minimum of (1.11).

The proof of Theorem 12 involves modifying the geometric measure theoretical results of García Trillos and Slepčev (2016). This leads to a substantially longer and more technical proof than the proof of Theorem 9, but the overall spirit of the proof remains the same in the sense that the Γ -convergence plays the leading role. Finally, we remark that analogous observations to the ones presented in Remark 11 apply to Theorem 12.

5. Background on Γ -convergence

We recall and discuss the notion of Γ -convergence. The usual Γ -convergence is defined for deterministic functionals. It extends to the random functionals that we consider in a natural way. Namely for almost every realization of the random event (in our case a sequence of random points in the domain) we require the Γ -convergence of resulting, deterministic, functionals. Such notion of Γ convergence has been used by Dirr and Ohtani (2009). We now define it precisely.

Let (X, d_X) be a metric space and let $(\Omega, \mathfrak{F}, \mathbb{P})$ be a probability space. Let $F_n : X \times \Omega \rightarrow [0, \infty]$ be a sequence of random functionals. For brevity, instead of writing $F_n(x, \omega)$ we simply write $F_n(x)$ with understanding that an element $\omega \in \Omega$ has been fixed.

Definition 13 *The sequence of random functionals $\{F_n\}_{n \in \mathbb{N}}$ Γ -converges with respect to metric d_X to the deterministic functional $F : X \rightarrow [0, \infty]$ as $n \rightarrow \infty$ if for \mathbb{P} -almost every ω , the following conditions hold simultaneously:*

1. *Liminf inequality:* For every $x \in X$ and every sequence $\{x_n\}_{n \in \mathbb{N}}$ converging to x ,

$$\liminf_{n \rightarrow \infty} F_n(x_n) \geq F(x).$$

2. *Limsup inequality:* For every $x \in X$ there exists a sequence $\{x_n\}_{n \in \mathbb{N}}$ converging to x satisfying

$$\limsup_{n \rightarrow \infty} F_n(x_n) \leq F(x).$$

We say that F is the Γ -limit of the sequence of functionals $\{F_n\}_{n \in \mathbb{N}}$ (with respect to the metric d_X).

Remark 14 *In most situations one does not prove the limsup inequality for all $x \in X$ directly. Instead, one proves the inequality for all x in a dense subset X' of X where it is somewhat easier to prove, and then deduce from this that the inequality holds for all $x \in X$. To be more precise, suppose that the limsup inequality is true for every x in a subset X' of X and the set X' is such that for every $x \in X$ there exists a sequence $\{x_k\}_{k \in \mathbb{N}}$ in X' converging to x and such that $F(x_k) \rightarrow F(x)$ as $k \rightarrow \infty$, then the limsup inequality is true for every $x \in X$. The proof of the claim is straightforward, using, for example Theorem 1.17(iii) of Braides (2002). This property is not related to the randomness of the functionals in any way.*

Definition 15 *We say that the sequence of nonnegative random functionals $\{F_n\}_{n \in \mathbb{N}}$ satisfies the compactness property if for \mathbb{P} -almost every ω , the following statement holds: any sequence $\{x_n\}_{n \in \mathbb{N}}$ bounded in X and for which*

$$\limsup_{n \rightarrow \infty} F_n(x_n) < +\infty,$$

is relatively compact in X .

Remark 16 *The boundedness assumption of $\{x_n\}_{n \in \mathbb{N}}$ in the previous definition is a necessary condition for relative compactness and so it is not restrictive.*

The notion of Γ -convergence is particularly useful when the functionals $\{F_n\}_{n \in \mathbb{N}}$ satisfy the compactness property. This is because it guarantees that with \mathbb{P} -probability one, minimizers (or approximate minimizers) of F_n converge to minimizers of F and it also guarantees convergence of the minimum energy of F_n to the minimum energy of F (this statement is made precise in the next proposition). This is the reason why Γ -convergence is said to be a variational type of convergence. The next proposition can be found in (Braides, 2002; Dal Maso, 1993) in the deterministic setting. We present its proof in this random setting for completeness and for the benefit of the reader. We also want to highlight the way this type of convergence works as ultimately this is one of the essential tools used to prove the main theorems of this paper.

Proposition 17 *Let $F_n : X \times \Omega \rightarrow [0, \infty]$ be a sequence of random nonnegative functionals which are not identically equal to $+\infty$, satisfying the compactness property and Γ -converging to the deterministic functional $F : X \rightarrow [0, \infty]$ which is not identically equal to $+\infty$. Suppose that for \mathbb{P} -almost every ω there is a bounded sequence $\{x_n\}_{n \in \mathbb{N}}$ (which may depend on ω) satisfying*

$$\lim_{n \rightarrow \infty} \left(F_n(x_n) - \inf_{x \in X} F_n(x) \right) = 0. \tag{5.1}$$

Then, with \mathbb{P} -probability one,

$$\lim_{n \rightarrow \infty} \inf_{x \in X} F_n(x) = \min_{x \in X} F(x), \tag{5.2}$$

every bounded sequence $\{x_n\}_{n \in \mathbb{N}}$ in X satisfying (5.1) is relatively compact, and each of its cluster points is a minimizer of F . In particular, if F has a unique minimizer, a bounded sequence $\{x_n\}_{n \in \mathbb{N}}$ satisfying (5.1) converges to the unique minimizer of F .

Proof Consider Ω' a set with \mathbb{P} -probability one for which all the statements in the definition of Γ -convergence together with the statement of the compactness property hold. We also assume that for every $\omega \in \Omega'$, there exists a bounded sequence $\{x_n\}_{n \in \mathbb{N}}$ satisfying (5.1). We fix such $\omega \in \Omega'$.

Let $\{x_n\}_{n \in \mathbb{N}}$ be a sequence as the one described above. Let $\tilde{x} \in X$ be arbitrary. By the limsup inequality we know that there exists a sequence $\{\tilde{x}_n\}_{n \in \mathbb{N}}$ with $\tilde{x}_n \rightarrow \tilde{x}$ and such that

$$\limsup_{n \rightarrow \infty} F_n(\tilde{x}_n) \leq F(\tilde{x}).$$

By 5.1 we deduce that

$$\limsup_{n \rightarrow \infty} F_n(x_n) = \limsup_{n \rightarrow \infty} \inf_{x \in X} F_n(x) \leq \limsup_{n \rightarrow \infty} F_n(\tilde{x}_n) \leq F(\tilde{x}), \tag{5.3}$$

and since \tilde{x} was arbitrary we conclude that

$$\limsup_{n \rightarrow \infty} F_n(x_n) \leq \inf_{x \in X} F(x). \tag{5.4}$$

The fact that F is not identically equal to $+\infty$ implies that the term on the right hand side of the previous expression is finite and thus $\limsup_{n \rightarrow \infty} F_n(x_n) < +\infty$. Since the

sequence $\{x_n\}_{n \in \mathbb{N}}$ was assumed bounded, we conclude from the compactness property for the sequence of functionals $\{F_n\}_{n \in \mathbb{N}}$ that $\{x_n\}_{n \in \mathbb{N}}$ is relatively compact.

Now let x^* be any accumulation point of the sequence $\{x_n\}_{n \in \mathbb{N}}$ (we know there exists at least one due to compactness), we want to show that x^* is a minimizer of F . Working along subsequences, we can assume without the loss of generality that $x_n \rightarrow x^*$. By the liminf inequality, we deduce that

$$\inf_{x \in X} F(x) \leq F(x^*) \leq \liminf_{n \rightarrow \infty} F(x_n). \quad (5.5)$$

The previous inequality and (5.3) imply that

$$F(x^*) \leq F(\tilde{x}),$$

where \tilde{x} is arbitrary. Thus, x^* is a minimizer of F and in particular $\inf_{x \in X} F(x) = \min_{x \in X} F(x)$. Finally, to establish (5.2) note that this follows from (5.4) and (5.5). ■

5.1 Γ -convergence of graph total variation

Of fundamental importance in obtaining our results is the Γ -convergence of the graph total variation proved in García Trillos and Slepčev (2016). Let us describe this functional and also let us state the results we use. Given a point cloud $X_n := \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \subseteq D$ where D is a domain in \mathbb{R}^d , we denote by $GTV_{n,\varepsilon_n} : TL^1(D) \rightarrow [0, \infty]$ the functional defined as follows: $GTV_{n,\varepsilon_n}(\mu, u_n) = \infty$ if $\mu \neq \nu_n$ and

$$GTV_{n,\varepsilon_n}(\nu_n, u_n) := \frac{1}{n^2 \varepsilon_n^{d+1}} \sum_{i,j=1}^n \eta \left(\frac{|\mathbf{x}_i - \mathbf{x}_j|}{\varepsilon_n} \right) |u_n(\mathbf{x}_i) - u_n(\mathbf{x}_j)|, \quad (5.6)$$

where η is a kernel satisfying conditions **(K1)**-**(K3)**. Since we consider GTV_{n,ε_n} only for $\mu = \nu_n$, from now on we only write $GTV_{n,\varepsilon_n}(u_n)$, instead of $GTV_{n,\varepsilon_n}(\nu_n, u_n)$. Using the empirical measure ν_n , we may alternatively write $GTV_{n,\varepsilon_n}(u_n)$ as

$$GTV_{n,\varepsilon_n}(u_n) = \frac{1}{\varepsilon_n^d} \iint \eta \left(\frac{|x-y|}{\varepsilon_n} \right) |u_n(x) - u_n(y)| d\nu_n(x) d\nu_n(y).$$

The connection of the functional GTV_{n,ε_n} to problem (1.1) is the following: if Y_n is a subset of X_n , then the graph total variation of the indicator function $\mathbf{1}_{Y_n}$ is equal to a rescaled version of the graph cut of Y_n , that is,

$$GTV_{n,\varepsilon_n}(\mathbf{1}_{Y_n}) = \frac{2\text{Cut}(Y_n, Y_n^c)}{n^2 \varepsilon_n^{d+1}},$$

we recall that $w_{ij} = \eta \left(\frac{|\mathbf{x}_i - \mathbf{x}_j|}{\varepsilon_n} \right)$.

Now we recall the Theorems 1.1 and 1.2 of García Trillos and Slepčev (2016).

Theorem 18 (Γ - Convergence) *Let the domain D , measure ν , kernel η , sample points $\{\mathbf{x}_i\}_{i \in \mathbb{N}}$, sequence $\{\varepsilon_n\}_{n \in \mathbb{N}}$, and graph \mathcal{G}_n satisfy the assumptions of Theorem 9. Then,*

GTV_{n,ε_n} , defined by (5.6), Γ -converge to $\sigma_\eta TV_\nu$ as $n \rightarrow \infty$ in the TL^1 sense, where σ_η is the surface tension associated to the kernel η (see condition **(K3)**) and TV_ν is the extension to $TL^1(D)$ of weighted (by ρ^2) total variation functional introduced in (3.1), defined as follows:

$$TV_\nu((u, \mu)) = \begin{cases} \sigma_\eta TV(u) & \text{if } \mu = \nu \\ +\infty & \text{else.} \end{cases}$$

Moreover, we have the following compactness result.

Theorem 19 (Compactness) *Under the hypothesis of Theorem 18, the sequence of functionals $\{GTV_{n,\varepsilon_n}\}_{n \in \mathbb{N}}$ satisfies the compactness property. Namely, for \mathbb{P} -almost every ω the following holds: if a sequence $\{u_n\}_{n \in \mathbb{N}}$ with $u_n \in L^1(\nu_n)$ satisfies*

$$\limsup_{n \in \mathbb{N}} \|u_n\|_{L^1(\nu_n)} < \infty,$$

and

$$\limsup_{n \in \mathbb{N}} GTV_{n,\varepsilon_n}(u_n) < \infty,$$

then $\{u_n\}_{n \in \mathbb{N}}$ is TL^1 -relatively compact.

To conclude this section, we present Corollary 1.3 in García Trillos and Slepčev (2016), which allows us to restrict the functionals GTV_{n,ε_n} and TV to characteristic functions of sets and still obtain Γ -convergence. Observe that the only subtle point is the limsup inequality as the liminf inequality and compactness statements are particular cases of Theorem 18 and Theorem 19.

Theorem 20 *Under the assumptions of Theorem 18, with probability one the following statement holds: for every $A \subseteq D$ measurable, there exists a sequence of sets $\{Y_n\}_{n \in \mathbb{N}}$ with $Y_n \subseteq X_n$ such that,*

$$\mathbf{1}_{Y_n} \xrightarrow{TL^1} \mathbf{1}_A$$

and

$$\limsup_{n \rightarrow \infty} GTV_{n,\varepsilon_n}(\mathbf{1}_{Y_n}) \leq \sigma_\eta TV(\mathbf{1}_A).$$

The results stated above are the main tools in order to establish our main theorems. In the next section we use them together with a careful treatment of the balance term appearing in the denominator of the Cheeger/ratio cut functional.

6. Consistency of two-way balanced cuts

Here we prove Theorem 9.

6.1 Outline of the proof

Before proving that minimal balanced cuts $\{Y_n^*, Y_n^{*c}\}$ converge to minimal continuum partitions $\{A^*, A^{*c}\}$ in the sense of Definition 6, we first pause to outline the main ideas. Rather than work directly with the graph-cut-based functional defined on the sets of vertices we work with its relaxation defined on the set of functions from the graph to reals, $L^1(\nu_n)$. The relaxed discrete functionals E_n are defined in (6.6) and the relaxed continuum one, E is defined in (3.12).

We first show, by an explicit construction in Subsection 6.2, that the rescaled indicator functions of minimal balanced cuts, $\bar{\mathbf{1}}_{Y_n}(x) := \alpha_n \mathbf{1}_{Y_n}(x)$, (for explicit coefficient α_n that we will define later),

$$u_n^* := \bar{\mathbf{1}}_{Y_n^*}(x), \quad u_n^{**} := \bar{\mathbf{1}}_{Y_n^{*c}}(x) \quad \text{minimize} \quad E_n(u_n) \quad \text{over all} \quad u_n \in L^1(\nu_n). \quad (6.1)$$

Similarly, in Subsection 3.2 we showed that the normalized indicator functions

$$u^* := \bar{\mathbf{1}}_{A^*}(x), \quad u^{**} := \bar{\mathbf{1}}_{A^{*c}}(x) \quad \text{minimize} \quad E(u) \quad \text{over all} \quad u \in L^1(\nu). \quad (6.2)$$

In Subsection 6.3 we show that the approximating functionals E_n Γ -converge to $\sigma_\eta E$ in the TL^1 -sense. In Lemma 23 we establish that u_n^* and u_n^{**} exhibit the required compactness. Thus, they must converge toward the normalized indicator functions $\bar{\mathbf{1}}_{A^*}(x)$ and $\bar{\mathbf{1}}_{A^{*c}}(x)$ up to relabeling (see Proposition 17). If $\{A^*, A^{*c}\}$ is the unique minimizer, the convergence of the sequences (up to relabeling) $\{u_n^*\}, \{u_n^{**}\}$ follows. The convergence of the partition $\{Y_n^*, Y_n^{*c}\}$ toward the partition $\{A^*, A^{*c}\}$ in the sense of Definition 6 is a direct consequence. The convergence (4.2) follows from (5.2) in Proposition 17.

6.2 Functional description of discrete cuts

We introduce functionals that describe the discrete ratio and Cheeger cuts in terms of functions on X_n , rather than in terms of subsets of X_n . This mirrors the description of continuum partitions provided in Subsection 3.2. For $u_n \in L^1(\nu_n)$, we start by defining

$$B_n^R(u_n) := \frac{1}{n} \sum_{i=1}^n |u_n(\mathbf{x}_i) - \text{mean}_n(u_n)| \quad \text{and} \quad B_n^C(u_n) := \min_{c \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^n |u_n(\mathbf{x}_i) - c|. \quad (6.3)$$

Here $\text{mean}_n(u_n) = \frac{1}{n} \sum_{i=1}^n u_n(\mathbf{x}_i)$. A straightforward computation shows that for $Y_n \subseteq X_n$

$$B_n^R(\mathbf{1}_{Y_n}) = \text{Bal}_R(Y_n, Y_n^c), \quad B_n^C(\mathbf{1}_{Y_n}) = \text{Bal}_C(Y_n, Y_n^c). \quad (6.4)$$

From here on we write B_n to represent either B_n^R or B_n^C depending on the context.

Instead of defining $E_n(u_n)$ simply as the ratio $GTY_{n,\varepsilon_n}(u_n)/B_n(u_n)$, which is the direct analogue of (1.1), it proves easier to work with suitably normalized indicator functions. Given $Y_n \subseteq X_n$ with $B_n(\mathbf{1}_{Y_n}) \neq 0$, the *normalized indicator function* $\mathbf{1}_{Y_n}(x)$ is defined by

$$\bar{\mathbf{1}}_{Y_n}(x) = \mathbf{1}_{Y_n}(x)/B_n^C(\mathbf{1}_{Y_n}) \quad \text{or} \quad \bar{\mathbf{1}}_{Y_n}(x) = \mathbf{1}_{Y_n}(x)/B_n^R(\mathbf{1}_{Y_n}).$$

Note that $B_n(\bar{\mathbf{1}}_A) = 1$. We also restrict the minimization of $E_n(u)$ to the set

$$\text{Ind}_n(D) := \{u_n \in L^1(\nu_n) : u_n = \bar{\mathbf{1}}_{Y_n} \text{ for some } Y_n \subseteq X_n \text{ with } B_n(\mathbf{1}_{Y_n}) \neq 0\}. \quad (6.5)$$

Now, suppose that $u_n \in \text{Ind}_n(D)$, in other words that $u_n = \bar{\mathbf{1}}_{Y_n}$, for some set Y_n with $B_n(\mathbf{1}_{Y_n}) > 0$. Using (3.9) together with the fact that GTY_{n,ε_n} (defined in (5.6)) is one-homogeneous implies, as in (3.11)

$$GTY_{n,\varepsilon_n}(u_n) = \frac{2}{n^{2-d+1}} \frac{\text{Cut}(Y_n, Y_n^c)}{\text{Bal}(Y_n, Y_n^c)}. \quad (6.6)$$

Thus, minimizing GTY_{n,ε_n} over all $u_n \in \text{Ind}_n(D)$ is equivalent to the balanced graph-cut problem (1.1) on the graph $\mathcal{G}_n = (X_n, W_n)$ constructed from the first n data points. We have therefore arrived at our destination, a proper reformulation of (1.1) defined over $TL^1(D)$ instead of subsets of X_n . The task is to

$$\text{Minimize} \quad E_n(\mu, u_n) := \begin{cases} GTY_{n,\varepsilon_n}(u_n) & \text{if } \mu = \nu_n \text{ and } u_n \in \text{Ind}_n(D) \\ +\infty & \text{otherwise.} \end{cases} \quad (6.7)$$

Since the measure is clear from context, from now on we write $E_n(u_n)$ for $E_n(\nu_n, u_n)$.

6.3 Γ -Convergence

Proposition 21 (Γ -Convergence) *Let domain D , measure ν , kernel η , sequence $\{\varepsilon_n\}_{n \in \mathbb{N}}$, sample points $\{\mathbf{x}_i\}_{i \in \mathbb{N}}$, and graph \mathcal{G}_n satisfy the assumptions of Theorem 9. Let E_n be as defined in (6.7) and E as in (3.12). Then*

$$E_n \xrightarrow{\Gamma} \sigma_\eta E \quad \text{with respect to } TL^1 \text{ metric as } n \rightarrow \infty$$

where σ_η is the surface tension defined in assumption (K3). This is implied by the following:

1. For any $u \in L^1(\nu)$ and any sequence $\{u_n\}_{n \in \mathbb{N}}$ with $u_n \in L^1(\nu_n)$ that converges to u in TL^1 ,

$$\sigma_\eta E(u) \leq \liminf_{n \rightarrow \infty} E_n(u_n). \quad (6.8)$$
2. For any $u \in L^1(\nu)$ there exists at least one sequence $\{u_n\}_{n \in \mathbb{N}}$ with $u_n \in L^1(\nu_n)$ which converges to u in TL^1 and also satisfies

$$\limsup_{n \rightarrow \infty} E_n(u_n) \leq \sigma_\eta E(u). \quad (6.9)$$

We leverage Theorem 18 to prove this claim. We first need a preliminary lemma which allows us to handle the presence of the additional balance terms in (6.7) and (3.12).

Lemma 22 *With probability one, the following hold:*

- (i) If $\{u_n\}_{n \in \mathbb{N}}$ is a sequence with $u_n \in L^1(\nu_n)$ and $u_n \xrightarrow{TL^1} u$ for some $u \in L^1(\nu)$, then $B_n(u_n) \rightarrow B(u)$.
- (ii) If $u_n = \bar{\mathbf{1}}_{Y_n}$, where $Y_n \subseteq X_n$, converges to $u = \bar{\mathbf{1}}_A$ in the TL^1 -sense, then $\mathbf{1}_{Y_n}$ converges to $\mathbf{1}_A$ in the TL^1 -sense.

Proof To prove (i), suppose that $u_n \in L^1(\nu_n)$ and that $u_n \xrightarrow{TL^1} u$. Let us consider $\{T_n\}_{n \in \mathbb{N}}$ a stagnating sequence of transportation maps between ν and $\{\nu_n\}_{n \in \mathbb{N}}$ (one such sequence exists with probability one by Proposition 5). Then, we have $u_n \circ T_n \xrightarrow{L^1(\nu)} u$ and thus by Lemma 7, we have that $B(u_n \circ T_n) \rightarrow B(u)$. To conclude the proof we notice that $B(u_n \circ T_n) = B_n(u_n)$ for every n . Indeed, by the change of variables (2.2) we have that for every $c \in \mathbb{R}$

$$\int_D |u_n(x) - c| d\nu_n(x) = \int_D |u_n \circ T_n(x) - c| d\nu(x). \quad (6.10)$$

In particular we have $B_C^n(u_n) = B_C(u_n \circ T_n)$. Applying the change of variables (2.2), we obtain $\text{mean}_n(u_n) = \text{mean}_\nu(u_n \circ T_n)$ and combining with (6.10) we deduce that $B_R^n(u_n) = B_R(u_n \circ T_n)$. ■

The proof of (ii) is straightforward.

Now we turn to the proof of Proposition 21.

Proof Liminf inequality. For arbitrary $u \in L^1(\nu)$ and arbitrary sequence $\{u_n\}_{n \in \mathbb{N}}$ with $u_n \in L^1(\nu_n)$ and with $u_n \xrightarrow{TL^1} u$, we need to show that

$$\liminf_{n \rightarrow \infty} E_n(u_n) \geq \sigma_\eta E(u).$$

First assume that $u \in \text{Ind}(D)$. In particular $E(u) = TV(u)$. Now, note that working along a subsequence we can assume that the liminf is actually a limit and that this limit is finite (otherwise the inequality would be trivially satisfied). This implies that for all n large enough we have $E_n(u_n) < +\infty$, which in particular implies that $E_n(u_n) = GTV_{n,\varepsilon_n}(u_n)$. Theorem 18 then implies that

$$\liminf_{n \rightarrow \infty} E_n(u_n) = \liminf_{n \rightarrow \infty} GTV_{n,\varepsilon_n}(u_n) \geq \sigma_\eta TV(u) = \sigma_\eta E(u).$$

Now let us assume that $u \notin \text{Ind}(D)$. Let us consider a stagnating sequence of transportation maps $\{T_n\}_{n \in \mathbb{N}}$ between $\{\nu_n\}_{n \in \mathbb{N}}$ and ν . Since $u_n \xrightarrow{TL^1} u$ then $u_n \circ T_n \xrightarrow{L^1(\nu)} u$. By Lemma 7, the set $\text{Ind}(D)$ is a closed subset of $L^1(\nu)$. We conclude that $u_n \circ T_n \notin \text{Ind}(D)$ for all large enough n . From the proof of Lemma 22 we know that $B_n(u_n) = B(u_n \circ T_n)$ and from this fact, it is straightforward to show that $u_n \circ T_n \notin \text{Ind}(D)$ if and only if $u_n \notin \text{Ind}_n(D)$. Hence, $u_n \notin \text{Ind}_n(D)$ for all large enough n and in particular $\liminf_{n \in \mathbb{N}} E_n(u_n) = +\infty$ which implies that the desired inequality holds in this case.

Limsup inequality. We now consider $u \in L^1(\nu)$. We want to show that there exists a sequence $\{u_n\}_{n \in \mathbb{N}}$ with $u_n \in L^1(\nu_n)$ such that

$$\limsup_{n \rightarrow \infty} E_n(u_n) \leq \sigma_\eta E(u).$$

Let us start by assuming that $u \notin \text{Ind}(D)$. In this case $E(u) = +\infty$. From Theorem 18 we know there exists at least one sequence $\{u_n\}_{n \in \mathbb{N}}$ with $u_n \in L^1(\nu_n)$ such that $u_n \xrightarrow{TL^1} u$. Since $E(u) = +\infty$, the inequality is trivially satisfied in this case.

On the other hand, if $u \in \text{Ind}(D)$, we know that $u = \bar{\mathbf{1}}_A$ for some measurable subset A of D with $B(\mathbf{1}_A) \neq 0$. By Theorem 20, there exists a sequence $\{Y_n\}_{n \in \mathbb{N}}$ with $Y_n \subseteq X_n$, satisfying $\mathbf{1}_{Y_n} \xrightarrow{TL^1} \mathbf{1}_A$ and

$$\limsup_{n \rightarrow \infty} GTV_{n,\varepsilon_n}(\mathbf{1}_{Y_n}) \leq \sigma_\eta TV(\mathbf{1}_A). \quad (6.11)$$

Since $\mathbf{1}_{Y_n} \xrightarrow{TL^1} \mathbf{1}_A$ Lemma 22 implies that

$$B_n(\mathbf{1}_{Y_n}) \rightarrow B(\mathbf{1}_A). \quad (6.12)$$

In particular $B_n(\mathbf{1}_{Y_n}) \neq 0$ for all n large enough, and thus we can consider the function $u_n := \bar{\mathbf{1}}_{Y_n} \in \text{Ind}_n(D)$. From (6.12) it follows that $u_n \xrightarrow{TL^1} u$ and together with (6.11) it follows that

$$\limsup_{n \rightarrow \infty} GTV_{n,\varepsilon_n}(u_n) = \limsup_{n \rightarrow \infty} \frac{1}{B_n(Y_n)} GTV_{n,\varepsilon_n}(\mathbf{1}_{Y_n}) \leq \frac{1}{B(\mathbf{1}_A)} \sigma_\eta TV(\mathbf{1}_A) = \sigma_\eta TV(u)$$

Since, $u_n \in \text{Ind}_n(D)$ for all n large enough, in particular we have $GTV_{n,\varepsilon_n}(\mathbf{1}_{Y_n}) = E_n(\mathbf{1}_{Y_n})$ and also since $u \in \text{Ind}(D)$, we have $E(u) = TV(u)$. These facts together with the previous chain of inequalities imply the result. ■

6.4 Compactness

Lemma 23 (Compactness) *With probability one the following statement holds: Any sequence $\{u_n\}_{n \in \mathbb{N}}$ with*

$$\limsup_{n \rightarrow \infty} E_n(u_n) < +\infty$$

is precompact in TL^1 . In particular, any sequence $\{u_n^\}_{n \geq 1}$, of minimizers of E_n (defined in (6.1) and (6.2)) are precompact in the TL^1 -sense.*

Proof Let u_n denote a sequence satisfying

$$\limsup_{n \rightarrow \infty} E_n(u_n) < +\infty.$$

To show that any subsequence of u_n has a convergent subsequence it suffices to show that both

$$\limsup_{n \rightarrow \infty} GTV_{n,\varepsilon_n}(u_n) < +\infty \quad (6.13)$$

$$\limsup_{n \rightarrow \infty} \|u_n\|_{L^1(\nu_n)} < +\infty \quad (6.14)$$

hold due to Theorem 19. Since the result is about asymptotic behavior, we can assume without loss of generality that $\sup_{n \in \mathbb{N}} E_n(u_n) < +\infty$. Inequality (6.13) follows from the fact that $E_n(u_n) = GTV_{n,\varepsilon_n}(u_n)$. Note that $E_n(u_n) < \infty$ in particular implies that u_n has the form $u_n = \frac{\mathbf{1}_{Y_n}}{B_n(Y_n)}$ for some $Y_n \subseteq X_n$.

To show (6.14), consider first the balance term that corresponds to the Cheeger cut. Define a sequence v_n as follows. Set $v_n := u_n$ if $|Y_n^c| \leq |Y_n^c|$ and $v_n = \frac{1_{Y_n^c}}{B_n(Y_n^c)}$ otherwise. It then follows that

$$\|v_n\|_{L^1(u_n)} = \frac{\min\{|Y_n|, |Y_n^c|\}}{\min\{|Y_n^c|, |Y_n^c|\}} = 1.$$

Also, note that $GT_{V_{n,\varepsilon_n}}(v_n) = GT_{V_{n,\varepsilon_n}}(u_n)$. Thus (6.13) and (6.14) hold for v_n , so that any subsequence of v_n has a convergent subsequence in the TL^1 -sense. Let $v_n \xrightarrow{TL^1} v$ denote a convergent subsequence. Thus, it follows from Proposition 21, that

$$\sigma_\eta E(v) \leq \liminf_{k \rightarrow \infty} E_{n_k}(v_{n_k}) < \infty,$$

and in particular v is a normalized characteristic function, that is, $v = \mathbf{1}_A/B(\mathbf{1}_A)$ for some $A \subseteq D$ with $B(\mathbf{1}_A) \neq 0$. Since $B_{n_k}(\mathbf{1}_{Y_{n_k}^c}) = B_{n_k}(\mathbf{1}_{Y_{n_k}^c})$, $v_{n_k} \xrightarrow{TL^1} v$ implies that

$$\frac{B_{n_k}(Y_{n_k})}{1} \rightarrow \frac{B(A)}{1}.$$

Therefore, for large enough k we have

$$\|u_{n_k}\|_{L^1(v_{n_k})} \leq \frac{1}{B_{n_k}(Y_{n_k})} \leq \frac{2}{B(A)}$$

We conclude that $\|u_{n_k}\|_{L^1(v_{n_k})}$ remains bounded in L^1 , so that it satisfies (6.14) and (6.13) simultaneously. This yields compactness in the Cheeger cut case.

Now consider the balance term $B(u) = B_n(u)$ that corresponds to the ratio cut. Define a sequence $v_n := u_n - \text{mean}_n(u_n)$, and note that $GT_{V_{n,\varepsilon_n}}(v_n) = GT_{V_{n,\varepsilon_n}}(u_n)$ since the total variation is invariant with respect to translation. It then follows that

$$\|v_n\|_{L^1(v)} = \int_D |u_n(x) - \text{mean}_n(u_n)| \rho(x) dx = B(u_n) = 1.$$

Thus the sequence $\{v_n\}_{n \in \mathbb{N}}$ is precompact in TL^1 . Let $v_n \xrightarrow{TL^1} v$ denote a convergent subsequence. Using a stagnating sequence of transportation maps $\{T_{n_k}\}_{k \in \mathbb{N}}$ between ν and the sequence of measures $\{\nu_{n_k}\}_{k \in \mathbb{N}}$, we have that $v_{n_k} \circ T_{n_k} \xrightarrow{L^1(\nu)} v$. By passing to a further subsequence if necessary, we may assume that $v_{n_k} \circ T_{n_k}(x) \rightarrow v(x)$ for ν -almost every x in D .

For any such x , we have that either $T_{n_k}(x) \in Y_{n_k}$ or $T_{n_k}(x) \in Y_{n_k}^c$ so that either

$$v_{n_k} \circ T_{n_k}(x) = \frac{1}{2|Y_{n_k}^c|} \quad \text{or} \quad v_{n_k} \circ T_{n_k}(x) = -\frac{1}{2|Y_{n_k}^c|}.$$

Now, by continuity of the balance term, we have

$$B(v) = \lim_{k \rightarrow \infty} B_{n_k}(v_{n_k}) = 1,$$

and also

$$\text{mean}_\rho(v) = \lim_{k \rightarrow \infty} \text{mean}_{n_k}(v_{n_k}) = 0.$$

In particular the ν -measure of the region in which v is positive is strictly greater than zero, and likewise the ν -measure of the region in which v is negative is strictly greater than zero. It follows that both $|Y_{n_k}^c|$ and $|Y_{n_k}^c|$ remain bounded away from zero for all k sufficiently large. As a consequence, the fact that

$$\|u_{n_k}\|_{L^1(v_{n_k})} = \frac{1}{2|Y_{n_k}^c|},$$

implies that both (6.13) and (6.14) hold along a subsequence, yielding the desired compactness. \blacksquare

6.5 Conclusion of the proof of Theorem 9

We may now turn to the final step of the proof. From Proposition 17, we know that any limit point of $\{u_n^*\}_{n \in \mathbb{N}}$ (in the TL^1 sense) must equal u^* or u^{**} . As a consequence, for any subsequence $u_{n_k}^*$ that converges to u^* we have that $\mathbf{1}_{Y_{n_k}^*} \xrightarrow{TL^1} \mathbf{1}_{A^*}$ by Lemma 22, while $\mathbf{1}_{Y_{n_k}^*} \xrightarrow{TL^1} \mathbf{1}_{A^{**}}$ if the subsequence converges to u^{**} instead. Moreover, in the first case we would also have $\mathbf{1}_{Y_{n_k}^{*c}} \xrightarrow{TL^1} \mathbf{1}_{A^{*c}}$ and in the second case $\mathbf{1}_{Y_{n_k}^{*c}} \xrightarrow{TL^1} \mathbf{1}_{A^{**}}$. Thus in either case we have

$$\{Y_{n_k}^{*c}, Y_{n_k}^{*c}\} \xrightarrow{TL^1} \{A^*, A^{*c}\}$$

Thus, for any subsequence of $\{Y_n^*, Y_n^{*c}\}_{n \in \mathbb{N}}$ it is possible to obtain a further subsequence converging to $\{A^*, A^{*c}\}$, and thus the full sequence converges to $\{A^*, A^{*c}\}$.

7. Consistency of multiway balanced cuts

Here we prove Theorem 12.

Just as what we did in the two-class case, the first step in the proof of Theorem 12 involves a reformulation of both the balanced graph-cut problem (1.4) and the analogous balanced domain-cut problem (1.11) as equivalent minimizations defined over spaces of functions and not just spaces of partitions or sets.

We let $B_n(u_n) := \text{mean}_n(u_n)$ for $u_n \in L^1(\nu_n)$ and $B(u) := \text{mean}_\rho(u)$ for $u \in L^1(\nu)$, to be the corresponding balance terms. Given this balance terms, we let $\text{Ind}_n(D)$ and $\text{Ind}(D)$ be defined as in (6.5) and (3.10) respectively.

We can then let the sets $\mathcal{M}_n(D)$ and $\mathcal{M}(D)$ consist of those collections $I = (u_1, \dots, u_R)$ comprised of exactly R disjoint, normalized indicator functions that cover D . The sets $\mathcal{M}_n(D)$ and $\mathcal{M}(D)$ are the multi-class analogues of $\text{Ind}_n(D)$ and $\text{Ind}(D)$ respectively.

Specifically, we let

$$\mathcal{M}_n(D) = \left\{ (u_1^n, \dots, u_R^n) : u_r^n \in \text{Ind}_n(D), \int_D u_r^n(x) u_s^n(x) d\nu_n(x) = 0 \text{ if } r \neq s, \sum_{r=1}^R u_r^n > 0 \right\} \quad (7.1)$$

$$\mathcal{M}(D) = \left\{ (u_1, \dots, u_R) : u_r \in \text{Ind}(D), \int_D u_r(x) u_s(x) d\nu(x) = 0 \text{ if } r \neq s, \sum_{r=1}^R u_r > 0 \right\}. \quad (7.2)$$

Note for example that if $\mathcal{U} = (u_1, \dots, u_R) \in \mathcal{M}(D)$, then the functions u_r are normalized indicator functions, $u_r = \mathbf{1}_{A_r}/|A_r|$ for $1 \leq r \leq R$, and the orthogonality constraints imply that $\{A_1, \dots, A_R\}$ is a collection of pairwise disjoint sets (up to Lebesgue-null sets). Additionally, the condition that $\sum_{r=1}^R u_r > 0$ holds almost everywhere implies that the sets $\{A_1, \dots, A_R\}$ cover D up to Lebesgue-null sets.

We proceed to define the functionals on the space of R -tuples of L^1 functions, namely

$$TL^1(D, R) := \{(\mu, \mathcal{U}) : \mu \in \mathcal{P}(D), \mathcal{U} = (u_1, \dots, u_R), u_i \in L^1(\mu) \text{ for } i = 1, \dots, R\}.$$

We note that convergence in $TL^1(D, R)$ is equivalent to convergence of the R components in $TL^1(D)$.

One may follow the same argument in the two-class case to conclude that the minimization

$$\text{Minimize } E_n(\mu, \mathcal{U}_n) := \begin{cases} \sum_{r=1}^R GTV_{n, \varepsilon_n}(u_r^n) & \text{if } \mu = \nu_n \text{ and } \mathcal{U}_n \in \mathcal{M}_n(D) \\ +\infty & \text{otherwise} \end{cases} \quad (7.3)$$

is equivalent to the balanced graph-cut problem (1.4), while the minimization

$$\text{Minimize } E(\mu, \mathcal{U}) := \begin{cases} \sum_{r=1}^R TV(u_r) & \text{if } \mu = \nu \text{ and } \mathcal{U} \in \mathcal{M}(D) \\ +\infty & \text{otherwise} \end{cases} \quad (7.4)$$

is equivalent to the balance domain-cut problem (1.11).

As in the two-class case we omit the first argument of E_n and E , when it is clear from context.

At this stage, the proof of Theorem 12, is completed by following the same steps as in the two-class case. In particular we want to show that E_n defined in (7.3) Γ -converges in the TL^1 -sense to $\sigma_n E$, where E is defined in (7.4).

Proposition 24 (Γ -Convergence) *Let domain D , measure ν , kernel η , sequence $\{\varepsilon_n\}_{n \in \mathbb{N}}$, sample points $\{\mathbf{x}_i\}_{i \in \mathbb{N}}$, and graph \mathcal{G}_n satisfy the assumptions of Theorem 9. Consider functionals E_n of (7.3) and E of (7.4). Then*

$$E_n \xrightarrow{\Gamma} \sigma_n E \quad \text{with respect to } TL^1(D, R) \text{ metric as } n \rightarrow \infty.$$

That is, with probability one, all of the following statements hold

1. For any $\mathcal{U} \in [L^1(\nu)]^R$ and any sequence $\mathcal{U}_n \in (L^1(\nu_n))^R$ that converges to \mathcal{U} in the TL^1 -sense,

$$E(\mathcal{U}) \leq \liminf_{n \rightarrow \infty} E_n(\mathcal{U}_n). \quad (7.5)$$

2. For any $\mathcal{U} \in [L^1(\nu)]^R$ there exists a sequence $\mathcal{U}_n \in (L^1(\nu_n))^R$ that both, converges to \mathcal{U} in the TL^1 -sense, and also satisfies

$$\limsup_{n \rightarrow \infty} E_n(\mathcal{U}_n) \leq E(\mathcal{U}). \quad (7.6)$$

The following lemma follows in a straightforward way. We omit its proof since it follows analogous arguments to the ones used in the proof of Lemma 23.

Lemma 25 (Compactness) *With probability one the following statement holds: Any sequence $\{\mathcal{U}_n\}_{n \in \mathbb{N}}$ with $\mathcal{U}_n \in [L^1(\nu_n)]^R$ satisfying*

$$\limsup_{n \rightarrow \infty} E_n(\mathcal{U}_n) < +\infty,$$

is precompact in the TL^1 -sense. In particular, any subsequence of $\{\mathcal{U}_n\}_{n \geq 1}$ of minimizers to (7.3) has a further subsequence that converges in the TL^1 -sense.

Finally, due to Proposition 24 and Lemma 25, the arguments presented in Subsections 6.1 and 6.5 can be adapted in a straightforward way to complete the proof of Theorem 12. So we focus on the proof of Proposition 24, where arguments not present in the two-class case are needed. On one hand, this is due to the presence of the orthogonality constraints in the definition of $\mathcal{M}_n(D)$ and $\mathcal{M}(D)$, and on the other hand, from a geometric measure theory perspective, due to the fact that an arbitrary partition of the domain D into more than two sets can not be approximated by smooth partitions as multiple junctions appear when more than two sets in the partition meet.

7.1 Proof of Proposition 24

The next lemma is the multiclass analogue of Lemmas 7 and 22 combined.

Lemma 26 (i) *If $\mathcal{U}_k \rightarrow \mathcal{U}$ in $(L^1(\nu))^R$ then $B(u_k^i) \rightarrow B(u_r)$ for all $1 \leq r \leq R$. (ii) The set $\mathcal{M}(D)$ is closed in $L^1(\nu)$. (iii) If $\{\mathcal{U}_n\}$ is a sequence with $\mathcal{U}_n \in (L^1(\nu_n))^R$ and $\mathcal{U}_n \xrightarrow{TL^1} \mathcal{U}$ for some $\mathcal{U} \in (L^1(\nu))^R$, then $B_n(u_n^i) \rightarrow B(u_r)$ for all $1 \leq r \leq R$. (iv) If $u_n = \mathbf{1}_{Y_n}$, where $Y_n \subset X_n$, converges to $u = \mathbf{1}_A$ in the TL^1 -sense, then $\mathbf{1}_{Y_n}$ converges to $\mathbf{1}_A$ in the TL^1 -sense.*

Proof Statements (i), (iii) follow directly from the proof of Proposition 22. Statement (iv) is exactly as in Proposition 22.

In order to prove the second statement, suppose that a sequence $\{\mathcal{U}_k\}_{k \in \mathbb{N}}$ in $\mathcal{M}(D)$ converges to some \mathcal{U} in $(L^1(\nu))^R$. We need to show that $\mathcal{U} \in \mathcal{M}(D)$. First of all note that for every $1 \leq r \leq R$, $u_k^r \xrightarrow{L^1(\nu)} u_r$. Since $u_k^i \in \text{Ind}(D)$ for every $k \in \mathbb{N}$, and since $\text{Ind}(D)$ is a closed subset of $L^1(\nu)$ (by Proposition 22), we deduce that $u_r \in \text{Ind}(D)$ for every r .

The orthogonality condition follows from Fatou's lemma. In fact, working along a subsequence we can without the loss of generality assume that for every r , $u_r^k \rightarrow u_r$ for almost every x in D . Hence, for $r \neq s$ we have

$$0 \leq \int_D u_r(x) u_s(x) dx(x) = \int_D \liminf_{k \rightarrow \infty} (u_r^k(x) u_s^k(x)) dx(x) \leq \liminf_{k \rightarrow \infty} \int_D u_r^k(x) u_s^k(x) dx(x) = 0$$

Now let us write $u_r^k = \mathbf{1}_{A_r^k}/B(\mathbf{1}_{A_r^k})$ and $u_r = \mathbf{1}_{A_r}/B(\mathbf{1}_{A_r})$. As in the proof of Proposition 22 we must have $B(\mathbf{1}_{A_r^k}) \rightarrow B(\mathbf{1}_{A_r})$, as $k \rightarrow \infty$. Thus, for almost every $x \in D$

$$\sum_{r=1}^R u_r(x) = \lim_{k \rightarrow \infty} \sum_{r=1}^R u_r^k(x) \geq \lim_{k \rightarrow \infty} \min_{r=1, \dots, R} \frac{1}{B(\mathbf{1}_{A_r^k})} = \min_{r=1, \dots, R} \frac{1}{B(\mathbf{1}_{A_r})} > 0.$$

■

Proof [of Proposition 24]

Liminf inequality. The proof of (7.5) follows the approach used in the two-class case. Let $\mathcal{U}_n \xrightarrow{TL} \mathcal{U}$ denote an arbitrary convergent sequence. As $\mathcal{M}(D)$ is closed, if $\mathcal{U} \notin \mathcal{M}(D)$ then as in the two-class case, it is easy to see that $\mathcal{U}_n \notin \mathcal{M}_n(D)$ for all n sufficiently large. The inequality (7.5) is then trivial in this case, as both sides of it are equal to infinity. Conversely, if $\mathcal{U} \in \mathcal{M}(D)$ then we may assume that $\mathcal{U}_n \in \mathcal{M}_n(D)$ for all n , since only those terms with $\mathcal{U}_n \in \mathcal{M}_n(D)$ can make the limit inferior less than infinity. In this case we easily have

$$\begin{aligned} \liminf_{n \rightarrow \infty} E_n(\mathcal{U}_n) &= \liminf_{n \rightarrow \infty} \sum_{r=1}^R GTV_{\eta_n, \varepsilon_n}(u_r^n) \geq \sum_{r=1}^R \liminf_{n \rightarrow \infty} GTV_{\eta_n, \varepsilon_n}(u_r^n) \\ &\geq \sigma_\eta \sum_{r=1}^R TV(u_r) = \sigma_\eta E(\mathcal{U}). \end{aligned}$$

The last inequality follows from Theorem 18. This establishes the first statement in Proposition 24.

Limsup inequality. We now turn to the proof of (7.6), which is significantly more involved than the two-class argument due to the presence of the orthogonality constraints. It proves useful to consider an extension of ρ to the whole \mathbb{R}^d by setting $\rho(x) = \lambda$ for $x \in \mathbb{R}^d \setminus D$. This extension is a lower semi-continuous function and has the same lower and upper bounds that the original ρ has.

Borrowing terminology from the Γ -convergence literature, we say that $\mathcal{U} \in (L^1(\rho))^R$ has a *recovery sequence* when there exists a sequence $\mathcal{U}_n \in (L^1(\rho_n))^R$ such that (7.6) holds. To show that each $\mathcal{U} \in (L^1(\rho))^R$ has a recovery sequence, we first remark that due to general properties of the Γ -convergence, it is enough to verify (7.6) for \mathcal{U} belonging to a dense subset of $\mathcal{M}(D)$ with respect to the energy E (see Remark 14). We furthermore remark that it is enough to consider $\mathcal{U} = (u_1, \dots, u_R) \in (L^1(D))^R$ for which $E(\mathcal{U}) < \infty$, as the other case is trivial. So we can consider $\mathcal{U} \in \mathcal{M}(D)$ that satisfy

$$\sum_{r=1}^R TV(u_r) < \infty.$$

Let $u_r = \mathbf{1}_{A_r}/B(\mathbf{1}_{A_r})$ and let $c_0 := \max\{B(\mathbf{1}_{A_1}), \dots, B(\mathbf{1}_{A_R})\}$ denote the size of the largest set in the collection. The fact that $E(\mathcal{U}) < \infty$ then implies that for every $s = 1, \dots, R$,

$$TV(\mathbf{1}_{A_s}) \leq c_0 TV(u_s) \leq c_0 \sum_{r=1}^R TV(u_r) < \infty,$$

so that all sets $\{A_1, \dots, A_R\}$ in the collection defining \mathcal{U} have finite perimeter. Additionally because $\mathcal{U} \in \mathcal{M}(D)$ implies that any two sets A_r, A_s with $r \neq s$ have empty intersection up to a Lebesgue-null set, we may freely assume without the loss of generality that the sets $\{A_1, \dots, A_R\}$ are mutually disjoint.

We say that a subset of \mathbb{R}^d has a *piecewise (PW) smooth boundary* if the boundary is a subset of the union of finitely many open $d-1$ -dimensional manifolds embedded in \mathbb{R}^d . We first construct a recovery sequence for \mathcal{U} , as above, whose defining sets $\{A_1, \dots, A_R\}$ are of the form $A_r = B_r \cap D$, where B_r has piecewise smooth boundary and satisfies $|D \setminus B_r|_{\rho^2}(\partial D) = 0$. We say that such \mathcal{U} is *induced by piecewise smooth sets*. We later prove that such partitions are dense among partitions of D by sets of finite perimeters. ²

Constructing a recovery sequence for \mathcal{U} induced by sets with piecewise smooth boundary. Let $Y_r^n = A_r \cap X_n$ denote the restriction of A_r to the first n data points. Now, let us consider the transportation maps $\{T_n\}_{n \in \mathbb{N}}$ from Proposition 5. We let A_n be the set for which $\mathbf{1}_{A_r^k} = \mathbf{1}_{Y_r^n} \circ T_n$.

We first notice that the fact that B_r has a piecewise smooth boundary in \mathbb{R}^d and the fact that $\mathbf{1}_{A_r^k} - \mathbf{1}_{A_r}$ is nontrivial only within the tubular neighborhood of ∂B_r of radius $\|\text{Id} - T_n\|_\infty$, imply that

$$\|\mathbf{1}_{A_r^k} - \mathbf{1}_{A_r}\|_{L^1(\rho)} \leq C_0(B_r) \|\text{Id} - T_n\|_\infty, \quad (7.7)$$

where $C_0(B_r)$ denotes some constant that depends on the set B_r . This inequality follows from the formulas for the volume of tubular neighborhoods (see Weyl (1939), page 461). In particular, note that by the change of variables (2.2) we have, $|Y_r^n| = |A_r^n| \rightarrow |A_r|$ as $n \rightarrow \infty$, so that in particular we can assume that $|Y_r^n| \neq 0$. We define $u_r^n := \mathbf{1}_{Y_r^n}/|Y_r^n|$ as the corresponding normalized indicator function. We claim that $\mathcal{U}_n := (u_1^n, \dots, u_R^n)$ furnishes the desired recovery sequence.

To see that $\mathcal{U}_n \in \mathcal{M}_n(D)$ we first note that each $u_r^n \in \text{Ind}_n(D)$ by construction. On the other hand, the fact that $\{A_1, \dots, A_R\}$ forms a partition of D implies that $\{Y_1^n, \dots, Y_R^n\}$ defines a partition of X_n . As a consequence,

$$E_n(\mathcal{U}_n) = \sum_{r=1}^R GTV_{\eta_n, \varepsilon_n}(u_r^n)$$

by definition of the E_n functionals.

Using (7.7), we can proceed as in remark 5.1 in García Trillos and Slepčev (2016). In particular, we can assume that η has the form $\eta(|z|) = a$ for $|z| < b$ and $\eta(z) = 0$ otherwise; the general case follows in a straightforward way by using an approximating procedure with

² Note that unlike in the two-class case, due to "multiple junctions" one cannot approximate a general partition by a partition with sets with smooth boundaries. This makes the construction more complicated.

kernels that are a finite sum of step functions like the one considered previously (see the proof of Theorem 1.1 in García Trillos and Slepčev (2016)).

We set $\tilde{\varepsilon}_n := \varepsilon_n + \frac{2}{n} \|\text{Id} - T_n\|_\infty$. Recall that by assumption $\|\text{Id} - T_n\|_\infty \ll \varepsilon_n$ (see the statement of Theorem 9 and Proposition 5), and thus $\tilde{\varepsilon}_n$ is a small perturbation of ε_n . Define the non-local total variation $TV_{\tilde{\varepsilon}_n}$ of an integrable function $u \in L^1(\nu)$ as

$$\widetilde{TV}_{\tilde{\varepsilon}_n}(u) := \frac{1}{\varepsilon_n^{d+1}} \int_{D \times D} \eta \left(\frac{|x-y|}{\tilde{\varepsilon}_n} \right) |u(x) - u(y)| \rho(x) \rho(y) \, dx dy.$$

Using the definition of $\tilde{\varepsilon}_n$, and the form of the kernel η , we deduce that for all $n \in \mathbb{N}$, and almost every $x, y \in D$ we have

$$\eta \left(\frac{|T_n(x) - T_n(y)|}{\varepsilon_n} \right) \leq \eta \left(\frac{|x-y|}{\tilde{\varepsilon}_n} \right).$$

This inequality an a change of variables (see 2.2) implies that

$$\frac{\varepsilon_n^{d+1}}{\tilde{\varepsilon}_n^{d+1}} GTV_{n,\varepsilon_n}(\mathbf{1}_{Y^n}) \leq \widetilde{TV}_{\tilde{\varepsilon}_n}(\mathbf{1}_{A^n}).$$

A straightforward computation shows that there exists a constant K_0 such that

$$|\widetilde{TV}_{\tilde{\varepsilon}_n}(\mathbf{1}_{A^n}) - \widetilde{TV}_{\tilde{\varepsilon}_n}(\mathbf{1}_{A_r})| \leq \frac{K_0}{\tilde{\varepsilon}_n} \|\mathbf{1}_{A^n} - \mathbf{1}_{A_r}\|_{L^1(\nu)} \leq K_0 C_0(B_r) \frac{\|\text{Id} - T_n\|_\infty}{\tilde{\varepsilon}_n}.$$

Since $\frac{\varepsilon_n}{\tilde{\varepsilon}_n} \rightarrow 1$, the previous inequalities imply that

$$\limsup_{n \in \mathbb{N}} GTV_{n,\varepsilon_n}(\mathbf{1}_{Y^n}) \leq \limsup_{n \in \mathbb{N}} \widetilde{TV}_{\tilde{\varepsilon}_n}(\mathbf{1}_{A^n}) = \limsup_{n \in \mathbb{N}} \widetilde{TV}_{\tilde{\varepsilon}_n}(\mathbf{1}_{A_r}).$$

Finally, from remark 4.3 in García Trillos and Slepčev (2016) we deduce that

$$\limsup_{n \rightarrow \infty} \widetilde{TV}_{\tilde{\varepsilon}_n}(\mathbf{1}_{A^n}) \leq \sigma_\eta TV(\mathbf{1}_{A_r}),$$

and thus we conclude that $\limsup_{n \rightarrow \infty} GTV_{n,\varepsilon_n}(\mathbf{1}_{A^n}) \leq \sigma_\eta TV(\mathbf{1}_{A_r})$. As a consequence we have

$$\limsup_{n \rightarrow \infty} GTV_{n,\varepsilon_n}(u^n) = \limsup_{n \rightarrow \infty} \frac{GTV_{n,\varepsilon_n}(\mathbf{1}_{Y^n})}{B_n(\mathbf{1}_{Y^n})} \leq \sigma_\eta \frac{TV(\mathbf{1}_{A_r})}{B(\mathbf{1}_{A_r})}$$

for each r , by continuity of the balance term. From the previous computations we conclude that $E_n(\mathcal{U}_n) \rightarrow E(\mathcal{U})$, and from (7.7), we deduce that $\mathcal{U}_n \rightarrow \mathcal{U}$ in the TL^1 -sense, so that \mathcal{U}_n does furnish the desired recovery sequence.

Density. To prove Proposition 24, we show that for any $\mathcal{U} = (\tilde{\mathbf{1}}_{A_1}, \dots, \tilde{\mathbf{1}}_{A_R})$ where each of the sets A_r has finite perimeter, there exists a sequence $\{\mathcal{U}_m = (\tilde{\mathbf{1}}_{A_1^m}, \dots, \tilde{\mathbf{1}}_{A_R^m})\}_{m \in \mathbb{N}}$ where each of the \mathcal{U}_m is induced by piecewise smooth sets, and such that for every $r \in \{1, \dots, R\}$

$$\mathbf{1}_{A_r^m} \xrightarrow{L^1(\nu)} \mathbf{1}_{A_r},$$

and

$$\lim_{m \rightarrow \infty} TV(\mathbf{1}_{A_r^m}; \nu) = TV(\mathbf{1}_{A_r}; \nu).$$

Note that in fact, by establishing the existence of such approximating sequence, it immediately follows that $\mathcal{U}_m \rightarrow \mathcal{U}$ in $(L^1(\nu))^R$ and that $\lim_{m \rightarrow \infty} E(\mathcal{U}_m) = E(\mathcal{U})$ (by continuity of the balance terms). We provide the construction of the approximating sequence $\{\mathcal{U}_m\}_{m \in \mathbb{N}}$ through the sequence of three lemmas presented below.

Lemma 27 *Let $\{A_1, \dots, A_R\}$ denote a collection of open and bounded sets with smooth boundary in \mathbb{R}^d that satisfy*

$$\mathcal{H}^{d-1}(\partial A_r \cap \partial A_s) = 0, \quad \forall r \neq s. \quad (7.8)$$

Let D denote an open and bounded set. Then there exists a permutation $\pi : \{1, \dots, R\} \rightarrow \{1, \dots, R\}$ such that

$$TV(\mathbf{1}_{A_{\pi(r)}} \cup_{s=\pi(r)+1}^R \mathbf{1}_{A_{\pi(s)}}; \nu) \leq TV(\mathbf{1}_{A_{\pi(r)}}; \nu), \quad \forall r \in \{1, \dots, R\}.$$

Proof The proof is by induction on R . **Base case:** Note that if $R = 1$ there is nothing to prove. **Inductive Step:** Suppose that the result holds when considering any $R-1$ sets as described in the statement. Let A_1, \dots, A_R be a collection of open, bounded sets with smooth boundary satisfying (7.8). By the induction hypothesis it is enough to show that we can find $r \in \{1, \dots, R\}$ such that

$$TV(\mathbf{1}_{A_r \cup_{s \neq r} A_s}; \nu) \leq TV(\mathbf{1}_{A_r}; \nu). \quad (7.9)$$

To simplify notation, denote by Γ_i the set ∂A_i and define a_{ij} as the quantity

$$a_{ij} := \int_{\Gamma_i \cap (A_j \cup_{s \neq i, k \neq j} A_k) \cap D} \rho^2(x) \, d\mathcal{H}^{d-1}(x).$$

Hypothesis (7.8) and (3.3) imply that the equality

$$TV(\mathbf{1}_{A_r \cup_{s \neq r} A_s}; \nu) = \int_{\partial(A_r \cup_{s \neq r} A_s) \cap D} \rho^2 \, d\mathcal{H}^{d-1} = \int_{\Gamma_r \cap (\cup_{s \neq r} A_s) \cap D} \rho^2 \, d\mathcal{H}^{d-1} + \sum_{s: s \neq r} a_{sr} \quad (7.10)$$

holds for every $r \in \{1, \dots, R\}$, as does the inequality

$$TV(\mathbf{1}_{A_r}; \nu) \geq \int_{\Gamma_r \cap (\cup_{s \neq r} A_s) \cap D} \rho^2(x) \, d\mathcal{H}^{d-1} + \sum_{s: s \neq r} a_{rs}. \quad (7.11)$$

If $TV(\mathbf{1}_{A_r \cup_{s \neq r} A_s}; \nu) > TV(\mathbf{1}_{A_r}; \nu)$ for every r then (7.11) and (7.10) would imply that

$$\sum_{s: s \neq r} a_{sr} > \sum_{s: s \neq r} a_{rs}, \quad \forall r,$$

which after summing over r would imply

$$\sum_{r=1}^R \sum_{s: s \neq r} a_{sr} > \sum_{r=1}^R \sum_{s: s \neq r} a_{rs} = \sum_{r=1}^R \sum_{s: s \neq r} a_{sr}.$$

This would be a contradiction. Hence there exists at least one r for which (7.9) holds. \blacksquare

Lemma 28 *Let D denote an open, bounded domain in \mathbb{R}^d with Lipschitz boundary and let (B_1, \dots, B_R) denote a collection of R bounded and mutually disjoint subsets of \mathbb{R}^d that satisfy*

$$(i) \text{TV}(\mathbf{1}_{B_i}; \mathbb{R}^d) < +\infty, \quad (ii) |D\mathbf{1}_{B_i}|_{\rho^2}(\partial D) = 0 \quad \text{and} \quad (iii) D \subseteq \bigcup_{r=1}^R B_r.$$

Then there exists a sequence of mutually disjoint sets $\{A_1^m, \dots, A_R^m\}$ with piecewise smooth boundaries which cover D and satisfy

$$\mathbf{1}_{A_r^m} \xrightarrow{L^1(\mathbb{R}^d)} \mathbf{1}_{B_r} \quad \text{and} \quad \lim_{m \rightarrow \infty} \text{TV}(\mathbf{1}_{A_r^m}; \nu) = \text{TV}(\mathbf{1}_{B_r}; \nu) \quad (7.12)$$

for all $1 \leq r \leq R$.

Proof The proof of this lemma follows very similar ideas to those used when proving that sets with smooth boundary approximate sets with finite perimeter (see Theorem 13.46 in Leoni (2009)). Since our goal is to approximate partitions of more than two sets, we need to modify the arguments slightly. We highlight the important steps in the proof and refer to Leoni (2009) and Ambrosio et al. (2000) for details.

First of all note that $\text{TV}(\mathbf{1}_{B_r}; \mathbb{R}^d)$ and $|D\mathbf{1}_{B_r}|_{\rho^2}(\partial D)$ are defined considering ρ as a function from \mathbb{R}^d into \mathbb{R} . We are using the extension considered when we introduced the weighted total variation at the beginning of subsection 3.1. Given that $\rho^2 : \mathbb{R}^d \rightarrow (0, \infty)$ is lower semi-continuous and bounded below and above by positive constants then, it belongs to the class of weights considered in Baldi (2001) where the weighted total variation is studied.

For $r = 1, \dots, R$, we consider sequences of functions $u_r^k \in C^\infty(\mathbb{R}^d, [0, 1])$ satisfying

$$u_r^k \xrightarrow{L^1(\mathbb{R}^d)} \mathbf{1}_{B_r} \quad \text{and} \quad \text{TV}(u_r^k; \nu) \rightarrow \text{TV}(\mathbf{1}_{B_r}; \nu), \quad \text{as } k \rightarrow \infty. \quad (7.13)$$

This can be achieved by using standard, radially symmetric mollifiers J_k and setting $u_r^k = J_k * \mathbf{1}_{B_r}$, where $*$ stands for convolution. The functions J_k have the form $J_k(x) = k^d J(k|x|)$, where $J : [0, \infty) \rightarrow [0, \infty)$ is a smooth, decreasing function satisfying $\int_{\mathbb{R}^d} J(|x|) dx = 1$. See Theorem 13.46 in Leoni (2009) for more details.

The (u_1^k, \dots, u_R^k) also satisfy one additional property that will prove useful: there exists a constant $\alpha > 0$ so that

$$\Sigma^k(x) := \sum_{r=1}^R u_r^k(x) = \mathbf{1}_D * J_k(x) \geq \alpha > 0 \quad \text{for all } x \in D.$$

To see this, note that the fact that D is an open and bounded set with Lipschitz boundary implies that (see Grisvard (1985), Theorem 1.2.2.2) there exists a cone $C \subseteq \mathbb{R}^d$ with non-empty interior and vertex at the origin, a family of rotations $\{R_{x,r}\}_{x \in D}$ and a number $\zeta > 0$ such that for every $x \in D$,

$$x + R_{x,r}(C \cap B(0, \zeta)) \subseteq D.$$

The isotropy of J_k implies that

$$\begin{aligned} \int_D J_k(x-y) dy &\geq \int_{x+R_{x,r}(C \cap B(0, \zeta))} J_k(x-y) dy = \int_{C \cap B(0, \zeta)} J_k(y) dy = \int_{C \cap B(0, \zeta)} J(|y|) dy \\ &\geq \int_{C \cap B(0, \zeta)} J(|y|) dy =: \alpha > 0, \end{aligned}$$

for some positive constant α . The summation $\Sigma^k(x)$ of all u_r^k therefore satisfies the pointwise estimate

$$\Sigma^k(x) := \sum_{r=1}^R u_r^k(x) = \int_{\mathbb{R}^d} J_k(x-y) \sum_{r=1}^R \mathbf{1}_{B_r}(y) dy \geq \int_D J_k(x-y) dy \geq \alpha$$

for all $x \in D$ as claimed.

Now, for $k \in \mathbb{N}$, $t \in (0, 1)$ and $r = 1, \dots, R$, we let $B_r^k(t) := \{x : u_r^k(x) > t\}$. From (7.13), Sard's lemma (see Corollary 13.45 of Leoni (2009)), the lower semi-continuity of the total variation and the coarea formula for total variation, it follows that for almost every $t \in (0, 1)$,

$$\partial B_r^k(t) \text{ is smooth } \forall k, \quad \lim_{k \rightarrow \infty} \text{TV}(\mathbf{1}_{B_r^k(t)}; \nu) = \text{TV}(\mathbf{1}_{B_r}; \nu), \quad \mathbf{1}_{B_r^k(t)} \xrightarrow{L^1(\nu)} \mathbf{1}_{B_r}, \quad (7.14)$$

for all $r = 1, \dots, R$.

Combining (7.14) with Lemma 2.95 in Ambrosio et al. (2000), we can find positive numbers t_1, \dots, t_R strictly smaller than α/R , such that for every $r = 1, \dots, R$

$$\partial B_r^k(t_r) \text{ is smooth } \forall k, \quad \lim_{k \rightarrow \infty} \text{TV}(\mathbf{1}_{B_r^k(t_r)}; \nu) = \text{TV}(\mathbf{1}_{B_r}; \nu), \quad \mathbf{1}_{B_r^k(t_r)} \xrightarrow{L^1(\nu)} \mathbf{1}_{B_r}, \quad (7.15)$$

and such that for $r \neq s$,

$$\mathcal{H}^{d-1}(\partial B_r^k(t_r) \cap \partial B_s^k(t_s)) = 0, \quad \forall k \in \mathbb{N}.$$

We let $B_r^k := B_r^k(t_r)$ for $r = 1, \dots, R$ and $k \in \mathbb{N}$. We claim that for every $k \in \mathbb{N}$, the sets B_1^k, \dots, B_R^k cover D . To see this, suppose there exists $x \in D \setminus (\bigcup_{r=1}^R B_r^k)$. This would imply that $u_r^k(x) \leq t_r$ for all r . In turn, $\Sigma^k(x) \leq \sum_{r=1}^R t_r < \alpha$, which contradicts the estimate on Σ^k obtained earlier.

For every $k \in \mathbb{N}$, we can now use the sets (B_1^k, \dots, B_R^k) as input in Lemma (27) to obtain a partition (A_1^k, \dots, A_R^k) of D , defined by

$$A_r^k := B_r^k \setminus \bigcup_{s=\pi_k^{-1}(r)+1}^R B_{\pi_k(s)}^k$$

where π_k is a permutation of $\{1, \dots, R\}$ guaranteeing that for every $r = 1, \dots, R$

$$\text{TV}(\mathbf{1}_{A_r^k}; \nu) \leq \text{TV}(\mathbf{1}_{B_r^k}; \nu). \quad (7.16)$$

Each A_r^k has a piecewise smooth boundary due to the fact that each B_r^k has a smooth boundary. The disjointness of (B_1, \dots, B_R) combines with the L^1 -convergence of $\mathbf{1}_{B_r^k}$ to $\mathbf{1}_{B_r}$ to show that $\mathbf{1}_{A_r^k} \xrightarrow{L^1(\mathbb{R}^d)} \mathbf{1}_{B_r}$ as well. Finally, the lower semi-continuity of the total variation together with (7.16) and (7.15) imply (7.12). ■

To complete the construction, and therefore to conclude the proof of Lemma 24, we need to verify the hypotheses (i) – (ii) of the previous lemma. This is the content of our final lemma.

Lemma 29 *Let D be an open bounded domain with Lipschitz boundary and let $\{A_1, \dots, A_R\}$ denote a disjoint collection of sets that satisfy*

$$A_r \subset D \quad \text{and} \quad TV(\mathbf{1}_{A_r}; \nu) < \infty.$$

Then, there exists a disjoint collection of bounded sets (B_1, \dots, B_R) that satisfy $B_r \cap D = A_r$ together with the properties

$$(i) \quad TV(\mathbf{1}_{B_r}; \mathbb{R}^d) < +\infty \quad \text{and} \quad (ii) \quad |D\mathbf{1}_{B_r}|_{\rho^2}(\partial D) = 0.$$

The proof follows from Remark 3.43 in Ambrosio et al. (2000) (which with minimal modifications applies to total variation with weight ρ^2). ■

8. Numerical Experiments

We now present numerical experiments to provide a concrete demonstration and visualization of the theoretical results developed in this paper. We conduct all of our experiments using the Cheeger cut algorithm of Bresson et al. (2012); we omit the ratio cut for the sake of brevity and to avoid redundancy. These experiments focus on elucidating when and how minimizers of the graph-based Cheeger cut problem,

$$u_n^* \in \operatorname{argmin}_{u \in L^1(\nu_n)} E_n(u) \quad \text{with} \quad B_n(u) := \min_{c \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^n |u(\mathbf{x}_i) - c|, \quad (8.1)$$

converge in the appropriate sense to a minimizer of the continuum Cheeger cut problem

$$u^* \in \operatorname{argmin}_{u \in L^1(\nu)} E(u) \quad \text{with} \quad B(u) := \min_{c \in \mathbb{R}} \int_D |u(x) - c| \, dx. \quad (8.2)$$

We always take $\rho(x) := 1/\operatorname{vol}(D)$ as the constant density. The data points $\tilde{X}_n := \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ therefore represent i.i.d. samples from the uniform distribution. We consider the following two rectangular domains

$$D_1 := (0, 1) \times (0, 4) \quad \text{and} \quad D_2 := (0, 1) \times (0, 1.5)$$

in our experiments. We may easily compute the optimal continuum Cheeger cut for these domains. The characteristic function

$$\mathbf{1}_{A_1}(x) \quad \text{for} \quad A_1 := \{(x, y) \in D_1 : y > 2\},$$

when appropriately normalized, provides a minimizer $u_1^* \in L^1(\nu)$ of the continuum Cheeger cut in the former case, while the characteristic function

$$\mathbf{1}_{A_2}(x) \quad \text{for} \quad A_2 := \{(x, y) \in D_2 : y > 0.75\}$$

analogously furnishes a minimizer $u_2^* \in L^1(\nu)$ in the latter case. Figure 2 provides an illustration of a sequence of discrete partitions, computed from the graph-based Cheeger cut problem, converging to the optimal continuum Cheeger cut.

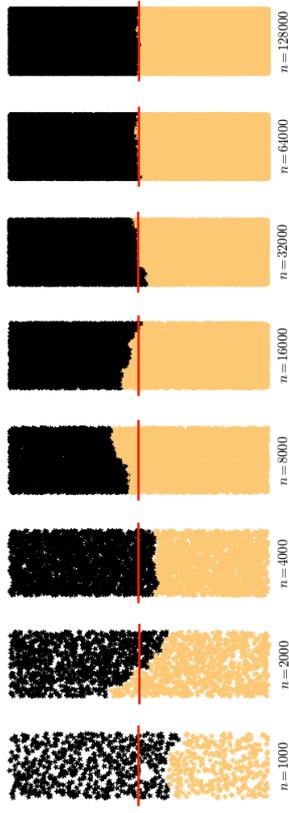


Figure 2: Visualization of the convergence process. Each figure depicts a computed optimal partition Y_n^* (in black) of one random realization of the random geometric graph $\mathcal{G}_n = (X_n, W_n)$ for each $k \in \{0, 1, \dots, 7\}$, where $n = 1000 \times 2^k$, $\varepsilon = n^{-0.3}$ and the domain considered is D_1 . Note that the scaling of ε with respect to n falls within the context of our theoretical results. The red line indicates the optimal cut, that is the boundary of the set $A_1 := \{(x, y) \in D_1 : y > 2\}$, at the continuum level.

Each of our experiments use the kernel $\eta(z) = \mathbf{1}_{\{|z| \leq 1\}}$ for the computation of the similarity weights,

$$w_{i,j} = \mathbf{1}_{\{\|\mathbf{x}_i - \mathbf{x}_j\| \leq \varepsilon_n\}},$$

so that the graphs $\mathcal{G}_n = (X_n, W_n)$ correspond to random geometric graphs (see Penrose (2003)). We use the domain D_1 only for the illustrations in Figure 2; all other experiments are conducted on the domain D_2 . We use the steepest descent algorithm of Bresson et al. (2012) to solve the graph-based Cheeger cut problem on these graphs. This algorithm relies upon a non-convex minimization, and its solutions depend upon the choice of initialization. We initialize it with the “ground-truth” partition $Y_n^i := A_i \cap X_n$ in an attempt to avoid sub-optimal solutions and to bias the algorithm towards the correct continuum cut. We terminate the algorithm once three consecutive iterates show 0% change in the corresponding partition of the graph. We let Y_n^* denote the partition of \mathcal{G}_n returned by the algorithm, which we view as the “optimal” solution of the graph-based Cheeger cut problem. Finally, we quantify the error between the optimal continuum partition $A_i \subseteq D_i$ and the n^{th} optimal graph-based partition Y_n^* of \mathcal{G}_n simply by using the percentage of misclassified data points,

$$\varepsilon_n = \min \left\{ \frac{1}{n} \sum_{i=1}^n |\mathbf{1}_{Y_n^i}(\mathbf{x}_i) - \mathbf{1}_{Y_n^*}(\mathbf{x}_i)|, \frac{1}{n} \sum_{i=1}^n |\mathbf{1}_{Y_n^i}(\mathbf{x}_i) - \mathbf{1}_{(Y_n^*)^c}(\mathbf{x}_i)| \right\}. \quad (8.3)$$

The rationale for this choice comes from the following observation. If $T_n(x)$ denotes a sequence of transportation maps between ν_n and ν that satisfy $\|\operatorname{Id} - T_n\|_\infty = o(1)$, then by

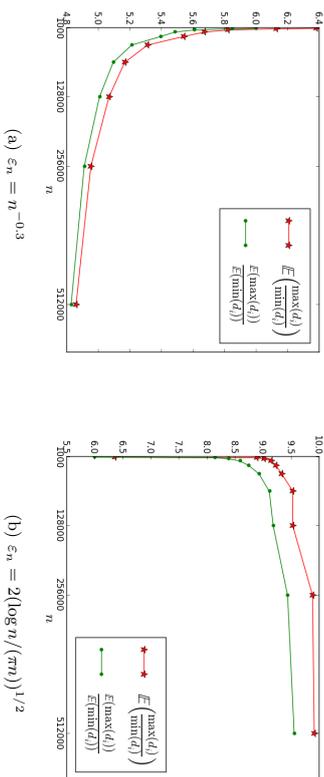


Figure 3: Graph regularity. We work with the domain D_2 . For each scaling of ε_n with n , the corresponding plot depicts two measures of regularity for the sequence of random geometric graphs. The first measure (in red) is the average $E[\max(d_i)/\min(d_i)]$, the average ratio of the maximal degree $\max(d_i)$ of G_n to the minimal degree. For each n , the average is computed over 1,440 independent graph realizations. The second measure (in green) corresponds to the ratio of the average maximal degree to the average minimal degree, computed over 1,440 independent trials as before. The graphs with $\varepsilon_n = n^{-0.3}$ become increasingly regular while the graphs with $\varepsilon_n = 2(\log n / (\pi n))^{1/2}$ become increasingly irregular.

the change of variables (2.2) (and ignoring the “min” for simplicity) we have

$$e_n = \int_D |\mathbf{1}_{A_i} \circ T_n(x) - \mathbf{1}_{Y_n^*} \circ T_n(x)| dx.$$

By the triangle inequality, we therefore obtain

$$\begin{aligned} \|\mathbf{1}_{A_i} - \mathbf{1}_{Y_n^*} \circ T_n\|_{L^1(\varphi)} &:= \int_D |\mathbf{1}_{A_i}(x) - \mathbf{1}_{Y_n^*} \circ T_n(x)| dx \\ &\leq e_n + \int_D |\mathbf{1}_{A_i}(x) - \mathbf{1}_{A_i} \circ T_n(x)| dx \leq e_n + O(\|\text{Id} - T_n\|_\infty). \end{aligned}$$

The last inequality follows since each A_i has a piecewise smooth boundary. In this way, if $\|\text{Id} - T_n\|_\infty = o(1)$ then verifying $e_n = o(1)$ suffices to show that TL^1 convergence of minimizers holds. Under the assumption that $\|\text{Id} - T_n\|_\infty = o(1)$, a similar argument shows that $e_n = o(1)$ is equivalent to TL^1 convergence. This equivalence motivates using e_n as a quantitative measure of TL^1 convergence in our experiments.

To check convergence, and to explore the issues related to Remark (2), we perform exhaustive numerical experiments for three distinct scalings of ε_n with respect to the total

number of sample points on the domain D_2 . Specifically, we consider the scalings

$$\varepsilon_n = n^{-0.3}, \quad \varepsilon_n = 2 \left(\frac{\log n}{\pi n} \right)^{1/2}, \quad \text{and} \quad \varepsilon_n = \left(\frac{\log n}{\pi n} \right)^{1/2}.$$

These scalings correspond to three distinct types of random geometric graphs. The first scaling falls well within the acceptable bounds for ε_n covered by our consistency theorems. Random graph theory shows that G_n is almost surely connected in this regime: the probability that G_n is disconnected vanishes in the $n \rightarrow \infty$ limit. The second scaling also gives rise to a sequence G_n of connected random geometric graphs for n sufficiently large (see Gupta and Kumar (1999), Penrose (2003)). However, the geometric graphs G_n exhibit rather different structural properties in this case: if $\varepsilon_n = n^{-0.3}$ then the graphs G_n become increasingly regular as $n \rightarrow \infty$, while if $\varepsilon_n = 2(\log n / (\pi n))^{1/2}$ then the graphs G_n become increasingly irregular. See Figure 3 for an illustration. The final scaling corresponds to a scaling below the connectivity threshold of random geometric graphs (see Gupta and Kumar (1999), Penrose (2003)). The graphs G_n are disconnected for large enough n under this scaling. However, in this regime each G_n has a “giant component” (a connected subgraph \mathcal{H}_n of G_n) that contains all but a small handful of vertices (see Figure 4 at right).

We designed our experiments to explore the extent to which a lack of graph-regularity or graph-connectivity might cause inconsistency of balanced cuts. The first scaling $\varepsilon_n = n^{-0.3}$ serves as a benchmark or control. It falls within the context of our consistency theorems, and so provides a means of determining the “typical” behavior of balanced cut algorithms when consistency holds. The second scaling, which falls outside the realm of our consistency results, tests whether connected graphs with different structural properties still lead to consistent results. The final scaling probes the realm where connectivity fails, but in a mild and easily correctable way. As the theory outlined above indicates, if we pose the balanced cut minimization over the full graph G_n then we can no longer expect consistency to hold. These graphs pose no practical difficulty, however, as we may simply extract the giant component \mathcal{H}_n of each G_n and then minimize the balanced cut over this connected subgraph. We simply assign each vertex in $G_n \setminus \mathcal{H}_n$ to one of the two classes uniformly at random. Our last experiment explores whether consistency might still hold using this modified approach.

Table 1 and Figure 4 report the results of these experiments. In all cases, we measure error by using the expected number of misclassified points (8.3) averaged over the number of trials indicated in Table 1. We used a smaller number of trials for large n simply due to the overwhelming computational burden. In general, we observe that sparser graphs lead to larger error (see Table 1). We caution that the corresponding rates reported in Figure 4 may not coincide with the true asymptotic rate of convergence, since we expect that as $n \rightarrow \infty$ the denser graph will still produce lower error. We furthermore remark that the measure of error we consider in Table 1 is also too weak to show convergence in the almost sure sense as provided by our consistency theorems. It does, however, indicate consistency in the weaker sense of convergence in probability (via Markov’s inequality). The algorithm we use to optimize the discrete Cheeger cut also relies upon a non-convex minimization (Bresson et al., 2012), so we cannot say with certainty that the corresponding computed optimizers are global. Instead, initializing the algorithm with the “ground truth” partition biases the

$n =$	1k	2k	4k	8k	16k	32k	64k
$\varepsilon_n = n^{-0.3}$:							
$\mathbb{E}(\varepsilon_n)$.0776	.0616	.0495	.0391	.0320	.0238	.0205
Trials	10^4	10^4	10^4	10^4	1008	1008	192
$\varepsilon_n = 2(\log n/(\pi n))^{1/2}$:							
$\mathbb{E}(\varepsilon_n)$.0710	.0603	.0509	.0427	.0366	.0303	.0256
Trials	10^4	10^4	10^4	10^4	1008	1008	192
$\varepsilon_n = (\log n/(\pi n))^{1/2}$:							
$\mathbb{E}(\varepsilon_n)$.3221	0.1984	.1216	.0883	.0672	.0528	.0424
Trials	10^4	10^4	10^4	10^4	1008	1008	192

Table 1: Average error $\mathbb{E}(\varepsilon_n)$ between partitions. For each n and each scaling of ε_n , we obtained an estimate of the error $\mathbb{E}(\varepsilon_n)$ by computing the mean of (8.3) over the indicated number of independent trials. Figure 4 provides a corresponding error plot.

algorithm toward the correct cut. If the algorithm were to fail under these circumstances, it would provide strong numerical evidence *against* consistency.

The results appear rather similar regardless of whether ε_n lies in the strongly connected ($\varepsilon_n = n^{-0.3}$), weakly connected ($\varepsilon_n = 2(\log n/(\pi n))^{1/2}$) or weakly disconnected ($\varepsilon_n = (\log n/(\pi n))^{1/2}$) regimes. Indeed, in each case the error $\mathbb{E}(\varepsilon_n)$ decays to zero with a polynomial rate. The varying degree properties of the random geometric graphs in these regimes do not seem to play much of a role. A disconnected graph, while more problematic, is not an insurmountable obstacle provided \mathcal{G}_n contains a giant component. A naive handling of the disconnected vertices still leads to plausibly consistent results. While certainly not conclusive evidence, it seems reasonable to conjecture that consistency should hold, perhaps in the weaker probabilistic sense, for ε_n as small as the critical scaling for connectivity. We leave a further exploration of this for future research.

Acknowledgments

The authors are grateful to the editor and the referees for many valuable suggestions.

They are also grateful to ICERM, where part of the research was done during the research cluster: *Geometric analysis methods for graph algorithms*. DS and NGT are grateful to NSF (grants DMS-1211760 and DMS-1516677) for its support. JvB was supported by NSF grant DMS 1312344/DMS 1521138. TL was supported by NSF (grant DMS-1414396). The authors would like to thank the Center for Nonlinear Analysis of the Carnegie Mellon University for its support.

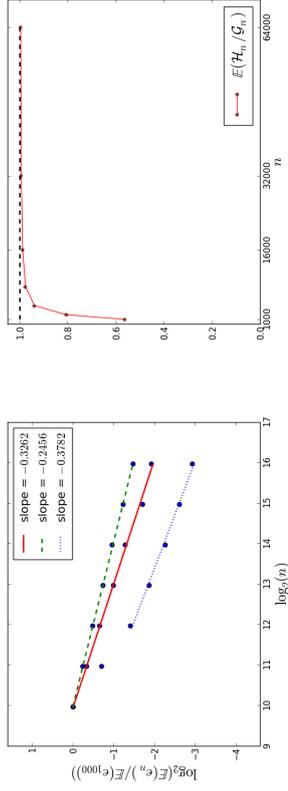


Figure 4: At left: a log-log plot of the relative expected errors $\mathbb{E}(\varepsilon_n)/\mathbb{E}(\varepsilon_{1000})$ computed in Table 1 together with a corresponding linear approximation for n large. The solid red line corresponds to the scaling $\varepsilon_n = n^{-0.3}$, the dashed green line corresponds to the scaling $\varepsilon_n = 2(\log n/(\pi n))^{1/2}$ and the dotted blue line corresponds to the scaling $\varepsilon_n = (\log n/(\pi n))^{1/2}$ of the disconnected regime. The linear approximation for the scaling $\varepsilon_n = (\log n/(\pi n))^{1/2}$ is given for those graphs \mathcal{G}_n that, in expectation, have more than 90% of vertices in the giant component. At right: the expected fraction of vertices that lie in the giant component \mathcal{H}_n of the disconnected random geometric graph \mathcal{G}_n .

References

G. Alberti and G. Bellettini. A non-local anisotropic model for phase transitions: asymptotic behaviour of rescaled energies. *European J. Appl. Math.*, 9(3):261–284, 1998. ISSN 0956-7925. doi: 10.1017/S0956792598003453. URL <http://dx.doi.org/10.1017/S0956792598003453>.

L. Ambrosio, N. Fusco, and D. Pallara. *Functions of bounded variation and free discontinuity problems*. Oxford Mathematical Monographs. The Clarendon Press, Oxford University Press, New York, 2000. ISBN 0-19-850245-1.

R. Andersen, F. Chung, and K. Lang. Local graph partitioning using pagerank vectors. In *Proceedings of the 47th Annual Symposium on Foundations of Computer Science (FOCS '06)*, pages 475–486, 2006.

E. Arias-Castro and B. Pelletier. On the convergence of maximum variance unfolding. *The Journal of Machine Learning Research*, 14(1):1747–1770, 2013.

E. Arias-Castro, B. Pelletier, and P. Pudlo. The normalized graph cut and Cheeger constant: from discrete to continuous. *Advances in Applied Probability*, 44:907–937, 2012.

- S. Arora, S. Rao, and U. Vazirani. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM (JACM)*, 56(2):5, 2009.
- A. Baldi. Weighted BV functions. *Houston J. Math.*, 27(3):683–705, 2001. ISSN 0362-1588.
- M. Belkin and P. Niyogi. Convergence of Laplacian eigenmaps. *Advances in Neural Information Processing Systems (NIPS)*, 2006.
- M. Belkin and P. Niyogi. Towards a theoretical foundation for Laplacian-based manifold methods. *J. Comput. System Sci.*, 74(8):1289–1308, 2008. ISSN 0022-0000. doi: 10.1016/j.jcss.2007.08.006. URL <http://dx.doi.org/10.1016/j.jcss.2007.08.006>.
- G. Bellettini, G. Bouchitté, and I. Fragala. BV functions with respect to a measure and relaxation of metric integral functionals. *J. Convex Anal.*, 6(2):349–366, 1999. ISSN 0944-6532.
- A. Braides. *Gamma-Convergence for Beginners*. Oxford Lecture Series in Mathematics and Its Applications, Oxford University Press, 2002.
- A. Braides and N. K. Yip. A quantitative description of mesh dependence for the discretization of singularly perturbed nonconvex problems. *SIAM J. Numer. Anal.*, 50(4):1883–1898, 2012. ISSN 0036-1429. doi: 10.1137/110822001. URL <http://dx.doi.org/10.1137/110822001>.
- X. Bresson and T. Laurent. Asymmetric Cheeger cut and application to multi-class unsupervised clustering. CAM report 12-27, UCLA, 2012.
- X. Bresson, T. Laurent, D. Uminsky, and J. von Brecht. Convergence and energy landscape for Cheeger cut clustering. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1394–1402, 2012.
- X. Bresson, T. Laurent, D. Uminsky, and J. von Brecht. Multiclass total variation clustering. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.
- A. Chambolle, A. Giacomini, and L. Lussardi. Continuous limits of discrete perimeters. *M2AN Math. Model. Numer. Anal.*, 44(2):207–230, 2010. ISSN 0764-583X. doi: 10.1051/m2an/2009044. URL <http://dx.doi.org/10.1051/m2an/2009044>.
- J. Cheeger. A Lower Bound for the Smallest Eigenvalue of the Laplacian. *Problems in Analysis*, pages 195–199, 1970.
- F. R. K. Chung. *Spectral Graph Theory*, volume 92 of *CBMS Regional Conference Series in Mathematics*. Published for the Conference Board of the Mathematical Sciences, Washington, DC, 1997.
- G. Dal Maso. *An Introduction to Γ -convergence*. Springer, 1993.
- N. Dirr and E. Orlandi. Sharp-interface limit of a Ginzburg-Landau functional with a random external field. *SIAM J. Math. Anal.*, 41(2):781–824, 2009. ISSN 0036-1410. doi: 10.1137/070684100. URL <http://dx.doi.org/10.1137/070684100>.
- R. M. Dudley. *Real analysis and probability*, volume 74 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 2002. ISBN 0-521-00754-2. doi: 10.1017/CBO9780511755347. URL <http://dx.doi.org/10.1017/CBO9780511755347>. Revised reprint of the 1989 original.
- S. Eshedoglu and F. Otto. Threshold dynamics for networks with arbitrary surface tensions. *Comm. Pure Appl. Math.*, 68(5):808–864, 2015. ISSN 0010-3640. doi: 10.1002/cpa.21527. URL <http://dx.doi.org/10.1002/cpa.21527>.
- N. García Trillos and D. Slepčev. On the rate of convergence of empirical measures in ∞ -transportation distance. *Canad. J. Math.*, 67(6):1358–1383, 2015. ISSN 0008-414X. doi: 10.4153/CJM-2014-044-6. URL <http://dx.doi.org/10.4153/CJM-2014-044-6>.
- N. García Trillos and D. Slepčev. Continuum limit of total variation on point clouds. *Arch. Ration. Mech. Anal.*, 220(1):193–241, 2016. ISSN 0003-9527. doi: 10.1007/s00205-015-0929-z. URL <http://dx.doi.org/10.1007/s00205-015-0929-z>.
- E. Giné and V. Koltchinskii. Empirical graph Laplacian approximation of Laplace-Beltrami operators: large sample results. In *High dimensional probability*, volume 51 of *IMS Lecture Notes Monogr. Ser.*, pages 238–259. Inst. Math. Statist., Beadwood, OH, 2006. doi: 10.1214/07492170600000888. URL <http://dx.doi.org/10.1214/07492170600000888>.
- M. Gobbino. Finite difference approximation of the Mumford-Shah functional. *Comm. Pure Appl. Math.*, 51(2):197–228, 1998. ISSN 0010-3640. doi: 10.1002/(SICI)1097-0312(199802)51:2<197::AID-CPA3.3.CO;2-K. URL [http://dx.doi.org/10.1002/\(SICI\)1097-0312\(199802\)51:2<197::AID-CPA3.3.CO;2-K](http://dx.doi.org/10.1002/(SICI)1097-0312(199802)51:2<197::AID-CPA3.3.CO;2-K).
- M. Gobbino and M. G. Mora. Finite-difference approximation of free-discontinuity problems. *Proc. Roy. Soc. Edinburgh Sect. A*, 131(3):567–595, 2001. ISSN 0308-2105. doi: 10.1017/S0308210500001001. URL <http://dx.doi.org/10.1017/S0308210500001001>.
- A. Goel, S. Rai, and B. Krishnamachari. Sharp thresholds for monotone properties in random geometric graphs. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 580–586, New York, 2004. ACM. doi: 10.1145/1007352.1007441. URL <http://dx.doi.org/10.1145/1007352.1007441>.
- P. Grisvard. *Elliptic problems in nonsmooth domains*, volume 24 of *Monographs and Studies in Mathematics*. Pitman (Advanced Publishing Program), Boston, MA, 1985. ISBN 0-273-08647-2.
- P. Gupta and P. R. Kumar. Critical power for asymptotic connectivity in wireless networks. In *Stochastic analysis, control, optimization and applications*, Systems Control Found. Appl., pages 547–566. Birkhäuser Boston, Boston, MA, 1999.
- L. Hagen and A. Kähng. New spectral methods for ratio cut partitioning and clustering. *IEEE Trans. Computer-Aided Design*, 11:1074–1085, 1992.
- J. Hartigan. Consistency of single linkage for high density clusters. *J. Amer. Statist. Assoc.*, 76:388–394, 1981.

- M. Hein and T. Bühler. An Inverse Power Method for Nonlinear Eigenproblems with Applications in 1-Spectral Clustering and Sparse PCA. In *Advances in Neural Information Processing Systems (NIPS)*, pages 847–855, 2010.
- M. Hein and S. Setzer. Beyond Spectral Clustering - Tight Relaxations of Balanced Graph Cuts. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.
- M. Hein, J.-Y. Audibert, and U. Von Luxburg. From graphs to manifolds—weak and strong pointwise consistency of graph Laplacians. In *Learning theory*, pages 470–485. Springer, 2005.
- R. Kannan, S. Vempala, and A. Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM (JACM)*, 51(3):497–515, 2004.
- G. Leoni. *A first course in Sobolev spaces*, volume 105 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2009. ISBN 978-0-8218-4708-8.
- M. Maier, U. von Luxburg, and M. Hein. How the result of graph clustering methods depends on the construction of the graph. *ESAIM: Probability and Statistics*, 17:370–418, 1 2013. ISSN 1262-3318. doi: 10.1051/ps/2012001. URL http://www.esaim-ps.org/article_S2810012000018.
- H. Narayanan, M. Belkin, and P. Niyogi. On the relation between low density separation, spectral clustering and graph cuts. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1025–1032, 2006.
- M. Penrose. A strong law for the longest edge of the minimal spanning tree. *Ann. Probab.*, 27(1):246–260, 1999. ISSN 0091-1798. doi: 10.1214/aop/1022677261. URL <http://dx.doi.org/10.1214/aop/1022677261>.
- M. Penrose. *Random geometric graphs*, volume 5 of *Oxford Studies in Probability*. Oxford University Press, Oxford, 2003. ISBN 0-19-850626-0. doi: 10.1093/acprof:oso/9780198506263.001.0001. URL <http://dx.doi.org/10.1093/acprof:oso/9780198506263.001.0001>.
- D. Pollard. Strong consistency of k-means clustering. *ann. statist.* 9 135–140. *Annals of Statistics*, 9:135–140, 1981.
- A. C. Ponce. A new approach to Sobolev spaces and connections to Γ -convergence. *Calc. Var. Partial Differential Equations*, 19(3):229–255, 2004. ISSN 0944-2669. doi: 10.1007/s00526-003-0195-z. URL <http://dx.doi.org/10.1007/s00526-003-0195-z>.
- O. Savin and E. Valdinoci. Γ -convergence for nonlocal phase transitions. *Ann. Inst. H. Poincaré Anal. Non Linéaire*, 29(4):479–500, 2012. ISSN 0294-1449. doi: 10.1016/j.auihpc.2012.01.006. URL <http://dx.doi.org/10.1016/j.auihpc.2012.01.006>.
- J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(8):888–905, 2000.
- A. Singer. From graph to manifold Laplacian: the convergence rate. *Appl. Comput. Harmon. Anal.*, 21(1):128–134, 2006. ISSN 1063-5203. doi: 10.1016/j.acha.2006.03.004. URL <http://dx.doi.org/10.1016/j.acha.2006.03.004>.
- D. Spielman and S. Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 81–90, 2004.
- D. Spielman and S. Teng. A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning. *SIAM Journal on Computing*, 42(1):1–26, 2013.
- A. Szlam and X. Bresson. Total variation and Cheeger cuts. In *International Conference on Machine Learning (ICML)*, pages 1039–1046, 2010.
- M. Thorpe, F. Theil, A. M. Johansen, and N. Cade. Convergence of the k -means minimization problem using Γ -convergence. *SIAM J. Appl. Math.*, 75(6):2444–2474, 2015. ISSN 0036-1399. doi: 10.1137/140974365. URL <http://dx.doi.org/10.1137/140974365>.
- D. Ting, L. Huang, and M. I. Jordan. An analysis of the convergence of graph Laplacians. In *Proceedings of the 27th International Conference on Machine Learning*, 2010.
- Y. van Gennip and A. L. Bertozzi. Γ -convergence of graph Ginzburg-Landau functionals. *Adv. Differential Equations*, 17(11-12):1115–1180, 2012. ISSN 1079-9389.
- C. Villani. *Topics in Optimal Transportation*. Graduate Studies in Mathematics. American Mathematical Society, 2003. ISBN 9780821833124. URL <http://books.google.com/books?id=q6kyE2ZkxrcC>.
- U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.
- U. von Luxburg, M. Belkin, and O. Bousquet. Consistency of spectral clustering. Technical Report TR 134, Max Planck Institute for Biological Cybernetics, 2004.
- U. von Luxburg, M. Belkin, and O. Bousquet. Consistency of spectral clustering. *The Annals of Statistics*, 36(2):555–586, 2008.
- Y.-C. Wei and C.-K. Cheng. Towards efficient hierarchical designs by ratio cut partitioning. In *Computer-Aided Design, 1989. ICCAD-89. Digest of Technical Papers, 1989 IEEE International Conference on*, pages 298–301. IEEE, 1989.
- H. Weyl. On the Volume of Tubes. *Amer. J. Math.*, 61(2):461–472, 1939. ISSN 0002-9327. doi: 10.2307/2371513. URL <http://dx.doi.org/10.2307/2371513>.
- S. X. Yu and J. Shi. Multiclass spectral clustering. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 313–319. IEEE, 2003.

Jointly Informative Feature Selection Made Tractable by Gaussian Modeling

Leonidas Lefakis*

Zalando Research

Zalando SE

Berlin, Germany

LEONIDAS.LEFAKIS@ZALANDO.DE

François Fleuret

Computer Vision and Learning group

Idiap Research Institute

Martigny, Switzerland

FRANCOIS.FLEURET@IDIAP.CH

Editor: Amos Storkey

Abstract

We address the problem of selecting groups of jointly informative, continuous, features in the context of classification and propose several novel criteria for performing this selection. The proposed class of methods is based on combining a Gaussian modeling of the feature responses with derived bounds on and approximations to their mutual information with the class label. Furthermore, specific algorithmic implementations of these criteria are presented which reduce the computational complexity of the proposed feature selection algorithms by up to two-orders of magnitude. Consequently we show that feature selection based on the joint mutual information of features and class label is in fact tractable; this runs contrary to prior works that largely depend on marginal quantities. An empirical evaluation using several types of classifiers on multiple data sets show that this class of methods outperforms state-of-the-art baselines, both in terms of speed and classification accuracy.

Keywords: feature selection, mutual information, entropy, mixture of Gaussians

1. Introduction

Given a collection of data in \mathbb{R}^D it is often advantageous to reduce its dimensionality by either extracting (Hinton and Salakhutdinov, 2006) or selecting (Guyon and Elisseeff, 2003) a subset of $d \ll D$ features, which carry “as much information as possible”. The motivation behind this dimensionality reduction can be either to control over-fitting by reducing the capacity of the classifier space or to improve the computational overhead by reducing the optimization domain. It can also be used as a tool to facilitate the understanding or the graphical representation of high-dimensional data.

In the present work we focus on the selection, rather than extraction, of features. In general feature selection methods can be divided into two large families: Techniques from the first group, *filters*, are predictor agnostic as they do not optimize the selection of features

for a specific prediction method. They are usually based on classical statistics or information theoretic tools; the novel methods we propose in this article belong to this category. Techniques from the second group, *wrappers*, choose features to optimize the performance of a certain predictor. They usually require the retraining of the predictor at each step of a greedy search. Consequently they are typically computationally more expensive than filters. Furthermore, as the selected features are tailored to a specific predictor, they often do not work well with other families of predictors.

In the following we present a filter approach to feature selection in the context of classification based on the maximization of the mutual information between the selected features and the class to predict. The use of mutual information as a criterion for feature selection has been extensively studied in the literature, and can be motivated in the context of classification by Fano’s inequality

$$H(Y|\hat{Y}) \leq 1 + P(e) \log(|\mathcal{Y}|-1)$$

where Y, \hat{Y} are the true and predicted labels respectively, while e is the event that $Y \neq \hat{Y}$. Combining the above with the data processing inequality (specifically the chain $Y \rightarrow X \rightarrow \hat{Y}$) results in the following lower bound on the probability of an incorrect classification P_E by

$$P(e) \geq \frac{H(Y) - I(X; Y) - 1}{(|\mathcal{Y}|-1)}$$

where $H(Y)$ is the entropy of the class prior and $I(X; Y)$ is the mutual information between the output (Y) and input (X) features.

An important issue that arises in this context, is that of the joint “informativeness” of the selected features. Though wrappers by their very nature tend to select features which are jointly informative, this issue is only partially addressed for filters due to the resulting computational complexity and need for very large training sets. Most existing methods (Peng et al., 2005; Fleuret, 2004; Hall and Smith, 1999; Liu and Yu, 2003) typically compromise by relying on the mutual information between individual features or very small groups of features (pairs, triplets) and the class. We argue however, that rather than compromising on the joint behavior of the selected features, it is preferable to accept a compromise on the density model, which will allow to analyze this joint behavior in an efficient manner.

In a classification task with continuous features context, if we aim at taking into account the joint behavior of features, a Gaussian model is a natural choice. This unfortunately leads to a technical difficulty: If such a model is used for the conditional distributions of the features given the class then the non-conditioned distribution is a mixture of Gaussians and its entropy has no simple analytical form. There is extensive prior work on the problem of approximating the entropy of a mixture of Gaussians (Hershey and Olsen, 2007), but most of the existing approximations are too computationally intensive to be used during an iterative optimization process which requires the estimation of the mutual information of a very large number of subsets of features with the class to predict.

We propose here an alternative approach wherein we derive both bounds on and approximations to the Mutual Information – and maximize these quantities *in lieu* of the true mutual information. We also propose specific algorithmic implementations that rely

* Part of this research was conducted while at the Computer Vision and Learning group of the Idiap Research Institute

on updating the inverses of covariance matrices iteratively instead of computing them from scratch drastically reducing the computational cost during the optimization of the selected feature set.

2. Related Works

When selecting d features from a pool of D candidates in the context of a classification task, it does not suffice to select features independently informative with respect to the class. When such greedy strategies are employed the risk of acquiring redundant, or even identical, features increases. Thus, it is also important that these features exhibit low redundancy between them: *joint* informativeness is at the core of feature selection.

As mentioned in the introduction, wrappers, due to their very nature, address this issue by creating subsets of features that perform well when combined with a specific predictor. Examples of such methods include iteratively training a *SVM*, removing at each iteration the features with the smallest weights (Guyon et al., 2002) or employing Adbost in connection with decision stumps to perform feature selection (Das, 2001). Other wrapper methods impose sparsity on the resulting predictor thus implicitly performing feature selection, for instance by using a Laplacian prior to perform sparse logistic regression (Cawley et al., 2006), casting a l_0 regularized *SVM* as a mixed integer programming problem (Tan et al., 2010), or imposing sparsity via a l_1 -norm regularizer (Argyriou et al., 2008) or l_1 -clipped norm (Xu et al., 2014). Such wrappers, that train predictors only once, tend to be much faster, however like other wrappers they tend not to generalize well across predictors. A wrapper approach that shares similarities with the algorithm proposed here, forward regression (Das and Kempe, 2011) iteratively augments a subset of features to build a linear regressor which is near-optimal in a least-squared error sense.

In the context of filters, the simplest methods are those that calculate statistics on individual features and then rank these features based on these values, keeping the d features of highest rank. Examples of such statistics are Fisher score, mutual information between the feature and the class (information gain), χ^2 etc. Though quick to compute, such approaches typically result in large feature redundancy and sub-optimal performance.

A slightly more computationally complex approach, the ReliefF algorithm (Robnik-Sikrija and Kononenko, 2003), looks at individual features assigning a score by randomly selecting samples and calculating for that feature and for each sample the difference in distance between the random sample and the nearest sample of the same class, dubbed “nearest hit”, and the random sample and the nearest sample of a different class, dubbed “nearest miss”. Despite looking at features in isolation, it has been shown to perform well in practice.

In the work on mRMR feature selection (Pang et al., 2005) the authors attempt to address the issue of redundancy by selecting feature of maximum relevance and minimum redundancy, that is features with high mutual information with the class and low mutual pairwise information with the remaining selected features, thus selecting features that are not pairwise redundant. Quadratic programming feature selection (Rodriguez-Lujan et al., 2010) casts the feature selection as an optimization task and can be used in conjunction with a number of similarity measure. When combined with mutual information it resembles

mRMR though it provides a ranking of features as opposed to the greedy selection process of mRMR.

Another approach (Vasconcelos, 2003) based on mutual information attempts to diversify the conditional distributions $p_{X|Y}$ by greedily choosing features that maximize the Kullback-Leibler divergence between the conditionals and the prior p_X . However only the marginal distributions pertaining to individual features are used and as such no joint informativeness of features is exploited.

The *FCBF* algorithm (Liu and Yu, 2003) uses symmetrical uncertainty $\frac{I(X;Y)}{I(X;Y)}$ as a quantitative criterion and adds features to a pool based on a novel concept of predominant correlation, namely that the feature is more highly correlated with the class than any of the features already in the pool. *CFS* (Hall and Smith, 1999) similarly combines symmetric uncertainty with Pearson’s correlation to add features exhibiting low correlation with the features already in the pool.

Redundancy may also be addressed via the concept of a Markov Blanket (Margaritis, 2009). The Markov blanket of a variable X is defined as the set of variables S such that X is independent of the remaining variables $D \setminus (S \cup X)$ given the values of the variables in S . Based on this concept, the authors in (Fleuret, 2004) select features that have high mutual information with the class when conditioned on one of the features already in the pool. The resulting algorithm is suitable only for binary data. Here however we explicitly address the problem of feature selection in a continuous domain.

Closely related, at least conceptually, to the work presented here is prior work (Torkkola, 2003) which similarly attempts to find features that are jointly informative by resorting to a Gaussian modeling. In that work however the aim is feature extraction and the mutual information is used as an objective to guide a gradient ascent algorithm.

Another promising line of work is that of (Song et al., 2012) which avoids density estimation necessary in mutual information based approaches by considering the Hilbert-Schmidt Independence Criterion. The authors show how to obtain unbiased estimates of the HSIC quantity. Furthermore the method can be kernelized thus allowing for the discovery of dependencies in high-(possibly infinite) dimensional feature space. The Hilbert-Schmidt Independence Criterion has also been used in conjunction with l_1 -norm regularization (Yamada et al., 2014).

Finally, we note a family of feature selection algorithms, which have become very popular in recent years, based on spectral clustering. In such approaches, features can be selected based on their influence on the affinity graph Laplacian (Jiang and Ren, 2011), or by analyzing the spectrum of the Laplacian matrix (Wolf and Shashua, 2005).

3. Feature Selection Criteria

We propose two novel criteria which characterize the informativeness of a set of features in a classification context using their mutual information with the class under a Gaussian model of the features given the class. While this approach is conceptually straight-forward, it requires the evaluation of the entropy of a mixture of Gaussians for which no closed-form expression is available.

Our first approach, dubbed “Gaussian compromise” and described in § 3.2.1, uses the entropy of a single Gaussian of same expectation and variance as the mixture to obtain an upper bound on, and subsequently an approximation to, the true entropy.

Table 1: Notation

$F = \{X_1, X_2, \dots, X_D\}$	the set of candidate features
X_j	a single feature
Y	the class label
S	a subset of F
S_{n-1}	the set of features selected up until iteration n
$\Sigma_{j,S}$	the covariance matrix of the features in S
$\Sigma_{j,S}^2$	the covariance vector of feature X_j and the features in S
$\sigma_{i,j}^2$	the covariance of features X_j and X_i
σ_i^2	the variance of feature i
Σ_y^S	the variance of the features in S conditioned on $Y = y$
$\sigma_{j S}^2$	the variance of feature X_j conditioned on the value of the features in S
$f_{j,S}$	the density of a normal approximation of the class conditional distributions
f^*	the Gaussian approximation of the joint law $\sum_y p_y f_y$
p_y	the prior on the class variable Y

Our second approach, described in § 3.2.2 is based on a decomposition of the mutual information, in the binary class case, as a sum of Kullback-Leibler divergence terms, which can be efficiently approximated. The n -class case is addressed by averaging the obtained quantity over the one-vs-all sub-problems.

3.1 Mutual Information and the Gaussian Model

Given a continuous variable X and a finite variable Y , their mutual information is defined as

$$I(X; Y) = H(X) - H(X|Y) \quad (1)$$

$$= H(X) - \sum_y H(X|Y = y)P(Y = y). \quad (2)$$

Using a Gaussian density model for continuous variables is a natural strategy, due in part to the simplicity of its parametrization, and to its ability to capture the joint behavior of its components. Moreover, the entropy of a n -dimensional multivariate Gaussian $X \sim \mathcal{N}(\mu, \Sigma)$ has a simple and direct expression, namely

$$H(X) = \frac{1}{2} \log(|\Sigma|) + \frac{n}{2} (\log 2\pi + 1).$$

3.2 Bounds on the Mutual Information and the Entropy

Estimating the mutual information as defined in equation (2) requires the estimation of the entropy of both the conditional distributions $X|Y = y$ for all y , and that of X itself. If we model the former with Gaussian distributions, the latter is a mixture of Gaussian distributions, the entropy of which has no simple analytic form.

We propose to mitigate this problem by deriving upper bounds and approximations with tractable forms. Let $f_y, y = 1, \dots, C$ denote Gaussian densities on \mathbb{R}^D , p_y a discrete distribution on $\{1, \dots, C\}$, and f^* the Gaussian approximation of the joint law

$$f = \sum_y f_y p_y,$$

that is the Gaussian density of same expectation and covariance matrix as the mixture. Let Y be a random variable of distribution p_y and X a continuous random variable with conditional distributions $\mu_{X|Y=y} = f_y$.

3.2.1 GAUSSIAN COMPROMISE CRITERION

As mentioned, we propose here to use an approximation of $H(f)$ based on the entropy of $H(f^*)$. While modeling f as a Gaussian is not consistent with the Gaussian models of the conditioned densities, estimating the mutual information with it still has all the important properties one desires for continuous feature selection:

- It captures the information content of individual features, since adding a non informative feature would change by the same amount all the terms of equation (2).
- It accounts for redundancy, since linearly dependent features would induce a small determinant of the covariance matrix and by extension small mutual information. This can be seen if we consider that the determinant of a matrix which has rows (or columns) which are linearly dependent is 0.
- It normalizes with respect to any affine transformation of the features, since such a transformation changes by the same amount all the densities in equation (2). This can be seen if we consider that translation has no effect on the covariance matrix and that a linear transformation A gives

$$H(AX) = \frac{1}{2} \log(|A\Sigma A^T|) + \frac{n}{2} (\log 2\pi + 1) = H(X) + \log(|A|).$$

However, this first-order approximation suffers from a core weakness, namely that the entropy of f^* can become arbitrarily larger than the entropy of $\sum_y p_y f_y$ (see figure 1). This leads to degenerated cases where families of features look “infinitely informative”. This effect can be mitigated by considering the following upper bound on the true entropy $H(\sum_y p_y f_y)$.

We have by definition

$$I(X; Y) = H\left(\sum_y f_y p_y\right) - \sum_y H(f_y) p_y.$$

Since f^* is a Gaussian density, it has the highest entropy for a given variance, and thus $H(f^*) \geq H(\sum_y f_y p_y)$, hence

$$I(X; Y) \leq H(f^*) - \sum_y H(f_y) p_y. \quad (3)$$

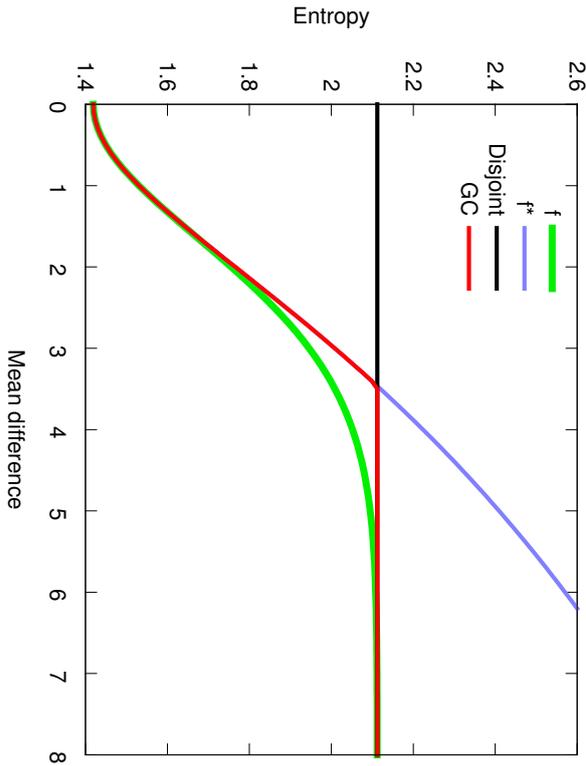


Figure 1: This graph shows several estimates of the entropy of a mixture of two IID Gaussian densities of variance 1, as a function of the difference between their means. The green curve is the true value of the entropy, estimated numerically. The blue curve is the entropy of a single Gaussian fitted on the mixture. The black line stands for the limit entropy when the two components are ‘far apart’. Finally, the red curve is the upper bound described in § 3.2.1.

The mutual information between X and Y is upper bounded by the entropy of Y as Y is discrete, hence

$$I(X; Y) \leq H(Y) = -\sum_y p_y \log p_y,$$

from which we get

$$I(X; Y) \leq \sum_y (H(f_y) - \log p_y) p_y - \sum_y H(f_y) p_y. \quad (4)$$

Taking the min of inequalities (3) and (4), we obtain the following upper bound

$$I(X; Y) \leq \min \left(H(f^*), \sum_y (H(f_y) - \log p_y) p_y \right) - \sum_y H(f_y) p_y. \quad (5)$$

Figure 1 illustrates the behavior of this bound in the case of two IID Gaussians. An upper bound on $H(X)$ follows directly:

From the concavity of the entropy function we obtain the following lower bound

$$H(X) \geq \sum_y p_y H(f_y). \quad (6)$$

Combining eq. (5), (6) gives

$$\min \left(H(f^*), \sum_y (H(f_y) - \log(p_{y_i})) p_{y_i} \right) \geq H(f) \geq \sum_y p_y H(f_y).$$

Note that the difference between upper and lower bound is itself bounded

$$-\sum_y p_y \log(p_{y_i}) \geq \min \left(H(f^*), \sum_y (H(f_y) - \log(p_{y_i})) p_{y_i} \right) - \sum_y p_y H(f_y).$$

The proposed GC criterion is based on an approximation to $H(X)$, namely

$$\tilde{H}(f) = \sum_y p_y \min(H(f^*), H(f_y) - \log(p_{y_i})).$$

We note that this approximation is also upper bounded by

$$\min \left(H(f^*), \sum_y (H(f_y) - \log(p_{y_i})) p_{y_i} \right) \geq \tilde{H}(f).$$

Furthermore, since $\forall y$

$$H(f_y) - \log(p_{y_i}) > H(f_y),$$

it follows that if $\forall y$

$$H(f^*) \geq H(f_y) \quad (7)$$

then

$$\tilde{H}(f) \geq \sum_y p_y H(f_y),$$

meaning the approximation $\tilde{H}(f)$ also lies between the two bounds and by extension

$$-\sum_y p_y \log(p_y) \geq |\tilde{H}(f) - H(f)|.$$

For (7) to hold it suffices that

$$\lambda_i^* \geq \lambda_i^y, \forall i, y \quad (8)$$

where λ_i^*, λ_i^y are the i th (sorted by magnitude) eigenvalues of Σ^* and Σ^y respectively, in which case

$$\prod_i \lambda_i^* \geq \prod_i \lambda_i^y.$$

That is to say a sufficient condition is that the variance of f^* when projected along any of the eigenvectors of the covariance matrix Σ^* is at least as large as the variance of f^y when projected along the corresponding eigenvector of $\Sigma^y, \forall y$. Alternatively, for (7) to hold it suffices that $\forall x, y$ (and in particular $\forall x$ which are eigenvectors of Σ^y)

$$x^T \Sigma^* x \geq x^T \Sigma^y x.$$

Given that

$$\Sigma^* = \sum_y p_y \Sigma^y + \sum_y p_y (\mu_y - \mu^*) (\mu_y - \mu^*)^T \quad (9)$$

this translates to

$$\sum_y p_y x^T \Sigma^y x + \sum_y p_y x^T (\mu_y - \mu^*) (\mu_y - \mu^*)^T x \geq x^T \Sigma^y x.$$

That is to say that it suffices that the variance of f_y along any direction can be accounted for either by the variances of the mixture components along this direction or by the variance of the mixture means in this direction. Note that in this case (8) also holds.

Based on the above, we use the following approximation to the mutual information to perform feature selection

$$\tilde{I}(X; Y) = \sum_y \min(H(f^*), H(f_y) - \log p_y) p_y - \sum_y H(f_y) p_y. \quad (10)$$

In figure 2 we show a comparison of the GC-approximation and the true mutual information. To compare the two we draw samples from a mixture of five Gaussians and use these samples to estimate the mutual information. Specifically, we create this mixture by sampling the expectations uniformly in $[-5, 5]$, sampling the standard deviations uniformly in $[0.001, 2.001]$ and the priors in $[0, 1]$ (which are then normalized). We observe that by taking the minimum over two sub-optimal estimators (the prior and the fitted Gaussian f^*) we obtain a very good estimator of the true mutual information.

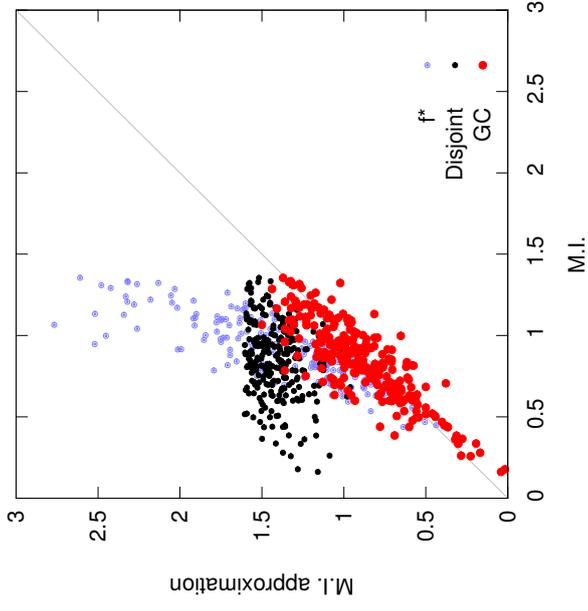


Figure 2: Comparison of the mutual information of a mixture of 5 Gaussians as estimated using the GC-approximation, using the fitted Gaussian f^* , and using the prior distribution (marked as *disjoint*), with the true mutual information calculated numerically.

3.2.2 KL-BASED BOUND

We derive here a more general bound in the case of a distribution f which is a mixture of two distributions $f = p_1 f_1 + p_2 f_2$. In this case we have:

$$H(f) = - \int_{-\infty}^{\infty} p_1 f_1(u) \log(p_1 f_1(u) + p_2 f_2(u)) \log(p_1 f_1(u) + p_2 f_2(u)) du.$$

Working with the first term in the above integral we have:

$$\begin{aligned} & - \int_{-\infty}^{\infty} p_1 f_1(u) \log(p_1 f_1(u) + p_2 f_2(u)) du \\ &= - \int_{-\infty}^{\infty} p_1 f_1(u) \log \left(1 + \frac{p_2 f_2(u)}{p_1 f_1(u)} \right) du \\ & \quad - \int_{-\infty}^{\infty} p_1 f_1(u) \log(p_1 f_1(u)) du \\ &= c - \int_{-\infty}^{\infty} p_1 f_1(u) \log \left(1 + \frac{p_2 f_2(u)}{p_1 f_1(u)} \right) du + p_1 H(f_1(u)) \\ &= p_1 D_{KL}(f_1(u) \| f_2(u)) + p_1 H(f_1(u)) + c', \end{aligned}$$

where c, c' are constants related to the mixture coefficients and the inequality comes from the fact that $\log(1+x) \geq \log(x)$. Similarly for the second term we have:

$$- \int_{-\infty}^{\infty} p_2 f_2(u) \log(p_1 f_1(u) + p_2 f_2(u)) du \leq p_2 D_{KL}(f_2(u) \| f_1(u)) + p_2 H(f_2(u)) + c''.$$

Based on this we have:

$$H(f) \leq p_2 D_{KL}(f_2(u) \| f_1(u)) + p_2 H(f_2(u)) + p_1 D_{KL}(f_1(u) \| f_2(u)) + p_1 H(f_1(u)) + c''',$$

and by extension we have the following bound on the mutual information:

$$I(X; Y) \leq p_2 D_{KL}(f_2(u) \| f_1(u)) + p_1 D_{KL}(f_1(u) \| f_2(u)) + c''''.$$

In the case where $f_1 = N(\mu_1, \Sigma_1)$ and $f_2 = N(\mu_2, \Sigma_2)$ are both multivariate Gaussian distributions of dimensionality D , we have that:

$$D_{KL}(f_1 \| f_2) = \frac{1}{2} \left(\text{tr}(\Sigma_2^{-1} \Sigma_1) - \ln \frac{|\Sigma_1|}{|\Sigma_2|} - D \right) + \frac{1}{2} (\mu_2 - \mu_1)^T \Sigma_2^{-1} (\mu_2 - \mu_1).$$

In the case of a binary classification problem, we can directly work with the above quantity for our mixture of two Gaussians. In the case where $|Y| > 2$, we consider the resulting $|Y|$ one-against-all binary classification problems and attempt to maximize the average of the upper bounds of the $|Y|$ mutual information values.

For each class y we consider the following mixture model:

$$f = p_y f_y + (1 - p_y) f_{Y \setminus y}$$

where the $f_{Y \setminus y}$ is the conditional distribution of $X|Y \neq y$. We then calculate the upper bound of the mutual information for all the possible mixtures f , one for each y .

Table 2: Greedy Forward Subset Selection

```

S0 ← ∅
for n = 1 . . . N do
  s* = 0
  for Xj ∈ F \ Sn-1 do
    s' ← Sn-1 ∪ Xj
    s ←  $\tilde{I}(S'; Y)$ 
    if s > s* then
      s* ← s
      S* ← S'
    end if
  end for
  Sn ← S*
end for
return SN

```

3.3 Greedy Forward Selection

The derived bounds and approximations provide measures for assessing the optimal set of features S_N^* of size N . However as there are $\frac{F!}{N!(F-N)!}$ possible sets S_N of size N finding the optimal one by checking all the candidate sets is computationally intractable. Due to this intractability we employ a greedy optimization process in order to find a good approximation S_N .

In particular, we use greedy forward selection (see table 2) to iteratively build a sequence of sets S_n with $n = 1, \dots, N$ where each set S_n is built by adding one feature $X_{j(n)}$ to the previous set S_{n-1} , i.e. $S_n = S_{n-1} \cup X_{j(n)}$. At a given iteration n , the greedy forward selection algorithm calculates for every candidate feature $X_j \in F \setminus S_{n-1}$, the mutual information between the set $S_n^* = S_{n-1} \cup \{X_j\}$ and the label Y . It then creates the set S_n by adding that feature which leads to the largest value of the optimization criterion.

3.4 Complexity of the Gaussian Compromise Method

Though forward selection leads to a computationally tractable feature selection algorithm, it remains nonetheless very expensive. In the case of the Gaussian compromise approach, at each iteration n and for each feature X_j not in S_{n-1} , forward selection requires the estimation of an approximation of $I(S_{n-1} \cup \{X_j\}; Y)$, which in turn requires the estimation of $|Y| + 1$ determinants of the size $n \times n$ covariance matrices. A naive approach would be to calculate these determinants from scratch, incurring a cubic cost of $O(n^3)$ for the calculation of each determinant and a $O(|Y| F \setminus S_{n-1} |n|^3)$ cost per iteration with a $O(|Y| n^2)$ memory requirement for storing the covariance matrices and their inverses.

As shown in previous work (Lefakis and Fleuret, 2014) however it is possible to derive both an $O(|Y| F \setminus S_{n-1} |n|^2)$ algorithm with $O(|Y| n^2)$ memory requirements and an $O(|Y| F \setminus S_{n-1} |n|)$ algorithm with $O(|Y| n^2 + |Y| F \setminus S_{n-1} |n|)$ memory requirements. In the following we expand upon those methods, in particular the $O(|Y| F \setminus S_{n-1} |n|)$ one, and

present an approach that allows for a $O(|Y||F \setminus S_{n-1}|n)$ with $O(|Y|n^2 + |Y||F \setminus S_{n-1}|)$ memory requirements.

In order to speed up computations, we first note that for three random variables X , Y , Z

$$I(X, Z; Y) = I(Z; Y) + I(X; Y | Z)$$

which in the context of our forward selection algorithm, $\forall X_j \in F \setminus S_{n-1}$, and with $S'_n = S_{n-1} \cup \{X_j\}$, translates to

$$I(S'_n; Y) = I(S_{n-1}; Y) + I(X_j; Y | S_{n-1}).$$

The first term in the above expression is common for all candidate features X_j , meaning that finding the feature X_j that maximizes $I(S'_n; Y)$ is equivalent to finding the feature that maximizes $I(X_j; Y | S_{n-1})$. If $\sigma_{j|S_{n-1}}^2$ denotes the variance of feature j conditioned on the features in S_{n-1} and $\sigma_{j|Y, S_{n-1}}^2$ the variance conditioned on the features in S_{n-1} and the class $Y = y$, we have

$$\begin{aligned} \operatorname{argmax}_{X_j \in F \setminus S_{n-1}} I(X_j; Y | S_{n-1}) &= \operatorname{argmax}_{X_j \in F \setminus S_{n-1}} (H(X_j | S_{n-1}) - H(X_j | Y, S_{n-1})) \\ &= \operatorname{argmax}_{X_j \in F \setminus S_{n-1}} \left(\log \sigma_{j|S_{n-1}}^2 - \sum_y P(Y = y) \log \sigma_{j|Y, S_{n-1}}^2 \right) \end{aligned} \quad (11)$$

where here and in the following, we slightly abuse notation and use S_n to denote both the set and its contents, which of the two is meant will in any case be clear from context. To derive equation 12 we have exploited the fact that the conditional variance $\sigma_{j|S_{n-1}}$ is independent of the specific values of the features in S_{n-1} and thus the integrations of the entropies over the conditioned values are straightforward. That is

$$\begin{aligned} H(X_j | S_{n-1}) &= \int_{\mathbb{R}^{|S_{n-1}|}} H(X_j | S_{n-1} = s) \mu_{S_{n-1}}(s) ds \\ &= \frac{1}{2} \log \sigma_{j|S_{n-1}}^2 + \frac{1}{2} (\log 2\pi + 1). \end{aligned}$$

Under the Gaussian assumption, we have

$$\sigma_{j|S_{n-1}}^2 = \sigma_j^2 - \sum_{j', S_{n-1}} \Sigma_{j', S_{n-1}}^{-1} \Sigma_{j, S_{n-1}}.$$

From the above we can derive an $O(|Y||F \setminus S_{n-1}|n^2)$ with $O(|Y|n^2)$ memory requirements by noting that computing $\Sigma_{j', S_{n-1}}^{-1} \Sigma_{j, S_{n-1}}$ incurs a cost of $O(n^2)$ and that this must be done for every candidate feature j and every class y .

3.4.1 EFFICIENT COMPUTATION OF $\sigma_{j|S_{n-1}}^2$

As stated, we can further speed-up the proposed algorithm by a factor of n by considering more carefully the calculation of $\sigma_{j|S_{n-1}}^2$. If $S_{n-1} = S_{n-2} \cup X_i$, we have

$$\Sigma_{j', S_{n-1}}^{-1} \Sigma_{j, S_{n-1}} = \left[\Sigma_{j', S_{n-2}}^{-1} \sigma_{j_i}^2 \right] \Sigma_{S_{n-1}}^{-1} \left[\Sigma_{j, S_{n-2}} \right] \sigma_{j_i}^2 \quad (13)$$

We note that $\Sigma_{S_{n-1}}$ differs from $\Sigma_{S_{n-2}}$ by the addition of a row and a column

$$\Sigma_{S_{n-1}} = \begin{bmatrix} \Sigma_{S_{n-2}} & \Sigma_{i, S_{n-2}} \\ \Sigma_{i, S_{n-2}}^T & \sigma_i^2 \end{bmatrix}$$

Thus $\Sigma_{S_{n-1}}$ is the result of a rank-two update to the augmented matrix

$$\begin{bmatrix} \Sigma_{S_{n-2}} & 0_{n-2} \\ 0_{n-2}^T & \sigma_i^2 \end{bmatrix},$$

specifically a one rank-one update corresponding to changing the final row and a rank-one update corresponding to changing the final column. By applying the Sherman-Morrison formula twice to update $\Sigma_{S_{n-2}}^{-1}$ to $\Sigma_{S_{n-1}}^{-1}$, we can obtain¹ an update formula of the form

$$\Sigma_{S_{n-1}}^{-1} = \begin{bmatrix} \Sigma_{S_{n-2}}^{-1} & -\frac{1}{\beta} \frac{u}{\sigma_i^2} \\ -\frac{1}{\beta} \frac{u^T}{\sigma_i^2} & \frac{1}{\beta} \left[\frac{u}{\sigma_i^2} \quad 0 \right] \end{bmatrix} \quad (14)$$

where

$$u = \Sigma_{S_{n-2}}^{-1} \Sigma_{i, S_{n-2}}$$

and

$$\beta = 1 - \frac{1}{\sigma_i^2} \Sigma_{i, S_{n-2}}^T \Sigma_{S_{n-2}}^{-1} \Sigma_{i, S_{n-2}}.$$

From equation (13) and (14) we have

$$\begin{aligned} \sigma_{j|S_{n-1}}^2 &= \sigma_j^2 - \Sigma_{j', S_{n-1}}^{-1} \left(\begin{bmatrix} \Sigma_{S_{n-2}}^{-1} & -\frac{1}{\beta} \frac{u}{\sigma_i^2} \\ -\frac{1}{\beta} \frac{u^T}{\sigma_i^2} & \frac{1}{\beta} \left[\frac{u}{\sigma_i^2} \quad 0 \right] \end{bmatrix} \Sigma_{j, S_{n-1}} \right) \\ &= \sigma_j^2 - \Sigma_{j', S_{n-2}}^{-1} \Sigma_{j, S_{n-2}} + \beta \frac{\sigma_{j_i}^2}{\sigma_i^2} u^T \Sigma_{j, S_{n-2}} \\ &\quad - \Sigma_{j', S_{n-1}}^{-1} \left[\begin{bmatrix} -\frac{1}{\beta} \frac{u}{\sigma_i^2} \\ \frac{1}{\beta} \frac{u^T}{\sigma_i^2} \end{bmatrix} \sigma_{j_i}^2 \right] \\ &\quad - \frac{1}{\beta \sigma_i^2} \left(\Sigma_{j', S_{n-1}}^{-1} \begin{bmatrix} u \\ 0 \end{bmatrix} \right) \left(\begin{bmatrix} u^T & 0 \end{bmatrix} \Sigma_{j, S_{n-1}} \right) \end{aligned} \quad (15)$$

In eq (15) the main computational cost is incurred by the calculation of $\Sigma_{j', S_{n-1}}^{-1} \Sigma_{j, S_{n-1}}$ which costs $O(n^2)$. However this quantity has been already calculated $\forall j$ during the previous iteration of the algorithm since this involves calculating

$$\sigma_{j_i|S_{n-2}}^2 = \sigma_j^2 - \Sigma_{j', S_{n-2}}^{-1} \Sigma_{j, S_{n-2}} \Sigma_{j_i, S_{n-2}}.$$

Thus we only need carry this result over from the previous iteration incurring an additional memory load of $O(|Y||F \setminus S_{n-1}|)$.

The remaining terms in equation (15) can be calculated in $O(n)$ given β and u . As β and u depend only on the feature i selected in the previous iteration, remaining constant throughout iteration n , they can be pre-computed once at the beginning of each iteration. Thus the cost of calculating $\sigma_{j|S_{n-2}}^2$ can be reduced to $O(n)$ and the overall computational cost per iteration to $O(|Y||F \setminus S_{n-1}|n)$.

¹. The proof follows from simple verification.

3.5 Complexity of the KL-based Algorithms

In the case of the KL-based algorithms, similarly with the Gaussian compromise approach, a naive implementation would incur a cost of $O(|Y|F \setminus S_{n-1}|n^3)$. In previous work (Lefakis and Fleuret, 2014) an algorithm was sketched which had a $O(|Y|n^2)$ memory footprint. Here we expand on this analysis and furthermore present an alternative algorithm with a $O(|Y|F \setminus S_{n-1}|n)$ complexity, which however has an increased memory footprint (specifically $O(|Y|F \setminus S_{n-1}|n^2)$).

Working with the value:

$$P(Y = y)D_{KL}(p(S|Y = y) \| p(S|Y \neq y)) + P(Y = y)H(S | Y = y) \\ + P(Y \neq y)D_{KL}(p(S|Y \neq y) \| p(S|Y = y)) + P(Y \neq y)H(S | Y \neq y) + e^m$$

we note that the entropy values $H(S | Y = y)$ can be computed efficiently as in the previous subsection. What remains is to efficiently compute the Kullback-Leibler divergences for each of the $|Y|$ binary classification problems.

As both distributions are assumed to be Gaussians, $D_{KL}(p(S|Y \neq y) \| p(S|Y = y))$ is equal to

$$\frac{1}{2} \left(\text{tr} \left(\Sigma_{S_n}^{y} \Sigma_{S_n}^{-1} \Sigma_{S_n}^{y^c} \right) - \log \frac{|\Sigma_{S_n}^{y^c}|}{|\Sigma_{S_n}^y|} + \left(\mu_{S_n}^{y^c} - \mu_{S_n}^y \right)^T \Sigma_{S_n}^{y^c} \Sigma_{S_n}^{-1} \left(\mu_{S_n}^{y^c} - \mu_{S_n}^y \right) - |S| \right).$$

In the following we show how each of these terms can be computed in time $O(n)$.

3.5.1 THE TERM $\log \frac{|\Sigma_{S_n}^{y^c}|}{|\Sigma_{S_n}^y|}$

From the chain rule,

$$H(S_{n-1} \cup X_j) = H(S_{n-1}) + H(X_j | S_{n-1})$$

we have

$$\log |\Sigma_{S_n}^y| + \frac{n}{2} (1 + \log 2\pi) = \log |\Sigma_{S_{n-1}}| + \frac{n-1}{2} (1 + \log 2\pi) + \log \sigma_{j|S_{n-1}}^2 + \frac{1}{2} (1 + \log 2\pi) \\ \log |\Sigma_{S_n}^{y^c}| = \log |\Sigma_{S_{n-1}}| + \log \sigma_{j|S_{n-1}}^2 \\ |\Sigma_{S_n}^y| = \sigma_{j|S_{n-1}}^2 |\Sigma_{S_{n-1}}|.$$

As shown in the previous section, the term $\sigma_{j|S_{n-1}}^2$ can be computed in $O(n)$ time. The term $|\Sigma_{S_{n-1}}|$ is independent of j and can be efficiently pre-computed from $|\Sigma_{S_{n-2}}|$ prior to iteration n using the matrix determinant lemma. By extension, the cost of calculating $\log \frac{|\Sigma_{S_n}^{y^c}|}{|\Sigma_{S_n}^y|}$ is itself $O(n)$.

3.5.2 CALCULATING $\Sigma_{S_n}^{y^c} u^{-1}$

Setting, here and in the rest of this section, $\Sigma_S = \Sigma_S^y$ for ease of exposition², we have similar to section 3.4 that

$$\Sigma_{S_n}^{y^c} u^{-1} = \begin{bmatrix} \Sigma_{S_{n-1}}^{-1} & -\frac{1}{\beta\sigma_j^2} u \\ -\frac{1}{\beta\sigma_j^2} u^T & \frac{1}{\beta\sigma_j^2} \end{bmatrix} + \frac{1}{\beta\sigma_j^2} \begin{bmatrix} u \\ 0 \end{bmatrix} \begin{bmatrix} u^T & 0 \end{bmatrix} \quad (16)$$

where

$$u = \Sigma_{S_{n-1}}^{-1} \Sigma_{j|S_{n-1}}$$

and

$$\beta = 1 - \frac{1}{\sigma_j^2} \Sigma_{j|S_{n-1}}^T \Sigma_{S_{n-1}}^{-1} \Sigma_{j|S_{n-1}}.$$

Here u and β cannot be pre-computed as they are different $\forall j$. They can either be calculated from scratch in $O(n^2)$ or, if we are willing to incur a memory overhead of $O(n)$, with $O(n)$ complexity from the product $\Sigma_{S_{n-2}}^{-1} \Sigma_{j|S_{n-2}}$ which has been computed during the previous iteration (as in section 3.4.1).

3.5.3 THE TERM $\left(\mu_{S_n}^{y^c} - \mu_{S_n}^y \right)^T \Sigma_{S_n}^{y^c} \Sigma_{S_n}^{-1} \left(\mu_{S_n}^{y^c} - \mu_{S_n}^y \right)$

Having computed u and β as defined above, we can efficiently calculate the product

$$M = \left(\mu_{S_n}^{y^c} - \mu_{S_n}^y \right)^T \Sigma_{S_n}^{-1} \left(\mu_{S_n}^{y^c} - \mu_{S_n}^y \right)$$

given that $\mu_{S_n}^y = \begin{bmatrix} \mu_{S_{n-1}}^y \\ \mu_j^y \end{bmatrix}$ by decomposing it as follows

$$M = \begin{pmatrix} \mu_{S_{n-1}}^{y^c} - \mu_{S_{n-1}}^y \\ \mu_j^{y^c} - \mu_j^y \end{pmatrix}^T \Sigma_{S_{n-1}}^{-1} \begin{pmatrix} \mu_{S_{n-1}}^{y^c} - \mu_{S_{n-1}}^y \\ \mu_j^{y^c} - \mu_j^y \end{pmatrix} \\ - \frac{\left(\mu_j^{y^c} - \mu_j^y \right)^T u^T \left(\mu_{S_{n-1}}^{y^c} - \mu_{S_{n-1}}^y \right)}{\beta\sigma_j^2} \\ + \left(\mu_{S_{n-1}}^{y^c} - \mu_{S_{n-1}}^y \right)^T \begin{bmatrix} -\frac{1}{\beta\sigma_j^2} u \\ \frac{1}{\beta\sigma_j^2} \end{bmatrix} \begin{pmatrix} \mu_j^{y^c} - \mu_j^y \end{pmatrix} \\ + \frac{1}{\beta\sigma_j^2} \left(\left(\mu_{S_{n-1}}^{y^c} - \mu_{S_{n-1}}^y \right)^T \begin{bmatrix} u \\ 0 \end{bmatrix} \right) \left(\begin{bmatrix} u^T & 0 \end{bmatrix} \left(\mu_{S_{n-1}}^{y^c} - \mu_{S_{n-1}}^y \right) \right)$$

Of these four terms, the final three involving the vectors u and β can be calculated in $O(n)$ as they only involve inner products. The first term requires $O(n^2)$, however as the term is independent of j it can be calculated once at the beginning of each iteration. Consequently the complexity of calculating $\left(\mu_{S_n}^{y^c} - \mu_{S_n}^y \right)^T \Sigma_{S_n}^{-1} \left(\mu_{S_n}^{y^c} - \mu_{S_n}^y \right)$ given u and β is $O(n)$.

² That is when a superscript is missing, u is implied.

3.5.4 THE TERM $\text{tr} \left(\Sigma_{S_n}^y -^{-1} \Sigma_{S_n'}^{y|y} \right)$

The term $\text{tr} \left(\Sigma_{S_n}^y -^{-1} \Sigma_{S_n'}^{y|y} \right)$ involves calculating, and summing, the main diagonal elements of the matrix product $\left(\Sigma_{S_n}^y -^{-1} \Sigma_{S_n'}^{y|y} \right)$. We have that

$$\Sigma_{S_n}^{y|y} = \begin{bmatrix} \Sigma_{S_{n-1}}^{y|y} & \Sigma_{j,S_{n-1}}^{y|y} \\ \Sigma_{j,S_{n-1}}^{y|y T} & \sigma_j^{y|y} \end{bmatrix}. \quad (17)$$

From equations (16) and (17) we see that the product can be decomposed into two parts. For the first

$$\begin{bmatrix} \Sigma_{S_{n-1}}^{-1} & -\frac{1}{\beta\sigma_j^2} u \\ -\frac{1}{\beta\sigma_j^2} u^T & \frac{1}{\beta\sigma_j^2} \end{bmatrix} \begin{bmatrix} \Sigma_{S_{n-1}}^{y|y} & \Sigma_{j,S_{n-1}}^{y|y} \\ \Sigma_{j,S_{n-1}}^{y|y T} & \sigma_j^{y|y} \end{bmatrix}$$

it is straightforward to show that the main diagonal elements can be calculated in $O(n)$ provided we have pre-computed the main diagonal elements of $\Sigma_{S_{n-1}}^{-1} \Sigma_{S_{n-1}}^{y|y}$. As this product does not depend on j this can be done prior to the beginning of the iteration. For the second we have

$$\frac{1}{\beta\sigma_j^2} \begin{bmatrix} \Sigma_{S_{n-1}}^{-1} \Sigma_{j,S_{n-1}} \\ 0 \end{bmatrix} \begin{bmatrix} \Sigma_{j,S_{n-1}}^{-1} \Sigma_{S_{n-1}}^{-1} & 0 \\ \Sigma_{j,S_{n-1}}^{y|y T} & \sigma_j^{y|y} \end{bmatrix}$$

which, given that $\text{tr}(w^T w A) = \text{tr}(w A w^T)$, is equal to the product of $\frac{1}{\beta\sigma_j^2}$ and the trace of

$$\Sigma_{j,S_{n-1}}^T \Sigma_{S_{n-1}}^{-1} \Sigma_{S_{n-1}}^{y|y} \Sigma_{S_{n-1}}^{-1} \Sigma_{j,S_{n-1}}.$$

As we have already calculated the vector $\Sigma_{S_{n-1}}^{-1} \Sigma_{j,S_{n-1}}$ when calculating $\Sigma_{S_{n-1}}^{-1}$ we concentrate here on calculating the vector $\Sigma_{j,S_{n-1}}^T \Sigma_{S_{n-1}}^{-1} \Sigma_{S_{n-1}}^{y|y}$.

As shown, the matrix $\Sigma_{S_{n-1}}^{-1}$ can be written in the form

$$\Sigma_{S_{n-1}}^{-1} = \begin{bmatrix} \Sigma_{S_{n-2}}^{-1} & \gamma v \\ \gamma v^T & -\gamma \end{bmatrix} \begin{bmatrix} v \\ 0 \end{bmatrix} \begin{bmatrix} v^T & 0 \end{bmatrix}$$

where

$$v = \Sigma_{S_{n-2}}^{-1} \Sigma_{i,S_{n-2}}$$

and

$$\gamma = -\frac{1}{\beta\sigma_j^2}.$$

Thus the vector $\Sigma_{j,S_{n-1}}^T \Sigma_{S_{n-1}}^{-1} \Sigma_{S_{n-1}}^{y|y}$ can be decomposed into

$$\Sigma_{j,S_{n-1}}^T \begin{bmatrix} \Sigma_{S_{n-2}}^{-1} & \gamma v \\ \gamma v^T & -\gamma \end{bmatrix} \Sigma_{S_{n-1}}^{y|y} - \gamma \Sigma_{j,S_{n-1}}^T \begin{bmatrix} v \\ 0 \end{bmatrix} \begin{bmatrix} v^T & 0 \end{bmatrix} \Sigma_{S_{n-1}}^{y|y}.$$

As v does not depend on j the product $\begin{bmatrix} v^T & 0 \end{bmatrix} \Sigma_{S_{n-1}}^{y|y}$ can be computed prior to the iteration and by extension the product $\gamma \Sigma_{j,S_{n-1}}^T \begin{bmatrix} v \\ 0 \end{bmatrix} \begin{bmatrix} v^T & 0 \end{bmatrix} \Sigma_{S_{n-1}}^{y|y}$ can be computed in $O(n)$. This leaves the final term

$$\Sigma_{j,S_{n-1}}^T \begin{bmatrix} \Sigma_{S_{n-2}}^{-1} & \gamma v \\ \gamma v^T & -\gamma \end{bmatrix} \Sigma_{S_{n-1}}^{y|y}.$$

We note that the vector $\Sigma_{j,S_{n-1}}^T \begin{bmatrix} \Sigma_{S_{n-2}}^{-1} & \gamma v \\ \gamma v^T & -\gamma \end{bmatrix}$ has the form

$$\begin{bmatrix} \Sigma_{j,S_{n-2}} \Sigma_{S_{n-2}}^{-1} + \gamma \sigma_{j,i}^2 v^T \\ \gamma \Sigma_{j,S_{n-2}}^T v - \gamma \sigma_{j,i}^2 \end{bmatrix}^T.$$

Thus we have

$$\begin{bmatrix} \Sigma_{j,S_{n-2}} \Sigma_{S_{n-2}}^{-1} \Sigma_{S_{n-2}}^{y|y} & 0 \\ \gamma \Sigma_{j,S_{n-2}}^T \Sigma_{S_{n-2}}^{y|y} & 0 \end{bmatrix} + \gamma \sigma_{j,i}^2 \begin{bmatrix} v^T & 0 \\ 0 & 0 \end{bmatrix} \Sigma_{S_{n-1}}^{y|y} + \left(\gamma \Sigma_{j,S_{n-2}}^T v - \gamma \sigma_{j,i}^2 \right) \Sigma_{i,S_{n-1}}^{y|y}.$$

During the previous iteration we have already computed the vector $\Sigma_{j,S_{n-2}} \Sigma_{S_{n-2}}^{-1} \Sigma_{S_{n-2}}^{y|y}$ and thus if we use $O(n)$ memory to store it between iterations, we can also compute this final term in $O(n)$.

4. Using the Eigen-decomposition to Bound Computations

The fast implementation of the GC-approximation method presented in 3.4 requires at iteration n the calculation, for each feature $X_j \in F \setminus S_{n-1}$, of the conditioned variance

$$\sigma_{j|S_{n-1}}^2 = \sigma_j^2 - \Sigma_{j,S_{n-1}}^T \Sigma_{S_{n-1}}^{-1} \Sigma_{j,S_{n-1}}.$$

As shown, each such computation can be done in $O(n)$. The main computational cost comes from the calculation of $\Sigma_{j,S_{n-1}}^T \Sigma_{S_{n-1}}^{-1} \Sigma_{j,S_{n-1}}$. Thus it would be advantageous to acquire a ‘‘cheap’’ (independent of n) bound which will allow us to skip the calculation of this quantity for certain non-promising features.

We note that $\Sigma_{S_{n-1}}^{-1}$ is positive definite and symmetric and thus can be decomposed as

$$\Sigma_{S_{n-1}}^{-1} = U \Lambda U^T,$$

where U is orthonormal and Λ is a diagonal matrix with positive elements. Thus

$$\Sigma_{j,S_{n-1}}^T \Sigma_{S_{n-1}}^{-1} \Sigma_{j,S_{n-1}} = \Sigma_{j,S_{n-1}}^T U \Lambda U^T \Sigma_{j,S_{n-1}}.$$

As U is orthonormal we have

$$\|\Sigma_{j,S_{n-1}}\|_2 = \|\Sigma_{j,S_{n-1}}^T U\|_2 = \|U^T \Sigma_{j,S_{n-1}}\|_2,$$

Symbolizing the eigenvalues as $\lambda_1, \lambda_2, \dots, \lambda_{n-1}$ and the elements of the vector $\Sigma_{j,S_{n-1}}^T U$ as x_1, x_2, \dots, x_{n-1} , we have that

$$\|\Sigma_{j,S_{n-1}}^T\|_2^2 \min_i \lambda_i \leq \Sigma_{j,S_{n-1}}^T \Sigma_{S_{n-1}}^{-1} \Sigma_{j,S_{n-1}} \leq \|\Sigma_{j,S_{n-1}}^T\|_2^2 \max_i \lambda_i. \quad (18)$$

Equation 18 gives us a bound, computable in $O(|Y|)$, which we can use to avoid unnecessary computations. Specifically, during iteration n of the algorithm, after having already calculated the scores of a subset of features, we have a candidate for best feature which has a score of s^* ; for each subsequent candidate feature X_j we can compute the following upper bound on the feature's score

$$ub_1(X_j) = \log \left(\sigma_j^2 - \|\Sigma_{j|S_{n-1}}^T\|_2^2 \max_i \lambda_i \right) - \sum_{y=0}^{|Y|-1} \left(\log \left(\sigma_j^{y^2} - \|\Sigma_{j|S_{n-1}}^{yT}\|_2^2 \min_i \lambda_i^y \right) \right), \quad (19)$$

where λ_i, λ_i^y are the eigenvalues of $\Sigma_{S_{n-1}}^{-1}$ and $\Sigma_{S_{n-1}}^{y^{-1}}$ respectively. Then if $ub(s) \leq s^*$, we can avoid the $O(|Y|n)$ computations required to estimate the feature's score s .

Furthermore should $ub_1(X_j) > s^*$ we can still proceed with the computations in a greedy manner. That is instead of calculating the exact score

$$\log \left(\sigma_j^2 |S_{n-1}| - \sum_{y=0}^{|Y|-1} \left(\log \left(\sigma_j^{y^2} |S_{n-1}| \right) \right) \right)$$

incurring $O(|Y|n)$ cost, we can compute the conditional variances one at a time and then reassess the upper bound. That is we first calculate $\sigma_{j|S_{n-1}}^2$ which costs us $O(n)$ and then re-estimate the upper bound as

$$ub_2(X_j) = \log \left(\sigma_{j|S_{n-1}}^2 \right) - \sum_{y=0}^{|Y|-1} \left(\log \left(\sigma_j^{y^2} - \|\Sigma_{j|S_{n-1}}^{yT}\|_2^2 \min_i \lambda_i^y \right) \right)$$

re-checking whether $ub(s) \leq s^*$. We can then continue, if necessary, by calculating

$$ub_3(X_j) = \log \left(\sigma_{j|S_{n-1}}^2 \right) - \log \left(\sigma_{j|S_{n-1}}^{y_0^2} \right) - \sum_{y=1}^{|Y|-1} \left(\log \left(\sigma_j^{y^2} - \|\Sigma_{j|S_{n-1}}^{yT}\|_2^2 \min_i \lambda_i^y \right) \right)$$

and so forth. Thus we can avoid estimating a number of conditional variances, incurring a smaller cost. This process can continue greedily by computing $ub_{1\dots|Y|+1}(X_j)$.

We note that the upper bound 19 involves $|Y|$ lower bounds

$$\log \left(\sigma_j^{y^2} - \|\Sigma_{j|S_{n-1}}^{yT}\|_2^2 \min_i \lambda_i^y \right)$$

on the conditional variances $\sigma_{j|S_{n-1}}^{y^2}$. As such the bound can become quite loose if the number of classes $|Y|$ is large. In order to empirically evaluate the usefulness of this bound in pruning computations, we considered 3 binary classification tasks: the binary task of the INRIA data set, and two tasks resulting from the CIFAR and STL data sets by a random partitioning of the classes (*i.e.* that is in each case 5 classes were randomly chosen to be labeled as positive and the rest as negative).

In figure 3 we can see for each of the three data sets, the number of features at each round for which we can skip a certain amount of computations. In the case $ub_1 \leq s^*$ we

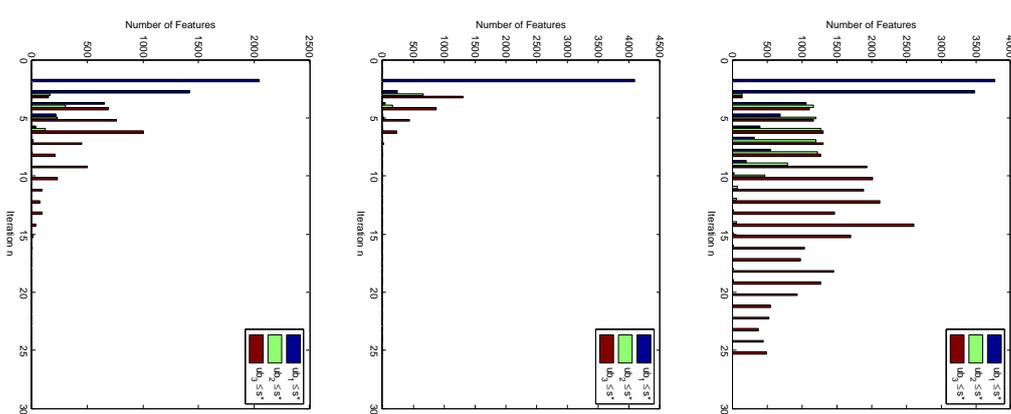


Figure 3: Number of features for which the upper bound ub_x allows us to skip computations for the top) INRIA, middle) STL, and bottom) CIFAR data sets.

can avoid all computations concerning the conditional variances. In the case $ub_2 \leq s^*$ only one conditional variance need be calculated, while in the case of $ub_3 \leq s^*$ two (out of a possible three). As can be seen, the bound can prove quite useful in pruning computations, as is the case in the INRIA data set. For the CIFAR data set, we see that the bound can still prove useful, especially early on. On the contrary in the case of the STL data set, the bound seems to provide little help in ways of avoiding computations.

Pruning computations using the above bounds requires access to the eigenvalues of the matrices $\Sigma_{S_{n-1}}^{y-1}$ which are the reciprocals of the eigenvalues of the matrices $\Sigma_{S_{n-1}}^y$. As computing the eigen-decomposition of a matrix, from scratch, can be expensive we present in the following a novel algorithm for efficiently calculating these eigenvalues, and the corresponding eigenvectors, of $\Sigma_{S_{n-1}}^{y-1}$ from the eigen-decomposition of $\Sigma_{S_{n-2}}^{y-1}$.

4.1 Eigen-system Update

The matrices $\Sigma_{S_{n-1}}^{y-1}$ results from the matrices $\Sigma_{S_{n-2}}^{y-1}$ by the addition of a row and a column. We are faced thus with the problem of updating the eigen-system of a symmetric and positive definite matrix Σ when a vector is inserted as an extra row and column.

More specifically, we shall present a method for *efficiently* computing the eigen-system U^{n+1}, Λ^{n+1} of a $(n+1) \times (n+1)$ matrix Σ^{n+1} when the eigen-system U^n, Λ^n of the $n \times n$ matrix Σ^n is known and Σ^{n+1} is related to Σ^n as follows:

$$\Sigma^{n+1} = \begin{bmatrix} \Sigma^n & \mathbf{v} \\ \mathbf{v}^T & c \end{bmatrix},$$

where \mathbf{v} and c are such that the matrix Σ^{n+1} is positive definite. Without loss of generality, we will consider in the following the special case $c = 1$.

If $\mathbf{u}_1, \dots, \mathbf{u}_n$, $\lambda_1^n, \dots, \lambda_n^n$ are the eigenvectors and respective eigenvalues of Σ^n , then $\begin{bmatrix} \mathbf{u}_i^n \\ 0 \end{bmatrix}$ is obviously an eigenvector of

$$\Sigma^y = \begin{bmatrix} \Sigma^n & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix},$$

with associated eigenvalue λ_i^n . Also, if $\forall i \in \{1, \dots, n+1\}$, \mathbf{e}_i is the i th standard basis vector of \mathbb{R}^{n+1} , then $\Sigma^y \mathbf{e}_{n+1} = \mathbf{e}_{n+1}$ and \mathbf{e}_{n+1} is an eigenvector of Σ^y with a corresponding eigenvalue of $\lambda_{n+1}^y = 1$.

The matrix Σ^{n+1} can be expressed as the result of a rank-two update to Σ^y

$$\Sigma^{n+1} = \Sigma^y + \mathbf{e}_{n+1} \begin{bmatrix} \mathbf{v} \\ 0 \end{bmatrix}^T + \begin{bmatrix} \mathbf{v} \\ 0 \end{bmatrix} \mathbf{e}_{n+1}^T.$$

If U^y, Λ^y denote the eigen-system of Σ^y , by multiplying

$$\Sigma^{n+1} \mathbf{u}^{n+1} = \Lambda^{n+1} \mathbf{u}^{n+1},$$

on the left by U^{yT} , we have

$$U^{yT} \left(\Sigma^y + \mathbf{e}_{n+1} \begin{bmatrix} \mathbf{v} \\ 0 \end{bmatrix}^T + \begin{bmatrix} \mathbf{v} \\ 0 \end{bmatrix} \mathbf{e}_{n+1}^T \right) \mathbf{u}^{n+1} = \Lambda^{n+1} U^{yT} \mathbf{u}^{n+1}$$

Given that Σ^y is positive definite and symmetric it follows that $U^y U^{yT} = I$ and

$$U^{yT} \left(\Sigma^y + \mathbf{e}_{n+1} \begin{bmatrix} \mathbf{v} \\ 0 \end{bmatrix}^T + \begin{bmatrix} \mathbf{v} \\ 0 \end{bmatrix} \mathbf{e}_{n+1}^T \right) U^y U^{yT} \mathbf{u}^{n+1} = \Lambda^{n+1} U^{yT} \mathbf{u}^{n+1}$$

and since $U^{yT} \Sigma^y U^y = \Lambda^y$ we have

$$(\Lambda^y + \mathbf{e}_{n+1} \mathbf{q}^T + \mathbf{q} \mathbf{e}_{n+1}^T) U^y U^{yT} \mathbf{u}^{n+1} = \Lambda^{n+1} U^{yT} \mathbf{u}^{n+1}$$

where $\mathbf{q} = U^{yT} \begin{bmatrix} \mathbf{v} \\ 0 \end{bmatrix}$, note $\mathbf{e}_{n+1}^T U^y = \mathbf{e}_{n+1}^T$.

Thus Σ^{n+1} and the matrix $\Sigma'' = (\Lambda^y + \mathbf{e}_{n+1} \mathbf{q}^T + \mathbf{q} \mathbf{e}_{n+1}^T)$ share eigenvalues. Furthermore the eigenvectors U^{n+1} are related to the eigenvectors U'' of Σ'' by $U^{n+1} = U^y U''$.

4.1.1 COMPUTING THE EIGENVALUES AND EIGENVECTORS OF Σ''

The matrix Σ'' has non-zero elements only on its main diagonal and on its last row and column. By developing the determinant $|\Sigma'' - \lambda I|$ along the final row we have

$$|\Sigma'' - \lambda I| = \prod_j (\lambda_j' - \lambda) + \sum_{i < n+1} \left(-q_i^2 \prod_{j \neq i, j < n+1} (\lambda_j' - \lambda) \right),$$

where q_i is the i th element of vector \mathbf{q} . The determinant thus has the same roots as the function

$$f(\lambda) = \frac{|\Sigma'' - \lambda I|}{\prod_{j < n+1} (\lambda_j' - \lambda)} \quad (20)$$

$$= \lambda'_{n+1} - \lambda + \sum_i \frac{-q_i^2}{(\lambda_i' - \lambda)}. \quad (21)$$

We have $\forall i, \lim_{\lambda \rightarrow \lambda_i'} f(\lambda) = +\infty$ and $\lim_{\lambda \rightarrow \lambda_i'} f(\lambda) = -\infty$. Furthermore we have

$$\frac{\partial f(\lambda)}{\partial \lambda} = -1 + \sum_i \frac{-q_i^2}{(\lambda_i' - \lambda)^2} \leq 0$$

meaning the function f is strictly decreasing between its poles. From this and from the positive definiteness of Σ'' we have that

$$0 < \lambda_1^{n+1} < \lambda_1' < \lambda_1^{n+1} < \dots < \lambda'_{n+1} < \lambda_{n+1}^{n+1}$$

i.e. the eigenvalues of Σ' and Σ'' are interlaced.

Though there is no analytical solution for finding the roots of $f(\lambda)$, given the above relationship they can be computed efficiently using a Householder method. We also note that $\text{tr}(\Lambda^y) = \text{tr}(\Sigma'')$ and consequently

$$\lambda_{n+1}^{n+1} = \sum_i \lambda_i' - \sum_{i < n+1} \lambda_i^{n+1}.$$

Once we have computed the eigenvalues λ'' , we can compute the eigenvectors U'' as follows: we first note that $\forall k$ the system of linear equations

$$\Sigma'' x = \lambda_k^{n+1} x$$

where

$$x = [x_1 \quad x_2 \quad \dots \quad x_{n+1}]^T$$

involves n equations of the form

$$\lambda_k' x_i + q_i x_{n+1} = \lambda_k^{n+1} x_i.$$

Given that $\lambda_k^{n+1} \neq \lambda_k'$, it follows that if $x_{n+1} = 0$ then $\forall i, x_i = 0$ and thus it must be that $x_{n+1} \neq 0$. As the system has one degree of freedom, we can set $x_{n+1} = 1$. We then have from the equations

$$\lambda_k' x_i + q_i x_{n+1} = \lambda_k^{n+1} x_i$$

that

$$x_i = \frac{q_i}{\lambda_k^{n+1} - \lambda_k'},$$

and by normalizing x we obtain the k th column of U'' . Finally we can obtain the eigen-decomposition of Σ^{n+1}

$$\Sigma^{n+1} = (U'' U''^T) \Lambda'' (U'' U''^T)^T.$$

4.1.2 COMPUTATIONAL EFFICIENCY

In order to empirically evaluate the derived eigen-decomposition update algorithm, we compare a C++ implementation against the LAPACK library's eigen-decomposition implementation. In figure 4.1.2 we see such a comparison of the CPU time required to compute Σ^{n+1} , from scratch, using the LAPACK library and using the proposed update algorithm, as a function of n .

4.1.3 NUMERICAL PRECISION

In order to test the numerical precision of the proposed update method, we consider two experimental setups. In the first setup, results on which are shown in figures 5.6, we iteratively augment a symmetric positive definite matrix by inserting a vector as an extra row and an extra column and at each iteration update its eigen-decomposition using the proposed method; that is to say at each iteration the eigen-decomposition is an updated version of previously updated decompositions. Figure 5 shows the maximum relative eigenvalue error as compared to the decomposition estimated by LAPACK which is assumed to be accurate. Similarly figure 6 shows the maximum angle between corresponding eigenvectors of the proposed method and the ones computed by the LAPACK library. As can be seen, after 2000 iterations the *maximum* relative eigenvalue error is of the order of magnitude of $\sim 1\%$, while the maximum angle between corresponding eigenvectors is less than 0.001 degrees.

In figure 7 we show the maximum relative eigenvalue when the eigen-decomposition is updated from the decomposition computed at the previous iteration using LAPACK.

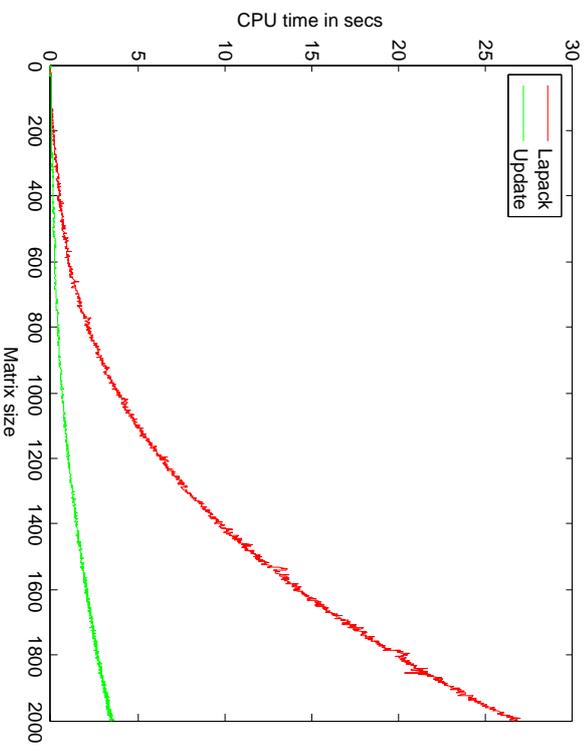


Figure 4: Comparison of computation time of the proposed update method and of computing the eigen-decomposition from scratch using the LAPACK library.

As can be seen the maximum relative error is of the order of magnitude of 10^{-10} . This value is related to the tolerance of the Newton method which was used to find the roots of equation 20 (which in these experiments we set to 10^{-10}). The maximum angle between corresponding eigenvectors was found to be 0, *i.e.* beneath double precision, in every case.

5. Experiments

In this section we present an empirical evaluation of the proposed algorithms. We first show on a synthetic controlled experiment that they behave as expected regarding groups of jointly informative features, and then provide results obtained on three popular real-world computer vision data sets.

5.1 Synthetic Examples

In order to show the importance of joint informativeness and the ability of the proposed algorithm to capture it we construct a simple synthetic experiment with a set of candidate features $F = \{X_1, X_2, X_3, X_4, X_5\}$ defined as follows:

$$\begin{aligned} X_1 &\sim N(0, 1) + 10^{-1}Y \\ X_2 &\sim (2Y - 1)X_1 + N(0, 1) \\ X_3 &\sim N(0, 1) \\ X_4 &\sim N(0, 1) + 10^{-1}Y \\ X_5 &\sim (2Y - 1)X_4 + N(0, 1) \end{aligned}$$

where Y is a classification label, $Y \sim B(0.5)$. Looking at the above marginals it can be seen that only X_1 and X_4 carry information regarding Y individually, X_2 and X_5 are very informative but only in conjunction with X_1 and X_4 respectively. X_3 is simply noise.

We generate 1,000 synthetic data sets of 25,000 data points each from the above distributions, and use the GC-approximation on the mutual information to select the features. In 49.2% of the experiments the algorithm ranks X_1 as the most informative feature. Even though X_4 would be the second most informative feature marginally, in every one of these experiments in the second iteration the algorithm chose the X_2 feature as it is jointly more informative when combined with X_1 . Similarly, in 47.1% of the experiments the algorithm ranked X_4 first and in each of these experiments selected X_5 second. In every one of the runs X_3 was ranked as the least informative of the 5 features.

5.2 Data Sets

We report results on three standard computer vision data-sets which we used for our experiments:

CIFAR-10 contains images of size 32×32 of 10 distinct classes depicting vehicles and animals. The training data consists of 5,000 images of each class. We pre-process the data as in (Coates and Ng, 2011) using the code provided by the authors. The original pool F of features consists of 2,048 candidates.

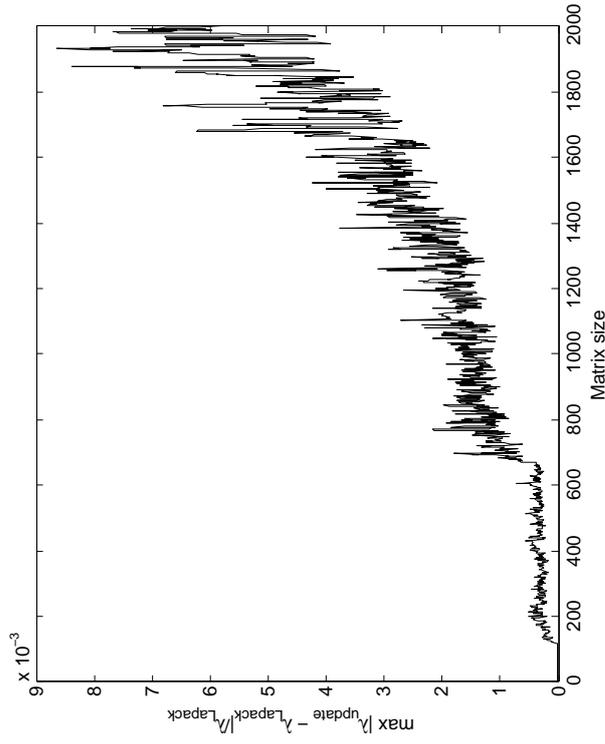


Figure 5: Numerical precision of the proposed update method when updating the decomposition iteratively. Maximum relative eigenvalue error.

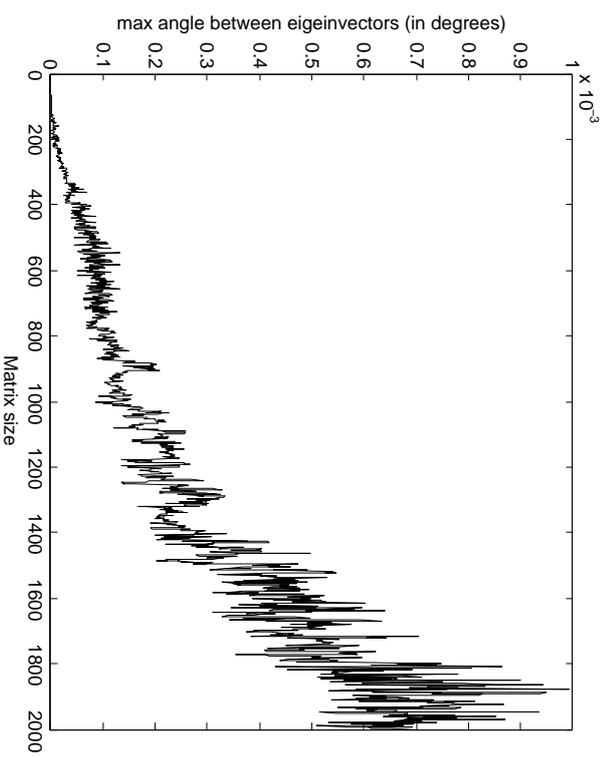


Figure 6: Numerical precision of the proposed update method when updating the decomposition iteratively. Maximum angle between corresponding eigenvectors.

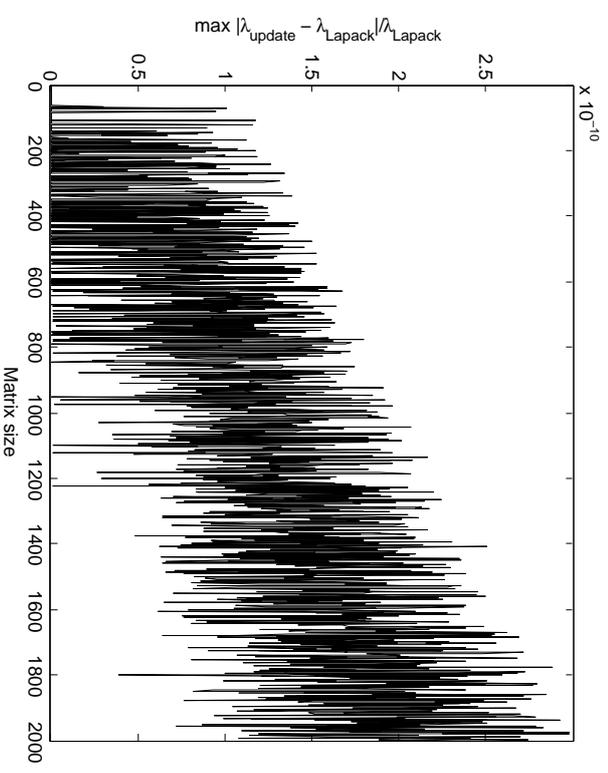


Figure 7: Maximum relative eigenvalue error when the eigen-decomposition is updated from the eigen-decomposition computed by LAPACK.

INRIA is a pedestrian detection data set. There are 12, 180 training images of size 64×128 of pedestrians and background images. We use 3, 780 HoG features that have been shown to perform well in practice (Dalal and Triggs, 2005).

STL-10 consists of images of size 96×96 belonging to 10 classes, each represented by 500 training images. As for CIFAR we pre-process the data as in (Coates and Ng, 2011), resulting in a pool F of 4, 096 features.

5.3 Baselines

We compare the proposed feature selection methods against a number of baselines. The **Fisher**, **T-test**, χ^2 , and **InfoGain** methods all compute statistics on individual features. In particular **InfoGain** calculates the mutual information of the individual features to the class, without taking into account their joint informativeness. As such, its comparison with our approaches is a very good indicator of the merit of joint informativeness and its effect on classification performance.

As noted in section 2, the **FCBF** (Liu and Yu, 2003) and **CFS** (Hall and Smith, 1999) baselines employ symmetric uncertainty criteria and check for pairwise redundancy of features. Similarly **MRRM** (Peng et al., 2005), uses mutual information to select features that have high relevance to the class while having low mutual information with the other selected features, thus checking for pairwise informativeness. Comparison with the proposed methods proves the importance of going beyond pairwise redundancy.

The **ReliefF** baseline (Robnik-Sikonja and Kononenko, 2003) looks at the nearest neighbors of random samples along the individual features. In order to compare with spectral clustering approaches we show results for (Wolf and Shashua, 2005), marked as **Spectral** in the tables, which we found to outperform (Zhao and Liu, 2007) in practice. Finally, we also show results for three wrapper methods, namely **SBMLR** (Cawley et al., 2006), which employs a logistic regression predictor, **CMTF** (Argyriou et al., 2008) which uses a sparsity inducing l_1 -norm, and **GBFS** (Xu et al., 2014) which uses gradient boosted trees.

We compare these baselines against the four methods proposed here, namely maximizing the entropy (**GC.E**) or the mutual information (**GC.MI**) under the Gaussian compromise approximation, and maximizing the KL-based entropy (**GKLE**) and mutual information (**GKL.MI**). In the case of the **GC** methods, when an iteration is reached where for all candidate features and for all classes the prior over the class variable is lower than the entropy of f^* , we halt the GC-approximation feature selection procedure and randomly select the remaining features.

5.4 Results

In tables 4, 5, 6, and 7, we show experimental results for the three data sets. In order to show the general applicability of the proposed methods, we combined the selected features with four different classifiers: AdaBoost with classification stumps, linear SVM, RBF-kernel SVM, and quadratic discriminant analysis (QDA). We show results for several numbers of selected features $\{10, 25, 50, 100\}$.

In each of these tables, for each data set and classifier we highlight the best three performing methods in bold, while underlining the best performing method. As can be seen

from these tables, **GC.MI** and **GKLE** consistently rank amongst the top three methods, with **GC.MI** ranking in the top three 24 out of 48 times and first 8 times, and **GKLE** being in the top three 27 out of 48 times and first 11 times. The only other comparable methods are **SBMLR**, which ranks in the top three 17 out of 48 times and first 10 times and **GBFS** which ranks in the top three 20 out of 48 times and first 5 times. We note that both of these methods are wrapper methods.

Furthermore as can be seen in table 3, the running time of the proposed methods³ is very competitive with respect to the more complex feature selection algorithms. The Gaussian Compromise algorithms are especially fast as they are two orders of magnitude faster than practically all other methods. We especially note that the **SBMLR** method which performs comparably in terms of accuracy is very slow when compared to **GC.MI**. Though the **MRRM** algorithm is faster than **GC.MI** on the INRIA data set, it performs considerably worse accuracy-wise on that data set; on the data sets where it does perform well (e.g. CIFAR) it is considerably slower than **GC.MI**. We also note that the **GBFS** method is quite slower than **GC.MI**.

The computation times provided were obtained with C++ implementations of the proposed methods. The **MRRM** algorithm is also implemented in C++, while the **Spectral** and **CMTF** baselines are implemented in MATLAB, as both these algorithms mainly use matrix algebra we believe these timings to be indicative. The remaining algorithms were implemented in Java. As noted in (Bouckaert et al., 2010) these implementations should be competitive in speed with C++ implementations.

	FCBF	MRRM	SBMLR	Spectral	GBFS	CFS	CMTF	ReliefF	GC.MI	GKLE
CIFAR	621	56	1449	1379	95	4262	394	1652	20	486
STL	68	20	1002	367	41	409	208	2089	5	887
INRIA	247	32	88	1072	233	2516	459	2413	43	135

Table 3: (Approximate) Cost in CPU time of running the more sophisticated feature selection algorithms in order to select 100 features on the three data sets. We highlight in bold the fastest algorithm for each data set.

5.5 Finite Sample Analysis

The proposed methods all depend on the covariance matrices Σ, Σ^y . Of course, in practice, one rarely has access to the true covariance matrices of the underlying distributions but rather estimates based on using finite sets of samples. Given a $N \times D$ matrix P where each row is a sample from the underlying D -dimensional distribution, we symbolize

$$\hat{\Sigma}_N = \frac{1}{N} P^T P$$

the empirical estimation $\hat{\Sigma}_N$ of matrix Σ , computed from these N samples. The accuracy of this approximation is important to the success of the proposed methods. It can be shown

3. We note that for the **GKLE/MI** methods we used an $O(n^2)$ (per feature per iteration) implementation, in practice and assuming access to adequate memory the method should be even faster.

	CIFAR					STL					INRIA				
	10	25	50	100	100	10	25	50	100	100	10	25	50	100	100
AdaBoost	29.23	36.96	42.07	49.06	31.86	35.78	39.72	41.81	86.90	89.83	90.38	91.45	91.45	91.45	91.45
Fisher	37.77	44.42	51.15	54.83	33.25	38.05	39.87	42.81	90.87	94.02	95.44	94.67	94.67	94.67	94.67
FCBF	39.42	45.84	49.76	54.85	32.24	39.61	40.61	43.00	81.53	88.48	93.48	94.91	94.91	94.91	94.91
MIRMR	28.13	35.54	43.68	49.46	29.61	36.88	39.39	41.89	92.81	93.11	93.94	94.91	94.91	94.91	94.91
χ^2	34.87	45.08	52.17	56.70	34.22	41.26	44.65	47.15	86.40	97.50	88.04	88.06	88.06	88.06	88.06
SBMLR	25.74	31.30	36.57	43.16	31.74	34.75	39.31	42.34	85.01	88.41	88.84	91.70	91.70	91.70	91.70
tTest	29.01	35.90	40.20	48.34	31.13	36.60	38.62	42.03	92.58	93.29	93.96	94.93	94.93	94.93	94.93
InfoGain	19.90	25.13	33.18	40.44	19.06	26.30	33.52	38.51	92.78	93.69	93.92	94.83	94.83	94.83	94.83
Spectral	28.13	34.64	40.85	47.70	33.91	37.46	42.79	45.22	91.79	95.44	95.83	96.43	96.43	96.43	96.43
RelieFF	33.50	38.96	44.58	54.22	30.75	38.40	41.85	44.39	89.69	92.60	96.41	97.69	97.69	97.69	97.69
CFS	33.50	38.96	44.58	54.22	30.75	38.40	41.85	44.39	89.69	92.60	96.41	97.69	97.69	97.69	97.69
CMTF	21.79	31.98	39.43	45.23	28.70	33.55	34.71	36.86	80.01	83.72	92.55	95.58	95.58	95.58	95.58
GBFS	32.02	40.20	48.87	54.34	30.96	38.56	42.30	45.57	93.90	95.87	96.90	97.66	97.66	97.66	97.66
G.C.E	32.45	42.54	50.15	55.06	31.86	37.41	42.19	46.99	89.54	90.09	94.30	95.81	95.81	95.81	95.81
G.C.MI	36.47	44.55	51.44	55.39	36.50	40.79	43.82	44.39	95.04	95.87	96.68	97.30	97.30	97.30	97.30
G.K.L.E	37.51	46.41	52.11	56.41	34.76	39.71	43.49	46.46	89.92	91.84	94.14	96.63	96.63	96.63	96.63
G.K.L.MI	33.71	40.04	47.17	51.12	33.00	38.80	42.13	43.58	92.18	93.09	95.21	96.15	96.15	96.15	96.15

Table 4: Test accuracy of an AdaBoost classifier trained on a different number of selected features {10, 25, 50, 100} on the three data sets.

	CIFAR					STL					INRIA				
	10	25	50	100	100	10	25	50	100	100	10	25	50	100	
SVMLin	25.19	33.53	39.47	48.12	26.09	30.79	34.63	38.02	92.55	93.73	94.03	94.68	94.68	94.68	94.68
Fisher	33.65	42.02	47.77	54.97	31.74	34.85	38.11	40.66	94.14	96.03	96.03	96.03	96.03	96.03	96.03
FCBF	35.48	42.53	46.02	52.64	32.50	39.06	43.69	49.36	79.85	84.18	91.73	93.91	93.91	93.91	93.91
MIRMR	21.77	32.06	40.65	48.58	22.61	31.82	34.29	37.96	92.94	93.27	93.50	94.61	94.61	94.61	94.61
χ^2	30.43	42.60	51.41	56.81	32.29	38.26	43.29	47.15	85.92	87.95	88.57	88.64	88.64	88.64	88.64
SBMLR	25.69	32.56	40.17	45.12	26.72	29.95	36.23	39.14	80.01	87.21	87.64	89.23	89.23	89.23	89.23
tTest	24.79	32.32	37.98	47.37	27.17	31.82	33.70	37.84	92.35	93.08	93.64	94.44	94.44	94.44	94.44
InfoGain	17.19	23.14	32.78	42.60	18.91	26.55	32.65	38.24	92.67	93.57	93.64	94.68	94.68	94.68	94.68
Spectral	24.56	30.60	38.17	46.51	29.16	32.40	38.05	42.94	90.99	95.04	95.97	96.36	96.36	96.36	96.36
RelieFF	31.49	36.46	42.17	51.70	28.63	34.45	38.54	41.88	88.64	91.68	96.11	97.53	97.53	97.53	97.53
CFS	31.49	36.46	42.17	51.70	28.63	34.45	38.54	41.88	88.64	91.68	96.11	97.53	97.53	97.53	97.53
CMTF	21.10	31.64	40.39	47.71	27.61	34.81	38.99	42.32	79.09	80.29	89.49	92.05	92.05	92.05	92.05
GBFS	28.37	38.18	45.89	52.36	30.78	39.29	45.06	50.39	76.79	87.55	95.38	97.03	97.03	97.03	97.03
G.C.E	28.76	41.14	48.70	55.16	31.20	37.60	43.31	49.75	87.73	92.57	91.96	93.13	93.13	93.13	93.13
G.C.MI	34.02	42.14	49.16	55.07	32.50	39.75	44.15	48.88	89.76	93.09	95.71	96.45	96.45	96.45	96.45
G.K.L.E	32.39	43.26	50.12	55.02	33.44	38.62	44.27	50.54	85.31	89.46	92.05	96.36	96.36	96.36	96.36
G.K.L.MI	28.67	34.65	43.30	48.69	32.16	39.35	44.87	47.96	85.66	90.99	92.14	95.16	95.16	95.16	95.16

Table 5: Test accuracy of a linear SVM trained on a different number of selected features {10, 25, 50, 100} on the three data sets.

	CIFAR					STL					INRIA				
	10	25	50	100	100	10	25	50	100	100	10	25	50	100	
SVMLRBF	29.11	39.22	46.05	54.68	34.71	40.13	43.87	45.77	92.44	93.55	93.38	92.97	92.97	92.97	92.97
Fisher	40.48	51.15	57.73	64.26	38.86	43.35	46.06	47.20	88.29	93.91	92.60	95.66	95.66	95.66	95.66
FCBF	41.80	51.97	57.31	62.14	38.39	44.87	47.02	48.92	80.07	79.99	88.89	90.48	90.48	90.48	90.48
MIRMR	27.16	38.23	47.60	54.70	32.53	41.27	43.22	44.88	92.78	93.16	93.02	93.25	93.25	93.25	93.25
χ^2	36.06	49.83	60.32	64.97	32.29	38.26	43.29	47.15	82.82	86.05	87.39	87.14	87.14	87.14	87.14
SBMLR	28.68	35.75	41.89	49.13	34.30	38.73	44.30	45.90	80.01	87.00	87.11	87.32	87.32	87.32	87.32
tTest	29.21	38.68	43.92	53.94	35.57	41.23	42.92	45.12	92.28	92.71	93.01	93.38	93.38	93.38	93.38
InfoGain	22.89	30.92	40.41	49.75	24.80	32.91	40.11	43.70	92.67	93.09	92.85	93.29	93.29	93.29	93.29
Spectral	29.49	37.08	45.39	53.96	38.22	42.36	47.27	50.35	90.62	94.56	95.05	95.20	95.20	95.20	95.20
RelieFF	35.50	43.74	50.98	61.01	35.32	42.72	47.46	49.82	88.34	91.31	95.44	97.14	97.14	97.14	97.14
CFS	35.50	43.74	50.98	61.01	35.32	42.72	47.46	49.82	88.34	91.31	95.44	97.14	97.14	97.14	97.14
CMTF	23.90	36.74	45.51	52.86	31.80	36.94	38.06	39.65	80.01	83.72	92.55	93.68	93.68	93.68	93.68
GBFS	34.98	45.07	54.70	61.27	33.65	43.99	49.04	51.52	93.00	95.25	95.83	96.48	96.48	96.48	96.48
G.C.E	35.29	51.12	60.34	65.76	36.16	42.64	45.37	47.79	87.73	87.67	91.96	93.13	93.13	93.13	93.13
G.C.MI	39.57	49.91	57.79	64.32	35.86	43.35	45.80	47.81	94.26	94.17	94.44	95.76	95.76	95.76	95.76
G.K.L.E	39.84	52.80	60.94	65.64	39.67	46.31	50.06	52.89	86.01	88.94	92.79	95.43	95.43	95.43	95.43
G.K.L.MI	34.49	43.09	51.48	56.54	35.95	41.65	45.27	45.86	91.03	91.91	93.36	93.98	93.98	93.98	93.98

Table 6: Test accuracy of a SVM with a RBF kernel when trained on a different number of selected features {10, 25, 50, 100} on the three data sets.

	CIFAR					STL					INRIA				
	10	25	50	100	100	10	25	50	100	100	10	25	50	100	
QDA	25.41	33.31	39.67	47.53	34.73	39.91	44.24	48.35	87.41	88.63	89.17	91.31	91.31	91.31	91.31
Fisher	35.02	43.97	52.32	58.99	37.44	41.89	45.70	48.89	89.95	94.00	94.00	94.00	94.00	94.00	94.00
FCBF	36.19	44.54	48.22	53.88	36.89	42.84	46.44	49.90	62.98	76.56	86.84	90.60	90.60	90.60	90.60
MIRMR	21.81	31.85	39.39	47.75	32.71	40.45	43.64	47.25	87.85	88.20	89.30	91.75	91.75	91.75	91.75
χ^2	31.71	43.46	53.31	58.86	36.89	45.26	49.58	51.65	76.30	80.36	81.00	81.49	81.49	81.49	81.49
SBMLR	26.34	33.39	39.16	45.33	33.92	40.09	45.05	47.63	76.16	82.50	82.85	85.23	85.23	85.23	85.23
tTest	22.38	31.61	37.65	46.47	34.17	40.94	44.51	47.81	87.99	88.08	89.49	91.77	91.77	91.77	91.77
InfoGain	17.97	24.80	34.99	44.25	25.39	35.45	44.39	49.68	87.99	88.26	89.07	91.26	91.26	91.26	91.26
S															

(see theorem 5.39 in (Vershynin, 2012)) that in the case of a matrix with (sub)-Gaussian generated rows, we have $\forall t \geq 0$ with probability at least $1 - 2e^{-ct^2}$

$$\|\hat{\Sigma}_N - \Sigma\| \leq \max(\delta, \delta^2),$$

where

$$\delta = \frac{C\sqrt{D} + t}{\sqrt{N}}$$

and c, C are constants related to the sub-Gaussian norm of the rows of P . Replacing t with $C't\sqrt{D}$ in the above (see Corollary 5.50 in Vershynin (2012)), we have, for sufficiently large $C', \forall \epsilon \in (0, 1)$, and $\forall t \geq 1$ with probability at least $1 - 2e^{-ct^2}$

$$\text{If } N \geq C(t/\epsilon)^2 D \text{ then } \|\hat{\Sigma}_N - \Sigma\| \leq \epsilon.$$

Thus $N = O(D)$ samples are needed to sufficiently approximate the covariance matrix by the finite sample covariance matrix when the underlying distribution is sub-Gaussian, compared to $O(D \log D)$ for an arbitrary distribution (Corollary 5.52 (Vershynin, 2012)).

Based in this, in order to select d features, the proposed methods theoretically require $O(d)$ samples. This however assumes that the feature selection methods depend solely on the covariance matrices Σ . This holds true only for the GC-approximation in section 3.2.1, the KL-based bound depends both on Σ and Σ^{-1} . Furthermore, as shown in section 3.4, the more efficient implementation of the GC-approximation also depends on Σ^{-1} .

Unfortunately, estimating the precision matrix by taking the inverse of the sample covariance matrix is known to be unstable (Cai et al., 2016). Though a number of methods have been proposed to address this issue (Cai et al., 2011), their complexity makes them unsuitable in the present setting. To investigate whether this instability affects the performance of the proposed methods, we present in the following an empirical analysis of the effect of sample size on prediction performance.

5.5.1 EMPIRICAL EVALUATION

In order to assess the influence of sample set size on performance, we consider the accuracy on the test set of a linear SVM. Specifically we perform feature selection using a subset of the training data by selecting uniformly at random without replacement. Thus the relevant sample covariance matrices are estimated using these smaller sets. We then train a linear SVM using the selected subset of features but using the entire set of samples; this is done to avoid any influence of sample size on the training of the SVM and by extension on the final results.

In figure 8 we show the empirical results on the three data sets (STL, CIFAR, INRIA) in the case where feature selection is performed using the GC-approximation. As can be seen, in the case of the STL and CIFAR data sets the feature selection method proves to be very robust with regards to sample set size; performance degrades only slightly when the sample set size is very small (50 samples per class). On the contrary, in the case of the INRIA data set, we see that the method does not prove to be so robust and the performance suffers. We note that in the plot for the INRIA data set, the x-axis relates the number of samples in the positive class, the number of samples sampled from the negative class was chosen to preserve the class ratio ($\sim 6/1$).

Similarly in figure 9 we show results for the KL-bound case. Here we see that sample size is more influential. A possible explanation is that though the feature selection method arising from the GC-approximation involves the estimation of $|Y| + 1$ inverse covariance matrices, in the case of the KL-bound feature selection method involves $2|Y|$ inverses. Examining the plots in figure 9, we see that in the case of the STL and CIFAR data sets there is some degradation of performance for small sample set sizes, though the performance quickly reaches that of the full set as the subset size increases. In the case of the INRIA data set however we see that the method performs considerably worse when only a subset of the data set is used to perform feature selection.

6. Conclusion

The present work concentrates on developing tractable algorithms to exploit information theoretic criteria for feature selection. The proposed methods focus on feature selection in the context of classification and demonstrate that it is possible to choose features that are jointly informative by careful density modeling and algorithmic implementation. Thus the joint mutual information of variables can in fact be employed efficiently for feature selection as opposed to using only the mutual information of marginal or pairwise distributions as has typically been used in the literature.

The proposed methods rely on modeling the conditional joint distributions of the features given the class to predict and subsequently maximizing either upper bounds on the information theoretic measures or relevant approximations. To reduce the computational cost of a forward feature selection scheme incorporating these criteria, we have proposed efficient implementations for both approaches, so that they are competitive with other state-of-the-art methods in terms of speed. We have also presented a novel method for updating the eigen-decompositions of a specific family of matrices (and updates) which our GC-approximation feature selection algorithm exploits.

Empirical results show the methods to be competitive with current state-of-the-art with respect to prediction accuracy. Furthermore, an empirical analysis of the performance of these methods in connection with the number of samples in the data sets has shown them to be relatively robust in this respect.

Acknowledgments

This work was supported by the Haeler Foundation through the MASH-2 project.

References

- Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.
- Renno R. Bouckaert, Eibe Frank, Mark A. Hall, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. WEKA—Experiences with a Java open-source project. *Journal Machine Learning Research (JMLR)*, 11:2533–2541, 2010.

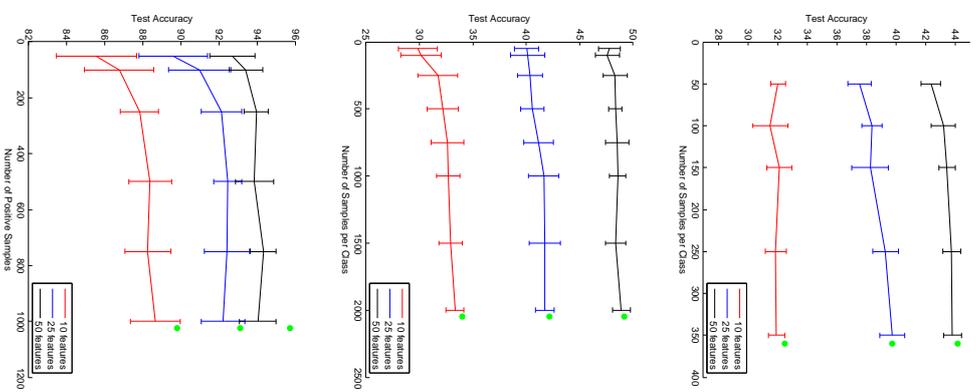


Figure 8: Effect of sample set size on performance when using the GC-approximation for the top) STL, middle) CIFAR, and bottom) INRIA data sets. The green circles mark the performance of the method when the entire data set is used.

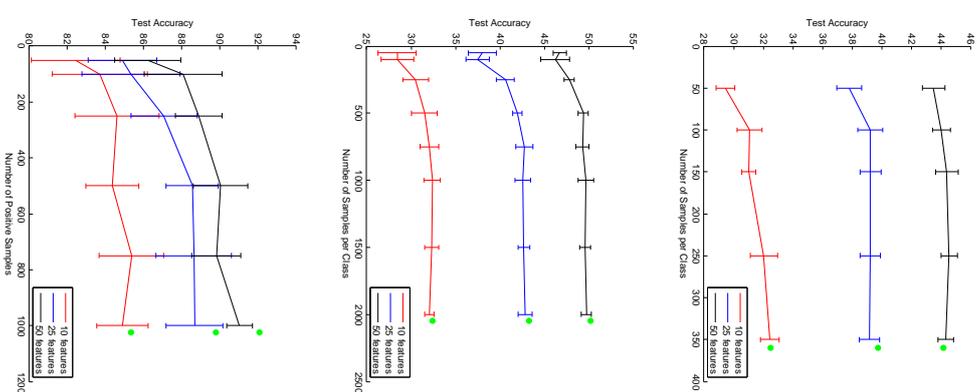


Figure 9: Effect of sample set size on performance when using the KL-bound for the top) STL, middle) CIFAR, and bottom) INRIA data sets. The green circles mark the performance of the method when the entire data set is used.

- T. Cai, W. Liu, and H. Zhou. Estimating sparse precision matrix: optimal rates of convergence and adaptive estimation. *Annals of Statistics*, 44:455–488, 2016.
- T.T. Cai, W. Liu, and X. Luo. A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 494:594–607, 2011.
- Gavin C. Cawley, Nicola L. C. Talbot, and Mark Girolami. Sparse multinomial logistic regression via Bayesian ℓ_1 regularisation. In *Proceedings of Neural Information Processing Systems (NIPS)*, pages 209–216, 2006.
- A. Coates and A. Ng. The importance of encoding versus training with sparse coding and vector quantization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 921–928, 2011.
- N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893, 2005.
- A. Das and D. Kempe. Submodular meets spectral: greedy algorithms for subset selection, sparse approximation and dictionary selection. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1057–1064, 2011.
- S. Das. Filters, wrappers and a boosting-based hybrid for feature selection. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 74–81, 2001.
- F. Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research (JMLR)*, 5:1531–1555, 2004.
- Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research (JMLR)*, 3:1157–1182, 2003. ISSN 1532-4435.
- Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.
- Mark A. Hall and Lloyd A. Smith. Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper. In *Proceedings of the International Florida Artificial Intelligence Research Society Conference*, pages 235–239, 1999.
- J.R. Hershey and P.A. Olsen. Approximating the kullback leibler divergence between Gaussian mixture models. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages IV–317–IV–320, 2007.
- G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.
- Yi Jiang and Jiangtao Ren. Eigenvector sensitive feature selection for spectral clustering. In *Proceedings of the European Conference on Machine Learning (ECML)*, pages 114–129, 2011.
- L. Lefakis and F. Fleuret. Jointly informative feature selection. In *Proceedings of the international conference on Artificial Intelligence and Statistics (AISTATS)*, 2014.
- H. Liu and L. Yu. Feature selection for high-dimensional data: a fast correlation-based filter solution. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 856–863, 2003.
- Dimitris Margaritis. Toward provably correct feature selection in arbitrary domains. In *Proceedings of Neural Information Processing Systems (NIPS)*, pages 1240–1248, 2009.
- H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 27(8):1226–1238, 2005.
- Marko Robnik-Šikonja and Igor Kononenko. Theoretical and empirical analysis of ReliefF and RReliefF. *Machine Learning*, 53(1-2):23–69, 2003.
- Irene Rodriguez-Lujan, Ramon Huerta, Charles Elkan, and Carlos Santa Cruz. Quadratic programming feature selection. *The Journal of Machine Learning Research (JMLR)*, 11:1491–1516, 2010.
- Le Song, Alex Smola, Arthur Gretton, Justin Bedo, and Karsten Borgwardt. Feature selection via dependence maximization. *Journal of Machine Learning Research*, 13(1):1393–1434, 2012.
- Mingkui Tan, Li Wang, and Ivor W. Tsang. Learning sparse SVM for feature selection on very high dimensional datasets. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1047–1054, 2010.
- Kari Torkkola. Feature extraction by non-parametric mutual information maximization. *Journal of Machine Learning Research (JMLR)*, 3:1415–1438, 2003.
- Nuno Vasconcelos. Feature selection by maximum marginal diversity: optimality and implications for visual recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.
- R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. *Compressed Sensing*, pages 210–268, 2012.
- Lior Wolf and Amnon Shashua. Feature selection for unsupervised and supervised inference: the emergence of sparsity in a weight-based approach. *Journal of Machine Learning Research (JMLR)*, 6:1855–1887, 2005.
- Zhixiang Xu, Gao Huang, Kilian Q Weinberger, and Alice X Zheng. Gradient boosted feature selection. In *Proceedings of the international conference on Knowledge discovery and data mining (SIGKDD)*, pages 522–531. ACM, 2014.
- Makoto Yamada, Wittawat Jitkrittum, Leonid Sigal, Eric P Xing, and Masashi Sugiyama. High-dimensional feature selection by feature-wise kernelized lasso. *Neural computation*, 26(1):185–207, 2014.

Zheng Zhao and Huan Liu. Spectral feature selection for supervised and unsupervised learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1151–1157, 2007.

Learning with Differential Privacy: Stability, Learnability and the Sufficiency and Necessity of ERM Principle

Yu-Xiang Wang^{1,2}

Jing Lei²

Stephen E. Fienberg^{1,2}

¹ *Machine Learning Department, Carnegie Mellon University, Pittsburgh, PA 15213*

² *Department of Statistics, Carnegie Mellon University, Pittsburgh, PA 15213*

YUXIANGW@CS.CMU.EDU

JINGLEI@ANDREW.CMU.EDU

FIENBERG@STAT.CMU.EDU

PA 15213

PA 15213

Editor: Moritz Hardt

Abstract

While machine learning has proven to be a powerful data-driven solution to many real-life problems, its use in sensitive domains has been limited due to privacy concerns. A popular approach known as *differential privacy* offers provable privacy guarantees, but it is often observed in practice that it could substantially hamper learning accuracy. In this paper we study the learnability (whether a problem can be learned by any algorithm) under Vapnik’s general learning setting with differential privacy constraint, and reveal some intricate relationships between privacy, stability and learnability. In particular, we show that a problem is privately learnable *if and only if* there is a private algorithm that asymptotically minimizes the empirical risk (AERM). In contrast, for non-private learning AERM alone is not sufficient for learnability. This result suggests that when searching for private learning algorithms, we can restrict the search to algorithms that are AERM. In light of this, we propose a conceptual procedure that always finds a universally consistent algorithm whenever the problem is learnable under privacy constraint. We also propose a generic and practical algorithm and show that under very general conditions it privately learns a wide class of learning problems. Lastly, we extend some of the results to the more practical (ϵ, δ) -differential privacy and establish the existence of a phase-transition on the class of problems that are approximately privately learnable with respect to how small δ needs to be.

Keywords: *differential privacy, learnability, characterization, stability, privacy-preserving machine learning*

1. Introduction

Increasing public concerns regarding data privacy have posed obstacles in the development and application of new machine learning methods as data collectors and curators may no longer be able to share data for research purposes. In addition to addressing the original goal of information extraction, privacy-preserving learning also requires the learning procedure to protect sensitive information of individual data entries. For example, the second Netflix Prize competition was canceled in response to a lawsuit and Federal Trade Commission privacy concerns, and the National Institute of Health decided in August 2008 to remove

aggregate Genome-Wide Association Studies (GWAS) data from the public web site, after learning about a potential privacy risk.

A major challenge in developing privacy-preserving learning methods is to quantify formally the amount of privacy leakage, given all possible and unknown auxiliary information the attacker may have, a challenge in part addressed by the notion of *differential privacy* (Dwork, 2006; Dwork et al., 2006b). Differential privacy has three main advantages over other approaches: (1) it rigorously quantifies the privacy property of any data analysis mechanism; (2) it controls the amount of privacy leakage regardless of the attacker’s resource or knowledge; (3) it has useful interpretations from the perspectives of Bayesian inference and statistical hypothesis testing, and hence fits naturally in the general framework of statistical machine learning, e.g., see (Dwork and Lei, 2009; Wasserman and Zhou, 2010; Smith, 2011; Lei, 2011; Wang et al., 2015), as well as applications involving regression (Chaudhuri et al., 2011; Thakurta and Smith, 2013) and GWAS data (Yu et al., 2014), etc.

In this paper we focus on the following fundamental question about differential privacy and machine learning: *What problems can we learn with differential privacy?* Most literature focuses on designing differentially private extensions of various learning algorithms, where the methods depend crucially on the specific context and differ vastly in nature. But with the privacy constraint, we have less choice in developing learning and data analysis algorithms. It remains unclear how such a constraint affects our ability to learn, and if it is possible to design a generic privacy-preserving analysis mechanism that is applicable to a wide class of learning problems.

Our Contributions We provide a general answer to the relationship between learnability and differential privacy under Vapnik’s General Learning Setting (Vapnik, 1995) in four aspects.

1. We characterize the subset of problems in the General Learning Setting that can be learned under differential privacy. Specifically, we show that a sufficient and necessary condition for a problem to be privately learnable is the existence of an algorithm that is differentially private and asymptotically minimizes the empirical risk. This characterization generalizes previous studies of the subject (Kasiviswanathan et al., 2011; Beimel et al., 2013a) that focus on binary classification in discrete domain under the PAC learning model. Technically, the result relies on the now well-known intuitive observation that “privacy implies algorithmic stability” and the argument in Shalev-Shwartz et al. (2010) that shows a variant of algorithmic stability is necessary for learnability.
2. We also introduce a weaker notion of learnability, which only requires consistency for a class of distributions \mathcal{D} . Problems that are not privately learnable (a surprisingly large class that includes simple problems such as 0-1 loss binary classification in continuous feature domain (Chaudhuri and Hsu, 2011)) are usually private \mathcal{D} -learnable for some “nice” distribution class \mathcal{D} . We characterize the subset of private \mathcal{D} -learnable problems that are also (non-privately) learnable using conditions analogous to those in distribution-free private learning.

3. Inspired by the equivalence between privacy learnability and private AERM, we propose a generic (but impractical) procedure that always finds a consistent and private algorithm for any privately learnable (or \mathcal{D} -learnable) problems. We also study a specific algorithm that aims at minimizing the empirical risk while preserving the privacy. We show that under a sufficient condition that relies on the geometry of the hypothesis space and the data distribution, this algorithm is able to privately learn (or \mathcal{D} -learn) a large range of learning problems including classification, regression, clustering, density estimation and etc, and it is computationally efficient when the problem is convex. In fact, this generic learning algorithm learns any privately learnable problems in the PAC learning setting (Beimel et al., 2013a). It remains an open problem whether the second algorithm also learns any privately learnable problem in the General Learning Setting.

4. Lastly, we provide a preliminary study of learnability under the more practical (ϵ, δ) -differential privacy. Our results reveal that whether there is separation between learnability and approximate private learnability depends on how fast δ is required to go to 0 with respect to the size of the data. Finding where the exact phase transition occurs is an open problem of future interest.

Our primary objective is to understand the conceptual impact of differential privacy and learnability under a general framework and the rates of convergence obtained in the analysis may be suboptimal. Although we do provide some discussion on polynomial time approximations to the proposed algorithm, learnability under computational constraints is beyond the scope of this paper.

Related work While a large amount of work has been devoted to finding consistent (and rate optimal) differentially private learning algorithms in various settings (e.g., Chaudhuri et al., 2011; Kifer et al., 2012; Jain and Thakurta, 2013; Bassily et al., 2014), the characterization of privately learnable problems were only studied in a few special cases.

Kasiviswanathan et al. (2011) showed that, for binary classification with a finite discrete hypothesis space, anything that is non-privately learnable is privately learnable under the agnostic Probably Approximately Correct (PAC) learning framework, therefore “finite VC-dimension” characterizes the set of private learnable problems in this setting. Beimel et al. (2013a) extends Kasiviswanathan et al. (2011) by characterizing the sample complexity of the same class of problems, but the result only applies to the realizable (non-agnostic) case. Chaudhuri and Hsu (2011) provided a counter-example showing that for continuous hypothesis space and data space, there is a gap between learnability and learnability under privacy constraint. They proposed to fix this issue by either weakening the privacy requirement to labels only or by restricting the class of potential distribution. While meaningful in some cases, these approaches do not resolve the learnability problem in general.

A key difference of our work from Kasiviswanathan et al. (2011); Chaudhuri and Hsu (2011); Beimel et al. (2013a) is that we consider a more general class of learning problems and provide a proper treatment in a statistical learning framework. This allows us to capture a wider collection of important learning problems (see Figure 1(a) and Table 1).

It is important to note that despite its generality, Yagnik’s general learning setting still does not nearly cover the full spectrum of private learning. In particular, our results do not apply to improper learning (learning using a different hypothesis class) as considered in Beimel et al. (2013a) or to structural loss minimization (the loss function jointly take all data points as input) considered in Beimel et al. (2013b). Also, our results do not address the sample complexity problem, which remains open in the general learning setting even for learning without privacy constraints.

Our characterization of private learnability (and private \mathcal{D} -learnability) in Section 3 uses a recent advance in the characterization of general learnability given by Shalev-Shwartz et al. (2010). Roughly speaking, they showed that a problem is learnable if and only if there exists an algorithm that (i) is stable under small perturbation of training data, and (ii) behaves like empirical risk minimization (ERM) asymptotically. We also makes use of a folklore observation that “Privacy \Rightarrow Stability \Rightarrow Generalization”. The connection of privacy and stability appeared as early as 2008 in a conference version of Kasiviswanathan et al. (2011). Further connection to “generalization” recently appeared in blog posts¹, stated as a theorem in Appendix F of Bassily et al. (2014), and was shown to hold with strong concentration in Dwork et al. (2015b).

Dwork et al. (2015b) is part of an independent line of work (Hardt and Ullman, 2014; Bassily et al., 2015; Dwork et al., 2015a; Blum and Hardt, 2015) on adaptive data analysis, which also stems from the observation that privacy implies stability and generalization. Comparing to adaptive data analysis works, our focus is quite different. Adaptive data analysis work focus on the impact of k on how fast the maximum absolute error of k -adaptively chosen queries goes to 0 as a function of n , while this paper is concerned with whether the error can go to 0 at all for each learning problem when we require the learning algorithm be differentially private with $\epsilon < \infty$. Nonetheless, we acknowledge that Theorem 7 in Dwork et al. (2015b) provides an interesting alternative proof for “differentially private learners have small generalization error”, when choosing the statistical query as evaluating a loss function at a privately learned hypothesis. The connection is not quite obvious and we provide a more detailed explanation in Appendix B.

The main tool used in the construction of our generic private learning algorithm in Section 4 is the Exponential Mechanism (McSherry and Talwar, 2007), which provides a simple and differentially-private approximation to the maximizer of a score function among a candidate set. In the general learning context, we use the negative empirical risk as the utility function, and apply the exponential mechanism to a possibly pre-discretized hypothesis space. This exponential mechanism approach was used in Bassily et al. (2014) for minimizing convex and Lipschitz functions. The sample discretization procedure has been considered in Chaudhuri and Hsu (2011) and Beimel et al. (2013a). Our scope and proof techniques are different. Our strategy is to show that, under some general regularity conditions, the exponential mechanism is stable and behaves like ERM. Our sublevel set condition has the same flavor

1. For instance, Frank McSherry described in a blog post an example of exploiting differential privacy for measure concentration <http://windsontheory.org/2014/02/04/differential-privacy-for-measure-concentration/>; Moritz Hardt discussed the connection of differential privacy to stability and generalization in his blog post <http://blog.mrtz.org/2014/01/13/false-discovery/>.

as that in the proof of Bassily et al. (2014, Theorem 3.2), although we do not need the loss function to be convex or Lipschitz.

Stability, privacy and generalization were also studied in Thakurta and Smith (2013) with different notions of stability. More importantly, their stability is used as an assumption rather than a consequence, so their result is not directly comparable to ours.

2. Background

2.1 Learnability under the General Learning Setting

In the General Learning Setting of Vapnik (1995), a learning problem is characterized by a triplet $(\mathcal{Z}, \mathcal{H}, \ell)$. Here \mathcal{Z} is the sample space (with a σ -algebra). The hypothesis space \mathcal{H} is a collection of models such that each $h \in \mathcal{H}$ describes some structures of the data. The loss function $\ell : \mathcal{H} \times \mathcal{Z} \rightarrow \mathbb{R}$ measures how well the hypothesis h explains the data instance $z \in \mathcal{Z}$. For example, in supervised learning problems $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ where \mathcal{X} is the feature space and \mathcal{Y} is the label space; \mathcal{H} defines a collection of mapping $h : \mathcal{X} \rightarrow \mathcal{Y}$; and $\ell(h, z)$ measures how well h predicts the feature-label relationship $z = (x, y) \in \mathcal{Z}$. This setting includes problems with continuous input/output in potentially infinite dimensional spaces (e.g. RKHS methods), hence is much more general than PAC learning. In addition, the general learning setting also covers a variety of unsupervised learning problems, including clustering, density estimation, principal component analysis (PCA) and variants (e.g. Sparse PCA, Robust PCA), dictionary learning, matrix factorization and even Latent Dirichlet Allocation (LDA). Details of these examples are given in Table 1 (the first few are extracted from Shalev-Shwartz et al. (2010)).

To account for the randomness in the data, we are primarily interested in the case where the data $Z = \{z_1, \dots, z_n\} \in \mathcal{Z}^n$ are independent samples drawn from an unknown probability distribution \mathcal{D} on \mathcal{Z} . We denote such a random sample by $Z \sim \mathcal{D}^n$. For a given distribution \mathcal{D} , let $R(h)$ be the expected loss of hypothesis h and $\hat{R}(h, Z)$ the empirical risk from a sample $Z \in \mathcal{Z}^n$.

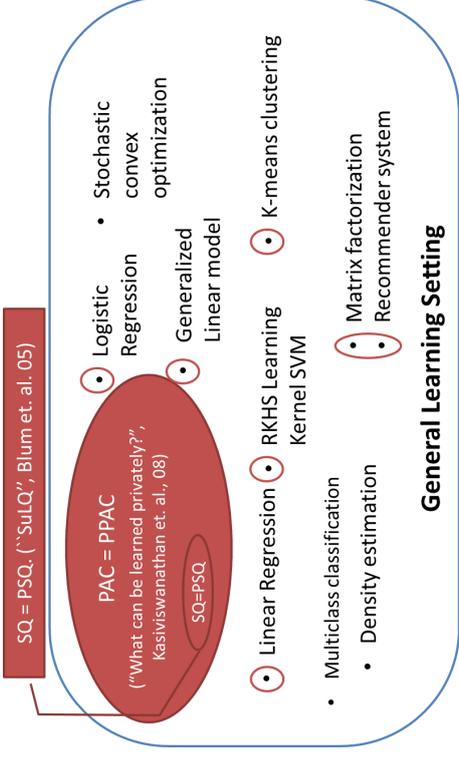
$$R(h) = \mathbb{E}_{z \sim \mathcal{D}} \ell(h, z), \quad \hat{R}(h, Z) = \frac{1}{n} \sum_{i=1}^n \ell(h, z_i).$$

The optimal risk $R^* = \inf_{h \in \mathcal{H}} R(h)$ and we assume that it is achieved by an optimal $h^* \in \mathcal{H}$. Similarly, the minimal empirical risk $\hat{R}^*(Z) = \inf_{h \in \mathcal{H}} \hat{R}(h, Z)$ is achieved by $\hat{h}^*(Z) \in \mathcal{H}$. For a possibly randomized algorithm $\mathcal{A} : \mathcal{Z}^n \rightarrow \mathcal{H}$ that learns some hypothesis $\mathcal{A}(Z) \in \mathcal{H}$ given data sample Z , we say \mathcal{A} is *consistent* if

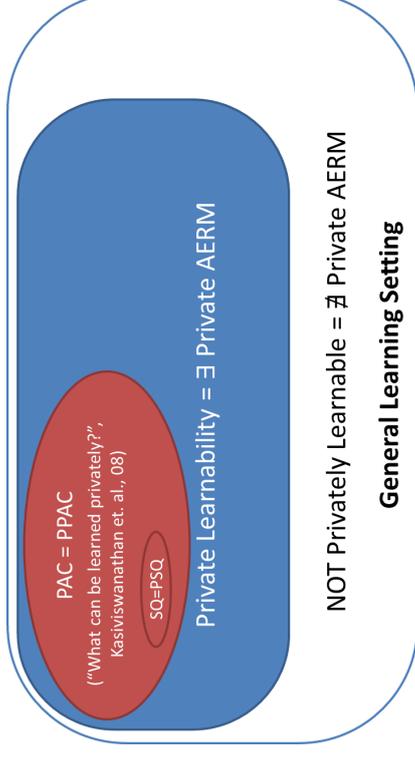
$$\lim_{n \rightarrow \infty} \mathbb{E}_{Z \sim \mathcal{D}^n} (\mathbb{E}_{h \sim \mathcal{A}(Z)} R(h) - R^*) = 0. \quad (1)$$

In addition, we say \mathcal{A} is consistent with rate $\xi(n)$ if

$$\mathbb{E}_{Z \sim \mathcal{D}^n} (\mathbb{E}_{h \sim \mathcal{A}(Z)} R(h) - R^*) \leq \xi(n), \quad \text{where } \lim_{n \rightarrow \infty} \xi(n) \rightarrow 0. \quad (2)$$



(a) Illustration of general learning setting. Examples of known DP extensions are circled in maroon.



(b) Our characterization of private learnable problems in the general learning setting (in blue).

Figure 1: The Big Picture: illustration of general learning setting and our contribution in understanding differentially private learnability.

Problem	Hypothesis class \mathcal{H}	\mathcal{Z} or $\mathcal{X} \times \mathcal{Y}$	Loss function ℓ
Binary classification	$\mathcal{H} \subset \{f : \{0, 1\}^d \rightarrow \{0, 1\}\}$	$\{0, 1\}^d \times \{0, 1\}$	$1(\ell(x) \neq y)$
Regression	$\mathcal{H} \subset \{f : [0, 1]^d \rightarrow \mathbb{R}\}$	$[0, 1]^d \times \mathbb{R}$	$ h(x) - y ^2$
Density Estimation	Bounded distributions on \mathcal{Z}	$\mathcal{Z} \subset \mathbb{R}^d$	$-\log(\ell(z))$
K-means Clustering	$\{S \subset \mathbb{R}^d : S = k\}$	$\mathcal{Z} \subset \mathbb{R}^d$	$\min_c \ c - z\ ^2$
RKHS classification	Bounded RKHS	$\text{RKHS} \times \{0, 1\}$	$\max_{\{0, 1\}} 1 - y \langle x, h \rangle $
RKHS regression	Bounded RKHS	\mathbb{R}^d	$ \langle x, h \rangle - y ^2$
Sparse PCA	Rank- r projection matrices	\mathbb{R}^d	$\ hz - z\ ^2 + \lambda \ h\ _1$
Robust PCA	All subspaces in \mathbb{R}^d	\mathbb{R}^d	$\ P_h(z) - z\ _1 + \lambda \text{rank}(h)$
Matrix Completion	All subspaces in \mathbb{R}^d	$\mathbb{R}^d \times \{1, 0\}^d$	$\min_{h \in \mathcal{H}} \ y \circ (\theta - x)\ ^2 + \lambda \text{rank}(h)$
Dictionary Learning	All dictionaries $\in \mathbb{R}^{d \times r}$	\mathbb{R}^d	$\min_{h \in \mathcal{H}} \ hb - z\ ^2 + \lambda \ h\ _1$
Non-negative MF	All dictionaries $\in \mathbb{R}^{d \times r}$	\mathbb{R}^d	$\min_{h \in \mathcal{H}} \ hb - z\ ^2$
Subspace Clustering	A set of k rank- r subspaces	\mathbb{R}^d	$\min_{h \in \mathcal{H}} \ P_h(z) - z\ ^2$
Topic models (LDA)	$\{\mathbb{P}(\text{word} \text{topic})\}$	Documents	$-\max_{h \in \{\text{Topic}\}} \sum_{w \in \mathcal{Z}} \log \mathbb{P}_{h,h}(w)$

Table 1: An illustration of problems in the General Learning setting.

Since the distribution \mathcal{D} is unknown, we cannot adapt the algorithm \mathcal{A} to \mathcal{D} , especially when privacy is a concern. Also, even if \mathcal{A} is pointwise consistent for any distribution \mathcal{D} , it may have different rates for different \mathcal{D} and potentially be arbitrarily slow for some \mathcal{D} . This makes it hard to evaluate whether \mathcal{A} indeed learns the learning problem and forbids the study of the learnability problem. In this study, we adopt the stronger notion of learnability considered in Shalev-Shwartz et al. (2010), which is a direct generalization of PAC-learnability (Valiant, 1984) and agnostic PAC-learnability (Kearns et al., 1992) to the General Learning Setting as studied by Haussler (1992).

Definition 1 (Learnability, Shalev-Shwartz et al., 2010) *A learning problem is learnable if there exists an algorithm \mathcal{A} and rate $\xi(n)$, such that \mathcal{A} is consistent with rate $\xi(n)$ for any distribution \mathcal{D} defined on \mathcal{Z} .*

This definition requires consistency to hold universally for any distribution \mathcal{D} with a uniform (distribution-independent) rate $\xi(n)$. This type of problem is often called *distribution-free learning* (Valiant, 1984), and an algorithm is said to be *universally consistent* with rate $\xi(n)$ if it realizes the criterion.

2.2 Differential privacy

Differential privacy requires that if we arbitrarily perturb a database by only one data point, the output should not differ much. Therefore, if one conducts a statistical test for whether any individual is in the database or not, the false positive and false negative probabilities cannot both be small (Wasserman and Zhou, 2010). Formally, define “Hamming distance”

$$d(Z, Z') := \#\{i = 1, \dots, n : z_i \neq z'_i\}. \quad (3)$$

Definition 2 (ϵ -Differential Privacy, Dwork, 2006) *An algorithm \mathcal{A} is ϵ -differentially private, if*

$$\mathbb{P}(\mathcal{A}(Z) \in H) \leq \exp(\epsilon) \mathbb{P}(\mathcal{A}(Z') \in H)$$

for $\forall Z, Z'$ obeying $d(Z, Z') = 1$ and any measurable subset $H \subseteq \mathcal{H}$.

There are weaker notions of differential privacy. For example (ϵ, δ) -differential privacy allows for a small probability δ where the privacy guarantee does not hold. In this paper, we will mainly work with the stronger ϵ -differential privacy. In Section 6 we discuss the problem of (ϵ, δ) -differential privacy and extend some of the results to this setting.

Our objective is to understand whether there is a gap between learnable problems and privately learnable problems in the general learning setting, and to quantify the tradeoff required to protect privacy. To achieve this objective, we need to show the existence of an algorithm that learns a class of problems while preserving differential privacy. More formally, we define

Definition 3 (Private learnability) *A learning problem is privately learnable with rate $\xi(n)$ if there exists an algorithm \mathcal{A} that satisfies both universal consistency (as in Definition 1) with rate $\xi(n)$ and ϵ -differential privacy with privacy parameter $\epsilon < \infty$.*

We can view the consistency requirement Definition 3 as a measure of utility. This utility is not a function of the observed data, however, but rather how the results generalize to unseen data.

The following lemma shows that the above definition of private learnability is actually equivalent to a seemingly much stronger condition with a vanishing privacy loss ϵ .

Lemma 4 *If there is an ϵ -DP algorithm that is consistent with rate $\xi(n)$ for some constant $0 < \epsilon < \infty$, then there is a $\frac{\epsilon}{\sqrt{n}}$ ($\epsilon - \epsilon^{-\gamma}$)-DP algorithm that is consistent with rate $\xi(\sqrt{n})$.*

The proof, given in Appendix A.1, uses a subsampling theorem adapted from Beimel et al. (2014, Lemma 4.4).

There are many approaches to design differentially private algorithms, such as noise perturbation using Laplace noise (Dwork, 2006; Dwork et al., 2006b) and the Exponential Mechanism (McSherry and Talwar, 2007). Our construction of generic differentially private learning algorithms applies the Exponential Mechanism to penalized empirical risk minimization. Our argument will make use of a general characterization of learnability described below.

2.3 Stability and Asymptotic ERM

An important breakthrough in learning theory is a full characterization of all learnable problems in the General Learning Setting in terms of stability and empirical risk minimization (Shalev-Shwartz et al., 2010). Without assuming uniform convergence of empirical risk, Shalev-Shwartz et al. showed that a problem is learnable if and only if there exists a “strongly uniform-RO stable” and “always asymptotically empirical risk minimization” (Always AERM) randomized algorithm that learns the problem. Here “RO” stands for “replace one”. Also,

any strongly uniform-RO stable and “universally” AERM (weaker than “always” AERM) learning rule learns the problem consistently. Here we give detailed definitions.

Definition 5 (Universally/Always AERM, Shalev-Shwartz et al., 2010) A (possibly randomized) learning rule \mathcal{A} is *Universally AERM* if for any distribution \mathcal{D} defined on domain \mathcal{Z}

$$\mathbb{E}_{Z \sim \mathcal{D}^n} [\mathbb{E}_{h \sim \mathcal{A}(Z)} \hat{R}(h, Z) - \hat{R}^*(Z)] \rightarrow 0, \text{ as } n \rightarrow \infty$$

where $\hat{R}^*(Z)$ is the minimum empirical risk for data set Z . We say \mathcal{A} is *Always AERM*, if in addition,

$$\sup_{Z \in \mathcal{Z}^n} \mathbb{E}_{h \sim \mathcal{A}(Z)} \hat{R}(h, Z) - \hat{R}^*(Z) \rightarrow 0, \text{ as } n \rightarrow \infty.$$

Definition 6 (Strongly Uniform RO-Stability, Shalev-Shwartz et al., 2010) An algorithm \mathcal{A} is *strongly uniform RO-stable* if

$$\sup_{z \in \mathcal{Z}} \sup_{\substack{z, z' \in \mathcal{Z}^n, \\ d(z, z') = 1}} |\mathbb{E}_{h \sim \mathcal{A}(Z)} \ell(h, z) - \mathbb{E}_{h \sim \mathcal{A}(Z')} \ell(h, z)| \rightarrow 0 \text{ as } n \rightarrow \infty.$$

where $d(Z, Z')$ is defined in (3), in other word, Z and Z' can differ by at most one data point.

Since we will not deal with other variants of algorithmic stability in this paper (e.g., hypothesis stability (Kearns and Ron, 1999), uniform stability (Bousquet and Elisseeff, 2002) and leave-one-out (LOO) stability in Mukherjee et al. (2006)), we simply call Definition 6 stability or uniform stability. Likewise, we will refer to ϵ -differential privacy as just “privacy” although there are several other notions of privacy in the literature.

3. Characterization of private learnability

We are now ready to state our main result. The only assumption we make is the uniform boundedness of the loss function. This is also assumed in Shalev-Shwartz et al. (2010) for the learnability problem without privacy constraints. Without loss of generality, we can assume $0 \leq \ell(h, z) \leq 1$.

Theorem 7 Given a learning problem $(\mathcal{Z}, \mathcal{H}, \ell)$, the following statements are equivalent.

1. The problem is *privately learnable*.
2. There exists a *differentially private universally AERM algorithm*.
3. There exists a *differentially private always AERM algorithm*.

The proof is simple yet revealing, we will present the arguments for $2 \Rightarrow 1$ (sufficiency of AERM) in Section 3.1 and $1 \Rightarrow 3$ (necessity of AERM) in Section 3.2. $3 \Rightarrow 2$ follows trivially from the definition of “always” and “universal” AERM.

The theorem says that we can stick to ERM-like algorithms for private learning, despite that ERM may fail for some problems in the (non-private) general learning setting (Shalev-Shwartz et al., 2010). Thus a standard procedure for finding universally consistent and

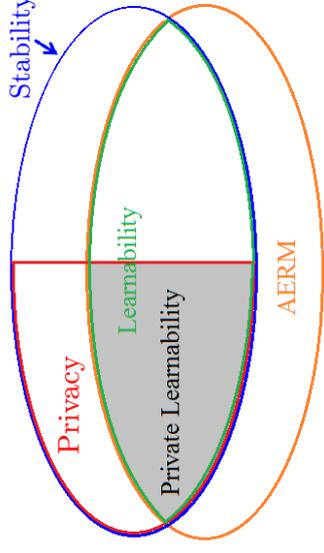


Figure 2: A summary of the relationships of various notions revealed by our analysis.

differentially private algorithms would be to approximately minimize the empirical risk using some differentially private procedures (Chaudhuri et al., 2011; Kifer et al., 2012; Bassily et al., 2014). If the utility analysis reveals that the method is AERM, we do not need to worry about generalization as it is guaranteed by privacy. This consistency analysis is considerably simpler than non-private learning problems where one typically needs to control generalization error either via uniform convergence (VC-dimension, Rademacher complexity, metric entropy, etc) or to adopt the stability argument (Shalev-Shwartz et al., 2010).

This result does not imply that privacy is helping the algorithm to learn in any sense, as the simplicity is achieved at the cost of having a smaller class of learnable problems. A concrete example of a problem being learnable but not privately learnable is given in (Chaudhuri and Hsu, 2011) and we will revisit it in Section 3.3. For some problems where ERM fails, it may not be possible to make it AERM while preserving privacy. In particular, we were not able to privatize the problem in Section 4.1 of Shalev-Shwartz et al. (2010).

To avoid any potential misunderstanding, we stress that Theorem 7 is a characterization of learnability, *not* learning algorithms. It does not prevent the existence of a universally consistent learning algorithm that is private but not AERM. Also, the characterization given in Theorem 7 is about consistency, and it does not claim anything on sample complexity. An algorithm that is AERM may be suboptimal in terms of convergence rate.

3.1 Sufficiency: Privacy implies stability

A key ingredient in the proof of sufficiency is a well-known heuristic observation that differential privacy by definition implies uniform stability, which is useful in its own right.

Lemma 8 (Privacy \Rightarrow Stability) Assume $0 \leq \ell(h, z) \leq 1$, any ϵ -differentially private algorithm satisfies $(\epsilon^\epsilon - 1)$ -stability. Moreover if $\epsilon \leq 1$ it satisfies 2ϵ -stability.

The proof of this lemma comes directly from the definition of differential privacy so it is algorithm independent. The converse, however, is not true in general (e.g., a non-trivial deterministic algorithm can be stable, but not differentially private.)

Corollary 9 (Privacy + Universal AERM \Rightarrow Consistency) *If a learning algorithm \mathcal{A} is $\epsilon(n)$ -differentially private and \mathcal{A} is universally AERM with rate $\xi(n)$, then \mathcal{A} is universally consistent with rate $\xi(n) + e^{\epsilon(n)} - 1 = O(\xi(n) + \epsilon(n))$.*

The proof of Corollary 9, provided in the Appendix, combines Lemma 8 and the fact that consistency is implied by stability and AERM (Theorem 28). Our Theorem 28 is based on minor modifications of Theorem 8 in Shalev-Shwartz et al. (2010). In fact, Corollary 9 can be stated in a stronger per distribution form, since universality is not used in the proof. We will revisit this point when we discuss a weaker notion of private learnability below.

Lemma 4 and Corollary 9 together establishes $2 \Rightarrow 1$ in Theorem 7.

If for a problem privacy and always AERM cannot coexist, then the problem is not privately learnable. This is what we will show next.

3.2 Necessity: Consistency implies Always AERM

To prove that the existence of an always AERM learning algorithm is necessary for any private learnable problems, it suffices to construct such a learning algorithm from or each learnable problem. any universally consistent learning algorithm.

Lemma 10 (Consistency + Privacy \Rightarrow Private Always AERM) *If \mathcal{A} is a universally consistent learning algorithm satisfying ϵ -DP with any $\epsilon > 0$ and consistent with rate $\xi(n)$, then there is another universally consistent learning algorithm \mathcal{A}' that is always AERM with rate $\xi(\sqrt{n})$ and satisfies $\frac{2}{\sqrt{n}}(\epsilon^e - e^{-\epsilon})$ -DP.*

Lemma 10 is proved in Appendix A.2. The proof idea is to run \mathcal{A} on a size $O(\sqrt{n})$ random subsample of Z , which will be universally consistent with a slower rate, differentially private with $\epsilon(n) \rightarrow 0$ (Lemma 27), and at the same time always AERM. The last part uses an argument in Lemma 24 of Shalev-Shwartz et al. (2010) which appeals to the universality of \mathcal{A} 's consistency on a specific discrete distribution supported on the given data set Z .

As pointed out by an anonymous reviewer, there is a simpler proof by invoking Theorem 10 of Shalev-Shwartz et al. (2010) that says any consistent and generalizing algorithm must be AERM and a result (e.g., Bassily et al., 2014, Appendix F) that says ‘‘privacy \Rightarrow generalization’’. This is a valid observation. But their Theorem 10 is proven using a detour through ‘‘generalization’’, which leads to a slower rate than what we are able to obtain in Lemma 10 using a more direct argument.

3.3 Private Learnability vs. Non-private Learnability

Now we have a characterization of all privately learnable problems, a natural question to ask is that whether any learnable problem is also privately learnable. The answer is ‘‘yes’’

for learning in Statistical Query (SQ)-model and PAC Learning model (binary classification) with finite hypothesis space, and is ‘‘no’’ for continuous hypothesis space (Chaudhuri and Hsu, 2011).

By definition, all privately learnable problems are learnable. But now that we know that privacy implies generalization, it is tempting to hope that privacy can help at least some problem to learn better than any non-private algorithm. In terms of learnability, the question becomes: Could there be a (learnable) problem that is *exclusively* learnable through private algorithms? We now show that such a problem does not exist.

Proposition 11 *If a learning problem is learnable by an ϵ -DP algorithm \mathcal{A} , then it is also learnable by a non-private algorithm.*

The proof is given in Appendix A.3. The idea is that $\mathcal{A}(Z)$ defines a distribution over \mathcal{H} . Pick an $z \in \mathcal{Z}$. If $z \notin Z$, algorithm $\mathcal{A}' = \mathcal{A}$. Otherwise, $\mathcal{A}'(Z)$ samples from a slightly different distribution than $\mathcal{A}(Z)$ that does not affect the expectation much.

On the other hand, not all learnable problems are privately learnable. This can already be seen from Chaudhuri and Hsu (2011), where the gap between learning and private learning is established. We revisit Chaudhuri and Hsu’s example in our notation under the general learning setting and produce an alternative proof by showing that differential privacy contradicts *always AERM*, then invoking Theorem 7 to show the problem is not privately learnable.

Proposition 12 (Chaudhuri and Hsu, 2011, Theorem 5) *There exists a problem that is learnable by a non-private algorithm, but not privately learnable. In particular, any private algorithm cannot be always AERM in this problem.*

We describe the counterexample and re-establish the impossibility of private learning for this problem using the contrapositive of Theorem 7, which suggests that if privacy and always AERM algorithm cannot coexist for some problem, then the problem is not privately learnable.

Consider the binary classification problem with $\mathcal{X} = [0, 1]$, $\mathcal{Y} = \{0, 1\}$ and 0-1 loss function. Let \mathcal{H} be the collection of threshold functions that output $h(x) = 1$ if $x > h$ and $h(x) = 0$ otherwise. This class has VC-dimension 1, and hence the problem is learnable.

Next we will construct $K = \lceil \exp(\epsilon_0 n) \rceil$ data sets such that if $K - 1$ of them obey AERM, the remaining one cannot be. Let $\eta = 1/\exp(\epsilon n)$, $K := \lceil 1/\eta \rceil$. Let h_1, h_2, \dots, h_K be a disjoint thresholds such that they are at least η apart and $[h_i - \eta/3, h_i + \eta/3]$ are disjoint intervals.

If we take $Z_i \subseteq [h_i - \eta/3, h_i + \eta/3]$ with half of the points in $[h_i - \eta/3, h_i)$ and the other half in $(h_i, h_i + \eta/3]$ and we label each data point in it with $\mathbf{1}(z > h_i)$, then empirical risk $\hat{R}(h_i, Z_i) = 0 \forall i = 1, \dots, K$. So for any AERM learning rule, $\mathbb{E}_{h \sim \mathcal{A}(Z)} \hat{R}(h, Z_i) \rightarrow 0$ for all i . For some sufficiently large n , $\mathbb{E}_{h \sim \mathcal{A}(Z)} \hat{R}(h, Z_i) < 0.1$.

Now consider Z_1 ,

$$\mathbb{P}(\mathcal{A}(Z_1) \notin [h_1 - \eta/3, h_1 + \eta/3]) \geq \sum_{i=2}^K \mathbb{P}(\mathcal{A}(Z_i) \in [h_i - \eta/3, h_i + \eta/3]), \quad (4)$$

since these intervals are disjoint. Then by the definition of ϵ -DP,

$$\mathbb{P}(\mathcal{A}(Z_1) \in [h_1 - \eta/3, h_1 + \eta/3]) \geq \exp(-\epsilon n) \mathbb{P}(\mathcal{A}(Z_i) \in [h_i - \eta/3, h_i + \eta/3]). \quad (5)$$

It follows that $\mathbb{P}(\mathcal{A}(Z_i) \in [h_i - \eta/3, h_i + \eta/3]) > 0.9$ otherwise $\mathbb{E}_{h \sim \mathcal{A}(Z_i)} \tilde{R}(h, Z_i) \geq 0.1$, therefore

$$\mathbb{P}(\mathcal{A}(Z_1) \notin [h_1 - \eta/3, h_1 + \eta/3]) \geq K \exp(-\epsilon n) 0.9 \geq 0.9, \quad (5)$$

and $\mathbb{E}_{h \sim \mathcal{A}(Z_i)} \tilde{R}(h, Z_i) \geq 0.9 \times 1 = 0.9$, which violates the “always AERM” condition that requires $\mathbb{E}_{h \sim \mathcal{A}(Z_1)} \tilde{R}(h, Z_1) < 0.1$. Therefore, the problem is not privately learnable.

As is pointed out by an anonymous reviewer, the same conclusion of this impossibility result of privately learning thresholds on $[0, 1]$ can be drawn numerically through the characterization of the sample complexity (Beimel et al., 2013a), via the bound that depends logarithmically on the $\log(|\mathcal{H}|)$ and on $[0, 1]$ this number is infinite. The above analysis provides different insights about the problem. We will be using it again for understanding the separation of learnability and learnability under (ϵ, δ) -Differential Privacy later in Section 6.

3.4 Private \mathfrak{D} -learnability

The above example implies that even very simple learning problems may not be privately learnable. To fix this caveat, note that most data sets of practical interest have nice distributions. Therefore, it makes sense to consider a smaller class of distributions, e.g., smooth distributions that have bounded k th order derivative, or those having bounded total variation. These are common assumptions in non-parametric statistics, such as kernel density estimation, smoothing spline regression and mode clustering. Similarly, in high dimensional statistics, there are often assumptions on the structures of the underlying distribution, such as sparsity, smoothness, and low-rank conditions.

Definition 13 ((Private) \mathfrak{D} -learnability) We say a learning problem $(\mathcal{Z}, \mathcal{H}, \ell)$ is \mathfrak{D} -learnable if there exists a learning algorithm \mathcal{A} that is consistent for every unknown distribution $\mathcal{D} \in \mathfrak{D}$. If in addition, the problem is \mathfrak{D} -learnable under ϵ -differential privacy for some $0 \leq \epsilon < \infty$, then we say the problem is privately \mathfrak{D} -learnable.

Almost all of our arguments hold in a per distribution fashion, therefore they also hold for any such subclass \mathfrak{D} . The only exception is the necessity of “always AERM” (Lemma 10), where we used the universal consistency on an arbitrary discrete uniform distribution in the proof. The characterization still holds if the class \mathfrak{D} contains all finite discrete uniform distributions. For general distribution classes, we characterize private \mathfrak{D} -learnability using a weaker “universally AERM” (instead of “always AERM”) under the assumption that the problem itself is learnable in a distribution-free setting without privacy constraints.

Lemma 14 (private \mathfrak{D} -learnability \Rightarrow private \mathfrak{D} -universal AERM) If an ϵ -DP algorithm \mathcal{A} is \mathfrak{D} -universally consistent with rate $\xi(n)$ and the problem itself is learnable in a distribution-free sense with rate $\xi'(n)$, then there exists a \mathfrak{D} -universally consistent learning algorithm \mathcal{A}' that is \mathfrak{D} -universally AERM with rate $12\xi'(n)^{1/4} + \frac{3\xi}{\sqrt{n}} + \xi(\sqrt{n})$ and satisfies $\frac{2}{\sqrt{n}}(\epsilon^\epsilon - \epsilon^{-\epsilon})$ -DP.

The proof, given in Appendix A.4, shows that the algorithm \mathcal{A}' that applies \mathcal{A} to a random subsample of size $\lfloor \sqrt{n} \rfloor$ is AERM for any distribution in the class \mathfrak{D} .

Theorem 15 (Characterization of private \mathfrak{D} -learnability) A problem is privately \mathfrak{D} -learnable if there exists an algorithm that is \mathfrak{D} -universally AERM and differentially private with privacy loss $\epsilon(n) \rightarrow 0$. If in addition, the problem is (distribution-free and non-privately) learnable, then the converse is also true.

Proof The “if” part is exactly the same as the argument in Section 3.1, since both Lemma 8 and Lemma 9 holds for each distribution independently. Under the additional assumption that the problem itself is learnable (distribution-free and non-privately), the “only if” part is given by Lemma 14. ■

This result may appear to be unsatisfactory due to the additional assumption of learnability. It is clearly a strong assumption because many problems that are \mathfrak{D} -learnable for a practically meaningful \mathfrak{D} are not actually learnable. We provide one such example here.

Example 1 Let the data space be $[0, 1]$, the hypothesis space be the class of all finite subset of $[0, 1]$ and the loss function $\ell(h, z) = 1_{z \notin h}$. This problem is not learnable, and not even \mathfrak{D} -learnable when \mathfrak{D} is the class of all discrete distributions with finite number of possible values. But it is \mathfrak{D} -learnable when \mathfrak{D} is further restricted with an upper bound on the total number of possible values.

Proof For any discrete distribution with a finite support set, there is an $h \in \mathcal{H}$ such that the optimal risk is 0. Assume the problem is learnable with rate $\xi(n)$, then for some n $\xi(n) < 0.5$. However, we can always construct a uniform distribution over $3n$ elements and it is information-theoretically impossible for any estimators based on n samples from the distribution to achieve a risk better than $2/3$. The problem is therefore not learnable. When we assume an upper bound N on the maximum number of bins of the underlying distribution, then the ERM which outputs just the support of all observed data will be universally consistent with rate $\xi(n) = N/n$. ■

It turns out that we cannot hope to completely remove the assumption from Theorem 15. The following example illustrates that some form of qualification (implied by the learnability assumption) is necessary for the converse statement to be true.

Example 2 Consider the learning problem in Example 1. Let \mathfrak{D} be the class of all continuous distributions. There is a learning problem that is s privately \mathfrak{D} -learnable but no private AERM algorithm exists.

Proof Let the learning problem be that in Example 1 and \mathcal{Q} be the class of all continuous distributions defined on $[0, 1]$. Consider the learning algorithm $\mathcal{A}(Z)$ always returns $h = \theta$. The optimal risk for any continuous distribution is 1 because any finite subset is of measure 0, output θ is 0-consistent and 0-generalizing, but not AERM, since the minimum empirical risk is 0. \mathcal{A} is also 0-differentially private, therefore the problem is privately \mathcal{Q} -learnable for \mathcal{Q} being the set of all continuous distributions.

However, it is not privately \mathcal{Q} -learnable via an AERM, i.e., no private AERM algorithm exists for this problem. We prove this by contradiction. Assume an ϵ -DP AERM algorithm exists, the subsampling lemma ensures the existence of an $\epsilon(n)$ -DP AERM algorithm \mathcal{A}' with $\epsilon(n) \rightarrow 0$. \mathcal{A}' is therefore generalizing by stability, and it follows that the \mathcal{A}' has risk $\mathbb{E}_{h \sim \mathcal{A}'(Z)} R(h)$ converging to 0. But there is no $h \in \mathcal{H}$ such that $R(h) < 1$, giving the contradiction. ■

Interestingly, this problem is \mathcal{Q} -learnable via a non-private AERM algorithm, which always outputs $h = Z$. This is 0-consistent, 0-AERM but not generalizing. This example suggests that \mathcal{Q} -learnability and learnability are quite different because for learnable problems, if an algorithm is consistent and AERM, then it must also be generalizing (Shalev-Shwartz et al., 2010, Theorem 10).

3.5 A generic learning algorithm

The characterization of private learnability suggests a generic (but impractical) procedure that learns all privately learnable problems (in the same flavor as the generic algorithm in Shalev-Shwartz et al. (2010)) that learns all learnable problems). This is to solve

$$\underset{\substack{(\mathcal{A}, \phi): \\ \mathcal{A}: \mathcal{Z}^n \rightarrow \mathcal{H}, \\ \mathcal{A} \text{ is } \epsilon\text{-DP}}}{\operatorname{argmin}} \left[\epsilon + \sup_{Z \in \mathcal{Z}^n} \left(\mathbb{E}_{h \sim \mathcal{A}(Z)} \hat{R}(h, Z) - \inf_{h \in \mathcal{H}} \hat{R}(h, Z) \right) \right], \quad (6)$$

or to privately \mathcal{Q} -learn the problem when (6) is not feasible

$$\underset{\substack{(\mathcal{A}, \phi): \\ \mathcal{A}: \mathcal{Z}^n \rightarrow \mathcal{H}, \\ \mathcal{A} \text{ is } \epsilon\text{-DP}}}{\operatorname{argmin}} \left[\epsilon + \sup_{D \in \mathcal{Q}} \mathbb{E}_{Z \sim \mathcal{D}^n} \left(\mathbb{E}_{h \sim \mathcal{A}(Z)} \hat{R}(h, Z) - \inf_{h \in \mathcal{H}} \hat{R}(h, Z) \right) \right]. \quad (7)$$

Theorem 16 *Assume the problem is learnable. If the problem is private learnable, (6) will always output a universally consistent private learning algorithm. If the problem is private \mathcal{Q} -learnable, (7) will always output a \mathcal{Q} -universally consistent private learning algorithm.*

Proof If the problem is private learnable, by Theorem 7 there exists an algorithm \mathcal{A} that is $\epsilon(n)$ -DP and always AERM with rate $\xi(n)$ and $\epsilon(n) + \xi(n) \rightarrow 0$. This \mathcal{A} is a witness in the optimization so we know that any minimizer of (6) will have a objective value that is no greater than $\epsilon(n) + \xi(n)$ for any n . Corollary 9 concludes its universal consistency. The second claim follows from the characterization of private \mathcal{Q} -learnability in Theorem 15. ■

Algorithm 1 Exponential Mechanism for regularized ERM

Input: Data points $Z = \{z_1, \dots, z_n\} \in \mathcal{Z}^n$, loss function ℓ , regularizer g_n , privacy parameter $\epsilon(n)$ and a hypothesis space \mathcal{H} .

1. Construct utility function $q(h, Z) := -\frac{1}{n} \sum_{i=1}^n \ell(h, z_i) - g_n(h)$, and its sensitivity $\Delta q := \sup_{h \in \mathcal{H}, d(Z, Z')=1} |q(h, Z) - q(h, Z')| \leq \frac{1}{n} \sup_{h \in \mathcal{H}, z \in \mathcal{Z}} \{\ell(h, z)\}$.

2. Sample $h \in \mathcal{H}$ with probability $\mathbb{P}(h) \propto \exp(\frac{\epsilon(n)}{2\Delta q} q(h, Z))$.

Output: h .

It is of course impossible to minimize the supremum over any data Z , nor is it possible to efficiently search over the space of all algorithms, let alone DP algorithms. But conceptually, this formulation may be of interest to theoretical questions related to the search of private learning algorithms and the fundamental limit of machine learning under privacy constraints.

4. Private learning for penalized ERM

Now we describe a generic and practical class of private learning algorithms, based on the idea of minimizing the empirical risk under privacy constraint:

$$\underset{h \in \mathcal{H}}{\operatorname{minimize}} F(Z, h) = \frac{1}{n} \sum_{i=1}^n \ell(h, z_i) + g_n(h). \quad (8)$$

The first term is empirical risk and the second term vanishes as n increases so that this estimator is asymptotically ERM. The same formulation has been studied before in the context of differentially private machine learning (Chandhuri et al., 2011; Kifer et al., 2012), but our focus is more generic and does not require the objective function to be convex, differentiable, continuous, or even have a finite dimensional Euclidean space embedding, hence covers a larger class of learning problems.

Our generic algorithm for differentially private learning is summarized in Algorithm 1. It applies the exponential mechanism (McSherry and Talwar, 2007) to penalized ERM. We note that this algorithm implicitly requires that $\int_{\mathcal{H}} \exp(\frac{\epsilon(n)}{2\Delta q} q(h, Z)) dh < \infty$, otherwise the distribution is not well-defined and it does not make sense to talk about differential privacy. In general, if \mathcal{H} is a compact set with a finite volume (with respect to a base measure, such as the Lebesgue measure or counting measure), then such a distribution always exists. We will revisit this point and discuss the practicality of this assumption in the Section 5.3.

Using the characterization results developed so far, we are able to give sufficient conditions for consistency of private learning algorithms without having to establish uniform convergence. Define the sublevel set as

$$S_{Z, t} = \{h \in \mathcal{H} \mid F(Z, h) \leq t + \inf_{h \in \mathcal{H}} F(Z, h)\}, \quad (9)$$

where $F(h, Z)$ is the regularized empirical risk function defined in (8). In particular, we assume the following conditions:

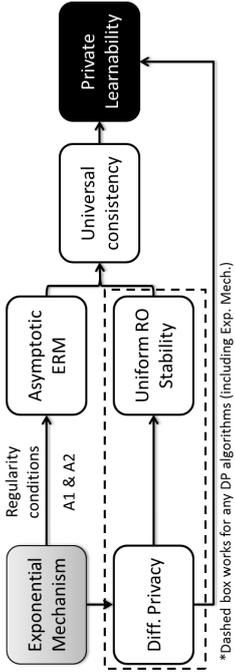


Figure 3: Illustration of Theorem 17: conditions for private learnability in general learning setting.

A1. Bounded loss function: $0 \leq \ell(h, z) \leq 1$ for any $h \in \mathcal{H}, z \in \mathcal{Z}$.

A2. Sublevel set condition: There exist constant positive integer n_0 , positive real number t_0 , and a sequence of regularizer g_n satisfying $\sup_{h \in \mathcal{H}} |g_n(h)| = o(n)$, such that for any $0 < t < t_0, n > n_0$

$$\mathbb{E}_{Z \sim \mathcal{D}^n} \left(\frac{\mu(\mathcal{H})}{\mu(\mathcal{S}_{Z,t})} \right) \leq K \left(\frac{1}{t} \right)^\rho, \tag{10}$$

where $K = K(n), \rho = \rho(n)$ satisfy $\log K + \rho \log n = o(n)$. Here the measure μ may depend on context, such as Lebesgue measure (\mathcal{H} is continuous) or counting measure (\mathcal{H} is discrete).

The first condition of boundedness is common. It is assumed in Vapnik’s characterization for ERM learnability and Shalev-Shwartz et al.’s general characterization of all learnable problems. In fact, we can always consider \mathcal{H} to be a sublevel set such that the boundedness condition holds. For the second condition, the intuition is that we require the sublevel set to be large enough such that the sampling procedure will return a good hypothesis with large probability. $\mu(\mathcal{S}_t)$ is a critical parameter in the utility guarantee for the exponential mechanism (McSherry and Talwar, 2007). Also, it is worth pointing out that A2 implies that the exponential distribution is well-defined.

Theorem 17 (General private learning) *Let $(\mathcal{Z}, \mathcal{H}, \ell)$ be any problem in the general learning setting. Suppose we can choose g_n such that A.1 and A.2 are satisfied with (ρ, K, g_n, n_0, t_0) for a distribution \mathcal{D} , then Algorithm 1 satisfies $\epsilon(n)$ -privacy and is consistent with rate*

$$\xi(n) = \frac{9 \lceil \log K + (\rho + 2) \log n \rceil}{n\epsilon(n)} + 2\epsilon(n) + \sup_{h \in \mathcal{H}} |g_n(h)|. \tag{11}$$

In particular, if $\epsilon(n) = o(1), \sup_{h \in \mathcal{H}} |g_n(h)| = o(1)$ and $\log K + \rho \log n = o(n\epsilon(n))$ for all \mathcal{D} (in \mathcal{D}) Algorithm 1 privately learns $(\mathcal{D}$ -learns) the problem.

We give an illustration of the proof in Figure 3. The detailed proof, based on the stability argument (Shalev-Shwartz et al., 2010), is deferred to Appendix A.5.

To see that Theorem 17 actually contains a large number of problems in the general learning setting. We provide concrete examples that satisfy A1 and A2 below for both privately learnable and privately \mathcal{D} -learnable problems that can be learned using Algorithm 1.

4.1 Examples of privately learnable problems

We start from a few cases where Algorithm 1 is universally consistent for all distributions.

Example 3 (Finite discrete \mathcal{H}) *Suppose \mathcal{H} can be fully encoded by M -bits, then*

$$\mu(\mathcal{S}_t) / \mu(\mathcal{H}) \geq |\mathcal{H}|^{-1} = 2^{-M},$$

since there are at least 1 optimal hypothesis for each function and now μ is the counting measure. In other word, we can take $K = 2^M$ and $\rho = 0$ in the (11). Plug this into the expression and take $g_n \equiv 0, \epsilon(n) = \sqrt{(M + \log n)/n}$, we get a rate of consistency $\xi(n) = O(\frac{M + \log n}{\sqrt{n}})$. In addition, if we can find a data-independent covering set for a continuous space, then we can discretize the space and the result same results follow. This observation will be used in the construction of many private learning algorithms below.

Example 4 (Lipschitz functions/Hölder class) *Let \mathcal{H} be a compact, β_p -regular subset of \mathbb{R}^d satisfying $\mu(B \cap \mathcal{H}) \geq \beta_p \mu(B)$ for any ℓ_p ball $B \subset \mathbb{R}^d$ that is small enough. Assume that $F(Z, \cdot)$ is L -Lipschitz on \mathcal{H} : for any $h, h' \in \mathcal{H}$,*

$$|F(Z, h) - F(Z, h')| \leq L \|h - h'\|_p.$$

Then for sufficiently small t , we have Lebesgue measure

$$\mu(\mathcal{S}_t) \geq \beta_p (t/L)^d$$

and Condition A.2 holds with $K = \mu(\mathcal{H})\beta_p^{-1}L^d, \rho = d$. Furthermore, if we take $\epsilon(n) = \sqrt{\frac{d(\log L + \log n) + \log(\mu(\mathcal{H})/\beta_p)}{n}}$, the algorithm is $O\left(\sqrt{\frac{d(\log L + \log n) + \log(\mu(\mathcal{H})/\beta_p)}{n}} + \sup_{h \in \mathcal{H}} |g_n(h)|\right)$ -consistent.

This shows that condition A2 holds for a large class of low-dimensional problems of interest in machine learning and one can learn the problem privately without actually needing to find a covering set algorithmically. Specifically, the example includes many practically used methods such as logistic regression, linear SVM, ridge regression, even multi-layer neural networks, since the loss functions in these methods are jointly bounded in (Z, h) and Lipschitz in h .

The example also raises an interesting observation that while differentially private classification is not possible in a distribution-free setting for 0-1 loss function (Chaudhuri and Hsu, 2011), it is learnable under smoother surrogate loss, e.g., logistic loss or hinge loss. In other words, private learnability and computational tractability both benefit from the same relaxation.

The Lipschitz condition still requires the dimension of the hypothesis space to be $o(n)$. Thus it does not cover high-dimensional machine learning problems where $d \gg n$, nor does it contain the example of Shalev-Shwartz et al. (2010) that ERM fails.

For high dimensional problems where d grows with n , typically some assumptions or restrictions need to be made either on the data or on the hypothesis space (so that it becomes essentially low-dimensional). We give one example here for the problem of sparse regression.

Example 5 (Best subset selection) Consider $\mathcal{H} = \{h \in \mathbb{R}^d : \|h\|_0 < s, \|h\|_2 \leq 1\}$ and let $\ell(l, z)$ be an L -Lipschitz loss function. The solution can only be chosen from $\binom{[d]}{s} < d^s$ different s -dimensional subspaces. We can apply Algorithm 1 twice to first sample a support set S with utility function being the $-\min_{h \in \mathcal{H}_S} F(Z, h)$, and then sample a solution in the chosen s -dimensional subspace. By the composition theorem this two-stage procedure is differentially private. Moreover, by the arguments in Example 3 and Example 4 respectively, we have $\mu(\mathcal{S}) \geq \left(\frac{1}{d}\right)^s$ for the subset selection and $\mu(\mathcal{S}) \geq \left(\frac{1}{T}\right)^s$ for the low-dimensional regression. Note that $p = 0$ in both cases and the dependency on the ambient dimension d is on the logarithm. The first stage ensures that for the chosen support set S , $\min_{h \in \mathcal{H}_S} F(Z, h)$ is close to $\min_{h \in \mathcal{H}} F(Z, h)$ by $O\left(\frac{s \log d + \log \frac{1}{\epsilon}}{n \epsilon(n)}\right)$ in expectation and (the second stage ensures that the sampled hypothesis from \mathcal{H}_S would have objective function close to $\min_{h \in \mathcal{H}_S} F(Z, h)$ by $O\left(\frac{s \log L + \log n + \log(\mu(\mathcal{H}_S)/\beta d)}{n \epsilon(n)}\right)$. This leads to an overall rate of consistency (they simply add up) of $O\left(\frac{s \log d + \log n + L + \log(\mu(\mathcal{H}_S)/\beta d)}{\sqrt{n}}\right)$ if we choose $\epsilon(n) = 1/\sqrt{n}$.

4.2 Examples of privately \mathfrak{D} -learnable problems.

For problems where private learnability is impossible to achieve, we may still apply Theorem 17 to prove the weaker private \mathfrak{D} -learnability for some specific class of distributions.

Example 6 (Finite Representation Dimension in the General Learning Setting) For binary classification problems with 0-1 loss (PAC learning), this has been well-studied. In particular, Beinel et al. (2013a) characterized the sample complexity of privately learnable problems using a combinatorial condition they call a ‘‘Probabilistic Representation’’, which basically involves finding a finite, data-independent set of hypotheses to approximate any hypothesis in the class. Their claim is that if the ‘‘representation dimension’’ is finite, then the problem is privately learnable, otherwise it is not. We can extend the notion of probabilistic representation beyond the finite discrete and countably infinite hypothesis class considered in Beinel et al. (2013a) to cases when the problem is not privately learnable (e.g. learning threshold functions on $[0, 1]$). The existence of probabilistic representation for all distributions in \mathfrak{D} would lead to a \mathfrak{D} -universally private learning algorithm.

Another way to define a class of distribution \mathfrak{D} is to assume the existence of a reference distribution that is close to any distribution of interest as in Chaudhuri and Hsu (2011).

Example 7 (Existence of a public reference distribution) To deal with the 0-1 loss classification problems on a continuous hypothesis domain, Chaudhuri and Hsu (2011) assume

that there exists a data-independent reference distribution \mathcal{D}^* , which by multiplying a fixed constant on its density, uniformly dominates any distribution of interest. This essentially produces a subset of distributions \mathfrak{D} . The consequence is that one can build an ϵ -net of \mathcal{H} with metric defined on the risk under \mathcal{D}^* and this will also be a (looser) covering set of any distribution $\mathcal{D} \in \mathfrak{D}$, thereby learning the problem for any distribution in the set.

The same idea can be applied to the general learning setting. For any fixed reference distribution \mathcal{D}^* defined on \mathcal{Z} and constant c ,

$$\mathfrak{D} = \{\mathcal{D} = (Z, \mathcal{F}, \mathbb{P}) \mid \mathbb{P}(\mathcal{D}(z \in A)) \leq c\mathbb{P}^*(z \in A) \text{ for } \forall A \in \mathcal{F}\}$$

is a valid set of distributions and we are able to \mathfrak{D} -privately learn this problem whenever we can construct a sufficiently small cover set with respect to \mathcal{D}^* and reduce the problem to Example 3. This class of problems includes high-dimensional and infinity dimensional problems such as density estimation, nonparametric regression, kernel methods and essentially any other problems that are strictly learnable (Vapnik, 1998), since they are characterized by one-sided uniform convergence (and the corresponding entropy condition).

4.3 Discussion on uniform convergence and private learnability

Uniform convergence requires that $\mathbb{E}_{Z \sim \mathcal{D}^n} \sup_{h \in \mathcal{H}} |\hat{R}(h, Z) - R(h)| \rightarrow 0$ for any distribution \mathcal{D} with a distribution independent rate. Most machine learning algorithms rely on uniform convergence to establish consistency result (e.g., through complexity measure such as VC-dimension, Rademacher Complexity, covering and bracketing numbers and so on). In fact, the learnability of ERM algorithm is characterized by the one-sided uniform convergence (Vapnik, 1998), which is only slightly weaker than requiring uniform convergence on both sides.

A key point in Shalev-Shwartz et al. (2010) is that the learnability (by any algorithm) in general learning setting is no longer characterized by variants of uniform convergence. However, the class of privately learnable problems is much smaller. Clearly, uniform convergence is not sufficient for a problem to be privately learnable (see Section 3.3), but is it necessary?

In binary classification with discrete domain (agnostic PAC Learning), since VC-dimension being finite characterizes the class of privately PAC learnable problems, the necessity of uniform convergence is clear. This could also be more explicitly seen from Beinel et al. (2013a) where the *probabilistic representation dimension* is a form of uniform convergence on its own.

In the general learning setting, the problem is still open. We were not able to prove that private learnability implies uniform convergence, but we could not construct a counter example either. All our examples in this section do implicitly or explicitly uses uniform convergence, which seems to hint at a positive answer.

5. Practical concerns

5.1 High confidence private learning via boosting

We have stated all results so far in expectation. We can easily convert these to the high-confidence learning paradigm by applying Markov's inequality, since convergence in expectation to the minimum risk implies convergence in probability to the minimum risk. While the $1/\delta$ dependence on the failure probability δ is not ideal, we can apply a similar meta-algorithm "boosting" (Schapire, 1990) as in Shalev-Shwartz et al. (2010, Section 7) to get a $\log(1/\delta)$ rate. The approach is similar to cross-validation. Given a pre-chosen positive integer a , the original boosting algorithm randomly partitions the data into $(a+1)$ subsamples of size $n/(a+1)$, and applies Algorithm 1 on the first a partitions, obtaining a candidate hypotheses. The method then returns the one hypothesis with smallest validation error, calculated using the remaining subsample. To ensure differential privacy, our method instead uses the exponential mechanism to sample the best candidate hypothesis, where the logarithm of sampling probability is proportional to the negative validation error.

Theorem 18 (High-confidence private learning) *If an algorithm \mathcal{A} privately learns a problem with rate $\xi(n)$ and privacy parameter $\epsilon(n)$, then the boosting algorithm \mathcal{A}' with $a = \log \frac{3}{\delta}$ is $\max \left\{ \epsilon \left(\frac{n}{\log(3/\delta)+1} \right), \frac{\log(3/\delta)+1}{\sqrt{n}} \right\}$ -differentially private, its output h obeys*

$$R(h) - R^* \leq \epsilon \xi \left(\frac{n}{\log(3/\delta)+1} \right) + C \sqrt{\frac{\log(3/\delta)}{n}}$$

for an absolute constant C with probability at least $1 - \delta$.

5.2 Efficient sampling algorithm for convex problems

Our proposed exponential sampling based algorithm is to establish a more explicit geometric condition upon which AERM holds, hence the algorithm may not be computationally tractable. Ignoring the difficulty of constructing the ϵ -covering set of an exponential number of elements, sampling from the set alone is not a polynomial time algorithm. But we can solve a subset of the continuous version of our Algorithm 1 described in Theorem 17 in polynomial time to arbitrary accuracy (see also Bassily et al. (2014, Theorem 3.4)).

Proposition 19 *If $n^{-1} \sum_{i=1}^n \ell(h, z_i) + g_n(h)$ is convex in h and \mathcal{H} is a convex set, then the sampling procedure in Algorithm 1 can be solved in polynomial time.*

Proof When $n^{-1} \sum_{i=1}^n \ell(h, z_i) + g_n(h)$ is convex, the utility function $q(h, Z)$ is concave in h . The density to be sampled from in Algorithm 1 is proportional to $\exp(\frac{aq(h, Z)}{B})$ and is log-concave. The Markov chain sampling algorithm in Applegate and Kannan (1991) is guaranteed to produce a sample from a distribution that is arbitrarily close to the target distribution (in the total variation sense) in polynomial time. ■

5.3 Exponential mechanism in infinite domain

As we mention earlier, the results in Section 4 based on the exponential mechanism implicitly assumes certain regularity conditions that ensures the existence of a probability distribution.

When \mathcal{H} is finite, the existence is trivial. On the other hand, an infinite set \mathcal{H} is tricky in that there may not exist a proper distribution that satisfies $\mathbb{P}(h) \propto e^{\frac{\epsilon}{2\Delta\gamma} q(Z;h)}$ for at least some $q(Z;h)$. For instance, if $\mathcal{H} = \mathbb{R}$ and $q(Z;h) \equiv 1$ then $\int_{\mathbb{R}} e^{\frac{\epsilon}{2\Delta\gamma} q(Z;h)} dh = \infty$. Such distributions that are only defined up to scale with no finite normalization constants are called improper distributions. In case of finite dimensional non-compact set, this translates into an additional assumption on the loss function and the regularization term.

Things get even trickier when \mathcal{H} is an infinite dimensional space, such as a subset of a Hilbert space. While probability measures can still be defined, no density function can be defined on such spaces. Therefore, we cannot use exponential mechanism to define a valid probability distribution.

The practical implication is that exponential mechanism is really only applicable to cases when the hypothesis space \mathcal{H} allows for definitions of densities in the usual sense, or then \mathcal{H} can be approximated by such a space. For example, a separable Hilbert space can be studied by finite-dimensional projections. Also, we can approximate RKHS induced by translation invariant kernels via random Fourier features (Rahimi and Recht, 2007).

6. Results for learnability under (ϵ, δ) -differential privacy

Another way to weaken the definition of private learnability is through (ϵ, δ) -approximate differential privacy.

Definition 20 (Dwork et al., 2006a) *An algorithm \mathcal{A} obeys (ϵ, δ) -differential privacy if for any Z, Z' such that $d(Z, Z') \leq 1$, and for any measurable set $S \subset \mathcal{H}$*

$$\mathbb{P}_{h \sim \mathcal{A}(Z)}(h \in S) \leq e^\epsilon \mathbb{P}_{h \sim \mathcal{A}(Z')} (h \in S) + \delta.$$

We define a version of the problem to be

Definition 21 (Approximately Private Learnability) *We say a learning problem is $\Delta(n)$ -approximately privately learnable for some pre-specified family of rate $\Delta(n)$ if for some $\epsilon < \infty$, $\delta(n) \in \Delta(n)$, there exists a universally consistent algorithm that is $(\epsilon, \delta(n))$ -DP.*

This is a completely different subject to study and the class of approximately privately learnable problems could be substantially larger than the pure privately learnable problems. Moreover, the picture may vary with respect to how small $\delta(n)$ is required to be. In this section, we present our preliminary investigation on this problem.

Specifically, we will consider two questions:

1. Does the existence of an (ϵ, δ) -DP always AERM algorithm characterize the class of approximately private learnable problems?

2. Are all learnable problems approximately privately learnable for different choices of $\Delta(n)$?

The minimal requirement in the same flavor of Definition 3 would be to require $\Delta(n) = \{\delta(n) | \delta(n) \rightarrow 0\}$. The learnability problem turns out to be trivial under this definition due to the following observation.

Lemma 22 *For any algorithm \mathcal{A} that acts on Z , \mathcal{A}' that runs \mathcal{A} on a randomly chosen subset of Z of size \sqrt{n} is $(0, \frac{1}{\sqrt{n}})$ -DP.*

Proof Let Z and Z' be adjacent datasets that differs only in data point i . For any i and any $S \in \sigma(\mathcal{H})$,

$$\begin{aligned} \mathbb{P}_I(\mathcal{A}(Z) \in S) &= \mathbb{P}_I(\mathcal{A}(Z_I) \in S | i \in I) + \mathbb{P}_I(\mathcal{A}(Z_I) \in S | i \notin I) \mathbb{P}(i \notin I) \\ &= \mathbb{P}_I(\mathcal{A}(Z_I) \in S | i \in I) \mathbb{P}(i \in I) + \mathbb{P}_I(\mathcal{A}(Z_I) \in S | i \notin I) \mathbb{P}(i \notin I) \\ &= \mathbb{P}(\mathcal{A}'(Z') \in S) + \mathbb{P}_I(\mathcal{A}(Z_I) \in S | i \in I) - \mathbb{P}_I(\mathcal{A}'(Z_I) \in S | i \in I) \mathbb{P}(i \in I) \\ &\leq \mathbb{P}(\mathcal{A}'(Z') \in S) + \mathbb{P}(i \in I) \\ &= e^0 \mathbb{P}(\mathcal{A}'(Z') \in S) + \frac{1}{\sqrt{n}}. \end{aligned}$$

This verifies the $(0, 1/\sqrt{n})$ -DP of algorithm \mathcal{A}' . \blacksquare

The above lemma suggests that if $\delta(n) = o(1)$ is all we need for the *approximately private learnability*, then any consistent learning algorithm can be made approximately DP by simply subsampling. In other words, any learnable problem is also learnable under approximate differential privacy.

To get around this triviality, we need to specify a sufficiently fast rate of $\delta(n)$ going to 0. While it is common to require that $\delta(n) = o(1/\text{poly}(n))$ ² for cryptographically strong privacy protection, requiring $\delta(n) = o(1/n)$ is already enough to invalidate the above subsampling argument and makes the problem of learnability a non-trivial one.

Again, the question is whether AERM characterizes approximately private learnability and whether there is a gap between the class of learnable and approximately privately learnable problems.

Here we show that the “folklore” Lemma 8 and subsampling lemma (Lemma 27) can be extended to work with (ϵ, δ) -DP and then we provide a positive answer to the first question.

Lemma 23 (Stability of (ϵ, δ) -DP) *If \mathcal{A} is (ϵ, δ) -DP, and $0 \leq \ell(h, z) \leq 1$, then \mathcal{A} is $(\epsilon^e - 1 + \delta)$ -Strongly Uniform RO-stable.*

2. Here the notation “ $o(1/\text{poly}(n))$ ” means “decays faster than any polynomial of n ”. A sequence $a(n) = o(1/\text{poly}(n))$ if and only if $a(n) = o(n^{-\gamma})$ for any $\gamma > 0$.

Proof For any Z, Z' such that $d(Z, Z') \leq 1$ and for any $z \in \mathcal{Z}$. Let the event $E = \{h | p(h) \geq p'(h)\}$,

$$\begin{aligned} |\mathbb{E}_{h \sim \mathcal{A}(Z)} \ell(h, z) - \mathbb{E}_{h \sim \mathcal{A}(Z')} \ell(h, z)| &= \left| \int_E \ell(h, z) p(h) dh - \int_h \ell(h, z) p'(h) dh \right| \\ &\leq \sup_{h, z} \ell(h, z) \int_E p(h) - p'(h) dh \leq \int_E p(h) - p'(h) dh = \mathbb{P}_{h \sim \mathcal{A}(Z)}(h \in E) - \mathbb{P}_{h \sim \mathcal{A}(Z')} (h \in E) \\ &\leq (\epsilon^e - 1) \mathbb{P}_{h \sim \mathcal{A}(Z')} (h \in E) + \delta \leq \epsilon^e - 1 + \delta. \end{aligned}$$

The last line applies the definition of (ϵ, δ) -DP. \blacksquare

Lemma 24 (Subsampling Lemma of (ϵ, δ) -DP) *If \mathcal{A} is (ϵ, δ) -DP, then \mathcal{A}' that acts on a random subsample of Z of size γn obeys (ϵ', δ') -DP with $\epsilon' = \log(1 + \gamma e^{\epsilon} - 1)$ and $\delta' = \gamma \epsilon \delta$.*

Proof For any event $E \in \sigma(\mathcal{H})$, let i be the coordinate where Z and Z' differs

$$\begin{aligned} \mathbb{P}_{h \sim \mathcal{A}'(Z)}(h \in E) &= \gamma \mathbb{P}_{h \sim \mathcal{A}(Z)}(h \sim E | i \in I) + (1 - \gamma) \mathbb{P}_{h \sim \mathcal{A}(Z)}(h \sim E | i \notin I) \\ &= \gamma \mathbb{P}_{h \sim \mathcal{A}(Z)}(h \sim E | i \in I) + (1 - \gamma) \mathbb{P}_{h \sim \mathcal{A}(Z')} (h \sim E | i \notin I) \\ &= \gamma \mathbb{P}_{h \sim \mathcal{A}(Z)}(h \sim E | i \in I) - \gamma \mathbb{P}_{h \sim \mathcal{A}(Z')} (h \sim E | i \in I) + \gamma \mathbb{P}_{h \sim \mathcal{A}(Z')} (h \sim E | i \in I) \\ &\quad + (1 - \gamma) \mathbb{P}_{h \sim \mathcal{A}(Z')} (h \sim E | i \notin I) \\ &= \mathbb{P}_{h \sim \mathcal{A}'(Z)}(h \in E) + \gamma [\mathbb{P}_{h \sim \mathcal{A}(Z)}(h \sim E | i \in I) - \mathbb{P}_{h \sim \mathcal{A}(Z')} (h \sim E | i \in I)] \\ &\leq \mathbb{P}_{h \sim \mathcal{A}'(Z)}(h \in E) + \gamma (\epsilon^e - 1) \mathbb{P}_{h \sim \mathcal{A}(Z')} (h \sim E | i \in I) + \gamma \delta, \end{aligned} \tag{12}$$

where in last line, we apply (ϵ, δ) -DP of \mathcal{A} .

It remains to show that $\mathbb{P}_{h \sim \mathcal{A}(Z')} (h \sim E | i \in I)$ is similar to $\mathbb{P}_{h \sim \mathcal{A}'(Z')} (h \in E)$. First,

$$\mathbb{P}_{h \sim \mathcal{A}'(Z')} (h \in E) = \gamma \mathbb{P}_{h \sim \mathcal{A}(Z')} (h \in E | i \in I) + (1 - \gamma) \mathbb{P}_{h \sim \mathcal{A}(Z')} (h \in E | i \notin I). \tag{13}$$

Denote $\mathcal{I}_1 = \{I | i \in I\}$, $\mathcal{I}_2 = \{I | i \notin I\}$. We know $|\mathcal{I}_1| = \binom{n-1}{\gamma n-1}$, and $|\mathcal{I}_2| = \binom{n-1}{\gamma n}$ and $|\mathcal{I}_1|/|\mathcal{I}_2| = \gamma n / (n - \gamma n)$. For every $I \in \mathcal{I}_2$ there are precisely γn elements $J \in \mathcal{I}_1$ such that $d(I, J) = 1$. Likewise, for every $J \in \mathcal{I}_1$, there are $n - \gamma n$ elements $I \in \mathcal{I}_2$ such that $d(I, J) = 1$. It follows by symmetry that if we apply (ϵ, δ) -DP to $1/\gamma n$ of each $I \in \mathcal{I}_2$ and change I to their corresponding $J \in \mathcal{I}_1$, then each $J \in \mathcal{I}_1$ will receive $(n - \gamma n) / \gamma n$ “contribution” in total from the sum over all $I \in \mathcal{I}_2$.

$$\begin{aligned} \mathbb{P}_{h \sim \mathcal{A}(Z')} (h \in E | i \notin I) &= \frac{1}{|\mathcal{I}_2|} \sum_{I \in \mathcal{I}_2} \mathbb{P}_{h \sim \mathcal{A}(Z')} (h \in E) \\ &= \frac{1}{|\mathcal{I}_2|} \sum_{I \in \mathcal{I}_2} \sum_{j=1}^{\gamma n} \frac{1}{\gamma n} \mathbb{P}_{h \sim \mathcal{A}(Z')} (h \in E) \\ &\geq \frac{|\mathcal{I}_1|}{|\mathcal{I}_2|} \frac{1}{|\mathcal{I}_1|} \sum_{J \in \mathcal{I}_1} \frac{n - \gamma n}{\gamma n} e^{-\epsilon} \mathbb{P}_{h \sim \mathcal{A}(Z')} (h \in E) - \delta \\ &= \frac{1}{|\mathcal{I}_1|} \sum_{J \in \mathcal{I}_1} e^{-\epsilon} (\mathbb{P}_{h \sim \mathcal{A}(Z')} (h \in E) - \delta) = e^{-\epsilon} \mathbb{P}_{h \sim \mathcal{A}(Z')} (h \in E | i \in I) - e^{-\epsilon} \delta \end{aligned}$$

Substitute into (13), we get

$$\mathbb{P}_{h \sim \mathcal{A}(Z_I)}(h \in E) \leq \frac{1}{\gamma + (1-\gamma)e^{-\epsilon}} \mathbb{P}_{h \sim \mathcal{A}(Z)}(h \in E) + \frac{(1-\gamma)e^{-\epsilon}}{\gamma + (1-\gamma)e^{-\epsilon}} \delta.$$

We further relax the upper bound to a simple form $e^{\epsilon} \mathbb{P}_{h \sim \mathcal{A}(Z)}(h \in E) + \delta$ and substitute into (12), we have

$$\mathbb{P}_{h \sim \mathcal{A}(Z)}(h \in E) \leq (1 + \gamma e^{\epsilon}(e^{\epsilon} - 1)) \mathbb{P}_{h \sim \mathcal{A}(Z)}(h \in E) + \gamma \delta + \gamma(e^{\epsilon} - 1)\delta,$$

which concludes the proof. \blacksquare

Using the above two lemmas, we are able to establish the same result which says that AERM characterizes the approximate private learnability for certain classes of $\Delta(n)$.

Theorem 25 *A problem is $\Delta(n)$ -approximately privately learnable implies that there exists an always AERM algorithm that is $(\epsilon(n), n^{-1/2}e^{\delta(\sqrt{n})})$ -DP for some $\epsilon(n) \rightarrow 0$ and $\delta(\sqrt{n}) \in \Delta(n)$. The converse is also true if $n^{-1/2}e^{\delta(\sqrt{n})} \in \Delta(n)$.*

Proof If we have an always AERM algorithm with $\xi_{erm}(n)$ that is $(\epsilon(n), \delta(n))$ -DP for $\delta(n) \in \Delta(n)$. Then by Lemma 23, this algorithm is strongly uniform RO-stable with rate $e^{\epsilon(n)} - 1 + \delta(n)$. By Theorem 28, the algorithm is universally consistent with rate $\xi_{erm}(n) + e^{\epsilon(n)} - 1 + \delta(n)$. This establishes the ‘‘if’’ part.

To see the ‘‘only if’’ part, by definition if a problem is $\Delta(n)$ -approximately privately learnable with ϵ and $\delta(n) \in \Delta(n)$. Then by Lemma 24 with $\gamma = 1/\sqrt{n}$, we get an algorithm that obeys the privacy condition. It remains to prove always AERM, which requires exactly the same arguments in the proof of Lemma 10. Details are omitted. \blacksquare

Note that the results above suggest that in the two canonical settings $\Delta(n) = o(1/n)$ or $\Delta(n) = o(1/\text{poly}(n))$, existence of a private AERM algorithm that satisfies the stronger constraint $\epsilon(n) = o(1)$ characterizes the learnability.

The next question that whether any learnable problems are also approximately privately learnable would depend on how fast $\delta(n)$ is required to decay. We know that when we only have $\Delta(n) = o(1)$, all learnable problems are approximately privately learnable, and when we have $\Delta(n) = \{0\}$, only a strict subset of these problems is privately learnable. The following result establishes that when $\delta(n)$ needs to go to 0 with a sufficiently fast rate, there is separation between learnability and approximately private learnability.

Proposition 26 *Let $\Delta(n) = \{\delta(n) \mid \delta(n) \leq \tilde{\delta}(n)\}$ for some sequence $\tilde{\delta}(n) \rightarrow 0$. The following statements are true.*

- All learnable problems are $\Delta(n)$ -approximately privately learnable, if $\tilde{\delta}(n) = \omega(1/n)$.
- There exists a problem that is learnable but not $\Delta(n)$ -approximately privately learnable, if $\tilde{\delta}(n) \leq \frac{\exp(-\epsilon(n^2)n^2)}{n}$.

Proof The first claim follows from the same argument in Lemma 22. If a problem is learnable, there exists a universally consistent learning algorithm \mathcal{A} . The algorithm that

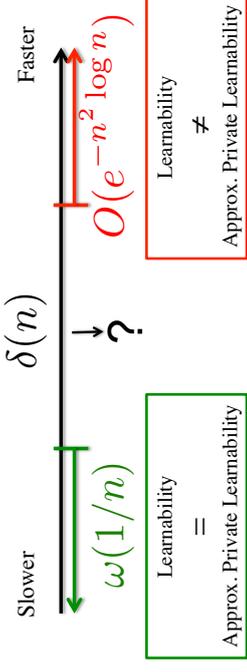


Figure 4: Illustration of Proposition 26 and the open problem.

applies \mathcal{A} on a $\tilde{\delta}(n)$ -fraction random subsample of the dataset is $(0, \tilde{\delta}(n))$ -DP and universally consistent with rate $\xi(n\tilde{\delta}(n))$. Since $\tilde{\delta}(n) = \omega(1/n)$, $n\tilde{\delta}(n) \rightarrow \infty$.

We now show that when we require a fast decaying $\delta(n)$, then suddenly the example in Section 3.3 due to Chaudhuri and Hsu (2011) becomes not approximately privately learnable even for (ϵ, δ) -DP. Let Z, Z' be two completely different data sets, by repeatedly applying the definition of (ϵ, δ) -DP, for any set $\mathcal{S} \subset \mathcal{H}$

$$\mathbb{P}(\mathcal{A}(Z) \in \mathcal{S}) \leq e^{\epsilon} \mathbb{P}(\mathcal{A}(Z) \in \mathcal{S}) + \sum_{i=1}^n e^{(i-1)\epsilon} \delta \leq e^{\epsilon} \mathbb{P}(\mathcal{A}(Z') \in \mathcal{S}) + ne^{(n-1)\epsilon} \delta.$$

When we shift the inequality around, we get

$$\mathbb{P}(\mathcal{A}(Z') \in \mathcal{S}) \leq e^{-\epsilon} \mathbb{P}(\mathcal{A}(Z') \in \mathcal{S}) - e^{-\epsilon} n \delta.$$

Consider the same example in Section 3.3 where we hope to learn a threshold on $[0, 1]$. Assuming there exists an algorithm \mathcal{A} that is universally AERM and $(\epsilon(n), \delta(n))$ -DP for $\epsilon(n) < \infty$ and $\delta(n) \leq 0.4ne^{-\epsilon n}$.

Everything up to (4) remains exactly the same. Now, apply the above implication of (ϵ, δ) -DP, we can replace (4) for each $i = 2, \dots, K$, by

$$\mathbb{P}(\mathcal{A}(Z_1) \in [h_i - \eta/3, h_i + \eta/3]) \geq \exp(-\epsilon n) \mathbb{P}(\mathcal{A}(Z_i) \in [h_i - \eta/3, h_i + \eta/3]) - n\delta(n).$$

Then (5) becomes

$$\mathbb{P}(\mathcal{A}(Z_1) \notin [h_1 - \eta/3, h_1 + \eta/3]) \geq K \exp(-\epsilon n) 0.9 - Ke^{-\epsilon} n \delta(n) \geq 0.9 \geq 0.5,$$

where the last inequality follows by $K > \exp(\epsilon n)$ and $\delta(n) \leq 0.4ne^{-\epsilon n}$. This yields the same contradiction to always AERM of \mathcal{A} on Z_1 , which requires $\mathbb{P}(\mathcal{A}(Z_1) \notin [h_1 - \eta/3, h_1 + \eta/3]) < 0.1$. Therefore, such AERM does not exist. By the contrapositive of Theorem 25, the problem is not approximately privately learnable for $\tilde{\delta}(n) \leq \frac{\exp(-\epsilon(n^2)n^2)}{n}$. \blacksquare

The bound can be further improved to $\exp(-\epsilon(n)/n)$ if we directly work with universal consistency on various distributions rather than through always AERM on specific data points. Even that is likely to be suboptimal as there might be more challenging problems and less favorable packings to consider.

The point of this exposition, however, is to illustrate that (ϵ, δ) -DP alone does not close the gap between learnability and private learnability. Additional relaxation on the specified rate of decay on δ does. We now know that the phase transition occurs when $\delta(n)$ is somewhere between $\Omega(\exp(-n^2 \log n))$ and $O(1/n)$; but there is still a substantial gap between the upper and lower bounds.³

7. Conclusion and future work

In this paper, we revisited the question “*What can we learned privately?*” and considered a broader class of statistical machine learning problems than those studied previously. Specifically, we characterized the learnability under privacy constraint by showing any privately learnable problems can be learned by a private algorithm that asymptotically minimizes the empirical risk for any data, and the problem is not privately learnable otherwise. This allows us to construct a conceptual procedure that privately learns any privately learnable problem. We also propose a relaxed notion of private learnability called private \mathfrak{D} -learnability, which requires the existence of an algorithm that is consistent for any the distribution within a class of distributions \mathfrak{D} . We characterized private \mathfrak{D} -learnability too with a weaker notion of AERM. For problems that can be formulated as penalized empirical risk minimization, we provide a sampling algorithm with a set of meaningful sufficient conditions on the geometry of the hypothesis space and demonstrate that it covers a large class of problems. In addition, we further extended the characterization to learnability under (ϵ, δ) -differential privacy and provided a preliminary analysis which establishes the existence of a phase transition from all learnable problems being approximately private learnable to some learnable problems being not approximately private learnable at some non-trivial rate of decay on $\delta(n)$.

Future work includes understanding the conditions under which privacy and AERM are contradictory (recall that we only have one example on learning thresholding functions due to Chandhuri and Hsu 2011), characterizing the rate of convergence, searching for practical algorithms that generically learns all privately learnable problems, and better understanding the gap between learnability and approximate private learnability.

3. After the paper was accepted for publication, we became aware that the phase transition occurs sharply at $O(1/n)$. The result follows from a sharp lower bound of sample complexity in learning threshold functions in Bun (2016, Theorem 4.5.2), which improves over a previously published result that requires $O(n^{-1-\epsilon})$ for any $\epsilon > 0$ in Bun et al. (2015). The consequence is that the general learning setting is hard for (ϵ, δ) -DP too unless δ becomes meaningfully large for privacy purposes.

Acknowledgment

We thank the AE and the anonymous reviewers for their comments that lead to significant improvement of this paper. The research was partially supported by NSF Award BCS-0941518 to the Department of Statistics at Carnegie Mellon University, and a grant by Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

Appendix A. Proofs of technical results

In this appendix, we provide detailed proofs to the technical results that in the main text.

A.1 Privacy in subsampling

Proof [Proof of Lemma 4] Let \mathcal{A} be the consistent ϵ -DP algorithm. Consider \mathcal{A}' that apply \mathcal{A} to a random subsample of $\lfloor \sqrt{n} \rfloor$ data points. By Lemma 27 with $\gamma = \frac{\lfloor \sqrt{n} \rfloor}{n} \leq \frac{1}{\sqrt{n}}$, we get the privacy claim. For the consistency claim, note that the given sample is an iid sample of size \sqrt{n} from the original distribution. ■

Lemma 27 (Subsampling theorem) *If Algorithm \mathcal{A} is ϵ -DP for $Z \in \mathcal{Z}^n$ for any $n = 1, 2, 3, \dots$, then the algorithm \mathcal{A}' that output the result of \mathcal{A} to a random subsample of size γn data points preserves $2\gamma(\epsilon^\epsilon - e^{-\gamma})$ -DP.*

Proof [Proof of Lemma 27 (Subsampling theorem)] This is a corollary of Lemma 4.4 in Beimel et al. (2014). To be self-contained, we reproduce the proof here in our notation.

Recall that \mathcal{A}' is the algorithm that first randomly subsample γn data points then apply \mathcal{A} . Let Z and Z' be any neighboring databases and assume they differ on the i th data point. Let $S \subset [n]$ be the indices of the random subset of the entries that are selected, and $\mathcal{R} \subset [n] \setminus \{i\}$ be a index size of size $\gamma n - 1$. We apply the law of total expectation twice and argue that for any adjacent Z, Z' , any event $E \subset \mathcal{H}$,

$$\begin{aligned} \mathbb{P}_{h \sim \mathcal{A}'(Z)}(h \in E) &= \gamma \mathbb{P}_{h \sim \mathcal{A}(Z_S)}(h \in E | i \in S) + (1 - \gamma) \mathbb{P}_{h \sim \mathcal{A}(Z_S)}(h \in E | i \notin S) \\ \mathbb{P}_{h \sim \mathcal{A}'(Z')} &= \gamma \mathbb{P}_{h \sim \mathcal{A}(Z'_S)}(h \in E | i \in S) + (1 - \gamma) \mathbb{P}_{h \sim \mathcal{A}(Z'_S)}(h \in E | i \notin S) \\ &= \sum_{\mathcal{R} \in [n] \setminus \{i\}} \mathbb{P}(\mathcal{R}) \left[\gamma \mathbb{P}_{h \sim \mathcal{A}(Z_S)}(h \in E | S = \mathcal{R} \cup \{i\}) + (1 - \gamma) \mathbb{P}_{h \sim \mathcal{A}(Z_S)}(h \in E | S = \mathcal{R} \cup \{j\}, j \neq i) \right] \\ &= \sum_{\mathcal{R} \in [n] \setminus \{i\}} \mathbb{P}(\mathcal{R}) \left[\gamma \mathbb{P}_{h \sim \mathcal{A}(Z_S)}(h \in E | S = \mathcal{R} \cup \{i\}) + (1 - \gamma) \mathbb{P}_{h \sim \mathcal{A}(Z_S)}(h \in E | S = \mathcal{R} \cup \{j\}, j \neq i) \right] \end{aligned}$$

By the given condition that \mathcal{A} is ϵ -DP, we can replace $\mathcal{R} \cup \{i\}$ with $\mathcal{R} \cup \{j\}$ for an arbitrary j bounded changes in the probability and the above likelihood ratio can be upper bounded by

$$\frac{(\gamma e^\epsilon + 1 - \gamma) \mathbb{P}_{\mathcal{R} \in [n] \setminus \{i\}} \left[\gamma \mathbb{P}_{h \sim \mathcal{A}(Z_S)}(h \in E | S = \mathcal{R} \cup \{j\}) \right]}{(\gamma e^{-\epsilon} + 1 - \gamma) \mathbb{P}_{\mathcal{R} \in [n] \setminus \{i\}} \left[\gamma \mathbb{P}_{h \sim \mathcal{A}(Z_S)}(h \in E | S = \mathcal{R} \cup \{j\}) \right]} = \frac{\gamma e^\epsilon + 1 - \gamma}{\gamma e^{-\epsilon} + 1 - \gamma} = \frac{1 + \gamma(e^\epsilon - 1)}{1 + \gamma(e^{-\epsilon} - 1)}.$$

By definition, the privacy loss of the algorithm \mathcal{A}' is therefore

$$\epsilon' \leq \log(1 + \gamma[e^\epsilon - 1]) - \log(1 + \gamma[e^{-\epsilon} - 1]).$$

Note that $\epsilon > 0$ implies that $-1 \leq e^{-\epsilon} - 1 < 0$ and $0 < e^\epsilon - 1 < \infty$. The result follows by applying the property of the natural logarithm:

$$\begin{aligned} \log(1+x) &\leq \frac{x}{2} + \frac{x}{1+x} && \text{for } 0 \leq x < \infty \\ \log(1+x) &\geq \frac{x}{2} + \frac{x}{1+x} \geq \frac{x}{1+x} && \text{for } -1 \leq x \leq 0 \end{aligned}$$

to upper bound the expression. \blacksquare

A.2 Characterization of private learnability

Privacy implies stability Lemma 8 says that an ϵ -differentially private algorithm is $(e^\epsilon - 1)$ -stable (and also $2e^\epsilon$ -stable if $\epsilon < 1$).

Proof [Proof of Lemma 8] Construct Z' by replacing an arbitrary data point in Z with z' and let the probability density/mass defined by $\mathcal{A}(Z)$ and $\mathcal{A}(Z')$ be $p(h)$ and $p'(h)$ respectively, then we can bound the stability as follows

$$\begin{aligned} & \left| \mathbb{E}_{h \sim \mathcal{A}(Z)} \ell(h, z) - \mathbb{E}_{h \sim \mathcal{A}(Z')} \ell(h, z) \right| \\ &= \left| \int_h \ell(h, z) p(h) dh - \int_h \ell(h, z) p'(h) dh \right| = \left| \int_h \ell(h, z) (p(h) - p'(h)) dh \right| \\ &\leq \sup_{h, z} |\ell(h, z)| \int_{p(h) \geq p'(h)} p(h) - p'(h) dh \leq 1 \cdot \int_{p(h) \geq p'(h)} p'(h) \left(\frac{p(h)}{p'(h)} - 1 \right) dh \\ &\leq (e^\epsilon - 1) \int_{p(h) \geq p'(h)} p'(h) dh \leq (e^\epsilon - 1). \end{aligned}$$

For $\epsilon < 1$ we have $\exp(\epsilon) - 1 < 2e$. \blacksquare

Stability + AERM \Rightarrow consistency

Theorem 28 (Randomized version of Shalev-Shwartz et al. 2010, Theorem 8) *If any algorithm is $\xi_1(n)$ -stable and $\xi_2(n)$ -AERM then it is consistent with rate $\xi(n) = \xi_1(n) + \xi_2(n)$.*

Proof

We will show the following the two steps as in Shalev-Shwartz et al. (2010)

1. Uniform RO stability \Rightarrow On average stability \Leftrightarrow On average generalization
2. AERM + On average generalization \Rightarrow consistency

The definition of these quantities is self-explanatory.

To show that ‘‘stability implies generalization’’, we have

$$\begin{aligned} & \left| \mathbb{E}_{Z \sim \mathcal{D}^n} \left(\mathbb{E}_{h \sim \mathcal{A}(Z)} R(h) - \mathbb{E}_{h \sim \mathcal{A}(Z)} \hat{R}(h, Z) \right) \right| \\ &= \left| \mathbb{E}_{Z \sim \mathcal{D}^n} \left(\mathbb{E}_{z \sim \mathcal{D}} \mathbb{E}_{h \sim \mathcal{A}(Z)} \ell(h, z) - \frac{1}{n} \mathbb{E}_{h \sim \mathcal{A}(Z)} \sum_{i=1}^n \ell(h, z_i) \right) \right| \\ &= \left| \mathbb{E}_{Z \sim \mathcal{D}^n, \{z'_1, \dots, z'_n\} \sim \mathcal{D}^n} \left(\frac{1}{n} \sum_{i=1}^n \mathbb{E}_{h \sim \mathcal{A}(Z)} \ell(h, z'_i) - \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{h \sim \mathcal{A}(Z^{(i)})} \ell(h, z'_i) \right) \right| \\ &\leq \sup_{Z, Z^{(i)} \in \mathcal{Z}^n, d(Z, Z^{(i)})=1, z' \in \mathcal{Z}} \left| \mathbb{E}_{h \sim \mathcal{A}(Z)} \ell(h, z') - \mathbb{E}_{h \sim \mathcal{A}(Z^{(i)})} \ell(h, z') \right| \leq \xi_1(n), \end{aligned}$$

where $Z^{(i)}$ is obtained by replacing the i th entry of Z with z'_i . Next, we show that ‘‘generalization and AERM implies consistency’’. Let $h^* \in \arg \inf_{h \in \mathcal{H}} R(h)$. By definition, we have $\mathbb{E}_{Z \sim \mathcal{D}^n} \hat{R}(h^*, Z) = R^*$. It follows that

$$\begin{aligned} & \mathbb{E}_{Z \sim \mathcal{D}^n} [\mathbb{E}_{h \in \mathcal{A}(Z)} \hat{R}(h) - R^*] = \mathbb{E}_{Z \sim \mathcal{D}^n} [\mathbb{E}_{h \in \mathcal{A}(Z)} R(h) - \hat{R}(h^*, Z)] \\ &= \mathbb{E}_{Z \sim \mathcal{D}^n} [\mathbb{E}_{h \in \mathcal{A}(Z)} \hat{R}(h, Z) - \mathbb{E}_{h \in \mathcal{A}(Z)} \hat{R}(h, Z)] + \mathbb{E}_{Z \sim \mathcal{D}^n} [\mathbb{E}_{h \in \mathcal{A}(Z)} \hat{R}(h, Z) - \hat{R}(h^*, Z)] \\ &\leq \mathbb{E}_{Z \sim \mathcal{D}^n} [\mathbb{E}_{h \in \mathcal{A}(Z)} \hat{R}(h, Z) - \mathbb{E}_{h \in \mathcal{A}(Z)} R(h, Z)] + \mathbb{E}_{Z \sim \mathcal{D}^n} [\mathbb{E}_{h \in \mathcal{A}(Z)} \hat{R}(h, Z) - \hat{R}(h^*, Z)] \\ &\leq \xi_1(n) + \xi_2(n). \end{aligned}$$

Privacy + AERM \Rightarrow consistency **Proof** [Proof of Corollary 9] It follows by combining Lemma 8 and Theorem 28. \blacksquare

Necessity **Proof** [Proof of Lemma 10] We construct an algorithm \mathcal{A}' by subsampling the data points using a random subset of \sqrt{n} and then running \mathcal{A} . The privacy claim follows from Lemma 27 directly.

To prove the ‘‘always AERM’’ claim, we adapt the proof of Lemma 24 in Shalev-Shwartz et al. (2010). For any fixed data set $Z \in \mathcal{Z}^n$,

$$\begin{aligned} \hat{R}(\mathcal{A}'(Z), Z) - \hat{R}^*(Z) &= \mathbb{E}_{Z' \subset Z, |Z'| = \lfloor \sqrt{n} \rfloor} [\hat{R}(\mathcal{A}(Z'), Z) - \hat{R}^*(Z)] \\ &= \mathbb{E}_{Z' \sim \text{Unif}(Z), \lfloor \sqrt{n} \rfloor} [\hat{R}(\mathcal{A}(Z'), Z) - \hat{R}^*(Z)] \text{ no duplicates} \\ &\leq \frac{\mathbb{E}_{Z' \sim \text{Unif}(Z), \lfloor \sqrt{n} \rfloor} [\hat{R}(\mathcal{A}(Z'), Z) - \hat{R}^*(Z)]}{\mathbb{P}(\text{no duplicates})}, \end{aligned}$$

where $\text{Unif}(Z)$ is the uniform distribution defined on the n points in Z . We need to condition on the event that there are no duplicates for the second equality to hold because Z' is a subsample taken without replacements. The last inequality is by the law of total expectation and the non-negativity of the conditional expectation. But \mathbb{P} (no duplicates) = $\prod_{i=0}^{\lfloor \sqrt{n} \rfloor - 1} (1 - i/n) \geq 1 - \sum_{i=0}^{\lfloor \sqrt{n} \rfloor - 1} i/n \geq 1/2$. By universal consistency, \mathcal{A} is consistent on the discrete uniform distribution defined on Z , so

$$\hat{R}_{\mathcal{A}}(Z, Z) - \hat{R}^*(Z) \leq 2\mathbb{E}_{Z' \sim \text{Unif}(Z)^{[n]}} [\hat{R}(\mathcal{A}(Z'), Z) - \hat{R}^*(Z)] \leq 2\xi(\sqrt{n}).$$

It is obvious that \mathcal{A}' is consistent with rate \sqrt{n} as it applies \mathcal{A} on a random sample of size \sqrt{n} . By Lemma 4, \mathcal{A}' is $2n^{-1/2}(e^\epsilon - e^{-\epsilon})$ differentially private. By Corollary 9, the new algorithm \mathcal{A}' is universally consistent. ■

A.3 Proofs for Section 3.3

Proof [Proof of Proposition 11] If $\mathcal{A}(Z)$ is a continuous distribution, we can pick $h \in \mathcal{H}$ at any point where $\mathcal{A}(Z)$ has finite density and set $\mathcal{A}'(Z) | z \in Z$ to be h with probability $1/n$ and the same as $\mathcal{A}(Z)$ with probability $1 - 1/n$. This breaks privacy because conditioned on two databases with z or without z , \mathcal{A} , the probability ratio of outputting h is ∞ .

If $\mathcal{A}(Z)$ is a discrete distribution or a mixed distribution, it must have the same support of the point mass for all Z . Otherwise it violates DP because we need $\frac{\mathbb{P}_{h \in \mathcal{A}(Z)(h)}}{\mathbb{P}_{h \in \mathcal{A}(Z')(h)}} \leq \exp(\epsilon n)$ for any $Z, Z' \in \mathcal{Z}^n$. Specifically, let the discrete set of point mass be $\tilde{\mathcal{H}}$ if $\mathcal{H} \setminus \tilde{\mathcal{H}} \neq \emptyset$, then we can use the same technique as in the continuous case by adding a small probability $1/n$ on $\mathcal{H} \setminus \tilde{\mathcal{H}}$ when $z \in Z$.

If $\tilde{\mathcal{H}} = \mathcal{H}$, then \mathcal{H} is a discrete set, if $|\mathcal{H}| < n$, then by boundedness and Hoeffding, ERM is a deterministic algorithm that learns any learnable problem. On the other hand, if $|\mathcal{H}| > n$, then by pigeon hole principle, there always exists a hypothesis h that has probability smaller than $1/n$ in $\mathcal{A}(Z)$ for any $Z \in \mathcal{Z}^n$ and we can construct \mathcal{A}' by outputting a sample of $\mathcal{A}(Z)$ if z is not observed and outputting a sample $\mathcal{A}(Z) | \mathcal{A}(Z) \neq h$ whenever z is observed.

The consistency of \mathcal{A}' follows easily as its risk is at most $1/n$ larger than that of \mathcal{A} . ■

A.4 Proofs for characterization of private \mathfrak{D} -learnability

Proof [Proof of Lemma 14] Let \mathcal{A}' be the algorithm that applies \mathcal{A} to a random subsample of size $\lfloor \sqrt{n} \rfloor$. If we can show that, for any $\mathcal{D} \in \mathfrak{D}$,

- (a) the empirical risk of \mathcal{A}' converges to the optimal population risk R^* in expectation;
- (b) the empirical risk of the ERM learning rule also converges to R^* in expectation,

then by triangle inequality, the empirical risk of \mathcal{A}' must also converge to the empirical risk of ERM, i.e., \mathcal{A}' is \mathfrak{D} -universal AERM.

We will start with (a). For any distribution $\mathcal{D} \in \mathfrak{D}$, we have

$$\begin{aligned} \mathbb{E}_{Z' \sim \mathcal{D}^n} \hat{R}(\mathcal{A}'(Z), Z) &= \mathbb{E}_{Z' \sim \mathcal{D}^n} \left[\mathbb{E}_{Z' \subset Z, |Z'| = \lfloor \sqrt{n} \rfloor} \hat{R}(\mathcal{A}(Z'), Z) \right] \\ &= \mathbb{E}_{Z' \sim \mathcal{D}^n} \left[\frac{\lfloor \sqrt{n} \rfloor}{n} \hat{R}(\mathcal{A}(Z'), Z') + \mathbb{E}_{Z'' \sim \mathcal{D}^{n - \lfloor \sqrt{n} \rfloor}} \left(\frac{n - \lfloor \sqrt{n} \rfloor}{n} \hat{R}(\mathcal{A}(Z''), Z'') \right) \right] \\ &= \mathbb{E}_{Z' \sim \mathcal{D}^n} \left[\frac{\lfloor \sqrt{n} \rfloor}{n} \hat{R}(\mathcal{A}(Z'), Z') + \frac{n - \lfloor \sqrt{n} \rfloor}{n} R(\mathcal{A}(Z'')) \right] \leq \frac{1}{\sqrt{n}} + R^* + \xi(\sqrt{n}). \end{aligned} \quad (14)$$

The last inequality uses the boundedness of the loss function to get $\hat{R}(\mathcal{A}(Z'), Z') \leq 1$ and the \mathfrak{D} -consistency of \mathcal{A} to bound the excess risk of $\mathbb{E}_{Z''} R(\mathcal{A}(Z''))$.

To show (b), we need to exploit the assumption that the problem is (non-privately) learnable. By Shalev-Shwartz et al. (2010, Theorem 7), the problem being learnable implies that there exists a universally consistent algorithm \mathcal{B} (not restricted to \mathfrak{D}), that is universally AERM with rate $3\xi'(n^{\frac{1}{4}}) + \frac{2}{\sqrt{n}}$ and stable with rate $\frac{2}{\sqrt{n}}$. Moreover, by Shalev-Shwartz et al. (2010, Theorem 8), \mathcal{B} 's stability and AERM implies that \mathcal{B} is also generalizing, with rate $6\xi'(n^{\frac{1}{4}}) + \frac{18}{\sqrt{n}}$. Here the term “generalizing” means that the empirical risk is close to the population risk. Therefore, we can establish (b) via the following chain of approximations

$$\mathbb{E}_{Z \sim \mathcal{D}^n} \hat{R}^*(Z) \stackrel{\text{Generalization of } \mathcal{B}}{\approx} \mathbb{E}_{Z \sim \mathcal{D}^n} \hat{R}(\mathcal{B}(Z), Z) \stackrel{\text{Consistency of } \mathcal{B}}{\approx} R^* \stackrel{\text{AERM of } \mathcal{B}}{\approx} R^*.$$

More precisely,

$$\begin{aligned} & \left| \mathbb{E}_{Z \sim \mathcal{D}^n} \hat{R}^*(Z) - R^* \right| \\ & \leq \left| \mathbb{E}_{Z \sim \mathcal{D}^n} \hat{R}^*(Z) - \mathbb{E}_{Z \sim \mathcal{D}^n} \hat{R} \right| + \left| \mathbb{E}_{Z \sim \mathcal{D}^n} \hat{R} - R(\mathcal{B}(Z), Z) \right| + |R(\mathcal{B}(Z), Z) - R^*| \\ & \leq [3\xi'(n^{\frac{1}{4}}) + \frac{8}{\sqrt{n}}] + [6\xi'(n^{\frac{1}{4}}) + \frac{18}{\sqrt{n}}] + [3\xi'(n^{\frac{1}{4}}) + \frac{10}{\sqrt{n}}] = 12\xi'(n^{\frac{1}{4}}) + \frac{36}{\sqrt{n}}. \end{aligned} \quad (15)$$

Combine (14) and (15), we obtain the AERM of \mathcal{A}' with rate $12\xi'(n^{1/4}) + \frac{37}{\sqrt{n}} + \xi(\sqrt{n})$ as required. The privacy of \mathcal{A}' follows from Lemma 27. ■

A.5 Proof for Theorem 17

We first present the proof for Theorem 17. Recall that the roadmap of the proof is summarized in Figure 3.

For readability, we denote $\epsilon(n)$ by simply ϵ .

Recall that the objective function is $F(h, Z) = \frac{1}{n} \sum_{i=1}^n \ell(h, z_i) + g_n(h)$ and the corresponding utility function $q(h, Z) = -F(h, Z)$. By the boundedness assumption, it is easy to show that if we replace one data point in any Z with something else, then sensitivity

$$\Delta q = \sup_{h \in \mathcal{H}, d(Z, Z')=1} |q(Z, h) - q(Z', h)| \leq \frac{2}{n}. \quad (16)$$

Then by McSherry and Talwar (2007, Theorem 6), Algorithm 1 that outputs $h \in \mathcal{H}$ with $\mathbb{P}(h) \propto \exp(\frac{\epsilon}{2\Delta q} q(h, Z))$ naturally ensures ϵ -differential privacy.

Denote shorthand $F^* := \inf_{h \in \mathcal{H}} F(Z, h)$ and $q^* := -F^*$, we can state an analog of the utility theorem of the exponential mechanism in (McSherry and Talwar, 2007).

Lemma 29 (Utility) *Assuming $\epsilon < \log n$ (otherwise the privacy protection is meaningless anyway), if assumption A1, A2 hold for distribution \mathcal{D} , then*

$$\mathbb{E}_{Z \sim \mathcal{D}^n} \mathbb{E}_{h \sim \mathcal{A}(Z)} q(Z, h) \geq -\mathbb{E}_{Z \sim \mathcal{D}^n} F^* - \frac{9[(\rho+2)\log n + \log K]}{n\epsilon}. \quad (17)$$

Proof By the boundedness of ℓ and g

$$q(Z, h) = -\frac{1}{n} \sum_{i=1}^n \ell(h, z_i) - g_n(h) \geq -(1 + \zeta(n)).$$

By Lemma 7 in McSherry and Talwar (2007) (translated to our case),

$$\mathbb{P}_{h \sim \mathcal{A}(Z)} [q(Z, h) < -F^* - 2t] \leq \frac{\mu(\mathcal{H})}{\mu(\mathcal{S}_t)} e^{-\frac{2\epsilon t}{2\Delta q}}, \quad (18)$$

Apply (16), take expectation over the data distribution on both sides, and applying assumption A2, we get

$$\mathbb{E}_{Z \sim \mathcal{D}^n} \mathbb{P}_{h \sim \mathcal{A}(Z)} [q(Z, h) < -F^* - 2t] \leq Kt^{-\rho} e^{-\frac{\epsilon t}{4}} = e^{-\frac{\epsilon t}{4} + \log K - \rho \log t} := e^{-\gamma}. \quad (19)$$

Take $t = \frac{4[(\rho+2)\log n + \log(K)]}{\epsilon n}$, by the assumption that $\epsilon < \log n$, we get $\log(nt) > 0$. Substitute t into the expression of γ we obtain

$$\gamma = \frac{\epsilon n}{4} t - \log K + \rho \log t = 2 \log n + \rho \log(nt) \geq 2 \log n,$$

and therefore

$$\mathbb{E}_{Z \sim \mathcal{D}^n} \mathbb{P}_{h \sim \mathcal{A}(Z)} [q(Z, h) < -F^* - 2t] \leq n^{-2}.$$

Denote $\mathbb{P}_{h \sim \mathcal{A}(Z)} [q(Z, h) < -F^* - 2t] =: p$, we can then bound the expectation from below as follows:

$$\begin{aligned} \mathbb{E}_{Z \sim \mathcal{D}^n} \mathbb{E}_{h \sim \mathcal{A}(Z)} q(Z, h) &\geq \mathbb{E}_{Z \sim \mathcal{D}^n} (-F^* - 2t)(1-p) + \min_{h \in \mathcal{H}, Z \in \mathcal{Z}^n} q(Z, h) \mathbb{E}_{Z \sim \mathcal{D}^n} p \\ &\geq \mathbb{E}_{Z \sim \mathcal{D}^n} (-F^* - 2t) + (-1 - \zeta(n)) n^{-2} \\ &\geq -\mathbb{E}_{Z \sim \mathcal{D}^n} F^* - \frac{8[(\rho+2)\log n + \log(K)]}{\epsilon n} - (1 + \zeta(n)) n^{-2} \\ &\geq -\mathbb{E}_{Z \sim \mathcal{D}^n} F^* - \frac{9[(\rho+2)\log n + \log(K)]}{\epsilon n}. \end{aligned}$$

Now we can say something about the learning problem. In particular, the AERM follows directly from the utility result and stability follows from the definition of differential privacy. \blacksquare

Lemma 30 (Universal AERM) *Assume A1 and A2, and $\epsilon \leq \log n$ (so Lemma 29 holds), then*

$$\mathbb{E}_{Z \sim \mathcal{D}^n} [\mathbb{E}_{h \sim \mathcal{A}(Z)} \hat{R}(h, Z) - \hat{R}^*(Z)] \leq \frac{9[(\rho+2)\log n + \log(1/K)]}{n\epsilon} + \zeta(n).$$

Proof This is a simple consequence of boundedness and Lemma 29.

$$\begin{aligned} &\mathbb{E}_{Z \sim \mathcal{D}^n} [\mathbb{E}_{h \sim \mathcal{A}(Z)} \hat{R}(h, Z) - \hat{R}^*(Z)] \\ &= \mathbb{E}_{Z \sim \mathcal{D}^n} \mathbb{E}_{h \sim \mathcal{A}(Z)} \frac{1}{n} \sum_{i=1}^n \ell(h, z_i) - \mathbb{E}_{Z \sim \mathcal{D}^n} \inf_h \frac{1}{n} \sum_{i=1}^n \ell(h, z_i) \\ &\leq \mathbb{E}_{Z \sim \mathcal{D}^n} \mathbb{E}_{h \sim \mathcal{A}(Z)} \left[\frac{1}{n} \sum_{i=1}^n \ell(h, z_i) + g_n(h) \right] - \mathbb{E}_{h \sim \mathcal{A}(Z)} g_n(h) \\ &\quad - \mathbb{E}_{Z \sim \mathcal{D}^n} \inf_h \left[\frac{1}{n} \sum_{i=1}^n \ell(h, z_i) + g_n(h) \right] + \sup_h (g_n(h)) \\ &= \mathbb{E}_{Z \sim \mathcal{D}^n} (-F^* - \mathbb{E}_{h \sim \mathcal{A}(Z)} q(Z, h)) + \sup_h g_n(h) - \mathbb{E}_{h \sim \mathcal{A}(Z)} g_n(h) \\ &\leq \frac{9[(\rho+2)\log n + \log(1/K)]}{n\epsilon} + 2\zeta(n). \end{aligned}$$

The last step applies Lemma 29 and $\sup_h |g_n(h)| \leq \zeta(n)$ as in Assumption A2 by using the fact that $\sup_h g_n(h) - \mathbb{E} g_n(h) \leq 2 \sup_h |g_n(h)|$ for any distribution of h the expectation is taken over. \blacksquare

The above theorem shows that Algorithm 1 is asymptotic ERM. By Theorem 8, the fact that this algorithm is ϵ -differential private implies that it is 2ϵ -stable. Now the proof follows by applying Theorem 28 which says that stability and AERM of an algorithm certify its consistency. Noting that this holds for any distribution \mathcal{D} completes our proof for learnability in Theorem 17.

A.6 Proofs of other technical results

High confidence private learning. **Proof** [Proof of Theorem 18] The algorithm \mathcal{A} privately learns the problem with rate $\xi(n)$ implies that

$$\mathbb{E}_{Z \in \mathcal{D}^n} \mathbb{E}_{h \sim \mathcal{A}(Z)} R(h) - R^* \leq \xi(n).$$

Let $h \sim \mathcal{A}(Z)$ and $Z \sim \mathcal{D}^n$, by Markov's inequality, with probability at least $1 - 1/e$,

$$R(h) - R^* \leq e\xi(n).$$

If we split the data randomly into $a + 1$ parts of size $n/(a + 1)$ and run \mathcal{A} on the first a partitions, then we get $h_j \sim \mathcal{A}(Z_j)$. Then with probability at least $1 - (1/e)^a$, at least one of them has risk

$$\min_{j \in [a]} R(h_j) - R^* \leq \epsilon \xi\left(\frac{n}{a+1}\right). \quad (20)$$

Since the $(a + 1)$ th partition are iid data, and ℓ is bounded, we can apply Hoeffding's inequality and union bound, so that with probability $1 - \delta_1$ for all $j = 1, \dots, a + 1$

$$\hat{R}(h_j, Z_{a+1}) - R(h_j) \leq \sqrt{\frac{\log(2a/\delta_1)}{2n}}. \quad (21)$$

This means that if exponential mechanism picked the one with the best validation risk it will be almost as good as the one with the best risk. Assume h_1 is the one that achieves the best validation risk.

Now it remains to bound the probability that exponential mechanism pick an $h \in \{h_1, \dots, h_a\}$ that is much worse than h_1 .

Recall that the utility function is the negative validation risk which depends only on the last partition I_{a+1} .

$$q(X, h) = \frac{1}{n/(a+1)} \sum_{i \in I_{a+1}} \ell_i(z_i, h).$$

This is in fact a random function of the data because we are picking the the validation set I_{a+1} randomly from the data. Suppose we arbitrarily replace one data point j from the dataset, the distribution of the output of function $q(Z, h)$ is a mixture of the two cases: $j \in I_{a+1}$ and $j \notin I_{a+1}$. Since in the first case, $q(Z, h) = q(Z', h)$ for all h , sensitivity for this case is 0. In the second case, by the boundedness assumption, the sensitivity is at most $2(a + 1)/n$. For the exponential mechanism guarantee ϵ differential privacy, it suffices to take the sensitivity parameter to be $2(a + 1)/n$.

By the utility theorem of the exponential mechanism,

$$\mathbb{P}\left[\hat{R}(h) > \hat{R}(h_1) + \frac{8(q)\log n + \log a}{\epsilon n/(a+1)}\right] \leq n^{-\eta}. \quad (22)$$

Combine (20)(21) and(22) we get

$$\mathbb{P}\left[R(h) - R^* > \epsilon \xi\left(\frac{n}{a+1}\right) + \sqrt{\frac{\log(2a/\delta_1)}{2n}} + \frac{8(q)\log n + \log a}{\epsilon n/(a+1)}\right] \leq n^{-\eta} + \delta_1 + e^{-a}.$$

Now by appropriately choosing $\eta = \log(3/\delta)/\log n$, $a = \log(3/\delta)$, $\delta_1 = \delta/3$, we get

$$\mathbb{P}\left[R(h) - R^* > \epsilon \xi\left(\frac{n}{\log(3/\delta)+1}\right) + \sqrt{\frac{\log(2\log(3/\delta)) + \log(3/\delta)}{2n}} + \frac{8(\log(3/\delta) + \log \log(3/\delta))}{\epsilon n/(\log(3/\delta)+1)}\right] \leq \delta$$

combine the terms and take $\epsilon = \frac{\log(3/\delta)+1}{\sqrt{n}}$, we get the bound of the excess risk in the theorem.

To get the privacy claim, note that we are applying \mathcal{A} on disjoint partitions of the data so the privacy parameter does not aggregate. Take the worst over all partitions, we get the overall privacy loss $\max\left\{\epsilon \left(\frac{n}{\log(3/\delta)+1}\right), \frac{\log(3/\delta)+1}{\sqrt{n}}\right\}$ as stated in the theorem. ■

The Lipschitz example. **Proof** [Proof of Example 4] Let $h^* \in \arg\min_{h \in \mathcal{H}} F(Z, h)$, the Lipschitz condition dictates that for any h ,

$$|F(h) - F(h^*)| \leq L\|h - h^*\|_p.$$

Choose a small enough $t < t_0$ such that h is in the small neighborhood of h^* , and we can construct a function \tilde{F} that within the sublevel set S_t , such that the above inequality (when we replace F with \tilde{F}) is equality, then for any $h \in S_{t_0}$, $F(h) \geq F(Z, h)$. Verify that the sublevel set of $\tilde{F}(h)$, denoted by \tilde{S}_t , always contains S_t . In addition, we can compute the measure $\mu(\tilde{S}_t)$ explicitly, since the function is a cone and

$$L\|h - h^*\|_p = |\tilde{F}(h) - \tilde{F}(h^*)| = \tilde{F}(h) - \tilde{F}(h^*) \leq t,$$

therefore

$$\tilde{S}_t = \{h \mid L\|h - h^*\|_p \leq t\}.$$

Since \mathcal{H} is β_p -regular, $\mu(B \cap \mathcal{H}) \geq \beta_p \mu(B)$ for any ℓ_p ball $B \subset \mathbb{R}^d$, the measure of the sublevel set can be lower bounded by β_p times the volume of the ℓ_p ball with radius t/L and since $\tilde{S}_t \subseteq S_t$, we have

$$\mu(\tilde{S}_t) \geq \mu(S_t) \geq \beta_p \mu(B(t/L)) = \beta_p (t/L)^d$$

as required. ■

Appendix B. Alternative proof of Corollary 9 via Dwork et al. (2015b), Theorem 7)

In this Appendix, we describe how the results in Dwork et al. (2015b) can be used to obtain the forward direction of our characterization without going through a stability argument. We first restate the result here in our notation:

Lemma 31 (Theorem 7 in Dwork et al. 2015b) *Let \mathcal{B} be an ϵ -DP algorithm such that given a dataset Z , \mathcal{B} outputs a function from \mathcal{Z} to $[0, 1]$. For any distribution \mathcal{D} over \mathcal{Z} and random variable $Z \sim \mathcal{D}^n$, we let $\phi \sim \mathcal{B}(Z)$. Then for any $\beta > 0$, $\tau > 0$ and $n \geq 12 \log(4/\beta)/\tau^2$, setting $\epsilon < \tau/2$ ensures*

$$\mathbb{P}_{\phi \sim \mathcal{B}(Z), Z \sim \mathcal{D}^n} \left[\left| \mathbb{E}_{z \sim \mathcal{D}} \phi(z) - \frac{1}{n} \sum_{z \in Z} \phi(z) \right| \geq \tau \right] \leq \beta.$$

This lemma was originally stated to prove the claim that privately generated mechanisms for answering statistical queries always generalize.

For statistical learning problems, we can simply take the statistical query ϕ to be the loss function $\ell(h, \cdot)$ parameterized by $h \in \mathcal{H}$. If an algorithm \mathcal{A} that samples from a distribution on \mathcal{H} upon observing data Z is ϵ -DP, then $\mathcal{B} : Z \rightarrow \ell(\mathcal{A}(Z), \cdot)$ is also ϵ -DP. The result therefore reduces to that the empirical risk and population risk are close with high probability. Due to the boundedness assumption, we can translate the high probability result to the expectation form, which verifies the definition of “generalization”.

However, “generalization” alone still does not imply “consistency”, as we also need

$$\mathbb{E}_{\phi \sim \mathcal{B}(Z)} \frac{1}{n} \sum_{z \in Z} \phi(z) \rightarrow R^* = \min_{\phi \in \Phi} \mathbb{E}_{z \sim \mathcal{D}} \phi(z)$$

as Z gets large, which does not hold for all DP-output ϕ . But when $\phi = \ell(h, \cdot)$, it can be obtained if we assume \mathcal{A} is AERM. This is shown via the following inequality

$$\mathbb{E}_{Z \in \mathcal{D}^n} \mathbb{E}_{\phi \sim \mathcal{B}(Z)} \frac{1}{n} \sum_{z \in Z} \phi(z) \rightarrow \mathbb{E}_{Z \in \mathcal{D}^n} \min_{\phi \in \Phi} \frac{1}{n} \sum_{z \in Z} \phi(z) \leq \mathbb{E}_{Z \in \mathcal{D}^n} \frac{1}{n} \sum_{z \in Z} \phi^*(z) = \mathbb{E} \phi^*(z) = R^*,$$

where $\phi^* = \ell(h^*, \cdot)$ and h^* is an optimal hypothesis function. This wraps up the proof of consistency.

The above proof of “consistency” via Lemma 31 and “AERM”, however, leads to a looser bound comparing to our result (Corollary 9) when the additional assumption on n and τ (equivalently ϵ) is active, i.e., when $\frac{\epsilon(n)}{\log(1/\epsilon(n))} < O\left(\frac{1}{\sqrt{n}}\right)$. In this case it only implies a $\xi(n) + \frac{\log n}{\sqrt{n}}$ bound due to that ϵ -DP implies ϵ' -DP for any $\epsilon' > \epsilon$. Our proof of Corollary 9 is considerably simpler and more general in that it does not require any assumption on the number of data points n .

This can easily lead to worse overall error bound for very simple learning problems with sufficiently fast rate. For example, in the problem of learning the mean of $X \in [0, 1]$, let the loss function be $|x - h|^{10}$. Consider the $\epsilon(n)$ -DP algorithm that outputs $\text{ERM} + \text{Laplace}\left(\frac{2}{\epsilon(n)n}\right)$ where $\epsilon(n)$ is chosen to be $n^{-9/10}$. This algorithm is AERM with rate $\xi(n) = \frac{10 \cdot 2!}{(\epsilon(n)n)^{10}} = O(n^{-1})$. By Corollary 9 we get an overall rate of $O(n^{-9/10})$ while through Lemma 31 and the argument that follows, we only get $O(n^{-1/2})$.

References

- David Applegate and Ravi Kannan. Sampling and integration of near log-concave functions. In *ACM Symposium on Theory of Computing (STOC-91)*, pages 156–163, 1991.
- Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization, revisited. *arXiv preprint arXiv:1405.7085*, 2014.
- Raef Bassily, Kobbi Nissim, Adam Smith, Thomas Steinke, Uri Stemmer, and Jonathan Ullman. Algorithmic stability for adaptive data analysis. *arXiv preprint arXiv:1511.02513*, 2015.

- Amos Beimel, Kobbi Nissim, and Uri Stemmer. Characterizing the sample complexity of private learners. In *Conference on Innovations in Theoretical Computer Science (ITCS-13)*, pages 97–110. ACM, 2013a.
- Amos Beimel, Kobbi Nissim, and Uri Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 363–378. Springer, 2013b.
- Amos Beimel, Hai Brenner, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. Bounds on the sample complexity for private learning and private data release. *Machine learning*, 94(3):401–437, 2014.
- Avrim Blum and Moritz Hardt. The ladder: A reliable leaderboard for machine learning competitions. In *International Conference on Machine Learning (ICML-15)*, pages 1006–1014, 2015.
- Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.
- Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil Vadhan. Differentially private release and learning of threshold functions. In *IEEE Symposium on Foundations of Computer Science (FOCS-15)*, pages 634–649. IEEE, 2015.
- Mark Mar Bun. *New Separations in the Complexity of Differential Privacy*. PhD thesis, Harvard University Cambridge, Massachusetts, 2016.
- Kamalika Chaudhuri and Daniel Hsu. Sample complexity bounds for differentially private learning. In *Conference on Learning Theory (COLT-11)*, volume 19, pages 155–186, 2011.
- Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12:1069–1109, 2011.
- Cynthia Dwork. Differential privacy. In *Automata, Languages and Programming*, pages 1–12. Springer, 2006.
- Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *ACM Symposium on Theory of Computing (STOC-09)*, pages 371–380. ACM, 2009.
- Cynthia Dwork, Krishnamurthi Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology-EUROCRYPT 2006*, pages 486–503. Springer, 2006a.
- Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography*, pages 265–284. Springer, 2006b.
- Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015a.
- Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. Preserving statistical validity in adaptive data analysis. In *ACM Symposium on Theory of Computing (STOC-15)*, pages 117–126. ACM, 2015b.
- Moritz Hardt and Jonathan Ullman. Preventing false discovery in interactive data analysis is hard. In *IEEE Symposium on Foundations of Computer Science (FOCS-14)*, pages 454–463. IEEE, 2014.

- David Haussler. Decision theoretic generalizations of the pac model for neural net and other learning applications. *Information and computation*, 100(1):78–150, 1992.
- Prateek Jain and Abhradeep Thakurta. Differentially private learning with kernels. In *International Conference on Machine Learning (ICML-13)*, pages 118–126, 2013.
- Shiva Prasad Kasiviswanathan, Homim K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.
- Michael Kearns and Dana Ron. Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. *Neural Computation*, 11(6):1427–1453, 1999.
- Michael J Kearns, Robert E Schapire, and Linda M Selie. Toward efficient agnostic learning. In *Workshop on Computational learning theory (COLT-92)*, pages 341–352. ACM, 1992.
- Daniel Kifer, Adam Smith, and Abhradeep Thakurta. Private convex empirical risk minimization and high-dimensional regression. *Journal of Machine Learning Research*, 1:41, 2012.
- Jing Lei. Differentially private m -estimators. In *Advances in Neural Information Processing Systems (NIPS-11)*, pages 361–369, 2011.
- Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *IEEE Symposium on Foundations of Computer Science, 2007 (FOCS-07)*, pages 94–103, 2007.
- Sayan Mukherjee, Partha Niyogi, Tomaso Poggio, and Ryan Rifkin. Learning theory: Stability is sufficient for generalization and necessary for consistency of empirical risk minimization. *Advances in Computational Mathematics*, 25(1-3):161–193, 2006.
- Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NIPS-07)*, pages 1177–1184, 2007.
- Robert E Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.
- Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, 11:2635–2670, 2010.
- Adam Smith. Privacy-preserving statistical estimation with optimal convergence rates. In *ACM Symposium on Theory of Computing (STOC-11)*, pages 813–822, 2011.
- Abhradeep Guha Thakurta and Adam Smith. Differentially private feature selection via stability arguments, and the robustness of the lasso. In *Conference on Learning Theory (COLT-13)*, pages 819–850, 2013.
- Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.
- Vladimir Vapnik. *Statistical Learning Theory*. Wiley, 1998.
- Vladimir N Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.
- Yu-Xiang Wang, Stephen E Fienberg, and Alex Smola. Privacy for free: Posterior sampling and stochastic gradient monte carlo. In *International Conference on Machine Learning (ICML-15)*, 2015.
- Larry Wasserman and Shuheng Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489):375–389, 2010.
- Fei Yu, Stephen E Fienberg, Aleksandra B Slavković, and Caroline Uhler. Scalable privacy-preserving data sharing methodology for genome-wide association studies. *Journal of biomedical informatics*, 50:133–141, 2014.

fastFM: A Library for Factorization Machines

Immanuel Bayer

University of Konstanz
78437 Konstanz, Germany

IMMANUEL.BAYER@UNI-KONSTANZ.DE

Editor: Cheng Soon Ong

Abstract

Factorization Machines (FM) are currently only used in a narrow range of applications and are not yet part of the standard machine learning toolbox, despite their great success in collaborative filtering and click-through rate prediction. However, Factorization Machines are a general model to deal with sparse and high dimensional features. Our Factorization Machine implementation (*fastFM*) provides easy access to many solvers and supports regression, classification and ranking tasks. Such an implementation simplifies the use of FM for a wide range of applications. Therefore, our implementation has the potential to improve understanding of the FM model and drive new development.

Keywords: Python, MCMC, matrix factorization, context-aware recommendation

1. Introduction

This work aims to facilitate research for matrix factorization based machine learning (ML) models. Factorization Machines are able to express many different latent factor models and are widely used for collaborative filtering tasks (Rendle, 2012b). An important advantage of FM is that the model equation

$$w_0 \in \mathbb{R}, x, w \in \mathbb{R}^p, v_i \in \mathbb{R}^k$$

$$\hat{y}^{FM}(x) := w_0 + \sum_{i=1}^p u_i x_i + \sum_{i=1}^p \sum_{j>i}^p \langle v_i, v_j \rangle x_i x_j \quad (1)$$

conforms to the standard notation for vector based ML. FM learn a factorized coefficient $\langle v_i, v_j \rangle$ for each feature pair $x_i x_j$ (eq. 1). This makes it possible to model very sparse feature interactions, as for example, encoding a sample as $x = \{\dots, 0, \underbrace{1}_{x_1}, \dots, 0, \underbrace{1}_{x_2}, 0, \dots\}$ yields $\hat{y}^{FM}(x) = w_0 + u_i + w_j + v_i^T v_j$ which is equivalent to (biased) matrix factorization $R_{i,j} \approx b_0 + b_i + b_j + u_i^T v_j$ (Srebro et al., 2004). Please refer to Rendle (2012b) for more encoding examples. FM have been the top performing model in various machine learning competitions (Rendle and Schmidt-Thieme, 2009; Rendle, 2012a; Bayer and Rendle, 2013) with different objectives (e.g. What Do You Know? Challenge¹, EMI Music Hackathon²). *fastFM* includes solvers for regression, classification and ranking problems (see Table 1) and addresses the following needs of the research community: (i) easy interfacing for dynamic

1. <http://www.kaggle.com/c/WhatDoYouKnow>
2. <http://www.kaggle.com/c/MusicHackathon>

and interactive languages such as R, Python and Matlab; (ii) a Python interface allowing interactive work; (iii) a publicly available test suite strongly simplifying modifications or adding of new features; (iv) code is released under the **BSD-license** allowing the integration in (almost) any open source project.

2. Design Overview

The *fastFM* library has a multi layered software architecture (Figure 1) that separates the interface code from the performance critical parts (*fastFM-core*). The core contains the solvers, is written in C and can be used stand alone. Two user interfaces are available: a command line interface (CLI) and a Python interface. Cython (Behnel et al., 2011) is used to create a Python extension from the C library. Both, the Python and C interface, serve as reference implementation for bindings to additional languages.

2.1 fastFM-core

FM are usually applied to very sparse design matrices, often with a sparsity over 95 %, due to their ability to model interaction between very high dimensional categorical features. We use the standard compressed row storage (CRS) matrix format as underlying data structure and rely on the CXSparse³ library (Davis, 2006) for fast sparse matrix / vector operations. This simplifies the code and makes memory sharing between Python and C straight forward.

fastFM contains a test suite that is run on each commit to the GitHub repository via a continuous integration server⁴. Solvers are tested using state of the art techniques, such as Posterior Quantiles (Cook et al., 2006) for the MCMC sampler and Finite Differences for the SGD based solvers.

2.2 Solver and Loss Functions

fastFM provides a range of solvers for all supported tasks (Table 1). The MCMC solver implements the Bayesian Factorization Machine model (Freudenthaler et al., 2011) via Gibbs sampling. We use the pairwise Bayesian Personalized Ranking (BPR) loss (Rendle et al., 2009) for ranking. More details on the classification and regression solvers can be found in Rendle (2012b).

Task	Solver	Loss
Regression	ALS, MCMC, SGD	Square Loss
Classification	ALS, MCMC, SGD	Probit (MAP), Probit, Sigmoid
Ranking	SGD	BPR (Rendle et al., 2009)

Table 1: Supported solvers and tasks

3. CXSparse is LGPL licensed.
4. <https://travis-ci.org/ibayer/fastFM-core>

2.3 Python Interface

The Python interface is compatible with the API of the widely-used `scikit-learn` library (Pedregosa et al., 2011) which opens the library to a large user base. The following code snippet shows how to use MCMC sampling for an FM classifier and how to make predictions on new data.

```
fm = mcmc.FMClassification(init_std=0.01, rank=8)
y_pred = fm.fit_predict(X_train, y_train, X_test)
```

`fastFM` provides additional features such as warm starting a solver from a previous solution (see MCMC example).

```
fm = als.FMRegression(init_std=0.01, rank=8, l2_reg=2)
fm.fit(X_train, y_train)
```

3. Experiments

`libFM`⁵ is the reference implementation for FM and the only one that provides ALS and MCMC solver. Our experiments show, that the ALS and MCMC solver in `fastFM` compare favorable to `libFM` with respect to runtime (Figure 2) and are indistinguishable in terms of accuracy. The experiments have been conducted on the Movielens 10M data set using the original split with a fixed number of 200 iterations for all experiments. The x-axis indicates the number of latent factors (rank), and the y-axis the runtime in seconds. The plots show that the runtime scales linearly with the rank for both implementations. The code snippet

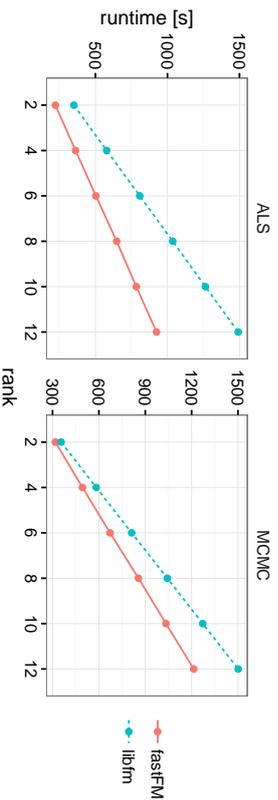


Figure 2: A runtime comparison between `fastFM` and `libFM` is shown. The evaluation is done on the Movielens 10M data set.

below shows how simple it is to write Python code that allows model inspection after every iteration. The induced Python function call overhead occurs only once per iteration and is therefore neglectable. This feature can be used for Bayesian Model Checking as demonstrated in Figure 3. The figure shows MCMC summary statistics for the first order hyper parameter σ_w . Please note that the MCMC solver uses Gaussian priors for the model parameter (Freudenthaler et al., 2011).

⁵ <http://libfm.org>

```
fm = mcmc.FMRegression(n_iter=0)
# initialize coefficients
fm.fit_predict(X_train, y_train, X_test)

for i in range(number_of_iterations):
    y_pred = fm.fit_predict(X_train, y_train, X_test, n_more_iter=1)
    # save, or modify (hyper) parameter
    print(fm.w_, fm.V_, fm.hyper_param_)
```

Many other analyses and experiments can be realized with a few lines of Python code without the need to read or recompile the performance critical C code.

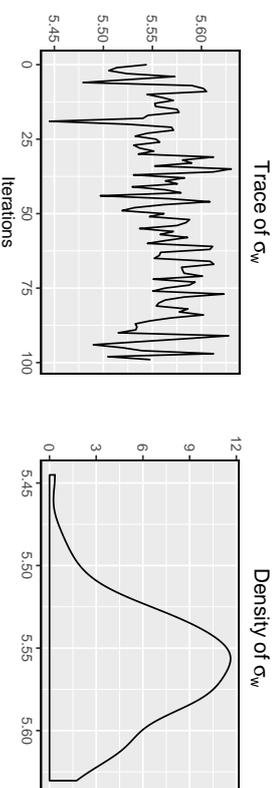


Figure 3: MCMC chain analysis and convergence diagnostics example for the hyperparameter σ_w evaluated on the Movielens 10M data set.

4. Related Work

Factorization Machines are available in the large scale machine learning libraries GraphLab (Low et al., 2014) and Bidmach (Canny and Zhao, 2013). The toolkit Svdfeatures by Chen et al. (2012) provides a general MF model that is similar to a FM. The implementations in GraphLab, Bidmach and Svdfeatures only support SGD solvers and don't provide a ranking loss. It's not our objective to replace these distributed machine learning frameworks: but to provide a FM implementation that is easy to use and easy to extend without sacrificing performance.

Acknowledgements

This work was supported by the DFG under grant Re 3311/2-1.

References

Immanuel Bayer and Steffen Rendle. Factor models for recommending given names. In *ECML/PKDD 2013 Discovery Challenge Workshop, part of the European Conference*

- on *Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 81–89, 2013.
- Stefan Behnel, Robert Bradshaw, Craig Citro, Lisandro Dalcin, Dag Sverre Seljebotn, and Kurt Smith. Cython: The best of both worlds. *Computing in Science & Engineering*, 13(2):31–39, 2011.
- John Canny and Huasha Zhao. Bidmach: Large-scale learning with zero memory allocation. In *BigLearn Workshop, NIPS*, 2013.
- Tianqi Chen, Weinan Zhang, Qixia Lu, Kailong Chen, Zhao Zheng, and Yong Yu. Svdfeature: a toolkit for feature-based collaborative filtering. *The Journal of Machine Learning Research*, 13(1):3619–3622, 2012.
- Samantha R Cook, Andrew Gelman, and Donald B Rubin. Validation of software for bayesian models using posterior quantiles. *Journal of Computational and Graphical Statistics*, 15(3), 2006.
- Timothy A Davis. *Direct methods for sparse linear systems*, volume 2. Siam, 2006.
- Christoph Freudenthaler, Lars Schmidt-thieme, and Steffen Rendle. Bayesian factorization machines. In *Proceedings of the NIPS Workshop on Sparse Representation and Low-rank Approximation*, 2011.
- Yucheng Low, Joseph E Gonzalez, Aapo Kyrola, Danny Bickson, Carlos E Guestrin, and Joseph Hellerstein. Graphlab: A new framework for parallel machine learning. *arXiv preprint arXiv:1408.2041*, 2014.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- Steffen Rendle. Social network and click-through prediction with factorization machines. In *KDD-Cup Workshop*, 2012a.
- Steffen Rendle. Factorization machines with libFM. *ACM Trans. Intell. Syst. Technol.*, 3(3):57:1–57:22, May 2012b. ISSN 2157-6904.
- Steffen Rendle and Lars Schmidt-Thieme. Factor models for tag recommendation in bibliography. In *ECML/PKDD 2008 Discovery Challenge Workshop, part of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 235–243, 2009.
- Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *UAI '09*, pages 452–461, Arlington, Virginia, United States, 2009.
- Nathan Srebro, Jason Rennie, and Tommi S Jaakkola. Maximum-margin matrix factorization. In *Advances in neural information processing systems*, pages 1329–1336, 2004.

The Factorized Self-Controlled Case Series Method: An Approach for Estimating the Effects of Many Drugs on Many Outcomes

Ramin Moghaddass

*Department of Industrial Engineering
University of Miami
Coral Gables, FL, USA*

RAMIN@MIAMI.EDU

Cynthia Rudin

*Department of Computer Science
Department of Electrical and Computer Engineering
Duke University
Durham, NC, USA*

CYNTHIA@CS.DUKE.EDU

David Madigan

*Department of Statistics
Columbia University
New York, NY, USA*

MADIGAN@STAT.COLUMBIA.EDU

Editor: Benjamin M. Marlin, C. David Page, and Suchi Sarin

Abstract

We provide a hierarchical Bayesian model for estimating the effects of transient drug exposures on a collection of health outcomes, where the effects of all drugs on all outcomes are estimated simultaneously. The method possesses properties that allow it to handle important challenges of dealing with large-scale longitudinal observational databases. In particular, this model is a generalization of the self-controlled case series (SCCS) method, meaning that certain patient specific baseline rates never need to be estimated. Further, this model is formulated with layers of latent factors, which substantially reduces the number of parameters and helps with interpretability by illuminating latent classes of drugs and outcomes. We believe our work is the first to consider multivariate SCCS (in the sense of multiple outcomes) and is the first to couple latent factor analysis with SCCS. We demonstrate the approach by estimating the effects of various time-sensitive insulin treatments for diabetics.

Keywords: Bayesian Analysis, Drug Safety, Self-Controlled Case Series, Matrix Factorization, Effect Size Estimation

1. Introduction

The medical community, the pharmaceutical industry, and health authorities are obligated to confirm that marketed medical products and prescription drugs have acceptable benefit-risk profiles; in fact, these entities have come under increasing scientific, regulatory, and public scrutiny to accurately estimate the effects of drugs. The increasing availability of large-scale longitudinal observational healthcare databases (LODs) opens up exciting new

opportunities to add to the evidence base concerning these issues, though the complexity and scale of some of the available databases presents interesting statistical and computational challenges. In what follows we focus on using longitudinal observational databases to make inferences about the effects of many drugs with respect to many outcomes simultaneously.

Many research studies have attempted to characterize the relationship between time-varying drug exposures and adverse events (AEs) related to health outcomes (e.g. in Madigan et al., 2011; Greene et al., 2011; Benchimol et al., 2013; Simpson et al., 2013; Chui et al., 2014) and the use of LODs to study *individual* drug-adverse effect combinations has become routine. The medical literature provides many examples and many different epidemiological and statistical approaches, often tailored to the specific drug and specific adverse effect. There is a major flaw in these approaches of estimating the effect of one drug on one outcome, which is that it is very clear that many drugs are closely related to each other (there are dozens of antibiotics for instance), and many health outcomes are closely related to each other (e.g., strokes, heart attacks, and other vascular diseases). In this work, we borrow strength across both drugs and outcomes in order to obtain better estimates for each individual drug and outcome. Since we are interested in the effects of drugs, and not in the patient-specific baseline rate of the outcome, we use the ideas of the self-controlled case series (SCCS) method of Farrington (1995), which is a conditional Poisson regression approach wherein each patient serves as his or her own control. The SCCS method has been widely applied, especially in vaccine studies (see the tutorial of Whitaker et al., 2006). SCCS controls for all fixed patient-level covariates but remains susceptible to time-varying confounding. The standard SCCS method focuses on one drug and one outcome. Simpson et al. (2013) introduced the high-dimensional multiple self-controlled case series (MSCCS) method that simultaneously provides effect estimates for multiple drugs and a single outcome. In fact, the MSCCS provides a self-controlled approach that can control for many time-varying covariates, drugs being a special case. Bayesian implementations of both SCCS and MSCCS provide significant advantages, especially in high-dimensional settings with thousands or even tens of thousands of drugs and outcomes and even larger numbers of interactions. Suchard et al. (2013a) and Madigan et al. (2014) describe large-scale empirical evaluations of SCCS and MSCCS in comparison with other standard methods for effect size estimation.

Neither SCCS nor MSCCS account for the fact that many drugs/treatments naturally form classes and therefore regression coefficients for drugs from within a single class might reasonably be modeled as arising exchangeably from a common prior distribution. Adverse events and health conditions can also be organized hierarchically, again affording an opportunity to “borrow strength” across related outcomes. For both drugs and outcomes, the hierarchy could extend to multiple levels. In what follows, we formalize these ideas within the framework of latent factor Bayesian hierarchical models.

Factor models, which have been traditionally used in behavioral sciences and bioinformatics, provide a flexible framework for modeling multivariate data via unobserved latent factors (e.g., Ghosh and Dumson, 2009; Carvalho et al., 2008). In this paper, we do not impose specific latent structure *a priori*. However, our approach can also be used for cases where classes of drugs and conditions are known *a priori*. We will show that the latent factor approach not only brings more interpretability to our model, but also can significantly contribute to reducing the computational complexity. To our knowledge, only a

few authors have previously considered matrix factorization-based data analysis techniques for drug safety and surveillance (for example, Ziftnik and Zupun 2014, for drug-induced liver injury prediction and Cobanoglu et al. 2013, for predicting drug-target interactions in neurobiological disorders, which are both very different from our study).

We introduce three models for predicting the effects of multiple drugs on multiple outcomes that use hierarchical Bayesian analysis. The first model (Model 0) does not use latent factors, and borrows strength across all drugs and outcomes. The second model (Model 1) uses one set of latent drugs and one set of latent outcomes, through a single matrix factorization. The third model (Model 2) uses two sets of latent factors, by factoring the matrix of coefficients into three matrices; one for converting drugs to latent drugs, another for converting outcomes to latent outcomes, and the third for modeling the effects of latent drugs on latent outcomes. By allowing for latent factors, the second and third models provide an increased level of interpretability; use fewer variables, and are thus more computationally efficient to estimate.

The rest of this paper is organized as follows: Section 2 provides an overview of the self-controlled case series (SCCS) method. In Sections 3, 4.1, 4.2, and 4.3 we describe the model and the Bayesian inference procedure. We then use a series of simulations in Section 5 to show that we can recover the true generating parameters from data. Finally, we demonstrate the approach in Section 6 for estimating the effects of various insulin treatments for diabetes. Our proposed methodology has broader applicability beyond estimating the effects of drugs considered in this paper.

2. Background: Overview of the Self-Controlled Case Series (SCCS)

The self-controlled case series method (Farrington, 1995) models the event rate during drug exposure in comparison to the baseline event rate while unexposed (see Whitaker et al., 2006; Madigan et al., 2010; Suchard et al., 2013a). In the self-controlled case series method, each individual also acts as their own control. Each treatment observation, which is a period of time that someone is drug-exposed, is considered with respect to other periods of time in which the same person is not exposed. This way of matching gracefully avoids patient-level selection bias; it controls for all fixed confounders, such as the individual's underlying frailty; the severity of their underlying disease, genetics, socioeconomic status, and so on. Further, because of the way the SCCS model is designed around this choice, the non-time dependent factors for each person cancel within the formula for the likelihood, and do not appear in the likelihood at all. This allows us to focus our modeling efforts on the time-dependent terms that involve the effects of the drugs.

To obtain SCCS's benefits, we also suffer its disadvantages and assumptions. First, SCCS is susceptible to bias due to potential unmeasured time-varying confounders. (However, SCCS does account for non-time-varying confounding.) This means we should include all features that affect the outcome and vary over time. Second, SCCS assumes that treatment effects are homogeneous across subjects. This avoids having to model patient-specific effects. However, it is possible that patients experience different effects from the various treatments. It is possible to create extensions of our approach that include patient specific random effects if desired. Third, the basic version of SCCS assumes that future outcomes are independent of past ones, but this can be changed, as discussed later. Conditional on

the model parameters, outcomes are assumed to be independent of each other, although because we are using latent factors, there can be marginal dependencies among the outcomes.

In the SCCS, events are modeled as arising from a non-homogeneous Poisson process. The event rate varies over time, based on exposure to drugs. Each patient $i = 1, \dots, N$ carries an unknown individual baseline event rate of e^{ϕ_i} . The exposure to drug $j = 1, \dots, J$ measured each day results in a multiplicative effect of e^{β_j} to this baseline rate e^{ϕ_i} . The historical data for patient i on day d ($d = 1, \dots, \tau_d$) includes a vector of drug exposure as $\mathbf{x}_{id} = [x_{id1}, x_{id2}, \dots, x_{idJ}]^T$, where $x_{idj} = 1$ if patient i is exposed to drug j on day d and 0 otherwise. The SCCS defines $\lambda_{id} = \exp(\phi_i + \mathbf{x}_{id}^T \boldsymbol{\beta})$ as the Poisson event rate for patient i on interval d , where $\boldsymbol{\beta} = [\beta_1, \beta_2, \dots, \beta_J]^T$ are regression coefficients. We denote y_{id} as the number of events that patient i experiences on day d . Conditioning on the total number of events for patient i , denoted by n_i , nuisance quantities ϕ_i cancel out of the SCCS likelihood, leaving log-likelihood as follows:

$$\mathcal{L}(\boldsymbol{\beta}) = \sum_i^N \left[\sum_d^{\tau_i} y_{id} \mathbf{x}_{id}^T \boldsymbol{\beta} - n_i \log \left(\sum_d^{\tau_i} e^{\mathbf{x}_{id}^T \boldsymbol{\beta}} \right) \right]. \quad (1)$$

Since larger LODs can contain millions of patients, avoiding estimation of the patient-specific baseline rates represents a significant computational and statistical advantage.

The most basic version of the SCCS deals with one drug and estimates a single unknown, β_1 , the effect estimate for the target drug of direct interest. However, most patients in longitudinal healthcare databases often take multiple drugs and treatments throughout the course of their observation and also experience multiple health outcomes. This motivates us to use a multiple-drug, multiple-outcome analysis.

3. Multi-drug, Multi-Outcome Self-Controlled Case Series - Notation and Inference

The methods proposed here generalize the self-controlled case series to handle multiple drugs/treatments and multiple outcomes/conditions. We describe the extended SCCS/MSCCS where there are J drugs and O health outcomes. The notation used throughout the paper is as follows:

- N : number of patients (i indexes individuals from 1 to N).
- x_{idj} : binary indicator reflecting whether patient i is exposed to drug j on interval d .
- $\mathbf{x}_{id} = [x_{id1}, x_{id2}, \dots, x_{idJ}]^T$: the vector of exposed drugs for patient i on interval d .
- J : number of drugs (treatments).
- O : number of health outcomes (adverse events).
- D_o^i : the set of observation intervals where patient i has outcome o .
- τ_o^i : the number of observation intervals where patient i has outcome o (the size of D_o^i).
- y_{id}^o : binary indicator reflecting whether patient i has outcome o on interval d .
- $\mathbf{y}_o^i = [y_{i1}^o, y_{i2}^o, \dots, y_{i\tau_o^i}^o]^T$: the vector of observed outcomes o for patient i .
- ϕ_o^i : baseline incidence of outcome o for patient i .

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_1^1 & \dots & \phi_1^O \\ \vdots & & \vdots \\ \phi_N^1 & \dots & \phi_N^O \end{pmatrix}: \text{baseline incidence matrix.}$$

β_j^o : regression coefficients associated with outcome o and drug j .
 $\beta^o = [\beta_1^o, \beta_2^o, \dots, \beta_J^o]^\top$: regression coefficients associated with outcome o .

$\mathbf{B} = \begin{pmatrix} \beta_1^O & \dots & \beta_1^O \\ \vdots & \ddots & \vdots \\ \beta_J^O & \dots & \beta_J^O \end{pmatrix}$: drug-outcome coefficient matrix.

$\lambda_{id}^0 = \exp(\phi_i^o + \mathbf{x}_{id}^\top \beta^o)$: the Poisson event rate of outcome o , for patient i , on interval d .

Similar to the SCCS, outcomes occur according to a nonhomogeneous Poisson process, where drug exposure can modulate the rate over time. Patient i has an individual baseline rate of $\exp(\phi_i^o)$ for outcome o that remains constant over time. Drug j has a multiplicative effect of $\exp(\beta_j^o)$ on the individual baseline rate $\exp(\phi_i^o)$ during its exposure period. The Poisson event rate for outcome o and patient i on interval d according to the SCCS is

$$\lambda_{id}^0 = \exp(\phi_i^o + \mathbf{x}_{id}^\top \beta^o).$$

The key benefit of the SCCS is that the ϕ_i^o terms do not need to be modeled, since we are interested in the ratio of Poisson intensities with and without the drug. For instance, considering only one drug j , comparing the intensity ratio for day d_1 to a different day d_2 with no exposure to the drug, we have

$$\frac{\lambda_{id_1}^0}{\lambda_{id_2}^0} = \frac{\exp(\phi_i^o + 1\beta_j^o)}{\exp(\phi_i^o + 0\beta_j^o)} = \exp(\beta_j^o).$$

As the Poisson rate is assumed to be constant within each interval, the number of outcomes o observed for patient i on interval d is distributed as a Poisson random variable (r.v.) denoted by Y_{id}^o as

$$\Pr(Y_{id}^o = y_{id}^o | \mathbf{x}_{id}) = \frac{e^{-\lambda_{id}^0} \lambda_{id}^0 y_{id}^o}{y_{id}^o!}.$$

Based on the above, the contribution to the likelihood for patient i and outcome o for the observed sequence of events $\mathbf{y}_i^o = [y_{i1}^o, y_{i2}^o, \dots, y_{iT^o}^o]^\top$, conditioned on the observed exposures $\mathbf{x}_i = [\mathbf{x}_{i1}, \dots, \mathbf{x}_{iT^o}]$ is

$$\begin{aligned} \mathcal{L}_i^o &= \Pr(\mathbf{y}_i^o | \mathbf{x}_i) = \prod_{d \in D_i^o} \Pr(y_{id}^o | \mathbf{x}_{id}) = \exp \left(- \sum_{d \in D_i^o} e^{\phi_i^o + \mathbf{x}_{id}^\top \beta^o} \right) \prod_{d \in D_i^o} \frac{(e^{\phi_i^o + \mathbf{x}_{id}^\top \beta^o})^{y_{id}^o}}{y_{id}^o!} \quad (2) \\ &= \exp \left(-e^{\phi_i^o} \sum_{d \in D_i^o} e^{\mathbf{x}_{id}^\top \beta^o} \right) \prod_{d \in D_i^o} e^{\phi_i^o y_{id}^o} \prod_{d \in D_i^o} \frac{(e^{\mathbf{x}_{id}^\top \beta^o})^{y_{id}^o}}{y_{id}^o!} \\ &= \exp \left(\phi_i^o n_i^o - e^{\phi_i^o} \sum_{d \in D_i^o} e^{\mathbf{x}_{id}^\top \beta^o} \right) \prod_{d \in D_i^o} \frac{(e^{\mathbf{x}_{id}^\top \beta^o})^{y_{id}^o}}{y_{id}^o!}, \end{aligned}$$

where $n_i^o = \sum_{d \in D_i^o} y_{id}^o$.

Two key assumptions underly the above likelihood. First, the model assumes that future outcomes are independent of past outcomes. For certain outcomes (e.g., myocardial infarction) this may not be reasonable. Simpson (2013), Schuemie et al. (2014), and Farrington et al. (2011) consider SCCS generalizations that allow for such dependence; in future work it is possible to consider similar generalizations of the method proposed here. The SCCS model also assumes that conditional on the parameters, outcomes are independent of each other. The latent structure, however, allows for arbitrary marginal dependence among outcomes.

One could form the full likelihood to estimate the unknown parameters (Φ, \mathbf{B}) . In order to avoid estimating the nuisance parameter set Φ , we can condition on its sufficient statistic, which removes the dependence on Φ . The cumulative intensity is a sum (rather than an integral) since we assume a constant intensity over each interval. Conditioning on n_i^o yields the following likelihood for person i :

$$\begin{aligned} \mathcal{L}_i^o &= \Pr(\mathbf{y}_i^o | \mathbf{x}_i, n_i^o) = \frac{\prod_{d \in D_i^o} \Pr(y_{id}^o | \mathbf{x}_{id})}{\Pr(n_i^o | \mathbf{x}_i)} = \frac{\prod_{d \in D_i^o} \Pr(y_{id}^o | \mathbf{x}_{id})}{\left[\frac{\exp \left(- \sum_{d \in D_i^o} \lambda_{id}^o \right) \left(\sum_{d \in D_i^o} \lambda_{id}^o \right)^{n_i^o}}{n_i^o!} \right]} \quad (3) \\ &\propto \exp \left(\prod_{d \in D_i^o} \left(\frac{e^{\mathbf{x}_{id}^\top \beta^o}}{\sum_{d'} e^{\mathbf{x}_{id'}^\top \beta^o}} \right)^{y_{id}^o} \right). \end{aligned}$$

Notice that because n_i^o is sufficient, the individual likelihood in the above expression no longer contains Φ . Assuming that patients are independent and outcomes are conditionally independent, the full conditional likelihood for event o is simply the product of the individual likelihoods (i.e. $\mathcal{L}^o = \prod_{i=1}^N \mathcal{L}_i^o$). Intuitively it follows that if i has no outcomes of type o , it cannot provide any information about the relative rate of outcome o .

Using the notation and the formula for the likelihood established in this section, we next present three hierarchical models called Factorized Self-Controlled Case Series methods, for multiple drug, multiple outcome analysis and discuss how to estimate the drug-outcome coefficient matrix \mathbf{B} . Two of the models have latent factors that allow \mathbf{B} to be expressed in a simpler and more interpretable way. In our experiments, the empirical performance of these methods is approximately the same.

4. Factorized Self-Controlled Case Series (FSCCS)

Building on the notation in the previous section, this section describes the proposed self-controlled case series methods within the three following subsections.

4.1 Model 0 - Hierarchical Model With No Latent Factors

Instead of estimating each coefficient independently, we borrow strength over both drugs and outcomes, which adds substantial regularization. This is particularly relevant when

considering a set of related outcomes and drugs, e.g., heart-disease related outcomes and the set of drugs one might prescribe for heart-related conditions. We take a hierarchical Bayesian approach. By analogy with ridge regression, we use normal priors for the regression parameters (sparsifying priors such as the double exponential could be used instead). We shrink the coefficients for drug j for all outcomes o to μ_j by placing an independent normal prior on each β_j^o as $\beta_j^o \sim \mathcal{N}(\mu_j, \sigma_j^2)$, $\forall (j, o)$, where $\mu_j \sim \mathcal{N}(0, \gamma^2)$, $\forall j$. This prior helps with numerical instability, overfitting, and makes the model more interpretable. We assume uniform priors for hyperparameters σ_j and γ as $\sigma_j \sim \mathbf{U}(0, a)$, $\forall j$ and $\gamma \sim \mathbf{U}(0, a)$, where hyperparameter a is a user-defined constant, which can also be determined through cross-validation. A natural extension of this model (not explored here) would be to have drugs belong to certain classes of drugs, so that priors can be defined based on each class of drugs; similarly with outcomes. The posterior density is as follows:

$$\begin{aligned} \Pr(\mathbf{B}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \gamma | \mathbf{y}, a) &\propto \Pr(\mathbf{y} | \mathbf{B}) \times \Pr(\boldsymbol{\mu} | \boldsymbol{\sigma}) \times \Pr(\gamma | a) \times \Pr(\boldsymbol{\sigma} | a) \\ &\propto \prod_o \prod_i \prod_{d \in D_i} \prod \left(\frac{\exp(\mathbf{x}_{id}^\top \boldsymbol{\beta}^o)}{\sum_{d' \in D_i} \exp(\mathbf{x}_{id'}^\top \boldsymbol{\beta}^o)} \right)^{y_{id}^{(o)}} \\ &\times \prod_j \prod_o \mathcal{N}(\beta_j^o | \mu_j, \sigma_j^2) \times \prod_j \mathcal{N}(\mu_j | 0, \gamma^2) \times \prod_j \Pr(\sigma_j | a) \times \Pr(\gamma | a). \end{aligned} \quad (4)$$

The negative log-posterior (which can be used for finding the MAP solution if desired) is:

$$\mathcal{L}_1 = -\log(\Pr(\mathbf{B}, \boldsymbol{\mu}, \boldsymbol{\sigma}, \gamma | \mathbf{y}, a)).$$

The graphical representation of this model is shown in Figure 1.

4.2 Model 1 - One Level of Latent Factors

Two considerations motivate this model. First, modeling the full posterior distribution of Model 0 can be computationally expensive, particularly for large N , J , and O , where J and O determine the number of variables to be estimated within the \mathbf{B} matrix. Second, Model 0 overlooks the fact that drugs and outcomes might come from a smaller number of latent classes; for instance, there are commonly several drugs that are extremely similar to each other for treating a set of highly related illnesses. We consider F latent factors for drugs and outcomes. We model the $J \times O$ matrix \mathbf{B} as $\mathbf{B} = \mathbf{L}^{(D)} \times \mathbf{L}^{(O)}$, where

$$\mathbf{L}^{(D)} = \begin{pmatrix} L_{1,1}^{(D)} & \dots & L_{1,F}^{(D)} \\ \vdots & & \vdots \\ L_{J,1}^{(D)} & \dots & L_{J,F}^{(D)} \end{pmatrix}, \mathbf{L}^{(O)} = \begin{pmatrix} L_{1,1}^{(O)} & \dots & L_{1,O}^{(O)} \\ \vdots & & \vdots \\ L_{F,1}^{(O)} & \dots & L_{F,O}^{(O)} \end{pmatrix}.$$

This way, we do not assume we know in advance which drugs have similar effects on which outcomes, instead we estimate this from data. The number of latent factors F can be determined by cross-validation. The total number of latent factors is $J \times F + F \times O$, which can be substantially less than $J \times O$. The coefficient β_j^o associated with outcome o and drug j can be calculated as $\beta_j^o = \sum_{f=1}^F L_{j,f}^{(D)} \times L_{f,o}^{(O)}$.

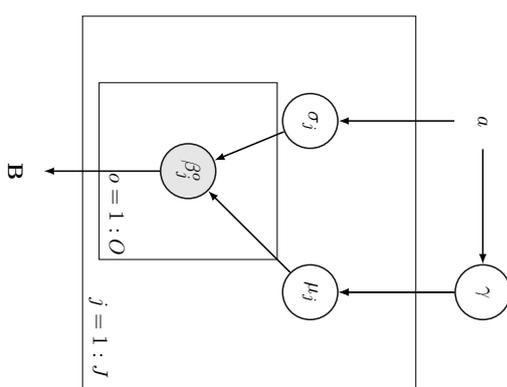


Figure 1: Graphical representation of Model 0

For drug latent factors, we place independent normal priors on the entries of $L^{(D)}$ as

$$L_{j,f}^{(D)} \sim \mathcal{N}(\mu_f^{(D)}, \sigma_f^{(D)2}), \forall (j, f), \text{ where } \mu_f^{(D)} \sim \mathcal{N}(0, \gamma^{(D)2}), \forall f.$$

Similarly, we define normal priors on the entries of $L^{(O)}$ as

$$L_{f,o}^{(O)} \sim \mathcal{N}(\mu_f^{(O)}, \sigma_f^{(O)2}), \forall (f, o), \text{ where } \mu_f^{(O)} \sim \mathcal{N}(0, \gamma^{(O)2}), \forall f.$$

We assume uniform priors for hyperparameters σ_j and γ as

$$\sigma_f^{(D)} \sim \mathbf{U}(0, a), \forall f, \sigma_f^{(O)} \sim \mathbf{U}(0, a), \forall f, \gamma^{(D)} \sim \mathbf{U}(0, b), \gamma^{(O)} \sim \mathbf{U}(0, b),$$

where (a, b) are known parameters. The posterior over the parameters is now defined as

$$\begin{aligned}
& \Pr(\mathbf{L}^{(D)}, \mathbf{L}^{(O)}, \boldsymbol{\mu}^{(D)}, \boldsymbol{\mu}^{(O)}, \boldsymbol{\sigma}^{(D)}, \boldsymbol{\sigma}^{(O)}, \gamma^{(D)}, \gamma^{(O)} | \mathbf{y}) \propto \Pr(\mathbf{y} | \mathbf{L}^{(D)}, \mathbf{L}^{(O)}) \\
& \times \Pr(\mathbf{L}^{(D)} | \boldsymbol{\mu}^{(D)}, \boldsymbol{\sigma}^{(D)}) \times \Pr(\mathbf{L}^{(O)} | \boldsymbol{\mu}^{(O)}, \boldsymbol{\sigma}^{(O)}) \times \Pr(\boldsymbol{\mu}^{(D)} | \gamma^{(D)}) \times \Pr(\boldsymbol{\mu}^{(O)} | \gamma^{(O)}) \\
& \times \Pr(\boldsymbol{\sigma}^{(D)} | a) \times \Pr(\boldsymbol{\sigma}^{(O)} | a) \times \Pr(\gamma^{(D)} | b) \times \Pr(\gamma^{(O)} | b) \\
& \propto \prod_{\sigma} \prod_i \prod_{d \in \mathcal{D}_i} \prod_{\sigma'} \left(\frac{\exp(\mathbf{x}_{id}^T \boldsymbol{\beta}^{\sigma'})}{\sum_{\sigma''} \exp(\mathbf{x}_{id}^T \boldsymbol{\beta}^{\sigma''})} \right)^{y_{id}^{\sigma'}} \\
& \times \prod_{j=1}^J \prod_{f=1}^F \mathcal{N}(L_{j,f}^{(D)} | \mu_{j,f}^{(D)}, \sigma_{j,f}^{(D)2}) \times \prod_{f=1}^F \prod_{o=1}^O \mathcal{N}(L_{f,o}^{(O)} | \mu_{f,o}^{(O)}, \sigma_{f,o}^{(O)2}) \\
& \times \prod_{f=1}^F \mathcal{N}(\mu_f^{(D)} | 0, \gamma^{(D)2}) \times \prod_{f=1}^F \mathcal{N}(\mu_f^{(O)} | 0, \gamma^{(O)2}) \\
& \times \Pr(\boldsymbol{\sigma}_f^{(D)} | b) \times \Pr(\boldsymbol{\sigma}_f^{(O)} | b) \times \Pr(\gamma^{(D)} | a) \times \Pr(\gamma^{(O)} | a).
\end{aligned} \tag{5}$$

The graphical representation of this hierarchical Bayesian model is given in Figure 2.

4.3 Model 2 - Two Levels of Latent Factors

Here we represent \mathbf{B} as

$$\mathbf{B} = \mathbf{L}^{(D)} \times \mathbf{L}^{(F)} \times \mathbf{L}^{(O)},$$

where

$$\mathbf{B} = \begin{pmatrix} L_{1,1}^{(D)} & \dots & L_{1,F_1}^{(D)} & \dots & L_{1,F_1}^{(D)} & \dots & L_{1,1}^{(F)} & \dots & L_{1,F_2}^{(F)} & \dots & L_{1,O}^{(O)} \\ \vdots & & \vdots \\ L_{F_1,1}^{(D)} & \dots & L_{F_1,F_1}^{(D)} & \dots & L_{F_1,F_1}^{(D)} & \dots & L_{F_1,1}^{(F)} & \dots & L_{F_1,F_2}^{(F)} & \dots & L_{F_2,O}^{(O)} \end{pmatrix}.$$

The number of latent factors is thus $J \times F_1 + F_1 \times F_2 + F_2 \times O$, which can be less than the number of variables of Model 1 in many cases. Its major benefit is interpretability, since now the number of latent drug factors and the number of latent outcome factors can be estimated differently. $\mathbf{L}^{(D)}$ represents the relationship between drugs and latent drug-related factors, $\mathbf{L}^{(F)}$ represents the relationship between latent drug-related factors and latent health-outcome-related factors, and $\mathbf{L}^{(O)}$ represents the relationship between latent health-outcome-related factors and health-outcome-related factors. $\mathbf{L}^{(F)}$ is really the core set of variables since they relate the latent treatments to the latent health outcomes.

The priors are $L_{j,f_1}^{(D)} \sim \mathcal{N}(\mu_{j,f_1}^{(D)}, \sigma_{f_1}^{(D)2})$, $L_{f_2,o}^{(O)} \sim \mathcal{N}(\mu_{f_2,o}^{(O)}, \sigma_{f_2,o}^{(O)2})$, $L_{f_1,f_2}^{(F)} \sim \mathcal{N}(\mu_{f_1,f_2}^{(F)}, \sigma_{f_1,f_2}^{(F)2})$, $\mu_{f_1}^{(D)} \sim \mathcal{N}(0, \gamma^{(D)2})$, $\mu_{f_2}^{(O)} \sim \mathcal{N}(0, \gamma^{(O)2})$, $\mu_{f_2}^{(F)} \sim \mathcal{N}(0, \gamma^{(F)2})$, $\mu_{f_1}^{(D)} \sim \mathcal{U}(0, a)$, $\sigma_{f_1}^{(D)} \sim \mathcal{U}(0, a)$, $\sigma_{f_2}^{(O)} \sim \mathcal{U}(0, a)$, $\gamma^{(D)} \sim \mathcal{U}(0, b)$, $\gamma^{(F)} \sim \mathcal{U}(0, b)$, and $\gamma^{(O)} \sim \mathcal{U}(0, b)$ for all (f_1, f_2, j, o) .

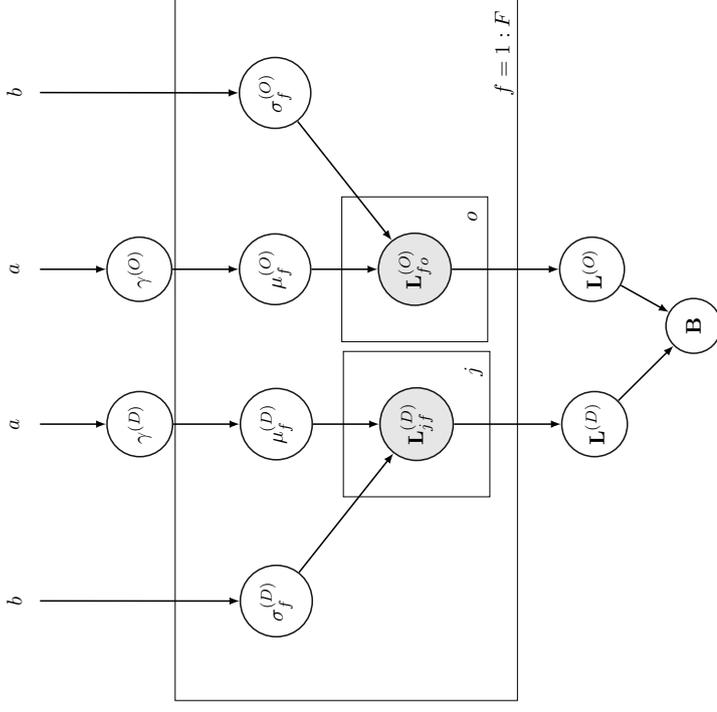


Figure 2: The Graphical Framework for Hierarchical Bayesian Model with one level of latent factors

The posterior density is

$$\begin{aligned}
& \Pr(\mathbf{L}^{(D)}, \mathbf{L}^{(F)}, \mathbf{L}^{(O)}, \boldsymbol{\mu}^{(D)}, \boldsymbol{\mu}^{(F)}, \boldsymbol{\mu}^{(O)}, \boldsymbol{\sigma}^{(D)}, \boldsymbol{\sigma}^{(F)}, \boldsymbol{\sigma}^{(O)}, \gamma^{(D)}, \gamma^{(F)}, \gamma^{(O)} | \mathbf{y}) \\
& \propto \Pr(\mathbf{y} | \mathbf{L}^{(D)}, \mathbf{L}^{(F)}, \mathbf{L}^{(O)}, \boldsymbol{\mu}^{(D)}, \boldsymbol{\mu}^{(F)}, \boldsymbol{\mu}^{(O)}, \boldsymbol{\sigma}^{(D)}, \boldsymbol{\sigma}^{(F)}, \boldsymbol{\sigma}^{(O)}) \times \Pr(\mathbf{L}^{(F)} | \boldsymbol{\mu}^{(F)}, \boldsymbol{\sigma}^{(F)}) \times \Pr(\mathbf{L}^{(O)} | \boldsymbol{\mu}^{(O)}, \boldsymbol{\sigma}^{(O)}) \\
& \times \Pr(\boldsymbol{\mu}^{(D)} | \gamma^{(D)}) \times \Pr(\boldsymbol{\mu}^{(F)} | \gamma^{(F)}) \times \Pr(\boldsymbol{\mu}^{(O)} | \gamma^{(O)}) \\
& \times \Pr(\boldsymbol{\sigma}^{(D)} | a) \times \Pr(\boldsymbol{\sigma}^{(F)} | a) \times \Pr(\boldsymbol{\sigma}^{(O)} | a) \times \Pr(\gamma^{(D)} | b) \times \Pr(\gamma^{(F)} | b) \times \Pr(\gamma^{(O)} | b).
\end{aligned} \tag{6}$$

Table 1 compares the number of parameters in each of the three models. Models 1 and 2 have much fewer parameters when F , F_1 , and F_2 are lower than J and O . We use Markov Chain Monte Carlo (MCMC) to approximate the entries of \mathbf{B} , specifically random walk Metropolis (RWM) Hasting. The algorithm employs a Gaussian proposal distribution

Model Name	# of Parameters	# of Hyperparameters	Total
Model 0	$J * O$	$2 * J + 1$	$J * O + 2 * J + 1$
Model 1	$J * F + F * O$	$4 * F + 2$	$J * F + F * O + 4 * F + 2$
Model 2	$J * F_1 + F_1 * F_2 + F_2 * O$	$2 * F_1 + 2 * F_2 + 5$	$J * F_1 + F_1 * F_2 + F_2 * O + 2 * F_1 + 2 * F_2 + 5$

Table 1: The number of parameters and hyperparameters in each model.

$J_i(x, x')$ which proposes a new parameter set x' given the current parameter set x . We denote Θ as the set of all parameters in the model excluding \mathbf{B} .

Step 1. Generate an initial state $\{\mathbf{B}^0, \Theta^0\}$ with positive probability $\Pr(\mathbf{B}^0, \Theta^0 | \mathbf{y})$ and set $t = 1$.

Repeat the following until stationary distribution and the desired number of samples are reached considering optional burn-in and/or thinning.

Step 2. Sample $\{\mathbf{B}^*, \Theta^*\}$ from the symmetric proposal distribution $J_i(\{\mathbf{B}^{t-1}, \Theta^{t-1}\}, \{\mathbf{B}^*, \Theta^*\})$.

Step 3. Calculate the acceptance probability

$$\alpha = \min \left(1, \frac{\Pr(\mathbf{B}^*, \Theta^* | \mathbf{y})}{\Pr(\mathbf{B}^{t-1}, \Theta^{t-1} | \mathbf{y})} \right).$$

Step 4. Draw a random number u from $\text{Unif}(0, 1)$. If $u \leq \alpha$, accept the proposal state $\{\mathbf{B}^*, \Theta^*\}$ and set $\mathbf{B}^t = \mathbf{B}^*$, $\Theta^t = \Theta^*$, else set $\mathbf{B}^t = \mathbf{B}^{t-1}$, $\Theta^t = \Theta^{t-1}$. Set $t: t + 1$.

Our implementation uses a component-wise sampling approach. For truly large-scale applications, blocked sampling approaches may be necessary.

5. Simulation Study

As a sanity check, we will show that for data generated from our model, the true data-generating parameters \mathbf{B} can be recovered. We simulated sample trajectories of drug exposure and health outcomes for 600 patients over 60 days. We set the number of drugs to $J = 4$, and the number of health conditions to $O = 4$. Each patient randomly took between 1 and J drugs over the past 60 days. The average exposure period was assumed to be 20% of the study interval for each patient (that is, on average, each patient was exposed to one or more drugs for at least 12 days). The exposure intervals are randomly selected, so these intervals could be multiple non-consecutive days, multiple consecutive days, or a combination of both. Drugs can have positive or negative contributions to the likelihood and intensity rate of each outcome. For each model (Model 0, Model 1, Model 2), we generated the elements of \mathbf{B} according to the model's hierarchy. Figure 3 and Figures 12-13 (which are given in the Appendix) show the posterior density for each parameter of \mathbf{B} , for Models 0, 1 and 2, as estimated by MCMC sampling. These figures show that the posterior samples were concentrated around the true values and the posterior mean of each variable was generally close to its true value. We summarize Figures 3, 12, and 13 in Figure 4, which provides a scatter plot of the posterior means and true values for each of the three models. It can be observed that each of the posterior means are very close to their true values.

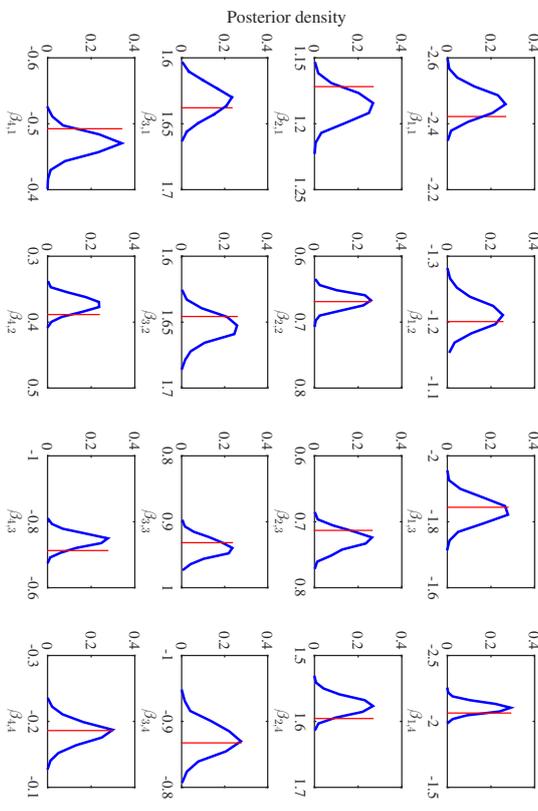


Figure 3: Normalized histograms of posterior samples for each element of \mathbf{B} in Model 0. The vertical line indicates the true value.

6. Application to Blood Glucose Analysis for Diabetes

We consider an application to Insulin-Dependent Diabetes Mellitus, where our goal is to predict blood glucose level outcomes under different circumstances of a patient's daily life, including their recent eating history, exercise, and insulin injections. Our data are longitudinal measurements taken multiple times per day from 70 patients (this is the AIM-94 data set provided by Michael Kahn, MD, PhD, Washington University, St. Louis, MO, Bache and Lichman, 2013). We aim mainly to illustrate (i) how the models we introduce can be used with complex longitudinal data to predict outcomes, (ii) the prediction power, and (iii) interpretability of the proposed models. It is well known that current therapies for regulating glucose level in diabetics are challenging and often frustrating, as they require patients to continuously regulate diet, exercise, and various medications – any deviations can be dangerous (Banchinnol et al., 2013). Blood glucose measurements, symptoms and insulin treatments were recorded with timestamps for each patient, over the course of several weeks to months. The two main classes of health outcomes considered here are hyperglycemia (high blood glucose) and hypoglycemia (low blood glucose). All other health outcomes we define later are related to these two classes. Figure 5 provides a schematic of the type of data we are considering for one patient over a course of day.

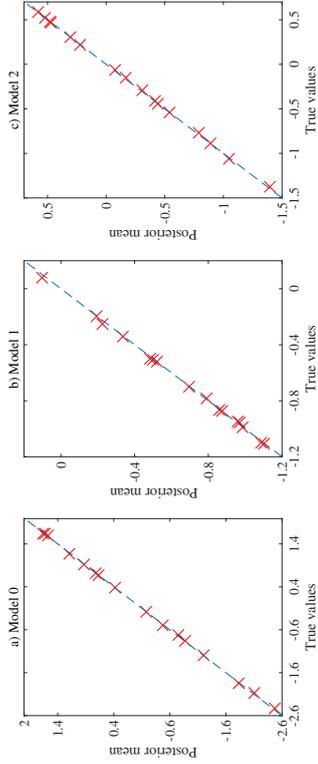


Figure 4: Scatter plot of posterior means vs. true parameter values of elements of **B** for Models 0, 1 and 2.

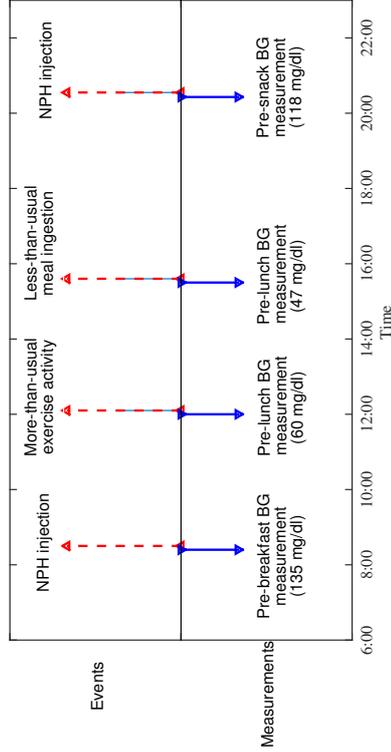


Figure 5: Sample longitudinal traces for a patient with multiple drug/treatments exposures and blood glucose measurements over a day. Downwards arrows indicate glucose measurements. Upwards arrows indicate treatments.

We will describe the setup in more detail:
Drugs/Treatments: Diet, exercise, and injected insulin were treated as three different classes of treatments. It is obvious that interactions among these treatments are important. Insulin doses are given one or more times a day, typically before meals and sometimes also at bedtime. Three types of insulin were considered: (1) regular, (2) Neutral Protamine Hagedorn (NPH), and (3) Ultralente. Each insulin type has its own characteristic time of

onset (O), time of peak action (P), and effective duration (D). The exposure time intervals for peak, and duration used in this paper, which were provided with the data set, are shown in Table 2 in the “Exposure Time” column in the bottom several rows of the table labeled “Insulin.”

	Treatment	Exposure Time	Health Outcome						
			Low Glucose Level			High Glucose Level			
			1. Too low	2. Low	3. D1	4. Hypo-Symptom	5. Too High	6. High	7. D10
Exercise	1. Normal	0-4 h	-	P	-	-	-	-	-
	2. Too High	0-4 h	-	P	-	-	-	-	-
	3. Low	0-4 h	-	P	-	-	-	-	-
Diet	4. Normal	0-4 h	-	-	-	-	-	-	-
	5. Too High	0-4 h	N	N	N	N	N	N	N
	6. Low	0-4 h	-	-	-	-	-	-	-
	7. After Meal	0-4 h	N	N	N	N	N	N	N
Insulin	8. Before Meal	0-4 h	P	P	P	P	P	P	P
	9. Peak	1-3 h	P	P	P	P	P	P	P
	10. Duration	0-6 h	P	P	P	P	P	P	P
	11. Peak	4-6 h	P	P	P	P	P	P	P
Ultralente	12. Duration	0-12 h	P	P	P	P	P	P	P
	13. Peak	14-24 h	P	P	P	P	P	P	P
	14. Duration	0-27 h	P	P	P	P	P	P	P

Table 2: The list of drugs/treatments and health outcomes and their known correlations. P means strong positive correlation, and N means strong negative correlation, D1 is lower decile and D10 means highest decile.

Based on the actual time of injection, we determined the intervals at which the patient is at peak and/or within the duration of an insulin injection. Based on this, six types of treatments were considered, (1) regular insulin on peak, (2) regular insulin on duration, (3) NPH insulin on peak, (4) NPH insulin on duration, (5) Ultralente insulin on peak, and (6) Ultralente insulin on duration. At each interval of time, the patient can be either insulin free or subject to one of the above six exposures.

The second class of treatment is exercise, which may have complex effects on the glucose level. For example, glucose levels can fall during exercise but also quite a few hours afterwards. Three types of exercise are reported, (1) normal exercise, (2) lower than normal exercise, and (3) higher than normal exercise. Each type of exercise was considered separately as a single treatment.

The third class of treatment is for diet, which also can have complex effects on the glucose level. For example, a larger meal may lead to a longer and possibly higher elevation of blood glucose. Missing a meal may put the patient at risk for low glucose levels in the hours that follow. Three types of diet are reported: (1) normal diet, (2) higher than normal diet, and (3) lower than normal diet. Each of these types of diet were taken as a single treatment. Since measurements were collected before a meal, after a meal, and at other times, we considered two other features in the model, (1) before meal measurement of blood glucose and (2) after meal measurement, and we treated them as binary features. These extra features allow us to distinguish whether the measurement was made before or after the meal (there is a big difference between glucose measurements taken before a meal and after a meal).

Based on all of the treatments described, the total number of variables associated with treatments in the model is 14 ($J = 14$). The variables are all listed on Table 2 in the “Treatment” column on the left.

Health Outcomes. The outcomes are divided into categories, based on glucose level. Given that normal pre-meal blood glucose ranges from approximately 80-120 mg, and post-meal blood glucose ranges from 80-140 mg/dl (Bache and Lichman, 2013), we considered seven health outcomes for glucose level: extremely low (below 40 mg/dl), low (between 40-80 mg/dl), high (over 140 mg/dl), extremely high (over 180 mg/dl), lower decile (lower 10% of glucose level for each patient), upper decile (upper 10% of glucose level for each patient), and hypoglycemic (low glucose) symptoms. Thus, the total number of outcomes considered for our analysis is $O = 7$. We can perform an evaluation only on intervals where we have glucose measurements, thus we only use those intervals. Note that more than one outcome can occur in each interval.

True Relationships Between Drugs and Outcomes. We wanted to determine whether our model reproduces known relationships between treatments and glucose levels from the data alone. The information about true relationships within Table 2 mainly come from material accompanying the data set and *www.diabetes.org*. We denoted known positive effects in Table 2 by \mathcal{P} , strong negative effects by \mathcal{N} , and relationships that were unknown were denoted by dashes “-”. For example, we expect NPH injection on peak to decrease the likelihood of having “Too High” glucose level, so the correlation between NPH on peak and “Too High” glucose level is known to be negative (\mathcal{N}).

Mixing. We performed cross-validation, dividing our data into five folds, training our models on four folds and testing on the fifth. We removed the first 5000 iterations (as burn-in) of Metropolis-Hastings sampling, and obtained 6000 additional samples to estimate the posterior. Figure 6 shows samples from the posterior of one of the variables, $\beta_{\text{NPH on peak}}^{\text{Too high GL}}$, for five separate model instantiations (Model 0, Model 1 with $F=2$, Model 1 with $F=3$, Model 2 with $F_1=2, F_2=2$, and Model 2 with $F_1=3, F_2=3$). Recall that in Models 1 and 2, we sample elements of the matrices of latent factors and then calculate \mathbf{B} . From this figure, we observe reasonable mixing for all models, and we observe that models with latent factors (Models 1 and 2) have better mixing and convergence, possibly due to the smaller number of variables.

Computation. The number of parameters differs substantially between models, which affects CPU time of the MCMC sampler. In Figure 7, the number of parameters for each model and the associated CPU time for running MCMC are shown. This figure shows a clear correlation between the number of variables and CPU time, that is CPU time increases with the number of parameters. In particular, Model 0 takes a long time to run, because it has substantially more variables than the other models. Interestingly, using latent factors has a purpose beyond interpretability and regularization, in that it helps with tractability.

Interpretation of coefficients in B and comparison with ground truth. We compare the estimated coefficients in \mathbf{B} to the ground truth signs of coefficients given in Table 2. It is not necessarily the case that the signs of the estimated coefficients need to agree with the ground truth signs in order for the model to perform well, but it is a reasonable aspect of the model to consider. To perform this comparison, we ranked the estimated coefficients in \mathbf{B} and used these rankings and the true signs of coefficients to generate an ROC curve. That is, the ROC curve was generated by placing thresholds at each point in

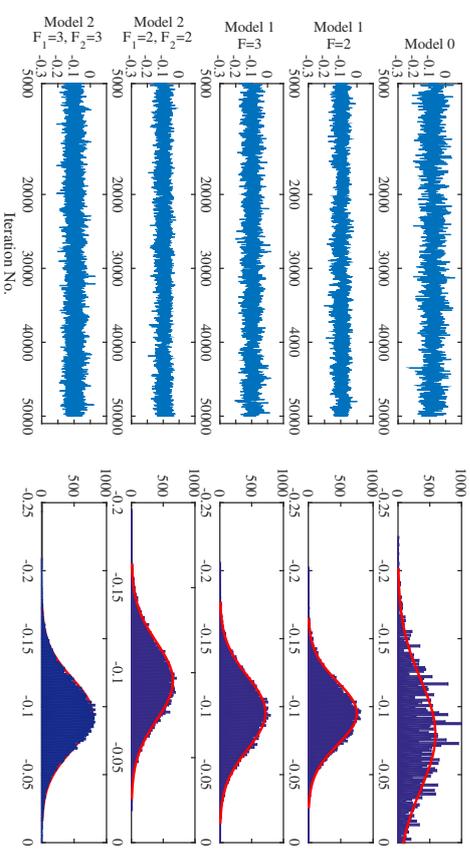


Figure 6: Samples from the posterior of $\beta_{\text{NPH on peak}}^{\text{Too high GL}}$ for five model instantiations (left), and their histograms (right). The first 5000 samples were discarded as burn-in. We observe better mixing and convergence in the models with latent factors (Model 1 and Model 2).

the ranking, and calculating the true positive rate and false positive rate with respect to the true coefficients. We computed these ROC curves for all five model instantiations to obtain Figure 8. These ROC curves indicate that all models performed well, in the sense of estimating reasonable signs for the coefficients. The curves also indicate that models with more latent factors performed slightly better than Model 0. The models with two latent treatment and outcome factors performed slightly better than the models with three latent treatment and outcome factors, though there was no significant difference in performance between Model 1 and Model 2 for the same number of latent treatment and outcome factors.

Drug Surveillance. We evaluated prediction performance of our model as follows: for each patient we calculated the Poisson rate of each condition at each hour considering all drug exposures. For each condition, we then checked whether or not the patient had the condition at that time. The Poisson rate acts as the score of each patient with regards to each condition. In Figure 9, we present the actual glucose level for a patient at 20 measurement points (upper figure) as well as the estimated intensity rate of the “Too-High glucose level for the same patient (lower figure). Each point in this figure represents the estimated $\mathbf{x}_{\text{int}}^T \hat{\beta}^e$, where $\hat{\beta}^e$ are the estimated regression coefficients associated with too-high glucose level and $\mathbf{x}_{\text{int}}^T$ is the known vector of drug exposures at interval d . The figure shows that the estimated intensity rate is reasonably close to the actual level of glucose, particularly when the glucose level is actually too high. In Figure 10, we repeated the

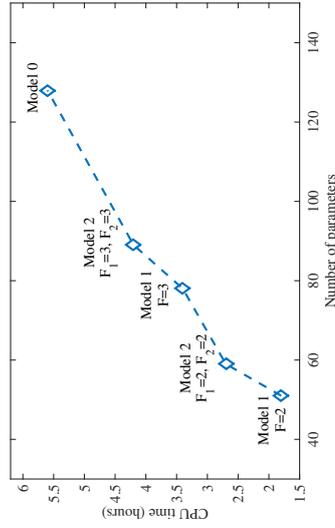


Figure 7: Comparison between the CPU time and number of variables in the five model instantiations. Model 0 has larger number of variables and significantly higher CPU time.

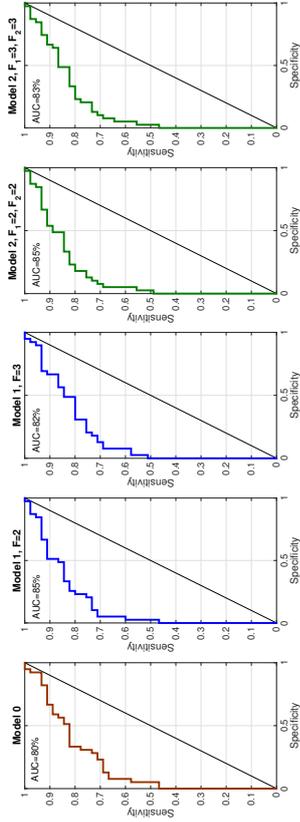


Figure 8: Receiver Operating Characteristic (curve) for evaluating the signs of coefficients against true signs for five model instantiations. See the text for details of how these curves were generated.

analysis for Too-Low glucose level. It can be observed from this figure that the times where this coefficient is particularly large are the same times where the glucose level drops substantially. This kind of dramatic agreement was not observed for all patients nor all conditions, so below we describe a more general evaluation procedure.

In Figure 11, we show the box plots for the estimated Poisson rates of the two conditions of “Too Low” glucose level and “Too High” glucose level on all seventy patients in the test sets. For comparison, we also plotted the box plots of the estimated Poisson rates for normal conditions, where patients did not have too-high or too-low glucose levels. For

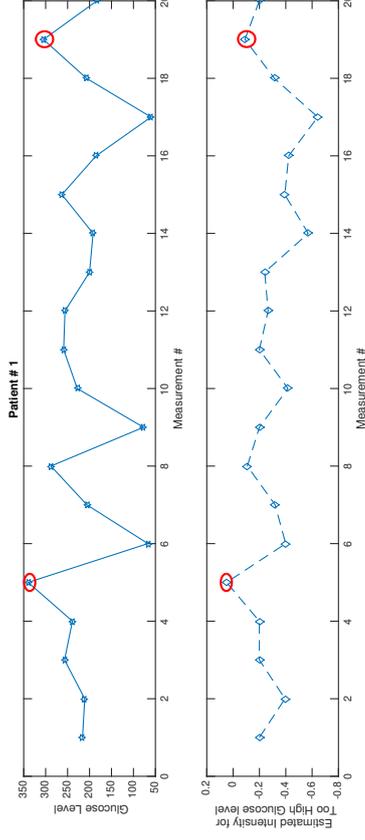


Figure 9: Monitoring Too-High glucose levels over 20 hours. The upper figure shows the true glucose levels obtained by measurement, and the lower figure shows the estimated $\alpha_{i,d}^T \beta^o$ for the Too-High glucose outcome over 20 consecutive measurements.

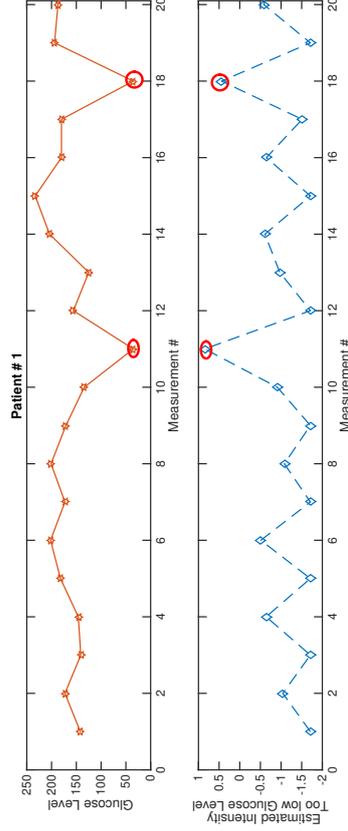


Figure 10: Monitoring too-low glucose levels over 20 hours. The upper figure shows the true glucose levels obtained by measurement, and the lower figure shows the estimated $\alpha_{i,d}^T \beta^o$ for the too-low glucose outcome over 20 hours.

clarity and fairness, we normalized the estimated Poisson rates for each patient. It can be observed from this figure that the estimated Poisson rate of these conditions are elevated when patients actually suffer from Too-Low (or Too-High, respectively) glucose. Thus, our model could be a useful approach for monitoring the likelihood of a condition, given the

timing of the drugs recently taken by the patient. The results were consistent across all five model instantiations.

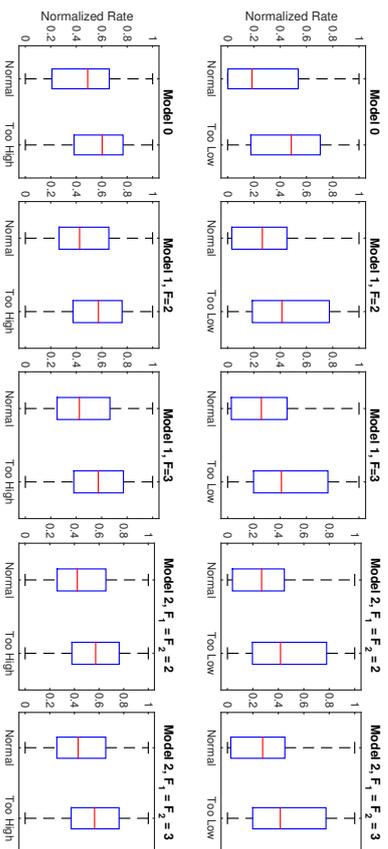


Figure 11: Comparison between the box-plots for Too-High and Too-Low glucose levels and the normal condition for all five model instantiations, where the Poisson rates were normalized for each person.

In Table 3, we report an AUC (area under the ROC curve) value that measures the probability that a method ranks a positive condition timepoint higher than a timepoint with no condition, for the same patient. In particular, we are testing whether the Poisson rate of a patient with a health condition is higher than the estimated rate of the same patient when no condition is present. For each model instantiation, we trained the model on four training folds and then calculated the AUC on the fifth fold, and we repeated this for each condition. We reported the average and standard deviation of AUC over all patients in the test sets. Again DI is the lower decile (lower 10% of glucose level for each patient), and D10 is the upper decile (upper 10% of glucose level for each patient).

Intuition of \mathbf{B} as compared with other methods. We compared the performance of the proposed models with that of the univariate self-controlled case series (SCCS), and multi-variate self-controlled case series (MSCCS) (Simpson et al., 2013) with and without an ℓ_2 regularization term on the coefficients. For each model instantiation and each method, the estimated entries of \mathbf{B} were compared to their known effects (positive or negative) from Table 2. For each method, we provide the area under the curve (AUC) for this comparison in Table 4. We also reported the mean, median, and standard deviation of all estimated coefficients in the group of Table 2’s positive group and Table 2’s negative group. Better models should have higher AUC, and the estimated coefficients of \mathbf{B} should agree in sign with those in Table 2. From Table 4, we observe that as expected, the MSCCS performed better than the SCCS, the regularized MSCCS worked slightly better than the normal

	Too low	Low	Too high	High	DI	D10	Hypo Symptom
Model 0	Mean	70.30%	62.92%	62.98%	61.26%	58.20%	56.39%
	sd	9.36%	2.79%	3.38%	2.70%	3.93%	3.51%
Model 1, $F = 2$	Mean	70.86%	62.70%	63.06%	61.58%	57.97%	56.05%
	sd	8.28%	2.47%	3.14%	2.11%	3.36%	3.04%
Model 1, $F = 3$	Mean	70.66%	62.68%	62.92%	61.51%	57.97%	56.35%
	sd	7.89%	2.52%	3.14%	2.12%	3.37%	2.90%
Model 2, $F_1 = F_2 = 2$	Mean	70.19%	62.71%	62.97%	61.44%	58.00%	55.91%
	sd	9.51%	2.43%	3.18%	2.13%	3.34%	3.06%
Model 2, $F_1 = 3, F_2 = 3$	Mean	70.07%	62.73%	62.99%	61.47%	58.05%	56.05%
	sd	9.52%	2.44%	3.16%	2.07%	3.31%	3.05%

Table 3: Average and standard deviation (sd) of AUC over 5 folds. The entries that were ranked for each AUC calculation are measurements for a patient including time points when the patient had a condition and time points when the patient did not have a condition. The AUC indicates whether our method ranks a randomly chosen time point where a patient had a condition higher than a randomly chosen time point where a patient did not have a condition.

MSCCS, and all FSCCS Bayesian model instantiations (Model 0-2) performed better than all of the traditional models, yielding higher AUC’s and better agreement in the mean signs of coefficients; further the standard deviations for the coefficient values were substantially lower. These performance benefits come in addition to the other benefits discussed earlier, including computational tractability and interpretability of the latent factors.

Measure	Method						
	Model 0	Model 1	Model 2	Model 2	SCCS	MSCCS	Regularized MSCCS
AUC	0.813	0.852	0.824	0.856	0.836	0.693	0.773
Mean	-0.190	-0.281	-0.276	-0.278	-0.282	-0.492	-0.663
	-0.077	-0.136	-0.143	-0.130	-0.141	-0.047	-0.122
sd	0.353	0.417	0.410	0.422	0.424	2.362	2.462
	0.071	0.099	0.096	0.105	0.096	-0.552	-0.561
Median	0.030	0.103	0.111	0.106	0.106	0.065	0.126
	0.368	0.369	0.385	0.368	0.378	3.158	3.149

Table 4: Comparison with existing models.

7. Concluding Remarks

The novel elements of this work are as follows. (1) We estimate the effects of many drugs on many health outcomes simultaneously. Borrowing strength across similar drugs and outcomes allows us to create better estimates across both drugs and outcomes. (2) We use latent factors to encode latent classes of drugs and outcomes, to help with interpretability, and to provide a computational benefit. Another type of computational benefit is provided naturally by using the SCCS's framework, since we do not need to estimate the baseline rates of outcomes for each patient. This approach is scalable to large longitudinal observational databases, is applicable to problems beyond healthcare, and provides a level of interpretability to physicians and patients that was not previously possible. Fully Bayesian inference via MCMC may not be feasible for truly large-scale problems. Recent developments in cyclic coordinate descent algorithms (see, for example, in Suchard et al., 2013b) would apply in our context and represent one possible approach for very scale MAP estimation.

Acknowledgments

We gratefully acknowledge funding from the Natural Science and Engineering Research Council of Canada (NSERC) and the MIT Big Data Initiative.

Appendix A. Figures 12 and 13.

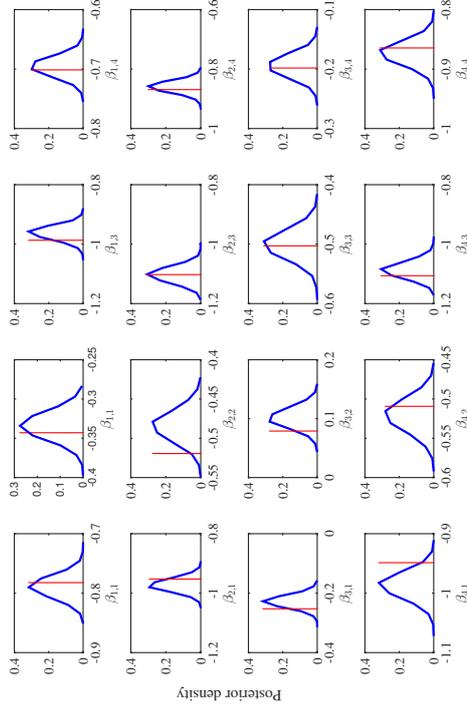


Figure 12: Normalized histograms of posterior samples for each element of \mathbf{B} in Model 1. The vertical line indicates the true value.

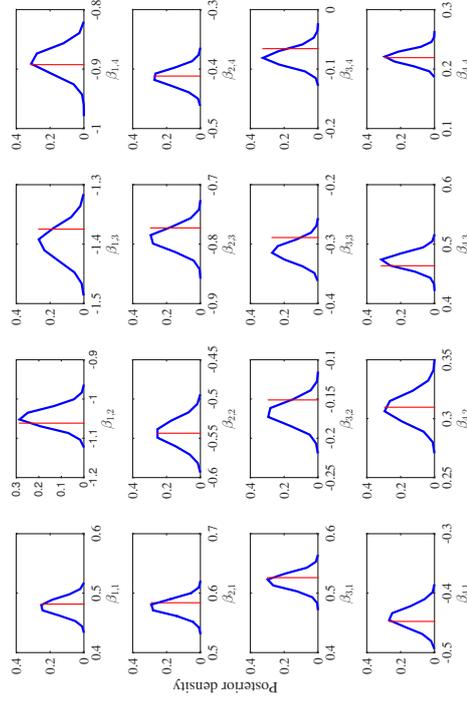


Figure 13: Normalized histograms of posterior samples for each element of \mathbf{B} in Model 2. The vertical line indicates the true value.

References

- K. Bachle and M. Lichman. UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences, 2013.
- E.I. Benchimol, S. Hawken, J.C. Kwong, and K. Wilson. Safety and utilization of influenza immunization in children with inflammatory bowel disease. *Pediatrics*, 131(6):1811–1820, 2013.
- C.M. Carvalho, J. Chang, J.E. Lucas, J.R. Nevins, Q. Wang, and M. West. High-dimensional sparse factor modeling: applications in gene expression genomics. *Journal of the American Statistical Association*, 103(484), 2008.
- C.S.L. Chui, K.K.C. Man, C.L. Cheng, E.W. Chan, W.C.Y. Lau, V.C.C. Cheng, D.S.H. Wong, Y.H. Yang, Kao, and I.C.K. Wong. An investigation of the potential association between retinal detachment and oral fluororoquinolones: A self-controlled case series study. *Journal of Antimicrobial Chemotherapy*, 69(9):2563–2567, 2014.
- M.C. Cobanoglu, C. Liu, F. Hu, Z.N. Olkvari, and I. Bahar. Predicting drug-target interactions using probabilistic matrix factorization. *Journal of Chemical Information and Modeling*, 53(12):3399–3409, 2013.
- P. Farrington. Relative incidence estimation from case series for vaccine safety evaluation. *Biometrics*, 51:228–235, 1995.
- P. C. Farrington, K. Anaya-Igquierdo, H.J. Whitaker, M.N. Hocine, I. Douglas, and L. Smeeth. Self-controlled case series analysis with event-dependent observation periods. *Journal of the American Statistical Association*, 106(494), 2011.
- J. Ghosh and D.B. Dunson. Default prior distributions and efficient posterior computation in Bayesian factor analysis. *Journal of Computational and Graphical Statistics*, 18(2):306–320, 2009.
- S.K. Greene, M. Kulldorff, R. Yin, W.K. Yih, T.A. Lieu, E.S. Weintraub, and G.M. Lee. Near real-time vaccine safety surveillance with partially accrued data. *Pharmacoepidemiology and Drug Safety*, 20(6):583–590, 2011.
- D. Madigan, P. Ryan, S.E. Simpson, and I. Zorych. *Bayesian methods in pharmacovigilance*, volume 9. Oxford University Press, 2010.
- D. Madigan, S. Simpson, W. Hua, A. Paredes, B. Fireman, and M. Macture. The self-controlled case series: Recent developments, 2011.
- D. Madigan, P.E. Strang, J.A. Berlin, M. Schummie, J.M. Overhage, M.A. Suchard, B. Dumenichel, A.G. Hartzema, and P.B. Ryan. A systematic statistical approach to evaluating evidence from observational studies. *Annual Review of Statistics and Its Application*, 1:11–39, 2014.
- M.J. Schummie, G. Tifirò, P.M. Coloma, P.B. Ryan, and D. Madigan. Detecting adverse drug reactions following long-term exposure in longitudinal observational data: The exposure-adjusted self-controlled case series. *Statistical Methods in Medical Research*, 2014.
- S.E. Simpson. A positive event dependence model for self-controlled case series with applications in postmarketing surveillance. *Biometrics*, 69(1):128–136, 2013.
- S.E. Simpson, D. Madigan, I. Zorych, M.J. Schummie, P.B. Ryan, and M.A. Suchard. Multiple self-controlled case series for large-scale longitudinal observational databases. *Biometrics*, 69(4):893–902, 2013.
- M.A. Suchard, I. Zorych, S.E. Simpson, M.J. Schummie, P.B. Ryan, and D. Madigan. Empirical performance of the self-controlled case series design: lessons for developing a risk identification and analysis system. *Drug Safety*, 36(1):83–93, 2013a.
- Marc A. Suchard, Shawn E. Simpson, Ivan Zorych, Patrick Ryan, and David Madigan. Massive parallelization of serial inference algorithms for a complex generalized linear model. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 23(1):10, 2013b.
- H. J. Whitaker, C. P. Farrington, B. Spiessens, and P. Musonda. Tutorial in biostatistics: the self-controlled case series method. *Statistics in Medicine*, 25(10):1768–1797, 2006.
- M. Zitnik and B. Zupan. Matrix factorization-based data fusion for drug-induced liver injury prediction. *Systems Biomedicine*, 2(1):16–22, 2014.

Electronic Health Record Analysis via Deep Poisson Factor Models

Ricardo Henao

*Duke Electrical and Computer Engineering
Duke University, Durham, NC 27708, USA*

R.HENAO@DUKE.EDU

James T. Lu

*Duke School of Medicine
Duke Electrical and Computer Engineering
Duke University, Durham, NC 27708, USA*

JAMES.LU@DUKE.EDU

Joseph E. Lucas

*Duke Electrical and Computer Engineering
Duke University, Durham, NC 27708, USA*

JOE@STAT.DUKE.EDU

Jeffrey Ferranti

*Duke School of Medicine
Duke University, Durham, NC 27708, USA*

JEFFREY.FERRANTI@DUKE.EDU

Lawrence Carin

*Duke Electrical and Computer Engineering
Duke University, Durham, NC 27708, USA*

LCARIN@DUKE.EDU

Editor: Benjamin M. Marlin, C. David Page, and Suchi Saria

Abstract

Electronic Health Record (EHR) phenotyping utilizes patient data captured through normal medical practice, to identify features that may represent computational medical phenotypes. These features may be used to identify at-risk patients and improve prediction of patient morbidity and mortality. We present a novel deep multi-modality architecture for EHR analysis (applicable to joint analysis of *multiple* forms of EHR data), based on Poisson Factor Analysis (PFA) modules. Each modality, composed of observed counts, is represented as a Poisson distribution, parameterized in terms of hidden binary units. Information from different modalities is shared via a deep hierarchy of common hidden units. Activation of these binary units occurs with probability characterized as Bernoulli-Poisson link functions, instead of more traditional logistic link functions. In addition, we demonstrate that PFA modules can be adapted to discriminative modalities. To compute model parameters, we derive efficient Markov Chain Monte Carlo (MCMC) inference that scales efficiently, with significant computational gains when compared to related models based on logistic link functions. To explore the utility of these models, we apply them to a subset of patients from the Duke-Durham patient cohort. We identified a cohort of over 16,000 patients with Type 2 Diabetes Mellitus (T2DM) based on diagnosis codes and laboratory tests out of our patient population of over 240,000. Examining the common hidden units uniting the PFA modules, we identify patient features that represent medical concepts. Experiments indicate that our learned features are better able to predict mortality and morbidity than clinical features identified previously in a large-scale clinical trial.

Keywords: Deep learning, multi-modality learning, Poisson factor model, electronic health records, phenotyping.

1. Introduction

Electronic health records (EHR) are quickly becoming a primary depository of detailed patient health information. These data, if properly analyzed, have the potential to be a nidus for novel insights that may improve patient diagnosis, treatment and safety. In particular, there has been increasing focus on utilizing such data to rapidly identify disease cohorts or “phenotypes” that can be leveraged in clinical and epidemiological studies. However, EHR data, a by-product of the often messy day-to-day interactions of physicians and patients in primary care hospital and emergency room settings, are often challenging to manipulate and interpret without expert input.

Many of the initial EHR phenotyping methods in the literature (Hripsak and Alberts, 2013; Mareedu et al., 2009) relied and continue to rely explicitly on heuristics generated through the collaboration of physicians and informatics. These “computable” phenotypes identified clusters of patients that, for example, suffer from a particular ailment (Newton et al., 2013). Computable phenotypes are often structured similar to decision trees that utilize multiple modes of patients data captured by the EHR¹ to filter patient groups. These modes may include physician and nursing notes from prior encounters, procedure and diagnosis codes, laboratory results, medications, radiology and pathology. Alternatively, other methods have relied on physician-labeled case and control samples (Chen et al., 2013), to identify patient features that may represent a patient phenotype.

Computable phenotypes resemble an analysis that physicians intuitively perform while diagnosing patients. At a high level, physicians assign patients to a latent space of plausible disease phenotypes that inform diagnosis and treatment. This assignment is based on heterogeneous data from the patient interview and physical exam, in combination with other data such as radiology reports, laboratory results and prior medication and medical history. For example, a young child who presents with multiple respiratory infections at an early age increases the probability of a cystic fibrosis phenotype, and thus may be a candidate for associated genetic testing.

Despite their success in *(i)* advancing medical record data mining across large medical institutions and *(ii)* genotype/phenotypes studies, efforts to develop computable phenotypes are nonetheless iterative, manual and difficult to scale. Further, there appears to be widespread variability in disease definitions for even putatively “well-defined” diseases. In Richesson et al. (2013) it was shown that even for phenotypes where there is widespread agreement on the disease definition, such as Type 2 Diabetes Mellitus, definitions by different clinical groups captured different patient populations.

A complementary approach to modeling patient phenotypes from EHR data relies on utilizing unsupervised models. These computational phenotypes have the ability to identify not only feature sets that represent known medical concepts, but they may also discover feature sets that may represent novel phenotypes that are: *(i)* subtypes of and/or *(ii)* run counter to clinically intuited groups. Applied to health-system data and CMS (Centers for Medicare & Medicaid Services) claims data, it has been demonstrated that sparse tensor

1. See <https://phekb.org/>.

factorization of multimodal patient data, transformed into count data, generates concise sets of sparse factors that are recognizable by medical professionals (Ho et al., 2014a,b). Patients can then be treated as a weighted composites of such factors.

While these automated models are efficient at extracting phenotype data and reducing manual input, they have several limitations (Chen et al., 2013; Ho et al., 2014a). Current models are unable to capture correlation both between and within data modes. For example, tensor factorization requires the presence of all modes of patient data within a limited time window to capture the patient-physician interaction. As the number of modes increase, the probability of all modes of data being captured within a limited time window decreases. This prevents leveraging subsets of data modes from (often) limited patient interactions with care givers. Meanwhile, models that concatenate multiple data modes, or evaluate each mode separately, lose correlation between data types. Additionally, current models do not allow one to integrate classification in a straightforward manner. Rather, prediction is conducted in a step-wise manner relying on defining factors first and then entering them into a classification procedure. Current models also often only incorporate a single layer of information, depriving the model of potentially rich higher-level correlation structure within and between modes.

Deep models, understood as multilayer modular networks, have recently generated significant interest from the machine learning community, in part because of their ability to obtain state-of-the-art performance in a wide variety of modalities. Commonly used modules include, but are not limited to, Restricted Boltzmann Machines (RBMs) (Hinton, 2002), Sigmoid Belief Networks (SBNs) (Neal, 1992), convolutional networks (LeCun et al., 1998), feedforward neural networks, and Dirichlet Processes (DPs) (Blei et al., 2004). Deep models are often employed in topic modeling, modeling data characterized by vectors of word counts. As discussed below, EHR data may often be expressed in terms of counts of entities (*e.g.*, counts of types of medications, tests or procedure codes, generalizing the concept of words). Topic models are therefore of interest for EHR data. Examples of deep topic models, composed of DP modules, include the nested Chinese Restaurant Process (nCRP) (Blei et al., 2004), the hierarchical DP (HDP) (Teh et al., 2006), and the nested HDP (nHDP) (Paisley et al., 2015). Alternatively, topic models built using modules other than DPs have been proposed recently, for instance the Replicated Softmax Model (RSM) (Hinton and Salakhutdinov, 2009) based on RBMs, the Neural Autoregressive Density Estimator (NADE) (Larochelle and Laitly, 2012) based on neural networks, the Over-replicated Softmax Model (OSM) (Srivastava et al., 2013) based on DBMs, and Deep Poisson Factor Analysis (DPFA) (Gan et al., 2015a) based on SBNs.

DP-based models have attractive characteristics from the standpoint of interpretability, in the sense that their generative mechanism is parameterized in terms of *distributions over topics*, with each topic characterized by a *distribution over words*. Alternatively, non-DP-based models, in which modules are parameterized by a deep hierarchy of *binary units* (Hinton and Salakhutdinov, 2009; Larochelle and Laitly, 2012; Srivastava et al., 2013), do not have parameters that are as readily interpretable in terms of topics of this type, although model performance is often excellent. The DPFA model in Gan et al. (2015a) is one of the first representations that characterizes documents based on distributions over topics and words, while simultaneously employing a deep architecture based on binary units. Specifically, Gan et al. (2015a) integrates the capabilities of Poisson Factor Analysis (PFA)

(Zhou et al., 2012) with a deep architecture composed of SBNs (Gan et al., 2015b). PFA is a nonnegative matrix factorization framework closely related to DP-based models. Results in Gan et al. (2015a) show that DPFA outperforms other well-known deep topic models.

Building on the success of DPFA, this paper proposes a new deep multi-modality architecture for topic modeling, based entirely on PFA modules. Our model merges two key aspects of DP and non-DP-based architectures, namely: (i) its nonnegative formulation relies on Dirichlet distributions, and is thus readily interpretable throughout all its layers, not just at the base layer as in DPFA (Gan et al., 2015a); (ii) it adopts the rationale of traditional non-DP-based models such as DBNs and DBMs, by connecting different modalities and layers via binary units, to enable learning of high-order statistics and structured correlations within and across modalities. The probability of a binary unit being on is controlled by a Bernoulli-Poisson link (Zhou, 2015) (rather than a logistic link, as in the SBN), allowing repeated application of PFA modules at all layers of the deep architecture. An early version of our approach, for the special case of single-modality data, but mainly focused on topic models was previously described in Henao et al. (2015).

The main contributions of this paper are as follows. (i) We develop a novel deep architecture for topic models based entirely on PFA modules. (ii) The model has inherent shrinkage in all its layers, thanks to the DP-like formulation of PFA. This is unlike DPFA, which is based on SBNs. (iii) The proposed model yields greatly improved mixing, compared to DPFA which requires sequential updates for its binary units; in our formulation these are updated in block. (iv) The proposed approach provides the ability to build deep multi-modality architectures and discriminative topic models with PFA modules. (v) We develop an efficient MCMC inference procedure that scales as a function of the number of *non-zeros* in the data and binary units. In contrast, models based on RBMs and SBNs scale with the size of the data and binary units. Finally, (vi) we demonstrate the applicability of this framework to the analysis of EHR data, with an associated interpretation of the inferred data features (topics and meta-topics, as detailed below).

2. Model

2.1 Poisson factor analysis as a module

Assume \mathbf{x}_n is an M -dimensional vector containing counts of M different entities (*e.g.*, words in documents), for the n -th of N data vectors. We impose the model

$$\mathbf{x}_n \sim \text{Poisson}(\Psi(\theta_n \circ \mathbf{h}_n)), \quad (1)$$

where $\Psi \in \mathbb{R}_+^{M \times K}$ is the factor loadings matrix with K factors, $\theta_n \in \mathbb{R}_+^K$ are factor intensities, $\mathbf{h}_n \in \{0, 1\}^K$ is a vector of binary units indicating which factors are active for observation n , and \circ represents the element-wise (Hadamard) product. The representation in (1) may be expressed as

$$x_{mn} = \sum_{k=1}^K x_{mkn}, \quad x_{mkn} \sim \text{Poisson}(\lambda_{mkn}), \quad \lambda_{mkn} = \psi_{mk} \theta_{kn} h_{kn} \quad (2)$$

where ψ_k is column k of Ψ , ψ_{mk} is component m of ψ_k , x_{mn} is component m of \mathbf{x}_n , θ_{kn} is component k of θ_n , and h_{kn} is component k of \mathbf{h}_n . In (2) we have used the additive property

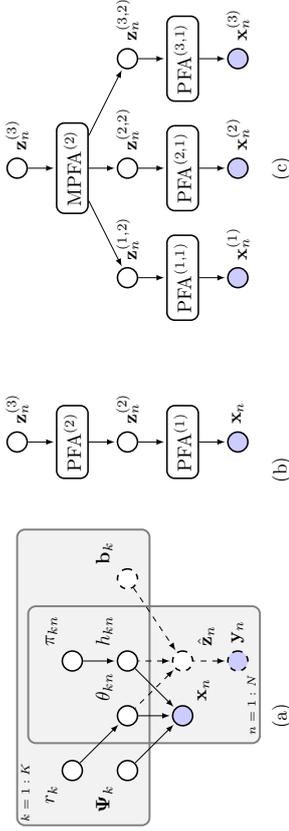


Figure 1: Graphical models. (a) Poisson Factor analysis module in (1)-(3). Nodes $(\mathbf{b}_k, \mathbf{z}_n)$ and \mathbf{y}_n and edges drawn with dashed lines correspond to the discriminative PFA described in Section 2.5. (b) Deep Poisson factor model in (5). (c) Deep Multi-task Poisson factor model in (6). Filled and empty nodes represent observed and latent variables, respectively.

of the Poisson distribution to decompose the m -th observed count of \mathbf{x}_n as K latent counts, $\{z_{mkn}\}_{k=1}^K$.

One possible prior specification for the model in (1), recently introduced by Zhou et al. (2012), is

$$\psi_k \sim \text{Dirichlet}(\eta \mathbf{1}_M), \quad \theta_{kn} \sim \text{Gamma}(r_k, (1-b)b^{-1}), \quad h_{kn} \sim \text{Bernoulli}(\pi_{kn}) \quad (3)$$

where $\mathbf{1}_M$ is an M -dimensional vector of all-ones. Furthermore, for simplicity we let $\eta = 1/K$, $b = 0.5$ and $r_k \sim \text{Gamma}(1, 1)$. Prior distributions for η and b that result in closed form conditionals exist, and can be used if desired; see for instance Escobar and West (1995) for η , and Zhou and Carin (2015) for b .

There is one parameter in (3) for which we have not specified a prior distribution, specifically $\mathbb{E}[\rho(h_{kn} = 1)] = \pi_{kn}$. In Zhou et al. (2012), h_{kn} is provided with a beta-Bernoulli process prior by letting $\pi_{kn} = \pi_k \sim \text{Beta}(c\epsilon, c(1-\epsilon))$, where usually $c = 1$ and $\epsilon = 1/K$, meaning that each of the N data vectors has on average the same probability of seeing a particular topic as active. It further assumes topics are independent of each other. These two assumptions are restrictive because: (i) in practice, the N data vectors often belong to a heterogeneous population (*e.g.*, patients); letting the data vectors have individual topic activation probabilities allows the model to better accommodate heterogeneity in the data. (ii) Some topics are likely to co-occur systematically, so being able to harness such correlation structures can improve the ability of the model for fitting the data.

The hierarchical model in (1)-(3) is denoted $\mathbf{x}_n \sim \text{PFA}(\Psi, \theta, \mathbf{h}_n; \eta, r_k, b)$, short for Poisson Factor Analysis (PFA), with graphical model representation shown in Figure 1(a). The model in (1)-(3) is closely related to other widely known topic model approaches, such as Latent Dirichlet Allocation (LDA) (Blei et al., 2003), HDP (Teh et al., 2006) and Focused Topic Modeling (FTM) (Williamson et al., 2010). Connections between these models are discussed in Section 2.7.

2.2 Deep representations with PFA modules

Several models have been proposed recently to address the limitations described in the previous section (Blei et al., 2004; Blei and Lafferty, 2007; Gan et al., 2015a; Teh et al., 2006). In particular, Gan et al. (2015a) proposed using multilayer SBNs (Neal, 1992) to impose correlation structure across topics, while providing each data vector with the ability to control its topic activation probabilities, without the need of a beta-Bernoulli process (Zhou et al., 2012). Here we follow the same rationale as Gan et al. (2015a), but without SBNs. We start by noting that for a binary vector \mathbf{h}_n with elements h_{kn} , we can write

$$h_{kn} = \mathbf{1}(z_{kn} \geq 1), \quad z_{kn} \sim \text{Poisson}(\tilde{\lambda}_{kn}), \quad (4)$$

where z_{kn} is a latent count for variable h_{kn} , parameterized by a Poisson distribution with rate $\tilde{\lambda}_{kn}$. The function $\mathbf{1}(\cdot)$ is defined as $\mathbf{1}(\cdot) = 1$ if the argument is true, and $\mathbf{1}(\cdot) = 0$ otherwise. The model in (4), recently proposed in Zhou (2015), is known as the Bernoulli-Poisson Link (BPL) and is denoted $\mathbf{h}_n \sim \text{BPL}(\tilde{\lambda}_n)$, for $\tilde{\lambda}_n \in \mathbb{R}_+^K$. After marginalizing out the latent count z_{kn} (Zhou, 2015), the model in (4) has the interesting property that $p(h_{kn} = 1) = \text{Bernoulli}(\pi_{kn})$, where $\pi_{kn} = 1 - \exp(-\tilde{\lambda}_{kn})$. In order to sample h_{kn} we do not need to instantiate latent count z_{kn} but the rate of its underlying distribution $\tilde{\lambda}_{kn}$. Hence, rather than using the logistic function to represent binary unit probabilities as in SBNs, we employ $\pi_{kn} = 1 - \exp(-\tilde{\lambda}_{kn})$.

The binary distribution based on $p(h = 1) = \text{Bernoulli}(1 - \exp(-\tilde{\lambda}))$ is reminiscent of the complementary log-log link function (Piegorisch, 1992; Collett, 2002), where $\tilde{\lambda} = \exp(-u)$ and $u \in \mathbb{R}$. Unlike the logistic function, that is symmetric around the origin, $u = 0$ for $p(h = 1) = \text{Bernoulli}(1/(1 + \exp(-u)))$, the complementary log-log link is asymmetric, making it appropriate for imbalanced modalities, where the proportion of zeros is large. In our setting, this behavior is ideal because it encourages sparsity, in that it supports the assumption that a given data vector (patient) is explained by a small subset of topics selected via binary units, \mathbf{h}_n .

In (2) and (4) we have represented the Poisson rates as λ_{mkn} and $\tilde{\lambda}_{kn}$, respectively, to distinguish between the two. However, the fact that the count vector in (3) and the binary variable in (4) are both represented in terms of Poisson distributions suggests the following deep model, based on PFA modules

$$\begin{aligned} \mathbf{x}_n &\sim \text{PFA}(\Psi^{(1)}, \theta_n^{(1)}, \mathbf{h}_n^{(1)}; \eta^{(1)}, r_k^{(1)}, b^{(1)}), & \mathbf{h}_n^{(1)} &= \mathbf{1}(\mathbf{z}_n^{(2)}), \\ \mathbf{z}_n^{(2)} &\sim \text{PFA}(\Psi^{(2)}, \theta_n^{(2)}, \mathbf{h}_n^{(2)}; \eta^{(2)}, r_k^{(2)}, b^{(2)}), & & \vdots \\ & & & \vdots \\ & & & \mathbf{h}_n^{(L-1)} = \mathbf{1}(\mathbf{z}_n^{(L)}), \\ \mathbf{z}_n^{(L)} &\sim \text{PFA}(\Psi^{(L)}, \theta_n^{(L)}, \mathbf{h}_n^{(L)}; \eta^{(L)}, r_k^{(L)}, b^{(L)}), & \mathbf{h}_n^{(L)} &= \mathbf{1}(\mathbf{z}_n^{(L+1)}), \end{aligned} \quad (5)$$

where L is the number of layers in the model, and $\mathbf{1}(\cdot)$ is a vector operation in which each component imposes the left operation in (4). In this Deep Poisson Factor Model (DPFM), shown as a graphical model in Figure 1(b) and also previously described in Henao et al. (2015), the binary units at layer $\ell \in \{1, \dots, L\}$ are drawn $\mathbf{h}_n^{(\ell)} \sim \text{BPL}(\tilde{\lambda}_n^{(\ell+1)})$, for

$\lambda_n^{(l)} = \Psi^{(l)}(\theta_n^{(l)} \circ \mathbf{h}_n^{(l)})$. The form of the model in (5) introduces latent variables $\{\mathbf{z}_n^{(l)}\}_{l=2}^{L+1}$ and the element-wise function $\mathbf{1}(\cdot)$, rather than explicitly drawing $\{\mathbf{h}_n^{(l)}\}_{l=1}^L$ from the BPL distribution. Concerning the top layer, we let $\mathbf{z}_n^{(L+1)} \sim \text{Poisson}(\lambda_n^{(L+1)})$ and $\lambda_n^{(L+1)} \sim \text{Gamma}(a_0, b_0)$.

2.3 Deep multi-modality representation with PFA modules

In a multi-modality setting, each individual is characterized by D different count vectors, each of which is characterized by a different vocabulary (different types of entities being counted, *i.e.*, different modalities). Individual $n \in \{1, \dots, N\}$, data type $i \in \{1, \dots, D\}$ is denoted as $\mathbf{x}_n^{(i)}$, corresponding to an M_i -dimensional count vector. The dataset described in Section 3 is composed of $D = 3$ data types: medications, laboratory tests and codes. Since all D data types are composed of count vectors, we can in principle concatenate the D vectors for patient n into a long $\sum_i M_i$ -dimensional vector, $\left[\mathbf{x}_n^{(1)\top}, \dots, \mathbf{x}_n^{(D)\top}\right]^\top$, that we can model with the DPFM in (5). Such an approach will allow us to learn about the correlation structure of the variables in the concatenated modalities, but it ignores the fact that due to context, each data type in general has its own correlation structure. Another simple approach consists of modeling each data type individually, again using (5); however, this fails to acknowledge that different modality types can be correlated, as they represent different contexts or “views” of a larger representational space. In Henao et al. (2015), the authors employ this single-modality approach to model medications from the dataset described in Section 3, however they do not provide insights on how to combine multiple modalities. Motivated by the shortcomings of these two simplistic approaches, we modify the model in (5) to learn correlation structures of individual modalities, but at the same time to be able to share information across them to leverage their correlation structure. In particular, we propose a data-type-specific first layer and a deep architecture of shared PFA modules, formally written as

$$\begin{aligned} \mathbf{x}_n^{(i)} &\sim \text{PFA}^{(i+1)}, & \mathbf{h}_n^{(i+1)} &= \mathbf{1}\left(\mathbf{z}_n^{(i+2)}\right), & i &= 1, \dots, D, \\ \mathbf{z}_n^{(1+2)}, \dots, \mathbf{z}_n^{(D+2)} &\sim \text{MPPFA}^{(2)}, & & & \\ & \vdots & \mathbf{h}_n^{(L-1)} &= \mathbf{1}\left(\mathbf{z}_n^{(L)}\right), & \\ \mathbf{z}_n^{(L)} &\sim \text{PFA}^{(L)}, & \mathbf{h}_n^{(L)} &= \mathbf{1}\left(\mathbf{z}_n^{(L+1)}\right), & \end{aligned} \quad (6)$$

where

$$\text{PFA}^{(i+1)} \stackrel{\text{def}}{=} \text{PFA}\left(\Psi^{(i+1)}, \theta_n^{(i+1)}, \mathbf{h}_n^{(i+1)}, \eta_n^{(i+1)}, r_k^{(i+1)}, b^{(i+1)}\right), \quad (7)$$

$$\text{MPPFA}^{(2)} \stackrel{\text{def}}{=} \prod_i \text{PFA}\left(\Psi^{(i+2)}, \theta_n^{(i+2)}, \mathbf{h}_n^{(i+2)}, \eta_n^{(i+2)}, r_k^{(i+2)}, b^{(i+2)}\right), \quad (8)$$

$$\mathbf{z}_n^{(i+2)} \sim \text{PFA}\left(\Psi^{(i+2)}, \theta_n^{(i+2)}, \mathbf{h}_n^{(i+2)}, \eta_n^{(i+2)}, r_k^{(i+2)}, b^{(i+2)}\right). \quad (9)$$

The first layer in (6) is composed of D independent PFA modules as in (7), with explicit hierarchical model in (3). The multi-modality PFA model, denoted MPPFA in (8), is a PFA model

in which each modality has an associated factor loadings matrix, $\Psi^{(i+2)}$, but shared factor intensities, $\theta_n^{(i+2)}$, binary units, $\mathbf{h}_n^{(i+2)}$ and parameters $\{\eta_n^{(i+2)}, r_k^{(i+2)}, b^{(i+2)}\}$. This means that modality-specific latent counts, $\mathbf{z}_n^{(i+2)}$, can be drawn from a PFA module restricted to $\Psi^{(i+2)}$ as in (9). The MPPFA module in (8) is a novel specification, not previously considered by Henao et al. (2015), that extends the functionality of Poisson factor analysis beyond applications in electronic health records analysis. The architecture of the Deep Multi-modality Poisson Factor Model (DMPFM) in (6) is fully specified by $\{K_{(1,1)}, \dots, K_{(D,1)}, K_{(2)}, \dots, K_{(L)}\}$ and L , where $K_{(i,1)}$ are modality-specific loadings sizes (number of topics), $K_{(A)}$ are modality-shared loadings sizes and L is the number of layers. For example, Figure 1 shows a graphical model representation for a specification with $D = 3$ and $L = 2$. When the model is set for a single-modality, $D = 1$, the architecture in (6) is equivalent to DPFM, as shown in (5) and previously described by Henao et al. (2015).

2.4 Model interpretation

Consider modality i in layer 1 of (6), from which $\mathbf{x}_n^{(i)}$ is drawn. Assuming $\mathbf{h}_n^{(i+1)}$ is known, this corresponds to a focused topic model (Williamson et al., 2010). The columns of $\Psi^{(i+1)}$ correspond to modality- i topics, with the k -th column $\psi_k^{(i+1)}$ defining the probability with which entities (*e.g.*, medications) of modality i are manifested for topic k (each $\psi_k^{(i+1)}$ is drawn from a Dirichlet distribution, as in (3)). Generalizing the notation from (2), $\lambda_{kn}^{(i+1)} = \psi_k^{(i+1)} \theta_n^{(i+1)} h_{kn}^{(i+1)} \in \mathbb{R}^{M_i}$ is the rate vector associated with topic k , modality i and patient n , and it is active when $h_{kn}^{(i+1)} = 1$. The entity-count vector for patient n in modality i manifested from topic k is $\mathbf{x}_{kn}^{(i)} \sim \text{Poisson}\left(\lambda_{kn}^{(i+1)}\right)$, and $\mathbf{x}_n^{(i)} = \sum_{k=1}^{K_{(i,1)}} \mathbf{x}_{kn}^{(i)}$, where $K_{(i,1)}$ is the number of topics in the modality. The columns of $\Psi^{(i+1)}$ define correlation among the entities associated with the topics: for a given topic (column of $\Psi^{(i+1)}$), some entities co-occur with high probability, and other entities are likely jointly absent.

We now consider a two-layer model, with $\mathbf{h}_n^{(2)}$ assumed known. To generate $\mathbf{h}_n^{(i+1)}$, we first draw $\mathbf{z}_n^{(i+2)}$, which, analogous to above, may be expressed as $\mathbf{z}_n^{(i+2)} = \sum_{k=1}^{K_{(i+2)}} \mathbf{z}_{kn}^{(i+2)}$, with $\mathbf{z}_{kn}^{(i+2)} \sim \text{Poisson}\left(\lambda_{kn}^{(i+2)}\right)$ and $\lambda_{kn}^{(i+2)} = \psi_k^{(i+2)} \theta_n^{(i+2)} h_{kn}^{(i+2)}$. Note that factor intensities and binary units, respectively $\theta_n^{(i+2)}$ and $h_{kn}^{(i+2)}$, are shared across the $i = 1, \dots, D$ modalities. Column k of $\Psi^{(i+2)}$ corresponds to a *meta-topic* specific to modality i , with $\psi_k^{(i+2)}$ a $K_{(i,1)}$ -dimensional probability vector, denoting the probability with which each of the modality- i layer-1 topics are “on” when layer-2 “meta-topic” k is on (*i.e.*, when $h_{kn}^{(i+2)} = 1$). The columns of $\Psi^{(i+2)}$ define correlation among the modality- i layer-1 topics: for a given layer-2 meta-topic (column of $\Psi^{(i+2)}$), some layer-1 topics co-occur with high probability, and other layer-1 topics are likely jointly absent. Furthermore, columns of the concatenated meta-topic matrix, $\left[\Psi^{(1+2)}\right]^\top \dots \left[\Psi^{(D+2)}\right]^\top$, define correlation structure among all layer-1 topics at the same time.

As one moves up the hierarchy, to layers $\ell > 2$, the meta-topics become increasingly more abstract and sophisticated, manifested in terms of probabilistic combinations of topics and meta-topics at the layers below. Because of the properties of the Dirichlet distribution, each column of a particular $\Psi^{(l)}$ is encouraged to be sparse, implying that a column of $\Psi^{(l)}$

encourages use of a small subset of columns of $\Psi^{(\ell-1)}$, with this repeated all the way down to the data layer, and the topics reflected in the columns of $\Psi^{(1)}$. This deep architecture imposes correlation across the layer-1 topics in all modalities, and it does it through use of PFA modules at all layers of the deep architecture, unlike Gan et al. (2015a) which uses an SBN for layers 2 through L , and a PFA at the bottom layer. In addition to the elegance of using a single class of modules at each layer, the proposed deep model has important computational benefits, as discussed in Section 2.6.

We emphasize that $\{\theta_{kn}^{(2)}, h_{kn}^{(2)}\}$ are shared across all D data types, or modalities. The hierarchy that resides above them is meant to model underlying latent correlations in aspects of disease and health. The underlying state of the patient is independent of the modality with which he/she is viewed. When $h_{kn}^{(2)} = 1$, the k th meta-topic of health/disease is “on” for patient n ; $\psi_k^{(i,2)}$ characterizes how meta-topic k impacts the presence/absence of each topic associated with modality i . The modality-dependence is manifested at the bottom of the deep model, near the data, and the deep architecture above it imposes statistical relationships among the meta-topics, and is meant to characterize latent health/disease.

2.5 PFA modules for multi-label classification

Assume there is a C -dimensional vector of binary labels $\mathbf{y}_n \in \{0, 1\}^C$ associated with patient n (presence/absence of C maladies or illnesses). Provided that labels share the same covariates (patient n , \mathbf{x}_n) and are oftentimes correlated, it is reasonable to model all labels jointly as opposed to build individual models for each label. We seek to learn the model for mapping $\mathbf{x}_n \rightarrow \mathbf{y}_n$ simultaneously with learning the deep topic representation in Section (2.2). In fact, the mapping $\mathbf{x}_n \rightarrow \mathbf{y}_n$ is based on the deep generative process for \mathbf{x}_n in (5). This means that we can leverage the correlation structure of count data vectors and labels at the same time. We represent each element of \mathbf{y}_n , y_{cn} , using (4). We impose the model

$$y_{cn} = \mathbf{1}(\hat{z}_{cn} \geq 1), \quad \hat{z}_{cn} \sim \text{Poisson}(\hat{\lambda}_{cn}), \quad (10)$$

where $\hat{\lambda}_{cn}$ is element c of $\hat{\lambda}_n$. First considering the single-modality case, $\hat{\lambda}_n = \mathbf{B}(\boldsymbol{\theta}_n^{(1)} \circ \mathbf{h}_n^{(1)})$ and $\mathbf{B} \in \mathbb{R}_+^{C \times K}$ is a matrix of nonnegative classification weights, with prior distribution $\mathbf{b}_k \sim \text{Dirichlet}(\zeta \mathbf{1}_C)$, where \mathbf{b}_k is a column of \mathbf{B} . Here, we denote latent counts as $\hat{\mathbf{z}}_n = [\hat{z}_{1n} \dots \hat{z}_{Cn}]^T$ to differentiate them from those coming from the DPFM, denoted as \mathbf{z}_n in (5). The matrix of classification weights, \mathbf{B} , in (10) serves two purposes: (i) learns the correlation structure of labels, since large entries in \mathbf{b}_k , say $b_{c'k}$ and $b_{c''k}$ indicate their corresponding labels, c and c' are proportionally correlated; and (ii) provided that the prior for \mathbf{B} encourages sparsity, the resulting classifier is parsimonious and easier to interpret than that of a classifier with dense \mathbf{B} .

Figure 1(a) shows a graphical model representation of a PFA module connected to the multi-label classifier in (10), where solid nodes and edges represent PFA module components and dashed lines are specific to the classification model. Combining (5) with (10) allows us to learn the mapping $\mathbf{x}_n \rightarrow \mathbf{y}_n$ via the shared first-layer local representation, $\boldsymbol{\theta}_n^{(1)} \circ \mathbf{h}_n^{(1)}$, that encodes topic usage for document n . This sharing mechanism allows the model to learn

topics, $\Psi^{(1)}$, and meta-topics, $\{\Psi^{(\ell)}\}_{\ell=2}^L$, biased towards discrimination, as opposed to just explaining the data, \mathbf{x}_n .

For the deep *multi-modality* model in (6), we learn the mapping $\mathbf{x}_n^{(1)}, \dots, \mathbf{x}_n^{(D)} \rightarrow \mathbf{y}_n$ through the first-layer local representations from all modalities, hence $\hat{\lambda}_n = \sum_{i=1}^D \mathbf{B}_i(\boldsymbol{\theta}_n^{(i,1)} \circ \mathbf{h}_n^{(i,1)})$, where $\mathbf{B}_i \in \mathbb{R}_+^{C \times K(i,1)}$, for $i = 1, \dots, D$. In this case, the classifier uses information from all modalities but at the same time biases modality-specific topics, $\Psi^{(i,1)}$, towards discrimination. We call this construction *discriminative* deep multi-modality Poisson factor model. In cases where multi-class, not multi-label classification is required, we can use the formulation introduced by Henao et al. (2015), based on a multinomial likelihood function, instead of a Bernoulli-Poisson link as in (10). Although other DP-based discriminative topic models have been proposed (Lacoste-Julien et al., 2009; McAuliffe and Blei, 2008), they rely on approximations in order to combine the topic model, usually LDA, with softmax-based classification approaches.

2.6 Inference

A convenient feature of the model in (5) and (6) is that all its conditional posterior distributions can be written in closed form, due to local conjugacy. In this section, we focus on Markov chain Monte Carlo (MCMC) via Gibbs sampling for our implementation. In applications where the fully Bayesian treatment becomes prohibitive computationally, Stochastic Variational Inference (SVI) can be used (Henao et al., 2015). See Appendix A for details about SVI implementation for models based on PFA modules. Other alternatives for scaling up inference in Bayesian models such as the parameter server (Ho et al., 2013; Li et al., 2014), conditional density filtering (Guhaniyogi et al., 2014) and stochastic gradient-based approaches (Chen et al., 2014; Ding et al., 2014; Welling and Teh, 2011), are also possible but beyond the scope of this work.

Gibbs sampling for the model in (5) and (6) involves sampling in sequence from the conditional posterior of all the parameters of the model. For instance, for the DPFM in (5), we have $\{\Psi^{(\ell)}, \boldsymbol{\theta}_n^{(\ell)}, \mathbf{h}_n^{(\ell)}, r_k^{(\ell)}\}$, for $\ell = 1, \dots, L$, and $\boldsymbol{\lambda}^{(x+1)}$. For the multi-modality model in (6) we also have to account for modality-specific parameters in (7). The remaining parameters of the model are set to fixed values: $\eta = 1/K$, $b = 0.5$ and $a_0 = b_0 = 1$. We note that priors for η , b , a_0 and b_0 exist that result in Gibbs-style updates, and can be readily incorporated into the model if desired; however, we opted to keep the model as simple as possible, without compromising flexibility. The most unique conditional posteriors for a single PFA module are shown below, without layer index for clarity,

$$\begin{aligned} \psi_k &\sim \text{Dirichlet}(\eta + x_{1k}, \dots, \eta + x_{Mk}), \\ \theta_{kn} &\sim \text{Gamma}(r_k b_{kn} + x_{\cdot kn}, b^{-1}), \\ h_{kn} &\sim \delta(x_{\cdot kn} = 0) \text{Bernoulli}(\tilde{\pi}_{kn}(\tilde{\pi}_{kn} + 1 - \pi_{kn})^{-1}) + \delta(x_{\cdot kn} \geq 1), \end{aligned} \quad (11)$$

where

$$x_{mk} = \sum_{n=1}^N x_{mkn}, \quad x_{\cdot kn} = \sum_{m=1}^M x_{mkn}, \quad \tilde{\pi}_{kn} = \pi_{kn}(1-b)^{r_k}.$$

Complete details, including those for DMPEM and discriminative DMPEM in Sections 2.3 and 2.5, respectively, are provided in Appendix B.

Initialization is done at random from prior distributions, followed by modality-wise and layer-wise fitting (*pre-training*). In the experiments, when pre-training we run 150 Gibbs sampling cycles per layer. We have observed that 50 cycles are usually enough to obtain good initial values of the global parameters of the model, namely $\{\Psi^{(i,1)}, r_k^{(i,1)}, \Psi^{(i)}, r_k^{(i)}\}$, for $i = 1, \dots, D$, $\ell = 2, \dots, L$ and $\lambda^{(L+1)}$.

2.6.1 IMPORTANCE OF COMPUTATIONS SCALING WITH THE NUMBER OF NON-ZEROS

From a practical standpoint, the most important feature of the models in (5) and (6) is that inference does not scale as a function of the size of the total dataset, but as a function of its number of non-zero elements, which is advantageous in cases where the input data is sparse (often the case). For instance, $\sim 4\%$ of the entries in the dataset described in Section 3 are non-zero. Similar proportions are also observed in datasets traditionally used to benchmark topic models (word documents), such as 20 Newsgroups, Reuters and Wikipedia (details of which are discussed below). Furthermore, this feature also extends to all modalities and layers of the model, regardless of $\{\mathbf{h}_n^{(\ell)}\}$ being latent. Similarly, for the discriminative DMPEM in Section 2.5, inference scales with the number of positive cases in $\{\mathbf{y}_n\}_{n=1}^N$, not CN . This is particularly appealing in cases where C is large and the number of positive cases is small (a patient typically has a small subset of possible illnesses), $\sim 8\%$ in the dataset described in Section 3.

In order to show that this scaling behavior holds, it is enough to see that by construction, from (2), if $x_{mn} = \sum_{k=1}^K x_{mkn} = 0$ (or $z_{mn} = 0$ for $\ell > 1$), thus $x_{mkn} = 0$, $\forall k$ with probability 1. Besides, from (4) we see that if $h_{kn} = 0$ then $z_{kn} = 0$ with probability 1. As a result, update equations for all parameters of the model except for $\{\mathbf{h}_n^{(\ell)}\}$, depend only on non-zero elements of \mathbf{x}_n and $\{z_n^{(\ell)}\}$. Updates for the binary variables can be cheaply obtained in block from $h_{kn}^{(\ell)} \sim \text{Bernoulli}(\pi_{kn}^{(\ell)})$ via $\lambda_{kn}^{(\ell)}$ as previously described.

It is worth mentioning that models based on multinomial or Poisson likelihoods such as LDA (Blei et al., 2003), HDP (Teh et al., 2006), FTM (Williamson et al., 2010) and PFA (Zhou et al., 2012), also enjoy this property (scaling based on number of non-zero observations). However, the recently proposed deep PFA (Gan et al., 2015a) does not use PFA modules on layers other than the first one: it uses SBNs or RBMs that are known to scale with the number of binary variables as opposed to their non-zero elements.

2.7 Related work

2.7.1 CONNECTIONS TO OTHER DP-BASED TOPIC MODELS

PFA is a nonnegative matrix factorization model with Poisson link, that is closely related to other DP-based models. Specifically, Zhou et al. (2012) showed that by making $p(h_{kn} = 1) = 1$ and letting θ_{kn} have a Dirichlet, instead of a gamma distribution as in (3), we can recover LDA by using the equivalence between Poisson and multinomial distributions. By looking at (11), we see that PFA and LDA have the same blocked Gibbs updates (Blei et al., 2003), when Dirichlet distributions for θ_{kn} are used. An equivalent analogy for SVI updates (Hoffman et al., 2010) can be derived from the update equations in Appendix A.

In Zhou et al. (2012), the authors showed that using the Poisson-gamma representation of the negative binomial distribution and a beta-Bernoulli specification for $p(h_{kn})$ in (3), we can recover the FTM formulation and inference in Williamson et al. (2010). More recently, Zhou and Carin (2015) showed that PFA is comparable to HDP in that the former builds group-specific DPs with normalized gamma processes. A more direct relationship between a three-layer HDP (Teh et al., 2006) and a two-layer version of (5) can be established by grouping count data vectors by categories. In the HDP, three DPs are set for topics, data-dependent topic usage and category-wise topic usage. In our model, $\Psi^{(1)}$ represent K_1 topics, $\theta_n^{(1)} \circ \mathbf{h}_n^{(1)}$ encodes data-vector-wise topic usage and $\Psi^{(2)}$ encodes topic usage for K_2 categories. In HDP, data vectors are assigned to categories a *a priori*, but in our model data-vector-category *soft* assignments are estimated and encoded via $\theta_n^{(2)} \circ \mathbf{h}_n^{(2)}$. As a result, the model in (5) is a more flexible alternative to HDP in that it groups data vectors into categories in an unsupervised manner.

2.7.2 SIMILAR MODELS

Non-DP-based deep models for topic modeling employed in the deep learning literature typically utilize RBMs or SBNs as building blocks. For instance, Hinton and Salakhutdinov (2009) and Maaloe et al. (2015) extended RBMs via DBNs to topic modeling. In addition, Srivastava et al. (2013) proposed the over-replicated softmax model, a deep version of RSM that generalizes RBMs.

Recently, Ranganath et al. (2014) proposed a framework for generative deep models using exponential family modules. Although they consider Poisson-Poisson and gamma-gamma factorization modules akin to our PFA modules, their model lacks the explicit binary unit linking between layers commonly found in traditional deep models. Further, their inference approach, *black-box* variational inference, is not as conceptually simple, but it scales with the number of non-zeros of our model.

DPEA, proposed in Gan et al. (2015a), is the model closest to ours. Nevertheless, our proposed model has a number of key differentiating features. *(i)* Both models learn topic correlations by building a multilayer modular representation on top of PFA. Our model uses PFA modules throughout all layers in a conceptually simple and easy to interpret way. DPEA uses Gaussian distributed weight matrices within SBN modules; these are hard to interpret in the context of topic modeling. *(ii)* SBN architectures have the shortcoming of not having block closed-form conditional posteriors for their binary variables, making them difficult to estimate, especially as the number of variables increases. *(iii)* Factor loading matrices in PFA modules have natural shrinkage to counter overfitting, thanks to the Dirichlet prior used for their columns. In SBN-based models, shrinkage has to be added via variable augmentation at the cost of increasing inference complexity. *(iv)* Inference for SBN modules scales with the number of hidden variables in the model, not with the number of non-zero elements, as in our case.

In Henaq et al. (2015), we presented an early version of our approach, for the single-modality case in (5), but mainly focused on topic models. In this previous work, we introduced inference procedures based on Gibbs sampling and stochastic variational inference, and considered a discriminative model for multi-class classification. In the present work, we extend the DPEA architecture to multiple modalities (DMPEA) and introduce a discrimi-

native model specification for multi-label classification, of particular interest in electronic health records applications, as one patient may suffer from multiple illnesses at the same time. In the experiments, we will show the benefits of explicitly modeling multiple modalities using a DMPFM specification, as opposed to naive constructs based on the single-modality model of Henaio et al. (2015).

Several deep architectures have been recently proposed for multi-modality problems (Srivastava and Salakhutdinov, 2012, 2014; Sohn et al., 2014). These models use RBMs as building blocks and are traditionally geared towards applications with image (pixel intensities) and text (word counts) modalities. The main goals of these applications are classification based on image and text latent features, and information retrieval, that is, predicting values of one modality given observations of the others. Unlike our discriminative DMPFM and SupDocNADE (Supervised Document Neural Autoregressive Distribution Estimator Zheng et al., 2014) based on SBNs, most existing deep multi-modality models based on RBMs build classifiers as a two-step procedure, not jointly with the generative model as in our case. In its current form, our model does not allow for mixed data-types, however it is not too difficult to extend it to such case, as we can seamlessly use sparse Gaussian factor models (Carvalho et al., 2008; Henaio and Winther, 2011; Henaio et al., 2014) and rank-likelihood factor models (Yuan et al., 2015) as first-layer modules for real and ordinal-valued data, respectively. We leave these extensions as interesting future work.

3. Motivating Data

We utilize three modes of data: self-reported medication usage, laboratory tests, and diagnosis and procedure codes. Count matrices for each mode for each patient were extracted from a Duke University 5-year dataset. Specifically, we consider electronic health data generated from 2007 to 2011 in the care of Durham County residents within the Duke University Health System (DUHS), including three hospitals and an extensive network of outpatient clinics. This dataset includes over 240,000 patients with over 4.4 million patient visits.

3.1 Data Reconciliation

Patient data originated from the various hospitals and outpatient clinics of DUHS. As names for medications, laboratory tests and diagnosis and procedure codes are uniquely named at each facility, the data must first be reconciled to a common data dictionary.

Our dataset included 39,429 medication names. These names, which included both brand and generic names at various dosages and formulations, were mapped to their pharmaceutical active ingredients (AI) using a custom Python script that leveraged the RxNorm API². RxNorm is a depository of medication information maintained by the National Library of Medicine and includes trade names, brand names, dosage information and active ingredients (Nelson et al., 2011). Compound medications that include multiple active ingredients incremented counts for all AI in that medication. We discovered 1,694 unique AI in our dataset.

The data also include 4,391 types of laboratory tests, mapped to the Logical Observation Identifiers Names and Codes (LOINC) ontology (Vreeman et al., 2015). The LOINC

2. See <http://rxnav.nlm.nih.gov/RxNormAPIs.html>.

standard is common terminology for laboratory and clinical observations maintained by the Regenstrief Institute³. Mappings to the LOINC database were performed with the RELMA tool⁴. Each suggested mapping was reviewed by a physician to ensure that appropriate test and measurement units were aligned. Counts for patient laboratory tests reflect the number of times a test appears in a patients record. We discovered 1,869 unique LOINC tests in the data.

Lastly, the data include 21,305 diagnosis and procedure codes. These were mapped using their unique ICD-9 (International Statistical Classification of Diseases and Related Health Problem) and CPT (Current Procedural Terminology) identifiers. ICD codes are the international diagnostic coding system, and are maintained by the World Health Organization⁵. CPT procedure codes are maintained by the American Medical Association and are designed to ease uniform communication performed medical services⁶. We identified 21,013 unique ICD-9 and CPT codes in the dataset.

3.2 Cohort and Count Matrix Generation

To narrow our analysis, we focused on a cohort of Type-2 Diabetes Mellitus (T2DM) patients, using previous phenotype criteria for T2DM (Richesson et al., 2013). T2DM is a chronic disease with high disease and treatment costs. Patients with diabetes are at increased risk of complications such as Coronary Heart Disease (CHD), Acute Myocardial Infarction (AMI), Cerebral Vascular Disease (CVD), Chronic Renal Failure (CRF), and amputation (American Diabetes Association, 2014). Prediction of these outcomes is important for communicating prognosis and targeting treatment to the high-risk patients.

We identified 16,756 patients in the dataset, by filtering for the following criteria: (i) at least two counts of 250.xx ICD-9 codes, (ii) at least one laboratory measurement of hemoglobin A1c (HgbA1C) greater than 4.5%, and (iii) a medication record including at least one of the following T2DM medications: insulin, metformin, sulfonylurea, or sitagliptin. We generated counts for each data mode by mapping each patient's records to the common data elements as described above. We then counted the total number of occurrences for each data element over a defined time window. In our initial experiment exploring the mapping of medical concepts to discovered factors, this time window represented two years of data. In our classification experiment, this time window was six months prior to the classification date.

3.3 Classification

3.3.1 UK PROSPECTIVE DIABETES STUDY (UKPDS) OUTCOMES MODEL

Prediction equations to determine the risk of various complications in diabetes have been studied extensively (Wilson et al., 1998; Clarke et al., 2004; van Dieren et al., 2011). These risk estimates are helpful for identifying high-risk populations that may need closer clinical observation and higher intensity treatment (Simmons et al., 2009). Several equations are currently available to estimate CHD, AMI and CVD risk (Metcalf et al., 2008; Tao et al.,

3. See <http://loinc.org/>.

4. See <http://loinc.org/downloads/reлма>.

5. See <http://www.who.int/classifications/icd/en/>.

6. See <http://www.ama-assn.org/ama>.

Outcome	ICD-9 codes	CPT codes
Acute Myocardial Infarction	410.*	—
Amputation	84.1*	—
Cardiac Catheterization	—	37.2*
Coronary Artery Disease	411.*, 413* and 414*	45.8*
Depression	293.*, 296.*, 300.4 and 311.*	—
Heart Failure	428.*	—
Kidney Disease	585.*, 249.* and 250.4*	56.1*
Neurological Diseases	249.6* and 250.6*	—
Obesity	278.*	85.*
Ophthalmic Disease	249.5* and 250.5*	—
Stroke	346.6*, 430.*, 431.*, 432.*, 433.*, 434.* and 435*	—
Unstable Angina	411.1	—
Death	date of death in the medical record	

Table 1: Definition of the 13 T2DM related outcomes for multi-label classification experiment in Section 2.5.

2013). UKPDS is a multicenter randomized trial involving 5,102 newly diagnosed patients with T2DM, recruited from 23 UK centers (King et al., 1999; Stevens et al., 2001); it has been utilized to generate outcome models for cardiovascular and cerebrovascular disease (Lu et al., 2012; Tao et al., 2013).

In the UKPDS model, the 1-year probability of CHD is:

$$\begin{aligned}
 p(\text{CHD}) \propto & b_0 + b_1 * (\text{Age} - 55) - b_2 \text{Female} - b_3 \text{AfroCaribbean} \\
 & + b_4 \text{Smoking} + b_5 (\text{HgbA1c} - 6.72) + b_6 (\text{SPB} - 135.7) / 10 \\
 & + b_7 (\log(\text{TC}/\text{HDL}) - 1.59),
 \end{aligned} \tag{12}$$

where mean Total Cholesterol (TC), mean High Density Lipoprotein (HDL), mean Systolic Blood Pressure (SPB), and mean Hemoglobin A1c (HgbA1c) are used, and $\{b_i\}_{i=0}^7$ are pre-specified classification weights (Stevens et al., 2001).

While patient care has changed rapidly since this study was performed (the original patients were recruited and followed prospectively from 1977-1997), numerous studies have since explored its application in more recent clinical cohorts. These differences as well as regional variation in health care access and disease burden compelled us to estimate the UKPDS parameters in our cohort to improve its classification results for our patients.

3.3.2 OUTCOMES IDENTIFICATION

We generated training and test cohorts for our classification experiment in Section 4.3 by defining well-known T2DM disease morbidities with diagnosis and procedure codes (American Diabetes Association, 2014). For each patient we capture the date of the 13 outcomes in Table 1.

3.3.3 GENERATING TEST AND TRAIN COHORTS

To generate training and test cohorts from our dataset, we selected a reference date that allowed us to (i) capture a large patient population with multiple patient encounters prior to the date, and (ii) evaluate encounters after the date for the existence of an outcome.

For the classification experiment we generated count matrices for each data mode by aggregating patient data for a six month period prior to the *patient visit*. We then determined if the patient had one or more of the above outcomes within 1 year of that visit. For the training cohort, the *patient visit* was defined as the encounter immediately prior to the date threshold of January 1, 2010. For our test set, we used a January 1, 2011 threshold date. We cleaned our data to remove any patients (i) that already had outcomes 6 months prior to the *patient visit* and (ii) removed any codes with less than 10 observations over the entire cohort. Lastly, we removed from the test set any patients that were in the original test set and did not have any additional visits since the training set threshold date. We also removed any individuals who did not have laboratory or vitals data in the prior 2 years, preventing us from computing a UKPDS risk score.

4. Experiments

In this section we start by presenting benchmark results using well-known corpora for topic models, the goal being to show how DPPM (single modality) compares to related deep models. Additional results for the single-modality case can be found in Henao et al. (2015). Next, we present extensive experiments using the motivating data described in Section 3. In particular, we evaluate DPPM and DMPPM in terms of model fit and classification performance. Finally, we analyze the topics estimated by DMPPM. All experiments were conducted on a 2.2GHz desktop machine with 8GB RAM. The code used, implemented in Matlab, will be made publicly available.

4.1 Benchmark corpora

We first evaluate the performance of the basic version of our model, specifically the deep single modality model in (5). For this purpose, we present experiments on three corpora: 20 Newsgroups (20 News), Reuters corpus volume 1 (RCV1) and Wikipedia (Wiki). 20 News is composed of 18,845 documents and 2,000 words, partitioned into a 11,315 document training set and a 7,531 document test set. RCV1 has 804,414 newswire articles containing 10,000 words. A random 10,000 subset of documents is used for testing. For Wiki, we obtained 10⁷ random documents, from which a subset of 1,000 is set aside for testing. Following Hoffman et al. (2010), we keep a vocabulary consisting of 7,702 words taken from the top 10,000 words in the Project Gutenberg Library.

As performance measure, we use held-out perplexity, a commonly used performance metric for topic models defined as the geometric mean of the inverse marginal likelihood of every word in the set. We cannot evaluate the intractable marginal for the model in (5), thus we compute the *predictive perplexity* on a 20% subset of the held-out set. The remaining 80% is used to learn document-specific variables of the model, $\{\theta_n^{(\ell)}, \mathbf{h}_n^{(\ell)}\}$, for $n = 1, \dots, N$ and $\ell = 1, \dots, L$. The training set is used to estimate the global parameters of the model, $\{\Psi^{(\ell)}, r_k^{(\ell)}\}$, for $\ell = 2, \dots, L$ and $\lambda^{(L+1)}$. For PFA-based models, the test perplexity for a

Model	Method	Size	20 News	RCV1	Wiki
DPFM	MCMC	128-64	780	908	783
DPEA-SBN	SGNHT	1024-512-256	—	942	770
DPEA-SBN	SGNHT	128-64-32	827	1143	876
DPEA-RBM	SGNHT	128-64-32	896	920	942
nHDP	SVI	(10,10,5)	889	1041	932
LDA	Ghbbs	128	893	1179	1059
FTM	Ghbbs	128	887	1155	991
RSM	CD5	128	877	1171	1001

Table 2: Held-out perplexities for 20 News, RCV1 and Wiki. Size indicates number of topics and/or binary units, accordingly.

single modality can be calculated as (Zhou et al., 2012)

$$\text{perplexity} = \exp \left(-\frac{1}{x_{..}} \sum_{m=1}^M \sum_{n=1}^N x_{mn} \log \frac{\sum_{s=1}^S \sum_{k=1}^K \phi_{mk}^s \theta_{kn}^s h_{kn}^s}{\sum_{s=1}^S \sum_{m=1}^M \sum_{k=1}^K \phi_{mk}^s \theta_{kn}^s h_{kn}^s} \right),$$

where we have omitted modality and layer indices for clarity, S is the total number of collected samples, $x_{..} = \sum_{m=1}^M \sum_{n=1}^N x_{mn}$ and x_{mn} , ψ_{mk} , θ_{kn} and h_{kn} are elements of \mathbf{x}_n , Ψ , Θ , and \mathbf{h}_n , respectively.

We compare our single-modality deep model in (5) (denoted DPFM), against LDA (Blei et al., 2003), FTM (Williamson et al., 2010), RSM (Hinton and Salakhutdinov, 2009), nHDP (Paisley et al., 2015) and DPEA with SBNs (DPEA-SBN) and RBMs (DPEA-RBM) (Gan et al., 2015a). For all these models, we use the settings described in Gan et al. (2015a). Inference methods for RSM and DPEA are contrastive divergence with step size 5 (CD5) and stochastic gradient Nose-Hoover thermostats (SGNHT), respectively. For our model, (after the aforementioned pre-training) we run 3,000 samples, from which the first 2,000 are discarded (burnin). For the Wiki corpus, MCMC-based DPFM is run on a random subset of 10^6 documents.

Table 2 shows results for the corpora being considered. Figures for methods other than DPFM were taken from Gan et al. (2015a). We see that multilayer models (DPFM, DPFA and nHDP) consistently outperform single layer ones (LDA, FTM and RSM), and that DPFM has the best performance across all corpora for models of comparable size. We verified empirically (results not shown) that doubling the number of hidden units, adding a third layer or increasing the number of samples/iterations for DPFM does not significantly change the results in Table 2. As a note on computational complexity, one iteration of the two-layer model on the 20 News corpus takes approximately 2 seconds. For comparison, we also ran the DPEA-SBN model in Gan et al. (2015a) using a two-layer model of the same size; in their case it takes about 24, 4 and 5 seconds to run one iteration using MCMC, conditional density filtering (CDFE) and SGNHT, respectively. Runtimes for DPEA-RBM are similar to those of DPEA-SBN. Additional results for DPFM using stochastic variational inference can be found in Henao et al. (2015).

Size	Naive1		Naive2		DMPFM				
	Med	Lab	Med	Lab	Med	Lab			
64-32	1.930	76.724	210.690	1.930	76.575	208.785	1.865	72.919	194.260
96-48	1.851	76.736	192.851	1.825	76.787	193.782	1.788	72.662	176.737
128-64	1.803	76.538	182.803	1.759	76.495	182.049	1.748	72.415	167.423
64-32-16	1.918	76.648	207.932	1.911	76.400	209.652	1.861	72.773	191.854
96-48-24	1.822	76.967	192.530	1.816	76.660	192.505	1.759	72.531	176.451
128-64-32	1.787	76.556	182.365	1.764	76.528	180.806	1.730	72.364	166.759

Table 3: Held-out perplexity for EHR data. Size indicates number of topics and/or binary units, accordingly. Naive1 uses one DPFM per modality and Naive2 one DPFM for stacked modalities (Meds, Labs and Codes). Naive2 and DMPFM use all modalities at once but perplexities are computed separately.

4.2 Model fitting

We evaluate the ability of the multi-modality model in (6) to fit the data introduced in Section 3. The data consist of 16,756 patients, and of these 7,892 were used for model fitting and 8,864 for testing. We considered the three modalities discussed above: 1,694 of the entities corresponded to medications (Meds), 1,869 corresponded to laboratory tests (Labs), and 21,013 corresponded to diagnosis and procedure codes (Codes). We filtered out variables with less than 10 occurrences over the entire cohort, reducing the data to 253, 606 and 4,222 entities for Meds, Labs and Codes, respectively. We consider three different models: (i) A single-modality approach, in which we treat each modality independently using (5) (denoted Naive1); (ii) another single-modality approach, in which all modalities are stacked into one data matrix, then modeled using (5) (denoted Naive2); and (iii) the multi-modality approach using (6) (denoted DMPFM). Note that Naive1 and Naive2 constitute direct applications of the DPFM model introduced by Henao et al. (2015). In all cases we collect 1,200 samples after running 1,200 burnin iterations. As the performance measure, we report held-out perplexities for each modality on a randomly selected 20% subset (the test set).

Table 3 shows predictive perplexities for different architectures. We consider two- and three-layer specifications (Size) in three different binary unit sizes each, for a total of 6 models. We see that Naive2 and DMPFM consistently outperform Naive1. These results demonstrate that sharing information across modalities produces a model with a richer correlation structure and improved model fit, compared to Naive1 and Naive2 that use the DPFM of Henao et al. (2015). We also see that DMPFM performs the best in all configurations, which highlights the importance of modeling correlation structure within and across modalities. In terms of number of layers, we see a modest perplexity improvement going from two to three layers in all cases. This is probably due to the size of the dataset; it is likely that more significant gains will be observed from cohorts with a larger number of variables and patients. It is worth noting that since our model is sparse, size K (on each layer) can be understood as an upper bound on the number of factors in the sense that the model has the ability of *turning off* entire factors by letting all its activations, h_{kn} , to be zero. In this experiment we present results for increasing values of K to show that the model is able to capture increasing amounts of detail (evidenced by decreasing perplexity)

Size	Med	Naive1 Lab	Code	Naive2 All	DMPFM All
64-32	0.592±0.05	0.594±0.05	0.745±0.06	0.751±0.06	0.771±0.07
96-48	0.596±0.04	0.583±0.05	0.727±0.06	0.750±0.06	0.781±0.06
128-64	0.590±0.04	0.590±0.05	0.725±0.06	0.751±0.06	0.779±0.06
64-32-16	0.601±0.05	0.594±0.05	0.726±0.05	0.742±0.06	0.771±0.06
96-48-24	0.587±0.04	0.588±0.06	0.735±0.06	0.758±0.07	0.785±0.07
128-64-32	0.590±0.04	0.588±0.05	0.732±0.05	0.757±0.06	0.784±0.07

Table 4: Mean test AUCs with standard deviations over 13 binary classification tasks. Size indicates number of topics and/or binary units, accordingly. Naive1 is one discriminative DPMFM per modality and Naive2 is one discriminative DPMFM for stacked modalities (Meds, Labs and Codes). Naive2 and DMPFM use all modalities to build classifiers.

as the model size grows, but also as a way to highlight that the model does not overfit, meaning that test performance (perplexity) does not deteriorate as the model size increases.

In terms of computational complexity, Naive1 and Naive2 take between 180 and 310 CPU depending on the size of the model: DMPFM takes between 190 and 480 minutes. Note that runtime include model fit, testing and perplexity calculations. In any case, runtimes are deemed reasonable considering the size of the dataset and the complexity of the models being evaluated.

4.3 Multi-label classification

We evaluate the discriminative DMPFM in Section 2.5 on the multi-label classification problem outlined in Section 3. We consider 13 well-known T2DM-related outcomes in Table 1, namely: Acute Myocardial Infarction (AMI), amputation, cardiac catheterization, coronary artery disease, depression, heart failure, chronic kidney disease, neurological disease, obesity, ophthalmic disease, stroke, unstable angina and death. We compare our discriminative DMPFM to discriminative versions of Naive1 and Naive2 based on DPMFMs. For a baseline comparison, we use the UKPDS model in (12) and sparse logistic regression (Friedman et al., 2001). For the UKPDS model we estimate model coefficients, $\{b_j\}_{j=0}^L$, for each outcome independently in a logistic regression setting. Note that UKPDS was originally intended for coronary heart disease, however we use its covariates (age, sex, race, smoking status, HgPA1c, SPB, TC and HDL) to build classifiers for all outcomes. We also use coronary heart disease interchangeably with coronary artery disease, which is the build up of plaque in the arteries of the heart and results in coronary heart disease. For PFA-based models, we collect 1,200 samples after running 1,200 burnin iterations. As a performance measure, we report area under the receiving operating characteristic (AUC) values on the test set (Fawcett, 2006). Provided that all classification tasks are very imbalanced, about 8% positive outcomes in average, we do not report test accuracies. Optimal thresholds can be obtained from ROC curves using outcome-specific prevalence information, if desired. Once threshold values have been selected, accuracies, true positive rates and true negative rates can be readily computed.

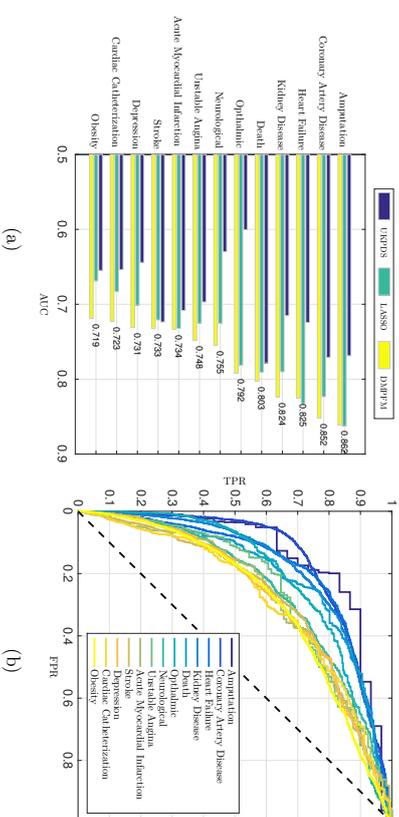


Figure 2: Test AUC and ROC curves from multi-label classification experiment. (a) AUC values for UKPDS, sparse logistic regression (LASSO) and DMPFM. Values beside each bar group correspond to AUC values obtained by DMPFM. (b) ROC curves from DMPFM. Each curve represents represents a classification task for an outcome. The dashed line is the AUC of a random classifier.

Table 4 shows average test AUCs for Naive1, Naive2 and DMPFM and different model architectures. Averages are computed over the 13 outcomes in Table 1. We see from the results for Naive1 that Codes carry significantly more classification power than Med and Lab modalities. Naive2, which combines all modalities into a single data matrix performs consistently better than Naive1, and DMPFM performs the best by a considerable margin, taking into account the size of the test set. In terms of model size, the largest three-layer model performs the best, closely followed by models of size 96-48-24 and 96-48.

Results in Figure 2a show test AUC values as bars for each outcome individually. We compare DMPFM against two baselines, UKPDS and sparse logistic regression. We see that DPMFM outperforms the others in nearly all classification tasks except for heart failure, in which sparse logistic regression (LASSO) performs best, and amputation, where DMPFM and LASSO perform about the same. Note that LASSO is considerably sparser than our model, because it tends to exclude heavily correlated variables, however we observed that our model tends to include the same variables deemed important by LASSO, as a subset. In Figure 2a we also show test AUC values obtained by DMPFM sorted in decreasing order, with corresponding ROC curves in Figure 2b.

The outcomes with the greatest predictive power was amputation and the lowest was obesity. Upon further examination these have potentially interesting clinical drivers. For example amputation of limbs in diabetic patients is often the result of longstanding neuropathy and microvascular damage hindering the ability of patients to not only identify injuries but also heal. We note that the second and fourth authors are medical doctors, and provided all medical analysis. A common clinical scenario involves patients with foot

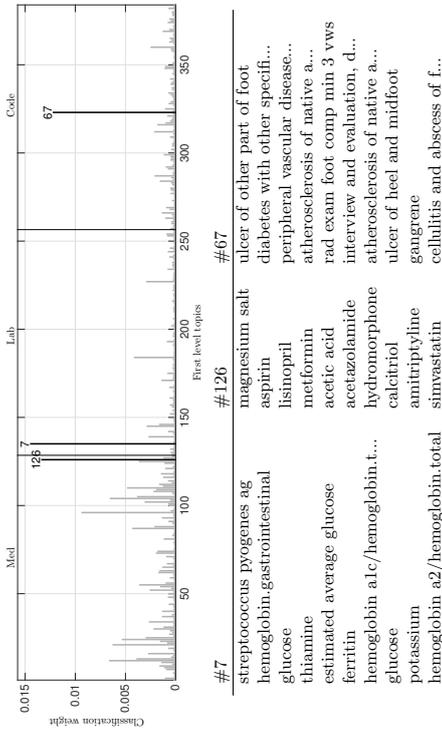


Figure 3: Top classification weights and topics associated with amputation. We show the top 10 words (bottom panel) from first-layer topics with the largest 3 classification weights (top panel), namely meds # 126, labs #7 and codes #67.

ulcerations that go undetected and result in gangrenous limbs and ultimately amputation. By plotting the classification coefficients for amputation (see Figure 3:top), we identify the top three contributors to this outcome (medications #126, labs #7, and codes #67) in Figure 3:bottom. In medications topic #126, we find the usage of standard diabetes, cholesterol and hypertension medications. Notably we also found Amitriptyline, which can be used to treat patients with diabetic neuropathy. In Labs #7, we identify laboratory tests that would be common in evaluating neuropathy (thiamine deficiency) and a test for a bacterial species common in skin infections, *Streptococcus Pyogenes* Antigen. This would be common antigen for a patient with skin infections including foot ulcerations. Lastly, the codes #67 refer to foot ulcers and peripheral vascular disease. Peripheral vascular disease can result from the accumulation of fatty deposits in the vasculature of the extremities and can be exacerbated by the microvascular damage of diabetes. While additional evaluation of topics with high classification coefficients may elicit unexpected predictors of amputation, this initial analysis revealed that the highest scoring topics correlated well with clinical intuition.

The poor predictive power for obesity likely rests with its prevalence in T2DM populations. Metabolic syndrome, a constellation of symptoms including hyperlipidemia, hypertension and obesity is a strong risk factor for obesity. An examination of contributing first-layer topics reveals medications that would be typical for a patient with symptoms of metabolic syndrome. The main lab first-layer topic shares the same topic #7 as in amputation. Interestingly, morbidly obese patients also share a high incidence of skin infections

#3	#13	#19
acid medication	digoxin	clididine
duloxetine	belladonna alkaloids	simvastatin
metformin	albuterol	valproate
budesonide	duloxetine	colchicine
fluphenazine	acetatolol	fluphenazine
dexamethasone	pseudoephedrine	hydralazine
insulin lispro	azithromycin	omeprazole
atazanavir	cyclosporine	bromfenac
azithromycin	alprostadil	meloxicam
glimperide	acai berry extract	acid medication

Table 5: Selected topics from medications modality. We show the top 10 words from first-layer topics #3, #13 and #19.

and pressure induced ulcerations due to their sedentary behaviors. Lastly, the code topics have codes related to abnormal weight gain.

4.4 Analysis of multi-modality model

We also examined the ability of the DPFA model to generate topics that represent intuitive medical concepts. For illustrative purposes, we discuss the intra-modality correlation of first and second level topics (meta-topics), starting with the medications mode and expand to other modalities. We plot the correlations between medication topics in Figure 4. We show first-layer topics (boxes) within a modality. Each box contains the first four words with the highest probability mass in that topic. The topics are connected into meta-topics (blue circles) representing both intra- and inter-modality correlations. As we see, the graph is considerably sparse taking into account that each meta-topic can have up to 128 edges. As a quantitative summary of graph sparsity, we note that the average node degree of the complete⁷ medication graph is 18 (edges in the graph are present if their weight is larger than 1e-2).

The center of the plot in Figure 4 includes topics (based on lowest layer in the model, touching the data) and meta-topics (based on the second layer in the model) with the highest level of correlation with other topics. Unsurprisingly, the central medicine topic (#3, see Table 5) includes the medication metformin, a first-line treatment for T2DM, and duloxetine, a treatment for peripheral neuropathy. Interestingly, this topic also includes several steroids including dexamethasone and budesonide, which can induce or worsen T2DM.

To better interpret the correlation structure, we started with meta-topics and explored both intra- and inter-modality correlation. To ease interpretability we focused on meta-topics with fewer connections. We found that many of the meta-topics represented coherent clinical narratives based on the discovered first-layer topics and meta-topics. For example the meta-topic #29 in Figure 5, includes first-layer medicine topics #19 #13 in Table 5. These two collections represent a wide variety of medications used to treat co-morbidities common to T2DM. However, this list also includes opioid pain killers and a chemotherapeutic agent, cyclosporine. While difficult to interpret with only intra-modality correlation,

7. The graph in Figure 4 only shows top four connections for clarity.

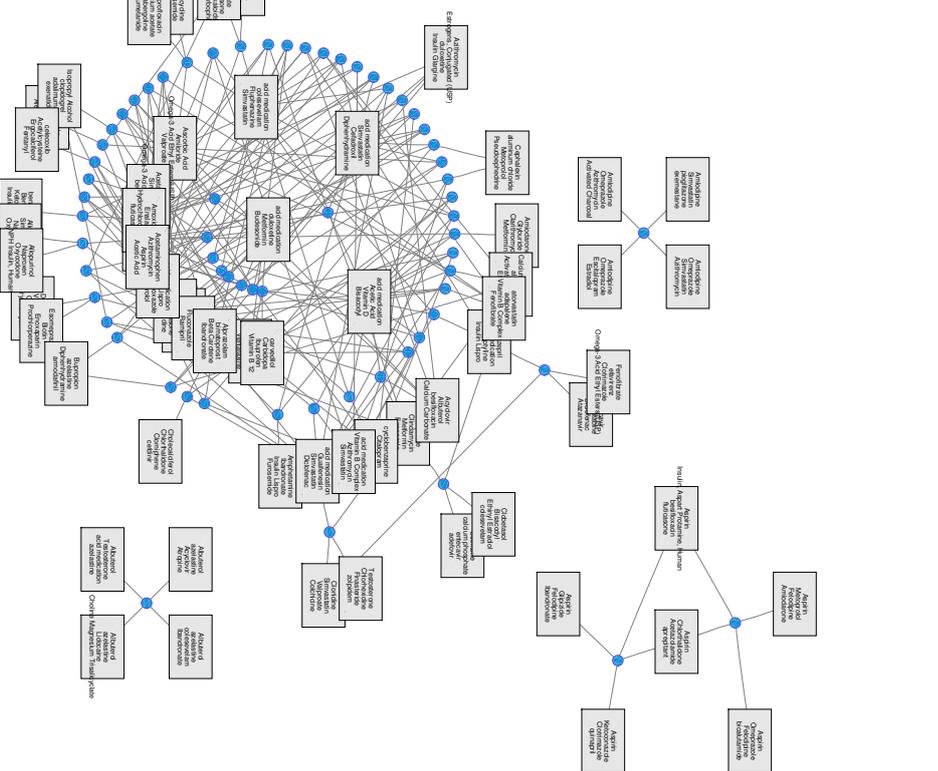


Figure 4: Graph representation obtained for medication interactions. Meta-topic are denoted by blue circles and first-layer topics as boxes, with word lists corresponding to the top four medications in first-layer topics, $\psi_k^{(1)}$. For clarity, we only show the top four connections between meta-topics and their associated topics.

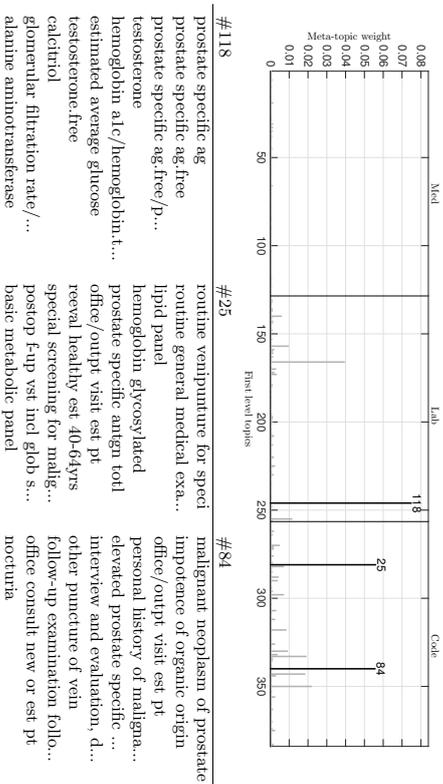


Figure 5: Top weights and topics associated with meta-topic #29. We show the top 10 words (bottom panel) from first-layer topics with the largest 3 meta-topic weights (top panel), namely labs #118 and codes #25 and #84.

further examination of first-layer topics across modalities that contribute disproportionately to this meta-topic identified laboratory and diagnostics codes which expand the narrative of this meta-topic. A patient weighted heavily with this meta-topic would have laboratory results characterized by hematuria (blood in urine) and prostate specific antigen testing. The combination of these medication and laboratory topics suggests a patient with metastatic prostate cancer causing pain and hematuria. This is confirmed when examining the first-layer code topic #84, which includes the code for malignant prostate cancer.

In another example, we explore meta-topic #3 in Figure 6, a topic that does not necessarily make intuitive sense, but could hint at the power of DPFA to identify novel correlations between different data. This meta-topic has two prominent first-layer medication topics. While the first medication topic, #33, contains a mix of hypertensive and antiviral medications, the second topic, #120, includes two notable drugs alprazolam (Xanax) and baclofen, a muscle relaxant. While these two medications may relate to the anxiety, myalgia and insomnia codes, we see in first-layer topic #19 for codes, it would be interesting to explore other first-layer topics contribute to this meta-topic and connect with other conditions identified such as major depressive disorder and chronic pain.

5. Discussion

In 2012, the American Diabetes Association estimated that the economic burden of diabetes in the United States exceeded 245 billion dollars⁸. High-throughput and widely available

8. See <http://diabetes.org/advocacy/news-events/cost-of-diabetes.html>.

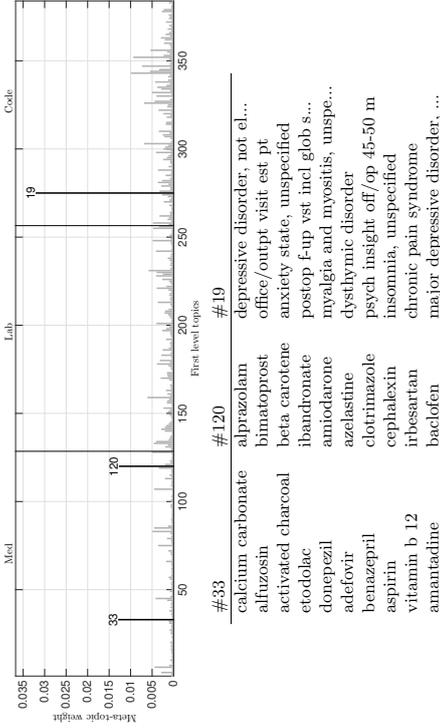


Figure 6: Top weights and topics associated with meta-topic #3. We show the top 10 words (bottom panel) from first-layer topics with the largest 3 meta-topic weights (top panel), namely labs #19 and codes #33 and #120.

methods to predict morbidity and mortality outcomes for patients would improve the deployment of medical resources, and may reduce costs through increased preventative care or reduced futile care.

Our initial evaluation of the proposed method illustrated that DMPFM can identify multiple candidate phenotypes from EHR data without expert or user supervision. Further, these candidate topics, when defined in the context of a classification task, can significantly outperform current benchmarks for risk prediction derived from large-scale clinical studies. This was perhaps unsurprising as we are able to utilize much richer datasets (in both number of clinical variables and patient numbers) than most clinical studies. We are also able to estimate the important factors directly from the data, thus minimizing bias on the behalf of the original study designers.

Despite these encouraging findings, many challenges remain before such high-throughput phenotyping efforts can be used in a clinical setting. First, the clinical evaluation of DMPFM for EHR relied on two clinical experts (second and fourth authors) to perform the data reconciliation and evaluate the topic groupings. This evaluation is subject to bias, and a more extensive study involving a panel of clinicians is necessary to validate this method's robustness. Second, as discussed above, some of the clinical phenotypes are not easily interpreted. Although our method encourages sparsity there remains a high level of correlation both within and between topics. These topics have many interconnections that remain to be fully explored. In addition, the appropriate metrics to evaluate words (entities), topics and meta-topics for clinical applications requires further research. Third, even in cases where we can generate a narrative around candidate phenotypes, many topics still contained

words/entities from each modality that appeared irrelevant to the larger meta-topic. Additional research is needed to establish the clinical relevance of these words/entities. Fourth, although we present a generative process for DMPFM, we did not perform experiments exploring the generation of topic weights for new patients. It would also be interesting to explore the number of patients that would be required to generate a fully comprehensive but sparse set of topics for any given patient population. Lastly, although it appears that we can generate meta-topics that represent patients, we did not perform case review of these patients that review those topics/meta-topics. It would be necessary to perform off-line review by physicians to establish clinical correlation between computational phenotypes and the true patient status.

The DMPFM is an extensible model applicable to any data modalities that can be represented with count data. This would naturally extend to free-text physician and nursing notes (in this case the counts are of actual words) as well as notes from specialty services, such as radiology and pathology. With the paucity of additional data that is contained with the medical record, we are confident that we can develop improved representations of patient traits that may lead to better diagnosis and treatment outcomes.

Acknowledgments

We would like to acknowledge support for this project from the Information Initiative at Duke, Duke Clinical Research Institute and Duke Medicine.

Appendix A. Stochastic variational inference

SVI is a scalable algorithm for approximating posterior distributions consisting of EM-style local-global updates, in which subsets of a dataset (*mini-batches*) are used to update in closed-form the variational parameters controlling both the local and global structure of the model in an iterative fashion Hoffman et al. (2013). This is done by using stochastic optimization with noisy natural gradients to optimize the variational objective function. Additional details and theoretical foundations of SVI can be found in Hoffman et al. (2013).

In practice the algorithm proceeds as follows, where again we have omitted the layer index for clarity: (i) let $\{\Psi^{(t)}, r_k^{(t)}, \lambda^{(t)}\}$ be the global variables at iteration t . (ii) Sample a mini-batch from the full dataset. (iii) Compute updates for the variational parameters of the local variables using (layer index omitted for clarity)

$$\begin{aligned} \theta_{mkn} &\propto \exp(\mathbb{E}[\log \psi_{mk}] + \mathbb{E}[\log \theta_{kn}]), \\ \theta_k &\sim \text{Gamma} \left(\mathbb{E}[r_k] \mathbb{E}[h_{kn}] + \sum_{m=1}^M \phi_{mkn}, b^{-1} \right), \\ h_{kn} &\sim \mathbb{E}[p(x_{:kn} = 0)] \text{Bernoulli}(\mathbb{E}[\tilde{\pi}_{kn}] \mathbb{E}[\tilde{\pi}_{kn}]^{-1}) + \mathbb{E}[p(x_{:kn} \geq 1)] \\ r_k &\sim \text{Gamma} \left(1 + \sum_n \mathbb{E}[u_{kn}], 1 - \sum_n \mathbb{E}[p(h_{kn} = 1)] \log(1 - b) \right), \\ z_{kn} &\sim \mathbb{E}[p(h_{kn} = 1)] \text{Poisson} + \left(\tilde{\lambda}_{kn} \right), \end{aligned}$$

where

$$\mathbb{E}[x_{mkn}] = \phi_{mkn}, \quad \mathbb{E}[\tilde{\pi}_{kn}] = \mathbb{E}[\pi_{kn}](1 - b_n)^{\mathbb{E}[r_k]}, \quad \mathbb{E}[u_{kn}] = \sum_{j=1}^{x_{kn}} \mathbb{E}[r_k] (\mathbb{E}[r_k] + j - 1)^{-1}.$$

In practice, expectations for θ_{kn} and h_{kn} are computed in log-domain. (iv) Compute a local update for the variational parameters of the global variables (only Ψ is shown) using

$$\hat{\psi}_{mk} = \eta + \frac{N}{NB} \sum_{n=1}^{N_B} x_{mn} \phi_{mkn}, \quad (13)$$

where N and N_B are sizes of the corpus and mini-batch, respectively. Finally, we update the global variables as $\psi_k^{(t+1)} = (1 - \rho_l) \psi_k^{(t)} + \rho_l \hat{\psi}_k$, where $\rho_l = (t + \tau)^{-\kappa}$. The forgetting rate, $\kappa \in (0.5, 1]$ controls how fast previous information is forgotten and the delay, $\tau \geq 0$, down-weights early iterations. These conditions for κ and τ guarantee that the iterative algorithm converges to a local optimum of the variational objective function. In the experiments, we set $\kappa = 0.7$ and $\tau = 128$.

Appendix B. Inference details

Conditional posteriors for Gibbs sampling (layer index omitted for clarity):

$$\begin{aligned} \psi_k &\sim \text{Dirichlet}(\eta + x_{1k}, \dots, \eta + x_{Mk}), \\ \theta_{kn} &\sim \text{Gamma}(r_k h_{kn} + x_{kn}, b^{-1}), \\ h_{kn} &\sim \delta(x_{kn} = 0) \text{Bernoulli}(\tilde{\pi}_{kn}(\tilde{\pi}_{kn} + 1 - \pi_{kn})^{-1}) + \delta(x_{kn} \geq 1), \\ r_k &\sim \text{Gamma}\left(1 + \sum_n u_{kn}, 1 - \sum_n h_{kn} \log(1 - b)\right), \\ z_{kn} &\sim \delta(h_{kn} = 1) \text{Poisson}_+(\tilde{\lambda}_{kn}), \end{aligned}$$

where $\text{Poisson}_+(\cdot)$ is the zero-truncated Poisson distribution and

$$\begin{aligned} x_{mk} &= \sum_{n=1}^N x_{mkn}, \\ x_{kn} &= \sum_{m=1}^M x_{mkn}, \\ \tilde{\pi}_{kn} &= \pi_{kn} (1 - b)^{r_k}, \\ u_{kn} &= \sum_{j=1}^{x_{kn}} u_{knj}, \quad u_{knj} \sim \text{Bernoulli}\left(\frac{r_k}{r_k + j - 1}\right). \end{aligned} \quad (14)$$

Note that for multilayer models, $\pi_{kn}^{(l)} = 1 - \exp(\lambda_{kn}^{(l+1)})$. The data augmentation scheme for r_k via u_{kn} is described in Zhou and Carin (2015).

For the discriminative DPFM, lets denote latent counts for \hat{y}_n as \hat{x}_{ckn} , with summaries analogous to (14), as \hat{x}_{ck} and \hat{x}_{kn} . Then,

$$\begin{aligned} \mathbf{b}_k &\sim \text{Dirichlet}(\zeta + \hat{x}_{1k}, \dots, \zeta + \hat{x}_{Ck}), \\ \theta_{kn} &\sim \text{Gamma}(r_k h_{kn} + x_{kn} + \hat{x}_{cnk}, b^{-1}), \\ h_{kn} &\sim \delta(x_{kn} = 0 \wedge \hat{x}_{cnk} = 0) \text{Bernoulli}(\tilde{\pi}_{kn}(\tilde{\pi}_{kn} + 1 - \pi_{kn})^{-1}) + \delta(x_{kn} \geq 1 \vee \hat{x}_{cnk} \geq 1). \end{aligned}$$

Provided that θ_n and \mathbf{b}_n are shared by two PFA modules, one for the count data, \mathbf{x}_n , and the other for the labels, \hat{y}_n , their conditional posteriors are functions of latent counts coming from both sources, x_{kn} and \hat{x}_{cnk} , respectively.

References

- American Diabetes Association. Diagnosis and classification of diabetes mellitus. *Diabetes Care*, 37(1):81–90, 2014.
- David M. Blei and John D. Lafferty. A correlated topic model of science. *Annals of Applied Statistics*, 1(1):17–35, 2007.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(993–1022), 2003.
- David M. Blei, Thomas L. Griffiths, Michael I. Jordan, and Joshua B. Tenenbaum. Hierarchical topic models and the nested Chinese restaurant process. In *Advances in Neural Information Processing Systems*, 2004.
- Carlos M. Carvalho, Jeffrey Chang, Joseph E. Lucas, Joseph R. Nevins, Qiantli Wang, and Mike West. High-dimensional sparse factor modeling: applications in gene expression genomics. *Journal of the American Statistical Association*, 103(484):1438–1456, 2008.
- Tiangqi Chen, Emily B. Fox, and Carlos Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In *International Conference on Machine Learning*, 2014.
- Yukun Chen, Robert J. Carroll, Eugenia R. McPeck Hinz, Anushi Shah, Anne E. Elyer, Joshua C. Denny, and Hua Xu. Applying active learning to high-throughput phenotyping algorithms for electronic health records data. *Journal of the American Medical Association*, 20(e2):e253–9, 2013.
- P. M. Clarke, A. M. Gray, A. Briggs, A. J. Farmer, P. Fenn, R. J. Stevenson, D. R. Matthews, I. M. Stratton, and R. R. Holman. A model to estimate the lifetime health outcomes of patients with type 2 diabetes: the United Kingdom Prospective Diabetes Study (UKPDS) outcomes model (UKPDS no. 68). *Diabetologia*, 47(10):1747–1759, 2004.
- David Collett. *Modelling binary data*. CRC Press, 2002.
- Nan Ding, Youhan Fang, Ryan Babush, Changyuan Chen, Robert D. Szeel, and Hartmut Neven. Bayesian sampling using stochastic gradient thermostats. In *Advances in Neural Information Processing Systems*, 2014.

- Michael D. Escobar and Mike West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.
- Tom Fawcett. An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874, 2006.
- Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Springer series in statistics Springer, Berlin, 2001.
- Zhe Gan, Changyou Chen, Ricardo Henao, David Carlon, and Lawrence Carin. Scalable deep Poisson factor analysis for topic modeling. In *International Conference on Machine Learning*, 2015a.
- Zhe Gan, Ricardo Henao, David Carlon, and Lawrence Carin. Learning deep sigmoid belief networks with data augmentation. In *International Conference on Artificial Intelligence and Statistics*, 2015b.
- Rajarsi Guhaniyogi, Shaan Qamar, and David B. Dunson. Bayesian conditional density filtering. *arXiv:1401.3632*, 2014.
- Ricardo Henao and Ole Winther. Sparse linear identifiable multivariate modeling. *Journal of Machine Learning Research*, 12:863–905, 2011.
- Ricardo Henao, Xin Yuan, and Lawrence Carin. Bayesian nonlinear support vector machines and discriminative factor modeling. In *Advances in Neural Information Processing Systems*, 2014.
- Ricardo Henao, Zhe Gan, James Lu, and Lawrence Carin. Deep Poisson factor modeling. In *Advances in Neural Information Processing Systems*, 2015.
- Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–800, 2002.
- Geoffrey E. Hinton and Ruslan R. Salakhutdinov. Replicated softmax: an undirected topic model. In *Advances in Neural Information Processing Systems*, 2009.
- Joyce C. Ho, Joydeep Ghosh, Steve R. Steinhilb, Walter F. Stewart, Joshua C. Denny, Bradley A. Malin, and Jimeng Sun. Limestone: high-throughput candidate phenotype generation via tensor factorization. *Journal of biomedical informatics*, 52:199–211, 2014a.
- Joyce C. Ho, Joydeep Ghosh, and Jimeng Sun. Marble: High-throughput phenotyping from electronic health records via sparse nonnegative tensor factorization. In *International Conference on Knowledge Discovery and Data Mining*, 2014b.
- Qirong Ho, James Cipar, Henggang Cui, Seungchak Lee, Jin Kyu Kim, Phillip B. Gibbons, Garth A. Gibson, Greg Ganger, and Eric P. Xing. More effective distributed ML via a stale synchronous parallel parameter server. In *Advances in Neural Information Processing Systems*, 2013.
- Matthew Hoffman, Francis R. Bach, and David M. Blei. Online learning for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems*, 2010.
- Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 2013.
- George Hripacsak and David J. Albers. Next-generation phenotyping of electronic health records. *Journal of the American Medical Informatics Association*, 20(1):117–21, 2013.
- P. King, I. Peacock, and R. Donnelly. The UK prospective diabetes study (UKPDS): clinical and therapeutic implications for type 2 diabetes. *British journal of clinical pharmacology*, 48(5):643–8, 1999.
- Simon Lacoste-Julien, Fei Sha, and Michael I Jordan. DiscLDA: Discriminative learning for dimensionality reduction and classification. In *Advances in Neural Information Processing Systems*, 2009.
- Hugo Larochelle and Stanislas Lauly. A neural autoregressive topic model. In *Advances in Neural Information Processing Systems*, 2012.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- Mu Li, David G. Andersen, Alex J. Smola, and Kai Yu. Communication efficient distributed machine learning with the parameter server. In *Advances in Neural Information Processing Systems*, 2014.
- Shou-En Lu, Gloria L. Beckles, Jesse C. Crosson, Dorian Bilik, Andrew J. Karter, Robert B. Gerzoff, Yong Lin, Sonja V. Ross, Laura N. McEwen, Beth E. Waitzfelder, David Marrero, Norman Lasser, and Arleen F. Brown. Evaluation of risk equations for prediction of short-term coronary heart disease events in patients with long-standing type 2 diabetes: the Translating Research Into Action for Diabetes (TRIAD) study. *BMC Endocrine Disorders*, 12(12):1–10, 2012.
- Lars Maaloe, Morten Armgren, and Ole Winther. Deep belief nets for topic modeling. *arXiv:1501.04325*, 2015.
- Ravi K. Mareedu, Falgun M. Modhia, Elenita I. Kaurin, James G. Linneman, Terrie Kitchner, Catherine A. McCarty, Ronald M. Krauss, and Russell A. Wilke. Use of an electronic medical record to characterize cases of intermediate statin-induced muscle toxicity. *Preventive cardiology*, 12(2):88–94, 2009.
- Jon D. McAuliffe and David M. Blei. Supervised topic models. In *Advances in Neural Information Processing Systems*, 2008.
- Patricia A. Metcalf, Susan Wells, Robert K. R. Scragg, and Rod Jackson. Comparison of three different methods of assessing cardiovascular disease risk in New Zealanders with type 2 diabetes mellitus. *The New Zealand medical journal*, 121(1281):49–57, 2008.
- Radford M. Neal. Connectionist learning of belief networks. *Artificial Intelligence*, 56(1):71–113, 1992.

- Stuart J. Nelson, Kelly Zeng, John Kilbourne, Tammy Powell, and Robin Moore. Normalized names for clinical drugs: RxNorm at 6 years. *Journal of the American Medical Informatics Association*, 18(4):441–8, 2011.
- Katherine M. Newton, Peggy L. Peissig, Abel Ngo Kho, Suzette J. Bielinski, Richard L. Berg, Vidhu Choudhary, Melissa Bastford, Christopher G. Chute, Itzhikhar J. Kullo, Rongling Li, Jennifer A. Pacheco, Luke V. Rasmussen, Leslie Spangler, and Joshua C. Denny. Validation of electronic medical record-based phenotyping algorithms: results and lessons learned from the eMERGE network. *Journal of the American Medical Informatics Association*, 20(e1):e147–54, 2013.
- John Paisley, Chong Wang, David M. Blei, and Michael I. Jordan. Nested hierarchical Dirichlet processes. *Pattern Analysis and Machine Intelligence*, 37(2):256–270, 2015.
- Walter W. Piegorsch. Complementary log regression for generalized linear models. *The American Statistician*, 46(2):94–99, 1992.
- Rajesh Ranganath, Linpeng Tang, Laurent Charlin, and David M. Blei. Deep exponential families. In *International Conference on Artificial Intelligence and Statistics*, 2014.
- Rachel L. Richesson, Shelley A. Rusincovitch, Douglas Wixted, Bryan C. Batch, Mark N. Feiglos, Marie Lynn Miranda, W. Ed Hammond, Robert M. Caffi, and Susan E. Spratt. A comparison of phenotype definitions for diabetes mellitus. *Journal of the American Medical Informatics Association*, 20(e2):e319–26, 2013.
- Rebecca K. Simmons, Ruth L. Coleman, Hermione C. Price, Rury R. Holman, Kay T. Khaw, Nicholas J. Wareham, and Simon J. Griffin. Performance of the UK prospective diabetes study risk engine and the Framingham risk equations in estimating cardiovascular disease in the EPIC-Norfolk cohort. *Diabetes Care*, 32(4):708–13, 2009.
- Kihyuk Sohn, Wenling Shang, and Honglak Lee. Improved multimodal deep learning with variation of information. In *Advances in Neural Information Processing Systems*, pages 2141–2149, 2014.
- Nitish Srivastava and Ruslan Salakhutdinov. Multimodal learning with deep Boltzmann machines. In *Advances in Neural Information Processing Systems*, 2012.
- Nitish Srivastava and Ruslan Salakhutdinov. Multimodal learning with deep Boltzmann machines. *Journal of Machine Learning Research*, 15:2949–2980, 2014.
- Nitish Srivastava, Ruslan Salakhutdinov, and Geoffrey E. Hinton. Modeling documents with deep Boltzmann machines. In *Uncertainty in Artificial Intelligence*, 2013.
- R. J. Stevens, V. Kothari, A. I. Adler, I. M. Stratton, and United Kingdom Prospective Diabetes Study (UKPDS) Group. The UKPDS risk engine: a model for the risk of coronary heart disease in type II diabetes (UKPDS 56). *Clinical science (London)*, 101(6):671–9, 2001.
- Libo Tao, Edward C. F. Wilson, Simon J. Griffin, Rebecca K. Simmons, and ADDITION-Europe study team. Performance of the UKPDS outcomes model for prediction of myocardial infarction and stroke in the ADDITION-Europe trial cohort. *Value Health*, 16(6):1074–80, 2013.
- Yee W. Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.
- S. van Dieren, L. M. Peelen, U. Nöthlings, Y. T. van der Schouw, G. E. Rutten, A. M. Spijkerman, D. L. van der A, D. Sluik, H. Boeing, K. G. Moons, and J. W. Beulens. External validation of the UK prospective diabetes study (UKPDS) risk engine in patients with type 2 diabetes. *Diabetologia*, 54(2):264–70, 2011.
- Daniel J. Vreeman, John Hook, and Brian E. Dixon. Learning from the crowd while mapping to LOINC. *Journal of the American Medical Informatics Association*, 2015.
- Max Welling and Yee W. Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *International Conference on Machine Learning*, 2011.
- Sinead Williamson, Chong Wang, Katherine Heller, and David Blei. The IBP compound Dirichlet process and its application to focused topic modeling. In *International Conference on Machine Learning*, 2010.
- P. W. Wilson, R. B. D’Agostino, D. Levy, A. M. Belanger, H. Silbershatz, and W. B. Kannel. Prediction of coronary heart disease using risk factor categories. *Circulation*, 97(18):1837–47, May 1998.
- Xin Yuan, Ricardo Henao, Ephraim L. Tsallik, Raymond Langley, and Lawrence Carin. Non-Gaussian discriminative factor models via the max-margin rank-likelihood. In *International Conference on Machine Learning*, 2015.
- Yin Zheng, Yu J. Zhang, and Hugo Larochelle. Topic modeling of multimodal data: an autoregressive approach. In *Computer Vision and Pattern Recognition*, 2014.
- Mingyuan Zhou. Infinite edge partition models for overlapping community detection and link prediction. In *International Conference on Artificial Intelligence and Statistics*, 2015.
- Mingyuan Zhou and Lawrence Carin. Negative binomial process count and mixture modeling. *Pattern Analysis and Machine Intelligence*, 37(2):307–320, 2015.
- Mingyuan Zhou, Lauren Hannah, David Dunson, and Lawrence Carin. Beta-negative binomial process and Poisson factor analysis. In *International Conference on Artificial Intelligence and Statistics*, 2012.

Low-Rank Doubly Stochastic Matrix Decomposition for Cluster Analysis

Zhirong Yang

*Helsinki Institute of Information Technology HIIT
University of Helsinki, Finland*

ZHIRONG.YANG@HEL.SINKI.FI

Jukka Corander

*Department of Mathematics and Statistics
Helsinki Institute for Information Technology HIIT
University of Helsinki, Finland*

JUKKA.CORANDER@HEL.SINKI.FI

Department of Biostatistics, University of Oslo, Norway

Erkki Oja

*Department of Computer Science
Aalto University, Finland*

ERKKI.OJA@AALTO.FI

Editor: Maya Gupta

Abstract

Cluster analysis by nonnegative low-rank approximations has experienced a remarkable progress in the past decade. However, the majority of such approximation approaches are still restricted to nonnegative matrix factorization (NMF) and suffer from the following two drawbacks: 1) they are unable to produce balanced partitions for large-scale manifold data which are common in real-world clustering tasks; 2) most existing NMF-type clustering methods cannot automatically determine the number of clusters. We propose a new low-rank learning method to address these two problems, which is beyond matrix factorization. Our method approximately decomposes a sparse input similarity in a normalized way and its objective can be used to learn both cluster assignments and the number of clusters. For efficient optimization, we use a relaxed formulation based on Data-Cluster-Data random walk, which is also shown to be equivalent to low-rank factorization of the doubly-stochastically normalized cluster incidence matrix. The probabilistic cluster assignments can thus be learned with a multiplicative majorization-minimization algorithm. Experimental results show that the new method is more accurate both in terms of clustering large-scale manifold data sets and of selecting the number of clusters.

Keywords: cluster analysis, probabilistic relaxation, doubly stochastic matrix, manifold, multiplicative updates

1. Introduction

Cluster analysis is a main task of exploratory data mining, and a common technique for statistical data analysis used in many fields. Usually, optimization of the clustering objectives is NP-hard and relaxation to “soft” clustering is a widely used technique. In the past decade, various low-rank matrix approximation objectives, together with a nonnegativity constraint on the cluster indicator matrix, have widely been used for the relaxation purpose.

The most popular nonnegative low-rank approximation method is *Nonnegative Matrix Factorization* (NMF; Lee and Seung, 1999, 2001). It finds a matrix which approximates the pairwise similarities between the data items and can be factorized into two nonnegative low-rank matrices. NMF was originally applied to vector data, for which Ding et al. (2010) did later show that it is equivalent to the classical k -means method. NMF has also been applied to weighted graph defined by the pairwise similarities of the data items. For example, Ding et al. (2008) presented Nonnegative Spectral Cuts by using a multiplicative algorithm; and Arora et al. (2011, 2013) proposed Left Stochastic Decomposition that approximates a similarity matrix based on Euclidean distance and a left-stochastic matrix. Topic modeling represents another example of a related factorization problem. Hofmann (1999) introduced a generative model in *Probabilistic Latent Semantic Indexing* (PLSI) for counting data, which is essentially equivalent to NMF using Kullback-Leibler (KL) divergence and tri-factorizations. Bayesian treatment of PLSI by using Dirichlet priors was later introduced by Blei et al. (2001). Symmetric PLSI with the same Bayesian treatment is called *Interaction Component Model* (ICM; Sinkkonen et al., 2008).

Despite the remarkable progress, the above NMF-type clustering methods still suffer from one or more of the following problems: 1) they are not accurate for the data in curved manifolds; 2) they cannot guarantee balanced partitions; 3) their learning objectives cannot be used to choose the number of clusters. The first problem is common in many real-world clustering tasks, where data are represented by raw features and simple similarity metrics such as Euclidean or Hamming are only accurate in small neighborhoods. This induces sparse accurate similarity information (e.g. as with the K -Nearest-Neighbor for a relatively small K) as input for clustering algorithms. In the big data era, the sparsity is also necessary for computation efficiency for large amounts of data objects. Most existing clustering methods, however, do not handle well the sparse similarity. The second problem arises due to the lack of suitable normalization of partitions in the objective function. Consequently, the resulting cluster sizes can be drastically and spuriously variable, which hampers the general reliability of the methods for applications. The third problem forces users to manually specify the number of clusters or to rely on external clustering evaluation methods which are usually inconsistent with the used learning objective.

In this paper we propose a new clustering method that addresses all the above-mentioned problems. Our learning objective is to minimize the discrepancy between the similarity matrix and the doubly-stochastically normalized cluster incidence matrix, which ensures balanced clusterings. Different from conventional squared Euclidean distance, the discrepancy in our approach is measured by Kullback-Leibler divergence which is more suitable for sparse similarity inputs. Minimizing the objective over all possible partitions automatically returns the most appropriate number of clusters.

We also propose an efficient and convenient algorithm for optimizing the introduced objective function. First, by using a new nonnegative matrix decomposition form, we find an equivalent solution space of low-rank nonnegative doubly stochastic matrices. This provides us a probabilistic surrogate learning objective called Data-Cluster-Data (DCD in short). Next we develop a new multiplicative algorithm for this surrogate function by using the majorization-minimization principle.

The new method is compared against many other clustering methods on various real-world data sets. The results show that the DCD method can often produce more accurate

clusterings, especially for large-scale manifold data sets containing up to hundreds of thousands of samples. We also demonstrate that it can select the number of clusters much more precisely than other existing approaches.

Although a preliminary version of our method has been presented earlier (Yang and Oja, 2012a), the current paper introduces several significant improvements. First, we show that the proposed clustering objective can be used to learn not only the cluster assignments but also the number of clusters. Second, we show that the proposed structural decomposition is equivalent to the low-rank factorization of a doubly stochastic matrix. Previously it was known that the former is a subset of the latter and now we prove that the reverse also holds. Third, we have performed much more extensive experiments, where the results further consolidate our conclusions.

The remainder of the paper is organized as follows. We review the basic clustering objective in Section 2. Its relaxed surrogate, the DCD learning objective, as well as optimization algorithm, are presented in Section 3, and the experimental results in Section 4, respectively. We discuss related work in Section 5 and conclude the paper by presenting possible directions for future research in Section 6.

2. Normalized Output Similarity Approximation Clusterability

Given a set of data objects $\mathcal{K} = \{x_1, \dots, x_N\}$, cluster analysis assigns them into r groups, called clusters, so that the objects in the same cluster are more similar to each other than to those in the other clusters. The cluster assignments can be represented by a *cluster indicator matrix* $\bar{F} \in \{0, 1\}^{N \times r}$, where $\bar{F}_{ik} = 1$ if x_i is assigned to the cluster C_k , and $\bar{F}_{ik} = 0$ otherwise. The *cluster incidence matrix* is defined as $M = \bar{F}\bar{F}^T$. Then $M \in \{0, 1\}^{N \times N}$ and $\bar{M}_{ij} = 1$ iff x_i and x_j are in the same cluster.

Let $S_{ij} \geq 0$ be a suitably normalized similarity measure between x_i and x_j . For a good clustering, it is natural to assume that the matrix S should be close to the cluster incidence matrix \bar{M} . Visually, \bar{M} is a blockwise matrix if we sort the data by their cluster assignment, and S should be nearly blockwise. The discrepancy or approximation error between S and \bar{M} can be measured by a certain divergence $D(S||\bar{M})$, e.g. Euclidean distance or Kullback-Leibler divergence. For example, He et al. (2011); Arora et al. (2011, 2013) used $D(S||\bar{M})$ with Euclidean distance to derive their respective NMF clustering methods.

Directly minimizing $D(S||\bar{M})$ can yield imbalanced partitions (see e.g., Shi and Malik, 2000). That is, some clusters are automatically of much smaller size than others, which is undesirable in many real-world applications. To achieve a balanced clustering, one can normalize \bar{M} to M in the approximation such that $\sum_{i=1}^N \bar{M}_{ij} = 1$ and $\sum_{j=1}^N \bar{M}_{ij} = 1$, or equivalently by normalizing \bar{F} to F with $F_{ik} = \bar{F}_{ik} / \sqrt{\sum_{v=1}^N \bar{F}_{iv}}$. The matrix $M = FF^T$ then becomes *doubly stochastic*. Such normalization has appeared in different clustering approaches (e.g., Ding et al., 2005; Shi and Malik, 2000). In this way, each cluster in M has unitary normalized volume (the ratio between sum of within-cluster similarities and the cluster size; also called normalized association (Shi and Malik, 2000)).

In this work we define clusterability as the maximum proximity between the data and a clustering solution. In similarity-based cluster analysis, $C(S) \stackrel{\text{def}}{=} -\min_{\bar{M}} D(S||\bar{M})$ can be used as a measure of clusterability for the similarity S over all normalized clusterings. For

an easy reference within this paper, we call it *Normalized Output Similarity Approximation Clusterability* (NOSAC) and $D(S||M)$ the NOSAC residual for a specific clustering M . Note that the optimum is taken over partitions which possibly have different values of r . Therefore the *optimization can be used to learn not only cluster assignments but also the number of clusters*. Similarly, minimizing $D(S||M)$ can also be used to select an optimum among partitions produced by e.g. different hyper-parameters or different initializations.

Minimizing $D(S||M)$ or $D(S||\bar{M})$ is equivalent to a combinatorial optimization problem in discrete space which is typically a difficult task (see e.g. Aloise et al., 2009; Mahajan et al., 2009; Shi and Malik, 2000). In practice it is customary to first solve a relaxed surrogate problem in continuous space and then perform discretization to obtain \bar{F} or F . Different relaxations include, for example, nonnegative matrices (Ding et al., 2008; Yang and Oja, 2010) and stochastic matrices Arora et al. (2011, 2013) for the unnormalized indicator \bar{F} in $D(S||\bar{M})$, and orthogonal matrices (Shi and Malik, 2000), as well as nonnegative and orthogonal matrices (Ding et al., 2006; Yang and Oja, 2012b; Yang and Laaksonen, 2007; Yoo and Choi, 2008; Pompili et al., 2013) for the normalized indicator F in $D(S||M)$.

In this paper we emphasize that the doubly stochasticity constraint is essential for balanced clustering, which cannot be guaranteed by the above relaxations. Besides this constraint, we also keep the nonnegativity constraint to achieve sparser low-rank factorizing matrix. These conditions as a whole provide a tighter relaxed solution space \mathbb{A} for M :

$$\mathbb{A} = \left\{ A \mid \forall_i, \sum_{j=1}^N A_{ij} = 1; A = UU^T; U \in \mathbb{R}^{N \times r}; \forall_i, k, U_{ik} \geq 0 \right\}. \quad (1)$$

To our knowledge, however, there is no existing technique that can minimize a generic cost function over low-rank A or over U . The major difficulty arises because the doubly stochasticity constraint is indirectly coupled with the factorizing matrix U . Note that this optimization problem is different from normalizing the input similarity matrix to be doubly stochastic before clustering (Zass and Shashua, 2006; He et al., 2011; Wang et al., 2012).

3. Low-rank Doubly Stochastic Matrix Decomposition by Probabilistic Relaxation

In this section we show how to minimize $D(S||A)$ over $A \in \mathbb{A}$ with suitable choice of discrepancy measure and complexity control. First we identify an equivalent solution space of \mathbb{A} which is easier for optimization. Then we develop the multiplicative minimization algorithm for finding a stationary point of the objective function.

3.1 Probabilistic relaxation

We find an alternative solution space by relaxing M to another matrix B in the matrix set

$$\mathbb{B} = \left\{ B \mid B_{ij} = \sum_{k=1}^r \frac{W_{ik}W_{jk}}{\sum_{v=1}^N W_{iv}W_{kv}}; W \in \mathbb{R}^{N \times r}; \forall_i \sum_{k=1}^r W_{ik} = 1; \forall_i, k, W_{ik} \geq 0 \right\}. \quad (2)$$

Comparing the original low-rank doubly stochastic matrix decomposition space \mathbb{A} with the DCD space \mathbb{B} shows that the following equivalence holds:

Theorem 1 $\mathbb{A} = \mathbb{B}$.

The proof is given in Appendix A. Previously it was known that $\mathbb{A} \supseteq \mathbb{B}$ (Yang and Oja, 2012a). Now this theorem shows that $\mathbb{A} \subseteq \mathbb{B}$ also holds, which implies that we do not miss any solution in \mathbb{A} by using \mathbb{B} . We prefer \mathbb{B} because the minimization in \mathbb{B} is easier, as there is no explicit doubly stochasticity constraint and we can work with the right stochastic matrix W . Note that W appears in both the numerator and denominator within the sum over k . Therefore this structural decomposition goes beyond conventional nonnegative matrix factorization schemes.

A probabilistic interpretation of B is as follows. Let $W_{ik} = P(k|i)$, the probability of assigning the i th data object to the k th cluster.¹ In the following, i, j , and v stand for data sample indices (from 1 to N) while k and l stand for cluster indices (from 1 to r). Without preference to any particular sample, we impose a uniform prior $P(j) = 1/N$ over the data samples. With this prior, we can compute

$$P(j|k) = \frac{P(k|j)P(j)}{\sum_{v=1}^r P(k|v)P(v)} = \frac{P(k|j)}{\sum_{v=1}^r P(k|v)} \quad (3)$$

by the Bayes' formula. Then we can see that

$$B_{ij} = \sum_{k=1}^r \frac{W_{ik}W_{jk}}{\sum_{v=1}^r W_{vk}} \quad (4)$$

$$= \sum_{k=1}^r \frac{P(k|j)}{\sum_{v=1}^r P(k|v)} P(k|i) \quad (5)$$

$$= \sum_{k=1}^r P(j|k)P(k|i) \quad (6)$$

$$= P(j|i). \quad (7)$$

That is, if we define a bipartite graph with the data samples and clusters as graph nodes, B_{ij} is the probability that the i th data node reaches the j th data node via a cluster node (see Figure 1). We thus call this DCD random walk or DCD decomposition after the Data-Cluster-Data walking paths. Since B is nonnegative and symmetric (which follows easily from the definition), in non-trivial cases it can be seen as another similarity measure between data objects. Our learning target is to find a good approximation between the input similarity matrix S and the low-rank output similarity B .

3.2 Kullback-Leibler divergence

Euclidean distance or Frobenius norm is a conventional choice for the discrepancy measure D . However, it is improper for many real-world clustering tasks where the raw data features are weakly informative. Similarities calculated with most simple metrics such as Euclidean distance or Hamming distance are then only accurate in a small neighborhood, whereas data

1. We use the abbreviation to avoid excessive notation. In full, $P(k|i) \stackrel{\text{def}}{=} P(\beta = C_k | \xi = x_i)$, where β and ξ are random variables with possible outcomes in clusters $\{C_k\}_{k=1}^r$ and data samples $\{x_i\}_{i=1}^N$, respectively. Similarly $P(i) \stackrel{\text{def}}{=} P(\xi = x_i)$ and $P(i|k) \stackrel{\text{def}}{=} P(\xi = x_i | \beta = C_k)$.

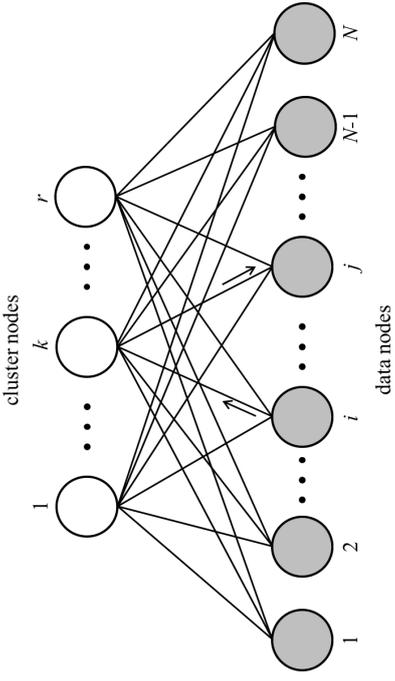


Figure 1: Data-cluster bipartite graph for N data samples and r clusters ($r < N$). The arrows show a Data-Cluster-Data (DCD) random walk path, which starts at the i th data node (sample) and ends at the j th data node via the k th cluster node.

points from different clusters often become mixed in wider neighborhoods. That is, only a small fraction of similarities, e.g. K -Nearest-Neighbor similarities with a relatively small value of K , are reliable and should be fed as a sparse input to clustering algorithms, while the similarities between the other, non-neighboring samples are set to zero. Least-square fitting with such a sparse similarity matrix is dominated by the approximation to the many zeros, which typically yields only poor or mediocre clustering results.

Here we propose to use (generalized) Kullback-Leibler divergence which is a more suitable approximation error measure between the sparse input similarity S and the dense output similarity B , because the approximation relies more heavily on the large values in S (see e.g. Févotte and Idier, 2011). The underlying Poisson likelihood models more appropriately the rare occurrences of reliable similarities. We thus formulate our learning objective in the relaxed DCD space as the following optimization problem:

$$\underset{W \geq 0}{\text{minimize}} \quad D_{\text{KL}}(S||B) = \sum_{i=1}^N \sum_{j=1}^N \left(S_{ij} \log \frac{S_{ij}}{B_{ij}} - S_{ij} + B_{ij} \right) \quad (8)$$

$$\text{subject to} \quad B_{ij} = \sum_{k=1}^r W_{ik}W_{jk} / \sum_{v=1}^r W_{vk}, \quad (9)$$

$$\sum_{k=1}^r W_{ik} = 1, \quad i = 1, \dots, N. \quad (10)$$

Dropping the constant terms in $D_{KL}(S||B)$, the objective function is equivalent to maximizing $\sum_{i=1}^N \sum_{j=1}^N S_{ij} \log B_{ij}$ because $\sum_{i=1}^N \sum_{j=1}^N B_{ij} = N$. Note that this form is similar to PLST (Hohmann, 1999) except for the decomposition form in B . Therefore we can enjoy the similar complexity control by using a Dirichlet prior (see Section 3.3).

Kullback-Leibler divergence also facilitates optimization, as we shall see in the algorithm development below. In objective and gradient calculation, it is straightforward to involve only the non-zero entries in S which brings an efficient implementation. Moreover, given properties of the logarithm function, we can break the comprehensive structure of B into several additive terms when majorizing the objective with the convex-concave procedure (Hunter and Lange, 2004). As a result, this yields relatively simple update rules in the algorithm. See Section 3.4 and Appendix B for details.

3.3 Regularization

The DCD learning objective function is parameterized by the matrix W whose rows sum to one. Assuming that these rows are observations from a common Dirichlet distribution, we can apply the log-Dirichlet prior to control the complexity in W . This gives the cost function of our DCD clustering method:

$$\mathcal{J}(W) = - \sum_{i=1}^N \sum_{j=1}^N S_{ij} \log B_{ij} - (\alpha - 1) \sum_{i=1}^N \sum_{k=1}^r \log W_{ik}. \quad (11)$$

This is also equivalent to regularization by using a total Shannon information term.

If S_{ij} are integers, the DCD objective is the log-likelihood of the following generative model: 1) draw the rows of W according to uniform Dirichlet distribution with parameter α ; 2) for $t = 1, \dots, T$, add one to entry $(i, j) \sim \text{Multinomial}(\frac{1}{N}B, 1)$. The Dirichlet prior vanishes when $\alpha = 1$. By using $\alpha > 1$, the prior gives further relaxation by smoothing the W entries, which is often desired in early stages of W learning.

Although it is possible to construct a multi-level graphical model similar to the Dirichlet process topic model (Blei et al., 2001; Sinkkonen et al., 2008), we emphasize that the smallest approximation error (i.e. with $\alpha = 1$) is the final DCD goal. The Dirichlet prior is used only in order to ease the optimization. Therefore we do not employ more complex generative models.

3.4 Optimization

Multiplicative updates are widely used in optimization for nonnegative matrix factorization problems. To minimize an objective \mathcal{J} over a nonnegative matrix W , we first calculate the gradient and separate it into two nonnegative parts ($\nabla_{ik}^+ \geq 0$ and $\nabla_{ik}^- \geq 0$):

$$\nabla_{ik} \stackrel{\text{def}}{=} \frac{\partial \mathcal{J}}{\partial W_{ik}} = \nabla_{ik}^+ - \nabla_{ik}^-. \quad (12)$$

Usually the separation can easily be identified from the gradient. Then the algorithm iteratively applies a multiplicative update rule $W_{ik} \leftarrow W_{ik} \frac{\nabla_{ik}^-}{\nabla_{ik}^+}$ until convergence. Such algorithms have several attractive properties, as they naturally maintain the positivity of

W and do not require extra effort to tune learning step size. For a variety of NMF problems, such multiplicative updates monotonically decrease \mathcal{J} after each iteration and therefore W can converge to a stationary point (Yang and Oja, 2011).

We cannot directly apply the above multiplicative fixed-point algorithm to DCD because there are probability constraints on the W rows. In practice, projecting the W rows to the probability simplex after each iteration would often lead to poor clustering result.

Instead, we employ a relaxing strategy (Zhu et al., 2013) to handle the probability constraint. We first introduce Lagrangian multipliers $\{\lambda_i\}_{i=1}^N$ for the constraints:

$$\mathcal{L}(W, \lambda) = \mathcal{J}(W) + \sum_i \lambda_i \left(\sum_{k=1}^r W_{ik} - 1 \right). \quad (13)$$

This suggests a preliminary multiplicative update rule for W :

$$W'_{ik} = W_{ik} \frac{\nabla_{ik}^- - \lambda_i}{\nabla_{ik}^+}, \quad (14)$$

where

$$\frac{\partial \mathcal{J}}{\partial W} = \underbrace{\left[(W^T Z W)_{kk} s_k^{-2} + W_{ik}^{-1} \right]}_{\nabla_{ik}^+} - \underbrace{\left[2 (Z W)_{kk} s_k^{-1} + \alpha W_{ik}^{-1} \right]}_{\nabla_{ik}^-}, \quad (15)$$

with $Z_{ij} = S_{ij}/B_{ij}$ and $s_k = \sum_{i=1}^N W_{ik}$. Imposing $\sum_k W'_{ik} = 1$ and isolating λ_i , we obtain

$$\lambda_i = \frac{b_i - 1}{a_i}, \quad (16)$$

where

$$a_i = \sum_{l=1}^r \frac{W_{il}}{\nabla_{il}^+}, \quad \text{and} \quad b_i = \sum_{l=1}^r \frac{W_{il}}{\nabla_{il}^-}. \quad (17)$$

Putting this λ back in Eq. 14, we obtain

$$W'_{ik} \leftarrow W_{ik} \frac{\nabla_{ik}^- a_i + 1 - b_i}{\nabla_{ik}^+ a_i}. \quad (18)$$

To maintain the positivity of W , we add b_i to both the numerator and denominator, which does not change the fixed point and gives the ultimate update rule:

$$W_{ik} \leftarrow W_{ik} \frac{\nabla_{ik}^- a_i + 1}{\nabla_{ik}^+ a_i + b_i}. \quad (19)$$

The above calculation steps are summarized in Algorithm 1. In implementation, one does not need to construct the whole matrix B . The ratio $Z_{ij} = S_{ij}/B_{ij}$ only requires calculation on the non-zero entries of S .

The above algorithm obeys a monotonicity guarantee provided by the following theorem.

Algorithm 1 Relaxed MM Algorithm for DCD

Input: similarity matrix S , number of clusters r , positive initial guess of W .

Output: cluster assigning probabilities W .

repeat

$$B_{ij} = \sum_{k=1}^r \frac{W_{ik}W_{jk}}{\sum_{v=1}^r W_{vk}}$$

$$Z_{ij} = S_{ij}/B_{ij}$$

$$s_k = \sum_{v=1}^N W_{vk}$$

$$\nabla_{ik}^- = 2(ZW)_{ik}s_k^{-1} + \alpha W_{ik}^{-1}$$

$$\nabla_{ik}^+ = (W^TZW)_{ik}s_k^{-2} + W_{ik}^{-1}$$

$$a_i = \sum_{l=1}^r \frac{W_{il}}{\nabla_{il}^+}, \quad b_i = \sum_{l=1}^r W_{il} \frac{\nabla_{il}^-}{\nabla_{il}^+}$$

$$W_{ik} \leftarrow \frac{\nabla_{ik}^- a_i + 1}{W_{ik} \frac{\nabla_{ik}^-}{\nabla_{ik}^+} a_i + b_i}$$

until W converges under the given tolerance

Theorem 2 Denote W^{new} the updated matrix after each iteration of Algorithm 1. It holds that $\mathcal{L}(W^{new}, \lambda) \leq \mathcal{L}(W, \lambda)$ with $\lambda_i = (b_i - 1)/a_i$.

The proof (given in Appendix B) mainly follows the Majorization-Minimization procedure (Hunter and Lange, 2004; Yang and Oja, 2011). The theorem shows that Algorithm 1 jointly minimizes the approximation error and drives the rows of W towards the probability simplex. The Lagrangian multipliers are adaptively and automatically selected by the algorithm, without extra human tuning effort. The quantities b_i are the row sums of the unconstrained multiplicative learning result, while the quantities a_i balance between the gradient learning force and the probability simplex attraction. Besides convenience, we find that this relaxation strategy works more robustly than the brute-force projection after each iteration.

DCD minimizes the relaxed NOSAC residual for a particular r . To select the best number of clusters, we can run Algorithm 1 over a range of r values, discretize W to obtain the hard cluster assignment \bar{F} , and return the one with smallest NOSAC residual. See Section 4.2 for examples.

3.5 Initialization

Proper initialization is needed to achieve satisfactory performance for practically any clustering method that involves non-convex optimization. DCD accepts any clustering results as its starting point. In our implementation, we add a small positive perturbation (e.g. 0.2) to all entries of the initial cluster indicator matrix. Next, the perturbed matrix is fed to our optimization algorithm (with $\alpha = 1$ in Algorithm 1). Among all runs of DCD, we return the clustering result with the smallest $D(S||M)$.

In particular, the regularized DCD (i.e. with various $\alpha \neq 1$) can also provide initialization for the non-regularized DCD (i.e. with $\alpha = 1$). That is, the parameter α only appears in the initialization and its best value is also determined by the smallest resulting $D(S||M)$.

4. Experiments

We have tested the DCD method and compared it with other existing cluster analysis approaches. The experiments were organized in two groups: 1) we ran the methods with a fixed number of clusters, mainly comparing their clustering accuracies; 2) we ran DCD across different r values, demonstrating how to use NOSAC residual to determine the optimal number of clusters as well as its advantage over several other clustering evaluation methods.

4.1 Clustering with known number of clusters

In the first group of experiments, the number of the ground truth classes in the data sets was known in advance, and we fixed r to that number. We compared DCD with a variety of state-of-the-art clustering methods, including Projective Nonnegative Matrix Factorization (PNMF; Yang et al., 2007; Yang and Oja, 2010), Nonnegative Spectral Clustering (NSC; Ding et al., 2008), Orthogonal Nonnegative Matrix Factorization (ONMF; Ding et al., 2006), Probabilistic Latent Semantic Indexing (PLSI; Hofmann, 1999), Left-Stochastic Decomposition (LSD; Arora et al., 2011, 2013), as well as two classical methods k-means (Lloyd, 1982) and Normalized Cut (Ncut; Shi and Malik, 2000). Besides NMF, we have also selected several recent clustering methods, including 1-Spectral (1-Spec; Hein and Bühler, 2010), Landmark-based Spectral Clustering (LSC; Chen and Cai, 2011; Cai and Chen, 2015), Sparse Subspace Clustering (SSC; Elhamifar and Vidal, 2009, 2013), and Multiclass Total Variation (MTV; Bresson et al., 2013). There are some other recent methods (e.g., Rodriguez and Laio, 2014; Liu and Tao, 2016), which is however not scalable to large numbers of samples and thus not included here.

We used default settings in the compared methods. The NMF-type methods were run with maximum 10,000 iterations of multiplicative updates and with convergence tolerance 10^{-6} . We used ratio Cheeger cut for 1-Spec. All methods except k-means, Ncut, 1-Spec, LSC, SSC, and MTV were initialized by Ncut. That is, their starting point was the Ncut cluster indicator matrix plus a small constant 0.2 to all entries.

We have compared the above methods on 43 data sets from various domains, including biology, image, video, text, remote sensing, etc. All data sets are publicly available on the Internet. The data sources and statistics are given in the supplemental document. For similarity-based clustering methods, we constructed K -Nearest-Neighbor graphs from the multivariate data with $K = 10$. The adjacency matrices of the graphs are then symmetrized and binarized to obtain S , i.e. $S_{ij} = 1$ if x_j is one of the K nearest neighbors of x_i or vice versa; otherwise $S_{ij} = 0$. This produces sparse similarity matrices.

We have used two performance measures for the clustering accuracies: the first is *cluster purity* which equals $\frac{1}{N} \sum_{k=1}^r \max_{1 \leq l \leq r} n_{kl}$, where n_{kl} is the number of data samples in the cluster k that belong to ground-truth class l ; the second performance measure is *Normalized Mutual Information* (NMI, Vinh et al., 2010) which equals $\sum_{k=1}^r \sum_{l=1}^r \frac{n_{kl}}{N} \log \frac{n_{kl}/N}{a_k b_l / N^2}$, with $a_k = \sum_{l=1}^r n_{kl}$ and $b_l = \sum_{k=1}^r n_{kl}$.

The resulting cluster purities and NMI's are shown in Tables 1 and 2, respectively. We can see that our method has much better performance than the other methods. DCD shows the optimal performance for 22 and 18 out of 43 data sets in purity and NMI, respectively, which is substantially more frequently than for any of the other methods. Even for some other data sets where DCD is not the winner, its cluster purities still remain close to the best method. Our method shows particularly superior performance when the number of samples grows. For the 19 data sets with $N > 4500$, DCD is the top performer in 17 and 11 cases in purity and NMI, respectively. Note that purity corresponds to classification accuracy up to a permutation between classes and clusters. In this sense, our method achieves accuracy very close to many modern supervised approaches for some large-scale data in a curved manifold such as MNIST², though our method does not use any class labels. For text document data set 20NG, DCD achieves comparable accuracy to those with comprehensive feature engineering and supervised classification (e.g. Srivastava et al., 2013), even though our method only uses simple bag-of-words TF-IDF features and no class labels at all.

4.2 Selecting the number of clusters

In the second group of experiments, we assume that the number of clusters is unknown and it must be automatically selected from a range around the number of ground truth classes. We ran DCD with different values of r and calculated the corresponding NOSAC residual after discretizing W to cluster indicator matrix: the best number of clusters was then selected by the r with the smallest NOSAC residual.

We have compared the above DCD selection method with several other clustering evaluation methods: Calinski-Harabasz (CH; Calinski and Harabasz, 1974), Davies-Bouldin (DB; Davies and Bouldin, 1979) and gap statistics (Tibshirani et al., 2001). We used their implementation in Matlab. Each cluster evaluation approach comes with a supported base clustering algorithm k-means (km) or linkage (lk). We thus have in total six methods for selecting the number of clusters to be compared against DCD. Some of these compared methods are very slow and required computation of more than five days for certain data sets. This drawback is especially severe for data sets with very high dimensionality such as CURLET and COLL100. In contrast, DCD required at most two hours for any tested data set.

The results are reported in Table 3, which shows that DCD performs the best also in terms of selecting the number of clusters. The corresponding curves of NOSAC residual vs. number of clusters by DCD are shown in Figure 2. In the selected number of clusters, the DCD results are closest to the ground truth for all data sets, much more accurately than for any of the other methods. DCD correctly selects the best for CURLET, OPTDIGITS, and MNIST, and almost correctly (only differing by 1) for BOTSWANA and PHONEME. For COLL20 and COLL100, the DCD results are also reasonably good because we did not deliberately tune the extent of the local neighborhoods. By simply replacing 10NN with 5NN as the input similarities, DCD respectively selects 21 for COLL20 and 99 for COLL100 as the best number of clusters.

Table 1: Clustering purities for the compared methods on various data sets. Boldface numbers indicate the best in each row. “-” means out-of-memory error.

Data set	N	DCD	k-means	Neut	PNMF	NSC	ONMF	PLSI	LSD	1-Spec	LSC	SSC	MIV
AMTALL	38	0.95	0.68	0.50	0.95	0.95	0.76	0.97	0.80	0.97	0.89	0.89	0.53
NCI	64	0.64	0.67	0.38	0.66	0.62	0.55	0.62	0.59	0.44	0.69	0.75	0.41
BT	106	0.47	0.45	0.29	0.50	0.50	0.44	0.49	0.48	0.42	0.34	0.34	0.47
IRIS	150	0.91	0.89	0.39	0.93	0.90	0.48	0.72	0.73	0.91	0.79	0.73	0.67
YALE	165	0.59	0.59	0.22	0.61	0.55	0.41	0.59	0.58	0.25	0.68	0.59	0.25
WINE	178	0.95	0.95	0.41	0.96	0.95	0.40	0.93	0.95	0.94	0.95	0.91	0.94
HCANCER	198	0.53	0.48	0.21	0.51	0.51	0.49	0.52	0.53	0.47	0.52	0.64	0.24
GLASS	214	0.87	0.88	0.38	0.75	0.83	0.81	0.71	0.78	0.68	0.75	0.80	0.83
VERTEBRAL	310	0.77	0.72	0.48	0.75	0.76	0.74	0.69	0.69	0.77	0.77	0.71	0.76
ECCOL	336	0.80	0.83	0.43	0.81	0.81	0.80	0.76	0.84	0.80	0.79	0.71	0.76
SVANGUIDE	391	0.71	0.63	0.57	0.61	0.58	0.57	0.64	0.66	0.61	0.57	0.71	0.66
ORL	400	0.73	0.75	0.17	0.77	0.76	0.64	0.70	0.77	0.75	0.77	0.77	0.29
VOWEL	990	0.55	0.39	0.14	0.37	0.37	0.37	0.30	0.34	0.28	0.31	0.28	0.30
MED	1.0K	0.58	0.63	0.33	0.55	0.55	0.53	0.53	0.57	0.42	0.37	0.55	0.33
COLL20	1.4K	0.82	0.61	0.11	0.69	0.73	0.49	0.41	0.56	0.30	0.83	0.82	0.75
YEAST	1.5K	0.55	0.52	0.34	0.50	0.51	0.53	0.48	0.51	0.54	0.52	0.46	0.46
ISOLET	1.6K	0.59	0.55	0.09	0.59	0.58	0.54	0.51	0.57	0.29	0.53	0.56	0.55
SEMION	1.6K	0.93	0.63	0.13	0.81	0.73	0.67	0.60	0.91	0.73	0.92	0.77	0.83
MPEAT	2.0K	0.80	0.57	0.13	0.71	0.64	0.44	0.51	0.59	0.37	0.76	0.80	0.65
DNA	2.0K	0.68	0.76	0.53	0.57	0.54	0.53	0.62	0.65	0.53	0.54	0.53	0.53
SEG	2.3K	0.74	0.57	0.17	0.54	0.57	0.48	0.26	0.35	0.43	0.78	0.56	0.40
BOTSWANA	3.2K	0.75	0.57	0.11	0.65	0.59	0.54	0.33	0.43	0.41	0.69	0.66	0.52
CITTESER	3.3K	0.50	0.68	0.22	0.31	0.25	0.29	0.25	0.37	0.36	0.48	0.50	0.27
WEBKB	4.2K	0.56	0.44	0.07	0.42	0.53	0.41	0.56	0.54	0.48	0.57	0.39	0.51
OUTTEX	4.3K	0.55	0.44	0.07	0.46	0.39	0.44	0.39	0.53	0.21	0.65	0.07	0.45
SATIMAGE	4.4K	0.76	0.75	0.25	0.67	0.77	0.68	0.49	0.62	0.82	0.74	0.70	0.48
PHONEME	4.5K	0.87	0.71	0.26	0.87	0.82	0.50	0.72	0.86	0.82	0.86	0.70	0.84
7SECTORS	4.6K	0.46	0.31	0.24	0.28	0.26	0.24	0.29	0.28	0.24	0.28	0.24	0.32
KSC	5.2K	0.64	0.18	0.18	0.60	0.53	0.39	0.33	0.42	0.50	0.64	0.50	0.54
BRUNNA	5.6K	0.94	0.56	0.05	0.87	0.80	0.82	0.53	0.76	0.42	0.78	0.70	0.78
OPTDIGITS	5.6K	0.98	0.74	0.12	0.86	0.76	0.76	0.46	0.77	0.60	0.92	0.89	0.98
GISLETTE	7.0K	0.93	0.68	0.51	0.52	0.64	0.51	0.39	0.61	0.93	0.90	0.78	0.52
COLL100	7.2K	0.81	0.60	0.05	0.68	0.70	0.51	0.39	0.61	0.20	0.64	0.80	0.66
ZIP	9.3K	0.85	0.54	0.17	0.57	0.67	0.41	0.46	0.63	0.81	0.79	0.74	0.80
TDT2	10K	0.86	0.86	0.89	0.89	0.89	0.88	0.88	0.87	0.53	0.85	0.85	0.76
PENDIGITS	11K	0.86	0.71	0.80	0.77	0.80	0.77	0.81	0.85	0.27	0.78	0.83	0.76
20NG	20K	0.62	0.39	0.40	0.38	0.40	0.39	0.47	0.48	0.06	0.50	0.17	0.19
LETTER	20K	0.38	0.29	0.29	0.29	0.36	0.31	0.37	0.34	0.34	0.11	0.35	0.33
MNIST	70K	0.97	0.48	0.77	0.84	0.79	0.73	0.79	0.81	0.88	0.87	-	0.85
NORB	97K	0.35	0.22	0.21	0.26	0.21	0.26	0.33	0.42	0.20	0.20	-	0.32
ACOUSSTIC	99K	0.61	0.60	0.55	0.54	0.56	0.54	0.57	0.54	0.57	0.57	-	0.51
MOCAP	217K	0.29	0.21	0.14	0.18	0.12	0.17	0.14	0.17	0.07	0.25	-	0.14
COVTYPE	581K	0.56	0.49	0.51	0.53	0.51	0.54	0.51	0.49	0.49	0.49	-	0.49

5. Discussion

The results in the previous section demonstrate the solid performance of DCD on a wide variety of data sets. In this section, we discuss the connections and differences between DCD and other related work, and we also discuss other implementations of the input similarities than the symmetrized K-Nearest-Neighbors.

² see <http://yann.lecun.com/exdb/mnist/>

Table 2: Clustering NMI for the compared methods on various data sets. Boldface numbers indicate the best in each row. “-” means out-of-memory error.

Data set	N	DCD	k-means	Ncut	PNNMF	NSC	ONMF	PLSI	LSD	1-Spec	LSC	SSC	MTV
AMLALL	38	0.85	0.37	0.03	0.81	0.81	0.49	0.91	0.72	0.91	0.77	0.68	0.04
NCI	64	0.66	0.68	0.44	0.66	0.64	0.60	0.66	0.61	0.46	0.71	0.75	0.45
BR	106	0.34	0.36	0.07	0.35	0.36	0.30	0.37	0.34	0.37	0.29	0.17	0.37
IRIS	150	0.81	0.76	0.02	0.82	0.78	0.10	0.59	0.59	0.81	0.59	0.61	0.76
YALE	165	0.62	0.64	0.27	0.62	0.57	0.50	0.61	0.61	0.37	0.66	0.62	0.29
WINE	178	0.84	0.83	0.01	0.85	0.84	0.00	0.80	0.84	0.83	0.84	0.72	0.83
14CANCER	198	0.53	0.47	0.20	0.52	0.51	0.50	0.54	0.54	0.50	0.51	0.63	0.21
GLASS	214	0.74	0.75	0.04	0.67	0.69	0.71	0.56	0.67	0.54	0.64	0.63	0.68
VERTBRAL	310	0.52	0.41	0.00	0.47	0.48	0.46	0.38	0.36	0.54	0.56	0.36	0.52
ECOLI	336	0.58	0.64	0.05	0.58	0.57	0.59	0.50	0.59	0.60	0.54	0.39	0.67
SVNGUIDE	391	0.22	0.08	0.00	0.12	0.10	0.12	0.13	0.21	0.11	0.02	0.21	0.17
ORL	400	0.85	0.88	0.40	0.86	0.86	0.80	0.84	0.86	0.61	0.90	0.88	0.50
VOWEL	990	0.40	0.45	0.02	0.37	0.41	0.35	0.29	0.33	0.35	0.29	0.23	0.31
MED	1.0K	0.59	0.60	0.15	0.57	0.57	0.54	0.56	0.58	0.46	0.33	0.55	0.21
COIL20	1.4K	0.88	0.76	0.05	0.81	0.80	0.71	0.56	0.70	0.52	0.92	0.89	0.83
YEAST	1.5K	0.27	0.26	0.02	0.22	0.27	0.26	0.22	0.23	0.27	0.21	0.20	0.20
ISOLET	1.6K	0.72	0.71	0.07	0.72	0.71	0.69	0.67	0.71	0.60	0.69	0.71	0.69
SEMION	1.6K	0.87	0.60	0.01	0.79	0.72	0.69	0.56	0.84	0.77	0.85	0.73	0.82
MFEAT	2.0K	0.75	0.58	0.01	0.68	0.61	0.45	0.43	0.56	0.66	0.73	0.75	0.68
DNA	2.0K	0.25	0.38	0.00	0.08	0.04	0.07	0.18	0.21	0.05	0.07	0.02	0.00
SEG	2.3K	0.66	0.57	0.00	0.44	0.49	0.44	0.09	0.21	0.59	0.68	0.45	0.30
BOTSWANA	3.2K	0.64	0.01	0.66	0.61	0.60	0.60	0.29	0.48	0.55	0.72	0.64	0.58
CITSEER	3.3K	0.22	0.39	0.00	0.10	0.07	0.08	0.08	0.10	0.15	0.17	0.27	0.03
WEBKB	4.2K	0.23	0.09	0.00	0.07	0.20	0.07	0.21	0.19	0.12	0.18	0.02	0.13
OUTEX	4.3K	0.67	0.65	0.02	0.66	0.55	0.63	0.50	0.64	0.49	0.82	0.12	0.60
SATIMAGE	4.4K	0.61	0.62	0.00	0.56	0.63	0.53	0.21	0.51	0.67	0.62	0.54	0.31
PHONEME	4.5K	0.82	0.57	0.00	0.81	0.81	0.36	0.58	0.82	0.81	0.81	0.62	0.84
7SECTORS	4.6K	0.21	0.08	0.00	0.04	0.06	0.01	0.04	0.09	0.03	0.10	0.04	0.05
KSC	5.2K	0.57	0.05	0.01	0.58	0.48	0.37	0.18	0.30	0.55	0.59	0.49	0.48
BRUN	5.6K	0.95	0.74	0.09	0.91	0.88	0.89	0.74	0.87	0.79	0.88	0.83	0.89
OPTDIGITS	5.6K	0.96	0.72	0.00	0.85	0.77	0.77	0.45	0.77	0.72	0.88	0.86	0.95
GISETTE	7.0K	0.65	0.12	0.00	0.00	0.06	0.00	0.03	0.22	0.62	0.54	0.25	0.00
COIL100	7.2K	0.90	0.82	0.16	0.85	0.84	0.78	0.69	0.81	0.44	0.83	0.92	0.85
ZIP	9.3K	0.82	0.46	0.00	0.59	0.61	0.50	0.35	0.59	0.83	0.77	0.78	0.78
TDT2	10K	0.70	0.71	0.74	0.72	0.74	0.71	0.73	0.71	0.61	0.69	0.74	0.64
PENDIGITS	11K	0.83	0.68	0.82	0.76	0.83	0.76	0.81	0.81	0.42	0.48	0.29	0.13
20NG	20K	0.54	0.44	0.52	0.37	0.52	0.38	0.47	0.44	0.08	0.48	0.29	0.13
LETTER	20K	0.49	0.36	0.39	0.45	0.37	0.45	0.41	0.42	0.20	0.43	0.38	0.36
MINST	70K	0.93	0.89	0.81	0.83	0.84	0.01	0.82	0.80	0.89	0.80	-	0.88
NORB	97K	0.19	0.00	0.01	0.03	0.01	0.02	0.09	0.26	0.00	0.00	-	0.07
ACOUSTIC	99K	0.17	0.15	0.15	0.07	0.15	0.07	0.14	0.09	0.15	0.14	-	0.07
MOCAP	217K	0.38	0.26	0.19	0.22	0.18	0.21	0.17	0.22	0.07	0.30	-	0.14
COVTYPE	581K	0.16	0.08	0.06	0.05	0.06	0.15	0.07	0.04	0.02	0.08	-	0.01

5.1 Comparison with related techniques

5.1.1 SPECTRAL CLUSTERING, K-MEANS, AND ORTHOGONALITY CONSTRAINT

Denote $X = [x_1, \dots, x_N]^T$ the data matrix (rows to be clustered). The classical k-means method seeks a clustering such that the sum of squared Euclidean distances between the samples and their assigned cluster means is minimized. This objective can be expressed by using the normalized cluster indicator matrix F defined in Section 2: $\min_F \|X - FF^T X\|_{F_0}^2 = \|X\|_{F_0}^2 - \text{Tr}(F^T X X^T F)$, where $\|\cdot\|_{F_0}$ is the Frobenius norm (Ding et al.,

Table 3: The automatically selected number of clusters. Boldface number indicates the closest (best) to the ground truth (number of classes, in the last column) in each row. The results marked with a star required computation of more than five days.

Data set	CH-km	DB-km	gap-km	CH-lk	DB-lk	gap-lk	DCD	#classes
COIL20	2	31	30*	7	2	40*	17	20
BOTSWANA	8	4	30*	7	4	9	13	14
PHONEME	3	2	10	3	2	4	5	5
CURET	41*	79*	80*	41*	77*	80*	61	61
OPTDIGITS	2	9	20	3	14	20	10	10
COIL100	61*	139*	140*	61*	137*	140*	79	100
MINST	2	20	20*	2	2	20*	10	10

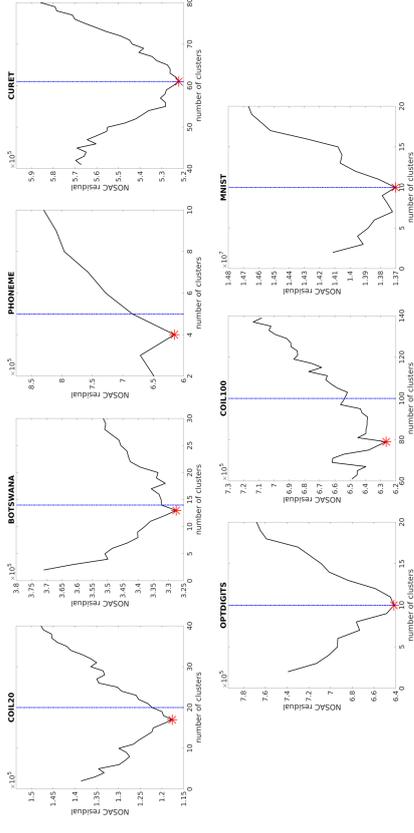


Figure 2: Selecting the best number of clusters using NOSAC residual. The red star shows the smallest NOSAC residual. The vertical blue dot-dashed line shows the ground truth (number of classes).

2005). The k-means method can be extended to a nonlinear case by replacing XX^T with another kernel matrix.

It is difficult to directly minimize over F in the combinatorial space (Aloise et al., 2009; Mahajan et al., 2009). A conventional way is to relax F to orthogonal matrix such that the optimization can be solved by eigendecomposition. This connects k-means or kernel k-means to spectral clustering (Ding et al., 2005). Despite the closed form solution, the obtained eigenvectors do not immediately reveal the cluster assignments. Extra effort such as k-means on the relaxed F rows (Ng et al., 2001) or iterative projection (Yu and Shi, 2003) is needed to convert the eigenvectors to the cluster indicator matrix. An alternative way is

to combine orthogonality and nonnegativity such that the relaxed F has only one non-zero entry in each row and thus indicates the cluster assignments (Ding et al., 2006; Yang and Oja, 2012b; Yang and Laaksonen, 2007; Yoo and Choi, 2008; Pomplii et al., 2013).

However, the orthogonality constraint does not necessarily guarantee balanced clustering because it does not restrict the magnitudes of the relaxed F rows. Moreover, the orthogonality favors Euclidean distance as the approximation error measure for simple update rules, which is against our requirement of the sparse similarity graph input.

In contrast, our relaxation employs doubly stochasticity of the relaxed FF^T (i.e. A or B), which ensures that each cluster has unitary (soft) normalized graph volume³ when combined with the nonnegativity constraint. Furthermore, although we do not explicitly use the orthogonality, in practice the resulting relaxed F (i.e. U) contains only one or a few significant non-zero entries in each row for clustered data. Therefore the best DCD objective is close to the discrete NOSAC residual by which we can select the number of clusters (see Section 4.2). This cannot be done in k -means or spectral clustering.

5.1.2 NONNEGATIVE MATRIX FACTORIZATION

Nonnegative Matrix Factorization (NMF) seeks nonnegative low-rank factorization of an input data matrix (Lee and Seung, 1999, 2001). Variants of NMF have been proposed for similarity-based clustering. For example, Ding et al. (2008) imposed nonnegativity to spectral clustering; Ding et al. (2006); Yang and Oja (2012b); Yang and Laaksonen (2007); Yoo and Choi (2008); Pomplii et al. (2013) proposed using both nonnegativity and orthogonality on the factorizing matrices; He et al. (2011) used the symmetric NMF for low-rank factors.

Probabilistic clustering is a natural way to relax the hard clustering problem. Recently Avora et al. (2011, 2013) introduced stochasticity for clustering, by using a left stochastic matrix in symmetric NMF. However, their method, called LSD, is restricted to the Euclidean distance. In addition, LSD does not prevent imbalanced clustering.

Our method has two major differences from LSD. First, our decomposition involves a normalizing factor which emphasizes balanced clusterings. Second, we use Kullback-Leibler divergence which is more suitable for sparse graph input or curved manifold data. This also enables us to make use of the Dirichlet and multinomial conjugacy pair to achieve more accurate clusterings.

5.1.3 PROBABILISTIC LATENT SEMANTIC ANALYSIS

Probabilistic Latent Semantic Analysis (PLSA, also known as PLSI especially in information retrieval) is a statistical technique for the analysis of two-mode and co-occurrence data. When PLSA applies to similarity-based clustering (Hofmann, 1999), it maximizes the log-likelihood $\sum_{i=1}^N \sum_{j=1}^N S_{ij} \log \sum_{k=1}^K P(i|k)P(j|k)$. Compared to the DCD objective, we can see that the major difference is in the decomposition form within the logarithm: PLSA learns the cluster prior $P(k)$ and the conditional likelihood of data points $P(i|k)$; while in DCD we assume uniform prior $P(i)$ and learn the conditional likelihood of clusters $P(k|i)$.

³ For the k th cluster, the soft cluster volume is $\sum_{i=1}^N \sum_{j=1}^N \frac{W_{i,j} W_{j,k}}{\sum_{i=1}^N \sum_{k=1}^K W_{i,j}}$ and soft cluster size is $\sum_{i=1}^N W_{i,k}$.

There are several reasons why the DCD decomposition is more beneficial than PLSA for cluster analysis. First, $P(k|i)$ in DCD is the direct answer to the probabilistic clustering problem, while the PLSA quantities are not. Second, in PLSA $\sum_{k=1}^K P(k)P(i|k)P(j|k) = P(i, j)$ is a joint probability matrix; it is not necessarily doubly stochastic and may not guarantee that each cluster has the same normalized graph volume, as DCD does. Third, DCD achieves a good balance in terms of the number of parameters, as it contains $N \times (r-1)$ free parameters while in PLSA there are $N \times r - 1$; this difference can be large when there are only few clusters (e.g. $r = 2$ or $r = 3$).

Both methods can be improved by using Dirichlet priors. In DCD the prior is only used in initialization and the prior parameter is chosen according to the smallest NOSAC residual. We find that this strategy is better than the conventional hyper-parameters tuning techniques in the topic model literature (e.g., Minka, 2000; Asuncion et al., 2009).

5.1.4 DOUBLY STOCHASTIC MATRIX PROJECTION

Normalizing a matrix to be doubly stochastic has been used to improve cluster analysis, but mainly on the input similarity matrix. The normalization dates back to the Sinkhorn-Knopp procedure (Sinkhorn and Knopp, 1967) or iterative proportional fitting procedure (Bishop et al., 1975). Zass and Shashua (2006) proposed to improve spectral clustering by replacing the original similarity matrix by its closest doubly stochastic similarities under L_1 or Frobenius norm. Wang et al. (2012) generalized the projection to the family of Bregman divergences. Note that the normalized matrix in general requires $O(N^2)$ memory if Frobenius norm projection is used.

In contrast, our method has three major differences: 1) it imposes the doubly stochastic constraint on the approximating matrix instead of the input similarity matrix; 2) the doubly stochastic matrix must be low-rank; in practice we need only $O(N \times r)$ memory; 3) our DCD decomposition equivalently fulfills the doubly stochastic requirement, and thus no extra normalization is needed.

5.1.5 CLUSTERABILITY

Clusterability or clustering tendency measures how “strong” or “conclusive” is the clustering structure of a given data set (Ackeman and Ben-David, 2009). The research dates back to Hopkins index for spatial randomness test (Hopkins and Skellam, 1954). Other notions include, for example, center perturbation clusterability (Ben-David et al., 2002), worst pair ratio clusterability (Eptel et al., 1999), separability clusterability (Ostrovsky et al., 2006), variance ratio clusterability (Ostrovsky et al., 2006), strict separation clusterability (Balcan et al., 2008), and target clusterability (Balcan et al., 2009). Ackeman and Ben-David (2009) gave a survey and comparison on the above clusterability notions.

These clusterability criteria, however, suffer from one or more of the following drawbacks: 1) they are defined over k -partitions with a fixed k and therefore cannot be used for clusterings with various k values; 2) they are restricted to center-based clustering methods and might not work for curved clusters; and 3) they employ minimum within-cluster distances and maximum between-cluster distances, which is sensitive to outlier data points.

In contrast, our NOSAC criterion does not have the above drawbacks. NOSAC is defined over all partitions, including those with a different number of clusters. The partitions can

Table 4: Clustering performance using approximated KNN: (top) purities and (bottom) NMI. Boldface numbers indicate the best in each row. “-” means out-of-memory error.

Data set	N	DCD	k-means	Neut	PNMF	NSC	ONMF	PLSI	LSD	1-Spec	LSC	SSC	MTV
TD72	10K	0.87	0.86	0.20	0.89	0.84	0.73	0.87	0.54	0.85	0.88	0.79	
PENDIGITS	11K	0.90	0.71	0.12	0.70	0.63	0.61	0.36	0.58	0.43	0.78	0.83	0.77
20NG	20K	0.61	0.39	0.06	0.39	0.45	0.28	0.17	0.34	0.06	0.50	0.17	0.21
LETTER	20K	0.36	0.29	0.05	0.35	0.24	0.33	0.16	0.26	0.10	0.35	0.33	0.26
MNIST	70K	0.96	0.48	0.11	0.68	0.63	0.58	0.40	0.70	0.78	0.87	-	0.92
NORB	97K	0.41	0.22	0.30	0.25	0.30	0.25	0.30	0.38	0.30	0.20	-	0.30
ACOUSTIC	99K	0.60	0.50	0.50	0.54	0.50	0.54	0.50	0.52	0.55	0.57	-	0.53
MOCAP	217K	0.29	0.21	0.05	0.20	0.07	0.19	0.06	0.17	0.06	0.25	-	0.09
COVTYPE	581K	0.55	0.49	0.51	0.50	0.51	0.50	0.49	0.49	0.49	0.49	-	0.49

Data set	N	DCD	k-means	Neut	PNMF	NSC	ONMF	PLSI	LSD	1-Spec	LSC	SSC	MTV
TD72	10K	0.71	0.71	0.08	0.71	0.74	0.68	0.62	0.70	0.63	0.69	0.74	0.66
PENDIGITS	11K	0.87	0.68	0.00	0.69	0.57	0.61	0.25	0.58	0.55	0.76	0.77	0.78
20NG	20K	0.54	0.44	0.00	0.38	0.53	0.26	0.11	0.27	0.08	0.48	0.29	0.13
LETTER	20K	0.48	0.36	0.00	0.44	0.28	0.42	0.18	0.37	0.18	0.43	0.38	0.35
MNIST	70K	0.91	0.39	0.00	0.68	0.61	0.65	0.32	0.70	0.82	0.80	-	0.87
NORB	97K	0.22	0.00	0.13	0.05	0.13	0.05	0.09	0.18	0.16	0.00	-	0.10
ACOUSTIC	99K	0.14	0.17	0.00	0.05	0.03	0.06	0.04	0.07	0.15	0.14	-	0.09
MOCAP	217K	0.37	0.26	0.00	0.21	0.05	0.19	0.03	0.19	0.06	0.30	-	0.07
COVTYPE	581K	0.16	0.08	0.06	0.04	0.06	0.04	0.07	0.04	0.03	0.08	-	0.01

be obtained from any clustering methods, not necessarily center-based. Moreover, we use matrix divergences instead of only minimum or maximum of individual distances, which provides a more robust measure against outliers.

5.2 Input similarities

The inputs to DCD are the pairwise similarities between data items, for example the symmetrized and binarized KNN graph used in Section 4. Naive implementation of KNN requires $O(N^2)$ computational cost. There exist accelerated algorithms taking advantage of the fact that it is often not necessary to calculate all pairs but only those in local neighborhoods. We have used a simple implementation with a vantage-point index (Yianilos, 1993), where we slightly modified the code to admit sparse data and with interface to Matlab. For MNIST where $N = 70,000$, the accelerated KNN (with $K = 10$) algorithm requires in practice only about 7 minutes to completion.

Exact KNN by the above acceleration is still expensive for even larger data sets. In practice, we find that using highly accurate approximated KNN is enough for maintaining the DCD performance. Table 4 shows the comparison for large-scale data sets ($N > 10,000$) using the Fast Library of Approximated Nearest Neighbors (FLANN; Muja and Lowe, 2014). We can see that the resulting DCD purities and NMIs are close to those with exact KNN, and that the accuracy gains over the other compared methods mostly remain. By using FLANN, we can obtain the similarities for DCD much faster; for example, FLANN takes about 30 seconds for MNIST with $K = 10$.

KNN is not the only choice of input similarities. There are other more advanced neighborhood descriptors which could further improve DCD, for example, *Entropy Affinities*

which locally scales the spherical Gaussian kernels such that the neighborhoods around every data point have the same given entropy (Vladymyrov and Carreira-Perpinán, 2013), *Sparse Manifold Clustering and Embedding* that learns a sparse coding with respect to local manifold geometry and cluster distribution (Elhamifar and Vidal, 2011), and *AnchorGraph* which learns the low-rank sparse coding with a set of pre-clustered landmarks (Liu et al., 2010). Other approaches such as metric learning (e.g., Kulis, 2013) could also be applied to obtain better input similarities.

6. Conclusions

We have presented a new clustering method based on low-rank approximation with two major contributions: 1) a clusterability criterion which can be used for learning both cluster assignments and the number of clusters; 2) a relaxed formulation with novel low-rank doubly stochastic matrix decomposition which allows efficient optimization, as well as its multiplicative majorization-minimization algorithm. Experimental results showed that our method works robustly for various selected data sets and can substantially improve clustering accuracy for large manifold data sets.

There are also some other generic characteristics which affect clustering performance. In the learning objective, there is the possibility of using other information divergences as the approximation error measure, including the matrix-wise and non-separable divergences (e.g., Cichocki et al., 2009; Dhillon and Tropp, 2007; Dikmen et al., 2015). In optimization, currently the multiplicative algorithm runs in batch mode. In the future we aim to develop even more scalable implementations such as streaming mini-batches of similarities and distributed computing. In this way DCD can be applicable to even bigger data sets and further improve clustering accuracy. In implementation, our practice indicates that initialization could play an important role because most current algorithms are only local optimizers. Using Dirichlet prior is only one way to smooth the objective function space. It is an open question whether other priors or regularization techniques could in general achieve better initializations.

Acknowledgments

This work was financially supported by the Academy of Finland (Finnish Centre of Excellence in Computational Inference Research COIN, grant no. 251170; Zhirong Yang additionally by decision number 140398). We acknowledge the computational resources provided by the Aalto Science-IT project.

Appendix A. Proof of Theorem 1

Proof 1) Given a matrix $B \in \mathbb{B}$ and its corresponding W , let $U_{ik} = W_{ik}/\sqrt{\sum_{v=1}^N W_{vk}}$. Then $B = UU^T$ and

$$\sum_{j=1}^N B_{ij} = \sum_{j=1}^N \sum_{k=1}^N \frac{W_{ik}W_{jk}}{\sum_{v=1}^N W_{vk}} = \sum_{k=1}^N \frac{W_{ik} \sum_{j=1}^N W_{jk}}{\sum_{v=1}^N W_{vk}} = \sum_k W_{ik} = 1. \quad (20)$$

That is, $B \in \mathbb{A}$. Therefore $\mathbb{B} \subseteq \mathbb{A}$.

2) Given a matrix $A \in \mathbb{A}$ and its corresponding U , let $W = UE$, where E is diagonal and $E_{kk} = \sum_{v=1}^N U_{vk}$. Using $\sum_{j=1}^N A_{ji} = 1$, we have

$$1 = \sum_{j=1}^N A_{ji} \quad (21)$$

$$= \sum_{j=1}^N (UU^T)_{ji} \quad (22)$$

$$= \sum_{j=1}^N (UE^{-1}EU^T)_{ji} \quad (23)$$

$$= \sum_{j=1}^N \sum_{k=1}^r (UE^{-1})_{jk} (EU^T)_{ki} \quad (24)$$

$$= \sum_{j=1}^N \sum_{k=1}^r \frac{U_{jk}}{\sum_{v=1}^N U_{vk}} W_{jk} \quad (25)$$

$$= \sum_{k=1}^r \frac{\sum_{j=1}^N U_{jk}}{\sum_{v=1}^N U_{vk}} W_{jk} \quad (26)$$

$$= \sum_{k=1}^r W_{jk}. \quad (27)$$

Using $\sum_{i=1}^N W_{ik} = \sum_{i=1}^N U_{ik} \sum_{v=1}^N U_{vk}$, we have

$$A_{ij} = \sum_{k=1}^r U_{ik} U_{jk} = \sum_{k=1}^r \frac{W_{ik}}{\sum_{l=1}^N U_{lk}} \frac{W_{jk}}{\sum_{v=1}^N U_{vk}} = \sum_{k=1}^r \frac{W_{ik} W_{jk}}{\sum_{v=1}^N W_{vk}} \quad (28)$$

That is, $A \in \mathbb{B}$. Therefore $\mathbb{B} \supseteq \mathbb{A}$. \blacksquare

Appendix B. Proof of Theorem 2

Proof We use W and \widetilde{W} to distinguish the current estimate and the variable, respectively. (Majorization)

$$\text{Let } \phi_{ijk} = \frac{W_{ik} W_{jk}}{\sum_{v=1}^N W_{vk}} \left(\sum_{l=1}^r \frac{W_{il} W_{jl}}{\sum_{v=1}^N W_{vl}} \right)^{-1}.$$

$$\mathcal{L}(\widetilde{W}, \lambda) \leq - \sum_{i=1}^N \sum_{j=1}^r \sum_{k=1}^r S_{ij} \phi_{ijk} \left[\log \widetilde{W}_{ik} + \log \widetilde{W}_{jk} - \log \sum_v \widetilde{W}_{vk} \right] \quad (29)$$

$$- (\alpha - 1) \sum_{i=1}^N \sum_{k=1}^r \log \widetilde{W}_{ik} + \sum_{i=1}^N \sum_{k=1}^r \lambda_i \widetilde{W}_{ik} + C_1 \quad (30)$$

$$\leq - \sum_{i=1}^N \sum_{j=1}^r \sum_{k=1}^r S_{ij} \phi_{ijk} \left[\log \widetilde{W}_{ik} + \log \widetilde{W}_{jk} - \frac{\sum_{v=1}^N \widetilde{W}_{vk}}{\sum_{v=1}^N W_{vk}} \right] \quad (31)$$

$$- (\alpha - 1) \sum_{i=1}^N \sum_{k=1}^r \log \widetilde{W}_{ik} + \sum_{i=1}^N \sum_{k=1}^r \lambda_i \widetilde{W}_{ik} + C_2 \quad (32)$$

$$\leq - \sum_{i=1}^N \sum_{j=1}^r \sum_{k=1}^r S_{ij} \phi_{ijk} \left[\log \widetilde{W}_{ik} + \log \widetilde{W}_{jk} - \frac{\sum_{v=1}^N \widetilde{W}_{vk}}{\sum_{v=1}^N W_{vk}} \right] \quad (33)$$

$$- (\alpha - 1) \sum_{i=1}^N \sum_{k=1}^r \log \widetilde{W}_{ik} + \sum_{i=1}^N \sum_{k=1}^r \lambda_i \widetilde{W}_{ik} \quad (34)$$

$$+ \sum_{i=1}^N \sum_{k=1}^r \left(\frac{1}{a_i} + \frac{\alpha}{W_{ik}} \right) W_{ik} \left(\frac{\widetilde{W}_{ik}}{W_{ik}} - \log \frac{\widetilde{W}_{ik}}{W_{ik}} - 1 \right) + C_2 \quad (35)$$

$$\stackrel{\text{def}}{=} G(\widetilde{W}, W), \quad (36)$$

where

$$C_1 = \sum_{i=1}^N \sum_{j=1}^r \sum_{k=1}^r S_{ij} \phi_{ijk} \log \phi_{ijk}, \quad (37)$$

$$C_2 = C_1 + \sum_{i=1}^N \sum_{j=1}^r \sum_{k=1}^r S_{ij} \phi_{ijk} \left(\log \sum_{v=1}^N W_{vk} - 1 \right) \quad (38)$$

are constants irrelevant to the variable \widetilde{W} . The first two inequalities follow the CCCP majorization (Yang and Oja, 2011) using the convexity and concavity of $-\log(\cdot)$ and $\log(\cdot)$, respectively. The third inequality is called ‘‘moving term’’ technique used in multiplicative updates (Yang and Oja, 2010). It adds the same constant $\frac{1}{a_i} + \frac{\alpha}{W_{ik}}$ to both numerator and denominator in order to guarantee that the updated matrix entries are positive, which is implemented by using the inequality $x \geq 1 + \log x$ for $x > 0$. All the above upper bounds are tight at $\widetilde{W} = W$, i.e. $G(\widetilde{W}, W) = \mathcal{L}(\widetilde{W}, \lambda)$.

(Minimization)

$$\frac{\partial G}{\partial \widetilde{W}_{ik}} = \nabla_{ik}^+ - \frac{\alpha}{\widetilde{W}_{ik}} - \frac{W_{ik}}{\widetilde{W}_{ik}} \left(\nabla_{ik}^- - \frac{\alpha}{\widetilde{W}_{ik}} \right) \tag{39}$$

$$+ \lambda_i + \left(\frac{1}{a_i} + \frac{\alpha}{\widetilde{W}_{ik}} \right) W_{ik} \left(\frac{1}{\widetilde{W}_{ik}} - \frac{1}{\widetilde{W}_{ik}} \right) \tag{40}$$

$$= -\frac{W_{ik}}{\widetilde{W}_{ik}} \left(\nabla_{ik}^- + \frac{1}{a_i} \right) + \left(\nabla_{ik}^+ + \frac{b_i}{a_i} \right). \tag{41}$$

Setting the gradient to zero gives

$$W_{ik}^{\text{new}} = W_{ik} \frac{\nabla_{ik}^- + \frac{1}{a_i}}{\nabla_{ik}^+ + \frac{b_i}{a_i}} \tag{42}$$

Multiplying both numerator and denominator by a_i gives the last update rule in Algorithm 1. Therefore, $\mathcal{L}(W^{\text{new}}, \lambda) \leq G(W^{\text{new}}, W) \leq \mathcal{L}(W, \lambda)$. ■

References

M. Ackerman and S. Ben-David. Clusterability: A theoretical study. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1–8, 2009.

D. Aloise, A. Deshpande, P. Hansen, and P. Papat. NP-hardness of euclidean sum-of-squares clustering. *Machine Learning*, 75(2):245–248, 2009.

R. Arora, M. Gupta, A. Kapila, and M. Fazel. Clustering by left-stochastic matrix factorization. In *International Conference on Machine Learning (ICML)*, pages 761–768, 2011.

R. Arora, M. Gupta, A. Kapila, and M. Fazel. Similarity-based clustering by left-stochastic matrix factorization. *Journal of Machine Learning Research*, 14:1715–1746, 2013.

A. Asuncion, M. Welling, P. Smyth, and Y. Teh. On smoothing and inference for topic models. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 27–34, 2009.

M. Balcan, A. Blum, and S. Vempala. A discriminative framework for clustering via similarity functions. In *ACM symposium on Theory of Computing*, pages 671–680, 2008.

M. Balcan, A. Blum, and A. Gupta. Approximate clustering without the approximation. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1068–1077, 2009.

S. Ben-David, N. Eiron, and H.U. Simon. The computational complexity of densest region detection. *Journal of Computer and System Sciences*, 64(1):22–47, 2002.

Y. Bishop, S. Fienberg, and P. Holland. *Discrete Multivariate Analysis: Theory and Practice*. MIT Press, 1975.

D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2001.

X. Bresson, T. Laurent, D. Uminsky, and J. von Brecht. Multiclass total variation clustering. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1421–1429, 2013.

D. Cai and X. Chen. Large scale spectral clustering via landmark-based sparse representation. *IEEE Transactions on Cybernetics*, 45(8):1669–1680, 2015.

T. Calinski and J. Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27, 1974.

X. Chen and D. Cai. Large scale spectral clustering with landmark-based representation. In *Conference on Artificial Intelligence (AAAI)*, 2011.

A. Cichocki, R. Zdunek, A. Phan, and S. Amari. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis*. John Wiley, 2009.

D. Davies and D. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):224–227, 1979.

I. Dhillon and J. Tropp. Matrix nearness problems with bregman divergences. *SIAM Journal on Matrix Analysis and Applications*, 29(4):1120–1146, 2007.

O. Dikmen, Z. Yang, and Erkki Oja. Learning the information divergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(7):1442–1454, 2015.

C. Ding, X. He, and H. Simon. On the equivalence of nonnegative matrix factorization and spectral clustering. In *SIAM International Conference on Data Mining*, 2005.

C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *International conference on Knowledge discovery and data mining (SIGKDD)*, pages 126–135, 2006.

C. Ding, T. Li, and M. Jordan. Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding. In *International Conference on Data Mining (ICDM)*, pages 183–192, 2008.

C. Ding, T. Li, and M. Jordan. Convex and semi-nonnegative matrix factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):45–55, 2010.

E. Elhamifar and R. Vidal. Sparse subspace clustering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

E. Elhamifar and R. Vidal. Sparse manifold clustering and embedding. In *Advances in Neural Information Processing Systems (NIPS)*, pages 55–63, 2011.

E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013.

- S. Eptel, M. Krishnamoorthy, and M. Zaki. Clusterability detection and initial seed selection in large data sets. Technical report, The International Conference on Knowledge Discovery in Databases, 1999.
- C. Févotte and Jérôme Idier. Algorithms for nonnegative matrix factorization with the β -divergence. *Neural Computation*, 23(9):2421–2456, 2011.
- Z. He, S. Xie, R. Zdunek, G. Zhou, and A. Cichocki. Symmetric nonnegative matrix factorization: Algorithms and applications to probabilistic clustering. *IEEE Transactions on Neural Networks*, 22(12):2117–2131, 2011.
- M. Hein and T. Bühler. An inverse power method for nonlinear eigenproblems with applications in 1-spectral clustering and sparse pca. In *Advances in Neural Information Processing Systems (NIPS)*, pages 847–855, 2010.
- T. Hofmann. Probabilistic latent semantic indexing. In *International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 50–57, 1999.
- B. Hopkins and J. Skellam. A new method for determining the type of distribution of plant individuals. *Annals of Botany*, 18(2):213–227, 1954.
- D. Hunter and K. Lange. A tutorial on MM algorithms. *The American Statistician*, 58(1): 30–37, 2004.
- B. Kulis. Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5(4): 287–364, 2013.
- D. Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.
- D. Lee and H. Seung. Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems (NIPS)*, 13:556–562, 2001.
- T. Liu and D. Tao. On the performance of manhattan nonnegative matrix factorization. *IEEE Transactions on Neural Networks and Learning Systems*, 27(9):1851–1863, 2016.
- W. Liu, J. He, and S. Chang. Large graph construction for scalable semi-supervised learning. In *International Conference on Machine Learning (ICML)*, pages 679–686, 2010.
- S. Lloyd. Least square quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.
- M. Mahajan, P. Nimbhorkar, and K. Varadarajan. The planar k-means problem is np-hard. In *Lecture Notes in Computer Science*, volume 5431, pages 274–285. Springer, 2009.
- T. Minka. Estimating a Dirichlet distribution, 2000.
- M. Muja and D. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2227–2240, 2014.
- A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems (NIPS)*, pages 849–856, 2001.
- R. Ostrovsky, Y. Rabani, L. Schulman, and C. Swamy. The effectiveness of Lloyd-type methods for the k-means problem. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 165–176, 2006.
- F. Pompili, N. Gillis, P. Absil, and F. Glineur. ONP-MF: An orthogonal nonnegative matrix factorization algorithm with application to clustering. In *European Symposium on Artificial Neural Networks*, pages 297–302, 2013.
- A. Rodriguez and A. Lajo. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014.
- J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.
- R. Sinkhorn and P. Knopp. Concerning non-negative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21:343–348, 1967.
- J. Sinkkonen, J. Aukia, and S. Kaski. Component models for large networks. ArXiv e-prints, 2008.
- N. Srivastava, R. Salakhutdinov, and G. Hinton. Modeling documents with deep Boltzmann machines. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2013.
- R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B*, 63, Part 2: 411–423, 2001.
- N. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. *Journal of Machine Learning and Research*, 11:2837–2854, 2010.
- M. Vladymyrov and M. Carreira-Perpiñán. Entropic affinities: properties and efficient numerical computation. In *International Conference on Machine Learning (ICML)*, pages 477–485, 2013.
- F. Wang, P. Li, A. König, and M. Wan. Improving clustering by learning a bi-stochastic data similarity matrix. *Knowledge Information Systems*, 32(2):351–382, 2012.
- Z. Yang and J. Laaksonen. Multiplicative updates for non-negative projections. *Neurocomputing*, 71(1-3):363–373, 2007.
- Z. Yang and E. Oja. Linear and nonlinear projective nonnegative matrix factorization. *IEEE Transaction on Neural Networks*, 21(5):734–749, 2010.
- Z. Yang and E. Oja. Unified development of multiplicative algorithms for linear and quadratic nonnegative matrix factorization. *IEEE Transactions on Neural Networks*, 22(12):1878–1891, 2011.

- Z. Yang and E. Oja. Clustering by low-rank doubly stochastic matrix decomposition. In *International Conference on Machine Learning (ICML)*, pages 831–838, 2012a.
- Z. Yang and E. Oja. Quadratic nonnegative matrix factorization. *Pattern Recognition*, 45(4):1500–1510, 2012b.
- Z. Yang, Z. Yuan, and J. Laaksonen. Projective non-negative matrix factorization with applications to facial image processing. *International Journal on Pattern Recognition and Artificial Intelligence*, 21(8):1353–1362, 2007.
- P. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 311–321, 1993. We have modified and used the code in <http://stevehanov.ca/blog/index.php?id=130>.
- J. Yoo and S. Choi. Orthogonal nonnegative matrix factorization: Multiplicative updates on Stiefel manifolds. In *Intelligent Data Engineering and Automated Learning (IDEAL)*, pages 140–147, 2008.
- S. Yu and J. Shi. Multiclass spectral clustering. In *IEEE International Conference on Computer Vision (ICCV)*, pages 313–319, 2003.
- R. Zass and A. Shashua. Doubly Stochastic Normalization for Spectral Clustering. In *Advances in Neural Information Processing Systems (NIPS)*, 2006.
- Z. Zhu, Z. Yang, and E. Oja. Multiplicative updates for learning with stochastic matrices. In *Scandinavian Conferences on Image Analysis (SCIA)*, pages 143–152, 2013.

A New Algorithm and Theory for Penalized Regression-based Clustering

Chong Wu*

Division of Biostatistics, University of Minnesota, Minneapolis, MN 55455, USA

WUXX0845@UMN.EDU

Sunghoon Kwon*

Department of Applied Statistics, Konkuk University, Seoul, South Korea

SHKWON0522@GMAIL.COM

School of Statistics, University of Minnesota, Minneapolis, MN 55455, USA

Xiaotong Shen

School of Statistics, University of Minnesota, Minneapolis, MN 55455, USA

XSHEN@UMN.EDU

Wei Pan†

Division of Biostatistics, University of Minnesota, Minneapolis, MN 55455, USA

WEIP@BIOSTAT.UMN.EDU

Editor: Inderjit Dhillon

Abstract

Clustering is unsupervised and exploratory in nature. Yet, it can be performed through penalized regression with grouping pursuit, as demonstrated in Pan et al. (2013). In this paper, we develop a more efficient algorithm for scalable computation and a new theory of clustering consistency for the method. This algorithm, called DC-ADMM, combines difference of convex (DC) programming with the alternating direction method of multipliers (ADMM). This algorithm is shown to be more computationally efficient than the quadratic penalty based algorithm of Pan et al. (2013) because of the former's closed-form updating formulas. Numerically, we compare the DC-ADMM algorithm with the quadratic penalty algorithm to demonstrate its utility and scalability. Theoretically, we establish a finite-sample mis-clustering error bound for penalized regression based clustering with the L_0 constrained regularization in a general setting. On this ground, we provide conditions for clustering consistency of the penalized clustering method. As an end product, we put R package *prclust* implementing PRclust with various loss and grouping penalty functions available on GitHub and CRAN.

Keywords: Alternating direction method of multipliers (ADMM), Difference of convex (DC) programming, Clustering consistency, Truncated L_1 -penalty (TLP).

1. Introduction

Clustering analysis separates a set of unlabeled data points into disparate groups, or clusters, based on some common properties of these points. It is a fundamental tool in machine learning, pattern recognition, and statistics, and has been widely applied in many fields, ranging from image processing to genetics. Clustering analysis has a long history, and, naturally, a large number of clustering methods have been developed; see Jain (2010) for an excellent overview.

Clustering analysis is regarded as unsupervised learning in absence of a class label, as opposed to supervised learning. Over the last few years, a new framework of clustering analysis has been introduced by treating it as a penalized regression problem (Pelckmans et al., 2005; Lindsten et al., 2011; Hocking et al., 2011; Pan et al., 2013; Chi and Lange, 2015) based on over-parameterization. Specifically, we parameterize p -dimensional observations, say x_i , $1 \leq i \leq n$, with its own centroid, say μ_i . Two observations are said to belong to the same cluster if their corresponding μ_i 's are equal. Then clustering analysis is formulated to identify a small subset of distinct values of these μ_i 's via solving the following optimization problem

$$\min_{\mu} \frac{1}{2} \sum_{i=1}^n \|x_i - \mu_i\|_2^2 + \lambda \mathcal{J}(\mu),$$

where λ is a nonnegative tuning parameter controlling the trade-off between the model fit and the number of clusters, and $\mathcal{J}(\mu)$ is a penalty on $\mu = (\mu_1', \dots, \mu_n')$. Perhaps due to computational simplicity, a convex $\mathcal{J}(\mu)$ has been extensively studied. For example, sum-of-norms clustering (Lindsten et al., 2011) defines $\mathcal{J}(\mu) = \sum_{j=1}^n \sum_{i < j} \|\mu_i - \mu_j\|_q$, where $\|\cdot\|_q$ is the L_q -norm. However, a convex $\mathcal{J}(\mu)$ usually yields biased parameter estimates, leading to difficulties in separating the clusters. To overcome this disadvantage, Pan et al. (2013) proposed penalized regression-based clustering (PRclust), which uses the non-convex grouped truncated lasso penalty (gTLP) $\mathcal{J}(\mu) = \sum_{i < j} \text{TLP}(\|\mu_i - \mu_j\|_2; \tau)$. Specifically, TLP is defined as $\text{TLP}(\alpha; \tau) = \min(|\alpha|, \tau)$ for a scalar α and a tuning parameter τ . It can be thought of as the L_1 -penalty for a small $|\alpha| \leq \tau$, but no further penalization for a large $|\alpha| > \tau$. One benefit of PRclust is that it can treat some complex clustering situations, for example, in the presence of non-convex clusters, in which traditional methods such as K-means break down (Pan et al., 2013).

To deal with the nonseparable and non-convex grouping penalty in μ_i 's, a quadratic penalty based algorithm (Pan et al., 2013) was developed by introducing some new parameters $\theta_{ij} = \mu_i - \mu_j$. This algorithm is relatively slow, and due to use of the quadratic penalty, the estimated centroids from the same cluster can never be exactly the same. To overcome these difficulties, we develop a novel and efficient computational algorithm called DC-ADMM, which combines the benefit of the alternating direction method of multipliers (ADMM) (Boyd et al., 2011) with that of the difference of convex (DC) method (Le Thi Hoai and Tao, 1997). As a result, DC-ADMM is much faster than the quadratic penalty based algorithm, in addition to that some estimated centroids can be exactly equal to each other when their corresponding observations come from the same cluster. As a by-product of this new method, we make R package *prclust* implementing both the quadratic penalty based algorithm and DC-ADMM available in CRAN (<https://cran.r-project.org>) and GitHub (<https://github.com/ChongWu-Bioostat/prclust>).

Clustering consistency of PRclust remains unknown, though operating characteristics of PRclust have been studied via some simulations and real data analysis (Pan et al., 2013). In the penalized regression based clustering framework, clustering consistency of some related models has been studied (Radchenko and Mukherjee, 2014; Zhu et al., 2014). For example, Radchenko and Mukherjee (2014) studied clustering consistency of another method with univariate observations; Zhu et al. (2014) extended this result to multivariate observations by assuming only two clusters. In this paper, with some distributional assumptions, we

*. These authors contributed equally.

†. WP is the corresponding author.

establish a general clustering consistency theory for a wide range of models, including PRclust as a special case. Our theory is applicable to multiple clusters and provide a finite-sample mis-clustering error bound in the absence of overlapping clusters. On this ground, we give sufficient conditions for PRclust to correctly identifying clusters in terms of the expected Hellinger loss. As a result, PRclust not only reconstructs the true clusters, but also yields optimal parameter estimation through the L_0 grouping penalty.

The remaining of this paper is organized as follows. Section 2 introduces the new DC-ADMM algorithm and discusses a stability criterion to select the tuning parameters. A simulation study is then performed to demonstrate the numerical performance of the new algorithm as compared to other methods. This is followed by a theory for accuracy of clustering in Section 3. A discussion of the results is given in Section 4. The proofs of the main results are given in an Appendix.

2. New Algorithm

To treat non-convexity more efficiently, we introduce a DC algorithm based on the ADMM, called DC-ADMM. We prove DC-ADMM yields a Karush-Kuhn-Tucker (KKT) solution, and some extensions are discussed.

2.1 DC-ADMM

DC-ADMM contains three steps: first, it rewrites the original unconstrained cost function into a constrained one and introduces some new variables to simplify optimization with respect to the non-convex grouping penalty; second, DC programming is applied to convert the non-convex optimization problem into a sequence of convex relaxations; third, each relaxed convex problem is solved by a standard ADMM. First, rewrite the PRclust cost function

$$\min_{\mu} \frac{1}{2} \sum_{i=1}^n \|x_i - \mu_i\|_2^2 + \lambda \sum_{i < j} \text{TLP}(\|\mu_i - \mu_j\|_2; \tau) \quad (1)$$

as the equivalent constrained problem

$$\min_{\mu, \theta} S(\mu, \theta) = \frac{1}{2} \sum_{i=1}^n \|x_i - \mu_i\|_2^2 + \lambda \sum_{i < j} \text{TLP}(\|\theta_{ij}\|_2; \tau)$$

$$\text{subject to } \theta_{ij} = \mu_i - \mu_j, \quad 1 \leq i < j \leq n,$$

where $\|\cdot\|_2$ is the L_2 -norm. Here, we introduce new variables $\theta_{ij} = \mu_i - \mu_j$ for the differences between the centroids and thus simplify optimization with respect to the grouping penalty.

To treat the non-convex gTLP on θ_{ij} 's, we apply DC programming (Le Thi Hoai and Tao, 1997). In particular, the cost function $S(\mu, \theta)$ is decomposed into a difference of two convex functions $S(\mu, \theta) = S_1(\mu, \theta) - S_2(\theta)$:

$$\begin{aligned} S_1(\mu, \theta) &= \frac{1}{2} \sum_{i=1}^n \|x_i - \mu_i\|_2^2 + \lambda \sum_{i < j} \|\theta_{ij}\|_2, \\ S_2(\theta) &= \lambda \sum_{i < j} (\|\theta_{ij}\|_2 - \tau)_+. \end{aligned}$$

where $(\alpha)_+$ denotes the positive part of α , which is α if $\alpha > 0$ and 0 otherwise.

Given the DC composition, we construct a sequence of upper approximations of $S(\mu, \theta)$ iteratively by replacing $S_2(\theta)$ at iteration $m + 1$ with its piecewise affine minorization

$$S_2^{(m)}(\theta) = S_2(\hat{\theta}^{(m)}) + \lambda \sum_{i < j} \left(\|\theta_{ij}\|_2 - \|\hat{\theta}_{ij}^{(m)}\|_2 \right) I\left(\|\hat{\theta}_{ij}^{(m)}\|_2 \geq \tau\right)$$

at the current estimate $\hat{\theta}^{(m)}$ from iteration m , leading to an upper convex approximating function at iteration $m + 1$:

$$\begin{aligned} S^{(m+1)}(\mu, \theta) &= \frac{1}{2} \sum_{i=1}^n \|x_i - \mu_i\|_2^2 \\ &+ \lambda \sum_{i < j} (\|\theta_{ij}\|_2) I\left(\|\hat{\theta}_{ij}^{(m)}\|_2 < \tau\right) + \lambda \tau \sum_{i < j} I\left(\|\hat{\theta}_{ij}^{(m)}\|_2 \geq \tau\right), \end{aligned} \quad (2)$$

where $I(\cdot)$ is the indicator function.

Then apply ADMM to solve the corresponding constrained convex problem at iteration $m + 1$

$$\min_{\mu, \theta} S^{(m+1)}(\mu, \theta), \quad \text{subject to } \theta_{ij} = \mu_i - \mu_j, \quad 1 \leq i < j \leq n. \quad (3)$$

ADMM solves (3) by minimizing the corresponding scaled augmented Lagrangian

$$\begin{aligned} L_{\rho}(\mu, \theta) &= \frac{1}{2} \sum_{i=1}^n \|x_i - \mu_i\|_2^2 + \lambda \sum_{i < j} (\|\theta_{ij}\|_2) I\left(\|\hat{\theta}_{ij}^{(m)}\|_2 < \tau\right) + \lambda \tau \sum_{i < j} I\left(\|\hat{\theta}_{ij}^{(m)}\|_2 \geq \tau\right) \\ &+ y' \sum_{i < j} (\theta_{ij} - (\mu_i - \mu_j)) + (\rho/2) \sum_{i < j} \|\theta_{ij} - (\mu_i - \mu_j)\|_2^2, \end{aligned} \quad (4)$$

where the dual variable y is a vector of Lagrange multipliers and ρ is a nonnegative penalty parameter. Using the scaled Lagrange multiplier $u = y/\rho$ (Boyd et al., 2011, §3.3.1), we can express ADMM as

$$\begin{aligned} \hat{\mu}_i^{k+1} &= \arg\min_{\mu_i} \frac{1}{2} \|x_i - \mu_i\|_2^2 + \frac{\rho}{2} \sum_{j>i} \|\hat{\theta}_{ij}^k - (\mu_i - \hat{\mu}_j^k) + \hat{u}_{ij}^k\|_2^2 \\ &+ \frac{\rho}{2} \sum_{j<i} \|\hat{\theta}_{ij}^k - (\mu_i - \hat{\mu}_j^{k+1}) + \hat{u}_{ij}^k\|_2^2, \\ \hat{\theta}_{ij}^{k+1} &= \arg\min_{\theta_{ij}} \begin{cases} \lambda \tau + \frac{\rho}{2} \|\theta_{ij} - (\hat{\mu}_i^{k+1} - \hat{\mu}_j^{k+1}) + \hat{u}_{ij}^k\|_2^2, & \text{if } \|\hat{\theta}_{ij}^{(m)}\|_2 \geq \tau; \\ \lambda \|\theta_{ij}\|_2 + \frac{\rho}{2} \|\theta_{ij} - (\hat{\mu}_i^{k+1} - \hat{\mu}_j^{k+1}) + \hat{u}_{ij}^k\|_2^2, & \text{if } \|\hat{\theta}_{ij}^{(m)}\|_2 < \tau; \end{cases} \\ \hat{u}_{ij}^{k+1} &= \hat{u}_{ij}^k + \hat{\theta}_{ij}^{k+1} - (\hat{\mu}_i^{k+1} - \hat{\mu}_j^{k+1}), \quad 1 \leq i < j \leq n, \end{aligned} \quad (5)$$

where k stands for step k in the standard ADMM. Using some simple algebra, we obtain the updating formula for μ as follows

$$\hat{\mu}_i^{k+1} = \frac{x_i + \rho \sum_{j>i} (\hat{\mu}_j^k + \hat{\theta}_{ij}^k + \hat{u}_{ij}^k) + \rho \sum_{j<i} (\hat{\mu}_j^{k+1} - \hat{\theta}_{ji}^k - \hat{u}_{ij}^k)}{1 + \rho(n-1)}.$$

Applying a block soft thresholding operator for the group lasso penalty (Yuan and Lin, 2006), we have

$$\hat{\theta}_{ij}^{k+1} = \begin{cases} \hat{\mu}_i^{k+1} - \hat{\mu}_j^{k+1} - \hat{u}_{ij}^k & \text{if } \|\hat{\theta}_{ij}^{(m)}\|_2 \geq \tau; \\ \text{ST}(\hat{\mu}_i^{k+1} - \hat{\mu}_j^{k+1} - \hat{u}_{ij}^k; \lambda/\rho) & \text{if } \|\hat{\theta}_{ij}^{(m)}\|_2 < \tau; \end{cases} \quad (6)$$

where $\text{ST}(\theta; \gamma) = (|\theta|_2 - \gamma)_+ \theta / \|\theta\|_2$. The convergence time of ADMM is highly related to the penalty parameter ρ . A poor selection of ρ can result in a slow convergence for the ADMM algorithm (Ghadimi et al., 2015) and thus DC-ADMM. In this paper, we fix $\rho = 0.4$ throughout for simplicity. For the subsequent relaxed convex problem (3), $\hat{\mu}^{(m+1)}$ and $\hat{\theta}^{(m+1)}$ are updated according to standard ADMM (5) until some stopping criteria, such as that both dual and primal residuals are small (Boyd et al., 2011), are met. We summarize the DC-ADMM algorithm in Algorithm 1.

Algorithm 1: DC-ADMM for penalized regression based clustering

Input : n observations $X = \{x_1, \dots, x_n\}$; tuning parameters λ, τ and ρ .

1 **Initialize:** Set $m = 0$, $\hat{u}_{ij}^{(0)} = 0$, $\hat{\mu}_i^{(0)} = x_i$ and $\hat{\theta}_{ij}^{(0)} = x_i - x_j$ for $1 \leq i < j \leq n$.

2 **while** $m = 0$ *or* $S(\hat{\mu}^{(m)}, \hat{\theta}^{(m)}) - S(\hat{\mu}^{(m-1)}, \hat{\theta}^{(m-1)}) < 0$ **do**

3 $m \leftarrow m + 1$

4 Update $\hat{\mu}^{(m)}$ and $\hat{\theta}^{(m)}$ based on (5) until convergence with a standard ADMM.

5 **end**

Output: Estimated centroids for the observations, $\hat{\mu}_1, \dots, \hat{\mu}_m$, from which a cluster label for each observation is assigned.

In Algorithm 1, for each iteration m , $\hat{\mu}_i^0 = x_i$ and $\hat{\theta}^0 = x_i - x_j$ for $1 \leq i < j \leq n$ are used as the starting values for (5); $(\hat{\mu}^{(m+1)}, \hat{\theta}^{(m+1)})$ is the limit point of the ADMM iterations in (5), or equivalently, is a minimizer of (3). $(\hat{\mu}^{(m+1)}, \hat{\theta}^{(m+1)})$ is then used to update the objective function $S^{(m+1)}(\mu, \theta)$ in (2) as a new approximation to $S(\mu, \theta)$. The process is iterated until the stopping criteria are met.

Since the cost function (3) is a sum of a differentiable and convex function and a convex penalty in θ (while $\theta^{(m)}$ is known), ADMM converges to its minimizer (Boyd et al., 2011). Then DC-ADMM's convergence in a finite number of steps follows by the facts that DC programming guarantees the decrease of the subsequent convex relaxations (2), and that $S^{(m+1)}(\mu, \theta)$ has only a finite set of possible forms across all m . Theorem 1 shows that the solution of the DC-ADMM converges to a KKT point.

Theorem 1 *In the DC-ADMM, $S(\hat{\mu}^{(m)}, \hat{\theta}^{(m)})$ converges in a finite number of steps; that is, there exists an $m^* < \infty$ with*

$$S(\hat{\mu}^{(m)}, \hat{\theta}^{(m)}) = S(\hat{\mu}^{(m^*)}, \hat{\theta}^{(m^*)}) \quad \text{for } m \geq m^*$$

Furthermore, $(\hat{\mu}^{(m^)}, \hat{\theta}^{(m^*)})$ is a KKT point.*

DC-ADMM only guarantees a local instead of a global minimizer. As shown in simulations, DC-ADMM performed well in terms of clustering accuracy. This suggests that DC-ADMM typically yields a good local solution, though not necessarily global. A variant of DC algorithms called outer approximation method of Breiman and Cutler (1993) gives a global minimizer, but may converge slowly. For a large-scale problem, we prefer the present version for its faster convergence at an expense of possibly missing global solutions.

With different random starting values, DC-ADMM could yield different KKT points for the same data and parameters. However, our limited numerical experience suggests that DC-ADMM gives good solutions with our proposed starting values.

Let $N_{\text{admm}}, N_{\text{quad}}$ be the numbers of iterations for running the standard ADMM and quadratic based algorithm, respectively. The computational complexity of updating θ and μ for one time is $O(pn^2)$. Note that the complexity of DC programming is $O(1)$ and N_{admm} typically scales as $O(1/\epsilon)$, where ϵ is the tolerance (He and Yuan, 2015). Then for the DC-ADMM algorithm, the computational complexity is $O(pn^2/\epsilon)$. In contrast, based on the empirical experience, N_{quad} relates to the number of observations n and quadratic based algorithm is much slower than DC-ADMM. In practice, especially in earlier iterations, one may not want to run the ADMM updates fully until convergence to save computing time. Another trick is that for the subsequent convex relaxations, we can initialize (warm start) $\hat{\mu}^0, \hat{\theta}^0$ and $\hat{\theta}^0$ at their optimal values from the previous relaxed convex problem, which significantly reduces the number of ADMM iterations.

In the DC-ADMM, the hard constraint guarantees that we can obtain exactly some $\hat{\mu}_i - \hat{\mu}_j - \hat{\theta}_{ij} = 0$; in contrast, in the quadratic penalty based algorithm (Pan et al., 2013), due to the use of soft constraint, we cannot obtain exactly $\hat{\mu}_i - \hat{\mu}_j - \hat{\theta}_{ij} = 0$ no matter how large the finite tuning parameter is chosen. Pan et al. (2013) provided an alternative algorithm (PRclust2) to force some $\hat{\mu}_i - \hat{\mu}_j - \hat{\theta}_{ij} = 0$ by running the quadratic based algorithm several times. Although PRclust2 leads to similar clustering results as DC-ADMM in our simulations, it is on average around 10 to 30 times slower than the quadratic based algorithm and is not feasible to large data sets.

2.2 Selection of the Number of Clusters

A generalized degrees of freedom (GDF) together with generalized cross validation (GCV) was proposed for selection of tuning parameters for clustering (Pan et al., 2013). This method, while yielding good performance, requires extensive computation and specification of a hyper-parameter, perturbation size. Here, we provide an alternative by modifying a stability-based criterion (Tibshirani and Walther, 2005; Liu et al., 2016) for determining the tuning parameters.

The main idea of the method is based on cross-validation. That is, (1) randomly partition the entire data set into a training set and a test set with an almost equal size; (2) cluster the training and test sets separately via PRclust with the same tuning parameters; (3) measure how well the training set clusters predict the test clusters. To be specific, first, randomly partition the entire data set into a training set X_{tr} and a test set X_{te} with a roughly equal size. Second, apply DC-ADMM (Algorithm 1) with the same tuning parameters to X_{tr} and X_{te} , leading to the corresponding clustering assignments l_{tr} and l_{te} , respectively. Third, assign X_{te} to clusters according to l_{tr} ; that is, assign each observation

in X_{ie} to the closest cluster of X_{ir} defined by l_{ir} in terms of the Euclidean distance, with $l_{e|ir}$ the corresponding clustering assignments. Note that the distance between an observation in X_{ie} and a cluster of X_{ir} is the minimum distance between the observation and each observations in the cluster. To measure how well the training set clusters predict the test clusters, we compute the adjusted Rand index (Hubert and Arabie, 1985) between $l_{e|ir}$ and l_{ie} as the prediction strength. Recall that the adjusted Rand index ranges between 0 and 1 with a higher value indicating a higher agreement. Repeat the above process T times and calculate the average prediction strength as the mean of T different prediction strengths. This process is repeated over various tuning parameter values, obtaining their corresponding average prediction strengths, then choose the set of the tuning parameters with the maximum average prediction strength. The intuition behind this idea is that if the tuning parameters lead to a stable clustering result, then the training set clusters will be similar to the test set clusters, and hence will predict them well, leading to a high average prediction strength.

2.3 Extensions

The K-means method uses squared L_2 -norm distances to generate cluster centroids, which may be inaccurate if outliers are present (Xu et al., 2005). In contrast, K-medians uses the L_1 -norm distance and is more robust to outliers. Corresponding to modifying the K-means to K-medians, we can extend PReclust by replacing the squared L_2 -norm with the L_1 -norm loss function and estimate the centroids μ through minimizing the following cost function

$$\min_{\mu} S_{L_1}(\mu) = \frac{1}{2} \sum_{i=1}^n \|x_i - \mu_i\| + \lambda \sum_{i < j} \text{TLIP}(\|\mu_i - \mu_j\|; \tau).$$

Due to the nature of the DC-ADMM algorithm, we just need to change the updating formula for $\hat{\mu}$ and leave the remaining updating formula (5), (6) unchanged. Note that

$$\begin{aligned} \hat{\mu}_i^{k+1} &= \operatorname{argmin}_{\mu_i} \frac{1}{2} \|x_i - \mu_i\| + \frac{\rho}{2} \sum_{j > i} \|\hat{\theta}_{ij}^k - (\mu_i - \mu_j^k) + \hat{u}_{ij}^k\|_2^2 \\ &\quad + \frac{\rho}{2} \sum_{j < i} \|\hat{\theta}_{ij}^k - (\mu_i - \mu_j^{k+1}) + \hat{u}_{ij}^k\|_2^2. \end{aligned}$$

To solve the above problem, we define $v_i = x_i - \mu_i$ and simplify the cost function with the L_1 -loss:

$$\begin{aligned} \hat{\mu}_i^{k+1} &= \operatorname{argmin}_{\mu_i} \frac{1}{2} \|v_i\|_1 + \frac{\rho}{2} \sum_{j > i} \|\hat{\theta}_{ij}^k - (x_i - v_i - \mu_j^k) + \hat{u}_{ij}^k\|_2^2 \\ &\quad + \frac{\rho}{2} \sum_{j < i} \|\hat{\theta}_{ij}^k - (x_i - v_i - \mu_j^{k+1}) + \hat{u}_{ij}^k\|_2^2. \end{aligned}$$

Using simple algebra and the soft thresholding operator for lasso (Tibshirani, 1996), we obtain an updating formula as:

$$\hat{\mu}_i^{k+1} = \text{STL} \left(\frac{\sum_{j > i} (\hat{\mu}_j^k + \hat{\theta}_{ij}^k + \hat{u}_{ij}^k - x_i) + \sum_{j < i} (\hat{\mu}_j^{k+1} - \hat{\theta}_{ij}^k - \hat{u}_{ij}^k - x_i)}{n-1}, \frac{1}{2\rho(n-1)} \right) + x_i,$$

where $\text{STL}(\alpha, \gamma) = \text{sign}(\alpha)(|\alpha| - \gamma)_+$. In this case, the scalar operation on a vector is element-wise.

In addition, we can also use other penalty functions. In an appendix, we provide details of the DC-ADMM algorithm for PReclust with lasso or TLP as grouping penalty.

2.4 Simulations

Consider two overlapped convex clusters with the same spherical shape in two dimensions. Specifically, a random sample of $n = 100$ observations was generated, with 50 from a bivariate Gaussian distribution $N((0, 0)', 0.3\mathbb{I})$, while the other 50 from $N((1, 1)', 0.3\mathbb{I})$, where \mathbb{I} is the identity matrix.

For PReclust, we searched $\tau \in \{0.1, 0.2, \dots, 1\}$ and $\lambda \in \{0.01, 0.05, 0.1, 0.2, 0.3, 0.5, 0.7, 1, 1.5, 2\}$. To evaluate the performance of selecting the tuning parameters, we used the Rand index (Rand, 1971) and adjusted Rand index (Hubert and Arabie, 1985), measuring the agreement between estimated cluster and the truth with a higher value indicating a higher agreement. PReclust with the stability based criterion selecting its tuning parameters performed well: the average number of clusters was 2.63, slightly larger than the truth, $K_0 = 2$; the correspond clustering results had high degrees of agreement with the truth, as evidenced by the high indices. Table 1 shows the frequencies of the number of clusters selected by the stability criterion: for the overwhelming majority (93%), either the correct number of cluster $K_0 = 2$ was selected, or a slightly larger $K = 3$ or 4 was selected. As expected, applying the quadratic penalty based algorithm with the stability criterion yielded a similar result. GCV with GDF yielded the similar results for clustering accuracy. However, to use GCV with GDF, the user has to specify the perturbation size, a hyper-parameter. In contrast, the stability based criterion is insensitive to the repeat times T . For the simulation, the average numbers of clusters selected with $T = 10, 50$ and 100 were 2.63, 2.68 and 2.76, respectively.

Now we illustrate differences between the two algorithms. First, we demonstrate how two algorithms operated differently with respect to various values of the tuning parameter λ while τ was fixed at 0.7 (Figure 1). Note that, due to the soft constraint of the quadratic penalty based algorithm, we cannot obtain exactly $\hat{\mu}_i - \mu_j - \hat{\theta}_{ij} = 0$. Even for a sufficiently large λ , there were still quite some unequal $\hat{\mu}_{i,1}$'s, which were all remarkably close to their true values 0 or 1. In contrast, due to using the hard constraint on $\hat{\theta}_{ij} = \mu_i - \mu_j$, DC-ADMM yielded some equal estimated centroids $\hat{\mu}_{i,1}$. In this simulation, the stability based criterion tended to select the most stable tuning parameters, confirming its selecting good tuning parameters and yielding good clustering results.

Figure 2 shows the run-time of two algorithms against the number of observations n and dimension p . As a matter of fact, the DC-ADMM is much faster than the quadratic penalty

Algorithm	Stability Based Criterion			GCV with GDF		
	Freq	\hat{K}	Rand	Freq	\hat{K}	Rand
DC-ADMM	All	2.63	0.950	All	3.29	0.956
	60	2.00	0.954	39	2.00	0.958
	26	3.00	0.949	22	3.00	0.965
	7	4.00	0.945	17	4.00	0.959
	5	5.00	0.924	8	5.00	0.940
	2	6.00	0.952	12	6.00	0.947
Quadratic	All	2.70	0.951	All	2.41	0.962
						9.925

Table 1: Comparison of the tuning parameter selection criteria based on 100 simulated data sets each with 2 clusters.

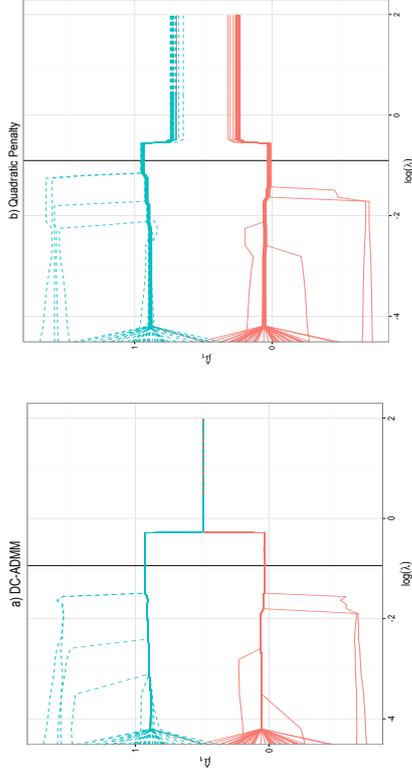


Figure 1: Solution paths of the first coordinate $\hat{\mu}_{i,1}$ for the first simulated data set. τ is fixed at 0.7. Vertical black line represents the tuning parameter selected by the stability based criterion.

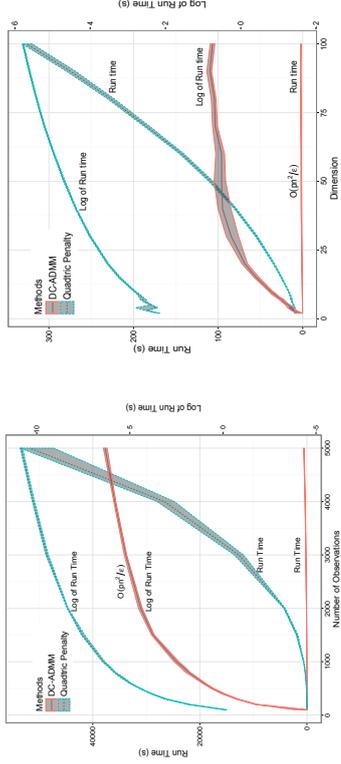


Figure 2: Comparison of run-times of DC-ADMM and quadratic penalty based algorithm based on the average of 100 simulations with different random seeds. Shaded regions represent the 25% and 75% quantiles of the run-times for corresponding algorithms. The complexity of DC-ADMM is $O(pn^2/\epsilon)$, whereas the quadratic penalty based algorithm is much slower.

based algorithm, particularly when either n or p is large. For DC-ADMM, the number of iteration was insensitive to the sample size and was around 100. In contrast, for the quadratic penalty based algorithm, it increased dramatically as the sample size increased; when the sample size was 200, the number of iteration was around 1,000; however, the number of iteration increased to around 85,000 when the sample size increased to 6,000. The complexity of DC-ADMM is quadratic in the sample size n (the ratio of run-time to n^2 was around 10^{-5}) and linear in the dimension p (the ratio of run-time to p was around 0.05), confirming that the computational complexity is $O(pn^2/\epsilon)$.

Figure 3 shows the solution paths for other methods. PRclust2 provided very similar results as DC-ADMM (Figure 3a). However, PRclust2 is extremely slow (around 10 to 30 times slower than the quadratic penalty based algorithm) and not feasible to large data sets. Convex penalties, such as the lasso and the L_2 -norm penalty, always shrink all the estimates towards zero and thus lead to severely biased parameter estimates. For example, if we used the L_2 -norm (Figure 3b) or the lasso (Figure 3c) as the grouping penalty, the estimated centroids were shrunk towards each other, leading to their convergence to the same point at the end and thus much worse performance in clustering. The TLP (Shen et al., 2012) performed much better than the lasso since it imposed no further penalty on large estimates (Figure 3d). Since the TLP does not borrow information from other variables, it performed slightly worse than its grouped version.

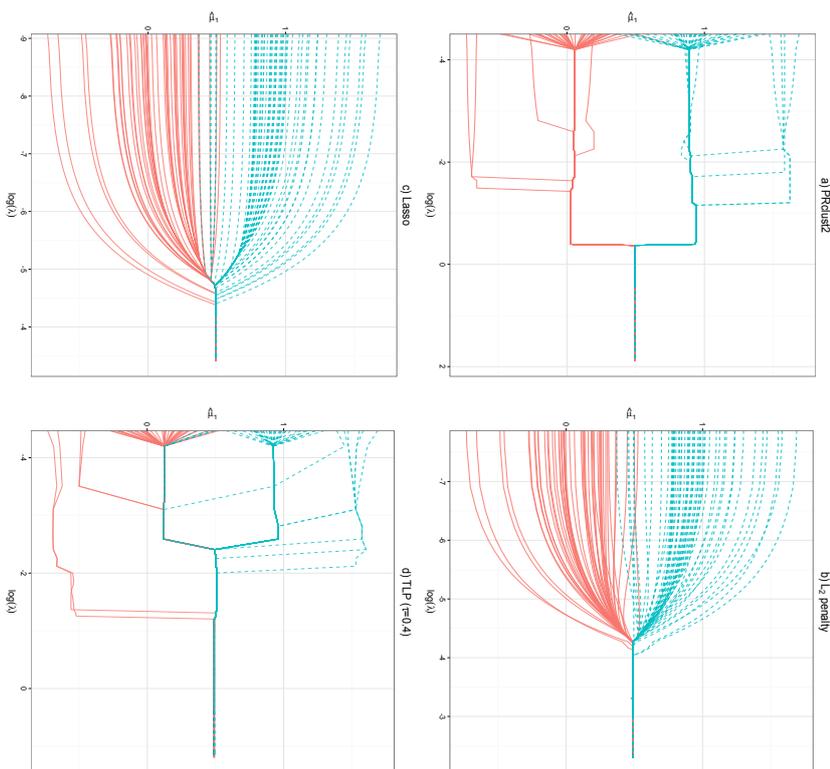


Figure 3: Solution paths of $\hat{\mu}_{i,1}$ for a) PRclust2, b) L_2 penalty, c) Lasso penalty and d) TTP for the first simulated data set.

3. Theory

Though operating characteristics of PRclust have been intensively studied, its clustering consistency properties remain unknown. In this section, based on the maximum likelihood estimation framework, we develop some theoretical properties for penalized regression based clustering method, which incorporates original PRclust (Pan et al., 2013) as a special case. Recall that PRclust does not put any distribution assumptions on the data, however, it can be treated as assuming a Gaussian distribution for the data implicitly as to be shown later. To avoid unaddressable complexity of over-parameterizing the underlying distribution, some mild technical assumptions are introduced. Then we develop a probability bound of clustering consistency which is slightly harder than clustering center consistency (Pollard, 1981).

3.1 PRclust in the Penalized Maximum Likelihood Framework

Assume $x_i \in \mathbb{R}^p \sim f_{\mu_i}(\cdot)$, $1 \leq i \leq n$ are n independent random samples, where f_{μ_i} is a probability density function of x_i with its centroid $\mu_i \in \mathbb{R}^p$. We obtain an estimate $\hat{\mu}^{L_0}$ of $\mu = (\mu_1, \dots, \mu_n)'$ $\in \mathbb{R}^{pn}$ via solving the following constrained L_0 -problem:

$$\min_{\mu} -\mathcal{L}(\mu) \quad \text{subject to} \quad \mathcal{J}(\mu) \leq J, \quad (7)$$

where J is a nonnegative tuning parameter controlling the trade-off between the model fit and the number of clusters, $\mathcal{L}(\mu) = \sum_{i=1}^n \log(f_{\mu_i}(x_i))$ is the log-likelihood that corresponds to the model fit, and $\mathcal{J}(\mu) = \sum_{i < j} I(d(\mu_i, \mu_j) \neq 0)$ is the grouping penalty that controls the number of clusters. $I(\cdot)$ is the indicator function and $d(\cdot, \cdot) : \mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}$ is a distance, which can be defined $d(\mu_i, \mu_j) = \|\mu_i - \mu_j\|_q = \{\sum_{m=1}^p |\mu_{im} - \mu_{jm}|^q\}^{1/q}$, $0 < q < \infty$. Then $\mathcal{J}(\mu)$ equals the number of distinct pairs of centroids $\mu_i \neq \mu_j$.

The regularization problem (7) is a constrained counterpart of the following penalized unconstrained L_0 -problem:

$$\min_{\mu} -\mathcal{L}(\mu) + \lambda \mathcal{J}(\mu), \quad (8)$$

where $\lambda \geq 0$ is a tuning parameter corresponding to J in (8). Note that (7) and (8) may not be equivalent in their global minimizers, which is unlike a convex problem.

In a high-dimensional situation, it is not computationally feasible to minimize a discontinuous cost function in (8) and (7). As a surrogate, we consider an estimator $\hat{\mu}^{TL_1}$ that minimizes the following truncated L_1 -problem:

$$\min_{\mu} -\mathcal{L}(\mu) + \lambda \mathcal{T}_{\tau}(\mu), \quad (9)$$

where $\mathcal{T}_{\tau}(\mu) = \sum_{i < j} \text{TTP}(d(\mu_i, \mu_j); \tau)$. Note that if assuming $x_i \sim MVN(\mu_i, \sigma^2 \mathbb{I})$, $1 \leq i \leq n$ and using L_2 -distance, we get $-\mathcal{L}(\mu) = \sum_{i=1}^n \log(f_{\mu_i}(x_i)) = \sum_{i=1}^n \log\left(\frac{1}{(2\pi)^p} \exp\left(-\frac{1}{2\sigma^2} \|x_i - \mu_i\|_2^2\right)\right)$ after ignoring some constants and $\mathcal{T}_{\tau}(\mu) = \sum_{i < j} \text{TTP}(\|\mu_i - \mu_j\|_2; \tau)$, which indicate that (9) reduces to the original PRclust (1) under multivariate Gaussian distribution assumption. When τ is sufficiently small, the truncated L_1 constraint has a good approximation to the L_0 loss (Shen et al., 2012).

3.2 A Fundamental Assumption for Over-parameterization

To reduce the unaddressable complexity to an addressable level, we propose a fundamental assumption. Let $C_k, 1 \leq k \leq K$ be K clusters that satisfy $\cup_{k=1}^K C_k = \{x_1, \dots, x_n\}$ and $C_i \cap C_j = \emptyset$ for $1 \leq i \neq j \leq K$. The number of partitions of n samples into K clusters is $(1/K!) \sum_{k=1}^K (-1)^{k-K} \binom{K}{k} k^n$, which in turn can be approximated by $K^n/K!$ (Steinley, 2006). Since PRclust is based on over-parameterization and assumes one parameter (centroid) for one corresponding sample, the complexity of PRclust is the same as all possible ways of constructing clusters based on all samples. Unfortunately, to the best of our knowledge, there is no possible probability bound that can cover this complexity that requires tail probability decreasing faster than $\exp(-n \log K)$. However, many of the clustering formulation lead to the overlapped clusters and there is no way to reconstruct the true clusters exactly. To recover non-overlapped true clusters, we put a mild technical restriction on the clustering formulation to reduce the complexity.

Assumption (A0): Partition samples x_1, x_2, \dots, x_n into K clusters. For any clusters C_1, C_2, \dots, C_K , there exists m points $y_1^{(k)}, \dots, y_m^{(k)} \in C_k$ such that $d(\bar{y}_m^{(k)}, x_k) \leq d(\bar{y}_m^{(k)}, x_c)$ for all $x_k \in C_k$ and $x_c \in C_c^c$, where $\bar{y}_m^{(k)} = \sum_{l=1}^m y_l^{(k)}/m$ and A^c denotes the complement of a set A . We define m as the *minimal disjoint centering number*.

Note that all the clusters are separated under (A0). Violating (A0) implies that there exist $x_k \in C_k$ and $x_c \in C_c^c$ such that $d(\bar{y}_m^{(k)}, x_k) > d(\bar{y}_m^{(k)}, x_c)$, indicating that there exists another cluster that overlaps with C_k . Interestingly, assumption of this kind seems necessary because clustering consistency is impossible when some clusters overlap, although it appears strong. Worth of note is that other papers, for instance Zhu et al. (2014), explicitly assume that different clusters are reasonably separable from each other for clustering consistency. Furthermore, (A0) excludes any irregular cluster structures, which are not constructed. Most importantly, Lemma 1 in the Appendix gives an upper bound of the number of ways of reconstructable clusters under (A0), reducing the overparameterization complexity from the super-exponential level in the sample size n , $\exp(-n \log K)$, to a polynomial level in n , $\exp(-mK \log n)$. Lastly, (A0) implies that all the clusters must include at least m samples, and guarantees cluster-center consistency asymptotically: for each $1 \leq k \leq K$, $\|\bar{y}_m^{(k)} - \mu_k\|_2 \rightarrow 0$ almost surely as $m \rightarrow \infty$, where μ_k is the centroid of the cluster C_k . Note that Pollard (1981) used a similar assumption for cluster-center consistency for the k -means method.

3.3 Clustering Consistency for L_0 -constrained Problem

Define $\mathcal{C} = \{\mathcal{C}(\mu) : \mu \in \mathbb{R}^{pn}\}$, where $\mathcal{C}(\mu) = \{C_1, \dots, C_K\}$ is a set of clusters based on μ such that for any cluster C_k , $d(\mu_i, \mu_j) = 0$, $\forall i, j \in C_k$ and $d(\mu_i, \mu_j) \neq 0$, $\forall i \in C_k, j \in C_k^c$. Let $\mu^* = (\mu_1^*, \dots, \mu_n^*) \in \mathbb{R}^{pn}$ with $\mu_i^* = (\mu_{i_1}^*, \dots, \mu_{i_p}^*) \in \mathbb{R}^p$ be the true centroid. We study asymptotic properties of $\hat{\mu}^{L_0}$ in (7) by giving a bound of the incorrect clustering probability: $P(\hat{\mu}^{L_0} \neq \mu^o)$, where $\mu^o = (\mu_{i_1}^o, \dots, \mu_{i_p}^o) = \text{argmin}_{\mathcal{C}(\mu) \in \mathcal{C}(\mu^*)} \mathcal{L}(\mu)$ is the oracle estimator that is usually unavailable unless the true clusters are known beforehand. Note that $\hat{\mu}^{L_0}$ is defined as a global minimizer of (7) and assume to be any global minimizer.

Before proceeding, we define a complexity measure for a given function space \mathcal{F} . For any $\epsilon > 0$, let $H(\epsilon, \mathcal{F})$ be the logarithm of the cardinality of the ϵ -bracketing of \mathcal{F} of the

smallest size. To be specific, let $S(\epsilon, \mathcal{F}, r) = \{f_1^r, f_1^r, \dots, f_r^r, f_r^r\}$ be the bracket covering of \mathcal{F} that satisfies $\max_{i \leq j \leq r} \|f_j^r - f_i^r\|_2 \leq \epsilon$, where $\|f\|_2 = (\int f^2 dv)^{1/2}$ and there exists a j such that $f_j^r \leq f \leq f_j^r$ for any $f \in \mathcal{F}$, then $H(\epsilon, \mathcal{F}) = \log(\min\{r : S(\epsilon, \mathcal{F}, r)\})$. For more discussions about metric entropy of this type, see Kolmogorov and Tikhomirov (1959). To construct the clustering consistency properties, we need the following two assumptions.

Assumption (A1): There exists some constant $d_0 > 0$ such that, for any $\epsilon > 0$,

$$\sup_{C \in \mathcal{C}; |C| \leq \mathcal{C}(\mu^*)} H(t, \mathcal{F}_C) \leq d_0 m \log(n) \log(\epsilon^2/2^8 t), \quad \epsilon^2/2^8 < t < 2^{1/2} \epsilon \leq 1,$$

where $\mathcal{F}_C = \{f_\mu : h_n^2(f_\mu, f_{\mu^*}) \leq \epsilon^2, \mu \in \mathcal{B}_C\}$, f_μ is the density of $x = (x_1', \dots, x_n')$, $\mathcal{B}_C = \{\mu : \mathcal{C}(\mu) = C\}$, $|A|$ is the cardinality of a set A and m is the *minimal disjoint centering number* defined in (A0).

Note that (A1) puts some constraints on the size of parameter space, which is similar as Assumption A in Shen et al. (2012) and is a direct modification of the assumption in Wong and Shen (1995).

Define

$$\mathcal{C}_{\min}(\mu^*) \equiv \inf_{\mu \in \mathcal{B}} \frac{h_n^2(f_\mu, f_{\mu^*})}{|\mathcal{C}(\mu)|}$$

to be the degree-of-separation or the level of difficulty of clustering, where $\mathcal{B} = \{\mu : \mathcal{C}(\mu) \neq \mathcal{C}(\mu^*), \mathcal{J}(\mu) \leq \mathcal{J}(\mu^*)\}$ is a parameter space of interest, $h_n(f_\mu, f_{\mu^*}) = \sum_{i=1}^n h(f_{\mu_i}, f_{\mu_i^*})/n^{1/2}$ is the averaged Hellinger metric with $h(f_{\mu_i}, f_{\mu_i^*}) = \left\{ \frac{1}{2} \int (f_{\mu_i}^{1/2} - f_{\mu_i^*}^{1/2})^2 dv \right\}^{1/2}$.

Assumption (A2): There exists some constant $d_1 > 0$ such that

$$\mathcal{C}_{\min}(\mu^*) > d_1 m \log(n)/n,$$

where m is the *minimal disjoint centering number* defined in (A0).

Assumption (A2) describes the least favorable situation through $\mathcal{C}_{\min}(\mu^*)$ under which we can identify the true cluster partition. In fact, $\mathcal{C}_{\min}(\mu^*)$ depends on the number of true clusters and the minimum distance among true cluster centers induced by the Hellinger loss. Since (A2) puts some regularity conditions on log-likelihood function via Hellinger loss, we do not make any regularity conditions for the log-likelihood function explicitly. Similar assumptions as (A2) can be found in the literature of feature selection. For example, Shen et al. (2012) assumed

$$\mathcal{C}_{\min}(\beta^*) \geq d_0 \log(p)/n, \quad (10)$$

where β^* is a true parameter vector of interest, d_0 is a positive constant, p is the dimension of β and n is the sample size. By assuming a probabilistic model such as the Gaussian distribution, (10) can be further specified as

$$\gamma_{\min}^2 = \min_{j: \beta_j^* \neq 0} |\beta_j^*| \geq d_0 \log(p)/n,$$

which implies the feature selection consistency can be constructed even when the minimal signal size is vanishing $\gamma_{\min} \rightarrow 0$ and the dimension of features is diverging $p \rightarrow \infty$. This assumption is much weaker than classical assumptions where γ_{\min} and p are usually fixed constants.

(A2) serves similar roles for clustering consistency as (10) for feature selection consistency. As to be shown later in Proposition 1, by assuming the Gaussian distribution, (A2) can be explicitly specified. More importantly, it allows the minimum distance among different cluster centroids decreases toward zero, $\alpha_{\min} = \min_{\mu_i^* \neq \mu_j^*} \|\mu_i^* - \mu_j^*\|^2 \rightarrow 0$, and the number of cluster diverge to infinity, $K \rightarrow \infty$, indicating that the assumption used here is weaker than many other studies where α_{\min} and K are usually fixed constants (Pollard, 1981; Pelckmans et al., 2005; Radchenko and Mukherjee, 2014; Zhu et al., 2014). Then we establish the main theory for clustering consistency as follows.

Theorem 2 Under Assumptions (A0) to (A2), if $J = \mathcal{J}(\mu^*)$, then, there exists some constant $c_3 > 0$, such that

$$P(\hat{\mu}^{l_0} \neq \hat{\mu}^0) \leq \exp(-c_2 n C_{\min}(\mu^*) + (n+1) \log(n) + 2),$$

provided that $d_1 > \max\{1/c_2, 2d_0(\log c_3)/c_1^2\}$. For example, we may use $c_2 = 4/(27 \times 1926)$, $c_3 = 10$ and $c_4 = (2/3)^{5/2}/512$. Further, $\hat{\mu}^{l_0}$ reconstructs the oracle estimator $\hat{\mu}^0$ with probability tending to one as $n \rightarrow \infty$. The following two asymptotic results hold as $n \rightarrow \infty$:

- (A) (Clustering consistency) $P(\hat{\mu}^{l_0} \neq \hat{\mu}^0) \rightarrow 0$ and hence $P(C(\hat{\mu}^{l_0}) \neq C(\hat{\mu}^*)) \rightarrow 0$.
(B) (Optimal parameter estimation) $E[h_n^2(f_{\hat{\mu}^{l_0}}, f_{\mu^*})] = (1+o(1))E[h_n^2(f_{\hat{\mu}^0}, f_{\mu^*})]$, provided that $c_2 n C_{\min}(\mu^*) + \log(E[h_n^2(f_{\hat{\mu}^0}, f_{\mu^*})]) \rightarrow \infty$.

Theorem 2 says that, under Assumptions (A0) to (A2), $\hat{\mu}^{l_0}$ consistently reconstructs the oracle estimator $\hat{\mu}^0$, and both an oracle clustering and an optimal parameter estimation with respect to expected Hellinger risk are asymptotically available by solving a constrained L_0 -problem. As pointed out by a reviewer, the number of clusters K is an important but unknown tuning parameter. Theorem 2 shows that if the tuning parameter J is chosen to be $J = |\mathcal{J}(\mu^*)|$ then optimal clustering can be constructed asymptotically. We believe that theory established here can be a starting point in developing some new tuning parameter selection criteria, though we have not fully explored in this aspect here. Theorem 2 provides an insight into or gives theoretical justification on when or under which condition the proposed method is expected to give correct clustering. For instance, the theory suggests that the optimal tuning parameters may depend on the underlying true parameters, which needs to be estimated for real data. This, together with the tuning parameter selection criterion lead to the estimated data-dependent tuning parameter for this real data set.

Although theoretical properties of penalized clustering have been intensively studied (Radchenko and Mukherjee, 2014; Zhu et al., 2014), our result is new and different from the proceeding ones. For example, Radchenko and Mukherjee (2014) proved clustering consistency with univariate samples, which are not practical and, in fact, relatively easy to prove in our context without assumption (A0) since the complexity of over-parametrization falls down to an addressable level. Zhu et al. (2014) extended the clustering consistency to multivariate samples by assuming there are only two clusters, say C_1 and C_2 with centroids μ_1 and μ_2 , respectively. To avoid some technical difficulties, Zhu et al. (2014) imposed an assumption that is not required in Theorem 2: two clusters C_1 and C_2 consist proportional number of samples in the sense that $|C_1|/|C_2| \rightarrow c$, where c is a positive constant. Theorem 2 established here extended clustering consistency to a more realistic situation: multivariate samples with many clusters.

3.4 Example: Truncated Multivariate Gaussian Distributions

In this example, we give a sufficient condition for (A2) to hold asymptotically, by constructing a lower bound of $C_{\min}(\mu^*)$ in terms of the minimum center distance $\alpha_{\min} = \min_{\mu_i^* \neq \mu_j^*} \|\mu_i^* - \mu_j^*\|^2$. Let ϕ_{μ_i} , $1 \leq i \leq n$ be the multivariate Gaussian density function with mean $\mu_i \in \mathbb{R}^p$ and identity covariance matrix $I_{p \times p}$; that is, $\phi_{\mu_i}(z) = (2\pi)^{-p/2} \exp(-\|z - \mu_i\|_2^2/2)$, $z \in \mathbb{R}^p$, $1 \leq i \leq n$. For notation simplicity, we denote $\phi_{\mu_i} = \phi$ when $\mu_i = 0$. Note that it is not generally anticipated for clustering consistency under the usual Gaussian distribution assumption since the Gaussian distribution leads to overlapped clusters and violates the assumption (A0). Hence we modify the underlying distributions for the results in Theorem 2 by considering non-overlapping situations.

Consider a class of the truncated densities $\phi_{\mu_i, \alpha}$, $1 \leq i \leq n$ with a truncation level $\alpha > 0$:

$$\phi_{\mu_i, \alpha}(z) = (1/c_\alpha) \phi_{\mu_i}(z) I(\|z - \mu_i\|_2 \leq \alpha/4), \quad (11)$$

where c_α is a normalizing constant. Note that $c_\alpha = \int_{A_{\mu_i, \alpha}} \phi_{\mu_i}(z) dz = \int_{A_\alpha} \phi(z) dz = \mathbf{X}_p(\alpha/4)$, where $A_{\mu_i, \alpha} = \{z : \|z - \mu_i\|_2 \leq \alpha/4\}$, $A_\alpha = \{z : \|z\|_2 \leq \alpha/4\}$ and \mathbf{X}_p is the chi-square distribution function with p degrees of freedom. Given two mean vectors $\mu_i \neq \mu_j$, $\phi_{\mu_i, \alpha}$ does not overlap with $\phi_{\mu_j, \alpha}$ if $\|\mu_i - \mu_j\|_2 > \alpha$. Since the truncated densities $\phi_{\mu_i, \alpha}$ for $1 \leq i \leq n$ in (11) are not overlapped to each other if we take $\alpha = \alpha_{\min} = \min_{\mu_i \neq \mu_j} \|\mu_i - \mu_j\|_2$, we assume that the samples are independently distributed with true truncated densities $\phi_{\mu_i^*, \alpha_{\min}}$, $1 \leq i \leq n$ with a truncation level α_{\min} .

Now, ignoring constraints, consider the problem in (7) for minimizing the minus log-likelihood $-\mathcal{L}(\hat{\mu}) = \sum_{i=1}^n \|x_i - \hat{\mu}\|_2^2/2$ under the constraint $\mathcal{J}(\hat{\mu}) \leq J$. To derive a sufficient condition for (A2), we construct a lower bound of $C_{\min}(\mu^*)$, the level of difficulty in recovering $C(\mu^*)$. Asymptotic properties cannot be established when cluster $C_j \in \mathcal{C}(\mu)$ only shares a finite number of samples with true clusters, and thus we make the following assumption.

Assumption (A3): For any $\mu \in \mathbb{R}^{np}$, there exists m_1 such that $\inf_{C \in \mathcal{C}(\mu)} C^* \text{e}^{\mathcal{C}(\mu^*)} C \cap C^* \neq \emptyset$ $|C \cap C^*| \geq m_1$.

Proposition 1 Let $r_{\alpha_{\min}} = \{\inf_{\mu \in \mathcal{B}} \inf_{\alpha_{\min} - \|\mu_i - \mu_i^*\|_2 \leq \phi_{\alpha_{\min}}} \mathbf{X}_p(t/4)\} / (4\mathbf{X}_p(\alpha_{\min}/4))$, where \mathbf{X}_p and \mathbf{X}_p are the chi-square density and distribution functions with p degrees of freedom, respectively. Under assumptions (A0), (A1) and (A3), if $J = \mathcal{J}(\mu^*)$ then the consistency results (A) and (B) in Theorem 2 hold, provided that

$$r_{\alpha_{\min}} \alpha_{\min} \geq d_1 m_1 K^* \log(n) / m_1, \quad (12)$$

for some constants $d_1 > \max\{1/c_2, 2d_0(\log c_3)/c_1^2\}$, where $K^* = \mathcal{C}(\mu^*)$.

Proposition 1 implies that (12) is a sufficient condition for (A2) for the truncated multivariate Gaussian distributions. In low dimensional situation, α_{\min} , p and K^* may be fixed, $r_{\alpha_{\min}}$ is bounded below, which implies the clustering consistency follows when $m \log(n) / m_1 \rightarrow \infty$ as $n \rightarrow \infty$. Moreover, clustering consistency holds when $\alpha_{\min} \rightarrow 0$ and $p \rightarrow \infty$. From L'Hopital's rule, $\lim_{\alpha_{\min} \rightarrow 0} r_{\alpha_{\min}} \leq \lim_{\alpha_{\min} \rightarrow 0} \mathbf{X}_p(\alpha_{\min}/4) / (4\mathbf{X}_p(\alpha_{\min}/4)) = \infty$ for any $p \geq 3$, which implies (12) is satisfied when $m_1 \alpha_{\min} / m_1 K^* \log(n) \rightarrow \infty$ as $n \rightarrow \infty$. For example, let $K^* \log(n) = n^k$, $m = n^p$ and $m_1 = n^{h_1}$ for some positive constants k, h and h_1 , then the theorem holds provided that $\alpha_{\min} n^{h_1 - (k+h)} \rightarrow \infty$ for any $k+h < h_1 < 1$,

implying that we can recover the true clusters even when $\alpha_{\min} \rightarrow 0$ and $K^* \rightarrow \infty$ as $n \rightarrow \infty$ for any $p \geq 3$.

At first sight, a truncated multivariate Gaussian distribution is an extreme example; however, after ignoring some constants the corresponding minus log-likelihood function $-\mathcal{L}(\mu) = \sum_{i=1}^n \|x_i - \mu\|_2^2$ is used in the original PRclust. Moreover, truncated multivariate Gaussian distributions guarantee that different clusters are separated from each other; non-truncated Gaussian distributions lead to overlapping clusters, and consistency of distance-based clustering methods, including ours, is not expected as a result. For example, suppose we have n observations, $n/2$ form a Gaussian distribution $N(-0.5, 1)$, while the other $n/2$ from $N(0.5, 1)$. According to the K-means cluster center consistency theory (Pollard, 1981), the cluster centers determined by the K-means with $K = 2$ converge to $a_1 = -0.9$ and $a_2 = 0.9$, not the original clusters centers at $\mu_1 = -0.5$ and $\mu_2 = 0.5$. The reason is that all the negative observations from the second distribution/cluster are mis-clustered into the first cluster, while all positive observations from the first clusters are incorrectly assigned to the second cluster by the K-means, leading to an under-estimated center for the first cluster; similarly the over-estimation of the second cluster center can be explained. This simple example suggests that clustering consistency cannot be established when non-truncated Gaussian distributions are used in K-means. Furthermore, previous works focused on establishing clustering consistency with the distance between clusters growing at a sufficiently fast rate. For example, Zhu et al. (2014) showed that if the distance between two clusters and sample size n grow at the same rate as $n \rightarrow \infty$, then the corresponding method can separate the two clusters perfectly. In contrast, clustering consistency established here still holds when minimum distance between the cluster centroids $\alpha_{\min} \rightarrow 0$, implying that the assumptions used here are weaker than the previous ones.

4. Discussion

The proposed new algorithm DC-ADMM bears some similarity to the quadratic penalty based algorithm in terms of the cost function and using difference convex programming. However, they differ significantly in their specific formulations. Instead of using the quadratic penalty technique, we use a hard constraint and an augmented Lagrangian in DC-ADMM. Consequently, the DC-ADMM is much faster than the quadratic penalty based algorithm and can be relatively easy to be extended to other cost functions that may have some advantages for certain problems.

The theory that states some sufficient conditions for clustering consistency and optimal parameter estimation in the PRclust framework covers a much wide range of loss functions and grouping penalties, which helps us study theoretical results uniformly for some specific PRclust implementations in the future. For example, when graph information is available, by adding a constraint on the two connected nodes in the graph, we can estimate a cluster partition and grouping structure of variables simultaneously. The mis-clustering error bound and asymptotic properties of this graph-based PRclust can be obtained via a slight modification to the theory established here.

The methods can be extended in several directions. First, the convergence of the DC-ADMM algorithm is related to the penalty parameter ρ . A poor choice of ρ may result in a slow convergence for the ADMM algorithm (Ghadimi et al., 2015). One may use an

over-relaxed ADMM algorithm to speed up. Other options exist; for example, we may use different values of ρ in each iteration (Wang and Liao, 2001). Second, since the algorithm is relatively fast, it is now feasible to deal with high dimensional data, for which variable selection is necessary. In principle, we may add a new penalty into the cost function for variable selection (Pan and Shen, 2007). Third, we may modify PRclust for noisy big data. Others have developed an iterative sub-sampling approach to improve the computational efficiency of a solution path clustering and to handle noisy big data (Marchetti and Zhou, 2014). A modification of PRclust along this direction may be useful.

An R package *prclust* implementing the DC-ADMM algorithm and the quadratic penalty algorithm with various loss and penalty functions is available at GitHub (<https://github.com/Chongwu-Biostat/prclust>) and CRAN (<http://cran.r-project.org/>).

Acknowledgments

The authors thank the reviewers for helpful comments. This research is partially supported by NIH grants R01GM113250, R01HL105397 and R01HL116720, and NSF grants DMS-1415500 and DMS-1207771. CW is supported by a University of Minnesota Doctoral Dissertation Fellowship.

Appendix A.

Proof of Theorem 1. The finite termination property of DC-ADMM follows from the following three facts. First, since (2) is closed, proper and convex and the augmented Lagrangian (4) has a saddle point, the standard ADMM converges (Boyd et al., 2011). Second, by construction of $S^{(m)}(\mu, \theta)$, for each $m \in \mathbb{N}$,

$$\begin{aligned} 0 &\leq S(\hat{\mu}^{(m)}, \hat{\theta}^{(m)}) = S^{(m+1)}(\hat{\mu}^{(m)}, \hat{\theta}^{(m)}) \leq S^{(m)}(\hat{\mu}^{(m)}, \hat{\theta}^{(m)}) \\ &\leq S^{(m)}(\hat{\mu}^{(m-1)}, \hat{\theta}^{(m-1)}) = S(\hat{\mu}^{(m-1)}, \hat{\theta}^{(m-1)}), \end{aligned}$$

implying that $S(\hat{\mu}^{(m)}, \hat{\theta}^{(m)})$ decreases in m ; otherwise the algorithm stops. Note that $(\hat{\mu}^{(m)}, \hat{\theta}^{(m)})$ is the limiting point of the ADMM iterations in (5). Third, since $S^{(m+1)}(\mu, \theta)$ depends on m only through that on the indicator functions $I(\|\hat{\theta}_{ij}^{(m)}\|_2 < \tau)$, which can be either 1 or 0, $S^{(m+1)}(\mu, \theta)$ has only a finite set of possible functional forms across all m , leading to a finite number of its possible and distinct minimal values. These facts imply that DC-ADMM terminates in a finite number of iterations.

To show that $(\hat{\mu}^{(m^*)}, \hat{\theta}^{(m^*)})$ is a KKT point of $S(\mu, \theta)$, we check if the solution satisfies a local optimality of $S(\mu, \theta)$. Since the subgradient of $S(\mu, \theta)$ and $S^m(\mu, \theta)$ are the same at the minimizer (Rockafellar, 2015), we verify the following requirement:

$$x_i + \rho \sum_{j>i} (\mu_j + \theta_{ij} + u_{ij}) + \rho \sum_{j<i} (\mu_j - \theta_{ji} - u_{ij}) - (1 + \rho(n-1))\mu_i = 0; \quad (13)$$

$$\begin{aligned} \lambda b_{ij} \theta_{ij} / \|\theta_{ij}\|_2 + \rho(\theta_{ij} - (\mu_i - \mu_j) + u_{ij}) &= 0; \\ \theta_{ij} - \mu_i - \mu_j &= 0, \end{aligned} \quad (14) \quad (15)$$

where b_{ij} is the regular subdifferential of $\min(\|\theta_{ij}\|_2, \tau)$ at $\|\theta_{ij}\|_2$. Easily, (15) is the hard constraint in the DC-ADMM and is met at convergence. Note that $(\hat{\mu}^{(m^*)}, \hat{\theta}^{(m^*)}, \hat{u}^{(m^*)}) = (\hat{\mu}^{(m^*-1)}, \hat{\theta}^{(m^*-1)}, \hat{u}^{(m^*-1)})$ at termination. Then (13) is satisfied with $(\mu, \theta, u) = (\hat{\mu}^{(m^*-1)}, \hat{\theta}^{(m^*-1)}, \hat{u}^{(m^*-1)})$. For (14), consider three cases.

- If $\|\hat{\theta}_{ij}^{(m^*-1)}\|_2 > \tau$, the $\hat{\theta}_{ij}^{(m^*-1)} = \hat{\mu}_i^{(m^*)} - \hat{\mu}_j^{(m^*)} - u_{ij}^{(m^*)}$, implying $\theta_{ij} = \hat{\theta}_{ij}^{(m^*)}$ since $b_{ij} = 0$.
- If $0 < \|\hat{\theta}_{ij}^{(m^*)}\|_2 < \tau$ and $\|\hat{\mu}_i^{(m^*)} - \hat{\mu}_j^{(m^*)} - \hat{u}_{ij}^{(m^*)}\|_2 > \lambda/\rho$, then

$$\hat{\theta}_{ij}^{(m^*)} = \left(\|\hat{\mu}_i^{(m^*)} - \hat{\mu}_j^{(m^*)} - \hat{u}_{ij}^{(m^*)}\|_2 - \frac{\lambda}{\rho} \right) \frac{\hat{\mu}_i^{(m^*)} - \hat{\mu}_j^{(m^*)} - \hat{u}_{ij}^{(m^*)}}{\|\hat{\mu}_i^{(m^*)} - \hat{\mu}_j^{(m^*)} - \hat{u}_{ij}^{(m^*)}\|_2},$$

hence that $\|\hat{\theta}_{ij}^{(m^*)}\|_2 = \|\hat{\mu}_i^{(m^*)} - \hat{\mu}_j^{(m^*)} - \hat{u}_{ij}^{(m^*)}\|_2 - \frac{\lambda}{\rho}$. Then (14) is met when $\theta_{ij} = \hat{\theta}_{ij}^{(m^*)}$ since $b_{ij} = 1$.

- If $0 < \|\hat{\theta}_{ij}^{(m^*)}\|_2 < \tau$ and $\|\hat{\mu}_i^{(m^*)} - \hat{\mu}_j^{(m^*)} - \hat{u}_{ij}^{(m^*)}\|_2 < \lambda/\rho$, then $\|\hat{\theta}_{ij}^{(m^*)}\|_2 = 0$, which contradicts to the fact that $0 < \|\hat{\theta}_{ij}^{(m^*)}\|_2 < \tau$.

This completes the proof. \blacksquare

Lemma 1. Given n observations $x_i \in \mathbb{R}^p$, $1 \leq i \leq n$, let the number of ways of constructing K clusters that satisfies disjoint condition (assumption A0) with the minimal disjoint centering number m be $c_{n,K,m}$. Then

$$c_{n,K,m} \leq (n-Km)^K \prod_{k=1}^K \binom{n-(k-1)m}{m}.$$

Proof of Lemma 1. Without loss of generality, we fix the first km points and form K disjoint subsets $S_k = \{x_{(k-1)m+1}, \dots, x_{km}\} \subset \{x_1, \dots, x_n\}$, $k \leq K$. Let $r_i^{(k)} = d(\bar{x}_m^{(k)}, x_i)$, $km+1 \leq i \leq n$ with $\bar{x}_m^{(k)} = \sum_{j=(k-1)m+1}^{km} x_j/m$ and $r_i^{(k)}$ be an ordered sequence of $r_i^{(k)}$ that satisfies $r_{km+1}^{(k)} \leq \dots \leq r_n^{(k)}$. Then a possible way of constructing a subset C_k based on S_k is including S_k and all the points within distance $r_i^{(k)}$. For a subset C_k , the number of constructing ways is $n - Km$ at most. Hence, the number of ways of constructing K subsets C_k , $k \leq K$ based on S_k is $(n - Km)^K$ at most.

Note that the number of ways of fixing possible K disjoint subsets S_k , $k \leq K$ is $\prod_{k=1}^K \binom{n-(k-1)m}{m}$. Hence the total number of ways of constructing K subsets along to the way described above is $(n - Km)^K \prod_{k=1}^K \binom{n-(k-1)m}{m}$ at most. Note that any cluster partition with K clusters that satisfies disjoint structure condition with the minimal disjoint centering number m can be constructed via the ways described above. Hence $c_{n,K,m} \leq (n - Km)^K \prod_{k=1}^K \binom{n-(k-1)m}{m}$. This completes the proof. \blacksquare

Proof of Theorem 2. On the set $\tilde{\mathcal{B}} = \{\mu : \mathcal{C}(\mu) = \mathcal{C}(\mu^*)\} \subset \{\mu : \mathcal{J}(\mu) \leq \mathcal{J}(\mu^*)\}$, we have $\hat{\mu}^{L_0} = \sup_{\mu \in \tilde{\mathcal{B}}} \mathcal{L}(\mu) = \hat{\mu}^0 = \sup_{\mathcal{C}(\mu) = \mathcal{C}(\mu^*)} \mathcal{L}(\mu)$. Let the parameter space of interest be $\mathcal{B} = \{\mu : \mathcal{C}(\mu) \neq \mathcal{C}(\mu^*), \mathcal{J}(\mu) \leq \mathcal{J}(\mu^*)\}$. Since $\mathcal{J}(\mu) \leq \mathcal{J}(\mu^*)$ implies $|\mathcal{C}(\mu)| \leq K^*$, we have $\mathcal{B} \subset \{\mu : \mathcal{C}(\mu) \neq \mathcal{C}(\mu^*), |\mathcal{C}(\mu)| \leq K^*\} \subset \cup_{k=1}^{K^*} \cup_{C \in \mathcal{C}_k} \mathcal{B}_C$, where $\mathcal{B}_C = \{\mu : \mathcal{C}(\mu) = C\}$ and $\mathcal{C}_k = \{C \in \mathcal{C} : C \neq \mathcal{C}(\mu^*), |C| = k\}$. Hence, using $\mathcal{L}(\hat{\mu}^0) \geq \mathcal{L}(\mu^*)$, we have

$$\begin{aligned} P(\hat{\mu}^{L_0} \neq \hat{\mu}^0) &\leq P^* \left(\sup_{\mu \in \tilde{\mathcal{B}}} \{\mathcal{L}(\mu) - \mathcal{L}(\hat{\mu}^0)\} > 0 \right) \\ &\leq P^* \left(\sup_{\mu \in \tilde{\mathcal{B}}} \{\mathcal{L}(\mu) - \mathcal{L}(\mu^*)\} > 0 \right) \\ &\leq \sum_{k=1}^{K^*} \sum_{C \in \mathcal{C}_k} P^* \left(\sup_{\mu \in \mathcal{B}_C} \{\mathcal{L}(\mu) - \mathcal{L}(\mu^*)\} > 0 \right), \end{aligned}$$

where P^* is the outer probability. Now we apply Theorem 1 of Wong and Shen (1995) to bound each term. For any $\mu \in \mathcal{B}_C$ and $C \in \mathcal{C}_k$, $h_2^2(f_\mu, f_{\mu^*}) \geq kC_{\min}(f_{\mu^*})$, there exists a constant $c_2 > 0$ such that

$$P^* \left(\sup_{\mu \in \mathcal{B}_C} \{\mathcal{L}(\mu) - \mathcal{L}(\mu^*)\} > 0 \right) \leq 4 \exp(-c_2 n k C_{\min}(f_{\mu^*})),$$

provided that the local entropy conditions are satisfied as follows: there exist constants $c_3 > 0$ and $c_4 > 0$ such that

$$\int_{\mathcal{E}^2/2^8}^{2^{1/2}\epsilon} H^{1/2}(t/c_3, \mathcal{F}_C) dt \leq c_4 n^{1/2} \epsilon^2 \quad (16)$$

for any $\epsilon^2 \geq k\mathcal{C}_{\min}(\mu^*)$. Let $\epsilon_n^2 = 2d_0 \log(c_3) m \log(n) / c_4^2 n$. Under (A1), ϵ_n solves the inequality

$$\max_{k \leq K^*} \sup_{C \in \mathcal{C}_0} \int_{\mathcal{E}^2/2^8}^{2^{1/2}\epsilon} H^{1/2}(t/c_3, \mathcal{F}_C) dt \leq (d_0 m \log(n))^{1/2} (2^{1/2}\epsilon_n \log(c_3))^{1/2} \leq c_4 n^{1/2} \epsilon_n^2$$

with respect to ϵ provided that $\epsilon_n < \epsilon$. Hence, (16) follows if $\mathcal{C}_{\min}(\mu^*) \geq \epsilon_n^2$, and from (A2), this holds when $d_1 \geq 2d_0 \log(c_3) m / c_4^2$. From Lemma 1, $|C_k| \leq (n - km)^k \prod_{j=1}^k \binom{n-j-1}{m} \leq n^{k+mk}$. Hence,

$$\begin{aligned} P(\hat{\mu}^{L_0} \neq \hat{\mu}^*) &\leq \sum_{k=1}^{K^*} 4 \exp(-c_2 n k \mathcal{C}_{\min}(\mu^*) + k(m+1) \log(n)) \\ &\leq 4R(\exp(-c_2 n \mathcal{C}_{\min}(\mu^*) + (m+1) \log(n)) \\ &\leq 5 \exp(-c_2 n \mathcal{C}_{\min}(\mu^*) + (m+1) \log(n)) \\ &\leq \exp(-c_2 n \mathcal{C}_{\min}(\mu^*) + (m+1) \log(n) + 2), \end{aligned}$$

where $R(x) = x/(1-x)$ is exponentiated logistic function.

Now (A) follows from $P(\mathcal{C}(\hat{\mu}^{L_0}) \neq \mathcal{C}(\mu^*)) \leq P(\hat{\mu}^{L_0} \neq \mu^*)$ and $d_1 > 1/c_2$. For the risk property, using $h_a^2(\hat{\mu}^{L_0}, \mu^*) \leq 1$,

$$\begin{aligned} E[h_a^2(\hat{\mu}^{L_0}, \mu^*)] &\leq E[h_a^2(\hat{\mu}^0, \mu^*)] + E[h_a^2(\hat{\mu}^{L_0}, \mu^*) I(\hat{\mu}^{L_0} \neq \hat{\mu}^0)] \\ &\leq E[h_a^2(\hat{\mu}^0, \mu^*)] + P(\hat{\mu}^{L_0} \neq \hat{\mu}^0) \\ &\leq (1 + o(1)) E[h_a^2(\hat{\mu}^0, \mu^*)] \end{aligned}$$

provided that $\exp(-c_2 n \mathcal{C}_{\min}(\mu^*)) / E[h_a^2(\hat{\mu}^0, \mu^*)] = o(1)$, and then (B) established. This completes the proof. \blacksquare

Proof of Proposition 1. It suffices to show that (12) is a sufficient condition for Assumption (A2). First, we give a lower bound of the Hellinger metric between $\phi_{\mu_i, \alpha_{\min}}$ and $\phi_{\mu_i^*, \alpha_{\min}}$ for a given $\mu \in \mathcal{B} = \{\mu : \mathcal{C}(\mu) \neq \mathcal{C}(\mu^*), \mathcal{J}(\mu) \leq \mathcal{J}(\mu^*)\}$. Let

$$A_{\alpha_{\min}} = \{z : \|z\|_2^2 \leq \alpha_{\min}/4\} \text{ and } A_{\mu_i, \alpha_{\min}} = \{z : \|z - \mu_i\|_2^2 < \alpha_{\min}/4\}, 1 \leq i \leq n.$$

Given $\mu_i \neq \mu_i^*$ with $\|\mu_i - \mu_i^*\|_2^2 \leq \alpha_{\min}$, let $\Delta_i^* = \mu_i - \mu_i^*$ then we have

$$\begin{aligned} &h^2(\phi_{\mu_i, \alpha_{\min}}, \phi_{\mu_i^*, \alpha_{\min}})^2 \\ &= h^2(\phi_{\alpha_{\min}}, \phi_{\Delta_i^*, \alpha_{\min}})^2 \\ &= \frac{1}{2} \int (\phi_{\alpha_{\min}}^{1/2}(z) - \phi_{\Delta_i^*, \alpha_{\min}}^{1/2}(z))^2 dz \\ &= \frac{1}{2\mathcal{X}_p^2(\alpha_{\min}/4)} \left(\int_{A_{\alpha_{\min}}} \phi(z) dz + \int_{A_{\Delta_i^*, \alpha_{\min}}} \phi_{\Delta_i^*}(z) dz - 2 \int_{A_{\alpha_{\min}} \cap A_{\Delta_i^*, \alpha_{\min}}} \phi(z)^{1/2} \phi_{\Delta_i^*}(z)^{1/2} dz \right) \\ &= \frac{1}{\mathcal{X}_p^2(\alpha_{\min}/4)} \left(\int_{A_{\alpha_{\min}}} \phi(z) dz - \int_{A_{\alpha_{\min}} \cap A_{\Delta_i^*, \alpha_{\min}}} \phi(z)^{1/2} \phi_{\Delta_i^*}(z)^{1/2} dz \right). \end{aligned}$$

Let $B_{\Delta_i^*, \alpha_{\min}} = \{z : \|z - \Delta_i^*\|_2^2 \leq \alpha/4 - \|\Delta_i^*\|_2^2/4\}$ then it is easy to see that

$$A_{\alpha_{\min}} \cap A_{\Delta_i^*, \alpha_{\min}} \subset B_{\Delta_i^*, \alpha_{\min}}.$$

By using the equality,

$$\phi(z)^{1/2} \phi_{\Delta_i^*}(z)^{1/2} = (2\pi)^{-p/2} \exp(-\|z - \Delta_i^*\|_2^2/2 - \|\Delta_i^*\|_2^2/8) = \exp(-\|\Delta_i^*\|_2^2/8) \phi_{\Delta_i^*/2}(z),$$

we have

$$\begin{aligned} &\int_{A_{\alpha_{\min}} \cap A_{\Delta_i^*, \alpha_{\min}}} \phi(z)^{1/2} \phi_{\Delta_i^*}(z)^{1/2} dz \leq \exp(-\|\Delta_i^*\|_2^2/8) \int_{B_{\Delta_i^*/2, \alpha_{\min}}} \phi_{\Delta_i^*/2}(z) dz \\ &= \exp(-\|\Delta_i^*\|_2^2/8) \mathcal{X}_p(\alpha_{\min}/4 - \|\Delta_i^*\|_2^2/4) \\ &\leq \mathcal{X}_p(\alpha_{\min}/4 - \|\Delta_i^*\|_2^2/4). \end{aligned}$$

According to the mean value theorem,

$$h^2(\phi_{\mu_i, \alpha_{\min}}, \phi_{\mu_i^*, \alpha_{\min}})^2 \geq \frac{\mathcal{X}_p(\alpha_{\min}/4) - \mathcal{X}_p(\alpha_{\min}/4 - \|\Delta_i^*\|_2^2/4)}{\mathcal{X}_p^2(\alpha_{\min}/4)} \geq r_{\alpha_{\min}} \|\mu_i - \mu_i^*\|_2^2, \quad (17)$$

where $r_{\alpha_{\min}} = \{\inf_{\mu \in \mathcal{B}} \inf_{\alpha_{\min} - \|\Delta_i^*\|_2^2 \leq \alpha_{\min}} \mathcal{X}_p(t/4)\} / 4\mathcal{X}_p(\alpha_{\min}/4)$. Next, we find a lower bound of $\mathcal{C}_{\min}(\mu^*)$. From (17), the inequality $h_a^2(f_\mu, f_{\mu^*}) \geq \sum_{i=1}^n h^2(f_{\mu_i}, f_{\mu_i^*})/n$ implies

$$n\mathcal{C}_{\min}(\mu^*) = n \inf_{\mu \in \mathcal{B}} h_a^2(f_\mu, f_{\mu^*}) / \mathcal{C}(\mu) \geq nr_{\alpha_{\min}} \inf_{\mu \in \mathcal{B}} \|\mu - \mu^*\|_2^2 / \mathcal{C}(\mu). \quad (18)$$

It is easy to see that $\inf_{\mu \in \{\mu : \mathcal{C}(\mu) < K^*\}} \|\mu - \mu^*\|_2^2 / \mathcal{C}(\mu) \geq \inf_{\mu \in \{\mu : \mathcal{C}(\mu) = K^*\}} \|\mu - \mu^*\|_2^2 / K^*$, since the sum of within cluster variances, $\|\mu - \mu^*\|_2^2 = \sum_{i=1}^n \|\mu_i - \mu_i^*\|_2^2$, is minimized when $|\mathcal{C}(\mu)| = K^*$. Hence, we have

$$\inf_{\mu \in \mathcal{B}} \|\mu - \mu^*\|_2^2 / \mathcal{C}(\mu) = \inf_{\mu \in \{\mu : \mathcal{C}(\mu) = K^*, \mathcal{C}(\mu) \neq \mathcal{C}(\mu^*)\}} \|\mu - \mu^*\|_2^2 / K^*. \quad (19)$$

Let $\mathcal{C}(\mu) = \{C_1, \dots, C_K\}$ and $\mathcal{C}(\mu^*) = \{C_1^*, \dots, C_{K^*}^*\}$. Since $\mathcal{C}(\mu) \neq \mathcal{C}(\mu^*)$, without loss of generality, we may assume that $C_s \cap C_t^* \neq \emptyset$ for $s, t = 1, 2$. Then the right-hand side of (19) achieves its minimum when $\mu \in \mathcal{B}_{12} = \{\mu : C_s \cap C_t^* \neq \emptyset \text{ for } s, t = 1, 2 \text{ and } \mu_i = \mu_i^* \text{ for } i \in$

$\cup_{3 \leq k \leq K^*} C_k^t$. Let $\mu_i = \nu_{s,i} \in C_s$ for $s = 1, 2$ and similarly let $\mu_i^t = \nu_i^*$, $i \in C_t^*$ for $t = 1, 2$. Then it follows that

$$\begin{aligned} \inf_{\mu \in \mathcal{B}} \|\mu - \mu^*\|_2^2 &= \inf_{\mu \in \mathcal{B}^{1,2}} \sum_{t=1,2} (n_{1t} \|\nu_1 - \nu_1^*\|_2^2 + n_{2t} \|\nu_2 - \nu_2^*\|_2^2) \\ &= \inf_{n_{s,t}; t=1,2} \sum_{t=1,2} (n_{1t} \|\nu_1^* - \nu_1^*\|_2^2 + n_{2t} \|\nu_2^* - \nu_2^*\|_2^2) \\ &= \inf_{n_{s,t}; t=1,2} \left(\frac{n_{11}n_{12}}{n_{11} + n_{12}} + \frac{n_{21}n_{22}}{n_{21} + n_{22}} \right) \|\nu_1^* - \nu_2^*\|_2^2, \end{aligned}$$

where $n_{st} = |C_s \cap C_t^*|$ for $s, t = 1, 2$ and $\nu_1^* = (n_{11}\nu_1^* + n_{12}\nu_2^*)/(n_{11} + n_{12})$ and $\nu_2^* = (n_{21}\nu_1^* + n_{22}\nu_2^*)/(n_{21} + n_{22})$ are the weighted means of ν_1^* 's and ν_2^* 's in C_1 and C_2 , respectively. From (A3), $n_{st} \geq m_1$ for $s, t = 1, 2$, which implies $n_{11}n_{12}/(n_{11} + n_{12}) = 1/(1/n_{11} + 1/n_{12}) \geq m_1/2$ and similarly, $n_{21}n_{22}/(n_{21} + n_{22}) \geq m_1/2$. Hence the lower bound becomes

$$\inf_{\mu \in \mathcal{B}} \|\mu - \mu^*\|_2^2 \geq m_1 \alpha_{\min}. \quad (20)$$

From (18), (19), (20) and definition of $C_{\min}(\mu^*)$, it is easy to see that (A2) is met if $C_{\min}(\mu^*) \geq r_{\alpha_{\min}} m_1 \alpha_{\min} / n K^* \geq d_1 m \log(n)/n$ which is equivalent to

$$r_{\alpha_{\min}} \alpha_{\min} \geq d_1 m K^* \log(n) / m_1.$$

This completes the proof. \blacksquare

Appendix B.

The cost function of PRclust with lasso grouping penalty will be convex and thus DC-ADMM is exactly same as ADMM and a global solution will be reached. Note that $\theta_{ij} = (\theta_{j1}, \dots, \theta_{jp})$, then the updating formulas can be summarized as follows:

$$\begin{aligned} \hat{\mu}_i^{(n+1)} &= \frac{x_i + \rho \sum_{j>i} \left(\hat{\mu}_j^{(n)} + \hat{\theta}_{ij}^{(n)} + \hat{u}_{ij}^{(n)} \right) + \rho \sum_{j<i} \left(\hat{\mu}_j^{(n+1)} - \hat{\theta}_{ji}^{(n)} - \hat{u}_{ij}^{(n)} \right)}{1 + \rho(n-1)}; \\ \hat{\theta}_{ij}^{(n+1)} &= \text{ST} \left(\hat{\mu}_{ij}^{(n+1)} - \hat{\mu}_{ji}^{(n+1)} - \hat{u}_{ij}^{(n)}, \lambda / \rho \right) \\ \hat{u}_{ij}^{(n+1)} &= \hat{u}_{ij}^{(n)} + \hat{\theta}_{ij}^{(n+1)} - (\hat{\mu}_{ij}^{(n+1)} - \hat{\mu}_{ji}^{(n+1)}), \quad 1 \leq i < j \leq n; i, l = 1, 2, \dots, p. \end{aligned}$$

The main difference between TLP and gTLP is that TLP is an element-wise penalty and the updating formulas (5) for PRclust with TLP can be summarized as follows, while the other part of DC-ADMM remains unchanged:

$$\begin{aligned} \hat{\mu}_i^{k+1} &= \frac{x_i + \rho \sum_{j>i} \left(\hat{\mu}_j^k + \hat{\theta}_{ij}^k + \hat{u}_{ij}^k \right) + \rho \sum_{j<i} \left(\hat{\mu}_j^{k+1} - \hat{\theta}_{ji}^k - \hat{u}_{ij}^k \right)}{1 + \rho(n-1)}; \\ \hat{\theta}_{ij}^{k+1} &= \begin{cases} \hat{\mu}_{ij}^{k+1} - \hat{\mu}_{ji}^{k+1} - \hat{u}_{ij}^k & \text{if } |\hat{\theta}_{ij}^{(m)}| \geq \tau; \\ \text{STL} \left(\hat{\mu}_{ij}^{k+1} - \hat{\mu}_{ji}^{k+1} - \hat{u}_{ij}^k; \lambda / \rho \right) & \text{if } |\hat{\theta}_{ij}^{(m)}| < \tau; \end{cases} \\ \hat{u}_{ij}^{k+1} &= \hat{u}_{ij}^k + \hat{\theta}_{ij}^{k+1} - (\hat{\mu}_{ij}^{k+1} - \hat{\mu}_{ji}^{k+1}), \quad 1 \leq i < j \leq n; i, l = 1, 2, \dots, p. \end{aligned}$$

References

- Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.
- Leo Breiman and Adele Cutler. A deterministic algorithm for global optimization. *Mathematical Programming*, 58(1-3):179–199, 1993.
- Eric C Chi and Kenneth Lange. Splitting methods for convex clustering. *Journal of Computational and Graphical Statistics*, 24(4):994–1013, 2015.
- Enhanma Ghadimi, André Teixeira, Iman Shames, and Mikael Johansson. Optimal parameter selection for the alternating direction method of multipliers (admm): quadratic problems. *IEEE Transactions on Automatic Control*, 60(3):644–658, 2015.
- Bingsheng He and Xiaoming Yuan. On non-ergodic convergence rate of Douglas-rachford alternating direction method of multipliers. *Numerische Mathematik*, 130(3):567–577, 2015.
- Toby Dylan Hoeking, Armand Joulin, Francis Bach, and Jean-Philippe Vert. Clustertpath an algorithm for clustering using convex fusion penalties. In *28th International Conference on Machine Learning (ICML)*, 2011.
- Lawrence Hubert and Phipps Arabe. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.
- Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.
- Andrei Nikolaevich Kolmogorov and Vladimir Mikhailovich Tikhomirov. ϵ -entropy and ϵ -capacity of sets in function spaces. *Uspekhi Matematicheskikh Nauk*, 14(2):3–86, 1959.
- An Le Thi Hoai and Pham Dinh Tao. Solving a class of linearly constrained indefinite quadratic problems by dc algorithms. *Journal of Global Optimization*, 11(3):253–285, 1997.
- Fredrik Lindsten, Henrik Ohlsson, and Lennart Ljung. Clustering using sum-of-norms regularization: With application to particle filter output computation. In *Statistical Signal Processing Workshop*, 2011.
- Binghui Liu, Xiaotang Shen, and Wei Pan. Integrative and regularized principal component analysis of multiple sources of data. *Statistics in Medicine*, 35(13):2235–50, 2016.
- Yuliya Marchetti and Qing Zhou. Iterative subsampling in solution path clustering of noisy big data. *arXiv preprint arXiv:1412.1559*, 2014.
- Wei Pan and Xiaotang Shen. Penalized model-based clustering with application to variable selection. *The Journal of Machine Learning Research*, 8(1):1145–1164, 2007.

- Wei Pan, Xiaotong Shen, and Binghui Liu. Cluster analysis: unsupervised learning via supervised learning with a non-convex penalty. *The Journal of Machine Learning Research*, 14(1):1865–1889, 2013.
- Kristiaan Peckmans, Joseph De Brabanter, JAK Stuykens, and B De Moor. Convex clustering shrinkage. In *PASCAL Workshop on Statistics and Optimization of Clustering Workshop*, 2005.
- David Pollard. Strong consistency of k -means clustering. *The Annals of Statistics*, 9(1):135–140, 1981.
- Peter Radchenko and Gourab Mukherjee. Consistent clustering using an l_1 fusion penalty. *arXiv preprint arXiv:1412.0753*, 2014.
- William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.
- Ralph Tyrell Rockafellar. *Convex Analysis*. Princeton university press, 2015.
- Xiaotong Shen, Wei Pan, and Yunzhang Zhu. Likelihood-based selection and sharp parameter estimation. *Journal of the American Statistical Association*, 107(497):223–232, 2012.
- Douglas Steinley. K-means clustering: a half-century synthesis. *British Journal of Mathematical and Statistical Psychology*, 59(1):1–34, 2006.
- Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288, 1996.
- Robert Tibshirani and Guenther Walther. Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics*, 14(3):511–528, 2005.
- S.L. Wang and L.Z. Liao. Decomposition method with a variable parameter for a class of monotone variational inequality problems. *Journal of Optimization Theory and Applications*, 109(2):415–429, 2001.
- Wing Hung Wong and Xiaotong Shen. Probability inequalities for likelihood ratios and convergence rates of sieve mles. *The Annals of Statistics*, 23(2):339–362, 1995.
- Rui Xu, Donald Wunsch, et al. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678, 2005.
- Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, 68(1):49–67, 2006.
- Changbo Zhu, Huan Xu, Chenlei Leng, and Shuicheng Yan. Convex optimization procedure for clustering: Theoretical revisit. In *Advances in Neural Information Processing Systems*, 2014.

Classification of Imbalanced Data with a Geometric Digraph Family

Artür Manukyan

*Graduate School of Sciences and Engineering
Koç University
Sarıyer, 34450, Istanbul, Turkey*

AMANUKYAN13@KU.EDU.TR

Elvan Ceyhan

*Department of Statistics
University of Pittsburgh
Pittsburgh, 15260, PA, USA*

ELVANCEYHAN@GMAIL.COM

Editor: William Cohen

Abstract

We use a geometric digraph family called class cover catch digraphs (CCCDs) to tackle the class imbalance problem in statistical classification. CCCDs provide graph theoretic solutions to the class cover problem and have been employed in classification. We assess the classification performance of CCCD classifiers by extensive Monte Carlo simulations, comparing them with other classifiers commonly used in the literature. In particular, we show that CCCD classifiers perform relatively well when one class is more frequent than the other in a two-class setting, an example of the *class imbalance problem*. We also point out the relationship between class imbalance and class overlapping problems, and their influence on the performance of CCCD classifiers and other classification methods as well as some state-of-the-art algorithms which are robust to class imbalance by construction. Experiments on both simulated and real data sets indicate that CCCD classifiers are robust to the class imbalance problem. CCCDs substantially undersample from the majority class while preserving the information on the discarded points during the undersampling process. Many state-of-the-art methods, however, keep this information by means of ensemble classifiers, but CCCDs yield only a single classifier with the same property, making it both appealing and fast.

Keywords: Class Cover Catch Digraphs, Class Cover Problem, Class Imbalance Problem, Class Overlapping Problem, Graph Domination, Prototype Selection, Support Estimation

1. Introduction

Class imbalance problem has recently become a topic of extensive research. In a two-class setting, imbalance in class(es) occurs when one class is represented by far more observations (points) than the other class in the data set (see, e.g., Chawla et al. (2004) and López et al. (2013)). Class imbalance problem is observed in many areas such as medicine, fraud detection and education. Some examples are clinical trials in which only 5% of patients in the data set have a certain disease, such as cancer (Mazurowski et al., 2008); detecting fraudulent customers where most individuals are law-abiding in insurance, credit card and telecommunications industries (Phua et al., 2004); and archives of college students where

mostly the ones who have fair results are kept (Thai-Nghe et al., 2009). In these and many other real life cases, majority class (i.e., the class with larger size) confounds the classifier performance by hindering the detection of subjects from the minority class (i.e., the class with fewer points).

The classification methods in machine learning usually suffer from the imbalance of class sizes in the data sets because most of these methods work on the assumption that class sizes are balanced (Japkowicz and Stephen, 2002). For example, the commonly used k -nearest neighbor (k -NN) classification algorithm is highly influenced by the class imbalance problem. In the k -NN approach, a new point is classified as the class of the most frequent one from its first k nearest neighbors (Fix and Hodges, 1989; Cover and Hart, 1967). As a result, in a two-class setting where one class substantially outnumbers the other, a point is more likely to be classified as the majority class by the k -NN classifier. In literature, sensitivity of k -NN classifier to the class imbalance problem and some solutions on choosing the appropriate k have been discussed in cases of imbalanced classes (see Mani and Zhang, 2003; Garcia et al., 2008; Hand and Vinciotti, 2003). Decision trees and support vector machines (SVM) are also some of the well known classifiers that are sensitive to the class imbalance in a data set (Japkowicz and Stephen, 2002; Tang et al., 2009). SVMs are among the most commonly used algorithms in the machine learning literature due to their well understood theory and high performance among popular algorithms (Wu et al., 2008; Fernández-Delgado et al., 2014), but these methods have been demonstrated to be inefficient against highly imbalanced data sets, although SVMs are still robust to moderately imbalanced data sets (Akbari et al., 2004; Raskutti and Kowalczyk, 2004).

We approach the classification of imbalanced data sets with methods that solve class cover problem (CCP), where the goal is to find a region that encapsulates all the members of the class of interest (i.e., target class). This particular region can be viewed as a *cover*; hence the name *class cover* (Cannon and Cowen, 2004). This problem is closely related to another problem in statistics, namely *support estimation*: estimating the support of a particular random variable defined in a measurable space (Schölkopf et al., 2001). Here, each cover can be realized as estimates of its associated class support. Priebe et al. (2001) introduced the class cover catch digraphs (CCCD) to find graph theoretic solutions to the CCP problem, and provided some results on the minimum dominating sets and the distribution of the domination number of such digraphs for one dimensional data. Priebe et al. (2003a) applied CCCDs on classification and showed that approximate minimum dominating sets of CCCDs (which were obtained by a greedy algorithm) and radii of the covering balls can be used to establish efficient classifiers. Moreover, DeVinney et al. (2002) defined random walk CCCDs (RW-CCCDs) where balls of class covers are more relaxed compared to previously introduced so called pure-CCCDs (P-CCCDs). In P-CCCDs, no member of the non-target class is covered, but RW-CCCDs allow some points of non-target class to be covered by the cover of the target class. Some target class points may also be uncovered in the process. Hence, RW-CCCDs may potentially avoid overfitting. CCCDs have been applied in face detection (Socolinsky et al., 2003) and latent class discovery in gene expression data (Priebe et al., 2003b). There are several other approaches in the literature to solve the class cover problem, including covering the classes with a set of boxes (Bereg et al., 2012) or set of convex hulls (Takigawa et al., 2009).

In this article, we study the effects of class imbalance on two CCCD classifiers, P-CCCD and RW-CCCD. Moreover, we report on the effects of class overlapping problem (which is defined as deterioration of classification performance when class supports overlap) along with the class imbalance problem to further investigate the performance of CCCD classifiers when imbalance and overlapping between classes co-exist. Thus, we show that when there is a considerable amount of class imbalance, whether class supports overlap or not, the CCCD classifiers perform better than the k -NN classifier. We show the robustness of CCCD classifiers to the class imbalance by simulating cases having increasing levels of class imbalance. We also compare CCCD classifiers with SVM classifiers which are potentially robust to moderate levels of class imbalance but not to high levels. With respect to class imbalance problem, the k -NN, SVM and decision tree classifiers may be referred to as “weak” classifiers; that is, these methods perform weakly when there is imbalance in the data set. However, such classifiers can be modified to address the unequal priors in a data set, and hence, can be converted to “strong” classifiers which are potentially robust to the class imbalance problem. We show that CCCD classifiers are also inherently robust (i.e., robust to class imbalance without any modification), and we compare the CCCD classifiers against the state-of-the-art strong classification methods which are constructed to perform well when class imbalance occurs. We consider ensemble learning, cost sensitive learning and resampling schemes in conjunction with k -NN, SVM and decision tree classifiers, and show that RW-CCCDs and P-CCCDs perform comparable to those strong classifiers.

Among the two variations of CCCD classifiers, we show that the RW-CCCD is more appealing in many aspects. For both simulated and real life examples, RW-CCCDs perform better than P-CCCDs and weak classifiers, and perform comparable to strong classifiers when the classes of data sets are imbalanced and/or overlapping. Moreover, we report on the complexity of the two CCCD classifiers and demonstrate that RW-CCCDs reduce the data sets substantially more than the other classifiers, thus increasing the testing speed. But most importantly, while reducing the majority class to mitigate the effects of class imbalances, CCCDs preserve the information on the discarded points of the majority class. CCCDs provide a novel potential solution to the class imbalance problem: that is, they capture the density around prototype points (i.e., members of the dominating sets) as radii of the covering balls. Hence, CCCDs preserve the information while reducing the data set. In the literature, only the strong classifiers based on hybrids of ensembles and resampling schemes achieve a similar task which requires multiple classifiers to be employed, and thus, result in lengthy training and testing time. However, CCCDs define single classifiers that undersample the data set with, possibly, a slight loss of information.

We provide a short review of the existing methods for classifying data sets with class imbalance in Section 2, introduce P-CCCD and RW-CCCD classifiers in Section 3, discuss the balancing effect of CCCD classifiers in Section 4. Finally, in Section 5, we compare the CCCD classifiers with the classifiers that are both sensitive (weak) and non-sensitive (strong) classifiers to class imbalance by simulated and real data sets, and report on the computational complexity of all weak classifiers.

2. Methods for Handling Class Imbalance Problem

Solving the class imbalance problem received considerable attention in the machine learning literature (see Chawla et al., 2004; Kotsiantis et al., 2006; Longadge and Dongre, 2013). Almost all algorithms designed to mitigate the effects of class imbalance incorporate a “weak” classifier which is modified to show some level of robustness to the class imbalance problem. The weak algorithm is modified either (i) in data level which involves a pre-processing of the data set being used in training, or (ii) in algorithmic level such that a “strong” classifier is constructed with a decision rule suited for the imbalances in the data set. Many modern algorithms are hybrids of both types; but in particular, there are mainly three of them: resampling methods, cost-sensitive methods, and ensemble methods (He and Garcia, 2009).

Resampling methods are commonly employed to remove the effects of class imbalance in the classification process. Resampling methods provide solutions to the class imbalance problem by (i) downsizing the majority class (undersampling) or (ii) generating new (synthetic) points for the minority class (oversampling). Hence, such methods modify the classifiers only at the data level. It might be useful to clean or erase some points in the majority class to balance the data (Drummond et al., 2003; Liu et al., 2009). However, in some cases, all points from both classes may be valuable/important, and hence, should be kept despite the differences in the class sizes. Oversampling methods generate synthetic points similar to the minority class to mitigate the class imbalance problem while preserving the information (Han et al., 2005). On the other hand, Batista et al. (2004) suggest that the combination of both over and undersampling methods can further improve the classification performance. One such method is the SMOTE+ENN method where the oversampling method SMOTE of Chawla et al. (2002) and edited nearest neighbors (ENN) method of Wilson (1972) are applied to an imbalanced data set, consecutively. While SMOTE balances the classes of the data set by generating artificial points between members of the minority class, ENN cleans the data set to further increase the classification performance of the weak classifier. Here, ENN method is an undersampling method that primarily aims to remove noisy points from the data set but not to balance the classes.

Another family of methods, namely cost-sensitive learning methods, has originated from real life: the cost of misclassifying a minority and a majority class member is usually not the same (Elkan, 2001). Frequently, the minority class has higher misclassification cost than the majority class. Classification methods such as decision trees (e.g., C4.5), can be modified to take these costs into account (see Ling et al., 2004; Zadrozny et al., 2003). C5.0 is an extended version of C4.5 incorporating the cost of each class (Kuhn and Johnson, 2013). Most weak classifiers can be easily modified so as to recognizing misclassification costs. The constrained violation cost C of SVM classifiers can be adjusted to individual class costs (Chang and Lin, 2011). As for k -NN, one solution is to appoint weights to all points of the data set with respect to their classes. Hence, such weights are the costs of classes giving precedence to minority class points (Barandela et al., 2003). On the other hand, for those algorithms that costs are not inherently recognizable or available, meta-learning schemes can be used along with weak classifiers without modifying the classifiers. Such learning methods are similar to ensemble learning methods (Domingos, 1999).

A fast developing field called ensemble learning also contributes to the family of methods handling the class imbalance problem (see Galar et al., 2012). The idea is to combine several classifiers to create a new classifier which has significantly better performance than its constituents (Rokach, 2010). AdaBoost is a popular algorithm among this family of learning methods (Freund and Schapire, 1997; Wu et al., 2008). AdaBoost assigns weights to each of the points in the data set and updates these weights in accordance with how well the points are estimated by each classifier. Galar et al. (2012) provide a survey of the most important ensemble learning methods that solve the class imbalance problem. However, it has been observed in some studies that ensemble learning methods work best when used together with resampling methods (López et al., 2013). In fact, ensemble and resampling schemes compensate the shortcomings of each other. The EasyEnsemble is a classifier with two levels of ensembles. First, a random undersampled majority class and the original minority class are used to train an ensemble classifier, then another random sample is drawn in the same way to train a second ensemble. This process is repeated several times to mitigate the effects of information loss as each ensemble would be applied on a different random subset of the majority class.

3. Classification with Class Cover Catch Digraphs

Class Cover Catch Digraphs (CCCDs) offer graph theoretic solutions to CCP (Priebe et al., 2001, 2003a). The objective of CCP is to find a region that covers the members of a specific class. More specifically, let (Ω, M) be a measurable space and let $\mathcal{X}_n = \{x_1, x_2, \dots, x_n\} \subset \Omega$ and $\mathcal{Y}_m = \{y_1, y_2, \dots, y_m\} \subset \Omega$ be observations from two classes \mathcal{X} and \mathcal{Y} with class conditional distributions F_X, F_Y and a joint cdf $F_{X,Y}$, respectively. Let $\Omega = \mathbb{R}^d$ and, without loss of generality, assume that the target class (i.e., the class of interest) is \mathcal{X} . In a CCCD, for $x_i, x_j \in \mathcal{X}_n \subset \mathbb{R}^d$ is the center of a ball with radius $r_i = r(x_i)$. Each ball is represented by $B_i = B(x_i, r_i)$ and if $x_j \in B_i$ then x_i is said to cover (or catch) x_j . Here, $d(\cdot, \cdot)$ can be any dissimilarity measure but we use the Euclidean distance henceforth. A CCCD is a digraph $D = D(V, A)$ with vertex set $V(D) = \mathcal{X}_n$ and the arc set $A(D)$ where $(x_i, x_j) \in A(D)$ if and only if $x_j \in B_i$. The term “catch” refers to arc (x_i, x_j) of the digraph D where x_i is said to catch x_j . The binary relation $x_i \sim x_j$, which is defined as $x_j \in B_i$, is asymmetric, thus the adjacency of x_i and x_j is represented with directed edges or arcs which yield a digraph instead of a graph.

In CCCDs, the goal is to find a subset of balls $C_X \subseteq B_X = \{B_1, B_2, \dots, B_n\}$ such that $Q_X \subseteq \cup_{B \in C_X} B$ for $Q_X \subseteq \mathcal{X}_n$ where the set Q_X is some desirable subset of the target class training set \mathcal{X}_n which we want to cover. Preferably, the goal is to find a set C_X such that $Q_X = \mathcal{X}_n$, however it might be desirable that the class cover may ignore some target class points to avoid overfitting. If a class cover of a CCCD fails to cover some target class points, it is called an *improper* cover, otherwise it is a *proper* cover. For covering \mathcal{Y}_m , we reverse the roles of classes \mathcal{X} and \mathcal{Y} . The class \mathcal{Y} becomes the target class and \mathcal{X} becomes the non-target class. Finding an appropriate cover C_X is equivalent to finding the dominating set of the CCCD with $V(D) = \mathcal{X}_n$. Let $N(s) = \{t \in V(D) : (s, t) \in A(D)\}$ be the *open neighborhood* of a vertex $s \in V(D)$: the set of vertices that have an arc from the vertex s , or the neighbors of s . A dominating set of a digraph D is defined as a subset of vertices $S \subseteq V(D)$ such that union of the closed neighborhoods, defined by $\bar{N}(s) = N(s) \cup \{s\}$, of elements of S

is the vertex set of the digraph: $\cup_{s \in \bar{N}(s)} = V(D)$. Among all dominating sets, usually the ones with minimum cardinality, called the minimum dominating sets, are preferable. The cardinality of the minimum dominating set(s) is referred to as the *domination number*, denoted as $\gamma(D)$. However, minimum dominating sets are often computationally intractable and finding them is, in general, an NP-hard optimization problem. Hence, greedy algorithms are often employed to find sets with approximately minimum cardinality (Chvatal, 1979; DeVinney, 2003).

CCCDs can easily be generalized to the multi-class case with k classes. To establish the set of covers $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$ associated with a set of classes $\mathfrak{X} = \{\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_k\}$, we merge the classes into two classes as $\mathcal{X}_T = \mathcal{X}_i$ and $\mathcal{X}_{YT} = \cup_{j \neq i} \mathcal{X}_j$ for $i, j = 1, \dots, k$. We refer to classes \mathcal{X}_T and \mathcal{X}_{YT} as target class and non-target class, respectively. More specifically, target class is the class we want to find the cover of, and the non-target class is the union of the remaining classes. We transform the multi-class case into a two-class setting and find the cover of i -th class, C_i , for each $i = 1, \dots, k$.

We employ two families of CCCDs, pure-CCCDs (P-CCCDs) and random walk CCCDs (RW-CCCDs) that differ in the definition of the radius $r(x)$. In these two digraphs, the (approximate) minimum dominating set S and the classifier are defined in slightly different ways; with the main distinction between the two being the way the covers are defined. The covering balls of P-CCCDs do not contain any non-target class point (hence the name “pure”) whereas RW-CCCDs possibly allow some non-target class points inside of the class cover of the target class so as to avoid overfitting. Moreover, some target class points may also be excluded from the covers of RW-CCCDs. Therefore, P-CCCDs construct pure and proper covers but RW-CCCD covers are not necessarily pure or proper.

3.1 Classification with P-CCCDs

In P-CCCDs, the covering balls $B_x = B(x, r(x))$ exclude all non-target class points. Thus, for a target class point $x \in \mathcal{X}_n$, which is the center of a ball B_x , the radius $r(x)$ should be smaller than the distance between x and the closest non-target point $y \in \mathcal{Y}_m$: $r(x) < \min_{y \in \mathcal{Y}_m} d(x, y)$. Given $\tau \in (0, 1]$, the radius $r(x)$ is defined as follows (Marchette, 2010):

$$r(x) := (1 - \tau)d(x, l(x)) + \tau d(x, u(x)), \quad (1)$$

where

$$u(x) := \operatorname{argmin}_{y \in \mathcal{Y}_m} d(x, y)$$

and

$$l(x) := \operatorname{argmax}_{z \in \mathcal{X}_n} \{d(x, z) : d(x, z) < d(x, u(x))\}.$$

The effect of parameter τ on the radius $r(x)$ is illustrated in Figure 1 (DeVinney, 2003). The ball with radius $r(x)$ catches the neighboring target class points, and for any $\tau \in (0, 1]$, the ball B_x catches the same points as well. Hence, the choice of τ does not effect the structure of digraph but might affect the classification performance which will be shown later in Section 5. On the other hand, for all $x \in \mathcal{X}_n$, the definition of $r(x)$ in Equation (1) keeps any non-target point $y \in \mathcal{Y}_m$ out of the ball B_x , that is $\mathcal{Y}_m \cup B_x = \emptyset$ for all $B_x \in C_X$. Here, B_x is an open ball: $B_x = \{z \in \mathbb{R}^d : d(x, z) < r(x)\}$. The digraph D is “pure” since

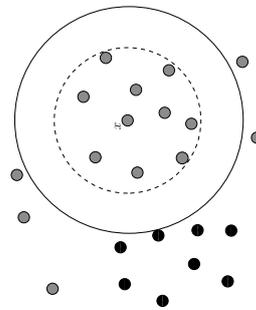


Figure 1: The effect of τ on the radius $r(x)$ of the target class point x in a two-class setting. Grey and black points represent the target and non-target classes, respectively. The solid circle centered at x is constructed with the radius associated with $\tau = 1$ and the dashed one with $\tau = 0.0001$ (DeVinney, 2003).

the balls contain only the target class points; hence, the name pure-CCCD. Once all balls are constructed, so is the digraph D . Therefore, we have to find a covering set C_X which is equivalent to finding a minimum dominating set $S \subseteq V(D)$. The greedy algorithm of finding an approximate minimum dominating set of a P-CCCD is given in Algorithm 1. At each iteration, the vertex which has the largest neighborhood (i.e., highest number of arcs) is removed from the graph together with its neighbors. Then, the process is repeated until all vertices of D are removed. The algorithm adds elements to the dominating set until all points are either dominated or dominate some other points. Hence, the covers established by P-CCCDs are proper covers: $Q_X = \mathcal{X}_n$ and $Q_Y = \mathcal{Y}_m$. The P-CCCD of one class, its associated class cover (constructed by the elements of the dominating set), and covers of both classes are illustrated in Figure 2.

Algorithm 1 The greedy algorithm for finding an approximate minimum dominating set of a digraph D . Here, $D[H]$ is the graph induced by the set of vertices $H \subseteq V(D)$ (see West, 2000).

Input: A digraph $D = D(V, A)$
Output: An approximate minimum dominating set, S

- 1: set $H = V(D)$ and $S = \emptyset$
- 2: **while** $H \neq \emptyset$ **do**
- 3: $\sigma^* = \text{argmax}_{v \in V(D)} |\bar{N}(v)|$
- 4: $S = S \cup \{\sigma^*\}$
- 5: $H = V(D) \setminus \bar{N}(\sigma^*)$
- 6: $D = D[H]$
- 7: **end while**

Before Algorithm 1 finds an approximate solution, we should first construct the digraph D . The P-CCCD cover C_X and the P-CCCD D depend on the distances between points of the target class \mathcal{X}_n , denoted by the matrix \mathcal{M}_X , and the distances from all points of \mathcal{X}_n to all points of \mathcal{Y}_m , denoted by matrix $\mathcal{M}_{X,Y}$. Later, we construct the set of balls

$B_X = \{B_1, B_2, \dots, B_n\}$, and get the set of arcs $A(D)$ where $V(D) = \mathcal{X}_n$. Hence, the minimum cardinality ball cover problem is reduced to a minimum dominating set problem. We find such a cover with Algorithm 2 which runs in quadratic time and, in addition, depends on the dimensionality of the training set $\mathcal{X}_n \cup \mathcal{Y}_m$.

Algorithm 2 The greedy algorithm for finding an approximate minimum cardinality ball cover C_X of the target class points \mathcal{X}_n given a set of non-target class points \mathcal{Y}_m .

Input: Points of the target class \mathcal{X}_n , the non-target class \mathcal{Y}_m , and the P-CCCD parameter $\tau \in (0, 1]$
Output: An approximate minimum cardinality ball cover C_X

- 1: $r(x) := (1 - \tau)d(x, l(x)) + \tau d(x, u(x))$ for all $x \in \mathcal{X}_n$
- 2: Construct the digraph D with the set B_X .
- 3: Find the approximate minimum dominating set S of digraph D by Algorithm 1.
- 4: $C_X := \cup_{x \in S} B(x, r(x))$

Theorem 1 Algorithm 2 is an $\mathcal{O}(\log n)$ -approximation algorithm and finds an approximate minimum cardinality ball cover C_X of the target class \mathcal{X} in $\mathcal{O}(n(n + m)d)$ time.

Proof. The algorithm is polynomial time reducible to a greedy minimum set cover algorithm which finds an approximate solution with size at most $\mathcal{O}(\log n)$ times of the optimum solution (Chvatal, 1979; Canny and Cowen, 2004). We first calculate the distance matrices \mathcal{M}_X and $\mathcal{M}_{X,Y}$ which take $\mathcal{O}(n^2d)$ and $\mathcal{O}(nm d)$ time, respectively. Constructing the digraph D requires computing $l(x)$ and $u(x)$ in Equation (1) for all $x \in \mathcal{X}_n$, taking $\mathcal{O}(n^2 + nm)$ time in total. Then, we set the arc set $A(D)$ in $\mathcal{O}(n^2)$ time. Finally, the algorithm finds a solution for the digraph D in $\mathcal{O}(n^2)$ time, hence the total running time of the algorithm is $\mathcal{O}(n(n + m)d)$. ■

When \mathcal{Y} is the target class, observe that the time complexity is $\mathcal{O}(n(n + m)d)$, and an approximate solution is of size at most $\mathcal{O}(\log m)$ times the optimal solution by Theorem 1, since $m = |\mathcal{Y}_m|$. A P-CCCD classifier consists of the covers of all classes, hence the total training time of finding CCCDs of a data set with two-class setting is $\mathcal{O}((n + m)^2d)$.

After establishing both class covers C_X and C_Y , any new data point can be classified in \mathbb{R}^d according to where it resides. Here, there are three cases according to the location of the given point, z , to be classified: z is (i) only in C_X or C_Y , (ii) in both C_X and C_Y or (iii) in neither of C_X and C_Y . The case (i) is straightforward: z belongs to class \mathcal{X} if $z \in C_X \setminus C_Y$ or to class \mathcal{Y} if $z \in C_Y \setminus C_X$. For cases (ii) and (iii), we need to find a way to decide the class of the point in a reasonable way. In fact, for all the cases, the estimated class of a given point z is determined by

$$\underset{C \in \{C_X, C_Y\}}{\text{argmin}} \left[\min_{x \in B(x, r)} \rho(z, x) \right] \quad (2)$$

where $\rho(z, x) = d(z, x)/r(x)$ (Marchette, 2010). The dissimilarity measure $\rho(x, z)$ indicates whether or not the point z is in the ball of radius $r(x)$ with center x , since $\rho(x, z) \leq 1$ if z is inside the (closure of the) ball and > 1 otherwise. The measure $\rho : \Omega \times \Omega \rightarrow \mathbb{R}_+$ is simply a scaled dissimilarity measure, since Euclidean distance between two points, $d(x, y)$, is divided (or scaled) with the radius, $r(x)$ or $r(y)$. This measure violates the symmetry axiom among metric axioms since $\rho(x, y) \neq \rho(y, x)$ whenever $r(x) \neq r(y)$. However, Priebe et al. (2003a) showed that the dissimilarity measure ρ satisfies the continuity condition,

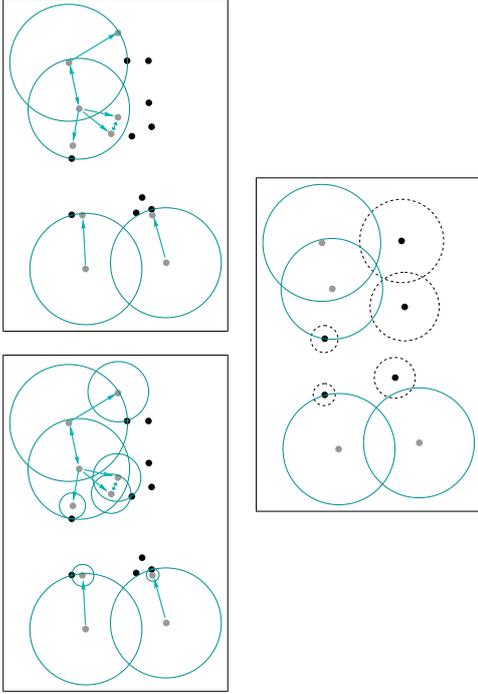


Figure 2: An illustration of the CCCDs (with the grey points representing the points from the target class) in a two-class setting. Presented in top left are all covering balls and the digraph $D = (V, A)$ and in top right are the balls that constitute a class cover for the target class and are centered at points which are the elements of the dominating set $S \subseteq V(D)$. In the bottom panel, we present the dominating sets of both classes and their associated balls which establish the class covers. The class cover of grey points is the union of solid circles, and that of black points is the union of dashed circles.

i.e., under the assumptions that both F_X and F_Y are continuous and strictly separable ($\inf_{x \in \mathcal{X}_n, y \in \mathcal{Y}_m} d(x, y) = \delta > 0$), P-CCCD classifiers are consistent; that is, their misclassification error approaches to the Bayes optimal classification error as $m, n \rightarrow \infty$. The measure ρ favors points with bigger radii; that is, for example, for a new point z equidistant to two points, the point with bigger radius is closer in terms of this scaled dissimilarity measure; for example, $\rho(x, z) < \rho(y, z)$ when $d(x, z) = d(y, z)$ and $r(x) > r(y)$. The radius $r(x)$ can be viewed as an indicator of the density around the point x . Thus, a point x with bigger radius might suggest that the point z is more likely be drawn from the same distribution (or class) where x is drawn (i.e., from the denser class).

3.2 Classification with Random Walk CCCDs

For P-CCCDs, the class covers defined by CCCDs were “pure” of non-target class points; that is, no member of the non-target class was allowed inside the cover of the target class. As in Figure 1, the ball centered at the point x cannot expand any further since its radius is restricted by the distance to the closest non-target class point. This strategy may cause the

cover to overfit or be sensitive to noise or outliers in the non-target class. By allowing some neighboring non-target class points inside the cover and some target class points outside the cover, the random walk CCCDs (RW-CCCDs) catch as much target class points as possible with an adaptive strategy of choosing the radii (DeVinney et al., 2002). For $x \in \mathcal{X}_n$, $|\mathcal{X}_n| = n$ and $|\mathcal{Y}_m| = m$, RW-CCCDs define a function on radius of a ball given by

$$R_x(r) = R_x(r; \mathcal{X}_n, \mathcal{Y}_m) \quad (3)$$

$$:= \frac{m}{n} |\{z \in \mathcal{X}_n : d(x, z) \leq r\}| - |\{z \in \mathcal{Y}_m : d(x, z) \leq r\}|.$$

where second and third arguments in $R_x(r; \mathcal{X}_n, \mathcal{Y}_m)$ are suppressed when there is no ambiguity. The function $R_x(r)$ can be viewed as a one-dimensional random walk. When the ball centered at $x \in \mathbb{R}^d$ expands, it hits either a target class point or a non-target class point which increases or decreases the random walk by one unit, respectively. The ratio m/n is included in the first term as to avoid the bias resulted by unequal sample sizes (i.e., class imbalance). An illustration is given in Figure 3 for the case $m = n$. The function $R_x(r)$ aims to find such radii that it contains a few non-target class points and sufficiently many target class points. In addition, we also want to avoid balls with large radii. Hence, the radius of x is the value maximizing $R_x(r)$ with an additional penalty function $P_x(r)$ which biases toward small radii:

$$r_x := \operatorname{argmax}_{r \in \{d(x, z) : z \in \mathcal{X}_n \cup \mathcal{Y}_m\}} R_x(r) - P_x(r). \quad (4)$$

Although a penalty function seems fit, DeVinney (2003) pointed out that the choice of $P_x(r) = 0$ usually works sufficiently well in practice. As in P-CCCDs, the radius of a ball represents the density of its center’s neighborhood. Maximizing $R_x(r)$ determines the best possible radius. Moreover, unlike P-CCCDs, the balls of RW-CCCDs are closed balls: $B_x = \{z \in \mathbb{R}^d : d(x, z) \leq r(x)\}$.

Similar to P-CCCDs, finding a cover, or a dominating set, of a RW-CCCD is an NP-hard problem. However, RW-CCCDs find the minimum dominating sets in a slightly different fashion. Instead of finding a set S such that $\cup_{s \in S} \bar{N}(s) = V(D)$ as in Algorithm 1, we first locate the vertex x^* (a target class point) which has maximum of some score, T_{x^*} , and remove all target and non-target class points covered with the ball of this vertex, B_{x^*} . In the next iteration, we recalculate the radii of remaining target class points, find the next point with the maximum score and continue until all target class points are covered. This greedy method of finding dominating set(s) S of RW-CCCDs is given in Algorithm 3. The resulting dominating set S has approximate minimum cardinality. For each target class point $x \in \mathcal{X}_n$, the score T_x is associated with $R_x(r_x)$ and is given by

$$T_x = R_x(r_x) - \frac{r_x^{n_u}}{2d_m(x)} \quad (5)$$

where n_u is the number of uncovered target class points in the current iteration, and $d_m(x) = \max_{z \in \mathcal{X}_n} d(x, z)$. The term which is linear in r_x of the right hand side of Equation (5) is similar to $P(r)$ in Equation (4): it biases the scores toward choosing dominating points with smaller radii. On the other hand, Algorithm 3 is likely to choose dominating points with radius $r = 0$. These points only dominate themselves but they are thought of being

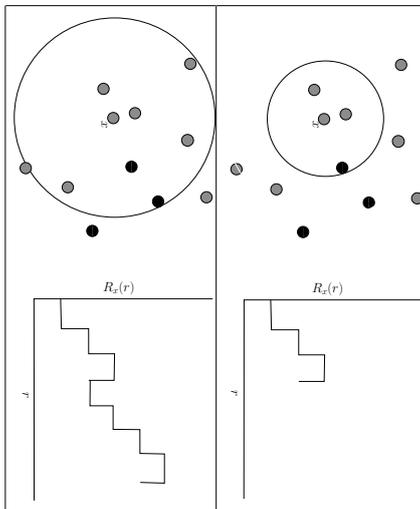


Figure 3: Two snapshots of $R_x(r)$ associated with the ball B_x centered at x for $m = n$.

Algorithm 3 The greedy algorithm for finding an approximate minimum dominating set for RW-CCCDs of points \mathcal{X}_n from the target class given non-target class points \mathcal{Y}_m .

Input: Target class points \mathcal{X}_n and non-target class points \mathcal{Y}_m
Output: Approximate dominating set S of \mathcal{X}_n

- 1: $H_0 = \mathcal{X}_n, H_1 = \mathcal{Y}_m$ and $S = \emptyset$
- 2: $\forall x \in \mathcal{X}_n, d_m(x) = \max_{z \in \mathcal{X}_n} d(x, z)$
- 3: **while** $H_0 \neq \emptyset$ **do**
- 4: $n_u = |H_0|$
- 5: **for all** $x \in \mathcal{X}_n$ **do**
- 6: $r(x) = \operatorname{argmax}_r, R_x(r; H_0, H_1)$ for $r \in \{d(x, z) : z \in H_0 \cup H_1\}$
- 7: **end for**
- 8: $x^* = \operatorname{argmax}_{x \in H_0} R_x(r(x); H_0, H_1) - \frac{r(x)n_u}{2d_m(x)}$
- 9: $S = S \cup \{x^*\}$
- 10: $H_0 = H_0 \setminus (\bar{N}(x^*) \cap \mathcal{X}_n)$ and $H_1 = H_1 \setminus (\bar{N}(x^*) \cap \mathcal{Y}_m)$
- 11: **end while**
- 12: $C_X := \cup_{s \in S} B(s, r(s))$

not covered since their balls have radii $r = 0$. Hence, RW-CCCDs may establish improper covers.

Algorithm 3 is similar to Algorithm 2, however after each iteration, a point is added to the set S and the random walk $R_x(r)$ is recalculated for all uncovered $x \in \mathcal{A}_0$. Hence, we need an additional sweep on the training set which makes Algorithm 3 run in cubic time.

Theorem 2 Algorithm 3 finds covers C_X of the target class \mathcal{X} and \mathcal{Y} in $\mathcal{O}((n + d + \log(n + m))(n + m)^2)$ time.

Proof. In Algorithm 3, the matrix of distances between points of training set $\mathcal{X}_n \cup \mathcal{Y}_m$ should be computed since, for all $x \in \mathcal{X}_n \cup \mathcal{Y}_m$, the entire data set is swept to maximize $R_x(r)$. This takes $\mathcal{O}((n + m)^2d)$ time. The algorithm runs until all target class points

are covered, but for each iteration, the random walk $R_x(r)$ is recalculated. The maximum $R_x(r_x)$ could be found by sorting the distances for all $x \in \mathcal{A}_0$ which could be done prior to the while loop. This sorting takes $\mathcal{O}((n + m)^2 \log(n + m))$ time. Since \mathcal{A}_0 and \mathcal{A}_1 are updated at each iteration, we can just erase the distances corresponding to points covered by $N(x^*)$ which does not change the order of sorted list provided before the while loop. Hence, $\operatorname{argmax} R_x(r_x)$ is found and the covered points erased in $\mathcal{O}((n + m)^2)$ time. The while loop iterates n times in the worst case, and hence the algorithm runs in a total of $\mathcal{O}((n + d + \log(n + m))(n + m)^2)$ time. ■

Note that Algorithm 3 finds a cover of \mathcal{Y}_m in $\mathcal{O}((m + d + \log(n + m))(n + m)^2)$ time which makes a RW-CCCD classifier trained in $\mathcal{O}((n + m)^3)$ time for $d < n$ and $\log(n + m) < n$. RW-CCCD classifiers are much better classifiers that potentially avoid overfitting, but with a cost of being much slower compared to the P-CCCD classifiers.

Since P-CCCD covers are pure and proper covers, P-CCCD classifiers tend to overfit (DeVinney, 2003). In RW-CCCDs, covering balls allow some points of \mathcal{Y}_m inside C_X to increase average classification performance. In that case, Algorithm 3 cannot be reduced to a minimum set cover problem since the definition of sets change after adding a single point to the dominating set. Hence, the upper bound $\mathcal{O}(\log n)$ does not apply to RW-CCCDs. However, we expect to get bigger balls in RW-CCCDs compared to the ones in P-CCCDs which intuitively suggests that the covers of RW-CCCDs are lower in cardinality. We conduct empirical studies to show that RW-CCCDs, in fact, produce dominating sets with lower size compared to P-CCCDs in some cases.

In RW-CCCD, once the class covers (or dominating sets) are determined, the scaled dissimilarity measure in Equation (2) is a good choice for estimating the class of a new point z . However, DeVinney (2003) incorporates the scores of each ball to produce better performing class covers in classification. Hence, the class of a new point z is determined by

$$\operatorname{argmin}_{C \in \{C_X, C_Y\}} \left[\min_{x \in B(x, r)} \rho(z, x) T_x^e \right]$$

where $\rho(z, x)$ is defined as in Equation (2). Here, $e \in [0, 1]$ controls at what level the score T_x is incorporated. We observe that for $d(z, x) < r(x)$, $\rho(z, x) = d(z, x)/r(x)$ decreases as T_x increases. Hence, if a new point z is in both covers, $z \in C_X \cap C_Y$, the score T_x is a good indicator to which class the new point z belongs since the bigger the T_x , the more likely the ball contains more target class points. For $e = 1$, we fully incorporate each score T_x of covering balls and with $e = 0$, we ignore the scores. By introducing a value for the parameter e in $(0, 1)$, it is possible to further improve the performance of RW-CCCD classifiers.

4. Balancing the Class Sizes with CCCDs

The CCCD classifiers substantially reduce the number of majority class observations in a data set. The reason is that balls of majority class members are more likely to catch neighboring points of the same class. The greedy algorithm given in Algorithm 1 selects vertices with the largest closed neighborhood. Similarly, Algorithm 3 selects vertices so that their balls are as dense as possible (i.e., target class points are abundant in the balls) with some contaminating non-target class points. Both algorithms choose balls with a large

number of target class points, and hence substantially reduce the data set (in particular, majority class points). Points of the minimum dominating set correspond to the centers of balls that establish the class covers. Hence CCCD classifiers can also be viewed as *prototype selection* methods where the objective is finding a set of points, or *prototypes*, S ; from the training set to preserve or increase the classification performance while substantially reducing the sample size. However, the radii of dominating set(s) are also stored and used in the classification process.

In Figure 4, we illustrate the behavior of balls associated with P-CCCDs and RW-CCCDs. Note that in both families of digraphs, balls of the majority class tend to be larger and hence are more likely to catch more majority class points. Since the majority class has much more members than the minority class, balls of the majority points are more likely to catch the neighboring majority points. CCCD classifiers keep the information of ball centers and their associated radii. Larger cardinality of the majority class allows the construction of bigger balls and hence, larger values of radii are more likely to correspond to larger number of caught class members. As a result, CCCDs balance the data set and, at the same time, preserve the information of the local density by retaining the radii. The data set becomes balanced since the center of balls are the points of the new training data set which will be employed later in classification.

The loss of information in undersampling schemes are of course inevitable, however it is possible to preserve a portion of that discarded information by other means. EasyEnsemble is an ensemble classifier used for that very purpose; however, it needs multiple classifiers to be employed. Each classifier is trained on a different balanced subset of the original training data set, and hence the ensemble classifier preserves the information on the entire data set given by a collection of unbiased classifiers. On the other hand, CCCDs achieve the same goal by transforming the density around points into the radii. CCCDs resemble cluster based resampling methods in that regard. Instead of randomly sampling the data set, cluster based sampling schemes divide each class into clusters, and then, oversample the minority class or undersample the majority class proportional to each subclass. Covering balls of CCCDs have a similar purpose which has also been discussed in Pritebe et al. (2003b). They use the covering balls of the minimum dominating sets to explore the latent subclasses of each class of gene expression data sets. In fact, the balls of CCCDs may correspond to clusters. Hence, sets of points associated with each cluster is undersampled to a single point (i.e., a prototype or a dominating point), and the information on the cluster is provided by the radius which represents the density of that cluster. The bigger the radius, the more influence a prototype has over the domain. In P-CCCDs, the radii may be sensitive to noise, but RW-CCCDs ignore noisy points to avoid overfitting. Moreover, in RW-CCCDs, we have an additional statistic provided by each cluster, the score given in Equation (5) based on the random walk. We use both the radii and these scores to define the RW-CCCD classifiers, and thus achieve better performing classifiers with more reduction and less information loss.

We approach the problem of class imbalance from the perspective of class overlapping problem as well. Several researchers on class imbalance revealed that overlap between the class supports degrade the classification performance of imbalanced data sets even more (see Prati et al., 2004; Batista et al., 2004, 2005; Galar et al., 2012). Let $E \subset \mathbb{R}^d$, and let $s(F_X)$ and $s(F_Y)$ be the supports of the classes \mathcal{X} and \mathcal{Y} , respectively. We define E as the overlapping region of these two class supports, $E := s(F_X) \cap s(F_Y)$. Moreover, let

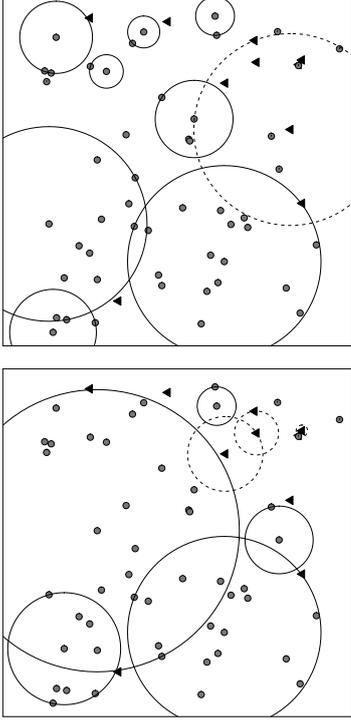


Figure 4: An illustration of the covering balls associated with majority and minority P-CCCD (left) and the corresponding RW-CCCDs ($\tau = 0.0001$) (right) of an imbalanced data set in a two-class setting where majority and minority class points are represented by grey dots and black triangles, respectively.

$q(E) := |\mathcal{Y}_m \cap E|/|\mathcal{X}_n \cap E|$ be ratio of class sizes restricted to the region $E \subset \mathbb{R}^d$. We say $q(E)$ is the “local” imbalance ratio with respect to E . Also, let the “global” imbalance ratio be $q = q(\mathbb{R}^d) = m/n$. Throughout this work, in both simulated and real data examples, we study and discuss the local imbalance ratio $q(E)$ restricted to the overlapping region E and the global imbalance ratio q . We specifically illustrate the performance of several classifiers for various levels of class imbalance (local or global) and class overlapping, and assess the performance of CCCD classifiers compared to weak and strong versions of k -NN, SVM and C4.5 classifiers.

5. Comparing CCCDs with Other Classifiers

We study the performance of CCCD classifiers in comparison with weak and strong classifiers in two separate sections. Recall that we call a classifier as “weak” when the method is inherently sensitive to class imbalance, and as “strong” when it is non-sensitive (or less sensitive). We use the area under curve (AUC) measure to evaluate the performance of the classifiers on the imbalanced data sets (López et al., 2013). AUC measure is often used on imbalanced real data classes. This measure has been shown to be better than the correct classification rate in general (Huang and Ling, 2005). We discuss the computational complexity of weak classifiers to emphasize the testing speed of CCCD classifiers when trained by imbalanced data sets. Finally, we compare both weak and strong classifiers with CCCDs on real data sets by considering the overlapping and imbalance ratios of all data sets.

5.1 Monte Carlo Simulation Study with Weak Classifiers

In this section, we compare the CCCD-based classifiers, namely P-CCCD and RW-CCCD, with k -NN, support vector machines (SVM) and C4.5, on simulated data sets. These classifiers are listed in Table 1. We employ the `cccd`, `e0171` and `Rkcka` packages in R to classify test data sets with the P-CCCD, SVM (with Gaussian kernel) and C4.5 classifiers, respectively (Marchette, 2013; Meyer et al., 2014; R Core Team, 2015).

For each of four classification methods other than C4.5, we assign the optimum parameter values which are the best performing values among all considered parameters. For example, an optimum the P-CCCD parameter τ is found in a preliminary (pilot) Monte Carlo simulation study associated with the main simulation setting (i.e., the same setting of the main simulation). In the pilot study, we perform a Monte Carlo simulation with 200 replications and count how many times a τ value has the maximum AUC among $\tau = 0.0, 0.1, \dots, 1.0$ in 200 trials. Note that, since $\tau \in (0, 1]$, we denote $\tau = \epsilon$ (machine epsilon) as $\tau = 0$ for the sake of simplicity. For each replication of the pilot simulation, we (i) classify the test data set with all τ values, (ii) record the τ values with maximum AUC and (iii) update the count of the recorded τ values. Finally, we appoint the one that has the maximum count (the best performing τ) as the τ^* , the optimum τ . Then, we use τ^* as the parameter of P-CCCD classifier in our main simulation. The parameters of optimal k -NN, SVM and RW-CCCD classifiers are defined similarly. SVM methods often incorporate both a kernel parameter γ and a constrained violation cost C . We only optimize γ since the selection of an optimum C parameter will be more crucial for cost-sensitive SVM methods. Moreover, we consider two versions of the C4.5 classifier where both incorporate Laplace smoothing. The first tree classifier, C45-LP, prunes the decision tree with %25 confidence level but the second classifier, C45-LNP, does not use pruning at all.

We first consider a simulation setting similar to the one in DeVinney et al. (2002) where CCCD classifiers showed relatively good performance compared to the k -NN classifier. Here, we simulate a two-class setting where observations from both classes are drawn from separate multivariate uniform distributions: $F_X = U(0, 1)^d$ and $F_Y = U(0.3, 0.7)^d$ for $d = 2, 3, 5, 10$. Notice that $s(F_Y) \subset s(F_X)$; i.e., $E = s(F_Y)$. We perform Monte Carlo replications where on each replication, we train the data with equal sizes of observations ($m = n$) from each class for $n = 50, 100, 200, 500$. On each replication, we record the AUC measures of the classifiers on the test data set with 100 observations from each class, resulting a test data set of size 200. We simulate test data sets until AUCs of all classifiers achieve a standard error below 0.0005. Average of AUCs of all classifiers in Table 1 are given in Figure 5 for all (n, d) combinations. Additionally, in Figure 6, we report the τ values of best performing P-CCCD classifiers in our pilot simulation study for all (n, d) combinations. In Figure 6, there are separate histograms for each combination. Each histogram represents the number of times a τ value has the maximum AUC. Also in Figure 7, we report the e values of the best performing RW-CCCD classifiers of the same pilot simulation study for $e = 0, 0.1, \dots, 1.0$.

We start by investigating the effect of τ and e on CCCD classifiers. The relationship between τ , n and d can also be observed in Figure 6. The higher the τ value, the better the performance of P-CCCD classifier with increasing d and decreasing n . This may indicate that balls with $\tau = 0$ (i.e., $\tau = \epsilon$) represent the density around their centers better for low dimensional data sets. However, with increasing dimensionality and lower class sizes, the

Method	Description
P-CCCD	P-CCCD with the optimum τ (in the pilot study) among $\tau = 0.0, 0.1, \dots, 1.0$
RW-CCCD	RW-CCCD with the optimum e (in the pilot study) among $e = 0.0, 0.1, \dots, 1.0$
k -NN	k -NN with optimum k (in the pilot study) among $k = 1, 2, \dots, 30$
SVM	SVM with the radial basis function (Gaussian) kernel with the optimum γ (in the pilot study) among $\gamma = 0.1, 0.2, \dots, 3.9, 4.0$ (Joachims, 1999)
C45-LP	C4.5 with Laplace smoothing and reduced error pruning (%25 confidence)
C45-LNP	C4.5 with Laplace smoothing and no pruning

Table 1: The description of classifiers employed in the article.

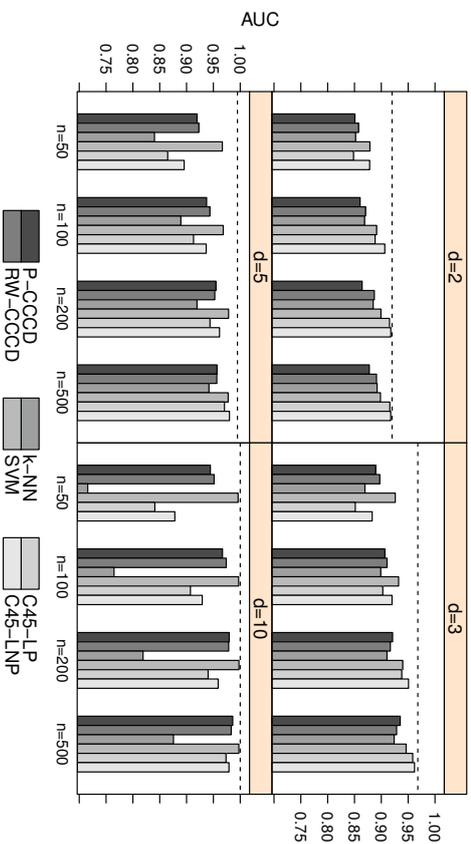


Figure 5: CCRs in the two-class setting, $F_X = U(0, 1)^d$ and $F_Y = U(0.3, 0.7)^d$ under various simulation settings, with $d = 2, 3, 5, 10$ and equal class sizes $m = n = 50, 100, 200, 500$.

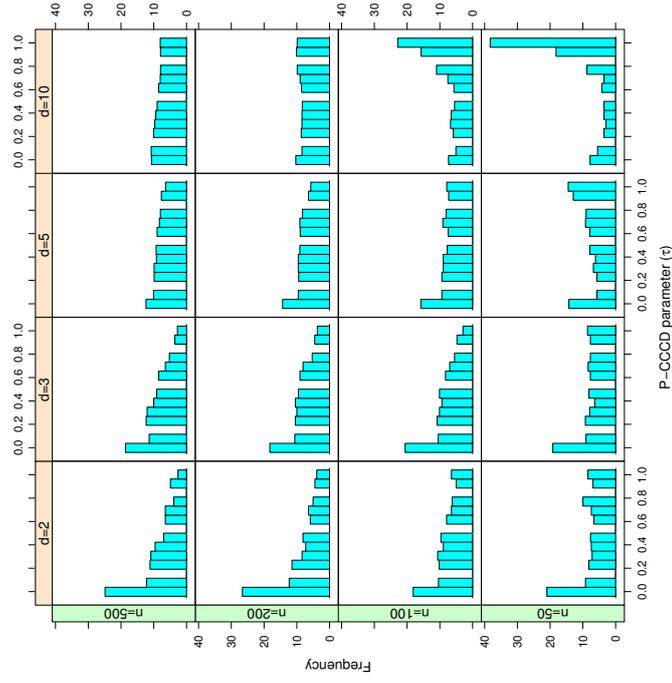


Figure 6: Frequencies of the best performing τ values among $\tau = 0.0, 0.1, \dots, 1.0$ in our pilot study (this is used to determine the optimal τ used in P-CCCD). The simulation setting is same as to the one presented in Figure 5.

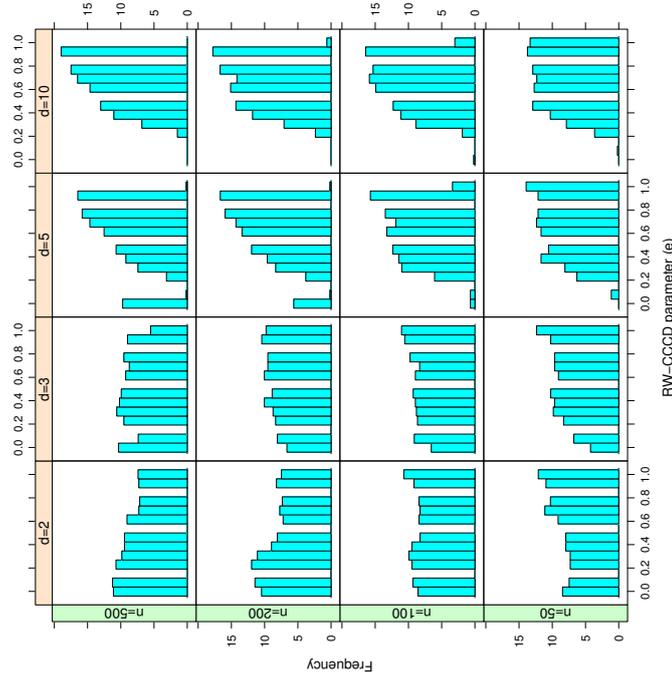


Figure 7: Frequencies of the best performing ϵ values among $\epsilon = 0.0, 0.1, \dots, 1.0$ in our pilot study (this is used to determine the optimal ϵ used in RW-CCCD). The simulation setting is same as to the one presented in Figure 5.

set of points gets sparser in \mathbb{R}^d . In the case of RW-CCCD, classifiers with high e values are either better or comparable to those with lower e values. The scores \mathcal{I}_x of covering balls are definitely beneficial to the performance of the RW-CCCD classifiers, however with increasing n and decreasing d (especially for $n = 500$ and $d = 2$) RW-CCCD with lower e is better since the radii successfully represent the density around the prototype points due to the high number of observations in the data set.

Figure 5 illustrates the AUCs of all classifiers along with the Bayes optimal performance given with the dashed line. Comparing the performance of CCCD classifiers with other classification methods, we observe that RW-CCCD and P-CCCD classifiers outperform the k -NN classifier when the support of one class is entirely embedded inside that of the other class. These results are similar to the conclusions of DeViney et al. (2002): with increasing dimensionality, the difference between k -NN and CCCD classifiers becomes more apparent, i.e., CCCD classifiers have nearly 0.20 AUC more than k -NN. On the other hand, the SVM classifier has about 0.05 more AUC than P-CCCD and RW-CCCD classifiers, especially for lower class sizes. Although, both versions of CCCD classifiers outperform the k -NN and C4.5 classifiers with increasing dimensionality, the gap between these two classifiers and CCCD classifier is getting narrower with increasing class sizes. The RW-CCCD classifier is slightly better than the P-CCCD classifier for lower n . In addition, C45-LNP achieves slightly better results than C45-LP.

In the setting presented in Figure 5, apparently, two classes overlap on the region $E = s(F_X) = [0.3, 0.7]^d$ which is the entire support of the class \mathcal{Y} . For equal class sizes, $q = m/n = 1$ but $q(E) \approx (1/0.4)^d = \text{Vol}(s(F_X))/\text{Vol}(s(F_X))$, where $\text{Vol}(\cdot)$ is the volume functional. The classes are clearly imbalanced in E , although $m = n$. Hence, class \mathcal{X} becomes the minority and class \mathcal{Y} becomes the majority class with respect to E . However, readjusting the class sizes m and n might change the performance of P-CCCD and RW-CCCD classifiers compared to the k -NN and C4.5 classifiers. Therefore, we conduct another simulation study with classes from the same uniform distributions, but we set $m = 50$ and $n = 200$ for $d = 2, 3$, and $m = 50$ and $n = 1000$ for $d = 5, 10$. In this experiment, we simulated 4 times more \mathcal{X} class members than \mathcal{Y} for $d = 2, 3$, and 20 times more for $d = 5, 10$. Results of this second experiment is given in Figure 8. k -NN and C4.5 classifiers outperform P-CCCD classifier in all d cases and has comparable AUC with SVM. However, only for $d = 2, 5$, RW-CCCD classifier achieves considerably more or comparable AUC compared to other classifiers. In this example, k -NN classifiers have nearly 0.05 more AUC than P-CCCD, and also RW-CCCD have, in general, 0.05 more AUC than k -NN classifiers.

Results from Figures 5 and 8 seem conflicting to each other, even though $E = s(F_X)$. In the simulation setting of Figure 8, we draw more samples from the class \mathcal{X} to balance the class sizes with respect to E . In fact, the effect on the difference of AUCs between CCCD, k -NN and C4.5 classifiers depends heavily on the local class imbalance restricted to the overlapping region E . The classes in region E are less imbalanced in setting of Figure 8 than in the setting of Figure 5. Observe that $q(E) \approx (1/0.4)^d/4$ when $(m, n) = (50, 200)$, $q(E) \approx (1/0.4)^d/20$ when $(m, n) = (50, 1000)$, and $q(E) \approx (1/0.4)^d$ in $(m, n) = (50, 50)$. Hence, d does also affect the balance between classes. With increasing d , the region E gets smaller in volume compared to $s(F_X)$ and, as a result, fewer points of the class \mathcal{X} falls in E . Thus, we need to draw more samples from \mathcal{X} as dimensionality increases, in order to balance the classes with respect to E . These results suggest that, the more imbalanced the data set

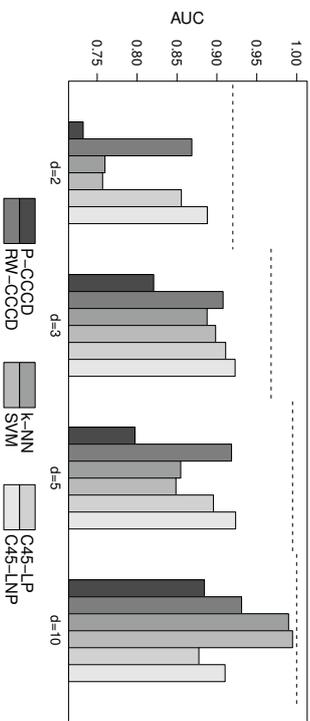


Figure 8: CCRs in a two-class setting, $F_X = U(0, 1)^d$ and $F_Y = U(0.3, 0.7)^d$ with fixed $n = 200$ and $m = 50$ in $d = 2, 3$, and with fixed $n = 1000$ and $m = 50$ in $d = 5, 10$.

in overlapping region E , the worse the performance of k -NN and C4.5 classifiers while CCCD classifiers preserve their classification performance. So, CCCD classifiers exhibit robustness (to the class imbalance problem). On the other hand, in Figure 5, we observe that the AUC of k -NN classifier approaches to the AUC of CCCD classifiers with increasing class sizes. Because, when q and $q(E)$ are fixed, the classification performance still depends on individual values of n or m . This result is in line with the results of Japkowicz and Stephen (2002) who reported that the effect of class imbalance on the classification performance diminishes if both class sizes are sufficiently large. Furthermore, SVM classifier performs better than all classifiers in Figure 5, and performs worse than RW-CCCD classifiers only for $d = 2, 5$ in Figure 8. This might be an indication that SVM classifier is also not affected by the local class imbalance with respect to E , and performs usually better than both P-CCCD and RW-CCCD classifiers if the support of one class is inside the other. For the C4.5 classifier, on the other hand, it is known for quite some time that the pruning is detrimental for classifying imbalanced data sets (Cieslak and Chawla, 2008). In any case, C45-LNP has more AUC than C45-LP in all simulation settings.

In a two-class setting with an overlapping region E , we should expect CCCD classifiers to outperform k -NN classifiers in cases of (global or local) class imbalance. Let $F_X = U(0, 1)^d$ and $F_Y = U(\delta, 1 + \delta)^d$ for $\delta, q = 0.05, 0.10, \dots, 0.95, 1.00$; $d = 2, 3, 5, 10$; $n = 400$ and $m = qn$. Here, the shifting parameter δ controls the level of overlap. The class supports get more overlapped with decreasing δ . Since $E = (\delta, 1)^d$ and the supports of both classes are unit boxes, observe that $q(E) \approx q$. The closer the value of q to 1, more balanced the classes are. We aim to address the relationship between the classifiers for various combinations of overlapping and global class imbalance ratios.

Figure 9 illustrates the difference between AUCs of CCCD and other classifiers (k -NN, SVM and C4.5) in separate heat maps for $d = 2, 3, 5, 10$. We use the unpurged C4.5 classifier

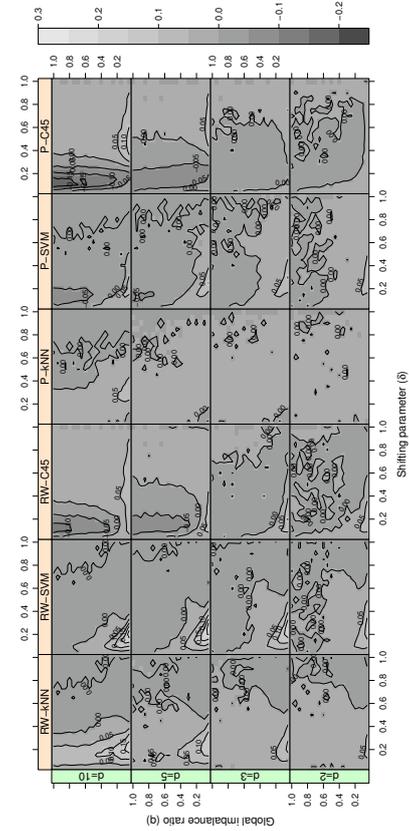


Figure 9: Differences between the AUCs of CCCD and other classifiers. For example, the panel titled with "RW-kNN" presents $AUC(RW\text{-}CCCD) - AUC(k\text{NN})$. In this two-class setting, classes are drawn from $F_X = U(0, 1)^d$ and $F_Y = U(\delta, 1 + \delta)^d$ with $d = 2, 3, 5, 10$. Each cell of the grey scale heat map corresponds to a single combination of simulation parameters $\delta, q = 0.05, 0.1, \dots, 0.95, 1.00$ with $n = 400$ and $m = qn$.

C4.5-LNP, since it tends to perform better for imbalanced data sets, and we refer to C4.5-LNP as C4.5 for simplicity. Each cell of a single heat map is associated with a combination of δ and q values. Lighter tone cells indicate that CCCD classifiers are better than the other classifiers in terms of AUC, and vice versa for the darker tones. When the classes are imbalanced and moderately overlapping, RW-CCCD classifier has at least 0.05 more AUC than all other non-CCCD classifiers but P-CCCD classifier is only better than all others provided that $d = 10$. If the classes are balanced or their supports are not considerably overlapping, there seem to be no visible difference between CCCD and the other classifiers. Thus, the other classifiers suffer from the imbalance of the data while CCCD classifiers show robustness to the class imbalance. But more importantly, this difference is getting more apparent with increasing dimensionality. When d is high, fewer points of the minority class fall in E although $q(E)$ is fixed. Even though the classes are imbalanced, if the minority class have substantially small size, the class imbalance problem becomes more detrimental (Japkowicz and Stephen, 2002). Under the conditions that the data set has substantial imbalance and overlapping, AUC of RW-CCCD classifier is followed, in order by, the AUC of C4.5, SVM and k -NN classifiers.

Unlike the comparison of CCCD and SVM classifiers in Figures 5 and 8, SVM classifier has less AUC than CCCD classifiers with low δ and low q values in Figure 9. In this setting, n is fixed to 400 and the lowest value of m is 20. Compared to our experiments in Figures 5 and 8, this setting produces highly imbalanced data sets (one class has far

more observations than the other, $m \ll n$). Akbani et al. (2004) conducted a detailed investigation and listed some reasons of SVM classifier being sensitive to highly imbalanced UCI data sets (Bache and Lichman, 2013). They did not, however, address the problem of overlapping class supports but offered a modification to SMOTE algorithm in order to improve the robustness of SVM. On the other hand, especially for $d = 5$ and $d = 10$, SVM, k -NN and C4.5 classifiers have more AUC than CCCD classifiers with increasing q and decreasing δ . This may indicate that other weak classifiers are better than CCCD classifiers for balanced classes.

The effects of class imbalance might also be observed when the class supports are well separated. If the class supports are disjoint, that is $s(F_X) \cap s(F_Y) = \emptyset$, the AUC is fairly high. However, it might still be affected by the global imbalance level, q . Therefore, we simulate a data set with two classes where $F_X = U(0, 1)^d$ and $F_Y = U(1 + \delta, 2 + \delta) \times U(0, 1)^{d-1}$. Figure 10 illustrates the results of this simulation study. Both class supports are d dimensional unit boxes as in the previous simulation setting, however they are now disjoint (separated along the first dimension). In addition, the parameter δ controls the smallest distance between the class supports where $\delta = 0.05, 0.10, \dots, 0.45, 0.50$. With increasing δ , the points of class \mathcal{Y} move further away from the points of \mathcal{X} . Figure 10 illustrates the difference between AUCs of CCCD and other classifiers under this simulation setting.

In Figure 10, unlike the performance of CCCD classifiers in Figure 9, P-CCCD classifiers have more AUC than RW-CCCD classifiers. When classes are imbalanced and supports are close, P-CCCD classifiers outperform both SVM and k -NN classifiers for all d values, but RW-CCCD classifiers have nearly 0.03 more AUC than these classifiers only in $d = 10$. However, this is not the case with C4.5 classifier since none of the classifiers outperform C4.5; that is, C4.5 yields over 0.04 more AUC than CCCD classifiers. A well separated data set is more likely to be classified better with C4.5 tree classifier because a single separating line exists between the two class supports. Hence, C4.5 locates such a line and efficiently classifies points regardless of the distance between class supports as long as the distance is positive. On the other hand, the balls of P-CCCD classifiers establish appealing covers for the class supports because the supports do not overlap. P-CCCD classifiers establish covering balls, big enough to catch substantial amount of points from the same class. Similarly, RW-CCCD classifiers establish pure covers, and this is the result of the separation between class supports. However, P-CCCD classifiers achieve better classification performance than RW-CCCD classifiers. When the classes are well separated, the radii of a ball in random walk, say from class \mathcal{X} , is likely $\max_{z \in \mathcal{X}_n} d(z, x)$ but in P-CCCD classifiers, it is $\min_{z \in \mathcal{Y}_n} d(z, x)$. In fact, the RW-CCCD classifiers are nearly equivalent to P-CCCD classifiers. Thus, when $\tau > 0$, P-CCCD classifiers are more likely to produce bigger balls than RW-CCCD classifiers, and potentially avoid overfitting.

In Figure 10, RW-CCCD classifiers have slightly or considerably less AUC than other classifiers when data sets are imbalanced and the supports are slightly far away from each other. The random walk contaminates the class cover with some non-target class points to improve the classification performance. However, since the classes are well separated and one class has substantially fewer points than the other, random walks are likely to yield balls to cover some points from the support of the non-target class, resulting in a degradation in the performance of RW-CCCD classifiers. On the other hand, P-CCCD classifiers outperform both k -NN and SVM classifiers for lower q and lower δ . The closer and more imbalanced the

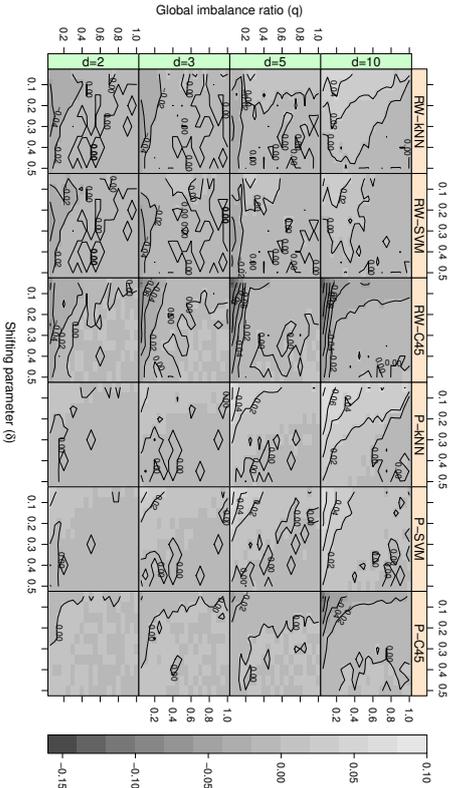


Figure 10: Differences between the AUCCs of CCCD and other classifiers (see Figure 9 for details). In this two-class setting, classes are drawn from $F_X = U(0, 1)^d$ and $F_Y = U(1 + \delta, 2 + \delta) \times U(0, 1)^{d-1}$ where $d = 2, 3, 5, 10$, $\delta = 0.05, 0.1, \dots, 0.45, 0.50$ and $q = 0.05, 0.1, \dots, 0.95, 1.00$ with $n = 400$ and $m = qn$. AUCCs of all classifiers are over 88% since the class supports are well separated.

data, the better the performance of P-COCEDs than other classifiers. Although the classes do not overlap, the effect of class imbalance is still observed when the supports are close. When there is mild imbalance between classes, COCCD classifiers have either comparable or less AUC. In addition, note that the performances of SVM and k -NN classifiers deteriorate but P-COCED classifiers preserve their AUC with increasing d . Let $E \subset \mathbb{R}^d$ be some region that contains points of both classes which are sufficiently close to the decision boundary. With increasing d , fewer minority class points are in this region, and hence fewer members of this class fall in E . As a result, the performance of both SVM and k -NN classifiers suffer from local class imbalance with respect to E .

Finally, we investigate the effect of dimensionality when classes are balanced (i.e., $q = 1$) and their supports are overlapping. In this setting, $F_X = U(0, 1)^d$ and $F_Y = U(\delta, 1 + \delta)^d$. Here, let $q(E) \approx q = 1$, hence the classes are also locally balanced with respect to E as well as being globally balanced. Also, δ controls the level of overlap between two classes. However, we define δ in such a way that the overlapping ratio $\alpha \in [0, 1]$ is fixed for all dimensions. When α is 0, the supports are well separated, and when α is 1, the supports of classes are the same, i.e., $s(F_X) = s(F_Y)$. The closer α to 1, the more the supports overlap. Observe that $\delta \in [0, 1]$ can be expressed in terms of the overlapping ratio α and

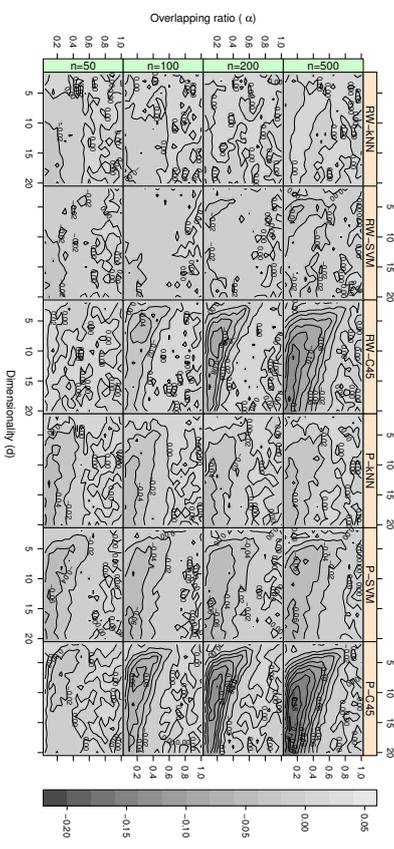


Figure 11: Differences between the AUCCs of CCCD and other classifiers (see Figure 9 for details). In this two-class setting, classes are drawn from $F_X = U(0, 1)^d$ and $F_Y = U(\delta, 1 + \delta)^d$ where $n = 50, 100, 200, 500$, $\alpha = 0.05, 0.1, \dots, 0.45, 1.00$ and $d = 2, 3, 4, \dots, 20$.

dimensionality d :

$$\alpha = \frac{\text{Vol}(s(F_X) \cap s(F_Y))}{\text{Vol}(s(F_X) \cup s(F_Y))} = \frac{(1 - \delta)^d}{2 - (1 - \delta)^d} \iff \delta = 1 - \left(\frac{2\alpha}{1 + \alpha} \right)^{1/d}. \quad (6)$$

Hence, we calculate δ for each (d, α) combination by the Equation (6). In Figure 11, each cell of the grey scale heat map corresponds to a single combination of simulation parameters $\alpha = 0.05, 0.1, \dots, 0.95, 1.00$ and $d = 2, 3, 4, \dots, 20$. In Figure 11, the differences between the AUCCs of COCCD classifiers and other classifiers are up to 0.20. The k -NN and SVM classifiers have comparable performance with COCCD classifiers, or outperform both COCCD classifiers. However, C4.5 has more AUC with increasing d . Employing COCCD classifiers do not considerably increase the classification performance over other classifiers when classes are balanced.

5.2 Empirical Comparison of COCCD-based and Strong Classifiers

In this section, we compare the COCCD-based classifiers with strong versions of k -NN, SVM and C4.5 classifiers on simulated data sets. Each classifier is modified in three different schemes, namely, resampling, ensemble and cost-sensitive schemes. We use SMOTE+ENN algorithm as the resampling scheme and EasyEnsemble algorithm as the ensemble scheme. As for the cost sensitive versions on weak classifier, we adjust the classifiers into recognizing class weights. For k -NN, we employ an algorithm giving more weight on neighboring minority class members; for SVM, we use two separate constrained violation costs for each

Method	Description
SMOTE+ENN	A combination of SMOTE ($t = 2$ and $k = 5$) and ENN ($k = 3$) (Batista et al., 2004).
EasyEnsemble	A combination of undersampling ($T = 4$) and Adaboost ($s_i = 10$) for $i = 1, 2, \dots, T$ (Liu et al., 2006).
C5.0	The cost sensitive version of C4.5 (Kuhn and Johnson, 2013).
CKNN	A cost sensitive version of k -NN (Barandela et al., 2003).
CSVM	A cost sensitive version of SVM (Chang and Lin, 2011).

Table 2: The description of classifiers employed in the article.

corresponding class; and for C4.5, we employ the C5.0. With three schemes and three weak classifiers, we get nine strong classifiers to study. We list and describe all these schemes in Table 2.

SMOTE+ENN algorithm, first, oversamples the entire training data set by generating artificial points in between a point and its neighbors. Specifically, for each point in the data set, t points among k neighbors are selected, and until the data set is balanced, new artificial points are generated in between these points and their selected neighbors. Later, ENN algorithm cleans the data set of noisy points by checking all points if the majority of their k neighbors are labeled as the class of the point. If not, the point is erased from the data set. Simply, SMOTE+ENN is a hybrid of over and undersampling methods. EasyEnsemble algorithm is a hybrid of undersampling and ensemble methods. An ensemble of weak classifiers is established by generating T many undersampled balanced data sets from the training data set. Then, each data set is used to train individual Adaboost classifiers with s_i many weak classifiers for $i = 1, 2, \dots, T$. Hence, EasyEnsemble is an ensemble of $\sum_{i=1}^T s_i$ many weak classifiers.

We choose one of the simulation settings conducted in Section 5.1. Since CCCD classifiers are observed to be better than other classifiers when both class imbalance and overlapping occurred, we only compare CCCD classifiers with strong classifiers on a single simulation setting. Hence we choose the setting presented in Figure 9, i.e., we let $F_X = U(0, 1)^d$ and $F_Y = U(\delta, 1 + \delta)^d$ for $\text{bura } \delta, q = 0.05, 0.10, \dots, 0.95, 1.00, n = 400$ and $m = qn$. We aim to highlight the differences between the strong classifiers and CCCD classifiers for various combinations of overlapping and class imbalance ratios. The results on average AUCs of each strong classifier is given in Figure 12. In general, RW-CCCDs seem to perform better than P-CCCDs. For $d > 2$, P-CCCDs have nearly 0.10 less AUC than RW-CCCDs when the classes are substantially overlapping and imbalanced, and it is observed that P-CCCDs are usually worse compared to the strong classifiers considered. However, the AUCs of RW-CCCD classifiers are either comparable or slightly less compared to others with the most difference being seen in the case of $d = 10$ when RW-CCCDs compared to EC4.5 and C5.0, ensemble and cost sensitive versions of the C4.5 classifier, respectively. However, with decreasing δ and q , the RW-CCCDs have only 0.05 less AUC than others. Also, RW-CCCDs seem to have 0.05 more AUC than C5.0 for moderately overlapping and imbalanced data sets, and seem to have 0.05 more AUC than EC4.5, ensemble based SVMs, when the data set is both overlapping and imbalanced. This suggests that RW-CCCDs yield comparable results in comparison to the state-of-the-art robust methods when class imbalance and class overlapping co-exist. Additionally, we show in Section 5.3 that RW-CCCDs generate prototype sets that considerably reduce the training data set.

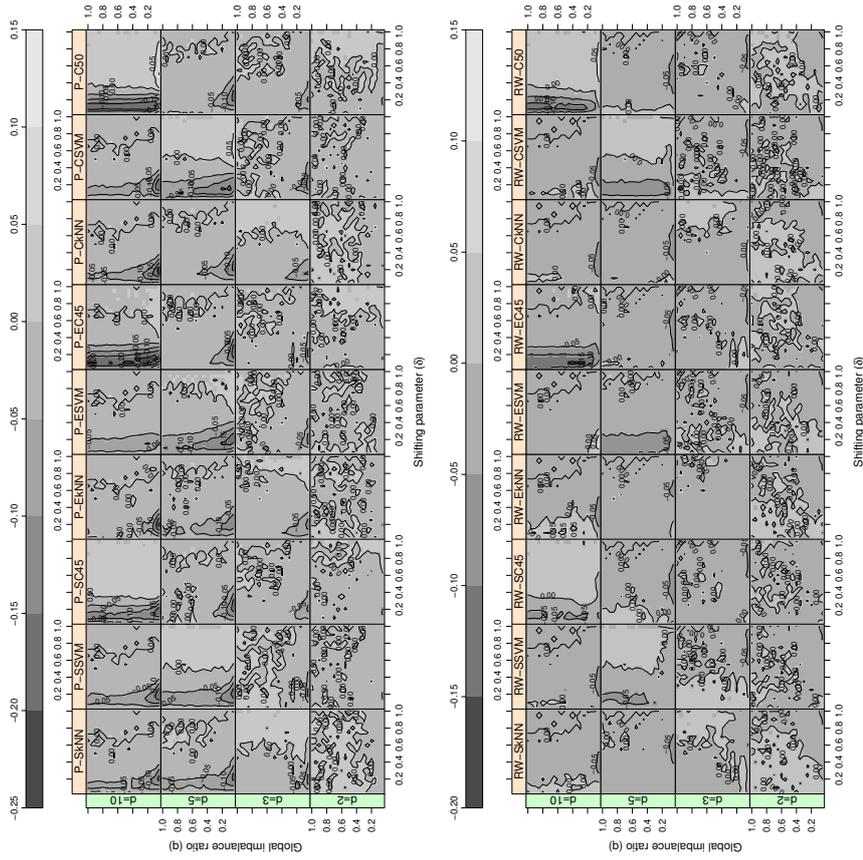


Figure 12: Differences between the AUCs of CCCD and other classifiers (see Figure 9 for details). P-CCCD in (top) and RW-CCCD in (bottom). Here, resampling scheme strong classifiers are coded with “S”, ensemble schemes with “E”, and cost sensitive schemes with “C”. For example, “SKNN” refers to the resampling scheme k -NN classifier. In this two-class setting, classes are drawn from $F_X = U(0, 1)^d$ and $F_Y = U(\delta, 1 + \delta)^d$ where $d = 2, 3, 5, 10, q, \delta = 0.05, 0.1, \dots, 0.95, 1.00$ with $n = 400$ and $m = qn$.

	Training		Testing	
	Time	Space	Time	Space
P-CCCD	$\mathcal{O}(N^2d)$	$\mathcal{O}(N^2)$	$\mathcal{O}(Nd)$	$\mathcal{O}(Nd)$
RW-CCCD	$\mathcal{O}(N^3 + N^2d)$	$\mathcal{O}(N^2)$	$\mathcal{O}(Nd)$	$\mathcal{O}(Nd)$
k -NN	\dots	\dots	$\mathcal{O}(Nd)$	$\mathcal{O}(Nd)$
SVM	$\mathcal{O}(N^3 + N^2d)$	$\mathcal{O}(N^2)$	$\mathcal{O}(Nd)$	$\mathcal{O}(Nd)$
C4.5-LNP	$\mathcal{O}(Nd^2)$	$\mathcal{O}(Nd)$	$\mathcal{O}(d)$	$\mathcal{O}(2^d)$

Table 3: Training and testing space and time complexities of the weak classifiers.

5.3 Complexity Analysis of the Classifiers

In Table 3, we compare training and testing time and space complexities of P-CCCDs, RW-CCCDs, k -NN, SVM and C4.5 classifiers. Let $N = n + m$ be the size of training data set. C4.5 is the fastest among all classifiers and requires the least space. However, unpruned C4.5 constitute a tree with its space complexity increasing exponentially on d since the data set is divided into at most two for all dimensions. The remaining classifiers are all instance based learning methods which depend on a matrix of distances between the points of training data set. Hence, their space complexity is at least $\mathcal{O}(N^2)$ and they run in at least $\mathcal{O}(N^2d)$ time. Both SVM and RW-CCCD classifiers run in $\mathcal{O}(N^3)$ time for $d < N$, and P-CCCD runs in $\mathcal{O}(N^2d)$ time. Minimum dominating set problem of P-CCCDs are polynomial time reducible to minimum set cover problems, and hence they run in $\mathcal{O}(N^2)$ time in the worst case but they require the computation of the distance matrix which takes the most time. However, in RW-CCCDs, covering balls are re-defined each time a new point is added to the prototype set. As a result, this operation requires an additional sweep on the training set on each iteration which makes RW-CCCD run in $\mathcal{O}(N^3)$ time, for $d < n$. For SVM, the training time of usual optimization algorithms is $\mathcal{O}(N^3)$ for $d < n$. However, it is possible to reduce the complexity to $\mathcal{O}(N^{2.3})$ with sequential minimal optimization (SMO) method (Chang and Lin, 2011).

Note that, k -NN does not require any training time or space, and should use the entire training data set to classify the test data set. However, CCCD and SVM classifiers reduce the training data set by means of prototype sets (minimum dominating sets in CCCD and support vectors in SVMs) even though their worst case testing space complexity is $\mathcal{O}(Nd)$. The entire training data set could be chosen as the prototype set for some cases, but we show that the data set is substantially reduced when the classes are imbalanced. In Figure 13, we compare the sizes of the set of prototypes in RW-CCCDs and the set of support vectors in SVM and GSYM classifiers. We consider the simulation settings with two classes for $F_X = U(0, 1)^d$ and $F_Y = U(\delta, 1 + \delta)^d$, $\delta = 0.1, 0.4, 0.7, 1.0$, $d = 2, 3, 5, 10$, $n = 400$ and $m = qn$.

In Figure 13, the number of both support vectors and prototypes decrease with increasing δ . The prototype set heavily depends on the overlapping ratio between class supports. Undoubtedly, when points of either class are further away from each other, covering balls get bigger for CCCDs, and the separating hyperplane requires less support vectors. On the other hand, observe that the number of support vectors are much higher than the number of prototypes of RW-CCCDs. The number of support vectors decreases with decreasing q .

The more imbalanced the data set, the fewer support vectors are generated. But in any case, RW-CCCDs still reduce the training data set more than SVMs. In Figure 14, we compare the number of prototypes in both CCCD classifier and the size of the C4.5 and C5.0 classifier trees for the same simulation setting.

The number of prototypes in P-CCCDs are, in general, much higher than that of other classifiers. Also, notice that the less imbalanced the classes are, the less the data reduction in P-CCCDs. However, there is not much change in the number in RW-CCCDs, C4.5 and C5.0, and since the size of trees grows exponentially on d , the size of trees get bigger than the size of CCCDs for some substantially high δ and d . Moreover, the size of trees in C5.0 is considerably less than that in C4.5 (Kuhn and Johnson, 2013). Although the number of prototypes are much higher than the size of trees in highly overlapped and imbalanced cases, RW-CCCDs reduce the training set substantially more than C4.5 and C5.0 in moderately imbalanced and moderately overlapped higher dimensional settings.

5.4 Real Data Examples

In this section, we compare the performance of CCCD classifiers and all other weak and strong classifiers on several data sets from UC Irvine (UCI) Machine Learning and KEEL repositories (Bache and Lichman, 2013; Alcalá-Fdez et al., 2011). To test the difference between the AUC of classifiers, we employ the 5x2 cross validation (CV) paired t -test (see Dietterich, 1998) and the combined 5x2 CV F -test (see Alpaydm, 1999). The 5x2 CV test has been devised by Dietterich (1998) and found to be the most powerful test among those with acceptable type-I error. However, the test statistics of 5x2 t -tests depend on which one of the ten folds is used. Hence, Alpaydm (1999) offered a combined 5x2 CV F -test which works as an omnibus test for all ten possible 5x2 t -tests (for each five repetitions there are two folds, hence ten folds in total). Basically, if a majority of ten 5x2 t -tests suggests that two classifiers are significantly different in terms of performance, the F -test also suggests a significant difference. Hence, an F -test with high p -value suggests that some of the ten t -tests fail to have low p -values.

We also provide the overlapping ratios and imbalance levels of these data sets. In a simulation study such as the one in Section 5.1, we have control on the overlapping region of two classes since we can choose the supports of the classes, hence their overlapping region is exactly known. However, in real data sets where the support of classes are neither defined nor available, we need methods to estimate the supports and hence estimate the overlapping ratios for the two classes. We employ the support vector data description (SVDD) method of Tax and Dunn (2004) for this purpose. The method finds a description (or a region) of a data set, which covers a desired percentage of the points. SVDDs are also used in novelty or outlier detection. It has been inspired by the SVM classifiers and is based on defining a sphere around the data set. Similar to SVM, kernel functions can be employed to define more relaxed regions. SVDD is also a one-class learning method where the goal is to decide if a new point belongs to this particular class or not (Juszczak et al., 2002). By using SVDD approach, Xiong et al. (2010) found the SVDD regions of each class and its overlapping region. We also use SVDD to find the overlapping region E of each pair and report on the imbalance ratio with respect to E . The overlapping ratio is the percentage of points from both classes that reside in A . We use the Dtools toolbox (Tax, 2014) of

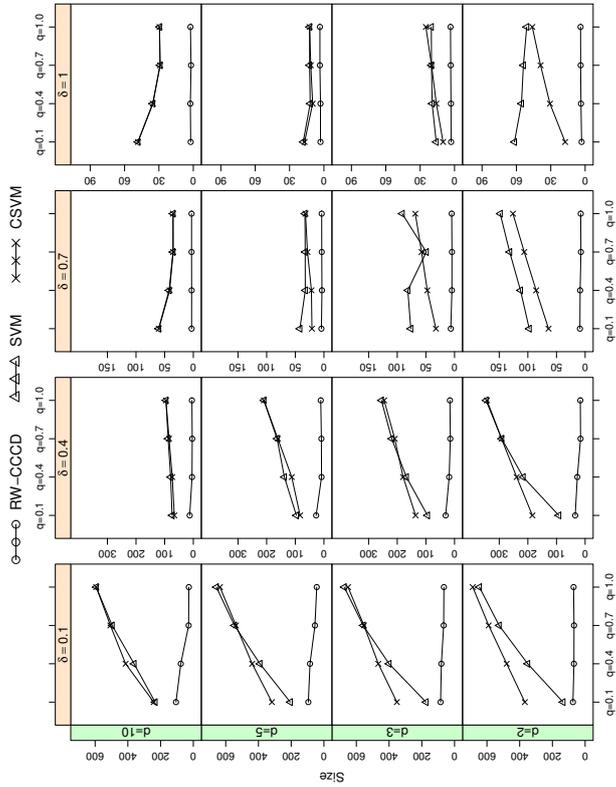


Figure 13: Comparison of the sizes of reduced data sets in RW-CCCDs, SVM and CSMV classifiers. Here “size” refers to the number of covering balls in RW-CCCD or the number of support vectors in SVM classifiers. In this two-class setting, classes are drawn from $F_X = U(0, 1)^d$ and $F_Y = U(\delta, 1 + \delta)^d$ where $\delta, q = 0.1, 0.4, 0.7, 1.0$ with $n = 400$ and $m = qn$.

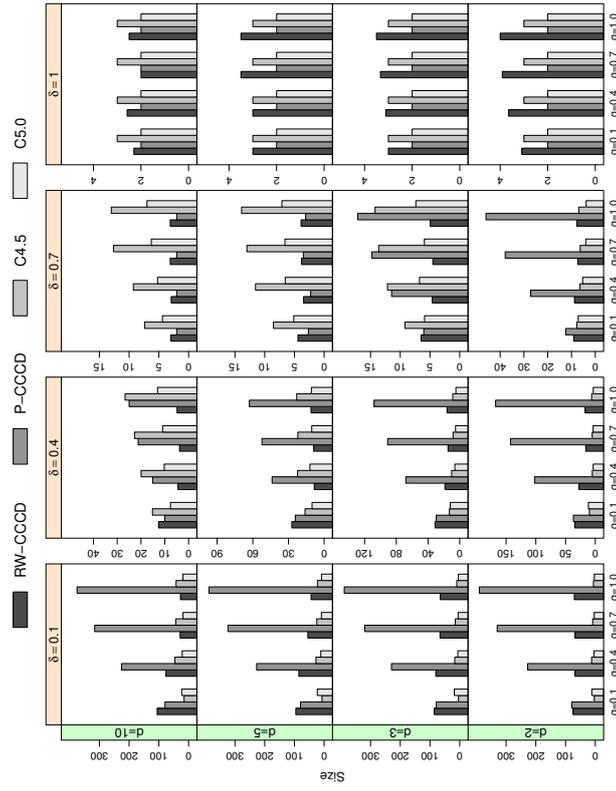


Figure 14: Comparison of the sizes of reduced data sets in CCCDs and C4.5. Here “size” refers to the number of covering balls in CCCDs or the number of nodes in the decision tree of C4.5 classifiers. The simulation setting is same as in Figure 13.

Data	$q = m/n$	N	d	$\sigma = 2$	$\sigma = 3$	$\sigma = 4$	$\sigma = 5$	$\sigma = 6$	$\sigma = 7$	$\sigma = 8$	$\sigma = 9$	$\sigma = 10$
Sowar	1.14	208	61	OR	4%	19%	23%	25%	26%	26%	27%	28%
				IR	1.22	1.04	0.96	0.93	0.96	0.97	0.97	0.96
Ionosphere	1.78	351	35	OR	25%	36%	66%	69%	66%	79%	61%	76%
				IR	90.00	62.50	8.70	6.20	5.44	3.67	5.02	3.98
Segment0	6.02	2308	20	OR	0%	0%	0%	0%	0%	0%	0%	0%
				IR	NA							
Page-Block0	8.79	5472	11	OR	0.6%	0.6%	0.6%	0.8%	0.5%	1%	1%	1%
				IR	0.47	0.22	0.22	0.25	0.40	0.39	0.43	0.51
Vow0	9.98	988	14	OR	0%	0%	0%	0%	0%	0%	0%	0%
				IR	NA							
Shuttle0v4	13.87	1829	10	OR	0%	0%	0%	0%	0%	0%	0%	0%
				IR	NA							
Yeast4	28.10	1484	9	OR	45%	27%	39%	37%	26%	26%	26%	31%
				IR	18.97	99.75	18.50	18.24	392.00	390.00	391.00	390.00
Yeast1289vs7	30.70	917	9	OR	45%	44%	69%	43%	30%	29%	29%	29%
				IR	24.47	23.76	24.34	23.00	47.33	45.83	45.66	45.50
Yeast5	32.70	1484	9	OR	NA							
				IR	NA	1.15	NA	NA	0%	0%	6%	7%
Yeast6	41.40	1484	9	OR	30%	46%	42%	31%	30%	30%	10%	13%
				IR	64.14	21.03	27.59	76.33	73.66	73.66	38.25	69.00
Abalone19	129.40	4174	9	OR	25%	20%	15%	14%	13%	12%	11%	11%
				IR	104.30	104.30	163.75	197.33	278.00	262.50	253.00	244.00

Table 4: Overlapping ratios and (local) imbalance ratios in the overlapping region of data sets. ‘‘IR’’ stands for the imbalance ratio in the overlapping region and ‘‘OR’’ stands for the overlapping ratio which is the percentage of points from both classes residing in the overlapping region. IR=‘‘NA’’ indicates that one of the classes has no members in the intersections of SVDD regions of classes.

MATLAB environment to produce the SVDDs of classes. Our choice of the kernel is the same as we have used with SVM classifiers in this study, the radial basis (i.e., Gaussian) kernel, for consistency. However, the selection of σ in the kernel is crucial for the SVDD region.

In Table 4, we present the overlapping ratios and the imbalance in the overlapping areas of all data sets for $\sigma = 2, 3, \dots, 10$. Although the value of σ produces different overlapping ratios, it is apparent that classes of data sets Ionosphere, Abalone19, Yeast4, Yeast6 and Yeast1289vs7 have more overlap than others, and these overlapping data sets have substantial local class imbalance in their respective overlapping regions. Other data sets have almost no overlapping nor imbalance in the overlapping regions even though their classes are globally imbalanced. One of these data sets is Yeast5 which has a imbalance ratio of $q = 32.70$ but has no imbalance in the small overlapping region.

In Table 5, we give the average AUC measures and their standard deviations of all CCCD-based and other classifiers according to the 5x2 CV scheme for the data sets. All other classifiers, weak or strong, have been two-way tested with 5x2 CV F -test against both RW-CCCD and P-CCCD classifiers. Their p -values are also provided in Table 5. For each of five repetitions, we divide the data into two folds. The AUC of fold 1 is given by using fold 1 as a training set and fold 2 as the test set. For fold 2, the process is similar. We repeated these experiments five times for all three classifiers. Looking at results from 11 data sets, RW-CCCD usually performs better than P-CCCD classifiers, and in addition,

ensemble based classifiers perform the best in general. Moreover, ensemble classifiers seem to perform better than RW-CCCDs but this difference is usually not significant, meaning RW-CCCDs perform comparable to ensemble classifiers in more than few folds of all ten folds. For example, compared to ensemble methods, RW-CCCD has nearly 0.07 less AUC in Yeast5, 0.02 less AUC in Yeast6, and 0.1 less AUC in Yeast1289vs7 data set. The difference is significant, however, with the data set Abalone19 with a level of < 0.03 . Although RW-CCCD achieves an average AUC value 0.6, ensemble classifiers achieve over 0.7. On the other hand, there is no significant difference between AUCs of RW-CCCD and ensembles in other highly overlapped and locally imbalanced data sets. On these data sets, RW-CCCD have significantly more AUC than weak classifiers and have AUC comparable to strong classifiers. Thus, these results from real data sets seem to resonate with the results from our simulations and further support the robustness of CCCD classifiers to the class imbalance problem.

6. Summary and Discussion

We assess the classification performance of various classifiers such as RW-CCCD, pure-CCCD, k -NN, SVM and C4.5 classifiers and their variants when class imbalance occurs, and we illustrate the robustness of CCCD classifiers to the class imbalance in data sets. This imbalance often occurs in real life data sets where, in two-class settings, minority class (the class with fewer number of observations) is usually dwarfed by the majority class. Class imbalances hinder the performance of many classification algorithms. We studied the performance of CCCD classifiers under class imbalance problem by first simulating a two-class setting similar to the one used in DeVinney (2003). In this setting, the support of one class is entirely embedded in the support of the other. Drawing equal number of observations from both class supports results in an imbalance between two classes with respect to their overlapping region, called *local (or restricted) class imbalance*. This difference in the class sizes was also the case in the example of DeVinney (2003), and it is the reason that CCCD classifiers show better results than the k -NN classifier. We show that P-CCCD classifiers with lower τ values tend to perform better than the ones with higher τ values. This is merely a result of balls with $\tau = 0$ representing the local density of the target class points better. Similarly, the RW-CCCD classifiers with lower e values are better when the dimensionality is low and the class sizes are high. This might indicate that the denser the data set in \mathbb{R}^d , the less useful the scores T_x . However, fully utilizing the scores usually increases the classification performance.

Analysis of both simulated and real data sets indicate that both CCCD classifiers show robustness to the class imbalance problem. We demonstrated this by studying the effects of the class overlapping problem together with the class imbalance problem. In fact, there are studies in the literature focusing on the performance of classification methods when class overlapping and class imbalance problems occur simultaneously (Prati et al., 2004; Demir and Trautenberg, 2010). Overlapping of classes is an important factor in the classification of imbalanced data sets; that is, it drastically affects the classification performance of most algorithms. When classes are both imbalanced and overlapping, performance of k -NN, SVM and C4.5 classifiers deteriorate whereas CCCD classifiers are not affected as severely as these methods. We use two alternatives of C4.5 classifiers where we prune the decision

Dataset	Method	AUC	p -value (vs RW)	p -value (vs P)
LongSphere	AUC	0.917±0.023	0.722±0.050	0.866±0.051
	p -value (vs P)	0.005	0.005	0.005
Sour	AUC	0.803±0.019	0.804±0.027	0.786±0.032
	p -value (vs P)	0.000	0.000	0.000
Year6	AUC	0.898±0.051	0.898±0.063	0.807±0.048
	p -value (vs P)	0.000	0.000	0.000
Year3	AUC	0.898±0.051	0.898±0.063	0.807±0.048
	p -value (vs P)	0.000	0.000	0.000
Year4	AUC	0.898±0.051	0.898±0.063	0.807±0.048
	p -value (vs P)	0.000	0.000	0.000
Year112987	AUC	0.898±0.051	0.898±0.063	0.807±0.048
	p -value (vs P)	0.000	0.000	0.000
Vow0	AUC	0.898±0.051	0.898±0.063	0.807±0.048
	p -value (vs P)	0.000	0.000	0.000
Shuttle04	AUC	0.898±0.051	0.898±0.063	0.807±0.048
	p -value (vs P)	0.000	0.000	0.000
Abalone19	AUC	0.898±0.051	0.898±0.063	0.807±0.048
	p -value (vs P)	0.000	0.000	0.000
Segment0	AUC	0.898±0.051	0.898±0.063	0.807±0.048
	p -value (vs P)	0.000	0.000	0.000
Page-Blocks	AUC	0.898±0.051	0.898±0.063	0.807±0.048
	p -value (vs P)	0.000	0.000	0.000
k-NN	AUC	0.803±0.019	0.804±0.027	0.786±0.032
	p -value (vs RW)	0.000	0.000	0.000
P-CCCD	AUC	0.934±0.032	0.805±0.045	0.755±0.053
	p -value (vs P)	0.000	0.000	0.000
RW-CCCD	AUC	0.917±0.023	0.722±0.050	0.866±0.051
	p -value (vs P)	0.005	0.005	0.005
k-NN	AUC	0.834±0.024	0.809±0.023	0.871±0.039
	p -value (vs RW)	0.008	0.008	0.008
k-NN	AUC	0.834±0.024	0.809±0.023	0.871±0.039
	p -value (vs P)	0.013	0.013	0.013
EC4.5	AUC	0.834±0.024	0.809±0.023	0.871±0.039
	p -value (vs RW)	0.008	0.008	0.008
EC4.5	AUC	0.834±0.024	0.809±0.023	0.871±0.039
	p -value (vs P)	0.013	0.013	0.013
ESVM	AUC	0.834±0.024	0.809±0.023	0.871±0.039
	p -value (vs RW)	0.008	0.008	0.008
ESVM	AUC	0.834±0.024	0.809±0.023	0.871±0.039
	p -value (vs P)	0.013	0.013	0.013
CSVM	AUC	0.834±0.024	0.809±0.023	0.871±0.039
	p -value (vs RW)	0.008	0.008	0.008
CSVM	AUC	0.834±0.024	0.809±0.023	0.871±0.039
	p -value (vs P)	0.013	0.013	0.013
CS.0	AUC	0.834±0.024	0.809±0.023	0.871±0.039
	p -value (vs RW)	0.008	0.008	0.008
CS.0	AUC	0.834±0.024	0.809±0.023	0.871±0.039
	p -value (vs P)	0.013	0.013	0.013

Table 5: Average of AUC values of ten folds, and standard deviations, of CCCD, weak and strong classifiers for data sets. The p -values of 5x2 CV F -tests show the results of two-way tests comparing both CCCDs with other classifiers. Some of best performers are given in bold.

tree in one and do not in the other. It is known for some time that pruning deteriorates the performance of tree classifiers under class imbalance. Moreover, SVM is robust to moderately imbalanced class sizes but demonstrates no robustness in highly imbalanced cases. However, whether the data set is highly or moderately imbalanced, CCCD classifiers seem to preserve their AUC compared to k -NN, SVM and C4.5 classifiers. Hence, our study suggests that CCCD classifiers are appealing alternatives when data have class imbalance. In addition, we mention the effect of the individual class sizes on the class imbalance problem (Japkowicz and Stephen, 2002). Whatever the ratio between class sizes is, if the minority class has a substantially high number of points, the effect of imbalances between classes tend to diminish.

The classifiers k -NN, SVM and C4.5 are referred to as weak classifiers since, by construction, they are sensitive to imbalances between classes in data sets. In addition, we consider three distinct families of methods to establish strong classifiers based on weak classifiers, and compare them with CCCD classifiers. We conduct simulation studies to determine how the classification performance jointly depends on both (global) class imbalance and class overlapping, parameterized as q and δ , respectively. Finally, we apply all these classifiers on several UCI and KEEL data sets. By using the SVDD method of Tax and Duin (2004), we estimated the overlapping ratios of all these data sets. We show that CCCD classifiers outperform or perform comparable to k -NN, SVM and C4.5 classifiers for some overlapping and imbalance ratios in both simulated and real data sets. In particular, CCCDs are better than SVM classifiers in highly imbalanced cases. The effect of high class imbalance on SVM classifier is also studied in Akbani et al. (2004) and Raskutti and Kowalczyk (2004). However, when no imbalance occurs between classes, CCCD classifiers usually show either comparable or slightly worse performance than the other classifiers. As for strong classifiers, we employ the most successful methods from three families of schemes where EasyEnsemble and SMOT-ENN methods are among them. In our simulation studies, we demonstrated that CCCD classifiers, especially RW-CCCDs, work well compared to these strong classifiers when there are considerable overlap and the high (local) imbalance between classes. However, these methods are slightly better than RW-CCCDs as these strong classifiers are the best performing ones among their respective families (Batista et al., 2004; López et al., 2013). Nevertheless, RW-CCCDs have still high performance compared to these classifiers with additional increase in testing speed.

We also investigate the performance of CCCD classifiers under different conditions. Specifically, in two different experiments, we simulate two classes where (i) classes are imbalanced but supports are not overlapping (well separated) and (ii) classes are balanced and supports are overlapping with increasing dimensionality. P-CCCD classifiers are better than RW-CCCD classifiers in experiment (i). Both CCCD classifiers mostly outperform k -NN and SVM classifiers when classes are imbalanced and not overlapping, however RW-CCCD classifiers outperform these classifiers only when dimensionality is sufficiently high. In experiment (ii), the classification performance of CCCD classifiers slightly degrade compared to k -NN and SVM classifiers, especially with increasing d . Among CCCD classifiers, random walk covers appear to be better when classes are both overlapping and imbalanced, however our results suggest the use of pure covers when classes are imbalanced and well separated (i.e., not overlapping). In fact, class supports are often overlapping in real life data sets, hence RW-CCCD classifiers seem to be more appealing in practice.

In practice, classifiers based on CCCD classifiers resemble prototype selection methods. CCCDs balance the class sizes by defining balls that catch surrounding points of the same class, and discard these points from the training set. The resulting data set is composed of the centers of these balls and associated radii which are used in scaled dissimilarity measures. Although, CCCD classifiers remove substantial amount of observations from the majority class, they preserve (most of) the information with the radii. The bigger the radius, the more likely that the balls of CCCD classifiers contain more points. The radii could be considered as an indicator of the local density of the target class. The real advantage of CCCD classifiers are these prototype sets which are of (approximately) minimum cardinality, although training time and space of P-CCCDs and RW-CCCDs may be considerably high. However, the number of points in the prototype set is substantially low, and hence testing speed is increased. In some cases, RW-CCCDs provide classifiers with the least testing space complexity. Only the decision tree based classifiers, C4.5 and C5.0, achieve comparable or slightly more reduction to RW-CCCDs. However, with increasing dimensionality, sizes of these trees grow exponentially, making them less appealing than RW-CCCDs in the sense of classification space complexity. Hence, CCCDs preserve important information regarding the data sets while substantially increasing the testing speed. In literature, many classifiers have been devised to preserve the information on the deleted majority class points, however they are all ensemble based classifiers which substantially increase both training and testing time complexities. In that regard, CCCDs offer a novel approach to this particular problem.

Eveland et al. (2005) modified RW-CCCD classifiers as to increase the speed of the face detection in which imbalances between classes occur naturally. They did only refer to the real life applications which consist of class imbalances. They did not, however, investigate the relationship between class imbalance and overlapping problems as thoroughly as our study does. On the other hand, establishing class covers with Euclidean balls raise the possibility of using different regions (the regions are Euclidean hyperballs around target class points in CCCD) to balance the data and, thus, construct non-parametric classifiers with more classification performance. Along this line, CCCDs can be generalized using *proximity maps* (Jaromczyk and Toussaint, 1992). For example, Ceyhan (2005) defined *proximity catch digraphs* (PCDs) that are generalized versions of CCCDs. Ceyhan (2005) has introduced three families of PCDs and used them to test spatial data patterns of segregation and association (see Ceyhan and Priebe, 2005; Ceyhan et al., 2006, 2007). PCDs can also be used to derive new graph-based classifiers which are potentially robust to the class imbalance problem. The study of their properties and performance is a topic of ongoing research by the authors.

Acknowledgments

Most of the Monte Carlo simulations presented in this article were executed at Koç University High Performance Computing Laboratory. This research was supported by the European Commission under the Marie Curie International Outgoing Fellowship Programme via Project # 329370 titled PRimHDD.

References

- R. Akpani, S. Kwek, and N. Japkowicz. Applying support vector machines to imbalanced datasets. In *Proceedings of 15th European Conference on Machine Learning*, pages 39–50, Pisa, Italy, 2004.
- J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, and S. García. KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Multiple-Valued Logic and Soft Computing*, 17(2-3):255–287, 2011.
- E. Alpaydm. Combined 5×2 cv F test for comparing supervised classification learning algorithms. *Neural Computation*, 11(8):1885–1892, 1999.
- K. Bache and M. Lichman. UCI machine learning repository, 2013. URL <http://archive.ics.utcl.edu/ml>.
- R. Barandela, J. S. Sánchez, V. García, and E. Rangel. Strategies for learning in class imbalance problems. *Pattern Recognition*, 36(3):849–851, 2003.
- G. E. Batista, R. C. Prati, and M. C. Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 6(1):20–29, 2004.
- G. E. Batista, R. C. Prati, and M. C. Monard. Balancing strategies and class overlapping. In *Proceedings of 6th International Symposium on Intelligent Data Analysis: Advances in Intelligent Data Analysis VI*, pages 24–35, Madrid, Spain, 2005.
- S. Bereg, S. Cabello, J. M. Díaz-Báñez, P. Pérez-Lantero, C. Seara, and I. Ventura. The class cover problem with boxes. *Computational Geometry*, 45(7):294–304, 2012.
- A. H. Cannon and L. J. Cowen. Approximation algorithms for the class cover problem. *Annals of Mathematics and Artificial Intelligence*, 40(3):215–223, 2004.
- E. Ceyhan. *An investigation of proximity catch digraphs in Delaunay tessellations*. PhD thesis, Johns Hopkins University, Baltimore, MD, USA, 2005.
- E. Ceyhan and C. E. Priebe. The use of domination number of a random proximity catch digraph for testing spatial patterns of segregation and association. *Statistics & Probability Letters*, 73(1):37–50, 2005.
- E. Ceyhan, C. E. Priebe, and J. C. Wiernman. Relative density of the random t-factor proximity catch digraph for testing spatial patterns of segregation and association. *Computational Statistics & Data Analysis*, 50(8):1925–1964, 2006.
- E. Ceyhan, C. E. Priebe, and D. J. Marchette. A new family of random graphs for testing spatial segregation. *Canadian Journal of Statistics*, 35(1):27–50, 2007.
- C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

- N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16(1):321–357, 2002.
- N. V. Chawla, N. Japkowicz, and A. Kotcz. Editorial: Special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, 6(1):1–6, 2004.
- V. Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979.
- D. A. Cieslak and N. V. Chawla. Learning decision trees for unbalanced data. In *Proceedings of the ECML PKDD 2008 Machine Learning and Knowledge Discovery in Databases: European Conference*, pages 241–256, Antwerp, Belgium, 2008.
- T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.
- M. Denil and T. Trappenberg. Overlap versus imbalance. In *Proceedings of the 23rd Canadian Conference on Advances in Artificial Intelligence*, pages 220–231, Berlin, Heidelberg, 2010.
- J. DeVinney, C. Priebe, D. Marchette, and D. Socolinsky. Random walks and catch digraphs in classification. In *Proceedings of the 34th Symposium on the Interface, Volume 34: Computing Science and Statistics*, Montreal, Quebec, Canada, 2002.
- J. G. DeVinney. *The class cover problem and its application in pattern recognition*. PhD thesis, Johns Hopkins University, Baltimore, MD, USA, 2003.
- T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, 1998.
- P. Domingos. MetaCost: A general method for making classifiers cost-sensitive. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '99, pages 155–164, New York, NY, USA, 1999.
- C. Drummond, R. C. Holte, et al. C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Workshop on Learning from Imbalanced Datasets (II)*, Washington DC, USA, 2003.
- C. Elkan. The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 973–978, Melbourne, Australia, 2001.
- C. K. Eveland, D. A. Socolinsky, C. E. Priebe, and D. J. Marchette. A hierarchical methodology for class detection problems with skewed priors. *Journal of Classification*, 22(1):17–48, 2005.
- M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim. Do we need hundreds of classifiers to solve real world classification problems. *Journal of Machine Learning Research*, 15(1):3133–3181, 2014.
- E. Fix and J. L. Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review*, 57(3):238–247, 1989.
- Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.
- M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera. A review on ensembles for the class imbalance problem: Bagging, boosting, and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 42(4):463–484, 2012.
- V. García, R. A. Molineda, and J. S. Sánchez. On the k -NN performance in a challenging scenario of imbalance and overlapping. *Pattern Analysis and Applications*, 11(3-4):269–280, 2008.
- H. Han, W.-Y. Wang, and B.-H. Mao. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In *Proceedings of the 2005 International Conference on Advances in Intelligent Computing - Volume Part I*, pages 878–887, Berlin, Heidelberg, 2005.
- D. J. Hand and V. Vinciotti. Choosing k for two-class nearest neighbour classifiers with unbalanced classes. *Pattern Recognition Letters*, 24(9):1555–1562, 2003.
- H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.
- J. Huang and C. X. Ling. Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 17(3):299–310, 2005.
- N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449, 2002.
- J. W. Jaromczyk and G. T. Toussaint. Relative neighborhood graphs and their relatives. *Proceedings of the IEEE*, 80(9):1502–1517, 1992.
- T. Joachims. Making large-scale SVM learning practical. In B. Schölkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 169–184. MIT Press, Cambridge, MA, 1999.
- P. Juszczak, D. M. Tax, and R. Duin. Feature scaling in support vector data description. In *Proceedings of 8th Annual Conference of the Advanced School for Computing and Imaging*, pages 95–102, Delft, Netherlands, 2002.
- S. Kotsiantis, D. Kanellopoulos, P. Pintelas, et al. Handling imbalanced datasets: A review. *CESTIS International Transactions on Computer Science and Engineering*, 30(1):25–36, 2006.
- M. Kuhn and K. Johnson. *Applied Predictive Modeling*. Springer, New York, USA, 2013.

- C. X. Ling, Q. Yang, J. Wang, and S. Zhang. Decision trees with minimal costs. In *Proceedings of the 21th International Conference on Machine Learning*, page 69, Banff, Alberta, Canada, 2004.
- X.-Y. Liu, J. Wu, and Z.-H. Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(2):539–550, 2009.
- R. Longadge and S. Dongre. Class imbalance problem in data mining: Review. *International Journal of Computer Science and Network*, 2(1):83–87, 2013.
- V. López, A. Fernández, S. García, V. Palade, and F. Herrera. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 250:113–141, 2013.
- I. Mami and I. Zhang. kNN approach to unbalanced data distributions: A case study involving information extraction. In *Proceedings of ICMML'2003 Workshop on Learning from Imbalanced Datasets II*, Washington, DC, USA, 2003.
- D. J. Marchette. Class cover catch digraphs. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(2):171–177, 2010.
- D. J. Marchette. *cccd: Class Cover Catch Digraphs*, 2013. URL <http://CRAN.R-project.org/package=cccd>. R package version 1.04.
- M. A. Mazurowski, P. A. Habas, J. M. Zurada, J. Y. Lo, J. A. Baker, and G. D. Tourassis. Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural Networks*, 21(2):427–436, 2008.
- D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, and F. Leisch. *e1071: Misc Functions of the Department of Statistics (e1071)*, TU Wien, 2014. URL <http://CRAN.R-project.org/package=e1071>. R package version 1.6-4.
- C. Phua, D. Alahakoon, and V. Lee. Minority report in fraud detection: Classification of skewed data. *ACM SIGKDD Explorations Newsletter*, 6(1):50–59, 2004.
- R. C. Prati, G. E. Batista, and M. C. Monard. Class imbalances versus class overlapping: An analysis of a learning system behavior. In *Proceedings of 3rd Mexican International Conference on Artificial Intelligence*, pages 312–321, Mexico City, Mexico, 2004.
- C. E. Priebe, J. G. DeVinney, and D. J. Marchette. On the distribution of the domination number for random class cover catch digraphs. *Statistics & Probability Letters*, 55(3):259–246, 2001.
- C. E. Priebe, D. J. Marchette, J. G. DeVinney, and D. A. Socolinsky. Classification using class cover catch digraphs. *Journal of Classification*, 20(1):3–23, 2003a.
- C. E. Priebe, J. L. Solka, D. J. Marchette, and B. T. Clark. Class cover catch digraphs for latent class discovery in gene expression monitoring by DNA microarrays. *Computational Statistics & Data Analysis*, 43(4):621–632, 2003b.
- R. Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015. URL <http://www.R-project.org/>.
- B. Raskutti and A. Kowalczyk. Extreme re-balancing for SVMs: A case study. *ACM SIGKDD Explorations Newsletter*, 6(1):60–69, 2004.
- L. Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1):1–39, 2010.
- B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.
- D. A. Socolinsky, J. D. Neuhäusel, C. E. Priebe, J. G. DeVinney, and D. J. Marchette. Fast face detection with a boosted CCCD classifier. In *Proceedings of the 35th Symposium on the Interface, Volume 34: Computing Science and Statistics*, Salt Lake City, Utah, USA, 2003.
- I. Takigawa, M. Kudo, and A. Nakamura. Convex sets as prototypes for classifying patterns. *Engineering Applications of Artificial Intelligence*, 22(1):101–108, 2009.
- Y. Tang, Y.-Q. Zhang, N. V. Chawla, and S. Krasser. SVMs modeling for highly imbalanced classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(1):281–288, 2009.
- D. M. Tax. Ddttools, the data description toolbox for MATLAB, July 2014. version 2.1.1.
- D. M. Tax and R. P. Duin. Support vector data description. *Machine Learning*, 54(1):45–66, 2004.
- N. Thai-Nghe, A. Busche, and L. Schmidt-Thieme. Improving academic performance prediction by dealing with class imbalance. In *Proceedings of 19th International Conference on Intelligent Systems Design and Applications*, pages 878–883, Pisa, Italy, 2009.
- D. B. West. *Introduction to Graph Theory*. Prentice Hall, New Jersey, USA, 2 edition, 2000.
- D. L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-2(3):408–421, July 1972.
- X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip, et al. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2008.
- H. Xiong, J. Wu, and L. Liu. Classification with class overlapping: A systematic study. In *Proceedings of the 1st International Conference on e-Business Intelligence*, Shanghai, China, 2010.
- B. Zadrozny, J. Langford, and N. Abe. Cost-sensitive learning by cost-proportionate example weighting. In *Proceedings of 3rd IEEE International Conference on Data Mining*, pages 435–442, Melbourne, Florida, USA, 2003.

A Variational Approach to Path Estimation and Parameter Inference of Hidden Diffusion Processes

Tobias Sutter

*Department of Electrical Engineering and Information Technology
ETH Zurich, Switzerland*

SUTTER@CONTROL.EE.ETHZ.CH

Arnab Ganguly

*Department of Mathematics
Louisiana State University, USA*

AGANGULY@LSU.EDU

Heinz Koeppl

*Department of Electrical Engineering and Information Technology
TU Darmstadt, Germany*

HEINZ.KOEPL@BCS.TU-DARMSTADT.DE

Editor: Manfred Opper

Abstract

We consider a hidden Markov model, where the signal process, given by a diffusion, is only indirectly observed through some noisy measurements. The article develops a variational method for approximating the hidden states of the signal process given the full set of observations. This, in particular, leads to systematic approximations of the smoothing densities of the signal process. The paper then demonstrates how an efficient inference scheme, based on this variational approach to the approximation of the hidden states, can be designed to estimate the unknown parameters of stochastic differential equations. Two examples at the end illustrate the efficacy and the accuracy of the presented method.

Keywords: Variational inference, stochastic differential equations, diffusion processes, hidden Markov model, optimal control

1. Introduction

Diffusion processes modeled by stochastic differential equations (SDEs) appear in several disciplines varying from mathematical finance to systems biology. For example, in systems biology stochastic differential equations are used for efficient modeling of the states of the chemical species in a reaction system when they are present in high abundance Wilkinson (2006). Oftentimes, the state of the system or the signal process is not directly observed, and inference of the state trajectories and parameter of the system has to be achieved based on noisy partial observations. Typically, in such a scenario, the observation data is conveniently modeled as a function of the hidden state corrupted with independent additive noise. However, generalizations of this basic setup, which, for example, could include stronger coupling between the hidden signal and the observation processes, are often used for modeling more complex phenomena.

In such a model optimal filtering theory concerns itself with recurrent estimation of the current state of the hidden signal process given the observation data until the present time. This is particularly useful in tracking problems where the estimation of the current location

of an object needs to be constantly updated as new noisy information flows in. On the other hand, optimal smoothing involves the class of methods which can be used in reconstruction of any past state of the signal process given a set of measurements up to the present time. More specifically, given the signal process X and the observation process Y , filtering theory entails computation of the conditional expectations of the form $\mathbb{E}[\phi(X_t)|\mathcal{F}_t^Y]$, where $\{\mathcal{F}_t^Y\}$ denotes the filtration generated by the process Y . The σ -algebra \mathcal{F}_t^Y contains all the information about the observation process Y up to the present time t . Smoothing, however, involves evaluation of the conditional expectations of the form $\mathbb{E}[\phi(X_s)|\mathcal{F}_t^Y]$, where $s < t$. The smoothing techniques can also be viewed as tools of estimation of the current state given a data set which includes future observations. This interpretation is particularly relevant in statistics, where such techniques are essentially the means of computing certain posterior conditional densities given the observation set. The present article focuses on a variational approach to this smoothing problem and later employs the method for estimation of parameters of diffusion processes.

Evaluation of such conditional expectations or densities are quite difficult, since they are often solutions of suitable (stochastic) partial differential equations. These are usually infinite-dimensional problems and analytical solutions are generally impossible. Hence, effort has been directed toward developing of a variety of numerical schemes for efficient approximation of these conditional densities. While Markov chain Monte Carlo methods for inference use discretization of the given SDE for writing down an approximate likelihood Kushner and Dupuis (2001); Pagès and Pham (2005); Andrieu et al. (2010), particle methods approximate the (posterior) conditional densities by suitably weighted point masses Crisan and Lyons (1999); Del Moral et al. (2001); Bain and Crisan (2009). However, these methods often rely on a suitable discretization of the problem which is mostly done in an ad-hoc way. Since a theoretical framework for obtaining approximations is not present, the approximation error might be difficult to quantify.

In contrast, the present paper focuses on a variational approach to this estimation problem. The main idea in such a method is to approximate the (posterior) conditional probability distribution of the system's state (given the observed data) by an appropriate Gaussian distribution, where the optimal parameters for the Gaussian distribution are obtained by minimizing the relative entropy (or Kullback-Leibler distance) between the posterior process and a suitable approximating SDE. Earlier works like Archambeau et al. (2007, 2008); Archambeau and Opper (2011); Cseke et al. (2013); Vrettas et al. (2015) considered the case when the signal process is modeled by an SDE with a constant diffusion term. The advantage of working with a constant diffusion term is that it implies that the approximating SDE will simply have a linear drift so that marginals are distributed as Gaussian. This simple expression of the SDE with a linear drift makes the subsequent optimization problem for finding the suitable parameters for this approximating SDE easier. However, since most physical phenomena cannot be realistically modeled by SDEs with constant diffusion term, there is a pressing need of extending the approach to general SDEs. One natural but naive approach in this regard could be to freeze the diffusion term at an appropriate value, that is, to take the zeroth order expansion of the diffusion coefficient. Although simple to implement, the efficacy of the method is not guaranteed by theoretical results and will vary from case to case, and a reasonable error analysis might require unreasonably restrictive conditions on the model.

Instead, the present article delves much deeper in to the problem and develops methods for finding the optimal approximating SDE such that the relative entropy between it and the true posterior process is minimized subject to the condition that the marginals of the former follow Gaussian distributions. The main obstacle that needs to be overcome in this approach stems from the fact that unlike the previous case, the approximating SDE here cannot be taken to be the one with a linear drift, and a suitable expression of it needs to be found so that the marginals are still Gaussian. This has been achieved in Theorem 9. In fact, our work outlines the most general techniques for approximating the posterior density by any density from the exponential family or mixture of exponential families. In this connection we would like to note that the reason for requiring that the marginals follow a Gaussian distribution or more generally, a distribution from the exponential family because this results in a finite-dimensional smoother which can be used for approximating a wide range of distributions.

It should be noted that the variational method considered here is different from the so-called extended Kalman filter (EKF) in two ways: first, EKF is employed for filtering problems; but more importantly, EKF starts by linearizing the signal (prior) SDE and then freezing its diffusion term, while the variational approach is concerned with approximation of the posterior SDE. Therefore even though in the constant diffusion term case, the approximating SDE happens to have linear drift and thus resulting in a Gaussian smoother, it is not based on the same philosophy behind the EKF. And as mentioned before, in the non-constant diffusion term case although our method can be used to obtain a finite-dimensional smoother, in particular, a Gaussian smoother, it completely avoids any form of linearization of the given SDE or subsequent freezing of the diffusion term.

In our paper this variational approximation method has been formulated as an optimal control problem. The advantage of this theoretical framework is that necessary conditions for global optimality are then obtained by employing the Pontryagin maximum principle. This leads to considerable computational advantages of the variational method compared to numerically solving the underlying (stochastic) PDEs, that is highlighted by two examples.

The later part of the paper focusses on the important topic of parameter inference of SDEs. The above scheme of estimating the hidden states and the smoothing densities is cleverly used in designing an efficient method for estimating parameters of SDEs. In particular, the paper proposes an iterative EM-type algorithm which aims to compute approximate maximum likelihood estimates of the parameters in a tractable way. Two illustrative examples, which are important in mathematical finance, demonstrate the accuracy and efficiency of the proposed algorithms. Future projects will address more complicated models.

The layout of this article is as follows: In Section 2 we formally introduce the problem setting. We consider as a running example throughout the manuscript a geometric Brownian motion. The variational approximation idea is motivated in Section 3 leading to a specific class of optimization problems that is addressed in Section 4. It is then reformulated in Section 5 as an optimal control problem and necessary conditions for optimality are derived. Section 6 explains how the variational approximation can be used to infer unknown parameters of the model. Section 7 discusses the presented variational approximation in the context of a discrete time measurement model. The theoretical results are applied in Section 8 to two examples: a geometric Brownian motion and to the Cox-Ingersoll-Ross

process. We finally conclude with some remarks and directions for future work in Section 9. Certain technical proofs are relegated to the appendix.

Notation. Hereafter, I_n is the n -dimensional identity matrix and E_t is the $n \times n$ matrix where the i -th entry is one and zero elsewhere. We let $\text{Sym}(n, \mathbb{R})$ and $\text{GL}(n, \mathbb{R})$ be respectively the set of symmetric and invertible $n \times n$ matrices with real entries. For matrices $A, B \in \mathbb{R}^{n \times n}$ let $\langle A, B \rangle := \text{tr}(A^T B)$ denote the Frobenius inner product. For a vector $b \in \mathbb{R}^n$ and a positive definite matrix A , we employ the norm $\|b\|_A := \sqrt{b^T A^{-1} b}$. We define the standard n -simplex as $\Delta_n := \{x \in \mathbb{R}^n : x \geq 0, \sum_{i=1}^n x_i = 1\}$. Let $\mathcal{C} := \mathcal{C}([0, T], \mathbb{R}^n)$ denote the space of continuous functions on $[0, T]$ taking values in \mathbb{R}^n . Let S be a metric space, equipped with its Borel σ -field $\mathcal{B}(S)$. The space of all probability measures on $(S, \mathcal{B}(S))$ is denoted by $\mathcal{P}(S)$. The relative entropy (or Kullback-Leibler divergence) between any two probability measures $\mu, \nu \in \mathcal{P}(S)$ is defined as

$$D(\mu \parallel \nu) := \begin{cases} \int \log \left(\frac{d\mu}{d\nu} \right) d\mu, & \text{if } \mu \ll \nu \\ +\infty, & \text{otherwise,} \end{cases}$$

where \ll denotes absolute continuity of measures and $\frac{d\mu}{d\nu}$ is the Radon-Nikodym derivative. By convention *measurable* means *Borel-measurable* in the sequel. Given an S -valued random variable X with Law $(X) = \mu \in \mathcal{P}(S)$, let $\mathbb{E}_\mu[X]$ denote the expectation of X .

2. Model setup

As usual, we will work on a complete probability space $(\Omega, \mathcal{F}, \mathbb{P})$ equipped with a filtration $\{\mathcal{F}_t\}$ satisfying the usual conditions, that is, $\{\mathcal{F}_t\}$ is complete, right continuous and contains all the \mathbb{P} -null sets. The basic objects in our study consist of a signal process X and an observation process Y , both of which are assumed to be $\{\mathcal{F}_t\}$ -adapted. The unobserved signal process X is modeled by the following stochastic differential equation describing the state evolution of a dynamical system:

$$dX_t = f(X_t)dt + \sigma(X_t)dW_t, \quad X_0 = x_0, \quad 0 \leq t \leq T, \tag{1}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, $\sigma : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$, and W is an n -dimensional Brownian motion independent of x_0 . The observation process Y is modeled as noisy measurements of some function of the signal process X . Mathematically, Y is defined as

$$Y_t = \int_0^t h(X_s)ds + B_t, \tag{2}$$

where $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is called the observation function and B is an m -dimensional Brownian motion independent of x_0 and W .

Assumption 1 We stipulate that

- (i) f and σ are globally Lipschitz;
- (ii) and h is twice continuously differentiable.

It is known Kallenberg (2002) that under Assumption 1 there exists a unique strong solution to the SDE (1). Given the observed data up to some time T , $\{Y_s : s \leq T\}$, the goal of the paper is to outline an approximation method for the smoothing density, $\mathcal{P}_S(x, t)$, which is the conditional probability density of X_t given $\{Y_s : s \leq T\}$. In other words, the smoothing density is defined by the equation:

$$\mathbb{E}[\phi(X_t) | \mathcal{F}_T^X] = \int \phi(x) \mathcal{P}_S(x, t) dx, \quad (3)$$

up to a.s. equivalence, where ϕ is any bounded measurable function from \mathbb{R}^n to \mathbb{R} and $\{\mathcal{F}_t^Y\}$ denotes the filtration generated by the process Y .

More generally, we will be interested in approximating the full conditional probability measure on the path space, $C \equiv C([0, T], \mathbb{R}^n)$. To describe this mathematically, assume that a regular conditional probability measure $\mathbb{P}[\cdot | \mathcal{F}_T^X]$ is chosen. Then there exists a measurable probability kernel $y \in C \rightarrow \Pi_{\text{post}}(\cdot, y) \in \mathcal{P}(C)$ such that for any measurable set $A \subset C$,

$$\mathbb{P}[X_{[0, T]} \in A | \mathcal{F}_T^X] = \Pi_{\text{post}}(A, Y_{[0, T]}).$$

Given the observation process up to time T , $Y_{[0, T]}$, we now describe a characterization of the probability measure $\Pi_{\text{post}}(\cdot, Y_{[0, T]})$, which will play a pivotal role for our purposes. The probability measure $\Pi_{\text{post}}(\cdot, Y_{[0, T]})$ is actually the distribution of a diffusion process \bar{X}^T on C , and the latter is obtained by a modification of the original signal process X :

$$d\bar{X}_t^T = g(\bar{X}_t^T, t)dt + \sigma(\bar{X}_t^T)d\bar{W}_t, \quad \bar{X}_0^T = x_0, \quad (4)$$

where \bar{W} is an $\{\mathcal{F}_t\}$ -adapted Brownian motion that is independent of Y . Notice that the diffusion coefficient of the above SDE (which we will henceforth call the posterior SDE or posterior diffusion) is same as that of the original SDE, and the drift of this posterior SDE is time-dependent and is obtained as

$$g(x, t) := f(x) + a(x)\nabla \log w(x, t), \quad (5)$$

where $a(x) := \sigma(x)\sigma(x)^\top$. We give details about the (random) function w a little later, but the important point to note here is that the new drift function is the old drift function with an extra additive term, and the observation process $Y_{[0, T]}$ enters into the characterization of $\Pi_{\text{post}}(\cdot, Y_{[0, T]})$ only through w .

To see this characterization of $\Pi_{\text{post}}(\cdot, Y_{[0, T]})$, we first look at the usual filtering density $\mathcal{P}_F(x, t)$, which is naturally defined by

$$\mathbb{E}[\phi(X_t) | \mathcal{F}_t^Y] = \int \phi(x) \mathcal{P}_F(x, t) dx. \quad (6)$$

Under suitable technical conditions, the filter density \mathcal{P}_F satisfies the Kushner-Stratonovich equation (for example, see Stratonovich (1960); Kushner (1967); Bain and Crisan (2009)). For our purposes, however, it is convenient to work with the unnormalized filter density $p(x, t)$, that is, $\mathcal{P}_F(x, t) = p(x, t) (\int_{\mathbb{R}^n} p(x, t) dx)^{-1}$, which satisfies the so-called Zakai equation Zakai (1969)

$$\begin{cases} dp(x, t) = \mathcal{A}^* p(x, t) dt + p(x, t) h(x)^\top dY_t \\ p(x, 0) = p_0(x). \end{cases} \quad (7)$$

Here p_0 denotes the density of x_0 and \mathcal{A}^* is the adjoint of the infinitesimal generator of the process X given by $\mathcal{A}\psi(x) = \sum_i f_i(x) \frac{\partial \psi}{\partial x_i} + \frac{1}{2} \sum_{i,j} a_{i,j}(x) \frac{\partial^2 \psi}{\partial x_i \partial x_j}$ for $\psi \in \mathcal{C}_0^2(\mathbb{R}^n, \mathbb{R})$. We next consider the backward stochastic partial differential equation (SPDE)

$$\begin{cases} dw(x, t) = -\mathcal{A}w(x, t)dt - w(x, t)h(x)^\top dY_t \\ w(x, T) = 1. \end{cases} \quad (8)$$

Conditions about existence of solutions to (7) and (8) can be found in Pardoux (1981/82). It is well known (Pardoux, 1981/82, Corollary 3.8) that the smoothing density can be expressed as

$$\mathcal{P}_S(x, t) = \frac{p(x, t)w(x, t)}{\int_{\mathbb{R}^n} p(x, t)w(x, t)dx}. \quad (9)$$

Now by using (7), (8) and (9), it can be shown¹ that the smoothing density solves the following Kolmogorov forward equation

$$\left(\frac{\partial}{\partial t} + \sum_i \frac{\partial}{\partial x_i} g(x, t) - \frac{1}{2} \sum_{i,j} \frac{\partial^2}{\partial x_i \partial x_j} a_{i,j}(x) \right) \mathcal{P}_S(x, t) = 0, \quad (10)$$

with the drift term g defined by (5). In other words, the conditional probability measure $\Pi_{\text{post}}(\cdot, Y_{[0, T]})$ on C is induced by the diffusion process \bar{X}^T as defined in (4).

Evaluating $\Pi_{\text{post}}(\cdot, Y_{[0, T]})$ is what is known as the path estimation problem. Except for a few simple cases, the SPDEs, that are involved in this estimation of the hidden path, are analytically intractable. The variational approach that we undertake in this paper actually has the goal of approximating $\Pi_{\text{post}}(\cdot, Y_{[0, T]})$. Toward this end, a natural objective is to approximate $\Pi_{\text{post}}(\cdot, Y_{[0, T]})$ by a probability measure such that the corresponding marginals of the latter come from a known family of distributions (e.g. exponential family). As a result, the marginal of this approximating probability measure at time t approximates the smoothing density $\mathcal{P}_S(x, t)$. The procedure adopted in this article involves finding the optimal parameters of this approximating distribution by minimizing the relative entropy between the posterior distribution and the approximating one.

2.1 Example: Geometric Brownian Motion

We present as a running example throughout the article the geometric Brownian motion that is used to model stock prices in the Black-Scholes model, see Shiryaev (1999). The system dynamics (1) is given by a one-dimensional geometric Brownian motion

$$dX_t = \kappa X_t dt + \lambda X_t dW_t, \quad X_0 = x_0 \sim \log \mathcal{N}(\mu, \sigma), \quad (11)$$

for $0 \leq t \leq T$, $\lambda, \kappa > 0$ and an observation process (2) defined by

$$Y_t = \int_0^t X_s ds + B_t. \quad (12)$$

It is straightforward to see that Assumption 1 holds in this setting.

1. See Appendix A for a detailed derivation.

3. Variational approximation: Motivation

Let Π_{prior} denote the distribution of the original signal process X on \mathcal{C} , that is, for a measurable $\mathcal{A} \subset \mathcal{C}$, $\Pi_{\text{prior}}(\mathcal{A}) = \mathbb{P}[X_{[0,T]} \in \mathcal{A}]$. Define the two terms

$$H_T(X_{[0,T]}, y) := -h(X_T)_{yT} + \int_0^T y_s dh(X_s) + \frac{1}{2} \int_0^T \|h(X_s)\|^2 ds \quad (13)$$

$$I(H_T(\cdot, y)) := -\log \left(\int \exp(-H_T(\cdot, y)) d\Pi_{\text{prior}} \right). \quad (14)$$

Let y be a sample path of the observation process Y on the interval $[0, T]$. Then notice that by the pathwise Kallianpur-Striebel formula (or the Bayes formula), we have

$$\frac{d\Pi_{\text{post}}(\cdot, y)}{d\Pi_{\text{prior}}} = \frac{\exp(-H_T(\cdot, y))}{\int \exp(-H_T(\cdot, y)) d\Pi_{\text{prior}}} = \frac{\exp(-H_T(\cdot, y))}{L(y)},$$

where $L(y) = \int \exp(-H_T(\cdot, y)) d\Pi_{\text{prior}}$. Consequently, $L(y)$ can be interpreted naturally as the likelihood of the path y , or equivalently, $I(H_T(\cdot, y))$ is viewed as the negative log-likelihood of the sample path y . The term $H_T(X_{[0,T]}, y)$ can be interpreted as the X -conditional information and the information in the observation that $Y = y$, see Mitter and Newton (2003) for more details. Now for any probability measure Q^2 on $C([0, T], \mathbb{R})$, the relative entropy between Q and $\Pi_{\text{post}}(\cdot, y)$ can be expressed by the following lemma.

Lemma 2 $D(Q||\Pi_{\text{post}}(\cdot, y)) = -I(H_T(\cdot, y)) + D(Q||\Pi_{\text{prior}}) + \mathbb{E}_Q[H_T(\cdot, y)]$.

Proof. The proof essentially follows the one in (van Handel, 2007, Lemma 2.2.1). Splitting the relative entropy and using the pathwise Kallianpur-Striebel formula yields

$$\begin{aligned} D(Q||\Pi_{\text{prior}}) &= \int \left[\log \left(\frac{dQ}{d\Pi_{\text{post}}(\cdot, y)} \right) + \log \left(\frac{d\Pi_{\text{post}}(\cdot, y)}{d\Pi_{\text{prior}}} \right) \right] dQ \\ &= D(Q||\Pi_{\text{post}}(\cdot, y)) + \int \log \left(\frac{d\Pi_{\text{post}}(\cdot, y)}{d\Pi_{\text{prior}}} \right) dQ \\ &= D(Q||\Pi_{\text{post}}(\cdot, y)) + \int \log \left(\frac{\exp(-H_T(\cdot, y))}{\int \exp(-H_T(\cdot, y)) d\Pi_{\text{prior}}} \right) dQ \\ &= D(Q||\Pi_{\text{post}}(\cdot, y)) - \mathbb{E}_Q[H_T(\cdot, y)] - \log \left(\int \exp(-H_T(\cdot, y)) d\Pi_{\text{prior}} \right). \quad \blacksquare \end{aligned}$$

Mitter and Newton Mitter and Newton (2003) provide an information-theoretic interpretation to this result. They interpret the term (14) as the *total information* available to the estimator Q through the sample path y . On the other hand, they call the quantity $\mathcal{F}(Q, y) := D(Q||\Pi_{\text{prior}}) + \mathbb{E}_Q[H_T(\cdot, y)]$ the *apparent information* of the estimator. By non-negativity of the relative entropy $\mathcal{F}(Q, y) \geq I(H_T(\cdot, y))$ with equality if and only if $Q = \Pi_{\text{post}}(\cdot, y)$. In this sense, a suboptimal estimator appears to have access to more information than is actually available.

² Q will be called the approximating probability measure in the sequel.

Since the total information $I(H_T(\cdot, y))$ does not depend on Q , minimizing the relative entropy between Q and $\Pi_{\text{post}}(\cdot, y)$ over a class of probability measures \mathcal{Q} is equivalent to minimizing the apparent information $\mathcal{F}(Q, y)$. This motivates to consider an approximating distribution Q on \mathcal{C} that is characterized as the solution to the following optimization problem:

Problem 3 Minimize $D(Q||\Pi_{\text{prior}}) + \mathbb{E}_Q[H_T(\cdot, y)]$ subject to

(i) Q is a probability distribution induced by an SDE of the form

$$dZ_t = u(Z_t, t)dt + \sigma(Z_t)dW_t, \quad Z_0 = x_0, \quad 0 \leq t \leq T; \quad (15)$$

(ii) The marginals of Q at time t , i.e., the distribution of Z_t , belong to a chosen family of distributions.

We will show in the remainder of this article how Problem 3 can be restated as an optimal control problem, which leads to a standard formulation of necessary optimality conditions in terms of Pontryagin's maximum principle.

Note that the objective function of Problem 3 is known to be strictly convex with respect to Q , see Giszszár (1975). The constraint (ii) restricts the feasible set approximating distributions Q to a nonconvex set. Note that such problems (i.e., absence of constraint (i)) have been studied in the literature Pinski et al. (2015)). In our setting, the set of feasible solutions is also coupled with the first constraint (i), that parametrizes the feasible set of distributions in terms of the drift function u . This coupling is investigated in Section 4, in particular Theorem 9 characterizes the set of all drift terms u such that the distribution induced by (15) has finite dimensional marginals that belong to a given family of distributions. Hence, Problem 3 can alternatively be interpreted as minimizing the objective function over a class of drift functions u that induce Q via (15) and such that Q satisfies constraint (ii). For example, if the goal is to approximate the posterior distribution Π_{post} by a distribution Q whose marginals are normal distributions, then one aims to find a drift term u such that the objective function is minimized and such that the solution Z_t to (15) admits a normal distribution.

Remark 4 Notice that the unconstrained optimization of the objective function in Problem 3 with respect to Q will simply yield the minimizer Q to be Π_{post} . Since, as discussed in the beginning of Section 2, Π_{post} is induced by the SDE, (4), the constraint (i) in Problem 3 is essentially inbuilt. In other words, it is the constraint (ii) which plays the crucial role in the methods outlined in this paper.

The objective function in Problem 3, in particular the relative entropy between the approximating distribution Q and the prior distribution Π_{prior} can be simplified, since due to the constraint (i) the underlying SDEs (15) and (1) share the same diffusion coefficient. In view of (15) and (1), consider two SDEs for $0 \leq t \leq T$

$$dX_t = f(X_t)dt + \sigma(X_t)dW_t, \quad dZ_t = u(Z_t, t)dt + \sigma(Z_t)dW_t, \quad X_0 = Z_0 = x_0,$$

with $u: \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$, $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$, $\sigma: \mathbb{R}^n \rightarrow \mathbb{R}^{n \times n}$, W an n -dimensional Brownian motion independent of x_0 and both SDEs satisfying Assumption 1. Let $(\Omega, \mathcal{F}_T, P)$ be a probability

space, where \mathcal{F}_T is the sigma algebra $\sigma(W_s : s \leq T)$ and let Π_{prior} and Q denote the the laws of X_t and Z_t with respect to P . It follows by Girsanov's Theorem Øksendal (2003), that

$$\mathbb{E}_Q \left[\log \left(\frac{dQ}{d\Pi_{\text{prior}}} \right) \right] = \frac{1}{2} \mathbb{E}_Q \left[\int_0^T \varphi(s, \omega)^\top \varphi(s, \omega) ds \right],$$

where $\varphi(s, \omega) := \sigma(Z_s(\omega))^{-1} (u(X_s(\omega)) - f(X_s(\omega)))$. Therefore, the relative entropy between Q and Π_{prior} is

$$D(Q \| \Pi_{\text{prior}}) = \frac{1}{2} \mathbb{E}_Q \left[\int_0^T \|u(X_s, s) - f(X_s)\|_{\sigma(X_s)}^2 ds \right],$$

where $\|u(x, s) - f(x)\|_{\sigma(x)}^2 := (u(x, s) - f(x))^\top a(x)^{-1} (u(x, s) - f(x))$. Hence, the objective function in Problem 3 can be expressed as

$$\begin{aligned} D(Q \| \Pi_{\text{prior}}) + \mathbb{E}_Q[H_T(\cdot, y)] \\ = \int_0^T \mathbb{E}_Q \left[\frac{1}{2} \|u(X_t, t) - f(X_t)\|_{\sigma(X_t)}^2 + y_t \left(u(X_t, t)^\top \nabla h(X_t) \right. \right. \\ \left. \left. + \frac{1}{2} \sigma(X_t)^\top \nabla^2 h(X_t) \sigma(X_t) \right) + \frac{1}{2} \|h(X_t)\|^2 \right] dt - y_T \mathbb{E}_Q[h(X_T)], \end{aligned} \quad (16)$$

where the last equality is due to Fubini's Theorem and Itô's Lemma. The two coupling constraints (i) and (ii) in Problem 3 are studied in the next section and will finally allow us to reformulated Problem 3 as an optimal control problem.

4. Multi-dimensional SDE with prescribed marginal law

This section establishes conditions on the drift function in the approximate SDE (15) such that the induced marginal distributions evolve in a given exponential family.

Definition 5 (Exponential family) Let $\mathcal{H}_1, \dots, \mathcal{H}_m$ be Hilbert spaces and let $\mathcal{H} = \prod_{i=1}^m \mathcal{H}_i$ be endowed with the inner product $\langle \cdot, \cdot \rangle$. Let the functions $c_i : \mathbb{R}^n \rightarrow \mathcal{H}_i$ for $i = 1, \dots, m$ be linearly independent, have at most polynomial growth, be twice continuously differentiable and denote $c(x) = (c_1(x), \dots, c_m(x))$. Assume that the convex set

$$\Gamma := \{\Theta \in \mathcal{H} : \psi(\Theta) = \log \int \exp(\langle \Theta, c(x) \rangle) dx < \infty\}$$
 has non-empty interior. Then

$$EM(c) = \{p(\Theta), \Theta \in \Lambda\}, \quad p(x, \Theta) := \exp(\langle \Theta, c(x) \rangle) - \psi(\Theta),$$

where $\Lambda \subseteq \Gamma$ is open, is called an exponential family of probability densities.

Definition 6 (Mixture of exponential families) Let $EM(c^{(i)})$ for $i = 1, \dots, k$ be exponential families according to Definition 5. Then

$$EM(c^{(1)}, \dots, c^{(k)}) = \left\{ \sum_{\ell=1}^k \nu_\ell p_\ell(\cdot, \Theta^{(\ell)}) : p_\ell(\cdot, \Theta^{(\ell)}) \in EM(c^{(\ell)}), \nu \in \Delta_k \right\}$$

is called a mixture of k exponential families of probability densities.

Consider the stochastic differential equation (15), where $u : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$, $\sigma : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times d}$ and W is a d -dimensional Brownian motion independent of x_0 .

Assumption 7

1. The SDE (15) satisfies Assumption 1.
2. The initial condition x_0 has a density p_0 that is absolutely continuous with respect to the Lebesgue measure and has finite moments of any order.
3. The unique solution X_t to (15) admits a density $p(x, t)$ that is absolutely continuous with respect to the Lebesgue measure and that satisfies the Kolmogorov forward equation.

Problem 8 Let $EM(c^{(1)}, \dots, c^{(k)})$ be a mixture of exponential families, let p_0 be a density contained in $EM(c^{(1)}, \dots, c^{(k)})$, let σ be a diffusion term and let $a(\cdot) := \sigma(\cdot)\sigma(\cdot)^\top$. Let $\mathcal{U}(x_0, \sigma)$ denote the set of all drifts u such that x_0, u, σ and its related SDE (15) satisfy Assumption 7. Assume $\mathcal{U}(x_0, \sigma)$ to be non-empty. Then given a curve $t \mapsto p(\cdot, \Theta_t^{(1)}, \dots, \Theta_t^{(k)})$ in $EM(c^{(1)}, \dots, c^{(k)})$, find a drift in $\mathcal{U}(x_0, \sigma)$ whose related SDE has a solution with marginal density $p(\cdot, \Theta_t^{(1)}, \dots, \Theta_t^{(k)})$.

A solution to Problem 8 is given by the following theorem.

Theorem 9 Given the assumptions and notation of Problem 8. Consider the SDE (15) with drift term

$$u_t(x, t) = \frac{1}{2} \sum_{j=1}^n \frac{\partial}{\partial x_j} a_{ij}(x) + \frac{1}{2} \sum_{j=1}^n a_{ij}(x) \frac{\partial}{\partial x_j} p(x, \Theta_t^{(1)}, \dots, \Theta_t^{(k)}) - \frac{1}{p(x, \Theta_t^{(1)}, \dots, \Theta_t^{(k)})} \sum_{\ell=1}^k \nu_\ell p_\ell(x, \Theta_t^{(\ell)}) \left\langle \dot{\Theta}_t^{(\ell)}, \mathcal{I}_t^{(\ell)}(x) \right\rangle,$$

for $i = 1, \dots, n$, where

$$\mathcal{I}_i^{(\ell)}(x) := \int_{-\infty}^{x_i} \varphi_i^{(\ell)}((x_{-i}, \xi_i), \Theta_t^{(\ell)}) \exp \left(\left\langle \Theta_t^{(\ell)}, c^{(\ell)}(x_{-i}, \xi_i) - c^{(\ell)}(x) \right\rangle \right) d\xi_i, \quad (17)$$

$(x_{i-}, \xi_i) := (x_1, \dots, x_{i-1}, \xi_i, x_{i+1}, \dots, x_n)^\top$ and the functions $\varphi_i^{(\ell)} : \mathbb{R}^n \times \mathcal{H} \rightarrow \mathcal{H}$ for all $\ell = 1, \dots, k$ satisfy

$$\sum_{i=1}^n \left\langle \dot{\Theta}_t^{(\ell)}, \varphi_i^{(\ell)} \left((x_{-i}, \xi_i), \Theta_t^{(\ell)} \right) \right\rangle_{\xi_i=x_i} = \left\langle \dot{\Theta}_t^{(\ell)}, c^{(\ell)}(x) - \nabla \psi_\ell(\Theta_t^{(\ell)}) \right\rangle. \quad (18)$$

If $u \in \mathcal{U}(x_0, \sigma)$, then the SDE (15) solves Problem 8, i.e., X_t has a density

$$p_{X_t}(x) = \sum_{\ell=1}^k \nu_\ell \exp \left(\left\langle \Theta_t^{(\ell)}, c^{(\ell)}(x) \right\rangle - \psi_\ell(\Theta_t^{(\ell)}) \right), \quad \text{for all } t \leq T.$$

The proof is provided in Appendix B.

Remark 10 1. For the non-mixture and one-dimensional case ($k = n = 1$), the result is known Brigo (2000) and coincides with Theorem 9. Furthermore, it can be seen by the proof in Brigo (2000) and by invoking the existence and uniqueness theorem for ODEs, that the drift function u is uniquely determined.

2. For the multi-dimensional case ($n > 1$), the drift function is not unique anymore, as there exist multiple choices for $\varphi_i^{(\theta)}$. This gives rise to a natural question, if there exist a particular choice of $\varphi_i^{(\theta)}$ such that the integral terms $\mathcal{I}_i^{(\theta)}$ in (17) admit closed-form expressions. In Section 4.1 (Proposition 11), we derive such functions $\varphi_i^{(\theta)}$ for the mixture of multivariate normal densities.

3. In a non-mixture setting ($k = 1$), the drift function simplifies to

$$u_i(x; t) = \frac{1}{2} \sum_{j=1}^n \frac{\partial}{\partial x_j} a_{i,j}(x) + \frac{1}{2} \sum_{j=1}^n a_{i,j}(x) \left\langle \Theta_i, \frac{\partial c(x)}{\partial x_j} \right\rangle - \left\langle \Theta_i, \int_{-\infty}^{x_i} \varphi_i((x_{-i}, \xi), \Theta_i) \exp \left[\langle \Theta_i, c(x_{-i}, \xi) - c(x) \rangle \right] d\xi \right\rangle,$$

where the functions φ_i have to satisfy (18).

As remarked, the drift term proposed in Theorem 9 consists of the integral terms (17), that depend on the particular exponential families considered. In the following, we restrict ourselves to the mixture of multivariate normal densities and show that these integral terms, and hence the drift function, admit a closed-form expression.

4.1 Mixture of multivariate normal densities

Consider the family of multivariate Gaussian distributions with mean $m \in \mathbb{R}^n$ and covariance matrix $S \in \text{Sym}(n, \mathbb{R})$, that can be expressed in terms of Definition 5 as follows. Let the Hilbert space $\mathcal{H} = \mathbb{R}^n \times \mathbb{R}^{m \times n}$ be endowed with the inner product $\langle (a, A), (b, B) \rangle = a^\top b + \text{tr}(A^\top B)$ and define

$$\begin{aligned} \Theta &= (\eta, \theta) := \left(S^{-1}m, -\frac{1}{2}S^{-1} \right) \in \mathcal{H}, \quad c : \mathbb{R}^n \rightarrow \mathcal{H}, \quad c(x) = (x, xx^\top) \\ \psi : \mathcal{H} &\rightarrow \mathbb{R}, \quad \psi(\Theta) &= -\frac{1}{4} \text{tr}(\eta \eta^\top \theta^{-1}) + \frac{1}{2} \log \det \left(-\frac{1}{2} \theta^{-1} \right) + \frac{n}{2} \log(2\pi). \end{aligned} \quad (19)$$

A direct computation, using $\text{tr}(\eta \eta^\top \theta^{-1}) = \eta^\top \theta^{-1} \eta$, leads to

$$p(x, \Theta) = \exp \left(\langle c(x), \Theta \rangle - \psi(\Theta) \right) = \frac{1}{(2\pi)^{\frac{n}{2}} (\det S)^{\frac{1}{2}}} \exp \left(-\frac{1}{2} (x - m)^\top S^{-1} (x - m) \right).$$

3. For example, $\varphi_i^{(\theta)}(x, \Theta_i^{(\theta)}) := \delta_{\eta_j} c^{(\theta)}(x) - \nabla_{\Theta} \psi(\Theta_i^{(\theta)})$ for all $j \in \{1, \dots, n\}$ are feasible choices for $\varphi_i^{(\theta)}$, as they satisfy (18).

We point out again that for the proposed variational method, it is favourable if the approximating SDE (15) has a drift function that admits a closed-form expression. Furthermore, since the drift function is not unique (cf. Remark 10), among all feasible solutions characterized by the $\varphi_i^{(\theta)}$ functions, we want to find one that can be computed analytically. The latter turns out to be a difficult task and depending heavily on the specific exponential family chosen. From now on, we consider the exponential family of the multivariate normal probability densities that is given by (19). In this setting, it is possible to find functions $\varphi_i^{(\theta)}$ such that the integral terms (17), and therefore the drift function, can be computed in closed form.

Proposition 11 For the mixture of multivariate normal densities, one possible choice for the drift function proposed by Theorem 9 is

$$u(x; t) = \frac{1}{2} \text{div}(a(x)) + \frac{\sum_{\ell=1}^k \nu_\ell p_\ell(x, \Theta_\ell^{(\theta)})}{p(x, \Theta^{(1)}, \dots, \Theta_\ell^{(k)})} \left(\frac{1}{4} \theta_\ell^{(\theta)-1} \theta_\ell^{(\theta)} \theta_\ell^{(\theta)-1} \theta_\ell^{(\theta)} - \frac{1}{2} \theta_\ell^{(\theta)-1} \dot{\eta}_\ell^{(\theta)} - \frac{1}{2} \theta_\ell^{(\theta)-1} \dot{\theta}_\ell^{(\theta)} x + a(x) \left(\frac{1}{2} \eta_\ell + \theta_\ell x \right) \right).$$

The proof is provided in Appendix C.

Remark 12 For the non-mixture setting the drift term simplifies to

$$u(x; t) = \frac{1}{2} \text{div}(a(x)) + \frac{1}{4} \theta_r^{-1} \dot{\theta}_r \theta_r^{-1} \eta_r - \frac{1}{2} \theta_r^{-1} \dot{\eta}_r - \frac{1}{2} \theta_r^{-1} \dot{\theta}_r x + a(x) \left(\frac{1}{2} \eta_r + \theta_r x \right),$$

that in the special case of a constant diffusion term is a linear function, as one would expect.

We introduce the following ansatz for the drift function

$$u(x; t) = \frac{1}{2} \text{div}(a(x)) + \frac{\sum_{\ell=1}^k \nu_\ell p_\ell(x, \Theta_\ell^{(\theta)}) \left(A_\ell^{(\theta)} + B_\ell^{(\theta)} x + a(x) \left(C_\ell^{(\theta)} + D_\ell^{(\theta)} x \right) \right)}{p(x, \Theta^{(1)}, \dots, \Theta_\ell^{(k)})}, \quad (20)$$

where $B_\ell^{(\theta)}, D_\ell^{(\theta)} \in \mathbb{R}^{n \times n}$ and $A_\ell^{(\theta)}, C_\ell^{(\theta)} \in \mathbb{R}^n$ for all $\ell = 1, \dots, k$. The coefficients $A_\ell^{(\theta)}, B_\ell^{(\theta)}, C_\ell^{(\theta)}$ and $D_\ell^{(\theta)}$ cannot be chosen arbitrarily. They are coupled according to Proposition 11. By comparing the coefficients of Proposition 11 and (20) one gets

$$A_\ell^{(\theta)} = \frac{1}{4} \theta_\ell^{(\theta)-1} \dot{\theta}_\ell^{(\theta)} \theta_\ell^{(\theta)-1} \theta_\ell^{(\theta)} - \frac{1}{2} \theta_\ell^{(\theta)-1} \dot{\eta}_\ell^{(\theta)}, \quad B_\ell^{(\theta)} = -\frac{1}{2} \theta_\ell^{(\theta)-1} \dot{\theta}_\ell^{(\theta)}, \quad C_\ell^{(\theta)} = \frac{1}{2} \eta_\ell^{(\theta)}, \quad D_\ell^{(\theta)} = \theta_\ell^{(\theta)}.$$

Hence, one directly sees that the four parameters $A_\ell^{(\theta)}, B_\ell^{(\theta)}, C_\ell^{(\theta)}$ and $D_\ell^{(\theta)}$ for all $\ell = 1, \dots, k$ are coupled via the two ODEs

$$\frac{dC_\ell^{(\theta)}}{dt} = -D_\ell^{(\theta)} A_\ell^{(\theta)} - B_\ell^{(\theta)\top} C_\ell^{(\theta)}, \quad \frac{dD_\ell^{(\theta)}}{dt} = -2D_\ell^{(\theta)} B_\ell^{(\theta)}. \quad (21)$$

Note that the parametrization introduced in (20) provides relatively simple expressions for the mean and variance of the variational approximation derived in the next section (Section 4.2). In the authors' opinion this parametrization therefore helps to keep the notation simple.

4.2 Equations for mean and variance

Theorem 9 provides an explicit formula for the drift term in the approximating SDE (15), that simplifies to (20) in the case of multi-normal marginal densities. Therefore, the mean and variance of the approximating SDE (15) are characterized via the following two ODEs.

Theorem 13 *Consider the SDE (15) with drift term u given by (20), such that the solution X_t has a marginal density $p(x, \Theta_t^{(1)}, \dots, \Theta_t^{(k)}) \in EM(c_1, \dots, c_k)$ that is an arbitrary convex combination of densities $p_\ell(x, \Theta_t^{(\ell)}) \in EM(c_\ell)$ for $\ell = 1, \dots, k$. Let $m_t^{(\ell)}$ and $S_t^{(\ell)}$ denote the mean and variance of X_t with respect to $p_t(x, \Theta_t^{(\ell)})$. Then,*

$$-\frac{dm_t^{(\ell)}}{dt} = \frac{1}{2} \mathbb{E}_{p_t^{(\ell)}}[\operatorname{div}(a(X))] + A_t^{(\ell)} + B_t^{(\ell)} m_t^{(\ell)} + \mathbb{E}_{p_t^{(\ell)}}[a(X)] C_t^{(\ell)} + \mathbb{E}_{p_t^{(\ell)}}[a(X) D_t^{(\ell)} X] \quad (22)$$

and

$$\begin{aligned} \frac{dS_t^{(\ell)}}{dt} &= \frac{1}{2} \mathbb{E}_{p_t^{(\ell)}}[X \operatorname{div}(a(X))] + \frac{1}{2} \mathbb{E}_{p_t^{(\ell)}}[\operatorname{div}(a(X)) X^\top] - \frac{1}{2} m_t^{(\ell)} \mathbb{E}_{p_t^{(\ell)}}[\operatorname{div}(a(X))]^\top \\ &\quad - \frac{1}{2} \mathbb{E}_{p_t^{(\ell)}}[\operatorname{div}(a(X))] m_t^{(\ell)\top} + \mathbb{E}_{p_t^{(\ell)}}[a(X)] + S_t^{(\ell)} B_t^{(\ell)\top} + B_t^{(\ell)} S_t^{(\ell)} \\ &\quad + \mathbb{E}_{p_t^{(\ell)}}[X C_t^{(\ell)\top} a(X)] + \mathbb{E}_{p_t^{(\ell)}}[a(X) C_t^{(\ell)} X^\top] - m_t^{(\ell)} C_t^{(\ell)\top} \mathbb{E}_{p_t^{(\ell)}}[a(X)] \\ &\quad - \mathbb{E}_{p_t^{(\ell)}}[a(X)] C_t^{(\ell)} m_t^{(\ell)\top} + \mathbb{E}_{p_t^{(\ell)}}[X X^\top D_t^{(\ell)} a(X)] + \mathbb{E}_{p_t^{(\ell)}}[a(X) D_t^{(\ell)} X X^\top] \\ &\quad - m_t^{(\ell)} \mathbb{E}_{p_t^{(\ell)}}[X^\top D_t^{(\ell)} a(X)] - \mathbb{E}_{p_t^{(\ell)}}[a(X) D_t^{(\ell)} X] m_t^{(\ell)\top}. \end{aligned} \quad (23)$$

The proof is provided in Appendix D. Note that given $m_t^{(\ell)}$ and $S_t^{(\ell)}$ the mean and variance of X_t can be expressed as $m_t = \sum_{\ell=1}^k \nu_\ell m_t^{(\ell)}$ and $S_t = \sum_{\ell=1}^k \nu_\ell S_t^{(\ell)} + \sum_{\ell=1}^k \nu_\ell m_t^{(\ell)} m_t^{(\ell)\top} - (\sum_{\ell=1}^k \nu_\ell m_t^{(\ell)}) (\sum_{\ell=1}^k \nu_\ell m_t^{(\ell)})^\top$, respectively.

Remark 14 If the coefficients ν_ℓ in the convex combination of the marginal density $p(x, \Theta_t^{(1)}, \dots, \Theta_t^{(k)})$ in Theorem 13 are fixed a priori, the ODEs (22) and (23) are only sufficient for describing $m_t^{(\ell)}$ and $S_t^{(\ell)}$. Oftentimes, however, one is interested in choosing those coefficients a posteriori, for example by solving an auxiliary optimization problem. In such a setting the ODEs given by Theorem 13 are necessary and sufficient.

We have studied how to reformulate the constraints (i) and (ii) of Problem 3 by deriving an expression for the drift term to the approximating SDE (15). In the case that the marginals in (ii) are restricted to a mixture of multivariate normal densities this reformulation reduces to the ODEs (21), (22) and (23).

4.3 Example: Geometric Brownian Motion

We continue the geometric Brownian motion example started in Section 2.1. The goal is to approximate the smoothing density by a normal density. Therefore, according to Proposition 11, the drift function for the approximating SDE (15) has to be chosen as

$$u(x, t) = A_t + (\lambda^2 + B_t)x + \lambda^2 x^2 (C_t + D_t x), \quad (24)$$

where the coefficients A_t, B_t, C_t, D_t are coupled via the two ODEs (21). This choice of drift function leads to ODEs for the mean and the variance of the posterior process, according to Theorem 13

$$\frac{dm_t}{dt} = \lambda^2 m_t + A_t + B_t m_t + \lambda^2 C_t (m_t^2 + S_t) + \lambda^2 D_t (m_t^3 + 3m_t S_t) \quad (25)$$

$$\frac{dS_t}{dt} = \lambda^2 (m_t^2 + 3S_t) + 2B_t S_t + 4\lambda^2 C_t m_t S_t + 6\lambda^2 D_t (m_t^2 S_t + S_t^2). \quad (26)$$

5. Optimal control problem formulation

In this section, we show that the optimization problem 3, using the results derived from Theorem 9, can be reformulated as a standard optimal control problem (OCP), which conceptually is similar to Mitter and Newton (2003)⁴. Therefore, the presented variational approximation method to the path estimation problem for SDEs can be expressed as an OCP and as such leads to a standard formulation of necessary global optimality conditions in terms of Pontryagin's maximum principle. Consider the vector spaces $\mathcal{V} := \mathbb{R}^n \times \mathbb{R}^{\alpha \times n}$, $\hat{\mathcal{Z}} := \mathbb{R}^n \times \operatorname{Sym}(n, \mathbb{R}) \times \mathbb{R}^n \times \operatorname{Sym}(n, \mathbb{R})$ and define the trajectories

$$\begin{aligned} [0, T] \ni t \mapsto v^{(\ell)}(t) &:= (A_t^{(\ell)}, B_t^{(\ell)}) \in \hat{\mathcal{V}} \\ [0, T] \ni t \mapsto z^{(\ell)}(t) &:= (m_t^{(\ell)}, S_t^{(\ell)}, C_t^{(\ell)}, D_t^{(\ell)}) \in \hat{\mathcal{Z}}, \end{aligned}$$

for $\ell = 1, \dots, k$. We introduce the state variable $z(t) := (z^{(1)}(t), \dots, z^{(k)}(t)) \in \prod_{\ell=1}^k \hat{\mathcal{Z}} :=: \mathcal{Z}$ and the control variable $v(t) := (v^{(1)}(t), \dots, v^{(k)}(t)) \in \prod_{\ell=1}^k \hat{\mathcal{V}} =: \mathcal{V}$ for $t \in [0, T]$. As a first step, in view of the cost functional (16) of Problem 3, the so-called Lagrangian

$$\begin{aligned} \mathbb{E}_Q \left[\frac{1}{2} \|u(X_t, t) - f(X_t)\|_{\alpha(X_t)}^2 \right. \\ \left. + y_t \left(u(X_t, t)^\top \nabla h(X_t) + \frac{1}{2} \sigma(X_t)^\top \nabla^2 h(X_t) \sigma(X_t) \right) + \frac{1}{2} \|h(X_t)\|^2 \right] \end{aligned} \quad (27)$$

is expressed as a function of only $z(t), v(t)$ and t . This step, while being exact in some cases, may require an approximation. In the case that the marginals of Q are mixtures of normal densities, the expectation of any polynomial in X_t can be expressed as a function of its mean and variance. If the diffusion term σ is a polynomial, and no mixture is considered ($k = 1$), the drift function u , according to (20), is a polynomial. We refer to Section 8 to see how the Lagrangian can be derived for two concrete examples. Consider a Lagrangian

$$L : [0, T] \times \mathcal{Z} \times \mathcal{V} \rightarrow \mathbb{R}, \quad L(t, z(t), v(t)) \approx (27),$$

where \approx indicates that in order to express the term (27) by the state and control variables only, an approximation might be needed, as explained above. Similarly to the Lagrangian, in view of the cost functional (16), we introduce a terminal cost $F : \mathcal{Z} \rightarrow \mathbb{R}$ by

$$F(z(T)) \approx -y_T \mathbb{E}_Q[h(X_T)].$$

⁴. Note that Mitter and Newton (2003) addresses a related problem, whose main difference, when compared to the presented method, is that the variational characterization considered there is exact.

Under the assumption that the drift term σ is a polynomial, the ODEs derived in the previous section can be expressed in standard form. We define the function $H : \mathcal{Z} \times \mathcal{V} \rightarrow \mathcal{Z}$ by

$$H(z(t), v(t)) = \left(H_1^{(1)}(z(t), v(t)), \dots, H_4^{(1)}(z(t), v(t)), \dots, H_1^{(k)}(z(t), v(t)), \dots, H_4^{(k)}(z(t), v(t)) \right),$$

where

$$\begin{aligned} \frac{dm_t^{(\ell)}}{dt} &= \frac{dz_1^{(\ell)}}{dt}(t) = H_1^{(\ell)}(z(t), v(t)), & \frac{dC_t^{(\ell)}}{dt} &= \frac{dz_3^{(\ell)}}{dt}(t) = H_3^{(\ell)}(z(t), v(t)), \\ \frac{dS_t^{(\ell)}}{dt} &= \frac{dz_2^{(\ell)}}{dt}(t) = H_2^{(\ell)}(z(t), v(t)), & \frac{dD_t^{(\ell)}}{dt} &= \frac{dz_4^{(\ell)}}{dt}(t) = H_4^{(\ell)}(z(t), v(t)), \end{aligned}$$

for $\ell = 1, \dots, k$ are given by (22), (23) and (21). Thus, we have shown so far in this article that Problem 3 can be reformulated as the following optimal control problem

$$\begin{cases} \text{minimize} & J(v) = \int_0^T L(t, z(t), v(t)) dt + F(z(T)) \\ \text{subject to} & \dot{z}(t) = H(z(t), v(t)), \quad t \in [0, T] \text{ a.e.} \\ & z(0) = z_0, \end{cases} \quad (28)$$

where $\mathcal{M}([0, T], \mathcal{V})$ denotes the space of measurable functions from $[0, T]$ to \mathcal{V} . It remains to discuss how to find the initial condition z_0 in the OCP (28). A straightforward, however, clearly not efficient, method for that is solving the Pardoux equation (8), which according to (9) provides the smoothing density at initial time as $\mathcal{P}_S(x, 0) = \frac{\rho_0(x) \nu(x, 0)}{\int_{\mathbb{R}^n} \rho_0(x) \nu(x, 0) dx}$, from where z_0 can be derived.

5.1 Maximum principle

We derive necessary conditions for global optimality of the optimization problem (28) that are provided by the Pontryagin maximum principle (PMP). Since the control set \mathcal{V} is unbounded, we need an extended setting of the standard PMP, see (Clarke, 2013, Section 22.4) for a comprehensive survey. It requires some further assumptions.

Assumption 15 Let the process $(z^*(t), v^*(t))_{t \in [0, T]}$ be a local minimizer for the OCP (28), that satisfies

- (i) The function F is continuously differentiable;
- (ii) The functions H and L are continuous and admit derivatives relative to z which are themselves continuous in all variables (t, z, v) ;
- (iii) There exist $\varepsilon > 0$, a constant c , and a summable function d such that for almost every $t \in [0, T]$, we have

$$|z - z^*(t)| \leq \varepsilon \Rightarrow |\nabla_z(H, L)(t, z, v^*(t))| \leq c(|(H, L)(t, z, v^*(t))| + d(t).$$

Note that Assumption 15(iii) is implied if

$$|\nabla_z H(t, z, v)| + |\nabla_z L(t, z, v)| \leq c(|H(t, z, v)| + |L(t, z, v)|) + d(t)$$

holds for all $v \in \mathcal{V}$ when z is restricted to a bounded set, which is satisfied by many systems. Moreover, the condition automatically holds if v^* happens to be bounded.

Lemma 16 (PMP (Clarke, 2013, Theorem 22.2)) *Given Assumption 15, let the process $(z^*(t), v^*(t))_{t \in [0, T]}$ be a local minimizer for the problem (28). Then there exists an absolutely continuous function $p : [0, T] \rightarrow \mathcal{Z}$ satisfying*

1. *the adjoint equation $\dot{p}(t) = -\nabla_z \langle p(t), H(z^*(t), v^*(t)) \rangle - \nabla_z L(t, z^*(t), v^*(t))$ for almost every $t \in [0, T]$;*
2. *the transversality condition $p(T) = \nabla_z F(z(T))$;*
3. *the maximum condition*

$$\langle p(t), H(z^*(t), v^*(t)) \rangle + L(t, z^*(t), v^*(t)) = \inf_{v \in \mathcal{V}} \langle p(t), H(z^*(t), v) \rangle + L(t, z^*(t), v)$$
almost every $t \in [0, T]$.

Remark 17 1. Given that an optimal process (z^*, v^*) exists⁵, the maximum condition 3 can be used to derive a feedback law

$$v^*(t) \in \arg \min_{v \in \mathcal{V}} \langle p(t), H(z^*(t), v) \rangle + L(t, z^*(t), v).$$

2. Lemma 16, basically leads to a boundary value problem with initial conditions for the states and terminal conditions for the adjoint states, that provides necessary conditions for global optimality of Problem 3.

We summarize the described method to approximate the smoothing density introduced so far. It basically consists of the following three steps, that provide a solution to Problem 3:

- Step 1** Fix a mixture of exponential families of probability densities, e.g., the mixture of multivariate normal densities. Theorem 9, that simplifies to Proposition 11 for the multivariate normal densities, characterizes the approximate posterior SDE (15) whose solution admits marginal densities evolving in the chosen mixture of exponential families.

- Step 2** Given the approximate posterior SDE (15), we derive an optimal control formulation of Problem 3. For the mixture of multivariate normal densities, this derivation is presented in Sections 4 and 5 and finally leads to the OCP (28).

- Step 3** Necessary conditions for optimality of the OCP (28), and hence for Problem 3, can be derived from Pontryagin's maximum principle and result in a structured boundary value problem.

5. Existence of an optimal process can be assured by standard existence results, see for example (Clarke, 2013, Theorem 23.11).

Remark 18 It is important to note that the presented method chooses the best approximating SDE in a desired class using an objective distance measure between the corresponding probability distributions. One crucial advantage of this approach is that this distance could be quantified and numerically calculated (note that the first term in Lemma 2 can be directly computed and the remaining two terms form the objective function of the optimal control problem considered), and hence the user gets an excellent estimate on the necessary approximating error. For instance, Figure 1d and Figure 2d demonstrated the accuracy of corresponding approximating SDEs by plotting the relative entropies between the approximate models and the exact ones for the two examples considered in the paper.

5.2 Computational complexity

If the initial condition to the OCP (28) is known, the PMP, Lemma 16, reduces to a boundary value problem, that can usually be solved numerically more efficiently than (S)PDEs by using numerical methods specifically tailored to these problems, such as the shooting method, see Stoer and Bulirsch (2002). Therefore, the major computational difficulty of the presented variational approach lies in estimating the initial condition to the OCP (28), for example via estimating the smoothing density at initial time. A straightforward, however clearly not efficient, method for that is solving the Pardoux equation (8), as explained in Section 2, which we used in the numerical examples in Section 8. As such, whereas the standard PDE approach for computing a smoothing density requires solving a Zakai equation and the Pardoux equation (8), the presented variational approach relies on only a Pardoux equation and the mentioned boundary value problem. This can usually be seen as a reduction in terms of computational effort required and is demonstrated by two numerical examples in Section 8, Table 1. Moreover, for future work, we aim to study the derivation of an estimator for the marginal smoothing density at terminal time without solving a Pardoux equation, that would then allow us to apply the proposed variational approximation method to high-dimensional problems, see Section 9 for more details. Another idea to circumvent the estimation of this mentioned terminal condition is to use an alternative approach to the PMP, for characterizing a solution to the OCP (28) that is briefly described in the following remark.

Remark 19 (Semidefinite programming) Solutions to the OCP (28) can be characterized via the so-called weak formulation which consists of an infinite-dimensional linear program, see (Lasserre, 2010, Chapter 10) for details. Therefore, numerical approximation schemes to such infinite-dimensional linear programs, that have been studied in the literature, can be employed to solve Problem 3. This approach seems particularly promising when the data of the OCP (dynamics and costs) are described by polynomials, as then the seminal Lasserre hierarchy based on solving a sequence of semidefinite programs, is applicable Lasserre (2001, 2010).

5.3 Example: Geometric Brownian Motion

We continue the geometric Brownian motion example started in Sections 2.1 and 4.3 and formulate the corresponding optimal control problem (28). Recall that the state variable is defined as $z(t) := (m_t, S_t, C_t, D_t)$ and the control variable as $v(t) := (A_t, B_t)$. The ODEs for

the state variables are given by (21), (25) and (26). The objective function of the optimal control problem (28) can be expressed as $F(x(T)) = -\gamma m_T$ and

$$\begin{aligned} L(t, z(t), v(t)) = & \frac{A_t^2}{2\lambda^2(m_t^2 + S_t)} + \frac{A_t(\lambda^2 + B_t - \kappa)}{\lambda^2 m_t} + \frac{(\lambda^2 + B_t - \kappa)^2}{2\lambda^2} + A_t C_t + \gamma_t A_t \\ & + m_t (C_t(\lambda^2 + B_t - \kappa) + A_t D_t + \gamma_t(\lambda^2 + B_t)) \\ & + (m_t^2 + S_t) \left(\frac{1}{2} \lambda^2 C_t^2 + D_t(\lambda^2 + B_t - \kappa) + \frac{1}{2} + \lambda^2 \gamma_t C_t \right) \\ & + (m_t^3 + 3m_t S_t) (\lambda^2 C_t D_t + \lambda^2 \gamma_t D_t) + (m_t^4 + 6m_t^2 S_t + 3S_t^2) \frac{\lambda^2}{2} D_t^2, \end{aligned}$$

where, in order to derive the cost function above, the first two inverse moments of X_t with respect to Q have been approximated. Due to the non-negativity of the GBM, we use the approximation $\mathbb{E}_Q[X_t^{-1}] \approx \mathbb{E}_Q[X_t^{-2}]^{-1} = m_t^{-1}$ and $\mathbb{E}_Q[X_t^{-2}] \approx \mathbb{E}_Q[X_t]^{-2} = (S_t + m_t^2)^{-1}$, whose accuracy has been investigated in Garcia and Palacios (2001).

6. Parameter inference

The goal of this section is to outline the use of the techniques, developed so far for path estimation, for inference of parameters in a hidden Markov model. We consider a class of dynamical systems

$$dX_t^\kappa = f(X_t^\kappa, \kappa)dt + \sigma(X_t^\kappa, \kappa)dW_t, \quad X_0^\kappa = x_0, \quad 0 \leq t \leq T, \quad (29)$$

parameterized by κ . The observation process can be modeled by (2), but as discussed in the next section, the approach discussed below can also be used with necessary modifications for a discrete observation process.

As a natural notation, for each parameter κ , the probability distribution of $X_{[0,T]}^\kappa$ on \mathcal{C} will be denoted by $\Pi_{\text{prior}}^\kappa$. Given a sample path $\{y_t : 0 \leq t \leq T\}$ of the observation process $Y_{[0,T]}$, the objective is to select an optimal $\kappa^* \in \mathbb{R}^d$ such that the observation process $(Y_t)_{t \in [0,T]}$ in (2) has a high probability of reproducing the given data y . This is basically the inference scheme based on classical maximum likelihood estimation, and we propose an algorithm similar to the lines of *expectation maximization (EM) algorithm* (see Cappé et al. (2005) for a comprehensive survey), which aims to obtain the optimal κ^* through multiple iterations. Recalling (14), for each κ , we define $I^\kappa(H_T(\cdot, y)) := -\log \left(\int \exp(-H_T(\cdot, y)) d\Pi_{\text{prior}}^\kappa \right)$. As already noted in Section 3, for each parameter κ , the term $I^\kappa(H_T(\cdot, y))$ provides the total information available through the sample path y , and can be interpreted as the negative log-likelihood of y given the parameter κ . However, minimizing this negative log-likelihood function, even if numerical evaluation of it can be done, usually is a hard problem. But, as mentioned in Section 3, Lemma 2 and non-negativity of the relative entropy together imply that an upper bound to this negative log-likelihood term is given by the apparent information, $\mathcal{F}(Q, \kappa) := D(Q || \Pi_{\text{prior}}^\kappa) + \mathbb{E}_Q[H_T(\cdot, y)]$. The advantage of this observation is that this upper bound to the negative log-likelihood function is also the objective function in Problem 3, for which the program for finding the minimizer Q is by now well-established. Therefore instead of minimizing the actual negative

log-likelihood, we minimize an upper bound of it. The path to find the *right* parameter κ corresponding to the sample path y is now quite standard in statistics. After initialization of the parameter κ , we find the optimal Q by solving the Problem 3, and then in the subsequent step, for this Q we obtain the optimal parameter κ by minimizing $\mathcal{F}(Q, \kappa)$. This yields an iterative EM-type algorithm whose details are given below.

EM-type algorithm

initialize $i = 0, \kappa_i := \hat{\kappa}_0$
while $i \leq M$
Step 1: compute Q_i by solving Problem 3 with parameter κ_i
Step 2: update parameter as $\kappa_{i+1} \in \arg \min_{\kappa} \mathcal{F}(Q_i, \kappa)$
Step 3: set $i \rightarrow i + 1$

Remark 20 Analyzing convergence of the above algorithm and consistency of the above corresponding estimator is the next important step and will be addressed in our future projects.

We refer to Section 8 for a numerical visualization of this variational parameter inference method applied to two examples and to Section 9 for a discussion about convergence and consistency of the estimator as a topic of further research.

7. Discrete time measurement model

In most practical examples, the measurements of physical quantities are processed by computers, and as such the data available are obtained only at discrete times, potentially restricted to a low number. The goal of this section is to outline how the discussed variational approximation scheme adapts naturally to such cases with obvious modifications.

In this case the signal process (1) is observed through noisy measured data $y := \{y_k\}_{k=1}^N$ at discrete times $t_1 \leq t_2 \leq \dots \leq t_N \leq T$. The canonical model for the observation process is thus given by

$$Y_k = h(X_k, t_k) + \rho_k, \quad \text{for } k = 1, \dots, N, \quad (30)$$

where $X_k := X_{t_k}$, $h: \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}^n$ is a measurable function, the ρ_k are \mathbb{R}^n -valued i.i.d. Gaussian random variables with zero mean and covariance R_k , and they are independent of x_0 and $\sigma(W_s : s \leq T)$. We consider m such that $t_m \leq t < t_{m+1}$ and similarly to Section 2 define the filter density p and smoothing density \mathcal{P}_S by

$$\mathbb{E}[\phi(X(t)) | Y_1, \dots, Y_m, x_0] = \int \phi(x) p(x, t) dx \quad (31)$$

$$\mathbb{E}[\phi(X(t)) | Y_1, \dots, Y_N, x_0] = \int \phi(x) \mathcal{P}_S(x, t) dx, \quad (32)$$

where ϕ is any measurable function from \mathbb{R}^n to \mathbb{R} . It is well known (see (Eynik, 2000, Appendix)) for a derivation) that the smoothing can be expressed as

$$\mathcal{P}_S(x, t) = \frac{p(x, t) w(x, t)}{\int_{\mathbb{R}^n} p(x, t) w(x, t) dx}, \quad (33)$$

where $p(x, t)$ and $w(x, t)$ in between the observation times are the solutions to the

$$\begin{cases} dp(x, t) = \mathcal{A}^* p(x, t) dt \\ p(x, 0) = p_0(x), \end{cases} \quad \text{Kolmogorov forward equation:} \quad (34)$$

$$\begin{cases} dw(x, t) = -\mathcal{A} w(x, t) dt \\ w(x, T) = 1, \end{cases} \quad \text{Kolmogorov backward equation:} \quad (35)$$

punctuated by jumps at the data points t_k for $k = 1, \dots, N$

$$p(x, t_k^+) \propto p(x, t_k) \exp\left(y_k^\top R_k^{-1} h(x, t_k) - \frac{1}{2} h(x, t_k)^\top R_k^{-1} h(x, t_k)\right) \quad (36)$$

$$w(x, t_k) \propto w(x, t_k^+) \exp\left(y_k^\top R_k^{-1} h(x, t_k) - \frac{1}{2} h(x, t_k)^\top R_k^{-1} h(x, t_k)\right). \quad (37)$$

Similar to the continuous time measurement model, it can be shown that the smoothing density solves the Kolmogorov forward equation given by (10), with drift function $g(x, t) := f(x) + a(x) \nabla \log w(x, t)$, where w is the solution to (35). As before, we denote the prior probability measure by $\Pi_{\text{prior}}(\mathcal{A}) = \mathbb{P}[X_{[0, T]} \in \mathcal{A}]$ and the posterior probability measure, induced by the solution to (10), by $\Pi_{\text{post}}(\mathcal{A}, Y) = \mathbb{P}[X_{[0, T]} \in \mathcal{A} | \mathcal{F}_T^Y]$, where $\mathcal{F}_T^Y = \sigma(x_0, Y_1, \dots, Y_N)$. Let y_k denote a realization of the observation process at the time t_k . The variational approximation derived in Section 3, and, in particular, Problem 3 carries over to the discrete time observation setting considered here. As before, the path to the objective function starts from Lemma 2, which holds in this case with

$$H_T(X, y) := \sum_{i=1}^N \left(\frac{1}{2} \|R_k^{-1} h(X_i, t_i)\|^2 - y_i^\top R_k^{-1} h(X_i, t_i) \right). \quad (38)$$

One way to see this is to recast the discrete model in the traditional setup of Section 2, and then use the Kallianpur-Striebel theorem. To do this, first assume that without loss of generality $R_k = I$. Define the function $\bar{h}: C \times [0, T] \rightarrow \mathbb{R}^n$ by

$$\bar{h}(x, t) = \sum_{k=t}^N (t_{k+1} - t_k)^{-1/2} h(x \circ \eta(t), \eta(t)) \mathbb{1}_{\{t_k \leq t < t_{k+1}\}},$$

where $\eta: [0, T] \rightarrow [0, T]$ is defined as

$$\eta(t) = t_k, \quad \text{if } t_k \leq t < t_{k+1}.$$

Define the observation model $\bar{Y}_t = \int_0^t \bar{h}(X_s, s) ds + B_t$. Notice that for each k ,

$$\bar{Y}_{k+1} - \bar{Y}_k = (t_{k+1} - t_k)^{1/2} h(X_k, t_k) + (B(t_{k+1}) - B(t_k)),$$

and hence

$$(t_{k+1} - t_k)^{-1/2}(\tilde{Y}_{k+1} - \tilde{Y}_k) = h(X_k, t_k) + \tilde{\rho}_k,$$

where $\tilde{\rho}_k \stackrel{Law}{=} \rho_k \sim \mathcal{N}(0, I)$. In other words, $(t_{k+1} - t_k)^{-1/2}(\tilde{Y}_{k+1} - \tilde{Y}_k) \stackrel{Law}{=} \tilde{Y}_k$, and in this sense the discrete measurement model can be subsumed in the observation model given by $\tilde{Y}_t = \int_0^t h(X_s, s) ds + \tilde{B}_t$.

Notice that by the definitions of \tilde{Y} and \bar{h} , the exponent in Kallianpur-Striebel formula is given by

$$\int_0^T \frac{1}{2} \|\bar{h}(X, s)\|^2 ds - \int_0^T \bar{h}(X, s) d\tilde{Y}(s) = \sum_{k=1}^N \left(\frac{1}{2} \|h(X_k, t_k)\|^2 - \frac{(\tilde{Y}_{k+1} - \tilde{Y}_k)^\top h(X_k, t_k)}{(t_{k+1} - t_k)^{1/2}} \right) - L_{\tilde{Y}}^w \sum_{k=1}^N \left(\frac{1}{2} \|h(X_k, t_k)\|^2 - Y_t^\top h(X_k, t_k) \right),$$

which leads to (38). Therefore, in this case the objective function in Problem 3 can be expressed as

$$D(Q \| \Pi)_{\text{prior}} + \mathbb{E}_Q[H_T(\cdot, y)] = \int_0^T \mathbb{E}_Q \left[\frac{1}{2} \|u(X_t, t) - f(X_t)\|_{\sigma(X_t)}^2 + \iota(X_t, t) \right] dt, \quad (39)$$

where

$$\iota(X_t, t) = \sum_{i=1}^N \left(y_i^\top R_k^{-1} h(X_t, t_i) - \frac{1}{2} \|R_k^{-1} h(X_t, t_i)\|^2 \right) \delta(t - t_i). \quad (40)$$

Section 4 is independent of the considered measurement model, and by following Section 5 we arrive at an optimal control problem (28), where the cost functional is replaced by (39). The derivation of necessary conditions for global optimality of the optimization problem (28), compared to the continuous time measurement model, here is somewhat nonstandard, due to the Dirac delta terms (40) involved in the Lagrangian. However, the problem can be seen as an OCP with so-called *intermediate constraints*, for which an extension of the PMP is available Dmitruk and Kaganovich (2008).

Assumption 21 Let the process $(z^*(t), v^*(t))_{t \in [0, T]}$ be a local minimizer for the optimal control problem (28), that satisfies

- (i) Assumptions 15(i) and (ii);
- (ii) v^* is measurable and essentially bounded.

Lemma 22 (Extended PMP) Let the process $(z^*(t), v^*(t))_{t \in [0, T]}$ be a local minimizer for the problem (28). Given Assumption 21, then there exists an absolutely continuous function $p: [0, T] \rightarrow \mathcal{Z}$ satisfying

1. the adjoint equation $\dot{p}(t) = -\nabla_z \langle p(t), H(z^*(t), v^*(t)) \rangle - \nabla_z L(t, z^*(t), v^*(t))$ for almost all $t \in [0, T]$;
2. the transversality conditions $p(t_i) = p(t_i^-) - \nabla_z \mathbb{E}_Q[\iota(X, t_i)]$ for $i = 1, \dots, N$ and $p(T) = 0$;

3. the maximum condition

$$\begin{aligned} & \langle p(t), H(z^*(t), v^*(t)) \rangle + L(t, z^*(t), v^*(t)) \\ &= \sup_{v \in \mathcal{M}([0, T], \mathcal{V})} \langle p(t), H(z^*(t), v(t)) \rangle + L(t, z^*(t), v(t)) \text{ for almost all } t \in [0, T]. \end{aligned}$$

Proof Follows directly from Dmitruk and Kaganovich (2008), when transforming problem (28) into an OCP with intermediate constraints. \blacksquare

Remark 23 1. Note that the data (measurements) enter the expression through the cost function, namely the term (40), which is nonzero only at measurement times $\{t_i\}_{i=1}^N$ and leads to jumps in the adjoint state.

2. Lemma 22, basically leads to a boundary value problem, that provides necessary conditions for optimality of Problem 3. See Section 5.2 for a discussion about how to numerically solve it. We refer to the numerical examples in Section 8 for the performance of such a solution.

8. Simulation results

In this section, we present two examples to illustrate the performance of the variational approximation method introduced. Both examples have important applications in mathematical finance. As a first example, we consider the geometric Brownian motion that we introduced as a running example in Sections 2.1, 4.3 and 5.3. The second example is concerned with the Cox-Ingersoll-Ross process, that is often used for describing the evolution of interest rates Cox et al. (1985).

8.1 Geometric Brownian motion

As presented in Sections 2.1, 4.3 and 5.3 we consider a one-dimensional geometric Brownian motion (GBM) (11) and assume that the available data are noisy observations $\{y_k\}_{k=1}^N$ at time t_k , modeled by the observation process

$$Y_k = X_{t_k} + \rho_k,$$

where $\{\rho_k\}_{k=1}^N$ are i.i.d. normal random variables with zero mean, standard deviation R and $t_N = T$.

PDE approach. As explained in Section 7, the smoothing density can be characterized by (33) that is the (normalized) product of two densities w and p . The first density satisfies equation (35) with jump conditions (37) at the measurement times and terminal condition $w(x, T) = \frac{1}{\sqrt{2\pi}R} \exp\left(-\frac{(x-y)^2}{2R^2}\right)$. Its marginals are shown in Figure 1a. The second density, called the filter density, is given by equation (34) with jump conditions (36) and initial condition $p(x, 0) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(\log x - \mu)^2}{2\sigma^2}\right)$ that is given by (11). Its marginals are shown in Figure 1b. The smoothing density is depicted in Figure 1c as the solid line.

Variational approximation. Following Section 4.3, the drift function for the approximating SDE (15) has to be chosen as (24). The optimal control problem can be formulated along the lines of Section 5.3, choosing the discrete-time measurement setting presented in Section 7. Note that Assumption 21 can be easily verified to hold, if we restrict the optimizers in (28) to bounded controls. We solve the boundary value problem obtained from Lemma 22 under the assumption that the smoothing density at initial time is available, see Section 5.5.2 for a discussion about this assumption. The solution is depicted in Figure 1c as the dashed line. Finally, Figure 1d shows the relative entropy between the marginals of the smoothing density obtained by the PDE approach and the variational method, and hence reflects the accuracy of the variational approximation.

Parameter inference. We consider the case where the drift parameter κ in (11) is assumed to be unknown. Figure 1e shows the performance of the EM-Algorithm introduced in Section 6 for an initial guess $\hat{\kappa}_0 = 4$ of the unknown parameter. It can be seen that the estimator $\hat{\kappa}$ is close to the true value of $\kappa = 1$ indicating the efficacy of our algorithm. Also, the algorithm converges quite rapidly.

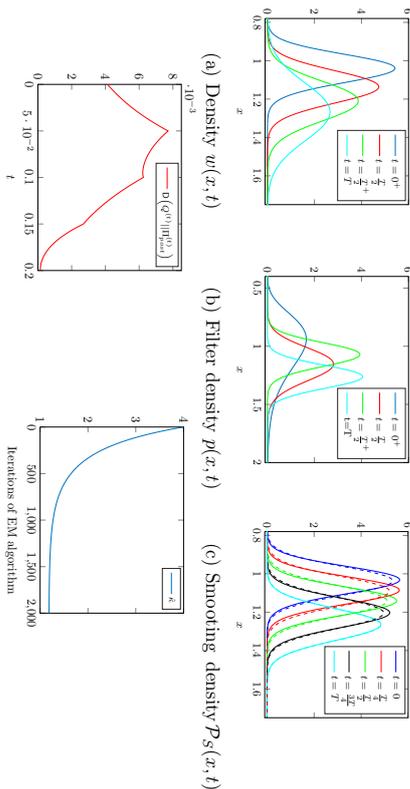


Figure 1: Geometric Brownian motion: Comparison of the PDE solution (solid) versus the variational approach (dashed). The considered numerical values are: $\kappa = 1$, $\lambda = 0.1$, $R = 0.15$, $T = 0.2s$, $\mu = 0$, $\sigma = 0.25$, $N = 4$, $t_1 = T/4$, $t_2 = T/2$, $t_3 = 3T/4$ and $t_4 = T$.

8.2 Cox-Ingersoll-Ross

Consider as underlying system a Cox-Ingersoll-Ross (CIR) process

$$dX_t = \kappa(b - X_t)dt + \lambda\sqrt{X_t}dW_t, \quad X_0 = x_0 \sim \mathcal{N}(\mu, \sigma), \quad (41)$$

for $0 \leq t \leq T$ and assume that the available data are noisy observations $\{y_k\}_{k=1}^N$ at time t_k , modeled by an observation process

$$Y_k = X_{t_k} + \rho_k,$$

where ρ_k are i.i.d. normal random variables with zero mean, standard deviation R and $t_N = T$.

PDE approach. As in the GBM example 8.1, we solve the underlying PDEs introduced in Section 7, to characterize the smoothing density as the (normalized) product of two densities v and p . Figure 2a shows the marginals of the first density w , the filter density p is depicted in Figure 2b and the smoothing density in Figure 2c as the solid line.

Variational approximation. The variational approximation is derived similarly to the GBM example 8.1, where we chose a drift function for the approximating SDE (15), according to Theorem 9, as

$$u(x, t) = \frac{1}{2}\lambda^2 x^2 + A(t) + B(t)x + \lambda^2 x(C(t) + D(t)x). \quad (42)$$

The variational approximation to the smoothing density is depicted in Figure 2c as the dashed line. Finally, Figure 2d shows the relative entropy between the marginals of the smoothing density obtained by the PDE solution and the variational approximation.

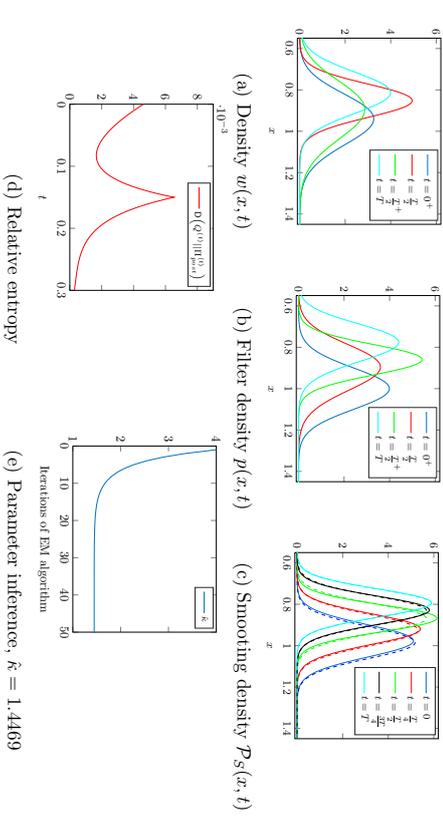


Figure 2: Cox-Ingersoll-Ross: Comparison of the PDE solution (solid) versus the variational approach (dashed). The considered numerical values are: $\lambda = 0.2$, $\kappa = 1$, $b = 0.3$, $\mu = 1$, $\sigma = 0.1$, $R = 0.1$, $T = 0.3s$, $N = 2$, $t_1 = T/2$ and $t_2 = T$.

Parameter inference. We consider the case where the parameter κ in the drift term of (41) is unknown. Figure 2e shows the EM-Algorithm introduced in Section 6 for an initial guess $\hat{\kappa}_0 = 4$ of the unknown parameter. It can be seen that the estimator $\hat{\kappa}$ is close to the true

Table 1: **Runtime comparison.** The presented variational approach for approximating the smoothing density is compared with the standard PDE approach for the two examples in Sections 8.1 and 8.2. All simulations are performed on a 2.3 GHz Intel Core i7 processor with 8 GB RAM using Matlab.

	Geometric Brownian motion	Cox-Ingersoll-Ross
Forward PDE (34)	2.02 s	1.33 s
Backward PDE (35)	2.87 s	2.38 s
Boundary value problem	0.10 s	0.23 s
PDE approach ⁶	4.89 s	3.70 s
Variational approach ⁷	2.97 s	2.61 s

value of $\kappa = 1$ indicating the efficacy of our algorithm. Also, the algorithm converges very fast.

Table 1 summarizes the runtimes of the two numerical examples above. It can be seen that the boundary value problems provided by the maximum principle can be solved by roughly one magnitude faster than the backward PDE (35), that is the reason for the speedup of the variational approach compared to the PDE approach. Moreover, it is highlighted that the main computational effort in the variational approach is needed to estimate the marginal smoothing density at initial time, which is done by solving the backward PDE (35). If, however, the backward density at initial time could be estimated in a more efficient way, e.g., by using an MCMC method, the proposed variational approximation method could be applicable to high-dimensional problems.

9. Conclusion

The paper is devoted to a variational method for estimating paths of a signal process in a hidden Markov model. In particular, this leads to approximations of smoothing density which can be used to reconstruct any past state of the signal process given a full set of observations. A crucial fact that plays an important role in our method is that the smoothing distribution is induced by a posterior SDE which itself is a modification of the original signal process. The presented variational approach proposes an approximate SDE which minimizes the relative entropy between the posterior SDE and a class of SDEs whose marginals belong to a chosen mixture of exponential families. In the simplest case of normal marginals and a posterior SDE with constant diffusion term, the approximating SDE consists of a linear drift and constant diffusion term, which is well known. It is shown that the prescribed approximation scheme can be formulated as an optimal control problem, and necessary conditions for global optimality are obtained by the Pontryagin maximum principle. The resulting numerical methods have considerable computational advantages over numerically solving the underlying (S)PDEs, that is highlighted by two examples. The developed approximation scheme is then used for designing an efficient method for parameter inference for SDEs.

6. consists of solving the two PDEs (34) and (35)

7. consists of the PDE (35) and the boundary value problem in order to solve Problem 3

For future work, as mentioned in Section 5.2, we aim to study how to efficiently estimate the backward density at initial time. Then, the presented variational approximation method reduces to solving a PMP-shooting-type boundary value problem that is tractable even in relatively large dimensions, compared to PDEs. Additionally, it would be interesting to study numerical methods specifically tailored to the boundary value problems resulting from the maximum principle, such as the shooting method, see Stoer and Bulirsch (2002) for a comprehensive summary, as well as the approach of solving the optimal control problem via its weak formulation as pointed out in Remark 19.

Our future projects will also delve into analyzing the convergence of the EM-type algorithm used for parameter inference as well as the properties of the obtained estimators. We will also focus on refining the basic inference algorithm to get better efficiency and speed. One promising path to take in this direction would be designing of suitable adaptive EM-type algorithms. It is also conceivable that the ideas mentioned in the paper can be combined with suitable MCMC schemes or techniques known as Assumed Density Filtering, see Harel et al. (2015), to get better accuracy and efficiency in high-dimensional models.

Acknowledgments

This work was supported by the ETH grant (ETH-15 12-2). The authors are grateful to Debasish Chatterjee, John Lygeros, and Peyman Mohajerin Esfahani for helpful discussions and pointers to references. H.K. acknowledges funding from the LOEWE research priority program ComputGene.

Appendix A. Derivation of Equation (10)

We consider the one-dimensional case; an extension to the multi-dimensional case is straightforward. According to (9) the smoothing density is given by $\mathcal{P}_S(x, t) = K(t)p(x, t)w(x, t)$, where $K(t) := (\int_{\mathbb{R}^n} p(x, t)w(x, t)dx)^{-1}$. The main idea is to recall that the process $(K(t))_{t \in [0, T]}$ is known to be almost surely constant (Pardoux, 1981/82, Theorem 3.2). Therefore

$$\begin{aligned} \frac{\partial}{\partial t} \mathcal{P}_S(x, t) &= K(t)p(x, t) \frac{\partial}{\partial t} w(x, t) + K(t)w(x, t) \frac{\partial}{\partial t} p(x, t) \\ &= \frac{\mathcal{P}_S(x, t)}{w(x, t)} \left(-f(x)w'(x, t) - \frac{1}{2}a(x)w''(x, t) - w(x, t)h(x)^\top dY_t' \right) \\ &\quad + w(x, t) \left(- \left(\frac{f(x)\mathcal{P}_S(x, t)}{w(x, t)} \right)' + \frac{1}{2} \left(\frac{a(x)\mathcal{P}_S(x)}{w(x, t)} \right)'' + \frac{\mathcal{P}_S(x, t)}{w(x, t)} h(x)^\top dY_t' \right). \end{aligned}$$

The proof follows by a straightforward computation. We compute in a preliminary step

$$\left(\frac{f(x)\mathcal{P}_S(x, t)}{w(x, t)} \right)' = \frac{1}{w^2(x, t)} \left((f'(x)\mathcal{P}_S(x, t) + f(x)\mathcal{P}_S'(x, t))w(x, t) - f(x)\mathcal{P}_S(x, t)w'(x, t) \right),$$

and

$$\begin{aligned} \left(\frac{a(x)\mathcal{P}_S(x, t)}{w(x, t)} \right)'' &= \frac{1}{w(x, t)} \left(a''(x)\mathcal{P}_S(x, t) + 2a'(x)\mathcal{P}_S'(x, t) + a(x)\mathcal{P}_S''(x, t) \right) \\ &\quad - \frac{1}{w^2(x, t)} \left(2w'(x, t)a'(x)\mathcal{P}_S(x, t) + 2w'(x, t)a(x)\mathcal{P}_S'(x, t) + a(x)\mathcal{P}_S(x, t)w''(x, t) \right) \\ &\quad + \frac{1}{w^3(x, t)} \left(2a(x)w'(x, t)^2\mathcal{P}_S(x, t) \right). \end{aligned}$$

Using this two preliminaries, we get

$$\begin{aligned} \frac{\partial}{\partial t} \mathcal{P}_S(x, t) &= -f'(x)\mathcal{P}_S(x, t) - f(x)\mathcal{P}_S'(x, t) + a'(x)\mathcal{P}_S'(x, t) + \frac{1}{2}(a''(x)\mathcal{P}_S(x, t) + a(x)\mathcal{P}_S''(x, t)) \\ &\quad - \frac{1}{w(x, t)} \left(a'(x)w'(x, t)\mathcal{P}_S(x, t) + a(x)w''(x, t)\mathcal{P}_S(x, t) + a(x)w'(x, t)\mathcal{P}_S'(x, t) \right) \\ &\quad + \frac{1}{w^2(x, t)} a(x)w'(x, t)^2\mathcal{P}_S(x, t) \\ &= - \left(f'(x)\mathcal{P}_S(x, t) + f(x)\mathcal{P}_S'(x, t) + \frac{1}{w(x, t)} \left(a'(x)w'(x, t)\mathcal{P}_S(x, t) \right. \right. \\ &\quad \left. \left. + a(x)w''(x, t)\mathcal{P}_S(x, t) + a(x)w'(x, t)\mathcal{P}_S'(x, t) - \frac{1}{w^2(x, t)} \left(a(x)w'(x, t)^2\mathcal{P}_S(x, t) \right) \right) \right. \\ &\quad \left. + \frac{1}{2} \left(a'(x)\mathcal{P}_S(x, t) + a(x)\mathcal{P}_S''(x, t) \right)' \right. \\ &= - \left(\left(f(x) + a(x) \frac{w'(x, t)}{w(x, t)} \right) \mathcal{P}_S(x, t) \right)' + \frac{1}{2} \left(a(x)\mathcal{P}_S(x, t) \right)'' \\ &= - \left(\left(f(x) + a(x) (\log w(x, t))' \right) \mathcal{P}_S(x, t) \right)' + \frac{1}{2} \left(a(x)\mathcal{P}_S(x, t) \right)'' , \end{aligned}$$

and as such (10) holds.

Appendix B. Proof of Theorem 9

Consider an arbitrary curve $t \mapsto p(\cdot, \Theta_t^{(1)}, \dots, \Theta_t^{(k)})$ evolving in $\text{EM}(c^{(1)}, \dots, c^{(k)})$. Define a diffusion

$$dZ_t = u(Z_t, t)dt + \sigma(Z_t)dB_t, \quad Z_0 = x_0,$$

with the given diffusion coefficient $a(\cdot) = \sigma(\cdot)\sigma(\cdot)^\top$. Clearly the density of Z_t coincides with $p(\cdot, \Theta_t^{(1)}, \dots, \Theta_t^{(k)})$ if and only if $p(\cdot, \Theta_t^{(1)}, \dots, \Theta_t^{(k)})$ satisfies the Kolmogorov forward equation for Z_t , i.e.,

$$\begin{aligned} \frac{\partial p(x, \Theta_t^{(1)}, \dots, \Theta_t^{(k)})}{\partial t} &= - \sum_{i=1}^n \frac{\partial}{\partial x_i} \left(u_i(x, t)p(x, \Theta_t^{(1)}, \dots, \Theta_t^{(k)}) \right) \\ &\quad + \frac{1}{2} \sum_{j=1}^n \sum_{j=1}^n \frac{\partial^2}{\partial x_j \partial x_j} \left(a_{ij}(x)p(x, \Theta_t^{(1)}, \dots, \Theta_t^{(k)}) \right). \end{aligned} \quad (43)$$

We will show this in two steps that (43) holds for the proposed drift term. Consider the decomposition $u_i(x, t) = g_i(x, t) + \gamma_i(x, t)$ for all $i = 1, \dots, n$, where

$$g_i(x, t) := \frac{1}{2} \sum_{j=1}^n \frac{\partial}{\partial x_j} a_{ij}(x) + \frac{1}{2} \sum_{j=1}^n a_{ij}(x) \frac{\partial}{\partial x_j} p(x, \Theta_t^{(1)}, \dots, \Theta_t^{(k)}) \quad (44)$$

and

$$\gamma_i(x, t) := \frac{-1}{p(x, \Theta_t^{(1)}, \dots, \Theta_t^{(k)})} \sum_{\ell=1}^k \nu_{q\ell}(x, \Theta_t^{(\ell)}) \left\langle \dot{\Theta}_t^{(0)}, \mathcal{I}_\ell^{(0)}(x) \right\rangle. \quad (45)$$

We use the shorthand notation $p(x, \Theta_t^{(1k)}) := p(x, \Theta_t^{(1)}, \dots, \Theta_t^{(k)})$.

Claim 24 *The functions g_i defined in (44) for all $i = 1, \dots, n$ satisfy*

$$\sum_{i=1}^n \frac{\partial}{\partial x_i} \left(g_i(x, t)p(x, \Theta_t^{(1k)}) \right) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \frac{\partial^2}{\partial x_i \partial x_j} \left(a_{ij}(x)p(x, \Theta_t^{(1k)}) \right).$$

Claim 24 follows from a straightforward computation, see Appendix B in the extended version Sutter et al. (2015) for a detailed derivation.

Claim 25 *The functions γ_i defined in (45) for all $i = 1, \dots, n$ satisfy*

$$\frac{\partial}{\partial t} p(x, \Theta_t^{(1k)}) = - \sum_{i=1}^n \frac{\partial}{\partial x_i} \left(\gamma_i(x, t)p(x, \Theta_t^{(1k)}) \right).$$

Proof.

$$\frac{\partial}{\partial t} p(x, \Theta_t^{(1k)}) = \sum_{\ell=1}^k \nu_\ell \frac{\partial}{\partial t} p(x, \Theta_t^{(\ell)}) = \sum_{\ell=1}^k \nu_\ell \left\langle \dot{\Theta}_t^{(\ell)}, c^{(\ell)}(x) - \nabla \theta \psi_\ell(\Theta_t^{(\ell)}) \right\rangle p(x, \Theta_t^{(\ell)}).$$

Moreover,

$$\frac{\partial}{\partial x_i} \gamma_i(x, t) = \frac{1}{p(x, \Theta_t^{(1k)})^2} \left(\frac{\partial}{\partial x_i} p(x, \Theta_t^{(1k)}) \right) \sum_{\ell=1}^k \nu_\ell q\ell(x, \Theta_t^{(\ell)}) \left\langle \dot{\Theta}_t^{(\ell)}, \mathcal{I}_\ell^{(0)}(x) \right\rangle$$

$$- \frac{1}{p(x, \Theta_t^{(\ell)})} \sum_{\ell=1}^k \nu_\ell \left\langle \dot{\Theta}_t^{(\ell)}, \varphi_i^{(\ell)}(x, \Theta_t^{(\ell)}) \exp \left(\langle \Theta_t^{(\ell)}, c^{(\ell)}(x) \rangle - \psi_t(\Theta_t^{(\ell)}) \right) \right\rangle,$$

where we used

$$\begin{aligned} p_t(x, \Theta_t^{(\ell)}) \left\langle \dot{\Theta}_t^{(\ell)}, \mathcal{I}_t^{(\ell)} \right\rangle \\ = \left\langle \dot{\Theta}_t^{(\ell)}, \int_{-\infty}^{x_i} \varphi_i^{(\ell)}(x-i, \xi_i), \Theta_t^{(\ell)} \exp \left(\langle \Theta_t^{(\ell)}, c^{(\ell)}(x-i, \xi_i) \rangle - \psi_t(\Theta_t^{(\ell)}) \right) d\xi_i \right\rangle. \end{aligned}$$

Therefore,

$$\begin{aligned} \frac{\partial}{\partial x_i} \left(\gamma_i(x, t) p(x, \Theta_t^{(1:k)}) \right) &= \left(\frac{\partial}{\partial x_i} \gamma_i(x, t) \right) p(x, \Theta_t^{(1:k)}) + \gamma_i(x, t) \left(\frac{\partial}{\partial x_i} p(x, \Theta_t^{(1:k)}) \right) \\ &= - \sum_{\ell=1}^k \nu_\ell \left\langle \dot{\Theta}_t^{(\ell)}, \varphi_i^{(\ell)}(x, \Theta_t^{(\ell)}) \right\rangle p_t(x, \Theta_t^{(\ell)}), \end{aligned}$$

and

$$\begin{aligned} - \sum_{i=1}^n \frac{\partial}{\partial x_i} \left(\gamma_i(x, t) p(x, \Theta_t^{(1:k)}) \right) &= \sum_{\ell=1}^k \nu_\ell p_t(x, \Theta_t^{(\ell)}) \left(\sum_{i=1}^n \left\langle \dot{\Theta}_t^{(\ell)}, \varphi_i^{(\ell)}(x, \Theta_t^{(\ell)}) \right\rangle \right) \\ &= \sum_{\ell=1}^k \nu_\ell p_t(x, \Theta_t^{(\ell)}) \left\langle \dot{\Theta}_t^{(\ell)}, c^{(\ell)}(x) - \nabla_{\theta} \psi_t(\Theta_t^{(\ell)}) \right\rangle = \frac{\partial}{\partial t} p(x, \Theta_t^{(1:k)}), \end{aligned}$$

where we used (18). ■

The two claims imply (43) and hence complete the proof. ■

Appendix C. Proof of Proposition 11

The proof basically requires Theorem 9 and two additional lemmas. We first propose in Lemma 26 a choice of functions $\varphi_i^{(\ell)}$ that satisfy (18) in Theorem 9. Then we show in a second step, in Lemma 27, that for this choice the integral terms (17) admit a closed form expression. We start with a few preparatory results that are needed to prove Proposition 11. Note that $\nabla_{\Theta} \psi(\Theta) = (\nabla_{\eta} \psi(\Theta), \nabla_{\theta} \psi(\Theta)) \in \mathcal{H}$ and recall that according to (Bernstein, 2009, p.631) for $A \in \mathbb{R}^{n \times m}$, $B \in \mathbb{R}^{m \times n}$ and $X \in \text{GL}(n, \mathbb{R})$ $\frac{d}{dX} \text{tr}(AX^{-1}B) = -X^{-1}BAX^{-1}$ and $\frac{d}{dX} \log \det(AX^{-1}B) = -X^{-1}B(AX^{-1}B)^{-1}AX^{-1}$ and therefore

$$\nabla_{\eta} \psi(\Theta) = -\frac{1}{2} \theta^{-1} \eta, \quad \nabla_{\theta} \psi(\Theta) = \theta^{-1} \left(\frac{1}{4} \eta \eta^{\top} \theta^{-1} - \frac{1}{2} \mathbf{I}_n \right). \quad (46)$$

Lemma 26 Consider the functions $\varphi_i^{(\ell)} : \mathbb{R}^n \times \mathcal{H} \rightarrow \mathcal{H}$, where $\varphi_i^{(\ell)} = \left(\varphi_{1,i}^{(\ell)}, \varphi_{2,i}^{(\ell)} \right)$ with $\varphi_{1,i}^{(\ell)} : \mathbb{R}^n \times \mathcal{H} \rightarrow \mathbb{R}^n$ and $\varphi_{2,i}^{(\ell)} : \mathbb{R}^n \times \mathcal{H} \rightarrow \mathbb{R}^{n \times n}$ for $\ell = 1, \dots, k$. Let $\varphi_{1,i}^{(\ell)}$ and $\varphi_{2,i}^{(\ell)}$ be defined as

$$\varphi_{1,i}^{(\ell)}(x-i, \xi_i), \Theta_t^{(\ell)} := \theta_t^{(\ell)-1} E_i \theta_t^{(\ell)}(c_1^{(\ell)}(x-i, \xi_i) - \nabla_{\eta} \psi_t(\Theta_t^{(\ell)}))$$

$$\begin{aligned} \varphi_{2,i}^{(\ell)}(x-i, \xi_i), \Theta_t^{(\ell)} &:= \theta_t^{(\ell)-1} E_i \left(\theta_t^{(\ell)}(c_2^{(\ell)}(x-i, \xi_i) - \nabla_{\theta} \psi_t(\Theta_t^{(\ell)})) \Theta_t^{(\ell)} \right. \\ &\quad \left. - \frac{1}{2} \theta_t^{(\ell)}(x-i, \xi_i) \eta_t^{(\ell)\top} + \frac{1}{2} \eta_t^{(\ell)}(x-i, \xi_i)^{\top} \theta_t^{(\ell)} \right) \theta_t^{(\ell)-1}. \end{aligned}$$

Then, (18) holds for all $\ell = 1, \dots, k$.

Proof of Lemma 26. According to (19) we have

$$\sum_{i=1}^n \left\langle \dot{\Theta}_t^{(\ell)}, \varphi_i^{(\ell)}(x, \Theta_t^{(\ell)}) \right\rangle = \sum_{i=1}^n \left(\left\langle \dot{\eta}_t^{(\ell)}, \varphi_{1,i}^{(\ell)}(x, \Theta_t^{(\ell)}) \right\rangle + \left\langle \dot{\theta}_t^{(\ell)}, \varphi_{2,i}^{(\ell)}(x, \Theta_t^{(\ell)}) \right\rangle \right),$$

consisting of the two components

$$\begin{aligned} \sum_{i=1}^n \left\langle \dot{\eta}_t^{(\ell)}, \varphi_{1,i}^{(\ell)}(x, \Theta_t^{(\ell)}) \right\rangle &= \sum_{i=1}^n \left\langle \dot{\eta}_t^{(\ell)}, \theta_t^{(\ell)-1} E_i \theta_t^{(\ell)}(c_1^{(\ell)}(x) - \nabla_{\eta} \psi_t(\Theta_t^{(\ell)})) \right\rangle \\ &= \left\langle \dot{\eta}_t^{(\ell)}, \sum_{i=1}^n \theta_t^{(\ell)-1} E_i \theta_t^{(\ell)}(c_1^{(\ell)}(x) - \nabla_{\eta} \psi_t(\Theta_t^{(\ell)})) \right\rangle = \left\langle \dot{\eta}_t^{(\ell)}, c_1^{(\ell)}(x) - \nabla_{\eta} \psi_t(\Theta_t^{(\ell)}) \right\rangle \end{aligned}$$

and

$$\begin{aligned} \sum_{i=1}^n \left\langle \dot{\theta}_t^{(\ell)}, \varphi_{2,i}^{(\ell)}(x, \Theta_t^{(\ell)}) \right\rangle &= \left\langle \dot{\theta}_t^{(\ell)}, \sum_{i=1}^n \left(\theta_t^{(\ell)-1} E_i \theta_t^{(\ell)}(c_2^{(\ell)}(x) - \nabla_{\theta} \psi_t(\Theta_t^{(\ell)})) \right. \right. \\ &\quad \left. \left. - \frac{1}{2} \theta_t^{(\ell)-1} E_i \theta_t^{(\ell)} x \eta_t^{(\ell)\top} \theta_t^{(\ell)-1} + \frac{1}{2} \theta_t^{(\ell)-1} E_i \eta_t^{(\ell)} x^{\top} \right) \right\rangle \\ &= \left\langle \dot{\theta}_t^{(\ell)}, c_2^{(\ell)}(x) - \nabla_{\theta} \psi_t(\Theta_t^{(\ell)}) \right\rangle - \frac{1}{2} \left\langle \dot{\theta}_t^{(\ell)}, x \eta_t^{(\ell)\top} \theta_t^{(\ell)-1} \right\rangle + \frac{1}{2} \left\langle \dot{\theta}_t^{(\ell)}, \theta_t^{(\ell)-1} \eta_t^{(\ell)} x^{\top} \right\rangle \\ &= \left\langle \dot{\theta}_t^{(\ell)}, c_2^{(\ell)}(x) - \nabla_{\theta} \psi_t(\Theta_t^{(\ell)}) \right\rangle, \end{aligned}$$

where we have used in the last step that $\left\langle \dot{\theta}_t^{(\ell)}, x \eta_t^{(\ell)\top} \theta_t^{(\ell)-1} \right\rangle = \left\langle \dot{\theta}_t^{(\ell)}, \theta_t^{(\ell)-1} \eta_t^{(\ell)} x^{\top} \right\rangle$, since for $A \in \text{Sym}(n, \mathbb{R})$ and $B \in \mathbb{R}^{n \times n}$ $\text{tr}(AB^{\top}) = \text{tr}(AB)$. ■

Lemma 27 For $i = 1, \dots, n$, $j = 1, 2$ and $\ell = 1, \dots, k$ consider

$$\mathcal{I}_{j,i}^{(\ell)}(s_i, x) := \int_{-\infty}^{s_i} \varphi_{j,i}^{(\ell)}((x-i, \xi_i), \Theta_t^{(\ell)}) \exp \left(\left\langle \dot{\Theta}_t^{(\ell)}, c^{(\ell)}(x-i, \xi_i) - c^{(\ell)}(x) \right\rangle \right) d\xi_i,$$

where the functions $\varphi_{j,i}^{(\ell)}$ are chosen according to Lemma 26. Then,

$$\begin{aligned} \mathcal{I}_{1,i}^{(\ell)}(s_i, x) &= \frac{1}{2} \theta_t^{(\ell)-1} e_i \exp \left(\left\langle \dot{\Theta}_t^{(\ell)}, c^{(\ell)}(x-i, s_i) - c^{(\ell)}(x) \right\rangle \right) \\ \mathcal{I}_{2,i}^{(\ell)}(x_i, x) &= \frac{1}{4} \theta_t^{(\ell)-1} e_i (2\theta_t^{(\ell)} x - \eta_t^{(\ell)})^{\top} \theta_t^{(\ell)-1}. \end{aligned}$$

Note that $\mathcal{I}_i^{(\theta)}(x) = (\mathcal{I}_{1,i}^{(\theta)}(x_i, x), \mathcal{I}_{2,i}^{(\theta)}(x_i, x))$, where $\mathcal{I}_i^{(\theta)}(x)$ is the function defined in (17) and e_i denote the canonical basis vectors of \mathbb{R}^n .

Proof of Lemma 27.

$$\begin{aligned} \mathcal{I}_{1,i}^{(\theta)}(s_i, x) &= \int_{-\infty}^{s_i} \varphi_{1,i}^{(\theta)}(x-i, \xi_i) \Theta_i^{(\theta)} \exp \left(\left\langle \Theta_i^{(\theta)}, c^{(\theta)}(x-i, \xi_i) - c^{(\theta)}(x) \right\rangle \right) d\xi_i \\ &= \frac{1}{2} \theta_i^{(\theta)-1} \int_{-\infty}^{s_i} E_i 2\theta_i^{(\theta)} (c_1^{(\theta)}(x-i, \xi_i) - \nabla_{\eta} \psi_i(\Theta_i^{(\theta)})) \exp \left(\left\langle \Theta_i^{(\theta)}, c^{(\theta)}(x-i, \xi_i) - c^{(\theta)}(x) \right\rangle \right) d\xi_i \\ &= \frac{1}{2} \theta_i^{(\theta)-1} \int_{-\infty}^{s_i} E_i 2\theta_i^{(\theta)} \left((x-i, \xi_i) + \frac{1}{2} \theta_i^{(\theta)-1} \eta_i^{(\theta)} \right) \exp \left(\left\langle \Theta_i^{(\theta)}, c^{(\theta)}(x-i, \xi_i) - c^{(\theta)}(x) \right\rangle \right) d\xi_i \\ &= \frac{1}{2} \theta_i^{(\theta)-1} \int_{-\infty}^{s_i} E_i \left(2\theta_i^{(\theta)}(x-i, \xi_i) + \eta_i^{(\theta)} \right) \exp \left(\left\langle \Theta_i^{(\theta)}, c^{(\theta)}(x-i, \xi_i) - c^{(\theta)}(x) \right\rangle \right) d\xi_i, \end{aligned}$$

where (46) was used. Consider the substitution $z := \left\langle \Theta_i^{(\theta)}, c^{(\theta)}(x-i, \xi_i) - c^{(\theta)}(x) \right\rangle$ that leads to

$$\mathcal{I}_{1,i}^{(\theta)}(s_i, x) = \frac{1}{2} \theta_i^{(\theta)-1} e_i \exp \left(\left\langle \Theta_i^{(\theta)}, c^{(\theta)}(x-i, s_i) - c^{(\theta)}(x) \right\rangle \right),$$

where we used that $\theta_i^{(\theta)} < 0$, since $\theta_i^{(\theta)} = -\frac{1}{2} S_i^{(\theta)-1}$ and the inverse of a negative definite matrix is negative definite. For the second integral term

$$\begin{aligned} \mathcal{I}_{2,i}^{(\theta)}(x) &= \int_{-\infty}^{x_i} \varphi_{2,i}^{(\theta)}((x-i, \xi_i), \Theta_i^{(\theta)}) \exp \left(\left\langle \Theta_i^{(\theta)}, c^{(\theta)}(x-i, \xi_i) - c^{(\theta)}(x) \right\rangle \right) d\xi_i \\ &= \frac{1}{4} \theta_i^{(\theta)-1} \int_{-\infty}^{x_i} E_i \left(4\theta_i^{(\theta)} (c_2^{(\theta)}(x-i, \xi_i) - \nabla_{\theta} \psi_i(\Theta_i^{(\theta)})) \theta_i^{(\theta)} - 2\theta_i^{(\theta)} (x-i, \xi_i) \eta_i^{(\theta)\top} \right. \\ &\quad \left. + 2\eta_i^{(\theta)}(x-i, \xi_i)^\top \theta_i^{(\theta)} \right) \exp \left(\left\langle \Theta_i^{(\theta)}, c^{(\theta)}(x-i, \xi_i) - c^{(\theta)}(x) \right\rangle \right) d\xi_i \\ &= \frac{1}{4} \theta_i^{(\theta)-1} \int_{-\infty}^{x_i} E_i \left(2\theta_i^{(\theta)}(x-i, \xi_i)^\top 2\theta_i^{(\theta)} - \eta_i^{(\theta)\top} \eta_i^{(\theta)} + 2\theta_i^{(\theta)} \right. \\ &\quad \left. - 2\theta_i^{(\theta)}(x-i, \xi_i) \eta_i^{(\theta)\top} + 2\eta_i^{(\theta)}(x-i, \xi_i)^\top \theta_i^{(\theta)} \right) \theta_i^{(\theta)-1} \exp \left(\left\langle \Theta_i^{(\theta)}, c^{(\theta)}(x-i, \xi_i) - c^{(\theta)}(x) \right\rangle \right) d\xi_i, \end{aligned}$$

where we have used (46). By expanding terms and using integration by parts together with the first assertion of this lemma

$$\begin{aligned} \mathcal{I}_{2,i}^{(\theta)}(x) &= \frac{1}{2} \int_{-\infty}^{x_i} \frac{1}{2} \theta_i^{(\theta)-1} E_i (2\theta_i^{(\theta)}(x-i, \xi_i) + \eta_i^{(\theta)}) (2\theta_i^{(\theta)}(x-i, \xi_i) - \eta_i^{(\theta)\top} \theta_i^{(\theta)-1} \\ &\quad \exp \left(\left\langle \Theta_i^{(\theta)}, c^{(\theta)}(x-i, \xi_i) - c^{(\theta)}(x) \right\rangle \right) d\xi_i \\ &\quad + \frac{1}{2} \int_{-\infty}^{x_i} \theta_i^{(\theta)-1} E_i \exp \left(\left\langle \Theta_i^{(\theta)}, c^{(\theta)}(x-i, \xi_i) - c^{(\theta)}(x) \right\rangle \right) d\xi_i \\ &= \frac{1}{2} \mathcal{I}_{1,i}^{(\theta)}(x_i, x) (2\theta_i^{(\theta)} x - \eta_i^{(\theta)})^\top \theta_i^{(\theta)-1} - \frac{1}{2} \int_{-\infty}^{x_i} \mathcal{I}_{1,i}^{(\theta)}(\xi_i, x) (2\theta_i^{(\theta)} e_i)^\top \theta_i^{(\theta)-1} d\xi_i \\ &\quad + \frac{1}{2} \int_{-\infty}^{x_i} \theta_i^{(\theta)-1} E_i \exp \left(\left\langle \Theta_i^{(\theta)}, c^{(\theta)}(x-i, \xi_i) - c^{(\theta)}(x) \right\rangle \right) d\xi_i \end{aligned}$$

$$\begin{aligned} &= \frac{1}{4} \theta_i^{(\theta)-1} e_i (2\theta_i^{(\theta)} x - \eta_i^{(\theta)})^\top \theta_i^{(\theta)-1} \\ &\quad - \frac{1}{2} \int_{-\infty}^{x_i} \frac{1}{2} \theta_i^{(\theta)-1} e_i \exp \left(\left\langle \Theta_i^{(\theta)}, c^{(\theta)}(x-i, \xi_i) - c^{(\theta)}(x) \right\rangle \right) e_i^\top 2\theta_i^{(\theta)} \theta_i^{(\theta)-1} d\xi_i \\ &\quad + \frac{1}{2} \int_{-\infty}^{x_i} \theta_i^{(\theta)-1} E_i \exp \left(\left\langle \Theta_i^{(\theta)}, c^{(\theta)}(x-i, \xi_i) - c^{(\theta)}(x) \right\rangle \right) d\xi_i \\ &= \frac{1}{4} \theta_i^{(\theta)-1} e_i (2\theta_i^{(\theta)} x - \eta_i^{(\theta)})^\top \theta_i^{(\theta)-1}. \quad \blacksquare \end{aligned}$$

Proof of Proposition 11 We decompose the function $u_i(x, t)$, given by Theorem 9 into $u_i(x, t) = g_i(x, t) + \gamma_i(x, t)$ for all $i = 1, \dots, n$, where

$$\begin{aligned} g_i(x, t) &:= \frac{1}{2} \sum_{j=1}^n \frac{\partial}{\partial x_j} a_{ij}(x) + \frac{1}{2} \sum_{j=1}^n a_{ij}(x) \frac{\partial}{\partial x_j} p(x, \Theta_i^{(1)}, \dots, \Theta_i^{(k)}) \\ &\quad p(x, \Theta_i^{(1)}, \dots, \Theta_i^{(k)}) \\ \gamma_i(x, t) &:= -\frac{1}{2} \sum_{\ell=1}^k \nu_\ell p_\ell(x, \Theta_i^{(\ell)}) \left\langle \Theta_i^{(\ell)}, \mathcal{I}_i^{(\ell)}(x) \right\rangle. \end{aligned}$$

As a preliminary step by invoking Lemma 27

$$\begin{aligned} \left\langle \Theta_i^{(\ell)}, \mathcal{I}_i^{(\ell)}(x) \right\rangle &= \left\langle \eta_i^{(\ell)}, \mathcal{I}_{1,i}^{(\ell)}(x) \right\rangle + \left\langle \theta_i^{(\ell)}, \mathcal{I}_{2,i}^{(\ell)}(x) \right\rangle \\ &= \left\langle \eta_i^{(\ell)}, \frac{1}{2} \theta_i^{(\ell)-1} e_i \right\rangle + \left\langle \theta_i^{(\ell)}, \frac{1}{2} \theta_i^{(\ell)-1} e_i x^\top - \frac{1}{4} \theta_i^{(\ell)-1} e_i \eta_i^{(\ell)\top} \theta_i^{(\ell)-1} \right\rangle \\ &= \frac{1}{2} \eta_i^{(\ell)\top} (\theta_i^{(\ell)-1} e_i) + \frac{1}{2} \text{tr} \left(\theta_i^{(\ell)} (\theta_i^{(\ell)-1} e_i x^\top)^\top \right) - \frac{1}{4} \text{tr} \left(\theta_i^{(\ell)} (\theta_i^{(\ell)-1} e_i \eta_i^{(\ell)\top} \theta_i^{(\ell)-1})^\top \right) \\ &= \frac{1}{2} \eta_i^{(\ell)\top} \theta_i^{(\ell)-1} e_i + \frac{1}{2} e_i^\top \theta_i^{(\ell)-1} \dot{\theta}_i^{(\ell)} x - \frac{1}{4} e_i^\top \theta_i^{(\ell)-1} \dot{\theta}_i^{(\ell)} \theta_i^{(\ell)-1} \eta_i^{(\ell)}. \end{aligned}$$

Therefore,

$$\begin{aligned} \gamma_i(x, t) &= -\frac{1}{2} \frac{1}{p(x, \Theta_i^{(1)}, \dots, \Theta_i^{(k)})} \sum_{\ell=1}^k \nu_\ell p_\ell(x, \Theta_i^{(\ell)}) \\ &\quad \left(\frac{1}{2} \theta_i^{(\ell)-1} \eta_i^{(\ell)} + \frac{1}{2} \theta_i^{(\ell)-1} \dot{\theta}_i^{(\ell)} x - \frac{1}{4} \theta_i^{(\ell)-1} \dot{\theta}_i^{(\ell)} \theta_i^{(\ell)-1} \eta_i^{(\ell)} \right). \end{aligned}$$

Furthermore,

$$\begin{aligned} \frac{\partial}{\partial x_j} p(x, \Theta_i^{(1)}, \dots, \Theta_i^{(k)}) &= \sum_{\ell=1}^k \nu_\ell \left\langle \Theta_i^{(\ell)}, \frac{\partial c^{(\ell)}(x)}{\partial x_j} \right\rangle \exp \left(\left\langle \Theta_i^{(\ell)}, c^{(\ell)}(x) \right\rangle - \psi_\ell(\Theta_i^{(\ell)}) \right) \\ &= \sum_{\ell=1}^k \nu_\ell \left\langle \Theta_i^{(\ell)}, (e_j, e_j x^\top + x e_j^\top) \right\rangle \exp \left(\left\langle \Theta_i^{(\ell)}, c^{(\ell)}(x) \right\rangle - \psi_\ell(\Theta_i^{(\ell)}) \right) \\ &= \sum_{\ell=1}^k \nu_\ell \left(\eta_i^{(\ell)\top} e_j + 2e_j^\top \theta_i^{(\ell)} x \right) \exp \left(\left\langle \Theta_i^{(\ell)}, c^{(\ell)}(x) \right\rangle - \psi_\ell(\Theta_i^{(\ell)}) \right), \end{aligned}$$

and therefore

$$g(x, t) = \frac{1}{2} \operatorname{div}(a(x)) + \frac{1}{2} \frac{\sum_{\ell=1}^k \nu_{\ell} p_{\ell}(x, \Theta_t^{(\ell)}) a(x) \left(\eta_t^{(\ell)} + 2\theta_t^{(\ell)} x \right)}{p(x, \Theta_t^{(1)}, \dots, \Theta_t^{(k)})}.$$

Note that our choice of $\varphi_t^{(\ell)}$ satisfy (18) as shown in Lemma 26, which then completes the proof. ■

Appendix D. Proof of Theorem 13

Lemma 28 For an SDE of the form (15) the mean m_t and covariance matrix S_t of X_t satisfy

$$\begin{aligned} dm_t &= \mathbb{E}[u(X_t, t)] dt, \\ dS_t &= \left(\mathbb{E}[X_t u(X_t, t)^\top] + \mathbb{E}[u(X_t, t) X_t^\top] + \mathbb{E}[\sigma(X_t) \sigma(X_t)^\top] \right. \\ &\quad \left. - m_t \mathbb{E}[u(X_t, t)]^\top - \mathbb{E}[u(X_t, t)] m_t^\top \right) dt. \end{aligned} \quad (47)$$

Proof. The equation for the mean is trivial. For the variance let $Y_t := X_t X_t^\top$. According to Itô's Lemma Øksendal (2003) $dY_t = X_t(u(X_t, t) dt + \sigma(X_t) dB_t)^\top + (u(X_t, t) dt + \sigma(X_t) dB_t) X_t^\top + \sigma(X_t) \sigma(X_t)^\top dt$, and similarly $dm_t^2 = (m_t \mathbb{E}[u(X_t, t)]^\top + \mathbb{E}[u(X_t, t)] m_t^\top) dt$. Hence,

$$dS(t) = \mathbb{E}[dY(t)] - dm_t^2 = \left(\mathbb{E}[X_t u(X_t, t)^\top] + \mathbb{E}[u(X_t, t) X_t^\top] + \mathbb{E}[\sigma(X_t) \sigma(X_t)^\top] \right. \\ \left. - m_t \mathbb{E}[u(X_t, t)]^\top - \mathbb{E}[u(X_t, t)] m_t^\top \right) dt. \quad \blacksquare$$

Lemma 29 Mean m_t and variance S_t satisfy

$$m_u = \sum_{\ell=1}^k \nu_{\ell} m_t^{(\ell)}, \quad S_t = \sum_{\ell=1}^k \nu_{\ell} S_t^{(\ell)} + \sum_{\ell=1}^k \nu_{\ell} m_t^{(\ell)} m_t^{(\ell)\top} - \left(\sum_{\ell=1}^k \nu_{\ell} m_t^{(\ell)} \right) \left(\sum_{\ell=1}^k \nu_{\ell} m_t^{(\ell)} \right)^\top.$$

Proof. The statement for the mean is straightforward. For the variance,

$$\begin{aligned} S_t &= \sum_{\ell=1}^k \nu_{\ell} \mathbb{E}_{p_{\ell}}[X X^\top] - \left(\sum_{\ell=1}^k \nu_{\ell} m_t^{(\ell)} \right) \left(\sum_{\ell=1}^k \nu_{\ell} m_t^{(\ell)} \right)^\top \\ &= \sum_{\ell=1}^k \nu_{\ell} \left(\mathbb{E}_{p_{\ell}}[X X^\top] - m_t^{(\ell)} m_t^{(\ell)\top} \right) + \sum_{\ell=1}^k \nu_{\ell} m_t^{(\ell)} m_t^{(\ell)\top} - \left(\sum_{\ell=1}^k \nu_{\ell} m_t^{(\ell)} \right) \left(\sum_{\ell=1}^k \nu_{\ell} m_t^{(\ell)} \right)^\top. \quad \blacksquare \end{aligned}$$

Proof of Theorem 13 Consider a drift function $u(x, t)$ given by (20). In view of Lemma

28

$$\frac{dm_t}{dt} = \sum_{\ell=1}^k \nu_{\ell} \left(\mathbb{E}_{p_{\ell}} \left[\frac{1}{2} \operatorname{div}(a(X)) \right] + A_t^{(\ell)} + B_t^{(\ell)} m_t^{(\ell)} + \mathbb{E}_{p_{\ell}}[a(X)] C_t^{(\ell)} + \mathbb{E}_{p_{\ell}}[a(X) D_t^{(\ell)} X^\top] \right),$$

which can be simplified according to Lemma 29, such that

$$\begin{aligned} \sum_{\ell=1}^k \nu_{\ell} \frac{dm_t^{(\ell)}}{dt} &= \sum_{\ell=1}^k \nu_{\ell} \left(\mathbb{E}_{p_{\ell}} \left[\frac{1}{2} \operatorname{div}(a(X)) \right] + A_t^{(\ell)} \right. \\ &\quad \left. + B_t^{(\ell)} m_t^{(\ell)} + \mathbb{E}_{p_{\ell}}[a(X)] C_t^{(\ell)} + \mathbb{E}_{p_{\ell}}[a(X) D_t^{(\ell)} X^\top] \right). \end{aligned} \quad (48)$$

Note that (48) has to hold for all $\nu_{\ell} \geq 0$ such that $\sum_{\ell=1}^k \nu_{\ell} = 1$. Therefore

$$\frac{dm_t^{(\ell)}}{dt} = \mathbb{E}_{p_{\ell}} \left[\frac{1}{2} \operatorname{div}(a(X)) \right] + A_t^{(\ell)} + B_t^{(\ell)} m_t^{(\ell)} + \mathbb{E}_{p_{\ell}}[a(X)] C_t^{(\ell)} + \mathbb{E}_{p_{\ell}}[a(X) D_t^{(\ell)} X^\top]. \quad (49)$$

For the variance, we have according to Lemma 29

$$\frac{dS_t}{dt} = \sum_{\ell=1}^k \nu_{\ell} \left(\frac{dS_t^{(\ell)}}{dt} + \frac{dm_t^{(\ell)}}{dt} m_t^{(\ell)\top} + m_t^{(\ell)} \left(\frac{dm_t^{(\ell)}}{dt} \right)^\top - \frac{dm_t^{(\ell)}}{dt} m_t^\top - m_t \left(\frac{dm_t^{(\ell)}}{dt} \right)^\top \right).$$

This implies

$$\sum_{\ell=1}^k \nu_{\ell} \frac{dS_t^{(\ell)}}{dt} = \frac{dS}{dt} + \sum_{\ell=1}^k \nu_{\ell} \left(\frac{dm_t^{(\ell)}}{dt} (m_t^\top - m_t^{(\ell)\top}) + (m_t - m_t^{(\ell)}) \left(\frac{dm_t^{(\ell)}}{dt} \right)^\top \right),$$

where $\frac{dS_t}{dt}$ is given according to Lemma 28, by

$$\frac{dS_t}{dt} = \mathbb{E}[X_t u(X_t, t)^\top] + \mathbb{E}[u(X_t, t) X_t^\top] + \mathbb{E}[\sigma(X_t) \sigma(X_t)^\top] - m_t \left(\frac{dm_t}{dt} \right)^\top - \frac{dm_t}{dt} m_t^\top.$$

Therefore,

$$\begin{aligned} \sum_{\ell=1}^k \nu_{\ell} \frac{dS_t^{(\ell)}}{dt} &= \mathbb{E}[X_t u(X_t, t)^\top] + \mathbb{E}[u(X_t, t) X_t^\top] + \mathbb{E}[\sigma(X_t) \sigma(X_t)^\top] \\ &\quad - \sum_{\ell=1}^k \nu_{\ell} \left(\frac{dm_t^{(\ell)}}{dt} m_t^{(\ell)\top} + m_t^{(\ell)} \left(\frac{dm_t^{(\ell)}}{dt} \right)^\top \right). \end{aligned} \quad (50)$$

Recall that $\frac{dm_t^{(\ell)}}{dt}$ is given by (49). Next, we compute

$$\begin{aligned} \mathbb{E}[X u(X, t)^\top] &= \sum_{\ell=1}^k \nu_{\ell} \left(\mathbb{E}_{p_{\ell}} \left[\frac{1}{2} X \operatorname{div}(a(X))^\top \right] + m_t^{(\ell)} A_t^{(\ell)\top} + (m_t^{(\ell)} m_t^{(\ell)\top} + S_t^{(\ell)}) B_t^{(\ell)\top} \right. \\ &\quad \left. + \mathbb{E}_{p_{\ell}}[X C_t^{(\ell)\top} a(X)] + \mathbb{E}_{p_{\ell}}[X X^\top D_t^{(\ell)} a(X)] \right), \\ \mathbb{E}[u(X, t) X^\top] &= \sum_{\ell=1}^k \nu_{\ell} \left(\mathbb{E}_{p_{\ell}} \left[\frac{1}{2} \operatorname{div}(a(X)) X^\top \right] + A_t^{(\ell)} m_t^{(\ell)-1} + B_t^{(\ell)} (m_t^{(\ell)} m_t^{(\ell)\top} + S_t^{(\ell)}) \right. \\ &\quad \left. + \mathbb{E}_{p_{\ell}}[a(X) C_t^{(\ell)} X^\top] + \mathbb{E}_{p_{\ell}}[a(X) D_t^{(\ell)} X X^\top] \right), \end{aligned}$$

$$\mathbb{E}_p[\sigma(X)\sigma(X)^\top] = \mathbb{E}_p[a(X)] = \sum_{\ell=1}^k \nu_\ell \mathbb{E}_{p_\ell}[a(X)],$$

such that by evaluating (50) and by recalling that it has to hold for all convex combinations, we get the assertion (23). ■

References

Christophe Andrieu, Armand Doucet, and Roman Holenstein. Particle Markov chain Monte Carlo methods. *J. R. Stat. Soc. Series B*, 72(3):269–342, 2010. ISSN 1369-7412. URL <http://dx.doi.org/10.1111/j.1467-9868.2009.00736.x>.

Cédric Archambeau and Manfred Opper. Approximate inference for continuous-time Markov processes. In *Bayesian Time Series Models*, pages 125–140. Cambridge University Press, 2011. ISBN 9780521196765.

Cédric Archambeau, Dan Cornford, Manfred Opper, and John Shawe-Taylor. Gaussian process approximations of stochastic differential equations. In *Gaussian Processes in Practice*, volume 1 of *JMLR Proceedings*, pages 1–16, 2007.

Cédric Archambeau, Manfred Opper, Yian Shen, Dan Cornford, and John Shawe-Taylor. Variational inference for diffusion processes. In *Advances in Neural Information Processing Systems 20*, pages 17–24. MIT Press, Cambridge, MA, 2008.

Alan Bain and Dan Crisan. *Fundamentals of stochastic filtering*. Stochastic Modelling and Applied Probability. Springer, New York, 2009. ISBN 978-0-387-76895-3.

D. S. Bernstein. *Matrix Mathematics*. Princeton University Press, 2 edition, 2009.

Damiano Brigo. On SDEs with marginal laws evolving in finite-dimensional exponential families. *Statistics and Probability Letters*, 49(2):127–134, 2000.

Olivier Cappé, Eric Moulines, and Tobias Rydén. *Inference in hidden Markov models*. Springer Series in Statistics. Springer, New York, 2005. ISBN 978-0387-40264-2; 0-387-40264-0.

Francis Clarke. *Functional analysis, calculus of variations and optimal control*, volume 264 of *Graduate Texts in Mathematics*. Springer, London, 2013. ISBN 978-1-4471-4819-7; 978-1-4471-4820-3. URL <http://dx.doi.org/10.1007/978-1-4471-4820-3>.

John C. Cox, Jonathan E. Ingersoll, Jr., and Stephen A. Ross. A theory of the term structure of interest rates. *Econometrica*, 53(2):385–407, 1985. ISSN 0012-9682. URL <http://dx.doi.org/10.2307/1911242>.

D. Crisan and T. Lyons. A particle approximation of the solution of the Kushner-Stratonovich equation. *Probab. Theory Related Fields*, 115(4):549–578, 1999. ISSN 0178-8051. URL <http://dx.doi.org/10.1007/s004400050249>.

Botond Cséke, Manfred Opper, and Guido Sanginetti. Approximate inference in latent gaussian-markov models from continuous time observations. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 971–979. Curran Associates, Inc., 2013.

I. Csizsár. I-divergence geometry of probability distributions and minimization problems. *Ann. Probability*, 3:146–158, 1975.

Pierre Del Moral, Jean Jacod, and Philip Protter. The Monte-Carlo method for filtering with discrete-time observations. *Probab. Theory Related Fields*, 120(3):346–368, 2001. ISSN 0178-8051. URL <http://dx.doi.org/10.1007/PL000008786>.

A.V. Dmitruk and A.M. Kaganovich. The hybrid maximum principle is a consequence of Pontryagin maximum principle. *Systems and Control Letters*, 57(11):964 – 970, 2008. ISSN 0167-6911. URL <http://www.sciencedirect.com/science/article/pii/S016769110800100X>.

G. L. Eyink. A Variational Formulation of Optimal Nonlinear Estimation. *ArXiv Physics e-prints*, November 2000. URL <http://arxiv.org/abs/physics/0011049v2>.

Nancy Lopes Garcia and José Luis Palacios. On inverse moments of nonnegative random variables. *Statist. Probab. Lett.*, 53(3):235–239, 2001. ISSN 0167-7152. URL [http://dx.doi.org/10.1016/S0167-7152\(01\)00008-6](http://dx.doi.org/10.1016/S0167-7152(01)00008-6).

Yves Harel, Ron Meir, and Manfred Opper. A tractable approximation to optimal point process filtering: Application to neural encoding. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1603–1611. Curran Associates, Inc., 2015.

Olav Kallenberg. *Foundations of modern probability*. Probability and Its Applications (New York). Springer-Verlag, New York, second edition, 2002. ISBN 0-387-95313-2. URL <http://dx.doi.org/10.1007/978-1-4757-4015-8>.

H. J. Kushner. Dynamical equations for optimal nonlinear filtering. *J. Differential Equations*, 3:179–190, 1967. ISSN 0022-0396.

Harold J. Kushner and Paul Dupuis. *Numerical methods for stochastic control problems in continuous time*, volume 24 of *Applications of Mathematics (New York)*. Springer-Verlag, New York, second edition, 2001. ISBN 0-387-95139-3. URL <http://dx.doi.org/10.1007/978-1-4613-0007-6>. Stochastic Modelling and Applied Probability.

J. B. Lasserre. *Moments, Positive Polynomials and Their Applications*, volume 1 of *Imperial College Press Optimization Series*. Imperial College Press, London, 2010.

Jean B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001. URL <http://dx.doi.org/10.1137/S1052623400366802>.

- Sanjoy K. Mitter and Nigel J. Newton. A variational approach to nonlinear estimation. *SIAM J. Control Optim.*, 42(5):1813–1833, 2003. ISSN 0363-0129. URL <http://dx.doi.org/10.1137/S0363012901393894>.
- Bernt Øksendal. *Stochastic differential equations*. Universitext. Springer, 6. ed. edition, 2003. ISBN 3-540-04758-1.
- Gilles Pagès and Huyèn Pham. Optimal quantization methods for nonlinear filtering with discrete-time observations. *Bernoulli*, 11(5):893–932, 2005. ISSN 1350-7265. URL <http://dx.doi.org/10.3150/bj/1130077599>.
- E. Pardoux. Équations du filtrage non linéaire, de la prédiction et du lissage. *Stochastics*, 6(3-4):193–231, 1981/82. ISSN 0090-9491. URL <http://dx.doi.org/10.1080/17442508208833204>.
- F. J. Pinski, G. Simpson, A. M. Stuart, and H. Weber. Kullback-leibler approximation for probability measures on infinite dimensional spaces. *SIAM Journal on Mathematical Analysis*, 47(6):4091–4122, 2015. doi: 10.1137/140962802. URL <http://dx.doi.org/10.1137/140962802>.
- Albert N. Shiryaev. *Essentials of stochastic finance*, volume 3 of *Advanced Series on Statistical Science & Applied Probability*. World Scientific Publishing Co., Inc., River Edge, NJ, 1999. ISBN 981-02-3605-0. URL <http://dx.doi.org/10.1142/9789812385192>. Facts, models, theory, Translated from the Russian manuscript by N. Kruzhilin.
- J. Stoer and R. Bulirsch. *Introduction to numerical analysis*, volume 12 of *Texts in Applied Mathematics*. Springer-Verlag, New York, third edition, 2002. ISBN 0-387-95452-X. URL <http://dx.doi.org/10.1007/978-0-387-24738-3>.
- R. L. Stratonovich. Conditional Markov processes. *Theory of Probability & Its Applications*, 5(2):156–178, 1960. URL <http://dx.doi.org/10.1137/1105015>.
- Tobias Sutter, Arnab Ganguly, and Heinz Koepl. A variational approach to path estimation and parameter inference of hidden diffusion processes. *ArXiv e-prints*, August 2015.
- R. van Handel. *Filtering, Stability, and Robustness*. PhD thesis, Caltech, 2007.
- Michael D. Vrettas, Manfred Opper, and Dan Cornford. Variational mean-field algorithm for efficient inference in large systems of stochastic differential equations. *Phys. Rev. E*, 91:012148, Jan 2015. doi: 10.1103/PhysRevE.91.012148. URL <http://link.aps.org/doi/10.1103/PhysRevE.91.012148>.
- Darren James Wilkinson. *Stochastic modelling for systems biology*. Chapman & Hall/CRC Mathematical and Computational Biology Series. Chapman & Hall/CRC, Boca Raton, FL, 2006. ISBN 978-1-58488-540-5; 1-58488-540-8.
- Moshe Zakai. On the optimal filtering of diffusion processes. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 11(3):230–243, 1969. ISSN 0044-3719. URL <http://dx.doi.org/10.1007/BF00536382>.

One-class classification of point patterns of extremes

Stijn Luca

*KU Leuven - Technology Campus Geel
Department of Electrical Engineering
Kleinhofstraat 4, 2440, Geel, Belgium*

STIJN.LUCA@KULEUVEN.BE

David A. Clifton

*University of Oxford
Department of Engineering Science
Old Road Campus Research Building
Roosevelt Drive, Oxford, OX3 7DQ, UK*

DAVIDC@ROBOTS.OX.AC.UK

Bart Vanrumste

*KU Leuven - Technology Campus Geel
Department of Electrical Engineering
Kleinhofstraat 4, 2440, Geel, Belgium*

BART.VANRUMSTE@KULEUVEN.BE

Editor: Amos Storkey

Abstract

Novelty detection or one-class classification starts from a model describing some type of ‘normal behaviour’ and aims to classify deviations from this model as being either novelties or anomalies.

In this paper the problem of novelty detection for point patterns $S = \{\mathbf{x}_1, \dots, \mathbf{x}_k\} \subset \mathbb{R}^d$ is treated where examples of anomalies are very sparse, or even absent. The latter complicates the tuning of hyperparameters in models commonly used for novelty detection, such as one-class support vector machines and hidden Markov models.

To this end, the use of extreme value statistics is introduced to estimate explicitly a model for the abnormal class by means of extrapolation from a statistical model X for the normal class. We show how multiple types of information obtained from any available extreme instances of S can be combined to reduce the high false-alarm rate that is typically encountered when classes are strongly imbalanced, as often occurs in the one-class setting (whereby ‘abnormal’ data are often scarce).

The approach is illustrated using simulated data and then a real-life application is used as an exemplar, whereby accelerometry data from epileptic seizures are analysed - these are known to be extreme and rare with respect to normal accelerometer data.

Keywords: Sequence classification; novelty detection; extreme value theory; class imbalance; asymptotic theory

1. Introduction

Novelty detection is a particular example of pattern recognition that addresses the problem of identifying new patterns in data that are previously unseen. It shares many similarities with anomaly detection where one also wishes to detect abnormalities, but where in the latter these may not necessarily be entirely novel; i.e. a small amount of the training data

may contain outliers or anomalies. Novelty detection has a broad range of applications ranging from intrusion detection in computer related systems; industrial damage detection; to healthcare (Pimentel et al., 2014). All these applications have in common the fact that data describing failure conditions (or other abnormal behaviour) are rare or even absent, such that traditional classification methods may perform suboptimally. Novelty detection provides an alternative approach that starts from a model of normal behaviour and then detects deviations from this model (Bishop, 1994). It is for this reason that novelty detection is also termed one-class classification where there is no explicit model for ‘abnormal behaviour’. It may also be described in terms of a hypothesis test, in which the null-hypothesis is described by the model of normality.

This article considers one-class classification of ‘point patterns’, defined as sets of vectors $S = \{\mathbf{x}_1, \dots, \mathbf{x}_k\}$, $k \in \mathbb{N}_0$ located in data space \mathbb{R}^d where each \mathbf{x}_i is a realization of a random variable X^1 . We propose a statistical approach that starts from a probability density function (PDF) $y = p(\mathbf{x})$ associated with X that models the normal behaviour described by a dataset $D \subset \mathbb{R}^d$. Novelty detection then addresses the question of whether a set S of vectors is drawn from the distribution X or not.

In this article the use of the use of extreme value theory (EVT) is introduced to tackle classification of sets S (Embrechts et al., 1997). The Poisson point process (PPP) characterization of EVT is used to extract count data describing the number of times measurements in S fall in low-density regions defined by X . Furthermore, asymptotic results are provided in this article that allow us to unify this count information with the mean and maximal excess in $p(S)$ with respect to a low threshold e^{-u} . The method is evaluated using synthetic as well as real-world data, and is compared with commonly used algorithms for outlier detection such as one-class support vector machines (OCSVMs) and hidden Markov models (HMMs).

In contrast to existing novelty detection methods, EVT enables us to define a model for the abnormal class, where data are sparse or even unobserved. This enables us to circumvent the optimization of hyperparameters that is typically encountered in using one-class classifiers and which often requires data from the abnormal class. In essence, the use of EVT relies on extrapolation from the normal class, providing a class of models for low-density regions; the latter are particularly beneficial for novelty detection, because the decision boundary is expected to be situated in regions where data are sparse.

The remainder of this paper is organized as follows. Section 2 is devoted to related work on sequence classifications and provides an introduction to EVT. Subsequently, Section 3 introduces the EVT-based one-class classifier. In Section 4, the method is evaluated and its limitations are discussed.

2. Related work and EVT

This section starts with a short review of related work on sequence classification. The necessary background of EVT is then reviewed.

1. The common convention in statistics is used that applies capital letters to refer to population attributes and lower-case letters to refer to sample attributes.

2.1 Related work

The problem setting in this article is an example of a collective novelty detection problem where the individual instances within a set S are not classified with respect to the distribution X . Instead, the entire set S of vectors is considered to be one single instance that is assigned a single label. This contrasts with conventional one-class classification, in which every element of S is classified independently. Closely related to this problem is that of sequential learning. However in the latter each instance of the set S is given a different label. Widely-used machine learning techniques for sequential learning, such as HMMs and conditional random fields (CRFs), are not able to learn from one class only (Bishop, 2006; Sutton and McCallum, 2011). A commonly-used technique to tackle sequence classification is to concatenate the separate labels that are obtained by applying a one-class classifier (e.g., an OCSVM) to each instance \mathbf{x}_i separately. The mean novelty score of all instances, for example, can be used to decide whether or not S is novel (Dettendorf, 2002). This latter approach, however, is more naturally expressed by taking a point-wise approach where, from a statistical point of view, a number (k) of hypothesis tests are considered:

$$\begin{aligned} H_0 : \mathbf{x}_i \text{ is a realization of } X \\ H_1 : \mathbf{x}_i \text{ is novel with respect to } X, \end{aligned}$$

where H_0 denotes the so-called null-hypothesis and H_1 the alternative hypothesis. Due to the multiple hypothesis-testing problem, the number of false alarms can increase considerably for $k > 1$. Indeed, while each hypothesis test is chosen to have a small type-I error α (i.e., the probability of wrongly classifying \mathbf{x}_i as being novel, which is a false positive), the error of making at least one type-I error among the k hypothesis tests corresponds to $\bar{\alpha} = 1 - (1 - \alpha)^k$, e.g., when $\alpha = 5\%$ and $k = 6$, $\bar{\alpha} = 26\%$.

To obtain the correct decision boundary corresponding to the significance level α , Clifton et al. (2011) considered the univariate distribution over the probability density values $p(\mathbf{x})$ on the image $\text{Im}(p) = \{p(\mathbf{x}) \mid \mathbf{x} \in \mathcal{D}\}$ by reducing the multivariate analysis of the multidimensional data set \mathcal{D} to an univariate analysis on $\text{Im}(p)$. The PDF $y = p(\mathbf{x})$ can be obtained, for example, using a kernel density estimator (Scott, 1992). The distribution Y of these densities is strongly related to that of X , with a density defined by:

$$q(y) = \frac{dQ}{dy}(y) \quad \text{and} \quad Q(y) = \int_{p^{-1}(0,y)} p(\mathbf{x}) d\mathbf{x}. \quad (1)$$

As will be made clear in the following section, univariate EVT can then be used to describe sets $S = \{\mathbf{x}_1, \dots, \mathbf{x}_k\}$, which have a typical minimal density with respect to $y = p(\mathbf{x})$. In this way, a distribution is obtained for the most ‘extreme’ vectors that possibly occur in (truly ‘normal’) samples S drawn from X . A new set S is then evaluated by comparing its most extreme vector w.r.t. this model of extremes. Although this approach enables one to obtain a correct statistical type I-error α in testing S , its main drawback is that it captures limited information concerning the set S (Luca et al., 2014b). Indeed, only the single most extreme element in S is used to obtain a decision, while (non-extreme) information contained in the remaining part of the set is discarded. In this article we show how EVT can be used to include information contained in the remaining part of the pattern S while maintaining the correct statistical type I-error when testing S .

2.2 An introduction to EVT

EVT is a statistical discipline where the objective is to model the stochastic behavior of a univariate process at unusually large (or small) levels. It has already been used for many applications ranging from biomedical engineering, structural health monitoring, meteorology, and risk assessment in financial domains (Embrechts et al., 1997).

The central result in EVT is the Fisher-Tippett theorem concerning the limiting distribution of maxima of a sequence of independent and identically distributed (i.i.d.) random variables X_1, \dots, X_k according to a common distribution X :

$$M_k = \max\{X_1, \dots, X_k\},$$

as $k \rightarrow +\infty$. It states that when the following convergence in distribution appears:

$$P\left(\frac{M_k - c_k}{d_k} \leq x\right) \rightarrow G_\xi(x), \quad \text{as } k \rightarrow +\infty \quad (2)$$

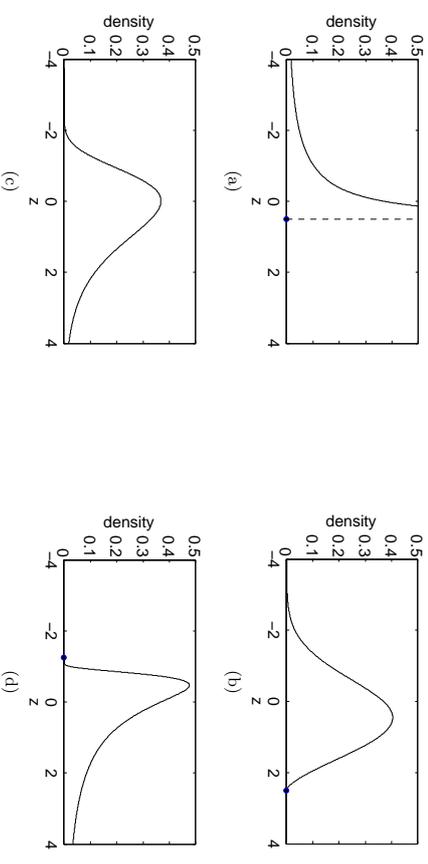


Figure 1: Different members of the GEV family in Eq. (3), with different values of the shape parameter ξ . The dot in the figures indicates the abscissa $z = -\frac{1}{\xi}$, where the density is zero, (a) $\xi = -2$ where we see that when $\xi \leq -1$ a short tail with an upper bound is described (b) $\xi = -0.4$ where we see that when $-1 < \xi < 0$ maxima have an upper bound (c) $\xi = 0$ where the maxima have no upper- or lower bound. Finally, (d) $\xi = 0.8$ where we see that for $\xi > 0$ the maxima have a lower bound.

for some normalizing constants c_k, d_k , the limiting distribution $G_\xi(x)$ is a member of the so-called family of *generalized extreme value (GEV) distributions*:

$$G_\xi(x) = \begin{cases} \exp\left\{-[1 + \xi x]^{-\frac{1}{\xi}}\right\}, & \xi \neq 0 \\ \exp\{-\exp(-x)\}, & \xi = 0. \end{cases} \quad (3)$$

For $\xi \neq 0$ the domain of the distribution is restricted to the set $\{x \mid 1 + \xi x > 0\}$. When the shape parameter ξ is negative, zero, or positive, the subset of members of the family correspond to the *Weibull*, *Gumbel* and *Fréchet* families respectively. The shape parameter thus determines the behaviour in the tail of the distribution of X , as shown in Figure 1.

The normalizing constants in (2) prevent a degenerate limit of the distribution of M_k , because clearly:

$$\lim_{k \rightarrow +\infty} P(M_k \leq x) = \lim_{k \rightarrow +\infty} \prod_{i=1}^k P(X_i \leq x)$$

which approaches zero for each $x < x_+$, where x_+ (possible $+\infty$) denotes the rightmost endpoint of the support of X .

The GEV family provides a model for block maxima, obtained by blocking (or windowing) the training data into blocks of equal length, and then fitting the GEV to the obtained

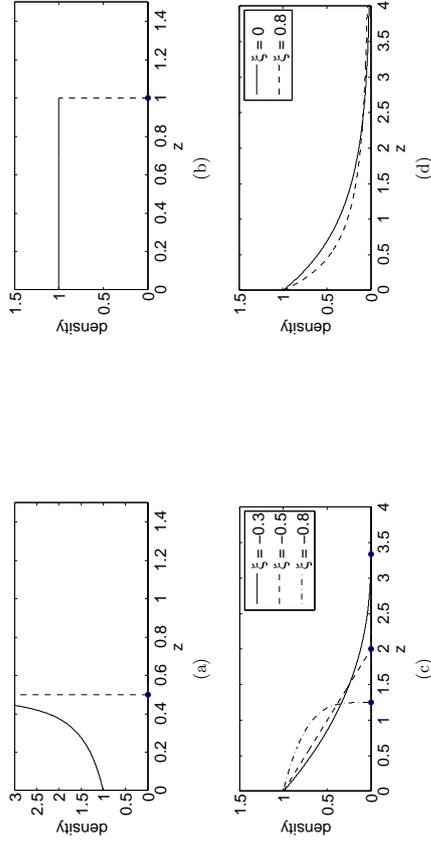


Figure 2: Different members of the GPD family in Eq. (3). The dot in the figures indicates the abscissa $z = -\frac{1}{\xi}$, where the density is zero, (a) $\xi = -2$, where $\xi < -1$, an asymptote occurs at $z = -\frac{1}{\xi}$. (b) $\xi = -1$ corresponds to an uniform distribution of excesses. (c) Different types of behaviour for $-1 < \xi < 0$ corresponding to excesses with an upper-bound. (d) For $\xi > 0$ the density has an intercept at $(0, 1)$.

set of block maxima. However, when these block are relatively large, this leads to using only a few block maxima, which can bias the estimation process. An alternative approach to overcome this problem is the so-called peaks over threshold (POT) method. In this approach, complete tails of a distribution X are modelled, defined as those measurements X_i of a sequence X_1, X_2, \dots that fall above some threshold u . A basic result of EVT states that when (2) holds for some member $G_\xi(x)$ of the GEV-family, the distribution of the exceedances $X - u$, conditional on $X > u$, satisfies the limiting property:

$$\lim_{u \uparrow x_+} P\left(\frac{X - u}{a(u)} < x \mid X > u\right) = H_\xi(x) \quad (4)$$

for some appropriate scaling factor $a(u)$ and

$$H_\xi = \begin{cases} 1 - (1 + \xi x)^{-1/\xi} & \text{if } \xi \neq 0 \\ 1 - e^{-x} & \text{if } \xi = 0 \end{cases} \quad (5)$$

denotes the family of generalized Pareto distributions (GPDs) where $x \geq 0$ for $\xi \geq 0$ and $0 \leq x \leq -\frac{1}{\xi}$ for $\xi \leq 0$, as shown Figure 2. For the Gumbel case $\xi = 0$, the scaling factor $a(u)$ is given by $E(X - u \mid X > u)$.

2.3 Poisson point processes and EVT

An elegant way to describe extremes, and one that unifies the block and POT approaches is based on Poisson point processes (PPPs). Any inference made using one of both above approaches could equally be made using the PPP model because it can be parametrized in terms of the GEV- and GPD- parameters. In this way, no extra computational effort is needed when using the PPP model.

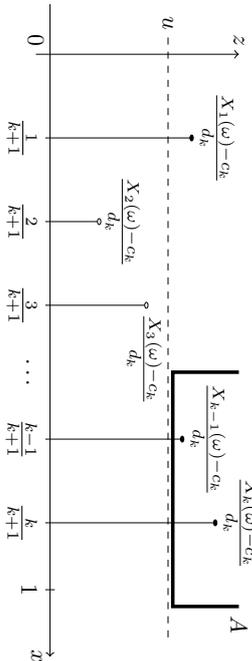
Generally a point process \mathcal{P} on a subset $U \subset \mathbb{R}^d$ is a stochastic model for which any one realization consists of a set of points $\{x_1, x_2, \dots, x_N\}$ that are randomly located in U and of which the number N is a random variable. The point processes closely related to EVT are the point processes of exceedances and consider those observations from sequences of random variables X_1, \dots, X_k which exceed a threshold u .

In particular, for a fixed choice of $k \in \mathbb{N}$, the point process of exceedances \mathcal{P}_k is defined on regions of the form $U =]0, 1[\times]u, +\infty[$ and considers those points that are situated in the intersection:

$$\mathcal{P}_k(\omega) = \left\{ \left(\frac{i}{k+1}, \frac{X_i(\omega) - c_k}{d_k} \mid 1 \leq i \leq k \right) \cap]0, 1[\times]u, +\infty[, \right. \quad (6)$$

where c_k and d_k are normalizing constants and ω denotes the stochastic event corresponding to a realization $\mathcal{P}_k(\omega)$ of the point process of exceedances. The indices are divided by the factor $k+1$ to rescale the process to the interval $]0, 1[$, as illustrated in Figure 3. The point processes \mathcal{P}_k can be characterised by *random counting measures*, which assign to each subset of the form $A =]t_1, t_2[\times]u + x, +\infty[\subset]0, 1[\times]u, +\infty[$ a random variable N_A describing the number of points of a realization that fall in region A :

$$N_A^k : \omega \mapsto \text{“number of points of } \mathcal{P}_k(\omega) \text{ in } A \text{”}$$

Figure 3: A realization $\mathcal{P}_k(\omega)$ of a point process of exceedances with $N_A^k(\omega) = 2$.

Indeed the values of these counting measures N_A^k for all subsets A give sufficient information to reconstruct completely those X_i that fall above a threshold of value $c_k + d_k u$. In fact, setting $A = \{\frac{k}{k+1}\} \times]z, +\infty[$, $N_A^k(\omega) > 0$ only applies when $X_i(\omega) > c_k + d_k z$.

The point process characterization of EVT is obtained by letting $k \rightarrow +\infty$. It is known (Embrechts et al., 1997) that when (2) holds for some normalization constants c_k and d_k , then the corresponding point processes of exceedances \mathcal{P}_k will converge to a PPP \mathcal{P} for $u > x_-$ where x_- denotes the leftmost endpoint of the support of the GEV-distribution in (2). This means that the following convergence of distributions holds:

$$N_A^k \xrightarrow{d} \text{Poi}[\Lambda(A)] \text{ as } k \rightarrow +\infty \quad (7)$$

on sets $A =]t_1, t_2[\times]u+x, +\infty[\subset U$ and where the distributions of N_A^k on non-overlapping sets A are mutually independent; i.e., the occurrence of a point at a location should not influence the probability of the occurrence of other points at other locations. In the limiting case, the rate parameter of the Poisson distribution $\Lambda(A)$ depends on the set A and is called the *intensity measure* $\Lambda(A)$ of the PPP. The fact that the PPP-characterization of extremes unifies the block and POT approach is due to the fact that the values of $\Lambda(A)$ in (7) can be written as a function of ξ (Embrechts et al., 1997):

$$\Lambda(A) = (t_2 - t_1) (1 + \xi(u+x))^{-1/\xi} = (t_2 - t_1) \lambda (1 + \xi \lambda^\xi x)^{-1/\xi} \quad (8)$$

with $\lambda = (1 + \xi u)^{-1/\xi}$. Therefore any inference made using the PPP limit of extremes yields immediately the shape parameter ξ in (2) and (21). In this way EVT describes three equivalent limiting properties (2), (4), and (7).

3. Learning from sparse data regions

In this article, a learning algorithm is proposed that explores the link between the three representations of extremes as introduced in the previous section. For this purpose so-called EVT-based features will be introduced in section 3.1 that describe characterizing measures of a set $S = \{\mathbf{x}_1, \dots, \mathbf{x}_k\}$ of vectors independently drawn from a distribution X . In Section 3.2, a joint asymptotic distribution of these features is calculated as $k \rightarrow +\infty$. Subsequently,

analytical expressions of cumulative scores with respect to this distribution are obtained that will be used as novelty scores to evaluate the novelty of S with respect to X for large k .

3.1 EVT-based features

Consider a d -dimensional random variable X with PDF $g = p(\mathbf{x})$. The transformation $Z = -\log p(X)$ allows us to study multivariate low-density regions $\{\mathbf{x} \mid p(\mathbf{x}) < e^{-u}\}$ with u some large real number, as a convex univariate region $\{z \mid z > u\}$. Associated with a sequence of i.i.d. random variables X_1, \dots, X_k , we define the following associated features based on the log-transformed sequence Z_1, \dots, Z_k , $Z_i = -\log p(X_i)$:

1. The number of exceedances among Z_1, \dots, Z_k above some threshold u_k :

$$N_k = \sum_{i=1}^k \mathbb{I}_{\{Z_i > u_k\}},$$

where $\mathbb{I}_{\{Z_i > u_k\}}$ denotes an indicator function taking the value 1 when $Z_i > u_k$ and zero otherwise. This feature describes the number of multivariate points from a sequence $\{X_1, \dots, X_k\}$ that are situated in a low density region $\mathcal{R}_k = \{\mathbf{x} \mid p(\mathbf{x}) < e^{-u_k}\}$.

2. The mean exceedance among Z_1, \dots, Z_k above some threshold u_k :

$$V_k = \frac{1}{N_k} \sum_{i=1}^k (Z_i - u_k) \mathbb{I}_{\{Z_i > u_k\}}$$

A high value of V_k indicates that, on average, the points of the sequence X_1, \dots, X_k are outlying with respect to the locus of the training data while a low value indicates that the sequence is situated near the locus of the training data.

3. The maximal exceedance among Z_1, \dots, Z_k above some threshold u_k :

$$M_k = \max_{1 \leq i \leq k} \{Z_i - u_k \mid Z_i > u_k\}$$

corresponding to the most outlying point of X_1, \dots, X_k with respect to the training data.

Note that the mean exceedance V_k and the maximal exceedance M_k are only well-defined when $N_k \geq 1$. The features above provide a natural way to summarize the extent to which densities of observations falling in low-density regions exceed some low threshold e^{-u_k} . Therefore when a set $S = \{\mathbf{x}_1, \dots, \mathbf{x}_k\}$ of k observations is novel with respect to the distribution X , it is expected that the corresponding features v_k , m_k , and ms_k of the sample S have a higher cumulative score given their respective distributions V_k , M_k , and N_k . Hence these features allow us to summarize the information contained in the tail of a d -dimensional distribution X (that can be of arbitrarily high dimension) in a 3-dimensional distribution. To determine the joint distribution of these EVT-based features, the PPP characterization (7) is applied to the univariate random variable Z whose tail describes the

multivariate points X that are lying in low-density regions. In the next section we will determine the joint distribution of these 3 features to fuse the information from each.

To apply the PPP characterization to Z , we consider the sequence of point processes \mathcal{P}_k on \mathbb{R}^2 associated with $Z = -\log p(X)$:

$$\mathcal{P}_k = \left\{ \left(\frac{i}{k+1}, Z_i \right) \mid 1 \leq i \leq k \right\}.$$

From the limiting property (7), the point processes \mathcal{P}_k will converge to a PPP as $k \rightarrow +\infty$ on regions of the form $[0, 1[\times]u_k, +\infty)$, with $u_k = c_k + ud_k$, $u \in \mathbb{R}$, and with c_k, d_k being the normalizing constants as in (6). Block maxima of Z_i are not bounded from above or below, and so the Gumbel distribution is the only possible limiting EVT distribution for this one-class formulation; i.e., $\xi = 0$ in the limiting property (7). For the Gumbel case it is known that the normalizing constants can be chosen as (Embrechts et al., 1997)²:

$$c_k = \inf \left\{ z \mid P(Z \leq z) \geq 1 - \frac{1}{k} \right\} \quad \text{and} \quad d_k = E(Z - c_k \mid X > c_k). \quad (9)$$

The intensity measure of the limiting PPP can be obtained by letting $\xi \rightarrow 0$ in (8):

$$\Lambda(A) = (t_2 - t_1)e^{-(x+u)} = (t_2 - t_1)\lambda e^{-x}, \quad \text{with } \lambda = e^{-u} \quad (10)$$

and where the parameter λ is given by the expected number of exceedances of Z above $u_k(x) = c_k + (u+x)d_k$. We can now state the following theorem that is proved in Appendix A.1 and that characterizes the distribution of the EVT features defined above.

Theorem 1 Consider the random variables N_k, V_k and M_k associated with sets S of k observations $\{X_1, \dots, X_k\}$ drawn from a d -dimensional random variable X . Denote $y = p(\mathbf{x})$ the PDF of X and suppose $Z = -\log p(X)$ satisfies the following limiting property:

$$\lim_{w \rightarrow +\infty} P \left(\frac{Z-w}{a(w)} > x \mid Z > w \right) = e^{-x}, \quad \forall x \in \mathbb{R}^+ \quad (11)$$

where $a(w) = E(X-w \mid X > w)$. Denoting, for $u \geq 0$, the following sequence of thresholds:

$$u_k = c_k + ud_k, \quad \text{with } c_k = \inf \left\{ z \mid P(Z \leq z) \geq 1 - \frac{1}{k} \right\}, \quad d_k = a(c_k),$$

the following limiting properties hold as $k \rightarrow +\infty$:

- (i) The distribution N_k of the number of observations among k of X that fall in regions $\{\mathbf{x} \mid p(\mathbf{x}) \leq e^{-u_k}\}$ converges to a Poisson distribution with a rate $\lambda = e^{-u}$:

$$\lim_{k \rightarrow +\infty} P(N_k = n) = \frac{\lambda^n e^{-\lambda}}{n!} \quad (12)$$

2. The operator \inf in (9) refers to the infimum or greatest lower bound.

- (ii) After normalization, the distribution of the maximal exceedance M_k above threshold u_k converge in distribution to a Gumbel member of the GEV family with $\mu = \log \lambda$ that is conditioned on the positive real line; i.e.,

$$\lim_{k \rightarrow +\infty} P \left(\frac{M_k - \mu}{d_k} \leq m \mid N_k \geq 1 \right) = \frac{\exp \left\{ -\exp \left[-(m - \log \lambda) \right] \right\}}{1 - e^{-\lambda}} \quad (13)$$

- (iii) After normalization, the mean exceedance V_k above u_k converges in distribution to a random variable distributed according to a cumulative distribution function:

$$\lim_{k \rightarrow +\infty} P \left(\frac{V_k - \mu}{d_k} \leq v \mid N_k \geq 1 \right) = 1 - \frac{1}{e^{\lambda v - 1}} \left(\sum_{j=0}^{+\infty} \frac{\lambda^j}{j!} (v)^j e^{-lv} \right) \quad (14)$$

Figure 4 illustrates the limiting properties obtained in Theorem 1 based on a two-dimensional distribution X given by a Gaussian mixture model (GMM) of two standard normal distributions centred at the origin and (1, 1) respectively. The constants c_k and d_k were estimated by an empirical estimation of (9) based on a simulated sample of length 5×10^6 from the mixture. Setting $u = 0$, the empirical distributions of N_k, M_k and V_k were estimated based on 5×10^3 sets of lengths $k \in \{5, 20, 50\}$ and compared with the analytical expression obtained in Theorem 1. The figure shows that the distributions are approximating the limiting case more closely as k increases, while for $k \geq 20$ this approximation may already be seen to be satisfactory.

3.2 EVT-based one-class classifier

A joint distribution is here calculated to fuse the information from the EVT-based features M_k, N_k , and V_k , as introduced in Section 3.1. For this purpose, we suppose that at least one exceedance of $-\log p(X_i)$ above u_k is observed in a sequence $S = \{X_1, \dots, X_k\}$ of length $|S| = k$. The proof of the following theorem can be found in Appendix A.2.

Theorem 2 Consider the random variables N_k, V_k , and M_k associated with sets S of k observations $\{X_1, \dots, X_k\}$ drawn from a d -dimensional random variable X . Denote $y = p(\mathbf{x})$ the density function of X and suppose $Z = -\log p(X)$ satisfies the following limiting property:

$$\lim_{w \rightarrow +\infty} P \left(\frac{Z-w}{a(w)} > x \mid Z > w \right) = e^{-x}, \quad \forall x \in \mathbb{R}^+ \quad (15)$$

where $a(w) = E(Z-w \mid Z > w)$. After normalization, the joint cumulative distribution function of (N_k, V_k, M_k) conditioned on $N_k \geq 1$ and related to the sequence of thresholds u_k as in Theorem 1:

$$F_k(v, m, n) = P \left(\frac{V_k - \mu}{d_k} \leq v, \frac{M_k - \mu}{d_k} \leq m, N_k \leq n \mid N_k \geq 1 \right), \quad (16)$$

converges on $D = \{(v, m, n) \mid \frac{m}{n} \leq v \leq m\}$ to a mixture of translated chi-squared distribution as k tends to infinity:

$$F(v, m, n) = \lim_{k \rightarrow +\infty} F_k(v, m, n) = \sum_{i=1}^n \frac{\lambda^i e^{-\lambda}}{i!(1 - e^{-\lambda})} \sum_{l=0}^r (-1)^l \binom{r}{l} e^{-im} \chi_{2l} (2(lv - im)) \quad (17)$$

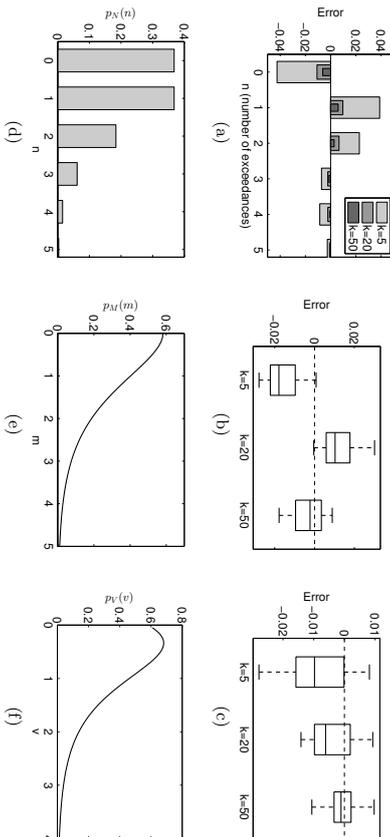


Figure 4: Comparison between limiting distribution as $k \rightarrow +\infty$ and empirical distribution functions for $k \in \{5, 20, 50\}$ using simulated data from a GMM when $u = 0$. (a) - (c) Differences between empirical distribution and asymptotic distribution for N_k , M_k , and V_k respectively; (d) - (f) Limiting PDFs p_N , p_M , and p_V from Eqs. (12) - (14) as $k \rightarrow +\infty$ for N_k , M_k , and V_k respectively.

where $r = \lfloor \frac{ln}{m} \rfloor$ (i.e. $\frac{ln}{m} \in [r, r+1[$, for $0 \leq r \leq l-1$), χ_p denotes the cumulative chi-squared distribution function with p degrees of freedom and $\lambda = e^{-u}$ is the exceedance rate of the limiting Poisson distribution of N_k as in Theorem 1-(i).

Note that the term in (17) for $l = 1$ has the identity line $m = v$ as its domain and the expression reduces to $\frac{\lambda e^{-\lambda}}{1 - e^{-\lambda}}(1 - e^{-m})$. The corresponding limiting joint density function of (N_k, V_k, M_k) on D can be found by partial derivation of formula (17):

$$f(v, m, n \mid n \geq 1) = \begin{cases} e^{-nv} \sum_{i=1}^{\lfloor \frac{mn}{m} \rfloor} c_{in} (nv - im)^{n-2}, & n \geq 2 \\ \frac{\lambda}{e^{-\lambda} - 1} e^{-m} \mathbb{I}_{i=m}, & n = 1 \end{cases} \quad (18)$$

where c_{in} are constants defined for $1 \leq i \leq n$ as:

$$c_{in} = -\frac{n\lambda^n}{(e^\lambda - 1)\Gamma(n)\Gamma(n-1)} (-1)^i \binom{n-1}{i-1}.$$

and where $\mathbb{I}_{i=m}(v, m)$ is an indicator function taking the value 1 when $v = m$, and which is zero elsewhere.

To apply Theorem 2, note that (15) implies that an exponential approximation of the exceedances is valid from some high threshold u_0 :

$$P(Z - u_0 > x \mid Z > u_0) \approx e^{-\frac{x}{\sigma}} \quad (19)$$

with $\sigma = \alpha(u_0) = E(Z - u_0 \mid Z > u_0)$ and $\sigma \approx d_k$. Then, based on Theorems 1 and 2, a novelty score of a sequence S with corresponding EVT features (v_S, m_S, n_S) can be defined:

$$XS = \begin{cases} P(N_k < n_S) + P(V_k \leq v_S, M_k \leq m_S, N_k = n) & \text{when } n_S > 0 \\ P(N_k = 0) & \text{when } n_S = 0 \end{cases}$$

and for large k this is approximated by:

$$XS \approx \begin{cases} \left(\sum_{i=0}^{n_S-1} \lambda^i e^{-\lambda} \right) + F\left(\frac{v_S}{\sigma}, \frac{m_S}{\sigma}, n_S\right) - F\left(\frac{v_S}{\sigma}, \frac{m_S}{\sigma}, n_S - 1\right) & \text{when } n_S > 0 \\ e^{-\lambda} & \text{when } n_S = 0 \end{cases} \quad (20)$$

These novelty scores quantify the ‘extremity’ of a sequence S by cumulatively summing the probability of having fewer than n_S exceedances, while the mean and maximal exceedances with respect to the threshold u_0 do not exceed v_S and m_S respectively. There is a valid probabilistic interpretation to XS making it a risk metric that quantifies the risk that S is novel; i.e., that S has some distribution other than X .

The choice of u_0 in the approximation (19) can be assessed by means of a *mean excess plot* which is a graphic diagnostic in which the sample means of the excesses $(Z - u)$ are plotted against a range of thresholds along with the confidence intervals (Embrechts et al., 1997). The threshold is chosen to be the lowest level where all the higher threshold-based sample mean excesses are consistent with a horizontal line. Alternatively an empirically driven rule-of-thumb can be chosen that specifies the tail fraction which satisfies the approximation in (19) and where u_0 is estimated as the quantile at $1 - \frac{n}{n \log \log(n)}$ of a sample of length n of the distribution (Scarrot and MacDonald, 2012). The parameters σ and λ can then be estimated by means of maximum likelihood estimation (Falk et al., 2011).

Figure 5(a)-(b) illustrates the limiting joint PDF (18) on the domain D conditioned on the number of exceedances for $n = 3$ and $n = 5$ for a GMM X of two standard normal distribution centred at $(0, 0)$ and $(1, 1)$. As the number of exceedances increases, the mode of the distributions moves diagonally upwards. Figure 5(c) shows a probability-probability (P-P) plot assessing the limiting property (17) for $k = 20$. For this purpose a sample of 5×10^3 sets of length $k = 20$ were simulated from X to estimate the cumulative probabilities $F_k(v, m, n)$ empirically, on a grid of $(v, m, n) \in [0, 10] \times [0, 10] \times \{2, 3, 5\}$ consisting of 300 vertices and compare these estimations with $F(v, m, n)$.

4. Experiments

In this section, the validity of our proposed method is illustrated using both artificial and real-world data sets. The novel EVT algorithm is compared with the conventional sequence classifiers HMMs and OCSVMs. To this end, 5-fold cross-validation is performed where in each run a random subset of the data from the normal class is used for training and the remainder of the data is split evenly between validation and test data. The randomized runs are kept the same across the different classifiers to allow a consistent comparison. The novelty score of a sequence with respect to a HMM or OCSVM is calculated as being the mean of the likelihoods assigned by the model to each individual instance of the sequence.

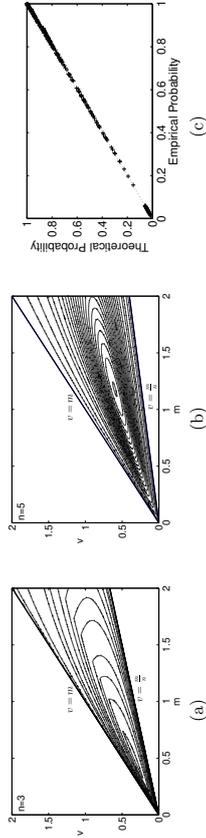


Figure 5: (a) - (b) Limiting joint PDF (18) on the domain D conditioned on $n = 3$ and $n = 5$ respectively, for a GMM X consisting of two standard normal distribution centred at $(0, 0)$ and $(1, 1)$. (b) A probability-probability (P-P)-plot comparing the joint empirical cumulative distribution of (V_k, M_k, N_k) for $k = 20$ with the limiting joint distribution.

Both HMMs and OCSVMs depend on hyperparameters, the value of which are estimated using the validation sets by maximizing a cost-function. For the HMM, the number of states varies from 1 – 4 (Rabiner and Murray, 1989), while for the OCSVM the standard hyperparameters (σ, ν) are optimized that respectively denote the kernel width of the Gaussian kernel that is used and an upper bound on the fraction of outliers (Schölkopf et al., 2001). The threshold on the novelty scores is optimized using the validation data.

For the EVT model, no validation step is performed and no data from the abnormal class are considered during training. A threshold of 95% is chosen on the novelty score (motivated from a probabilistic viewpoint). The density of the distribution X describing the normal class is estimated using a kernel density estimation with Gaussian kernels, and where the kernel width is estimated by minimization of the mean integrated squared error (Scott, 1992).

4.1 Synthetic data set

In order to validate the use of our EVT-based method a simulated dataset is constructed where data from the abnormal class are situated in the tail regions of a planar Gaussian mixture X consisting of two components located at $(-2, -2)$ and $(0, 0)$ respectively with covariance matrix $\frac{1}{2}I_2$, with I_2 the identity matrix in $\mathbb{R}^{2 \times 2}$. The training data of the normal class consisted of 100 sets of length $k = 20$ points drawn from X . Several experiments were performed where the proportion of abnormal instances in the validation and test sets varied in the range $p_a \in \{0.01, 0.05, 0.1, 0.5\}$. The abnormal class of patterns contained a mixture of normal instances from X and abnormal instances coming from the tail region where the density $p(\mathbf{x}) \leq 5 \times 10^{-4}$. In a 5-fold cross-validation experiment, the ability of the detection of these patterns between an OCSVM, a HMM, and our EVT model is compared.

Figure 6(a) shows the contours of the tail region obtained from applying the Gumbel model of M_k on the densities that are estimated using a kernel density estimation of X . The dark contour surrounding the central region indicates the tail region estimated by the Gumbel model. In this region, the dark contour corresponds to an empirical estimation of

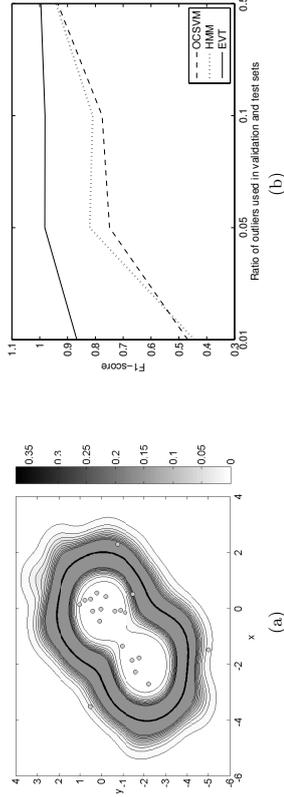


Figure 6: (a) The estimation of the tails of a Gaussian mixture X using a Gumbel model on the distribution of densities (1). The bold contour indicates the estimation of the EVT threshold e^{-u_k} for $k = 20$ on the likelihoods as defined in Theorem 1. (b) F1-scores averaged over the runs of a 5-fold-validation experiment across different ratios of available abnormalities.

the threshold $u_k = c_k + \nu d_k$ where u is set to zero (see Theorem 1). It is with respect to this threshold that the number of exceedances N_k and the maximal and mean exceedance M_k and V_k are calculated. Using our EVT-based method, an abnormal sequence can be evaluated as a cumulative probability score (20) with respect to the joint distribution of the EVT-based features. For example, the sequence of gray points shown in Figure 6 contains three exceedances with respect to the threshold u_k and has a score $X_S = 98.97\%$ such that it is classified as being novel with respect to X . Figure 6(b) shows the F1-scores of the classifiers, averaged over the 5 folds in our cross-validation experiment. When the ratio of abnormal patterns in the training phase is 50% the classifiers perform equally well. EVT, however, is able to outperform the classifiers when data from the abnormal class become sparse, as is typically the case for novelty detection problems. When there is a lack of examples from the abnormal class, the optimization of the hyperparameters and the novelty threshold in a HMM and an OCSVM is suboptimal. EVT, on the other hand, provides a class of models for the tail region where training data are sparse and is able to estimate the threshold exactly by using a statistical distribution that is obtained by extrapolation from the normal class (where data are usually abundant).

4.2 Accelerometer data for the detection of epileptic seizures

In this section, a case study in the healthcare domain is considered using a set of acceleration data collected from movements of patients suffering from epilepsy (Cuppens et al., 2013). The acceleration data were recorded during several nights using four 3D acceleration sensors attached to the extremities of 7 children with hypermotor seizures, all between the age of 5 and 16 years. Hypermotor seizures are epileptic convulsions that are marked by a strong and uncontrolled movement of the arms and legs that can last from a couple of seconds to a number of minutes. Due to the exaggerated movement involved, the patient can injure

themselves during the seizure, which increases the need for an alarm system with high sensitivity to abnormality.

In a pre-processing phase, movement events E_s are extracted from the data set using an energy-based threshold. We denote the acceleration vectors in these events as

$$E_s = \{a_{t,l} | 1 \leq t \leq T, 1 \leq l \leq 4\}$$

where the indices refer to the time index and the limb respectively (1 = left arm, 2 = right arm, 3 = left leg, 4 = right leg). Cuppens et al. (2013) performed a feature analysis where 3 features were identified as being relevant to this application:

- i) Movement length, $f_1 = |E_s| = T$
- ii) Average energy in a movement:

$$f_2 = \frac{1}{T} \sum_{t,l} \|a_{t,l}\|^2$$

- iii) The maximal energy in an arm movement:

$$f_3 = \max_{1 \leq l \leq T} \left\{ \|a_{t,1}\|^2, \|a_{t,2}\|^2 \right\}$$

The features are calculated within sliding windows containing 125 samples (Luca et al., 2014a) which are randomly subsampled to obtain sets $S = \{\mathbf{x}_1, \dots, \mathbf{x}_k\}$ of fixed length $k = 20$ containing data instances $\mathbf{x}_i = (f_1, f_2, f_3) \in \mathbb{R}^3$ on which the EVT algorithm for sequence classification can be applied.

The data are highly unbalanced as may be seen in Table 1. Only three patient recordings contain more than 3 examples of seizures. For these patients, an OCSVM and HMM were trained in a 5-fold cross-validation experiment where in each fold the seizures are randomly split between validation and test sets to optimize the following cost-function (Cuppens et al., 2013):

$$C(\boldsymbol{\lambda}) = 2 \cdot SS(\boldsymbol{\lambda}) + PPV(\boldsymbol{\lambda})$$

with respect to the hyper-parameters $\boldsymbol{\lambda}$ of the model. Here, the weight of the sensitivity (SS) is higher than the weight of the positive predictive value (PPV), because missing a

Table 1: Overview of epileptic accelerometry data set.

Patient number	Nights of monitoring	Hypermotor seizures	Normal movements
pat 1	1	2	117
pat 2	2	2	287
pat 3	2	2	439
pat 4	1	2	239
pat 5	5	26	784
pat 6	2	7	381
pat 7	2	3	468
<i>total</i>	<i>15</i>	<i>51</i>	<i>2715</i>

Table 2: SS and PPV scores of different approaches used in the detection of epileptic seizures (a) OCSVM, (b) HMM, and (c) EVT. Mean and standard deviations (SD) are calculated over the folds in a 5-fold-cross-validation experiment.

	SS		PPV		F1	
	mean	SD	mean	SD	mean	SD
OCSVM						
pat2	100.0	0.00	48.03	13.19	64.07	11.62
pat5	64.62	18.53	34.08	2.68	43.59	2.22
pat6	100.00	0.00	31.85	7.20	47.96	8.14
			(a)			
HMM						
pat2	70.00	20.92	89.33	15.35	76.83	14.09
pat5	56.92	8.77	46.57	16.75	49.71	10.40
pat6	80.00	29.81	85.00	13.69	77.43	15.53
			(b)			
EVT						
pat2	100.0	0.00	69.65	21.38	80.63	14.75
pat5	35.38	10.32	21.80	2.73	26.80	4.95
pat6	100.0	0.00	48.21	15.15	64.05	12.34
pat1	100.0	0.00	19.68	9.55	32.05	13.08
pat3	100.0	0.00	56.67	25.28	70.00	18.26
pat4	100.0	0.00	48.33	30.28	61.33	23.64
pat7	100.0	0.00	66.67	31.18	76.67	22.36
			(c)			

seizure is more costly than generating a false-positive classification for this type of seizure. Tables 2(a) and 2(b) show the mean performance scores calculated over the different test sets in the runs for three patients of which more than 3 examples of seizures were available for the training of these models. As there are at most 3 seizures present for the remaining patients, at most two seizures could be used in the validation set when training the HMMs and OCSVMs. In this way at most one of the seizures could be held out and detected by the algorithms during the different cross-validation experiments.

Table 2(c) shows performance scores related to the EVT approach. In contrast to the OCSVM and HMM, performance scores could easily be obtained for all patients without the need for optimization using validation data. As hypermotor seizures are marked by strong and uncontrolled movements, the use of EVT is very suitable in this application to recognize this type of ‘extremity’ from the class of normal movement events. In contrast to an OCSVM our EVT-based method was able to improve PPV values in patients 2 and 6 (averaged over the folds, a decrease of 3 false alarms while testing 50 normal movements was obtained) while the SS scores remained 100%. The OCSVM was able to outperform the EVT method for patient 5. This is mainly due to (i) the seizures for this patient are less extreme than in the rest of the data (Cuppens et al., 2013); (ii) a sufficient amount of seizures is present giving the OCSVM the ability to perform a thorough optimization of the hyperparameters during the training phase. A HMM was not able to detect all seizures and obtained better PPV values compared to our EVT-based method.

5. Conclusion

This article focuses on the problem of novelty detection, where data instances from the normal class are abundant but where examples from the abnormal class are sparse. In particular a new approach is introduced that is based on the use of EVT and which is particularly well-suited to detecting outliers that present ‘extreme’ behaviour with respect to a statistical model X . It is shown how EVT can be adapted to define a model over

regions where data are sparse (or even unavailable) circumventing the need for optimization of hyperparameters as otherwise occurs when using conventional OCSVMs or HMMs. This leads to a more robust and exact estimation of the support of X when abnormal data are limited in availability.

One of the main challenges in novelty detection is to improve the PPV. Indeed, when classes are highly unbalanced, an unusually high accuracy is required to overcome a high false-alarm rate. Therefore rich models that combine several types of information in a natural way are needed to increase the PPV of a novelty detector. An estimation procedure from EVT is proposed that encodes the three different types of EVT-based information for a sequence S . Given a threshold u and an estimation $y = \hat{p}(\mathbf{x})$ of the density of X , the following types of information were fused: (i) the maximal exceedance of $-\log p(S)$ above u ; (ii) the mean exceedance of $-\log p(S)$ above u ; and (iii) the number of exceedances of $-\log p(S)$ above u .

We have demonstrated the use of this method on both artificial data and a real-world set of acceleration data collected from movements of patients that suffer from epilepsy. By applying the proposed method, it was shown that SS scores and PPV scores could be improved compared to the use of conventional HMMs and OCSVMs, especially when examples from the abnormal class are sparse.

Acknowledgments

Special thanks go to Peter Karsmakers for the fruitful discussions concerning the validation steps and preprocessing study of the epileptic seizure data. This data set is collected in collaboration with the Pulderbos rehabilitation Center for Children and Youth in Zandhoven (Pulderbos), Belgium and the assistance of Bertien Ceulemans, Lieven Lagae, Anouk Van de Vel and Sabine Van Huffel in the framework of an IWT TBM project 100404. The authors would also like to acknowledge networking support by the ICT COST action IC1303 (AAPLE). David A. Clifton is funded by the Royal Academy of Engineering and an EPSRC Healthcare Technologies Challenge Award.

Appendix A. Proofs

In this appendix we prove the results obtained in Section 3.

A.1 Proof of Theorem 1

Proof In terms of the normalized sequence of random variables $\frac{Z-c_i}{d_i}$, it can be shown that (11) is equivalent to:

$$\lim_{i \rightarrow +\infty} P \left(\frac{Z - c_i}{d_i} < u + x \mid \frac{Z - c_i}{d_i} > u \right) = 1 - e^{-x} \quad (21)$$

with $u \in \mathbb{R}$ and $x \geq 0$ (Falk et al., 2011, p.21). The statements (i)-(iii) can now be proven as follows.

(i) This result follows by applying the link between the limiting properties (2), (4) and (7)

on the transformed variable $Z = -\log(p(X))$ as discussed in Sections 2.2 and 2.3. The exceedance rate of the PPP can be found by calculating the limit:

$$\begin{aligned} \lim_{k \rightarrow +\infty} kP(Z \geq c_k + d_k u) &= \lim_{k \rightarrow +\infty} \frac{P(Z \geq c_k + d_k u)}{P(Z > c_k)}, \quad \text{as } P(Z \leq c_k) = 1 - \frac{1}{k} \\ &= \lim_{k \rightarrow +\infty} P(Z \geq c_k + d_k u | Z > c_k) \\ &= \lim_{k \rightarrow +\infty} P \left(\frac{Z - c_k}{a(c_k)} \geq u | Z > c_k \right), \quad \text{as } d_k = a(c_k) \\ &= \lim_{w \rightarrow +\infty} P \left(\frac{Z - w}{a(w)} \geq u | Z > w \right), \quad \text{as } \lim_{k \rightarrow +\infty} c_k = +\infty \\ &= e^{-u} \end{aligned}$$

(ii) The limiting distribution of the maximal exceedance M_k conditioned on the number of exceedances $N_k \geq 1$ is obtained as:

$$\begin{aligned} \lim_{k \rightarrow +\infty} P \left(\frac{M_k}{d_k} \leq m | N_k = l \right) &= \lim_{k \rightarrow +\infty} P \left(\frac{Z - u_k}{d_k} \leq m \mid Z > u_k \right)^l \\ &= \lim_{k \rightarrow +\infty} P \left(\frac{Z - c_k}{d_k} - u \leq m \mid \frac{Z - c_k}{d_k} > u \right)^l \\ &= (1 - e^{-m})^l. \end{aligned} \quad (22)$$

where we used (21). The distribution of M_k is found by marginalization over the number of excesses $1 \leq l \leq k$ conditioned on $N_k \geq 1$. From (i) one finds:

$$\begin{aligned} \lim_{k \rightarrow +\infty} P \left(\frac{M_k}{d_k} \leq m | N_k \geq 1 \right) &= \lim_{k \rightarrow +\infty} \sum_{l=1}^k P \left(\frac{M_k}{d_k} \leq m | N_k = l \right) P(N_k = l | N_k \geq 1) \\ &= \frac{1}{1 - e^{-\lambda}} \sum_{l=1}^{+\infty} (1 - e^{-m})^l \left(\frac{\lambda^l}{l!} e^{-\lambda} \right) \end{aligned}$$

Further simplification leads to:

$$\begin{aligned} \lim_{k \rightarrow +\infty} P \left(\frac{M_k}{d_k} \leq m | N_k \geq 1 \right) &= \frac{e^{-\lambda}}{1 - e^{-\lambda}} \sum_{l=1}^{+\infty} \frac{(\lambda(1 - e^{-m}))^l}{l!} \\ &= \frac{e^{-\lambda}}{1 - e^{-\lambda}} \left[\exp \left\{ \lambda - \lambda e^{-m} \right\} - 1 \right] \\ &= \frac{\exp \left\{ -\exp \left[-(m - \ln \lambda) \right] \right\} - e^{-\lambda}}{1 - e^{-\lambda}} \end{aligned}$$

which is the cumulative distribution function of a Gumbel member of the family (3) located at $\mu = \ln \lambda$ and conditioned on the positive real line.

(iii) From (21) it follows that the excesses $\frac{Z - c_i}{d_i} - u$ converge in distribution to an exponential distribution as $i \rightarrow +\infty$. Therefore, from the continuous mapping theorem (stating that

convergence is preserved by continuous transformation (Embrechts et al., 1997, p. 561), the mean of n such independent excesses converges to the distribution of a mean of n independent variables that are distributed according to an exponential distribution. Thus the limiting distribution conditioned on $N_k = l \geq 1$ is given by an Erlang distribution with shape-parameter l and rate parameter l (Feller, 1971, p. 11) with a cumulative distribution function:

$$\lim_{k \rightarrow +\infty} P\left(\frac{V_k}{d_k} \leq v \mid N_k = l\right) = 1 - \sum_{j=0}^{l-1} \frac{1}{j!} (lv)^j e^{-lv}$$

Marginalisation over the number of exceedances leads to:

$$\begin{aligned} \lim_{k \rightarrow +\infty} P\left(\frac{V_k}{d_k} \leq v \mid N_k \geq 1\right) &= \lim_{k \rightarrow +\infty} \sum_{l=1}^k P\left(\frac{V_k}{d_k} \leq v \mid N_k = l\right) P(N_k = l \mid N_k \geq 1) \\ &= \sum_{l=1}^{+\infty} \left(1 - \sum_{j=0}^{l-1} \frac{1}{j!} (lv)^j e^{-lv}\right) \left(\frac{\lambda^l e^{-\lambda}}{l! 1 - e^{-\lambda}}\right) \\ &= \frac{1}{e^{\lambda} - 1} \left((e^{\lambda} - 1) - \sum_{l=1}^{+\infty} \sum_{j=0}^{l-1} \frac{\lambda^l}{l! j!} (lv)^j e^{-lv} \right) \\ &= 1 - \frac{1}{e^{\lambda} - 1} \left(\sum_{l=1}^{+\infty} \sum_{j=0}^{l-1} \frac{\lambda^l}{l! j!} (lv)^j e^{-lv} \right) \end{aligned}$$

■

A.2 Proof of Theorem 2

Proof Convergence in distribution is expressed in terms of the joint (cumulative) distribution of the features V_k, M_k and N_k conditioned on $N_k \geq 1$:

$$F_k(v, m, n) = P\left(\frac{V_k}{d_k} \leq v, \frac{M_k}{d_k} \leq m, N_k \leq n \mid N_k \geq 1\right). \quad (23)$$

Clearly, the mean v of a sequence of n positive numbers is situated between $\frac{m}{n}$ and m such that the support of F_k is situated in $D = \{(v, m, n) \mid \frac{m}{n} \leq v \leq m\}$. The conditioned joint distribution (23) can be written as:

$$\begin{aligned} F_k(v, m, n) &= \sum_{l=1}^n \frac{P\left(\frac{V_k}{d_k} \leq v, \frac{M_k}{d_k} \leq m, N_k = l\right)}{1 - P(N_k = 0)} \\ &= \sum_{l=1}^n \frac{P\left(\frac{V_k}{d_k} \leq v \mid \frac{M_k}{d_k} \leq m, N_k = l\right) P\left(\frac{M_k}{d_k} \leq m \mid N_k = l\right) P(N_k = l)}{1 - P(N_k = 0)} \end{aligned} \quad (24)$$

The limiting distribution of (23) can be obtained by considering the limit of each factor in the nominators of the terms in (24) as $k \rightarrow +\infty$. Firstly, from Theorem 1-(i), it follows

$$\text{that:} \quad \lim_{k \rightarrow +\infty} P(N_k = l) = \frac{\lambda e^{-\lambda}}{l!}, \quad \lambda = e^{-u}. \quad (25)$$

Secondly, the limiting distribution of $P\left(\frac{M_k}{d_k} \leq m \mid N_k = l\right)$ is given by (22). Thirdly, the distribution $P\left(\frac{V_k}{d_k} \leq v \mid \frac{M_k}{d_k} \leq m, N_k = l\right)$ corresponds to the distribution of the mean of l independent exceedances that each converge in distribution to an exponential distribution truncated at m :

$$\begin{aligned} \lim_{k \rightarrow +\infty} P\left(\frac{Z - u_k}{d_k} \leq v \mid \frac{Z - u_k}{d_k} \leq m\right) &= \lim_{k \rightarrow +\infty} P\left(\frac{Z - c_k}{d_k} - u \leq v \mid \frac{Z - c_k}{d_k} - u \leq m\right) \\ &= \frac{1 - e^{-v}}{1 - e^{-m}}. \end{aligned}$$

Therefore, according to the continuous mapping theorem (Embrechts et al., 1997), the distribution of V_k converge in distribution to the sum of l truncated exponential distributions such that (Bain and Weeks, 1964):

$$\lim_{k \rightarrow +\infty} P\left(\frac{V_k}{d_k} \leq v \mid \frac{M_k}{d_k} \leq m, N_k = l\right) = \frac{1}{(1 - e^{-m})^l} \sum_{i=0}^r (-1)^i \binom{l}{i} e^{-im} \chi_{2i} (2(lv - im))$$

for $r = \lfloor \frac{lv}{m} \rfloor$. Substituting the latter expression together with (22) and (25) in the factorisation (24) gives the desired result. ■

References

- L.J. Bain and D.L. Weeks. A note on the truncated exponential distribution. *The Annals of Mathematical Statistics*, 35(3):1366–1367, 1964.
- C.M. Bishop. Novelty detection and neural network validation. In *Proceedings of the IEEE Conference on Vision, Image and Signal Processing*, volume 141, pages 217–222. IEE, London, 1994.
- C.M. Bishop. *Pattern Recognition and machine learning*. Springer, New York, USA, 2006.
- D.A. Clifton, S. Hugueny, and L. Tarassenko. Novelty detection with multivariate extreme value statistics. *Journal of Signal Processing Systems*, 65:371–389, 2011.
- K. Cuppens, P. Karsmakers, A. Van de Vel, B. Bomoy, M. Milosevic, S. Luca, B. Ceuilemans, L. Lagae, S. Van Huffel, and B. Vanrumste. Accelerometer based home monitoring for detection of nocturnal hypertor seizures based on novelty detection. *IEEE Journal of Biomedical and Health Informatics*, In Press, 2013.
- T.G. Dietterich. Machine learning for sequential data: A review. In *Proceedings of the Joint International Workshop on Structural Syntactic and Statistical Pattern Recognition*, pages 15–30. Springer-Verlag, London, 2002.

- P. Embrechts, C. Klüppelberg, and T. Mikosch. *Modelling Extremal Events for Insurance and Finance*. Springer, Berlin, 1997.
- M. Falk, J. Hüslér, and R.-D. Reiss. *Laws of small numbers: Extremes and rare events*. Birkhäuser, 3rd edition, 2011.
- W. Feller. *An Introduction to Probability Theory and Its Applications, Vol. 2*. Wiley, New York, 2nd edition, 1971.
- S. Luca, P. Karsmakers, K. Cuppens, T. Croonenborghs, A. Van de Vel, B. Ceulemans, L. Lagae, S. Van Huffel, and B. Vanrumste. Detecting rare events using extreme value statistics applied to epileptic convulsions in children. *Journal of Artificial Intelligence In Medicine*, 60(2):89–96, 2014a.
- S. Luca, P. Karsmakers, and B. Vanrumste. Anomaly detection using the Poisson process limit for extremes. In R. Kumar, H. Toivonen, J. Pei, Zhexue H., and X. Wu, editors, *IEEE International Conference on Data Mining*, pages 370–379, 2014b.
- Marco A. F. Pimentel, D.A. Clifton, L. Clifton, and L. Tarassenko. A review of novelty detection. *Signal Processing*, 99:215 – 249, 2014.
- L.R. Rabiner and H. Murray. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, volume 77, pages 257 – 286. IEEE, 1989.
- C. Scarrat and A. MacDonald. A review of extreme value threshold estimation and uncertainty quantification. *REVSTAT - Statistical journal*, 10(1):33–60, 2012.
- B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, and R.C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.
- D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley and Sons, New York, 1992.
- C. Sutton and A. McCallum. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4):267–373, 2011.

On the Influence of Momentum Acceleration on Online Learning

Kun Yuan

Bicheng Ying

Ali H. Sayed

Department of Electrical Engineering

University of California

Los Angeles, CA 90095, USA

KUNYUAN@UCLA.EDU

YBC@UCLA.EDU

SAYED@UCLA.EDU

Editor: Leon Bottou

Abstract

The article examines in some detail the convergence rate and mean-square-error performance of momentum stochastic gradient methods in the constant step-size and slow adaptation regime. The results establish that momentum methods are equivalent to the standard stochastic gradient method with a re-scaled (larger) step-size value. The size of the re-scaling is determined by the value of the momentum parameter. The equivalence result is established for all time instants and not only in steady-state. The analysis is carried out for general strongly convex and smooth risk functions, and is not limited to quadratic risks. One notable conclusion is that the well-known benefits of momentum constructions for deterministic optimization problems do not necessarily carry over to the adaptive online setting when small constant step-sizes are used to enable continuous adaptation and learning in the presence of persistent gradient noise. From simulations, the equivalence between momentum and standard stochastic gradient methods is also observed for non-differentiable and non-convex problems.

Keywords: Online Learning, Stochastic Gradient, Momentum, Acceleration, Heavy-ball Method, Nesterov's Method, Mean-Square-Error Analysis, Convergence Rate

1. Introduction

Stochastic optimization focuses on the problem of optimizing the expectation of a loss function, written as

$$\min_{w \in \mathbb{R}^M} J(w) \triangleq \mathbb{E}_{\theta} [Q(w; \theta)], \quad (1)$$

where θ is a random variable whose distribution is generally unknown and $J(w)$ is a convex function (usually strongly-convex due to regularization). If the probability distribution of the data, θ , is known beforehand, then one can evaluate $J(w)$ and seek its minimizer by means of a variety of gradient-descent or Newton-type methods (Polyak, 1987; Bertsekas, 1999; Nesterov, 2004). We refer to these types of problems, where $J(w)$ is known, as *deterministic* optimization problems. On the other hand, when the probability distribution of the data is unknown, then the risk function $J(w)$ is unknown as well; only instances

of the loss function, $Q(w; \theta)$, may be available at various observations θ_i , where i refers to the sample index. We refer to these types of problems, where $J(w)$ is unknown but defined implicitly as the expectation of some known loss form, as *stochastic* optimization problems. This article deals with this second type of problems, which are prevalent in online adaptation and learning contexts (Widrow and Stearns, 1985; Haykin, 2008; Sayed, 2008; Theodoridis, 2015).

When $J(w)$ is differentiable, one of the most popular techniques to seek minimizers for (1) is to employ the *stochastic* gradient method. This algorithm is based on employing instantaneous approximations for the true (unavailable) gradient vectors, $\nabla_w J(w)$, by using the gradients of the loss function, $\nabla_w Q(w; \theta_i)$, evaluated at successive samples of the streaming data θ_i over the iteration index i , say, as:

$$w_i = w_{i-1} - \mu \nabla_w Q(w_{i-1}; \theta_i), \quad i \geq 0. \quad (2)$$

where $\mu > 0$ is a step-size parameter. Note that we are denoting the successive iterates by w_i and using the boldface notation to refer to the fact that they are random quantities in view of the randomness in the measurements $\{\theta_i\}$. Due to their simplicity, robustness to noise and uncertainty, and scalability to big data, such stochastic gradient methods have become popular in large-scale optimization, machine learning, and data mining applications (Zhang, 2004; Bottou, 2010; Gemulla et al., 2011; Sutskever et al., 2013; Kahou et al., 2013; Cevher et al., 2014; Szegedy et al., 2015; Zareba et al., 2015).

1.1 Convergence Rate

Stochastic-gradient algorithms can be implemented with decaying step-sizes, such as $\mu(i) = \tau/i$ for some constant τ , or with constant step-sizes, $\mu > 0$. The former generally ensure asymptotic convergence to the true minimizer of (1), denoted by w^o , at a convergence rate that is on the order of $O(1/i)$ for strongly-convex risk functions. This guarantee, however, comes at the expense of turning off adaptation and learning as time progresses since the step-size value approaches zero in the limit, as $i \rightarrow \infty$. As a result, the algorithm loses the ability to track concept drifts. In comparison, constant step-sizes keep adaptation and learning alive and infuse a desirable tracking mechanism into the operation of the algorithm: even if the minimizers drift with time, the algorithm will generally be able to adjust and track their locations. Moreover, convergence can now occur at the considerably faster exponential rate, $O(\alpha^i)$, for some $\alpha \in (0, 1)$. These favorable properties come at the expense of a small deterioration in the limiting accuracy of the iterates since almost-sure convergence is not guaranteed any longer. Instead, the algorithm converges in the mean-square-error sense towards a small neighborhood around the true minimizer, w^o , whose radius is on the order of $O(\mu)$. This is still a desirable conclusion because the value of μ is controlled by the designer and can be chosen sufficiently small.

A well-known tradeoff therefore develops between convergence rate and mean-square-error (MSE) performance. The asymptotic MSE performance level approaches $O(\mu)$ while the convergence rate is given by $\alpha = 1 - O(\mu)$ (Polyak, 1987; Sayed, 2014a). It is nowadays well-recognized that the small $O(\mu)$ degradation in performance is acceptable in most large-scale learning and adaptation problems (Bousquet and Bottou, 2008; Bottou, 2010; Sayed, 2014b). This is because, in general, there are always modeling errors in formulating

optimization problems of the form (1): the cost function may not reflect perfectly the scenario and data under study. As such, insisting on attaining asymptotic convergence to the true minimizer may not be necessarily the best course of action or may not be worth the effort. It is often more advantageous to tolerate a small steady-state error that is negligible in most cases, but is nevertheless attained at a faster exponential rate of convergence than the slower rate of $O(1/n)$. Furthermore, the data models in many applications are more complex than assumed, with possibly local minima. In these cases, constant step-size implementations can help reduce the risk of being trapped at local solutions.

For these various reasons, and since our emphasis is on algorithms that are able to learn continuously, we shall focus on small *constant* step-size implementations. In these cases, gradient noise is always present, as opposed to decaying step-size implementations where the gradient noise terms get annihilated with time. The analysis in the paper will establish analytically, and illustrate by simulations, that, for sufficiently small step-sizes, any benefit from a momentum stochastic-construction can be attained by adjusting the step-size parameter for the original stochastic-gradient implementation. We emphasize here the qualification “small” for the step-size. The reason we focus on small step-sizes (which correspond to the slow adaptation regime) is because, in the stochastic context, mean-square-error stability and convergence require small step-sizes.

1.2 Acceleration Methods

In the *deterministic* optimization case, when the true gradient vectors of the smooth risk function $J(w)$ are available, the iterative algorithm for seeking the minimizer of $J(w)$ becomes the following gradient-descent recursion

$$w_i = w_{i-1} - \mu \nabla_w J(w_{i-1}), \quad i \geq 0, \quad (3)$$

There have been many ingenious methods proposed in the literature to enhance the convergence of these methods for both cases of convex and strongly-convex risks, $J(w)$. Two of the most notable and successful techniques are the heavy-ball method (Polyak, 1964, 1987; Qian, 1999) and Nesterov’s acceleration method (Nesterov, 1983, 2004, 2005) (the recursions for these algorithms are described in Section 3.1). The two methods are different but they both rely on the concept of adding a momentum term to the recursion. When the risk function $J(w)$ is ν -strongly convex and has δ -Lipschitz continuous gradients, both methods succeed in accelerating the gradient descent algorithm to attain a faster exponential convergence rate (Polyak, 1987) (Nesterov, 2004), and this rate is proven to be optimal for problems with smooth $J(w)$ and cannot be attained by standard gradient descent methods. Specifically, it is shown in (Polyak, 1987) (Nesterov, 2004) that for heavy-ball and Nesterov’s acceleration methods, the convergence of the iterates w_i towards w^o occurs at the rate:

$$\|w_i - w^o\|^2 \leq \left(\frac{\sqrt{\delta} - \sqrt{\nu}}{\sqrt{\delta} + \sqrt{\nu}} \right)^2 \|w_{i-1} - w^o\|^2, \quad (4)$$

In comparison, in Theorem 2.1.15 of (Nesterov, 2005) and Theorem 4 in Section 1.4 of (Polyak, 1987), the fastest rate for gradient descent method is shown to be

$$\|w_i - w^o\|^2 \leq \left(\frac{\delta - \nu}{\delta + \nu} \right)^2 \|w_{i-1} - w^o\|^2. \quad (5)$$

It can be verified that

$$\frac{\sqrt{\delta} - \sqrt{\nu}}{\sqrt{\delta} + \sqrt{\nu}} < \frac{\delta - \nu}{\delta + \nu} \quad (6)$$

when $\delta > \nu$. This inequality confirms that the momentum algorithm can achieve a faster rate in deterministic optimization and, moreover, this faster rate cannot be attained by standard gradient descent.

Motivated by these useful acceleration properties in the *deterministic* context, momentum terms have been subsequently introduced into *stochastic* optimization algorithms as well (Polyak, 1987; Proakis, 1974; Sharma et al., 1998; Shynk and Roy, June 1988; Roy and Shynk, 1990; Tugay and Tank, 1989; Bellanger, 2001; Wiegand et al., 1994; Hu et al., 2009; Xiao, 2010; Lan, 2012; Ghadimi and Lan, 2012; Zhong and Kwok, 2014) and applied, for example, to problems involving the tracking of chirped sinusoidal signals (Ting et al., 2000) or deep learning (Sutskever et al., 2013; Kahou et al., 2013; Szegedy et al., 2015; Zareba et al., 2015). However, the analysis in this paper will show that the advantages of the momentum technique for deterministic optimization do not necessarily carry over to the *adaptive* online setting due to the presence of stochastic gradient noise (which is the difference between the actual gradient vector and its approximation). Specifically, for sufficiently small step-sizes and for a momentum parameter not too close to one, we will show that any advantage brought forth by the momentum term can be achieved by staying with the original stochastic-gradient algorithm and adjusting its step-size to a larger value. For instance, for optimization problem (1), we will show that if the step-sizes, μ_m for the momentum (heavy-ball or Nesterov) methods and μ for the standard stochastic gradient algorithms, are sufficiently small and satisfy the relation

$$\mu = \frac{\mu_m}{1 - \beta} \quad (7)$$

where β , a positive constant that is not too close to 1, is the momentum parameter, then it will hold that

$$\mathbb{E}\|w_{m,i} - w_i\|^2 = O(\mu^{3/2}), \quad i = 0, 1, 2, \dots \quad (8)$$

where $w_{m,i}$ and w_i denote the iterates generated at time i by the momentum and standard implementations, respectively. In the special case when $J(w)$ is quadratic in w , as happens in mean-square-error design problems, we can tighten (8) to

$$\mathbb{E}\|w_{m,i} - w_i\|^2 = O(\mu^2), \quad i = 0, 1, 2, \dots \quad (9)$$

What is important to note is that, we will show that these results hold *for every i* , and not only asymptotically. Therefore, when μ is sufficiently small, property (8) establishes that the stochastic gradient method and the momentum versions are fundamentally equivalent

since their iterates evolve close to each other at all times. We establish this equivalence result under the situation where the risk function is convex and differentiable. However, as our numerical simulations over a multi-layer fully connected neural network and a second convolutional neural network (see Section 7.4) show, the equivalence between standard and momentum stochastic gradient methods are also observed in non-convex and non-differentiable scenarios.

1.3 Related Works in the Literature

There are useful results in the literature that deal with special instances of the general framework developed in this work. These earlier results focus mainly on the mean-square-error case when $J(w)$ is quadratic in w , in which case the stochastic gradient algorithm reduces to the famed least-mean-squares (LMS) algorithm. We will not be limiting our analysis to this case so that our results will be applicable to a broader class of learning problems beyond mean-square-error estimation (e.g., logistic regression would be covered by our results as well). As the analysis and derivations will reveal, the treatment of the general $J(w)$ case is demanding because the Hessian matrix of $J(w)$ is now w -dependent, whereas it is a constant matrix in the quadratic case.

Some of the earlier investigations in the literature led to the following observations. It was noted in (Polyak, 1987) that, for quadratic costs, stochastic gradient implementations with a momentum term do not necessarily perform well. This work remarks that although the heavy-ball method can lead to faster convergence in the early stages of learning, it nevertheless converges to a region with worse mean-square-error in comparison to standard stochastic-gradient (or LMS) iteration. A similar phenomenon is also observed in (Proakis, 1974; Sharma et al., 1998). However, in the works (Proakis, 1974; Polyak, 1987; Sharma et al., 1998), no claim is made or established about the equivalence between momentum and standard methods.

Heavy-ball LMS was further studied in the useful works (Roy and Shynk, 1990) and (Tugay and Tanik, 1989). The reference (Roy and Shynk, 1990) claimed that no significant gain is achieved in convergence speed if both the heavy-ball and standard LMS algorithms approach the same *steady-state* MSE performance. Reference (Tugay and Tanik, 1989) observed that when the step-sizes satisfy relation (7), then heavy-ball LMS is “equivalent” to standard LMS. However, they assumed Gaussian measurement noise in their data model, and the notion of “equivalence” in this work is only referring to the fact that the algorithms have similar starting convergence rates and similar steady-state MSE levels. There was no analysis in (Tugay and Tanik, 1989) of the behavior of the algorithms during all stages of learning – see also (Bellanger, 2001). Another useful work is (Wiegierneck et al., 1994), which considered the heavy-ball stochastic gradient method for general risk, $J(w)$. By assuming a sufficiently small step-size, and by transforming the error difference recursion into a differential equation, the work concluded that heavy-ball can be equivalent to the standard stochastic gradient method asymptotically (i.e., for i large enough). No results were provided for the earlier stages of learning.

All of these previous works were limited to examining the heavy-ball momentum technique; none of them considered other forms of acceleration such as Nesterov’s technique although this latter technique is nowadays widely applied to stochastic gradient learning,

including deep learning (Sutskever et al., 2013; Kahou et al., 2013; Szegedy et al., 2015; Zareba et al., 2015). The performance of Nesterov’s acceleration with *deterministic* and *bounded* gradient error was examined in (d’Aspremont, 2008; Devolder et al., 2014; Lessard et al., 2016). The source of the inaccuracy in the gradient vector in these works is either because the gradient was assessed by solving an auxiliary “simpler” optimization problem or because of numerical approximations. Compared to the standard gradient descent implementation, the works by (d’Aspremont, 2008; Lessard et al., 2016) claimed that Nesterov’s acceleration is not robust to the errors in gradient. The work by (Devolder et al., 2014) also observed that the superiority of Nesterov’s acceleration is no longer absolute when inexact gradients are used, and they further proved that the performance of Nesterov’s acceleration may be even worse than gradient descent due to error accumulation. These works assumed bounded errors in the gradient vectors and focused on the context of deterministic optimization. None of the works examined the stochastic setting where the gradient error is random in nature and where the assumption of bounded errors are generally unsuitable. We may add that there have also been analyses of Nesterov’s acceleration for *stochastic* optimization problems albeit for *decaying* step-sizes in more recent literature (Hu et al., 2009; Xiao, 2010; Lan, 2012; Ghadimi and Lan, 2012; Zhong and Kwok, 2014). These works proved that Nesterov’s acceleration can improve the convergence rate of stochastic gradient descent at the initial stages when deterministic risk components dominate; while at the asymptotic stages when the stochastic gradient noise dominates, the momentum correction cannot accelerate convergence any more. Another useful study is (Flammarion and Bach, 2015), in which the authors showed that momentum and averaging methods for stochastic optimization are equivalent to the same second-order difference equations but with different step-sizes. However, (Flammarion and Bach, 2015) does not study the equivalence between standard and momentum stochastic gradient methods, and they focus on quadratic problems and also employ decaying step-sizes.

Finally, we note that there are other forms of stochastic gradient algorithms for empirical risk minimization problems where momentum acceleration has been shown to be useful. Among them, we list recent algorithms like SAG (Roux et al., 2012), SVRG (Johnson and Zhang, 2013) and SAGA (Defazio et al., 2014). In these algorithms, the variance of the stochastic gradient noise diminishes to zero and the deterministic component of the risk becomes dominant in the asymptotic regime. In these situations, momentum acceleration helps improve the convergence rate, as noted by (Nitanda, 2014) and (Zhu, 2016). Another family of algorithms to solve empirical risk minimization problems are stochastic dual coordinate ascent (SDCA) algorithms. It is proved in (Shalev-Shwartz, 2015; Johnson and Zhang, 2013) that SDCA can be viewed as a variance-reduced stochastic algorithm, and hence momentum acceleration can also improve its convergence for the same reason noted by (Shalev-Shwartz and Zhang, 2014).

In this paper, we are studying online training algorithms where data can stream in continuously as opposed to running multiple passes over a finite amount of data. In this case, the analysis will help clarify the limitations of momentum acceleration in the slow adaptation regime. We are particularly interested in the constant step-size case, which enables continuous adaptation and learning and is regularly used, e.g., in deep learning implementations. There is a non-trivial difference between the decaying and constant step-size situations. This is because gradient noise is always present in the constant step-size

case, while it is annihilated in the decaying step-size case. The presence of the gradient noise interferes with the dynamics of the algorithms in a non-trivial way, which is what our analysis discovers. There are limited analyses for the constant step-sizes scenario.

1.4 Outline of Paper

The outline of the paper is as follows. In Section 2, we introduce some basic assumptions and review the stochastic gradient method and its convergence properties. In Section 3 we embed the heavy-ball and Nesterov’s acceleration methods into a unified momentum algorithm, and subsequently establish the mean-square stability and fourth-order stability of the error moments. Next, we analyze the equivalence between momentum and standard LMS algorithms in Section 4 and then extend the results to general risk functions in Section 5. In Section 6 we extend the equivalence results into a more general setting with diagonal step-size matrices. We illustrate our results in Section 7, and in Section 8 we comment on the stability ranges of standard and momentum stochastic gradient methods.

2. Stochastic Gradient Algorithms

In this section we review the stochastic gradient method and its convergence properties. We denote the minimizer for problem (1) by w^o , i.e.,

$$w^o \triangleq \arg \min_w J(w). \quad (10)$$

We introduce the following assumption on $J(w)$, which essentially amounts to assuming that $J(w)$ is strongly-convex with Lipschitz gradient. These conditions are satisfied by many problems of interest, especially when regularization is employed (e.g., mean-square-error risks, logistic risks, etc.). Under the strong-convexity condition, the minimizer w^o is unique.

Assumption 1 (Conditions on risk function) *The cost function $J(w)$ is twice differentiable and its Hessian matrix satisfies*

$$0 < \nu I_M \leq \nabla^2 J(w) \leq \delta I_M, \quad (11)$$

for some positive parameters $\nu \leq \delta$. Condition (11) is equivalent to requiring $J(w)$ to be ν -strongly convex and for its gradient vector to be δ -Lipschitz, respectively (Boyd and Vandenberghe, 2004; Sayed, 2014a). ■

The stochastic-gradient algorithm for seeking w^o takes the form (2), with initial condition w_{-1} . The difference between the true gradient vector and its approximation is designated *gradient noise* and is denoted by:

$$s_i(w_{i-1}) \triangleq \nabla_w Q(w_{i-1}; \theta_i) - \nabla_w \mathbb{E}[Q(w_{i-1}; \theta_i)]. \quad (12)$$

In order to examine the convergence of the standard and momentum stochastic gradient methods, it is necessary to introduce some assumptions on the stochastic gradient noise.

Assumptions (13) and (14) below are satisfied by important cases of interest, as shown in (Sayed, 2014a) and (Sayed, 2014b), such as logistic regression and mean-square-error risks. Let the symbol \mathcal{F}_{i-1} represent the filtration generated by the random process w_j for $j \leq i-1$ (basically, the collection of past history until time $i-1$):

$$\mathcal{F}_{i-1} \triangleq \text{filtration}\{w_{-1}, w_0, w_1, \dots, w_{i-1}\}.$$

Assumption 2 (Conditions on gradient noise) *It is assumed that the first and second-order conditional moments of the gradient noise process satisfy the following conditions for any $w \in \mathcal{F}_{i-1}$:*

$$\mathbb{E}[s_i(w) | \mathcal{F}_{i-1}] = 0 \quad (13)$$

$$\mathbb{E}[\|s_i(w)\|^2 | \mathcal{F}_{i-1}] \leq \gamma^2 \|w^o - w\|^2 + \sigma_s^2 \quad (14)$$

almost surely, for some nonnegative constants γ^2 and σ_s^2 . ■

Condition (13) essentially requires the gradient noise process to have zero mean, which amounts to requiring the approximate gradient to correspond to an unbiased construction for the true gradient. This is a reasonable requirement. Condition (14) requires the size of the gradient noise (i.e., its mean-square value) to diminish as the iterate w gets closer to the solution w^o . This is again a reasonable requirement since it amounts to expecting the gradient noise to get reduced as the algorithm approaches the minimizer. Under Assumptions 1 and 2, the following conclusion is proven in Lemma 3.1 of (Sayed, 2014a).

Lemma 1 (Second-order stability) *Let Assumptions 1 and 2 hold, and consider the stochastic gradient recursion (2). Introduce the error vector $\tilde{w}_i = w^o - w_i$. Then, for any step-sizes μ satisfying*

$$\mu < \frac{2\nu}{\delta^2 + \gamma^2}, \quad (15)$$

it holds for each iteration $i = 0, 1, 2, \dots$ that

$$\mathbb{E}\|\tilde{w}_i\|^2 \leq (1 - \mu\nu)\mathbb{E}\|\tilde{w}_{i-1}\|^2 + \mu^2\sigma_s^2, \quad (16)$$

and, furthermore,

$$\limsup_{i \rightarrow \infty} \mathbb{E}\|\tilde{w}_i\|^2 \leq \frac{\sigma_s^2 \mu}{\nu} = O(\mu). \quad (17)$$

■

We can also examine the stability of the fourth-order error moment, $\mathbb{E}\|\tilde{w}_i\|^4$, which will be used later in Section 5 to establish the equivalence between the standard and momentum stochastic implementations. For this case, we tighten the assumption on the gradient noise by replacing the bound in (14) on its second-order moment by a similar bound involving its fourth-order moment. Again, this assumption is satisfied by problems of interest, such as mean-square-error and logistic risks (Sayed, 2014a,b).

Assumption 3 (Conditions on gradient noise) *It is assumed that the first and fourth-order conditional moments of the gradient noise process satisfy the following conditions for any $\mathbf{w} \in \mathcal{F}_{i-1}$:*

$$\mathbb{E}[\mathbf{s}_i(\mathbf{w}) | \mathcal{F}_{i-1}] = 0 \quad (18)$$

$$\mathbb{E}[\|\mathbf{s}_i(\mathbf{w})\|^4 | \mathcal{F}_{i-1}] \leq \gamma_4^4 \|\mathbf{w}^o - \mathbf{w}\|^4 + \sigma_{s,4}^4 \quad (19)$$

■ *almost surely, for some nonnegative constants γ_4^4 and $\sigma_{s,4}^4$.*

It is straightforward to check that if Assumption 3 holds, then Assumption 2 will also hold. The following conclusion is a modified version of Lemma 3.2 of (Sayed, 2014a).

Lemma 2 (Fourth-order stability) *Let the conditions under Assumptions 1 and 3 hold, and consider the stochastic gradient iteration (2). For sufficiently small step-size μ , it holds that*

$$\mathbb{E}\|\tilde{\mathbf{w}}_i\|^4 \leq \rho^{i+1} \mathbb{E}\|\tilde{\mathbf{w}}_{-1}\|^4 + A\sigma_s^2(i+1)\rho^{i+1}\mu^2 + \frac{B\sigma_s^4\mu^2}{\nu^2} \quad (20)$$

where $\rho \triangleq 1 - \mu\nu$, and A and B are some constants. Furthermore,

$$\limsup_{i \rightarrow \infty} \mathbb{E}\|\tilde{\mathbf{w}}_i\|^4 \leq \frac{B\sigma_s^4\mu^2}{\nu^2} = O(\mu^2) \quad (21)$$

■ **Proof** See Appendix A.

3. Momentum Acceleration

In this section, we present a generalized momentum stochastic gradient method, which captures both the heavy-ball and Nesterov's acceleration methods as special cases. Subsequently, we derive results for its convergence property.

3.1 Momentum Stochastic Gradient Method

Consider the following general form of a stochastic-gradient implementation, with two momentum parameters $\beta_1, \beta_2 \in [0, 1)$:

$$\psi_{i-1} = \mathbf{w}_{i-1} + \beta_1(\mathbf{w}_{i-1} - \mathbf{w}_{i-2}), \quad (22)$$

$$\mathbf{w}_i = \psi_{i-1} - \mu_m \nabla_w Q(\psi_{i-1}; \boldsymbol{\theta}_i) + \beta_2(\psi_{i-1} - \psi_{i-2}), \quad (23)$$

with initial conditions

$$\mathbf{w}_{-2} = \psi_{-2} = \text{initial states}, \quad (24)$$

$$\mathbf{w}_{-1} = \mathbf{w}_{-2} - \mu_m \nabla_w Q(\mathbf{w}_{-2}; \boldsymbol{\theta}_{-1}), \quad (25)$$

where μ_m is some constant step-size. We refer to this formulation as the momentum stochastic gradient method.¹

When $\beta_1 = 0$ and $\beta_2 = \beta$ we recover the heavy-ball algorithm (Polyak, 1964, 1987), and when $\beta_2 = 0$ and $\beta_1 = \beta$, we recover Nesterov's algorithm (Nesterov, 2004). We note that Nesterov's method has several useful variations that fit different scenarios, such as situations involving smooth but not strongly-convex risks (Nesterov, 1983, 2004) or non-smooth risks (Nesterov, 2005; Beck and Teboulle, 2009). However, for the case when $J(w)$ is strongly convex and has Lipschitz continuous gradients, the Nesterov construction reduces to what is presented above, with a constant momentum parameter. This type of construction has also been studied in (Lessard et al., 2016; Dieuleveut et al., 2016) and applied in deep learning implementations (Sutskever et al., 2013; Kahou et al., 2013; Szegedy et al., 2015; Zareba et al., 2015).

In order to capture both the heavy-ball and Nesterov's acceleration methods in a unified treatment, we will assume that

$$\beta_1 + \beta_2 = \beta, \quad \beta_1, \beta_2 = 0, \quad (26)$$

for some fixed constant $\beta \in [0, 1)$. Next we introduce a condition on the momentum parameter.

Assumption 4 *The momentum parameter β is a constant that is not too close to 1, i.e., there exists a small fixed constant $\epsilon > 0$ such that $\beta \leq 1 - \epsilon$.* ■

Assumption 4 is quite common in studies on adaptive signal processing and neural networks — see, e.g., (Tugay and Tanik, 1989; Roy and Slynn, 1990; Bellanger, 2001; Wiegierinck et al., 1994; Attoh-Okine, 1999). Also, in recent deep learning applications it is common to set $\beta = 0.9$, which satisfies Assumption 4 (Krizhevsky et al., 2012; Szegedy et al., 2015; Zhang and LeCun, 2015). Under (26), the work (Flammario and Bach, 2015) also considers recursions related to (22)–(23) for the special case of quadratic risks.

3.2 Mean-Square Error Stability

In preparation for studying the performance of the momentum stochastic gradient method, we first show in the next result how recursions (22)–(23) can be transformed into a first-order recursion by defining extended state vectors. We introduce the transformation matrices:

$$V = \begin{bmatrix} I_M & -\beta I_M \\ I_M & -I_M \end{bmatrix}, \quad V^{-1} = \frac{1}{1-\beta} \begin{bmatrix} I_M & -\beta I_M \\ I_M & -I_M \end{bmatrix}. \quad (27)$$

Recall $\tilde{\mathbf{w}}_i = \mathbf{w}^o - \mathbf{w}_i$ and define the transformed error vectors, each of size $2M \times 1$:

$$\begin{bmatrix} \hat{\tilde{\mathbf{w}}}_i \\ \check{\tilde{\mathbf{w}}}_i \end{bmatrix} \triangleq V^{-1} \begin{bmatrix} \tilde{\mathbf{w}}_i \\ \tilde{\mathbf{w}}_{i-1} \end{bmatrix} = \frac{1}{1-\beta} \begin{bmatrix} \tilde{\mathbf{w}}_i - \beta \tilde{\mathbf{w}}_{i-1} \\ \tilde{\mathbf{w}}_i - \tilde{\mathbf{w}}_{i-1} \end{bmatrix}. \quad (28)$$

¹ Traditionally, the terminology of a ‘‘momentum method’’ has been used more frequently for the heavy-ball method, which corresponds to the special case $\beta_1 = 0$ and $\beta_2 = \beta$. Given the unified description (22)–(23), we will use this same terminology to refer to both the heavy-ball and Nesterov's acceleration methods.

Lemma 3 (Extended recursion) Under Assumption 1 and condition (26), the momentum stochastic gradient recursion (22)–(23) can be transformed into the following extended recursion:

$$\begin{bmatrix} \widehat{\mathbf{w}}_i \\ \widetilde{\mathbf{w}}_i \end{bmatrix} = \begin{bmatrix} I_M - \frac{\mu_m}{1-\beta} \mathbf{H}_{i-1} & \frac{\mu_m \beta'}{1-\beta} \mathbf{H}_{i-1} \\ -\frac{\mu_m}{1-\beta} \mathbf{H}_{i-1} & \beta I_M + \frac{\mu_m \beta'}{1-\beta} \mathbf{H}_{i-1} \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{w}}_{i-1} \\ \widetilde{\mathbf{w}}_{i-1} \end{bmatrix} + \frac{\mu_m}{1-\beta} \begin{bmatrix} \mathbf{s}_i(\psi_{i-1}) \\ \mathbf{s}_i(\psi_{i-1}) \end{bmatrix}, \quad (29)$$

where $\mathbf{s}_i(\psi_{i-1})$ is defined according to (12) and

$$\beta' \triangleq \beta \beta_1 + \beta_2, \quad (30)$$

$$\mathbf{H}_{i-1} \triangleq \int_0^1 \nabla_w^2 J(w^\circ - t\widetilde{\psi}_{i-1}) dt, \quad (31)$$

where $\widetilde{\psi}_{i-1} = w^\circ - \psi_{i-1}$.

Proof See Appendix B. \blacksquare

The transformed recursion (29) is important for at least two reasons. First, it is a first-order recursion, which facilitates the convergence analysis of $\widehat{\mathbf{w}}_i$ and $\widetilde{\mathbf{w}}_i$ and, subsequently, of the error vector $\widehat{\mathbf{w}}_i$ in view of (28) — see next theorem. Second, as we will explain later, the first row of (29) turns out to be closely related to the standard stochastic gradient iteration; this relation will play a critical role in establishing the claimed equivalence between momentum and standard stochastic gradient methods.

The following statement establishes the convergence property of the momentum stochastic gradient algorithm. It shows that recursions (22)–(23) converge exponentially fast to a small neighborhood around w° with a steady-state error variance that is on the order of $O(\mu_m)$. Note that in the following theorem the notation $a \preceq b$, for two vectors a and b , signifies element-wise comparisons.

Theorem 4 (Mean-square stability) Let Assumptions 1, 2 and 4 hold and recall conditions (26). Consider the momentum stochastic gradient method (22)–(23) and the extended recursion (29). Then, when step-sizes μ_m satisfies

$$\mu_m \leq \frac{(1-\beta)^2 \nu}{32\gamma^2 \nu^2 + 4\delta^2}, \quad (32)$$

it holds that the mean-square values of the transformed error vectors evolve according to the following recursive inequality:

$$\begin{bmatrix} \mathbb{E}\|\widehat{\mathbf{w}}_i\|^2 \\ \mathbb{E}\|\widetilde{\mathbf{w}}_i\|^2 \end{bmatrix} \preceq \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} \mathbb{E}\|\widehat{\mathbf{w}}_{i-1}\|^2 \\ \mathbb{E}\|\widetilde{\mathbf{w}}_{i-1}\|^2 \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}, \quad (33)$$

where

$$\begin{aligned} a &= 1 - \frac{\mu_m \nu}{1-\beta} + O(\mu_m^2), & b &= \frac{\mu_m \beta^2 \delta^2}{\nu(1-\beta)} + O(\mu_m^2), & c &= \frac{2\mu_m^2 \delta^2}{(1-\beta)^3} + \frac{2\mu_m^2 \gamma^2 (1+\beta_1)^2 \nu^2}{(1-\beta)^2}, \\ d &= \beta + O(\mu_m^2), & e &= \frac{\mu_m^2 \sigma_s^2}{(1-\beta)^2}, & f &= \frac{\mu_m^2 \sigma_s^2}{(1-\beta)^2}. \end{aligned} \quad (34)$$

and the coefficient matrix appearing in (33) is stable, namely,

$$\rho \left(\begin{bmatrix} a & b \\ c & d \end{bmatrix} \right) < 1. \quad (35)$$

Furthermore, if μ_m is sufficiently small it follows from (33) that

$$\limsup_{i \rightarrow \infty} \mathbb{E}\|\widehat{\mathbf{w}}_i\|^2 = O\left(\frac{\mu_m \sigma_s^2}{(1-\beta)^\nu}\right), \quad \limsup_{i \rightarrow \infty} \mathbb{E}\|\widetilde{\mathbf{w}}_i\|^2 = O\left(\frac{\mu_m^2 \sigma_s^2}{(1-\beta)^3}\right), \quad (36)$$

and, consequently,

$$\limsup_{i \rightarrow \infty} \mathbb{E}\|\widetilde{\mathbf{w}}_i\|^2 = O\left(\frac{\mu_m \sigma_s^2}{(1-\beta)^\nu}\right). \quad (37)$$

Proof See Appendix C. \blacksquare

Although $\mathbb{E}\|\mathbf{w}_i\|^2 = O(\mu_m^2)$ in result (36) is shown to hold asymptotically in the statement of the theorem, it can actually be strengthened and shown to hold for *all* time instants. This fact is crucial for our later proof of the equivalence between standard and momentum stochastic gradient methods.

Corollary 5 (Uniform mean-square bound) Under the same conditions as Theorem 4, it holds for sufficiently small step-sizes that

$$\mathbb{E}\|\mathbf{w}_i\|^2 = O\left(\frac{(\delta^2 + \gamma^2)\rho^{i+1}\mu_m^2}{(1-\beta)^4} + \frac{\sigma_s^2 \mu_m^2}{(1-\beta)^3}\right), \quad \forall i = 0, 1, 2, \dots \quad (38)$$

where $\rho_1 \triangleq 1 - \frac{\mu_m \nu}{2(1-\beta)}$, and $\widetilde{\mathbf{w}}_i$ is defined in (29).

Proof See Appendix D. \blacksquare

Corollary 5 has two implications. First, since $\beta, \delta, \gamma, \sigma_s^2$ are all constants, and $\rho_1 < 1, \alpha < 1$, we conclude that

$$\mathbb{E}\|\widetilde{\mathbf{w}}_i\|^2 = O(\mu_m^2), \quad \forall i = 0, 1, 2, \dots \quad (39)$$

Besides, since $\rho_1^i \rightarrow 0$ as $i \rightarrow \infty$, according to (38) we also achieve

$$\limsup_{i \rightarrow \infty} \mathbb{E}\|\mathbf{w}_i\|^2 = O\left(\frac{\sigma_s^2 \mu_m^2}{(1-\beta)^3}\right), \quad (40)$$

which is consistent with (36).

3.3 Stability of Fourth-Order Error Moment

In a manner similar to the treatment in Section 2, we can also establish the convergence of the fourth-order moments of the error vectors, $\mathbb{E}\|\tilde{\mathbf{w}}_i\|^4$ and $\mathbb{E}\|\tilde{\mathbf{w}}_i\|^4$.

Theorem 6 (Fourth-order stability) *Let Assumptions 1, 3 and 4 hold and recall conditions (26). Then, for sufficiently small step-sizes μ_m , it holds that*

$$\limsup_{i \rightarrow \infty} \mathbb{E}\|\tilde{\mathbf{w}}_i\|^4 = O(\mu_m^2), \quad (41)$$

$$\limsup_{i \rightarrow \infty} \mathbb{E}\|\tilde{\mathbf{w}}_i\|^4 = O(\mu_m^4), \quad (42)$$

$$\limsup_{i \rightarrow \infty} \mathbb{E}\|\tilde{\mathbf{w}}_i\|^4 = O(\mu_m^2). \quad (43)$$

■ **Proof** See Appendix E.

Again, result (42) is only shown to hold asymptotically in the statement of the theorem. In fact, $\mathbb{E}\|\tilde{\mathbf{w}}_i\|^4$ can also be shown to be bounded for all time instants, as the following corollary states.

Corollary 7 (Uniform forth-moment bound) *Under the same conditions as Theorem 6, it holds for sufficiently small step-sizes that*

$$\mathbb{E}\|\tilde{\mathbf{w}}_i\|^4 = O\left(\frac{\gamma^2 \rho_2^{i+1}}{(1-\beta)^3 \mu_m} + \left[\frac{\sigma_s^2(\delta^2 + \gamma^2)(i+1)\rho_2^{i+1}}{(1-\beta)^7} + \frac{(\gamma^2 + \nu^2)\sigma_s^4 + \nu^2\sigma_{s,4}^4}{(1-\beta)^6 \nu^2} \mu_m^4\right]\right) \quad (44)$$

where $\rho_2 \triangleq 1 - \frac{\mu_m \nu}{4(1-\beta)} \in (0, 1)$.

■ **Proof** See Appendix F.

Corollary 7 also has two implications. First, since β , δ , γ , σ_s and $\sigma_{s,4}$ are constants, we conclude that

$$\mathbb{E}\|\tilde{\mathbf{w}}_i\|^4 = O(\mu_m^2), \quad \forall i = 0, 1, 2, \dots \quad (45)$$

Besides, since $\rho_2^i \rightarrow 0$ and $i\rho_2^i \rightarrow 0$ as $i \rightarrow \infty$, we will achieve the following fact according to (44)

$$\limsup_{i \rightarrow \infty} \mathbb{E}\|\tilde{\mathbf{w}}_i\|^4 = O\left(\frac{(\gamma^2 + \nu^2)\sigma_s^4 + \nu^2\sigma_{s,4}^4}{(1-\beta)^6 \nu^2} \mu_m^4\right) = O(\mu_m^4), \quad (46)$$

which is consistent with (42).

4. Equivalence in the Quadratic Case

In Section 3 we showed the momentum stochastic gradient algorithm (22)–(23) converges exponentially for sufficiently small step-sizes. But some important questions remain. Does the

momentum implementation converge faster than the standard stochastic gradient method (2)? Does the momentum implementation lead to superior steady-state mean-square-deviation (MSD) performance, measured in terms of the limiting value of $\mathbb{E}\|\tilde{\mathbf{w}}_i\|^2$? Is the momentum method generally superior to the standard method when considering both the convergence rate and MSD performance? In this and the next sections, we answer these questions in some detail. Before treating the case of general risk functions, $J(w)$, we examine first the special case when $J(w)$ is quadratic in w to illustrate the main conclusions that will follow.

4.1 Quadratic Risks

We consider mean-square-error risks of the form

$$J(w) = \frac{1}{2} \mathbb{E} \left(\mathbf{d}(i) - \mathbf{u}_i^\top w \right)^2, \quad (47)$$

where $\mathbf{d}(i)$ denotes a streaming sequence of zero-mean random variables with variance $\sigma_d^2 = \mathbb{E} \mathbf{d}^2(i)$, and $\mathbf{u}_i \in \mathbb{R}^M$ denotes a streaming sequence of independent zero-mean random vectors with covariance matrix $R_{u_i} = \mathbb{E} \mathbf{u}_i \mathbf{u}_i^\top > 0$. The cross covariance vector between $\mathbf{d}(i)$ and \mathbf{u}_i is denoted by $r_{du} = \mathbb{E} \mathbf{d}(i) \mathbf{u}_i$. The data $\{\mathbf{d}(i), \mathbf{u}_i\}$ are assumed to be wide-sense stationary and related via a linear regression model of the form:

$$\mathbf{d}(i) = \mathbf{u}_i^\top w^o + \mathbf{v}(i), \quad (48)$$

for some unknown w^o , and where $\mathbf{v}(i)$ is a zero-mean white noise process with power $\sigma_v^2 = \mathbb{E} \mathbf{v}^2(i)$ and assumed independent of \mathbf{u}_j for all i, j . If we multiply (48) by \mathbf{u}_i from the left and take expectations, we find that the model parameter w^o satisfies the normal equations $R_u w^o = r_{du}$. The unique solution that minimizes (47) also satisfies these same equations. Therefore, minimizing the quadratic risk (47) enables us to recover the desired w^o . This observation explains why mean-square-error costs are popular in the context of regression models.

4.2 Adaptation Methods

For the least-mean-squares problem (47), the true gradient vector at any location \mathbf{w} is

$$\nabla_w J(\mathbf{w}) = R_u \mathbf{w} - r_{du} = -R_u(w^o - \mathbf{w}), \quad (49)$$

while the approximate gradient vector constructed from an instantaneous sample realization is:

$$\nabla_w Q(\mathbf{w}; \mathbf{d}(i), \mathbf{u}_i) = -\mathbf{u}_i (\mathbf{d}(i) - \mathbf{u}_i^\top \mathbf{w}). \quad (50)$$

Here the loss function is defined by

$$Q(\mathbf{w}; \mathbf{d}(i), \mathbf{u}_i) \triangleq \frac{1}{2} \mathbb{E} \left(\mathbf{d}(i) - \mathbf{u}_i^\top \mathbf{w} \right)^2 \quad (51)$$

The resulting LMS (stochastic-gradient) recursion is given by

$$\mathbf{w}_i = \mathbf{w}_{i-1} + \mu \mathbf{u}_i (\mathbf{d}(i) - \mathbf{u}_i^\top \mathbf{w}_{i-1}) \quad (52)$$

and the corresponding gradient noise process is

$$\mathbf{s}_i(\mathbf{w}) = (R_u - \mathbf{u}_i \mathbf{u}_i^\top)(w^\circ - \mathbf{w}) - \mathbf{u}_i v(i). \quad (53)$$

It can be verified that this noise process satisfies Assumption 2 — see Example 3.3 in (Sayed, 2014a). Subtracting w° from both sides of (52), and recalling that $\tilde{\mathbf{w}}_i = w^\circ - \mathbf{w}_i$, we obtain the error recursion that corresponds to the LMS implementation:

$$\tilde{\mathbf{w}}_i = (I_M - \mu R_u) \tilde{\mathbf{w}}_{i-1} + \mu \mathbf{s}_i(\mathbf{w}_{i-1}), \quad (54)$$

where μ is some constant step-size. In order to distinguish the variables for LMS from the variables for the momentum LMS version described below, we replace the notation $\{\mathbf{w}_i, \tilde{\mathbf{w}}_i\}$ for LMS by $\{\mathbf{x}_i, \tilde{\mathbf{x}}_i\}$ and keep the notation $\{\mathbf{w}_i, \tilde{\mathbf{w}}_i\}$ for momentum LMS, i.e., for the LMS implementation (54) we shall write instead

$$\tilde{\mathbf{x}}_i = (I_M - \mu R_u) \tilde{\mathbf{x}}_{i-1} + \mu \mathbf{s}_i(\mathbf{x}_{i-1}). \quad (55)$$

On the other hand, we conclude from (22)–(23) that the momentum LMS recursion will be given by:

$$\psi_{i-1} = \mathbf{w}_{i-1} + \beta_1(\mathbf{w}_{i-1} - \mathbf{w}_{i-2}), \quad (56)$$

$$\mathbf{w}_i = \psi_{i-1} + \mu_m \mathbf{u}_i (\mathbf{d}(i) - \mathbf{u}_i^\top \psi_{i-1}) + \beta_2(\psi_{i-1} - \psi_{i-2}), \quad (57)$$

Using the transformed recursion (29), we can transform the resulting relation for $\tilde{\mathbf{w}}_i$ into:

$$\begin{bmatrix} \tilde{\mathbf{w}}_i \\ \tilde{\mathbf{w}}_i \end{bmatrix} = \begin{bmatrix} I_M - \frac{\mu_m}{1-\beta} R_u & \frac{\mu_m \beta'}{1-\beta} R_u \\ -\frac{\mu_m}{1-\beta} R_u & \beta I_M + \frac{\mu_m \beta'}{1-\beta} R_u \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{w}}_{i-1} \\ \tilde{\mathbf{w}}_{i-1} \end{bmatrix} + \frac{\mu_m}{1-\beta} \begin{bmatrix} \mathbf{s}_i(\psi_{i-1}) \\ \mathbf{s}_i(\psi_{i-1}) \end{bmatrix}, \quad (58)$$

where the Hessian matrix, \mathbf{H}_{i-1} , is independent of the weight iterates and given by R_u for quadratic risks. It follows from the first row that

$$\hat{\mathbf{w}}_i = \left(I_M - \frac{\mu_m}{1-\beta} R_u \right) \hat{\mathbf{w}}_{i-1} + \frac{\mu_m \beta'}{1-\beta} R_u \tilde{\mathbf{w}}_{i-1} + \frac{\mu_m}{1-\beta} \mathbf{s}_i(\psi_{i-1}). \quad (59)$$

Next, we assume the step-sizes $\{\mu, \mu_m\}$ and the momentum parameter are selected to satisfy

$$\mu = \frac{\mu_m}{1-\beta}. \quad (60)$$

Since $\beta \in [0, 1)$, this means that $\mu_m < \mu$. Then, recursion (59) becomes

$$\hat{\mathbf{w}}_i = (I_M - \mu R_u) \hat{\mathbf{w}}_{i-1} + \mu \beta' R_u \tilde{\mathbf{w}}_{i-1} + \mu \mathbf{s}_i(\psi_{i-1}). \quad (61)$$

Comparing (61) with the LMS recursion (55), we find that both relations are quite similar, except that the momentum recursion has an extra driving term dependent on $\tilde{\mathbf{w}}_{i-1}$. However, recall from (28) that $\tilde{\mathbf{w}}_{i-1} = (\tilde{\mathbf{w}}_{i-2} - \tilde{\mathbf{w}}_{i-1})/(1-\beta)$, which is the difference between two consecutive points generated by momentum LMS. Intuitively, it is not hard to see that $\tilde{\mathbf{w}}_{i-1}$ is in the order of $O(\mu)$, which makes $\mu \beta' R_u \tilde{\mathbf{w}}_{i-1}$ in the order of $O(\mu^2)$. When the step-size μ is very small, this $O(\mu^2)$ term can be ignored. Consequently, the above recursions for $\hat{\mathbf{w}}_i$ and $\tilde{\mathbf{x}}_i$ should evolve close to each other, which would help to prove that \mathbf{w}_i and \mathbf{x}_i will also evolve close to each other as well. This conclusion can be established formally as follows, which proves the equivalence between the momentum and standard LMS methods.

Theorem 8 (Equivalence for LMS) *Consider the LMS and momentum LMS recursions (52) and (56)–(57). Let Assumptions 1, 2 and 4 hold. Assume both algorithms start from the same initial states, namely, $\psi_{-2} = \mathbf{w}_{-2} = \mathbf{x}_{-1}$. Suppose conditions (26) holds, and that the step-sizes $\{\mu, \mu_m\}$ satisfy (60). Then, it holds for sufficiently small μ that for $\forall i = 0, 1, 2, 3, \dots$*

$$\|\mathbf{w}_i - \mathbf{x}_i\|^2 = O\left(\left[\frac{\delta^2 + \gamma^2}{(1-\beta)^2} \rho_1^{i+1} + \frac{\delta^2 \sigma_s^2}{\nu^2(1-\beta)}\right] \mu^2 + \frac{\delta^2(\delta^2 + \gamma^2)(i+1)\rho_1^{i+1}}{\nu(1-\beta)^2} \mu^3\right). \quad (62)$$

where $\rho_1 = 1 - \frac{\mu^2}{2} \in (0, 1)$.

Proof See Appendix G. ■

Similar to Corollary 5 and 7, Theorem 8 also has two implications. First, it holds that

$$\|\mathbf{w}_i - \mathbf{x}_i\|^2 = O(\mu^2), \quad \forall i = 0, 1, 2, \dots \quad (63)$$

Besides, since $\rho_1^i \rightarrow 0$ and $i\rho_1^i$ as $i \rightarrow \infty$, we also conclude

$$\limsup_{i \rightarrow \infty} \|\mathbf{w}_i - \mathbf{x}_i\|^2 = O\left(\frac{\delta^2 \sigma_s^2 \mu^2}{\nu^2(1-\beta)}\right). \quad (64)$$

Theorem 8 establishes that the standard and momentum LMS algorithms are fundamentally equivalent since their iterates evolve close to each other at all times for sufficiently small step-sizes. More interpretation of this result is discussed in Section 5.2.

5. Equivalence in the General Case

We now extend the analysis from quadratic risks to more general risks (such as logistic risks). The analysis in this case is more demanding because the Hessian matrix of $J(\mathbf{w})$ is now w -dependent, but the same equivalence conclusion will continue to hold as we proceed to show.

5.1 Equivalence in the General Case

Note from the momentum recursion (29) that

$$\hat{\mathbf{w}}_i = \left(I_M - \frac{\mu_m}{1-\beta} \mathbf{H}_{i-1} \right) \hat{\mathbf{w}}_{i-1} + \frac{\mu_m \beta'}{1-\beta} \mathbf{H}_{i-1} \tilde{\mathbf{w}}_{i-1} + \frac{\mu_m}{1-\beta} \mathbf{s}_i(\psi_{i-1}), \quad (65)$$

where \mathbf{H}_{i-1} is defined by (31). In the quadratic case, this matrix was constant and equal to the covariance matrix, R_u . Here, however, it is time-variant and depends on the error vector, $\tilde{\psi}_{i-1}$, as well. Likewise, for the standard stochastic gradient iteration (2), we obtain that the error recursion in the general case is given by:

$$\tilde{\mathbf{x}}_i = (I_M - \mu \mathbf{R}_{i-1}) \tilde{\mathbf{x}}_{i-1} + \mu \mathbf{s}_i(\mathbf{x}_{i-1}), \quad (66)$$

where we are introducing the matrix

$$\mathbf{R}_{i-1} = \int_0^1 \nabla_w^2 J(w^\circ - \tau \tilde{\mathbf{x}}_{i-1}) d\tau \quad (67)$$

and $\tilde{\mathbf{x}}_i = w^o - \mathbf{x}_i$. Note that \mathbf{H}_{i-1} and \mathbf{R}_{i-1} are different matrices. In contrast, in the quadratic case, they are both equal to R_w .

Under the assumed condition (60) relating $\{\mu, \mu_m\}$, if we subtract (66) from (65) we obtain:

$$\begin{aligned} \hat{\mathbf{w}}_i - \tilde{\mathbf{x}}_i &= (I_M - \mu \mathbf{H}_{i-1})(\hat{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}) + \mu(\mathbf{R}_{i-1} - \mathbf{H}_{i-1})\tilde{\mathbf{x}}_{i-1} \\ &\quad + \mu^\beta \mathbf{H}_{i-1} \hat{\mathbf{w}}_{i-1} + \mu \mathbf{s}_i(\psi_{i-1}) - \mathbf{s}_i(\mathbf{x}_{i-1}). \end{aligned} \quad (68)$$

In the quadratic case, the second term on the right-hand side is zero since $\mathbf{R}_{i-1} = \mathbf{H}_{i-1} = R_w$. It is the presence of this term that makes the analysis more demanding in the general case.

To examine how close $\hat{\mathbf{w}}_i$ gets to $\tilde{\mathbf{x}}_i$ for each iteration, we start by noting that

$$\begin{aligned} \mathbb{E}\|\hat{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 &= \mathbb{E}\|(I_M - \mu \mathbf{H}_{i-1})(\hat{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}) + \mu(\mathbf{R}_{i-1} - \mathbf{H}_{i-1})\tilde{\mathbf{x}}_{i-1} + \mu^\beta \mathbf{H}_{i-1} \hat{\mathbf{w}}_{i-1} \\ &\quad + \mu^\beta \mathbb{E}\|\mathbf{s}_i(\psi_{i-1}) - \mathbf{s}_i(\mathbf{x}_{i-1})\|^2. \end{aligned} \quad (69)$$

Now, applying a similar derivation to the one used to arrive at (137) in Appendix C, and the inequality $\|a + b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$, we can conclude from (69) that

$$\begin{aligned} \mathbb{E}\|\hat{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 &\leq (1 - \mu\nu)\mathbb{E}\|\hat{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2 + \frac{2\mu\beta^2\delta^2}{\nu}\mathbb{E}\|\hat{\mathbf{w}}_{i-1}\|^2 \\ &\quad + \frac{2\mu\mathbb{E}\|(\mathbf{R}_{i-1} - \mathbf{H}_{i-1})\tilde{\mathbf{x}}_{i-1}\|^2 + \mu^2\mathbb{E}\|\mathbf{s}_i(\psi_{i-1}) - \mathbf{s}_i(\mathbf{x}_{i-1})\|^2}{\nu}. \end{aligned} \quad (70)$$

Using the Cauchy-Schwartz inequality we can bound the cross term as

$$\mathbb{E}\|(\mathbf{R}_{i-1} - \mathbf{H}_{i-1})\tilde{\mathbf{x}}_{i-1}\|^2 \leq \mathbb{E}(\|\mathbf{R}_{i-1} - \mathbf{H}_{i-1}\|^2 \|\tilde{\mathbf{x}}_{i-1}\|^2) \leq \sqrt{\mathbb{E}\|\mathbf{R}_{i-1} - \mathbf{H}_{i-1}\|^4 \mathbb{E}\|\tilde{\mathbf{x}}_{i-1}\|^4}. \quad (71)$$

In the above inequality, the term $\mathbb{E}\|\tilde{\mathbf{x}}_{i-1}\|^4$ can be bounded by using the result of Lemma 2. Therefore, we focus on bounding $\mathbb{E}\|\mathbf{R}_{i-1} - \mathbf{H}_{i-1}\|^4$ next. To do so, we need to introduce the following smoothness assumptions on the second and fourth-order moments of the gradient noise process and on the Hessian matrix of the risk function. These assumptions hold automatically for important cases of interest, such as least-mean-squares and logistic regression problems — see Appendix H for the verification.

Assumption 5 Consider the iterates ψ_{i-1} and \mathbf{x}_{i-1} that are generated by the momentum recursion (22) and the stochastic gradient recursion (2). It is assumed that the gradient noise process satisfies:

$$\mathbb{E}\|\mathbf{s}_i(\psi_{i-1}) - \mathbf{s}_i(\mathbf{x}_{i-1})\|^2 | \mathcal{F}_{i-1} \leq \xi_1 \|\psi_{i-1} - \mathbf{x}_{i-1}\|^2, \quad (72)$$

$$\mathbb{E}\|\mathbf{s}_i(\psi_{i-1}) - \mathbf{s}_i(\mathbf{x}_{i-1})\|^4 | \mathcal{F}_{i-1} \leq \xi_2 \|\psi_{i-1} - \mathbf{x}_{i-1}\|^4. \quad (73)$$

for some constants ξ_1 and ξ_2 .

Assumption 6 The Hessian of the risk function $J(w)$ in (1) is Lipschitz continuous, i.e., for any two variables $w_1, w_2 \in \text{dom } J(w)$, it holds that

$$\|\nabla_w^2 J(w_1) - \nabla_w^2 J(w_2)\| \leq \kappa \|w_1 - w_2\|. \quad (74)$$

for some constant $\kappa \geq 0$.

Using these assumptions, we can now establish two auxiliary results in preparation for the main equivalence theorem in the general case.

Lemma 9 (Uniform bound) Consider the standard and momentum stochastic gradient recursions (2) and (22)-(23) and assume they start from the same initial states, namely, $\psi_{-2} = \mathbf{w}_{-2} = \mathbf{x}_{-1}$. We continue to assume conditions (26), and (60). Under Assumptions 1, 3, 4, 5 and for sufficiently small step-sizes μ , the following result holds:

$$\mathbb{E}\|\tilde{\psi}_i - \tilde{\mathbf{x}}_i\|^4 = O\left(\frac{\delta^4(i+1)^{\delta+1}\mu}{\nu^3} + \frac{\delta^4\sigma_s^4}{\nu^6}\mu^2\right), \quad (75)$$

where $\rho_2 = 1 - \mu\nu/4$.

Proof See Appendix I. ■

Although sufficient for our purposes, we remark that the bound (75) for $\mathbb{E}\|\tilde{\psi}_i - \tilde{\mathbf{x}}_i\|^4$ is not tight. The reason is that in the derivation in Appendix I we employed a looser bound for the term $\mathbb{E}\|(\mathbf{R}_{i-1} - \mathbf{H}_{i-1})\tilde{\mathbf{x}}_{i-1}\|^4$ in order to avoid the appearance of higher-order powers, such as $\mathbb{E}\|\mathbf{R}_{i-1} - \mathbf{H}_{i-1}\|^8$ and $\mathbb{E}\|\tilde{\mathbf{x}}_{i-1}\|^8$. To avoid this possibility, we employed the following bound (using (11) to bound $\|\mathbf{R}_{i-1}\|^4$ and $\|\mathbf{H}_{i-1}\|^4$ and the inequality $\|a + b\|^4 \leq 8\|a\|^4 + 8\|b\|^4$):

$$\begin{aligned} \mathbb{E}\|(\mathbf{R}_{i-1} - \mathbf{H}_{i-1})\tilde{\mathbf{x}}_{i-1}\|^4 &\leq \mathbb{E}\|\mathbf{R}_{i-1} - \mathbf{H}_{i-1}\|^4 \|\tilde{\mathbf{x}}_{i-1}\|^4 \\ &\leq 8\mathbb{E}\{(\|\mathbf{R}_{i-1}\|^4 + \|\mathbf{H}_{i-1}\|^4)\|\tilde{\mathbf{x}}_{i-1}\|^4\} \leq 16\delta^4\mathbb{E}\|\tilde{\mathbf{x}}_{i-1}\|^4. \end{aligned} \quad (76)$$

Based on Lemma 9, we can now bound $\mathbb{E}\|\mathbf{R}_{i-1} - \mathbf{H}_{i-1}\|^4$, which is what the following lemma states.

Lemma 10 (Bound on Hessian difference) Consider the same setting of Lemma 9. Under Assumptions 1, 3, 4, 6 and for sufficiently small step-sizes μ , the following two result holds:

$$\mathbb{E}\|\mathbf{R}_{i-1} - \mathbf{H}_{i-1}\|^4 = O\left(\frac{\delta^4 i \rho_2^5 \mu}{\nu^3} + \frac{\delta^4 \sigma_s^4}{\nu^6} \mu^2\right), \quad (77)$$

where $\rho_2 = 1 - \mu\nu/4$.

Proof See Appendix J. ■

With the upper bounds of $\mathbb{E}\|\mathbf{R}_{i-1} - \mathbf{H}_{i-1}\|^4$ and $\mathbb{E}\|\tilde{\mathbf{x}}_{i-1}\|^4$ established in Lemma 10 and Lemma 2 respectively, we are able to bound $\|(\mathbf{R}_{i-1} - \mathbf{H}_{i-1})\tilde{\mathbf{x}}_{i-1}\|$ in (71), which in turn helps to establish the main equivalence result.

Theorem 11 (Equivalence for general risks) Consider the standard and momentum stochastic gradient recursions (2) and (22)-(23) and assume they start from the same initial states, namely, $\psi_{-2} = \mathbf{w}_{-2} = \mathbf{x}_{-1}$. Suppose conditions (26) and (60) hold. Under

Assumptions 1, 3, 4, 5, and 6, and for sufficiently small step-size μ , it holds that

$$\mathbb{E}\|\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 = O\left(\frac{\delta^2 \sigma_s^2 \tau_2^{i+1} \mu^{3/2}}{(1-\beta)^{\nu^2/2}} + \left[\frac{\delta^2 + \gamma^2}{(1-\beta)^2} \beta_1^{i+1} + \frac{\delta^2 \sigma_s^4}{(1-\beta)^2 \nu^2}\right] \mu^2\right), \quad \forall i = 0, 1, 2, 3, \dots \quad (78)$$

where $\rho_1 = 1 - \frac{\mu^2}{2} \in (0, 1)$ and $\tau_2 \triangleq \sqrt{1 - \mu\nu/4} \in (0, 1)$.

Proof See Appendix K. ■

Similar to Corollary 5, 7 and Theorem 8, Theorem 11 implies that

$$\mathbb{E}\|\mathbf{w}_i - \mathbf{x}_i\|^2 = O(\mu^{3/2}), \quad \forall i = 0, 1, 2, \dots \quad (79)$$

Besides, since $\rho_1^i \rightarrow 0$ and $i^2 \tau_2^i \rightarrow 0$ as $i \rightarrow \infty$, we will also conclude

$$\limsup_{i \rightarrow \infty} \mathbb{E}\|\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 = O\left(\frac{\delta^2 \sigma_s^4 \mu^2}{(1-\beta)^{\nu^2}}\right) = O(\mu^2). \quad (80)$$

Remark When we refer to ‘‘sufficiently small step-sizes’’ in Theorems 8 and 11, we mean that step-sizes are smaller than the stability bound, and are also small enough to ensure a desirable level of mean-square-error based on the performance expressions.

5.2 Interpretation of Equivalence Result

The result of Theorem 11 shows that, for sufficiently small step-sizes, the trajectories of momentum and standard stochastic gradient methods remain within $O(\mu^{3/2})$ from each other for *every* i (for quadratic cases the trajectories will remain within $O(\mu^2)$ as stated in Theorem 8). This means that these trajectories evolve together for all practical purposes and, hence, we shall say that the two implementations are ‘‘equivalent’’ (meaning that their trajectories remain close to each other in the mean-square-error sense).

A second useful insight from Theorem 8 is that the momentum method is essentially equivalent to running a standard stochastic gradient method with a larger step-size (since $\mu > \mu_{\text{opt}}$). This interpretation explains why the momentum method is observed to converge faster during the transient phase albeit towards a worse MSD level in steady-state than the standard method. This is because, as is well-known in the adaptive filtering literature (Sayed, 2008, 2014a) that larger step-sizes for stochastic gradient method do indeed lead to faster convergence but worse limiting performance.

In addition, Theorem 11 enables us to compute the steady-state MSD performance of the momentum stochastic gradient method. It is guaranteed by Theorem 11 that momentum method is equivalent to standard stochastic gradient method with larger step-size, $\mu = \mu_{\text{opt}}/(1-\beta)$. Therefore, once we compute the MSD performance of the standard stochastic gradient, according to (Haykin, 2008; Sayed, 2008, 2014a), we will also know the MSD performance for the momentum method.

Another consequence of the equivalence result is that any benefits that would be expected from a momentum stochastic gradient descent can be attained by simply using a

standard stochastic gradient implementation with a larger step-size; this is achieved without the additional computational or memory burden that the momentum method entails.

Besides the theoretical analysis given above, there is an intuitive explanation as to why the momentum variant leads to worse steady-state performance. While the momentum terms $\mathbf{w}_i - \mathbf{w}_{i-1}$ and $\psi_i - \psi_{i-1}$ can smooth the convergence trajectories, and hence accelerate the convergence rate, they nevertheless introduce additional noise into the evolution of the algorithm because all iterates \mathbf{w}_i and ψ_i are distorted by perturbations. This fact illustrates the essential difference between stochastic methods with constant step-sizes, and stochastic or deterministic methods with decaying step-sizes: in the former case, the presence of gradient noise essentially eliminates the benefits of the momentum term.

5.3 Stochastic Gradient Method with Diminishing Momentum

(Tygert, 2016; Yuan et al., 2016) suggest one useful technique to retain the advantages of the momentum implementation by employing a *diminishing* momentum parameter, $\beta(i)$, and by ensuring $\beta(i) \rightarrow 0$ in order not to degrade the limiting performance of the implementation. By doing so, the momentum term will help accelerate the convergence rate during the transient phase because it will smooth the trajectory (Nedjic and Bertsekas, 2001; Xiao, 2010; Lan, 2012). On the other hand, momentum will not cause degradation in MSD performance because the momentum effect would have died before the algorithm reaches state-state.

According to (Tygert, 2016; Yuan et al., 2016), we adapt the momentum stochastic method into the following algorithm

$$\begin{aligned} \psi_{i-1} &= \mathbf{w}_{i-1} + \beta_1(i)(\mathbf{w}_{i-1} - \mathbf{w}_{i-2}), \\ \mathbf{w}_i &= \psi_{i-1} - \mu \nabla_w Q(\psi_{i-1}; \boldsymbol{\theta}_i) + \beta_2(i)(\psi_{i-1} - \psi_{i-2}), \end{aligned} \quad (81)$$

$$(82)$$

with the same initial conditions as in (24)–(25). Similar to condition (26), $\beta_1(i)$ and $\beta_2(i)$ also need to satisfy

$$\beta_1(i) + \beta_2(i) = \beta(i), \quad \beta_1(i)\beta_2(i) = 0, \quad (83)$$

The efficacy of (81)–(82) will depend on how the momentum decay, $\beta(i)$, is selected. A satisfactory sequence $\{\beta(i)\}$ should decay slowly during the initial stages of adaptation so that the momentum term can induce an acceleration effect. However, the sequence $\{\beta(i)\}$ should also decrease drastically prior to steady-state so that the vanishing momentum term will not introduce additional gradient noise and degrade performance. One strategy, which is also employed in the numerical experiments in Section 7, is to design $\beta(i)$ to decrease in a stair-wise fashion, namely,

$$\beta(i) = \begin{cases} \beta_0 & \text{if } i \in [1, T], \\ \beta_0/T^\alpha & \text{if } i \in [T+1, 2T], \\ \beta_0/(2T)^\alpha & \text{if } i \in [2T+1, 3T], \\ \beta_0/(3T)^\alpha & \text{if } i \in [3T+1, 4T], \\ \dots & \dots \end{cases} \quad (84)$$

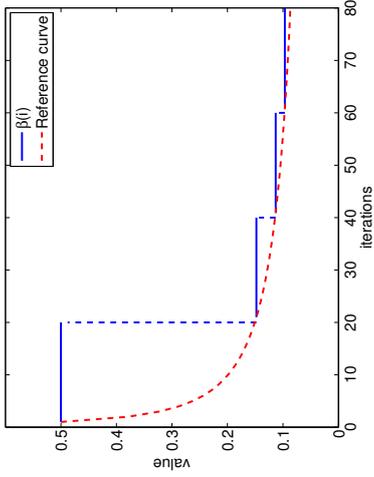


Figure 1: $\beta(i)$ changes with iteration i according to (84), where $\beta_0 = 0.5$, $T = 20$ and $\alpha = 0.4$. The reference curve is $f(i) = 0.5/t^{0.4}$.

where the constants $\beta_0 \in [0, 1]$, $\alpha \in (0, 1)$ and $T > 0$ determines the width of the stair steps. Fig. 1 illustrates how $\beta(i)$ varies when $T = 20$, $\beta_0 = 0.5$ and $\alpha = 0.4$.

Algorithm (81)–(82) works well when $\beta(i)$ decreases according to (84) (see Section 7). However, with Theorems 8 and 11, we find that this algorithm is essentially equivalent to the standard stochastic gradient method with decaying step-size, i.e.,

$$\mathbf{x}_i = \mathbf{x}_{i-1} - \mu_s(i) \nabla_w Q(\mathbf{x}_{i-1}; \boldsymbol{\theta}_i), \quad (85)$$

where

$$\mu_s(i) = \frac{\mu}{1 - \beta(i)} \quad (86)$$

will decrease from $\mu/[1 - \beta(0)]$ to μ . In another words, the stochastic algorithm with decaying momentum is still not helpful.

6. Diagonal Step-size Matrices

Sometimes it is advantageous to employ separate step-size for the individual entries of the weight vectors, see (Duchi et al., 2011). In this section we comment on how the results from the previous sections extend to this scenario. First, we note that recursion (2) can be generalized to the following form, with a diagonal matrix serving as the step-size parameter:

$$\mathbf{x}_i = \mathbf{x}_{i-1} - D \nabla_w Q(\mathbf{x}_{i-1}; \boldsymbol{\theta}_i), \quad i \geq 0, \quad (87)$$

where $D = \text{diag}\{\mu_1, \mu_2, \dots, \mu_M\}$. Here, we continue to use the letter “ \mathbf{x} ” to refer to the variable iterates for the standard stochastic gradient descent iteration, while we reserve

the letter “ \mathbf{w} ” for the momentum recursion. We let $\mu_{\max} = \max\{\mu_1, \dots, \mu_M\}$. Similarly, recursions (22) and (23) can be extended in the following manner:

$$\psi_{i-1} = \mathbf{w}_{i-1} + B_1(\mathbf{w}_{i-1} - \mathbf{w}_{i-2}), \quad (88)$$

$$\mathbf{w}_i = \psi_{i-1} - D_m \nabla_w Q(\psi_{i-1}; \boldsymbol{\theta}_i) + B_2(\psi_{i-1} - \psi_{i-2}), \quad (89)$$

with initial conditions

$$\mathbf{w}_{-2} = \psi_{-2} = \text{initial states}, \quad (90)$$

$$\mathbf{w}_{-1} = \mathbf{w}_{-2} - D_m \nabla_w Q(\mathbf{w}_{-2}; \boldsymbol{\theta}_{-1}), \quad (91)$$

where $B_1 = \text{diag}\{\beta_1^1, \dots, \beta_M^1\}$ and $B_2 = \text{diag}\{\beta_1^2, \dots, \beta_M^2\}$ are momentum coefficient matrices, while D_m is a diagonal step-size matrix for momentum stochastic gradient method. In a manner similar to (26), we also assume that

$$0 \leq B_k < I_M, \quad k = 1, 2, \quad B_1 + B_2 = B, \quad B_1 B_2 = 0. \quad (92)$$

where $B = \text{diag}\{\beta_1, \dots, \beta_M\}$ and $0 < B < I_M$. In addition, we further assume that B is not too close to I_M , i.e.

$$B \leq (1 - \epsilon)I_M, \quad \text{for some constant } \epsilon > 0. \quad (93)$$

The following results extend Theorems 1, 3, and 4 and they can be established following similar derivations.

Theorem 1B (Mean-square stability). *Let Assumptions 1 and 2 hold and recall conditions (92) and (93). Then, for the momentum stochastic gradient method (88)–(89), it holds under sufficiently small step-size μ_{\max} that*

$$\limsup_{i \rightarrow \infty} \mathbb{E} \|\tilde{\mathbf{w}}_i\|^2 = O(\mu_{\max}). \quad (94)$$

■ **Theorem 3B (Equivalence for quadratic costs).** *Consider recursions (52) and (56)–(57) with $\{\mu, \mu_m, \beta_1, \beta_2\}$ replaced by $\{D, D_m, B_1, B_2\}$. Assume they start from the same initial states, namely, $\psi_{-2} = \mathbf{w}_{-2} = \mathbf{x}_{-1}$. Suppose further that conditions (92) and (93) hold, and that the step-sizes matrices $\{D, D_m\}$ satisfy a relation similar to (60), namely,*

$$D = (I - B)^{-1} D_m. \quad (95)$$

Then, it holds under sufficiently small μ_{\max} , that

$$\mathbb{E} \|\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 = O(\mu_{\max}^2), \quad \forall i = 0, 1, 2, 3, \dots \quad (96)$$

■ **Theorem 4B (Equivalence for general costs).** *Consider the stochastic gradient recursion (87) and the momentum stochastic gradient recursions (88)–(89) to solve the general problem (1). Assume they start from the same initial states, namely, $\psi_{-2} = \mathbf{w}_{-2} = \mathbf{x}_{-1}$.*

Suppose conditions (92), (93), and (95) hold. Under Assumptions 1, 3, 5, and 6, and for sufficiently small step-sizes, it holds that

$$\mathbb{E}\|\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 = O(\mu_{\max}^{3/2}), \quad \forall i = 0, 1, 2, 3, \dots \quad (97)$$

Furthermore, in the limit,

$$\limsup_{i \rightarrow \infty} \mathbb{E}\|\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 = O(\mu_{\max}^2). \quad (98)$$

7. Experimental Results

In this section we illustrate the main conclusions by means of computer simulations for both cases of mean-square-error designs and logistic designs. We also run simulations for algorithm (81)–(82) and verify its advantages in the stochastic context.

7.1 Least Mean-Squares Error Designs

We apply the standard LMS algorithm to (47). To do so, we generate data according to the linear regression model (48), where $\mathbf{w}^o \in \mathbb{R}^{10}$ is chosen randomly, and $\mathbf{u}_i \in \mathbb{R}^{10}$ is i.i.d and follows $\mathbf{u}_i \sim \mathcal{N}(0, \Lambda)$ where $\Lambda \in \mathbb{R}^{10 \times 10}$ is randomly-generated diagonal matrix with positive diagonal entries. Besides, $\mathbf{v}(i)$ is also i.i.d and follows $\mathbf{v}(i) \sim \mathcal{N}(0, \sigma_v^2 \mathbf{I}_{10})$, where $\sigma_v^2 = 0.01$. All results are averaged over 300 random trials. For each trial we generated 800 samples of \mathbf{u}_i , $\mathbf{v}(i)$ and $\mathbf{d}(i)$.

We first compare the standard and momentum LMS algorithms using $\mu = \mu_m = 0.003$. The momentum parameter β is set as 0.9. Furthermore, we employ the heavy-ball option for the momentum LMS, i.e., $\beta_1 = 0, \beta_2 = \beta$. Both the standard and momentum LMS methods are illustrated in the left plot in Fig. 2 with blue and red curves, respectively. It is seen that the momentum LMS converges faster, but the MSD performance is much worse. Next we set $\mu_m = \mu(1 - \beta) = 0.0003$ and illustrate this case with the magenta curve. It is observed that the magenta and blue curves are almost indistinguishable, which confirms the equivalence predicted by Theorem 8 for all time instants. We also illustrate an implementation with a decaying momentum parameter $\beta(i)$ by the green curve. In this simulation, we set $\mu_m = 0.003$ and make $\beta(i)$ decrease in a stair-wise fashion: when $i \in [1, 100]$, $\beta(i) = 0.9$; when $i \in [101, 200]$, $\beta(i) = 0.9/(100^{0.3})$; \dots ; when $i \in [2401, 2500]$, $\beta(i) = 0.9/(2400^{0.3})$. With this decaying $\beta(i)$, it is seen that the momentum LMS method recovers its faster convergence rate and attains the same steady-state MSD performance as the LMS implementation. Finally, we also implemented the standard LMS with initial step-size $\mu = 0.003$ and then decrease it gradually according to $\mu_s(i) = \mu/[1 - \beta(i)]$. As implied by Theorem 8, it is observed that the green and black curves are also almost indistinguishable, which confirms that the LMS algorithm with decaying momentum is still equivalent to the standard LMS with appropriately chosen decaying step-sizes. We also compared the standard and momentum LMS algorithms when $\mu = \mu_m = 0.003$ and β is set as 0.5, 0.6, 0.7, 0.8, and the same performance as the left plot in Fig. 2 is observed. To save space, we show the right plot in Fig. 2 in which $\beta = 0.5$ and omit the figures when β is set as 0.6, 0.7, 0.8.

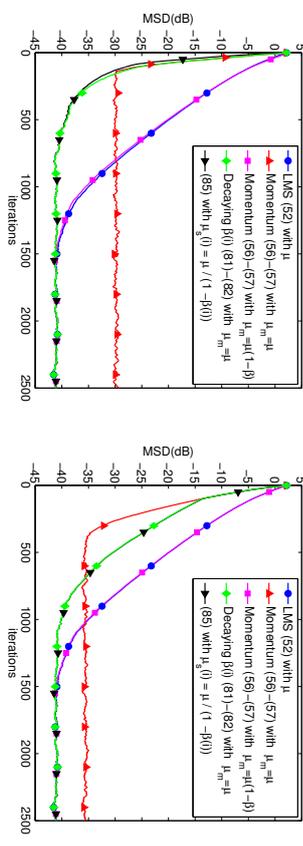


Figure 2: Convergence behavior of standard and momentum LMS (heavy-ball LMS) algorithms applied to the mean-square-error design problem (47) with $\beta = 0.9$ in the left plot and $\beta = 0.5$ in the right plot. Mean-square-deviation (MSD) means $\mathbb{E}\|\mathbf{w}^o - \mathbf{w}_i\|^2$.

Next we employ the Nesterov's acceleration option for the momentum LMS method, and compare it with standard LMS. The experimental settings are exactly the same as the above except that $\beta_1 = \beta$ and $\beta_2 = 0$. Both the standard and momentum LMS methods are illustrated in Fig. 3. As implied by Theorem 8, it is observed that Nesterov's acceleration applied to LMS is equivalent to standard LMS with rescaled step-size. Besides, by comparing Figs. 2 and 3, it is also observed that both momentum options, the heavy-ball and the Nesterov's acceleration, have the same performance. To save space, in the following experiments in Section 7.2–7.4 we just show the performance of momentum method with the option of heavy-ball.

7.2 Regularized Logistic Regression

We next consider a regularized logistic regression risk of the form:

$$J(\mathbf{w}) \triangleq \frac{\rho}{2} \|\mathbf{w}\|^2 + \mathbb{E} \left\{ \ln [1 + \exp(-\gamma(i) \mathbf{h}_i^T \mathbf{w})] \right\} \quad (99)$$

where the approximate gradient vector is chosen as

$$\nabla_{\mathbf{w}} Q(\mathbf{w}; \mathbf{h}_i, \gamma(i)) = \rho \mathbf{w} - \frac{\exp(-\gamma(i) \mathbf{h}_i^T \mathbf{w})}{1 + \exp(-\gamma(i) \mathbf{h}_i^T \mathbf{w})} \gamma(i) \mathbf{h}_i \quad (100)$$

In the simulation, we generate 20000 samples $(\mathbf{h}_i, \gamma(i))$. Among these training points, 10000 feature vectors \mathbf{h}_i correspond to label $\gamma(i) = 1$ and each $\mathbf{h}_i \sim \mathcal{N}(1.5 \times \mathbf{I}_{10}, R_h)$ for some diagonal covariance R_h . The remaining 10000 feature vectors \mathbf{h}_i correspond to label $\gamma(i) = -1$ and each $\mathbf{h}_i \sim \mathcal{N}(-1.5 \times \mathbf{I}_{10}, R_h)$. We set $\rho = 0.1$. The optimal solution \mathbf{w}^o is computed via the classic gradient descent method. All simulation results shown below are averaged over 300 trials.

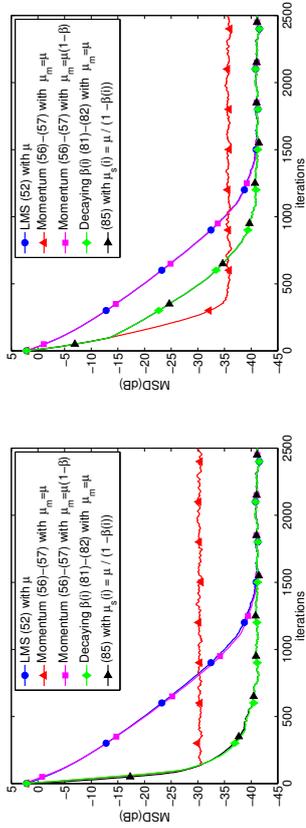


Figure 3: Convergence behavior of standard and momentum LMS (Nesterov’s acceleration LMS) algorithms applied to the mean-square-error design problem (47) with $\beta = 0.9$ in the left plot and $\beta = 0.5$ in the right plot.

Similar to the least-mean-squares error problem, we first compare the standard and momentum stochastic methods using $\mu = \mu_m = 0.005$. The momentum parameter β is set to 0.9. These two methods are illustrated in Fig. 4 with blue and red curves, respectively. It is seen that the momentum method converges faster, but the MSD performance is much worse. Next we set $\mu_m = \mu(1 - \beta) = 0.0005$ and illustrate this case with the magenta curve. It is observed that the magenta and blue curves are indistinguishable, which confirms the equivalence predicted by Theorem 11 for all time instants. Again we illustrate an implementation with a decaying momentum parameter $\beta(i)$ by the green curve. In this simulation, we set $\mu_m = 0.005$ and make $\beta(i)$ decrease in a stair-wise manner: when $i \in [1, 200]$, $\beta(i) = 0.9$; when $i \in [201, 400]$, $\beta(i) = 0.9/(200^{0.3})$; when $i \in [401, 600]$, $\beta(i) = 0.9/(400^{0.3})$; ...; when $i \in [1801, 2000]$, $\beta(i) = 0.9/(1800^{0.3})$. With this decaying $\beta(i)$, it is seen that the momentum method recovers its faster convergence rate and attains the same steady-state MSD performance as the stochastic-gradient implementation. Finally, we implemented the standard stochastic gradient descent with initial step-size $\mu_m = \mu = 0.005$ and then decrease it gradually according to $\mu_s(i) = \mu/[1 - \beta(i)]$. As implied by Theorem 11, it is observed that the green and black curves are almost indistinguishable, which confirms that the algorithm with decaying momentum is still equivalent to the standard stochastic gradient descent with appropriately chosen decaying step-sizes.

Next, we test the standard and momentum stochastic methods for regularized logistic regression problem over a benchmark data set — the Adult Data Set². The aim of this dataset is to predict whether a person earns over \$50K a year based on census data such as age, workclass, education, race, etc. The set is divided into 6414 training data and 26147 test data, and each feature vector has 123 entries. In the simulation, we set $\mu = 0.1$, $\rho = 0.1$, and $\beta = 0.9$. To check the equivalence of the algorithms, we set $\mu_m = (1 - \beta)\mu = 0.01$. In Fig. 5,

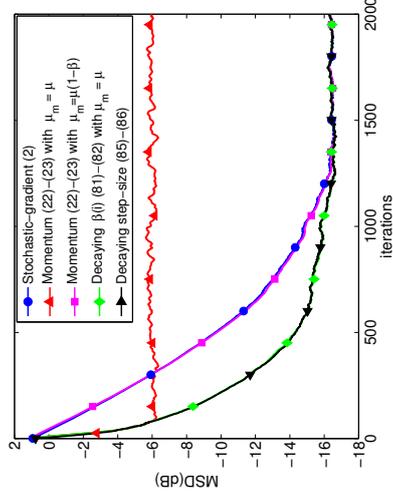


Figure 4: Convergence behaviors of standard and momentum stochastic gradient methods applied to the logistic regression problem (99).

the curve shows how the accuracy performance, i.e., the percentage of correct prediction, over the test dataset evolved as the algorithm received more training data³. The horizontal x-axis indicates the number of training data used. It is observed that the momentum and standard stochastic gradient methods cannot be distinguished, which confirms their equivalence when training the Adult Data Set.

For the experiments shown in this section, Section 7.3 and 7.4, we also tested the cases when β is set as 0.5, 0.6, 0.7, 0.8. Since the experimental results with different β are similar, we just plot the situation when $\beta = 0.9$, a setting which is usually employed in practice (Szegedy et al., 2015; Krizhevsky et al., 2012; Zhang and LeCun, 2015).

7.3 Further Verification of Theorems 8 and 11

In this section we further illustrate the conclusions of Theorems 8 and 11 by checking the behavior of the iterate difference, i.e., $\mathbb{E}\|\mathbf{w}_i - \mathbf{x}_i\|^2$, between the standard and momentum stochastic gradient methods.

For the least-mean-squares error problem, the selection of \mathbf{u}_i , $\mathbf{v}(i)$, $\mathbf{d}(i)$ and β is the same as in the simulation generated earlier in Subsection 7.1. For some specific step-size μ , \mathbf{x}_i is the iterate generated through LMS recursion (52) with step-size μ , and \mathbf{w}_i is the iterate generated momentum LMS recursion (56)–(57) with step-size $\mu_m = \mu(1 - \beta)$. Now we introduce the maximum difference:

$$d_{\max}(\mu) = \max_i \mathbb{E}\|\mathbf{w}_i - \mathbf{x}_i\|^2 \quad (101)$$

3. To smooth the performance curve, we applied the weighted average technique from equation (74) of (Ying and Sayed, 2015, 2016).

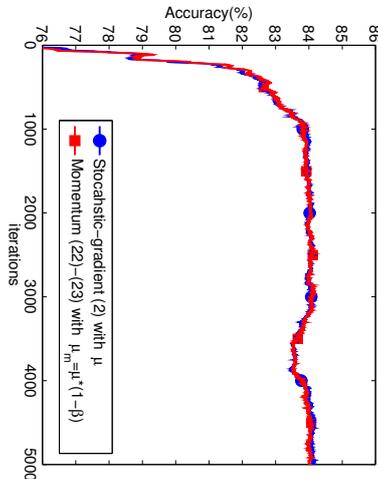


Figure 5: Performance accuracy of the standard and momentum stochastic gradient methods applied to logistic regression classification on the adult data test set.

and the difference at steady state

$$d_{ss}(\mu) = \limsup_{t \rightarrow \infty} \mathbb{E} \|\mathbf{w}_t - \mathbf{x}_2\|^2. \quad (102)$$

Note that both $d_{\max}(\mu)$ and $d_{ss}(\mu)$ are related with μ and we will examine how they vary according to different step-sizes. Obviously, since $\mathbb{E} \|\mathbf{w}_t - \mathbf{x}_2\|^2 \leq d_{\max}(\mu)$, if $d_{\max}(\mu)$ is illustrated to be on the order of $O(\mu^2)$, then it follows that $\mathbb{E} \|\mathbf{w}_t - \mathbf{x}_2\|^2 = O(\mu^2)$ for $t \geq 0$. Similarly, if we can illustrate $d_{ss}(\mu) = O(\mu^2)$, then it follows that $\limsup_{t \rightarrow \infty} \mathbb{E} \|\mathbf{w}_t - \mathbf{x}_2\|^2 = O(\mu^2)$.

Note that the fact $d_{\max}(\mu) = c\mu^2$ for some constant c holds if and only if

$$d_{\max}(\mu)(\text{dB}) = 20 \log \mu + 10 \log c, \quad (103)$$

where $d_{\max}(\mu)(\text{dB}) = 10 \log d_{\max}(\mu)$. Relation (103) can be confirmed with red circle line in Fig. 6. In this simulation, we choose 8 different step-size values $\{\mu_k\}_{k=1}^8$, and it can be verified that each data pair $(\log \mu_k, d_{\max}(\mu_k)(\text{dB}))$ satisfies relation (103). For example, in the red circle solid line, at $\mu_1 = 10^{-2}$ we read $d_{\max}(\mu_1)(\text{dB}) = -32\text{dB}$; while at $\mu_2 = 10^{-4}$ we read $d_{\max}(\mu_2)(\text{dB}) = -72\text{dB}$. It can be verified that

$$d_{\max}(\mu_1)(\text{dB}) - d_{\max}(\mu_2)(\text{dB}) = 20(\log \mu_1 - \log \mu_2) = 40. \quad (104)$$

Using a similar argument, the blue square solid line can also implies that $d_{ss} = O(\mu^2)$.

Figure 6 also reveals the order of d_{\max} and d_{ss} , with magenta and green dash lines respectively; for the regularized logistic regression problem from Subsection 7.2. With the same argument as above, $d_{ss}(\mu)$ can be confirmed on the order of $O(\mu^2)$. Now we check the order of $d_{\max}(\mu)$. The fact that $d_{\max}(\mu) = c\mu^{3/2}$ holds if and only if

$$d_{\max}(\mu)(\text{dB}) = 15 \log \mu + 10 \log c. \quad (105)$$

According to the above relation, at $\mu_1 = 10^{-2}$ and $\mu_2 = 10^{-4}$ we should have

$$d_{\max}(\mu_1)(\text{dB}) - d_{\max}(\mu_2)(\text{dB}) = 15(\log \mu_1 - \log \mu_2) = 30. \quad (106)$$

However, in the triangle magenta dash line we read $d_{\max}(\mu_1) = -30\text{dB}$ while $d_{\max}(\mu_2) = -66\text{dB}$ and hence

$$30\text{dB} < d_{\max}(\mu_1)(\text{dB}) - d_{\max}(\mu_2)(\text{dB}) < 40\text{dB}$$

Therefore, the order of d_{\max} should be between $O(\mu^{3/2})$ and $O(\mu^2)$, which still confirms Theorem 11.

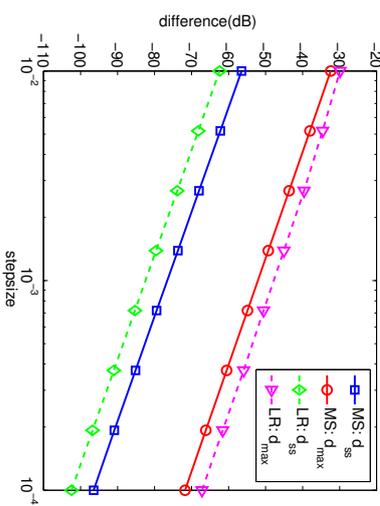


Figure 6: d_{\max} and d_{ss} as a function of the step-size μ . *MS* stands for mean-square-error and *LR* stands for logistic regression.

7.4 Visual Recognition

In this subsection we illustrate the conclusions of this work by re-examining the problem of training a neural network to recognize objects from images. We employ the CIFAR-10 database⁴, which is a classical benchmark dataset of images for visual recognition. The CIFAR-10 dataset consists of 60000 color images in 10 classes, each with 32×32 pixels. There are 50000 training images and 10000 test images. Similar to (Sutskever et al., 2013), and since the focus of this paper is on optimization, we only report training errors in our experiment.

To help illustrate that the conclusions also hold for non-differentiable and non-convex problems, in this experiment we train the data with two different neural network structures: (a) a 6-layer fully connected neural network and (b) a 4-layer convolutional neural network, both with ReLU activation functions. For each neural network, we will compare the performance of the momentum and standard stochastic gradient methods.

4. <https://www.cs.toronto.edu/~kriz/cifar.html>

6-Layer Fully Connected Neural Network. For this neural network structure, we employ the softmax measure with ℓ_2 regularization as a cost objective, and the ReLU as an activation function. Each hidden layer has 100 units, the coefficient of the ℓ_2 regularization term is set to 0.001, and the initial value w_{-1} is generated by a Gaussian distribution with 0.05 standard deviation. We employ mini-batch stochastic-gradient learning with batch size equal to 100. First, we apply a momentum backpropagation (i.e., momentum stochastic gradient) algorithm to train the 6-layer neural network. The momentum parameter is set to $\beta = 0.9$, and the initial step-size μ_m is set to 0.01. To achieve better accuracy, we follow a common technique (e.g., (Szegedy et al., 2015)) and reduce μ_m to $0.95\mu_m$ after every epoch. With the above settings, we attain an accuracy of about 90% in 80 epochs.

However, what is interesting, and somewhat surprising, is that the same 90% accuracy can also be achieved with the standard backpropagation (i.e., stochastic gradient descent) algorithm in 80 epochs. According to the step-size relation $\mu = \mu_m / (1 - \beta)$, we set the initial step-size μ of SGD to 0.1. Similar to the momentum method, we also reduce μ to 0.95μ after every epoch for SGD, and hence the relation $\mu = \mu_m / (1 - \beta)$ still holds for each iteration. From Figure 7, we observe that the accuracy performance curves for both scenarios, with and without momentum, are overlapping even when the overall risk is not necessarily convex or differentiable.

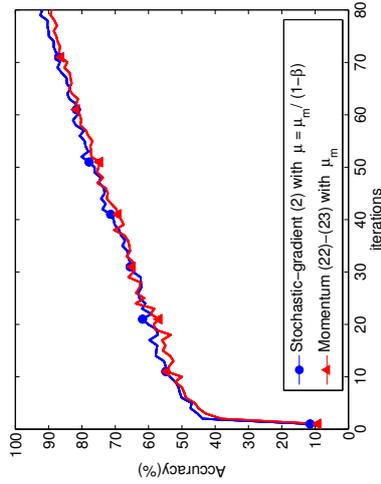


Figure 7: Classification accuracy of the standard and momentum stochastic gradient methods applied to a 6-layer fully-connected neural network on the CIFAR-10 test data set.

4-Layer Convolutional Neural Network. In a second experiment, we consider a 4-layer convolutional neural network. We employ the same objective and activation functions. This network has the structure:

$$(\text{conv} - \text{ReLU} - \text{pool}) \times 2 - (\text{affine} - \text{ReLU}) - \text{affine}$$

In the first convolutional layer, we use filters of size $7 \times 7 \times 3$, stride value 1, zero padding 3, and the number of these filters is 32. In the second convolutional layer, we use filters of size

$7 \times 7 \times 32$, stride value 1, zero padding 3, and the number of filters is still 32. We implement MAX operation in all pooling layers, and the pooling filters are of size 2×2 , stride value 2 and zero padding 0. The hidden layer has 500 units. The coefficient of the ℓ_2 regularization term is set to 0.001, and the initial value w_{-1} is generated by a Gaussian distribution with 0.001 standard deviation. We employ mini-batch stochastic-gradient learning with batch size equal to 50, and the step-size decreases by 5% after each epoch.

First, we apply the momentum backpropagation algorithm to train the neural network. The momentum parameter is set at $\beta = 0.9$, and we performed experiments with step-sizes $\mu_m \in \{0.01, 0.005, 0.001, 0.0005, 0.0001\}$ and find that $\mu_m = 0.001$ gives the highest training accuracy after 10 epochs. In Fig. 8 we draw the momentum stochastic gradient method with red curve when $\mu_m = 0.001$ and $\beta = 0.9$. The curve reaches an accuracy of 94%. Next we set the step-size of the standard backpropagation $\mu = \mu_m / (1 - \beta) = 0.01$, and illustrate its convergence performance with the blue curve. It is also observed that the two curves are indistinguishable. The numerical results shown in Figs. 7 and 8 imply that the performance of momentum SGD can still be achieved by standard SGD by properly adjusting the step-size according to $\mu = \mu_m / (1 - \beta)$.

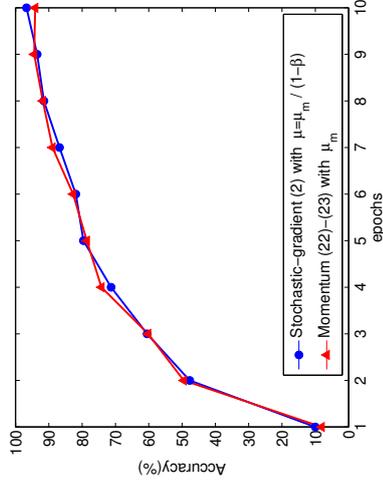


Figure 8: Classification accuracy of the standard and momentum stochastic gradient methods applied to a 4-layer convolutional neural network on the CIFAR-10 training data set.

8. Comparison for Larger Step-sizes

According to Theorem 11, the equivalence results between the standard and momentum stochastic gradient methods hold for sufficiently small step-sizes μ . When larger values for μ are used, the $O(\mu^{3/2})$ term is not negligible any longer so that the momentum and gradient-descent implementations are not equivalent anymore under these conditions. While in practical implementations small step-sizes are widely employed in order to ensure satisfactory steady-state MSD performance, one may still wonder how both algorithms would

compare to each other under larger step-sizes. For example, it is known that the larger the step-size value is, the more likely it is that the stochastic-gradient algorithm will become unstable. Does the addition of momentum help enlarge the stability range and allow for proper adaptation and learning over a wider range of step-sizes?

Unfortunately, the answer to the above question is generally negative. In fact, we can construct a simple numerical example in which the momentum can hurt the stability range. This example considers the case of quadratic risks, namely problems of the form (47). We suppose $M = 5$, $\mathbf{u}_i \sim \mathcal{N}(0, 0.5I_5)$ and $\mathbf{d}(i) = \mathbf{u}_i^T \mathbf{w}^o + \mathbf{v}(i)$ where $\mathbf{v}(i) \sim \mathcal{N}(0, 0.01)$. We compare the convergence of standard LMS and Nesterov's acceleration method with fixed parameter $\beta_2 = 0$ and $\beta = 0.5$. Both algorithms are set with the same step-size $\mu = \mu_m = 0.4$, which is a relatively large step-size. All results are averaged over 1000 random trials. For each trial we generated 200 samples of \mathbf{u}_i , $\mathbf{v}(i)$ and $\mathbf{d}(i)$. In Fig. 9, it shows that standard LMS converges at $\mu = 0.4$ while momentum LMS diverges, which indicates that momentum LMS has narrower stability range than standard LMS.

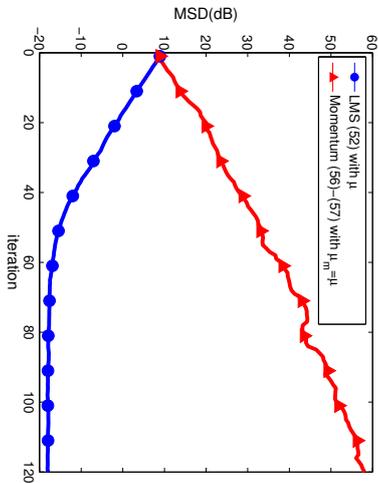


Figure 9: Convergence comparison between standard and momentum LMS algorithms when $\mu = \mu_m = 0.4$ and $\beta = 0.5$.

9. Conclusion

In this paper we analyzed the convergence and performance behavior of momentum stochastic gradient methods in the constant step-size and slow adaptation regime. The results establish that the momentum method is equivalent to employing the standard stochastic gradient method with a re-scaled (larger) step-size value. The size of the re-scaling is determined by the momentum parameter, β . The analysis was carried out under general conditions and was not limited to quadratic risks, but is also applicable to broader choices of the risk function. Overall, the conclusions indicate that the well-known benefits of momentum constructions in the deterministic optimization scenario do not necessarily carry over

to the stochastic setting when adaptation becomes necessary and gradient noise is present. The analysis also comments on a way to retain some of the advantages of the momentum construction by employing a decaying momentum parameter: one that starts at a constant level and decays to zero over time. adaptation is retained without the often-observed degradation in MSD performance.

Acknowledgments

This work was supported in part by NSF grants CIF-1524250 and ECCS-1407712, by DARPA project N66001-14-2-4029, and by a Visiting Professorship from the Leverhulme Trust, United Kingdom. The authors would like to thank PhD student Chung-Kai Yu for contributing to Section 5.3, and undergraduate student Gabrielle Robertson for contributing to the simulation in Section 7.4.

Appendix A. Proof of Lemma 2

It is shown in Eq. (3.76) of (Sayed, 2014a) that $\mathbb{E}\|\tilde{\mathbf{w}}_i\|^4$ evolves as follows:

$$\mathbb{E}\|\tilde{\mathbf{w}}_i\|^4 \leq (1 - \mu\nu)\mathbb{E}\|\tilde{\mathbf{w}}_{i-1}\|^4 + a_1\mu^2\mathbb{E}\|\tilde{\mathbf{w}}_{i-1}\|^2 + a_2\mu^4, \quad (107)$$

where the constants a_1 and a_2 are defined as

$$a_1 \triangleq 16\sigma_s^2, \quad a_2 \triangleq 3\sigma_s^4. \quad (108)$$

If we iterate (16) we find that

$$\mathbb{E}\|\tilde{\mathbf{w}}_i\|^2 \leq (1 - \mu\nu)^{i+1}\mathbb{E}\|\tilde{\mathbf{w}}_{-1}\|^2 + a_3\mu, \quad (109)$$

where a_3 is defined as

$$a_3 \triangleq \frac{\sigma_s^2}{\nu}. \quad (110)$$

Substituting inequality (109) into (107), we find that it holds for each iteration $i = 0, 1, 2, \dots$

$$\begin{aligned} \mathbb{E}\|\tilde{\mathbf{w}}_i\|^4 &\leq (1 - \mu\nu)\mathbb{E}\|\tilde{\mathbf{w}}_{i-1}\|^4 + a_2\mu^4 + a_1a_3\mu^3 + a_4\mu^2(1 - \mu\nu)^i, \\ &= \rho\mathbb{E}\|\tilde{\mathbf{w}}_{i-1}\|^4 + a_2\mu^4 + a_1a_3\mu^3 + a_4\mu^2\rho^i \end{aligned} \quad (111)$$

where

$$\rho \triangleq 1 - \mu\nu, \quad a_4 \triangleq a_1\mathbb{E}\|\tilde{\mathbf{w}}_{-1}\|^2. \quad (112)$$

Iterating the inequality (111) we get

$$\begin{aligned} \mathbb{E}\|\tilde{\mathbf{w}}_i\|^4 &\leq \rho^{i+1}\mathbb{E}\|\tilde{\mathbf{w}}_{-1}\|^4 + a_2\mu^4 \sum_{s=0}^i \rho^s + a_1a_3\mu^3 \sum_{s=0}^i \rho^s + a_4\mu^2(i+1)\rho^i \\ &\leq \rho^{i+1}\mathbb{E}\|\tilde{\mathbf{w}}_{-1}\|^4 + \frac{a_2\mu^4}{1-\rho} + \frac{a_1a_3\mu^3}{1-\rho} + a_4\mu^2(i+1)\rho^i \end{aligned}$$

$$\begin{aligned} &\leq \rho^{i+1} \mathbb{E} \|\tilde{\mathbf{w}}_{-1}\|^4 + a_5 \mu^3 + a_6 \mu^2 + a_4 \mu^2 (i+1) \rho^i \\ &\stackrel{(a)}{\leq} \rho^{i+1} \mathbb{E} \|\tilde{\mathbf{w}}_{-1}\|^4 + 2a_6 \mu^2 + a_4 \mu^2 (i+1) \rho^i, \end{aligned} \quad (113)$$

where

$$a_5 \triangleq \frac{a_2}{\nu}, \quad a_6 \triangleq \frac{a_1 a_3}{\nu}, \quad (114)$$

and (a) holds because for sufficiently small μ such that $a_6 \mu^2 > a_5 \mu^3$, we have

$$a_5 \mu^3 + a_6 \mu^2 = 2a_6 \mu^2 - (a_6 \mu^2 - a_5 \mu^3) \leq 2a_6 \mu^2. \quad (115)$$

Substituting (108), (110), (112) and (114) into (113), we get

$$\begin{aligned} \mathbb{E} \|\tilde{\mathbf{w}}_i\|^4 &\leq \rho^{i+1} \mathbb{E} \|\tilde{\mathbf{w}}_{-1}\|^4 + A_1 \sigma_s^2 (i+1) \rho^i \mu^2 + \frac{A_2 \sigma_s^4 \mu^2}{\nu^2} \\ &= \rho^{i+1} \mathbb{E} \|\tilde{\mathbf{w}}_{-1}\|^4 + \frac{A_1}{\rho} \sigma_s^2 (i+1) \rho^{i+1} \mu^2 + \frac{A_2 \sigma_s^4 \mu^2}{\nu^2} \end{aligned} \quad (116)$$

for some constants A_1 and A_2 . When μ is sufficiently small, there must exist some constant A_3 such that

$$\frac{A_1}{\rho} = \frac{A_1}{1 - \mu\nu} \leq A_3. \quad (117)$$

Therefore, (116) becomes

$$\mathbb{E} \|\tilde{\mathbf{w}}_i\|^4 \leq \rho^{i+1} \mathbb{E} \|\tilde{\mathbf{w}}_{-1}\|^4 + A_3 \sigma_s^2 (i+1) \rho^{i+1} \mu^2 + \frac{A_2 \sigma_s^4 \mu^2}{\nu^2}. \quad (118)$$

Appendix B. Proof of Lemma 3

We substitute the expression for the gradient noise from (12), evaluated at ψ_{i-1} , into (23) to get:

$$\mathbf{w}_i = \psi_{i-1} - \mu_m \nabla_w J(\psi_{i-1}) + \beta_2 (\psi_{i-1} - \psi_{i-2}) - \mu_m \mathbf{s}_i(\psi_{i-1}). \quad (119)$$

Let again $\tilde{\mathbf{w}}_i = \mathbf{w}^o - \mathbf{w}_i$ and $\tilde{\psi}_i = \mathbf{w}^o - \psi_i$. Subtracting both sides of (119) from \mathbf{w}^o gives:

$$\tilde{\mathbf{w}}_i = \tilde{\psi}_{i-1} + \mu_m \nabla_w J(\psi_{i-1}) - \beta_2 (\psi_{i-1} - \psi_{i-2}) + \mu_m \mathbf{s}_i(\psi_{i-1}). \quad (120)$$

We now appeal to the mean-value theorem (relation (D.9) in (Sayed, 2014a)) to write

$$\nabla J_w(\psi_{i-1}) = - \left(\int_0^1 \nabla_w^2 J_w(\mathbf{w}^o - t\tilde{\psi}_{i-1}) dt \right) \tilde{\psi}_{i-1} \triangleq -\mathbf{H}_{i-1} \tilde{\psi}_{i-1}. \quad (121)$$

and express the momentum term in the form

$$\psi_{i-1} - \psi_{i-2} = \psi_{i-1} - \mathbf{w}^o + \mathbf{w}^o - \psi_{i-2} = -\tilde{\psi}_{i-1} + \tilde{\psi}_{i-2}. \quad (122)$$

Then, expression (120) can be rewritten as

$$\tilde{\mathbf{w}}_i = (I_M + \beta_2 I_M - \mu_m \mathbf{H}_{i-1}) \tilde{\psi}_{i-1} - \beta_2 \tilde{\psi}_{i-2} + \mu_m \mathbf{s}_i(\psi_{i-1}). \quad (123)$$

On the other hand, expression (22) gives

$$\tilde{\psi}_{i-1} = \tilde{\mathbf{w}}_{i-1} + \beta_1 (\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{w}}_{i-2}). \quad (124)$$

Substituting (124) into (123), we have

$$\tilde{\mathbf{w}}_i = \mathbf{J}_{i-1} \tilde{\mathbf{w}}_{i-1} + \mathbf{K}_{i-1} \tilde{\mathbf{w}}_{i-2} + L \tilde{\mathbf{w}}_{i-3} + \mu_m \mathbf{s}_i(\psi_{i-1}), \quad (125)$$

where boldface quantities denote random variables:

$$\mathbf{J}_{i-1} = (1 + \beta_1)(1 + \beta_2) I_M - \mu_m (1 + \beta_1) \mathbf{H}_{i-1} \stackrel{(26)}{=} (1 + \beta) I_M - \mu_m (1 + \beta_1) \mathbf{H}_{i-1} \quad (126)$$

$$\mathbf{K}_{i-1} = -(\beta_1 + \beta_2 + 2\beta_1 \beta_2) I_M + \mu_m \beta_1 \mathbf{H}_{i-1} = -\beta I_M + \mu_m \beta_1 \mathbf{H}_{i-1} \quad (127)$$

$$L = \beta_1 \beta_2 = 0 \quad (128)$$

It follows that we can write the extended relation:

$$\begin{bmatrix} \tilde{\mathbf{w}}_i \\ \tilde{\mathbf{w}}_{i-1} \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{i-1} & \mathbf{K}_{i-1} \\ I_M & 0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{w}}_{i-1} \\ \tilde{\mathbf{w}}_{i-2} \end{bmatrix} + \mu_m \begin{bmatrix} \mathbf{s}_i(\psi_{i-1}) \\ 0 \end{bmatrix}. \quad (129)$$

$$\stackrel{\triangleq}{\mathbf{B}_{i-1}}$$

where we are denoting the coefficient matrix by \mathbf{B}_{i-1} , which can be written as the difference

$$\mathbf{B}_{i-1} \triangleq P - \mathbf{M}_{i-1}, \quad (130)$$

with

$$P = \begin{bmatrix} (1 + \beta) I_M & -\beta I_M \\ I_M & 0 \end{bmatrix}, \quad \mathbf{M}_{i-1} = \begin{bmatrix} \mu_m (1 + \beta_1) \mathbf{H}_{i-1} & -\mu_m \beta_1 \mathbf{H}_{i-1} \\ 0 & 0 \end{bmatrix}. \quad (131)$$

The eigenvalue decomposition of P can be easily seen to be given by $P = V D V^{-1}$, where

$$V = \begin{bmatrix} I_M & -\beta I_M \\ I_M & -I_M \end{bmatrix}, \quad V^{-1} = \frac{1}{1 - \beta} \begin{bmatrix} I_M & -\beta I_M \\ I_M & -I_M \end{bmatrix}, \quad D = \begin{bmatrix} I_M & 0 \\ 0 & \beta I_M \end{bmatrix}. \quad (132)$$

Therefore, we have

$$\mathbf{B}_{i-1} = V (D - V^{-1} \mathbf{M}_{i-1} V) V^{-1} = V \begin{bmatrix} I_M - \frac{\mu_m}{1 - \beta} \mathbf{H}_{i-1} & \frac{\mu_m \beta'}{1 - \beta} \mathbf{H}_{i-1} \\ -\frac{\mu_m}{1 - \beta} \mathbf{H}_{i-1} & \beta I_M + \frac{\mu_m \beta'}{1 - \beta} \mathbf{H}_{i-1} \end{bmatrix} V^{-1}, \quad (133)$$

where

$$\beta' \triangleq \beta \beta_1 + \beta - \beta_1 = \beta \beta_1 + \beta_2. \quad (134)$$

Multiplying both sides of (129) by V^{-1} from the left and recalling definition (28), we obtain

$$\begin{bmatrix} \hat{\mathbf{w}}_i \\ \hat{\mathbf{w}}_i \end{bmatrix} = \begin{bmatrix} I_M - \frac{\mu_m}{1 - \beta} \mathbf{H}_{i-1} & \frac{\mu_m \beta'}{1 - \beta} \mathbf{H}_{i-1} \\ -\frac{\mu_m}{1 - \beta} \mathbf{H}_{i-1} & \beta I_M + \frac{\mu_m \beta'}{1 - \beta} \mathbf{H}_{i-1} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{w}}_{i-1} \\ \hat{\mathbf{w}}_{i-1} \end{bmatrix} + \frac{\mu_m}{1 - \beta} \begin{bmatrix} \mathbf{s}_i(\psi_{i-1}) \\ \mathbf{s}_i(\psi_{i-1}) \end{bmatrix}. \quad (135)$$

Appendix C. Proof of Theorem 4

From the first row of recursion (29) we have

$$\hat{\mathbf{w}}_t = \left(I_M - \frac{\mu_m \mathbf{H}_{t-1}}{1-\beta} \right) \hat{\mathbf{w}}_{t-1} + \frac{\mu_m \beta' \mathbf{H}_{t-1}}{1-\beta} \hat{\mathbf{w}}_{t-1} + \frac{\mu_m}{1-\beta} \mathbf{s}_t(\psi_{t-1}). \quad (136)$$

Let $t \in (0, 1)$. Squaring both sides and taking expectations conditioned on \mathcal{F}_{t-1} , and using Jensen's inequality, we obtain under Assumptions 1 and 2:

$$\begin{aligned} & \mathbb{E} \|\hat{\mathbf{w}}_t\|^2 | \mathcal{F}_{t-1}] \\ &= \left\| \left(I_M - \frac{\mu_m}{1-\beta} \mathbf{H}_{t-1} \right) \hat{\mathbf{w}}_{t-1} + \frac{\mu_m \beta'}{1-\beta} \mathbf{H}_{t-1} \hat{\mathbf{w}}_{t-1} \right\|^2 + \frac{\mu_m^2}{(1-\beta)^2} \mathbb{E} \|\mathbf{s}_t(\psi_{t-1})\|^2 | \mathcal{F}_{t-1}] \\ &\stackrel{(a)}{\leq} \left\| (1-t) \frac{1}{1-t} \left(I_M - \frac{\mu_m}{1-\beta} \mathbf{H}_{t-1} \right) \hat{\mathbf{w}}_{t-1} + t \frac{1}{t} \frac{\mu_m \beta'}{1-\beta} \mathbf{H}_{t-1} \hat{\mathbf{w}}_{t-1} \right\|^2 + \frac{\mu_m^2}{(1-\beta)^2} (\gamma^2 \|\tilde{\psi}_{t-1}\|^2 + \sigma_s^2) \\ &\leq \frac{1}{1-t} \left\| \left(I_M - \frac{\mu_m}{1-\beta} \mathbf{H}_{t-1} \right) \hat{\mathbf{w}}_{t-1} \right\|^2 + \frac{1}{t} \left\| \frac{\mu_m \beta'}{1-\beta} \mathbf{H}_{t-1} \hat{\mathbf{w}}_{t-1} \right\|^2 + \frac{\mu_m^2}{(1-\beta)^2} (\gamma^2 \|\tilde{\psi}_{t-1}\|^2 + \sigma_s^2) \\ &\stackrel{(b)}{\leq} \frac{1}{1-t} \left(1 - \frac{\mu_m \nu}{1-\beta} \right)^2 \|\hat{\mathbf{w}}_{t-1}\|^2 + \frac{1}{t} \frac{\mu_m^2 \beta'^2 \delta^2}{(1-\beta)^2} \|\hat{\mathbf{w}}_{t-1}\|^2 + \frac{\mu_m^2}{(1-\beta)^2} (\gamma^2 \|\tilde{\psi}_{t-1}\|^2 + \sigma_s^2) \\ &\stackrel{(c)}{=} \left(1 - \frac{\mu_m \nu}{1-\beta} \right) \|\hat{\mathbf{w}}_{t-1}\|^2 + \frac{\mu_m \beta'^2 \delta^2}{\nu(1-\beta)} \|\hat{\mathbf{w}}_{t-1}\|^2 + \frac{\mu_m^2}{(1-\beta)^2} (\gamma^2 \|\tilde{\psi}_{t-1}\|^2 + \sigma_s^2). \end{aligned} \quad (137)$$

where (a) holds because of equation (14) in Assumption (2), (b) holds because $\nu I \leq \mathbf{H}_{t-1} \leq \delta I$ under Assumption (1), and (c) holds because we selected $t = \frac{\mu_m \nu}{1-\beta}$. Taking expectation again, we remove the conditioning to find:

$$\mathbb{E} \|\hat{\mathbf{w}}_t\|^2 \leq \left(1 - \frac{\mu_m \nu}{1-\beta} \right) \mathbb{E} \|\hat{\mathbf{w}}_{t-1}\|^2 + \frac{\mu_m^2}{\nu(1-\beta)^2} (\gamma^2 \mathbb{E} \|\tilde{\psi}_{t-1}\|^2 + \sigma_s^2). \quad (138)$$

Furthermore, squaring (124) and using the inequality $\|a + b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$ we get

$$\begin{aligned} \|\tilde{\mathbf{w}}_{t-1}\|^2 &\leq 2(1 + \beta_1)^2 \|\tilde{\mathbf{w}}_{t-1}\|^2 + 2\beta_1^2 \|\tilde{\mathbf{w}}_{t-2}\|^2 \leq 2(1 + \beta_1)^2 (\|\tilde{\mathbf{w}}_{t-1}\|^2 + \|\tilde{\mathbf{w}}_{t-2}\|^2) \\ &= 2(1 + \beta_1)^2 \left\| \begin{bmatrix} \tilde{\mathbf{w}}_{t-1} \\ \tilde{\mathbf{w}}_{t-2} \end{bmatrix} \right\|^2 = 2(1 + \beta_1)^2 \|V V^{-1} \begin{bmatrix} \tilde{\mathbf{w}}_{t-1} \\ \tilde{\mathbf{w}}_{t-2} \end{bmatrix}\|^2 \\ &\leq 2(1 + \beta_1)^2 \|V\|^2 \left\| \begin{bmatrix} \tilde{\mathbf{w}}_{t-1} \\ \tilde{\mathbf{w}}_{t-2} \end{bmatrix} \right\|^2. \end{aligned} \quad (139)$$

It is known that there exists some constant $d > 0$ such that $\|V\|^2 \leq d \|V\|_F^2$. From expression (27) for V we have

$$\|V\|_F^2 = 3\|I_M\|_F^2 + \beta^2 \|I_M\|_F^2 \leq 4\|I_M\|_F^2 = 4M.$$

Let $v^2 \triangleq 4dM$, so that $\|V\|^2 \leq v^2$. Therefore, under expectation, we conclude that it also holds:

$$\mathbb{E} \|\tilde{\psi}_{t-1}\|^2 \leq 2(1 + \beta_1)^2 v^2 (\mathbb{E} \|\hat{\mathbf{w}}_{t-1}\|^2 + \mathbb{E} \|\hat{\mathbf{w}}_{t-1}\|^2). \quad (140)$$

Substituting (140) into (138), we get

$$\begin{aligned} \mathbb{E} \|\hat{\mathbf{w}}_t\|^2 &\leq \left(1 - \frac{\mu_m \nu}{1-\beta} + \frac{2(1 + \beta_1)^2 \gamma^2 v^2}{(1-\beta)^2} \mu_m^2 \right) \mathbb{E} \|\hat{\mathbf{w}}_{t-1}\|^2 + \frac{\mu_m^2 \sigma_s^2}{(1-\beta)^2} \\ &\quad + \left(\frac{\mu_m \beta'^2 \delta^2}{\nu(1-\beta)} + \frac{2(1 + \beta_1)^2 \gamma^2 v^2}{(1-\beta)^2} \mu_m^2 \right) \mathbb{E} \|\hat{\mathbf{w}}_{t-1}\|^2. \end{aligned} \quad (141)$$

Now, let us consider the second row of (29), namely,

$$\hat{\mathbf{w}}_t = -\frac{\mu_m}{1-\beta} \mathbf{H}_{t-1} \hat{\mathbf{w}}_{t-1} + \left(\beta I_M + \frac{\mu_m \beta'}{1-\beta} \mathbf{H}_{t-1} \right) \hat{\mathbf{w}}_{t-1} + \frac{\mu_m}{1-\beta} \mathbf{s}_t(\psi_{t-1}). \quad (142)$$

As before, squaring and taking expectations of both sides, and using Jensen's inequality, we obtain under Assumptions 1 and 2:

$$\begin{aligned} & \mathbb{E} \|\hat{\mathbf{w}}_t\|^2 \\ &\leq \mathbb{E} \left\| \beta \hat{\mathbf{w}}_{t-1} + \left(\frac{\mu_m \beta'}{1-\beta} \mathbf{H}_{t-1} \hat{\mathbf{w}}_{t-1} - \frac{\mu_m}{1-\beta} \mathbf{H}_{t-1} \hat{\mathbf{w}}_{t-1} \right) \right\|^2 + \frac{\mu_m^2}{(1-\beta)^2} (\gamma^2 \mathbb{E} \|\tilde{\psi}_{t-1}\|^2 + \sigma_s^2) \\ &\stackrel{(a)}{\leq} \beta \mathbb{E} \|\hat{\mathbf{w}}_{t-1}\|^2 + \frac{1}{1-\beta} \mathbb{E} \left\| \frac{\mu_m \beta'}{1-\beta} \mathbf{H}_{t-1} \hat{\mathbf{w}}_{t-1} - \frac{\mu_m}{1-\beta} \mathbf{H}_{t-1} \hat{\mathbf{w}}_{t-1} \right\|^2 + \frac{\mu_m^2}{(1-\beta)^2} (\gamma^2 \mathbb{E} \|\tilde{\psi}_{t-1}\|^2 + \sigma_s^2) \\ &\leq \beta \mathbb{E} \|\hat{\mathbf{w}}_{t-1}\|^2 + \frac{2\mu_m^2 \beta'^2 \delta^2}{(1-\beta)^3} \mathbb{E} \|\hat{\mathbf{w}}_{t-1}\|^2 + \frac{2\mu_m^2 \delta^2}{(1-\beta)^3} \mathbb{E} \|\hat{\mathbf{w}}_{t-1}\|^2 + \frac{\mu_m^2}{(1-\beta)^2} (\gamma^2 \mathbb{E} \|\tilde{\psi}_{t-1}\|^2 + \sigma_s^2) \\ &= \left(\beta + \frac{2\mu_m^2 \beta'^2 \delta^2}{(1-\beta)^3} \right) \mathbb{E} \|\hat{\mathbf{w}}_{t-1}\|^2 + \frac{2\mu_m^2 \delta^2}{(1-\beta)^3} \mathbb{E} \|\hat{\mathbf{w}}_{t-1}\|^2 + \frac{\mu_m^2}{(1-\beta)^2} (\gamma^2 \mathbb{E} \|\tilde{\psi}_{t-1}\|^2 + \sigma_s^2), \end{aligned} \quad (143)$$

where (a) holds since $\mathbb{E} \|\beta \mathbf{x} + \mathbf{y}\|^2 = \mathbb{E} \|\beta \mathbf{x} + (1-\beta) \frac{1}{1-\beta} \mathbf{y}\|^2 \leq \beta \mathbb{E} \|\mathbf{x}\|^2 + \frac{1}{1-\beta} \mathbb{E} \|\mathbf{y}\|^2$. Substituting (139) into (143), it follows that:

$$\begin{aligned} \mathbb{E} \|\hat{\mathbf{w}}_t\|^2 &\leq \left(\beta + \frac{2\mu_m^2 \beta'^2 \delta^2}{(1-\beta)^3} + \frac{2\mu_m^2 \gamma^2 (1 + \beta_1)^2 v^2}{(1-\beta)^2} \right) \mathbb{E} \|\hat{\mathbf{w}}_{t-1}\|^2 + \frac{\mu_m^2 \sigma_s^2}{(1-\beta)^2} \\ &\quad + \left(\frac{2\mu_m^2 \delta^2}{(1-\beta)^3} + \frac{2\mu_m^2 \gamma^2 (1 + \beta_1)^2 v^2}{(1-\beta)^2} \right) \mathbb{E} \|\hat{\mathbf{w}}_{t-1}\|^2 \end{aligned} \quad (144)$$

Combining relations (141) and (144) leads to the desired result (33)–(34). Let us now examine the stability of the 2×2 coefficient matrix:

$$\Gamma \triangleq \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad (145)$$

where

$$\begin{aligned} a &= 1 - \frac{\mu_m \nu}{1-\beta} + \frac{2(1 + \beta_1)^2 \gamma^2 v^2}{(1-\beta)^2} \mu_m^2, & b &= \frac{\mu_m \beta'^2 \delta^2}{\nu(1-\beta)} + \frac{2(1 + \beta_1)^2 \gamma^2 v^2}{(1-\beta)^2} \mu_m^2, \\ c &= \frac{2\mu_m^2 \delta^2}{(1-\beta)^3} + \frac{2\mu_m^2 \gamma^2 (1 + \beta_1)^2 v^2}{(1-\beta)^2}, & d &= \beta + \frac{2\mu_m^2 \beta'^2 \delta^2}{(1-\beta)^3} + \frac{2\mu_m^2 \gamma^2 (1 + \beta_1)^2 v^2}{(1-\beta)^2}. \end{aligned} \quad (146)$$

When μ_m is sufficiently small, a, b, c, d are all positive. Since the spectral radius of a matrix is upper bounded by its 1-norm, we have that

$$\rho(\Gamma) \leq \max\{a + c, b + d\}. \quad (147)$$

From (146), we further have

$$\begin{aligned} a + c &\leq 1 - \frac{\mu_m \nu}{1 - \beta} + \frac{2(1 + \beta_1)^2 \gamma^2 \nu^2}{(1 - \beta)^2} \mu_m^2 + \frac{2\mu_m^2 \delta^2}{(1 - \beta)^3} + \frac{2\mu_m^2 \gamma^2 (1 + \beta_1)^2 \nu^2}{(1 - \beta)^2} \\ &= 1 - \frac{\mu_m \nu}{1 - \beta} + \frac{4(1 + \beta_1)^2 \gamma^2 \nu^2 + 2\delta^2}{(1 - \beta)^3} \mu_m^2 \\ &\leq 1 - \frac{\mu_m \nu}{1 - \beta} + \frac{16\gamma^2 \nu^2 + 2\delta^2}{(1 - \beta)^3} \mu_m^2, \end{aligned} \quad (148)$$

where the last inequality holds because $1 - \beta < 1$ and $1 + \beta_1 < 2$. Similarly, we also have

$$b + d \leq \beta + \frac{\delta^2 \mu_m}{\nu(1 - \beta)} + \frac{16\gamma^2 \nu^2 + 2\delta^2}{(1 - \beta)^3} \mu_m^2. \quad (149)$$

Combining (147)–(149), we reach

$$\rho(\Gamma) \leq \max \left\{ 1 - \frac{\mu_m \nu}{1 - \beta} + \frac{16\gamma^2 \nu^2 + 2\delta^2}{(1 - \beta)^3} \mu_m^2, \beta + \frac{\delta^2 \mu_m}{\nu(1 - \beta)} + \frac{16\gamma^2 \nu^2 + 2\delta^2}{(1 - \beta)^3} \mu_m^2 \right\}. \quad (150)$$

If the step-size μ_m is small enough to satisfy the following conditions

$$\begin{cases} \frac{\mu_m \nu}{1 - \beta} > \frac{16\gamma^2 \nu^2 + 2\delta^2}{(1 - \beta)^3} \mu_m^2, \\ \frac{\delta^2 \mu_m}{\nu(1 - \beta)} > \frac{16\gamma^2 \nu^2 + 2\delta^2}{(1 - \beta)^3} \mu_m^2, \\ 1 - \beta > \frac{2\delta^2 \mu_m}{\nu(1 - \beta)}, \end{cases} \quad (151)$$

which is also equivalent to

$$\mu_m < \min \left\{ \frac{(1 - \beta)^2 \nu}{32\gamma^2 \nu^2 + 4\delta^2}, \frac{(1 - \beta)^2 \delta^2}{16\gamma^2 \nu^3 + 2\delta^2 \nu}, \frac{\nu(1 - \beta)^2}{2\delta^2} \right\} = \frac{(1 - \beta)^2 \nu}{32\gamma^2 \nu^2 + 4\delta^2}, \quad (152)$$

then it holds that

$$\rho(\Gamma) < \max \left\{ 1 - \frac{\mu_m \nu}{2(1 - \beta)}, \beta + \frac{2\delta^2 \mu_m}{\nu(1 - \beta)} \right\} \leq 1, \quad (153)$$

in which case Γ will be a stable matrix.

When Γ is stable, it then follows from (33) that

$$\limsup_{i \rightarrow \infty} \frac{\mathbb{E} \|\tilde{\mathbf{w}}_i\|^2}{\mathbb{E} \|\mathbf{w}_i\|^2} \leq (I_2 - \Gamma)^{-1} \begin{bmatrix} e \\ f \end{bmatrix}. \quad (154)$$

Notice that

$$(I_2 - \Gamma)^{-1} = \begin{bmatrix} 1 - a & -b \\ -c & 1 - d \end{bmatrix}^{-1} = \frac{1}{(1 - a)(1 - d) - bc} \begin{bmatrix} 1 - d & b \\ c & 1 - a \end{bmatrix} \quad (155)$$

$$\begin{aligned} &\stackrel{(146)}{=} \frac{1}{\mu_m \nu + p_1 \mu_m^2 + p_2 \mu_m^3 + p_3 \mu_m^4} \begin{bmatrix} 1 - \beta + p_4 \mu_m^2 & \frac{\mu_m \beta^2 \delta^2}{\nu(1 - \beta)} + p_5 \mu_m^2 \\ \frac{2\mu_m^2 \delta^2}{(1 - \beta)^3} + \frac{2\mu_m^2 \gamma^2 (1 + \beta_1)^2 \nu^2}{(1 - \beta)^2} & \frac{\mu_m \nu}{1 - \beta} + p_6 \mu_m^2 \end{bmatrix} \end{aligned} \quad (155)$$

where

$$\begin{aligned} p_1 &\triangleq -\frac{2(1 + \beta_1)^2 \gamma^2 \nu^2}{1 - \beta} < 0, \quad p_4 \triangleq -\frac{2\beta^2 \delta^2}{(1 - \beta)^3} - \frac{2\gamma^2 (1 + \beta_1)^2 \nu^2}{(1 - \beta)^2} < 0, \\ p_5 &\triangleq \frac{2(1 + \beta_1)^2 \gamma^2 \nu^2}{(1 - \beta)^2} > 0, \quad p_6 \triangleq -\frac{2(1 + \beta_1)^2 \gamma^2 \nu^2}{(1 - \beta)^2} < 0. \end{aligned} \quad (156)$$

For simplicity, we omit the expression of p_2 and p_3 here. Notice that

$$\mu_m \nu + p_1 \mu_m^2 + p_2 \mu_m^3 + p_3 \mu_m^4 = \frac{\mu_m \nu}{2} + \left(\frac{\mu_m \nu}{2} + p_1 \mu_m^2 + p_2 \mu_m^3 + p_3 \mu_m^4 \right). \quad (157)$$

Although $p_1 < 0$, it still holds that $\frac{\mu_m \nu}{2} + p_1 \mu_m^2 + p_2 \mu_m^3 + p_3 \mu_m^4 > 0$ when μ_m is sufficiently small, which implies that

$$\mu_m \nu + p_1 \mu_m^2 + p_2 \mu_m^3 + p_3 \mu_m^4 > \frac{\mu_m \nu}{2}. \quad (158)$$

Similarly, it holds that

$$\frac{\mu_m \beta^2 \delta^2}{\nu(1 - \beta)} + p_5 \mu_m^2 = \frac{2\mu_m \beta^2 \delta^2}{\nu(1 - \beta)} - \left(\frac{\mu_m \beta^2 \delta^2}{\nu(1 - \beta)} - p_5 \mu_m^2 \right) \leq \frac{2\mu_m \beta^2 \delta^2}{\nu(1 - \beta)}, \quad (159)$$

where the last inequality holds because $\frac{2\mu_m \beta^2 \delta^2}{\nu(1 - \beta)} - p_5 \mu_m^2 > 0$ for sufficiently small step-size. Furthermore, since $p_4 < 0$ and $p_6 < 0$, we also have

$$1 - \beta + p_4 \mu_m^2 < 1 - \beta, \quad \frac{\mu_m \nu}{1 - \beta} + p_6 \mu_m^2 < \frac{\mu_m \nu}{1 - \beta}. \quad (160)$$

Substitute (158), (159) and (160) into (155), we have

$$(I_2 - \Gamma)^{-1} \leq \begin{bmatrix} \frac{4\mu_m \delta^2}{(1 - \beta)^3 \nu} + \frac{\mu_m \nu}{4\mu_m \gamma^2 (1 + \beta_1)^2 \nu^2} & \frac{4\beta^2 \delta^2}{\nu^2 (1 - \beta)} \\ \frac{2(1 - \beta)}{4\mu_m \delta^2} + \frac{4\beta^2 \delta^2 \nu}{(1 - \beta)^2 \nu} & \frac{4\beta^2 \delta^2}{\nu^2 (1 - \beta)} \end{bmatrix} \quad (161)$$

Combining (154) and (161), we have

$$\begin{aligned} \limsup_{i \rightarrow \infty} \frac{\mathbb{E} \|\tilde{\mathbf{w}}_i\|^2}{\mathbb{E} \|\mathbf{w}_i\|^2} &\leq (I_2 - \Gamma)^{-1} \begin{bmatrix} e \\ f \end{bmatrix} \\ &\leq \begin{bmatrix} \frac{2(1 - \beta)}{4\mu_m \delta^2} + \frac{\mu_m \nu}{4\mu_m \gamma^2 (1 + \beta_1)^2 \nu^2} & \frac{4\beta^2 \delta^2}{\nu^2 (1 - \beta)} \\ \frac{2(1 - \beta)}{4\mu_m \delta^2} + \frac{4\beta^2 \delta^2 \nu}{(1 - \beta)^2 \nu} & \frac{4\beta^2 \delta^2}{\nu^2 (1 - \beta)} \end{bmatrix} \begin{bmatrix} \frac{\mu_m^2 \delta^2}{(1 - \beta)^2} \\ \frac{\mu_m \nu}{1 - \beta} \end{bmatrix} \\ &= \begin{bmatrix} \frac{2\mu_m^2 \delta^2}{(1 - \beta)^3} + \frac{4\mu_m^3 \delta^2 \nu}{(1 - \beta)^2 \nu} & \frac{2\mu_m \delta^2}{(1 - \beta) \nu} + \frac{4\beta^2 \delta^2 \nu \mu_m^2}{4\mu_m \gamma^2 (1 + \beta_1)^2 \nu^2} \\ \frac{2\mu_m^2 \delta^2}{(1 - \beta)^3} + \frac{4\mu_m^3 \delta^2 \nu}{(1 - \beta)^2 \nu} & \frac{2\mu_m \delta^2}{(1 - \beta) \nu} + \frac{4\beta^2 \delta^2 \nu \mu_m^2}{4\mu_m \gamma^2 (1 + \beta_1)^2 \nu^2} \end{bmatrix} \leq \begin{bmatrix} \frac{3\mu_m \delta^2}{(1 - \beta)^2} \\ \frac{3\mu_m \nu}{(1 - \beta)^2} \end{bmatrix} \end{aligned} \quad (162)$$

where in the last inequality we choose sufficiently small μ_m such that

$$\frac{4\beta^2\delta^2\sigma_s^2\mu_m^2}{(1-\beta)^3\nu^2} < \frac{\mu_m\sigma_s^2}{(1-\beta)\nu} + \frac{4\mu_m^3\delta^2\sigma_s^2}{(1-\beta)^5\nu} + \frac{4\mu_m^3\gamma^2(1+\beta)^2\nu^2\sigma_s^2}{(1-\beta)^4\nu} < \frac{\mu_m^2\sigma_s^2}{(1-\beta)^3} \quad (163)$$

Therefore, we have the following result

$$\limsup_{t \rightarrow \infty} \mathbb{E} \|\hat{w}_t\|^2 = O\left(\frac{\mu_m\sigma_s^2}{(1-\beta)\nu}\right), \quad \limsup_{t \rightarrow \infty} \mathbb{E} \|w_t\|^2 = O\left(\frac{\mu_m^2\sigma_s^2}{(1-\beta)^3}\right). \quad (164)$$

and

$$\begin{aligned} \limsup_{t \rightarrow \infty} \mathbb{E} \left\| \begin{bmatrix} \hat{w}_t \\ \hat{w}_{t-1} \end{bmatrix} \right\|^2 &= \limsup_{t \rightarrow \infty} \mathbb{E} \left\| V \begin{bmatrix} \hat{w}_t \\ \hat{w}_{t-1} \end{bmatrix} \right\|^2 \\ &\leq \nu^2 \left(\limsup_{t \rightarrow \infty} \mathbb{E} \left\| \begin{bmatrix} \hat{w}_t \\ \hat{w}_{t-1} \end{bmatrix} \right\|^2 \right) \\ &= \nu^2 \left(\limsup_{t \rightarrow \infty} (\mathbb{E} \|\hat{w}_t\|^2 + \mathbb{E} \|\hat{w}_{t-1}\|^2) \right) = O\left(\frac{\mu_m\sigma_s^2}{(1-\beta)\nu}\right), \end{aligned} \quad (165)$$

from which we conclude that (37) holds.

Appendix D. Proof of Corollary 5

To simplify the notation, we refer to (33) and introduce the quantities:

$$z_t = \begin{bmatrix} \mathbb{E} \|\hat{w}_t\|^2 \\ \mathbb{E} \|\hat{w}_{t-1}\|^2 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad r = \begin{bmatrix} e \\ f \end{bmatrix}. \quad (166)$$

Then, relation (33) can be rewritten as

$$z_t \preceq \Gamma z_{t-1} + r. \quad (167)$$

It follows that, in terms of the 1-norm,

$$\|z_t\|_1 \leq \|\Gamma\|_1 \|z_{t-1}\|_1 + \|r\|_1, \quad (168)$$

where

$$\|\Gamma\|_1 = \max \left\{ 1 - \frac{\mu_m\nu}{1-\beta} + B_1\mu_m^2, \beta + B_3\mu_m \right\} \quad (169)$$

for some constant B_1 and B_2 . Now we can choose μ_m sufficiently small to satisfy

$$B_1\mu_m^2 < \frac{\nu\mu_m}{2(1-\beta)}, \quad \left(B_2 + \frac{\nu}{2(1-\beta)} \right) \mu_m < 1 - \beta, \quad (170)$$

which implies that

$$\|\Gamma\|_1 \leq 1 - \frac{\mu_m\nu}{1-\beta} + B_1\mu_m^2 \leq 1 - \frac{\mu_m\nu}{2(1-\beta)} \triangleq \rho_1 < 1 \quad (171)$$

Then, from (168) we have

$$\|z_t\|_1 \leq \rho_1 \|z_{t-1}\|_1 + \|r\|_1. \quad (172)$$

Iterating (172) gives

$$\|z_t\|_1 \leq \rho_1^{t+1} \|z_{-1}\|_1 + \frac{\|r\|_1}{1-\rho_1}. \quad (173)$$

Recall the expressions of e and f from (34), we have $\|r\|_1 \leq \frac{B_3\mu_m^2\sigma_s^2}{(1-\beta)^3}$ for some constant B_3 .

Since $1 - \rho_1 = \frac{\mu_m\nu}{2(1-\beta)}$, we get $\|r\|_1 / (1 - \rho_1) \leq \frac{2B_3\mu_m\sigma_s^2}{(1-\beta)\nu}$. From (173), we have

$$\|z_t\|_1 \leq \rho_1^{t+1} \|z_{-1}\|_1 + \frac{2B_3\mu_m\sigma_s^2}{(1-\beta)\nu}. \quad (174)$$

Accordingly, using

$$\|z_t\|_1 = \mathbb{E} \|\hat{w}_t\|^2 + \mathbb{E} \|\hat{w}_{t-1}\|^2 \quad (175)$$

we also find that

$$\mathbb{E} \|\hat{w}_t\|^2 \leq \rho_1^{t+1} \|z_{-1}\|_1 + \frac{2B_3\mu_m\sigma_s^2}{(1-\beta)\nu}. \quad (176)$$

On the other hand, we know from the second row of (33) that

$$\mathbb{E} \|\hat{w}_t\|^2 \leq (\beta + c_1\mu_m^2) \mathbb{E} \|\hat{w}_{t-1}\|^2 + c_2\mu_m^2 \mathbb{E} \|\hat{w}_{t-1}\|^2 + c_3\mu_m^2 \quad (177)$$

for constants

$$c_1 \triangleq \frac{2\beta^2\delta^2}{(1-\beta)^3} + \frac{2\gamma^2(1+\beta_1)^2\nu^2}{(1-\beta)^2}, \quad c_2 \triangleq \frac{2\delta^2}{(1-\beta)^3} + \frac{2\gamma^2(1+\beta)^2\nu^2}{(1-\beta)^2}, \quad c_3 \triangleq \frac{\sigma_s^2}{(1-\beta)^2}. \quad (178)$$

To simplify the notation, with the facts that $\beta' < 1$, $\beta < 1$ and $\beta_1 < 1$, we have

$$c_1 \leq \frac{B_4(\delta^2 + \gamma^2)}{(1-\beta)^3} \triangleq c_4, \quad c_2 \leq \frac{B_4(\delta^2 + \gamma^2)}{(1-\beta)^3} = c_4 \quad (179)$$

for some constant B_4 . Substituting (179) into (177) we get

$$\mathbb{E} \|\hat{w}_t\|^2 \leq (\beta + c_4\mu_m^2) \mathbb{E} \|\hat{w}_{t-1}\|^2 + c_4\mu_m^2 \mathbb{E} \|\hat{w}_{t-1}\|^2 + c_3\mu_m^2. \quad (180)$$

Now we substitute (176) into (180), and reach

$$\mathbb{E} \|w_t\|^2 \leq (\beta + c_4\mu_m^2) \mathbb{E} \|\hat{w}_{t-1}\|^2 + c_4\mu_m^2 \|z_{-1}\|_1 + \frac{2B_3c_4\sigma_s^2}{(1-\beta)\nu} \mu_m^3 + c_3\mu_m^2. \quad (181)$$

When μ_m is sufficiently small such that

$$\frac{2B_3c_4\sigma_s^2}{(1-\beta)\nu} \mu_m^3 \leq c_3\mu_m^2, \quad (182)$$

(181) becomes

$$\mathbb{E}\|\tilde{\mathbf{w}}_i\|^2 \leq (\beta + c_4\mu_m^2)\mathbb{E}\|\tilde{\mathbf{w}}_{i-1}\|^2 + c_4\rho_1^i\|z_{-1}\|_1\mu_m^2 + 2c_3\mu_m^2. \quad (183)$$

Notice that

$$\beta + c_4\mu_m^2 = 1 - (1 - \beta) + c_4\mu_m^2 = 1 - \frac{1 - \beta}{2} + \left(c_4\mu_m^2 - \frac{1 - \beta}{2}\right). \quad (184)$$

It is clear that we can choose a sufficiently small μ_m for the last term between brackets to become negative, in which case

$$\beta + c_4\mu_m^2 \leq 1 - \frac{1 - \beta}{2} = \frac{1 + \beta}{2} \triangleq \alpha < 1 \quad (185)$$

It follows that

$$\begin{aligned} \mathbb{E}\|\tilde{\mathbf{w}}_i\|^2 &\leq \alpha\mathbb{E}\|\tilde{\mathbf{w}}_{i-1}\|^2 + (c_4\|z_{-1}\|_1\rho_1^i)\mu_m^2 + 2c_3\mu_m^2 \\ &\leq \alpha^{i+1}\mathbb{E}\|\tilde{\mathbf{w}}_{-1}\|^2 + c_4\|z_{-1}\|_1\mu_m^i\rho_1^i \sum_{s=0}^i \left(\frac{\alpha}{\rho_1}\right)^s + \frac{2c_3\mu_m^2}{1 - \alpha}. \end{aligned} \quad (186)$$

Recall that $\rho_1 = 1 - \frac{\mu_m\nu}{2(1-\beta)}$ and $\alpha = 1 - (1 - \beta)/2$. Therefore, it holds that $\alpha/\rho_1 < 1$ for sufficiently small μ_m . As a result, we have

$$\sum_{s=0}^i \left(\frac{\alpha}{\rho_1}\right)^s \leq \frac{1}{1 - \frac{\alpha}{\rho_1}} = \frac{\rho_1}{\rho_1 - \alpha} = \frac{2(1 - \beta) - \mu_m\nu}{(1 - \beta)^2 - \mu_m\nu} \leq \frac{B_5}{1 - \beta} \quad (187)$$

for some constant B_5 when μ_m is sufficiently small. Substituting (187) into (186), we get

$$\mathbb{E}\|\tilde{\mathbf{w}}_i\|^2 \leq \alpha^{i+1}\mathbb{E}\|\tilde{\mathbf{w}}_{-1}\|^2 + \frac{B_5c_4\|z_{-1}\|_1\rho_1^i}{1 - \beta}\mu_m^2 + \frac{4c_3\mu_m^2}{1 - \beta}. \quad (188)$$

To assess the term that depends on the initial state, $\mathbb{E}\|\tilde{\mathbf{w}}_{-1}\|^2$, let us consider the boundary conditions (24)–(25). Then, from (28) it holds that

$$\tilde{\mathbf{w}}_{-1} = \frac{\tilde{\mathbf{w}}_{-1} - \tilde{\mathbf{w}}_{-2}}{1 - \beta} = \frac{\mathbf{w}_{-2} - \mathbf{w}_{-1}}{1 - \beta} = \frac{\mu_m\nabla_w Q(\mathbf{w}_{-2}; \boldsymbol{\theta}_{-1})}{1 - \beta} \quad (189)$$

so that $\mathbb{E}\|\tilde{\mathbf{w}}_{-1}\|^2 = c_5\mu_m^2$, where

$$c_5 \triangleq \mathbb{E}\|\nabla_w Q(\mathbf{w}_{-2}; \boldsymbol{\theta}_{-1})\|^2 / (1 - \beta)^2. \quad (190)$$

Substituting this conclusion into (188), and recalling the expression of c_3 , c_4 and c_5 , we arrive at

$$\begin{aligned} \mathbb{E}\|\tilde{\mathbf{w}}_i\|^2 &\leq \frac{B_6\alpha^{i+1}\mu_m^2}{(1 - \beta)^2} + \frac{B_7(\delta^2 + \gamma^2)\rho_1^i}{(1 - \beta)^4}\mu_m^2 + \frac{B_8\mu_m^2\sigma_s^2}{(1 - \beta)^3} \\ &\stackrel{(a)}{\leq} \frac{B_9\rho_1^{i+1}\mu_m^2}{(1 - \beta)^2} + \frac{B_9(\delta^2 + \gamma^2)\rho_1^{i+1}}{(1 - \beta)^4}\mu_m^2 + \frac{B_8\mu_m^2\sigma_s^2}{(1 - \beta)^3} \end{aligned}$$

$$\stackrel{(b)}{\leq} \frac{B_{10}(\delta^2 + \gamma^2)\rho_1^{i+1}}{(1 - \beta)^4}\mu_m^2 + \frac{B_8\mu_m^2\sigma_s^2}{(1 - \beta)^3}, \quad (191)$$

where (a) holds because $\alpha \leq \rho_1$ when μ_m is sufficiently small, and there must exist some constant B_9 such that $B_7/\rho_1 < B_9$; (b) holds because there must exist some constant B_{10} such that

$$B_6(1 - \beta)^2 + B_9(\delta^2 + \gamma^2) \leq B_{10}(\delta^2 + \gamma^2). \quad (192)$$

Appendix E. Proof of Theorem 6

The argument below is motivated by the derivation of Theorem 9.2 in (Sayed, 2014a). Here, however, we extend the arguments and expand the details in order to clearly identify the constants inside the $O(\mu)$ notation, which was not necessary in (Sayed, 2014a). The derivation becomes more demanding, as the arguments show.

From the first row of recursion (29) we have

$$\tilde{\mathbf{w}}_i = \left(I_M - \frac{\mu_m \mathbf{H}_{i-1}}{1 - \beta}\right) \tilde{\mathbf{w}}_{i-1} + \frac{\mu_m \beta' \mathbf{H}_{i-1}}{1 - \beta} \tilde{\mathbf{w}}_{i-1} + \frac{\mu_m}{1 - \beta} \mathbf{s}_i(\psi_{i-1}). \quad (193)$$

Now applying the following inequality, for any two vectors $\{a, b\}$:

$$\|a + b\|^4 \leq \|a\|^4 + 3\|b\|^4 + 8\|a\|^2\|b\|^2 + 4\|a\|^2(a^\top b) \quad (194)$$

we get

$$\begin{aligned} &\mathbb{E}\|\tilde{\mathbf{w}}_i\|^4 \mathcal{F}_{i-1} \\ &= \mathbb{E}\left\| \left(I_M - \frac{\mu_m \mathbf{H}_{i-1}}{1 - \beta}\right) \tilde{\mathbf{w}}_{i-1} + \frac{\mu_m \beta' \mathbf{H}_{i-1}}{1 - \beta} \tilde{\mathbf{w}}_{i-1} + \frac{3\mu_m}{(1 - \beta)^4} \mathbb{E}\|\mathbf{s}_i(\psi_{i-1})\|^4 \mathcal{F}_{i-1} \right\|^4 \\ &\quad + \frac{8\mu_m^2}{(1 - \beta)^2} \mathbb{E}\left\| \left(I_M - \frac{\mu_m \mathbf{H}_{i-1}}{1 - \beta}\right) \tilde{\mathbf{w}}_{i-1} + \frac{\mu_m \beta' \mathbf{H}_{i-1}}{1 - \beta} \tilde{\mathbf{w}}_{i-1} \right\|^2 \mathbb{E}\|\mathbf{s}_i(\psi_{i-1})\|^2 \mathcal{F}_{i-1} \\ &\leq \mathbb{E}\left\| \left(I_M - \frac{\mu_m \mathbf{H}_{i-1}}{1 - \beta}\right) \tilde{\mathbf{w}}_{i-1} + \frac{\mu_m \beta' \mathbf{H}_{i-1}}{1 - \beta} \tilde{\mathbf{w}}_{i-1} \right\|^4 + \frac{3\mu_m^4 (\gamma_4^4 \|\tilde{\psi}_{i-1}\|^4 + \sigma_{s,4})}{(1 - \beta)^4} \\ &\quad + \frac{8\mu_m^2}{(1 - \beta)^2} \mathbb{E}\left\| \left(I_M - \frac{\mu_m \mathbf{H}_{i-1}}{1 - \beta}\right) \tilde{\mathbf{w}}_{i-1} + \frac{\mu_m \beta' \mathbf{H}_{i-1}}{1 - \beta} \tilde{\mathbf{w}}_{i-1} \right\|^2 (\gamma^2 \|\tilde{\psi}_{i-1}\|^2 + \sigma_s^2). \end{aligned} \quad (195)$$

We next bound each of the terms that appear on the right-hand side. Using Jensen's inequality, the lower and upper bounds on the Hessian matrix from (11), we have

$$\begin{aligned} &\left\| \left(I_M - \frac{\mu_m \mathbf{H}_{i-1}}{1 - \beta}\right) \tilde{\mathbf{w}}_{i-1} + \frac{\mu_m \beta' \mathbf{H}_{i-1}}{1 - \beta} \tilde{\mathbf{w}}_{i-1} \right\|^4 \\ &= \left\| (1-t) \frac{1}{1-t} \left(I_M - \frac{\mu_m \mathbf{H}_{i-1}}{1 - \beta}\right) \tilde{\mathbf{w}}_{i-1} + t \frac{1}{t} \frac{\mu_m \beta' \mathbf{H}_{i-1}}{1 - \beta} \tilde{\mathbf{w}}_{i-1} \right\|^4 \end{aligned}$$

$$\begin{aligned}
&\leq \frac{1}{(1-t)^3} \left(1 - \frac{\mu_m \nu}{1-\beta}\right)^4 \|\hat{\mathbf{w}}_{i-1}\|^4 + \frac{1}{t^3} \frac{\mu_m^4 \beta^4 \delta^4}{(1-\beta)^4} \|\hat{\mathbf{w}}_{i-1}\|^4 \\
&\stackrel{(a)}{=} \left(1 - \frac{\mu_m \nu}{1-\beta}\right) \|\hat{\mathbf{w}}_{i-1}\|^4 + \frac{\mu_m \beta^4 \delta^4}{(1-\beta) \nu^3} \|\hat{\mathbf{w}}_{i-1}\|^4 \\
&= (1 - q_1 \mu_m) \|\hat{\mathbf{w}}_{i-1}\|^4 + q_2 \mu_m \|\hat{\mathbf{w}}_{i-1}\|^4,
\end{aligned} \tag{196}$$

where (a) holds because we set $t = \mu_m \nu / (1 - \beta)$, and q_1, q_2 are defined as

$$q_1 \triangleq \frac{\nu}{1-\beta}, \quad q_2 \triangleq \frac{\beta^4 \delta^4}{(1-\beta) \nu^3}. \tag{197}$$

Next we check the terms $\mathbb{E}[\|\mathbf{s}_i(\psi_{i-1})\|^2 | \mathcal{F}_{i-1}]$ and $\mathbb{E}[\|\mathbf{s}_i(\psi_{i-1})\|^4 | \mathcal{F}_{i-1}]$. From (139) we have

$$\|\tilde{\psi}_{i-1}\|^2 \leq B_1 (\|\hat{\mathbf{w}}_{i-1}\|^2 + \|\mathbf{w}_{i-1}\|^2), \tag{198}$$

where $B_1 = 2(1 + \beta_1)^2 \nu^2$, which also implies that

$$\begin{aligned}
\|\tilde{\psi}_{i-1}\|^4 &\leq B_1^2 (\|\hat{\mathbf{w}}_{i-1}\|^2 + \|\mathbf{w}_{i-1}\|^2)^2 \\
&\leq 2B_1^2 (\|\hat{\mathbf{w}}_{i-1}\|^4 + \|\mathbf{w}_{i-1}\|^4) = B_2 (\|\hat{\mathbf{w}}_{i-1}\|^4 + \|\mathbf{w}_{i-1}\|^4),
\end{aligned} \tag{199}$$

where $B_2 = 2B_1^2$. Furthermore, recall in (137) that

$$\begin{aligned}
&\left\| \left(I_M - \frac{\mu_m \mathbf{H}_{i-1}}{1-\beta} \right) \hat{\mathbf{w}}_{i-1} + \frac{\mu_m \beta \mathbf{H}_{i-1}}{1-\beta} \tilde{\mathbf{w}}_{i-1} \right\|^2 \\
&\leq \left(1 - \frac{\mu_m \nu}{1-\beta}\right) \|\hat{\mathbf{w}}_{i-1}\|^2 + \frac{\mu_m \beta^2 \delta^2}{\nu(1-\beta)} \|\tilde{\mathbf{w}}_{i-1}\|^2 \\
&= (1 - q_1 \mu_m) \|\hat{\mathbf{w}}_{i-1}\|^2 + q_3 \mu_m \|\tilde{\mathbf{w}}_{i-1}\|^2,
\end{aligned} \tag{200}$$

where we define

$$q_3 \triangleq \frac{\beta^2 \delta^2}{\nu(1-\beta)}. \tag{201}$$

Now substituting (196), (198), (199) and (200) into (195), we get

$$\begin{aligned}
&\mathbb{E}[\|\hat{\mathbf{w}}_i\|^4 | \mathcal{F}_{i-1}] \\
&\leq (1 - q_1 \mu_m) \|\hat{\mathbf{w}}_{i-1}\|^4 + q_2 \mu_m \|\hat{\mathbf{w}}_{i-1}\|^4 + \frac{3B_2^4 \mu_m^4}{(1-\beta)^4} (\|\hat{\mathbf{w}}_{i-1}\|^4 + \|\tilde{\mathbf{w}}_{i-1}\|^4) + \frac{3\sigma_s^4 \mu_m^4}{(1-\beta)^4} \\
&\quad + \frac{8\mu_m^2}{(1-\beta)^2} [(1 - q_1 \mu_m) \|\hat{\mathbf{w}}_{i-1}\|^2 + q_3 \mu_m \|\tilde{\mathbf{w}}_{i-1}\|^2] [\gamma^2 B_1 (\|\hat{\mathbf{w}}_{i-1}\|^2 + \|\tilde{\mathbf{w}}_{i-1}\|^2) + \sigma_s^2] \\
&= (1 - q_1 \mu_m) \|\hat{\mathbf{w}}_{i-1}\|^4 + q_2 \mu_m \|\hat{\mathbf{w}}_{i-1}\|^4 + q_4 \mu_m^4 (\|\hat{\mathbf{w}}_{i-1}\|^4 + \|\tilde{\mathbf{w}}_{i-1}\|^4) + q_5 \mu_m^4 \\
&\quad + q_6 \mu_m^2 [(1 - q_1 \mu_m) \|\hat{\mathbf{w}}_{i-1}\|^2 + q_3 \mu_m \|\tilde{\mathbf{w}}_{i-1}\|^2] [\gamma^2 B_1 (\|\hat{\mathbf{w}}_{i-1}\|^2 + \|\tilde{\mathbf{w}}_{i-1}\|^2) + \sigma_s^2], \\
&= (1 - q_1 \mu_m + q_4 \mu_m^4) \|\hat{\mathbf{w}}_{i-1}\|^4 + (q_2 \mu_m + q_4 \mu_m^4) \|\tilde{\mathbf{w}}_{i-1}\|^4 + q_5 \mu_m^4 \\
&\quad + q_6 \gamma^2 B_1 (1 - q_1 \mu_m) \mu_m^2 \|\hat{\mathbf{w}}_{i-1}\|^4 + q_6 q_3 \gamma^2 B_1 \mu_m^3 \|\tilde{\mathbf{w}}_{i-1}\|^4
\end{aligned}$$

$$\begin{aligned}
&+ q_6 \gamma^2 B_1 \mu_m^2 (1 - q_1 \mu_m + q_3 \mu_m) \|\hat{\mathbf{w}}_{i-1}\|^2 \|\tilde{\mathbf{w}}_{i-1}\|^2 \\
&+ q_6 \sigma_s^2 \mu_m^2 (1 - q_1 \mu_m) \|\hat{\mathbf{w}}_{i-1}\|^2 + q_6 q_3 \sigma_s^2 \mu_m^3 \|\tilde{\mathbf{w}}_{i-1}\|^2 \\
&\stackrel{(a)}{\leq} (1 - q_1 \mu_m + q_4 \mu_m^4) \|\hat{\mathbf{w}}_{i-1}\|^4 + (q_2 \mu_m + q_4 \mu_m^4) \|\tilde{\mathbf{w}}_{i-1}\|^4 + q_5 \mu_m^4 \\
&+ q_6 \gamma^2 B_1 (1 - q_1 \mu_m) \mu_m^2 \|\hat{\mathbf{w}}_{i-1}\|^2 + q_6 q_3 \gamma^2 B_1 \mu_m^3 \|\tilde{\mathbf{w}}_{i-1}\|^2 \\
&+ q_6 \gamma^2 B_1 \mu_m^2 (1 - q_1 \mu_m + q_3 \mu_m) (\|\hat{\mathbf{w}}_{i-1}\|^4 + \|\tilde{\mathbf{w}}_{i-1}\|^4) \\
&+ q_6 \sigma_s^2 \mu_m^2 (1 - q_1 \mu_m) \|\hat{\mathbf{w}}_{i-1}\|^2 + q_6 q_3 \sigma_s^2 \mu_m^3 \|\tilde{\mathbf{w}}_{i-1}\|^2,
\end{aligned} \tag{202}$$

where we define

$$q_4 \triangleq \frac{3B_2^4 \gamma^4}{(1-\beta)^4}, \quad q_5 \triangleq \frac{3\sigma_s^4}{(1-\beta)^4}, \quad q_6 \triangleq \frac{8}{(1-\beta)^2}, \tag{203}$$

and (a) holds because for any two variables $a, b > 0$ we have

$$ab < 2ab \leq a^2 + b^2. \tag{204}$$

When μ_m is chosen sufficiently small, from (202) we reach

$$\begin{aligned}
&\mathbb{E}[\|\hat{\mathbf{w}}_i\|^4 | \mathcal{F}_{i-1}] \\
&\leq \left(1 - \frac{q_1 \mu_m}{2}\right) \|\hat{\mathbf{w}}_{i-1}\|^4 + 2q_2 \mu_m \|\hat{\mathbf{w}}_{i-1}\|^4 + q_6 \sigma_s^2 \mu_m^2 \|\hat{\mathbf{w}}_{i-1}\|^2 + q_6 q_3 \sigma_s^2 \mu_m^3 \|\tilde{\mathbf{w}}_{i-1}\|^2 + q_5 \mu_m^4 \\
&= \left(1 - \frac{q_1 \mu_m}{2}\right) \|\hat{\mathbf{w}}_{i-1}\|^4 + 2q_2 \mu_m \|\hat{\mathbf{w}}_{i-1}\|^4 + q_7 \mu_m^2 \|\hat{\mathbf{w}}_{i-1}\|^2 + q_8 \mu_m^3 \|\tilde{\mathbf{w}}_{i-1}\|^2 + q_5 \mu_m^4,
\end{aligned} \tag{205}$$

where we define

$$q_7 \triangleq q_6 \sigma_s^2, \quad q_8 \triangleq q_6 q_3 \sigma_s^2. \tag{206}$$

On the other hand, recall from (142) that

$$\hat{\mathbf{w}}_i = -\frac{\mu_m}{1-\beta} \mathbf{H}_{i-1} \hat{\mathbf{w}}_{i-1} + \left(\beta I_M + \frac{\mu_m \beta'}{1-\beta} \mathbf{H}_{i-1} \right) \tilde{\mathbf{w}}_{i-1} + \frac{\mu_m}{1-\beta} \mathbf{s}_i(\psi_{i-1}). \tag{207}$$

Now we also apply inequality (194) to the above equation and get

$$\begin{aligned}
&\mathbb{E}[\|\hat{\mathbf{w}}_i\|^4 | \mathcal{F}_{i-1}] \\
&= \left\| -\frac{\mu_m}{1-\beta} \mathbf{H}_{i-1} \hat{\mathbf{w}}_{i-1} + \left(\beta I_M + \frac{\mu_m \beta'}{1-\beta} \mathbf{H}_{i-1} \right) \tilde{\mathbf{w}}_{i-1} \right\|^4 + \frac{3\mu_m^4}{(1-\beta)^4} \mathbb{E}[\|\mathbf{s}_i(\psi_{i-1})\|^4 | \mathcal{F}_{i-1}] \\
&\quad + \frac{8\mu_m^2}{(1-\beta)^2} \left\| -\frac{\mu_m}{1-\beta} \mathbf{H}_{i-1} \hat{\mathbf{w}}_{i-1} + \left(\beta I_M + \frac{\mu_m \beta'}{1-\beta} \mathbf{H}_{i-1} \right) \tilde{\mathbf{w}}_{i-1} \right\|^2 \mathbb{E}[\|\mathbf{s}_i(\psi_{i-1})\|^2 | \mathcal{F}_{i-1}] \\
&\leq \left\| -\frac{\mu_m}{1-\beta} \mathbf{H}_{i-1} \hat{\mathbf{w}}_{i-1} + \left(\beta I_M + \frac{\mu_m \beta'}{1-\beta} \mathbf{H}_{i-1} \right) \tilde{\mathbf{w}}_{i-1} \right\|^4 + \frac{3\mu_m^4}{(1-\beta)^4} \frac{\gamma^4 (\|\hat{\psi}_{i-1}\|^4 + \sigma_s^4)}{(1-\beta)^4} \\
&\quad + \frac{8\mu_m^2}{(1-\beta)^2} \left\| -\frac{\mu_m}{1-\beta} \mathbf{H}_{i-1} \hat{\mathbf{w}}_{i-1} + \left(\beta I_M + \frac{\mu_m \beta'}{1-\beta} \mathbf{H}_{i-1} \right) \tilde{\mathbf{w}}_{i-1} \right\|^2 (\gamma^2 \|\tilde{\psi}_{i-1}\|^2 + \sigma_s^2).
\end{aligned} \tag{208}$$

We next bound each of the terms that appear on the right-hand side. Using Jensen's inequality, the lower and upper bounds on the Hessian matrix from (11), and the inequality $\|a + b\|^4 \leq 8\|a\|^4 + 8\|b\|^4$, we have

$$\begin{aligned} & \left\| \frac{\mu_m \mathbf{H}_{i-1}}{1-\beta} \hat{\mathbf{w}}_{i-1} + \left(\beta \mathbf{I}_M + \frac{\mu_m \beta'}{1-\beta} \mathbf{H}_{i-1} \right) \hat{\mathbf{w}}_{i-1} \right\|^4 \\ &= \left\| \beta \hat{\mathbf{w}}_{i-1} + (1-\beta) \left(\frac{\mu_m \beta'}{(1-\beta)^2} \mathbf{H}_{i-1} \hat{\mathbf{w}}_{i-1} - \frac{\mu_m}{(1-\beta)^2} \mathbf{H}_{i-1} \hat{\mathbf{w}}_{i-1} \right) \right\|^4 \\ &\leq \beta \|\hat{\mathbf{w}}_{i-1}\|^4 + (1-\beta) \left\| \frac{\mu_m \beta'}{(1-\beta)^2} \mathbf{H}_{i-1} \hat{\mathbf{w}}_{i-1} - \frac{\mu_m}{(1-\beta)^2} \mathbf{H}_{i-1} \hat{\mathbf{w}}_{i-1} \right\|^4 \\ &\leq \beta \|\hat{\mathbf{w}}_{i-1}\|^4 + \frac{8\mu_m^4 \beta'^4 \delta^4}{(1-\beta)^7} \|\hat{\mathbf{w}}_{i-1}\|^4 + \frac{8\mu_m^4 \delta^4}{(1-\beta)^7} \|\hat{\mathbf{w}}_{i-1}\|^4 \\ &= (\beta + p_1 \mu_m^4) \|\hat{\mathbf{w}}_{i-1}\|^4 + p_2 \mu_m^4 \|\hat{\mathbf{w}}_{i-1}\|^4, \end{aligned} \quad (209)$$

where we define

$$p_1 \triangleq \frac{8\beta^4 \delta^4}{(1-\beta)^7}, \quad p_2 \triangleq \frac{8\delta^4}{(1-\beta)^7}.$$

Moreover, recall in (143) that

$$\begin{aligned} & \left\| \beta \hat{\mathbf{w}}_{i-1} + \left(\frac{\mu_m \beta'}{1-\beta} \hat{\mathbf{w}}_{i-1} - \frac{\mu_m \mathbf{H}_{i-1}}{1-\beta} \hat{\mathbf{w}}_{i-1} \right) \right\|^2 \\ &\leq \left(\beta + \frac{2\mu_m^2 \beta'^2 \delta^2}{(1-\beta)^3} \right) \|\hat{\mathbf{w}}_{i-1}\|^2 + \frac{2\mu_m^2 \delta^2}{(1-\beta)^3} \|\hat{\mathbf{w}}_{i-1}\|^2 \\ &= (\beta + p_3 \mu_m^2) \|\hat{\mathbf{w}}_{i-1}\|^2 + p_4 \mu_m^2 \|\hat{\mathbf{w}}_{i-1}\|^2, \end{aligned} \quad (210)$$

where we define

$$p_3 \triangleq \frac{2\beta'^2 \delta^2}{(1-\beta)^3}, \quad p_4 \triangleq \frac{2\delta^2}{(1-\beta)^3}. \quad (212)$$

Now substituting (209), (211), (198) and (199) into (208), we have

$$\begin{aligned} & \mathbb{E}[\|\hat{\mathbf{w}}_i\|^4 | \mathcal{F}_{i-1}] \\ &\leq (\beta + p_1 \mu_m^4) \|\hat{\mathbf{w}}_{i-1}\|^4 + p_2 \mu_m^4 \|\hat{\mathbf{w}}_{i-1}\|^4 + \frac{3B_2 \gamma^4 \mu_m^4}{(1-\beta)^4} (\|\hat{\mathbf{w}}_{i-1}\|^4 + \|\hat{\mathbf{w}}_{i-1}\|^4) + \frac{3\sigma_s^4 \mu_m^4}{(1-\beta)^4} \\ &\quad + \frac{8\mu_m^2}{(1-\beta)^2} [(\beta + p_3 \mu_m^2) \|\hat{\mathbf{w}}_{i-1}\|^2 + p_4 \mu_m^2 \|\hat{\mathbf{w}}_{i-1}\|^2] [\gamma^2 B_1 (\|\hat{\mathbf{w}}_{i-1}\|^2 + \|\hat{\mathbf{w}}_{i-1}\|^2) + \sigma_s^2] \\ &= (\beta + p_1 \mu_m^4) \|\hat{\mathbf{w}}_{i-1}\|^4 + p_2 \mu_m^4 \|\hat{\mathbf{w}}_{i-1}\|^4 + p_5 \mu_m^4 (\|\hat{\mathbf{w}}_{i-1}\|^4 + \|\hat{\mathbf{w}}_{i-1}\|^4) + p_6 \mu_m^4 \\ &\quad + p_7 \mu_m^2 [(\beta + p_3 \mu_m^2) \|\hat{\mathbf{w}}_{i-1}\|^2 + p_4 \mu_m^2 \|\hat{\mathbf{w}}_{i-1}\|^2] [\gamma^2 B_1 (\|\hat{\mathbf{w}}_{i-1}\|^2 + \|\hat{\mathbf{w}}_{i-1}\|^2) + \sigma_s^2] \\ &= [\beta + (p_1 + p_5) \mu_m^4] \|\hat{\mathbf{w}}_{i-1}\|^4 + (p_2 + p_5) \mu_m^4 \|\hat{\mathbf{w}}_{i-1}\|^4 + p_6 \mu_m^4 \\ &\quad + p_7 \gamma^2 B_1 \mu_m^2 (\beta + p_3 \mu_m^2) \|\hat{\mathbf{w}}_{i-1}\|^4 + p_4 p_7 \gamma^2 B_1 \mu_m^4 \|\hat{\mathbf{w}}_{i-1}\|^4 \\ &\quad + p_7 \gamma^2 B_1 \mu_m^2 (\beta + p_3 \mu_m^2) \|\hat{\mathbf{w}}_{i-1}\|^2 \|\hat{\mathbf{w}}_{i-1}\|^2. \end{aligned}$$

$$\begin{aligned} & + p_7 \sigma_s^2 \mu_m^2 (\beta + p_3 \mu_m^2) \|\hat{\mathbf{w}}_{i-1}\|^2 + p_4 p_7 \sigma_s^2 \mu_m^4 \|\hat{\mathbf{w}}_{i-1}\|^2 \\ &\leq [\beta + (p_1 + p_5) \mu_m^4] \|\hat{\mathbf{w}}_{i-1}\|^4 + (p_2 + p_5) \mu_m^4 \|\hat{\mathbf{w}}_{i-1}\|^4 + p_6 \mu_m^4 \\ &\quad + p_7 \gamma^2 B_1 \mu_m^2 (\beta + p_3 \mu_m^2) \|\hat{\mathbf{w}}_{i-1}\|^4 + p_4 p_7 \gamma^2 B_1 \mu_m^4 \|\hat{\mathbf{w}}_{i-1}\|^4 \\ &\quad + p_7 \gamma^2 B_1 \mu_m^2 (\beta + p_3 \mu_m^2) (\|\hat{\mathbf{w}}_{i-1}\|^4 + \|\hat{\mathbf{w}}_{i-1}\|^4) \\ &\quad + p_7 \sigma_s^2 \mu_m^2 (\beta + p_3 \mu_m^2) \|\hat{\mathbf{w}}_{i-1}\|^2 + p_4 p_7 \sigma_s^2 \mu_m^4 \|\hat{\mathbf{w}}_{i-1}\|^2, \end{aligned} \quad (213)$$

where we define

$$p_5 \triangleq \frac{3B_2 \gamma^4}{(1-\beta)^4}, \quad p_6 \triangleq \frac{3\sigma_s^4}{(1-\beta)^4}, \quad p_7 \triangleq \frac{8}{(1-\beta)^2}. \quad (214)$$

When μ_m is sufficiently small, we obtain from (213):

$$\begin{aligned} & \mathbb{E}[\|\hat{\mathbf{w}}_i\|^4 | \mathcal{F}_{i-1}] \\ &\leq (\beta + 2p_7 \gamma^2 B_1 \mu_m^2) \|\hat{\mathbf{w}}_{i-1}\|^4 + 2p_7 \gamma^2 B_1 \mu_m^2 \|\hat{\mathbf{w}}_{i-1}\|^4 + p_4 p_7 \sigma_s^2 \mu_m^4 \|\hat{\mathbf{w}}_{i-1}\|^2 \\ &\quad + 2p_7 \beta \sigma_s^2 \mu_m^4 \|\hat{\mathbf{w}}_{i-1}\|^2 + p_6 \mu_m^4 \\ &= (\beta + p_8 \mu_m^2) \|\hat{\mathbf{w}}_{i-1}\|^4 + p_8 \mu_m^2 \|\hat{\mathbf{w}}_{i-1}\|^4 + p_9 \mu_m^4 \|\hat{\mathbf{w}}_{i-1}\|^2 + p_{10} \mu_m^2 \|\hat{\mathbf{w}}_{i-1}\|^2 + p_6 \mu_m^4 \end{aligned} \quad (215)$$

where we define

$$p_8 \triangleq 2p_7 \gamma^2 B_1, \quad p_9 \triangleq p_4 p_7 \sigma_s^2, \quad p_{10} \triangleq 2p_7 \beta \sigma_s^2. \quad (216)$$

Combining (205) and (215), we have

$$\left[\frac{\mathbb{E}[\|\hat{\mathbf{w}}_i\|^4 | \mathcal{F}_{i-1}]}{\mathbb{E}[\|\hat{\mathbf{w}}_i\|^4 | \mathcal{F}_{i-1}]} \right] \leq \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} \|\hat{\mathbf{w}}_{i-1}\|^4 \\ \|\hat{\mathbf{w}}_{i-1}\|^2 \end{bmatrix} + \begin{bmatrix} a' & b' \\ c' & d' \end{bmatrix} \begin{bmatrix} e \\ f \end{bmatrix}, \quad (217)$$

where the constants are

$$\begin{aligned} a &\triangleq 1 - \frac{q_1}{2} \mu_m, & b &\triangleq 2q_2 \mu_m, & d &\triangleq q_6 \sigma_s^2 \mu_m^2, & d' &\triangleq q_3 q_6 \sigma_s^2 \mu_m^3, \\ c &\triangleq p_8 \mu_m^2, & d &\triangleq \beta + p_8 \mu_m^2, & c' &\triangleq p_9 \mu_m^4, & d' &\triangleq p_{10} \mu_m^2, \\ e &\triangleq q_5 \mu_m^4, & f &\triangleq p_6 \mu_m^4. \end{aligned} \quad (218)$$

Taking expectations again over \mathcal{F}_{i-1} for both sides of the inequality (217), we have

$$\left[\frac{\mathbb{E}[\|\hat{\mathbf{w}}_i\|^4]}{\mathbb{E}[\|\hat{\mathbf{w}}_i\|^4]} \right] \leq \underbrace{\begin{bmatrix} a & b \\ c & d \end{bmatrix}}_{\mathbf{r}} \begin{bmatrix} \mathbb{E}[\|\hat{\mathbf{w}}_{i-1}\|^4] \\ \mathbb{E}[\|\hat{\mathbf{w}}_{i-1}\|^2] \end{bmatrix} + \begin{bmatrix} a' & b' \\ c' & d' \end{bmatrix} \begin{bmatrix} e \\ f \end{bmatrix}, \quad (219)$$

Recall from Theorem 4 that

$$\limsup_{t \rightarrow \infty} \mathbb{E}[\|\hat{\mathbf{w}}_{i-1}\|^2] = O(\mu_m), \quad \limsup_{t \rightarrow \infty} \mathbb{E}[\|\hat{\mathbf{w}}_{i-1}\|^4] = O(\mu_m^2), \quad (220)$$

then it holds that

$$\limsup_{t \rightarrow \infty} \begin{bmatrix} a' & b' \\ c' & d' \end{bmatrix} \begin{bmatrix} \mathbb{E}[\|\hat{\mathbf{w}}_{i-1}\|^2] \\ \mathbb{E}[\|\hat{\mathbf{w}}_{i-1}\|^4] \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} = \begin{bmatrix} O(\mu_m^3) \\ O(\mu_m^3) \end{bmatrix}. \quad (221)$$

When μ_m is sufficiently small, it can be verified that Γ is stable. Therefore, it holds that

$$\begin{aligned} \left[\limsup_{i \rightarrow \infty} \mathbb{E} \|\tilde{\mathbf{w}}_i\|^4 \right] &= (I - \Gamma)^{-1} \left(\limsup_{i \rightarrow \infty} \begin{bmatrix} a' & b' \\ c' & d' \end{bmatrix} \begin{bmatrix} \mathbb{E} \|\tilde{\mathbf{w}}_{i-1}\|^2 \\ \mathbb{E} \|\tilde{\mathbf{w}}_{i-1}\|^4 \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \right) \\ &= \begin{bmatrix} O(\mu_m^2) \\ O(\mu_m^4) \end{bmatrix}. \end{aligned} \quad (222)$$

Furthermore,

$$\limsup_{i \rightarrow \infty} \mathbb{E} \|\tilde{\mathbf{w}}_i\|^4 \leq 2\nu^4 \limsup_{i \rightarrow \infty} [\mathbb{E} \|\tilde{\mathbf{w}}_i\|^4 + \mathbb{E} \|\mathbf{w}_i\|^4] = O(\mu_m^2). \quad (223)$$

Appendix F. Proof of Corollary 7

Recall from (219) that

$$\underbrace{\begin{bmatrix} \mathbb{E} \|\tilde{\mathbf{w}}_i\|^4 \\ \mathbb{E} \|\tilde{\mathbf{w}}_i\|^4 \end{bmatrix}}_{y_i} \leq \underbrace{\begin{bmatrix} a & b \\ c & d \end{bmatrix}}_{\Gamma_1} \underbrace{\begin{bmatrix} \mathbb{E} \|\tilde{\mathbf{w}}_{i-1}\|^4 \\ \mathbb{E} \|\tilde{\mathbf{w}}_{i-1}\|^4 \end{bmatrix}}_{y_{i-1}} + \underbrace{\begin{bmatrix} a' & b' \\ c' & d' \end{bmatrix}}_{\Gamma_2} \underbrace{\begin{bmatrix} \mathbb{E} \|\tilde{\mathbf{w}}_{i-1}\|^2 \\ \mathbb{E} \|\tilde{\mathbf{w}}_{i-1}\|^2 \end{bmatrix}}_{z_{i-1}} + \underbrace{\begin{bmatrix} e \\ f \end{bmatrix}}_{\tau}. \quad (224)$$

We then have

$$\|y_i\|_1 \leq \|\Gamma_1\|_1 \|y_{i-1}\|_1 + \|\Gamma_2\|_1 \|z_{i-1}\|_1 + \|\tau\|_1. \quad (225)$$

Notice that

$$\|\Gamma_1\|_1 = \max \left\{ 1 - \frac{q_1 \mu_m}{2} + p_8 \mu_m^2, \beta + 2q_2 \mu_m + p_8 \mu_m^2 \right\}, \quad (226)$$

we can always choose μ_m small enough such that

$$\|\Gamma_1\|_1 \leq 1 - \frac{q_1 \mu_m}{4} = 1 - \frac{\mu_m \nu}{4(1-\beta)} \triangleq \rho_2. \quad (227)$$

Similarly, we can also choose μ_m small enough such that

$$\|\Gamma_2\|_1 = \max \{ q_6 \sigma_s^2 \mu_m^2 + p_9 \mu_m^4, q_3 q_6 \sigma_s^2 \mu_m^3 + p_{10} \mu_m^2 \} \leq (q_6 \sigma_s^2 + p_{10}) \mu_m^2. \quad (228)$$

Also recall (174) that

$$\|z_i\|_1 \leq \rho_1^{i+1} \|z_{-1}\|_1 + s_1 \mu_m \stackrel{(a)}{\leq} \rho_2^{i+1} \|z_{-1}\|_1 + s_1 \mu_m \quad (229)$$

where we define

$$s_1 \triangleq \frac{2E_1 \sigma_s^2}{(1-\beta) \nu^4}, \quad (230)$$

for some constant E_1 , and (a) holds because $\rho_1 = 1 - \frac{\mu_m \nu}{2(1-\beta)} \leq \rho_2$. Substituting (227), (228) and (229) into (225), we have

$$\|y_i\|_1 \leq \rho_2 \|y_{i-1}\|_1 + (q_6 \sigma_s^2 + p_{10}) (\rho_2^i \|z_{-1}\|_1 + s_1 \mu_m) \mu_m^2 + 2p_8 \mu_m^4$$

$$\begin{aligned} &= \rho_2 \|y_{i-1}\|_1 + s_2 \rho_2^i \mu_m^2 + s_3 \mu_m^3 + 2p_8 \mu_m^4 \\ &\stackrel{(a)}{\leq} \rho_2 \|y_{i-1}\|_1 + s_2 \rho_2^i \mu_m^2 + 2s_3 \mu_m^3, \end{aligned} \quad (231)$$

where (a) holds when μ_m is sufficiently small, and the constants s_2 and s_3 are defined as

$$s_2 \triangleq (q_6 \sigma_s^2 + p_{10}) \|z_{-1}\|_1, \quad s_3 \triangleq (q_6 \sigma_s^2 + p_{10}) s_1. \quad (232)$$

Iterating (231), we reach

$$\|y_i\|_1 \leq \rho_2^{i+1} \|y_{-1}\|_1 + s_2 (\dot{i} + 1) \rho_2^i \mu_m^2 + \frac{2s_3 \mu_m^3}{1-\rho_2}. \quad (233)$$

Since $\|y_i\|_1 = \mathbb{E} \|\tilde{\mathbf{w}}_i\|^4 + \mathbb{E} \|\mathbf{w}_i\|^4$, we have

$$\begin{aligned} \mathbb{E} \|\tilde{\mathbf{w}}_i\|^4 &\leq \rho_2^{i+1} \|y_{-1}\|_1 + s_2 (\dot{i} + 1) \rho_2^i \mu_m^2 + \frac{2s_3 \mu_m^3}{1-\rho_2} \\ &= \rho_2^{i+1} \|y_{-1}\|_1 + s_2 (\dot{i} + 1) \rho_2^i \mu_m^2 + \frac{8(1-\beta) s_3 \mu_m^2}{\nu} \\ &= \rho_2^{i+1} \|y_{-1}\|_1 + s_2 (\dot{i} + 1) \rho_2^i \mu_m^2 + s_4 \mu_m^2, \end{aligned} \quad (234)$$

where s_4 is defined as

$$s_4 \triangleq \frac{8(1-\beta) s_3}{\nu} \quad (235)$$

Now we substitute (234) and (176) into the second row of (219) and reach

$$\begin{aligned} \mathbb{E} \|\mathbf{w}_i\|^4 &\leq (\beta + p_8 \mu_m^2) \mathbb{E} \|\mathbf{w}_{i-1}\|^4 + p_8 \mu_m^2 [\rho_2^i \|y_{-1}\|_1 + s_2 i \rho_2^{i-1} \mu_m^2 + s_4 \mu_m^2] \\ &\quad + p_{10} \rho_2^i \mu_m^2 \|\mathbf{w}_{i-1}\|^2 + p_9 \mu_m^4 [\rho_2^i \|z_{-1}\|_1 + s_1 \mu_m] + p_8 \mu_m^4. \end{aligned} \quad (236)$$

Using the bounds for $\mathbb{E} \|\mathbf{w}_i\|^2$ from Corollary 5, the above inequality becomes

$$\begin{aligned} \mathbb{E} \|\mathbf{w}_i\|^4 &\leq (\beta + p_8 \mu_m^2) \mathbb{E} \|\mathbf{w}_{i-1}\|^4 + p_8 \mu_m^2 [\rho_2^i \|y_{-1}\|_1 + s_2 i \rho_2^{i-1} \mu_m^2 + s_4 \mu_m^2] \\ &\quad + p_{10} E_2 \mu_m^2 \left(\frac{(\rho_2^2 + \gamma^2) \rho_1^i \mu_m^2}{(1-\beta)^4} + \frac{\sigma_s^2 \mu_m^2}{(1-\beta)^3} \right) \\ &\stackrel{(a)}{\leq} (1 - \frac{1-\beta}{2}) \mathbb{E} \|\mathbf{w}_{i-1}\|^4 + s_5 \rho_2^i \mu_m^2 + s_6 i \rho_2^{i-1} \mu_m^4 + s_7 \mu_m^4 \\ &\quad + s_8 \rho_1^i \mu_m^4 + s_9 \mu_m^4 + s_{10} \rho_2^i \mu_m^4 + p_9 \mu_m^4 + s_{11} p_9 \mu_m^5 \\ &\stackrel{(b)}{\leq} \alpha \mathbb{E} \|\mathbf{w}_{i-1}\|^4 + s_5 \rho_2^i \mu_m^2 + s_6 i \rho_2^{i-1} \mu_m^4 + s_8 \rho_1^i \mu_m^4 + s_{10} \rho_2^i \mu_m^4 \\ &\quad + (s_7 + s_9 + 2p_9) \mu_m^4 \\ &\stackrel{(c)}{\leq} \alpha \mathbb{E} \|\mathbf{w}_{i-1}\|^4 + s_5 \rho_2^i \mu_m^2 + s_6 i \rho_2^{i-1} \mu_m^4 + s_8 \rho_1^i \mu_m^4 + s_{10} \rho_2^i \mu_m^4 \\ &\quad + (s_7 + s_9 + 2p_9) \mu_m^4 \end{aligned} \quad (237)$$

where E_2 is some constant. The inequality (a) holds because step-size μ_m is chosen small enough such that $(1-\beta)/2 > p_8\mu_m^2$, (b) holds because $\alpha = 1 - (1-\beta)/2$ and μ_m is chosen such that $p_6\mu_m^4 > s_1p_9\mu_m^5$, and (c) holds because $\rho_1 < \rho_2$. Moreover, the other constants are defined as

$$\begin{aligned} s_5 &\triangleq p_8\|y_{-1}\|_1, & s_6 &\triangleq p_8s_2, & s_7 &\triangleq p_8s_4 \\ s_8 &\triangleq \frac{p_{10}E_2(\delta^2 + \gamma^2)}{(1-\beta)^4}, & s_9 &\triangleq \frac{p_{10}E_2\sigma_s^2}{(1-\beta)^3}, & s_{10} &\triangleq p_9\|z_{-1}\|_1. \end{aligned} \quad (238)$$

Now we continue iterating (237) and reach

$$\begin{aligned} \mathbb{E}\|\tilde{\mathbf{w}}_i\|^4 &\leq \alpha^{i+1}\mathbb{E}\|\tilde{\mathbf{w}}_{-1}\|^4 + s_5\mu_m^2 \sum_{k=0}^i \left(\frac{\alpha}{\rho_2}\right)^k + s_6\mu_m^4 \sum_{k=0}^{i-1} (i-k) \left(\frac{\alpha}{\rho_2}\right)^k \\ &\quad + s_8\mu_m^4 \rho_2^i \sum_{k=0}^i \left(\frac{\alpha}{\rho_2}\right)^k + s_{10}\mu_m^4 \rho_2^i \sum_{k=0}^i \left(\frac{\alpha}{\rho_2}\right)^k + \frac{2(s_7 + s_9 + 2p_6)\mu_m^4}{1-\beta}. \end{aligned} \quad (239)$$

Recall $\rho_2 = 1 - \frac{\mu_m\nu}{4(1-\beta)}$, and we can choose μ_m small enough such that $\rho_2 > \alpha = 1 - \frac{1-\beta}{2}$. In this situation, we have

$$\frac{\alpha}{\rho_2} < 1, \quad \text{and} \quad \sum_{k=0}^i \left(\frac{\alpha}{\rho_2}\right)^k < \sum_{k=0}^{\infty} \left(\frac{\alpha}{\rho_2}\right)^k = \frac{\rho_2}{\rho_2 - \alpha} \leq \frac{E_3}{1-\beta}, \quad (240)$$

where E_3 is some constant. Meanwhile, we also have

$$\sum_{k=0}^i (i-k) \left(\frac{\alpha}{\rho_2}\right)^k \leq i \sum_{k=0}^i \left(\frac{\alpha}{\rho_2}\right)^k \leq i \sum_{k=0}^{\infty} \left(\frac{\alpha}{\rho_2}\right)^k \leq \frac{iE_3}{1-\beta}. \quad (241)$$

We substitute (240) and (241) into (239), and reach

$$\begin{aligned} \mathbb{E}\|\tilde{\mathbf{w}}_i\|^4 &\leq \rho_2^{i+1}\mathbb{E}\|\tilde{\mathbf{w}}_{-1}\|^4 + \frac{E_3s_5\mu_m^2\rho_2^i}{1-\beta} + \frac{iE_3s_6\mu_m^4\rho_2^{i-1}}{1-\beta} \\ &\quad + \frac{E_3s_8\mu_m^4\rho_2^i}{1-\beta} + \frac{E_3s_{10}\mu_m^4\rho_2^i}{1-\beta} + \frac{2(s_7 + s_9 + 2p_6)\mu_m^4}{1-\beta}. \end{aligned} \quad (242)$$

Recall from (189) that $\mathbb{E}\|\tilde{\mathbf{w}}_{-1}\|^4 = E_4\mu_m^4$, where $E_4 = \mathbb{E}\|\nabla_w Q(\mathbf{w}_{-2}; \boldsymbol{\theta}_{-1})\|^4$. Substituting into (242) we reach that

$$\begin{aligned} \mathbb{E}\|\tilde{\mathbf{w}}_i\|^4 &\leq E_4\mu_m^{i+1} + \frac{E_3s_5\mu_m^2\rho_2^i}{1-\beta} + \frac{iE_3s_6\mu_m^4\rho_2^{i-1}}{1-\beta} \\ &\quad + \frac{E_3s_8\mu_m^4\rho_2^i}{1-\beta} + \frac{E_3s_{10}\mu_m^4\rho_2^i}{1-\beta} + \frac{2(s_7 + s_9 + 2p_6)\mu_m^4}{1-\beta}. \end{aligned} \quad (243)$$

Substituting (238) into (243) and recall $\alpha < \rho_2$, we finally reach

$$\mathbb{E}\|\tilde{\mathbf{w}}_i\|^4 = O\left(\frac{\gamma^2\rho_2^i}{(1-\beta)^3}\mu_m^2 + \rho_2^{i+1}\mu_m^4 + \frac{\gamma^2\sigma_s^2 i \rho_2^{i-1}}{(1-\beta)^5}\mu_m^4 + \frac{\sigma_s^2(\delta^2 + \gamma^2)\rho_2^i}{(1-\beta)^7}\mu_m^4 + \frac{\delta^2\sigma_s^2\rho_2^i}{(1-\beta)^6}\mu_m^4\right)$$

$$+ \frac{\gamma^2\sigma_s^4}{(1-\beta)^5\nu^2}\mu_m^4 + \frac{\sigma_{s,4}^4}{(1-\beta)^6}\mu_m^4 + \frac{\sigma_{s,4}^4}{(1-\beta)^5}\mu_m^4 \Big). \quad (244)$$

Since there must exist some constants E_5, E_6 and E_7 such that

$$\begin{aligned} \frac{\gamma^2\rho_2^i}{(1-\beta)^3}\mu_m^2 &\leq \frac{E_5\gamma^2\rho_2^{i+1}}{(1-\beta)^3}\mu_m^2, \\ \rho_2^{i+1}\mu_m^4 + \frac{\gamma^2\sigma_s^2 i \rho_2^{i-1}}{(1-\beta)^5}\mu_m^4 + \frac{\sigma_s^2(\delta^2 + \gamma^2)\rho_2^i}{(1-\beta)^7}\mu_m^4 + \frac{\delta^2\sigma_s^2\rho_2^i}{(1-\beta)^6}\mu_m^4 &\leq \frac{E_6\sigma_s^2(\delta^2 + \gamma^2)(i+1)\rho_2^{i+1}}{(1-\beta)^7}\mu_m^4, \\ \frac{\gamma^2\sigma_{s,4}^4}{(1-\beta)^5\nu^2}\mu_m^4 + \frac{\sigma_{s,4}^4}{(1-\beta)^6}\mu_m^4 + \frac{\sigma_{s,4}^4}{(1-\beta)^5}\mu_m^4 &\leq \frac{E_7[(\gamma^2 + \nu^2)\sigma_{s,4}^4 + \sigma_{s,4}^4\nu^2]}{(1-\beta)^6\nu^2}\mu_m^4, \end{aligned} \quad (245)$$

we finally reach the conclusion in (44).

Appendix G. Proof of Theorem 8

Subtracting (55) and (61) we get

$$\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_i = (I_M - \mu R_u)(\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}) + \mu(\mathbf{s}_i(\boldsymbol{\psi}_{i-1}) - \mathbf{s}_i(\mathbf{x}_{i-1})) + \mu\beta R_u \tilde{\mathbf{w}}_{i-1}, \quad (246)$$

where

$$\mathbf{s}_i(\boldsymbol{\psi}_{i-1}) = (R_u - \mathbf{u}_i \mathbf{u}_i^\top) \tilde{\boldsymbol{\psi}}_{i-1} - \mathbf{u}_i \mathbf{v}(i), \quad \mathbf{s}_i(\mathbf{x}_{i-1}) = (R_u - \mathbf{u}_i \mathbf{u}_i^\top) \tilde{\mathbf{x}}_{i-1} - \mathbf{u}_i \mathbf{v}(i). \quad (247)$$

Substituting into (246) gives

$$\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_i = (I_M - \mu R_u)(\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}) + \mu(R_u - \mathbf{u}_i \mathbf{u}_i^\top)(\tilde{\boldsymbol{\psi}}_{i-1} - \tilde{\mathbf{x}}_{i-1}) + \mu\beta R_u \tilde{\mathbf{w}}_{i-1}. \quad (248)$$

Now note that in the quadratic case, the Hessian matrix of $J(w)$ is equal to R_u . It follows that condition (11) is satisfied with the identifications $\nu = \lambda_{\min}(R_u)$, $\delta = \lambda_{\max}(R_u)$. Let $t \in (0, 1)$. By squaring (248) and taking expectations, and applying Jensen's inequality, we obtain

$$\begin{aligned} \mathbb{E}\|\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 &\leq \mathbb{E}\|(I_M - \mu R_u)(\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}) + \mu\beta R_u \tilde{\mathbf{w}}_{i-1}\|^2 + \mu^2 \mathbb{E}\|R_u - \mathbf{u}_i \mathbf{u}_i^\top\|^2 \mathbb{E}\|\tilde{\boldsymbol{\psi}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2 \\ &\stackrel{(a)}{\leq} \frac{1}{1-t} (1-\mu\nu)^2 \mathbb{E}\|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2 + \frac{1}{t} \mu^2 \beta^2 \delta^2 \mathbb{E}\|\tilde{\mathbf{w}}_{i-1}\|^2 + B_1 \mu^2 \mathbb{E}\|\tilde{\boldsymbol{\psi}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2 \\ &\stackrel{(b)}{\leq} (1-\mu\nu) \mathbb{E}\|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2 + \frac{\mu\beta^2\delta^2}{\nu} \mathbb{E}\|\tilde{\mathbf{w}}_{i-1}\|^2 + B_1 \mu^2 \mathbb{E}\|\tilde{\boldsymbol{\psi}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2, \end{aligned} \quad (249)$$

where (a) holds because of Jensen's inequality and we let $B_1 = \mathbb{E}\|R_u - \mathbf{u}_i \mathbf{u}_i^\top\|^2$, and (b) holds by choosing $t = \mu\nu$. To bound the last term in the above relation, we use (124) to note that

$$\tilde{\boldsymbol{\psi}}_i - \tilde{\mathbf{x}}_i = \tilde{\mathbf{w}}_i + \beta_1(\tilde{\mathbf{w}}_i - \tilde{\mathbf{w}}_{i-1}) - \tilde{\mathbf{x}}_i = (\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_i) - \beta_1(\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{w}}_i) \quad (250)$$

On the other hand, from (28) we have

$$\hat{\mathbf{w}}_i - \tilde{\mathbf{x}}_i = \frac{1}{1-\beta}(\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_i) - \frac{\beta}{1-\beta}(\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_i) = (\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_i) - \frac{\beta}{1-\beta}(\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{w}}_i). \quad (251)$$

so that

$$\tilde{\psi}_i - \tilde{\mathbf{x}}_i = \hat{\mathbf{w}}_i - \tilde{\mathbf{x}}_i + \frac{\beta - \beta_1 + \beta\beta_1}{1-\beta}(\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{w}}_i) = \hat{\mathbf{w}}_i - \tilde{\mathbf{x}}_i + \frac{\beta'}{1-\beta}(\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{w}}_i) = \hat{\mathbf{w}}_i - \tilde{\mathbf{x}}_i - \beta' \tilde{\mathbf{w}}_i. \quad (252)$$

where we used the definition for β' from (30) and the definition for $\tilde{\mathbf{w}}_i$ from (28). Therefore, from Jensen's inequality again, we get

$$\mathbb{E}\|\tilde{\psi}_i - \tilde{\mathbf{x}}_i\|^2 \leq 2\mathbb{E}\|\hat{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 + 2\beta'^2\mathbb{E}\|\tilde{\mathbf{w}}_i\|^2. \quad (253)$$

Substituting into (249) gives

$$\begin{aligned} \mathbb{E}\|\hat{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 &\leq (1 - \mu\nu + 2B_1\mu^2)\mathbb{E}\|\hat{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2 + \left(\frac{\mu\beta^2\delta^2}{\nu} + 2B_1\beta^2\mu^2\right)\mathbb{E}\|\tilde{\mathbf{w}}_{i-1}\|^2 \\ &\stackrel{(a)}{\leq} \left(1 - \frac{\mu\nu}{2}\right)\mathbb{E}\|\hat{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2 + \frac{2\mu\beta^2\delta^2}{\nu}\mathbb{E}\|\tilde{\mathbf{w}}_{i-1}\|^2 \\ &\leq \left(1 - \frac{\mu\nu}{2}\right)\mathbb{E}\|\hat{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2 + \frac{B_2\mu\delta^2}{\nu}\mathbb{E}\|\tilde{\mathbf{w}}_{i-1}\|^2, \end{aligned} \quad (254)$$

where $B_2 = 2\beta^2$ and the inequality (a) holds when μ is chosen small enough such that

$$\frac{\mu\nu}{2} > 2B_1\mu^2 \quad \text{and} \quad \frac{\mu\beta^2\delta^2}{\nu} > 2B_1\beta^2\mu^2. \quad (255)$$

Recall from Corollary 5 that

$$\mathbb{E}\|\tilde{\mathbf{w}}_i\|^2 \leq C_1 \left(\frac{(\delta^2 + \gamma^2)\rho_1^{i+1}\mu_m^2}{(1-\beta)^4} + \frac{\sigma_s^2\mu_s^2}{(1-\beta)^3} \right) \quad (256)$$

for each iteration $i = 0, 1, 2, 3, \dots$, where C_1 is some constant. Recall $\mu = \mu_m/(1-\beta)$, we then have

$$\mathbb{E}\|\tilde{\mathbf{w}}_i\|^2 \leq C_1 \left(\frac{(\delta^2 + \gamma^2)\rho_1^{i+1}\mu^2}{(1-\beta)^2} + \frac{\sigma_s^2\mu^2}{1-\beta} \right) \quad (257)$$

This fact, together with inequality (254), leads to

$$\mathbb{E}\|\hat{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 \leq \left(1 - \frac{\mu\nu}{2}\right)\mathbb{E}\|\hat{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2 + B_2C_1 \left(\frac{\delta^2(\delta^2 + \gamma^2)\rho_1^i\mu^3}{\nu(1-\beta)^2} + \frac{\delta^2\sigma_s^2\mu^3}{\nu(1-\beta)} \right). \quad (258)$$

Recall from Corollary 5 that $\rho_1 = 1 - \frac{\mu\nu}{2(1-\beta)} = 1 - \frac{\mu\nu}{2}$, then (258) becomes

$$\mathbb{E}\|\hat{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 \leq \rho_1\mathbb{E}\|\hat{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2 + B_2C_1 \left(\frac{\delta^2(\delta^2 + \gamma^2)\rho_1^i\mu^3}{\nu(1-\beta)^2} + \frac{\delta^2\sigma_s^2\mu^3}{\nu(1-\beta)} \right). \quad (259)$$

For brevity, we denote

$$e_1 \triangleq \frac{B_2C_1(\delta^2 + \gamma^2)\delta^2}{\nu(1-\beta)^2}, \quad e_2 \triangleq \frac{B_2C_1\delta^2\sigma_s^2}{\nu(1-\beta)}. \quad (260)$$

Inequality (259) will become

$$\begin{aligned} \mathbb{E}\|\hat{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 &\leq \rho_1\mathbb{E}\|\hat{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2 + e_1\rho_1^i\mu^3 + e_2\mu^3 \\ &\leq \rho_1^{i+1}\mathbb{E}\|\hat{\mathbf{w}}_{-1} - \tilde{\mathbf{x}}_{-1}\|^2 + e_1(i+1)\rho_1^i\mu^3 + \frac{e_2\mu^3}{1-\rho_1}. \end{aligned} \quad (261)$$

Recall from the first equation in (251) that for $i = -1$:

$$\hat{\mathbf{w}}_{-1} - \tilde{\mathbf{x}}_{-1} = \frac{1}{1-\beta}(\tilde{\mathbf{w}}_{-1} - \tilde{\mathbf{x}}_{-1}) - \frac{\beta}{1-\beta}(\tilde{\mathbf{w}}_{-2} - \tilde{\mathbf{x}}_{-1}). \quad (262)$$

Now, using the assumption that the momentum and standard recursions started from the same initial states, $\mathbf{w}_{-2} = \mathbf{x}_{-1}$ and $\mathbf{w}_{-1} = \mathbf{w}_{-2} - \mu_m\nabla_w Q(\mathbf{w}_{-2}; \mathbf{d}(-1), \mathbf{u}_{-1})$, and recall $\mu = \mu_m/(1-\beta)$, then we have

$$\hat{\mathbf{w}}_{-1} - \tilde{\mathbf{x}}_{-1} = \frac{1}{1-\beta}(\tilde{\mathbf{w}}_{-1} - \tilde{\mathbf{w}}_{-2}) = \mu\nabla_w Q(\mathbf{w}_{-2}; \mathbf{d}(-1), \mathbf{u}_{-1}). \quad (263)$$

Therefore, it holds that

$$\mathbb{E}\|\hat{\mathbf{w}}_{-1} - \tilde{\mathbf{x}}_{-1}\|^2 = B_4\mu^2, \quad (264)$$

where $B_4 = \mathbb{E}\|\nabla_w Q(\mathbf{w}_{-2}; \mathbf{d}(-1), \mathbf{u}_{-1})\|^2$. Substituting (264) and (260) into (261), we reach

$$\mathbb{E}\|\hat{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 \leq B_4\rho_1^{i+1}\mu^2 + \frac{B_2C_1(\delta^2 + \gamma^2)\delta^2(i+1)\rho_1^i}{\nu(1-\beta)^2}\mu^3 + \frac{4B_2C_1\delta^2\sigma_s^2}{\nu^2(1-\beta)}\mu^2. \quad (265)$$

Furthermore, using (251) and $\tilde{\mathbf{w}}_i = \frac{\tilde{\mathbf{w}}_i - \tilde{\mathbf{w}}_{i-1}}{1-\beta}$ from (28) we have

$$\hat{\mathbf{w}}_i - \tilde{\mathbf{x}}_i = (\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_i) + \beta\tilde{\mathbf{w}}_i, \quad (266)$$

which implies that

$$\mathbb{E}\|\hat{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 \leq 2\mathbb{E}\|\hat{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 + 2\beta^2\mathbb{E}\|\tilde{\mathbf{w}}_i\|^2 \leq 2\mathbb{E}\|\hat{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 + 2\mathbb{E}\|\tilde{\mathbf{w}}_i\|^2. \quad (267)$$

Substituting (257) and (265) into (267), we have

$$\begin{aligned} \mathbb{E}\|\hat{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 &\leq B_4\rho_1^{i+1}\mu^2 + \frac{B_2C_1(\delta^2 + \gamma^2)\delta^2(i+1)\rho_1^i}{\nu(1-\beta)^2}\mu^3 + \frac{4B_2C_1\delta^2\sigma_s^2}{\nu^2(1-\beta)}\mu^2 \\ &\quad + 2C_1 \left(\frac{(\delta^2 + \gamma^2)\rho_1^{i+1}\mu^2}{(1-\beta)^2} + \frac{\sigma_s^2\mu^2}{1-\beta} \right) \\ &= O \left(\frac{\delta^2 + \gamma^2}{(1-\beta)^2}\rho_1^{i+1}\mu^2 + \frac{\delta^2(\delta^2 + \gamma^2)(i+1)\rho_1^{i+1}}{\nu(1-\beta)^2}\mu^3 + \frac{\delta^2\sigma_s^2\mu^2}{\nu^2(1-\beta)} \right). \end{aligned} \quad (268)$$

Appendix H. Verifying Assumptions 5 and 6

Least-mean-squares problem. Consider first the mean-squares cost (47). Since in this case $\mathbf{H}_{i-1} = \mathbf{R}_{i-1} = \mathbf{R}_u$, we find that Assumption 6 holds automatically. With regards to Assumption 5, at any iteration i , we have

$$\mathbf{s}_i(\mathbf{w}_{i-1}) - \mathbf{s}_i(\mathbf{x}_{i-1}) = (\mathbf{R}_u - \mathbf{u}_i \mathbf{u}_i^\top)(\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}). \quad (269)$$

so that, under the assumption of independent and stationary regression vectors,

$$\mathbb{E}[\|\mathbf{s}_i(\mathbf{w}_{i-1}) - \mathbf{s}_i(\mathbf{x}_{i-1})\|^2 | \mathcal{F}_{i-1}] \leq \xi_1 \|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2, \quad (270)$$

where $\xi_1 = \mathbb{E}\|\mathbf{R}_u - \mathbf{u}_i \mathbf{u}_i^\top\|^2$. Similarly,

$$\mathbb{E}[\|\mathbf{s}_i(\mathbf{w}_{i-1}) - \mathbf{s}_i(\mathbf{x}_{i-1})\|^4 | \mathcal{F}_{i-1}] \leq \xi_2 \|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^4, \quad (271)$$

where $\xi_2 = \mathbb{E}\|\mathbf{R}_u - \mathbf{u}_i \mathbf{u}_i^\top\|^4$. Therefore, Assumption 5 holds.

Regularized logistic regression. Consider next the regularized logistic regression risk

$$J(w) \triangleq \frac{\rho}{2} \|w\|^2 + \mathbb{E} \left\{ \ln [1 + \exp(-\gamma(i) \mathbf{h}_i^\top w)] \right\}, \quad (272)$$

where $\mathbf{h}_i \in \mathbb{R}^M$ is a streaming sequence of independent feature vectors with $R_h = \mathbb{E} \mathbf{h}_i \mathbf{h}_i^\top > 0$, and $\gamma(i) \in \{-1, +1\}$ is a streaming sequence of class labels. We assume the random processes $\{\gamma(i), \mathbf{h}_i\}$ are wide-sense stationary. Moreover, $\rho > 0$ is a regularization parameter. We first verify the feasibility of Assumption 5. Note that the approximate gradient vector is given by:

$$\widehat{\nabla_w J}(w) = \rho w - \frac{\exp(-\gamma(i) \mathbf{h}_i^\top w)}{1 + \exp(-\gamma(i) \mathbf{h}_i^\top w)} \gamma(i) \mathbf{h}_i \quad (273)$$

and, hence,

$$\begin{aligned} & \widehat{\nabla_w J}(\psi_{i-1}) - \widehat{\nabla_w J}(\mathbf{x}_{i-1}) \\ & \leq \rho \|\psi_{i-1} - \mathbf{x}_{i-1}\| + \|\mathbf{h}_i\| \left\| \frac{\exp(-\gamma(i) \mathbf{h}_i^\top \psi_{i-1})}{1 + \exp(-\gamma(i) \mathbf{h}_i^\top \psi_{i-1})} - \frac{\exp(-\gamma(i) \mathbf{h}_i^\top \mathbf{x}_{i-1})}{1 + \exp(-\gamma(i) \mathbf{h}_i^\top \mathbf{x}_{i-1})} \right\| \end{aligned} \quad (274)$$

Note that

$$\begin{aligned} & \left\| \frac{\exp(-\gamma(i) \mathbf{h}_i^\top \psi_{i-1})}{1 + \exp(-\gamma(i) \mathbf{h}_i^\top \psi_{i-1})} - \frac{\exp(-\gamma(i) \mathbf{h}_i^\top \mathbf{x}_{i-1})}{1 + \exp(-\gamma(i) \mathbf{h}_i^\top \mathbf{x}_{i-1})} \right\| \\ & = \left\| \frac{\exp(-\gamma(i) \mathbf{h}_i^\top \psi_{i-1}) - \exp(-\gamma(i) \mathbf{h}_i^\top \mathbf{x}_{i-1})}{[1 + \exp(-\gamma(i) \mathbf{h}_i^\top \psi_{i-1})][1 + \exp(-\gamma(i) \mathbf{h}_i^\top \mathbf{x}_{i-1})]} \right\| \\ & \leq \left\| \frac{\exp(-\gamma(i) \mathbf{h}_i^\top \psi_{i-1}) - \exp(-\gamma(i) \mathbf{h}_i^\top \mathbf{x}_{i-1})}{\exp(-\gamma(i) \mathbf{h}_i^\top \psi_{i-1}) + \exp(-\gamma(i) \mathbf{h}_i^\top \mathbf{x}_{i-1})} \right\| \\ & = \left\| \frac{\exp(\gamma(i) \mathbf{h}_i^\top \frac{\mathbf{x}_{i-1} - \psi_{i-1}}{2}) - \exp(-\gamma(i) \mathbf{h}_i^\top \frac{\mathbf{x}_{i-1} - \psi_{i-1}}{2})}{\exp(\gamma(i) \mathbf{h}_i^\top \frac{\mathbf{x}_{i-1} - \psi_{i-1}}{2}) + \exp(-\gamma(i) \mathbf{h}_i^\top \frac{\mathbf{x}_{i-1} - \psi_{i-1}}{2})} \right\| \end{aligned}$$

$$= \left| \tanh \left(\gamma(i) \mathbf{h}_i^\top (\mathbf{x}_{i-1} - \psi_{i-1}) / 2 \right) \right| \leq \frac{1}{2} \|\mathbf{h}_i\| \|\psi_{i-1} - \mathbf{x}_{i-1}\|. \quad (275)$$

where in the last inequality we used the property $|\tanh(y)| \leq |y|$, $\forall y \in \mathbb{R}$. Substituting (275) into (274), we get

$$\left\| \widehat{\nabla_w J}(\psi_{i-1}) - \widehat{\nabla_w J}(\mathbf{x}_{i-1}) \right\| \leq \eta_{i,i} \|\psi_{i-1} - \mathbf{x}_{i-1}\|, \quad (276)$$

where $\eta_{i,i} = \rho + \|\mathbf{h}_i\|^2 / 2$ is a random variable.

On the other hand, it is shown in Eq. (2.20) of (Sayed, 2014a) that the Hessian matrix $\nabla_w^2 J(w)$ is upper bounded by δJ_M , where $\delta = (\rho + \lambda_{\max}(R_h))$. We conclude from Lemma E.3 in the same reference that $\nabla_w J(w)$ is Lipschitz continuous with modulus δ , i.e.,

$$\left\| \nabla_w J(\psi_{i-1}) - \nabla_w J(\mathbf{x}_{i-1}) \right\| \leq \delta \|\psi_{i-1} - \mathbf{x}_{i-1}\|. \quad (277)$$

Combining these results we get

$$\begin{aligned} \|\mathbf{s}_i(\psi_{i-1}) - \mathbf{s}_i(\mathbf{x}_{i-1})\| &= \left\| \widehat{\nabla_w J}(\psi_{i-1}) - \widehat{\nabla_w J}(\mathbf{x}_{i-1}) - [\nabla_w J(\psi_{i-1}) - \nabla_w J(\mathbf{x}_{i-1})] \right\| \\ &\leq \eta_i \|\psi_{i-1} - \mathbf{x}_{i-1}\|. \end{aligned} \quad (278)$$

where $\eta_i = \eta_{1,i} + \delta$ is a random variable. Since the $\{\mathbf{h}_i\}$ are independent feature vectors and η_i is only related to \mathbf{h}_i , it follows that

$$\mathbb{E}[\|\mathbf{s}_i(\psi_{i-1}) - \mathbf{s}_i(\mathbf{x}_{i-1})\|^2 | \mathcal{F}_{i-1}] \leq \xi_1 \|\psi_{i-1} - \mathbf{x}_{i-1}\|^2, \quad (279)$$

$$\mathbb{E}[\|\mathbf{s}_i(\psi_{i-1}) - \mathbf{s}_i(\mathbf{x}_{i-1})\|^4 | \mathcal{F}_{i-1}] \leq \xi_2 \|\psi_{i-1} - \mathbf{x}_{i-1}\|^4, \quad (280)$$

where $\xi_1 = \mathbb{E} \eta_i^2$ and $\xi_2 = \mathbb{E} \eta_i^4$.

Next we check the feasibility of Assumption 6. For simplicity, we write γ instead of $\gamma(i)$. It can be verified that for the cost function $J(w)$ in (272):

$$\nabla_w^2 J(w) = \rho J_M + \mathbb{E} \left\{ \mathbf{h}_i \mathbf{h}_i^\top \left(\frac{\exp(-\gamma \mathbf{h}_i^\top w)}{[1 + \exp(-\gamma \mathbf{h}_i^\top w)]^2} \right) \right\}. \quad (281)$$

Now, for any two variables w_1 and w_2 we have

$$\begin{aligned} & \left\| \nabla_w^2 J(w_1) - \nabla_w^2 J(w_2) \right\| \\ &= \left\| \mathbb{E} \left\{ \mathbf{h}_i \mathbf{h}_i^\top \left(\frac{\exp(-\gamma \mathbf{h}_i^\top w_1)}{[1 + \exp(-\gamma \mathbf{h}_i^\top w_1)]^2} - \frac{\exp(-\gamma \mathbf{h}_i^\top w_2)}{[1 + \exp(-\gamma \mathbf{h}_i^\top w_2)]^2} \right) \right\} \right\| \\ &\leq \mathbb{E} \left\| \mathbf{h}_i \mathbf{h}_i^\top \left(\frac{\exp(-\gamma \mathbf{h}_i^\top w_1)}{[1 + \exp(-\gamma \mathbf{h}_i^\top w_1)]^2} - \frac{\exp(-\gamma \mathbf{h}_i^\top w_2)}{[1 + \exp(-\gamma \mathbf{h}_i^\top w_2)]^2} \right) \right\| \\ &\leq \mathbb{E} \left\{ \left\| \mathbf{h}_i \mathbf{h}_i^\top \right\| \left\| \frac{\exp(-\gamma \mathbf{h}_i^\top w_1)}{[1 + \exp(-\gamma \mathbf{h}_i^\top w_1)]^2} - \frac{\exp(-\gamma \mathbf{h}_i^\top w_2)}{[1 + \exp(-\gamma \mathbf{h}_i^\top w_2)]^2} \right\| \right\} \end{aligned} \quad (282)$$

Let $\mathbf{x}_1 = -\gamma \mathbf{h}_i^\top w_1$ and $\mathbf{x}_2 = -\gamma \mathbf{h}_i^\top w_2$. Then,

$$\left\| \frac{\exp(-\gamma \mathbf{h}_i^\top w_1)}{[1 + \exp(-\gamma \mathbf{h}_i^\top w_1)]^2} - \frac{\exp(-\gamma \mathbf{h}_i^\top w_2)}{[1 + \exp(-\gamma \mathbf{h}_i^\top w_2)]^2} \right\|$$

$$\begin{aligned}
&= \left\| \frac{\exp(\mathbf{x}_1)}{[1 + \exp(\mathbf{x}_1)]^2} - \frac{\exp(\mathbf{x}_2)}{[1 + \exp(\mathbf{x}_2)]^2} \right\| \\
&= \left\| \frac{\exp(\mathbf{x}_1)[1 + \exp(\mathbf{x}_2)]^2 - \exp(\mathbf{x}_2)[1 + \exp(\mathbf{x}_1)]^2}{[1 + \exp(\mathbf{x}_1)]^2[1 + \exp(\mathbf{x}_2)]^2} \right\| \\
&\stackrel{(a)}{\leq} \left\| \frac{\exp(-\mathbf{x}_2) - \exp(-\mathbf{x}_1) + \exp(\mathbf{x}_2) - \exp(\mathbf{x}_1)}{2(\exp(-\mathbf{x}_2) + \exp(-\mathbf{x}_1) + \exp(\mathbf{x}_2) + \exp(\mathbf{x}_1))} \right\| \\
&\leq \left\| \frac{\exp(-\mathbf{x}_2) - \exp(-\mathbf{x}_1)}{2(\exp(-\mathbf{x}_2) + \exp(-\mathbf{x}_1))} \right\| + \left\| \frac{\exp(\mathbf{x}_2) - \exp(\mathbf{x}_1)}{2(\exp(-\mathbf{x}_2) + \exp(-\mathbf{x}_1) + \exp(\mathbf{x}_2) + \exp(\mathbf{x}_1))} \right\| \\
&\leq \left\| \frac{\exp(-\mathbf{x}_2) - \exp(-\mathbf{x}_1)}{2(\exp(-\mathbf{x}_2) + \exp(-\mathbf{x}_1))} \right\| + \left\| \frac{\exp(\mathbf{x}_2) - \exp(\mathbf{x}_1)}{2(\exp(\mathbf{x}_2) + \exp(\mathbf{x}_1))} \right\| \\
&\stackrel{(b)}{=} \frac{1}{2} \left\| \frac{\exp(-\frac{\mathbf{x}_2 - \mathbf{x}_1}{2}) - \exp(\frac{\mathbf{x}_2 - \mathbf{x}_1}{2})}{\exp(-\frac{\mathbf{x}_2 - \mathbf{x}_1}{2}) + \exp(\frac{\mathbf{x}_2 - \mathbf{x}_1}{2})} \right\| + \frac{1}{2} \left\| \frac{\exp(\frac{\mathbf{x}_2 - \mathbf{x}_1}{2}) - \exp(-\frac{\mathbf{x}_2 - \mathbf{x}_1}{2})}{\exp(\frac{\mathbf{x}_2 - \mathbf{x}_1}{2}) + \exp(-\frac{\mathbf{x}_2 - \mathbf{x}_1}{2})} \right\| \\
&= \left\| \tanh(\frac{\mathbf{x}_2 - \mathbf{x}_1}{2}) \right\|,
\end{aligned} \tag{283}$$

where (a) holds because of the following two facts:

$$\begin{aligned}
&\exp(\mathbf{x}_1)[1 + \exp(\mathbf{x}_2)]^2 - \exp(\mathbf{x}_2)[1 + \exp(\mathbf{x}_1)]^2 \\
&= \exp(\mathbf{x}_1) + \exp(\mathbf{x}_1 + 2\mathbf{x}_2) - \exp(\mathbf{x}_2) - \exp(\mathbf{x}_2 + 2\mathbf{x}_1) \\
&= \exp(\mathbf{x}_1 + \mathbf{x}_2)[\exp(-\mathbf{x}_2) + \exp(\mathbf{x}_2) - \exp(-\mathbf{x}_1) - \exp(\mathbf{x}_1)],
\end{aligned} \tag{284}$$

and

$$\begin{aligned}
&[1 + \exp(\mathbf{x}_1)]^2[1 + \exp(\mathbf{x}_2)]^2 \\
&= (1 + 2\exp(\mathbf{x}_1) + \exp(2\mathbf{x}_1))(1 + 2\exp(\mathbf{x}_2) + \exp(2\mathbf{x}_2)) \\
&\geq 2\exp(\mathbf{x}_1) + 2\exp(\mathbf{x}_2) + 2\exp(\mathbf{x}_1 + 2\mathbf{x}_2) + 2\exp(\mathbf{x}_2 + 2\mathbf{x}_1) \\
&= 2\exp(\mathbf{x}_1 + \mathbf{x}_2)[\exp(-\mathbf{x}_2) + \exp(\mathbf{x}_2) + \exp(-\mathbf{x}_1) + \exp(\mathbf{x}_1)].
\end{aligned} \tag{285}$$

In addition, (b) holds if we extract $\exp(-\frac{\mathbf{x}_1 \pm \mathbf{x}_2}{2})$ and $\exp(\frac{\mathbf{x}_1 \pm \mathbf{x}_2}{2})$ from both the denominator and numerator of the first and second terms respectively.

Using the definitions for \mathbf{x}_1 and \mathbf{x}_2 , this last expression gives

$$\left\| \tanh(\frac{\mathbf{x}_2 - \mathbf{x}_1}{2}) \right\| = \left\| \tanh\left(\frac{1}{2}\gamma\mathbf{h}_i^\top(w_2 - w_1)\right) \right\| \leq \frac{1}{2}\|\mathbf{h}_i\|\|w_2 - w_1\|. \tag{286}$$

Substituting (286) into (282), we obtain $\|\nabla_w^2 J(w_1) - \nabla_w^2 J(w_2)\| \leq \kappa\|w_1 - w_2\|$, where $\kappa = \mathbb{E}\|\mathbf{h}_i\mathbf{h}_i^\top\| \|\mathbf{h}_i\|/2$. Therefore, Assumption 6 holds.

Appendix I. Proof of Lemma 9

Referring to relation (68) and apply the inequality (194), we reach

$$\begin{aligned}
&\mathbb{E}\|\hat{w}_i - \tilde{w}_i\|^4 \|\mathcal{F}_{i-1}\| \\
&= \|\mathbf{I}_M - \mu\mathbf{H}_{i-1}\|(\hat{w}_{i-1} - \tilde{w}_{i-1}) + \mu(\mathbf{R}_{i-1} - \mathbf{H}_{i-1})\tilde{w}_{i-1} + \mu\beta^3\mathbf{H}_{i-1}\tilde{w}_{i-1}\|^4
\end{aligned}$$

$$\begin{aligned}
&+ 3\mu^4\mathbb{E}\|s_i(\psi_{i-1}) - s_i(\mathbf{x}_{i-1})\|^4 \|\mathcal{F}_{i-1}\| + 8\mu^2\|(I_M - \mu\mathbf{H}_{i-1})(\hat{w}_{i-1} - \tilde{w}_{i-1}) \\
&+ \mu(\mathbf{R}_{i-1} - \mathbf{H}_{i-1})\tilde{w}_{i-1} + \mu\beta^3\mathbf{H}_{i-1}\tilde{w}_{i-1}\|^2\mathbb{E}\|s_i(\psi_{i-1}) - s_i(\mathbf{x}_{i-1})\|^2 \|\mathcal{F}_{i-1}\| \\
&\leq \frac{1}{(1-t)^3}\|(I_M - \mu\mathbf{H}_{i-1})(\hat{w}_{i-1} - \tilde{w}_{i-1})\|^4 + \frac{8\mu^4}{t^3}\|\mathbf{R}_{i-1} - \mathbf{H}_{i-1}\|\|\tilde{w}_{i-1}\|^4 + \frac{8\mu^4\beta^4}{t^3}\|\mathbf{H}_{i-1}w_{i-1}\|^4 \\
&+ 3\mu^4\mathbb{E}\|s_i(\psi_{i-1}) - s_i(\mathbf{x}_{i-1})\|^4 \|\mathcal{F}_{i-1}\| + 8\mu^2\left(\frac{1}{1-t}\|(I_M - \mu\mathbf{H}_{i-1})(\hat{w}_{i-1} - \tilde{w}_{i-1})\|^2\right. \\
&\left. + \frac{2\mu^2}{t}\|\mathbf{R}_{i-1} - \mathbf{H}_{i-1}\|\|\tilde{w}_{i-1}\|^2 + \frac{2\mu^2\beta^2}{t}\|\mathbf{H}_{i-1}w_{i-1}\|^2\right)\mathbb{E}\|s_i(\psi_{i-1}) - s_i(\mathbf{x}_{i-1})\|^2 \|\mathcal{F}_{i-1}\| \\
&\stackrel{(b)}{\leq} (1 - \mu\nu)\|\hat{w}_{i-1} - \tilde{w}_{i-1}\|^4 + \frac{8\mu}{\nu^3}\|\mathbf{R}_{i-1} - \mathbf{H}_{i-1}\|\|\tilde{w}_{i-1}\|^4 + \frac{8\mu\beta^4\delta^4}{\nu^3}\|\tilde{w}_{i-1}\|^4 \\
&+ 3\mu^4\mathbb{E}\|s_i(\psi_{i-1}) - s_i(\mathbf{x}_{i-1})\|^4 \|\mathcal{F}_{i-1}\| + 8\mu^2\left((1 - \mu\nu)\|\hat{w}_{i-1} - \tilde{w}_{i-1}\|^2\right. \\
&\left. + \frac{2\mu}{\nu}\|(\mathbf{R}_{i-1} - \mathbf{H}_{i-1})\tilde{w}_{i-1}\|^2 + \frac{2\mu\beta^2\delta^2}{\nu}\|w_{i-1}\|^2\right)\mathbb{E}\|s_i(\psi_{i-1}) - s_i(\mathbf{x}_{i-1})\|^2 \|\mathcal{F}_{i-1}\|,
\end{aligned} \tag{287}$$

where (a) holds because the facts that for any $a, b, c \in \mathbb{R}^m$,

$$\begin{aligned}
\|a + b + c\|^4 &= \|(1-t)\frac{1}{1-t}a + t\frac{1}{t}(b + c)\|^4 \\
&\leq (1-t)\|\frac{1}{1-t}a\|^4 + t\|\frac{1}{t}(b + c)\|^4 = \frac{1}{(1-t)^3}\|a\|^4 + \frac{1}{t^3}\|b + c\|^4 \\
&\leq \frac{1}{(1-t)^3}\|a\|^4 + \frac{8}{t^3}\|b\|^4 + \frac{8}{t^3}\|c\|^4,
\end{aligned} \tag{288}$$

and

$$\begin{aligned}
\|a + b + c\|^2 &= \|(1-t)\frac{1}{1-t}a + t\frac{1}{t}(b + c)\|^2 \\
&\leq (1-t)\|\frac{1}{1-t}a\|^2 + t\|\frac{1}{t}(b + c)\|^2 = \frac{1}{1-t}\|a\|^2 + \frac{1}{t}\|b + c\|^2 \\
&\leq \frac{1}{1-t}\|a\|^2 + \frac{2}{t}\|b\|^2 + \frac{2}{t}\|c\|^2,
\end{aligned} \tag{289}$$

In addition, (b) holds by choosing $t = \mu\nu$.

To further simplify inequality (287), we first note that

$$\begin{aligned}
\|\mathbf{R}_{i-1} - \mathbf{H}_{i-1}\|^2 &\leq 2(\|\mathbf{R}_{i-1}\|^2 + \|\mathbf{H}_{i-1}\|^2) \leq 4\delta^2, \\
\|\mathbf{R}_{i-1} - \mathbf{H}_{i-1}\|^4 &\leq 8(\|\mathbf{R}_{i-1}\|^4 + \|\mathbf{H}_{i-1}\|^4) \leq 16\delta^4.
\end{aligned} \tag{290}$$

As a result, we have

$$\begin{aligned}
\frac{8\mu}{\nu^3}\|\mathbf{R}_{i-1} - \mathbf{H}_{i-1}\|\|\tilde{w}_{i-1}\|^4 &\leq \alpha_1\mu\|\tilde{w}_{i-1}\|^4, \\
\frac{2\mu}{\nu}\|(\mathbf{R}_{i-1} - \mathbf{H}_{i-1})\tilde{w}_{i-1}\|^2 &\leq \alpha_2\mu\|\tilde{w}_{i-1}\|^2,
\end{aligned} \tag{291}$$

where we define

$$\alpha_1 \triangleq 128\delta^4/\nu^3, \quad \alpha_2 \triangleq 8\delta^2/\nu. \tag{293}$$

On the other hand, from conditions (72)–(73), we have

$$3\mu^4\mathbb{E}\|s_i(\psi_{i-1}) - s_i(\mathbf{x}_{i-1})\|^4\|\mathcal{F}_{i-1}\| \leq 3\xi_2\mu^4\|\tilde{\psi}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^4, \quad (294)$$

$$8\mu^2\mathbb{E}\|s_i(\psi_{i-1}) - s_i(\mathbf{x}_{i-1})\|^2\|\mathcal{F}_{i-1}\| \leq 8\xi_1\mu^2\|\tilde{\psi}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2, \quad (295)$$

In addition, from (252) we get

$$\|\tilde{\psi}_i - \tilde{\mathbf{x}}_i\|^2 \leq 2\|\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 + 2\beta^2\|\tilde{\mathbf{w}}_i\|^2, \quad \|\tilde{\psi}_i - \tilde{\mathbf{x}}_i\|^4 \leq 8\|\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^4 + 8\beta^4\|\tilde{\mathbf{w}}_i\|^4. \quad (296)$$

Combining (294)–(296), we have

$$3\mu^4\mathbb{E}\|s_i(\psi_{i-1}) - s_i(\mathbf{x}_{i-1})\|^4\|\mathcal{F}_{i-1}\| \leq 24\xi_2\mu^4\|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^4 + 24\xi_2\mu^4\|\tilde{\mathbf{w}}_{i-1}\|^4, \quad (297)$$

$$8\mu^2\mathbb{E}\|s_i(\psi_{i-1}) - s_i(\mathbf{x}_{i-1})\|^2\|\mathcal{F}_{i-1}\| \leq 16\xi_1\mu^2\|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2 + 16\xi_1\mu^2\|\tilde{\mathbf{w}}_{i-1}\|^2, \quad (298)$$

In this way, relation (287) becomes

$$\begin{aligned} & \mathbb{E}\|\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^4\|\mathcal{F}_{i-1}\| \\ & \leq (1 - \mu\nu)\|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^4 + a_1\mu\|\tilde{\mathbf{w}}_{i-1}\|^4 + a_3\mu\|\tilde{\mathbf{w}}_{i-1}\|^4 + [(1 - \mu\nu)\|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2 \\ & \quad + a_4\mu\|\tilde{\mathbf{w}}_{i-1}\|^2 + a_2\mu\|\tilde{\mathbf{x}}_{i-1}\|^2][a_5\mu^2\|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2 + a_5\mu^2\|\tilde{\mathbf{w}}_{i-1}\|^2] \\ & \quad + a_6\mu^4\|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^4 + a_6\mu^4\|\tilde{\mathbf{w}}_{i-1}\|^4 \\ & \leq (1 - \mu\nu)\|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^4 + a_1\mu\|\tilde{\mathbf{x}}_{i-1}\|^4 + a_3\mu\|\tilde{\mathbf{w}}_{i-1}\|^4 + a_5(1 - \mu\nu)\mu^2\|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^4 \\ & \quad + a_5\mu^2(1 - \mu\nu + a_4\mu)\|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2\|\tilde{\mathbf{w}}_{i-1}\|^2 + a_4a_5\mu^3\|\tilde{\mathbf{w}}_{i-1}\|^4 \\ & \quad + a_2a_5\mu^3\|\tilde{\mathbf{x}}_{i-1}\|^2\|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2 + a_2a_5\mu^3\|\tilde{\mathbf{x}}_{i-1}\|^2\|\tilde{\mathbf{w}}_{i-1}\|^2 \\ & \quad + a_6\mu^4\|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^4 + a_6\mu^4\|\tilde{\mathbf{w}}_{i-1}\|^4 \\ & \stackrel{(a)}{\leq} (1 - \mu\nu)\|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^4 + a_1\mu\|\tilde{\mathbf{w}}_{i-1}\|^4 + a_3\mu\|\tilde{\mathbf{w}}_{i-1}\|^4 + a_5(1 - \mu\nu)\mu^2\|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^4 \\ & \quad + a_5\mu^2(1 - \mu\nu + a_4\mu)\|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^4 + a_5\mu^2(1 - \mu\nu + a_4\mu)\|\tilde{\mathbf{w}}_{i-1}\|^4 + a_4a_5\mu^3\|\tilde{\mathbf{w}}_{i-1}\|^4 \\ & \quad + a_2a_5\mu^3\|\tilde{\mathbf{x}}_{i-1}\|^4 + a_2a_5\mu^3\|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^4 + a_2a_5\mu^3\|\tilde{\mathbf{x}}_{i-1}\|^4 + a_2a_5\mu^3\|\tilde{\mathbf{w}}_{i-1}\|^4 \\ & \quad + a_6\mu^4\|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^4 + a_6\mu^4\|\tilde{\mathbf{w}}_{i-1}\|^4 \\ & \leq (1 - \frac{\mu\nu}{2})\|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^4 + 2a_1\mu\|\tilde{\mathbf{x}}_{i-1}\|^4 + 2a_3\mu\|\tilde{\mathbf{w}}_{i-1}\|^4. \end{aligned} \quad (299)$$

where we define

$$a_3 \triangleq \frac{8\beta^4\delta^4}{\nu^4}, \quad a_4 \triangleq \frac{2\beta^2\delta^2}{\nu}, \quad a_5 \triangleq 16\xi_1, \quad a_6 \triangleq 24\xi_2. \quad (300)$$

Taking expectations over \mathcal{F}_{i-1} for both sides of (299), we have

$$\begin{aligned} \mathbb{E}\|\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^4 & \leq (1 - \frac{\mu\nu}{2})\mathbb{E}\|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^4 + 2a_1\mu\mathbb{E}\|\tilde{\mathbf{x}}_{i-1}\|^4 + 2a_3\mu\mathbb{E}\|\tilde{\mathbf{w}}_{i-1}\|^4 + 2a_3\mu\mathbb{E}\|\tilde{\mathbf{w}}_{i-1}\|^4 \\ & = \rho_1\mathbb{E}\|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^4 + 2a_1\mu\mathbb{E}\|\tilde{\mathbf{x}}_{i-1}\|^4 + 2a_3\mu\mathbb{E}\|\tilde{\mathbf{w}}_{i-1}\|^4. \end{aligned} \quad (301)$$

Now recall from (20) that

$$\mathbb{E}\|\tilde{\mathbf{x}}_i\|^4 \leq \rho^{i+1}\mathbb{E}\|\tilde{\mathbf{x}}_{-1}\|^4 + A_1\sigma_s^2(i+1)\rho^{i+1}\mu^2 + \frac{A_2\sigma_s^4\mu^2}{\nu^2}, \quad (302)$$

where $\rho = 1 - \mu\nu$ and A_1 and A_2 are some constants. On the other hand, recall from (44) that

$$\mathbb{E}\|\tilde{\mathbf{w}}_i\|^4 \leq \frac{B_1\gamma^2\rho_2^{i+1}}{1-\beta}\mu^2 + \frac{B_1\sigma_s^2(\delta^2 + \gamma^2)(i+1)\rho_2^{i+1}}{(1-\beta)^3}\mu^4 + \frac{B_1[(\gamma^2 + \nu^2)\sigma_s^4 + \sigma_{s,4}\nu^2]}{(1-\beta)^2\nu^2}\mu^4, \quad (303)$$

where $\rho_2 = 1 - \mu\nu/4$. Besides, we denote $\rho_1 = 1 - \mu\nu/2$ and clearly $\rho < \rho_1 < \rho_2$. Substituting (302) and (303) into (301) we reach

$$\begin{aligned} \mathbb{E}\|\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^4 & \leq \rho_1\mathbb{E}\|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^4 + a_7\rho_1^i\mu + a_8i^i\rho_1^i\mu^3 + a_9\mu^3 + a_{10}i^i\mu^3 + a_{11}i^i\rho_2^i\mu^5 + a_{12}\mu^5 \\ & \stackrel{(a)}{\leq} \rho_1\mathbb{E}\|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^4 + a_7\rho_1^i\mu + a_8i^i\rho_2^i\mu^3 + a_{10}i^i\mu^3 + a_{11}i^i\rho_2^i\mu^5 + 2a_9\mu^3, \end{aligned} \quad (304)$$

where the constants are defined as

$$\begin{aligned} a_7 & \triangleq 2a_1\mathbb{E}\|\tilde{\mathbf{x}}_{-1}\|^4, \quad a_8 \triangleq 2A_1a_1\sigma_s^2, \quad a_9 \triangleq \frac{2A_2a_1\sigma_s^4}{\nu^2}, \quad a_{10} \triangleq \frac{2B_1a_3\gamma^2}{1-\beta} \\ a_{11} & \triangleq \frac{2B_1a_3\sigma_s^2(\delta^2 + \gamma^2)}{(1-\beta)^3}, \quad a_{12} \triangleq \frac{2B_1a_3[(\gamma^2 + \nu^2)\sigma_s^4 + \sigma_{s,4}\nu^2]}{(1-\beta)^2\nu^2}. \end{aligned} \quad (305)$$

The inequality (a) holds because μ is chosen small enough such that $a_9\mu^3 > a_{12}\mu^5$. Now we continue iterating (304) and get

$$\begin{aligned} \mathbb{E}\|\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^4 & \leq \rho_1^{i+1}\mathbb{E}\|\tilde{\mathbf{w}}_{-1} - \tilde{\mathbf{x}}_{-1}\|^4 + a_7(i+1)\rho_1^i\mu + a_8i^i\mu^3 \sum_{k=0}^i (i-k) \left(\frac{\rho_1}{\rho_2}\right)^k \\ & \quad + a_{10}\rho_2^i\mu^3 \sum_{k=0}^i \left(\frac{\rho_1}{\rho_2}\right)^k + a_{11}\rho_2^i\mu^5 \sum_{k=0}^i (i-k) \left(\frac{\rho_1}{\rho_2}\right)^k + \frac{2a_9\mu^3}{1-\rho_2}. \end{aligned} \quad (306)$$

Note that $\rho_1 < \rho_2$, we then have

$$\sum_{k=0}^i \left(\frac{\rho_1}{\rho_2}\right)^k \leq \sum_{k=0}^{\infty} \left(\frac{\rho_1}{\rho_2}\right)^k \leq \frac{\rho_2}{\rho_2 - \rho_1} = \frac{4 - \mu\nu}{\mu\nu} \leq \frac{B_2}{\mu\nu}, \quad (307)$$

where B_2 is some constant. Meanwhile, it also holds that

$$\sum_{k=0}^i (i-k) \left(\frac{\rho_1}{\rho_2}\right)^k \leq i \sum_{k=0}^i \left(\frac{\rho_1}{\rho_2}\right)^k \leq \frac{iB_2}{\mu\nu}. \quad (308)$$

Substituting (307) and (308) into (306), we get

$$\begin{aligned} \mathbb{E}\|\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^4 & \leq \rho^{i+1}\mathbb{E}\|\tilde{\mathbf{w}}_{-1} - \tilde{\mathbf{x}}_{-1}\|^4 + a_7(i+1)\rho_1^i\mu + \frac{a_8B_2i^i\mu^2}{\nu} \\ & \quad + \frac{a_{10}B_2\rho_2^i\mu^2}{\nu} + \frac{a_{11}i^i\rho_2^i\mu^4}{\nu} + \frac{4a_9\mu^2}{\nu}. \end{aligned} \quad (309)$$

Recall from (263) that $\tilde{\mathbf{w}}_{-1} - \tilde{\mathbf{x}}_{-1} = \mu \nabla_w Q(\mathbf{w}_{-2}; \boldsymbol{\theta}_{-1})$. Then it holds that

$$\mathbb{E} \|\tilde{\mathbf{w}}_{-1} - \tilde{\mathbf{x}}_{-1}\|^4 = B_3 \mu^4, \quad (310)$$

where $B_3 = \mathbb{E} \|\nabla_w Q(\mathbf{w}_{-2}; \boldsymbol{\theta}_{-1})\|^4$. With this fact, expressions (309) becomes

$$\begin{aligned} & \mathbb{E} \|\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^4 \\ & \leq B_3 \rho_2^{i+1} \mu^4 + \alpha \tau (i+1) \rho_1^i \mu + \frac{\alpha_8 B_2 i \rho_2^i \mu^2}{\nu} + \frac{\alpha_{10} B_2 \rho_2^i \mu^2}{\nu} + \frac{\alpha_{11} i \rho_2^i \mu^4}{\nu} + \frac{4\alpha_9 \mu^2}{\nu} \\ & \leq B_3 \rho_2^{i+1} \mu^4 + \alpha \tau (i+1) \rho_2^i \mu + \frac{\alpha_8 B_2 i \rho_2^i \mu^2}{\nu} + \frac{\alpha_{10} B_2 \rho_2^i \mu^2}{\nu} + \frac{\alpha_{11} i \rho_2^i \mu^4}{\nu} + \frac{4\alpha_9 \mu^2}{\nu}. \end{aligned} \quad (311)$$

Furthermore, recall from (296) that

$$\mathbb{E} \|\tilde{\psi}_i - \tilde{\mathbf{x}}_i\|^4 \leq 8\mathbb{E} \|\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^4 + 8\mathbb{E} \|\tilde{\mathbf{w}}_i\|^4. \quad (312)$$

and recall the upper bound of $\mathbb{E} \|\tilde{\mathbf{w}}_i\|^4$ in (303). With the definition of all constants we finally reach

$$\mathbb{E} \|\tilde{\psi}_i - \tilde{\mathbf{x}}_i\|^4 = O\left(\frac{\delta^4 i \rho_2^i \mu}{\nu^3} + \frac{\delta^4 (\sigma_s^2 + \gamma^2) i \rho_2^i \mu^2}{(1-\beta)\nu^5} + \frac{\delta^4 (\delta^2 + \gamma^2) \sigma_s^2 i \rho_2^i \mu^4}{(1-\beta)^3 \nu} + \frac{\delta^4 \sigma_s^4}{\nu^6} \mu^2\right). \quad (313)$$

To further simplify the notation, we notice that when μ is sufficiently small it holds that

$$\begin{aligned} & \frac{\delta^4 i \rho_2^i \mu}{\nu^3} + \frac{\delta^4 (\sigma_s^2 + \gamma^2) i \rho_2^i \mu^2}{(1-\beta)\nu^5} + \frac{\delta^4 (\delta^2 + \gamma^2) \sigma_s^2 i \rho_2^i \mu^4}{(1-\beta)^3 \nu} \\ & = i \rho_2^i \mu \left(\frac{\delta^4}{\nu^3} + \frac{\delta^4 (\sigma_s^2 + \gamma^2) \mu}{(1-\beta)\nu^5} + \frac{\delta^4 (\delta^2 + \gamma^2) \sigma_s^2 \mu^3}{(1-\beta)^3 \nu} \right) \\ & \leq \frac{2\delta^4 i \rho_2^i \mu}{\nu^3} \leq \frac{B_4 \delta^4 (i+1) \rho_2^{i+1} \mu}{\nu^3} \end{aligned} \quad (314)$$

for some constant B_4 . As a result, we obtain

$$\mathbb{E} \|\tilde{\psi}_i - \tilde{\mathbf{x}}_i\|^4 = O\left(\frac{\delta^4 (i+1) \rho_2^{i+1} \mu}{\nu^3} + \frac{\delta^4 \sigma_s^4}{\nu^6} \mu^2\right). \quad (315)$$

Appendix J. Proof of Lemma 10

Under Assumption 6, we have

$$\begin{aligned} & \|\mathbf{H}_{i-1} - \mathbf{R}_{i-1}\| \\ & = \left\| \int_0^1 \nabla_w^2 J(w^0 - r\tilde{\psi}_{i-1}) dr - \int_0^1 \nabla_w^2 J(w^0 - r\tilde{\mathbf{x}}_{i-1}) dr \right\| \\ & \leq \int_0^1 \|\nabla_w^2 J(w^0 - r\tilde{\psi}_{i-1}) - \nabla_w^2 J(w^0 - r\tilde{\mathbf{x}}_{i-1})\| dr \\ & \leq \int_0^1 \kappa r \|\tilde{\psi}_{i-1} - \tilde{\mathbf{x}}_{i-1}\| dr = \frac{\kappa}{2} \|\tilde{\psi}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|. \end{aligned} \quad (316)$$

As a result, it holds that

$$\mathbb{E} \|\mathbf{R}_{i-1} - \mathbf{H}_{i-1}\|^4 \leq \bar{\kappa} \mathbb{E} \|\tilde{\psi}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^4, \quad (317)$$

where $\bar{\kappa} = \kappa^4/16$. Using (75), we reach the desired bounds shown in (77).

Appendix K. Proof of Theorem 11

For (72) in Assumption 5, if we take expectation over \mathcal{F}_{i-1} of both sides, it holds that

$$\mathbb{E} \|s_i(\psi_{i-1}) - s_i(\mathbf{x}_{i-1})\|^2 \leq \xi_1 \mathbb{E} \|\psi_{i-1} - \mathbf{x}_{i-1}\|^2. \quad (318)$$

Combining the above fact and inequalities (70)–(71), we get

$$\begin{aligned} & \mathbb{E} \|\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 \\ & \leq (1 - \mu\nu) \mathbb{E} \|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2 + \tau_1 \mu \mathbb{E} \|\tilde{\mathbf{w}}_{i-1}\|^2 \\ & \quad + \tau_2 \mu \sqrt{\mathbb{E} \|\mathbf{R}_{i-1} - \mathbf{H}_{i-1}\|^4 \mathbb{E} \|\tilde{\mathbf{x}}_{i-1}\|^4} + \xi_1 \mu^2 \mathbb{E} \|\tilde{\psi}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2, \end{aligned} \quad (319)$$

where the constants are defined as

$$\tau_1 \triangleq \frac{2\beta^2 \sigma^2}{\nu}, \quad \tau_2 \triangleq \frac{2}{\nu}. \quad (320)$$

Likewise, from (253) we have

$$\mathbb{E} \|\tilde{\psi}_i - \tilde{\mathbf{x}}_i\|^2 \leq 2\mathbb{E} \|\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 + 2\mathbb{E} \|\tilde{\mathbf{w}}_i\|^2. \quad (321)$$

Substituting the above inequality along with (77) into (319) gives

$$\begin{aligned} & \mathbb{E} \|\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 \\ & \leq (1 - \mu\nu + 2\xi_1 \mu^2) \mathbb{E} \|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2 + (\tau_1 \mu + 2\xi_1 \mu^2) \mathbb{E} \|\tilde{\mathbf{w}}_{i-1}\|^2 + \left[\sqrt{\tau_3} \rho_2^{i/2} \mu^{3/2} + r_4 \mu^2 \right] \sqrt{\mathbb{E} \|\tilde{\mathbf{x}}_{i-1}\|^4} \\ & \leq \left(1 - \frac{\mu\nu}{2}\right) \mathbb{E} \|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2 + 2\tau_1 \mu \mathbb{E} \|\tilde{\mathbf{w}}_{i-1}\|^2 + \left[r_3 \sqrt{\tau_3} \rho_2^{i/2} \mu^{3/2} + r_4 \mu^2 \right] \sqrt{\mathbb{E} \|\tilde{\mathbf{x}}_{i-1}\|^4} \\ & = \rho_1 \mathbb{E} \|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2 + 2\tau_1 \mu \mathbb{E} \|\tilde{\mathbf{w}}_{i-1}\|^2 + \left[r_3 \sqrt{\tau_3} \rho_2^{i/2} \mu^{3/2} + r_4 \mu^2 \right] \sqrt{\mathbb{E} \|\tilde{\mathbf{x}}_{i-1}\|^4} \end{aligned} \quad (322)$$

where the constants are defined as

$$r_3 \triangleq \frac{r_2 \delta^2}{\nu^{3/2}}, \quad r_4 \triangleq \frac{r_2 \delta^2 \sigma_s^2}{\nu^3}. \quad (323)$$

Recall the upper bound of $\mathbb{E} \|\tilde{\mathbf{w}}_i\|^2$ in (38) that

$$\mathbb{E} \|\tilde{\mathbf{w}}_i\|^2 \leq \frac{C_2 (\delta^2 + \gamma^2) \rho_1^i \mu^2}{(1-\beta)^2} + \frac{C_1 \sigma_s^2 \mu^2}{1-\beta} \quad (324)$$

where $\alpha = 1 - \epsilon/2 < \rho_1$, and C_1 and C_2 are some constants. Substituting (324) into (322), we have

$$\begin{aligned} & \mathbb{E} \|\tilde{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 \\ & \leq \rho_1 \mathbb{E} \|\tilde{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2 + (\tau_5 \rho_1^i \mu^3 + r_6 \mu^3) + \left[2r_3 \sqrt{\tau_3} \rho_2^{i/2} \mu^{3/2} + r_6 \mu^2 \right] \sqrt{\mathbb{E} \|\tilde{\mathbf{x}}_{i-1}\|^4}, \end{aligned} \quad (325)$$

where the constants are defined as

$$r_5 \triangleq \frac{2C_2 \tau_1 (\delta^2 + \gamma^2)}{(1-\beta)^2}, \quad r_6 \triangleq \frac{2C_1 \tau_1 \sigma_s^2}{1-\beta}. \quad (326)$$

Next, using (20) we have

$$\begin{aligned} \sqrt{\mathbb{E}\|\tilde{\mathbf{x}}_i\|^4} &\leq \sqrt{\rho^{i+1}\mathbb{E}\|\tilde{\mathbf{x}}_{i-1}\|^4 + A_3\sigma_s^2(i+1)\rho^{i+1}\mu^2 + \frac{A_2\sigma_s^4\mu^2}{\nu^2}} \\ &\leq C_3\rho^{(i+1)/2} + C_4\sigma_s\sqrt{i+1}\rho^{(i+1)/2}\mu + C_5\frac{\sigma_s^2\mu}{\nu} \\ &\leq C_3\rho_2^{(i+1)/2} + C_4\sigma_s\sqrt{i+1}\rho_2^{(i+1)/2}\mu + C_5\frac{\sigma_s^2\mu}{\nu}. \end{aligned} \quad (327)$$

Substituting (327) into (325), we reach

$$\begin{aligned} \mathbb{E}\|\hat{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 &\leq \rho_1\mathbb{E}\|\hat{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2 + (r_5\rho_1^i\mu^3 + r_6\mu^3) + r_7\sqrt{i}\rho_2^i\mu^{3/2} + r_8i\rho_2^i\mu^{5/2} + r_9\sqrt{i}\rho_2^i\mu^{5/2} \\ &\quad + r_{10}\mu^2\rho_2^{i/2} + r_{11}\sqrt{i}\rho_2^{i/2}\mu^3 + r_{12}\mu^3, \end{aligned} \quad (328)$$

where the constants are defined as

$$\begin{aligned} r_7 &\triangleq 2C_3r_3, \quad r_8 \triangleq 2C_4r_3\sigma_s, \quad r_9 \triangleq \frac{2C_5r_3\sigma_s^2}{\nu} \\ r_{10} &\triangleq C_3r_6, \quad r_{11} \triangleq C_4r_6\sigma_s, \quad r_{12} \triangleq \frac{C_5r_6\sigma_s^2}{\nu}. \end{aligned} \quad (329)$$

Now we denote

$$\tau_1 \triangleq \rho_1^{1/2}, \quad \tau_2 \triangleq \rho_2^{1/2}. \quad (330)$$

Clearly, we have

$$\rho_1 < \tau_1, \quad \rho_2 < \tau_2, \quad \tau_1 < \tau_2. \quad (331)$$

With the above relation, expressions (328) becomes

$$\begin{aligned} \mathbb{E}\|\hat{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 &\leq \tau_2\mathbb{E}\|\hat{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2 + (r_5\tau_1^i\mu^3 + r_6\mu^3) + r_7\sqrt{i}\tau_2^i\mu^{3/2} + r_8i\tau_2^i\mu^{5/2} + r_9\sqrt{i}\tau_2^i\mu^{5/2} \\ &\quad + r_{10}\mu^2\tau_2^{i/2} + r_{11}\sqrt{i}\tau_2^{i/2}\mu^3 + r_{12}\mu^3 \\ &\leq \tau_2\mathbb{E}\|\hat{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2 + 2r_{10}\tau_2^i\mu^2 + 2r_7\sqrt{i}\tau_2^i\mu^{3/2} + r_8i\tau_2^i\mu^{5/2} + (r_6 + r_{12})\mu^3 \\ &\leq \tau_2^{i+1}\mathbb{E}\|\hat{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2 + 2r_{10}(i+1)\tau_2^i\mu^2 + 2r_7\tau_2^i\mu^{3/2} \left(\sum_{k=0}^i \sqrt{i-k} \right) \\ &\quad + r_8\tau_2^i\mu^{5/2}(i+1) + \frac{(r_6 + r_{12})\mu^3}{1 - \tau_2}. \end{aligned} \quad (332)$$

Note that $\tau_2 = \sqrt{\rho_2} = \sqrt{1 - \mu\nu/4}$. When μ is sufficiently small, we have $\tau_2 = 1 - \mu\nu/8$ and hence $1 - \tau_2 = \mu\nu/2$. With this fact and recall that $\mathbb{E}\|\hat{\mathbf{w}}_{i-1} - \tilde{\mathbf{x}}_{i-1}\|^2 = C_6\mu^2$, finally we can show that

$$\mathbb{E}\|\hat{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 \leq C_6\tau_2^{i+1}\mu^2 + 2r_{10}(i+1)\tau_2^i\mu^2 + 2r_7\tau_2^i\mu^{3/2} \left(\sum_{k=0}^i \sqrt{i-k} \right)$$

$$+ r_8\tau_2^i\mu^{5/2}(i+1) + \frac{8(r_6 + r_{12})\mu^2}{\nu}. \quad (333)$$

Substituting the definitions of all constants, we get

$$\begin{aligned} \mathbb{E}\|\hat{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 &\leq C_7 \left(\frac{\delta^2 s_1(i)\tau_2^i\mu^{3/2}}{\nu^{5/2}} + \tau_2^{i+1}\mu^2 + \frac{\sigma_s^2\delta^2(i+1)\tau_2^i\mu^{5/2}}{(1-\beta)\nu} + \frac{\sigma_s\delta^2s_2(i)\tau_2^i\mu^{5/2}}{\nu^{5/2}} + \frac{\delta^2\sigma_s^4\mu^2}{(1-\beta)\nu^2} \right) \\ &\leq C_8 \left(\frac{\delta^2\sigma_s^2s_2(i)\tau_2^{i+1}\mu^{3/2}}{(1-\beta)\nu^{5/2}} + \frac{\delta^2\sigma_s^4\mu^2}{(1-\beta)\nu^2} \right). \end{aligned} \quad (334)$$

where

$$s_1(i) \triangleq \sum_{k=0}^i \sqrt{i-k}, \quad s_2(i) \triangleq i(i+1) \quad (335)$$

Furthermore, it holds that

$$\mathbb{E}\|\hat{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 \leq 2\mathbb{E}\|\hat{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 + 2\beta^2\mathbb{E}\|\hat{\mathbf{w}}_i\|^2. \quad (336)$$

Using the upper bound for $\mathbb{E}\|\hat{\mathbf{w}}_i\|^2$ in (324), we then have

$$\mathbb{E}\|\hat{\mathbf{w}}_i - \tilde{\mathbf{x}}_i\|^2 = O\left(\frac{\delta^2\sigma_s^2s_2(i)\tau_2^{i+1}\mu^{3/2}}{(1-\beta)\nu^{5/2}} + \frac{(\delta^2 + \gamma^2)\rho_1^{i+1}\mu^2}{(1-\beta)^2} + \frac{\delta^2\sigma_s^4\mu^2}{(1-\beta)\nu^2} \right). \quad (337)$$

References

- N. O. Attoh-Okin. Analysis of learning rate and momentum term in backpropagation neural network algorithm trained to predict pavement performance. *Advances in Engineering Software*, 30(4):291–302, 1999.
- A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.
- M. Bellanger. *Adaptive Digital Filters and Signal Analysis*. 2nd Edition, Marcel Dekker, 2001.
- D. P. Bertsekas. *Nonlinear programming*. Athena Scientific, 1999.
- L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Proc. International Conference on Computational Statistics*, pages 177–186. Springer, Paris, France, 2010.
- O. Bousquet and L. Bottou. The tradeoffs of large scale learning. In *Proc. Advances in Neural Information Processing Systems*, pages 161–168, Vancouver, Canada, 2008.
- S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

- Y. Cevher, S. Becker, and M. Schmidt. Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics. *IEEE Signal Processing Magazine*, 31(5):32–43, 2014.
- A. d’Aspremont. Smooth optimization with approximate gradient. *SIAM Journal on Optimization*, 19(3):1171–1183, 2008.
- A. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Proc. Advances in Neural Information Processing Systems*, pages 1646–1654, Montreal, Canada, 2014.
- O. Devolder, F. Glineur, and Y. Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1-2):37–75, 2014.
- A. Diehlevent, N. Flammarion, and F. Bach. Harder, better, faster, stronger convergence rates for least-squares regression. *arXiv: 1602.05419*, Feb. 2016.
- J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(2):2121–2159, 2011.
- N. Flammarion and F. Bach. From averaging to acceleration, there is only a step-size. *Journal of Machine Learning Research*, 40(1):1–38, 2015.
- R. Gennulla, E. Nijkamp, P. J. Haas, and Y. Sismanis. Large-scale matrix factorization with distributed stochastic gradient descent. In *Proc. International Conference on Knowledge Discovery and Data Mining*, pages 69–77, Alberta, Canada, 2011.
- S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization I: A generic algorithmic framework. *SIAM Journal on Optimization*, 22(4):1469–1492, 2012.
- S. Haykin. *Adaptive Filter Theory*. Fourth Edition, Prentice-Hall, NJ, 2008.
- C. Hu, W. Pan, and J. T. Kwok. Accelerated gradient methods for stochastic optimization and online learning. In *Proc. Advances in Neural Information Processing Systems*, pages 781–789, Vancouver, Canada, 2009.
- R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 315–323, Lake Tahoe, Nevada, 2013.
- S. Kahou, C. Pal, X. Bouthillier, P. Froumenty, and et al. Combining modality specific deep neural networks for emotion recognition in video. In *Proc. International Conference on Multimodal Interaction*, pages 543–550, Sydney, Australia, 2013.
- A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proc. Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.
- G. Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1-2):365–397, 2012.
- L. Lessard, B. Recht, and A. Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 26(1):57–95, 2016.
- A. Nedic and D. P. Bertsekas. Convergence rate of incremental subgradient algorithms. In S. Uryasev and M. Pardalos P, editors, *Stochastic Optimization: Algorithms and Applications*, volume 54, pages 223–264. Springer, 2001.
- Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.
- Y. Nesterov. *Introductory Lectures on Convex Optimization*. Springer, 2004.
- Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.
- A. Ntanda. Stochastic proximal gradient descent with acceleration techniques. In *Proc. Advances in Neural Information Processing Systems*, pages 1574–1582, Montreal, Canada, 2014.
- B. T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.
- B. T. Polyak. *Introduction to Optimization*. Optimization Software, NY, 1987.
- J. G. Proakis. Channel identification for high speed digital communications. *IEEE Transactions on Automatic Control*, 19(6):916–922, 1974.
- N. Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1):145–151, 1999.
- N. L. Roux, M. Schmidt, and F. R. Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, pages 2663–2671, Lake Tahoe, Nevada, 2012.
- S. Roy and J. J. Shynk. Analysis of the momentum LMS algorithm. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(12):2088–2098, 1990.
- A. H. Sayed. *Adaptive Filters*. Wiley, NY, 2008.
- A. H. Sayed. Adaptation, learning, and optimization over networks. *Foundations and Trends in Machine Learning*, 7(4-5):311–801, Jul. 2014a.
- A. H. Sayed. Adaptive networks. *Proceedings of the IEEE*, 102(4):460–497, 2014b.
- S. Shalev-Shwartz. SDCA without duality. *arXiv:1502.06177*, Feb. 2015.
- S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In *Proc. International Conference on Machine Learning*, pages 64–72, Beijing, China, 2014.
- R. Sharma, W. A. Sethares, and J. A. Bucklew. Analysis of momentum adaptive filtering algorithms. *IEEE Transactions on Signal Processing*, 46(5):1430–1434, 1998.

- J. J. Shynk and S. Roy. The LMS algorithm with momentum updating. In *Proc. IEEE International Symposium on Circuits and Systems*, pages 2651–2654, Espoo, Finland, June 1988.
- I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *Proc. International Conference on Machine Learning*, pages 1139–1147, Atlanta, USA, 2013.
- C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelo, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, Boston, USA, June 2015.
- S. Theodoridis. *Machine Learning: A Bayesian and Optimization Perspective*. Academic Press, NY, 2015.
- L. K. Ting, C. F. N. Cowan, and R. F. Woods. Tracking performance of momentum LMS algorithm for a chirped sinusoidal signal. In *Proc. European Signal Processing Conference*, pages 1–4, Tampere, Finland, 2000.
- M. A. Tugay and Y. Tanik. Properties of the momentum LMS algorithm. *Signal Processing*, 18(2):117–127, 1989.
- M. Tytgert. Poor starting points in machine learning. *arXiv:1602.02823*, Feb. 2016.
- B. Widrow and S. D. Stearns. *Adaptive Signal Processing*. Prentice-Hall, NJ, 1985.
- W. Wiegand, A. Komoda, and T. Heskes. Stochastic dynamics of learning with momentum in neural networks. *Journal of Physics A: Mathematical and General*, 27(13):4425–4438, 1994.
- L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11(Oct):2543–2596, 2010.
- B. Ying and A. H. Sayed. Performance limits of online stochastic sub-gradient learning. *arXiv:1511.07902*, Oct. 2015.
- B. Ying and A. H. Sayed. Performance limits of single-agent and multi-agent sub-gradient stochastic learning. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, pages 4905–4909, Shanghai, China, March 2016.
- K. Yuan, B. Ying, and A. H. Sayed. On the influence of momentum acceleration on online learning. In *Proc. International Conference on Acoustics, Speech and Signal Processing*, pages 4915–4919, Shanghai, China, March 2016.
- S. Zareba, A. Gonczarek, J. M. Tomczak, and J. Świątek. Accelerated learning for restricted Boltzmann machine with momentum term. In *Proc. International Conference on Systems Engineering*, pages 187–192, Coventry, UK, 2015.
- T. Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proc. International Conference on Machine Learning*, page 116, Alberta, Canada, 2004.
- X. Zhang and Y. LeCun. Text understanding from scratch. *arXiv:1502.01710*, Feb. 2015.
- W. Zhong and J. T. Kwok. Accelerated stochastic gradient method for composite regularization. In *Proc. International Conference on Artificial Intelligence and Statistics*, pages 1086–1094, Reykjavik, Iceland, 2014.
- Z. Zhu. Katyusha: Accelerated variance reduction for faster SGD. *arXiv:1603.05953*, Mar. 2016.

Data-driven Rank Breaking for Efficient Rank Aggregation

Ashish Khetan
Sewoong Oh

*Department of Industrial and Enterprise Systems Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA*

KHETAN2@ILLINOIS.EDU
SWOH@ILLINOIS.EDU

Editor: Benjamin Recht

Abstract

Rank aggregation systems collect ordinal preferences from individuals to produce a global ranking that represents the social preference. Rank-breaking is a common practice to reduce the computational complexity of learning the global ranking. The individual preferences are broken into pairwise comparisons and applied to efficient algorithms tailored for independent paired comparisons. However, due to the ignored dependencies in the data, naive rank-breaking approaches can result in inconsistent estimates. The key idea to produce accurate and consistent estimates is to treat the pairwise comparisons unequally, depending on the topology of the collected data. In this paper, we provide the optimal rank-breaking estimator, which not only achieves consistency but also achieves the best error bound. This allows us to characterize the fundamental tradeoff between accuracy and complexity. Further, the analysis identifies how the accuracy depends on the spectral gap of a corresponding comparison graph.

Keywords: Rank aggregation, Plackett-Luce model, Sample complexity

1. Introduction

In several applications such as electing officials, choosing policies, or making recommendations, we are given partial preferences from individuals over a set of alternatives, with the goal of producing a global ranking that represents the collective preference of the population or the society. This process is referred to as *rank aggregation*. One popular approach is *learning to rank*. Economists have modeled each individual as a rational being maximizing his/her perceived utility. Parametric probabilistic models, known collectively as Random Utility Models (RUMs), have been proposed to model such individual choices and preferences (McFadden, 1980). This allows one to infer the global ranking by learning the inherent utility from individuals' revealed preferences, which are noisy manifestations of the underlying true utility of the alternatives.

Traditionally, learning to rank has been studied under the following data collection scenarios: pairwise comparisons, best-out-of- k comparisons, and k -way comparisons. *Pairwise comparisons* are commonly studied in the classical context of sports matches as well as more recent applications in crowdsourcing, where each worker is presented with a pair of choices and asked to choose the more favorable one. *Best-out-of- k comparisons* data sets are commonly available from purchase history of customers. Typically, a set of k alternatives are offered among which one is chosen or purchased by each customer. This has been widely studied in operations research in the context of modeling customer choices for revenue management and assortment optimization. The *k-way comparisons* are assumed in traditional rank aggregation scenarios, where each person reveals his/her preference as

a ranked list over a set of k items. In some real-world elections, voters provide ranked preferences over the whole set of candidates (Lundell, 2007). We refer to these three types of ordinal data collection scenarios as 'traditional' throughout this paper.

For such traditional data sets, there are several computationally efficient inference algorithms for finding the Maximum Likelihood (ML) estimates that provably achieve the minimax optimal performance (Negahban et al., 2012; Shah et al., 2015a; Hajek et al., 2014). However, modern data sets can be unstructured. Individual's revealed ordinal preferences can be implicit, such as movie ratings, time spent on the news articles, and whether the user finished watching the movie or not. In crowdsourcing, it has also been observed that humans are more efficient at performing batch comparisons (Gomes et al., 2011), as opposed to providing the full ranking or choosing the top item. This calls for more flexible approaches for rank aggregation that can take such diverse forms of ordinal data into account. For such non-traditional data sets, finding the ML estimate can become significantly more challenging, requiring run-time exponential in the problem parameters.

To avoid such a computational bottleneck, a common heuristic is to resort to *rank-breaking*. The collected ordinal data is first transformed into a bag of pairwise comparisons, ignoring the dependencies that were present in the original data. This is then processed via existing inference algorithms tailored for *independent* pairwise comparisons, hoping that the dependency present in the input data does not lead to inconsistency in estimation. This idea is one of the main motivations for numerous approaches specializing in learning to rank from pairwise comparisons, e.g., (Ford Jr., 1957; Negahban et al., 2014; Azari Soufiani et al., 2013). However, such a heuristic of full rank-breaking defined explicitly in (1), where all pairwise comparisons are weighted and treated equally ignoring their dependencies, has been recently shown to introduce inconsistency (Azari Soufiani et al., 2014).

The key idea to produce accurate and consistent estimates is to treat the pairwise comparisons unequally, depending on the topology of the collected data. A fundamental question of interest to practitioners is how to choose the weight of each pairwise comparison in order to achieve not only consistency but also the best accuracy, among those consistent estimators using rank-breaking. We study how the accuracy of resulting estimate depends on the topology of the data and the weights on the pairwise comparisons. This provides a guideline for the optimal choice of the weights, driven by the topology of the data, that leads to accurate estimates.

Problem formulation. The premise in the current race to collect more data on user activities is that, a hidden true preference manifests in the user's activities and choices. Such data can be explicit, as in ratings, ranked lists, pairwise comparisons, and like/dislike buttons. Others are more implicit, such as purchase history and viewing times. While more data in general allows for a more accurate inference, the heterogeneity of user activities makes it difficult to infer the underlying preferences directly. Further, each user reveals her preference on only a few contents.

Traditional collaborative filtering fails to capture the diversity of modern data sets. The sparsity and heterogeneity of the data renders typical similarity measures ineffective in the nearest-neighbor methods. Consequently, simple measures of similarity prevail in practice, as in Amazon's "people who bought ... also bought ..." scheme. Score-based methods require translating heterogeneous data into numeric scores, which is a priori a difficult task. Even if explicit ratings are observed, those are often unreliable and the scale of such ratings vary from user to user.

We propose aggregating ordinal data based on users' revealed preferences that are expressed in the form of *partial orderings* (notice that our use of the term is slightly different from its original

use in revealed preference theory). We interpret user activities as manifestation of the hidden preferences according to discrete choice models (in particular the Plackett-Luce model defined in (1)). This provides a more reliable, scale-free, and widely applicable representation of the heterogeneous data as partial orderings, as well as a probabilistic interpretation of how preferences manifest. In full generality, the data collected from each individual can be represented by a *partially ordered set (poset)*. Assuming consistency in a user's revealed preferences, any ordered relations can be seamlessly translated into a poset, represented as a Hasse diagram by a directed acyclic graph (DAG). The DAG below represents ordered relations $a > \{b, d\}$, $b > c$, $\{c, d\} > e$, and $e > f$. For example, this could have been translated from two sources: a five star rating on a and a three star ratings on b, c, d , a two star rating on e , and a one star rating on f ; and the item b being purchased after reviewing c as well.

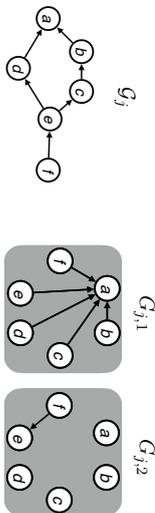


Figure 1: A DAG representation of consistent partial ordering of a user j , also called a Hasse diagram (left). A set of rank-breaking graphs extracted from the Hasse diagram for the separator item a and e , respectively (right).

There are n users or agents, and each agent j provides his/her ordinal evaluation on a subset S_j of d items or alternatives. We refer to $S_j \subset \{1, 2, \dots, d\}$ as *offerings* provided to j , and use $k_j = |S_j|$ to denote the size of the offerings. We assume that the partial ordering over the offerings is a manifestation of her preferences as per a popular choice model known as Plackett-Luce (PL) model. As we explain in detail below, the PL model produces total orderings (rather than partial ones). The data collector queries each user for a partial ranking in the form of a poset over S_j . For example, the data collector can ask for the top item, unordered subset of three next preferred items, the fifth item, and the least preferred item. In this case, an example of such poset could be $a < \{b, c, d\} < e < f$, which could have been generated from a total ordering produced by the PL model and taking the corresponding partial ordering from the total ordering. Notice that we fix the topology of the DAG first and ask the user to fill in the node identities corresponding to her total ordering as (randomly) generated by the PL model. Hence, the structure of the poset is considered deterministic, and only the identity of the nodes in the poset is considered random. Alternatively, one could consider a different scenario where the topology of the poset is also random and depends on the outcome of the preference, which is out-side the scope of this paper and provides an interesting future research direction.

The PL model is a special case of *random utility models*, defined as follows (Walker and Ben-Akiva, 2002; Azari Soufiani et al., 2012). Each item i has a real-valued latent utility θ_i . When presented with a set of items, a user's revealed preference is a partial ordering according to noisy manifestation of the utilities, i.e. i.i.d. noise added to the true utility θ_i 's. The PL model is a special case where the noise follows the standard Gumbel distribution, and is one of the most popular model in social choice theory (McFadden, 1973; McFadden and Train, 2000). PL has several important

properties, making this model realistic in various domains, including marketing (Guadagni and Little, 1983), transportation (McFadden, 1980; Ben-Akiva and Lerman, 1985), biology (Shan and Curtis, 1995), and natural language processing (Mikolov et al., 2013). Precisely, each user j , when presented with a set S_j of items, draws a noisy utility of each item i according to

$$u_i = \theta_i + Z_i,$$

where Z_i 's follow the independent standard Gumbel distribution. Then we observe the ranking resulting from sorting the items as per noisy observed utilities u_j 's. Alternatively, the PL model is also equivalent to the following random process. For a set of alternatives S_j , a ranking $\sigma_j : [|S_j|] \rightarrow S_j$ is generated in two steps: (1) independently assign each item $i \in S_j$ an unobserved value X_i , exponentially distributed with mean $e^{-\theta_i}$; (2) select a ranking σ_j so that $X_{\sigma_j(1)} \leq X_{\sigma_j(2)} \leq \dots \leq X_{\sigma_j(|S_j|)}$.

The PL model (i) satisfies Luce's 'independence of irrelevant alternatives' in social choice theory (Ray, 1973), and has a simple characterization as sequential (random) choices as explained below; and (ii) has a maximum likelihood estimator (MLE) which is a convex program in θ in the traditional scenarios of pairwise, best-out-of- k and k -way comparisons. Let $\mathbb{P}(a > \{b, c, d\})$ denote the probability a was chosen as the best alternative among the set $\{a, b, c, d\}$. Then, the probability that a user reveals a linear order ($a > b > c > d$) is equivalent as making sequential choice from the top to bottom:

$$\begin{aligned} \mathbb{P}(a > b > c > d) &= \mathbb{P}(a > \{b, c, d\}) \mathbb{P}(b > \{c, d\}) \mathbb{P}(c > d) \\ &= \frac{e^{\theta_a}}{(e^{\theta_a} + e^{\theta_b} + e^{\theta_c} + e^{\theta_d})} \frac{e^{\theta_b}}{(e^{\theta_b} + e^{\theta_c} + e^{\theta_d})} \frac{e^{\theta_c}}{(e^{\theta_c} + e^{\theta_d})} \frac{e^{\theta_d}}{(e^{\theta_c} + e^{\theta_d})}. \end{aligned}$$

We use the notation ($a > b$) to denote the event that a is preferred over b . In general, for user j presented with offerings S_j , the probability that the revealed preference is a total ordering σ_j is $\mathbb{P}(\sigma_j) = \prod_{i \in \{1, \dots, k_j - 1\}} (e^{\theta_{\sigma_j^{-1}(i)}}) / (\sum_{i' \in S_j} e^{\theta_{\sigma_j^{-1}(i')}})$. We consider the true utility $\theta^* \in \Omega_b$, where we define Ω_b as

$$\Omega_b \equiv \left\{ \theta \in \mathbb{R}^d \mid \sum_{i \in [d]} \theta_i = 0, |\theta_i| \leq b \text{ for all } i \in [d] \right\}.$$

Note that by definition, the PL model is invariant under shifting the utility θ_i 's. Hence, the centering ensures uniqueness of the parameters for each PL model. The bound b on the dynamic range is not a restriction, but is written explicitly to capture the dependence of the accuracy in our main results.

We have n users each providing a partial ordering of a set of offerings S_j according to the PL model. Let G_j denote both the DAG representing the partial ordering from user j 's preferences. With a slight abuse of notations, we also let G_j denote the set of rankings that are consistent with this DAG. For general partial orderings, the probability of observing G_j is the sum of all total orderings that is consistent with the observation, i.e. $\mathbb{P}(G_j) = \sum_{\sigma \in G_j} \mathbb{P}(\sigma)$. The goal is to efficiently learn the true utility $\theta^* \in \Omega_b$ from the n sampled partial orderings. One popular approach is to compute the maximum likelihood estimate (MLE) by solving the following optimization:

$$\underset{\theta \in \Omega_b}{\text{maximize}} \quad \sum_{j=1}^n \log \mathbb{P}(G_j).$$

This optimization is a simple convex optimization, in particular a logit regression, when the structure of the data $\{\mathcal{G}_j\}_{j \in [n]}$ is traditional. This is one of the reasons the PL model is attractive. However, for general posets, this can be computationally challenging. Consider an example of position- p ranking, where each user provides which item is at p -th position in his/her ranking. Each term in the log-likelihood for this data involves summation over $O((p-1)!) rankings, which takes $O(n(p-1)!) operations to evaluate the objective function. Since p can be as large as d , such a computational blow-up renders MLE approach impractical. A common remedy is to resort to rank-breaking, which might result in inconsistent estimates.$$

Rank-breaking. Rank-breaking refers to the idea of extracting a set of pairwise comparisons from the observed partial orderings and applying estimators tailored for paired comparisons treating each piece of comparisons as independent. Both the choice of which paired comparisons to extract and the choice of parameters in the estimator, which we call *weights*, turns out to be crucial as we will show. Inappropriate selection of the paired comparisons can lead to inconsistent estimators as proved in Azari Soufiani et al. (2014), and the standard choice of the parameters can lead to a significantly suboptimal performance.

A naive rank-breaking that is widely used in practice is to apply rank-breaking to all possible pairwise relations that one can read from the partial ordering and weighing them equally. We refer to this practice as *full rank-breaking*. In the example in Figure 1, full rank-breaking first extracts the bag of comparisons $\mathcal{C} = \{(a > b), (a > c), (a > d), (a > e), (a > f), \dots, (e > f)\}$ with 13 paired comparison outcomes, and apply the maximum likelihood estimator treating each paired outcome as independent. Precisely, the *full rank-breaking estimator* solves the convex optimization of

$$\hat{\theta} \in \arg \max_{\theta \in \Omega_{\theta}} \sum_{(i, i') \in \mathcal{C}} (\theta_i - \log(e^{\theta_i} + e^{\theta_{i'}})) \quad (1)$$

There are several efficient implementation tailored for this problem (Ford Jr., 1957; Hunter, 2004; Negahban et al., 2012; Maystre and Grossglauser, 2015a), and under the traditional scenarios, these approaches provably achieve the minimax optimal rate (Hajek et al., 2014; Shah et al., 2015a). For general non-traditional data sets, there is a significant gain in computational complexity. In the case of position- p ranking, where each of the n users report his/her p -th ranking item among κ items, the computational complexity reduces from $O(n(p-1)!) for the MLE in (1) to $O(np(\kappa-p)) for the full rank-breaking estimator in (1). However, this gain comes at the cost of accuracy. It is known that the full-rank breaking estimator is inconsistent (Azari Soufiani et al., 2014); the error is strictly bounded away from zero even with infinite samples.$$

Perhaps surprisingly, Azari Soufiani et al. (2014) recently characterized the entire set of consistent rank-breaking estimators. Instead of using the bag of paired comparisons, the sufficient information for consistent rank-breaking is a set of rank-breaking graphs defined as follows.

Recall that a user j provides his/her preference as a poset represented by a DAG \mathcal{G}_j . Consistent rank-breaking first identifies all *separators* in the DAG. A node in the DAG is a separator if one can partition the rest of the nodes into two parts. A partition A_{top} which is the set of items that are preferred over the separator item, and a partition A_{bottom} which is the set of items that are less preferred than the separator item. One caveat is that we allow A_{top} to be empty, but A_{bottom} must have at least one item. In the example in Figure 1, there are two separators: the item a and the item e . Using these separators, one can extract the following partial ordering from the original poset: $(a > \{b, c, d\} > e > f)$. The items a and e separate the set of offerings into partitions, hence

the name separator. We use ℓ_j to denote the number of separators in the poset \mathcal{G}_j from user j . Let $p_{j,a}$ denote the ranked position of the a -th separator in the poset \mathcal{G}_j , and we sort the positions such that $p_{j,1} < p_{j,2} < \dots < p_{j,\ell_j}$. The set of separators is denoted by $\mathcal{P}_j = \{p_{j,1}, p_{j,2}, \dots, p_{j,\ell_j}\}$. For example, since the separator a is ranked at position 1 and e is at the 5-th position, $\ell_j = 2$, $p_{j,1} = 1$, and $p_{j,2} = 5$. Note that f is not a separator (whereas a is) since corresponding A_{bottom} is empty.

Conveniently, we represent this extracted partial ordering using a set of DAGs, which are called *rank-breaking graphs*. We generate one rank-breaking graph per separator. A rank breaking graph $G_{j,a} = (S_j, E_{j,a})$ for user j and the a -th separator is defined as a directed graph over the set of offerings S_j , where we add an edge from a node that is less preferred than the a -th separator to the separator, i.e. $E_{j,a} = \{(i, i') \mid i' \text{ is the } a\text{-th separator, and } \sigma_j^{-1}(i) > p_{j,a}\}$. Note that by the definition of the separator, $E_{j,a}$ is a non-empty set. An example of rank-breaking graphs are shown in Figure 1.

This rank-breaking graphs were introduced in Azari Soufiani et al. (2013), where it was shown that the pairwise ordinal relations that is represented by edges in the rank-breaking graphs are sufficient information for using any estimation based on the idea of rank-breaking. Precisely, on the converse side, it was proved in Azari Soufiani et al. (2014) that any pairwise outcomes that is not present in the rank-breaking graphs $G_{j,a}$'s lead to inconsistency for a general θ^* . On the achievability side, it was proved that all pairwise outcomes that are present in the rank-breaking graphs give a consistent estimator, as long as all the paired comparisons in each $G_{j,a}$ are weighted equally.

It should be noted that rank-breaking graphs are defined slightly differently in Azari Soufiani et al. (2013). Specifically, Azari Soufiani et al. (2013) introduced a different notion of rank-breaking graph, where the vertices represent positions in total ordering. An edge between two vertices i_1 and i_2 denotes that the pairwise comparison between items ranked at position i_1 and i_2 is included in the estimator. Given such observation from the PL model, Azari Soufiani et al. (2013) and Azari Soufiani et al. (2014) prove that a rank-breaking graph is consistent if and only if it satisfies the following property. If a vertex i_1 is connected to any vertex i_2 , where $i_2 > i_1$, then i_1 must be connected to all the vertices i_3 such that $i_3 > i_1$. Although the specific definitions of rank-breaking graphs are different from our setting, the mathematical analysis of Azari Soufiani et al. (2013) still holds when interpreted appropriately. Specifically, we consider only those rank-breaking that are consistent under the conditions given in Azari Soufiani et al. (2013). In our rank-breaking graph $G_{j,a}$, a separator node is connected to all the other item nodes that are ranked below it (numerically higher positions).

In the algorithm described in (33), we satisfy this sufficient condition for consistency by restricting to a class of convex optimizations that use the same weight $\lambda_{j,a}$ for all $(\kappa - p_{j,a})$ paired comparisons in the objective function, as opposed to allowing more general weights that defer from a pair to another pair in a rank-breaking graph $G_{j,a}$.

Algorithm. Consistent rank-breaking first identifies separators in the collected posets $\{\mathcal{G}_j\}_{j \in [n]}$ and transform them into rank-breaking graphs $\{G_{j,a}\}_{j \in [n], a \in [\ell_j]}$ as explained above. These rank-breaking graphs are input to the MLE for paired comparisons, assuming all directed edges in the rank-breaking graphs are independent outcome of pairwise comparisons. Precisely, the *consistent*

rank-breaking estimator solves the convex optimization of maximizing the paired log likelihoods

$$\mathcal{L}_{\text{RB}}(\theta) = \sum_{j=1}^n \sum_{a=1}^{\ell_j} \lambda_{j,a} \left\{ \sum_{(i,i') \in E_{j,a}} \left(\theta_{i'} - \log \left(e^{\theta_i} + e^{\theta_{i'}} \right) \right) \right\}, \quad (2)$$

where $E_{j,a}$'s are defined as above via separators and different choices of the non-negative weights $\lambda_{j,a}$'s are possible and the performance depends on such choices. Each weight $\lambda_{j,a}$ determine how much we want to weigh the contribution of a corresponding rank-breaking graph $G_{j,a}$. We define the consistent rank-breaking estimate $\hat{\theta}$ as the optimal solution of the convex program:

$$\hat{\theta} \in \arg \max_{\theta \in \Omega_b} \mathcal{L}_{\text{RB}}(\theta). \quad (3)$$

By changing how we weigh each rank-breaking graph (by choosing the $\lambda_{j,a}$'s), the convex program (3) spans the entire set of consistent rank-breaking estimators, as characterized in Azari Soufiani et al. (2014). However, only asymptotic consistency was known, which holds independent of the choice of the weights $\lambda_{j,a}$'s. Naturally, a uniform choice of $\lambda_{j,a} = \lambda$ was proposed in (Azari Soufiani et al., 2014).

Note that this can be efficiently solved, since this is a simple convex optimization, in particular a logit regression, with only $O(\sum_{j=1}^n \ell_j \kappa_j)$ terms. For a special case of position- p breaking, the $O(n(p-1))$ complexity of evaluating the objective function for the MLE is now significantly reduced to $O(n(k-p))$ by rank-breaking. Given this potential exponential gain in efficiency, a natural question of interest is "what is the price we pay in the accuracy?". We provide a sharp analysis of the performance of rank-breaking estimators in the finite sample regime, that quantifies the price of rank-breaking. Similarly, for a practitioner, a core problem of interest is how to choose the weights in the optimization in order to achieve the best accuracy. Our analysis provides a data-driven guideline for choosing the optimal weights.

Contributions. In this paper, we provide an upper bound on the error achieved by the rank-breaking estimator of (3) for any choice of the weights in Theorem 8. This explicitly shows how the error depends on the choice of the weights, and provides a guideline for choosing the optimal weights $\lambda_{j,a}$'s in a data-driven manner. We provide the explicit formula for the optimal choice of the weights and provide the error bound in Theorem 2. The analysis shows the explicit dependence of the error in the problem dimension d and the number of users n that matches the numerical experiments.

If we are designing surveys and can choose which subset of items to offer to each user and also can decide which type of ordinal data we can collect, then we want to design such surveys in a way to maximize the accuracy for a given number of questions asked. Our analysis provides how the accuracy depends on the topology of the collected data, and provides a guidance when we do have some control over which questions to ask and which data to collect. One should maximize the spectral gap of corresponding comparison graph. Further, for some canonical scenarios, we quantify the price of rank-breaking by comparing the error bound of the proposed data-driven rank-breaking with the lower bound on the MLE, which can have a significantly larger computational cost (Theorem 4).

Notations. Following is a summary of all the notations defined above. We use d to denote the total number of items and index each item by $i \in \{1, 2, \dots, d\}$. $\theta \in \Omega_b$ denotes vector of utilities

associated with each item. θ^* represents true utility and $\hat{\theta}$ denotes the estimated utility. We use n to denote the number of users/agents and index each user by $j \in \{1, 2, \dots, n\}$. $S_j \subseteq \{1, \dots, d\}$ refer to the offerings provided to the j -th user and we use $\kappa_j = |S_j|$ to denote the size of the offerings. G_j denote the DAG (Hasse diagram) representing the partial ordering from user j 's preferences. $P_j = \{p_{j,1}, p_{j,2}, \dots, p_{j,\ell_j}\}$ denotes the set of separators in the DAG G_j , where $p_{j,1}, \dots, p_{j,\ell_j}$ are the positions of the separators, and ℓ_j is the number of separators. $G_{j,a} = (S_j, E_{j,a})$ denote the rank-breaking graph for the a -th separator extracted from the partial ordering G_j of user j .

For any positive integer N , let $[N] = \{1, \dots, N\}$. For a ranking σ over S , i.e., σ is a mapping from $|S|$ to S , let σ^{-1} denote the inverse mapping. For a vector x , let $\|x\|_2$ denote the standard ℓ_2 norm. Let $\mathbf{1}$ denote the all-ones vector and $\mathbf{0}$ denote the all-zeros vector with the appropriate dimension. Let S^d denote the set of $d \times d$ symmetric matrices with real-valued entries. For $X \in S^d$, let $\lambda_1(X) \leq \lambda_2(X) \leq \dots \leq \lambda_d(X)$ denote its eigenvalues sorted in increasing order. Let $\text{Tr}(X) = \sum_{i=1}^d \lambda_i(X)$ denote its trace and $\|X\| = \max\{|\lambda_1(X)|, |\lambda_d(X)|\}$ denote its spectral norm. For two matrices $X, Y \in S^d$, we write $X \succeq Y$ if $X - Y$ is positive semi-definite, i.e., $\lambda_1(X - Y) \geq 0$. Let e_i denote a unit vector in \mathbb{R}^d along the i -th direction.

2. Comparisons Graph and the Graph Laplacian

In the analysis of the convex program (3), we show that, with high probability, the objective function is strictly concave with $\lambda_2(H(\theta)) \leq -C_b \gamma \lambda_2(L) < 0$ (Lemma 11) for all $\theta \in \Omega_b$ and the gradient is bounded by $\|\nabla \mathcal{L}_{\text{RB}}(\theta^*)\|_2 \leq C_b' \sqrt{\log d \sum_{j \in [n]} \ell_j}$ (Lemma 10). Shortly, we will define γ and $\lambda_2(L)$, which captures the dependence on the topology of the data, and C_b' and C_b are constants that only depend on b . Putting these together, we will show that there exists a $\theta \in \Omega_b$ such that

$$\|\hat{\theta} - \theta^*\|_2 \leq \frac{2\|\nabla \mathcal{L}_{\text{RB}}(\theta^*)\|_2}{-\lambda_2(H(\theta))} \leq C_b'' \frac{\sqrt{\log d \sum_{j \in [n]} \ell_j}}{\gamma \lambda_2(L)}.$$

Here $\lambda_2(H(\theta))$ denotes the second largest eigenvalue of a negative semi-definite Hessian matrix $H(\theta)$ of the objective function. The reason the second largest eigenvalue shows up is because the top eigenvector is always the all-ones vector which by the definition of Ω_b is infeasible. The accuracy depends on the topology of the collected data via the comparison graph of given data.

Definition 1. (Comparison graph \mathcal{H}). We define a graph $\mathcal{H}([d], E)$ where each alternative corresponds to a node, and we put an edge (i, i') if there exists an agent j whose offerings is a set S_j such that $i, i' \in S_j$. Each edge $(i, i') \in E$ has a weight $A_{ii'}$ defined as

$$A_{ii'} = \sum_{j \in [n]: i, i' \in S_j} \frac{\ell_j}{\kappa_j(\kappa_j - 1)},$$

where $\kappa_j = |S_j|$ is the size of each sampled set and ℓ_j is the number of separators in S_j defined by rank-breaking in Section 1.

Define a diagonal matrix $D = \text{diag}(A\mathbf{1})$, and the corresponding graph Laplacian $L = D - A$, such that

$$L = \sum_{j=1}^n \frac{\ell_j}{\kappa_j(\kappa_j - 1)} \sum_{i < i' \in S_j} (e_i - e_{i'})(e_i - e_{i'})^\top. \quad (4)$$

Let $0 = \lambda_1(L) \leq \lambda_2(L) \leq \dots \leq \lambda_d(L)$ denote the (sorted) eigenvalues of L . Of special interest is $\lambda_2(L)$, also called the spectral gap, which measured how well-connected the graph is. Intuitively, one can expect better accuracy when the spectral gap is larger, as evidenced in previous learning to rank results in simpler settings (Negahban et al., 2014; Shah et al., 2015a; Hajek et al., 2014). This is made precise in (4), and in the main result of Theorem 2, we appropriately rescale the spectral gap and use $\alpha \in [0, 1]$ defined as

$$\alpha \equiv \frac{\lambda_2(L)(d-1)}{\text{Tr}(L)} = \frac{\lambda_2(L)(d-1)}{\sum_{j=1}^n \ell_j}. \quad (5)$$

The accuracy also depends on the topology via the maximum weighted degree defined as $D_{\max} \equiv \max_{i \in [d]} D_{ii} = \max_{i \in [d]} \left\{ \sum_{j: i \in S_j} \ell_j / \kappa_j \right\}$. Note that the average weighted degree is $\sum_i D_{ii} / d = \text{Tr}(L) / d$, and we rescale it by D_{\max} such that

$$\beta \equiv \frac{\text{Tr}(L)}{dD_{\max}} = \frac{\sum_{j=1}^n \ell_j}{dD_{\max}}. \quad (6)$$

We will show that the performance of rank breaking estimator depends on the topology of the graph through these two parameters. The larger the spectral gap α the smaller error we get with the same effective sample size. The degree imbalance $\beta \in [0, 1]$ determines how many samples are required for the analysis to hold. We need smaller number of samples if the weighted degrees are balanced, which happens if β is large (close to one).

The following quantity also determines the convexity of the objective function.

$$\gamma \equiv \min_{j \in [n]} \left\{ \left(1 - \frac{p_{i\ell_j}}{\kappa_j} \right)^{\lfloor 2e^{2b} \rfloor - 2} \right\}. \quad (7)$$

Note that γ is between zero and one, and a larger value is desired as the objective function becomes more concave and a better accuracy follows. When we are collecting data where the size of the offerings κ_j 's are increasing with d but the position of the separators are close to the top, such that $\kappa_j = \omega(d)$ and $p_{i\ell_j} = O(1)$, then for $b = O(1)$ the above quantity γ can be made arbitrarily close to one, for large enough problem size d . On the other hand, when $p_{i\ell_j}$ is close to κ_j , the accuracy can degrade significantly as stronger alternatives might have small chance of showing up in the rank breaking. The value of γ is quite sensitive to b . The reason we have such a inferior dependence on b is because we wanted to give a universal bound on the Hessian that is simple. It is not difficult to get a tighter bound with a larger value of γ , but will inevitably depend on the structure of the data in a complicated fashion.

To ensure that the (second) largest eigenvalue of the Hessian is small enough, we need enough samples. This is captured by η defined as

$$\eta \equiv \max_{j \in [n]} \{\eta_j\}, \quad \text{where} \quad \eta_j = \frac{\kappa_j}{\max\{\ell_j, \kappa_j - p_{i\ell_j}\}}. \quad (8)$$

Note that $1 < \eta_j \leq \kappa_j / \ell_j$. A smaller value of η is desired as we require smaller number of samples, as shown in Theorem 2. This happens, for instance, when all separators are at the top, such that $p_{i\ell_j} = \ell_j$ and $\eta_j = \kappa_j / (\kappa_j - \ell_j)$, which is close to one for large κ_j . On the other hand, when all separators are at the bottom of the list, then η can be as large as κ_j .

We discuss the role of the topology of data captures by these parameters in Section 4.

3. Main Results

We present the main theoretical results accompanied by corresponding numerical simulations in this section.

3.1 Upper Bound on the Achievable Error

We present the main result that provides an upper bound on the resulting error and explicitly shows the dependence on the topology of the data. As explained in Section 1, we assume that each user provides a partial ranking according to his/her position of the separators. Precisely, we assume the set of offerings S_j , the number of separators ℓ_j , and their respective positions $\mathcal{P}_j = \{p_{j,1}, \dots, p_{j,\ell_j}\}$ are predetermined. Each user draws the ranking of items from the PL model, and provides the partial ranking according to the separators of the form of $\{a > \{b, c, d\} > e > f\}$ in the example in the Figure 1.

Theorem 2. *Suppose there are n users, d items parametrized by $\theta^* \in \Omega_b$, each user j is presented with a set of offerings $S_j \subseteq [d]$, and provides a partial ordering under the PL model. When the effective sample size $\sum_{j=1}^n \ell_j$ is large enough such that*

$$\sum_{j=1}^n \ell_j \geq \frac{2^{11} e^{18b} \eta \log(\ell_{\max} + 2)^2}{\alpha^2 \gamma^2 \beta} d \log d, \quad (9)$$

where $b \equiv \max_i \{\theta_i^*\}$ is the dynamic range, $\ell_{\max} \equiv \max_{j \in [n]} \ell_j$, α is the (rescaled) spectral gap defined in (5), β is the (rescaled) maximum degree defined in (6), γ and η are defined in Eqs. (7) and (8), then the rank-breaking estimator in (3) with the choice of

$$\lambda_{j,a} = \frac{1}{\kappa_j - p_{j,a}}, \quad (10)$$

for all $a \in [\ell_j]$ and $j \in [n]$ achieves

$$\frac{1}{\sqrt{d}} \|\hat{\theta} - \theta^*\|_2 \leq \frac{4\sqrt{2}e^{4b}(1+e^{2b})^2}{\alpha\gamma} \sqrt{\frac{d \log d}{\sum_{j=1}^n \ell_j}}, \quad (11)$$

with probability at least $1 - 3e^3 d^{-3}$.

Consider an ideal case where the spectral gap is large such that α is a strictly positive constant and the dynamic range b is finite and $\max_{j \in [n]} p_{j,\ell_j} / \kappa_j = C$ for some constant $C < 1$ such that γ is also a constant independent of the problem size d . Then the upper bound in (11) implies that we need the effective sample size to scale as $O(d \log d)$, which is only a logarithmic factor larger than the number of parameters to be estimated. Such a logarithmic gap is also unavoidable and due to the fact that we require high probability bounds, where we want the tail probability to decrease at least polynomially in d . We discuss the role of the topology of the data in Section 4.

The upper bound follows from an analysis of the convex program similar to those in (Negahban et al., 2012; Hajek et al., 2014; Shah et al., 2015a). However, unlike the traditional data collection scenarios, the main technical challenge is in analyzing the probability that a particular pair of items appear in the rank-breaking. We provide a proof in Section 8.1.

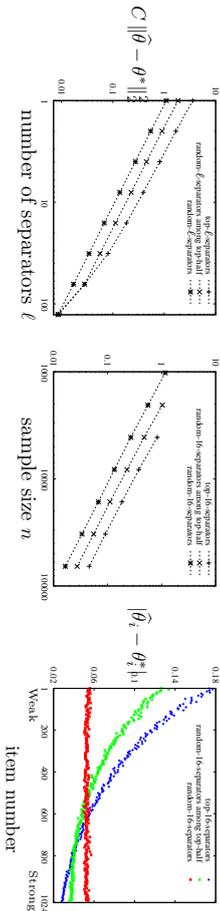


Figure 2: Simulation confirms $\|\hat{\theta}^* - \hat{\theta}\|_2^2 \propto 1/(\ell n)$, and smaller error is achieved for separators that are well spread out.

In Figure 2, we verify the scaling of the resulting error via numerical simulations. We fix $d = 1024$ and $\kappa_j = \kappa = 128$, and vary the number of separators $\ell_j = \ell$ for fixed $n = 128000$ (left), and vary the number of samples n for fixed $\ell_j = \ell = 16$ (middle). Each point is average over 100 instances. The plot confirms that the mean squared error scales as $1/(\ell n)$. Each sample is a partial ranking from a set of κ alternatives chosen uniformly at random, where the partial ranking is from a PL model with weights θ^* chosen i.i.d. uniformly over $[-b, b]$ with $b = 2$. To investigate the role of the position of the separators, we compare three scenarios. The *top- ℓ -separators* choose the top ℓ positions for separators, the *random- ℓ -separators among top-half* choose ℓ positions uniformly random from the top half, and the *random- ℓ -separators* choose the positions uniformly at random. We observe that when the positions of the separators are well spread out among the κ offerings, which happens for *random- ℓ -separators*, we get better accuracy.

The figure on the right provides an insight into this trend for $\ell = 16$ and $n = 16000$. The absolute error $|\hat{\theta}_i^* - \hat{\theta}_i|$ is roughly same for each item $i \in [d]$ when breaking positions are chosen uniformly at random between 1 to $\kappa - 1$ whereas it is significantly higher for weak preference score items when breaking positions are restricted between 1 to $\kappa/2$ or are top- ℓ . This is due to the fact that the probability of each item being ranked at different positions is different, and in particular probability of the low preference score items being ranked in top- ℓ is very small. The third figure is averaged over 1000 instances. Normalization constant C is n/d^2 and $10^3\ell/d^2$ for the first and second figures respectively. For the first figure n is chosen relatively large such that $n\ell$ is large enough even for $\ell = 1$.

3.2 The Price of Rank Breaking for the Special Case of Position- p Ranking

Rank-breaking achieves computational efficiency at the cost of estimation accuracy. In this section, we quantify this tradeoff for a canonical example of position- p ranking, where each sample provides the following information: an unordered set of $p - 1$ items that are ranked high, one item that is ranked at the p -th position, and the rest of $\kappa_j - p$ items that are ranked on the bottom. An example of a sample with position-4 ranking six items $\{a, b, c, d, e, f\}$ might be a partial ranking of $\{(a, b, d) > \{c, f\}\}$. Since each sample has only one separator for $2 < p$, Theorem 2 simplifies to the following Corollary.

Corollary 3. *Under the hypotheses of Theorem 2, there exist positive constants C and c that only depend on b such that if $n \geq C(\eta d \log d)/(\alpha^2 \gamma^2 \beta)$ then*

$$\frac{1}{\sqrt{d}} \|\hat{\theta} - \theta^*\|_2 \leq \frac{c}{\alpha \gamma} \sqrt{\frac{d \log d}{n}}. \quad (12)$$

Note that the error only depends on the position p through γ and η , and is not sensitive. To quantify the price of rank-breaking, we compare this result to a fundamental lower bound on the minimax rate in Theorem 4. We can compute a sharp lower bound on the minimax rate, using the Cramér-Rao bound, and a proof is provided in Section 8.3.

Theorem 4. *Let \mathcal{U} denote the set of all unbiased estimators of θ^* and suppose $b > 0$, then*

$$\inf_{\hat{\theta}} \sup_{\theta^* \in \Omega_b} \mathbb{E} \|\hat{\theta} - \theta^*\|_2^2 \geq \frac{1}{2p \log(\kappa_{\max})^2} \sum_{i=2}^d \frac{1}{\lambda_i(L)} \geq \frac{1}{2p \log(\kappa_{\max})^2} \frac{(d-1)^2}{n},$$

where $\kappa_{\max} = \max_{j \in [n]} |S_j|$ and the second inequality follows from the Jensen's inequality.

Note that the second inequality is tight up to a constant factor, when the graph is an expander with a large spectral gap. For expanders, α in the bound (12) is also a strictly positive constant. This suggests that rank-breaking gains in computational efficiency by a super-exponential factor of $(p-1)!$, at the price of increased error by a factor of p , ignoring poly-logarithmic factors.

3.3 Tighter Analysis for the Special Case of Top- ℓ Separators Scenario

The main result in Theorem 2 is general in the sense that it applies to any partial ranking data that is represented by positions of the separators. However, the bound can be quite loose, especially when γ is small, i.e. $p_j \ell_j$ is close to κ_j . For some special cases, we can tighten the analysis to get a sharper bound. One caveat is that we use a slightly sub-optimal choice of parameters $\lambda_{j,a} = 1/\kappa_j$ instead of $1/(\kappa_j - a)$, to simplify the analysis and still get the order optimal error bound we want. Concretely, we consider a special case of top- ℓ separators scenario, where each agent gives a ranked list of her most preferred ℓ_j alternatives among κ_j offered set of items. Precisely, the locations of the separators are $(p_{j,1}, p_{j,2}, \dots, p_{j,\ell_j}) = (1, 2, \dots, \ell_j)$.

Theorem 5. *Under the PL model, n partial orderings are sampled over d items parametrized by $\theta^* \in \Omega_b$, where the j -th sample is a ranked list of the top- ℓ_j items among the κ_j items offered to the agent. If*

$$\sum_{j=1}^n \ell_j \geq \frac{2^{12} e^{6b}}{\beta \alpha^2} d \log d, \quad (13)$$

where $b \equiv \max_{i,x} |\theta_i^* - \theta_x^*|$ and α, β are defined in (5) and (6), then the rank-breaking estimator in (3) with the choice of $\lambda_{j,a} = 1/\kappa_j$ for all $a \in [k_j]$ and $j \in [n]$ achieves

$$\frac{1}{\sqrt{d}} \|\hat{\theta} - \theta^*\|_2 \leq \frac{16(1 + e^{2b})^2}{\alpha} \sqrt{\frac{d \log d}{\sum_{j=1}^n \ell_j}}, \quad (14)$$

with probability at least $1 - 3e^{-3d}$.

A proof is provided in Section 8.4. In comparison to the general bound in Theorem 2, this is tighter since there is no dependence in γ or η . This gain is significant when, for example, p_{j,ℓ_j} is close to κ_j . As an extreme example, if all agents are offered the entire set of alternatives and are asked to rank all of them, such that $\kappa_j = d$ and $\ell_j = d - 1$ for all $j \in [n]$, then the generic bound in (11) is loose by a factor of $(e^{db}/2\sqrt{2})d^{(2e^{2b})-2}$, compared to the above bound.

In the top- ℓ separators scenario, the data set consists of the ranking among top- ℓ_j items of the set S_j , i.e., $[\sigma_j(1), \sigma_j(2), \dots, \sigma_j(\ell_j)]$. The corresponding log-likelihood of the PL model is

$$\mathcal{L}(\theta) = \sum_{j=1}^n \sum_{m=1}^{\ell_j} \left[\theta_{\sigma_j(m)} - \log \left(\exp(\theta_{\sigma_j(m)}) + \exp(\theta_{\sigma_j(m+1)}) + \dots + \exp(\theta_{\sigma_j(\kappa_j)}) \right) \right], \quad (15)$$

where $\sigma_j(a)$ is the alternative ranked at the a -th position by agent j . The Maximum Likelihood Estimator (MLE) for this *traditional* data set is efficient. Hence, there is no computational gain in rank-breaking. Consequently, there is no loss in accuracy either, when we use the optimal weights proposed in the above theorem. Figure 3 illustrates that the MLE and the data-driven rank-breaking estimator achieve performance that is identical, and improve over naive rank-breaking that uses uniform weights. We also compare performance of Generalized Method-of-Moments (GMM) proposed by Azari Soufiani et al. (2013) with our algorithm. In addition, we show that performance of GMM can be improved by optimally weighing pairwise comparisons with $\lambda_{j,a}$. MSE of GMM in both the cases, uniform weights and optimal weights, is larger than our rank-breaking estimator. However, GMM is on average about four times faster than our algorithm. We choose $\lambda_{j,a} = 1/(\kappa_j - a)$ in the simulations, as opposed to the $1/\kappa_j$ assumed in the above theorem. This settles the question raised in Hajek et al. (2014) on whether it is possible to achieve optimal accuracy using rank-breaking under the top- ℓ separators scenario. Analytically, it was proved in (Hajek et al., 2014) that under the top- ℓ separators scenario, naive rank-breaking with uniform weights achieves the same error bound as the MLE, up to a constant factor. However, we show that this constant factor gap is not a weakness of the analyses, but the choice of the weights. Theorem 5 provides a guideline for choosing the optimal weights, and the numerical simulation results in Figure 3 show that there is in fact no gap in practice, if we use the optimal weights. We use the same settings as that of the first figure of Figure 2 for the figure below.

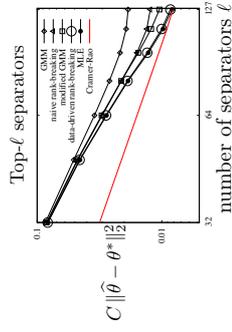


Figure 3: The proposed data-driven rank-breaking achieves performance identical to the MLE, and improves over naive rank-breaking with uniform weights.

To prove the order-optimality of the rank-breaking approach up to a constant factor, we can compare the upper bound to a Cramér-Rao lower bound on any unbiased estimators, in the following theorem. A proof is provided in Section 8.5.

Theorem 6. Consider ranking $\{\sigma_j(i)\}_{i \in [\ell_j]}$ revealed for the set of items S_j , for $j \in [n]$. Let \mathcal{U} denote the set of all unbiased estimators of $\theta^* \in \Omega_b$. If $b > 0$, then

$$\inf_{\hat{\theta} \in \mathcal{U}} \sup_{\theta^* \in \Omega_b} \mathbb{E}[\|\hat{\theta} - \theta^*\|^2] \geq \left(1 - \frac{1}{\ell_{\max}} \sum_{i=1}^{\ell_{\max}} \frac{1}{\kappa_{\max} - i + 1} \right)^{-1} \sum_{i=2}^d \frac{1}{\lambda_i(L)} \geq \frac{(d-1)^2}{\sum_{j=1}^n \ell_j}, \quad (16)$$

where $\ell_{\max} = \max_{j \in [n]} \ell_j$ and $\kappa_{\max} = \max_{j \in [n]} \kappa_j$. The second inequality follows from the Jensen's inequality.

Consider a case when the comparison graph is an expander such that α is a strictly positive constant, and $b = O(1)$ is also finite. Then, the Cramér-Rao lower bound show that the upper bound in (14) is optimal up to a logarithmic factor.

3.4 Optimality of the Choice of the Weights

We propose the optimal choice of the weights $\lambda_{j,a}$'s in Theorem 2. In this section, we show numerical simulations results comparing the proposed approach to other naive choices of the weights under various scenarios. We fix $d = 1024$ items and the underlying preference vector θ^* is uniformly distributed over $[-b, b]$ for $b = 2$. We generate n rankings over sets S_j of size κ for $j \in [n]$ according to the PL model with parameter θ^* . The comparison sets S_j 's are chosen independently and uniformly at random from $[d]$.

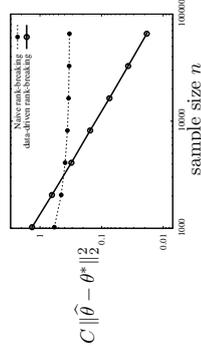


Figure 4: Data-driven rank-breaking is consistent, while a random rank-breaking results in inconsistency.

Figure 4 illustrates that a naive choice of rank-breakings can result in inconsistency. We create partial orderings data set by fixing $\kappa = 128$ and select $\ell = 8$ random positions in $\{1, \dots, 127\}$. Each data set consists of partial orderings with separators at those 8 random positions, over 128 randomly chosen subset of items. We vary the sample size n and plot the resulting mean squared error for the two approaches. The data-driven rank-breaking, which uses the optimal choice of the weights, achieves error scaling as $1/n$ as predicted by Theorem 2, which implies consistency. For fair comparisons, we feed the same number of pairwise orderings to a naive rank-breaking estimator. This estimator uses randomly chosen pairwise orderings with uniform weights, and is

generally inconsistent. However, when sample size is small, inconsistent estimators can achieve smaller variance leading to smaller error. Normalization constant C is $10^3\ell/d^2$, and each point is averaged over 100 trials. We use the minorization-maximization algorithm from Hunter (2004) for computing the estimates from the rank-breakings.

Even if we use the consistent rank-breakings first proposed in Azari Soufiani et al. (2014), there is ambiguity in the choice of the weights. We next study how much we gain by using the proposed optimal choice of the weights. The optimal choice, $\lambda_{j,a} = 1/(\kappa_j - p_{j,a})$, depends on two parameters: the size of the offerings κ_j and the position of the separators $p_{j,a}$. To distinguish the effect of these two parameters, we first experiment with fixed $\kappa_j = \kappa$ and illustrate the gain of the optimal choice of $\lambda_{j,a}$'s.

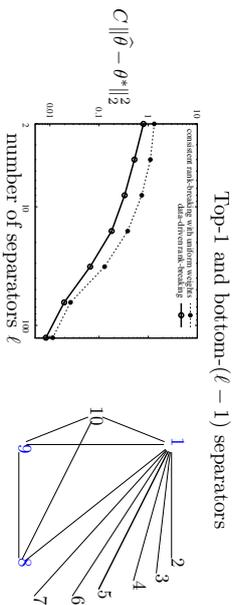


Figure 5: There is a constant factor gain of choosing optimal $\lambda_{j,a}$'s when the size of offerings are fixed, i.e. $\kappa_j = \kappa$ (left). We choose a particular set of separators where one separator is at position one and the rest are at the bottom. An example for $\ell = 3$ and $\kappa = 10$ is shown, where the separators are indicated by blue (right).

Figure 5 illustrates that the optimal choice of the weights improves over consistent rank-breaking with uniform weights by a constant factor. We fix $\kappa = 128$ and $n = 128000$. As illustrated by a figure on the right, the position of the separators are chosen such that there is one separator at position one, and the rest of $\ell - 1$ separators are at the bottom. Precisely, $(p_{j,1}, p_{j,2}, p_{j,3}, \dots, p_{j,\ell}) = (1, 128 - \ell + 1, 128 - \ell + 2, \dots, 127)$. We consider this scenario to emphasize the gain of optimal weights. Observe that the MSE does not decrease at a rate of $1/\ell$ in this case. The parameter γ which appears in the bound of Theorem 2 is very small when the breaking positions $p_{j,a}$ are of the order κ_j as is the case here, when ℓ is small. Normalization constant C is n/d^2 .

The gain of optimal weights is significant when the size of S_j 's are highly heterogeneous. Figure 6 compares performance of the proposed algorithm, for the optimal choice and uniform choice of weights $\lambda_{j,a}$ when the comparison sets S_j 's are of different sizes. We consider the case when n_1 agents provide their top- ℓ_1 choices over the sets of size κ_1 , and n_2 agents provide their top-1 choice over the sets of size κ_2 . We take $n_1 = 1024$, $\ell_1 = 8$, and $n_2 = 10n_1\ell_1$. Figure 6 shows MSE for the two choice of weights, when we fix $\kappa_1 = 128$, and vary κ_2 from 2 to 128. As predicted from our bounds, when optimal choice of $\lambda_{j,a}$ is used MSE is not sensitive to sample set sizes κ_2 . The error decays at the rate proportional to the inverse of the effective sample size, which is $n_1\ell_1 + n_2\ell_2 = 11n_1\ell_1$. However, with $\lambda_{j,a} = 1$ when $\kappa_2 = 2$, the MSE is roughly 10 times worse. Which reflects that the effective sample size is approximately $n_1\ell_1$, i.e. pairwise comparisons counting

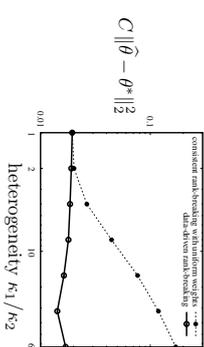


Figure 6: The gain of choosing optimal $\lambda_{j,a}$'s is significant when κ_j 's are highly heterogeneous.

from small set size do not contribute without proper normalization. This gap in MSE corroborates bounds of Theorem 8. Normalization constant C is $10^3/d^2$.

4. The Role of the Topology of the Data

We study the role of topology of the data that provides a guideline for designing the collection of data when we do have some control, as in recommendation systems, designing surveys, and crowdsourcing. The core optimization problem of interest to the designer of such a system is to achieve the best accuracy while minimizing the number of questions.

4.1 The Role of the Graph Laplacian

Using the same number of samples, comparison graphs with larger spectral gap achieve better accuracy, compared to those with smaller spectral gaps. To illustrate how graph topology effects the performance, we reproduce known spectral properties of canonical graphs, and numerically compare the performance of data-driven rank-breaking for several graph topologies. We follow the examples and experimental setup from Shah et al. (2015a) for a similar result with pairwise comparisons. Spectral properties of graphs have been a topic of wide interest for decades. We consider a scenario where we fix the size of offerings as $\kappa_j = \kappa = O(1)$ and each agent provides partial ranking with ℓ separators, positions of which are chosen uniformly at random. The resulting spectral gap α of different choices of the set S_j 's are provided below. The total number edges in the comparisons graph (counting hyper-edges as multiple edges) is defined as $|E| \equiv \binom{\kappa}{\ell} n$.

- Complete graph: when $|E|$ is larger than $\binom{d}{\ell}$, we can design the comparison graph to be a complete graph over d nodes. The weight A_{ij} on each edge is $n\ell/(d(d-1))$, which is the effective number of samples divided by twice the number of edges. Resulting spectral gap is one, which is the maximum possible value. Hence, complete graph is optimal for rank aggregation.
- Sparse random graph: when we have limited resources we might not be able to afford a dense graph. When $|E|$ is of order d^2 , we have a sparse graph. Consider a scenario where each set S_j is chosen uniformly at random. To ensure connectivity, we need $n = \Omega(\log d)$. Following standard spectral analysis of random graphs, we have $\alpha = \Theta(1)$. Hence, sparse random graphs are near-optimal for rank-aggregation.

- Chain graph: we consider a chain of sets of size κ overlapping only by one item. For example, $S_1 = \{1, \dots, \kappa\}$ and $S_2 = \{\kappa, \kappa + 1, \dots, 2\kappa - 1\}$, etc. We choose n to be a multiple of $\tau \equiv (d - 1)/(\kappa - 1)$ and offer each set n/τ times. The resulting graph is a chain of size κ cliques, and standard spectral analysis shows that $\alpha = \Theta(1/d^2)$. Hence, a chain graph is strictly sub-optimal for rank aggregation.
- Star-like graph: We choose one item to be the center, and every offer set consists of this center node and a set of $\kappa - 1$ other nodes chosen uniformly at random without replacement. For example, center node = $\{1\}$, $S_1 = \{1, 2, \dots, \kappa\}$ and $S_2 = \{1, \kappa + 1, \kappa + 2, \dots, 2\kappa - 1\}$, etc. n is chosen in the way similar to that of the Chain graph. Standard spectral analysis shows that $\alpha = \Theta(1)$ and star-like graphs are near-optimal for rank-aggregation.
- Barbell-like graph: We select an offering $S = \{S', i, j\}$, $|S'| = \kappa - 2$ uniformly at random and divide rest of the items into two groups V_1 and V_2 . We offer set S $n\kappa/d$ times. For each offering of set S , we offer $d/\kappa - 1$ sets chosen uniformly at random from the two groups $\{V_1, i\}$ and $\{V_2, j\}$. The resulting graph is a barbell-like graph, and standard spectral analysis shows that $\alpha = \Theta(1/d^2)$. Hence, a chain graph is strictly sub-optimal for rank aggregation.

Figure 7 illustrates how graph topology effects the accuracy. When θ^* is chosen uniformly at random, the accuracy does not change with d (left), and the accuracy is better for those graphs with larger spectral gap. However, for a certain worst-case θ^* , the error increases with d for the chain graph and the barbell-like graph, as predicted by the above analysis of the spectral gap. We use $\ell = 4$, $\kappa = 17$ and vary d from 129 to 2049. κ is kept small to make the resulting graphs more like the above discussed graphs. Figure on left shows accuracy when θ^* is chosen i.i.d. uniformly over $[-b, b]$ with $b = 2$. Error in this case is roughly same for each of the graph topologies with chain graph being the worst. However, when θ^* is chosen carefully for chain graph and barbell-like graph increases with d as shown in the figure right. We chose θ^* such that all the items of a set have same weight, either $\theta_i = 0$ or $\theta_i = b$ for chain graph and barbell-like graph. We divide all the sets equally between the two types for chain graph. For barbell-like graph, we keep the two types of sets on the two different sides of the connector set and equally divide items of the connector set into two types. Number of samples n is $100(d - 1)/(\kappa - 1)$ and each point is averaged over 100 instances. Normalization constant C is $n\ell/d^2$.

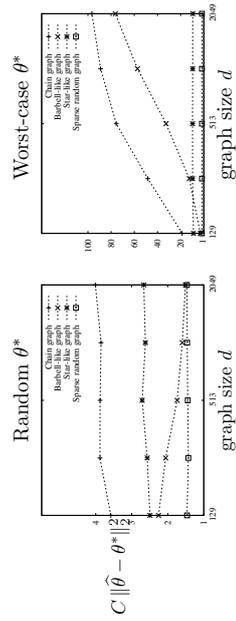


Figure 7: For randomly chosen θ^* the error does not change with d (left). However, for particular worst-case θ^* the error increases with d for the Chain graph and the Barbell-like graph as predicted by the analysis of the spectral gap (right).

4.2 The Role of the Position of the Separators

As predicted by theorem 2, rank-breaking fails when γ is small, i.e. the position of the separators are very close to the bottom. An extreme example is the bottom- ℓ separators scenario, where each person is offered κ randomly chosen alternatives, and is asked to give a ranked list of bottom ℓ alternatives. In other words, the ℓ separators are placed at $(p_{j,1}, \dots, p_{j,\ell}) = (\kappa_j - \ell, \dots, \kappa_j - 1)$. In this case, $\gamma \simeq 0$ and the error bound is large. This is not a weakness of the analysis. In fact we observe large errors under this scenario. The reason is that many alternatives that have large weights θ_i 's will rarely be even compared once, making any reasonable estimation infeasible.

Figure 8 illustrates this scenario. We choose $\ell = 8$, $\kappa = 128$, and $d = 1024$. The other settings are same as that of the first figure of Figure 2. The left figure plots the magnitude of the estimation error for each item. For about 200 strong items among 1024, we do not even get a single comparison, hence we omit any estimation error. It clearly shows the trend: we get good estimates for about 400 items in the bottom, and we get large errors for the rest. Consequently, even if we only take those items that have at least one comparison into account, we still get large errors. This is shown in the figure right. The error barely decays with the sample size. However, if we focus on the error for the bottom 400 items, we get good error rate decaying inversely with the sample size. Normalization constant C in the second figure is $10^2 \cdot x \cdot d/\ell$ and $10^2(400)d/\ell$ for the first and second lines respectively, where x is the number of items that appeared in rank-breaking at least once. We solve convex program (3) for θ restricted to the items that appear in rank-breaking at least once. The second figure of Figure 8 is averaged over 1000 instances.

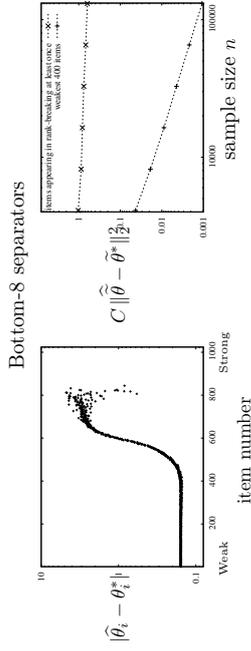


Figure 8: Under the bottom- ℓ separators scenario, accuracy is good only for the bottom 400 items (left). As predicted by Theorem 7, the mean squared error on the bottom 400 items scale as $1/n$, where as the overall mean squared error does not decay (right).

We make this observation precise in the following theorem. Applying rank-breaking to only to those weakest \tilde{d} items, we prove an upper bound on the achieved error rate that depends on the choice of the \tilde{d} . Without loss of generality, we suppose the items are sorted such that $\theta_1^* \leq \theta_2^* \leq \dots \leq \theta_d^*$. For a choice of $\tilde{d} = \ell d / (2\kappa)$, we denote the weakest \tilde{d} items by $\theta^* \in \mathbb{R}^{\tilde{d}}$ such that $\theta_i^* = \theta_{\ell i}^* - (1/\tilde{d}) \sum_{j=i}^{\tilde{d}} \theta_j^*$, for $i \in [\tilde{d}]$. Since $\theta^* \in \Omega_b$, $\theta^* \in [-2b, 2b]^{\tilde{d}}$. The space of all possible preference vectors for $[\tilde{d}]$ items is given by $\tilde{\Omega} = \{\theta \in \mathbb{R}^{\tilde{d}} : \sum_{i=1}^{\tilde{d}} \theta_i = 0\}$ and $\tilde{\Omega}_{2b} = \tilde{\Omega} \cap [-2b, 2b]^{\tilde{d}}$.

Although the analysis can be easily generalized, to simplify notations, we fix $\kappa_j = \kappa$ and $\ell_j = \ell$ and assume that the comparison sets S_j , $|S_j| = \kappa$, are chosen uniformly at random from the set of

d items for all $j \in [n]$. The rank-breaking log likelihood function $\mathcal{L}_{\text{RB}}(\tilde{\theta})$ for the set of items $[d]$ is given by

$$\mathcal{L}_{\text{RB}}(\tilde{\theta}) = \sum_{j=1}^n \sum_{a=1}^{\ell_j} \lambda_{j,a} \left\{ \sum_{(i,i') \in E_{j,a}} \mathbb{I}_{\{i,i' \in [d]\}} \left(\theta_{i'} - \log \left(e^{\theta_i} + e^{\theta_{i'}} \right) \right) \right\}. \quad (17)$$

We analyze the rank-breaking estimator

$$\hat{\tilde{\theta}} \equiv \max_{\tilde{\theta} \in \Omega_{2b}} \mathcal{L}_{\text{RB}}(\tilde{\theta}). \quad (18)$$

We further simplify notations by fixing $\lambda_{j,a} = 1$, since from Equation (24), we know that the error increases by at most a factor of 4 due to this sub-optimal choice of the weights, under the special scenario studied in this theorem.

Theorem 7. *Under the bottom- ℓ separators scenario and the PL model, S_j 's are chosen uniformly at random of size κ and n partial orderings are sampled over d items parametrized by $\theta^* \in \Omega_b$. For $d = \ell d / (2\kappa)$ and any $\ell \geq 4$, if the effective sample size is large enough such that*

$$n\ell \geq \left(\frac{2^{14} e^{8b} \kappa^3}{\chi^2 \ell^3} \right) d \log d, \quad (19)$$

where

$$\chi \equiv \frac{1}{4} \left(1 - \exp \left(- \frac{2}{9(\kappa - 2)} \right) \right), \quad (20)$$

then the rank-breaking estimator \hat{m} (18) achieves

$$\frac{1}{\sqrt{d}} \|\hat{\tilde{\theta}} - \tilde{\theta}^*\|_2 \leq \frac{128(1 + e^{4b})^2 \kappa^3 \ell^2}{\chi \ell^{3/2}} \sqrt{\frac{d \log d}{n\ell}}, \quad (21)$$

with probability at least $1 - 3e^3 d^{-3}$.

Consider a scenario where $\kappa = O(1)$ and $\ell = \Theta(\kappa)$. Then, χ is a strictly positive constant, and also κ/ℓ is a finite constant. It follows that rank-breaking requires the effective sample size $n\ell = O(d \log d / \varepsilon^2)$ in order to achieve arbitrarily small error of $\varepsilon > 0$, on the weakest $\tilde{d} = \ell d / (2\kappa)$ items.

5. Real-World Data Sets

On real-world data sets on sushi preferences (Kamishima, 2003), we show that the data-driven rank-breaking improves over Generalized Method-of-Moments (GMM) proposed by Azari Soufiani et al. (2013). This is a widely used data set for rank aggregation, for instance in Azari Soufiani et al. (2013, 2012); Maystre and Grossglauser (2015b); Le Van et al. (2015); Lu and Boutlier (2011a, b). The data set consists of complete rankings over 10 types of sushi from $n = 5000$ individuals. Below, we follow the experimental scenarios of the GMM approach in Azari Soufiani et al. (2013) for fair comparisons.

To validate our approach, we first take the estimated PL weights of the 10 types of sushi, using Hunter (2004) implementation of the ML estimator, over the entire input data of 5000 complete rankings. We take thus created output as the ground truth θ^* . To create partial rankings and compare the performance of the data-driven rank-breaking to the state-of-the-art GMM approach in Figure 9, we first fix $\ell = 6$ and vary n to simulate top- ℓ separators scenario by removing the known ordering among bottom $10 - \ell$ alternatives for each sample in the data set (left). We next fix $n = 1000$ and vary ℓ and simulate top- ℓ separators scenarios (right). Each point is averaged over 1000 instances. The mean squared error is plotted for both algorithms.

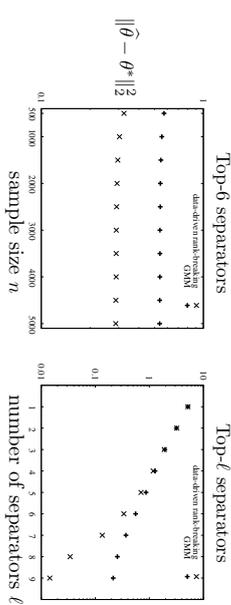


Figure 9: The data-driven rank-breaking achieves smaller error compared to the state-of-the-art GMM approach.

Figure 10 illustrates the Kendall rank correlation of the rankings estimated by the two algorithms and the ground truth. Larger value indicates that the estimate is closer to the ground truth, and the data-driven rank-breaking outperforms the state-of-the-art GMM approach.

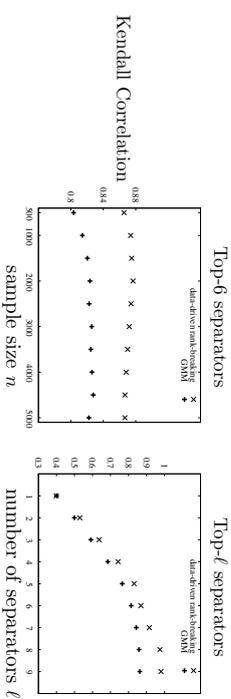


Figure 10: The data-driven rank-breaking achieves larger Kendall rank correlation compared to the state-of-the-art GMM approach.

To validate whether PL model is the right model to explain the sushi data set, we compare the data-driven rank-breaking, MLE for the PL model, GMM for the PL model, Borda count and Spearman's footrule optimal aggregation. We measure the Kendall rank correlation between the estimates and the samples and show the result in Table 1. In particular, if $\sigma_1, \sigma_2, \dots, \sigma_n$ denote sample rankings and $\hat{\sigma}$ denote the aggregated ranking then the correlation value is $(1/n) \sum_{i=1}^n (1 - \frac{4K(\hat{\sigma}, \sigma_i)}{K(\hat{\sigma}, \hat{\sigma})})$, where $K(\sigma_1, \sigma_2) = \sum_{i < j \in [k]} \mathbb{I}_{(\sigma_1^{-1}(i) - \sigma_1^{-1}(j))(\sigma_2^{-1}(i) - \sigma_2^{-1}(j)) < 0}$. The results are reported

for different number of samples n and different values of ℓ under the top- ℓ separators scenarios. When $\ell = 9$, we are using all the complete rankings, and all algorithms are efficient. When $\ell < 9$, we have partial orderings, and Spearman’s footrule optimal aggregation is NP-hard. We instead use scaled footrule aggregation (SFO) given in Dwork et al. (2001). Most approaches achieve similar performance, except for the Spearman’s footrule. The proposed data-driven rank-breaking achieves a slightly worse correlation compared to other approaches. However, note that none of the algorithms are necessarily maximizing the Kendall correlation, and are not expected to be particularly good in this metric.

	MLE under PL	data-driven RB	GMM	Borda count	Spearman’s footrule
$n = 500, \ell = 9$	0.306	0.291	0.315	0.315	0.159
$n = 5000, \ell = 9$	0.309	0.309	0.315	0.315	0.079
$n = 5000, \ell = 2$	0.199	0.199	0.201	0.200	0.113
$n = 5000, \ell = 5$	0.217	0.200	0.217	0.295	0.152

Table 1: Kendall rank correlation on sushi data set.

We compare our algorithm with the GMM algorithm on two other real-world data-sets as well. We use jester data set (Goldberg et al., 2001) that consists of over 4.1 million continuous ratings between -10 to $+10$ of 100 jokes from 48,483 users. The average number of jokes rated by an user is 72.6 with minimum and maximum being 36 and 100 respectively. We convert continuous ratings into ordinal rankings. This data-set has been used by Miyahara and Pazzani (2000); Polat and Du (2005); Cortes et al. (2007); Lebanon and Mao (2007) for rank aggregation and collaborative filtering.

Similar to the settings of sushi data experiments, we take the estimated PL weights of the 100 jokes over all the rankings as ground truth. Figure 11 shows comparative performance of the data-driven rank-breaking and the GMM for the two scenarios. We first fix $\ell = 10$ and vary n to simulate random-10 separators scenario (left). We next take all the rankings $n = 73421$ and vary ℓ to simulate random- ℓ separators scenario (rights). Since sets have different sizes, while varying ℓ we use full breaking if the setsize is smaller than ℓ . Each point is averaged over 100 instances. The mean squared error is plotted for both algorithms.

We perform similar experiments on American Psychological Association (APA) data-set (Diaconis, 1989). The APA elects a president each year by asking each member to rank order a slate of five candidates. The data-set represents full rankings given by 5738 members of the association in 1980’s election. The mean squared error is plotted for both algorithms under the settings similar to that of jester data-set.

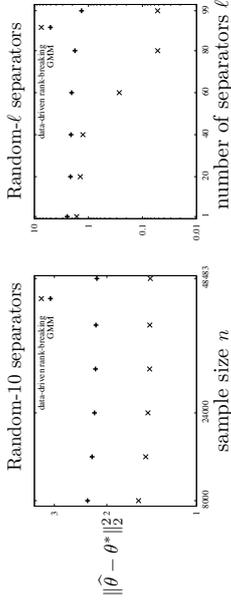


Figure 11: jester data set: The data-driven rank-breaking achieves smaller error compared to the state-of-the-art GMM approach.

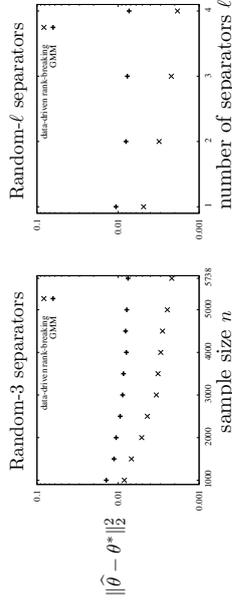


Figure 12: APA data set: The data-driven rank-breaking achieves smaller error compared to the state-of-the-art GMM approach.

6. Related Work

Initially motivated by elections and voting, rank aggregation has been a topic of mathematical interest dating back to Condorcet and Borda (De Condorcet, 1785; de Borda, 1781). Using probabilistic models to infer preferences has been popularized in operations research community for applications such as assortment optimization and revenue management. The PL model studied in this paper is a special case of MultiNomial Logit (MNL) models commonly used in discrete choice modeling, which has a long history in operations research (McFadden, 1980). Efficient inference algorithms has been proposed to either find the MLE efficiently or approximately, such as the iterative approaches in Ford Jr. (1957); Dykstra (1960), minorization-maximization approach in Hunter (2004), and Markov chain approaches in Negahban et al. (2012); Maystre and Grossglauser (2015a). These approaches are shown to achieve minimax optimal error rate in the traditional comparisons scenarios. Under the pairwise comparisons scenario, Negahban et al. (2012) provided Rank Centrality that provably achieves minimax optimal error rate for randomly chosen pairs, which was later generalized to arbitrary pairwise comparisons in Negahban et al. (2014). The analysis shows the explicit dependence on the topology of data shows that the spectral gap of comparisons graph similar to the one presented in this paper. This analysis was generalized to k -way comparisons in Hajek et al. (2014) and generalized to best-of- k comparisons with sharper bounds in Shah et al. (2015a). In an effort to give a guarantee for exact recovery of the top- ℓ items in the ranking, Chen

and Suh (2015) proposed a new algorithm based on Rank Centrality that provides a tighter error bound for L_∞ norm, as opposed to the existing L_2 error bounds. Another interesting direction in learning to rank is non-parametric learning from paired comparisons, initiated in several recent papers such as Duchi et al. (2010); Rajkumar and Agarwal (2014); Shah et al. (2015b); Shah and Wainwright (2015).

More recently, a more general problem of learning *personal* preferences from ordinal data has been studied (Yi et al., 2013; Lu and Boutilier, 2011b; Ding et al., 2015). The MNL model provides a natural generalization of the PL model to this problem. When users are classified into a small number of groups with same preferences, mixed MNL model can be learned from data as studied in Ammar et al. (2014); Oh and Shah (2014); Wu et al. (2015). A more general scenario is when each user has his/her individual preferences, but inherently represented by a lower dimensional feature. This problem was first posed as an inference problem in Lu and Negahban (2014) where convex relaxation of nuclear norm minimization was proposed with provably optimal guarantees. This was later generalized to k -way comparisons in Oh et al. (2015). A similar approach was studied with a different guarantees and assumptions in Park et al. (2015). Our algorithm and ideas of rank-breaking can be directly applied to this collaborative ranking under MNL, with the same guarantees for consistency in the asymptotic regime where sample size grows to infinity. However, the analysis techniques for MNL rely on stronger assumptions on how the data is collected, and especially on the independence of the samples. It is not immediate how the analysis techniques developed in this paper can be applied to learn MNL.

In an orthogonal direction, new discrete choice models with sparse structures has been proposed recently in Farias et al. (2009) and optimization algorithms for revenue management has been proposed Farias et al. (2013). In a similar direction, new discrete choice models based on Markov chains has been introduced in Blanchet et al. (2013), and corresponding revenue management algorithms has been studied in Feldman and Topaloglu (2014). However, typically these models are analyzed in the asymptotic regime with infinite samples, with the exception of Ammar and Shah (2011). A non-parametric choice models for pairwise comparisons also have been studied in Rajkumar and Agarwal (2014); Shah et al. (2015b). This provides an interesting opportunities to studying learning to rank for these new choice models.

We consider a fixed design setting, where inference is separate from data collection. There is a parallel line of research which focuses on adaptive ranking, mainly based on pairwise comparisons. When performing sorting from noisy pairwise comparisons, Braverman and Mossel (2009) proposed efficient approaches and provided performance guarantees. Following this work, there has been recent advances in adaptive ranking Alon (2011); Jannison and Nowak (2011); Maystre and Grossglauser (2015b).

7. Discussion

We study the problem of learning the PL model from ordinal data. Under the traditional data collection scenarios, several efficient algorithms find the maximum likelihood estimates and at the same time provably achieve minimax optimal performance. However, for some non-traditional scenarios, computational complexity of finding the maximum likelihood estimate can scale super-exponentially in the problem size. We provide the first finite-sample analysis of computationally efficient estimators known as rank-breaking estimators. This provides guidelines for choosing the

weights in the estimator to achieve optimal performance, and also explicitly shows how the accuracy depends on the topology of the data.

This paper provides the first analytical result in the sample complexity of rank-breaking estimators, and quantifies the price we pay in accuracy for the computational gain. In general, more complex higher-order rank-breaking can also be considered, where instead of breaking a partial ordering into a collection of paired comparisons, we break it into a collection of higher-order comparisons. The resulting higher-order rank-breakings will enable us to traverse the whole spectrum of computational complexity between the pairwise rank-breaking and the MLE. We believe this paper opens an interesting new direction towards understanding the whole spectrum of such approaches. However, analyzing the Hessian of the corresponding objective function is significantly more involved and requires new technical innovations.

8. Proofs

8.1 Proof of Theorem 2

We prove a more general result for an arbitrary choice of the parameter $\lambda_{j,a} > 0$ for all $j \in [n]$ and $a \in [f_j]$. The following theorem proves the (near)-optimality of the choice of $\lambda_{j,a}$'s proposed in (10), and implies the corresponding error bound as a corollary.

Theorem 8. *Under the hypotheses of Theorem 2 and any $\lambda_{j,a}$'s, the rank-breaking estimator achieves*

$$\frac{1}{\sqrt{d}} \|\hat{\theta} - \theta^*\|_2 \leq \frac{4\sqrt{2}e^{4\theta} (1 + e^{2\theta})^2 \sqrt{d \log d} \sqrt{\sum_{j=1}^n \sum_{a=1}^{f_j} (\lambda_{j,a})^2 (\kappa_j - p_{j,a} + 1)}}{\alpha^\gamma \sum_{j=1}^n \sum_{a=1}^{f_j} \lambda_{j,a} (\kappa_j - p_{j,a})}, \quad (22)$$

with probability at least $1 - 3e^3 d^{-3}$, if

$$\sum_{j=1}^n \sum_{a=1}^{f_j} \lambda_{j,a} (\kappa_j - p_{j,a}) \geq 2^6 e^{18\theta} \frac{\eta \delta}{\alpha^2 \beta \gamma^2 \tau} d \log d, \quad (23)$$

where γ , η , τ , δ , α , β , are now functions of $\lambda_{j,a}$'s and defined in (7), (8), (25), (27) and (30).

We first claim that $\lambda_{j,a} = 1/(\kappa_j - p_{j,a} + 1)$ is the optimal choice for minimizing the above upper bound on the error. From Cauchy-Schwarz inequality and the fact that all terms are non-negative, we have that

$$\frac{\sqrt{\sum_{j=1}^n \sum_{a=1}^{f_j} (\lambda_{j,a})^2 (\kappa_j - p_{j,a}) (\kappa_j - p_{j,a} + 1)}}{\sum_{j=1}^n \sum_{a=1}^{f_j} \lambda_{j,a} (\kappa_j - p_{j,a})} \geq \frac{1}{\sqrt{\sum_{j=1}^n \sum_{a=1}^{f_j} \frac{(\kappa_j - p_{j,a})}{(\kappa_j - p_{j,a} + 1)}}}, \quad (24)$$

where $\lambda_{j,a} = 1/(\kappa_j - p_{j,a} + 1)$ achieves the universal lower bound on the right-hand side with an equality. Since $\sum_{j=1}^n \sum_{a=1}^{f_j} \frac{(\kappa_j - p_{j,a})}{(\kappa_j - p_{j,a} + 1)} \geq \sum_{j=1}^n f_j$, substituting this into (22) gives the desired error bound in (11). Although we have identified the optimal choice of $\lambda_{j,a}$'s, we choose a slightly different value of $\lambda = 1/(\kappa_j - p_{j,a})$ for the analysis. This achieves the same desired error bound in (11), and significantly simplifies the notations of the sufficient conditions.

We first define all the parameters in the above theorem for general $\lambda_{j,a}$. With a slight abuse of notations, we use the same notations for \mathcal{H} , L , α and β for both the general $\lambda_{j,a}$'s and also the specific choice of $\lambda_{j,a} = 1/(\kappa_j - p_{j,a})$. It should be clear from the context what we mean in each case. Define

$$\tau \equiv \min_{j \in [n]} \tau_j, \quad \text{where } \tau_j \equiv \frac{\sum_{a=1}^{\ell_j} \lambda_{j,a} (\kappa_j - p_{j,a})}{\ell_j} \quad (25)$$

$$\delta_{j,1} \equiv \left\{ \max_{a \in [\ell_j]} \left\{ \lambda_{j,a} (\kappa_j - p_{j,a}) \right\} + \sum_{a=1}^{\ell_j} \lambda_{j,a} \right\}, \quad \text{and } \delta_{j,2} \equiv \sum_{a=1}^{\ell_j} \lambda_{j,a} \quad (26)$$

$$\delta \equiv \max_{j \in [n]} \left\{ 4\delta_{j,1}^2 + \frac{2(\delta_{j,1}\delta_{j,2} + \delta_{j,2}^2)\kappa_j}{\eta_j \ell_j} \right\}. \quad (27)$$

Note that $\delta \geq \delta_{j,1}^2 \geq \max_a \lambda_{j,a}^2 (\kappa_j - p_{j,a})^2 \geq \tau^2$, and for the choice of $\lambda_{j,a} = 1/(\kappa_j - p_{j,a})$ it simplifies as $\tau = \tau_j = 1$. We next define a comparison graph \mathcal{H} for general $\lambda_{j,a}$, which recovers the proposed comparison graph for the optimal choice of $\lambda_{j,a}$'s

Definition 9. (Comparison graph \mathcal{H}). Each item $i \in [d]$ corresponds to a vertex i . For any pair of vertices i, i' , there is a weighted edge between them if there exists a set S_j such that $i, i' \in S_j$; the weight equals $\sum_{j:i, i' \in S_j} \frac{\tau_j \ell_j}{\kappa_j (\kappa_j - 1)}$.

Let A denote the weighted adjacency matrix, and let $D = \text{diag}(A1)$. Define,

$$D_{\max} \equiv \max_{i \in [d]} \left\{ \sum_{j:i \in S_j} \frac{\tau_j \ell_j}{\kappa_j} \right\} \geq \tau \min_{i \in [d]} \left\{ \sum_{j:i \in S_j} \frac{\ell_j}{\kappa_j} \right\}. \quad (28)$$

Define graph Laplacian L as $L = D - A$, i.e.,

$$L = \sum_{j=1}^n \frac{\tau_j \ell_j}{\kappa_j (\kappa_j - 1)} \sum_{i < i' \in S_j} (e_i - e_{i'}) (e_i - e_{i'})^\top. \quad (29)$$

Let $0 = \lambda_1(L) \leq \lambda_2(L) \leq \dots \leq \lambda_d(L)$ denote the sorted eigenvalues of L . Note that $\text{Tr}(L) = \sum_{i=1}^d \sum_{j:i \in S_j} \tau_j \ell_j / \kappa_j = \sum_{j=1}^n \tau_j \ell_j$. Define α and β such that

$$\alpha \equiv \frac{\lambda_2(L)(d-1)}{\text{Tr}(L)} = \frac{\lambda_2(L)(d-1)}{\sum_{j=1}^n \tau_j \ell_j} \quad \text{and} \quad \beta \equiv \frac{\text{Tr}(L)}{dD_{\max}} = \frac{\sum_{j=1}^n \tau_j \ell_j}{dD_{\max}}. \quad (30)$$

For the proposed choice of $\lambda_{j,a} = 1/(\kappa_j - p_{j,a})$, we have $\tau_j = 1$ and the definitions of \mathcal{H} , L , α , and β reduce to those defined in Definition 1. We are left to prove an upper bound, $\delta \leq 32(\log(\ell_{\max} + 2))^2$, which implies the sufficient condition in (9) and finishes the proof of Theorem 2. We have,

$$\begin{aligned} \delta_{j,1} &\equiv \max_{a \in [\ell_j]} \left\{ \lambda_{j,a} (\kappa_j - p_{j,a}) \right\} + \sum_{a=1}^{\ell_j} \lambda_{j,a} = 1 + \sum_{a=1}^{\ell_j} \frac{1}{\kappa_j - p_{j,a}} \\ &\leq 1 + \sum_{a=1}^{\ell_j} \frac{1}{a} \\ &\leq 2 \log(\ell_j + 2), \end{aligned} \quad (31)$$

where in the first inequality follows from taking the worst case for the positions, i.e. $p_{j,a} = \kappa_j - \ell_j + a - 1$. Using the fact that for any integer x , $\sum_{a=0}^{x-1} 1/(x+a) \leq \log((x+\ell-1)/(x-1))$, we also have

$$\begin{aligned} \frac{\delta_{j,2}\kappa_j}{\eta_j \ell_j} &\leq \sum_{a=1}^{\ell_j} \frac{1}{\kappa_j - p_{j,a}} \frac{\max\{\ell_j, \kappa_j - p_{j,\ell_j}\}}{\ell_j} \\ &\leq \min \left\{ \log(\ell_j + 2), \log \left(\frac{\kappa_j - p_{j,\ell_j} + \ell_j - 1}{\kappa_j - p_{j,\ell_j} - 1} \right) \right\} \frac{\max\{\ell_j, \kappa_j - p_{j,\ell_j}\}}{\ell_j} \\ &\leq \frac{\log(\ell_j + 2)\ell_j}{\max\{\ell_j, \kappa_j - p_{j,\ell_j}\}} \frac{\ell_j}{\ell_j} \\ &\leq 2 \log(\ell_j + 2), \end{aligned} \quad (32)$$

where the first inequality follows from the definition of η_j , Equation (8). From (31), (32), and the fact that $\delta_{j,2} \leq \log(\ell_j + 2)$, we have

$$\delta = \max_{j \in [n]} \left\{ 4\delta_{j,1}^2 + \frac{2(\delta_{j,1}\delta_{j,2} + \delta_{j,2}^2)\kappa_j}{\eta_j \ell_j} \right\} \leq 28(\log(\ell_{\max} + 2))^2.$$

8.2 Proof of Theorem 8

We first introduce two key technical lemmas. In the following lemma we show that $\mathbb{E}_{\theta^*}[\nabla \mathcal{L}_{\text{RB}}(\theta^*)] = 0$ and provide a bound on the deviation of $\nabla \mathcal{L}_{\text{RB}}(\theta^*)$ from its mean. The expectation $\mathbb{E}_{\theta^*}[\cdot]$ is with respect to the randomness in the samples drawn according to θ^* . The log likelihood Equation (2) can be rewritten as

$$\mathcal{L}_{\text{RB}}(\theta) = \sum_{j=1}^n \sum_{a=1}^{\ell_j} \sum_{i < i' \in S_j} \mathbb{I}_{\{(i, i') \in G_{j,a}\}} \lambda_{j,a} \left(\theta_{i,i'} \mathbb{I}_{\{\sigma_j^{-1}(i) < \sigma_j^{-1}(i')\}} + \theta_{i',i} \mathbb{I}_{\{\sigma_j^{-1}(i) > \sigma_j^{-1}(i')\}} - \log(e^{\theta_i} + e^{\theta_{i'}}) \right). \quad (33)$$

We use $(i, i') \in G_{j,a}$ to mean either (i, i') or (i', i) belong to $E_{j,a}$. Taking the first-order partial derivative of $\mathcal{L}_{\text{RB}}(\theta)$, we get

$$\nabla_{i'} \mathcal{L}_{\text{RB}}(\theta^*) = \sum_{j:i \in S_j} \sum_{a=1}^{\ell_j} \sum_{i' \in S_j, i' \neq i} \lambda_{j,a} \mathbb{I}_{\{(i, i') \in G_{j,a}\}} \left(\mathbb{I}_{\{\sigma_j^{-1}(i) < \sigma_j^{-1}(i')\}} - \frac{\exp(\theta_{i'}^*)}{\exp(\theta_i^*) + \exp(\theta_{i'}^*)} \right). \quad (34)$$

Lemma 10. Under the hypotheses of Theorem 2, with probability at least $1 - 2e^3 d^{-3}$,

$$\|\nabla \mathcal{L}_{\text{RB}}(\theta^*)\|_2 \leq \sqrt{\frac{6 \log d \sum_{j=1}^n \sum_{a=1}^{\ell_j} (\lambda_{j,a})^2 (\kappa_j - p_{j,a}) (\kappa_j - p_{j,a} + 1)}{\theta_{i,i'}^2 d^3}}.$$

The Hessian matrix $H(\theta) \in \mathcal{S}^d$ with $H_{i,i'}(\theta) = \frac{\partial^2 \mathcal{L}_{\text{RB}}(\theta)}{\partial \theta_i \partial \theta_{i'}}$ is given by

$$H(\theta) = - \sum_{j=1}^n \sum_{a=1}^{\ell_j} \sum_{i < i' \in S_j} \mathbb{I}_{\{(i, i') \in G_{j,a}\}} \lambda_{j,a} \left((e_i - e_{i'}) (e_i - e_{i'})^\top \frac{\exp(\theta_i + \theta_{i'})}{[\exp(\theta_i) + \exp(\theta_{i'})]^2} \right). \quad (35)$$

It follows from the definition that $-H(\theta)$ is positive semi-definite for any $\theta \in \mathbb{R}^d$. The smallest eigenvalue of $-H(\theta)$ is equal to zero and the corresponding eigenvector is all-ones vector. The following lemma lower bounds its second smallest eigenvalue $\lambda_2(-H(\theta))$.

Lemma 11. *Under the hypotheses of Theorem 2, if*

$$\sum_{j=1}^n \sum_{a=1}^{\ell_j} \lambda_{j,a}(\kappa_j - p_{j,a}) \geq 2^6 e^{18b} \frac{\eta\delta}{\alpha^2 \beta^2 \tau} d \log d \quad (36)$$

then with probability at least $1 - d^{-3}$, the following holds for any $\theta \in \Omega_b$:

$$\lambda_2(-H(\theta)) \geq \frac{e^{-4b}}{(1 + e^{2b})^2} \frac{\alpha\gamma}{d-1} \sum_{j=1}^n \sum_{a=1}^{\ell_j} \lambda_{j,a}(\kappa_j - p_{j,a}). \quad (37)$$

Define $\Delta = \hat{\theta} - \theta^*$. It follows from the definition that Δ is orthogonal to the all-ones vector. By the definition of $\hat{\theta}$ as the optimal solution of the optimization (3), we know that $\mathcal{L}_{\text{RB}}(\hat{\theta}) \geq \mathcal{L}_{\text{RB}}(\theta^*)$ and thus

$$\mathcal{L}_{\text{RB}}(\hat{\theta}) - \mathcal{L}_{\text{RB}}(\theta^*) - \langle \nabla \mathcal{L}_{\text{RB}}(\theta^*), \Delta \rangle \geq -\langle \nabla \mathcal{L}_{\text{RB}}(\theta^*), \Delta \rangle \geq -\|\nabla \mathcal{L}_{\text{RB}}(\theta^*)\|_2 \|\Delta\|_2, \quad (38)$$

where the last inequality follows from the Cauchy-Schwartz inequality. By the mean value theorem, there exists a $\theta = a\hat{\theta} + (1-a)\theta^*$ for some $a \in [0, 1]$ such that $\theta \in \Omega_b$ and

$$\mathcal{L}_{\text{RB}}(\hat{\theta}) - \mathcal{L}_{\text{RB}}(\theta^*) - \langle \nabla \mathcal{L}_{\text{RB}}(\theta^*), \Delta \rangle = \frac{1}{2} \Delta^\top H(\theta) \Delta \leq -\frac{1}{2} \lambda_2(-H(\theta)) \|\Delta\|_2^2, \quad (39)$$

where the last inequality holds because the Hessian matrix $-H(\theta)$ is positive semi-definite with $H(\theta)\mathbf{1} = \mathbf{0}$ and $\Delta^\top \mathbf{1} = 0$. Combining (38) and (39),

$$\|\Delta\|_2 \leq \frac{2\|\nabla \mathcal{L}_{\text{RB}}(\theta^*)\|_2}{\lambda_2(-H(\theta))}. \quad (40)$$

Note that $\theta \in \Omega_b$ by definition. Theorem 8 follows by combining Equation (40) with Lemma 10 and Lemma 11.

8.2.1 PROOF OF LEMMA 10

The idea of the proof is to view $\nabla \mathcal{L}_{\text{RB}}(\theta^*)$ as the final value of a discrete time vector-valued martingale with values in \mathbb{R}^d . Define $\nabla \mathcal{L}_{G_{j,a}}(\theta^*)$ as the gradient vector arising out of each rank-breaking graph $\{G_{j,a}\}_{j \in [n], a \in [\ell_j]}$ that is

$$\nabla_i \mathcal{L}_{G_{j,a}}(\theta^*) \equiv \sum_{\substack{i \in S_j \\ i' \neq i}} \lambda_{j,a} \mathbb{I}_{\{(i,i') \in G_{j,a}\}} \left(\mathbb{I}_{\{\sigma_j^{-1}(i) < \sigma_j^{-1}(i')\}} - \frac{\exp(\theta_i^*)}{\exp(\theta_i^*) + \exp(\theta_{i'}^*)} \right). \quad (41)$$

Consider $\nabla \mathcal{L}_{G_{j,a}}(\theta^*)$ as the incremental random vector in a martingale of $\sum_{j=1}^n \ell_j$ time steps. Lemma 12 shows that the expectation of each incremental vector is zero. Observe that the conditioning event $\{i' < i : \sigma^{-1}(i') < p_{j,a}\}$ given in (43) is equivalent to conditioning on the history

$\{G_{j,a'}\}_{a' < a}$. Therefore, using the assumption that the rankings $\{\sigma_j\}_{j \in [n]}$ are mutually independent, we have that the conditional expectation of $\nabla \mathcal{L}_{G_{j,a}}(\theta^*)$ conditioned on $\{G_{j',a'}\}_{j' < j, a' \in [\ell_{j'}]}$ is zero. Further, the conditional expectation of $\nabla \mathcal{L}_{G_{j,a}}(\theta^*)$ is zero even when conditioned on the rank breaking due to previous separators $\{G_{j',a'}\}_{a' < a}$ that are ranked higher (i.e. $a' < a$), which follows from the next lemma.

Lemma 12. *For a position- p rank breaking graph G_p , defined over a set of items S , where $p \in [|S| - 1]$,*

$$\mathbb{P}[\sigma^{-1}(i) < \sigma^{-1}(i') \mid (i, i') \in G_p] = \frac{\exp(\theta_i^*)}{\exp(\theta_i^*) + \exp(\theta_{i'}^*)}, \quad (42)$$

for all $i, i' \in S$ and also

$$\mathbb{P}[\sigma^{-1}(i) < \sigma^{-1}(i') \mid (i, i') \in G_p \text{ and } \{i'' \in S : \sigma^{-1}(i'') < p\}] = \frac{\exp(\theta_i^*)}{\exp(\theta_i^*) + \exp(\theta_{i'}^*)}. \quad (43)$$

This is one of the key technical lemmas since it implies that the proposed rank-breaking is consistent, i.e. $\mathbb{E}_{\theta^*}[\nabla \mathcal{L}_{\text{RB}}(\theta^*)] = 0$. Throughout the proof of Theorem 2, this is the only place where the assumption on the proposed (consistent) rank-breaking is used. According to a companion theorem in Azari Souhaimi et al. (2014, Theorem 2), it also follows that any rank-breaking that is not union of position- p rank-breakings results in inconsistency, i.e. $\mathbb{E}_{\theta^*}[\nabla \mathcal{L}_{\text{RB}}(\theta^*)] \neq 0$. We claim that for each rank-breaking graph $G_{j,a}$, $\|\nabla \mathcal{L}_{G_{j,a}}(\theta^*)\|_2^2 \leq (\lambda_{j,a})^2 (\kappa_j - p_{j,a}) (\kappa_j - p_{j,a} + 1)$. By Lemma 13 which is a generalization of the Azuma-Hoeffding inequality found in (Hayes, 2005, Theorem 1.8), we have

$$\mathbb{P}[\|\nabla \mathcal{L}_{\text{RB}}(\theta^*)\|_2 \geq \delta] \leq 2e^3 \exp\left(-\frac{\delta^2}{2 \sum_{j=1}^n \sum_{a=1}^{\ell_j} (\lambda_{j,a})^2 (\kappa_j - p_{j,a}) (\kappa_j - p_{j,a} + 1)}\right),$$

which implies the result.

Lemma 13. *Let (X_1, X_2, \dots, X_n) be real-valued martingale taking values in \mathbb{R}^d such that $X_0 = 0$ and for every $1 \leq i \leq n$, $\|X_i - X_{i-1}\|_2 \leq c_i$, for some non-negative constant c_i . Then for every $\delta > 0$,*

$$\mathbb{P}[\|X_n\|_2 \geq \delta] \leq 2e^3 e^{-\frac{\delta^2}{2 \sum_{i=1}^n c_i^2}}. \quad (44)$$

It follows from the upper bound on $\|\nabla \mathcal{L}_{G_{j,a}}(\theta^*)\|_2^2 \leq c_i^2$ with $c_i^2 = \lambda^2 (\kappa_j - p_{j,a})^2 + (\kappa_j - p_{j,a})$. In the expression (41), $\nabla \mathcal{L}_{G_{j,a}}(\theta^*)$ has one entry at $p_{j,a}$ -th position that is compared to $(\kappa_j - p_{j,a})$ other items and $(\kappa_j - p_{j,a})$ entries that is compared only once, giving the bound

$$\|\nabla \mathcal{L}_{G_{j,a}}(\theta^*)\|_2^2 \leq \lambda_{j,a}^2 (\kappa_j - p_{j,a})^2 + \lambda_{j,a}^2 (\kappa_j - p_{j,a}).$$

8.2.2 PROOF OF LEMMA 12

Define event $E \equiv \{(i, i') \in G_p\}$. Observe that

$$E = \left\{ \left(\mathbb{I}_{\{\sigma^{-1}(i)=p\}} + \mathbb{I}_{\{\sigma^{-1}(i')=p\}} = 1 \right) \wedge \left(\sigma^{-1}(i), \sigma^{-1}(i') \geq p \right) \right\}.$$

Consider any set $\Omega \subset S \setminus \{i, i'\}$ such that $|\Omega| = p - 1$. Let M denote an event that items of the set Ω are ranked in top- $(p - 1)$ positions in a particular order. It is easy to verify the following:

$$\begin{aligned} \mathbb{P}\left[\sigma^{-1}(i) < \sigma^{-1}(i') \mid E, M\right] &= \frac{\mathbb{P}\left[\left(\sigma^{-1}(i) < \sigma^{-1}(i')\right), E, M\right]}{\mathbb{P}\left[E, M\right]} \\ &= \frac{\mathbb{P}\left[\left(\sigma^{-1}(i) = p\right), M\right]}{\mathbb{P}\left[\left(\sigma^{-1}(i) = p\right), M\right] + \mathbb{P}\left[\left(\sigma^{-1}(i') = p\right), M\right]} \\ &= \frac{\exp(\theta_i^*)}{\exp(\theta_i^*) + \exp(\theta_{i'}^*)} = \mathbb{P}\left[\sigma^{-1}(i) < \sigma^{-1}(i')\right]. \end{aligned}$$

Since M is any particular ordering of the set Ω and Ω is any subset of $S \setminus \{i, i'\}$ such that $|\Omega| = p - 1$, conditioned on event E probabilities of all the possible events M over all the possible choices of set Ω sum to 1.

8.2.3 PROOF OF LEMMA 13

It follows exactly along the lines of proof of Theorem 1.8 in (Hayes, 2005).

8.2.4 PROOF OF LEMMA 11

The Hessian $H(\theta)$ is given in (35). For all $j \in [n]$, define $M^{(j)} \in \mathcal{S}^d$ as

$$M^{(j)} \equiv \sum_{a=1}^{\ell_j} \lambda_{j,a} \sum_{i < i' \in S_j} \mathbb{I}_{\{(i, i') \in G_{j,a}\}} (e_i - e_{i'})(e_i - e_{i'})^\top, \quad (45)$$

and let $M \equiv \sum_{j=1}^n M^{(j)}$. Observe that M is positive semi-definite and the smallest eigenvalue of M is zero with the corresponding eigenvector given by the all-ones vector. If $|\theta_i| \leq b$, for all $i \in [d]$, $\frac{\exp(\theta_i + \theta_{i'})}{\exp(\theta_i) + \exp(\theta_{i'})} \geq \frac{e^{2b}}{(1 + e^{2b})^2}$. Recall the definition of $H(\theta)$ from Equation (35). It follows that $-H(\theta) \succeq \frac{e^{2b}}{(1 + e^{2b})^2} M$ for $\theta \in \Omega_b$. Since, $-H(\theta)$ and M are symmetric matrices, from Weyl's inequality we have, $\lambda_2(-H(\theta)) \geq \frac{e^{2b}}{(1 + e^{2b})^2} \lambda_2(M)$. Again from Weyl's inequality, it follows that

$$\lambda_2(M) \geq \lambda_2(\mathbb{E}[M]) - \|M - \mathbb{E}[M]\|, \quad (46)$$

where $\|\cdot\|$ denotes the spectral norm. We will show in (51) that $\lambda_2(\mathbb{E}[M]) \geq 2\gamma e^{-6b} (\alpha/(d - 1)) \sum_{j=1}^n \tau_j \ell_j$, and in (63) that $\|M - \mathbb{E}[M]\| \leq 8e^{3b} \sqrt{\frac{\eta^b \log d}{\beta \pi d}} \sum_{j=1}^n \tau_j \ell_j$.

$$\lambda_2(M) \geq \frac{2e^{-6b} \alpha \gamma}{d - 1} \sum_{j=1}^n \tau_j \ell_j - 8e^{3b} \sqrt{\frac{\eta^b \log d}{\beta \pi d}} \sum_{j=1}^n \tau_j \ell_j \geq \frac{e^{-6b} \alpha \gamma}{d - 1} \sum_{j=1}^n \tau_j \ell_j, \quad (47)$$

where the last inequality follows from the assumption that $\sum_{j=1}^n \tau_j \ell_j \geq 2^6 e^{18b} \frac{\eta^b}{\alpha^2 \beta \gamma \pi} d \log d$. This proves the desired claim.

To prove the lower bound on $\lambda_2(\mathbb{E}[M])$, notice that

$$\mathbb{E}[M] = \sum_{j=1}^n \sum_{a=1}^{\ell_j} \lambda_{j,a} \sum_{i < i' \in S_j} \mathbb{P}\left[(i, i') \in G_{j,a} \mid (i, i') \in S_j\right] (e_i - e_{i'})(e_i - e_{i'})^\top. \quad (48)$$

The following lemma provides a lower bound on $\mathbb{P}[(i, i') \in G_{j,a} \mid (i, i') \in S_j]$.

Lemma 14. Consider a ranking σ over a set $S \subseteq [d]$ such that $|S| = \kappa$. For any two items $i, i' \in S$, $\theta \in \Omega_b$, and $1 \leq \ell \leq \kappa - 1$,

$$\mathbb{P}_\theta \left[\sigma^{-1}(i) = \ell, \sigma^{-1}(i') > \ell \right] \geq \frac{e^{-6b} (\kappa - \ell)}{\kappa (\kappa - 1)} \left(1 - \frac{\ell}{\kappa} \right)^{\alpha_{i, i', \ell, \theta} - 2}, \quad (49)$$

where the probability \mathbb{P}_θ is with respect to the sampled ranking resulting from PL weights $\theta \in \Omega_b$, and $\alpha_{i, i', \ell, \theta}$ is defined as $1 \leq \alpha_{i, i', \ell, \theta} = \lceil \tilde{\alpha}_{i, i', \ell, \theta} \rceil$, and $\tilde{\alpha}_{i, i', \ell, \theta}$ is,

$$\tilde{\alpha}_{i, i', \ell, \theta} \equiv \max_{\ell' \in [d] \setminus \Omega \setminus \{i, i'\}} \max_{\substack{|\Omega| = \kappa - \ell' \\ \left\{ \frac{\exp(\theta_i) + \exp(\theta_{i'})}{\sum_{j \in \Omega} \exp(\theta_j)} \right\} / |\Omega|}}. \quad (50)$$

Note that we do not need $\max_{\ell' \in [d]}$ in the above equation as the expression achieves its maxima at $\ell' = \ell$, but we keep the definition to avoid any confusion. In the worst case, $2e^{-2b} \leq \tilde{\alpha}_{i, i', \ell, \theta} \leq 2e^{2b}$. Therefore, using definition of rank breaking graph $G_{j,a}$, and Equations (48) and (49) we have,

$$\begin{aligned} \mathbb{E}[M] &\succeq \gamma e^{-6b} \sum_{j=1}^n \sum_{a=1}^{\ell_j} \lambda_{j,a} \frac{2(\kappa_j - p_{j,a})}{\kappa_j (\kappa_j - 1)} \sum_{i < i' \in S_j} (e_i - e_{i'}) (e_i - e_{i'})^\top \\ &\succeq 2\gamma e^{-6b} \sum_{j=1}^n \sum_{a=1}^{\ell_j} \frac{1}{\kappa_j (\kappa_j - 1)} \lambda_{j,a} (\kappa_j - p_{j,a}) \sum_{i < i' \in S_j} (e_i - e_{i'}) (e_i - e_{i'})^\top \\ &= 2\gamma e^{-6b} L, \end{aligned} \quad (51)$$

where we used $\gamma \leq (1 - p_{i, \ell_j} / \kappa_j)^{\alpha_{i, i', \ell, \theta} - 2}$ which follows for the definition in (7). (51) follows from the definition of Laplacian L , defined for the comparison graph \mathcal{H} in Definition 9. Using $\lambda_2(L) = (\alpha/(d - 1)) \sum_{j=1}^n \tau_j \ell_j$ from (30), we get the desired bound $\lambda_2(\mathbb{E}[M]) \geq 2\gamma e^{-6b} (\alpha/(d - 1)) \sum_{j=1}^n \tau_j \ell_j$.

Next we need to upper bound $\|\sum_{j=1}^n \mathbb{E}[M^{(j)}]\|$ to bound the deviation of M from its expectation. To this end, we prove an upper bound on $\mathbb{P}[\sigma_j^{-1}(i) = p_{j,a} \mid i \in S_j]$ in the following lemma.

Lemma 15. Under the hypotheses of Lemma 14,

$$\mathbb{P}_\theta \left[\sigma^{-1}(i) = \ell \right] \leq \frac{e^{6b}}{\kappa} \left(1 - \frac{\ell}{\kappa + \alpha_{i, \ell, \theta}} \right)^{\alpha_{i, \ell, \theta} - 1} \leq \frac{e^{6b}}{\kappa - \ell}, \quad (52)$$

where $0 \leq \alpha_{i, \ell, \theta} = \lceil \tilde{\alpha}_{i, \ell, \theta} \rceil$, and $\tilde{\alpha}_{i, \ell, \theta}$ is,

$$\tilde{\alpha}_{i, \ell, \theta} \equiv \min_{\ell' \in [d] \setminus \Omega \setminus \{i\}} \min_{\substack{|\Omega| = \kappa - \ell' + 1 \\ \left\{ \frac{\exp(\theta_i)}{\sum_{j \in \Omega} \exp(\theta_j)} \right\} / |\Omega|}}. \quad (53)$$

In the worst case, $e^{-2b} \leq \tilde{\alpha}_{i, \ell, \theta} \leq e^{2b}$. Note that $\alpha_{i, \ell, \theta} = 0$ gives the worst upper bound.

Therefore using Equation (52), for all $i \in [d]$, we have,

$$\mathbb{P}[\sigma_j^{-1}(i) \in \mathcal{P}_j] \leq \min \left\{ 1, \frac{e^{6b}\ell_j}{\kappa_j - p_{j,\ell_j}} \right\} \leq \frac{e^{6b}\ell_j}{\max\{\ell_j, \kappa_j - p_{j,\ell_j}\}} \leq \frac{e^{6b}\eta\ell_j}{\kappa_j}, \quad (54)$$

where we used η defined in Equation (8). Define a diagonal matrix $D^{(i)} \in \mathcal{S}^d$ and a matrix $A^{(i)} \in \mathcal{S}^d$,

$$A_{i\ell_j}^{(i)} \equiv \mathbb{I}_{\{i, i' \in \mathcal{S}_j\}} \sum_{a=1}^{\ell_j} \lambda_{j,a} \mathbb{I}_{\{(i, i') \in G_{j,a}\}}, \quad \text{for all } i, i' \in [d], \quad (55)$$

and $D_{i\ell_j}^{(i)} = \sum_{i' \neq i} A_{i\ell_j}^{(i')}$. Observe that $M^{(i)} = D^{(i)} - A^{(i)}$. For all $i \in [d]$, we have,

$$\begin{aligned} D_{i\ell_j}^{(i)} &= \mathbb{I}_{\{i \in \mathcal{S}_j\}} \sum_{i'=1}^{\kappa_j} \mathbb{I}_{\{\sigma_j^{-1}(i) = i'\}} \sum_{a=1}^{\ell_j} \lambda_{j,a} \deg_{G_{j,a}}(\sigma_j^{-1}(i')) \\ &\leq \mathbb{I}_{\{i \in \mathcal{S}_j\}} \left\{ \mathbb{I}_{\{\sigma_j^{-1}(i) \in \mathcal{P}_j\}} \left(\max_{a \in [k_j]} \left\{ \lambda_{j,a} (\kappa_j - p_{j,a}) \right\} + \sum_{a=1}^{\ell_j} \lambda_{j,a} \right) + \mathbb{I}_{\{\sigma_j^{-1}(i) \notin \mathcal{P}_j\}} \left(\sum_{a=1}^{\ell_j} \lambda_{j,a} \right) \right\} \\ &= \mathbb{I}_{\{i \in \mathcal{S}_j\}} \left\{ \mathbb{I}_{\{\sigma_j^{-1}(i) \in \mathcal{P}_j\}} \delta_{j,1} + \mathbb{I}_{\{\sigma_j^{-1}(i) \notin \mathcal{P}_j\}} \delta_{j,2} \right\}, \end{aligned} \quad (56)$$

where the last equality follows from the definition of $\delta_{j,1}$ and $\delta_{j,2}$ in Equation (26). Note that $\max_{i \in [d]} |D_{i\ell_j}| = \delta_{j,1}$. Using (54) and (56), we have,

$$\mathbb{E} \left[D_{i\ell_j}^{(i)} \right] \leq \mathbb{I}_{\{i \in \mathcal{S}_j\}} \left\{ \frac{e^{6b}\eta\ell_j}{\kappa_j} \left(\delta_{j,1} + \frac{\delta_{j,2}\kappa_j}{\eta\ell_j} \right) \right\}. \quad (57)$$

Similarly we have,

$$\mathbb{E} \left[(D_{i\ell_j}^{(i)})^2 \right] \leq \mathbb{I}_{\{i \in \mathcal{S}_j\}} \left\{ \frac{e^{6b}\eta\ell_j}{\kappa_j} \left(\delta_{j,1}^2 + \frac{\delta_{j,2}^2\kappa_j}{\eta\ell_j} \right) \right\} \quad (58)$$

For all $i \in [d]$, we have,

$$\begin{aligned} \mathbb{E} \left[\sum_{i'=1}^d ((A^{(i)})^2)_{i\ell_j} \right] &\leq \mathbb{E} \left[\left(\sum_{i'=1}^d A_{i\ell_j}^{(i')} \right) \max_{i \in [d]} \left\{ \sum_{i'=1}^d A_{i\ell_j}^{(i')} \right\} \right] \\ &\leq \mathbb{E} \left[D_{i\ell_j}^{(i)} \delta_{j,1} \right] \\ &\leq \mathbb{I}_{\{i \in \mathcal{S}_j\}} \left\{ \frac{e^{6b}\eta\ell_j}{\kappa_j} \left(\delta_{j,1}^2 + \frac{\delta_{j,1}\delta_{j,2}\kappa_j}{\eta\ell_j} \right) \right\}. \end{aligned} \quad (59)$$

Using (58) and (59), we have, for all $i \in [d]$,

$$\begin{aligned} &\sum_{i'=1}^d \left| \mathbb{E} \left[(M^{(i)})^2 \right]_{i\ell_j} \right| \\ &= \sum_{i'=1}^d \left| \mathbb{E} \left[(D^{(i)})^2 \right]_{i\ell_j} - \mathbb{E} \left[D^{(i)} A^{(i)} \right]_{i\ell_j} - \mathbb{E} \left[(A^{(i)} D^{(i)}) \right]_{i\ell_j} + \mathbb{E} \left[(A^{(i)})^2 \right]_{i\ell_j} \right| \\ &\leq 2\mathbb{E} \left[(D_{i\ell_j}^{(i)})^2 \right] + \sum_{i'=1}^d \left(\mathbb{E} \left[\delta_{j,1} (A^{(i)}) \right]_{i\ell_j} + \mathbb{E} \left[(A^{(i)})^2 \right]_{i\ell_j} \right) \\ &\leq \mathbb{I}_{\{i \in \mathcal{S}_j\}} \left\{ \frac{e^{6b}\eta\ell_j}{\kappa_j} \left(4\delta_{j,1}^2 + \frac{2(\delta_{j,1}\delta_{j,2} + \delta_{j,2}^2)\kappa_j}{\eta\ell_j} \right) \right\} \\ &= \mathbb{I}_{\{i \in \mathcal{S}_j\}} \left\{ \frac{e^{6b}\delta\eta\ell_j}{\kappa_j} \right\}, \end{aligned} \quad (60)$$

where the last equality follows from the definition of δ , Equation (27).

To bound $\|\sum_{j=1}^n \mathbb{E}[(M^{(j)})^2]\|$, we use the fact that for $J \in \mathbb{R}^{\delta \times d}$, $\|J\| \leq \max_{i \in [d]} \sum_{i'=1}^d |J_{i\ell_j}^i|$. Therefore, we have

$$\begin{aligned} \left\| \sum_{j=1}^n \mathbb{E} \left[(M^{(j)})^2 \right] \right\| &\leq e^{6b}\delta\eta \max_{i \in [d]} \left\{ \sum_{j: i \in \mathcal{S}_j} \frac{\ell_j}{\kappa_j} \right\} \\ &= \frac{e^{6b}\eta\delta}{\tau} D_{\max} \\ &= \frac{e^{6b}\eta\delta}{\beta\tau d} \sum_{j=1}^n \tau_j \ell_j, \end{aligned} \quad (61)$$

$$= \frac{e^{6b}\eta\delta}{\beta\tau d} \sum_{j=1}^n \tau_j \ell_j, \quad (62)$$

where (61) follows from the definition of D_{\max} in Equation(28) and (62) follows from the definition of β in (30). Observe that from Equation (56), $\|M^{(i)}\| \leq 2\delta_{j,1} \leq 2\sqrt{\delta}$. Applying matrix Bernstein inequality, we have,

$$\mathbb{P} \left[\|M - \mathbb{E}[M]\| \geq t \right] \leq d \exp \left(\frac{-t^2/2}{\frac{e^{6b}\eta\delta}{\beta\tau d} \sum_{j=1}^n \tau_j \ell_j + 4\sqrt{\delta t}/3} \right).$$

Therefore, with probability at least $1 - d^{-3}$, we have,

$$\|M - \mathbb{E}[M]\| \leq 4e^{3b} \sqrt{\frac{\eta\delta \log d}{\beta\tau d} \sum_{j=1}^n \tau_j \ell_j} + \frac{64\sqrt{\delta} \log d}{3} \leq 8e^{3b} \sqrt{\frac{\eta\delta \log d}{\beta\tau d} \sum_{j=1}^n \tau_j \ell_j}, \quad (63)$$

where the second inequality uses $\sum_{j=1}^n \tau_j \ell_j \geq 2^6(\beta\tau/\eta)d \log d$ which follows from the assumption that $\sum_{j=1}^n \tau_j \ell_j \geq 2^6 e^{18b} \frac{\eta\delta}{\beta\tau\alpha\beta} d \log d$ and the fact that $\alpha, \beta \leq 1, \gamma \leq 1, \eta \geq 1$, and $\delta > \tau^2$.

8.2.5 PROOF OF LEMMA 14

Since providing a lower bound on $\mathbb{P}_\theta[\sigma^{-1}(i) = \ell, \sigma^{-1}(i') > \ell]$ for arbitrary θ is challenging, we construct a new set of parameters $\{\tilde{\theta}_j\}_{j \in [d]}$ from the original θ . These new parameters are constructed such that it is both easy to compute the probability and also provides a lower bound on the original distribution. We denote the sum of the weights by $W \equiv \sum_{j \in S} \exp(\theta_j)$. We define a new set of parameters $\{\tilde{\theta}_j\}_{j \in S}$:

$$\tilde{\theta}_j = \begin{cases} \log(\tilde{\alpha}_{i,i',\ell,\theta}/2) & \text{for } j = i \text{ or } i', \\ 0 & \text{otherwise.} \end{cases} \quad (64)$$

Similarly define $\tilde{W} \equiv \sum_{j \in S} \exp(\tilde{\theta}_j) = \kappa - 2 + \tilde{\alpha}_{i,i',\ell,\theta}$. We have,

$$\begin{aligned} & \mathbb{P}_\theta[\sigma^{-1}(i) = \ell, \sigma^{-1}(i') > \ell] \\ &= \sum_{\substack{j_1 \in S \\ j_1 \neq i, i'}} \left(\frac{\exp(\theta_{j_1})}{W} \sum_{\substack{j_2 \in S \\ j_2 \neq i, i', j_1}} \left(\frac{\exp(\theta_{j_2})}{W - \exp(\theta_{j_1})} \dots \right. \right. \\ & \quad \left. \left. \left(\sum_{\substack{j_{l-1} \in S \\ j_{l-1} \neq i, i', \\ j_1, \dots, j_{l-2}}} \frac{\exp(\theta_{j_{l-1}})}{W - \sum_{k=j_1}^{j_{l-2}} \exp(\theta_k)} \frac{\exp(\theta_{j_l})}{W - \sum_{k=j_1}^{j_{l-1}} \exp(\theta_k)} \dots \right) \right) \right) \\ &= \frac{\exp(\theta_i)}{W} \sum_{\substack{j_1 \in S \\ j_1 \neq i, i'}} \left(\frac{\exp(\theta_{j_1})}{W - \exp(\theta_{j_1})} \sum_{\substack{j_2 \in S \\ j_2 \neq i, i', j_1}} \left(\frac{\exp(\theta_{j_2})}{W - \exp(\theta_{j_1}) - \exp(\theta_{j_2})} \dots \right. \right. \\ & \quad \left. \left. \sum_{\substack{j_{l-1} \in S \\ j_{l-1} \neq i, i', \\ j_1, \dots, j_{l-2}}} \left(\frac{\exp(\theta_{j_{l-1}})}{W - \sum_{k=j_1}^{j_{l-1}} \exp(\theta_k)} \dots \right) \right) \right) \end{aligned} \quad (65)$$

Consider the last summation term in the above equation and let $\Omega_\ell = S \setminus \{i, i', j_1, \dots, j_{l-2}\}$. Observe that, $|\Omega_\ell| = \kappa - \ell$ and from equation (50), $\frac{\exp(\theta_i) + \exp(\theta_{i'})}{\sum_{j \in \Omega_\ell} \exp(\theta_j)} \leq \frac{\tilde{\alpha}_{i,i',\ell,\theta}}{\kappa - \ell}$. We have,

$$\begin{aligned} & \sum_{j_{l-1} \in \Omega_\ell} \frac{\exp(\theta_{j_{l-1}})}{W - \sum_{k=j_1}^{j_{l-1}} \exp(\theta_k)} \\ &= \sum_{j_{l-1} \in \Omega_\ell} \frac{\exp(\theta_{j_{l-1}})}{W - \sum_{k=j_1}^{j_{l-2}} \exp(\theta_k) - \exp(\theta_{j_{l-1}})} \\ &\geq \frac{\sum_{j_{l-1} \in \Omega_\ell} \exp(\theta_{j_{l-1}})}{W - \sum_{k=j_1}^{j_{l-2}} \exp(\theta_k) - (\sum_{j_{l-1} \in \Omega_\ell} \exp(\theta_{j_{l-1}})) / |\Omega_\ell|} \end{aligned} \quad (66)$$

$$\begin{aligned} &= \frac{\sum_{j_{l-1} \in \Omega_\ell} \exp(\theta_{j_{l-1}})}{\exp(\theta_i) + \exp(\theta_{i'}) + \sum_{j_{l-1} \in \Omega_\ell} \exp(\theta_{j_{l-1}}) - (\sum_{j_{l-1} \in \Omega_\ell} \exp(\theta_{j_{l-1}})) / |\Omega_\ell|} \\ &= \left(\frac{\exp(\theta_i) + \exp(\theta_{i'})}{\sum_{j_{l-1} \in \Omega_\ell} \exp(\theta_{j_{l-1}})} + 1 - \frac{1}{\kappa - \ell} \right)^{-1} \\ &\geq \left(\frac{\tilde{\alpha}_1}{\kappa - \ell} + 1 - \frac{1}{\kappa - \ell} \right)^{-1} \end{aligned} \quad (67)$$

$$\begin{aligned} &= \frac{\kappa - \ell}{\tilde{\alpha}_1 + \kappa - \ell - 1} \\ &= \sum_{j_{l-1} \in \Omega_\ell} \frac{\exp(\tilde{\theta}_{j_{l-1}})}{\tilde{W} - \sum_{k=j_1}^{j_{l-2}} \exp(\tilde{\theta}_k) - \exp(\tilde{\theta}_{j_{l-1}})}, \end{aligned} \quad (68)$$

where (66) follows from the Jensen's inequality and the fact that for any $c > 0, 0 < x < c, \frac{x}{c-x}$ is convex in x . Equation (67) follows from the definition of $\tilde{\alpha}_{i,i',\ell,\theta}$, (50), and the fact that $|\Omega_\ell| = \kappa - \ell$. Equation (68) uses the definition of $\{\tilde{\theta}_j\}_{j \in S}$.

Consider $\{\Omega_\ell\}_{2 \leq \ell \leq l-1}$, $|\Omega_\ell| = \kappa - \ell$, corresponding to the subsequent summation terms in (65). Observe that $\frac{\exp(\theta_i) + \exp(\theta_{i'})}{\sum_{j \in \Omega_\ell} \exp(\theta_j)} \leq \tilde{\alpha}_{i,i',\ell,\theta} / |\Omega_\ell|$. Therefore, each summation term in equation (65) can

be lower bounded by the corresponding term where $\{\theta_j\}_{j \in S}$ is replaced by $\{\tilde{\theta}_j\}_{j \in S}$. Hence, we have

$$\begin{aligned}
& \mathbb{P}_\theta \left[\sigma^{-1}(i) = \ell_i, \sigma^{-1}(i') > \ell \right] \\
& \geq \frac{\exp(\theta_i)}{W} \sum_{\substack{j_1 \in S \\ j_1 \neq i, i'}} \left(\frac{\exp(\tilde{\theta}_{j_1})}{W - \exp(\tilde{\theta}_{j_1})} \sum_{\substack{j_2 \in S \\ j_2 \neq i, i', j_1}} \left(\frac{\exp(\tilde{\theta}_{j_2})}{W - \exp(\tilde{\theta}_{j_1}) - \exp(\tilde{\theta}_{j_2})} \dots \right. \right. \\
& \quad \left. \left. \sum_{\substack{j_{l-1} \in S \\ j_{l-1} \neq i, i', \\ j_1, \dots, j_{l-2}}} \left(\frac{\exp(\tilde{\theta}_{j_{l-1}})}{W - \sum_{k=1}^{j_{l-1}} \exp(\tilde{\theta}_k)} \right) \right) \right) \\
& \geq \frac{e^{-4b} \exp(\tilde{\theta}_i)}{\tilde{W}} \sum_{\substack{j_1 \in S \\ j_1 \neq i, i'}} \left(\frac{\exp(\tilde{\theta}_{j_1})}{W - \exp(\tilde{\theta}_{j_1})} \sum_{\substack{j_2 \in S \\ j_2 \neq i, i', j_1}} \left(\frac{\exp(\tilde{\theta}_{j_2})}{W - \exp(\tilde{\theta}_{j_1}) - \exp(\tilde{\theta}_{j_2})} \dots \right. \right. \\
& \quad \left. \left. \sum_{\substack{j_{l-1} \in S \\ j_{l-1} \neq i, i', \\ j_1, \dots, j_{l-2}}} \left(\frac{\exp(\tilde{\theta}_{j_{l-1}})}{W - \sum_{k=1}^{j_{l-1}} \exp(\tilde{\theta}_k)} \right) \right) \right) \\
& = (e^{-4b}) \mathbb{P}_{\tilde{\theta}} \left[\sigma^{-1}(i) = \ell_i, \sigma^{-1}(i') > \ell \right]. \tag{69}
\end{aligned}$$

The second inequality uses $\frac{\exp(\tilde{\theta}_i)}{W} \geq e^{-2b}/\kappa$ and $\frac{\exp(\tilde{\theta}_i)}{W} \leq e^{2b}/\kappa$. Observe that $\exp(\tilde{\theta}_j) = 1$ for all $j \neq i, i'$ and $\exp(\tilde{\theta}_i) + \exp(\tilde{\theta}_{i'}) = \tilde{\alpha}_{i, i', \ell, \theta} \leq [\tilde{\alpha}_{i, i', \ell, \theta}] = \alpha_{i, i', \ell, \theta} \geq 1$. Therefore, we have

$$\begin{aligned}
& \mathbb{P}_\theta \left[\sigma^{-1}(i) = \ell_i, \sigma^{-1}(i') > \ell \right] \\
& = \frac{\binom{\kappa-2}{\ell-1}}{(\kappa-2)!} \frac{(\tilde{\alpha}_{i, i', \ell, \theta}/2)(\ell-1)!}{(\kappa-2 + \tilde{\alpha}_{i, i', \ell, \theta})(\kappa-2 + \alpha_{i, i', \ell, \theta} - 1) \dots (\kappa-2 + \tilde{\alpha}_{i, i', \ell, \theta} - (\ell-1))} \\
& \geq \frac{(\kappa-2)!}{(\kappa-2)!} \frac{(\tilde{\alpha}_{i, i', \ell, \theta}/2)(\ell-1)!}{e^{-2b}} \\
& = \frac{(\kappa-\ell-1)! (\kappa + \alpha_{i, i', \ell, \theta} - 2)(\kappa + \alpha_{i, i', \ell, \theta} - 3) \dots (\kappa + \alpha_{i, i', \ell, \theta} - (\ell+1))}{e^{-2b} (\kappa - \ell + \alpha_{i, i', \ell, \theta} - 2)(\kappa - \ell + \alpha_{i, i', \ell, \theta} - 3) \dots (\kappa - \ell)} \\
& = \frac{(\kappa + \alpha_{i, i', \ell, \theta} - 2)(\kappa + \alpha_{i, i', \ell, \theta} - 3) \dots (\kappa - 1)}{(\kappa - 1) (\kappa - \ell + \alpha_{i, i', \ell, \theta} - 2)(\kappa - \ell + \alpha_{i, i', \ell, \theta} - 3) \dots (\kappa - \ell)} \\
& \geq \frac{e^{-2b}}{\binom{\kappa-1}{\ell-1}} \left(\frac{1 - \frac{\ell}{\kappa}}{\alpha_{i, i', \ell, \theta} - 1} \right) \\
& = \frac{e^{-2b} (\kappa - \ell) \binom{\ell}{\kappa}}{\kappa (\kappa - 1) \binom{\ell}{\kappa}} \alpha_{i, i', \ell, \theta}^{-2}, \tag{71}
\end{aligned}$$

where (70) follows from the fact that $\tilde{\alpha}_{i, i', \ell, \theta} \geq 2e^{-2b}$. Claim (49) follows by combining Equations (69) and (71).

8.2.6 PROOF OF LEMMA 15

Analogous to the proof of Lemma 14, we construct a new set of parameters $\{\tilde{\theta}_j\}_{j \in [d]}$ from the original θ . We denote the sum of the weights by $W \equiv \sum_{j \in S} \exp(\theta_j)$. We define a new set of parameters $\{\tilde{\theta}_j\}_{j \in S}$:

$$\tilde{\theta}_j = \begin{cases} \log(\tilde{\alpha}_{i, \ell, \theta}) & \text{for } j = i, \\ 0 & \text{otherwise.} \end{cases} \tag{72}$$

Similarly define $\tilde{W} \equiv \sum_{j \in S} \exp(\tilde{\theta}_j) = \kappa - 1 + \tilde{\alpha}_{i, \ell, \theta}$. We have,

$$\begin{aligned}
& \mathbb{P}_\theta \left[\sigma^{-1}(i) = \ell \right] \\
& = \sum_{\substack{j_1 \in S \\ j_1 \neq i}} \left(\frac{\exp(\theta_{j_1})}{W} \sum_{\substack{j_2 \in S \\ j_2 \neq i, j_1}} \left(\frac{\exp(\theta_{j_2})}{W - \exp(\theta_{j_1})} \dots \left(\sum_{\substack{j_{l-1} \in S \\ j_{l-1} \neq i, \\ j_1, \dots, j_{l-2}}} \frac{\exp(\theta_{j_{l-1}})}{W - \sum_{k=1}^{j_{l-1}} \exp(\theta_k)} \right) \right) \right) \\
& \leq \sum_{\substack{j_1 \in S \\ j_1 \neq i}} \left(\frac{\exp(\theta_{j_1})}{W} \sum_{\substack{j_2 \in S \\ j_2 \neq i, j_1}} \left(\frac{\exp(\theta_{j_2})}{W - \exp(\theta_{j_1})} \dots \left(\sum_{\substack{j_{l-1} \in S \\ j_{l-1} \neq i, \\ j_1, \dots, j_{l-2}}} \frac{\exp(\theta_{j_{l-1}})}{W - \sum_{k=1}^{j_{l-1}} \exp(\theta_k)} \right) \right) \right) \frac{e^{2b}}{\kappa - \ell + 1} \tag{73}
\end{aligned}$$

Consider the last summation term in the equation (73), and let $\Omega_\ell = S \setminus \{i, j_1, \dots, j_{l-2}\}$, such that $|\Omega_\ell| = \kappa - \ell + 1$. Observe that from equation (53), $\sum_{j \in \Omega_\ell} \frac{\exp(\theta_j)}{\exp(\theta_j)} \geq \kappa - \ell + 1$. We have,

$$\begin{aligned}
& \sum_{j_{l-1} \in \Omega_\ell} \frac{\exp(\theta_{j_{l-1}})}{W - \sum_{k=1}^{j_{l-2}} \exp(\theta_k)} = \frac{\sum_{j_{l-1} \in \Omega_\ell} \exp(\theta_{j_{l-1}})}{\exp(\theta_i) + \sum_{j_{l-1} \in \Omega_\ell} \exp(\theta_{j_{l-1}})} \\
& \leq \left(\frac{\tilde{\alpha}_{i, \ell, \theta}}{\kappa - \ell + 1} + 1 \right)^{-1} \\
& = \frac{\tilde{\alpha}_{i, \ell, \theta} + \kappa - \ell + 1}{\kappa - \ell + 1} \\
& = \sum_{j_{l-1} \in \Omega_\ell} \frac{\exp(\tilde{\theta}_{j_{l-1}})}{W - \sum_{k=1}^{j_{l-2}} \exp(\tilde{\theta}_k)}, \tag{74}
\end{aligned}$$

where (74) follows from the definition of $\{\tilde{\theta}_j\}_{j \in S}$.

Consider $\{\Omega_j\}_{2 \leq j \leq l-1}$, $|\Omega_j| = \kappa - \tilde{\ell} + 1$, corresponding to the subsequent summation terms in (73). Observe that $\frac{\exp(\theta_j)}{\sum_{j \in \Omega_j} \exp(\theta_j)} \geq \tilde{\alpha}_{i, \ell, \theta} / |\Omega_j|$. Therefore, each summation term in equation (65) can be lower bounded by the corresponding term where $\{\theta_j\}_{j \in S}$ is replaced by $\{\tilde{\theta}_j\}_{j \in S}$. Hence, we

have

$$\begin{aligned}
& \mathbb{P}_\theta[\sigma^{-1}(i) = \ell] \\
& \leq \sum_{\substack{j_1 \in S \\ j_1 \neq i}} \left(\frac{\exp(\tilde{\theta}_{j_1})}{\tilde{W}} \sum_{\substack{j_2 \in S \\ j_2 \neq i, j_1}} \left(\frac{\exp(\tilde{\theta}_{j_2})}{\tilde{W} - \exp(\tilde{\theta}_{j_1})} \cdots \left(\sum_{\substack{j_{\ell-1} \in S \\ j_{\ell-1} \neq i, \\ j_1, \dots, j_{\ell-2}}} \frac{\exp(\tilde{\theta}_{j_{\ell-1}})}{\tilde{W} - \sum_{k=j_1}^{j_{\ell-2}} \exp(\tilde{\theta}_k)}} \right) \right) \frac{e^{2b}}{\kappa - \ell + 1} \right) \\
& \leq e^{4b} \sum_{\substack{j_1 \in S \\ j_1 \neq i}} \left(\frac{\exp(\tilde{\theta}_{j_1})}{\tilde{W}} \sum_{\substack{j_2 \in S \\ j_2 \neq i, j_1}} \left(\frac{\exp(\tilde{\theta}_{j_2})}{\tilde{W} - \exp(\tilde{\theta}_{j_1})} \cdots \right. \right. \\
& \quad \left. \left. \left(\sum_{\substack{j_{\ell-1} \in S \\ j_{\ell-1} \neq i, \\ j_1, \dots, j_{\ell-2}}} \frac{\exp(\tilde{\theta}_{j_{\ell-1}})}{\tilde{W} - \sum_{k=j_1}^{j_{\ell-2}} \exp(\tilde{\theta}_k)} \frac{\exp(\tilde{\theta}_{j_{\ell-1}})}{\tilde{W} - \sum_{k=j_1}^{j_{\ell-2}} \exp(\tilde{\theta}_k)}} \right) \right) \right) \\
& \leq e^{4b} \mathbb{P}_\theta[\sigma^{-1}(i) = \ell]
\end{aligned} \tag{75}$$

The second inequality uses $\tilde{\alpha}_2/(\kappa - \ell + \tilde{\alpha}_{i,\ell,\theta}) \geq e^{-2b}/(\kappa - \ell + 1)$. Observe that $\exp(\tilde{\theta}_j) = 1$ for all $j \neq i$ and $\exp(\tilde{\theta}_i) = \tilde{\alpha}_{i,\ell,\theta} \geq [\tilde{\alpha}_{i,\ell,\theta}] = \alpha_{i,\ell,\theta} \geq 0$. Therefore, we have

$$\begin{aligned}
\mathbb{P}_\theta[\sigma^{-1}(i) = \ell] & = \frac{(\kappa-1)}{(\ell-1)!} \frac{\tilde{\alpha}_{i,\ell,\theta}(\ell-1)!}{(\kappa-1 + \tilde{\alpha}_{i,\ell,\theta})(\kappa-2 + \tilde{\alpha}_{i,\ell,\theta}) \cdots (\kappa-\ell + \tilde{\alpha}_{i,\ell,\theta})} \\
& \leq \frac{(\kappa-\ell)!}{(\kappa-\ell)!} \frac{e^{2b}}{(\kappa-1 + \alpha_{i,\ell,\theta})(\kappa-2 + \alpha_{i,\ell,\theta}) \cdots (\kappa-\ell + \alpha_{i,\ell,\theta})} \\
& \leq \frac{e^{2b}}{\kappa} \left(1 - \frac{\ell}{\kappa + \alpha_{i,\ell,\theta}} \right)^{\alpha_{i,\ell,\theta} - 1},
\end{aligned} \tag{76}$$

Note that equation (76) holds for all values of $\alpha_{i,\ell,\theta} \geq 0$. Claim 52 follows by combining Equations (75) and (76).

8.3 Proof of Theorem 4

Let $H(\theta) \in \mathcal{S}^d$ be Hessian matrix such that $H_{i'i'}(\theta) = \frac{\partial^2 \mathcal{L}(\theta)}{\partial \theta_i \partial \theta_{i'}}$. The Fisher information matrix is defined as $I(\theta) = -\mathbb{E}_\theta[H(\theta)]$. Fix any unbiased estimator $\hat{\theta}$ of $\theta \in \Omega_b$. Since, $\hat{\theta} \in \mathcal{U}$, $\hat{\theta} - \theta$ is orthogonal to $\mathbf{1}$. The Cramér-Rao lower bound then implies that $\mathbb{E}[\|\hat{\theta} - \theta\|^2] \geq \sum_{i=2}^d \frac{1}{\lambda_i(I(\theta))}$. Taking the supremum over both sides gives

$$\sup_{\theta} \mathbb{E}[\|\hat{\theta} - \theta\|^2] \geq \sup_{\theta} \sum_{i=2}^d \frac{1}{\lambda_i(I(\theta))} \geq \sum_{i=2}^d \frac{1}{\lambda_i(I(\mathbf{0}))}.$$

The following lemma provides a lower bound on $\mathbb{E}_\theta[H(\mathbf{0})]$, where $\mathbf{0}$ indicates the all-zeros vector.

Lemma 16. *Under the hypotheses of Theorem 4,*

$$\mathbb{E}_\theta[H(\mathbf{0})] \succeq - \sum_{j=1}^n \frac{2p \log(\kappa_j)^2}{\kappa_j(\kappa_j - 1)} \sum_{i' < i \in S_j} (e_i - e_{i'})(e_i - e_{i'})^\top. \tag{77}$$

Observe that $I(\mathbf{0})$ is positive semi-definite. Moreover, $\lambda_1(I(\mathbf{0}))$ is zero and the corresponding eigenvector is the all-ones vector. It follows that

$$\begin{aligned}
I(\mathbf{0}) & \preceq \sum_{j=1}^n \frac{2p \log(\kappa_j)^2}{\kappa_j(\kappa_j - 1)} \sum_{i' < i \in S_j} (e_i - e_{i'})(e_i - e_{i'})^\top \\
& \preceq \underbrace{2p \log(\kappa_{\max})^2 \sum_{j=1}^n \frac{1}{\kappa_j(\kappa_j - 1)} \sum_{i' < i \in S_j} (e_i - e_{i'})(e_i - e_{i'})^\top}_{=L},
\end{aligned}$$

where L is the Laplacian defined for the comparison graph \mathcal{H} , Definition 1, as $\ell_j = 1$ for all $j \in [n]$ in this setting. By Jensen's inequality, we have

$$\sum_{i=2}^d \frac{1}{\lambda_i(L)} \geq \frac{(d-1)^2}{\sum_{i=2}^d \lambda_i(L)} = \frac{(d-1)^2}{\text{Tr}(L)} = \frac{(d-1)^2}{n}.$$

8.3.1 PROOF OF LEMMA 16

Define $\mathcal{L}_j(\theta)$ for $j \in [n]$ such that $\mathcal{L}(\theta) = \sum_{j=1}^n \mathcal{L}_j(\theta)$. Let $H^{(j)}(\theta) \in \mathcal{S}^d$ be the Hessian matrix such that $H_{i'i'}^{(j)}(\theta) = \frac{\partial^2 \mathcal{L}_j(\theta)}{\partial \theta_i \partial \theta_{i'}}$ for $i, i' \in S_j$. We prove that for all $j \in [n]$,

$$\mathbb{E}_\theta[H^{(j)}(\mathbf{0})] \succeq - \frac{2p \log(\kappa_j)^2}{\kappa_j(\kappa_j - 1)} \sum_{i' < i \in S_j} (e_i - e_{i'})(e_i - e_{i'})^\top. \tag{78}$$

In the following, we omit superscript/subscript j for brevity. With a slight abuse of notation, we use $\mathbb{1}_{\{\Omega^{-1}(i)=a\}} = 1$ if item i is ranked at the a -th position in all the orderings $\sigma \in \Omega$. Let $\mathbb{P}[\theta]$ be the likelihood of observing $\Omega^{-1}(p) = i^{(p)}$ and the set Λ (the set of the items that are ranked before the p -th position). We have,

$$\mathbb{P}(\theta) = \sum_{\sigma \in \Omega} \left(\frac{\exp(\sum_{m=1}^p \theta_{\sigma(m)})}{\prod_{\sigma=1}^p \left(\sum_{m'=a}^{\kappa} \exp(\theta_{\sigma(m')}) \right)} \right). \tag{79}$$

For $i, i' \in S_j$, we have

$$H_{i'i'}(\theta) = \frac{1}{\mathbb{P}(\theta)} \frac{\partial^2 \mathbb{P}(\theta)}{\partial \theta_i \partial \theta_{i'}} - \frac{\nabla_i \mathbb{P}(\theta) \nabla_{i'} \mathbb{P}(\theta)}{(\mathbb{P}(\theta))^2} \tag{80}$$

We claim that at $\theta = \mathbf{0}$,

$$-H_{i'i'}(\mathbf{0}) = \begin{cases} C_1 & \text{if } i = i', \{\Omega^{-1}(i) \geq p\} \\ C_2 + A_3^2 - C_3 & \text{if } i = i', \{\Omega^{-1}(i) < p\} \\ -B_1 & \text{if } i \neq i', \{\Omega^{-1}(i) \geq p, \Omega^{-1}(i') \geq p\} \\ -B_2 & \text{if } i \neq i', \{\Omega^{-1}(i) \geq p, \Omega^{-1}(i') < p\} \\ -B_2 & \text{if } i \neq i', \{\Omega^{-1}(i) < p, \Omega^{-1}(i') \geq p\} \\ -(B_3 + B_4 - A_3^2) & \text{if } i \neq i', \{\Omega^{-1}(i) < p, \Omega^{-1}(i') < p\}. \end{cases} \tag{81}$$

where constants $A_3, B_1, B_2, B_3, B_4, C_1, C_2$ and C_3 are defined in Equations (88), (90), (91), (92), (93), (95), (96) and (97) respectively. From this computation of the Hessian, note that we have

$$H(\mathbf{0}) = \sum_{i' < i \in S} (e_i - e_{i'}) (e_i - e_{i'})^\top (H_{i'i'}(\mathbf{0})). \quad (82)$$

which follows directly from the fact that the diagonal entries are summations of the off-diagonals, i.e. $C_1 = B_1(\kappa - p) + B_2(p - 1)$ and $C_2 + A_3^2 - C_3 = B_2(\kappa - p + 1) + (B_3 + B_4 - A_3^2)(p - 2)$. The second equality follows from the fact that $C_2 = B_2(\kappa - p + 1) + B_3(p - 2)$ and $A_3^2(p - 1) = B_4(p - 2) + C_3$. Note that since $\theta = \mathbf{0}$, all items are exchangeable. Hence, $\mathbb{E}[H_{i'i'}(\mathbf{0})] = \mathbb{E}[H_{i'i'}(\mathbf{0})]/(\kappa - 1)$, and substituting this into (82) and using Equations (81), we get

$$\begin{aligned} & \mathbb{E}[H(\mathbf{0})] \\ &= -\frac{1}{\kappa - 1} \left(\mathbb{P}[\Omega^{-1}(i) \geq p] C_1 + \mathbb{P}[\Omega^{-1}(i) < p] (C_2 + A_3^2 - C_3) \right) \sum_{i' < i \in S} (e_i - e_{i'}) (e_i - e_{i'})^\top \\ &\succeq -\frac{1}{\kappa(\kappa - 1)} \sum_{i' < i \in S} (e_i - e_{i'}) (e_i - e_{i'})^\top \\ &\quad \left((\kappa - p + 1) \log \left(\frac{\kappa}{\kappa - p} \right) + (p - 1) \left(\log \left(\frac{\kappa}{\kappa - p + 1} \right) + \log \left(\frac{\kappa}{\kappa - p + 1} \right)^2 \right) \right) \\ &\succeq -\frac{2p \log(\kappa)^2}{\kappa(\kappa - 1)} \sum_{i' < i \in S} (e_i - e_{i'}) (e_i - e_{i'})^\top, \end{aligned} \quad (83)$$

where (83) uses $\sum_{a=1}^p \frac{1}{\kappa - a + 1} \leq \log \left(\frac{\kappa}{\kappa - p} \right)$ and $C_3 \geq 0$. Equation (84) follows from the fact that for any $x > 0$, $\log(1 + x) \leq x$. To prove (81), we have the first order partial derivative of $\mathbb{P}(\theta)$ given by

$$\nabla_{\theta_i} \mathbb{P}(\theta) = \mathbb{I}_{\{\Omega^{-1}(i) \leq p\}} \mathbb{P}(\theta) - \sum_{\sigma \in \Omega} \left(\frac{\exp \left(\sum_{m=1}^p \theta_{\sigma(m)} \right)}{\prod_{l=1}^{\kappa} \left(\sum_{m'=a}^{\kappa} \exp \left(\theta_{\sigma(m')} \right) \right)} \right) \left(\sum_{a=1}^p \frac{\mathbb{I}_{\{\sigma^{-1}(i) \geq a\}} \exp(\theta_i)}{\sum_{m'=a}^{\kappa} \exp \left(\theta_{\sigma(m')} \right)} \right) \quad (85)$$

Define constants A_1, A_2 and A_3 such that

$$A_1 \equiv \mathbb{P}(\theta) \Big|_{\{\theta=0\}} = \frac{(p-1)!}{\kappa(\kappa-1) \cdots (\kappa-p+1)}, \quad (86)$$

$$A_2 \equiv \left(\sum_{a=1}^p \frac{\exp(\theta_i)}{\sum_{m'=a}^{\kappa} \exp \left(\theta_{\sigma(m')} \right)} \right) \Big|_{\{\theta=0\}} = \left(\frac{1}{\kappa} + \frac{1}{\kappa-1} + \cdots + \frac{1}{\kappa-p+1} \right), \quad (87)$$

$$A_3 \equiv \left(\frac{(p-1)(p-2)!}{(p-1)!(\kappa)} + \frac{(p-2)(p-2)!}{(p-1)!(\kappa-1)} + \cdots + \frac{(p-2)!}{(p-1)!(\kappa-p+2)} \right). \quad (88)$$

Observe that, for all $i \in [d]$,

$$\nabla_{\theta_i} \mathbb{P}(\theta) \Big|_{\{\theta=0\}} = A_1 \left(\mathbb{I}_{\{\Omega_j^{-1}(i)=p\}} (1 - A_2) + \mathbb{I}_{\{\Omega_j^{-1}(i)<p\}} (1 - A_3) - \mathbb{I}_{\{\Omega_j^{-1}(i)>p\}} A_2 \right). \quad (89)$$

Further define constants B_1, B_2, B_3 and B_4 such that

$$B_1 \equiv \left(\frac{1}{\kappa^2} + \frac{1}{(\kappa-1)^2} + \cdots + \frac{1}{(\kappa-p+1)^2} \right), \quad (90)$$

$$B_2 \equiv \left(\frac{p-1}{(p-1)\kappa^2} + \frac{p-2}{(p-1)(\kappa-1)^2} + \cdots + \frac{1}{(p-1)(\kappa-p+2)^2} \right), \quad (91)$$

$$B_3 \equiv \left(\frac{(p-1)(p-2)(p-3)!}{(p-1)!\kappa^2} + \frac{(p-2)(p-3)(p-3)!}{(p-1)!(\kappa-1)^2} + \cdots + \frac{2(p-3)!}{(p-1)!(\kappa-p+3)^2} \right), \quad (92)$$

$$B_4 \equiv \frac{(p-3)!}{(p-1)!} \left(\sum_{a,b \in [p-1], b \neq a} \left(\frac{1}{\kappa} + \frac{1}{\kappa-1} + \cdots + \frac{1}{\kappa-a+1} \right) \left(\frac{1}{\kappa} + \frac{1}{\kappa-1} + \cdots + \frac{1}{\kappa-b+1} \right) \right) \quad (93)$$

Observe that,

$$\begin{aligned} & \frac{\partial^2 \mathbb{P}(\theta)}{\partial \theta_i^2} \Big|_{\theta=0} \\ &= \mathbb{I}_{\{\Omega^{-1}(i), \Omega^{-1}(i') > p\}} A_1 \left((-A_2) (-A_2) + B_1 \right) \\ &\quad + \left(\mathbb{I}_{\{\Omega^{-1}(i) > p, \Omega^{-1}(i') = p\}} + \mathbb{I}_{\{\Omega^{-1}(i) = p, \Omega^{-1}(i') > p\}} \right) A_1 \left((-A_2)(1 - A_2) + B_1 \right) \\ &\quad + \left(\mathbb{I}_{\{\Omega^{-1}(i) = p, \Omega^{-1}(i') < p\}} + \mathbb{I}_{\{\Omega^{-1}(i) < p, \Omega^{-1}(i') = p\}} \right) A_1 \left((1 - A_3) + (-A_2)(1 - A_3) + B_2 \right) \\ &\quad + \left(\mathbb{I}_{\{\Omega^{-1}(i) > p, \Omega^{-1}(i') < p\}} + \mathbb{I}_{\{\Omega^{-1}(i) < p, \Omega^{-1}(i') > p\}} \right) A_1 \left((-A_2)(1 - A_3) + B_2 \right) \\ &\quad + \mathbb{I}_{\{\Omega^{-1}(i) < p, \Omega^{-1}(i') < p\}} A_1 \left((1 - A_3) + (-A_3) + B_4 + B_3 \right). \end{aligned} \quad (94)$$

The claims in (81) are easy to verify by combining Equations (89) and (94) with (80). Also, define constants C_1, C_2 and C_3 such that,

$$C_1 \equiv \left(\frac{\kappa-1}{(\kappa)^2} + \frac{\kappa-2}{(\kappa-1)^2} + \cdots + \frac{\kappa-p}{(\kappa-p+1)^2} \right), \quad (95)$$

$$C_2 \equiv \left(\frac{(p-1)(p-2)(\kappa-1)}{(p-1)!(\kappa)^2} + \frac{(p-2)(p-2)(\kappa-2)}{(p-1)!(\kappa-1)^2} + \cdots + \frac{(p-2)(\kappa-p+1)}{(p-1)!(\kappa-p+2)^2} \right), \quad (96)$$

$$C_3 \equiv \frac{(p-2)!}{(p-1)!} \left(\sum_{a,b \in [p-1], b \neq a} \left(\frac{1}{\kappa} + \frac{1}{\kappa-1} + \cdots + \frac{1}{\kappa-a+1} \right) \left(\frac{1}{\kappa} + \frac{1}{\kappa-1} + \cdots + \frac{1}{\kappa-b+1} \right) \right) \quad (97)$$

such that,

$$\begin{aligned} & \frac{\partial^2 \mathbb{P}(\theta)}{\partial \theta_i^2} \Big|_{\theta=0} \\ &= \mathbb{I}_{\{\Omega^{-1}(i) > p\}} A_1 \left((-A_2) (-A_2) - C_1 \right) + \mathbb{I}_{\{\Omega^{-1}(i) = p\}} A_1 \left((1 - A_2) - A_2(1 - A_2) - C_1 \right) \\ &\quad + \mathbb{I}_{\{\Omega^{-1}(i) < p\}} A_1 \left((1 - A_3) - A_3 - C_2 + C_3 \right). \end{aligned} \quad (98)$$

The claims (81) is easy to verify by combining Equations (89) and (98) with (80).

8.4 Proof of Theorem 5

The proof is analogous to the proof of Theorem 8. It differs primarily in the lower bound that is achieved for the second smallest eigenvalue of the Hessian matrix $H(\theta)$, (35).

Lemma 17. *Under the hypotheses of Theorem 5, if $\sum_{j=1}^n \ell_j \geq (2^{12} e^{6b} / \beta \alpha^2) d \log d$ then with probability at least $1 - d^{-3}$,*

$$\lambda_2(-H(\theta)) \geq \frac{\alpha}{2(1 + e^{2b})^2} \frac{1}{d-1} \sum_{j=1}^n \ell_j. \quad (99)$$

Using Lemma 10 that is derived for the general value of $\lambda_{j,a}$ and $p_{j,a}$, and by substituting $\lambda_{j,a} = 1/(\kappa_j - 1)$ and $p_{j,a} = a$ for each $j \in [n]$, we get that with probability at least $1 - 2e^{\alpha} d^{-3}$,

$$\|\nabla \mathcal{L}_{\text{RB}}(\theta^*)\|_2 \leq \sqrt{16 \log d \sum_{j=1}^n \ell_j}. \quad (100)$$

Theorem 5 follows from Equations (100), (99) and (40).

8.4.1 PROOF OF LEMMA 17

Define $M^{(j)} \in \mathbb{S}^d$ as

$$M^{(j)} = \frac{1}{\kappa_j - 1} \sum_{i < i' \in S_j} \mathbb{1}_{\{(i,i') \in G_{j,a}\}} (e_i - e_{i'})(e_i - e_{i'})^\top, \quad (101)$$

and let $M = \sum_{j=1}^n M^{(j)}$. Similar to the analysis carried out in the proof of Lemma 11, we have $\lambda_2(-H(\theta)) \geq \frac{e^{2b}}{(1+e^{2b})^2} \lambda_2(M)$, when $\lambda_{j,a} = 1/(\kappa_j - 1)$ is substituted in the Hessian matrix $H(\theta)$, Equation (35). From Weyl's inequality we have that

$$\lambda_2(M) \geq \lambda_2(\mathbb{E}[M]) - \|M - \mathbb{E}[M]\|. \quad (102)$$

We will show in (107) that $\lambda_2(\mathbb{E}[M]) \geq e^{-2b}(\alpha/(d-1)) \sum_{j=1}^n \ell_j$ and in (112) that $\|M - \mathbb{E}[M]\| \leq 32e^b \sqrt{\frac{\log d}{\beta d} \sum_{j=1}^n \ell_j}$.

$$\lambda_2(M) \geq \frac{\alpha e^{-2b}}{d-1} \sum_{j=1}^n \ell_j - 32e^b \sqrt{\frac{\log d}{\beta d} \sum_{j=1}^n \ell_j} \geq \frac{\alpha e^{-2b}}{2(d-1)} \sum_{j=1}^n \ell_j, \quad (103)$$

where the last inequality follows from the assumption that $\sum_{j=1}^n \ell_j \geq (2^{12} e^{6b} / \beta \alpha^2) d \log d$. This proves the desired claim.

To prove the lower bound on $\lambda_2(\mathbb{E}[M])$, notice that

$$\mathbb{E}[M] = \sum_{j=1}^n \frac{1}{\kappa_j - 1} \sum_{i < i' \in S_j} \mathbb{E} \left[\sum_{\alpha=1}^{\ell_j} \mathbb{1}_{\{(i,i') \in G_{j,a}\}} \right] (e_i - e_{i'})(e_i - e_{i'})^\top. \quad (104)$$

Using the fact that $p_{j,a} = a$ for each $j \in [n]$, and the definition of rank-breaking graph $G_{j,a}$, we have that

$$\begin{aligned} \mathbb{E} \left[\sum_{\alpha=1}^{\ell_j} \mathbb{1}_{\{(i,i') \in G_{j,a}\}} \right] (i, i' \in S_j) &= \mathbb{P} \left[\mathbb{1}_{\{\sigma_j^{-1}(i) \leq \ell_j\}} + \mathbb{1}_{\{\sigma_j^{-1}(i') \leq \ell_j\}} \geq 1 \mid (i, i' \in S_j) \right] \\ &\geq \mathbb{P} \left[(\sigma^{-1}(i) \leq \ell_j) \mid (i, i' \in S_j) \right]. \end{aligned} \quad (105)$$

The following lemma provides a lower bound on $\mathbb{P}[(\sigma^{-1}(i) \leq \ell_j) \mid (i, i' \in S_j)]$.

Lemma 18. *Consider a ranking σ over a set of items S of size κ . For any item $i \in S$,*

$$\mathbb{P}[(\sigma^{-1}(i) \leq \ell] \geq e^{-2b} \frac{\ell}{\kappa}. \quad (106)$$

Therefore, using the fact that $(e_i - e_{i'})(e_i - e_{i'})^\top$ is positive semi-definite, and Equations (104), (105) and (106) we have

$$\mathbb{E}[M] \succeq e^{-2b} \sum_{j=1}^n \frac{\ell_j}{\kappa_j(\kappa_j - 1)} \sum_{i < i' \in S_j} (e_i - e_{i'})(e_i - e_{i'})^\top = e^{-2b} L, \quad (107)$$

where L is the Laplacian defined for the comparison graph \mathcal{H} , Definition 1. Using $\lambda_2(L) = (\alpha/(d-1)) \sum_{j=1}^n \ell_j$ from (5), we get the desired bound $\lambda_2(\mathbb{E}[M]) \geq e^{-2b}(\alpha/(d-1)) \sum_{j=1}^n \ell_j$.

For top- ℓ_j rank breaking, $M^{(j)}$ is also given by

$$M^{(j)} = \frac{1}{\kappa_j - 1} \left((\kappa_j - \ell_j) \text{diag}(e_{\{I_j\}}) + \ell_j \text{diag}(e_{\{S_j\}}) - e_{\{I_j\}} e_{\{S_j\}}^\top - e_{\{S_j\}} e_{\{I_j\}}^\top \right), \quad (108)$$

where $e_{\{S_j\}}, e_{\{I_j\}} \in \mathbb{R}^d$ are zero-one vectors, $e_{\{S_j\}}$ has support corresponding to the set of items S_j and $e_{\{I_j\}}$ has support corresponding to the random top- ℓ_j items in the ranking σ_j . $I_j = \{\sigma_j(1), \sigma_j(2), \dots, \sigma_j(\ell_j)\}$ for $j \in [n]$. $(M^{(j)})^2$ is given by

$$\begin{aligned} (M^{(j)})^2 &= \frac{1}{(\kappa_j - 1)^2} \left((\kappa_j^2 - \ell_j^2) \text{diag}(e_{\{I_j\}}) + \ell_j^2 \text{diag}(e_{\{S_j\}}) - \right. \\ &\quad \left. (\kappa_j + \ell_j)(e_{\{I_j\}} e_{\{S_j\}}^\top + e_{\{S_j\}} e_{\{I_j\}}^\top) - e_{\{I_j\}} e_{\{I_j\}}^\top - e_{\{I_j\}} e_{\{S_j\}}^\top - e_{\{S_j\}} e_{\{I_j\}}^\top \right). \end{aligned}$$

Note that $\mathbb{P}[i \in I_j \mid i \in S_j] \leq \ell_j e^{2b} / \kappa_j$ for all $i \in S_j$. Its proof is similar to the proof of Lemma 18. Therefore, we have $\mathbb{E}[\text{diag}(e_{\{I_j\}})] \preceq \ell_j e^{2b} / \kappa_j \text{diag}(e_{\{I\}})$. To bound $\|\sum_{j=1}^n \mathbb{E}[(M^{(j)})^2]\|$, we use the fact that for $J \in \mathbb{R}^{d \times d}$, $\|J\| \leq \max_{i \in [d]} \sum_{i'=1}^d |J_{i'i'}|$. Maximum of row sums of $\mathbb{E}[e_{\{I_j\}} e_{\{I_j\}}^\top]$ is upper

bounded by $\max_{i \in [d]} \{\ell_j \mathbb{P}[i \in I_j | i \in S_j]\} \leq \ell_j e^{2b} / \kappa_j$. Therefore using triangle inequality, we have,

$$\begin{aligned}
& \left\| \sum_{j=1}^n \mathbb{E}[(M^{(j)})^2] \right\| \\
& \leq \max_{i \in [d]} \left\{ \sum_{j: i \in S_j} \frac{1}{(\kappa_j - 1)^2} \left(\frac{(\kappa_j^2 - \ell_j^2) \ell_j e^{2b}}{\kappa_j} + \ell_j^2 + e^{2b} (\kappa_j + \ell_j) (2\ell_j + \ell_j^2 / \kappa_j) + \ell_j \kappa_j \right) \right\} \\
& \leq \max_{i \in [d]} \left\{ \sum_{j: i \in S_j} \frac{\ell_j e^{2b}}{\kappa_j} \left(\frac{(\kappa_j^2 - \ell_j^2)}{(\kappa_j - 1)^2} + \frac{\ell_j \kappa_j}{(\kappa_j - 1)^2} + \frac{2(\kappa_j + \ell_j) \kappa_j}{(\kappa_j - 1)^2} + \frac{(\kappa_j + \ell_j) \ell_j}{(\kappa_j - 1)^2} + \frac{\kappa_j^2}{(\kappa_j - 1)^2} \right) \right\} \\
& \leq \max_{i \in [d]} \left\{ \sum_{j: i \in S_j} \frac{\ell_j e^{2b}}{\kappa_j} \left(\frac{(\kappa_j^2 - 1)}{(\kappa_j - 1)^2} + \frac{\kappa_j (\kappa_j - 1)}{(\kappa_j - 1)^2} + \frac{4\kappa_j^2}{(\kappa_j - 1)^2} + \frac{2\kappa_j (\kappa_j - 1)}{(\kappa_j - 1)^2} + \frac{\kappa_j^2}{(\kappa_j - 1)^2} \right) \right\} \\
& \leq \max_{i \in [d]} \left\{ \sum_{j: i \in S_j} \frac{\ell_j e^{2b}}{\kappa_j} \left(3 + 2 + 16 + 4 + 4 \right) \right\} \\
& \leq 29e^{2b} \max_{i \in [d]} \left\{ \sum_{j: i \in S_j} \frac{\ell_j}{\kappa_j} \right\} \\
& = 29e^{2b} D_{\max} \\
& = \frac{29e^{2b}}{\beta d} \sum_{j=1}^n \ell_j,
\end{aligned} \tag{110}$$

where (109) uses the fact that $\kappa_j \geq 2$ and $1 \leq \ell_j \leq \kappa_j - 1$ for all $j \in [n]$. (110) follows from the definition of D_{\max} , Definition 1 and (111) follows from the Equation (6). Also, note that $\|M_j\| \leq 2$ for all $j \in [n]$. Applying matrix Bernstein inequality, we have,

$$\mathbb{P} \left[\|M - \mathbb{E}[M]\| \geq t \right] \leq d \exp \left(\frac{-t^2/2}{\frac{29e^{2b}}{\beta d} \sum_{j=1}^n \ell_j + 4t/3} \right).$$

Therefore, with probability at least $1 - d^{-3}$, we have,

$$\|M - \mathbb{E}[M]\| \leq 22e^b \sqrt{\frac{\log d}{\beta d} \sum_{j=1}^n \ell_j} + \frac{64 \log d}{3} \leq 32e^b \sqrt{\frac{\log d}{\beta d} \sum_{j=1}^n \ell_j}, \tag{112}$$

where the second inequality follows from the assumption that $\sum_{j=1}^n \ell_j \geq 2^{12} d \log d$ and $\beta \leq 1$.

8.4.2 PROOF OF LEMMA 18

Define $i_{\min} \equiv \arg \min_{i \in S} \theta_i$. We claim the following. For all $i \in S$ and any $1 \leq \ell \leq |S| - 1$,

$$\mathbb{P}[\sigma^{-1}(i) > \ell] \leq \mathbb{P}[\sigma^{-1}(i_{\min}) > \ell] \text{ and } \mathbb{P}[\sigma^{-1}(i_{\min}) = \ell] \geq \mathbb{P}[\sigma^{-1}(i_{\min}) = 1]. \tag{113}$$

Therefore $\mathbb{P}[\sigma^{-1}(i) \leq \ell] \geq \mathbb{P}[\sigma^{-1}(i_{\min}) \leq \ell]$. Using $\mathbb{P}[\sigma^{-1}(i_{\min}) = 1] > e^{-2b} / \kappa$, we get the desired bound $\mathbb{P}[\sigma^{-1}(i) \leq \ell] \geq e^{-2b} \ell / \kappa$.

To prove the claim (113), let $\hat{\sigma}_\ell^i$ denote a ranking of top- ℓ items of the set S and $\mathbb{P}[\hat{\sigma}_\ell^i]$ be the probability of observing $\hat{\sigma}_\ell^i$. Let $i \in (\hat{\sigma}_\ell^i)^{-1}$ denote that $i = (\hat{\sigma}_\ell^i)^{-1}(j)$ for some $1 \leq j \leq \ell$. Let

$$\Omega_1 = \left\{ \hat{\sigma}_\ell^i : i \notin (\hat{\sigma}_\ell^i)^{-1}, i_{\min} \in (\hat{\sigma}_\ell^i)^{-1} \right\} \text{ and } \Omega_2 = \left\{ \hat{\sigma}_\ell^i : i \in (\hat{\sigma}_\ell^i)^{-1}, i_{\min} \notin (\hat{\sigma}_\ell^i)^{-1} \right\}.$$

We have $\mathbb{P}[\sigma^{-1}(i) > \ell] = \mathbb{P}[\sigma^{-1}(i_{\min}) > \ell] = \sum_{\hat{\sigma}_\ell^i \in \Omega_1} \mathbb{P}[\hat{\sigma}_\ell^i] - \sum_{\hat{\sigma}_\ell^i \in \Omega_2} \mathbb{P}[\hat{\sigma}_\ell^i]$. Now, take any ranking $\hat{\sigma}_\ell^i \in \Omega_1$ and construct another ranking $\hat{\sigma}_\ell^i$ from $\hat{\sigma}_\ell^i$ by replacing i_{\min} with i -th item. Observe that $\mathbb{P}[\hat{\sigma}_\ell^i] \leq \mathbb{P}[\hat{\sigma}_\ell^i]$ and $\hat{\sigma}_\ell^i \in \Omega_2$. Moreover, such a construction gives a bijective mapping between Ω_1 and Ω_2 . Hence, the first claim is proved. For the second claim, let

$$\hat{\Omega}_1 = \left\{ \hat{\sigma}_\ell^i : (\hat{\sigma}_\ell^i)^{-1}(i_{\min}) = 1 \right\} \text{ and } \hat{\Omega}_2 = \left\{ \hat{\sigma}_\ell^i : (\hat{\sigma}_\ell^i)^{-1}(i_{\min}) = \ell \right\}.$$

We have $\mathbb{P}[\sigma^{-1}(i_{\min}) = 1] = \mathbb{P}[\sigma^{-1}(i_{\min}) = 1] = \sum_{\hat{\sigma}_\ell^i \in \hat{\Omega}_1} \mathbb{P}[\hat{\sigma}_\ell^i] - \sum_{\hat{\sigma}_\ell^i \in \hat{\Omega}_2} \mathbb{P}[\hat{\sigma}_\ell^i]$. Now, take any ranking $\hat{\sigma}_\ell^i \in \hat{\Omega}_1$ and construct another ranking $\hat{\sigma}_\ell^i$ from $\hat{\sigma}_\ell^i$ by swapping items at 1st position and ℓ -th position. Observe that $\mathbb{P}[\hat{\sigma}_\ell^i] \leq \mathbb{P}[\hat{\sigma}_\ell^i]$ and $\hat{\sigma}_\ell^i \in \hat{\Omega}_2$. Moreover, such a construction gives a bijective mapping between $\hat{\Omega}_1$ and $\hat{\Omega}_2$. Hence, the claim is proved.

8.5 Proof of Theorem 6

The first order partial derivative of $\mathcal{L}(\theta)$, Equation (15), is given by

$$\begin{aligned}
& \nabla_i \mathcal{L}(\theta) \\
& = \sum_{j: i \in S_j} \sum_{m=1}^{\ell_j} \mathbb{I}_{\{\sigma_j^{-1}(i) \geq m\}} \left[\frac{\mathbb{I}_{\{\sigma_j(m)=i\}}}{\exp(\theta_{\sigma_j(m)}) + \exp(\theta_{\sigma_j(m+1)}) + \dots + \exp(\theta_{\sigma_j(\kappa_j)})} \right], \forall i \in [d]
\end{aligned}$$

and the Hessian matrix $H(\theta) \in S^d$ with $H_{ir}(\theta) = \frac{\partial^2 \mathcal{L}(\theta)}{\partial \theta_i \partial \theta_r}$ is given by

$$\begin{aligned}
H(\theta) &= - \sum_{j=1}^n \sum_{i < r \in S_j} (e_i - e_r)(e_i - e_r)^\top \sum_{m=1}^{\ell_j} \frac{\exp(\theta_i + \theta_r) \mathbb{I}_{\{\sigma_j^{-1}(i), \sigma_j^{-1}(r) \geq m\}}}{[\exp(\theta_{\sigma_j(m)}) + \exp(\theta_{\sigma_j(m+1)}) + \dots + \exp(\theta_{\sigma_j(\kappa_j)})]^2}.
\end{aligned} \tag{114}$$

It follows from the definition that $-H(\theta)$ is positive semi-definite for any $\theta \in \mathbb{R}^n$.

The Fisher information matrix is defined as $I(\theta) = -\mathbb{E}_\theta[H(\theta)]$ and given by

$$I(\theta) = \sum_{j=1}^n \sum_{i < r \in S_j} (e_i - e_r)(e_i - e_r)^\top \sum_{m=1}^{\ell_j} \mathbb{E} \left[\frac{\mathbb{I}_{\{\sigma_j^{-1}(i), \sigma_j^{-1}(r) \geq m\}}}{[\exp(\theta_{\sigma_j(m)}) + \dots + \exp(\theta_{\sigma_j(\kappa_j)})]^2} \right] \exp(\theta_i + \theta_r).$$

Since $-H(\theta)$ is positive semi-definite, it follows that $I(\theta)$ is positive semi-definite. Moreover, $\lambda_1(I(\theta))$ is zero and the corresponding eigenvector is the all-ones vector. Fix any unbiased estimator $\hat{\theta}$ of $\theta \in \Omega_n$. Since, $\hat{\theta} \in \mathcal{U}$, $\hat{\theta} - \theta$ is orthogonal to $\mathbf{1}$. The Cramér-Rao lower bound then implies that $\mathbb{E}[\|\hat{\theta} - \theta\|^2] \geq \sum_{i=2}^d \frac{1}{\lambda_i(I(\hat{\theta}))}$. Taking the supremum over both sides gives

$$\sup_{\hat{\theta}} \mathbb{E}[\|\hat{\theta} - \theta\|^2] \geq \sup_{\hat{\theta}} \sum_{i=2}^d \frac{1}{\lambda_i(I(\hat{\theta}))} \geq \sum_{i=2}^d \frac{1}{\lambda_i(I(\mathbf{0}))}.$$

If θ equals the all-zero vector, then

$$\mathbb{P}_\theta[\sigma_j^{-1}(i), \sigma_j^{-1}(i') \geq m] = \frac{\binom{\kappa_j - m + 1}{2}}{\binom{\kappa_j}{2}} = \frac{(\kappa_j - m + 1)(\kappa_j - m)}{\kappa_j(\kappa_j - 1)}.$$

It follows from the definition that

$$\begin{aligned} I(0) &= \sum_{j=1}^n \sum_{i < i' \in S_j} (e_i - e_{i'})^T \sum_{m=1}^{\ell_j} \frac{(\kappa_j - m)}{\kappa_j(\kappa_j - 1)(\kappa_j - m + 1)} \\ &\leq \ell \left(1 - \sum_{m=1}^{\ell_j} \frac{1}{\kappa_{\max} - m + 1} \right) \underbrace{\sum_{j=1}^n \frac{1}{\kappa_j(\kappa_j - 1)} \sum_{i < i' \in S_j} (e_i - e_{i'})^T}_{=L}, \end{aligned}$$

where L is the Laplacian defined for the comparison graph \mathcal{H} , Definition 1. By Jensen's inequality, we have

$$\sum_{i=2}^d \lambda_i(L) \geq \frac{(d-1)^2}{\sum_{i=2}^d \lambda_i(L)} = \frac{(d-1)^2}{\text{Tr}(L)} = \frac{(d-1)^2}{n}.$$

8.6 Proof of Theorem 7

We prove a slightly more general result that implies the desired theorem. For $\ell \geq 4$, we can choose $\beta_1 = 1/2$. Then, the condition that $\gamma_{\beta_1} \leq 1$ implies $\bar{d} \leq (\ell/2 + 1)(d-2)/(\kappa-2)$, which implies $\bar{d} \leq \ell d/(2\kappa)$. With the choice of $\bar{d} = \ell d/(2\kappa)$, this implies Theorem 7.

Theorem 19. *Under the bottom- ℓ separators scenario and the PL model, n partial orderings are sampled over d items parametrized by $\theta^* \in \Omega_b$. For any β_1 with $0 \leq \beta_1 \leq \frac{\ell-2}{\ell}$, define*

$$\gamma_{\beta_1} \equiv \frac{\bar{d}(\kappa-2)}{(\lfloor \ell \beta_1 \rfloor + 1)(d-2)}, \quad (115)$$

and for $\gamma_{\beta_1} \leq 1$,

$$\chi_{\beta_1} \equiv (1 - \lfloor \ell \beta_1 \rfloor / \ell)^2 \left(1 - \exp \left(- \frac{(\lfloor \ell \beta_1 \rfloor + 1)^2 (1 - \gamma_{\beta_1})^2}{2(\kappa-2)} \right) \right). \quad (116)$$

If

$$n\ell \geq \left(\frac{2^{12} e^{8b} d^2 \kappa}{\chi_{\beta_1}^2 \bar{d}^2 \ell} \right) d \log d, \quad (117)$$

then the rank-breaking estimator in (18) achieves

$$\frac{1}{\sqrt{\bar{d}}} \|\hat{\theta} - \tilde{\theta}^*\|_2 \leq \frac{32\sqrt{2}(1 + e^{4b})^2 d^{3/2}}{\chi_{\beta_1}} \frac{d^{3/2}}{\bar{d}^{3/2}} \sqrt{\frac{d \log d}{n\ell}}, \quad (118)$$

with probability at least $1 - 3e^3 d^{-3}$.

Proof is very similar to the proof of Theorem 8. It mainly differs in the lower bound that is achieved for the second smallest eigenvalue of the Hessian matrix $H(\tilde{\theta})$ of $\mathcal{L}_{\text{RB}}(\tilde{\theta})$, Equation (17). Equation (17) can be rewritten as

$$\mathcal{L}_{\text{RB}}(\tilde{\theta}) = \sum_{j=1}^n \sum_{\alpha=1}^{\ell} \sum_{i < i' \in S_j} \mathbb{I}_{\{(i, i') \in G_{j,\alpha}\}} \lambda_{j,\alpha} \left(\tilde{\theta}_i \mathbb{I}_{\{\sigma_j^{-1}(i) < \sigma_j^{-1}(i')\}} + \tilde{\theta}_{i'} \mathbb{I}_{\{\sigma_j^{-1}(i) > \sigma_j^{-1}(i')\}} - \log \left(e^{\tilde{\theta}_i} + e^{\tilde{\theta}_{i'}} \right) \right), \quad (119)$$

where $(i, i') \in G_{j,\alpha}$ implies either (i, i') or (i', i) belong to $E_{j,\alpha}$. The Hessian matrix $H(\tilde{\theta}) \in \mathcal{S}^{\bar{d}}$ with $H_{i,i'}(\tilde{\theta}) = \frac{\partial^2 \mathcal{L}_{\text{RB}}(\tilde{\theta})}{\partial \tilde{\theta}_i \partial \tilde{\theta}_{i'}}$ is given by

$$H(\tilde{\theta}) = - \sum_{j=1}^n \sum_{\alpha=1}^{\ell} \sum_{i < i' \in S_j} \mathbb{I}_{\{(i, i') \in G_{j,\alpha}\}} \left((\tilde{e}_i - \tilde{e}_{i'}) (\tilde{e}_i - \tilde{e}_{i'})^T \frac{\exp(\tilde{\theta}_i) + \tilde{\theta}_{i'}}{[\exp(\tilde{\theta}_i) + \exp(\tilde{\theta}_{i'})]^2} \right). \quad (120)$$

The following lemma gives a lower bound for $\lambda_2(-H(\tilde{\theta}))$.

Lemma 20. *Under the hypothesis of Theorem 19, with probability at least $1 - d^{-3}$,*

$$\lambda_2(-H(\tilde{\theta})) \geq \frac{\chi_{\beta_1}}{8(1 + e^{4b})^2} \frac{n\bar{d}\ell^2}{d^2}. \quad (121)$$

Observe that although $\tilde{\theta}^* \in \mathbb{R}^{\bar{d}}$, Lemma 10 can be directly applied to upper bound $\|\nabla \mathcal{L}_{\text{RB}}(\tilde{\theta}^*)\|_2$. It might be possible to tighten the upper bound, given that $\bar{d} \leq d$. However, for $\ell \ll \kappa$, for the smallest preference score item, $i_{\min} \equiv \arg \min_{i \in [d]} \theta_i^*$, the upper bound $\mathbb{P}[\sigma^{-1}(i_{\min}) > \kappa - \ell] \leq 1$ is tight upto constant factor (Lemma 15). Substituting $\lambda_{j,\alpha} = 1$ and $p_{j,\alpha} = \kappa - \ell + a$ for each $j \in [n]$, $a \in [d]$, in Lemma 10, we have that with probability at least $1 - 2e^3 d^{-3}$,

$$\|\nabla \mathcal{L}_{\text{RB}}(\tilde{\theta}^*)\|_2 \leq (\ell - 1) \sqrt{8n\ell \log d}. \quad (122)$$

Theorem 19 follows from Equations (40), (121) and (122).

8.6.1 PROOF OF LEMMA 20

Define $\tilde{M}^{(j)} \in \mathcal{S}^{\bar{d}}$,

$$\tilde{M}^{(j)} = \sum_{i < i' \in S_j} \mathbb{I}_{\{(i, i') \in G_{j,\alpha}\}} (\tilde{e}_i - \tilde{e}_{i'}) (\tilde{e}_i - \tilde{e}_{i'})^T, \quad (123)$$

and let $\tilde{M} = \sum_{j=1}^n \tilde{M}^{(j)}$. Similar to the analysis in Lemma 11, we have $\lambda_2(-H(\tilde{\theta})) \geq \frac{e^{4b}}{(1 + e^{4b})^2} \lambda_2(\tilde{M})$. Note that we have e^{4b} instead of e^{2b} as $\tilde{\theta} \in \tilde{\Omega}_{2b}$. We will show a lower bound on $\lambda_2(\mathbb{E}[\tilde{M}])$ in (129) and an upper bound on $\|\tilde{M} - \mathbb{E}[\tilde{M}]\|$ in (133). Therefore using $\lambda_2(\tilde{M}) \geq \lambda_2(\mathbb{E}[\tilde{M}]) - \|\tilde{M} - \mathbb{E}[\tilde{M}]\|$,

$$\lambda_2(\tilde{M}) \geq \frac{e^{-4b}}{4} \underbrace{(1 - \beta_1)^2 \left(1 - \exp \left(- \frac{(\lfloor \ell \beta_1 \rfloor + 1)^2 (1 - \gamma_{\beta_1})^2}{2(\kappa-2)} \right) \right)}_{\equiv \chi_{\beta_1}} \frac{n\bar{d}\ell^2}{d^2} - 8\ell \sqrt{\frac{n\kappa \log d}{d}}. \quad (124)$$

The desired claim follows from the assumption that $n\ell \geq \left(\frac{2^{12}e^{8\kappa}}{\chi_{\beta_1}^2} \frac{\ell^2 \kappa}{d^2}\right) d \log d$, where χ_{β_1} is defined in (117). To prove the lower bound on $\lambda_2(\mathbb{E}[\widetilde{M}])$, notice that

$$\mathbb{E}[\widetilde{M}] = \sum_{j=1}^n \sum_{i, i' \in [\bar{d}]} \mathbb{E} \left[\sum_{\alpha=1}^{\ell} \mathbb{I}_{\{i, i' \in G_{j,\alpha}\}} \left| (i, i' \in S_j) \right. \right] \mathbb{P} \left[i, i' \in S_j \right] (\tilde{c}_i - \tilde{c}_{i'}) (\tilde{c}_i - \tilde{c}_{i'})^\top. \quad (125)$$

Since the sets S_j are chosen uniformly at random, $\mathbb{P}[i, i' \in S_j] = \kappa(\kappa - 1)/d(d - 1)$. Using the fact that $p_{j,\alpha} = \kappa - \ell + a$ for each $j \in [n]$, and the definition of rank breaking graph $G_{j,\alpha}$, we have that

$$\mathbb{E} \left[\sum_{\alpha=1}^{\ell} \mathbb{I}_{\{i, i' \in G_{j,\alpha}\}} \left| (i, i' \in S_j) \right. \right] = \mathbb{P} \left[(\sigma_j^{-1}(i), \sigma_j^{-1}(i')) > \kappa - \ell \mid (i, i' \in S_j) \right]. \quad (126)$$

The following lemma provides a lower bound on $\mathbb{P}[(\sigma_j^{-1}(i), \sigma_j^{-1}(i')) > \kappa - \ell \mid (i, i' \in S_j)]$.

Lemma 21. *Under the hypotheses of Theorem 19, for any two items $i, i' \in [\bar{d}]$, the following holds:*

$$\mathbb{P} \left[\sigma^{-1}(i), \sigma^{-1}(i') > \kappa - \ell \mid i, i' \in S \right] \geq \frac{e^{-4b}(1 - \beta_1)^2(1 - \exp(-\eta_{\beta_1}(1 - \gamma_{\beta_1})^2))}{2} \frac{\ell^2}{\kappa^2}, \quad (127)$$

where $\gamma_{\beta_1} \equiv \bar{d}(\kappa - 2)/(\lfloor \ell \beta_1 \rfloor + 1)(d - 2)$ and $\eta_{\beta_1} \equiv (\lfloor \ell \beta_1 \rfloor + 1)^2/2(\kappa - 2)$.

Therefore, using Equations (125), (126) and (127) we have,

$$\mathbb{E}[\widetilde{M}] \succeq \frac{e^{-4b}(1 - \beta_1)^2(1 - \exp(-\eta_{\beta_1}(1 - \gamma_{\beta_1})^2))}{2} \frac{\ell^2 \kappa(\kappa - 1)}{\kappa^2 d(d - 1)} \sum_{j=1}^n \sum_{i, i' \in [\bar{d}]} (\tilde{c}_i - \tilde{c}_{i'}) (\tilde{c}_i - \tilde{c}_{i'})^\top. \quad (128)$$

Define $\tilde{L} = \sum_{j=1}^n \sum_{i, i' \in [\bar{d}]} (\tilde{c}_i - \tilde{c}_{i'}) (\tilde{c}_i - \tilde{c}_{i'})^\top$. We have, $\lambda_1(\tilde{L}) = 0$ and $\lambda_2(\tilde{L}) = \lambda_3(\tilde{L}) = \dots = \lambda_{\bar{d}}(\tilde{L})$. Therefore, using $\lambda_2(\tilde{L}) = \text{Tr}(\tilde{L})/(\bar{d} - 1) = n\bar{d}$. Using the fact that $\mathbb{E}[\widetilde{M}]$ and \tilde{L} are symmetric matrices, we have,

$$\lambda_2(\mathbb{E}[\widetilde{M}]) \geq \frac{e^{-4b}(1 - \beta_1)^2(1 - \exp(-\eta_{\beta_1}(1 - \gamma_{\beta_1})^2))}{4} \frac{n\bar{d}\ell^2}{d^2}. \quad (129)$$

To get an upper bound on $\|\widetilde{M} - \mathbb{E}[\widetilde{M}]\|$, notice that $\widetilde{M}^{(i)}$ is also given by,

$$\widetilde{M}^{(i)} = \ell \text{diag}(\tilde{e}_{i,j}) - \tilde{e}_{i,j} \tilde{e}_{i,j}^\top, \quad (130)$$

where $\tilde{e}_{i,j} \in \mathbb{R}^{\bar{d}}$ is a zero-one vector with support corresponding to the bottom- ℓ subset of items in the ranking σ_j . $I_j = \{\sigma_j(\kappa - \ell + 1), \dots, \sigma_j(\kappa)\}$ for $j \in [n]$. $(\widetilde{M}^{(i)})^2$ is given by

$$(\widetilde{M}^{(i)})^2 = \ell^2 \text{diag}(\tilde{e}_{i,j}) - \ell \tilde{e}_{i,j} \tilde{e}_{i,j}^\top. \quad (131)$$

Using the fact that sets $\{S_j\}_{j \in [n]}$ are chosen uniformly at random and $\mathbb{P}[i \in S_j] \leq 1$, we have $\mathbb{E}[\text{diag}(\tilde{e}_{i,j})] \preceq (\kappa/d) \text{diag}(e_{i,j})$. Maximum of row sums of $\mathbb{E}[\tilde{e}_{i,j}] \tilde{e}_{i,j}^\top$ is upper bounded by

$\ell\kappa/d$. Therefore, from triangle inequality we have $\|\sum_{j=1}^n \mathbb{E}[(\widetilde{M}^{(i)})^2]\| \leq 2n\ell^2\kappa/d$. Also, note that $\|\widetilde{M}^{(i)}\| \leq 2\ell$ for all $j \in [n]$. Applying matrix Bernstein inequality, we have that

$$\mathbb{P} \left[\|\widetilde{M} - \mathbb{E}[\widetilde{M}]\| \geq t \right] \leq d \exp \left(\frac{-t^2/2}{2n\ell^2\kappa/d + 4\ell t/3} \right). \quad (132)$$

Therefore, with probability at least $1 - d^{-3}$, we have,

$$\|\widetilde{M} - \mathbb{E}[\widetilde{M}]\| \leq 4\ell \sqrt{\frac{2n\kappa \log d}{d}} + \frac{64\ell \log d}{3} \leq 8\ell \sqrt{\frac{n\kappa \log d}{d}}, \quad (133)$$

where the second inequality follows from the assumption that $n\ell \geq 2^{12}d \log d$.

8.6.2 PROOF OF LEMMA 21

Without loss of generality, assume that $i' < i$, i.e., $\tilde{\theta}_{i'}^* \leq \tilde{\theta}_i^*$. Define Ω such that $\Omega = \{j : j \in S, j \neq i, i'\}$. For any $\beta_1 \in [0, (\ell - 2)/\ell]$, define event E_{β_1} that occurs if in the randomly chosen set S there are at most $\lfloor \ell \beta_1 \rfloor$ items that have preference scores less than $\tilde{\theta}_{i'}^*$, i.e.,

$$E_{\beta_1} \equiv \left\{ \sum_{j \in \Omega} \mathbb{I}_{\{\tilde{\theta}_j^* > \tilde{\theta}_{i'}^*\}} \leq \lfloor \ell \beta_1 \rfloor \right\}. \quad (134)$$

We have,

$$\begin{aligned} & \mathbb{P} \left[\sigma^{-1}(i), \sigma^{-1}(i') > \kappa - \ell \mid i, i' \in S \right] \\ & > \mathbb{P} \left[\sigma^{-1}(i), \sigma^{-1}(i') > \kappa - \ell \mid i, i' \in S; E_{\beta_1} \right] \mathbb{P} \left[E_{\beta_1} \mid i, i' \in S \right] \end{aligned} \quad (135)$$

The following lemma provides a lower bound on $\mathbb{P}[\sigma^{-1}(i), \sigma^{-1}(i') > \kappa - \ell \mid i, i' \in S; E_{\beta_1}]$.

Lemma 22. *Under the hypotheses of Lemma 21,*

$$\mathbb{P} \left[\sigma^{-1}(i), \sigma^{-1}(i') > \kappa - \ell \mid i, i' \in S; E_{\beta_1} \right] \geq \frac{e^{-4b}(1 - \lfloor \ell \beta_1 \rfloor / \ell)^2 \ell^2}{2 \kappa^2}. \quad (136)$$

Next, we provide a lower bound on $\mathbb{P}[E_{\beta_1} \mid i, i' \in S]$. Fix i, i' such that $i, i' \in S$. Selecting a set uniformly at random is probabilistically equivalent to selecting items one at a time uniformly at random without replacement. Without loss of generality, assume that i, i' are the 1st and 2nd pick. Define Bernoulli random variables Y_j for $3 \leq j \leq \kappa$ corresponding to the outcome of the j 'th random pick from the set of $(d - j' - 1)$ items to generate the set Ω such that $Y_j = 1$ if and only if $\tilde{\theta}_j^* > \tilde{\theta}_{i'}^*$.

Recall that $\gamma_{\beta_1} \equiv \bar{d}(\kappa - 2)/(\lfloor \ell \beta_1 \rfloor + 1)(d - 2)$ and $\eta_{\beta_1} \equiv (\lfloor \ell \beta_1 \rfloor + 1)^2/2(\kappa - 2)$. Construct Doob's martingale (Z_2, \dots, Z_κ) from $\{Y_k\}_{3 \leq k \leq \kappa}$ such that $Z_j = \mathbb{E}[\sum_{k=3}^{\kappa} Y_k \mid Y_3, \dots, Y_j]$, for $2 \leq j \leq \kappa$. Observe that, $Z_2 = \mathbb{E}[\sum_{k=3}^{\kappa} Y_k] \leq \frac{(d-2)(\kappa-2)}{d-2} \leq \gamma_{\beta_1}(\lfloor \ell \beta_1 \rfloor + 1)$, where the last inequality follows only if $\tilde{\theta}_{i'}^* > \tilde{\theta}_{i'}^*$.

from the assumption that $i \leq \tilde{d}$. Also, $|Z_{j'} - Z_{j'-1}| \leq 1$ for each j' . Therefore, we have

$$\begin{aligned} \mathbb{P}\left[\sum_{j \in \Omega} \mathbb{1}_{\{\tilde{\theta}_i^* > \tilde{\theta}_{j'}^*\}} \leq \lfloor \ell \beta_1 \rfloor\right] &= \mathbb{P}\left[\sum_{j'=3}^{\kappa} Y_{j'} \leq \lfloor \ell \beta_1 \rfloor\right] \\ &= 1 - \mathbb{P}\left[\sum_{j'=3}^{\kappa} Y_{j'} \geq \lfloor \ell \beta_1 \rfloor + 1\right] \\ &\geq 1 - \mathbb{P}\left[Z_{\kappa-2} - Z_2 \geq (\ell \beta_1 + 1) - \gamma(\lfloor \ell \beta_1 \rfloor + 1)\right] \\ &\geq 1 - \exp\left(-\frac{(\lfloor \ell \beta_1 \rfloor + 1)^2(1 - \gamma_1)^2}{2(\kappa - 2)}\right) \\ &= 1 - \exp\left(-\eta_{\beta_1}(1 - \gamma_{\beta_1})^2\right), \end{aligned} \quad (137)$$

where the inequality follows from the Azuma-Hoeffding bound. Since, the above inequality is true for any fixed $i, i' \in S$, for random indices i, i' we have $\mathbb{P}[E_{\beta_1} \mid i, i' \in S] \geq 1 - \exp(-\eta_{\beta_1}(1 - \gamma_{\beta_1})^2)$. Claim (127) follows by combining Equations (135), (136) and (137).

8.6.3 PROOF OF LEMMA 22

Without loss of generality, assume that $i' < i$, i.e., $\tilde{\theta}_{i'}^* \leq \tilde{\theta}_i^*$. Define $\Omega = \{j : j \in S, j \neq i, i'\}$, and event $E_{\beta_1} = \{i, i' \in S; \sum_{j \in \Omega} \mathbb{1}_{\{\tilde{\theta}_i^* > \tilde{\theta}_j^*\}} \leq \lfloor \ell \beta_1 \rfloor\}$. Since set S is chosen randomly, i, i' and $j \in \Omega$ are random. Throughout this section, we condition on the random indices i, i' and the set Ω such that event E_{β_1} holds. To get a lower bound on $\mathbb{P}[\sigma^{-1}(i), \sigma^{-1}(i') > \kappa - \ell]$, define independent exponential random variables $X_j \sim \exp(e^{j'})$ for $j \in S$. Observe that given event E_{β_1} holds, there exists a set $\Omega_1 \subseteq \Omega$ such that

$$\Omega_1 = \left\{j \in S : \tilde{\theta}_i^* \leq \tilde{\theta}_{j'}^*\right\}, \quad (138)$$

and $|\Omega_1| = \kappa - \lfloor \ell \beta_1 \rfloor - 2$. In fact there can be many such sets, and for the purpose of the proof we can choose one such set arbitrarily. Note that $\lfloor \ell \beta_1 \rfloor + 2 \leq \ell$ by assumption on β_1 , so $|\Omega_1| \geq \kappa - \ell$. From the Random Utility Model (RUM) interpretation of the PL model, we know that the PL model is equivalent to ordering the items as per *random cost* of each item drawn from exponential random variable with mean $e^{j'}$. That is, we rank items according to X_j 's such that the lower cost items are ranked higher. From this interpretation, we have that

$$\begin{aligned} \mathbb{P}\left[\sigma^{-1}(i), \sigma^{-1}(i') > \kappa - \ell\right] &= \mathbb{P}\left[\sum_{j \in \Omega} \mathbb{1}_{\{\min\{X_i, X_{i'}\} > X_j\}} \geq \kappa - \ell\right] \\ &> \mathbb{P}\left[\sum_{j' \in \Omega_1} \mathbb{1}_{\{\min\{X_i, X_{i'}\} > X_{j'}\}} \geq \kappa - \ell\right] \end{aligned} \quad (139)$$

The above inequality follows from the fact that $\Omega_1 \subseteq \Omega$ and $|\Omega_1| \geq \kappa - \ell$. It excludes some of the rankings over the items of the set S that constitute the event $\{\sigma^{-1}(i), \sigma^{-1}(i') > \kappa - \ell\}$. Define $\Omega_2 = \{\Omega_1, i, i'\}$. Observe that items i, i' have the least preference scores among all the items in the set Ω_2 . Therefore, the term in Equation (139) is the probability of the least two preference score items in the set Ω_2 , that is of size $(\kappa - \lfloor \ell \beta_1 \rfloor)$, being ranked in bottom $(\ell - \lfloor \ell \beta_1 \rfloor)$ positions.

The following lemma shows that the probability of the least two preference score items in a set being ranked at any two positions is lower bounded by their probability of being ranked at 1st and 2nd position.

Lemma 23. Consider a set of items S and a ranking σ over it. Define $i_{\min} \equiv \arg \min_{i \in S} \theta_i$, $i_{\min_2} \equiv \arg \min_{i \in S \setminus \{i_{\min}\}} \theta_i$. For all $1 \leq i_1, i_2 \leq |S|$, $i_1 \neq i_2$, following holds:

$$\mathbb{P}\left[\sigma^{-1}(i_{\min}) = i_1, \sigma^{-1}(i_{\min_2}) = i_2\right] \geq \mathbb{P}\left[\sigma^{-1}(i_{\min}) = 1, \sigma^{-1}(i_{\min_2}) = 2\right]. \quad (140)$$

Using the fact that $i' = \arg \min_{j \in \Omega_2} \tilde{\theta}_{j'}^*$, $i = \arg \min_{j \in \Omega_2 \setminus \{i'\}} \tilde{\theta}_j^*$, for all $1 \leq i_1, i_2 \leq \kappa - \lfloor \ell \beta_1 \rfloor$, $i_1 \neq i_2$, we have that

$$\mathbb{P}\left[\sigma^{-1}(i') = i_1, \sigma^{-1}(i) = i_2\right] \geq \mathbb{P}\left[\sigma^{-1}(i') = 1, \sigma^{-1}(i) = 2\right] \geq e^{-4b} \frac{1}{\kappa^2}, \quad (141)$$

where the second inequality follows from the definition of the PL model and the fact that $\tilde{\theta}^* \in \tilde{\Omega}_{2b}$. Together with Equation (141) and the fact that there are a total of $(\ell - \lfloor \ell \beta_1 \rfloor)(\ell - \lfloor \ell \beta_1 \rfloor - 1) \geq (\ell - \lfloor \ell \beta_1 \rfloor)^2/2$ pair of positions that i, i' can occupy in order to be ranked in bottom $(\ell - \lfloor \ell \beta_1 \rfloor)$, we have,

$$\mathbb{P}\left[\sigma^{-1}(i), \sigma^{-1}(i') > \kappa - \ell\right] \geq \frac{e^{-4b}(1 - \lfloor \ell \beta_1 \rfloor / \ell)^2 \ell^2}{2 \kappa^2}. \quad (142)$$

Since, the above inequality is true for any fixed i, i' and $j \in \Omega$ such that event E holds, it is true for random indices i, i' and $j \in \Omega$ such that event E holds, hence the claim is proved.

8.6.4 PROOF OF LEMMA 23

Let $\hat{\sigma}$ denote a ranking over the items of the set S and $\mathbb{P}[\hat{\sigma}]$ be the probability of observing $\hat{\sigma}$. Let

$$\hat{\Omega}_1 = \left\{\hat{\sigma} : \hat{\sigma}^{-1}(i_{\min}) = i_1, \hat{\sigma}^{-1}(i_{\min_2}) = i_2\right\} \text{ and } \hat{\Omega}_2 = \left\{\hat{\sigma} : \hat{\sigma}^{-1}(i_{\min}) = 1, \hat{\sigma}^{-1}(i_{\min_2}) = 2\right\}. \quad (143)$$

Now, take any ranking $\hat{\sigma} \in \hat{\Omega}_1$ and construct another ranking $\tilde{\sigma}$ from $\hat{\sigma}$ as following. If $i_1 = 2, i_2 = 1$, then swap the items at i_1 -th and i_2 -th position in ranking $\hat{\sigma}$ to get $\tilde{\sigma}$. Else, if $i_1 < i_2$, then first: swap items at i_1 -th position and 1st position, and second: swap items at i_2 -th position and 2nd position, to get $\tilde{\sigma}$; if $i_2 < i_1$, then first: swap items at i_2 -th position and 2nd position, and second: swap items at i_1 -th position and 1st position, to get $\tilde{\sigma}$.

Observe that $\mathbb{P}[\tilde{\sigma}] \leq \mathbb{P}[\hat{\sigma}]$ and $\tilde{\sigma}_i^* \in \hat{\Omega}_2$. Moreover, such a construction gives a bijective mapping between $\hat{\Omega}_1$ and $\hat{\Omega}_2$. Hence, the claim is proved.

Acknowledgements

The authors thank the anonymous reviewers for their constructive feedback. This work was partially supported by National Science Foundation Grants MES-1450848, CNS-1527754, and CCF-1553452.

References

- N. Ailon. Active learning ranking from pairwise preferences with almost optimal query complexity. In *Advances in Neural Information Processing Systems*, pages 810–818, 2011.

- A. Ammar and D. Shah. Ranking: Compare, don't score. In *Communication, Control, and Computing (Allerton), 2011 49th Annual Allerton Conference on*, pages 776–783. IEEE, 2011.
- A. Ammar, S. Oh, D. Shah, and L. Voloch. What's your choice? learning the mixed multi-nomial logit model. In *Proceedings of the ACM SIGMETRICS/international conference on Measurement and modeling of computer systems*, 2014.
- H. Azari Soufiani, D. C. Parkes, and L. Xia. Random utility theory for social choice. In *NIPS*, pages 126–134, 2012.
- H. Azari Soufiani, W. Chen, D. C Parkes, and L. Xia. Generalized method-of-moments for rank aggregation. In *Advances in Neural Information Processing Systems 26*, pages 2706–2714, 2013.
- H. Azari Soufiani, D. Parkes, and L. Xia. Computing parametric ranking models via rank-breaking. In *Proceedings of The 31st International Conference on Machine Learning*, pages 360–368, 2014.
- M. E. Ben-Akiva and S. R. Lerman. *Discrete choice analysis: theory and application to travel demand*, volume 9. MIT press, 1985.
- J. Blanchet, G. Gallego, and V. Goyal. A Markov chain approximation to choice modeling. In *EC*, pages 103–104, 2013.
- M. Braverman and E. Mossel. Sorting from noisy information. *arXiv preprint arXiv:0910.1191*, 2009.
- Y. Chen and C. Suh. Spectral mle: Top- k rank aggregation from pairwise comparisons. *arXiv preprint arXiv:1504.07218*, 2015.
- C. Cortes, M. Mohri, and A. Rastogi. Magnitude-preserving ranking algorithms. In *Proceedings of the 24th international conference on Machine learning*, pages 169–176. ACM, 2007.
- J. C. de Borda. Mémoire sur les élections au scrutin. 1781.
- N. De Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. L'imprimerie royale, 1785.
- P. Diaconis. A generalization of spectral analysis with application to ranked data. *The Annals of Statistics*, pages 949–979, 1989.
- W. Ding, P. Ishwar, and V. Saligrama. Learning mixed membership mallows models from pairwise comparisons. *arXiv preprint arXiv:1504.00757*, 2015.
- J. C. Duchin, L. Mackey, and M. I. Jordan. On the consistency of ranking algorithms. In *Proceedings of the ICMML Conference*, Haifa, Israel, June 2010.
- C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th international conference on World Wide Web*, pages 613–622. ACM, 2001.
- O. Dykstra. Rank analysis of incomplete block designs: A method of paired comparisons employing unequal repetitions on pairs. *Biometrics*, 16(2):176–188, 1960.
- V. F. Farias, S. Jagabathula, and D. Shah. A data-driven approach to modeling choice. In *NIPS*, pages 504–512, 2009.
- V. F. Farias, S. Jagabathula, and D. Shah. A nonparametric approach to modeling choice with limited data. *Management Science*, 59(2):305–322, 2013.
- J. B. Feldman and H. Topaloglu. Revenue management under the markov chain choice model. 2014.
- L. R. Ford Jr. Solution of a ranking problem from binary comparisons. *The American Mathematical Monthly*, 64(8):28–33, 1957.
- K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, 2001.
- Ryan G. Gomes, Peter Welinder, Andreas Krause, and Pietro Perona. Crowdclustering. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 558–566. Curran Associates, Inc., 2011. URL <http://papers.nips.cc/paper/4187-crowdclustering.pdf>.
- P. M. Guadagni and J. D. Little. A logit model of brand choice calibrated on scanner data. *Marketing science*, 2(3):203–238, 1983.
- B. Hajek, S. Oh, and J. Xu. Minimax-optimal inference from partial rankings. In *Advances in Neural Information Processing Systems 27*, pages 1475–1483, 2014.
- T. P. Hayes. A large-deviation inequality for vector-valued martingales. *Combinatorics, Probability and Computing*, 2005.
- D. R. Hunter. Mm algorithms for generalized bradley-terry models. *Annals of Statistics*, pages 384–406, 2004.
- K. G. Jamieson and R. Nowak. Active ranking using pairwise comparisons. In *Advances in Neural Information Processing Systems*, pages 2240–2248, 2011.
- T. Kaminishima. Nontonic collaborative filtering: recommendation based on order responses. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 583–588. ACM, 2003.
- T. Le Van, M. van Leeuwen, S. Nijssen, and L. De Raedt. Rank matrix factorisation. In *Advances in Knowledge Discovery and Data Mining*, pages 734–746. Springer, 2015.
- G. Lebanon and Y. Mao. Non-parametric modeling of partially ranked data. In *Advances in neural information processing systems*, pages 857–864, 2007.
- T. Lu and C. Boutilier. Learning mallows models with pairwise preferences. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 145–152, 2011a.
- T. Lu and C. Boutilier. Budgeted social choice: From consensus to personalized decision making. In *IJCAI*, volume 11, pages 280–286, 2011b.

- Y. Lu and S. Negahban. Individualized rank aggregation using nuclear norm regularization. *arXiv preprint arXiv:1410.0860*, 2014.
- J. Lundell. Second report of the irish commission on electronic voting. *Voting Matters*, 23:13–17, 2007.
- L. Maystre and M. Grossglauser. Fast and accurate inference of plackett-luce models. In *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, 2015a.
- L. Maystre and M. Grossglauser. Robust active ranking from sparse noisy comparisons. *arXiv preprint arXiv:1502.05556*, 2015b.
- D. McFadden. Conditional logit analysis of qualitative choice behavior. *Frontiers in Econometrics*, pages 105–142, 1973.
- D. McFadden. Econometric models for probabilistic choice among products. *Journal of Business*, 53(3):S13–S29, 1980.
- D. McFadden and K. Train. Mixed mnl models for discrete response. *Journal of applied Econometrics*, 15(5):447–470, 2000.
- T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- K. Miyahara and M. J. Pazzani. Collaborative filtering with the simple bayesian classifier. In *PRICAI 2000 Topics in Artificial Intelligence*, pages 679–689. Springer, 2000.
- S. Negahban, S. Oh, and D. Shah. Iterative ranking from pair-wise comparisons. In *NIPS*, pages 2483–2491, 2012.
- S. Negahban, S. Oh, and D. Shah. Rank centrality: Ranking from pair-wise comparisons. preprint arXiv:1209.1688, 2014.
- S. Oh and D. Shah. Learning mixed multinomial logit model from ordinal data. In *Advances in Neural Information Processing Systems*, pages 595–603, 2014.
- S. Oh, K. K. Thekumparampil, and J. Xu. Collaboratively learning preferences from ordinal data. In *Advances in Neural Information Processing Systems 28*, pages 1900–1908, 2015.
- D. Park, J. Neeman, J. Zhang, S. Saughavi, and I. S. Dhillon. Preference completion: Large-scale collaborative ranking from pairwise comparisons. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 1907–1916, 2015.
- H. Polat and W. Du. Svd-based collaborative filtering with privacy. In *Proceedings of the 2005 ACM symposium on Applied computing*, pages 791–795. ACM, 2005.
- A. Rajkumar and S. Agarwal. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In *Proceedings of The 31st International Conference on Machine Learning*, pages 118–126, 2014.
- P. Ray. Independence of irrelevant alternatives. *Econometrica: Journal of the Econometric Society*, pages 987–991, 1973.
- N. B. Shah and M. J. Wainwright. Simple, robust and optimal ranking from pairwise comparisons. *arXiv preprint arXiv:1512.08949*, 2015.
- N. B. Shah, S. Balakrishnan, J. Bradley, A. Parekh, K. Ramchandran, and M. J. Wainwright. Estimation from pairwise comparisons: Sharp minimax bounds with topology dependence. *arXiv preprint arXiv:1505.01462*, 2015a.
- N. B. Shah, S. Balakrishnan, A. Guntuboyina, and M. J. Wainwright. Stochastically transitive models for pairwise comparisons: Statistical and computational issues. *arXiv preprint arXiv:1510.05610*, 2015b.
- P. Sham and D. Curtis. An extended transmission/disequilibrium test (tdt) for multi-allele marker loci. *Annals of human genetics*, 59(3):323–336, 1995.
- J. Walker and M. Ben-Akiva. Generalized random utility model. *Mathematical Social Sciences*, 43(3):303–343, 2002.
- R. Wu, J. Xu, R. Srikant, L. Massoulié, M. Lelarge, and B. Hajek. Clustering and inference from pairwise comparisons. *arXiv preprint arXiv:1502.04631*, 2015.
- J. Yi, R. Jin, S. Jain, and A. Jain. Inferring users’ preferences from crowdsourced pairwise comparisons: A matrix completion approach. In *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

Optimal Learning Rates for Localized SVMs

Mona Meister

*Corporate Research
Robert Bosch GmbH
70465 Stuttgart, Germany*

MONA.MEISTER@DE.BOSCH.COM

Ingo Steinwart

*Institute for Stochastics and Applications
University of Stuttgart
70569 Stuttgart, Germany*

INGO.STEINWART@MATHematik.UNI-STUTTGART.DE

Editor: Sara van de Geer

Abstract

One of the limiting factors of using support vector machines (SVMs) in large scale applications are their super-linear computational requirements in terms of the number of training samples. To address this issue, several approaches that train SVMs on many small chunks separately have been proposed in the literature. With the exception of random chunks, which is also known as divide-and-conquer kernel ridge regression, however, these approaches have only been empirically investigated. In this work we investigate a spatially oriented method to generate the chunks. For the resulting localized SVM that uses Gaussian kernels and the least squares loss we derive an oracle inequality, which in turn is used to deduce learning rates that are essentially minimax optimal under some standard smoothness assumptions on the regression function. In addition, we derive local learning rates that are based on the local smoothness of the regression function. We further introduce a data-dependent parameter selection method for our local SVM approach and show that this method achieves the same almost optimal learning rates. Finally, we present a few larger scale experiments for our localized SVM showing that it achieves essentially the same test error as a global SVM for a fraction of the computational requirements. In addition, it turns out that the computational requirements for the local SVMs are similar to those of a vanilla random chunk approach, while the achieved test errors are significantly better.

Keywords: least squares regression, support vector machines, localization

1. Introduction

Based on a training set $D := ((x_1, y_1), \dots, (x_n, y_n))$ of i.i.d. input/output observations drawn from an unknown distribution P on $X \times Y$, where $X \subset \mathbb{R}^d$ and $Y \subset \mathbb{R}$, the goal of non-parametric least squares regression is to find a function $f_D : X \rightarrow \mathbb{R}$ that is a good estimate of the unknown conditional mean $f^*(x) := \mathbb{E}(Y|x)$, $x \in X$. For this classical estimation problem various methods have been proposed and studied in the literature, see e.g., (Simonoff, 1996) and the book (Györfi et al., 2002) for detailed accounts.

In this paper, we consider kernel-based regularized empirical risk minimizers, also known as support vector machines (SVMs), which solve the regularized problem

$$f_{D,\lambda} \in \arg \min_{f \in H} \lambda \|f\|_H^2 + \mathcal{R}_{L,D}(f). \quad (1)$$

Here, $\lambda > 0$ is a fixed real number and H is a reproducing kernel Hilbert space (RKHS) over X with reproducing kernel $k : X \times X \rightarrow \mathbb{R}$, see e.g., (Aronszajn, 1950; Berlinet and Thomas-Agnan, 2004; Steinwart and Christmann, 2008). The function $L : X \times Y \times \mathbb{R} \rightarrow [0, \infty)$ is a loss function, where in the following we either consider the least squares loss $L_{LS} : Y \times \mathbb{R} \rightarrow [0, \infty)$ defined by $(y, t) \mapsto (y - t)^2$, or variants of it that may depend on $x \in X$. Besides, $\mathcal{R}_{L,D}(f)$ denotes the empirical risk of a function $f : X \rightarrow \mathbb{R}$, that is

$$\mathcal{R}_{L,D}(f) = \frac{1}{n} \sum_{i=1}^n L(x_i, y_i, f(x_i)),$$

where D is the empirical measure associated to the data D defined by $D := \frac{1}{n} \sum_{i=1}^n \delta_{(x_i, y_i)}$ with Dirac measure $\delta_{(x_i, y_i)}$ at (x_i, y_i) . Recall that the empirical SVM solution $f_{D,\lambda}$ exists and is unique (cf. Steinwart and Christmann, 2008, Theorem 5.5) whenever the loss L is convex in its last argument, which is true for the least squares loss and its variants that will be considered later on. Moreover, an SVM is L -risk consistent under a few assumptions on the RKHS H and the regularization parameter λ , see (Steinwart and Christmann, 2008, Section 6.4) for more details.

An essential theoretical task, which has attracted many considerations, is the investigation of learning rates for SVMs. For example, such rates for SVMs using the least squares loss and generic kernels can be found in (Cucker and Smale, 2002; De Vito et al., 2005; Smale and Zhou, 2007; Caponnetto and De Vito, 2007; Mendelson and Neuman, 2010; Steinwart et al., 2009) and the references mentioned therein. At this point, we do not want to take a closer look at these results, instead we relegate to (Eberts and Steinwart, 2013), where a detailed discussion can be found. More important for our purposes is the fact that Eberts and Steinwart (2011, 2013) establish (essentially) asymptotically optimal learning rates for least squares SVMs (LS-SVMs) using Gaussian RBF kernels. More precisely, for a domain $X \subset B_{\ell_2}^d$, $Y := [-M, M]$ with $M > 0$, a distribution P on $X \times Y$ such that P_X has a bounded Lebesgue density on X , and for f^* contained in the Sobolev space $W_2^\alpha(P_X)$, $\alpha \in \mathbb{N}$, or in the Besov-like space $B_{2,\infty}^{\alpha}(P_X)$, $\alpha \geq 1$, respectively, the LS-SVM using Gaussian kernels learns for all $\xi > 0$ with rate $n^{-\frac{2\alpha}{2\alpha+7\xi}}$ with a high probability. In other words, it learns at least with a rate that is arbitrarily close to the optimal learning rate.

Although these rates are essentially asymptotically optimal, they depend on the order of smoothness of the regression function on the *entire* input space X . That is, if the regression function f^* is on some area of X smoother than on another area, the learning rate is determined by the part of X , where the regression function f^* is least smooth. In contrast to this, it would be desirable to achieve a learning rate on every region of X that corresponds with the order of smoothness of f^* on this region. Therefore, one of our goals of this paper is to modify the standard SVM approach such that we achieve local learning rates that are asymptotically optimal.

Our technique to achieve such local learning rates is a special data splitting approach, which first creates a geometrically well-behaved partition of the input space X and then finds a separate SVM on each of the resulting cells with the help of the training samples that fall into these cells. Recall that various other *local* splitting approaches have already been extensively investigated in the literature, but mostly to speed-up the training time, see for instance, the early works (Bottou and Vapnik, 1992; Vapnik and Bottou, 1993).

Here the basic idea of most other local approaches is to *a)* split the training data and just consider a few examples near a testing sample, *b)* train on this small subset of the training data, and *c)* use the solution for a prediction w.r.t. the test sample. Here, many up-to-date investigations use SVMs to train on the local data set but, yet there are different ways to split the whole training data set into smaller local sets. For example, Chang et al. (2010); Wu et al. (1999); Bennett and Blhe (1998) use decision trees while in (Hable, 2013; Segata and Blanzieri, 2010, 2008; Blanzieri and Melgani, 2008; Blanzieri and Bryl, 2007a,b; Zhang et al., 2006) local subsets are built considering k nearest neighbors. The latter approaches further vary; for example, Zhang et al. (2006); Blanzieri and Bryl (2007a); Hable (2013) consider different metrics w.r.t. the input space whereas Segata and Blanzieri (2008); Blanzieri and Melgani (2008); Blanzieri and Bryl (2007b) consider metrics w.r.t. the feature space. Nonetheless, the basic idea of all these articles is that an SVM problem based on k training samples is solved for *each* test sample. Another approach using k nearest neighbors is investigated in (Segata and Blanzieri, 2010). Here, k -neighborhoods consisting of training samples and collectively covering the training data set are constructed and an SVM is calculated on each neighborhood. The prediction for a test sample is then made according to the nearest training sample that is a center of a k -neighborhood. As for the other nearest neighbor approaches, however, the results are mainly experimental. An exception to this rule is (Hable, 2013), where universal consistency for localized versions of SVMs, or more precisely, a large class of regularized kernel methods, is proven. Another article presenting theoretical results for localized versions of learning methods is (Zakai and Ritov, 2009). Here, the authors show that a consistent learning method behaves locally, i.e., the prediction is essentially influenced by close by samples. However, this result is based on a localization technique considering only training samples contained in a neighborhood with a fixed radius and center x when an estimate in x is sought. Probably closest to our approach is the one examined in (Cheng et al., 2010) and (Cheng et al., 2007), where the training data is split into clusters and then an SVM is trained on each cluster. However, the presented results are again only of experimental character.

Unlike in the papers mentioned above, our main goal is to theoretically investigate local SVMs based on local splitting. Namely, we establish both global and local learning rates for our local splitting approach (VP-SVM) that do match the best existing and essentially optimal rates for global SVMs derived by Eberts and Steinwart (2013). In addition, we show that these rates can be obtained without knowing characteristics of P by a simple and well-known hold-out technique. Furthermore, we empirically compare our VP-SVM to another data splitting approach known as random chunking (RC-SVM) or divide-and-conquer kernel ridge regression for which learning rates, at least for generic kernels, have been recently established by Zhang et al. (2015); Lin et al. (2016). In these experiments it turns out that for splittings that lead to comparable training times, our VP-SVM has a significantly smaller test error than RC-SVMs.

Investigating other speed-up schemes for SVMs theoretically has been in the focus of research in the last few years. For example, Zhang et al. (2015); Lin et al. (2016) established optimal learning rates in expectation for RC-SVMs under the assumption that the conditional mean f^* is contained in the used RKHS, or in the image of a fractional integral operator, respectively. Although these results are very interesting they are not very useful for SVMs with Gaussian kernels, since for these kernels the imposed assumptions on f^*

imply $f^* \in C^\infty$, which is usually considered to be too restrictive. For a similar reason the results by Rudi et al. (2015) for the popular Nyström method require too restrictive assumptions when applied to SVMs with Gaussian kernels. On a side note, we like to mention that this difference between generic kernels on the one hand and Gaussian kernels on the other hand already appears for the standard global SVMs. Indeed, in the generic case, one usually addresses the approximation error by assuming the conditional mean to be contained in the image of a fractional integral operator, which can in turn be identified as an interpolation space of the real method, see (Steinwart and Scovel, 2012). For certain kernels, the classical theory of interpolation spaces then identifies the considered interpolation spaces as Besov spaces, so that the approximation error assumption has a clear intuitive meaning. On the other hand, for Gaussian kernels with fixed width it has been shown by Smale and Zhou (2003) that their interpolation spaces consist of C^∞ -functions, so that the generic theory would again lead to a too restrictive approximation error assumption. To address this issue, one considers widths that change with the sample size. However, to make this approach successful, one requires both a manual estimation of the approximation error, see (Eberts and Steinwart, 2011), and eigenvalue/entropy number bounds that do depend on the kernel width. For these reasons, learning rates for SVMs with Gaussian kernels under realistic assumptions are, in general, harder to obtain. Nonetheless, they are important, since in practice, Gaussian kernels are by far the most often used kernels.

The rest of this paper is organized as follows: In Section 2 we describe our splitting approach in detail. Section 3 then presents some theoretical results on RKHSs that enable the analysis of our method. After that, Section 4 contains the main results, namely an oracle inequality and learning rates for our localized SVM method. Moreover, a data-dependent parameter selection method is studied that induces the same rates. Section 5 then presents some experimental results w.r.t. the localized SVM technique. Finally, Section 6 collects the proofs for the results of the earlier sections as well as some necessary and important ancillary findings.

2. Description of the Localized SVM Approach

In this section, we introduce some general notations and assumptions. Based on the latter we modify the standard SVM approach. Let us start with the probability measure P on $X \times Y$, where $X \subset \mathbb{R}^d$ is non-empty, $Y := [-M, M]$ for some $M > 0$, and P_X is the marginal distribution of X . Depending on the learning target one chooses a loss function L , i.e., a function $L : X \times Y \times \mathbb{R} \rightarrow [0, \infty)$ that is measurable. Then, for a measurable function $f : X \rightarrow \mathbb{R}$, the L -risk is defined by

$$\mathcal{R}_{L,P}(f) = \int_{X \times Y} L(x, y, f(x)) dP(x, y)$$

and the optimal L -risk, called the Bayes risk with respect to P and L , is given by

$$\mathcal{R}_{L,P}^*(f) := \inf \{ \mathcal{R}_{L,P}(f) \mid f : X \rightarrow \mathbb{R} \text{ measurable} \}.$$

A measurable function $f_{L,P}^* : X \rightarrow \mathbb{R}$ with $\mathcal{R}_{L,P}(f_{L,P}^*) = \mathcal{R}_{L,P}^*$ is called a Bayes decision function. For the commonly used losses such as the least squares loss treated in Section 4 the Bayes decision function $f_{L,P}^*$ is P_X -almost surely $[-M, M]$ -valued, since $Y = [-M, M]$.

In this case, it seems obvious to consider estimators with values in $[-M, M]$ on X . To this end, we introduce the concept of clipping the decision function. Let \hat{f} be the clipped value of some $t \in \mathbb{R}$ at $\pm M$ defined by

$$\hat{f} := \begin{cases} -M & \text{if } t < -M \\ t & \text{if } t \in [-M, M] \\ M & \text{if } t > M. \end{cases}$$

Then, a loss is called clippable at $M > 0$ if, for all $(x, y, t) \in X \times Y \times \mathbb{R}$, we have

$$L(x, y, \hat{f}) \leq L(x, y, t).$$

Obviously, the latter implies $\mathcal{R}_{L,P}(\hat{f}) \leq \mathcal{R}_{L,P}(f)$ for all $f : X \rightarrow \mathbb{R}$. In other words, restricting the decision function to the interval $[-M, M]$ containing our labels cannot worsen the risk, in fact, clipping this function typically reduces the risk. Hence, we consider the clipped version \hat{f}_D of the decision function as well as the risk $\mathcal{R}_{L,P}(\hat{f}_D)$ instead of the risk $\mathcal{R}_{L,P}(f_D)$ of the unclipped decision function. Note that this clipping idea does *not* change the required solver since it is performed *after* the training phase.

To modify the standard SVM approach (1), we assume that $(A_j)_{j=1,\dots,m}$ is a partition of X such that all its cells have non-empty interior, that is $A_j \neq \emptyset$ for every $j \in \{1, \dots, m\}$. Now, the basic idea of our approach is to consider for each cell of the partition an individual SVM. To describe this approach in a mathematically rigorous way, we have to introduce some more definitions and notations. Let us begin with the index set

$$I_j := \{i \in \{1, \dots, n\} : x_i \in A_j\}, \quad j = 1, \dots, m,$$

indicating the samples of D contained in A_j , as well as the corresponding data set

$$D_j := \{(x_i, y_i) \in D : i \in I_j\}, \quad j = 1, \dots, m.$$

Moreover, for every $j \in \{1, \dots, m\}$, we define a (local) loss $L_j : X \times Y \times \mathbb{R} \rightarrow [0, \infty)$ by

$$L_j(x, y, t) := \mathbb{1}_{A_j}(x)L(x, y, t),$$

where $L : X \times Y \times \mathbb{R} \rightarrow [0, \infty)$ is the loss that corresponds to our learning problem at hand. We further assume that H_j is an RKHS over A_j with kernel $k_j : A_j \times A_j \rightarrow \mathbb{R}$. Here, every function $f \in H_j$ is only defined on A_j even though a function $f_D : X \rightarrow \mathbb{R}$ is finally sought. To this end, for $f \in H_j$, we define the zero-extension $\hat{f} : X \rightarrow \mathbb{R}$ by

$$\hat{f}(x) := \begin{cases} f(x), & x \in A_j, \\ 0, & x \notin A_j. \end{cases}$$

Then, the space $\hat{H}_j := \{\hat{f} : f \in H_j\}$ equipped with the norm

$$\|\hat{f}\|_{\hat{H}_j} := \|f\|_{H_j}, \quad \hat{f} \in \hat{H}_j,$$

is an RKHS on X (cf. Lemma 2), which is isometrically isomorphic to H_j . With these preparations we can now formulate our local SVM approach. To this end, for every $j \in \{1, \dots, m\}$, we consider the local SVM optimization problem

$$f_{D_j, \lambda_j} = \arg \min_{f \in H_j} \lambda_j \|f\|_{H_j}^2 + \frac{1}{n} \sum_{i=1}^n L_j(x_i, y_i, \hat{f}(x_i)), \quad (2)$$

where $\lambda_j > 0$ for every $j \in \{1, \dots, m\}$. Based on these empirical SVM solutions, we then define the decision function $f_{D, \lambda} : X \rightarrow \mathbb{R}$ by

$$f_{D, \lambda}(x) := \sum_{j=1}^m f_{D_j, \lambda_j}(x) = \sum_{j=1}^m \mathbb{1}_{A_j}(x) f_{D_j, \lambda_j}(x), \quad (3)$$

where $\lambda := (\lambda_1, \dots, \lambda_m)$. Since all f_{D_j, λ_j} in (2) are usual empirical SVM solutions the common properties hold. Moreover, for arbitrary $j \in \{1, \dots, m\}$, $f_{D_j, \lambda_j}(x_i) = 0$ if $x_i \notin A_j$ for all $i \in \{1, \dots, n\}$. Furthermore, note that the SVM optimization problem (2) equals the SVM optimization problem (1) using H_j , D_j , and the regularization parameter $\lambda_j := \frac{n}{|I_j|} \lambda_j$. That is, f_{D_j, λ_j} as in (2) and $h_{D_j, \tilde{\lambda}_j} := \arg \min_{f \in H_j} \tilde{\lambda}_j \|f\|_{H_j}^2 + \mathcal{R}_{L, P, D_j}(f)$ coincide on A_j . Besides, it is easy to show that, whenever a Bayes decision function $f_{L, P}^*$ w.r.t. P and L exists, it additionally is a Bayes decision function w.r.t. P and L_j .

Let us now briefly discuss the required computing time of our modified SVM. To this end, recall that the costs for solving an usual SVM problem are $\mathcal{O}(n^q)$ where $q \in [2, 3]$. For the new approach we consider m working sets of size n_1, \dots, n_m where for simplicity we assume $n_i \approx \frac{n}{m}$ for all $i \in \{1, \dots, m\}$. Then for each working set an usual SVM problem has to be solved such that, altogether, the modified SVM induces a computational cost of $\mathcal{O}(m(\frac{n}{m})^q)$. Therefore, if $m \approx n^\beta$ for some $\beta > 0$, then our approach is computationally cheaper than a traditional SVM. Note that our strategy using a partition of the input space is a typical way to speed-up SVMs. Other techniques that possess similar properties are, e.g., applied in the articles cited in the introduction. Besides, we refer to (Tsang et al., 2007) and (Tsang et al., 2005) using enclosing ball problems to solve an SVM, to (Graf et al., 2005) presenting a model of multiple filtering SVMs and to (Collobert et al., 2001) investigating a mixture of SVMs based on several subsets of the training set.

To describe the above SVM approach $(A_j)_{j=1,\dots,m}$ only has to be some partition of X . However, for the theoretical investigations concerning learning rates of our new approach, we have to further specify the partition. To this end, we denote the closed unit ball of the d -dimensional Euclidean space \mathbb{R}^d by $B_{\mathbb{R}^d}$ and we define balls B_1, \dots, B_m with radius $r > 0$ and mutually distinct centers $z_1, \dots, z_m \in B_{\mathbb{R}^d}$ by

$$B_j := B_r(z_j) := \{x \in \mathbb{R}^d : \|x - z_j\|_2 \leq r\}, \quad j \in \{1, \dots, m\}, \quad (4)$$

where $\|\cdot\|_2$ is the Euclidean norm in \mathbb{R}^d . Moreover, we choose r and z_1, \dots, z_m such that

$$B_{\mathbb{R}^d} \subset \bigcup_{j=1}^m B_j,$$

i.e., such that the balls B_1, \dots, B_m cover $B_{\rho_d^c}$ and, simultaneously, any non-empty set $X \subset B_{\rho_d^c}$ (cf. Figure 1). The following well-known lemma relates the radius of such a cover with the number of centers.

Lemma 1 For all $c > 0$ and $r \in (0, c]$, there exist balls $(B_r(z_j))_{j=1, \dots, m}$ with radius r and centers $z_1, \dots, z_m \in cB_{\rho_d^c}$ such that $\bigcup_{j=1}^m B_r(z_j)$ covers $cB_{\rho_d^c}$ and $r \leq 3cm^{-\frac{1}{d}}$.

For simplicity of notation, we assume in the following that $X \subset B_{\rho_d^c}$. Thus, according to Lemma 1, there exists a cover $(B_j)_{j=1, \dots, m}$ of X with

$$r \leq 3m^{-\frac{1}{d}}. \quad (5)$$

Let us finally specify the partition $(A_j)_{j=1, \dots, m}$ of X by the following assumption.

(A) Let $r \in (0, 1]$ and $(A'_j)_{j=1, \dots, \tilde{m}}$ be a partition of $B_{\rho_d^c}$ such that $A'_j \neq \emptyset$ as well as $\overline{A'_j} = \overline{A'_j}$ for every $j \in \{1, \dots, \tilde{m}\}$ and such that there exist balls $B_j := B_r(z_j) \supset A'_j$ with radius r and mutually distinct centers $z_1, \dots, z_{\tilde{m}} \in B_{\rho_d^c}$ satisfying (5). In addition, assume that X is a non-empty, closed subset of $B_{\rho_d^c}$ satisfying $\overline{X} = X$. W.l.o.g. we assume that, for some $m \leq \tilde{m}$, $A'_j \cap X \neq \emptyset$ for all $j \in \{1, \dots, m\}$ and $A'_j \cap X = \emptyset$ for all $j \in \{m+1, \dots, \tilde{m}\}$. Then we define $A''_j := A'_j \cap X$ for all $j \in \{1, \dots, m\}$ and assume that $(A_j)_{j=1, \dots, m}$ is a partition of X satisfying $A''_j \subset A_j \subset \overline{A''_j}$.

Note that the partition $(A_j)_{j=1, \dots, m}$ of X in Assumption (A) satisfies, for every $j \in \{1, \dots, m\}$, $A_j \subset B_j$ for B_j as in (A) and $\dot{A}_j \neq \emptyset$, where the latter is shown in Lemma 8 in the Appendix. Obviously, for the partition $(A_j)_{j=1, \dots, m}$, r and m fulfill (5).

In Assumption (A) $(A'_j)_{j=1, \dots, \tilde{m}}$ is a partition of $B_{\rho_d^c}$ from which we build a partition $(A_j)_{j=1, \dots, m}$ of $X \subset B_{\rho_d^c}$. However, for the construction of our local SVM approach and the proofs of the belonging learning rates, it will be negligible whether we first consider a partition $(A'_j)_{j=1, \dots, \tilde{m}}$ of $B_{\rho_d^c}$ or only a partition $(A_j)_{j=1, \dots, m}$ of X , since the cells $A''_{m+1}, \dots, A''_{\tilde{m}}$ which are removed, have zero mass w.r.t. the marginal distribution P_X of X if $P_X(\partial X) = 0$. In the remaining sections we will frequently refer to Assumption (A). Thus, let us illustrate by the following example that (A) is indeed a natural assumption.

Example 1 For some $r \in (0, 1]$, let us consider an r -net z_1, \dots, z_m of $B_{\rho_d^c}$, where z_1, \dots, z_m are mutually distinct. Moreover, we assume that $X \subset B_{\rho_d^c}$ satisfies $\overline{X} = X$. Based on the r -net z_1, \dots, z_m , a Voronoi partition $(A_j)_{j=1, \dots, m}$ of X is defined by

$$A_j := \left\{ x \in X : \min_{k \in \{1, \dots, m\}} \|x - z_k\|_2 = j \right\}, \quad (6)$$

cf. Figure 2. That is, A_j contains all $x \in X$ such that the center z_j is the nearest center to x , and in the case of ties the center with the smallest index is taken. Obviously, $(A_j)_{j=1, \dots, m}$ is a partition of X with $A_j \neq \emptyset$ and $A_j \subset B_k(z_j)$ for all $j \in \{1, \dots, m\}$, and hence it satisfies condition (A), i.e. r and m fulfill (5).

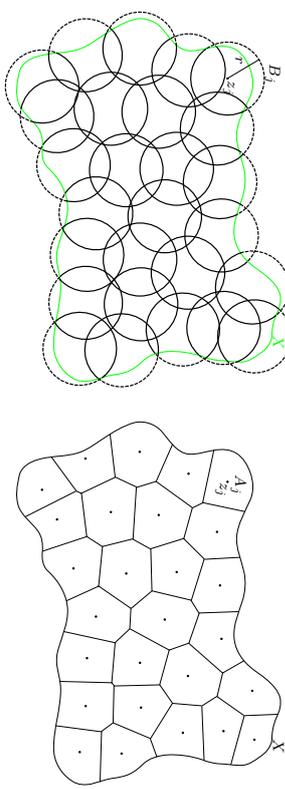


Figure 1: Cover $(B_j)_{j=1, \dots, m}$ of X , where B_1, \dots, B_m are balls with radius r and centers z_j ($j = 1, \dots, m$).

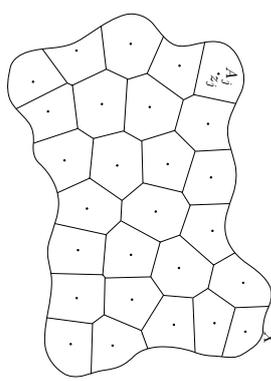


Figure 2: Voronoi partition $(A_j)_{j=1, \dots, m}$ of X defined by (6), where $A_j \subset B_j$ for every $j \in \{1, \dots, m\}$.

Motivated by Example 1, we call the learning method producing \hat{f}_D given by (3) a *Voronoi partition support vector machine*, in short VP-SVM. Despite this name, however, we just take a partition $(A_j)_{j=1, \dots, m}$ satisfying (A) as basis here instead of requesting $(A_j)_{j=1, \dots, m}$ to be a Voronoi partition.

Recall that our goal is to derive not only global but also local learning rates for this VP-SVM approach. To this end, we additionally consider a $T \subset X$ with $P_X(T) > 0$. Then we examine the learning rate of the VP-SVM on this subset T of X . To formalize this, it is necessary to introduce some basic notations related to T . Let us define the index set J_T by

$$J_T := \{j \in \{1, \dots, m\} : A_j \cap T \neq \emptyset\} \quad (7)$$

specifying every set A_j that has at least one common point with T . Note that, for every non-empty set $T \subset X$, the index set J_T is also non-empty, i.e., $|J_T| \geq 1$. Besides, deriving local rates on T requires us to investigate the excess risk of the VP-SVM with respect to the distribution P and the loss $L_T : X \times Y \times \mathbb{R} \rightarrow [0, \infty)$ defined by

$$L_T(x, y, t) := \mathbb{1}_T(x)L(x, y, t). \quad (8)$$

However, to manage the analysis we additionally need the loss $L_{J_T} : X \times Y \times \mathbb{R} \rightarrow [0, \infty)$ given by

$$L_{J_T}(x, y, t) := \mathbb{1}_{\bigcup_{j \in J_T} A_j}(x)L(x, y, t), \quad (9)$$

which may only be nonzero, if x is contained in some set A_j with $j \in J_T$. Note that the risks $\mathcal{R}_{L_T, P}(f)$ and $\mathcal{R}_{L_{J_T}, P}(f)$ quantify the quality of some function f just on T and

$$A_T := \bigcup_{j \in J_T} A_j \supset T,$$

respectively. Hence, examining the excess risks

$$\mathcal{R}_{L_T, P}(\hat{f}_D, \lambda) - \mathcal{R}_{L_T, P}^*(\hat{f}_D, \lambda) \leq \mathcal{R}_{L_{J_T}, P}(\hat{f}_D, \lambda) - \mathcal{R}_{L_{J_T}, P}^*(\hat{f}_D, \lambda)$$

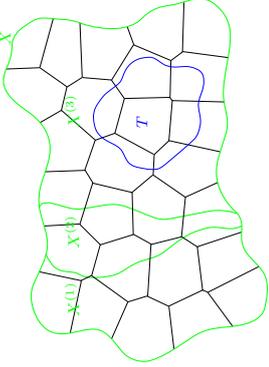


Figure 3: The input space X with the corresponding partition $(A_j)_{j=1, \dots, m}$ and the subset T , where the local learning rate should be examined.

leads to learning rates on A_T and implicitly on T . Recapitulatory, let us declare a set of notations that will be frequently used in the remainder of the paper.

(T) For $T \subset X$, we define an index set J_T by (7), loss functions $L_T, L_{J_T} : X \times Y \times \mathbb{R} \rightarrow [0, \infty)$ by (8) and (9), and the set $A_T := \bigcup_{j \in J_T} A_j$.

3. Building Weighted Global Kernels

In this section, we first focus on RKHSs and direct sums of RKHSs. Then, we show that a VP-SVM solution is also the solution of an usual SVM.

Let us begin with some basic notations. For $q \in [1, \infty]$ and a measure ν , we denote by $L_q(\nu)$ the Lebesgue spaces of order q w.r.t. ν and for the Lebesgue measure μ on $X \subset \mathbb{R}^d$ we write $L_q(X) := L_q(\mu)$. In addition, for a measurable space X , the set of all real-valued measurable functions on X is given by $\mathcal{L}_0(X) := \{f : X \rightarrow \mathbb{R} \mid f \text{ measurable}\}$. Moreover, for a measure ν on X and measurable $\tilde{X} \subset X$, we define the trace measure $\nu|_{\tilde{X}}$ of ν in \tilde{X} by $\nu|_{\tilde{X}}(A) = \nu(A \cap \tilde{X})$ for every $A \subset X$.

Our first goal is to show that $f_{D,\lambda}$ in (3) is actually an ordinary SVM solution. To this end, we consider an RKHS on some $A \subset X$ and extend it to an RKHS on X by the following lemma, where we omit the obvious proof.

Lemma 2 Let $A \subset X$ and H_A be an RKHS on A with corresponding kernel k_A . Denote by \tilde{f} the zero-extension of $f \in H_A$ to X defined by

$$\tilde{f}(x) := \begin{cases} f(x), & \text{for } x \in A, \\ 0, & \text{for } x \in X \setminus A. \end{cases}$$

Then, the space $\tilde{H}_A := \{\tilde{f} : f \in H_A\}$ equipped with the norm $\|\tilde{f}\|_{\tilde{H}_A} := \|f\|_{H_A}$ is an RKHS on X and its reproducing kernel is given by

$$\tilde{k}_A(x, x') := \begin{cases} k_A(x, x'), & \text{if } x, x' \in A, \\ 0, & \text{else.} \end{cases} \quad (10)$$

Based on this lemma, we are now able to construct an RKHS by a direct sum of RKHSs \hat{H}_A and \hat{H}_B with $A, B \subset X$ and $A \cap B = \emptyset$. Here, we skip the proof once more, since the assertion follows immediately using, for example, orthonormal bases of \hat{H}_A and \hat{H}_B .

Lemma 3 For $A, B \subset X$ such that $A \cap B = \emptyset$ and $A \cup B \subset X$, let H_A and H_B be RKHSs of the kernels k_A and k_B over A and B , respectively. Furthermore, let \hat{H}_A and \hat{H}_B be the RKHSs of all functions of H_A and H_B extended to X in the sense of Lemma 2 and let k_A and k_B given by (10) be the associated reproducing kernels. Then, $\hat{H}_A \cap \hat{H}_B = \{0\}$ and hence the direct sum

$$H := \hat{H}_A \oplus \hat{H}_B \quad (11)$$

exists. For $\lambda_A, \lambda_B > 0$ and $f \in H$, let $\hat{f}_A \in \hat{H}_A$ and $\hat{f}_B \in \hat{H}_B$ be the unique functions such that $f = \hat{f}_A + \hat{f}_B$. Then, we define the norm $\|\cdot\|_H$ by

$$\|f\|_H^2 := \lambda_A \|\hat{f}_A\|_{\hat{H}_A}^2 + \lambda_B \|\hat{f}_B\|_{\hat{H}_B}^2 \quad (12)$$

and H equipped with the norm $\|\cdot\|_H$ is again an RKHS for which

$$k(x, x') := \lambda_A^{-1} \hat{k}_A(x, x') + \lambda_B^{-1} \hat{k}_B(x, x'), \quad x, x' \in X,$$

is the reproducing kernel.

To relate Lemmas 2 and 3 with (3), we have to introduce some more notations. For pairwise disjoint sets $A_1, \dots, A_m \subset X$, let H_j be an RKHS on A_j for every $j \in \{1, \dots, m\}$. Then, based on RKHSs $\hat{H}_1, \dots, \hat{H}_m$ on X defined by Lemma 2, a joined RKHS can be designed analogously to Lemma 3. That is, for an arbitrary index set $J \subset \{1, \dots, m\}$ and a vector $\lambda = (\lambda_j)_{j \in J} \in (0, \infty)^{|J|}$, the direct sum

$$H_J := \bigoplus_{j \in J} \hat{H}_j = \left\{ f = \sum_{j \in J} f_j : f_j \in \hat{H}_j \text{ for all } j \in J \right\} \quad (13)$$

is again an RKHS equipped with the norm

$$\|f\|_{H_J}^2 = \sum_{j \in J} \lambda_j \|f_j\|_{\hat{H}_j}^2. \quad (14)$$

If $J = \{1, \dots, m\}$, we simply write $H := H_J$. Note that H contains inter alia $f_{D,\lambda}$ given by (3).

Let us briefly investigate the regularized empirical risk of $f_{D,\lambda} = \sum_{j=1}^m \mathbb{1}_{A_j} \hat{f}_{j,\lambda_j}$, where \hat{f}_{j,λ_j} , $j = 1, \dots, m$, are defined by (2). For an arbitrary $f \in H$, we have

$$\begin{aligned} \|f_{D,\lambda}\|_H^2 + \mathcal{R}_{L,D}(\hat{f}_{D,\lambda}) &= \sum_{j=1}^m \left(\lambda_j \|f_{D_j,\lambda_j}\|_{\hat{H}_j}^2 + \mathcal{R}_{L_j,D}(\hat{f}_{D_j,\lambda}) \right) \\ &\leq \sum_{j=1}^m \left(\lambda_j \|\mathbb{1}_{A_j} f\|_{\hat{H}_j}^2 + \mathcal{R}_{L_j,D}(f) \right) \end{aligned}$$

$$= \|f\|_H^2 + \mathcal{R}_{L,D}(f), \quad (15)$$

where we used $\mathcal{R}_{L,D}(f) = \sum_{j=1}^m \mathcal{R}_{L_j,D}(f)$, which immediately follows by Lemma 9 given in the appendix. That is, $f_{D,\lambda}$ is the decision function of an SVM using H and L as well as the regularization parameter $\lambda = 1$. In other words, the latter SVM equals the VP-SVM given by (3). This will be a key insight used in our analysis.

Subsequently, we only consider RKHSs of Gaussian RBF kernels. For this purpose, we summarize some assumptions for the Gaussian case of joined RKHSs in the following assumption set.

(G) For pairwise disjoint subsets A_1, \dots, A_m of X , let $H_j := H_{\gamma_j}(A_j)$, $j \in \{1, \dots, m\}$, be the RKHS of the Gaussian kernel k_{γ_j} with width $\gamma_j \in (0, r]^\dagger$ over A_j . Consequently, for $\lambda := (\lambda_1, \dots, \lambda_m) \in (0, \infty)^m$, we define the joined RKHS $H := \bigoplus_{j=1}^m H_{\gamma_j}(A_j)$ and equip it with the norm (14).

In the following we do not consider SVMs with a fixed kernel, thus, we use a more detailed notation than (2) and (3) specifying the kernel width γ_j of the RKHS $H_{\gamma_j}(A_j)$ at hand. Namely, for all $j \in \{1, \dots, m\}$ and $\gamma := (\gamma_1, \dots, \gamma_m)$, we write

$$f_{D_j, \lambda_j \gamma_j} = \arg \min_{f \in H_{\gamma_j}(A_j)} \|f\|_{H_{\gamma_j}(A_j)}^2 + \frac{1}{n} \sum_{i=1}^n L_{\gamma_j}(x_i, y_i, f(x_i)),$$

and

$$f_{D,\lambda,\gamma} := \sum_{j=1}^m f_{D_j, \lambda_j \gamma_j}$$

instead of f_{D_j, λ_j} and $f_{D,\lambda}$ in the remainder of this work.

4. Learning Rates for Least Squares VP-SVMs

In this section, the non-parametric least squares regression problem is considered using the least squares loss $L : Y \times \mathbb{R} \rightarrow [0, \infty)$ defined by $L(g, t) := (g - t)^2$. It is well known that, in this case, the Bayes decision function $f_{L,P}^* : \mathbb{R}^d \rightarrow \mathbb{R}$ is given by $f_{L,P}^*(x) = \mathbb{E}_P(Y|x)$ for P -almost all $x \in \mathbb{R}^d$. Moreover, this function is unique up to zero-sets. Besides, for the least squares loss the equality

$$\mathcal{R}_{L,P}(f) - \mathcal{R}_{L,P}^* = \|f - f_{L,P}^*\|_{L_2(P^X)}^2$$

can be shown by some simple, well-known transformations. In the first part of Subsection 4.1 we introduce some tools to describe smoothness properties of $f_{L,P}^*$, which are then used in the oracle inequalities and learning rates of the second part. In Subsection 4.2 we then investigate a simple parameter selection strategy for which we will show that it is adaptive.

4.1 Basic Oracle Inequalities for LS-VP-SVMs

To formulate oracle inequalities and derive rates for VP-SVMs using the least squares loss, the target function $f_{L,P}^*$ is assumed to satisfy certain smoothness conditions. To this end, we initially recall the modulus of smoothness, a device to measure the smoothness of functions, see e.g., DeVore and Lorentz, 1993, p. 44; DeVore and Popov, 1988, p. 398; as well as Berens and DeVore, 1978, p. 360. Denote by $\|\cdot\|_2$ the Euclidean norm and let $\Omega \subset \mathbb{R}^d$ be a subset with non-empty interior, ν be an arbitrary measure on Ω , $p \in [0, \infty]$, and $f : \Omega \rightarrow \mathbb{R}$ be contained in $L_p(\nu)$. Then, for $s \in \mathbb{N}$, the s -th modulus of smoothness of f is defined by

$$\omega_{s, L_p(\nu)}(f, t) = \sup_{\|h\|_2 \leq t} \|\Delta_h^s(f, \cdot)\|_{L_p(\nu)}, \quad t \geq 0,$$

where $\Delta_h^s(f, \cdot)$ denotes the s -th difference of f given by

$$\Delta_h^s(f, x) = \begin{cases} \sum_{j=0}^s \binom{s}{j} (-1)^{s-j} f(x + jh) & \text{if } x \in \Omega_{s,h} \\ 0 & \text{if } x \notin \Omega_{s,h} \end{cases}$$

for $h = (h_1, \dots, h_d) \in \mathbb{R}^d$ and $\Omega_{s,h} := \{x \in \Omega : x + th \in \Omega \text{ f.a. } t \in [0, s]\}$. Based on the modulus of smoothness, we introduce Besov-like spaces, i.e., function spaces that provide a finer scale of smoothness than the commonly used Sobolev spaces and that will thus be assumed to contain the target function later on. To this end, let $\alpha > 0$, $s := [\alpha] + 1$, and ν be an arbitrary measure. Then, the Besov-like space $B_{2,\infty}^\alpha(\nu)$ is defined by

$$B_{2,\infty}^\alpha(\nu) := \left\{ f \in L_2(\nu) : \|f\|_{B_{2,\infty}^\alpha(\nu)} < \infty \right\},$$

where the semi-norm $\|\cdot\|_{B_{2,\infty}^\alpha(\nu)}$ is given by

$$\|f\|_{B_{2,\infty}^\alpha(\nu)} := \sup_{t>0} (t^{-\alpha} \omega_{s, L_2(\nu)}(f, t))$$

and the norm by $\|f\|_{B_{2,\infty}^\alpha(\nu)} := \|f\|_{L_2(\nu)} + \|f\|_{B_{2,\infty}^\alpha(\nu)}$. Here, note that we defined Besov-like spaces for arbitrary measures ν on $\Omega \subset \mathbb{R}^d$ whereas in the literature Besov spaces are usually defined for the Lebesgue measure. Nevertheless, our definition of Besov-like spaces is well-defined. Moreover, for the proofs it is important to notice that, if $\Omega = \mathbb{R}^d$ and ν is a distribution on Ω with $\text{supp } \nu \subseteq \Omega$, then $\Omega_{s,h}$ still equals \mathbb{R}^d , i.e., $\Omega_{s,h} = \Omega$. Also note that for the Lebesgue measure on Ω , where $\Omega = \mathbb{R}^d$ or Ω is a bounded Lipschitz domain in \mathbb{R}^d , our definition of Besov-like spaces actually coincides, up to equivalent norms, to the definition of the classical Besov spaces in the literature, see e.g., (Adams and Fournier, 2003, Section 7), (Triebel, 2006, Section 1), (Triebel, 1992, Section 1), and (Triebel, 2010, Sections 2 and 3), where this classical type of Besov spaces is also defined for $1 \leq p, q \leq \infty$ and $\alpha > 0$. For more details on the equivalences of our definition of Besov-like spaces and the classical definitions, we refer to (Ehberts, 2015, Section 3.1). If ν is the Lebesgue measure on Ω , we write $B_{2,\infty}^\alpha(\Omega) := B_{2,\infty}^\alpha(\nu)$. Additionally, let us briefly consider a few embedding properties for Besov-like spaces $B_{2,\infty}^\alpha(\nu)$ where the corresponding proofs can be found in (Ehberts, 2015, Section 3.1). To this end, let ν be a finite measure on \mathbb{R}^d such that $\text{supp } \nu :=: \Omega \subset \mathbb{R}^d$ has non-empty interior and ν has a Lebesgue density g on Ω . If g is bounded away from 0

on Ω , then $B_{2,\infty}^\alpha(\nu) \subset B_{2,\infty}^\alpha(\Omega)$ for $\alpha > 0$. Alternatively, for $g \in L_\infty(\Omega)$ and $\alpha > 0$, we have $B_{2,\infty}^\alpha(\mathbb{R}^d) \subset B_{2,\infty}^\alpha(\nu)$ and $(B_{2,\infty}^\alpha(\Omega^{+\delta}) \cap L_\infty(\mathbb{R}^d)) \subset B_{2,\infty}^\alpha(\nu)$, where $\delta > 0$ and $\Omega^{+\delta} := \{x \in \mathbb{R}^d : \exists x' \in \Omega \text{ such that } \|x - x'\|_2 \leq \delta\}$. For the sake of completeness, recall from, e.g., (Adams and Fournier, 2003, Section 3) and (Triebel, 2010, Sections 2 and 3) the scale of Sobolev spaces $W_2^\alpha(\nu)$ defined by

$$W_2^\alpha(\nu) := \left\{ f \in L_p(\nu) : \partial^{(\beta)} f \in L_2(\nu) \text{ exists for all } \beta \in \mathbb{N}_0^d \text{ with } |\beta| \leq \alpha \right\},$$

where $\alpha \in \mathbb{N}_0$, ν is an arbitrary measure, and $\partial^{(\beta)}$ is the β -th weak derivative for a multi-index $\beta = (\beta_1, \dots, \beta_d) \in \mathbb{N}_0^d$ with $|\beta| = \sum_{i=1}^d \beta_i$. That is, $W_2^\alpha(\nu)$ is the space of all functions in $L_2(\nu)$ whose weak derivatives up to order α exist and are contained in $L_2(\nu)$. Moreover, the Sobolev space is equipped with the Sobolev norm

$$\|f\|_{W_2^\alpha(\nu)}^p := \sum_{|\beta| \leq \alpha} \|\partial^{(\beta)} f\|_{L_2(\nu)}^2,$$

(cf. Adams and Fournier, 2003, p. 60). We write $W_2^0(\nu) = L_2(\nu)$ and, for the Lebesgue measure μ on $\Omega \subset \mathbb{R}^d$, we define $W_2^\alpha(\Omega) := W_2^\alpha(\mu)$. It is well-known, see e.g., (Edmunds and Triebel, 1996, p. 25 and p. 44), that the Sobolev spaces $W_2^\alpha(\mathbb{R}^d)$ fall into the scale of Besov spaces, e.g., $W_2^\alpha(\mathbb{R}^d) \subset B_{2,\infty}^\alpha(\mathbb{R}^d)$ for $\alpha \in \mathbb{N}$. Furthermore, note that functions $f : \Omega \rightarrow \mathbb{R}^d$ can be extended to functions $\tilde{f} : \mathbb{R}^d \rightarrow \mathbb{R}$ such that \tilde{f} inherits the smoothness properties of f , whenever $\Omega \subset \mathbb{R}^d$ is a bounded Lipschitz domain. More precisely, in this case Stein's Extension Theorem (cf. Stein, 1970, p. 181) guarantees the existence of a linear extension operator \mathfrak{E} mapping functions $f : \Omega \rightarrow \mathbb{R}$ to functions $\mathfrak{E}f : \mathbb{R}^d \rightarrow \mathbb{R}$ such that $\mathfrak{E}f|_\Omega = f$ and such that \mathfrak{E} continuously maps $W_2^m(\Omega)$ into $W_2^m(\mathbb{R}^d)$ for all integers $m \geq 0$ and $B_{2,\infty}^\alpha(\Omega)$ into $B_{2,\infty}^\alpha(\mathbb{R}^d)$ for all $\alpha \geq 0$ simultaneously. For more details, we refer to Stein (1970, p. 181), Triebel (2006, Section 1.11.5), and Adams and Fournier (2003, Chapter 5). In this case, Eberts (2015, Corollary 3.4) shows, for a finite measure ν on \mathbb{R}^d such that $\text{supp } \nu =: \Omega \supset \Omega'$ and such that ν has a Lebesgue density g on Ω with $g \in L_\infty(\Omega)$, that $f \in B_{2,\infty}^\alpha(\Omega)$ implies $\mathfrak{E}f \in B_{2,\infty}^\alpha(\nu)$.

Based on the least squares loss and RKHSs using Gaussian kernels over the partition sets A_j , the subsequent theorem presents an oracle inequality for VP-SVMs.

Theorem 4 Let $Y := [-M, M]$ for $M > 0$, $L : Y \times \mathbb{R} \rightarrow [0, \infty)$ be the least squares loss, and P be a distribution on $\mathbb{R}^d \times Y$. We denote the marginal distribution of P onto \mathbb{R}^d by P_X , write $X := \text{supp } P_X$, and assume $P_X(\partial X) = 0$. Furthermore, let (\mathbf{A}) and (\mathbf{G}) be satisfied. In addition, for an arbitrary subset $T \subset X$, we assume (\mathbf{T}) . Moreover, let $f_{L,P}^* : \mathbb{R}^d \rightarrow \mathbb{R}$ be a Bayes decision function such that $f_{L,P}^* \in L_2(\mathbb{R}^d) \cap L_\infty(\mathbb{R}^d)$ as well as $f_{L,P}^* \in B_{2,\infty}^\alpha(P_X|_{A_T})$ for some $\alpha \geq 1$. Then, for all $p \in (0, 1)$, $n \geq 1$, $\tau \geq 1$, $\gamma = (\gamma_1, \dots, \gamma_m) \in (0, r]^m$, and $\lambda = (\lambda_1, \dots, \lambda_m) > 0$, the VP-SVM given by (3) using $\hat{H}_{\gamma_1}(A_1), \dots, \hat{H}_{\gamma_m}(A_m)$, and the loss L_{J_T} satisfies

$$\sum_{j=1}^m \lambda_j \|f_{D_{j,\lambda_j,\gamma_j}}\|_{\hat{H}_{\gamma_j}(A_j)}^2 + \mathcal{R}_{L_{J_T},P}(\hat{f}_{D,\lambda,\gamma}) - \mathcal{R}_{L_{J_T},P}^*$$

$$\leq C_{M,\alpha,p} \left(\sum_{j \in J_T} \lambda_j \gamma_j^{-d} + \left(\frac{\max_{j \in J_T} \gamma_j}{\min_{j \in J_T} \gamma_j} \right)^d \max_{j \in J_T} \gamma_j^{2\alpha} + r^{2p} \left(\sum_{j=1}^m \lambda_j^{-1} \gamma_j^{-\frac{d+2p}{p}} P_X(A_j) \right)^p n^{-1} + \tau n^{-1} \right)$$

with probability P^n not less than $1 - e^{-\tau}$, where $C_{M,\alpha,p} > 0$ is a constant only depending on $M, \alpha, p, d, \|f_{L,P}^*\|_{L_2(\mathbb{R}^d)}, \|f_{L,P}^*\|_{L_\infty(\mathbb{R}^d)}$, and $\|f_{L,P}^*\|_{B_{2,\infty}^\alpha(P_X|_{A_T})}$.

We like to emphasize that in the theorem above $X := \text{supp } P_X$ only serves as a notation. Indeed, the partition $(A_j)_{j=1,\dots,m}$ of (\mathbf{A}) can be found without knowing $\text{supp } P_X$, and whether we actually remove the cells that do not intersect the interior of $\text{supp } P_X$ is irrelevant since these cells will neither contain samples nor will they contribute to the overall risk of our decision function $\hat{f}_{D,\lambda,\gamma}$ as we assumed $P_X(\partial X) = 0$. Despite from this, the proofs anyway do not require that X exactly corresponds to the support of the distribution P_X . Instead we can as well assume $\text{supp } P_X \subset X \subset B_{\text{eff}}^d$. Moreover, note for the proofs that the considered Besov-like space $B_{2,\infty}^\alpha(P_X|_{A_T})$ is defined w.r.t. $\Omega = \mathbb{R}^d$.

Theorem 4 only focuses on the least squares loss, however, a similar version can be shown under more general assumptions for generic losses and RKHSs, where we refer the interested reader to (Eberts, 2015, Theorem 4.4). Moreover, considering a trivial partition consisting of only one set A_1 the oracle inequalities for VP-SVMs are comparable to the already known ones, see (Eberts, 2015, p. 81) for more details.

Using the oracle inequality of Theorem 4, we derive learning rates w.r.t. the loss L_{J_T} for the learning method described by (2) and (3) in the following theorem.

Theorem 5 Let $\tau \geq 1$ be fixed and $\beta \geq \frac{2\alpha}{d} + 1$. Under the assumptions of Theorem 4 and with

$$r_n = c_1 n^{-\frac{1}{\beta\alpha}}, \quad (16)$$

$$\lambda_{n,j} = c_2 r_n^d n^{-1}, \quad (17)$$

$$\gamma_{n,j} = c_3 n^{-\frac{1}{2\alpha+d}}, \quad (18)$$

for every $j \in \{1, \dots, m_n\}$, we have, for all $n \geq 1$ and $\xi > 0$,

$$\mathcal{R}_{L_{J_T},P}(\hat{f}_{D,\lambda_n,\gamma_n}) - \mathcal{R}_{L_{J_T},P}^* \leq C \left(n^{-\frac{2\alpha}{2\alpha+d} + \xi} + \tau n^{-1} \right)$$

with probability P^n not less than $1 - e^{-\tau}$, where $\lambda_n := (\lambda_{n,1}, \dots, \lambda_{n,m_n})$ as well as $\gamma_n := (\gamma_{n,1}, \dots, \gamma_{n,m_n})$ and C, c_1, c_2, c_3 are positive constants with $c_3 \leq c_1$.

In the latter theorem the condition $\beta \geq \frac{2\alpha}{d} + 1$ is required to ensure $\gamma_{n,j} \leq r_n$, $j = 1, \dots, m_n$, which in turn is a prerequisite arising from Theorem 12 and the used entropy estimate. Let us briefly examine the extreme case $\beta = \frac{2\alpha}{d} + 1$. Using $r_n \approx n^{-\frac{1}{\beta\alpha}}$ and (5) leads to covering numbers of the form $m_n \approx n^{\frac{d}{2\alpha+d}}$ and computational costs of $\mathcal{O}(m_n (\frac{2\alpha}{m_n})^q) = \mathcal{O}(n^{\frac{2\alpha q}{2\alpha+d}})$ which is actually less than the computational cost of order n^q , $q \in [2, 3]$, of an usual SVM. Note that for increasing β the computational costs of an VP-SVM are increasing as well. However, for $\beta > \frac{2\alpha}{d} + 1$, $r_n \approx n^{-\frac{1}{\beta\alpha}}$, and $m_n \approx n^{\frac{d}{\beta}}$, a VP-SVM has costs of $\mathcal{O}(n^{\frac{1+\beta-1/q}{\beta}})$ which still is less than $\mathcal{O}(n^q)$.

Let us finally take a closer look at the VP-SVM given by (3) and the considerations related to (15), where $f_{b,\lambda} \in H = \bigoplus_{j=1}^m H_j$ solves the minimization problem

$$f_{b,\lambda} = \arg \min_{f \in \tilde{H}_1, \dots, f_m \in \tilde{H}_m} \sum_{j=1}^m \lambda_j \|f_j\|_{\tilde{H}_j}^2 + \mathcal{R}_{L,D} \left(\sum_{j=1}^m f_j \right).$$

Choosing $\lambda_1 = \dots = \lambda_m$, the VP-SVM problem can be understood as particular ℓ_2 -multiple kernel learning (MKL) problem using the RKHSs $\tilde{H}_1, \dots, \tilde{H}_m$. Learning rates for MKL have been treated, for example, in (Suzuki, 2011) and (Kloft and Blanchard, 2012). Assuming $f_{L,P}^* \in H$, the learning rate achieved in (Suzuki, 2011) is $mm^{-\frac{1}{1+\alpha}}$ for dense settings, where s is the so-called spectral decay coefficient. In addition, Kloft and Blanchard (2012) obtain essentially the same rates under these assumptions. Let us therefore briefly investigate the above rate of (Suzuki, 2011). For RKHSs that are continuously embedded in a Sobolev space $W_2^s(X)$, we have $s = \frac{d}{2\alpha}$ such that the learning rate reduces to $mn^{-\frac{2\alpha}{2\alpha+d}}$. Note that this learning rate is m times the optimal learning rate $n^{-\frac{2\alpha}{2\alpha+d}}$, where the number $m = mn_n$ of kernels may increase with the sample size n . In particular, if $m_n \rightarrow \infty$ polynomially, then the rates obtained in (Suzuki, 2011) become substantially worse than the optimal rate. In contrast, due to the special choice of the RKHSs, this is not the case for our VP-SVM problem, provided that m_n does not grow faster than $n^{1/\beta}$.

Note that the oracle inequalities and learning rates achieved in Theorems 4 and 5 require $f_{L,P}^* \in B_{2,\infty}^\alpha(\mathbb{P}_X \cup_{j \in J_T} A_j)$. However, for an increasing sample size n , the sets A_j shrink and the index set J_T , indicating every set A_j such that $A_j \cap T \neq \emptyset$ and $T \subset \bigcup_{j \in J_T} A_j$, increases. In particular, this also involves that the set $\bigcup_{j \in J_T} A_j$ covering T changes in tandem with n . Since this is very inconvenient and since it would be desirable to assume a certain level of smoothness of the target function on a fixed region for all $n \in \mathbb{N}$, we consider the set T enlarged by an δ -tube. To this end, for $\delta > 0$, we define $T^{+\delta}$ by

$$T^{+\delta} := \{x \in X \mid \exists k \in T : \|x - k\|_2 \leq \delta\},$$

which implies $T \subset T^{+\delta} \subset X$, cf. Figure 4. Note that, for every $\delta > 0$, there exists an $n_\delta \in \mathbb{N}$ such that, for every $n \geq n_\delta$, the union of all partition sets A_j , having at least one common point with T , is contained in $T^{+\delta}$, i.e.

$$\forall \delta > 0 \quad \exists n_\delta \in \mathbb{N} \quad \forall n \geq n_\delta \quad : \quad \bigcup_{j \in J_T} A_j \subset T^{+\delta}, \quad (19)$$

where $J_T := \{j \in \{1, \dots, m_n\} : A_j \cap T \neq \emptyset\}$. Collectively, this implies $T \subset \bigcup_{j \in J_T} A_j \subset T^{+\delta}$ for all $n \geq n_\delta$. Furthermore, since every set A_j is contained in a ball with radius $r_n = cn^{-\frac{1}{d}}$ satisfying (5), the lowest sample size n_δ in (19) can be determined by choosing the smallest $n_\delta \in \mathbb{N}$ such that $\delta \geq 2r_{n_\delta}$, that is

$$n_\delta = \left\lceil \left(\frac{2c}{\delta} \right)^{\beta d} \right\rceil.$$

This leads to the following corollary, which presents an oracle inequality and learning rates assuming the smoothness level α of the target function on a fixed region.

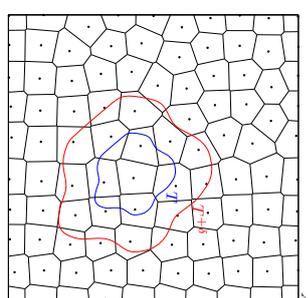


Figure 4: An input space X with a Voronoi partition as well as a subset $T \subset X$ enlarged by an δ -tube to $T^{+\delta}$.

Corollary 6 Let $Y := [-M, M]$ for $M > 0$, $L : Y \times \mathbb{R} \rightarrow [0, \infty)$ be the least squares loss, and \mathbb{P} be a distribution on $\mathbb{R}^d \times Y$. We denote the marginal distribution of \mathbb{P} onto \mathbb{R}^d by \mathbb{P}_X , write $X := \text{supp } \mathbb{P}_X$, and assume $\mathbb{P}_X(\partial X) = 0$. Furthermore, let **(A)** and **(G)** be satisfied. In addition, for an arbitrary subset $T \subset X$, we assume **(T)**. Moreover, let $f_{L,P}^* : \mathbb{R}^d \rightarrow \mathbb{R}$ be a Bayes decision function with $f_{L,P}^* \in L_2(\mathbb{R}^d) \cap L_\infty(\mathbb{R}^d)$ as well as

$$f_{L,P}^* \in B_{2,\infty}^\alpha(\mathbb{P}_X|_{T^{+\delta}})$$

for $\alpha \geq 1$ and some $\delta > 0$. Then, for all $p \in (0, 1)$, $n \geq n_\delta$, $\tau \geq 1$, $\gamma = (\gamma_1, \dots, \gamma_m) \in (0, r]^{m_n}$, and $\lambda = (\lambda_1, \dots, \lambda_m) > 0$, the VP-SVM given by (3) using $\tilde{H}_n(A_1), \dots, \tilde{H}_n(A_{m_n})$, and the loss L_T satisfies

$$\begin{aligned} & \sum_{j=1}^{m_n} \lambda_j \|f_{D_j, \lambda_j \gamma_j}\|_{\tilde{H}_{n_j}(A_j)}^2 + \mathcal{R}_{L_T, \mathbb{P}}(\tilde{f}_{D, \lambda, \gamma}) - \mathcal{R}_{L_T, \mathbb{P}}^* \\ & \leq C_{M, \alpha, p} \left(\sum_{j \in J_T} \lambda_j \gamma_j^{-d} + \left(\frac{\max_{j \in J_T} \gamma_j}{\min_{j \in J_T} \gamma_j} \right)^d \max_{j \in J_T} \gamma_j^{2\alpha} + r^{2p} \left(\sum_{j=1}^{m_n} \lambda_j^{-1} \gamma_j^{-\frac{d+2\beta}{r}} \mathbb{P}_X(A_j) \right)^p n^{-1+\tau n^{-1}} \right) \end{aligned}$$

with probability \mathbb{P}^n not less than $1 - e^{-\tau}$, where $C_{M, \alpha, p} > 0$ is the same constant as in Theorem 4.

Additionally, let $\beta \geq \frac{2\alpha}{d} + 1$ as well as, for every $j \in \{1, \dots, m_n\}$, r_n , $\lambda_{n,j}$, and $\gamma_{n,j}$ be as in (16), (17), and (18), respectively, where c_1, c_2, c_3 are user-specified positive constants with $c_3 \leq c_1$. Then, for all $n \geq n_\delta = \left\lceil \left(\frac{2c_3}{\delta} \right)^{\beta d} \right\rceil$ and $\xi > 0$, we have

$$\mathcal{R}_{L_T, \mathbb{P}}(\tilde{f}_{D, \lambda_n, \gamma_n}) - \mathcal{R}_{L_T, \mathbb{P}}^* \leq C \left(n^{-\frac{2\alpha}{2\alpha+d} + \xi} + \tau n^{-1} \right)$$

with probability \mathbb{P}^n not less than $1 - e^{-\tau}$, where $\lambda_n := (\lambda_{n,1}, \dots, \lambda_{n,m_n})$, $\gamma_n := (\gamma_{n,1}, \dots, \gamma_{n,m_n})$, and C is a positive constant.

Note that the assumption $f_{L,p}^* \in B_{2,\infty}^\alpha(\mathbb{P}_X|_{T^{+\varepsilon}})$ made in Corollary 6 is satisfied if, for example, \mathbb{P}_X has a bounded Lebesgue density on $T^{+\delta}$, $f_{L,p}^* \in L_\infty(T^{+\delta})$, and either $f_{L,p}^* \in B_{2,\infty}^\alpha(T^{+2\delta})$ for $\alpha \geq 1$ or $f_{L,p}^* \in W_2^\alpha(T^{+2\delta}) \subset B_{2,\infty}^\alpha(T^{+2\delta})$ for $\alpha \in \mathbb{N}$ and a bounded Lipschitz domain $\bar{T} \subset \mathbb{R}^d$ such that $T^{+2\delta} \subset \bar{T}$. Moreover, if this density of \mathbb{P}_X is even bounded away from 0, it is well-known that the minimax rate is $n^{-\frac{2\alpha}{2\alpha+1}}$ for $\alpha > d/2$ and target functions $f_{L,p}^* \in W_2^\alpha(T)$ as well as for $\alpha > d$ and $f_{L,p}^* \in B_{2,\infty}^\alpha(T)$. Modulo ξ , our rate is therefore asymptotically optimal in a minimax sense on T .

Although the obtained learning rates are arbitrary close to the optimal rates, it is needless to say that the results are not fully satisfying. Indeed, an ideal result would not contain a gap of the form n^ξ , and a close to ideal result would at least replace the gap n^ξ by a logarithmic factor. Unfortunately, even for global SVMs using Gaussian kernels, such results seem to be currently out of reach, see (Eberts and Steinwart, 2013) for the latter case. Let us briefly describe the technical obstacles. One key ingredient for both the local and the global approach are estimates on the entropy numbers e_i of the embeddings $\text{id} : H_\gamma \rightarrow L_2(P_X)$ or $\text{id} : H_\gamma \rightarrow \ell_\infty(X)$, see Section 6 for a definition. Several such estimates do exist. For example, Zhou (2002) and Kühn (2011) proved (optimal) super-polynomial estimates but unfortunately their bounds have a unfavorable dependence on γ , which makes it impossible to get arbitrarily close to the optimal rates, see e.g., (Xiang and Zhou, 2009) for a similar situation in which this problem occurs. For this reason we followed the path of (Eberts and Steinwart, 2013), in which we employ an entropy estimate of the form

$$e_i(\text{id} : H_\gamma \rightarrow L_2(\mathbb{P}_X)) \leq c_{p,d} \gamma^{-p} i^{-\frac{d}{2}}, \quad i \geq 1, \gamma \in (0, 1],$$

where $c_{p,d} \geq 1$ is a constant only depending on $p \in \mathbb{N}$ and d . Note that this estimate is clearly sub-optimal in i , but it has a significantly better behavior in γ compared to the above mentioned results. Now, using this entropy estimate, Eberts and Steinwart (2013) obtain an oracle inequality of the form

$$\mathcal{R}_{L,p}(f_{D,\lambda,\gamma}) - \mathcal{R}_{L,p}^* \leq K_p \left(\lambda \gamma^{-d} + \gamma^{2\alpha} + \frac{c_{p,d}^{d/p} \gamma^{-d}}{\lambda^{\frac{d}{2p}} n} + \frac{\tau}{n} \right),$$

where the constant K_p is independent of γ , λ , τ , and n , and its dependence on p can be tracked, cf. (Steinwart and Christmann, 2008, p. 267). Note that for the local approach a structurally identical formula is derived implicitly in the proof of Theorem 4. Now, the rates in this paper as well as in (Eberts and Steinwart, 2013) are obtained by optimizing the right hand side with respect to both λ and γ for an arbitrarily large but fixed p . Since the resulting rates become better the larger we pick p it is tempting to consider $p = p_n \rightarrow \infty$. Unfortunately, however, this only becomes feasible if we have an explicit expression describing how $c_{p,d}$ may depend on p . For example, some preliminary considerations suggest that we could already replace the gap n^ξ by a logarithmic factor if we had a rough bound of the form $c_{p,d} \leq c_0 p^{\theta p}$. Unfortunately, we neither could derive such a bound for $c_{p,d}$ nor could we find it in the literature. Even worse, we also asked several experts for bounding entropy numbers of function space embeddings without any success. In addition, we are unaware of any other technique that has the potential to fill the gap in either the global or the local case, and therefore we leave this problem as an open question for future research.

4.2 Data-Dependent Parameter Selection for VP-SVMs

In the previous theorems the choice of the regularization parameters $\lambda_{n,1}, \dots, \lambda_{n,m_n}$ and the kernel widths $\gamma_{n,1}, \dots, \gamma_{n,m_n}$ requires us to know the smoothness parameter α . Unfortunately, in practice, we usually do not know either this value nor its existence. In this subsection, we thus show that a training/validation approach similar to the one examined in (Steinwart and Christmann, 2008, Chapters 6.5, 7.4, 8.2) and (Eberts and Steinwart, 2013) achieves the same rates adaptively, i.e., without knowing α . For this purpose, let $\Lambda := (\Lambda_n)$ and $\Gamma := (\Gamma_n)$ be sequences of finite subsets $\Lambda_n \subset (0, r_n^d]$ and $\Gamma_n \subset (0, r_n]$. For a data set $D := ((x_1, y_1), \dots, (x_n, y_n))$, we define

$$\begin{aligned} D_1 &:= ((x_1, y_1), \dots, (x_l, y_l)), \\ D_2 &:= ((x_{l+1}, y_{l+1}), \dots, (x_n, y_n)), \end{aligned}$$

where $l := \lfloor \frac{n}{2} \rfloor + 1$ and $n \geq 4$. We further split these sets in data sets

$$\begin{aligned} D_j^{(1)} &:= \{(x_i, y_i) \in D_1 : x_i \in A_j\}, & j \in \{1, \dots, m_n\}, \\ D_j^{(2)} &:= \{(x_i, y_i) \in D_2 : x_i \in A_j\}, & j \in \{1, \dots, m_n\}, \end{aligned}$$

and define $l_j := |D_j^{(1)}|$ for all $j \in \{1, \dots, m_n\}$ such that $\sum_{j=1}^{m_n} l_j = l$. For every $j \in \{1, \dots, m_n\}$, we basically use $D_j^{(1)}$ as a training set, i.e., based on D_1 in combination with the loss function $L_j := \mathbb{1}_{A_j} L$ we compute SVM decision functions

$$f_{D_j^{(1)}, \lambda_j, \gamma_j} := \arg \min_{f \in H_{\gamma_j}(A_j)} \lambda_j \|f\|_{H_{\gamma_j}(A_j)}^2 + \mathcal{R}_{L_j, D_1}(f), \quad (\lambda_j, \gamma_j) \in \Lambda_n \times \Gamma_n.$$

Note that $f_{D_j^{(1)}, \lambda_j, \gamma_j} = 0$ if $D_j^{(1)} = \emptyset$. Next, for each j , we use D_2 in tandem with L_j (or essentially $D_j^{(2)}$) to determine a pair $(\lambda_{D_{2,j}}, \gamma_{D_{2,j}}) \in \Lambda_n \times \Gamma_n$ such that

$$\mathcal{R}_{L_j, D_2} \left(\hat{f}_{D_j^{(1)}, \lambda_{D_{2,j}}, \gamma_{D_{2,j}}} \right) = \min_{(\lambda_j, \gamma_j) \in \Lambda_n \times \Gamma_n} \mathcal{R}_{L_j, D_2} \left(\hat{f}_{D_j^{(1)}, \lambda_j, \gamma_j} \right).$$

Finally, combining the decision functions $f_{D_j^{(1)}, \lambda_{D_{2,j}}, \gamma_{D_{2,j}}}$ for all $j \in \{1, \dots, m_n\}$, and defining $\lambda_{D_2} := (\lambda_{D_{2,1}}, \dots, \lambda_{D_{2,m_n}})$ and $\gamma_{D_2} := (\gamma_{D_{2,1}}, \dots, \gamma_{D_{2,m_n}})$, we obtain a function

$$f_{D_1, \lambda_{D_2}, \gamma_{D_2}} := \sum_{j=1}^{m_n} f_{D_j^{(1)}, \lambda_{D_{2,j}}, \gamma_{D_{2,j}}} = \sum_{j=1}^{m_n} \mathbb{1}_{A_j} f_{D_j^{(1)}, \lambda_{D_{2,j}}, \gamma_{D_{2,j}}},$$

and we call every learning method that produces these resulting decision functions $f_{D_1, \lambda_{D_2}, \gamma_{D_2}}$ a *training validation Voronoi partition support vector machine* (TV-VP-SVM) w.r.t. $\Lambda \times \Gamma$. Moreover, we have, for $\lambda := (\lambda_1, \dots, \lambda_{m_n})$ and $\gamma := (\gamma_1, \dots, \gamma_{m_n})$,

$$\mathcal{R}_{L, D_2} \left(\hat{f}_{D_1, \lambda_{D_2}, \gamma_{D_2}} \right) = \sum_{j=1}^{m_n} \mathcal{R}_{L_j, D_2} \left(\hat{f}_{D_j^{(1)}, \lambda_{D_{2,j}}, \gamma_{D_{2,j}}} \right)$$

$$\begin{aligned}
&= \sum_{j=1}^{m_n} \min_{(\lambda_j, \gamma_j) \in \Lambda_n \times \Gamma_n} \mathcal{R}_{L_j, D_2} \left(\widehat{f}_{D_j^{(1)}, \lambda_j, \gamma_j} \right) \\
&= \min_{(\lambda, \gamma) \in (\Lambda_n \times \Gamma_n)^{m_n}} \sum_{j=1}^{m_n} \mathcal{R}_{L_j, D_2} \left(\widehat{f}_{D_j^{(1)}, \lambda_j, \gamma_j} \right) \\
&= \min_{(\lambda, \gamma) \in (\Lambda_n \times \Gamma_n)^{m_n}} \mathcal{R}_{L, D_2} \left(\widehat{f}_{D_1, \lambda, \gamma} \right),
\end{aligned}$$

where $f_{D_1, \lambda, \gamma} := \sum_{j=1}^{m_n} f_{D_j^{(1)}, \lambda_j, \gamma_j}$ with $(\lambda_j, \gamma_j) \in \Lambda_n \times \Gamma_n$ for all $j \in \{1, \dots, m_n\}$. In other words, the function $f_{D_1, \lambda, \gamma, \tau, D_2}$ really minimizes the empirical risk \mathcal{R}_{L, D_2} w.r.t. the validation data set D_2 and the loss L_1 , where the minimum is taken over all functions $\widehat{f}_{D_1, \lambda, \gamma}$ with $(\lambda, \gamma) \in (\Lambda_n \times \Gamma_n)^{m_n}$.

Before we analyze the TV-VP-SVM algorithm, let us briefly discuss the computational complexity of the hyper-parameter selection step. To this end, we first note that the parameter selection on, e.g., the j -th cell is *completely independent* of the parameter selection on all other cells. Maybe the easiest way to visualize this is by thinking of having two cells and candidates $\Lambda = (\lambda_1, \dots, \lambda_k)$, only. Naively, this would give the candidate set $\Lambda \times \Lambda$ for the overall hyper-parameter selection procedure. However, inspecting the candidates on the first cell, we see the same results for the candidates in $\Lambda \times \{\lambda_1\}$ and in $\Lambda \times \{\lambda_2\}$ since any decision we make on the second cell does not influence our situation on the first cell. Consequently, we only need to consider the candidates $\Lambda \times \{\lambda_1\}$, that is the candidates in Λ , when performing parameter selection on the first cell, and analogously we only need to consider the candidates $\{\lambda_1\} \times \Lambda$ for the parameter selection on the second cell. Together this gives $2|\Lambda|$ many candidates, instead of $|\Lambda|^2$ many candidates of the naive approach.

Generalizing the reasoning above to m cells and $\Lambda \times \Gamma$, we easily see that our parameter selection strategy leads to the inspection of $m \times |\Lambda| \times |\Gamma|$ many candidates. Moreover, because of the independence of all cells, we could actually perform parameter selection on the cells in parallel. Clearly such a parallel approach would be easy to implement and would have minimal synchronization and communication overhead.

The following theorem presents learning rates for the above described TV-VP-SVM.

Theorem 7 *Let $r_n := cn^{-\frac{1}{2d}}$ with constants $c > 0$ and $\beta > 1$. Under the assumptions of Theorem 4 we fix sequences $\Lambda := (\Lambda_n)$ and $\Gamma := (\Gamma_n)$ of finite subsets $\Lambda_n \subset (0, r_n^d]$ and $\Gamma_n \subset (0, r_n]$ such that Λ_n is an $(r_n^d \varepsilon_n)$ -net of $(0, r_n^d]$ and Γ_n is a δ_n -net of $(0, r_n]$ with $\varepsilon_n \leq n^{-1}$ and $\delta_n \leq n^{-\frac{1}{2+2d}}$. Furthermore, assume that the cardinalities $|\Lambda_n|$ and $|\Gamma_n|$ grow polynomially in n . Then, for all $\xi > 0$, $\tau \geq 1$, and $\alpha < \frac{\beta-1}{2}d$, the TV-VP-SVM producing the decision functions $f_{D_1, \lambda, \gamma, \tau, D_2}$ satisfies*

$$\mathbb{P}^n \left(\mathcal{R}_{L_{j^*}, P}(\widehat{f}_{D_1, \lambda, \gamma, \tau, D_2}) - \mathcal{R}_{L_{j^*}, P}^* \leq c \left(n^{-\frac{2\alpha}{2d+\tau} + \xi} + \tau n^{-1} \right) \geq 1 - e^{-\tau}, \right.$$

where $c > 0$ is a constant independent of n and τ .

Once more, we can replace the assumption $f_{L^*}^* \in B_{2, \infty}^\alpha(\mathbb{P}_{X|Y})$ by $f_{L, P}^* \in B_{2, \infty}^\alpha(\mathbb{P}_{X|Y+\delta})$ for some $\delta > 0$ and obtain the same learning rate as in Theorem 7 for all $n \geq n_\delta$ although

$T^{+\delta}$ is fixed for all $n \in \mathbb{N}$. Here, recall that $f_{L, P}^* \in B_{2, \infty}^\alpha(\mathbb{P}_{X|Y+\delta})$ whenever \mathbb{P}_X has a bounded Lebesgue density on $T^{+\delta}$, $f_{L, P}^* \in L_\infty(T^{+\delta})$, and either $f_{L, P}^* \in B_{2, \infty}^\alpha(T^{+2\delta})$ for $\alpha \geq 1$ or $f_{L, P}^* \in W_2^\alpha(\widetilde{T}) \subset B_{2, \infty}^\alpha(T^{+2\delta})$ for $\alpha \in \mathbb{N}$ and a bounded Lipschitz domain $\widetilde{T} \subset \mathbb{R}^d$ such that $T^{+2\delta} \subset \widetilde{T}$. Moreover, let us assume that $\widetilde{T} \supseteq T^{+\delta}$ is a bounded Lipschitz domain in \mathbb{R}^d such that Stein's extension operator \mathfrak{E} exists and that P is a distribution on $\mathbb{R}^d \times Y$ such that \mathbb{P}_X has a Lebesgue density g on $T^{+\delta}$ with $g \in L_\infty(T^{+\delta})$. Then, the assumptions $f_{L, P}^* \in B_{2, \infty}^\alpha(T)$ and $f_{L, P}^* \in L_\infty(T)$ yield $\mathfrak{E}f_{L, P}^* \in B_{2, \infty}^\alpha(\mathbb{P}_{X|Y+\delta})$ and $\mathfrak{E}f_{L, P}^* \in L_2(\mathbb{R}^d) \cap L_\infty(\mathbb{R}^d)$, see (Eberts, 2015, Corollary 3.4 and Theorem 3.2) for more details. Thus, applying $\mathcal{R}_{L_{j^*}, P}^* = \mathcal{R}_{L_{j^*}, P}(\mathfrak{E}f_{L, P}^*)$ and choosing $f_0 := \sum_{j \in J_n} \mathbb{1}_{A_j} \cdot (K_j * \mathfrak{E}f_{L, P}^*)$, we obtain the same results as in Corollary 6 and Theorem 7 for $n \geq n_\delta$. Obviously, the same is true, if we assume $f_{L, P}^* \in W_2^\alpha(\widetilde{T})$ instead of $f_{L, P}^* \in B_{2, \infty}^\alpha(T)$. For all these cases, note that, if \mathbb{P}_X has a Lebesgue density that is bounded away from 0 and ∞ and either $f_{L, P}^* \in W_2^\alpha(T)$ for $\alpha > d/2$ or $f_{L, P}^* \in B_{2, \infty}^\alpha(T)$ for $\alpha > d$, the achieved learning rate $n^{-\frac{2\alpha}{2d+\tau}}$ is again asymptotically optimal modulo ξ on T in a minmax sense. Here, we only derived learning rates when using the least squares loss. However, similar rates are shown by Eberts (2015, Section 9) for quantile regression using the pinball loss.

To derive the above learning rates, we need the condition $\alpha < \frac{\beta-1}{2}d$. However, this condition restricts the set of α -values where we obtain learning rates adaptively. To be more precise, there is a trade-off between α and β . On the one hand, for small values of β only a small number of possible values for α is covered. On the other hand, for larger values of β the set of α -values where we achieve rates adaptively is increasing but the savings in terms of computing time is decreasing.

Finally, we note that if we have a fixed computational budget in terms of RAM and/or computing time, this trade-off can be approximately resolved in the following way. First, we consider a couple of candidates for β , or the resulting number of cells m . Then, we pick a suitably sized random subset of the entire training set and build Voronoi partitions of this random subset for the different candidates. For each cell of these partitions we then estimate the computational costs and finally we pick the largest candidate β for which the resulting partition still satisfies our computational budget. This procedure has several benefits: *a)* it is very cheap compared to the subsequent training and parameter selection phase, *b)* the choice of β , or m , has a clear meaning for the user, *c)* it approximately leads to widest adaptivity we can afford by our computational budget, and *d)* our experiments in the next section show that there is no significant risk for the user by focusing on the maximal computational resources.

5. Experimental Results

In this section we report a few experiments for VP-SVMs, which illustrate the influence of the chosen radius and which compare them to standard global SVMs as well as to RC-SVMs in terms of both training time and test error.

In the experiments we report here, we consider the classical COVTYPE data set, which contains 581.012 samples of dimension 54. More experimental results on additional data sets can be found in (Eberts, 2015) and in the earlier arXiv version (Eberts and Steinwart, 2014) of this paper. The code we used was an early version of Steinwart (2016), which provides

Algorithm 1 Determine a Voronoi partition of the input data

Require: Input data set $D_X = \{x_1, \dots, x_n\}$ with sample size $n \in \mathbb{N}$ and some radius $r > 0$.

Ensure: Working sets indicating a Voronoi partition of D_X .

```

1: Pick an arbitrary  $z \in D_X$ 
2:  $Cover_1 \leftarrow z$ 
3:  $m \leftarrow 1$ 
4: while  $\max_{x \in D_X} \|x - Cover\|_2 > r$  do
5:    $z \leftarrow \arg \max_{x \in D_X} \|x - Cover\|_2$ 
6:    $m \leftarrow m + 1$ 
7:    $Cover_m \leftarrow z$ 
8:    $WorkingSet_m \leftarrow \emptyset$ 
9: end while
10: for  $i = 1$  to  $n$  do
11:    $k \leftarrow \arg \min_{j \in \{1, \dots, m\}} \|x_i - Cover_j\|_2$ 
12:    $WorkingSet_k \leftarrow WorkingSet_k \cup \{x_i\}$ 
13: end for
14: return  $WorkingSet_1, \dots, WorkingSet_m$ 

```

highly efficient SVM solvers for different loss functions based on the ideas developed by (Steinwart et al., 2011). In particular, it is easy to repeat every experiment by the current version of the code.

In order to prepare the data set for the experiments, we first merged the split raw data sets so that we obtained one data set. In a next step, we scaled the data component-wise such that all samples including labels lie in $[-1, 1]^{d+1}$, where d is the dimension of the input data. Finally, we generated random subsets that were afterwards randomly split into a training and a test data set. In this manner, we obtained training sets consisting of $n = 1\,000$, $2\,500$, $5\,000$, $10\,000$, $25\,000$, $50\,000$, $100\,000$, $250\,000$, and $500\,000$ samples. The test data sets associated to the various training sets consist of $n_{\text{test}} = 50\,000$ random samples, apart from the training sets with $n_{\text{train}} \leq 5\,000$, for which we took $n_{\text{test}} = 10\,000$ test samples. To minimize random effects, we repeated the experiment for each setting several times. Since experiments using large data sets entail long run times, we reran every experiment using a training set of size $n \geq 50\,000$ only three times while for training sets of size $n = 10\,000$, $25\,000$ we performed ten repetitions and for smaller training sets, namely of size $n = 1\,000$, $2\,500$, $5\,000$, even 100 runs.

To train the global SVM for sufficiently large data sets we used a professional compute server equipped with four INTEL XEON E7-4830 (2.13 GHz) 8-core processor, 256 GB RAM. In order to have comparable run times, we ran the experiments for the VP-SVMs and RC-SVMs on this machine, too. In all experiments we used eight cores to pre-compute the kernel matrix and to evaluate the final decision functions on the test set, but only one core for the actual solver.

Let us quickly illustrate the routines of the VP- and the RC-SVM implemented around the LS-solver. For the VP-SVM, we first split the training set by Algorithm 1 in several working sets representing a Voronoi partition w.r.t. the user-specified radius. For this purpose, Algorithm 1 initially determines a cover of the input data applying the farthest

first traversal algorithm, see (Dasgupta, 2008) and (Gonzalez, 1985) for more details. Note that this procedure induces working sets whose sizes may be considerably varying. In the case of an RC-SVM the working sets form a random partition of the training samples, where their sizes are basically equal and the number of working sets is predefined by the user. Then, for the VP-SVM- as well as for the RC-SVM-algorithm the implemented LS-solver is applied on every working set. For each working set, we randomly split the respective training data set of size n_{train} in five folds to apply 5-fold cross-validation in order to deal with the hyper-parameters λ and γ taken from an 10 by 10 grid geometrically generated in $[0.001 \cdot n_{\text{train}}, 0.1] \times [0.5 \cdot n_{\text{train}}^{-1/d}, 10]$. Finally, we obtain one decision function for each working set. To further process these decision functions the VP-SVM-algorithms picks exactly one decision function depending on the working set affiliation of the input value. On the contrary, the RC-SVM-algorithm simply takes the average of all the decision functions. Moreover, the computed decision functions are clipped at ± 1 . Altogether, note that the usual LS-SVM-algorithm can be interpreted as special case of both the VP-SVM- and the RC-SVM-algorithm using one working set.

The results, which are displayed in Figure 5, can be quickly summarized: Not surprisingly, smaller radii for the VP-SVM lead to less crowded cells, which in turn reduces the training time significantly. In addition, the VP-SVM is, unlike the global SVM, not affected by the amount of available memory, so that runs with more than 100,000 samples, which would require kernel matrix caching for the global SVM, are still very feasible for the VP-SVM. Despite these advantages in terms of required computational resources, however, the test errors of the VP-SVM are only a bit worse than those of the global SVM. Moreover, the test errors become slightly better with increasing radii, so that there is a clear trade-off between computational resources and test accuracy as discussed in the previous section. When comparing the RC-SVM with the global SVM, we see, not surprisingly, the same computational advantages, but the test errors become significantly worse. As a consequence, the VP-SVM clearly outperforms the RC-SVM in terms of test errors, when both approaches have about the same training time. In this respect we also like to mention that in terms of test time, the VP-SVM was significantly faster than the RC-SVM, simply because for the VP-SVM each decision function evaluation only requires the support vector of the corresponding cell, whereas the final decision function of the RC-SVM requires all support vectors. See (Eberts and Steinwart, 2014) for details.

6. Proofs

This section is dedicated to prove the results of the previous sections.

We begin by recalling the definition of entropy and covering numbers. To this end, let (T, d) be a metric space. Then, the i -th (dyadic) entropy number of T is

$$\epsilon_i(T, d) := \inf \left\{ \varepsilon > 0 : \exists s_1, \dots, s_{2^{i-1}} \in T \text{ such that } T \subset \bigcup_{j=1}^{2^{i-1}} B(s_j, \varepsilon) \right\},$$

where $B_d(s, \varepsilon) := \{t \in T : d(t, s) \leq \varepsilon\}$ and $\inf \emptyset := \infty$. Moreover, if $S : E \rightarrow F$ is a bounded linear operator between the normed spaces E and F , then its (dyadic) entropy numbers are defined by $e_i(S : E \rightarrow F) := \epsilon_i(SB_E, \|\cdot\|_F)$, where B_E denotes the closed unit ball of E .

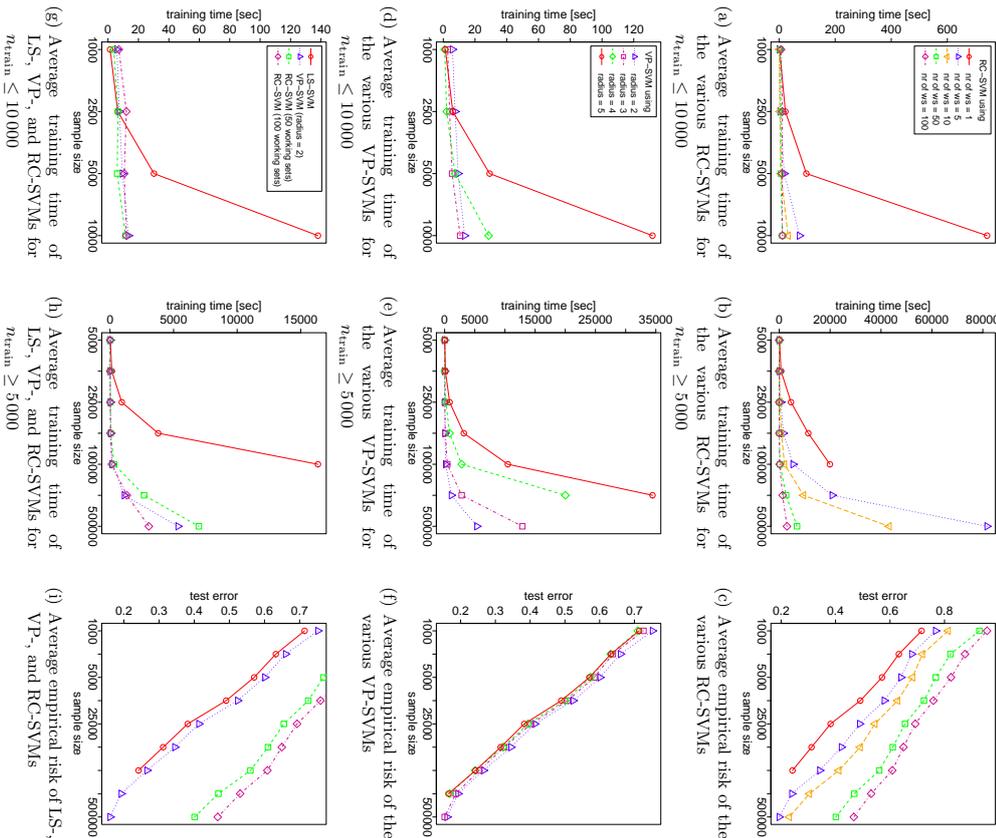


Figure 5: Average training time and test error of LS-, VP-, and RC-SVMs for the real-world data COVTYPE depending on the training set size $n_{\text{train}} = 1,000, \dots, 500,000$. Subfigures (a)–(c) show the results for RC-SVMs using different numbers of working sets and Subfigures (d)–(f) illustrate the results for VP-SVMs using various radii. At the bottom, Subfigures (g)–(i) contain the average training times and the average test errors of the LS-SVM, one VP-SVM and two RC-SVMs. Here, the VP-SVM is the one which trains fastest for $n_{\text{train}} = 500,000$ and the two RC-SVMs are those which achieve for $n_{\text{train}} = 500,000$ roughly the same training time as the chosen VP-SVM. Here, note that, for $n_{\text{train}} = 10,000$, the RC-SVM using one working set trains substantially slower than the LS-SVM, even though this RC-SVM is basically an LS-SVM. As a reason for this phenomenon, we conjecture that the used compute server was busy because of other influences.

Similarly, the ε -covering number of T is defined by

$$\mathcal{N}(T, d, \varepsilon) := \inf \left\{ n \geq 1 : \exists s_1, \dots, s_n \in T \text{ such that } T \subset \bigcup_{i=1}^n B_d(s_i, \varepsilon) \right\},$$

and again, this definition can be applied to bounded linear operators $S : E \rightarrow F$ by considering the set SBE . Moreover, every subset $S \subset T$ for which for all $t \in T$ there exists an $s \in S$ with $d(s, t) \leq \varepsilon$ is called an ε -net of T . Consequently, $\mathcal{N}(T, d, \varepsilon)$ is the size of the smallest ε -net of T . Recall that entropy and covering numbers are in some sense inverse to each other. To be more precise, for all constants $a > 0$ and $q > 0$, the implication

$$e_i(T, d) \leq a i^{-1/q}, \quad i \geq 1 \quad \implies \quad \ln \mathcal{N}(T, d, \varepsilon) \leq \ln(4) \left(\frac{a}{\varepsilon} \right)^q, \quad \forall \varepsilon > 0 \quad (20)$$

holds by (Steinwart and Christmann, 2008, Lemma 6.21). Additionally, (Steinwart and Christmann, 2008, Exercise 6.8) yields the opposite implication, namely

$$\ln \mathcal{N}(T, d, \varepsilon) < \left(\frac{a}{\varepsilon} \right)^q, \quad \varepsilon > 0 \quad \implies \quad e_i(T, d) \leq 3^{1/q} a i^{-1/q}, \quad \forall i \geq 1. \quad (21)$$

With these preparations, we can now prove Lemma 1, which relates the radius r of a cover $B_{r_1}(z_1), \dots, B_{r_m}(z_m)$ of $B_{\tilde{r}} \supset X$ defined by (4) with the number m of centers z_1, \dots, z_m .

Proof [of Lemma 1] It is easy to show that $\mathcal{N}(cB_{\tilde{r}}(c\tilde{r}), r) = \mathcal{N}(B_{\tilde{r}}(c\tilde{r}), \tilde{r})$ holds for all $r, c > 0$. Moreover, applying Proposition 1.1 of (Temlyakov, 2013) yields

$$r^{-d} \leq \mathcal{N}(B_{\tilde{r}}(c\tilde{r}), \tilde{r}) \leq \left(1 + \frac{2}{\tilde{r}} \right)^d, \quad \tilde{r} \in (0, 1].$$

Consequently, we can find a cover $(B_{r_i}(z_i))_{i=1, \dots, m}$ of $X \subset cB_{\tilde{r}}(c\tilde{r})$ with centers $z_j \in cB_{\tilde{r}}(c\tilde{r})$ and radius $r \leq c$ such that

$$\left(\frac{r}{c} \right)^{-d} \leq m \leq \left(1 + \frac{2c}{r} \right)^d.$$

Since $r \leq c$, we thus have $r \leq (r + 2c) m^{-\frac{1}{d}} \leq 3cm^{-\frac{1}{d}}$. \blacksquare

Next, we consider a lemma that is part of our construction of the partition $(A_j)_j$ of X .

Lemma 8 *Let $(A'_j)_{j=1, \dots, m}$ be a partition of $B_{\tilde{r}}(c\tilde{r})$ such that $A'_j \neq \emptyset$ as well as $\overline{A'_j} = \overline{A_j}$ for every $j \in \{1, \dots, m\}$. Let \tilde{X} be some closed subset of $B_{\tilde{r}}(c\tilde{r})$ such that $\tilde{X} \neq \emptyset$ and $\tilde{X} = X$.*

Without loss of generality we further assume that there is an $m_0 \leq m$ such that $A'_j \cap \tilde{X} \neq \emptyset$ for all $j \in \{1, \dots, m_0\}$ and $A'_j \cap \tilde{X} = \emptyset$ for all $j \in \{m_0 + 1, \dots, m\}$. Then, we define $A''_j := A'_j \cap \tilde{X}$ for all $j \in \{1, \dots, m_0\}$. Moreover, let $(A_j)_{j=1, \dots, m_0}$ be a partition of \tilde{X} with $A''_j \subset A_j \subset \overline{A''_j}$. Then, for every $j \in \{1, \dots, m_0\}$, we have $A''_j \neq \emptyset$, and thus $A_j \neq \emptyset$.

Proof Let us assume that there is an $j \in \{1, \dots, m_0\}$ with $\dot{A}_j'' = \emptyset$. By our assumption we then know $A_j'' \cap \dot{X} \neq \emptyset$, i.e., there exists some $x \in A_j'' \cap \dot{X}$. Since

$$\emptyset = \dot{A}_j'' = \text{interior}(A_j'' \cap \dot{X}) = \dot{A}_j'' \cap \text{interior} \dot{X} = \dot{A}_j'' \cap \dot{X},$$

where we used the notation $\text{interior } B := \overset{\circ}{B}$, it immediately follows that $x \in \partial A_j'' \subset \overline{A_j''} = \dot{A}_j''$. Hence, there exists a sequence $(x_n)_n \subset \dot{A}_j''$ such that $x_n \xrightarrow{n \rightarrow \infty} x$. On the other hand, $x \in A_j'' \subset \dot{X}$ together with the fact that \dot{X} is open, gives $x_n \in \dot{X}$ for all sufficiently large n . For such an n , we obtain $x_n \in \dot{A}_j'' \cap \dot{X} = \dot{A}_j''$, which contradicts the assumed $\dot{A}_j'' = \emptyset$. The second assertion follows from $\dot{A}_j'' \subset \dot{A}_j$. ■

Next, let us consider a crucial property of the risk of functions contained in a joined RKHS.

Lemma 9 Let P be a distribution on $X \times Y$ and $L : X \times Y \times \mathbb{R} \rightarrow [0, \infty)$ be a loss function. For $A, B \subset X$ such that $A \cup B = X$ and $A \cap B = \emptyset$, define loss functions $L_A, L_B : X \times Y \times \mathbb{R} \rightarrow [0, \infty)$ by $L_A(x, y, t) = \mathbb{1}_A(x)L(x, y, t)$ and $L_B(x, y, t) = \mathbb{1}_B(x)L(x, y, t)$, respectively. Furthermore, let $f_A : X \rightarrow \mathbb{R}$ as well as $f_B : X \rightarrow \mathbb{R}$ be measurable functions and $f : X \rightarrow \mathbb{R}$ be defined by $f(x) = \mathbb{1}_A(x)f_A(x) + \mathbb{1}_B(x)f_B(x)$ for all $x \in X$. Then, we have

$$\mathcal{R}_{L, P}(f) = \mathcal{R}_{L_A, P}(f_A) + \mathcal{R}_{L_B, P}(f_B).$$

as well as

$$\mathcal{R}_{L, P}(f) - \mathcal{R}_{L, P}^*(f) = (\mathcal{R}_{L_A, P}(f_A) - \mathcal{R}_{L_A, P}^*(f_A)) + (\mathcal{R}_{L_B, P}(f_B) - \mathcal{R}_{L_B, P}^*(f_B)).$$

Proof Simple transformations using $A \cup B = X$ and $A \cap B = \emptyset$ show

$$\begin{aligned} \mathcal{R}_{L, P}(f) &= \int_{X \times Y} L(x, y, \mathbb{1}_A(x)f_A(x) + \mathbb{1}_B(x)f_B(x)) dP(x, y) \\ &= \int_{X \times Y} \mathbb{1}_A(x)L(x, y, f_A(x)) + \mathbb{1}_B(x)L(x, y, f_B(x)) dP(x, y) \\ &= \mathcal{R}_{L_A, P}(f_A) + \mathcal{R}_{L_B, P}(f_B). \end{aligned}$$

The second assertion follows immediately. ■

6.1 Some General Estimates on Entropy Numbers

To derive an oracle inequality for VP-SVMs we will have to relate the entropy numbers of H_j , $j \in \{1, \dots, m\}$, to those of H . Our first result establishes such a relationship for covering numbers, instead.

Lemma 10 Let ν be a distribution on X and $A, B \subset X$ with $A \cap B = \emptyset$. Moreover, let H_A and H_B be RKHSs on A and B that are embedded into $L_2(\nu_A)$ and $L_2(\nu_B)$, respectively. Let the extended RKHSs \hat{H}_A and \hat{H}_B be defined as in Lemma 2 and denote their direct sum by H as in (11), where the norm is given by (12) with $\lambda_A, \lambda_B > 0$. Then, for the ε -covering number of H w.r.t. $\|\cdot\|_{L_2(\nu)}$, we have

$$\mathcal{N}(B_H, \|\cdot\|_{L_2(\nu)}, \varepsilon) \leq \mathcal{N}\left(\lambda_A^{-1/2} B_{\hat{H}_A}, \|\cdot\|_{L_2(\nu_A)}, \varepsilon_A\right) \cdot \mathcal{N}\left(\lambda_B^{-1/2} B_{\hat{H}_B}, \|\cdot\|_{L_2(\nu_B)}, \varepsilon_B\right),$$

where $\varepsilon_A, \varepsilon_B > 0$ and $\varepsilon := \sqrt{\varepsilon_A^2 + \varepsilon_B^2}$.

Proof First of all, we assume that there exist $a, b \in \mathbb{N}$ and functions $\hat{f}_1, \dots, \hat{f}_a \in \lambda_A^{-1/2} B_{\hat{H}_A}$ and $\hat{h}_1, \dots, \hat{h}_b \in \lambda_B^{-1/2} B_{\hat{H}_B}$ such that $\{\hat{f}_1, \dots, \hat{f}_a\}$ is an ε_A -cover of $\lambda_A^{-1/2} B_{\hat{H}_A}$ w.r.t. $\|\cdot\|_{L_2(\nu_A)}$, $\{\hat{h}_1, \dots, \hat{h}_b\}$ is an ε_B -cover of $\lambda_B^{-1/2} B_{\hat{H}_B}$ w.r.t. $\|\cdot\|_{L_2(\nu_B)}$,

$$a = \mathcal{N}\left(\lambda_A^{-1/2} B_{\hat{H}_A}, \|\cdot\|_{L_2(\nu_A)}, \varepsilon_A\right) \quad \text{and} \quad b = \mathcal{N}\left(\lambda_B^{-1/2} B_{\hat{H}_B}, \|\cdot\|_{L_2(\nu_B)}, \varepsilon_B\right). \quad (22)$$

That is, for every function $\hat{g}_A \in \lambda_A^{-1/2} B_{\hat{H}_A}$, there exists an $i_A \in \{1, \dots, a\}$ such that

$$\|\hat{g}_A - \hat{f}_{i_A}\|_{L_2(\nu_A)} \leq \varepsilon_A, \quad (22)$$

and for every function $\hat{g}_B \in \lambda_B^{-1/2} B_{\hat{H}_B}$, there exists an $i_B \in \{1, \dots, b\}$ such that

$$\|\hat{g}_B - \hat{h}_{i_B}\|_{L_2(\nu_B)} \leq \varepsilon_B. \quad (23)$$

Let us now consider an arbitrary function $g \in B_H$. Then, there exists an $\hat{g}_A \in \lambda_A^{-1/2} B_{\hat{H}_A}$ and an $\hat{g}_B \in \lambda_B^{-1/2} B_{\hat{H}_B}$ such that $g = \hat{g}_A + \hat{g}_B$. Together with (22) and (23), this implies

$$\begin{aligned} \left\|g - \left(\hat{f}_{i_A} + \hat{h}_{i_B}\right)\right\|_{L_2(\nu)}^2 &= \left\|\left(\hat{g}_A - \hat{f}_{i_A}\right) + \left(\hat{g}_B - \hat{h}_{i_B}\right)\right\|_{L_2(\nu)}^2 \\ &= \left\|\hat{g}_A - \hat{f}_{i_A}\right\|_{L_2(\nu_A)}^2 + \left\|\hat{g}_B - \hat{h}_{i_B}\right\|_{L_2(\nu_B)}^2 \\ &\leq \varepsilon_A^2 + \varepsilon_B^2 \\ &=: \varepsilon^2. \end{aligned}$$

With this, we know that

$$\left\{\hat{f}_{i_A} + \hat{h}_{i_B} : \hat{f}_{i_A} \in \{\hat{f}_1, \dots, \hat{f}_a\} \text{ and } \hat{h}_{i_B} \in \{\hat{h}_1, \dots, \hat{h}_b\}\right\}$$

is an ε -net of H w.r.t. $\|\cdot\|_{L_2(\nu)}$. Concerning the ε -covering number of H , this finally implies $\mathcal{N}(B_H, \|\cdot\|_{L_2(\nu)}, \varepsilon) \leq a \cdot b = \mathcal{N}\left(\lambda_A^{-1/2} B_{\hat{H}_A}, \|\cdot\|_{L_2(\nu_A)}, \varepsilon_A\right) \cdot \mathcal{N}\left(\lambda_B^{-1/2} B_{\hat{H}_B}, \|\cdot\|_{L_2(\nu_B)}, \varepsilon_B\right)$.

Based on Lemma 10, the following theorem relates entropy numbers of H_A and H_B to those of H . ■

Theorem 11 *Let P_X be a distribution on X and $A_1, \dots, A_m \subset X$ be pairwise disjoint. Moreover, for $j \in \{1, \dots, m\}$, let H_j be a separable RKHS of a measurable kernel k_j over A_j such that $\|k_j\|_{L_2(P_{X|A_j})}^2 := \int_X k_j(x, x) dP_{X|A_j}(x) < \infty$. Define RKHSs $\hat{H}_1, \dots, \hat{H}_m$ by Lemma 2 and the joined RKHS H by (13) with the norm (14) and weights $\lambda_1, \dots, \lambda_m > 0$. In addition, assume that there exist constants $p \in (0, 1)$ and $a_j > 0$, $j \in \{1, \dots, m\}$, such that for every $j \in \{1, \dots, m\}$*

$$\epsilon_i(\text{id} : H_j \rightarrow L_2(P_{X|A_j})) \leq a_j i^{-\frac{1}{2p}}, \quad i \geq 1. \quad (24)$$

Then, we have

$$\epsilon_i(\text{id} : H \rightarrow L_2(P_X)) \leq 2\sqrt{m} \left(3 \ln(4) \sum_{j=1}^m \lambda_j^{-p} a_j^{2p} \right)^{\frac{1}{2p}} i^{-\frac{1}{2p}}, \quad i \geq 1,$$

and, for the average entropy numbers,

$$\mathbb{E}_{D_X \sim P_X^n} \epsilon_i(\text{id} : H \rightarrow L_2(D_X)) \leq c_p \sqrt{m} \left(\sum_{j=1}^m \lambda_j^{-p} a_j^{2p} \right)^{\frac{1}{2p}} i^{-\frac{1}{2p}}, \quad i, n \geq 1.$$

Proof First of all, note that the restriction operator $\mathcal{I} : B_{H_j} \rightarrow B_{H_j}$ with $\mathcal{I}f = f$ is an isometric isomorphism. Together with (Steinwart and Christmann, 2008, (A.36)) and assumption (24), this yields

$$\begin{aligned} \epsilon_i(\lambda_j^{-\frac{1}{2}} B_{H_j}, L_2(P_{X|A_j})) &= 2\lambda_j^{-\frac{1}{2}} \epsilon_i(B_{H_j}, L_2(P_{X|A_j})) \\ &\leq 2\lambda_j^{-\frac{1}{2}} \|\mathcal{I} : B_{H_j} \rightarrow B_{H_j}\| \epsilon_i(B_{H_j}, L_2(P_{X|A_j})) \\ &\leq 2\lambda_j^{-\frac{1}{2}} a_j i^{-\frac{1}{2p}}. \end{aligned}$$

Furthermore, we know by (20) that

$$\ln \mathcal{N} \left(\lambda_j^{-\frac{1}{2}} B_{H_j}, \|\cdot\|_{L_2(P_{X|A_j})}, \varepsilon \right) \leq \ln(4) \left(2\lambda_j^{-\frac{1}{2}} a_j \right)^{2p} \varepsilon^{-2p}$$

holds for all $\varepsilon > 0$. With this and $\varepsilon_j := \frac{\varepsilon}{\sqrt{m}}$ for every $j \in \{1, \dots, m\}$, Lemma 10 implies

$$\ln \mathcal{N}(B_H, \|\cdot\|_{L_2(P_X)}, \varepsilon) \leq \ln \left(\prod_{j=1}^m \mathcal{N} \left(\lambda_j^{-\frac{1}{2}} B_{H_j}, \|\cdot\|_{L_2(P_{X|A_j})}, \varepsilon_j \right) \right)$$

$$\begin{aligned} &= \sum_{j=1}^m \ln \mathcal{N} \left(\lambda_j^{-\frac{1}{2}} B_{H_j}, \|\cdot\|_{L_2(P_{X|A_j})}, \frac{\varepsilon}{\sqrt{m}} \right) \\ &\leq \sum_{j=1}^m \ln(4) \left(2\lambda_j^{-\frac{1}{2}} a_j \right)^{2p} \left(\frac{\sqrt{m}}{\varepsilon} \right)^{2p} \\ &= \left(2 \ln(4) \right)^{\frac{1}{2p}} \sqrt{m} \left(\sum_{j=1}^m \lambda_j^{-p} a_j^{2p} \right)^{\frac{1}{2p}} \varepsilon^{-2p}. \end{aligned}$$

Using (21), the latter bound for the covering number of B_H finally implies the following entropy estimate

$$\begin{aligned} \epsilon_i(\text{id} : H \rightarrow L_2(P_X)) &\leq 3^{\frac{1}{2p}} \left(2 \ln(4) \right)^{\frac{1}{2p}} \sqrt{m} \left(\sum_{j=1}^m \lambda_j^{-p} a_j^{2p} \right)^{\frac{1}{2p}} i^{-\frac{1}{2p}} \\ &\leq 2 \left(3 \ln(4) \right)^{\frac{1}{2p}} \sqrt{m} \left(\sum_{j=1}^m \lambda_j^{-p} a_j^{2p} \right)^{\frac{1}{2p}} i^{-\frac{1}{2p}}. \end{aligned}$$

The second assertion immediately follows by (Steinwart and Christmann, 2008, Corollary 7.31). ■

In the following subsections, we first focus on RKHSs using Gaussian RBF kernels and examine the associated entropy numbers to specify (24). Subsequently, we additionally consider the least squares loss to prove Theorem 4.

6.2 Entropy Estimates for Local Gaussian RKHSs

In this subsection, we derive an estimate in terms of assumption (24) for the RKHS $H_{\gamma_1}(A)$ over A of the Gaussian RBF kernel k_{γ_1} on $A \subset \mathbb{R}^d$ given by

$$k_{\gamma_1}(x, x') := \exp(-\gamma^{-2} \|x - x'\|_2^2), \quad x, x' \in A,$$

for some width $\gamma > 0$. More precisely, in the subsequent theorem we determine an upper bound for the entropy numbers of the operator $\text{id} : H_{\gamma_1}(A) \rightarrow L_2(P_{X|A})$.

Theorem 12 *Let $X \subset \mathbb{R}^d$, P_X be a distribution on X and $A \subset X$ be such that $\hat{A} \neq \emptyset$ and such that there exists an Euclidean ball $B \subset \mathbb{R}^d$ with radius $r > 0$ containing A , i.e., $A \subset B$. Moreover, for $0 < \gamma \leq r$, let $H_{\gamma_1}(A)$ be the RKHS of the Gaussian RBF kernel k_{γ_1} over A . Then, for all $p \in (0, 1)$, there exists a constant $c_p > 0$ such that*

$$\epsilon_i(\text{id} : H_{\gamma_1}(A) \rightarrow L_2(P_{X|A})) \leq c_p \sqrt{P_X(A)} r^{\frac{d+2p}{2p}} \gamma^{-\frac{d+2p}{2p}} i^{-\frac{1}{2p}}, \quad i \geq 1.$$

Proof First of all, we consider the commutative diagram

$$\begin{array}{ccc} H_\gamma(A) & \xrightarrow{\text{id}} & L_2(\mathbb{P}_{X|A}) \\ \mathcal{I}_B^{-1} \circ \mathcal{I}_A \downarrow & & \uparrow \text{id} \\ H_\gamma(B) & \xrightarrow{\text{id}} & \ell_\infty(B) \end{array}$$

where the extension operator $\mathcal{I}_A : H_\gamma(A) \rightarrow H_\gamma(\mathbb{R}^d)$ and the restriction operator $\mathcal{I}_B^{-1} : H_\gamma(\mathbb{R}^d) \rightarrow H_\gamma(B)$ given by (Steinwart and Christmann, 2008, Corollary 4.43) are isometric isomorphisms, so that $\|\mathcal{I}_B^{-1} \circ \mathcal{I}_A : H_\gamma(A) \rightarrow H_\gamma(B)\| = 1$. Furthermore, for $f \in \ell_\infty(B)$, where $\ell_\infty(B)$ is the space of all bounded functions on B , we have

$$\|f\|_{L_2(\mathbb{P}_{X|A})} = \left(\int_X \mathbb{1}_A(x) |f(x)|^2 d\mathbb{P}_X(x) \right)^{\frac{1}{2}} \leq \|f\|_\infty \left(\int_X \mathbb{1}_A(x) d\mathbb{P}_X(x) \right)^{\frac{1}{2}} = \sqrt{\mathbb{P}_X(A)} \|f\|_\infty,$$

i.e., $\|\text{id} : \ell_\infty(B) \rightarrow L_2(\mathbb{P}_{X|A})\| \leq \sqrt{\mathbb{P}_X(A)}$. Together with (Steinwart and Christmann, 2008, (A.38) and (A.39)) as well as (Steinwart and Christmann, 2008, Theorem 6.27), we obtain for all $t \geq 1$

$$\begin{aligned} \epsilon_t(\text{id} : H_\gamma(A) &\rightarrow L_2(\mathbb{P}_{X|A})) \\ &\leq \|\mathcal{I}_B^{-1} \circ \mathcal{I}_A : H_\gamma(A) \rightarrow H_\gamma(B)\| \cdot \epsilon_t(\text{id} : H_\gamma(B) \rightarrow \ell_\infty(B)) \cdot \|\text{id} : \ell_\infty(B) \rightarrow L_2(\mathbb{P}_{X|A})\| \\ &\leq \sqrt{\mathbb{P}_X(A)} c_{m,d} t^m \gamma^{-m} t^{-\frac{m}{2}}, \end{aligned}$$

where $m \geq 1$ is an arbitrary integer and $c_{m,d}$ a positive constant. For $p \in (0, 1)$, the choice $m = \lfloor \frac{d}{2p} \rfloor$ finally yields

$$\epsilon_t(\text{id} : H_\gamma(A) \rightarrow L_2(\mathbb{P}_{X|A})) \leq \sqrt{\mathbb{P}_X(A)} c_{m,d} t^m \gamma^{-m} t^{-\frac{m}{2}} \leq c_p \sqrt{\mathbb{P}_X(A)} r^{-\frac{dt}{2p}} \gamma^{-\frac{dt}{2p}} t^{-\frac{1}{2p}}.$$

■

6.3 Proofs Related to the Least Squares VP-SVMs

In this subsection, we prove the results that are linked with the least squares loss, i.e., the results of Section 4. Before we elaborate on the oracle inequality for VP-SVMs using the least squares loss as well as RKHSs of Gaussian kernels, we have to examine the excess risk

$$\mathcal{R}_{L_{J_T, P}}(f_0) - \mathcal{R}_{L_{J_T, P}}^* = \|f_0 - f_{L_T, P}^*\|_{L_2(\mathbb{P}_{X|A_T})}^2. \quad (25)$$

Let us begin by writing for fixed $\gamma_j > 0$

$$K_j : \mathbb{R}^d \rightarrow \mathbb{R}, \quad x \mapsto \sum_{\ell=1}^s \binom{s}{\ell} (-1)^{1-\ell} \left(\frac{2}{\ell^2 \gamma_j^2 \pi} \right)^{\frac{\ell}{2}} \exp\left(-\frac{2\|x\|_2^2}{\ell^2 \gamma_j^2}\right), \quad (26)$$

and choosing $f_0 := \sum_{j=1}^m \mathbb{1}_{A_j} \cdot (K_j * f_{L_T, P}^*)$. Then, (25) can be estimated with the help of the following theorem, which is together with its proof basically a modification of (Eberts and Steinwart, 2013, Theorem 2.2). Indeed, the proofs proceed mainly identically. Note that we use the notation

$$\gamma_{\max} := \max\{\gamma_1, \dots, \gamma_m\} \quad \text{and} \quad \gamma_{\min} := \min\{\gamma_1, \dots, \gamma_m\}$$

in the following theorem and the associated proof. For the sake of generality, we do not only consider the Besov-like space $B_{q, \infty}^{2s}(\nu)$ in the following theorem but instead the Besov-like spaces $B_{q, \infty}^\alpha(\nu)$ for arbitrary $q \in [1, \infty)$. These Besov-like spaces are defined analogously to $B_{2, \infty}^\alpha(\nu)$, however, applying the modulus of smoothness for the $L_q(\nu)$ -norm instead of the $L_2(\nu)$ -norm. For an explicit definition of these spaces we refer to (Eberts, 2015, Section 3.1)

Theorem 13 *Let us fix some $q \in [1, \infty)$. Assume that ν is a finite measure on \mathbb{R}^d with $\text{supp } \nu := X \subset cB_{q, \infty}^d \subset \mathbb{R}^d$ for some $c > 0$. Let $(A'_j)_{j=1, \dots, m}$ be a partition of $cB_{q, \infty}^d$. Then, $A_j := A'_j \cap X$ for all $j \in \{1, \dots, m\}$ defines a partition $(A_j)_{j=1, \dots, m}$ of X . Furthermore, let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be such that $f \in B_{q, \infty}^\alpha(\nu)$ for some $\alpha \geq 1$. For the functions $K_j : \mathbb{R}^d \rightarrow \mathbb{R}$, $j \in \{1, \dots, m\}$, defined by (26), where $s := \lfloor \alpha \rfloor + 1$ and $\gamma_1, \dots, \gamma_m > 0$, we then have*

$$\left\| \sum_{j=1}^m \mathbb{1}_{A_j} \cdot (K_j * f) - f \right\|_{L_q(\nu)} \leq C_{\alpha, q} \left(\frac{\gamma_{\max}}{\gamma_{\min}} \right)^{\frac{q\alpha}{2}} \gamma_{\max}^{\alpha},$$

where $C_{\alpha, q} := \|f\|_{B_{q, \infty}^\alpha(\nu)}^q \left(\frac{d}{2}\right)^{\frac{q\alpha}{2}} \pi^{-\frac{1}{2}} \Gamma\left(q\alpha + \frac{1}{2}\right)^{\frac{1}{2}}$.

Proof In the following, we write $J := \{1, \dots, m\}$. To show

$$\left\| \sum_{j \in J} \mathbb{1}_{A_j} \cdot (K_j * f) - f \right\|_{L_q(\nu)}^q \leq \|f\|_{B_{q, \infty}^\alpha(\nu)}^q \left(\frac{d}{2}\right)^{\frac{q\alpha}{2}} \pi^{-\frac{1}{2}} \Gamma\left(q\alpha + \frac{1}{2}\right)^{\frac{1}{2}} \left(\frac{\gamma_{\max}}{\gamma_{\min}}\right)^{\frac{q\alpha}{2}} \gamma_{\max}^{\alpha},$$

we have to proceed in a similar way as in the proof of (Eberts and Steinwart, 2013, Theorem 2.2). First of all, we use the translation invariance of the Lebesgue measure and $\exp(-\|u\|_2^2) = \exp(-\| -u \|^2)$ ($u \in \mathbb{R}^d$) to obtain, for $x \in X$ and $j \in J$,

$$\begin{aligned} K_j * f(x) &= \int_{\mathbb{R}^d} \sum_{\ell=1}^s \binom{s}{\ell} (-1)^{1-\ell} \left(\frac{2}{\gamma_j^2 \pi}\right)^{\frac{\ell}{2}} \exp\left(-\frac{2\|x-t\|_2^2}{\ell^2 \gamma_j^2}\right) f(t) dt \\ &= \int_{\mathbb{R}^d} \left(\frac{2}{\gamma_j^2 \pi}\right)^{\frac{\ell}{2}} \exp\left(-\frac{2\|h\|_2^2}{\gamma_j^2}\right) \left(\sum_{\ell=1}^s \binom{s}{\ell} (-1)^{1-\ell} f(x+th)\right) dh. \end{aligned}$$

With this we can derive, for $q \geq 1$,

$$\left\| \sum_{j \in J} \mathbb{1}_{A_j} \cdot (K_j * f) - f \right\|_{L_q(\nu)}^q$$

$$\begin{aligned}
&= \int_{\mathbb{R}^d} \left| \sum_{j \in J} \mathbb{1}_{A_j}(x) (K_j * f)(x) - f(x) \right|_{L_q(\nu)}^q dv(x) \\
&\leq \int_{\mathbb{R}^d} \left(\sum_{j \in I} \mathbb{1}_{A_j}(x) |K_j * f(x) - f(x)| \right)^q dv(x) \\
&= \int_{\mathbb{R}^d} \sum_{j \in J} \mathbb{1}_{A_j}(x) |K_j * f(x) - f(x)|^q dv(x) \\
&= \sum_{j \in J} \int_{\mathbb{R}^d} \mathbb{1}_{A_j}(x) |K_j * f(x) - f(x)|^q dv(x) \\
&= \sum_{j \in J} \int_{\mathbb{R}^d} \mathbb{1}_{A_j}(x) \left| \int_{\mathbb{R}^d} \left(\frac{2}{\gamma_j^2 \pi} \right)^{\frac{d}{2}} \exp\left(-\frac{2\|h\|_2^2}{\gamma_j^2}\right) \left(\sum_{\ell=0}^s \binom{s}{\ell} (-1)^{2s+1-\ell} f(x+\ell h) \right) dh \right|^q dv(x) \\
&= \sum_{j \in J} \int_{\mathbb{R}^d} \mathbb{1}_{A_j}(x) \left| \int_{\mathbb{R}^d} (-1)^{s+1} \left(\frac{2}{\gamma_j^2 \pi} \right)^{\frac{d}{2}} \exp\left(-\frac{2\|h\|_2^2}{\gamma_j^2}\right) \Delta_h^s(f, x) dh \right|^q dv(x) \\
&\leq \sum_{j \in J} \int_{\mathbb{R}^d} \mathbb{1}_{A_j}(x) \left(\int_{\mathbb{R}^d} \left(\frac{2}{\gamma_j^2 \pi} \right)^{\frac{d}{2}} \exp\left(-\frac{2\|h\|_2^2}{\gamma_j^2}\right) |\Delta_h^s(f, x)| dh \right)^q dv(x).
\end{aligned}$$

Then, Hölder's inequality and $\int_{\mathbb{R}^d} \exp(-2\gamma_j^{-2}\|h\|_2^2) dh = \left(\frac{\gamma_j^2 \pi}{2}\right)^{d/2}$ yield, for $q > 1$,

$$\begin{aligned}
&\left\| \sum_{j \in J} \mathbb{1}_{A_j} \cdot (K_j * f) - f \right\|_{L_q(\nu)}^q \\
&\leq \sum_{j \in J} \int_{\mathbb{R}^d} \mathbb{1}_{A_j}(x) \left(\int_{\mathbb{R}^d} \left(\frac{2}{\gamma_j^2 \pi} \right)^{\frac{d}{2}} \exp\left(-\frac{2\|h\|_2^2}{\gamma_j^2}\right) dh \right)^{\frac{q-1}{q}} \\
&\quad \left(\int_{\mathbb{R}^d} \left(\frac{2}{\gamma_j^2 \pi} \right)^{\frac{d}{2}} \exp\left(-\frac{2\|h\|_2^2}{\gamma_j^2}\right) |\Delta_h^s(f, x)|^q dh \right)^{\frac{1}{q}} dv(x) \\
&= \sum_{j \in J} \int_{\mathbb{R}^d} \mathbb{1}_{A_j}(x) \int_{\mathbb{R}^d} \left(\frac{2}{\gamma_j^2 \pi} \right)^{\frac{d}{2}} \exp\left(-\frac{2\|h\|_2^2}{\gamma_j^2}\right) |\Delta_h^s(f, x)|^q dh dv(x) \\
&= \sum_{j \in J} \int_{\mathbb{R}^d} \left(\frac{2}{\gamma_j^2 \pi} \right)^{\frac{d}{2}} \exp\left(-\frac{2\|h\|_2^2}{\gamma_j^2}\right) \int_{\mathbb{R}^d} \mathbb{1}_{A_j}(x) |\Delta_h^s(f, x)|^q dv(x) dh \\
&\leq \int_{\mathbb{R}^d} \left(\frac{2}{\pi \gamma_{\min}^2} \right)^{\frac{d}{2}} \exp\left(-\frac{2\|h\|_2^2}{\gamma_{\max}^2}\right) \int_{\mathbb{R}^d} \mathbb{1}_{A_j}(x) |\Delta_h^s(f, x)|^q dv(x) dh
\end{aligned}$$

$$\begin{aligned}
&= \int_{\mathbb{R}^d} \left(\frac{2}{\pi \gamma_{\min}^2} \right)^{\frac{d}{2}} \exp\left(-\frac{2\|h\|_2^2}{\gamma_{\max}^2}\right) \|\Delta_h^s(f, \cdot)\|_{L_q(\nu)}^q dh \\
&\leq \int_{\mathbb{R}^d} \left(\frac{2}{\pi \gamma_{\min}^2} \right)^{\frac{d}{2}} \exp\left(-\frac{2\|h\|_2^2}{\gamma_{\max}^2}\right) \omega_{s, L_q(\nu)}^q(f, \|h\|_2) dh.
\end{aligned}$$

Moreover, for $q = 1$, we have

$$\begin{aligned}
&\left\| \sum_{j \in I} \mathbb{1}_{A_j} \cdot (K_j * f) - f \right\|_{L_1(\nu)} \\
&\leq \sum_{j \in I} \int_{\mathbb{R}^d} \mathbb{1}_{A_j}(x) \int_{\mathbb{R}^d} \left(\frac{2}{\gamma_j^2 \pi} \right)^{\frac{d}{2}} \exp\left(-\frac{2\|h\|_2^2}{\gamma_j^2}\right) |\Delta_h^s(f, x)| dh dv(x) \\
&\leq \int_{\mathbb{R}^d} \left(\frac{2}{\pi \gamma_{\min}^2} \right)^{\frac{d}{2}} \exp\left(-\frac{2\|h\|_2^2}{\gamma_{\max}^2}\right) \int_{\mathbb{R}^d} \mathbb{1}_{A_j}(x) |\Delta_h^s(f, x)| dv(x) dh \\
&\leq \int_{\mathbb{R}^d} \left(\frac{2}{\pi \gamma_{\min}^2} \right)^{\frac{d}{2}} \exp\left(-\frac{2\|h\|_2^2}{\gamma_{\max}^2}\right) \omega_{s, L_1(\nu)}(f, \|h\|_2) dh.
\end{aligned}$$

Consequently, we can proceed in the same way for all $q \geq 1$. To this end, note that the assumption $f \in B_{q, \infty}^s(\nu)$ implies $\omega_{s, L_q(\nu)}(f, t) \leq \|f\|_{B_{q, \infty}^s(\nu)} t^{\alpha}$ for $t > 0$. The latter together with Hölder's inequality yields

$$\begin{aligned}
&\left\| \sum_{j \in I} \mathbb{1}_{A_j} \cdot (K_j * f) - f \right\|_{L_q(\nu)}^q \\
&\leq \int_{\mathbb{R}^d} \left(\frac{2}{\pi \gamma_{\min}^2} \right)^{\frac{d}{2}} \exp\left(-\frac{2\|h\|_2^2}{\gamma_{\max}^2}\right) \omega_{s, L_q(\nu)}^q(f, \|h\|_2) dh \\
&\leq \|f\|_{B_{q, \infty}^s(\nu)}^q \left(\frac{2}{\pi \gamma_{\min}^2} \right)^{\frac{d}{2}} \int_{\mathbb{R}^d} \|h\|_2^{q\alpha} \exp\left(-\frac{2\|h\|_2^2}{\gamma_{\max}^2}\right) dh \\
&\leq \|f\|_{B_{q, \infty}^s(\nu)}^q \left(\frac{2}{\pi \gamma_{\min}^2} \right)^{\frac{d}{2}} \left(\int_{\mathbb{R}^d} \exp\left(-\frac{2\|h\|_2^2}{\gamma_{\max}^2}\right) dh \right)^{\frac{1}{2}} \left(\int_{\mathbb{R}^d} \|h\|_2^{2q\alpha} \exp\left(-\frac{2\|h\|_2^2}{\gamma_{\max}^2}\right) dh \right)^{\frac{1}{2}} \\
&= \|f\|_{B_{q, \infty}^s(\nu)}^q \left(\frac{2\gamma_{\max}^2}{\pi \gamma_{\min}^2} \right)^{\frac{d}{4}} \left(\int_{\mathbb{R}^d} \|h\|_2^{2q\alpha} \exp\left(-\frac{2\|h\|_2^2}{\gamma_{\max}^2}\right) dh \right)^{\frac{1}{2}}.
\end{aligned}$$

Using the embedding constant $d^{\frac{q\alpha-1}{2q\alpha}}$ of $\mathcal{G}_{2q\alpha}^{\beta d}$ to $\mathcal{G}_{2q\alpha}^{\beta d}$, we obtain

$$\begin{aligned}
&\int_{\mathbb{R}^d} \|h\|_2^{2q\alpha} \exp\left(-\frac{2\|h\|_2^2}{\gamma_{\max}^2}\right) dh \leq d^{q\alpha-1} \sum_{\ell=1}^d \int_{\mathbb{R}^d} h_{\ell}^{2q\alpha} \prod_{\ell=1}^d \exp\left(-\frac{2h_{\ell}^2}{\gamma_{\max}^2}\right) d(h_1, \dots, h_d) \\
&= d^{q\alpha-1} \sum_{\ell=1}^d \left(\frac{\gamma_{\max}^2 \pi}{2} \right)^{\frac{d-1}{2}} \int_{\mathbb{R}} h_{\ell}^{2q\alpha} \exp\left(-\frac{2h_{\ell}^2}{\gamma_{\max}^2}\right) dh_{\ell}
\end{aligned}$$

$$= 2d^{q\alpha} \left(\frac{\gamma_{\max}^2 \pi}{2} \right)^{\frac{d-1}{2}} \int_0^\infty t^{2q\alpha} \exp\left(-\frac{2t^2}{\gamma_{\max}}\right) dt.$$

for $\gamma > 0$. With the substitution $t = (\frac{1}{2}\gamma_{\max}^2 u)^{\frac{1}{2}}$, the functional equation $\Gamma(t+1) = t\Gamma(t)$ of the Gamma function Γ , and $\Gamma(\frac{1}{2}) = \sqrt{\pi}$ we further have

$$\begin{aligned} \int_0^\infty t^{2q\alpha} \exp\left(-\frac{2t^2}{\gamma_{\max}}\right) dt &= \frac{1}{2} \frac{\gamma_{\max}}{\sqrt{2}} \left(\frac{\gamma_{\max}^2}{2}\right)^{q\alpha} \int_0^\infty u^{(q\alpha+\frac{1}{2})-1} \exp(-u) du \\ &= \frac{1}{2} \frac{\gamma_{\max}}{\sqrt{2}} \left(\frac{\gamma_{\max}^2}{2}\right)^{q\alpha} \Gamma\left(q\alpha + \frac{1}{2}\right). \end{aligned}$$

Altogether, we finally obtain

$$\begin{aligned} & \left\| \sum_{j \in J} \mathbb{1}_{A_j} \cdot (K_j * f) - f \right\|_{L_q(\nu)}^q \\ & \leq \|f\|_{B_{q,\infty}^{\alpha}(\nu)}^q \left(\frac{2\gamma_{\max}^2}{\pi\gamma_{\min}^4} \right)^{\frac{d}{4}} \left(\int_{\mathbb{R}^d} \|h\|_2^{2q\alpha} \exp\left(-\frac{2\|h\|_2^2}{\gamma_{\max}}\right) dh \right)^{\frac{1}{2}} \\ & \leq \|f\|_{B_{q,\infty}^{\alpha}(\nu)}^q \left(\frac{2\gamma_{\max}^2}{\pi\gamma_{\min}^4} \right)^{\frac{d}{4}} \left(\frac{d}{2} \right)^{q\alpha} \left(\frac{\pi^{d-1}}{2^d} \right)^{\frac{1}{2}} \frac{1}{\gamma_{\max}^{2q\alpha+d}} \Gamma\left(q\alpha + \frac{1}{2}\right) \\ & = \|f\|_{B_{q,\infty}^{\alpha}(\nu)}^q \left(\frac{d}{2} \right)^{\frac{q\alpha}{2}} \pi^{-\frac{1}{4}} \Gamma\left(q\alpha + \frac{1}{2}\right)^{\frac{1}{2}} \left(\frac{\gamma_{\max}}{\gamma_{\min}} \right)^{q\alpha}. \end{aligned}$$

■

Based on Theorems 11, 12, and 13, we can now show Theorem 4, where we denote by $L \circ f$ the function $(x, y) \mapsto L(x, y, f(x))$.

Proof [of Theorem 4] First of all, since H_1, \dots, H_m are RKHSs of Gaussian kernels, the joined RKHS H is separable and its kernel is measurable. Moreover, since Theorem 12 provides $\epsilon_i(\text{id} : H_{\gamma_j}(A_j) \rightarrow L_2(\mathbb{P}_{X|A_j})) \leq a_j t^{-\frac{1}{2p}}$ for $i \geq 1$ with $a_j = \tilde{c}_p \sqrt{\mathbb{P}_X(A_j)} r^{\frac{d+2p}{2p}} \gamma_j^{-\frac{1}{2p}}$, Theorem 11 yields

$$\mathbb{E}_{D_{X \times Y} \sim \mathbb{P}_X^{\otimes n} \times \mathbb{P}_Y^{\otimes n}} \epsilon_i(\text{id} : H \rightarrow L_2(D_X)) \leq c_p \sqrt{m} \left(\sum_{j=1}^m \lambda_j^{-p} a_j^{2p} \right)^{\frac{1}{2p}} t^{-\frac{1}{2}}, \quad i, n \geq 1.$$

Note that, for the least squares loss, which can be clipped at M with $Y = [-M, M]$, the supremum bound

$$L(x, y, t) \leq B, \quad \forall (x, y) \in X \times Y, t \in [-M, M] \quad (27)$$

holds for $B = 4M^2$ and the variance bound

$$\mathbb{E}_{\mathbb{P}} (L \circ f - L \circ f_{L,P}^*)^2 \leq V \cdot (\mathbb{E}_{\mathbb{P}} (L \circ f - L \circ f_{L,P}^*))^{\theta}, \quad \forall f : X \rightarrow [-M, M] \quad (28)$$

for $V = 16M^2$ and $\theta = 1$ (cf. Steinwart and Christmann, 2008, Example 7.3). Actually, (27) immediately yields the supremum bound for $L_{J,P}$, too. The same holds for the variance bound (28), which can be easily shown by the use of $\tilde{f}(x) := \mathbb{1}_{\bigcup_{j \in J_P} A_j}(x) f(x) + \mathbb{1}_{X \setminus (\bigcup_{j \in J_P} A_j)}(x) f_{L,P}^*(x)$ for all $f : X \rightarrow [-M, M]$. Using the constant B , we now have

$$\begin{aligned} & \left(\max \left\{ c_p \sqrt{m} \left(\sum_{j=1}^m \lambda_j^{-p} a_j^{2p} \right)^{\frac{1}{2p}}, B \right\} \right)^{2p} \\ & = \left(\max \left\{ c_p \tilde{c}_p \sqrt{m} r^{\frac{d+2p}{2p}} \left(\sum_{j=1}^m \left(\lambda_j^{-1} \gamma_j^{-\frac{d+2p}{p}} \mathbb{P}_X(A_j) \right)^p \right)^{\frac{1}{2p}}, B \right\} \right)^{2p} \\ & \leq \left(\max \left\{ c_p \tilde{c}_p m^{\frac{1}{2}} r^{\frac{d+2p}{2p}} \left(\sum_{j=1}^m \lambda_j^{-1} \gamma_j^{-\frac{d+2p}{p}} \mathbb{P}_X(A_j) \right)^{\frac{1}{2}}, B \right\} \right)^{2p} \\ & \leq \left(\max \left\{ c_p \tilde{c}_p 3^{\frac{d}{2p}} r \left(\sum_{j=1}^m \lambda_j^{-1} \gamma_j^{-\frac{d+2p}{p}} \mathbb{P}_X(A_j) \right)^{\frac{1}{2}}, B \right\} \right)^{2p} \\ & \leq C_p r^{2p} \left(\sum_{j=1}^m \lambda_j^{-1} \gamma_j^{-\frac{d+2p}{p}} \mathbb{P}_X(A_j) \right)^p + B^{2p} \\ & =: a^{2p}, \end{aligned}$$

where we used $\|\cdot\|_{\ell_p^m} \leq m^{\frac{1-2}{p}} \|\cdot\|_{\ell_1^m}$, $m r^d \leq 3^d$ by (5), and $C_p := \tilde{c}_p^{2p} \tilde{c}_p^{2p} 3^d$. Then, we can apply (Steinwart and Christmann, 2008, Theorem 7.23) using the regularization parameter $\lambda = 1$. That is, for $\lambda_1, \dots, \lambda_m > 0$, all fixed $\tau > 0$, and for an $f_0 \in H$ and a constant $B_0 \geq B$ such that $\|L_{J,P} \circ f_0\|_{\infty} \leq B_0$, we obtain

$$\begin{aligned} & \sum_{j=1}^m \lambda_j \|f_{D_j, \lambda_j}\|_{\tilde{H}_j}^2 + \mathcal{R}_{L_{j,P}}(\hat{f}_{D, \lambda}) - \mathcal{R}_{L_{j,P}}^* \\ & = \|f_{D, \lambda}\|_{\tilde{H}}^2 + \mathcal{R}_{L_{j,P}}(\hat{f}_{D, \lambda}) - \mathcal{R}_{L_{j,P}}^* \\ & \leq 9 \left(\|f_0\|_{\tilde{H}}^2 + \mathcal{R}_{L_{j,P}}(f_0) - \mathcal{R}_{L_{j,P}}^*(f_0) + C \left(a^{2p} n^{-1} \right)^{\frac{1}{2-p-\theta+2p}} + 3 \left(\frac{72V\tau}{n} \right)^{\frac{1}{2-p}} + \frac{15B_0\tau}{n} \right) \\ & \leq 9 \left(\sum_{j=1}^m \lambda_j \| \mathbb{1}_{A_j} f_0 \|_{\tilde{H}_j}^2 + \mathcal{R}_{L_{j,P}}(f_0) - \mathcal{R}_{L_{j,P}}^*(f_0) + C \left(a^{2p} n^{-1} \right)^{\frac{1}{2-p-\theta+2p}} + 3 \left(\frac{72V\tau}{n} \right)^{\frac{1}{2-p}} + \frac{15B_0\tau}{n} \right) \quad (29) \end{aligned}$$

with probability \mathbb{P}^n not less than $1 - 3e^{-\tau}$, where $C > 0$ is the constant of (Steinwart and Christmann, 2008, Theorem 7.23) only depending on p, M, V, θ , and B . To continue estimate (29), we have to choose a function $f_0 \in H$. To this end, we define functions $K_j : \mathbb{R}^d \rightarrow \mathbb{R}$, $j \in \{1, \dots, m\}$, by (26), where $s := [\alpha] + 1$ and $\gamma_j > 0$. Then, we define f_0 by convolving each K_j with the Bayes decision function $f_{L,P}^*$, that is

$$f_0(x) := \sum_{j \in J_P} \mathbb{1}_{A_j}(x) \cdot (K_j * f_{L,P}^*)(x), \quad x \in \mathbb{R}^d.$$

Now, to show that f_0 is indeed a suitable function to bound the approximation error, we first need to ensure that f_0 is contained in H . In addition, we need to derive bounds for both, the regularization term and the excess risk of f_0 . To this end, we apply (Eberts and Steinwart, 2013, Theorem 2.3) and obtain, for every $j \in J_T$,

$$(K_j * f_{L,P}^*)|_{A_j} \in H_{\gamma_j}(A_j)$$

with

$$\begin{aligned} \|\mathbb{1}_{A_j} f_0\|_{H_{\gamma_j}(A_j)} &= \|\mathbb{1}_{A_j}(K_j * f_{L,P}^*)\|_{H_{\gamma_j}(A_j)} \\ &= \|(K_j * f_{L,P}^*)|_{A_j}\|_{H_{\gamma_j}(A_j)} \\ &\leq (\gamma_j \sqrt{\pi})^{-\frac{d}{2}} (2^s - 1) \|f_{L,P}^*\|_{L_2(\mathbb{R}^d)}. \end{aligned}$$

This implies

$$f_0 = \sum_{j \in J_T} \underbrace{\mathbb{1}_{A_j}(K_j * f_{L,P}^*)}_{\in H_{\gamma_j}(A_j)} \in H_{J_T}.$$

Besides, note that $0 \in \hat{H}_{\gamma_j}(A_j)$ for every $j \in \{1, \dots, m\}$ such that f_0 can be written as $f_0 = \sum_{j=1}^m f_j$, where

$$f_j := \begin{cases} \mathbb{1}_{A_j}(K_j * f_{L,P}^*), & j \in J_T, \\ 0, & j \notin J_T. \end{cases}$$

Obviously, the latter implies $f_0 \in H$. Furthermore, for $A_T := \bigcup_{j \in J_T} A_j$, (25) and Theorem 13 yield

$$\begin{aligned} \mathcal{R}_{L_{J_T}, P}(f_0) - \mathcal{R}_{L_{J_T}, P}^* &= \|f_0 - f_{L,P}^*\|_{L_2(P_{X|A_T})}^2 \\ &= \left\| \sum_{j \in J_T} \mathbb{1}_{A_j}(K_j * f_{L,P}^*) - f_{L,P}^* \right\|_{L_2(P_{X|A_T})}^2 \\ &\leq C_{\alpha,2} \left(\frac{\max_{j \in J_T} \gamma_j^{-d}}{\min_{j \in J_T} \gamma_j} \right)^d \max_{j \in J_T} \gamma_j^{2\alpha}, \end{aligned}$$

where $C_{\alpha,2}$ is a constant only depending on α , d , and $\|f_{L,P}^*\|_{B_{2,\infty}^2(P_{X|A_T})}$. Next, we derive a bound for $\|L \circ f_0\|_\infty$ using (Eberts and Steinwart, 2013, Theorem 2.3) which provides, for every $x \in X$, the supremum bound

$$|f_0(x)| = \left| \sum_{j \in J_T} \mathbb{1}_{A_j}(x) \cdot (K_j * f_{L,P}^*)(x) \right| \leq \sum_{j \in J_T} \mathbb{1}_{A_j}(x) |K_j * f_{L,P}^*(x)| \leq (2^s - 1) \|f_{L,P}^*\|_{L_\infty(\mathbb{R}^d)}.$$

The latter implies

$$\|L_{J_T} \circ f_0\|_\infty = \sup_{(x,\hat{y}) \in X \times Y} |L(\hat{y}, f_0(x))|$$

$$\begin{aligned} &\leq \sup_{(x,\hat{y}) \in X \times Y} (M^2 + 2M|f_0(x)| + |f_0(x)|^2) \\ &\leq 4^s \max \left\{ M^2, \|f_{L,P}^*\|_{L_\infty(\mathbb{R}^d)}^2 \right\}, \end{aligned}$$

i.e., $B_0 := 4^s \max\{M^2, \|f_{L,P}^*\|_{L_\infty(\mathbb{R}^d)}^2\}$. Applying (29) then yields

$$\begin{aligned} &\mathcal{R}_{L_{J_T}, P}(\widehat{f}_{D,\lambda,\gamma}) - \mathcal{R}_{L_{J_T}, P}^* \\ &\leq \sum_{j=1}^m \lambda_j \|f_{D_j, \lambda_j, \gamma_j}\|_{H_{\gamma_j}(A_j)}^2 + \mathcal{R}_{L_{J_T}, P}(\widehat{f}_{D,\lambda,\gamma}) - \mathcal{R}_{L_{J_T}, P}^* \\ &\leq 9 \left(\sum_{j=1}^m \lambda_j \| \mathbb{1}_{A_j} f_0 \|_{H_{\gamma_j}(A_j)}^2 + \mathcal{R}_{L_{J_T}, P}(f_0) - \mathcal{R}_{L_{J_T}, P}^* \right) \\ &\quad + C (a^{2p} n^{-1})^{\frac{1}{2-p+\theta p}} + 3 \left(\frac{72V\tau}{n} \right)^{\frac{1}{2-p}} + \frac{15B_0\tau}{n} \\ &\leq 9 \left(\sum_{j \in J_T} \lambda_j (\gamma_j \sqrt{\pi})^{-d} (2^s - 1)^2 \|f_{L,P}^*\|_{L_2(\mathbb{R}^d)}^2 + C_{\alpha,2} \left(\frac{\max_{j \in J_T} \gamma_j^{-d}}{\min_{j \in J_T} \gamma_j} \right)^d \max_{j \in J_T} \gamma_j^{2\alpha} \right) \\ &\quad + CC_p^{2p} \left(\sum_{j=1}^m \lambda_j^{-1} \gamma_j^{-\frac{d+2p}{p}} P_X(A_j) \right)^p n^{-1} + CB^{2p} n^{-1} + \frac{3456M^2\tau}{n} \\ &\quad + 15 \cdot 4^s \max\{M^2, \|f_{L,P}^*\|_{L_\infty(\mathbb{R}^d)}^2\} \frac{\tau}{n} \\ &\leq 9(2^s - 1)^2 \pi^{-\frac{d}{2}} \|f_{L,P}^*\|_{L_2(\mathbb{R}^d)}^2 \sum_{j \in J_T} \lambda_j \gamma_j^{-d} + 9C_{\alpha,2} \left(\frac{\max_{j \in J_T} \gamma_j^{-d}}{\min_{j \in J_T} \gamma_j} \right)^d \max_{j \in J_T} \gamma_j^{2\alpha} \\ &\quad + CC_p^{2p} \left(\sum_{j=1}^m \lambda_j^{-1} \gamma_j^{-\frac{d+2p}{p}} P_X(A_j) \right)^p n^{-1} + 16^p C M^{4p} n^{-1} \\ &\quad + (3456M^2 + 15 \cdot 4^s \max\{M^2, \|f_{L,P}^*\|_{L_\infty(\mathbb{R}^d)}^2\}) \frac{\tau}{n} \end{aligned}$$

with probability P^n not less than $1 - 3e^{-\tau}$. Finally, for $\hat{\tau} \geq 1$, a variable transformation implies

$$\begin{aligned} &\sum_{j=1}^m \lambda_j \|f_{D_j, \lambda_j, \gamma_j}\|_{H_{\gamma_j}(A_j)}^2 + \mathcal{R}_{L_{J_T}, P}(\widehat{f}_{D,\lambda,\gamma}) - \mathcal{R}_{L_{J_T}, P}^* \\ &\leq C_{M,\alpha,p} \left(\sum_{j \in J_T} \lambda_j \gamma_j^{-d} + \left(\frac{\max_{j \in J_T} \gamma_j^{-d}}{\min_{j \in J_T} \gamma_j} \right)^d \max_{j \in J_T} \gamma_j^{2\alpha} + n^{2p} \left(\sum_{j=1}^m \lambda_j^{-1} \gamma_j^{-\frac{d+2p}{p}} P_X(A_j) \right)^p n^{-1} + \hat{\tau} n^{-1} \right) \end{aligned}$$

with probability P^n not less than $1 - e^{-\hat{\tau}}$, where the constant $C_{M,\alpha,p}$ is defined by

$$C_{M,\alpha,p} := \max \left\{ 9(2^s - 1)^2 \pi^{-\frac{d}{2}} \|f_{L,P}^*\|_{L_2(\mathbb{R}^d)}^2, 9 \|f_{L,P}^*\|_{B_{2,\infty}^2(P_{X|A_T})}^2 \left(\frac{d}{2} \right)^\alpha \pi^{-\frac{d}{4}} \Gamma \left(2\alpha + \frac{1}{2} \right)^{\frac{1}{2}} \right\},$$

$$3^d C_{\mathcal{P}}^{2p} c_{\mathcal{P}}^{2p}, 16^p C_M^{4p} + (3456M^2 + 15 \cdot 4^p \max\{M^2, \|f_{L,P}^*\|_{L_\infty(\mathbb{R}^d)}^2\}) (1 + \ln(3)) \Big\}. \quad \blacksquare$$

Next, using the just proven oracle inequality presented in Theorem 4, we show the learning rates of Theorem 5 in only a few steps.

Proof [of Theorem 5] First of all, we define sequences $\tilde{\lambda}_n := c_2 n^{-1}$ and $\tilde{\gamma}_n := c_3 n^{-\frac{1}{2\alpha+d}}$ to simplify the presentation. Then, Theorem 4, $\sum_{j=1}^{m_n} \mathbb{P}X(A_j) = 1$, and $|J_T| \leq m_n \leq 3^d r_n^{-d}$ together with $\lambda_{n,j} = r_n^d \tilde{\lambda}_n$ and $\gamma_{n,j} = \tilde{\gamma}_n$ for all $j \in \{1, \dots, m_n\}$ yield

$$\begin{aligned} & \mathcal{R}_{L_{J_T}, P}(\tilde{f}_D, \lambda_n, \tilde{\gamma}_n) - \mathcal{R}_{L_{J_T}, P}^* \\ & \leq C_{M, \alpha, p} \left(\sum_{j \in J_T} \lambda_{n,j} \tilde{\gamma}_n^{-d} + \left(\frac{\max_{j \in J_T} \gamma_{n,j}}{\min_{j \in J_T} \gamma_{n,j}} \right)^d \max_{j \in J_T} \gamma_{n,j}^{2\alpha} + r_n^{2p} \left(\sum_{j=1}^{m_n} \lambda_{n,j}^{-1} \gamma_{n,j}^{-\frac{d+2p}{p}} \mathbb{P}X(A_j) \right)^p n^{-1} + \frac{\tau}{n} \right) \\ & = C_{M, \alpha, p} \left(|J_T| r_n^d \tilde{\lambda}_n \tilde{\gamma}_n^{-d} + \tilde{\gamma}_n^{2\alpha} + r_n^{(2-d)p} \tilde{\lambda}_n \tilde{\gamma}_n^{-(d+2p)} \left(\sum_{j=1}^{m_n} \mathbb{P}X(A_j) \right)^p n^{-1} + \tau n^{-1} \right) \\ & \leq 3^d C_{M, \alpha, p} \left(\tilde{\lambda}_n \tilde{\gamma}_n^{-d} + \tilde{\gamma}_n^{2\alpha} + \tilde{\lambda}_n^{-p} \tilde{\gamma}_n^{-(d+2p)} r_n^{(2-d)p} n^{-1} + \tau n^{-1} \right). \end{aligned}$$

Using the choices $\tilde{\lambda}_n = c_2 n^{-1}$, $\tilde{\gamma}_n = c_3 n^{-\frac{1}{2\alpha+d}}$, as well as $r_n = c_1 n^{-\frac{1}{\beta d}}$ finally implies

$$\begin{aligned} & \mathcal{R}_{L_{J_T}, P}(\tilde{f}_D, \lambda_n, \tilde{\gamma}_n) - \mathcal{R}_{L_{J_T}, P}^* \\ & \leq 3^d C_{M, \alpha, p} \left(\tilde{\lambda}_n \tilde{\gamma}_n^{-d} + \tilde{\gamma}_n^{2\alpha} + \tilde{\lambda}_n^{-p} \tilde{\gamma}_n^{-(d+2p)} r_n^{(2-d)p} n^{-1} + \tau n^{-1} \right) \\ & \leq \hat{C}_{M, \alpha, p} \left(n^{-1} n^{\frac{d}{2\alpha+d}} + n^{-\frac{2\alpha}{2\alpha+d}} + n^p n^{\frac{d+2p}{2\alpha+d}} n^{-\frac{(2-d)p}{\beta d}} n^{-1} + \tau n^{-1} \right) \\ & = \hat{C}_{M, \alpha, p} \left(n^{-\frac{2\alpha}{2\alpha+d}} + n^{-\frac{2\alpha}{2\alpha+d}} + n^{-\frac{2\alpha}{2\alpha+d}} + \left(1 + \frac{2}{2\alpha+d} + \frac{1}{\beta} - \frac{2}{\beta d}\right) p + \tau n^{-1} \right) \\ & \leq C \left(n^{-\frac{2\alpha}{2\alpha+d} + \xi} + \tau n^{-1} \right) \end{aligned}$$

with probability \mathbb{P}^n not less than $1 - e^{-\tau}$, where $C > 0$ is a constant and $\xi \geq \left(1 + \frac{2}{2\alpha+d} + \frac{1}{\beta} - \frac{2}{\beta d}\right) p > 0$. \blacksquare

Proof [of Corollary 6] For simplicity of notation, we write λ , λ_j , γ , and γ_j instead of λ_n , $\lambda_{n,j}$, γ_n , and $\gamma_{n,j}$. Since $\cup_{j \in J_T} A_j \subset T^{+\delta}$ for all $n \geq n_\delta$, the assumption $f_{L,P}^* \in B_{2,\infty}^\alpha(\mathbb{P}X|_{T^{+\delta}})$ implies

$$f_{L,P}^* \in B_{2,\infty}^\alpha(\mathbb{P}X|_{\cup_{j \in J_T} A_j}).$$

With this, Theorems 4 and 5 immediately yield

$$\mathcal{R}_{L_{J_T}, P}(\tilde{f}_D, \lambda, \gamma) - \mathcal{R}_{L_{J_T}, P}^*$$

$$\begin{aligned} & \leq \sum_{j=1}^m \lambda_j \|f_{D_j, \lambda_j, \gamma_j}\|_{H_{r_j}(A_j)}^2 + \mathcal{R}_{L_{J_T}, P}(\tilde{f}_D, \lambda, \gamma) - \mathcal{R}_{L_{J_T}, P}^* \\ & \leq \sum_{j=1}^m \lambda_j \|f_{D_j, \lambda_j, \gamma_j}\|_{H_{r_j}(A_j)}^2 + \mathcal{R}_{L_{J_T}, P}(\tilde{f}_D, \lambda, \gamma) - \mathcal{R}_{L_{J_T}, P}^* \\ & \leq C_{M, \alpha, p} \left(\sum_{j \in J_T} \lambda_j \gamma_j^{-d} + \left(\frac{\max_{j \in J_T} \gamma_j}{\min_{j \in J_T} \gamma_j} \right)^d \max_{j \in J_T} \gamma_j^{2\alpha} + r^{2p} \left(\sum_{j=1}^m \lambda_j^{-1} \gamma_j^{-\frac{d+2p}{p}} \mathbb{P}X(A_j) \right)^p n^{-1} + \frac{\tau}{n} \right) \\ & \leq C \left(n^{-\frac{2\alpha}{2\alpha+d} + \xi} + \tau n^{-1} \right) \end{aligned}$$

with probability \mathbb{P}^n not less than $1 - e^{-\tau}$, where $\xi \geq \left(1 + \frac{2}{2\alpha+d} + \frac{1}{\beta} - \frac{2}{\beta d}\right) p > 0$. Moreover, the constants $C_{M, \alpha, p} > 0$ and $C > 0$ coincide with those of Theorems 4 and 5. \blacksquare

It remains to prove Theorem 7. However, we previously have to consider the following technical lemma.

Lemma 14 Let $d \geq 1$ and $r_n := cn^{-\frac{1}{\beta d}}$ with $\beta > 1$ and a constant $c > 0$. We fix finite subsets $\Lambda_n \subset (0, r_n^d]$ and $\Gamma_n \subset (0, r_n]$ such that Λ_n is an $(r_n^d \varepsilon_n)$ -net of $(0, r_n^d]$ and Γ_n is an δ_n -net of $(0, r_n]$ with $0 < \varepsilon_n \leq n^{-1}$, $\delta_n > 0$, $r_n^d \in \Lambda_n$, and $r_n \in \Gamma_n$. Moreover, let $J \subset \{1, \dots, m_n\}$ be an arbitrary non-empty index set and $|J| \leq m_n \leq 3^d r_n^{-d}$. Then, for all $0 < \alpha < \frac{\beta-1}{2}d$, $n \geq 1$, and all $p \in (0, 1)$ with $p \leq \frac{\beta d - 2\alpha - d}{2\alpha + d + 2}$, we have

$$\begin{aligned} & \inf_{(\lambda_j, \gamma_j)_{j=1}^{m_n} \in (\Lambda_n \times \Gamma_n)^{m_n}} \left(\sum_{j \in J} \lambda_j \gamma_j^{-d} + \left(\frac{\max_{j \in J} \gamma_j}{\min_{j \in J} \gamma_j} \right)^d \max_{j \in J} \gamma_j^{2\alpha} + r_n^{2p} \left(\sum_{j=1}^{m_n} \lambda_j^{-1} \gamma_j^{-\frac{d+2p}{p}} \mathbb{P}X(A_j) \right)^p n^{-1} \right) \\ & \leq C \left(n^{-\frac{2\alpha}{2\alpha+d} + \xi} + \delta_n^{2\alpha} \right), \end{aligned}$$

where $\xi := \left(\frac{2\alpha(2\alpha+d+2)}{(2\alpha+d)(2\alpha+d)(1+p)+2p} + \max\left\{\frac{d-2}{\beta d}, 0\right\} p \right) p$ and $C > 0$ is a constant independent of n , Λ_n , ε_n , Γ_n , and δ_n .

Proof Without loss of generality, we may assume that Λ_n and Γ_n are of the form $\Lambda_n = \{\lambda^{(1)}, \dots, \lambda^{(v)}\}$ and $\Gamma_n = \{\gamma^{(1)}, \dots, \gamma^{(v)}\}$ with $\lambda^{(v)} = r_n^d$ and $\gamma^{(v)} = r_n$ as well as $\lambda^{(\ell-1)} < \lambda^{(\ell)}$ and $\gamma^{(\ell-1)} < \gamma^{(\ell)}$ for all $i = 2, \dots, v$. With $\lambda^{(0)} := 0$ and $\gamma^{(0)} := 0$ it is easy to see that

$$\lambda^{(\ell)} - \lambda^{(\ell-1)} \leq 2r_n^d \varepsilon_n \quad \text{and} \quad \gamma^{(\ell)} - \gamma^{(\ell-1)} \leq 2\delta_n \quad (30)$$

hold for all $i = 1, \dots, v$ and $\ell = 1, \dots, v$. Furthermore, define $\lambda^* := n^{-\frac{2\alpha+d}{(2\alpha+d)(1+p)+2p}}$ and $\gamma^* := cn^{-\frac{2\alpha+d}{(2\alpha+d)(1+p)+2p}}$. Then, there exist indices $i \in \{1, \dots, v\}$ and $\ell \in \{1, \dots, v\}$ with $\lambda^{(\ell-1)} \leq r_n^d \lambda^* \leq \lambda^{(\ell)}$ and $\gamma^{(\ell-1)} \leq \gamma^* \leq \gamma^{(\ell)}$. Together with (30), this yields

$$r_n^d \lambda^* \leq \lambda^{(\ell)} \leq r_n^d \lambda^* + 2r_n^d \varepsilon_n \quad \text{and} \quad \gamma^* \leq \gamma^{(\ell)} \leq \gamma^* + 2\delta_n. \quad (31)$$

Moreover, the definition of λ^* implies $\varepsilon_n \leq \lambda^*$ and the one of γ^* implies $\gamma^* \leq \tau_n$ for $\alpha < \frac{d-1}{2}d$ and $p \in (0, p^*]$, where $p^* := \frac{2d-2\alpha-d}{2\alpha+d+2}$. Additionally, it is easy to check that

$$\lambda^* (\gamma^*)^{-d} + (\gamma^*)^{2\alpha} + (\lambda^*)^{-p} (\gamma^*)^{-(d+2p)} r_n^{(2-d)p} n^{-1} \leq \tilde{c} n^{-\frac{2\alpha}{(2\alpha+d)(1+2p)} + \max\{\frac{d-2}{\beta d}, 0\}p}, \quad (32)$$

where \tilde{c} is a positive constant. Using (31), the bound $|J| \leq m_n \leq 3^d r_n^{-d}$, and (32), we obtain

$$\begin{aligned} & \inf_{(\lambda_j, \gamma_j)_{j=1}^{m_n} \in (\Lambda_n \times \Gamma_n)^{m_n}} \left(\sum_{j \in J} \lambda_j \gamma_j^{-d} + \left(\frac{\max_{j \in J} \gamma_j^2}{\min_{j \in J} \gamma_j} \right)^d \max_{j \in J} \gamma_j^{2\alpha} + r_n^{2p} \left(\sum_{j=1}^{m_n} \lambda_j^{-1} \gamma_j^{-p} \mathbb{P}_X(A_j) \right) n^{-1} \right)^p \\ & \leq \sum_{j \in J} \lambda^{(j)} (\gamma^{(j)})^{-d} + (\gamma^{(j)})^{2\alpha} + \left(\sum_{j=1}^{m_n} \lambda^{(j)} \right)^{-1} (\gamma^{(j)})^{-\frac{d+2p}{p}} \mathbb{P}_X(A_j) \Big)^p \\ & \leq |J| \lambda^{(j)} (\gamma^{(j)})^{-d} + (\gamma^{(j)})^{2\alpha} + (\lambda^{(j)})^{-p} (\gamma^{(j)})^{-(d+2p)} r_n^{2p} n^{-1} \\ & \leq |J| \left(r_n^d \lambda^* + 2r_n^d \varepsilon_n \right) (\gamma^*)^{-d} + (\gamma^* + 2\delta_n)^{2\alpha} + \left(r_n^d \lambda^* \right)^{-p} (\gamma^*)^{-(d+2p)} r_n^{2p} n^{-1} \\ & \leq 3^d \cdot 3 \lambda^* (\gamma^*)^{-d} + (\gamma^* + 2\delta_n)^{2\alpha} + (\lambda^*)^{-p} (\gamma^*)^{-(d+2p)} r_n^{(2-d)p} n^{-1} \\ & \leq \tilde{c} \left(\lambda^* (\gamma^*)^{-d} + (\gamma^*)^{2\alpha} + (\lambda^*)^{-p} (\gamma^*)^{-(d+2p)} r_n^{(2-d)p} n^{-1} \right) + \tilde{c} \delta_n^{2\alpha} \\ & \leq \tilde{c} \tilde{c} n^{-\frac{2\alpha}{(2\alpha+d)(1+2p)} + \max\{\frac{d-2}{\beta d}, 0\}p} + \tilde{c} \delta_n^{2\alpha} \\ & \leq C \left(n^{-\frac{2\alpha}{2\alpha+d} + \xi} + \delta_n^{2\alpha} \right) \end{aligned}$$

with $\xi := \left(\frac{2\alpha(2\alpha+d+2)}{(2\alpha+d)(2\alpha+d)(1+2p)} + \max\{\frac{d-2}{\beta d}, 0\} \right) p$ and constants $\tilde{c} > 0$ and $C > 0$ independent of n , Λ_n , ε_n , Γ_n , and δ_n . ■

In the end, we show Theorem 7 using Theorem 4 as well as Lemma 14.

Proof [of Theorem 7] Let l be defined by $l := \lfloor \frac{n}{2} \rfloor + 1$, i.e., $l \geq \frac{n}{2}$. With this, Theorem 4 yields with probability \mathbb{P}^l not less than $1 - |\Lambda_n \times \Gamma_n|^{m_n} e^{-\tau}$ that

$$\begin{aligned} & \mathcal{R}_{L_{J_n, \mathbb{P}}}(\tilde{f}_{D_1, \lambda, \gamma}) - \mathcal{R}_{L_{J_n, \mathbb{P}}}^* \\ & \leq \frac{c_1}{2} \left(\sum_{j \in J_n} \lambda_j \gamma_j^{-d} + \left(\frac{\max_{j \in J_n} \gamma_j^2}{\min_{j \in J_n} \gamma_j} \right)^d \max_{j \in J_n} \gamma_j^{2\alpha} + r_n^{2p} \left(\sum_{j=1}^{m_n} \lambda_j^{-1} \gamma_j^{-p} \mathbb{P}_X(A_j) \right) \right)^p l^{-1} + \tau l^{-1} \\ & \leq c_1 \left(\sum_{j \in J_n} \lambda_j \gamma_j^{-d} + \left(\frac{\max_{j \in J_n} \gamma_j^2}{\min_{j \in J_n} \gamma_j} \right)^d \max_{j \in J_n} \gamma_j^{2\alpha} + r_n^{2p} \left(\sum_{j=1}^{m_n} \lambda_j^{-1} \gamma_j^{-p} \mathbb{P}_X(A_j) \right) \right)^p n^{-1} + \tau n^{-1} \end{aligned} \quad (33)$$

for all $(\lambda_j, \gamma_j) \in \Lambda_n \times \Gamma_n$, $j \in \{1, \dots, m_n\}$, simultaneously, where $c_1 > 0$ is a constant independent of n , τ , λ , and γ . Furthermore, the oracle inequality of Steinwart and Christmann, 2008, Theorem 7.2) for empirical risk minimization, $n - l \geq \frac{n}{2} - 1 \geq \frac{n}{4}$,

and $\tau_n := \tau + \ln(1 + |\Lambda_n \times \Gamma_n|^{m_n})$ yield

$$\begin{aligned} & \mathcal{R}_{L_{J_n, \mathbb{P}}}(\tilde{f}_{D_1, \lambda D_2, \gamma D_2}) - \mathcal{R}_{L_{J_n, \mathbb{P}}}^* \\ & < 6 \left(\inf_{(\lambda_j, \gamma_j)_{j=1}^{m_n} \in (\Lambda_n \times \Gamma_n)^{m_n}} \mathcal{R}_{L_{J_n, \mathbb{P}}}(\tilde{f}_{D_1, \lambda, \gamma}) - \mathcal{R}_{L_{J_n, \mathbb{P}}}^* \right) + 512M^2 \frac{\tau_n}{n-1} \\ & < 6 \left(\inf_{(\lambda_j, \gamma_j)_{j=1}^{m_n} \in (\Lambda_n \times \Gamma_n)^{m_n}} \mathcal{R}_{L_{J_n, \mathbb{P}}}(\tilde{f}_{D_1, \lambda, \gamma}) - \mathcal{R}_{L_{J_n, \mathbb{P}}}^* \right) + 2048M^2 \frac{\tau_n}{n} \end{aligned} \quad (34)$$

with probability \mathbb{P}^{n-l} not less than $1 - e^{-\tau}$. With (33), (34), and Lemma 14 we can conclude

$$\begin{aligned} & \mathcal{R}_{L_{J_n, \mathbb{P}}}(\tilde{f}_{D_1, \lambda D_2, \gamma D_2}) - \mathcal{R}_{L_{J_n, \mathbb{P}}}^* \\ & < 6 \left(\inf_{(\lambda_j, \gamma_j)_{j=1}^{m_n} \in (\Lambda_n \times \Gamma_n)^{m_n}} \mathcal{R}_{L_{J_n, \mathbb{P}}}(\tilde{f}_{D_1, \lambda, \gamma}) - \mathcal{R}_{L_{J_n, \mathbb{P}}}^* \right) + 2048M^2 \frac{\tau_n}{n} \\ & \leq 6c_1 \left(\inf_{(\lambda_j, \gamma_j)_{j=1}^{m_n} \in (\Lambda_n \times \Gamma_n)^{m_n}} \left(\sum_{j \in J_n} \lambda_j \gamma_j^{-d} + \left(\frac{\max_{j \in J_n} \gamma_j^2}{\min_{j \in J_n} \gamma_j} \right)^d \max_{j \in J_n} \gamma_j^{2\alpha} \right. \right. \\ & \quad \left. \left. + r_n^{2p} \left(\sum_{j=1}^{m_n} \lambda_j^{-1} \gamma_j^{-p} \mathbb{P}_X(A_j) \right) \right) n^{-1} + \tau n^{-1} \right) + 2048M^2 \frac{\tau_n}{n} \\ & \leq 6c_1 \left(C \left(n^{-\frac{2\alpha}{2\alpha+d} + \xi} + \delta_n^{2\alpha} \right) + \tau n^{-1} \right) + 2048M^2 \frac{\tau_n}{n} \\ & \leq 12c_1 C n^{-\frac{2\alpha}{2\alpha+d} + \xi} + (6c_1 \tau + 2048M^2 \tau_n) n^{-1} \end{aligned}$$

with probability \mathbb{P}^n not less than $1 - (1 + |\Lambda_n \times \Gamma_n|^{m_n}) e^{-\tau}$. Finally, a variable transformation yields

$$\begin{aligned} & \mathcal{R}_{L_{J_n, \mathbb{P}}}(\tilde{f}_{D_1, \lambda D_2, \gamma D_2}) - \mathcal{R}_{L_{J_n, \mathbb{P}}}^* \\ & < 12c_1 C n^{-\frac{2\alpha}{2\alpha+d} + \xi} + (6c_1 (\tau + \ln(1 + |\Lambda_n \times \Gamma_n|^{m_n}))) n^{-1} \\ & \quad + 2048M^2 (\tau + 2 \ln(1 + |\Lambda_n \times \Gamma_n|^{m_n})) n^{-1} \\ & \leq 12c_1 C n^{-\frac{2\alpha}{2\alpha+d} + \xi} + (6c_1 + 2048M^2) (\tau + 2m_n \ln(1 + |\Lambda_n \times \Gamma_n|)) n^{-1} \\ & \leq 12c_1 C n^{-\frac{2\alpha}{2\alpha+d} + \xi} + (6c_1 + 2048M^2) (\tau + 2 \cdot 3^d r_n^{-d} \ln(1 + |\Lambda_n \times \Gamma_n|)) n^{-1} \\ & = 12c_1 C n^{-\frac{2\alpha}{2\alpha+d} + \xi} + (6c_1 + 2048M^2) (\tau n^{-1} + 2 \cdot 3^d c^{-d} \ln(1 + |\Lambda_n \times \Gamma_n|) n^{-\frac{\beta-1}{\beta}}) \\ & < \left(12c_1 C + 2 \cdot 3^d c^{-d} (6c_1 + 2048M^2) \ln(1 + |\Lambda_n \times \Gamma_n|) \right) n^{-\frac{2\alpha}{2\alpha+d} + \xi} + (6c_1 + 2048M^2) \tau n^{-1} \end{aligned}$$

with probability \mathbb{P}^n not less than $1 - e^{-\tau}$, where we used

$$\alpha < \frac{\beta-1}{2} d \iff n^{-\frac{\beta-1}{\beta}} < n^{-\frac{2\alpha}{2\alpha+d}}$$

in the last step. ■

Acknowledgements

We would like to thank the Institute for Applied Analysis and Numerical Simulation of the University of Stuttgart for placing their professional compute servers at our disposal. We further like to thank the anonymous reviewers for many valuable comments, which improved the final version of this paper. Finally, we like to thank the action editor, S. van de Geer, for her patience during the revision process.

References

- R. A. Adams and J. J. F. Fournier. *Sobolev Spaces*. Academic Press, New York, 2nd edition, 2003.
- N. Aronszajn. Theory of reproducing kernels. *Trans. Amer. Math. Soc.*, 68:337–404, 1950.
- K.P. Bennett and J.A. Blue. A support vector machine approach to decision trees. In *The 1998 IEEE International Joint Conference on Neural Networks*, volume 3, pages 2396–2401 vol.3, 1998.
- H. Berens and R. DeVore. Quantitative Korovkin theorems for positive linear operators on L_p -spaces. *Trans. Amer. Math. Soc.*, 245:349–361, 1978.
- A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Kluwer, Boston, 2004.
- E. Blanzieri and A. Bryl. Instance-based spam filtering using SVM nearest neighbor classifier. In *Proceedings of FLAIRS 2007*, pages 441–442, 2007a.
- E. Blanzieri and A. Bryl. Evaluation of the highest probability SVM nearest neighbor classifier with variable relative error cost. In *Proceedings of 4th Conference on Email and Anti-Spam, CEAS'2007*, 2007b.
- E. Blanzieri and F. Melgani. Nearest neighbor classification of remote sensing images with the maximal margin principle. *IEEE Transactions on Geoscience and Remote Sensing*, 46:1804–1811, 2008.
- L. Bottou and V. Vapnik. Local learning algorithms. *Neural Computation*, 4:888–900, 1992.
- A. Caponnetto and E. De Vito. Optimal rates for regularized least squares algorithm. *Found. Comput. Math.*, 7:331–368, 2007.
- F. Chang, C.-Y. Guo, X.-R. Lin, and C.-J. Lu. Tree decomposition for large-scale SVM problems. *J. Mach. Learn. Res.*, 11:2935–2972, 2010.
- H. Cheng, P.-N. Tan, and R. Jin. Localized support vector machine and its efficient algorithm. In *SIAM International Conference on Data Mining*, 2007.
- H. Cheng, P.-N. Tan, and R. Jin. Efficient algorithm for localized support vector machine. *IEEE Transactions on Knowledge and Data Engineering*, 22:537–549, 2010.
- R. Collobert, S. Bengio, and Y. Bengio. A parallel mixture of SVMs for very large scale problems. In *Advances in Neural Information Processing Systems*, pages 633–640, 2001.
- F. Cucker and S. Smale. On the mathematical foundations of learning. *Bull. Amer. Math. Soc.*, 39:1–49, 2002.
- S. Dasgupta. Lecture 1: Clustering in metric spaces. CSE 291: Topics in unsupervised learning, 2008. URL <http://cseweb.ucsd.edu/~dasgupta/291-unsup/1ec1.pdf>.
- E. De Vito, A. Caponnetto, and L. Rosasco. Model selection for regularized least-squares algorithm in learning theory. *Found. Comput. Math.*, 5:59–85, 2005.
- R.A. DeVore and G.G. Lorentz. *Constructive Approximation*. Springer-Verlag, Berlin, 1993.
- R.A. DeVore and V.A. Popov. Interpolation of Besov spaces. *Trans. Amer. Math. Soc.*, 305:397–414, 1988.
- M. Eberts. *Adaptive Rates for Support Vector Machines*. Shaker, Aachen, 2015.
- M. Eberts and I. Steinwart. Optimal learning rates for least squares SVMs using Gaussian kernels. In J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F.C.N. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1539–1547. 2011.
- M. Eberts and I. Steinwart. Optimal regression rates for SVMs using Gaussian kernels. *Electron. J. Statist.*, 7:1–42, 2013.
- M. Eberts and I. Steinwart. Optimal learning rates for localized SVMs. 2014. URL <http://arxiv.org/pdf/1507.06615.pdf>.
- D. E. Edmunds and H. Triebel. *Function Spaces, Entropy Numbers, Differential Operators*. Cambridge University Press, Cambridge, 1996.
- T.F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.
- H.P. Graf, E. Cosatto, L. Bottou, I. Durdanovic, and V. Vapnik. Parallel support vector machines: The cascade SVM. In *Advances in Neural Information Processing Systems*, pages 521–528, 2005.
- L. Györfi, M. Kohler, A. Krzyżak, and H. Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer, New York, 2002.
- R. Hable. Universal consistency of localized versions of regularized kernel methods. *J. Mach. Learn. Res.*, 14, 2013.
- M. Kloft and G. Blanchard. On the convergence rate of ℓ_p -norm multiple kernel learning. *J. Mach. Learn. Res.*, 13:2465–2502, 2012.
- T. Kühn. Covering numbers of Gaussian reproducing kernel Hilbert spaces. *J. Complexity*, 27:489–499, 2011.

- S. Lin, X. Guo, and D.-X. Zhou. Distributed learning with regularized least squares. 2016. URL <https://arxiv.org/abs/1608.03339>.
- S. Mendelson and J. Neeman. Regularization in kernel learning. *Ann. Statist.*, 38:526–565, 2010.
- A. Rudi, R. Camoriano, and L. Rosasco. Less is more: Nystrom computational regularization. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1657–1665, 2015.
- N. Segata and E. Blanzieri. Empirical assessment of classification accuracy of local SVM. Technical report, University of Trento, Information Engineering and Computer Science, 2008. URL [eprints.biblio.unimn.it/1398/1/014.pdf](https://biblio.unimn.it/1398/1/014.pdf).
- N. Segata and E. Blanzieri. Fast and scalable local kernel machines. *J. Mach. Learn. Res.*, 11:1883–1926, 2010.
- J.S. Simonoff. *Smoothing Methods in Statistics*. Springer, New York, 1996.
- S. Smale and D.-X. Zhou. Estimating the approximation error in learning theory. *Anal. Appl.*, 1:17–41, 2003.
- S. Smale and D.-X. Zhou. Learning theory estimates via integral operators and their approximations. *Constr. Approx.*, 26:153–172, 2007.
- E.M. Stein. *Singular Integrals and Differentiability Properties of Functions*. Princeton University Press, Princeton, NJ, 1970.
- I. Steinwart. A fast SVM toolbox. 2016. URL <http://www.isa.uni-stuttgart.de/software/>.
- I. Steinwart and A. Christmann. *Support Vector Machines*. Springer, New York, 2008.
- I. Steinwart and C. Scovel. Mercer’s theorem on general domains: on the interaction between measures, kernels, and RKHSs. *Constr. Approx.*, 35:363–417, 2012.
- I. Steinwart, D. Hush, and C. Scovel. Optimal rates for regularized least squares regression. In S. Dasgupta and A. Klivans, editors, *Proceedings of the 22nd Annual Conference on Learning Theory*, pages 79–93, 2009.
- I. Steinwart, D. Hush, and C. Scovel. Training SVMs without offset. *J. Mach. Learn. Res.*, 12:141–202, 2011.
- T. Suzuki. Unifying framework for fast learning rate of non-sparse multiple kernel learning. In J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F.C.N. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1575–1583, 2011.
- V. Temlyakov. A remark on covering. 2013. URL <http://arxiv.org/pdf/1301.3043.pdf>.
- H. Triebel. *Theory of Function Spaces II*. Springer, Basel, 1992.
- H. Triebel. *Theory of function spaces III*. Birkhäuser, Basel, 2006.
- H. Triebel. *Theory of Function Spaces*. Birkhäuser, Basel, 2010.
- I.W. Tsang, J.T. Kwok, and P.-K. Cheung. Core vector machines: Fast SVM training on very large data sets. *J. Mach. Learn. Res.*, 6:363–392, 2005.
- I.W. Tsang, A. Kocsor, and J.T. Kwok. Simpler core vector machines with enclosing balls. In *Proceedings of the 21th international conference on Machine learning*, pages 911–918, 2007.
- V. Vapnik and L. Bottou. Local algorithms for pattern recognition and dependencies estimation. *Neural Computation*, 5:893–909, 1993.
- D. Wu, K.P. Bennett, N. Cristianini, and J. Shawe-Taylor. Large margin trees for induction and transduction. In *Proceedings of the 17th International Conference on Machine Learning*, pages 474–483, 1999.
- D.-H. Xiang and D.-X. Zhou. Classification with Gaussians and convex loss. *J. Mach. Learn. Res.*, 10:1447–1468, 2009.
- A. Zaki and Y. Ritov. Consistency and localizability. *J. Mach. Learn. Res.*, 10:827–856, 2009.
- H. Zhang, A.C. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2126–2136, 2006.
- Y. Zhang, J. Duchi, and M. Wainwright. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *J. Mach. Learn. Res.*, 16:3299–3340, 2015.
- D.-X. Zhou. The covering number in learning theory. *J. Complexity*, 18:739–767, 2002.

Bipartite Ranking: a Risk-Theoretic Perspective

Aditya Krishna Menon
Robert C. Williamson

*Data61 and the Australian National University
Canberra, ACT, Australia*

ADITYA.MENON@DATA61.CSIRO.AU
BOB.WILLIAMSON@ANU.EDU.AU

Editor: Nicolas Vayatis

Abstract

We present a systematic study of the bipartite ranking problem, with the aim of explicating its connections to the class-probability estimation problem. Our study focuses on the properties of the statistical risk for bipartite ranking with general losses, which is closely related to a generalised notion of the area under the ROC curve: we establish alternate representations of this risk, relate the Bayes-optimal risk to a class of probability divergences, and characterise the set of Bayes-optimal scorers for the risk. We further study properties of a generalised class of bipartite risks, based on the p -norm push of Rudin (2009). Our analysis is based on the rich framework of proper losses, which are the central tool in the study of class-probability estimation. We show how this analytic tool makes transparent the generalisations of several existing results, such as the equivalence of the minimisers for four seemingly disparate risks from bipartite ranking and class-probability estimation. A novel practical implication of our analysis is the design of new families of losses for scenarios where accuracy at the head of ranked list is paramount, with comparable empirical performance to the p -norm push.

Keywords: bipartite ranking, class-probability estimation, proper losses, Bayes-optimality, ranking the best

1. The Bipartite Ranking Problem

Bipartite ranking problems (Freund et al., 2003; Agarwal et al., 2005; Cléménçon et al., 2008; Kotlowski et al., 2011) have received considerable attention from the machine learning community. In such problems, we have as input a training set of examples, each of which comprises an *instance* (typically a vector of features describing some entity) with an associated *binary label* (typically denoted “positive” or “negative”, describing whether the instance possesses some attribute). The goal is to learn a *scorer*, which assigns to each instance a real number, such that positive instances have a higher score than negative instances. Violations of this condition are penalised according to some loss ℓ , and the *bipartite ranking risk* of a scorer is its expected penalty according to ℓ . When ℓ corresponds to the 0-1 loss, the bipartite ranking risk is one minus the *area under the ROC curve* (AUC) of the scorer (Agarwal and Niyogi, 2005; Cléménçon et al., 2008; Krzanowski and Hand, 2009). Applications of bipartite ranking range from content recommendation, where the goal is to rank a set of items based on an individual’s preference for them, to epidemiological studies, where the goal is to rank a set of individuals based on how likely they are to have a particular disease.

While bipartite ranking has received considerable study, the focus has primarily been on *algorithm design*. There has been relatively little theoretical study of issues such as the properties of its statistical risk, and it is only recently that its relationship to extant supervised learning problems has been formally established (Narasimhan and Agarwal, 2013b). While the design of computationally and statistically efficient methods for bipartite ranking is important, we believe there is value in explicating the statistical risk assumed by the problem, the optimal solutions that result from it, and the implied relationships to other learning problems.

To this end, in this paper,¹ we systematically study bipartite ranking through its statistical risk. In brief, we study the properties of the bipartite risk (and hence the AUC) for an arbitrary scorer, the properties of the Bayes-optimal bipartite risk and the bipartite regret for an arbitrary scorer, and characterise the set of the Bayes-optimal scorers. While some of these topics have been touched upon in prior studies, we aim to

1. A preliminary version of this work appeared in (Menon and Williamson, 2014).

provide a comprehensive, unified treatment of the material. Our analysis rests heavily upon the framework of proper losses (Buja et al., 2005; Reid and Williamson, 2010)—the machinery underlying the analysis of the class-probability estimation problem—which we hope to demonstrate to be the natural lens with which to study bipartite ranking problems. The proper loss framework has previously been employed in the analysis of a reduction of bipartite ranking to class-probability estimation (Agarwal, 2014). In this paper, we show how this framework additionally provides a clean way of generalising existing results on the Bayes-optimal scorers (§7.3, §9.5), makes transparent the connections between bipartite ranking and class-probability estimation (§10.2), and immediately establishes the equivalence of minimisers for seemingly disparate risks (§11). A novel practical implication is a means of designing losses suitable for the task of “ranking the best” (§9.6), which we show to perform favourably compared with existing approaches (§9.7).

Table 1 provides an overview of the material covered in this paper. In more detail:

- We formally define the bipartite ranking problem for a general loss via its statistical risk (§3.3), and derive its equivalence to a classification problem over pairs (§4).
- We study the properties of the ROC curve, such as its connection to the calibration transform (§5.2.5), and its value in determining thresholds for cost-sensitive classification (§5.2.6). We derive a (to our knowledge novel) result (Proposition 13) on how dominance of one calibrated scorer over another in ROC space implies dominance with respect to *any* proper composite loss, which establishes the coherence of using the ROC curve to compare calibrated scorers.
- We discuss several interpretations of the AUC, including its relationship to the bipartite risk (§5.5) and a number of integral representations (§5.6). We show how one of these representations, due to Hand (2009), is related to the integral representation for proper losses, and discuss its implications for the coherence of the use of AUC to compare scorers (§5.6.2).
- We relate the Bayes-optimal bipartite risk to an f -divergence between product measures for the class-conditional distributions (§6.2), generalising a result for the case of 0-1 loss due to Torgersen (1991). We further relate the bipartite regret to a generative Bregman divergence (§6.3).
- We determine the set of Bayes-optimal scorers for surrogate bipartite ranking risks (§7.3, §7.4, §7.5), demonstrating how the proper loss framework helps generalise existing results on the topic. We use these results to derive surrogate regret bounds, and thus AUC-consistency, for algorithms that minimise a suitable surrogate loss over pairs (§8).
- We formalise the “ranking the best” extension to bipartite ranking (§9.1), and the study the Bayes-optimal scorers for the p -norm push risk (§9.5). We show how the risk can be related to a proper composite loss with asymmetric weight function over misclassification costs (§9.6.1).
- We show how the weight function view of a proper composite loss suggests a strategy for designing losses suitable for “ranking the best”. We then describe several such new loss functions (§9.6.3). We evaluate these losses empirically on a number of real-world data sets (§9.7), and demonstrate their favourable empirical performance compared to the p -norm push risk.
- Based on the corresponding Bayes-optimal solutions, we relate bipartite ranking to the learning problems of pairwise ranking, class-probability estimation, and classification (§10.1, §10.2). This formally elucidates the relative “difficulty” of each of these problems.
- Based on the corresponding Bayes-optimal solutions, we establish the equivalence between the minimisers of seemingly disparate risks for four popular approaches to bipartite ranking (§11). This further illustrates the close links between bipartite ranking and class-probability estimation.
- We relate bipartite ranking to axiomatic characterisations of ranking relations, and in particular show how theorems characterising the existence of utility representations describe the class of ranking problems over pairs that it can model (§12.4).

Topic	Description	Reference
RISK	Bipartite ranking risk for general loss ℓ	\$3.3
	Equivalence to pairwise ranking risk	\$4
RELATION TO AUC	AUC and generalisation to a general loss ℓ	\$5.3, \$5.4
	Equivalence of bipartite ranking risk and ℓ -AUC	\$5.5
	Integral representations of the AUC and ℓ -AUC	\$5.6
	AUC and the Neyman-Pearson problem	\$D
OPTIMAL RISK	Relationship between Bayes risk and f -divergences	\$6.2
	Relationship between regret and generative Bregman divergences	\$6.3
OPTIMAL SCORERS	Bayes-optimal pair-scorers	\$7.3
	Bayes-optimal univariate scorers	\$7.4, \$7.5
SURROGATE REGRET	Surrogate regret bounds for pairwise minimisation	\$8
GENERALISED RISK	Ranking the best formulation	\$9.1
	Bayes-optimal scorers for p -norm push	\$9.5
	Proper composite approach to ranking the best	\$9.6
EQUIVALENCES	Empirical comparison of algorithms for ranking the best	\$9.7
	Reduction to classification and class-probability estimation	\$10.1, \$10.2
AXIOMATIC CHARACTERISATION	Equivalent risks for bipartite ranking	\$11
	Utility representation theorems	\$12.4

Table 1: Summary of the results on bipartite ranking in this paper.

Before initiating our study with a description of bipartite risk, we fix notation and provide definitions of key quantities that will be used throughout the paper.

2. Preliminary Definitions and Notation

We define the relevant quantities used in the rest of the paper, and fix some notation. Table 2 provides a glossary of some frequently used symbols.

2.1 Notation

We use scripted calligraphic fonts e.g. \mathcal{X}, \mathcal{Y} to denote sets. We use $\mathcal{X} \setminus \mathcal{Y}$ to denote set difference, and \emptyset to denote the empty set.

We denote by \mathbb{R} the set of real numbers, and $\mathbb{R}_+ = [0, \infty)$. For a positive integer n , we write $[n] = \{1, 2, \dots, n\}$. For a function $f: \mathcal{X} \rightarrow \mathbb{R}$, we denote its image or range by $\text{Im}(f)$. If f is differentiable, we denote its derivative by f' . For functions $f, g, f \circ g$ denotes functional composition, so that $(f \circ g)(x) = f(g(x))$. For a nonincreasing function $f: \mathbb{R} \rightarrow \mathbb{R}$, define the pseudo-inverse $f^{-1}: \mathbb{R} \rightarrow \mathbb{R}$ by

$$f^{-1}(y) = \inf\{x \in \mathbb{R} : f(x) \leq y\}. \tag{1}$$

If f is strictly decreasing, this coincides with the standard inverse function. When f is nondecreasing, we replace the inf with a sup in Equation 1. For a constant $c \in \mathbb{R}$, we write $f \equiv c$ to mean that $f(x) = c$ for every $x \in \mathcal{X}$.

Symbol	Meaning	Definition
$\mathbb{1}$	Indicator function	\$2.1
\wedge, \vee	Minimum and maximum	\$2.1
$\sigma(\cdot)$	Sigmoid function	\$2.1
\mathcal{X}	Instance space, typically \mathbb{R}^n	\$2.1
\mathcal{Y}	Label space, typically $\{\pm 1\}$	\$2.1
Δ_S	Set of all distributions over a set S	\$2.2
\mathcal{X}, \mathcal{Y}	Random variables, typically samples from D	\$2.2
D	Distribution over $\mathcal{X} \times \{\pm 1\}$	\$3.1
R	Distribution over $\mathcal{X} \times \mathcal{X} \times \{\pm 1\}$	\$3.4
D_{pair}	Pairwise ranking distribution derived from D	\$3.3
P, Q, μ, q	Class-conditional distributions and densities of D	\$3.2
M, μ	Observation distribution and density of D	\$3.2
η	Observation-conditional distribution of D	\$3.2
η_{pair}	Observation-conditional distribution of D_{pair}	\$3.3
π	Positive class base rate of D	\$3.2
$D = \langle P, Q, \pi \rangle = \langle M, \mu \rangle$	Constituent components of distribution D	\$3.1
$s: \mathcal{X} \rightarrow \mathbb{R}$	Scorer	\$2.4
$s_{\text{pair}}: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$	Pair-scorer	\$2.4
$\text{Diff}(\phi): \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$	Difference pair-scorer	\$2.4
S_{decomp}	Set of all decomposable pair-scorers	\$2.4
$\text{Pth}(\cdot; D, s): \mathbb{R} \rightarrow [0, 1]$	Score-to-probability transform	Equation 5
$\text{Call}(\cdot; D, s): \mathcal{X} \rightarrow [0, 1]$	Calibration transform	Equation 5
S, S'	Random variable and distribution of scores	\$2.4
$\ell(\cdot, \cdot)$	Loss function	\$2.6
$\ell_{\text{sym}}(\cdot, \cdot)$	Symmetrised loss function	Equation 8
$\psi(\cdot)$	Link function for a proper composite loss	\$2.6
$w(\cdot)$	Weight function for a proper composite loss	\$2.6
\mathcal{L}_{sc}	Set of all strictly proper composite losses	\$2.6
$\mathcal{L}_{\text{sc}}(\Psi)$	Set of all strictly proper composite losses with link ψ	\$2.6
$\mathcal{L}_{\text{decomp}}$	Set of all losses with decomposable Bayes-optimal pair-scorer	Equation 20
$L(\cdot, \cdot; \ell)$	Conditional risk for loss ℓ	\$3.2
$L(\cdot; \ell)$	Bayes-optimal conditional risk for loss ℓ	\$3.2
$L(\cdot; D, \ell)$	Risk for loss ℓ	\$3.2
$L_{\text{pair}}(\cdot; D, \ell)$	Bipartite risk for loss ℓ	\$3.3
$L_{\text{pair}}^*(\cdot; D, \ell)$	Bayes-optimal (minimal) risk for loss ℓ	\$3.2
$L_{\text{pair}}^{\text{min}}(D, \ell)$	Bayes-optimal (minimal) bipartite risk over scorers for loss ℓ	\$3.3
$S^*(D, \ell)$	Set of Bayes-optimal scorers for classification for loss ℓ	\$3.2
$S_{\text{pair}}^*(D, \ell)$	Set of Bayes-optimal scorers for bipartite ranking for loss ℓ	\$3.3
$\text{regret}(\cdot; D, \ell)$	Classification regret of scorer for loss ℓ	\$3.2
$\text{regret}_{\text{pair}}(\cdot; D, \ell)$	Bipartite ranking regret of scorer for loss ℓ	\$3.3
$f(\cdot, \cdot)$	f -divergence between distributions	Equation 3
$B_f(\cdot, \cdot)$	Generative Bregman divergence between distributions	Equation 4
TPR, TNR, FPR, FNR	True (false) positive (negative) rates	Equation 4
AUC, AUC $_c$	Area under the ROC curve, 0-1 and ℓ loss	\$5.3, \$5.4

Table 2: Glossary of frequently used symbols used in this paper.

For any function $f : \mathcal{X} \rightarrow \mathbb{R}$, we denote by $\text{Argmin}_{x \in \mathcal{X}} f(x)$ the set of all minimisers i.e. all $x \in \mathcal{X}$ such that $f(x) \leq f(x')$ for all $x' \in \mathcal{X}$. When the set is a singleton, so that f has a unique minimiser, we denote this minimiser by $\text{argmin}_{x \in \mathcal{X}} f(x)$.

We denote by $\text{Diff}(f) : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ the function satisfying $(\text{Diff}(f))(x, x') = f(x) - f(x')$ for every $x, x' \in \mathcal{X}$. For a set of functions $\mathcal{F} = \{f : \mathcal{X} \rightarrow \mathbb{R}\}$, we define $\text{Diff}(\mathcal{F}) = \{\text{Diff}(f) : f \in \mathcal{F}\}$.

Given any $a, b \in \mathbb{R}$, we use $a \wedge b = \min(a, b)$ and $a \vee b = \max(a, b)$. We use the Iverson bracket $(\text{Knuth}, 1992)$ $[p]$ to denote the indicator function, whose value is 1 if p is true and 0 otherwise. For any $x_0 \in \mathbb{R}$, we use $\delta_{x_0}(\cdot)$ to denote the Dirac delta centred at x_0 , which is a generalised function² satisfying $\int_{\mathbb{R}} \delta_{x_0}(x) dx = f(x_0)$ for any continuous $f : \mathbb{R} \rightarrow \mathbb{R}$.

For any $z \in \mathbb{R}$, we define $\text{sign}(z) = \llbracket z \geq 0 \rrbracket - \llbracket z \leq 0 \rrbracket$. The sigmoid function $\sigma(\cdot)$ is defined by

$$(\forall z \in \mathbb{R}) \sigma(z) = \frac{1}{1 + e^{-z}}, \quad (2)$$

with its inverse $\sigma^{-1}(\cdot)$ being the logit function,

$$(\forall y \in (0, 1)) \sigma^{-1}(y) = \log \frac{y}{1-y}.$$

2.2 Probability Distributions and Random Variables

We use sans-serif fonts e.g. X, Y to denote random variables. We denote by $X \sim D$ that X is a random variable with probability distribution D . We denote by $\mathbb{P}_{x \sim D} [X \in \mathcal{A}]$ the probability that a random draw of X according to D falls in the set \mathcal{A} . We denote by $\mathbb{E}_{x \sim D} [X]$ the expected value of the random variable X .

Given distributions P and Q such that P is absolutely continuous with respect to Q , we use $\frac{dP}{dQ}$ to denote the Radon-Nikodym density of P with respect to Q . When it exists, we refer to the density of a random variable with respect to Lebesgue measure (unless noted otherwise) by p_X . Alternately, when the random variable is clear from context, we refer to the density of the underlying distribution (e.g. Q) by the corresponding lowercase letter (e.g. q).

Given a set \mathcal{S} , we denote by $\Delta_{\mathcal{S}}$ by the set of all probability distributions on \mathcal{S} . We denote by $\text{Ber}(\theta)$ the Bernoulli distribution with parameter $\theta \in [0, 1]$.

2.3 f - and Bregman-Divergences

For convex $f : \mathbb{R}_+ \rightarrow \mathbb{R}$, the f -divergence (Csiszár, 1963) between distributions P, Q is

$$\mathbb{I}_f(P, Q) = \mathbb{E}_{x \sim Q} \left[\frac{dP}{dQ}(x) \right]. \quad (3)$$

This can be seen as a notion of discrepancy between P and Q . For normalisation purposes, one typically enforces $f(1) = 0$.

A *generative Bregman divergence* is a distinct notion of discrepancy between two probability distributions. It relies on the notion of a *Bregman divergence* (Bregman, 1967). For convex, differentiable $f : \mathbb{R} \rightarrow \mathbb{R}$, the Bregman divergence B_f between points $x, y \in \mathbb{R}$ is

$$B_f(x, y) = f(x) - f(y) - f'(y) \cdot (x - y).$$

The generative Bregman divergence B_f between distributions P, Q with densities p, q with respect to some distribution M is then the average divergence between the densities (Reid and Williamson, 2011, Section 3.3),

$$\mathbb{E}_f(P, Q) = \mathbb{E}_{x \sim M} [B_f(p(x), q(x))]. \quad (4)$$

2. Strictly, the Dirac delta is defined as a distribution or functional such that $\delta_{x_0}(f) = f(x_0)$ for every smooth f (Rudin, 1973, pg. 156; Strichartz, 1994, pg. 5), or one interprets the integral $\int_{\mathbb{R}} \delta_{x_0}(x) f(x) dx = \int_{\mathbb{R}} f(x) \mu_{x_0}(dx)$ for μ_{x_0} being the Dirac measure.

2.4 Scorers and Pair-Scorers

We are interested in supervised learning problems involving an instance or feature space \mathcal{X} (often \mathbb{R}^n), and a label space \mathcal{Y} . We call an element $x \in \mathcal{X}$ an *instance* or *feature vector*, and an element $y \in \mathcal{Y}$ a *label*. A *scorer* s for the space $(\mathcal{X}, \mathcal{Y})$ is some (measurable) function $s : \mathcal{X} \rightarrow \mathcal{V}$, where $\mathcal{V} \subseteq \mathbb{R}^{|\mathcal{Y}|}$ is the prediction space of the scorer. The magnitude of each element of s corresponds to the degree of belief in an instance having the corresponding label. A *classifier* is a scorer with $\mathcal{V} = \mathcal{Y}$, so that an instance is directly annotated with one of the labels. A *class-probability estimator* is a scorer with $\mathcal{V} = \Delta_{[|\mathcal{Y}|]}$, so that an instance is annotated by a distribution over its possible labels.

This paper focuses on the setting of *binary labels*, where $\mathcal{Y} = \{\pm 1\}$. In the case of binary labels, a scorer s for some $\mathcal{X} : \mathcal{X} \rightarrow \mathcal{V}$, where $\mathcal{V} \subseteq \mathbb{R}$. Here, classifier c is often derived from a scorer³ s via $c(x; t) = 2 \llbracket s(x) \geq t \rrbracket - 1$ for some threshold $t \in \mathbb{R}$. Similarly, a class-probability estimator f is often derived from a scorer s via $f = \Psi^{-1} \circ s$ for some (inverse) link function $\Psi^{-1} : \mathbb{R} \rightarrow [0, 1]$.

When a scorer is applied to instances drawn from some distribution, one can consider the induced distribution over scores. If X is a random variable over instances with distribution M , we denote by S the induced distribution over the scores. When it exists, we refer to the induced (marginal) distribution of the scorer as M_S , and the distributions on the positive and negative classes by P_S and Q_S respectively. We denote the marginal density of the score distribution by μ_S , and the score densities on the positive and negative classes by p_S and q_S respectively.

A *pair-scorer* s_{pair} for a product space $\mathcal{X} \times \mathcal{X}$ is some function $s_{\text{pair}} : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{V}$. The magnitude of s_{pair} corresponds to a degree of belief in the first instance having a “larger” label than the second, according to some metric. A *ranker* is a pair-scorer with $\mathcal{V} = \{\pm 1\}$. A ranker r is typically derived from a pair-scorer s_{pair} via $r(x, x'; t) = 2 \llbracket s(x, x') \geq t \rrbracket - 1$ for some threshold $t \in \mathbb{R}$. Given a (standard, or univariate) scorer s , we can construct a pair-scorer $\text{Diff}(s) : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{V} - \mathcal{V}$ (where $-$ denotes Minkowski subtraction) via

$$(\forall x, x' \in \mathcal{X}) \text{Diff}(s)(x, x') = s(x) - s(x').$$

We call a pair-scorer s_{pair} *decomposable* if

$$s_{\text{pair}} \in \mathcal{S}_{\text{Decomp}} = \{s_{\text{pair}} \in \mathcal{V}^{\mathcal{X} \times \mathcal{X}} : (\exists s : \mathcal{X} \rightarrow \mathbb{R}) s_{\text{pair}} = \text{Diff}(s)\}.$$

We call a pair-scorer *anti-symmetric* if, for every $x, x' \in \mathcal{X}$, $s_{\text{pair}}(x, x') = -s_{\text{pair}}(x', x)$. Every decomposable scorer is anti-symmetric, but not conversely.

2.5 Calibration Transform

Given a scorer s and distribution $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$, the *score-to-probability transform* $\text{Prb}(\cdot; D, s) : \mathbb{R} \rightarrow [0, 1]$ maps each score to the actual probability when the score is observed:

$$(\forall a \in \text{Im}(s)) \text{Prb}(a; D, s) = \mathbb{P}_{(X, Y) \sim D} [Y = 1 | s(X) = a].$$

We call a scorer s *calibrated* with respect to D if each predicted score equals the probability of $Y = 1$ when that prediction is made (DeGroot and Fienberg, 1983):

$$(\forall a \in \text{Im}(s)) \text{Prb}(a; D, s) = a.$$

A scorer must be a class-probability estimator to be calibrated (i.e. it cannot output values outside $[0, 1]$). The *calibration transform* $\text{Cal}(\cdot; D, s) : \mathcal{X} \rightarrow [0, 1]$ converts a scorer into a class-probability estimator via

$$(\forall x \in \mathcal{X}) \text{Cal}(x; D, s) = \text{Prb}(s(x); D, s). \quad (5)$$

It is easy to check that for any scorer $s : \mathcal{X} \rightarrow \mathbb{R}$, $\text{Cal}(\cdot; D, s)$ is automatically calibrated.

3. The case of $s(x) = t$ can be considered as a tie, thus requiring a tie-breaking scheme. The above definition corresponds to breaking ties in favour of the positive class.

2.6 Loss Functions and Conditional Risks

A *binary classification loss* ℓ , often just referred to as a *loss*, is some (measurable) function $\ell : \{\pm 1\} \times \mathbb{R} \rightarrow \mathbb{R}_+$. An important example is the 0-1 loss,⁴

$$\ell_{0,1}(y, v) = \mathbb{1}[v < 0] + \frac{1}{2} \cdot \mathbb{1}[v = 0].$$

Given a loss ℓ , we use $\ell_+(v) = \ell(1, v)$ and $\ell_-(v) = \ell(-1, v)$ to denote the individual *partial losses*. We will sometimes refer to a loss via the tuple $\ell^{(0)} = (\ell_-, \ell_+)$.

We define the *conditional ℓ -risk* $L(\cdot, \cdot; \ell)$: $[0, 1] \times \mathbb{R} \rightarrow \mathbb{R}_+$ to be

$$(\forall \eta \in [0, 1], s \in \mathbb{R}) L(\eta, s; \ell) \doteq \mathbb{E}_{Y \sim \text{Ber}(\eta)} [\ell(Y, s)] = \eta \cdot \ell_+(s) + (1 - \eta) \cdot \ell_-(s). \quad (6)$$

The *Bayes-optimal conditional ℓ -risk* $L^*(\cdot; \ell)$: $[0, 1] \rightarrow \mathbb{R}_+$ is then the best possible conditional risk,

$$(\forall \eta \in [0, 1]) L^*(\eta; \ell) \doteq \inf_{s \in \mathbb{R}} L(\eta, s; \ell).$$

For the 0-1 loss, the optimal risk is attained for any score with the same sign as $\eta - \frac{1}{2}$. More generally, we call a loss *classification calibrated* if for every $\eta \in [0, 1] \setminus \{\frac{1}{2}\}$ (Bartlett et al., 2006, Definition 1),

$$L^*(\eta; \ell) < \inf_{s: s \in (2\eta-1) \leq 0} L(\eta, s; \ell), \quad (7)$$

i.e. every optimal prediction has the same sign as $2\eta - 1$.

We call a loss ℓ *symmetric* if, for every $y \in \{\pm 1\}$ and $v \in \mathbb{R}$, $\ell(y, v) = \ell(-y, -v)$. We denote the *symmetrised version* of an arbitrary loss by

$$\ell_{\text{sym}}(v) \doteq \frac{\ell(1, v) + \ell(-1, -v)}{2}. \quad (8)$$

We call a loss ℓ a *margin loss* if $\ell(y, z) = \phi(yz)$ for some function $\phi : \mathbb{R} \rightarrow \mathbb{R}_+$. A loss is symmetric if and only if it is a margin loss; sufficiency is straightforward, and to see necessity, note that for a symmetric loss, $\ell(y, v) = \mathbb{1}[y = 1] \ell_+(v) + \mathbb{1}[y = -1] \ell_-(v) = \phi(yv) = \phi(yv)$ for $\phi(v) = \ell_+^+(v)$.

2.7 Proper and Proper-Composite Losses

A *probability estimation loss* λ is some (measurable) function $\lambda : \{\pm 1\} \times [0, 1] \rightarrow \mathbb{R}_+ \cup \{+\infty\}$. We call a probability estimation loss *proper*⁵ if its conditional risk is optimised by predicting the underlying probability (Buja et al., 2005; Reid and Williamson, 2010),

$$(\forall \eta, \eta' \in [0, 1]) L(\eta, \eta'; \lambda) \leq L(\eta, \eta'; \lambda). \quad (9)$$

We call a loss *strictly proper* if the inequality is strict.

In the following, we assume two mild regularity conditions: that $\lambda_+(1) = \lambda_-(0) = 0$, and

$$\lim_{u \rightarrow 0} u \cdot \lambda_+(u) = \lim_{u \rightarrow 1} (1 - u) \cdot \lambda_-(u) = 0.$$

4. When $V = \{\pm 1\}$, so that we are assessing a classifier, the canonical definition of 0-1 loss is $\ell_{0,1}(y, v) = \mathbb{1}[y \neq v]$. When assessing a scorer with $V = \mathbb{R}$, we simply derive a classifier from the scores and compute the resulting 0-1 loss. The only mild complication is that we consider a score of $v = 0$ to be a tie, which thus requires a tie-breaking scheme. In our definition, we break ties uniformly at random, and thus generate a *randomised classifier*. This results in the extra second term compared to the classification case.

5. Proper losses are sometimes referred to as *proper scoring rules* (Gneiting and Raftery, 2007), especially in the statistics literature. A “scoring rule” is distinct from our notion of a “scorer”: the former is a loss, and the latter is a prediction. In the literature on scoring rules, our notion of a scorer is sometimes referred to as a (probabilistic) “forecast” (Gneiting and Raftery, 2007).

Name	$\lambda_-(u)$	$\lambda_+(u)$	$w(c)$	$L^*(\eta; \lambda)$
0-1	$\mathbb{1}[u > \frac{1}{2}]$	$\mathbb{1}[u < \frac{1}{2}]$	$\delta_{1/2}(c)$	$\eta \wedge (1 - \eta)$
Cost-sensitive	$c^* \cdot \mathbb{1}[u > \frac{1}{2}]$	$(1 - c^*) \cdot \mathbb{1}[u < \frac{1}{2}]$	$\delta_{c^*}(c)$	$((1 - c^*) \cdot \eta) \wedge (c^* \cdot (1 - \eta))$
Brier	u^2	$(1 - u)^2$	2	$\eta \cdot (1 - \eta)$
Log	$-\log(1 - u)$	$-\log u$	$\frac{1}{c(1-c)}$	$-\eta \cdot \log \eta - (1 - \eta) \cdot \log(1 - \eta)$
Boosting	$\left(\frac{u}{1-u}\right)^{1/2}$	$\left(\frac{1-u}{u}\right)^{1/2}$	$\frac{1}{2 \cdot (c(1-c))^{3/2}}$	$2 \cdot \sqrt{\eta \cdot (1 - \eta)}$

Table 3: Examples of proper losses λ with associated weight functions w and conditional Bayes risks $L^*(\cdot; \lambda)$.

Name	Symbol	$\ell(y; v)$	λ	$\Psi(u)$	$\Psi^{-1}(v)$
Square	ℓ_{sq}	$\frac{1}{4} \cdot (1 - yv)^2$	Brier	$2u - 1$	$\left(\frac{v+1}{2} \vee 0\right) \wedge 1$
Logistic	ℓ_{log}	$\log(1 + e^{-yv})$	Log	$\log \frac{u}{1-u}$	$\frac{1}{1+e^v}$
Exponential	ℓ_{exp}	e^{-yv}	Boosting	$\frac{1}{2} \cdot \log \frac{u}{1-u}$	$\frac{1+e^{2v}}{1+e^v}$
Matushita	ℓ_{ms}	$\sqrt{1 + \frac{c^2}{4}} - \frac{yc}{2}$	Boosting	$\frac{2u-1}{\sqrt{u(1-u)}}$	$\frac{1}{2} \cdot \left(1 + \frac{v/2}{\sqrt{1+(v/2)^2}}\right)$

Table 4: Examples of proper composite losses with associated underlying proper loss λ and link function Ψ .

Any proper loss satisfying these conditions admits an integral representation as a weighted combination of cost-sensitive losses (Shuford Jr. et al., 1966; Schervish, 1989, Theorem 4.2),

$$\lambda : (y; u) \mapsto \int_0^1 w(c) \cdot \lambda_{\text{CS}(c)}(y, u) dc, \quad (10)$$

where $w : [0, 1] \rightarrow \mathbb{R}_+ \cup \{+\infty\}$ is called the *weight function* of the loss, and $\lambda_{\text{CS}(c)}$ is the *cost-sensitive loss*

$$\begin{aligned} \lambda_{\text{CS}(c)}(+1, u) &\doteq (1 - c) \cdot \mathbb{1}[u < c] + \frac{1}{2} \cdot \mathbb{1}[u = c] \\ \lambda_{\text{CS}(c)}(-1, u) &\doteq c \cdot \mathbb{1}[u > c] + \frac{1}{2} \cdot \mathbb{1}[u = c]. \end{aligned} \quad (11)$$

We call Equation 10 *Shuford’s representation*. A loss is strictly proper if and only if its weight function is strictly positive. One can relate the weight function and conditional Bayes risk via (Reid and Williamson, 2010, Corollary 3) $w(c) = -(L^*(\cdot; \lambda))'(c)$.

We call a loss ℓ (*strictly*) *proper composite* if there is some invertible *link function* $\Psi : [0, 1] \rightarrow \mathbb{R} \cup \{\pm\infty\}$ such that the probability estimation loss $\lambda(y; u) \doteq \ell(y; \Psi(u))$ is (strictly) proper (Reid and Williamson, 2010). By Equation 10, this implies that a proper composite loss also admits an integral representation; hence, we define the weight function of a proper composite loss as that of its underlying proper loss. We denote the set of strictly proper composite losses with invertible link function Ψ by $\mathcal{L}_{\text{SPC}}(\Psi)$, and the set of all proper composite losses by \mathcal{L}_{SPC} .

When the proper composite loss ℓ is differentiable, we have (Reid and Williamson, 2010, Corollary 12)

$$(\forall u \in \mathbb{R}) \Psi^{-1}(u) = \left(1 - \frac{\ell'(u)}{\ell_+'(u)}\right)^{-1}. \quad (12)$$

Problem	Input space	Output	Risk
Classification	$\mathcal{X} \times \mathcal{Y}$	$c : \mathcal{X} \rightarrow \mathcal{Y}$	$\mathbb{E}_{(\mathcal{X}, \mathcal{Y}) \sim D} [\ell(\mathcal{Y}, s(\mathcal{X}))]$
Class-probability estimation	$\mathcal{X} \times \mathcal{Y}$	$\hat{\eta} : \mathcal{X} \rightarrow \Delta_{ \mathcal{Y} }$	
Bipartite ranking	$\mathcal{X} \times \{\pm 1\}$	$s : \mathcal{X} \rightarrow \mathbb{R}$	$\mathbb{E}_{\mathcal{X} \sim P, \mathcal{X}' \sim Q} \ell_{\text{symm}}(s(\mathcal{X}) - s(\mathcal{X}'))$
Pairwise ranking	$\mathcal{X} \times \mathcal{X} \times \{\pm 1\}$	$s_{\text{pair}} : \mathcal{X} \times \mathcal{X} \rightarrow \{\pm 1\}$	$\mathbb{E}_{(\mathcal{X}, \mathcal{X}') \sim R} [\ell(\mathcal{Z}, s_{\text{pair}}(\mathcal{X}, \mathcal{X}'))]$

Table 5: Summary of learning problems in terms of input, output, and statistical risk.

Given a proper loss λ , we call a link function $\Psi : [0, 1] \rightarrow \mathbb{R} \cup \{\pm\infty\}$ *canonical* for that loss if

$$(\forall c \in [0, 1]) \Psi'(c) = u(c). \quad (13)$$

The canonical link function is monotone increasing, but is strictly so if and only if the loss λ is strictly proper. For a strictly proper loss λ , the proper composite loss $\ell_{\lambda}(y, v) = \lambda(y, \Psi^{-1}(v))$ is convex (Reid and Williamson, 2010, Theorem 28).

Table 3 provides some examples of popular proper losses, and Table 4 of popular proper composite losses, along with their associated underlying proper loss λ and link function Ψ .

3. Classification and Ranking: Statistical Setups

We now formally define the problems of interest in this paper. For each problem, we state the nature of their assumed input, produced output, and measure of statistical performance (or risk). Table 5 provides a summary of these problems. Our starting point is a general statistical perspective on learning from binary labels.

3.1 Learning from Binary Labels: Distributions and their Decompositions

Most problems of interest in this paper concern learning based on samples from some distribution $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ over binary labels. The precise nature of these problems shall be specified momentarily, but we first note two decompositions of D that shall prove useful. The first involves splitting any $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ into

$$\begin{aligned} (\forall A \subseteq \mathcal{X}) P(A) &= \mathbb{P}[X \in A | Y = 1] \\ (\forall A \subseteq \mathcal{X}) Q(A) &= \mathbb{P}[X \in A | Y = -1] \\ \pi &= \mathbb{P}[X \in \mathcal{X}, Y = 1]. \end{aligned}$$

Equivalently, we may decompose $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ into

$$\begin{aligned} (\forall A \subseteq \mathcal{X}) M(A) &= \mathbb{P}[X \in A, Y \in \{\pm 1\}] \\ (\forall x \in \mathcal{X}) \eta(x) &= \mathbb{P}[Y = 1 | X = x]. \end{aligned}$$

We refer to P, Q as the *class conditional distributions*, and π the *base rate*; we refer to M as the *observation distribution* and η as the *observation-conditional distribution* or *class-probability function*. We will denote the densities of P, Q with respect to M by p, q . When M possesses a density, we refer to it as μ . When we wish to refer to these constituent distributions of D and their densities, we will explicitly parameterise D as either $D = \langle P, Q, \pi \rangle$ or $D = \langle M, \eta \rangle$ as appropriate.

We now proceed to formalising the problems considered in this paper.

3.2 Classification and Class-Probability Estimation

Classification is the canonical supervised machine learning task, and has received several decades' worth of study from a theoretical and practical perspective. Classification is often motivated by appealing to several real-world problems, such as predicting whether an email message is spam based on its contents, predicting whether or not a person is sick based on test results, or predicting whether or not an individual will enjoy a movie based on its characteristics.

Formally, in statistical classification (Devroye et al., 1996), we are given samples from some distribution $D \in \Delta_{\mathcal{X} \times \mathcal{Y}}$, and wish to learn a classifier $c : \mathcal{X} \rightarrow \mathcal{Y}$. In *class-probability estimation* (Buja et al., 2005; Reid and Williamson, 2010), the input is identical, except we wish to learn a class-probability estimator $\hat{\eta} : \mathcal{X} \rightarrow \Delta_{|\mathcal{Y}|}$. In the binary case, classification involves learning some $c : \mathcal{X} \rightarrow \{\pm 1\}$ and class-probability estimation involves learning some $\hat{\eta} : \mathcal{X} \rightarrow [0, 1]$. In this setting, one typically first learns a general scorer $s : \mathcal{X} \rightarrow \mathbb{R}$, and performs either thresholding to get a classifier or a monotone transformation to get a class-probability estimator.

It remains to define how the performance of a candidate scorer $s : \mathcal{X} \rightarrow \mathbb{R}$ is assessed for the two problems. Intuitively, as these problems assume examples drawn from a probability distribution, one would like to incur a small *disutility* or *loss* for a *randomly drawn* example. This notion is captured by the notion of *statistical risk*. Given any $D = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and loss ℓ , we define the *ℓ -classification risk* for a scorer s to be the average loss incurred on a random sample from D ,

$$\mathbb{L}(s; D, \ell) \doteq \mathbb{E}_{(\mathcal{X}, \mathcal{Y}) \sim D} [\ell(\mathcal{Y}, s(\mathcal{X}))] = \mathbb{E}_{\mathcal{X} \sim M} [L(\eta(\mathcal{X}), s(\mathcal{X}); \ell)], \quad (14)$$

recalling that $L(\cdot, \cdot; \ell)$ is the conditional ℓ -risk (Equation 6). The *Bayes-optimal ℓ -classification risk*, or simply the *Bayes risk*, is the infimal ℓ -classification risk:

$$\mathbb{L}^*(D, \ell) \doteq \inf_{s : \mathcal{X} \rightarrow \mathbb{R}} \mathbb{L}(s; D, \ell).$$

Similarly, the *Bayes-optimal conditional risk*, or simply the *conditional Bayes risk* is

$$L^*(\eta; \ell) \doteq \inf_{s \in \mathbb{R}} L(\eta, s; \ell).$$

When the infimum is achievable,⁶ the set of *Bayes-optimal scorers* for a loss ℓ and distribution D comprises scorers that attain the Bayes-optimal ℓ -classification risk:

$$S^*(D, \ell) \doteq \text{Argmin}_{s : \mathcal{X} \rightarrow \mathbb{R}} \mathbb{L}(s; D, \ell).$$

Under appropriate measurability assumptions, this set may be discerned pointwise, by studying the minimisers of the conditional risk $L(\cdot, \cdot; \ell)$ (Steinwart, 2007). Finally, the *ℓ -regret* of a scorer $s : \mathcal{X} \rightarrow \mathbb{R}$ is its excess ℓ -classification risk over the Bayes ℓ -classification risk:

$$\text{regret}(s; D, \ell) \doteq \mathbb{L}(s; D, \ell) - \mathbb{L}^*(D, \ell).$$

In binary classification, the canonical goal is to minimise the risk for ℓ_{01} :

$$\mathbb{L}(s; D, \ell_{01}) = \mathbb{P}_{(\mathcal{X}, \mathcal{Y}) \sim D} [Y \cdot s(\mathcal{X}) < 0] + \frac{1}{2} \cdot \mathbb{P}_{(\mathcal{X}, \mathcal{Y}) \sim D} [s(\mathcal{X}) = 0], \quad (15)$$

i.e. we want our scorer s to achieve low misclassification error on future samples. Observe that the second term above encodes that ties in the classification are broken uniformly at random. In class probability estimation on the other hand, the canonical goal is to minimise $\mathbb{L}(s; D, \ell)$ for some proper composite loss ℓ .

6. A simple example where this is not true is the case of separable data, where $\eta(x) \in \{0, 1\}$ for every x . Under log-loss, which is proper, the optimal scorer $s^*(x) = \eta(x)$. Under logistic loss, which is proper composite, the optimal scorer is in general $s^*(x) = \log \frac{\eta(x)}{1-\eta(x)}$. But for separable data, that would require $s^*(x) \in \{\pm\infty\}$, and so the infimum is not attainable. Working with the extended reals $\mathbb{R} \cup \{\pm\infty\}$ is one possible fix, but in the sequel we will always assume the infimum is attainable.

The binary classification ℓ -risk has been extensively studied. Much of the literature has focussed on the design of losses ℓ with favourable computational and statistical properties, with specific focus on margin losses such as in the support vector machine (which employs the hinge loss, $\ell_{\text{hinge}}(y, v) = \max(0, 1 - yv)$), boosting (which employs the exponential loss $\ell_{\text{exp}}(y, v) = e^{-yv}$), and logistic regression (which employs the logistic loss $\ell_{\text{log}}(y, v) = \log(1 + e^{-yv})$). There has also been theoretical study of how regret with respect to a loss ℓ translates into the regret with respect to the 0-1 loss, via *surrogate regret bounds* (Zhang, 2004; Bartlett et al., 2006).

3.3 Instance Ranking

Many applications traditionally used to motivate binary classification are more appropriately cast as *ranking* problems. For example, rather than merely predicting an individual’s enjoyment of a single movie, it is potentially more useful to have a system capable of producing a *ranked list* of movies that the individual may enjoy. Similarly, in epidemiological studies, rather than merely predicting whether a single individual has a disease, it is potentially more useful to have a system that can produce a ranked list of individuals deemed most likely to have a particular disease. Instance ranking is of similar interest in most other settings where classification is, such as credit fraud detection and clickthrough rate analysis.

Formally, in *instance ranking* (Falkner and Hüllermeier, 2010, pg. 6; Grammer and Singer, 2001; Shashua and Levin, 2002), we are given samples from some distribution $D \in \Delta_{\mathcal{X} \times \mathcal{Y}}$, where each sample comprises a single $x \in \mathcal{X}$ and a label $y \in \mathcal{Y}$, where \mathcal{Y} is a (finite) totally ordered set. The goal is to learn a scorer $s : \mathcal{X} \rightarrow \mathbb{R}$ such that the ordering of instances by their scores mimics the ordering by their labels. Note that \mathcal{Y} , while ordered, does not necessarily have an associated metric e.g. for $\mathcal{Y} = \{\text{Hate}, \text{Indifferent}, \text{Like}\}$, there is a natural ordering over the outcomes, but it may not be possible to assign numeric values to the comparison of two outcomes. When \mathcal{Y} does not have an associated metric, instance ranking is identical to *ordinal regression* (Agresti, 1984), and thus one can solve ordinal regression problems by ranking methods (Herbrich et al., 2000). When \mathcal{Y} has an associated metric, instance ranking is a hybrid between traditional multi-class classification (where \mathcal{Y} is finite but unordered) and regression (where \mathcal{Y} is ordered but not finite) (Li and Lin, 2006).

When $\mathcal{Y} = \{\pm 1\}$, the goal of instance ranking can be seen as scoring positive examples higher than negative examples. This special case is known as the *bipartite ranking* problem, and has received considerable attention (Freund et al., 2003; Agarwal and Niyogi, 2005; Clémengon et al., 2008; Kotowski et al., 2011). Bipartite ranking is the main focus of this paper.

As per the previous section, to specify the bipartite ranking problem, we begin with its underlying risk. Given any $D = \langle P, Q, \pi \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and loss ℓ , we define the ℓ -*bipartite risk* for a scorer $s : \mathcal{X} \rightarrow \mathbb{R}$ to be

$$\begin{aligned} \mathbb{L}_{\text{BR}}(s; D, \ell) &= \mathbb{E}_{\mathcal{X} \sim P, \mathcal{X}' \sim Q} [\ell_{\text{symm}}(s(\mathcal{X}) - s(\mathcal{X}'))] \\ &= \frac{1}{\pi \cdot (1 - \pi)} \mathbb{E}_{\mathcal{X} \sim M, \mathcal{X}' \sim M'} [h(\mathcal{X}) \cdot (1 - h(\mathcal{X}')) \cdot \ell_{\text{symm}}(s(\mathcal{X}) - s(\mathcal{X}'))]. \end{aligned} \quad (16)$$

The first equation indicates that this risk is *independent* of the base rate π . When the loss ℓ is symmetric, the equation reduces to

$$\mathbb{L}_{\text{BR}}(s; D, \ell) = \mathbb{E}_{\mathcal{X} \sim P, \mathcal{X}' \sim Q} [\ell(s(\mathcal{X}) - s(\mathcal{X}'))]. \quad (17)$$

A canonical goal is to minimise the bipartite risk for ℓ_{01} , which is

$$\mathbb{L}_{\text{BR}}(s; D, \ell_{01}) = \mathbb{P}_{\mathcal{X} \sim P, \mathcal{X}' \sim Q} [s(\mathcal{X}) - s(\mathcal{X}') < 0] + \frac{1}{2} \cdot \mathbb{P}_{\mathcal{X} \sim P, \mathcal{X}' \sim Q} [s(\mathcal{X}) - s(\mathcal{X}') = 0],$$

i.e. we want s to achieve low pairwise misclassification error: in the sense of scoring a negative higher than a positive, on future samples. The reader may wonder how, if at all, this relates to the misclassification error of Equation 15; such discussion shall be deferred to §10.

Equipped with a notion of statistical risk, we can define the Bayes-optimal risk, Bayes-optimal scorers, and regret by analogy with the quantities from the previous section:

$$\begin{aligned} \mathbb{L}_{\text{BR}}^*(D, \ell) &= \inf_{s : \mathcal{X} \rightarrow \mathbb{R}} \mathbb{L}_{\text{BR}}(s; D, \ell) \\ S_{\text{BR}}^*(D, \ell) &= \text{Argmin}_{s : \mathcal{X} \rightarrow \mathbb{R}} \mathbb{L}_{\text{BR}}(s; D, \ell) \\ \text{regret}_{\text{BR}}(s; D, \ell) &= \mathbb{L}_{\text{BR}}(s; D, \ell) - \mathbb{L}_{\text{BR}}^*(D, \ell). \end{aligned} \quad (18)$$

As in binary classification, a large body of literature has focussed on the design of losses ℓ with favourable numerical and statistical properties, again with specific focus on margin losses such as in SVMRank (Joachims, 2002; Herbrich et al., 1998) (corresponding to hinge loss), RankBoost (Freund et al., 2003) (corresponding to exponential loss), and RankNet (Burgess et al., 2005) (corresponding to logistic loss).

3.4 Pairwise Ranking

Pairwise ranking problems involve labelled *pairs* of instances, where we only know which of two instances is more likely to possess some characteristic. For example, in web search click-log data, we can elicit pairwise preferences to determine which of two search results is more likely to be clicked (Joachims, 2002). (In information retrieval, the problem is sometimes referred to as “average view paper ranking”, Liu, 2009, pg. 203.)

Formally, in *pairwise ranking* (Cohen et al., 1999; Herbrich et al., 1998; Falkner and Hüllermeier, 2010, pg. 7) we are given samples from some distribution $R \in \Delta_{\mathcal{X} \times \mathcal{X} \times \{\pm 1\}}$, where each sample comprises pairs of instances $(x^{(1)}, x^{(2)})$, and a label $\{\pm 1\}$, denoting that $x^{(1)}$ ranks above or below $x^{(2)}$ respectively. Following the notation used for binary classification, we will write R as either $R = \langle P_{\text{pair}}, Q_{\text{pair}}, \pi_{\text{pair}} \rangle \in \Delta_{\mathcal{X} \times \mathcal{X} \times \{\pm 1\}}$ or $R = \langle M_{\text{pair}}, \eta_{\text{pair}} \rangle$ where for example $(\forall A \subseteq \mathcal{X} \times \mathcal{X}) P_{\text{pair}}(A) = \mathbb{P}[(\mathcal{X}, \mathcal{X}') \in A | Z = 1]$ for random variables $\mathcal{X}, \mathcal{X}', Z$ defined over the instances and label respectively.

The goal in pairwise ranking is to learn a pair-classifier $c_{\text{pair}} : \mathcal{X} \times \mathcal{X} \rightarrow \{\pm 1\}$ that specifies whether or not the first instance ranks above the second. As with standard classification, this is often done by thresholding a pair-scorer $s_{\text{pair}} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$. A pair-scorer is nothing more than a scorer defined on $\mathcal{X} \times \mathcal{X}$, and thus the notion of risk for pairwise ranking is as expected: given any $R = \langle M_{\text{pair}}, \eta_{\text{pair}} \rangle \in \Delta_{\mathcal{X} \times \mathcal{X} \times \{\pm 1\}}$ and loss ℓ , we define the ℓ -*pairwise ranking risk* for a pair-scorer s_{pair} to be

$$\mathbb{L}(s_{\text{pair}}; R, \ell) = \mathbb{E}_{(\mathcal{X}, \mathcal{X}') \sim R} [\ell(Z, s_{\text{pair}}(\mathcal{X}, \mathcal{X}'))] = \mathbb{E}_{(\mathcal{X}, \mathcal{X}') \sim M_{\text{pair}}} [L(\eta_{\text{pair}}(\mathcal{X}, \mathcal{X}'), s_{\text{pair}}(\mathcal{X}, \mathcal{X}'), \ell)]. \quad (19)$$

A canonical goal is to minimise this risk for the 0-1 loss,

$$\mathbb{L}(s_{\text{pair}}; R, \ell_{01}) = \mathbb{P}_{(\mathcal{X}, \mathcal{X}') \sim R} [Z \cdot s_{\text{pair}}(\mathcal{X}, \mathcal{X}') < 0] + \frac{1}{2} \mathbb{P}_{(\mathcal{X}, \mathcal{X}') \sim R} [s_{\text{pair}}(\mathcal{X}, \mathcal{X}') = 0],$$

i.e. we want our pair-scorer s_{pair} to achieve low pairwise misclassification error on future samples. Observe that the second term above encodes that ties in the ranking are broken uniformly at random.

As in the previous sections, we may define the Bayes-optimal risk, Bayes-optimal scorers, and regret for pairwise ranking as:

$$\begin{aligned} \mathbb{L}^*(R, \ell) &= \inf_{s_{\text{pair}} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}} \mathbb{L}(s_{\text{pair}}; R, \ell) \\ S^*(R, \ell) &= \text{Argmin}_{s_{\text{pair}} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}} \mathbb{L}(s_{\text{pair}}; R, \ell) \\ \text{regret}(s_{\text{pair}}; R, \ell) &= \mathbb{L}(s_{\text{pair}}; R, \ell) - \mathbb{L}^*(R, \ell). \end{aligned}$$

7. Nonetheless, making a distinction with a standard scorer shall be useful in our subsequent analysis.

4. Reducing Bipartite to Pairwise Ranking

Having defined the risks for three seemingly disparate problems, it is worth noting how they relate to each other. First, the pairwise ranking risk (Equation 19) is clearly equivalent to the classification risk (Equation 14), except that we operate over pairs of instances $\mathcal{X} \times \mathcal{X}$. Thus, pairwise ranking is equivalent to classification on pairs; see also §10.1. Further, we may view bipartite ranking as a special case of pairwise ranking in the following sense: any $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ may be converted to a pairwise ranking distribution, $D_{\text{BR}} \in \Delta_{\mathcal{X} \times \mathcal{X} \times \{\pm 1\}}$, such that the risks of the two problems are equivalent. Thus, the methods described above for pairwise ranking are equally applicable for bipartite ranking.

Definition 1 (Derived pairwise ranking distribution) Given any $D = \langle P, Q, \pi \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$, the derived pairwise ranking distribution $D_{\text{BR}} \in \Delta_{\mathcal{X} \times \mathcal{X} \times \{\pm 1\}}$ corresponds to drawing $(X, X', Z) \in \mathcal{X} \times \mathcal{X} \times \{\pm 1\}$ via:

- Draw $Z \sim \text{Ber}(1/2)$
- If $Z = +1$, draw $X \sim P, X' \sim Q$
- If $Z = -1$, draw $X \sim Q, X' \sim P$.

An equivalent process is:

- Draw $(X, Y) \sim D$
- Draw $(X', Y') \sim D$.
- If $Y = Y'$, reject and re-sample; else, let $Z = Y$.

The following risk equivalence can be easily verified, and is well-known for the case of ℓ_0 (Balcan et al., 2008; Kotowski et al., 2011; Agarwal, 2014). (See Proposition 62 for a proof of a more general result).

Lemma 2 For any distribution $D = \langle P, Q, \pi \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$, loss ℓ and scorer $s : \mathcal{X} \rightarrow \mathbb{R}$,

$$\mathbb{L}_{\text{BR}}(s; D, \ell) = \mathbb{L}(\text{Diff}(s); D_{\text{BR}}, \ell).$$

Proof By Equation 16,

$$\begin{aligned} \mathbb{L}_{\text{BR}}(s; D, \ell) &= \mathbb{E}_{X \sim P, X' \sim Q} [\ell_{\text{symm}}(s(X) - s(X'))] \\ &= \frac{1}{2} \cdot \mathbb{E}_{X \sim P, X' \sim Q} [\ell_1(s(X) - s(X')) + \ell_{-1}(s(X') - s(X))] \\ &= \frac{1}{2} \cdot \mathbb{E}_{X \sim P, X' \sim Q} [\ell_1(s(X) - s(X'))] + \frac{1}{2} \cdot \mathbb{E}_{X \sim P, X' \sim Q} [\ell_{-1}(s(X') - s(X))] \\ &= \frac{1}{2} \cdot \mathbb{E}_{X \sim P, X' \sim Q} [\ell_1(s(X) - s(X'))] + \frac{1}{2} \cdot \mathbb{E}_{X \sim Q, X' \sim P} [\ell_{-1}(s(X') - s(X))] \\ &= \frac{1}{2} \cdot \mathbb{E}_{(X, X') \sim (P \times Q)} [\ell_1(s(X) - s(X'))] + \frac{1}{2} \cdot \mathbb{E}_{(X, X') \sim (Q \times P)} [\ell_{-1}(s(X') - s(X))], \end{aligned}$$

where in the penultimate equation we have simply renamed the random variables in the second expression. By definition of D_{BR} and the pairwise ranking risk (Equation 19), this is exactly $\mathbb{L}(\text{Diff}(s); D_{\text{BR}}, \ell)$. ■

We summarise the conditional and marginal probabilities of D_{BR} in Appendix B. For example, if (X, X', Z) denotes the random variables distributed according to D_{BR} , it is not hard to check that

$$\begin{aligned} \mathbb{P}[(X, X') \in \mathcal{A} \times \mathcal{B} | Z = +1] &= P(\mathcal{A}) \cdot Q(\mathcal{B}) \\ \mathbb{P}[(X, X') \in \mathcal{A} \times \mathcal{B} | Z = -1] &= P(\mathcal{B}) \cdot Q(\mathcal{A}). \end{aligned}$$

The risk equivalence in Lemma 2 has a subtlety that will prove important in our subsequent analysis: the bipartite risk for a scorer relates to the pairwise risk of a decomposable pair-scorer. Consequently, we cannot in general equate the Bayes-optimal bipartite risks for the two problems, as the following makes precise.

Proposition 3 For any distribution $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and loss ℓ ,

$$\mathbb{L}_{\text{BR}}^*(D, \ell) = \mathbb{L}^*(D_{\text{BR}}, \ell) \iff S^*(D_{\text{BR}}, \ell) \cap S_{\text{Decomp}} \neq \emptyset.$$

Proof By definition,

$$\begin{aligned} \mathbb{L}_{\text{BR}}^*(D, \ell) &= \inf_{s : \mathcal{X} \rightarrow \mathbb{R}} \mathbb{L}_{\text{BR}}(s; D, \ell) \\ &= \inf_{s : \mathcal{X} \rightarrow \mathbb{R}} \mathbb{L}(\text{Diff}(s); D_{\text{BR}}, \ell) \text{ by Lemma 2} \\ &= \inf_{s_{\text{Pair}} \in S_{\text{Decomp}}} \mathbb{L}(s_{\text{Pair}}; D_{\text{BR}}, \ell). \end{aligned}$$

By contrast,

$$\mathbb{L}^*(D_{\text{BR}}, \ell) = \inf_{s_{\text{Pair}} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}} \mathbb{L}(s_{\text{Pair}}; D_{\text{BR}}, \ell).$$

The two Bayes-risks involve minimisation of the same functional, $\mathbb{L}(s_{\text{Pair}}; D_{\text{BR}}, \ell)$, but the former requires the constraint $s_{\text{Pair}} \in S_{\text{Decomp}}$. For the results of an unconstrained and constrained minimisation to coincide, at least one solution to the unconstrained minimisation must belong to the constraint set. Thus, the two Bayes-risks will coincide if and only if there is at least one minimiser of $\mathbb{L}(s_{\text{Pair}}; D_{\text{BR}}, \ell)$ that is in S_{Decomp} , i.e. $\ell \in \mathcal{L}_{\text{Decomp}}$. ■

The condition in the right hand side above shall prove important in our subsequent analysis, so much so that we shall use $\mathcal{L}_{\text{Decomp}}$ to denote the set of losses satisfying it:

$$\mathcal{L}_{\text{Decomp}} \doteq \{ \ell \mid (\forall D \in \Delta_{\mathcal{X} \times \{\pm 1\}}) S^*(D_{\text{BR}}, \ell) \cap S_{\text{Decomp}} \neq \emptyset \}. \quad (20)$$

We shall revisit this condition when computing the Bayes-optimal scorers for the bipartite ranking risk. In particular, Proposition 44 characterises the set of proper composite losses for which $\ell \in \mathcal{L}_{\text{Decomp}}$, which turns out to involve a condition on the link function for the loss.

5. The Area Under the ROC Curve (AUC) and Bipartite Risk

The canonical performance measure for a scorer in bipartite ranking is the area under the ROC curve (AUC). In this section, we formally define the ROC curve and AUC, and show how the latter is related to the bipartite risk defined in Equation 16. We then establish several properties of the ROC curve and AUC, and contrast them to those of the classification risk for proper composite losses. We also describe a generalisation of the AUC based on our general bipartite risk. While some of these results are not new, presenting them together shall further delineate the distinctions between bipartite ranking and class-probability estimation.

Before describing the ROC curve, we must define the true and false positive rates for a scorer.

5.1 True and False Positive Rates

The goal of bipartite ranking is to produce a scorer $s : \mathcal{X} \rightarrow \mathbb{R}$. However, as many practical applications require a classifier $c : \mathcal{X} \rightarrow \{\pm 1\}$, it is of interest to convert a scorer into a classifier. The simplest approach to doing so is to *threshold* the scorer at some $t \in \mathbb{R}$, producing a classifier $c(x; t) = \mathbb{I}[s(x) \geq t]$. One may assess the performance of the resulting classifier based on the *true* and *false positive rates*,⁸ which intuitively measure the accuracy (error) rates on each of the classes. These are defined below.

⁸ Depending on the literature, different terms may be used for these quantities. The true positive rate is sometimes referred to as the *recall* or *sensitivity*, and the true negative rate the *specificity*. The false positive rate is sometimes referred to as the *Type I error rate*, and the false negative rate the *Type II error rate*.

Definition 4 (True and false positive rates) Given any $D = \langle P, Q, \pi \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$, and a scorer $s: \mathcal{X} \rightarrow \mathbb{R}$, define the true (false) positive (negative) rates of the scorer at a threshold $t \in \mathbb{R} \cup \{\pm\infty\}$ to be

$$\begin{aligned} \text{TPR}(t; D, s) &\doteq \mathbb{P}_{\mathcal{X} \sim p}[\mathcal{S}(X) > t] + \frac{1}{2} \cdot \mathbb{P}_{\mathcal{X} \sim p}[\mathcal{S}(X) = t] \\ \text{FPR}(t; D, s) &\doteq \mathbb{P}_{\mathcal{X} \sim q}[\mathcal{S}(X') > t] + \frac{1}{2} \cdot \mathbb{P}_{\mathcal{X} \sim q}[\mathcal{S}(X') = t] \\ \text{TNR}(t; D, s) &\doteq 1 - \text{FPR}(t; D, s) = \mathbb{P}_{\mathcal{X} \sim q}[\mathcal{S}(X') < t] + \frac{1}{2} \cdot \mathbb{P}_{\mathcal{X} \sim q}[\mathcal{S}(X') = t] \\ \text{FNR}(t; D, s) &\doteq 1 - \text{TPR}(t; D, s) = \mathbb{P}_{\mathcal{X} \sim p}[\mathcal{S}(X) < t] + \frac{1}{2} \cdot \mathbb{P}_{\mathcal{X} \sim p}[\mathcal{S}(X) = t]. \end{aligned}$$

When the scorer s and distribution D are clear from context, we use e.g. $\text{TPR}(t)$ to denote $\text{TPR}(t; D, s)$.

In order to describe the properties of the true and false positive rates, it will be convenient to express them in terms of the distribution of the scorer. Denoting by P_S, Q_S the conditional distribution of scores on the positive and negative class when applied to instances drawn from D , we may write the true and false positive rates as:

$$\begin{aligned} \text{TPR}(t; D, s) &= \mathbb{P}_{S \sim P_S}[\mathcal{S} > t] + \frac{1}{2} \cdot \mathbb{P}_{S \sim P_S}[\mathcal{S} = t] \\ \text{FPR}(t; D, s) &= \mathbb{P}_{S' \sim Q_S}[\mathcal{S}' > t] + \frac{1}{2} \cdot \mathbb{P}_{S' \sim Q_S}[\mathcal{S}' = t]. \end{aligned}$$

The second term in each expression encodes that ties are broken uniformly at random between the positive and negative classes. Observe that the second term is zero unless there are isolated scores; i.e., the distribution of scores has a discrete random variable component. It is evident that both rates are non-increasing functions of t , and thus possess pseudo-inverses (Equation 1).

We are now in a position to describe the ROC curve and its basic properties.

5.2 The ROC Curve and Its Properties

As discussed above, given a scorer $s: \mathcal{X} \rightarrow \mathbb{R}$, one may threshold it to produce a classifier. However, it is not *a priori* clear how to choose a suitable threshold. The ROC curve (Egan, 1975; Fawcett, 2006) is a graphical representation of a scorer that spells out the implications of every threshold choice. It is formed by tracing out the relationship between the true and false positive rates of a scorer across all possible thresholds.

Definition 5 (ROC curve) Given any $D = \langle P, Q, \pi \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$, and a scorer $s: \mathcal{X} \rightarrow \mathbb{R}$, the ROC curve is defined by the parametric representation⁹

$$\text{ROC}(s; D) = \{(\text{FPR}(t; D, s), \text{TPR}(t; D, s)) : t \in \mathbb{R} \cup \{\pm\infty\}\} \subseteq [0, 1]^2.$$

Equivalently, let $\rho(\alpha; D, s)$ be the power of s at a false-positive rate of α , defined as

$$(\forall \alpha \in [0, 1]) \rho(\alpha; D, s) \doteq \text{TPR}(\text{FPR}^{-1}(\alpha)), \quad (21)$$

where we use the pseudo-inverse (Equation 1) of the false-positive rate. Then,

$$\text{ROC}(s; D) = \{(\alpha, \rho(\alpha; D, s)) : \alpha \in \text{Im}(\text{FPR})\} \subseteq [0, 1]^2.$$

Put plainly, for all possible thresholds $t \in \mathbb{R}$, we create a classifier from s using the threshold, and assess the resulting accuracy on the positive and negative classes respectively. Every point on the ROC curve represents a pair of *realisable* true and false positive rates, i.e., rates that may be attained by a classifier based on an 9. At a threshold $t = +\infty$, both the FPR and TPR are 0, while at $t = -\infty$, the FPR and TPR are both 1. Thus, varying t in this way traces a curve from left to right.

appropriate thresholding of the scorer. Thus, the ROC curve visually summarises the set of realisable error rates for various choices of classifier derived from the underlying scorer s .

In general, two points on the ROC curve represent different tradeoffs in terms of the true and false positive rates. Thus, there is no clear automated mechanism to pick the “best” threshold for generating a classifier without additional information as to one’s underlying utility.

5.2.1 BASIC PROPERTIES OF THE ROC CURVE

We make some basic observations about the ROC curve for any scorer $s: \mathcal{X} \rightarrow \mathbb{R}$ and distribution D .

- (i) we have $\{(0, 0), (1, 1)\} \subseteq \text{ROC}(s; D)$. This is because $\lim_{t \rightarrow -\infty} \text{TPR}(t) = 0$, $\lim_{t \rightarrow -\infty} \text{TPR}(t) = 1$, and similarly for the FPR.
- (ii) the curve does not have self-intersections. This is because FPR and TPR are monotone functions.
- (iii) the curve is invariant to strictly monotone increasing transformations of the scorer s . This is because the FPR and TPR are invariant to strictly monotone increasing transformations¹⁰ (see also Proposition 15).
- (iv) the curve is not necessarily surjective onto $[0, 1]^2$, and may comprise only a finite number of points in $[0, 1]^2$. This is a consequence of the fact that FPR and TPR are not necessarily *strictly* monotone.
- (v) even if the curve is surjective, the set of points below the curve is not necessarily a convex set i.e. $\rho(\cdot)$ is not necessarily a concave function.¹⁰

Figure 1 gives examples of various properties an ROC curve may possess, and in particular illustrates points (iv) and (v). We now discuss these last two points further.

5.2.2 INTERPOLATION OF THE ROC CURVE

In practice, one typically only has access to an empirical distribution \hat{D} with finite support. The resulting empirical ROC curve will thus comprise a number of isolated points. In such situations, it is common to linearly interpolate between these points. To justify this interpolation, recall that each point on the ROC curve summarises the performance of a classifier derived from a specific threshold. Every interpolated point is similarly achievable by some *randomised* classifier derived from s (Scott et al., 1998, Theorem 1, Provost and Fawcett, 2001, Theorem 7). To see why this is so, suppose $\text{ROC}(s; D) = \{(f_i, t_i)\}_{i=1}^n$, where $f_i \leq f_{i+1}$, $t_i \leq t_{i+1}$. For any $t \in [t_n - 1]$, pick any $\alpha \in [0, 1]$, and consider the interpolated point (f^t, t^t) defined by

$$(f^t, t^t) = \alpha \cdot (f_i, t_i) + (1 - \alpha) \cdot (f_{i+1}, t_{i+1}).$$

This corresponds exactly to the false positive and true positive rates of a randomised classifier derived from s using a threshold t^t , where ties are broken in favour of positives with probability α . Therefore, linear interpolation of the ROC curve summarises the performance of *all* possible classifiers with *randomised* tie-breaking that can be derived from the underlying scorer.

5.2.3 THE CONVEXIFIED ROC CURVE

To further build upon linear interpolation, one may construct a *convexified ROC curve* from the convex hull of $\text{ROC}(s; D)$ ¹¹ (Provost and Fawcett, 2001), which we denote by $\text{ROC}_{\text{cvx}}(s; D)$. As with the linearly interpolated ROC curve, every point on this curve is achievable with a suitable randomised classifier derived from the scorer (Provost and Fawcett, 2001, Theorem 7). It is not hard to justify the use of classifiers derived from $\text{ROC}_{\text{cvx}}(s; D)$, rather than those of $\text{ROC}(s; D)$: for every false positive rate, the classifiers derived from the

¹⁰ Recall a function is convex iff its epigraph is a convex set, and is concave iff its negation is convex.

¹¹ Strictly, $\text{cvx}(\cdot)$ denotes the convex hull of a set, one considers $\text{ROC}_{\text{cvx}}(s; D) = ((0, 0), (1, 1)) \cup \text{conv}(\text{ROC}(s; D))$ ($\alpha, \rho \mid \rho > \alpha$), so that only the portion of the convex hull strictly above the diagonal line is retained. When the original curve is entirely below the diagonal line, one just uses the diagonal line itself.

former possess a true positive rate at least as good as that of a classifier derived from the latter. That is, the curve $\text{ROC}_{\text{cvx}}(s; D)$ dominates $\text{ROC}(s; D)$.

When $\text{ROC}(s; D)$ comprises a number of isolated points, $\text{ROC}_{\text{cvx}}(s; D)$ is trivially the linear interpolation of the ROC curve for a scorer whose realisable true and false positive rates are the hull points. Such a scorer may be derived from s by introducing suitable ties. This process can be shown to be equivalent to performing an isotonic regression (Ayer et al., 1955) on the original scorer s (Fawcett and Niculescu-Mizil, 2007).

5.2.4 EXTREMAL ROC CURVES

Given a distribution D , different scorers will induce different ROC curves. It is natural to ask whether there is a “best” possible curve for a given D , that is, a curve that dominates every other one. It turns out that there is such a curve, corresponding to any scorer which is a strictly monotone increasing transformation of the class-probability function η . This fact is a consequence of the classical Neyman-Pearson lemma, which plays an important role in hypothesis testing. Appendix D provides some background on this lemma.

Lemma 6 Given any distribution $D = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and scorer $s : \mathcal{X} \rightarrow \mathbb{R}$,

$$(\forall \alpha \in [0, 1]) \rho(\alpha; D, s) \leq \rho(\alpha; D, \phi \circ \eta)$$

where ϕ is any strictly monotone increasing function.

Proof By the Neyman-Pearson lemma, the uniformly most powerful test (i.e. the test with maximal $\rho(\cdot)$ value) at any α is given by the likelihood ratio. For the distribution D , this is nothing but a strictly monotone transformation of η . As the ROC curve is invariant to such transformations, the result follows. ■

We may equally consider the “worst” possible curve for a given D , that is, a curve that is dominated by every other one. Since the curve for a scorer s has as mirror image the curve for the scorer $-s$, evidently, such a curve will correspond to any strictly monotone decreasing function of η , such as e.g. $s = -\eta$.

Finally, it is easy to check that a non-informative scorer s that uniformly predicts a constant (e.g. $s \equiv 0$) will induce a ROC curve that is the diagonal, viz. $\rho(\alpha) = \alpha$. Thus, intuitively, a scorer must induce a ROC curve significantly away from the diagonal in order to be useful.

5.2.5 DIFFERENTIABILITY AND CONCAVITY OF THE ROC CURVE

We now consider properties of the derivative of the ROC curve, when it exists. When the curve comprises a number of isolated points, then the derivatives of the interpolated version of the curve can be easily computed. More generally, from the definition of the power function (Equation 21), it is apparent that the differentiability of the ROC curve relies on that of the true- and false-positive rates. The following makes this precise.

Proposition 7 (Cléménçon and Vayatis, 2009, Proposition 24) Given a distribution $D = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ with M absolutely continuous and scorer $s : \mathcal{X} \rightarrow \mathbb{R}$ with corresponding induced class-conditional distributions P_S, Q_S , the curve $\text{ROC}(s; D)$ is differentiable if and only if P_S, Q_S are absolutely continuous.

While in practice one’s operating P_S, Q_S may be discrete, it is common to construct continuous approximations to them e.g. the binormal ROC model (Krzyszowski and Hand, 2009, Section 2.5). Observe that when the false and true positive rates are differentiable, we have

$$\begin{aligned} (\forall t \in \mathbb{R}) \text{TPR}'(t) &= -p_S(t) \\ \text{FPR}'(t) &= -q_S(t), \end{aligned} \tag{22}$$

where p_S, q_S are the densities of the class-conditional score distributions. When the ROC curve is differentiable, the derivative turns out to have a well-known form involving the score-to-probability transform of the scorer, as stated below. (Appendix C gives a proof for completeness.)

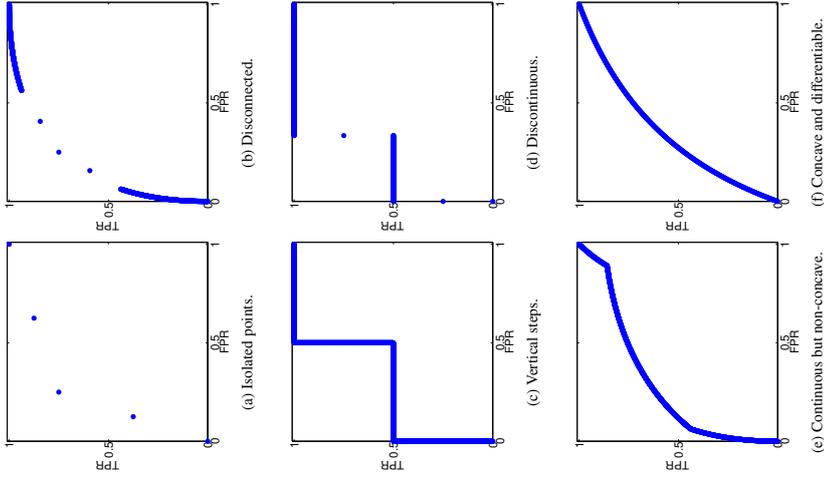


Figure 1: Illustration of various properties an ROC curve may possess. In each example, we use $\mathcal{X} = [0, 1]$, and in most cases fix $\eta(x) = x$. For (a), we use the scorer $s = \text{sign}(2 \cdot \eta - 1)$, so that the score distributions P_S, Q_S are discrete. For (b), we use the scorer $s(x) = \eta(x) \vee 0.75$ when $\eta(x) < 0.5$ and $s(x) = \eta(x) \wedge 0.25$ else, so that the score distributions have isolated areas. For (c), we round η to $\{0, 1\}$ and use the scorer s predicting uniformly 0.5 or 1 when $\eta(x) > 0.5$ and $\frac{3}{8} \cdot \eta(x)$ else, so that the positive score distribution has isolated elements. For (d), we round η to $\{0, 1\}$ and use the scorer s predicting uniformly in $[0.25, 0.5] \cup [0.9, 1]$ when $\eta(x) > 0.5$ and $[0, 0.25] \cup [0.5, 0.75]$ else, so that the score distributions have interleaved support. For (e), we use the scorer $s(x) = \eta(x) - 0.5$ when $\eta(x) < 0.5$ and $s(x) = 4 \cdot \eta(x) - 3$ else, so that the score distributions are overlapping. For (f), we use the scorer $s = \eta$, so that the result is the maximal ROC curve.

Proposition 8 (Krzyszowski and Hand, 2009, pg. 22; Clemençon and Vayatis, 2009b, Lemma 1) Given a distribution $D = \langle P, Q, \pi \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and scorer $s : \mathcal{X} \rightarrow \mathbb{R}$ such that $\text{ROC}(s; D)$ is differentiable, the slope of the ROC curve at a false positive rate $\alpha \in (0, 1)$ is

$$\begin{aligned} \rho'(\alpha) &= \frac{P_S(\text{FPR}^{-1}(\alpha))}{q_S(\text{FPR}^{-1}(\alpha))} \\ &= \frac{1 - \pi}{\pi} \cdot \frac{\text{Prb}(\text{FPR}^{-1}(\alpha))}{1 - \text{Prb}(\text{FPR}^{-1}(\alpha))}. \end{aligned} \quad (23)$$

Proposition 8 establishes that the score-to-probability transform $\text{Prb}(\cdot; D, s)$, and thus the calibration transform $\text{Cal}(\cdot; D, s)$, may be computed based on the derivatives of the ROC curve. The fact that the ROC curve may be used to obtain a class-probability estimator is well-known (Fawcett and Niculescu-Mizil, 2007; Flach, 2010). Indeed, the relationship implies a characterisation of the monotonicity of the calibration transform when the ROC curve is differentiable.

Corollary 9 Given a distribution $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and scorer $s : \mathcal{X} \rightarrow \mathbb{R}$ with differentiable ROC curve and invertible rates, $\text{ROC}(s; D)$ is (strictly) concave if and only if the function $\text{Prb}(\cdot; D, s)$ is (strictly) increasing.

Proof By Proposition 8, the derivative of the curve at any point comprises a strictly monotone transform of Prb composed with the invertible and hence strictly monotone function FPR^{-1} . As (strict) concavity of the curve is equivalent to (strict) monotonicity of its derivative, the result follows. ■

Non-concave regions of the ROC curve thus correspond to regions where the function $\text{Prb}(\cdot; D, s)$ is non-invertible. As we have seen, one may compute the convex hull of the ROC curve to remove such regions, which corresponds to introducing ties in the scorer. We note that the assumption of invertibility of rates in Corollary 9 does not by itself imply invertibility of $\text{Prb}(\cdot; D, s)$, as the former relates to the cumulative distributions of the scores, while latter relates to the densities of the scores. For example, when $P_S = Q_S$ and thus $P_S = q_S$, we might have invertibility of the rates, but we will not have invertibility of $\text{Prb}(\cdot; D, s)$.

When the scorer s is calibrated (Equation 5), Equation 23 implies that the slope of $\text{ROC}(s; D)$ at a false positive rate $\alpha \in (0, 1)$ simplifies to

$$\rho'(\alpha) = \frac{1 - \pi}{\pi} \cdot \frac{\text{FPR}^{-1}(\alpha)}{1 - \text{FPR}^{-1}(\alpha)}. \quad (24)$$

Further, Proposition 8 implies the following useful fact.

Corollary 10 Given a distribution $D = \langle P, Q, \pi \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and calibrated scorer $s : \mathcal{X} \rightarrow \mathbb{R}$ with differentiable ROC curve and invertible rates, $\text{ROC}(s; D)$ is strictly concave.

Proof As s is calibrated, by definition $\text{Prb}(\cdot; D, s)$ is the identity mapping, and thus strictly monotone. By Corollary 9 the result follows. ■

This implies that for the calibrated scorer $s = \eta$, the ROC curve is concave (provided that the curve is additionally differentiable), as noted before in e.g. (Clemençon and Vayatis, 2009, Proposition 8). We note that Corollary 10 relies crucially on the ROC curve being differentiable: the trivially calibrated scorer $s \equiv \pi$ would have an ROC curve comprising isolated points, whose interpolation would not be strictly concave.

5.2.6 THE ROC CURVE AND COST-SENSITIVE THRESHOLD SELECTION

The true positive and negative rates measure the accuracy of a classifier on the positive and negative classes respectively. Observe that the standard 0-1 classification risk (Equation 15) can be expressed in terms of the

false positive and negative rates using a threshold of 0. More generally, given a cost ratio $c \in [0, 1]$, the cost-sensitive risk of a scorer $s : \mathcal{X} \rightarrow \mathbb{R}$ when using a threshold t may be written

$$\mathbb{L}(s; D, \ell_{\text{CS}(c,t)}) = \pi \cdot (1 - c) \cdot \text{FNR}(t; D, s) + (1 - \pi) \cdot c \cdot \text{FPR}(t; D, s), \quad (25)$$

where $\ell_{\text{CS}(c,t)}$ denotes the cost-sensitive loss with cost ratio c and threshold choice t :

$$\begin{aligned} \ell_{\text{CS}(c,t)}(+1, z) &= (1 - c) \cdot \ell_0(+1, z - t) \\ \ell_{\text{CS}(c,t)}(-1, z) &= c \cdot \ell_0(-1, z - t). \end{aligned} \quad (26)$$

Clearly, $\ell_{\text{CS}(c,t)} = \ell_{\text{CS}(c,0)}$, and for $t = 0$ and $c = \frac{1}{2}$ we recover the (scaled) 0-1 loss. The optimal threshold function $t^* : [0, 1] \rightarrow 2^{\mathcal{X}}$ maps costs to the set of thresholds yielding minimal cost-sensitive risk:

$$(\forall c \in [0, 1]) t^*(c; D, s) = \underset{t \in \mathbb{R}}{\text{Argmin}} \mathbb{L}(s; D, \ell_{\text{CS}(c,t)}).$$

Determining the optimal threshold function for a scorer is intimately related to the gradient of the ROC curve. (Appendix C provides a proof for completeness.)

Proposition 11 (Krzyszowski and Hand, 2009, pg. 24) Given any $D = \langle P, Q, \pi \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and scorer $s : \mathcal{X} \rightarrow \mathbb{R}$ with differentiable ROC curve, for any cost ratio $c \in (0, 1)$, any optimal threshold $t_0 \in t^*(c; D, s)$ for the cost-sensitive risk $\mathbb{L}(s; D, \ell_{\text{CS}(c,t)})$ satisfies

$$\rho'(\text{FPR}(t_0)) = \frac{c}{1 - c} \cdot \frac{1 - \pi}{\pi},$$

or equivalently, $\text{Prb}(t_0) = c$.

Further, when the score-to-probability mapping $\text{Prb}(\cdot; D, s)$ is invertible,

$$t^*(c) = \{\text{Prb}^{-1}(c)\}.$$

Intuitively, there are multiple optimal thresholds for a given cost c when $\text{Prb}(\cdot; D, s)$ is not invertible, which by Corollary 9 occurs when the ROC curve is not strictly concave. The derivative may also have an image that is a strict subset of \mathbb{R}_+ . In this case, the risk in Equation 25 is monotone as a function of t , and so the optimal threshold is one of $\pm\infty$.

The relationship between the slope of the ROC and the optimal threshold is useful in two ways. First, given a particular c , to find the optimal threshold, we draw a line of slope $\frac{c}{1-c} \cdot \frac{1-\pi}{\pi}$. The point at which it touches the ROC curve corresponds to the optimal threshold. Second, at a given false positive rate α achieved by some threshold, the derivative of the ROC curve gives us the cost for which the given threshold is optimal.

For the case of calibrated scorers, we have a simpler characterisation of the optimal threshold for a cost-sensitive loss: it is simply the corresponding cost itself.

Corollary 12 Given any $D = \langle P, Q, \pi \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and calibrated scorer $s : \mathcal{X} \rightarrow \mathbb{R}$ with differentiable ROC curve, for any cost $c \in (0, 1)$, the optimal threshold function for the cost-sensitive risk is $t^*(c) = \{c\}$.

Proof As s is calibrated, $\text{Prb}(\cdot; D, s)$ is the identity mapping, and thus invertible. Thus, applying Proposition 11, we know there is a unique optimal threshold $t_0(c)$ for each cost ratio c . Applying Equation 24, the optimal threshold satisfies

$$\frac{t_0(c)}{1 - t_0(c)} = \frac{c}{1 - c},$$

i.e. the optimal threshold is $t_0(c) = c$. ■

Corollary 12 is again evident for the trivially calibrated scorer $s = \eta$.

5.2.7 ROC DOMINATION AND CLASSIFICATION RISK

Suppose one scorer dominates another in ROC space. What can one say about the risks of the two scorers with respect to a proper loss? The following establishes that, for calibrated scorers, dominance in ROC space implies dominance with respect to any proper composite risk. This supports the use of the ROC curve as a means of assessing the performance of a scorer. While simple to prove, the result is to our knowledge novel.

Proposition 13 *Pick any distribution $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$, and let $s_1, s_2 : \mathcal{X} \rightarrow \mathbb{R}$ be any scorers that are calibrated with respect to D , with differentiable ROC curves. If the ROC curve of s_2 dominates that of s_1 ,*

$$(\forall \alpha \in [0, 1]) \rho(\alpha; D, s_1) \leq \rho(\alpha; D, s_2).$$

then any proper loss λ, s_2 must have risk no larger than s_1 :

$$\mathbb{L}(s_2; D, \lambda) \leq \mathbb{L}(s_1; D, \lambda).$$

Proof Our basic idea will be to show that s_2 has a lower cost-sensitive risk than s_1 for any conceivable cost-ratio, which by Shuford's representation (Equation 10) will yield the desired result. For brevity, in the following we use e.g. $\text{TPR}_c(c)$ as a shorthand for $\text{TPR}(c; D, s)$.

Pick any $c \in (0, 1)$. Consider the cost-sensitive risk of a scorer s using a threshold t (Equation 25). Since s_1, s_2 are calibrated, by Corollary 12, they both have optimal threshold $t^*(c; D, s) = \{c\}$. With this threshold, scorer s_1 achieves a false-positive rate of $\alpha_1 = \text{FPR}_{s_1}(c)$, and true-positive rate $\text{TPR}_{s_1}(c)$. By the ROC domination assumption, we have

$$\text{TPR}_{s_1}(c) = \rho(\alpha_1; D, s_1) \leq \rho(\alpha_1; D, s_2) = \text{TPR}_{s_2}((\text{FPR}_{s_2})^{-1}(\alpha_1)).$$

Thus, the corresponding false negative rate of s_2 must be smaller than that of s_1 , i.e.

$$\text{FNR}_{s_2}((\text{FPR}_{s_2})^{-1}(\alpha_1)) \leq \text{FNR}_{s_1}(c).$$

This implies that using a threshold of $t_2 = (\text{FPR}_{s_2})^{-1}(\alpha_1)$, s_2 achieves a lower cost-sensitive risk than s_1 :

$$\begin{aligned} \mathbb{L}(s_2; \ell_{\text{CS}(c, t_2)}) &= \pi \cdot (1 - c) \cdot \text{FNR}_{s_2}(t_2) + (1 - \pi) \cdot c \cdot \text{FPR}_{s_2}(t_2) \\ &\leq \pi \cdot (1 - c) \cdot \text{FNR}_{s_1}(c) + (1 - \pi) \cdot c \cdot \alpha_1 \\ &= \pi \cdot (1 - c) \cdot \text{FNR}_{s_1}(c) + (1 - \pi) \cdot c \cdot \text{FPR}_{s_1}((\text{FPR}_{s_1})^{-1}(\alpha_1)) \\ &= \mathbb{L}(s_1; c, \ell_{\text{CS}(c)}). \end{aligned}$$

But since c is the optimal threshold for s_2 , we further have

$$\begin{aligned} \mathbb{L}(s_2; \ell_{\text{CS}(c)}) &= \mathbb{L}(s_2; \ell_{\text{CS}(c, c)}) \\ &\leq \mathbb{L}(s_2; \ell_{\text{CS}(c, t_2)}) \\ &\leq \mathbb{L}(s_1; \ell_{\text{CS}(c, c)}) \\ &= \mathbb{L}(s_1; \ell_{\text{CS}(c)}). \end{aligned}$$

Since s_2 has a lower cost-sensitive risk than s_1 for any cost ratio c , by Shuford's representation (Equation 10), it must also have lower risk with respect to any proper loss. ■

An immediate consequence of the above is that for scorers with strictly monotone calibration transforms $\text{Cal}(c; D, s)$ (in turn relying on strict concavity of the ROC curve, by Corollary 10), ROC dominance implies dominance with respect to any proper loss. Note also that for the optimal ROC curve, given by $s^* = \eta$, the above is trivially true as any proper loss will have its risk minimised by exactly η .

Proposition 13 relies on ROC dominance. When the ROC curves for two scorers cross, it is not hard to construct examples of losses where one scorer is superior to the other. This indicates that in such situations, caution must be used before declaring one scorer to be superior to another, as is well known (Hand, 2009).

5.3 The Area Under the ROC Curve (AUC)

The ROC curve is a graphical display of the performance of a scorer. It is often desirable to additionally have a single numeric summary of performance. One such popular summary statistic is the *area under the ROC curve*, or *AUC*.

Definition 14 (Area under the ROC curve (AUC)) *Given any $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and scorer $s : \mathcal{X} \rightarrow \mathbb{R}$, the area under the ROC curve or AUC of s is the area under the curve $\text{ROC}(s; D)$,*

$$\text{AUC}(s; D) \doteq \int_0^1 \rho(\alpha; D, s) d\alpha, \quad (27)$$

where $\rho(\alpha; D, s)$ is the power of s at α (Equation 21).

A subtlety in the above definition is that we only defined $\text{ROC}(s; D)$ in terms of the power when $\alpha \in \text{Im}(\text{FPR})$. However, the power itself is defined for every $\alpha \in [0, 1]$, due to the use of the pseudo-inverse. Thus, the integral is well-defined even when $\text{ROC}(s; D)$ comprises isolated points.

A further subtlety is that the curve traced out by $(\alpha, \text{TPR}(\text{FPR}^{-1}(\alpha))) : \alpha \in [0, 1]$ is *not* always equivalent to that generated by linear interpolation of $\text{ROC}(s; D)$. Nonetheless, the area under the two curves will in general be the same. To see this, suppose P_S, Q_S have an isolated component at some $t \in \mathbb{R}$, with $\text{FPR}(t^-) = \alpha_1, \text{FPR}(t^+) = \alpha_2$ and $\text{TPR}(t^-) = \beta_1, \text{TPR}(t^+) = \beta_2$, and further $\text{TPR}(t) = \frac{1}{2}(\beta_1 + \beta_2)$ due to the breaking of ties uniformly at random. We will then have two consecutive disconnected points in $\text{ROC}(s; D)$, (α_1, β_1) and (α_2, β_2) . With linear interpolation, this region of the ROC curve has area $\frac{1}{2} \cdot (\alpha_2 - \alpha_1) \cdot (\beta_1 + \beta_2)$.

For any $\alpha \in (\alpha_1, \alpha_2)$, we have $(\text{FPR})^{-1}(\alpha) = t$, and so $\text{TPR}((\text{FPR})^{-1}(\alpha)) = \text{TPR}(t) = \frac{1}{2}(\beta_1 + \beta_2)$. Thus with the pseudo-inverse, the corresponding area of this region is that of the corresponding rectangle with height $\frac{1}{2}(\beta_1 + \beta_2)$ and width $(\alpha_2 - \alpha_1)$, which is also exactly $\frac{1}{2} \cdot (\alpha_2 - \alpha_1) \cdot (\beta_1 + \beta_2)$.

5.3.1 BASIC PROPERTIES OF THE AUC

Some basic properties of the AUC are immediate from the above definition. First, the AUC is in $[0, 1]$; as we shall subsequently discuss, higher AUC values indicate a ‘‘better’’ scorer.

Second, the AUC is independent of the base rate π , and only depends on the class-conditional distributions P, Q for a distribution $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$. This means that a scorer s will have the same AUC with respect to all distributions in the family $\{D = \langle P, Q, \pi \rangle\}_{\pi \in [0, 1]}$.

Third, the AUC is invariant to strictly monotone increasing transforms of the scorer, as shown below.

Proposition 15 (Cl em en on and Vayatis, 2009, Proposition 24) *Given any $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and scorer $s : \mathcal{X} \rightarrow \mathbb{R}$, for any strictly monotone increasing $\phi : \mathbb{R} \rightarrow \mathbb{R}$,*

$$\text{AUC}(s; D) = \text{AUC}(\phi \circ s; D).$$

Proof This is because the 0-1 loss is invariant to monotone increasing transformations of the scorer:

$$\begin{aligned} (\forall t \in \mathbb{R}) \text{FPR}(t; D, \phi \circ s) &= \mathbb{E}_{\mathcal{X} \sim Q} [\ell_{01}(-1, \phi(s(X)) - t)] \\ &= \mathbb{E}_{\mathcal{X} \sim Q} [\ell_{01}(-1, s(X) - t)] \\ &= \text{FPR}(t; D, s), \end{aligned}$$

and similarly for TPR. It follows that the power function ρ is unaffected by ϕ , and thus so is the ROC curve and the area under it. ■

Fourth, the AUC is optimised by any strictly monotone transformation of the underlying η .

Corollary 16 Given any $D = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and any strictly monotone increasing $\phi : [0, 1] \rightarrow \mathbb{R}$,

$$\sup_{s : \mathcal{X} \rightarrow \mathbb{R}} \text{AUC}(s; D) = \text{AUC}(\phi \circ \eta; D).$$

Proof By Lemma 6, the optimal ROC curve is achieved by any $\phi \circ \eta$. Such a scorer must thus have maximal AUC. ■

We now show how the AUC can be viewed in terms of loss functions, which makes apparent its connection to the bipartite ranking risk.

5.3.2 A LOSS REPRESENTATION OF THE AUC

Although not immediately obvious from the above definition, the AUC of a scorer with respect to a distribution $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ is the probability a randomly drawn positive has a higher score than a randomly drawn negative, with ties broken uniformly at random. This observation goes back to at least Hanley and McNeil (1982, Section III), and has been noted in machine learning community in e.g. Cortes and Mohri (2003, Lemma 1), Clévençon et al. (2008).¹²

Proposition 17 (Cortes and Mohri, 2003, Lemma 1) Given any $D = \langle P, Q, \pi \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and scorer $s : \mathcal{X} \rightarrow \mathbb{R}$,

$$\text{AUC}(s; D) = \mathbb{P}_{\mathcal{X} \sim P, \mathcal{X}' \sim Q} [s(\mathcal{X}) > s(\mathcal{X}')] + \frac{1}{2} \cdot \mathbb{P}_{\mathcal{X} \sim P, \mathcal{X}' \sim Q} [s(\mathcal{X}) = s(\mathcal{X}')]. \quad (28)$$

Equation 28 is often taken as the starting definition of the AUC, due to its convenience to manipulate. Indeed, on a sample $\hat{D} = \{(\mathcal{x}_i, +1)\}_{i=1}^n \cup \{(\mathcal{x}_j, -1)\}_{j=1}^m$, the empirical AUC is

$$\text{AUC}(s; \hat{D}) = \frac{1}{n \cdot m} \cdot \sum_{i=1}^n \sum_{j=1}^m \mathbb{1}[s(\mathcal{x}_i) > s(\mathcal{x}_j)] + \frac{1}{2} \cdot \mathbb{1}[s(\mathcal{x}_i) = s(\mathcal{x}_j)],$$

which is equivalent to the Mann-Whitney statistic (Mann and Whitney, 1947). The resulting empirical estimate can be computed in $O(N \log N)$ time for $N = n + m$ with a single sort operation, rather than attempting numerical integration of the empirical ROC curve (Hand and Till, 2001).

Observe that Proposition 17 may be expressed in terms of a risk involving 0-1 loss as follows.

Corollary 18 Given any $D = \langle P, Q, \pi \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and scorer $s : \mathcal{X} \rightarrow \mathbb{R}$,

$$\begin{aligned} \text{AUC}(s; D) &= 1 - \mathbb{E}_{\mathcal{X} \sim P, \mathcal{X}' \sim Q} \left[\mathbb{1}[s(\mathcal{X}) - s(\mathcal{X}') < 0] + \frac{1}{2} \mathbb{1}[s(\mathcal{X}) = s(\mathcal{X}')] \right] \\ &= 1 - \mathbb{E}_{\mathcal{X} \sim P, \mathcal{X}' \sim Q} \left[\ell_{01}(1, s(\mathcal{X}) - s(\mathcal{X}')) \right]. \end{aligned} \quad (29)$$

Building on this representation, we now describe a generalisation of the AUC, which will be a useful basis for further analysis.

5.4 Generalisation: the ℓ -AUC

We define the following generalisation of the AUC, which uses any loss function $\ell : \{\pm 1\} \times \mathbb{R} \rightarrow \mathbb{R}_+$.

Definition 19 (ℓ -AUC) Given any $D = \langle P, Q, \pi \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and a loss $\ell : \{\pm 1\} \times \mathbb{R} \rightarrow \mathbb{R}_+$, define the ℓ -AUC of a scorer s with respect to $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ as

$$\text{AUC}(s; D, \ell) = 1 - \mathbb{E}_{\mathcal{X} \sim P, \mathcal{X}' \sim Q} \left[\ell_{\text{sym}}(s(\mathcal{X}) - s(\mathcal{X}')) \right],$$

¹² Compared to the cited works, we have introduced an extra term that accounts for ties.

ℓ	$\text{AUC}(s; D, \ell)$
ℓ_{01}	$1 - \mathbb{E}_{\mathcal{X} \sim P, \mathcal{X}' \sim Q} \left[\mathbb{1}[s(\mathcal{X}) < s(\mathcal{X}')] + \frac{1}{2} \cdot \mathbb{1}[s(\mathcal{X}) = s(\mathcal{X}')] \right]$
ℓ_{sq}	$1 - \mathbb{E}_{\mathcal{X} \sim P, \mathcal{X}' \sim Q} \left[(1 - s(\mathcal{X}) + s(\mathcal{X}'))^2 \right]$
ℓ_{\log}	$1 - \mathbb{E}_{\mathcal{X} \sim P, \mathcal{X}' \sim Q} \left[\log \left(1 + e^{s(\mathcal{X}') - s(\mathcal{X})} \right) \right]$
ℓ_{exp}	$1 - \mathbb{E}_{\mathcal{X} \sim P, \mathcal{X}' \sim Q} \left[e^{s(\mathcal{X}') - s(\mathcal{X})} \right]$

Table 6: Examples of ℓ -AUC for various losses, given some $D = \langle P, Q, \pi \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$.

recalling that ℓ_{sym} is the symmetrised version of ℓ (Equation 8). When ℓ is symmetric, this simplifies to

$$\text{AUC}(s; D, \ell) = 1 - \mathbb{E}_{\mathcal{X} \sim P, \mathcal{X}' \sim Q} \left[\ell(s(\mathcal{X}) - s(\mathcal{X}')) \right].$$

Table 6 provides some examples of the ℓ -AUC. Clearly, $\text{AUC}(s; D, \ell_{01})$ is the standard AUC as defined earlier (see e.g. Equation 28). When we do not explicitly refer to the loss, it is understood that we are referring to the standard AUC.

When using the cost-sensitive loss $\ell_{\text{CS}(c)}$ of Equation 26 with a threshold of $\tau = 0$ and any $c \in (0, 1)$, we also recover the standard AUC. This is because the symmetrised version of such a loss is

$$\begin{aligned} \ell_{\text{CS}(c;0)}(1, v) + \ell_{\text{CS}(c;0)}(-1, -v) &= (1 - c) \cdot \mathbb{1}[v < 0] + c \cdot \mathbb{1}[-v > 0] + \frac{1}{2} \cdot \mathbb{1}[v = 0] \\ &= (1 - c) \cdot \mathbb{1}[v < 0] + c \cdot \mathbb{1}[v < 0] + \frac{1}{2} \cdot \mathbb{1}[v = 0] \\ &= \mathbb{1}[v < 0] + \frac{1}{2} \cdot \mathbb{1}[v = 0] \\ &= \ell_{01}(1, v) \end{aligned}$$

This is intuitively because of the symmetry inherent in the bipartite ranking problem: scoring a positive below a negative is equivalent to scoring a negative above a positive. Therefore, one cannot expect to have different costs associated with the two errors.

We have assumed the prediction space for the loss ℓ above to be \mathbb{R} because we require the prediction space to be closed under negation, and for an arbitrary scorer that requires we can compute the loss for any real valued prediction. This rules out using a proper loss (or indeed any probability estimation loss), which is only defined on the prediction space $[0, 1]$. However, we may use a proper composite loss, which operates on a real-valued prediction space but converts this to $[0, 1]$ via a link function.

5.4.1 THE ℓ -AUC AS AN AREA

One property of the AUC that is inherited is that the ℓ -AUC may be interpreted as the area under a curve parameterised by the false positive rate, and a smoothed version of the true positive and false positive rates, defined below.

Definition 20 Given any $D = \langle P, Q, \pi \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$, loss ℓ , and a scorer $s : \mathcal{X} \rightarrow \mathbb{R}$, define the ℓ -true (false) positive (negative) rates at a threshold $t \in \mathbb{R}$ to be

$$\begin{aligned} \text{FNR}(t; D, s, \ell) &= \mathbb{E}_{\mathcal{X} \sim P} [\ell_1(s(\mathcal{X}) - t)] \\ \text{FPR}(t; D, s, \ell) &= \mathbb{E}_{\mathcal{X} \sim Q} [\ell_{-1}(s(\mathcal{X}) - t)] \\ \text{TPR}(t; D, s, \ell) &= 1 - \text{FNR}(t; D, s, \ell) \end{aligned}$$

$$\text{TNr}(t; D, s, \ell) = 1 - \text{FPr}(t; D, s, \ell).$$

When the scorer s and distribution D are clear from context, we shall use e.g. $\text{TPR}_\ell(t)$ to denote $\text{TPR}(t; D, s, \ell)$. We then have the following analogue to Equation 27 for a general loss.

Proposition 21 Given any $D \in \Delta_{\mathbb{X} \times \{\pm 1\}}$, loss ℓ , and scorer $s : \mathbb{X} \rightarrow \mathbb{R}$ with differentiable ROC curve and invertible false- and true-positive rates¹³,

$$\text{AUC}(s; D, s, \ell) = \int_0^1 \rho(\alpha; D, s, \ell) d\alpha, \quad (30)$$

where $\rho(\alpha; D, s, \ell)$ is the ℓ -power of s at α , defined for $\alpha \in [0, 1]$ as

$$\rho(\alpha; D, s, \ell) = \frac{1}{2} \cdot (\text{TPR}_\ell(\text{FPr}^{-1}(\alpha)) + \text{TNr}_\ell(\text{TPR}^{-1}(\alpha))). \quad (31)$$

When ℓ is symmetric, this simplifies to

$$\text{AUC}(s; D, \ell) = \int_0^1 \text{TPR}_\ell(\text{FPr}^{-1}(\alpha)) d\alpha = \int_0^1 \text{TNr}_\ell(\text{TPR}^{-1}(\alpha)) d\alpha.$$

Proof The proof is a generalisation of e.g. Cortes and Mohri (2003, Lemma 1); Cléménçon et al. (2008, Proposition B.2) to cover an arbitrary loss $\ell : \{\pm 1\} \times \mathbb{R} \rightarrow \mathbb{R}_+$. Recall that for any $t \in \mathbb{R}$,

$$\begin{aligned} \text{TPR}_\ell(t) &= 1 - \frac{\mathbb{E}}{\mathbb{X} \rightarrow P} [\ell_1(s(X) - t)] \\ \text{TNr}_\ell(t) &= 1 - \frac{\mathbb{E}}{\mathbb{X} \rightarrow Q} [\ell_{-1}(s(X) - t)]. \end{aligned}$$

Starting from the definition, and by swapping the expectation and integral (which is justified by Tonelli's theorem Folland, 1999, pg. 67, since ℓ_1 is nonnegative and measurable):

$$\begin{aligned} \int_0^1 \text{TPR}_\ell(\text{FPr}^{-1}(\alpha)) d\alpha &= 1 - \int_0^1 \frac{\mathbb{E}}{\mathbb{X} \rightarrow P} [\ell_1(s(X) - \text{FPr}^{-1}(\alpha))] d\alpha \\ &= 1 - \frac{\mathbb{E}}{\mathbb{X} \rightarrow P} \left[\int_0^1 \ell_1(s(X) - \text{FPr}^{-1}(\alpha)) d\alpha \right] \\ &= 1 - \frac{\mathbb{E}}{\mathbb{X} \rightarrow P} \left[- \int_{-\infty}^{\infty} \ell_1(s(X) - t) \cdot \text{FPr}'(t) dt \right] \text{ with } \alpha = \text{FPr}(t) \\ &= 1 - \frac{\mathbb{E}}{\mathbb{X} \rightarrow P} \left[\int_{-\infty}^{\infty} \ell_1(s(X) - t) \cdot q_S(t) dt \right] \\ &= 1 - \frac{\mathbb{E}}{\mathbb{X} \rightarrow P} \left[\int_{-\infty}^{\infty} \ell_1(s(X) - t) \cdot \frac{\mathbb{E}}{\mathbb{X} \rightarrow Q} [\delta_{s(X)}(t)] dt \right] \\ &= 1 - \frac{\mathbb{E}}{\mathbb{X} \rightarrow P \times \mathbb{X} \rightarrow Q} [\ell_1(s(X) - s(X'))], \end{aligned}$$

where the last line follows from the definition of Dirac delta. Similarly,

$$\begin{aligned} \int_0^1 \text{TNr}_\ell(\text{TPR}^{-1}(\alpha)) d\alpha &= 1 - \int_0^1 \frac{\mathbb{E}}{\mathbb{X} \rightarrow Q} [\ell_{-1}(s(X') - \text{TPR}^{-1}(\alpha))] d\alpha \\ &= 1 - \frac{\mathbb{E}}{\mathbb{X} \rightarrow Q} \left[\int_0^1 \ell_{-1}(s(X') - \text{TPR}^{-1}(\alpha)) d\alpha \right] \end{aligned}$$

¹³. This restriction ensures that we can employ the integral substitution formula in the proof.

$$\begin{aligned} &= 1 - \frac{\mathbb{E}}{\mathbb{X} \rightarrow Q} \left[- \int_{-\infty}^{\infty} \ell_{-1}(s(X') - t) \cdot \text{TPR}'(t) dt \right] \\ &= 1 - \frac{\mathbb{E}}{\mathbb{X} \rightarrow Q} \left[\int_{-\infty}^{\infty} \ell_{-1}(s(X') - t) \cdot p_S(t) dt \right] \\ &= 1 - \frac{\mathbb{E}}{\mathbb{X} \rightarrow Q} \left[\int_{-\infty}^{\infty} \ell_{-1}(s(X') - t) \cdot \frac{\mathbb{E}}{\mathbb{X} \rightarrow P} [\delta_{s(X)}(t)] dt \right] \\ &= 1 - \frac{\mathbb{E}}{\mathbb{X} \rightarrow Q \times \mathbb{X} \rightarrow P} [\ell_{-1}(s(X') - s(X))]. \end{aligned}$$

Thus,

$$\text{AUC}(s; D, \ell) = \frac{1}{2} \int_0^1 (\text{TPR}_\ell(\text{FPr}^{-1}(\alpha)) + \text{TNr}_\ell(\text{TPR}^{-1}(\alpha))) d\alpha. \quad \blacksquare$$

For a symmetric loss, we can thus interpret the ℓ -AUC as being the area under a curve that plots the false positive rates of a scorer against the ℓ -true positive rates, or equivalently, the true positive rates against the ℓ -true negative rates. (These two curves are not equivalent in general, but the area under them is.) Clearly, when $\ell = \ell_{0,1}$, the ℓ -power is the standard power (Equation 21), and the above is exactly the standard AUC.

5.4.2 BASIC PROPERTIES OF THE ℓ -AUC

Like the standard AUC, the ℓ -AUC does not depend on the base rate x . However, the ℓ -AUC does not inherit all properties of the standard AUC. First, in general the ℓ -AUC lies in \mathbb{R} , not necessarily $[0, 1]$. Second, the ℓ -AUC is not invariant to strictly monotone increasing transforms of the scoring functions. Indeed, the ℓ -AUC penalises the *magnitude* of differences between predictions. Intuitively, this makes the ℓ -AUC closer to a classification or class-probability estimation risk, as it is not sufficient to simply order instances well. Similar magnitude-sensitive metrics have been explored by Wu and Flach (2005); Ferri et al. (2005); for example, Wu and Flach (2005, Equation 5) proposed the ‘‘scored AUC’’,

$$\text{AUC}_{\text{scor}}(s; D) \doteq \frac{\mathbb{E}}{\mathbb{X} \rightarrow P \times \mathbb{X} \rightarrow Q} [\max(0, s(X) - s(X'))].$$

corresponding to the ℓ -AUC with non-convex loss $\ell_1(v) = (1 - v) \wedge 1$.

5.5 The ℓ -AUC and Bipartite Ranking Risk

As mentioned, the AUC is a canonical measure of performance for bipartite ranking problems. Thus far, we have used the bipartite risk (Equation 16) as our measure of performance. In fact, these measures are equivalent: from the definition of the ℓ -AUC for a scorer s (Definition 19), it is apparent that it is a linear transformation of the bipartite ranking risk for the pair-scorer $\text{Diff}(s)$. Equivalently, by Lemma 2, the ℓ -AUC may be seen as a linear transformation of the pairwise ranking risk over D_{BR} .

Lemma 22 For any $D \in \Delta_{\mathbb{X} \times \{\pm 1\}}$, loss ℓ , and scorer $s : \mathbb{X} \rightarrow \mathbb{R}$,

$$\begin{aligned} \text{AUC}(s; D, \ell) &= 1 - \mathbb{L}_{\text{BR}}(s; D, \ell) \\ &= 1 - \mathbb{L}(\text{Diff}(s); D_{\text{BR}}, \ell), \end{aligned} \quad (32) \quad (33)$$

so that the ℓ -AUC can be seen as a linear transform of ℓ -bipartite ranking risk.

We may further relate the Bayes-optimal scorers for AUC and bipartite risk. For a distribution $D \in \Delta_{\mathbb{X} \times \{\pm 1\}}$, define the Bayes-optimal ℓ -AUC to be the supremal ℓ -AUC:

$$\text{AUC}^*(D, \ell) = \sup_{s : \mathbb{X} \rightarrow \mathbb{R}} \text{AUC}(s; D, \ell).$$

Representation of $\text{AUC}(s; D)$	Interpretation	Reference
$\int_0^1 \text{TPR}(\text{FPR}^{-1}(a)) da$	Area under ROC curve	Equation 27
$\int_{\mathbb{X} \sim P \times \mathbb{X}' \sim Q} [s(\mathbb{X}) > s(\mathbb{X}')] + \frac{1}{2} \cdot \mathbb{P}_{\mathbb{X} \sim P \times \mathbb{X}' \sim Q} [s(\mathbb{X}) = s(\mathbb{X}')]$	Probability of random positive scoring higher than random negative	Equation 28
$1 - \mathbb{E}_{\mathbb{X} \sim P \times \mathbb{X}' \sim Q} [\ell_{01}(1, s(\mathbb{X}) - s(\mathbb{X}'))]$	Average 0-1 accuracy on pairs	Equation 29
$1 - \mathbb{L}_{\text{BR}}(\text{Diff}(s); D, \ell_{01})$	One minus bipartite ranking risk	Equation 32
$1 - \frac{1}{2\pi(1-\pi)} \mathbb{E}_{\mathbb{X} \times \mathbb{X}' \sim D} \left[\int_0^1 \mathcal{H}_{\text{Cal}(c; D, s)}(c) dc \right]$	Weighted combination of cost-sensitive losses	Equation 34
$2 \cdot \mathbb{E}_{\mathbb{X} \sim P} [\text{BACC}(c; \mathbb{X}; D, s)] - \frac{1}{2}$	Average balanced accuracy	Equation 39
$\mathbb{E}_{\mathbb{X} \sim P} [\text{TPR}(c; \mathbb{X})]$	Average rank of positive examples	Equation 37
$\mathbb{E}_{\mathbb{X}' \sim Q} [\text{TPR}(c; \mathbb{X}')]$	Average rank of negative examples	Equation 38

Table 7: Various representations for the AUC of a scorer $s: \mathbb{X} \rightarrow \mathbb{R}$ with respect to a distribution $D \in \Delta_{\mathbb{X} \times \{\pm 1\}}$. Each gives a different interpretation, and possibly means of estimating the AUC.

This supremal AUC is simply one minus the Bayes-optimal bipartite ranking risk: thus, it is a measure of the inherent difficulty of bipartite ranking with a given distribution D .

Corollary 23 For any $D \in \Delta_{\mathbb{X} \times \{\pm 1\}}$ and loss ℓ ,

$$\text{AUC}^*(D, \ell) = 1 - \mathbb{L}_{\text{BR}}^*(D, \ell).$$

Proof Take the supremum of both terms in Equation 32. ■

5.6 Alternate Representations of the AUC

We now outline several equivalent representations of the AUC, summarised in Figure 7. Each gives a different perspective about how it measures the performance of a scorer, as well as potentially different means of estimating it from samples. Many of these representations are specific to ℓ_{01} , but we begin with two related representations that hold for general ℓ .

5.6.1 THE SHUFORD REPRESENTATION

Suppose the loss ℓ is proper composite with link Ψ . Recall Shuford’s integral representation (Equation 10),

$$\ell(y; v) = \int_0^1 u(c) \cdot \lambda_{\text{CS}(c)}(y; \Psi^{-1}(c)) dc.$$

We can apply this to the definition of pairwise ranking risk (Equation 19) to get a representation of the bipartite ranking risk in terms of cost-sensitive bipartite ranking risks, assuming Ψ has some symmetry.

Proposition 24 For any $D \in \Delta_{\mathbb{X} \times \{\pm 1\}}$, $\ell \in \mathcal{L}_{\text{SRC}}(\Psi)$ where $\Psi^{-1}(-v) = 1 - \Psi^{-1}(v)$ and scorer $s: \mathbb{X} \rightarrow \mathbb{R}$,

$$\mathbb{L}_{\text{BR}}(s; D, \ell) = \int_0^1 u(c) \cdot \mathbb{L}(\Psi^{-1} \circ \text{Diff}(s); D_{\text{BR}}, \lambda_{\text{symm,CS}(c)}) dc.$$

where $u(\cdot)$ is the weight function corresponding to the proper loss $\lambda = \ell \circ \Psi^{-1}$, and

$$(\forall c \in [0, 1]) (\forall u \in [0, 1]) \lambda_{\text{symm,CS}(c)}(u) = \frac{\lambda_{\text{CS}(c)}(+1, u) + \lambda_{\text{CS}(c)}(-1, 1 - u)}{2}.$$

Proof By the equivalence of bipartite ranking and classification on pairs, the given condition on the link function, and applying Shuford’s representation to the partial losses $\ell_{\pm 1}$,

$$\begin{aligned} \mathbb{L}_{\text{BR}}(s; D, \ell) &= \mathbb{L}(\text{Diff}(s); D_{\text{BR}}, \ell) \\ &= \mathbb{E}_{(\mathbb{X}, \mathbb{X}') \sim (P \times Q)} [\ell_{\text{symm}}(\text{Diff}(s))(\mathbb{X}, \mathbb{X}')] \\ &= \mathbb{E}_{(\mathbb{X}, \mathbb{X}') \sim (P \times Q)} \left[\frac{\ell_1(\text{Diff}(s))(\mathbb{X}, \mathbb{X}') + \ell_{-1}(-\text{Diff}(s))(\mathbb{X}, \mathbb{X}')}{2} \right] \\ &= \mathbb{E}_{(\mathbb{X}, \mathbb{X}') \sim (P \times Q)} \left[\frac{\ell_1(\Psi^{-1}(\text{Diff}(s))(\mathbb{X}, \mathbb{X}')) + \lambda_{-1}(\Psi^{-1}(-\text{Diff}(s))(\mathbb{X}, \mathbb{X}'))}{2} \right] \\ &= \mathbb{E}_{(\mathbb{X}, \mathbb{X}') \sim (P \times Q)} \left[\frac{\lambda_1(\Psi^{-1}(\text{Diff}(s))(\mathbb{X}, \mathbb{X}')) + \lambda_{-1}(1 - \Psi^{-1}(\text{Diff}(s))(\mathbb{X}, \mathbb{X}'))}{2} \right] \\ &= \mathbb{E}_{(\mathbb{X}, \mathbb{X}') \sim (P \times Q)} \left[\frac{\lambda_{\text{CS}(c)}(1, \Psi^{-1}(\text{Diff}(s))(\mathbb{X}, \mathbb{X}')) + \lambda_{\text{CS}(c)}(-1, 1 - \Psi^{-1}(\text{Diff}(s))(\mathbb{X}, \mathbb{X}'))}{2} \right] \\ &= \int_0^1 u(c) \cdot \mathbb{L}(\Psi^{-1} \circ \text{Diff}(s); D_{\text{BR}}, \lambda_{\text{symm,CS}(c)}) dc, \end{aligned}$$

where

$$\lambda_{\text{symm,CS}(c)}(u) = \frac{\lambda_{\text{CS}(c)}(+1, u) + \lambda_{\text{CS}(c)}(-1, 1 - u)}{2}.$$

Without the assumption on Ψ above, one can still obtain an integral representation, but it will not cleanly relate to weighted combination of probability estimation loss risks.

Given the connection between the ℓ -AUC and bipartite ranking risk (Lemma 22), one might hope for a result of the form

$$\text{AUC}(s; D, \ell) = \int_0^1 u(c) \cdot \text{AUC}(s; D, \lambda_{\text{CS}(c)}) dc.$$

However, the AUC can only be equated with the pairwise ranking risk when the latter uses a decomposable scorer (Equation 33). In the above equation, if we require $\Psi^{-1} \circ \text{Diff}(s)$ to be decomposable, then it must be that Ψ^{-1} is the identity function, i.e. the loss ℓ must be proper, and not merely proper composite. But recall that the ℓ -AUC is not defined for a proper loss, because its prediction space is not closed under negation. Thus, an integral representation cannot be found here, either.

Interestingly, if we allow the weights to depend on the scorer s , it is possible to get an expression for the AUC with strong resemblance to Shuford’s result for proper losses, as we now explore.

5.6.2 HAND’S REPRESENTATION

We now generalise a result of Hand (2009, Section 4) to the case of general ℓ -AUC. Informally, the result is a representation of the ℓ -AUC as a weighted combination of cost-sensitive risks, where the weighting is distribution and scorer dependent. More specifically, the result states that the AUC of a scorer s is a weighted combination of cost-sensitive risks, where the weighting factor on costs depends on s , and the threshold for cost c is set to the optimal choice (c.f. Proposition 11) of $\text{Prb}^{-1}(c)$.

Proposition 25 Given any $D \in \Delta_{\mathbb{X} \times \{\pm 1\}}$, loss ℓ , and scorer $s: \mathbb{X} \rightarrow \mathbb{R}$ such that $\text{ROC}(s; D)$ is differentiable and $\text{Prb}: D, s$ is differentiable and invertible,

$$\text{AUC}(s; D, \ell) = 1 - \frac{1}{2\pi(1-\pi)} \int_0^1 Y_s(c) \cdot \mathbb{L}(s; D, \ell_{\text{trans}(c)}) dc$$

where the transformed loss $\ell_{\text{trans}(c)}$ with parameter c is

$$\begin{aligned} (\ell_{\text{trans}(c)})(y, v) &= (\mathbb{T}(\ell, c, \text{Prb}^{-1}(c)))(y, v) \\ (\mathbb{T}(\ell, c, t))(y, v) &= (1 - c) \cdot \mathbb{I}[y = 1] \cdot \ell_1(v - t) + c \cdot \mathbb{I}[y = -1] \cdot \ell_{-1}(v - t), \end{aligned}$$

and the weighting factor is

$$V_S(c) = (\text{Prb}^{-1})'(c) \cdot \mu_S(\text{Prb}^{-1}(c))$$

for marginal distribution over scores μ_S .

Proof By the rate-based representation from Equations 30, 31,

$$1 - \text{AUC}(s; D, \ell) = \frac{1}{2} \int_{-\infty}^{\infty} (\text{TNR}'(t) \cdot \text{FNR}'_\ell(t) + \text{FNR}'(t) \cdot \text{FPR}'_\ell(t)) dt.$$

Recall from §2.4 that we refer to the marginal density of scores by μ_S , and the class-conditional densities by p_S, q_S . Recall from Equation 22 that for any $t \in \mathbb{R}$,

$$\begin{aligned} \text{TNR}'(t) &= q_S(t) \\ &= \frac{1}{1 - \pi} \cdot \mu_S(t) \cdot (1 - \text{Prb}(t)), \end{aligned}$$

and similarly,

$$\text{FNR}'(t) = \frac{1}{\pi} \cdot \mu_S(t) \cdot \text{Prb}(t).$$

Thus, the integrand is

$$\begin{aligned} \ell(t) &= \text{TNR}'(t) \cdot \text{FNR}'_\ell(t) + \text{FNR}'(t) \cdot \text{FPR}'_\ell(t) \\ &= \frac{1}{\pi(1 - \pi)} \cdot \mu_S(t) \cdot (\pi \cdot (1 - \text{Prb}(t)) \cdot \text{FNR}'_\ell(t) + (1 - \pi) \cdot \text{Prb}(t) \cdot \text{FPR}'_\ell(t)) \\ &= \frac{1}{\pi(1 - \pi)} \cdot \mu_S(t) \cdot \mathbb{L}(s; D, \ell_{\text{trans}(\text{Prb}(t))}), \end{aligned}$$

where

$$\begin{aligned} (\ell_{\text{trans}(c)})(y, v) &= (\mathbb{T}(\ell, c, \text{Prb}^{-1}(c)))(y, v) \\ (\mathbb{T}(\ell, c, t))(y, v) &= (1 - c) \cdot \mathbb{I}[y = 1] \cdot \ell_1(v - t) + c \cdot \mathbb{I}[y = -1] \cdot \ell_{-1}(v - t), \end{aligned}$$

transforms the loss ℓ to use a cost weighting c and corresponding optimal threshold $\text{Prb}^{-1}(c)$ (c.f. Proposition 11). Thus, returning the original integral,

$$\begin{aligned} 1 - \text{AUC}(s; D, \ell) &= \frac{1}{2\pi(1 - \pi)} \int_{-\infty}^{\infty} \mu_S(t) \cdot \mathbb{L}(s; D, \ell_{\text{trans}(\text{Prb}(t))}) dt \\ &= \frac{1}{2\pi(1 - \pi)} \int_0^1 V_S(c) \cdot \mathbb{L}(s; D, c) dc, \end{aligned}$$

using the substitution $c = \text{Prb}(t)$, with $V_S(c) = (\text{Prb}^{-1})'(c) \cdot \mu_S(\text{Prb}^{-1}(c))$. ■

The right hand side above features two seemingly opaque objects: a transformed version of the loss and a weighting factor, both of which depend on the score-to-probability transformation $\text{Prb}(c; D, s)$. Fortunately, both of these simplify when we consider ℓ_{01} (corresponding to the standard AUC), and calibrated scorers. First, it is easy to check that for ℓ_{01} , the transformed loss is the cost-sensitive loss with threshold t ,

$$(\mathbb{T}(\ell, c, t))(y, v) = \ell_{\text{CS}(c,t)}(y, v)$$

where $\ell_{\text{CS}(c,t)}$ is as in Equation 26. Further, if we calibrate our scorer, we find

$$\begin{aligned} (\text{Prb} \circ \text{Cal})(t) &= \mathbb{P}[Y = 1 | s(X) = \text{Prb}^{-1}(t)] = t \\ V_{\text{Cal}(c; D, s)}(c) &= \mu_C(c), \end{aligned}$$

where C is the distribution of the calibrated scorer $\text{Cal}(\cdot; D, s)$. When the calibration transform is invertible, we may use this to express the standard AUC as the following, which is also a consequence of Hand (2009, Equation 6).

Corollary 26 Given any $D \in \Delta_{\mathcal{X} \times (\pm 1)}$, loss ℓ , and scorer $s: \mathcal{X} \rightarrow \mathbb{R}$ such that $\text{ROC}(s; D)$ is differentiable and $\text{Cal}(\cdot; D, s)$ is strictly monotone increasing,

$$\text{AUC}(s; D) = 1 - \frac{1}{2\pi(1 - \pi)} \cdot \mathbb{E}_{(X, Y) \sim D} \left[\int_0^1 \mu_C(c) \cdot \mathcal{A}_{\text{CS}(c)}(Y, \text{Cal}(X; D, s)) dc \right]. \quad (34)$$

If in particular s is calibrated with respect to D ,

$$\text{AUC}(s; D) = 1 - \frac{1}{2\pi(1 - \pi)} \cdot \mathbb{E}_{(X, Y) \sim D} \left[\int_0^1 \mu_S(c) \cdot \mathcal{A}_{\text{CS}(c)}(Y, s(X)) dc \right].$$

Equation 34 gives two interesting perspectives on the AUC. First, when it is possible to calibrate a scorer without losing information, the AUC can be thought of as implicitly calibrating a scorer before computing a particular risk. Second, by Shuford's representation (Equation 10), the risk computed is in fact identical to that for a proper loss, with the caveat that one considers a *score- and distribution-dependent* weight function $w(c) = \mu_C(c)$. Thus, the AUC is equivalent to the risk of a score- and distribution-dependent proper loss. (This is the finding of, for example, Hernández-Orallo et al. (2012, Theorem 34), which equates the AUC to the squared loss risk under a special case. We illustrate this equivalence empirically in Appendix K.) In particular, for a fixed distribution D , the AUC employs a different weighting for different scorers. Consequently, Hand (2009) calls the AUC an ‘‘incoherent’’ measure of classifier performance. Hand (2009, Section 6) proposes replacing this scorer dependent weight with one derived from the Beta family:

$$w(c; \alpha, \beta) = \frac{1}{B(\alpha, \beta)} \cdot c^{\alpha-1} \cdot (1 - c)^{\beta-1},$$

where $B(\alpha, \beta)$ is the normaliser for the Beta distribution. From Shuford's integral representation, it is apparent that the corresponding risk exactly corresponds to that of a proper composite loss. Indeed, the Beta family was proposed as a template for generating a proper loss in Buja et al. (2005, Section 11). For another perspective on this issue from the perspective of threshold selection, see Flach et al. (2011); Hernández-Orallo et al. (2012).

Another perspective on the ‘‘incoherence’’ can be gained from Proposition 13. Suppose we have (calibrated) scorers s_1, s_2 such that $\text{AUC}(s_1; D) > \text{AUC}(s_2; D)$. If it is further true that the ROC curve of s_1 dominates that of s_2 , then we know that s_1 will have lower risk than s_2 with respect to any proper composite risk, or equivalently any (distribution-independent) weighted combination of cost-sensitive risks. Thus, it is ‘‘coherent’’ to compare the two scorers based on their AUC in this case, because every other measure will result in s_1 being adjudged superior. When the ROC curves of the two scorers cross, however, the AUC is an incomplete measure of performance; with some proper losses, s_1 will be favoured over s_2 , and vice versa.

5.6.3 RATE-BASED REPRESENTATIONS

We proceed with some rate-based representations for the AUC. From a graphical perspective, these all derive from the fact that the AUC is the area under the ROC curve, and that this area is invariant to rotation of the horizontal and vertical axes i.e. instead of plotting the false positive versus true positive rate, we can equally plot the true negative versus false negative rate.

14. By definition $\int_0^1 \mu_S(c) dc = 1$, and so this equivalence can only be established to proper losses that satisfy $\int_0^1 w(c) dc < \infty$. This rules out, for example, logistic and exponential risk being equivalent to AUC.

Proposition 27 Given any $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and scorer $s : \mathcal{X} \rightarrow \mathbb{R}$ with differentiable ROC curve and invertible false- and true-positive rates,

$$\begin{aligned} \text{AUC}(s; D) &= \int_0^1 \text{TPR}(\text{FPR}^{-1}(\alpha)) d\alpha \\ &= \int_0^1 \text{TPR}(\text{TNR}^{-1}(\alpha)) d\alpha \\ &= \int_0^1 \text{TNR}(\text{TPR}^{-1}(\alpha)) d\alpha \\ &= \int_0^1 \text{TNR}(\text{FNR}^{-1}(\alpha)) d\alpha. \end{aligned}$$

Proof The first equation is simply Equation 27. The subsequent expressions follow from a few simple facts. First, if $f(x) = 1 - g(x)$ and f is invertible with inverse f^{-1} , then

$$g^{-1}(x) = f^{-1}(1 - x).$$

This implies that

$$\begin{aligned} \text{FPR}^{-1}(\alpha) &= \text{TNR}^{-1}(1 - \alpha) \\ \text{TPR}^{-1}(\alpha) &= \text{FNR}^{-1}(1 - \alpha). \end{aligned}$$

Second, for any f ,

$$\int_0^1 f(1 - x) dx = \int_0^1 f(x) dx.$$

Combined with the above, this implies

$$\begin{aligned} \int_0^1 \text{TPR}(\text{FPR}^{-1}(\alpha)) d\alpha &= \int_0^1 \text{TPR}(\text{TNR}^{-1}(\alpha)) d\alpha \\ \int_0^1 \text{TNR}(\text{TPR}^{-1}(\alpha)) d\alpha &= \int_0^1 \text{TNR}(\text{FNR}^{-1}(\alpha)) d\alpha. \end{aligned}$$

Third, for any f, g , integration by parts implies that

$$\int_a^b f'(x)g(x) dx = f(b)g(b) - f(a)g(a) - \int_a^b f(x)g'(x) dx. \quad (35)$$

Fourth, for any f, g such that $g(a) = 0, g(b) = 1$,

$$\int_0^1 f(g^{-1}(t)) dt = \int_a^b g'(x)f(x) dx. \quad (36)$$

This implies that:

$$\begin{aligned} \int_0^1 \text{TPR}(\text{TNR}^{-1}(\alpha)) d\alpha &= \int_{-\infty}^{\infty} \text{TNR}'(t) \cdot \text{TPR}(t) dt \\ &= - \int_{-\infty}^{\infty} \text{TPR}'(t) \cdot \text{TNR}(t) dt \text{ by Equation 35} \\ &= \int_0^1 \text{TNR}(\text{TPR}^{-1}(\alpha)) d\alpha \text{ by Equation 36.} \end{aligned}$$

Now recalling the definition of the AUC (Definition 14) as $\int_0^1 \text{TPR}(\text{FPR}^{-1}(\alpha)) d\alpha$, we see that we have proved the proposition. ■

From the above proof, we see that one can equivalently express the AUC as a weighted average of individual rates over a range of thresholds.

Corollary 28 Given any $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and scorer $s : \mathcal{X} \rightarrow \mathbb{R}$ with differentiable ROC curve and invertible false- and true-positive rates,

$$\begin{aligned} \text{AUC}(s; D) &= - \int_{-\infty}^{\infty} \text{FPR}'(t) \cdot \text{TPR}(t) dt \\ &= \int_{-\infty}^{\infty} \text{TNR}'(t) \cdot \text{TPR}(t) dt \\ &= - \int_{-\infty}^{\infty} \text{TPR}'(t) \cdot \text{TNR}(t) dt \\ &= \int_{-\infty}^{\infty} \text{FNR}'(t) \cdot \text{TNR}(t) dt. \end{aligned}$$

The weighting over thresholds as expressed above is not particularly intuitive, but recall from Equation 22 that the derivatives of the rates are the corresponding class-conditional densities of the scores. That means that we can interpret the above choice as equivalently drawing thresholds from these distributions. This is explored in the next section.

5.6.4 RANK REPRESENTATION

We now show how the AUC can be interpreted as the average of ranks of the instances, where the average is over thresholds drawn in accordance with the distribution of scores.

Corollary 29 Given any $D = \langle P, Q, \pi \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and scorer $s : \mathcal{X} \rightarrow \mathbb{R}$,

$$\text{AUC}(s; D) = \mathbb{E}_{\mathcal{X} \sim P} [\text{TNR}(s(\mathcal{X}))] \quad (37)$$

$$= \mathbb{E}_{\mathcal{X} \sim Q} [\text{TPR}(s(\mathcal{X}'))]. \quad (38)$$

Proof This follows immediately from Corollary 28 and the definition of the derivatives of the rates. Alternatively, by Proposition 17, and rewriting probabilities as expectations,

$$\begin{aligned} \text{AUC}(s; D) &= \mathbb{P}_{\mathcal{X} \sim P, \mathcal{X}' \sim Q} [s(\mathcal{X}) > s(\mathcal{X}')] + \frac{1}{2} \cdot \mathbb{P}_{\mathcal{X} \sim P, \mathcal{X}' \sim Q} [s(\mathcal{X}) = s(\mathcal{X}')] \\ &= \mathbb{E}_{\mathcal{X} \sim P} \left[\mathbb{E}_{\mathcal{X}' \sim Q} \left[\mathbb{1}[s(\mathcal{X}) > s(\mathcal{X}')] + \frac{1}{2} \mathbb{1}[s(\mathcal{X}) = s(\mathcal{X}')] \right] \right] \\ &= \mathbb{E}_{\mathcal{X} \sim P} [\text{TNR}(s(\mathcal{X}))]. \end{aligned}$$

Swapping the order of expectations in the other direction gives Equation 38. ■

On a finite training set, the empirical version of $\text{TNR}(s(\mathcal{X}))$ is related to the (normalised version of) what is typically called the “rank” of an instance $x \in \mathcal{X}$, where a higher rank is better. Specifically, the empirical $\text{TNR}(s(\mathcal{X}))$ counts the fraction of negative instances that x is scored higher than. In this sense, the AUC can be seen as measuring the average rank of the positive examples.

5.6.5 BALANCED ACCURACY REPRESENTATION

Our final representation of the AUC more explicitly relates it to a measure of classification performance: we show how to rewrite it as the average “balanced accuracy” across a range of thresholds, where the balanced accuracy is the average of the accuracies on the positive and negative class individually (Chan and Stolfo, 1998).

$$\text{BACC}(t; D, s) = \frac{\text{TPR}(t; D, s) + \text{TNR}(t; D, s)}{2}.$$

This suggests that the AUC explicitly considers good performance on both classes simultaneously, which indicates it is useful for problems with class imbalance (Ling and Li, 1998). For a related representation on a finite training set, see Flach et al. (2011, Theorem 4, 5), while for a different proof strategy, see Menon et al. (2015, Proposition 20).

Proposition 30 For any $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and scorer $s : \mathcal{X} \rightarrow \mathbb{R}$ with differentiable ROC curve and invertible false- and true-positive rates,

$$\begin{aligned} \text{AUC}(s; D) &= 2 \cdot \mathbb{E}_{\mathcal{X} \sim P} [\text{BACC}(s(X); D, s)] - \frac{1}{2} \\ &= 2 \cdot \mathbb{E}_{\mathcal{X}' \sim Q} [\text{BACC}(s(X'); D, s)] - \frac{1}{2}. \end{aligned} \quad (39)$$

Proof This is from Corollary 29 and the fact that

$$\begin{aligned} \mathbb{E}_{\mathcal{X} \sim P} [\text{TPR}(s(X); D, s)] &= \mathbb{E}_{S \sim P_S} [\text{TPR}(S; D, s)] \\ &= \int_{-\infty}^{\infty} -\text{TPR}'(t) \cdot \text{TPR}(t) dt \\ &= \int_0^1 u du \text{ with } u = \text{TPR}(t) \\ &= \frac{1}{2}. \end{aligned} \quad (40)$$

■ A similar argument yields the second identity.

The representation of Equation 39 can be contrasted with the risk for a proper composite loss. Using Shuford’s formula (Equation 10), for a proper composite loss ℓ with surjective and differentiable link Ψ , we have (Reid and Williamson, 2011, Proposition 20)

$$\begin{aligned} \mathbb{L}(s; D, \ell) &= \mathbb{E}_{(\mathcal{X}, \mathcal{Y}) \sim D} [\lambda(\mathcal{Y}, \Psi^{-1}(s(\mathcal{X})))] \\ &= \int_0^1 u(c) \cdot \mathbb{E}_{(\mathcal{X}, \mathcal{Y}) \sim D} [\lambda_{\text{CSI}(c)}(\mathcal{Y}, \Psi^{-1}(s(\mathcal{X})))] \\ &= \int_0^1 u(c) \cdot ((1 - \pi) \cdot c \cdot \text{FPR}(\Psi(c); D, s) + \pi \cdot (1 - c) \cdot \text{FNR}(\Psi(c); D, s)) dc \\ &= \int_{-\infty}^{\infty} \frac{u(\Psi^{-1}(t))}{\Psi(\Psi^{-1}(t))} \cdot ((1 - \pi) \cdot \Psi^{-1}(t) \cdot \text{FPR}(t; D, s) + \pi \cdot (1 - \Psi^{-1}(t)) \cdot \text{FNR}(t; D, s)) dt. \end{aligned}$$

Compared to Equation 39, we see that the proper composite risk has potentially asymmetric, but distribution independent, weights on the FPR and FNR. We also observe that the weights on the FPR and FNR are not equal, and vary with the thresholds. As per Hand’s representation (Equation 34), we may find that for a fixed distribution, there is a choice of link Ψ and weight u such that the two representations agree.

5.7 Relation to Existing Work

Lemma 22, which relates the AUC with the pairwise ranking (and hence pairwise classification) risk, is well known for the case of 0-1 loss (Kotowski et al., 2011; Agarwal, 2014). The extension to an arbitrary loss ℓ , while simple, is to our knowledge new. More generally, our definition of the ℓ -AUC as a generalisation of the standard AUC appears to be new, although in the special case where ℓ is a convex margin loss, the risk counterpart has been discussed (Cléménçon et al., 2008). The study of integral representations of the ℓ -AUC is to our knowledge new.

The representations in §5.6 are not new, although several of them only appear to have been stated for a finite training set.

6. Relating the Bayes Risk and Regret to Divergences

The previous sections studied the bipartite risk for an arbitrary scorer. In this section, we study the bipartite risk for the *Bayes-optimal* scorer, as well as the *regret* or *excess* risk for an arbitrary scorer. These help understand the inherent difficulty of a bipartite ranking problem, and formalise the sense in which “closeness” to the optimal scorer relates to the minimisation of the risk. Our characterisations rely on two classes of divergences between distributions, namely, the f - and Bregman-divergences. A review of the role of these divergences in characterising Bayes-risk and regret for classification is provided in Appendix E.

6.1 Warm-up: Bayes-Optimal Pairwise Ranking Risk and Regret

As pairwise ranking is readily shown to be equivalent to binary classification over pairs of instances (see §10.4), plugging in a pairwise ranking distribution $R \in \Delta_{\mathcal{X} \times \mathcal{X} \times \{\pm 1\}}$ into existing results for classification immediately implies the following.

Proposition 31 For any $R = \langle P_{\text{pair}}, Q_{\text{pair}}, \pi_{\text{pair}} \rangle \in \Delta_{\mathcal{X} \times \mathcal{X} \times \{\pm 1\}}$, convex $f : \mathbb{R}_+ \rightarrow \mathbb{R}$, and loss ℓ with conditional Bayes risk

$$(\forall \eta \in (0, 1)) L^*(\eta; \ell) = -\frac{1 - \eta}{1 - \pi_{\text{pair}}} \cdot f\left(\frac{1 - \pi_{\text{pair}}}{\pi_{\text{pair}}} \cdot \frac{\eta}{1 - \eta}\right),$$

the Bayes pairwise ranking risk can be written

$$\mathbb{L}^*(R, \ell) = L^*(\pi_{\text{pair}}; \ell) - \mathbb{J}_f(P_{\text{pair}}, Q_{\text{pair}}). \quad (41)$$

Conversely, Equation 41 holds for any $R = \langle P_{\text{pair}}, Q_{\text{pair}}, \pi_{\text{pair}} \rangle \in \Delta_{\mathcal{X} \times \mathcal{X} \times \{\pm 1\}}$, loss ℓ with concave conditional Bayes risk $L^* : [0, 1] \rightarrow \mathbb{R}_+$, and $f : \mathbb{R}_+ \rightarrow \mathbb{R}$ defined by

$$(\forall t \in \mathbb{R}) f(t) = L^*(\pi_{\text{pair}}; \ell) - (\pi_{\text{pair}} \cdot t + 1 - \pi_{\text{pair}}) \cdot L^*\left(\frac{\pi_{\text{pair}} \cdot t}{\pi_{\text{pair}} \cdot t + 1 - \pi_{\text{pair}}}; \ell\right).$$

An important example of the above is for the case of the 0-1 loss ℓ_{01} , where the corresponding f -divergence is the variational divergence (or total variation distance) $V(\cdot, \cdot)$, given by

$$V(P, Q) = \sup_{A \subseteq \mathcal{X}} 2 \cdot |P(A) - Q(A)| = \int_{\mathcal{X}} |p(x) - q(x)| dx.$$

Proposition 31 thus implies that the Bayes pairwise ranking risk is an affine transformation of $V(P_{\text{pair}}, Q_{\text{pair}})$. We similarly have a simple expression for the pairwise ranking regret with a proper loss.

Proposition 32 For any $R = \langle M_{\text{pair}}, \eta_{\text{pair}} \rangle \in \Delta_{\mathcal{X} \times \mathcal{X} \times \{\pm 1\}}$, $\ell \in \mathcal{L}_{\text{SPC}}(\Psi)$, and pair-scorer $s_{\text{pair}} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$,

$$\text{regret}(s_{\text{pair}}; R, \ell) = \mathbb{B}_{-f}(\eta_{\text{pair}}, \Psi^{-1} \circ s_{\text{pair}})$$

where in an abuse of notation $L^* = L^*(\cdot; \ell)$.

We now see how these results can be translated to the bipartite ranking setting.

6.2 Bayes-Optimal Bipartite Risk as an f -Divergence

For bipartite ranking, we can hope to exploit the connection between bipartite and pairwise ranking (Lemma 2), and derive analogues of the above for the distribution D_{BR} . A subtlety is that, as noted in Proposition 3, it is *not* necessarily true that $\mathbb{L}_{\text{BR}}^*(D, \ell) = \mathbb{L}^*(D_{\text{BR}}, \ell)$. Therefore, to translate the previous results, we need an additional condition ensuring this holds, which simply that $\ell \in \mathcal{L}_{\text{Decomp}}$:

Proposition 33 For any $D = \langle P, Q, \pi \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$, convex $f : \mathbb{R}_+ \rightarrow \mathbb{R}$, and loss $\ell \in \mathcal{L}_{\text{Decomp}}$ with conditional Bayes risk

$$(\forall \eta \in (0, 1)) L^*(\eta; \ell) = -2 \cdot (1 - \eta) \cdot f\left(\frac{\eta}{1 - \eta}\right),$$

the Bayes-risk can be written

$$\mathbb{L}_{\text{BR}}^*(D, \ell) = L^*(1/2; \ell) - \mathbb{J}_f(P \times Q, Q \times P). \quad (42)$$

Conversely, Equation 42 holds for any $D = \langle P, Q, \pi \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$, loss $\ell \in \mathcal{L}_{\text{Decomp}}$ with concave conditional Bayes risk $L^*(\cdot; \ell) : [0, 1] \rightarrow \mathbb{R}_+$, and $f : \mathbb{R}_+ \rightarrow \mathbb{R}$ defined by

$$(\forall t \in \mathbb{R}) f(t) \doteq L^*(1/2; \ell) - \frac{1+t}{2} \cdot L^*\left(\frac{t}{1+t}; \ell\right).$$

Proof By Proposition 3, the assumption on ℓ inducing a decomposable Bayes-optimal pair-scorer for D_{BR} implies we can equate $\mathbb{L}_{\text{BR}}^*(D, \ell)$ and $\mathbb{L}^*(D_{\text{BR}}, \ell)$. We then apply Proposition 31 to D_{BR} , so that

$$\begin{aligned} \mathbb{L}_{\text{BR}}^*(D, \ell) &= \mathbb{L}^*(D_{\text{BR}}, \ell) \text{ by Proposition 3} \\ &= L^*(P_{\text{pair}}; \ell) - \mathbb{J}_f(P_{\text{pair}}, Q_{\text{pair}}) \text{ by Proposition 31} \\ &= L^*(1/2; \ell) - \mathbb{J}_f(P \times Q, Q \times P) \text{ by Appendix B.} \end{aligned}$$

The other direction follows similarly. \blacksquare

As we shall see in Proposition 44, the requirement that $\ell \in \mathcal{L}_{\text{Decomp}}$ for a proper composite loss is equivalent to a condition on its link function. Importantly, there is *no* restriction on the underlying proper-loss itself. Therefore, the above holds for a large class of losses. One such example is the 0-1 loss $\ell = \ell_{01}$, where we have following relationship between the Bayes-optimal AUC and the variational divergence between the product measures $P \times Q$ and $Q \times P$.

Corollary 34 Given any $D = \langle P, Q, \pi \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$, the Bayes-optimal bipartite ranking risk is related to the variational divergence between the product distributions $P \times Q$ and $Q \times P$ via:

$$\mathbb{L}^*(D_{\text{BR}}, \ell_{01}) = \frac{1}{2} - \frac{1}{4} \cdot V(P \times Q, Q \times P).$$

Proof This follows from Proposition 33 and the fact that for ℓ_{01} ,

$$(\forall \eta \in [0, 1]) L^*(\eta; \ell_{01}) = \eta \wedge (1 - \eta) = \frac{1}{2} - \left| \eta - \frac{1}{2} \right|,$$

with $L^*(1/2; \ell_{01}) = 1/2$. It is easy to check that this corresponds to $f(t) = |t - 1/4 + (1 - t)/4|$, which is a scaled version of the convex generator for the variational divergence. \blacksquare

By Corollary 23 and Equation 48, Corollary 34 is equivalent to a result of Torgersen (1991, pg. 582),

$$\text{AUC}^*(D) = \frac{1}{2} + \frac{1}{4} \cdot V(P \times Q, Q \times P).$$

This may be further manipulated to explicitly express the Bayes-optimal AUC in terms of the concentration of the values of η (Clemençon et al., 2008),

$$\text{AUC}^*(D) = \frac{1}{2} + \frac{1}{4\pi(1 - \pi)} \cdot \mathbb{E}_{\mathcal{X} \sim M, \mathcal{X}' \sim M} [|\eta(\mathcal{X}) - \eta(\mathcal{X}')|].$$

This expression may be further related to the earth mover's distance (or L_1 -Wasserstein metric) between the class-conditional distribution of scores (Clemençon et al., 2009).

6.3 Bipartite Ranking Regret as a Generative Bregman Divergence

The bipartite ranking regret for proper composite losses may similarly be re-expressed by exploiting the reduction to classification on pairs. We again need to restrict ourselves to those proper composite losses that induce a decomposable Bayes-optimal pair-scorer.

Proposition 35 Pick any $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ with derived pairwise ranking distribution $D_{\text{BR}} = \langle M_{\text{pair}}, \eta_{\text{pair}} \rangle$ and any $\ell \in \mathcal{L}_{\text{SPC}}(\mathcal{Y}) \cap \mathcal{L}_{\text{Decomp}}$. Then, for any scorer $s : \mathcal{X} \rightarrow \mathbb{R}$,

$$\text{regret}_{\text{BR}}(s; D, \ell) = \mathbb{E}_{L^*}(\eta_{\text{pair}}, \mathbf{y}^{-1} \circ \text{Diff}(s))$$

where in an abuse of notation $L^* \doteq L^*(\cdot; \ell)$.

Proof By definition of the bipartite regret (Equation 18),

$$\begin{aligned} \text{regret}_{\text{BR}}(s; D, \ell) &= \mathbb{L}_{\text{BR}}(s; D, \ell) - \mathbb{L}_{\text{BR}}^*(D, \ell) \\ &= \mathbb{L}(\text{Diff}(s); D_{\text{BR}}; \ell) - \mathbb{L}_{\text{BR}}^*(D, \ell) \\ &= \text{regret}(\text{Diff}(s); D_{\text{BR}}, \ell) + \mathbb{L}^*(D_{\text{BR}}, \ell) - \mathbb{L}_{\text{BR}}^*(D, \ell) \\ &= \mathbb{E}_{L^*}(\eta_{\text{pair}}, \mathbf{y}^{-1} \circ \text{Diff}(s)) + \mathbb{L}^*(D_{\text{BR}}, \ell) - \mathbb{L}_{\text{BR}}^*(D, \ell), \end{aligned}$$

where the last line is by the standard expression for the regret with respect to a proper composite loss (Proposition 78). We thus need $\mathbb{L}^*(D_{\text{BR}}, \ell) = \mathbb{L}_{\text{BR}}^*(D, \ell)$, which by Proposition 3 is true iff $\ell \in \mathcal{L}_{\text{Decomp}}$. \blacksquare

In the case of $\ell = \ell_{01}$, the regret can be seen to measure the concentration of η in the region where the candidate scorer s disagrees with η , as is well known.

Corollary 36 (Clemençon et al., 2008; Agarwal, 2014, Theorem 11) For any $D = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and scorer $s : \mathcal{X} \rightarrow \mathbb{R}$,

$$\text{regret}_{\text{BR}}(s; D, \ell_{01}) = \mathbb{E}_{\mathcal{X} \sim M, \mathcal{X}' \sim M} [|\eta(\mathcal{X}) - \eta(\mathcal{X}')| \cdot \mathbb{I}(s, \eta; \mathcal{X}, \mathcal{X}')]]$$

where

$$\mathbb{I}(s, \eta; \mathcal{X}, \mathcal{X}') = \mathbb{I}(\eta(\mathcal{X}) - \eta(\mathcal{X}') \cdot (s(\mathcal{X}) - s(\mathcal{X}')) < 0] + \frac{1}{2} \cdot \mathbb{I}[\eta(\mathcal{X}) = \eta(\mathcal{X}')].$$

6.4 Relation to Existing Work

As noted above, the connection between Bayes risks and f -divergences in classification is well known (Osercher and Vajda, 1993). For the bipartite ranking problem, the connection between the AUC and variational divergence has been made by (Torgersen, 1991; Reid and Williamson, 2011). The extension to the case of general ℓ -bipartite risk (Proposition 33) is simple in hindsight, as the variational divergence is well known to correspond to the use of 0-1 loss; however, to our knowledge, the extension is novel.

7. Bayes-Optimal Scorers for Bipartite Ranking

The previous section studied the risk associated with a Bayes-optimal scorer for bipartite ranking. We now characterise the set of Bayes-optimal scorers itself. Knowledge of the optimal scorers gives further insight into the problem, and helps relate it to the more familiar tasks of binary classification and class-probability estimation. As we shall see in the next section, this will also help in establishing the statistical consistency of the minimisation of surrogate losses on pairs for the task of AUC maximisation.

Before proceeding, we first recall the Bayes-optimal scorers for the latter problems given some $D = \langle P, Q, \pi \rangle = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \mathcal{X}(\pm 1)}$. Table 8 summarises the findings of this section.

Loss type	Bayes-optimal (pair-)scorers	Reference
Classification-calibrated	$S^*(D, \ell) \subseteq \left\{ s : \mathcal{X} \rightarrow \mathbb{R} : \begin{array}{l} \eta(x) \neq 1/2 \implies \\ \text{sign}(s(x)) = \text{sign}(2\eta(x) - 1) \end{array} \right\}$	Equation 43
	$S^*(D_{\text{BR}}, \ell) \subseteq \left\{ s_{\text{pair}} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R} : \begin{array}{l} \eta(x) \neq \eta(x') \implies \\ \text{sign}(s_{\text{pair}}(x, x')) = \\ \text{sign}(\eta(x) - \eta(x')) \end{array} \right\}$	Equation 49
	$\{\phi \circ \eta\} \subseteq S_{\text{BR}}^*(D, \ell_{01}) = \{s : \mathcal{X} \rightarrow \mathbb{R} : \eta = \phi \circ s\}$	Proposition 42
Proper composite with link Ψ	$\{\Psi \circ \eta\} \subseteq S^*(D, \ell)$	Equation 44
	$\{\Psi \circ \sigma \circ \text{Diff}(\sigma^{-1} \circ \eta)\} \subseteq S^*(D_{\text{BR}}, \ell)$	Equation 50
	$S_{\text{BR}}^*(D, \ell) = \{\Psi \circ \eta + b : b \in \mathbb{R}\}$ for $\Psi \in \Sigma_{\text{sig}}$	Corollary 45

Table 8: Bayes-optimal scorers and pair-scorers for various classification and bipartite ranking risks.

7.1 Binary Classification

Consider a binary classification problem with distribution $D = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \mathcal{X}(\pm 1)}$. If the loss ℓ is classification-calibrated (Equation 7), every Bayes-optimal scorer must have the same sign as $\eta(x) - 1/2$, with the prediction for $\eta(x) = 1/2$ being irrelevant (Bartlett et al., 2006):

$$S^*(D, \ell) \subseteq \{s : \mathcal{X} \rightarrow \mathbb{R} : \eta(x) \neq 1/2 \implies \text{sign}(s(x)) = \text{sign}(2\eta(x) - 1)\}. \quad (43)$$

When ℓ is the 0-1 loss, the above is an equality (Devroye et al., 1996). Thus, for ℓ_{01} , what is of interest is determining whether or not each instance has a greater than random chance of being labelled positive.

When ℓ is a proper composite loss with link Ψ , from the definition of properness (Equation 9) we can specify one minimiser of the conditional risk, which applied pointwise gives:

$$\{\Psi \circ \eta\} \subseteq S^*(D, \ell). \quad (44)$$

This is an equality if and only if ℓ is strictly proper composite. Thus, a strictly proper composite loss requires precise information about η , unlike ℓ_{01} . Observe that $\Psi \circ \eta$ may be trivially transformed to give an optimal scorer for ℓ_{01} ; thus, exactly solving class-probability estimation also solves binary classification. For an approximate solution, one can bound the excess ℓ_{01} error via a surrogate regret bound (Reid and Williamson, 2009).

7.2 Pairwise Ranking

Recall from §3.4 that pairwise ranking is identical to binary classification over pairs of instances. Thus, in the pairwise ranking setting with distribution $R = \langle M_{\text{pair}}, \eta_{\text{pair}} \rangle \in \Delta_{\mathcal{X} \times \mathcal{X}(\pm 1)}$, the above results can be

translated. For a classification-calibrated loss, the Bayes-optimal pair-scorers must have the same sign as $\eta_{\text{pair}}(x, x') - 1/2$, with the prediction for $\eta_{\text{pair}}(x, x') = 1/2$ being irrelevant:

$$S^*(R, \ell) \subseteq \{s_{\text{pair}} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R} : \eta_{\text{pair}}(x, x') \neq 1/2 \implies \text{sign}(s_{\text{pair}}(x, x')) = \text{sign}(2\eta_{\text{pair}}(x, x') - 1)\}. \quad (45)$$

When ℓ is the 0-1 loss, the above is an equality.

When ℓ is a proper composite loss with link Ψ , by definition we can specify one Bayes-optimal pair-scorer, though there may be others:

$$\{\Psi \circ \eta_{\text{pair}}\} \subseteq S^*(R, \ell). \quad (46)$$

This is an equality if and only if ℓ is strictly proper composite.

7.3 Bipartite Ranking

The relationship between bipartite and pairwise ranking (Lemma 2) suggests we can simply compute the Bayes-optimal scorers for pairwise ranking with D_{BR} . Specifically, let $D_{\text{BR}} = \langle M_{\text{pair}}, \eta_{\text{pair}} \rangle$. To determine $S^*(D_{\text{BR}}, \ell)$, we need the following elementary but important property of η_{pair} .

Lemma 37 For any $D = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \mathcal{X}(\pm 1)}$, D_{BR} has observation-conditional distribution given by

$$\eta_{\text{pair}} = \sigma \circ \text{Diff}(\sigma^{-1} \circ \eta), \quad (47)$$

where $\sigma(\cdot)$ denotes the sigmoid function (Equation 2).

Proof Suppose $(X, X', Z) \sim D_{\text{BR}}$. Recall $\mathbb{P}[Z = +1] = \frac{1}{2}$. Then,

$$\begin{aligned} (\forall x, x' \in \mathcal{X}) \eta_{\text{pair}}(x, x') &= \mathbb{P}[Z = +1 | X = x, X' = x'] \\ &= \frac{\mathbb{P}[X = x, X' = x' | Z = +1] \cdot \mathbb{P}[Z = +1]}{\mathbb{P}[X = x, X' = x']} \\ &= \frac{\mathbb{P}[X = x | Z = +1] \cdot \mathbb{P}[X' = x' | Z = +1] \cdot \mathbb{P}[Z = +1]}{\mathbb{P}[X = x, X' = x']} \\ &= \frac{\mathbb{P}[X = x | Z = +1] \cdot \mathbb{P}[X' = x' | Z = +1] + \mathbb{P}[X = x | Z = -1] \cdot \mathbb{P}[X' = x' | Z = -1]}{1 + \frac{\mathbb{P}[X = x | Z = -1] \cdot \mathbb{P}[X' = x' | Z = -1]}{\mathbb{P}[X = x | Z = +1] \cdot \mathbb{P}[X' = x' | Z = +1]}} \\ &= \sigma(\sigma^{-1}(\eta(x)) - \sigma^{-1}(\eta(x'))). \end{aligned}$$

The last identity follows because

$$\sigma^{-1}(\eta(x)) = \log \frac{\pi}{1 - \pi} + \log \frac{\mathbb{P}[X = x | Z = +1]}{\mathbb{P}[X = x' | Z = -1]}.$$

From Lemma 37, the Bayes-optimal scorers for a classification-calibrated loss are immediate. ■

Lemma 38 For any $D = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \mathcal{X}(\pm 1)}$, and any classification-calibrated ℓ ,

$$S^*(D_{\text{BR}}, \ell) \subseteq \{s_{\text{pair}} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R} : \eta(x) \neq \eta(x') \implies \text{sign}(s_{\text{pair}}(x, x')) = \text{sign}(\eta(x) - \eta(x'))\}.$$

When ℓ is the 0-1 loss, the above is an equality.

Proof For a classification calibrated loss ℓ , for every $x, x' \in \mathcal{X}$ such that $\eta(x) \neq \eta(x')$, any Bayes-optimal pair-scorer $s_{\text{Pair}}^* \in \mathcal{S}^*(D_{\text{BR}}, \ell)$ must satisfy

$$\begin{aligned} \text{sign}(s_{\text{Pair}}^*(x, x')) &= \text{sign}(2\eta_{\text{Pair}}(x, x') - 1) \\ &= \text{sign}(2\sigma(\sigma^{-1}(\eta(x)) - \sigma^{-1}(\eta(x')) - 1)) \text{ by Lemma 37} \\ &= \text{sign}(\sigma^{-1}(\eta(x)) - \sigma^{-1}(\eta(x'))) \\ &= \text{sign}(\eta(x) - \eta(x')). \end{aligned} \quad (48)$$

When $\eta(x) = \eta(x')$, we can pick any $s_{\text{Pair}}^*(x, x') \in \mathbb{R}$. Thus, for all $D = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and classification-calibrated losses ℓ ,

$$\mathcal{S}^*(D_{\text{BR}}, \ell) \subseteq \{s_{\text{Pair}} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R} : \eta(x) \neq \eta(x') \implies \text{sign}(s_{\text{Pair}}(x, x')) = \text{sign}(\eta(x) - \eta(x'))\}. \quad (49)$$

For ℓ_{01} , every pair-scorer satisfying the above is optimal, thus yielding an equality. ■

When ℓ is a proper composite loss with link Ψ , by definition we can specify one Bayes-optimal pair-scorer, though there may be others:

$$\{\Psi \circ \eta_{\text{Pair}}\} = \{\Psi \circ \sigma \circ \text{Diff}(\sigma^{-1} \circ \eta)\} \subseteq \mathcal{S}^*(D_{\text{BR}}, \ell). \quad (50)$$

This is an equality if and only if ℓ is strictly proper composite.

Having computed the optimal pair-scorers, when attempting to translate the results to bipartite ranking, we immediately face a challenge, as

$$\begin{aligned} \text{Argmin}_{s : \mathcal{X} \rightarrow \mathbb{R}} \mathbb{L}_{\text{BR}}(s; D, \ell) &= \text{Argmin}_{s : \mathcal{X} \rightarrow \mathbb{R}} \mathbb{L}(\text{Diff}(s); D_{\text{BR}}, \ell) \\ &= \text{Argmin}_{s_{\text{Pair}} \in \mathcal{S}_{\text{Decomp}}^*} \mathbb{L}(s_{\text{Pair}}; D_{\text{BR}}, \ell). \end{aligned}$$

That is, finding the set of scorers s that minimise $\mathbb{L}_{\text{BR}}(\text{Diff}(s))$ is equivalent to finding the set of pair-scorers s_{Pair} that minimise $\mathbb{L}(s_{\text{Pair}}; D_{\text{BR}}, \ell)$, *subject to* the pair-scorers being decomposable. Formally, in the notation of Equation 20, we need every such optimal $s_{\text{Pair}}^* \in \mathcal{L}_{\text{Decomp}}^*$. While the latter constraint seems innocuous, it means we need to reason about a minimiser in a *restricted* function class. Thus, in general, it is no longer possible to simply study the conditional risk and make a pointwise analysis.

Of course, we can easily make progress in the special case where the optimal pair-scorer is in fact decomposable. In this case, we can effectively ignore the restricted function class, because the optimal pair-scorer must be the difference of the optimal univariate scorer. The following makes this precise.

Proposition 39 *Given any $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and loss ℓ ,*

$$\ell \in \mathcal{L}_{\text{Decomp}} \iff \mathcal{S}^*(D_{\text{BR}}, \ell) \cap \mathcal{S}_{\text{Decomp}}^* = \text{Diff}(\mathcal{S}_{\text{BR}}^*(D, \ell)).$$

Proof The (\implies) direction is immediate, since $\mathcal{S}_{\text{BR}}^*(D, \ell) \neq \emptyset$ and thus $\text{Diff}(\mathcal{S}_{\text{BR}}^*(D, \ell)) \neq \emptyset$. We show the (\impliedby) direction.

(C). Pick any $s_{\text{Pair}}^* \in \mathcal{S}^*(D_{\text{BR}}, \ell) \cap \mathcal{S}_{\text{Decomp}}^*$. Then $s_{\text{Pair}}^* = \text{Diff}(s)$ for some $s : \mathcal{X} \rightarrow \mathbb{R}$. By optimality of s_{Pair}^*

$$(\forall s : \mathcal{X} \rightarrow \mathbb{R}) \mathbb{L}_{\text{BR}}(s) = \mathbb{L}(s_{\text{Pair}}^*; D_{\text{BR}}, \ell) \leq \mathbb{L}(\text{Diff}(s); D_{\text{BR}}, \ell) = \mathbb{L}_{\text{BR}}(s).$$

Thus $s \in \mathcal{S}_{\text{BR}}^*(D, \ell)$, and so $s_{\text{Pair}}^* \in \text{Diff}(\mathcal{S}_{\text{BR}}^*(D, \ell))$.

(D). Pick any $s^* \in \mathcal{S}_{\text{BR}}^*(D, \ell)$, and let $s_{\text{Pair}}^* = \text{Diff}(s^*)$. Then, by definition,

$$s_{\text{Pair}}^* \in \text{Argmin}_{s_{\text{Pair}} \in \mathcal{S}_{\text{Decomp}}^*} \mathbb{L}(s_{\text{Pair}}; D_{\text{BR}}, \ell).$$

This is a constrained optimisation problem. When $\ell \in \mathcal{L}_{\text{Decomp}}$, there is at least one solution to the *unconstrained* optimisation that lies in $\mathcal{S}_{\text{Decomp}}^*$; call it r_{Pair} . Clearly r_{Pair} is a feasible solution for the constrained problem above. Thus, it must have an identical risk to s_{Pair}^* . But then s_{Pair}^* is a solution to the unconstrained problem as well, and so $s_{\text{Pair}}^* \in \mathcal{S}^*(D_{\text{BR}}, \ell) \cap \mathcal{S}_{\text{Decomp}}^*$. ■

The result simplifies somewhat when *every* Bayes-optimal pair-scorer is decomposable, which occurs when there is a unique optimal pair-scorer.

Corollary 40 *Given any $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and loss ℓ ,*

$$\mathcal{S}^*(D_{\text{BR}}, \ell) \subseteq \mathcal{S}_{\text{Decomp}}^* \iff \mathcal{S}^*(D_{\text{BR}}, \ell) = \text{Diff}(\mathcal{S}_{\text{BR}}^*(D, \ell)).$$

Proof (\implies) follows by Proposition 39, and (\impliedby) follows by definition of decomposability. ■

Simply put, the decomposable Bayes-optimal pair-scorers are exactly the Bayes-optimal scorers passed through $\text{Diff}(\cdot)$. Thus, if we can show that $\mathcal{S}_{\text{BR}}^*(D, \ell) \cap \mathcal{S}_{\text{Decomp}}^* \neq \emptyset$ for a loss ℓ , we automatically deduce the Bayes-optimal scorer from the results of the previous section. We determine when this condition holds below.

7.4 Bipartite ranking: Decomposable Case

We study the Bayes-optimal scorers for losses that induce a decomposable Bayes-optimal pair-scorer. We begin with the case ℓ_{01} .

7.4.1 OPTIMAL UNIVARIATE SCORER FOR 0-1 LOSS

For ℓ_{01} , it is not hard to see that our earlier results imply that D_{BR} has at least one decomposable Bayes-optimal pair-scorer.

Lemma 41 *Given any $D = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$,*

$$\mathcal{S}^*(D_{\text{BR}}, \ell_{01}) \cap \mathcal{S}_{\text{Decomp}}^* \neq \emptyset.$$

Proof By Equation 49, we see that $\{\text{Diff}(\eta)\} \subseteq \mathcal{S}^*(D_{\text{BR}}, \ell_{01}) \cap \mathcal{S}_{\text{Decomp}}^*$. That is, ℓ_{01} induces at least one decomposable Bayes-optimal pair-scorer in the pairwise ranking risk. ■

We can now show that the optimal scorers for the 0-1 bipartite risk are those that preserve the ordering of the class-probability η , which includes all strictly monotone transformations of η .

Proposition 42 *Given any $D = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$,*

$$\mathcal{S}_{\text{BR}}^*(D, \ell_{01}) = \{s : \mathcal{X} \rightarrow \mathbb{R} : \eta = \phi \circ s \text{ for } \phi : \mathbb{R} \rightarrow [0, 1] \text{ non-decreasing}\}.$$

Proof Let $\mathcal{A} = \mathcal{S}^*(D_{\text{BR}}, \ell_{01}) \cap \mathcal{S}_{\text{Decomp}}^*$. Since \mathcal{A} is nonempty by Proposition 41, $\mathcal{A} = \text{Diff}(\mathcal{S}_{\text{BR}}^*(D, \ell_{01}))$ by Proposition 39. Equivalently, by Lemma 38,

$$\begin{aligned} \mathcal{A} &= \{s_{\text{Pair}} \in \mathcal{S}_{\text{Decomp}}^* : \eta(x) \neq \eta(x') \implies \text{sign}(s_{\text{Pair}}(x, x')) = \text{sign}(\eta(x) - \eta(x'))\} \\ &= \text{Diff}(\{s : \mathcal{X} \rightarrow \mathbb{R} : \eta(x) \neq \eta(x') \implies \text{sign}(s(x) - s(x')) = \text{sign}(\eta(x) - \eta(x'))\}) \\ &= \text{Diff}(\{s : \mathcal{X} \rightarrow \mathbb{R} : \eta = \phi \circ s \text{ for non-decreasing } \phi\}) \text{ by Lemma 72.} \end{aligned}$$

We thus have equality of the differences of the sets of interest. But for any sets of scorers $\mathcal{S}_1, \mathcal{S}_2$, $\text{Diff}(\mathcal{S}_1) = \text{Diff}(\mathcal{S}_2) \implies (\forall s_1 \in \mathcal{S}_1)(\exists s_2 \in \mathcal{S}_2, c \in \mathbb{R}) s_1 = s_2 + c$, i.e. the scorers in the two sets must be related by a

linear translation. But if for a scorer s we have $\eta = \phi \circ s$ for some monotone ϕ , then it must also be true that $\eta = \tilde{\phi} \circ (s + c)$ where $\tilde{\phi} : x \mapsto \phi(x - c)$ is also monotone. Thus, the result follows. ■

The transform ϕ in Proposition 42 is not required to be strictly monotone increasing since if $\eta(x) = \eta(x')$ for some $x \neq x' \in \mathcal{X}$, it is allowed for $s(x) \neq s(x')$. (In the extreme case where $\eta(x) \equiv c$ for every x , then every scorer will trivially be Bayes-optimal.) Nonetheless, an immediate corollary is that any strictly monotone increasing transform of η is necessarily an optimal univariate scorer.¹⁵

Corollary 43 Given any $D = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and any strictly monotone increasing $\phi : [0, 1] \rightarrow \mathbb{R}$,

$$\phi \circ \eta \in S_{\text{BR}}^*(D, \ell_{01}).$$

We see that like class-probability estimation, bipartite ranking with ℓ_{01} aims to find a transformation of η . Unlike class-probability estimation, one is satisfied with *any* strictly monotone transformation, not necessarily one specified by the loss itself. Loosely, then, bipartite ranking is less “strict” than class-probability estimation. (See also §10.)

7.4.2. OPTIMAL UNIVARIATE SCORER FOR STRICTLY PROPER COMPOSITE LOSSES

We now proceed to the case where ℓ is a strictly proper composite loss. To apply Corollary 40, we characterise the subset of proper composite losses for which there exists a decomposable pair-scorer. This shall turn out to rely on the following set of inverse link functions from the sigmoid family,

$$\Sigma_{\text{sig}} := \left\{ \Psi^{-1} : \mathbb{R} \rightarrow [0, 1] \mid (\exists a \in \mathbb{R} \setminus \{0\}) (\forall v \in \mathbb{R}) \Psi^{-1}(v) = \frac{1}{1 + e^{-av}} \right\}. \quad (51)$$

Proposition 44 (Decomposability of Bayes-optimal bipartite pair-scorer.) Given any $\ell \in \mathcal{L}_{\text{SPC}}(\Psi)$ with Ψ differentiable,

$$(\forall D \in \Delta_{\mathcal{X} \times \{\pm 1\}}) S^*(D_{\text{BR}}, \ell) \subseteq S_{\text{Decomp}} \iff \Psi^{-1} \in \Sigma_{\text{sig}}.$$

Proof (\Leftarrow) Let the link function of ℓ have the specified form, so that $\Psi(v) = \frac{1}{a} \log \frac{v}{1-v} = \frac{1}{a} \sigma^{-1}(v)$, and so $(\Psi \circ \sigma)(v) = \frac{v}{a}$. From Equation 50, the¹⁶ Bayes-optimal pair-scorer is

$$\begin{aligned} s_{\text{Pair}}^* &= \frac{1}{a} \cdot \text{Diff}(\sigma^{-1} \circ \eta) \\ &= \text{Diff} \left(\left(\frac{1}{a} \cdot \sigma^{-1} \right) \circ \eta \right) \\ &\in S_{\text{Decomp}}. \end{aligned}$$

Thus $s_{\text{Pair}}^* \in S^*(D_{\text{BR}}, \ell) \cap S_{\text{Decomp}}$.

(\Rightarrow) The proof here uses a similar idea to Uematsu and Lee (2012, Theorem 7). If $\ell \in \mathcal{L}_{\text{Decomp}}$

$$\Psi \circ \sigma \circ \text{Diff}(\sigma^{-1} \circ \eta) \in S_{\text{Decomp}}.$$

We wish to determine the nature of Ψ that permits this to hold. Let $f = \Psi \circ \sigma \circ \log$, so that the above becomes

$$(\forall x, x' \in \mathcal{X}) f \left(\frac{e^{\sigma^{-1}(\eta(x))}}{e^{\sigma^{-1}(\eta(x'))}} \right) = g(x) - g(x')$$

15. Combined with the connection between the 0-1 optimal scorer for the bipartite ranking risk and AUC (Corollary 23), this constitutes an alternate proof of Corollary 16, without an appeal to the Neyman-Pearson lemma.

16. For a *non-strict* proper composite loss, the following argument holds, but only for one possible optimal pair-scorer. Thus, it may not be true that *all* Bayes-optimal pair-scorers are decomposable; however, for the choice of link function above, we can guarantee that there is *at least* one that is. Nonetheless, for a *strictly* proper composite loss, there is a unique Bayes-optimal pair-scorer. Thus, the above result characterises when this pair-scorer is decomposable.

for some $g : \mathcal{X} \rightarrow \mathbb{R}$. Now note that

$$\begin{aligned} (\forall x, x', x'' \in \mathcal{X}) f \left(\frac{e^{\sigma^{-1}(\eta(x))}}{e^{\sigma^{-1}(\eta(x'))}} \right) &= g(x) - g(x'') + g(x'') - g(x') \\ &= f \left(\frac{e^{\sigma^{-1}(\eta(x))}}{e^{\sigma^{-1}(\eta(x'))}} \right) + f \left(\frac{e^{\sigma^{-1}(\eta(x''))}}{e^{\sigma^{-1}(\eta(x'))}} \right). \end{aligned}$$

We require this to hold for any D , and thus for any η . Therefore, equivalently, we have

$$(\forall a, b \in \mathbb{R}_+) f(a \cdot b) = f(a) + f(b).$$

Note that f is continuous by assumed differentiability of Ψ . Thus the only solution to the equation is $f(z) = \frac{1}{a} \cdot \log z$ for some $a \in \mathbb{R}$ (Kannappan, 2009, Corollary 1.43), or equivalently that $\Psi^{-1}(v) = \sigma(a \cdot v) = \frac{1}{1 + e^{-av}}$. Note that the case $a = 0$ is ruled out by assumed invertibility of Ψ , and thus equivalently of f . ■

We emphasise that the class of proper composite losses satisfying the above condition is “large” in the following sense: one may take *any* strictly proper loss and compose it with any member of the given link family. Two specific implications are noteworthy. First, the loss ℓ need not be symmetric; Appendix G has an empirical illustration of this fact. Second, the loss ℓ may be non-convex; nonetheless, we can easily determine the optimal scorers for all such losses, as below.

Corollary 45 Given any $D = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and $\ell \in \mathcal{L}_{\text{SPC}}(\Psi)$ with inverse link function $\Psi^{-1} \in \Sigma_{\text{sig}}$,

$$S_{\text{BR}}^*(D, \ell) = \{\Psi \circ \eta + b : b \in \mathbb{R}\}.$$

Consequently, when Ψ is monotone increasing (viz. when $a \in \mathbb{R}_+$ in Equation 51),

$$S_{\text{BR}}^*(D, \ell) \subseteq S_{\text{BR}}^*(D, \ell_{01}).$$

Proof By Proposition 44 and Corollary 40,

$$\text{Diff}(S_{\text{BR}}^*(D, \ell)) = S_{\text{BR}}^*(D, \ell).$$

Further, by Equation 50, and letting $\Psi^{-1} : v \mapsto \sigma(a \cdot v)$,

$$S_{\text{BR}}^*(D, \ell) = \text{Diff} \left(\frac{1}{a} \cdot \sigma^{-1} \circ \eta \right) = \text{Diff}(\Psi \circ \eta).$$

The result follows because

$$\text{Diff}(f) = \text{Diff}(g) \iff (\exists b \in \mathbb{R}) f = g + b.$$

The admissible family of links Σ_{sig} can be easily checked to contain those employed for the logistic and exponential losses, and thus we can deduce the decomposability of the Bayes-optimal scorers for these losses.

Corollary 46 For the strictly proper composite logistic and exponential losses,

$$\begin{aligned} \ell_{\log}(y, z) &= \log(1 + e^{-yz}) \\ \ell_{\exp}(y, z) &= e^{-yz}, \end{aligned}$$

with inverse link functions $\Psi^{-1}(v) = \frac{1}{1+e^{-2v}}$ and $\Psi^{-1}(v) = \frac{1}{1+e^{-2v}}$ respectively, the Bayes-optimal pair-scorer is decomposable for any $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ i.e.

$$S^*(D_{\text{BR}}, \ell) \subseteq S_{\text{Decomp}}.$$

While Proposition 44 follows easily from the proper loss machinery, the requirement on the link function is *a priori* non-obvious. What is special about link functions that are scaled versions of the sigmoid? The answer is simply that the score η_{pair} , inherently involves a sigmoid link function (Lemma 3.7). This form of η_{pair} in turn can be understood via utility representations for binary relations on sets, as we discuss in §12.

7.4.3 COMMENT ON CONVEXITY

In general, an invertible link function Ψ can be composed with any proper loss to yield a proper composite loss. For numerical convenience, it is useful to consider only those proper losses which yield a convex proper composite loss. For a proper loss λ , let $\ell(Y, v) = \lambda(Y, \Psi^{-1}(v))$ for $\Psi^{-1}(v) \in \Sigma_{\text{sig}}$ i.e. $\Psi^{-1}(v) = \frac{1+e^{-v}}{2}$ for some v . Such a loss will have Bayes-optimal scorer as given by Proposition 44, but when will such a loss be additionally convex? Suppose the weight function w for λ is normalised such that $w(\frac{1}{2}) = 1$. Then, ℓ is convex only if the weight function w satisfies (Reid and Williamson, 2010, Theorem 29)

$$w(c) \in \left[\min \left(\frac{1}{a \cdot c^2 \cdot (1-c)}, \frac{1}{a \cdot c \cdot (1-c)^2} \right), \max \left(\frac{1}{a \cdot c^2 \cdot (1-c)}, \frac{1}{a \cdot c \cdot (1-c)^2} \right) \right].$$

The above gives necessary conditions¹⁷ for obtaining a convex proper composite loss with the given link.

As a sanity check, two losses encountered earlier in Corollary 46 will indeed satisfy the above. For the admissible weight function $w(c) = \frac{1}{a \cdot c \cdot (1-c)^{3/2}}$, it is easy to check that with a sigmoidal link we recover a generalised version of the exponential loss, $\ell(Y, v) = \frac{1}{a} e^{-v/w}$. (We will revisit these family of losses in a different context in §9.5.) Recall from Equation 13 that a link Ψ is canonical for a given proper loss λ with weight function w when $w(c) = \Psi'(c)$. For $\Psi^{-1}(v) = \frac{1}{1+e^{-v/w}}$, we have $w(c) = \frac{1}{a} \cdot \left(\log \frac{c}{1-c} \right)' = \frac{1}{a \cdot c \cdot (1-c)}$. The resulting proper composite loss is

$$\ell(Y, v) = \lambda(Y; \Psi^{-1}(v)) = \frac{1}{a} \cdot \log(1 + e^{-v/w}),$$

which is a generalised logistic loss. (Masmadi-Shirazi and Vasconcelos (2010) call this the canonical logistic loss.) Note that $\lim_{v \rightarrow -\infty} \frac{1}{a} \log(1 + e^{-v/w}) = \max(0, -v)$, which is the perceptron loss.

7.5 Bipartite Ranking: Non-Decomposable Case

We now turn to the case where the loss ℓ does *not* have a decomposable Bayes-optimal pair-scorer. As noted earlier, we can no longer resort to reasoning solely via the conditional risk. Fortunately, the simple structure of $\mathcal{S}_{\text{Decomp}}$ means that we can hope to directly compute the risk minimiser via an appropriate derivative. Under some assumptions on the loss, it turns out that the Bayes-optimal scorer is still a strictly monotone transform of η ; however, the transform is now *distribution dependent*, rather than simply the fixed link function Ψ .

Proposition 47 Pick any $D = \langle M, \eta \rangle = \langle P, Q, \pi \rangle \in \Delta^{\mathcal{X} \times \mathcal{Y} \times \{+1\}}$ and a differentiable, convex, symmetric strictly proper composite loss $\ell(Y, v) = \phi(Yv)$. If ϕ' is bounded,¹⁸ or the support of D is finite,

$$\mathcal{S}_{\text{BR}}^*(D, \ell) = \{s^* : \mathcal{X} \rightarrow \mathbb{R} \mid \eta = f_{D, s^*} \circ s^*\},$$

where

$$(vD \in \mathcal{Y}) f_{D, s^*}(v) = \frac{\pi \cdot \mathbb{E}_{\mathcal{X} \sim P} [e^{\ell'}_{-1}(v - s^*(X))] }{\pi \cdot \mathbb{E}_{\mathcal{X} \sim P} [e^{\ell'}_{-1}(v - s^*(X))] - (1 - \pi) \cdot \mathbb{E}_{\mathcal{X} \sim Q} [e^{\ell'}(v - s^*(X))]}.$$

¹⁷ More complex sufficient conditions may also be derived; see (Reid and Williamson, 2010, Theorem 24).

¹⁸ We suspect this requirement may be dropped, but defer to future work investigating minimal conditions for the result to hold.

Proof The basic proof strategy follows Uematsu and Lee (2012, Theorem 3), although the subsequent steps and connection to proper loss concepts are novel; we will shortly discuss the connection to the results of that paper.

Let $\ell(Y, v) = \phi(Yv)$. For fixed D , let $\mathcal{S}(D)$ denote the space of all Lebesgue-measurable scorers $s : \mathcal{X} \rightarrow \mathbb{R}$, with addition and scalar multiplication defined pointwise, such that

$$\mathbb{L}_{\text{BR}}(s; D, \ell) = \mathbb{E}_{\mathcal{X} \sim P, \mathcal{X}' \sim Q} [\phi(s(X) - s(X'))] < \infty.$$

Then $\mathbb{L}_{\text{BR}} : \mathcal{S}(D) \rightarrow \mathbb{R}$ is a convex functional, by virtue of ϕ being convex. Thus, its minimisers may be determined by considering an appropriate notion of functional derivative. We shall employ the Gâteaux variation.

Pick any $s, t \in \mathcal{S}(D)$. For any $\epsilon > 0$, define

$$\begin{aligned} F_s(\epsilon) &= \mathbb{L}_{\text{BR}}(s + \epsilon t) \\ &= \mathbb{E}_{\mathcal{X} \sim P, \mathcal{X}' \sim Q} [\phi(s(X) - s(X') + \epsilon(t(X) - t(X')))]. \end{aligned}$$

The Gâteaux variation of \mathbb{L}_{BR} at s in the direction of t is (Trotman, 1996, pg. 45; Giacomini and Hildbrandt, 2004, pg. 10)

$$\begin{aligned} \delta \mathbb{L}_{\text{BR}}(s; t) &= \lim_{\epsilon \rightarrow 0} \frac{\mathbb{L}_{\text{BR}}(s + \epsilon t) - \mathbb{L}_{\text{BR}}(s; D, \ell)}{\epsilon} \\ &= F'_s(0), \end{aligned}$$

assuming the latter exists. To show that $F'_s(0)$ exists, we will justify interchange of the derivative and expectation. For any $\epsilon \in (0, 1]$ and $x, x' \in \mathcal{X}$, by convexity and nonnegativity of ϕ ,

$$\begin{aligned} \left| \frac{\phi(\text{Diff}(s + \epsilon t)(x, x')) - \phi(\text{Diff}(s))(x, x')}{\epsilon} \right| &\leq |\phi(\text{Diff}(s + t)(x, x')) - \phi(\text{Diff}(s)(x, x'))| \\ &\leq \phi(|\text{Diff}(s + t)(x, x')|) + \phi(|\text{Diff}(s)(x, x')|). \end{aligned}$$

By assumption, $\mathbb{L}_{\text{BR}}(s + t)$ and $\mathbb{L}_{\text{BR}}(s; D, \ell)$ are both finite. Further,

$$\lim_{\epsilon \rightarrow 0} \frac{\phi(s(x) - s(x') + \epsilon(t(x) - t(x'))) - \phi(s(x) - s(x'))}{\epsilon} = (t(x) - t(x')) \cdot \phi'(s(x) - s(x')).$$

Thus, by the dominated convergence theorem (Folland, 1999, pg. 56), we have

$$\begin{aligned} F'_s(0) &= \mathbb{E}_{\mathcal{X} \sim P, \mathcal{X}' \sim Q} [t(X) - t(X')] \cdot \phi'(s(X) - s(X')) \\ &= \mathbb{E}_{\mathcal{X} \sim P, \mathcal{X}' \sim Q} [t(X) \cdot \phi'(s(X) - s(X'))] - \mathbb{E}_{\mathcal{X} \sim Q, \mathcal{X}' \sim P} [t(X) \cdot \phi'(s(X') - s(X))] \\ &= \int_{\mathcal{X}} t(x) \cdot r(x) dx, \end{aligned}$$

where

$$(vX \in \mathcal{X}) r(x) = \rho(x) \cdot \mathbb{E}_{\mathcal{X}' \sim Q} [\phi'(s(x) - s(X'))] - q(x) \cdot \mathbb{E}_{\mathcal{X} \sim P} [\phi'(s(X) - s(x))].$$

Now suppose $s^* : \mathcal{X} \rightarrow \mathbb{R}$ minimises the functional \mathbb{L}_{BR} . By convexity of \mathbb{L}_{BR} , it is necessary and sufficient that the Gâteaux variation is zero for $t \in \mathcal{L}(D)$ (Gelfand and Fomin, 2000, Theorem 2; Trotman, 1996, Proposition 3.3). That is,

$$(\forall t \in \mathcal{L}(D)) 0 = \int_{\mathcal{X}} t(x) \cdot r(x) dx.$$

A sufficient condition for this to hold is that r is zero (almost) everywhere, and this is in fact necessary as well (Lemma 7.1). That is, for (almost) every $x_0 \in \mathcal{X}$, we equivalently need

$$p(x_0) \cdot \mathbb{E}_{\mathcal{X} \sim Q} [\phi'(s^*(x_0) - s^*(X'))] = q(x_0) \cdot \mathbb{E}_{\mathcal{X} \sim P} [\phi'(s^*(X) - s^*(x_0))],$$

which means for (almost) every $x_0 \in \mathcal{X}$,

$$\begin{aligned} & \frac{\eta(x_0)}{1 - \eta(x_0)} \cdot \frac{1 - \pi}{\pi} = \frac{p(x_0)}{q(x_0)} \\ &= \frac{\mathbb{E}_{\mathcal{X} \sim P} [\phi'(s^*(X) - s^*(x_0))]}{\mathbb{E}_{\mathcal{X}' \sim Q} [\phi'(s^*(x_0) - s^*(X'))]} \\ &= \frac{\mathbb{E}_{\mathcal{X} \sim P} [\ell'_{-1}(s^*(X) - s^*(x_0)) - \ell'_{-1}(s^*(x_0) - s^*(X))]}{\mathbb{E}_{\mathcal{X}' \sim Q} [-\ell'_{-1}(s^*(x_0) - s^*(X')) + \ell'_{-1}(s^*(X) - s^*(x_0))]} \\ &= \frac{\mathbb{E}_{\mathcal{X} \sim P} [\ell'_{-1}(s^*(x_0) - s^*(X)) - \ell'_{-1}(s^*(X) - s^*(x_0))]}{\mathbb{E}_{\mathcal{X}' \sim Q} [\ell'_{-1}(s^*(x_0) - s^*(X)) - \ell'_{-1}(s^*(X) - s^*(x_0))]} \\ &= \frac{\mathbb{E}_{\mathcal{X} \sim P} [\ell'_{-1}(s^*(x_0) - s^*(X))]}{\mathbb{E}_{\mathcal{X}' \sim Q} [\ell'_{-1}(s^*(x_0) - s^*(X))]} \text{ since } \ell' \text{ is symmetric,} \end{aligned}$$

which means

$$\eta = f_{D,s^*} \circ s^*,$$

where f_{D,s^*} is given by

$$(f_{D,s^*})(v) = \frac{\pi \cdot \mathbb{E}_{\mathcal{X} \sim P} [\ell'_{-1}(v - s^*(X))]}{\pi \cdot \mathbb{E}_{\mathcal{X} \sim P} [\ell'_{-1}(v - s^*(X)) - (1 - \pi) \cdot \mathbb{E}_{\mathcal{X}' \sim Q} [\ell'_{-1}(v - s^*(X'))]}.$$

In order to express any optimal scorer s^* in terms of η , as we have done for the previous cases, it remains to check whether or not the above the function f_{D,s^*} defined above is invertible. The following corollary provides sufficient conditions for this to hold.

Corollary 48 *Pick any $D = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and margin-based strictly proper composite loss $\ell(y, v) = \phi(yv)$, where ϕ is differentiable, strictly convex, and satisfies*

$$(\forall v \in \mathbb{R}) \phi'(v) = 0 \iff \phi'(-v) \neq 0.$$

Then if ϕ' is bounded or the support of D is finite, f_{D,s^} is invertible and*

$$S_{\text{BR}}^*(D, \ell) = \{s^* : \mathcal{X} \rightarrow \mathbb{R} \mid s^* = (f_{D,s^*})^{-1} \circ \eta\} \subseteq S_{\text{BR}}^*(D, \ell_0),$$

where f_{D,s^} is defined as in Proposition 47.*

Proof We show that f_{D,s^*} strictly monotone, by establishing the strict monotonicity of

$$g : v \mapsto \frac{\mathbb{E}_{\mathcal{X}' \sim Q} [\ell'_{-1}(v - s^*(X'))]}{\mathbb{E}_{\mathcal{X} \sim P} [\ell'_{-1}(v - s^*(X))]}.$$

The derivative of this function is

$$g'(v) = \frac{1}{\left(\mathbb{E}_{\mathcal{X} \sim P} [\ell'_{-1}(v - s^*(X))] \right)^2} \cdot \left(\mathbb{E}_{\mathcal{X} \sim P, \mathcal{X}' \sim Q} \left[\ell''_{-1}(v - s^*(X)) \ell''_{-1}(v - s^*(X')) \right] - \ell''_{-1}(v - s^*(X)) \ell''_{-1}(v - s^*(X')) \right).$$

By convexity of ℓ , the terms $\ell''_{-1}(v - s^*(X'))$ and $\ell''_{-1}(v - s^*(X))$ are positive. Further, by (Vernet et al., 2011, Proposition 15), ℓ_{-1} and ℓ_{-1} are increasing and decreasing respectively, or vice-versa. By assumption their derivatives cannot simultaneously be zero. Therefore the expectand is always positive or negative for every v , and hence $g'(v)$ is always strictly positive or negative. Thus g is strictly monotone, which means f_{D,s^*} is as well. Therefore, $s^* = (f_{D,s^*})^{-1} \circ \eta$. ■

The link function f_{D,s^*} is at first glance peculiar because it depends on the distribution D , as well as the optimal scorer s^* . From a practical perspective, the result is thus not helpful in terms of helping quickly discover the optimal scorer s^* . However, what is of interest to us is simply that this function is strictly monotone. This means that from an AUC perspective, using a proper composite surrogate loss asymptotically results in a desirable score i.e. if we rank examples according to s^* , it is equivalent to ranking them according to η .

As before, any optimal scorer for such a proper composite loss is also optimal for ℓ_{01} , despite the link function f_{D,s^*} depending on the distribution D . Appendix H provides an empirical illustration that this link is indeed invertible under the specified conditions, albeit distribution dependent. The results of this section established that a suitably restricted notion of convexity is *sufficient* for the optimal scorer to be a strictly monotone transform of η , while the previous section established convexity is *not necessary*, since one can have a non-convex loss resulting from a suitable link $\Psi = \frac{1}{t} \sigma^{-1}$.

7.5.1 COMPARISON WITH STANDARD LINK FUNCTION

Recall from Equation 12 that the link function Ψ associated with a proper composite loss ℓ satisfies

$$\Psi^{-1}(v) = \frac{\ell'_{-1}(v)}{\ell'_{-1}(v) - \ell'_1(v)}.$$

This is in general *not* the same as the inverse link function $(f_{D,s^*})^{-1}$ from the above result for a simple reason: the latter potentially depends on the distribution D , and the optimal scoring function s^* itself. However, the forms of the two functions are closely related: in $(f_{D,s^*})^{-1}$, each quantity from the inverse link Ψ^{-1} is replaced by its expected value under an appropriate distribution.

In the previous section, we saw that under certain conditions $s^* = \Psi \circ \eta$, where Ψ is the standard link function associated with ℓ . In such cases, it is of interest to see whether f_{D,s^*} simplifies. For example, for the case of exponential loss $\ell(y, v) = e^{-yv}$, with inverse link function $\Psi^{-1}(v) = \frac{1}{1 + e^{-2v}}$, we get

$$(f_{D,s^*})^{-1}(v) = \frac{\mathbb{E}_{\mathcal{X} \sim M} [\eta(X) \cdot e^{s^*(X)}]}{\mathbb{E}_{\mathcal{X} \sim M} [\eta(X) \cdot e^{s^*(X)}] + e^{-2v} \cdot \mathbb{E}_{\mathcal{X} \sim M} [(1 - \eta(X)) \cdot e^{-s^*(X)}]}.$$

It can be verified that when plugging in $s^* = \Psi \circ \eta$, one finds $(f_{D,s^*})^{-1}(v) = \Psi^{-1}(v)$ as expected, and therefore the dependence on the distribution “disappears”. Determining conditions beyond $\Psi \in \Sigma_{\text{sig}}$ for which f_{D,s^*} simplifies would be of interest.

In class-probability estimation with a proper composite loss, there is a separation of concerns between the underlying proper loss and the link function Ψ , with the latter primarily chosen for computational convenience, and not affecting statistical properties of the proper loss (Reid and Williamson, 2010). For bipartite ranking,

however, such a separation of concerns is guaranteed only when one operates with the family of link functions from Proposition 44. For this family, the Bayes-optimal scorer is any translation of $\Psi \circ \eta$, while Corollary 48 indicates that outside this family, the optimal scorer may be a distribution-dependent transformation of η . Thus, changing the link function in bipartite ranking can change the optimal solutions to the risk in a non-trivial way.

7.6 Relation to Existing Work

The study of Bayes-optimal scorers for pairwise and bipartite ranking problems does not seem as extensive as for the binary classification setting. The study of the Bayes-optimal scorers for both problems under proper composite losses appears to be novel, although Agarwal (2014) employs proper losses in theoretical analysis related to the bipartite ranking problem. Even our derivation of the form of η_{pair} does not have many precedents, though Uematsu and Lee (2012) implicitly derive the formula.

This section generalised and unified several earlier results through the theory of proper losses. For ℓ_{01} -our Corollary 43 is well-known in the context of scorers that maximise the AUC, which is one minus the bipartite ℓ_{01} risk. The result is typically established by the Neyman-Pearson lemma (Torgersen, 1991), whereas we simply use a reduction to binary classification over pairs. For exponential loss with a linear hypothesis class, Erekin and Rudin (2011) studied the (empirical) Bayes-optimal solutions. For a convex margin loss, Uematsu and Lee (2012) and Gao and Zhou (2012, 2015) independently studied conditions for the Bayes-optimal scorers to be transformations of η . Our Proposition 44 is a generalisation of Uematsu and Lee (2012, Theorem 7); Gao and Zhou (2012, Lemma 3); Gao and Zhou, 2015, (Corollary 1), where our result holds for non-symmetric and non-convex proper composite losses; Appendix G has an empirical illustration of this. Our Corollary 48 is essentially equivalent to Uematsu and Lee (2012, Theorem 3); Gao and Zhou (2012, Theorem 5); Gao and Zhou (2015, Theorem 2), although we explicitly provide the form of the link function relating η and s^* . (We translate these results in terms of proper losses so that the connection is more apparent in Appendix F.)

Uematsu and Lee (2012, Theorem 5) also showed that for the hinge loss, which is classification calibrated, there are possibly ties introduced in the ranking, which is not covered by our results as the hinge loss is not proper; Gao and Zhou (2012, Theorem 2, 3); Gao and Zhou (2015, Lemma 3) similarly show that hinge and absolute loss do not produce a consistent ranking.

8. Surrogate Regret Bounds for Pairwise Surrogate Minimisation

At this stage, we have established that for suitable ℓ , minimisers of the ℓ -bipartite risk will also minimise the ℓ_{01} bipartite risk. Equivalently, these scorers will maximise the AUC. This can be seen as a justification for the minimisation of a pairwise surrogate to ℓ_{01} for the task of maximising the AUC; this is sometimes referred to as the pairwise approach to bipartite ranking.

In practice, of course, we cannot expect to perfectly minimise L_{BR} due to having access to a finite sample, and (or) using a restricted function class of scorers. Thus, we would ideally like a bound as to how much worse the AUC can be when using a *suboptimal* minimiser of $L_{\text{BR}}(s; D, \ell)$. This is the analogue of *surrogate regret bounds* for classification, which establish that convex risk minimisation is consistent for the problem of 0-1 minimisation (Zhang, 2004; Bartlett et al., 2006; Reid and Williamson, 2009).

Surrogate regret bounds have previously been shown for bipartite ranking by Cléménçon et al. (2008, Section 7), under the implicit assumption of decomposable scorers, and for the symmetric exponential and logistic losses by Gao and Zhou (2012, Corollary 6, 7); Gao and Zhou (2015, Corollary 5). It turns out that the analysis of the previous section automatically implies the existence of surrogate regret bounds for pairwise minimisation of suitable (possibly asymmetric) proper composite losses. Formally, we have the following.

Proposition 49 *Pick any $D \in \Delta_{\mathcal{X} \times \mathcal{X}}^{\pm 1}$ and scorer $s : \mathcal{X} \rightarrow \mathbb{R}$. Given any $\ell \in \mathcal{L}_{\text{SPEC}}(\Psi)$ with inverse link function $\Psi^{-1} \in \Sigma_{\text{SIG}}^+$, there exists a convex function $F_{\ell} : [0, 1] \rightarrow \mathbb{R}_+$ with $F_{\ell}(0) = 0$ such that,*

$$F_{\ell}(\text{regret}_{\text{BR}}(s; D, \ell_{01})) \leq \text{regret}_{\text{BR}}(s; D, \ell).$$

Proof For any $s : \mathcal{X} \rightarrow \mathbb{R}$, $s_{\text{pair}} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, recall that

$$\begin{aligned} \text{regret}(s_{\text{pair}}; D_{\text{BR}} \cdot \ell) &= \mathbb{L}(s_{\text{pair}}; D_{\text{BR}} \cdot \ell) - \inf_{f_{\text{pair}} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}} \mathbb{L}(f_{\text{pair}}; D_{\text{BR}} \cdot \ell) \\ \text{regret}_{\text{BR}}(s; D, \ell) &= \mathbb{L}_{\text{BR}}(s; D, \ell) - \inf_{f : \mathcal{X} \rightarrow \mathbb{R}} \mathbb{L}_{\text{BR}}(f). \end{aligned}$$

Existing surrogate regret bounds for proper composite losses (Reid and Williamson, 2009) imply that there exists some convex $F_{\ell} : [0, 1] \rightarrow \mathbb{R}_+$ such that, for any $D \in \Delta_{\mathcal{X} \times \mathcal{X}}^{\pm 1}$ and scorer $s : \mathcal{X} \rightarrow \mathbb{R}$,

$$F_{\ell}(\text{regret}(\text{Diff}(s); D_{\text{BR}} \cdot \ell_{01})) \leq \text{regret}(\text{Diff}(s); D_{\text{BR}} \cdot \ell).$$

By the reduction of bipartite ranking to classification over pairs (Lemma 2), for any ℓ satisfying the conditions of the proposition,

$$\begin{aligned} \text{regret}_{\text{BR}}(s; D, \ell) &= \mathbb{L}_{\text{BR}}(s; D, \ell) - \inf_{f : \mathcal{X} \rightarrow \mathbb{R}} \mathbb{L}_{\text{BR}}(f) \\ &= \mathbb{L}(\text{Diff}(s); D_{\text{BR}} \cdot \ell) - \inf_{f_{\text{pair}} \in \mathcal{S}_{\text{Decomp}}} \mathbb{L}(f_{\text{pair}}; D_{\text{BR}} \cdot \ell) \\ &= \mathbb{L}(\text{Diff}(s); D_{\text{BR}} \cdot \ell) - \inf_{f_{\text{pair}} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}} \mathbb{L}(f_{\text{pair}}; D_{\text{BR}} \cdot \ell) \\ &= \text{regret}(\text{Diff}(s); D_{\text{BR}} \cdot \ell), \end{aligned}$$

where in the penultimate line we have used the fact that the restriction to $\mathcal{S}_{\text{Decomp}}$ can be removed by virtue of the loss ℓ inducing a decomposable Bayes-optimal pair-scorer for D_{BR} (by Proposition 44). We similarly know that ℓ_{01} induces a decomposable pair-scorer (Proposition 41). Thus, we can write the regret bound as

$$F_{\ell}(\text{regret}_{\text{BR}}(s; D, \ell_{01})) \leq \text{regret}_{\text{BR}}(s; D, \ell). \quad \blacksquare$$

The function $F_{\ell} : [0, 1] \rightarrow \mathbb{R}_+$ in Proposition 49 is exactly that which appears in bounds relating 0-1 to ℓ classification regret for proper composite losses, and may be specified in terms of the conditional Bayes-risk of ℓ as (Reid and Williamson, 2009, Theorem 3)

$$\begin{aligned} F_{\ell} : u &\mapsto L^*\left(\frac{1}{2}\right) + G_{\ell}(u) \vee G_{\ell}(-u) \\ G_{\ell} : u &\mapsto -L^*\left(\frac{1}{2} + u\right) + (L^*)'\left(\frac{1}{2}\right) \cdot u. \end{aligned}$$

We make three observations about this result. First, the bound implies the consistency of pairwise surrogate minimisation for losses satisfying the conditions of the proposition, and whose underlying proper loss λ additionally satisfy the regularity condition $L^*(0) = 0$, as

$$\begin{aligned} \text{regret}_{\text{BR}}(s; D, \ell) \rightarrow 0 &\implies F_{\ell}(\text{regret}_{\text{BR}}(s; D, \ell_{01})) \rightarrow 0 \\ &\implies \text{regret}_{\text{BR}}(s; D, \ell_{01}) \rightarrow 0, \end{aligned}$$

where the second line is because $L^*(u) > 0$ on $(0, 1/2]$ by strict concavity of the conditional Bayes risk (a consequence of strict properness of the loss), and $L^*(0) = 0$ by assumption.

Second, the bound places no convexity restriction on ℓ . This is akin to similar regret bounds for classification (Bartlett et al., 2006, Theorem 1), where the surrogate loss need not be convex. Of course, for non-convex ℓ , guaranteeing $\text{regret}_{\text{BR}}(s; D, \ell) \rightarrow 0$ is more challenging.

Third, when the optimal pair-scorer is *not* decomposable, the proof breaks when attempting to equate $\text{regret}_{\text{BR}}(s; D, \ell)$ and $\text{regret}(\text{Diff}(s); D_{\text{BR}} \cdot \ell)$, and so more effort is needed to derive a surrogate regret bound. This further illustrates the value of the decomposability of the Bayes-optimal pair-scorer as studied in the previous section. Note that while we do not have a regret bound for such losses, Corollary 48 established a

sufficient condition for agreement of the Bayes-optimal scorers. Further, [Gao and Zhou \(2012\)](#), [Theorem 2](#) showed that for a subset of such losses, one has asymptotic consistency of the surrogate minimisation (even when a regret bound is elusive).

As a final note, the above is distinct from [Agarwal \(2014\)](#) as the latter bounds the AUC regret in terms of the regret with respect to a proper composite loss. That is, the result shows the consistency of the class-probability estimation approach to bipartite ranking. This is distinct to our bound, which shows the consistency of the surrogate pairwise ranking approach to bipartite ranking.

9. Ranking the Best Instances

In most practical applications of ranking, accuracy at the head of the ranked list is more important than accuracy at the tail. For example, in information retrieval, typically only the first few elements of the ranked result set for a query are considered by a user of the system. It is thus of interest to consider notions of risk that focus on accuracy at the head of the list. This problem is sometimes called *ranking the best* ([Cléménçon and Vayatis, 2007](#)) or *accuracy at the top* ([Boyd et al., 2013](#)). We will use the terminology “ranking the best”.

We will now formalise the ranking the best problem, and see how the tools we have developed thus far may be applied to address it.

9.1 Formal Definition of Ranking the Best

[Corollary 43](#) shows the AUC is maximised by any strictly monotone increasing transformation of η , the observation-conditional distribution. Thus, from the perspective of the AUC, the optimal ranked list in bipartite problems involves ordering instances based on their η values. In the ranking the best problem, our goal is to ensure that instances $x \in \mathcal{X}$ for which $\eta(x)$ is large are correctly ordered relative to other instances, potentially at the expense of incorrectly ordering instances $x' \in \mathcal{X}$ for which $\eta(x')$ is small.

Formally, given any $q \in [0, 1]$, we call a loss ℓ a q -RTB loss (for q -rank-the-best) if the Bayes-optimal scorer for $D = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ is

$$S^*(D, \ell) = \{\Psi_q \circ \eta\},$$

where

$$(\forall z \in [0, 1]) \Psi_q(z) \in \begin{cases} \{z\} & \text{if } z \geq q \\ [0, z] & \text{if } z < q, \end{cases} \quad (52)$$

Such a scorer does not demand accurate estimation of η below the fixed threshold q : all that is required is that the ordering is preserved relative to those instances with score bigger than q .

According to [Equation 52](#), any loss for which $S^*(D, \ell) = \{\Psi_{\text{opt}}\}$ for some invertible Ψ is also a $\Psi^{-1}(q)$ -RTB loss for any $q \in [0, 1]$. This simply says that if we accurately model *all* ranks, then by definition we accurately model ranks at the head of the list. Indeed, if we could operate on the distribution directly, there would be no tradeoff to be made between accurately modelling any particular portion of the list: η would be recovered exactly, and thus the entire list could be ranked perfectly. The value of a loss that relaxes the modelling requirements for $\eta < q$ arises when we have either finite samples or a misspecified hypothesis class, in which case tradeoffs are necessary.

Having defined the goal of the ranking the best problem, we look to review some performance measures for this task. We then explore alternate risks that have similar characteristics, but are designed using the theory of proper composite losses. We shall begin with a general framework that follows [Cléménçon and Vayatis \(2008\)](#); [Rudin \(2009\)](#).

9.2 The (Reverse) (ℓ, g)-Push Framework for Ranking the Best

Most established performance measures encourage focussing on the head of the ranked list in one of two (related) ways:

- (i) ensuring that most negative instances appear below the positive instances, or

- (ii) ensuring that most positive instances appear near the head of the list.

The former approach involves ensuring that the false negative rate is small for certain negative instances. The latter approach relies on minimising the following quantity: given an instance $x \in \mathcal{X}$, we define its *normalised rank* under a scorer $s : \mathcal{X} \rightarrow \mathbb{R}$ and distribution $D = \langle M, \eta \rangle$ to be the fraction of examples that have a higher score than it:

$$\begin{aligned} \text{NRank}(x; D, s) &= \mathbb{P}_{\mathcal{X} \sim M} [s(X) > s(x)] + \frac{1}{2} \cdot \mathbb{P}_{\mathcal{X} \sim M} [s(X) = s(x)] \\ &= \pi \cdot \text{TPR}(s(x); D, s) + (1 - \pi) \cdot \text{FPR}(s(x); D, s). \end{aligned} \quad (53)$$

Small normalised ranks are desired for positive examples, and large ranks for negative examples. Our definition of the normalised rank is simply called the “rank” by [Rudin \(2009\)](#), [Section 7](#).

For each of the above approaches, we consider a general family of risks that can be specialised to yield various performance measures of interest. For approach (i), [Rudin \(2009\)](#); [Swamidass et al. \(2010\)](#) studied a family of risks parameterised by a monotone increasing function.¹⁹ Generalising these proposals to the case of an arbitrary symmetric loss ℓ , and a pair-scorer $s_{\text{pair}} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, we obtain the (ℓ, g)-push risk:

$$\text{Push}(s_{\text{pair}}; D, \ell, g) = \mathbb{E}_{\mathcal{X} \sim Q} \left[g \left(\mathbb{E}_{\mathcal{X}' \sim P} [\ell_1(s_{\text{pair}}(X, X'))] \right) \right],$$

where $g : \mathbb{R} \rightarrow \mathbb{R}_+$ is a monotone increasing function. For decomposable pair-scorers,

$$\text{Push}(\text{Diff}(s); D, \ell, g) = \mathbb{E}_{\mathcal{X}' \sim Q} \left[g \left(\text{FNR}_{\ell}(s(X')) \right) \right].$$

Compared to the standard bipartite ℓ -ranking risk ([Equation 16](#)), the difference is that the inner expectation is transformed by the function $g(\cdot)$. Intuitively, if $g(\cdot)$ is convex with a sharply increasing slope, the risk penalises high false negative rates, or equally encourages high true positive rates when thresholding around the negatives.

For approach (ii), following [Cléménçon and Vayatis \(2008\)](#); [Rudin \(2009\)](#), [Section 7](#), we have an analogous *reverse*²⁰ (ℓ, g)-push risk, where the order of expectations is reversed, and one uses the normalised rank in place of a rate:

$$\text{RevPush}(s_{\text{pair}}; D, \ell, g) = \mathbb{E}_{\mathcal{X} \sim P} \left[g \left(\mathbb{E}_{\mathcal{X}' \sim M} [\ell_1(s_{\text{pair}}(X, X'))] \right) \right],$$

where $g : \mathbb{R} \rightarrow \mathbb{R}_+$ is a monotone increasing function. For decomposable pair-scorers,

$$\text{RevPush}(\text{Diff}(s); D, \ell, g) = \mathbb{E}_{\mathcal{X} \sim P} \left[g \left(\text{NRank}_{\ell}(s(X)) \right) \right],$$

where NRank_{ℓ} involves replacing the TPR and FPR in [Equation 53](#) with their TPR_{ℓ} and FPR_{ℓ} counterparts. To gain some intuition for these approaches, we show how the AUC is a special case of each.

9.3 The AUC and the (Reverse) (ℓ, g)-Push Risk

We first relate the AUC to the (ℓ, g)-push risk. Recall from [Equation 38](#) that the AUC is

$$\text{AUC}(s; D) = \mathbb{E}_{\mathcal{X}' \sim Q} \left[\text{TPR}(s(X'); D, s) \right].$$

A high AUC thus means that the negative instances, on average, are placed below the positive instances: were this not the case, then we would achieve only a low true positive rate when thresholding scores around the negatives. Formally, we have

$$\text{AUC}(s; D) = \text{Push}(\text{Diff}(s); D, \ell_{01}, g)$$

¹⁹ This family has also been considered, in a different context, by [Xie and Priebe \(2002\)](#).

²⁰ Our use of the word “reverse” is as per [Rudin \(2009\)](#), [Section 7](#).

where $g : x \mapsto 1 - x$.

We next relate the AUC to the reverse (ℓ, g) -push risk. At first glance, the AUC appears to involve a different consideration, as

$$\text{AUC}(g; D) = \mathbb{E}_{X \sim P} [\text{TNR}(g(X); D, s)],$$

so that the AUC focusses on placing positives ahead of negatives, while when maximising the normalised rank, we also consider the relationship of positives to other positives. (A similar observation is made by Rudin (2009, Section 7), where the false positive rate is called the “reverse height”). However, note that

$$\begin{aligned} \mathbb{E}_{X \sim P} [\text{NRank}(X; D, s)] &= \pi \cdot \mathbb{E}_{X \sim P} [\text{TPR}(g(X); D, s)] + (1 - \pi) \cdot \mathbb{E}_{X \sim P} [\text{FPR}(g(X); D, s)] \\ &= \pi \cdot \mathbb{E}_{X \sim P} [\text{TPR}(g(X); D, s)] + (1 - \pi) \cdot (1 - \text{AUC}(g; D)) \text{ by Equation 37} \\ &= \frac{\pi}{2} + (1 - \pi) \cdot (1 - \text{AUC}(g; D)) \text{ following Equation 40,} \end{aligned}$$

and so

$$\text{AUC}(g; D) = \frac{2 - \pi}{2 \cdot (1 - \pi)} - \frac{1}{1 - \pi} \cdot \mathbb{E}_{X \sim P} [\text{NRank}(X; D, s)].$$

A high AUC thus means that on average, the positive instances have a small normalised rank i.e. they appear near the head of the list. Formally, we thus have

$$\text{AUC}(g; D) = \text{RevPush}(\text{Dif}(g); D, g)$$

where $g : x \mapsto \frac{2 - \pi}{2(1 - \pi)} - \frac{1}{1 - \pi} \cdot x$.

9.4 Established Performance Measures for Ranking the Best

In both the above interpretations of the AUC, one focusses on average case behaviour. This is manifest in the AUC having a linear dependence on the true positive rate, as well as on the normalised rank. The basic idea of adapting the measure to focus on the head of the list is to consider a suitable nonlinear transformation $g(\cdot)$ in the (reverse) (ℓ, g) -push risk, so as to strongly penalise errors at the head over the tail. We now define some popular measures²¹ for ranking the best that do precisely this. Table 9 summarises the measures considered.

9.4.1 PARTIAL AUC

The *partial AUC* (PAUC) (McClish, 1989; Dodd and Pepe, 2003; Narasimhan and Agarwal, 2013a) of a scorer only computes the area under the ROC curve for false positive rates between $[a, b] \subseteq [0, 1]$:

$$\begin{aligned} \text{PAUC}(g; D, a, b) &= \int_a^b \text{TPR}(\text{FPR}^{-1}(a)) da \\ &= \mathbb{E}_{X \sim P} [\text{TPR}(g(X)) \cdot \mathbb{1}[a \leq \text{FPR}(g(X)) \leq b]]. \end{aligned}$$

When $a = 0$ and $b \ll 1$, this intuitively focusses only on performance at the head of the ranked list (as this corresponds to thresholds with low false positive rate). This measure is evidently related to the special case of the reverse (ℓ_0, g) -push risk for $g : x \mapsto (x \vee a) \wedge b$, where one uses the true positive rate in place of the normalised rank.

²¹ Most of these measures have their origins in information retrieval. Here, they are typically stated in terms of results for “queries”. We effectively treat our labelled samples as the set of results for a single query. Measures that average across multiple queries, which would correspond to a multilabel learning problem, are thus not considered.

Performance measure	Symbol	Definition
Partial AUC	$\text{PAUC}(g; D, b)$	$\mathbb{E}_{X \sim P} [\text{TPR}(g(X)) \cdot \mathbb{1}[\text{FPR}(g(X)) \leq b]]$
Average precision	$\text{AP}(g; D)$	$\mathbb{E}_{X \sim P} \left[\frac{\text{TPR}(g(X; D, s))}{\text{NRank}(X; D, s)} \right]$
Discounted cumulative gain	$\text{DCG}(g; D)$	$\mathbb{E}_{X \sim P} \left[\frac{1}{\log(1 + \text{NRank}(X; D, s))} \right]$
Average reciprocal rank	$\text{ARR}(g; D)$	$\mathbb{E}_{X \sim P} \left[\frac{1}{\text{NRank}(X; D, s)} \right]$
Reciprocal rank	$\text{RR}(g; D)$	$\sup_{Y \in \text{supp}(P)} \frac{1}{\text{NRank}(X; D, s)}$
(Negated) p -norm push	$\text{Push}(g; D, p)$	$\mathbb{E}_{X \sim Q} \left[-(\text{NRank}(g(X); D, s))^p \right]$
Positives at top	$\text{PTop}(g; D)$	$\inf_{Y \in \text{supp}(Q)} \text{TPR}(g(X^*); D, s)$

Table 9: Performance measures for ranking the best. For each measure, larger values are desirable.

9.4.2 AVERAGE PRECISION

Our next measure relies on the following two quantities.

Definition 50 Given any distribution $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and scorer $s : \mathcal{X} \rightarrow \mathbb{R}$, define the *precision and recall at a threshold* $t \in \mathbb{R} \cup \{\pm\infty\}$ to be

$$\begin{aligned} \text{Prec}(t; D, s) &= \mathbb{P}[Y = 1 | s(X) > t] \\ \text{Rec}(t; D, s) &= \text{TPR}(t; D, s) = \mathbb{P}[s(X) > t | Y = 1] + \frac{1}{2} \cdot \mathbb{P}[s(X) = t | Y = 1]. \end{aligned}$$

When the scorer and distribution are clear from context, we shall drop the dependence on them and simply write $\text{Prec}(t)$, $\text{Rec}(t)$.

The precision may be related to the more familiar rates introduced earlier: if the distribution of scores has no discrete components, then by Bayes’ rule,

$$\begin{aligned} \text{Prec}(t; D, s) &= \frac{\mathbb{P}[s(X) > t | Y = 1] \cdot \mathbb{P}[Y = 1]}{\mathbb{P}[s(X) > t]} \\ &= \frac{\pi \cdot \text{TPR}(t) + (1 - \pi) \cdot \text{FPR}(t)}{\pi \cdot \text{TPR}(t)} \\ &= \left(1 + \frac{1 - \pi}{\pi} \cdot \frac{\text{FPR}(t)}{\text{TPR}(t)} \right)^{-1}. \end{aligned} \tag{54}$$

Note that if we use as threshold $t = s(x)$ for some $x \in \mathcal{X}$, then the denominator of Equation 54 is nothing but $\text{NRank}(t)$.

We now define the *average precision* (AP) of a scorer s (Yue et al., 2007; Chakrabarti et al., 2008; Agarwal, 2011; Boyd et al., 2012) to be the average of the precisions obtained using the scores of positive examples as thresholds:

$$\begin{aligned} \text{AP}(g; D) &= \mathbb{E}_{X \sim P} [\text{Prec}(g(X); D, s)] \\ &= \pi \cdot \mathbb{E}_{X \sim P} \left[\frac{\text{TPR}(g(X); D, s)}{\text{NRank}(X; D, s)} \right], \end{aligned} \tag{55}$$

where the second equation follows from Equations 53 and 54.

The average precision can be shown to favour accuracy at the head of the list more than the AUC (Yue et al., 2007). Intuitively, this is because when there is a spurious negative example high in the ranked list, it will substantially affect the precision of the nearby positive examples. This may also be seen through Equation 55: we encourage very low normalised ranks for the positive examples, which corresponds to placing them at the very top of the list. Compared to the AUC, placing a few positives at the very top gives a greater gain than placing many positives only roughly near the top.

Further intuition for the average precision can be gained by considering the precision-recall curve, a complement to the ROC curve.

Definition 51 Given any distribution $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and scorer $s: \mathcal{X} \rightarrow \mathbb{R}$, the *precision-recall curve* is defined by the *parametric representation*

$$\text{PR}(s; D) \doteq \{(\text{Rec}(t; D, s), \text{Prec}(t; D, s)) : t \in \mathbb{R} \cup \{\pm\infty\}\} \subseteq [0, 1]^2.$$

The area under the precision recall curve (AUPRC) of s is the area under the curve $\text{PR}(s; D)$ (Boyd et al., 2013):

$$\text{AUPRC}(s; D) \doteq \int_0^1 \text{Prec}(\text{Rec}^{-1}(\alpha)) d\alpha.$$

We immediately see that compared to the ROC curve, the PR curve depends on the base rate π . We also see that like the ROC curve, the curve can be computed from the TPR and FPR. Following the Neyman-Pearson analysis (Corollary 16), we can conclude that the area under the PR curve is optimized by any strictly monotone increasing transform of $\eta(x)$ (Cléménçon and Vayatis, 2009a).

In fact, the average precision is exactly equal to the AUPRC.

Lemma 52 (Boyd et al., 2013) Given any $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and scorer $s: \mathcal{X} \rightarrow \mathbb{R}$ with differentiable ROC curve and invertible rates,

$$\text{AUPRC}(s; D) = \text{AP}(s; D).$$

Proof By definition,

$$\begin{aligned} \text{AUPRC}(s; D) &= \int_0^1 \text{Prec}(\text{Rec}^{-1}(\alpha)) d\alpha \\ &= - \int_{-\infty}^{\infty} \text{TPR}'(t) \cdot \text{Prec}(t) dt \text{ using } \alpha = \text{TPR}(t) \\ &= \int_{-\infty}^{\infty} p_S(t) \cdot \text{Prec}(t) dt \text{ by Equation 22} \\ &= \mathbb{E}_{\mathcal{X} \sim P} \left[\int_{-\infty}^{\delta_{s(X)}(t)} \text{Prec}(t) dt \right] \\ &= \mathbb{E}_{\mathcal{X} \sim P} [\text{Prec}(s(X))]. \end{aligned}$$

■

9.4.3 DISCOUNTED CUMULATIVE GAIN

The *discounted cumulative gain* (DCG) of a scorer s (Järvelin and Kekäläinen, 2002; Agarwal, 2011; Boyd et al., 2012) is the average of the inverse logarithm of the normalised rank for all positive examples:

$$\text{DCG}(s; D) \doteq \mathbb{E}_{\mathcal{X} \sim P} \left[\frac{1}{\lg(1 + \text{NRank}(X; D, s))} \right].$$

Compared to average precision (Equation 55), the DCG applies a nonlinear decay on the effect of lower ranked positives. It is evident that the DCG is a special case of the reverse $(\ell_{0,1}, g)$ -push risk with $g: x \mapsto 1/\log(1+x)$. It may also be seen as a limiting case of the family of risks with $g_p: x \mapsto (1/p)/((1+x)^p/p - 1)$ as $p \rightarrow \infty$.

9.4.4 AVERAGE RECIPROCAL RANK

The *average reciprocal rank* (ARR) of a scorer s (Rudin, 2009, Section 7) is the inverse of the harmonic mean of the normalised ranks for all positive examples:

$$\text{ARR}(s; D) = \mathbb{E}_{\mathcal{X} \sim P} \left[\frac{1}{\text{NRank}(X; D, s)} \right].$$

This measure encourages small normalised rank values for the positives, meaning that spurious negatives near the head of the list will adversely affect the scores for several examples. Compared to the average precision (Equation 55), one does not additionally weigh this inverse rank by the true positive rate. It is evident that the ARR is a special case of the reverse $(\ell_{0,1}, g)$ -push risk with $g: x \mapsto 1/x$.

9.4.5 RECIPROCAL RANK

The *reciprocal rank* (RR) of a scorer s (Voorhees, 2001; Chakrabarti et al., 2008) is the inverse of the rank of the top positive:

$$\text{RR}(s; D) \doteq \sup_{x \in \text{supp}(P)} \frac{1}{\text{NRank}(x; D, s)}.$$

where $\text{supp}(\cdot)$ denotes the support of a distribution. This measure directly encourages the first element of the ranked list to be a positive. Compared to average precision (Equation 55), roughly, we replace the average performance over all positives with simply the performance of the best positive. The RR can be seen as a limiting case of a family of reverse $(\ell_{0,1}, g)$ -push risks with $g = g_p: x \mapsto 1/x^p$ as $p \rightarrow \infty$.

Compared to the RR, the ARR considers the ranks of *all* positives, not just the top one. This intuitively makes the ARR more suitable when one is interested not just at the very first element of the list, but rather on the first k elements for some small constant k .

9.4.6 THE p -NORM PUSH

Rudin (2009) provides a detailed study of the p -norm push, which the (ℓ, g) -push risk for the choice $g: x \mapsto x^p$ for $p \in [1, \infty)$ and symmetric ℓ , leading to the p -norm push risk:

$$\begin{aligned} \text{Push}(\text{Diff}(s); D, \ell, \cdot^p) &= \mathbb{E}_{\mathcal{X}' \sim Q} \left[(\text{FNR}_{\ell}(s(X')))^p \right] \\ &= \mathbb{E}_{\mathcal{X}' \sim Q} \left[\left(\mathbb{E}_{\mathcal{X} \sim P} [\ell_1(s(X) - s(X'))] \right)^p \right]. \end{aligned} \quad (56)$$

By increasing the value of p , one strongly penalises high false negative rates.

9.4.7 POSITIVES AT THE TOP

The *fraction of positives at the top* (PTop) of a scorer s (Agarwal, 2011; Boyd et al., 2012) is typically defined on an empirical sample $\mathcal{D} = \{(x_j, 1)\}_{j=1}^m \cup \{(x_j, -1)\}_{j=1}^m$ as the number of positive instances ranked above the highest negative instance, or equally, the minimum over all negative instances of the number of positives ranked above that instance.

$$\text{PTop}(s; \mathcal{D}) \doteq \min_{1 \leq j \leq m} \ell_{01}(s(x_j) - s(x_j)).$$

Li et al. (2014) provided an efficient algorithm to optimise surrogates to this measure. Evidently, its population counterpart is

$$\text{PTop}(s; D) = \inf_{s' \in \text{Supp}(Q)} \text{TPR}(s(s'; D, s),$$

with negation

$$1 - \text{PTop}(s; D) = \sup_{s' \in \text{Supp}(Q)} \text{FNR}(s(s'; D, s). \quad (57)$$

Compared to the AUC (Equation 38), which looks to make the *average* rank of all negative instances small, the PTop looks to make the *worst possible* rank over all negative instances small. The PTop can be related to the p -norm push risk, as

$$\lim_{p \rightarrow \infty} \text{Push}(s; D, \ell, \rho)^{1/p} = \sup_{s' \in \text{Supp}(Q)} \text{FNR}_{\rho}(s(s'; D).$$

For the case of ℓ_{01} , this is exactly the negation of the positives at the top measure (Equation 57).

9.5 Bayes-Optimal Scorers for the (ℓ, g) -Push Risk

Having introduced a number of performance measures, we now study why suitable choices of g for the (ℓ, g) -push risk can be seen to focus attention at the head of the list. This is done by analysing the Bayes-optimal scorers for this family of risks, and seeing how they align with Equation 52. Specifically, we aim to determine the Bayes-optimal pair and univariate scorers for the (ℓ, g) -push risk, and study them in light of Equation 52. Unlike bipartite ranking, the risk in this case cannot (obviously) be expressed as a classification risk over pairs of instances; therefore, we separately consider the optimal pair- and univariate-scorers,

$$\begin{aligned} \text{S}_{\text{push}}^{\text{pair}^*}(D, \ell, g) &= \underset{\text{S}_{\text{pair}} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}}{\text{Argmin}} \text{Push}(\text{S}_{\text{pair}}; D, \ell, g) \\ \text{S}_{\text{push}}^*(D, \ell, g) &= \underset{s : \mathcal{X} \rightarrow \mathbb{R}}{\text{Argmin}} \text{Push}(\text{Diff}(s); D, \ell, g). \end{aligned}$$

We first analyse the case of pair-scorers, and then proceed to univariate scorers. While most of our analysis is for general ℓ and g , we shall find the p -norm push risk of Equation 56 to be particularly amenable to analysis when combined with the exponential loss.

9.5.1 BAYES-OPTIMAL PAIR-SCORERS

As with the standard bipartite risk, determining the Bayes-optimal scorer for the (ℓ, g) push is challenging due to the implicit restricted function class $\mathcal{S}_{\text{Decomp}}$. In fact, this is difficult even for the pair-scorer case: the (ℓ, g) push risk is not easily expressible in terms of a conditional risk. Thus, we explicitly compute the derivative of the risk, as in the proof of Proposition 47. We end up with the following distribution-dependent transformation of η_{pair} as our optimal scorer.

Proposition 53 *Given any $D = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \{ \pm 1 \} }$, a differentiable function $g : \mathbb{R} \rightarrow \mathbb{R}$, and a differentiable strictly proper composite loss ℓ with link function Ψ , if ℓ' , ℓ'' are bounded or \mathcal{X} is finite,*

$$\text{S}_{\text{push}}^{\text{pair}^*}(D, \ell, g) = \{ s_{\text{pair}}^* : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R} : s_{\text{pair}}^* = \Psi \circ \sigma \circ (\text{Diff}(\sigma^{-1} \circ \eta) - G(D, s_{\text{pair}}^*)) \}, \quad (58)$$

where

$$\begin{aligned} G(x, x'; D, s_{\text{pair}}) &= \log \frac{g'(F(x; D, s_{\text{pair}}))}{g'(F(x'; D, s_{\text{pair}}))} \\ F(x; D, s_{\text{pair}}) &= \mathbb{E}_{x \sim p} \left[\frac{\ell_+(s_{\text{pair}}(x, x)) + \ell_-(s_{\text{pair}}(x, X))}{2} \right]. \end{aligned}$$

$$\text{Push}_{\text{S}_{\text{pair}}^*}(D, \ell, g) = \mathbb{E}_{x \sim Q} \left[g(F(X'; D, s_{\text{pair}}^*)) \right].$$

Proof First, in the notation above,

$$\begin{aligned} R(\epsilon; \text{S}_{\text{pair}}^*, r_{\text{pair}}) &= \text{Push}(\text{S}_{\text{pair}}^* + \epsilon \cdot r_{\text{pair}}; D, \ell) \\ &= \mathbb{E}_{x \sim Q} \left[g(F(X'; D, s_{\text{pair}}^* + \epsilon \cdot r_{\text{pair}})) \right]. \end{aligned}$$

For fixed D , let $\mathcal{S}(D)$ denote the space of all Lebesgue-measurable pair-scorers $\text{S}_{\text{pair}} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, with addition and scalar multiplication defined pointwise, such that $\text{Push}(\text{S}_{\text{pair}}; D, \ell, g) < \infty$. As before, we consider the Gateaux variation of the functional. Pick any $\text{S}_{\text{pair}}^*, r_{\text{pair}} \in \mathcal{S}(D)$. For any $\epsilon > 0$, define

For simplicity, in the following we shall not explicitly write the dependence of F and G on $D, \text{S}_{\text{pair}}^*$. Now consider

$$\begin{aligned} R'(0; \text{S}_{\text{pair}}^*, r_{\text{pair}}) &= \mathbb{E}_{x \sim Q} \left[g'(F(X')) \cdot \mathbb{E}_{x \sim p} \left[r_{\text{pair}}(X, X') \cdot \frac{\ell'_{\text{S}_{\text{pair}}^*}(X, X')}{2} \right] \right] + \\ &\quad r_{\text{pair}}(X', X) \cdot \frac{\ell'_{\text{S}_{\text{pair}}^*}(X', X)}{2} \\ &= \frac{1}{2} \int_{\mathcal{X} \times \mathcal{X}} r_{\text{pair}}(x, x') \cdot (\rho(x)q(x') \cdot g'(F(x')) \cdot \ell'_{\text{S}_{\text{pair}}^*}(x, x') + \\ &\quad \rho(x')q(x) \cdot g'(F(x')) \cdot \ell'_{\text{S}_{\text{pair}}^*}(x', x)) dx dx', \end{aligned}$$

where as in the proof of Proposition 47, the interchange of derivative and expectation is justified when \mathcal{X} is finite, or when the derivatives $\ell'_{\text{S}_{\text{pair}}^*}, \ell''_{\text{S}_{\text{pair}}^*}$ are bounded.

For the optimal pair-scorer S_{pair}^* , the derivative must be zero for every r_{pair} . A sufficient condition for this to hold is that the second term in the integrand is zero for (almost) every $x, x' \in \mathcal{X}$.

Now, since ℓ is strictly proper composite, for any $\eta \in [0, 1]$, the solution to

$$\eta \cdot \ell'(s) + (1 - \eta) \cdot \ell'_{-1}(s) = 0$$

is $s = \Psi(\eta)$, by virtue of the above being the derivative of the conditional risk. Thus, the solution to

$$\frac{a}{a+b} \cdot \ell'(s) + \frac{b}{a+b} \cdot \ell'_{-1}(s) = 0$$

for $a, b > 0$ is $s = \Psi(a/(a+b)) = \Psi(\sigma(\log(a/b)))$. Letting

$$\begin{aligned} a &= g'(F(x')) \cdot \rho(x) \cdot q(x') \\ b &= g'(F(x)) \cdot q(x) \cdot \rho(x'), \end{aligned}$$

the optimal pair-scorer is, for every $x, x' \in \mathcal{X}$,

$$\begin{aligned} \text{S}_{\text{pair}}^*(x, x') &= \Psi \circ \sigma \circ \log \frac{\rho(x) \cdot q(x') \cdot g'(F(x'))}{\rho(x') \cdot q(x) \cdot g'(F(x))} \\ &= \Psi \circ \sigma \circ (\sigma^{-1}(\rho(x)) - \sigma^{-1}(\rho(x')) - G(D, s_{\text{pair}}^*)), \end{aligned}$$

where the second line is since

$$\frac{\rho(x)}{q(x)} = \frac{\eta(x)}{1 - \eta(x)} \cdot \frac{1 - \pi}{\pi}.$$

Thus,

$$\text{S}_{\text{pair}}^* = \Psi \circ \sigma \circ (\text{Diff}(\sigma^{-1} \circ \eta) - G(D, s_{\text{pair}}^*)).$$

The result follows by dividing through by the numerator. ■

Requiring differentiability of the loss means that we cannot compute the optimal solution for ℓ_{01} . However, we can compute the Bayes-optimal pair-scorers for a sequence of proper composite losses that approach ℓ_{01} . Consider a sequence of losses $\{\ell^{(p)}\}_{p \in \mathbb{N}}$ with corresponding links $\{\Psi^{(p)}\}_{p \in \mathbb{N}}$, with $(\forall v \in \mathbb{R}) \lim_{p \rightarrow \infty} (\Psi^{(p)})^{-1}(v) = [v > 0]$. This suggests the optimal scorer of

$$S_{\text{push}}^{\text{pair},*}(D, \ell_{01}, g) = \left\{ s_{\text{pair}}^* : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R} : \text{sign}(s_{\text{pair}}^*) = \text{sign}(\text{Diff}(\sigma^{-1} \circ \eta)) - G(D, s_{\text{pair}}^*) \right\},$$

but we defer a formal proof to future work.

When $g : x \mapsto x$, which corresponds to the standard ℓ -bipartite ranking risk, the term G above is $\equiv 0$ and so $s_{\text{pair}}^* = \Psi \circ \eta_{\text{pair}}$ as expected. For general (ℓ, g) , however, it is unclear how to simplify the term G any further. In general, s_{pair}^* appears to be a strictly monotone transform of η_{pair} , where the transform is distribution dependent. However, surprisingly, for the special case of ℓ being the exponential loss and $g : x \mapsto x^p$, the optimal scorer is explicitly determinable as a simple transform of the conditional probability.

Proposition 54 *Pick any distribution $D = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$. Let $\ell(y, v) = e^{-yv}$ be the exponential loss and $g : x \mapsto x^p$ for any $p \geq 1$. Then, the optimal pair-scorer s_{pair}^* for the (ℓ, g) -push bipartite ranking risk is*

$$s_{\text{pair}}^* = \frac{1}{p+1} \cdot \text{Diff}(\sigma^{-1} \circ \eta).$$

Proof We establish this by verifying that $s_{\text{pair}} = \frac{1}{p+1} \text{Diff}(\sigma^{-1} \circ \eta)$ satisfies the implicit equation in Equation 58. We begin with the term $F(x; D, s_{\text{pair}})$ as defined in Proposition 53. Plugging in $g : x \mapsto x^p$ and

$$s_{\text{pair}} = \frac{1}{p+1} \cdot \sigma^{-1} \circ \eta_{\text{pair}} = \frac{1}{p+1} \cdot \text{Diff}(\sigma^{-1} \circ \eta),$$

we get

$$\begin{aligned} (\forall x \in \mathcal{X}) F(x; D, s_{\text{pair}}) &= \mathbb{E}_{\mathcal{X} \times \mathcal{P}} \left[\frac{\ell_1(s_{\text{pair}}(X, x)) + \ell_{-1}(s_{\text{pair}}(x, X))}{2} \right] \\ &= \mathbb{E}_{\mathcal{X} \times \mathcal{P}} \left[\frac{e^{-s_{\text{pair}}(X, x)} + e^{s_{\text{pair}}(x, X)}}{2} \right] \\ &= \frac{1}{2} \mathbb{E}_{\mathcal{X} \times \mathcal{P}} \left[\left(\frac{\eta_{\text{pair}}(X, x)}{1 - \eta_{\text{pair}}(X, x)} \right)^{-1/(p+1)} + \left(\frac{\eta_{\text{pair}}(x, X)}{1 - \eta_{\text{pair}}(x, X)} \right)^{1/(p+1)} \right] \\ &= \mathbb{E}_{\mathcal{X} \times \mathcal{P}} \left[\exp(\sigma^{-1}(\eta(x)) - \sigma^{-1}(\eta(X))) / (p+1) \right] \\ &= \exp(\sigma^{-1}(\eta(x)) / (p+1)) \cdot \mathbb{E}_{\mathcal{X} \times \mathcal{P}} \left[\exp(-\sigma^{-1}(\eta(X))) / (p+1) \right], \end{aligned}$$

where crucially the dependence on η is separated from the dependence on the rest of the distribution.

Thus, for $g : x \mapsto x^p$,

$$(\forall x, x' \in \mathcal{X}) \frac{g'(F(x; D, s_{\text{pair}}))}{g'(F(x'; D, s_{\text{pair}}))} = \frac{\exp(\sigma^{-1}(\eta(x)) \cdot (p-1) / (p+1))}{\exp(\sigma^{-1}(\eta(x')) \cdot (p-1) / (p+1))}$$

with the result now a simple function of η , and

$$(\forall x, x' \in \mathcal{X}) \log \frac{g'(F(x; D, s_{\text{pair}}))}{g'(F(x'; D, s_{\text{pair}}))} = \frac{p-1}{p+1} \cdot (\sigma^{-1}(\eta(x)) - \sigma^{-1}(\eta(x'))),$$

Now recall that the link function for exponential loss is $\Psi = \frac{1}{2} \sigma^{-1}$. Plugging the above into the right hand side of Equation 58, we get

$$\begin{aligned} \Psi \circ \sigma \circ (\text{Diff}(\sigma^{-1} \circ \eta)) - G(D, s_{\text{pair}}^*) &= \left(\frac{1}{2} - \frac{p-1}{2(p+1)} \right) \cdot \text{Diff}(\sigma^{-1} \circ \eta) \\ &= \frac{1}{p+1} \cdot \text{Diff}(\sigma^{-1} \circ \eta) \\ &= s_{\text{pair}}^*. \end{aligned}$$

Therefore $s_{\text{pair}} = \frac{1}{p+1} \text{Diff}(\sigma^{-1} \circ \eta)$ satisfies the implicit equation of Proposition 53, and hence must be an optimal pair-scorer for exponential loss. ■

To see why exponential loss simplifies matters, we note that the risk can be decomposed into

$$\text{Push}(\text{Diff}(s); D, \exp, \cdot^p) = \left(\mathbb{E}_{\mathcal{X} \times \mathcal{P}} [e^{-s(X)}] \right)^p \cdot \left(\mathbb{E}_{\mathcal{X} \times \mathcal{Q}} [e^{p \cdot s(X')}] \right).$$

This decomposition into the product of two expectations simplifies the derivatives considerably. In fact, an alternate strategy to determine the minimisers of the risk is to consider

$$\arg \max_{s : \mathcal{X} \rightarrow \mathbb{R}} \mathbb{E}_{\mathcal{X} \times \mathcal{Q}} [e^{p \cdot s(X')}] : \left(\mathbb{E}_{\mathcal{X} \times \mathcal{P}} [e^{-s(X)}] \right)^p \leq C;$$

this is reminiscent of the Neyman-Pearson approach to arguing for the optimal scorers for the AUC (which incidentally is the strategy we shall employ for proving Proposition 56).

As with Proposition 47, we suspect the finiteness assumption on \mathcal{X} can be dropped, although we have been unsuccessful in establishing this. Nonetheless, for this special case, the optimal scorer can be expressed as $\frac{2}{p+1} \cdot \Psi \circ \eta_{\text{pair}}$, where Ψ is the link function corresponding to exponential loss; comparing this to the optimal pair-scorer for the standard bipartite risk (Equation 50), we see that the effect of the function $g : x \mapsto x^p$ is equivalent to slightly transforming the loss ℓ ; we will explore this more in the next section.

For other losses, the optimal pair-scorer appears to be a genuinely distribution specific transformation of η_{pair} , as we illustrate in Appendix I.

9.5.2 BAYES-OPTIMAL UNIVARIATE SCORERS

We now turn attention to computing $S_{\text{push}}^*(D, \ell, g)$. For ℓ_{01} , we were unsuccessful in computing the optimal pair-scorer; nonetheless, a different technique lets us establish the optimal univariate scorers. The basic observation is that the (ℓ_{01}, g) -push risk can be interpreted as the area under the parametric curve

$$\{ (\text{FPR}(t; D, s), g(\text{FNR}(t; D, s))) : t \in \mathbb{R} \},$$

which, compared to the ROC curve $\text{ROC}(s; D)$, transforms the true positive rates (or, equivalently, one minus the true negative rates) at each corresponding false positive rate. By manipulating the choice of g , the area under this curve can thus be focus more attention on certain ranges of false negative rates. The equivalence is formalised below.

Proposition 55 *Given any $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$, $g : \mathbb{R} \rightarrow \mathbb{R}$, and scorer $s : \mathcal{X} \rightarrow \mathbb{R}$ with differentiable ROC curve and invertible rates,*

$$\text{Push}(\text{Diff}(s); D, \ell_{01}, g) = \int_0^1 g(\text{FNR}(\text{FPR}^{-1}(a))) da.$$

Proof The proof follows how we established the 0-1 bipartite risk to an area under the curve (Proposition 21):

$$\begin{aligned} \text{Push}(\text{Diff}(s); D, \ell_{01}, g) &= \mathbb{E}_{X^{\sim}Q} \left[g(\text{FNR}(s(X))) \right] \\ &= \mathbb{E}_{X^{\sim}Q} \left[\int_{-\infty}^{\infty} \delta_{s(X)}(t) \cdot g(\text{FNR}(t)) dt \right] \\ &= \int_{-\infty}^{\infty} \mathbb{E}_{X^{\sim}Q} \left[\delta_{s(X)}(t) \cdot g(\text{FNR}(t)) \right] dt \\ &= \int_{-\infty}^{\infty} q_S(t) \cdot g(\text{FNR}(t)) dt \\ &= \int_{-\infty}^{\infty} -\text{FPR}'(t) \cdot g(\text{FNR}(t)) dt \text{ by Equation 22} \\ &= \int_0^1 g(\text{FNR}(\text{FPR}^{-1}(\alpha))) d\alpha. \end{aligned}$$

■

We can now establish the Bayes-optimal univariate scorers. (A similar result for the case of the reverse (ℓ, g) -push risk was shown in Cléménçon and Vayatis 2008, Proposition 7.)

Proposition 56 *Let g be a nonnegative, monotone increasing function. Given any $D = \langle M, \eta \rangle \in \Delta_{X \times \{\pm 1\}}$,*

$$\phi_{\text{opt}} \in \mathcal{S}_{\text{push}}^*(D, \ell_{01}, g),$$

for any strictly monotone increasing $\phi : [0, 1] \rightarrow \mathbb{R}$.

Proof Recall from Proposition 55 that the p -norm risk for the case of 0-1 loss is simply an area under the parametric curve

$$(\text{FPR}(t), g(\text{FNR}(t))) : t \in \mathbb{R} \cup \{\pm\infty\}.$$

Following the Neyman-Pearson approach to ROC maximisation (Proposition 76), maximisation of the 0-1 risk is thus equivalent to solving, for each $\alpha \in [0, 1]$

$$\underset{s : X \rightarrow \mathbb{R}}{\text{Argmin}} \quad g(\text{FNR}(t; D, s)) \text{ subject to } \text{FPR}(t; D, s) \leq \alpha.$$

Since g is a monotone increasing function, it preserves the optimal solution of the case of $g(x) = x$ (although potentially introducing new ones), which is the standard Neyman-Pearson problem. This means that for monotone increasing g , one family of optimal solutions is given by $s^* = \phi \circ \eta$, where ϕ is strictly monotone increasing. ■

Proposition 56 says that the (ℓ_{01}, g) -push objective is optimised by accurately recovering the entire ranked list. However, while they share the same optimal solution, the ordering over scorers induced by the (ℓ_{01}, g) -push risk is different from that induced by the standard bipartite ranking risk. This means that under misspecification or with finite samples, one will likely choose a different scorer by optimising the (ℓ_{01}, g) -push objective rather than the standard bipartite ranking objective. The examples in Rudin (2009) indicate that in many such cases, the solutions of the (ℓ_{01}, g) -push objective are superior to those of bipartite ranking at the head of the list.

The above trick does not work when we use a general proper composite loss ℓ , as we need to analyse a generalised Neyman-Pearson problem. However, for exponential loss and $g : x \mapsto x^p$, we can use the results of the previous section.

$$\mathcal{S}_{\text{push}}^*(D, \ell_{\text{exp}}, \eta) = \left\{ \frac{1}{p+1} \cdot (\sigma^{-1} \circ \eta) + b : b \in \mathbb{R} \right\}.$$

Proposition 57 *Pick any $D = \langle M, \eta \rangle \in \Delta_{X \times \{\pm 1\}}$. Let $\ell = \ell_{\text{exp}}$ and $p > 0$. Then, η^{opt} is finite.*

Proof By Proposition 54, the unique optimal pair-scorer is $s_{\text{Fair}}^* = \frac{1}{p+1} \cdot \text{Diff}(\sigma^{-1} \circ \eta) = \text{Diff}\left(\frac{1}{p+1}(\sigma^{-1} \circ \eta)\right)$, which is decomposable. Corollary 40 may be adapted here to argue that any optimal univariate scorer s^* must satisfy $s_{\text{Fair}}^* = \text{Diff}(s^*)$, and so $s^* = \frac{1}{p+1} \cdot (\sigma^{-1} \circ \eta) + b$ for some $b \in \mathbb{R}$. ■

For other losses, the optimal univariate scorer again appears to be distribution specific, as we illustrate in Appendix 1.

As before, the Bayes-optimal scorers for the p -norm push are closely related to those for appropriate proper composite losses (namely, those with link functions given by $\Psi = \frac{1}{p+1} \cdot \sigma$). We now study how the theory of proper composite losses suggests a recipe for constructing a family of alternate losses suitable for the ranking the best tasks.

9.6 Proper Composite Losses for Ranking the Best

Having studied the Bayes-optimality properties of the p -norm push, we now examine what this implies about the design of alternate proper composite losses for ranking the best. As shall be made precise, the p -norm push can be understood in terms of a suitable weight function over misclassification costs.

9.6.1 A WEIGHT FUNCTION PERSPECTIVE OF THE p -NORM PUSH

From Proposition 57, we see that changing p results in a scaling of the link function Ψ that is composed with η . Thus, the p -norm push has equivalent Bayes-optimal solutions, up to translation, as any strictly proper composite loss with the same link function $\Psi^{(p)} = \frac{1}{p+1} \cdot \sigma^{-1}$. One might then hope to understand the p -norm push risk by considering the risks corresponding to a family of proper composite losses $\{\ell^{(p)}\}_{p \in \mathbb{N}}$, where each member of the family comprises some fixed proper loss λ composed with an appropriately scaled sigmoidal link $\Psi^{(p)}$. However, for any $p > 0$, the resulting proper composite loss is

$$\ell^{(p)}(y, v) = \lambda(y, \Psi^{(p)}(v)) = \lambda(y, \sigma((p+1) \cdot v)) = \ell^{(0)}(y, (p+1)v).$$

That is, changing p simply scales the prediction space, and has no real impact on learning. This means that even on a finite sample, and with a restricted function class, the family of proper composite losses given by $\{\ell^{(p)}\}_{p \in \mathbb{N}}$ will have risks whose optimal solutions that are scalings of one another.

As with the ℓ_{01} case, this is not surprising. It merely indicates that the p -norm push risk must be understood in terms of its behaviour under a restricted function class or finite sample. Doing so requires that one move away from Bayes-optimal scorers, which assume access to infinite samples and an unrestricted function class. Our standard analysis based on the conditional risk thus cannot be applied.

Remarkably, it is possible to show that the p -norm push risk is equivalent to a specific proper composite risk even when minimising over a linear function class: Erekin and Rudin (2011, Theorem 1) shows that for a linear function class, the p -norm push risk with exponential loss is equivalent to the proper composite risk corresponding to the p -classification loss, defined by

$$(\forall v \in \mathbb{R}) \ell_{\text{pd}}(v; p) = \left(\frac{1}{p} \cdot e^{pv}, e^{-v} \right). \quad (59)$$

Interestingly, this loss is proper composite.

Lemma 58 For any $p > 0$, let $\ell = \ell_{\text{ped}}(\cdot; p)$ be the p -classification loss of Equation 59. Then, $\ell \in \mathcal{L}_{\text{SPC}}(\Psi^{(p)})$, where $(\Psi^{(p)})^{-1} : v \mapsto \sigma((p+1) \cdot v)$. Further, the underlying proper-loss $\lambda_{\text{ped}}(\cdot; p)$ is

$$(\forall u \in [0, 1]) \lambda_{\text{ped}}(u; p) = \left(\frac{1}{p} \cdot \left(\frac{u}{1-u} \right)^{1-\frac{1}{p+1}}, \left(\frac{1-u}{u} \right)^{\frac{1}{p+1}} \right).$$

Proof We can check that for $\Psi = \Psi^{(p)}$,

$$\begin{aligned} (\forall v \in \mathbb{R}) \Psi^{-1}(v) &= \frac{1}{1 - \frac{\ell(v)}{\sigma(v)}} \\ &= \frac{1}{1 + e^{-(p+1)v}} \\ &= \sigma((p+1) \cdot v), \end{aligned}$$

which is invertible, thus guaranteeing that ℓ is proper composite. It may be checked that the underlying proper loss is

$$\begin{aligned} (\forall u \in [0, 1]) \lambda_{\text{ped}}(u; p) &= \ell_{\text{ped}}(\Psi^{(p)}(u); p) \\ &= \left(\frac{1}{p} \cdot \left(\frac{u}{1-u} \right)^{1-\frac{1}{p+1}}, \left(\frac{1-u}{u} \right)^{\frac{1}{p+1}} \right). \end{aligned}$$

Given the loss ℓ_{ped} is proper composite, the agreement of the p -norm and p -classification risk minimisers is trivial in the unrestricted function class setting; however, it is not obvious in the linear class setting. The equivalence to p -classification is valuable, since we can analyse the proper composite loss to understand how it focusses accuracy at the head of the ranked list. We will do this by considering the corresponding weight function for the proper loss $\lambda = \lambda_{\text{ped}}(\cdot; p)$,

$$\begin{aligned} (\forall c \in (0, 1)) u_{\text{ped}}(c; p) &= -\frac{\lambda'(c)}{1-c} \text{ (by (Reid and Williamson, 2010), Theorem 1))} \\ &= \frac{1}{p+1} \cdot \frac{1}{c^{1+\frac{1}{p+1}} \cdot (1-c)^{2-\frac{1}{p+1}}}. \end{aligned}$$

This is a generalised version of the weight for the boosting loss (Table 3), which corresponds to $p = 1$.

The above weight function view has at least three benefits. First, given the equivalence of the p -classification and p -norm push risk, we have some insight as to how the latter encourages solutions to maximise accuracy at the head of the ranked list: as p increases, the loss is seen to place relatively more weight on larger values of c . That is, we pay attention to those instances with high η values, as accurate modelling of these is essential for determining the behaviour about the boundary $\eta(x) = c$.

Second, we can design normalised versions of the p -classification loss that have more interpretable behaviour when $p \rightarrow \infty$. Evidently, $u_{\text{ped}}(c; p)$ above tends to the trivial zero weight as $p \rightarrow \infty$, owing to the scaling factor of $(p+1)^{-1}$. Removing this scaling factor ensures that the weights are normalised for every $p > 0$, in the sense that $w(1/2; p) = 8$. Further, the resulting proper loss is easily verified to be

$$(\forall u \in [0, 1]) \lambda(u; p) = \left(\left(1 + \frac{1}{p} \right) \cdot \left(\frac{u}{1-u} \right)^{1-\frac{1}{p+1}}, (p+1) \cdot \left(\frac{1-u}{u} \right)^{\frac{1}{p+1}} - 1 \right),$$

with non-trivial limiting case as $p \rightarrow \infty$ of

$$(\forall u \in [0, 1]) \lambda(u; +\infty) = \left(\frac{u}{1-u}, -\log \frac{u}{1-u} \right).$$

When composed with the sigmoid link $\sigma(\cdot)$, this normalised family of proper losses results in the proper composite family

$$(\forall v \in \mathbb{R}) \ell(v; p) = \left(\left(1 + \frac{1}{p} \right) \cdot e^{\frac{p}{p+1}v}, (p+1) \cdot \left(e^{-\frac{v}{p+1}} - 1 \right) \right), \quad (60)$$

with non-trivial limiting case as $p \rightarrow \infty$ of

$$(\forall v \in \mathbb{R}) \ell(v; +\infty) = (e^v, -v).$$

Third, the weight function view suggests a scheme of *designing new losses* for ranking the best, by constructing appropriate weight functions emphasising large values of η . We now pursue this idea.

9.6.2 STRICT PROPERNESS AND q -RTB LOSSES

We now study the design of q -RTB losses based on the theory of proper composite losses. We begin with the simple observation that any strictly proper composite loss is a q -RTB loss for every $q \in [0, 1]$, by virtue of choosing the corresponding link function Ψ in Equation 52. More generally, the set of q -RTB losses is exactly the set of proper composite losses ℓ for which $\lambda = \ell \circ \Psi^{-1}$ is strictly proper on the interval $\eta \in [q, 1]$ and (not necessarily strictly) proper on the interval $[0, q]$. This suggests a simple recipe for designing q -RTB losses; however, as we shall see, not enforcing strict properness poses computational challenges.

A key difficulty in designing q -RTB losses is the following. Suppose λ is a proper loss that is not strictly proper on an interval $I \subseteq [0, 1]$. Then, λ is non-convex, and more importantly, cannot be made convex via a link function. To see this, recall that the canonical link function Ψ for a proper loss λ is the function for which $\lambda \circ \Psi^{-1}$ has the largest modulus of convexity. The weight function w of λ is related to the canonical link function Ψ by $w = \Psi'$. As λ is not strictly proper on I , we must have that $w \equiv 0$ on I . But then Ψ must be constant on I , and hence not invertible. Thus, to maintain convexity, it is essential to maintain strictness of the proper composite loss.

As a simple example, suppose λ is some strictly proper loss with weight function w . Now for some $q > 0$, consider the loss $\lambda_{\text{RTB}}(\cdot; q)$ with weight function

$$u_{\text{RTB}}(c; q) = \mathbb{I}[c \geq q] \cdot w(c). \quad (61)$$

This loss is not strictly proper on the interval $[0, q]$, and is strict on $[q, 1]$. Therefore, the loss aims to accurately model instances x for which $\eta(x) \geq q$. We can explicitly compute the partial losses for $\lambda_{\text{RTB}}(\cdot; q)$ as follows.

Lemma 59 Pick any proper loss λ with weight function w . For any $q > 0$, let $w_{\text{RTB}}(c; q)$ be the weight function given by Equation 61. Then, this weight has corresponding proper loss

$$\begin{aligned} (\forall u \in [0, 1]) \lambda_{\text{RTB}}(-1, u; q) &= \lambda_{-1}(u) - \lambda_{-1}(u \wedge q) \\ \lambda_{\text{RTB}}(+1, u; q) &= \lambda_1(u \vee q). \end{aligned}$$

Proof By Shuford's integral representation (Equation 10),

$$\begin{aligned} \lambda_{\text{RTB}}(-1, u; q) &= \int_0^1 \mathbb{I}[c < u] \cdot \mathbb{I}[c \geq q] \cdot c \cdot w(c) dc \\ &= \int_0^1 \mathbb{I}[c < u] \cdot c \cdot w(c) dc - \int_0^1 \mathbb{I}[c < u] \cdot \mathbb{I}[c < q] \cdot c \cdot w(c) dc \end{aligned}$$

$$\begin{aligned}
&= \int_0^1 \llbracket c < u \rrbracket \cdot c \cdot w(c) dc - \int_0^1 \llbracket c < u \wedge q \rrbracket \cdot c \cdot w(c) dc \\
&= \lambda_{-1}(u) - \lambda_{-1}(u \wedge q). \\
\lambda_{\text{RTB}(q)}(+1, u; q) &= \int_0^1 \llbracket c > u \rrbracket \cdot \llbracket c \geq q \rrbracket \cdot (1 - c) \cdot w(c) dc \\
&= \int_0^1 \llbracket c > u \vee q \rrbracket \cdot (1 - c) \cdot w(c) dc \\
&= \lambda_1(u \vee q).
\end{aligned}$$

When the prediction $u \geq q$, the partial losses of $\lambda_{\text{RTB}}(\cdot; q)$ are unchanged from those of λ , barring a translation for λ_{-1} . However, when $u < q$, the partial loss for the positive class plateaus, whereas the partial loss for the negative class drops to zero. The resulting loss is clearly non-convex, and further, *no* invertible link function can be applied to make it convex: composing $\lambda_{\text{RTB}}(\cdot; q)$ with an invertible link function Ψ yields a loss ℓ_{RTB} with partial losses

$$\begin{aligned}
\ell_{\text{RTB}}(-1, v; q) &= \ell_{-1}(v) - \ell_{-1}(v \wedge \Psi(q)) \\
\ell_{\text{RTB}}(+1, v; q) &= \ell_1(v \vee \Psi(q)).
\end{aligned}$$

which are not convex for any choice of Ψ .

A natural alternative to a loss that is not strictly proper is one that is “nearly” so, i.e. one whose weight function w is close to, but never exactly 0. However, this must be done with the following fact in mind: for any $\alpha > 0$, the proper loss with scaled weight function $\alpha \cdot w$ is simply the scaled loss $\alpha \cdot \lambda$. Thus, uniformly scaling a weight function does not affect the strict properness of the underlying loss. Scaling a loss on an interval $I \subset [0, 1]$ will however induce a qualitatively different loss: minimally, the new loss will be asymmetric, and have a non-trivially different set of Bayes-optimal scorers compared to the original loss. We now explore how proper composite losses can be designed to approximate a q -RTB loss.

9.6.3 PROPER COMPOSITE q -RTB SURROGATES

Our basic recipe for generating a q -RTB loss will be to combine the weight functions for two existing losses. Specifically, let w_ν, w_μ be weight functions corresponding to proper losses ν, μ . We assume that w_μ grows faster near 1 than w_ν does near 0, i.e. $\lim_{\alpha \rightarrow -1} \frac{w_\mu(\alpha)}{w_\nu(1-\alpha)} > 1$. (We will typically be interested in the case where the limit is $+\infty$.) We now consider a hybrid weight function of the form

$$\begin{aligned}
(\forall c \in (0, 1)) w(c; q) &= \begin{cases} w_\nu(c) & \text{if } c < q \\ \alpha(q) \cdot w_\mu(c) & \text{if } c \geq q \end{cases} \\
&= w_\nu(c) \cdot \llbracket c < q \rrbracket + \alpha(q) \cdot w_\mu(c) \cdot \llbracket c \geq q \rrbracket,
\end{aligned} \tag{62}$$

where $\alpha(q) = \frac{w_\nu(q)}{w_\mu(q)}$ so that there is no discontinuity at $c = q$.

Since $w(c; q)$ is the sum of two weights, we can compute the corresponding proper losses for each component to get the form of the corresponding proper loss.

Lemma 60 *Pick any weight functions $w_\nu, w_\mu : [0, 1] \rightarrow \mathbb{R}_+$ with corresponding proper losses ν, μ . For any $q > 0$, let $w(\cdot; q)$ be as per Equation 62. Then, the proper loss corresponding to this weight is*

$$\begin{aligned}
(\forall u \in [0, 1]) \bar{\lambda}_{-1}(u; q) &= \alpha(q) \cdot \mu_{-1}(u) + \nu_{-1}(u \wedge q) - \alpha(q) \cdot \mu_{-1}(u \wedge q) \\
\bar{\lambda}_1(u; q) &= \nu_1(u) + \alpha(q) \cdot \mu_1(u \vee q) - \nu_1(u \vee q),
\end{aligned} \tag{63}$$

where $\alpha(q) = \frac{w_\nu(q)}{w_\mu(q)}$.

Proof By Lemma 59, the weight $w_\nu(c) \cdot \llbracket c < q \rrbracket = w_\nu(c) - w_\nu(c) \cdot \llbracket c \geq q \rrbracket$ corresponds to the proper loss

$$\bar{\nu}(u) = (\nu_{-1}(u \wedge q), \nu_1(u) - \nu_1(u \vee q))$$

while the weight $\alpha(q) \cdot w_\mu(c) \cdot \llbracket c \geq q \rrbracket$ corresponds to the proper loss

$$\bar{\mu}(u) = (\alpha(q) \cdot \mu_{-1}(u) - \alpha(q) \cdot \mu_{-1}(u \wedge q), \alpha(q) \cdot \mu_1(u \vee q)).$$

Thus, the weight w corresponds to the sum of these losses, which is of the given form. \blacksquare

We may similarly combine proper composite losses corresponding to the underlying weights into a continuous proper composite loss corresponding to the weight $w(\cdot; q)$:

Lemma 61 *Suppose that proper losses ν, μ have corresponding proper composite losses $\bar{\nu}, \bar{\mu}$ using invertible link functions Ψ, Φ . For any $q > 0$, let $w(\cdot; q)$ be as per Equation 62. Then, $w(\cdot; q)$ has corresponding proper composite loss ℓ with components*

$$\begin{aligned}
(\forall v \in \mathbb{R}) \bar{\ell}_{-1}(v; q) &= \begin{cases} \rho_{-1}(v) & \text{if } v < v_0(q) \\ \bar{\kappa}_{-1}(v; q) + \rho_{-1}(v_0) - \bar{\kappa}_{-1}(v_0; q) & \text{else} \end{cases} \\
\bar{\ell}_1(v; q) &= \begin{cases} \rho_1(v) + \bar{\kappa}_1(v_0; q) - \rho_1(v_0) & \text{if } v < v_0(q) \\ \bar{\kappa}_1(v; q) & \text{else,} \end{cases}
\end{aligned}$$

where $v_0(q) = \Psi(q)$, and

$$\begin{aligned}
\bar{\kappa}(y; v; q) &= \alpha(q) \cdot \kappa \left(y, \frac{v - \beta(q)}{\gamma(q)} \right) \\
\beta(q) &= \Psi(q) - \gamma(q) \cdot \Phi(q) \\
\gamma(q) &= \frac{\Psi'(q)}{\Phi'(q)}.
\end{aligned}$$

Proof By definition, we have

$$\begin{aligned}
\rho(y; v) &= \nu(y, \Psi^{-1}(v)) \\
\kappa(y; v) &= \mu(y, \Phi^{-1}(v)).
\end{aligned}$$

Given any $q > 0$, we will construct a proper composite loss using the proper loss $\bar{\lambda}(\cdot; q)$ of Equation 63, composed with a link function Π that is a suitable combination of Ψ and Φ . The technical detail to attend to is to ensure there are no discontinuities with the resulting loss.

First, to ensure that the link for the two pieces of $\bar{\lambda}$ coincide, with the same derivative at the threshold q , we modify the link for the second loss to

$$\bar{\Phi}(c; q) \doteq \gamma(q) \cdot \Phi(c) + \beta(q),$$

where

$$\begin{aligned}
\beta(q) &= \Psi(q) - \gamma(q) \cdot \Phi(q) \\
\gamma(q) &= \frac{\Psi'(q)}{\Phi'(q)}.
\end{aligned}$$

Note that if $\Psi'(q) = \Phi'(q)$, this simplifies to the translated link $\Phi(c) + \Psi(q) - \Phi(q)$. This modified link has inverse

$$\tilde{\Phi}^{-1}(v; q) = \Phi^{-1}\left(\frac{v - \beta(q)}{\gamma(q)}\right).$$

Now define the hybrid link

$$\Pi(c; q) = \llbracket c < q \rrbracket \cdot \Psi(c) + \llbracket c \geq q \rrbracket \cdot \tilde{\Phi}(c; q)$$

with corresponding inverse

$$\Pi^{-1}(v; q) = \llbracket v < v_0(q) \rrbracket \cdot \Psi^{-1}(v) + \llbracket v \geq v_0(q) \rrbracket \cdot \tilde{\Phi}^{-1}(v; q)$$

for $v_0(q) = \Psi(q) = \tilde{\Phi}(q; q)$. Then, the proper loss corresponding to \tilde{w} can be made proper composite, with

$$\begin{aligned} \tilde{\ell}(-1, v; q) &= \bar{\kappa}_{-1}(v; q) + \rho_{-1}(v \wedge v_0) - \bar{\kappa}_{-1}(v \wedge v_0; q) \\ \tilde{\ell}(+1, v; q) &= \rho_1(v) + \bar{\kappa}_1(v \vee v_0; q) - \rho_1(v \vee v_0), \end{aligned}$$

where

$$\begin{aligned} \bar{\kappa}(y, v; q) &= \alpha(q) \cdot \mu(y; (\tilde{\Phi}^{-1}(v))) \\ &= \alpha(q) \cdot \kappa\left(y, \frac{v - \beta(q)}{\gamma(q)}\right). \end{aligned}$$

The basic idea of the loss ℓ' in Equation 63 is intuitive: one switches between choosing one of the underlying losses based on some threshold on the scores. The only additional ingredient is that scaling and translating one of the losses to ensure continuity of the end result. We provide examples that illustrate the basic idea.

- Consider the weight function

$$w(c) = \begin{cases} \frac{1}{c^{1-\alpha}} & \text{if } c < q \\ \frac{2\sqrt{q(1-q)}}{c^{3/2}(1-c)^{3/2}} & \text{if } c \geq q, \end{cases}$$

which is a hybrid of the weights for logistic and exponential loss. Using the sigmoid link yields the loss (for $q = \frac{1}{2}$)

$$\ell(v) = \begin{cases} \log(1 + e^v) & \text{if } v < 0 \\ e^{v/2} + \log 2 - 1 & \text{if } v \geq 0 \end{cases}, \begin{cases} \log(1 + e^{-v}) - \log 2 + 1 & \text{if } v < 0 \\ e^{-v/2} & \text{if } v \geq 0 \end{cases}.$$

- Consider the weight function

$$w(c) = \begin{cases} 4 & \text{if } c < q \\ \frac{2\sqrt{q(1-q)}}{c^{3/2}(1-c)^{3/2}} & \text{if } c \geq q, \end{cases}$$

which is a hybrid of the weights for square and exponential loss. Using an appropriate hybrid of the identity and sigmoid link yields the loss (for $q = \frac{1}{2}$)

$$\ell(v) = \begin{cases} \frac{1}{2} \cdot (1 + v)^2 & \text{if } v < 0 \\ e^{v/2} - \frac{1}{2} & \text{if } v \geq 0 \end{cases}, \begin{cases} \frac{1}{2} \cdot (1 - v)^2 + \frac{1}{2} & \text{if } v < 0 \\ e^{-v/2} & \text{if } v \geq 0 \end{cases}.$$

- Consider for $p > 0$ the family of weights

$$w(c; p) = \frac{1}{c \cdot (1 - c)^{2 - \frac{1}{p}}},$$

which are similar to those employed by p -classification, except that the behaviour near $c = 0$ is fixed, and does not vary with p . Though not explicitly a hybrid, the weight is asymmetric, and thus the role of p is to tune the degree of focus on large values of η . For example, when $p = 1$, with the sigmoid link we have the proper composite loss

$$\ell(v) = \left(\frac{2}{\sqrt{\sigma(-v)}}, 2 \tanh^{-1}(\sqrt{\sigma(-v)}) \right).$$

As another example, when $p = 3$, we get

$$\ell(v) = \left(\frac{4}{3\sigma(-v)^{3/4}}, 2 \tanh^{-1}((\sigma(-v))^{1/4}) + 2 \tanh^{-1}((\sigma(-v))^{1/4}) \right).$$

It is clear that the above recipe can be applied for any suitable combination of weight functions, which we have argued to focus attention at the head of the ranked list. How do we choose amongst several such candidate weight functions? Put another way, can we characterise which hybrid weight function is the “best”? Answering such a question requires a precise sense in which one loss is “better” than another. This issue is only superficially simple, as even in binary classification for example, one cannot expect any given surrogate to be uniformly superior to all others in terms of resulting misclassification error (Reid and Williamson, 2010, Appendix A). Nonetheless, relating for example the weight function of a proper loss to generalisation ability in terms of a performance measure such as PTop would be of interest.

We emphasise also that the above represents just one recipe for generating suitable proper composite losses. If one can generate a suitable parametrised family of weights and link function generating a convex loss (e.g. the scaled p -classification loss of Equation 60), these would also be suitable for ranking the best problems.

9.7 Experiments with Proper Composite Losses for Ranking the Best

We present experiments that assess the efficacy of several proper composite losses proposed in the previous section for the problem of maximising accuracy at the head of the ranked list. The aim of our experiments is *not* to position the new losses as a superior alternative to the existing p -classification and p -norm push approaches. Rather, we wish to demonstrate that the proper composite interpretation gives one way of generating a family of losses for this problem, with the p -classification loss being but one example of this family. An attraction of these losses is that they are simple to optimise using gradient-based methods, with complexity linear in the number of training examples (as opposed to methods that operate on pairs of examples).

To clarify the effect of the choice of loss and choice of risk, we consider all combinations of the three risk types considered in this paper—proper composite (Equation 14), bipartite (Equation 17), and p -norm push (Equation 56)—and the loss functions of interest. On the one hand, one expects the p -norm push risk to perform best when combined with a loss suitable for ranking the best. On the other hand, our analysis in the previous section indicates that there is promise in the minimisation of a suitable proper composite risk.

For our losses, we experiment with the standard logistic and exponential losses, as well as the p -classification loss. Based on our hybrid loss proposal in Lemma 61, we consider the following:

- The proper composite loss with weight $w(c) = \frac{1}{c^{2-\frac{1}{p}}}$, and sigmoid link, which we term the “Log-classification Hybrid”;
- The proper composite loss with weight being a hybrid of $\frac{1}{c(1-c)}$ and $\frac{1}{2 \cdot e^{v/2} \cdot (1-c)^{3/2}}$ about threshold $\frac{1}{p+1}$, and sigmoid link, which we term the “Log-Exp Hybrid”;

- The proper composite loss with weight being a hybrid of 4 and $\frac{1}{2 \cdot \sigma^{1/2} \cdot (1 - \sigma)^{1/2}}$ about threshold $\frac{1}{p+1}$ and link being a hybrid of the identity and sigmoid link, which we term the ‘‘Square-Exp Hybrid’’.

We compare these methods on four UCI data sets: ionosphere, housing, german and car. Each method was trained with a regularised linear model, where the training objective was minimised using L-BFGS (Nocedal and Wright, 2006, pg. 177). For each data set, we created 5 random train-test splits in the ratio 2 : 1. For each split, we performed 5-fold cross-validation on the training set to tune the strength of regularisation $\lambda \in \{10^{-6}, 10^{-5}, \dots, 10^2\}$, and where appropriate the constant 22 $p \in \{1, 2, 4, 8, 16, 32, 64\}$. We then evaluated performance on the test set, and report the average across all splits. As performance measures, we used the AUC, ARR, DCG, AP, and PTop (Agarwal, 2011; Boyd et al., 2012). For all measures, a higher score is better. Parameter tuning was done based on the AP on the test folds.

The results are summarised in Tables 10–13, with the average ranks of each method with respect to each metric summarised in Table 14. No single method clearly outperforms all others in all metrics. However, we observe that the candidate proper composite losses are very competitive with the p -classification loss—the ‘‘Log-exp hybrid’’ and ‘‘Square-exp’’ hybrid in particular consistently perform comparably, and often better than p -classification. We especially find that the newly proposed proper composite losses perform well even when used as a surrogate loss as part of the bipartite risk. This confirms that the weight function perspective of the p -classification loss, and thus the p -norm push, is potentially practically useful for the design of losses suitable for ranking the best.

9.8 Existing Work

Clemenson and Vayatis (2007) identified two subproblems in ranking the best instances. The first problem is determining which instances qualify as the best. The second problem is ranking amongst these identified best instances. The first problem can be thought of as simply recovering an appropriate level set of $\eta(x)$, without determining the specific $\eta(x)$ value, i.e. we simply wish to discover

$$\{x \in \mathcal{X} : \eta(x) \geq q\}.$$

When q is fixed, this can be solved by reducing the problem to cost-sensitive classification (Scott andavenport, 2007). More generally, Clemenson and Vayatis (2007) considered the setting where q depends on the quantile of the scoring function. This poses challenges for analysis and estimation. The quantile version of the problem has been studied theoretically by Clemenson and Vayatis (2007), and (Boyd et al., 2012) gave a practical convex optimisation solution for the case of hinge loss. In both cases, the threshold q was specified as a quantile of the η .

The problem of ranking amongst the best instances with a quantile-based threshold was studied theoretically by Clemenson and Vayatis (2007), who proposed that the optimal univariate scoring function here must satisfy

$$s^*(x) \in \begin{cases} \{\eta(x)\} & \text{if } \eta(x) \geq q_p \\ [0, q_p] & \text{if } \eta(x) < q_p \end{cases}$$

Observe that this is identical to our Equation 52, except that q is now a function of η . They showed that two ‘‘local’’ versions of the AUC criterion, one of which is related to the partial AUC mentioned earlier, are optimised by this scorer. To our knowledge, our analysis in terms of proper losses for the simpler case where q is a fixed constant has not been done before.

Ertekin and Rudin (2011, Theorem 1) showed that for the case of a linear hypothesis class and $p \geq 1$, the Bayes optimal scorer for the p -norm push coincides with the classification risk for the asymmetric p -classification loss function of Equation 59. The optimal scorer for this loss is easily checked to be $s^* = \frac{1}{p+1} \cdot (\sigma^{-1} \circ \eta)$, and so in the unrestricted hypothesis class setting, the result agrees with ours. Our result is

22. We have observed that the parameter p selected by cross-validation may not necessarily correspond to the one that gives best test set performance, possibly a result of the limited sizes of the data sets in consideration. Treating each choice of p as resulting in a separate loss might therefore reveal slightly different rankings of the (loss, risk) combinations we consider.

Method	AUC	ARR	DCG	AP	PTop
Proper Logistic	0.9113 ± 0.0208 (15)	0.0583 ± 0.0056 (10)	0.2192 ± 0.0050 (12)	0.9243 ± 0.0339 (15)	13.0000 ± 17.0880 (9)
Proper Exponential	0.9128 ± 0.0166 (14)	0.0585 ± 0.0056 (9)	0.2193 ± 0.0050 (11)	0.9262 ± 0.0318 (14)	12.8000 ± 12.9499 (10)
Proper P-Classification	0.9152 ± 0.0160 (9)	0.0598 ± 0.0053 (5)	0.2207 ± 0.0045 (6)	0.9349 ± 0.0232 (8)	11.6000 ± 8.8487 (12)
Proper Log- p -classification Hybrid	0.9034 ± 0.0220 (16)	0.0606 ± 0.0021 (3)	0.2208 ± 0.0020 (5)	0.9236 ± 0.0194 (16)	8.4000 ± 5.4129 (13)
Proper Log-Exp Hybrid	0.9240 ± 0.0180 (2)	0.0601 ± 0.0054 (4)	0.2211 ± 0.0046 (4)	0.9430 ± 0.0263 (2)	16.2000 ± 13.7004 (3)
Proper Square-Exp Hybrid	0.9153 ± 0.0110 (8)	0.0601 ± 0.0052 (4)	0.2211 ± 0.0041 (4)	0.9395 ± 0.0191 (4)	16.8000 ± 10.5688 (2)
Bipartite Logistic	0.9157 ± 0.0195 (6)	0.0587 ± 0.0057 (8)	0.2197 ± 0.0049 (10)	0.9316 ± 0.0315 (10)	14.8000 ± 15.1228 (5)
Bipartite Exponential	0.9149 ± 0.0149 (11)	0.0590 ± 0.0053 (7)	0.2198 ± 0.0046 (9)	0.9292 ± 0.0292 (13)	13.0000 ± 12.7475 (9)
Bipartite P-Classification	0.9151 ± 0.0287 (10)	0.0575 ± 0.0077 (11)	0.2188 ± 0.0070 (13)	0.9294 ± 0.0361 (12)	15.6000 ± 14.6731 (4)
Bipartite Log- p -classification Hybrid	0.9207 ± 0.0131 (3)	0.0612 ± 0.0027 (2)	0.2218 ± 0.0028 (2)	0.9407 ± 0.0172 (3)	16.8000 ± 14.2373 (2)
Bipartite Log-Exp Hybrid	0.9166 ± 0.0160 (5)	0.0596 ± 0.0055 (6)	0.2205 ± 0.0046 (7)	0.9341 ± 0.0277 (9)	14.6000 ± 13.1643 (6)
Bipartite Square-Exp Hybrid	0.9284 ± 0.0273 (1)	0.0618 ± 0.0025 (1)	0.2227 ± 0.0025 (1)	0.9522 ± 0.0281 (1)	28.8000 ± 19.8293 (1)
P-Norm Logistic	0.9129 ± 0.0182 (13)	0.0596 ± 0.0058 (6)	0.2204 ± 0.0050 (8)	0.9314 ± 0.0292 (11)	14.0000 ± 11.6833 (7)
P-Norm Exponential	0.9154 ± 0.0147 (7)	0.0598 ± 0.0053 (5)	0.2207 ± 0.0044 (6)	0.9354 ± 0.0222 (7)	12.0000 ± 9.5131 (11)
P-Norm P-Classification	0.9152 ± 0.0287 (9)	0.0575 ± 0.0077 (11)	0.2188 ± 0.0070 (13)	0.9294 ± 0.0362 (12)	15.6000 ± 14.6731 (4)
P-Norm Log- p -classification Hybrid	0.7893 ± 0.0618 (17)	0.0475 ± 0.0026 (12)	0.2018 ± 0.0031 (14)	0.8215 ± 0.0589 (17)	1.2000 ± 1.6432 (14)
P-Norm Log-Exp Hybrid	0.9167 ± 0.0167 (4)	0.0598 ± 0.0055 (5)	0.2207 ± 0.0047 (6)	0.9358 ± 0.0264 (6)	13.4000 ± 11.4586 (8)
P-Norm Square-Exp Hybrid	0.9144 ± 0.0122 (12)	0.0612 ± 0.0024 (2)	0.2217 ± 0.0023 (3)	0.9361 ± 0.0152 (5)	13.0000 ± 9.6177 (9)

Table 10: Results of various ‘‘ranking the best’’ methods on ionosphere data set.

Table 11: Results of various "ranking the best" methods on housing data set.

Method	AUC	ARR	DCG	AP	P _{Top}
Proper Logistic	0.7597 ± 0.0415 (2)	0.0438 ± 0.0179 (10)	0.2068 ± 0.0209 (7)	0.1490 ± 0.0623 (8)	0.0000 ± 0.0000 (3)
Proper Exponential	0.7563 ± 0.0824 (3)	0.0625 ± 0.0580 (5)	0.2213 ± 0.0441 (2)	0.1762 ± 0.0752 (1)	0.0000 ± 0.0000 (3)
Proper P-Classification	0.7344 ± 0.0964 (10)	0.0364 ± 0.0125 (15)	0.1991 ± 0.0198 (15)	0.1404 ± 0.0628 (13)	0.0000 ± 0.0000 (3)
Proper Log-p-classification Hybrid	0.7254 ± 0.1002 (15)	0.0424 ± 0.0190 (11)	0.2037 ± 0.0245 (10)	0.1423 ± 0.0689 (11)	0.0000 ± 0.0000 (3)
Proper Log-Exp Hybrid	0.7785 ± 0.0461 (1)	0.0402 ± 0.0135 (12)	0.2045 ± 0.0172 (9)	0.1490 ± 0.0578 (8)	0.0000 ± 0.0000 (3)
Proper Square-Exp Hybrid	0.7498 ± 0.0729 (5)	0.0402 ± 0.0205 (12)	0.2021 ± 0.0241 (12)	0.1429 ± 0.0682 (10)	0.0000 ± 0.0000 (3)
Bipartite Logistic	0.7280 ± 0.1085 (13)	0.0616 ± 0.0589 (6)	0.2187 ± 0.0471 (5)	0.1707 ± 0.0839 (5)	0.4000 ± 0.8944 (1)
Bipartite Exponential	0.7306 ± 0.0882 (11)	0.0652 ± 0.0578 (1)	0.2222 ± 0.0466 (1)	0.1740 ± 0.0837 (3)	0.4000 ± 0.8944 (1)
Bipartite P-Classification	0.7282 ± 0.0889 (12)	0.0627 ± 0.0586 (4)	0.2198 ± 0.0471 (3)	0.1704 ± 0.0841 (6)	0.4000 ± 0.8944 (1)
Bipartite Log-p-classification Hybrid	0.7382 ± 0.0711 (7)	0.0585 ± 0.0512 (7)	0.2170 ± 0.0398 (6)	0.1645 ± 0.0666 (7)	0.2000 ± 0.4472 (2)
Bipartite Log-Exp Hybrid	0.7547 ± 0.0710 (4)	0.0636 ± 0.0578 (2)	0.2222 ± 0.0445 (1)	0.1760 ± 0.0760 (2)	0.4000 ± 0.8944 (1)
Bipartite Square-Exp Hybrid	0.7273 ± 0.1094 (14)	0.0632 ± 0.0590 (3)	0.2195 ± 0.0486 (4)	0.1709 ± 0.0874 (4)	0.4000 ± 0.8944 (1)
P-Norm Logistic	0.6987 ± 0.1159 (17)	0.0317 ± 0.0129 (17)	0.1913 ± 0.0213 (17)	0.1190 ± 0.0501 (17)	0.0000 ± 0.0000 (3)
P-Norm Exponential	0.7377 ± 0.0691 (8)	0.0440 ± 0.0185 (9)	0.2054 ± 0.0233 (8)	0.1442 ± 0.0621 (9)	0.0000 ± 0.0000 (3)
P-Norm P-Classification	0.7495 ± 0.0632 (6)	0.0368 ± 0.0132 (14)	0.1998 ± 0.0195 (13)	0.1414 ± 0.0609 (12)	0.0000 ± 0.0000 (3)
P-Norm Log-p-classification Hybrid	0.7354 ± 0.0834 (9)	0.0348 ± 0.0116 (16)	0.1970 ± 0.0183 (16)	0.1354 ± 0.0591 (15)	0.0000 ± 0.0000 (3)
P-Norm Log-Exp Hybrid	0.6875 ± 0.1557 (18)	0.0469 ± 0.0316 (8)	0.2023 ± 0.0206 (11)	0.1353 ± 0.0545 (16)	0.2000 ± 0.4472 (2)
P-Norm Square-Exp Hybrid	0.7055 ± 0.1290 (16)	0.0400 ± 0.0162 (13)	0.1996 ± 0.0215 (14)	0.1364 ± 0.0657 (14)	0.0000 ± 0.0000 (3)

Table 12: Results of various "ranking the best" methods on german data set.

Method	AUC	ARR	DCG	AP	P _{Top}
Proper Logistic	0.8121 ± 0.0285 (7)	0.0393 ± 0.0040 (4)	0.1870 ± 0.0040 (5)	0.6236 ± 0.0637 (7)	2.4000 ± 1.9494 (4)
Proper Exponential	0.8131 ± 0.0311 (3)	0.0372 ± 0.0047 (13)	0.1855 ± 0.0040 (12)	0.6218 ± 0.0677 (10)	1.8000 ± 2.0494 (7)
Proper P-Classification	0.8115 ± 0.0282 (9)	0.0389 ± 0.0048 (7)	0.1867 ± 0.0038 (7)	0.6226 ± 0.0621 (9)	2.4000 ± 2.3022 (4)
Proper Log-p-classification Hybrid	0.8103 ± 0.0278 (13)	0.0407 ± 0.0039 (1)	0.1883 ± 0.0036 (1)	0.6285 ± 0.0576 (1)	3.4000 ± 2.3022 (1)
Proper Log-Exp Hybrid	0.8111 ± 0.0290 (10)	0.0379 ± 0.0040 (12)	0.1860 ± 0.0040 (11)	0.6205 ± 0.0666 (14)	2.0000 ± 2.0000 (6)
Proper Square-Exp Hybrid	0.8086 ± 0.0322 (16)	0.0391 ± 0.0037 (6)	0.1867 ± 0.0040 (7)	0.6188 ± 0.0661 (15)	2.2000 ± 1.7889 (5)
Bipartite Logistic	0.8136 ± 0.0299 (2)	0.0382 ± 0.0043 (11)	0.1863 ± 0.0043 (9)	0.6233 ± 0.0682 (8)	2.6000 ± 2.7019 (3)
Bipartite Exponential	0.8118 ± 0.0268 (8)	0.0393 ± 0.0037 (4)	0.1870 ± 0.0038 (5)	0.6216 ± 0.0631 (11)	2.4000 ± 1.9494 (4)
Bipartite P-Classification	0.8131 ± 0.0298 (3)	0.0393 ± 0.0037 (4)	0.1871 ± 0.0038 (4)	0.6245 ± 0.0657 (4)	2.2000 ± 1.6432 (5)
Bipartite Log-p-classification Hybrid	0.8101 ± 0.0296 (14)	0.0401 ± 0.0036 (3)	0.1878 ± 0.0036 (3)	0.6279 ± 0.0637 (2)	2.2000 ± 1.3038 (5)
Bipartite Log-Exp Hybrid	0.8138 ± 0.0311 (1)	0.0384 ± 0.0048 (10)	0.1864 ± 0.0039 (8)	0.6244 ± 0.0664 (5)	2.2000 ± 2.1679 (5)
Bipartite Square-Exp Hybrid	0.8127 ± 0.0304 (5)	0.0387 ± 0.0052 (8)	0.1867 ± 0.0042 (7)	0.6240 ± 0.0640 (6)	2.4000 ± 2.0736 (4)
P-Norm Logistic	0.8129 ± 0.0296 (4)	0.0392 ± 0.0049 (5)	0.1870 ± 0.0038 (5)	0.6240 ± 0.0639 (6)	2.8000 ± 2.5884 (2)
P-Norm Exponential	0.8105 ± 0.0277 (11)	0.0385 ± 0.0044 (9)	0.1863 ± 0.0035 (9)	0.6206 ± 0.0614 (13)	2.0000 ± 2.0000 (6)
P-Norm P-Classification	0.8095 ± 0.0275 (15)	0.0382 ± 0.0042 (11)	0.1861 ± 0.0033 (10)	0.6188 ± 0.0604 (15)	1.6000 ± 1.3416 (8)
P-Norm Log-p-classification Hybrid	0.8104 ± 0.0294 (12)	0.0404 ± 0.0032 (2)	0.1880 ± 0.0033 (2)	0.6270 ± 0.0645 (3)	2.2000 ± 1.3038 (5)
P-Norm Log-Exp Hybrid	0.8104 ± 0.0277 (12)	0.0384 ± 0.0044 (10)	0.1863 ± 0.0034 (9)	0.6209 ± 0.0614 (12)	2.0000 ± 2.0000 (6)
P-Norm Square-Exp Hybrid	0.8124 ± 0.0278 (6)	0.0389 ± 0.0047 (7)	0.1868 ± 0.0035 (6)	0.6244 ± 0.0602 (5)	2.2000 ± 1.9235 (5)

Method	AUC	ARR	DCG	AP	PtOp
Proper Logistic	0.9976 ± 0.0012 (2)	0.1706 ± 0.0284 (1)	0.3411 ± 0.0275 (1)	0.9391 ± 0.0370 (3)	13.2000 ± 3.9623 (1)
Proper Exponential	0.9976 ± 0.0012 (2)	0.1705 ± 0.0286 (2)	0.3410 ± 0.0277 (2)	0.9376 ± 0.0339 (5)	12.8000 ± 3.9623 (3)
Proper P-Classification	0.9968 ± 0.0022 (8)	0.1703 ± 0.0290 (4)	0.3405 ± 0.0283 (6)	0.9316 ± 0.0394 (13)	12.8000 ± 3.9623 (3)
Proper Log- p -classification Hybrid	0.9973 ± 0.0014 (5)	0.1705 ± 0.0288 (2)	0.3409 ± 0.0280 (3)	0.9356 ± 0.0355 (8)	13.0000 ± 3.9370 (2)
Proper Log-Exp Hybrid	0.9973 ± 0.0014 (5)	0.1696 ± 0.0282 (8)	0.3399 ± 0.0273 (10)	0.9274 ± 0.0455 (17)	11.8000 ± 4.3243 (6)
Proper Square-Exp Hybrid	0.9972 ± 0.0018 (6)	0.1691 ± 0.0277 (9)	0.3395 ± 0.0269 (11)	0.9265 ± 0.0597 (18)	11.8000 ± 5.7184 (6)
Bipartite Logistic	0.9976 ± 0.0013 (2)	0.1704 ± 0.0286 (3)	0.3409 ± 0.0276 (3)	0.9371 ± 0.0375 (6)	12.8000 ± 3.9623 (3)
Bipartite Exponential	0.9976 ± 0.0012 (2)	0.1704 ± 0.0287 (3)	0.3408 ± 0.0278 (4)	0.9364 ± 0.0348 (7)	12.2000 ± 4.1473 (5)
Bipartite P-Classification	0.9977 ± 0.0012 (1)	0.1706 ± 0.0290 (1)	0.3411 ± 0.0281 (1)	0.9394 ± 0.0340 (1)	12.4000 ± 4.1593 (4)
Bipartite Log- p -classification Hybrid	0.9975 ± 0.0014 (3)	0.1702 ± 0.0278 (5)	0.3406 ± 0.0268 (5)	0.9354 ± 0.0457 (10)	13.0000 ± 4.3589 (2)
Bipartite Log-Exp Hybrid	0.9975 ± 0.0012 (3)	0.1703 ± 0.0287 (4)	0.3408 ± 0.0277 (4)	0.9355 ± 0.0360 (9)	12.2000 ± 4.1473 (5)
Bipartite Square-Exp Hybrid	0.9973 ± 0.0017 (5)	0.1698 ± 0.0284 (7)	0.3401 ± 0.0275 (8)	0.9293 ± 0.0523 (15)	13.0000 ± 4.4159 (2)
P-Norm Logistic	0.9976 ± 0.0013 (2)	0.1706 ± 0.0286 (1)	0.3411 ± 0.0277 (1)	0.9392 ± 0.0384 (2)	13.2000 ± 3.9623 (1)
P-Norm Exponential	0.9968 ± 0.0021 (8)	0.1702 ± 0.0288 (5)	0.3405 ± 0.0282 (6)	0.9307 ± 0.0370 (14)	12.8000 ± 3.9623 (3)
P-Norm P-Classification	0.9968 ± 0.0020 (8)	0.1704 ± 0.0291 (3)	0.3406 ± 0.0285 (5)	0.9318 ± 0.0358 (12)	13.0000 ± 3.9370 (2)
P-Norm Log- p -classification Hybrid	0.9969 ± 0.0019 (7)	0.1698 ± 0.0281 (4)	0.3400 ± 0.0272 (9)	0.9278 ± 0.0459 (16)	12.4000 ± 4.3359 (4)
P-Norm Log-Exp Hybrid	0.9976 ± 0.0012 (2)	0.1705 ± 0.0284 (2)	0.3410 ± 0.0275 (2)	0.9388 ± 0.0351 (4)	12.8000 ± 3.9623 (3)
P-Norm Square-Exp Hybrid	0.9974 ± 0.0015 (4)	0.1700 ± 0.0280 (6)	0.3403 ± 0.0270 (7)	0.9320 ± 0.0498 (11)	13.2000 ± 3.7683 (1)

Table 13: Results of various “ranking the best” methods on car data set.

Method	AUC	ARR	DCG	AP	PtOp
Proper Logistic	6.5000	6.2500	6.2500	8.2500	4.2500
Proper Exponential	5.5000	7.2500	6.7500	7.5000	5.2500
Proper P-Classification	9.0000	7.7500	8.5000	10.7500	5.5000
Proper Log- p -classification Hybrid	12.2500	4.2500	4.7500	9.0000	4.7500
Proper Log-Exp Hybrid	4.5000	9.0000	8.5000	10.2500	4.5000
Proper Square-Exp Hybrid	8.7500	7.7500	8.5000	11.7500	4.0000
Bipartite Logistic	5.7500	7.0000	6.7500	7.2500	3.0000
Bipartite Exponential	8.0000	3.7500	4.7500	8.5000	4.7500
Bipartite P-Classification	6.5000	5.0000	5.2500	5.7500	3.5000
Bipartite Log- p -classification Hybrid	6.7500	4.2500	4.0000	5.5000	2.7500
Bipartite Log-Exp Hybrid	3.2500	5.5000	5.0000	6.2500	4.2500
Bipartite Square-Exp Hybrid	6.2500	4.7500	5.0000	6.5000	2.0000
P-Norm Logistic	9.0000	7.2500	7.7500	9.0000	3.2500
P-Norm Exponential	8.5000	7.0000	7.2500	10.7500	5.7500
P-Norm P-Classification	9.5000	9.7500	10.2500	12.7500	4.2500
P-Norm Log- p -classification Hybrid	11.2500	9.2500	10.2500	12.7500	6.5000
P-Norm Log-Exp Hybrid	9.0000	6.2500	7.0000	9.5000	4.7500
P-Norm Square-Exp Hybrid	9.5000	7.0000	7.5000	8.7500	4.5000

Table 14: Average ranks of various “ranking the best” methods for each performance measure across all data sets.

more general in the sense of being for an unrestricted hypothesis class, and uses proper loss techniques. The result of [Erekin and Rudin \(2011\)](#) holds for the case of a linear (possibly misspecified) function class.

Variants of the p -norm push have been proposed, although the focus has been on algorithmic issues ([Rudin, 2009; Agarwal, 2011; Li et al., 2014](#)).

[Cossock and Zhang \(2008\)](#) proposed to use an importance weighting approach to the problem of focussing on the head of the ranked list, and showed that the DCG of a ranking can be bounded by importance weighted squared error:

10. Exact Compositional Reductions Between Classification and Ranking

We have introduced several seemingly distinct problems above, among them classification, class-probability estimation, pairwise ranking, and bipartite ranking. We now map out the relationships between these problems.²³ Our focus is on whether the Bayes-optimal solution for one of these problems can be transformed to give the optimal solution for another problem. Formally, for some pair of problems (A, B) —where we understand a “problem” to mean a specification of a distribution and loss, and hence the Bayes-optimal solutions—we would like to know if

$$(\forall s \in \mathcal{S}^{B \rightarrow \mathbb{R}})(\exists f : \mathbb{R} \rightarrow \mathbb{R}) f \circ s \in \mathcal{S}^{A*}$$

and vice-versa. When the above is true, we have an *exact compositional reduction* from A to B , in the sense of [problem A providing an optimal solution for problem B via a transformational \$f\$](#) .²⁴ According to our definition,

23. We will focus on the case of 0-1 loss for bipartite ranking, as this is the canonical performance measure in most studies. Recalling that the ℓ -bipartite ranking risk has equivalent Bayes-optimal solutions to class-probability estimation for certain strictly proper composite ℓ , the results derived here for class-probability estimation can be translated to the ℓ -bipartite ranking risk as well.

24. In practice, one must consider the impact misspecified hypothesis classes and finite samples have on transforming the solution of one problem to another. The recent work of [Narasimhan and Agarwal \(2013b\)](#) considers regret and generalisation bounds to this end.

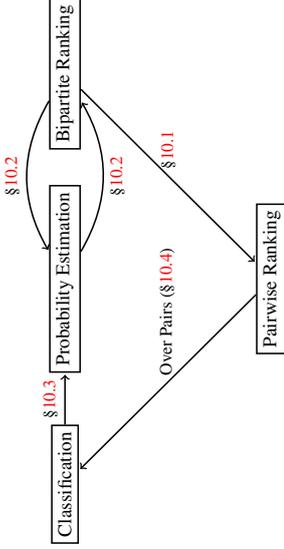


Figure 2: Relationships amongst various ranking and classification problems. An arrow $A \rightarrow B$ denotes that A is a special case of B , with the label on the arrow providing context on the relationship.

the transformation f may depend on the distribution specified by B . When this is so, the reduction is “weak” in the terminology of Narasimhan and Agarwal (2013b). When f is independent of B , the reduction is “strong” in the terminology of Narasimhan and Agarwal (2013b).

A sufficient condition for two problems to have the same Bayes-optimal solutions is the equivalence of the risks for the two problems. In some cases, this equivalence will be apparent from the specifications of the problem.

Figure 2 summarises the relationships amongst the various classification and ranking problems discussed in this paper. We now discuss these relationships in more detail.

10.1 Bipartite Ranking \subset Pairwise Ranking

The bipartite ranking risk for a pair-scorer is defined with respect to a classification distribution D , while the pairwise ranking risk is defined with respect to a ranking distribution R . Therefore, in general for the two quantities to be equal, we need to have some relationship between the distributions $D \in \Delta_{\mathcal{X} \times \mathcal{X}(\pm 1)}$ and $R \in \Delta_{\mathcal{X} \times \mathcal{X}(\pm 1)}$. The following shows that bipartite ranking is strictly a special case of pairwise ranking.

Proposition 62 *Pick any loss ℓ and scorer $s : \mathcal{X} \rightarrow \mathbb{R}$. Then, for every $D \in \Delta_{\mathcal{X} \times \mathcal{X}(\pm 1)}$ there is some $R \in \Delta_{\mathcal{X} \times \mathcal{X}(\pm 1)}$ such that*

$$\mathbb{L}_{\text{BR}}(s; D, \ell) = \mathbb{L}(\text{Diff}(s); R, \ell).$$

Further, there exists some $R \in \Delta_{\mathcal{X} \times \mathcal{X}(\pm 1)}$ such that

$$(\exists D \in \Delta_{\mathcal{X} \times \mathcal{X}(\pm 1)}) \mathbb{L}_{\text{BR}}(s; D, \ell) = \mathbb{L}(\text{Diff}(s); R, \ell).$$

Proof The first statement is immediate by setting $R = D_{\text{BR}}$, from Lemma 2. For the second statement, suppose there were such a D . Then by Lemma 2, it must be true that

$$\mathbb{L}(\text{Diff}(s); R, \ell) = \mathbb{L}(\text{Diff}(s); D_{\text{BR}}, \ell).$$

Thus, it must be true that $R = D_{\text{BR}}$. However, if we choose R with base rate different from $\frac{1}{2}$, this cannot be possible, since D_{BR} has base rate $\frac{1}{2}$ by construction. We have a contradiction, and the result is shown. ■

Proposition 62 formalises the intuition that pairwise ranking is more general than bipartite ranking: there exist instances of the former (even when operating over decomposable pair-scorers) that cannot be solved by the latter.

10.2 Bipartite Ranking = Class-Probability Estimation

We show that the Bayes-optimal solutions for bipartite ranking and class-probability estimation (with a strictly proper composite loss) may be transformed to one another. For bipartite ranking, the Bayes-optimal solution must be transformed in a distribution dependent manner (specifically, it must be calibrated with respect to the distribution).

Proposition 63 *Given any $D = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and $\ell \in \mathcal{L}_{\text{SPC}}$,*

$$\begin{aligned} & (\forall s \in S^*(D, \ell)) s \in S_{\text{BR}}^*(D, \ell_{01}) \\ & (\forall s \in S_{\text{BR}}^*(D, \ell_{01})) (\exists f^D) f^D \circ s \in S^*(D, \ell). \end{aligned}$$

Proof Pick $s \in S^*(D, \ell)$. By Equation 44, this s is unique, and satisfies $s = \Psi \circ \eta$. Thus, by Corollary 43, $s \in S_{\text{BR}}^*(D, \ell_{01})$.

Now pick $s \in S_{\text{BR}}^*(D, \ell_{01})$. By Proposition 42, $\eta = \phi \circ s$ for some non-decreasing ϕ . Thus, by Lemma 74, the calibrated version $\text{Cal}(s; D)$ of s must equal η . Letting $f^D = \Psi \circ \text{Cal}(\cdot; D)$ gives the result. ■

The strict properness of the proper composite loss is essential for these results. Given a non-strictly proper composite loss, such as 0-1 loss, it is not true that every Bayes-optimal solution is also optimal for bipartite ranking, as we will now see.

10.3 Classification \subset Class-Probability Estimation

Class-probability estimation can be shown to be a more general problem than binary classification, in the sense that an optimal solution for the former can be transformed to one for the latter, but the other direction is only true for certain classes of distributions.

Proposition 64 *Given any $D = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ and $\ell \in \mathcal{L}_{\text{SPC}}(\Psi)$,*

$$\begin{aligned} & (\forall s \in S^*(D, \ell)) (\exists f) f \circ s \in S^*(D, \ell_{01}) \\ & (\forall s \in S^*(D, \ell_{01})) (\exists f^D) f^D \circ s \in S^*(D, \ell) \iff (\forall x \in \mathcal{X}) \eta(x) \in \{a, b\}, \end{aligned}$$

where $a = b \neq \frac{1}{2}$ or $(2a - 1)(2b - 1) < 0$, i.e. η is constant or takes on exactly two values on different sides of $1/2$.

Proof For the first result, pick $s \in S^*(D, \ell)$. By Equation 44, this s is unique, and satisfies

$$s = \Psi \circ \eta.$$

Since $2\eta - 1 \in S^*(D, \ell_{01})$ (Equation 43), we can transform s to get an optimal solution for 0-1 loss.

For the second result, suppose that $\eta \in \{\frac{1}{2}, 1\}$. Then the scorer $\lfloor \eta \geq 1/2 \rfloor \equiv 1$ is in $S^*(D, \ell_{01})$ by Equation 43. This scorer takes on exactly one value. Therefore, it cannot be transformed to a function that takes on two or more values. (Note that there may exist some scorers that can be transformed to give η , but this is not the proposition in question.) A similar argument shows that we cannot handle the case where η takes on three or more values.

Now suppose that η satisfies the given conditions, and pick any $s \in S^*(D, \ell_{01})$. We know that $\text{sign}(s) = \text{sign}(2\eta - 1)$. Supposing without loss of generality that $a \leq b$, we have

$$\Psi \circ f^D \circ s = \Psi \circ \eta \in S^*(D, \ell)$$

where $f^D(x) = b \cdot \lfloor x > 0 \rfloor + a \cdot \lfloor x < 0 \rfloor$. ■

The restriction on η above is satisfied by *separable* or *noiseless* distributions, where for every $x \in \mathcal{X}$, $\eta(x) \in \{0, 1\}$. The above thus shows the intuitive fact that in the absence of noise, classification and class-probability estimation are equivalent in terms of their end goals.²⁵

10.4 Classification Over Pairs = Pairwise Ranking

We can confirm that pairwise ranking is equivalent to binary classification over the instance space $\mathcal{X} \times \mathcal{X}$ by simply comparing the risks for the two problems (Equations 19 and 14). This implies that bipartite ranking is also a special case of binary classification over $\mathcal{X} \times \mathcal{X}$, due to the relationship between bipartite and pairwise ranking established in §10.1.

10.5 Classification Over Singletons \subset Bipartite Ranking

While bipartite ranking reduces to classification over *pairs*, it does not reduce to classification over *singletons*, except in special cases. We now present conditions for the equivalence of the Bayes-optimal solutions of the two problems.

Proposition 65 *Given any $D = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \{1\}}$:*

$$\begin{aligned} (\forall s \in \mathcal{S}_{\text{BR}}^*(D, \ell_{01})) (\exists f) f \circ s \in \mathcal{S}^*(D, \ell_{01}) \\ (\forall s \in \mathcal{S}^*(D, \ell_{01})) (\exists f) f \circ s \in \mathcal{S}_{\text{BR}}^*(D, \ell_{01}) \iff (\forall x \in \mathcal{X}) \eta(x) \in \{a, b\}, \end{aligned}$$

where $a = b \neq \frac{1}{2}$ or $(2a - 1)(2b - 1) < 0$, i.e. η is constant or takes on exactly two values on different sides of 1/2.

Proof The result follows by the established relationships between classification and class-probability estimation (§10.3), and class-probability estimation and bipartite ranking (§10.2). ■

As with class-probability estimation, the above shows that for separable distributions, bipartite ranking is equivalent to binary classification in terms of the end goal.

10.6 Relation to Existing Work

For the case of 0-1 loss, the fact that the bipartite ranking risk exactly equals a specific pairwise classification risk (and hence a specific pairwise ranking risk) is well known (Clemençon et al., 2008; Kotowski et al., 2011; Demaiusi and Lee, 2012; Agarwal, 2014). The derived ranking distribution D_{BR} , which explicitly specifies the pairwise ranking distribution for which this holds, has been invoked by Balcan et al. (2008); Kotowski et al. (2011); Agarwal (2014), among others. Our generalisation to an arbitrary loss ℓ , while simple, appears novel.

Our study of the relationship between the bipartite and pairwise ranking problems differs from that of Balcan et al. (2008); Ailon and Mohri (2007) in at least two aspects. First, those works look at a *subset* version of bipartite ranking, where the goal is to rank a given subset of instances. Second, those works consider the goal of bipartite ranking to produce a univariate scorer rather than a pair-scorer. Therefore, they consider the question of how one can derive a univariate scoring function suitable for ranking from a classifier over pairs. The main result of Balcan et al. (2008) is that, given a classifier of pairs that achieves small classification risk, one can produce a univariate scorer with bipartite ranking risk that is worse by at most a factor of two.

11. Four Bipartite Ranking Risks With Equivalent Minimisers

Consider the following approaches to producing a pair-scorer, given a strictly proper composite ℓ :

²⁵ However, as noted earlier, for separable data the infimum in the definition of the Bayes risk may be unattainable for a strictly proper composite loss.

$$\begin{array}{ll} (1) \text{ Diff} \left(\underset{s: \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{R}}{\text{argmin}} \mathbb{E}_{X \sim P_X, X' \sim Q} [e^{-\langle s(X), X' \rangle}] \right) & (2) \text{ Diff} \left(\underset{s: \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{R}}{\text{argmin}} \mathbb{E}_{X \sim P_X, X' \sim Q} [e^{-\langle s(X), -s(X') \rangle}] \right) \\ (3) \underset{\Psi_{\text{BR}}: \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{R}}{\text{argmin}} \mathbb{E}_{X \sim P_X, X' \sim Q} [e^{-\langle \Psi_{\text{BR}}(X, X') \rangle}] & (4) \text{ Diff} \left(\underset{s: \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{R}}{\text{argmin}} \mathbb{E} \left[\left(\mathbb{E}_{X \sim P} [e^{-\langle s(X), -s(X') \rangle}] \right)^p \right] \right) \end{array}$$

Table 15: Four approaches for obtaining a pair-scorer in a bipartite ranking problem, using exponential loss. Our results show that the all approaches have the same theoretical minimiser.

- (1) Minimise the ℓ -classification risk $L(\mathcal{G}; D, \ell)$, and Diff (·) the result.
- (2) Minimise the ℓ -bipartite ranking risk $L_{\text{BR}}(\mathcal{G}; D, \ell)$ over all scorers, and Diff (·) the result.
- (3) Minimise the ℓ -pairwise ranking risk $L(\mathcal{S}_{\text{Pair}}; D_{\text{BR}}, \ell)$ over all pair-scorers.
- (4) Minimise the p -norm push risk $\text{Push}(S_{\text{Pair}}; D, \ell_{\text{exp}}, g^p)$ over decomposable pair-scorers.

Superficially, these appear very different: method (4) is the only one that departs from the standard conditional risk framework, method (3) is the only one to use a pair-scorer during minimisation, and method (1) is the only one to operate on single instances rather than pairs. It is thus surprising that our results provide conditions under which all methods have the *same* output; it is further surprising that the condition involves the choice of link function in the loss ℓ , which is typically chosen for computational rather than statistical reasons (Reid and Williamson, 2010).

Proposition 66 *Given any $D \in \Delta_{\mathcal{X} \times \{1\}}$ and $\ell \in \mathcal{L}_{\text{Spec}}(\Psi)$ with $\Psi \in \Sigma_{\text{sig}}$, methods (1), (2) and (3) produce the same pair-scorer, if \mathcal{X} is finite and $p = a - 1$ for $a > 1$, method (4) also produces the same pair-scorer.*

Proof By Equation 44 and Corollary 45, methods (1) and (2) produce the same scorer Ψ_{opt} , up to a translation which is nullified by the Diff operator. By Equation 50, this pair-scorer is equivalent to that produced by method (3). Further, if $p = a - 1$ for $a > 1$, then by Proposition 57, method (4) returns Ψ_{opt} up to a translation which is nullified by the Diff operator. ■

In hindsight, these equivalences are not surprising by virtue of the Bayes-optimal scorer for each type of risk depending on the observation-conditional distribution η . They are not however *a priori* obvious, given how ostensibly different the risks appear. To illustrate these superficial differences, Table 15 provides a concrete example of the four methods when $\ell = \ell^{\text{exp}}$ is the exponential loss, whose link $\Psi = \frac{1}{2} \sigma^{-1}$ satisfies the required condition of Proposition 66.

11.1 Implications of Equivalences

The above shows the “equivalence” between four seemingly disparate risks, where our definition of “equivalent” is that two methods have the same optimal scorer. This does not imply that the methods are interchangeable in practice. A statistical caveat to these equivalences is that they ignore the issues of finite samples and a restricted function class. When one or both of these situations hold, it may be that one of these methods is more preferable. A computational caveat is that methods (2)—(4) rely on minimisation over pairs of examples. On a finite training set, this requires roughly quadratic complexity, compared to the linear complexity of method (1). These practical issues deserve investigation, but are beyond the scope of this paper.

This caveat in mind, we believe the results at least illuminate similarities between seemingly disparate approaches. For the problem of minimising the ℓ -bipartite risk for an appropriate surrogate ℓ , the above provides evidence that minimising the ℓ -classification risk is a suitable proxy. That is, performing class-probability estimation is a suitable proxy for ranking; this can be formalised with surrogate regret bounds (Agarwal, 2014; Narasimhan and Agarwal, 2013b). Similarly, for the problem of minimising the p -norm push

Type	Property ($\forall x, x', x'' \in \mathcal{X}$)
Total	$\neg(R(x, x') = -1 \text{ and } R(x', x) = -1)$
Anti-symmetric	$\neg(R(x, x') = +1 \text{ and } R(x', x) = +1)$
Transitive	$R(x, x') = +1 \text{ and } R(x', x'') = +1 \implies R(x, x'') = +1$
Reflexive	$R(x, x) = +1$
Continuous	for every pair of convergent sequences $(x_n), (y_n)$, $(\forall n \in \mathbb{N}) R(x_n, y_n) = +1 \implies R\left(\lim_{n \rightarrow \infty} x_n, \lim_{n \rightarrow \infty} y_n\right) = +1$
Preorder	Reflexive, Transitive
Total preorder	Total, Transitive
Partial order	Reflexive, Transitive, Anti-symmetric
Total order	Total, Transitive, Anti-symmetric

Table 16: Some common types of binary relation R , and their defining properties.

objective, we have evidence that minimising the ℓ -classification or bipartite risk is a suitable proxy. As seen in §9.7, certain proper composite losses do indeed give comparable performance to the p -norm push.

11.2 Relation to Existing Work

Subsets of the above equivalences have been observed earlier for special cases. For the specific case of exponential loss and a linear hypothesis class, the equivalence between methods (1) and (2) was made by [Ertekin and Rudin \(2011, Theorem 3\)](#); [Gao and Zhou \(2012, Lemma 4\)](#); [Gao and Zhou \(2015, Theorem 7\)](#), while the equivalence between method (1) and (4) was shown by [Ertekin and Rudin \(2011, Theorem 1\)](#); here, method (1) represents AdaBoost, and method (2) RankBoost. For the special case of convex margin losses, the equivalence between methods (2) and (3) was shown by [Uematsu and Lee \(2012\)](#). [Ertekin and Rudin \(2011, Section 4.4\)](#) conjectured a lack of an equivalence between LogitBoost and logistic regression, based on empirical findings; this apparent contradiction with our results is because the latter focusses on the case of a linear hypothesis class, which is possibly misspecified, while our results are for an unrestricted hypothesis class, or equally for a correctly specified one.

12. A Utility Representation Perspective of Bipartite Ranking

Our final topic of study is what the theory of *utility representations* tells us about the bipartite ranking problem. In particular, we look at how this theory provides insight into the particular form of the observation-conditional distribution $\mathcal{P}_{\text{pair}}$ of D_{PAR} (Equation 47), which we saw had non-obvious implications for the derivation of Bayes-optimal scorers (§7.4.2) and surrogate regret bounds (§8).

12.1 Binary Relations

Given a set \mathcal{X} , a *binary relation* \mathcal{R} on \mathcal{X} is some subset of $\mathcal{X} \times \mathcal{X}$. There are two standard ways of referring to a relation. The first is the operator $\succeq_{\mathcal{R}}$, with the semantics $x \succeq_{\mathcal{R}} x' \iff (x, x') \in \mathcal{R}$. The second is the function $r: \mathcal{X} \times \mathcal{X} \rightarrow \{\pm 1\}$, with the semantics $r(x, x') = +1 \iff (x, x') \in \mathcal{R}$. We will use these two representations interchangeably. Table 16 summarises some standard properties of binary relations, and examples of specific types of binary relations.

A *probabilistic binary relation* (sometimes called a *reciprocal* or *ipsodual binary relation*) ([Baets et al., 2005, pg. 419](#)) on a set \mathcal{X} is some function $p: \mathcal{X} \times \mathcal{X} \rightarrow [0, 1]$ satisfying $p(x, x') + p(x', x) = 1$ for every $x, x' \in \mathcal{X}$. The pair (\mathcal{X}, p) is sometimes referred to as a *forced choice pair comparison system* ([Roberts, 1984, pg. 273](#)). Every probabilistic binary relation has an induced binary relation \succeq_p , with

$$x \succeq_p x' \iff p(x, x') \geq \frac{1}{2}.$$

Suppose $g: \left[\frac{1}{2}, 1\right] \times \left[\frac{1}{2}, 1\right] \rightarrow [0, 1]$ is some function that is commutative (i.e. $g(x, y) = g(y, x) \forall x, y$) and monotone increasing in both arguments. A probabilistic binary relation p is said to be *g-stochastically transitive* ([Baets et al., 2005, pg. 419](#)) if

$$(\forall x, x', x'' \in \mathcal{X}) x \succeq_p x' \text{ and } x' \succeq_p x'' \implies p(x, x'') \geq g(p(x, x'), p(x', x'')).$$

Special cases of the function g correspond to popular notions of stochastic transitivity: the case $g(x, y) = 1/2$ is known as *weak stochastic transitivity* ([Roberts, 1984, pg. 283](#)), $g(x, y) = x \wedge y$ as *moderate stochastic transitivity* ([Roberts, 1984, pg. 284](#)), and $g(x, y) = x \vee y$ as *strong stochastic transitivity* ([Roberts, 1984, pg. 284](#)). It is easy to check that weak stochastic transitivity of p corresponds to transitivity of the associated binary relation \succeq_p , provided ties are broken in favour of the relation existing.

12.2 Utility Representations for Binary Relations

Let $\succeq_{\mathcal{R}}$ be a binary relation on \mathcal{X} . We say that $\succeq_{\mathcal{R}}$ has a *utility representation* if there is some $s: \mathcal{X} \rightarrow \mathbb{R}$ such that

$$(\forall x, x' \in \mathcal{X}) x \succeq_{\mathcal{R}} x' \iff s(x) \geq s(x').$$

We say that a probabilistic binary relation p has a *generalised utility representation* if there is some function $H: \mathbb{R} \times \mathbb{R} \rightarrow [0, 1]$ and some function $s: \mathcal{X} \rightarrow \mathbb{R}$ such that

$$(\forall x, x' \in \mathcal{X}) p(x, x') = H(s(x), s(x')), \tag{64}$$

where H is increasing in its first argument, decreasing in its second argument. In [Świątalski \(2003\)](#), it is additionally assumed that H is concave-convex. In psychometrics, such a model is sometimes also referred to as a simple scalability assumption ([Krantz, 1967](#)).

Table 17 summarises various special cases of the generalised utility representation that have been studied; see ([Roberts, 1984, pg. 273–280](#)) for details. The utility representations are ordered as follows:

$$\text{Strict} \subsetneq \text{Fechnerian} \subseteq \text{Strong} \subsetneq \text{Weak}.$$

Note that the inclusions above are all strict, except for that of Fechnerian and Strong representations. In fact, p has a strong utility representation if and only if it has a *restricted* Fechnerian utility representation, where the restriction is that the strictly monotone increasing ϕ for the Fechnerian representation is only defined on the Minkowski self-difference $f(\mathcal{C}) - f(\mathcal{C})$ ([Roberts, 1984, pg. 279](#)).

12.3 Existence of Utility Representations

A key question is whether one can characterise when a given (probabilistic) binary relation possesses a specific type of utility representation. This lets us relate the ordering properties of the relation—such as whether it is symmetric, transitive, *et cetera*—with its mathematical representation as a function.

A classical result of [Debreu \(1954\)](#) (which generalises a result of [Eilenberg \(1941\)](#)) characterises, when a (non-probabilistic) binary relation may be expressed via a real-valued utility function. Recall that a binary relation $\succeq_{\mathcal{R}}$ is a total preorder if it is total (and hence reflexive) and transitive. The theorem is as follows.

26. More generally, one may define a family of relations, where each member specifies a different scheme for how ties—corresponding to $p(x, x') = \frac{1}{2}$ —are broken.

Utility type	Definition	Characterisation
Weak	$H(a, b) \geq \frac{1}{2} \iff a \geq b$	Weak stochastic transitivity, contour sets closed
Strong	$H(a, b) \geq H(c, d) \iff a - b \geq c - d$	Quadruple condition (under stochastic continuity)
Fechnerian	$H(a, b) = \phi(a - b)$, ϕ monotone	Quadruple or bicancellative condition
Strict	$H(a, b) = \frac{a}{a+b}$	Product rule

Table 17: Summary of various types of utility representations for binary probabilistic relations, from most to least general. By “definition” we mean the conditions required of H in the general utility representation of Equation 64. By “characterisation” we mean necessary and sufficient conditions on a probabilistic binary relation for the representation to hold. See text for details.

Proposition 67 (Eilenberg, 1941; Debreu, 1954; Debreu, 1964; Bridges and Mehta, 1995, pg. 46) Let \mathcal{X} be a topological space that is either (a) connected and separable, or (b) second countable. Let $\succeq_{\mathcal{R}}$ be a binary relation on \mathcal{X} . Then $\succeq_{\mathcal{R}}$ defines a continuous total preorder if and only if there is some $s : \mathcal{X} \rightarrow \mathbb{R}$ such that

$$x \succeq_{\mathcal{R}} x' \iff s(x) \geq s(x').$$

The result characterises precisely the class of binary relations that may be represented via a utility function $s : \mathcal{X} \rightarrow \mathbb{R}$. While the “if” direction is straightforward, the “only if” direction is not: it implies that any continuous total preorder can be perfectly represented via the standard ordering relation \geq on the reals, for an appropriate choice of utility function $s : \mathcal{X} \rightarrow \mathbb{R}$.

For probabilistic binary relations, we will focus on the case of a strict utility representation,²⁷ for which

$$H(a, b) = \frac{a}{a+b} = \frac{1}{1 + \frac{b}{a}} = \sigma(\sigma^{-1}(a') - \sigma^{-1}(b')),$$

where $d' = \frac{a}{a+1}$, $b' = \frac{b}{b+1}$. Thus, if a probabilistic binary relation p possesses this representation,

$$(\exists s : \mathcal{X} \rightarrow \mathbb{R})(\forall x, x' \in \mathcal{X}) p(x, x') = \sigma(\sigma^{-1}(s(x)) - \sigma^{-1}(s(x'))). \quad (65)$$

We have the following characterisation of the existence of a strict utility representation.

Proposition 68 (Luce and Suppes, 1965; Theorem 48, pg. 350) Suppose η_{pair} is a binary probabilistic relation. Then, η_{pair} has a strict utility representation (Equation 65) if and only if it satisfies the product rule,

$$(\forall x, x', x'' \in \mathcal{X}) \eta_{\text{pair}}(x, x') \cdot \eta_{\text{pair}}(x', x'') \cdot \eta_{\text{pair}}(x'', x) = \eta_{\text{pair}}(x, x'') \cdot \eta_{\text{pair}}(x'', x') \cdot \eta_{\text{pair}}(x', x), \quad (66)$$

which, when $\eta_{\text{pair}} \neq \{0, 1\}$, is equivalently

$$(\forall x, x', x'' \in \mathcal{X}) \eta_{\text{pair}}(x, x') = \sigma(\sigma^{-1}(\eta_{\text{pair}}(x, x'')) + \sigma^{-1}(\eta_{\text{pair}}(x'', x')) - \sigma^{-1}(\eta_{\text{pair}}(x', x))).$$

The product rule encodes that the probability of an intransitive cycle of relations $\{x \succeq_p x', x' \succeq_p x'', x'' \succeq_p x\}$ equals the probability of the cycle $\{x \succeq_p x', x' \succeq_p x'', x'' \succeq_p x\}$. The product rule necessarily implies the quadruple condition introduced earlier (and hence strong stochastic transitivity), since a strict utility implies a Fechnerian one.

²⁷ In psychometrics, the strict utility representation is referred to as the Bradley-Terry-Luce model (Luce, 1959; Bradley and Terry, 1952).

12.4 Implications for Bipartite Ranking

Bipartite ranking is fundamentally concerned with learning a scorer $s : \mathcal{X} \rightarrow \mathbb{R}$, and using this to rank instances. We can equivalently think of the problem as a special case of pairwise ranking, where we restrict attention to decomposable pair-scorers (Lemma 2). The pairwise ranking problem can in turn be interpreted as one of learning a binary relation: if $r : \mathcal{X} \times \mathcal{X} \rightarrow \{\pm 1\}$ is a binary relation on $\mathcal{X} \times \mathcal{X}$, then for any $R \in \Delta_{\mathcal{X} \times \mathcal{X} \times \{\pm 1\}}$ such that the Bayes-optimal 0-1 pair-scorers match r in sign, i.e.

$$(\forall x, x' \in \mathcal{X}) r(x, x') = \text{sign}(\mathcal{Q}_{\text{pair}}(x, x') - 1),$$

learning a pair-scorer from R is equivalent to learning the binary relation r . Thus, one can then ask what implications the decomposability restriction has on learning a binary relation. Observe that when thresholded at 0, the pair-scorer $\text{Diff}(s)$ yields a binary relation r over $\mathcal{X} \times \mathcal{X}$, with

$$r(x, x') = +1 \iff (\text{Diff}(s))(x, x') \geq 0 \iff s(x) \geq s(x').$$

By Proposition 67, the resulting relation r is a continuous, total preorder, as can be easily checked by the properties of $\text{Diff}(s)$. Less obviously, Proposition 67 implies that for any continuous, total preorder $\succeq_{\mathcal{R}}$ over $\mathcal{X} \times \mathcal{X}$ (for \mathcal{X} with suitable topological properties), the problem of learning $\succeq_{\mathcal{R}}$ can be expressed as a bipartite ranking problem.

Proposition 69 Let \mathcal{X} be a topological space that is either (a) connected and separable, or (b) second countable. Let $\succeq_{\mathcal{R}}$ be a continuous, total preorder on \mathcal{X} . Then, there is a $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ for which the Bayes-optimal bipartite scorer for 0-1 loss induces the same ranking as $\succeq_{\mathcal{R}}$.

Proof For any continuous, total preorder $\succeq_{\mathcal{R}}$, there is a corresponding utility representation $s : \mathcal{X} \rightarrow \mathbb{R}$ (by Proposition 67). Pick any strictly monotone increasing $\phi : \mathbb{R} \rightarrow [0, 1]$, and let $\eta = \phi \circ s$. Further, pick any marginal distribution M over \mathcal{X} . Then, $D = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ with corresponding D_{GR} has

$$(\forall x, x' \in \mathcal{X}) \text{sign}(\mathcal{Q}_{\text{pair}}(x, x') - 1) = \text{sign}(\eta(x) - \eta(x')) = \text{sign}(s(x) - s(x')) = r(x, x'),$$

so that the Bayes-optimal bipartite scorer for 0-1 loss induces an ordering over $\mathcal{X} \times \mathcal{X}$ identical to $\succeq_{\mathcal{R}}$. ■

One can also consider the implications of utility representation theory for learning a probabilistic binary relation. As above, it is clear that the problem of learning a probabilistic binary relation p that satisfies the product rule can be expressed as finding $\text{SGR}(D, \ell)$ for some suitable D and proper composite ℓ with decomposable Bayes-optimal scorer. Equation 47 implies that for any D with derived distribution D_{GR} , the corresponding η_{pair} possesses a strict utility representation. Thus, setting η_{pair} to coincide with the utility representation of p gives a means of learning the relation p .

Conversely, we can gain some insight as to the form of observation-conditional distribution η_{pair} of D_{GR} is of the special form given by Equation 47, which in turn explains why the sigmoidal family of links arises in §7.4.2.

Proposition 70 Given any $D \in \Delta_{\mathcal{X} \times \{\pm 1\}}$, the resulting $D_{\text{GR}} = \langle M_{\text{pair}}, \eta_{\text{pair}} \rangle \in \Delta_{\mathcal{X} \times \mathcal{X} \times \{\pm 1\}}$ satisfies the product rule, and has

$$(\forall x, x' \in \mathcal{X}) \eta_{\text{pair}}(x, x') = \sigma(\sigma^{-1}(s(x)) - \sigma^{-1}(s(x')))$$

for some $s : \mathcal{X} \rightarrow \mathbb{R}$.

Proof For $D = \langle P, Q, \pi \rangle$, from the construction of $D_{\text{GR}} = \langle M_{\text{pair}}, \eta_{\text{pair}} \rangle = \langle P_{\text{pair}}, Q_{\text{pair}}, \pi_{\text{pair}} \rangle$, it is immediate that the probabilistic relation η_{pair} it represents satisfies the product rule—this is because Bayes’ rule and the nature of P_{pair} implies that

$$(\forall x, x' \in \mathcal{X}) \eta_{\text{pair}}(x, x') = \frac{P_{\text{pair}}(x, x') \cdot \pi_{\text{pair}}}{M_{\text{pair}}(x, x')}$$

$$= \frac{p(x) \cdot q(x')}{2\eta_{\text{pair}}(x, x')},$$

so that the condition for the product rule (Equation 66) may be written

$$(\forall x, x' \in \mathcal{X}) \frac{p(x) \cdot q(x')}{\eta_{\text{pair}}(x, x')} \cdot \frac{p(x') \cdot q(x'')}{\eta_{\text{pair}}(x', x'')} \cdot \frac{p(x'') \cdot q(x)}{\eta_{\text{pair}}(x'', x)} = \frac{p(x) \cdot q(x'')}{\eta_{\text{pair}}(x, x'')} \cdot \frac{p(x'') \cdot q(x)}{\eta_{\text{pair}}(x'', x)} \cdot \frac{p(x') \cdot q(x)}{\eta_{\text{pair}}(x', x)}$$

The numerators are clearly identical, and the denominators can be shown to be identical by explicit multiplication of the form of η_{pair} in Appendix B. Consequently, Proposition 68 thus implies that η_{pair} must possess a strict utility representation, meaning it is of the form

$$(\forall x, x' \in \mathcal{X}) \eta_{\text{pair}}(x, x') = \sigma(\sigma^{-1}(s(x)) - \sigma^{-1}(s(x')))$$

for some $s : \mathcal{X} \rightarrow \mathbb{R}$. ■

Thus, we have an explanation for the specific form of Equation 47—it is due to the probabilistic relation implicitly underlying the bipartite ranking problem satisfying the product rule, in conjunction with the utility representation theorem (Proposition 68) for all such relations.

13. Conclusion and Future Work

We have provided a systematic study of the bipartite ranking problem through its statistical risk. In particular:

- We described a fundamental connection between bipartite ranking and classification over pairs (§4).
- We studied several properties of the ROC curve, including a to our knowledge novel result (Proposition 13) on how dominance in ROC space implies dominance with respect to *any* proper composite loss.
- We derived a number of integral representations of the AUC (§5.6), relating them to the integral representation for proper losses, (§5.6.2).
- We related the Bayes-optimal bipartite risk to an f -divergence between product measures for the class-conditional densities (§6.2), generalising a result for the case of 0-1 loss due to Torgersen (1991).
- We determined the set of Bayes-optimal scorers for bipartite ranking (§7.3, §7.4, §7.5), and thus surrogate regret bounds for the minimisation of pairwise surrogates (§8).
- We studied Bayes-optimal scorers for the p -norm push risk (§9.5), and explained the risk in terms of the weight function for proper losses (§9.6.3). We used this to derive several new loss functions (§9.6), which demonstrated favourable empirical performance compared to the p -norm push risk on a number of real data sets (§9.7).
- We mapped out the relationships between bipartite ranking and other learning problems, such as pairwise ranking, class-probability estimation, and classification (§10.1, §10.2), and the equivalence between several seemingly disparate risks for popular approaches in bipartite ranking (§11).
- We showed how theorems of utility representation describe the class of ranking problems over pairs that can be modelled by bipartite ranking (§12.4).

Our results built upon the rich framework of proper composite losses, which are central to the study of class-probability estimation. We hope our results illustrate the value of the proper loss machinery in studying bipartite ranking problems.

We outline several possible areas for future work.

- *Beyond the Bayes-risk.* We acknowledge that the study of the risk is only an initial step in the broader understanding of the bipartite ranking problem. For example, in the study of the Bayes-optimal scorers, we have assumed no restrictions of the set of allowed scorers and pair-scorers. In practice, we have access to only a finite number of samples, and typically use a restricted function class. Understanding the impact this has on the risk equivalences we have established is of interest. For example, can we characterise when one risk is more preferable from a statistical or computational point of view?
 - *From the AUC to AUPRC.* One may hope to extend our analysis to other popular performance metrics for bipartite ranking, such as the area under the precision-recall curve (AUPRC). While we briefly touched upon the AUPRC in §9.4.2, we deferred detailed study due to difficulties in expressing it as a risk. Understanding the properties this broader class of performance measures is of interest.
 - *Extension to instance ranking.* A natural extension of our results would be the study of the more general instance ranking problem, where the label space \mathcal{Y} is not binary. With suitable assumptions on \mathcal{Y} , it is possible to leverage some analysis from bipartite ranking for such problems (Clémentçon et al., 2013). It is possible that tools from multi-class probability estimation (Vernet et al., 2011) may also be a useful tool in the study of this problem.
 - *Converting pair-scorers to a univariate scorer.* Given a pair-scorer, it may be desirable to construct a procedure that converts it to a univariate scorer. Balean et al. (2008) devises such a procedure and provides guarantees on its performance for the standard AUC. It is of interest whether this can be generalised.
 - *Applications.* We believe there is scope for our analysis to be extended to problems where both class-probability estimation and bipartite ranking are heavily employed. For example, a challenging learning scenario is where one has access to only positive and unlabelled examples. Recent work has shown the value of class-probability estimation (Elkan and Noto, 2008) and bipartite ranking methods (Sella-manickam et al., 2011) for this task. We hope that our analysis can offer directions for theoretical and algorithmic development for this and related problems.
- More broadly, by focussing on the statistical risk and abstracting away finite sample and optimisation issues, we have aimed to perform a *problem-oriented* rather than *method-oriented* analysis, as per Platt (1962). We believe this gives deeper insight into the connections between problems, and disentangles computational and statistical concerns. For example, akin to the distributional analysis of the probing reduction (Langford and Zadrozny, 2005) in Reid and Williamson (2009), we have seen in §10 the broad theoretical connections between bipartite ranking, class-probability estimation and classification, such as the use of a calibration transform to convert a scorer that ranks instances optimally to a class-probability estimator. We hope our results demonstrate the value and encourage further pursuit of this style of analysis for other learning problems.

Acknowledgments

This work was conducted as part of NICTA, and was supported by the Australian Research Council (RCW) and NICTA (AKM and RCW). NICTA was funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program. Thanks to Cynthia Rudin and the COLT referees for their helpful comments on a preliminary version of this work, to the JMLR referees for their valuable suggestions, and to Brendan van Rooyen for suggesting a simple proof of Lemma 72.

Appendix A. Assorted Lemmas

We collect some assorted lemmas that are employed in proofs, but are not directly related to bipartite ranking.

Lemma 71 Let $s : \mathcal{X} \rightarrow \mathbb{R}$ be any scorer. If

$$(\forall t : \mathcal{X} \rightarrow \mathbb{R}) \int_{\mathcal{X}} s(x) \cdot t(x) dx = 0,$$

then s is zero almost everywhere.

Proof This is in fact a special case of the fundamental lemma of the calculus of variations, which in turn is a special case of the du Bois-Reymond lemma (Trounman, 1996, pg. 99; Giacquinta and Hildebrandt, 2004, pg. 16), both of which only require the statement hold for all infinitely differentiable t . We show the contrapositive. Suppose $\varepsilon \neq 0$ on a set of nonzero measure. Then $t = s^2 : \mathcal{X} \rightarrow \mathbb{R}_+$ is > 0 on this same set of nonzero measure. Thus,

$$\int_{\mathcal{X}} s(x) \cdot s(x) dx = \int_{\mathcal{X}} s(x)^2 dx > 0,$$

where the inequality holds by Folland (1999, Proposition 2.16). Thus, the statement holds. ■

Lemma 72 Let $f, g : \mathcal{X} \rightarrow \mathbb{R}$. Then,

$$(\forall x, x' \in \mathcal{X}) f(x) < f(x') \implies g(x) < g(x')$$

if and only if $f = \phi \circ g$ for some non-decreasing $\phi : \mathbb{R} \rightarrow \mathbb{R}$.

Proof (\Leftarrow). This is easily verified by the definition of ϕ being non-decreasing (\implies). We will construct such a non-decreasing ϕ . For any $y \in \text{Im}(g)$, let

$$\mathcal{J}(y) = \{x \in \mathcal{X} : g(x) = y\}$$

be the preimage of y under g . For any $y \in \mathbb{R}$, let

$$\phi(y) \doteq \min\{f(x) : x \in \mathcal{J}(y)\}.$$

We will check that $f = \phi \circ g$, and that ϕ is non-decreasing.

First, note that for any $x, x' \in \mathcal{J}(y)$, by definition $g(x) = g(x')$. By the contrapositive of the assumption,

$$g(x) \geq g(x') \implies f(x) \geq f(x')$$

and by swapping x, x' ,

$$g(x) \leq g(x') \implies f(x) \leq f(x')$$

so that

$$g(x) = g(x') \implies f(x) = f(x').$$

Thus for any $x, x' \in \mathcal{J}(y)$, $f(x) = f(x')$. Thus, for any $x \in \mathcal{J}(y)$,

$$\phi(y) = f(x).$$

Now, for any $x_0 \in \mathcal{X}$,

$$\begin{aligned} \phi(g(x_0)) &= \min\{f(x) : x \in \mathcal{J}(g(x_0))\} \\ &= f(x_0). \end{aligned}$$

Thus, $f = \phi \circ g$. To see that ϕ is non-decreasing, pick $y < y'$ and $x \in \mathcal{J}(y)$, $x' \in \mathcal{J}(y')$. Then $y = g(x) < g(x') = y'$. Since $g(x) < g(x')$ implies $f(x) = \phi(y) < \phi(y') = f(x')$, we see that $y < y' \implies \phi(y) < \phi(y')$. ■

Lemma 73 Let $f, g : \mathcal{X} \rightarrow \mathbb{R}$. Then,

$$(\forall x, x' \in \mathcal{X}) \text{sign}(f(x) - f(x')) = \text{sign}(g(x) - g(x'))$$

if and only if $f = \phi \circ g$ for some strictly monotone increasing $\phi : \mathbb{R} \rightarrow \mathbb{R}$.

Proof We can equivalently write the condition as

$$(\forall x, x' \in \mathcal{X}) f(x) < f(x') \iff g(x) < g(x').$$

Thus, by Lemma 72, $f = \phi \circ g$ for some monotone increasing ϕ , and $g = \phi_2 \circ f$ for some monotone increasing ϕ_2 . Thus $f = \phi_1 \circ \phi_2 \circ f$, and so $\phi_1 = \phi_2^{-1}$. This implies that ϕ_1 and ϕ_2 are invertible, or equivalently, that they both correspond to strictly monotone increasing transforms. ■

Lemma 74 Given any $D = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$, and any $s : \mathcal{X} \rightarrow \mathbb{R}$ such that $\eta = \phi \circ s$ for some non-decreasing ϕ ,

$$(\forall x \in \mathcal{X}) \text{Cal}(x; D, s) = \eta(x).$$

In particular, the above equation holds for any s such that $s = \phi \circ \eta$ for some strictly monotone ϕ .

Proof Assuming that the distribution of scores is discrete, for every $x \in \mathcal{X}$,

$$\begin{aligned} \text{Cal}(x; D, s) &= \mathbb{P}[Y = 1 | s(X) = s(x)] \\ &= \frac{\mathbb{E}_{(X,Y) \sim D} \mathbb{1}[Y = 1, s(X) = s(x)]}{\mathbb{E}_{(X,Y) \sim D} \mathbb{1}[s(X) = s(x)]} \\ &= \frac{\mathbb{E}_{X \sim M} \mathbb{E}_{Y \sim D} [\eta(X) \cdot \mathbb{1}[s(X) = s(x)]]}{\mathbb{E}_{X \sim M} \mathbb{E}_{Y \sim D} [\mathbb{1}[s(X) = s(x)]]} \\ &= \frac{\mathbb{E}_{X \sim M} \mathbb{E}_{Y \sim D} [\phi(s(X)) \cdot \mathbb{1}[s(X) = s(x)]]}{\mathbb{E}_{X \sim M} \mathbb{E}_{Y \sim D} [\mathbb{1}[s(X) = s(x)]]} \quad \text{by assumption on } \eta \\ &= \frac{\mathbb{E}_{X \sim M} \mathbb{E}_{Y \sim D} [\phi(s(x)) \cdot \mathbb{1}[s(X) = s(x)]]}{\mathbb{E}_{X \sim M} \mathbb{E}_{Y \sim D} [\mathbb{1}[s(X) = s(x)]]} \\ &= \frac{\mathbb{E}_{X \sim M} \mathbb{E}_{Y \sim D} [\mathbb{1}[s(X) = s(x)]]}{\mathbb{E}_{X \sim M} \mathbb{E}_{Y \sim D} [\mathbb{1}[s(X) = s(x)]]} \\ &= \phi(s(x)) \\ &= \eta(x). \end{aligned}$$

When the distribution of scores is continuous, we repeat the above, but using Dirac delta instead of indicator functions. ■

Appendix B. Properties of the Derived Ranking Distribution D_{gr}

Suppose we have a distribution $D = \langle P, Q, \rho \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$. Assume that M, P, Q have densities μ, p, q . We summarise some properties of the resulting distribution over pairs, $D_{gr} \in \Delta_{\mathcal{X} \times \mathcal{X} \times \{\pm 1\}}$. We will

associate this distribution with the random variable triplet (X, X', Z) . By definition, we have

$$\begin{aligned}\mathbb{P}[Z = 1] &= \frac{1}{2} \\ p_{X|Z=z}(x) &= \mathbb{I}[z = 1] \cdot p(x) + \mathbb{I}[z = -1] \cdot q(x) \\ p_{X'|Z=z}(x') &= \mathbb{I}[z = 1] \cdot q(x') + \mathbb{I}[z = -1] \cdot p(x').\end{aligned}$$

From these, we may derive other marginals and conditionals for D_{BIR} , and relate them to those of D :

$$\begin{aligned}p_{X, X'|Z=z}(x, x') &= p_{X|Z=z}(x) \cdot p_{X'|Z=z}(x') \\ &= \mathbb{I}[z = 1] \cdot p(x) \cdot q(x') + \mathbb{I}[z = -1] \cdot p(x') \cdot q(x) \\ p_{X, X'}(x, x') &= \frac{p(x) \cdot q(x') + p(x') \cdot q(x)}{2} \\ &= \frac{1}{2\pi(1-\pi)} \cdot \mu(x) \cdot \mu(x') \cdot (\eta(x) \cdot (1-\eta(x')) + \eta(x') \cdot (1-\eta(x))) \\ p_X(x) &= \frac{p(x) + q(x)}{2} \\ p_{X|X'=x'}(x) &= \frac{p(x) \cdot q(x') + p(x') \cdot q(x)}{p(x') + q(x)} \\ \mathbb{P}[Z = 1|X = x] &= \frac{p(x)}{p(x) + q(x)} \\ &= \sigma(\sigma^{-1}(\eta(x)) - \sigma^{-1}(\pi)) \\ \mathbb{P}[Z = 1|X = x, X' = x'] &= \frac{p(x) \cdot q(x') + p(x') \cdot q(x)}{p(x') \cdot q(x')} \\ &= \frac{1 + \frac{q(x)}{p(x)} \cdot \frac{p(x')}{q(x')}}{\sigma(\sigma^{-1}(\eta(x)) - \sigma^{-1}(\eta(x')))} \\ &= \sigma(\sigma^{-1}(\eta(x)) - \sigma^{-1}(\eta(x'))).\end{aligned}$$

The last identity follows because

$$\sigma^{-1}(\eta(x)) = \log \frac{\pi}{1-\pi} + \log \frac{p(x)}{q(x)}.$$

Thus, the distributions of primary interest in the paper are:

$$\begin{aligned}P_{\text{pair}} &= P \times Q \\ Q_{\text{pair}} &= Q \times P \\ \pi_{\text{pair}} &= \frac{1}{2} \\ M_{\text{pair}} &= \frac{P \times Q + Q \times P}{2}\end{aligned}$$

$$(\forall x, x' \in \mathcal{X}) \eta_{\text{pair}}(x, x') = \sigma(\sigma^{-1}(\eta(x)) - \sigma^{-1}(\eta(x'))).$$

Appendix C. Properties of the ROC Curve

For completeness we present two well known results about the derivative of the ROC curve, and its relationship to the optimal threshold for cost-sensitive learning.

Proof [Proof of Proposition 8] By the chain rule,

$$(\forall \alpha \in (0, 1)) \rho'(\alpha) = (\text{FPR}^{-1})'(\alpha) \cdot \text{TPR}(\text{FPR}^{-1}(\alpha))$$

$$\begin{aligned}&= \frac{\text{TPR}(\text{FPR}^{-1}(\alpha))}{\text{FPR}(\text{FPR}^{-1}(\alpha))} \\ &= \frac{p_S(\text{FPR}^{-1}(\alpha))}{q_S(\text{FPR}^{-1}(\alpha))} \text{ by Equation 22.}\end{aligned}$$

By Bayes' rule, for a random variable $S \sim S$ for S the distribution of scores,

$$(\forall a \in \mathbb{R}) p_S(a) = p_{S|Y=a}(a) = \frac{\mathbb{P}[Y = 1|S = a] \cdot \mathbb{P}[S = a]}{\mathbb{P}[Y = 1]},$$

and similarly for q_S . Thus

$$\frac{p_S(a)}{q_S(a)} = \frac{1-\pi}{\pi} \cdot \frac{\mathbb{P}[Y = 1|S = a]}{1-\mathbb{P}[Y = 1|S = a]},$$

and by definition, $\text{Prb}(a; D, s) = \mathbb{P}[Y = 1|S = a]$. ■

Proof [Proof of Proposition 11] Let

$$\begin{aligned}(\forall t \in \mathbb{R}) R(t) &\doteq \mathbb{L}(s-t; D, \mathcal{L}_{\text{CS}(c)}) \\ &= \pi \cdot (1-c) \cdot \text{FNR}(t) + (1-\pi) \cdot c \cdot \text{FPR}(t)\end{aligned}$$

be the risk for a fixed scorer s when using a threshold t . Pick any optimal threshold $t_0 \in \mathcal{I}^*(c; D, s)$. This must satisfy

$$0 = R'(t_0) = \pi \cdot (1-c) \cdot \text{FNR}'(t_0) + (1-\pi) \cdot c \cdot \text{FPR}'(t_0),$$

i.e.

$$\frac{\pi}{1-\pi} \cdot \frac{\text{TPR}'(t_0)}{\text{FPR}'(t_0)} = \frac{c}{1-c}.$$

This is exactly

$$\rho'(\text{FPR}(t_0)) = \frac{c}{1-c} \cdot \frac{1-\pi}{\pi}.$$

By Proposition 8, this implies

$$\frac{\text{Prb}(t_0)}{1-\text{Prb}(t_0)} = \frac{c}{1-c}.$$

When $\text{Prb}(\cdot; D, s)$ is invertible, we thus have

$$t_0 = \text{Prb}^{-1}(c),$$

and since the choice of t_0 was arbitrary from the set of optimal thresholds, we conclude that $\mathcal{I}^*(c) = \{\text{Prb}^{-1}(c)\}$. ■

Appendix D. Relationship Between the ROC Curve and the Neyman-Pearson Problem

The maximal ROC curve is intimately related to the following classical hypothesis testing problem. Suppose we have (known) probability distributions P_{+1}, P_{-1} over an instance space \mathcal{X} , with densities p_{+1}, p_{-1} with respect to some reference measure (this could simply be $(P_{+1} + P_{-1})/2$). We are given a sample x drawn from one of $P_{\pm 1}$. We wish to determine whether or not $t = 1$, i.e. conduct a hypothesis test between $H_0 : i = -1$ and $H_1 : i = +1$. The Neyman-Pearson problem (Lehmann and Romano, 2005, pg. 59) asks for the test that has the most *power* in discriminating between the two alternatives, assuming the false positive rate is fixed at some value $\alpha \in [0, 1]$.

Definition 75 (Neyman-Pearson problem) Pick any $\pi \in (0, 1)$, and let $D = \langle P_+, P_- \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$. For a fixed $\alpha \in [0, 1]$, the Neyman-Pearson problem is

$$\max_{h \in \{\pm 1\}^{\mathcal{X}}} \text{TPR}(h; D) : \text{FPR}(h; D) \leq \alpha,$$

where in an abuse of notation

$$\begin{aligned} \text{TPR}(h; D) &= \mathbb{P}_{\mathcal{X} \sim P_+} [h(X) = +1] \\ \text{FPR}(h; D) &= \mathbb{P}_{\mathcal{X} \sim P_-} [h(X) = +1]. \end{aligned}$$

The optimal classifier $h^*(x)$ is called the uniformly most powerful test at α .

From a learning perspective, a test h is simply a classifier $h : \mathcal{X} \rightarrow \{\pm 1\}$ that specifies which of the two hypotheses is preferred. Further, we can view the densities as being class-conditionals $p_y(x) = P_{\mathcal{X}|Y=y}(x)$. Thus, given a distribution $D = \langle P, Q, \pi \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$, the Neyman-Pearson problem arises when we wish to find a classifier that has maximal true positive rate for a fixed false positive rate.

D.1 The Neyman-Pearson Lemma

The Neyman-Pearson lemma (Lehmann and Romano, 2005, pg. 60) specifies the optimal solution to the Neyman-Pearson problem.

Lemma 76 (Neyman-Pearson lemma) Pick any $\pi \in (0, 1)$, and let $D = \langle P_+, P_- \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$ where $P_{\{\pm 1\}}$ have densities $P_{\{\pm 1\}}$ with respect to some reference measure. For any $\alpha \in [0, 1]$, the uniformly most powerful test at α is

$$h^*(x; \alpha, D) = \begin{cases} P_+(x) \\ P_-(x) \end{cases} \geq r^*(\alpha; D) \Bigg]$$

where $r^*(\alpha; D)$ is such that the classifier achieves desired false positive rate,

$$\text{FPR}(h^*(x; \alpha, D); D) = \alpha.$$

92). For completeness, we present a proof based on Lagrange multipliers, following Hippensiefel (2001, pg. 92).

Proof Given a classifier h , let

$$A_h = \{x \in \mathcal{X} : h(x) = +1\}.$$

Then,

$$\begin{aligned} \text{TPR}(h; D) &= \mathbb{P}_{\mathcal{X} \sim P_+} [\mathbf{X} \in A_h] \\ \text{FPR}(h; D) &= \mathbb{P}_{\mathcal{X} \sim P_-} [\mathbf{X} \in A_h]. \end{aligned}$$

Thus, the problem is equivalent to

$$\max_{A \subseteq \mathcal{X}} \int_{x \in A} p_+(x) dx \text{ subject to } \int_{x \in A} p_-(x) dx \leq \alpha.$$

Now consider the Lagrangian

$$\mathcal{L}(A, \lambda) = \int_{x \in A} (p_+(x) - \lambda \cdot p_-(x)) dx + \lambda \cdot \alpha.$$

Clearly, the A which maximises \mathcal{L} is such that the integrand is always nonnegative:

$$A^*(\lambda^*) = \left\{ x \in \mathcal{X} : \frac{p_+(x)}{p_-(x)} \geq \lambda^* \right\} = \left\{ x \in \mathcal{X} : \phi \left(\frac{p_+(x)}{p_-(x)} \right) \geq \phi(\lambda^*) \right\},$$

for any ϕ strictly monotone increasing. Thus, the optimal test or classifier is based on thresholding a scorer of the form

$$s^*(x) = \phi \left(\frac{p_+(x)}{p_-(x)} \right)$$

with the threshold λ^* being the solution to the equation

$$\alpha = \int_{x \in A^*(\lambda^*)} p_-(x).$$

■

In practical learning settings, the optimal solution to the Neyman-Pearson problem cannot be computed as it assumes full knowledge of the underlying distributions. A natural approach is to use empirical versions of the appropriate distributions. For a fixed false positive rate α , various optimisation schemes have been proposed such as neural network based density estimation (Streit, 1990), SVMs (Davenport et al., 2010) and non-convex optimisation (Casso et al., 2011).

D.2 Implications of the Neyman-Pearson Lemma

From a hypothesis testing perspective, the optimal scoring function is seen to be a strictly monotone increasing transform of the likelihood ratio

$$(\forall x \in \mathcal{X}) \Lambda(x) = \frac{p_+(x)}{p_-(x)}.$$

From an ROC perspective, the Neyman-Pearson problem can be seen as picking a particular point on the horizontal axis (by virtue of fixing the FPR), and asking for the scoring function that yields the maximum value along the vertical axis (by virtue of maximising the TPR). The Neyman-Pearson lemma concludes that this is achieved by a strictly monotone increasing transformation of $\Lambda(x)$ regardless of the FPR value. Thus, maximising the AUC can be seen as solving a Neyman-Pearson problem for every possible false positive rate, as in Corollary 16.

Appendix E. Bayes-Optimal Classification Risk and Regret

A classical result establishes that the Bayes 0-1 risk can be expressed in terms of the variational divergence between the class-conditional distributions P and Q (Devroye et al., 1996, pg. 14). This gives an interpretation of the Bayes 0-1 risk, which is the inherent ‘‘difficulty’’ of the problem, in terms of amount of overlap between the distributions for the two classes. In fact, the variational divergence can be replaced by any f -divergence, with the Bayes 0-1 risk being replaced by the Bayes ℓ -classification risk for suitable ℓ .

Proposition 77 (Österreicher and Vajda, 1993; Reid and Williamson, 2011, Theorem 9) For any $D = \langle P, Q, \pi \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$, convex $f : \mathbb{R}_+ \rightarrow \mathbb{R}$, and loss ℓ with conditional Bayes risk

$$(\forall \eta \in (0, 1)) L^*(\eta; \ell) = \frac{1-\eta}{1-\pi} \cdot f \left(\frac{1-\pi}{\pi} \cdot \frac{\eta}{1-\eta} \right),$$

$$L^*(D, \ell) = L^*(\alpha; \ell) - \mathbb{I}_f(P, Q), \tag{67}$$

the Bayes-risk can be written

Conversely, Equation 67 holds for any $D = \langle P, Q, \pi \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$, loss ℓ with concave conditional Bayes risk $L^*(\cdot; \ell) : [0, 1] \rightarrow \mathbb{R}_+$, and $f : \mathbb{R}_+ \rightarrow \mathbb{R}$ defined by

$$(\forall t \in \mathbb{R}) f(t) = L^*(\alpha; \ell) - (\alpha \cdot t + 1 - \pi) \cdot L^* \left(\frac{\pi \cdot t}{\pi \cdot t + 1 - \pi}; \ell \right).$$

Recalling that for a proper composite loss ℓ with underlying proper loss λ , the conditional Bayes risks coincide i.e. $L_\eta^* = L_\eta^*$, we see that for a proper loss λ Proposition 77 holds for any choice of proper composite ℓ resulting from the composition of λ with an invertible link function Ψ .

One can also relate the regret with respect to any proper composite loss to an appropriate generative Bregman divergence.

Proposition 78 (Baja et al., 2005; Reid and Williamson, 2011) For any $D = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$, $\ell \in \mathcal{L}_{\text{SPC}}(\Psi)$, and scorer $s : \mathcal{X} \rightarrow \mathbb{R}$,

$$\text{regret}(s; D, \ell) = \mathbb{E}_{-L^*(\eta, \Psi^{-1} \circ s)}$$

where in an abuse of notation $L^* = L^*(\cdot; \ell)$.

Proposition 78 shows that if a scorer has low ℓ -risk with respect to some proper composite loss, then $\hat{\eta} = \Psi^{-1} \circ s$ is a good estimate of η in a precise sense: it has low average Bregman divergence to η .

Appendix F. Interpretation of Uematsu and Lee (2012) in Terms of Proper Losses

The following are the results shown in Uematsu and Lee (2012).

Proposition 79 (Uematsu and Lee, 2012, Theorem 3) Suppose $\ell(y, v) = \phi(yv)$ for some $\phi : \mathbb{R} \rightarrow \mathbb{R}_+$, where ϕ is differentiable, monotone decreasing, convex, and $\phi'(0) < 0$. For a given distribution $D = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$, let

$$s^* \in S_{\text{BR}}^*(D, \ell).$$

Then,

$$(\forall x, x' \in \mathcal{X}) \eta(x) \neq \eta(x') \implies \text{sign}(\text{Diff}(s^*)(x, x')) = \text{sign}(\eta(x) - \eta(x')).$$

If ϕ is strictly convex, then the above also holds when $\eta(x) = \eta(x')$.

Proposition 80 (Uematsu and Lee, 2012, Theorem 7) Suppose $\ell(y, v) = \phi(yv)$ for some $\phi : \mathbb{R} \rightarrow \mathbb{R}_+$, where ϕ is differentiable, strictly monotone decreasing, convex, and $f : s \mapsto \frac{\phi(-s)}{\phi(s)}$ is strictly increasing. Given any $D = \langle M, \eta \rangle \in \Delta_{\mathcal{X} \times \{\pm 1\}}$,

$$S_{\text{BR}}^*(D, \ell) \subseteq S_{\text{Decomp}}$$

if and only if $\phi'(-s)/\phi'(s) = e^{as}$ for some $a > 0$.

We show how to interpret these results in terms of proper composite losses. First, we show that the conditions of their theorems imply that ℓ is a proper composite margin loss.

Proposition 81 Let ϕ be differentiable with $\phi'(0) < 0$, monotone decreasing, and strictly convex. Then, $\ell(y, v) = \phi(yv)$ is strictly proper composite.

Proof Let ϕ meet the stated conditions. Since ϕ is convex and monotone decreasing with $\phi'(0) < 0$, then it must be true that

$$(\forall v \in \mathbb{R}) (\phi'(v) \neq 0 \vee \phi'(-v) \neq 0).$$

Further, the function

$$f(v) = \frac{\phi'(v)}{\phi'(-v)}$$

is continuous by differentiability of ϕ , and monotone by monotonicity and convexity of ϕ , since

$$f'(v) = \frac{1}{(\phi'(v))^2} \cdot (\phi'(-v)\phi''(v) + \phi'(-v)\phi''(v)) \leq 0.$$

When ϕ is strictly convex, f is strictly monotone because the numerator above cannot be 0. Thus, the conditions of Corollary 16 in Vernet et al. (2011) hold, and so ℓ is strictly proper composite. ■

Lemma 82 Let ϕ be differentiable, strictly monotone decreasing, convex, and such that $f : s \mapsto \frac{\phi'(-s)}{\phi'(s)}$ is strictly increasing. Then, $\ell(y, v) = \phi(yv)$ is strictly proper composite.

Proof The proof follows by the conditions of Corollary 16 in Vernet et al. (2011), as before, with invertibility $f : s \mapsto \frac{\phi'(-s)}{\phi'(s)}$ directly assumed rather than derived as a consequence of strict convexity. ■

By Lemma 73, the statement of Uematsu and Lee (2012, Theorem 3) is equivalent to saying that $\eta = g \circ s^*$ for some non-decreasing g when ϕ is convex, and g is strictly increasing when ϕ is strictly convex. Thus, this strictly convex part of the result is as per Corollary 48, except that the latter explicitly provides the form of the link function relating η and s^* .

The following shows that the conditions in their Theorem 7 imply that the inverse link function is of the form $\Psi^{-1}(v) = \frac{1}{1+e^{-av}}$, which means the result is a special case of Proposition 44 where ℓ is a margin loss.

Lemma 83 Let $\ell \in \mathcal{L}_{\text{SPC}}(\Psi)$ be such that $\ell(y, v) = \phi(yv)$ for some differentiable $\phi : \mathbb{R} \rightarrow \mathbb{R}_+$. Then,

$$(\forall a \in \mathbb{R} \setminus \{0\}) (\forall v \in \mathbb{R}) \Psi^{-1}(v) = \frac{1}{1+e^{-av}} \iff \phi'(-v)/\phi'(v) = e^{av}.$$

Proof The link function for a differentiable proper composite loss satisfies

$$\begin{aligned} (\forall v \in \mathbb{R}) \Psi^{-1}(v) &= \frac{1}{1 - \frac{\phi'(v)}{\phi'(-v)}} \\ &= \frac{1}{1 + \frac{\phi'(v)}{\phi'(-v)}} \\ &= \frac{1}{1 + e^{-av}} \end{aligned}$$

where the last line is true iff the asserted statement holds. ■

Appendix G. Empirical Illustration of Corollary 45

We present an empirical illustration of the fact that Corollary 45 holds for an asymmetric proper composite loss. We work with a discrete distribution over N instances, where the instance i has probability M_i of being drawn, and has an associated probability η_i of having a positive label. A scorer s is then some vector in \mathbb{R}^N . Given a loss ℓ , the bipartite risk of the scorer s is

$$\begin{aligned} \mathbb{L}_{\text{BR}}(s; D, \ell) &= \mathbb{E}_{\mathcal{X} \sim P, \mathcal{X}' \sim Q} [\ell_{\text{symm}}(s(\mathcal{X}) - s(\mathcal{X}'))], \\ &= \sum_{i=1}^N \sum_{j=1}^N [\eta_i \cdot (1 - \eta_j) \cdot (\ell_+(s_i - s_j) + \ell_-(s_j - s_i))] \\ &= \sum_{i=1}^N \sum_{j=1}^N [\eta_i \cdot (1 - \eta_j) \cdot (\ell_+(s, e_i - e_j) + \ell_-(s, e_j - e_i))], \end{aligned}$$

where e_i is the i th standard basis vector in \mathbb{R}^N . The Bayes-optimal risk is simply the minimiser of the above objective, and may be computed by numerical optimisation.

We performed 20 repetitions of the following experiment: for $N = 10$ instances, we draw $\eta_i \sim \text{Beta}(4, 3)$, $Z_i \sim \text{Beta}(6, 2)$, and set $M_i = Z_i / \sum_j Z_j$. We then scaled the η_i 's to lie in $[0.01, 0.99]$, ensuring that the

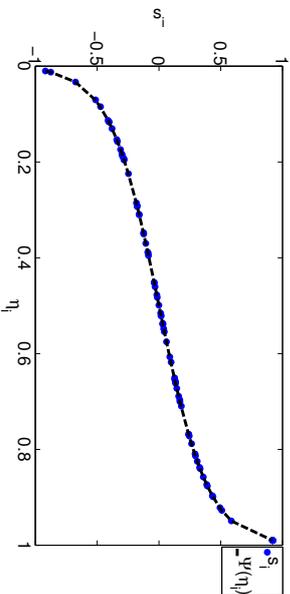


Figure 3: Results of 20 simulation trials to illustrate Bayes-optimal scorer (Corollary 45) for the case of an asymmetric loss. Here, the distribution D is varied across each trial, and the relationship between the (η, s^*) pairs across all trials is plotted. The relationship exactly matches that of $s^* = \Psi(\eta)$.

minimum and maximum values are attained. Given this distribution, we minimised the bipartite risk using L-BFGS, obtaining the Bayes-optimal scorer s^* . As the risk is invariant to translations, we transformed the solution so that its minimum value equals $\Psi(0.01)$ (thus agreeing with that of the expected optimal solution). We collected the corresponding pairs of (η, s^*) values for all 20 repetitions. We then plotted the graph of the resulting η values versus the s^* values. If s^* is a strictly monotone transform of η , then the plot will reflect this (as the different η values from the trials represent different sampling points of the domain of this function).

Figure 3 shows the results where ℓ is the asymmetric p -classification loss for $p = 2$.

$$f(\eta) = \left(\frac{1}{2} \cdot e^{2\eta}, e^{-\eta} \right)$$

We see that the relationship between the two is strictly monotone. Also shown on the graph is the plot of η versus $\Psi(\eta)$, where $\Psi = \frac{1}{2}\sigma^{-1}$; this perfectly agrees with the observed s^* values, as predicted by the theory.

Appendix H. Empirical Illustration of Corollary 48

We now present an empirical illustration of the facts that for a proper composite loss whose Bayes-optimal pair-scorer is non-decomposable, (a) the optimal univariate scorer is a strictly monotone transform of η , and (b) the transformation is distribution dependent. We repeated the setup of Appendix G, except that we worked with ℓ being the squared loss, $f(y, v) = (1 - yv)^2$, and the canonical boosting loss (Bujia et al., 2005).

$$f(y, v) = \frac{yv}{2} + \sqrt{1 + \frac{v^2}{4}}$$

Squared loss employs the identity link, while the canonical boosting loss uses the link $\Psi(\eta) = \frac{2\eta-1}{\sqrt{\eta(1-\eta)}}$, and thus neither induce a decomposable pair-scorer according to Proposition 44.

Figure 4 shows that the relationship between η and s^* for these losses across multiple trials is *not* monotone, and significantly deviates from the optimal solution in the class-probability estimation setting, viz. $s^* = \Psi(\eta)$ for Ψ the identity mapping. This indicates that in general, the relationship between η and s^* is distribution dependent.

Figures 5 and 6 further studies the relationship between the two quantities for each individual trial. We see that, for a given trial (or equivalently for a given distribution), the relationship between η and s^* is strictly

monotone, as expected. However, across different trials, it is evident that the precise monotone transformation is different.

Appendix I. Empirical Illustration of Optimal p -Norm Push Pair-Scorer

We now present an empirical illustration of the fact that for a general proper composite loss, (a) the optimal p -norm push pair-scorer is a strictly monotone transform of η_{pair} and (b) the transformation is distribution dependent. We repeated the setup of Appendix G, except that we worked with the p -norm push risk for $p = 4$, with ℓ being logistic loss, and considered $n = 5$ to reduce the number of η_{pair} values.

Figure 7 shows the relationship between η_{pair} and s_{pair}^* across multiple trials is *not* monotone, and significantly deviates from the optimal solution in the class-probability estimation setting, viz. $s^* = \Psi(\eta)$ for $\Psi = \frac{1}{p}\sigma^{-1}$. This indicates that in general, the relationship between η_{pair} and s_{pair}^* is distribution dependent. Figure 8 further studies the relationship between the two quantities for each individual trial. We see that, for a given trial (or equivalently for a given distribution), the relationship between η_{pair} and s_{pair}^* is strictly monotone, as expected.

Appendix J. Empirical Illustration of Optimal p -Norm Push Univariate Scorer

We now present an empirical illustration of the fact that for a general proper composite loss, the optimal p -norm push univariate scorer is a distribution dependent transform of η . We repeat the setup of Appendix G, except that we worked with the p -norm push risk for $p = 4$, with ℓ being logistic loss, and considered $n = 5$ to reduce the number of η_{pair} values.

Figure 9 shows the relationship between η and s^* across multiple trials is *not* monotone, and significantly deviates from the optimal solution in the class-probability estimation setting, viz. $s^* = \Psi(\eta)$ for $\Psi = \frac{1}{p}\sigma^{-1}$.

Appendix K. Illustration of Hand's Representation and Proper Loss Equivalence

We illustrate empirically that the AUC for certain calibrated scorers is equivalent to a suitable risk with respect to a proper loss, owing to Hand's representation (Equation 34). For a fixed n , we consider a finite instance space $\mathcal{X} = \{0, 1/n, 2/n, \dots, 1\}$. We consider a distribution D over \mathcal{X} where the marginal M is uniform and the class-probability function η is of a pre-specified form. Specifically, we considered $\mathbb{P}(\eta|X) = c] \propto w(c)$, where $w(c)$ is the weight function corresponding to a proper loss. We considered two choices of $w(c) = 1$, corresponding to square loss, and $w(c) = 1/\sqrt{c} \cdot (1 - c)$, corresponding to the arcsin loss of Bujia et al. (2005, Section 11).

For a given n , we then computed the AUC of the scorer $s^* = \eta$, and compared it to the corresponding proper loss of the scorer. Hand's representation (Equation 34) suggests that as $n \rightarrow \infty$, so that the distribution of the scores is exactly the weight of the proper loss, the two will be equivalent.

Figure 10 shows that for a large n , one minus the AUC and the appropriate proper risk converge. The results compare the two over 100 trials, where each trial corresponds to a different random draw of η from the distribution with density given by the appropriately normalised weight $w(\cdot)$.

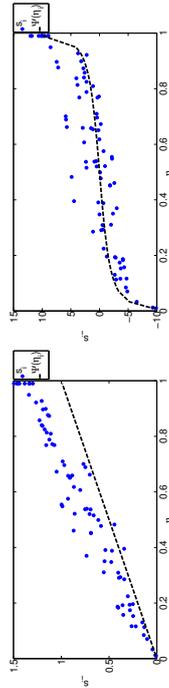


Figure 4: Results of 20 simulation trials to illustrate distribution dependent Bayes-optimal scorer (Proposition 47) for the case of squared and canonical boosting losses. Here, the distribution D is varied across each trial, and the relationship between the (η, s^*) pairs across all trials is plotted.

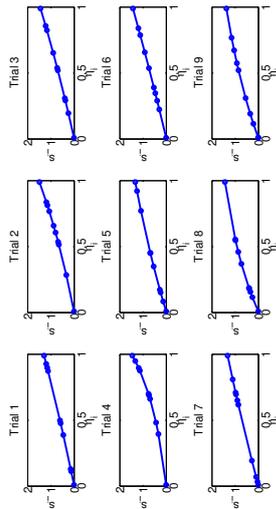


Figure 5: Results of 9 simulation trials to illustrate order preserving Bayes-optimal scorer (Corollary 48) for the case of squared loss. Here, the distribution D is varied across each trial, and each panel represents the relationship between η and s^* for a *specific* trial.

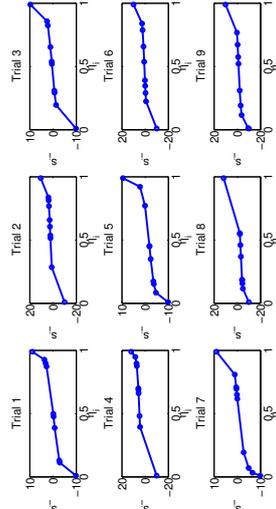


Figure 6: Results of 9 simulation trials to illustrate order preserving Bayes-optimal scorer (Corollary 48) for the case of canonical boosting loss. Here, the distribution D is varied across each trial, and each panel represents the relationship between η and s^* for a *specific* trial.

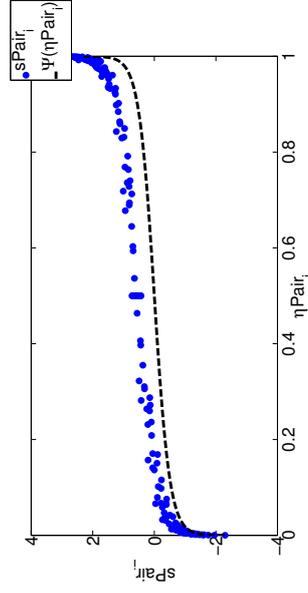


Figure 7: Results of 9 simulation trials to illustrate the relationship between η_{Pair_i} and $s_{\text{Pair}_i}^*$ for p -norm push with logistic loss. Here, the distribution D is varied across each trial, and the relationship between the $(\eta_{\text{Pair}_i}, s_{\text{Pair}_i}^*)$ pairs across all trials is plotted.

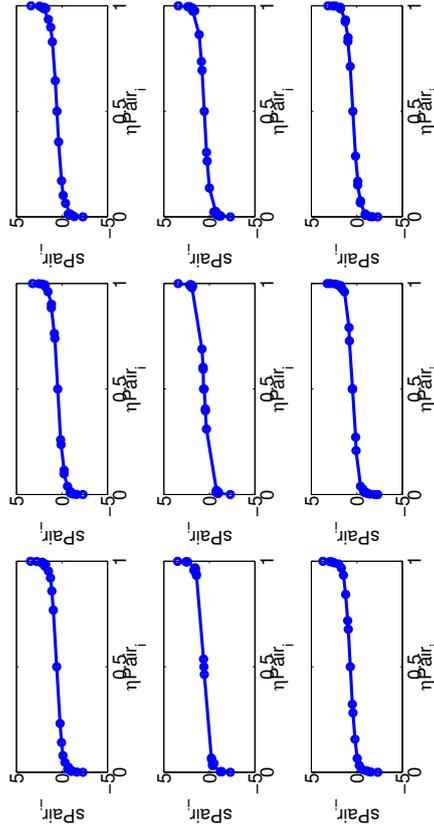


Figure 8: Results of 9 simulation trials to illustrate the distribution dependent relationship between η_{Pair_i} and $s_{\text{Pair}_i}^*$ for p -norm push with logistic loss. Here, the distribution D is varied across each trial, and each panel represents the relationship between η_{Pair_i} and $s_{\text{Pair}_i}^*$ for a *specific* trial.

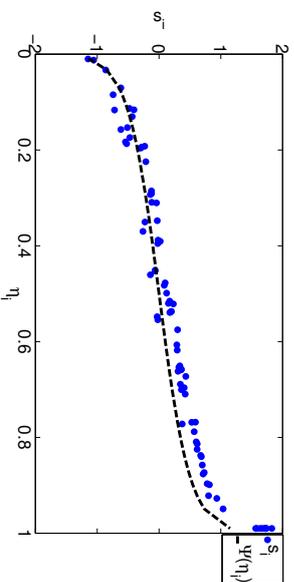


Figure 9: Results of 20 simulation trials to illustrate the relationship between η and s^* for p -norm push with logistic loss. Here, the η_i and M_i values were varied across each trial, and each panel represents the relationship between η and s^* for a specific trial.

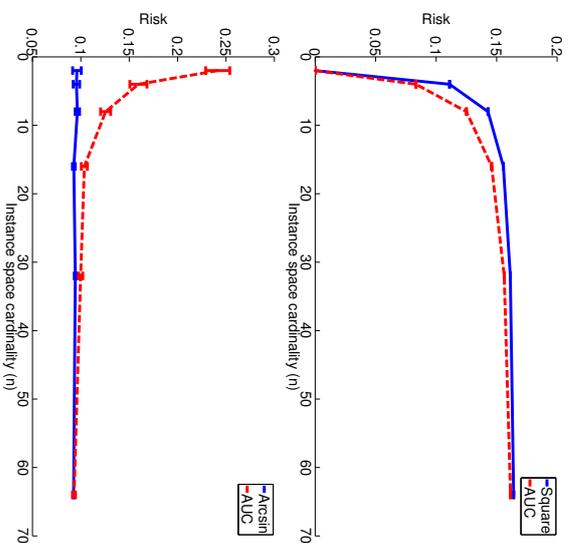


Figure 10: Results of 100 simulation trials to illustrate the relationship between the AUC and a proper risk, for different choices of distribution on the underlying optimal scorer $s^* = \eta$ on an instance space with n elements.

References

- Shivani Agarwal. The infinite push: A new support vector ranking algorithm that directly optimizes accuracy at the absolute top of the list. In *SIAM International Conference on Data Mining (SDM)*, pages 839–850, 2011.
- Shivani Agarwal. Surrogate regret bounds for bipartite ranking via strongly proper losses. *Journal of Machine Learning Research*, 15:1653–1674, 2014.
- Shivani Agarwal and Partha Niyogi. Stability and generalization of bipartite ranking algorithms. In *Conference on Learning Theory (COLT)*, pages 32–47, Berlin, Heidelberg, 2005.
- Shivani Agarwal, Thore Graepel, Ralf Herbrich, Sarel Har-Peled, and Dan Roth. Generalization bounds for the area under the ROC curve. *Journal of Machine Learning Research*, 6:393–425, December 2005.
- Alan Agresti. *Analysis of ordinal categorical data*. Wiley Series in Probability and Statistics, 1984.
- Nir Ailon and Mehryar Mohri. An efficient reduction of ranking to classification. *CoRR*, abs/0710.2889, 2007.
- Miriam Ayer, Hugh D. Brunk, George M. Ewing, William T. Reid, and Edward Silverman. An empirical distribution function for sampling with incomplete information. *The Annals of Mathematical Statistics*, 26(4):641–647, 12 1995.
- Bernard De Baets, Hans De Meyer, and Bart De Schuymer. Transitive comparison of random variables. In Erich Petr Klement and Radko Mesiar, editors, *Logical, Algebraic, Analytic and Probabilistic Aspects of Triangular Norms*, pages 415–442. Elsevier, Amsterdam, 2005.
- Maria-Florina Balcan, Nikhil Bansal, Alina Beygelzimer, Don Coppersmith, John Langford, and Gregory B. Sorkin. Robust reductions from ranking to classification. *Machine Learning*, 72(1-2):139–153, 2008.
- Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.
- Kendrick Boyd, Kevin H. Eng, and C. David Page. Area under the precision-recall curve: Point estimates and confidence intervals. In Hendrik Blockeel, Kristian Kersting, Stegfrid Nijssen, and Filip Zelezny, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 8190 of *Lecture Notes in Computer Science*, pages 451–466. Springer Berlin Heidelberg, 2013.
- Stephen P. Boyd, Corinna Cortes, Mehryar Mohri, and Ana Radovanovic. Accuracy at the top. In *Advances in Neural Information Processing Systems (NIPS)*, pages 962–970, 2012.
- Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs I: The method of paired comparisons. *Biometrika*, 39(3/4):pp.324–345, 1952.
- Lev M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200 – 217, 1967.
- Douglas S. Bridges and Ghanshyam B. Mehta. *Representations of preference orderings*. Lecture notes in economics and mathematical systems. Springer, 1995.
- Andreas Bujia, Werner Suetzle, and Yi Shen. Loss functions for binary class probability estimation and classification: Structure and applications. www-stat.wharton.upenn.edu/~buja/PAPERS/paper-proper-scoring.pdf, 2005. Unpublished manuscript.

- Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *International Conference on Machine Learning (ICML)*, pages 89–96, 2005.
- Soumen Chakrabarti, Rajiv Khanna, Uma Sawant, and Chiru Bhattacharyya. Structured learning for non-smooth ranking losses. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 88–96, 2008.
- Philip K. Chan and Salvatore J. Stolfo. Learning with non-uniform class and cost distributions: Effects and a multi-classifier approach. In *KDD 1998 Workshop on Distributed Data Mining*, pages 1–9, 1998.
- Stéphan Cléménçon and Nicolas Vayatis. Ranking the best instances. *Journal of Machine Learning Research*, 8:2671–2699, December 2007.
- Stéphan Cléménçon and Nicolas Vayatis. Empirical performance maximization for linear rank statistics. In *Advances in Neural Information Processing Systems (NIPS)*, pages 305–312, 2008.
- Stéphan Cléménçon and Nicolas Vayatis. Nonparametric estimation of the precision-recall curve. In *International Conference on Machine Learning (ICML)*, pages 185–192, 2009a.
- Stéphan Cléménçon and Nicolas Vayatis. Adaptive estimation of the optimal ROC curve and a bipartite ranking algorithm. In *International Conference on Algorithmic Learning Theory (ALT)*, pages 216–231, 2009b.
- Stéphan Cléménçon, Gábor Lugosi, and Nicolas Vayatis. Ranking and Empirical Minimization of U-statistics. *The Annals of Statistics*, 36(2):844–874, April 2008.
- Stéphan Cléménçon, Marine Depecker, and Nicolas Vayatis. AUC optimization and the two-sample problem. In *Advances in Neural Information Processing Systems (NIPS)*, pages 360–368, 2009.
- Stéphan Cléménçon and Nicolas Vayatis. Tree-based ranking methods. *IEEE Transactions on Information Theory*, 55(9):4316–4336, Sept 2009.
- Stéphan Cléménçon, Sylvain Robbiano, and Nicolas Vayatis. Ranking data with ordinal labels: optimality and pairwise aggregation. *Machine Learning*, 91(1):67–104, 2013.
- William W. Cohen, Robert E. Schapire, and Yoram Singer. Learning to order things. *Journal of Artificial Intelligence Research*, 10(1):243–270, May 1999.
- Corinna Cortes and Mehryar Mohri. AUC optimization vs. error rate minimization. In *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2003.
- David Cossock and Tong Zhang. Statistical analysis of Bayes optimal subset ranking. *IEEE Transactions on Information Theory*, 54(11):5140–5154, November 2008.
- Koby Crammer and Yoram Singer. Pranking with ranking. In *Advances in Neural Information Processing Systems (NIPS)*, pages 641–647. MIT Press, 2001.
- Imre Csiszár. Eine informationstheoretische Ungleichung und ihre Anwendung auf den Beweis der Ergodizität von Markoffschen Ketten. *A Magyar Tudományos Akadémia Matematikai és Fizikai Tudományok Osztályának Közleményei*, 8:85–108, 1963.
- Mark A. Davenport, Richard G. Baraniuk, and Clayton D. Scott. Tuning support vector machines for minimax and Neyman-Pearson classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1888–1898, October 2010.
- Gerard Debreu. Representation of a preference ordering by a numerical function. In R. M. Thrall, C. H. Coombs, and R. L. Davis, editors, *Decision Processes*, pages 159–65. Wiley, New York, 1954.
- Gerard Debreu. Continuity properties of Paretian utility. *International Economic Review*, 5(3):pp.285–293, 1964.
- Morris H. DeGroot and Stephen E. Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society, Series D (The Statistician)*, 32(1/2):pp.12–22, 1983.
- Luc Devroye, László Györfi, and Gábor Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.
- Lori E. Dodd and Margaret S. Pepe. Partial AUC estimation and regression. *Biometrics*, 59(3):pp.614–623, 2003.
- James P. Egan. *Signal Detection Theory and ROC Analysis*. Series in Cognition and Perception. Academic Press, 1975.
- Samuel Eilenberg. Ordered topological spaces. *American Journal of Mathematics*, 63(1):pp.39–45, 1941.
- Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 213–220, 2008.
- Şeyda Ertekin and Cynthia Rudin. On equivalence relationships between classification and ranking algorithms. *Journal of Machine Learning Research*, 12:2905–2929, Oct 2011.
- Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, June 2006.
- Tom Fawcett and Alexandru Niculescu-Mizil. PAV and the ROC convex hull. *Machine Learning*, 68(1):97–106, 2007.
- César Ferri, Peter Flach, and Ahmane Senad. Modifying ROC curves to incorporate predicted probabilities. In *International Conference on Machine Learning (ICML) Workshop on ROC Analysis in ML*, 2005.
- Peter Flach, José Hernández-Orallo, and César Ferri. A coherent interpretation of AUC as a measure of aggregated classification performance. In *International Conference on Machine Learning (ICML)*, June 2011.
- Peter A. Flach. ROC analysis. In Claude Sammut and Geoffrey I. Webb, editors, *Encyclopedia of Machine Learning*, pages 869–875. Springer, 2010.
- Gerald B. Folland. *Real Analysis: Modern Techniques and Their Applications*. Wiley Interscience, New York, 1999.
- Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, December 2003.
- Johannes Fürnkranz and Eyke Hüllermeier. *Preference Learning*. Springer-Verlag, 1st edition, 2010.
- Wei Gao and Zhi-Hua Zhou. On the consistency of AUC optimization. *CoRR*, abs/1208.0645, 2012.
- Wei Gao and Zhi-Hua Zhou. On the consistency of AUC pairwise optimization. In *International Joint Conference on Artificial Intelligence*, 2015.
- Gilles Gasso, Aristidis Pappaioannou, Marina Spivak, and Léon Bottou. Batch and online learning algorithms for nonconvex Neyman-Pearson classification. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):28:1–28:19, May 2011.

- Izrail M. Gelfand and Sergei V. Fomin. *Calculus of Variations*. Dover, 2000.
- Mariano Giugliotta and Stefan Hildebrandt. *Calculus of Variations I: The Lagrangian formalism*. Springer-Verlag, Berlin, 2nd edition, 2004.
- Tilmann Gneiting and Mathias Katzfuss. Probabilistic forecasting. *Annual Review of Statistics and Its Applications*, 1(1):125–151, 2014.
- Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, March 2007.
- David J. Hand. Measuring classifier performance: a coherent alternative to the area under the ROC curve. *Machine Learning*, 77(1):103–123, October 2009.
- David J. Hand and Robert J. Till. A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning*, 45(2):171–186, 2001.
- James A. Hanley and Barbara J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143:29–36, 1982.
- Ralf Herbrich, Thore Graepel, Peter Bolmann-Scorza, and Klaus Obermayer. Learning Preference Relations for Information Retrieval. In *AAAI Workshop Text Categorization and Machine Learning*, pages 80–84, Madison, 1998.
- Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Large margin rank boundaries for ordinal regression. In A.J. Smola, P.L. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 115–132. MIT Press, Cambridge, MA, 2000.
- José Hernández-Orallo, Peter Flach, and César Ferri. A Unified View of Performance Metrics: Translating Threshold Choice into Expected Classification Loss. *Journal of Machine Learning Research*, 13:2813–2868, October 2012.
- Ralph D. Hippenstiel. *Detection Theory: Applications and Digital Signal Processing*. CRC Press, University of Texas at Tyler, USA, 2001.
- Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, October 2002.
- Thorsten Joachims. Optimizing search engines using clickthrough data. In *ACM SIGKDD International Conference on Knowledge discovery and data mining (KDD)*, pages 133–142, 2002.
- Palaniappan Kannappan. *Functional equations and inequalities with applications*. Springer, New York, 2009.
- Donald E. Knuth. Two notes on notation. *American Mathematical Monthly*, 99(5):403–422, May 1992.
- Wojciech Kotłowski, Krzysztof Dembczynski, and Eryk Hüllermeier. Bipartite ranking through minimization of univariate loss. In *International Conference on Machine Learning (ICML)*, pages 1113–1120, 2011.
- David H. Krantz. Rational distance functions for multidimensional scaling. *Journal of Mathematical Psychology*, 4:226–245, 1967.
- Wojtek J. Krzanowski and David J. Hand. *ROC Curves for Continuous Data*. Chapman & Hall/CRC, 1st edition, 2009.
- John Langford and Bianca Zadrozny. Estimating class membership probabilities using classifier learners. In *International Workshop on Artificial Intelligence and Statistics (AISTATS)*, 2005.
- Erik L. Lehmann and Joseph P. Romano. *Testing statistical hypotheses*. Springer Texts in Statistics, Springer, New York, third edition, 2005.
- Ling Li and Hsuan-Tien Lin. Ordinal regression by extended binary classification. In *Advances In Neural Information Processing Systems (NIPS)*, pages 865–872, 2006.
- Nan Li, Rong Jin, and Zhi-Hua Zhou. Top rank optimization in linear time. *Advances in Neural Information Processing Systems*, pages 1–9, 2014.
- Charles X. Ling and Chenghui Li. Data mining for direct marketing: Problems and solutions. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 217–225, 1998.
- Te-Yan Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, March 2009.
- Robert Duncan Luce. *Individual Choice Behavior*. Wiley, New York, 1959.
- Robert Duncan Luce and Patrick Suppes. Preference, Utility, and Subjective Probability. *Handbook of mathematical psychology*, 3(171):249–410, 1965.
- Henry B. Mann and Donald R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18(1):50–60, 03 1947.
- Hamed Masnadi-Shirazi and Nuno Vasconcelos. Variable margin losses for classifier design. In *Advances In Neural Information Processing Systems (NIPS)*, pages 1576–1584, 2010.
- Donna Kazman McClish. Analyzing a portion of the ROC curve. *Medical Decision Making*, 9(3):190–195, 1989.
- Aditya Krishna Menon and Robert C. Williamson. Bayes-optimal scorers for bipartite ranking. In *Conference on Learning Theory (COLT)*, 2014.
- Aditya Krishna Menon, Brendan van Rooyen, Cheng Soon Ong, and Robert C. Williamson. Learning from corrupted binary labels via class-probability estimation. In *International Conference on Machine Learning (ICML)*, pages 125–134, 2015.
- Harkrishna Narasimhan and Shivani Agarwal. SVM_{hinge}: a new support vector method for optimizing partial AUC based on a tight convex upper bound. In *ACM SIGKDD International Conference on Knowledge discovery and data mining (KDD)*, pages 167–175, 2013a.
- Harkrishna Narasimhan and Shivani Agarwal. On the relationship between binary classification, bipartite ranking, and binary class probability estimation. In *Advances In Neural Information Processing Systems (NIPS)*, pages 2913–2921, 2013b.
- Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006.
- Ferdinand Oesterreicher and Igor Vajda. Statistical information and discrimination. *IEEE Transactions on Information Theory*, 39(3):1036–1039, 1993.
- John R. Platt. Strong inference. *Science*, 146(3642):347–353, October 1962.
- Foster Provost and Tom Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42(3):203–231, March 2001.
- Mark D. Reid and Robert C. Williamson. Surrogate regret bounds for proper losses. In *International Conference on Machine Learning (ICML)*, pages 897–904, 2009.

- Mark D. Reid and Robert C. Williamson. Composite binary losses. *Journal of Machine Learning Research*, 11:2387–2422, December 2010.
- Mark D. Reid and Robert C. Williamson. Information, divergence and risk for binary experiments. *Journal of Machine Learning Research*, 12:731–817, Mar 2011.
- Fred S. Roberts. *Measurement theory with Applications to Decision Making, Utility, and the Social Sciences*, volume 7 of *Encyclopedia of Mathematics and Its Applications*. Addison-Wesley, Reading, MA, 1984.
- Cynthia Rudin. The p-norm push: A simple convex ranking algorithm that concentrates at the top of the list. *Journal of Machine Learning Research*, 10:2233–2271, December 2009.
- Walter Rudin. *Functional Analysis*. McGraw-Hill Book Co., New York, 2nd edition, 1973. McGraw-Hill Series in Higher Mathematics.
- Mark J. Schervish. A general method for comparing probability assessors. *The Annals of Statistics*, 17(4):1856–1879, 12 1989.
- Clayton Scott and Mark Davenport. Regression level set estimation via cost sensitive classification. *IEEE Transactions on Signal Processing*, 55:2752–2757, 2007.
- Martin J. J. Scott, Mahesan Niranjan, and Richard W. Prager. Realisable classifiers: Improving operating performance on variable cost problems. In *British Machine Vision Conference*, pages 304–315, 1998.
- Sundararajan Sellamanickam, Priyanka Garg, and Sathya Keerthi Selvaraj. A pairwise ranking based approach to learning with positive and unlabeled examples. In *ACM International Conference on Information and Knowledge Management (CIKM)*, pages 663–672, 2011.
- Amnon Shashua and Anat Levin. Ranking with large margin principle: Two approaches. In *Advances In Neural Information Processing Systems (NIPS)*, pages 937–944, 2002.
- Emir H. Shuford Jr., Arthur Albert, and H. Edward Massengill. Admissible probability measurement procedures. *Psychometrika*, 31(2):125–145, 1966.
- Ingo Steinwart. How to compare different loss functions and their risks. *Constructive Approximation*, 26(2):225–287, 2007.
- Roy L. Streit. A neural network for optimum Neyman-Pearson classification. In *International Joint Conference on Neural Networks (IJCNN)*, volume 1, pages 685–690, 1990.
- Robert S. Strichartz. *A Guide to Distribution Theory and Fourier Transforms*. Studies in Advanced Mathematics. CRC Press, Boca Raton, FL, 1994.
- S. Joshua Swamidass, Chloé-Agathe Azencott, Kenny Dailly, and Pierre Baldi. A CROC stronger than ROC. *Bioinformatics*, 26(10):1348–1356, May 2010.
- Zbigniew Świtalski. General transitivity conditions for fuzzy reciprocal preference matrices. *Fuzzy Sets and Systems*, 137(1):85–100, 2003.
- Erik N. Torgersen. *Comparison of Statistical Experiments*. Cambridge University Press, 1991.
- John L. Troutman. *Variational Calculus and Optimal Control: Optimization with Elementary Convexity*. Undergraduate Texts in Mathematics. Springer, 1996.
- Kazuki Uematsu and Yoonyoung Lee. On theoretically optimal ranking functions in bipartite ranking. <http://www.stat.osu.edu/~ykleee/mss/bipartitrank.rev.pdf>, 2012. Unpublished manuscript.
- Elodie Vernet, Mark D. Reid, and Robert C. Williamson. Composite multiclass losses. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1224–1232, 2011.
- Ellen M. Voorhees. The TREC question answering track. *Natural Language Engineering*, 7(4):361–378, December 2001. ISSN 1351-3249.
- Shaomin Wu and Peter Flach. A scored AUC metric for classifier evaluation and selection. In *International Conference on Machine Learning (ICML) Workshop on ROC Analysis in ML*, 2005.
- Jingdong Xie and Carey E. Priebe. A weighted generalization of the Mann-Whitney-Wilcoxon statistic. *Journal of Statistical Planning and Inference*, 102(2):441 – 466, 2002.
- Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. A support vector method for optimizing average precision. In *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 271–278, 2007.
- Tong Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *The Annals of Statistics*, 32:56–134, March 2004.

Bayesian group factor analysis with structured sparsity

Shiwen Zhao

Computational Biology and Bioinformatics Program

Department of Statistical Science

Duke University

Durham, NC 27708, USA

SHIWEN.ZHAO@DUKE.EDU

Chuan Gao

Department of Statistical Science

Duke University

Durham, NC 27708, USA

CHUAN.GAO@DUKE.EDU

Sayan Mukherjee

Departments of Statistical Science, Computer Science, Mathematics

Duke University

Durham, NC 27708, USA

SAYAN@STAT.DUKE.EDU

Barbara E Engelhardt

Department of Computer Science

Center for Statistics and Machine Learning

Princeton University

Princeton, NJ 08540, USA

BEE@PRINCETON.EDU

Editor: Samuel Kaski

Abstract

Latent factor models are the canonical statistical tool for exploratory analyses of low-dimensional linear structure for a matrix of p features across n samples. We develop a structured Bayesian group factor analysis model that extends the factor model to multiple coupled observation matrices; in the case of two observations, this reduces to a Bayesian model of canonical correlation analysis. Here, we carefully define a structured Bayesian prior that encourages both element-wise and column-wise shrinkage and leads to desirable behavior on high-dimensional data. In particular, our model puts a structured prior on the joint factor loading matrix, regularizing at three levels, which enables element-wise sparsity and unsupervised recovery of latent factors corresponding to structured variance across arbitrary subsets of the observations. In addition, our structured prior allows for both dense and sparse latent factors so that covariation among either all features or only a subset of features can be recovered. We use fast parameter-expanded expectation-maximization for parameter estimation in this model. We validate our method on simulated data with substantial structure. We show results of our method applied to three high-dimensional data sets, comparing results against a number of state-of-the-art approaches. These results illustrate useful properties of our model, including i) recovering sparse signal in the presence of dense effects; ii) the ability to scale naturally to large numbers of observations; iii) flexible observation- and factor-specific regularization to recover factors with a wide variety of sparsity levels and percentage of variance explained; and iv) tractable inference that scales to modern genomic and text data sizes.

Keywords: Bayesian structured sparsity, canonical correlation analysis, sparse priors, sparse and low-rank matrix decomposition, mixture models, parameter expansion

1. Introduction

Factor analysis models have attracted attention recently due to their ability to perform exploratory analyses of the latent linear structure in high-dimensional data (West, 2003; Carvalho et al., 2008; Engelhardt and Stephens, 2010). A latent factor model finds a low-dimensional representation $\mathbf{x}_i \in \mathbb{R}^{k \times 1}$ of high-dimensional data with p features, $\mathbf{y}_i \in \mathbb{R}^{p \times 1}$ in $i = 1, \dots, n$ samples. A sample in the low-dimensional space is linearly projected to the original high-dimensional space through a *loadings matrix* $\mathbf{A} \in \mathbb{R}^{p \times k}$ with Gaussian noise $\boldsymbol{\epsilon}_i \in \mathbb{R}^{p \times 1}$:

$$\mathbf{y}_i = \mathbf{A}\mathbf{x}_i + \boldsymbol{\epsilon}_i, \quad (1)$$

for $i = 1, \dots, n$. It is often assumed that \mathbf{x}_i follows a $\mathcal{N}_k(\mathbf{0}, \mathbf{I}_k)$ distribution, where \mathbf{I}_k is the identity matrix of dimension k , and $\boldsymbol{\epsilon}_i \sim \mathcal{N}_p(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is a $p \times p$ diagonal covariance matrix with σ_j^2 for $j = 1, \dots, p$ on the diagonal. In many applications of factor analysis, the number of latent factors k is much smaller than the number of features p and the number of samples n . Integrating over factor \mathbf{x}_i , this model produces a low-rank estimation of the feature covariance matrix. In particular, the covariance of \mathbf{y}_i , $\boldsymbol{\Omega} \in \mathbb{R}^{p \times p}$, is estimated as

$$\boldsymbol{\Omega} = \mathbf{A}\mathbf{A}^T + \boldsymbol{\Sigma} = \sum_{h=1}^k \boldsymbol{\lambda}_h \boldsymbol{\lambda}_h^T + \boldsymbol{\Sigma},$$

where $\boldsymbol{\lambda}_h$ is the h^{th} column of \mathbf{A} . This factorization suggests that each factor contributes to the covariance of the sample through its corresponding loading. Traditional exploratory data analysis methods including principal component analysis (PCA) (Hotelling, 1933), independent component analysis (ICA) (Comon, 1994), and canonical correlation analysis (CCA) (Hotelling, 1936) all have interpretations as latent factor models. Indeed, the field of latent variable models is extremely broad, and robust unifying frameworks are desirable (Cunningham and Ghahramani, 2015).

Considering latent factor models (Equation 1) as capturing a low-rank estimate of the feature covariance matrix, we can characterize canonical correlation analysis (CCA) as modeling paired observations $\mathbf{y}_i^{(1)} \in \mathbb{R}^{p_1 \times 1}$ and $\mathbf{y}_i^{(2)} \in \mathbb{R}^{p_2 \times 1}$ across n samples to identify a linear latent space for which the correlations between the two observations are maximized (Hotelling, 1936; Bach and Jordan, 2005). The Bayesian CCA (BCCA) model extends this covariance representation to two observations: the combined loading matrix jointly models covariance structure shared across both observations and covariance local to each observation (Klami et al., 2013). Group factor analysis (GFA) models further extend this representation to m coupled observations for the same sample, modeling, in its fullest generality, the covariance associated with every subset of observations (Virtanen et al., 2012; Klami et al., 2014b). GFA becomes intractable when m is large due to exponential explosion of covariance matrices to estimate.

In a latent factor model, the loading matrix \mathbf{A} plays an important role in the subspace mapping. In applications where there are fewer samples than features—the $n \ll p$ scenario (West, 2003)—it is essential to include strong regularization on the loading matrix

because the optimization problem is under-constrained and has many equivalent solutions that optimize the data likelihood. In the machine learning and statistics literature, priors or penalties are used to regularize the elements of the loading matrix, occasionally by inducing sparsity. Element-wise sparsity corresponds to *feature selection*. This has the effect that a latent factor contributes to variation in only a subset of the observed features, generating interpretable results (West, 2003; Carvalho et al., 2008; Knowles and Ghahramani, 2011). For example, in gene expression analysis, sparse factor loadings are interpreted as non-disjoint clusters of co-regulated genes (Pounnara and Wernisch, 2007; Lucas et al., 2010; Gao et al., 2013).

Element-wise sparsity has been imposed in latent factor models through regularization via l_1 type penalties (Zou et al., 2006; Witten et al., 2009; Salzmänn et al., 2010). More recently, Bayesian shrinkage methods using sparsity-inducing priors have been introduced for latent factor models (Archambeau and Bach, 2009; Carvalho et al., 2008; Virtanen et al., 2012; Bhattacharya and Dunson, 2011; Klami et al., 2013). The spike-and-slab prior (Mitchell and Beauchamp, 1988), the classic two-groups Bayesian sparsity-inducing prior, has been used for sparse Bayesian latent factor models (Carvalho et al., 2008). A computationally tractable one-group prior, the automatic relevance determination (ARD) prior (Neal, 1995; Tipping, 2001), has also been used to induce sparsity in latent factor models (Engelhardt and Stephens, 2010; Pritchard-Malinici et al., 2011). More sophisticated structured regularization approaches for linear models have been studied in classical statistics (Zou and Hastie, 2005; Kowalski and Torrésani, 2009; Jenatton et al., 2011; Huang et al., 2011).

Global structured regularization of the loading matrix, in fact, has been used to extend latent factor models to multiple observations. The BCCA model (Klami et al., 2013) assumes a latent factor model for each observation through a shared latent vector $\mathbf{x}_i \in \mathbb{R}^{k \times 1}$. This BCCA model may be written as a latent factor model by vertical concatenation of observations, loading matrices, and Gaussian residual errors. By inducing group-wise sparsity—explicit blocks of zeros—in the combined loading matrix, the covariance shared across the two observations and the covariance local to each observation are estimated (Klami and Kaski, 2008; Klami et al., 2013). Extensions of this approach to multiple coupled observations $\mathbf{y}_i^{(1)} \in \mathbb{R}^{p \times 1}, \dots, \mathbf{y}_i^{(m)} \in \mathbb{R}^{p_m \times 1}$ have resulted in group factor analysis models (GFA) (Archambeau and Bach, 2009; Salzmänn et al., 2010; Jia et al., 2010; Virtanen et al., 2012).

In addition to linear factor models, flexible non-linear latent factor models have been developed. The Gaussian process latent variable model (GPLVM) (Lawrence, 2005) extends Equation (1) to non-linear mappings with a Gaussian process prior on latent variables. Extensions of GPLVM include models that allow multiple observations (Shon et al., 2005; Ek et al., 2008; Salzmänn et al., 2010; Danahon et al., 2012). Although our focus will be on linear maps, we will keep the non-linear possibility open for model extensions, and we will include the GPLVM model in our model comparisons.

The primary contribution of this study is that we develop a GFA model using Bayesian shrinkage with hierarchical structure that encourages both element-wise and column-wise sparsity; the resulting flexible Bayesian GFA model is called BASS (Bayesian group factor Analysis with Structured Sparsity). The structured sparsity in our model is achieved with multi-scale application of a hierarchical sparsity-inducing prior that has a computa-

tionally tractable representation as a scale mixture of normals; the three parameter beta prior (\mathcal{TPB}) (Armagan et al., 2011; Gao et al., 2013). Our BASS model i) shrinks the loading matrix globally; removing factors that are not supported in the data; ii) shrinks loading columns to decouple latent spaces from arbitrary subsets of observations; iii) allows factor loadings to have either an element-wise sparse or a non-sparse prior, combining interpretability with dimension reduction. In addition, we developed a parameter-expanded expectation maximization (PX-EM) method based on rotation augmentation to tractably find *maximum a posteriori* estimates of the model parameters (Rocková and George, 2015). PX-EM has the same computational complexity as the standard EM algorithm, but produces more robust solutions by enabling fast searching over posterior modes.

In Section 2 we review current work in sparse latent factor models and describe our BASS model. In Sections 3 and 4, we briefly review Bayesian shrinkage priors and introduce the structured hierarchical prior in BASS. In Section 5, we introduce our PX-EM algorithms for parameter estimation. In Section 6, we show the behavior of our model for recovering simulated sparse signals among m observation matrices and compare the results from BASS with state-of-the-art methods. In Section 7, we present results that illustrate the performance of BASS on three high-dimensional data sets. We first show that the estimates of shared factors from BASS can be used to perform multi-label learning and prediction in the Mutan Library data and the 20 Newsgroups data. Then we demonstrate that BASS can be used to find biologically meaningful structure and construct condition-specific co-regulated gene networks using the sparse factors specific to observations. We conclude by considering possible extensions to this model in Section 8.

2. Bayesian group factor model

Here, we review current work in sparse latent factor models and describe our Bayesian group factor Analysis with Structured Sparsity (BASS) model in the context of related work.

2.1 Latent factor models

Factor analysis has been extensively used for dimension reduction and low-dimensional covariance matrix estimation. For concreteness, we re-write the basic factor analysis model here as

$$\mathbf{y}_i = \mathbf{A}\mathbf{x}_i + \epsilon_i,$$

where $\mathbf{y}_i \in \mathbb{R}^{p \times 1}$ is modeled as a linear transformation of a latent vector $\mathbf{x}_i \in \mathbb{R}^{k \times 1}$ through loading matrix $\mathbf{A} \in \mathbb{R}^{p \times k}$ (Figure 1A). Here, \mathbf{x}_i is assumed to follow a $\mathcal{N}_{\mathbf{A}}(\mathbf{0}, \mathbf{I}_k)$ distribution, where \mathbf{I}_k is the k -dimensional identity matrix, and $\epsilon_i \sim \mathcal{N}_\epsilon(\mathbf{0}, \mathbf{\Sigma})$, where $\mathbf{\Sigma}$ is a $p \times p$ diagonal matrix. With an isotropic noise assumption, $\mathbf{\Sigma} = \mathbf{I}_p \sigma^2$, this model has a probabilistic principal components interpretation (Roweis, 1998; Tipping and Bishop, 1999b). For factor analysis, and in this work, it is assumed that $\mathbf{\Sigma} = \text{diag}(\sigma_1^2, \dots, \sigma_p^2)$ representing independent idiosyncratic noise (Tipping and Bishop, 1999a).

Integrating over the factors \mathbf{x}_i , we see that the covariance of \mathbf{y}_i is estimated with a low-rank matrix factorization: $\mathbf{A}\mathbf{A}^T + \mathbf{\Sigma}$. We further let $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_n]$ be the collection of n samples \mathbf{y}_i , and similarly let $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]$ and $\mathbf{E} = [\epsilon_1, \dots, \epsilon_n]$. Then the factor

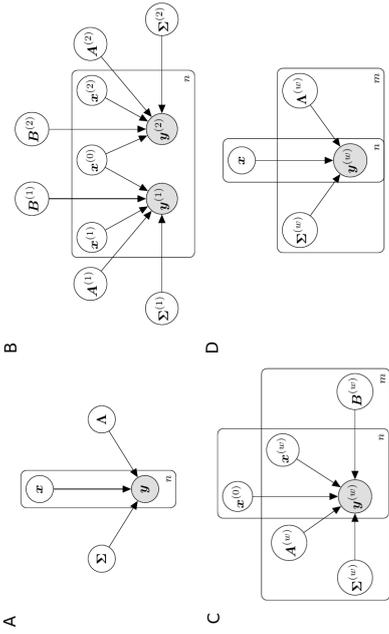


Figure 1: **Graphical representation of different latent factor models.** Panel A: Factor analysis model. Panel B: Bayesian canonical correlation analysis model (BCCA). Panel C: An extension of BCCA model to multiple observations. Panel D: Our Bayesian group factor analysis model (BASS).

analysis model for the observation \mathbf{Y} is written as

$$\mathbf{Y} = \mathbf{\Lambda}\mathbf{X} + \mathbf{E}. \quad (2)$$

2.2 Probabilistic canonical correlation analysis

In the context of two paired observations $\mathbf{y}_i^{(1)} \in \mathbb{R}^{p_1 \times 1}$ and $\mathbf{y}_i^{(2)} \in \mathbb{R}^{p_2 \times 1}$ on the same n samples, canonical correlation analysis (CCA) seeks to find linear projections (canonical directions) such that the sample correlations in the projected space are mutually maximized (Hotelling, 1936). The work of interpreting CCA as a probabilistic model can be traced back to classical descriptions (Bach and Jordan, 2005). With a common latent factor, $\mathbf{x}_i \in \mathbb{R}^{k \times 1}$, $\mathbf{y}_i^{(1)}$ and $\mathbf{y}_i^{(2)}$ are modeled as

$$\begin{aligned} \mathbf{y}_i^{(1)} &= \mathbf{\Lambda}^{(1)}\mathbf{x}_i + \boldsymbol{\epsilon}_i^{(1)}, \\ \mathbf{y}_i^{(2)} &= \mathbf{\Lambda}^{(2)}\mathbf{x}_i + \boldsymbol{\epsilon}_i^{(2)}. \end{aligned} \quad (3)$$

In this model, the errors are distributed as $\boldsymbol{\epsilon}_i^{(1)} \sim \mathcal{N}_{p_1}(\mathbf{0}, \boldsymbol{\Psi}^{(1)})$ and $\boldsymbol{\epsilon}_i^{(2)} \sim \mathcal{N}_{p_2}(\mathbf{0}, \boldsymbol{\Psi}^{(2)})$, where $\boldsymbol{\Psi}^{(1)}$ and $\boldsymbol{\Psi}^{(2)}$ are positive semi-definite matrices, and not necessarily diagonal, allowing dependencies among the residual errors within an observation. The maximum likelihood estimates of the loading matrices in the classical CCA framework, $\mathbf{\Lambda}^{(1)}$ and $\mathbf{\Lambda}^{(2)}$, are the first k canonical directions up to orthogonal transformations (Bach and Jordan, 2005).

2.3 Bayesian CCA with group-wise sparsity

Building on the probabilistic CCA model, a Bayesian CCA (BCCA) model has the following form (Klami et al., 2013)

$$\begin{aligned} \mathbf{y}_i^{(1)} &= \mathbf{A}^{(1)}\mathbf{x}_i^{(0)} + \mathbf{B}^{(1)}\mathbf{x}_i^{(1)} + \boldsymbol{\epsilon}_i^{(1)}, \\ \mathbf{y}_i^{(2)} &= \mathbf{A}^{(2)}\mathbf{x}_i^{(0)} + \mathbf{B}^{(2)}\mathbf{x}_i^{(2)} + \boldsymbol{\epsilon}_i^{(2)}, \end{aligned} \quad (4)$$

with $\mathbf{x}_i^{(0)} \in \mathbb{R}^{k_0 \times 1}$, $\mathbf{x}_i^{(1)} \in \mathbb{R}^{k_1 \times 1}$ and $\mathbf{x}_i^{(2)} \in \mathbb{R}^{k_2 \times 1}$ (Figure 1B). The latent vector $\mathbf{x}_i^{(0)}$ is shared by both $\mathbf{y}_i^{(1)}$ and $\mathbf{y}_i^{(2)}$, and captures their common variation through loading matrices $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$. Two additional latent vectors, $\mathbf{x}_i^{(1)}$ and $\mathbf{x}_i^{(2)}$, are specific to each observation; they are multiplied by observation-specific loading matrices $\mathbf{B}^{(1)}$ and $\mathbf{B}^{(2)}$. The two residual error terms are $\boldsymbol{\epsilon}_i^{(1)} \sim \mathcal{N}_{p_1}(\mathbf{0}, \boldsymbol{\Sigma}^{(1)})$ and $\boldsymbol{\epsilon}_i^{(2)} \sim \mathcal{N}_{p_2}(\mathbf{0}, \boldsymbol{\Sigma}^{(2)})$, where $\boldsymbol{\Sigma}^{(1)}$ and $\boldsymbol{\Sigma}^{(2)}$ are diagonal matrices. This model was originally called inter-battery factor analysis (IBFA) (Browne, 1979) and recently has been studied under a full Bayesian inference framework (Klami et al., 2013). It may be interpreted as the probabilistic CCA model (Equation 3) with an additional low-rank factorization of the observation-specific error covariance matrices. In particular, we re-write the residual error term specific to observation w ($w = 1, 2$) from the probabilistic CCA model (Equation 3) as $\boldsymbol{\epsilon}_i^{(w)} = \mathbf{B}^{(w)}\mathbf{x}_i^{(w)} + \boldsymbol{\epsilon}_i^{(w)}$; then marginally $\boldsymbol{\epsilon}_i^{(w)} \sim \mathcal{N}_{p_w}(\mathbf{0}, \boldsymbol{\Psi}^{(w)})$ where $\boldsymbol{\Psi}^{(w)} = \mathbf{B}^{(w)}(\mathbf{B}^{(w)})^T + \boldsymbol{\Sigma}^{(w)}$.

Recent work has re-written the BCCA model as a factor analysis model with group-wise sparsity in the loading matrix (Klami et al., 2013). Let $\mathbf{y}_i \in \mathbb{R}^{p \times 1}$ (where $p = p_1 + p_2$) be the vertical concatenation of $\mathbf{y}_i^{(1)}$ and $\mathbf{y}_i^{(2)}$; let $\mathbf{x}_i \in \mathbb{R}^{k \times 1}$ (where $k = k_0 + k_1 + k_2$) be the vertical concatenation of $\mathbf{x}_i^{(0)}$, $\mathbf{x}_i^{(1)}$ and $\mathbf{x}_i^{(2)}$; and let $\boldsymbol{\epsilon}_i \in \mathbb{R}^{p \times 1}$ be the vertical concatenation of the two residual errors. Then, the BCCA model (Equation 4) may be written as a factor analysis model

$$\mathbf{y}_i = \mathbf{\Lambda}\mathbf{x}_i + \boldsymbol{\epsilon}_i,$$

with $\boldsymbol{\epsilon}_i \sim \mathcal{N}_p(\mathbf{0}, \boldsymbol{\Sigma})$, where

$$\mathbf{\Lambda} = \begin{bmatrix} \mathbf{A}^{(1)} & \mathbf{B}^{(1)} & \mathbf{0} \\ \mathbf{A}^{(2)} & \mathbf{0} & \mathbf{B}^{(2)} \end{bmatrix}, \quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}^{(1)} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}^{(2)} \end{bmatrix}.$$

The structure in the loading matrix $\mathbf{\Lambda}$ has a specific meaning: the non-zero columns (i.e., $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$) project the shared latent factors (i.e., the first k_0 elements of \mathbf{x}_i) to $\mathbf{y}_i^{(1)}$ and $\mathbf{y}_i^{(2)}$, respectively; these latent factors represent the covariance shared across the observations. The columns with zero blocks (i.e., $[\mathbf{B}^{(1)}; \mathbf{0}]$ or $[\mathbf{0}; \mathbf{B}^{(2)}]$) relate factors to only one of the two observations; they model covariance specific to that observation. Under this model, the block sparse structure of $\mathbf{\Lambda}$ is imposed via observation-wise sparsity on each factor.

2.4 Extensions to multiple observations

Classical and Bayesian extensions of the CCA model to allow multiple observations ($m > 2$) have been proposed (McDonald, 1970; Browne, 1980; Archambeau and Bach, 2009; Qu and

Chen, 2011; Ray et al., 2014). Generally, these approaches partition the latent variables into those that are shared and those that are observation-specific as follows:

$$\mathbf{y}_i^{(w)} = \mathbf{A}^{(w)} \mathbf{x}_i^{(0)} + \mathbf{B}^{(w)} \mathbf{x}_i^{(w)} + \boldsymbol{\epsilon}_i^{(w)} \quad \text{for } w = 1, \dots, m.$$

By vertical concatenation of $\mathbf{y}_i^{(w)}$, $\mathbf{x}_i^{(w)}$ and $\boldsymbol{\epsilon}_i^{(w)}$, this model can be viewed as a latent factor model (Equation 1) with the joint loading matrix \mathbf{A} having a similar observation-wise sparsity pattern as the BCGA model

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}^{(1)} & \mathbf{B}^{(1)} & \dots & \mathbf{0} \\ \mathbf{A}^{(2)} & \mathbf{0} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{(m)} & \mathbf{0} & \dots & \mathbf{B}^{(m)} \end{bmatrix}. \quad (5)$$

Here, the first column of blocks ($\mathbf{A}^{(w)}$) is a non-zero loading matrix across the features of all observations; the remaining columns have a block diagonal structure with observation-specific loading matrices ($\mathbf{B}^{(w)}$) on the diagonal. However, those extensions are limited by the strict diagonal structure of the loading matrix. Structuring the loading matrix in this way prevents this model from capturing covariance structure among arbitrary subsets of observations. On the other hand, there are an exponential number of possible subsets intractable for large m .

The structure on \mathbf{A} in Equation (5) has been relaxed to model covariance among subsets of the observations (Jia et al., 2010; Virtanen et al., 2012; Klami et al., 2014b). In the relaxed formulation, each observation $\mathbf{y}_i^{(w)}$ is modeled by its own loading matrix $\mathbf{A}^{(w)}$ and a shared latent vector \mathbf{x}_i (Figure 1D):

$$\mathbf{y}_i^{(w)} = \mathbf{A}^{(w)} \mathbf{x}_i + \boldsymbol{\epsilon}_i^{(w)} \quad \text{for } w = 1, \dots, m. \quad (6)$$

By allowing columns in $\mathbf{A}^{(w)}$ to be zero, the model decouples certain latent factors from certain observations. The covariance structure of an arbitrary subset of observations is modeled by factors with non-zero loading columns corresponding to the observations in that subset. Factors that correspond to non-zero entries for only one observation capture covariance specific to that observation. Two different approaches have been proposed to achieve column-wise shrinkage in this framework: Bayesian shrinkage (Virtanen et al., 2012; Klami et al., 2014b) and explicit penalties (Jia et al., 2010). The group factor analysis (GFA) model puts an ARD prior (Tipping, 2001) on the loading column for each observation to allow column-wise shrinkage (Virtanen et al., 2012; Klami et al., 2014b):

$$\begin{aligned} \lambda_{j/h}^{(w)} &\sim \mathcal{N}\left(0, \left(\alpha_{j/h}^{(w)}\right)^{-1}\right) & \text{for } j = 1, \dots, p_w, \\ \alpha_{j/h}^{(w)} &\sim \text{Ga}(a_0, b_0), \end{aligned}$$

for observation $w = 1, \dots, m$ and loading column $h = 1, \dots, k$. This prior assumes that each element of observation-specific loading $\lambda_{j/h}^{(w)}$ is jointly regularized. This prior encourages the

parameter $\alpha_{j/h}^{(w)}$ to have large values or values near zero, either pushing elements of $\lambda_{j/h}^{(w)}$ toward zero or imposing minimal shrinkage, and enabling observation-specific, column-wise sparsity.

Other work puts alternative structured regularizers on $\mathbf{A}^{(w)}$ (Jia et al., 2010). To induce observation-specific, column-wise sparsity, GFA used mixed norms: an l_1 norm penalizes each observation-specific column, and either l_2 or l_∞ norms penalize the elements in an observation-specific column:

$$\phi(\mathbf{A}^{(w)}) = \sum_{h=1}^k \|\lambda_{j/h}^{(w)}\|_2 \quad \text{or} \quad \phi(\mathbf{A}^{(w)}) = \sum_{h=1}^k \|\lambda_{j/h}^{(w)}\|_\infty.$$

The l_1 norm penalty achieves observation-specific column-wise shrinkage. Both of these mixed-norm penalties create a bi-convex problem in \mathbf{A} and \mathbf{X} .

These two approaches of adaptive structured regularization in GFA models capture covariance uniquely shared among arbitrary subsets of the observations and avoid modeling shared covariance in non-maximal subsets. But neither the ARD approach nor the mixed-norm penalties encourages element-wise sparsity within loading columns. Adding element-wise sparsity is important because it results in interpretable latent factors, where features with non-zero loadings in a specific factor have an interpretation as a cluster (West, 2003; Carralho et al., 2008). To induce element-wise sparsity, one can either use Bayesian shrinkage on each loading (Carralho et al., 2010) or a mixed norm with l_1 type penalties on each element (i.e., $\sum_{h=1}^k \sum_{j=1}^p |\lambda_{j/h}^{(w)}|$).

A more recent GFA model is a step toward both column-wise and element-wise sparsity (Khan et al., 2014). In this model, element-wise sparsity is achieved by putting independent ARD priors on each loading element, and column-wise sparsity is achieved by a spike-and-slab prior on the loading columns. However, ARD priors do not allow the model to adjust shrinkage levels within each factor, and this approach does not include sparse and dense factors. One contribution of our work is to define a carefully structured Bayesian shrinkage prior on the loading matrix of a GFA model that encourages both element-wise and column-wise shrinkage, and that includes both sparse and dense factors.

3. Bayesian structured sparsity

The column-wise sparse structure of \mathbf{A} in GFA models belongs to a general class of structured sparsity methods that has drawn attention recently (Zou and Hastie, 2005; Yan and Lin, 2006; Jenatton et al., 2011, 2010; Kowalski, 2009; Kowalski and Torrèsani, 2009; Zhao et al., 2009; Huang et al., 2011; Jia et al., 2010). For example, in structured sparse PCA, the loading matrix is constrained to have specific patterns (Jenatton et al., 2010). Later work discussed more general structured variable selection methods in a regression framework (Jenatton et al., 2011; Huang et al., 2011). However, there has been little work in using Bayesian structured sparsity, with some exceptions (Kyung et al., 2010; Engelhardt and Adams, 2014; Wu et al., 2014). Starting from Bayesian sparse priors, we propose a structured hierarchical sparse prior that includes three levels of shrinkage, which is conceptually similar to tree structured shrinkage (Romberg et al., 2001), or global-local priors in the regression framework (Polson and Scott, 2011).

3.1 Bayesian sparsity-inducing priors

Bayesian shrinkage priors have been widely used in latent factor models due to their flexible and interpretable solutions (West, 2003; Carvalho et al., 2008; Polson and Scott, 2011; Knowles and Ghahramani, 2011; Bhattacharya and Dunson, 2011). In Bayesian statistics, a regularizing term, $\phi(\mathbf{A})$, may be viewed as a marginal prior proportional to $\exp(-\phi(\mathbf{A}))$; the regularized optimum then becomes the maximum a posteriori (MAP) solution (Polson and Scott, 2011). For example, the well known ℓ_2 penalty for coefficients in linear regression models corresponds to Gaussian priors, also known as ridge regression or Tikhonov regularization (Hoerl and Kennard, 1970). In contrast, an ℓ_1 penalty corresponds to double exponential or Laplace priors, also known as the Bayesian Lasso (Tibshirani, 1996; Park and Casella, 2008; Hans, 2009).

When the goal of regularization is to induce sparsity, the prior distribution should be chosen so that it has substantial probability mass around zero, which draws small effects toward zero, and heavy tails, which allows large signals to escape from substantial shrinkage (O’Hagan, 1979; Carvalho et al., 2010; Armagan et al., 2011). The canonical Bayesian sparsity-inducing prior is the spike-and-slab prior, which is a mixture of a point mass at zero and a flat distribution across the space of real values, often modeled as a Gaussian with a large variance term (Mitchell and Beauchamp, 1988; West, 2003). The spike-and-slab prior has elegant interpretability by estimating the probability that certain loadings are excluded, modeled by the ‘spike’ distribution, or included, modeled by the ‘slab’ distribution (Carvalho et al., 2008). This interpretability comes at the cost of having exponentially many possible configurations of model inclusion parameters in the loading matrix.

Recently, scale mixtures of normal priors have been proposed as a computationally efficient alternative to the two component spike-and-slab prior (West, 1987; Carvalho et al., 2010; Polson and Scott, 2011; Armagan et al., 2013, 2011; Bhattacharya et al., 2014). Such priors generally assume normal distributions with a mixed variance term. The mixing distribution of the variance allows strong shrinkage near zero but weak regularization away from zero. For example, an inverse gamma distribution on the variance term results in an ARD prior (Tipping, 2001), and an exponential distribution on the variance term results in a Laplace prior (Park and Casella, 2008). The horseshoe prior, with a half Cauchy distribution on the standard deviation as the mixing density, has become popular due to its strong shrinkage and heavy tails (Carvalho et al., 2010).

A more general class of beta mixtures of normals is the three parameter beta distribution (Armagan et al., 2011). Although these continuous shrinkage priors do not directly model the probability of feature inclusion, it has been shown in the regression framework that two layers of regularization—global regularization, across all coefficients, and local regularization, specific to each coefficient (Polson and Scott, 2011)—has behavior that is similar to the spike-and-slab prior in effectively modeling signal and noise separately, but with computational tractability (Carvalho et al., 2009). In this study, we extend and structure the beta mixture of normals prior to three levels of hierarchy to induce desirable behavior in the context of GFA models.

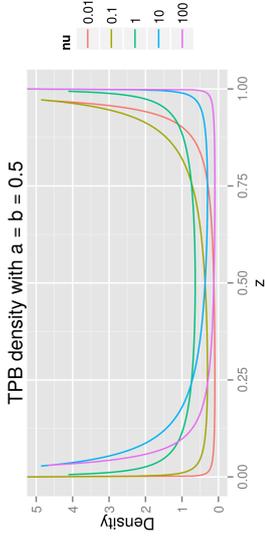


Figure 2: **Density of the three parameter beta (TPB) distribution with different values of ν .** Five different values of $\nu = \{0.01, 0.1, 1, 10, 100\}$ for the three parameter beta distribution with $a = b = 0.5$. The x-axis represents the value of random variable z , and the y-axis represents the density of random variable z .

3.2 Three parameter beta prior

The three parameter beta (TPB) distribution for a random variable $Z \in (0, 1)$ has the following density (Armagan et al., 2011):

$$f(z; a, b, \nu) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \nu^z b^{-1} (1-z)^{a-1} \{1 + (\nu-1)z\}^{-(a+b)}, \quad (7)$$

where $a, b, \phi > 0$. We denote this distribution as $\mathcal{TPB}(a, b, \nu)$. When $0 < a < 1$ and $0 < b < 1$, the distribution is bimodal, with modes at 0 and 1 (Figure 2). The variance parameter ν gives the distribution freedom: with fixed a and b , smaller values of ν put greater probability on $z = 1$, while larger values of ν move the probability mass towards $z = 0$ (Armagan et al., 2011). With $\nu = 1$, this distribution is identical to a beta distribution (i.e., $Be(b, a)$).

Let λ denote the parameter to which we are applying sparsity-inducing regularization. We assign the following TPB normal scale mixture distribution, \mathcal{TPBN} , to λ :

$$\lambda | \varphi \sim \mathcal{N}\left(0, \frac{1}{\varphi} - 1\right), \quad \text{with} \quad \varphi \sim \mathcal{TPB}(a, b, \nu),$$

where the *shrinkage parameter* φ follows a TPB distribution. With $a = b = 1/2$ and $\nu = 1$, this prior becomes the horseshoe prior (Carvalho et al., 2010; Armagan et al., 2011; Gao et al., 2013). The bimodal property of φ induces two distinct shrinkage behaviors: the mode near one encourages $\frac{1}{\varphi} - 1$ towards zero and induces strong shrinkage on λ ; the mode near zero encourages $\frac{1}{\varphi} - 1$ large, creating a diffuse prior on λ . Further decreasing the variance parameter ν supports stronger shrinkage (Armagan et al., 2011; Gao et al., 2013). If we let $\theta = \frac{1}{\varphi} - 1$, then this mixture has the following hierarchical representation:

$$\lambda \sim \mathcal{N}(0, \theta), \quad \theta \sim Ga(a, \delta), \quad \delta \sim Ga(b, \nu).$$

Note the difference between the ARD prior and the \mathcal{TPB} : the ARD prior induces sparsity using an inverse gamma prior on θ , whereas the \mathcal{TPB} induces sparsity by using a gamma prior on the θ variable and then regularizing the rate parameter δ using a second gamma prior. These differences lead to different behavior of ARD and the \mathcal{TPB} in theory (Polson and Scott, 2011) and in practice, as we show below.

3.3 Global-factor-local shrinkage

The flexible representation of the \mathcal{TPB} prior makes it an ideal choice for latent factor models. Our recent work extended the \mathcal{TPB} prior to three levels of regularization on a loading matrix (Gao et al., 2013):

$$\begin{aligned} \varrho &\sim \mathcal{TPB}(c, f, \nu), & (\text{Global}) \\ \zeta_h &\sim \mathcal{TPB}\left(c, d, \frac{1}{\varrho} - 1\right), & (\text{Factor-specific}) \\ \varphi_{jh} &\sim \mathcal{TPB}\left(a, b, \frac{1}{\zeta_h} - 1\right), & (\text{Local}) \\ \lambda_{jh} &\sim \mathcal{N}\left(0, \frac{1}{\varphi_{jh}} - 1\right). \end{aligned} \tag{8}$$

At each of the three levels, a \mathcal{TPB} distribution is used to induce sparsity via its estimated variance parameter (ν in Equation 7), which in turn is regularized using a \mathcal{TPB} distribution. Specifically, the global shrinkage parameter ϱ applies strong shrinkage across the k columns of the loading matrix and jointly adjusts the support of column-specific parameter ζ_h , $h \in \{1, \dots, k\}$ close to either zero or one. This can be interpreted as inducing sufficient shrinkage across loading columns to recover the number of factors supported by the observed data. In particular, when ζ_h is close to one, all elements of column h are close to zero, effectively removing the h^{th} component. When near zero, the factor-specific regularization parameter ζ_h adjusts the shrinkage applied to each element of the h^{th} loading column, estimating the column-wise shrinkage by borrowing strength across all elements (i.e., features) in that column. The local shrinkage parameter, φ_{jh} , creates element-wise sparsity in the loading matrix through a \mathcal{TPBN} . Three levels of shrinkage allow us to model both column-wise and element-wise shrinkage simultaneously, and give the model nonparametric behavior in the number of factors via model selection.

Equivalently, this global-factor-local shrinkage prior can be written as (Ammagan et al., 2011; Gao et al., 2013):

$$\begin{aligned} &\text{Global} && \begin{cases} \gamma \sim \text{Gal}(f, \nu), \\ \eta \sim \text{Gal}(c, \gamma), \end{cases} \\ &\text{Factor-specific} && \begin{cases} \tau_h \sim \text{Gal}(d, \eta), \\ \phi_h \sim \text{Gal}(c, \tau_h), \end{cases} \\ &\text{Local} && \begin{cases} \delta_{jh} \sim \text{Gal}(b, \phi_h), \\ \theta_{jh} \sim \text{Gal}(a, \delta_{jh}), \end{cases} \\ &&& \lambda_{jh} \sim \mathcal{N}(0, \theta_{jh}). \end{aligned} \tag{9}$$

We further extend our prior to jointly model sparse and dense components by assigning to the local shrinkage parameter a two-component mixture distribution (Gao et al., 2013):

$$\theta_{jh} \sim \pi \text{Gal}(a, \delta_{jh}) + (1 - \pi) \delta_{\phi_h}(\cdot), \tag{10}$$

where $\delta_{\phi_h}(\cdot)$ is the Dirac delta function centered at ϕ_h . The motivation for this two component mixture is that, in real applications such as the analysis of gene expression data, it has been shown that much of the variation in the observation is due to technical (e.g., batch, platform) or biological effects (e.g., sex, ethnicity), which impact a large number of features (Leek et al., 2010). Therefore, loadings corresponding to these effects will often not be sparse. A two-component mixture (Equation 10) allows the prior on the loading (Equation 8) to select between element-wise sparsity or column-wise sparsity. Element-wise sparsity is encouraged via the \mathcal{TPBN} prior. Column-wise sparsity jointly regularizes each element of the column with a shared variance term: $\lambda_{jh} \sim \mathcal{N}\left(0, \frac{1}{\zeta_h} - 1\right)$. Modeling each element in a column using a shared regularized variance term has two possible behaviors: i) ζ_h in Equation (8) is close to 1 and the entire column is shrunk towards zero, effectively removing this factor; ii) ζ_h is close to zero, and all elements of the column have a shared Gaussian distribution, inducing only non-zero elements in that loading. We call included factors that have only non-zero elements *dense factors*.

Jointly modeling sparse and dense factors effectively combines low-rank covariance factorization with interpretability (Zou et al., 2006; Parkhomenko et al., 2009). The dense factors capture the broad effects of observation confounders, model a low-rank approximation of the covariance matrix, and usually account for a large proportion of variance explained (Chandrasekaran et al., 2011). The sparse factors, on the other hand, capture the small groups of interacting features in a (possibly) high-dimensional sparse space, and usually account for a small proportion of the variance explained.

We introduce indicator variables z_h , $h = 1, \dots, k$, to indicate which mixture component each θ_{jh} is generated from in Equation (10), where $z_h = 1$ means $\theta_{jh} \sim \text{Gal}(a, \delta_{jh})$ and $z_h = 0$ means $\theta_{jh} \sim \delta_{\phi_h}(\cdot)$. Thus, a component is a sparse factor when $z_h = 1$ and either a dense factor or eliminated when $z_h = 0$. We let $\mathbf{z} = [z_1, \dots, z_k]$ and put a Bernoulli distribution with parameter π on z_h . We further let π have a flat beta distribution $\text{Bet}(1, 1)$. This construct allows us to quantify the posterior probability that each factor h is generated from each mixture component type via z_h .

4. Bayesian group factor analysis with structured sparsity

In this work, we use global-factor-local \mathcal{TPB} priors in the GFA model to enable both element-wise and column-wise shrinkage. Specifically, we put a \mathcal{TPB} prior independently on each loading matrix corresponding to the w^{th} observation, $\mathbf{A}^{(w)}$. Let $\mathbf{Z} = [\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(m)}] \in \mathbb{R}^{m \times k}$. The indicator variable $z_h^{(w)}$ is associated with the h^{th} factor and specific to observation w . When $z_h^{(w)} = 1$, the h^{th} factor has a sparse loading for observation w ; when $z_h^{(w)} = 0$, then either the h^{th} factor has a dense loading column for observation w , or observation w is not represented in that loading column. A zero loading column for observation w effectively decouples the factor from that observation, leading to the column-wise sparse behavior in previous GFA models (Virtanen et al., 2012; Klami et al., 2014b). In our model, factors

that include no observations in the associated loading column are removed from the model. We refer to this model as Bayesian group factor Analysis with Structured Sparsity (BASS).

We summarize BASS as follows. The generative model for m coupled observations $\mathbf{y}_i^{(w)}$ with $w = 1, \dots, m$ and $i = 1, \dots, n$ is

$$\mathbf{y}_i^{(w)} = \mathbf{\Lambda}^{(w)} \mathbf{x}_i + \boldsymbol{\epsilon}_i^{(w)}, \quad \text{for } w = 1, \dots, m.$$

This model is written as a latent factor model by concatenating the m feature vectors into vector \mathbf{y}_i .

$$\begin{aligned} \mathbf{y}_i &= \mathbf{\Lambda} \mathbf{x}_i + \boldsymbol{\epsilon}_i, \\ \mathbf{x}_i &\sim \mathcal{N}_k(0, \mathbf{I}_k), \\ \boldsymbol{\epsilon}_i &\sim \mathcal{N}_n(0, \boldsymbol{\Sigma}), \end{aligned} \quad (11)$$

where $\boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, \dots, \sigma_p^2)$ and $p = \sum_{w=1}^m p_w$. We put independent global-factor-local \mathcal{TPB} priors (Equation 9) on $\mathbf{\Lambda}^{(w)}$:

$$\begin{aligned} \text{Global} & \begin{cases} \gamma^{(w)} \sim Ga(f, \nu), \\ \eta^{(w)} \sim Ga(c, \gamma^{(w)}), \end{cases} \\ \text{Factor-specific} & \begin{cases} \tau_h^{(w)} \sim Ga(d, \eta^{(w)}), \\ \phi_h^{(w)} \sim Ga(c, \tau_h^{(w)}), \end{cases} \\ \text{Local} & \begin{cases} \delta_{jh}^{(w)} \sim Ga(b, \phi_h^{(w)}), \\ \theta_{jh}^{(w)} \sim Ga(a, \delta_{jh}^{(w)}), \end{cases} \\ & \lambda_{jh}^{(w)} \sim \mathcal{N}(0, \theta_{jh}^{(w)}). \end{aligned}$$

We allow local shrinkage to follow a two-component mixture

$$\theta_{jh}^{(w)} \sim \pi^{(w)} Ga(a, \delta_{jh}^{(w)}) + (1 - \pi^{(w)}) \delta_{\phi_h^{(w)}}^{(w)}(\cdot),$$

where the mixture proportion has a beta distribution

$$\pi^{(w)} \sim B\epsilon(1, 1).$$

We put a conjugate inverse gamma distribution on the residual variance parameters

$$\sigma_j^{-2} \sim Ga(a_\sigma, b_\sigma).$$

In our application of BASS, we set the hyperparameters of the global-factor-local \mathcal{TPB} prior to $a = b = c = d = e = f = 0.5$, which recapitulates the horseshoe prior at all three levels of the hierarchy. The hyperparameters for the error variances, a_σ and b_σ , were set to 1 and 0.3 respectively to allow a relatively wide support of variances (Bhattacharya and Dunson, 2011). When there are two coupled observations, the BASS framework is a Bayesian CCA model (Equation 4) based on its column-wise shrinkage.

5. Parameter estimation

Given our setup, the full joint distribution of the BASS model factorizes as

$$\begin{aligned} p(\mathbf{Y}, \mathbf{X}, \mathbf{\Lambda}, \boldsymbol{\Theta}, \boldsymbol{\Delta}, \boldsymbol{\Phi}, \mathbf{T}, \boldsymbol{\eta}, \boldsymbol{\gamma}, \mathbf{Z}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) \\ = p(\mathbf{Y} | \mathbf{\Lambda}, \mathbf{X}, \boldsymbol{\Sigma}) p(\mathbf{X}) \\ \times p(\mathbf{\Lambda} | \boldsymbol{\Theta}) p(\boldsymbol{\Theta} | \boldsymbol{\Delta}, \mathbf{Z}, \boldsymbol{\Phi}) p(\boldsymbol{\Delta} | \boldsymbol{\Phi}) p(\boldsymbol{\Phi} | \mathbf{T}) p(\mathbf{T} | \boldsymbol{\eta}) p(\boldsymbol{\eta} | \boldsymbol{\gamma}) \\ \times p(\boldsymbol{\Sigma}) p(\mathbf{Z} | \boldsymbol{\pi}) p(\boldsymbol{\pi}), \end{aligned}$$

where $\boldsymbol{\Theta} = \{\theta_{jh}^{(w)}\}$, $\boldsymbol{\Delta} = \{\delta_{jh}^{(w)}\}$, $\boldsymbol{\Phi} = \{\phi_h^{(w)}\}$, $\mathbf{T} = \{\tau_h^{(w)}\}$, $\boldsymbol{\eta} = \{\eta^{(w)}\}$ and $\boldsymbol{\gamma} = \{\gamma^{(w)}\}$ are the collections of the global-factor-local \mathcal{TPB} prior parameters. The posterior distributions of model parameters may be either simulated through Markov chain Monte Carlo (MCMC) methods or approximated using variational Bayes approaches. We derive an MCMC algorithm based on a Gibbs sampler (Appendix A). The MCMC algorithm updates the joint loading matrix row by row using block updates, enabling relatively fast mixing (Bhattacharya and Dunson, 2011).

In many applications, we are interested in a single point estimate of the parameters instead of the complete posterior estimate; thus, often an expectation maximization (EM) algorithm is used to find a *maximum a posteriori* (MAP) estimate of model parameters using conjugate gradient optimization (Dempster et al., 1977). In EM, the latent factors \mathbf{X} and the indicator variables \mathbf{Z} are treated as missing data and their expectations estimated in the E-step conditioned on the current values of the parameters; then the model parameters are optimized in the M-step conditioning on the current expectations of the latent variables. Let $\boldsymbol{\Xi} = \{\mathbf{\Lambda}, \boldsymbol{\Theta}, \boldsymbol{\Delta}, \boldsymbol{\Phi}, \mathbf{T}, \boldsymbol{\eta}, \boldsymbol{\pi}, \boldsymbol{\Sigma}\}$ be the collection of the parameters optimized in the M-step. The expected complete log likelihood, denoted $Q(\cdot)$, may be written as

$$Q(\boldsymbol{\Xi} | \boldsymbol{\Xi}_{(-s)}) = \mathbb{E}_{\mathbf{X}, \mathbf{Z} | \boldsymbol{\Xi}_{(-s)}, \mathbf{Y}} [\log p(\boldsymbol{\Xi}, \mathbf{X}, \mathbf{Z} | \mathbf{Y})].$$

Since \mathbf{X} and \mathbf{Z} are conditionally independent given $\boldsymbol{\Xi}$, the expectation may be calculated using the full conditional distributions of \mathbf{X} and \mathbf{Z} derived for the MCMC algorithm. The derivation of the EM algorithm for BASS is then straightforward (Appendix B); note that, when estimating $\mathbf{\Lambda}$, the loading columns specific to each observation are estimated jointly.

5.1 Identifiability

The latent factor model (Equation 1) is identifiable up to orthonormal rotations: for any orthogonal matrix \mathbf{P} with $\mathbf{P}^T \mathbf{P} = \mathbf{I}$, letting $\mathbf{\Lambda}' = \mathbf{\Lambda} \mathbf{P}^T$ and $\mathbf{x}' = \mathbf{P} \mathbf{x}$ produces the same estimate of the data covariance matrix and has an identical likelihood. When using factor analysis for prediction or covariance estimation, rotational invariance is irrelevant. However, for all applications that interpret the factors or use individual factors or loadings for downstream analysis, this rotational invariance cannot be ignored. One traditional solution is to restrict the loading matrix to be lower triangular (West, 2003; Carvalho et al., 2008). This solution gives a special role to the first $k - 1$ features in \mathbf{y} , namely, that the h^{th} feature does not contribute to the $k - h^{\text{th}}$ through the k^{th} factor. For this reason, the lower triangular approach does not generalize easily and requires domain knowledge that may not be available (Carvalho et al., 2008).

In the BASS model, we have rotational invariance when we right multiply the joint loading matrix by \mathbf{P}^T and left multiply \mathbf{x} by \mathbf{P} , producing an identical covariance matrix and likelihood. This rotational invariance is addressed in BASS because the non-sparse rotations of the loading matrix violates the prior structure induced by the observation-wise and element-wise sparsity.

Scale invariance is a second identifiability problem inherent in latent factor models. In particular, scale invariance means that a loading can be multiplied by a non-zero constant and the corresponding factor by the inverse of that constant, and this will result in the same data likelihood. This problem we and others have addressed satisfactorily by using posterior probabilities as optimization objectives instead of likelihoods and by including regularizing priors on the factors that restrict the magnitude of the constant. We make an effort to not interpret the relative or absolute scale of the factors or loadings including sign beyond setting a reasonable threshold for zero.

Finally, factor analysis is identifiable up to *label switching*, or shuffling the $h = 1, \dots, k$ indices of the loadings and factors, assuming we do not take the lower triangular approach. Other approaches put distributions on the loading sparsity or proportion of variance explained in order to address this problem (Bhattacharya and Dunson, 2011). We do not explicitly order or interpret the order of the factors, so we do not address this non-identifiability in the model. Label switching is handled here and elsewhere by a post-processing step, such as ordering factors according to proportion of variance explained. In our simulation studies, we interpret results with this non-identifiability in mind.

5.2 Sparse rotations via PX-EM

Another general problem with latent factor models, including BASS, is the convergence to local optima and sensitivity to parameter initializations. Once the model parameters are initialized, the EM algorithm may be stuck in locally optimal but globally suboptimal regions with undesirable factor orientations. To address this problem, we take advantage of the rotational invariance of the factor analysis framework. Parameter expansion (PX) has been shown to reduce the initialization dependence by introducing auxiliary variables that rotate the current estimate of the loading matrix to best respect the prior while keeping the likelihood stable (Liu et al., 1998; Dyk and Meng, 2001).

We extend our model (Equation 11) using parameter expansion \mathbf{R} , a positive definite $k \times k$ matrix, as

$$\begin{aligned} \mathbf{y}_i &= \mathbf{A}\mathbf{R}_L^{-1}\mathbf{x}_i + \epsilon_i, \\ \mathbf{x}_i &\sim N_k(\mathbf{0}, \mathbf{R}), \\ \epsilon_i &\sim N_k(\mathbf{0}, \mathbf{\Sigma}), \end{aligned}$$

where \mathbf{R}_L is the lower triangular matrix of the Cholesky decomposition of \mathbf{R} . The covariance of \mathbf{y}_i is invariant under this expansion, and, correspondingly, the likelihood is stable. Note \mathbf{R}_L^{-1} is not an orthogonal matrix; however, because it is full rank, it can be transformed into an orthogonal matrix times a rotation matrix via a polar decomposition (Rocková and George, 2015). We let $\mathbf{A}^* = \mathbf{A}\mathbf{R}_L^{-1}$ and assign our BASS \mathcal{TPPEN} prior to this *rotated* loading matrix.

We let $\mathbf{\Xi}^* = \{\mathbf{A}^*, \Theta, \Delta, \Phi, \mathbf{T}, \eta, \gamma, \pi, \Sigma\}$, and the parameters of our expanded model are $\{\mathbf{\Xi}^* \cup \mathbf{R}\}$. The EM algorithm in this expanded parameter space generates a sequence of parameter estimates $\{\mathbf{\Xi}^{*(1)} \cup \mathbf{R}^{(1)}, \mathbf{\Xi}^{*(2)} \cup \mathbf{R}^{(2)}, \dots\}$, which corresponds to a sequence of parameter estimates in the original space $\{\mathbf{\Xi}^{(1)}, \mathbf{\Xi}^{(2)}, \dots\}$, where \mathbf{A} is recovered via $\mathbf{A}^*\mathbf{R}_L$ (Rocková and George, 2015). We initialize $\mathbf{R}_{(0)} = \mathbf{I}_k$. The expected complete log likelihood of this PX BASS model is

$$Q(\mathbf{\Xi}^*, \mathbf{R} | \mathbf{\Xi}^{(s)}) = \mathbb{E}_{\mathbf{X}, \mathbf{Z} | \mathbf{\Xi}^{(s)}, \mathbf{Y}, \mathbf{R}_0} \log(p(\mathbf{\Xi}^*, \mathbf{R}, \mathbf{X}, \mathbf{Z} | \mathbf{Y})). \quad (12)$$

In our parameter-expanded EM (PX-EM) for BASS, the conditional distributions of \mathbf{X} and \mathbf{Z} still factorize in the expectation. However, the distribution of x_i depends on expansion parameter \mathbf{R} . The full joint distribution (Equation 11) has a single change in $p(\mathbf{X})$, with \mathbf{A}^* in the place of \mathbf{A} . In the M-step, the \mathbf{R} that maximizes Equation (12) is

$$\mathbf{R}_{(s)} = \arg \max_{\mathbf{R}} Q(\mathbf{\Xi}^*, \mathbf{R} | \mathbf{\Xi}^{(s)}) = \arg \max_{\mathbf{R}} \left(\text{const} - \frac{n}{2} \log |\mathbf{R}| - \frac{1}{2} \text{tr}(\mathbf{R}^{-1} \mathbf{S}^{\mathbf{X}\mathbf{X}}) \right),$$

where $\mathbf{S}^{\mathbf{X}\mathbf{X}} = \sum_{i=1}^n \langle \mathbf{x}_i, \mathbf{x}_i^T \rangle$. The solution is $\mathbf{R}_{(s)} = \frac{1}{n} \mathbf{S}^{\mathbf{X}\mathbf{X}}$. For the E-step, \mathbf{A} is first calculated and the expectation is taken in the original space (details in Appendix C).

Note that the proposed PX-EM for the BASS model keeps the likelihood invariant but does not keep the prior invariant after transformation of \mathbf{A} . This is different from the earlier PX-EM algorithm (Lin et al., 1998), as discussed in recent work (Rocková and George, 2015). Because the resulting posterior is not invariant, we run PX-EM only for a few iterations and then switch to the EM algorithm. The effect is that the BASS model is substantially less sensitive to initialization (see simulation results). By introducing expansion parameter \mathbf{R} , the posterior modes in the original space are intersected with equal likelihood curves indexed by \mathbf{R} in expanded space. Those curves facilitate traversal between posterior modes in the original space and encourage initial parameter estimates with appropriate sparse structure in the loading matrix (Rocková and George, 2015).

5.3 Computational complexity

The computational complexity of the block Gibbs sampler for the BASS model is demanding. Updating each loading row requires the inversion of a $k \times k$ matrix with $O(k^3)$ complexity and then calculating means with $O(k^2n)$ complexity. The complexity of updating the full loading matrix repeats this calculation p times. Other updates are of lower order relative to updating the loading. Our Gibbs sampler has $O(k^3p + k^2pn)$ complexity per iteration, which makes MCMC difficult to apply when p is large.

In the BASS EM algorithm, the E-step has complexity $O(k^3)$ for a matrix inversion, complexity $O(k^2p + kpn)$ for calculating the first moment, and complexity $O(k^2n)$ for calculating the second moment. Calculations in the M-step are all of a lower order. Thus, the EM algorithm has complexity $O(k^3 + k^2p + k^2n + kpn)$ per iteration.

Our PX-EM algorithm for the BASS model requires an additional Cholesky decomposition with complexity $O(k^3)$ and a matrix multiplication with complexity $O(k^2p)$ above the EM algorithm. The total complexity is therefore the same as the original EM algorithm, although in practice we note that the constants have a negative impact on the running time.

6. Simulations and comparisons

We demonstrate the performance of our model on simulated data in three settings: paired observations, four observations, and ten observations.

6.1 Simulations

We describe the details of the three types of simulations here.

6.1.1 SIMULATIONS WITH PAIRED OBSERVATIONS (CCA)

We simulated two data sets with $p_1 = 100$, $p_2 = 120$ in order to compare results from our method to results from state-of-the-art CCA methods. The number of samples in these simulations was $n = \{20, 30, 40, 50\}$, chosen to be smaller than both p_1 and p_2 to reflect the large p , small n regime (West, 2003) that motivated our structured approach. We first simulated observations with only sparse latent factors (*Sim1*). In particular, we set $k = 6$, where two sparse factors are shared by both observations (factors 1 and 2; Table 1), two sparse factors are specific to $\mathbf{y}^{(1)}$ (factors 3 and 4; Table 1), and two sparse factors are specific to $\mathbf{y}^{(2)}$ (factors 5 and 6; Table 1). The elements in the sparse loading matrix were randomly generated from a $\mathcal{N}(0, 4)$ Gaussian distribution, and sparsity was induced by setting 90% of the elements in each loading column to zero at random (Figure 3A). We zeroed values of the sparse loadings for which the absolute values were less than 0.5. Latent factors \mathbf{x} were generated from $\mathcal{N}_6(0, \mathbf{I}_6)$. Residual error was generated by first generating the $p = p_1 + p_2$ diagonals on the residual covariance matrix Σ from a uniform distribution on $(0.5, 1.5)$, and then generating each column of the error matrix from $\mathcal{N}_p(\mathbf{0}, \Sigma)$.

We performed a second simulation that included both sparse and dense latent factors (*Sim2*). In particular, we extended *Sim1* to $k = 8$ latent factors, where one of the shared sparse factors is now dense, and two dense factors, each specific to one observation, were added. For all dense factors, each loading was generated according to a $\mathcal{N}(0, 4)$ Gaussian distribution (Table 1; Figure 3B).

Factors	<i>Sim1</i>						<i>Sim2</i>							
	1	2	3	4	5	6	1	2	3	4	5	6	7	8
$\mathbf{Y}^{(1)}$	S	S	S	S	-	-	S	D	S	S	D	-	-	-
$\mathbf{Y}^{(2)}$	S	S	-	-	S	S	S	D	-	-	S	S	D	-

Table 1: **Latent factors in *Sim1* and *Sim2* with two observation matrices.** S represents a sparse vector; D represents a dense vector; - represents no contribution to that observation from the factor.

6.1.2 SIMULATIONS WITH FOUR OBSERVATIONS (GFA)

We performed two simulations (*Sim3* and *Sim4*) including four observations with $p_1 = 70$, $p_2 = 60$, $p_3 = 50$ and $p_4 = 40$. The number of samples, as above, was set to $n = \{20, 30, 40, 50\}$. In *Sim3*, we let $k = 6$ and only simulated sparse factors: the first three factors were specific to $\mathbf{y}^{(1)}$, $\mathbf{y}^{(2)}$ and $\mathbf{y}^{(3)}$, respectively, and the last three corresponded to different subsets of the observations (Table 2). In *Sim4* we let $k = 8$, and, as with *Sim2*,

Factors	<i>Sim3</i>						<i>Sim4</i>							
	1	2	3	4	5	6	1	2	3	4	5	6	7	8
$\mathbf{Y}^{(1)}$	S	-	S	-	-	-	S	-	-	-	D	-	-	-
$\mathbf{Y}^{(2)}$	-	S	S	S	-	-	-	S	-	S	-	D	-	-
$\mathbf{Y}^{(3)}$	-	-	S	S	S	-	-	-	S	-	S	-	-	D
$\mathbf{Y}^{(4)}$	-	-	-	-	-	S	-	-	-	S	-	-	-	D

Table 2: **Latent factors in *Sim3* and *Sim4* with four observation matrices.** S represents a sparse vector; D represents a dense vector; - represents no contribution to that observation from the factor.

Factors	<i>Sim5</i>						<i>Sim6</i>									
	1	2	3	4	5	6	1	2	3	4	5	6	7	8	9	10
$\mathbf{Y}^{(1)}$	S	-	-	-	-	-	S	-	-	-	-	-	D	-	-	-
$\mathbf{Y}^{(2)}$	S	-	S	-	-	-	-	S	-	-	S	-	-	D	-	-
$\mathbf{Y}^{(3)}$	S	-	S	S	-	-	-	-	-	-	S	-	-	D	-	-
$\mathbf{Y}^{(4)}$	S	S	S	S	-	-	-	-	-	S	-	-	-	D	-	-
$\mathbf{Y}^{(5)}$	-	S	-	S	-	-	-	-	-	S	-	S	-	D	-	-
$\mathbf{Y}^{(6)}$	-	S	-	-	S	-	-	-	-	S	-	S	-	-	D	-
$\mathbf{Y}^{(7)}$	-	-	S	-	-	S	-	-	-	S	-	S	-	-	D	-
$\mathbf{Y}^{(8)}$	-	-	-	S	-	-	-	-	-	S	-	S	-	-	-	D
$\mathbf{Y}^{(9)}$	-	-	S	-	-	-	-	-	-	S	-	-	S	-	-	-
$\mathbf{Y}^{(10)}$	-	-	-	-	-	S	-	-	-	-	S	-	-	-	S	-

Table 3: **Latent factors in *Sim5* and *Sim6* with four observation matrices.** S represents a sparse vector; D represents a dense vector; - represents no contribution to that observation from the factor.

included both sparse and dense factors (Table 2). Samples from these two simulations were generated following the same procedure as the simulations with two observations.

6.1.3 SIMULATIONS WITH TEN OBSERVATIONS (GFA)

To further evaluate BASS on multiple observations, we performed two additional simulations (*Sim5* and *Sim6*) on ten coupled observations with $p_w = 50$ for $w = 1, \dots, 10$. The number of samples was set to $n = \{20, 30, 40, 50\}$. In *Sim5*, we let $k = 8$ and only simulated sparse factors (Table 3). In *Sim6* we let $k = 10$ and simulated both sparse and dense factors (Table 3). Samples in these two simulations were generated following the same method as in the simulations with two observations.

6.2 Methods for comparison

We compared BASS to five available linear models that accept multiple observations: the Bayesian group factor analysis model with an ARD prior (GFA) (Klami et al., 2013), an extension of GFA that allows element-wise sparsity with independent ARD priors (sGFA) (Khan et al., 2014; Suvritaval et al., 2014), a regularized version of CCA (RCCA) (González et al., 2008), sparse CCA (SCCA) (Witten and Tibshirani, 2009), and Bayesian joint factor

analysis (JFA) (Ray et al., 2014). We also included the linear version of a flexible non-linear model, manifold relevance determination (MRD) (Dannanoun et al., 2012). To evaluate the sensitivity of BASS to initialization, we compared three different initialization methods: random initialization (EM), 50 iterations of MCMC (MCMC-EM), and 20 iterations of PX-EM (PX-EM); each of these were followed with EM until convergence, reached when both the number of non-zero loadings do not change for t iterations and the log likelihood changes $< 1 \times 10^{-5}$ within t iterations. We performed 20 runs for each version of inference in BASS: EM, MCMC-EM, and PX-EM. In *Sim1* and *Sim2*, we set the initial number of factors to $k = 10$. In *Sim2*, *Sim4*, *Sim5*, and *Sim6*, we set the initial number of factors to 15.

The GFA model (Klami et al., 2013) uses an ARD prior to encourage column-wise shrinkage of the loading matrix, but not sparsity within the loadings. The computational complexity of this GFA model with variational updates is $O(k^3m + k^2p + k^2n + kpn)$ per iteration, which is nearly identical to BASS but includes an additional factor m , the number of observations, scaling the k^3 term. In our simulations, we ran the GFA model with the factor number set to the correct value.

The sGFA model (Khan et al., 2014) encourages element-wise sparsity using independent ARD priors on loading elements. Loading columns are modeled with a spike-and-slab type mixture to encourage column-wise sparsity. Inference is performed with a Gibbs sampler without using block updates. Its complexity is $O(k^3 + k^2pn)$ per iteration, which, when k is large, will dominate the per-iteration complexity of BASS; furthermore, Gibbs samplers typically require greater numbers of iterations than EM-based methods. We ran the sGFA model with the correct number of factors in our six simulations.

We ran the regularized version of classical CCA (RCCA) for comparison in *Sim1* and *Sim2* (González et al., 2008). Classical CCA tries to find k canonical projection directions \mathbf{u}_h and \mathbf{v}_h ($h = 1, \dots, k$) for $\mathbf{Y}^{(1)}$ and $\mathbf{Y}^{(2)}$ respectively such that i) the correlation between $\mathbf{u}_h^T \mathbf{Y}^{(1)}$ and $\mathbf{v}_h^T \mathbf{Y}^{(2)}$ is maximized for $h = 1, \dots, k$; and ii) $\mathbf{u}_h^T \mathbf{Y}^{(1)}$ is orthogonal to $\mathbf{u}_{h'}^T \mathbf{Y}^{(1)}$ with $h' \neq h$, and similarly for \mathbf{v}_h and $\mathbf{Y}^{(2)}$. Let these two projection matrices be denoted $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_k] \in \mathbb{R}^{p_1 \times k}$ and $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_k] \in \mathbb{R}^{p_2 \times k}$. These matrices are the maximum likelihood estimates of the shared loading matrices in the Bayesian CCA model up to orthogonal transformations (Bach and Jordan, 2005). However, classical CCA requires the observation covariance matrices to be non-singular and thus is not applicable in the current simulations where $n < p_1, p_2$.

Here, we used a regularized version of CCA (RCCA) (González et al., 2008), which regularizes CCA using an ℓ_2 -type penalty by adding $\lambda_1 \mathbf{I}_{p_1}$ and $\lambda_2 \mathbf{I}_{p_2}$ to the two sample covariance matrices. The effect of this penalty is not to induce sparsity but instead to allow application to $p \gg n$ data sets. The two regularization parameters (λ_1 and λ_2) were chosen according to leave-one-out cross-validation with the search space defined on a 11×11 grid from 0.0001 to 0.01. The projection directions \mathbf{U} and \mathbf{V} were estimated using the best regularization parameters. We let $\mathbf{A}' = [\mathbf{U}; \mathbf{V}]$; this matrix was comparable to the simulated loading matrix up to orthogonal transformations. We calculated the matrix \mathbf{P} such that the Frobenius norm between $\mathbf{A}' \mathbf{P}^T$ and simulated \mathbf{A} was minimized, with the constraint that $\mathbf{P}^T \mathbf{P} = \mathbf{I}$. This was done by the constraint-preserving updates of the objective function (Wen and Yin, 2013). After finding the optimal orthogonal transformation matrix, we recovered $\mathbf{A}' \mathbf{P}^T$ as the estimated loading matrix. We set the number of projections to

6 and 8 in *Sim1* and *Sim2*, respectively, representing the true number of latent factors. RCCA does not apply to multiple coupled observations, and therefore it was not included in further simulations.

The sparse CCA (SCCA) method (Witten and Tibshirani, 2009) maximizes correlation between two observations after projecting the original space with a sparsity-inducing penalty onto the latent components, producing sparse matrices \mathbf{U} and \mathbf{V} . This method is encoded in the R package PMA (Witten et al., 2013). For *Sim1* and *Sim2*, as with RCCA, we found an optimal orthogonal transformation matrix \mathbf{P} such that the Frobenius norm between $\mathbf{A}' \mathbf{P}^T$ and simulated \mathbf{A} was minimized, where \mathbf{A}' was the vertical concatenation of the recovered sparse \mathbf{U} and \mathbf{V} . We chose 6 and 8 sparse projections in *Sim1* and *Sim2*, respectively, representing the true number of linear factors. Because both RCCA and SCCA are both deterministic and greedy, the results for $k < 6$ are all implicitly available by subsetting the factors in the $k = 6$ results.

An extension of SCCA allows for multiple observations (Witten and Tibshirani, 2009). For *Sim3* and *Sim4*, we recovered four sparse projection matrices $\mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)}, \mathbf{U}^{(4)}$, and for *Sim5* and *Sim6*, we recovered ten projection matrices. \mathbf{A}_s was calculated with the concatenation of those projection matrices. Then the orthogonal transformation matrix \mathbf{P} was calculated similarly by minimizing the Frobenius norm between $\mathbf{A}_s \mathbf{P}^T$ and the true loading matrix \mathbf{A} . The number of canonical projections was set to 6 in *Sim3*, 8 in *Sim4* and *Sim5*, and 10 in *Sim6*, corresponding to the true number of latent factors.

The Bayesian joint factor analysis model (JFA) (Ray et al., 2014) puts an Indian buffet process (IBP) prior (Griffiths and Ghahramani, 2011) on the factors, inducing element-wise sparsity, and an ARD prior on the variance of the loadings. The idea of putting an IBP on a latent factor model, which gives desirable nonparametric behavior in the number of latent factors and also produces element-wise sparsity in the loading matrix, was described for the Nonparametric Sparse Factor Analysis (NSEFA) model (Knowles and Ghahramani, 2011). Similarly, in JFA, element-wise sparsity is encouraged both in the factors and in the loadings. JFA partitions latent factors into a fixed number of observation-specific factors and factors shared by all observations, and does not include column-wise sparsity. Its complexity is $O(k^3 + k^2pn)$ per iteration of the Gibbs sampler. We ran JFA on our simulations with the number of factors set to the correct values. Because the JFA model uses a sparsity-inducing prior instead of an independent Gaussian prior on the latent factors, the resulting model does not have a closed form posterior predictive distribution (Equation 13); therefore, we excluded the JFA model from prediction results.

The non-linear manifold relevance determination (MRD) model (Dannanoun et al., 2012) extends the notable Gaussian process latent variable (GPLVM) model (Lawrence, 2005) to include multiple observations. A GPLVM puts a Gaussian process prior on the latent variable space. GPLVM has an interpretation of a dual probabilistic PCA model that marginalizes loading columns using Gaussian priors. MRD extends GPLVM by putting multiple weight vectors on the latent variables using a Gaussian process kernel. Each of the weight vectors corresponds to one observation, therefore they determine a soft partition of latent variable space. The complexity of MRD is quadratic in the number of samples n per iteration using a sparse Gaussian process. Posterior inference and prediction using the MRD model was performed with Matlab package `varGPLM` (Dannanoun et al., 2012). We used the linear kernel with feature selection (i.e., `Linard2` kernel), meaning that we

used the linear version of this model for a fair comparison. We ran the MRD model on our simulated data with the correct number of factors.

We summarize the parameter choices for all methods here:

sGFA: We used the `getDefaultOpts` function in the `sGFA` package to set the default parameters. In particular, the ARD prior was set to $G\alpha(10^{-3}, 10^{-3})$. The prior on the inclusion probabilities was set to $\text{beta}(1, 1)$. *Total MCMC iterations* were set to 10^5 with *sampling iterations* set to 1,000 and *thinning steps* set to 5.

GFA: We used the `getDefaultOpts()` function in the `GFA` package to set the default parameters. In particular, the ARD prior for both loading and error variance was set to $G\alpha(10^{-14}, 10^{-14})$. The *maximum iteration* parameter was set to 10^5 , and the ‘‘L-BFGS’’ optimization method was used.

RCCA: The regularization parameter was chosen using leave-one-out cross-validation on an 11×11 grid from 0.0001 to 0.01 using the function `est.im.regul` in the `CCA` package.

SCCA: We used the `PMA` package with Lasso penalty (the `typex` and `typez` parameters in the function `CCA` were set to ‘‘standard’’). This corresponds to setting the ℓ_1 bound of the projection vector to $0.3\sqrt{p_w}$ for $w = 1, 2$.

JFA: The ARD priors for both the loading and factor scores were set to $G\alpha(10^{-5}, 10^{-5})$. The parameters of the beta process prior were set to $\alpha = 0.1$ and $c = 10^4$. The MCMC iterations were set to 1,000 with 200 iterations of burn-in. As is the default settings, we did not thin the chain.

MRD: We used the `svargplvm_init` function in the `GPLVM` package to initialize parameters. The `linear2` kernel was chosen for all observations. Latent variables were initialized by concatenating the observation matrices first (the ‘‘concatenated’’ option) and then performing PCA. Other parameters were set by `svargplvm_init` with default options.

6.3 Metrics for comparison

To compare the results of BASS with the alternative methods, we used the sparse and dense stability indices (Gao et al., 2013) to quantify the distance between the simulated loadings and the recovered loadings. The sparse stability index (SSI) measures the similarity between columns of sparse matrices. SSI is invariant to column scale and label switching, but it penalizes factor splitting and matrix rotation; larger values of SSI indicate better recovery. Let $\mathbf{C} \in \mathbb{R}^{k_1 \times k_2}$ be the absolute correlation matrix of columns of two sparse loading matrices. Then SSI is calculated by

$$SSI = \frac{1}{2k_1} \sum_{h_1=1}^{k_1} \left(\max(\mathbf{c}_{h_1, \cdot}) - \frac{\sum_{h_2=1}^{k_2} I(\mathbf{c}_{h_1, h_2} > \bar{\mathbf{c}}_{h_1, \cdot}) \mathbf{c}_{h_1, h_2}}{k_2 - 1} \right) + \frac{1}{2k_2} \sum_{h_2=1}^{k_2} \left(\max(\mathbf{c}_{\cdot, h_2}) - \frac{\sum_{h_1=1}^{k_1} I(\mathbf{c}_{h_1, h_2} > \bar{\mathbf{c}}_{\cdot, h_2}) \mathbf{c}_{h_1, h_2}}{k_1 - 1} \right).$$

The dense stability index (DSI) quantifies the difference between dense matrix columns, and is invariant to orthogonal matrix rotation, factor switching, and scale; DSI values closer to zero indicate better recovery. Let \mathbf{M}_1 and \mathbf{M}_2 be the dense matrices. DSI is calculated by

$$DSI = \frac{1}{p^2} \text{tr}(\mathbf{M}_1 \mathbf{M}_1^T - \mathbf{M}_2 \mathbf{M}_2^T).$$

We extended the stability indices to allow multiple coupled observations as in our simulations. In *Sim1*, *Sim3*, and *Sim5*, all factors are sparse, and SSIs were calculated between the true sparse loading matrices and recovered sparse loading matrices. In *Sim2*, *Sim4*, and *Sim6*, because none of the methods other than BASS explicitly distinguished sparse and dense factors, we categorized each recovered factor as follows. We first selected a global sparsity threshold on the elements of the combined loading matrix; here we set that value to 0.15. Elements below this threshold were set to zero in the loading matrix. Then we chose the first five loading columns with the fewest non-zero elements as the sparse loadings in *Sim2*, first four such loadings as the sparse loadings in *Sim4*, and first six such loadings as sparse in *Sim6*. The remaining loading columns were considered dense loadings and were not zeroed according to the global sparsity threshold. We found that varying the sparsity threshold did not affect the separation of sparse and dense loadings significantly across methods. SSIs were then calculated for the true sparse loading matrix and the recovered sparse loadings across methods.

To calculate DSIs, we treated the loading matrices $\mathbf{\Lambda}^{(w)}$ for each observation separately, and calculated the DSI for the recovered dense components of each observation. The DSI for each method was the sum of the m separate DSIs. Because the loading matrix is marginalized out in MRD (Lawrence, 2005), we excluded MRD from this comparison.

We further evaluated the prediction performance of BASS and other methods. In the BASS model (Equation 6), the joint distribution of any one observation $\mathbf{y}_i^{(w)}$ and all other observations $\mathbf{y}_i^{(-w)}$ can be written as

$$\begin{pmatrix} \mathbf{y}_i^{(w)} \\ \mathbf{y}_i^{(-w)} \end{pmatrix} \sim \mathcal{N} \left[\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \mathbf{\Lambda}^{(w)}(\mathbf{\Lambda}^{(w)})^T + \mathbf{\Sigma}^{(w)} & \mathbf{\Lambda}^{(w)}(\mathbf{\Lambda}^{(-w)})^T \\ \mathbf{\Lambda}^{(-w)}(\mathbf{\Lambda}^{(w)})^T & \mathbf{\Lambda}^{(-w)}(\mathbf{\Lambda}^{(-w)})^T + \mathbf{\Sigma}^{(-w)} \end{pmatrix} \right],$$

where $\mathbf{\Lambda}^{(-w)}$ and $\mathbf{\Sigma}^{(-w)}$ are the loading matrix and residual covariance excluding the w^{th} observation. Therefore, the conditional distribution of $\mathbf{y}_i^{(w)}$ is a multivariate response in a multivariate linear regression model, where $\mathbf{y}_i^{(-w)}$ are the predictors; the mean term takes the form

$$\mathbb{E}(\mathbf{y}_i^{(w)} | \mathbf{y}_i^{(-w)}) = \mathbf{\Lambda}^{(w)}(\mathbf{\Lambda}^{(-w)})^T (\mathbf{\Lambda}^{(-w)}(\mathbf{\Lambda}^{(-w)})^T + \mathbf{\Sigma}^{(-w)})^{-1} \mathbf{y}_i^{(-w)} \\ = \sum_{h=1}^k \boldsymbol{\lambda}_h^{(w)} (\boldsymbol{\lambda}_h^{(-w)})^T (\mathbf{\Lambda}^{(-w)}(\mathbf{\Lambda}^{(-w)})^T + \mathbf{\Sigma}^{(-w)})^{-1} \mathbf{y}_i^{(-w)}. \quad (13)$$

We used this conditional distribution to predict specific observations given others. For the six simulations, we used the simulated data as training data for training sample sizes $n_t = \{30, 50\}$, and, additionally, simulated data sets with training sample sizes $n_t =$

	EM	MCMC-EM	PX-EM
<i>Sim1</i>	79.17%	99.17%	91.67%
<i>Sim2</i>	61.25%	93.75%	85.62%
<i>Sim3</i>	50.00%	78.57%	73.57%
<i>Sim4</i>	62.78%	86.11%	82.78%
<i>Sim5</i>	17.22%	86.67%	66.67%
<i>Sim6</i>	13.64%	60.45%	62.73%

Table 4: **Percentage of latent factors correctly identified across 20 runs with $n = 40$.** The columns represent the runs of EM, EM initialized with MCMC (MCMC-EM), and EM initialized with PX-EM.

{10, 100, 200}. Then, we generated $n_s = 200$ samples as test data using the true model parameters, simulating the corresponding test data factors $\mathbf{X} \sim \mathcal{N}(0, \mathbf{I})$. For each simulation study, we chose at least one observation in the test data as the response and used the other observations and model parameters estimated from the training data to perform prediction. Mean squared error (MSE) was used to evaluate the prediction performance. For *Sim1* and *Sim2*, $\mathbf{y}_i^{(2)}$ was the response; for *Sim3* and *Sim4*, $\mathbf{y}_i^{(3)}$ was the response; and for *Sim5* and *Sim6*, $\mathbf{y}_i^{(8)}$, $\mathbf{y}_i^{(9)}$ and $\mathbf{y}_i^{(10)}$ were the responses.

6.4 Results of the simulation comparison

We first evaluated the performance of BASS and the other methods in terms of recovering the correct number of sparse and dense factors in the six simulations (Figures S3-S8). We calculated the percentage of correctly identified factors across 20 runs in the simulations with $n = 40$ (Table 4). Qualitatively, BASS recovered the closest matches to the simulated loading matrices across all methods (Figures 3, S1, S2). The correctly estimated loading matrices by the three different BASS initializations produced similar results, we only plot matrices from the PX-EM method.

6.4.1 RESULTS ON SIMULATIONS WITH TWO OBSERVATIONS (CCA)

Comparing results with two observations (*Sim1* and *Sim2*), our model produced the best SSIs and DSIs among all methods across all sample sizes (Figures 4). sGFA’s performance was limited for these simulations because the ARD prior does not produce sufficient element-wise sparsity, resulting in low SSIs (Figure 4). As a consequence of not matching sparse loadings well, sGFA had difficulty recovering dense loadings, especially with small sample sizes (Figure 4). GFA had difficulty recovering sparse loadings because of column-wise ARD priors with the same limitation (Figure 3, Figure 4). Its dense loadings were indirectly affected by the lack of sufficient sparsity for small sample sizes (Figure 4). RCCA also had difficulty in the two simulations because the recovered loadings were not sufficiently sparse using the ℓ_2 -type penalty (Figure 3).

SCCA recovered shared sparse loadings well in *Sim1* (Figure 3). However SCCA does not model local covariance structure, and therefore was unable to recover the sparse loadings specific to either of the observations in *Sim1* (Figure 3A) resulting in poor SSIs (Figure

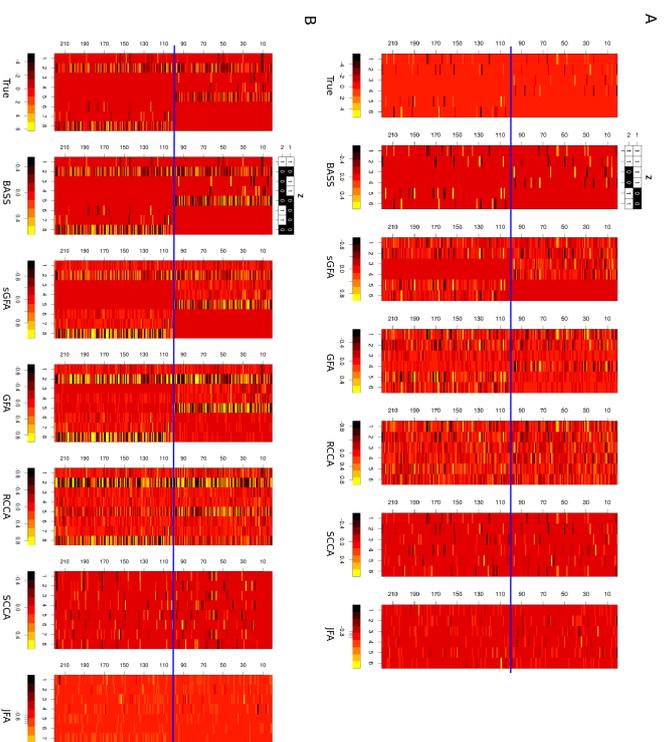


Figure 3: **Simulation results with two paired observations.** We reordered the columns of the recovered matrices and, where necessary, multiplied columns by -1 for easier visual comparisons. Horizontal lines separate the two observations. Panel A: Comparison of the recovered loading matrices using different models on *Sim1*. Panel B: Comparison of the recovered loading matrices using different models on *Sim2*.

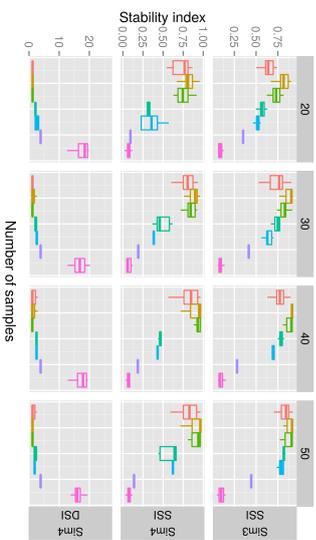


Figure 5: Comparison of stability indices on recovered loading matrices with four observations. Each stability index is plotted across 20 runs. For SSL, a larger value indicates better recovery; for DSL, a smaller value indicates better recovery. The boundaries of the box are the first and third quartiles. The line extends to the highest and lowest values that are within 1.5 times the distance of the first and third quartiles beyond the box boundaries.

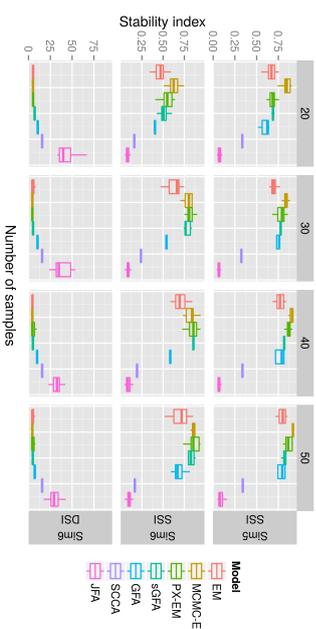


Figure 6: Comparison of stability indices on recovered loading matrices with ten observations. Each stability index is plotted across 20 runs. For SSL, a larger value indicates better recovery; for DSL, a smaller value indicates better recovery. The boundaries of the box are the first and third quartiles. The line extends to the highest and lowest values within 1.5 times the distance of the first and third quartiles beyond the box boundaries.

n_s	BASS													
	EM		MCMC-EM		PX-EM		sGFA		GFA		SCCA		MRD-hn	
	Err	SD	Err	SD	Err	SD	Err	SD	Err	SD	Err	SD	Err	SD
10	1.01	0.020	1.00	0.011	1.00	0.007	0.99	0.008	1.00	0.002	0.99	1.49	0.001	0.005
30	0.88	0.023	0.86	0.018	0.87	0.005	0.87	0.005	0.88	0.002	0.99	0.97	0.005	0.005
50	0.85	0.007	0.85	<1e-3	0.85	0.002	0.86	0.003	0.87	0.001	1.01	0.92	0.039	0.039
100	0.85	0.006	0.84	<1e-3	0.84	0.001	0.84	0.001	0.83	0.001	0.96	1.06	0.105	0.105
200	0.61	0.164	0.57	0.116	0.51	0.031	0.58	0.012	0.75	0.011	0.97	1.00	<1e-3	0.006
Sim5	0.49	0.160	0.40	0.093	0.38	0.007	0.43	0.006	0.40	0.005	0.98	0.46	0.006	0.006
Sim6	0.44	0.089	0.39	0.011	0.39	0.004	0.41	0.002	0.40	0.001	1.01	0.42	0.009	0.009
Sim7	0.39	0.033	0.39	0.004	0.39	0.001	0.39	0.002	0.39	0.001	1.01	0.42	0.249	0.249
200	0.58	0.063	0.58	0.001	0.58	0.001	0.59	0.001	0.59	0.001	1.01	0.40	0.050	0.050

Table 7: Prediction mean squared error with ten observations on $n_s = 200$ test samples. Test samples $y_i^{(s)}$, $y_i^{(9)}$ and $y_i^{(10)}$ are treated as the response and the rest of the observations are used as the training data to estimate parameters used to predict the response. Prediction accuracy is measured by mean squared error (MSE) between simulated responses and predicted responses. Values presented are the mean MSE (Err) and standard deviation (SD) across 20 runs of each method. Standard deviation (SD) is missing for SCCA because the method is deterministic.

S2). For the stability indices, BASS with MCMC-EM and PX-EM produced the best SSIs in *Sim5* across all methods and for almost all sample sizes (Figures 6). Here sGFA achieved equal or better SSIs than BASS EM, highlighting the sensitivity of BASS EM to initializations. GFA had equivalent or worse SSIs than BASS EM. In this pair of simulations, the advantages of BASS for flexible and robust column-wise and element-wise shrinkage are apparent (Figures 6). BASS also achieved the best prediction performance in *Sim5* and *Sim6* with ten observations (Table 6).

Across the three BASS methods, MCMC-EM had the most accurate performance across nearly all simulation settings. However, this performance boost comes with the price of running a small number of Gibbs sampling iterations with complexity of $O(k^3p + k^2pn)$ per iteration. When p is large, even a few iterations are computationally infeasible. PX-EM, on the other hand, has the same complexity as EM, and showed robust and accurate simulation results relative to EM. In the following real applications, we used BASS EM initialized with a small number of iterations of PX-EM.

7. Applying BASS to Mulan Library, genomics data, and text analysis

In this section we considered three real data applications of BASS. In the first application, we evaluated the prediction performance for multiple correlated response variables in the Mulan Library (Tosumakas et al., 2011). In the second application, we applied BASS to gene expression data from the Cholesterol and Pharmacogenomic (CAP) study. The data consist of expression measurements for about ten thousands genes in 480 lymphoblastoid cell lines (LCLs) under two experimental conditions (Mangravite et al., 2013; Brown et al., 2013). BASS was used to detect sparse covariance structures specific to each experimental condition. In the third application, we applied BASS to approximately 20,000 newsgroup posts to 20 newsgroups (Joachims, 1997) in order to perform multiclass classification.

7.1 Multivariate response prediction: The Mulan Library

The Mulan Library consists of multiple data sets collected for the purpose of evaluating multi-label predictions (Tosumakas et al., 2011). This library was used to test the Bayesian CCA model (GFA in our simulations) for multi-label prediction vectors converted to multi-label binary label vectors (one-hot encoding) (Klami et al., 2013). There are two observations

($m = 2$): the matrix of labels were treated as one observation ($\mathbf{Y}^{(1)}$) and the features were treated as another ($\mathbf{Y}^{(2)}$). Recently Mulan added multiple regression data sets with continuous variables. We chose ten benchmark data sets from the Mulan Library. Four of them (`bibtex`, `delicious`, `mediamill`, `scene`) have binary responses and were studied previously (Klami et al., 2013). Another six data sets (`rf1`, `rf2`, `scmid`, `scm20d`, `atp1d`, `atp7d`) have continuous responses (Table 8). For all data sets, we removed features with identical values for all samples in the training set as uninformative. For the continuous response data sets, for each value, we subtracted the mean and divided by the standard deviation of each feature.

We ran BASS, sGFA, GFA, and MRD-lin on the ten data sets, and compared the results using prediction accuracy. For data sets with binary labels, we quantified prediction error using the Hamming loss between the predicted labels and true labels. The predicted labels on the test samples were calculated using the same thresholding rules as in earlier work (Klami et al., 2013). The value of the threshold was chosen so that the Hamming loss between the estimated labels and the true labels in the training set was minimized. We used the R package `PresenceAbsence` and Matlab function `perfcurve` to find the thresholds to produce binary classifications from continuous predictions. In particular, the R package `PresenceAbsence` selects the threshold by maximizing the percent correctly classified, which corresponds to minimizing the Hamming loss. For continuous variables, mean squared error (MSE) was used to evaluate prediction accuracy. We initialized BASS with 500 factors and 50 PX-EM iterations. The other models were set to the default parameters with the number of factors set to $\min(p_1, p_2, 50)$ (see Simulations for details). All methods were run 20 times, and minimum errors were reported (Tables S1-S11).

BASS achieved the best prediction accuracy in five of the ten data sets (Table 8). For the data sets with a binary response, sGFA produced the best performance compared with other methods, achieving the smallest MSE in all four data sets. GFA had the most stable results in terms of SD in the four data sets. For the continuous response, BASS outperformed the other models in four out of six data sets. GFA again had the most stable MSE compared with other methods. The good performance of BASS on the data sets with continuous response variables may be attributed to the structured sparsity on the loading matrix, achieving the intended gains in generalization error from flexible regularization. Although the ARD prior used in GFA did not produce consistently sparse loadings, this model generated the most stable predictive results.

7.2 Gene expression data analysis

We applied our BASS model to gene expression data from the Cholesterol and Pharmacogenomic (CAP) study, consisting of expression measurements for 10,195 genes in 480 lymphoblastoid cell lines (LCLs) after 24-hour exposure to either a control buffer ($\mathbf{Y}^{(1)}$) or $2\mu M$ simvastatin acid ($\mathbf{Y}^{(2)}$) (Mangravite et al., 2013; Brown et al., 2013). In this example, the number of observations ($m = 2$) represents gene expression levels on the same samples and genes after the two different exposures. The expression levels were preprocessed to adjust for experimental traits (batch effects and cell growth rate) and clinical traits of donors (age, BMI, smoking status, and sex). We projected the adjusted expression levels to the quantiles of a standard normal within gene to control for outlier effects and applied BASS

Data Set	p_1	p_2	n_t	n_s	BASS		sGFA		GFA		MRD-lin	
					Err	SD	Err	SD	Err	SD	Err	SD
bibtex	1836	159	4880	2515	0.014	0.001	0.014	0.001	0.014	<1e-3	0.014	0.001
delicious	983	500	12920	3185	0.016	0.001	0.016	<1e-3	0.017	<1e-3	0.020	<1e-3
mediamill	120	101	30993	12914	0.032	0.001	0.032	0.005	0.034	<1e-3	0.043	<1e-3
scene	294	6	1211	1196	0.131	0.016	0.123	0.029	0.130	0.002	0.138	0.026
rf1	64	8	4108	5017	0.292	0.050	0.390	0.008	0.309	<1e-3	0.370	0.146
rf2	576	8	4108	5017	0.271	0.027	0.478	0.004	0.427	0.001	0.438	0.160
scmid	280	16	8145	1658	0.211	0.005	0.225	0.028	0.213	<1e-3	0.212	0.163
scm20d	61	16	7463	1503	0.650	0.015	0.538	0.006	0.720	0.002	0.608	0.033
atp1d	370	6	237	100	0.176	0.032	0.208	0.006	0.201	0.001	0.219	0.113
atp7d	370	6	196	100	0.597	0.063	0.537	0.015	0.537	0.003	0.545	0.049

Table 8: **Multi-variate response prediction in the Mulan library.** p_1 : the number of features; p_2 : the number of responses; n_t : the number of training samples; n_s : the number of test samples. The first four data sets have binary responses, and the final six are continuous responses. For binary responses, error (Err) is evaluated using Hamming loss between predicted labels and test labels in test samples. For continuous responses, mean squared error (MSE) is used to quantify error. Values shown are the minimum Hamming loss or MSE across 20 runs, and the standard deviation (SD).

with the initial number of factors set to $k = 2, 000$. We performed parameter estimation 100 times on these data with 100 iterations of PX-EM to initialize EM. Across these 100 runs, the estimated number of recovered factors was approximately 870 (Table S2), with only a few dense factors (Table S12) likely due to the adjustments made in the preprocessing step. The total percentage of variance explained (PVE) by the recovered latent structure was 14.73%, leaving 85.27% of the total variance to be captured in the residual error.

We computed the PVE of the sparse factors alone (Figure S9A). The PVE for the h^{th} factor was calculated as the variance explained by the h^{th} factor divided by the total variance: $\text{tr}(\boldsymbol{\Lambda}_h \boldsymbol{\Lambda}_h^T) / \text{tr}(\boldsymbol{\Lambda} \boldsymbol{\Lambda}^T + \boldsymbol{\Sigma})$. Shared sparse factors explained more variance than observation-specific sparse factors, suggesting that variation in expression levels across genes was driven by structure shared across the exposures to a greater degree than by exposure-specific structure. Moreover, 87.5% of the observation-specific sparse factors contained fewer than 100 genes, and 0.7% had more than 500 genes. The shared sparse factors had on average, more genes than the observation-specific factors: 72% shared sparse factors had fewer than 100 genes, and 4.5% had more than 500 genes. (Figure S9B).

The sparse factors specific to each observation characterized the local sparse covariance estimates. As we pursue more carefully elsewhere (Gao et al., 2014), we used observation-specific sparse factors to construct a gene co-expression network that is uniquely found in the samples from that exposure while explicitly controlling for shared covariance across exposures (Zou et al., 2013). The problem of constructing condition specific co-expression networks has been studied by both machine learning and computational biology communities (Li, 2002; Ma et al., 2011). BASS provides an alternative approach to solve this problem. We denote $\mathbf{B}_s^{(w)}$ as the sparse loadings in $\mathbf{B}^{(w)}$ ($w \in \{1, 2\}$) and $\mathbf{X}_s^{(w)}$ as the factors corresponding to sparse loadings for observation w . Then, $\boldsymbol{\Omega}_s^{(w)} = \mathbf{B}_s^{(w)} \text{Var}(\mathbf{X}_s^{(w)}) \mathbf{B}_s^{(w)T} + \boldsymbol{\Sigma}^{(w)}$ represents the regularized estimate of the covariance matrix specific to each observation after controlling for the contributions of the dense factors.

In our model, $\text{Var}(\mathbf{X}_s^{(w)}) = \mathbf{I}$, and so the covariance matrix becomes $\mathbf{\Omega}_s^{(w)} = \mathbf{B}_s^{(w)}(\mathbf{B}_s^{(w)})^T + \mathbf{\Sigma}^{(w)}$. We inverted this positive definite covariance matrix to get a precision matrix $\mathbf{R}^{(w)} = (\mathbf{\Omega}_s^{(w)})^{-1}$. The partial correlation between gene j_1 and j_2 , representing the correlation between the two features conditioned on the remaining features, is then calculated by normalizing each entry in the precision matrix (Edwards, 2000; Schäfer and Sührmer, 2005):

$$\rho_{j_1 j_2}^{(w)} = -\frac{r_{j_1 j_2}^{(w)}}{\sqrt{r_{j_1 j_1}^{(w)} r_{j_2 j_2}^{(w)}}}.$$

A partial correlation that is (near) zero for two genes (j_1, j_2) suggests that they are conditionally independent; non-zero partial correlation implies a direct relationship between two genes, and a network edge is added between the genes. The resulting undirected network is an instance of a Gaussian Markov random field, also known as a Gaussian graphical model (Edwards, 2000; Koller and Friedman, 2009). We note that BASS was the only method that enables construction of a condition specific network: sGFA could not be applied to data of this magnitude, GFA did not shrink the column selection sufficiently to recover sparsity in the condition specific covariance matrix, and SCGA only recovers shared sparse projections.

We used the following method to combine the results of 100 runs to construct a single observation-specific gene co-expression network for each observation. For each run, we first constructed a network by connecting genes with partial correlation greater than a threshold (0.01). Then we combined the 100 run-specific networks to construct a single network by removing all network edges that appeared in fewer than 50 (50%) of the networks. The two observation-specific gene co-expression networks contained 160 genes and 1,244 edges (buffer treated, Figure 7A), and 154 genes and 1,030 edges (statin-treated, Figure 7B), respectively.

7.3 Twenty newsgroups analysis

In this application, we used BASS and related methods for multiclass classification in the 20 Newsgroups data (Joachims, 1997). The documents were processed so that duplicates and headers were removed, resulting 18,846 documents. The data were downloaded using the `scikit-learn` Python package (Pedregosa et al., 2011). We converted the raw data into TF-IDF feature vectors and selected 319 words using SVM feature selection from `scikit-learn`. One document had a zero vector across the subset of vocabulary words and was removed. We held out 10 documents at random from each newsgroup as test data (Table S14).

We applied BASS to the transposed data matrices with the 20 newsgroups as 20 observations. We set the initial number of factors to $k = 1,000$ and ran EM 100 times from random starting points, each with 100 initial PX-EM iterations. There were on average 820 factors recovered across the runs.

To analyze the newsgroup-specific words, we calculated the Pearson correlation of each estimated loading and newsgroup indicator vectors consisting of ones for all of the documents in one newsgroup and zeros for documents in the other groups. Then, for each newsgroup, the loadings with the ten largest absolute value correlation coefficients were used to find the

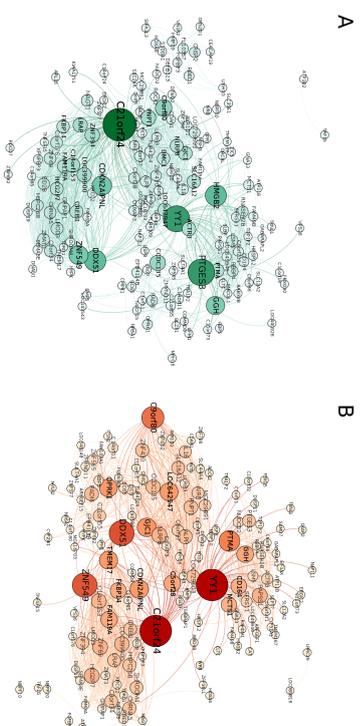


Figure 7: **Observation-specific gene co-expression networks from the CAP data.** The two networks represent the co-expressed genes specific to buffer-treated samples (Panel A) and statin-treated samples (Panel B). The node size is scaled according to the number of shortest paths from all vertices to all others that pass through that node (*betweenness centrality*).

ten words with the largest absolute value factor scores. The results from one run include, for example, the `rec.autos` newsgroup with ‘car’, ‘dealer’ and ‘oil’ as top words, and the `rec.sport.baseball` newsgroup with ‘baseball’, ‘braves’, and ‘runs’ as top words (Table 9).

We further partitioned the newsgroups into six classes according to subject matter to analyze the top words across newsgroups subgroups (Table 10). As above, we calculated the Pearson correlation with the binary indicator vectors for documents in newsgroup subgroups, and we analyzed the top ten words in the ten factors with largest absolute value correlation coefficients with these subsets of newsgroups (Table 10). We found, for example, that the newsgroups `talk.religion.misc`, `alt.athelism` and `soc.religion.christian` had ‘god’, ‘bible’ and ‘christian’ as top shared words. Examining one of the selected shared loadings for this newsgroup subgroup (Figure 8A), we noticed that documents outside of these three newsgroups, for the most part, have negligible loadings. This analysis highlights the ability of BASS to recover meaningful shared structure among 20 observations.

News group classes	Top ten shared words	News group classes	Top ten shared words
comp.graphics	windows	dos	shipping
comp.os.ms-windows.misc	thanks	mac	ca
comp.sys.ibm.pc.hardware	graphics	go	condition
comp.sys.mac.hardware	file	scsi	wanted
comp.windows.x	window	server	offer
	dod	baseball	forsale
rec.autos	car	rifle	government
rec.motorcycles	bike	talk.politics.misc	israeli
rec.sport.baseball	motorcycle	talk.politics.guns	jews
rec.sport.hockey	game	talk.politics.mideast	gun
	clipper	henry	firearms
sci.crypt	encryption	orbit	god
sci.electronics	space	people	bible
sci.med	chip	circuit	christian
sci.space	digex	voltage	clh
			jesus
			church

Table 10: Top ten words in the factors shared among specific subgroups of news-groups. In the shared recovered components corresponding to subsets of news groups, we show the ten most significant words in these shared components for six different subsets of news groups.

To assess prediction quality, we used the factors estimated from the training set to classify documents in the test set into one of 20 news groups. To estimate the loadings in the test set, we left-multiplied the test data matrix by the Moore-Penrose pseudoinverse of factors estimated from training data. This gave a rough estimate of the loading matrix for test data. Then test labels were predicted using the ten nearest neighbors in the loading rows estimated for the training documents. For the 200 test documents, BASS achieved 58.3% accuracy (Hamming loss; Figure 8B). Because some of the news groups were closely related to each other with respect to topic, we partitioned the 20 news groups into six topics according to subject matter. Then, the ten nearest neighbors were used to predict the topic of the test data. In this experiment, BASS achieved approximately 74.12% accuracy (Hamming loss; Figure 8C; Table S3).

8. Discussion

There exists a rich set of methods to explore latent structure in paired or multiple observations jointly (e.g., Parkhomenko et al., 2009; Witten and Tibshirani, 2009; Zhao and Li, 2012, among others). The multiple trajectories of interpretation of these approaches as linear factor analysis models includes the original inter-battery and multi-battery models (Browne, 1979, 1980), the probabilistic CCA model (Bach and Jordan, 2005), the sparse probabilistic projection (Archambeau and Bach, 2009), and, most recently, the Bayesian CCA model (Klami et al., 2013) and GFA model (Klami et al., 2014b). Only recently has the idea of column-wise shrinkage, or group-wise sparsity, been applied to develop useful

alt.atheism	comp.graphics	comp.os.ms-windows.misc	comp.sys.ibm.pc.hardware	comp.sys.mac.hardware
islamm	graphics	file	drive	mac
keith	3d	go	motherboard	apple
okforun	tiff	dos	thanks	quadra
atheism	image	cica	of	duo
livesey	image	dos	isa	centris
comp.windows.x	misc.forsale	rec.autos	rec.motorcycles	rec.sport.baseball
window	sale	car	dod	baseball
motif	for	cars	bike	litter
server	the	engine	motorcycle	ball
widget	sell	ford	ride	year
lcs	condition	cars	bike	players
rec.sport.hockey	sci.crypt	sci.electronics	sci.med	sci.space
hockey	encryption	circuit	geb	people
nhl	clipper	usa	medical	orbit
game	chip	usa	diet	henry
team	key	pgp	caucer	moon
leafs	des	tapped	audio	slutleg
soc.religion.christian	talk.politics.guns	talk.politics.mideast	talk.politics.misc	talk.religion.misc
god	af	israeli	government	morality
clh	firearms	stratus	optlink	jesus
church	guns	batf	kaldis	religion
christian	gun	stratus	clinton	god
heaven	handheld	waco	cramer	christian
			tax	objecting

Table 9: Most significant words in the news group-specific factors for 20 news groups. For each news group, we include the top ten words in the news group-specific components.

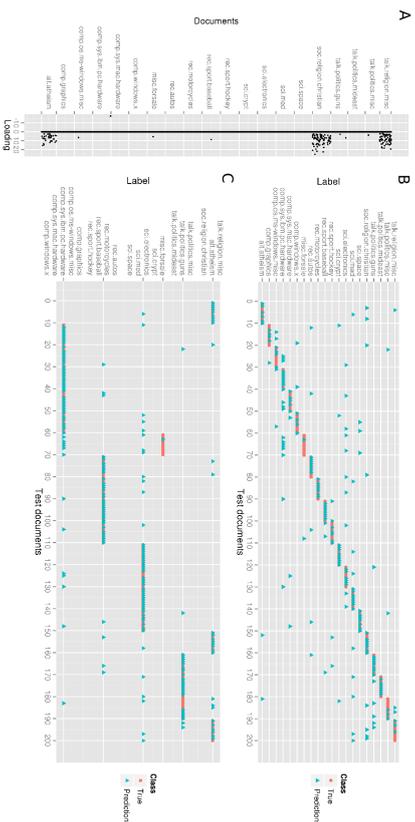


Figure 8: **Newsrroup prediction on 200 test documents.** Panel A: One factor loading selected as shared by three newsgroups (`talk.religion.misc`, `alt.atheism` and `soc.religion.christian`). Panel B: 20 Newsrroups predictions on 200 test documents using ten nearest neighbors from loadings estimated from the training data. Panel C: Document subgroup predictions based on six groups of similar newsgroups using ten nearest neighbors based on loadings estimated from the training data.

models for this problem. The advantage of column-wise shrinkage is to decouple portions of the latent space from specific observations and adaptively select the number of factors.

While the innovation of column-wise sparsity is primarily due to the ideas developed in the Bayesian CCA model (Virtanen et al., 2011), additional layers of shrinkage were required to create both column-wise and element-wise sparsity as is essential in real data analyses. The most recent attempt to develop such combined effects is the sGFA model (Khan et al., 2014) using a combination of an element-wise ARD prior with spike-and-slab prior for column selection. In our work here, we developed the necessary Bayesian prior and methodology framework to realize these advantages for the analysis of large data sets. In particular, we developed a structured sparse prior using three hierarchical layers of the three parameter beta (TPB) distribution. This carefully formulated prior combines both column-wise and element-wise shrinkage with global shrinkage to adapt the level of sparsity—both column-wise and element-wise—to the underlying data, creating robustness to parameter settings that cannot be achieved using a single-layer ARD prior. The resulting BASS model also allows sparse and dense factor loadings, which proved essential for data scenarios that have this low-rank and sparse structure and has been pursued in classical statistics (Chandrasekaran et al., 2009; Candès et al., 2011; Zhou et al., 2011). We showed in the simulations that this regularization is essential for problems in the $p \gg n$ data scenario, which motivated this work. With the assumption of full column rank of dense loadings and one single observation, our model provides a Bayesian solution to the sparse and low-rank decomposition problem.

Column-wise shrinkage in BASS was achieved using the observation-specific global and column-specific TPB priors. With current parameter settings, it is equivalent to the horseshoe prior put on the entire column. The horseshoe prior has been shown to induce better shrinkage effects compared to the ARD prior, the Laplace prior (Bayesian lasso), and other similar shrinkage priors while remaining computationally tractable (Carvalho et al., 2010). In addition, our local shrinkage encourages element-wise sparsity. A two component mixture allows both dense and sparse factors to be recovered for any subset of observations. These shared factors have an interpretation as a supervised low-rank projection when one observation is supervised labels (e.g., the Mulan Library data). To the best of our knowledge, the BASS model is the first model in either the Bayesian or classical statistical literature that is able to capture low-rank and sparse decompositions among multiple observations.

We developed three algorithms that estimate the posterior distribution of our model or MAP parameter values. We found that EM with random initialization would occasionally get stuck in poor local optima. This motivated the development of a fast and robust PX-EM algorithm by introducing an auxiliary rotation matrix (Rocková and George, 2015). Initializing EM with PX-EM enabled EM to escape from poor initializations, illustrated in simulations. Our PX-EM and EM algorithms have better computational complexity than two competing approaches, GFA and sGFA, allowing for large-scale data application.

Extending multiple observation linear factor models to non-linear or non-Gaussian models has been studied recently (Salomatin et al., 2009; Damianou et al., 2012; Klami et al., 2014a; Klami, 2014). The ideas in this paper of inducing structured sparsity in the loadings has parallels in both of these settings. For example, we may consider structured Gaussian process kernels in the non-linear setting, where structure corresponds to known shared and observation-specific structure. A number of issues remain, including robustness of the recovered sparse factors across runs, scaling these methods to current studies in genomics, neuroscience, or text analysis, allowing for missing data, and developing approaches to include domain-specific structure across samples or features.

Acknowledgments

The authors would like to thank David Dunson and Sanvesh Srivastava for helpful discussions. The authors also appreciate constructive comments from Arto Klami and three anonymous reviewers. BEE, CG, and SZ were funded by NIH R00 HG0066265 and NIH R01 MH101822. SZ was also funded in part by NSF DMS-1418261 and a Graduate Fellowship from Duke University. SM was supported in part by NSF DMS-1418261, NSF DMS-1209155, NSF IIS-1320357, and AFOSR under Grant FA9550-10-1-0436. All code and data are publicly available. The software for BASS is available at <https://github.com/judyboon/BASS>. The gene expression data were acquired through Gene Expression Omnibus (GEO) Accession number GSE36868. We acknowledge the PARC investigators and research team, supported by NHLBI, for collection of data from the Cholesterol and Pharmacogenetics clinical trial.

Appendix A. Markov chain Monte Carlo (MCMC) algorithm for posterior inference

We first derive the MCMC algorithm with Gibbs sampling steps for BASS. We write the joint distribution of the full model as

$$\begin{aligned} p(\mathbf{Y}, \mathbf{X}, \Lambda, \Theta, \Delta, \Phi, \mathbf{T}, \eta, \gamma, \mathbf{Z}, \Sigma, \pi) \\ = p(\mathbf{Y} | \Lambda, \mathbf{X}, \Sigma) p(\mathbf{X}) \\ \times p(\Lambda | \Theta) p(\Theta | \Delta, \mathbf{Z}, \Phi) p(\Delta | \Phi | \mathbf{T}) p(\mathbf{T} | \eta) p(\eta | \gamma) \\ \times p(\Sigma) p(\mathbf{Z} | \pi) p(\pi), \end{aligned}$$

where $\Theta = \{\delta_{jh}^{(w)}\}$, $\Delta = \{\phi_h^{(w)}\}$, $\Phi = \{\tau_h^{(w)}\}$, $\mathbf{T} = \{\eta^{(w)}\}$, and $\gamma = \{\gamma^{(w)}\}$ are the collections of global-factor-local TPB prior parameters.

The full conditional distribution for latent factor \mathbf{x}_i is

$$\mathbf{x}_i | - \sim \mathcal{N}_k \left((\Lambda^T \Sigma^{-1} \Lambda + \mathbf{I})^{-1} \Lambda^T \Sigma^{-1} \mathbf{y}_i, (\Lambda^T \Sigma^{-1} \Lambda + \mathbf{I})^{-1} \right), \quad (14)$$

for $i = 1, \dots, n$.

For Λ , we derive the full conditional distributions of its p rows, λ_j , for $j = 1, \dots, p$,

$$\lambda_j^T | - \sim \mathcal{N}_k \left((\sigma_j^{-2} \mathbf{X} \mathbf{X}^T + \mathbf{D}_j + \mathbf{D}_j^{-1})^{-1} \sigma_j^{-2} \mathbf{X} \mathbf{y}_j^T, (\sigma_j^{-2} \mathbf{X} \mathbf{X}^T + \mathbf{D}_j^{-1})^{-1} \right),$$

where

$$\mathbf{D}_j^{-1} = \text{diag} \left((\theta_{j1}^{(w)}) I_{(z_1^{(w)})} I_{(z_1^{(w)})=1}, (\theta_{j2}^{(w)}) I_{(z_2^{(w)})=0}, \dots, (\theta_{jk}^{(w)}) I_{(z_k^{(w)})=1}, (\theta_{jk}^{(w)}) I_{(z_k^{(w)})=0} \right),$$

and w_j represents the observation that the j^{th} row belongs to.

The full conditional distributions of $\theta_{jh}^{(w)}$, $\delta_{jh}^{(w)}$ and $\phi_h^{(w)}$ with $z_h^{(w)} = 1$ are

$$\begin{aligned} \theta_{jh}^{(w)} | - &\sim \mathcal{IG} \left(a - 1/2, 2\delta_{jh}^{(w)}, (\lambda_{jh}^{(w)})^2 \right), \\ \delta_{jh}^{(w)} | - &\sim \mathcal{Ga} \left(a + b, \phi_h^{(w)} + \theta_{jh}^{(w)} \right), \\ \phi_h^{(w)} | - &\sim \mathcal{Ga} \left(p_w b + c, \sum_{j=1}^{p_w} \delta_{jh}^{(w)} + \tau_h^{(w)} \right), \end{aligned}$$

where \mathcal{IG} is the generalized inverse Gaussian distribution.

The full conditional distribution of $\phi_h^{(w)}$ with $z_h^{(w)} = 0$ is

$$\phi_h^{(w)} | - \sim \mathcal{IG} \left(c - p_w/2, 2\tau_h^{(w)}, \sum_{j=1}^{p_w} (\lambda_{jh}^{(w)})^2 \right).$$

The full conditional distributions of the remaining parameters are

$$\tau_h^{(w)} | - \sim \mathcal{Ga}(c + d, \phi_h^{(w)} + \eta^{(w)}),$$

$$\begin{aligned} \eta^{(w)} | - &\sim \mathcal{Ga} \left(kd + e, \gamma^{(w)} + \sum_{h=1}^k \tau_h^{(w)} \right), \\ \gamma^{(w)} | - &\sim \mathcal{Ga}(e + f, \eta^{(w)} + \nu), \\ \pi^{(w)} | - &\sim \text{beta} \left(1 + \sum_{h=1}^k z_h^{(w)}, 1 + k - \sum_{h=1}^k z_h^{(w)} \right). \end{aligned}$$

The full conditional distribution of $z_h^{(w)}$ is

$$\begin{aligned} \Pr(z_h^{(w)} = 1 | -) &\propto \pi^{(w)} \prod_{j=1}^{p_w} \mathcal{N}(\lambda_{jh}^{(w)}; 0, \theta_{jh}^{(w)}) \mathcal{Ga}(\theta_{jh}^{(w)}; a, \delta_{jh}^{(w)}) \mathcal{Ga}(\delta_{jh}^{(w)}; b, \phi_h^{(w)}), \\ \Pr(z_h^{(w)} = 0 | -) &\propto (1 - \pi^{(w)}) \prod_{j=1}^{p_w} \mathcal{N}(\lambda_{jh}^{(w)}; 0, \phi_h^{(w)}). \end{aligned}$$

We further integrate out $\delta_{jh}^{(w)}$ in $\Pr(z_h^{(w)} = 1 | -)$:

$$\begin{aligned} \Pr(z_h^{(w)} = 1 | -) &\propto \pi^{(w)} \prod_{j=1}^{p_w} \int \mathcal{N}(\lambda_{jh}^{(w)}; 0, \theta_{jh}^{(w)}) \mathcal{Ga}(\theta_{jh}^{(w)}; a, \delta_{jh}^{(w)}) \mathcal{Ga}(\delta_{jh}^{(w)}; b, \phi_h^{(w)}) d\delta_{jh}^{(w)} \\ &= \pi^{(w)} \prod_{j=1}^{p_w} \mathcal{N}(\lambda_{jh}^{(w)}; 0, \theta_{jh}^{(w)}) \frac{\Gamma(a+b) \Gamma(\theta_{jh}^{(w)})^{a-1} (\phi_h^{(w)})^b}{\Gamma(a)\Gamma(b) (\theta_{jh}^{(w)} + \phi_h^{(w)})^{a+b}}. \end{aligned}$$

The full conditional distribution of σ_j^{-2} for $j = 1, \dots, p$ is

$$\sigma_j^{-2} | - \sim \mathcal{Ga} \left(n/2 + a_\sigma, \frac{1}{2} (\mathbf{y}_j - \lambda_j \cdot \mathbf{X})(\mathbf{y}_j - \lambda_j \cdot \mathbf{X})^T + b_\sigma \right).$$

Appendix B. Variational expectation maximization (EM) algorithm for MAP estimates

Expectation Step: Given model parameters, the distribution of latent factor \mathbf{X} was written in Appendix A (Equation 14). The expected sufficient statistics of \mathbf{X} is

$$\begin{aligned} \langle \mathbf{x}_i \rangle &= (\Lambda^T \Sigma^{-1} \Lambda + \mathbf{I})^{-1} \Lambda^T \Sigma^{-1} \mathbf{y}_i, \\ \langle \mathbf{x}_i \mathbf{x}_i^T \rangle &= \langle \mathbf{x}_i \rangle \langle \mathbf{x}_i \rangle^T + (\Lambda^T \Sigma^{-1} \Lambda + \mathbf{I})^{-1}. \end{aligned} \quad (15) \quad (16)$$

The expectation of the indicator variable $\rho_h^{(w)} = \langle z_h^{(w)} \rangle$ is

$$\rho_h^{(w)} = \frac{\pi^{(w)} \prod_{j=1}^{p_w} \mathcal{N}(\lambda_{jh}^{(w)}; 0, \theta_{jh}^{(w)}) \mathcal{Ga}(\theta_{jh}^{(w)}; a, \delta_{jh}^{(w)}) \mathcal{Ga}(\delta_{jh}^{(w)}; b, \phi_h^{(w)})}{(1 - \pi^{(w)}) \prod_{j=1}^{p_w} \mathcal{N}(\lambda_{jh}^{(w)}; 0, \phi_h^{(w)}) + \pi^{(w)} \prod_{j=1}^{p_w} \mathcal{N}(\lambda_{jh}^{(w)}; 0, \theta_{jh}^{(w)}) \mathcal{Ga}(\theta_{jh}^{(w)}; a, \delta_{jh}^{(w)}) \mathcal{Ga}(\delta_{jh}^{(w)}; b, \phi_h^{(w)})}.$$

Maximization Step: The log posterior of Λ is written as

$$\log(p(\Lambda | -)) \propto \text{tr}(\Sigma^{-1} \Lambda \mathbf{S} \mathbf{X}^Y) - \frac{1}{2} \text{tr}(\Lambda^T \Sigma^{-1} \Lambda \mathbf{S} \mathbf{X} \mathbf{X}) - \frac{1}{2} \sum_{h=1}^k \lambda_h^T \mathbf{D}_h \lambda_{-h},$$

where

$$\begin{aligned} \mathbf{D}_h &= \text{diag} \left(\frac{\rho_h^{(1)}}{\phi_h^{(1)}} + \frac{1 - \rho_h^{(1)}}{\phi_h^{(m)}}, \dots, \frac{\rho_h^{(m)}}{\phi_{p_m h}^{(m)}} + \frac{1 - \rho_h^{(m)}}{\phi_h^{(m)}} \right), \\ \mathbf{S}^{\mathbf{X}^{\mathbf{Y}}} &= \sum_{i=1}^n \langle \mathbf{x}_i; \mathbf{y}_i^T \rangle, \text{ and } \mathbf{S}^{\mathbf{X}^{\mathbf{X}}} = \sum_{i=1}^n \langle \mathbf{x}_i; \mathbf{x}_i^T \rangle. \end{aligned}$$

We take the derivative with respect to the loading column λ_h to get the MAP estimate. The derivative of first part in the right hand side is

$$\begin{aligned} \frac{\partial \text{tr}(\boldsymbol{\Sigma}^{-1} \mathbf{A} \mathbf{S}^{\mathbf{X}^{\mathbf{Y}}})}{\partial \lambda_h} &= (\mathbf{1}_k^h \otimes \mathbf{I}_p) \times \text{vec}[\boldsymbol{\Sigma}^{-1} \mathbf{S}^{\mathbf{Y}^{\mathbf{X}}}] = \text{vec} \left(\boldsymbol{\Sigma}^{-1} \mathbf{S}^{\mathbf{Y}^{\mathbf{X}}} \mathbf{1}_k^h \right) \\ &= \boldsymbol{\Sigma}^{-1} \mathbf{S}^{\mathbf{Y}^{\mathbf{X}}} \mathbf{1}_k^h, \end{aligned}$$

where vec is the vectorization of a matrix. $\mathbf{1}_k^h \in \mathbb{R}^{k \times 1}$ is a zero vector with a single 1 in the h^{th} element, and $\mathbf{S}^{\mathbf{Y}^{\mathbf{X}}} = (\mathbf{S}^{\mathbf{X}^{\mathbf{Y}}})^T$. For the second part

$$\begin{aligned} \frac{\partial \text{tr}(\mathbf{A}^T \boldsymbol{\Sigma}^{-1} \mathbf{A} \mathbf{S}^{\mathbf{X}^{\mathbf{X}}})}{\partial \lambda_h} &= 2(\mathbf{1}_k^h \otimes \mathbf{I}_p) \times \text{vec}[\boldsymbol{\Sigma}^{-1} \mathbf{A} \mathbf{S}^{\mathbf{X}^{\mathbf{X}}}] = 2 \times \text{vec} \left(\boldsymbol{\Sigma}^{-1} \mathbf{A} \mathbf{S}^{\mathbf{X}^{\mathbf{X}}} \mathbf{1}_k^h \right) \\ &= 2\boldsymbol{\Sigma}^{-1} \mathbf{A} \mathbf{S}^{\mathbf{X}^{\mathbf{X}}} \mathbf{1}_k^h. \end{aligned}$$

For the third part, the derivative is $\mathbf{D}_h \lambda_h$. The MAP estimates for λ_h are found by setting the derivative to zero:

$$\hat{\lambda}_h = |\mathbf{S}^{\mathbf{X}^{\mathbf{X}}} \mathbf{I}_p + \boldsymbol{\Sigma} \mathbf{D}_h|^{-1} \left(\mathbf{S}^{\mathbf{Y}^{\mathbf{X}}} - \sum_{h' \neq h} \lambda_{h'} \mathbf{S}^{\mathbf{X}^{\mathbf{X}}} \right),$$

where $\mathbf{S}_{ij}^{\mathbf{X}^{\mathbf{X}}}$ is the $(i, j)^{\text{th}}$ element of $\mathbf{S}^{\mathbf{X}^{\mathbf{X}}}$, and $\mathbf{S}_{jh}^{\mathbf{Y}^{\mathbf{X}}}$ is the h^{th} column of $\mathbf{S}^{\mathbf{Y}^{\mathbf{X}}}$. The matrix inverse is for a diagonal matrix; thus $\hat{\lambda}_h$ can be calculated efficiently. The MAP estimate for the other model parameters are found from their full conditional distributions with the latent variables replaced by their expectations. We list the parameter updates for those variables here

$$\begin{aligned} \hat{\phi}_{jh}^{(w)} &= \frac{2a - 3 + \sqrt{(2a - 3)^2 + 8(\lambda_{jh}^{(w)})^2 \delta_{jh}^{(w)}}}{4\delta_{jh}^{(w)}}, \\ \hat{\delta}_{jh}^{(w)} &= \frac{a + b}{\theta_{jh}^{(w)} + \phi_{jh}^{(w)}}, \\ \hat{\phi}_h^{(w)} &= \frac{p' - 1 + \sqrt{(p' - 1)^2 + a'H}}{a'}, \text{ with} \\ p' &= \rho_h^{(w)} p_{wb} - (1 - \rho_h^{(w)}) p_{w2} / 2 + c, \\ a' &= 2(\rho_h^{(w)} \sum_{j=1}^{p_w} \delta_{jh}^{(w)} + \tau_h^{(w)}), \end{aligned}$$

$$b' = (1 - \rho_h^{(w)}) \sum_{j=1}^{p_w} (\lambda_{jh}^{(w)})^2$$

$$\hat{\gamma}_h^{(w)} = \frac{c + d}{\phi_h^{(w)} + \eta^{(w)}},$$

$$\hat{\eta}^{(w)} = \frac{dk + e}{\gamma^{(w)} + \sum_{h=1}^k \tau_h^{(w)}},$$

$$\hat{\gamma}^{(w)} = \frac{e + f}{\eta^{(w)} + \nu},$$

$$\hat{\pi}^{(w)} = \frac{\sum_{h=1}^k \rho_h^{(w)}}{k},$$

$$\hat{\sigma}_j^{-2} = \frac{n/2 + a_\sigma - 1}{1/2(\mathbf{y}_j - \lambda_j \langle \mathbf{X} \rangle)(\mathbf{y}_j - \lambda_j \langle \mathbf{X} \rangle)^T + b_\sigma}.$$

Appendix C. Parameter-expanded EM (PX-EM) algorithm for robust MAP estimates

We introduce a positive semidefinite matrix \mathbf{R} in our original model to obtain a parameter-expanded version:

$$\begin{aligned} \mathbf{y}_i &\sim \mathbf{A} \mathbf{R}_L^{-1} \mathbf{x}_i + \epsilon_i, \\ \mathbf{x}_i &\sim \mathcal{N}_k(\mathbf{0}, \mathbf{R}), \\ \epsilon_i &\sim \mathcal{N}_k(\mathbf{0}, \boldsymbol{\Sigma}). \end{aligned}$$

Here, \mathbf{R}_L is the lower triangular part of the Cholesky decomposition of \mathbf{R} . Marginally, the covariance matrix is still $\boldsymbol{\Omega} = \mathbf{A} \mathbf{A}^T + \boldsymbol{\Sigma}$, as this additional parameter keeps the likelihood invariant. This additional parameter reduces the coupling effects between the updates of loading matrix and latent factors (Liu et al., 1998; Dyk and Meng, 2001) and serves to connect different posterior modes with equal likelihood curves indexed by \mathbf{R} (Rocková and George, 2015).

Let $\mathbf{A}^* = \mathbf{A} \mathbf{R}_L^{-1}$ and $\boldsymbol{\Xi}^* = \{\mathbf{A}^*, \boldsymbol{\Theta}, \boldsymbol{\Delta}, \boldsymbol{\Phi}, T, \boldsymbol{\eta}, \boldsymbol{\gamma}, \boldsymbol{\pi}, \boldsymbol{\Sigma}\}$. Then the parameters of our expanded model are $\{\boldsymbol{\Xi}^* \cup \mathbf{R}\}$. We assign our structured prior on \mathbf{A}^* . Thus, the updates of $\boldsymbol{\Xi}^*$ are unchanged given the estimates of the first and second moments of \mathbf{X} . The estimates of $\langle \mathbf{X} \rangle$ and $\langle \mathbf{X} \mathbf{X}^T \rangle$ are calculated using Equations (15 and 16) in Appendix B after mapping the loading matrix back to the original matrix: $\mathbf{A} = \mathbf{A}^* \mathbf{R}_L$. It remains to estimate \mathbf{R} . Write the expected complete log likelihood in the expanded model as

$$Q(\boldsymbol{\Xi}^*, \mathbf{R} | \boldsymbol{\Xi}^{(s)}) = \mathbb{E}_{\mathbf{X}, \mathbf{Z} | \boldsymbol{\Xi}^{(s)}, \mathbf{Y}, \mathbf{R}_0} \log(p(\boldsymbol{\Xi}^*, \mathbf{R}, \mathbf{X}, \mathbf{Z} | \mathbf{Y})).$$

The only term involving \mathbf{R} is $p(\mathbf{X})$. Therefore, the \mathbf{R} that maximizes this function is

$$\mathbf{R}_{(s)} = \arg \max_{\mathbf{R}} Q(\boldsymbol{\Xi}^*, \mathbf{R} | \boldsymbol{\Xi}^{(s)}) = \arg \max_{\mathbf{R}} \left(\text{const} - \frac{n}{2} \log |\mathbf{R}| - \frac{1}{2} \text{tr}(\mathbf{R}^{-1} \mathbf{S}^{\mathbf{X}^{\mathbf{X}}}) \right).$$

The solution is $\mathbf{R}_{(s)} = \frac{1}{n} \mathbf{S}^{\mathbf{X}^{\mathbf{X}}}$.

The EM algorithm in this parameter-expanded space generates the sequence $\{\Xi^*(1) \cup \mathbf{R}_{(1)}, \Xi^*(2) \cup \mathbf{R}_{(2)}, \dots\}$. This sequence corresponds to a sequence of parameter estimates in the original space $\{\Xi_{(1)}, \Xi_{(2)}, \dots\}$, where \mathbf{A} in the original space is equal to $\mathbf{A}^* \mathbf{R}_l$ (Rocková and George, 2015). We initialize $\mathbf{R}_{(0)} = \mathbf{I}_k$.

References

- Cédric Archambeau and Francis R. Bach. Sparse probabilistic projections. In *Advances in Neural Information Processing Systems 21*, pages 73–80, 2009.
- Artin Armagan, Merlise Clyde, and David B. Dunson. Generalized beta mixtures of Gaussians. In *Advances in Neural Information Processing Systems 24*, pages 523–531, 2011.
- Artin Armagan, David B. Dunson, and Jaeyong Lee. Generalized double Pareto shrinkage. *Statistica Sinica*, 23(1):119, 2013.
- Francis R. Bach and Michael I. Jordan. A probabilistic interpretation of canonical correlation analysis. *Technical Report 688, Department of Statistics, University of California, Berkeley*, 2005.
- Anirban Bhattacharya and David B. Dunson. Sparse Bayesian infinite factor models. *Biometrika*, 98(2):291–306, 2011.
- Anirban Bhattacharya, Debdeep Pati, Natesh S. Pillai, and David B. Dunson. Dirichlet-Laplace priors for optimal shrinkage. *Journal of the American Statistical Association*, Accepted for publication, 2014.
- Christopher D. Brown, Lara M. Mangravite, and Barbara E. Engelhardt. Integrative modeling of eQTLs and cis-regulatory elements suggests mechanisms underlying cell type specificity of eQTLs. *PLoS Genetics*, 9(8):e1003649, 2013.
- Michael W. Browne. The maximum-likelihood solution in inter-battery factor analysis. *British Journal of Mathematical and Statistical Psychology*, 32(1):75–86, 1979.
- Michael W. Browne. Factor analysis of multiple batteries by maximum likelihood. *British Journal of Mathematical and Statistical Psychology*, 33(2):184–199, 1980.
- Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):11, 2011.
- Carlos M. Carvalho, Jeffrey Chang, Joseph E. Lucas, Joseph R. Nevins, Quanli Wang, and Mike West. High-dimensional sparse factor modeling: Applications in gene expression genomics. *Journal of the American Statistical Association*, 103(484), 2008.
- Carlos M. Carvalho, Nicholas G. Polson, and James G. Scott. Handling sparsity via the horseshoe. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5, pages 73–80, 2009.
- Carlos M. Carvalho, Nicholas G. Polson, and James G. Scott. The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480, 2010.
- ZHAO, GAO, MUKHERJEE, ENGELHARDT
- Venkat Chandrasekaran, Sujay Sanghavi, Pablo A. Parrilo, and Alan S. Willsky. Sparse and low-rank matrix decompositions. In *47th Annual Allerton Conference on Communication, Control, and Computing*, pages 962–967, 2009.
- Venkat Chandrasekaran, Sujay Sanghavi, Pablo A. Parrilo, and Alan S. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.
- Pierre Comon. Independent component analysis, A new concept? *Signal Processing*, 36(3):287–314, 1994.
- John P. Cunningham and Zoubin Ghahramani. Linear dimensionality reduction: Survey, insights, and generalizations. *Journal of Machine Learning Research*, 16, 2015.
- Andreas Damianou, Carl Ek, Michalis Titsias, and Neil Lawrence. Manifold relevance determination. In *29th International Conference on Machine Learning*, pages 145–152, 2012.
- Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B*, 39(1):1–38, 1977.
- David A. van Dyk and Xiao-Li Meng. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50, 2001.
- David Edwards. *Introduction to Graphical Modelling*. Springer, New York, 2nd edition, June 2000. ISBN 9780387950549.
- Carl Henrik Ek, Jon Rihani, Philip H.S. Torr, Grégory Rogez, and Neil D. Lawrence. Ambiguity modeling in latent spaces. In *Machine Learning for Multimodal Interaction*, pages 62–73. Springer, 2008.
- Barbara E. Engelhardt and Ryan P. Adams. Bayesian structured sparsity from Gaussian fields. *arXiv:1407.2295*, 2014.
- Barbara E. Engelhardt and Matthew Stephens. Analysis of population structure: A unifying framework and novel methods based on sparse factor analysis. *PLoS Genetics*, 6(9):e1001117, 2010.
- Chuan Gao, Christopher D. Brown, and Barbara E. Engelhardt. A latent factor model with a mixture of sparse and dense factors to model gene expression data with confounding effects. *arXiv:1310.4792*, 2013.
- Chuan Gao, Shiwen Zhao, Ian C. McDowell, Christopher D. Brown, and Barbara E. Engelhardt. Differential gene co-expression networks via Bayesian biclustering models. *arXiv:1411.1997*, 2014.
- Ignacio González, Sébastien Déjean, Pascal G.P. Martin, and Alain Baccini. CCA: An R package to extend canonical correlation analysis. *Journal of Statistical Software*, 23(12):1–14, 2008.

- Thomas L. Griffiths and Zoubin Ghahramani. The Indian buffet process: An introduction and review. *The Journal of Machine Learning Research*, 12:1185–1224, 2011.
- Chris Hans. Bayesian lasso regression. *Biometrika*, 96(4):835–845, 2009.
- Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.
- Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417, 1933.
- Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.
- Junzhou Huang, Tong Zhang, and Dimitris Metaxas. Learning with structured sparsity. *The Journal of Machine Learning Research*, 12:3371–3412, 2011.
- Rodolphe Jenatton, Guillaume Obozinski, and Francis Bach. Structured sparse principal component analysis. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 366–373, 2010.
- Rodolphe Jenatton, Jean-Yves Audibert, and Francis Bach. Structured variable selection with sparsity-inducing norms. *The Journal of Machine Learning Research*, 12:2777–2824, 2011.
- Yangqing Jia, Mathieu Salzmann, and Trevor Darrell. Factorized latent spaces with structured sparsity. In *Advances in Neural Information Processing Systems 23*, pages 982–990, 2010.
- Thorstan Joachims. A probabilistic analysis of the Rocchio algorithm with TF-IDF for text categorization. In *Proceedings of the 14th International Conference on Machine Learning*, pages 143–151, 1997.
- Suleiman A. Khan, Seppo Virtanen, Olli P. Kallioniemi, Kristofer Wannerberg, Antti Poso, and Samuel Kaski. Identification of structural features in chemicals associated with cancer drug response: A systematic data-driven analysis. *Bioinformatics*, 30(17):1497–1504, 2014.
- Arto Klami. Polya-gamma augmentations for factor models. In *The 6th Asian Conference on Machine Learning*, pages 112–128, 2014.
- Arto Klami and Samuel Kaski. Probabilistic approach to detecting dependencies between data sets. *Neurocomputing*, 72(1):39–46, 2008.
- Arto Klami, Seppo Virtanen, and Samuel Kaski. Bayesian canonical correlation analysis. *Journal of Machine Learning Research*, 14:965–1003, 2013.
- Arto Klami, Guillaume Bouchard, and Abhishek Tripathi. Group-sparse embeddings in collective matrix factorization. In *International Conference on Learning Representations*, 2014a.
- Arto Klami, Seppo Virtanen, Eemeli Leppäaho, and Samuel Kaski. Group factor analysis. *IEEE Transactions on Neural Networks and Learning Systems*, 26(9):2136–2147, 2014b.
- David Knowles and Zoubin Ghahramani. Nonparametric Bayesian sparse factor models with application to gene expression modeling. *The Annals of Applied Statistics*, 5(2B):1534–1552, 2011.
- Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, first edition, July 2009.
- Mathieu Kowalski. Sparse regression using mixed norms. *Applied and Computational Harmonic Analysis*, 27(3):303–324, 2009.
- Mathieu Kowalski and Bruno Torrèsani. Structured sparsity: From mixed norms to structured shrinkage. In *Processing with Adaptive Sparse Structured Representations*, 2009.
- Minjung Kyung, Jeff Gill, Malay Ghosh, and George Casella. Penalized regression, standard errors, and Bayesian lassos. *Bayesian Analysis*, 5(2):369–411, 2010.
- Neil Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *The Journal of Machine Learning Research*, 6:1783–1816, 2005.
- Jeffrey T. Leek, Robert B. Schaefer, Hector Corrada Bravo, David Simcha, Benjamin Langmead, W. Evan Johnson, Donald Geman, Keith Baggerly, and Rafael A. Irizarry. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics*, 11(10):733–739, 2010.
- Ken-Chan Li. Genome-wide coexpression dynamics: Theory and application. *Proceedings of the National Academy of Sciences*, 99(26):16875–16880, 2002.
- Chunhai Lin, Donald B. Rubin, and Ying Nian Wu. Parameter expansion to accelerate EM: The PX-EM algorithm. *Biometrika*, 85(4):735–770, 1998.
- Joseph E. Lucas, Hsin-Ni Kung, and Jen-Tsan A. Chi. Latent factor analysis to discover pathway-associated putative segmental aneuploidies in human cancers. *PLoS Computational Biology*, 6(9):e1000920, 2010.
- Haisu Ma, Eric E. Schadt, Lee M. Kaplan, and Hongyu Zhao. COSINE: Condition-specific sub-network identification using a global optimization method. *Bioinformatics*, 27(9):1290–1298, 2011.
- Lara M. Mangravite, Barbara E. Engelhardt, Marisa W. Medina, Joshua D. Smith, Christopher D. Brown, Daniel I. Chasman, Brigham H. Meacham, Bryan Howe, Heejung Shim, Deseah Naidoo, et al. A statin-dependent QTL for *CATM* expression is associated with statin-induced myopathy. *Nature*, 502(7471):377–380, 2013.
- Roderick P. McDonald. Three common factor models for groups of variables. *Psychometrika*, 35(1):111–128, 1970.
- Toby J. Mitchell and John J. Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032, 1988.

- Radford M. Neal. *Bayesian learning for neural networks*. PhD thesis, University of Toronto, 1995.
- Anthony O'Hagan. On outlier rejection phenomena in Bayes inference. *Journal of the Royal Statistical Society, Series B*, 41(3):358–367, 1979.
- Trevor Park and George Casella. The Bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.
- Elena Parkhomenko, David Tritchler, and Joseph Beyene. Sparse canonical correlation analysis with application to genomic data integration. *Statistical Applications in Genetics and Molecular Biology*, 8(1):1–34, 2009.
- Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.
- Nicholas G. Polson and James G. Scott. Shrink globally, act locally: Sparse Bayesian regularization and prediction. In *Bayesian Statistics 9*, eds. *J.M. Bernardo et al.*, pages 501–538. Oxford University Press, 2011.
- Iosifina Pounmara and Lorenz Wernisch. Factor analysis for gene regulatory networks and transcription factor activity profiles. *BMC Bioinformatics*, 8:61, 2007.
- Julian Pruteanu-Malinici, Daniel L. Mace, and Uwe Ohler. Automatic annotation of spatial expression patterns via Bayesian factor models. *PLoS Computational Biology*, 7(7):e1002098, 2011.
- Xinjuan Qu and Xinlei Chen. Sparse structured probabilistic projections for factorized latent spaces. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 1389–1394, 2011.
- Priyadip Ray, Lingling Zheng, Joseph Lucas, and Lawrence Carin. Bayesian joint analysis of heterogeneous genomics data. *Bioinformatics*, 30(10):1370–1376, 2014.
- Veronika Rocková and Edward I. George. Fast Bayesian factor analysis via automatic rotations to sparsity. *Journal of the American Statistical Association*, 2015.
- Justin K. Romberg, Hyeokho Choi, and Richard G. Baraniuk. Bayesian tree-structured image modeling using wavelet-domain hidden Markov models. *IEEE Transactions on Image Processing*, 10(7):1056–1068, 2001.
- Sam Roweis. EM algorithms for PCA and SPCA. In *Advances in Neural Information Processing Systems 10*, pages 626–632, 1998.
- Konstantin Salomatkin, Yiming Yang, and Abhimanyu Lad. Multi-field correlated topic modeling. In *SIAM International Conference on Data Mining*, pages 628–637, 2009.
- Mathieu Salzmann, Carl H. Ek, Raquel Urtasun, and Trevor Darrell. Factorized orthogonal latent spaces. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 701–708, 2010.
- Juliane Schäfer and Korbinian Strimmer. An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics*, 21(6):754–764, 2005.
- Aaron Shou, Keith Grochow, Aaron Hertzmann, and Rajesh P. Rao. Learning shared latent structure for image synthesis and robotic imitation. In *Advances in Neural Information Processing Systems 18*, pages 1233–1240, 2005.
- Tommi Suviavaal, Juuso A. Parkkinen, Seppo Virtanen, and Samuel Kaski. Cross-organism toxicogenomics with group factor analysis. *Systems Biomedicine*, 2:e29291, 2014.
- Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996.
- Michael E. Tipping. Sparse Bayesian learning and the relevance vector machine. *The Journal of Machine Learning Research*, 1:211–244, 2001.
- Michael E. Tipping and Christopher M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 1999a.
- Michael E. Tipping and Christopher M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61(3):611–622, 1999b.
- Grigorios Tsoumakas, Eleftherios Spyromitros-Xioulfis, Jozef Vilcek, and Ioannis Vlahavas. Mulan: A Java library for multi-label learning. *Journal of Machine Learning Research*, 12:2411–2414, 2011.
- Seppo Virtanen, Arto Klami, and Samuel Kaski. Bayesian CCA via group sparsity. In *Proceedings of the 28th International Conference on Machine Learning*, pages 457–464, 2011.
- Seppo Virtanen, Arto Klami, Suleiman A. Khan, and Samuel Kaski. Bayesian group factor analysis. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22, pages 1269–1277, 2012.
- Zaiwen Wen and Wotao Yin. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 142(1-2):397–434, 2013.
- Mike West. On scale mixtures of normal distributions. *Biometrika*, 74(3):646–648, 1987.
- Mike West. Bayesian factor regression models in the "large p, small n" paradigm. In *Bayesian Statistics 7*, eds. *J.M. Bernardo et al.*, pages 723–732. Oxford University Press, 2003.
- Daniela Witten, Rob Tibshirani, Sam Gross, and Balasubramanian Narasimhan. *PMA: Penalized Multivariate Analysis*, 2013. URL <http://CRAN.R-project.org/package=PMA>. R package version 1.0-9.

- Daniela M. Witten and Robert J. Tibshirani. Extensions of sparse canonical correlation analysis with applications to genomic data. *Statistical Applications in Genetics and Molecular Biology*, 8(1):1–27, 2009.
- Daniela M. Witten, Robert Tibshirani, and Trevor Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534, 2009.
- Angqi Wu, Mijung Park, Oluwasami O Koyejo, and Jonathan W Pillow. Sparse Bayesian structure learning with “dependent relevance determination priors. In *Advances in Neural Information Processing Systems*, pages 1628–1636, 2014.
- Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, 68(1):49–67, 2006.
- Peng Zhao, Guilherme Rocha, and Bin Yu. The composite absolute penalties family for grouped and hierarchical variable selection. *The Annals of Statistics*, 37(6A):3468–3497, 2009.
- Shiwen Zhao and Shao Li. A co-module approach for elucidating drug-disease associations and revealing their molecular basis. *Bioinformatics*, 28(7):955–961, 2012.
- Tianyi Zhou, Daoheng Tao, and Xindong Wu. Manifold elastic net: A unified framework for sparse dimension reduction. *Data Mining and Knowledge Discovery*, 22(3):340–371, 2011.
- Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, 67(2):301–320, 2005.
- Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15(2):265–286, 2006.
- James Y Zou, Daniel J Hsu, David C Parkes, and Ryan P Adams. Contrastive learning using spectral methods. In *Advances in Neural Information Processing Systems*, pages 2238–2246, 2013.

Machine Learning in an Auction Environment

Patrick Hummel

1600 Amphitheatre Parkway

Google Inc.

Mountain View, CA 94043, USA

PHUMMEL@GOOGLE.COM

R. Preston McAfee

One Microsoft Way

Microsoft Corp.

Redmond, WA 98052, USA

PRESTON@MCAFEE.CC

Editor: Shie Mannor

Abstract

We consider a model of repeated online auctions in which an ad with an uncertain click-through rate faces a random distribution of competing bids in each auction and there is discounting of payoffs. We formulate the optimal solution to this explore/exploit problem as a dynamic programming problem and show that efficiency is maximized by making a bid for each advertiser equal to the advertiser's expected value for the advertising opportunity plus a term proportional to the variance in this value divided by the number of impressions the advertiser has received thus far. We then use this result to illustrate that the value of incorporating active exploration in an auction environment is exceedingly small.

Keywords: Auctions, Explore/exploit, Machine learning, Online advertising

1. Introduction

In standard Internet auctions in which bidders bid by specifying how much they are willing to pay per click, it is standard to rank the advertisers by a product of their bid and their click-through rate, or their expected cost-per-1000-impressions (eCPM) bids. While this is a sensible way to determine the best ad to show for a particular query, it is potentially a suboptimal approach if one cares about showing the best possible ads in the long run. In online auctions, new ads are constantly entering the system, and for these ads one will typically have uncertainty in the true eCPM of the ad due to the fact that one will not know the click-through rate of a brand new ad with certainty. In this case, it can be desirable to show an ad where one has a high amount of uncertainty about the true eCPM of the ad so one can learn more about the ad's true eCPM by observing whether the ad received a click. Thus even if one believes that a high uncertainty ad is not the best ad for this particular query, it may be valuable to show this ad so one can learn more about the eCPM of the ad and make better decisions about whether to show this ad in the future.

While there is an extensive literature that analyzes strategic experimentation in these types of multi-armed bandit problems, the online advertising setting differs substantially from these existing models. In online auctions there is a tremendous amount of random variation in the quality of competition that an ad with unknown eCPM faces in the auction

due to the fact that the ad is constantly competing in a wide variety of different auctions. In these settings, there will always be a certain amount of free exploration that takes place due to the fact that there will be some auctions in which there are no ads with eCPMs that are known to be high, and one can use these opportunities to explore ads with uncertain eCPMs. Almost all existing models of multi-armed bandits that can be applied to online auctions fail to take this possibility into account.

This paper presents a model of repeated auctions in which an ad with an uncertain click-through rate faces a random distribution of competing bids in each auction and there is discounting of payoffs in the sense that an auctioneer values a dollar received in the distant future less highly than a dollar received today. We formulate this problem as a dynamic programming problem and show that the optimal solution to this problem takes a remarkably simple form. In each period, the auctioneer should rank the advertisers on the basis of the sum of an advertiser's expected eCPM plus a term that represents the value of learning about the eCPM of a particular ad. One then runs the auction by ranking the ads by these social values rather than their expected eCPMs.

While there have been previous papers on multi-armed bandits that have proposed ranking arms by a term equal to the expected value of showing an ad plus an additional term representing the value of learning about the true value of that arm,¹ the value of learning in the problem that we consider is dramatically different from the value of learning in standard multi-armed bandit problems. In standard multi-armed bandit problems (Auer et al., 2002) where there is no discounting of payoffs and no random variation in the competition that an arm faces, typical solutions involve ranking the ads according to a sum of the expected value of the arm plus a term proportional to the standard deviation in the arm's value. By contrast, we find that the value of learning in our setting is proportional to the variance in an ad's expected eCPM divided by the number of impressions that an ad has received. Thus the incremental increase in the probability that a particular ad is shown varies with $\frac{1}{k^2}$, where k denotes the number of impressions this ad has received so far. This is an order of magnitude smaller than the corresponding incremental increase in standard machine learning algorithms. In fact, we show that if we attempted to rank the ads on the basis of the sum of an advertiser's expected eCPM plus a term equal to a constant times the standard deviation in the advertiser's eCPM, the optimal constant would be zero.

Our baseline model considers a simple situation in which there is a single advertiser with unknown eCPM that competes in each period against an advertiser with known eCPM whose eCPM bid is a random draw from some distribution. But our conclusions about the value of learning are not restricted to this simple model. We show that our conclusions about the optimal bidding strategies extend to a variety of more complicated models including models in which there are multiple advertisers with unknown eCPMs as well as models in which there is correlation between the unknown eCPMs of multiple different advertisers and information from showing one advertiser can help one refine one's estimate of the eCPM for some other advertiser. We also illustrate an asymptotic equivalence between

1. In addition, Iyer et al. (2014) illustrate that bidders may have an incentive to make a bid equal to their expected value plus a term proportional to their value of learning about their value if bidders have uncertainty about their own value. This paper differs from ours in that it considers an environment in which bidders are attempting to learn their own values rather than an auctioneer attempting to learn the eCPMs.

the theoretically optimal strategies and the strategies that would be selected by a simple one-step look ahead policy often referred to as “knowledge gradients” (Prazier et al., 2009; Ryzhov et al., 2010, 2012).

A consequence of these small incremental changes in the probability that an ad is shown is that the total value from adding active exploration in the online auction setting is exceedingly small. Not only does the incremental increase in the probability that a particular ad is shown vary with $\frac{1}{k^2}$, but on top of that, the expected payoff increase that one obtains conditional on showing a different ad than would be shown without active learning also varies with $\frac{1}{k^2}$. This implies that the total value of adding active exploration in the setting we consider will vary with $\frac{1}{k^4}$ for large numbers of impressions k , an exceedingly small amount.

We further obtain finite sample results illustrating that for realistic amounts of uncertainty in the eCPMs of ads, the maximum total efficiency gain that could ever be achieved by adding active learning in this auction environment is exceedingly small, typically only a few hundredths of a percentage point. Finally, we empirically verify these findings through simulations and illustrate that adding active learning in the auction environment we consider only changes overall efficiency by a few hundredths of a percentage point.

Perhaps the most closely related paper to our work is a paper by Li et al. (2010). This paper is the only other paper we are aware of that considers questions related to the value of learning about the eCPMs of ads with uncertain eCPMs in a setting where there is discounting in payoffs as well as random variation in the quality of the competition that an ad faces from competing ads in the auction. Li et al. (2010) demonstrate that the value of showing an ad with an uncertain eCPM will generally exceed the immediate value of showing that ad because one will learn information about the eCPM of the ad that will enable one to make better ranking decisions in the future. However, Li et al. (2010) do not attempt to characterize the optimal solution in this setting, as we do in the present paper.

There is also an extensive literature in statistics and machine learning that addresses questions related to multi-armed bandits (Audibert and Bubeck, 2010; Auer et al., 2002, 2003; Gittins, 1979; Hazan and Kale, 2011; Lai and Robbins, 1985; Mannor and Tsiitsiklis, 2004; May et al., 2012; Shivkins, 2014) as well as some papers that focus specifically on the auction context (Agrawal et al., 2009; Babaioff et al., 2009; Devanur and Kakade, 2009; Wortman et al., 2007). However, none of these papers considers appropriate methods for exploring ads in a context where there is random variation in the quality of the competition that an ad faces in an auction. The optimal methods for exploring ads in such a scenario turn out to be completely different from the methods considered in any of these previous papers, and as such, our work is completely different from existing machine learning literature.

Finally, there is an extensive literature in economics related to questions on strategic experimentation. Within economics, this literature has considered a variety of questions including consumers trying to learn about the quality of various products (Bergemann and Välimäki, 1996, 1997, 2000), firms and sellers trying to learn about demand (Aghion et al., 1993; Fishman and Rob, 1998; Ghate, 2015; Keller and Rady, 1999; Mirman et al., 1993; Rustichini and Wolinsky, 1995), learning to play repeated games (Anthoussen, 2002; Gale and Rosenthal, 1999), learning about untried policies in political economy (Callander, 2011; Callander and Hummel, 2014; Strulovici, 2010), learning from the actions of others (Banerjee and Fudenberg, 2004; Gale, 1996; Vives, 1997), as well as general results on experimentation

(Aghion et al., 1991; Banks and Sundaram, 1992; Bergemann and Välimäki, 2001; Bolton and Harris, 1999; Brezzi and Lai, 2002; Keller and Rady, 2010; Keller et al., 2005; Moscarini and Smith, 2001; Rothschild, 1974; Schlag, 1998; Weitzman, 1979). However, the economics literature has not considered strategic experimentation in auctions, as we do in the present paper.

2. The Model

There is a new ad with an uncertain eCPM that will bid into a second-price auction for a single advertising opportunity with competing advertisers.² Throughout we let x denote the actual unknown but fixed value (or eCPM) for showing the new ad, z denote the eCPM bid the auctioneer places on behalf of this advertiser,³ and let k denote the number of impressions the ad has received so far. We also suppose that the highest eCPM bid that this advertiser competes against may vary from auction to auction, and that in each auction, this highest competing eCPM bid is a random draw from some cumulative distribution function $F(\cdot)$ with corresponding continuous and twice differentiable density $f(\cdot)$.

At any given point in time, the auctioneer does not necessarily know the exact value of x . Instead the auctioneer only knows that x is drawn from some distribution. We let \bar{x} denote a generic distribution corresponding to the auctioneer’s estimate of the distribution of possible values of x . This distribution will evolve over time as an ad has received more impressions and we have a better sense of the underlying eCPM of the ad.

Throughout we also let \bar{x} denote an unbiased estimate of the true value of x given the auctioneer’s estimate of the distribution of possible values of x . We also let σ_k^2 denote the variance in our estimate of the eCPM for the new ad when the ad has been shown k times. In the limit when k is large, σ_k^2 will be well approximated by $\frac{s^2(\bar{x})}{k}$ for some constant $s^2(\bar{x})$ that depends only on \bar{x} , and we assume that $\sigma_k^2 = \frac{s^2(\bar{x})}{k} + \frac{h(\bar{x})}{k^2} + o(\frac{1}{k^2})$ for some continuously differentiable functions $s^2(\bar{x})$ and $h(\bar{x})$.⁴

In addition, we let $\delta \in (0, 1)$ denote the per-period discount rate so the auctioneer only values advertising opportunities that take place at time T by a factor of δ^T as much as opportunities that take place at the present time period. Throughout we assume that the auctioneer wishes to maximize total efficiency; that is, if v_t denotes the total value of the ad displayed in period t (the true eCPM of this ad), then the auctioneer’s payoff is $\sum_{t=0}^{\infty} \delta^t v_t$. Since online ad auctions are typically designed to select the efficiency-maximizing allocation, this is a logical objective to optimize.

2. These second-price auctions for a single advertising slot are ubiquitous throughout the display advertising industry. In such auctions, advertisers have an incentive to make a bid equal to their true value for a click.
3. Typically advertisers bid by indicating how much they are willing to pay per click, and the auctioneer then uses this cost-per-click bid as well as an estimate of the probability the ad will be clicked to calculate an eCPM bid for the advertiser that the auctioneer then places on behalf of the advertiser in the auction.
4. This assumption will hold for most common priors about the distribution from which the uncertain eCPM of the ad is drawn, such as a beta prior. It is also worth noting that the weaker assumption that $\sigma_k^2 = \frac{s^2(\bar{x})}{k} + O(\frac{1}{k^2})$ for some constant $s^2(\bar{x})$ is sufficient to prove our main result about the value of learning being $O(\frac{1}{k^2})$. The additional assumption that $\sigma_k^2 = \frac{s^2(\bar{x})}{k} + \frac{h(\bar{x})}{k^2} + o(\frac{1}{k^2})$ is only used to further prove that the value of learning is of the form $\frac{v(\bar{x})}{k(k+1)} + o(\frac{1}{k^2})$ for some function $v(\bar{x})$.

3. Preliminaries

Before proceeding to analyze the precise model given above, we first address a closely related question about the extent to which a particular advertising opportunity increases total welfare if the eCPM of this advertising opportunity is known. In particular, we consider a concept that we refer to as the *long-term value* of a particular advertisement. The long-term value of a particular advertisement gives the total increase in the auctioneer's payoff that arises as a result of this ad being in the system from the various auctions that take place over time. Understanding the long-term value of a particular advertisement when the eCPM of that ad is known will serve as a useful benchmark for understanding how one should behave when there is uncertainty about the eCPM of the ad.

Theorem 1 *If the eCPM of an ad is known, then the total long-term value of this ad is a convex function of the eCPM of the ad and a strictly convex function for regions where the eCPM of the ad is within the support of the distribution of the highest competing eCPM.*

All proofs are in the appendix. The fact that the value for any particular advertisement is a convex function of the eCPM of the ad if the eCPM of the ad is known indicates that if there is uncertainty about the eCPM of the ad, then the expected long-term value of this ad will be greater than the long-term value of the expected eCPM of the ad. From this it follows that if there is uncertainty about the eCPM of the ad, then it will be optimal to behave as if this particular ad had a known eCPM that is greater than the expected eCPM of the ad. The precise additional amount that this advertiser's bid should be increased will be pinned down by the solution to the dynamic programming problem governed by the game described in the model.

4. Dynamic Programming Problem

In this section, we formulate the value of a particular ad as a dynamic programming problem and use this formulation to derive the optimal bidding strategy. First we derive the auctioneer's payoff that arises in a particular period when the auctioneer makes a particular bid on behalf of the advertiser with uncertain eCPM.

Note that if the auctioneer places a bid of z on behalf of the advertiser with uncertain eCPM in the auction and the actual value of showing this particular ad is x , then the auctioneer's payoff from running the auction once is

$$\begin{aligned} u(z, x) &= \int_z^\infty yf(y) dy + \int_0^z xf(y) dy = -y(1-F(y))\Big|_z^\infty + \int_z^\infty (1-F(y)) dy + xF(z) \\ &= z(1-F(z)) + \int_z^\infty (1-F(y)) dy + xF(z). \end{aligned}$$

In general placing a bid of z rather than x in a one-shot auction will result in some inefficiencies in the one-shot auction since it would be optimal for efficiency to place a bid exactly equal to x on behalf of this advertiser in a one-shot auction. The payoff loss that arises in a one-shot auction as a result of placing a bid of z instead of x is

$$L = u(x, x) - u(z, x)$$

$$\begin{aligned} &= x(1-F(x)) + \int_x^\infty (1-F(y)) dy + xF(x) - z(1-F(z)) - \int_z^\infty (1-F(y)) dy - xF(z) \\ &= (x-z)(1-F(z)) + \int_x^z (1-F(y)) dy = \int_x^z F(z) - F(y) dy. \end{aligned}$$

If we define the per-period reward to be the negative of this per-period loss, then the auctioneer seeks to maximize the discounted sum of these per period rewards. Let $V_k(\bar{x})$ denote the value of this discounted sum when the auctioneer follows the optimal bidding strategy. Also note that, from the perspective of the auctioneer, x is a random variable that can be expressed as $x = \bar{x} + \sigma_k \epsilon$, where σ_k denotes the standard deviation in our estimate of the ad with uncertain eCPM when the ad has been shown k times, and ϵ is a random variable with mean zero and variance one. We use this notation to prove the following:

Lemma 2 $V_k(\bar{x})$ can be expressed as the value of a dynamic programming problem by

$$V_k(\bar{x}) = \frac{1}{1-\delta} \left(\max_z E_\epsilon \left[- \int_{\bar{x} + \sigma_k \epsilon}^z F(z) - F(y) dy + \delta F(z) (E_{\bar{x}}[V_{k+1}(\bar{x})] - V_k(\bar{x})) \right] \right),$$

where \bar{x} denotes the uncertain realization of \bar{x} after an ad receives an additional impression.

By using the expression for the value of the dynamic programming problem in the previous lemma, we can derive the bid that the auctioneer should place on behalf of the advertiser to maximize the auctioneer's payoff. This is done in the theorem below:

Theorem 3 *The optimal bidding strategy in the dynamic programming problem when an ad has been shown k times entails setting $z = \bar{x} + \delta(E_{\bar{x}}[V_{k+1}(\bar{x})] - V_k(\bar{x}))$.*

Thus the optimal bidding strategy in this dynamic programming problem can be written in a form where the bid the auctioneer makes on behalf of the bidder with uncertain eCPM is equal to the bidder's expected eCPM plus a term that represents the value of learning about the true eCPM of that bidder, $\delta(E_{\bar{x}}[V_{k+1}(\bar{x})] - V_k(\bar{x}))$. In order to calculate this value of learning, we need to get a sense of the size of the $V_k(\bar{x})$ terms.

5. Value of Dynamic Program for Large Numbers of Impressions

In the previous section, we have given exact expressions for the value of the dynamic program and the optimal bidding strategy that should be followed under this dynamic programming problem. In this section, we seek to derive accurate estimates of the value of this dynamic program in the limit when an ad has already been shown a large number of times.

The main purpose of this section is to illustrate that the value of learning term given in the previous section will vary with $\frac{1}{k^2}$ for large k . We prove this by first showing that the expected efficiency loss arising due to the uncertainty in the eCPM of the ad varies with $\frac{1}{k}$ for large k , and then use this to show that the value of learning term varies with $\frac{1}{k} - \frac{1}{k+1}$, which varies with $\frac{1}{k^2}$ for large k .

When an ad has already been shown a large number of times, the value of σ_k that is estimated for the ad is likely to be very small. For small values of σ_k , we can use a Taylor expansion to approximate the value of the above dynamic programming problem. In particular, we obtain the following result:

Lemma 4 $E_c \int_{\bar{x}-\sigma_k \epsilon}^{\bar{x}} F(z) - F(y) dy = \int_{\bar{x}}^z F(z) - F(y) dy + \frac{1}{2} \sigma_k^2 f(\bar{x}) + a(\bar{x}) \sigma_k^4 + o(\sigma_k^4)$ for some constant $a(\bar{x})$ for large k .

Using the results from the previous lemma, one can immediately illustrate that V_k must be on the order of $\frac{1}{k}$ for large values of k .

Theorem 5 $V_k(\bar{x}) = \Theta(\frac{1}{k})$ for large k .

To understand the intuition behind this result, note that the average error in the estimate of the eCPM of the ad is proportional to the standard error of this estimate, σ_k , which varies with $\frac{1}{\sqrt{k}}$, so the probability that the auctioneer will display the wrong ad as a result of miscalculating the eCPM of the ad varies with $\frac{1}{\sqrt{k}}$. At the same time, conditional on displaying the wrong ad as a result of miscalculating the eCPM of the ad, the average efficiency loss that one suffers varies with $\frac{1}{\sqrt{k}}$. Thus the expected efficiency loss that the auctioneer incurs varies with $\frac{1}{k}$, which in turn implies the result in Theorem 5.

Theorem 5 suggests that we may be able to write $V_k(\bar{x}) = -\frac{v(\bar{x})}{k} + o(\frac{1}{k})$ for large k , where v is a function that depends only on \bar{x} . To prove that $V_k(\bar{x})$ can be expressed this way, it is necessary to show that $kV_k(\bar{x})$ indeed converges to a function of \bar{x} in the limit as $k \rightarrow \infty$. This is done in the following theorem:

Theorem 6 $kV_k(\bar{x})$ converges to a function of \bar{x} in the limit as $k \rightarrow \infty$. Furthermore, it must be the case that $kV_k(\bar{x}) = -\frac{v(\bar{x})}{2(1-\delta)} s^2(\bar{x}) f(\bar{x}) + O(\frac{1}{k})$ for large k .

From Theorem 6, it follows that we can express $V_k(\bar{x})$ by $V_k(\bar{x}) = -\frac{v(\bar{x})}{k} + O(\frac{1}{k^2})$ for large k , where v is a function that satisfies $v(\bar{x}) = \frac{v(\bar{x})}{2(1-\delta)} s^2(\bar{x}) f(\bar{x})$. In order to complete our approximation of the solution the dynamic programming problem for large k , it is also necessary to bound the expression $E_{\bar{x}}[V_{k+1}(\bar{x}')] - V_k(\bar{x})$ that appears in the dynamic programming problem. This is done in the following theorem:

Theorem 7 $E_{\bar{x}}[V_{k+1}(\bar{x}')] - V_k(\bar{x}) = \frac{v(\bar{x})}{k(k+1)} + o(\frac{1}{k^2})$ for large k .

The intuition behind this result is that since the efficiency loss that the auctioneer incurs due to uncertainty in the eCPM of an ad varies with $\frac{1}{k}$, the value of learning will be proportional to the reduction in the future efficiency loss that the auctioneer suffers as a result of learning more about the eCPM of the ad, meaning the value of learning will vary with $\frac{1}{k} - \frac{1}{k+1}$, which varies with $\frac{1}{k^2}$. The fact that $E_{\bar{x}}[V_{k+1}(\bar{x}')] - V_k(\bar{x})$ varies with $\frac{1}{k^2}$ indicates that the incremental increase in an advertiser's bid also varies with $\frac{1}{k^2}$ in the limit when k is large. This in turn implies that the incremental increase in an advertiser's probability of winning the auction will also vary with $\frac{1}{k^2}$ for large k .

The result in Theorem 7 suggests that the optimal method for adding active exploration will only rarely have an effect on which ad wins the auction, as the probability that this active exploration changes which ad is shown varies with $\frac{1}{k^2}$ for large k . This result about the value of learning varying with $\frac{1}{k^2}$ for large k stands in marked contrast to algorithms that have been proposed for active exploration in standard multi-armed bandit problems with no discounting of payoffs and no random variation in the competition that an arm faces

in a given period (Auer et al., 2002). In these types of algorithms, the value of learning tends to vary with $\frac{1}{\sqrt{k}}$, which means the value of learning is an order of magnitude smaller in our setting than in standard multi-armed bandit problems.

Ultimately we seek to use these insights to derive results about the change in payoff that would result from incorporating active learning in this setting. Before doing this, we first illustrate how the conclusions of this section about the value of the dynamic programming problem and the optimal bidding strategy extend to a variety of more complicated scenarios including settings where there are multiple different ads with uncertain eCPMs whose true eCPMs may be correlated and we also illustrate a natural correspondence between the optimal solution to the full dynamic programming problem and a simple one-step look-ahead strategy. First we tackle the problem of computing the value of the dynamic program when an ad with an uncertain eCPM has only received a small number of impressions.

6. Value of Dynamic Program for Small Numbers of Impressions

To calculate the value of $V_k(\bar{x})$ for small values of k , we apply backwards induction. At some large value of k , it will necessarily be the case that the incremental value of additional exploration is so small that the advertiser simply bids $z = \bar{x}$ because the smallest possible increment the advertiser would be allowed to adjust its bid exceeds the tiny incremental value of additional exploration. Thus if K denotes the earliest stage at which an advertiser always sets $z = \bar{x}$, then for all $k \geq K$, it is necessarily the case that the value of learning is zero, and $V_k(\bar{x}) = \frac{1}{1-\delta} \left(E_c \left[-\int_{\bar{x}-\sigma_k \epsilon}^{\bar{x}} F(\bar{x}) - F(y) dy \right] \right) \approx 0$.

For values of $k < K$, we have

$$(1 - \delta) V_k(\bar{x}) = E_c \left[-\int_{\bar{x}+\sigma_k \epsilon}^{z_k} F(z_k) - F(y) dy \right] + \delta F(z_k) (E_{\bar{x}}[V_{k+1}(\bar{x}')] - V_k(\bar{x}))$$

or

$$V_k(\bar{x}) = \frac{E_c \left[-\int_{\bar{x}+\sigma_k \epsilon}^{z_k} F(z_k) - F(y) dy \right] + \delta F(z_k) E_{\bar{x}}[V_{k+1}(\bar{x}')] }{1 - \delta + \delta F(z_k)}.$$

Thus by empirically measuring the values of σ_k and $F(\cdot)$, we can apply backward induction to approximate $V_k(\bar{x})$ for small values of k . We now address the question of what these values of $V_k(\bar{x})$ will be approximately equal to for an important class of advertisers.

Many ads that have only received a small number of impressions are ads that typically fail to win auctions because the machine learning system is pessimistic about the ad's true eCPM. The estimated eCPMs for these ads may be several orders of magnitude smaller than the typical eCPMs of the ads that have been shown many times. In these cases, even if the percentage uncertainty in the eCPMs of these ads is quite high, the absolute amount of uncertainty in the eCPMs of these ads will be small compared to the typical eCPMs of the ads that have been shown many times. Thus in these cases, \bar{x} will be close to zero, and $F(\bar{x})$ and σ_k^2 will be close to zero as well. Under these circumstances, we have the following result:

Theorem 8 If \bar{x} (and σ_k^2) are close to zero for small values of k , then $V_k(\bar{x}) = -\frac{v(\bar{x})}{2(1-\delta)} f(\bar{x}) \sigma_k^2 + o(f(\bar{x}) \sigma_k^2)$ for small values of k .

Theorem 8 indicates that even in small sample environments, it is still frequently reasonable to approximate $V_k(\bar{x})$ by writing $V_k(\bar{x}) \approx -\frac{1}{2(1-\delta)} f(\bar{x}) \sigma_k^2$, where σ_k^2 denotes the variance in our estimate of the ad's eCPM for a particular value of k . This theorem in turn implies that if a machine learning system is quite pessimistic about the true eCPM of a new ad, then there will be little value to actively exploring the ad because the value of learning term, $E_{\mathcal{F}'}[V_{k+1}(\bar{x}')] - V_k(\bar{x})$, will be quite small.

7. Ads with Correlated Values

So far we have restricted attention settings in which we only seek to learn the eCPM of one advertiser's ad. However, in many situations we may seek to learn the eCPMs of multiple advertisers' ads and the eCPMs of the various advertisers may be correlated. In these situations, information about one ad's eCPM may help one learn about the eCPMs of other related advertisers. On top of this, even if there is only one ad for which we are uncertain about the advertiser's eCPM, this ad may bid in several different contexts where the ad has substantially different eCPMs and the ad faces substantially different competing landscapes of bids.⁵ In these cases, information about an ad's eCPM in one context may also help one learn about the ad's eCPM in other contexts.

To address how this affects the results, we extend the model to allow for the possibility that there are multiple different ads that bid in multiple different contexts where we seek to learn the eCPMs of the ads and these eCPMs may be correlated. In particular, we suppose that there are m different ad-context pairs where we seek to learn the eCPM of the ad in that particular context. For the ad-context pair a , we let x_a denote the actual, unknown value of the eCPM of that ad in that context, and we let $x = (x_1, \dots, x_m)$ denote the actual unknown eCPMs of the ads in all m contexts. We also let k denote the total number of impressions that these advertisers have received in the various contexts and let β_a denote the fraction of these impressions that were received in context a . Thus we have $\sum_{a=1}^m \beta_a = 1$.

We again assume the auctioneer does not know the exact value of x , and instead the auctioneer only knows that x is drawn from some distribution. We again let \bar{x} denote a generic distribution corresponding to the auctioneer's estimate of the distribution of possible values of x . This distribution allows for the possibility that the auctioneer may believe there is correlation in the unknown eCPMs of the advertisers in the different contexts, and the distribution will again evolve over time as an ad has received more impressions and we have a better sense of the underlying eCPM of the ad.

Throughout we also let \bar{x} denote an unbiased estimate of the true value of x given the auctioneer's estimate of the distribution of possible values of x and we let \bar{x}_a denote an unbiased estimate of the true value of x_a given this distribution. We also let σ_{a,k_a}^2 denote the variance in our estimate of the eCPM of the ad-context pair a when there have been a total of k_a impressions in this ad-context pair. In the limit when k_a is large, σ_{a,k_a}^2 will be well approximated by $\frac{s_a^2(\bar{x}_a)}{k_a} = \frac{s_a^2(\bar{x}_a)}{\beta_a k}$ for some constant $s_a^2(\bar{x}_a)$ that depends only on \bar{x}_a , and

5. Contextual bandit problems in which an arm's payoff may vary from context to context have appeared in the literature before in different settings. See, for example, work by May et al. (2012) and Shlivkins (2014).

we again let $\sigma_{a,k_a}^2 = \frac{s_a^2(\bar{x}_a)}{\beta_a k_a} + \frac{h_a(\bar{x}_a)}{\beta_a^2 k_a^2} + o(\frac{1}{k_a^2})$ for some continuously differentiable functions $s_a^2(\bar{x}_a)$ and $h_a(\bar{x}_a)$. In addition, we let $\delta \in (0, 1)$ denote the per-period discount rate so that the mechanism designer only values advertising opportunities that take place at time T by a factor of δ^T as much as opportunities that take place at the present time period.

In each period t , there is an auction for a single advertising opportunity. The auction can involve any one of the m possible ad-context pairs for which we do not know the eCPM of the ad in that context. We let π_a denote the probability that there will be an auction involving ad-context pair a in any given time period. Thus we have $\sum_{a=1}^m \pi_a = 1$. We further suppose that if there is an auction involving ad-context pair a , then the distribution of the values of the competing advertisers is such that the highest eCPM for a competing ad is a random draw from some cumulative distribution function $F_a(\cdot)$ with corresponding continuous and twice differentiable density $f_a(\cdot)$.

In this setting, the total long-term value of a particular ad-context pair is again a convex function of the eCPM of the ad for the same reasons as in Theorem 1 and we can again formulate this problem as a dynamic program. To do this, let $\vec{k} \equiv (k_1, \dots, k_m)$ denote a vector that gives the number of impressions that have been received by the various ad-context pairs $1, \dots, m$. Also let $V_{a,\vec{k}}(\bar{x})$ denote the value of the dynamic program when the next auction involves the advertiser-context pair a , the eCPMs of the ads are \bar{x} , and there have been \vec{k} impressions in each of the various ad-context pairs, and let $V_{\vec{k}}(\bar{x}) \equiv E_a[V_{a,\vec{k}}(\bar{x})]$ denote the value of the same dynamic program unconditional on which ad-context pair is involved in the next auction. By using similar reasoning to that in Lemma 2, we know that $V_{\vec{k}}(\bar{x})$ equals

$$\frac{1}{1-\delta} E_a \left[\max_{z_a} E_{\epsilon} \left[- \int_{\bar{x}_a + \sigma_{a,k_a \epsilon}}^{z_a} F_a(z_a) - F_a(y) dy + \delta F_a(z_a)(E_{\mathcal{F}'}(a)[V_{\vec{k}'(a,\vec{k})}(\bar{x}'(a))] - V_{\vec{k}}(\bar{x})) \right] \right],$$

where $\vec{k}'(a, \vec{k}) \equiv (k'_1, \dots, k'_m)$ is a vector that satisfies $k'_b = k_b$ for all $b \neq a$ and $k'_a = k_a + 1$, and $\bar{x}'(a)$ denotes the uncertain realization of \bar{x} if the advertiser-context pair a receives an additional impression. Furthermore, the optimal bid z_a if there is an auction involving the advertiser-context pair a satisfies $z_a = \bar{x}_a + \delta F_a(z_a)(E_{\mathcal{F}'}(a)[V_{\vec{k}'(a,\vec{k})}(\bar{x}'(a))] - V_{\vec{k}}(\bar{x}))$ by similar logic to that given in Theorem 3, and the result in Lemma 4 is just a general mathematical result that holds regardless of the model we are considering. Thus natural analogs of Theorems 1 and 3 and Lemmas 2 and 4 continue to hold in this revised model.

By using these insights, one can further show that $V_{\vec{k}}(\bar{x})$ must be on the order of $\frac{1}{k}$ for large k . This is done in the following theorem:

Theorem 9 *When there are multiple ads with correlated values, $V_{\vec{k}}(\bar{x}) = \Theta(\frac{1}{k})$.*

While this result indicates that $V_{\vec{k}}(\bar{x})$ varies with $\frac{1}{k}$ for large values of k , this alone does not guarantee the convergence of this function for large values of k . We verify that this function does indeed converge for large values of k in the following theorem:

Theorem 10 *When there are multiple ads with correlated values, $kV_{\vec{k}}(\bar{x})$ converges to a function of \bar{x} in the limit as $k \rightarrow \infty$. Furthermore, it must be the case that $kV_{\vec{k}}(\bar{x}) = -\frac{1}{2(1-\delta)} \sum_{a=1}^m \pi_a \frac{1}{\beta_a} s_a^2(\bar{x}_a) f_a(\bar{x}_a) + O(\frac{1}{k})$ for large k .*

Theorem 10 indicates that the results about the limiting value of $kV_k(\bar{x})$ derived in Theorem 6 extend naturally to the case where there are multiple ads with possibly correlated values. When there are multiple ad-context pairs that we must learn about, the value function corresponding to that in Theorem 6 differs only in that we take a weighted sum over the various possible advertiser-context pairs, where the weights are a function of the relative probabilities with which each advertiser-context pair arises. Thus there is a clear analog between the limiting properties of the value function when there are multiple advertiser-context pairs and the value function in the main model.

By using the results in the previous theorem, one can further derive properties of the limiting value of $E_{\bar{x}(a)}[V_{k(a,\bar{k})}(\bar{x}(a))] - V_k(\bar{x})$ that is proportional to the additional amount that one should bid in the auction beyond the expected value that one has for the advertising opportunity. This is stated below in the following theorem:

Theorem 11 *When there are multiple ads with correlated values, $E_{\bar{x}(a)}[V_{k(a,\bar{k})}(\bar{x}(a))] - V_k(\bar{x}) = -\frac{\sigma_{\bar{x}}}{2(1-\delta)\beta k^2} s_a^2(\bar{x}_a) f_a(\bar{x}_a) + o(\frac{1}{k^2})$ for large k .*

The proof of this result is substantively identical to the proof of Theorem 7 and is thus omitted. Theorem 11 illustrates that the substantive conclusions of Theorem 7 extend to this alternative environment in which there are multiple ads with possibly correlated values. When there are multiple ads, it remains optimal to increase one's bid by an amount proportional to the variances in our estimates of the eCPMs of the ads or $\frac{\sigma_{\bar{x}}}{k^2}$ for large k .

8. Knowledge Gradients

Throughout the paper so far, we have considered a standard dynamic programming approach in which the optimal decision at any given point in time is affected in part by how this decision will affect future decisions when looking at the infinite horizon ahead. While this is a standard approach to take in these types of problems, recently there has been work considering an alternative approach often referred to as "knowledge gradients" in which the decision one takes in a given period is the decision that one would take if one faced an infinite-horizon game but this period was the last period in which the information one learned could be used to inform future actions.

The main advantage of these knowledge gradients over the standard dynamic programming approach is that they have the virtue of being much easier to calculate than the optimal bidding strategy under the standard dynamic programming problem. This simplicity does potentially come at a performance cost. However, various papers have illustrated that using this simple one-step look-ahead approach can nonetheless achieve a performance that is competitive with that of other standard methods in contexts unrelated to advertising (Frazier et al., 2009; Ryzhov et al., 2010, 2012). In this section, we investigate whether this alternative knowledge gradient approach can indeed achieve a performance comparable to that of the theoretically optimal dynamic programming approach.

To address this question, for simplicity we consider the baseline model in which there is one advertisement for which we are seeking to learn the eCPM of the ad, though similar results can easily be derived under the more general model we have considered with multiple ads and correlated values. Let $U_k(\bar{x})$ denote the value that one would obtain for the rest of

the game when an ad has received k impressions so far, one's estimate of the eCPM of the ad is \bar{x} , and one will not be able to use information that one learns in the future to inform future bidding decisions. Note that in this case, the optimal bidding strategy will be to submit a bid of $z = \bar{x}$ in every remaining period, and the auctioneer's expected per-period payoff will be $E_\epsilon \left[-\int_{\bar{x}+\sigma_{k\epsilon}}^z F(z) - F(y) dy \right]$ in every future period, where ϵ denotes some random variable with mean zero and variance one, and $z \equiv \bar{x}$. The total value the auctioneer will obtain for the rest of the game is then $U_k(\bar{x}) = \frac{1}{1-\delta} E_\epsilon \left[-\int_{\bar{x}+\sigma_{k\epsilon}}^{\bar{x}} F(\bar{x}) - F(y) dy \right]$.

Also let $U_{k+1}(\bar{x})$ denote the value that one would obtain for the rest of the game when an ad has received $k+1$ impressions so far, one's estimate of the eCPM of the ad is \bar{x} , and one will not be able to use information that one learns in the future to inform future bidding decisions. The total value the auctioneer will obtain for the rest of the game is then $U_{k+1}(\bar{x}) = \frac{1}{1-\delta} E_\epsilon \left[-\int_{\bar{x}+\sigma_{k+1\epsilon}}^{\bar{x}} F(\bar{x}) - F(y) dy \right]$.

Now consider the bidding strategy that one would employ if one faced an infinite-horizon game but this period was the last period in which the information one learned could be used to inform future actions. The auctioneer's payoff from bidding z that arises in the current period equals $-\int_{\bar{x}+\sigma_{k\epsilon}}^z F(z) - F(y) dy$. And the expected value that the auctioneer obtains from future periods by bidding z in the current period is $F(z) E_{\bar{x}}[U_{k+1}(\bar{x})] + (1 - F(z)) U_k(\bar{x})$ by the same reasoning used in the proof of Lemma 2. From this it follows that the expected payoff from placing a bid of z in a given period is

$$E_\epsilon \left[-\int_{\bar{x}+\sigma_{k\epsilon}}^z F(z) - F(y) dy + \delta(F(z) E_{\bar{x}}[U_{k+1}(\bar{x})] + (1 - F(z)) U_k(\bar{x})) \right].$$

There is a clear similarity between this expression and the expression for the expected payoff from placing a bid of z in the standard dynamic programming approach. The main difference is that the terms $U_{k+1}(\bar{x})$ and $U_k(\bar{x})$ have replaced the terms $V_{k+1}(\bar{x})$ and $V_k(\bar{x})$ in the standard dynamic programming approach.

It is worth noting, however, that the payoffs that result from the one-step look ahead strategies in the model in this paper take a different form than those given in other knowledge gradient papers (Frazier et al., 2009; Ryzhov et al., 2010, 2012). The reason for this difference is that in the model in our paper, there is a competing ad whose eCPM is known in each period but is also a random draw from some distribution in each period. No such random changes in the values of the arms from period to period are present in existing knowledge gradient papers, so the payoffs and strategies in our paper are formulated differently than those given in existing knowledge gradient papers.

From the equation we've derived for the auctioneer's payoff from bidding z , we can calculate the optimal bidding strategy under the knowledge gradient formulation. This bidding strategy is given in the following theorem:

Theorem 12 *The optimal bidding strategy in the knowledge gradient framework when an ad has been shown k times entails setting $z = \bar{x} + \delta(E_{\bar{x}}[U_{k+1}(\bar{x})] - U_k(\bar{x}))$.*

The proof of this result is substantively identical to that in Theorem 3 and is thus omitted. This result indicates that in the knowledge gradient framework, the incremental amount that one increases one's bid beyond the immediate expected reward is again of the

form $\delta[E_{\bar{x}}[U_{k+1}(\bar{x}')] - U_k(\bar{x})]$, the only difference being that $U_k(\bar{x})$ corresponds to the value of the dynamic program under the knowledge gradient framework.

To better understand the incremental amount that one would increase one's bid, we present two results that illustrate how the incremental amount that one would increase one's bid under the knowledge gradient framework compares to the incremental amount that one would increase one's bid under the full dynamic programming problem. First we present a finite sample result about how these incremental bid increases compare in the two frameworks.

In our first result, we consider what we refer to as the expected value of all future learning. To reflect the fact that $V_k(\bar{x})$ gives the auctioneer's payoff from the full dynamic programming problem when the auctioneer is able to make use of additional information in future periods, whereas $U_k(\bar{x})$ gives the auctioneer's payoff from the corresponding game in which the auctioneer is not able to make use of information that he learns, we define this expected value of all future learning term to be the difference between $V_k(\bar{x})$ and $U_k(\bar{x})$. With this definition in mind, we obtain the following result:

Theorem 13 *Suppose the expected value of all future learning is lower after the ad has been shown $k + 1$ times than it is after the ad has been shown k times. Then the incremental amount by which one would increase one's bid under the knowledge gradient framework is greater than it is under the full dynamic programming problem.*

Theorem 13 indicates that the solution to the one-step look ahead problem will generally involve increasing one's bid beyond the immediate expected value of the advertising opportunity by a greater amount than one would do so under the full dynamic programming problem. This makes sense intuitively. If the current period were the last period in which one could ever use information that one learns to inform future actions, then one would place quite a high premium on being able to learn this information while one still can. By contrast, in the full dynamic programming problem, there will always be plenty of opportunities to learn this information later, so there is relatively less incentive to substantially increase one's bid beyond the immediate expected reward. This explains the result in Theorem 13.

Theorem 13 requires a technical condition that the expected value of all future learning is lower if an ad has been shown $k + 1$ times than if the ad has been shown k times, but this is just a mild technical constraint that we would expect to hold in virtually any situation. When an ad has been shown $k + 1$ times, one has more precise information about the true eCPM of the ad than when the ad has only been shown k times, so there is less value to learning more about the true eCPM of the ad.

While Theorem 13 suggests that one might increase one's bid by too much under the knowledge gradient framework compared to the strategy that one should follow under the full dynamic programming problem, these differences in bidding strategies turn out to be relatively small. We illustrate this by characterizing the value of $E_{\bar{x}}[U_{k+1}(\bar{x}')] - U_k(\bar{x})$:

Theorem 14 *In the knowledge gradient framework, $E_{\bar{x}}[U_{k+1}(\bar{x}')] - U_k(\bar{x}) = \frac{v(\bar{x})}{k(k+1)} + o\left(\frac{1}{k^2}\right)$ for large k , where $v(\bar{x}) \equiv \frac{1}{2(1-\beta)} s^2(\bar{x}) f(\bar{x})$.*

This result also immediately implies the following corollary:

Corollary 15 *The ratio of the incremental amount by which one wants to increase one's bid in the knowledge gradient framework and the incremental amount by which one wants to increase one's bid in the standard dynamic programming approach becomes arbitrarily close to 1 in the limit as the amount of uncertainty in an ad's eCPM becomes arbitrarily small.*

Thus using the knowledge gradient to formulate one's bidding strategy will result in payoffs that are asymptotically equivalent to those that would result from using the theoretically optimal bidding strategy. These results suggest that the knowledge gradient framework is indeed an appealing framework for computing bidding strategies in an environment where one wishes to learn about the unknown eCPMs of advertisers, as using this approach will result in little loss from the theoretically optimal approach.

9. Learning About Multiple Advertisers in the Same Auction

In the analysis so far, we have assumed that in any given auction, there is only one advertiser whose eCPM is unknown. But in many real-life auctions there may be multiple advertisers with unknown eCPMs. In these cases, an auctioneer must decide both which advertiser with unknown eCPM will have the highest bid as well as what bid to submit for this advertiser.

In this setting, it is not clear whether the decision maker's optimal strategy can simply be represented by submitting a bid for each advertiser that is equal to the sum of the best estimate of the advertiser's eCPM as well as a value of learning term. It may be the case that the optimal bid for advertiser i if advertiser i submits the highest bid of the advertisers with unknown eCPMs is higher than the optimal bid for some other advertiser j if advertiser j submits the highest bid of the advertisers with unknown eCPMs, even though the decision maker would prefer to submit a higher bid for advertiser j than for advertiser i . We address whether this possibility can arise in this section.

To address this question, suppose that in each auction, there are n ads with unknown eCPMs. The actual eCPMs of these ads are x_1, \dots, x_n , and we let z_1, \dots, z_n denote the bids placed by these advertisers in the auction. Also let i denote the advertiser who submits the highest eCPM bid amongst these n bidders and let j denote the advertiser who actually has the highest eCPM amongst these advertisers. In each auction, these advertisers with unknown eCPMs compete against other advertisers and the highest such competing eCPM is drawn from a cumulative distribution function $F(\cdot)$ with corresponding density $f(\cdot)$.

Note that in this case, the utility that the decision maker obtains in a given period from having advertiser i submit a bid of z_i that is the highest eCPM bid amongst these n bidders is $u = z_i(1 - F(z_i)) + \int_{z_i}^{\infty} (1 - F(y)) dy + x_i F(z_i)$. At the same time, this decision maker would obtain a utility of $u = x_j(1 - F(x_j)) + \int_{x_j}^{\infty} (1 - F(y)) dy + x_j F(x_j)$ in a given period from making the optimal decision in a given period. Thus the loss that this decision maker obtains in a given period as a result of having advertiser i submit a bid of z_i that is the highest eCPM bid amongst these n bidders is the difference between these two utilities or $L = x_j - z_i + (z_i - x_j)F(z_i) + \int_{x_j}^{z_i} (1 - F(y)) dy = \int_{x_j}^{z_i} F(z_i) dy - \int_{x_j}^{z_i} F(y) dy$.

Now let k_i denote the number of impressions that advertiser i has received so far, let $\vec{k} \equiv (k_1, \dots, k_n)$ denote a vector that gives the number of times each of these ads has been shown, let \bar{x}_i denote our best estimate of the expected value of the eCPM of advertiser i ,

and let $\bar{x} \equiv (\bar{x}_1, \dots, \bar{x}_n)$ denote a vector of these best estimates. Also let $V_k^c(\bar{x})$ denote the value of the dynamic program as a function of these quantities.

By similar reasoning to that in the proof of Lemma 2, it follows that if i denotes the advertiser who submits the highest eCPM bid amongst the n bidders with unknown eCPMs, $R^c(i) \equiv (k_1^i, \dots, k_n^i)$ is the vector where $k_j^i = k_j$ for all $j \neq i$ and $k_i^i = k_i + 1$, and $\bar{x}^c(i)$ denotes the uncertain realization of \bar{x} if advertiser i receives an additional impression, then $V_k^c(\bar{x})$ equals

$$\frac{1}{1-\delta} \left(\max_{z_i} E_{R_i, \bar{x}_i} \left[\int_{\bar{x}_i}^{z_i} F(y) dy - \int_{\bar{x}_i}^{z_i} F(z_i) dy + \delta F(z_i) (E_{R^c(i)}[V_k^c(\bar{x}^c(i))] - V_k^c(\bar{x})) \right] \right),$$

where the difference in the values of the dynamic programs is due to the fact that the loss in a given period is now $\int_{\bar{x}_i}^{z_i} F(z_i) dy - \int_{\bar{x}_i}^{z_i} F(y) dy$. Similarly, the optimal bid for advertiser i if advertiser i submits the highest eCPM bid amongst the n bidders with unknown eCPMs still satisfies $z_i = \bar{x}_i + \delta (E_{R^c(i)}[V_k^c(\bar{x}^c(i))] - V_k^c(\bar{x}))$. We use these insights to prove the following:

Theorem 16 *Suppose the optimal bid for advertiser i if advertiser i submits the highest bid of the advertisers with unknown eCPMs is higher than the optimal bid for all other advertisers with unknown eCPMs if one of these other advertisers submits the highest bid of the advertisers with unknown eCPMs. Then it is also optimal for advertiser i to have the highest bid of all the advertisers with unknown eCPMs.*

Theorem 16 guarantees that if there are multiple ads with unknown eCPMs, then one can simply compute the optimal bids for each of these ads in the case where the ad in question was guaranteed to have a higher bid than the other ads with unknown eCPMs. The ad that has the highest such optimal bid will then be guaranteed to be the ad for which the mechanism designer would want to submit the highest such bid. Thus even when there are multiple ads in the same auction with unknown eCPMs, one can continue to make optimal decisions by computing bids for the advertisers equal to their estimated eCPMs plus a value of learning term for the ad and then rank the advertisers on this basis.

10. Performance Guarantees

We now return to the baseline setting in Section 2. The results in the previous sections suggest a possible algorithm that will approximate the optimal bidding strategies for an auctioneer who seeks to maximize long-run efficiency. This algorithm would compute the expected eCPM for an advertiser with unknown eCPM, \bar{x} , the density for the distribution of competing eCPM bids at this value of \bar{x} , $f(\bar{x})$, the variance $s^2(\bar{x})$ in the eCPM for an ad with estimated eCPM \bar{x} that has only received one impression, and the number of impressions k that the ad has received. One then decides which ad to show by computing a score equal to $\bar{x} + \frac{\delta}{2(1-\delta)^{k/(k+1)}} s^2(\bar{x}) f(\bar{x})$ for each ad, where δ is the auctioneer's discount factor, and showing the ad with the highest such score. We refer to this strategy as the *approximately optimal bidding strategy*, and in this section we address questions related to the size of the performance guarantees that can be obtained by using this algorithm and related algorithms.

First we address questions related to how the algorithms we have considered in this paper will compare to other plausible algorithms in the machine learning literature. One other algorithm that is standard for multi-armed bandit problems involves ranking the arms by a term equal to the expected value of the arm plus a term proportional to the standard deviation in the arm (Auer et al., 2002). More generally, one can rank advertisers by a term equal to the eCPM of the advertiser plus a term proportional to $\frac{1}{k^\alpha}$ for any $\alpha \leq \frac{1}{2}$, where k denotes the number of impressions that the ad has received so far. However, these algorithms are not well-suited towards the auction environment, as the following theorem illustrates:

Theorem 17 *Suppose the auctioneer uses a bid for the advertiser with unknown eCPM of the form $z = \bar{x} + \frac{c(\bar{x})}{k^\alpha}$, where $\alpha \leq \frac{1}{2}$ and $c(\bar{x})$ is a bounded non-negative constant that depends only on \bar{x} and the distribution of competing bids. Then the optimal constant $c(\bar{x})$ for any such algorithm is $c(\bar{x}) = 0$ for sufficiently large k .*

This result immediately implies that standard existing algorithms for exploration which involve adding a term proportional to the standard deviation to the eCPM of the ad, such as the UCB algorithm, are actually dominated by the simple greedy approach of always making a bid equal to the eCPM of the ad. These existing algorithms do too much exploration, and as a result, lead to lower payoffs than not doing any active exploration at all.⁶

Next we turn to the question of what guarantees can be made about the size of the performance improvement that could be obtained by using the approximately optimal bidding strategy rather than the simple greedy algorithm. Our next result illustrates that one will indeed obtain a performance improvement by using the approximately optimal bidding strategy, but the size of the performance improvement is likely to be very small.

Theorem 18 *Suppose the auctioneer follows the approximately optimal bidding strategy. Then the expected payoff that the auctioneer will obtain by using this algorithm will exceed the expected payoff that the auctioneer would obtain by using the purely greedy approach by an amount $\frac{\delta^2}{8(1-\delta)^{k/\alpha}} s^4(\bar{x}) f^3(\bar{x}) + o(\frac{1}{k^\alpha})$.⁷*

Theorem 18 indicates that the performance improvement that can be obtained as a result of using the approximately optimal bidding strategy is only on the order of $\frac{1}{k^\alpha}$, where k denotes the number of impressions that an ad has received. This follows from the fact that the incremental increase in the probability that a particular ad is shown varies with $\frac{1}{k^\alpha}$, and on top of that, the expected payoff increase that one obtains conditional on showing a different ad than would be shown without active learning also varies with $\frac{1}{k^\alpha}$. Since this represents a fourth-order improvement in performance relative to the purely greedy approach, this result indicates that the performance improvement that can be obtained by following our algorithm rather than simply ranking the ads by their eCPMs is small.

6. Similarly, an algorithm such as epsilon-greedy, in which the ad with the highest eCPM is chosen with probability $1 - \epsilon$, and an ad is chosen uniformly at random with probability ϵ , will also lead to lower payoffs than not doing any active exploration at all for large k . We prove this in Observation 23 in the appendix.

7. The expected payoff increase that we refer to in this theorem is for the subgame beginning from the point when the ad with uncertain eCPM has already received k impressions.

It is worth noting, however, that the result in Theorem 18 is not due to our algorithm being a suboptimal implementation of incorporating active exploration. Our next result illustrates that while the size of the performance improvement that can be obtained by using our algorithm is small, this algorithm will, in fact, obtain nearly the maximum possible performance improvement over the purely greedy approach of ranking ads by their eCPMs.

Theorem 19 *Suppose the auctioneer uses the approximately optimal bidding strategy. Then the difference between the auctioneer’s payoff under this strategy and the maximum possible payoff the auctioneer could obtain under the theoretically optimal strategy becomes vanishingly small compared to the difference between the auctioneer’s payoff under this strategy and the auctioneer’s payoff under the greedy strategy for large k .*

The results in the previous theorems suggest that the maximum possible payoff increase that can be achieved by incorporating active exploration is quite small for auctions involving ads that have already received a large number of impressions. However, in many auctions, there are frequently advertisers that have only received a small number of impressions, so it is desirable to know whether these conclusions for ads that have received large numbers of impressions will also hold for ads that have only received a small number of impressions. Under the mild technical condition discussed in Theorem 13, where the expected value of future learning is lower after the advertiser with unknown eCPM has been shown once rather than never having been shown at all, we obtain the following result:

Theorem 20 *Suppose the bidder with unknown eCPM has a cost-per-click bid of 1 and a click-through rate drawn from a beta distribution. Also suppose that this bidder’s expected eCPM is ω and the standard deviation in this bidder’s true eCPM is $\gamma\omega$. Then the difference between the maximum possible payoff the auctioneer could obtain under the theoretically optimal strategy and the auctioneer’s payoff from the greedy strategy is no greater than $\frac{\delta^2 \gamma^2 \omega^2 \bar{f}^3}{8(1-\delta)^3(1-\omega)^2}$, where \bar{f} denotes the supremum of $f(\cdot)$.*

Theorem 20 presents bounds on the maximum performance improvement that can be achieved over the purely greedy strategy by using active learning, but it is not immediately clear from this result whether these bounds imply there are significant limitations on the performance improvement that can be achieved by using active learning. We thus seek to shed some light on this under empirically realistic values of the parameters.

If the typical eCPM bids for the winning advertisers are roughly $\xi\omega$, then the auctioneer’s total payoff for the game will be roughly $\frac{\xi\omega}{1-\delta}$, and the result in Theorem 20 indicates that the maximum fractional increase in expected payoff that one can achieve from using the theoretically optimal strategy rather than the greedy strategy is roughly $\frac{\delta^2 \gamma^2 \omega^2 \bar{f}^3}{8\xi(1-\delta)^2(1-\omega)^2}$.

Furthermore, if the typical eCPM bids for the highest competing advertisers in an auction are roughly $\xi\omega$, then \bar{f} is likely to also be on the order of $\frac{1}{\xi\omega}$. This holds, for example, if the highest competing eCPM bids are drawn from a lognormal distribution, as the largest value of the density of a lognormal distribution with parameters μ and σ^2 is equal to $\frac{c(\sigma^2)}{\xi\omega}$, where $\xi\omega$ is the expected value of the lognormal distribution and $c(\sigma^2) \equiv \frac{e^{-\sigma^2/2}}{\sqrt{2\pi\sigma^2}}$ is a constant that depends only on σ^2 . Furthermore $c(\sigma^2)$ is likely to be close to 1 for realistic values of σ^2 since $c(\sigma^2) \in [0.93, 1.09]$ for values of $\sigma^2 \in [0.2, 1.1]$. The lognormal distribution

is a realistic representation of the distribution of highest competing bids in online auctions since both Lahate and McAfee (2011) and Ostrovsky and Schwarz (2009) have noted that the distribution of highest bids can be well-represented by a lognormal distribution using data from sponsored search auctions at Yahoo!.

By using the facts that the value of \bar{f} is likely to be on the order of $\frac{1}{\xi\omega}$, and the maximum fractional increase in expected payoff that one can achieve from using the theoretically optimal strategy rather than the greedy strategy is roughly $\frac{\delta^2 \gamma^2 \omega^2 \bar{f}^3}{8\xi(1-\delta)^2(1-\omega)^2}$, it then follows that the maximum fractional increase in expected payoff that one can achieve from using the theoretically optimal strategy rather than the greedy strategy is roughly $\frac{\delta^2 \gamma^2 \omega^2}{8\xi(1-\delta)^2(1-\omega)^2}$.

There is empirical evidence that indicates that the typical click-through rates for ads in online auctions tend to be on the order of $\frac{1}{100}$ or $\frac{1}{1000}$ for search ads and display ads respectively (Bax et al., 2011), so $(1-\omega)^2$ will be very close to 1 and ω^2 is likely to be less than 10^{-4} (for search ads) or 10^{-6} (for display ads). Furthermore, even for a brand new ad, the typical errors in a machine learning system’s predictions are unlikely to exceed 30% of the true click-through rate of the ad, so $\gamma \leq 0.3$ is likely to hold in most practical applications. Finally, ξ is a measure of by how much the highest bid in an auction exceeds the typical eCPM bid of an average ad in the auction. Since there are normally hundreds of ads competing in online auctions, it seems that one can conservatively estimate that $\xi \geq 3$ is likely to hold in most real-world online auctions.

By combining the estimates in the previous paragraph, it follows that $\frac{\delta^2 \gamma^2 \omega^2}{8\xi(1-\omega)^2}$ will almost certainly be less than 10^{-11} in search auctions and 10^{-13} in display auctions. Now if $\delta \leq 0.9999$, $\frac{\delta^2}{(1-\delta)^2}$ will be no greater than 10^8 , and if $\delta \leq 0.99999$, $\frac{\delta^2}{(1-\delta)^2}$ will be no greater than 10^{10} . Thus even for values of δ that are exceedingly close to 1 ($\delta = 0.9999$ for search ads and $\delta = 0.99999$ for display ads), $\frac{\delta^2 \gamma^2 \omega^2}{8\xi(1-\omega)^2(1-\delta)^2}$ will be no greater than 0.001. Thus as long as $\delta \leq 0.9999$ (or $\delta \leq 0.99999$ for display auctions), the bound given in Theorem 20 guarantees that under empirically realistic scenarios, the maximum possible performance improvement that can be achieved by incorporating active learning into a machine learning system is at most a few hundredths of a percentage point. This is a finite sample result that does not require a diverging number of impressions in order to hold.

11. Simulations

The results of the previous section suggest that the overall benefit that can be obtained by incorporating active exploration in an auction environment is exceedingly small. We now seek to empirically verify that the benefit that can be obtained from active exploration is indeed quite small by conducting simulations under some empirically realistic scenarios.

To do this, we consider a scenario in which there is a repeated auction in which a cost-per-click (CPC) bidder competes against CPM bidders in each auction. The CPC bidder has a CPC bid of 1 and a fixed unknown click-through rate. The CPM bidders’ CPM bids vary from period to period, and in each period, we assume that the highest CPM bid is a random draw from a distribution with probability density function $f(\cdot)$. Throughout we assume that payoffs are discounted at a rate of $\delta = 0.9995$ and that there are $T = 10000$ time periods.

While we are not aware of any empirical evidence regarding the form of the uncertainty of an advertiser’s click-through rate, for simplicity we assume that the CPC bidder’s click-through rate is initially drawn from a beta distribution with parameters α and β . The auctioneer may refine this estimate over time. In particular, just before the auction in period t , the auctioneer believes that the CPC bidder’s true click-through rate is a random draw from the beta distribution with parameters α_t and β_t where α_t is equal to α plus the number of clicks the CPC bidder has received so far and β_t is equal to β plus the number of times the CPC bidder’s ad was shown but did not receive a click.

We compare total welfare under two possible scenarios. The first scenario we consider is a standard ranking algorithm in which the ads are ranked purely on the basis of their expected eCPM bids. The second scenario we consider is one in which the CPC bidder makes a bid of the form $\bar{x}_t + \frac{\delta(1-\delta^{T-t})}{2(1-\delta)} \frac{\alpha_t/\beta_t}{(\alpha_t+\beta_t)^2(\alpha_t+\beta_t+1)^2} f(\bar{x}_t)$ in each period t , where \bar{x}_t denotes the CPC bidder’s expected click-through rate just before the auction in period t . This second scenario corresponds to adding a term equal to the value of learning to the CPC bidder’s expected eCPM bid in the game with finite time horizons.

Throughout we focus on scenarios that are motivated by empirical evidence on the likely expected click-through rates for ads in online auctions. In particular, since empirical evidence indicates that the typical click-through rates for ads in online auctions tend to be on the order of $\frac{1}{100}$ or $\frac{1}{1000}$ (Bax et al., 2011), we focus on situations in which the expected click-through rate of the CPC bidder is on the order of $\frac{1}{100}$.

Similarly, since it is unlikely that there will be substantial errors in the estimate of a new ad’s predicted click-through rate, we focus on situations in which there is only moderate uncertainty in the click-through rate of a new ad. In particular, we consider distributions of the CPC bidder’s bid such that the standard deviation in the advertiser’s click-through rate is no greater than 20 or 30% of the expected value. We thus consider values of α and β satisfying $(\alpha, \beta) = (10, 1000)$ and $(20, 2000)$ (for 30% and 20% standard errors respectively).

Finally, since there is evidence that the distribution of highest bids is well modeled by a lognormal distribution (Lahaie and McAfee, 2011; Ostrovsky and Schwarz, 2009), we assume throughout that the CPM bidder’s bid is drawn from a lognormal distribution with parameters μ and σ^2 . We use a value of $\sigma^2 = \log(2)$ to match the variance in the lognormal distribution estimated by Ostrovsky and Schwarz (2009). And Varian (2009) has noted that the total value enjoyed by advertisers is typically about 2 – 2.3 times their total expenditure. If the auction consisted of only two advertisers, this would suggest that the appropriate value of μ would be such that the highest competing bidder had a CPM bid that is roughly double that of the CPC bidder in expectation. However, since there are more than two bidders in most real auctions, the appropriate value of μ will be larger than this. We thus consider a range of values of μ from -4.25 (for the case in which the highest competing CPM bid is roughly double that of the CPC bidder in expectation) to -3.5 (for the case in which the highest competing CPM bid is roughly four times that of the CPC bidder in expectation).

Table 1 reports the results our simulations. The conclusions from these simulations are striking. While we have conducted enough simulations to estimate the efficiency gain that can be obtained from adding active exploration to within a few hundredths of a percentage point, none of the resulting estimated efficiency gains in Table 1 are statistically significant. Indeed one can conclude from these simulations that the maximum possible efficiency gain

Conditions	Percentage increase in efficiency
$\alpha = 10, \beta = 1000, \mu = -4.25, \sigma^2 = \log(2)$	0.021% (0.017%)
$\alpha = 10, \beta = 1000, \mu = -4, \sigma^2 = \log(2)$	-0.016% (0.011%)
$\alpha = 10, \beta = 1000, \mu = -3.75, \sigma^2 = \log(2)$	-0.008% (0.007%)
$\alpha = 10, \beta = 1000, \mu = -3.5, \sigma^2 = \log(2)$	0.003% (0.004%)
$\alpha = 20, \beta = 2000, \mu = -4.25, \sigma^2 = \log(2)$	0.003% (0.009%)
$\alpha = 20, \beta = 2000, \mu = -4, \sigma^2 = \log(2)$	0.001% (0.006%)
$\alpha = 20, \beta = 2000, \mu = -3.75, \sigma^2 = \log(2)$	0.001% (0.004%)
$\alpha = 20, \beta = 2000, \mu = -3.5, \sigma^2 = \log(2)$	-0.002% (0.002%)

Table 1: Average percentage increase in efficiency from incorporating active learning (with standard errors in parentheses) after 2500 simulations. None of these results are statistically significant at the $p < .05$ level.

that could be achieved in these settings is at most a few hundredths of a percentage point. These empirical results provide further support for our theoretical conclusions that the value of adding active exploration in an auction setting is exceedingly small.

The reason for the results observed in Table 1 is that an optimal exploration algorithm will only do a tiny additional amount of exploration compared to the greedy strategy of always submitting a bid for the CPC bidder equal to the CPC bidder’s estimated eCPM. For instance, for the first simulation considered in Table 1, the incremental increase in an advertiser’s bid in the first period of the game as a result of active exploration is only 4.2%, implying only a 1.8% increase in the probability that the CPC bidder will be shown as well as only a 2.1% increase in expected payoff conditional on the auctioneer showing a different ad under active exploration than under the purely greedy strategy. Thus the incremental expected payoff increase that can be achieved by incorporating active exploration in this auction setting is at most a few hundredths of a percentage point.

The results in Table 1 make use of distributions that we regard as empirically realistic in the sense that there is a realistic amount of uncertainty in the click-through rate of the CPC bidder as well as a realistic amount of variation in the distribution of competing CPM bids. It is worth noting that if one relaxes the requirement that there be a realistic amount of uncertainty about these variances, then it is possible for the algorithm we have proposed to substantially outperform the purely greedy strategy of making a bid for the CPC bidder that always equals the CPC bidder’s expected eCPM. In particular, if we instead assume that there is substantially more uncertainty about the CPC bidder’s click-through rate than

we have assumed in the simulations in Table 1 and we also assume that there is substantially less variance in the distribution of competing CPM bids than we have allowed for in Table 1, then there will be considerably greater benefits to adding active exploration because there is both more to learn about the CPC bidder's true eCPM bid as well as less exploration that will take place for free solely due to random variation in the competing bids. In this case, there may well be significant benefits to adding active exploration.

Conditions	Percentage increase in efficiency
$\alpha = 2, \beta = 200, \mu = -4, \sigma^2 = \log(2)/4$	0.15% (0.05%)
$\alpha = 2, \beta = 200, \mu = -3.75, \sigma^2 = \log(2)/4$	0.17% (0.05%)

Table 2: Average percentage increase in efficiency from incorporating active learning (with standard errors in parentheses) after 10000 simulations. These results are both statistically significant at the $p < .005$ level.

Table 2 reports the results of simulations that were conducted using distributions in which there is substantially more uncertainty about the CPC bidder's click-through rate and substantially less variance in the CPM bidder's competing CPM bid than in the distributions considered in Table 1. These simulations indeed reveal statistically significant efficiency gains as a result of active exploration. Nonetheless it is worth noting that the efficiency gains reported in Table 2 are still fairly small. Even when we make assumptions that bias the case in favor of active exploration being important, none of the efficiency gains reported in Table 2 are greater than a few tenths of a percentage point.

Finally, while the gains achieved through active exploration in Table 2 are small, one would not achieve greater gains by using a standard algorithm such as UCB. To test this, we considered the same setting in the first row of this table, but instead of making a bid for the CPC bidder of the form $\bar{x}_t + \frac{\delta(1-\delta^{T-t})}{2(1-\delta)} \frac{\alpha t \beta_t}{(\alpha + \beta_t)^2 (\alpha + \beta_t + 1)^2} f(\bar{x}_t)$ in each period t , we made a bid of the form $\bar{x}_t + c(\bar{x}_t) \frac{1}{\sqrt{\alpha + \beta_t}}$, where the constant $c(\bar{x}_t)$ was chosen so that this bid would equal $\bar{x}_t + \frac{\delta(1-\delta^{T-t})}{2(1-\delta)} \frac{\alpha t \beta_t}{(\alpha + \beta_t)^2 (\alpha + \beta_t + 1)^2} f(\bar{x}_t)$ in time period $t = 1$. Thus our implementation of UCB performed the same amount of exploration as the main algorithm we considered in the very first period of the game, while performing more exploration in later periods due to the fact that the rate of exploration declines with $\frac{1}{(\alpha + \beta_t)^2}$ under our proposed algorithm, while only declining with $\frac{1}{\sqrt{\alpha + \beta_t}}$ under UCB.

In this setting, we found that using the UCB algorithm rather than the purely greedy strategy resulted in an average efficiency loss of 1.04% (with a standard error of 0.07%). Thus while we were able to achieve an improvement by using the new algorithm we have proposed, using the UCB algorithm instead resulted in significant efficiency losses. The fact that UCB performed worse than the purely greedy strategy is not surprising since we know from Theorem 17 that UCB performs worse than the purely greedy strategy once an ad has received enough impressions.

12. Conclusion

In online auctions, there may be value to exploring ads with uncertain eCPMs to learn about the true eCPM of the ad and be able to make better ranking decisions in the future. But the online auction setting is very different from standard multi-armed bandit problems because there may be considerable variation in the quality of competition that an advertiser with unknown eCPM faces in an auction, and as a result there will typically be plenty of free opportunities to explore an ad with uncertain eCPM in auctions where there simply are no ads with eCPM bids that are known to be high.

We have presented a model of the explore/exploit problem in online auctions that explicitly considers this random variation in competing bids that is present in real auctions. We find that the optimal solution for ranking the ads is dramatically different than the optimal solution in standard multi-armed bandit problems, and in particular, that the optimal amount of active exploration is considerably smaller than in standard multi-armed bandit problems. This in turn implies that the improvement in the auctioneer's payoff that can be achieved by adding active learning in online auctions is also exceedingly small. Thus while it is theoretically possible to improve efficiency by incorporating active learning, in a practical exchange environment, a purely greedy strategy of simply ranking the ads by their expected eCPMs is likely to perform nearly as well as any other strategy.

We conclude by discussing one other point. Throughout our analysis we have focused on the problem of an auctioneer who wants to maximize efficiency. Although this is a sensible objective, one might also envision scenarios in which the mechanism designer wishes to maximize a weighted average of efficiency and revenue. While incorporating active exploration in online auctions can only have a small effect on efficiency, this active exploration may significantly improve revenue. The reason for this is that if we rank the ads by the sum of their expected eCPMs and a value of learning term, the value of learning term may be larger for ads that typically lose the auctions, and incorporating this value of learning term may increase pricing pressure for the winning ads and thereby increase revenue.⁸ In fact, in several of the simulations considered in the previous section in which incorporating active exploration failed to show significant efficiency gains, the algorithm that we considered still showed significant revenue gains over the purely greedy strategy of ranking the ads by their expected eCPMs. But while it is still possible to achieve significant revenue gains by incorporating active exploration in the type of environment considered in this paper, the maximum possible efficiency gains are likely to be exceedingly small.

Acknowledgments

We especially thank Martin Zinkevich for numerous helpful discussions. We are also grateful to Joshua Dillon, Pierre Griaupan, Chris Harris, Tim Lipus, Mohammad Mahdian, Hal Varian, and the anonymous referees for helpful comments and discussions.

This work is based on an earlier work: "Machine Learning in an Auction Environment", in *Proceedings of the 23rd International Conference on the World Wide Web (WWW)* (2014) ©ACM, 2014. <http://doi.acm.org/10.1145/2566486.2567974>.

8. A similar point has been previously noted by Li et al. (2010) and McAfee (2011).

Appendix A. Proofs of Theorems

Proof of Theorem 1: Suppose it is known that the eCPM of the ad is x . If the highest eCPM for a competing ad is p , then the presence of this ad with eCPM x increases total welfare by $x - p$ if $x > p$ and 0 otherwise. Thus the expected increase in total welfare from this ad with eCPM of x competing in the auction is $\int_0^x (x - p) f(p) dp$. The total long-term value from having this advertisement is then the discounted sum of this expected total increase in welfare or $\frac{1}{1-\delta} \int_0^x (x - p) f(p) dp$.

Now if $V(x) \equiv \frac{1}{1-\delta} \int_0^x (x-p) f(p) dp$, then $V'(x) = \frac{1}{1-\delta} \int_0^x f(p) dp$ and $V''(x) = \frac{1}{1-\delta} f(x)$. From this it follows that $V''(x) \geq 0$ for all x and $V''(x) > 0$ if x is contained in the support of F . Thus the long-term value of the advertisement is a convex function of the eCPM of the ad and a strictly convex function if the eCPM of the ad is contained within the support of the distribution of the highest competing eCPM. ■

Proof of Lemma 2: Suppose an ad has been shown k times. The value of the dynamic program that arises from placing the optimal bid z in the current period, $V_k(\bar{x})$, equals the immediate reward from bidding z (or the negative of the loss function) in the current period plus δ times the expected value of the dynamic program that arises in the next period.

Now if the new advertiser places a bid of z , then the probability the advertiser wins the auction is $F(z)$, in which case the expected value of the dynamic program that arises next period is $E_{\bar{x}'}[V_{k+1}(\bar{x}')]]$, where the expectation is taken over the randomness in the changes in the estimates of the eCPM of the ad \bar{x}' that arise as a result of showing this ad. The probability the advertiser does not win the auction is $1 - F(z)$, in which case the value of the dynamic program remains at $V_k(\bar{x})$. Thus the expected value of the dynamic program that arises in the next period is $F(z)E_{\bar{x}'}[V_{k+1}(\bar{x}')] + (1 - F(z))V_k(\bar{x})$.

At the same time, we have already seen that the reward from bidding z that arises in the current period equals $-\int_{\bar{x}+z}^z E_{\bar{x}'}[V_{k+1}(\bar{x}')] - F(y) dy$. By combining this with the insights in the previous paragraphs, it follows that

$$V_k(\bar{x}) = \max_z E_{\bar{x}'} \left[-\int_{\bar{x}+z}^z F(z) - F(y) dy + \delta(F(z)E_{\bar{x}'}[V_{k+1}(\bar{x}')] + (1 - F(z))V_k(\bar{x})) \right].$$

By subtracting $\delta V_k(\bar{x})$ from both sides and dividing by $1 - \delta$, it follows that

$$V_k(\bar{x}) = \frac{1}{1-\delta} \left(\max_z E_{\bar{x}'} \left[-\int_{\bar{x}+z}^z F(z) - F(y) dy + \delta F(z)(E_{\bar{x}'}[V_{k+1}(\bar{x}')] - V_k(\bar{x})) \right] \right). \quad \blacksquare$$

Proof of Theorem 3: By differentiating the expression in Lemma 2 with respect to z , we see that the first order condition for z to be an optimal bid is

$$\begin{aligned} 0 &= E_{\bar{x}'} \left[-\int_{\bar{x}+z}^z f(z) dy + \delta f(z)(E_{\bar{x}'}[V_{k+1}(\bar{x}')] - V_k(\bar{x})) \right] \\ &= E_{\bar{x}'} \left[-f(z)(z - \bar{x} - \sigma_k \epsilon) + \delta f(z)(E_{\bar{x}'}[V_{k+1}(\bar{x}')] - V_k(\bar{x})) \right] \\ &= f(z)(\bar{x} - z + \delta(E_{\bar{x}'}[V_{k+1}(\bar{x}')] - V_k(\bar{x}))) \end{aligned}$$

From this it follows that $z = \bar{x} + \delta(E_{\bar{x}'}[V_{k+1}(\bar{x}')] - V_k(\bar{x}))$ satisfies the first order conditions. Moreover, at this value of z , the second order conditions are also satisfied. Thus optimal bidding entails setting $z = \bar{x} + \delta(E_{\bar{x}'}[V_{k+1}(\bar{x}')] - V_k(\bar{x}))$. ■

Proof of Lemma 4: Let $\Phi(\cdot|\sigma_k)$ denote the distribution from which ϵ is drawn for any given value of σ_k . For any given σ_k , we know that $\Phi(\cdot|\sigma_k)$ has mean zero and variance one. We also know from the Bayesian central limit theorem that as $\sigma_k \rightarrow 0$ (and $k \rightarrow \infty$) that $\Phi(\cdot|\sigma_k)$ converges to the standard normal distribution. For any given σ_k , we can write $E_{\epsilon}[\int_{\bar{x}+\sigma_k \epsilon}^z F(z) - F(y) dy] = E_{\epsilon}[\int_{\bar{x}+\sigma_k \epsilon}^z F(z) - F(y) dy | \epsilon \sim \Phi(\cdot|\sigma_k)]$. We seek to show that $J(\sigma_k)$ is of the form given in the statement of the lemma.

First note that

$$J'(\sigma_k) = -E_{\epsilon}[\epsilon(F(z) - F(\bar{x} + \sigma_k \epsilon)) | \epsilon \sim \Phi(\sigma_k)] + \frac{d}{d\Phi} E_{\epsilon} \left[\int_{\bar{x}+\sigma_k \epsilon}^z F(z) - F(y) dy | \epsilon \sim \Phi(\cdot|\sigma_k) \right]$$

where $\frac{d}{d\Phi} E_{\epsilon}[Z(\epsilon, \sigma_k) | \epsilon \sim \Phi(\cdot|\sigma_k)]$ denotes the derivative of the expectation of $Z(\epsilon, \sigma_k)$ arising through the changes in $\Phi(\cdot|\sigma_k)$ induced by changes in σ_k (that is, if $\partial(\epsilon; \sigma_k)$ denotes the density corresponding to $\Phi(\cdot|\sigma_k)$, then $\frac{d}{d\Phi} E_{\epsilon}[Z(\epsilon, \sigma_k) | \epsilon \sim \Phi(\cdot|\sigma_k)] \equiv \int_{-\infty}^{\infty} Z(\epsilon, \sigma_k) \frac{\partial \partial_{\sigma_k}}{\partial \sigma_k}(\epsilon; \sigma_k) d\epsilon$). Similarly, letting $\frac{d^m}{d\Phi^m} E_{\epsilon}[Z(\epsilon, \sigma_k) | \epsilon \sim \Phi(\cdot|\sigma_k)] \equiv \int_{-\infty}^{\infty} Z(\epsilon, \sigma_k) \frac{\partial^m \partial_{\sigma_k}}{\partial \sigma_k^m}(\epsilon; \sigma_k) d\epsilon$ for all m , we have

$$\begin{aligned} J''(\sigma_k) &= E_{\epsilon}[\epsilon^2 f(\bar{x} + \sigma_k \epsilon) | \epsilon \sim \Phi(\cdot|\sigma_k)] - 2 \frac{d}{d\Phi} E_{\epsilon}[\epsilon(F(z) - F(\bar{x} + \sigma_k \epsilon)) | \epsilon \sim \Phi(\cdot|\sigma_k)] \\ &\quad + \frac{d^2}{d\Phi^2} E_{\epsilon} \left[\int_{\bar{x}+\sigma_k \epsilon}^z F(z) - F(y) dy | \epsilon \sim \Phi(\cdot|\sigma_k) \right], \end{aligned}$$

$$\begin{aligned} J'''(\sigma_k) &= E_{\epsilon}[\epsilon^3 f'(\bar{x} + \sigma_k \epsilon) | \epsilon \sim \Phi(\cdot|\sigma_k)] + 3 \frac{d}{d\Phi} E_{\epsilon}[\epsilon^2 f(\bar{x} + \sigma_k \epsilon) | \epsilon \sim \Phi(\cdot|\sigma_k)] \\ &\quad - 3 \frac{d^2}{d\Phi^2} E_{\epsilon}[\epsilon(F(z) - F(\bar{x} + \sigma_k \epsilon)) | \epsilon \sim \Phi(\cdot|\sigma_k)] \\ &\quad + \frac{d^3}{d\Phi^3} E_{\epsilon} \left[\int_{\bar{x}+\sigma_k \epsilon}^z F(z) - F(y) dy | \epsilon \sim \Phi(\cdot|\sigma_k) \right], \end{aligned}$$

and

$$\begin{aligned} J^{(m)}(\sigma_k) &= E_{\epsilon}[\epsilon^4 f^{(m)}(\bar{x} + \sigma_k \epsilon) | \epsilon \sim \Phi(\cdot|\sigma_k)] + 4 \frac{d}{d\Phi} E_{\epsilon}[\epsilon^3 f'(\bar{x} + \sigma_k \epsilon) | \epsilon \sim \Phi(\cdot|\sigma_k)] \\ &\quad + 6 \frac{d^2}{d\Phi^2} E_{\epsilon}[\epsilon^2 f(\bar{x} + \sigma_k \epsilon) | \epsilon \sim \Phi(\cdot|\sigma_k)] \\ &\quad - 4 \frac{d^3}{d\Phi^3} E_{\epsilon}[\epsilon(F(z) - F(\bar{x} + \sigma_k \epsilon)) | \epsilon \sim \Phi(\cdot|\sigma_k)] \\ &\quad + \frac{d^4}{d\Phi^4} E_{\epsilon} \left[\int_{\bar{x}+\sigma_k \epsilon}^z F(z) - F(y) dy | \epsilon \sim \Phi(\cdot|\sigma_k) \right], \end{aligned}$$

Note that when $\sigma_k = 0$, we have $E_{\epsilon}[\int_{\bar{x}+\sigma_k \epsilon}^z F(z) - F(y) dy | \epsilon \sim \Phi(\cdot|\sigma_k)] = \int_{\bar{x}}^z F(z) - F(y) dy$ for any distribution $\Phi(\cdot|\sigma_k)$, $E_{\epsilon}[\epsilon(F(z) - F(\bar{x} + \sigma_k \epsilon)) | \epsilon \sim \Phi(\cdot|\sigma_k)] = E_{\epsilon}[\epsilon(F(z) - F(\bar{x})) | \epsilon \sim \Phi(\cdot|\sigma_k)] = 0$ for any distribution $\Phi(\cdot|\sigma_k)$ with mean zero, and $E_{\epsilon}[\epsilon^2 f(\bar{x} + \sigma_k \epsilon) | \epsilon \sim \Phi(\cdot|\sigma_k)] = E_{\epsilon}[\epsilon^2 f(\bar{x}) | \epsilon \sim \Phi(\cdot|\sigma_k)] = f(\bar{x})$ for any distribution $\Phi(\cdot|\sigma_k)$ with mean zero and variance one. Thus $\frac{d^m}{d\Phi^m} E_{\epsilon}[\int_{\bar{x}+\sigma_k \epsilon}^z F(z) - F(y) dy | \epsilon \sim \Phi(\cdot|\sigma_k)] = 0$, $\frac{d^m}{d\Phi^m} E_{\epsilon}[\epsilon(F(z) - F(\bar{x} + \sigma_k \epsilon)) | \epsilon \sim \Phi(\cdot|\sigma_k)] = 0$, and $\frac{d^m}{d\Phi^m} E_{\epsilon}[\epsilon^2 f(\bar{x} + \sigma_k \epsilon) | \epsilon \sim \Phi(\cdot|\sigma_k)] = 0$ for all m when evaluated at $\sigma_k = 0$.

By using these facts, the fact that $\Phi(\cdot|0)$ is standard normal, and the above expressions for $J(\sigma_k)$ and its derivatives, it follows that $J(0) = \int_{\bar{x}}^z F(z) - F(y) dy$, $J'(0) = 0$, $J''(0) = F(\bar{x})$, $J'''(0) = 0$, and $J''''(0) = E_\epsilon[\epsilon^4 J''(\bar{x})] + 4 \frac{d}{d\sigma_k} E_\epsilon[\epsilon^3 J'(\bar{x})] + 4 \frac{d^2}{d\sigma_k^2} E_\epsilon[\epsilon^2 J(\bar{x})] + 4 \frac{d^3}{d\sigma_k^3} E_\epsilon[\epsilon J(\bar{x})] + 4 \frac{d^4}{d\sigma_k^4} E_\epsilon[J(\bar{x})]$. This in turn implies that the fourth-order Taylor approximation to $E_\epsilon[\int_{\bar{x}+\sigma_k\epsilon}^z F(z) - F(y) dy]$ is

$$E_\epsilon \left[\int_{\bar{x}+\sigma_k\epsilon}^z F(z) - F(y) dy \right] = \int_{\bar{x}}^z F(z) - F(y) dy + \frac{1}{2} \sigma_k^2 f(\bar{x}) + a(\bar{x}) \sigma_k^4 + o(\sigma_k^4),$$

where $a(\bar{x}) \equiv \frac{1}{24} [E_\epsilon[\epsilon^4 J''(\bar{x})] + 4 \frac{d}{d\sigma_k} E_\epsilon[\epsilon^3 J'(\bar{x})] + 4 \frac{d^2}{d\sigma_k^2} E_\epsilon[\epsilon^2 J(\bar{x})] + 4 \frac{d^3}{d\sigma_k^3} E_\epsilon[\epsilon J(\bar{x})] + 4 \frac{d^4}{d\sigma_k^4} E_\epsilon[J(\bar{x})]]$. ■

Proof of Theorems 5 and 6: Since these results are special cases of Theorems 9 and 10 respectively, the proofs of these results are omitted.

Before proving Theorem 7, we first introduce some notation for the finite-horizon version of this game. If the game has a finite time horizon and will last an additional T periods, we let $V_{k,T}(\bar{x})$ denote the value of the dynamic program that arises when the auctioneer follows the optimal strategy. By analogy to Lemma 2, we know that

$$V_{k,T}(\bar{x}) = \frac{1}{1-\delta} \left(\max_z \left[- \int_{\bar{x}+\sigma_k\epsilon}^z F(z) - F(y) dy + \delta F(z) (E_{\bar{x}}[V_{k+1,T-1}(\bar{x})] - V_{k,T-1}(\bar{x})) \right] \right)$$

when $T > 0$ and $V_{k,T}(\bar{x}) = -E_\epsilon[\int_{\bar{x}+\sigma_k\epsilon}^z F(\bar{x}) - F(y) dy]$ when $T = 0$. Also note that $\lim_{T \rightarrow \infty} V_{k,T}(\bar{x}) = V_k(\bar{x})$, where $V_k(\bar{x})$ is the value of the dynamic program for the original infinite-horizon game. Finally note that

Lemma 21 $V_{k,T}(\bar{x})$ is twice differentiable in \bar{x} for all k and T . Furthermore, $\lim_{k \rightarrow \infty} V_{k,T}(\bar{x}) = 0$ and $\lim_{k \rightarrow \infty} V_{k,T}'(\bar{x}) = 0$ for all T .

Proof We prove this result by induction on T . The base case, $T = 0$, holds because the fact that $f(\cdot)$ is continuously differentiable implies $F(\cdot)$ is twice differentiable and $V_{k,T}(\bar{x}) = -E_\epsilon[\int_{\bar{x}+\sigma_k\epsilon}^z F(\bar{x}) - F(y) dy]$ is also twice differentiable in \bar{x} . Furthermore, $V_{k,T}'(\bar{x}) = E_\epsilon[F(\bar{x}) - F(\bar{x} + \sigma_k\epsilon) - \int_{\bar{x}+\sigma_k\epsilon}^{\bar{x}} f(\bar{x}) - F(\bar{x} + \sigma_k\epsilon)]$ and $V_{k,T}''(\bar{x}) = E_\epsilon[f(\bar{x}) - f(\bar{x} + \sigma_k\epsilon)]$, which both tend to zero as $k \rightarrow \infty$. Thus the result holds for $T = 0$. Now suppose the result is true for $T - 1$ and use this to prove the result must also hold for T . Since

$$V_{k,T}(\bar{x}) = \frac{1}{1-\delta} \left(\max_z \left[- \int_{\bar{x}+\sigma_k\epsilon}^z F(z) - F(y) dy + \delta F(z) (E_{\bar{x}}[V_{k+1,T-1}(\bar{x})] - V_{k,T-1}(\bar{x})) \right] \right),$$

we know from analogy to Theorem 3 that the optimal bid z satisfies $z = \bar{x} + \delta [E_{\bar{x}}[V_{k+1,T-1}(\bar{x})] - V_{k,T-1}(\bar{x})]$. From the induction hypothesis, we thus know that the optimal bid $z_k(\bar{x})$ is twice differentiable in \bar{x} and that $\lim_{k \rightarrow \infty} z_k'(\bar{x}) = 1$ and $\lim_{k \rightarrow \infty} z_k''(\bar{x}) = 0$.

This in turn implies that $E_\epsilon[- \int_{\bar{x}+\sigma_k\epsilon}^{z_k(\bar{x})} F(z_k(\bar{x})) - F(y) dy]$ is twice differentiable in \bar{x} and that $\frac{d}{d\bar{x}} E_\epsilon[- \int_{\bar{x}+\sigma_k\epsilon}^{z_k(\bar{x})} F(z_k(\bar{x})) - F(y) dy] = E_\epsilon[F(z_k(\bar{x})) - F(\bar{x} + \sigma_k\epsilon) - \int_{\bar{x}+\sigma_k\epsilon}^{z_k(\bar{x})} f(z_k(\bar{x})) dy]$, which tends to zero as $k \rightarrow \infty$ since $\lim_{k \rightarrow \infty} z_k(\bar{x}) = \bar{x}$. This also further implies that $\frac{d^2}{d\bar{x}^2} E_\epsilon[- \int_{\bar{x}+\sigma_k\epsilon}^{z_k(\bar{x})} F(z_k(\bar{x})) - F(y) dy] = E_\epsilon[z_k'(\bar{x}) f(z_k(\bar{x})) - f(\bar{x} + \sigma_k\epsilon)] - (z_k'(\bar{x}) - 1) z_k'(\bar{x}) f(z_k(\bar{x})) -$

$\int_{\bar{x}+\sigma_k\epsilon}^{z_k(\bar{x})} f(z_k(\bar{x})) + (z_k'(\bar{x}))^2 f'(z_k(\bar{x})) dy]$, which tends to zero as $k \rightarrow \infty$ since $\lim_{k \rightarrow \infty} z_k(\bar{x}) = \bar{x}$ and $\lim_{k \rightarrow \infty} z_k'(\bar{x}) = 1$.

From the induction hypothesis, we also know that $F(z_k(\bar{x})) (E_{\bar{x}}[V_{k+1,T-1}(\bar{x})] - V_{k,T-1}(\bar{x}))$ is twice differentiable in \bar{x} and that the first and second derivatives of this expression with respect to \bar{x} tend to zero as $k \rightarrow \infty$. By combining this with the results in the previous two paragraphs, it follows that $V_{k,T}(\bar{x})$ is twice differentiable in \bar{x} and $\lim_{k \rightarrow \infty} V_{k,T}'(\bar{x}) = 0$ and $\lim_{k \rightarrow \infty} V_{k,T}''(\bar{x}) = 0$ for all T . The result follows by induction. ■

We use these observations about the finite-horizon game to first prove that $E[V_{k+1}(\bar{x}') - V_k(\bar{x})] = O(\frac{1}{k^2})$ in Lemma 22. Then we use this preliminary result to prove Theorem 7.

Lemma 22 $E[V_{k+1}(\bar{x}') - V_k(\bar{x})] = O(\frac{1}{k^2})$ for large k .

Proof Note that if an ad is displayed, then one of two possible things will happen to the ad—either the ad will receive a click or the ad will not receive a click. Let p denote the probability that the ad will receive a click, let \bar{x}_c denote the estimated eCPM of the ad if the ad receives a click, and let \bar{x}_n denote the estimated eCPM of the ad if the ad does not receive a click. Note that $p\bar{x}_c + (1-p)\bar{x}_n = \bar{x}$.

From Lemma 21 we know that $V_{k,T}(\bar{x})$ is twice differentiable in \bar{x} for all k and T . Thus the second-order Taylor approximations for $V_{k+1,T}(\bar{x}_c)$ and $V_{k+1,T}(\bar{x}_n)$ are

$$V_{k+1,T}(\bar{x}_c) = V_{k+1,T}(\bar{x}) + V_{k+1,T}'(\bar{x})(\bar{x}_c - \bar{x}) + \frac{1}{2} V_{k+1,T}''(\bar{x})(\bar{x}_c - \bar{x})^2 + o(\bar{x}_c - \bar{x})^2$$

and

$$V_{k+1,T}(\bar{x}_n) = V_{k+1,T}(\bar{x}) + V_{k+1,T}'(\bar{x})(\bar{x}_n - \bar{x}) + \frac{1}{2} V_{k+1,T}''(\bar{x})(\bar{x}_n - \bar{x})^2 + o(\bar{x}_n - \bar{x})^2.$$

Thus if \bar{x}' denotes the actual realization of the estimated eCPM after the ad has been shown $k+1$ times (\bar{x}' will equal \bar{x}_c with probability p and \bar{x}_n with probability $1-p$), then by using the fact that $p\bar{x}_c + (1-p)\bar{x}_n = \bar{x}$ and by taking a weighted average of the two previous equations, we find that

$$\begin{aligned} E[V_{k+1,T}(\bar{x}')] &= pV_{k+1,T}(\bar{x}_c) + (1-p)V_{k+1,T}(\bar{x}_n) \\ &= V_{k+1,T}(\bar{x}) + \frac{1}{2} V_{k+1,T}''(\bar{x}) E[(\bar{x}' - \bar{x})^2] + o(E[(\bar{x}' - \bar{x})^2]). \end{aligned}$$

From this it follows that

$$E[V_{k+1,T}(\bar{x}') - V_k(\bar{x})] = V_{k+1,T}(\bar{x}) - V_k(\bar{x}) + \frac{1}{2} V_{k+1,T}''(\bar{x}) E[(\bar{x}' - \bar{x})^2] + o(E[(\bar{x}' - \bar{x})^2]). \quad (1)$$

If c denotes the number of clicks that an ad has received so far, then the predicted click-through rate for an ad that has received a large number of impressions, k , will be approximately $\frac{c}{k}$. Thus if b denotes the bid per click that the ad places, then the eCPM for an ad that has received c clicks and has been shown k times will be $\bar{x} \approx \frac{bc}{k}$. From this it follows that $\bar{x}_c \approx \frac{b(c+1)}{k+1}$, $\bar{x}_n \approx \frac{bc}{k+1}$, $\bar{x}_c - \bar{x} \approx \frac{bc}{k(k+1)}$, and $\bar{x}_n - \bar{x} \approx -\frac{bc}{k(k+1)}$. Thus

$\bar{x}' - \bar{x} = O(\frac{1}{k})$ for all possible realizations of \bar{x}' , and $(\bar{x}' - \bar{x})^2 = O(\frac{1}{k^2})$. Furthermore, from Lemma 21 we know that $\lim_{k \rightarrow \infty} V_{k+1,T}^{\#}(\bar{x}) = 0$. Thus we can rewrite equation (1) as

$$E[V_{k+1,T}(\bar{x}) - V_k,T(\bar{x})] = V_{k+1,T}(\bar{x}) - V_k,T(\bar{x}) + o\left(\frac{1}{k^2}\right).$$

By using the fact that $\lim_{T \rightarrow \infty} V_k,T(\bar{x}) = V_k(\bar{x})$, where $V_k(\bar{x})$ denotes the value of the dynamic program in the original infinite horizon game, we then know that

$$\begin{aligned} E[V_{k+1}(\bar{x}) - V_k(\bar{x})] &= V_{k+1}(\bar{x}) - V_k(\bar{x}) + o\left(\frac{1}{k^2}\right) \\ &= \frac{v(\bar{x})}{k} - \frac{v(\bar{x})}{k+1} + O\left(\frac{1}{k^2}\right) \\ &= O\left(\frac{1}{k^2}\right). \end{aligned}$$

■

Proof of Theorem 7: We have seen in the proof of Lemma 22 that $E[V_{k+1}(\bar{x}') - V_k(\bar{x})] = V_{k+1}(\bar{x}) - V_k(\bar{x}) + o(\frac{1}{k^2})$. When combined with the fact that $V_k(\bar{x}) = -\frac{v(\bar{x})}{k} + O(\frac{1}{k^2})$, this immediately implied that $E[V_{k+1}(\bar{x}') - V_k(\bar{x})] = O(\frac{1}{k^2})$. If we are able to further prove that we can write $V_k(\bar{x}) = -\frac{v(\bar{x})}{k} + \frac{w(\bar{x})}{k^2} + o(\frac{1}{k^2})$ for some function $w(\bar{x})$, it will then follow that $V_{k+1}(\bar{x}) - V_k(\bar{x}) = \frac{v(\bar{x})}{k(k+1)} + o(\frac{1}{k^2})$. Thus we first seek to show that we can write $V_k(\bar{x})$ as $V_k(\bar{x}) = -\frac{v(\bar{x})}{k} + \frac{w(\bar{x})}{k^2} + o(\frac{1}{k^2})$.

Since $E[V_{k+1}(\bar{x}) - V_k(\bar{x})] = O(\frac{1}{k^2})$ for large k and the optimal bidding strategy entails setting $z = \bar{x} + \delta(E[V_{k+1}(\bar{x}') - V_k(\bar{x})])$, it must be the case that $z = \bar{x} + O(\frac{1}{k^2})$ for large k . From this it follows that $\int_{\bar{x}}^z F(z) - F(y) dy = o(\frac{1}{k^2})$ under the optimal bidding strategy z for large k .

Now we have seen in Lemma 4 that $E_\epsilon[\int_{\bar{x}+\alpha_k^\epsilon}^z F(z) - F(y) dy] = \int_{\bar{x}}^z F(z) - F(y) dy + \frac{1}{2}\sigma_k^2 f(\bar{x}) + o(a(\bar{x})\sigma_k^4 + o(\sigma_k^4))$ for some constant $a(\bar{x})$ for large k . Since $\int_{\bar{x}}^z F(z) - F(y) dy = o(\frac{1}{k^2})$ under the optimal bidding strategy z and $\sigma_k^2 = \frac{s^2(\bar{x})}{k} + \frac{h(\bar{x})}{k^2} + o(\frac{1}{k^2})$ for large k , it then follows that $E_\epsilon[\int_{\bar{x}+\alpha_k^\epsilon}^z F(z) - F(y) dy] = \frac{1}{2k}s^2(\bar{x})f(\bar{x}) + \frac{1}{k^2}[\frac{h(\bar{x})f(\bar{x})}{2} + a(\bar{x})s^4(\bar{x})] + o(\frac{1}{k^2})$ for large k , which we can rewrite as $E_\epsilon[\int_{\bar{x}+\alpha_k^\epsilon}^z F(z) - F(y) dy] = \frac{1}{2k}s^2(\bar{x})f(\bar{x}) + \frac{1}{k^2}w(\bar{x}) + o(\frac{1}{k^2})$, where $w(\bar{x}) \equiv \frac{h(\bar{x})f(\bar{x})}{2} + a(\bar{x})s^4(\bar{x})$.

But $-E_\epsilon[\int_{\bar{x}+\alpha_k^\epsilon}^z F(z) - F(y) dy]$ represents the auctioneer's per-period payoff in the next auction. Thus if \bar{x}' denotes the estimated eCPM of the ad after an additional j periods have passed and k' denotes the number of impressions the ad has received after an additional j periods have passed, then the auctioneer's per-period payoff in the period after an additional j periods have passed is $-\frac{1}{2k'}s^2(\bar{x}')f(\bar{x}') - \frac{1}{2k'}u(\bar{x}') + o(\frac{1}{k'^2})$. The difference between this and $-\frac{1}{2k}s^2(\bar{x})f(\bar{x})$ is

$$-\frac{1}{2k'}s^2(\bar{x}')f(\bar{x}') - \frac{1}{2k'}u(\bar{x}') + \frac{1}{2k}s^2(\bar{x})f(\bar{x}) + o\left(\frac{1}{k^2}\right)$$

$$\begin{aligned} &= \frac{k's^2(\bar{x})f(\bar{x}) - ks^2(\bar{x}')f(\bar{x}')}{2kk'} - \frac{1}{2k^2}u(\bar{x}) + \left[\frac{1}{2k^2}u(\bar{x}) - \frac{1}{2kk'^2}u(\bar{x}')\right] + o\left(\frac{1}{k^2}\right) \\ &= \frac{k's^2(\bar{x})f(\bar{x}) - ks^2(\bar{x}')f(\bar{x}')}{2kk'} - \frac{1}{2k^2}u(\bar{x}) + \frac{k'^2u(\bar{x}) - k^2u(\bar{x}')}{2k^2k'^2} + o\left(\frac{1}{k^2}\right) \\ &= \frac{k's^2(\bar{x})f(\bar{x}) - k[s^2(\bar{x})f(\bar{x}) + (\bar{x}' - \bar{x})d(\bar{x}) + o(\bar{x}' - \bar{x})]}{2kk'} - \frac{1}{2k^2}u(\bar{x}) \\ &\quad + \frac{k'^2u(\bar{x}) - k^2[u(\bar{x}) + O(\bar{x}' - \bar{x})]}{2k^2k'^2} + o\left(\frac{1}{k^2}\right), \end{aligned} \tag{2}$$

where $d(\bar{x})$ denotes the derivative of the function $s^2(\bar{x})f(\bar{x})$ with respect to \bar{x} . By the same reasoning as in the proof of Lemma 22, we know that $\bar{x}' - \bar{x} = O(\frac{1}{k})$ for all possible realizations of \bar{x}' . Thus we can rewrite the expression in equation (2) as

$$\begin{aligned} &\frac{(k' - k)s^2(\bar{x})f(\bar{x}) - k(\bar{x}' - \bar{x})d(\bar{x})}{2kk'} - \frac{1}{2k^2}u(\bar{x}) + \frac{(k'^2 - k^2)u(\bar{x})}{2k^2k'^2} + o\left(\frac{1}{k^2}\right) \\ &= \frac{(k' - k)s^2(\bar{x})f(\bar{x}) - k(\bar{x}' - \bar{x})d(\bar{x})}{2kk'} - \frac{1}{2k^2}u(\bar{x}) + o\left(\frac{1}{k^2}\right). \end{aligned} \tag{3}$$

Now note that $E[\bar{x}'] = \bar{x}$, where the expectation is taken over the uncertain realization of \bar{x}' in another j periods. Thus the expectation of the expression in equation (3) is

$$E\left[\frac{(k' - k)s^2(\bar{x})f(\bar{x})}{2kk'}\right] - \frac{1}{2k^2}u(\bar{x}) + o\left(\frac{1}{k^2}\right), \tag{4}$$

where the expectation is taken over the uncertain realization of k' . This expression can in turn be written as

$$\frac{\bar{\pi}_j(\bar{x})s^2(\bar{x})f(\bar{x}) - u(\bar{x})}{2k^2} + o\left(\frac{1}{k^2}\right), \tag{5}$$

where $\bar{\pi}_j(\bar{x})$ denotes the expected number of additional impressions that the ad with uncertain eCPM receives after an additional j periods have passed (which will equal 0 when $j = 0$ and vary approximately linearly with j for large k).

The expression in equation (5) gives the difference between the auctioneer's actual expected payoff in the period after an additional j periods have passed and $-\frac{1}{2k}s^2(\bar{x})f(\bar{x})$. From this it follows that the difference between the auctioneer's actual payoff $V_k(\bar{x})$ and the payoff the auctioneer would receive if the auctioneer obtained a payoff of $-\frac{1}{2k}s^2(\bar{x})f(\bar{x})$ in every future period is $\sum_{j=0}^{\infty} \delta^j \frac{\bar{\pi}_j(\bar{x})s^2(\bar{x})f(\bar{x}) - u(\bar{x})}{2k^2} + o(\frac{1}{k^2})$, which can be written as $\frac{w(\bar{x})}{k^2} + o(\frac{1}{k^2})$ for some function $w(\bar{x})$. Thus we can write $V_k(\bar{x})$ as $V_k(\bar{x}) = -\frac{v(\bar{x})}{k} + \frac{w(\bar{x})}{k^2} + o(\frac{1}{k^2})$ for some function $w(\bar{x})$.

But we have seen in the proof of Lemma 22 that $E[V_{k+1}(\bar{x}') - V_k(\bar{x})] = V_{k+1}(\bar{x}) - V_k(\bar{x}) + o(\frac{1}{k^2})$. Since $V_k(\bar{x}) = -\frac{v(\bar{x})}{k} + \frac{w(\bar{x})}{k^2} + o(\frac{1}{k^2})$, it then follows that $V_{k+1}(\bar{x}') - V_k(\bar{x}) = \frac{v(\bar{x})}{k(k+1)} + o(\frac{1}{k^2})$. ■

Proof of Theorem 8: Recall that

$$V_k(\bar{x}) = \frac{1}{1 - \delta} \left(\max_z E_\epsilon \left[- \int_{\bar{x}+\alpha_k^\epsilon}^z F(z) - F(y) dy + \delta F(z)(E_{\bar{x}'}[V_{k+1}(\bar{x}')] - V_k(\bar{x})) \right] \right)$$

and a second-order Taylor approximation for $E_\epsilon[\int_{\bar{x}+\sigma_k\epsilon}^z F(z) - F(y) dy] =$

$$E_\epsilon \left[\int_{\bar{x}+\sigma_k\epsilon}^z F(z) - F(y) dy \right] = \int_{\bar{x}}^z F(z) - F(y) dy + \frac{1}{2} \sigma_k^2 f(\bar{x}) + o(\sigma_k^2)$$

Now $z = \bar{x} + \delta(E_{\bar{x}}[V_{k+1}(\bar{x})] - V_k(\bar{x}))$, so $z - \bar{x} \leq -\delta V_k(\bar{x})$, a term which is $O(f(\bar{x})\sigma_k^2)$. From this it follows that $\int_{\bar{x}}^z F(z) - F(y) dy = O(F(\bar{x})f(\bar{x})\sigma_k^2) = o(f(\bar{x})\sigma_k^2)$. And we also know that $\delta F(z)(E_{\bar{x}}[V_{k+1}(\bar{x})] - V_k(\bar{x})) \leq -\delta F(z)V_k(\bar{x}) = O(F(\bar{x})f(\bar{x})\sigma_k^2) = o(f(\bar{x})\sigma_k^2)$.

Combining these results gives $E_\epsilon[\int_{\bar{x}+\sigma_k\epsilon}^z F(z) - F(y) dy] = \frac{1}{2} \sigma_k^2 f(\bar{x}) + o(f(\bar{x})\sigma_k^2)$ and $F(z)(E_{\bar{x}}[V_{k+1}(\bar{x})] - V_k(\bar{x})) = o(f(\bar{x})\sigma_k^2)$. Substituting this in to our expression for $V_k(\bar{x})$ then gives $V_k(\bar{x}) = -\frac{1}{2(1-\delta)} f(\bar{x})\sigma_k^2 + o(f(\bar{x})\sigma_k^2)$. ■

Proof of Theorem 9: First note that it must be the case that $V_k(\bar{x}) = \Omega(\frac{1}{k})$ for large k . We know that $\sigma_{a, k_a}^2 = \Theta(\frac{1}{k_a}) = \Theta(\frac{1}{\bar{x}k})$ for large k , so the immediate reward in any given period is at least on the same order as $\frac{1}{k}$ regardless of which ad-context pair a arises in the auction. Thus we know that $V_k(\bar{x}) = \Omega(\frac{1}{k})$ for large k .

We also know that $V_k(\bar{x}) = O(\frac{1}{k})$ for large k . To see this, note that the auctioneer can ensure that his loss in an auction involving the ad-context pair a in any given period is $O(\frac{1}{k_a}) = O(\frac{1}{k})$ by bidding $z_a = \bar{x}_a$, so the auctioneer can thus ensure that his expected loss in any given period is $O(\frac{1}{k})$ unconditional on the precise ad-context pair that arises. And if the auctioneer's loss in any given period is $O(\frac{1}{k})$, then the player's total loss from the game will also be no greater than $O(\frac{1}{k})$ because the present value of the sum of losses that are $O(\frac{1}{k})$, $\sum_{j=k}^{\infty} \delta^{j-k} \frac{1}{j}$, is also $O(\frac{1}{k})$ since $1 < \sum_{j=k}^{\infty} \delta^{j-k} \frac{1}{j} < \sum_{j=k}^{\infty} \delta^{j-k} = \frac{1}{1-\delta}$ implies $\frac{1}{k} < \sum_{j=k}^{\infty} \delta^{j-k} \frac{1}{j} < \frac{1}{(1-\delta)k}$. Thus $V_k(\bar{x}) = \Theta(\frac{1}{k})$ for large k . ■

Proof of Theorem 10: Since $V_k(\bar{x}) = \Theta(\frac{1}{k})$ for large k , we have $E_{\bar{x}(a), \bar{x}}[V_{k(a), \bar{x}}(\bar{x}(a))] - V_k(\bar{x}) = O(\frac{1}{k})$ for large k . Thus since the optimal bidding strategy entails setting $z_a = \bar{x}_a + \delta(E_{\bar{x}(a)}[V_{k(a), \bar{x}}(\bar{x}(a))] - V_k(\bar{x}))$, it must be the case that $z_a = \bar{x}_a + O(\frac{1}{k})$ for large k . From this it follows that $\int_{\bar{x}_a}^{z_a} F_a(z_a) - F_a(y) dy = O(\frac{1}{k^2})$ under the optimal bidding strategy for large k .

Now we have seen in Lemma 4 that $E_\epsilon[\int_{\bar{x}_a+\sigma_{a, k_a}\epsilon}^{z_a} F_a(z_a) - F_a(y) dy] = \int_{\bar{x}_a}^{z_a} F_a(z_a) - F_a(y) dy + \frac{1}{2} \sigma_{a, k_a}^2 f_a(\bar{x}_a) + O(\sigma_{a, k_a}^4)$ for large k . Since $\int_{\bar{x}_a}^{z_a} F_a(z_a) - F_a(y) dy = O(\frac{1}{k^2})$ under the optimal bidding strategy z_a and $\sigma_{a, k_a}^2 = \frac{\sigma_a^2(\bar{x}_a)}{k_a} + O(\frac{1}{k^2})$ for large k , it then follows that $E_\epsilon[\int_{\bar{x}_a+\sigma_{a, k_a}\epsilon}^{z_a} F_a(z_a) - F_a(y) dy] = \frac{\sigma_a^2(\bar{x}_a)}{2k_a} f_a(\bar{x}_a) + O(\frac{1}{k^2})$ for large k .

But $-E_\epsilon[\int_{\bar{x}_a+\sigma_{a, k_a}\epsilon}^{z_a} F_a(z_a) - F_a(y) dy]$ represents the auctioneer's per-period payoff if the next auction is an auction for the advertiser-context pair a . Thus the auctioneer's expected per-period payoff unconditional on what ad-context pair appears in the next auction is $\sum_{a=1}^m \pi_a \frac{\sigma_a^2(\bar{x}_a)}{2k_a} f_a(\bar{x}_a) + O(\frac{1}{k^2}) = \sum_{a=1}^m \pi_a \frac{1}{2\bar{x}_a k_a} \sigma_a^2(\bar{x}_a) f_a(\bar{x}_a) + O(\frac{1}{k^2})$ for large k . From this it follows that if $g(\bar{x}) \equiv \sum_{a=1}^m \pi_a \frac{1}{2\bar{x}_a} \sigma_a^2(\bar{x}_a) f_a(\bar{x}_a)$, then the expected per-period utility that one obtains at each point in the game unconditional on what ad-context pair appears in the next auction is $\frac{1}{k} g(\bar{x}) + O(\frac{1}{k^2})$.

Since $V_k(\bar{x})$ can alternatively be expressed as the discounted sum of the per-period utility that one can obtain at each point in the game, it then follows that $|kV_k(\bar{x})| \leq \sum_{j=k}^{\infty} \delta^{j-k} [g(\bar{x}) + O(\frac{1}{k})]$, meaning $|kV_k(\bar{x})| \leq \frac{1}{1-\delta} g(\bar{x}) + O(\frac{1}{k})$ and $|kV_k(\bar{x})| \geq \sum_{j=k}^{\infty} \delta^{j-k} \frac{1}{j} g(\bar{x}) +$

$O(\frac{1}{k}) = \frac{1}{1-\delta} g(\bar{x}) + O(\frac{1}{k})$ in the limit as $k \rightarrow \infty$. From this it follows that $|kV_k(\bar{x})| = \frac{1}{1-\delta} g(\bar{x}) + O(\frac{1}{k})$ and $kV_k(\bar{x}) = -\frac{1}{1-\delta} g(\bar{x}) + O(\frac{1}{k}) = -\frac{1}{2(1-\delta)} \sum_{a=1}^m \pi_a \frac{1}{\bar{x}_a} \sigma_a^2(\bar{x}_a) f_a(\bar{x}_a) + O(\frac{1}{k})$. ■

Proof of Theorem 13: Under the knowledge gradient framework, the incremental amount that one increases one's bid by beyond the expected value of the advertising opportunity is $\delta(E_{\bar{x}}[U_{k+1}(\bar{x}')] - U_k(\bar{x}))$. And under the full dynamic programming problem, the incremental amount that one increases one's bid is $\delta(E_{\bar{x}}[V_{k+1}(\bar{x}')] - V_k(\bar{x}))$. Thus if ΔV_{k+1} denotes the difference between the values of $E_{\bar{x}}[V_{k+1}(\bar{x}')] - V_k(\bar{x})$ and $E_{\bar{x}}[U_{k+1}(\bar{x}')] - U_k(\bar{x})$ denotes the difference between the values of $V_k(\bar{x})$ and $U_k(\bar{x})$, then the difference between the incremental amount that one increases one's bid under the full dynamic programming problem and under the knowledge gradient framework is $\delta(\Delta V_{k+1} - \Delta V_k)$.

But a condition of the theorem is that $\Delta V_{k+1} < \Delta V_k$. Thus $\delta(\Delta V_{k+1} - \Delta V_k) < 0$, and the difference between the incremental amount that one increases one's bid under the full dynamic programming problem and the incremental amount that one increases one's bid under the knowledge gradient framework is negative. From this it follows that the incremental amount by which one would increase one's bid under the knowledge gradient framework is indeed greater than it is under the full dynamic programming problem. ■

Proof of Theorem 14: In the knowledge gradient framework, $U_k(\bar{x})$ is just the discounted sum of the value of simply bidding $z = \bar{x}$ in each period when an ad has received k impressions so far and one's best estimate for the eCPM of the ad is \bar{x} . Now we know from applying Lemma 4 to the special case in which $z = \bar{x}$ that the per-period payoff from bidding $z = \bar{x}$ in each period when an ad has received k impressions so far and one's best estimate for the eCPM of the ad is \bar{x} is $-\frac{1}{2} \sigma_k^2 f(\bar{x}) - a(\bar{x})\sigma_k^4 + o(\sigma_k^4) = -\frac{1}{2k} \sigma_k^2(\bar{x}) f(\bar{x}) + O(\frac{1}{k^2})$. Thus $U_k(\bar{x}) = -\frac{1}{2(1-\delta)k} \sigma_k^2(\bar{x}) f(\bar{x}) + O(\frac{1}{k^2})$.

But we have seen in Theorem 7 that when $V_k(\bar{x}) = -\frac{1}{2(1-\delta)k} \sigma_k^2(\bar{x}) f(\bar{x}) + O(\frac{1}{k^2})$ and the per-period payoff from making the optimal bid is $-\frac{1}{2} \sigma_k^2 f(\bar{x}) - a(\bar{x})\sigma_k^4 + o(\sigma_k^4)$, then it must be the case that $E_{\bar{x}}[V_{k+1}(\bar{x}')] - V_k(\bar{x}) = \frac{v(\bar{x})}{k(k+1)} + o(\frac{1}{k^2})$ for large k , where $v(\bar{x}) \equiv \frac{1}{2(1-\delta)} \sigma_k^2(\bar{x}) f(\bar{x})$. An identical argument illustrates that when $U_k(\bar{x}) = -\frac{1}{2(1-\delta)k} \sigma_k^2(\bar{x}) f(\bar{x}) + O(\frac{1}{k^2})$ and the per-period payoff from making the optimal bid is $-\frac{1}{2} \sigma_k^2 f(\bar{x}) - a(\bar{x})\sigma_k^4 + o(\sigma_k^4)$, then it must be the case that $E_{\bar{x}}[U_{k+1}(\bar{x}')] - U_k(\bar{x}) = \frac{v(\bar{x})}{k(k+1)} + o(\frac{1}{k^2})$ for large k , where $v(\bar{x}) \equiv \frac{1}{2(1-\delta)} \sigma_k^2(\bar{x}) f(\bar{x})$. The result then follows. ■

Proof of Theorem 16: Since the optimal bid for advertiser i if advertiser i submits the highest eCPM bid amongst the bidders with unknown eCPMs satisfies $z_i = \bar{x}_i + \delta(E_{\bar{x}(i)}[V_{k(i)}(\bar{x}(i))] - V_k(\bar{x})) = 0$ when advertiser i submits the optimal bid z_i . By substituting this into the equation for the value of the dynamic programming problem $V_k(\bar{x})$, it follows that the value of this dynamic programming problem is always equal to $\int_{\bar{x}}^{z_i} F(y) dy$, which is an increasing function of z_i . From this it follows that if the optimal bid for advertiser i if advertiser i submits the highest bid of the advertisers with unknown eCPMs is higher than the optimal bid for all other advertisers with unknown eCPMs if one of these other advertisers submits the highest bid of the advertisers with unknown eCPMs, then the decision maker's

payoff from the game is maximized by having advertiser i submit the highest bid of all the advertisers with unknown eCPMs. ■

Proof of Theorem 17: Recall from Lemma 4 that the auctioneer’s per-period payoff if the auctioneer uses a bid for the advertiser with unknown eCPM that is equal to z is $-E_c[\int_{\bar{x}+o(\frac{1}{k})}^z F(z) - F(y)] dy = -\int_{\bar{x}}^z F(z) - F(y) dy - \frac{1}{2}\sigma_k^2 f(\bar{x}) + o(\sigma_k^2)$ for large k . Now if $z = \bar{x} + \frac{c(\bar{x})}{k\alpha}$ for some constant $c(\bar{x}) \neq 0$, then $\int_{\bar{x}}^z F(z) - F(y) dy = \int_{\bar{x}}^{\bar{x} + \frac{c(\bar{x})}{k\alpha}} f(\bar{x})(\bar{x} + \frac{c(\bar{x})}{k\alpha} - y) dy + o(\frac{1}{k\alpha}) = f(\bar{x})\frac{c(\bar{x})^2}{2k\alpha^2} + o(\frac{1}{k\alpha})$. Thus the auctioneer’s per-period payoff if the auctioneer uses a bid for the ad with unknown eCPM of the form $z = \bar{x} + \frac{c(\bar{x})}{k\alpha}$ is $-\frac{c(\bar{x})^2}{2k\alpha^2} f(\bar{x}) - \frac{1}{2}\sigma_k^2 f(\bar{x}) + o(\frac{1}{k\alpha})$ if $c(\bar{x}) \neq 0$ and $-\frac{1}{2}\sigma_k^2 f(\bar{x}) + o(\sigma_k^2)$ if $c(\bar{x}) = 0$.

Thus if $c(\bar{x}) = 0$, the auctioneer’s per-period payoff is $-\frac{1}{2k} s^2(\bar{x}) f(\bar{x}) + o(\frac{1}{k})$. We then know from similar reasoning to that in the proof of Theorem 10 that if this is the auctioneer’s per-period payoff, then the auctioneer’s total payoff from the game is $-\frac{1}{2(1-\delta)k} s^2(\bar{x}) f(\bar{x}) + o(\frac{1}{k})$ regardless of the learning rate. Similarly, if $c(\bar{x}) \neq 0$ and $\alpha = \frac{1}{2}$, then the auctioneer’s per-period payoff is $-\frac{1}{2k} f(\bar{x})(\sigma^2(\bar{x}) + c^2(\bar{x})) + o(\frac{1}{k})$, and we know from identical reasoning that the auctioneer’s total payoff is $-\frac{1}{2(1-\delta)k} f(\bar{x})(s^2(\bar{x}) + c^2(\bar{x})) + o(\frac{1}{k})$, which is strictly less than the auctioneer’s total payoff from the game when $c(\bar{x}) = 0$ for sufficiently large k .

Finally, if $c(\bar{x}) \neq 0$ and $\alpha < \frac{1}{2}$, the auctioneer’s per-period payoff is $-\frac{c(\bar{x})^2}{2k\alpha^2} f(\bar{x}) + o(\frac{1}{k\alpha})$. Since the auctioneer’s total payoff is the discounted sum of the auctioneer’s per-period payoffs, it follows that if $V_k(\bar{x})$ denotes the auctioneer’s total payoff from using this strategy, then $k^{2\alpha} V_k(\bar{x}) \leq \sum_{j=k}^{\infty} \delta^{j-k} [-\frac{1}{2}(\frac{k}{j})^{2\alpha} c^2(\bar{x}) f(\bar{x})] + o(1) = -\frac{1}{2(1-\delta)^{1-\alpha}} c^2(\bar{x}) f(\bar{x}) + o(1)$ in the limit as $k \rightarrow \infty$. Thus if $c(\bar{x}) \neq 0$ and $\alpha < \frac{1}{2}$, the auctioneer’s total payoff is no greater than $-\frac{1}{2(1-\delta)^{1-\alpha}} c^2(\bar{x}) f(\bar{x}) + o(\frac{1}{k\alpha})$, which is less than $-\frac{1}{2k} s^2(\bar{x}) f(\bar{x}) + o(\frac{1}{k})$, the auctioneer’s payoff from using the constant $c(\bar{x}) = 0$ for sufficiently large k . From this and the result in the previous paragraph it follows that if the auctioneer uses the strategy in the statement of this theorem, the auctioneer’s payoff will be maximized when $c(\bar{x}) = 0$ for sufficiently large k . ■

Observation 23 Suppose the auctioneer displays the ad with the highest eCPM bid with probability $1 - \epsilon$ and displays an ad uniformly at random with probability $\epsilon > 0$. Then the optimal constant ϵ for such an algorithm is $\epsilon = 0$ for sufficiently large k .

Proof Recall from Lemma 4 that the auctioneer’s per-period payoff if the auctioneer uses a bid for the advertiser with unknown eCPM that is equal to z is $-E_c[\int_{\bar{x}+o(\frac{1}{k})}^z F(z) - F(y)] dy = -\int_{\bar{x}}^z F(z) - F(y) dy - \frac{1}{2}\sigma_k^2 f(\bar{x}) + o(\sigma_k^2)$ for large k . Note that displaying an ad uniformly at random is equivalent to making a bid of 0 for the ad with unknown eCPM with probability $\frac{1}{2}$ and making a bid of ∞ for the ad with unknown eCPM with probability $\frac{1}{2}$. Since $\int_{\bar{x}}^{\infty} F(z) - F(y) dy > 0$ for either $z = 0$ or $z = \infty$, it follows that the auctioneer’s expected per-period payoff if the auctioneer follows the strategy in the statement of the observation is no greater than $-c\epsilon$ for some constant $c > 0$ for large k .

However, if the auctioneer always uses a bid of $z = \bar{x}$ (as would be the case when $\epsilon = 0$), then we know from the proof of Theorem 17 that the auctioneer’s per-period payoff is $-\frac{1}{2k} s^2(\bar{x}) f(\bar{x}) + o(\frac{1}{k})$ for large k . Thus for sufficiently large k , the auctioneer always

achieves a larger per-period payoff by setting $\epsilon = 0$ than by using any positive value of ϵ , so the optimal constant for this algorithm is $\epsilon = 0$. ■

Proof of Theorem 18: We know from Theorem 7 that $E_{\pi} [V_{k+1}(\bar{x})] - V_k(\bar{x}) = \frac{v(\bar{x})}{k(k+1)} + o(\frac{1}{k^2})$ for large k , where $v(\bar{x}) = \frac{1}{2(1-\delta)} s^2(\bar{x}) f(\bar{x})$, and we also know from the proof of Theorem 3 that the derivative of the seller’s expected payoff from making a bid of z with respect to z is $f(z)(\bar{x} - z + \delta(E_{\pi} [V_{k+1}(\bar{x})] - V_k(\bar{x})))$. Thus if $\Delta V \equiv E_{\pi} [V_{k+1}(\bar{x})] - V_k(\bar{x})$, then the difference between the auctioneer’s expected payoff from making a bid of \bar{x} and a bid of $\bar{x} + \frac{\delta}{2(1-\delta)k(k+1)} s^2(\bar{x}) f(\bar{x})$ is

$$\frac{1}{1-\delta} \int_{\bar{x}}^{\bar{x} + \delta \Delta V + o(\Delta V)} f(z)(\bar{x} - z + \delta(\Delta V) + o(\Delta V)) dz = \frac{f(\bar{x})\delta^2(\Delta V)^2}{2(1-\delta)} + o((\Delta V)^2).$$

And since $\Delta V = E_{\pi} [V_{k+1}(\bar{x})] - V_k(\bar{x}) = \frac{v(\bar{x})}{k(k+1)} + o(\frac{1}{k^2}) = \frac{s^2(\bar{x})f(\bar{x})}{2(1-\delta)k(k+1)} + o(\frac{1}{k^2})$, it follows that the difference between the auctioneer’s payoff from making a bid of \bar{x} and a bid of $\bar{x} + \frac{\delta}{2(1-\delta)k(k+1)} s^2(\bar{x}) f(\bar{x})$ is $\frac{\delta^2}{8(1-\delta)^3 k^4} s^4(\bar{x}) f^3(\bar{x}) + o(\frac{1}{k^4})$. ■

Proof of Theorem 19: The theoretically optimal strategy for the auctioneer would entail submitting a bid of $z = \bar{x} + \delta(E_{\pi} [V_{k+1}(\bar{x})] - V_k(\bar{x}))$ in each time period. By the same reasoning as in the proof of Theorem 18, we know the difference between the auctioneer’s expected payoff from making a bid of \bar{x} and making a bid of $z = \bar{x} + \delta(E_{\pi} [V_{k+1}(\bar{x})] - V_k(\bar{x}))$ is $\frac{f(\bar{x})\delta^2(\Delta V)^2}{2(1-\delta)} + o((\Delta V)^2)$. Since the auctioneer’s payoff from using the approximately optimal bidding strategy is also $\frac{f(\bar{x})\delta^2(\Delta V)^2}{2(1-\delta)} + o((\Delta V)^2)$, it follows that the difference between the auctioneer’s payoff under the approximately optimal bidding strategy and the maximum possible payoff the auctioneer could obtain theoretically is $o((\Delta V)^2) = o(\frac{1}{k^4})$.

But we know from Theorem 18 that the difference between the auctioneer’s payoff under the approximately optimal bidding strategy and the greedy strategy is $\frac{\delta^2}{8(1-\delta)^3 k^4} s^4(\bar{x}) f^2(\bar{x}) + o(\frac{1}{k^4})$. Thus the difference between the auctioneer’s payoff under this strategy and the maximum possible payoff the auctioneer could obtain under the theoretically optimal strategy becomes vanishingly small compared to the difference between the auctioneer’s payoff under this strategy and the auctioneer’s payoff under the greedy strategy for large k . ■

Proof of Theorem 20: A consequence of Theorem 13 is that the difference between the auctioneer’s payoff under the theoretically optimal strategy and the auctioneer’s payoff from the greedy strategy is no greater than the difference between the auctioneer’s payoff from the theoretically optimal strategy and the auctioneer’s payoff under the greedy strategy when no learning is possible in future periods. Thus we seek to bound the difference between the auctioneer’s payoff from the theoretically optimal strategy and the auctioneer’s payoff under the greedy strategy when no learning is possible in future periods.

Let α and β denote the parameters of the beta distribution. If no learning ever took place and the auctioneer followed the greedy strategy, then in all periods the auctioneer would show the highest competing bidder if this bidder had an eCPM bid p satisfying $p > \frac{\alpha+1}{\alpha+\beta}$ and show the bidder with unknown eCPM otherwise. If the auctioneer showed the ad with unknown eCPM in the first period, this ad received a click, and no learning

took place in future periods, then in future periods the auctioneer would show the highest competing bidder if and only if this bidder had an eCPM bid p satisfying $p > \frac{\alpha+1}{\alpha+\beta+1}$. And if the auctioneer showed the ad with unknown eCPM, this ad did not receive a click, and no learning took place in future periods, then in future periods the auctioneer would show the highest competing bidder if and only if this bidder had an eCPM bid p satisfying $p > \frac{\alpha}{\alpha+\beta+1}$.

From this it follows that if no learning takes place in future periods, then the auctioneer's payoff in any given period in the future is guaranteed to be the same regardless of whether the ad with unknown eCPM was shown in the first period if either $p > \frac{\alpha+1}{\alpha+\beta+1}$ or $p < \frac{\alpha}{\alpha+\beta+1}$ in that particular period. The only circumstances under which the auctioneer's expected payoff in a future period t will differ as a result of showing the ad with unknown eCPM in the first period is if this ad receives a click in the first period and $p \in (\frac{\alpha}{\alpha+\beta}, \frac{\alpha+1}{\alpha+\beta+1})$ in period t or if the ad does not receive a click in the first period and $p \in (\frac{\alpha}{\alpha+\beta+1}, \frac{\alpha}{\alpha+\beta})$ in period t . In the first case, the auctioneer's payoff in period t as a result of showing the ad with unknown eCPM in the first period exceeds the auctioneer's payoff under normal circumstances by $\frac{\alpha+1}{\alpha+\beta+1} - p$, and in the second case the auctioneer's payoff in period t as a result of showing the ad with unknown eCPM in the first period exceeds the auctioneer's payoff under normal circumstances by an amount $p - \frac{\alpha}{\alpha+\beta+1}$.

Now the probability the ad with unknown eCPM receives a click in the first period if this ad is shown is $\frac{\alpha}{\alpha+\beta}$ and the probability this ad does not receive a click in the first period if this ad is shown is $\frac{\beta}{\alpha+\beta}$. By combining this with the result in the previous paragraph, it follows that the maximum possible expected payoff difference that the auctioneer can obtain from future periods as a result of showing the ad with unknown eCPM in the first period is $\frac{\delta}{1-\delta} [\frac{\alpha}{\alpha+\beta} \int_{\frac{\alpha}{\alpha+\beta}}^{\frac{\alpha+1}{\alpha+\beta+1}} (\frac{\alpha+1}{\alpha+\beta+1} - p) \bar{f} dp + \frac{\beta}{\alpha+\beta} \int_{\frac{\alpha}{\alpha+\beta+1}}^{\frac{\alpha+\beta}{\alpha+\beta+1}} (p - \frac{\alpha}{\alpha+\beta+1}) \bar{f} dp]$, where $\bar{f} \equiv \sup_p f(p)$. This payoff difference equals $\frac{\delta \bar{f}}{2(1-\delta)} [\frac{\alpha}{\alpha+\beta} (\frac{\alpha+1}{\alpha+\beta+1} - \frac{\alpha}{\alpha+\beta})^2 + \frac{\beta}{\alpha+\beta} (\frac{\alpha}{\alpha+\beta+1})^2]$, which is $\frac{\delta \bar{f}}{2(1-\delta)} \frac{\alpha \beta}{(\alpha+\beta)^2 (\alpha+\beta+1)^2} = \frac{\delta \bar{f}}{2(1-\delta)} \frac{\alpha \beta}{(\alpha+\beta)^2 (\alpha+\beta+1)^2}$.

Now the expected value for a beta distribution is $\frac{\alpha}{\alpha+\beta}$ and the variance in a beta distribution is $\frac{\alpha \beta}{(\alpha+\beta)^2 (\alpha+\beta+1)}$. Thus since the bidder's expected eCPM is ω and the standard deviation in the bidder's expected eCPM is $\gamma \omega$, it follows that $\frac{\alpha}{\alpha+\beta} = \frac{1}{\omega}$, $\frac{\beta}{\alpha+\beta} = \frac{1}{\omega}$, and $\frac{\alpha \beta}{(\alpha+\beta)^2 (\alpha+\beta+1)^2} = \gamma^4 \omega^4$. From this it follows that $\frac{\delta \bar{f}}{2(1-\delta)} \frac{\alpha \beta}{(\alpha+\beta)^2 (\alpha+\beta+1)^2} = \frac{\delta \bar{f}}{2(1-\delta)} \frac{\gamma^4 \omega^4}{1-\omega}$. Thus the maximum additional payoff increase that one can obtain from future periods as a result of showing the ad with unknown eCPM in the first period is no greater than $\frac{\delta \bar{f}}{2(1-\delta)} \frac{\gamma^4 \omega^4}{1-\omega}$.

Now if ΔV denotes the change in payoff that one obtains from future periods as a result of showing the ad with uncertain eCPM in the first period, then the value of showing the ad with uncertain eCPM in the first period is $\omega + \Delta V$. Thus the theoretically optimal strategy will specify a bid of $\omega + \Delta V$ for the bidder with uncertain eCPM, whereas the greedy strategy will specify a bid of ω , so the theoretically optimal strategy will only show a different ad when the highest competing eCPM bid, p , satisfies $p \in [\omega, \omega + \Delta V]$. Furthermore, in the cases where the theoretically optimal strategy specifies a different bid, the theoretically optimal strategy achieves a payoff that exceeds that of the greedy strategy by an amount $\omega + \Delta V - p$, where p denotes the highest competing eCPM bid. From this it follows that the

difference in expected payoff that one obtains as a result of using the theoretically optimal strategy rather than the greedy strategy is no greater than $\frac{1}{1-\delta} \int_{\omega}^{\omega+\Delta V} (\omega + \Delta V - p) f(p) dp$.

Thus if $\bar{f} \equiv \sup_p f(p)$, this payoff difference is no greater than $\frac{1}{1-\delta} \int_{\omega}^{\omega+\Delta V} (\omega + \Delta V - p) dp = \frac{\bar{f} (\Delta V)^2}{1-\delta}$. Thus the difference between the auctioneer's payoff under the theoretically optimal strategy and the auctioneer's payoff from the greedy strategy is no greater than $\frac{\bar{f} (\Delta V)^2}{1-\delta}$.

But we have seen earlier that the maximum additional payoff increase that one can obtain from future periods as a result of showing the ad with uncertain eCPM in the first period is no greater than $\frac{\delta \bar{f}}{2(1-\delta)} \frac{\gamma^4 \omega^4}{1-\omega}$. Thus we know that $\Delta V \leq \frac{\delta \bar{f}}{2(1-\delta)} \frac{\gamma^4 \omega^4}{1-\omega}$. By combining this with the result in the previous paragraph, we see that the difference between the maximum possible payoff the auctioneer could obtain under the theoretically optimal strategy and the auctioneer's payoff from the greedy strategy is no greater than $\frac{\delta^2 \gamma^4 \omega^4 \bar{f}^3}{8(1-\delta)^3 (1-\omega)^2}$. ■

References

- D. Agarwal, B.C. Chen, and P. Elango. Explore/exploit schemes for web content optimization. In *Proceedings of the 9th Industrial Conference on Data Mining (ICDM)*, pages 1-10. IEEE, 2009.
- P. Aghion, P. Bolton, C. Harris, and B. Jullien. Optimal learning by experimentation. *Review of Economic Studies*, 58(4):621-654, 1991.
- P. Aghion, M.P. Espinosa, and B. Jullien. Dynamic duopoly with learning through market experimentation. *Economic Theory*, 3(3):517-539, 1993.
- N. Anthonisen. On learning to cooperate. *Journal of Economic Theory*, 107(2):253-287, 1993.
- N. Anthonisen. Regret bounds and minimax policies under partial monitoring. *Journal of Machine Learning Research*, 11:2785-2836, 2010.
- P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multi-armed bandit problem. *Machine Learning*, 47(2-3):235-256, 2002.
- P. Auer, N. Cesa-Bianchi, and P. Fischer. The nonstochastic multi-armed bandit problem. *SIAM Journal on Computing*, 32(1):48-77, 2003.
- M. Babaioff, Y. Sharma, and A. Slivkins. Characterizing truthful multi-armed bandit mechanisms. In *Proceedings of the 10th ACM Conference on Electronic Commerce (EC)*, pages 79-88. ACM, 2009.
- A. Banerjee and D. Fudenberg. Word-of-mouth learning. *Games and Economic Behavior*, 46(1):1-22, 2004.
- J.S. Banks and R.K. Sundaram. Denumerable-armed bandits. *Econometrica*, 60(5):1071-1096, 2004.

- E. Bax, A. Kuratti, P. McAfee, and J. Romero. Comparing predicted prices in auctions for online advertising. *International Journal of Industrial Organization*, 30(1):80-88, 2011.
- D. Bergemann and J. Välimäki. Experimentation in markets. *Review of Economic Studies* 67(2):213-234, 2000.
- D. Bergemann and J. Välimäki. Learning and strategic pricing. *Econometrica*, 64(5):1125-1149, 1996.
- D. Bergemann and J. Välimäki. Market diffusion with two-sided learning. *RAND Journal of Economics* 28(4):773-795, 1997.
- D. Bergemann and J. Välimäki. Stationary multi-choice bandit problems. *Journal of Economic Dynamics and Control* 25(1):1585-1594, 2001.
- P. Bolton and C. Harris. Strategic experimentation. *Econometrica* 67(2):349-374, 1999.
- M. Brezzi and T.L. Lai. Optimal learning and experimentation in bandit problems. *Journal of Economic Dynamics and Control* 27(1):87-108, 2002.
- S. Callander. Searching for good policies. *American Political Science Review* 105(4):643-662, 2011.
- S. Callander and P. Hummel. Preemptive policy experimentation. *Econometrica* 82(4):1509-1528, 2014.
- S.E. Chick and N. Gans. Economic analysis of simulation selection problems. *Management Science* 55(3):421-437, 2009.
- N.R. Devanur and S.M. Kulkade. The price of truthfulness for pay-per-click auctions. In *Proceedings of the 10th ACM Conference on Electronic Commerce (EC)*, pages 99-106. ACM, 2009.
- A. Fishman and R. Rob. Experimentation and competition. *Journal of Economic Theory* 78(2):299-320, 1998.
- P. Frazier, W. Powell, and S. Dayanik. The knowledge-gradient policy for correlated normal beliefs. *INFORMS Journal on Computing* 21(4):599-613, 2009.
- D. Gale. What have we learned from social learning? *European Economic Review* 40(3-5):617-628, 2011.
- D. Gale and R.W. Rosenthal. Experimentation, imitation, and stochastic stability. *Journal of Economic Theory* 84(1):1-40, 2011.
- A. Ghate. Optimal minimum bids and inventory scrapping in sequential, single-unit, Vickrey auctions with demand learning. *European Journal of Operations Research* 245(2):555-570, 2015.
- J.C. Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society, Series B* 41(2):148-177, 1979.
- E. Hazan and S. Kale. Better algorithms for benign bandits. *Journal of Machine Learning Research* 12:1287-1311, 2011.
- K. Iyer, R. Johari, and M. Sundararajan. Mean field equilibria of dynamic auctions with learning. *Management Science* 60(12):2949-2970, 2014.
- S.M. Kulkade, I. Lohel, and H. Nazerzadeh. Optimal dynamic mechanism design and the virtual pivot mechanism. *Operations Research* 61(4):837-854, 2013.
- G. Keller and S. Rady. Optimal experimentation in a changing environment. *Review of Economic Studies* 66(3):475-503, 1999.
- G. Keller and S. Rady. Strategic experimentation with Poisson bandits. *Theoretical Economics* 5(2):275-311, 2010.
- G. Keller, S. Rady, and M. Cripps. Strategic experimentation with exponential bandits. *Econometrica* 73(1):39-68, 2010.
- S. Lahaie and R.P. McAfee. Efficient ranking in sponsored search. In *Proceedings of the 7th International Workshop on Internet and Network Economics (WINE)*, pages 254-265. Springer, 2011.
- T.L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics* 6:4-22, 1985.
- S.M. Li, M. Mahdian, and R.P. McAfee. Value of learning in sponsored search auctions. In *Proceedings of the 6th International Workshop on Internet and Network Economics (WINE)*, pages 294-305. Springer, 2010.
- S. Mannor and J.N. Tsiitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research* 5:623-648, 2004.
- B.C. May, N. Korda, A. Lee, and D.S. Leslie. Optimistic Bayesian sampling in contextual-bandit problems. *Journal of Machine Learning Research* 13(1): 2069-2106, 2012.
- R.P. McAfee. The design of advertising exchanges. *Review of Industrial Organization* 39(3):169-185, 2011.
- L.J. Mirman, L. Samuelson, and A. Urbano. Monopoly experimentation. *International Economic Review* 34(3):549-563, 1993.
- G. Moscarini and L. Smith. The optimal level of experimentation. *Econometrica* 69(6):1629-1644, 2001.
- M. Ostrovsky and M. Schwarz. Reserve prices in Internet advertising auctions: a field experiment. Stanford University Typescript, 2009.
- A. Pavan, I. Segal and J. Toikka. Dynamic mechanism design: a Myersonian approach. *Econometrica* 82(2):601-653, 2014.

- M. Rothschild. A two-armed bandit theory of market pricing. *Journal of Economic Theory* 9(2):185-202, 1974.
- A. Rusitcchini and A. Wolinsky. Learning about variable demand in the long run. *Journal of Economic Dynamics and Control* 19(5-7):1283-1292, 1995.
- I.O. Ryzhov, P.I. Frazier, and W.B. Powell. On the robustness of a one-period look-ahead policy in multi-armed bandit problems. *Procedia Computer Science* 1:1629-1639, 2010.
- I.O. Ryzhov, W.B. Powell, and P.I. Frazier. The knowledge gradient algorithm for a general class of online learning problems. *Operations Research* 60(1):180-195, 2012.
- K.H. Schlag. Why imitate, and if so, how? A boundedly rational approach to multi-armed bandits. *Journal of Economic Theory* 78(1):130-156, 1998.
- A. Slivkins. Contextual bandits with similarity information. *Journal of Machine Learning Research* 15(1):2533-2568, 2014.
- B. Strulovici. Learning while voting: determinant of collective experimentation. *Econometrica* 78(3):933-971, 2010.
- H.R. Varian. Online ad auctions. *American Economic Review: Papers & Proceedings* 99(2):430-434, 2009.
- X. Vives. Learning from others: a welfare analysis. *Games and Economic Behavior* 20(2):177-200, 1997.
- M.L. Weitzman. Optimal search for the best alternative. *Econometrica* 47(3):641-654, 1979.
- J. Wortman, Y. Vorobeychik, L. Li, and J. Langford. Maintaining equilibria during exploration in sponsored search auctions. In *Proceedings of the 3rd International Workshop on Internet and Network Economics (WINE)*, pages 119-130. Springer, 2007.

Wavelet decompositions of Random Forests - smoothness analysis, sparse approximation and applications

Oren Elisha *School of Mathematical Sciences
University of Tel-Aviv
and GE Global Research
Israel*

Shai Dekel *School of Mathematical Sciences
University of Tel-Aviv
and GE Global Research
Israel*

Editors: Lawrence Carin

Abstract

In this paper we introduce, in the setting of machine learning, a generalization of wavelet analysis which is a popular approach to low dimensional structured signal analysis. The wavelet decomposition of a Random Forest provides a sparse approximation of any regression or classification high dimensional function at various levels of detail, with a concrete ordering of the Random Forest nodes: from ‘significant’ elements to nodes capturing only ‘insignificant’ noise. Motivated by function space theory, we use the wavelet decomposition to compute numerically a ‘weak-type’ smoothness index that captures the complexity of the underlying function. As we show through extensive experimentation, this sparse representation facilitates a variety of applications such as improved regression for difficult datasets, a novel approach to feature importance, resilience to noisy or irrelevant features, compression of ensembles, etc.

Keywords: Random Forest, Wavelets, Besov spaces, adaptive approximation, feature importance.

1. Introduction

Our work brings together Function Space theory, Harmonic Analysis and Machine Learning for the analysis of high dimensional big data. In the field of (low-dimensional) signal processing, there is a complete theory that models structured datasets (e.g. audio, images, video) as functions in certain Besov spaces (DeVore 1998), (DeVore et al. 1992). When representing the signal using time-frequency localized dictionaries, this theory characterizes

the performance of adaptive approximation and is used in a variety of applications, such as denoising, compression, feature extraction, etc. using very simple algorithms.

The first contribution of this work is a construction of wavelet decomposition of Random Forests (Breiman 2001), (Biau and Scornet 2016), (Denil et al. 2014). Wavelets (Daubechies 1992), (Mallat 2009) and geometric wavelets (Dekel and Leviatan 2005), (Alani et al. 2007), (Dekel and Gershtansky 2012), are a powerful yet simple tool for constructing sparse representations of ‘complex’ functions. The Random Forest (RF) (Biau and Scornet 2016), (Criminisi et al. 2011), (Hastie et al. 2009) introduced by Breiman (Breiman 2001), (Breiman 1996), is a very effective machine learning method that can be considered as a way to overcome the ‘greedy’ nature and high variance of a single decision tree. When combined, the wavelet decomposition of the RF unravels the sparsity of the underlying function and establishes an order of the RF nodes from ‘important’ components to ‘negligible’ noise. Therefore, the method provides a better understanding of any constructed RF. This helps to avoid over-fitting in certain scenarios (e.g. small number of trees), to remove noise or provide compression. Our approach could also be considered as an alternative method for pruning of ensembles (Chen et al. 2009), (Kulkarni and Sinha 2012), (Yang et al. 2012), (Joly et al. 2012) where the most important decision nodes of a huge and complex ensemble of models can be quickly and efficiently extracted. Thus, instead of controlling complexity by restricting trees’ depth or node size, one controls complexity through adaptive wavelet approximation.

Our second contribution is to generalize the function space characterization of adaptive algorithms (DeVore 1998), (DeVore and Lorentz 1993), to a typical machine learning setup. Using the wavelet decomposition of a RF, we can actually numerically compute a ‘weak-type’ smoothness index of the underlying regression or classification function overcoming noise. We prove the first part of the characterization and demonstrate, using several examples, the correspondence between the smoothness of the underlying function and properties such as compression.

Applying a ‘wavelet-type’ machinery for learning tasks, using ‘Treelets’, was introduced by (Lee et al. 2008). Treelets provide a decomposition of the domain into localized basis functions that enable a sparse representation of smooth signals. This method performs a bottom-up construction in the feature space, where at each step, a local PCA among two correlated variables generates a new node in a tree. Our method is different, since for supervised learning tasks, the response variable should be used during the construction of the adaptive representation. Also, our work significantly improves upon the ‘wavelet-type’ construction of (Gavish et al. 2010). First, since our wavelet decomposition is built on the solid foundations of RFs, it leverages on the well-known fact that over-complete representations/ensembles outperform the critical sampled representations/single decision trees

in problems such as regression, estimation, etc. Secondly, from the theoretical perspective, the Lipschitz space analysis of (Gavish et. al. 2010) is generalized by our Besov space analysis, which is the right mathematical setup for adaptive approximation using wavelets.

The paper is organized as follows: In Section 2 we review Random Forests. In Section 3 we present our main wavelet construction and list some of its key properties. In section 4 we present some theoretical aspects of function space theory and its connection to sparsity. This characterization quantifies the sparsity of the data with respect to the response variable. In Section 5 we review how a novel form of Variable Importance (VI) is computed using our approach. Section 6 provides extensive experimental results that demonstrate the applicative added value of our method in terms of regression, classification, compression and variable importance quantification.

2. Overview of Random Forests

We begin with an overview of single trees. In statistics and machine learning (Breiman et. al. 1984), (Alpaydm 2004), (Bian and Scornet 2016), (Denil et. al. 2014), (Hastie et. al. 2009) the construction is called a Decision Tree or the Classification and Regression Tree (CART) while in image processing and computer graphics (Radha et. al. 1996), (Salembier and Garrido 2000) it is coined as the Binary Space Partition (BSP) tree. We are given a real-valued function $f \in L_2(\Omega_0)$ or a discrete dataset $\{x_i \in \Omega_0, f(x_i)\}_{i \in I}$ in some convex bounded domain $\Omega_0 \subset \mathbb{R}^n$. The goal is to find an efficient representation of the underlying function, overcoming the complexity, geometry and possibly non-smooth nature of the function values. To this end, we subdivide the initial domain Ω_0 into two subdomains, e.g. by intersecting it with a hyper-plane. The subdivision is performed to minimize a given cost function. This subdivision process then continues recursively on the subdomains until some stopping criterion is met, which in turn, determines the leaves of the tree. We now describe one instance of the cost function which is related to minimizing variance. At each stage of the subdivision process, at a certain node of the tree, the algorithm finds, for the convex domain $\Omega \subset \mathbb{R}^n$ associated with the node:

- (i) A partition by a hyper-plane into two convex subdomains Ω', Ω'' (see Figure 1),
- (ii) Two multivariate polynomials $Q_{\Omega'}, Q_{\Omega''} \in \Pi_{r-1}(\mathbb{R}^n)$, of fixed (typically low) total degree $r - 1$.

The subdomains and the polynomials are chosen to minimize the following quantity

$$\|f - Q_{\Omega'}\|_{L_p(\Omega')}^p + \|f - Q_{\Omega''}\|_{L_p(\Omega'')}^p, \quad \Omega' \cup \Omega'' = \Omega. \quad (1)$$

Here, for $1 \leq p < \infty$, we used the definition

$$\|g\|_{L_p(\Omega)} := \left(\int_{\Omega} |g(x)|^p dx \right)^{1/p},$$

If the dataset is discrete, consisting of feature vectors $x_i \in \mathbb{R}^n, i \in I$, with response values $f(x_i)$, then a discrete functional is minimized

$$\sum_{x_i \in \Omega'} |f(x_i) - Q_{\Omega'}(x_i)|^p + \sum_{x_i \in \Omega''} |f(x_i) - Q_{\Omega''}(x_i)|^p, \quad \Omega' \cup \Omega'' = \Omega. \quad (2)$$

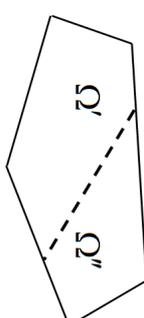


Figure 1: Illustration of a subdivision by an hyperplane of a parent domain Ω into two children Ω', Ω'' .

Observe that for any given subdividing hyperplane, the approximating polynomials in (2) can be uniquely determined for $p = 2$ by least square minimization (see (Avery)) for a survey of local polynomial regression). For the order $r = 1$, the approximating polynomials are nothing but the mean of the function values over each of the subdomains

$$Q_{\Omega'}(x) = C_{\Omega'} = \frac{1}{\#\{x_i \in \Omega'\}} \sum_{x_i \in \Omega'} f(x_i), \quad Q_{\Omega''}(x) = C_{\Omega''} = \frac{1}{\#\{x_i \in \Omega''\}} \sum_{x_i \in \Omega''} f(x_i). \quad (3)$$

In many applications of decision trees, the high-dimensionality of the data does not allow to search through all possible subdivisions. As in our experimental results, one may restrict the subdivisions to the class of hyperplanes aligned with the main axes. In contrast, there are cases where one would like to consider more advanced form of subdivisions, where they take certain hyper-surface form, such as conic-sections. Our paradigm of wavelet decompositions can support in principle all of these forms.

Random Forest (RF) is a popular machine learning tool that collects decision trees into an ensemble model (Breiman 2001), (Bernard et. al 2012), (Bian 2012), (Bian and Scornet 2016). The trees are constructed independently in a diverse fashion and prediction is done by a voting mechanism among all trees. A key element (Breiman 2001), is that large diversity between the trees reduces the ensemble's variance. There are many RFs variations

that differ in the way randomness is injected into the model, e.g bagging, random feature subset selection and the partition criterion (Boulesteix et. al. 2012), (Criminisi et. al. 2011), (Hastie et. al. 2009). Our wavelet decomposition paradigm is applicable to most of the RF versions known from the literature.

Bagging (Breiman 1996) is a method that produces partial replicates of the training data for each tree. A typical approach is to randomly select for each tree a certain percentage of the training set (e.g. 80%) or to randomly select samples with repetitions (Hastie et. al. 2009). From an approximation theoretical perspective, this form of RF allows to create an over-complete representation (Christensen 2002) of the underlying function that overcomes the ‘greedy’ nature of a single tree .

Additional methods to inject randomness can be achieved at the node partitioning level. For each node, we may restrict the partition criteria to a small random subset of the parameter values (hyper-parameter). A typical selection is to search for a partition from a random subset of \sqrt{n} features (Breiman 2001). This technique is also useful for reducing the amount of computations when searching the appropriate partition for each node. Bagging and random feature selections are not mutually exclusive and could be used together.

For $j = 1, \dots, J$, one creates a decision tree \mathcal{T}_j , based on a subset of the data, X^j . One then provides a weight (score) w_j to the tree \mathcal{T}_j , based on the estimated performance of the tree. In the supervised learning, one typically uses the remaining data points $x_i \notin X^j$ to evaluate the performance of \mathcal{T}_j . We note that for any point $x \in \Omega_0$, the approximation associated with the j^{th} tree, denoted by $\tilde{f}_j(x)$, is computed by finding the leaf $\Omega \in \mathcal{T}_j$ in which x is contained and then evaluating $\tilde{f}_j(x_i) := Q_\Omega(x)$, where Q_Ω is the corresponding polynomial associated with the decision node Ω . One then assigns a weight $w_j > 0$ to each tree \mathcal{T}_j , such that $\sum_{j=1}^J w_j = 1$. For simplicity, we will mostly consider in this paper the choice of uniform weights $w_j = 1/J$. One then assigns a value to any point $x \in \Omega_0$ by

$$\tilde{f}(x) = \sum_{j=1}^J w_j \tilde{f}_j(x).$$

Typically, in classification problems, the response variables does not have a numeric value, but rather are labeled by one of L classes. In this scenario, each input training point $x_i \in \mathbb{R}^n$ is assigned with a class $Cl(x_i)$. To convert the problem to the ‘functional’ setting described above one assigns to each class Cl the value of a node on the regular simplex consisting of L vertices in \mathbb{R}^{L-1} (all with equal pairwise distances). Thus, we may assume that the input data is in the form

$$\{x_i, Cl(x_i)\}_{i \in I} \in (\mathbb{R}^n, \mathbb{R}^{L-1}).$$

In this case, if we choose approximation using constants ($r = 1$), then the calculated mean over any subdomain Ω is in fact a point $\tilde{E}_\Omega \in \mathbb{R}^{L-1}$, inside the simplex. Obviously, any value inside the multidimensional simplex, can be mapped back to a class, along with an estimated certainty level, by calculating the closest vertex of the simplex to it. As will become obvious, these mappings can be applied to any wavelet approximation of functions receiving multidimensional values in the simplex.

3. Wavelet decomposition of a random forest

In some applications, there is a need to understand which nodes of a forest encapsulate more information than the others. Furthermore, in the presence of noise, one popular approach is to limit the levels of the tree, so as not to over-fit and contaminate the decisions by noise. Following the classic paradigm of nonlinear approximation using wavelets (Daubechies 1992), (DeVore 1998), (Mallat 2009) and the geometric function space theory presented in (Dekel and Leviatan 2005), (Karaivanov and Petrushev 2003), we present a construction of a wavelet decomposition of a forest. Some aspects of the theoretical justification for the construction are covered in the next section. Let Ω' be a child of Ω in a tree \mathcal{T} , i.e. $\Omega' \subset \Omega$ and Ω' was created by a partition of Ω as in Figure 1. Denote by $\mathbf{1}_{\Omega'}$, the indicator function over the child domain Ω' , i.e. $\mathbf{1}_{\Omega'}(x) = 1$, if $x \in \Omega'$ and $\mathbf{1}_{\Omega'}(x) = 0$, if $x \notin \Omega'$. We use the polynomial approximations $Q_{\Omega'}, Q_\Omega \in \Pi_{r-1}(\mathbb{R}^n)$, computed by the local minimization (1) and define

$$\psi_{\Omega'} := \psi_{\Omega'}(f) := \mathbf{1}_{\Omega'}(Q_{\Omega'} - Q_\Omega), \quad (4)$$

as the **geometric wavelet** associated with the subdomain Ω' and the function f , or the given discrete dataset $\{x_i, f(x_i)\}_{i \in I}$. Each wavelet $\psi_{\Omega'}$, is a ‘local difference’ component that belongs to the detail space between two levels in the tree, a ‘low resolution’ level associated with Ω and a ‘high resolution’ level associated with Ω' . Also, the wavelets (4) have the ‘zero moments’ property, i.e., if the response variable is sampled from a polynomial of degree $r-1$ over Ω , then our local scheme will compute $Q_{\Omega'}(x) = Q_\Omega(x) = f(x)$, $\forall x \in \Omega$, and therefore $\psi_{\Omega'} = 0$.

Under certain mild conditions on the tree \mathcal{T} and the function f , we have by the nature of the wavelets, the ‘telescopic’ sum of differences

$$f = \sum_{\Omega \in \mathcal{T}} \psi_\Omega, \quad \psi_{\Omega_0} := Q_{\Omega_0}. \quad (5)$$

For example, (5) holds in L_p -sense, $1 \leq p < \infty$, if $f \in L_p(\Omega_0)$ and for any $x \in \Omega_0$ and series of domains $\Omega_l \in \mathcal{T}$, each on a level l with $x \in \Omega_l$, we have that $\lim_{l \rightarrow \infty} \text{diam}(\Omega_l) = 0$.

In the setting of a real-valued function, the norm of a wavelet is computed by

$$\|\psi_{\Omega'}\|_2^2 = \int_{\Omega'} (Q_{\Omega'}(x) - Q_{\Omega}(x))^2 dx,$$

and in the discrete case by,

$$\|\psi_{\Omega'}\|_2^2 = \sum_{x_i \in \Omega'} |Q_{\Omega'}(x_i) - Q_{\Omega}(x_i)|^2, \quad (6)$$

where Ω' is a child of Ω .

Observe that for $r = 1$, the subdivision process for partitioning a node by minimizing (1) is equivalent to maximizing the sum of squared norms of the wavelets that are formed in that partition

Lemma 1 For any partition $\Omega = \Omega' \cup \Omega''$ denote

$$V_{\Omega} := \sum_{x_i \in \Omega'} |f(x_i) - C_{\Omega'}|^2 + \sum_{x_i \in \Omega''} |f(x_i) - C_{\Omega''}|^2,$$

where $C_{\Omega'}$, $C_{\Omega''}$ are defined in (3) and

$$W_{\Omega} := \|\psi_{\Omega'}\|_2^2 + \|\psi_{\Omega''}\|_2^2.$$

Then, the minimization (2) of V_{Ω} is equivalent to maximization of W_{Ω} over all choices of subdomains Ω' , Ω'' , $\Omega = \Omega' \cup \Omega''$ and constants $C_{\Omega'}$, $C_{\Omega''}$.

Proof See Appendix.

Recall that our approach is to convert classification problems into a ‘functional’ setting by assigning the L class labels to vertices of a simplex in \mathbb{R}^{L-1} . In such cases of multi-valued functions, choosing $r = 1$, the wavelet $\psi_{\Omega'}$: $\mathbb{R}^n \rightarrow \mathbb{R}^{L-1}$ is

$$\psi_{\Omega'} = \mathbf{1}_{\Omega'} (\bar{E}_{\Omega'} - \bar{E}_{\Omega}),$$

and its norm is given by

$$\|\psi_{\Omega'}\|_2^2 = \sum_{x_i \in \Omega'} \|\bar{E}_{\Omega'} - \bar{E}_{\Omega}\|_{l_2}^2 = \|\bar{E}_{\Omega'} - \bar{E}_{\Omega}\|_{l_2}^2 \# \{x_i \in \Omega'\}, \quad (7)$$

where for $\vec{v} \in \mathbb{R}^{L-1}$, $\|\vec{v}\|_{l_2} := \sqrt{\sum_{i=1}^{L-1} v_i^2}$.

Using any given weights assigned to the trees, we obtain a wavelet representation of the entire RF

$$\tilde{f}(x) = \sum_{j=1}^J \sum_{\Omega \in \mathcal{T}_j} w_j \psi_{\Omega}(x). \quad (8)$$

The theory (see Theorem 4 below) tells us that sparse approximation is achieved by ordering the wavelet components based on their norm

$$w_{f(\Omega_{k_1})} \|\psi_{\Omega_{k_1}}\|_2 \geq w_{f(\Omega_{k_2})} \|\psi_{\Omega_{k_2}}\|_2 \geq w_{f(\Omega_{k_3})} \|\psi_{\Omega_{k_3}}\|_2 \dots \quad (9)$$

with the notation $\Omega \in \mathcal{T}_j \Rightarrow j(\Omega) = j$. Thus, the adaptive M-term approximation of a RF is

$$f_M(x) := \sum_{m=1}^M w_{f(\Omega_{k_m})} \psi_{\Omega_{k_m}}(x). \quad (10)$$

Observe that, contrary to existing tree pruning techniques, where each tree is pruned separately, the above approximation process applies a ‘global’ pruning strategy where the significant components can come from any node of any of the trees at any level. For simplicity, one could choose $w_j = 1/j$, and obtain

$$f_M(x) = \frac{1}{J} \sum_{m=1}^M \psi_{\Omega_{k_m}}(x). \quad (11)$$

Figure 2 below depicts an M-term (11) selected from an RF ensemble. The red colored nodes illustrate the selection of the M wavelets with the highest norm values from the entire forest. Observe that they can be selected from any tree at any level, with no connectivity restrictions.

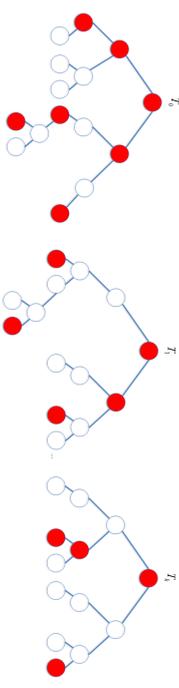


Figure 2: Selection of an M-term approximation from the entire forest.

Figure 3 depicts how the parameter M is selected for the challenging ‘Red Wine Quality’ dataset from the UCI repository (UCI repository). The generation of 10 decision trees on the training set creates approximately 3500 wavelets. The parameter M is then selected by minimization of the approximation error on a validation set. In contrast with other pruning

methods (Loh 2011), using (9), the wavelet approximation method may select significant components from any tree and any level in the forest. By this method, one does not need to predetermine the maximal depth of the trees and over-fitting is controlled by the selection of significant wavelet components.

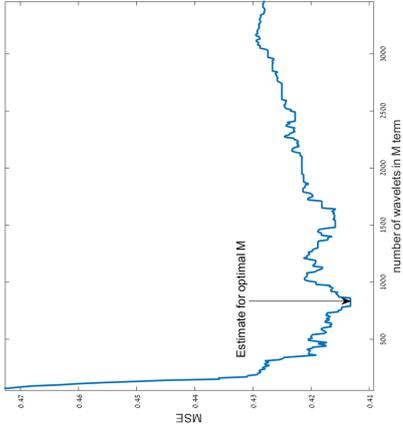


Figure 3: “Red Wine Quality” dataset - Numeric computation of M for optimal regression.

In a similar manner to certain successful applications in signal processing (e.g. coefficient quantization in the image compression standard JPEG), one may replace the selection of the parameter M in (11), with a threshold parameter $\varepsilon > 0$, chosen suitably for the problem (see for example Section 6.2). One then creates a wavelet approximation using all wavelet terms with norm (6) greater than ε .

In some cases, as presented in (Strobl et. al. 2006) explanatory attributes may be non-descriptive and even noisy, leading to the creation of problematic nodes in the decision trees. Nevertheless, in these cases, the corresponding wavelet norms are controlled and these nodes can be pruned out of the sparse representation (11). The following example demonstrates exactly this, that with high probability, the wavelets associated with the correct variables have relatively higher norms than wavelets associated with non-descriptive variables. Hence the wavelet based criterion will choose, with high probability the correct variable.

Example 1 Let $\{y_i\}_{i=1}^m$, where $y_i \sim \text{Ber}(1/2)$ i.i.d. and $\{x_i\}_{i=1}^m \subset [0, 1]^n$, $x_i = (x_{i1}, \dots, x_{ik}, \dots, x_{in}) \in \mathbb{R}^n$ with $x_{ik} = y_i$ and x_{ij} , $j \neq k$, uniformly distributed in $[0, 1]$. Then, for a subdivision along the j th axis, $[0, 1]^n = \Omega' \cup \Omega''$, and given $\delta \in (0, 1)$, w.p. $\geq 1 - \delta$,

1. If $j \neq k$, then $\|\psi_{\Omega'}\|_2^2, \|\psi_{\Omega''}\|_2^2 \leq 2 \log(2/\delta)$,
2. If $j = k$ and the subdivision minimizes (1), then

$$\|\psi_{\Omega'}\|_2^2, \|\psi_{\Omega''}\|_2^2 \geq \left(\frac{m}{2} - \sqrt{\frac{\log(2/\delta)}{2m}} \right)^3 / m^2.$$

Proof See Appendix.

4. ‘Weak-Type’ Smoothness and Sparse Representations of the response variable

In this section, we generalize to unstructured and possibly high dimensional datasets, a theoretical framework that has been applied in the context of signal processing, where the data is well structured and of low dimension (DeVore 1998), (DeVore et. al. 1992). The ‘sparsity’ of a function in some representation is an important property that provides a robust computational framework (Elad 2010(©)). Approximation Theory relates the sparsity of a function to its Besov smoothness index and supports cases where the function is not even continuous. Our motivation is to provide additional tools that can be used in the context of machine learning to associate a Besov-index, which is roughly a ‘complexity’ score, to the underlying function of a dataset. As the theory below and the experimental results show, this index correlates well with the performance of RFs and wavelet decompositions of RFs.

For a function $f \in L_r(\Omega)$, $0 < r \leq \infty$, $h \in \mathbb{R}^n$ and $r \in \mathbb{N}$, we recall the r -th order difference operator

$$\Delta_h^r(f, x) := \Delta_h^r(f, \Omega, x) := \begin{cases} \sum_{k=0}^r (-1)^{r+k} \binom{r}{k} f(x + kh) & [x, x + rh] \subset \Omega, \\ 0 & \text{otherwise,} \end{cases}$$

where $[x, y]$ denotes the line segment connecting any two points $x, y \in \mathbb{R}^n$. The **modulus of smoothness of order r** over Ω is defined by

$$\omega_r(f, t)_\tau := \sup_{|h| \leq t} \|\Delta_h^r(f, \Omega, \cdot)\|_{L_r(\Omega)}, \quad t > 0,$$

where for $h \in \mathbb{R}^n$, $\|h\|$ denotes the norm of h . We also denote

$$\omega_r(f, \Omega)_\tau := \omega_r\left(f, \frac{\text{diam}(\Omega)}{\tau}\right)_\tau.$$

Definition 2 For $0 < p < \infty$ and $\alpha > 0$, we set $\tau = \tau(\alpha, p)$, to be $1/\tau := \alpha + 1/p$. For a given function $f \in L_p(\Omega_0)$, $\Omega_0 \subset \mathbb{R}^n$, and tree \mathcal{T} , we define the associated B -space smoothness in $B_p^{\alpha, \tau}(\mathcal{T})$, $r \in \mathbb{N}$, by

$$|f|_{B_p^{\alpha, \tau}(\mathcal{T})} := \left(\sum_{\Omega \in \mathcal{T}} (|\Omega|^{-\alpha} \omega_r(f, \Omega)_r)^\tau \right)^{1/\tau}, \quad (12)$$

where, $|\Omega|$ denotes the volume of Ω .

We now show that a ‘well clustered’ function is in fact infinitely smooth in the right adaptively chosen Besov space.

Lemma 3 Let $f(x) = \sum_{k=1}^K P_k(x) \mathbf{1}_{B_k}(x)$, where each $B_k \subset \Omega_0$ is a box with sides parallel to the main axes and $P_k \in \Pi_{r-1}$. We further assume that $B_k \cap B_j = \emptyset$, whenever $j \neq k$. Then, there exists an adaptive tree partition \mathcal{T} , such that $f \in B_p^{\alpha, \tau}(\mathcal{T})$, for any $\alpha > 0$.

Proof See Appendix.

For a given forest $\mathcal{F} = \{\mathcal{T}_j\}_{j=1}^J$ and weights $w_j = 1/J$, the α Besov semi-norm associated with the forest is

$$|f|_{B_p^{\alpha, \tau}(\mathcal{F})} := \frac{1}{J} \left(\sum_{j=1}^J |f|_{B_p^{\alpha, \tau}(\mathcal{T}_j)}^\tau \right)^{1/\tau}. \quad (13)$$

The Besov index of f is determined by the maximal index α for which (13) is finite. The above definition generalizes the classical function space theory of Besov spaces, where the tree partitions are non-adaptive. That is, classical Besov spaces may be defined by the special case of partitioning into dyadic cubes, each time using n levels of the tree.

Remark An active research area of approximation theory is the characterization of more geometrically adaptive approximation algorithms by generalizations of the classic ‘isotropic’ Besov space to more ‘geometric’ Besov-type spaces (Dahmen et. al 2001), (Dekel and Leviatan 2005), (Karaivanov and Petrushev 2003). It is known that different geometric approximation schemes are characterized by different flavors of Besov-type smoothness. In this work, for example, we assume all trees are created using partitions along the main n axes. This restriction may lead in general to potentially lower Besov smoothness of the underlying function and the sparsity of the wavelet representation. Yet, the theoretical definitions and results of this paper can also apply to more generalized schemes where for example the tree partitions are by arbitrary hyper-planes. In such a case, the smoothness index of a given function may increase.

Next, for a given tree \mathcal{T} and parameter $0 < \tau < p$ we denote the τ -strength of the tree by

$$N_\tau(f, \mathcal{T}) = \left(\sum_{\Omega \in \mathcal{T}} \|\psi_\Omega\|_p^\tau \right)^{1/\tau}. \quad (14)$$

Observe that

$$\lim_{\tau \rightarrow 0} N_\tau(f, \mathcal{T})^\tau = \#\{\Omega \in \mathcal{T} : \psi_\Omega \neq 0\}.$$

Let us further denote the τ -strength of a forest \mathcal{F} , by

$$\begin{aligned} N_\tau(f, \mathcal{F}) &:= \frac{1}{J} \left(\sum_{j=1}^J \sum_{\Omega \in \mathcal{T}_j} \|\psi_\Omega\|_p^\tau \right)^{1/\tau} \\ &= \frac{1}{J} \left(\sum_{j=1}^J N_\tau(f, \mathcal{T}_j)^\tau \right)^{1/\tau}. \end{aligned}$$

In the setting of a single tree constructed to represent a real-valued function, under mild conditions on the partitions (see remark after (5) and condition (17)), the theory of (Dekel and Leviatan 2005) proves the equivalence

$$|f|_{B_p^{\alpha, \tau}(\mathcal{T})} \sim N_\tau(f, \mathcal{T}). \quad (15)$$

This implies that there are constants $0 < C_1 < C_2 < \infty$, that depend on parameters such as α, p, n, r and ρ in condition (17) below, such that

$$C_1 |f|_{B_p^{\alpha, \tau}(\mathcal{T})} \leq N_\tau(f, \mathcal{T}) \leq C_2 |f|_{B_p^{\alpha, \tau}(\mathcal{T})}.$$

Therefore, we also have for the forest model

$$|f|_{B_p^{\alpha, \tau}(\mathcal{F})} \sim N_\tau(f, \mathcal{F}). \quad (16)$$

We now present a ‘Jackson-type estimate’ for the degree of the adaptive wavelet forest approximation. Its proof is in the Appendix.

Theorem 4 Let $\mathcal{F} = \{\mathcal{T}_j\}_{j=1}^J$ be a forest. Assume there exists a constant $0 < p < 1$, such that for any domain $\Omega \in \mathcal{F}$ on a level l and any domain $\Omega' \in \mathcal{F}$, on the level $l+1$, with $\Omega \cap \Omega' \neq \emptyset$, we have

$$|\Omega'| \leq \rho |\Omega|, \quad (17)$$

where $|E|$ denotes the volume of $E \subset \mathbb{R}^n$. Denote formally $f = \sum_{\Omega \in \mathcal{F}} w_j(\Omega) \psi_\Omega$, and assume that $|f|_{B_T^{\alpha,r}(\mathcal{F})} < \infty$, where

$$\frac{1}{\tau} = \alpha + \frac{1}{p}.$$

Then, for the M -term approximation (10) we have

$$\sigma_M(f) := \|f - f_M\|_p \leq C(p, \alpha, \rho) JM^{-\alpha} |f|_{B_T^{\alpha,r}(\mathcal{F})}. \quad (18)$$

One important contribution of this work is the attempt to generalize to the setting of machine learning, the function space theoretical perspective. There are several candidate numeric methods to estimate the critical ‘weak-type’ Besov smoothness index α from the given data. That is, the maximal α for which the Besov norm is finite. Our goal is to estimate the true smoothness of the underlying function, removing influences of noise and outliers if exist within the given dataset. One potential method is to use the equivalence (16) and then search for a transient value of τ for which $N_\tau(f, \mathcal{F})$ becomes ‘infinite’. However, we choose to generalize the numeric algorithm of (DeVore et. al. 1992) and estimate the critical index α using a numeric exponential fit of the error σ_M in (18). We found that it is somewhat more robust to fit each decision tree in the forest with an estimated smoothness index α_j and then average to obtain the estimated forest smoothness α . Thus, based on (18), we model the error function by $\sigma_{j,m} \sim c_j m^{-\alpha_j}$ for unknown c_j, α_j , where $\sigma_{j,m}$ is the approximation error when using the m most significant wavelets of the j th tree. First, notice that we can estimate $c_j \sim \sigma_{j,1}$. Then, using $\int_1^M m^{-\alpha} dm = (M^{1-\alpha} - 1)/(1 - \alpha)$, we estimate α_j by

$$\min_{\alpha_j} \left| \frac{M^{1-\alpha_j} - 1}{1 - \alpha_j} \sigma_{j,1} - \sum_{m=1}^{M-1} \sigma_{j,m} \right|. \quad (19)$$

Similarly to (DeVore et. al. 1992), we select only M significant terms, to avoid fitting the tail of the exponential expression. This is done by discarding wavelets that are overfitting the error on the Out Of Bag (OOB) samples (see Figure 3). Let us see some examples of how this works in practice. As can be seen in Figure 4, the estimate of the Besov index of two target functions using (19) stabilizes after a relatively small number of trees are added.

Next, we show that when an underlying function is not ‘well clustered’ and has a sharp transition of values across the boundary of two domains, then the Besov index is limited in the general case and suffers from the curse of dimensionality. Again, it should make sense to the practitioners, that such a function can be learnt but with more effort, e.g. trees with higher depth.

Lemma 5 Let $f(x) = \mathbf{1}_{\tilde{\Omega}}(x)$, where $\tilde{\Omega} \subset [0, 1]^n$ is a compact domain with a smooth boundary. Then, $f \in B_T^{\alpha,r}(\mathcal{T})$, for $\alpha < 1/p(n-1)$, $\tau^{-1} = \alpha + 1/p$, and any $r \geq 1$, where \mathcal{T}

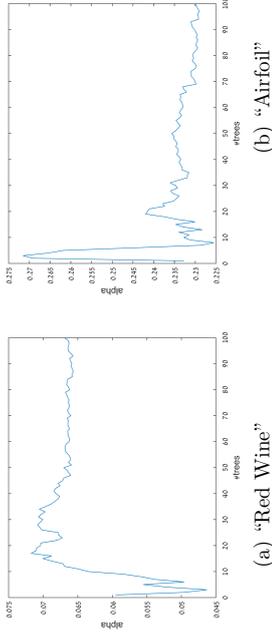


Figure 4: Estimation of the Besov critical smoothness index

is the tree with isotropic dyadic partitions, creating dyadic cubes of side lengths 2^{-k} on the level nk .

Proof See Appendix.

We note that in the general case, when subdivisions along main axes are used, the non-adaptive tree of the above lemma is almost best possible. That is, one cannot hope for significantly higher smoothness index using an adaptive tree with subdivisions along main axes. In Figure 5(a) we see 5000 random points and in (b) 250 random points, sampled from a uniform distribution taking a response value of $f(x) = \mathbf{1}_{\tilde{\Omega}}(x)$, where $\tilde{\Omega} \subset \mathbb{R}^2$, is the unit sphere.

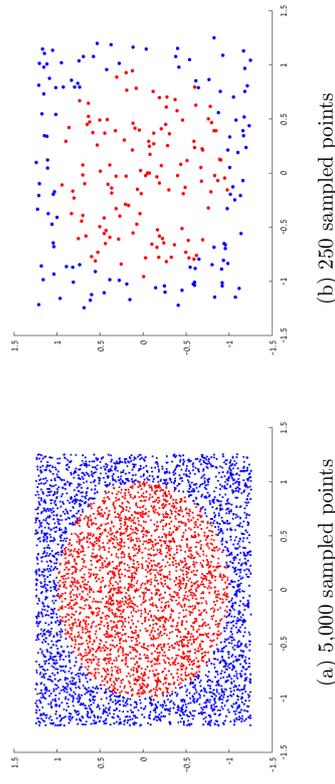


Figure 5: Dataset created by random sampling points of the indicator function of a unit sphere

By Lemma 4.3, the lower bound for the critical Besov exponent of f is $\alpha = 0.5$, for $p = 2$. This should correlate with the intuition of machine learning practitioners: the dataset does have two well defined clusters, but the boundary between the clusters (boundary of the sphere) is a non-trivial curve and any classification algorithm will need to learn the geometry of the curve.

In Figure 6 we see a plot of the numeric calculation of the α Besov index for given number of sampling points of f . We see relatively fast convergence to $\alpha = 0.51$. As discussed, our method attempts to capture the geometric properties of the ‘true’ underlying function that is potentially buried in the noisy input data. To show this, we constructed from a dataset of 10k samples of f , a ten dimensional dataset, by adding additional eight noisy features, with uniform distribution in $[0, 1]$ and no bearing on the response variable. The numeric computation in this example was again, $\alpha = 0.51$, which demonstrates that the method is stable under this noisy embedding in \mathbb{R}^n as well.

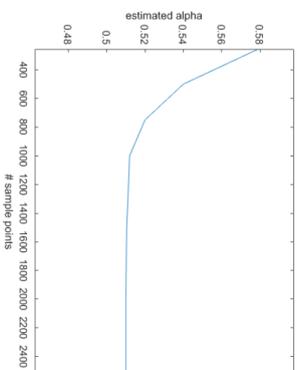


Figure 6: Numeric calculation of the α Besov index for given number of sampling points of the indicator function of a unit sphere.

5. Wavelet-based variable importance

In many cases, there is a need to understand in greater detail in what way the different variables influence the response variable (Guyon and Elisseeff 2003). Which of the possibly hundreds of parameters is more critical? What are the interactions between the significant variables? Also, the property of obtaining fewer features that provide equivalent prediction could be used for feature engineering and for ‘feature budget algorithms’ such as in (Feng et. al. 2015), (Vens and Costa 2011). As described in (Genner et. al. 2010), the use of RF for variable importance detection has several advantages.

There are several existing Variable Importance (VI) quantification methods that use RF. A popular approach for measuring the importance of a variable is summing the total decrease in node impurities when splitting on the variable, averaged over all trees (RF in R), (Hastie et. al. 2009). As suggested in the RF package documentation of the R language (RF in R): “For classification, the node impurity is measured by the Gini Index. For regression; it is measured by the residual sum of squares”. Although not stated specifically in (RF in R), it is common practice to multiply the information gain of each node by its size (Raibman and Stoffel 2004), (Du and Zhan 2002), (Rokach and Maimon 2005). Additional methods for variable importance measure are the ‘Permutation Importance’ measure (Genner et. al. 2010), or similarly ‘OOB randomization’ (Hastie et. al. 2009). With these latter two methods, sequential predictions of RF are done, when each time one feature is being permuted as the rest of the features remain. Then, the measure for variable importance is the difference in prediction accuracy before and after a feature is permuted in MSE terms.

However, both ‘Impurity gain’ and ‘Permutation’ have some pitfalls that should be considered, when used for variable importance. As shown by (Strobl et. al. 2006), the ‘Impurity gain’ tends to be in favor of variables with more varying values. As shown in (Strobl et. al. 2008), ‘Permutation’ tends to overestimate the variable importance of highly correlated variables.

The wavelet-based VI is derived by imposing a restriction on the adaptive re-ordering of the wavelet components (11), such that they must appear in ‘feature related blocks’. To make this precise, let $\{x \in \mathbb{R}^n, f(x)\}$ be a dataset and let \tilde{f} represent the RF decomposition, as in (8). We evaluate the importance of the i -th feature by

$$S_i^\tau := \frac{1}{J} \sum_{j=1}^J \sum_{\Omega \in T_j^{\cap V_i}} \|\psi_{i\Omega}\|_2^\tau, \quad i = 1, \dots, n, \quad (20)$$

where, $\tau > 0$ and V_i is the set of child domains formed by partitioning their parent domain along the i th variable. This allows us to score the variables, using the ordering $S_i^\tau \geq S_{i_2}^\tau \geq \dots$. Recall that our wavelet-based approach transforms classification problems into the functional setting (see section 2) by mapping each label l_k to a vertex $\vec{l}_k \in \mathbb{R}^{L-1}$ of a regular simplex. Therefore, in classification problems, the wavelet norms in (20) are given by (7) which implies that we provide a unified approach to VI.

It is crucial to observe that from an approximation theoretical perspective, the more suitable choice in (20) is $\tau = 1$, since with this choice, the ordering is related to ordering

the variables by the approximation error of their corresponding wavelet subset

$$\begin{aligned}
 & \left\| \bar{f} - \frac{1}{J} \sum_{j=1}^J \sum_{\Omega \in \mathcal{T}_j \cap V_i} \psi_\Omega \right\|_2 \\
 &= \min_{1 \leq i \leq n} \left\| \frac{1}{J} \sum_{k \neq i} \sum_{j=1}^J \sum_{\Omega \in \mathcal{T}_j \cap V_k} \psi_\Omega \right\|_2 \\
 &\leq \min_{1 \leq i \leq n} \frac{1}{J} \sum_{k \neq i} \sum_{j=1}^J \sum_{\Omega \in \mathcal{T}_j \cap V_k} \|\psi_\Omega\|_2 \\
 &= \min_{1 \leq i \leq n} \sum_{k \neq i} S_k^1 \\
 &= \sum_{1 \leq k \leq n} S_k^1 - \max_{1 \leq i \leq n} S_i^1.
 \end{aligned}$$

What is interesting is that, in regression problems, when using piecewise constant approximation in (1),(4), the VI score (20) with $\tau = 2$, is in fact exactly as in (Louppe et. al. 2013) when variance is used as the impurity measure. To see this, for any dataset $\{x \in \mathbb{R}^n, f(x)\}$ and domain $\tilde{\Omega}$ of an RF, denote briefly

$$K_{\tilde{\Omega}} := \#\{x_i \in \tilde{\Omega}\}, \quad \text{Var}(\tilde{\Omega}) = \frac{1}{\#\{x_i \in \tilde{\Omega}\}} \sum_{x_i \in \tilde{\Omega}} (f(x_i) - C_{\tilde{\Omega}})^2.$$

For any domain Ω of a RF, with children Ω', Ω'' , the variance impurity measure is

$$\Delta(\Omega) := \text{Var}(\Omega) - \frac{K_{\Omega'} \text{Var}(\Omega')}{K_\Omega} - \frac{K_{\Omega''} \text{Var}(\Omega'')}{K_\Omega}.$$

The importance of the variable i (up to normalization by the size of the dataset) is defined in (Louppe et. al. 2013) by

$$\frac{1}{J} \sum_{j=1}^J \sum_{\text{children of } \Omega \text{ in } \mathcal{T}_j \cap V_i} K_{\Omega} \Delta(\Omega). \quad (21)$$

Theorem 6 *The variable importance methods of (20) and (21) are identical for $\tau = 2$.*

Proof For any domain Ω and its two children Ω', Ω'' ,

$$\begin{aligned}
 K_\Omega \Delta(\Omega) &= K_\Omega \left(\text{Var}(\Omega) - \frac{K_{\Omega'} \text{Var}(\Omega')}{K_\Omega} - \frac{K_{\Omega''} \text{Var}(\Omega'')}{K_\Omega} \right) \\
 &= \sum_{x_i \in \tilde{\Omega}} (f(x_i) - C_\Omega)^2 - \sum_{x_i \in \Omega'} (f(x_i) - C_{\Omega'})^2 - \sum_{x_i \in \Omega''} (f(x_i) - C_{\Omega''})^2 \\
 &= \|\psi_{\Omega'}\|_2^2 + \|\psi_{\Omega''}\|_2^2.
 \end{aligned}$$

Therefore,

$$\frac{1}{J} \sum_{j=1}^J \sum_{\text{children of } \Omega \text{ in } \mathcal{T}_j \cap V_i} K_\Omega \Delta(\Omega) = S_i^1. \quad \diamond$$

Further to the choice of $\tau = 1$ over $\tau = 2$ in (20), the novelty of the wavelet-based VI approach is targeted at difficult noisy datasets. In these cases, one should compute VI at various degrees of approximation, using only subsets of ‘significant’ nodes, by thresholding out wavelet components with norm below some $\epsilon > 0$

$$S_i^1(\epsilon) := \sum_{j=1}^J \sum_{\Omega \in \mathcal{T}_j \cap V_i, \|\psi_\Omega\|_2 \geq \epsilon} \|\psi_\Omega\|_2. \quad (22)$$

As pointed out, a popular RF approach for identifying important variables is summing the total decrease in node impurities when splitting on the variable, averaged over all trees (RF in R), (Hastie et. al. 2009). However, this method may not be reliable in situations where potential predictor variables vary in their scale of measurement or their number of categories (Strobl et. al. 2006). This restriction is very limiting in practice, as in many cases binary variables such as ‘Gender’ are very descriptive where less descriptive variables (or noise) may vary with many values.

To demonstrate this problem, we follow the experiment suggested in (Strobl et. al. 2006). We set a number of samples to $m = 120$, where each sample has two explanatory independent variables: $x_1 \sim N(0, 1)$ and $x_2 \sim \text{Ber}(0.5)$. A correlation between $y = f(x_1, x_2)$ and x_2 is established by:

$$y \sim \begin{cases} \text{Ber}(0.7), & x_2 = 0, \\ \text{Ber}(0.3), & x_2 = 1. \end{cases} \quad (23)$$

In accordance with the point made in (Strobl et. al. 2006), when applying the VI of (RF in R), (Hastie et. al. 2009) we observe that the important variable is the ‘noisy’ uncorrelated feature x_1 . As shown in Example 1, while we may obtain many false partitions along the noise, with high probability their wavelet norm is controlled, and relatively small. In Figure 7 we see a histogram of the wavelet norms (taken from one of the RF trees) for the example (23). We see that the wavelet norms of the important variable x_2 are larger, but that there exists a long tail of wavelet norms relating to x_1 . Therefore, applying the thresholding strategy (22) as part of the feature importance estimation could be advantageous in such a case.

We now address the choice of ϵ in (22). So as to remove the noisy wavelet components from the VI scoring process, we choose the threshold as norm of the M -th wavelet, where

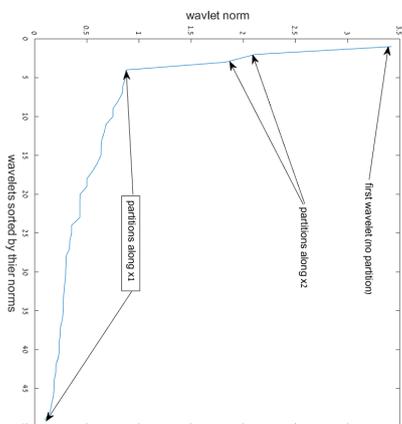


Figure 7: wavelets norms taken from one of the RF trees constructed for the example (23)

M is the selected using the M -term wavelet that minimizes the approximation error on the validation set $\{x_i, f(x_i)\}_{i=1\dots k}$ by

$$\epsilon = \|\psi_M\|_2, \quad s.t. \min_M \left\{ \sum_{i=1}^k \left(f(x_i) - \frac{1}{J} \sum_{m=1}^M \psi_{\gamma_{k_m}}(x_i) \right)^2 \right\}. \quad (24)$$

The calculation of ϵ for the ‘‘Pima diabetes’’ dataset using a validation set is depicted in Figure 8. In Section 6.2 we demonstrate the advantage of the wavelet-based thresholding technique in VI on several datasets.

6. Applications and Experimental Results

For our experimental results, we implemented C# code that supports RF construction, Besov index analysis, wavelet decompositions of RF and applications such as wavelet-based VI, etc. (source code is available, see link in (Wavelet RF code)). The algorithms are executed on the Amazon Web Services cloud, using up to 120 CPUs. Most datasets are taken from the UCI repository (UCI repository), which allows us to compare our results to previous work.

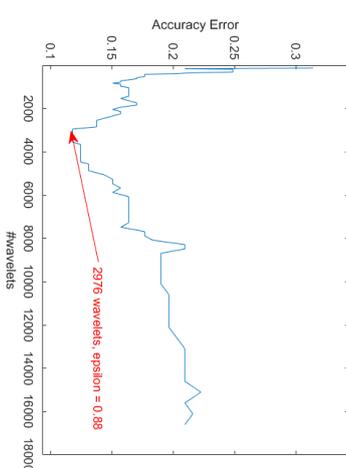


Figure 8: ‘‘Pima diabetes’’ - Choice of ϵ in (22) using the validation set

6.1 Ensemble Compression

In applications, constructed predictive models, such as RF, need to be stored, transmitted and applied to new data. In such cases the size of the model becomes a consideration, especially when using many trees to predict large amounts of incoming new data over distributed architectures. Furthermore, as presented in (Geurts and Gilles 2011), the number of total nodes of the RF and the average tree depth impact the memory requirements and evaluation performance of the ensemble.

In order to demonstrate the correlation between the Besov index of the underlying function and the ‘complexity’ of these datasets we need to compare on the same scale different datasets of different sizes and dimensions. Therefore, we replaced the commonly used metrics in machine learning such as MSE (Mean Square Error) by the normalized PSNR (Peak Signal To Noise Ratio) metric which is commonly used in the context of signal processing. For a given dataset $\{x_i, f(x_i)\}$ and an approximation $\{x_i, f_A(x_i)\}$ PSNR is defined by

$$\text{PSNR} := 10 \cdot \log_{10} \frac{\max_{i,j} \left\{ |f(x_i) - f(x_j)|^2 \right\}}{\frac{1}{\#\{x_i\}} \sum_i (f(x_i) - f_A(x_i))^2}.$$

Observe that higher PSNR implies smaller error. In Figure 9 we observe the rate-distortion performance measured on validation points in a fivefold cross validation of M -term wavelet approximation and standard RF, as trees are added. It can be seen that for functions that are smoother in ‘weak-type’ sense (e.g. higher α), wavelet approximation outperforms the standard RF. Table 1 below shows an extensive list of more datasets.

We now compare wavelet-based compression with existing RF pruning strategies. As stated by (Kulkarni and Sinha 2012), most of the current efforts in pruning RF are based on ‘Over-produce-and-Choose’ strategy, where the forest is grown to a fixed number of trees, and then only a subset of these trees are chosen by a ‘leave one out strategy’ as in (Martinez-Muoz et. al. 2009), (Yang et. al. 2012). For each dataset we first computed a point at which the graph of wavelet approximation error begins to ‘flatten out’ on the validation set. We then used this target error pre-saturation point for both wavelet shrinkage and the pruning methods that aim for a minimal number of nodes to achieve it on a validation set of fivefold cross validation. To this end, we have generated RF with 100 decision trees with 80% bagging and \sqrt{n} hyper-parameter. The two pruning strategies are based on a ‘leave one out’ strategy as presented in (Yang et. al. 2012). In this approach trees are recursively omitted according to their correspondence with the rest of the ensemble (based on the correspondence of the margins in classification and MSE in regression). We have collected the results of the experiment described above applied to 12 UCI datasets in Table 1. The datasets for classification are marked (C) and regression (R). One may observe from Table 1 that the wavelet-based method performs better than conventional pruning. Also, as expected, there is significant correlation between the performance of compression and the function smoothness. That is, the compression is more effective for smoother functions.

Note that when computing an M -term wavelet approximation, some components may be unconnected as depicted in Figure 2. Obviously, any compression of the wavelet approximation would need to encode the nodal data associated with these unconnected components. Therefore, we enforce connectivity on any wavelet approximation we compute, by adding all wavelet components along the tree paths leading to the selected significant wavelet components. Thus, the wavelet compression appearing in Table 1 is in the form of a collection of J connected subtrees.

6.2 Variable Importance

We first demonstrate how the wavelet-based method (20) with $\tau = 1$, succeeds in identifying the features with the highest impact on the prediction. In Figure 10 (a), we show an histogram of VI scores for the ‘Red Wine’ dataset using the wavelet-based approach (20). As can be seen, the top three features in the histogram are ‘Alcohol’, ‘Volatile acidity’ and ‘Sulphates’. We then constructed RFs for all the possible triple combinations of features of the UCI repository ‘Wine Quality’ dataset (165 simulations) using 100 trees and 80% bagging. In Figure 10(b), we can see the MSE of each of these RFs. One can verify that the triple ‘Alcohol’, ‘Volatile acidity’ and ‘Sulphates’ (index 81) has the smallest error, as identified by our wavelet-based method.

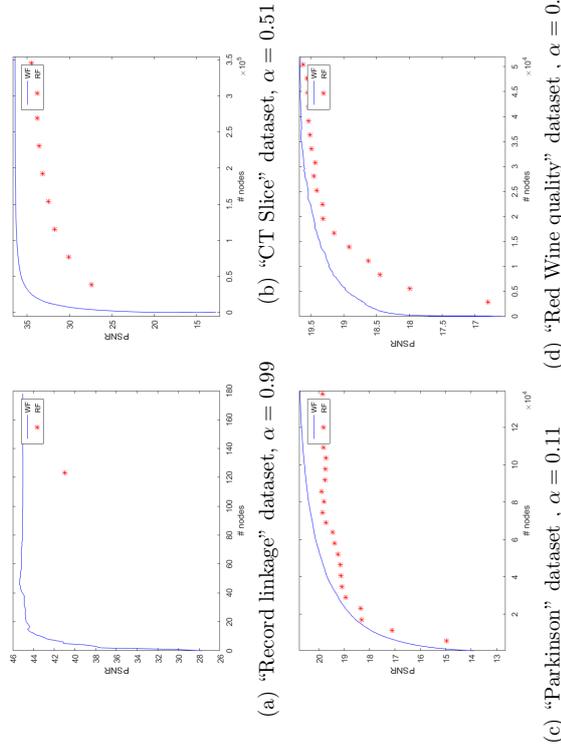
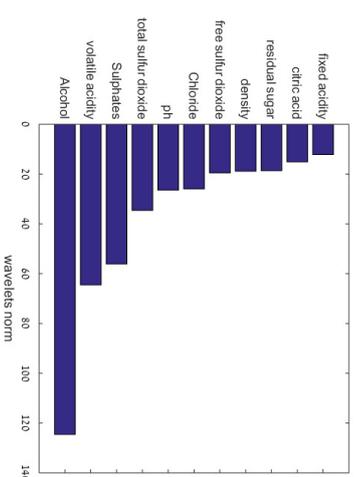


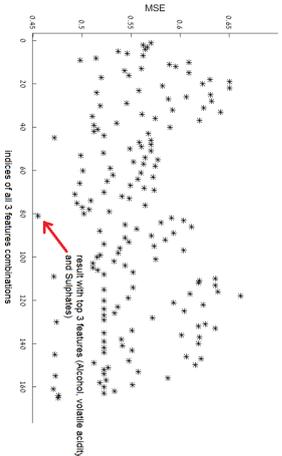
Figure 9: PSNR of four UCI data sets.

Table 1: Compression - number of nodes required to reach the error pre-saturation point

Dataset	error	Pruning Min-D (Yang et. al. 2012)		Pruning Mean-D (Yang et. al. 2012)		Wavelet subtrees		α
		#trees	#nodes	#trees	#nodes	#trees	#nodes	
Record linkage (C)	2%	1	123	1	123	1	6	0.99
CT Slice (R)	2.9 MSE	2	77042	2	76396	2	5141	0.51
Titanic (C)	17%	3	711	10	2248	1	34	0.42
Balanced scale (C)	22%	1	185	1	185	1	55	0.34
Concrete (R)	15 MSE	19	2297	8	966	3	64	0.32
Magic Gamma (C)	13%	9	26793	5	14961	3	1657	0.25
Airfoil (R)	3.2 MSE	5	4533	3	7487	3	1929	0.23
California Housing (R)	0.5 MSE	4	65436	9	149863	4	7292	0.2
EEG (C)	8%	7	17845	11	28355	6	12808	0.15
Parkinson (R)	3.2 MSE	18	103822	19	110187	12	20947	0.11
Wine quality (R)	0.4 MSE	14	30350	13	36439	12	29089	0.07
Year Prediction (R)	88 MSE	21	10657799	24	12201588	19	9300284	0.02



(a) Wavelet-based feature importance histogram



(b) Error of RFs constructed over all possible 3 feature subsets

Figure 10: Wavelet-based variable importance of the UCI Red wine data set

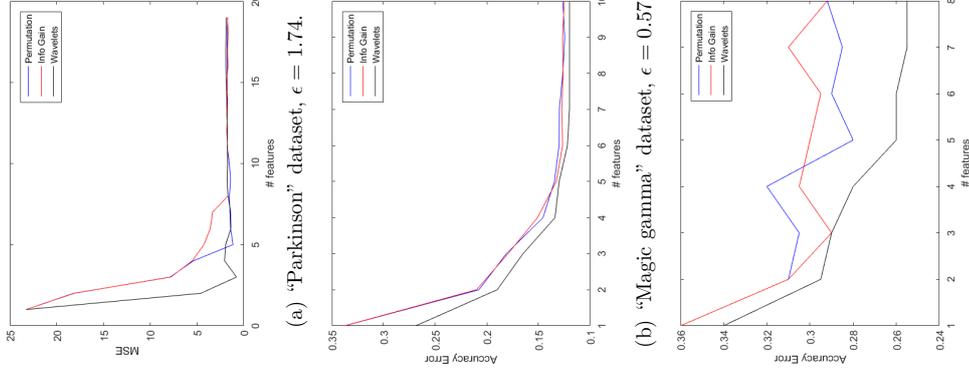
Next, we show that the wavelet-based VI approach, in particular, the noise-removal variant using (22) with $\epsilon > 0$, can provide a better estimation of VI than the existing methods in R (RF in R). Note that we apply the wavelet-based VI method in classification problems as well, competing with, for example, the standard Gini-based algorithm of R. To this end we employ a test methodology used in (Feng et. al. 2015). Using each VI method we first calculate a corresponding VI score of the features. Each method uses an RF with 100 trees and 80% bagging. However, the wavelet-based method was computed using our implementation, based on (22) while the Permutation and Information Gain based methods were applied using R. After each method ‘decides’ on the order of the features by importance, we iterate by adding features one-by-one, where at the k -th iteration, only the selected first k features are used for prediction. Here also, we used wavelet-based choice of k most important features to construct a wavelet-based best prediction, while for the other methods, we used their choice of k most important features as the input for an R based RF. The results of fivefold cross validation are presented in Figure 11. For example, in Figure 11(a), we see that on the “Parkinson” dataset, the wavelet-based method reaches better prediction using the first three features it selected. This is due to fact that the wavelet-based method selected different features (‘Age’, ‘Time’ and ‘Gender’) than the other methods.

6.3 Classification and regression tasks

In this Section we focus on difficult datasets, such as small sets or with high bias, bad features, mis-labeling and outliers (see for example “Pima Diabetes” dataset with only 768 samples with 8 attributes in Figure 11(c)) and show that in such cases the wavelet-based approach provides smaller predictive errors.

We begin with a demonstration of a case of ‘false labeling’ using the R machine learning benchmark “Spirals” (Spiral dataset). From the given dataset we create a dataset with mis-labeling by randomly replacing 20% of the values of the response variable. The original and noisy datasets are rendered in Figure 12. We then compare the predictive performance of the standard RF and the M -term wavelet approximation (11), where optimal M values are computed automatically as depicted in Figure 3. We also compare the M -term performance to a minimal node size restriction as in (Biau and Scornet 2016), setting this value to 5, as in (Denil et. al. 2014). We perform RF construction with 1000 trees and 5 fold cross validation.

When the training dataset contains ‘false labeling’, the correspondence with the testing set is reduced. Trying to restrict the tree depth, can potentially miss the geometry of the underlying function, while too many levels can lead to overfitting. As seen in Table 2, the wavelet approach selects the right significant components from any tree and any level and thus outperforms the standard RF method. Observe that the added value of the wavelet



(c) “Pima Diabetes” dataset, $\epsilon = 0.93$.

Figure 11: Comparisons of performance of standard VI methods used in R, with the wavelet-based method (22)

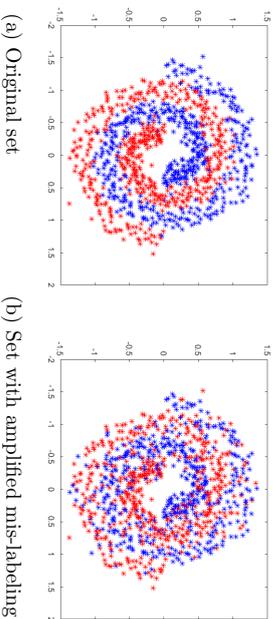


Figure 12: ‘Spirals’ dataset (Spiral dataset)

approach is more significant in the second case with more ‘false labeling’ in the training set.

Table 2: ‘Spirals’ dataset - Classification results.

	Wavelet error	RF error	Pruned RF error
Original spiral set	$12.2 \pm 0.9\%$	$14.4 \pm 1.1\%$	$15.9 \pm 0.8\%$
Set with amplified mis-labeling	$13.9 \pm 1.2\%$	$17.8 \pm 1.3\%$	$22.7 \pm 1.6\%$

Next, we compare the performance of wavelet-based regression with state-of-the-art method on a challenging problem. The authors of (Denil et. al. 2014) provide comparative results of different pruning strategies for the difficult “Wine Quality” dataset. Learning this dataset is challenging since the data is very biased and depends on the personal taste of the wine experts. In Table 3 below, we collect the results of (Biau 2012), (Biau et. al. 2008), (Breiman 2001) and (Denil et. al. 2014) (as listed in (Denil et. al. 2014)). The RFs are all constructed of 1000 trees and fivefold cross validation is applied. We follow the notation presented in (Denil et. al. 2014) and use the abbreviation that was provided for each method variation (‘+’, ‘F’, ‘S’, ‘NB’, ‘T’). In our RF implementation, we used bootstrapping with 80% and randomized \sqrt{n} features at each node. M was selected automatically using 10 percent of the training set.

Another form of a challenging dataset is when some of the features are extremely noisy or uncorrelated with the response variable. As shown in (Strobl et. al. 2006) (see the discussion in Section 5), in such cases, RF partitions sometimes are influenced by these variables and the constructed ensemble is of lower quality. To explore the impact of our approach on such datasets, we used the “Poker Hand” dataset from the UCI repository (UCI repository) in two modes: with and without a very non-descriptive feature “instance

Table 3: Performance comparison on the “Wine Quality”

Algorithm	MSE
Biau08	0.53
Biau12	0.59
Biau12+T	0.57
Biau12+S	0.57
Denil	0.48
Denil+F	0.48
Denil+S	0.41
Breiman	0.4
Breiman+NB	0.39
Wavelets	0.36

id”. As can be seen from Figure 13, the wavelet method significantly outperforms the standard RF regression, especially in the second scenario with the ‘bad’ feature included.

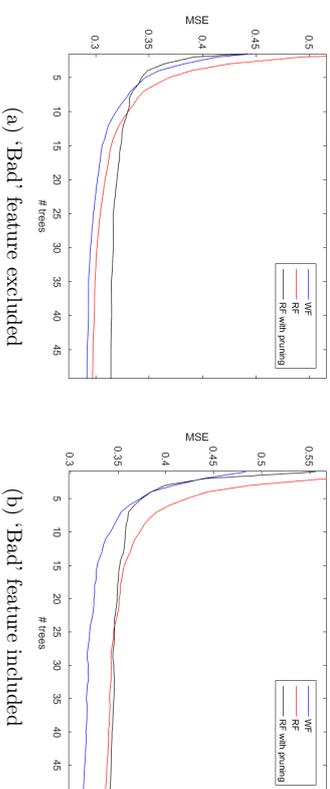


Figure 13: The impact of a bad feature on the regression of the “Poker Hand” dataset

Acknowledgments

The authors would like to thank the reviewers for their careful reading of several versions of this work and valuable comments which resulted in a substantially revised manuscript. This work was supported by an ‘AWS in Education Grant award’.

References

- Alani D., Averbuch A. and Dekel S., Image coding using geometric wavelets, *IEEE transactions on image processing* 16:69-77, 2007.
- Alpaydin E., *Introduction to machine learning*, MIT Press, 2004.
- Avery M., Literature Review for Local Polynomial Regression, <http://www4.ncsu.edu/mravery/AveryReview2.pdf>.
- Bernard S., Adam S. and Heutte L., Dynamic random forests, *Pattern Recognition Letters* 33:1580-1586.
- Biau G., Analysis of a random forests model, *Journal of Machine Learning Research* 13: 1063-1095, 2012.
- Biau G., Devroye L. and Lugosi G., Consistency of random forests and other averaging classifiers, *Journal of Machine Learning Research* 9:2015-2033, 2008.
- Biau G. and Scornet E., A random forest guided tour, *TEST* 25(2):197-227, 2016.
- Breiman L., Random forests, *Machine Learning* 45:5-32, 2001.
- Breiman L., Bagging predictors, *Machine Learning* 24(2):123-140, 1996.
- Breiman L, Friedman J., Stone C. and Olshen R., *Classification and Regression Trees*, Chapman and Hall/CRC, 1984.
- Boulesteix A., Janitza S., Kruppa J. and König I., Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2(6):493-507, 2012.
- Chen H., Tino P. and Yao X., Predictive Ensemble Pruning by Expectation Propagation, *IEEE journal of knowledge and data engineering* 21:999-1013, 2009.
- Christensen O., *An introduction to Frames and Riesz Bases*, Birkäuser, 2002.
- Criminisi A., Shotton J. and Konukoglu E., Forests for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning, *Microsoft Research technical report*, report TR-2011-114, 2011.
- Dahmen W., Dekel S. and Petrushev P., Two-level-split decomposition of anisotropic Besov spaces, *Constructive approximation* 31:149-194, 2001.
- Daubechies I., *Ten lectures on wavelets*, CBMS-NSF Regional Conference Series in Applied Mathematics, 1992.
- Dekel S., Gershtansky I., Active Geometric Wavelets, In *Proceedings of Approximation Theory XIII 2010*, 95-109, 2012.
- Dekel S. and Leviatan D., Adaptive multivariate approximation using binary space partitions and geometric wavelets, *SIAM Journal on Numerical Analysis* 43:707-732, 2005.
- Denil M., Matheson D. and De Freitas N., Narrowing the gap Random forests in theory and in practice, In *Proceedings of the 31st International Conference on Machine Learning* 32, 2014.
- DeVore R., Nonlinear approximation, *Acta Numerica* 7:51-150, 1998.
- DeVore R. and Lorentz G., *Constructive approximation*, Springer Science and Business, 1993.
- DeVore R., Jawerth B. and Lucier B., Image compression through wavelet transform coding, *IEEE transactions on information theory* 38(2):719-746, 1992.
- Du W. and Zhan Z., Building decision tree classifier on private data, In *Proceedings of the IEEE international conference on Privacy, security and data mining* 14:1-8, 2002.
- Elad M., *Sparse and redundant representations: from theory to applications in signal and image processing*, Springer Science and Business Media, 2010.
- Feng N., Wang J. and Saligrama V., Feature-Budgeted Random Forest, In Proceedings of The 32nd International Conference on Machine Learning, 1983-1991, 2015.
- Kelley P. and Barry R., Sparse spatial autoregressions, *Statistics and Probability Letters* 33(3):291-297, 1997.
- Gavish M., Nadler B., Coifman R., Multiscale wavelets on trees, graphs and high dimensional data: Theory and applications to semi supervised learning, In *Proceedings of the 27th International Conference on Machine Learning*, 367-374, 2010.
- Genuer R., Poggi J. and Christine T., Variable selection using Random Forests, *Pattern Recognition Letters* 31(14): 2225-2236, 2010.

- Geurts P. and Gilles L., Learning to rank with extremely randomized trees, In *JMLR: Workshop and Conference Proceedings* 14:49-61, 2011.
- Guyon I. and Elisseeff A., An introduction to variable and feature selection, *Journal of Machine Learning Research* 3:1157-1182, 2003.
- Hastie T., Tibshirani R. and Friedman J., *The elements of statistical learning*, Springer, 2009.
- Joly A., Schmitzler F., Geurts P. and Wehenkel L., L1-based compression of random forest models, In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 375-380, 2012.
- Karavaynov B. and Petrushev P., Nonlinear piecewise polynomial approximation beyond Besov spaces, *Applied and computational harmonic analysis* 15:177-223, 2003.
- Kulkarni V. and Sinha P., Pruning of Random Forest classifiers: A survey and future directions, In *International Conference on data science and engineering*, 64-68, 2012.
- Lee A., Nadler B. and Wasserman L., Treetlets: an adaptive multi-scale basis for sparse unordered data, *Annals of Applied Statistics* 2(2):435-471, 2008.
- Loh W., Classification and regression trees, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 1(1):14-23, 2011.
- Louppe G., Wehenkel L., Sutura A. and Geurts P., Understanding variable importances in forests of randomized trees, *Advances in Neural Information Processing Systems* 26:431-439, 2013.
- Mallat S., *A Wavelet tour of signal processing, 3rd edition (the sparse way)*, Academic Press, 2009.
- Martinez-Muoz G., Hernández-Lobato D. and Suarez A., An analysis of ensemble pruning techniques based on ordered aggregation, *IEEE Transactions on pattern analysis and machine intelligence* 31:245-259, 2009.
- Radha H., Vetterli M. and Leonardti R., Image compression using binary space partitioning trees, *IEEE transactions on image processing* 5:1610-1624, 1996.
- Raileanu L. and Stoffel K., Theoretical comparison between the Gini index and information gain criteria, *Annals of Mathematics and Artificial Intelligence* 41(1):77-93, 2004.
- 'Random Forest' package in R, <http://cran.r-project.org/web/packages/randomForest/randomForest.pdf>
- Rokach L. and Maimon O., Top-down induction of decision trees classifiers-a survey, *IEEE transactions on systems, man, and cybernetics, part C: applications and reviews* 35(4):476-487, 2005.
- Salembier P. and Garrido L., Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval, *IEEE transactions on image processing* 9:561-576, 2000.
- Spiral dataset, <http://www.inside-r.org/packages/cran/mlbench/docs/mlbench.spirals>.
- Strobl C., Boulesteix A., Zeileis A. and Hothorn T., Bias in random forest variable importance measures, In *Workshop on Statistical Modelling of Complex Systems*, 2006.
- Strobl C., Boulesteix A., Kneib T., Augustin T. and Zeileis A., Conditional variable importance for random forests, *BMC bioinformatics* 9(1):1-11, 2008.
- UCI machine learning repository, <http://archive.ics.uci.edu/ml/>.
- Vens C. and Costa F., Random forest based feature induction, In *IEEE international conference on data mining*, 744-753, 2011.
- Yang F., Lu W., Luo L. and Li T., Margin optimization based pruning for random forest, *Neurocomputing* 94:54-63, 2012.
- Wavelet-based Random Forest source code, <https://github.com/orenelis/WaveletForest.git>.

Appendix

Proof of Lemma 1

Denoting briefly for any domain $\tilde{\Omega}$, $K_{\tilde{\Omega}} := \#\{x_i \in \tilde{\Omega}\}$ we have

$$\begin{aligned}
 \sum_{x_i \in \Omega} (f(x_i) - C_{\Omega})^2 - V_{\Omega} &= \sum_{x_i \in \Omega} (f(x_i) - C_{\Omega})^2 - \sum_{x_i \in \Omega'} (f(x_i) - C_{\Omega'})^2 - \sum_{x_i \in \Omega''} (f(x_i) - C_{\Omega''})^2 \\
 &= \sum_{x_i \in \Omega'} [(f(x_i) - C_{\Omega})^2 - (f(x_i) - C_{\Omega'})^2] + \\
 &\quad \sum_{x_i \in \Omega''} [(f(x_i) - C_{\Omega})^2 - (f(x_i) - C_{\Omega''})^2] \\
 &= 2(C_{\Omega'} - C_{\Omega}) \sum_{x_i \in \Omega'} f(x_i) + K_{\Omega'} (C_{\Omega}^2 - C_{\Omega'}^2) \\
 &\quad + 2(C_{\Omega''} - C_{\Omega}) \sum_{x_i \in \Omega''} f(x_i) + K_{\Omega''} (C_{\Omega}^2 - C_{\Omega''}^2) \\
 &= 2(C_{\Omega'} - C_{\Omega}) K_{\Omega'} C_{\Omega'} + K_{\Omega'} (C_{\Omega}^2 - C_{\Omega'}^2) \\
 &\quad + 2(C_{\Omega''} - C_{\Omega}) K_{\Omega''} C_{\Omega''} + K_{\Omega''} (C_{\Omega}^2 - C_{\Omega''}^2) \\
 &= K_{\Omega'} (C_{\Omega'} - C_{\Omega})^2 + K_{\Omega''} (C_{\Omega''} - C_{\Omega})^2 \\
 &= \|\psi_{\Omega'}\|_2^2 + \|\psi_{\Omega''}\|_2^2 = W_{\Omega}.
 \end{aligned}$$

Now, since $\sum_{x_i \in \Omega} (f(x_i) - C_{\Omega})^2$ is independent of the selection of the partition of Ω and since W_{Ω} is always positive, the search for minimizing V_{Ω} is equivalent to maximizing W_{Ω} .

◇

Proof of Example 1

1. For any attribute $j \neq k$ we denote $m_1 := \#\{x_i \in \Omega'\}$ and $m_2 := m - m_1$. Hence, for any $\delta \in (0, 1)$, applying the Hoeffding bound gives w.p. $\geq 1 - \delta$

$$\left| C_{\Omega'} - \frac{1}{2} \right| \leq \sqrt{\frac{\log(2/\delta)}{2m_1}}, \quad \left| C_{\Omega''} - \frac{1}{2} \right| \leq \sqrt{\frac{\log(2/\delta)}{2m_2}}. \quad (25)$$

Note, that we can write $C_{\Omega} = \frac{m_1}{m} C_{\Omega'} + \frac{m_2}{m} C_{\Omega''}$. Thus, using (25) we get w.p. $\geq 1 - \delta$,

$$\begin{aligned}
 (C_{\Omega} - C_{\Omega'})^2 &= \frac{m_2^2}{m^2} (C_{\Omega''} - C_{\Omega'})^2 \\
 &\leq \frac{m_2^2}{m^2} \left(\frac{1}{m_1} + \frac{1}{m_2} \right) \log(2/\delta).
 \end{aligned}$$

Therefore, w.p. $\geq 1 - \delta$,

$$\begin{aligned}
 \|\psi_{\Omega'}\|_2^2 &= m_1 (C_{\Omega} - C_{\Omega'})^2 \\
 &\leq \frac{m_1 m_2^2}{m^2} \left(\frac{1}{m_1} + \frac{1}{m_2} \right) \log(2/\delta) \\
 &= \left(\frac{m_2^2}{m^2} + \frac{m_1 m_2}{m^2} \right) \log(2/\delta) \\
 &\leq 2 \log(2/\delta).
 \end{aligned}$$

2. Observe that for the case $j = k$, a subdivision that minimizes (1) is $x_k = 1/2$. Denote $m_1 = \#\{x_i \in \Omega : y_i = 1\}$. Applying the Hoeffding bound with $\delta \in (0, 1)$ yields w.p. $\geq 1 - \delta$

$$\left| m_1 - \frac{m}{2} \right| \leq \sqrt{\frac{\log(2/\delta)}{2m}}.$$

If Ω' is the subset of $[0, 1]^m$ where $x_k > 1/2$, then $C_{\Omega'} = 1$ and $\|\psi_{\Omega'}\|_2^2 = m_1 (1 - \frac{m_1}{m})^2$. Plugging into the bound above we conclude that w.p. $\geq 1 - \delta$,

$$\begin{aligned}
 \|\psi_{\Omega'}\|_2^2 &\geq \left(\frac{m}{2} - \sqrt{\frac{\log(2/\delta)}{2m}} \right) \left(1 - \frac{\frac{m}{2} + \sqrt{\frac{\log(2/\delta)}{2m}}}{m} \right)^2 \\
 &= \left(\frac{m}{2} - \sqrt{\frac{\log(2/\delta)}{2m}} \right)^3 / m^2.
 \end{aligned}$$

◇

Proof of Theorem 4 We prove the case $1 < p < \infty$ (the case $0 < p \leq 1$ is easier). We need to show two essential properties. First, for any $\Omega' \in \mathcal{F}$ and any $x \in \Omega'$, denoting $\Lambda := \{\Omega \in \mathcal{F} : x \in \Omega, |\Omega| \geq |\Omega'|\}$, we have

$$\sum_{\Omega \in \Lambda} \left(\frac{|\Omega'|}{|\Omega|} \right)^{1/p} \leq C(\rho, p) J. \quad (26)$$

Indeed, using (17), recursively for all domains on lower levels intersecting with Ω' we have

$$\begin{aligned}
 \sum_{\Omega \in \Lambda} \left(\frac{|\Omega'|}{|\Omega|} \right)^{1/p} &\leq \sum_{k=0}^{\infty} J \rho^{k/p} \\
 &\leq \frac{J}{1 - \rho^{1/p}}.
 \end{aligned}$$

Secondly, we need the property that

$$\|\psi_\Omega\|_\infty \leq c|\Omega|^{-1/p} \|\psi_\Omega\|_p, \quad \forall \Omega \in \mathcal{F}. \quad (27)$$

It is easy to see property (27) for the case $r = 1$, where $\psi_\Omega = \mathbf{1}_\Omega C_\Omega$, but it is also known for the general case of $r \geq 1$ and convex domains (see e.g. (Dekel and Leviatan 2005)). This allows us to prove the following Lemma

Lemma 7 For $1 < p < \infty$, let $F(x) = \sum_{i=1}^I w_{j_i(\Omega_i)} \psi_{\Omega_i}(x)$, $\Omega_i \in \mathcal{F}$, where $\|w_{j_i(\Omega_i)} \psi_{\Omega_i}\|_p \leq L$. Then

$$\|F\|_p \leq cJLL^{1/p}. \quad (28)$$

Proof Applying property (27) gives

$$\begin{aligned} \|F\|_p &\leq \left\| \sum_{i=1}^I w_{j_i(\Omega_i)} \psi_{\Omega_i} \right\|_\infty \left\| \mathbf{1}_{\Omega_i}(\cdot) \right\|_p \\ &\leq L \left\| \sum_{i=1}^I |\Omega_i|^{-1/p} \mathbf{1}_{\Omega_i}(\cdot) \right\|_p. \end{aligned}$$

We define

$$\Gamma(x) := \begin{cases} \min_{1 \leq i \leq I} \{|\Omega_i| : x \in \Omega_i\}, & x \in \bigcup_{i=1}^I \Omega_i, \\ 0, & \text{else.} \end{cases}$$

Then, (26) yields

$$\sum_{i=1}^I |\Omega_i|^{-1/p} \mathbf{1}_{\Omega_i}(x) \leq cJ\Gamma(x)^{-1/p}, \quad \forall x \in \Omega_0.$$

Thus,

$$\begin{aligned} \|F\|_p &\leq cL \left\| \Gamma(\cdot)^{-1/p} \right\|_p \\ &= cJL \left(\int_{\bigcup \Omega_i} \Gamma(x)^{-1} dx \right)^{1/p} \\ &\leq cJL \left(\sum_{i=1}^I |\Omega_i|^{-1} \int_{\Omega_i} dx \right)^{1/p} = cJLL^{1/p}. \end{aligned}$$

◇

We now proceed with the proof of the Theorem. Observe that we may use (16), that is, $|f|_{B_{2^{\nu}}(\mathcal{F})} \sim N_\tau(f, \mathcal{F})$. For $\nu = 1, 2, \dots$, denote

$$\Xi_\nu := \left\{ \Omega \in \mathcal{F} : 2^{-\nu} N_\tau(f, \mathcal{F}) \leq w_{j(\Omega)} \|\psi_\Omega\|_p < 2^{-\nu+1} N_\tau(f, \mathcal{F}) \right\}.$$

Recall that for any non-negative discrete sequence $\beta = \{\beta_k\}_{k=1}^\infty$, the weak- l_r norm $\|\beta\|_{w_{l_r}}$ is defined as the infimum (if exists) over all $A > 0$, for which

$$\#\{\beta_k : \beta_k > \varepsilon\} \varepsilon^\tau \leq A^\tau, \quad \forall \varepsilon > 0.$$

Since $\|\beta\|_{w_{l_r}} \leq \|\beta\|_r$, this implies that

$$\#\Xi_m \leq \sum_{\nu \leq m} \#\Xi_\nu = \# \bigcup_{\nu \leq m} \Xi_\nu \leq 2^m \tau.$$

Let $F_\nu(x) := \sum_{\Omega \in \Xi_\nu} w_{j(\Omega)} \psi_\Omega(x)$. For the special case $M := \sum_{\nu \leq m} \#\Xi_\nu$, we have by (28)

$$\begin{aligned} \|f - f_M\|_p &\leq \left\| \sum_{\nu=m+1}^\infty F_\nu \right\|_p \\ &\leq \sum_{\nu=m+1}^\infty \|F_\nu\|_p \\ &\leq cJ \sum_{\nu=m+1}^\infty 2^{-\nu} N_\tau(f, \mathcal{F}) (\#\Xi_\nu)^{1/p} \\ &\leq cJ N_\tau(f, \mathcal{F}) \sum_{\nu=m+1}^\infty 2^{-\nu(1-\tau/p)} \\ &\leq cJ N_\tau(f, \mathcal{F}) M^{-(1/\tau-1/p)} = cJ N_\tau(f, \mathcal{F}) M^{-\alpha}. \end{aligned}$$

Extending this result for any $M \geq 1$ is standard (using a larger leading constant). This completes the proof.

◇

Proof of Lemma 3 Since there are a finite number of boxes, there exists a, possibly unbalanced, binary tree that after at most $K2^n$ partitions, has also the boxes $\{B_k\}$ as nodes of the tree. Since the modulus of smoothness of order r of polynomials of degree $r-1$ is zero (DeVore 1998), (Devore and Lorentz 1993), for any of these box nodes we have that

$$\omega_r(f, B_k)_\tau = \omega_r(P_k, B_k)_\tau = 0.$$

Similarly, for any descendant node $\Omega' \subset B_k$, for some $1 \leq k \leq K$,

$$\omega_r(f, \Omega')_\tau = \omega_r(P_k, \Omega')_\tau = 0.$$

For any node Ω such that $\Omega \cap B_k = \emptyset$, $1 \leq k \leq K$, we have

$$\omega_r(f, \Omega)_\tau = \omega_r(0, \Omega)_\tau = 0.$$

We may then conclude that $\omega_r(f, \Omega)_\tau \neq 0$, for only a finite low-level subset Λ of the tree nodes, each strictly containing at least one B_k . Therefore, for any $\alpha > 0$,

$$\begin{aligned} |f|_{B_k^{\alpha, r}} &= \left(\sum_{\Omega \in \mathcal{T}} (|\Omega|^{-\alpha} \omega_r(f, \Omega))^\tau \right)^{1/\tau} \\ &= \left(\sum_{\Omega \in \Lambda} (|\Omega|^{-\alpha} \omega_r(f, \Omega)_\tau)^\tau \right)^{1/\tau} \\ &\leq 2^r \|f\|_\tau \left(\min_k |B_k| \right)^{-\alpha} (K2^n)^{1/\tau}, \end{aligned}$$

where we have used the inequality

$$\omega_r(f, \Omega)_\tau \leq 2^r \|f\|_{L_r(\Omega)} \leq 2^r \|f\|_{L_r(\Omega_0)}.$$

◇

Proof of Lemma 5 As stated, the tree \mathcal{T}_l with isotropic dyadic partitions, creates dyadic cubes of side lengths 2^{-k} at the level nk . Let us denote by $D := \{D_k\}_{k=0}^\infty$, the collection of dyadic cubes of $[0, 1]^n$, where D_k is the collection of cubes with side lengths 2^{-k} . Observe that any domain $\Omega' \in \mathcal{T}_l$, at a level $nk < l < n(k+1)$, is contained in some dyadic cube $\Omega \in \mathcal{T} \cap D_k$ at the level nk . Also, from the properties of the modulus of smoothness, $\Omega' \subset \Omega \Rightarrow \omega_r(f, \Omega')_\tau \leq \omega_r(f, \Omega)_\tau$. Combining these two observations gives

$$|\Omega'|^{-\alpha} \omega_r(f, \Omega')_\tau \leq 2^{n\alpha} |\Omega|^{-\alpha} \omega_r(f, \Omega)_\tau.$$

Next, observe that for any $\Omega \in D_k$

$$\omega_r(f, \Omega)_\tau \begin{cases} = 0, & \Omega \cap \partial\tilde{\Omega} = \emptyset, \\ \leq 2^{-kn/\tau}, & \Omega \cap \partial\tilde{\Omega} \neq \emptyset, \end{cases}$$

where $\partial\tilde{\Omega}$ is the boundary of $\tilde{\Omega}$. Therefore,

$$\begin{aligned} |f|_{B_k^{\alpha, r}(\mathcal{T})} &\leq c(n, \alpha, \tau) \left(\sum_{\Omega \in D} (|\Omega|^{-\alpha} \omega_r(f, \Omega)_\tau)^\tau \right)^{1/\tau} \\ &\leq c(n, \alpha, \tau, r) \left(\sum_{k=0}^\infty 2^{kn(\alpha\tau-1)} \#\left\{ \Omega \in D_k : \Omega \cap \partial\tilde{\Omega} \neq \emptyset \right\} \right)^{1/\tau}. \end{aligned}$$

Thus, it remains to estimate the maximal number of dyadic cubes of side length 2^{-k} that can intersect a smooth boundary of a domain $\tilde{\Omega} \subset [0, 1]^n$. For sufficiently large k , only one connected component of the boundary $\partial\tilde{\Omega}$ intersects a dyadic cube $\Omega \in D_k$, in similar manner to an hyperplane of dimension $n-1$ with surface area $\leq c2^{-k(n-1)}$. Therefore, for sufficiently large k

$$\#\left\{ \Omega \in D_k : \Omega \cap \partial\tilde{\Omega} \neq \emptyset \right\} \leq c2^{k(n-1)}.$$

This gives

$$|f|_{B_k^{\alpha, r}(\mathcal{T})} \leq c \left(n, \alpha, \tau, r, \partial\tilde{\Omega} \right) \left(\sum_{k=0}^\infty 2^{kn(\alpha\tau-1)} 2^{k(n-1)} \right)^{1/\tau}.$$

Therefore, if $\tau^{-1} = \alpha + 1/p$, then

$$\alpha < \frac{1}{p(n-1)} \Rightarrow |f|_{B_k^{\alpha, r}(\mathcal{T})} < \infty.$$

◇

Mutual Information Based Matching for Causal Inference with Observational Data

Lei Sun

*Department of Industrial and Systems Engineering
University at Buffalo, Buffalo, NY 14260, USA*

LEISUN@BUFFALO.EDU

Alexander G. Nikolaev

*Department of Industrial and Systems Engineering
University at Buffalo, 312 Bell Hall, Buffalo, NY 14260, USA
Department of Computer Science and Information Systems
University of Jyväskylä, Jyväskylä, FIN-40014, Finland*

ANIKOLAE@BUFFALO.EDU

Editor: Peter Spirtes

Abstract

This paper presents an information theory-driven matching methodology for making causal inference from observational data. The paper adopts a “potential outcomes framework” view on evaluating the strength of cause-effect relationships: the population-wide average effects of binary treatments are estimated by comparing two groups of units – the treated and untreated (control). To reduce the bias in such treatment effect estimation, one has to compose a control group in such a way that across the compared groups of units, treatment is independent of the units’ covariates. This requirement gives rise to a subset selection / matching problem. This paper presents the models and algorithms that solve the matching problem by minimizing the mutual information (MI) between the covariates and the treatment variable. Such a formulation becomes tractable thanks to the derived optimality conditions that tackle the non-linearity of the sample-based MI function. Computational experiments with mixed integer-programming formulations and four matching algorithms demonstrate the utility of MI based matching for causal inference studies. The algorithmic developments culminate in a matching heuristic that allows for balancing the compared groups in polynomial (close to linear) time, thus allowing for treatment effect estimation with large data sets.

Keywords: Observational Causal Inference, Mutual Information, Matching, Subset Selection, Optimization

1. Introduction

The tools for making inference based on observational data are useful for estimating the effects of binary treatments that are non-randomly assigned to the units of a studied population (Cochran, 1965). Causal investigations are of importance in various domains of science including economics (Abadie and Imbens, 2006), medical research (da Veiga and Wilder, 2008), political science (Ho et al., 2007), sociology (Morgan and Harding, 2006), law (Rubin, 2001), etc. As a conventional recipe, *matching* of treated and untreated units allows one to compare them and distill the effect of the treatment, while blocking the effects of confounding unit covariates.

The most widely adopted conventional matching methods employ various distance metrics (e.g., Mahalanobis distance) and propensity scores (see Section 2 for a detailed review); the success of a matching venture is typically assessed by checking if the compared groups are “well-balanced”, i.e., if the distributions of covariates within them are similar. The methods introduced more recently strive to directly optimize balance (Zubizarreta, 2012). In particular, Nikolaev et al. (2013) re-cast matching as a subset selection problem with the objective to optimize a measure of covariate balance across groups (as opposed to individual unit pairs). The approach was coined Balance Optimization Subset Selection, with its applicability illustrated by employing linear programming models (Nikolaev et al., 2013) and simulated annealing heuristics (Tam Cho et al., 2013).

Note, however, that improving balance, expressed via some metric(s) capturing the difference between the distributions of covariates in the compared groups, is just one approach that defines a matching procedure objective. It is as good as any other approach that would achieve the reduction of the dependence between the covariates and the treatment variable in the matched groups. This observation is exploited in the present paper, as it explores a new form of covariate balance and an alternative approach to doing matching.

This paper frames matching as an optimization problem with a mutual information (MI) based objective. The presented methods are non-parametric, and hence, do not suffer from human bias in model selection. The value of information theory in empirical statistics research and computer science has been emphasized over the past decade (Burnham and Anderson, 2002). However, while this thrust has been successful in facilitating hypothesis testing, optimization problems with information measures have proven to be difficult, mainly due to the inherent non-linearity of entropy and MI functions (Shannon, 1948). This paper presents a way to treat such non-linearity in subset selection problems, which arise in applying information theory logic for making causal inference with observational data.

MI has been used to formulate various problems involving feature selection (Estévez et al., 2009), dependency analysis (Kraskov et al., 2004) and chaotic data identification (Fraser and Swinney, 1986). It measures the level of dependence between random variables; e.g., when evaluated for two variables, it takes a high value when one random variable contains much information about the other, signifying high dependence, while zero MI implies that the variables are independent. We show that the difference between the covariate distributions among the treated and untreated units can be directly evaluated MI, exploiting the fact that randomization in treatment assignment implies zero MI between covariates and the treatment variable. To the best of the authors’ knowledge, no MI based method has yet been employed for grouping observations (units) to achieve a particular group property – most likely due to the non-linearity in the expression defining MI. This paper tackles this challenge and offers the models and algorithms that make theoretical and practical advances in subset selection, or simply, matching for treatment effect estimation.

While some optimization methods have already been employed for the methodological developments in causal inference with observational data (Hansen, 2004), the use of mathematical programming techniques for statistics-oriented applications is still rare. One such notable contribution is due to Bertsimas and Shioda (2007) who re-framed the classification and regression problems using integer programming. Similar to their efforts, this paper motivates the use of non-linear integer programming techniques in causal inference research.

First, this paper identifies pathways for the effective use of information theoretic measures (namely, MI) in optimization problems. The presented theoretical analysis techniques for treating non-linearity are generic, and hence, can be adopted in other applications, where making assumptions on model/data structures is undesirable. More generally, this paper may open up venues for the application of mathematical programming and optimization techniques in information theory itself.

Second, this paper explains how MI can serve as the basis of a new form of covariate balance. The resulting MI-based matching method for selecting control groups for causal inference is flexible in that it can achieve solutions of pre-specified quality, with pre-set control group size, – moreover, it can optimize the latter. The presented algorithmic developments produce a matching heuristic that runs in polynomial (close to linear) time: it thus allows for causal effect estimation with large data sets that are nowadays becoming available through mining social networks, health records, etc. While this work is not the first effort to employ the information theoretic tools for the needs of causal inference Hainmüller (2012), it appears to be the first where mutual information is used as an optimization objective.

The paper is organized as follows. Section 2 explains the problem of causal inference with observational data, and motivates optimization-driven subset selection approaches to attacking it. Section 3 introduces a class of MI-based matching problems with different objectives. Section 4 derives optimality conditions for matched groups using MI, and presents the mixed integer programming-based and sequential selection-based matching algorithms that work to balance the covariate distributions across the treatment and control groups. Section 5 showcases the practical value of the MI-based matching approach by comparing the designed algorithms’ performance against the best previously existing matching methods. Section 6 discusses the MIM limitations and future research directions. Section 7 provides concluding remarks and discusses the promising extensions of this line of work.

2. Causal Inference with Observation Data

Observational studies are often the only source of information about a program, policy, or treatment. For example, people non-randomly choose to participate in economy-boosting programs, political movements, online activities such as post re-tweeting, question answering, service subscription, etc. In estimating any causal effect with such data, the researchers resort to the nonparametric data preprocessing, commonly referred to as matching (Ho et al., 2011).

In a real-world causal inference problem instance, a treatment group (a group of *treated* units) is typically smaller than the size of a pool of available *control* (untreated) units: a control *group* can then be selected by a researcher from this pool. When a matching procedure is performed (Rubin, 2006), a control group is designed to contain the units that are similar in covariate values to those in the treatment group (differing only on the treatment indicators). A rule-of-thumb for evaluating the success of a matching procedure posits that better balance on covariates leads to smaller bias in the treatment effect estimation (Rosenbaum and Rubin, 1985); here, balance is understood as similarity between the empirical covariate distributions in the treatment and control groups. Note that theoretically, if an optimal matching does not exist, no guarantee as to the bias reduction amount can be given.

Among different types of matching recipes, the first proposed and well-used one is the nearest neighbor matching (Rubin, 1973). It prescribes to pair up each observed treatment unit with a control unit so as to minimize a weighted distance between the units’ covariate vectors in each such pair. Mahalanobis distance is widely used for this purpose (Rubin, 1980), however, as a measure of divergence, it relies on elliptical distributions of covariates (Sekhon, 2008). Another widely-used recipe prescribes to match units on propensity score (Rosenbaum and Rubin, 1983) defined as the probability of a unit to receive treatment.

The Mahalanobis distance and propensity score based matching methods can be combined in various ways (Rubin, 2001; Diamond and Sekhon, 2013). However, such methods require assumptions on model and/or data structure. As such, true units’ propensity score values are generally unknown, and must be estimated via regression on covariates, which makes room for the researcher’s bias in data analysis (when one can “thinker with” with an analysis tool to make it output the result that one anticipates, perhaps subconsciously). This weakness has led to controversial exchanges between the authors analyzing the same data and reaching conflicting conclusions (Dehejia and Wahba, 1999, 2002; Smith and Todd, 2005b; Dehejia, 2005; Smith and Todd, 2005a).

Both the Mahalanobis distance and propensity score based matching methods are applied with the objective to minimize the differences between the units in the treatment group and the control group of the same size. In contrast, Iacus et al. (2012) introduce a new class of matching methods, the Monotonic Imbalance Bounding (MIB) matching, which looks to assemble matched control groups consisting of a sufficiently large number of observations with a fixed pre-set level of maximum allowed imbalance. Based on the imbalance level, an algorithm is designed to split the range of each covariate into several coarse categories, so that any exact matching algorithm can be applied to solve this discretized problem.

Methodologies for direct optimization of balance have been proposed by researchers just recently. Rosenbaum et al. (2007) introduce a *fine balance* method, where exact balance is sought on several categorized nominal covariates and approximate matching is conducted on the remaining ones. For the exact matching part, a matrix of Mahalanobis distance values across all pairs of treatment and control units is defined, and then the classic assignment algorithm is used to minimize the total distance. Nikolaev et al. (2013) introduce Balance Optimization Subset Selection (BOSS) approach, optimizing explicit measures of balance and treating several models with exact and heuristic methods. Zubizarreta (2012) builds mixed integer programming models to optimize covariate balance directly by minimizing the total sum of the distances between the treated units and matched control units. The latter two lines of research work to measure the difference between the covariate distributions in the treatment group and control pool by employing chi-square, correlations, quantiles and Kolmogorov-Smirnov statistics, which are fundamentally different from the information theory-driven approach developed in the present paper.

3. Problem Definition

This section begins by presenting several matching problems, using illustrative examples, and explains how mutual information can guide a matching process. Then, relying on the mutual information function, nonlinear integer optimization problems are formally stated.

3.1 Motivating the Use of Matching for Causal Inference: Problem Statements

Given a set of observed units that have been treated, termed a treatment pool, and a set of observed untreated units, termed a control pool \mathcal{C} , the causal inference problem objective is to evaluate the degree of influence of the treatment on the population units, termed treatment effect. For an observable unit u , let Y_u^1 (Y_u^0) denote a treated (untreated) response and t_u a treatment indicator (1 means treated, 0 means not treated). Per Rubin's model of causal inference, these responses are referred to as *potential outcomes*, reflecting the fact that it is impossible to observe both Y_u^1 and Y_u^0 on the same unit u (Holland, 1986). For this reason, in estimating the population-wide effects of a treatment, researchers have to resort to comparing the averages across the treatment and control groups (Holland, 1986). One commonly targeted quantity of interest in causal inference studies, and the one this paper focuses on, is the average treatment effect for the treated (ATT), $E(Y^1|t=1) - E(Y^0|t=1)$, i.e., the average effect of treatment on the units that actually receive it.

Assume that a treatment group, $\mathcal{T} : |\mathcal{T}| < |\mathcal{C}|$, is given (randomly selected from a treatment pool), so $E(Y^1|t=1)$ can be estimated directly. A decision has to be made about selecting a control subset $\mathcal{S} \subset \mathcal{C}$ so that the units in \mathcal{T} and \mathcal{S} can be compared. If the two groups have the same distribution of covariates, one can use the value $E(Y^0|t=0)$ over \mathcal{S} as an estimate of $E(Y^0|t=1)$ over the entire population (refer Rosenbaum and Rubin (1983) for more statistical fundamental work), and then, obtain an estimate of ATT.

The goal of a matching procedure is to ensure that the covariate distributions in the treatment and control groups are as similar as possible. The key insight this paper exploits is that, if a matching procedure is successful, then it should make it impossible to distinguish the treatment units from the control units based on the covariates, or, in other words, learn the treatment status of an observation based on the information captured by its covariate values. For example, randomization guarantees that the treated and control units are indistinguishable by making the covariate distributions in both groups be identical to that in the whole population; in other words, randomization tends to balance covariates on expectation. The information about the treatment captured in the covariates can be quantified as the MI between the covariates and the treatment variable, and more specifically, expressed using either the joint covariate distribution or the marginal covariate distributions. This paper considers both these formulations, separately.

Let K be the set of covariates. For an observed unit, the $|K|$ -dimensional covariate vector is denoted by $\mathbf{X} = \{X_1, X_2, \dots, X_{|K|}\}$. Assume that every covariate is or can be made categorical. The discretization of continuous covariates can be accomplished by applying a binning scheme (Iacus et al., 2012; Nikolaev et al., 2013) to divide the range of values for each such covariate into a fixed set of intervals. These categorical or interval bounds partition the covariate hyperspace into subspaces. Define a marginal bin as the largest subspace associated with an interval from a covariate's range, and a joint bin as a covariate subspace that is not further subdivided into any smaller subspaces. Then, by design, a joint bin is an intersection of $|K|$ marginal bins, and every observed unit is contained in one such bin. Let b denote a joint bin, B denote the set of all joint bins, m denote a marginal bin and M denote the set of all marginal bins. With the binning scheme, units with covariate values falling into the same joint bin can no longer be distinguished from each other.

(a) Treatment group		(b) Matching on joint distribution		(c) Matching on marginal distribution	
10	10	10	10	11	9
15		16		15	
	2		1(*)		11
	10		10		

Figure 1: Different control groups selected when the perfect matching cannot be achieved due to the lack of control units in bin (*).

Note that the matching problem is trivial if there exists a control group that perfectly matches the treatment group (i.e., the empirical covariate distributions in the groups are identical). Consider the treatment group in Figure 1a, where the two-dimensional grid (built for two covariates) contains in its cells, termed bins, the number of units found in each bin. If a perfect matching of the control units to the treated ones does not exist, then the selection of a good control group becomes challenging. When a joint distribution is used to capture the dependence between the treatment variable and the covariates, the joint bins can be viewed as being independent and all equally important for representing the distribution. A good matching method should select some control units to form a group with a minimum loss in the joint distribution (Figure 1b). On the other hand, since the joint bins are formed as the intersections of $|K|$ multiple marginal bins, the assumption of the independence between the bins may not be well justified. Then, one can take an alternative approach and select the control group that achieves the best matching in all the marginal distributions (Figure 1c), albeit sacrificing some information captured in the copula. In summary, the problems of matching on the joint or marginal distributions each have their pros and cons, which is why the ensuing computational studies use and compare them both for treatment effect estimation (see Section 5).

3.2 Nonlinear Integer Optimization Problems

The objective of our matching problem is to select such a subset $S \subseteq C$ that minimizes the MI between the treatment indicator and covariate vector over set $S \cup T$. The MI between t and \mathbf{X} (or all the X_k) is denoted by $I(t; \mathbf{X})$ if the computation is based on the full joint distribution of the covariates, and by $\sum_{k \in K} I(t; X_k)$ if the computation is based on the marginal distributions of individual covariates. Since these expressions have similar mathematical forms, only $I(t, X)$ will be used for notation in the following discussion, with X representing either \mathbf{X} or X_k , depending on the context. Note that $I(t, X)$ is an unambiguous notation for MI in a problem with a single covariate. Meanwhile, for a problem with multiple covariates, the units in the joint or marginal bins can be thought of as being projected

into a one-dimensional range, and hence, can also be treated as a single-covariate problem, albeit possibly with the additional constraints capturing the copula-based dependencies.

In order to express $I(t; X)$ using the empirical covariate distribution for the units in a given problem, denote the covariate value for any unit contained in bin b by the same variable X_b . Let $p(b)$ be the probability that a unit is treated, and $p(X_b)$ be the probability that its covariate value falls into bin b , with $\sum_{b \in B} p(X_b) = 1$. Also, let $p(X_b, t)$ be the probability that the covariate value of a unit with treatment indicator t falls into bin b . Then, the empirical MI between the treatment indicator t and covariate X can be expressed as

$$I(t; X) = \sum_{b \in B} \sum_{t \in \{0,1\}} p(X_b, t) \log \frac{p(X_b, t)}{p(X_b)p(t)}. \quad (1)$$

Let S_b (or T_b , C_b) denote the number of units in group S (or T , C) with covariate values falling into bin b . From the characteristics of the units in $S \cup T$, the probabilities in equation (1) can be estimated. If $t = 0$, $p(X_b, t) = \frac{S_b}{|S|+|T|}$ and $p(t) = \frac{|S|}{|S|+|T|}$; if $t = 1$, $p(X_b, t) = \frac{T_b}{|S|+|T|}$ and $p(t) = \frac{T_b}{|S|+|T|}$; also, $p(X_b) = \frac{T_b + S_b}{|S|+|T|}$.

In general, an MI estimation bias (which is different from the causal estimation bias discussed above) arises when the MI estimation is done based on a fixed limited number of observations (1) (Panzetti and Tereves, 1996; Ronitston, 1999). However, this paper analyzes the *empirical* distributions of the variables defined for the units in the control and treatment groups, which are available in their entirety, and hence, by (1), the MI is exactly given,

$$I(t; X) = \log(|S| + |T|) + \frac{1}{|S| + |T|} \left(\sum_{b \in B} S_b [\log S_b - \log(T_b + S_b)] - \log|S| \right) + \sum_{b \in B} T_b [\log T_b - \log(T_b + S_b)] - \log|T|. \quad (2)$$

Two alternative MI-based objective functions are analyzed in this paper: $I(t; \mathbf{X})$ and $\sum_{k \in K} I(t; X_k)$. Formally, a problem from the class of **Mutual Information based Matching (MIMM)** problems is stated:

Given: $|K|$ covariates; treatment group T ; control pool C with $|C| > |T|$; for each observed unit $u \in T \cup C$, the covariate vectors $\mathbf{X} = \{X_1, X_2, \dots, X_{|K|}\}$; segmented covariate space with joint bins $b \in B$ and marginal bins $m \in M$; a fixed integer N as the target control group size.

Objective: find a subset $S \subseteq C$ such that

- $|S| = N$ and $I(t; \mathbf{X})$ is minimized (MIM-Joint problem), or
- $|S| = N$ and $\sum_{k \in K} I(t; X_k)$ is minimized (MIM-Marginal problem).

A matching problem based on either joint or marginal covariate distribution(s) is designed with the decision variables returning the number of control units to be selected from each joint bin. Complete enumeration of feasible solutions in a problem with any of these two objective types would take an exponentially growing number of computing operations in the size of the control pool. Another challenge lies in the nonlinearity of the objective functions, further analysis of which is required in order to arrive at tractable mathematical programming formulations for MIM.

Theorem 1 *The decision version of the MIM-Marginal problem, $\min_{S \subseteq C} \sum_{k \in K} I(t; X_k)$ subject to $|S| = N$, is NP-complete.*

Proof See Appendix A. ■

4. Solution Approaches

This section investigates the properties of solutions with the minimum MI, with the goal of developing a method for treating the nonlinearity in the objective function of MIM problems. The derivations presented in this section unfold from the problem of minimizing $I(t; X)$ under the assumption that the contents of the bins capturing the distribution of covariate X are independent. The obtained insights are next extended to the MIM-Joint and MIM-Marginal problems. The mixed integer programming models and matching algorithms are then developed for selecting control subsets for MIM-Joint and MIM-Marginal problems.

4.1 Analyses of Optimality Conditions

Consider the expression of MI in (2): observe that since the treatment group is given, and the target control group size is known, $|S| = N$, several terms in equation (2) are constant. Also, $\sum_{b \in B} S_b \log|S| + \sum_{b \in B} T_b \log|T| = |T| \log|T| + |S| \log|S|$. Then, the term $\sum_{b \in B} S_b [\log S_b - \log(T_b + S_b)] + \sum_{b \in B} T_b [\log T_b - \log(T_b + S_b)]$ remains the only one to be considered for MI minimization. For the ease of presentation, this term can now be rewritten based not on the bins' aggregate contents but on the individual units' locations in the bins. Because all the observed units, whose covariate values X^u are contained in the same bin, have the same values of T_b and S_b , the minimization of (2) is equivalent to that of

$$R \equiv \prod_{u \in S, X^u \in b} \frac{S_b}{T_b + S_b} \prod_{u \in T, X^u \in b} \frac{T_b}{T_b + S_b}. \quad (3)$$

Consider the MIM problem instance illustrated by Figure 2, where $N - 1$ control units have been selected from the control pool into a control group (not necessarily optimally). In order to complete the selection of units into the control group, one last unit has to be selected from any of the bins with $S_b < C_b$. All such bins can be partitioned into three subsets: $B^1 = \{b : S_b < T_b\}$, $B^2 = \{b : S_b \geq T_b, T_b \neq 0\}$, $B^3 = \{b : T_b = 0\}$. Given that the last unit added to the control group is contained in bin b , let I_b denote the resulting MI between t and X , and R_b denote the resulting objective function value in (3).

The following two lemmas provide the guidelines for the optimal selection of the last unit to be included into the control group.

Lemma 2 *Consider an instance where an incomplete control group has $N - 1$ units in it, and three candidate units (that could complete it) are contained in bins $b_1 \in B^1$, $b_2 \in B^2$ and $b_3 \in B^3$, respectively. Then, $I_1 < I_2 < I_3$.*

Proof If the candidate unit from bin $b_1 \in B^1$ is selected, then the value of S_b increases by 1, while all the other S_b and T_b values stay unchanged. Thus, the objective function value in (3) becomes $R_1 = \hat{R} \left(\frac{S_{b_1} + 1}{T_{b_1} + S_{b_1} + 1} \right)^{S_{b_1} + 1} \left(\frac{T_{b_1}}{T_{b_1} + S_{b_1} + 1} \right)^{T_{b_1}} \left(\frac{S_{b_2}}{T_{b_2} + S_{b_2}} \right)^{S_{b_2}} \left(\frac{T_{b_2}}{T_{b_2} + S_{b_2}} \right)^{T_{b_2}}$,

$I(t; X)$, with $\frac{S_{b_1-1-\Delta-A}}{T_{b_1}} < \frac{S_{b_1-1-A}}{T_{b_1}} < \frac{S_{b_2-A}}{T_{b_2}} < \frac{S_{b_2+\Delta-A}}{T_{b_2}}$ holding for $\forall \Delta > 0$. Note again that b_1 and b_2 were arbitrarily picked. Such unit shuffling (i.e., removal and addition) operations can repeat until S is modified to become identical to S^* . Since in this process, $I(t; X)$ increases with every shuffle, then S^* could not be optimal, which is a contradiction. If $\frac{S_{b_1-1-A}}{T_{b_1}} = \frac{S_{b_2-A}}{T_{b_2}}$, then as a result of removing a unit from b_1 and adding one into b_2 , $I(t; X)$ will not change. In such a case, if the updated S becomes identical to S^* , then this means that S is an alternative optimal solution with the minimum $I(t; X)$. Otherwise, one can continue shuffling units, with $\frac{S_{b_1-1-\Delta-A}}{T_{b_1}} < \frac{S_{b_1-1-A}}{T_{b_1}} = \frac{S_{b_2-A}}{T_{b_2}} < \frac{S_{b_2+\Delta-A}}{T_{b_2}}$ holding for $\forall \Delta > 0$. Similarly, $I(t; X)$ will continue increasing, leading to S^* not being optimal, i.e., to a contradiction.

Theorem 4 provides the necessary and sufficient optimality conditions for the control groups with the minimum $I(t; X)$. Its value lies in condition (4) being linear in S_{b_i} unlike the minimization problem objective (2). Note, however, that Theorem 4 only works to determine whether a control group is optimal or not; it cannot be used to assess or compare the quality of suboptimal control groups.

In order to effectively apply Theorem 4 in practice, one would like to avoid the exhaustive traversal of bin pairs. Corollaries 5 and 6 allow for tackling this problem and provide a means for efficient optimal control group selection.

Corollary 5 Consider an instance of minimizing $I(t; X)$ where $N > \sum_{b \in \{b: T_b \geq 1\}} C_b$. Then, a control group S is optimal if it includes all the control units in all $b \in \{b : T_b \geq 1\}$.

Proof Follows directly from Lemma 2.

Corollary 6 Consider an instance of minimizing $I(t; X)$, where $N \leq \sum_{b \in \{b: T_b \geq 1\}} C_b$. Then, a control group S is optimal if and only if for every pair of bins b_1 and b_2 such that

$$b_1 \in \operatorname{argmax}_{b \in B} \left\{ \frac{S_b - 1 - A}{T_b} \right\} \quad (5)$$

and

$$b_2 \in \operatorname{argmin}_{b \in B} \left\{ \frac{S_b - A}{T_b} : |C_b - S_b| \geq 1 \right\}, \quad (6)$$

one has $\frac{S_{b_1-1-A}}{T_{b_1}} \leq \frac{S_{b_2-A}}{T_{b_2}}$, where $A \approx -0.47$.

Proof Consider any pair of bins, b_2 and b_1 , with $|C_{b_1} - S_{b_1}| \geq 1$. In order to determine if S is optimal, Theorem 4 prescribes to compare the left-hand and right-hand sides of inequality (4) for bins b_2 and b_1 . By the statement of this corollary, one has $\frac{S_{b_1-1-A}}{T_{b_1}} \geq \frac{S_{b_2-A}}{T_{b_2}}$ and $\frac{S_{b_2-A}}{T_{b_2}} \leq \frac{S_{b_1-1-A}}{T_{b_1}}$. Then, if inequality (4) holds for bins b_1 and b_2 , then it also holds for bins B_3 and B_4 , because $\frac{S_{b_3-1-A}}{T_{b_3}} \leq \frac{S_{b_1-1-A}}{T_{b_1}} \leq \frac{S_{b_2-A}}{T_{b_2}} \leq \frac{S_{b_4-A}}{T_{b_4}}$, and vice versa.

4.2 Mixed Integer Programming-Based Matching Algorithms

The optimality conditions in Theorem 4 and Corollary 6 allow one to construct an alternative formulation for the problem of minimizing $I(t; X)$ with $N \leq \sum_{b \in \{b: T_b \geq 1\}} C_b$, using the expression $\frac{S_b - 1 - A}{T_b}$. Note that this ratio is undefined for bins with $T_b = 0$; however, per Lemma 2, an optimal solution can contain control units from such bins only if all the available control units from other bins have been exhausted. In order to reformulate the objective function of minimizing $I(t; X)$, the expression $\frac{S_b - 1 - A}{T_b}$ should first be revised so that its denominator evaluates to a fixed number, $\alpha \in (0, 1)$, small enough to make the selection of control units from bins with $T_b = 0$ very costly. In order to search for a control group satisfying the condition in Corollary 6, the following optimization problem is formulated:

$$\min \left\{ \max_{b \in B} \frac{S_b - 1 - A}{T_b}, \alpha \right\}, \quad (7)$$

where α is a positive parameter small enough to distinguish $T_b = 0$ from other positive values of T_b , e.g., $\alpha = 0.01$.

By solving (7), one can work to construct an optimal control group through minimizing the maximum value of the function in (5). Having found an optimal solution to (7), one can check if (for this solution) the set in (5) is a singleton. If it is, then the condition in Corollary 6 holds. Otherwise, satisfying (7) may not be sufficient for satisfying Corollary 6, since it requires one to check every pair of bins in both the set in (5) and the set in (6). As an example of this situation, suppose that there exist two bins, b_1 and b_2 , in the set in (5), and a bin b_3 in the set in (6) such that $\frac{S_{b_1-1-A}}{T_{b_1}} = \frac{S_{b_2-1-A}}{T_{b_2}} > \frac{S_{b_3-1-A}}{T_{b_3}}$. If a unit is removed from b_1 while another unit is added into b_2 , the objective value of (7) does not improve because $\frac{S_{b_2-1-A}}{T_{b_2}}$ does not decrease. Thus, the optimization process based purely on solving (7) would terminate early without guaranteeing an optimal matching.

To handle the situation where the set in (5) is not a singleton for a solution of (7), an algorithm is developed to iteratively solve for the optimal number of units to be selected from each bin. In any iteration, if solving (7) returns multiple bins with values of $\frac{S_b - 1 - A}{T_b}$ equal to the maximum (over all the bins), then one of these bins is added to a ‘‘forbidden bin set’’, denoted by B^F and initialized at an empty set before the first iteration. Every time B^F is updated, problem (7) is reformulated, with all the bins that are not in B^F , and solved again in the next iteration. After several such iterations, once the set in (5) is found to be a singleton for a solution to (7), one can be sure that an optimal control group has been found. In order to ensure that the unit picking in a given iteration does not mess up the optimality achieved within any bin in the previous iteration(s), a bin with the smallest number of the treatment units in the non-singleton set (5) is always fixed first. In every iteration, (7) is solved as a mixed integer programming (MIP) model,

$$\min \quad q \quad (8)$$

$$s.t. \quad q \geq \frac{S_b - 1 - A}{\max\{T_b, \alpha\}} \quad \forall b \notin B^F, \quad (9)$$

$$\sum_{b \in B} S_b = N, \quad (10)$$

$$\begin{aligned}
 S_b &\leq C_b \quad \forall b \in B, & (11) \\
 S_b &\geq 0 \quad \forall b \in B, & (12) \\
 S_b &: \text{integer} \quad \forall b \in B, & (13) \\
 B &= \{b : C_b + T_b \geq 1\}. & (14)
 \end{aligned}$$

The decision variables in this MIP are the numbers of the control units, S_b , to be selected from each bin. Since it is only necessary to consider the bins in $\{b : C_b + T_b \geq 1\}$, then despite the fact that the total number of joint bins grows exponentially with the binning partition granularity, the number of the decision variables is bounded by $|T| + |C|$. The contents of the forbidden bin set B^F are updated iteratively in the described algorithm. The minimax optimization problem (7) is formulated with the objective function (8) and the constraint set (9). Constraint (10) ensures that the total number of units in the control group is equal to N . Constraints (11), (12) and (13) restrict the range of S_b to nonnegative integers not exceeding the number of available control units in each respective bin.

Note that for solving any MIM-Joint problem, since the bins' contents can be treated as being independent from each other, the described procedure for minimizing $I(t; X)$ can be exactly followed to minimize $I(t; \mathbf{X})$, with bins b in (8)-(14) being the joint bins.

Algorithm 1 MIP-based matching for MIM-Joint problem

- 1: Initialize the bin set $\{b : C_b + T_b \geq 1\}$ consisting of all the bins occupied by the units in $\mathcal{T} \cup \mathcal{C}$; compute T_b and C_b ; forbidden bin set $B^F = \emptyset$.
 - 2: Update and solve the corresponding instantiation of formulation (8)-(14) to obtain S_b for every bin b .
 - 3: If $\text{argmax}_{b \notin B^F} \{\frac{S_b - 1 - A}{T_b}\}$ is a singleton, go to step 4. Otherwise, add the bin with the smallest number of treatment units in $\text{argmax}_{b \notin B^F} \{\frac{S_b - 1 - A}{T_b}\}$ into set B^F , record and fix the optimal number of control units to be selected in it, and go to step 2.
 - 4: Construct a control group complying with the obtained values of S_b over all the initialized bins b . Stop.
-

However, for solving an MIM-Marginal problem, modifications to the above formulation and the algorithm are necessary due to the fact that the marginal bins cannot be assumed independent. The decision variables of an MIP-Marginal model are the numbers of control units to be selected into \mathcal{S} for every joint bin (b still denotes a joint bin), and constraints (10)-(14) remain a part of the optimization problem. Let m denote a marginal bin, M^F denote the forbidden marginal bin set, and T_m , C_m and S_m denote the number of all the treatment units, number of all the control units and number of the selected control units in m , respectively. Equation (9) is then replaced by (15). Also, an additional constraint (16) is added to the formulation to ensure that the number of units in any marginal bin equals the summed total number of units in all the corresponding joint bins.

$$q \geq \frac{S_m - 1 - A}{\max\{T_m, \alpha\}} \quad \forall m \notin M^F, \quad (15)$$

$$S_m = \sum_{b: X_m \in b} S_b \quad \forall m \in \{m : C_m + T_m \geq 1\} \quad (16)$$

Recall that in every iteration of solving MIM-Joint using Algorithm 1, one checks whether the set of bins with the maximum value of $\frac{S_b - 1 - A}{T_b}$ is a singleton. Because of the dependence between the contents of marginal bins, this condition by itself does not guarantees optimality for MIM-Marginal problem. Specifically, given a feasible solution to an MIM-Marginal problem, if exactly one marginal bin is found to achieve the maximum value of $\frac{S_b - 1 - A}{T_b}$ and this marginal bin is associated with covariate k , then because the bins in the same covariate are independent and according to Corollary 6, $I(t; X_k)$ is minimized. But the MI in other covariates might still be improved without changing $I(t; X_k)$, e.g., by adding and removing the same number of control units to/from the same marginal bin in covariate k . Thus, while solving for the optimal number of units in each marginal bin, and adding the marginal bins one-by-one to a forbidden bin set, one should not stop until the sets of bins with the maximum values of $\frac{S_b - 1 - A}{T_b}$ in all the $|K|$ covariates become singletons.

Algorithm 2 MIP-based matching for MIM-Marginal problem

- 1: Initialize the joint bin set $\{b : C_b + T_b \geq 1\}$ and the marginal bin set $\{m : C_m + T_m \geq 1\}$ consisting of all the bins occupied by the units in $\mathcal{T} \cup \mathcal{C}$; compute T_b , C_b , T_m and C_m ; forbidden bin set $M^F = \emptyset$.
 - 2: Update and solve the corresponding instantiation of formulation (8), (10)-(16) to obtain S_b for every joint bin b .
 - 3: If $\text{argmax}_{m \notin M^F} \{\frac{S_m - 1 - A}{T_m}\}$ is a singleton for every covariate, go to step 4. Otherwise, add the marginal bin with the smallest number of treatment units in $\text{argmax}_{m \notin M^F} \{\frac{S_m - 1 - A}{T_m}\}$ into set M^F , record and fix the optimal number of control units to be selected in it, and go to step 2.
 - 4: Construct a control group complying with the obtained values of S_b over all the initialized bins b . Stop.
-

4.3 Sequential Selection Matching Algorithms

Based on the optimality conditions in Theorem 4, Algorithms 1 and 2 are guaranteed to achieve best matched control groups for MIM-fixed and MIM-marginal problem instances. However, their MIPs may become difficult to solve for problems of large size, which, however, can be avoided by utilizing the result captured in Theorem 7, presented for the problem of minimizing $I(t; X)$.

Theorem 7 *If control group \mathcal{S} has the minimum $I(t; X)$ among all the control groups of size N , then a group with the minimum $I(t; X)$ among all the control groups of size $N + 1$ ($N - 1$) can be obtained from \mathcal{S} by adding to it a single unit from bin $b \in \text{argmin}_{b \in B} \{\frac{S_b - A}{T_b} : |C_b - S_b| \geq 1\}$ (removing from it a single unit from bin $b \in \text{argmax}_{b \in B} \{\frac{S_b - 1 - A}{T_b}\}$).*

Proof Let \mathcal{S}^+ denote a control group obtained from \mathcal{S} by adding to it a single unit from bin $b \in \text{argmin}_{b \in B} \{\frac{S_b - A}{T_b} : |C_b - S_b| \geq 1\}$. According to Lemma 3, the MI between \mathcal{T} and X over set $\mathcal{T} \cup \mathcal{S}^+$ is minimal among all the groups that can be built on \mathcal{S} . The following proof will show \mathcal{S}^+ is also globally optimal.

Let S_b^+ denote the number of control units selected into S^+ in bin b . Let b_1 and b_2 be two bins such that $b_1 \in \operatorname{argmax}_{b \in B} \{\frac{S_b^{+-A}}{T_b}\}$ and $|C_{b_2} - S_{b_2}^+| \geq 1$. If b_1 is the bin where S^+ has one more unit than S , then $S_{b_1}^+ - 1 = S_{b_1}$ and $S_{b_2}^+ = S_{b_2}$, and then $\frac{S_{b_1}^{+-A}}{T_{b_1}} = \frac{S_{b_1}^+ - A}{T_{b_1}} \leq \frac{S_{b_2}^+ - A}{T_{b_2}} = \frac{S_{b_2}^+ - A}{T_{b_2}}$. Note that by Theorem 4, because S is optimal, one has $\frac{S_{b_1}^+ - A}{T_{b_1}} \leq \frac{S_{b_2}^+ - A}{T_{b_2}}$. If b_2 is the bin where S^+ has one more unit than S , then $S_{b_2}^+ - 1 = S_{b_2}$ and $S_{b_1}^+ = S_{b_1}$, and then $\frac{S_{b_1}^{+-A}}{T_{b_1}} = \frac{S_{b_1}^+ - A}{T_{b_1}} \leq \frac{S_{b_2}^+ - A}{T_{b_2}} < \frac{S_{b_2}^+ - A}{T_{b_2}}$. If S and S^+ have the same numbers of units in both b_1 and b_2 , then $\frac{S_{b_1}^+ - A}{T_{b_1}} = \frac{S_{b_1}^+ - A}{T_{b_1}} = \frac{S_{b_1}^+ - A}{T_{b_1}} \leq \frac{S_{b_2}^+ - A}{T_{b_2}} = \frac{S_{b_2}^+ - A}{T_{b_2}}$. Therefore, by Corollary 6, S^+ is a group with the minimum $I(t; X)$ among all the control groups of size $N + 1$.

Let S^- denote a control group obtained from S by removing a single unit from bin $b \in \operatorname{argmax}_{b \in B} \{\frac{S_b^{-A}}{T_b}\}$. Let S_b^- denote the number of control units selected into S^- in bin b . Let b_1 and b_2 be two bins such that $b_1 \in \operatorname{argmax}_{b \in B} \{\frac{S_b^{-A}}{T_b}\}$ and $|C_{b_2} - S_{b_2}^-| \geq 1$. If b_2 is the bin where S has one more unit than S^- , then $S_{b_2}^- - 1 = S_{b_2}$ and $S_{b_1}^- = S_{b_1}$, and then $\frac{S_{b_1}^{-A}}{T_{b_1}} = \frac{S_{b_1}^- - A}{T_{b_1}} \leq \frac{S_{b_2}^- - A}{T_{b_2}} = \frac{S_{b_2}^- - A}{T_{b_2}}$. Note that due to the optimality of S , $\frac{S_{b_1}^{-A}}{T_{b_1}} \leq \frac{S_{b_2}^- - A}{T_{b_2}}$. If b_1 is the bin where S has one more unit than S^- , then $S_{b_1}^- - 1 = S_{b_1}$ and $S_{b_2}^- = S_{b_2}$, and then $\frac{S_{b_1}^{-A}}{T_{b_1}} = \frac{S_{b_1}^- - A}{T_{b_1}} \leq \frac{S_{b_2}^- - A}{T_{b_2}} < \frac{S_{b_2}^- - A}{T_{b_2}}$. Therefore, by Corollary 6, S^- is a group with the minimum $I(t; X)$ among all the control groups of size $N - 1$. ■

Theorem 7 provides a method for finding optimal control groups for MIM problems, without solving any programming models. One can iteratively build control groups of increasing sizes until an optimal solution of the desired size N is obtained. Each control group in this process results in the minimum value of MI among all the groups of the same size. Also, Theorem 7 provides establishes a relationship between the optimal subsets for problems with different target control group sizes. This result will be important for seeking the minimum MI in the problems with an unrestricted (flexible) control group size.

For the MIM-Joint problem, since its bins are treated as independent from each other, Theorem 7 directly applies, and Algorithm 3 guarantees to return an optimal solution. Note that Algorithm 3 is polynomial. In the worst case, it needs to make comparisons of $\frac{S_b^{+-A}}{T_b}$ for N multiples of the number of bins occupied by treatment units.

With the MIM-Marginal problem, a challenge arises due to the dependence between the marginal bins' contents. By comparing the terms $\frac{S_{b_m}^{+-A}}{T_{b_m}}$, one can identify the marginal bin to which a control unit should be added, but one still needs to pick some joint bin. Even further, since the marginal bins on the same covariate are independent from each other, the comparison of $\frac{S_{b_m}^{+-A}}{T_{b_m}}$ can reveal the most favorable marginal bin for each covariate, but the bin that lies at the intersection of those $|K|$ marginal bins may not contain any control unit that could be added to the control group. Algorithm 4 offers an organized way

Algorithm 3 Sequential selection matching for MIM-Joint problem

- 1: Initialize the joint bin set $\{b : C_b + T_b \geq 1\}$ consisting of all the bins occupied by the units in $T \cup C$; compute T_b and C_b ; set $S_b = 0$ for all b .
- 2: Select a bin $b \in \operatorname{argmin}_{b \in B} \{\frac{S_b^{+-A}}{T_b} : |C_b - S_b| \geq 1\}$, update S_b by adding 1.
- 3: If N units are selected, go to step 4. Otherwise, go to step 2.
- 4: Construct a control group complying with the obtained values of S_b over all the initialized bins b . Stop.

to achieve good (but not necessarily optimal) solutions to MIM-Marginal instances in the following manner. For each joint unit, $|K|$ ratios of the form $\frac{S_{b_m}^{+-A}}{T_{b_m}}$ are evaluated (one per covariate); these $|K|$ ratios are organized in a descending order; then, the joint bin with the lexicographically minimal ratio gets one more unit added to it. Again, the resulting Algorithm 4 is an approximate method, but it is polynomial, and in practice, is found to return solutions of high quality for diverse matching problem instances (see Section 5).

Algorithm 4 Sequential selection matching for MIM-Marginal problem

- 1: Initialize the joint bin set $\{b : C_b + T_b \geq 1\}$ consisting of all the bins occupied by the units in $T \cup C$; compute T_b and C_b ; set $S_b = 0$ for all b .
 - 2: Update $\frac{S_{b_m}^{+-A}}{T_{b_m}}$ for each marginal bin, and order all associated $\frac{S_{b_m}^{+-A}}{T_{b_m}}$ by values in descend sequence for each joint bin.
 - 3: Find a bin b in set $\{b : |C_b - S_b| \geq 1\}$ such that its ordered set of $\frac{S_{b_m}^{+-A}}{T_{b_m}}$ is lexicographically minimal; increase the value of the decision variable, S_b , corresponding to this bin, by 1.
 - 4: If N units are selected, go to step 5. Otherwise, go to step 2.
 - 5: Construct a control group complying with the obtained values of S_b over all the initialized bins b . Stop.
-

The complexity of Algorithm 4 depends on the number of covariates $|K|$, the number of marginal bins $|M|$, treatment group size $|T|$, control pool size $|C|$ and target control group size N . In the worst case, every unit (treated or control) occupies one unique joint bin, with each joint bin contributing to $|K|$ marginal bins. The storage of these data requires a space of size $\mathcal{O}(|K|(|T| + |C|))$. In the binning step, both the treatment group and control pool are traversed, with each unit being assigned to the appropriate marginal and joint bins; this operation takes $\mathcal{O}(|M|(|T| + |C|))$ time. In the matching step, all the occupied joint bins are traversed for the lexicographic comparison, which takes $\mathcal{O}(N(|T| + |C|)|K|^2)$ time.

5. Computational Analyses

This section presents the results of the computational experiments with synthetic and real-world (Lalonde, 1986) data sets, evaluating the performance of the MIM method in estimating causal effects, and comparing it to the BOSS method (Nikolaev et al., 2013) and the widely used propensity score based matching method (Rosenbaum and Rubin, 1983).

5.1 Algorithm Performance Assessment

In order to evaluate the performance of the MIM algorithms, a series of tests is first conducted with the data set designed by Sauppe et al. (2014), which was found challenging for the existing matching methods¹. This synthetic data set with 25 covariates contains 100 treatment units and 10,000 control units. All the covariate values are drawn from normal distributions with mean 0. All the treatment and control units have the same, highly nonlinear response function. Thus, by design, the average treatment effect for the treated (ATT) for the created population is zero.

The experiments were conducted with the number of the considered covariates varied in the range from 1 to 25. In optimizing the covariate balance, Sauppe et al. (2014) uniformly partitioned the range of the observed unit values in each covariate into 20 bins, and used Balance Optimization Subset Selection (BOSS) for control group selection. In order to limit the instances resulting in large MIP formulations, Sauppe et al. (2014) adopted a time limit heuristic. They achieved quite well balanced control groups; however, the limited computational efficiency remains the key challenge for the existing BOSS methods, especially when the data sets to make inference from are very large.

With the same settings as in Sauppe et al. (2014), this section compares the performance of the following matching methods: the Mahalanobis distance-based one, the propensity score-based one, the BOSS methods from Sauppe et al. (2014), and three MIM methods – MIM-Joint, MIM-Marginal MIP, and MIM-Marginal sequential selection. Note that the first two of these methods are widely used and included in several existing matching packages, e.g., MatchIt (Ho et al., 2011) and optmatch (Hansen and Klopfer, 2012). We also used Coarsened Exact Matching (CEM) in MatchIt (Ho et al., 2011) and fullmatch method in optmatch (Hansen and Klopfer, 2012). However, CEM excluded many treatment units from the matched treatment group in the experiments with five or more covariates, and thus, was not found suitable for ATT estimation. Also, under the pre-set control group size, the results of fullmatch were no different from those of the standard Mahalanobis distance or propensity score matching (depending on the selected parameter settings).

The MIP models for MIM were solved using CPLEX. To reduce the runtime in solving some large-size problems, a heuristic was applied that utilized the solution of sequential selection to convert MIP to an integer program. Generally, MIP-based methods (be it for BOSS or MIM) are time-consuming. However, the sequential selection algorithm always runs very quickly: it took about 7 seconds on average to find solutions for the instances with 25 covariates on a desktop with an Intel Xeon E5-2420 1.9GHz CPU and 16G RAM.

Table 1 presents the ATT estimates obtained with the considered methods for the instances with the varied number of covariates; Figure 3 provides a graphical illustration of the results. Recall that by design, the ATT is zero, so the closer an estimate is to zero, the better. Table 2, Figure 4a and Figure 4b report the Kolmogorov-Smirnov (KS) test statistic scores and associated p -values for checking whether the underlying covariate distributions differ in the treated and control groups. In Table 2, column ‘‘Avg’’ reports the average

1. The data set, named 25c10k, features a highly nonlinear response function and is available in full in the online supplement of Sauppe et al. (2014). The response function in it is $y = 0.8x_1(1.0 - x_1) + 0.5x_2(0.7 + x_1) + 0.27x_3x_2 - 0.9x_4^2 + 0.7x_5(0.5 + x_5)x_2 - 0.6x_6x_1 + 0.4x_7 - 0.8x_8 + 0.6x_9(0.9 - x_9) + 0.2x_{10}^2(0.3 - x_7) + 0.5x_{11}^2 - 1.4x_{12} - 0.8x_{13} - 0.9x_{14}^2 + 0.5x_{15}^2(0.1 + x_{15}) + 0.8x_{16} - 0.9x_{17}(0.2 - x_{13}) + 1.5x_{18} - 1.2x_{19}(1.0 + x_{11}) + 0.7x_{20}(0.8 - x_{20}) - 0.5x_{21} - 1.3x_{22}(1.0 + x_{22}) + 1.1x_{23} - 1.2x_{24}(1.0 + x_{23}) + 0.4x_{25}^2(0.6 - x_{25}) + N(0, 1)$.

Table 1: Estimated treatment effects with different matching methods.

# Covariates	Mahalanobis metric	Propensity score	BOSS	MIM-Joint	MIM-Marginal (MIP)	MIM-Marginal (Sequential)
1	-0.133	-0.117	0.218	0.006	0.006	0.006
5	0.626	-1.223	0.045	6.361	0.005	1.721
10	0.39	5.437	-2.646	16.646	-1.123	0.517
15	6.261	19.126	3.074	30.593	2.590	2.403
20	10.164	-11.849	7.782	35.306	5.927	8.084
25	16.074	-14.753	6.618	50.643	12.170	8.448

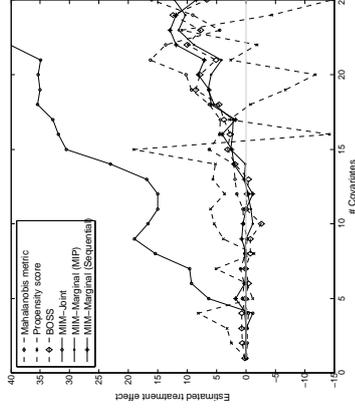


Figure 3: Trends for estimated treatment effects with different matching methods.

KS distance or p -value over all the covariates, and columns ‘‘Max’’ and ‘‘Min’’ report the maximum KS test statistic and minimum p -values, respectively. The smaller the KS score and the larger the p -value, the better balance is achieved.

In general, the MIM-Joint does not output accurate treatment effect estimates in the multi-covariate cases. If an exact or almost-exact matching solution exists, the MIM-Joint performs well, e.g., as in the one-covariate case. However, it is too sensitive to imbalance. As the number of the covariates grows, there remain fewer and fewer bins in which exact matching is possible, making the MIM-Joint formulation not-so-useful for most practical cases. At the same time, the marginal bin-based matching methods succeed in obtaining rather accurate ATT estimates.

Because of the high non-linearity of the response function, propensity score matching does not produce good ATT estimates, and there is no clear trend in its performance as it degrades. Mahalanobis metric matching, BOSS and two MIM-Marginal methods all produce similar estimates, with the MIM-Marginal MIP performing slightly better than the other methods. For the instances with fewer than 15 covariates, the estimates produced by these four methods are close zero (the true ATT value 0), but MIM-Marginal methods achieve much better balance in covariates judging by the KS test scores and p -values. For the

Table 2: Marginal balance quality for matching solutions.

# Covariates	Mahalanobis metric				Propensity score				BOSS			
	KS		pVal		KS		pVal		KS		pVal	
	Avg	Max	Avg	Min	Avg	Max	Avg	Min	Avg	Max	Avg	Min
1	0.010	0.010	1.000	1.000	0.020	0.020	1.000	1.000	0.060	0.060	0.994	0.994
5	0.062	0.070	0.984	0.967	0.158	0.180	0.190	0.078	0.058	0.070	0.991	0.967
10	0.112	0.160	0.588	0.155	0.154	0.200	0.271	0.037	0.067	0.090	0.947	0.813
15	0.127	0.180	0.452	0.078	0.162	0.200	0.190	0.037	0.068	0.100	0.942	0.699
20	0.140	0.230	0.413	0.010	0.166	0.230	0.193	0.010	0.087	0.150	0.790	0.211
25	0.140	0.230	0.393	0.010	0.155	0.230	0.252	0.010	0.098	0.190	0.708	0.054

MIM-Marginal (MIP)				MIM-Marginal (Sequential)			
KS		pVal		KS		pVal	
Avg	Max	Avg	Min	Avg	Max	Avg	Min
0.000	0.000	1.000	1.000	0.000	0.000	1.000	1.000
0.006	0.010	1.000	1.000	0.003	0.010	1.000	1.000
0.034	0.050	0.999	0.998	0.029	0.040	1.000	1.000
0.070	0.120	0.891	0.475	0.069	0.140	0.914	0.281
0.106	0.190	0.614	0.080	0.109	0.190	0.605	0.054
0.131	0.220	0.471	0.026	0.119	0.240	0.556	0.006

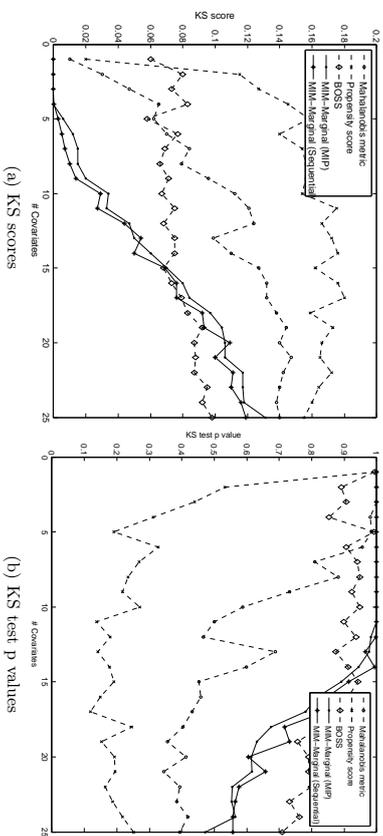


Figure 4: Trends for marginal balance quality for matching solutions.

instances with more than 15 covariates, the large number of bins makes for a large variance in the ATT estimates; comparing only the 100 treatment units and the 100 matched control units, one observes significant divergence between the ATT and its estimates obtained with all the methods, even though BOSS achieves the smallest KS scores. Considering both the matching quality and runtime performance, the MIM-Marginal with sequential selection algorithm comes out as the most efficient matching method for practical purposes.

5.2 The Experiences with Using Mutual Information as a Measure of Balance

The next set of experiments reveals that the MI function, employed as the objective in MIM, can be viewed as a surrogate measure of covariate balance. In order to trace the dependence between the MI values, obtained with different control groups, and the corresponding ATT estimates, an additional set of results is reported with the data set of Section 5.1 with 10 covariates. This set is also used to help us assess the impact of the MIM algorithm parameter settings on the matching quality.

For a large number of randomly generated control groups, the MI values were recorded together with the resulting treatment effect estimates (see Figure 5). Among these, four groups were found to have the MI less than 0.002 and at least 100 groups fell in each of the other intervals, into which the MI range was divided. Observe that, as the MI grows, the average of the ATT estimates tends to increase, and the standard deviation of the estimate values over the intervals grows as well. This confirms the premise that minimizing MI is a valid approach to guiding the matching process.

Another benefit of using MI lies in the ability to directly compare the matching problem solutions (control groups) of different sizes. Indeed, with the empirical covariate distribution in a given treatment group being fixed, the decision-maker has the freedom of selecting the target control group size. With the same binning scheme, the MI values obtained with

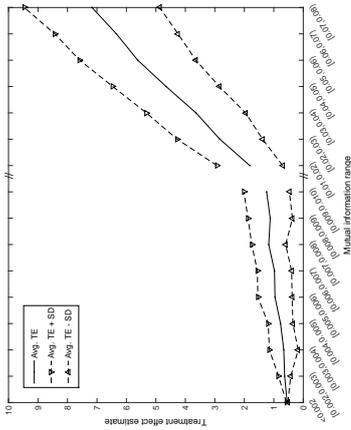


Figure 5: Trends for estimated treatment effects with different mutual information ranges.

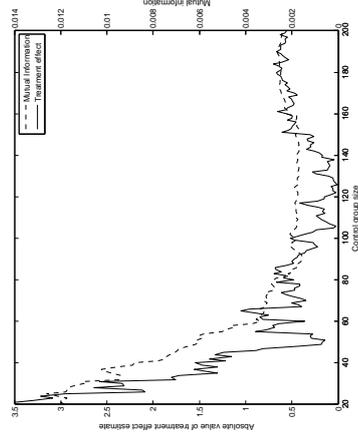


Figure 6: Trends for estimated treatment effects with different control group sizes.

the control groups of different sizes can be compared on the same scale, and hence, the optimization of the control group size becomes possible.

Figure 6 shows the MI values and MIM-based ATT estimates as functions of the control group size in the range from 20 to 200 (of control units). Despite the noise, it is clear that for some control group sizes, the MIM method achieves lower MI values, and simultaneously, higher quality estimates. Most importantly, the group size range, over which the MI is consistently low, coincides with the range, for which the estimates are closest to the truth. As such, in the considered example, the MI gets closer to zero with the lowest noise for the control group of sizes of about 130; a control group of this size returns the ATT estimate value of 0.099.

5.3 Experiences with Large Data Sets

In order to test the performance of the presented MIM methods with large data sets, three suitable real-world data sets were identified. The first one contains weighted census data extracted from the 1994 and 1995 Current Population Surveys conducted by the U.S. Census Bureau (Lichman, 2013): it contains 199,523 records with 41 demographic and employment related variables. The second one was extracted from the 1994 Census database (Lichman, 2013): it contains 32,561 records with 14 variables. The third one was collected in a study focusing on the National Supported Work Demonstration Program (NSW) (LaLonde, 1986), where the randomized job training experiment benchmark was obtained for the treatment effect: it contains 16,177 records with 8 variables. Of the three data sets, only the third one was originally created for a matching purpose. The aim of its creator was to examine how well the statistical methods would perform in trying to replicate the result of a randomized experiment (LaLonde, 1986). To design the test instances with the different number of covariates and varied control pool and treatment group sizes, the first data set was split into a treatment group and a control pool by “US citizenship” and “Business ownership” indicators, respectively; and the second data set was split by “US native” and “Doctorate degree” indicators, respectively. To apply the MIM-Marginal method, each continuous covariate’s range was partitioned into 20 bins, while all the categorical covariates kept their original categories. The target control group size was set equal to the treatment group size. Table 3 gives an aggregate view of the data sets’ specifics, and MIM results and runtimes.

Overall, the MIM-marginal method achieves very good balance across all the covariates. The average KS scores in all the tests are below 0.01 and the average p -values are all greater than 0.05. Note that since the treated and control data sets in every test are distinct, the results cannot be meaningfully compared across the test instances. For example, the best balance metric values were achieved in the experiment the “Business ownership” data set, even though it had more units and covariates than some other data sets. In the experiment with the “US citizenship” data set, the matching algorithm performed the worst. Indeed, this was a challenging test instance with the target control group size of 13,401, amounting to about 7.2% of the control pool: in such a case, the method is forced to pick non-optimal units to reach the target size, and hence, increases the imbalance. Note, however, that selecting such a large size control group might not be a good idea in practice anyway.

Excellent computational efficiency of the MIM-Marginal method is unparalleled by any other matching method, making it highly practical for data mining: its runtime requirement grows polynomially with the problem size. For example, the “US citizenship” test instance features a very large data set: compared to the training data set, it has 11.6 times more control units, 72.4 times more treated units, significantly larger target control group size, and 5 times more covariates. Yet, the MIM-Marginal runtime with the “US citizenship” is only 38,040 times larger.

The NSW data set is the most famous one in the matching literature, because an ATT benchmark of 1,794 has been separately obtained for the problem that it addresses (LaLonde, 1986). Deleija and Walha (1999) reported the estimate based on propensity score method was 1,691. Tam Cho et al. (2013) used BOSS and obtained the best individual matched solution resulting in the estimate of 1,741, as well as a set of alternative solutions, with mean 1,595 and standard deviation 281.

Table 3: Performance of the MIM-Marginal method on practical data sets.

	Census 94-95 (US Citizenship)	Census 94-95 (Business Ownership)	Extracted 94 (US Native)	Extracted 94 (Doctorate)	NSW 86 (Training)
Data	# Control Units	186,122	196,825	29,170	32,148
	# Treated Units	13,401	2,698	3,391	413
	# Continuous Covariates	8	8	6	6
	# Categorical Covariates	32	32	7	7
	# Marginal Bins	645	647	180	206
Balance	Avg KS	0.064	0.001	0.020	0.095
	Max KS	0.412	0.002	0.133	0.200
	Avg p-value	0.524	1.000	0.769	0.746
	Min p-value	0.109	1.000	0.518	0.459
	Avg MI	0.020	0.001	0.004	0.085
Time	Binning (seconds)	4,410.84	4,319.71	46.23	47.62
	Matching (seconds)	66,950.58	14,235.49	399.70	49.83

The runs of MIM-Marginal with the NSW data set produce a solution set with mean 1,851.5 and standard deviation 92.1. The average MI over this set is 0.002383; see the achieved balance metrics in Table 3. Importantly, if one removes the target control group size restriction and allows the MIM-Marginal to optimize over it, then a solution with 169 control units is obtained, with the MI of 0.001824 and ATT estimate set with mean 1,818.2 and standard deviation 91.2.

6. MIM Limitations and Future Research Directions

While the presented computational investigations demonstrate the utility of the MIM methodology, this work has its limitations and desirable directions for further improvement.

First, this paper does not offer an approach to the calculated selection of a binning scheme. The discretization of the covariate space affects the MI-based estimation outputs, however, the present MIM algorithms take the binning scheme as an input and do not work to perturb it to account for the differences in the shapes of the distributions of different covariates or the distances between bins. Intuitively, if the binning is coarse then the MIM cannot be expected to produce high quality solutions. While binning has been a point of research in multiple branches of optimization-based matching literature, the design of binning structures is still an open question. Another point, relevant to the MI based methods specifically, is that mutual information could be employed in its continuous form, in which case the accuracy of matching might be improved without the use of bins.

Second, in its current form as a non-parametric matching methodology, MIM does not differentiate the covariates by relative importance. Moreover, if there is any indispensable information of the form of the response function, covariate relationships, or covariate distributions that is not captured via binning, MIM may underperform. There may even exist circumstances where MIM would be consistently unsuccessful in producing accurate treatment effect estimates: such circumstances, as those exposed by Sauppe et al. (2014) with propensity score-based matching, are yet to be explored with MIM. In any case, prior to using MIM, the researcher must be careful about selecting the covariates to work with. Indeed, data preprocessing has been a topic of research worth much attention. Distance-based matching methods employ weights to emphasize the importance of balancing certain some covariates over others. The developments in propensity score-based methods led to the introduction of the concept of “fine balance”. Expanding the MIM research in a similar direction would add to its value.

Finally, by relaxing the integrality constraints of the MIM problems’ decision variables, one could produce linear (non necessarily integer) solutions allowing for insightful interpretations. The research in this direction might remedy the MIM dependence on binning.

7. Conclusion

The problem of causal inference based on observational data lies in selecting control units from a large unit pool to achieve control groups that are similar in covariate distributions to a given treatment group. To address this problem, this paper presents a set of methods with the objective of minimizing the mutual information between the treatment and covariates over the merged set of selected control and treatment units. Optimal conditions are derived

for matching on a single covariate and on the joint distribution of multiple covariates, allowing one to remove non-linear terms from the original mutual information formula and leading to a mixed integer programming formulation of the problem. A sequential selection algorithm is presented that runs in polynomial time and obtains optimal solutions for the problems of matching on a single covariate and matching on a joint distribution of multiple covariates.

Matching problems formulated in this paper for both joint distribution and marginal covariate distributions are analyzed theoretically, and the resulting solution methods tested computationally. The problem of group matching with marginal covariate distributions is proven to be NP-complete, and a fast sub-optimal algorithm is presented. The reported computational study shows that the matching problem formulation with marginal covariate distributions is more valuable than that based on the joint covariate distribution for obtaining accurate causal effect estimates in practice.

Appendix A. Proof of Theorem 1

The decision version of the matching problem on marginal covariate distributions with a fixed target control group size (MIM-Marginal) can be stated as follows. Given a treatment group \mathcal{T} , a control pool \mathcal{C} and a set of covariates X_k , $k \in K$. Let m be a marginal bin. (In this proof, it is not necessary to indicate which covariate this marginal bin partitions.) Given parameters γ and N , do there exist subsets $\mathcal{S} \subset \mathcal{C}$ such that $\sum_{k \in K} I(\mathcal{T}; X_k) \leq \gamma$ and $|\mathcal{S}| = N$?

First, it has to be proven that MIM-Marginal belongs to the NP class. For any given subset, one can check that the subset contains exactly N units, and then, calculate the mutual information value as in 2 to check if it is smaller or equal to γ . This can be completed in polynomial time, thus MIM-Marginal belongs to NP.

Second, it has to be proven that MIM-Marginal is NP-hard. Let δ_{um} be a binary variable, with $\delta_{um} = 1$ if unit u belongs to m , and 0 otherwise; let η_u be another binary variable, with $\eta_u = 1$ if unit u is selected into \mathcal{S} , and 0 otherwise. Let T_m denote the number of units in group \mathcal{T} with the values of covariates falling into bin m . If let $\gamma = 0$ and $N = |\mathcal{T}|$, then problem's objective is to check whether a perfect matching exists, i.e., whether the following constraints can be simultaneously satisfied:

$$\sum_{u=1}^{|\mathcal{C}|} \delta_{um} \eta_u = T_m \quad \forall m, \quad (17)$$

$$\sum_{u=1}^{|\mathcal{C}|} \eta_u = N, \quad (18)$$

$$\eta_u \in \{0, 1\} \quad \forall u,$$

where η_u is the decision variable. Constraint (17) ensures that a perfect matching is achieved in each covariate. Constraint (18) limits the size of a control group that can be selected. Note that, since $N = |\mathcal{T}|$ and each unit belongs to exactly $|K|$ marginal bins, then converting the constraints (17) and (18) into inequalities does not affect the optimal set of the problem, which can now be stated as

$$\sum_{u=1}^{|\mathcal{C}|} \delta_{um} \eta_u \geq T_m \quad \forall m,$$

$$\sum_{u=1}^{|\mathcal{C}|} \eta_u \leq N,$$

$$\eta_u \in \{0, 1\} \quad \forall u.$$

Now, the set cover (SC) problem can be reduced to MIM-Marginal problem. The SC problem is known to be NP-Hard (Garey and Johnson, 1979), and can be stated as follows. Given: an element set J , a collection I of finite subsets of J , and a fixed number n . Question: does I contain a subcollection of sets such that the total number of sets in this subcollection is at most n , and each element of J is included in at least one of the selected sets?

Let δ_{ij}^I be a binary variable, with $\delta_{ij}^I = 1$ if element j is included in set $I_i \in I$, and 0 otherwise; let η_i^I be another binary variable, with $\eta_i^I = 1$ if set I_i is selected, and 0 otherwise. The objective of the SC problem is to find η^I such that

$$\begin{aligned} \prod_{i=1}^{|I|} \delta_{ij}^I \eta_i^I &\geq 1 \quad \forall j \in J, \\ \sum_{i=1}^{|I|} \eta_i^I &\leq n, \end{aligned}$$

$$\eta_i^I \in \{0, 1\} \quad \forall i.$$

Define the following mapping: $T_m = 1$, $N = n$, $u = i$, $m = j$, $\delta_{um} = \delta_{ij}^I$ and $\eta_u = \eta_j^I$. Thus, the SC problem has a feasible solution if and only if the corresponding MIM-Marginal has a solution. The transformation required to execute the described mapping can be completed in polynomial time in the size of problem inputs. This completes the proof.

Appendix B. Properties of the Ratio Function $f(x, y)$ in Lemma 3

This Appendix analyzes how the values of function $f(x, y) = \frac{(1+\frac{1}{x})^x}{(1+\frac{1}{x+y})^x(1+\frac{1}{x+1})^x}$ can be compared for arbitrary inputs (x, y) , with $x > 0$ and $y > 0$.

Define function $g(x, y) \equiv \log f(x, y) = x \log(1 + \frac{1}{x}) - (x+y) \log(1 + \frac{1}{x+y}) - \log(1 + \frac{x}{1+x}) = x(\log(1+x) - \log(x)) - (x+y)(\log(1+x+y) - \log(x+y)) - (\log(1+x+y) - \log(1+x)) = (1+x) \log(1+x) - x \log(x) - (1+x+y) \log(1+x+y) + (x+y) \log(x+y)$.

Then, $\frac{\partial g}{\partial x} = \log(1+x) - \log(x) + \log(x+y) - \log(1+x+y)$ and $\frac{\partial g}{\partial y} = \log(x+y) - \log(1+x+y)$. Also, $\frac{\partial g}{\partial x} = \frac{1}{x} \frac{\partial f}{f}$ and $\frac{\partial g}{\partial y} = \frac{1}{y} \frac{\partial f}{f}$.

Because $x > 0$ and $y > 0$, one has $1+x+y > 1+x > 0$. Also, the logarithm is a monotonically increasing function with the monotonically decreasing slope, and hence, one has $\log(1+x) - \log(x) > \log(1+x+y) - \log(x+y) > 0$, which implies that $\frac{\partial g}{\partial x} > 0$ and $\frac{\partial g}{\partial y} < 0$. Moreover, since $f > 0$, then $\frac{\partial f}{\partial x} > 0$ and $\frac{\partial f}{\partial y} < 0$.

To sum up, the function of interest is monotonic along both x and y directions. This prompts one to study its contour lines over the feasible range of inputs (x, y) (Figure 7). Unfortunately, even though the contours look linear, it does not seem possible to produce a closed-form expression for them, of the form $f(x, y) = C$, with constant C .

Since $f(x, y) = C$ is approximately a straight line for every specific C , then the values of $f(x, y)$ under different inputs can be compared by evaluating the slopes of the corresponding contour lines: $\frac{\partial f}{\partial x} = -\frac{\frac{\partial f}{\partial y}}{\frac{\partial f}{\partial x}} = -\frac{\log(1+x) - \log(x)}{\log(1+x+y) - \log(x+y)} - 1$. Let $h(x, y) \equiv \frac{\partial f}{\partial x}$. For arbitrary

(x_0, y_0) and (x_1, y_1) such that $f(x_0, y_0) = f(x_1, y_1)$, one has $h(x_0, y_0) = h(x_1, y_1) = \frac{y_1 - y_0}{x_1 - x_0}$; thus, one can study the linearization $h(x, y)$ of $f(x, y)$. Since these contour lines are straight and do not intersect in the first quadrant, they must have a unique, common intersection point. Thus, one can write $h(x, y) = \frac{y - A_1}{x - A_2}$, where (A_1, A_2) are the coordinates of that unknown intersection point. By numerical approximation, one can derive that $(A_1, A_2) \approx (-0.47, 0)$.

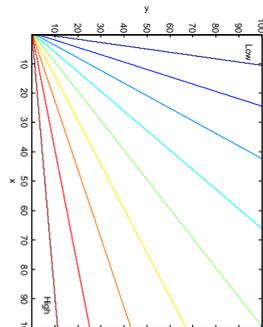


Figure 7: A sketch of the contour lines of $f(x, y)$ on (x, y) plane.

Due to the monotonicity of $f(x, y)$, the greater the slope of a contour line that the point (x, y) lies on, the smaller its corresponding function value. In other words, the smaller $h^{-1}(x, y) = \frac{y - A_1}{x - A_2}$, the smaller $f(x, y)$, and vice versa.

Appendix C. A Complete List of Notations Used

- T : Treatment group.
- C : Control pool.
- S : Control group.
- N : A given integer as the target control group size, i.e. $|S| = N$ for an eligible matched control group.
- u : An observable unit, $u \in \mathcal{T} \cup C$.
- t : Treatment indicator (1 means treated; 0 means not treated).
- Y_u^1 (or Y_u^0): Treated (or untreated) response of unit u .
- b : Joint bin. b_1, b_2 and b_3 are different bins used in proofs.
- B : Set of joint bins. B^1, B^2 and B^3 are different bin sets used in proofs. B^F is particular bin set in the MIP-based algorithm.
- m : Marginal bin.
- M^F : Set of marginal bins. M^F is particular bin set in the MIP-based algorithm.
- k : A covariate.
- K : Set of covariates.
- X_k : Value of covariate k .
- \mathbf{X} : Covariate vector $\{X_1, X_2, \dots, X_{|K|}\}$.
- X : A generalized covariate value to represent \mathbf{X} and X_k for writing convenience.
- X_k^u : Covariate value of unit u .
- X_k^m : Covariate value for any unit contained in bin b .
- X_m : Covariate value for any unit contained in marginal bin m .
- $p(t)$: Probability that a unit is treated.
- $p(X_k|t)$: Probability that a unit's covariate value falls into bin b .
- $p(X_k, t)$: Probability that the covariate value of a unit with treatment indicator t falls into

bin b .
 S_b (or T_b, C_b): Number of units in group \mathcal{S} (or \mathcal{T}, \mathcal{C}) with covariate values falling into bin b . $S_{b_1}, S_{b_2}, S_{b_3}, T_{b_1}, T_{b_2}, T_{b_3}$ and C_{b_2} are the number of units in different groups used in proofs.
 S_m (or T_m, C_m): Number of units in group \mathcal{S} (or \mathcal{T}, \mathcal{C}) with covariate values falling into marginal bin m .
 $I(t; X)$ (or $I(t; \mathbf{X}), I(t; X_k)$): Mutual information between treatment indicator and covariate value X (or covariate vector \mathbf{X} , covariate value X_k).
 I_b : Mutual information treatment indicator and covariate value if a unit in bin b is added to the control group, e.g. I_1, I_2 and I_3 .
 A and α : Constant numbers.
 q : Objective value of MIP models.

References

- Alberto Abadie and Guido W. Imbens. Large sample properties of matching estimators for average treatment effects. *Econometrica*, 74(1):235–267, 2006.
- Dimitris Bertsimas and Romy Shioda. Classification and regression via integer optimization. *Operations Research*, 55(2):252–271, March 2007. ISSN 0030-364X.
- Kenneth P. Burnham and David R. Anderson. *Model selection and multi-model inference: a practical information-theoretic approach*. Springer Verlag, 2002.
- W. G. Cochran. The planning of observational studies of human populations. *Journal of the Royal Statistical Society. Series A (General)*, 128(2):234–266, 1965.
- Paula da Veiga and Ronald Wilder. Maternal smoking during pregnancy and birthweight: A propensity score matching approach. *Maternal and Child Health Journal*, 12:194–203, 2008.
- Rajeev Dehejia. Practical propensity score matching: a reply to smith and todd. *Journal of Econometrics*, 125(1):355–364, 2005.
- Rajeev H. Dehejia and Sadek Wahba. Causal effects in nonexperimental studies: Reevaluating the evaluation of training programs. *Journal of the American Statistical Association*, 94(448):pp. 1053–1062, 1999.
- Rajeev H. Dehejia and Sadek Wahba. Propensity score-matching methods for nonexperimental causal studies. *Review of Economics and Statistics*, 84(1):151–161, 2002.
- Alexis Diamond and Jasjeet S. Sekhon. Genetic matching for estimating causal effects: A general multivariate matching method for achieving balance in observational studies. *Review of Economics and Statistics*, 95(3):932–945, 2013.
- Pablo A. Estévez, Michel Tesmer, Claudio A. Perez, and Jacek M. Zurada. Normalized mutual information feature selection. *IEEE Transactions on Neural Networks*, 20(2):189–201, 2009.
- Andrew M. Fraser and Harry L. Swinney. Independent coordinates for strange attractors from mutual information. *Physical Review A*, 33:1134–1140, 1986.
- Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.
- Jens Hainmueller. Entropy balancing for causal effects: A multivariate reweighting method to produce balanced samples in observational studies. *Political Analysis*, 20(1):25–46, 2012.
- Ben B. Hansen. Full matching in an observational study of coaching for the sat. *Journal of the American Statistical Association*, 99(467):609–618, 2004.
- Ben B. Hansen and Stephanie O. Klopfer. Optimal full matching and related designs via network flows. *Journal of Computational and Graphical Statistics*, 2012.
- Daniel Ho, Kosuke Imai, Gary King, and Elizabeth Stuart. Matchit: Nonparametric preprocessing for parametric causal inference. *Journal of Statistical Software*, 42(1):1–28, 2011.
- Daniel E. Ho, Kosuke Imai, Gary King, and Elizabeth A. Stuart. Matching as nonparametric preprocessing for reducing model dependence in parametric causal inference. *Political Analysis*, 15(3):199–236, 2007.
- Paul W. Holland. Statistics and causal inference. *Journal of the American Statistical Association*, 81(396):945–960, 1986.
- Stefano M. Iacus, Gary King, and Giuseppe Porro. Causal inference without balance checking: Coarsened exact matching. *Political Analysis*, 20(1):1–24, 2012.
- Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E*, 69:066138, 2004.
- Robert J. LaLonde. Evaluating the econometric evaluations of training programs with experimental data. *The American Economic Review*, 76(4):604–620, 1986.
- M. Lichman. UCI machine learning repository, 2013. URL <http://archive.ics.uci.edu/ml>.
- Stephen L. Morgan and David J. Harding. Matching estimators of causal effects. *Sociological Methods & Research*, 35(1):3–60, 2006.
- Alexander G. Nikolaev, Sheldon H. Jacobson, Wendy K. Tam Cho, Jason J. Stauppe, and Edward C. Sewell. Balance optimization subset selection (BOSS): An alternative approach for causal inference with observational data. *Operations Research*, 61(2):398–412, 2013.
- Stefano Panzeri and Alessandro Treves. Analytical estimates of limited sampling biases in different information measures. *Network Computation in Neural Systems*, 7(1):87–107, 1996.

- Paul R. Rosenbaum and Donald B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.
- Paul R. Rosenbaum and Donald B. Rubin. Constructing a control group using multivariate matched sampling methods that incorporate the propensity score. *The American Statistician*, 39(1):33–38, 1985.
- Paul R. Rosenbaum, Richard N. Ross, and Jeffrey H. Silber. Minimum distance matched sampling with fine balance in an observational study of treatment for ovarian cancer. *Journal of the American Statistical Association*, 102(477):75–83, 2007.
- Mark S. Roulston. Estimating the errors on measured entropy and mutual information. *Physica D: Nonlinear Phenomena*, 125:285–294, 1999.
- Donald B. Rubin. Matching to remove bias in observational studies. *Biometrics*, 29:159–183, 1973.
- Donald B. Rubin. Bias reduction using mahalanobis-metric matching. *Biometrics*, 36:293–298, 1980.
- Donald B. Rubin. Using propensity scores to help design observational studies: Application to the tobacco litigation. *Health Services and Outcomes Research Methodology*, 2:169–188, 2001.
- Donald B. Rubin. *Matched Sampling for Causal Effects*. Cambridge University Press, New York, 2006.
- Jason J. Saupe, Sheldon H. Jacobson, and Edward C. Sewell. Complexity and approximation results for the balance optimization subset selection model for causal inference in observational studies. *INFORMS Journal on Computing*, 26(3):547–566, 2014.
- Jasjeet S. Sekhon. Multivariate and propensity score matching software with automated balance optimization: The matching package for R. *Journal of Statistical Software*, 42(107), 2008.
- Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.
- Jeffrey Smith and Petra Todd. Rejoinder. *Journal of Econometrics*, 125(1):365–375, 2005a.
- Jeffrey Smith and Petra Todd. Does matching overcome lalonde’s critique of nonexperimental estimators? *Journal of Econometrics*, 125(1):305–353, 2005b.
- Wendy K. Tann Cho, Jason J. Saupe, Alexander G. Nikolaev, Sheldon H. Jacobson, and Edward C. Sewell. An optimization approach for making causal inferences. *Statistica Neerlandica*, 67(2):211–226, 2013.
- Jos R. Zubizarreta. Using mixed integer programming for matching in an observational study of kidney failure after surgery. *Journal of the American Statistical Association*, 107(500):1360–1371, 2012.

Online Trans-dimensional von Mises-Fisher Mixture Models for User Profiles

Xiangju Qin

Pádraig Cunningham

School of Computer Science

University College Dublin

Belfield, Dublin 4, Ireland

XIANGJU.QIN@UCDCONNECT.IE

PADRAIG.CUNNINGHAM@UCD.IE

Michael Salter-Townshend

Department of Statistics

University of Oxford

24-29 St Giles, Oxford, OX1 3LB, UK

SALTER@STATS.OX.AC.UK

Editor: David Dunson

Abstract

The proliferation of online communities has attracted much attention to modelling user behaviour in terms of social interaction, language adoption and contribution activity. Nevertheless, when applied to large-scale and cross-platform behavioural data, existing approaches generally suffer from expressiveness, scalability and generality issues. This paper proposes trans-dimensional von Mises-Fisher (TvMF) mixture models for L_2 normalised behavioural data, which encapsulate: (1) a Bayesian framework for vMF mixtures that enables prior knowledge and information sharing among clusters, (2) an extended version of reversible jump MCMC algorithm that allows adaptive changes in the number of clusters for vMF mixtures when the model parameters are updated, and (3) an online TvMF mixture model that accommodates the dynamics of clusters for time-varying user behavioural data. We develop efficient collapsed Gibbs sampling techniques for posterior inference, which facilitates parallelism for parameter updates. Empirical results on simulated and real-world data show that the proposed TvMF mixture models can discover more interpretable and intuitive clusters than other widely-used models, such as k-means, non-negative matrix factorization (NMF), Dirichlet process Gaussian mixture models (DP-GMM), and dynamic topic models (DTM). We further evaluate the performance of proposed models in real-world applications, such as the churn prediction task, that shows the usefulness of the features generated.

Keywords: Mixture Models, von Mises-Fisher, Bayesian Nonparametric, Temporal Evolution, User Modelling

1. Introduction

Recent years have witnessed an increasing population of online peer production communities, such as *Wikipedia*, *Stack Overflow* and *OpenStreetMap*, which rely on contributions from volunteers to build knowledge, software artifacts and navigational tools, respectively. The growing popularity and importance of these communities requires a better understanding and characterisation of user behaviour so that the communities can be better managed, new services delivered, challenges and opportunities detected. For instance, by understanding the general lifecycles that users go through and the key features that distinguish different user groups and different life stages, we can develop

techniques for the following applications (Qin et al., 2014): (i) predict whether a user is likely to abandon the community; (ii) develop intelligent software to recommend tasks for users within the same life-stage. Moreover, social interaction and contribution behaviour of contributors plays a significant role in shaping the health and sustainability of online communities.

Different from text documents, which are commonly represented as term-frequency vectors, user behavioural data derived from online communities are generally represented as unit vectors. For instance, the level of linguistic change for online users in beer rating websites is denoted as a numeric feature (Danescu-Niculescu-Mizil et al., 2013). The measures used to quantify the centrality of members' positions in social networks are naturally numeric measurements (Rowe, 2013; Chan et al., 2010). The quality of questions, answers, and comments posted by users on Q&A sites are also numeric measures (Furtado et al., 2013). Existing approaches to identify patterns of user behaviour include principle component analysis, clustering analysis and entropy-based methods. However, these studies tend to be application-specific and suffer from scalability and generality issues due to the constrained feature set and the inherent limitations of the approaches employed. Additionally, the existing approaches fail to capture a mixture of user interests over time.

On the other hand, over the last decade, there have been significant advances in topic models which develop automatic text analysis techniques to discover latent structures from time-varying document collections (e.g. Blei and Lafferty (2006); Ahmed and Xing (2010); Gopal and Yang (2014)) and from time-varying user activity data in computational advertising (Ahmed et al., 2011). Topic models generally work well in document collections and user activity data where the data are represented in term-frequency format. However, many traditional topic models are not applicable to scenarios where the data are represented as unit vectors, e.g. the term frequency-inverse document frequency (tf-idf) representation of documents. Based on von Mises-Fisher (vMF) distributions, Gopal and Yang (2014) proposed dynamic clustering models which combine the success of normalised representation and flexibility of graphical models, and found that their proposed models can discover more intuitive clusters than existing approaches. Nevertheless, the vMF clustering models by Gopal and Yang (2014) did not take into consideration the dynamic evolution in the number of clusters and the birth/death of clusters over time. This work makes the following contributions:

- Extend the reversible jump Markov Chain Monte Carlo (RJCMC) algorithm (Richardson and Green, 1997) for directional distribution (i.e. vMF mixtures).
- Enhance the Bayesian and temporal vMF mixture models by Gopal and Yang (2014) by integrating with our extended version of RJCMC algorithm, which empowers both models with the ability to change the number of clusters automatically and to refine an inappropriate initialization of model parameters.
- Apply the model to analyse time-varying user behaviour data in online communities, in particular Wikipedia. Compared with previous works in this direction (Danescu-Niculescu-Mizil et al., 2013; Rowe, 2013; Furtado et al., 2013; Chan et al., 2010), the proposed model is more general and can be applied to model user behaviour in online communities (e.g. *Wikipedia*, *Stack Overflow*, *Twitter*, and *Facebook*) whenever user behavioural data are available.

We develop efficient collapsed Gibbs sampling techniques for the proposed models, which allows parallelism for parameter updates. The empirical comparison on synthetic and real-world data demonstrates that the proposed model can generate a more intuitive and interpretable clustering than other popular tools, such as k-means (Hartigan and Wong, 1979), non-negative matrix factorization

(NMF) (Lee and Seung, 1999), Dirichlet process Gaussian mixture models (DP-GMM) (Chang and Fisher III, 2013), and dynamic topic models (DTM) (Blei and Lafferty, 2006).

The rest of the paper is organised as follows. Section 2 provides an overview of related work, followed by an introduction to the von Mises-Fisher (VMF) distribution in Section 3. In Section 4, we present posterior inference for Bayesian von Mises-Fisher mixture models using collapsed Gibbs sampling techniques, model exploration using our extension of the reversible jump MCMC algorithm, and an online trans-dimensional von Mises-Fisher mixture model for time-varying user behavioural data. We demonstrate the empirical performance of the proposed models using synthetic and real-world data in Section 5. We then present an application of the features generated by the models for user clustering and churn prediction tasks, followed by discussion and concluding remarks in Section 6. The appendix includes detailed derivations for model inference and additional detailed analysis of the models.

2. Related Work

In this section, we provide an overview of the main lines of research underpinning this work, and discuss how our work leverages and advances the state-of-the-art techniques.

2.1 Modelling User Behaviour

Recently, researchers have approached the issue of modelling online user behaviour from different perspectives. They have so far focused on a separate set or combination of user properties, such as information exchange behaviour in discussion forums (Chan et al., 2010), social and/or lexical dynamics in online platforms (Danesescu-Niculescu-Mizil et al., 2013; Rowe, 2013), and diversity of participation behaviour on Q&A sites (Furrado et al., 2013). These studies generally employed either principle component analysis and clustering analysis to identify user profiles (Chan et al., 2010; Furrado et al., 2013) or entropy measures to track social and/or linguistic changes throughout user lifecycles (Danesescu-Niculescu-Mizil et al., 2013; Rowe, 2013). While previous studies provide insights into community composition, user profiles and their dynamics, they have limitations either in their definition of lifecycle periods (e.g. dividing each user’s lifetime using a fixed time-slicing approach (Danesescu-Niculescu-Mizil et al., 2013) or a fixed activity-slicing approach (Rowe, 2013)) or in the expressiveness of user lifecycles in terms of the evolution of expertise and user activity for users and online communities over time. Specifically, previous studies failed to capture a mixture of user interests over time. Different from previous works, Qin et al. (2014) employed topic modelling to study the evolving patterns of editor behaviour in Wikipedia. They found that a number of editor roles (e.g. Technical Experts, Social Networkers) prevail in the temporal Wikipedia editor activity data, and that the features inspired by latent space representation are beneficial for the churn prediction task. However, two major limitations exist in the topic model used by Qin et al. (2014): (1) the inability to deal with numeric behavioural data, and (2) the inability to capture the birth/death of topics over time.

2.2 Parametric and Nonparametric Temporal Models

Parametric models, such as Latent Dirichlet Allocation (LDA) by Blei et al. (2003) and non-negative matrix factorization (NMF) by Lee and Seung (1999), generally assume a fixed pre-specified number of topics a priori. This assumption inevitably involves computationally expensive model com-

parison using model selection criteria such as penalised log-likelihoods (AIC and BIC) in order to choose an appropriate number of topics for a given dataset. While log-likelihood related criteria have shown success in static settings, it is obvious that doing model selection for each time period may lead to a sub-optimal solution as it ignores the role of context in model selection, or alternatively, local optima problems in model selection at each epoch may accumulate and result in a globally suboptimal solution (Wang et al., 2007; Ahmed and Xing, 2008). Nonparametric models have been suggested in order to relax the assumption of a fixed number of topics for stationary and time-varying data.

There are two major lines of research for non-parametric models: (1) hierarchical Dirichlet process (HDP, Teh et al. (2006)), and (2) trans-dimensional (split/merge) approaches. The hierarchical Dirichlet process is a Bayesian nonparametric model that can be used to cluster groups of data with a potentially infinite number of components. In HDP based nonparametric models, each observation within a group is a draw from a Dirichlet process (DP) (or mixture model), the group-specific DPs can be linked together via another DP to ensure the sharing of mixture components between groups; the well-known clustering property of the DP can provide a nonparametric prior for the number of mixture components within each group (Teh et al., 2006). HDP has been widely used to learn recurring patterns (or “topics”) from document collections (e.g. the nonparametric Topics over Time (npTOT) model by Dubey et al. (2013)) and time-varying user activity data in computational advertising (the Time-Varying User Model (TVUM) by Ahmed et al. (2011)). In contrast, trans-dimension nonparametric models adapt the number of components by using a split-merge or birth-death mechanism while preserving certain properties (e.g. the zeroth, first and second moments) of the components. One well-known example of trans-dimensional models in this direction is the reversible jump MCMC algorithm by Richardson and Green (1997). Recently, Chang and Fisher III (2013) proposed a novel parallel restricted Gibbs sampling algorithm for Dirichlet process Gaussian mixture models (DP-GMM) with sub-cluster split/merge moves, and showed that their proposed sampler is orders of magnitude faster than other exact MCMC methods. In addition, based on the small-variance limit of Bayesian nonparametric von-Mises Fisher (VMF) mixture distributions (i.e. the birth-death strategy), Straub et al. (2015a) proposed two novel flexible and efficient k -means-like clustering algorithms for directional data such as surface normals in computer vision applications.

The evolving nature of data in different scenarios, such as scientific publications, news stories, and user query logs for search engines, has motivated research about temporal models to cope with the challenges of learning coherent and interpretable clusters over time (Ahmed and Xing, 2008). A good evolutionary model should be able to accommodate the dynamics of different aspects of the evolving clusters (Ahmed and Xing, 2008; Ahmed and Xing, 2010), specifically:

- Dynamics of cluster parameters. For example, in Gaussian mixture models, the mean and covariance for a mixture of Gaussians should be able to evolve according to a given time series model, such as Kalman filter that is used in the dynamic topic model (DTM) by Blei and Lafferty (2006). One principle for choosing a time series model for cluster parameters is to guarantee the smoothness of cluster parameters over time. One common strategy for this is to draw the cluster parameters at time epoch t from the corresponding distribution at the previous time $t - 1$ by leveraging the smoothness assumption over cluster parameters (Blei and Lafferty, 2006; Ahmed and Xing, 2008; Ahmed and Xing, 2010; Ahmed et al., 2011; Gopal and Yang, 2014).

- Popularity of clusters over time. The popularity of clusters can change over time due to the evolving nature of data. Existing models have relied on the rich gets richer assumption to capture the trends of clusters over time (Ahmed and Xing, 2008; Ahmed and Xing, 2010; Ahmed et al., 2011).
- Automatic change in the number of clusters over time. Non-parametric models allow the clusters to remain, die out or emerge over time (Ahmed and Xing, 2008; Ahmed and Xing, 2010; Ahmed et al., 2011; Dubey et al., 2013).

To the authors' knowledge, while many aforementioned models have focused on categorical data, only the models by Gopal and Yang (2014) are designed for \mathcal{L}_2 normalised data. However, their models are parametric which require using model selection criteria to choose the appropriate number of clusters. In this work, based on the smoothness assumption and the idea of reversible jump MCMC algorithm, we extend their Bayesian vMF mixture model and propose an online trans-dimensional von Mises-Fisher mixture model (OTvMFMM) for time-varying user behavioural data. The proposed model not only allows us to explore the model space for clusters, but more importantly, it can model time-varying clusters that are consistent.

2.3 Models for Directional Data

The existence of directional data in many applications has attracted much attention from researchers to build models for clustering on the hypersphere. There are three lines of research related to clustering directional data: (1) Euclidean geometry based algorithms, which ignore the geometric properties of the data and usually use Euclidean distance to measure similarity or distance between data points (e.g. k-means by Hartigan and Wong (1979), the Dirichlet process Gaussian mixture model (DPGMM) by Rasmussen (2000)); (2) spherical geometry based models, which consider the inherent geometry of the data and use cosine similarity to measure similarity between data points with standardized length (e.g. the von Mises-Fisher mixture model (vMFMM) by Banerjee et al. (2005), the Spherical Topic Model (SAM) for documents by Reisinger et al. (2010), the Dirichlet process vMFMM for radiation therapy data by Bangert et al. (2010), the temporal vMF mixture model (Temporal vMFMM) for time-varying document collections by Gopal and Yang (2014)); and (3) models that consider spherical geometry and anisotropic covariance of directional data, which capture the geometric properties and different variances in each dimension of the data (e.g. the Dirichlet process tangential Gaussian mixture model (DP-TGMM) by Straub et al. (2015b)).

Essentially, the vMF distribution can be considered as a variant of the multivariate Gaussian with spherical covariance on \mathbb{S}^{D-1} , parameterized by cosine distance rather than Euclidean distance (Reisinger et al., 2010). Cosine distance belongs to the normalized correlation coefficient and takes into consideration the directions of the \mathcal{L}_2 -normalized feature vectors when computing the similarity. Empirical studies have suggested the advantages of such type of directional measure over Euclidean distance in high-dimensional data particularly in information retrieval (Banerjee et al., 2005; Zhong and Ghosh, 2005; Reisinger et al., 2010; Gopal and Yang, 2014). The vMF distribution can capture the absence/presence of words, which the Multinomial distribution cannot. For instance, let $\theta = [1/3, 1/3, 1/3]$ be a Multinomial parameter (i.e. topic-word) vector, $d = [n_1, n_2, n_3]$ denote the number of occurrences of word w_1, w_2 and w_3 in document d . Assume we have two documents: $d_1 = [1, 1, 1]$ and $d_2 = [3, 0, 0]$. The two documents are more likely to be clustered together under $\text{Multi}(\cdot|\theta)$, whereas d_1 and d_2 have different densities under a corresponding $\text{vMF}(\cdot|\theta)$. The von Mises-Fisher mixture models have been shown to model sparse data (e.g. text) more accurately

than their Multinomial counterparts (Banerjee et al., 2005; Zhong and Ghosh, 2005; Reisinger et al., 2010; Gopal and Yang, 2014).

In this work, we employ the von Mises-Fisher distribution to deal with \mathcal{L}_2 normalised user behavioural data. Table 1 compares the capabilities of the proposed model and previous approaches. Although, it is possible to make the proposed OTvMFMM aiming for anisotropic covariance using Fisher-Bingham distribution as in (Kent, 1982; Peel et al., 2001), extensions to high-dimensional data are difficult due to the normaliser of the probability density function (Straub et al., 2015b).

Table 1: Capabilities of different models

Models (Authors)	Spherical Geometry	Bayesian Inference	Anisotropic Covariance	Nonparametric	Parallelizable	Temporal
DTM (Blei and Lafferty, 2006)	✓	✓	✓	✓	✓	✓
npTOT (Dubey et al., 2013)	✓	✓	✓	✓	✓	✓
TVUM (Ahmed et al., 2011)	✓	✓	✓	✓	✓	✓
vMFMM (Banerjee et al., 2005)	✓	✓	✓	✓	✓	✓
Temporal vMFMM (Gopal and Yang, 2014)	✓	✓	✓	✓	✓	✓
DP-GMM (Chang and Fisher III, 2013)	✓	✓	✓	✓	✓	✓
DP-TGMM (Straub et al., 2015b)	✓	✓	✓	✓	✓	✓
OTvMFMM (proposed)	✓	✓	✓	✓	✓	✓

Notation. In this work, random variables are denoted by capital letters (e.g. X, Y, Z), the observations of random variables (or vectors) are represented by the corresponding lower-case letters (e.g. x, y, z, μ). The set of N observations corresponding to random variable X is denoted by $\mathcal{X} = \{x_i\}_{i=1}^N$. The probability of a set of events A is denoted by $P(A)$, the probability of A given B (i.e. conditional probability) is written as $P(A|B)$. Probability density functions (for continuous random variables) and probability mass functions (for discrete random variables) are denoted by lower-case letters, e.g. $f(x), f(x|\theta)$, p or q , where θ is the set of parameters for a specific distribution. $p(\cdot)$ and $q(\cdot)$ are frequently used to represent the distributions of random variables. The set of observations or prior parameters is denoted by a calligraphic letter (e.g. $\mathcal{X}, \mathcal{Z}, \mathcal{B}$). The set of real numbers is denoted by \mathbb{R} ; the norm $\|\cdot\|$ denotes the \mathcal{L}_2 norm.

3. Preliminaries

This section provides a brief review of the finite von Mises-Fisher mixture models that facilitates a better understanding of the proposed models.

3.1 Von Mises-Fisher Distribution

A random D -dimensional unit vector x (i.e. $x \in \mathbb{R}^D$ and $\|x\|=1$) is said to follow the D -variate von Mises-Fisher (vMF) distribution if its probability density function is given by

$$f(x|\mu, \kappa) = C_D(\kappa) \exp(\kappa \mu^T x); \quad C_D(\kappa) = \frac{\kappa^{D/2-1}}{(2\pi)^{D/2} I_{D/2-1}(\kappa)} \quad (1)$$

where $\|\mu\| = 1, \kappa \geq 0, D \geq 2$; $C_D(\kappa)$ is the normalising constant, $I_{D/2-1}(\kappa)$ denotes the modified Bessel function of the first kind with order $D/2 - 1$ and argument κ . The concentration parameter,

κ , quantifies how tightly the distribution is concentrated around the mean direction μ . Note that $\mu^T x$ is the cosine similarity between x and μ and that κ plays the role of the inverse of variances (precision). The VMF distribution is used for clustering data on the unit hypersphere, whereas the Gaussian distribution is used for modelling data with a multivariate Normal distribution. A very useful property of the VMF distribution that we will use in this work is the preservation of the functional form under multiplication (Chiusso and Picci, 1998)

$$f(x | \mu_1, \kappa_1) f(x | \mu_2, \kappa_2) = f(x | \mu, \kappa), \quad \text{where } \mu = \frac{\kappa_1 \mu_1 + \kappa_2 \mu_2}{\|\kappa_1 \mu_1 + \kappa_2 \mu_2\|}, \quad \kappa = \|\kappa_1 \mu_1 + \kappa_2 \mu_2\| \quad (2)$$

Following [Mardia and Jupp \(2000\)](#), the sample covariance matrix of directional data, $\hat{\Sigma} = \{x_i\}_{i=1}^N$, about μ is defined by:

$$S = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu)(x_i - \mu)^T \quad (3)$$

which is an unbiased estimator of the covariance matrix (i.e. $\Sigma = S$).

3.2 Finite von Mises-Fisher Mixture

Let $f(x|\theta_h)$ denote a VMF distribution with parameters $\theta_h = (\mu_h, \kappa_h)$ for $h \in [1, H]$. [Banerjee et al. \(2005\)](#) proposed a simple VMF mixture model with the density of VMF mixtures given by

$$f(x | \{\pi_h, \theta_h\}_{h=1}^H) = \sum_{h=1}^H \pi_h f(x|\theta_h) \quad (4)$$

where $\Theta = \{\pi_h, \theta_h\}_{h=1}^H$ are the set of parameters to be estimated, π_h are the mixing proportions which are non-negative and sum to 1 (i.e. $0 \leq \pi_h \leq 1, \sum_h \pi_h = 1$). To sample a point from this mixture distribution, we randomly choose the h -th component with probability π_h , and then sample a point x following $f(x|\theta_h)$. Let $\mathcal{X} = \{x_i\}_{i=1}^N$ be a set of N data points that are sampled independently following Eq. (4). The mixture model in Eq. (4) can be interpreted as a missing data model if we introduce a set of membership variables (a.k.a. latent/hidden variables), $\mathcal{Z} = \{z_i\}_{i=1}^N$, for the data points [Celeux et al., 2006](#), indicating the specific VMF distribution from which the points are sampled. Each membership variable is a H -dimensional indicator vector, denoted by $z_i = (z_{i1}, \dots, z_{iH})$, $z_{ih} \in \{0, 1\}$, so that $z_{ih} = 1$ if and only if x_i is generated from the VMF distribution $f(\cdot | \{\pi_h, \theta_h\}_{h=1}^H)$, conditioning on z_i . The z_{ih} are assumed to be drawn independently from the following distributions

$$P(z_{ih} = 1) = \pi_h, \quad i \in [1, N], \quad h \in [1, H] \quad (5)$$

where $P(z_i) = \prod_{h=1}^H \pi_h^{z_{ih}}$. The density of the corresponding VMF mixture model with latent variables is given by [Celeux et al., 2006](#))

$$f(x_i, z_i | \{\pi_h, \theta_h\}_{h=1}^H) = P(z_i) f(x_i | z_i, \{\pi_h, \theta_h\}_{h=1}^H) = \prod_{h=1}^H \{\pi_h f(x_i | \theta_h)\}^{z_{ih}} \quad (6)$$

4. Proposed Trans-dimensional VMF Mixture Models

In this section, we first describe the Bayesian von Mises-Fisher mixture model (BvMFEMM) by [Gopal and Yang \(2014\)](#), including inference via efficient collapsed Gibbs sampling. We then present

our extension of the reversible jump MCMC algorithm for model exploration of Bayesian VMF mixtures, and introduce the proposed online trans-dimensional von Mises-Fisher mixture model for temporal user behavioural data.

4.1 Formulation of Bayesian VMF Mixture Model

The Bayesian VMF mixture model views the VMF mixture model parameters as random variables and introduces prior distributions on them. This brings advantages, such as sharing statistical strength among mean directions and flexibility for parameter estimation [\(Gopal and Yang, 2014\)](#). The Bayesian VMF mixture model is very similar to the Spherical Topic Model [\(Raisinger et al., 2010\)](#), the only difference lies in the fact that the former allows learning cluster-specific concentration parameters while the latter keeps the concentration parameters fixed. The generative process of BvMFEMM proceeds as follows:

1. Draw topic proportions, $\pi \sim \text{Dirichlet}(\{\alpha\})$, from a Dirichlet with hyperparameter α .
2. Draw topics, $\mu_h \sim \text{VMF}(\mu_0, C_0)$, on the unit hypersphere for $h \in [1, H]$.
3. Draw concentration parameters, $\kappa_h \sim \log\text{Normal}(\ln, \sigma^2)$, from a log-normal distribution with mean m and variance σ^2 for $h \in [1, H]$.
4. For each \mathcal{L}_2 normalised data point $x_i \in \{x_i\}_{i=1}^N$:
 - (a) Draw topic indicator $z_i \sim \text{Multic}(\cdot; \pi)$.
 - (b) Draw the data point $x_i \sim \text{VMF}(\mu_{z_i}, \kappa_{z_i})$.

where $\text{Multic}(\cdot; \pi)$ denotes a Multinomial distribution. The plate notation¹ of BvMFEMM is given in [Figure 1](#).

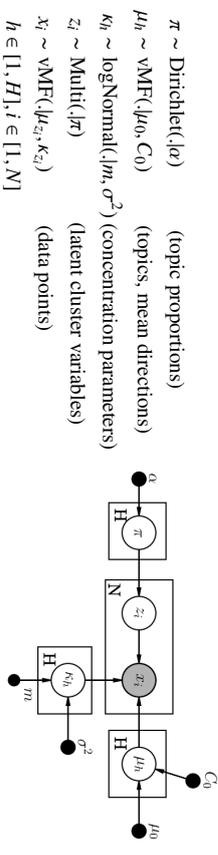


Figure 1: A graphical model representation of BvMFEMM, in which nodes represent random variables, arrows denote dependency among variables, and plates denote replication. Shaded nodes correspond to observed variables, unshaded nodes represent hidden variables, and solid nodes represent the hyperparameters.

For a fully Bayesian VMF mixture model, the number of components H is considered as a random variable for which a posterior distribution should be found. [Nobile \(2005\)](#) suggested that compared with a uniform prior, Poisson(1) prior is strongly biased towards low H and removes

¹ All the plate notations for graphical models used in this work were drawn using the Dact software provided by Dan Foreman-Mackey and David W. Hogg, available at: <http://dact-pgm.org/>

empty clusters. In other words, to some extent, the Poisson(1) prior acts as a penalty term that favors simpler models (i.e., ones with fewer components or factors), which is similar to the idea of AIC or BIC styled model comparison metrics (McLachlan and Peel, 2000; Gershman and Blei, 2012). Therefore, we use Poisson(1) as the prior distribution for H . From the topology of the Bayesian network, the likelihood of the complete-data, i.e., the joint distribution of all known and hidden variables that corresponds to the above generative process and the graphical model of BvMFMM in Figure 1 is given by

$$P(H, X, Z, \{\eta_h, \theta_h\}_{h=1}^H | \mathcal{B}) = P(H) f(\pi(\alpha)) \prod_{h=1}^H f(\mu_h | \mu_0, C_0) f(\kappa_h | m, \sigma^2) \prod_{i=1}^N P(z_i | \pi) f(x_i | \mu_{z_i}, \kappa_{z_i}) \quad (7)$$

where $P(H)$ is the prior probability for H , $f(\pi|\alpha)$ is the prior probability for π , $f(\mu_h | \mu_0, C_0)$ and $f(\kappa_h | m, \sigma^2)$ are the prior probabilities for μ and κ respectively. $P(z_i | \pi)$ are the mixed membership priors for each data point. $f(x_i | \mu_{z_i}, \kappa_{z_i})$ are the pdfs of each observation given its mixed membership and model parameters.

In Eq. (7), the global variables are the topics μ_h , the concentration parameters κ_h and the topic proportions π , while the local variables are the per-data point topic indicators z_i . The user specified prior parameters are $\mathcal{B} = \{\alpha, \mu_0, C_0, m, \sigma^2\}$. We can learn these prior parameters using empirical Bayes and do not rely on the user to set any prior parameters. Alternatively, to avoid potential problems such as overfitting caused by estimating too many parameters, we can specify the values for certain prior parameters with empirical knowledge and update other parameters using empirical Bayes method. The empirical Bayes estimate for prior parameters are given in Appendix B.

Following Heinrich (2009), we can obtain the likelihood of x_i as one of its marginal distributions by integrating out the distributions π , μ and κ and summing over z_i :

$$f(x_i | m, \sigma^2, \mu_0, C_0, \alpha) = \int_{\mu} \int_{\kappa} \text{Mult}(z_i | \pi) f(\pi | \alpha) \times \prod_{j=1}^N f(x_j | \mu_{z_j}, \kappa_{z_j}) f(\mu | \mu_0, C_0) f(\kappa | m, \sigma^2) \quad (8)$$

Finally, the corresponding likelihood of the complete data $\{x_i\}_{i=1}^N$ is given by:

$$f(\{x_i\}_{i=1}^N | m, \sigma^2, \mu_0, C_0, \alpha) = \prod_{i=1}^N f(x_i | m, \sigma^2, \mu_0, C_0, \alpha) \quad (9)$$

4.2 Inference via Collapsed Gibbs Sampling

Because the likelihood of x_i in Eq. (8) is not a closed form distribution, the exact inference of the high dimensional vMF mean parameter μ_h is generally intractable. Following Gopal and Yang (2014), we make use of the fact that the vMF distributions are conjugate and employ this fact to completely integrate out the mean parameters μ and the mixing proportions π . Therefore we need to infer only two sets of parameters $\{Z, \kappa\}$. The conditional distribution for z_i is given by²

$$\begin{aligned} \gamma(z_{ih}) &\equiv P(z_{ih} = i | Z_{-i}, \{x_j\}_{j=1}^N, \kappa, m, \sigma^2, \mu_0, C_0, \alpha) \\ &\propto (\alpha + n_{h,-i}) C_D(\kappa_h) \frac{C_D(\|\kappa_h \sum_{j \neq i} z_{jh} x_j + C_0 \mu_0\|)}{C_D(\|\kappa_h (x_i + \sum_{j \neq i} z_{jh} x_j) + C_0 \mu_0\|)} \end{aligned} \quad (10)$$

² We defer the detailed derivations to Appendix A. The $\gamma(z_{ih})$ can also be seen as the responsibility of component h for explaining the i -th data point, subject to $0 \leq \gamma(z_{ih}) \leq 1$, $\sum_h \gamma(z_{ih}) = 1$.

where Z_{-i} indicates excluding the contribution of the i -th data point. Similarly, the conditional distribution for κ_h is given by²

$$f(\kappa_h | X, Z, \kappa, m, \sigma^2, \mu_0, C_0) \propto \frac{C_D(\kappa_h)^{n_h} C_D(C_0)}{C_D(\|\kappa_h \sum_{j=1}^N z_{jh} x_j + C_0 \mu_0\|)} \log \text{Normal}(\kappa_h | m, \sigma^2) \quad (11)$$

where n_h is the number of observations assigned to the h -th component, $n_h = \sum_i z_{ih}$. Since the conditional distribution for κ_h is not a closed form, we need to use a step of MCMC sampling (with appropriate distribution) to draw κ_h .

Sampling for μ and π . Finally, we need to estimate (π, μ) . According to their usual definitions as directional distributions with vMF priors, applying Bayes' rule on the component $z_{ih} = 1$ in Eq. (9) gives the full conditional posterior density function for μ as follows:

$$\begin{aligned} f(\mu_h | \{x_i\}_{i=1}^N, \{z_i\}_{i=1}^N, \mu_0, C_0) &= \frac{\prod_{z_{ih}=1}^N f(x_i | \mu_h, \kappa_h) P(\mu_h | \mu_0, C_0)}{\int \prod_{z_{ih}=1}^N f(x_i | \mu_h, \kappa_h) P(\mu_h | \mu_0, C_0) d\mu} \\ &= \frac{\prod_{z_{ih}=1}^N C_D(\kappa_h) C_D(C_0) \exp\{\kappa_h^T x_i + C_0 \mu_0^T \mu_h\}}{Z_{\mu}} \propto \exp\left\{\kappa_h \sum_i z_{ih} x_i^T + C_0 \mu_0^T \mu_h\right\} \end{aligned} \quad (12)$$

where the updates for concentration parameter and mean direction of the posterior distribution are given by $\|\kappa_h \sum_i z_{ih} x_i + C_0 \mu_0\|$ and $\frac{\kappa_h \sum_i z_{ih} x_i + C_0 \mu_0}{\|\kappa_h \sum_i z_{ih} x_i + C_0 \mu_0\|}$, respectively. The update for the mean parameter μ_h is drawn from the von Mises-Fisher distribution in Eq. (12). Similarly, according to their usual definitions as Multinomial distributions with Dirichlet³ priors, applying Bayes' rule in Eq. (9) yields the posterior distribution for π :

$$\begin{aligned} f(\pi | \{x_i\}_{i=1}^N, \{z_i\}_{i=1}^N, \vec{\alpha}) &= \frac{\prod_{i=1}^N f(\pi | \vec{\alpha}) P(z_i | \pi)}{\int \prod_{i=1}^N f(\pi | \vec{\alpha}) P(z_i | \pi) d\pi} = \frac{1}{Z_{\pi}} \frac{1}{B(\vec{\alpha})} \prod_{h=1}^{(c_h-1)} \prod_{i=1}^N \pi_h^{z_{ih}} \\ &= \frac{1}{Z_{\pi}} \frac{1}{B(\vec{\alpha})} \prod_{h=1}^H \pi_h^{(\alpha_h + n_h - 1)} = \text{Dir}(\vec{\pi} | \vec{\alpha} + \vec{n}_h) \end{aligned} \quad (13)$$

Empirical Bayes estimate for prior parameters. When the user does not have enough information to specify the prior parameters, a general approach is to estimate them directly from the data. The prior parameters can be estimated by maximising the summation of the marginal likelihood of the data. The details are discussed in Appendix B.

4.3 Reversible Jump MCMC Algorithm

This section presents our extension of the reversible jump MCMC algorithm for directional distribution (i.e. vMF distribution), and then ensembles the new algorithm with the Bayesian vMF mixture model to allow adaptive change in the number of components, leading to trans-dimensional von Mises-Fisher mixture model (TvMFMM). Compared with BvMFMM, the proposed TvMFMM has the ability to explore multiple models simultaneously, which brings additional benefit - refining an inappropriate initialization of model parameters which parametric models are incapable of.

³ The Dirichlet distribution of order $H \geq 2$ with parameters $\alpha_1, \dots, \alpha_H > 0$ is given by: $\text{Dir}(\vec{\pi} | \vec{\alpha}) = \frac{1}{B(\vec{\alpha})} \prod_{h=1}^H \pi_h^{\alpha_h - 1}$, where $B(\vec{\alpha})$ is the multinomial Beta function defined as $B(\vec{\alpha}) = \frac{\prod_{h=1}^H \Gamma(\alpha_h)}{\Gamma(\sum_{h=1}^H \alpha_h)}$.

4.3.1 REVERSIBLE JUMP MOVE TYPES

Richardson and Green (1997) developed a methodology to perform Bayesian inference and model exploration for the univariate Gaussian mixture model by using the reversible jump MCMC algorithm, one sweep of which consists of six types of moves:

1. Updating the weights, π , following Eq. (13);
2. Updating the parameters, (μ, κ) , following Eq. (12) for μ and Eq. (11) for κ using MCMC sampling;
3. Updating the allocation, z , following Eq. (10);
4. Updating the hyperparameters, \mathcal{G} ;
5. Splitting one mixture component into two, or combining two into one;
6. The birth or death of an empty component.

Moves (5) and (6) involve changing H by 1 and making necessary corresponding changes to (μ, κ, π, z) , and are used for model exploration via the Metropolis-Hastings algorithm (**Metropolis et al., 1953; Hastings, 1970**). Assume that a proposed move type t , from $s=(Z, \Theta, H)$ to $\tilde{s}=(\tilde{Z}, \Theta', H+1)=f(s, u)$, where $f(s, u)$ is an invertible deterministic function (**Richardson and Green, 1997**). The reverse of the move (from \tilde{s} to s) can be accomplished by using the inverse transformation, so that the proposal is deterministic. The acceptance probabilities from s to \tilde{s} and from \tilde{s} to s are $\min\{1, A\}$ and $\min\{1, A^{-1}\}$ respectively, where

$$\begin{aligned} A &= \text{likelihood ratio} \times \text{prior ratio} \times \text{proposal ratio} \times \text{Jacobian} \\ &= \frac{f(\tilde{s}|X)}{f(s|X)} \times \frac{r_1(\tilde{s})}{r_1(s)q(u)} \times \left| \frac{\partial \tilde{s}}{\partial(s, u)} \right| \\ &= \frac{f(\tilde{s}|Z, \Theta', \mathcal{G}, H+1)}{f(s|Z, \Theta, \mathcal{G}, H)} \times \frac{P(H+1)P(\tilde{Z}, \Theta'| \mathcal{G}, H+1)}{P(H)P(Z, \Theta| \mathcal{G}, H)} \times \frac{r_1(\tilde{s})}{r_1(s)q(u)} \times \left| \frac{\partial \tilde{s}}{\partial(s, u)} \right| \end{aligned} \quad (14)$$

where $r_t(s)$ is the probability of choosing move type t when in state s , $q(u)$ is the density function of the auxiliary random variables u , the final term is the Jacobian determinant arising from the change of variables from (s, u) to \tilde{s} . The birth-death moves in (6) are supplements to the split-merge moves in (5) in a sense that the former are used for empty components, whereas the latter are used for non-empty components.

4.3.2 SPLIT AND MERGE MOVES

In the split-merge move in (5), the RJMCMC algorithm makes a random choice between splitting or merging existing component(s) with probabilities b_h and $d_h=1-b_h$ respectively, depending on h . Generally, $d_1=0$, $b_{H_{max}}=0$, and $b_h=d_h=0.5$ for $h \in [2, H_{max}-1]$. The merging proposal works by choosing two random components j_1 and j_2 , subject to the following constraint (adjacency condition)

$$\mu_{j_1} < \mu_{j_2}, \quad \text{with no other } \mu_s \in [\mu_{j_1}, \mu_{j_2}], s \neq j_1, j_2 \quad (15)$$

In the univariate setting, **Richardson and Green (1997)** proposed the constraints of preserving the zeroth, first and second moments of components before and after the split-merge move. In the multivariate setting, previous works (**Delaportas and Papageorgiou, 2006; Zhang et al., 2004**)

generally preserved the mean vectors and covariance matrices of components via spectral decomposition of the covariance matrices. The first two moments of vMF distribution are mean direction and concentration parameter. However, a direct extension of the RJMCMC algorithm (i.e. preserving the first two moments) for vMF mixture model would be impractical for the split-merge move. Recall that the concentration parameter controls how tightly the distribution is concentrated around the mean direction, which resembles the idea behind covariance matrices - groups of similar data points would result in high (variance) values in the diagonal of the matrices. In other words, to some extent, preserving the concentration parameter is similar to preserving the statistical properties of the covariance matrices. In this work, we make use of the spectral decomposition of the covariance matrices for the split-merge moves of vMF mixtures.

Let $\Sigma_h = V_h \Lambda_h V_h^T$ be the spectral decomposition of the covariance matrix, Σ_h , of the components in Eq. (6), where Λ_h is a diagonal matrix $\Lambda_h = \text{diag}(\lambda_{h1}, \dots, \lambda_{hD})$ with the eigenvalues of Σ_h in increasing order, and V_h is an orthogonal matrix with the eigenvectors of Σ_h in order corresponding to the eigenvalues in Λ_h . Let λ_{hd} denote the d -th largest eigenvalue of Σ_h . Let j_s be one of the H components to be considered to split, j_1, j_2 be the two proposed components, π_{j_1}, π_{j_2} , the corresponding weights, μ_{j_1}, μ_{j_2} , the corresponding mean vectors, $\kappa_{j_1}, \kappa_{j_2}$, the corresponding concentration parameters, and $\Sigma_{j_1}, \Sigma_{j_2}$, the corresponding variance matrices. Let $u_1, u_2 = (u_{21}, \dots, u_{2D})^T, u_3 = (u_{31}, \dots, u_{3D})^T$ be the $2D+1$ random variables needed to construct weights, means and eigenvalues for the split move. They are generated from beta and uniform distributions

$$\begin{aligned} u_1 &\sim \text{Beta}(2, 2), \quad u_2 \sim \text{Beta}(1, 2D), \quad u_{2d} \sim U(-1, 1) \\ u_{31} &\sim \text{Beta}(1, D), \quad u_{3d} \sim U(0, 1), \quad d \in [2, D] \end{aligned} \quad (16)$$

Let P be $D \times D$ rotation matrix with columns orthonormal unit vectors which has $D(D-1)/2$ free parameters. The elements in the lower triangular are randomly generated from uniform distribution $U(0, 1)$, and the elements in other positions are determined by the fact that P is an orthonormal matrix. Then, the proposed split moves are given by

$$\begin{aligned} \pi_{j_1} &= u_1 \pi_{j_s}, \quad \pi_{j_2} = (1 - u_1) \pi_{j_s} \\ \mu_{j_1} &= \mu_{j_s} - \sqrt{\frac{\pi_{j_1}}{\pi_{j_2}}} \left(\sum_{d=1}^D u_{2d} \sqrt{\lambda_{j_s, d}} V_{j_s, d} \right) \\ \mu_{j_2} &= \mu_{j_s} + \sqrt{\frac{\pi_{j_2}}{\pi_{j_1}}} \left(\sum_{d=1}^D u_{2d} \sqrt{\lambda_{j_s, d}} V_{j_s, d} \right) \\ \lambda_{j_1, d} &= u_{3d} (1 - u_{2d}^2) \lambda_{j_s, d} \frac{\pi_{j_1}}{\pi_{j_s}} \\ \lambda_{j_2, d} &= (1 - u_{3d}) (1 - u_{2d}^2) \lambda_{j_s, d} \frac{\pi_{j_2}}{\pi_{j_s}} \\ V_{j_1} &= P V_{j_s}, \quad V_{j_2} = P^T V_{j_s}, \quad d \in [1, D] \\ \mu_{j_1} &= \frac{\mu_{j_1}}{\|\mu_{j_1}\|}, \quad \mu_{j_2} = \frac{\mu_{j_2}}{\|\mu_{j_2}\|} \quad (\text{Normalized mean directions}) \end{aligned} \quad (17)$$

It can be readily shown that these are indeed valid, with weights positive and covariance matrices positive-definite. Now we need to check whether the adjacency condition in Eq. (15) is satisfied. If the condition is satisfied, we reallocate those with $\arg \max_h \gamma(z_{ih}) = j_s$ to j_1 or j_2 using the formula

$P(z_{ih} = 1 | \dots) \propto \pi_{ih} f(x_i | \pi_{ih}, \theta_{ih})_{h=1}^H$. If the test is not passed, then the move is rejected in order to preserve the reversibility of the split/merge move.

The corresponding merge move is specified by the following expressions

$$\begin{aligned} \pi_{j_1} &= \pi_{j_1} / \pi_{j_2} \\ \pi_{j_2} &= \pi_{j_1} + \pi_{j_2} \\ \pi_{j_1} \mu_{j_1} &= \pi_{j_1} \mu_{j_1} + \pi_{j_2} \mu_{j_2} \\ \pi_{j_1} [(\mu_{j_1}^T V_{j_1,d})^2 + \lambda_{j_1,d}] &= \pi_{j_1} [(\mu_{j_1}^T V_{j_1,d})^2 + \lambda_{j_1,d}] + \pi_{j_2} [(\mu_{j_2}^T V_{j_2,d})^2 + \lambda_{j_2,d}] \end{aligned} \quad (18)$$

$$\mu_{j_1} = \frac{\mu_{j_1}}{\|\mu_{j_1}\|} \quad (\text{Normalized mean directions})$$

The solutions of $u_1, u_2, u_3, \lambda_{j_1,d}, V_{j_1}$ and P are as follows:

$$\begin{aligned} u_1 &= \pi_{j_1} / \pi_{j_2} \\ u_{2,d} &= (\mu_{j_1}^T V_{j_1,d} - \mu_{j_2}^T V_{j_1,d}) / \left(\sqrt{\lambda_{j_1,d} \frac{\pi_{j_1}}{\pi_{j_2}}} \right) \\ u_{3,d} &= \pi_{j_1} \lambda_{j_1,d} / [\pi_{j_1} \lambda_{j_1,d} (1 - u_{2,d}^2)] \\ \lambda_{j_1,d} &= \pi_{j_1}^{-1} \left\{ \pi_{j_1} [(\mu_{j_1}^T V_{j_1,d})^2 + \lambda_{j_1,d}] + \pi_{j_2} [(\mu_{j_2}^T V_{j_2,d})^2 + \lambda_{j_2,d}] - (\mu_{j_2}^T V_{j_1,d})^2 \right\} \\ V_{j_1} &= \frac{1}{2} (P^T V_{j_1} + P V_{j_2}), d \in [1, D] \end{aligned} \quad (19)$$

For successful merge move, we have to reallocate those observations x_i with $\arg \max_h \gamma(z_{ih}) = j_1$ or $\arg \max_h \gamma(z_{ih}) = j_2$ to j_1 . At this point, we calculate the acceptance probabilities of split and merge moves: $\min(1, A)$ and $\min(1, A^{-1})$ according to Eq. (14), where

$$A = \frac{P(H+1, \mathcal{Z}, \Theta, \mathcal{B}) \int d_{H+1}}{P(H, \mathcal{Z}, \Theta, \mathcal{B}) b_H P_{alloc} q(t)} \times \left| \det \left(\frac{\partial \Sigma}{\partial (\lambda, V)} \right) \right| \times | \det(\mathcal{J}) | \quad (20)$$

where P_{alloc} is the probability of making this particular allocation of data to j_1 and j_2 given by (Bouguila and Elguebaly, 2012)

$$P_{alloc} = \frac{\prod_{z_{ij}=1} \pi_{j_1} f(x_i | \mu_{j_1}, \kappa_{j_1}) \prod_{z_{ij}=2} \pi_{j_2} f(x_i | \mu_{j_2}, \kappa_{j_2})}{\prod_{z_{ij}=1} \pi_{j_1} f(x_i | \mu_{j_1}, \kappa_{j_1}) + \prod_{z_{ij}=2} \pi_{j_2} f(x_i | \mu_{j_2}, \kappa_{j_2})} \quad (21)$$

The $\frac{\partial \Sigma}{\partial (\lambda, V)}$ term is the Jacobian of the transformation from the Σ of the components to the eigenvalues and eigenvectors (Dellaportas and Papageorgiou, 2006); \mathcal{J} is the Jacobian of the parameter transformation. The calculation of the Jacobian terms and the factorization of $P(H, \mathcal{Z}, \Theta, \mathcal{B})_{(x_i)_{i=1}^N}$ are given in Appendix C and Appendix D.

It is worth noting that it can be computationally inefficient to calculate the Jacobian term $\left| \det \left(\frac{\partial \Sigma}{\partial (\lambda, V)} \right) \right|$ particularly for high-dimensional data. To solve this issue, Zhang et al. (2004) proposed a simplified multivariate Gaussian mixture model (GMM) with reversible jump MCMC algorithm, which imposed a common eigenvector matrix for the covariance matrices of all components. Their experiments showed that their proposed simplified GMM obtains good estimates on general GMMs, especially on their model exploration. In this work, we follow Zhang et al. (2004) and use a common eigenvector matrix for the covariance matrices of all the vMF components when splitting/merging the components.

Algorithm 1: Collapsed Gibbs sampling for TvMFMM

Input:

Data points: $\mathcal{X} = \{x_i\}_{i=1}^N$ (Unit vectors on the hypersphere)
 Prior parameters: $\mathcal{B} = (\alpha, \mu_0, C_0, m, \sigma^2)$
 Initial number of components: H

Output:

Estimation of model parameters
 1 Initialise parameters: $\pi = \{\pi_h\}_{h=1}^H, \mu = \{\mu_h\}_{h=1}^H, \kappa = \{\kappa_h\}_{h=1}^H$;
 2 Initialise latent variables $\mathcal{Z} = \{(z_{ih})_{h=1}^H\}_{i=1}^N$ arbitrarily;
 3 $n \leftarrow 0$;
 4 **while** *NotConverged* **do**
 5 Sample a random variable u from Uniform(0,1): $u \leftarrow U(0,1)$;
 6 **if** $u \leq b_{birth-death}$ **then**
 7 Create or delete an empty component;
 8 **else if** $u \leq b_{birth-death} + b_{split-merge}$ **then**
 9 Split one nonempty component into two, or merge two into one;
 10 **else**
 11 Sample latent variables, z_{ih} , following the parallel sampling in Algorithm 2;
 12 **end if**
 13 **if** *Accept the birth-death/split-merge move or Update latent variables* **then**
 14 Sample the weights, π_h , following Eq. (13);
 15 Sample the mean directions, μ_h , following Eq. (12);
 16 Sample the concentration parameters, κ_h , using MCMC sampling;
 17 Sample hyperparameters;
 18 **end if**
 19 $n \leftarrow n + 1$;
 20 Check for convergence;
 21 **end while**
 22 Employ the k-means style algorithm by Celeux et al. (2000) to solve label switching problem for RJMCMC chains for mixture model estimation;

4.3.3 BIRTH AND DEATH MOVES

Our birth-death move can be adopted straightforwardly from the one used in (Richardson and Green, 1997; Zhang et al., 2004). We first make a random choice between birth or death of an empty component with the same probabilities b_k and d_k as above. For a birth, a mixing weight and parameters of the proposed component are drawn using the following distributions

$$\pi_{j_1} \sim \text{Beta}(1, H), \mu_{j_1} \sim \text{vMF}(\mu_0, C_0), \kappa_{j_1} \sim \text{logNormal}(\ln, \sigma^2) \quad (22)$$

It is necessary to rescale the existing weights in order to ‘make space’ for the new component using $\pi_j \gamma = \pi_j(1 - \pi_{j_1})$, so that $\sum \pi_h = 1.0$. The acceptance probabilities for birth and death moves are

$\min(1, A)$ and $\min(1, A^{-1})$ respectively, where

$$\begin{aligned} A &= \frac{P(H+1)}{P(H)} \times \frac{P(\alpha^2(H+1, \alpha))}{P(\pi(H, \alpha))} \times \frac{r(\beta)}{r(\beta)} \times \left| \frac{\beta^{\beta}}{[\beta(s, u)]} \right| \\ &= \frac{P(H+1)}{P(H)} \times \frac{\prod_{h=1}^{L(H+1, \alpha)} \pi_{r_h}^{m_h + \alpha - 1}}{\prod_{h=1}^{L(H, \alpha)} \pi_{r_h}^{m_h + \alpha - 1}} \times \frac{d_{H+1}}{(H_0 + 1)^{b_H}} \times \frac{1}{g_{1,H}(\sigma_{r_j})} \times (1 - \pi_j)^{H-1} \\ &= \frac{P(H+1)}{P(H)} \times \frac{1}{\prod_{i=1}^H \pi_{r_i}^{m_i + \alpha - 1}} \times \frac{d_{H+1}}{(H_0 + 1)^{b_H}} \times \frac{1}{g_{1,H}(\sigma_{r_j})} \times (1 - \pi_j)^{H-1} \end{aligned} \quad (23)$$

In Eq. (23), H_0 is the number of empty components, $g_{1,H}(\cdot)$ is the probability density function of $\text{Beta}(1, H)$ distribution.

In our implementation of the reversible jump MCMC algorithm, rather than passing through each of the six moves deterministically, following (Andrieu et al., 2003; Dellaportas and Papageorgiou, 2006), we choose to randomly select one of the three moves (i.e. Moves (3), (5) and (6)) with fixed probabilities in each iteration. We have used (1., 4., 5) as probabilities to choose the three moves respectively. This allows some extra tuning which can potentially speed up the convergence and improve the mixing of the RJMCMC chain. The resulting collapsed Gibbs sampling procedure of TVMFMM is summarised in Algorithm 1. Note that we employ parallel sampling to update the latent variables in Algorithm 2, which is very similar to state synchronization in the parallel topic models proposed by Smola and Narayanaswamy (2010).

Algorithm 2: Parallel sampling for latent variables \mathcal{Z}

```

1 function INFERENCE( $X, \mathcal{Z}, \alpha, \mu_0, C_0, m, \sigma^2$ )
2   Initialise  $\gamma^{old}(z_{ih}) = \gamma(z_{ih})$  for all  $i, h$ ;
3   while Sampling do
4     Read global stats  $\{n_h\}_{h=1}^H$ 
5     for every component  $h \in [1, H]$  do
6       Sample a latent,  $z_{ih}$ , conditioned on all other labels following Eq. (10).
7     end
8     Normalise local latent variables for data point  $x_i$ :
9        $\gamma(z_{ih}) = \frac{\gamma(z_{ih})}{\sum_{h=1}^H \gamma(z_{ih})}$  for every  $h \in [1, H]$ 
10    Lock  $\{n_h\}_{h=1}^H$  globally.
11    Update  $n_h = n_h + [\gamma(z_{ih}) - \gamma^{old}(z_{ih})]$  for every  $h \in [1, H]$ .
12    Release  $\{n_h\}_{h=1}^H$  globally.
13  end
    
```

4.4 Online Trans-dimensional vMF Mixture Model

This section presents the procedures of collapsed Gibbs sampling for the proposed online trans-dimensional von Mises-Fisher mixture model (OTVMFMM). Given data, $X = \{x_i, i=1, \dots, T\}$, where $x_i \in \mathbb{R}^D$, OTVMFMM assumes the generative model for the data given in Figure 2.

The user specified prior parameters are $\mathcal{B} = (\alpha, \mu_{0,0}, C_0, m, \sigma^2)$. To some extent, the prior parameter C_0 acts as a smoothing term to ensure that the parameters of the next time epoch to be similar to the previous one. Following Gopal and Yang (2014), the concentration parameters of the clusters at time t ($k_{t,i}$) are drawn from a log-Normal distribution with mean m and variance σ^2 . The

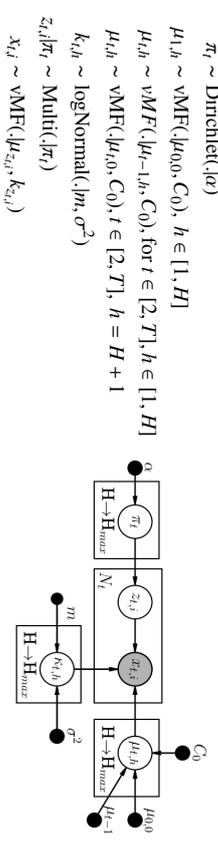


Figure 2: A graphical model representation of OTVMFMM, in which nodes represent random variables, arrows denote dependency among variables, and plates denote replication.

cluster-specific mean parameters at time t ($\mu_{t,h}$) are drawn from a vMF distribution centered around the corresponding clusters at the previous time $t-1$ or centered around $\mu_{0,0}$ with concentration C_0 . This evolutionary change of the cluster parameters introduces flexibility and enables OTVMFMM to accommodate smooth changes in the mean parameter within a given cluster over time.

The inference and reversible jump MCMC algorithm for OTVMFMM can be adapted straightforwardly from those of TVMFMM introduced in Section 4.2, 4.3. The likelihood of the complete data is given by

$$\begin{aligned} &P(H, \{x_i, i=1, \dots, T\}, \{z_i, i=1, \dots, T\}, \{\pi_{t,h}, \theta_{t,h}\}_{h=1}^H | \mathcal{B}) \\ &= P(H) f(\pi | \alpha) P(\{z_i, i=1, \dots, T\} | \pi) f(\mu | \mu_{0,0}, C_0) f(k | m, \sigma^2) f(\{x_i, i=1, \dots, T\} | \{\pi_{t,h}, \theta_{t,h}\}_{h=1}^H) \\ &= P(H) f(\pi | \alpha) \prod_{h=1}^H f(\mu_{t,h} | \mu_{t-1,h}, C_0) f(k_{t,h} | m, \sigma^2) \prod_{i=1}^N P(z_i, i | \pi) f(x_i | \mu_{z_i}, k_{z_i}) \end{aligned} \quad (24)$$

The likelihood of x_i can be defined as one of its marginal distributions by integrating out the distributions $\pi_{t,h}$, $\mu_{t,h}$ and $k_{t,h}$ and summing over $z_{t,i}$:

$$\begin{aligned} &f(x_i | m, \sigma^2, \mu_{t-1,h}, C_0, \alpha) = \\ &= \int_{\pi} \int_{\mu} \int_{k} f(\pi | \alpha) \prod_{h=1}^H P(z_i, i | h) = 1) f(x_i | \mu_{t,h}, k_{t,h}) f(\mu_{t,h} | \mu_{t-1,h}, C_0) f(k_{t,h} | m, \sigma^2) \\ &= \int_{\pi} f(\pi | \alpha) \prod_{h=1}^H P(z_i, i | h) = 1) \int_{\mu} \int_{k} \prod_{h=1}^H f(x_i | \mu_{t,h}, k_{t,h}) f(\mu_{t,h} | \mu_{t-1,h}, C_0) f(k_{t,h} | m, \sigma^2) \end{aligned} \quad (25)$$

Similarly to Section 4.2, we obtain the following updates for the posterior parameters.

$$\begin{aligned}
 \gamma(z_{i,t,h}) &\equiv P(z_{i,t,h} = 1 | \mathbf{Z}_{i,t-h}, \{x_{i,t}\}_{i=1}^{N_t}, \kappa_t, m, \sigma^2, \mu_{t-1,h}, C_0, \alpha) \\
 &\propto (n_{t,h} + \alpha) C_D(\kappa_{t,h}) \frac{C_D(\|\kappa_{t,h} \sum_{j \neq i} z_{i,j,h} x_{i,j} + C_0 \mu_{t-1,h}\|)}{C_D(\|\kappa_{t,h} \sum_{j \neq i} z_{i,j,h} x_{i,j} + C_0 \mu_{t-1,h}\|)} \\
 f(\kappa_{t,h} | X_t, \mathbf{Z}_t, \kappa_t, m, \sigma^2, \mu_{t-1,h}, C_0) &\propto \frac{C_D(\kappa_{t,h})^{n_{t,h}} C_D(C_0)}{C_D(\|\kappa_{t,h} \sum_j z_{i,j,h} x_{i,j} + C_0 \mu_{t-1,h}\|)} \log \text{Normal}(\kappa_{t,h} | m, \sigma^2) \\
 f(\mu_{t,h} | X_t, \mathbf{Z}_t, \mu_{t-1,h}, C_0) &\propto \exp \left\{ \left\{ \kappa_{t,h} \sum_i z_{i,t,h} x_{i,t} + C_0 \mu_{t-1,h} \right\} \mu_{t,h} \right\}
 \end{aligned} \tag{26}$$

$$f(\pi_{t,h} | \{x_{i,t}\}_{i=1}^{N_t}, \{z_{i,t}\}_{i=1}^{N_t}, \alpha) \propto (n_{t,h} + \alpha)$$

where $n_{t,h}$ is the number of observations assigned to the h -th component at time t , $n_{t,h} = \sum_i z_{i,t,h}$. The empirical updates for the prior parameters are given as follows

$$\begin{aligned}
 \mu_{t,0} &= \frac{\sum_{h=1}^H \mu_{t,h}}{\|\sum_{h=1}^H \mu_{t,h}\|}, C_0 = \frac{\bar{r}D - \bar{r}^3}{1 - \bar{r}^2}, \text{ where } \bar{r} = \frac{\|\sum_{h=1}^H \mu_{t,h}\|}{H}, t \in [2, T] \\
 \arg \max_{\alpha > 0} -\log \left(\prod_{h=1}^H \Gamma(\alpha) \right) + (\alpha - 1) \sum_{h=1}^H \pi_{t,h} &\tag{27} \\
 m &= \frac{1}{H} \sum_{h=1}^H \log(\kappa_{t,h}), \sigma^2 = \frac{1}{H} \sum_{h=1}^H \log(\kappa_{t,h}^2) - m^2
 \end{aligned}$$

Similarly, the move (5) and (6) of the RJMCMC algorithm for OTvMFMM can be straightforwardly derived following the strategy for TvMFMM in Section 4.3. The acceptance probabilities of split and merge moves are $\min(1, A)$ and $\min(1, A^{-1})$ respectively, where

$$A = \frac{P(H+1, \mathbf{Z}_t, \{\pi_{t,h}, \theta_{t,h}\}_{h=1}^H, \mathcal{B} | X_t)}{P(H, \mathbf{Z}_t, \{\pi_{t,h}, \theta_{t,h}\}_{h=1}^H, \mathcal{B} | X_t)} \times \frac{d_{H+1}}{b_H P_{\text{alloc}}(u)} \times \left| \frac{\partial \Sigma}{\partial(\lambda, V)} \right| \times | \det(J) | \tag{28}$$

The acceptance probabilities for birth and death moves are $\min(1, A)$ and $\min(1, A^{-1})$ respectively, where

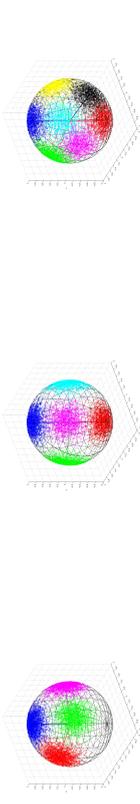
$$\begin{aligned}
 A &= \frac{P(H+1)}{P(H)} \times \frac{1}{B(\alpha, H\alpha)} \times \pi_{i,j}^{\alpha-1} (1 - \pi_{i,j})^{N+H\alpha-M} \times (H+1) \\
 &\times \frac{d_{H+1}}{(H_0+1)b_H} \times \frac{1}{g_{1,H}(\pi_{i,j})} \times (1 - \pi_{i,j})^H
 \end{aligned} \tag{29}$$

In Eq. (29), H_0 is the number of empty components, $g_{1,H}(\cdot)$ is the probability density function of $Beta(1, H)$ distribution.

5. Empirical Evaluation

In this section, we evaluate the proposed models on synthetic and real-world data. We used the movMF software⁴ provided by Banerjee et al. (2005) to generate synthetic data with: a) 4 well-separated components; b) 5 well-separated components; c) 7 not well-separated components. Each of the synthetic datasets has a training size of 10000 and held-out test data size of 2500. The visualisation of synthetic dataset is presented in Figure 3.

4. <http://suwit.de/woztk/soft/movmf/>



(a) 4 well-separated components (b) 5 well-separated components (c) 7 not well-separated components
Figure 3: Visualisation of synthetic data on 3-d unit sphere.

Preprocessing of Wikipedia data. In Wikipedia, pages are subdivided into ‘namespaces’⁵ which represent general categories of pages based on their function. For instance, the article (or main) namespace is the most common namespace and is used to organise encyclopedia articles. In many practical applications, we might be more interested in how the actual interests of editors (in terms of the categories of Wikipedia articles they have edited) change over time. For this reason, rather than using the main namespace as one feature, we further group Wikipedia articles into clusters based on their macro-categories⁶. Because the categories for articles given by Wikipedia are generally not fine-grained, we infer the macro-categories for articles by identifying candidate categories from the DBpedia⁷ category graph. DBpedia is one of the best known multi-domain knowledge bases which extracts structure information from Wikipedia Categorization system and forms a semantic graph of concepts and relations. The association between Wikipedia categories and DBpedia concepts is defined using the **subject** property of the DCIM terms vocabulary (pre-fixed by **dcterms**) (Hulpus et al., 2013). A category’s parent and child categories can be extracted by querying for properties **skos:broad** and **skos:broaderof**, these category-subcategory relationships create connections between DBpedia concepts. We can obtain a DBpedia category graph⁸ by merging all the connections among DBpedia concepts together. With the category graph available, we can identify the macro-categories for Wikipedia articles by searching for the shortest paths from the categories associated with the articles to the macro-categories in the category graph. If multiple shortest paths exist, then the article is assigned to multiple macro-categories with weights proportional to the number of paths leading to a specific macro-category. For other complex methods of labelling topics, we recommend the readers refer to Hulpus et al. (2013).

Users can make edits to any namespace or article based on their interests and expertise. The amount of edits across all the 28 namespaces and 22 macro-categories can be considered as work archetypes. A namespace or macro-category can be considered as a ‘term’ in the vector space for document collections, the number of edits to that namespace/category is analogous to word frequency. A user’s edit activity across different namespaces/categories in a time period can be regarded as a ‘document’. The main motivation of this work is to apply topic models on the evolving user behavioural data in order to identify and characterise the patterns of change in user edit activity

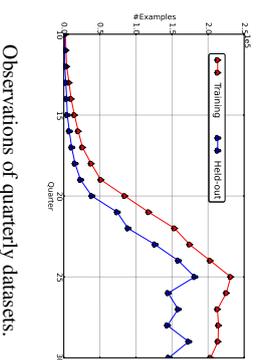
5. <http://en.wikipedia.org/wiki/Wikipedia:Namespace>

6. At the time we collected data for this work, there were 22 macro-categories: http://en.wikipedia.org/wiki/Category:Main_topic_classifications

7. <http://dbpedia.org>

8. The category graph is a directed one due to the nature of category-subcategory structure.

(i.e. common work archetypes) over time in Wikipedia. For this purpose, we parsed the May 2014 dump of English Wikipedia⁹, collected the edit activity of all registered users, then aggregated the edit activity of each user on a quarterly basis. In this way, we obtained a time-varying dataset consisting of the quarterly editing activity of all users from the inception of Wikipedia till May 2nd, 2014. There is an overwhelming number of users who stayed active for only one quarter. To avoid a bias towards behaviours most dominant in dataset with larger user bases, following [Furtado et al. \(2013\)](#), we randomly selected 20% of users who stayed active for only one quarter, included these users and those who were active for at least two quarters as our training dataset; the remaining 80% of short-term users were used as held-out dataset. The statistics and an example of the dataset are given in [Figure 4](#). The size of the quarterly datasets range from several hundreds to about 200,000.



Observations of quarterly datasets.

An example of a simple entry in the dataset.

Username	Quarter	Mathematics	Science	Article talk
User A	10	233	650	2

	Wikipedia	Wikipedia talk	user	user talk
	299	33	2	81

Dataset Statistics.		
Wikipedia dataset	#Features	#Quarters
	50	21
		#True clusters
		unknown

Figure 4: Statistics and an example of Wikipedia dataset.

We implement the models in Python, and parallelize the models wherever possible by using the parallel functionality of Python. Specifically, we make use of the *multiprocessing*¹⁰ package to implement parallelism for parameter updates, and use *Value* and *Array* data structures provided by the package to enable data sharing among multiprocessors.

Experimental settings. All our experiments were run on 32 core AMD Opteron 6134 @ 2.25Ghz with 252GB RAM. The main computational bottleneck in our collapsed Gibbs sampling algorithm is the computation of z and κ . We compared the proposed models with the following algorithms and models that are widely-used in the literature:

- K-means ([Hartigan and Wong, 1979](#)) and MiniBatchKMeans ([Sculley, 2010](#)) algorithms with random and k-means++ ([Arthur and Vassilvitski, 2007](#)) initialization for the centroids. Different initialization of the centroids for k-means can affect its convergence and may lead to a local minimum. The k-means++ initialization scheme selects initial cluster centers in a heuristic way which can speed up convergence and lead to better results than random initialization.
- Non-negative matrix factorization (NMF) model ([Lee and Seung, 1999](#)). We used the Non-negative Double Singular Value Decomposition (NNDSDVD) strategy ([Boutsidis and Gallopoulos, 2008](#)) to choose initial factors for NMF, which can produce deterministic results and avoid a poor local minimum.

- Dynamic topic model (DTM)¹¹ ([Blei and Lafferty, 2006](#)), designed for clustering of temporal document collections represented in term-frequency style format.
- Bayesian von Mises-Fisher mixture model (BvMFMM) ([Gopal and Yang, 2014](#)), designed for clustering of static numeric (and directional) data. The model is initialised with k-means++ ([Arthur and Vassilvitski, 2007](#)) method.
- Dirichlet process Gaussian mixture model (DP-GMM) with sub-cluster split/merge moves¹² ([Chang and Fisher III, 2013](#)).

The generated synthetic data is \mathcal{L}_2 normalised. For Wikipedia datasets, we used the tf-idf normalised representation for NMF, BvMFMM, DP-GMM and OTvMFMM, and feature count representation (without normalisation) for DTM. For k-means, MiniBatchKMeans and NMF, we used the implementation in the scikit-learn package. If not specified, we run the models with their default parameters. If not explained specifically, TvMFMM is initialised with k-means++ strategy, and OTvMFMM is initialised with posterior estimation from the previous time point. To determine the number of clusters for parametric models (i.e. NMF, BvMFMM, and DTM) on real-world data, we experimented with different number of clusters $k \in [5, 45]$ with steps of 5 on the quarterly Wikipedia editor datasets using Non-negative Matrix Factorization (NMF) clustering, and then employed measures such as normalised pairwise mutual information (NPMI) as suggested by [O’Callaghan et al. \(2015\)](#) to assess model coherence for different ks . We found that overall, the run with 10 clusters generates more coherent clusters, so we set $H = 10$ for parametric models. Detailed analysis is given in [Appendix F](#).

Chaining of clusters¹³. Label switching of the components (stemming from the posterior distribution being invariant with respect to the permutation of the component labels) is an issue which needs to be solved when analyzing MCMC samples. Briefly, label switching can be solved by imposing certain ordering constraints on component parameters. [Celeux et al. \(2000\)](#) proposed a k-means style clustering algorithm to deal with label switching problem for MCMC chains, and showed the advantages and justification of k-means clustering over loss functions for label switching problem in terms of avoiding storage of the complete MCMC chain. In this work, we employ their k-means style algorithm for two purposes: (1) to solve label switching problem for RMCMC chains for mixture model estimation, and (2) to chain the clusters in different quarters together in order to visualise the popularity and dynamics of clusters over time.

To evaluate the influence of the two different strategies for split-merge moves, we compared the performance of TvMF mixture model with/without common eigenvectors for split-merge moves on synthetic data. The results suggest that TvMFMM with two different strategies for the split-merge moves show similar performance and mixing properties on synthetic data. Therefore, for the results presented in the following sections, we ran TvMFMM and OTvMFMM with common eigenvectors for split-merge moves. The detailed results are provided in [Appendix E](#).

The prior parameters for all the vMF mixtures are $\{\alpha, \mu_0, C_0, m, \sigma^2\}$. Although we provide an empirical Bayes step to estimate the priors from data, estimating too many parameters is prone to problems such as overfitting. The acceptance rate of the birth-death move in [Eq. \(23, 29\)](#) is sensitive to the value of α , larger value of α tends to result in lower acceptance rate for the move. We set

⁹ <http://dumps.wikimedia.org/enwiki/20140502/>
¹⁰ <https://docs.python.org/2/library/multiprocessing.html>

¹¹ Available at: <https://www.cs.princeton.edu/~blei/topicmodeling.html>
¹² Available at: <http://people.csail.mit.edu/johang7/code.php>
¹³ Throughout the paper, we use the terms topic, component, cluster, mixture, and common user role interchangeably to denote the same concept.

$\alpha = 1.0$ for all models. The prior parameter μ_0 is estimated using empirical Bayes. Gopal and Yang (2014) suggested setting the prior parameter manually with relative low values (typically smaller than 1.0) rather than directly learning it from data. Following the suggestion, we set $C_0 = 2.0$ for all the vMF mixtures. The prior parameters m and σ^2 control the range of the concentration parameters κ , which can affect the mixing property / convergence of the RJMCMC chain. More details about this effect are discussed in Appendix H. In the following analyses (excluding those in Appendix H), we used the trace plot of the number of components and log likelihood over sweeps to assist setting appropriate values for m and σ^2 .

5.1 Clustering Performance on Synthetic Data

This section compares the clustering performance of TvMF mixture model with other models on synthetic data. Following Gopal and Yang (2014), we compared the clustering performance of TvMFMM with other models on synthetic datasets using the following performance metrics:

- Adjusted Rand Index (ARI) ¹⁴.
- Normalised Mutual Information (NMI) ¹⁴.
- Adjusted Mutual Information (AMI) ¹⁴.
- Purity-related metrics ¹⁴: Homogeneity, quantifies the extent to which each cluster contains only members of a single class; Completeness, quantifies the extent to which all members of a given class are assigned to the same cluster.

Table 2 presents the comparison of clustering performance. The results are the average of 10 runs for each method. K-means and MiniBatchKMeans algorithms are run with the true number of components and hard assignments for each dataset but with 2 different strategies to initialise the centroids. BvMFMM is also run with the true number of components. The posterior estimation of the number of components for TvMFmix are 4, 5, and 7 for the three synthetic datasets, respectively, based on the results in Appendix E. To further verify our findings, we conducted two-way significance tests using paired t-tests between TvMFMM and other models for every metric. The null hypothesis is that there is no significant difference between the performance of TvMFMM and other models.

The results show that TvMFMM performs statistically significantly better than BvMFMM on synthetic datasets with 7 not well-separated components, but both models have similar performance on datasets with 4 and 5 well-separated components; TvMFMM performs statistically significantly better than DP-GMM on all the three synthetic datasets; TvMFMM has similar Homogeneity scores as k-means and MiniBatchKMeans, but performs statistically significantly better than k-means and MiniBatchKMeans on the other four performance metrics. Note that the clustering performance of BvMFMM relies on setting the optimal number of clusters and reasonable initialisation of cluster centres, while TvMFMM solves the two issues via the use of RJMCMC algorithm. For BvMFMM, even if the number of clusters could be tuned carefully to be the optimal number for a specific dataset, a reasonable initialisation of the cluster parameters (i.e. mean directions) would still be difficult particularly for not well-separated datasets. This explains why TvMFMM performs better than BvMFMM on the synthetic dataset with 7 components. Overall, the results suggest that the proposed TvMF mixture model performs significantly better than other widely-used models such as, k-means, BvMFMM, and DP-GMM on synthetic data, i.e. points on the unit sphere.

Table 2: Comparison of clustering performance for different models on synthetic data. Bold face numbers indicate best performing method for the corresponding metric. The results of the paired t-test against TvMFMM are denoted by: * for significance at 5% level, ** for significance at 1% level.

True H	ARI	AMI	NMI	Homogeneity	Completeness
TvMFMM					
4	0.925	0.898	0.899	0.898	0.899
5	0.947	0.941	0.945	0.947	0.944
7	0.809	0.821	0.829	0.827	0.832
BvMFMM					
4	0.881	0.858	0.86	0.859	0.861
5	0.94	0.931	0.931	0.931	0.932
7	0.612**	0.676**	0.719**	0.677**	0.765*
DPGMM					
4	0.662**	0.652**	0.726**	0.652**	0.811**
5	0.599**	0.626**	0.727**	0.626**	0.847**
7	0.436**	0.494**	0.614**	0.495**	0.767*
Kmeans with kmeans++ initialization					
4	0.517**	0.577**	0.726**	0.911	0.578**
5	0.675**	0.710**	0.825**	0.958	0.711**
7	0.769*	0.766**	0.808*	0.853	0.767**
Kmeans with random initialization					
4	0.516**	0.576**	0.724**	0.909	0.577**
5	0.674**	0.710**	0.825**	0.958	0.711**
7	0.770*	0.766**	0.809*	0.854	0.767**
MiniBatchKMeans with kmeans++ initialization					
4	0.540**	0.581**	0.726**	0.906	0.582**
5	0.692**	0.714**	0.826**	0.954	0.715**
7	0.765*	0.764**	0.807*	0.85	0.765**
MiniBatchKMeans with random initialization					
4	0.544**	0.584**	0.729**	0.909	0.585**
5	0.710**	0.722**	0.833**	0.959	0.723**
7	0.775*	0.770**	0.811*	0.853	0.771**

5.2 Performance on Real-world Data

In this section, we evaluate the performance of the proposed online trans-dimensional von Mises-Fishes mixture model (OTvMFMM) on Wikipedia editor activity data over 21 quarters. Because the first 9 quarters had a small number of user activity records (generally less than 1000), we choose to analyse the editor activity from the 10-th quarter to the 30-th quarter.

14. <http://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation>

5.2.1 CONVERGENCE DIAGNOSTICS OF OTVMFMM

Before proceeding to model exploration, we diagnose the convergence of the OTVMFMMs. We follow the methodology of [Brooks and Giudici \(2000\)](#) which monitors particular functions of parameters (e.g. log likelihood). The method requires running I independent chains with $2T$ iterations, and then divides the I sequences into batches of length b , which gives a series of sequences of chains with length $2kb$ (where $k \in [1, T/b]$). With sequences of chains ready, we then calculate the total variations of log likelihood both between chains and between models to diagnose the convergence of the RJMCMC chain. Following [Brooks and Giudici \(2000\)](#), six quantities are computed:

- The total variation \hat{V} and the within-chain variance W_c . Essentially, the ratio \hat{V}/W_c is analogous to the potential scale reduction factor (PSRF, denoted by \hat{R}) of [Gelman and Rubin \(1992\)](#).
- The within-model variance W_m , the variance within both chains and models $W_m W_c$. The comparison of W_m and $W_m W_c$ which should well approximate the true mean of within-model variance, tells us how well the chains are mixing within models.
- The between-model variance B_m , and the within-chain variation split between and averaged over models $B_m W_c$. The comparison of B_m and $B_m W_c$, which should well approximate the true between-model variance, tells us how well the chains are mixing between models.

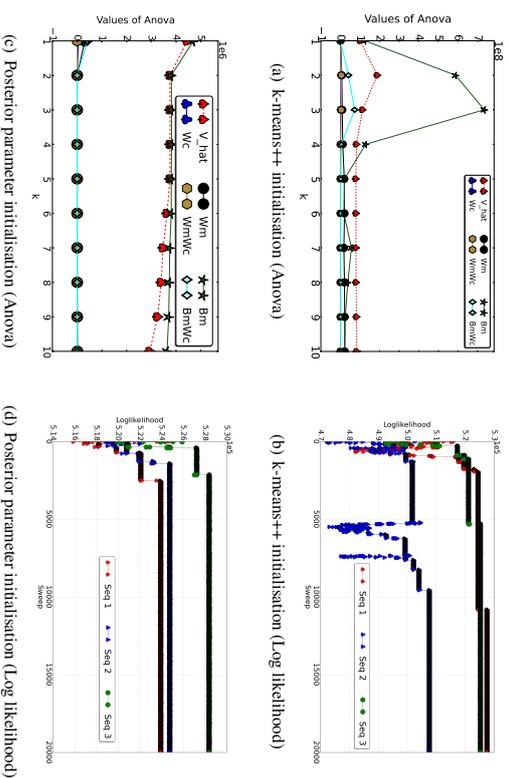


Figure 5: Diagnostic plots for convergence analysis and trace of the log likelihood on Wikipedia data: (a)-(b) models started with k-means++ initial means; (c)-(d) models started with the posterior mean of the previous quarter.

For details of the quantities readers can refer to [Brooks and Giudici \(2000\)](#). Three independent chains (each chain running 20000 iterations) were used to diagnose the convergence of OTVMFMMs started with k-means++ initial mean parameters (the initial number of components was set to 10) and started with the posterior mean of the previous quarter, respectively. We use log likelihood as the scalar parameter for convergence diagnostics which has also been used by ([Brooks and Giudici, 2000](#); [Zhong and Girolami, 2009](#)). Figure 5 gives the diagnostic plots of convergence analysis on the 15th quarter of Wikipedia editor activity data.

The results showed that in general, OTVMFMMs started with k-means++ initial mean parameters have higher level of variations in log likelihood than those started with posterior parameter initialisation. OTVMFMMs started with k-means++ initial mean parameters became convergent after $k=4$, as evidenced in the corresponding trace plot for log likelihood: the overall variations of log likelihood for OTVMFMMs started with posterior mean initialisation were relatively stable throughout the range of k s, the chains were mixing very well after 3000 iterations. The results suggested that OTVMFMM started with posterior mean initialisation converges faster than that started with k-means++ initial mean parameters. When analysing the massive temporal Wikipedia data, we notice that OTVMFMMs started with posterior mean initialisation tend to converge within 3000 iterations; we generally determine the convergence of OTVMFMM by checking the trace of the log likelihood.

5.2.2 QUALITATIVE ANALYSIS

To show how the OTVMFMM can generate more coherent, interpretable and intuitive clusters (or common user roles) than existing models for time-varying user behavioural data, we compare the popularity of clusters over time and the evolution of top terms for selected similar clusters identified by different models. Figure 6 presents the trends of clusters over time for different models¹⁵. We hand-labelled the clusters generated by DTM and OTVMFMM evolve relatively smoothly in their trends over time: NMF tends to generate many clusters that appear in less than 4 quarters, indicating smoothness issues in the clusters; DTM fails to capture the birth/death of clusters over time, while OTVMFMM can capture the birth/time of clusters over time; BvMFMM generates the least smooth clusters in terms of popularity over time. Comparing the cluster labels for DP-GMM and OTVMFMM in Figure 6, we notice that DP-GMM tends to generate fewer clusters that are relatively general, whereas OTVMFMM tends to generate a larger number of clusters that are specific, interpretable and intuitive.

Figure 7 visualises the evolution of top terms for selected common user roles identified by different models. We observe that user roles generated by DTM and OTVMFMM generally contain a few most dominant terms, indicating interpretable and intuitive clusters; NMF also generates a few interpretable clusters, BvMFMM and DP-GMM tends to generate clusters with many dominant terms, indicating general and less interpretable clusters. The visualisation of more common user roles are provided in Figure 18 of [Appendix 1](#). Overall, the results suggest that DTM and OTVMFMM tend to generate interpretable and intuitive clusters, NMF tends to generate many clusters that appear in less than 4 quarters, BvMFMM and DP-GMM tend to generate general clusters that lack of most dominant terms.

[Appendix G](#) presents a discriminative analysis of the topical representations by different models on the 18th quarter of Wikipedia Editor dataset.

¹⁵. To improve readability, we only visualise the trends for clusters that appear at least 4 quarters.

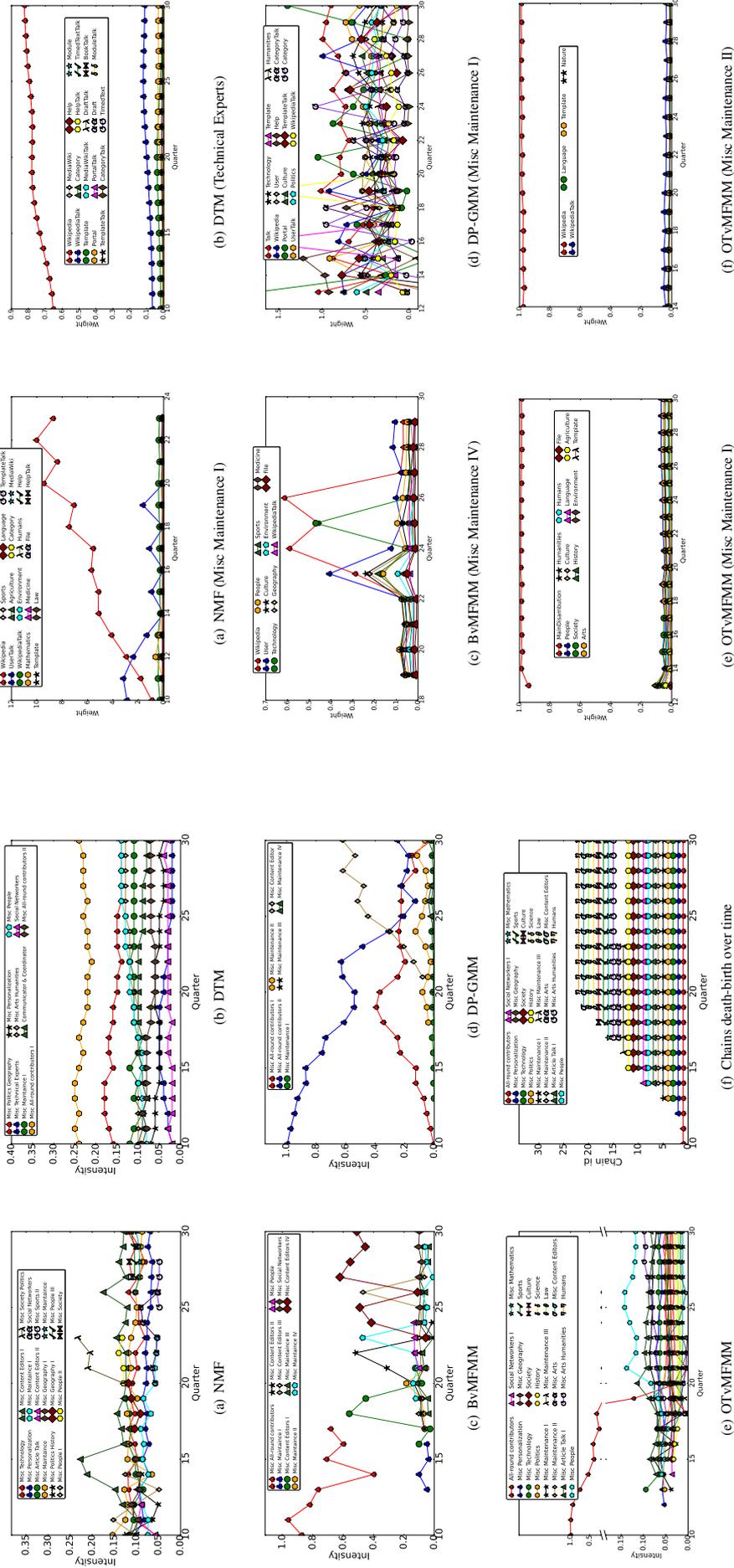


Figure 6: Popularity of common user roles over time for different models and birth-death of user roles over time for OTvMFMM. Quarter corresponds to the index of quarter. Intensity indicates the weight of user roles in each quarter, and is calculated as the percentage of user profiles assigned to that user role in a quarter. Each curve represents the trend of one user role over time. Chain id indicates the corresponding user role for OTvMFMM. NB: in (e), we make use of a discontinuity in the y-axis to better visualise the detail in the bottom of the plot.

Figure 7: Evolution of top terms for similar common user roles (*Misc Maintenance*) identified by different models. Weight indicates the weight of features for the user roles, and is available from the model parameters.

5.2.3 QUANTITATIVE ANALYSIS

We evaluate the performance of OTvMFMM and its counterparts quantitatively from two aspects: the coherence of the clusters generated, and the perplexity (equivalently, held-out log likelihood) of the models. Traditional predictive metrics, such as perplexity, are commonly used in the literature to evaluate topic models; these metrics capture the model’s predictive ability over a test set of un-

seen documents based on the parameters learned from a training set (Chang et al., 2009). Following these authors, the predictive likelihood of data point x can be approximated using $P(x | \chi_{train}) = \int_0^1 P(x, \Theta | \chi_{train}) \approx P(x | \Theta) P(\Theta | \chi_{train})$, where Θ are the posterior estimation of model parameters learned from the training set. Chang et al. (2009) showed that perplexity was often negatively correlated with human judgements of topic quality, and suggested alternative measures such as topic coherence or focusing upon real-world task performance that includes human knowledge to evaluate topic quality. Topic coherence can capture the semantic interpretability of discovered topics based on their corresponding descriptor terms using measures such as normalised Pointwise Mutual Information (NPMI) (Chang et al., 2009; O’Callaghan et al., 2015). Higher coherence scores indicate better semantic interpretability, thus more coherent and interpretable topics. Figure 8 compares the coherence of clusters, and held-out log likelihood for different models¹⁶.

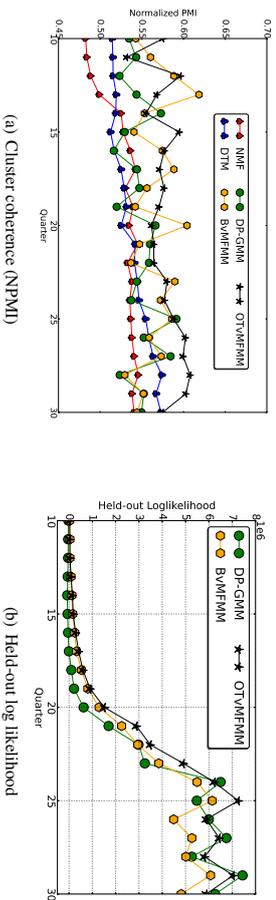


Figure 8: Mean normalised PMI and held-out log likelihood.

We observe that the NPMI values of OTvMFMM are constantly higher than those of NMF and DTM in the range of quarters considered; the NPMI values of OTvMFMM are higher than those of BvMFMM and DP-GMM in most quarters. The NPMI values of BvMFMM and DP-GMM experience certain level of fluctuation compared with those of the other three models. The held-out log likelihood of OTvMFMM is constantly higher than that of BvMFMM in all the quarters considered; the held-out log likelihood of OTvMFMM is higher than that of DP-GMM from quarter 10 to 23, after which the measures of the two models are approximately at the same level. To summarise, the results suggest that OTvMFMM presents better predictive ability on unseen data than BvMFMM and DP-GMM, and that OTvMFMM generates more interpretable and coherent clusters than other models.

5.2.4 TIME ANALYSIS

Table 3 compares the learning time of different models on the 13th to 16th quarter of Wikipedia datasets. For DTM, the time reported was the total learning time on the 4 datasets. It is obvious from the table that NMF took the least time to learn the model; DTM took the second least time to learn the model; DP-GMM and BvMFMM took about the same amount of time to train the

¹⁶ We did not compare with the perplexity of DTM because DTM inferred model parameters using variational Bayes, and gave a lower bound of held-out log likelihood.

model. Comparing with the other four models, OTvMFMM took the maximum amount of time to learn the model on all the datasets considered. This is due to two reasons: (i) it generally takes a substantial amount of time for reversible jump MCMC styled algorithms (e.g. Richardson and Green (1997); Zhang et al. (2004); Delaportas and Papageorgiou (2006); Zhong and Giridhari (2009)) to generate convergent MCMC chains; (ii) compared with other models, updating the latent variables in OTvMFMM involves calculating the normalising constants of von Mises-Fisher distributions, which is computationally expensive, particularly for large datasets.

Table 3: Learning time in seconds for number of iterations (#iterations) after which the algorithms converged for different models.

	NMF	DTM	DP-GMM	BvMFMM	OTvMFMM	#Obs
#iterations	–	200	5000	40	10000	–
Quarter 13	12.65		867.95	735.66	53333.33	7344
Quarter 14	15.83		1170.56	1503.28	69504.62	10217
Quarter 15	32.34	6217.0	1722.03	1742.40	119349.18	14829
Quarter 16	27.92		2215.67	2032.45	108577.73	20129

6. Applications of User Profiles

In this section, we explore: (1) how discriminative are the generated features in distinguishing different groups of users, and (2) how useful are the features generated from patterns of change in editor activities by different models for the churn prediction task. Identifying the key features that distinguish different user groups and different life stages makes it possible to develop techniques for important applications, such as churn prediction and task recommendation. Churners present a great challenge for community management and maintenance as the turnover of established members can have a detrimental effect on the community in terms of creating communication gaps, knowledge gaps or other gaps. Qin et al. (2014) presented similar applications of user profiles. This work replicates their application scenarios in order to provide insights into the usefulness of the features generated by the proposed models in real-world applications.

6.1 Group Level Change in User Profiles

Different users are more likely to follow slightly or totally different trajectories in their lifecycle. In this analysis, we examine how different groups of users evolve throughout their lifecycle periods by comparing how each user’s profile in one period is different from that in the previous periods. The historical comparison of the distribution of user profiles toward user roles is a useful indicator of how the user changes edit activity relative to past behaviour. Cross-period entropy can be used to gauge the cross-period variation in user’s edit activity throughout lifecycle periods. The cross-entropy of one probability distribution P (from a given lifecycle period) with respect to another distribution Q from an earlier period (e.g. the previous quarter) is defined as follows (Rowe, 2013):

$$C(P, Q) = - \sum_x p(x) \log q(x) \quad (30)$$

prediction. Specifically, we make predictions based on features generated from editor profile distributions in a sliding window with $w=4$ quarters. An editor is in the ‘departed’ class if she leaved the community before being active for less than $m=1$ quarter after the sliding window, denote the interval $[w, w+m]$ as the departed range. Similarly, an editor is in the ‘staying’ class if she was active in the community long enough for a relatively large $n \geq 3$ quarters after the sliding-window, term the interval $[w+n, +\infty]$ as the staying range.

6.2.1 FEATURES FOR THE TASK

Our features are generated based on the findings reported in the previous section. For simplicity, we assume the w quarters included in the t -th sliding-window being $t = [j, \dots, j+w-1]$ ($j \in [1, 0, 30]$), and denote the Probability Of Activity Profile of an editor in quarter j assigned to the k -th user role as $POAP_{t,j,k}$. We use the following features to characterise the patterns of change in editor profile distributions:

- *First active quarter*: the quarter in which an editor began edits in Wikipedia. The timestamp a user joined the community may affect her decision about whether to stay for longer.
- *Cumulative active quarters*: the total number of quarters an editor had been active in the community till the last quarter in the sliding window.
- *Fraction of active quarters in lifespan*: the proportion of quarters a user was active till the sliding window.
- *Fraction of active quarters in sliding window*: the fraction of quarters a user was active in current sliding window.
- *Similarity of profile distribution in sliding window*: quantifies the similarity of user profile distributions in any two successive quarters using cosine similarity.
- *Diversity of edit activity*: denotes the entropy of $POAP_{t,j,k}$ for each quarter j in window t . This measure captures the extent to which an editor diversified her edits toward multiple namespaces and categories of articles.
- *Cross-entropy of edit activity*: denotes the historical variation in $POAP_{t,j,k}$ compared to the same measure in previous quarters, calculated using Eq. (30). This measure captures the extent to which an editor changed her edit activity compared to her past behaviour.
- *mean POAP_{t,j,k}*: denotes the average of $POAP_{t,j,k}$ for each user role k in window t , and captures whether an editor focused her edits on certain namespaces and categories of articles in window t .
- $\Delta POAP_{t,j,k}$: denotes the change in $POAP_{t,j,k}$ between the quarter $j-1$ and j , measured by $-\Delta POAP_{t,j,k} = (POAP_{t,j,k} - POAP_{t,j-1,k} + \delta) / (POAP_{t,j-1,k} + \delta)$, where δ is a small positive real number (i.e. 0.001) to avoid the case when $POAP_{t,j-1,k}$ is 0. This measure also captures the fluctuation of $POAP_{t,j,k}$ for each user role k in window t .

For each editor, the first three features are global-level features which may be updated with the sliding window, the remaining features are window-level features and are recalculated within each sliding window. The intuition behind the last four features is to approximate the evolution of editor lifecycle we sought to characterise in the previous section. The dataset is of the following form: $D = (x_i, y_i)$, where y_i denotes the churn status of the editor, $y_i \in \{\text{Churner, Non-churner}\}$; x_i denotes the feature vector for the editor.

6.2.2 TASK PERFORMANCE

Table 5: Performance of sliding-window based churn prediction using features generated from different models. The measures are averaged over all sliding windows for each model.

Models	FP Rate	Precision	Recall	F-Measure	ROC Area
DTM	0.40±0.02	0.72±0.01	0.73±0.01	0.72±0.01	0.77±0.01
OTV-MFMM	0.47±0.03	0.69±0.01	0.71±0.01	0.69±0.01	0.72±0.02
NMF	0.45±0.02	0.70±0.01	0.71±0.01	0.69±0.01	0.73±0.02
BvMFMM	0.48±0.03	0.69±0.01	0.71±0.01	0.68±0.01	0.70±0.03
DP-GMM	0.45±0.04	0.70±0.01	0.71±0.01	0.69±0.02	0.72±0.04

Table 5 gives the averaged performance of sliding-window based churn prediction using features generated from different models. We observe that the best performance is obtained using features generated from DTM; churn prediction using features generated from the other four models presents similar performance. Notice that DTM and NMF generally generate a mixture of topics for user profiles, while the other three models tend to generate unique topics for user profiles. This suggests that different models may be applied to applications with different requirements. Alternatively, models such as DTM and NMF may be better at capturing change in user behaviour for churn prediction, while discriminative models such as DP-GMM and OTV-MFMM may be better at identifying user expertise for task recommendation. Our results suggest that sudden changes in user behaviour can be a signal that the user is likely to abandon the community, and that features inspired by topic models are useful for churn prediction.

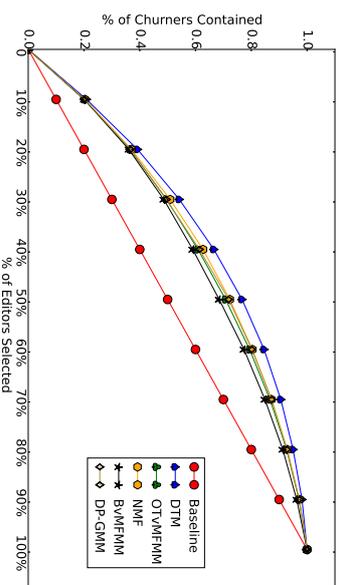


Figure 11: Lift chart of churn prediction using features generated from different models. The measures are averaged over all sliding windows.

Lift factors have been widely used by researchers to evaluate the performance of churn-prediction models (Weia and Chub, 2002). The lift factors achieved by different models are shown in Figure 11. In a lift chart, the diagonal line represents a baseline which randomly selects a subset of editors as potential churners, i.e., it selects $s\%$ of the editors that will contain $s\%$ of the true churners, resulting in a lift factor of 1. For instance, in Figure 11, on average, all models (except Baseline) was capable of identifying 10% of editors that contained at least 20% of true churners (i.e. a lift factor of 2.0). DTM was capable of identifying 20% of editors that contained 39.3% of true churners (i.e. a lift factor of 1.966), and 30% of editors that contained 53.9% of true churners (i.e. a lift factor of 1.799). Evidently, DTM achieved slightly higher lift factors than the other four models, all models achieved higher lift factors than the baseline. Thus if the objective of the lift analysis is to identify a small subset of likely churners for an intervention that might persuade them not to churn, then this analysis suggests that all models can identify a set of 10% of users where the probability of churning is more than twice the baseline figure.

7. Conclusion

This work proposed an online trans-dimensional von Mises-Fisher mixture model (OTvMFMM) for temporal user behavioural data, which (a) enables information sharing among clusters via a Bayesian framework, (b) allows adaptive change in the number of clusters by using our extended version of the reversible jump MCMC algorithm, and (c) accommodates the dynamics of clusters for time-varying user behavioural data based on the smoothness assumption. Our efficient collapsed Gibbs sampling algorithms make the models applicable to large-scale real-world data such as Wikipedia dataset. Empirical results on synthetic and real-world data show that the proposed models can discover more interpretable and intuitive clusters than other widely-used models, such as k-means, Non-negative Matrix Factorization (NMF), Dirichlet process Gaussian mixture models (DP-GMM), and dynamic topic models (DTM). We further evaluated the performance of proposed models in real-world applications, such as churn prediction task, that shows the usefulness of the features generated.

The results show that the proposed OTvMFMM can discover more interpretable and intuitive clusters for evolving user behavioural data than DP-GMM with sub-cluster split/merge moves by Chang and Fisher III (2013), whereas the latter is found to converge much faster than the former. An interesting and promising future direction is to replace the reversible jump MCMC algorithm (Richardson and Green, 1997) with the subcluster split-merge strategy (Chang and Fisher III, 2013) in order to allow adaptive change in the number of clusters for the Bayesian von Mises-Fisher mixture models. The new integration would lead to more efficient and interpretable non-parametric models for L_2 normalised data that combine the advantages of both OTvMFMM and DP-GMM. In addition, heterogeneous user behavioural data are ubiquitous in the sense of multiplicity of features and multiple data sources available. We would like to handle heterogeneous data and apply the models to analyse user behavioural data in other online communities.

Acknowledgements

This work was supported by Science Foundation Ireland (SFI) under Grant No. SFI/12/RC/2289 (Insight Centre for Data Analytics). Xiangju Qin was funded by University College Dublin and China Scholarship Council (UCD-CSC Joint Scholarship 2011). We thank the anonymous reviewers for their constructive suggestions.

References

- Amr Ahmed and Eric P. Xing. Dynamic Non-Parametric Mixture Models and the Recurrent Chinese Restaurant Process: with Applications to Evolutionary Clustering. In *Proc. of the Eighth SIAM International Conference on Data Mining (SDM)*, pages 219–230, 2008.
- Amr Ahmed and Eric P. Xing. Timeline: A Dynamic Hierarchical Dirichlet Process Model for Recovering Birth/Death and Evolution of Topics in Text Stream. In *Proc. of the 26th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 20–29, 2010.
- Amr Ahmed, Yucheng Low, Mohamed Aly, Vanja Josifovski, and Alexander J. Smola. Scalable distributed inference of dynamic user interests for behavioral targeting. In *Proc. of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 114–122, 2011.
- Christophe Andrieu, Freitas Nando de, Arnaud Doucet, and Michael I. Jordan. An Introduction to MCMC for Machine Learning. *Machine Learning*, 50(1–2):5–43, 2003.
- David Arthur and Sergei Vassilvitskii. k-means++: The Advantages of Careful Seeding. In *Proc. of the 18th Annual ACM-SIAM symposium on Discrete algorithms (SODA)*, pages 1027–1035, 2007.
- Arindam Banerjee, Inderjit S. Dhillon, Joydeep Ghosh, and Suvrit Sra. Clustering on the Unit Hypersphere using von Mises-Fisher Distributions. *Journal of Machine Learning Research (JMLR)*, 6(1):1345–1382, 2005.
- Mark Bangert, Philipp Hennig, and Uwe Oelfke. Using an Infinite Von Mises-Fisher Mixture Model to Cluster Treatment Beam Directions in External Radiation Therapy. In *Proc. of the Ninth IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 746–751, 2010.
- David M. Blei and John D. Lafferty. Dynamic Topic Models. In *Proc. of the 23rd International Conference on Machine Learning (ICML)*, pages 113–120, 2006.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research (JMLR)*, 3:993–1022, 2003.
- Nizar Bouguila and Tarek Elguebauly. A fully Bayesian model based on reversible jump MCMC and finite Beta mixtures for clustering. *Expert Systems with Applications (ESA)*, 39(5):5946–5959, 2012.
- Christos Boutsidis and Efstratios Gallopoulos. SVD based initialization: A head start for non-negative matrix factorization. *Pattern Recognition*, 2008.
- Stephen P. Brooks and Paolo Giudici. Markov Chain Monte Carlo Convergence Assessment via Two-Way Analysis of Variance. *Journal of Computational and Graphical Statistics*, 9(2):266–285, 2000.
- Richard A. Brualdi and Hans Schneider. Determinantal Identities: Gauss, Schur, Cauchy, Sylvester, Kronecker, Jacobi, Binet, Laplace, Muir, and Cayley. *Linear Algebra and its Applications*, 52(53(1)):769–791, 1983.

- Gilles Celeux, Merrilee Hurn, and Christian P. Robert. Computational and Inferential Difficulties with Mixture Posterior Distributions. *Journal of the American Statistical Association (ASA)*, 95(451):957–970, 2000.
- Gilles Celeux, Florence Forbes, Christian P. Robert, and D. Mike Titterton. Deviance Information Criteria for Missing Data Models. *Bayesian Analysis*, 1(4):651–674, 2006.
- Jeffrey Chan, Conor Hayes, and Elizabeth M. Daly. Decomposing Discussion Forums using User Roles. In *Proc. of the Fourth International Conference on Weblogs and Social Media (ICWSM)*, pages 215–218, 2010.
- Jason Chang and John W. Fisher III. Parallel Sampling of DP Mixture Models using Sub-Cluster Splits. In *Proc. of the Advances in Neural Information Processing Systems (NIPS)*, pages 620–628, 2013.
- Jonathan Chang, Jordan L. Boyd-Graber, Sean Gerrish, Chong Wang, and David M. Blei. Reading Tea Leaves: How Humans Interpret Topic Models. In *Proc. of the Advances in Neural Information Processing Systems (NIPS)*, pages 288–296, 2009.
- Alessandro Chiuso and Giorgio Pecci. Visual Tracking of Points as Estimation on the Unit Sphere. In *The confluence of vision and control*, pages 90–105. Springer Link, 1998.
- Christian Danescu-Niculescu-Mizil, Robert West, Dan Jurafsky, June Leskovec, and Christopher Potts. No Country for Old Members: User Lifecycle and Linguistic Change in Online Communities. In *Proc. of the 22nd International World Wide Web Conference (WWW)*, pages 307–318. Rio de Janeiro, Brazil, 2013.
- Petros Dellaportas and Ioulia Papageorgiou. Multivariate mixtures of normals with unknown number of components. *Statistics and Computing*, 16(1):57–68, 2006.
- Avinava Dubey, Ahmed Helmy, Sinead Williamson, and Eric P. Xing. A nonparametric mixture model for topic modeling over time. In *Proc. of the 13th SIAM International Conference on Data Mining (SDM)*, pages 530–538, 2013.
- Adalberto Furtado, Nazareno Andrade, Nigini Oliveira, and Francisco Brasileiro. Contributor Profiles, their Dynamics, and their Importance in Five Q&A Sites. In *Proc. of the 19th ACM Conference on Computer Supported Cooperative Work and Social Computing (CSCW)*, pages 1237–1252, 2013.
- Andrew Gelman and Donald B. Rubin. Inference from Iterative Simulation. Using Multiple Sequences. *Statistical Science*, 7(4):457–511, 1992.
- Samuel J. Gershman and David M. Blei. A tutorial on Bayesian nonparametric models. *Journal of Mathematical Psychology*, 56(6):1–12, 2012.
- Siddharth Gopal and Yiming Yang. Von Mises-Fisher Clustering Models. In *Proc. of the 31st International Conference on Machine Learning (ICML)*, pages 154–162, 2014.
- John Hartigan and Manchek Wong. Algorithm AS 136: A K-Means Clustering Algorithm. *Applied statistics*, 28(1):100–108, 1979.
- W. Keith Hastings. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*, 57(1):97–109, 1970.
- Gregor Heinrich. Parameter Estimation for Text Analysis. Technical report version 2.9. vsonix GmbH + University of Leipzig, 2009.
- Ioana Hulpus, Conor Hayes, Marcel Kamstedt, and Derek Greene. Unsupervised graph-based topic labelling using dbpedia. In *Proc. of the Sixth ACM International Conference on Web Search and Data Mining (WSDM)*, pages 465–474, 2013.
- John T. Kent. The Fisher-Bingham Distribution on the Sphere. *Journal of the Royal Statistical Society*, 44:71–80, 1982.
- Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.
- Jan R. Magnus and Heinz Neudecker. *Matrix Differential Calculus with Applications in Statistics and Economics*. John Wiley and Sons, NY, USA, 1988.
- Kanti V. Mardia and Peter E. Jupp. *Directional Statistics*. Academic Press Inc., London, UK, 2000.
- Geoffrey McLachlan and David Peel. *Finite Mixture Models*. John Wiley & Sons, 2000.
- Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.
- Agostino Nobile. Bayesian finite mixtures: a note on prior specification and posterior computation. Technical report, Department of Statistics, University of Glasgow, 2005.
- Derek O’Callaghan, Derek Greene, Joe Carthy, and Pádraig Cunningham. An analysis of the coherence of descriptors in topic modeling. *Expert Systems with Applications (ESA)*, 42(13):5645–5657, 2015.
- David Peel, William J. Whitten, and Geoffrey J. McLachlan. Fitting Mixtures of Kent Distributions to Aid in Joint Set Identification. *Journal of the American Statistical Association*, 96(453):56–63, 2001.
- Xiangju Qin, Derek Greene, and Pádraig Cunningham. A latent space analysis of editor lifecycles in wikipedia. In *Proc. of the 5th International Workshop on Mining Ubiquitous and Social Environments (MUSE) at ECML/PKDD 2014*, pages 3–18, 2014.
- Carl Edward Rasmussen. The Infinite Gaussian Mixture Model. *Proc. of the Advances in Neural Information Processing Systems (NIPS)*, 12:554–560, 2000.
- Joseph Reisinger, Austin Waters, Bryan Silvertorn, and Raymond J. Mooney. Spherical Topic Models. In *Proc. of the 27th International Conference on Machine Learning (ICML)*, pages 903–910, 2010.
- Sylvia Richardson and Peter J. Green. On Bayesian Analysis of Mixtures with an Unknown Number of Components. *Statistical Methodology*, 59(4):731–792, 1997.

Matthew Rowe. Mining User Lifecycles from Online Community Platforms and their Application to Churn Prediction. In *Proc. of the 13th IEEE International Conference on Data Mining (ICDM)*, pages 1–10, 2013.

Diane Mary Sculley. Web-Scale K-Means Clustering. In *Proc. of the 19th International World Wide Web Conference (WWW)*, pages 1177–1178, 2010.

Alexander Smola and Shrawan Narayanamurthy. An architecture for parallel topic models. *Proc. of the VLDB Endowment*, 3(1-2):703–710, 2010.

Michael Spivak. *Calculus on Manifolds A Modern Approach to Classical Theorems of Advanced Calculus*. Addison-Wesley Publishing Company, USA, 1965.

Julian Straub, Trevor Campbell, Jonathan P. How, and John W. Fisher III. Small-Variance Non-parametric Clustering on the Hypersphere. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 334–342, 2015a.

Julian Straub, Jason Chang, Oren Freifeld, and John W. Fisher III. A Dirichlet Process Mixture Model for Spherical Data. In *Proc. of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 930–938, 2015b.

Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association (JASA)*, 101(476):1566–1581, 2006.

Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research (JMLR)*, 9(Nov):2579–2605, 2008.

Yi Wang, Shi-Xia Liu, Lizhu Zhou, and Hui Su. Mining Naturally Smooth Evolution of Clusters from Dynamic Data. In *Proc. of the Seventh SIAM International Conference on Data Mining (SDM)*, pages 125–134, 2007.

Greg C. G. Wei and Martin A. Tanner. A Monte Carlo Implementation of the EM Algorithm and the Poor Man's Data Augmentation Algorithms. *Journal of the American Statistical Association*, 81(411):699–704, 1990.

Chih-Ping Weia and I-Tang Chiu. Turning telecommunications call details to churn prediction: a data mining approach. *Expert Systems with Applications (ESA)*, 23(2):103–112, 2002.

Zhihua Zhang, Kap Luk Chan, Yiming Wu, and Chibiao Chen. Learning a multivariate Gaussian mixture model with the reversible jump MCMC algorithm. *Statistics and Computing*, 14(4): 343–355, 2004.

Mingjun Zhong and Mark Girolami. Reversible Jump MCMC for Non-Negative Matrix Factorization. In *Proc. of the Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 663–670, 2009.

Shi Zhong and Joydeep Ghosh. Generative Model-based Document Clustering: A Comparative Study. *Knowledge and Information Systems (KAIS)*, 8(3):374–384, 2005.

Appendices

Appendix A Detailed Inference for Collapsed Gibbs Sampling

Updates for z_i . The conditional distribution for z_i is given by

$$\begin{aligned} \gamma(z_{ih}) &\equiv P(z_{ih} = i | \mathcal{Z}_{-i}, \{x_j\}_{j=1}^N, \kappa, m, \sigma^2, \mu_0, C_0, \alpha) \propto \\ &\int_{\pi} \int_{\mu} f(x_i | \mu_{z_i}, \kappa_{z_i}) P(z_i | \pi) \prod_{j \neq i} P(z_j | \pi) f(x_j | \mu_{z_j}, \kappa_{z_j}) \prod_{h=1}^H f(\mu_h | \mu_0, C_0) f(\kappa_h | m, \sigma^2) \\ &\propto \int_{\pi} P(z_i | \pi) f(\pi | \alpha) \prod_{j \neq i} P(z_j | \pi) \times \\ &\quad \prod_{h=1}^H \int_{\mu_h} f(x_i | \mu_{z_i}, \kappa_{z_i}) \prod_{j \neq i, z_j \neq h} f(x_j | \mu_h, \kappa_h) f(\mu_h | \mu_0, C_0) f(\kappa_h | m, \sigma^2) \\ &\propto \int_{\pi} P(z_i | \pi) f(\pi | \alpha) \prod_{j \neq i} P(z_j | \pi) \times \prod_{h=1}^H \int_{\mu_h} f(x_i | \mu_{z_i}, \kappa_{z_i}) \prod_{j \neq i, z_j \neq h} f(x_j | \mu_h, \kappa_h) f(\mu_h | \mu_0, C_0) \end{aligned} \quad (\text{A-1})$$

where the description of each term in Eq. (A-1) can be referred to Eq. (7). Expanding out the Dirichlet priors and the discrete distributions according to their usual definitions, i.e., $P(\pi | \alpha) \sim \text{Dirichlet}(H, \alpha)$, $P(z_i | \pi) \sim \text{Multi}(\cdot | \pi)$, yields¹⁸:

$$\begin{aligned} \int_{\pi} P(z_{ih} = 1 | \pi) f(\pi | \alpha) \prod_{j \neq i} P(z_j | \pi) &= \int_{\pi} \prod_{h=1}^H \pi_h^{z_{ih}} \prod_{h=1}^H \frac{\Gamma(H\alpha)}{\Gamma(\alpha)} \prod_{h=1}^H \prod_{j \neq i} \pi_h^{\alpha-1} \prod_{j \neq i, h=1}^H \pi_h^{z_{jh}} \\ &= \frac{\Gamma(H\alpha)}{\prod_{h=1}^H \Gamma(\alpha)} \int_{\pi} \pi_h^{\alpha + n_{h,-i} + z_{ih} - 1} \\ &= \frac{\Gamma(H\alpha)}{\prod_{h=1}^H \Gamma(\alpha)} \prod_{h=1}^H \frac{\Gamma(\alpha + n_{h,-i} + z_{ih})}{\Gamma(H\alpha + N)} \\ &\propto \Gamma(\alpha + n_{h,-i} + 1) \propto (\alpha + n_{h,-i}) \Gamma(\alpha + n_{h,-i}) \propto (\alpha + n_{h,-i}) \end{aligned} \quad (\text{A-2})$$

Similarly, expanding out the probabilities $f(x_i | \mu_{z_i}, \kappa_{z_i})$, $f(x_j | \mu_h, \kappa_h)$ and $f(\mu_h | \mu_0, C_0)$ according to their usual definitions, $f(x_i | \mu_{z_i}, \kappa_{z_i}) = C_D(\kappa_{z_i}) \exp(\kappa_{z_i} \mu_{z_i}^T x_i)$ and $f(\mu_h | \mu_0, C_0) = C_D(C_0) \exp(C_0 \mu_h^T \mu_h)$

¹⁸ In the calculation, following [Heinrich \(2009\)](#), the Dirichlet integral of the first kind for summation function, $\sum_{\pi, \pi_0=1} \frac{\Gamma(\alpha)^H}{\Gamma(H\alpha)} = \int_{\pi} \prod_{h=1}^H \pi_h^{\alpha-1}$ is used, analogous to the identity of the beta integral: $B(\alpha_1, \alpha_2) = \int_0^1 x^{\alpha_1-1} (1-x)^{\alpha_2-1} dx$. The identity $\Gamma(x+1) = x\Gamma(x)$ is used in the last line.

yields ¹⁹:

$$\begin{aligned}
 & \prod_{h=1}^H \int_{\mu_h} \left(f(x_i | \mu_{z_i}, \kappa_{z_i}) \prod_{j \neq i, z_j=h} f(x_j | \mu_h, \kappa_h) \right) f(\mu_h | \mu_0, C_0) = \\
 & = \prod_{h=1}^H \int_{\nu_{\mu_h}} C_D(\kappa_h)^{n_h + z_{2h}} C_D(C_0) \exp \left\{ C_0 \mu_0^T \mu_h + \kappa_h \mu_h^T \left(x_i + \sum_{j \neq i} z_{jh} x_j^T \right) \right\} \\
 & = \prod_{h=1}^H C_D(\kappa_h)^{n_h + z_{2h}} C_D(C_0) \int_{\mu_h} \exp \left\{ C_0 \mu_0^T + \kappa_h x_i^T + \kappa_h \sum_{j \neq i} z_{jh} x_j^T \right\} \mu_h \\
 & = \prod_{h=1}^H \frac{C_D(\kappa_h)^{n_h + z_{2h}} C_D(C_0)}{C_D(\|\kappa_h (x_i + \sum_{j \neq i} z_{jh} x_j^T) + C_0 \mu_0\|)}
 \end{aligned} \tag{A-3}$$

Substituting Eq. (A-2)-(A-3) in Eq. (A-1) yields the following:

$$\begin{aligned}
 P(z_{2h} = 1 | z_{-i}, \{x_j\}_{j=1}^N, \kappa, m, \sigma^2, \mu_0, C_0, \alpha) \\
 & \propto \int_{\pi} P(z_{2h} | \pi) f(\pi | \alpha) \prod_{j \neq i} P(z_j | \pi) \prod_{h=1}^H \int_{\mu_h} \left(f(x_i | \mu_{z_i}, \kappa_{z_i}) \prod_{j \neq i, z_j=h} f(x_j | \mu_h, \kappa_h) \right) f(\mu_h | \mu_0, C_0) \\
 & \propto (\alpha + n_{h,-}) \prod_{h=1}^H \frac{C_D(\kappa_h)^{n_{h,-} + z_{2h}} C_D(C_0)}{C_D(\|\kappa_h (x_i + \sum_{j \neq i} z_{jh} x_j^T) + C_0 \mu_0\|)} + C_0 \mu_0 \| \\
 & \propto (\alpha + n_{h,-}) C_D(\kappa_h) \frac{C_D(\|\kappa_h \sum_{j \neq i} z_{jh} x_j^T + C_0 \mu_0\|)}{C_D(\|\kappa_h (x_i + \sum_{j \neq i} z_{jh} x_j^T) + C_0 \mu_0\|)}
 \end{aligned} \tag{A-4}$$

Updates for κ . Similarly, the conditional distribution for κ_h is given by

$$\begin{aligned}
 f(\kappa_h | Z, \{x_j\}_{j=1}^N, \kappa, m, \sigma^2, \mu_0, C_0) & \propto \prod_{z_{2h}=1} \prod_{\mu_h} f(x_i | \mu_h, \kappa_h) f(\mu_h | \mu_0, C_0) f(\kappa_h | m, \sigma^2) \\
 & \propto \prod_{z_{2h}=1} \prod_{\mu_h} C_D(\kappa_h) C_D(C_0) \exp \left\{ \kappa_h \mu_h^T x_i + C_0 \mu_0^T \mu_h \right\} \log \text{Normal}(\kappa_h | m, \sigma^2) \\
 & \propto \int_{\mu_h} C_D(\kappa_h)^{n_h} C_D(C_0) \exp \left\{ \kappa_h \mu_h^T \sum_{z_{2h}=1} x_i + C_0 \mu_0^T \mu_h \right\} \log \text{Normal}(\kappa_h | m, \sigma^2) \\
 & \propto C_D(\kappa_h)^{n_h} C_D(C_0) \log \text{Normal}(\kappa_h | m, \sigma^2) \int_{\mu_h} \exp \left\{ \kappa_h \mu_h^T \sum_{z_{2h}=1} x_i + C_0 \mu_0^T \mu_h \right\} \\
 & \propto \frac{C_D(\kappa_h)^{n_h} C_D(C_0)}{C_D(\|\kappa_h \sum_{j: z_{jh}=1} z_{jh} x_j^T + C_0 \mu_0\|)} \log \text{Normal}(\kappa_h | m, \sigma^2)
 \end{aligned} \tag{A-5}$$

where n_h is the number of observations assigned to the h -th component.

¹⁹ In this calculation, the identity of the von Mises-Fisher integral (Mardia and Jupp (2000), page 168; Chiu and Pici (1998)) is used. $\int_{S^{k-1}} \exp \left\{ \kappa \mu^T x \right\} dx = (2\pi)^{\frac{k}{2}} \left(\frac{\kappa}{2}\right)^{1-D/2} I_{D/2-1}(\kappa) = \frac{\Gamma(D/2)}{\Gamma(D/2)}$, where $\kappa = \|\kappa\|$.

Appendix B Empirical Bayes Estimates for Prior Parameters

The joint likelihood function of the prior parameters $(\mu_0, C_0, m, \sigma^2, \alpha)$ is given by:

$$\begin{aligned}
 \mathcal{L}(\mu_0, C_0, m, \sigma^2, \alpha | \{x_i\}_{i=1}^N, \mu, \kappa, \pi) & = f(\{x_i\}_{i=1}^N | \mu, \kappa, \pi) f(\mu | \mu_0, C_0) f(\kappa | m, \sigma^2) f(\pi | \alpha) \\
 & \propto \prod_{h=1}^H C_D(C_0) \exp \left\{ C_0 \mu_0^T \mu_h \right\} \prod_{h=1}^H \frac{1}{\kappa_h \sigma \sqrt{2\pi}} \exp \left\{ -\frac{(\log \kappa_h - m)^2}{2\sigma^2} \right\} \frac{\Gamma(H\alpha)}{\prod_{h=1}^H \Gamma(\alpha)} \prod_{h=1}^H \pi_h^{\alpha-1}
 \end{aligned} \tag{B-1}$$

Since the prior parameters are assumed to be independent, we have the following log-likelihood functions according to Eq. (B-1)

$$\begin{aligned}
 \log \mathcal{L}(C_0, \mu_0 | \{x_i\}_{i=1}^N, \mu) & = H \log C_D(C_0) + C_0 \mu_0^T \left(\sum_{h=1}^H \mu_h \right) \\
 \log \mathcal{L}(m, \sigma^2 | \{x_i\}_{i=1}^N, \kappa) & = -\frac{H}{2} \log(\sigma^2) - \frac{1}{2\sigma^2} \sum_{h=1}^H (\log(\kappa_h)^2 - 2m \log(\kappa_h) + m^2) \\
 \log \mathcal{L}(\alpha | \{x_i\}_{i=1}^N, \pi) & = -\log \left(\frac{\Gamma(H\alpha)}{\prod_{h=1}^H \Gamma(\alpha)} \right) + (\alpha - 1) \prod_{h=1}^H \pi_h
 \end{aligned} \tag{B-2}$$

Following Gopal and Yang (2014), the empirical Bayes estimate for C_0, μ_0 is given by:

$$\arg \max_{\mu_0, C_0} H \log C_D(C_0) + C_0 \mu_0^T \left(\sum_{h=1}^H \mu_h \right) \tag{B-3}$$

which suggests the following updates (Gopal and Yang, 2014)

$$\mu_0 = \frac{\sum_{h=1}^H \mu_h}{\|\sum_{h=1}^H \mu_h\|}, \quad C_0 = \frac{\bar{r}D - \bar{r}^3}{1 - \bar{r}^2}, \quad \text{where, } \bar{r} = \frac{\|\sum_{h=1}^H \mu_h\|}{H} \tag{B-4}$$

Similarly, the empirical Bayes estimate for α is given by Gopal and Yang (2014):

$$\arg \max_{\alpha > 0} -\log \left(\frac{\Gamma(H\alpha)}{\prod_{h=1}^H \Gamma(\alpha)} \right) + (\alpha - 1) \sum_{h=1}^H \pi_h \tag{B-5}$$

Since there exists no closed-form solution for Eq. (B-5), we rely on numerical optimization such as gradient descent to find the Maximum Likelihood Estimate for α .

The empirical Bayes estimate for m, σ^2 can be obtained in a similar way by taking the partial derivative on Eq. (B-2) w. r. t. m and σ^2 , respectively, which gives:

$$\begin{aligned}
 \frac{\partial \log \mathcal{L}(m, \sigma^2 | X, \kappa)}{\partial m} & = -\frac{1}{2\sigma^2} \sum_{h=1}^H [-2 \log(\kappa_h) + 2m] = 0 \\
 \frac{\partial \log \mathcal{L}(m, \sigma^2 | X, \kappa)}{\partial \sigma^2} & = -\frac{H}{2\sigma^2} + \frac{1}{2\sigma^4} \sum_{h=1}^H (\log(\kappa_h)^2 - 2m \log(\kappa_h) + m^2) = 0 \quad (\text{Let } x = \sigma^2) \\
 \Rightarrow m & = \frac{1}{H} \sum_{h=1}^H \log(\kappa_h), \quad x = \sigma^2 = \frac{1}{H} \sum_{h=1}^H \log(\kappa_h)^2 - m^2
 \end{aligned} \tag{B-6}$$

Alternatively, we can use the Monte Carlo Expectation-Maximum (MCEM) algorithm (Wei and Tanner, 1990) to estimate the prior parameters m and σ^2 .

- E-step: Randomly sample L times the value for κ_i , $(\kappa_h^{(i)})_{h=1}^L, i \in [1, L]$. Note that previously, we use MCMC sampling to estimate κ_{iS} . Here, we can reuse the set of generated κ_{iS} in the MCMC step.

- M-step: Estimate m and σ^2 by maximising the log-likelihood:

$$m = \frac{1}{LH} \sum_{i=1}^L \sum_{h=1}^H \log(\kappa_h^{(i)}), \quad \sigma^2 = \frac{1}{LH} \sum_{i=1}^L \sum_{h=1}^H \log(\kappa_h^{(i)})^2 - m^2 \quad (\text{B-7})$$

Appendix C Factorization of Acceptance Probability

We can factorize $P(H, \mathcal{Z}, \{\pi_h, \theta_h\}_{h=1}^H, \mathcal{B}(\{x_i\}_{i=1}^N))$ below:

$$\begin{aligned} P(H, \mathcal{Z}, \{\pi_h, \theta_h\}_{h=1}^H, \mathcal{B}(\{x_i\}_{i=1}^N)) &= \frac{P(H, \mathcal{Z}, \pi, \mu, \kappa, m, \sigma^2, \mu_0, C_0, \alpha, \{x_i\}_{i=1}^N)}{P(\{x_i\}_{i=1}^N)} \\ &= \frac{P(H) f(\pi(\alpha)) P(\mathcal{Z}|\pi) f(\mu|C_0, \mu_0) f(\kappa|m, \sigma^2) f(\{x_i\}_{i=1}^N | \mathcal{Z}, \mu, \kappa)}{P(\{x_i\}_{i=1}^N)} \end{aligned} \quad (\text{C-1})$$

where each term in Eq. (C-2) can be specified as follows:

$$\begin{aligned} \frac{P(H+1, \mathcal{Z}', \Theta', \mathcal{B}(\{x_i\}_{i=1}^N))}{P(H, \mathcal{Z}, \Theta, \mathcal{B}(\{x_i\}_{i=1}^N))} &= \frac{P(H+1)}{P(H)} \times \frac{f(\pi'|\alpha)}{f(\pi|\alpha)} \times \frac{P(\mathcal{Z}'|\pi')}{P(\mathcal{Z}|\pi)} \times \frac{f(\mu'|\mu_0, C_0)}{f(\mu|\mu_0, C_0)} \\ &= \frac{f(\kappa'|m, \sigma^2)}{f(\kappa|m, \sigma^2)} \times \frac{f(\{x_i\}_{i=1}^N | \mathcal{Z}', \mu', \kappa')}{f(\{x_i\}_{i=1}^N | \mathcal{Z}, \mu, \kappa)} \end{aligned} \quad (\text{C-2})$$

$$\begin{aligned} \frac{P(H+1)}{P(H)} &= \frac{f(H+1; 1)}{f(H; 1)}, \quad \frac{P(\{z_i'\}_{i=1}^N | \pi')}{P(\{z_i\}_{i=1}^N | \pi)} = \frac{\pi_{j_1}^{n_{j_1}} \pi_{j_2}^{n_{j_2}}}{\pi_{j_1}^{n_{j_1}} \pi_{j_2}^{n_{j_2}}} \\ \frac{f(\pi'|\alpha)}{f(\pi|\alpha)} &= \frac{\Gamma((H+1)\alpha)}{\Gamma(\alpha)^{H+1}} \prod_{h=1}^{H+1} \pi_h^{\alpha-1} = \frac{1}{\pi_h^{\alpha-1}} \frac{\pi_h^{\alpha-1}}{\pi_{j_1}^{\alpha-1}} \\ \frac{f(\mu'|C_0, \mu_0)}{f(\mu|C_0, \mu_0)} &= (H+1) \frac{f(\mu_{j_1}|C_0, \mu_0)}{f(\mu_{j_1}|C_0, \mu_0)} \\ \frac{f(\kappa'|m, \sigma^2)}{f(\kappa|m, \sigma^2)} &= \frac{f(\kappa_{j_1}|m, \sigma^2) f(\kappa_{j_2}|m, \sigma^2)}{f(\kappa_{j_1}|m, \sigma^2)} \\ \frac{f(\{x_i\}_{i=1}^N | \{z_i'\}_{i=1}^N, \mu', \kappa')}{f(\{x_i\}_{i=1}^N | \{z_i\}_{i=1}^N, \mu, \kappa)} &= (\text{likelihood ratio}) = \frac{\prod_{i=1}^N f(x_i | \mu_{z_i}', \kappa_{z_i}')}{\prod_{i=1}^N f(x_i | \mu_{z_i}, \kappa_{z_i})} \end{aligned} \quad (\text{C-3})$$

$$\text{where } n_{j_1} = \sum_{i=1}^N z_{ij_1}, n_{j_2} = \sum_{i=1}^N z_{ij_2}, n_{j_3} = n_{j_1} + n_{j_2}$$

where $B(\cdot, \cdot)$ is the Beta function, the $(H+1)$ -factor in the third line being the ratio $(H+1)!/H!$ from the order statistics densities for the parameters (π, μ, κ) (i.e. label switching for the parameters). The calculation of the Jacobian matrix J and its determinant is similar to that in Zhang et al. (2004) and given in Appendix D.

Following Dellaportas and Papageorgiou (2006), the Jacobian term $\left| \frac{\partial \Sigma}{\partial(\lambda, V)} \right|$ can be computed by using the following formulae

$$\partial \lambda = V_d' (\partial \Sigma) V_d, \quad \partial V = (\lambda_d I_D - \Sigma)^+ (\partial \Sigma) V_d \quad (\text{C-4})$$

where λ_d and V_d , $d=1, \dots, D$, are the specific eigenvalue-eigenvector pairs of Σ ; $(A)^+$ denotes the Moore-Penrose pseudo-inverse matrix of A (See Magnus and Neudecker (1988) p.179). For symmetric perturbations, Magnus and Neudecker (1988) (p.181) suggested that applying the properties of vec operator (i.e. $\text{vec } ABC = (C \otimes A) \text{vec } B$) and the chain rule, Eq. (C-4) can be rewritten as follows

$$\partial \lambda = (V_d' \otimes V_d') \mathbf{D} \partial \text{vec}(\Sigma), \quad \partial V = (V_d' \otimes (\lambda_d I_D - \Sigma)^+) \mathbf{D} \partial \text{vec}(\Sigma) \quad (\text{C-5})$$

where \mathbf{D} is the duplication matrix (see Magnus and Neudecker (1988) Chapter 3). From Eq. (C-5), we obtain the derivatives

$$\begin{aligned} \frac{\partial \lambda}{\partial(\text{vec } \Sigma)} &= V_d' \otimes V_d' \\ \frac{\partial V}{\partial(\text{vec } \Sigma)} &= V_d' \otimes (\lambda_d I_D - \Sigma)^+ \end{aligned} \quad (\text{C-6})$$

According to the inverse function theorem (Spivak, 1965), the inverse of the Jacobian matrix of an invertible function is equivalent to the Jacobian matrix of the inverse function. Specifically, if the Jacobian of the function $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$ is continuous and non-singular at the point $p \in \mathbb{R}^n$, then F is invertible in some neighbourhood of p and we have

$$J_{F^{-1}}(F(p)) = (J_F(p))^{-1} \quad (\text{C-7})$$

Thus, if $\left| \frac{\partial(\lambda, V)}{\partial(\text{vec } \Sigma)} \right|$ is non-zero, then we can have

$$\left| \frac{\partial \Sigma}{\partial(\lambda, V)} \right| = \left| \frac{\partial(\lambda, V)}{\partial(\text{vec } \Sigma)} \right|^{-1} \quad (\text{C-8})$$

Appendix D Calculation of Jacobian Matrix

Let $s = \{\pi_{j_1}, \mu_{j_1}, g_{j_1}\}$ and $s' = \{\pi_{j_2}, \mu_{j_2}, g_{j_2}\}$ denote the state of Markov chain before and after the split move, respectively, where $g_{j_1} = (\lambda_{j_1}, \dots, \lambda_{j_1 D})^T$, $g_{j_2} = (\lambda_{j_2}, \dots, \lambda_{j_2 D})^T$, $g_{j_3} = (\lambda_{j_3}, \dots, \lambda_{j_3 D})^T$. Denote the set of continuous random variables needed for the split move as $u = \{u_1, u_2, u_3\}$, where $u_2 = (u_{21}, \dots, u_{2D})^T$, $u_3 = (u_{31}, \dots, u_{3D})^T$. Thus, from the transformation defined by Eq. (17), we can obtain the Jacobian matrix J for the split move (from (s, u) to (s') as follows (Zhang et al., 2004)

$$J = \frac{\partial s'}{\partial(s, u)} = \begin{bmatrix} \frac{\partial \pi_{j_1}}{\partial \pi_{j_1}} & \frac{\partial \pi_{j_1}}{\partial \pi_{j_2}} & \frac{\partial \pi_{j_1}}{\partial \pi_{j_3}} & \frac{\partial \pi_{j_1}}{\partial \mu_{j_1}} & \frac{\partial \pi_{j_1}}{\partial \mu_{j_2}} & \frac{\partial \pi_{j_1}}{\partial \mu_{j_3}} & \frac{\partial \pi_{j_1}}{\partial g_{j_1}} & \frac{\partial \pi_{j_1}}{\partial g_{j_2}} & \frac{\partial \pi_{j_1}}{\partial g_{j_3}} \\ \frac{\partial \pi_{j_2}}{\partial \pi_{j_1}} & \frac{\partial \pi_{j_2}}{\partial \pi_{j_2}} & \frac{\partial \pi_{j_2}}{\partial \pi_{j_3}} & \frac{\partial \pi_{j_2}}{\partial \mu_{j_1}} & \frac{\partial \pi_{j_2}}{\partial \mu_{j_2}} & \frac{\partial \pi_{j_2}}{\partial \mu_{j_3}} & \frac{\partial \pi_{j_2}}{\partial g_{j_1}} & \frac{\partial \pi_{j_2}}{\partial g_{j_2}} & \frac{\partial \pi_{j_2}}{\partial g_{j_3}} \\ \frac{\partial \pi_{j_3}}{\partial \pi_{j_1}} & \frac{\partial \pi_{j_3}}{\partial \pi_{j_2}} & \frac{\partial \pi_{j_3}}{\partial \pi_{j_3}} & \frac{\partial \pi_{j_3}}{\partial \mu_{j_1}} & \frac{\partial \pi_{j_3}}{\partial \mu_{j_2}} & \frac{\partial \pi_{j_3}}{\partial \mu_{j_3}} & \frac{\partial \pi_{j_3}}{\partial g_{j_1}} & \frac{\partial \pi_{j_3}}{\partial g_{j_2}} & \frac{\partial \pi_{j_3}}{\partial g_{j_3}} \\ \frac{\partial \mu_{j_1}}{\partial \pi_{j_1}} & \frac{\partial \mu_{j_1}}{\partial \pi_{j_2}} & \frac{\partial \mu_{j_1}}{\partial \pi_{j_3}} & \frac{\partial \mu_{j_1}}{\partial \mu_{j_1}} & \frac{\partial \mu_{j_1}}{\partial \mu_{j_2}} & \frac{\partial \mu_{j_1}}{\partial \mu_{j_3}} & \frac{\partial \mu_{j_1}}{\partial g_{j_1}} & \frac{\partial \mu_{j_1}}{\partial g_{j_2}} & \frac{\partial \mu_{j_1}}{\partial g_{j_3}} \\ \frac{\partial \mu_{j_2}}{\partial \pi_{j_1}} & \frac{\partial \mu_{j_2}}{\partial \pi_{j_2}} & \frac{\partial \mu_{j_2}}{\partial \pi_{j_3}} & \frac{\partial \mu_{j_2}}{\partial \mu_{j_1}} & \frac{\partial \mu_{j_2}}{\partial \mu_{j_2}} & \frac{\partial \mu_{j_2}}{\partial \mu_{j_3}} & \frac{\partial \mu_{j_2}}{\partial g_{j_1}} & \frac{\partial \mu_{j_2}}{\partial g_{j_2}} & \frac{\partial \mu_{j_2}}{\partial g_{j_3}} \\ \frac{\partial \mu_{j_3}}{\partial \pi_{j_1}} & \frac{\partial \mu_{j_3}}{\partial \pi_{j_2}} & \frac{\partial \mu_{j_3}}{\partial \pi_{j_3}} & \frac{\partial \mu_{j_3}}{\partial \mu_{j_1}} & \frac{\partial \mu_{j_3}}{\partial \mu_{j_2}} & \frac{\partial \mu_{j_3}}{\partial \mu_{j_3}} & \frac{\partial \mu_{j_3}}{\partial g_{j_1}} & \frac{\partial \mu_{j_3}}{\partial g_{j_2}} & \frac{\partial \mu_{j_3}}{\partial g_{j_3}} \\ \frac{\partial g_{j_1}}{\partial \pi_{j_1}} & \frac{\partial g_{j_1}}{\partial \pi_{j_2}} & \frac{\partial g_{j_1}}{\partial \pi_{j_3}} & \frac{\partial g_{j_1}}{\partial \mu_{j_1}} & \frac{\partial g_{j_1}}{\partial \mu_{j_2}} & \frac{\partial g_{j_1}}{\partial \mu_{j_3}} & \frac{\partial g_{j_1}}{\partial g_{j_1}} & \frac{\partial g_{j_1}}{\partial g_{j_2}} & \frac{\partial g_{j_1}}{\partial g_{j_3}} \\ \frac{\partial g_{j_2}}{\partial \pi_{j_1}} & \frac{\partial g_{j_2}}{\partial \pi_{j_2}} & \frac{\partial g_{j_2}}{\partial \pi_{j_3}} & \frac{\partial g_{j_2}}{\partial \mu_{j_1}} & \frac{\partial g_{j_2}}{\partial \mu_{j_2}} & \frac{\partial g_{j_2}}{\partial \mu_{j_3}} & \frac{\partial g_{j_2}}{\partial g_{j_1}} & \frac{\partial g_{j_2}}{\partial g_{j_2}} & \frac{\partial g_{j_2}}{\partial g_{j_3}} \\ \frac{\partial g_{j_3}}{\partial \pi_{j_1}} & \frac{\partial g_{j_3}}{\partial \pi_{j_2}} & \frac{\partial g_{j_3}}{\partial \pi_{j_3}} & \frac{\partial g_{j_3}}{\partial \mu_{j_1}} & \frac{\partial g_{j_3}}{\partial \mu_{j_2}} & \frac{\partial g_{j_3}}{\partial \mu_{j_3}} & \frac{\partial g_{j_3}}{\partial g_{j_1}} & \frac{\partial g_{j_3}}{\partial g_{j_2}} & \frac{\partial g_{j_3}}{\partial g_{j_3}} \end{bmatrix} \quad (\text{D-1})$$

From the transformation in Eq. (17), we calculate the partial derivatives:

$$\begin{aligned}
 \frac{\partial \pi_{1i}}{\partial \pi_{1j}} &= u_1, & \frac{\partial \pi_{1i}}{\partial u_1} &= \pi_{1j}, & \frac{\partial \pi_{1i}}{\partial \pi_{1j}} &= 0_{1 \times D}, & \frac{\partial \pi_{1i}}{\partial u_2} &= 0_{1 \times D} \\
 \frac{\partial \pi_{1i}}{\partial \pi_{1j}} &= 0_{1 \times D}, & \frac{\partial \pi_{1i}}{\partial u_2} &= 0_{1 \times D} \\
 \frac{\partial \pi_{2i}}{\partial \pi_{1j}} &= 1 - u_1, & \frac{\partial \pi_{2i}}{\partial u_1} &= -\pi_{1j}, & \frac{\partial \pi_{2i}}{\partial \pi_{1j}} &= 0_{1 \times D}, & \frac{\partial \pi_{2i}}{\partial u_2} &= 0_{1 \times D} \\
 \frac{\partial \pi_{2i}}{\partial \pi_{1j}} &= 0_{1 \times D}, & \frac{\partial \pi_{2i}}{\partial u_1} &= 0_{1 \times D} \\
 \frac{\partial \pi_{3i}}{\partial \pi_{1j}} &= 0_{D \times 1}, & \frac{\partial \pi_{3i}}{\partial u_1} &= 0_{D \times 1}, & \frac{\partial \pi_{3i}}{\partial \pi_{1j}} &= I, & \frac{\partial \pi_{3i}}{\partial u_2} &= 0_{D \times D} \\
 \frac{\partial \pi_{3i}}{\partial \pi_{1j}} &= 0_{D \times 1}, & \frac{\partial \pi_{3i}}{\partial u_1} &= 0_{D \times 1}, & \frac{\partial \pi_{3i}}{\partial \pi_{1j}} &= I, & \frac{\partial \pi_{3i}}{\partial u_2} &= 0_{D \times D} \\
 \frac{\partial \pi_{3i}}{\partial \pi_{1j}} &= 0_{D \times 1}, & \frac{\partial \pi_{3i}}{\partial u_1} &= 0_{D \times 1}, & \frac{\partial \pi_{3i}}{\partial \pi_{1j}} &= 0_{D \times D}, & \frac{\partial \pi_{3i}}{\partial u_2} &= 0_{D \times D} \\
 \frac{\partial \pi_{3i}}{\partial \pi_{1j}} &= 0_{D \times 1}, & \frac{\partial \pi_{3i}}{\partial u_1} &= 0_{D \times 1}, & \frac{\partial \pi_{3i}}{\partial \pi_{1j}} &= 0_{D \times D}, & \frac{\partial \pi_{3i}}{\partial u_2} &= 0_{D \times D}
 \end{aligned} \tag{D-2}$$

The other partial derivatives can be calculated as:

$$\begin{aligned}
 \frac{\partial \mu_{1i}}{\partial u_{2d}} &= -\sqrt{\frac{\pi_{1i}}{\pi_{1j}}} \lambda^{\frac{1}{2}} V_{j,d} \\
 \frac{\partial \mu_{1i}}{\partial u_{2d}} &= \sqrt{\frac{\pi_{1i}}{\pi_{1j}}} \lambda^{\frac{1}{2}} V_{j,d} \\
 \frac{\partial \lambda_{j,d}}{\partial u_{2i}} &= \begin{cases} -2u_{3d}u_{2d}\lambda_{j,d}\frac{\pi_{1i}}{\pi_{1j}} & l = d, \\ 0 & l \neq d \end{cases} \\
 \frac{\partial \lambda_{j,d}}{\partial u_{2i}} &= \begin{cases} -2(1 - u_{3d})u_{2d}\lambda_{j,d}\frac{\pi_{1i}}{\pi_{1j}} & l = d, \\ 0 & l \neq d \end{cases} \\
 \frac{\partial \lambda_{j,d}}{\partial u_{3i}} &= \begin{cases} u_{3d}(1 - u_{2d})\frac{\pi_{1i}}{\pi_{1j}} & l = d, \\ 0 & l \neq d \end{cases} \\
 \frac{\partial \lambda_{j,d}}{\partial \lambda_{j,i}} &= \begin{cases} (1 - u_{3d})(1 - u_{2d})\frac{\pi_{1i}}{\pi_{1j}} & l = d, \\ 0 & l \neq d \end{cases} \\
 \frac{\partial \lambda_{j,d}}{\partial \lambda_{j,i}} &= \begin{cases} \lambda_{j,d}(1 - u_{2d})\frac{\pi_{1i}}{\pi_{1j}} & l = d, \\ 0 & l \neq d \end{cases} \\
 \frac{\partial \lambda_{j,d}}{\partial u_{3i}} &= \begin{cases} -\lambda_{j,d}(1 - u_{2d})\frac{\pi_{1i}}{\pi_{1j}} & l = d, \\ 0 & l \neq d, \quad d \in [1, D] \end{cases}
 \end{aligned} \tag{D-3}$$

Therefore, we have the following expressions

$$\begin{aligned}
 \frac{\partial \mu_{1i}}{\partial u_2} &= -\sqrt{\frac{\pi_{1i}}{\pi_{1j}}} V_{j,i} \Lambda_{j,i}^{-\frac{1}{2}} \\
 \frac{\partial \mu_{1i}}{\partial u_2} &= \sqrt{\frac{\pi_{1i}}{\pi_{1j}}} V_{j,i} \Lambda_{j,i}^{-\frac{1}{2}} \\
 \frac{\partial \mu_{1i}}{\partial u_2} &= -2\frac{\pi_{1i}}{\pi_{1j}} \Lambda_{j,i} U_3 U_2 \\
 \frac{\partial \mu_{1i}}{\partial u_2} &= 2\frac{\pi_{1i}}{\pi_{1j}} \Lambda_{j,i} (U_3 - I) U_2 \\
 \frac{\partial \mu_{1i}}{\partial \pi_{1j}} &= -\frac{1}{2} \sqrt{\frac{\pi_{1i}}{\pi_{1j}}} V_{j,i} \Lambda_{j,i}^{-\frac{1}{2}} U_2 \\
 \frac{\partial \mu_{1i}}{\partial \pi_{1j}} &= \frac{1}{2} \sqrt{\frac{\pi_{1i}}{\pi_{1j}}} V_{j,i} \Lambda_{j,i}^{-\frac{1}{2}} U_2 \\
 \frac{\partial \mu_{1i}}{\partial \pi_{1j}} &= \frac{\pi_{1i}}{\pi_{1j}} U_3 (1 - U_2^2) \\
 \frac{\partial \mu_{1i}}{\partial \pi_{1j}} &= \frac{\pi_{1i}}{\pi_{1j}} (I - U_3)(I - U_2^2) \\
 \frac{\partial \mu_{1i}}{\partial \pi_{1j}} &= \frac{\pi_{1i}}{\pi_{1j}} \Lambda_{j,i} (I - U_2^2) \\
 \frac{\partial \mu_{1i}}{\partial \pi_{1j}} &= \frac{\pi_{1i}}{\pi_{1j}} \Lambda_{j,i} (U_2^2 - I)
 \end{aligned} \tag{D-4}$$

where $U_2 = \text{diag}(u_{21}, \dots, u_{2D})$ and $U_3 = \text{diag}(u_{31}, \dots, u_{3D})$ are diagonal matrices. Substituting Eq. (D-2) and (D-4) into Eq. (D-1) yields the Jacobian matrix, J , as follows:

$$J = \begin{bmatrix} u_1 & \pi_{1i} & \mathbf{0}_{1 \times D} & \mathbf{0}_{1 \times D} & \mathbf{0}_{1 \times D} & \mathbf{0}_{1 \times D} \\ 1 - u_1 & -\pi_{1i} & \mathbf{0}_{1 \times D} & \mathbf{0}_{1 \times D} & \mathbf{0}_{1 \times D} & \mathbf{0}_{1 \times D} \\ \mathbf{0}_{D \times 1} & \mathbf{0}_{D \times 1} & \mathbf{I} & -\sqrt{\frac{\pi_{1i}}{\pi_{1j}}} V_{j,i} \Lambda_{j,i}^{-\frac{1}{2}} & -\frac{1}{2} \sqrt{\frac{\pi_{1i}}{\pi_{1j}}} V_{j,i} \Lambda_{j,i}^{-\frac{1}{2}} U_2 & \mathbf{0}_{D \times D} \\ \mathbf{0}_{D \times 1} & \mathbf{0}_{D \times 1} & \mathbf{I} & \sqrt{\frac{\pi_{1i}}{\pi_{1j}}} V_{j,i} \Lambda_{j,i}^{-\frac{1}{2}} & \frac{1}{2} \sqrt{\frac{\pi_{1i}}{\pi_{1j}}} V_{j,i} \Lambda_{j,i}^{-\frac{1}{2}} U_2 & \mathbf{0}_{D \times D} \\ \mathbf{0}_{D \times 1} & \mathbf{0}_{D \times 1} & \mathbf{0}_{D \times D} & -2\frac{\pi_{1i}}{\pi_{1j}} \Lambda_{j,i} U_3 U_2 & \frac{\pi_{1i}}{\pi_{1j}} U_3 (I - U_2^2) & \frac{\pi_{1i}}{\pi_{1j}} \Lambda_{j,i} (I - U_2^2) \\ \mathbf{0}_{D \times 1} & \mathbf{0}_{D \times 1} & \mathbf{0}_{D \times D} & 2\frac{\pi_{1i}}{\pi_{1j}} \Lambda_{j,i} (U_3 - I) U_2 & \frac{\pi_{1i}}{\pi_{1j}} (I - U_3)(I - U_2^2) & \frac{\pi_{1i}}{\pi_{1j}} \Lambda_{j,i} (U_2^2 - I) \end{bmatrix} \tag{D-5}$$

where $\mathbf{0}_{D \times 1}$ is the $D \times 1$ zero vector, $\mathbf{0}_{1 \times D}$ is the $1 \times D$ zero vector, $\mathbf{0}_{D \times D}$ the $D \times D$ zero matrix, $U_2 = \text{diag}(u_{21}, u_{22}, \dots, u_{2D})$ and $U_3 = \text{diag}(u_{31}, u_{32}, \dots, u_{3D})$ are diagonal matrices.

By blocking the Jacobian matrix J defined by Eq. (D-5), we have

$$|\det(J)| = \pi_{1i} \cdot |\det(J_1)| \tag{D-6}$$

where

$$J_1 = \begin{bmatrix} \mathbf{I} & -\sqrt{\frac{\pi_{1i}}{\pi_{1j}}} V_{j,i} \Lambda_{j,i}^{-\frac{1}{2}} & \mathbf{0}_{D \times D} \\ \mathbf{I} & \sqrt{\frac{\pi_{1i}}{\pi_{1j}}} V_{j,i} \Lambda_{j,i}^{-\frac{1}{2}} & \mathbf{0}_{D \times D} \\ \mathbf{0}_{D \times D} & -2\frac{\pi_{1i}}{\pi_{1j}} \Lambda_{j,i} U_3 U_2 & \frac{\pi_{1i}}{\pi_{1j}} U_3 (I - U_2^2) \\ \mathbf{0}_{D \times D} & 2\frac{\pi_{1i}}{\pi_{1j}} \Lambda_{j,i} (U_3 - I) U_2 & \frac{\pi_{1i}}{\pi_{1j}} (I - U_3)(I - U_2^2) \\ & & \frac{\pi_{1i}}{\pi_{1j}} \Lambda_{j,i} (U_2^2 - I) \end{bmatrix} \tag{D-7}$$

We partitioned J_1 into $J_1 = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix}$ as indicated by the vertical and horizontal lines in Eq. (D-7). When J_{11} is invertible, according to Theorem by [Brunaldi and Schneider \(1983\)](#), we have

$$|\det(J_1)| = |\det(J_{11})| \cdot |\det(J_{22} - J_{21} J_{11}^{-1} J_{12})| \tag{D-8}$$

Calculating each determinant in Eq. (D-8) and substituting them into Eq. (D-6) yields the absolute of determinant of J , $\det(J)$, as follows

$$|\det(J)| = \frac{\pi_{1i}^{3D+1}}{(\pi_{1j} \pi_{1i})^{\frac{3D}{2}}} \prod_{d=1}^D \lambda_{j,d}^{1/2} (1 - u_{2d}^2) \tag{D-9}$$

Appendix E: Comparison of Different Split-Merge Moves

In this appendix, we compared the performance of TvMF mixture model with common eigenvectors (i.e. simplified RIMCMC move) and without common eigenvectors (i.e. original RIMCMC move) for split-merge moves on synthetic data from three aspects: (1) clustering performance, (2) acceptance rate for the moves and posterior estimation for the number of components H , and (3) the trace plot of the log likelihood and number of components over sweeps (in Figure 12). Table 6 compares the clustering performance and posterior estimation of H for TvMFMM with two different strategies for split-merge moves.

Table 6: Clustering performance and posterior estimation of H for TvMFMM with/without common eigenvectors for split-merge moves on synthetic data.

True H	ARI	AMI	NMI	Homogeneity	Completeness
TvMFMM with common eigenvectors for split-merge move					
4	0.925	0.898	0.899	0.898	0.899
5	0.947	0.941	0.945	0.947	0.944
7	0.809	0.821	0.829	0.827	0.832
TvMFMM without common eigenvectors for split-merge move					
4	0.931	0.904	0.904	0.904	0.904
5	0.936	0.931	0.935	0.936	0.934
7	0.792	0.812	0.82	0.814	0.826

True H	Portion of moves accepted (%)		Posterior estimation for H	
	Split-merge	Birth-death	Split-merge	Birth-death
TvMFMM with common eigenvectors for split-merge move				
4	0.89	0.15	$P(4)=0.997$, $P(5)=0.002$, $P(6)=0.001$	
5	0.25	0.16	$P(5)=0.999$, $P(6)=0.001$	
7	0.69	0.3	$P(7)=0.977$, $P(8)=0.020$, $P(9)=0.002$, $P(10)=0.001$	
TvMFMM without common eigenvectors for split-merge move				
4	0.08	0.05	$P(4)=1.000$	
5	0.02	0.1	$P(5)=0.999$, $P(6)=0.001$	
7	0.42	0.11	$P(7)=0.855$, $P(8)=0.143$, $P(9)=0.001$	

We observe from Table 6 that TvMFMM with and without common eigenvectors for split-merge moves present very similar clustering performance on the three datasets; TvMFMM with common eigenvectors for split-merge move achieves relatively better clustering performance. Both versions of TvMFMM give an accurate estimation of the number of components with the highest posterior probability for the true value of H . In addition, TvMFMM with both settings show very low acceptance rate for the split-merge and birth-death moves. This is one of the characteristics of the RJMCMC algorithm, which generally has very low acceptance rate for the moves, as observed by other researchers (Richardson and Green, 1997; Zhang et al., 2004; Dellaportas and Papageorgiou, 2006).

Figure 12 shows that TvMFMM with/without common eigenvectors for split-merge moves present similar mixing properties for the RJMCMC chains in terms of the smoothness of the log likelihood and number of components over sweeps. It is obvious that the RJMCMC chain becomes stable gradually in terms of the number of components and log likelihood after 5000 sweeps. We actually choose to use a burn-in period of 10000 sweeps for comparison of clustering performance on synthetic data. Overall, the results suggest that TvMFMM with/without common eigenvectors for the split-merge moves show similar performance and mixing rates on synthetic data. Therefore, for the results presented in the empirical evaluation, we ran TvMFMM and OTvMFMM with common eigenvectors for split-merge moves.

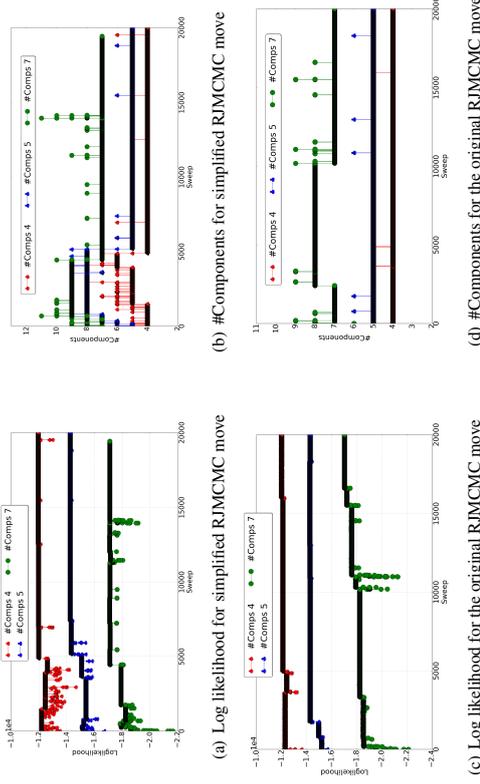


Figure 12: Trace plot of the log likelihood and estimated number of components for TvMFMM with simplified RJMCMC move and with original RJMCMC move on synthetic data.

Appendix F. Model Parameter Analysis

To choose the optimal number of clusters for parametric models, we used NMF model with different number of clusters to analyse Wikipedia quarterly datasets, and measured cluster coherence using normalised pairwise mutual information (NPMI). The results are presented in Figure 13. Recall that the NPMI metric captures the semantic interpretability of discovered clusters based on the corresponding descriptor terms. Higher coherence scores indicate better semantic interpretability, thus more coherent and interpretable topics. One obvious trend in Figure 13 is that NMF models running with larger number of clusters result in lower values of NPMI scores, indicating less coherent and interpretable clusters; whereas models running with smaller number of clusters have higher values of NPMI scores, suggesting more coherent and interpretable clusters. Moreover, NMF models running with 5, 10 and 15 clusters generate very close NPMI scores on all the quarterly datasets, of which models with 10 clusters output the best overall NPMI scores. Therefore, we choose $H = 10$ for parametric models.

We perform further analysis of the log likelihood and cluster coherence of Bayesian vMF mixture model on selected Wikipedia quarterly datasets for varying number of clusters, which are given in Figure 14. As the number of clusters increase from 5 to 20, the log likelihood of the model on the training and held-out datasets also increases, after which the log likelihood becomes relatively stable for any increase in #clusters. On the other hand, as #clusters increases from 5 to 10, the NPMI scores also go up, after which the NPMI scores reduce slightly and become relatively stable. This suggests that larger numbers of clusters does not indicate improved cluster coherence

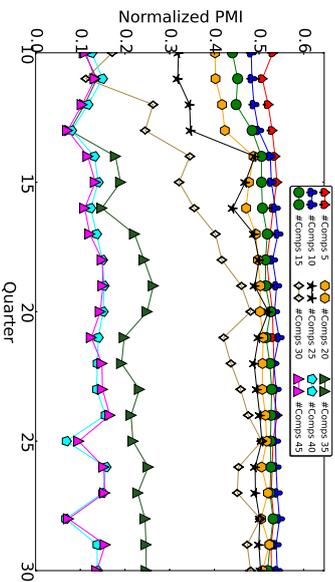


Figure 13: Cluster coherence (NPMI) of NMF model for varied number of clusters ($k \in [5, 45]$) on Wikipedia quarterly training datasets.

/ interpretability. The analysis provides further support for our choice of $H = 10$ for parametric models.

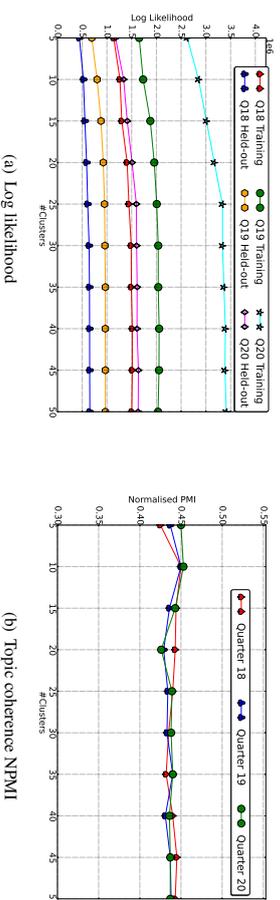


Figure 14: Log likelihood and cluster coherence for Bayesian vMFMM with varied number of clusters ($k \in [5, 50]$) on the 18th (Q18), 19th (Q19) and 20th (Q20) quarter of Wikipedia editor dataset.

One main advantage of the reversible jump MCMC algorithm is the ability to explore multiple models simultaneously, which brings side-benefit that can refine the inappropriate initialization of model parameters. To explore this point, we present a plot of log likelihood and estimated number of components of TVMFMM on selected Wikipedia and synthetic datasets after every 100 iterations. The results are presented in Figure 15, from which we observe that: the log likelihood of the model increases in the beginning, but then becomes relatively stable as more iterations proceed; there are some fluctuations in the log likelihood corresponding to synthetic training dataset with 7

components. The number of components increases with the iterations in the beginning, and then experiences some fluctuations with more iterations. By checking the output of model statistics at these points, we notice that the fluctuation points correspond to the accepted split-merge / birth-death moves where the model explores alternative models. In addition, the statistics of TVMFMM on Wikipedia data and synthetic data with 5 components shows better mixing property (i.e. less fluctuations) than those on synthetic data with 7 components. The results are consistent with our statement about the advantage of RMCMC.

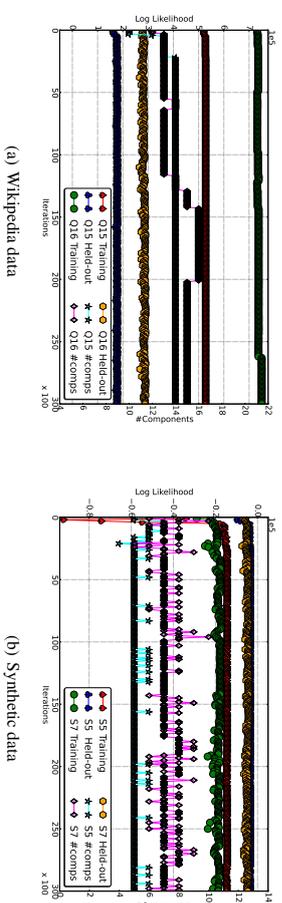
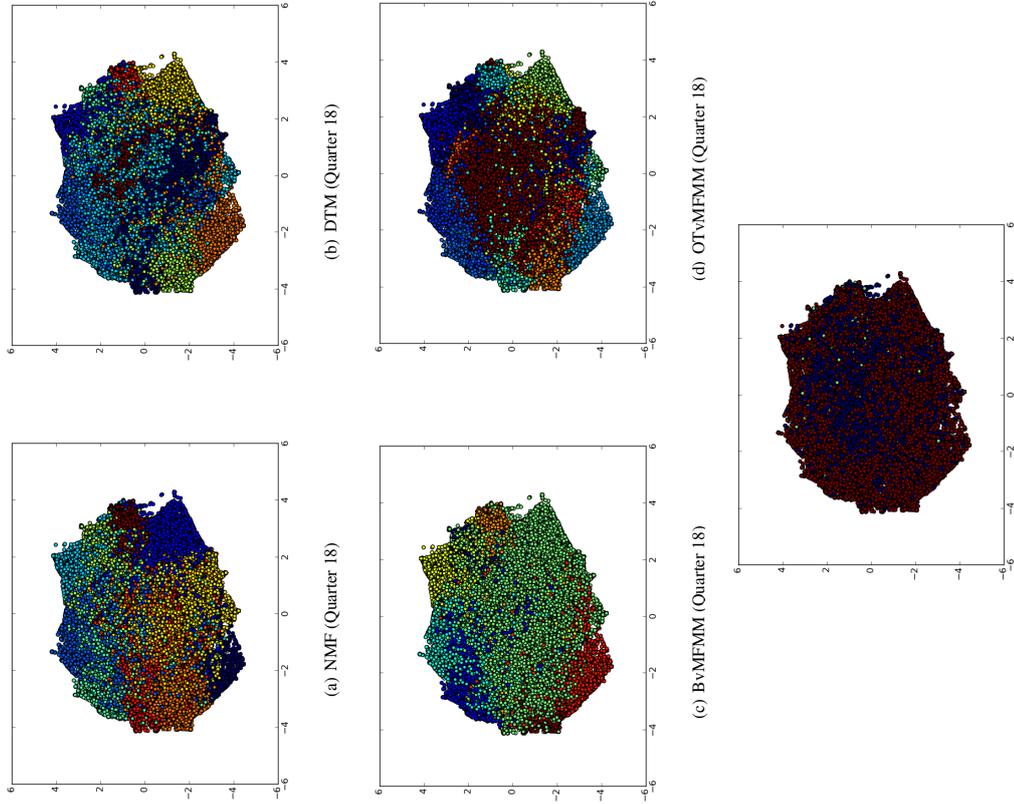


Figure 15: Trace plot of log likelihood and estimated number of components for TVMFMM after each 100 iterations on the 15th (Q15) and 16th (Q16) quarter of Wikipedia editor datasets, and synthetic datasets with 5 (S5) and 7 (S7) components. Where the left yaxis corresponds to log likelihood and the right yaxis represents #components.

Appendix G: Discriminative Analysis for Different Models

Embedding methods, such as t-distributed stochastic neighbourhood embedding (t-SNE; van der Maaten and Hinton (2008)) can be used to visualise high-dimensional data in a two or three-dimensional map. This visualisation provides a unique insight into the discriminative ability of mixture models in terms of separating data points in low-dimensional representation.

Figure 16 presents a 2D embedding of the inferred topic estimation by five models, using the t-SNE method, where each dot represents an entry of user behavioural data and each color-shape represents a topic. Visually, the proposed OTvMFMM produces a relatively better separation of data points than NMF, DTM and BvMFMM, while DP-GMM does not produce a well-separated embedding, and data points assigned to different clusters tend to mix together. This is consistent with our qualitative analysis in Section 5.2.2 that OTvMFMM can produce more interpretable and intuitive topics than other models. Intuitively, a well-separated representation is more discriminative for data separation.



(c) DP-GMM (Quarter 18)

Figure 16: t-SNE 2D embedding of the topical representations by different models on the 18th quarter of Wikipedia Editor dataset.

Appendix H Effects of Prior Parameters

The prior parameters m and σ^2 control the range of the concentration parameters κ , where the value of κ affects the mixing property / convergence of the RJMCMC chain. Figure 17 presents the trace plot of the number of components and log likelihood for TvMFMM with different values for m and σ^2 on synthetic data with 5 and 7 components. The values of m and σ^2 are chosen so that the corresponding ranges of κ are able to illustrate the effects of appropriate and inappropriate priors on the convergence of the chain. If the priors m and σ^2 are appropriately set, the convergence of RJMCMC algorithm can be sped up, leading to well mixing chain, as observed in Figure 17 (a-b) that the chains begin converging from around the 5000th sweep onwards. On the other hand, when the priors m and σ^2 are inappropriately set, the convergence of RJMCMC algorithm can be slowed down, resulting in poor mixing chain, as obvious in Figure 17 (c-d) that the chains experience more fluctuations in the number of components and log likelihood over sweeps. The observations suggest that the trace plot can be used to diagnose whether the priors are appropriately set.

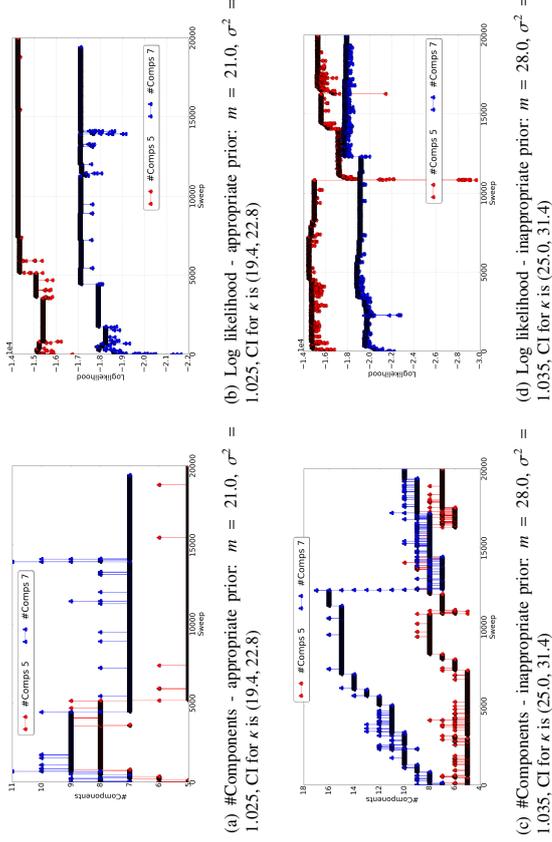


Figure 17: Trace plot of log likelihood and the number of components over iterations for TvMFMM with appropriate/inappropriate values for the prior parameters m and σ^2 on synthetic data, including 99.9% confidence interval (CI) for κ .

Appendix I Dynamics of Top Terms for User Roles – *Content Editors*

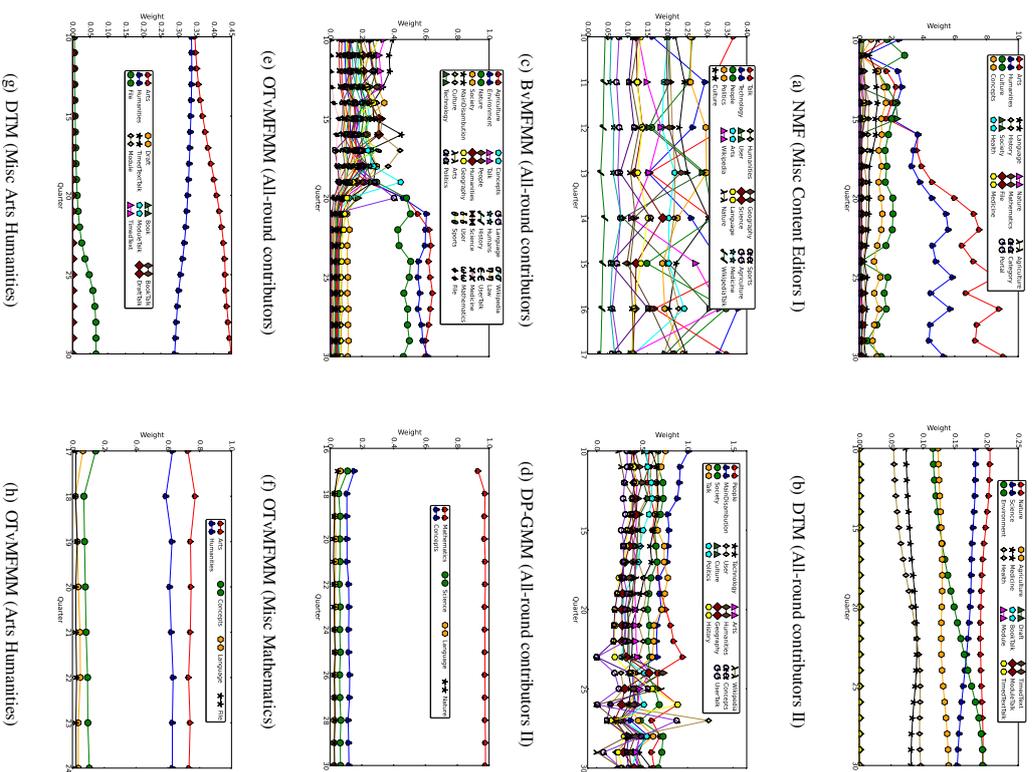


Figure 18: Evolution of top terms for common user roles (*Content Editors*) identified by different models.

Multivariate Spearman's ρ for Aggregating Ranks Using Copulas

Justin Bedó

The Walter and Eliza Hall Institute, 1G Royal Parade, Parkville Victoria 3052, Australia
The Department of Computing and Information Systems, the University of Melbourne, VIC 3010 Australia

cu@CUA0.ORG

Cheng Soon Ong

Data61, CSIRO, 7 London Circuit, Canberra ACT 2601 Australia
Research School of Computer Science, the Australian National University, Australia
The Department of Electrical and Electronic Engineering, the University of Melbourne, VIC 3010 Australia

CHENGSOON.ONG@ANU.EDU.AU

Editor: Jie Peng

Abstract

We study the problem of rank aggregation: given a set of ranked lists, we want to form a consensus ranking. Furthermore, we consider the case of extreme lists: i.e., only the rank of the best or worst elements are known. We impute missing ranks and generalise Spearman's ρ to extreme ranks. Our main contribution is the derivation of a non-parametric estimator for rank aggregation based on multivariate extensions of Spearman's ρ , which measures correlation between a set of ranked lists. Multivariate Spearman's ρ is defined using copulas, and we show that the geometric mean of normalised ranks maximises multivariate correlation. Motivated by this, we propose a weighted geometric mean approach for learning to rank which has a closed form least squares solution. When only the best (top-k) or worst (bottom-k) elements of a ranked list are known, we impute the missing ranks by the average value, allowing us to apply Spearman's ρ . We discuss an optimistic and pessimistic imputation of missing values, which respectively maximise and minimise correlation, and show its effect on aggregating university rankings. Finally, we demonstrate good performance on the rank aggregation benchmarks MQ2007 and MQ2008.

1. Introduction

Ranking is a central task in many applications such as information retrieval, recommender systems and bioinformatics. It may also be a subtask of other learning problems such as feature selection, where features are scored according to their predictiveness, and then the most significant ones are selected. One major advantage of ranks over scores is that the resulting predicted ranks are automatically normalised and hence can be used to combine diverse sources of information. However, unlike many other supervised learning problems, the problem of learning to rank (Lebanon and Mao, 2008; Liu, 2011) does not have the simple one example one label paradigm. This has led to many formulations of learning tasks, depending on what label information is available, including pairwise ranking, listwise ranking and rank aggregation.

This paper considers a novel formulation of rank aggregation based on multivariate extensions to Spearman's ρ . For a set of n objects from the domain Ω , we are given a set of d experts that rank these objects providing rankings R_1, \dots, R_d . Each rank is a permutation of the n objects, and can be represented as a vector of unique integers from 1 to n . The problem of rank aggregation is to construct a new vector R that is most similar to the set of d ranks provided by the experts. In this paper we use Spearman's correlation ρ , a widely used correlation measure for ranks Spearman (1904). Instead of decomposing the association into a combination of pairwise similarities, $\rho(R, R_1), \rho(R, R_2), \dots, \rho(R, R_d)$, we directly maximise the multivariate correlation

$$R^* = \arg \max_R \rho(R, R_1, R_2, \dots, R_d).$$

Measures of association such as Spearman's ρ capture the concordance between random variables (Nelsen, 2006). Informally, random variables are concordant if large values of one tend to be associated with large values of the other. Let (x_i, y_i) and (x_j, y_j) be two observations of a pair of continuous random variables. We say that (x_i, y_i) and (x_j, y_j) are *concordant* if $x_i < x_j$ and $y_i < y_j$ or $x_i > x_j$ and $y_i > y_j$. If the inequalities disagree, we say that the samples are *discordant*. The concept of concordance captures only the order of the random variables, and is invariant to their values, and therefore is ideal for analysing ranks. As will be described in section 3.3, Spearman's ρ is based on the difference between the concordance and discordance of the samples.

In short, Spearman's correlation can be defined as the concordance Q between the copula C corresponding to the data and the independent copula π

$$\rho \propto Q(C, \pi).$$

We review the concept of copulas in section 2 and derive our generalisation of concordance in section 3. While the mathematical machinery to derive our proposed algorithm relies on constructions that may not be familiar to some machine learners, the resulting algorithm for rank aggregation is straightforward. We solve a least squares problem for n items,

$$\min_{\omega} \sum_{x=1}^n \left(l(x) - \sum_{j=1}^d \omega_j r_d(x) \right)^2,$$

where we minimise the weights $\omega_1, \dots, \omega_d$ corresponding to the d experts. It turns out that the appropriate transformation to learn weights between experts is to use logarithmic scaled ranks. In the above equation, $l(x)$ and $r(x)$ denote the logarithm of the labels and individual expert ranks respectively, with all ranks normalised uniformly to the interval $(0, 1)$. Since it is a least squares problem, there is a closed form solution for the optimal weights. This is in contrast to previous approaches to rank aggregation that involve complex optimisation methods or sampling.

1.1 Our Contributions

We theoretically justify why the above least squares problem provides a meaningful way to weight experts. We show that the geometric mean of a set of normalised ranks maximises

multivariate Spearman's ρ . This motivates our method which finds a setting of weights that maximise multivariate Spearman's ρ for a specific target (supervised rank aggregation).

As previously mentioned, in many applications of rank aggregation, only extreme ranks are available, whereas the standard definitions of Spearman's ρ require full ranks. For practical problems, the expert may only rank the most liked (top- k) or most disliked (bottom- k) objects where k can be different for each expert. We propose a method for estimating Spearman's ρ for extreme ranks by inputting the remaining ranks. We describe this method and show that it is an unbiased estimator in section 4.

This results in a non-parametric approach for rank aggregation that learns the weights of experts by solving a least squares problem. The weights in this case model dependencies between the rankings, i.e., the rankings are not independent. This is different to much prior work (see section 1.3) in that we explicitly learn the dependencies between experts simultaneously and not in a pairwise fashion. Our method thus offers significant computational benefits, modelling flexibility in the presence of dependencies between experts, and also interpretability due to the simplicity of the model. In section 6 we describe our empirical results for rank aggregation and show that our simple algorithm performs better than current state of the art results.

1.2 Multiple Representations of Ranks

There are a wide range of applications which benefit from rank analysis, resulting in various equivalent ways to represent ranks and orderings. The basic representation often used in introductory texts is to provide the list of objects, for example $[a, b, c, d, e, f]$, denoting the fact that a is the most highly ranked object and f is the lowest ranked. It is often more convenient to numerically represent the rank for computational purposes, that is to keep a list of integers $1, \dots, n$ corresponding to the rank of a particular object. For the example above, by maintaining the set of objects as is, the ranks are then $[1, 2, 3, 4, 5, 6]$. It turns out for empirical copula modeling, it is important that the numerical values are in the interval $(0, 1)$, and therefore we normalise the numerical representation by $n + 1$, that is $[\frac{1}{n+1}, \frac{2}{n+1}, \dots, \frac{n}{n+1}]$. However, note that the numerical representation is actually dependent on the fact that we have maintained the set of objects in a particular fashion. In fact, by the above numerical list, we are saying that object a has rank $\frac{1}{n+1}$, and object f has rank $\frac{n}{n+1}$. In other words, we are defining a permutation mapping $R : \Omega \rightarrow (0, 1)$ from the space of objects Ω to the interval $(0, 1)$.

1.3 Related Work

There are two related rank aggregation tasks: score based rank aggregation and order based rank aggregation. For score based rank aggregation objects are associated with scores, while for order based rank aggregation only the relative order of objects are available. There has been recent work on combining both scores and ranks (Scutley, 2010; Iyer and Bilmes, 2013). We consider the learning task referred to as the listwise approach in Lin (2011), where the input is a set of ranked lists of documents from multiple experts, and the learner has to predict the final ranks. Numerous proposals for solving the problem of combining multiple lists into a single list are surveyed in Lin (2011). Nin et al. (2012) has focused on learning a

good ranking from given features. A good review of probability and statistics applied to permutations is Diaconis (1988).

Spearman's ρ is a natural measure of similarity for distributions of permutations (Mallows, 1957; Eligner and Verducci, 1986). Interestingly, there has not been much work using Spearman's ρ for dealing with ranked data, but instead the focus has been on Kendall's τ . One difficulty of inference with the Mallows model (Mallows, 1957) for Spearman's ρ is that it involves estimating the permanent of a matrix. Our model is derived from the copula form of Spearman's ρ and allows a simple formulation for aggregation that does not require any computationally complex operations, thus providing a significant computational advantage.

Other previous approaches (Klementiev et al., 2008; Iyer and Bilmes, 2012) to rank aggregation considers pairwise comparisons between ranked lists. In contrast, our approach does not consider pairwise combinations and operates over all lists. We prove a result saying that the geometric mean of normalised ranks maximise Spearman's ρ (theorem 17), which is similar in spirit to the result in Iyer and Bilmes (2012) that shows that for Lovász-Bregman divergences the best aggregator is the arithmetic mean. This provides a computational advantage over pairwise methods as the number of lists grows.

Our work builds heavily on copula theory, and we use results from Nelsen (2006). Brief introductions to copulas can be found in Trivedi and Zimmer (2005), Genest and Favre (2007), and Elidan (2013). Further details on copula modeling are available in a recent book (Joe, 2014). Many of these results are presented for bivariate copulas only. There are fewer results on multivariate copulas (Joe, 1990; Nelsen, 1996) and their relation to Spearman's ρ (Ubeda Flores, 2005; Schmid et al., 2010), which we shall discuss later in this paper.

Finally, other well known measures of bivariate dependence have forms under the copula framework and have multivariate extensions. In particular, multivariate extensions of Kendall's τ have been proposed (Joe, 2014). It is possible investigations into these copula formulations results in other efficient aggregation methods with different tradeoffs, however in this work we focus on Spearman's ρ .

The work on partial ranks goes back to at least Critchlow (1985), who describes the rank aggregation task in terms of distances between rankings. We have applied the results of this paper to rank aggregation (Macintyre et al., 2014) and stability estimation (Bedő et al., 2014) in the domain of life sciences.

2. Copulas

Copulas are functions from the unit hypercube to the unit interval (Elidan, 2013). In this section we briefly review the bivariate setting, in preparation for the multivariate setting in the next section. The expert reader may skip directly to section 3 to see the definition of multivariate Spearman's ρ in terms of the multivariate copula.

2.1 Definition of Copulas

Intuitively, for continuous random variables copulas model the dependence component of a multivariate distribution after discounting for univariate marginal effects. We let \mathbb{R} denote the ordinary real line $(-\infty, \infty)$, and \mathbb{R} denote the extended real line $[-\infty, \infty]$. The following algebraic definition of bivariate copulas is generalised to the multivariate setting in section 3.

It essentially constrains copulas to be functions that are *monotonically increasing* along each dimension as well as towards the diagonal of the volume.

Definition 1 Let A_1 and A_2 be nonempty subsets of \mathbb{R} , and let $H(\cdot, \cdot)$ be a real function such that the domain of $H = A_1 \times A_2$. Let $B = [x_1, x_2] \times [y_1, y_2]$ be a rectangle all of whose vertices are in the domain of H . Then the H -volume of B is given by:

$$V_H(B) = H(x_1, y_1) + H(x_2, y_2) - H(x_1, y_2) - H(x_2, y_1).$$

Definition 2 A real function $H(\cdot, \cdot)$ is 2-increasing if its H -volume is non-negative, that is $V_H(B) \geq 0$ for all rectangles B whose vertices lie in the domain of H .

Definition 3 A copula is a function $C: [0, 1]^2 \rightarrow [0, 1]$ with the following properties:

1. For every $u, v \in [0, 1]$,

$$C(u, 0) = 0 = C(0, v)$$

$$C(u, 1) = u \quad \text{and} \quad C(1, v) = v$$

2. C is 2-increasing.

2.2 Relation Between Bivariate Cumulative Density Functions and Copulas

Sklar's theorem is central to the theory of copulas and is the foundation of many applications in statistics. Indeed, Sklar's theorem can be defined for general distribution functions outside of probabilistic settings. However, since we are interested in statistical applications we will consider cumulative distribution functions.

Theorem 4 (Sklar's theorem) Let $H(\cdot, \cdot)$ be a cumulative distribution function with marginals $F(\cdot)$ and $G(\cdot)$. Then there exists a copula $C: [0, 1]^2 \rightarrow [0, 1]$ such that for all x, y in \mathbb{R} ,

$$H(x, y) = C(F(x), G(y)).$$

If $F(\cdot)$ and $G(\cdot)$ are continuous then $C(\cdot, \cdot)$ is unique; otherwise $C(\cdot, \cdot)$ is uniquely determined on the ranges of $F(\cdot)$ and $G(\cdot)$.

Conversely, if $C(\cdot, \cdot)$ is a copula and $F(\cdot)$ and $G(\cdot)$ are cumulative distribution functions then the function $H(\cdot, \cdot)$ is a bivariate cumulative distribution function with marginals $F(\cdot)$ and $G(\cdot)$.

3. Spearman's ρ

We briefly review the bivariate model to lay out the approach for estimating the copula using data, the so-called empirical copula.

3.1 Empirical bivariate Spearman's ρ

Let R and S be ranking functions, which are bijections mapping elements x in the domain U to $[1, 2, \dots, n]$. The domain U represents the space of objects that we are interested in ranking, such as documents retrieved in response to a query or the biomarkers most

associated with a disease. Since we consider only the ranks of the object $R(x)$ and $S(x)$, the actual domain U does not affect the analysis. The sums below are over the n objects x . Similar to the approach of Pearson's correlation for the measure of dependence, Spearman's ρ is a measure of correlation between ranks, empirically given by:

$$\rho_n = \frac{\sum_x (R(x) - \bar{R})(S(x) - \bar{S})}{\sqrt{\sum_x (R(x) - \bar{R})^2 \sum_x (S(x) - \bar{S})^2}}, \tag{1}$$

where $\bar{R} := \frac{1}{n} \sum_x R(x)$ and $\bar{S} := \frac{1}{n} \sum_x S(x)$ are the empirical means of the respective random variables. This is equivalent to applying Pearson's correlation to the ranks instead of the values of the score function itself. There is no direct way to generalise this expression to more than two ranking functions, but as we shall see in section 3.3 we can obtain an expression via the copula.

By substituting the definitions of the empirical means and rearranging the terms, we obtain

$$\rho_n = \left(\frac{n+1}{n-1} \right) \left[\frac{12}{n} \sum_x \frac{R(x) S(x)}{n+1} - 3 \right].$$

The constants 12 and 3 seem strange, but are a natural consequence of the mean and variance of a list of ranks. As we will see later, these constants are dependent only on the dimension of the copula. Similar to the definition of an empirical CDF, we define an empirical copula as:

$$C_n(u, v) = \frac{1}{n} \sum_x \mathbf{1} \left(\frac{R(x)}{n+1} \leq u, \frac{S(x)}{n+1} \leq v \right),$$

where $\mathbf{1}$ is the indicator function. This allows us to re-express the form of ρ_n above in terms of an integral over the unit square,

$$\rho_n = \left(\frac{n+1}{n-1} \right) \left[\frac{12}{n} \sum_x \frac{R(x) S(x)}{n+1} - 3 \right] = \left(\frac{n+1}{n-1} \right) \left[12 \int_{[0,1]^2} uv C_n(u, v) - 3 \right].$$

It can be shown (Nelsen, 2006; Genest and Favre, 2007) that ρ_n is an asymptotically unbiased estimator of

$$\rho = 12 \int_{[0,1]^2} C(u, v) du dv - 3,$$

where C is the population version of C_n .

3.2 Multivariate Copulas

We now generalise the definitions in section 2.1 to the multivariate case. The concepts are essentially the same, constraining the copula to be "monotonically increasing" in the interval $[0, 1]$ and also towards the center of the volume (Durante and Sempi, 2010).

Definition 5 Let A_j be nonempty subsets of \mathbb{R} for $j = 1, \dots, d$, and let $H_d: A_1 \times \dots \times A_d \rightarrow \mathbb{R}$. Let $B = [a_1, b_1] \times \dots \times [a_d, b_d]$ be the d -box where all vertices are contained in $\text{Dom } H_d$. Then the H_d -volume of B is the d^{th} order difference:

$$V_{H_d}(B) = \Delta_{a_d}^{b_d} \dots \Delta_{a_1}^{b_1} H_d(\vec{v}),$$

where

$$\Delta_{a_i}^b H(\vec{t}) = H_d(t_1, \dots, t_{i-1}, b_i, t_{i+1}, \dots, t_d) - H_d(t_1, \dots, t_{i-1}, a_i, t_{i+1}, \dots, t_d).$$

Definition 6 A real function H_d is grounded if $H_d(\vec{t}) = 0$ for all $t \in \text{Dom } H_d$ such that $t_j = a_j$ for at least one $j \in \{1, \dots, d\}$.

Definition 7 A real function H_d is d -increasing if $V_{H_d}(B) \geq 0$ for all n -boxes B whose vertices lie in the domain of H .

Definition 8 A multivariate copula has the following properties:

1. $\text{Dom } C = [0, 1]^d$
2. C has margins $C_j(u) = C(1, \dots, 1, u, 1, \dots, 1) = u$ for all j and $u \in I$
3. C is grounded
4. C is d -increasing.

There is an alternative probabilistic definition that may be more familiar to readers with a statistical background.

Definition 9 Let U_1, \dots, U_d be real uniformly distributed random variables on the unit interval $\sim U([0, 1])$. A copula function $C: [0, 1]^d \rightarrow [0, 1]$ is a joint distribution

$$C(u_1, \dots, u_d) = P(U_1 \leq u_1, \dots, U_d \leq u_d).$$

Let $X \sim F$ be a continuous random variable such that the inverse of the CDF F^{-1} exists. What is the distribution of $F(x) = P(X \leq x)$?

$$\begin{aligned} P(F(X) \leq u) &= P(F^{-1}(F(X)) \leq F^{-1}(u)) \\ &= P(X \leq F^{-1}(u)) \\ &= F(F^{-1}(u)) = u \end{aligned}$$

The above calculation shows that the distribution is uniform, i.e. $F(x) \sim U([0, 1])$. This can be considered to be the *copula trick*, as the user has the freedom to choose the copula independently of the marginal distributions.

3.3 Multivariate Extension of Spearman's ρ

We generalise the concept of concordance to the multivariate setting such that we can define multivariate Spearman's ρ in an analogous way to the bivariate ρ as defined in Nelsen (2006).

Recall that two random variables are concordant if they tend to be in the same order, that is (x_i, y_i) and (x_j, y_j) are concordant if $(x_i - x_j)(y_i - y_j) > 0$, and are discordant if $(x_i - x_j)(y_i - y_j) < 0$. The concordance function Q denotes the difference between the probabilities of concordance and discordance, and as the following theorem shows, can be expressed in terms of the copulas. The proof is in Nelsen (2006).

Theorem 10 (Concordance function) Let (X_1, Y_1) and (X_2, Y_2) be two independent vectors with joint distributions $H_1(x, y) = C_1(F(x), G(y))$ and $H_2(x, y) = C_2(F(x), G(y))$ respectively. Then the concordance function Q is given by

$$\begin{aligned} Q(C_1, C_2) &:= P[(X_1 - X_2)(Y_1 - Y_2) > 0] - P[(X_1 - X_2)(Y_1 - Y_2) < 0] \\ &= 4 \int_{[0, 1]^2} C_2(u, v) dC_1(u, v) - 1 \end{aligned}$$

We now state the generalisation of concordance to the multivariate case (Nelsen, 1996; Joe, 1990). Further details of multivariate concordance can be found in Taylor (2007) and Schmid et al. (2010).

Definition 11 (Multivariate concordance) Let (X_1, \dots, X_d) and (Y_1, \dots, Y_d) be two independent d -vectors with joint distributions $C_X(F(x))$ and $C_Y(F(y))$ where $F(x) = F_1(x_1), \dots, F_d(x_d)$ and $F(y) = F_1(y_1), \dots, F_d(y_d)$ are the marginal distributions, and C_X, C_Y are the respective d copulas. Then the concordance function Q is given by

$$Q(C_X, C_Y) := 2^d \int_{[0, 1]^d} C_X(u) dC_Y(u) - 1.$$

Note that although the integral is a straight forward generalisation of theorem 10, it is no-longer equal to the difference between the probability of concordance and discordance. Consequently, the properties possessed by Q are different for $d > 2$.

There are three copulas that are of particular interest: the independent copula $\pi(u) := \prod_i u_i$, and the upper and lower Fréchet-Hoeffding bounds, $M(u) = \min\{u_1, u_2, \dots, u_d\}$ and $W(u) = \max\{u_1 + u_2 + \dots + u_d - (d-1), 0\}$ respectively (Joe, 2014, pg. 48). Note that that while W is point-wise sharp, this lower bound is not itself a copula, and hence the lower bound is not tight (Ubeda Flores, 2005).

Theorem 12 Let C, C' , and Q be given as in definition 11, M and W be the upper and lower Fréchet-Hoeffding bounds respectively, and assume $d > 2$. Then

1. Q is symmetric in its arguments if $C = C'$.
2. Q is non-decreasing in the first argument, and both arguments if $C = C'$.
3. $-1 \leq Q(W, W) \leq Q(C, C) \leq Q(M, M) = 2^{d-1} - 1$.
4. $Q(\pi, \pi) = 0$.

Proof

Property 1 The first property is clear from the definition of $Q(C, C')$ and the properties of integration.

Property 2 Q is non-decreasing in the first argument by properties of integration. For the second part, notice that

$$\begin{aligned} \int C(u) dC(u) &= C^2(u) - \int C(u) dC(u) \\ &\Rightarrow \int C(u) dC(u) = \frac{1}{2} C^2(u) \end{aligned}$$

by applying integration by parts. The property now follows.

Property 3 It follows that

$$\begin{aligned} Q(M, M) &= 2^d \int_{[0,1]^d} M(u) \, dM(u) - 1 \\ &= 2^d \int_0^1 u \, du - 1 \\ &= 2^{d-1} - 1, \end{aligned}$$

and

$$\begin{aligned} Q(W, W) &= 2^d \int_{[0,1]^d} W(u) \, dW(u) - 1 \\ &\geq 2^d \int_0^1 0 \, du - 1 \\ &= -1. \end{aligned}$$

Property 3 now follows from the first two properties.

Property 4

$$\begin{aligned} Q(\pi, \pi) &= 2^d \int_{[0,1]^d} \pi(u) \, d\pi(u) - 1 \\ &= 2^d \int_{[0,1]^d} u \, du - 1 \\ &= \frac{2^d}{2^d} - 1 \\ &= 0 \end{aligned}$$

■

It is clear from this theorem that Q is well calibrated at $Q(W, W)$ and $Q(\pi, \pi)$, however not for $Q(M, M)$. Consequently, with this multidimensional extension it becomes increasingly difficult to estimate discordance as d increases.

Proposition 13 Let Q be given as in definition 11, and M and π be the upper Fréchet–Hoeffding bound and the independent copula respectively, then

$$Q(M, \pi) = Q(\pi, M) = \frac{2^d - (d + 1)}{d + 1}. \tag{2}$$

Proof To show the symmetry,

$$\begin{aligned} Q(M, \pi) &= 2^d \int_{[0,1]^d} M(u) \, d\pi(u) - 1 \\ &= 2^d \int_{[0,1]^d} u_1 u_2 \cdots u_d \, du - 1 \end{aligned}$$

and

$$\begin{aligned} Q(\pi, M) &= 2^d \int_{[0,1]^d} \pi(u) \, dM(u) - 1 \\ &= 2^d \int_{[0,1]^d} u_1 u_2 \cdots u_d \, du - 1. \end{aligned}$$

To obtain the second equality, we observe that

$$\begin{aligned} \int_{[0,1]^d} u_1 u_2 \cdots u_d \, du &= \int_0^1 u^d \, du \\ &= \frac{1}{d+1} u^{d+1} \Big|_0^1 \\ &= \frac{1}{d+1}, \end{aligned}$$

and therefore the expression for $Q(M, \pi)$ follows. ■

In terms of the concordance function, Spearman's ρ is given by the concordance between the copula C and the independent copula $\pi(u) := \prod_i u_i$. However, unlike the symmetry in proposition 13, the concordance function is in general not symmetric with respect to its arguments. This gives us two possible ways of defining multivariate Spearman's ρ , corresponding to $Q(C, \pi)$ and $Q(\pi, C)$. Both generalisations are equivalent in the bivariate case, and has been called ρ_d^- and ρ_d^+ by Nelsen (1996) and ρ_1 and ρ_2 by Schmid and Schmidt (2007) respectively. Naturally, there is a third symmetric generalisation which is the average of them.

Definition 14 (Multivariate Spearman's ρ)

$$\rho_d^- = h(d)Q(\pi, C) = h(d) \left[2^d \int_{[0,1]^d} C(u) \, du - 1 \right] \tag{3}$$

and

$$\rho_d^+ = h(d)Q(C, \pi) = h(d) \left[2^d \int_{[0,1]^d} \pi(u) \, dC(u) - 1 \right], \tag{4}$$

where $h(d) = \frac{d+1}{2^d - (d+1)}$ is the normalisation factor.

The scaling factor $h(d)$ is derived such that the maximum correlation is 1. Thus, for Spearman's ρ , this is the concordance between the maximum copula M and the independent copula π , which we obtain by proposition 13:

$$h(d) = 1/Q(M, \pi) = \frac{d+1}{2^d - (d+1)}. \tag{5}$$

Spearman's correlation can equivalently be seen as measuring average orthant dependence, and the two versions ρ_d^+ and ρ_d^- correspond to whether we look at the upper or lower orthant (Nelsen, 1996). Positive upper orthant dependence is defined as

$$P(X > x) \geq \prod_{i=1}^d P(X_i > x_i),$$

and positive lower orthant dependence is defined as

$$P(X \leq x) \geq \prod_{i=1}^d P(X_i \leq x_i).$$

When $d = 2$, the two definitions are the same and are called positive quadrant dependence (Lehmann, 1966), as we have already observed for the concordance function:

$$\begin{aligned} P(X_1 > x_1, X_2 > x_2) &\geq P(X_1 > x_1)P(X_2 > x_2) \\ &\geq [1 - P(X_1 \leq x_1)][1 - P(X_2 \leq x_2)] \\ &\geq 1 - P(X_1 \leq x_1) - P(X_2 \leq x_2) + P(X_1 \leq x_1)P(X_2 \leq x_2). \end{aligned}$$

Rearranging gives

$$P(X_1 > x_1, X_2 > x_2) + P(X_1 \leq x_1) + P(X_2 \leq x_2) - 1 \geq P(X_1 \leq x_1)P(X_2 \leq x_2).$$

The left hand side is $P(X_1 \leq x_1, X_2 \leq x_2)$.

Observe that the scaling factor $h(d)$ is the same for both ρ_d^- and ρ_d^+ due to proposition 13. Furthermore, since $P(X_i > x_i) = 1 - P(X_i \leq x_i)$ for each random variable, the two versions of Spearman's ρ correspond to looking at whether we interpret the ranks as top down or bottom up. Converting from one version to the other can be done by reinterpreting the data. For a particular application, the choice of which version to use depends on the ranks that are available. We will focus on ρ_d^+ henceforth.

Recall that for a set of n objects from the domain Ω , we are given a set of d experts that rank these objects providing ranks R_1, \dots, R_d , where each R_j is a bijection to $(0, 1)$. Putting (4) and (5) together, we obtain the following expression for multivariate Spearman's correlation:

$$\rho(R_1, \dots, R_d) = h(d)Q(C; \pi) = \frac{d+1}{2^d - (d+1)} \left[2^d \int_{[0,1]^d} \pi(u) dC(u) - 1 \right]. \quad (6)$$

In practice, we do not have access to the population version of the copula $C(u)$ but have the empirical copula $C_n(u)$. We discuss this further in section 5.

Unlike the bivariate case, as the number of dimensions increases, the lower bound of Spearman's ρ tends to zero. This counterintuitive fact can be understood by considering the three dimensional case. Consider three rankings R_1, R_2 , and R_3 . If R_1 and R_2 are anti-correlated ($\rho = -1$), and at the same time R_1 and R_3 are also anti-correlated, this implies that R_2 and R_3 must be perfectly correlated ($\rho = 1$). Hence, the overall 3 dimensional

correlation is no longer -1. This can be made precise by considering the inclusion-exclusion principle, which results in the following relation from Nelsen (1996):

$$\frac{1}{2}(\rho_d^-(R_1, R_2, R_3) + \rho_d^+(R_1, R_2, R_3)) = \frac{1}{3}(\rho(R_1, R_2) + \rho(R_1, R_3) + \rho(R_2, R_3)).$$

The following corollary defines the lower bound as the number of dimensions increases.

Corollary 15 Under the minimum Fréchet-Hoeffding bound W , $Q(W, \pi) \geq -1$ and

$$\lim_{d \rightarrow \infty} \rho(R_1, \dots, R_d) \geq h(d)Q(W, \pi) = 0.$$

In particular, for dimension d ,

$$\rho(R_1, \dots, R_d) \geq \frac{2^n - (n+1)!}{n!(2^n - (n+1))}.$$

Proof This follows immediately from the bound $-1 \leq Q(W, \pi) \leq 0$ (from theorem 12) since $h(d)$ goes to zero as $d \rightarrow \infty$. The lower bound has also been observed in Nelsen (1996) and Schmid et al. (2010). ■

In summary, the multivariate extension of Spearman's correlation is still calibrated under maximum correlation as it achieves a value of 1, but it becomes increasingly difficult to observe anti-correlated sets of ranks as the number of lists to be aggregated increases. In the next section, we investigate an aggregation algorithm that maximises correlation. The effect of the lower bound is discussed with respect to imputing missing values in section 5.

4. Optimal Aggregation with Spearman's ρ

The empirical copula requires R and S to comprise of ranks for the same set of elements, that is $\text{Dom } R = \text{Dom } S$. Recall from section 3.1 that ranks map to the range $\{1, \dots, n\}$, but the empirical copula is expressed in terms of fractional ranks (divided by $n+1$). In the following it is convenient to work with normalised ranks, that is to consider R and S as bijections to $(0, 1)$. The expression for the empirical copula then simplifies to

$$C_n(u, v) = \frac{1}{|\Omega|} \sum_{x \in \Omega} \mathbf{1}(R(x) \leq u, S(x) \leq v), \quad (7)$$

where Ω is the domain of the objects we are interested in ranking. Correspondingly, the d dimensional empirical copula for n objects given by

$$C_n(u) = \frac{1}{n} \sum_x \prod_{j=1}^d \mathbf{1}(R_j(x) \leq u_j), \quad (8)$$

where $R_1(x), \dots, R_d(x)$ is the rankings of the d experts. Plugging the empirical copula (8) expression into Spearman's ρ (6), and observing that integrating the product over the copula is the product of the ranks Schmid and Schmidt (2007), we obtain an empirical expression for multivariate Spearman's correlation:

$$\rho_n(R_1, \dots, R_d) = h(d) \left[\frac{2^d}{n} \sum_x \prod_{j=1}^d R_j(x) - 1 \right]. \quad (9)$$

4.1 Geometric Mean is Optimal

We are now in a position to derive the deceptively simple result: the ranking R that maximises correlation with a given set of rankings $\{R_1, \dots, R_d\}$ is given by the geometric mean of R_1, \dots, R_d . The following definition is needed to capture the notion that ranks only depend on the order.

Definition 16 (Rank generator) $\sigma: \mathbb{R}^{[d]} \rightarrow [0, 1]^{[d]}$ is a rank generator if:

- for all $x, y \in \Omega$ and R with domain Ω , $R(x) < R(y) \iff \sigma \circ R(x) < \sigma \circ R(y)$;
- for any rankings R, R' with domain Ω there exists a permutation ξ such that $\sigma \circ R' = \sigma \circ \xi \circ R$;
- for any permutation ξ , $\xi \circ \sigma = \sigma \circ \xi$.

A rank generator formalises the idea of generating a rank: the ranks it generates must be invariant to scale and only dependent on the ordering of elements. The standard ranking functions from statistics such as fractional ranking and dense ranking fit into this framework.

Theorem 17 Let $\{R_1, R_2, \dots, R_d\}$ be a set of rankings with common domain Ω and σ be a rank generator. Then

$$\arg \max_{R \in \text{codom } \sigma} \rho_n(R, R_1, R_2, \dots, R_d) = \sigma \left(\prod_{j=1}^d R_j \right).$$

Proof Consider the expression for Spearman's ρ_n (9):

$$\rho_n(R, R_1, R_2, \dots, R_d) = h(d+1) \left[\frac{2^{d+1}}{n} \sum_x \left(R(x) \prod_{j=1}^d R_j(x) \right) - 1 \right].$$

Focusing on the terms in the sum, showing that the best possible $R(x)$ is $\prod_{j=1}^d R_j(x)$ reduces to showing

$$\sum_{x \in U} \sigma \circ P(x) P(x)$$

is maximal, where $P := \prod_j R_j$. Suppose there exists an P' such that

$$\sum_{x \in U} \sigma \circ P'(x) P(x) > \sum_{x \in U} \sigma \circ P(x) P(x).$$

By definition of σ , there exists a permutation ξ such that

$$\begin{aligned} \sum_{x \in U} \sigma \circ P'(x) P(x) &= \sum_{x \in U} \sigma \circ \xi \circ P(x) P(x) \\ &= \sum_{x \in U} \xi \circ \sigma \circ P(x) P(x) \\ &> \sum_{x \in U} \sigma \circ P(x) P(x). \end{aligned}$$

This is a contradiction for any permutation ξ as σ is order preserving. ■

Corollary 18 The converse applies, that is:

$$\arg \min_{R \in \text{codom } \sigma} \rho_n(R, R_1, R_2, \dots, R_d) = \sigma \left(\prod_{j=1}^d (1 - R_j) \right).$$

Proof Proof follows from a similar argument. ■

5. Empirical Copulas with Partial Lists

In many applications it is prohibitive to obtain complete annotations of the object ranks. For example, in the document retrieval setting, this amounts to providing ranks for all documents. The empirical copula requires the set of rankings $\{R_1, \dots, R_d\}$ to comprise of ranks for the same set of elements, that is $\text{Dom } R_1 = \dots = \text{Dom } R_d$. Hence, a key challenge in applying Spearman's ρ to rank aggregation is to estimate the statistic on incompletely labelled lists.

Recall the definition of the empirical copula (7). We now consider the case where $\text{Dom } R \neq \text{Dom } S$, but R and S are generated from two *top ranked* lists. We define extended rankings R', S' with codomain $[0, 1]$ such that $\text{Dom } R' = \text{Dom } R \cup \text{Dom } S = \text{Dom } S'$. One way to impute the missing values is to set them to a constant value for all the ranks below the top- k ranks. This value is chosen to be the mid point between the start and end of the missing section. The values in the top- k are retained to be the original values in the extension. The definition below formally defines this notion. Note that we have to renormalise the values.

Definition 19 (non-informative extension) Let R be a ranking operator and R' be its extension to domain $\text{Dom } R'$. Then,

$$R'(x) = \begin{cases} \frac{|\text{Dom } R|}{|\text{Dom } R'|} R(x) & x \in \text{Dom } R \\ \frac{|\text{Dom } R| + |\text{Dom } R'|}{2|\text{Dom } R'|} & \text{otherwise} \end{cases} \quad (10)$$

$\forall x \in \text{Dom } R'$.

We call this the non-informative extension since it assumes that all items that are not ranked have the same rank (the mean of the missing ranks). Note that the two experts R_i and R_j may have ranked different numbers of objects. An advantage of this extension is that it can easily deal with the case of more than two experts. Consider d experts R_1, \dots, R_d , each of which may have ranked a different subset of the objects. Hence the extension has to impute values on the union of items from all experts. Denote $\text{Dom } R' := \text{Dom } R_1 \cup \dots \cup \text{Dom } R_d$, then we can apply definition 19 to complete each ranking operator R_j . An additional advantage to the non-informative extension is that it results in a consistent ranking.

Definition 20 An extended ranking R' of R is called consistent if the following axioms hold:

1. $R'(x) < R'(y) \forall x, y \in \text{Dom } R$ with $R(x) < R(y)$
2. $R'(x) = R'(y) \forall x, y \in \text{Dom } R$ with $R(x) = R(y)$
3. $R'(y) > R'(x) \forall x \in \text{Dom } R, y \in \text{Dom } R'$

If $E[R] = E[R']$ also holds, then R' is called strictly consistent.

Lemma 21 Definition 19 produces a consistent ranking. If $E[R] = \frac{1}{2}$ then (10) produces a strictly consistent ranking.

Proof The notation $|\text{Dom } R|$ can become unwieldy in following proof. We therefore adopt the shorthand notations $r := |\text{Dom } R|$ and $r' := |\text{Dom } R'|$ for the size of the respective sets. Axioms 1 and 2 are satisfied by definition as the map $x \mapsto \frac{x}{r}$ is monotonic. For all $x \in \text{Dom } R' \setminus \text{Dom } R$,

$$R'(x) = \frac{r+r'}{2r'} \leq 2R(y) \frac{r+r'}{2r'} \leq \frac{2R(y)r}{2r'} = \frac{r}{r'} R(y) = R'(y)$$

for any $y \in \text{Dom } R$, satisfying axiom 3. Furthermore, as

$$\begin{aligned} E[R'] &= \frac{1}{r'} \left(\sum_{x \in \text{Dom } R} R'(x) + \sum_{x \in \text{Dom } R' \setminus \text{Dom } R} R'(x) \right) \\ &= \frac{1}{r'} \left(\frac{r}{r'} \sum_{x \in \text{Dom } R} R(x) + (r' - r) \frac{r+r'}{2r'} \right) \\ &= \frac{1}{r'} \left(\frac{r^2}{r'} E[R] + (r' - r) \frac{r+r'}{2r'} \right) \\ &= \frac{r^2(2E[R] - 1) + r'}{2r'^2}, \end{aligned}$$

R' is strictly consistent if $E[R] = \frac{1}{2}$. ■

Definition 19 is called a *non-informative* extension as it uses no additional information and does not bias the inputted elements in anyway: imputed values are all considered tied and mapped to the same value. Furthermore, the strictly consistent property that definition 19 satisfied is important when using fractional ranking as it guarantees no introduction of bias.

Note also that there is a dual imputation whereby missing values are assigned to the top of the list rather than the bottom. This is equivalent to the above imputation applied to reverse rankings. The choice of top or bottom imputation is application dependent.

5.1 Empirical Upper and Lower Bounds

Proposition 22 For top- k lists where k of n items are ranked by all d experts with codomain $\{1, \dots, n\}$ (i.e., unnormalised ranks), the Spearman's ρ is bounded by

$$\rho_n(R_1, R_2, \dots, R_d) = \rho_k(R_1, R_2, \dots, R_d) + C,$$

where

$$\begin{aligned} 2^d h(d) & \left(\frac{k(k+1)^d - n(n+1)^d}{\sum_{i=1}^k \prod_{j=1}^d \frac{R_{ij}(i)}{k+1}} + k \frac{\sum_{i=k+1}^n i^{\frac{d}{2}} (k-i+n+1)^{\frac{d}{2}}}{n(n+1)^d k} \right) \\ & \leq C \leq \\ & 2^d h(d) \left(\frac{k(k+1)^d - n(n+1)^d}{\sum_{i=1}^k \prod_{j=1}^d \frac{R_{ij}(i)}{k+1}} + \frac{\sum_{i=k+1}^n i^d}{\sum_{i=k+1}^n i^d} k \right) \\ & \frac{2^d h(d)}{n(n+1)^d k}. \end{aligned}$$

Proof Proof sketch: the definition of ρ for unnormalised rankings is

$$\rho_n(R_1, R_2, \dots, R_d) = h(d+1) \left[\frac{2^{d+1}}{n} \sum_{i=1}^n \left(\prod_{j=1}^d \frac{R_{ij}(i)}{n+1} \right) - 1 \right]$$

By considering the difference $\rho_n(R_1, R_2, \dots, R_d) - \rho_k(R_1, R_2, \dots, R_d)$ and factorising out the common terms, we obtain

$$C = \frac{(d+1) 2^d \left(k \left(\sum_{i=1}^n \prod_{j=1}^d \frac{R_{ij}(i)}{n+1} \right) - \left(\sum_{i=1}^k \prod_{j=1}^d \frac{R_{ij}(i)}{k+1} \right) n \right)}{(2^d - d - 1) k n}.$$

The term $\sum_{i=1}^n \prod_{j=1}^d \frac{R_{ij}(i)}{n+1}$ can be bounded above by

$$\sum_{i=1}^n \prod_{j=1}^d \frac{R_{ij}(i)}{n+1} = \sum_{i=1}^k \prod_{j=1}^d \frac{R_{ij}(i)}{n+1} + \sum_{i=1+k}^n \prod_{j=1}^d \frac{R_{ij}(i)}{n+1} \leq \sum_{i=1}^k \prod_{j=1}^d \frac{R_{ij}(i)}{n+1} + \sum_{i=1+k}^n \left(\frac{i}{n+1} \right)^d,$$

and below by

$$\begin{aligned} \sum_{i=1}^n \prod_{j=1}^d \frac{R_{ij}(i)}{n+1} &= \sum_{i=1}^k \prod_{j=1}^d \frac{R_{ij}(i)}{n+1} + \sum_{i=1+k}^n \prod_{j=1}^d \frac{R_{ij}(i)}{n+1} \\ &\geq \sum_{i=1}^k \prod_{j=1}^d \frac{R_{ij}(i)}{n+1} + \sum_{i=k+1}^n \left(\prod_{j=i}^{\lfloor \frac{i}{2} \rfloor} \frac{i}{n+1} \right) \prod_{j=\lceil \frac{i}{2} \rceil}^d \frac{n-i+1+k}{n+1}, \end{aligned}$$

giving us the bounds in the proposition. ■

5.2 Optimal Imputation

An alternative to the previously presented imputation method is to impute such that ρ is maximised or minimised. In general this is a NP-hard problem as it involves searching all permutations. In this section, we formulate this as an optimisation problem.

Let $\mathbb{I} = \{1, \dots, n\} \times \{1, \dots, d\}$ be indices over n items and d experts. Let $\mathbb{O} \subset \mathbb{I}$ be the observed indices (for which we have a rank) and define $\mathbb{U} := \mathbb{I} \setminus \mathbb{O}$. We then have a rank function $R: \mathbb{O} \rightarrow \{1, \dots, n\}$. Recall that Spearman's ρ is determined by a sum of the products over ranks. By introducing a log transformation, we convert the product into a sum using the logarithm rule:

$$\sum_{i=1}^n \left(\prod_{j=1}^d \frac{R_j(i)}{n+1} \right) = \sum_{i=1}^n \left(\exp \log \prod_{j=1}^d \frac{R_j(i)}{n+1} \right) = \sum_{i=1}^n \left(\exp \sum_{j=1}^d \log \frac{R_j(i)}{n+1} \right).$$

5.2.1 IMPUTING TO MAXIMISE CORRELATION

We can maximise Spearman's ρ by introducing binary indicators $x_{i,j,k}$ indexed over $\mathbb{I} \times \{1, \dots, n\}$ to denote a rank of k for item i in list j .

$$\max_{x_{i,j,k}} \sum_{i=1}^n \exp \left[\sum_{j=1}^d \sum_{k=1}^n x_{i,j,k} \log \left(\frac{k}{n+1} \right) \right]$$

such that

$$\sum_k x_{i,j,k} = 1 \quad \forall i, j \quad (11)$$

$$\sum_i x_{i,j,k} = 1 \quad \forall k, j \quad (12)$$

$$\sum_k x_{i,j,k} k = R(i, j) \quad \forall (i, j) \in \mathbb{O} \quad (13)$$

$$x_{i,j,k} \in \{0, 1\} \quad \forall i, j, k \quad (14)$$

Constraint (11) ensures an item is only assigned one rank per expert, and constraint (12) ensures a rank is only assigned once per expert. Finally, the third constraint (13) ensures known ranks are assigned.

5.2.2 IMPUTING TO MINIMISE CORRELATION

Analogously, we can consider the problem of minimising Spearman's ρ .

$$\min_{x_{i,j,k}} \sum_{i=1}^n \exp \left[\sum_{j=1}^d \sum_{k=1}^n x_{i,j,k} \log \left(\frac{k}{n+1} \right) \right] \quad (15)$$

such that

$$\sum_k x_{i,j,k} = 1 \quad \forall i, j$$

$$\sum_i x_{i,j,k} = 1 \quad \forall k, j$$

$$\sum_k x_{i,j,k} k = R(i, j) \quad \forall (i, j) \in \mathbb{O}$$

$$x_{i,j,k} \in \{0, 1\} \quad \forall i, j, k$$

By considering the relaxation of $x_{i,j,k} \in \{0, 1\}$ to $x_{i,j,k} \in [0, 1]$, we obtain a convex optimisation problem.

Proposition 23 *The relaxation of optimisation problem (15) such that $x_{i,j,k}$ is in the interval $[0, 1]$ is a convex optimisation problem.*

Proof The objective has the form $\sum_i \exp((x_i, \omega))$ with $\omega \in [\log(\frac{1}{n+1}), \dots, \log(\frac{n}{n+1})]^{nd}$. Thus, as each term in the sum is convex, and as the sum of convex functions is convex, the objective is convex. The constraints are all linear, hence this is a convex optimisation problem. ■

However, as a consequence of corollary 15, as $d \rightarrow \infty$ we know that $\rho \geq 0$, hence the minimum ρ will approach the ρ when using the non-informative extension (the non-informative extension has $\rho = 0$), thus there is little need to solve the optimisation problem after a sufficient number of dimensions is reached.

5.3 Experiments on University Ranking

The optimal imputation algorithm presented in section 5.2 is difficult to solve due to the integer constraints. We evaluated the performance of a relaxed version of the program, whereby the constraints are relaxed such that the variables may take a value in the range $[0, 1]$. To solve the relaxed problem, we used a BFGS based optimiser by shifting the equality constraints into the objective function with high penalties. Final ranks were determined by ranking item i in list j based on the score $\sum_k x_{i,j,k}$.

We evaluated this relaxed solution on imputing rankings for universities. To this end, the top-200 universities ranked by QS in 2014, Shanghai in 2014, and Times in 2015 were obtained. In aggregating these three lists, there are a total of 266 ranks that need to be imputed.

Measuring multivariate Spearman's ρ on all three lists imputing the missing elements using the non-informative extension gives $\rho = 0.632$. In comparison, the relaxed optimal imputation found a solution that obtained $\rho = 0.683$, a modest increase in the correlation. We also developed an interactive website¹ showing the detailed results for all universities, which also allows the user to alter the weights of each of the original experts. The top 36 aggregate rankings for the universities are given in appendix B.

1. <http://uni.cua0.org>

6. Supervised Learning to Rank

We now consider the task of learning rank aggregation from extreme ranks. Theorem 17 and definition 19 provide the core of our algorithm. Using theorem 17, we can find an average rank that aggregates a set of ranks, and by extending top- k and bottom- k ranks to a common domain, we can apply it to partially labelled data.

6.1 Weighted Mixture of Experts

As a result of theorem 17 we have a way of finding the ranking (according to some rank generator) that is closest to a set of ranks. Consider the learning problem where we have a ranking L which comprise our labels, and a set of d experts $\{R_j\}$. During training, we would like to find a weighting of the input rankings such that it gives the label. Given a target ranking L , we would like to optimise the weights ω ,

$$\max_{\omega} \rho_n(L, R_1^{\omega_1}, R_2^{\omega_2}, \dots, R_n^{\omega_n}).$$

Here we have introduced weights ω over each rank to control the influence of each rank over the final consensus rank; the intuition here is that ranks with $\omega_i > 1$ are replicated with more influence, which is easy to see when ω_i are natural numbers. For example, a weight of 2 would mean the ranked list has appeared twice in the calculation of the consensus rank. While it is convenient to have integer weights for interpretability, the weights ω could be any real number in general. In the following, we consider $\omega \in \mathbb{R}^n$. Instead of performing this high-dimensional optimisation, we decompose it into a pairwise (bivariate) comparison between the label L and the weighted geometric mean, where we now explicitly show the fact that the ranks are a function of the n objects x

$$\max_{\omega} \sum_x \rho_n(L(x), \sigma(R_1^{\omega_1} \otimes R_2^{\omega_2} \otimes \dots \otimes R_n^{\omega_n})(x)),$$

where the notation \otimes indicates the product operator. Observe that we have used theorem 17 to convert the d dimensional problem into the product of ranks R_j and the Spearman's correlation above is only two dimensional. For bivariate Spearman's ρ , this can be expressed in terms of the squared difference (1). We further assume that σ is the identity mapping to simplify the problem, giving us:

$$\min_{\omega} \sum_x (L(x) - R_1^{\omega_1}(x) R_2^{\omega_2}(x) \dots R_n^{\omega_n}(x))^2. \quad (16)$$

The objective (16) minimises the distance between the label ranks and the weighted expert ranks.

6.2 Least Squares Method on Logarithm of Ranks

Recall that we consider normalised ranks (divided by $n+1$). By using the logarithm identity, we convert the power scaling in (16) into a multiplicative scaling. Our algorithm is:

1. Extend incomplete ranks $\{R_i\}$ to $\{R_i^l\}$ by imputing the average missing value such that $\text{Dom } R_i^l = \text{Dom } L$;

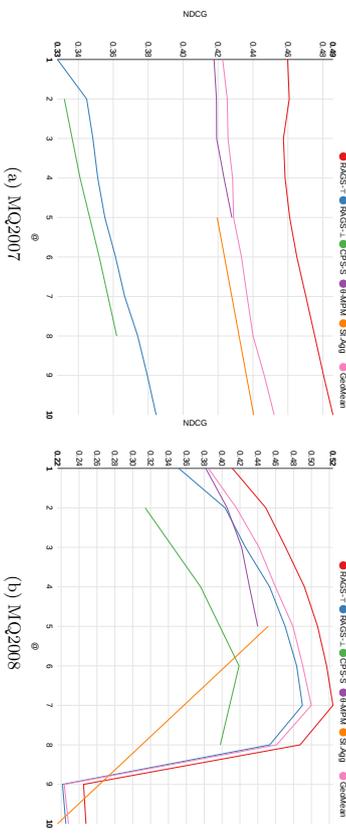


Figure 1: Results on MQ2007-aggr (a, left) and MQ2008-aggr (b, right): NDCG@k. Our method is labelled RAGS- and RAGS- corresponding to top and bottom non-informative imputation respectively. The results for CPS-S was the best reported in Qin et al. (2010a). The results of θ -MPM was the best among the reported results in Volkovs and Zennel (2012) from BordatCount, CPS, SVP, Bradley-Terry model, and Plackett-Luce model. The results of St-Agg was the best among the reported results in Nin et al. (2013) and was the best among MCLK, SVP, Plackett-Luce model, θ -MPM, BordatCount and RRF.

2. Convert to log-ranks $r_i^l = \log \circ R_i^l$ and $l = \log \circ L$;
3. Learn weights ω by minimising

$$\sum_x \left(l(x) - \sum_{j=1}^d \omega_j r_j^l(x) \right)^2, \quad \text{where the outer sum is over the } n \text{ examples } x.$$

A log transformation of the ranks is used as it naturally encodes the weights as a power scaling in the framework of theorem 17, i.e., the weighted consensus rank is given by $\prod_j r_j^l(x)^{\omega_j}$. Note that this is still solving (16) as we are optimising Spearman's ρ , which is sensitive only to ordering, and therefore though the final weights are different ρ is maximised via (1).

In the following experiments we also included a bias/offset term in the least squares problem, which can be interpreted as adding a ranking that is constant (gives all objects the same rank). It is interesting to note that the final step in this procedure is closely related to Borda Count, except our consensus rank is the geometric mean instead of the arithmetic mean. Since this is a least squares estimation problem, we directly use the closed form solution.

6.3 Benchmarking on LETOR 4.0

We tested our method on the MQ2007-aggr and MQ2008-aggr list aggregation benchmarks Qin et al. (2010b). The goal in these challenges is to aggregate 21 and 25 different rankers respec-

tively over a set of query-document pairs. Each data set has 5 pre-defined cross-validation folds with each fold providing a training, testing and validation data set (60%/20%/20%). We trained our model on the training set and tested on the testing set, leaving the validation set unused since we have no hyperparameters.

In the following we consider two types of experts: either experts $\{R_i\}$ are top- k experts, that is they only rank the best k samples from Ω , or experts are bottom- k experts, that is they identify the worst k samples from Ω . We call our proposed method RAGS- and RAGS- respectively. We assume that the ranked documents in the benchmark data sets are either top- k or bottom- k respectively, with potentially different numbers of documents k labelled by each expert. Ties are given the average rank of tied documents.

To evaluate the agreement, we use the standard evaluation tool from the LETOR website², which implements the Normalised Discounted Cumulative Gain (NDCG). In fig. 1a, we see that our approach RAGS- performs better than all other methods at any selection size on the MQ2007-agg data set. Indeed, we also perform better than Qin et al. (2010a) where the best result uses a coset-permutation distance based stagewise (CPS) model with Spearman's ρ in a probabilistic model. Recall that our approach considers the multivariate Spearman's ρ whereas Qin et al. (2010a) uses bivariate Spearman's ρ in a pairwise fashion. For MQ2008-agg (fig. 1b), again our approach performs better than all other methods.

To tease apart the effect of inputting missing ranks and the effect of weighting the experts, we compared our proposed method with and without training (uniform weights). GeoMean denotes the results for the geometric mean (uniform weights on the experts) after performing imputation assuming top- k ranking by the experts. First we observe that our proposed approach outperforms the geometric mean, which is a good sanity check. It is surprising that the geometric mean performs quite well in MQ2007. The major difference is that we are inputting the missing ranks, and the other methods suffer from assigning them to an arbitrary value. This demonstrates the importance of imputation.

6.4 Strictly Ordered Labels

One issue with the benchmark aggregation data set is that the labels are only $\{0,1,2\}$ relevance scores, and hence it is unclear exactly what the rankings are within the relevance classes. We create a new data set which is formed by taking the intersection between the documents retrieved by a particular query between MQ2007-agg and MQ2007-list. This new data set contains the strictly ordered labels from MQ2007-list, but uses the aggregation data from MQ2007-agg. The same procedure is used to create the corresponding data set for MQ2008-agg and MQ2008-list. These data sets are available for download at the LETOR website. We maintain exactly the same 5-fold cross validation splits and report our results in table 3.

Considering the results for Spearman's ρ , we observe that our learning method performs well. Note that the geometric mean outperforms Borda count on both data sets, which confirms that our theoretically justified model performs better than the heuristic model. It is interesting to observe that optimising for Spearman's ρ could result in a decrease in Kendall's τ . This demonstrates the importance of choosing the appropriate objective function for learning.

² <http://research.microsoft.com/letor>

Table 1: Results on MQ2007-agg: NDCG. Our method is labelled RAGS- and RAGS- corresponding to top and bottom non informative imputation respectively. The results for CPS-S was the best reported in Qin et al. (2010a). The results of θ -MPM was the best among the reported results in Volkovs and Zemel (2012) from BordaCount, CPS, SVP, Bradley-Terry model, and Plackett-Luce model. The results of St.Agg was the best among the reported results in Niu et al. (2013) and was the best among MCLK, SVP, Plackett-Luce model, θ -MPM, BordaCount and RRF.

Fold	@1	@2	@3	@4	@5	@6	@7	@8	@9	@10
RAGS-	0.45986	0.46078	0.45744	0.45838	0.46102	0.46512	0.4703	0.47538	0.48042	0.4858
RAGS-	0.32804	0.3448	0.34836	0.35114	0.3552	0.36132	0.36656	0.37402	0.37952	0.38458
CPS-S		0.332		0.341		0.352		0.362		
-MPM	0.4177	0.4191	0.4192	0.4234	0.4279					
St.Agg					0.4195					0.4404
GeoMean	0.42264	0.42528	0.42570	0.42834	0.42886	0.43342	0.43664	0.44004	0.44648	0.45216

Table 2: Results on MQ2008-agg: NDCG

Fold	@1	@2	@3	@4	@5	@6	@7	@8	@9	@10
RAGS-	0.41158	0.44898	0.47118	0.4922	0.50696	0.51706	0.52416	0.48732	0.24498	0.24768
RAGS-	0.35156	0.40338	0.42624	0.45326	0.4706	0.48352	0.48994	0.45336	0.22138	0.22514
CPS-S		0.314		0.376		0.419		0.398		
-MPM	0.3817	0.4057	0.4219	0.4307	0.4399					
St.Agg					0.4515					0.2157
GeoMean	0.38470	0.41600	0.44142	0.45976	0.47938	0.49042	0.49986	0.46108	0.22334	0.22812

Table 3: Results on MQ2007-agglst and MQ2008-agglst. The left column shows the results for multivariate Spearman's ρ and the right column shows the result for Kendall's τ .

Method	MQ2007-agglst		MQ2008-agglst	
	ρ	τ	ρ	τ
RAGS-	0.4394	0.6201	0.7235	0.6931
RAGS-	0.2992	0.2488	0.6349	0.5560
GeoMean	0.2457	0.3011	0.5777	0.6578
Borda	0.2217	0.1790	0.5519	0.5869

7. Discussion and Conclusion

We propose an approach for learning weights between experts for the task of rank aggregation. By generalising the derivation of concordance functions, we obtain an expression for multivariate Spearman's ρ . Furthermore, we show that the geometric mean of the expert ranks is the optimal aggregator under Spearman's correlation. Motivated by this, our method solves a least squares estimation problem for logarithmic normalised ranks to find optimal weights.

One possible extension of our work is to compute the correlation for all possible subsets of rankings. While corollary 15 shows that the overall correlation cannot be negative as the number of rankings increase, there may be subgroups which are positively correlated within groups but negatively correlated between groups. By computing the correlation on the power set, we could use a clustering method to find such subgroups.

Though we have focused on ρ_d^+ , our results are equally applicable to ρ_d^- ; indeed it is a simple reversal of ranks that give ρ_d^- . The choice between ρ_d^+ and ρ_d^- is thus problem dependent: for tasks where being ranked highly is more informative ρ_d^+ is a better choice; conversely ρ_d^- is more suitable for tasks where being ranked lowly is more informative.

In contrast to other rank aggregation approaches, our method is very computationally efficient. However, the core of our method requires a complete set of rankings and hence does not handle missing variables. To resolve this, we propose three imputation methods (unbiased, optimistic, pessimistic) for completing top- k ranked lists that allows us to apply Spearman's ρ to aggregate ranks from partial lists. Our method is thus applicable for large scale applications with top- k rankings that arise in areas such as text mining and bioinformatics. One subtlety is that imputation from top- k should not be confused with the choice of using ρ_d^+ , which is an separate design choice.

Surprisingly, our weighted geometric mean shows state of the art results on benchmark data sets, without the need for tuning hyperparameters or expensive computation. The simplicity of our model makes it easier to interpret, and the weights give a direct estimate of the influence of each expert. This problem has wide applications to ensemble learning, voting, text mining, recommender systems and bioinformatics.

Acknowledgments

This work was completed when both authors were employed by NICTA. NICTA was funded by the Australian Government through the Department of Communications and the Australian Research Council through the ICT Centre of Excellence Program.

Appendix A. Bivariate Spearman's ρ and Squared Distance

This well known result³ shows that Spearman's ρ can be expressed in terms of the squared distance between ranks.

In the following derivation, we use the expressions for the sum of integers and the sum of squares of integers:

$$\sum_{k=1}^n k_i = \frac{n(n+1)}{2} \quad \sum_{k=1}^n k_i^2 = \frac{n(n+1)(2n+1)}{6}.$$

Recall that Spearman's ρ is defined (1) as:

$$\rho_n = \frac{\sum_x (R(x) - \bar{R})(S(x) - \bar{S})}{\sqrt{\sum_x (R(x) - \bar{R})^2 \sum_x (S(x) - \bar{S})^2}}.$$

Since there are no ties, both $R(x)$ and $S(x)$ consist of integers from 1 to n inclusive, and the two squared sums in the denominator are the same. Recall that the mean rank is

$$\bar{R} = \bar{S} = \frac{n+1}{2}, \quad \text{and} \quad \sum_x R(x) = \frac{n(n+1)}{2} = n\bar{R}.$$

Therefore, the denominator can be expressed as a function of n :

$$\begin{aligned} \sqrt{\sum_x (R(x) - \bar{R})^2 \sum_x (S(x) - \bar{S})^2} &= \sum_x (R(x) - \bar{R})^2 \\ &= \sum_x (R(x)^2 - 2R(x)\bar{R} + \bar{R}^2) \\ &= \sum_x R(x)^2 - 2\bar{R} \sum_x R(x) + n\bar{R}^2 \\ &= \sum_x R(x)^2 - n\bar{R}^2 \\ &= \frac{n(n+1)(2n+1)}{6} - n \left(\frac{n+1}{2} \right)^2 \\ &= n(n+1) \left(\frac{2n+1}{6} - \frac{n+1}{4} \right) \\ &= n(n+1) \left(\frac{n-1}{12} \right) \\ &= \frac{n(n^2-1)}{12}. \end{aligned}$$

3. http://en.wikipedia.org/wiki/Spearman's_rank_correlation_coefficient accessed on 20 May 2014

Since both $R(x)$ and $S(x)$ consists of the same integers, we can express the squared difference in terms of the product.

$$\begin{aligned} \sum_x \frac{1}{2}(R(x) - S(x))^2 &= \sum_x \frac{1}{2}(R(x)^2 - 2R(x)S(x) + S(x)^2) \\ &= \sum_x \frac{1}{2}(R(x)^2 + S(x)^2) - \sum_x R(x)S(x) \\ &= \sum_x R(x)^2 - \sum_x R(x)S(x), \end{aligned}$$

where the first term is a function of n .

We express the product of the means, which appears in the numerator later, to match the denominator:

$$\begin{aligned} n \binom{n+1}{2}^2 &= \frac{n(n+1)}{12} 3(n+1) \\ &= -\frac{n(n+1)}{12} [(n-1) - (4n+2)] \\ &= -\frac{n(n+1)(n-1)}{12} + \frac{n(n+1)(2n+1)}{6} \\ &= -\frac{n(n^2-1)}{12} + \sum_x R(x)^2, \end{aligned}$$

where the last term in the sum is the expression for the sum of squares. We can now derive the expression for the numerator:

$$\begin{aligned} \sum_x (R(x) - \bar{R})(S(x) - \bar{S}) &= \sum_x R(x)S(x) - \bar{R} \sum_x S(x) - \bar{S} \sum_x R(x) + n\bar{R}\bar{S} \\ &= \sum_x R(x)S(x) - n\bar{R}\bar{S} \\ &= \sum_x R(x)S(x) - n \binom{n+1}{2}^2 \\ &= \sum_x R(x)S(x) + \frac{n(n^2-1)}{12} - \sum_x R(x)^2 \\ &= \frac{n(n^2-1)}{12} - \sum_x \frac{1}{2}(R(x) - S(x))^2, \end{aligned}$$

where the last line uses the expression of the sum of squared differences above. Putting together the expressions for the numerator and denominator together gives the desired result:

$$\rho_n = 1 - \frac{6 \sum_x (R(x) - S(x))^2}{n(n^2 - 1)}.$$

Appendix B. University Aggregate Rankings

Rank	University
1	Harvard University
2	Massachusetts Institute of Technology
3	Stanford University
4	California Institute of Technology
5	University of Cambridge
6	University of Oxford
7	Princeton University
8	University of Chicago
9	University of California, Berkeley
10	Imperial College London
11	ETH Zurich
12	University College London
13	Yale University
14	Columbia University
15	Johns Hopkins University
16	Cornell University
17	University of California, Los Angeles
18	University of Pennsylvania
19	University of Michigan
20	University of Toronto
21	Duke University
22	Northwestern University
23	University of Edinburgh
24	University of California, San Diego
25	King's College London
26	University of Washington
27	University of Tokyo
28	National University of Singapore
29	New York University
30	École Polytechnique Fédérale de Lausanne
31	McGill University
32	University of Melbourne
33	University of Illinois at Urbana-Champaign
34	University of Wisconsin-Madison
35	University of British Columbia
36	University of Manchester
37	Australian National University

References

- Justin Bedó, David Rawlinson, Benjamin Goudey, and Cheng Soon Ong. Stability of bivariate GWAS biomarker detection. *PLoS ONE*, 9:e93319, 04 2014.
- Douglas E. Critchlow. *Metric Methods for Analyzing Partially Ranked Data*. Springer-Verlag, 1985.
- Persi Diaconis. *Group representations in probability and statistics*, volume 11 of *Lecture Notes – Monograph Series*. Institute of Mathematical Statistics, 1988.
- Fabrizio Durante and Carlo Sempi. Copula theory: An introduction. In *Copula Theory and Its Applications*, pages 3–31, 2010.
- Gal Elidan. Copulas in machine learning. In *Copulae in Mathematical and Quantitative Finance*, 2013.
- M.A. Fligner and J.S. Verducci. Distance based ranking models. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 359–369, 1986.
- Christian Genest and Anne-Catherine Favre. Everything you always wanted to know about copula modeling but were afraid to ask. *Journal of Hydrologic Engineering*, 12(4):347–368, 2007.
- Rishabh Iyer and Jeff Bilmes. The submodular Bregman and Lovász-Bregman divergences with applications. In *NIPS*, 2012.
- Rishabh Iyer and Jeff Bilmes. The Lovász-Bregman divergence and connections to rank aggregation, clustering and web ranking. In *UAI*, 2013.
- Harry Joe. Multivariate concordance. *Journal of Multivariate Analysis*, 35:12–30, 1990.
- Harry Joe. *Dependence Modeling with Copulas*. CRC Press, 2014.
- Alexandre Klementiev, Dan Roth, and Kevin Small. Unsupervised rank aggregation with distance-based models. In *International Conference on Machine Learning*, 2008.
- Guy Lebanon and Yi Mao. Non-parametric modeling of partially ranked data. *Journal of Machine Learning Research*, 9:2401–2429, 2008.
- E.L. Lehmann. Some concepts of dependence. *Annals of Mathematical Statistics*, 37: 1137–1153, 1966.
- Tie-Yan Liu. *Learning to Rank for Information Retrieval*. Springer-Verlag, 2011.
- Geoff Macintyre, Antonio Jimeno Yepes, Cheng Soon Ong, and Karin Verspoor. Associating disease-related genetic variants in intergenic regions to the genes they impact. *PeerJ*, 2:e639, 2014.
- C. L. Mallows. Non-null ranking models. *Biometrika*, 44(1):114–130, 1957.
- Roger B. Nelsen. Nonparametric measures of multivariate association. *Distributions with Fixed Marginals and Related Topics*, 28:223–232, 1996.
- Roger B. Nelsen. *An Introduction to Copulas*. Springer, second edition, 2006.
- Shuzi Niu, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. Top-k learning to rank: Labeling, ranking and evaluation. In *SIGIR*, pages 751–760, 2012. ISBN 9781450314725.
- Shuzi Niu, Yanyan Lan, Jiafeng Guo, and Xueqi Cheng. Stochastic rank aggregation. In *UAI*, 2013.
- Tao Qin, Xiubo Geng, and Tie-Yan Liu. A new probabilistic model for rank aggregation. In *NIPS*, 2010a.
- Tao Qin, Tie-Yan Liu, Jun Xu, and Hang Li. LETOR: A benchmark collection for research on learning to rank for information retrieval. *Information Retrieval Journal*, 2010b.
- Friedrich Schmid and Rafael Schmidt. Multivariate extensions of spearman's rho and related statistics. *Statistics & Probability Letters*, 77:407–416, 2007.
- Friedrich Schmid, Rafael Schmidt, Thomas Blumentritt, Sandra Gaisser, and Martin Ruppert. Copula-based measures of multivariate association. In *Copula Theory and Its Applications*, pages 209–236, 2010.
- D. Sculley. Combined regression and ranking. In *KDD*, 2010.
- C. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904.
- Michael. D. Taylor. Multivariate measures of concordance. *Annals of the Institute of Statistical Mathematics*, 59(4):789–806, 2007.
- Pravin K. Trivedi and David M. Zimmer. Copula modeling: An introduction for practitioners. *Foundations and Trends in Econometrics*, 1(1):1–111, 2005.
- Mannel Úbeda Flores. Multivariate versions of blomqvist's beta and spearman's footrule. *Annals of the Institute of Statistics and Mathematics*, 57(4):781–788, 2005.
- Maksims N. Volkovs and Richard S. Zemel. A flexible generative model for preference aggregation. In *WWW*, 2012.

Nonparametric Network Models for Link Prediction

Sinead A. Williamson

Department of Statistics and Data Science/

Department of Information, Risk and Operations Management,

University of Austin at Texas

SINEAD.WILLIAMSON@MCCOMB.UTEXAS.EDU

Editor: Edo Airolidi

Abstract

Many data sets can be represented as a sequence of interactions between entities—for example communications between individuals in a social network, protein-protein interactions or DNA-protein interactions in a biological context, or vehicles' journeys between cities. In these contexts, there is often interest in making predictions about future interactions, such as who will message whom.

A popular approach to network modeling in a Bayesian context is to assume that the observed interactions can be explained in terms of some latent structure. For example, traffic patterns might be explained by the size and importance of cities, and social network interactions might be explained by the social groups and interests of individuals. Unfortunately, while elucidating this structure can be useful, it often does not directly translate into an effective predictive tool. Further, many existing approaches are not appropriate for sparse networks, a class that includes many interesting real-world situations.

In this paper, we develop models for sparse networks that combine structure elucidation with predictive performance. We use a Bayesian nonparametric approach, which allows us to predict interactions with entities outside our training set, and allows the both the latent dimensionality of the model and the number of nodes in the network to grow in expectation as we see more data. We demonstrate that we can capture latent structure while maintaining predictive power, and discuss possible extensions.

Keywords: Dirichlet process, networks, Bayesian nonparametrics, Gibbs sampling, hierarchical modeling

1. Introduction

We are often interested in characterizing and predicting the interactions between objects, be they individuals within an organization, proteins within a cell, or transportation hubs within a region. We can represent these objects as nodes in a network, with the non-zero edges of the network describing the interactions between nodes. For example, we can represent a social network as a binary network, where each node corresponds to an individual, and an edge between nodes corresponds to a friendship between individuals. Patterns of email communication can be modeled using an integer-valued network, with integer-valued edges representing the number of emails sent from one individual to another. Interactions between proteins can be represented using a real-valued network, where the nodes correspond to proteins and the edges correspond to interaction strength.

A number of statistical models for such networks have been proposed. Many of these models fall under the *stochastic blockmodel* (SB) framework (Holland et al., 1983; Wang

and Wong, 1987; Suijders and Nowicki, 1997), where each node is assumed to belong to one of K latent groups, and the interaction between two nodes depends only on their group assignments. This basic model can be extended by allowing the number of latent groups to be unbounded, as in the infinite relational model (IRM, Kemp et al., 2006), or by allowing each node to exhibit membership in multiple latent groups, as in the mixed membership stochastic blockmodel (MMSB, Airolidi et al., 2008). One thing that these models have in common is that they treat nodes as exchangeable, and assume that there exists a fixed, stationary network between these nodes. Each node is represented by the totality of its interactions with other nodes, and we use this information to cluster (or, in the case of the MMSB, co-cluster) the data into distinct groups.

In this paper we follow a different approach: we treat the interactions, rather than the nodes, as data points, and construct an exchangeable sequence of directed binary links. Each link corresponds to a single interaction—such as “friending” or “liking” in a social network, or sending a single email—and is characterized in terms of an ordered pair of nodes. We may observe multiple links between two nodes; this corresponds to repeated interactions (for example, sending multiple emails).

This approach has a number of advantages. Unlike the stochastic blockmodel family, the approach described in this paper allows us to model sparse graphs, where the number of non-zero entries grows as $O(M)$, where M is the number of nodes. Sparsity is a property of many real-life networks, which tend to exhibit small-world behavior (Caron and Fox, 2015; Orbanz and Roy, 2014).

Another advantage is that our model is explicitly designed for the prediction task. In many scenarios, we might be interested in what the next interaction will be: who will email whom, for example. Stochastic blockmodels aim to model a fully observed network, where the absence of an observed edge is interpreted as an explicitly observed zero. In this setting, any predictions must directly contradict these observed zeros. While it is possible to explicitly mark edges as “missing”, we can only do this for a small subset of unobserved edges—if we assume all zero edges in a stochastic blockmodel are in fact unobserved, the maximum likelihood network will have all the edges equal to one. Conversely, by constructing an integer-valued network via an exchangeable sequence of links, we frame our problem in a manner that directly provides a predictive distribution over the location of the next link, and allows us to continuously update our posterior predictive distribution in the face of new data. Further, by choosing to place a nonparametric distribution over the sequence of links, we can easily incorporate previously unseen nodes, without any prior knowledge of the number of such nodes.

A further advantage is seen in the computational complexity of the model. Under a stochastic blockmodel with M nodes and K clusters, the computational cost of evaluating the likelihood of the i th node belonging to the k th cluster grows as $O(M)$, meaning that the overall computational cost of inferring the cluster allocations, without resorting to approximations, scales as $O(M^2K)$. Conversely, under the proposed model, the cluster likelihood for a link involves only the two nodes associated with the interaction, yielding $O(N)$ computational complexity where N is the number of links. If N grows significantly slower than M^2 —as is the case if we assume sparsity—this will lead to computational savings as our data set grows.

This paper begins by examining existing Bayesian network models in Section 2, before going on to present our model in Section 3. We begin by introducing the Dirichlet network distribution in Section 3.1, before proceeding to describe how a mixture of such distributions can create a flexible network model with interesting latent structure in Section 3.2. While the focus of this paper is on integer-valued networks, we also consider extensions to the binary case in Section 3.3. While the primary method we consider does not exhibit exchangeable links, we can sample an auxiliary integer-value network and leverage its exchangeability to obtain predictive distributions. In Section 4, we describe an MCMC sampler for the mixture of Dirichlet network distributions, before presenting experimental results in Section 5.

1.1 Notation

We will use the notation Z to represent an $M \times M$ network, with elements $z_{sr} \in \mathbb{N}$ indicating the relationship between nodes s and r . If $z_{sr} \in \{0, 1\}$, then a non-zero value indicates the presence of a relationship. If z_{sr} is allowed to take on arbitrary non-negative integer values, we take this to indicate the number of interactions (for example, emails in a social network, packages in a computer network) between nodes s and r . Unless otherwise specified, we will assume Z to be a directed network, where $Z \neq Z^T$.

It will sometimes be more convenient to represent the matrix Z as a sequence of interactions $Y = y_1, y_2, \dots$ where each interaction y_i consists of an ordered pair of nodes. We can reconstruct the matrix Z by setting $z_{sr} = \sum_i \mathbb{I}(y_i = (s, r))$, where $\mathbb{I}(\cdot)$ represents an indicator function, that returns one iff the statement it refers to is true.

2. Related Work

A number of models have been proposed for modeling interactions within a network. In a Bayesian context, most of these models fall under the general category of stochastic blockmodels (SBs), a class of clustering-based models that generate dense networks. We discuss stochastic blockmodels in Section 2.1.

Recently, there has been growing interest in models for sparse networks, where the number of edges grows linearly (rather than quadratically) with the number of nodes. We discuss relevant work in this area in Section 2.2.

2.1 Stochastic Blockmodels and Related Models

The basic stochastic blockmodel (Holland et al., 1983; Wang and Wong, 1987) posits that each node belongs to one of K latent clusters. For each pair (i, j) of clusters there is a cluster-specific distribution over interactions governed by some parameter $\theta_{i,j}$; conditioned on their cluster allocations c_s and c_r , interactions between nodes s and r are i.i.d. samples from the distribution parametrized by θ_{c_s, c_r} . Typically, this is a Bernoulli distribution, giving rise to a binary matrix; however Maradjasson et al. (2010) show that the basic idea can be extended to real- or integer-valued networks by using different choices of distribution (for example, Gaussian or Poisson).

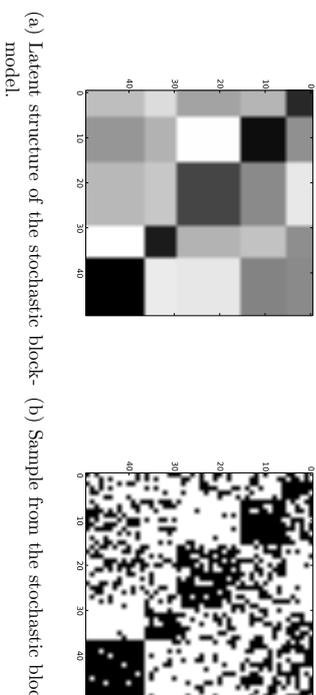


Figure 1: A stochastic blockmodel with 5 clusters.

To position SBs in a Bayesian context (Snijders and Nowicki, 1997), we can place appropriate conjugate priors on the cluster parameters $\theta_{i,j}$ and the cluster proportions. For example, a Bayesian version of the standard Bernoulli stochastic blockmodel takes the form

$$\begin{aligned} \pi &\sim \text{Dirichlet}(\phi) \\ \theta_{i,j} &\sim \text{Beta}(\alpha, \beta), \quad i, j \in \{1, \dots, K\} \\ c_s &\sim \text{Discrete}(\pi), \quad s \in \{1, \dots, M\} \\ z_{sr} &\sim \text{Bernoulli}(\theta_{c_s, c_r}). \end{aligned}$$

Figure 1a shows the underlying partitioning of the nodes, with the color of each partition indicating the corresponding value of θ_{c_s, c_r} in a binary SB. Figure 1b shows an instantiation of a network generated from this structure.

The basic stochastic blockmodel can be extended in a number of ways. The infinite relational model (IRM; Kemp et al., 2006) allows an unbounded number of latent clusters by placing a Dirichlet process prior over the cluster probabilities; this removes the need to pre-specify a latent dimensionality and allows the number of clusters to grow (in expectation) with the number of nodes. A more general class of models is obtained if we associate each pair of nodes with a location in some metric space, and place a continuous or piecewise-continuous parameter function over this space (Lloyd et al., 2012); this class of model includes the SB and the IRM.

The SB and the IRM both assume that each node belongs to a single cluster. The mixed-membership stochastic blockmodel (MMSB, Airoldi et al., 2008) relaxes this assumption by associating each node with a *distribution* over latent clusters. To generate the interaction between nodes s and r , each node selects a cluster from their individual distributions over clusters; the interaction is then generated according to the distribution associated with this pair. This extension allows the model to capture the fact that individuals may perform multiple “roles” leading to different patterns of interaction: Ian may be friends with Janelle because they both play tennis, and friends with Keira because they both study computer science.

While SBs are well-suited to community detection, they are less appropriate for the task of predicting unseen interactions, i.e., asking questions such as “who will Ian interact with next?”. Stochastic blockmodels, and related models, explicitly model the entire network, with the likelihood for a data point’s cluster allocation(s) depending on its interactions with all M nodes in the network. Unless explicitly marked as missing, zeros in the network indicate the observed absence of a relationship, and affect the likelihood. As a result, predictions about the locations of unseen interactions must directly contradict the zeros present in the training set. This cannot be avoided by marking all zeros as unobserved, since in this case the maximum likelihood network is maximally connected.

Further, if we explicitly model the absence of interactions, the model likelihood is changed if we discover the existence of an $(M + 1)$ st node who hasn’t yet interacted with anyone—since we must explicitly cluster this node and include its interactions in other nodes’ likelihood terms. Therefore, if we want to allow prediction of links to or from individuals not included in our training set, we must know in advance the number of such individuals. This is not realistic in many settings: we do not know how many people will join a social network in the future.

Another consequence of modeling both non-zero- and zero-valued edges is that the computational cost of evaluating the conditional probability that node i belongs to cluster k scales linearly with the number of nodes, since

$$P(c_s = k | c_{-s}, Z, \pi, \{\theta_{i,j}\}) \propto \pi_k \prod_{r=1}^M P(z_{rs} | \theta_{k,c_r}) P(z_{rs} | \theta_{c_r,k}).$$

Therefore, resampling the cluster allocations of all M cluster allocations scales quadratically with M (and linearly with K). As M grows, Gibbs sampling (Suijders and Nowicki, 1997) quickly becomes computationally infeasible. Variational methods (Mariadassou et al., 2010; Airoldi et al., 2008) generally give faster inference (albeit at the cost of lower estimate quality); however they still scale quadratically in the number of nodes. A number of approximate methods have been proposed that reduce the computational cost by approximating the full likelihood (Amini et al., 2013; Ho et al., 2012); however as with variational methods these approaches are no longer asymptotically guaranteed to sample from the true model.

2.2 Statistical Models for Sparse Networks

A limitation of stochastic blockmodel-type approaches—and indeed the more general class of models discussed in Lloyd et al. (2012)—is that they yield dense models almost surely (Orbanz and Roy, 2014). In other words, the number of non-zero entries in the resulting network grows as $O(M^2)$. This follows from the structure shown in Figure 1a: each partition is a simple Erdős-Rényi $G(n, p)$ model, with all possible relationships being equally likely, and the number of non-zero relationships growing in expectation with the number of pairs of nodes in that partition.

This contrasts with the sparse nature of many real-life social networks, where the number of non-zero entries grows as $O(M)$, since the number of interactions a person makes does not grow proportionately with the size of the network. In general, an individual will only interact with a small subset of the total population. Caron and Fox (2015) have shown that

several real-world networks, including the ENRON data set explored in this paper, have a very high probability of exhibiting this form of sparsity.

Caron and Fox (2015) presented a construction for sparse networks based on a Poisson process. An integer-valued network is represented as a discrete measure $Z = \sum_{n=1}^{\infty} z_n \delta_{s_n, r_n}$ on \mathbb{R}^2 , where each atom’s size z_n indicates the edge value, and the location (s_n, r_n) specifies a pair of nodes. The atoms are distributed according to a Poisson process, with base measure given by the outer product of two generalized gamma process (GGP)-distributed random measures (Brix, 1999), i.e.

$$\begin{aligned} W &\sim \text{GGP}(\rho, \lambda) \\ Z &\sim \text{PP}(W \times W). \end{aligned} \tag{1}$$

This construction can also be used to generate a binary network, by thresholding the integer-valued network N . Caron and Fox (2015) demonstrate that the construction in Equation 1 can be used to generate networks that are sparse in terms of the number of edges, and exhibit power-law degree distribution. These are both properties that are commonly found in real world networks.

A link between the sparse binary models of Caron and Fox (2015) and the stochastic blockmodel family has recently been made explicit by Veitch and Roy (2015). Under their “graphex” construction for random binary networks, a candidate set of nodes, and associated node-specific parameters, is selected via a Poisson process on \mathbb{R}^2 . As with the SB, the probability of a link between two nodes is governed by the nodes’ parameter values via an appropriate link function, meaning that the SB is a member of this graphex-based class of models. However, different choices of link models can yield sparse graphs including the binary model of Caron and Fox (2015).

3. Nonparametric Models for Networks

As we saw in Section 2, stochastic blockmodels assume a fixed, fully observed network, where zero-valued entries are taken to represent the observed absence of an interaction, and model the network by clustering these nodes. We take a different approach: We model a network as a sequence of observed interactions, and aim to predict the locations of future interactions by explicitly clustering the interactions, rather than the nodes.

To do so, we consider distributions over a sequence of links connecting a set of nodes. Each link, therefore, is associated with an (ordered) pair of nodes sampled from some distribution over such pairs; we may have multiple links associated with a given pair. To allow the network to expand over time, and to facilitate out-of-sample prediction, we let this set of nodes be countably infinite and use a Bayesian nonparametric distribution to assign probabilities to potential pairs.

The main focus of this paper is on integer-valued networks—we will use the running example of an email network—where there can be multiple links between the same pair of nodes. While not explored in as much depth, we also suggest modifications that allow us to model binary networks in Section 3.3.

3.1 Dirichlet Network Distributions

A simple way of constructing an integer-valued network with an unbounded number of nodes is to place a probability distribution G over a countably infinite number of actors. We can represent such a network as a sequence of (sender, receiver) pairs; each pair might, for example, correspond to a single email from a sender to a receiver, or a single journey between two cities. The value of a (directed) edge from a “sender” s to a “receiver” r is the number of times we have seen the pair (s, r) . We call each individual pair in the sequence a link; the value of an edge between two nodes is the number of links between them.

To generate such a pair, we simply sample a sender and a receiver according to G . Let N be the total number of links in our network—that is, the total sum of the edge values. An appropriate prior over G might be the Dirichlet process, so that

$$\begin{aligned} G &\sim \text{DP}(\tau, \Theta) \\ s_n, r_n &\stackrel{i.i.d.}{\sim} G, \quad n = 1, \dots, N \\ z_{ij}^{(N)} &= \sum_{n=1}^N \mathbb{I}(s_n = i, r_n = j). \end{aligned} \quad (2)$$

In other words, we generate a sequence of links by sampling with replacement from the distribution implied by the product measure $G \times G$. We will refer to this construction as a symmetric Dirichlet network distribution (DND). Figure 2a shows a network constructed in this manner.

This model is related to the sparse network model proposed by Caron and Fox (2015) and described in Section 2.2. As we described in Equation 1, Caron and Fox generate a directed, integer-valued graph Z by sampling interactions according to a Poisson process, with rate given by the product measure $W \times W$ where $W \sim \text{GGP}$. If W has finite total mass $W(\Omega)$, then this can equivalently be described as:

$$\begin{aligned} W &\sim \text{GGP}(\rho, \lambda) \\ N &\sim \text{Poisson}(W(\Omega)^2) \\ s_n, r_n &\stackrel{i.i.d.}{\sim} \frac{W}{W(\Omega)}, \quad n = 1, \dots, N \\ Z &= \sum_{n=1}^N \delta_{(s_n, r_n)}. \end{aligned} \quad (3)$$

If we choose a standard gamma process as the random measure W in Equation 3 then, conditioned on the total number of links N , we recover the symmetric nonparametric model described in Equation 2. In this paper, we focus on the gamma process/Dirichlet process case in order to achieve simple inference strategies; however the models proposed in this section can easily be extended to use a normalized generalized gamma process (Lijoi et al., 2008), or some other random probability measure such as a Pitman-Yor process (Pitman and Yor, 1997), in place of the Dirichlet process.

This basic model assumes that the probability of a node being the “sender” of a link is the same as the probability of being a “receiver”. In practice, this is often not a reasonable

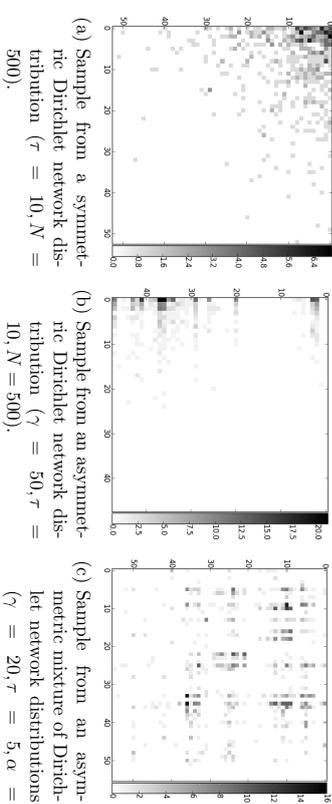


Figure 2: Samples from Dirichlet network distributions and mixtures of Dirichlet network distributions.

assumption. For example, in a university’s email network, administrators may send out a large number of group emails—that is, they often operate in the “sender” role—but receive a relatively small number of emails. When modeling human migration patterns, the United States had over 13 times as many immigrants as emigrants in 2015, whereas Micronesia had around twice as many emigrants as immigrants (United Nations, Department of Economic and Social Affairs, Population Division, 2015). We can capture this form of asymmetry by replacing the single distribution G over nodes (Equation 2) with a pair of distributions, A and B . To ensure the two distributions have the same support (meaning that any node can have both incoming and outgoing links), we couple these two distributions via a shared, discrete base measure H . The generalized process becomes

$$\begin{aligned} H &:= \sum_{i=1}^{\infty} h_i \delta_{\theta_i} \sim \text{DP}(\gamma, \Theta) \\ A &:= \sum_{i=1}^{\infty} a_i \delta_{\theta_i} \sim \text{DP}(\tau, H) \\ B &:= \sum_{i=1}^{\infty} b_i \delta_{\theta_i} \sim \text{DP}(\tau, H) \\ s_n &\sim A, \quad n = 1, \dots, N \\ r_n &\sim B, \quad n = 1, \dots, N \\ z_{ij}^{(N)} &= \sum_{n=1}^N \mathbb{I}(s_n = i, r_n = j). \end{aligned} \quad (4)$$

Figure 2b shows a network constructed in this manner; note the nodes that “send” a large number of links are not necessarily those that “receive” a large number of links. The concentration parameter τ governs how similar the two distributions are to the base measure, and hence to each other.

On their own, like the sparse models explored by Caron and Fox (2015), the symmetric and asymmetric Dirichlet network distributions allow very little internal structure. While we do see some preferential attachment due to the discrete nature of the underlying random measures, there is no clustering structure—if we know that an email was sent by a given

sender, this tells us nothing about the receiver. This makes it a poor model for real-world networks, where we observe cliques of users who are strongly interconnected, or groups that interact with other groups in characterizable manners. While we cannot capture such behavior with the basic symmetric or asymmetric Dirichlet network distributions described above, we can use them as a component of more complex and flexible models.

3.2 Mixtures of Dirichlet Network Distributions

Rather than use a single Dirichlet network distribution over integer-valued networks, as described by Equations 2 and 4, we can use a *mixture* of such distributions, which we will refer to as a mixture of Dirichlet network distributions, or MDND. By ensuring both sender and receiver belong to a common mixture component, we break the independence between sender and receiver, allowing us to identify communication patterns that cannot be captured using the DND or the related Caron and Fox (2015) model. To allow links between the subgraphs associated with each mixture component, we couple the networks using a shared, discrete base measure H . Concretely, in the asymmetric case, let

$$\begin{aligned} D &:= (d_k, k \in \mathbb{N}) \sim \text{GEM}(\alpha) \\ H &:= \sum_{i=1}^{\infty} h_i \delta_{\theta_i} \sim \text{DP}(\gamma, \Theta) \\ A_k &:= \sum_{i=1}^{\infty} a_{k,i} \delta_{\theta_i} \sim \text{DP}(\tau, H), \quad k = 1, 2, \dots \\ B_k &:= \sum_{i=1}^{\infty} b_{k,i} \delta_{\theta_i} \sim \text{DP}(\tau, H) \end{aligned} \quad (5)$$

where $\text{GEM}(\alpha)$ is the distribution over the size-biased atom sizes of a Dirichlet process with concentration parameter α . Figure 2c shows a network constructed in this manner. A symmetric version of the MDND is recovered if we replace the sender- and receiver-specific distributions A_k and B_k with a shared distribution $G_k \sim \text{DP}(\tau, H)$, and is appropriate when we believe the distribution over edges originating from a given node is similar to the distribution over edges ending at that node. For the remainder of this paper, we will focus on the asymmetric setting.

We can verbalize the generative process of the asymmetric MDND as follows. To generate the n th link, we first select a cluster c_n . We then select a “sender” s_n and a “receiver” r_n —identifying a link (s_n, r_n) —according to the cluster-specific distributions A_{c_n} and B_{c_n} . The concentration parameters α , τ and γ can be manipulated to obtain differing network properties. The parameter α controls the number of clusters, with the total number of clusters used to model N links growing approximately as $O(\gamma \log N)$. The parameter τ controls the degree of similarity between the clusters: As τ decreases, the overlap between clusters will tend to decrease. Increasing γ increases the overall number of nodes represented in the network.

Since the pairs are sampled i.i.d. given the random measures, the resulting sequence is exchangeable, meaning the construction is appropriate for sequences of links where there

is no specific ordering of the links, or where the order is believed to be irrelevant. This has useful implications for inference: It means we can easily construct a Gibbs sampler, as described in Section 4. However, in many networks the order in which links are formed does carry information. In Section 3.3, we discuss an extension for explicitly ordered binary links, and in Section 6, we will discuss possible extensions of the integer-valued network model to explicitly ordered links.

3.3 Extension: Binary-valued Networks

Many real-world networks exhibit binary, rather than real-valued, edges. One way of capturing this behavior is to threshold the integer-valued edges generated by the DND or the MDND. The most straightforward version of this is simply to sample a network $Z = (z_{ij})$ according to the DND or the MDND, and then generate a binary network $Y = (y_{ij})$ by letting $y_{ij} = 1$ iff $z_{ij} > 0$. If we place a negative binomial distribution over $N = \sum_{i,j} z_{ij}$, so that $N \sim \text{NB}(\alpha, 1/(1+\beta))$ for some $\beta > 0$, we can represent this thresholded model as

$$\begin{aligned} \Gamma &:= \sum_{k=1}^{\infty} \mu_k \delta_{\theta_k} \sim \text{GP}(\alpha D_0, \beta) & s_n^{(k)} &\sim G_k, \quad n = 1, \dots, N_k \\ & & r_n^{(k)} &\sim G_k \\ G_0 &\sim \text{DP}(\gamma, H_0) & z_j^{(N)} &= \sum_{k=1}^{\infty} \sum_{n=1}^{N_k} \mathbb{I}(s_n^{(k)} = j, r_n^{(k)} = j) \\ G_k &\sim \text{DP}(\tau, G_0), \quad k = 1, 2, \dots & N &:= \sum_k N_k \\ N_k &\sim \text{Poisson}(\mu_k) & y_{ij}^{(N)} &= \mathbb{I}(z_{ij}^{(N)} > 0), \end{aligned} \quad (6)$$

where GP indicates a gamma process. This approach is a form of *restricted exchangeable distribution*, as described by Williamson et al. (2013). As in the unrestricted, integer-valued network, the distribution over edge events is exchangeable. We note that this truncation technique is the same as that used by Caron and Fox (2015) to generate binary networks, and indeed if Z is the symmetric version of the DND given by Equation 3, and if we obtain a symmetric network by mirroring the link counts, then this construction corresponds to the binary network described in Caron and Fox (2015), itself a special case of the class of binary network models described by Veitch and Roy (2015).

If our observations are explicitly ordered, and we believe that ordering to be important, we can modify the DND or the MDND to sample *without* replacement from the set of possible links, giving a non-exchangeable model. This form of non-exchangeability mimics behavior found in many naturally-occurring networks. For example, in a social network, a user will add their close friends first, and then over time add more distant acquaintances. In integer-valued networks exchangeability can represent the fact that close friends will communicate both early and often, but in an exchangeable binary setting all relationships appear identical.

The resulting binary network model is mathematically equivalent to a *censored* DND or MDND, where we only observe the first instance of a link between nodes i and j :

$$\begin{aligned}
D &:= (d_k, k \in \mathbb{N}) \sim \text{GEM}(\alpha) \\
H &:= \sum_{i=1}^{\infty} h_i \delta_{\theta_i} \sim \text{DP}(\gamma_i, \Theta) \\
A_k &:= \sum_{i=1}^{\infty} a_{k,i} \delta_{\theta_i} \sim \text{DP}(\tau, H), \quad k = 1, 2, \dots \\
B_k &:= \sum_{i=1}^{\infty} b_{k,i} \delta_{\theta_i} \sim \text{DP}(\tau, H)
\end{aligned}$$

$$\begin{aligned}
c_t &\sim D, \quad t = 1, 2, \dots \\
s_t &\sim A_{c_t} \\
r_t &\sim B_{c_t} \\
y_{ij}^{(t)} &= \mathbb{I}\left(\sum_{t'=1}^t \mathbb{I}(s_{t'} = i, r_{t'} = j) > 0\right).
\end{aligned}$$

Due to the finite probability of sampling an existing (s, r) pair, $Y^{(t+1)}$ may be the same as $Y^{(t)}$; instead we would likely work with the corresponding non-repeating sequence $(Z^{(n)}, n \in \mathbb{N})$, where $Z^{(n)} = \min_{t'} \left(Y^{(t')} : \sum_{t'=1}^{t'} \mathbb{I}(Y^{(t')} \neq Y^{(t-1)}) = n \right)$. While this censored model is no longer exchangeable, we can make use of the underlying exchangeable sequence of interactions to make predictions.

3.4 Relationship to Other Models

As we described in Section 3.1, the Dirichlet network distribution is strongly related to the sparse network models of Caron and Fox (2015)—in fact, conditioned on the total number of links, the integer-valued model of Caron and Fox (2015) is a special case of the symmetric DND, and the truncated model of Equation 6 describes the binary model of Caron and Fox (2015) as a special case. However, the DND on its own lacks the flexibility to model structured networks, where nodes tend to belong to locally connected sub-networks, and where knowing who sent a message tells us something about the intended recipient. The mixture of Dirichlet network distributions allows us to capture multiple sub-networks, while allowing interaction between sub-networks via a common Dirichlet process-distributed base measure H .

A related integer-valued network model is described by Grane and Dempsey (2016). This model is explicitly designed for multi-way interactions, such as collaborations or actors co-starring in movies, but can be modified to give two-way interactions. The number of “roles” in an interaction is sampled from an appropriate distribution, and for each role, nodes are sampled from a (single) Pitman–Yor process. The two-way interaction setting corresponds to a Pitman–Yor variant of the symmetric Dirichlet network distribution obtained by replacing the Dirichlet process in Equation 2 with a Pitman–Yor process; as such it is unable to capture the clustering behavior obtained using the mixture of Dirichlet network distributions.

The models described in this paper also bear some similarity to stochastic blockmodels, which were described in Section 2. The main difference between the stochastic blockmodel family and the models proposed in this paper is that, under the blockmodel paradigm, nodes are clustered into a (potentially) infinite, in the case of the IRM) number of clusters. Conversely, the MDND directly clusters links, rather than nodes, and represents a network as a (potentially) infinite sequence of pairs of nodes. This creates a natural framework for questions of prediction. For the basic symmetric DND described in Section 3.1, the predictive distribution over the next pair of nodes is available in analytic form via an urn

representation. For the mixture models proposed in Sections 3.2 and 3.3, the predictive distribution depends on the values of latent variables, but we can easily sample from this distribution, as we will see in Section 4.

While the MDND does not explicitly cluster the nodes, we can obtain a similar mixed-membership interpretation to that found in the MMSB. We can think of each cluster of links representing a latent topic of conversation between nodes. Each topic of conversation is described by distributions over the nodes likely to take part in such a conversation. If we condition on the fact that the s th node is sending an email, we can use these distributions to infer the probability of that email belonging to a given discussion. Since the hierarchical construction of Equation 5 ensures that the topics of conversation have overlapping participants, the node will be associated with a conditional distribution over an unbounded number of conversations.

In Section 2.2, we discussed how the graphex construction for binary exchangeable networks by Veitch and Roy (2015) allows us to represent the stochastic blockmodel and the sparse binary model of Caron and Fox (2015) using a common framework. While the integer-valued MDND, and the non-exchangeable binary network described in Section 3.3, do not fall under the graphex framework, it suggests an alternative way to describe the exchangeable binary network obtained by thresholding a MDND (as described in Equation 6).

4. Fully Nonparametric Inference via an Urn Scheme

In the simple symmetric network model of Equation 2, we can directly evaluate the predictive distribution over the n th link, given the previous $n - 1$ links, via a straightforward extension of the Polya urn sampler for the Chinese restaurant process Neal (1998), where the probability of seeing a link between two nodes is proportional to the product of those nodes’ degrees (excluding the link in question):

$$P(y_n = (s, r) | y_1, \dots, y_{n-1}) = \begin{cases} \frac{m_{s, (m_s + \mathbb{I}(s=j))}}{(2n-2+\tau)(2n-1+\tau)} & \text{if } m_i \neq 0, m_j \neq 0 \\ \frac{m_i m_j}{(2n-2+\tau)(2n-1+\tau)} & \text{if } m_i \neq 0, m_j = 0 \\ \frac{m_i \tau}{(2n-2+\tau)(2n-1+\tau)} & \text{if } m_i = 0, m_j \neq 0 \\ \frac{\tau + \mathbb{I}(s=r)}{(2n-2+\tau)(2n-1+\tau)} & \text{if } m_i = 0, m_j = 0, \end{cases}$$

where $m_i = \sum_{n=1}^{n-1} \mathbb{I}(s_n = i) + \mathbb{I}(r_n = i)$ is the sum of the links to or from node i .

The MDND is based on a mixture of coupled hierarchical Dirichlet processes, allowing us to construct a collapsed Gibbs sampler by modifying the direct assignment sampler for the hierarchical Dirichlet process introduced by Teh et al. (2006). Recall that associated with each cluster k we have a sender-specific distribution A_k and a receiver-specific distribution B_k .¹ Let $\eta_k = \sum_{i=1}^N I_{c_i=k}$ be the number of links associated with cluster k ; let $m_{k,i}^{(1)}$ be the number of edges associated with cluster k that originate from node i (that is, edges where node i is the “sender”); and let $m_{k,i}^{(2)}$ be the number of edges associated with cluster k that end at node i (that is, edges where node i is the “receiver”). We also introduce auxiliary

1. In this section, we focus on the asymmetric MDND, where there are separate distributions over senders and receivers; extension to the symmetric case is straightforward.

count variables $\rho_{k,i}^{(1)}$ and $\rho_{k,i}^{(2)}$ and a probability vector $(\beta_1, \dots, \beta_J, \beta_u) \sim \text{Dir}(\rho_1^{(1)}, \dots, \rho_J^{(1)}, \gamma)$, where $\rho_i = \sum_k \rho_{k,i}^{(1)} + \rho_{k,i}^{(2)}$; here β_1, \dots, β_J correspond to the atoms of H that are associated with represented nodes, and $\beta_u = \sum_{j=J+1}^{\infty} h_j$.

The distribution over the cluster assignment of the n th link, given β and all other $N-1$ links, is given by:

$$P(c_n = k | s_n, r_n, c^{-n}, \beta) \propto \begin{cases} \eta_k^{-n} (m_{k,s_n}^{(1)-n} + \tau \beta_{s_n}) (m_{k,r_n}^{(2)-n} + \tau \beta_{r_n}) & \text{if } \eta_k^{-n} > 0 \\ \alpha \tau^2 \beta_{s_n} \beta_{r_n} & \text{if } \eta_k^{-n} = 0. \end{cases} \quad (7)$$

where we use c^{-n} to indicate the sequence $(c_{n'} : n' \neq n)$ and $m_{k,i}^{(1)-n} = \sum_{c_{n'} \neq n} I_{c_{n'}=k, s_{n'}=i}$, i.e., the $-n$ notation is used to exclude the value associated with the current observation.

Following Fox et al. (2007) we can sample the ‘‘dish counts’’ $\rho_{k,i}^{(1)}$ ($\rho_{k,i}^{(2)}$) by simulating the partitioning of $m_{k,i}^{(1)}$ ($m_{k,i}^{(2)}$) according to a Chinese restaurant process with parameter $\tau \beta_k$.

Conditioned on the cluster assignments for the first N links, and the probability vector $(\beta_1, \dots, \beta_J, \beta_u)$, we can evaluate the predictive distribution over the $N+1$ st link as:

$$P(y_{N+1} = (s, r) | c_{1:N}, y_{1:N}, \beta) = \begin{cases} \sum_{k=1}^{K+} \frac{\eta_k}{N+\alpha} \frac{m_{k,s}^{(1)+\tau\beta_s} m_{k,r}^{(2)+\tau\beta_r}}{\eta_k + \tau} + \frac{\alpha}{N+\alpha} \beta_s \beta_r & \text{if } s, r \leq J \\ \sum_{k=1}^{K+} \frac{\eta_k}{N+\alpha} \frac{m_{k,s}^{(1)+\tau\beta_s} \beta_r}{\eta_k + \tau} + \frac{\alpha}{N+\alpha} \beta_s \beta_u & \text{if } s \leq J, r > J \\ \sum_{k=1}^{K+} \frac{\eta_k}{N+\alpha} \beta_u \frac{m_{k,r}^{(2)+\tau\beta_r}}{\eta_k + \tau} + \frac{\alpha}{N+\alpha} \beta_u \beta_r & \text{if } r \leq J, s > J \\ \beta_u^2 & \text{if } r, s > J. \end{cases}$$

To improve mixing, we augmented the sampler with split/merge moves, proposed using the Restricted Gibbs method of Jain and Neal (2004).

5. Experimental Evaluation

We begin by demonstrating the ability of the MDND to recover latent network structure in a two synthetically-generated data sets, before performing a quantitative analysis on three real-world networks.

5.1 Synthetic Data

In Figure 3, we show the performance of the MDND evaluated on a 60-node network generated according to a stochastic blockmodel. A distribution over cluster memberships was drawn from a Dirichlet(5, 5, 5, 5, 5) distribution. Intra-cluster link parameters $\theta_{i,i}$ were distributed according to $\theta_{i,i} \sim \text{Gamma}(5, 1)$, and inter-cluster link parameters $\theta_{i,j}$ were distributed according to $\theta_{i,j} \sim \text{Gamma}(0.5, 1)$. Link counts z_{sr} for each pair (s, r) were Poisson-distributed given the appropriate parameter θ_{c_s, c_r} , where c_s is the cluster of

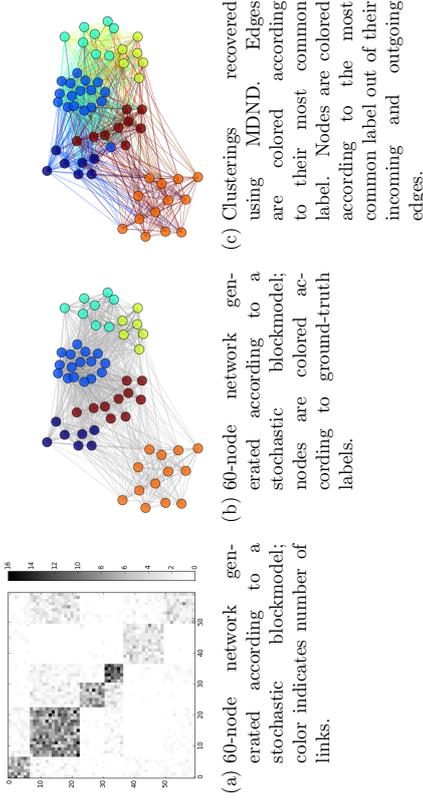


Figure 3: Structure recovery: stochastic blockmodel.

node s . The raw matrix of link counts is shown in Figure 3a, and a visualization of the matrix is shown in Figure 3b, where the edge color indicates the number of nodes and the node color indicates the ground-truth cluster labels.³

Figure 3c shows the structure recovered by the MDND. Each edge in the MDND representation has multiple cluster labels, one per link; the edges are colored according to their most common cluster label. The nodes are colored according to the most common cluster label amongst their incoming and outgoing edges. As we can see from Figure 3c, all but one node is most commonly associated with its ground-truth label.

Figure 4 evaluates the MDND on a 50-node with manually constructed overlapping blocks. The network shown in Figure 4a was generated from 5 equiprobable clusters; each cluster puts 95% of its probability mass on one of five overlapping blocks, and the remaining mass is uniformly distributed over the entire population. Figure 4b shows a visualization of this matrix, where the edge color indicates the number of nodes and the node color indicates most common ground-truth label amongst the incoming and outgoing edges, as in Figure 3b. Figure 4c shows the structure recovered by the MDND; as before, the edges are colored according to their most common cluster label, and the nodes are colored according to the most common cluster label amongst their incoming and outgoing edges. Again, we have very high agreement between the ground-truth labels and the recovered clustering.

5.2 Real Network Data

We compared the mixture of Dirichlet network distributions to the infinite relational model, the mixed membership stochastic blockmodel, and several baselines, in three real-world scenarios: A small network representing character interactions in Shakespeare’s ‘Macbeth’;

3. The visualization layouts in this section (Figures 3b, 3c, 4b and 4c) were generated using the Python package NetworkX with a Fruchterman-Reingold force-directed algorithm.

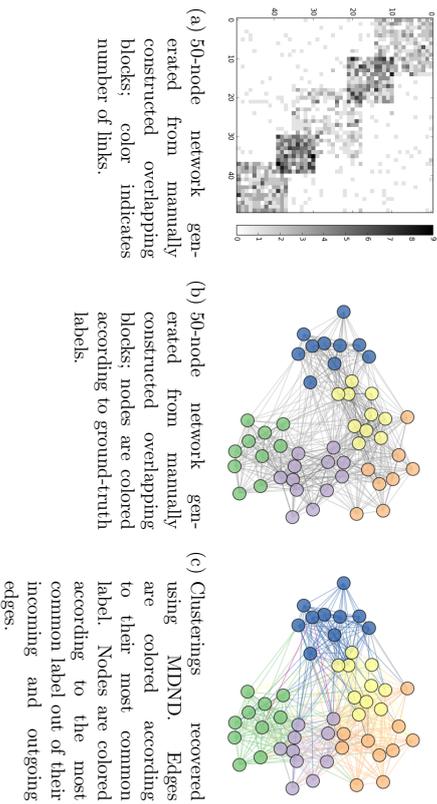


Figure 4: Structure recovery: Overlapping blocks.

a medium-sized network representing political interactions; and a large network representing email interactions. In Section 5.2.1 we describe the comparison methods, and in Section 5.2.2 we describe the three networks and present our results.

5.2.1 COMPARISON METHODS

We compare the mixture of Dirichlet network distributions to a single symmetric Dirichlet network distribution; to integer-valued variants of the mixed-membership stochastic block-model (MMSB, Airoldi et al., 2008) and the infinite relational model (IRM, Kemp et al., 2006); and to two baseline methods.

1. **Symmetric Dirichlet network distribution.** We modeled the data using a single symmetric DND as described in Section 3.1, with Dirichlet process concentration parameter $\tau = 1$.

2. **Infinite relational model.** Kemp et al. (2006) describe a variant of the IRM appropriate for integer-valued data. Each pair of clusters (i, j) is associated with a positive real-valued parameter θ_{ij} , and the N links are assigned to clusters according to a multinomial distribution parameterized by the θ_{ij} . Inference in this model is performed using existing C code released by the authors of Kemp et al. (2006). This code was not able to handle the number of nodes present in the Enron data sets.

3. **Mixed-membership stochastic blockmodel.** While the MMSB is designed for binary-valued networks, it can trivially be extended to integer-valued networks by replacing the Bernoulli distributions with Poisson distributions, and placing gamma priors on the Poisson parameters. While, to the best of our knowledge, this extension has not yet been explored in the literature, it is a natural, and easily implementable,

extension. We perform inference in this model, with $K = 50$ clusters, using Gibbs sampling, via an existing R package (Chang, 2012) that was modified to replace the beta/Bernoulli pairs with gamma/Poisson pairs. Since inference in this model was significantly slower than inference in the IRM and the DNM, we only compared with the gamma/Poisson MMSB on our smallest data set.

4. **Baseline 1: equiprobable links.** Our first baseline assumed that all (sender, receiver) pairs are equally likely, provided the sender and receiver are different—so if we have M nodes, the probability of a given link is $1/M(M-1)$.

5. **Baseline 2: Dirichlet-multinomial distribution over links.** Here we assumed an $M(M-1)$ -dimensional Dirichlet prior over the potential links, and looked at the conditional distribution given the N observed links,

$$P((s_{N+1}, r_{N+1}) = (i, j)) = \frac{\sum_{m=1}^N \mathbb{I}(s_m = i, r_m = j) + \alpha_{ij}}{M(M-1) + N}.$$

We note that this corresponds to an IRM where each pair of nodes is in its own cluster.

5.2.2 EVALUATION ON THREE REAL NETWORKS

We compared the mixture of Dirichlet network distributions to the comparison methods described above, on three real data sets. We describe the data sets below; Table 1 summarizes the networks' statistics.

1. **Macbeth.** This data set represents the implied social network in Shakespeare's 'Macbeth'. We constructed a directed network where each link indicates an uninterrupted block of speech from the speaker to all other characters presently on stage. To evaluate predictive performance, we split the play into 5 contiguous subsets, each containing (approximately) $N/5$ consecutive edges. We used these subsets to generate a 5-fold split into training data (4 of the 5 subsets) and a test set (the remaining one subset), and evaluated the joint predictive likelihood on the test set.

2. **Militarized disputes.** Next, we evaluated our model on a network representing militarized disputes between 188 countries (Maoz, 2005). The data set contains 8650 disputes between 1816 and 1976. We split this data set into 10 subsets, and used these subsets to generate 10 train/test splits, where 9 subsets were used for training and 1 for evaluation.

3. **ENRON email network.** Finally, we looked at subsets of the ENRON email data set (Klimmt and Yang, 2004). We looked at five training sets, corresponding to the total set of emails sent and received in each of the first 5 months of 2000. For each data set, we evaluated predictive performance on the first 1000 emails sent in the subsequent month.

Table 2 shows the log predictive likelihoods obtained on the above data sets. In each case, we condition on the latent structure obtained on the training set, and consider the joint conditional distribution over the test set. For the MMSB and the IRM, our training

Network	Number of links (N)	Number of nodes (M)	N/M^2
Macbeth	2153	39	1.42
Military Disputes	8650	188	0.245
Enron (average)	13994	3883.8	$9.14e-4$
Enron (range)	8692–20464	3006–4652	$7.55e-4$ – $1.03e-3$

Table 1: Network statistics. Note that the Macbeth and Military Disputes data sets were split into equally sized subsets, while the Enron data set was split according to month. We therefore report both the average and the range of the Enron monthly statistics.

	Macbeth	Military Disputes	ENRON
Symmetric DND	-2900.56 ± 193.06	-5019.35 ± 38.70	-15030.77 ± 208.71
MDND	-1769.74 ± 71.60	-5000.74 ± 54.64	-9053.68 ± 235.02
IRM	-1941.50 ± 102.69	-6984.25 ± 16.49	-
MMSB	-3077.22 ± 65.42	-	-
Baseline 1	-2723.33 ± 0	-5748.78 ± 40.59	-16509.75 ± 125.47
Baseline 2	-2462.94 ± 70.05	-5433.24 ± 38.76	-15800.01 ± 138.84

Table 2: Test set log likelihood (mean \pm standard error).

set explicitly included those nodes with interactions present in the test set but not in the training set. We note that this is an unrealistic setting that gives an advantage to the MMSB and the IRM: In most situations, the number of new nodes is unknown. However, without including at least the correct number of unseen nodes, we would be unable to obtain an estimate for the predictive performance on the data sets used.

For the IRM and the MMSB, we already have cluster assignments for each pair of nodes, and can use the cluster parameters to directly obtain a probability distribution over pairs of nodes. We use this probability distribution to directly calculate the joint predictive log likelihood. For the MDND, we do not yet have cluster assignments for the test set links. It is analytically intractable to sum over all possible test set cluster assignments. Instead, we estimate the predictive log likelihood using our Gibbs sampler to generate 100 samples of the test cluster assignments, conditioned on the training set assignments and the test set links. We then use the harmonic mean of the likelihoods given these cluster assignments as an estimator for the overall joint predictive likelihoods.

The MMSB code was unable to run on the Disputes and ENRON data sets, and the IRM code was unable to run on the ENRON data set. By looking at the ratio M/N^2 in Table 1, we see that the Disputes data set is much sparser than Macbeth, and the ENRON data set is much sparser than Macbeth. This means that the blockmodel-based approaches, which scale quadratically with the number of nodes, are unable to run. We note that our experiments were based on of existing implementations of the MMSB and IRM; while different implementations may be able to process the entire data set, it is likely to be much slower than the MDND, due to the density of the network.

We see that, in each case, the MDND out-performs the comparison methods—even though we have included more information in the MMSB and the IRM by making use of the number of unseen nodes.

6. Discussion and Future Work

We have presented a new Bayesian nonparametric model, the mixture of Dirichlet network distributions, for integer-valued networks where the number of nodes is unbounded and grows in expectation with the number of binary links. This model allows us to capture sparse networks with latent structure. Existing network models focus either on latent structure—capturing the fact that each node will have a different pattern over which nodes it connects with—or on capturing sparsity; this is, to our knowledge, the first model that combines these two goals. Further, unlike most existing Bayesian network models, this model is explicitly designed for prediction. We can use the mixture of Dirichlet network distributions to obtain an explicit predictive distribution over the nodes associated with an as-yet unseen observation, even if we have not observed these nodes in our training set; we have shown good predictive and qualitative performance on a variety of data sets.

The mixture of Dirichlet network distributions is based on a simpler network model that we refer to as a Dirichlet network distribution. In the symmetric setting—where a common distribution is used for both senders and receivers—this corresponds to a special case of the integer-valued network models of Caron and Fox (2015) and Crane and Dempsey (2016). While these models can be used to obtain desirable properties such as network sparsity and power law degree distribution, they are unable to capture community-type structure in the network. By using a mixture of these networks, we can capture multiple modalities of interaction between nodes; by using a nonparametric hierarchical framework we ensure that both the number of nodes is unbounded, and that nodes can interact as part of multiple clusters. The MDND therefore increases the modeling flexibility of this class of models, while retaining desirable sparsity properties.

The mixture of Dirichlet network distributions is an exchangeable model: It is invariant to permutations of the order in which we observe links. While this is computationally appealing and leads to a straightforward predictive distribution, it does not allow us to capture network dynamics in integer-valued networks. In practice, such dynamics may be important: an individual’s level of activity within a topic may vary over time, and the overall popularity of topics may change. A number of authors have found that adding temporal dynamics to network models improves performance (Ishiguro et al., 2010; King et al., 2010; Xu and Hero III, 2013). In the case of Dirichlet network distributions, similar temporal dynamics could be incorporated by replacing some or all of the component Dirichlet processes with *dependent* Dirichlet processes (MacEachern, 2000; Lin et al., 2010; Ren et al., 2008); we intend to explore this in a future work.

In addition to the base model for integer-valued networks, we also discussed extensions to binary networks. The methods considered involve truncating the exchangeable integer-valued network; while it is possible to obtain exchangeable binary networks related to those considered by Caron and Fox (2015) and Veitch and Roy (2015), we argued that a dynamic truncation, yielding a non-exchangeable model, is more appropriate for a temporally expanding binary network. Unfortunately, inference in such truncated models is trickier

than the integer-valued case. While we can analytically sample the censored observations as auxiliary variables and recover the integer network, the number of censored observations grows according to a coupon-collector problem with the number of observed links, making this approach infeasible for large data sets. An interesting avenue for future research is to develop scalable inference methods for this setting.

Acknowledgments

Sinead Williamson is supported by NSF grant 1447721.

References

- C. Kemp, J.B. Tenenbaum, T.L. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *National Conference on Artificial Intelligence (AAAI)*, pages 381–388, 2006.
- B. Klimmt and Y. Yang. Introducing the Enron corpus. In *Conference on Email and Anti-Spam (CEAS)*, 2004.
- A. Lijoi, I. Prünster, and S.G. Walker. Investigating nonparametric priors with Gibbs structure. *Statistica Sinica*, 18(4):1653–1668, 2008.
- D. Lin, E. Grimson, and J.W. Fisher III. Construction of dependent Dirichlet processes based on Poisson processes. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1396–1404, 2010.
- J. Lloyd, P. Orbanz, Z. Ghahramani, and D.M. Roy. Random function priors for exchangeable arrays with applications to graphs and relational data. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1007–1015, 2012.
- S.N. MacEachern. Dependent Dirichlet processes. *Unpublished manuscript, Department of Statistics, The Ohio State University*, 2000.
- Z. Maoz. Dyadic militarized interstate dispute dataset, version 2.0. <http://psfaculty.ucdavis.edu/zmaoz/dyadmid.html>, 2005.
- M. Marinčič, S. Robin, and C. Vaucher. Uncovering latent structure in valued graphs: A variational approach. *Annals of Applied Statistics*, 4(2):715–742, 2010.
- R.M. Neal. Markov chain sampling methods for Dirichlet process mixture models. Technical Report 9815, Dept. of Statistics, University of Toronto, 1998.
- P. Orbanz and D. Roy. Bayesian models of graphs, arrays and other exchangeable random structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):437–461, 2014.
- J. Pitman and M. Yor. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, 25(2):855–900, 1997.
- L. Ren, D.B. Dunson, and L. Carin. The dynamic hierarchical Dirichlet process. In *International Conference on Machine Learning (ICML)*, pages 824–831, 2008.
- T.A.B. Snijders and T. Nowicki. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14(1):75–100, 1997.
- Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.
- United Nations, Department of Economic and Social Affairs, Population Division. Trends in international migrant stock: The 2015 revision. United Nations database, POP/DB/MIG/Stock/Rev.2015, 2015.
- E.M. Airoldi, D.M. Blei, S.E. Fienberg, and E.P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014, 2008.
- A.A. Amini, A. Chen, P.J. Bickel, and E. Levina. Pseudo-likelihood methods for community detection in large sparse networks. *The Annals of Statistics*, 41(4):2097–2122, 2013.
- A. Brix. Generalized gamma measures and shot-noise Cox processes. *Advances in Applied Probability*, 31(4):929–953, 1999.
- F. Caron and E.B. Fox. Sparse graphs using exchangeable random measures. arXiv:1401.1137 [stat.ME], 2015.
- J. Chang. *lda: Collapsed Gibbs sampling methods for topic models*, 2012. URL <http://CRAN.R-project.org/package=lda>. R package version 1.3.2.
- H. Crane and W. Dempsey. Edge exchangeable models for network data. arXiv:1603.04571 [math.ST], 2016.
- E.B. Fox, E.B. Sudderth, M.I. Jordan, and A.S. Willsky. The sticky HDP-HMM: Bayesian nonparametric hidden Markov models with persistent states. Technical Report P-2777, Massachusetts Institute of Technology, 2007.
- Q. Ho, J. Yin, and E.P. Xing. On triangular versus edge representations—towards scalable modeling of networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2132–2140, 2012.
- P.W. Holland, K.B. Laskey, and S. Leinhardt. Stochastic blockmodels: First steps. *Social networks*, 5(2):109–137, 1983.
- K. Ishiguro, T. Iwata, N. Ueda, and J.B. Tenenbaum. Dynamic infinite relational model for time-varying relational data analysis. In *Advances in Neural Information Processing Systems (NIPS)*, pages 919–927, 2010.
- S. Jain and R.M. Neal. A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13(1):158–182, 2004.

- V. Veitch and D. Roy. The class of random graphs arising from exchangeable random measures. arXiv:1512.03099 [math.ST], 2015.
- Y.J. Wang and G.Y. Wong. Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, 82(397):8–19, 1987.
- S.A. Williamson, S.N. MacEachern, and E.P. Xing. Restricting nonparametric distributions. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2598–2606, 2013.
- E.P. Xing, W. Fu, and L. Song. A state-space mixed membership blockmodel for dynamic network tomography. *The Annals of Applied Statistics*, 4(2):535–566, 2010.
- K.S. Xu and A.O. Hero III. Dynamic stochastic blockmodels: Statistical models for time-evolving networks. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction*, pages 201–210. Springer, 2013.

Guarding against Spurious Discoveries in High Dimensions

Jianqing Fan

*Department of Operations Research and Financial Engineering
Princeton University
Princeton, NJ 08544, USA*

jqfan@PRINCETON.EDU

Wen-Xin Zhou

*Department of Operations Research and Financial Engineering
Princeton University
Princeton, NJ 08544, USA*

wenxinz@PRINCETON.EDU

Editor: Hui Zou

Abstract

Many data mining and statistical machine learning algorithms have been developed to select a subset of covariates to associate with a response variable. Spurious discoveries can easily arise in high-dimensional data analysis due to enormous possibilities of such selections. How can we know statistically our discoveries better than those by chance? In this paper, we define a measure of goodness of spurious fit, which shows how good a response variable can be fitted by an optimally selected subset of covariates under the null model, and propose a simple and effective LAMM algorithm to compute it. It coincides with the maximum spurious correlation for linear models and can be regarded as a generalized maximum spurious correlation. We derive the asymptotic distribution of such goodness of spurious fit for generalized linear models and L_1 regression. Such an asymptotic distribution depends on the sample size, ambient dimension, the number of variables used in the fit, and the covariance information. It can be consistently estimated by multiplier bootstrapping and used as a benchmark to guard against spurious discoveries. It can also be applied to model selection, which considers only candidate models with goodness of fits better than those by spurious fits. The theory and method are convincingly illustrated by simulated examples and an application to the binary outcomes from German Neuroblastoma Trials.

Keywords: Bootstrap, Gaussian approximation, generalized linear models, L_1 regression, model selection, sparsity, spurious correlation, spurious fit

1. Introduction

Technological developments in science and engineering lead to collections of massive amounts of high-dimensional data. Scientific advances have become more and more data-driven, and researchers have been making efforts to understand the contemporary large-scale and complex data. Among these efforts, variable selection plays a pivotal role in high-dimensional statistical modeling, where the goal is to extract a small set of explanatory variables that are associated with given responses such as biological, clinical, and societal outcomes. Toward this end, in the past two decades, statisticians have developed many data learning methods and algorithms, and have applied them to solve problems arising from diverse fields of sciences, engineering and humanities, ranging from genomics, neurosciences and health

sciences to economics, finance and machine learning. For an overview, see Bühlmann and van de Geer (2011) and Hastie, Tibshirani and Wainwright (2015).

Linear regression is often used to investigate the relationship between a response variable Y and explanatory variables $\mathbf{X} = (X_1, \dots, X_p)^T$. In the high-dimensional linear model $Y = \mathbf{X}^T \boldsymbol{\beta}^* + \varepsilon$, the coefficient $\boldsymbol{\beta}^*$ is assumed to be sparse with support $S_0 = \text{supp}(\boldsymbol{\beta}^*)$. Variable selection techniques such as the forward stepwise regression, the Lasso (Tibshirani, 1996) and folded concave penalized least squares (Fan and Li, 2001; Zou and Li, 2008) are frequently used. However, it has been recently noted in Fan, Guo and Hao (2012) that high dimensionality introduces large spurious correlations between response and unrelated covariates, which may lead to wrong statistical inference and false scientific discoveries. As an illustration, Fan, Shao and Zhou (2015) considered a real data example using the gene expression data from the international ‘HapMap’ project (Thorisson et al., 2005). There, the sample correlation between the observed and post-Lasso fitted responses is as large as 0.92. While conventionally it is a common belief that a correlation of 0.92 between the response and a fit is noteworthy, in high-dimensional scenarios, this intuition may no longer be true. In fact, even if the response and all the covariates are scientifically independent in the sense that $\boldsymbol{\beta}^* = \mathbf{0}$, simply by chance, some covariates will appear to be highly correlated with the response. As a result, the findings obtained via any variable selection techniques are hardly impressive unless they are proven to be better than by chance. To simplify terminology, in this paper we say that the discovery (by a variable selection method) is spurious if it is no better than by chance.

To guard against spurious discoveries, one naturally asks how good a response can be fitted by optimally selected subsets of covariates, even when the response variable and the covariates are not causally related to each other, that is, when they are independent. Such a measure of the goodness of spurious fit (GOSF) is a random variable whose distribution can provide a benchmark to gauge whether the discoveries by statistical machine learning methods any better than a spurious fit (chance). Measuring such a goodness of spurious fit and estimating its theoretical distributions are the aims of this paper. This problem arises from not only high-dimensional linear models and generalized linear models, but also robust regression and other statistical model fitting. To formally measure the degree of spurious fit, Fan, Shao and Zhou (2015) derived the distributions of maximum spurious correlations, which provide a benchmark to assess the strength of the spurious associations (between response and independent covariates) and to judge whether discoveries by a certain variable selection technique are any better than by chance.

The response, however, is not always a quantitative value. Instead, it is often binary; for example, positive or negative, presence or absence and success or failure. In this regard, generalized linear models (GLIM) serve as a flexible parametric approach to modeling the relationship between explanatory and response variables (McCullagh and Nelder, 1989). Prototypical examples include linear, logistic and Poisson regression models which are frequently encountered in practice.

In GLIM, the relationship between the response and covariates is more complicated and cannot be effectively measured via Pearson correlation coefficient, which is essentially a measure of the linear correlation between two variables. We need to extend the concept of spurious correlation or the measure of goodness of spurious fit to more general models and study its null distribution. A natural measure of goodness of fit is the likelihood

ratio statistic, denoted by $\mathcal{LR}_n(s, p)$, where n is the sample size and s is size of optimally fitted model. It measures the goodness of spurious fit when \mathbf{X} and Y are independent. This generalization is consistent with the spurious correlation studied in Fan, Shao and Zhou (2015), that is, applying $\mathcal{LR}_n(s, p)$ to linear regression yields the maximum spurious correlation. We plan to study the limiting null distribution of $2\mathcal{LR}_n(s, p)$ under various scenarios. This reference distribution then serves as a benchmark to determine whether the discoveries are spurious.

To gain further insights, let us illustrate the issue by using the gene expression profiles for 10,707 genes from 251 patients in the German Neuroblastoma Trials NB90-NB2004 (Oberthuer et al., 2006). The response labeled as “3-year event-free survival” (3-year EFS) is a binary outcome indicating whether each patient survived 3 years after the diagnosis of neuroblastoma. Excluding five outlier arrays, there are 246 subjects (101 females and 145 males) with 3-year EFS information available. Among them, 56 are positives and 190 are negatives. We apply Lasso using the logistic regression model with tuning parameter selected via ten-fold cross validation (40 genes are selected). The fitted likelihood ratio $2\mathcal{LR} = 211.96$. To judge the credibility of the finding of these 40 genes, we should compare the value 211.96 with the distribution of the Goodness Of Spurious Fit (GOSF) $2\mathcal{LR}_n(s, p)$ when \mathbf{X} and Y are indeed independent, where $n = 246$, $p = 10,707$ and $s = 40$. This requires some new methodology and technical work. Figure 1 shows the distribution of the GOSF estimated by our proposed method below and indicates how abnormal the value 211.96 is. It can be concluded that the goodness of fit to the binary outcome is not statistically significantly better than GOSF.

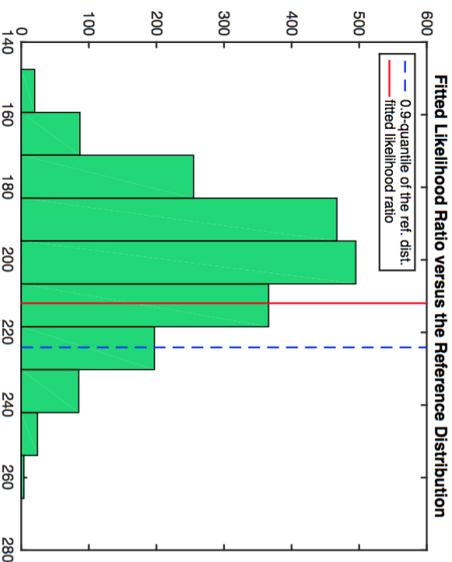


Figure 1: Lasso fitted likelihood ratio $2\mathcal{LR}$ in comparison to the distribution of GOSF $2\mathcal{LR}_n(s, p)$ with $n = 246$, $p = 10,707$ and $s = 40$.

The above result shows that the 10-fold cross-validation chooses a too large model with 40 variables. This prompts us to reduce the model sizes along the Lasso path such that their fits are better than GOSF. The results are reported in Table 2. The largest model along the LASSO path that fits better than GOSF has model size 17. We can use the cross-validation to select a model with model size no more than 17 or to select a best model among all models that fit better than GOSF. This is another important application of our method.

1.1 Structure of the paper

In Section 2, we introduce a general measure of spurious fit via generalized likelihood ratios, which extends the concept of spurious correlation in the linear model to more general models, including generalized linear models and robust linear regression. We also introduce a local adaptive majorization-minimization (LAMM) algorithm to compute the GOSF. Section 3 presents the main results on the limiting laws of goodness of spurious fit and their bootstrap approximations. For conducting inference, we use the proposed LAMM algorithm to compute the bootstrap statistic. In Section 4, we discuss an application of our theoretical findings to high-dimensional statistical inference and model selection. Section 5 presents numerical studies. Proofs of the main results, Theorems 2 and 6, are provided in Section 6; in each case, we break down the key steps in a series of lemmas with proofs deferred to the appendix.

1.2 Notations

We collect standard pieces of notation here for readers’ convenience. For two sequences $\{a_n\}$ and $\{b_n\}$ of positive numbers, we write $a_n = O(b_n)$ or $a_n \lesssim b_n$ if there exists a constant $C > 0$ such that $a_n/b_n \leq C$ for all sufficiently large n ; we write $a_n \asymp b_n$ if there exist constants $C_1, C_2 > 0$ such that, for all n large enough, $C_1 \leq a_n/b_n \leq C_2$; and we write $a_n = o(b_n)$ if $\lim_{n \rightarrow \infty} a_n/b_n = 0$, respectively. For $a, b \in \mathbb{R}$, we write $a \vee b = \max(a, b)$.

For every positive integer ℓ , we write $[\ell] = \{1, 2, \dots, \ell\}$, and for any set S , we use S^c to denote its complement and $|S|$ for its cardinality. For any real-valued random variable X , its sub-Gaussian norm is defined by $\|X\|_{\psi_2} = \sup_{\ell \geq 1} \ell^{-1/2} (\mathbb{E}|X|^\ell)^{1/\ell}$. We say that a random variable X is sub-Gaussian if $\|X\|_{\psi_2} < \infty$.

Let p, q be two positive integers. For every p -vector $\mathbf{u} = (u_1, \dots, u_p)^T$, we define its ℓ_q -norm to be $\|\mathbf{u}\|_q = (\sum_{i=1}^p |u_i|^q)^{1/q}$, and set $\|\mathbf{u}\|_0 = \sum_{i=1}^p I\{u_i \neq 0\}$. Let $S^{p-1} = \{\mathbf{u} \in \mathbb{R}^p : \|\mathbf{u}\|_2 = 1\}$ be the unit sphere in \mathbb{R}^p . Moreover, for each subset $S \subseteq [p]$ with $|S| = s \in [p]$, we denote by \mathbf{u}_S the s -variate sub-vector of \mathbf{u} containing only the coordinates indexed by S . We use $\|\mathbf{M}\|$ to denote the spectral norm of a matrix \mathbf{M} .

2. Goodness of spurious fit

Let Y, Y_1, \dots, Y_n be independent and identically distributed (i.i.d.) random variables with mean zero and variance $\sigma^2 > 0$, and $\mathbf{X}, \mathbf{X}_1, \dots, \mathbf{X}_n$ be i.i.d. p -dimensional random vectors. We write

$$\mathbf{X} = (X_1, \dots, X_p)^T, \mathbb{X} = (\mathbf{X}_1, \dots, \mathbf{X}_n)^T \in \mathbb{R}^{n \times p} \text{ and } \mathbf{X}_i = (X_{i1}, \dots, X_{ip})^T, \quad i = 1, \dots, n.$$

For $s \in [p]$, the maximum s -multiple correlation between Y and \mathbf{X} is given by

$$\widehat{R}_n(s, p) = \max_{\alpha \in \mathbb{R}^p: \|\alpha\|_0 \leq s} \widehat{\text{corr}}_n(Y, \alpha^\top \mathbf{X}), \quad (1)$$

where $\widehat{\text{corr}}_n(\cdot, \cdot)$ denotes the sample Pearson correlation coefficient. When Y and \mathbf{X} are independent, we regard $\widehat{R}_n(s, p)$ as the maximum spurious (multiple) correlation. The limiting distribution of $\widehat{R}_n(s, p)$ is studied in Cai and Jiang (2012) and Fan, Guo and Hao (2012) when $s = 1$ and $X \sim N(\mathbf{0}, \mathbf{I}_p)$ (the standard normal distribution in \mathbb{R}^p), and later in Fan, Shao and Zhou (2015) under a general setting where $s \geq 1$ and \mathbf{X} is sub-Gaussian with an arbitrary covariance matrix.

For binary data, the sample Pearson correlation is not effective for measuring the regression effect. We need a new metric. In classical regression analysis, the multiple correlation coefficient, also known as the R^2 , is the proportion of variance explained by the regression model. For each submodel $S \subseteq [p]$, its R^2 statistic can be computed as

$$R_S^2 = \max_{\theta \in \mathbb{R}^S} \widehat{\text{corr}}_n^2(Y, \mathbf{X}_S^\top \theta). \quad (2)$$

Then, the maximum s -multiple correlation $\widehat{R}_n(s, p)$ can be expressed as the maximum R^2 statistic:

$$\widehat{R}_n^2(s, p) = \max_{S \subseteq [p]: |S| = s} R_S^2. \quad (3)$$

The concept of R^2 can be extended to more general models. For binary response models, Maddala (1983) suggested the following generalization: $-\log(1 - R^2) = \frac{2}{n} \{\ell(\widehat{\beta}) - \ell(\mathbf{0})\}$, where $\ell(\widehat{\beta}) = \log L(\widehat{\beta})$ and $\ell(\mathbf{0}) = \log L(\mathbf{0})$ denote the log-likelihoods of the fitted and the null model, respectively. This motivates us to use the likelihood ratio as a generalization of the goodness of fit beyond the linear model.

Let $L_n(\beta)$, $\beta \in \mathbb{R}^p$ be the negative logarithm of a quasi-likelihood process of the sample $\{(Y_i, \mathbf{X}_i)\}_{i=1}^n$. For a given model size $s \in [p]$, the best subset fit is $\widehat{\beta}(s) := \underset{\beta \in \mathbb{R}^p: \|\beta\|_0 \leq s}{\text{argmin}} L_n(\beta)$. The goodness of such a fit, in comparison with the baseline fit $L_n(\mathbf{0})$, can be measured by

$$\mathcal{LR}_n(s, p) := L_n(\mathbf{0}) - L_n(\widehat{\beta}(s)) = L_n(\mathbf{0}) - \min_{\beta \in \mathbb{R}^p: \|\beta\|_0 \leq s} L_n(\beta). \quad (4)$$

When \mathbf{X} and Y are independent, it becomes the Goodness Of Spurious Fit (GOSF). According to (2) and (3), this definition is consistent with the maximum spurious correlation when it is applied to the linear model with Gaussian quasi-likelihood, where $L_n(\beta; \beta_0, \sigma) = \frac{1}{2} \log(2\pi\sigma^2) + \frac{1}{2} \|\mathbf{Y} - \beta_0 - \mathbb{X}\beta\|_2^2/\sigma^2$ and $\mathbf{Y} = (Y_1, \dots, Y_n)^\top$.

Throughout, we refer to $L_n(\cdot)$ as the loss function which is assumed to be convex. This setup encompasses the generalized linear models (McCullagh and Nelder, 1989) with $L_n(\beta) = \sum_{i=1}^n \{b(\mathbf{X}_i^\top \beta) - Y_i \mathbf{X}_i^\top \beta\}$ under the canonical link where $b(\cdot)$ is a model-dependent convex function (we take the dispersion parameter as one, as we don't consider the dispersion issue), robust regression with $L_n(\beta) = \sum_{i=1}^n |Y_i - \mathbf{X}_i^\top \beta|$, the hinge loss $L_n(\beta) = \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^\top \beta)_+$ in the support vector machine (Vapnik, 1995) and exponential loss $L_n(\beta) =$

$\sum_{i=1}^n \exp(-Y_i \mathbf{X}_i^\top \beta)$ in AdaBoost (Freund and Schapire, 1997) in classification with Y taking values ± 1 .

The prime goal of this paper is to derive the limiting laws of GOSF $\mathcal{LR}_n(s, p)$ in the setting where the response Y and the explanatory variables \mathbf{X} are independent. Here, both s and p can depend on n , as we shall use double-array asymptotics. We will mainly focus on the GLIM and robust linear regression that are of particular interest in statistics.

2.1 Generalized linear models

Recall that $(Y_1, \mathbf{X}_1), \dots, (Y_n, \mathbf{X}_n)$ are i.i.d. copies of (Y, \mathbf{X}) . Assume that the conditional distribution of Y given $\mathbf{X} = \mathbf{x} \in \mathbb{R}^p$ belongs to the canonical exponential family with the probability density function taking the form (McCullagh and Nelder, 1989)

$$f(y; \mathbf{x}, \beta^*) = \exp \{ [y \mathbf{x}^\top \beta^* - b(\mathbf{x}^\top \beta^*)] / \phi + c(y, \phi) \}, \quad (5)$$

where $\beta^* = (\beta_1^*, \dots, \beta_p^*)^\top$ is the unknown p -dimensional vector of regression coefficients, and $\phi > 0$ is the dispersion parameter. The log-likelihood function with respect to the given data $\{(Y_i, \mathbf{X}_i)\}_{i=1}^n$ is $\sum_{i=1}^n c(Y_i, \phi) + \phi^{-1} \sum_{i=1}^n \{Y_i \mathbf{X}_i^\top \beta - b(\mathbf{X}_i^\top \beta)\}$. For simplicity, we take $\phi = 1$ with the exception that in the linear model with Gaussian noise, $\phi = \sigma^2$ is the variance. Two other showcases are

1. Logistic regression: $b(u) = \log(1 + e^u)$, $u \in \mathbb{R}$ and $\phi = 1$.

2. Poisson regression: $b(u) = e^u$, $u \in \mathbb{R}$ and $\phi = 1$.

In GLIM, the loss function is $L_n(\beta) = \sum_{i=1}^n \{b(\mathbf{X}_i^\top \beta) - Y_i \mathbf{X}_i^\top \beta\}$. By (4), the generalized measure of goodness of fit for GLIM is

$$\mathcal{LR}_n(s, p) = nb(0) - \min_{\beta \in \mathbb{R}^p: \|\beta\|_0 \leq s} L_n(\beta). \quad (6)$$

In Section 3, we derive under mild regularity conditions the limiting distribution of GOSF $\mathcal{LR}_n(s, p)$ in the null model. This extends the classical Wilks theorem (Wilks, 1938). Here, we interpret $\mathcal{LR}_n(s, p)$ as the degree of spuriousness caused by the high-dimensionality.

2.2 L_1 regression

In this section, we revisit the high-dimensional linear model

$$\mathbf{Y} = \mathbb{X}\beta^* + \varepsilon \quad \text{or} \quad Y_i = \mathbf{X}_i^\top \beta^* + \varepsilon_i, \quad i = 1, \dots, n, \quad (7)$$

where $\mathbf{Y} = (Y_1, \dots, Y_n)^\top$ is the response vector and $\varepsilon = (\varepsilon_1, \dots, \varepsilon_n)^\top$ is the n -vector of measurement errors. Robustness considerations lead to least absolute deviation (LAD) regression and more generally quantile regression (Koenker, 2005). For simplicity, we consider the ℓ_1 -loss $L_n(\beta) = \sum_{i=1}^n |Y_i - \mathbf{X}_i^\top \beta|$, $\beta \in \mathbb{R}^p$. The generalized measure of goodness of fit (4) now becomes

$$\mathcal{LR}_n(s, p) = \|\mathbf{Y}\|_1 - \min_{\beta \in \mathbb{R}^p: \|\beta\|_0 \leq s} L_n(\beta). \quad (8)$$

The limiting distribution of GOSF $\mathcal{LR}_n(s, p)$ is studied in Section 3.4.

In particular, if $\varepsilon_1, \dots, \varepsilon_n$ in (7) are i.i.d. from the double exponential distribution with density $f_\varepsilon(u) = \frac{1}{2}e^{-|u|}$, $u \in \mathbb{R}$, the ℓ_1 -loss $L_n(\cdot)$ corresponds to the negative log-likelihood function. In general, we assume that the regression error ε_i has median zero, that is, $\mathbb{P}(\varepsilon_i \leq 0) = \frac{1}{2}$. Hence, the conditional median of Y_i given \mathbf{X}_i is $\mathbf{X}_i^T \boldsymbol{\beta}^*$ for $i \in [n]$, and $\boldsymbol{\beta}^* = \operatorname{argmin}_{\boldsymbol{\beta} \in \mathbb{R}^p} \mathbb{E} \mathbf{X}^T L_n(\boldsymbol{\beta})$, where $\mathbb{E} \mathbf{X}(\cdot) = \mathbb{E}[\mathbf{X}_1, \dots, \mathbf{X}_n]$ denotes the conditional expectation given $\{\mathbf{X}_i\}_{i=1}^n$.

2.3 An LAMM algorithm

The computation of the best subset regression coefficient $\hat{\boldsymbol{\beta}}(s)$ in (4) requires solving a combinatorial optimization problem with a cardinality constraint, and therefore is NP-hard. In the following, we suggest a fast and easily implementable method, which combines the forward selection (stepwise addition) algorithm and a local adaptive majorization-minimization (LAMM) algorithm (Lange, Hunter and Yang, 2000; Fan et al., 2015) to provide an approximate solution.

Our optimization problem is $\min_{\boldsymbol{\beta} \in \mathbb{R}^p: \|\boldsymbol{\beta}\|_0 \leq s} f(\boldsymbol{\beta})$, where $f(\boldsymbol{\beta}) = L_n(\boldsymbol{\beta})$. We say that a function $g(\boldsymbol{\beta}) | \boldsymbol{\beta}^{(k)}$ majorizes $f(\boldsymbol{\beta})$ at the point $\boldsymbol{\beta}^{(k)}$ if $f(\boldsymbol{\beta}^{(k)}) = g(\boldsymbol{\beta}^{(k)} | \boldsymbol{\beta}^{(k)})$ and $f(\boldsymbol{\beta}) \leq g(\boldsymbol{\beta} | \boldsymbol{\beta}^{(k)})$ for all $\boldsymbol{\beta} \in \mathbb{R}^p$. An majorization-minimization (MM) algorithm initializes at $\boldsymbol{\beta}^{(0)}$ and then iteratively computes $\boldsymbol{\beta}^{(k+1)} = \operatorname{argmin}_{\boldsymbol{\beta} \in \mathbb{R}^p: \|\boldsymbol{\beta}\|_0 \leq s} g(\boldsymbol{\beta} | \boldsymbol{\beta}^{(k)})$. The target value of such an algorithm is non-increasing since

$$f(\boldsymbol{\beta}^{(k+1)}) \stackrel{\text{majorization}}{\leq} g(\boldsymbol{\beta}^{(k+1)} | \boldsymbol{\beta}^{(k)}) \stackrel{\text{minimization}}{\leq} g(\boldsymbol{\beta}^{(k)} | \boldsymbol{\beta}^{(k)}) \stackrel{\text{initialization}}{=} f(\boldsymbol{\beta}^{(k)}). \quad (9)$$

We now majorize $f(\boldsymbol{\beta})$ at $\hat{\boldsymbol{\beta}}^{(k)}$ by an isotropic quadratic function

$$g_\lambda(\boldsymbol{\beta} | \hat{\boldsymbol{\beta}}^{(k)}) = f(\boldsymbol{\beta}) + \langle \nabla f(\hat{\boldsymbol{\beta}}^{(k)}), \boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{(k)} \rangle + \frac{\lambda}{2} \|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}^{(k)}\|_2^2, \quad \boldsymbol{\beta} \in \mathbb{R}^p. \quad (10)$$

This is a valid majorization as long as $\lambda \geq \max_{\boldsymbol{\beta}} \|\nabla^2 f(\boldsymbol{\beta})\|$ (this will be relaxed below).

The isotropic form on the right-hand side of (10) allows a simple analytic solution given by

$$\hat{\boldsymbol{\beta}}_\lambda^{(k+1)} = \operatorname{argmin}_{\boldsymbol{\beta} \in \mathbb{R}^p: \|\boldsymbol{\beta}\|_0 \leq s} g(\boldsymbol{\beta} | \boldsymbol{\beta}^{(k)}) = \{\hat{\boldsymbol{\beta}}^{(k)} - \lambda^{-1} \nabla f(\hat{\boldsymbol{\beta}}^{(k)})\}_{[1:s]}.$$

Here, we used the notation that for any $\boldsymbol{\beta} \in \mathbb{R}^p$, $\boldsymbol{\beta}_{[1:s]} \in \mathbb{R}^p$ retains the s largest (in magnitude) entries of $\boldsymbol{\beta}$ and assigns the rest to zero.

Remark 1 To implement the MM algorithm, we need to compute the gradient of the objective function of interest. In the L_1 regression, the loss function $L_n(\boldsymbol{\beta}) = \sum_{i=1}^n |Y_i - \mathbf{X}_i^T \boldsymbol{\beta}|$, $\boldsymbol{\beta} \in \mathbb{R}^p$ is not differentiable everywhere. Recall that the subdifferential of the absolute function $h(x) = |x|$, $x \in \mathbb{R}$ is given by

$$\partial h(x) = \begin{cases} \{1\}, & \text{if } x > 0, \\ [-1, 1], & \text{if } x = 0, \\ \{-1\}, & \text{if } x < 0. \end{cases}$$

With slight abuse of notation, we suggest a randomized algorithm using the stochastic subgradient $\nabla L_n(\boldsymbol{\beta}) = \sum_{i=1}^n I(Y_i - \mathbf{X}_i^T \boldsymbol{\beta} > 0) - I(Y_i - \mathbf{X}_i^T \boldsymbol{\beta} < 0) + U_i I(Y_i - \mathbf{X}_i^T \boldsymbol{\beta} = 0)$, where U_1, \dots, U_n are i.i.d. random variables uniformly distributed on $[-1, 1]$.

We propose to use the stepwise forward selection algorithm to compute an initial estimator $\hat{\boldsymbol{\beta}}^{(0)}$. As the MM algorithm decreases the target value as shown in (9), the resulting target value is no larger than that produced by the stepwise forward selection algorithm.

To properly choose the isotropic parameter $\lambda > 0$ without computing the maximum eigenvalue, we use the local adaptive procedure as in Fan et al. (2015). Note that, in order to have a non-increasing target value, the majorization is not actually required. As long as $f(\boldsymbol{\beta}^{(k+1)}) \leq g(\boldsymbol{\beta}^{(k+1)} | \boldsymbol{\beta}^{(k)})$, arguments in (9) hold. Starting from a prespecified value $\lambda = \lambda_0$, we successively inflate λ by a factor $\rho > 1$. After the k th iteration, $\lambda = \lambda_k = \rho^{k-1} \lambda_0$. We take the first ℓ such that $f(\hat{\boldsymbol{\beta}}_{\lambda_k}^{(k+1)}) \leq g_{\lambda_k}(\hat{\boldsymbol{\beta}}_{\lambda_k}^{(k+1)} | \hat{\boldsymbol{\beta}}^{(k)})$ and set $\hat{\boldsymbol{\beta}}^{(k+1)} = \hat{\boldsymbol{\beta}}_{\lambda_k}^{(k+1)}$. Such an ℓ always exists as a large ℓ will major the function f . We then continue with the iteration in the MM part. A simple criteria for stopping the iteration is that $|f(\hat{\boldsymbol{\beta}}^{(k+1)}) - f(\hat{\boldsymbol{\beta}}^{(k)})| \leq \epsilon$ for a sufficiently small ϵ , say 10^{-5} . We refer to Fan et al. (2015) for a detailed computational complexity analysis of the LAMM algorithm.

While the LAMM algorithm can be applied to compute $\hat{\boldsymbol{\beta}}(s)$ in a general setting, in our application, the algorithm is mainly applied to compute GOSF under the null model (see Figure 1 and Section 3.5). From our simulation experiences, our algorithm delivers a good enough solution under the null model. It always provides an upper certificate $f(\hat{\boldsymbol{\beta}}_0)$ to the problem $\min_{\|\boldsymbol{\beta}\|_0 \leq s} f(\boldsymbol{\beta})$, where $\hat{\boldsymbol{\beta}}_0$ is the output of the LAMM algorithm. As in Bertsimas, King and Mazumder (2016), if needed to verify the accuracy of our method, a lower certificate is $f(\hat{\boldsymbol{\beta}}_1)$, where $\hat{\boldsymbol{\beta}}_1$ is the solution to the convex problem $\min_{\|\boldsymbol{\beta}\|_0 \leq B_s} f(\boldsymbol{\beta})$, and B_s is a sufficient large constant so that the L_∞ -solution satisfies $\|\hat{\boldsymbol{\beta}}(s)\|_\infty \leq B_s$. For example, under the null model, it is well known that $\|\hat{\boldsymbol{\beta}}(s)\|_\infty = O_{\mathbb{P}}\{s \sqrt{(\log p)/n}\}$. Therefore, we can take $B_s = C_s s \sqrt{(\log p)/n}$ for a sufficiently large constant C_s . A data-driven heuristic approach is to take $B_s = 2\|\hat{\boldsymbol{\beta}}_1(s)\|_\infty$ along the Lasso path such that $\|\hat{\boldsymbol{\beta}}_1(s)\|_0 = s$.

Note that the minimum target value falls in the interval $[f(\hat{\boldsymbol{\beta}}_1), f(\hat{\boldsymbol{\beta}}_0)]$. If this interval is very tight, we have certified that $\hat{\boldsymbol{\beta}}_0$ is an accurate solution.

3. Asymptotic distribution of goodness of spurious fit

3.1 Preliminaries

Define $p \times p$ covariance matrices

$$\boldsymbol{\Sigma} = \mathbb{E}(\mathbf{X} \mathbf{X}^T) \quad \text{and} \quad \hat{\boldsymbol{\Sigma}} = n^{-1} \sum_{i=1}^n \mathbf{X}_i \mathbf{X}_i^T. \quad (11)$$

For $s \in [p]$, we say that $S \subseteq [p]$ is an s -subset if $|S| = s$. For every s -subset $S \subseteq [p]$, let $\boldsymbol{\Sigma}_{SS}$ and $\hat{\boldsymbol{\Sigma}}_{SS}$ be the $s \times s$ sub-matrices of $\boldsymbol{\Sigma}$ and $\hat{\boldsymbol{\Sigma}}$ containing the entries indexed by $S \times S$, that is,

$$\boldsymbol{\Sigma}_{SS} = \mathbb{E}(\mathbf{X}_S \mathbf{X}_S^T), \quad \hat{\boldsymbol{\Sigma}}_{SS} = n^{-1} \sum_{i=1}^n \mathbf{X}_{iS} \mathbf{X}_{iS}^T. \quad (12)$$

Condition 3.1 The covariates are standardized to have unit second moment, that is, $\mathbb{E}(\mathbf{X}_j^2) = 1$ for $j = 1, \dots, p$. There exists a random vector $\mathbf{U} \in \mathbb{R}^p$ satisfying $\mathbb{E}(\mathbf{U} \mathbf{U}^T) = \mathbf{I}_p$, such that $\mathbf{X} = \boldsymbol{\Sigma}^{1/2} \mathbf{U}$ and $A_0 := \sup_{\rho \in \mathcal{S}^{p-1}} \|\rho^T \mathbf{U}\|_{q_2} < \infty$.

For $1 \leq s \leq p$, the s -sparse condition number of Σ is given by

$$\gamma_s = \gamma_s(\Sigma) = \sqrt{\lambda_{\max}(s)/\lambda_{\min}(s)}, \quad (13)$$

where $\lambda_{\max}(s) = \max_{\mathbf{u} \in \mathbb{S}^{p-1}, \|\mathbf{u}\|_0 \leq s} \mathbf{u}^\top \Sigma \mathbf{u}$ and $\lambda_{\min}(s) = \min_{\mathbf{u} \in \mathbb{S}^{p-1}, \|\mathbf{u}\|_0 \leq s} \mathbf{u}^\top \Sigma \mathbf{u}$ denote the s -sparse largest and smallest eigenvalues of Σ , respectively.

Let $\mathbf{G} = (G_1, \dots, G_p)^\top \sim N(\mathbf{0}, \Sigma)$ be a centered Gaussian random vector with covariance matrix Σ . For any s -subset $S \subseteq [p]$, $\mathbf{G}_S \sim N(\mathbf{0}, \Sigma_{SS})$. Define the random variable

$$R_0(s, p) = \max_{S \subseteq [p], |S|=s} \|\Sigma_{SS}^{-1/2} \mathbf{G}_S\|_2, \quad (14)$$

which is the maximum of the ℓ_2 -norms of a sequence of dependent chi-squared random variables with s degrees of freedom. The distribution of $R_0(s, p)$ depends on the unknown Σ and can be estimated by the multiplier bootstrap in Section 3.5. It will be shown that this distribution is the asymptotic distribution of GOSF. In particular, for the isotropic case where $\Sigma = \mathbf{I}_p$, $R_0(s, p) = G_{(1)}^2 + \dots + G_{(s)}^2$, the sum of the largest s order statistics of p independent χ_1^2 random variables.

3.2 Generalized linear models

For i.i.d. observations $\{(Y_i, \mathbf{X}_i)\}_{i=1}^n$ from the distribution in (5), define individual residuals $\varepsilon_i = Y_i - \mathbb{E}\mathbf{X}^\top(Y_i) = Y_i - b(\mathbf{X}_i^\top \boldsymbol{\beta}^*)$ with conditional variance $\text{Var}_{\mathbf{X}}(\varepsilon_i) = \phi b''(\mathbf{X}_i^\top \boldsymbol{\beta}^*)$, where $\text{Var}_{\mathbf{X}}(\cdot) = \mathbb{E}\mathbf{X}^\top \{\cdot - \mathbb{E}\mathbf{X}^\top(\cdot)\}^2$. In particular, under the null model, Y is independent of \mathbf{X} with mean $\mu_Y := \mathbb{E}(Y) = b'(0)$ and variance $\sigma_Y^2 := \text{Var}(Y) = \phi b''(0)$.

Condition 3.2 There exists $a_0 > 0$ such that $\mathbb{E} \exp\{u \sigma_Y^{-1}(Y - \mu_Y)\} \leq \exp(a_0 u^2/2)$ holds for all $u \in \mathbb{R}$. The function $b(\cdot)$ in (5) satisfies

$$\min_{u: |u| \leq 1} b''(u) \geq a_1 \quad \text{and} \quad \max_{u: |u| \leq 1} |b'''(u)| \leq A_1 \quad (15)$$

for some constants $a_1, A_1 > 0$.

Condition 3.2 is satisfied by a wide class of GLIMs, including the logistic and Poisson regression models. The following theorem shows that, under certain moment and regularity conditions, the distribution of the generalized likelihood ratio statistic $2\mathcal{LR}_n(s, p)$ can be consistently approximated by that of $R_0^2(s, p)$ given in (14).

Theorem 2 Let Conditions 3.1 and 3.2 be satisfied. Assume that $\phi = 1$ in (5), $p, n \geq 3$ and $1 \leq s \leq \min(p, n)$. Then, under the null model (7) with $\boldsymbol{\beta}^* = \mathbf{0}$,

$$\sup_{t \geq 0} \mathbb{P}\{2\mathcal{LR}_n(s, p) \leq t\} - \mathbb{P}\{R_0^2(s, p) \leq t\} \leq C \left[\{s \log(\gamma_s p m)\}^{7/8} n^{-1/8} + \sigma_Y^{1/2} \{s \log(\gamma_s p m)\}^2 n^{-1/2} \right], \quad (16)$$

where $C > 0$ is a constant depending only on a_0, a_1, A_0, A_1 in Conditions 3.1 and 3.2.

Remark 3 We regard Theorem 2 as a nonasymptotic, high-dimensional version of the celebrated Wilks theorem. In the low-dimensional setting where $s = p$ is fixed, Theorem 2 reduces to the conventional Wilks theorem, which asserts that the generalized likelihood ratio statistic converges in distribution to χ_p^2 . In addition, we also provide a Berry-Esseen bound in (16).

3.3 Linear least squares regression

As a specific case of GLIM, we consider the linear regression model (7) with the loss function $L_n(\boldsymbol{\beta}) = \frac{1}{2} \|\mathbf{Y} - \mathbb{X}\boldsymbol{\beta}\|_2^2$. The corresponding likelihood ratio statistic

$$\mathcal{LR}_n(s, p) = \frac{1}{2} \|\mathbf{Y}\|_2^2 - \min_{\boldsymbol{\beta} \in \mathbb{R}^p: \|\boldsymbol{\beta}\|_0 \leq s} L_n(\boldsymbol{\beta}) \quad (17)$$

then coincides with that in (6) with $b(u) = \frac{1}{2} u^2$. We state the null limiting distribution of $\mathcal{LR}_n(s, p)$ in a general case, where $\varepsilon_1, \dots, \varepsilon_n$ are i.i.d. copies of a sub-Gaussian random variable ε . Specifically, we assume that

Condition 3.3 ε is a centered, sub-Gaussian random variable with $\text{Var}(\varepsilon) = \sigma^2 > 0$ and $K_0 := \|\varepsilon\|_{\psi_2} < \infty$. Moreover, write $v_\ell = \mathbb{E}(|\varepsilon|^\ell)$ for $\ell \geq 3$.

The following corollary is a particular case of the general result Theorem 2 with $b(u) = \frac{1}{2} u^2$, $u \in \mathbb{R}$ and $\phi = \sigma^2$. By examining the proof of Theorem 2 and noting that $b'' \equiv 0$, it can be easily shown that the second term on the right-side of (16) vanishes. Hence, the proof is omitted.

Corollary 4 Let Conditions 3.1 and 3.3 hold. Assume that $p, n \geq 3$ and $1 \leq s \leq \min(p, n)$. Then, under the null model (7) with $\boldsymbol{\beta}^* = \mathbf{0}$,

$$\sup_{t \geq 0} \mathbb{P}\{2\mathcal{LR}_n(s, p) \leq t\} - \mathbb{P}\{\sigma^2 R_0^2(s, p) \leq t\} \leq C \{s \log(\gamma_s p m)\}^{7/8} n^{-1/8},$$

where $C > 0$ is a constant depending only on A_0 and K_0 in Conditions 3.1 and 3.3.

Remark 5 Under the null model, the variance σ^2 can be consistently estimated by $\hat{\sigma}_0^2 = n^{-1} \sum_{i=1}^n (Y_i - \bar{Y})^2$, where $\bar{Y} = n^{-1} \sum_{i=1}^n Y_i$. Under the same conditions of Corollary 4, it can be proved that

$$\sup_{t \geq 0} \mathbb{P}\{2\mathcal{LR}_n(s, p) \leq t\} - \mathbb{P}\{\hat{\sigma}_0^2 R_0^2(s, p) \leq t\} \lesssim \{s \log(\gamma_s p m)\}^{7/8} n^{-1/8},$$

which is in line with Theorem 3.1 in Fan, Shao and Zhou (2015). To see this, note that

$$\begin{aligned} 2\mathcal{LR}_n(s, p) &= \|\mathbf{Y}\|_2^2 - \min_{S \subseteq [p], |S|=s} \min_{\boldsymbol{\theta} \in \mathbb{R}^S} \|\mathbf{Y} - \mathbb{X}_S \boldsymbol{\theta}\|_2^2 \\ &= \max_{S \subseteq [p], |S|=s} \mathbf{Y}^\top \mathbb{X}_S (\mathbb{X}_S^\top \mathbb{X}_S)^{-1} \mathbb{X}_S^\top \mathbf{Y} = \max_{\boldsymbol{\alpha} \in \mathbb{R}^p: \|\boldsymbol{\alpha}\|_0 \leq s} (\mathbf{Y}^\top \mathbb{X} \boldsymbol{\alpha})^2 / \|\mathbb{X} \boldsymbol{\alpha}\|_2^2. \end{aligned}$$

The estimator $\hat{\sigma}_0^2$, used in computing the maximum spurious correlation, can be seriously biased beyond the null model and hence adversely affect the power. Thus, we suggest using either the refitted cross-validation procedure (Fan, Guo and Hao, 2012) or the scaled Lasso estimator (Sun and Zhang, 2012) to estimate σ^2 .

3.4 Linear median regression

We now state an analogous result to Theorem 2 regarding the ℓ_1 -loss considered in Section 2.2.

Condition 3.4 The noise $\varepsilon_1, \dots, \varepsilon_n$ in (7) are i.i.d. copies of a random variable ε satisfying $\mathbb{E}|\varepsilon|^\kappa < \infty$ for some $1 < \kappa \leq 2$. There exist positive constants $a_2 < (\mathbb{E}|\varepsilon|)^{-1}$, A_2 and A_3 such that the distribution function $F_\varepsilon(\cdot)$ and the density function $f_\varepsilon(\cdot)$ of ε satisfy

$$2 \max\{1 - F_\varepsilon(u), F_\varepsilon(-u)\} \leq (1 + a_2 u)^{-1} \quad \text{for all } u \geq 0, \quad (18)$$

$$\max_{u \in \mathbb{R}} f_\varepsilon(u) \leq A_2 \quad \text{and} \quad \max_{|u| \leq 1} \max_{|v| \leq 1} \{ |f_\varepsilon(u+v)|, |f_\varepsilon(u-v)| \} \leq A_3. \quad (19)$$

Theorem 6 If $p, n \geq 3$ and $1 \leq s \leq \min(p, n)$, then under the null model (7) with $\beta^* = \mathbf{0}$ and Conditions 3.1 and 3.4, we have

$$\begin{aligned} \sup_{t \geq 0} \mathbb{P}\{2\mathcal{LR}_n(s, p) \leq t\} &= \mathbb{P}\{R_0^2(s, p)/\{2f_\varepsilon(0)\} \leq t\} \\ &\leq C_1 n^{1-\kappa} + C_2 \{s \log(\gamma_s p n)\}^{7/8} n^{-1/8} + \gamma_s^{1/4} \{s \log(\gamma_s p n)\}^{3/2} n^{-1/4}, \end{aligned} \quad (20)$$

where $\mathcal{LR}_n(s, p)$ is given by (8), $C_1 > 0$ is a constant depending on $a_2, \kappa, \mathbb{E}|\varepsilon|, \mathbb{E}|\varepsilon|^\kappa$ and $C_2 > 0$ is a constant depending on a_2, A_0, A_2 and A_3 in Conditions 3.1 and 3.4.

Remark 7 Under the null model, the unknown parameter $f_\varepsilon(0)$ can be consistently estimated by the kernel density estimator $\widehat{f}_\varepsilon(0) = (nh)^{-1} \sum_{i=1}^n K(Y_i/h)$, where $K(\cdot)$ is a kernel function and $h = h_n > 0$ is the bandwidth. For simplicity, we may use the Epanechnikov kernel function $K_{\text{Epa}}(u) = \frac{3}{4}(1 - u^2)I(|u| \leq 1)$ along with the rule-of-thumb bandwidth $h_{\text{EOT}} = 2.34 \widehat{\sigma}_0 n^{-1/5}$, where $\widehat{\sigma}_0^2 = n^{-1} \sum_{i=1}^n (Y_i - \bar{Y})^2$.

3.5 Multiplier bootstrap procedure

The distribution of the random variable $R_0(s, p)$ given by (14) depends on the unknown covariance matrix Σ . In practice, it is natural to replace Σ by $\widehat{\Sigma} = n^{-1} \sum_{i=1}^n \mathbf{X}_i \mathbf{X}_i^T$ and $\mathbf{G} \sim N(0, \Sigma)$ by $\widehat{\mathbf{G}} \sim N(0, \widehat{\Sigma})$ in the definition of $R_0(s, p)$. With this substitution, the distribution of $R_0(s, p)$ can be simulated. In particular, $\widehat{\mathbf{G}}$ can be simulated as $n^{-1/2} \sum_{i=1}^n \varepsilon_i \mathbf{X}_i$, where $\varepsilon_1, \dots, \varepsilon_n$ are i.i.d. standard normal random variables that are independent of $\{\mathbf{X}_i\}_{i=1}^n$. The resulting estimator is

$$R_n(s, p) = \max_{S \subseteq [p]: |S|=s} \|\widehat{\Sigma}_{SS}^{-1/2} \widehat{\mathbf{G}}_S\|_2, \quad (21)$$

which is a multiplier bootstrap version of $R_0(s, p)$. The following proposition follows directly from Theorem 3.2 in Fan, Shao and Zhou (2015).

Proposition 8 Assume that Condition (3.1) holds, $1 \leq s \leq \min(p, n)$ and $s \log(\gamma_s p n) = o(n^{1/5})$ as $n \rightarrow \infty$. Then $\sup_{p \geq 20} \mathbb{P}\{R_0(s, p) \leq t\} - \mathbb{P}\{R_n(s, p) \leq t | \mathbf{X}_1, \dots, \mathbf{X}_n\} \rightarrow 0$ in probability.

The computation of $R_n(s, p)$ requires solving a combinatorial optimization. This can be alleviated by using the LAMM algorithm in Section 2.3. To begin with, by Remark 5, we write $R_n(s, p)$ in (21) as

$$R_n^2(s, p) = \max_{S \subseteq [p]: |S|=s} \mathbf{e}^T \mathbb{X}_S (\mathbb{X}_S^T \mathbb{X}_S)^{-1} \mathbb{X}_S^T \mathbf{e} = \|\mathbf{e}\|_2^2 - \min_{\beta \in \mathbb{R}^p: \|\beta\|_0 \leq s} \|\mathbf{e} - \mathbb{X} \beta\|_2^2,$$

where $\mathbf{e} = (\varepsilon_1, \dots, \varepsilon_n)^T$ and $\mathbb{X}_S = (\mathbf{X}_{1S}, \dots, \mathbf{X}_{nS})^T$ for every subset $S \subseteq [p]$. This can be computed approximately by the LAMM algorithm in Section 2.3, resulting in the solution $\widehat{\beta}(s)$. Finally, we set $R_n^2(s, p) = \|\mathbf{e}\|_2^2 - \|\mathbf{e} - \mathbb{X} \widehat{\beta}(s)\|_2^2$.

The numerical performance may be improved by employing mixed integer optimization formulations (Bertsimas, King and Mazumdar, 2016). Such an attempt, however, is beyond the scope of the paper and we leave it for future research.

4. Spurious discoveries and model selection

Based on the theoretical developments in Section 3, here we address the question whether discoveries by machine learning and data mining techniques for GLIM are any better than by chance. For simplicity, we focus on the Lasso. Let $q_\alpha(s, p)$ be the upper α -quantile of the random variable $R_0(s, p)$ defined by (14). Assume that the dispersion parameter ϕ in (5) equals 1. By Theorem 2, we see that for any prespecified $\alpha \in (0, 1)$,

$$\mathbb{P}\{2\mathcal{LR}_n(s, p) \leq q_\alpha^2(s, p)\} \rightarrow 1 - \alpha, \quad (22)$$

where $\mathcal{LR}_n(s, p)$ is as in (6).

Let $\widehat{\beta}_\lambda = \arg \min_{\beta} \{L_n(\beta) + \lambda \|\beta\|_1\}$ be the ℓ_1 -penalized maximum likelihood estimator with $\widehat{s}_\lambda = \|\widehat{\beta}_\lambda\|_0 = \lceil \text{supp}(\widehat{\beta}_\lambda) \rceil$, where $\lambda > 0$ is the regularization parameter. The goodness of fit is likelihood ratio $L_n(\mathbf{0}) - L_n(\widehat{\beta}_\lambda)$. Since \widehat{s}_λ covariates are selected, it should be compared with the distribution of GOSF $\mathcal{LR}_n(s, p)$ by taking $s = \widehat{s}_\lambda$. In view of (22), if

$$L_n(\widehat{\beta}_\lambda) \geq L_n(\mathbf{0}) - q_\alpha^2(\widehat{s}_\lambda, p)/2 = nb(0) - q_\alpha^2(\widehat{s}_\lambda, p)/2,$$

then we may regard the discovery of variables \widehat{s}_λ as unimpressive, no better than fitting by chance, or simply spurious.

In practice, the unknown quantile $q_\alpha(s, p)$ should be replaced by its bootstrap version $q_{n,\alpha}(s, p)$, the upper α -quantile of $R_n(s, p)$ defined by (21). This leads to the following data-driven criteria for judging where the discovery $\widehat{S}(\lambda)$ is spurious:

$$L_n(\widehat{\beta}_\lambda) \geq nb(0) - q_{n,\alpha}^2(\widehat{s}_\lambda, p)/2. \quad (23)$$

The theoretical justification is given by Theorem 2 and Proposition 8. In particular, when the loss is quadratic, this reduces to the case studied by Fan, Shao and Zhou (2015).

The concept of GOSF and its theoretical quantile provide important guidelines for model selection. Let $\widehat{\beta}_{cv}$ be a cross-validated Lasso estimator, which selects $\widehat{s}_{cv} = \|\widehat{\beta}_{cv}\|_0$ important variables. Due to the bias of the ℓ_1 penalty, the Lasso typically selects far larger model size since the visible bias in Lasso forces the cross-validation procedure to choose a smaller value of λ . This phenomenon is documented in the simulations studies. See Table 1 in

Section 5.2. With an over-selected model, both the goodness of fit $\widehat{\mathcal{LR}}_\lambda = L_n(\mathbf{0}) - L_n(\widehat{\beta}_\lambda)$ and the spurious fit can be very large, and so is the finite sample Wilks approximation error. To avoid over-selecting, we suggest an alternative procedure that uses the quantity $q_{n,\alpha}(s, p)$ as a guidance to choose the tuning parameter, which guards us from spurious discoveries. More specifically, for each λ in the Lasso solution path, we compute $\widehat{\mathcal{LR}}_\lambda$ and $q_{n,\alpha}(s, p)_{s=\widehat{s}_\lambda}$ with a prespecified α . Starting from the largest λ , we stop the Lasso path the first time that the sign of $2\widehat{\mathcal{LR}}_\lambda - q_{n,\alpha}(\widehat{s}_\lambda, p)$ is changed from positive to negative, and let $\widehat{\lambda}_{\text{fit}}$ be the smallest λ satisfying $2\widehat{\mathcal{LR}}_\lambda \geq q_{n,\alpha}(\widehat{s}_\lambda, p)$. Denote by \widehat{n}_{fit} the corresponding selected model size. This value can be regarded as the maximum model size for Lasso (or any other variable selection technique such as SCAD) to choose from. Another viable alternative is to only select the best cross-validated model among those whose fit are better than GOSF. We will show in Section 5.2 by simulation studies that this procedure selects much smaller model size which is closer to the truth.

5. Numerical studies

5.1 Accuracy of the Gaussian approximation

First we ran a simulation study to examine how accurate the Gaussian approximation $R_0^2(s, p)$ is to the generalized likelihood ratio statistic $2\mathcal{LR}_n(s, p)$ in the null model. To illustrate the method, we focus on the logistic regression model: $\mathbb{P}(Y = 1 | \mathbf{X}) = \exp(\mathbf{X}^T \boldsymbol{\beta}^*) / (1 + \exp(\mathbf{X}^T \boldsymbol{\beta}^*))$. Under the null model $\boldsymbol{\beta}^* = 0$, Y_1, \dots, Y_n are i.i.d. Bernoulli random variables with success probability 1/2. Independent of Y_i 's, we generate $\mathbf{X}_i \sim N(0, \boldsymbol{\Sigma})$ with two different covariance matrices: $\boldsymbol{\Sigma}_1 = (\rho^{|j-k|})_{1 \leq j, k \leq p}$ and $\boldsymbol{\Sigma}_2 = (\sigma_{2,jk})_{1 \leq j, k \leq p}$, where

$$\sigma_{2,jk} = (| |j - k| + 1 |^{2\rho} + | |j - k| - 1 |^{2\rho} - 2 |j - k| - 1 |^{2\rho}) / 2, \quad 1 \leq j, k \leq p.$$

The first design has an AR(1) correlation structure (a short-memory process), whereas the second design reflects strong long memory dependence. We take $\rho = 0.8$ in both cases.

Figure 2 reports the distributions of generalized likelihood ratios (GLRs) and their Gaussian approximations (GARs) when $n = 400$, $p = 1000$ and $s \in \{1, 2, 5, 10\}$. The results show that the accuracy of Gaussian approximation is fairly reasonable and is affected by the size of s as well as the dependence between the coordinates of \mathbf{X} .

5.2 Detection of spurious discoveries

In this section, we conduct a moderate scale simulation study to examine how effective the multiplier bootstrap quantile $q_{n,\alpha}(s, p)$ serves as a benchmark for judging whether the discovery is spurious. To illustrate the main idea, again we restrict our attention to the logistic regression model and the Lasso procedure.

The results reported here are based on 200 simulations with the ambient dimension $p = 400$ and the sample size n taken values in $\{120, 160, 200\}$. The true regression coefficient vector $\boldsymbol{\beta}^* \in \mathbb{R}^p$ is $(3, -1, 3, -1, 3, 0, \dots, 0)^T$. We consider two random designs: $\boldsymbol{\Sigma} = \mathbf{I}_p$ (independent) and $\boldsymbol{\Sigma} = (\sigma_{2,j-k})_{1 \leq j, k \leq p}$ (dependent).

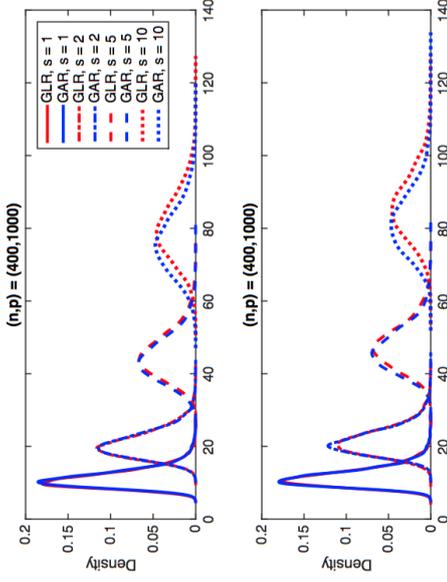


Figure 2: Distributions of generalized likelihood ratios (red) and Gaussian approximations (blue) based on 5000 simulations for $n = 400$, $p = 1000$ and $s = 1, 2, 5, 10$ when $\boldsymbol{\Sigma}$ is equal to $\boldsymbol{\Sigma}_1$ (upper panel) or $\boldsymbol{\Sigma}_2$ (lower panel).

Let $\widehat{\beta}_{\text{cv}}$ be the five-fold cross-validated Lasso estimator, which selects a model of size $\widehat{s}_{\text{cv}} = \|\widehat{\beta}_{\text{cv}}\|_0$. For a given $\alpha \in (0, 1)$, consider the spurious discovery probability (SDP)

$$\mathbb{P}\{n \log(2) - L_n(\widehat{\beta}_{\text{cv}}) \leq q_{n,\alpha}^2(\widehat{s}_{\text{cv}}, p) / 2\},$$

which is basically the probability of the type II error since the simulated model is not null. We take $\alpha = 0.1$ and compute the empirical SDP based on 200 simulations. For each simulated data set, $q_{n,\alpha}(s, p)|_{s=\widehat{s}_{\text{cv}}}$ is computed based on 1000 bootstrap replications. The results are depicted in Table 1 below.

Table 1: The empirical power and the median size of the selected models with its robust standard deviation (RSD) in the parenthesis based on 200 simulations when $p = 400$ and $\alpha = 10\%$. RSD is the interquantile range divided by 1.34.

	$n = 120$			$n = 160$			$n = 200$		
	Ind.	Dep.	Dep.	Ind.	Dep.	Dep.	Ind.	Dep.	Dep.
Power	0.595	0.750	0.925	0.980	0.980	1.000	1.000	1.000	1.000
\widehat{s}_{cv}	32.0 (13.43)	24.5 (11.94)	40.0 (13.81)	25.5 (12.69)	42.0 (14.18)	29.0 (14.18)	42.0 (14.18)	42.0 (14.18)	42.0 (14.18)

As reflected by Table 1, the empirical power, which is one minus the empirical SDP, increases rapidly as the sample size n grows. This is in line with our intuition that the more data we have, the less likely that the discovery by a variable selection method is spurious. When the sample size is small, the SDP can be high and hence the discovery

$\hat{S}_{cv} = \text{supp}(\hat{\beta}_{\alpha})$ should be interpreted with caution. We need either more samples or more powerful variable selection methods.

We see from Table 1 that the Lasso with cross-validation selects far larger model size than the true one, which is 5. This is because the intrinsic bias in Lasso forces the cross-validation procedure to choose a smaller value of λ . We now use our procedure in Section 4 to choose the tuning parameter from the Lasso solution path. As before, we take $\alpha = 0.1$ in $q_{n,\alpha}(s, p)$ to provide an upper bound on the model size from perspective of guarding against spurious discoveries. The empirical median of \hat{m}_n and its robust standard deviation are 9 and 1.87 over 200 simulations when $(n, p) = (200, 400)$ and $\Sigma = (0.5^{j-k})_{1 \leq j, k \leq p}$. The feature over-selection phenomenon is considerably alleviated.

5.3 Neuroblastoma data

In this section, we apply the idea of detecting spurious discoveries to the neuroblastoma data reported in Oberthner et al. (2006). This data set consists of 251 patients of the German Neuroblastoma Trials NB90-NB2004, diagnosed between 1989 and 2004. The complete data set, obtained via the MicroArray Quality Control phase-II (MAQC-II) project (Shi et al., 2010), includes gene expression over 10,707 probe sites. There are 246 subjects with 3-year event-free survival information available (56 positive and 190 negative). See Oberthner et al. (2006) for more details about the data sets.

For each $\lambda > 0$, we apply Lasso using the logistic regression model to select $\hat{\Sigma}_\lambda$ genes. In particular, ten-fold cross-validated Lasso selects $\hat{S}_{cv} = 40$ genes. Then we calculate the goodness of fit $\widehat{\mathcal{LR}}_\lambda := L_n(\mathbf{0}) - L_n(\hat{\beta}_\lambda) = n \log(2) - L_n(\hat{\beta}_\lambda)$. Along the Lasso path, we record in Table 2 the number of selected probes, the corresponding square-root the goodness of fit $(2\widehat{\mathcal{LR}}_\lambda)^{1/2}$ and upper α -quantiles of the multiplier bootstrap approximations $R_n(s, p)_{s=\hat{S}_\alpha, p=10,707}$ with $\alpha = 10\%$ and 5% based on 2000 bootstrap replications. For illustrative purposes, we only display partial Lasso solutions with selected model size $\hat{\Sigma}_\lambda$ lying between 20 and 40. From Table 2, we observe that only the discovery of 17 probes has a generalized measure of the goodness of fit better than GOSF at $\alpha = 5\%$, whereas the finding (of the 40 probes) via the cross-validation procedure is likely to over-select.

6. Proofs

We now turn to the proofs of Theorems 2 and 6. In each proof, we provide the primary steps, with more technical details stated as lemmas and proved in the appendix.

6.1 Proof of Theorem 2

Throughout, we work with the quasi-likelihood $\mathcal{L}_n(\beta) = -L_n(\beta) = \sum_{i=1}^n \{Y_i \mathbf{X}_i^T \beta - b(\mathbf{X}_i^T \beta)\}$ and consider the general case where the dispersion parameter ϕ in (5) is specified (not necessarily equals 1 to facilitate the derivations for the normal case). For a given $s \in [p]$, define

$$Q_n(s, p) = \max_{\beta \in \mathbb{R}^p: \|\beta\|_0 \leq s} \mathcal{L}_n(\beta) \quad \text{and} \quad Q_n^* = \mathcal{L}_n(\mathbf{0}).$$

We divide the proof into three steps. First, for each s -subset $S \subseteq [p]$, we prove Wilks's result for the S -restricted model where only a subset of the covariates indexed by S are

Table 2: Lasso fitted square-root likelihood ratio statistic, the mean cross-validated error, and upper 0.1- and 0.05-quantiles of the multiplier bootstrap approximation based on 2000 bootstrap samples.

λ	\hat{S}_λ	$(2\widehat{\mathcal{LR}}_\lambda)^{1/2}$	$q_{n,0.1}(\hat{S}_\lambda, p)$	$q_{n,0.05}(\hat{S}_\lambda, p)$	Mean Cross-Validated Error
0.2117	3	9.1389	6.4898	6.6519	1.0641
0.1929	4	9.4753	7.2464	7.4353	1.0450
0.1841	6	9.7273	8.4241	8.6061	1.0346
0.1678	7	10.1670	8.8959	9.0750	1.0092
0.1601	8	10.3675	9.3121	9.5102	0.9974
0.1459	9	10.7263	9.7115	9.9097	0.9751
0.1329	11	11.0739	10.3954	10.6071	0.9543
0.1269	12	11.2376	10.7042	10.9207	0.9452
0.1211	13	11.4330	10.9875	11.2085	0.9359
0.1104	14	11.7764	11.2576	11.4849	0.9186
0.1006	15	12.0756	11.5084	11.7407	0.9006
0.0960	17	12.2096	11.9664	12.2000	0.8934
0.0875	20	12.4788	12.5543	12.7891	0.8815
0.0761	25	12.9535	13.3824	13.6022	0.8651
0.0575	31	13.8675	14.1407	14.3703	0.8361
0.0456	40	14.5588	14.9712	15.2099	0.8255

included. Specifically, we show that the square root deviation of the S -restricted maximum log-likelihood from its baseline value under the null model can be well approximated by the ℓ_2 -norm of the normalized score vector. Second, based on a high-dimensional invariance principle, we prove the Gaussian/chi-squared approximation for the maximum of the ℓ_2 -norms of normalized score vectors. Finally, we apply an anti-concentration argument to construct non-asymptotic Wilks approximation for $2\{Q_n(s, p) - Q_n^*\}$.

Step 1: Wilks approximation. In the null model where Y and \mathbf{X} are independent, the true parameter β^* in (5) is zero, and thus the density function of Y has the form $f(y) = \exp\{-\phi^{-1}b(0) + c(y, \phi)\}$. Moreover, we have

$$\arg \max_{\beta \in \mathbb{R}^p} \mathbb{E}_{\mathbf{X}} \{ \mathcal{L}_n(\beta) \} = \arg \max_{\beta \in \mathbb{R}^p} \sum_{i=1}^n \mathbb{E}_{\mathbf{X}} \{ Y_i \mathbf{X}_i^T \beta - b(\mathbf{X}_i^T \beta) \} = \mathbf{0}.$$

To this see, note that in model (5) with $\beta^* = \mathbf{0}$, $\mathbb{E}(Y) = b'(0)$ and $\text{Var}(Y) = \phi b''(0)$. This implies that $\mathbb{E}_{\mathbf{X}} \{ \mathcal{L}_n(\beta) \} = \sum_{i=1}^n \{ b'(0) \mathbf{X}_i^T \beta - b(\mathbf{X}_i^T \beta) \}$. This function is strictly concave with respect to β and $\beta = \mathbf{0}$ satisfies its first order condition, and hence is its maximizer.

For each s -subset $S \subseteq [p]$, define the S -restricted log-likelihood $\mathcal{L}_n^S(\theta) = \sum_{i=1}^n \{ Y_i \mathbf{X}_{iS}^T \theta - b(\mathbf{X}_{iS}^T \theta) \}$ and the score function $\nabla \mathcal{L}_n^S(\theta) = \sum_{i=1}^n \{ Y_i - b'(\mathbf{X}_{iS}^T \theta) \} \mathbf{X}_{iS}$, $\theta \in \mathbb{R}^s$. In this

notation, it can be seen from (6) that

$$Q_n(s, p) = \max_{S \subseteq [p]: |S|=s} \max_{\boldsymbol{\theta} \in \mathbb{R}^s} \mathcal{L}_n^S(\boldsymbol{\theta}) = \max_{S \subseteq [p]: |S|=s} \mathcal{L}_n^S(\hat{\boldsymbol{\theta}}_S), \quad (24)$$

where

$$\hat{\boldsymbol{\theta}}_S = (\hat{\theta}_{S1}, \dots, \hat{\theta}_{Ss})^\top = \arg \max_{\boldsymbol{\theta} \in \mathbb{R}^s} \mathcal{L}_n^S(\boldsymbol{\theta}) \quad (25)$$

denotes the maximum likelihood estimate of the target parameter for the S -restricted model, which is given by $\boldsymbol{\theta}_S^* := \arg \max_{\boldsymbol{\theta} \in \mathbb{R}^s} \mathbb{E} \mathbf{X} \{ \mathcal{L}_n^S(\boldsymbol{\theta}) \} = \mathbf{0}$.

Given the i.i.d. observations $\{(Y_i, \mathbf{X}_i)\}_{i=1}^n$, $\nabla \mathbb{E} \mathbf{X} \{ \mathcal{L}_n^S(\boldsymbol{\theta}) \} = \sum_{i=1}^n \{ b'(0) - b(\mathbf{X}_{iS}^\top \boldsymbol{\theta}) \} \mathbf{X}_{iS}$ and $\mathbf{H}_S(\boldsymbol{\theta}) := -\nabla^2 \mathbb{E} \mathbf{X} \{ \mathcal{L}_n^S(\boldsymbol{\theta}) \} = \sum_{i=1}^n b''(\mathbf{X}_{iS}^\top \boldsymbol{\theta}) \mathbf{X}_{iS} \mathbf{X}_{iS}^\top$ for $\boldsymbol{\theta} \in \mathbb{R}^s$. In particular, write

$$\mathbf{H}_S^* := \mathbf{H}_S(\mathbf{0}) = n b''(0) \widehat{\boldsymbol{\Sigma}}_{SS} \quad (26)$$

for $\boldsymbol{\Sigma}_{SS}$ as in (12). Further, define the S -restricted normalized score

$$\hat{\boldsymbol{\xi}}_S = \mathbf{H}_S^{*-1/2} \nabla \mathcal{L}_n^S(\mathbf{0}) = \{ n b''(0) \}^{-1/2} \widehat{\boldsymbol{\Sigma}}_{SS}^{-1/2} \sum_{i=1}^n \varepsilon_i \mathbf{X}_{iS}, \quad \varepsilon_i = Y_i - b(0). \quad (27)$$

The following result is a conditional analogue of Corollary 1.12 in the supplement of Spokoiny (2012), which provides an exponential inequality for the ℓ_2 -norm of $\hat{\boldsymbol{\xi}}_S$ given $\{\mathbf{X}_i\}_{i=1}^n$. The proofs of this Lemma and other lemmas can be found in the appendix.

Lemma 9 *Assume that Conditions 3.1 and 3.2 hold. Then, for every $t \geq 0$,*

$$\mathbb{P} \mathbf{x} \{ \|\hat{\boldsymbol{\xi}}_S\|_2^2 \geq a_0 \phi \Delta(s, t) \} \leq 2e^{-t} \quad (28)$$

holds almost surely on the event $\{\widehat{\boldsymbol{\Sigma}}_{SS} \succ \mathbf{0}\}$, where

$$\Delta(s, t) := \begin{cases} s + (8ts)^{1/2}, & \text{if } 0 \leq t \leq \frac{1}{18}(2s)^{1/2}, \\ s + 6t, & \text{if } t > \frac{1}{18}(2s)^{1/2}. \end{cases} \quad (29)$$

The following lemma characterizes the Wilks phenomenon from a non-asymptotic perspective. Recall that $\hat{\boldsymbol{\theta}}_S$ at (25) is the S -restricted maximum likelihood estimator, and in the null model, $\mathcal{L}_n^S(\mathbf{0}) = \mathcal{L}_n(\mathbf{0}) = -nb(0)$, $\sigma_Y^2 = \text{Var}(Y) = \phi b''(0)$. For every $\tau > 0$, define the event

$$\mathcal{E}_0(\tau) = \bigcap_{S \subseteq [p]: |S|=s} \left\{ \widehat{\boldsymbol{\Sigma}}_{SS} \succ \mathbf{0}, \max_{1 \leq i \leq n} \mathbf{X}_{iS}^\top \widehat{\boldsymbol{\Sigma}}_{SS}^{-1} \mathbf{X}_{iS} \leq \tau \right\}. \quad (30)$$

Lemma 10 *Assume that Conditions 3.1 and 3.2 hold. Then, on the event $\mathcal{E}_0(\tau)$, for any $\tau > 0$,*

$$\mathbb{P} \mathbf{x} \left(\max_{S \subseteq [p]: |S|=s} \left| 2\{\mathcal{L}_n^S(\hat{\boldsymbol{\theta}}_S) - \mathcal{L}_n(\mathbf{0})\} \right|^{1/2} - \|\hat{\boldsymbol{\xi}}_S\|_2 \leq C_1 \phi \tau^{1/2} s \log(pm) \sqrt{\frac{s}{n}} \right) \leq 5n^{-1} \quad (31)$$

whenever $n \geq C_2 \phi \tau s \log(pm)$, where C_1 and C_2 are positive constants depending only on a_0, σ_1, A_1 and $b''(0)$.

To apply Lemma 10, we need to show first that for properly chosen τ , the event $\mathcal{E}_0(\tau)$ occurs with high probability. First, applying Theorem 5.39 in Vershynin (2012) to the random vectors $\boldsymbol{\Sigma}_{SS}^{-1/2} \mathbf{X}_{1S}, \dots, \boldsymbol{\Sigma}_{SS}^{-1/2} \mathbf{X}_{nS}$ yields that, for every $t \geq 0$,

$$\left\| \boldsymbol{\Sigma}_{SS}^{-1/2} \widehat{\boldsymbol{\Sigma}}_{SS} \boldsymbol{\Sigma}_{SS}^{-1/2} - \mathbf{I}_s \right\| = \left\| n^{-1} \boldsymbol{\Sigma}_{SS}^{-1/2} \mathbf{X}_S^\top \mathbf{X}_S \boldsymbol{\Sigma}_{SS}^{-1/2} - \mathbf{I}_s \right\| \leq \max(\delta, \delta^2) \quad (32)$$

holds with probability at least $1 - 2e^{-t}$, where $\delta = C_3(s \vee t)^{1/2} n^{-1/2}$, and $C_3 > 0$ is a constant depending only on A_0 . This, together with Boole's inequality implies by taking $t = s \log \frac{ep}{s} + \log n$ that, with probability at least $1 - 2n^{-1}$,

$$\max_{S \subseteq [p]: |S|=s} \left\| \boldsymbol{\Sigma}_{SS}^{-1/2} \widehat{\boldsymbol{\Sigma}}_{SS} \boldsymbol{\Sigma}_{SS}^{-1/2} - \mathbf{I}_s \right\| \leq C_3 \left(\frac{s \log \frac{ep}{s} + \log n}{n} \right) \leq \frac{1}{2} \quad (33)$$

whenever $n \geq 4C_3^2(s \log \frac{ep}{s} + \log n)$. Providing (33) holds, the smallest eigenvalue of $\boldsymbol{\Sigma}_{SS}^{-1/2} \widehat{\boldsymbol{\Sigma}}_{SS} \boldsymbol{\Sigma}_{SS}^{-1/2}$ is bounded from below by $\frac{1}{2}$ so that $\lambda_{\min}(\widehat{\boldsymbol{\Sigma}}_{SS}) \geq \frac{1}{2} \lambda_{\min}(\boldsymbol{\Sigma}_{SS})$. Moreover,

$$\mathbf{X}_{iS}^\top \widehat{\boldsymbol{\Sigma}}_{SS}^{-1} \mathbf{X}_{iS} \leq 2\lambda_{\min}^{-1}(\boldsymbol{\Sigma}_{SS}) \|\mathbf{X}_{iS}\|_2^2 \leq 2s\lambda_{\min}^{-1}(\boldsymbol{\Sigma}_{SS}) \max_{j \in S} X_{ij}^2. \quad (34)$$

For the last term on the right-hand side of (34), let $\mathbf{e}_j = (0, \dots, 0, 1, 0, \dots, 0)^\top$ be the unit vector in \mathbb{R}^p with 1 at the j th position and note that $\mathbf{X}_{ij} = \mathbf{e}_j^\top \mathbf{X}_i = \mathbf{e}_j^\top \boldsymbol{\Sigma}_{SS}^{1/2} \mathbf{U}_i$ with $\|\mathbf{e}_j^\top \boldsymbol{\Sigma}_{SS}^{1/2}\|_2 = 1$, where $\mathbf{U}_1, \dots, \mathbf{U}_n$ are i.i.d. p -dimensional random vectors with covariance matrix \mathbf{I}_p . By Condition 3.1, $\|\mathbf{X}_{ij}\|_{\psi_2} = \|\mathbf{e}_j^\top \boldsymbol{\Sigma}_{SS}^{1/2} \mathbf{U}_i\|_{\psi_2} \leq A_0$ and hence for every $t \geq 0$,

$$\mathbb{P} \left(\max_{1 \leq i \leq n} \max_{1 \leq j \leq p} X_{ij}^2 \geq t \right) \leq 2 \sum_{i=1}^n \sum_{j=1}^p \exp(-C_4^{-1}t) \leq 2 \exp\{\log(pm) - C_4^{-1}t\},$$

where $C_4 > 0$ is a constant depending only on A_0 . This, together with (34) implies by taking $t = 2C_4 \log(pm)$ that, with probability at least $1 - 3n^{-1}$,

$$\max_{1 \leq i \leq n} \max_{S \subseteq [p]: |S|=s} \mathbf{X}_{iS}^\top \widehat{\boldsymbol{\Sigma}}_{SS}^{-1} \mathbf{X}_{iS} \leq 2\lambda_{\min}^{-1}(s) \{1 + 2C_4 s \log(pm)\}. \quad (35)$$

Now, by (30) and (35), we take $\tau_0 = 2\lambda_{\min}^{-1}(s) \{1 + 2C_4 s \log(pm)\}$ such that the event $\mathcal{E}_0(\tau_0)$ occurs with probability greater than $1 - 3n^{-1}$ as long as $n \geq 4C_3^2(s \log \frac{ep}{s} + \log n)$. This, together with Lemma 10 yields that with probability at least $1 - 8n^{-1}$,

$$\max_{S \subseteq [p]: |S|=s} \left| 2\{\mathcal{L}_n^S(\hat{\boldsymbol{\theta}}_S) - \mathcal{L}_n(\mathbf{0})\} \right|^{1/2} - \|\hat{\boldsymbol{\xi}}_S\|_2 \leq C_5 \phi \lambda_{\min}^{-1/2}(s) \{s \log(pm)\}^{3/2} n^{-1/2} \quad (36)$$

whenever $n \geq C_6(1 \vee \phi) \lambda_{\min}^{-1}(s) \{s \log(pm)\}^2$, where $C_5, C_6 > 0$ are constants depending only on a_0, σ_1, A_0, A_1 and $b''(0)$.

Step 2: Gaussian approximation. For any $i = 1, \dots, n$ and $S \subseteq [p]$, define $\mathbf{Z}_i = \{b''(0)\}^{-1/2} \varepsilon_i \mathbf{X}_i$ and $\mathbf{Z}_{iS} = \{b''(0)\}^{-1/2} \varepsilon_i \mathbf{X}_{iS}$ such that $\hat{\boldsymbol{\xi}}_S = n^{-1/2} \sum_{i=1}^n \widehat{\boldsymbol{\Sigma}}_{SS}^{-1/2} \mathbf{Z}_{iS}$. Moreover, define

$$\boldsymbol{\xi} = n^{-1/2} \sum_{i=1}^n \mathbf{Z}_i \quad \text{and} \quad \boldsymbol{\xi}_S = n^{-1/2} \sum_{i=1}^n \boldsymbol{\Sigma}_{SS}^{-1/2} \mathbf{Z}_{iS}. \quad (37)$$

The following result shows that for each s -subset $S \subseteq [p]$, the ℓ_2 -norm of the S -restricted normalized score $\hat{\boldsymbol{\xi}}_S$ is close to that of $\boldsymbol{\xi}_S$ with overwhelmingly high probability.

Lemma 11 *Assume that Condition 3.1 holds. Then, for every s -subset $S \subseteq [p]$ and for every $0 \leq t \leq \frac{3}{4}(n - 2s)$,*

$$\mathbb{P}\left[\|\hat{\boldsymbol{\xi}}_S\|_2 - \|\boldsymbol{\xi}_S\|_2 > C_7\{s + t\}\phi\Delta(s, t)\}^{1/2}n^{-1/2}\right] \leq 12.4e^{-t}, \quad (38)$$

provided that $n \geq C_8(s + t)$, where $\Delta(s, t)$ is as in (29) and $C_7, C_8 > 0$ are constants depending only on a_0 and A_0 .

Using the union bound and taking $t = s \log \frac{e^2}{s} + \log n$ in Lemma 11, we see that with probability at least $1 - 12.4n^{-1}$,

$$\max_{S \subseteq [p]: |S|=s} \|\hat{\boldsymbol{\xi}}_S\|_2 - \|\boldsymbol{\xi}_S\|_2 \leq C_7\phi^{1/2}(s \log \frac{e^2}{s} + \log n)n^{-1/2} \quad (39)$$

whenever $n \geq C_9(s \log \frac{e^2}{s} + \log n)$.

Note that, the random vectors $\boldsymbol{\xi}$ and $\boldsymbol{\xi}_S$, $S \subseteq [p]$ defined in (37) satisfy $\mathbb{E}(\boldsymbol{\xi}) = \mathbf{0}$, $\mathbb{E}(\boldsymbol{\xi}_S^T) = \phi \boldsymbol{\Sigma}$, $\mathbb{E}(\boldsymbol{\xi}_S) = \mathbf{0}$ and $\mathbb{E}(\boldsymbol{\xi}_S \boldsymbol{\xi}_S^T) = \phi \mathbf{I}_s$. The following lemma provides a coupling inequality, showing that the random variable $\max_{S \subseteq [p]: |S|=s} \|\phi^{-1/2} \boldsymbol{\xi}_S\|_2$ can be well approximated, with high probability, by some random variable which is distributed as the maximum of the ℓ_2 -norms of a sequence of normalized Gaussian random vectors, that is, $\{\|\boldsymbol{\Sigma}_S^{-1/2} \mathbf{G}_S\|_2 : S \subseteq [p], |S|=s\}$.

Lemma 12 *Assume that Condition 3.1 holds. Then, there exists a random variable $T_0 \stackrel{d}{=} R_0(s, p)$ such that for any $\delta \in (0, 1]$,*

$$\left| \max_{S \subseteq [p]: |S|=s} \|\phi^{-1/2} \boldsymbol{\xi}_S\|_2 - T_0 \right| \leq C_{10}[\delta + \{s \log(\gamma_s p n)\}^{1/2}n^{-1/2} + \{s \log(\gamma_s p n)\}^2 n^{-3/2}] \quad (40)$$

holds with probability greater than $1 - C_{11}[\delta^{-3}n^{-1/2}\{s \log(\gamma_s p n)\}^2 \vee \delta^{-4}n^{-1}\{s \log(\gamma_s p n)\}^5]$, where $C_{10}, C_{11} > 0$ are constants depending only on a_0 and A_0

Step 3: Completion of the proof. We now apply an anti-concentration argument to construct the Berry-Esseen bound for the square root of the excess $2\phi^{-1}\{Q_n(s, p) - Q_n^*\}$. To this end, taking $\delta = \{s \log(\gamma_s p n)\}^{3/8}n^{-1/8}$ in Lemma 12 leads to that, with probability at least $1 - C_{11}\{s \log(\gamma_s p n)\}^{7/8}n^{-1/8}$,

$$\left| \max_{S \subseteq [p]: |S|=s} \|\phi^{-1/2} \boldsymbol{\xi}_S\|_2 - T_0 \right| \leq C_{12}\{s \log(\gamma_s p n)\}^{3/8}n^{-1/8} \quad (41)$$

whenever $n \geq \{s \log(\gamma_s p n)\}^3$. Further, for $R_0(s, p)$ in (14), note that

$$R_0^2(s, p) = \max_{S \subseteq [p]: |S|=s} \max_{\mathbf{u} \in \mathbb{S}^{s-1}} \frac{(\mathbf{u}^T \mathbf{G}_S)^2}{\mathbf{u}^T \boldsymbol{\Sigma}_S \mathbf{u}} = \max_{\mathbf{u} \in \mathcal{F}(s, p)} \frac{(\mathbf{u}^T \mathbf{G})^2}{\mathbf{u}^T \boldsymbol{\Sigma} \mathbf{u}},$$

where $\mathbf{G} \sim N(\mathbf{0}, \boldsymbol{\Sigma})$ and $\mathcal{F}(s, p) := \{\mathbf{x} \mapsto \mathbf{u}^T \mathbf{x} : \mathbf{u} \in \mathbb{S}^{p-1}, \|\mathbf{u}\|_0 \leq s\}$ is a class of linear functions $\mathbb{R}^p \mapsto \mathbb{R}$. Hence, it follows from Lemma 7.3 in Fan, Shao and Zhou (2015) with slight modification and Lemma A.1 in the supplement of Chernozhukov, Chetverikov and Kato (2014) that, for every $t > 0$,

$$\sup_{u \geq 0} \mathbb{P}\{|T_0 - u| \leq t\} = \sup_{u \geq 0} \mathbb{P}\{|R_0(s, p) - u| \leq t\} \leq C_{13}(s \log \frac{2se^2}{s})^{1/2}t, \quad (42)$$

where $C_{13} > 0$ is an absolute constant. Combining (42) with the preceding results (36), (39) and (41) proves (16). \blacksquare

6.2 Proof of Theorem 6

The main strategy of the proof is similar to that of Theorem 2 but technical details are substantially different. As before, we define the quasi-likelihood $\mathcal{L}_n(\boldsymbol{\beta}) = -\sum_{i=1}^n |Y_i - \mathbf{X}_i^T \boldsymbol{\beta}|$, $\boldsymbol{\beta} \in \mathbb{R}^p$, and observe that $\max_{\boldsymbol{\beta} \in \mathbb{R}^p: \|\boldsymbol{\beta}\|_0 \leq s} \mathcal{L}_n(\boldsymbol{\beta}) = \max_{S \subseteq [p]: |S|=s} \max_{\boldsymbol{\theta} \in \mathbb{R}^S} \mathcal{L}_n^S(\boldsymbol{\theta})$, where $\mathcal{L}_n^S(\boldsymbol{\theta}) = -\sum_{i=1}^n |Y_i - \mathbf{X}_{iS}^T \boldsymbol{\theta}|$. In the null model (7) with $\boldsymbol{\beta}^* = \mathbf{0}$, we have for each s -subset $S \subseteq [p]$, $\arg \max_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{X}} \{\mathcal{L}_n^S(\boldsymbol{\theta})\} = \mathbf{0}$ by the first order condition and concavity, and the S -restricted least absolute deviation estimator can be written as

$$\hat{\boldsymbol{\theta}}_S = \arg \max_{\boldsymbol{\theta} \in \mathbb{R}^S} \mathcal{L}_n^S(\boldsymbol{\theta}). \quad (43)$$

We first establish in Lemma 13 an upper bound for the maximum ℓ_2 -risks of $\hat{\boldsymbol{\theta}}_S$.

Lemma 13 *Assume that (18) holds and that $\mathbb{E}|\varepsilon|^\kappa < \infty$ for some $1 < \kappa \leq 2$. Then, on the event $\mathcal{E}_0(\tau)$ for $\tau > 0$, the sequence of LAD estimators $\{\hat{\boldsymbol{\theta}}_S : S \subseteq [p], |S|=s\}$ satisfies*

$$\max_{S \subseteq [p]: |S|=s} \|\hat{\boldsymbol{\Sigma}}_{SS}^{1/2} \hat{\boldsymbol{\theta}}_S\|_2 \leq C_1 a_2^{-1} \{s \log(pn)\}^{1/2} n^{-1/2} \quad (44)$$

with conditional probability (over the randomness of $\{\varepsilon_i\}_{i=1}^n$) greater than $1 - c_1 n^{-1} - c_2 n^{1-\kappa}$, where $C_1, c_1 > 0$ are absolute constants and $c_2 > 0$ is a constant depending only on $a_2, \kappa, \mathbb{E}|\varepsilon|$ and $\mathbb{E}|\varepsilon|^\kappa$.

Based on Lemma 13, we further study the concentration property of the Wilks expansion for the excess $\mathcal{L}_n^S(\hat{\boldsymbol{\theta}}_S) - \mathcal{L}_n^S(\mathbf{0})$. Since the function $\mathcal{L}_n^S(\cdot)$ is concave, we use $\nabla \mathcal{L}_n^S(\cdot)$ to denote its subgradient. For $\boldsymbol{\theta} \in \mathbb{R}^s$, let $\zeta^S(\boldsymbol{\theta}) = \mathcal{L}_n^S(\boldsymbol{\theta}) - \mathbb{E}_{\mathbf{X}} \mathcal{L}_n^S(\boldsymbol{\theta})$ be the stochastic component of $\mathcal{L}_n^S(\boldsymbol{\theta})$. Then, it is easy to see that

$$\nabla \zeta^S(\boldsymbol{\theta}) = -2 \sum_{i=1}^n w_i^S(\boldsymbol{\theta}) \mathbf{X}_{iS}, \quad \nabla \mathbb{E}_{\mathbf{X}} \mathcal{L}_n^S(\boldsymbol{\theta}) = - \sum_{i=1}^n \{2\mathbb{P}_{\mathbf{X}}(Y_i \leq \mathbf{X}_{iS}^T \boldsymbol{\theta}) - 1\} \mathbf{X}_{iS}, \quad (45)$$

where $w_i^S(\boldsymbol{\theta}) := I(Y_i \leq \mathbf{X}_{iS}^T \boldsymbol{\theta}) - \mathbb{P}_{\mathbf{X}}(Y_i \leq \mathbf{X}_{iS}^T \boldsymbol{\theta})$. In particular, we have $\nabla \zeta^S(\mathbf{0}) = -\sum_{i=1}^n \{2I(\varepsilon_i \leq 0) - 1\} \mathbf{X}_{iS}$. Recall that f_ε and F_ε denote, respectively, the density function and the cumulative distribution function of ε . By the second expression in (45), $\nabla \mathbb{E}_{\mathbf{X}} \mathcal{L}_n^S(\boldsymbol{\theta}) = -\sum_{i=1}^n \{2F_\varepsilon(\mathbf{X}_{iS}^T \boldsymbol{\theta}) - 1\} \mathbf{X}_{iS}$ and

$$\mathbf{H}_S(\boldsymbol{\theta}) := -\nabla^2 \mathbb{E}_{\mathbf{X}} \mathcal{L}_n^S(\boldsymbol{\theta}) = 2 \sum_{i=1}^n f_\varepsilon(\mathbf{X}_{iS}^T \boldsymbol{\theta}) \mathbf{X}_{iS} \mathbf{X}_{iS}^T. \quad (46)$$

In line with (26), we have $\mathbf{H}_S^* = \mathbf{H}_S(\mathbf{0}) = 2nf_\varepsilon(\mathbf{0})\widehat{\Sigma}_{SS}$, which is the negative Hessian of $\mathbb{E}_X \mathcal{L}_n^S(\mathbf{0})$. As in (27), define the normalized score

$$\widehat{\xi}_S = \mathbf{H}_S^{*-1/2} \nabla \mathcal{L}_n^S(\mathbf{0}) = \{2nf_\varepsilon(\mathbf{0})\}^{-1/2} \widehat{\Sigma}_{SS}^{-1/2} \sum_{i=1}^n \{2I(\varepsilon_i \leq 0) - 1\} \mathbf{X}_{iS}. \quad (47)$$

The following result is a non-asymptotic, conditional version of the Wilks theorem, saying that with high probability, the square root of the excess $\max_{\boldsymbol{\theta}} \mathcal{L}_n^S(\boldsymbol{\theta}) - \mathcal{L}_n^S(\mathbf{0})$ and the ℓ_2 -norm of the normalized score $\widehat{\xi}_S$ are sufficiently close uniformly over all s -subsets $S \subseteq [p]$.

Lemma 14 *Assume that Conditions 3.1 and 3.4 are satisfied. Then*

$$\begin{aligned} \max_{S \subseteq [p]; |S|=s} \left| 2\{\mathcal{L}_n^S(\widehat{\boldsymbol{\theta}}_S) - \mathcal{L}_n^S(\mathbf{0})\}^{1/2} - \|\widehat{\xi}_S\|_2 \right| \\ \leq C_2 \{f_\varepsilon(\mathbf{0})\}^{-1/2} [\lambda_{\min}^{-1/2}(s) \{s \log(pm)\}^{3/2} n^{-1/2} + \lambda_{\min}^{-1/4}(s) s \log(pm) n^{-1/4}] \end{aligned} \quad (48)$$

holds with probability greater than $1 - c_2 n^{1-\kappa} - c_3 n^{-1}$ whenever $n \geq C_3 \lambda_{\min}^{-1}(s) \{s \log(pm)\}^2$, where $C_2 > 0$ is a constant depending only on a_2, A_2 and A_3 , c_2 is as in Lemma 13, $c_3 > 0$ is an absolute constant and $C_3 > 0$ is a constant depending only on a_2 and A_2 .

Further, write $\tilde{\varepsilon}_i = 2I(\varepsilon_i \leq 0) - 1$ and $\widetilde{\mathbf{X}}_i = \tilde{\varepsilon}_i \mathbf{X}_i$. Note that $\tilde{\varepsilon}_1, \dots, \tilde{\varepsilon}_n$ are i.i.d. Rademacher random variables and thus $\widetilde{\mathbf{X}}_1, \dots, \widetilde{\mathbf{X}}_n$ are sub-exponential random vectors. In this notation, we have $\widehat{\xi}_S = \{2nf_\varepsilon(\mathbf{0})\}^{-1/2} \sum_{i=1}^n \widehat{\Sigma}_{SS}^{-1/2} \widetilde{\mathbf{X}}_{iS}$. For each $S \subseteq [p]$, define

$$\xi_S = \{2nf_\varepsilon(\mathbf{0})\}^{-1/2} \sum_{i=1}^n \widehat{\Sigma}_{SS}^{-1/2} \mathbf{X}_{iS}.$$

Then, applying Lemma 11 with slight modification and the union bound we obtain that, with probability at least $1 - c_4 n^{-1}$,

$$\max_{S \subseteq [p]; |S|=s} \left| \|\widehat{\xi}_S\|_2 - \|\xi_S\|_2 \right| \leq C_4 \{f_\varepsilon(\mathbf{0})\}^{-1/2} s \log(pm) n^{-1/2} \quad (49)$$

for all $n \geq C_5 s \log(pm)$, where $c_4 > 0$ is an absolute constant and $C_4, C_5 > 0$ are constants depending only on A_0 .

Observe that $\mathbb{E}(\widetilde{\mathbf{X}}_i) = \mathbb{E}[\mathbf{X}_i \{2\mathbb{P}(\varepsilon_i \leq 0) \mathbf{X}_i - 1\}] = 0$ and $\mathbb{E}(\widetilde{\mathbf{X}}_i \widetilde{\mathbf{X}}_i^\top) = \mathbb{E}(\mathbf{X}_i \mathbf{X}_i^\top) = \boldsymbol{\Sigma}$. Hence, it follows from Lemma 12 that there exists a random variable $T_0 \stackrel{d}{=} R_0(s, p)$ such that for any $\delta \in (0, 1]$,

$$\left| \sqrt{2f_\varepsilon(\mathbf{0})} \max_{S \subseteq [p]; |S|=s} \|\xi_S\|_2 - T_0 \right| \leq C_6 [\delta + \{s \log(\gamma_s pm)\}^{1/2} n^{-1/2} + \{s \log(\gamma_s pm)\}^2 n^{-3/2}] \quad (50)$$

holds with probability at least $1 - C_7 [\delta^{-3} n^{-1/2} \{s \log(\gamma_s pm)\}^2 \vee \delta^{-4} n^{-1} \{s \log(\gamma_s pm)\}^5]$, where $C_6, C_7 > 0$ are constants depending only on A_0 .

Finally, combining (48), (49), (50) and (42) proves (20). \blacksquare

Acknowledgments

We would like to acknowledge support for this project from the National Science Foundation Grants DMS-1206464 and DMS-1406266 and the National Institutes of Health Grant R01-GM072611-10.

References

- Dimitris Bertsimas, Angela King, and Rahul Mazumder. Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44(2):813–852, 2016.
- Peter Bühlmann and Sara van de Geer. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer-Verlag, Berlin Heidelberg, 2011.
- T. Tony Cai and Tiefeng Jiang. Phase transition in limiting distributions of coherence of high-dimensional random matrices. *Journal of Multivariate Analysis*, 107:24–39, 2012.
- T. Tony Cai, Jianqing Fan, and Tiefeng Jiang. Distributions of angles in random packing on spheres. *Journal of Machine Learning Research*, 14:1837–1864, 2013.
- Victor Chernozhukov, Denis Chetverikov, and Kengo Kato. Gaussian approximation of suprema of empirical processes. *The Annals of Statistics*, 42(4):1564–1597, 2014.
- Jianqing Fan, Shaojun Guo, and Ning Hao. Variance estimation using refitted cross-validation in ultrahigh dimensional regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(1):37–65, 2012.
- Jianqing Fan and Runze Li. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.
- Jianqing Fan, Han Liu, Qiang Sun, and Tong Zhang. TAC for sparse learning: Simultaneous control of algorithmic complexity and statistical error. *arXiv preprint arXiv:1507.01037*, 2015.
- Jianqing Fan, Qi-Man Shao, and Wen-Xin Zhou. Are discoveries spurious? Distributions of maximum spurious correlations and their applications. *arXiv preprint arXiv:1502.04237*, 2015.
- Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.
- Trevor Hastie, Robert Tibshirani, and Martin Wainwright *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press, 2015.
- Roger Koenker. *Quantile Regression*. Cambridge University Press, Cambridge, 2005.
- Kenneth Lange, David R. Hunter, and Iisoon Yang. Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics*, 9(1):1–20, 2000.

- Gangadharrao S. Maddala. *Limited-Dependent and Qualitative Variables in Econometrics*. Cambridge University Press, Cambridge, 1983.
- Peter McCullagh and John A. Nelder. *Generalized Linear Models*. Chapman & Hall/CRC, London, 1989.
- André Oberthner, Frank Berthold, Patrick Warnat, Barbara Hero, Yvonne Kahbert, Rüdiger Spitz, Karen Ernestus, Rahner König, Stefan Haas, Roland Eils, Manfred Schwab, Benedikt Brors, Frank Westermann, and Matthias Fischer. Customized oligonucleotide microarray gene expression based classification of neuroblastoma patients outperforms current clinical risk stratification. *Journal of Clinical Oncology*, 24(31):5070–5078, 2006.
- Lenning Shi, et al. (MAQC Consortium). The MicroArray Quality Control (MAQC)-II study of common practices for the development and validation of microarray-based predictive models. *Nature Biotechnology*, 28(8):827–841, 2010.
- Vladimir Spokoiny. Parametric estimation. Finite sample theory. *The Annals of Statistics*, 40(6):2877–2909, 2012.
- Vladimir Spokoiny. Bernstein-von Mises theorem for growing parameter dimension. *arXiv preprint arXiv:1302.3430*, 2013.
- Vladimir Spokoiny and Mayya Zhilova. Bootstrap confidence sets under model misspecification. *The Annals of Statistics*, 43(6):2653–2675, 2015.
- Tingni Sun and Cun-Hui Zhang. Scaled sparse linear regression. *Biometrika*, 99(4):879–898, 2012.
- Gundamrur A. Thorisson, Albert V. Smith, Lalitha Krishnan, and Lincoln D. Stein. The International HapMap Project Web site. *Genome Research*, 15:1592–1593, 2005.
- Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 58(1):267–288, 1996.
- Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.
- Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. In *Compressed Sensing: Theory and Applications*, pages 210–268, Cambridge University Press, Cambridge, 2012.
- Lie Wang. The L_1 penalized LAD estimator for high dimensional linear regression. *Journal of Multivariate Analysis*, 120:135–151, 2013.
- Samuel S. Wilks. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The Annals of Mathematical Statistics*, 9(1):60–62, 1938.
- Hui Zou and Runze Li. One-step sparse estimates in nonconcave penalized likelihood models. *The Annals of Statistics*, 36(4):1509–1533, 2008.

Appendix A. Appendix A.

In this appendix we prove the technical lemmas appeared in Section 6.

A.1 Proof of Lemma 9

Define the loss function $\ell(y, z) = yz - b(z)$ for $y, z \in \mathbb{R}$. For each s -subset $S \subseteq [p]$ and $\theta \in \mathbb{R}^s$, define $\zeta^S(\theta) = L_n^S(\theta) - \mathbb{E} \mathbf{X} \mathcal{L}_n^S(\theta) = \sum_{i=1}^n \zeta_i^S(\theta)$, where $\zeta_i^S(\theta) = \ell(Y_i, \mathbf{X}_{iS}^T \theta) - \mathbb{E} \mathbf{X} \ell(Y_i, \mathbf{X}_{iS}^T \theta)$. Note that $\nabla \zeta_i^S(\theta)|_{\theta=0} = \varepsilon_i \mathbf{X}_{iS}$ with $\varepsilon_i = Y_i - b(0)$. Thus, we have $\mathbf{V}_0^S := \text{Var} \mathbf{X} \{\nabla \zeta^S(\mathbf{0})\} = n\phi^{S'}(0) \widehat{\Sigma}_{SS}$.

For every $\mathbf{u} \in \mathbb{R}^s \setminus \{\mathbf{0}\}$ and $u \in \mathbb{R}$,

$$\begin{aligned} \mathbb{E} \mathbf{X} \exp \left\{ u \frac{\mathbf{u}^T \nabla \zeta^S(\mathbf{0})}{\|\mathbf{V}_0 \mathbf{u}\|_2} \right\} &= \prod_{i=1}^n \mathbb{E} \mathbf{X} \exp \left(u \frac{\mathbf{u}^T \mathbf{X}_{iS} \varepsilon_i}{\|\mathbf{V}_0 \mathbf{u}\|_2} \right) \\ &= \prod_{i=1}^n \mathbb{E} \mathbf{X} \exp \left\{ \frac{u}{\sqrt{n}} \times \frac{\mathbf{u}^T \mathbf{X}_{iS}}{(\mathbf{u}^T \widehat{\Sigma}_{SS} \mathbf{u})^{1/2}} \times \frac{\varepsilon_i}{(\text{Var } \varepsilon_i)^{1/2}} \right\} \\ &\leq \exp \left\{ \frac{1}{2} a_0 u^2 \times \frac{1}{n} \sum_{i=1}^n \frac{(\mathbf{u}^T \mathbf{X}_{iS})^2}{\mathbf{u}^T \widehat{\Sigma}_{SS} \mathbf{u}} \right\} = \exp(a_0 u^2 / 2). \end{aligned}$$

This verifies condition (ED_0) with $\eta_0^S = a_0$ in Theorem B.3 from the supplement of Spokoiny and Zhilova (2015). Consequently, taking $\mathbb{B}^2 = \mathbf{H}_S^{-1/2} \mathbf{V}_0^2 \mathbf{H}_S^{*-1/2} = \phi \mathbf{I}_s$ and $\mathbf{g} = \{C \text{tr}(\mathbb{B}^2)\}^{1/2}$ for some $C \geq 2$ there, we have $\lambda_{\max}(\mathbb{B}^2) = \phi$, $\text{tr}(\mathbb{B}^2) = \phi s$, $\text{tr}(\mathbb{B}^4) = \phi^2 s^2$ and $\mathbf{x}_c = \frac{1}{2}(\frac{2}{3}C - 1 - \log 3)s \geq \frac{3}{4}(C - 2)s$. This implies that almost surely on the event $\{\widehat{\Sigma}_{SS} \succ \mathbf{0}\}$, with conditional probability at least $1 - 2e^{-t} - 8.4 e^{-x_c}$,

$$\|\widehat{\xi}_s\|_2^2 \leq a_0 \phi \times \begin{cases} s + (8ts)^{1/2}, & \text{if } 0 \leq t \leq \frac{1}{18}(2s)^{1/2}, \\ s + 6t, & \text{if } \frac{1}{18}(2s)^{1/2} < t \leq x_c. \end{cases}$$

Finally, letting $C \rightarrow \infty$ proves (28). ■

A.2 Proof of Lemma 10

We prove this lemma by applying the conditional version of Theorem 2.3 in Spokoiny (2013). To this end, we need to verify conditions (ED_0) , (ED_2) , (\mathcal{L}_0) , (\mathcal{I}) and (\mathcal{L}) . In line with the notation used therein, we fix $S \subseteq [p]$ and write

$$\mathbf{D}^2(\theta) = -\nabla^2 \mathbb{E} \mathbf{X} \{\mathcal{L}_n^S(\theta)\} = \sum_{i=1}^n \theta^{i'} (\mathbf{X}_{iS}^T \theta) \mathbf{X}_{iS} \mathbf{X}_{iS}^T, \quad \mathbf{D}_0^2 = \mathbf{D}^2(\mathbf{0}) = n\theta^{i'}(0) \widehat{\Sigma}_{SS}.$$

The validity of (ED_0) is guaranteed from the proof of Lemma 9, and (ED_2) is automatically satisfied with $\omega \equiv 0$ since $\nabla^2 \zeta^S(\theta)$ vanishes for all $\theta \in \mathbb{R}^s$. Turning to (\mathcal{L}_0) , observe

that

$$\begin{aligned}
& \|\mathbf{D}_0^{-1} \mathbf{D}^2(\boldsymbol{\theta}) \mathbf{D}_0^{-1} - \mathbf{I}_s\| \\
&= \left\| \mathbf{D}_0^{-1} \sum_{i=1}^n \{b''(\mathbf{X}_{iS}^T \boldsymbol{\theta}) - b''(0)\} \mathbf{X}_{iS} \mathbf{X}_{iS}^T \mathbf{D}_0^{-1} \right\| \\
&= \left\| \mathbf{D}_0^{-1} \sum_{i=1}^n b'''(\eta_i) \mathbf{X}_{iS}^T \boldsymbol{\theta} \mathbf{X}_{iS} \mathbf{X}_{iS}^T \mathbf{D}_0^{-1} \right\|, \tag{51}
\end{aligned}$$

where η_i lies between 0 and $\mathbf{X}_{iS}^T \boldsymbol{\theta}$. For $r > 0$, define $\Theta_0(r) = \{\boldsymbol{\theta} \in \mathbb{R}^s : \|\mathbf{D}_0 \boldsymbol{\theta}\|_2 \leq r\}$. On the event $\mathcal{E}_0(\tau)$ for some $\tau > 0$ and for $\boldsymbol{\theta} \in \Theta_0(r)$,

$$\|\mathbf{X}_{iS}^T \boldsymbol{\theta}\| = |\boldsymbol{\theta}^T \mathbf{D}_0 \mathbf{D}_0^{-1} \mathbf{X}_{iS}| \leq \|\mathbf{D}_0^{-1} \mathbf{X}_{iS}\|_2 \leq \{nb''(0)\}^{-1/2} \tau^{1/2} r. \tag{52}$$

This together with (51) implies that

$$\|\mathbf{D}_0^{-1} \mathbf{D}^2(\boldsymbol{\theta}) \mathbf{D}_0^{-1} - \mathbf{I}_s\| \leq \frac{\max_{|\mu| \leq \{nb''(0)\}^{-1/2} \tau^{1/2} r} |b'''(\mu)|}{\{b''(0)\}^{3/2}} \tau^{1/2} r := \delta(\tau, r). \tag{53}$$

Recalling that $\mathbf{V}_2^S = \text{Var} \mathbf{x}\{\mathcal{L}_n^S(\mathbf{0})\} = \phi \mathbf{D}_0^S$, (\mathcal{L}) is satisfied with $a = \phi^{1/2}$.

To verify $(\mathcal{L}v)$, define $g(t) = b'(0)t - b(t)$ so that $g'(t) = b'(0) - b'(t)$ and $g''(t) = -b''(t)$. Then, for any $\boldsymbol{\theta} \in \mathbb{R}^s$ satisfying $\|\mathbf{D}_0 \boldsymbol{\theta}\|_2 = r > 0$, it follows from the second-order Taylor expansion that

$$\begin{aligned}
& -2\{\mathbb{E} \mathbf{x} \mathcal{L}_n^S(\boldsymbol{\theta}) - \mathbb{E} \mathbf{x} \mathcal{L}_n^S(\mathbf{0})\} = -2 \sum_{i=1}^n \{g(\mathbf{X}_{iS}^T \boldsymbol{\theta}) - g(0)\} \\
&= -2 \sum_{i=1}^n \{g'(0) \mathbf{X}_{iS}^T \boldsymbol{\theta} + \frac{1}{2} g''(\eta_i) (\mathbf{X}_{iS}^T \boldsymbol{\theta})^2\} = \sum_{i=1}^n b''(\eta_i) (\mathbf{X}_{iS}^T \boldsymbol{\theta})^2, \tag{54}
\end{aligned}$$

where η_i is a point lying between 0 and $\mathbf{X}_{iS}^T \boldsymbol{\theta}$. On the event $\mathcal{E}_0(\tau)$, the right-hand side of (54) is further bounded from below by

$$r^2 \{b''(0)\}^{-1} \min_{|\mu| \leq \{nb''(0)\}^{-1/2} \tau^{1/2} r} b''(\mu).$$

When $\|\mathbf{D}_0 \boldsymbol{\theta}\|_2 = r \leq \{nb''(0)/\tau\}^{1/2}$, $-2\{\mathbb{E} \mathbf{x} \mathcal{L}_n^S(\boldsymbol{\theta}) - \mathbb{E} \mathbf{x} \mathcal{L}_n^S(\mathbf{0})\}$ is bounded from below by $a_1 r^2$ for a_1 as in (15). Further, from the convexity of the function $\boldsymbol{\theta} \mapsto -\mathbb{E} \mathbf{x}\{\mathcal{L}_n^S(\boldsymbol{\theta}) - \mathcal{L}_n^S(\mathbf{0})\}$, we see that $-\mathbb{E} \mathbf{x}\{\mathcal{L}_n^S(\boldsymbol{\theta}) - \mathcal{L}_n^S(\mathbf{0})\} \geq a_1 r \{nb''(0)/\tau\}^{1/2}$, for all $\boldsymbol{\theta}$ satisfying $\|\mathbf{D}_0 \boldsymbol{\theta}\|_2 = r \geq \{nb''(0)/\tau\}^{1/2}$. Define the function $r \mapsto b(r)$ as

$$b(r) = \begin{cases} a_1 & \text{if } 0 \leq r \leq \{nb''(0)/\tau\}^{1/2}, \\ a_1 r^{-1} \{nb''(0)/\tau\}^{1/2} & \text{if } r > \{nb''(0)/\tau\}^{1/2}. \end{cases} \tag{55}$$

By definition, $r\dot{b}(r)$ is non-decreasing in $r \geq 0$ and for $\boldsymbol{\theta} \in \mathbb{R}^s$ satisfying $\|\mathbf{D}_0 \boldsymbol{\theta}\|_2 = r$,

$$-\frac{2\mathbb{E} \mathbf{x}\{\mathcal{L}_n^S(\boldsymbol{\theta}) - \mathcal{L}_n^S(\mathbf{0})\}}{\|\mathbf{D}_0 \boldsymbol{\theta}\|_2^2} \geq b(r). \tag{56}$$

With the above preparations, we apply Theorem 2.3 in Spokoiny (2013) with slight modification on the constant. In view of (29) and (55), set

$$r_0 = 2(\phi a_0)^{1/2} a_1^{-1} [s + 6(s \log \frac{\phi}{s} + \log n)]^{1/2}, \tag{57}$$

such that Condition 2.3 there is satisfied on $\mathcal{E}_0(\tau)$ whenever $n \geq \{b''(0)\}^{-1} r_0^2 \tau$. Hence, it follows from Theorem 2.3 in Spokoiny (2013) and the union bound that, conditional on the event $\mathcal{E}_0(\tau)$,

$$\mathbb{P} \mathbf{x} \left(\max_{S \subseteq [p]: |S|=s} |2\{\mathcal{L}_n^S(\widehat{\boldsymbol{\theta}}_S) - \mathcal{L}_n^S(\mathbf{0})\}|^{1/2} - \|\widehat{\boldsymbol{\xi}}_S\|_2 \leq 5\delta(\tau, r_0) r_0 \right) \leq 5n^{-1}, \tag{58}$$

where $\delta(\tau, r)$ and r_0 are as in (53) and (57), respectively. This proves (31) by properly choosing C_1 and C_2 . \blacksquare

A.3 Proof of Lemma 11

To begin with, note that for each s -subset $S \subseteq [p]$, $\mathbf{Z}_{iS}, \dots, \mathbf{Z}_{nS}$ are i.i.d. s -dimensional random vectors with mean zero and covariance matrix $\phi \boldsymbol{\Sigma}_{SS}$. By (27) and (37),

$$\|\widehat{\boldsymbol{\xi}}_S\|_2^2 - \|\boldsymbol{\xi}_S\|_2^2 = \boldsymbol{\xi}_S^T (\boldsymbol{\Sigma}_{SS}^{1/2} \widehat{\boldsymbol{\Sigma}}_{SS}^{-1} \boldsymbol{\Sigma}_{SS}^{1/2} - \mathbf{I}_s) \boldsymbol{\xi}_S.$$

Write $\mathbb{X}_S = (\mathbf{X}_{1S}, \dots, \mathbf{X}_{nS})^T \in \mathbb{R}^{n \times s}$, then $\mathbb{X}_S \boldsymbol{\Sigma}_{SS}^{-1/2}$ is an $n \times s$ matrix whose rows are independent sub-Gaussian random vectors in \mathbb{R}^s . Further, observe that $\mathbf{X}_{iS} = \mathbf{P}_S \mathbf{X}_i$ and $\boldsymbol{\Sigma}_{SS} = \mathbf{P}_S \boldsymbol{\Sigma} \mathbf{P}_S^T$, where $\mathbf{P}_S \in \mathbb{R}^{s \times p}$ is a projection matrix. Under Condition 3.1, $\|\mathbf{u}^T \boldsymbol{\Sigma}_{SS}^{-1/2} \mathbf{X}_{iS}\|_{\psi_2} = \|\mathbf{u}^T \boldsymbol{\Sigma}_{SS}^{-1/2} \mathbf{P}_S \boldsymbol{\Sigma}_{SS}^{1/2} \mathbf{U}\|_{\psi_2} \leq A_0 \|\boldsymbol{\Sigma}_{SS}^{1/2} \mathbf{P}_S^T \boldsymbol{\Sigma}_{SS}^{-1/2} \mathbf{u}\|_2 = A_0$ for $\mathbf{u} \in \mathbb{S}^{s-1}$. Then, it follows from (32) that for all sufficient large n so that $\delta \leq \frac{1}{2}$, $\|\boldsymbol{\Sigma}_{SS}^{1/2} \widehat{\boldsymbol{\Sigma}}_{SS}^{-1} \boldsymbol{\Sigma}_{SS}^{1/2} - \mathbf{I}_s\| \leq 2\delta$ and hence,

$$\begin{aligned}
\|\widehat{\boldsymbol{\xi}}_S\|_2 - \|\boldsymbol{\xi}_S\|_2 &= \frac{\|\widehat{\boldsymbol{\xi}}_S\|_2^2 - \|\boldsymbol{\xi}_S\|_2^2}{\|\widehat{\boldsymbol{\xi}}_S\|_2 + \|\boldsymbol{\xi}_S\|_2} \\
&\leq \|\boldsymbol{\xi}_S\|_2^{-1} \times \|\widehat{\boldsymbol{\xi}}_S\|_2^2 - \|\boldsymbol{\xi}_S\|_2^2 \leq 2C_3 (s \vee t)^{1/2} n^{-1/2} \times \|\boldsymbol{\xi}_S\|_2. \tag{59}
\end{aligned}$$

Next we upper bound the quadratic term $\|\boldsymbol{\xi}_S\|_2$. First we show that $\boldsymbol{\Sigma}_{SS}^{-1/2} \mathbf{Z}_{iS} = \phi^{1/2} \boldsymbol{\Sigma}_{SS}^{-1/2} \widetilde{\boldsymbol{\varepsilon}}_i \mathbf{X}_i$ are sub-exponential random vectors, where $\widetilde{\boldsymbol{\varepsilon}}_i := \boldsymbol{\varepsilon}_i / (\text{Var } \boldsymbol{\varepsilon}_i)^{1/2}$. In fact, for every $\mathbf{u} \in \mathbb{S}^{s-1}$, $\|\mathbf{u}^T \boldsymbol{\Sigma}_{SS}^{-1/2} \mathbf{Z}_{iS}\|_{\psi_1} \leq 2\|\widetilde{\boldsymbol{\varepsilon}}_i\|_{\psi_2} \|\mathbf{u}^T \boldsymbol{\Sigma}_{SS}^{-1/2} \mathbf{X}_i\|_{\psi_2} \leq 2A_0 A_0$, where $A_0 > 0$ is a constant depending only on a_0 in Condition 3.1. Following the proof of Lemma 5.15 in Vershynin (2012), we derive that for every $\mathbf{u} \in \mathbb{R}^s$ satisfying $\|\mathbf{u}\|_2 \leq \phi^{-1/2} (4eA_0' A_0)^{-1} \sqrt{n}$,

$$\begin{aligned}
\log \mathbb{E} \exp(\mathbf{u}^T \boldsymbol{\xi}_S) &= \sum_{i=1}^n \log \mathbb{E} \exp(n^{-1/2} \mathbf{u}^T \boldsymbol{\Sigma}_{SS}^{-1/2} \mathbf{Z}_{iS}) \\
&\leq 2e^2 \|\mathbf{u}\|_2^2 n^{-1} \sum_{i=1}^n \|\mathbf{u} / \|\mathbf{u}\|_2\|_2^T \boldsymbol{\Sigma}_{SS}^{-1/2} \mathbf{Z}_{iS}\|_{\psi_1}^2 \\
&\leq (4eA_0' A_0)^2 \phi \frac{\|\mathbf{u}\|_2^2}{2}.
\end{aligned}$$

Consequently, applying Corollary 1.12 in the supplement of Spokoiny (2012) with $\mathbf{g} = \sqrt{n}$, $\mathbb{B} = \mathbf{I}_s$ and $\mathbf{x}_c = \frac{3}{4}n - \frac{1}{2}(1 + \log 3)s \geq \frac{3}{4}n - \frac{3}{2}s$ to the random vector $(4eA_0^*A_0)^{-1}\phi^{-1/2}\boldsymbol{\xi}_S$ yields that, for every $0 \leq t \leq \mathbf{x}_c$,

$$\mathbb{P}\{\|\boldsymbol{\xi}_S\|_2 \geq 4eA_0^*A_0\{\phi\Delta(s,t)\}^{1/2}\} \leq 2e^{-t} + 8.4e^{-\mathbf{x}_c}. \quad (60)$$

Finally, combining (59) and (60) completes the proof of (38). ■

A.4 Proof of Lemma 12

First, observe that

$$\max_{S \subseteq [p]: |S|=s} \|\boldsymbol{\xi}_S\|_2 = \max_{\mathbf{u} \in \mathcal{F}(s,p)} n^{-1/2} \sum_{i=1}^n \frac{\mathbf{u}^T \mathbf{Z}_i}{(\mathbf{u}^T \boldsymbol{\Sigma} \mathbf{u})^{1/2}},$$

where $\mathcal{F}(s,p) = \{\mathbf{x} \mapsto \mathbf{u}^T \mathbf{x} : \mathbf{u} \in \mathbb{S}^{p-1}, \|\mathbf{u}\|_0 \leq s\}$. Recall that $\mathbf{Z}_1, \dots, \mathbf{Z}_n$ are i.i.d. p -dimensional centered random vectors with covariance matrix $\mathbb{E}[\mathbf{Z}_i \mathbf{Z}_i^T] = \phi \boldsymbol{\Sigma}$. As in the proof of Lemma 11, we have for any $\mathbf{u} \in \mathbb{S}^{p-1}$,

$$\|\phi^{-1/2} \mathbf{u}^T \mathbf{Z}_i\|_{\psi_1} \leq 2\|\varepsilon_i / (\text{Var } \varepsilon_i)^{1/2}\|_{\psi_2} \|\mathbf{u}^T \boldsymbol{\Sigma}^{1/2} \mathbf{T}_i\|_{\psi_2} \leq 2A_0^*A_0(\mathbf{u}^T \boldsymbol{\Sigma} \mathbf{u})^{1/2}.$$

Consequently, it follows from Lemma 7.5 in Fan, Shao and Zhou (2015) that there exists a random variable $T_0 \stackrel{d}{=} R_0(s,p) = \max_{\mathbf{u} \in \mathcal{F}(s,p)} \frac{\mathbf{u}^T \mathbf{G}}{(\mathbf{u}^T \boldsymbol{\Sigma} \mathbf{u})^{1/2}}$ for $\mathbf{G} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ such that, for any $\delta \in (0, 1]$,

$$\begin{aligned} \mathbb{P}\left\{ \max_{S \subseteq [p]: |S|=s} \|\phi^{-1/2} \boldsymbol{\xi}_S\|_2 - T_0 \geq C_1 A_0^* A_0 \left(\delta + \frac{\gamma_{s,p}^{1/2}}{\sqrt{n}} + \frac{\gamma_{s,p}^2}{n^{3/2}} \right) \right\} \\ \leq C_2 \left[\frac{\{s \log(\gamma_{s,p})\}^2}{\delta^3 \sqrt{n}} + \frac{\{s \log(\gamma_{s,p})\}^5}{\delta^4 n} \right], \end{aligned}$$

where $\gamma_{s,p} = s \log \frac{\gamma_{s,p}}{s} + \log n$ and $C_1, C_2 > 0$ are absolute constants. This proves (40). ■

A.5 Proof of Lemma 13

The proof employs techniques from empirical process theory which modify the arguments used in Wang (2013). To begin with, note that

$$\hat{\boldsymbol{\theta}}_S = \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^s} f(\boldsymbol{\theta}) := \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^s} \|\mathbf{Y} - \mathbb{X}_S \boldsymbol{\theta}\|_1.$$

Under the null model, $\mathbf{Y} = \mathbb{X}_S \boldsymbol{\theta}^* + \boldsymbol{\varepsilon}$ with $\boldsymbol{\theta}^* = \mathbf{0}$. Then the sub-differential of $f(\boldsymbol{\theta})$ at $\boldsymbol{\theta} = \mathbf{0}$ can be written as $\nabla f(\mathbf{0}) = -\mathbb{X}_S^T \text{sgn}(\boldsymbol{\varepsilon})$, where $\text{sgn}(\boldsymbol{\varepsilon}) = (\text{sgn}(\varepsilon_1), \dots, \text{sgn}(\varepsilon_n))^T$ with $\text{sgn}(u) := I(u > 0) - I(u < 0)$. Define $\mathbf{z} = (z_1, \dots, z_n)^T = \text{sgn}(\boldsymbol{\varepsilon})$, and note that z_1, \dots, z_n are i.i.d. random variables satisfying $\mathbb{P}(z_i = 1) = \mathbb{P}(z_i = -1) = 1/2$. Since $\hat{\boldsymbol{\theta}}_S$ minimizes $\|\mathbf{Y} - \mathbb{X}_S \boldsymbol{\theta}\|_1$ over \mathbb{R}^s , we have the following basic inequality

$$\|\mathbf{Y} - \mathbb{X}_S \hat{\boldsymbol{\theta}}_S\|_1 = \|\mathbb{X}_S \hat{\boldsymbol{\theta}}_S - \boldsymbol{\varepsilon}\|_1 \leq \|\boldsymbol{\varepsilon}\|_1. \quad (61)$$

Further, define a random process $\{Q(\boldsymbol{\theta})\}$ indexed by $\boldsymbol{\theta} \in \mathbb{R}^s$:

$$Q(\boldsymbol{\theta}) = n^{-1/2} \sum_{i=1}^n (\mathbf{X}_{iS}^T \boldsymbol{\theta} - \varepsilon_i) - |\varepsilon_i|. \quad (62)$$

In what follows, we prove that with overwhelmingly high probability, $Q(\boldsymbol{\theta})$ is concentrated around its expectation $Q_{\mathbb{X}}(\boldsymbol{\theta}) := \mathbb{E}_{\mathbf{X}}\{Q(\boldsymbol{\theta})\}$ uniformly over $\boldsymbol{\theta} \in \mathbb{R}^s$ via a straightforward adaptation of the peeling argument.

For $\delta_1 > 0$ and $\ell = 1, 2, \dots$, consider the following sequence of events

$$\mathcal{G}(\delta_1) = \{\boldsymbol{\theta} \in \mathbb{R}^s : \|\widehat{\boldsymbol{\Sigma}}_{SS}^{1/2} \boldsymbol{\theta}\|_2 \geq \delta_1\}, \quad \mathcal{G}_\ell(\delta_1) = \{\boldsymbol{\theta} \in \mathbb{R}^s : \alpha^{\ell-1} \delta_1 \leq \|\widehat{\boldsymbol{\Sigma}}_{SS}^{1/2} \boldsymbol{\theta}\|_2 \leq \alpha^\ell \delta_1\}, \quad (63)$$

where $\alpha = \sqrt{2}$. Here, δ_1 can be regarded as a tolerance parameter, and it is easy to see that $\mathcal{G}(\delta_1) = \cup_{\ell=1}^{\infty} \mathcal{G}_\ell(\delta_1)$. For $R > 0$, set $\mathcal{V}(R) = \{\boldsymbol{\theta} \in \mathcal{G}(\delta_1) : \|\widehat{\boldsymbol{\Sigma}}_{SS}^{1/2} \boldsymbol{\theta}\|_2 \leq R\}$ and let $\Delta(R)$ be the maximum deviation over the elliptic vicinity $\mathcal{V}(R)$:

$$\Delta(R) = \max_{\boldsymbol{\theta} \in \mathcal{V}(R)} |Q(\boldsymbol{\theta}) - Q_{\mathbb{X}}(\boldsymbol{\theta})|. \quad (64)$$

For every $\boldsymbol{\theta} \in \mathbb{R}^s$, define the rescaled vector $\tilde{\boldsymbol{\theta}} = \widehat{\boldsymbol{\Sigma}}_{SS}^{1/2} \boldsymbol{\theta}$ such that

$$\Delta(R) = \max_{\delta_1 \leq \|\tilde{\boldsymbol{\theta}}\|_2 \leq R} |Q(\widehat{\boldsymbol{\Sigma}}_{SS}^{-1/2} \tilde{\boldsymbol{\theta}}) - Q_{\mathbb{X}}(\widehat{\boldsymbol{\Sigma}}_{SS}^{-1/2} \tilde{\boldsymbol{\theta}})|.$$

For every $0 < \epsilon \leq R$, there exists an ϵ -net \mathcal{N}_ϵ of the Euclidean ball $\mathbb{B}_{\mathbb{R}^s}(R)$ with cardinality bounded by $(1 + \frac{2R}{\epsilon})^s$. For $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathbb{B}_{\mathbb{R}^s}(R)$ satisfying $\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2 \leq \epsilon$, observe that

$$\begin{aligned} \left| Q(\widehat{\boldsymbol{\Sigma}}_{SS}^{-1/2} \tilde{\boldsymbol{\theta}}_1) - Q(\widehat{\boldsymbol{\Sigma}}_{SS}^{-1/2} \tilde{\boldsymbol{\theta}}_2) \right| &\leq n^{-1/2} \sum_{i=1}^n \left| \mathbf{X}_{iS}^T \widehat{\boldsymbol{\Sigma}}_{SS}^{-1/2} (\tilde{\boldsymbol{\theta}}_1 - \tilde{\boldsymbol{\theta}}_2) \right| \\ &\leq \|\mathbb{X}_S \widehat{\boldsymbol{\Sigma}}_{SS}^{-1/2} (\tilde{\boldsymbol{\theta}}_1 - \tilde{\boldsymbol{\theta}}_2)\|_2 \leq \epsilon n^{1/2}. \end{aligned}$$

Then, it is easy to see that

$$\Delta(R) \leq \max_{\tilde{\boldsymbol{\theta}} \in \mathcal{N}_\epsilon} |Q(\widehat{\boldsymbol{\Sigma}}_{SS}^{-1/2} \tilde{\boldsymbol{\theta}}) - Q_{\mathbb{X}}(\widehat{\boldsymbol{\Sigma}}_{SS}^{-1/2} \tilde{\boldsymbol{\theta}})| + 2\epsilon n^{1/2}. \quad (65)$$

For each $\tilde{\boldsymbol{\theta}} \in \mathbb{B}_{\mathbb{R}^s}(R)$ fixed, $Q(\widehat{\boldsymbol{\Sigma}}_{SS}^{-1/2} \tilde{\boldsymbol{\theta}}) - Q_{\mathbb{X}}(\widehat{\boldsymbol{\Sigma}}_{SS}^{-1/2} \tilde{\boldsymbol{\theta}})$ is a sum of independent random variables with zero means and for $i = 1, \dots, n$, $\|\mathbf{X}_{iS}^T \widehat{\boldsymbol{\Sigma}}_{SS}^{-1/2} \tilde{\boldsymbol{\theta}} - \varepsilon_i\| \leq \|\mathbf{X}_{iS}^T \widehat{\boldsymbol{\Sigma}}_{SS}^{-1/2} \tilde{\boldsymbol{\theta}}\|$. Therefore, it follows from Hoeffding's inequality that for every $t > 0$,

$$\begin{aligned} \mathbb{P}_{\mathbf{X}} \left\{ \left| Q(\widehat{\boldsymbol{\Sigma}}_{SS}^{-1/2} \tilde{\boldsymbol{\theta}}) - Q_{\mathbb{X}}(\widehat{\boldsymbol{\Sigma}}_{SS}^{-1/2} \tilde{\boldsymbol{\theta}}) \right| \geq t \right\} \\ \leq 2 \exp \left\{ - \frac{nt^2}{2 \sum_{i=1}^n (\mathbf{X}_{iS}^T \widehat{\boldsymbol{\Sigma}}_{SS}^{-1/2} \tilde{\boldsymbol{\theta}})^2} \right\} = 2 \exp \left(- \frac{t^2}{2\|\tilde{\boldsymbol{\theta}}\|_2^2} \right). \end{aligned}$$

In other words, for every $\tilde{\boldsymbol{\theta}} \in \mathbb{B}_2^s(R)$ and $\delta > 0$,

$$\left| Q(\widehat{\Sigma}_{SS}^{-1/2} \tilde{\boldsymbol{\theta}}) - Q_{\mathbf{X}}(\widehat{\Sigma}_{SS}^{-1/2} \tilde{\boldsymbol{\theta}}) \right| \leq (2\delta)^{1/2} \|\tilde{\boldsymbol{\theta}}\|_2 \leq (2\delta)^{1/2} R$$

holds with probability at least $1 - 2e^{-\delta}$. This, together with the union bound yields

$$\mathbb{P}_{\mathbf{X}} \left\{ \max_{\tilde{\boldsymbol{\theta}} \in \mathcal{V}} \left| Q(\widehat{\Sigma}_{SS}^{-1/2} \tilde{\boldsymbol{\theta}}) - Q_{\mathbf{X}}(\widehat{\Sigma}_{SS}^{-1/2} \tilde{\boldsymbol{\theta}}) \right| \geq (2\delta)^{1/2} R \right\} \leq \exp \left\{ s \log \left(1 + \frac{2R}{\epsilon} \right) - \delta \right\}. \quad (66)$$

In particular, by taking $\epsilon = Rn^{-1}$ in (65) and $\delta = s \log(1 + \frac{2R}{\epsilon}) + t \leq 2s \log n + t$ in (66) we conclude that

$$\mathbb{P}_{\mathbf{X}} \left\{ \Delta(R) \geq R(2t)^{1/2} + 2R(s \log n)^{1/2} + 2Rn^{-1/2} \right\} \leq 2e^{-t} \quad (67)$$

holds almost surely on the event $\mathcal{E}_0(\tau)$ for any $\tau > 0$.

In particular, by taking $t = cnR^2$ in (67) for some $c > 0$ to be specified below (72) and the union bound, we have

$$\begin{aligned} & \mathbb{P}_{\mathbf{X}} \left[\exists \boldsymbol{\theta} \in \mathcal{G}(\delta_1), \text{ s.t. } |Q(\boldsymbol{\theta}) - Q_{\mathbf{X}}(\boldsymbol{\theta})| \geq 2^{3/2} \|\tilde{\boldsymbol{\theta}}\|_2 \{ \|\tilde{\boldsymbol{\theta}}\|_2 (cn)^{1/2} + (s \log n)^{1/2} + n^{-1/2} \} \right] \\ & \leq \sum_{\ell=1}^{\infty} \mathbb{P}_{\mathbf{X}} \left[\exists \boldsymbol{\theta} \in \mathcal{G}_{\ell}(\delta_1), \text{ s.t. } |Q(\boldsymbol{\theta}) - Q_{\mathbf{X}}(\boldsymbol{\theta})| \geq (\alpha^{\ell} \delta_1)^2 (2cn)^{1/2} + 2\alpha^{\ell} \delta_1 \{ (s \log n)^{1/2} + n^{-1/2} \} \right] \\ & \leq \sum_{\ell=1}^{\infty} \mathbb{P}_{\mathbf{X}} \left[\Delta(\alpha^{\ell} \delta_1) \geq (\alpha^{\ell} \delta_1)^2 (2cn)^{1/2} + 2\alpha^{\ell} \delta_1 \{ (s \log n)^{1/2} + n^{-1/2} \} \right] \\ & \leq 2 \sum_{\ell=1}^{\infty} \exp \{ -cn(\alpha^{\ell} \delta_1)^2 \} \leq 2 \sum_{\ell=1}^{\infty} \exp \{ -2c\ell \log(\alpha) n \delta_1^2 \} \leq \frac{2 \exp(-c_0 n \delta_1^2)}{1 - \exp(-c_0 n \delta_1^2)}, \end{aligned}$$

where $c_0 = c \log 2$. This implies that with probability at least $1 - 4 \exp(-c_0 n \delta_1^2)$,

$$|Q(\boldsymbol{\theta}) - Q_{\mathbf{X}}(\boldsymbol{\theta})| \leq 2^{3/2} \sqrt{c} \|\widehat{\Sigma}_{SS}^{-1/2} \boldsymbol{\theta}\|_2 + 2^{3/2} \|\widehat{\Sigma}_{SS}^{-1/2} \boldsymbol{\theta}\|_2 \{ (s \log n)^{1/2} + n^{-1/2} \} \quad (68)$$

holds for all $\boldsymbol{\theta} \in \mathcal{G}(\delta_1)$ whenever $n \geq c^{-1} \delta_1^{-2}$.

For the (conditional) expectation

$$Q_{\mathbf{X}}(\boldsymbol{\theta}) = n^{-1/2} \sum_{i=1}^n \mathbb{E}_{\mathbf{X}} (\mathbf{X}_{iS}^{\top} \boldsymbol{\theta} - \epsilon_i - |\epsilon_i|) = n^{-1/2} (\mathbb{E}_{\mathbf{X}} \|\mathbf{X}_S \boldsymbol{\theta} - \boldsymbol{\epsilon}\|_1 - \mathbb{E} \|\boldsymbol{\epsilon}\|_1),$$

applying Lemmas 5 and 6 in Wang (2013) with slight modifications gives

$$Q_{\mathbf{X}}(\boldsymbol{\theta}) \geq \begin{cases} \frac{1}{4\sqrt{n}} \|\mathbf{X}_S \boldsymbol{\theta}\|_1 = \frac{\sqrt{n}}{4} \|n^{-1} \mathbf{X}_S \boldsymbol{\theta}\|_1 & \text{if } \|\mathbf{X}_S \boldsymbol{\theta}\|_1 \geq \frac{2n}{a_2}, \\ \frac{a_2}{8\sqrt{n}} \|\mathbf{X}_S \boldsymbol{\theta}\|_2^2 = \frac{a_2 \sqrt{n}}{8} \|\widehat{\Sigma}_{SS}^{-1/2} \boldsymbol{\theta}\|_2^2 & \text{if } \|\mathbf{X}_S \boldsymbol{\theta}\|_1 < \frac{2n}{a_2}, \end{cases} \quad (69)$$

where a_2 is as in Condition 3.4. For the sequence of LAD estimators $\{\widehat{\boldsymbol{\theta}}_S : S \subseteq [p], |S| = s\}$, from (61) it can be seen that $\|\mathbf{X}_S \widehat{\boldsymbol{\theta}}_S\|_1 \leq \|\mathbf{X}_S \widehat{\boldsymbol{\theta}}_S - \boldsymbol{\epsilon}\|_1 + \|\boldsymbol{\epsilon}\|_1 \leq 2\|\boldsymbol{\epsilon}\|_1$, and hence

$$\max_{S \subseteq [p]: |S|=s} \|n^{-1} \mathbf{X}_S \widehat{\boldsymbol{\theta}}_S\|_1 \leq 2 \left\{ \mathbb{E} \|\boldsymbol{\epsilon}\|_1 + n^{-1} \sum_{i=1}^n (\|\epsilon_i\|_1 - \mathbb{E} \|\epsilon_i\|_1) \right\}.$$

For every $t > 0$ and $1 < \kappa \leq 2$, by Markov's inequality we have

$$\mathbb{P} \left\{ \sum_{i=1}^n (\|\epsilon_i\|_1 - \mathbb{E} \|\epsilon_i\|_1) \geq t \right\} \leq t^{-\kappa} \mathbb{E} \left| \sum_{i=1}^n (\|\epsilon_i\|_1 - \mathbb{E} \|\epsilon_i\|_1) \right|^{\kappa} \leq 4^{2-\kappa} t^{-\kappa} n \mathbb{E} \|\epsilon\|^{\kappa},$$

where we used the inequality $|1 + x|^{\kappa} \leq 1 + \kappa x + 2^{2-\kappa} |x|^{\kappa}$ for $1 < \kappa \leq 2$ and $x \in \mathbb{R}$. The last two displays together imply that, with probability at least $1 - \delta_2$,

$$\max_{S \subseteq [p]: |S|=s} \|n^{-1} \mathbf{X}_S \widehat{\boldsymbol{\theta}}_S\|_1 \leq 2\mathbb{E} \|\epsilon\|_1 \left\{ 1 + 4^{(2-\kappa)/\kappa} (\mathbb{E} \|\epsilon\|_1)^{-1/\kappa} \delta_2^{-1/\kappa} n^{-1+1/\kappa} \right\}.$$

By Condition 3.4, we have $a_2 \mathbb{E} \|\epsilon\|_1 < 1$. Therefore, as long as the sample size n satisfies

$$n \geq \left\{ \frac{4^{2-q} a_2^q \mathbb{E} \|\epsilon\|^{\kappa}}{(1 - a_2 \mathbb{E} \|\epsilon\|_1)^{\kappa}} \right\}^{1/(\kappa-1)} \delta_2^{-1/(\kappa-1)}, \quad (70)$$

the event

$$\mathcal{E}_1 := \left\{ \max_{S \subseteq [p]: |S|=s} \|n^{-1} \mathbf{X}_S \widehat{\boldsymbol{\theta}}_S\|_1 \leq 2a_2^{-1} \right\} \quad (71)$$

occurs with probability at least $1 - \delta_2$.

Now, by (61), we have $Q(\widehat{\boldsymbol{\theta}}_S) \leq 0$ and thus $-Q(\widehat{\boldsymbol{\theta}}_S) - Q_{\mathbf{X}}(\widehat{\boldsymbol{\theta}}_S) \geq Q_{\mathbf{X}}(\widehat{\boldsymbol{\theta}}_S)$ holds for every s -subset $S \subseteq [p]$. Together with (68)–(71) and the union bound, this implies that on the event $\mathcal{E}_0(\tau) \cap \mathcal{E}_1$ for any $\tau > 0$,

$$\max_{S \subseteq [p]: |S|=s} \|\widehat{\Sigma}_{SS}^{-1/2} \widehat{\boldsymbol{\theta}}_S\|_2 \leq \min \left[\delta_1, 32\sqrt{2} a_2^{-1} \left\{ \left(\frac{s \log n}{n} \right)^{1/2} + \frac{1}{n} \right\} \right] \quad (72)$$

holds with (conditional) probability $1 - 4 \binom{p}{s} \exp(-c_0 n \delta_1^2) - \delta_2$, provided that the sample size n satisfies $n \geq 2 \cdot 32^2 (a_2 \delta_1)^{-2}$ and (70).

Finally, taking

$$\delta_1 = \frac{32}{a_2} \sqrt{\frac{2}{\log(2)}} \left(\frac{s \log \frac{ep}{s} + \log n}{n} \right)^{1/2} \quad \text{and} \quad \delta_2 = \frac{4^{2-q} a_2^q \mathbb{E} \|\epsilon\|^{\kappa}}{(1 - a_2 \mathbb{E} \|\epsilon\|_1)^{\kappa} n^{\kappa-1}}$$

in (72) proves (44). \blacksquare

A.6 Proof of Lemma 14

We prove this lemma by employing the arguments similar to those used in Spokoiny (2013), where the likelihood function $\mathcal{L}(\boldsymbol{\theta})$ is assumed to be twice differentiable with respect to $\boldsymbol{\theta}$. It is worth noticing that both Conditions (L) and (ED₂) in Spokoiny (2013) are not satisfied in the current situation. We provide here a self-contained proof in which Lemma 13 also plays an important role.

Step 1: Local linear approximation of $\nabla \mathcal{L}_n^S(\boldsymbol{\theta})$. Let $\chi_1^S(\boldsymbol{\theta})$ be the normalized residual of the local linear approximation of $\nabla \mathcal{L}_n^S(\boldsymbol{\theta})$ given by

$$\begin{aligned} \chi_1^S(\boldsymbol{\theta}) &= \mathbf{D}_0^{-1} \{ \nabla \mathcal{L}_n^S(\boldsymbol{\theta}) - \nabla \mathcal{L}_n^S(\mathbf{0}) + \mathbf{D}_0^{\top} \boldsymbol{\theta} \} \\ &= \mathbf{D}_0^{-1} \{ \mathbf{U}(\boldsymbol{\theta}) + \nabla \mathbb{E}_{\mathbf{X}} \mathcal{L}_n^S(\boldsymbol{\theta}) - \nabla \mathbb{E}_{\mathbf{X}} \mathcal{L}_n^S(\mathbf{0}) + \mathbf{D}_0^{\top} \boldsymbol{\theta} \}, \end{aligned} \quad (73)$$

where $\mathbf{U}(\boldsymbol{\theta}) = \nabla \zeta^S(\boldsymbol{\theta}) - \nabla \zeta^S(\mathbf{0})$ and $\mathbf{D}_0^2 = -\nabla^2 \mathbb{E} \mathbf{X} \{ \mathcal{L}_n^S(\mathbf{0}) \} = 2f_\varepsilon(0) \sum_{i=1}^n \mathbf{X}_{iS} \mathbf{X}_{iS}^T$. Then it follows from the mean value theorem that

$$\mathbb{E} \mathbf{X} \{ \chi_1^S(\boldsymbol{\theta}) \} = \{ \mathbf{I}_s - \mathbf{D}_0^{-1} \mathbf{D}^2(\tilde{\boldsymbol{\theta}}) \mathbf{D}_0^{-1} \} \mathbf{D}_0 \boldsymbol{\theta}, \quad (74)$$

where $\mathbf{D}^2(\boldsymbol{\theta}) = -\nabla^2 \mathbb{E} \mathbf{X} \{ \mathcal{L}_n^S(\boldsymbol{\theta}) \} = 2 \sum_{i=1}^n f_\varepsilon(\mathbf{X}_{iS}^T \boldsymbol{\theta}) \mathbf{X}_{iS} \mathbf{X}_{iS}^T$ and $\tilde{\boldsymbol{\theta}} = \lambda \boldsymbol{\theta}$ for some $0 \leq \lambda \leq 1$. As before, for every $r \geq 0$, define the local elliptic neighborhood of $\mathbf{0}$ as

$$\Theta_0(r) = \{ \boldsymbol{\theta} \in \mathbb{R}^s : \|\mathbf{D}_0 \boldsymbol{\theta}\|_2 \leq r \}.$$

On the event $\mathcal{E}_0(r)$ for some $r > 0$,

$$\|\mathbf{X}_{iS}^T \boldsymbol{\theta}\| \leq \|\mathbf{D}_0 \boldsymbol{\theta}\|_2 \|\mathbf{D}_0^{-1} \mathbf{X}_{iS}\|_2 \leq \{2n f_\varepsilon(0)\}^{-1/2} r^{1/2} r^{1/2} \quad (75)$$

for all $\boldsymbol{\theta} \in \Theta_0(r)$. Thus it follows from the Taylor expansion that for $r \leq \{2n f_\varepsilon(0)/\tau\}^{1/2}$,

$$\begin{aligned} & \|\mathbf{I}_s - \mathbf{D}_0^{-1} \mathbf{D}^2(\tilde{\boldsymbol{\theta}}) \mathbf{D}_0^{-1}\| \\ &= 2 \left\| \mathbf{D}_0^{-1} \sum_{i=1}^n \{ f_\varepsilon(\mathbf{X}_{iS}^T \tilde{\boldsymbol{\theta}}) - f_\varepsilon(0) \} \mathbf{X}_{iS} \mathbf{X}_{iS}^T \mathbf{D}_0^{-1} \right\| \leq \frac{A_3}{\sqrt{2} f_\varepsilon^3(0)} \frac{\tau^{1/2} r}{n^{1/2}} := \delta(\tau, r). \end{aligned} \quad (76)$$

Together, (74) and (76) imply that under the same constraint for (76),

$$\|\mathbb{E} \mathbf{X} \{ \chi_1^S(\boldsymbol{\theta}) \}\|_2 \leq \delta(\tau, r)r. \quad (77)$$

Turning to the stochastic component $\mathbf{D}_0^{-1} \mathbf{U}(\boldsymbol{\theta}) = \chi_1^S(\boldsymbol{\theta}) - \mathbb{E} \mathbf{X} \{ \chi_1^S(\boldsymbol{\theta}) \}$, we aim to bound $\max_{\boldsymbol{\theta} \in \Theta_0(r)} \|\mathbf{D}_0^{-1} \mathbf{U}(\boldsymbol{\theta})\|_2$, which can be written as

$$\max_{\boldsymbol{\theta} \in \Theta_0(r)} \max_{\|\mathbf{u}\|_2 \leq 1} \mathbf{u}^T \mathbf{D}_0^{-1} \mathbf{U}(\boldsymbol{\theta}) = r^{-1} \max_{\mathbf{u}, \boldsymbol{\theta} \in \Theta_0(r)} \mathbf{v}^T \mathbf{U}(\boldsymbol{\theta}). \quad (78)$$

Note that $\{\mathbf{v}^T \mathbf{U}(\boldsymbol{\theta}) : \mathbf{v}, \boldsymbol{\theta} \in \mathbb{R}^{2s}\}$ is a bivariate process indexed by $(\mathbf{v}^T, \boldsymbol{\theta}^T)^T \in \mathbb{R}^{2s}$. Define

$$\begin{aligned} \bar{\boldsymbol{\theta}} &= (\mathbf{v}^T, \boldsymbol{\theta}^T)^T \in \mathbb{R}^{2s}, \quad \bar{\mathbf{D}}_0 = \begin{pmatrix} \mathbf{D}_0 & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_0 \end{pmatrix} \in \mathbb{R}^{(2s) \times (2s)}, \\ \bar{\mathbf{U}}(\bar{\boldsymbol{\theta}}) &= \mathbf{v}^T \mathbf{U}(\boldsymbol{\theta}), \quad \bar{\Theta}_0(r) = \{ \bar{\boldsymbol{\theta}} \in \mathbb{R}^{2s} : \|\bar{\mathbf{D}}_0 \bar{\boldsymbol{\theta}}\|_2 \leq r \}. \end{aligned}$$

In this notation, from (78) and the identity $\bar{\mathbf{D}}_0 \bar{\boldsymbol{\theta}} = \mathbf{D}_0 \mathbf{v} + \mathbf{D}_0 \boldsymbol{\theta}$, it is easy to see that

$$\max_{\boldsymbol{\theta} \in \Theta_0(r)} \|\mathbf{D}_0^{-1} \mathbf{U}(\boldsymbol{\theta})\|_2 \leq r^{-1} \max_{\bar{\boldsymbol{\theta}} \in \bar{\Theta}_0(2r)} \bar{\mathbf{U}}(\bar{\boldsymbol{\theta}}). \quad (79)$$

Recall that $\nabla \zeta^S(\boldsymbol{\theta}) - \nabla \zeta^S(\mathbf{0}) = -2 \sum_{i=1}^n \{ I(Y_i \leq \mathbf{X}_{iS}^T \boldsymbol{\theta}) - I(Y_i \leq 0) + 1/2 - F_\varepsilon(\mathbf{X}_{iS}^T \boldsymbol{\theta}) \} \mathbf{X}_{iS}$, where for $i = 1, \dots, n$, $I(Y_i \leq \mathbf{X}_{iS}^T \boldsymbol{\theta}) - I(Y_i \leq 0) + 1/2 - F_\varepsilon(\mathbf{X}_{iS}^T \boldsymbol{\theta})$ is equal to

$$\begin{cases} I(0 < Y_i \leq \mathbf{X}_{iS}^T \boldsymbol{\theta}) - \mathbb{P} \mathbf{X}(0 < Y_i \leq \mathbf{X}_{iS}^T \boldsymbol{\theta}) & \text{if } \mathbf{X}_{iS}^T \boldsymbol{\theta} \geq 0, \\ -I(\mathbf{X}_{iS}^T \boldsymbol{\theta} < Y_i \leq 0) + \mathbb{P} \mathbf{X}(\mathbf{X}_{iS}^T \boldsymbol{\theta} < Y_i \leq 0) & \text{if } \mathbf{X}_{iS}^T \boldsymbol{\theta} < 0. \end{cases}$$

For $\boldsymbol{\theta} \in \mathbb{R}^s$, define random variables $\varepsilon_{i\boldsymbol{\theta}} = I(0 < Y_i \leq \mathbf{X}_{iS}^T \boldsymbol{\theta}) - I(\mathbf{X}_{iS}^T \boldsymbol{\theta} < Y_i \leq 0)$ satisfying

(i) conditional on $\mathbf{X}_{iS}^T \boldsymbol{\theta} \geq 0$, $\varepsilon_{i\boldsymbol{\theta}} = 1$ with probability $P_{i\boldsymbol{\theta}} - 1/2$ and $\varepsilon_{i\boldsymbol{\theta}} = 0$ with probability $3/2 - P_{i\boldsymbol{\theta}}$;

(ii) conditional on $\mathbf{X}_{iS}^T \boldsymbol{\theta} < 0$, $\varepsilon_{i\boldsymbol{\theta}} = -1$ with probability $1/2 - P_{i\boldsymbol{\theta}}$ and $\varepsilon_{i\boldsymbol{\theta}} = 0$ with probability $1/2 + P_{i\boldsymbol{\theta}}$,

where $P_{i\boldsymbol{\theta}} = F_\varepsilon(\mathbf{X}_{iS}^T \boldsymbol{\theta})$. In this notation, $\nabla \zeta^S(\boldsymbol{\theta}) - \nabla \zeta^S(\mathbf{0}) = -2 \sum_{i=1}^n (\text{Id} - \mathbb{E} \mathbf{X}) \varepsilon_{i\boldsymbol{\theta}} \mathbf{X}_{iS}$. For every $\lambda \in \mathbb{R}$ and $\mathbf{u} \in \mathbb{R}^s$, we have

$$\begin{aligned} & \mathbb{E} \mathbf{X} \exp \{ \lambda \mathbf{u}^T \{ \nabla \zeta^S(\boldsymbol{\theta}) - \nabla \zeta^S(\mathbf{0}) \} \} \\ &= \prod_{i=1}^n \left[\mathbb{E} \mathbf{X} \{ e^{-2\lambda \mathbf{u}^T \mathbf{X}_{iS} (I - \mathbb{E} \mathbf{X}) \varepsilon_{i\boldsymbol{\theta}}} \} I(\mathbf{X}_{iS}^T \boldsymbol{\theta} \geq 0) + \mathbb{E} \mathbf{X} \{ e^{-2\lambda \mathbf{u}^T \mathbf{X}_{iS} (I - \mathbb{E} \mathbf{X}) \varepsilon_{i\boldsymbol{\theta}}} \} I(\mathbf{X}_{iS}^T \boldsymbol{\theta} < 0) \right] \\ &= \prod_{i=1}^n \left\{ e^{-2\lambda \mathbf{u}^T \mathbf{X}_{iS} (3/2 - P_{i\boldsymbol{\theta}})} (P_{i\boldsymbol{\theta}} - 1/2) + e^{2\lambda \mathbf{u}^T \mathbf{X}_{iS} (P_{i\boldsymbol{\theta}} - 1/2)} (3/2 - P_{i\boldsymbol{\theta}}) \right\} I(\mathbf{X}_{iS}^T \boldsymbol{\theta} \geq 0) \\ &\quad + \left\{ e^{2\lambda \mathbf{u}^T \mathbf{X}_{iS} (1/2 + P_{i\boldsymbol{\theta}})} (1/2 - P_{i\boldsymbol{\theta}}) + e^{2\lambda \mathbf{u}^T \mathbf{X}_{iS} (P_{i\boldsymbol{\theta}} - 1/2)} (1/2 + P_{i\boldsymbol{\theta}}) \right\} I(\mathbf{X}_{iS}^T \boldsymbol{\theta} < 0). \end{aligned}$$

Further, using the inequalities $|e^u - 1 - u| \leq \frac{1}{2} u^2 e^{|u|}$ and $1 + u \leq e^u$ which hold for all $u \in \mathbb{R}$, the last term above can be bounded by

$$\begin{aligned} & \prod_{i=1}^n \left[\left\{ 1 + 2\lambda^2 (\mathbf{u}^T \mathbf{X}_{iS})^2 (P_{i\boldsymbol{\theta}} - 1/2) (3/2 - P_{i\boldsymbol{\theta}}) e^{2\lambda |\mathbf{u}^T \mathbf{X}_{iS}|} \right\} I(\mathbf{X}_{iS}^T \boldsymbol{\theta} \geq 0) \right. \\ & \quad \left. + \left\{ 1 + 2\lambda^2 (\mathbf{u}^T \mathbf{X}_{iS})^2 (1/2 - P_{i\boldsymbol{\theta}}) (1/2 + P_{i\boldsymbol{\theta}}) e^{2\lambda |\mathbf{u}^T \mathbf{X}_{iS}|} \right\} I(\mathbf{X}_{iS}^T \boldsymbol{\theta} < 0) \right] \\ & \leq \prod_{i=1}^n \left\{ 1 + 2\lambda^2 (\mathbf{u}^T \mathbf{X}_{iS})^2 |P_{i\boldsymbol{\theta}} - 1/2| e^{2\lambda |\mathbf{u}^T \mathbf{X}_{iS}|} \right\} \\ & \leq \prod_{i=1}^n \exp \{ 2\lambda^2 (\mathbf{u}^T \mathbf{X}_{iS})^2 |P_{i\boldsymbol{\theta}} - 1/2| e^{2\lambda |\mathbf{u}^T \mathbf{X}_{iS}|} \}. \end{aligned}$$

Consequently, for every $\bar{\boldsymbol{\theta}} = (\mathbf{v}^T, \boldsymbol{\theta}^T)^T \in \bar{\Theta}_0(2r)$,

$$\begin{aligned} & \log \mathbb{E} \mathbf{X} \exp \left\{ \lambda \frac{\bar{\mathbf{U}}(\bar{\boldsymbol{\theta}}) - \bar{\mathbf{U}}(\mathbf{0})}{\|\mathbf{D}_0 \bar{\boldsymbol{\theta}}\|_2} \right\} = \log \mathbb{E} \mathbf{X} \exp \left\{ \lambda \frac{\mathbf{v}^T \{ \zeta^S(\boldsymbol{\theta}) - \zeta^S(\mathbf{0}) \}}{\|\mathbf{D}_0 \bar{\boldsymbol{\theta}}\|_2} \right\} \\ & \leq \frac{2\lambda^2}{\|\mathbf{D}_0 \mathbf{v}\|_2^2 + \|\mathbf{D}_0 \boldsymbol{\theta}\|_2^2} \sum_{i=1}^n (\mathbf{v}^T \mathbf{X}_{iS})^2 |P_{i\boldsymbol{\theta}} - 1/2| \exp \left(\frac{2\lambda |\mathbf{v}^T \mathbf{X}_{iS}|}{\|\mathbf{D}_0 \bar{\boldsymbol{\theta}}\|_2} \right). \end{aligned} \quad (80)$$

On the event $\mathcal{E}_0(\tau)$ for some $\tau > 0$, we have $|P_{i\boldsymbol{\theta}} - 1/2| \leq 2A_2 \{2n f_\varepsilon(0)\}^{-1/2} \tau^{1/2} r$ and $|\mathbf{v}^T \mathbf{X}_{iS}| \leq \|\mathbf{D}_0 \mathbf{v}\|_2 \|\mathbf{D}_0^{-1} \mathbf{X}_{iS}\|_2 \leq \|\mathbf{D}_0 \mathbf{v}\|_2 \{2n f_\varepsilon(0)\}^{-1/2} \tau^{1/2}$. Together with (80), this yields that for all $|\lambda| \leq \{2n f_\varepsilon(0)/\tau\}^{1/2}$,

$$\log \mathbb{E} \mathbf{X} \exp \left\{ \lambda \frac{\bar{\mathbf{U}}(\bar{\boldsymbol{\theta}}) - \bar{\mathbf{U}}(\mathbf{0})}{\|\mathbf{D}_0 \bar{\boldsymbol{\theta}}\|_2} \right\} \leq \frac{\lambda^2 4e^2 A_2 r}{2 f_\varepsilon(0)} \sqrt{\frac{\tau}{2n f_\varepsilon(0)}}. \quad (81)$$

In view of (81), define

$$w_0(\tau) = 2e\sqrt{\frac{A_2 r_0}{f_\varepsilon(0)} \left\{ \frac{\tau}{2nf_\varepsilon(0)} \right\}^{1/4}} \quad (82)$$

for some $r_0 > 0$ to be specified (see (88) below), such that for any $\hat{\boldsymbol{\theta}} = (\boldsymbol{v}^\top, \boldsymbol{\theta}^\top)^\top \in \widehat{\Theta}_0(2r)$ with $0 \leq r \leq r_0$,

$$\mathbb{E} \mathbf{X} \exp \left\{ \frac{\lambda}{w_0(\tau)} \frac{\bar{\mathbf{U}}(\hat{\boldsymbol{\theta}}) - \bar{\mathbf{U}}(\mathbf{0})}{\|\mathbf{D}_0 \hat{\boldsymbol{\theta}}\|_2} \right\} \leq \exp(\lambda^2/2) \quad (83)$$

holds almost surely on $\mathcal{E}_0(\tau)$ for all

$$|\lambda| \leq 2e\sqrt{\frac{A_2 r_0}{f_\varepsilon(0)} \left\{ \frac{2nf_\varepsilon(0)}{\tau} \right\}^{1/4}} := g_0(\tau). \quad (84)$$

By (83), it follows from Corollary 2.2 in the supplement of Spokoiny (2012) and (79) that, for any $\tau > 0$, $0 \leq r \leq r_0$ and $0 < t \leq \frac{1}{2}g_0^2(\tau) - 2s$,

$$\mathbb{P} \mathbf{X} \left\{ \max_{\boldsymbol{\theta} \in \Theta_0(\tau)} \|\mathbf{D}_0^{-1} \mathbf{U}(\boldsymbol{\theta})\|_2 \geq 6w_0(\tau)(2t + 4s)^{1/2} \right\} \leq e^{-t} \quad (85)$$

holds almost surely on $\mathcal{E}_0(\tau)$, where g_0 is given at (84).

Combining (74) and (85) we obtain that for any $\tau > 0$, $0 \leq r \leq r_0 \leq \{2nf_\varepsilon(0)/\tau\}^{1/2}$ and $0 < t \leq \frac{1}{2}g_0^2(\tau) - 2s$,

$$\mathbb{P} \mathbf{X} \left\{ \max_{\boldsymbol{\theta} \in \Theta_0(\tau)} \|\chi_1^S(\boldsymbol{\theta})\|_2 \geq \delta(\tau, r)r + 6w_0(\tau)(2t + 4s)^{1/2} \right\} \leq e^{-t} \quad (86)$$

almost surely on $\mathcal{E}_0(\tau)$. For a given triplet (τ, r, t) , define the event

$$\Omega_0^S(\tau, r, t) = \left\{ \max_{\boldsymbol{\theta} \in \Theta_0(\tau)} \|\chi_1^S(\boldsymbol{\theta})\|_2 \leq \delta(\tau, r)r + 6w_0(\tau)(2t + 4s)^{1/2} \right\}. \quad (87)$$

Step 2: Fisher approximation. By Lemma 13,

$$\begin{aligned} & \max_{s \leq |\hat{p}|: |\hat{S}|=s} \|\mathbf{D}_0 \hat{\boldsymbol{\theta}}_S\|_2 \\ &= \{2nf_\varepsilon(0)\}^{1/2} \max_{s \leq |\hat{p}|: |\hat{S}|=s} \|\widehat{\Sigma}_{SS}^{1/2} \hat{\boldsymbol{\theta}}_S\|_2 \leq C_1 \sigma_2^{-1} \{2f_\varepsilon(0)s \log(pn)\}^{1/2} := r_0 \end{aligned} \quad (88)$$

holds with probability at least $1 - c_1 n^{-1} - c_2 n^{1-\kappa}$. Moreover, since $\hat{\boldsymbol{\theta}}_S$ maximizes $\mathcal{L}_n^S(\boldsymbol{\theta})$ over $\boldsymbol{\theta} \in \mathbb{R}^s$ for each s -subset $S \subseteq [p]$, we have $\nabla \mathcal{L}_n^S(\hat{\boldsymbol{\theta}}_S) = \mathbf{0}$ and $\chi_1^S(\hat{\boldsymbol{\theta}}) = \mathbf{D}_0 \hat{\boldsymbol{\theta}}_S - \widehat{\boldsymbol{\xi}}_S$. Thus, together with (87) implies that on the event $\{\hat{\boldsymbol{\theta}}_S \in \Theta_0(r_0) \cap \Omega_0^S(\tau, r_0, t)\}$,

$$\|\mathbf{D}_0 \hat{\boldsymbol{\theta}}_S - \widehat{\boldsymbol{\xi}}_S\|_2 \leq \delta(\tau, r_0)r_0 + 6w_0(\tau)(2t + 4s)^{1/2} \quad (89)$$

whenever $n \geq \{2f_\varepsilon(0)\}^{-1}\tau r_0^2$.

Step 3: Wilks approximation. For $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \Theta_0(r)$, define

$$\chi_2^S(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \mathcal{L}_n^S(\boldsymbol{\theta}) - \mathcal{L}_n^S(\boldsymbol{\theta}_2) - (\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2)^\top \nabla \mathcal{L}_n^S(\boldsymbol{\theta}_2) + \frac{1}{2} \|\mathbf{D}_0(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2)\|_2^2. \quad (90)$$

Noting that $\nabla_{\boldsymbol{\theta}_1} \chi_2^S(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \nabla \mathcal{L}_n^S(\boldsymbol{\theta}_1) - \nabla \mathcal{L}_n^S(\boldsymbol{\theta}_2) + \mathbf{D}_0^2(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2) = \mathbf{D}_0 \{\chi_1^S(\boldsymbol{\theta}_1) - \chi_1^S(\boldsymbol{\theta}_2)\}$, we have

$$|\chi_2^S(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)| = |\chi_2^S(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) - \chi_2^S(\boldsymbol{\theta}_2, \boldsymbol{\theta}_2)| \leq 2 \|\mathbf{D}_0(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2)\|_2 \max_{\boldsymbol{u} \in \Theta_0(r)} \|\chi_1^S(\boldsymbol{u})\|_2, \quad (91)$$

where $\tilde{\boldsymbol{\theta}} = \lambda \boldsymbol{\theta}$ for some $0 \leq \lambda \leq 1$. Let $r_0 > 0$ be as in (88). Then, it follows from (91) that on $\Omega_0^S(\tau, r_0, t)$ with $n \geq \{2f_\varepsilon(0)\}^{-1}\tau r_0^2$,

$$\max_{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \Theta_0(r_0)} \frac{|\chi_2^S(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)|}{\|\mathbf{D}_0(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2)\|_2} \leq 2\delta(\tau, r_0)r_0 + 12w_0(\tau)(2t + 4s)^{1/2}.$$

In view of (90), $\mathcal{L}_n^S(\hat{\boldsymbol{\theta}}_S) - \mathcal{L}_n^S(\mathbf{0}) - \frac{1}{2} \|\mathbf{D}_0 \hat{\boldsymbol{\theta}}_S\|_2^2 = -\chi_2^S(\mathbf{0}, \hat{\boldsymbol{\theta}}_S)$. Therefore, on the event $\{\hat{\boldsymbol{\theta}}_S \in \Theta_0(r_0) \cap \Omega_0^S(\tau, r_0, t)\}$ we have

$$\begin{aligned} & \left| 2\{\mathcal{L}_n^S(\hat{\boldsymbol{\theta}}_S) - \mathcal{L}_n^S(\mathbf{0})\}^{1/2} - \|\mathbf{D}_0 \hat{\boldsymbol{\theta}}_S\|_2 \right| \\ & \leq \frac{2\{\mathcal{L}_n^S(\hat{\boldsymbol{\theta}}_S) - \mathcal{L}_n^S(\mathbf{0})\} - \|\mathbf{D}_0 \hat{\boldsymbol{\theta}}_S\|_2^2}{\|\mathbf{D}_0 \hat{\boldsymbol{\theta}}_S\|_2} \leq \frac{2|\chi_2^S(\mathbf{0}, \hat{\boldsymbol{\theta}}_S)|}{\|\mathbf{D}_0 \hat{\boldsymbol{\theta}}_S\|_2} \leq 4\{\delta(\tau, r_0)r_0 + 6w_0(\tau)(2t + 4s)^{1/2}\}, \end{aligned}$$

provided that $n \geq \{2f_\varepsilon(0)\}^{-1}\tau r_0^2$. Together with (89), this implies that conditional on the event $\cap_{s \leq |\hat{p}|: |\hat{S}|=s} \{\hat{\boldsymbol{\theta}}_S \in \Theta_0(r_0) \cap \Omega_0^S(\tau, r_0, t)\}$,

$$\max_{s \leq |\hat{p}|: |\hat{S}|=s} \left| 2\{\mathcal{L}_n^S(\hat{\boldsymbol{\theta}}_S) - \mathcal{L}_n^S(\mathbf{0})\}^{1/2} - \|\widehat{\boldsymbol{\xi}}_S\|_2 \right| \leq 5\{\delta(\tau, r_0)r_0 + 6w_0(\tau)(2t + 4s)^{1/2}\} \quad (92)$$

whenever $n \geq \{2f_\varepsilon(0)\}^{-1}\tau r_0^2$, where $\delta(\tau, r)$, r_0 and $w_0(\tau)$ are as in (76), (88) and (82).

Finally, taking $\tau = \tau_0 \asymp \lambda_{\min}^{-1}(s)s \log(pn)$ as in (36) and setting $t = s \log \frac{pn}{s} + \log n$ in the concentration bound (86) prove (48) using Boole's inequality. \blacksquare

Bayesian Graphical Models for Multivariate Functional Data

Hongxiao Zhu

*Department of Statistics
Virginia Tech, 250 Drillfield Drive (MC 0439)
Blacksburg, VA 24061, USA*

HONGXIAO@VT.EDU

Nate Strawn

*Department of Mathematics and Statistics
Georgetown University
Washington D.C. 20057, USA*

NATE.STRAWN@GEORGETOWN.EDU

David B. Dunson

*Department of Statistical Science
Duke University
Durham NC 27708, USA*

DUNSON@DUKE.EDU

Editor: Jie Peng

Abstract

Graphical models express conditional independence relationships among variables. Although methods for vector-valued data are well established, functional data graphical models remain underdeveloped. By functional data, we refer to data that are realizations of random functions varying over a continuum (e.g., images, signals). We introduce a notion of conditional independence between random functions, and construct a framework for Bayesian inference of undirected, decomposable graphs in the multivariate functional data context. This framework is based on extending Markov distributions and hyper Markov laws from random variables to random processes, providing a principled alternative to naive application of multivariate methods to discretized functional data. Markov properties facilitate the composition of likelihoods and priors according to the decomposition of a graph. Our focus is on Gaussian process graphical models using orthogonal basis expansions. We propose a hyper-inverse-Wishart-process prior for the covariance kernels of the infinite coefficient sequences of the basis expansion, and establish its existence and uniqueness. We also prove the strong hyper Markov property and the conjugacy of this prior under a finite rank condition of the prior kernel parameter. Stochastic search Markov chain Monte Carlo algorithms are developed for posterior inference, assessed through simulations, and applied to a study of brain activity and alcoholism.

Keywords: graphical model, functional data analysis, gaussian process, model uncertainty, stochastic search

1. Introduction

Graphical models provide a powerful tool for describing conditional independence structures between random variables. In the multivariate data case, Dawid and Lauritzen (1993) defined Markov distributions (distributions with Markov property over a graph) of random vectors which can be factorized according to the structure of a graph. They also introduced

hyper-Markov laws serving as prior distributions in Bayesian analysis. The special case of Gaussian graphical models, in which a multivariate Gaussian distribution is assumed and the graph structure corresponds to the zero pattern of the precision matrix (Dempster, 1972; Lauritzen, 1996), is well studied. Computational algorithms, such as Markov chain Monte Carlo (MCMC) and stochastic search, are developed to estimate the graph based on the conjugate hyper-inverse-Wishart prior and its extensions (Giudici and Green, 1999; Roverato, 2002; Jones et al., 2005; Scott and Carvalho, 2008; Carvalho and Scott, 2009).

In the frequentist literature, notable works on graphical models include the graphical LASSO (Yuan and Lin, 2007; Friedman et al., 2008; Mazumder and Hastie, 2012a,b) and the neighborhood selection approach (Meinshausen and Bühlmann, 2006; Ravikumar et al., 2010). The graphical LASSO induces sparse estimation of the precision matrix of the Gaussian likelihood through l_1 regularization. The neighborhood selection approach relies on estimating the neighborhood of each node separately by regressing each variable on all the remaining variables, sparsifying with l_1 regularization, and then stitching the neighborhoods together to form the global graph estimate. Various extensions, computational methods, and theoretical properties have been developed in these frameworks (Lam and Fan, 2009; Höfling and Tibshirani, 2009; Cai et al., 2011; Witten et al., 2011; Yang et al., 2012; Mazumder and Hastie, 2012a,b; Anandkumar et al., 2012; Loh and Wainwright, 2013).

The graphical modeling literature focuses primarily on vector-valued data with each node corresponding to one variable. Many applications, however, involve *functional data*—data that are realizations of random functions varying over a continuum such as a time interval or a spatial domain. Common types of functional data include signals, images, and many emerging high-throughput digital measurements. The dependence structure of functional data is of interest in a wide range of applications. For example, in neuroimaging, we are often interested in the dependence network across brain regions, where data from each region are of functional form (e.g. EEG/ERP signals, MRI/fMRI regions). In bioinformatics, we often need to model gene networks based on time-course gene expression data (Ma et al., 2006), treating each time-course as a continuous process. In epigenetics, it is of interest to study how cells are differentiated into organs (cell lineage and differentiation) by exploring the dependence structure of genome-wide methylation levels across different cell types, and for each cell type, the methylation level can be considered as a function of the genomic locations.

Although there is increasingly rich literature on generalizations to accommodate matrix-variate graphical models (Wang and West, 2009), time varying graphical models (Zhou et al., 2010; Kolar and Xing, 2011), and dynamic linear models (Carvalho and West, 2007), the generalization to functional data has not received much attention in the literature. In recent work, Qiao et al. (2015) extended the graphical LASSO of Yuan and Lin (2007) to the functional data case. They estimate the graph by maximizing a penalized log-Gaussian likelihood constructed through truncated basis expansion, and prove the consistency of the estimated edges. In this paper, we propose Bayesian graphical models for functional data following a fundamentally different approach. In particular, we construct the graphical model directly in the space of infinite dimensional random functions through establishing the Markov distributions and hyper Markov laws for random processes, and propose a Bayesian framework that generally holds for all random processes. We then demonstrate the special case of a multivariate Gaussian process in the space of square integrable func-

tions. Through representing the random functions with orthogonal basis expansions, we transform functional data from the function space to the isometrically isomorphic space of basis coefficients, where Markov distributions and hyper Markov laws can be conveniently constructed. We further propose a hyper-inverse-Wishart-process prior for the covariance kernels of the coefficient sequences, and study theoretical properties of the proposed prior such as existence and uniqueness. We also establish the strong hyper Markov property and conjugacy of this prior under a finite rank condition for the prior kernel parameter, which implies that the covariance kernel of the coefficient sequences is a priori finite dimensional. To perform posterior inference, we introduce a regularity condition which allows us to write the likelihood and prior density and design stochastic search MCMC algorithms for posterior sampling. Performance of the proposed approach is demonstrated through simulation studies and analysis of brain activity and alcoholism data.

To our knowledge, the proposed approach is the first considering functional data graphical models from a Bayesian perspective. It extends the theory of Dawid and Lauritzen (1993) from multivariate data to multivariate functional data. Most existing graphical model approaches often naively apply multivariate methods to functional data after performing discretization or feature extraction. Such approaches may not take full advantage of the fact that data arise from a function and can lack reasonable limiting behavior. Our graphical model framework guarantees proper theoretical behavior as well as computational convenience.

2. Graphical Models for Multivariate Functional Data

In this section, we first review graphical models for multivariate data in Section 2.1, then introduce graphical models for multivariate functional data in Section 2.2, and finally present the specific case of Gaussian process graphical models in Section 2.3.

2.1 Review of Graph Theory and Gaussian Graphical Models

We follow Dawid and Lauritzen (1993), Lauritzen (1996), and Jones et al. (2005). Let $G = (V, E)$ denote an undirected graph with a vertex set V and a set of edge pairs $E = \{(i, j)\}$. Each vertex corresponds to one variable. Two variables a and b are conditionally independent if and only if $(a, b) \notin E$. A graph or a subgraph is *complete* if all possible pairs of vertices are joined by edges. A complete subgraph is *maximal* if it is not contained within another complete subgraph. A maximal subgraph is called a *clique*. If A, B, C are subsets of V with $V = A \cup B$, $C = A \cap B$, then C is said to separate A from B if every path from a vertex in A to a vertex in B goes through C . C is called a *separator* and the pair (A, B) forms a decomposition of G . The separator is *minimal* if it does not contain a proper subgraph which also separates A from B . While keeping the separators minimal, we can iteratively decompose a graph into a sequence of *prime components* – a sequentially defined collection of subgraphs that cannot be further decomposed (Jones et al., 2005). If all the prime components of a connected graph are complete, the graph is called *decomposable*. All the prime components of a decomposable graph are cliques. Iteratively decomposing a decomposable graph G produces a *perfectly ordered* sequence of cliques and separators $(C_1, S_2, C_3, \dots, S_m, C_m)$ such that $S_i = H_{i-1} \cap C_i$ and $H_{i-1} = C_1 \cup \dots \cup C_{i-1}$. Let $\mathcal{C} = \{C_1, \dots, C_m\}$ denote the set of cliques and $\mathcal{S} = \{S_2, \dots, S_m\}$ denote the set of

separators. The perfect ordering means that for every $i = 2, \dots, m$, there is a $j < i$ with $S_i \subset C_j$ (Lauritzen, 1996, page 15).

If the components of a random vector $\mathbf{X} = (X_1, \dots, X_p)^T$ obey conditional independence according to a decomposable graph G , the joint density can be factorized as

$$p(\mathbf{X} | G) = \frac{\prod_{C \in \mathcal{C}} p(\mathbf{X}_C)}{\prod_{S \in \mathcal{S}} p(\mathbf{X}_S)},$$

where $\mathbf{X}_A = \{X_i, i \in A\}$. If \mathbf{X} is Gaussian with zero mean and precision matrix $\mathbf{\Omega} = \mathbf{\Sigma}^{-1}$, then X_i is conditionally independent of X_j given $\mathbf{X}_{V \setminus \{i, j\}}$, denoted by $X_i \perp\!\!\!\perp X_j | \mathbf{X}_{V \setminus \{i, j\}}$, if and only if the (i, j) th element of $\mathbf{\Omega}$ is zero. In this case $p(\mathbf{X} | G)$ is uniquely determined by marginal covariances $\{\mathbf{\Sigma}_C, \mathbf{\Sigma}_S, C \in \mathcal{C}, S \in \mathcal{S}\}$, which are sub-diagonal blocks of $\mathbf{\Sigma}$ according to the clique and separator sets. For a given G , a convenient conjugate prior for $\mathbf{\Sigma}$ is hyper-inverse-Wishart (HIW) with density

$$p(\mathbf{\Sigma} | G, \delta, \mathbf{U}) = \frac{\prod_{C \in \mathcal{C}} p(\mathbf{\Sigma}_C | \delta, \mathbf{U}_C)}{\prod_{S \in \mathcal{S}} p(\mathbf{\Sigma}_S | \delta, \mathbf{U}_S)},$$

where $p(\mathbf{\Sigma}_C | \delta, \mathbf{U}_C)$ and $p(\mathbf{\Sigma}_S | \delta, \mathbf{U}_S)$ are densities of inverse-Wishart (IW) distributions. In this paper, the inverse-Wishart follows the parameterization of Dawid (1981), i.e., $\mathbf{\Sigma} \sim \text{IW}(\delta, \mathbf{U})$ if and only if $\mathbf{\Sigma}^{-1}$ has a Wishart distribution $W(\delta + p - 1, \mathbf{U}^{-1})$, where $\delta > 0$ and $\mathbf{\Sigma}$ is a p by p matrix.

2.2 Graphical Models for Multivariate Functional Data

Let $\mathbf{f} = \{f_j\}_{j=1}^p$ denote a collection of random processes where each component f_j is in $L^2(T_j)$ and each T_j is a closed subset of the real line. The domain of \mathbf{f} is denoted by $T = \bigcup_{j=1}^p T_j$, where \bigcup denotes the disjoint union defined by $\bigcup_{j=1}^p T_j = \bigcup_{j=1}^p \{(t, j) : t \in T_j\}$. For each j , let $\{\phi_{j,k}\}_{k=1}^{\infty}$ denote an orthonormal basis of $L^2(T_j)$. The extended basis functions $\psi_{j,k} = (0, \dots, 0, \phi_{j,k}, 0, \dots, 0)$, with $\phi_{j,k}$ in the j th component and 0 functions elsewhere for $j = 1, \dots, p$ and $k = 1, \dots, \infty$, form an orthonormal basis of $L^2(T)$. Let $(L^2(T), \mathcal{B}(L^2(T)), P)$ be a probability space, where $\mathcal{B}(L^2(T))$ is the Borel σ -algebra on $L^2(T)$. For $V = \{1, 2, \dots, p\}$ and $A \subset V$, denote by \mathbf{f}_A the subset of \mathbf{f} with domain $T_A = \bigcup_{i \in A} T_i$. We define the conditional independence relationships for components of \mathbf{f} in Definition 1.

Definition 1 Let A, B , and C be subsets of V . Then \mathbf{f}_A is *conditionally independent of* \mathbf{f}_B given \mathbf{f}_C under P , written as $\mathbf{f}_A \perp\!\!\!\perp \mathbf{f}_B | \mathbf{f}_C [P]$, if for any $\mathbf{f}_A \in D_A$, where D_A is a measurable set in $L^2(T_A)$, there exists a version of the conditional probability $P(\mathbf{f}_A \in D_A | \mathbf{f}_B, \mathbf{f}_C) = P(\mathbf{f}_A \in D_A | \mathbf{f}_C)$ which is $\mathcal{B}(L^2(T_C))$ measurable, and hence one may write $P(\mathbf{f}_A \in D_A | \mathbf{f}_B, \mathbf{f}_C) = P(\mathbf{f}_A \in D_A | \mathbf{f}_C)$. Here, $\mathcal{B}(L^2(T_C))$ denotes the Borel σ -algebra on $L^2(T_C)$. Note that this implies $P(\mathbf{f}_A \in D_A, \mathbf{f}_B \in D_B | \mathbf{f}_C) = P(\mathbf{f}_A \in D_A | \mathbf{f}_C) P(\mathbf{f}_B \in D_B | \mathbf{f}_C)$.

We would like to use a decomposable graph $G = (V, E)$ to describe the conditional independence relationships of components in \mathbf{f} , whereby a Bayesian framework can be constructed and G can be inferred through posterior inference. To this end, we link the probability measure P of \mathbf{f} with G by assuming that P is Markov over G , as defined in Definition 2.

Definition 2 Let $G = (V, E)$ denote a decomposable graph. A probability measure P of \mathbf{f} is called Markov over G if for any decomposition (A, B) of G , $\mathbf{f}_A \perp\!\!\!\perp \mathbf{f}_B \mid \mathbf{f}_{A \cap B}[P]$.

Given a decomposable graph G , a probability measure of \mathbf{f} with Markov property may be constructed. To enable the construction, we first state Lemma 1, which generalizes Lemma 2.5 of Dawid and Lauritzen (1993) from the random variable to the random process case.

Lemma 1 Let $\mathbf{f} = (f_1, \dots, f_p)$ be a collection of random processes in $L^2(T)$. For subsets $A, B \subset V = \{1, \dots, p\}$ with $A \cap B \neq \emptyset$, suppose that P_1 and P_2 are probability measures of \mathbf{f}_A and \mathbf{f}_B , respectively. If P_1 and P_2 are consistent, meaning that they induce the same measure for $\mathbf{f}_{A \cap B}$, then there exists a unique probability measure P for $\mathbf{f}_{A \cup B}$ such that (i) $P_A = P_1$, (ii) $P_B = P_2$, and (iii) $\mathbf{f}_A \perp\!\!\!\perp \mathbf{f}_B \mid \mathbf{f}_{A \cap B}[P]$. The measure P is called a Markov combination of P_1 and P_2 , denoted as $P = P_1 \star P_2$.

We provide a proof of Lemma 1 through construction in Appendix B. The main idea is to first construct the conditional probability $P_1 \{ \cdot \mid \pi_{A \cap B}(\mathbf{f}_A) \}$ from P_1 , where $\pi_{A \cap B} : L^2(T_A) \rightarrow L^2(T_{A \cap B})$ is a projection map and $T_A = \bigsqcup_{j \in A} T_j$. We then define $P \{ \cdot \mid \pi_B(\mathbf{f}) \}$ based upon $P_1 \{ \cdot \mid \pi_{A \cap B}(\mathbf{f}_A) \}$ using disintegration theory (Chang and Pollard, 1997), and finally construct the joint measure P that satisfies conditions (i)–(iii). With Lemma 1, we can construct a joint probability measure for \mathbf{f} that is Markov over G . The construction is based on the perfectly ordered decomposition $(C_1, S_2, C_2, \dots, S_m, C_m)$ of G with $S_i = H_{i-1} \cap C_i$ and $H_{i-1} = C_1 \cup \dots \cup C_{i-1}$. Let $\{M_{C_i}, i = 1, \dots, m\}$ be a sequence of pairwise consistent probability measures for $\{\mathbf{f}_{C_i}, i = 1, \dots, m\}$. We construct a Markov probability measure P over G through the following recursive procedure

$$\begin{aligned} P_{C_1} &= M_{C_1}, & (1) \\ P_{H_{i+1}} &= P_{H_i} \star M_{C_{i+1}}, \quad i = 1, \dots, m-1. & (2) \end{aligned}$$

One can show that the probability measure constructed this way is the unique Markov probability measure over G with marginals $\{M_{C_i}\}$, and the proof follows that of Theorem 2.6 in Dawid and Lauritzen (1993). We call the probability distribution induced by the probability measure constructed above the *Markov distribution* of \mathbf{f} over G .

Denote the Markov distributions of \mathbf{f} constructed in (1)–(2) by P_G , and denote the space of all Markov distributions over G by $\mathcal{M}(G)$. A prior law for P_G is then supported on $\mathcal{M}(G)$. We follow Dawid and Lauritzen (1993) to define hyper Markov laws and use them as prior laws for P_G . A prior law \mathcal{L} of P_G is called *hyper Markov* over G if for any decomposition (A, B) of G , $(P_G)_A \perp\!\!\!\perp (P_G)_B \mid (P_G)_{A \cap B}[\mathcal{L}]$, where $(P_G)_A$ takes values in $\mathcal{M}(G_A)$ which is the space of all Markov distributions over subgraph G_A . Here, we have assumed that G is collapsible onto A , therefore $\phi \in \mathcal{M}(G_A)$ if and only if $\phi = (P_G)_A$ for some $(P_G) \in \mathcal{M}(G)$. The following Proposition 1 states that the theory of hyper Markov laws of Dawid and Lauritzen (1993) applies to our random process setup.

Proposition 1 The theory of hyper Markov laws over undirected decomposable graphs, as described in Section 3 of Dawid and Lauritzen (1993), holds for random processes.

According to the theory of hyper Markov laws, one can construct a prior law for P_G using a sequence of consistent marginal laws $\{\mathcal{L}_C, C \in \mathcal{C}\}$ in a similar fashion as (1)–(2).

Denote by \mathcal{L}_G the constructed hyper Markov prior for P_G and by Π a prior distribution for the graph G . A Bayesian graphical model for the collection of random processes \mathbf{f} can be described as

$$\mathbf{f} \sim P_G; \quad P_G \sim \mathcal{L}_G; \quad G \sim \Pi. \quad (3)$$

As we have yet to specify a concrete example for the probability measure P_G , the above Bayesian framework remains abstract at the moment. In Section 2.3, we construct P_G using Gaussian processes and propose a hyper-inverse-Wishart-process law as the prior for P_G . The prior distribution Π is supported on the finite dimensional space of decomposable graphs with p nodes.

2.3 Gaussian Process Graphical Models for Multivariate Functional Data

Let $\mathbf{f}_0 = (f_{01}, \dots, f_{0p})$ be an element in $L^2(T)$. Denote by $\mathcal{K} = \{k_{ij} : T_i \times T_j \rightarrow \mathbb{R}\}$ a collection of covariance kernels such that $\text{cov}\{f_i(s), f_j(t)\} = k_{ij}(s, t), s \in T_i, t \in T_j$. We assume that \mathcal{K} is positive semidefinite and trace class. Positive semidefinite means that

$$\sum_{i,j=1}^p \sum_{k,l=1}^p c_{ik} c_{jl} \int_{T_j} \int_{T_i} k_{ij}(s, t) \phi_{ik}(s) \phi_{jl}(t) ds dt \geq 0$$

for any square summable sequence $\{c_{ik}, i = 1, \dots, p, k = 1, \dots, \infty\}$; trace class means that

$$\sum_{j=1}^p \sum_{l=1}^p \int_{T_j} \int_{T_l} k_{jl}(s, t) \phi_{jl}(s) \phi_{jl}(t) ds dt < \infty.$$

Then \mathbf{f}_0 and \mathcal{K} uniquely determine a Gaussian process on $L^2(T)$ (Prato, 2006), which we call a multivariate Gaussian process, and write $\text{MGP}(\mathbf{f}_0, \mathcal{K})$. The definition of multivariate Gaussian process implies that for $A \subset V$, $\mathbf{f}_A \sim \text{MGP}(\mathbf{f}_{0,A}, \mathcal{K}_A)$ where $\mathcal{K}_A = \{k_{ij}, i, j \in A\}$. Furthermore, on a sequence of cliques $\mathcal{C} = \{C_1, \dots, C_m\}$, the marginal Gaussian process measures for $\{\mathbf{f}_C, C \in \mathcal{C}\}$ are automatically consistent because they are induced from the same joint distribution. Therefore, we can construct a Markov distribution for \mathbf{f} over G through procedure (1)–(2). We denote the resulting distribution of \mathbf{f} by $\text{MGP}_G(\mathbf{f}_0, \mathcal{K}_G)$, where $\mathcal{K}_G = \{k_{ij} : i, j \in C, C \in \mathcal{C}\}$. It is clear from this construction that the distribution MGP_G is Markov over G whereas MGP is not.

For the convenience of both theoretical analysis and computation, we represent elements in $L^2(T)$ using orthonormal basis expansions and construct a Bayesian graphical model in the dual space of basis coefficients. Let $\{\phi_{jk}\}_{k=1}^{\infty}$ denote an orthonormal basis of $L^2(T_j)$. For example, $\{\phi_{jk}\}_{k=1}^{\infty}$ could be a wavelet basis. We have the representation $f_j(t) = \sum_{k=1}^{\infty} c_{jk} \phi_{jk}(t)$ where $c_{jk} = \langle f_j, \phi_{jk} \rangle = \int_{T_j} f_j(t) \phi_{jk}(t) dt$. The coefficient sequence $c_j = \{c_{jk}, k = 1, \dots, \infty\}$ lies in the space of square-summable sequences, denoted by $\ell_j^2 = \{c_jk : \sum_{k=1}^{\infty} c_jk^2 < \infty\}$. Denote $\ell^2 = \prod_{j=1}^p \ell_j^2$. Since ℓ_j^2 and $L^2(T_j)$ are isometrically isomorphic for each j , once an orthonormal basis of $L^2(T)$ has been chosen, we have an identification between the Borel probability measures defined on ℓ^2 and $L^2(T)$; therefore we can construct statistical models on ℓ^2 without loss of generality. Let $\mathbf{c} = (c_1, \dots, c_p)$ denote

the coefficient sequence of \mathbf{f} . Then $\mathbf{f} \sim \text{MGP}(\mathbf{f}_0, \mathcal{K})$ corresponds to $\mathbf{c} \sim \text{dMGP}(\mathbf{c}_0, \mathcal{Q})$, where dMGP denotes the infinite dimensional discrete multivariate Gaussian processes, \mathbf{c}_0 is the coefficient sequence of \mathbf{f}_0 and $\mathcal{Q} = \{q_{ij}(\cdot, \cdot), i, j \in V\}$. Here, q_{ij} is the covariance kernel so that $\text{cov}(c_{k_h}, c_{l_j}) = q_{ij}(k, l)$ for $k, l \in \{1, 2, 3, \dots\}$. Similarly, $\mathbf{f} \sim \text{MGP}_G(\mathbf{f}_0, \mathcal{K}_G)$ corresponds to $\mathbf{c} \sim \text{dMGP}_G(\mathbf{c}_0, \mathcal{Q}_G)$ where $\mathcal{Q}_G = \{q_{ij}(\cdot, \cdot), i, j \in C, C \in \mathcal{C}\}$. The collection \mathcal{Q} is also positive semidefinite and trace class, so that $\sum_{i,j=1}^p \sum_{k,l=1}^{\infty} c_{k_h} c_{l_j} q_{ij}(k, l) \geq 0$ for any square summable sequence $\{c_{k_h}, i = 1, \dots, p, k = 1, \dots, \infty\}$, and $\sum_{j=1}^p \sum_{k=1}^{\infty} q_{ij}(k, k) < \infty$. Furthermore, \mathcal{K} relates to \mathcal{Q} through equation $k_{ij}(s, t) = \sum_{k_h=1}^p \sum_{l_j=1}^p q_{ij}(k_h, l_j) \phi_{k_h}(s) \phi_{l_j}(t)$. Denote by \mathbf{p}^C and \mathbf{p}^f the probability measures of \mathbf{c} and \mathbf{f} respectively, then $\mathbf{f}_A \perp \mathbf{f}_B \mid \mathbf{f}_C [\mathbf{p}^f]$ implies $\mathbf{c}_A \perp \mathbf{c}_B \mid \mathbf{c}_C [\mathbf{p}^c]$ and vice versa. Thus, the distribution $\text{dMGP}_G(\mathbf{c}_0, \mathcal{Q}_G)$ of \mathbf{c} is again Markov.

Assume that $\mathbf{c} \sim \text{dMGP}_G(\mathbf{c}_0, \mathcal{Q}_G)$. The parameters involved in this distribution include \mathbf{c}_0 and \mathcal{Q}_G . In this study, we assume that \mathbf{c}_0 is fixed (e.g., a zero sequence) so that the distribution of \mathbf{c} is uniquely determined by \mathcal{Q}_G . As indicated in Section 2.2, we would like to construct a hyper Markov law for the dMGP $_G$ distribution. Since dMGP $_G$ is uniquely determined by \mathcal{Q}_G , it is equivalent to construct a hyper Markov law for \mathcal{Q}_G . Given a positive integer δ and a collection $\mathcal{U} = \{u_{ij} : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{R}, i, j \in V\}$ which is symmetric, positive semidefinite, and trace class, we construct a hyper-inverse-Wishart-process (HIWP) prior for \mathcal{Q}_G following Theorem 1.

Theorem 1 Assume that $\mathbf{c} \sim \text{dMGP}_G(\mathbf{c}_0, \mathcal{Q}_G)$. Suppose that δ is a positive integer, and \mathcal{U} is a collection of kernels that is symmetric, positive semidefinite and trace class. Then there exists a sequence of pairwise consistent inverse-Wishart processes determined by δ and $\mathcal{U}_C = \{u_{ij}, i, j \in C\}$, $C \in \mathcal{C}$, based on which one can construct a unique hyper Markov law for \mathcal{Q}_G , which we call a hyper-inverse-Wishart-process, and write $\mathcal{Q}_G \sim \text{HIWP}_G(\delta, \mathcal{U}_C)$, where $\mathcal{U}_C = \{u_{ij}, i, j \in C, C \in \mathcal{C}\}$.

Based on Theorem 1, a Bayesian Gaussian process graphical model can be written as

$$\mathbf{c} \sim \text{dMGP}_G(\mathbf{c}_0, \mathcal{Q}_G), \quad \mathcal{Q}_G \sim \text{HIWP}_G(\delta, \mathcal{U}_C), \quad G \sim \Pi. \quad (4)$$

It is of interest to investigate the properties of the HIWP prior and the corresponding posterior distribution. As shown in Dawid and Lauritzen (1993), one nice property of the HIW law is the strong hyper Markov property, which leads to conjugacy as well as convenient posterior computation at each clique. In case of the HIWP prior, the strong hyper Markov property is defined such that for any decomposition (A, B) of G in model (4), $\mathcal{Q}_{B|A} \perp \mathcal{Q}_A$, where $\mathcal{Q}_{B|A}$ denotes the conditional distribution (i.e., conditional covariance) of \mathbf{c}_B given \mathbf{c}_A . In the following proposition, we show that the HIWP $_G$ prior constructed in Theorem 1 is strong hyper Markov when $\text{rank}(u_{ij}) < \infty$ for $i, j \in V$.

Proposition 2 Suppose that the collection of kernels \mathcal{U} satisfies that $\text{rank}(u_{ij}) < \infty$ for $i, j \in V$, then the hyper-inverse-Wishart-process prior constructed in Theorem 1 satisfies the strong hyper Markov property. That is, if $\mathcal{Q}_G \sim \text{HIWP}_G(\delta, \mathcal{U}_C)$, then for any decomposition (A, B) of G , $\mathcal{Q}_{B|A} \perp \mathcal{Q}_A$, where $\mathcal{Q}_{B|A}$ denotes the conditional distribution (e.g., conditional covariance) of \mathbf{c}_B given \mathbf{c}_A .

The finite rank condition for the prior parameters $\{u_{ij}\}$ in Proposition 2 is a relatively strong condition under which the HIWP $_G$ satisfies the strong hyper Markov property. It

implies that the covariance kernel \mathcal{Q}_G , thus the sequence \mathbf{c} , is a priori finite dimensional. Whether the strong hyper Markov property still holds without this condition remains a challenging open problem. In the online appendix, we have included several interesting results made through our preliminary study, which may provide useful insights into further investigations of this problem. The strong hyper Markov property of HIWP $_G$ ensures that the joint posterior of \mathcal{Q}_G (conditional on G) can be constructed from the marginal posterior of \mathcal{Q}_G (conditional on G) at each clique C , as stated in Theorem 2. Therefore one essentially transforms the Bayesian analysis to a sequence of sub-analyses at the cliques, which substantially reduces the size of the problem.

Theorem 2 Suppose that $\mathbf{c}_i \sim \text{dMGP}_G(\mathbf{c}_0, \mathcal{Q}_G)$, $i = 1, \dots, n$ are independent and identically distributed. Further assume that the prior of \mathcal{Q}_G is $\text{HIWP}_G(\delta, \mathcal{U}_C)$ where the collection of kernels \mathcal{U} satisfies that $\text{rank}(u_{ij}) < \infty$ for $i, j \in V$. Then the conditional posterior of \mathcal{Q}_G given $\{\mathbf{c}_i\}$ and G is $\text{HIWP}_G(\tilde{\delta}, \tilde{\mathcal{U}}_C)$, where $\tilde{\delta} = \delta + n$, $\tilde{\mathcal{U}}_C = \{\tilde{u}_{ij}, i, j \in C, C \in \mathcal{C}\}$ and $\tilde{u}_{ij} = u_{ij} + \sum_{r=1}^n (\mathbf{c}_r - \mathbf{c}_0) \otimes (\mathbf{c}_r - \mathbf{c}_0)$. Here \otimes denotes the outer product. Furthermore, the marginal distribution of $\{\mathbf{c}_i\}$ given $\{G, \mathbf{c}_0, \delta, \tilde{\mathcal{U}}_C\}$ is again Markov over G .

Theorem 2 implies that when $\text{rank}(u_{ij}) < \infty$ for $i, j \in V$, the HIWP $_G(\delta, \mathcal{U}_C)$ prior is a conjugate prior for \mathcal{Q}_G in the dMGP $_G(\mathbf{c}_0, \mathcal{Q}_G)$ likelihood. Note that here the likelihood, the prior, and the posterior are all conditional on G , which makes Bayesian inference of G tractable. Model (4) and results in Theorem 2 provide the theoretical foundation for practical Bayesian inference under a reasonable regularity condition, as discussed in Section 3.

3. Bayesian Posterior Inference

Despite the fact that functional data are realizations of inherently infinite-dimensional random processes, data can only be collected at a finite number of measurement points. Essentially, estimating the conditional independence structure of infinite-dimensional random processes based on a finite number of measurement points is an inverse problem and therefore requires regularization. Müller and Yao (2008) reviewed two main approaches for regularization in functional data analysis—finite approximation through, e.g., suitably truncating the basis expansion representation and penalized likelihood. In this paper, we suggest performing posterior inference based on approximating the underlying random processes with orthogonal basis functions. In particular, we assume the following regularity condition:

Condition 1 The functional data \mathbf{f} are observed discretely on a dense grid $\mathbf{t} = \bigsqcup_{j=1}^m \mathbf{t}_j$ with $\mathbf{t}_j = (t_{j1}, \dots, t_{jn_j(n)})$ and $n_j(n) \rightarrow \infty$ as $n \rightarrow \infty$. One can find $N_j(n)$ so that the underlying random process f_j can be approximated with an N_j -term orthogonal basis expansion $\tilde{f}_j = \sum_{l=1}^{N_j} c_{jl} \phi_{jl}$, with approximation error $\|f_j - \tilde{f}_j\|_2 = O_p(n^{-\beta})$ with $\beta \geq 1/2$ for all $j \in V$.

Essentially, Condition 1 requires that the discretely-measured functional data capture sufficient information about the underlying random processes, so that we can approximate each f_j with a negligible approximation error. This condition provides the consistency of the basis representation, i.e., the approximation error converges to zero with order $O_p(n^{-\beta})$

when M_j increases with the sample size n . We need such a condition in order to guarantee that the behavior of $\{f_j\}$ is not too outrageous. Certain assumptions, such as the decay rate of the eigenvalues of f_j , the smoothness property of f_j , or the characteristics of the basis functions, will determine the specific rate β (De Boor, 2001; Jansen and Oonincx, 2015). However, under our generic setup, since we prefer not to specify particular assumptions, we only require a mild range for the convergence rate. Condition 1 is a basic assumption in the functional setting, and a similar regularity condition has been adopted by Qiao et al. (2015) in a functional graphical model based on the group LASSO penalty.

3.1 Bayesian Posterior Inference under the Regularization Condition

The regularity from Condition 1 enables us to write the density functions of the Markov distributions and hyper Markov laws so that posterior inference can be practically implemented. Denoting $M = (M_1, \dots, M_p)$, we can explicitly write the density function for the truncated process $\mathbf{c}^M = (c_1^M, \dots, c_p^M)$, and an MCMC algorithm can then be designed for the posterior inference of the underlying graph G . The density function of \mathbf{c}^M is

$$p(\mathbf{c}^M | \mathbf{c}_0^M, \mathbf{Q}_C, G) = \frac{\prod_{C \in \mathcal{C}} p(\mathbf{c}_C^M | \mathbf{c}_{0_C}^M, \mathbf{Q}_C)}{\prod_{S \in \mathcal{S}} p(\mathbf{c}_S^M | \mathbf{c}_{0_S}^M, \mathbf{Q}_S)}, \quad (5)$$

where \mathbf{Q}_C is a block-wise covariance matrix with the (i, j) th block formed by $\{q_{ij}(k, l), k = 1, \dots, M_i, l = 1, \dots, M_j\}$, and $\mathbf{Q}_C, \mathbf{Q}_S$ are submatrices of \mathbf{Q}_C corresponding to clique C and separator S , respectively. The HIWPC prior of \mathbf{Q}_C induces a hyper inverse-Wishart prior with density

$$p(\mathbf{Q}_C | G) = \frac{\prod_{C \in \mathcal{C}} p(\mathbf{Q}_C | \delta, \mathbf{U}_C)}{\prod_{S \in \mathcal{S}} p(\mathbf{Q}_S | \delta, \mathbf{U}_S)}, \quad (6)$$

where $p(\mathbf{Q}_C | \delta, \mathbf{U}_C)$ is the density of inverse-Wishart defined in Dawid (1981), \mathbf{U}_C is a submatrix of \mathbf{U}_C corresponding to clique C , and \mathbf{U}_C is a block-wise matrix formed by $\{u_{ij}\}$ in the same way as \mathbf{Q}_C is formed by $\{q_{ij}\}$. The $p(\mathbf{Q}_S | \delta, \mathbf{U}_S)$ component in the denominator is defined similarly. Based on (5) and (6), and assuming that $\{c_i, i = 1, \dots, n\}$ is a random sample of \mathbf{c} , one can further integrate out \mathbf{Q}_C to get the marginal density

$$p(\{\mathbf{c}_i^M\} | \mathbf{c}_0^M, G) = (2\pi)^{-\frac{d}{2}(\sum_i M_i)} \frac{h(\delta, \mathbf{U}_C)}{h(\delta, \mathbf{U}_C)}, \quad (7)$$

where

$$h(\delta, \mathbf{U}_C) = \frac{\prod_{C \in \mathcal{C}} \frac{1}{2} |\mathbf{U}_C| \left(\frac{d+d_C-1}{2}\right) \Gamma_{d_C}^{-1} \left\{ \frac{1}{2}(\delta + d_C - 1) \right\}}{\prod_{S \in \mathcal{S}} \frac{1}{2} |\mathbf{U}_S| \left(\frac{d+d_S-1}{2}\right) \Gamma_{d_S}^{-1} \left\{ \frac{1}{2}(\delta + d_S - 1) \right\}},$$

and d_C and d_S are the dimensions of \mathbf{U}_C and \mathbf{U}_S respectively, and $\Gamma_b(a) = \pi^{b(b-1)/4} \prod_{l=0}^{b-1} \Gamma(a - l/2)$. The denominator $h(\delta, \mathbf{U}_C)$ in (7) is defined in the same way. Based on these results, posterior inference can be done through sampling from the posterior density

$$p(G | \{\mathbf{c}_i^M\}, \mathbf{c}_0^M) \propto p(\{\mathbf{c}_i^M\} | \mathbf{c}_0^M, G) p(G), \quad (8)$$

where $p(G)$ is the density function corresponding to the prior distribution $G \sim \Pi$, which is a discrete distribution supported on all decomposable graphs with p nodes. Giudici and Green (1999) used the discrete uniform prior $\Pr(G = G_0) = 1/d$ for any fixed p -node decomposable graph G_0 , where d is the total number of such graphs; Jones et al. (2005) used the independent Bernoulli prior with probability $2/(p-1)$ for each edge, which favors sparser graphs (Giudici, 1996). The following MCMC algorithm describes the steps to generate posterior samples based on (8).

Algorithm 1

Step 0. Set an initial decomposable graph G and set the prior parameters $\mathbf{c}_0, \delta,$ and \mathbf{U}_C .

Step 1. With probability $1 - q$, propose \tilde{G} by randomly adding or deleting an edge from G (each with probability 0.5) within the space of decomposable graphs; with probability q , propose \tilde{G} from a discrete uniform distribution supported on the set of all decomposable graphs. Accept the new \tilde{G} with probability

$$\alpha = \min \left\{ 1, \frac{p(\tilde{G} | \{\mathbf{c}_i^M\}, \mathbf{c}_0^M) p(G | \tilde{G})}{p(G | \{\mathbf{c}_i^M\}, \mathbf{c}_0^M) p(\tilde{G} | G)} \right\}.$$

Repeat Step 1 for a large number of iterations until convergence is achieved.

Detailed derivations are available in the online appendix. The above algorithm is a Metropolis-Hastings sampler with a mixture of local and heavier-tailed proposals, also called a *small-world sampler*. The ‘‘local’’ move involves randomly adding or deleting one edge based on the current graph, and the ‘‘global’’ move is achieved through the discrete uniform proposal. Guan et al. (2006) and Guan and Krone (2007) have shown that the small-world sampler leads to much faster convergence especially when the posterior distribution is either multi-modal or spiky.

3.2 Bayesian Posterior Inference for Noisy Functional Data

The theory in Section 2 and the posterior inference in Section 3.1 relies on the assumption that the distribution of \mathbf{f} (and \mathbf{c}) is Markov over G . In many situations, it is more desirable to make such an assumption in a hierarchical model. For example, when functional data are subject to measurement error, one might wish to incorporate an additive error term and consider the following model

$$y_{ijt} = f_{ij}(t) + \varepsilon_{ijt}, \quad i = 1, \dots, n, \quad j = 1, \dots, p, \quad t \in \mathbf{t}_j, \quad (9)$$

where $\{y_{ijt}, t \in \mathbf{t}_j\}$ are noisy observations measured on a dense grid $\mathbf{t}_j = (t_{j1}, \dots, t_{jm_j})$, $\{f_{ij}\}$ are the underlying true functions, and $\{\varepsilon_{ijt}, t \in \mathbf{t}_j\}$ are measurement errors. We assume that $\{f_{ij}\}$ and $\{\varepsilon_{ijt}, t \in \mathbf{t}_j\}$ are mutually independent of each other. The inference of model (9) involves both smoothing (i.e., estimating f_{ij}) and estimation of the underlying graph G . We achieve these goals simultaneously through fitting a Bayesian hierarchical model.

In particular, we assume that $\{f_{ij}\}$ are Gaussian processes in $L^2(\mathcal{T}_j)$, and denote $\{c_{ijk}\}$ their basis coefficients corresponding an orthonormal basis $\{\phi_{jk}\}_{k=1}^\infty$. With this representation, model (9) has the form $y_{ijt} = \sum_{k=1}^\infty c_{ijk} \phi_{jk}(t) + \varepsilon_{ijt}$, and $\{c_{ijk}\}$ is a discrete Gaussian

process. We further assume that the measurement error $\epsilon_{tj} = \{\epsilon_{tjh}, t \in \mathbf{t}_j\}$ is Gaussian while noise with variance σ_j^2 , i.e., $\epsilon_{tjh} \sim N(0, \sigma_j^2)$ independently across all t for $t \in \mathbf{t}_j$. Truncating at the M_j th basis element, we can reparameterize the model as

$$y_{jt} = \sum_{k=1}^{M_j} c_{tjk} \phi_{jk}(t) + \tilde{\epsilon}_{tjh}, \quad t \in \mathbf{t}_j \quad (10)$$

where $\tilde{\epsilon}_{tjh}$ is a new residual term that consists of the approximation error of the truncated series $(\epsilon_{tG}, \sum_{k=M_j+1}^{\infty} c_{tjk} \phi_{jk}(t))$ and the measurement error. If we concatenate the noisy observations to form a vector

$$\mathbf{y}_i = (y_{i1t_1}, \dots, y_{iM_1t_1}, \dots, y_{iM_p t_p}, \dots, y_{iM_p t_p})^T$$

and denote \mathbf{c}_i^M the vector formed by the basis coefficients $\{c_{tjk}, j = 1, \dots, p, k = 1, \dots, M_j\}$, then model (10) can be written as $\mathbf{y}_i = \Phi \mathbf{c}_i^M + \tilde{\epsilon}_i$, where $\Phi = \text{diag}\{\phi_{11}, \dots, \phi_{p1}\}$ is a $\sum_j M_j$ by $\sum_j M_j$ block-diagonal matrix with the j th diagonal block containing $\phi_j = [\phi_{j1}(\mathbf{t}_j), \dots, \phi_{jM_j}(\mathbf{t}_j)]$, and $\tilde{\epsilon}_i$ denote the concatenated vector of the new residual terms. We assume that $\tilde{\epsilon}_i \sim N(0, \Lambda)$ where $\Lambda = \text{diag}(s_{1m_1}^2, \dots, s_{pM_p}^2)$. Notice that if $\mathbf{Q}_C = \text{cov}(\mathbf{c}_i^M)$, then $\text{cov}(\mathbf{y}_i^M) = \Phi \mathbf{Q}_C \Phi^T + \Lambda$. The diagonals of $\Phi \mathbf{Q}_C \Phi^T$ and Λ can not be separately identifiable. Therefore, we treat Λ as a fixed model parameter, whose quantity can be pre-determined by the approximation $s_j^2 \approx \hat{\sigma}_j^2$, where $\hat{\sigma}_j^2$ is the estimation of σ_j^2 using local smoothing on $\{\tilde{\epsilon}_{tjh}\}$.

Applying a prior for \mathbf{c}_i^M in the form of (5) (conditional on G) and the HMWP prior for the covariance matrix \mathbf{Q}_C in the form of (6), we obtain the density function for the joint posterior

$$p(\{\mathbf{c}_i^M\}, \mathbf{Q}_C, G | \{\mathbf{y}_i\}) \propto \prod_{i=1}^n p(\mathbf{y}_i | \mathbf{c}_i^M, \Lambda) p(\mathbf{c}_i^M | \mathbf{c}_0^M, \mathbf{Q}_C, G) p(\mathbf{Q}_C | G) p(G). \quad (11)$$

From (11), we can integrate out \mathbf{Q}_C to obtain the marginal posterior distribution of $\{\mathbf{c}_i^M\}$ and G . The MCMC algorithm for generating posterior samples based on (11) is listed in Algorithm 2.

Algorithm 2

Step 0 Set initial values for $\{\mathbf{c}_i^M\}$, G and set the model parameters δ , \mathbf{c}_0^M , \mathbf{U} and Λ .

Step 1 Conditional on $\{\mathbf{c}_i^M\}$, update $G \sim p(G | \{\mathbf{c}_i^M\}, \mathbf{c}_0^M)$ using the small-world sampler as described in Step 1 of Algorithm 1, where $p(G | \{\mathbf{c}_i^M\}, \mathbf{c}_0^M)$ is computed based on (11).

Step 2 Given G , update $\mathbf{Q}_C \sim p(\mathbf{Q}_C | \{\mathbf{c}_i^M\}, G)$, which takes the same form as (6) except that δ and \mathbf{U} are replaced by δ and $\tilde{\mathbf{U}}$ respectively using the formulae in Theorem 2.

Step 3 Conditional on G and \mathbf{Q}_C , update $\mathbf{c}_i^M \sim N(\boldsymbol{\mu}_i, \mathbf{V})$, where $\mathbf{V} = (\Phi^T \Lambda^{-1} \Phi + \mathbf{Q}_C^{-1})^{-1}$ and $\boldsymbol{\mu}_i = \mathbf{V}(\Phi^T \Lambda^{-1} \mathbf{y}_i + \mathbf{Q}_C^{-1} \mathbf{c}_0^M)$.

Repeat Step 1 \sim 3 for a large number of iterations until convergence is achieved.

3.3 Other Practical Computational Issues

Calculating the coefficient sequences $\{c_j\}$ from the functional observations $\{\mathbf{f}_j\}$ requires the selection of an orthonormal basis $\{\phi_{jk}, j = 1, \dots, p, k = 1, \dots, \infty\}$. If a known basis is chosen (e.g., Fourier), the coefficient sequences can be estimated by $c_{tjk} = \langle \mathbf{f}_{tj}, \phi_{jk} \rangle$ using numerical integration. Another convenient choice is the eigenbasis of the autocovariance operators of $\{\mathbf{f}_j\}$, in which case the coefficient sequences are called functional principal component (FPC) scores. The corresponding basis representation is called Karhunen-Loève expansion. The eigenbasis can be estimated using the method of Ramsay and Silverman (2005) or the Principal Analysis by Conditional Expectation (PACE) algorithm of Yao et al. (2005). Owing to the rapid decay of the eigenvalues, the eigenbasis provides a more parsimonious and efficient representation compared with other bases. Furthermore, the FPC scores within a curve are mutually uncorrelated, so one may set the prior parameter \mathbf{U}_C to be a matrix with blocks of diagonal sub-matrices, or simply a diagonal matrix.

In addition to the estimation of coefficient sequences, a suitable truncation of the infinite sequences $\{c_j\}$ is needed to facilitate practical posterior inference. We suggest to pre-determine the truncation parameters using approximation criteria, following Rice and Silverman (1991) or Yao et al. (2005). This includes cross-validation (Rice and Silverman, 1991), applying the pseudo Akaike information criterion (Yao et al., 2005), or controlling the fraction-of-variance-explained (FVE) in the FPC analysis (Lei et al., 2014).

4. Simulation Study

Three simulation studies were conducted to assess the performance of posterior inference using the Gaussian process graphical models outlined in Section 2.3 and Section 3. Simulation 1 corresponds to the smooth functional data case (without measurement error), and Simulation 2 corresponds to the noisy data case when measurement error is considered. Both simulations are based on a true underlying graph with 6 nodes, demonstrated in Figure 1 (a). In simulation 3, we show the performance of the proposed Bayesian inference in a $p > n$ case, with the number of nodes $p = 60$ and the sample size $n = 50$.

4.1 Simulation 1: Graph Estimation for Smooth Functional Data

Multivariate functional data are generated on the domain $[0, 1]$ using Fourier basis with the number of basis functions $\{M_j\}_{j=1}^p$ varying from 3 to 7. The true eigenvalues are generated from Gamma distributions and are subject to exponential decay. The conditional independence structure is determined by a $p \times p$ correlation matrix \mathbf{R}_0 , with the inverse \mathbf{R}_0^{-1} containing a zero pattern corresponding to the graph in Figure 1 (a). We then generate principal component scores from a multivariate normal distribution with zero mean and a block-wise covariance matrix $\mathbf{Q} = \mathbf{Z}\mathbf{R}\mathbf{Z}$, which has dimension $\sum_{j=1}^p M_j$. Here \mathbf{R} is a block-wise correlation matrix that has a diagonal form in each block. In particular, the (i, j) th block of \mathbf{R} , denoted by \mathbf{R}_{ij} , satisfies that $\mathbf{R}_{ij} = (\mathbf{R}_0)_{ij} \mathbf{I}$ where \mathbf{I} is a rectangular identity matrix with size $M_i \times M_j$. An image plot of \mathbf{R} is shown in Figure 1(d), with its data-domain counterpart (the correlation of \mathbf{f} evaluated on a grid \mathbf{t}) shown in Figure 1(c). The multivariate functional data are finally generated through linearly combining the eigenbasis using the principal component scores. A common mean function is added to each curve.

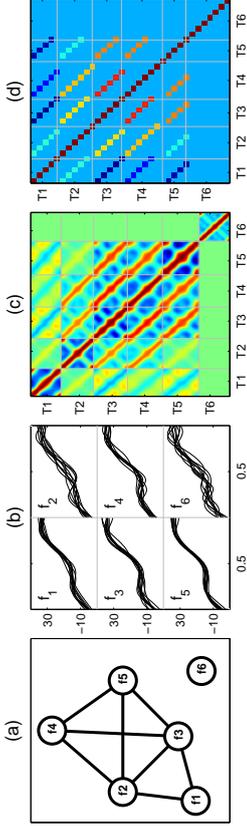


Figure 1: Plots of Simulation 1. (a) The true underlying graph; (b) The first 10 samples of $\{f_{ij}, j = 1, \dots, 6\}$; (c) The image plot of the underlying data-domain correlation matrix; (d) The image plot of the underlying correlation matrix \mathbf{R} .

The generated data contain $n = 200$ independent samples, and each sample contains six curves measured on six different grids. We display the first 10 samples in Figure 1(b).

Based on the data generated above, we estimate the principal component scores $\{c_j\}$ using the PACE algorithm of Yao et al. (2005) and determine the truncation parameter $\{M_j\}$ using the FVE criterion with a 90% threshold, resulting in $\{M_j\}$ values around 5. We apply Algorithm 1 and set $\delta = 5$ and $\mathbf{U} = \widehat{\mathbf{R}}\widehat{\mathbf{R}}^T$, where $\widehat{\mathbf{R}} = \text{diag}\{\widehat{\lambda}_{jk}^{1/2}, k = 1, \dots, M_j, j = 1, \dots, p\}$, $\{\widehat{\lambda}_{jk}\}$ are the estimated eigenvalues and $\widehat{\mathbf{R}}$ is set to be the identity matrix. A total of 5,000 MCMC iterations are performed. Starting from the empty graph, the chain reaches the true underlying graph in around 500 iterations. We have also tried implementing Algorithm 1 with different initial graphs; all implementations resulted in the same posterior mode at the true underlying graph.

We compare the performance of our approach with three other related methods: the Gaussian graphical model of Jones et al. (2005) based on Metropolis-Hastings (GGM-MH), the graphical LASSO (GLASSO) of Friedman et al. (2008), and the matrix-normal graphical model (MNGM) of Wang and West (2009). As both GGM-MH and GLASSO assume that each node is associated with one variable, we reduce the dimension of the functional data by retaining only the first principal component score. The MNGM method assumes matrix data, so we take the first five principal component scores and stack them up to form a 6×5 matrix for each sample. In the MNGM method, graph estimates across the rows and columns are obtained simultaneously, and only that across the rows is of interest to us.

The simulation results are demonstrated in the top panel of Table 1. Summary statistics, such as running-time, mis-estimation rate, sensitivity and specificity are calculated for each method. The running-time was obtained using a laptop with Intel(R) Core(TM) i5 CPU, M430 with 2.27 GHz processor and 4GB RAM. The comparison of running-time shows that the GLASSO method is the fastest. This is because GLASSO does not require posterior sampling. However, GLASSO relies on a penalized optimization approach which requires determination of the tuning parameter. In this simulation, we have selected the tuning parameter that results in the lowest mis-estimation rate with respect to the underlying true

graph. When the true graph is unknown, the tuning procedure can be time-consuming. The MNGM is much slower to implement, perhaps due to the numerical approximation of the marginal density in the MCMC algorithm.

Data	Method	nFPC	Time	nEdge	nUnique	MisR	Sen	Spec
Smooth	FDGM-S	3 - 5	38	7.66	3	0.02	0.96	1.0
	GGM-MH	1	0.15	9.55	63	0.10	1.0	0.78
	GLASSO	1	-	-	-	0.13	-	-
Noisy	MNGM	5	4067.73	5.83	36	0.21	0.66	0.93
	FDGM-N	3 - 5	64	7.86	5	0.01	0.98	1.0
	GGM-MH	1	0.39	9.62	59	0.11	1.0	0.77
	GLASSO	1	-	-	-	0.13	-	-
	MNGM	5	4086.38	6.33	18	0.26	0.65	0.85

Table 1: Summary statistics of simulation 1 and 2. nFPC: number of FPCs used to approximate each curve; Time: running time (in seconds) based on 5000 MCMC iterations; nEdge: total number of edges of the graph averaged across all posterior samples; nUnique: number of unique graphs visited after the burnin period; MisR: mean mis-estimation rate with respect to the true graph; Sen: sensitivity; Spec: specificity; FDGM-S: the proposed functional data graphical model for smooth data, based on Algorithm 1; FDGM-N: the proposed functional data graphical model for noisy data, based on Algorithm 2; GGM-MH: Gaussian graphical model; GLASSO: graphical LASSO; MNGM: matrix-normal graphical model.

In Table 1, the mis-estimation rate is defined as the proportion of mis-estimated edges, obtained by averaging across all posterior samples. The sensitivity is the proportion of missed edges among the true edges, and the specificity is the proportion of over-estimated edges among the true non-edge pairs. The top panel of Table 1 shows that the proposed functional data graphical model provides the smallest mis-estimation rate as well as the highest sensitivity and specificity. We also observe that, although relying on excessive dimension reduction, the Gaussian graphical model and the GLASSO still provide reasonably good estimates. This suggests that for problems involving more nodes (>50), we can use these methods to obtain an initial estimate before applying our approach.

4.2 Simulation 2: Graph Estimation for Noisy Functional Data

We add Gaussian white noise to the functional data generated in Simulation 1 to demonstrate the performance of posterior inference for noisy data. The variances of the additive Gaussian white noise $\{\varepsilon_{ij}, t \in \mathbf{t}_j\}$ are generated from a gamma distribution with mean 2.5 and variance 0.25, resulting in a signal-to-noise ratio around 9, where the signal-to-noise ratio is defined by $f_{ij}(t)/\text{var}\{\varepsilon_{ij}\}$ and is averaged across the grid points and the samples. We apply model (11) and generate posterior samples using Algorithm 2. The eigenbasis and the variance of the noise are estimated simultaneously using the PACE algorithm. The parameter \mathbf{A} is determined using the estimated variance of the Gaussian white noise, and the other model parameters are set to be the same as in Simulation 1. The posterior infer-

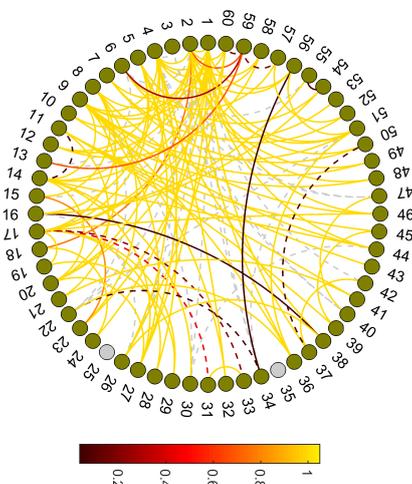


Figure 2: Plot of Simulation 3. The estimated graph based on the marginal inclusion probability for each edge.

ence results are compared with the other three methods in the bottom panel of Table 1. Similar patterns are observed as in Simulation 1. In particular, the proposed functional data graphical model shows a clear advantage in accurately estimating the graph. Estimates of the functions $\{f_j\}$ and their time-domain correlations are provided in the online appendix.

4.3 Simulation 3: Graph Estimation When p is Greater than n

To further investigate the performance of the proposed approach when the number of nodes p is greater than the sample size n , we design another simulation study with $p = 60$ and $n = 55$. The true graph contains 60 nodes, among which 2 are singletons and 58 are connected with edges. The total number of edges in the true graph is 121. Smooth functional data are simulated following the procedure described in Section 4.1. With the simulated data, we apply the PACE algorithm to estimate $\{c_j\}$ and determine the truncation parameters using the FVE criterion with a 95% threshold. We then apply Algorithm 1 and set prior parameters δ and \mathbf{U} following Simulation 1. Posterior samples of the graph are obtained for 30,000 MCMC iterations after removing 10,000 burn-in samples.

The posterior inference results are summarized in a circular graph plot in Figure 2, where we show an estimated graph by thresholding the marginal inclusion probability for each edge—the proportion that each edge is included in the posterior samples—to be greater than 0.03. In Figure 2, the colors indicate the levels of the marginal inclusion probabilities, the colored dashed lines indicate edges that are mistakenly estimated, and the gray dashed lines indicate edges that are missed. This gives 105 estimated edges, among which 98 are correctly estimated, and 7 are mistakenly estimated. Additionally, 23 edges in the true graph are missed. We have also calculated the summary statistics similarly as in previous

simulations, resulting in mean mis-estimation rate 0.02, sensitivity 0.77, and specificity 0.99. Extra simulation runs show that the sensitivity level is improved when we increase the sample size n .

5. Analysis of Event-related Potential Data in an Alcoholism Study

We apply the proposed method to event-related potential data from an alcoholism study. Data were initially obtained from 64 electrodes placed on subjects' scalps that captured EEG signals at 256 Hz during a one-second period. The measurements were taken from 122 subjects, of which 77 belonged to the alcoholism group and 45 to the control group. Each subject completed 120 trials. During each trial, the subject was exposed to either a single stimulus (a single picture) or two stimuli (a pair of pictures) shown on a computer monitor. We band-pass filtered the EEG signals to extract the α frequency band in the range of 8–12.5 Hz. The filtering was performed by applying the `egfilt` function in the EEGLAB toolbox of Matlab. The α -band signal is known to be associated with inhibitory control (Knyazev, 2007). Research has shown that, relative to control subjects, alcoholic subjects demonstrate unstable or poor rhythm and lower signal power in the α -band signal (Porjesz et al., 2005; Finn and Justus, 1999), indicating decreased inhibitory control (Sher et al., 2005). Moreover, regional asymmetric patterns have been found in alcoholics—alcoholics exhibit lower left α -band activities in anterior regions relative to right (Hayden et al., 2006). In this study, we aim to estimate the conditional independence relationships of α -band signals from different locations of the scalp, and expect to find evidence that reflects differences in brain connectivity and asymmetric pattern between the two groups.

Since multiple trials were measured over time for each subject, the EEG measurements may not be treated as independent due to the time dependence of the trials. Furthermore, since the measurements were taken under different stimuli, the signals could be influenced by different stimulus effects. To remove the potential dependence between the measurements and the influence of different stimulus types, for each subject, we averaged the band-filtered EEG signals across all trials under the single stimulus, resulting in one *Event-related potential* (ERP) curve per electrode per subject. ERP is a type of electrophysiological signal generated by averaging EEG segments recorded under repeated applications of a stimulus, with the averaging serving to reduce biological noise levels and enhance the stimulus evoked neurological signal (Brandeis and Lehmann, 1986; Bressler, 2002). Based on the preprocessed ERP curves, we further removed subjects with missing nodes, and balanced the sample size across the two groups, producing multivariate functional data with $n = 44$ and $p = 64$ for both the alcoholic and the control group. We applied model (4) using coefficients of the eigenbasis expansion. The number of eigenbasis $\{M_j\}$ was determined through retaining 90% of the total variation; this resulted in 4–7 coefficients per f_j . We collected 30,000 posterior samples using Algorithm 1, in which the first 10,000 were treated as the burn-in period. The model was fitted for both the alcoholic and the control group, and convergence of the MCMC was justified by running multiple chains starting with various initial values.

The posterior results are summarized in Figure 3. The plots in (a) and (b) show the marginal inclusion probabilities for edges in the alcoholic and the control group respectively, where the edge color indicates the proportion that each edge is included in the posterior

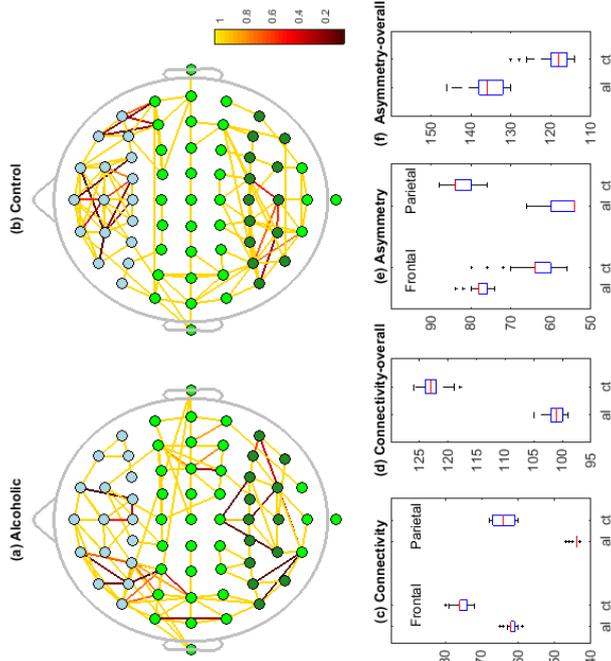


Figure 3: Summary of posterior inference: the marginal inclusion probabilities for edges in the alcoholic group (a) and the control group (b); the boxplots of connectivity measures: the number of edges connecting with nodes in the frontal and parietal regions (c), and the overall total number of edges (d); the boxplots of asymmetry measures: the number of asymmetric edges for nodes in the frontal and the parietal regions (e), and the overall total number of asymmetric edges (f). In (a) and (b), the edge color indicates the magnitude of the posterior inclusion probability. In (c)–(f), the alcoholic group is abbreviated as “al”, and the control group is abbreviated as “ct”.

samples. To distinguish different regions, we used light blue to highlight nodes in the frontal region, used dark green to highlight nodes in the parietal region, and used green to indicate nodes in the central and occipital regions. Comparing (a) with (b), we see that the alcoholic group contains more edges connecting the left frontal-central, right central, and right parietal regions than the control group. The control group, on the other hand, contains more edges connecting the middle and right frontal regions, as well as the left parietal region than the alcoholic group.

To further compare with established results, we calculated two summary statistics for connectivity: the number of edges connected with nodes in a specific region, and the overall total number of edges. We also calculated two additional summary statistics for asymmetry: the number of asymmetric edges for all nodes in a specific region, and the overall total number of asymmetric edges. We summarized these summary statistics across the two groups using boxplots in Figure 3 (c)–(f), and calculated the posterior probability that the alcoholic group is greater than, equal to, or less than the control group for each statistic. Results show that, with probability ≈ 1 , the alcoholic group has fewer edges than the control group in the frontal and the parietal region, and has fewer overall total number of edges; with probability 0.95, the alcoholic group has more asymmetric edges than the control group in the frontal region; and with probability ≈ 1 , the alcoholic group has higher overall total number of asymmetric edges than the control group. These results indicate that the alcoholic group exhibits decreased regional and overall connectivity, increased asymmetry in the frontal region, and increased overall asymmetry. These observations are consistent with the findings of Hayden et al. (2006), who studied the asymmetric patterns at two frontal electrodes (F3, F4) and two parietal electrodes (P3, P4) using the analysis of variance method based on the resting-state α -band power. In comparison, our analysis provides connectivity and asymmetric pattern of all 64 electrodes simultaneously whereas Hayden et al. (2006) only focuses on the four representative electrodes.

6. Discussion

We have constructed a theoretical framework for graphical models of multivariate functional data and proposed a HIWP prior for the special case of Gaussian process graphical models. For practical implementation, we have suggested a posterior inference approach based on a regularization condition, which enables posterior sampling through MCMC algorithms.

One concern is whether it is possible to perform exact posterior inference without the regularity condition on approximation, i.e., inferring the graph directly from the joint posterior $p(G|\{\mathbf{c}_i\}) \propto p(\{\mathbf{c}_i\}|G)p(G)$ based on model (4), where $p(\{\mathbf{c}_i\}|G)$ is the marginal likelihood (with the covariance kernel \mathcal{Q}_C integrated out) and $p(G)$ is the prior distribution for G . Although the above joint posterior is theoretically well-defined according to Theorem 2, exact posterior sampling is difficult due to the fact that the density function for the marginal likelihood can only be calculated on a finite dimensional projection of $\{\mathbf{c}_i\}$.

In posterior inference, the influence of the approximation error on the posterior distribution can be quantified empirically. Assuming that the functional data are pre-smoothed, the approximation error can be quantified by calculating the difference of the ℓ^2 norms between the full sequence and the truncated sequence. The influence on the posterior distribution can be quantified by measuring the sensitivity of the posterior distribution to the change of truncation (Saltelli et al., 2000). For example, based on model (4) one may calculate the Kullback-Leibler divergence for two different truncation parameters M and M' . An alternative method for pre-determining the truncation parameter is to choose a prior for M in a Bayesian hierarchical model, in which case hybrid MCMC algorithms are needed for fitting both models (4) and (11). The posterior sampling in these models would become

more complicated because the dimension of the truncated sequences and the size of the covariance matrix \mathbf{Q}_P would change whenever M is updated.

We have demonstrated the application of the proposed approach through an ERP data set. By treating ERPs as functional data, we are estimating the systematic brain connectivity that is common across a group of subjects and a time interval. For other modeling purposes, such as estimating the individual level or dynamic brain connectivity, one could use multivariate graphical models described in Carralho and West (2007) or Bihnes (2010).

We have focused on decomposable graphs. In case of non-decomposable graphs, the proposed HiWP prior may still apply if we replace the inverse-Wishart process prior for each clique with that for a prime component of the graph. For a non-complete prime component P , the inverse-Wishart processes prior for \mathbf{Q}_P is subject to extra constraint induced by missing edges.

We have applied the proposed method to graphs of small to moderate size, with number of nodes as large as 60. To deal with larger scale problems (e.g. multivariate functional data with hundreds or thousands of functional components), more efficient large-scale computational techniques such as the fast Cholesky factorization (Li et al., 2012) can be readily combined with our MCMC algorithms. Furthermore, non-MCMC algorithms may be more computationally efficient in case of large graphs. For example, based on the posterior distribution of G in (8), a fast search algorithm may be developed to search for the maximum a posteriori (MAP) solution following ideas similar to Dauriné III (2007) and Jalali et al. (2011).

Acknowledgments

This material was based upon work partially supported by the National Science Foundation under Grant DMS-1127914 to the Statistical and Applied Mathematical Sciences Institute. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Hongxiao Zhu's research is supported by National Science Foundation (NSF-DMS 1611901). David B. Dunson's research is supported by Office of Naval Research (N00014-14-1-0245).

Appendix A. Definitions

Definitions used in the lemmas, theorems and their proofs are listed as follows: (I) *Projection map*. Let \mathbb{R} be the real line and T be an index set. Consider the Cartesian product space $\mathbb{R}^{T \times T} = \prod_{(\alpha, \beta) \in T \times T} \mathbb{R}^{(\alpha, \beta)}$. For a fixed point $(\alpha, \beta) \in T \times T$, we define the projection map $\pi_{(\alpha, \beta)} : \mathbb{R}^{T \times T} \rightarrow \mathbb{R}^{(\alpha, \beta)}$ as $\pi_{(\alpha, \beta)}(\{x_{(l,m)} : (l, m) \in T \times T\}) = x_{(\alpha, \beta)}$. For a subset $B \subset T \times T$, we define the partial projection $\pi_B : \mathbb{R}^{T \times T} \rightarrow \mathbb{R}^B$ as $\pi_B(\{x_{(l,m)} : (l, m) \in T \times T\}) = \{x_{(s,t)} : (s, t) \in B\}$. More generally, for subsets B_1, B_2 , such that $B_2 \subset B_1 \subset T \times T$, we define the partial sub-projections $\pi_{B_2 \leftarrow B_1} : \mathbb{R}^{B_1} \rightarrow \mathbb{R}^{B_2}$, by $\pi_{B_2 \leftarrow B_1}(\{x_{(l,m)} : (l, m) \in B_1\}) = \{x_{(s,t)} : (s, t) \in B_2\}$. (II) *The pullback of a σ -algebra*. Let $\mathcal{B}_{(\alpha, \beta)}$ be a σ -algebra on $\mathbb{R}^{(\alpha, \beta)}$. We can create a σ -algebra on $\mathbb{R}^{T \times T}$ by pulling back the $\mathcal{B}_{(\alpha, \beta)}$ using the inverse of the projection map and define $\pi_{(\alpha, \beta)}^*(\mathcal{B}_{(\alpha, \beta)}) = \{\pi_{(\alpha, \beta)}^{-1}(A) : A \in \mathcal{B}_{(\alpha, \beta)}\}$. One can verify that $\pi_{(\alpha, \beta)}^*(\mathcal{B}_{(\alpha, \beta)})$ is a σ -algebra. (III) *Product σ -algebra*. We define the product σ -algebra as $\mathcal{B}(\mathbb{R}^{T \times T}) = \prod_{(\alpha, \beta) \in T \times T} \mathcal{B}_{(\alpha, \beta)}$, where $\prod_{(\alpha, \beta) \in T \times T} \mathcal{B}_{(\alpha, \beta)} = \sigma\left(\bigcup_{(\alpha, \beta) \in T \times T} \pi_{(\alpha, \beta)}^*(\mathcal{B}_{(\alpha, \beta)})\right)$. (IV) *Pushforward measure*. Given a measure $\mu_{T \times T}$ on the product σ -algebra, and a subset B of $T \times T$, we define the pushforward measure $\mu_B = (\pi_B)_* \mu_{T \times T}$ on \mathbb{R}^B as $\mu_B(A) = \mu_{T \times T}(\pi_B^{-1}(A))$ for all $A \in \mathcal{B}_B$, where $\mathcal{B}_B = \prod_{(\alpha, \beta) \in B} \mathcal{B}_{(\alpha, \beta)}$. (V) *Compatibility*. Given subsets B_1, B_2 of $T \times T$ such that $B_2 \subset B_1 \subset T \times T$, the pushforward measures μ_{B_1} and μ_{B_2} are said to obey compatibility relation if $(\pi_{B_2 \leftarrow B_1})_* \mu_{B_1} = \mu_{B_2}$.

Appendix B. Proof of Lemma 1

This proof involves some measure-theoretic arguments. The essential idea is to use disintegration theory Chang and Pollard (1997) to first construct the conditional probability measure $P_1\{\cdot | \pi_{A \cap B}(\mathbf{f}_A)\}$ on $\mathcal{B}(L^2(T_A))$, extend this to $P\{\cdot | \pi_B(\mathbf{f})\}$ on $\mathcal{B}(L^2(T_{A \cup B}))$, and finally construct the joint measure P which satisfies conditions (i)–(iii).

Denote $T_A = \bigcup_{j \in A} T_j$. Since P_1 is a finite Radon measure and the projection $\pi_{A \cap B} : L^2(T_A) \rightarrow L^2(T_{A \cap B})$ is measurable, we invoke the disintegration theorem to obtain measures $P_1\{\cdot | \pi_{A \cap B}(\mathbf{f}_A)\}$ on $\mathcal{B}(L^2(T_A))$ satisfying:

$$(a.1) \quad P_1(\mathcal{X} | \mathbf{f}_{A \cap B}) = P_1\{\mathcal{X} \cap [L^2(T_{A \cap B}) \times \{\pi_{A \cap B}(\mathbf{f}_A)\}] | \pi_{A \cap B}(\mathbf{f}_A)\},$$

$$(b.1) \quad \text{the map } \mathbf{f}_{A \cap B} \mapsto (P_1)_{\mathbf{f}_{A \cap B}} H : = \int H(\mathbf{f}_A) dP_1(\mathbf{f}_A | \mathbf{f}_{A \cap B}) \text{ is measurable for all non-negative measurable } H : L^2(T_A) \rightarrow \mathbb{R},$$

$$(c.1) \quad P_1 H = ((\pi_{A \cap B})_* P_1)(P_1)_{\mathbf{f}_{A \cap B}} H \text{ for all nonnegative measurable } H : L^2(T_A) \rightarrow \mathbb{R},$$

where $(\pi_{A \cap B})_* P_1$ is the push-forward measure of P_1 .

Now, we define the measure $P\{\cdot | \pi_B(\mathbf{f})\}$ by setting $P\{\mathcal{A} | \pi_B(\mathbf{f})\} = P_1\{\pi_A(A \cap [L^2(T_{A \setminus B}) \times \{\pi_B(\mathbf{f})\}]) | \pi_{A \cap B}(\mathbf{f})\}$. Note that this is well defined for all measurable $A \in \mathcal{B}(L^2(T_{A \cup B}))$ since the sections $\pi_A(A \cap [L^2(T_{A \setminus B}) \times \{\pi_B(\mathbf{f})\}])$ are always measurable, and also that (a) $P\{\mathcal{A} | \pi_B(\mathbf{f})\} = P\{\mathcal{A} \cap [L^2(T_{A \setminus B}) \times \{\pi_B(\mathbf{f})\}] | \pi_B(\mathbf{f})\}$ holds by construction. Now, let \mathcal{M} denote the set of measurable functions from $L^2(T_{A \cup B})$ to \mathbb{R} satisfying (b) $\mathbf{f}_B \mapsto P_B H$ is a measurable function on $L^2(T_B)$. We shall argue that \mathcal{M} is a monotone class. First,

suppose H_n is a sequence of positive measurable functions in \mathcal{M} increasing pointwise to a bounded measurable function H . For each fixed \mathbf{f}_B in $L^2(T_B)$, we then have that H_n is a sequence of positive measurable functions increasing pointwise to H , and hence the monotone convergence theorem implies $P_{T_B} H_n \rightarrow P_{T_B} H$ in an increasing manner. Since this holds for each \mathbf{f}_B , we conclude that $P_{T_B} H$ is the point-wise increasing limit of measurable functions on $L^2(T_B)$, and hence it is measurable. Moreover, it is simple to see that $P_{T_B} \mathbf{1}_{\mathcal{X} \times \mathcal{Y}} = P_1(\mathcal{X} | \mathbf{f}_{A \cap B}) \mathbf{1}_{\mathcal{Y}}(\mathbf{f}_{B \setminus A})$ is a measurable function on $L^2(T_B)$ for all $\mathcal{X} \in \mathcal{B}(L^2(T_A))$ and $\mathcal{Y} \in \mathcal{B}(L^2(T_{B \setminus A}))$, and hence $\mathbf{1}_{\mathcal{X} \times \mathcal{Y}} \in \mathcal{M}$. By the Monotone Class Theorem, we then have that all bounded measurable functions on $L^2(T_{A \cup B})$ satisfy (b), and hence it will hold for all positive measurable functions on $L^2(T_{A \cup B})$. Since (b) is satisfied for all positive measurable functions, we may define the measure $PH = P_2 P_{T_B} H$. By construction, we have that $P_1 L^2(T_{A \setminus B}) \times \{\mathbf{f}_{A \cap B}\} = P_2 P_1(L^2(T_{A \setminus B}) | \mathbf{f}_{A \cap B}) \mathbf{1}_{\mathcal{Y}}(\mathbf{f}_B) = P_2(\mathcal{Y})$ and $P_1 \mathbf{1}_{\mathcal{X} \times L^2(T_{B \setminus A})} = P_2 P_1(\mathcal{X} | \mathbf{f}_{A \cap B}) = ((\pi_{A \cap B})_* P_2) P_1(\mathcal{X} | \mathbf{f}_{A \cap B}) = ((\pi_{A \cap B})_* P_1) P_1(\mathcal{X} | \mathbf{f}_{A \cap B}) = P_1(\mathcal{X})$. Thus, we also have that $PH = P_2 P_{T_B} H = ((\pi_B)_* P) P_{\pi_B(\mathcal{G})} H$ for all measurable H , and this is the final property establishing that $P(\cdot | \mathbf{f}_B)$ is a disintegration of P with respect to the map π_B . By the disintegration theorem, this disintegration is a version of the regular conditional probability of \mathbf{f}_A given \mathbf{f}_B . Since this version only depends upon $\mathbf{f}_{A \cap B}$, we conclude that (iii) holds. Finally, we note that any other measure satisfying these properties must agree with the measure we have constructed on π -system, and therefore the uniqueness of P immediately follows. ■

Appendix C. Proof of Proposition 1

Proof. The Properties 1 - 4 in Dawid and Lauritzen (1993) are treated as axioms; they are universal properties thus also hold when X, Y, Z are random processes. Since the graph G is undirected and decomposable, the results on graphical theory in Appendix A of Dawid and Lauritzen (1993) continue to hold. Properties 1 - 4 and results in Appendix A imply that results in B1- B7 of Dawid and Lauritzen (1993) continue to hold when P is a Markov distribution constructed in Lemma 1. Theorem 2.6 and Corollary 2.7 of Dawid and Lauritzen (1993) are also implied. These results, combined with the definition of marginal distribution defined by pushforward measure and the definition of conditional probability measure based on disintegration theory, prove that Lemmas 3.1, 3.3, Theorems 3.9 - 3.10 as well as Propositions 3.11, 3.13, 3.15, 3.16, 3.18 from Dawid and Lauritzen (1993) hold. ■

Appendix D. Lemma 2 and Proof

Lemma 2 Let \mathbb{N} be the set of positive integers and I an arbitrary finite subset of it. Suppose that $\delta > 4$ is a positive integer and that $u : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{R}$ is a symmetric positive semidefinite and trace class kernel so that the matrix $\mathbf{U}_{I \times I}$ formed by $\{u(i, j), i, j \in I\}$ is symmetric positive semidefinite. Then there exists a unique probability measure μ on $(\mathbb{R}^{\mathbb{N} \times \mathbb{N}}, \mathcal{B}(\mathbb{R}^{\mathbb{N} \times \mathbb{N}}))$ satisfying

- i. $(\pi_{I \times I})_* \mu = \mu_{I \times I}$, where $\mu_{I \times I}$ is the law of $\text{IW}(\delta, \mathbf{U}_{I \times I})$ defined in Dawid (1981);
- ii. if $B = \{(\alpha_i, \beta_i)\}_{i=1}^n \subset \mathbb{N} \times \mathbb{N}$ and $\mathbf{g} = \{\alpha_i\}_{i=1}^n \cup \{\beta_i\}_{i=1}^n$, then $(\pi_B)_* \mu = \mu_B$, where $\mu_B = (\pi_{B \times \mathbb{R} \times \mathbb{R}})_* \mu_{\mathbb{R} \times \mathbb{R}}$.

Setting $\mu = \text{IWP}(\delta, \mathbf{U})$ so that $(\mathbf{U})_{ij} = u(i, j)$, we further have that if $\mathbf{Q} \sim \text{IWP}(\delta, \mathbf{U})$ and $\delta > 4$, the countably infinite array \mathbf{Q} is a positive semidefinite trace class operator on $\ell^2(\mathbb{N})$ almost surely.

Proof. Let $\mathbf{U}_{I \times I}$ be a matrix with the law $\mu_{I \times I}$. We will prove following Tao (2011, Theorem 2.4.3) as follows: (1) we verify the compatibility of μ_B for all finite $B \subset \mathbb{N} \times \mathbb{N}$. There are two successive cases we shall consider. Case 1: Suppose $I_2 \subset I_1$ are two finite subsets of \mathbb{N} , then $\mathbf{Q}_{I_2 \times I_2}$ is the sub-matrix of $\mathbf{Q}_{I_1 \times I_1}$ obtained by deleting the rows and columns with indices in $I_1 \setminus I_2$. If $\mathbf{Q}_{I_1 \times I_1}$ has law $\mu_{I_1 \times I_1} = \text{IW}(\delta, \mathbf{U}_{I_1 \times I_1})$, then $\mathbf{Q}_{I_2 \times I_2}$ has law $\text{IW}(\delta, \mathbf{U}_{I_2 \times I_2})$ due to the consistency property of the inverse-Wishart distribution (Dawid and Lauritzen, 1993, Lemma 7.4). Consequently, $(\pi_{I_2 \times I_2 \leftarrow I_1 \times I_1})_* \mu_{I_1 \times I_1} = \mu_{I_2 \times I_2}$. Case 2: Let $B_1 = \{(\alpha_i, \beta_i)\}_{i=1}^n \subset \mathbb{N} \times \mathbb{N}$ and suppose $B_2 = \{(\tilde{\alpha}_i, \tilde{\beta}_i)\}_{i=1}^m \subset B_1$. Set $\mathbf{g}_1 = \{\alpha_i\}_{i=1}^n \cup \{\beta_i\}_{i=1}^n$ and $\mathbf{g}_2 = \{\tilde{\alpha}_i\}_{i=1}^m \cup \{\tilde{\beta}_i\}_{i=1}^m$ so that $\mathbf{g}_2 \times \mathbf{g}_2 \subset \mathbf{g}_1 \times \mathbf{g}_1$. It is clear that $\pi_{B_2 \leftarrow B_1} \circ \pi_{B_1 \leftarrow \mathbf{g}_1} = \pi_{B_2 \leftarrow \mathbf{g}_1} = \pi_{B_2 \leftarrow \mathbf{g}_2 \times \mathbf{g}_2} \circ \pi_{\mathbf{g}_2 \times \mathbf{g}_2 \leftarrow \mathbf{g}_1 \times \mathbf{g}_1}$. Thus,

$$\begin{aligned} (\pi_{B_2 \leftarrow B_1})_* \mu_{B_1} &= (\pi_{B_2 \leftarrow B_1})_* (\pi_{B_1 \leftarrow \mathbf{g}_1})_* \mu_{\mathbf{g}_1 \times \mathbf{g}_1} = (\pi_{B_2 \leftarrow B_1} \circ \pi_{B_1 \leftarrow \mathbf{g}_1})_* \mu_{\mathbf{g}_1 \times \mathbf{g}_1} \\ &= (\pi_{B_2 \leftarrow \mathbf{g}_2 \times \mathbf{g}_2} \circ \pi_{\mathbf{g}_2 \times \mathbf{g}_2 \leftarrow \mathbf{g}_1 \times \mathbf{g}_1})_* \mu_{\mathbf{g}_1 \times \mathbf{g}_1} = (\pi_{B_2 \leftarrow \mathbf{g}_2 \times \mathbf{g}_2})_* (\pi_{\mathbf{g}_2 \times \mathbf{g}_2 \leftarrow \mathbf{g}_1 \times \mathbf{g}_1})_* \mu_{\mathbf{g}_1 \times \mathbf{g}_1} \\ &= (\pi_{B_2 \leftarrow \mathbf{g}_2 \times \mathbf{g}_2})_* \mu_{\mathbf{g}_2 \times \mathbf{g}_2} = \mu_{B_2}, \end{aligned}$$

where the second to last equality holds because of our demonstration in Case 1. (2) Second, we claim that the finite dimensional measure $\mu_{I \times I} = \text{IW}(\delta, \mathbf{U}_{I \times I})$ is an inner regular probability measure on the product σ -algebra $\mathcal{B}_{I \times I}$. We will show that $\mu_{I \times I}$ is a finite Borel measure on a Polish space, which then implies that $\mu_{I \times I}$ is regular, hence inner regular by Bauer (2001, Lemma 26.2). This is done through (a)-(c) as follows: (a) For finite I , $\mathbf{Q}_{I \times I}$ takes values in the space of symmetric and positive semidefinite matrices, denoted by $\Psi_{|I|}$ where $|I|$ denotes the number of elements in I . Since the subset of symmetric matrices is closed in $\mathbb{R}^{I \times I}$, it is Polish. Furthermore, the space of symmetric positive semidefinite matrices is an open convex cone in the space of symmetric matrices, hence it is Polish as well. Therefore the space $\Psi_{|I|}$ is Polish. (b) Since $\mu_{I \times I}$, the law of $\mathbf{Q}_{I \times I} \sim \text{IW}(\delta, \mathbf{U}_{I \times I})$, has an almost everywhere continuous density function, $\mu_{I \times I}$ is a measure defined by Lebesgue integration against an almost everywhere continuous function. Therefore $\mu_{I \times I}$ is Borel on $\Psi_{|I|}$. As $\Psi_{|I|} \subset \mathbb{R}^{I \times I}$, we may extend the measure $\mu_{I \times I}$ from $\Psi_{|I|}$ to $\mathbb{R}^{I \times I}$ via the Carathéodory theorem (Tao, 2011, Theorem 1.7.3). In particular, define $\mu_{I \times I}(A) = \mu_{I \times I}(A \cap \Psi_{|I|})$ for $A \in \mathcal{B}(\mathbb{R}^{I \times I})$. With extension, $\mu_{I \times I}$ is Borel on $\mathbb{R}^{I \times I}$, and the σ -algebra associated is $\mathcal{B}(\mathbb{R}^{I \times I}) = \mathcal{B}_{I \times I} = \prod_{(\alpha, \beta) \in I \times I} \mathcal{B}_{(\alpha, \beta)}$. (c) The measure $\mu_{I \times I}$ is certainly finite since it is a probability measure.

The compatibility and regularity conditions in (1) and (2) ensure that the Kolmogorov extension theorem holds. Therefore there exists a unique probability measure μ on the product σ -algebra $\mathcal{B}(\mathbb{R}^{\mathbb{N} \times \mathbb{N}})$ that satisfies (i) and (ii).

We now prove that if $\mathbf{Q} \sim \text{IWP}(\delta, \mathbf{U})$, then the countably infinite array \mathbf{Q} is a well-defined positive semidefinite trace class operator on $\ell^2(\mathbb{N})$ almost surely. First, we note that the spectral theorem ensures the existence of an orthonormal basis of $\ell^2(\mathbb{N})$ that diagonalizes \mathbf{U} . Thus, without loss of generality, we may assume that \mathbf{Q} is drawn from $\text{IWP}(\delta, \mathbf{U})$ where \mathbf{U} is a diagonal positive semidefinite trace class operator on $\ell^2(\mathbb{N})$.

First, we show each row of $\mathbf{Q}\mathbf{x}$ is finite almost surely hence is well-defined for all $\mathbf{x} \in \ell^2(\mathbb{N})$. It is sufficient to show that $E[|(\mathbf{Q}\mathbf{x})_i|] < \infty$. We note that for arbitrary $i \neq j$,

$\begin{pmatrix} q_{ii} & q_{ij} \\ q_{ij} & q_{jj} \end{pmatrix} \sim \text{IW}\left(\delta, \begin{pmatrix} u_{ii} & 0 \\ 0 & u_{jj} \end{pmatrix}\right)$ and hence using the moments of finite dimensional inverse-Wishart, $E(q_{ii}^2) = u_{ii}^2(\delta - 2)^{-1}(\delta - 4)^{-1}$, $E(q_{ij}^2) = u_{ii}u_{jj}(\delta - 1)^{-1}(\delta - 2)^{-1}(\delta - 4)^{-1}$, for $\delta > 4$. By Tonelli's theorem, we have that $E\sum_j q_{ij}^2 = \sum_j E q_{ij}^2 \leq C\sum_j u_{ii}u_{jj} = C u_{ii} \sum_j u_{jj}$, where C is the maximum of the above constants. Thus

$$E[|\langle \mathbf{Q}\mathbf{x}, \mathbf{x} \rangle|] \leq \|\mathbf{x}\| \sqrt{E\sum_j q_{ij}^2} < \infty.$$

Because there are only countably many rows, we have that $\mathbf{Q}\mathbf{x}$ is finite almost surely for all rows simultaneously. Consequently, we have that $\mathbf{Q}\mathbf{x}$ is well-defined for all $\mathbf{x} \in \ell^2(\mathbb{N})$. Now we show that $\mathbf{Q}\mathbf{x} \in \ell^2(\mathbb{N})$ almost surely. By similar considerations, let $\mathbf{q}_i = (\mathbf{Q}\mathbf{x})_i$, then $E\|\sum_i \mathbf{q}_i\|^2 \leq C\sum_i u_{ii}^2 < \infty$ and $\|\mathbf{Q}\mathbf{x}\|^2 \leq C\|\mathbf{x}\|^2 \sum_i \|q_i\|^2$; this implies that $\|\mathbf{Q}\mathbf{x}\| < \infty$ almost surely hence $\mathbf{Q}\mathbf{x} \in \ell^2(\mathbb{N})$ almost surely, and it also implies that the operator norm $\|\mathbf{Q}\|_{\text{op}}$ is finite almost surely.

By construction, we must have that \mathbf{Q} is positive semidefinite almost surely since $\langle \mathbf{Q}\mathbf{x}, \mathbf{x} \rangle = \lim_{\eta \rightarrow \infty} \langle \mathbf{Q}_\eta \mathbf{x}, \mathbf{x} \rangle \geq 0$, where \mathbf{Q}_η is the restriction of \mathbf{Q} to its n by n leading principal minor. Finally, \mathbf{Q} is trace class almost surely since $E[|\text{tr}(\mathbf{Q})|] = \sum_i E(q_{ii}) = (\delta - 2)^{-1} \sum_i u_{ii} < \infty$. ■

Appendix E. Proof of Theorem 1

Proof. Based on Lemma 2, we can define a sequence of inverse-Wishart process prior for \mathcal{Q}_C , denoted by $\mathcal{Q}_C \sim \text{IWP}(\delta, \mathcal{U}_C)$, $C \in \mathcal{C}$. These sequences are pairwise consistent due to the consistency of inverse-Wishart processes and the fact that \mathcal{U}_C is a common collection of kernels. Therefore, we can construct a unique hyper Markov law for \mathcal{Q}_C following procedure (12) - (13) of Dawid and Lauritzen (1993). And Theorem 3.9 of Dawid and Lauritzen (1993) guarantees that the constructed hyper Markov law is unique. ■

Appendix F. Proof of Proposition 2

Proof. Note that an operator drawn from a hyper-inverse-Wishart process with the parameter \mathcal{U} satisfies $\text{rank}(u_{ij}) < \infty$ for $i, j \in V$ will have finite-rank almost surely. This follows by noting that if $\mathcal{Q} \sim \text{HIWP}(\delta, \mathcal{U})$ and \mathcal{W} is a fixed unitary transformation on ℓ^2 , then $\mathcal{W}^T \mathcal{Q} \mathcal{W} \sim \text{HIWP}(\delta, \mathcal{W}^T \mathcal{U} \mathcal{W})$. Thus, choosing \mathcal{W} so that the block representation $\mathcal{W}^T \mathcal{U} \mathcal{W} = \begin{pmatrix} U & 0 \\ 0 & 0 \end{pmatrix}$ holds (here, U is a finite matrix and 0's represent infinite arrays of zeros), we see that the block representation $\mathcal{W}^T \mathcal{Q} \mathcal{W} = \begin{pmatrix} Q & 0 \\ 0 & 0 \end{pmatrix}$ holds almost surely, and that $Q \sim \text{IW}(\delta, U)$. Consequently, we have reduced to the finite-dimensional setting where the result is well-known. ■

Appendix G. Proof of Theorem 2

Proof. By the result of Proposition 1, the HIWPG prior is a strong hyper Markov law. So by Corollary 5.5 of Dawid and Lauritzen (1993), the posterior law of \mathcal{Q}_C is the unique

hyper Markov law specified by the marginal posterior laws at each clique. In other words, we just need to find the posterior law for the model: $\mathbf{c}_{i,C} \sim \text{DMGP}(\mathbf{c}_{0,C}; \mathcal{Q}_C)$ with prior $\mathcal{Q}_C \sim \text{IWP}(\delta, \mathcal{U}_C)$ for each \mathcal{Q}_C , and use them to construct the posterior law of \mathcal{Q}_C following (12) - (13) of Dawid and Lauritzen (1993). As in the last proof, choosing an appropriate transformation reduces this to the finite-dimensional case which is well-known. Finally, by Proposition 5.6 of Dawid and Lauritzen (1993), the marginal distribution of $\{c_j\}$ given $C, \mathbf{c}_0, \delta, \mathcal{U}_C$ is again Markov over G . ■

Online Appendix

The online appendix contains more detailed derivations, discussions, and simulation results.

References

- A. Anandkumar, V. Y. F. Tan, F. Huang, and A. S. Willsky. High-dimensional structure estimation in Ising models: local separation criterion. *Ann. Statist.*, 40(3):1346–1375, 06 2012.
- H. Bauer. *Measure and Integration Theory*. De Gruyter Studies in Mathematics. W. de Gruyter, 2001.
- J. Bilmes. Dynamic graphical models. *IEEE Signal Processing Magazine*, 27(6):29–42, 2010.
- D. Brandeis and D. Lehmann. Event-related potentials of the brain and cognitive processes: approaches and applications. *Neuropsychologia*, pages 151–168, 1986.
- S. L. Bressler. Event-related potentials. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 412–415. MIT Press, Cambridge MA, 2002.
- T. Cai, W. Lin, and X. Luo. A constrained L_1 minimization approach to sparse precision matrix estimation. *J. Amer. Statist. Assoc.*, 106(494):594–607, 2011.
- C. M. Carvalho and J. G. Scott. Objective Bayesian model selection in Gaussian graphical models. *Biometrika*, 96(3):497–512, 2009.
- C. M. Carvalho and M. West. Dynamic matrix-variate graphical models. *Bayesian Anal.*, 2(1):69–98, 2007.
- J. T. Chang and D. Pollard. Conditioning as disintegration. *Statistica Neerlandica*, 51(3): 287–317, 1997.
- H. Dammé III. Fast search for dirichlet process mixture models. In *Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2007.
- A. P. Dawid. Some matrix-variate distribution theory: Notational considerations and a Bayesian application. *Biometrika*, 68(1):265–274, 1981.
- A. P. Dawid and S. L. Lauritzen. Hyper Markov laws in the statistical analysis of decomposable graphical models. *Ann. Statist.*, 21(3):1272–1317, 1993.

- C. De Boor. *A Practical Guide to Splines*. Applied Mathematical Sciences. Springer, Berlin, 2001.
- A. P. Dempster. Covariance selection. *Biometrics*, 28:157–175, 1972.
- P. R. Finn and A. Justus. Reduced EEG alpha power in the male and female offspring of alcoholics. *Alcohol. Clin. Exp. Res.*, 23:256–262, 1999.
- J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.
- P. Giudici. Learning in graphical Gaussian models. *Bayesian Statistics 5*, pages 621–628, 1996.
- P. Giudici and P. J. Green. Decomposable graphical Gaussian model determination. *Biometrika*, 86(4):785–801, 1999.
- Y. Gnan and S. M. Krone. Small-world MCMC and convergence to multi-modal distributions: From slow mixing to fast mixing. *Ann. Appl. Probab.*, 17:284–304, 2007.
- Y. Guan, R. Fleissner, P. Joyce, and S. M. Krone. Markov chain Monte Carlo in small worlds. *Stat. Comput.*, 16:193–202, 2006.
- E. P. Hayden, R. E. Wiegand, E. T. Meyer, L. O. Bauer, S. J. O’Connor, J. I. Nurnberger, D. B. Chorlian, B. Porjesz, and H. Begleiter. Patterns of regional brain activity in alcohol-dependent subjects. *Alcohol. Clin. Exp. Res.*, 30(12):1986 – 1991, 2006.
- H. Höfling and R. Tibshirani. Estimation of sparse binary pairwise Markov networks using pseudo-likelihoods. *J. Mach. Learn. Res.*, 10:883–906, 2009.
- A. Jalali, C. C. Johnson, and P. K. Ravikumar. On learning discrete graphical models using greedy methods. In J. Shawe-taylor, R.s. Zemel, P. Bartlett, F.c.n. Pereira, and K.q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1935–1943. 2011.
- M. Jansen and P. J. Ooninx. *Second Generation Wavelets and Applications*. Springer-Verlag, London, 2015.
- B. Jones, C. Carvalho, A. Dobra, C. Hans, C. Carter, and M. West. Experiments in stochastic computation for high-dimensional graphical models. *Statist. Sci.*, 20(4):388–400, 2005.
- Gennady G. Knyazev. Motivation, emotion, and their inhibitory control mirrored in brain oscillations. *Neurosci. Biobehav. Rev.*, 31(3):377 – 395, 2007.
- M. Kolar and E. Xing. On time varying undirected graphs. *J. Mach. Learn. Res.*, 15: 407–415, 2011.
- C. Lam and J. Fan. Sparsistency and rates of convergence in large covariance matrix estimation. *Ann. Statist.*, 37(6B):4254–4278, 12 2009.
- S. L. Lauritzen. *Graphical Models*. Clarendon Press, Oxford, 1996.
- E. Lei, F. Yao, N. Heckman, and K. Meyer. Functional data model for genetically related individuals with application to cow growth. *J. Comp. and Graph. Stat.*, 2014.
- S. Li, M. Gu, C. J. Wu, and J. Xia. New efficient and robust HSS Cholesky factorization of SPD matrices. *SIAM J. Matrix Anal. Appl.*, pages 886–904, 2012.
- P.-L. Loh and M. J. Wainwright. Structure estimation for discrete graphical models: Generalized covariance matrices and their inverses. *Ann. Statist.*, 41(6):3022–3049, 12 2013.
- P. Ma, C. I. Castillo-Davis, W. Zhong, and J. S. Liu. A data-driven clustering method for time course gene expression data. *Nucleic Acids Res.*, 34(4):1261–1269, 2006.
- R. Mazumder and T. Hastie. The graphical lasso: New insights and alternatives. *Electron. J. Statist.*, 6:2125–2149, 2012a.
- R. Mazumder and T. Hastie. Exact covariance thresholding into connected components for large-scale graphical lasso. *J. Mach. Learn. Res.*, 13:781–794, 2012b.
- N. Meinshausen and P. Bühlmann. High dimensional graphs and variable selection with the Lasso. *Ann. Statist.*, 34(3):1436–1462, 2006.
- H. G. Müller and F. Yao. Functional additive models. *J. Am. Statist. Assoc.*, 103:1534–1544, 2008.
- B. Porjesz, M. Rangaswamy, C. Kamarajan, K. A. Jones, A. Padmanabhapillai, and H. Begleiter. The utility of neurophysiological markers in the study of alcoholism. *Clin. Neurophysiol.*, 116(5):993 – 1018, 2005.
- G. D. Prato. *An Introduction to Infinite-Dimensional Analysis*. Springer, New York, 2006.
- X. Qiao, C. James, and J. Lv. Functional graphical models. Technical report, University of Southern California, 2015.
- J. O. Ramsay and B. W. Silverman. *Functional Data Analysis, Section Edition*. Springer, New York, 2005.
- P. Ravikumar, M. J. Wainwright, and J. D. Lafferty. High-dimensional Ising model selection using ℓ_1 -regularized logistic regression. *Ann. Statist.*, 38(3):1287–1319, 06 2010.
- J. A. Rice and B. W. Silverman. Estimating the mean and covariance structure nonparametrically when the data are curves. *J. R. Stat. Soc., Series B*, 53:233–243, 1991.
- A. Roverato. Hyper inverse Wishart distribution for non-decomposable graphs and its application to Bayesian inference for Gaussian graphical models. *Scand. J. Stat.*, 29: 391–411, 2002.
- A. Saltelli, K. Chan, and E. M. Scott, editors. *Sensitivity Analysis*. John Wiley & Sons, Ltd., New York, 2000.

- J. G. Scott and C. M. Carvalho. Feature-inclusion stochastic search for Gaussian graphical models. *J. Comput. Graph. Statist.*, 17(4):790–808, 2008.
- K. J. Sher, E.R. Grekin, and N. A. Williams. The development of alcohol use disorders. *Annu. Rev. Clin. Psychol.*, 1:493–523, 2005.
- T. Tao. *An Introduction to Measure Theory*. Graduate Studies in Mathematics. Amer. Math. Soc., 2011.
- H. Wang and M. West. Bayesian analysis of matrix normal graphical models. *Biometrika*, 96(4):821–834, 2009.
- D. M. Witten, J. H. Friedman, and N. Simon. New insights and faster computations for the graphical Lasso. *J. Comp. Graph. Stat.*, 20(4):892–900, 2011.
- E. Yang, G. Allen, Z. Lin, and P. K. Ravikumar. Graphical models via generalized linear models. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1358–1366. Curran Associates, Inc., 2012.
- F. Yao, H. G. Müller, and J. L. Wang. Functional data analysis for sparse longitudinal data. *J. Am. Statist. Assoc.*, 100:577–590, 2005.
- M. Yuan and Y. Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94:19–35, 2007.
- S. Zhou, J. D. Lafferty, and L. A. Wasserman. Time varying undirected graphs. *Mach. Learn.*, 80:295–319, 2010.

Neural Autoregressive Distribution Estimation

Benigno Urias
Google DeepMind
 London, UK

BENIGNO.URIA@GMAIL.COM

Marc-Alexandre Côté
Department of Computer Science
Université de Sherbrooke
 Sherbrooke, J1K 2R1, QC, Canada

MARC-ALEXANDRE.COTE@USHERBROOKE.CA

Karol Gregor
Google DeepMind
 London, UK

KAROL.GREGOR@GMAIL.COM

Iain Murray
School of Informatics
University of Edinburgh
 Edinburgh EH8 9AB, UK

I.MURRAY@ED.AC.UK

Hugo Larochelle
Twitter

HLAROCHELLE@TWITTER.COM

141 Portland St, Floor 6
 Cambridge MA 02139, USA

Editor: Russian Salakhutdinov

Abstract

We present Neural Autoregressive Distribution Estimation (NADE) models, which are neural network architectures applied to the problem of unsupervised distribution and density estimation. They leverage the probability product rule and a weight sharing scheme inspired from restricted Boltzmann machines, to yield an estimator that is both tractable and has good generalization performance. We discuss how they achieve competitive performance in modeling both binary and real-valued observations. We also present how deep NADE models can be trained to be agnostic to the ordering of input dimensions used by the autoregressive product rule decomposition. Finally, we also show how to exploit the topological structure of pixels in images using a deep convolutional architecture for NADE.

Keywords: deep learning, neural networks, density modeling, unsupervised learning

1. Introduction

Distribution estimation is one of the most general problems addressed by machine learning. From a good and flexible distribution estimator, in principle it is possible to solve a variety of types of inference problem, such as classification, regression, missing value imputation, and many other predictive tasks.

Currently, one of the most common forms of distribution estimation is based on directed graphical models. In general these models describe the data generation process as sampling

a latent state \mathbf{h} from some prior $p(\mathbf{h})$, followed by sampling the observed data \mathbf{x} from some conditional $p(\mathbf{x} | \mathbf{h})$. Unfortunately, this approach quickly becomes intractable and requires approximations when the latent state \mathbf{h} increases in complexity. Specifically, computing the marginal probability of the data, $p(\mathbf{x}) = \sum_{\mathbf{h}} p(\mathbf{x} | \mathbf{h}) p(\mathbf{h})$, is only tractable under fairly constraining assumptions on $p(\mathbf{x} | \mathbf{h})$ and $p(\mathbf{h})$.

Another popular approach, based on undirected graphical models, gives probabilities of the form $p(\mathbf{x}) = \exp\{\phi(\mathbf{x})\} / Z$, where ϕ is a tractable function and Z is a normalizing constant. A popular choice for such a model is the restricted Boltzmann machine (RBM), which substantially out-performs mixture models on a variety of binary data sets (Salakhutdinov and Murray, 2008). Unfortunately, we often cannot compute probabilities $p(\mathbf{x})$ exactly in undirected models either, due to the normalizing constant Z .

In this paper, we advocate a third approach to distribution estimation, based on autoregressive models and feed-forward neural networks. We refer to our particular approach as Neural Autoregressive Distribution Estimation (NADE). Its main distinguishing property is that computing $p(\mathbf{x})$ under a NADE model is tractable and can be computed efficiently, given an arbitrary ordering of the dimensions of \mathbf{x} .

The NADE framework was first introduced for binary variables by Larochelle and Murray (2011), and concurrent work by Gregor and LeCun (2011). The framework was then generalized to real-valued observations (Urias et al., 2013), and to versions based on deep neural networks that can model the observations in any order (Urias et al., 2014). This paper pulls together an extended treatment of these papers, with more experimental results, including some by Urias (2015). We also report new work on modeling 2D images by incorporating convolutional neural networks into the NADE framework. For each type of data, we're able to reach competitive results, compared to popular directed and undirected graphical model alternatives.

2. NADE

We consider the problem of modeling the distribution $p(\mathbf{x})$ of input vector observations \mathbf{x} . For now, we will assume that the dimensions of \mathbf{x} are binary, that is $x_d \in \{0, 1\} \forall d$. The model generalizes to other data types, which is explored later (Section 3) and in other work (Section 8).

NADE begins with the observation that any D -dimensional distribution $p(\mathbf{x})$ can be factored into a product of one-dimensional distributions, in any order o (a permutation of the integers $1, \dots, D$):

$$p(\mathbf{x}) = \prod_{d=1}^D p(x_{o_d} | \mathbf{x}_{o_{<d}}). \quad (1)$$

Here $o_{<d}$ contains the first $d - 1$ dimensions in ordering o and $\mathbf{x}_{o_{<d}}$ is the corresponding subvector for these dimensions. Thus, one can define an 'autoregressive' generative model of the data simply by specifying a parameterization of all D conditionals $p(x_{o_d} | \mathbf{x}_{o_{<d}})$.

Frey et al. (1996) followed this approach and proposed using simple (log-)linear logistic regression models for these conditionals. This choice yields surprisingly competitive results, but are not competitive with non-linear models such as an RBM. Bengio and Bengio (2000) proposed a more flexible approach, with a single-layer feed-forward neural network for each

conditional. Moreover, they allowed connections between the output of each network and the hidden layer of networks for the conditionals appearing earlier in the autoregressive ordering. Using neural networks led to some improvements in modeling performance, though at the cost of a really large model for very high-dimensional data.

In NADE, we also model each conditional using a feed-forward neural network. Specifically, each conditional $p(x_{o_d} | \mathbf{x}_{<d})$ is parameterized as follows:

$$p(x_{o_d} = 1 | \mathbf{x}_{o_{<d}}) = \text{sigm}(\mathbf{V}_{o_d} \mathbf{h}_{o_d} + b_{o_d}) \quad (2)$$

$$\mathbf{h}_d = \text{sigm}(\mathbf{W}_{\cdot, o_{<d}} \mathbf{x}_{o_{<d}} + \mathbf{c}), \quad (3)$$

where $\text{sigm}(a) = 1/(1 + e^{-a})$ is the logistic sigmoid, and with H as the number of hidden units, $\mathbf{V} \in \mathbb{R}^{D \times H}$, $\mathbf{b} \in \mathbb{R}^D$, $\mathbf{W} \in \mathbb{R}^{H \times D}$, $\mathbf{c} \in \mathbb{R}^H$ are the parameters of the NADE model.

The hidden layer matrix \mathbf{W} and bias \mathbf{c} are shared by each hidden layer \mathbf{h}_d (which are all of the same size). This parameter sharing scheme (illustrated in Figure 1) means that NADE has $O(HD)$ parameters, rather than $O(HD^2)$ required if the neural networks were separate. Limiting the number of parameters can reduce the risk of over-fitting. Another advantage is that all D hidden layers \mathbf{h}_d can be computed in $O(HD)$ time instead of $O(HD^2)$. Denoting the pre-activation of the d^{th} hidden layer as $\mathbf{a}_d = \mathbf{W}_{\cdot, o_{<d}} \mathbf{x}_{o_{<d}} + \mathbf{c}$, this complexity is achieved by using the recurrence

$$\mathbf{h}_1 = \text{sigm}(\mathbf{a}_1), \quad \text{where } \mathbf{a}_1 = \mathbf{c} \quad (4)$$

$$\mathbf{h}_d = \text{sigm}(\mathbf{a}_d), \quad \text{where } \mathbf{a}_d = \mathbf{W}_{\cdot, o_{<d}} \mathbf{x}_{o_{<d}} + \mathbf{c} = \mathbf{W}_{\cdot, o_{d-1}} \mathbf{x}_{o_{d-1}} + \mathbf{a}_{d-1} \quad (5)$$

for $d \in \{2, \dots, D\}$,

where Equation 5 given vector \mathbf{a}_{d-1} can be computed in $O(H)$. Moreover, the computation of Equation 2 given \mathbf{h} is also $O(H)$. Thus, computing $p(\mathbf{x})$ from D conditional distributions (Equation 1) costs $O(HD)$ for NADE. This complexity is comparable to that of regular feed-forward neural network models.

NADE can be trained by maximum likelihood, or equivalently by minimizing the average negative log-likelihood,

$$\frac{1}{N} \sum_{n=1}^N -\log p(\mathbf{x}^{(n)}) = \frac{1}{N} \sum_{n=1}^N \sum_{d=1}^D -\log p(x_{o_d}^{(n)} | \mathbf{x}_{o_{<d}}^{(n)}), \quad (6)$$

usually by stochastic (minibatch) gradient descent. As probabilities $p(\mathbf{x})$ cost $O(HD)$, gradients of the negative log-probability of training examples can also be computed in $O(HD)$. Algorithm 1 describes the computation of both $p(\mathbf{x})$ and the gradients of $-\log p(\mathbf{x})$ with respect to NADE's parameters.

2.1 Relationship with the RBM

The proposed weight-tying for NADE isn't simply motivated by computational reasons. It also reflects the computations of approximation inference in the RBM.

Denoting the energy function and distribution under an RBM as

$$E(\mathbf{x}, \mathbf{h}) = -\mathbf{h}^T \mathbf{W} \mathbf{x} - \mathbf{b}^T \mathbf{x} - \mathbf{c}^T \mathbf{h} \quad (7)$$

$$p(\mathbf{x}, \mathbf{h}) = \exp\{-E(\mathbf{x}, \mathbf{h})\} / Z, \quad (8)$$

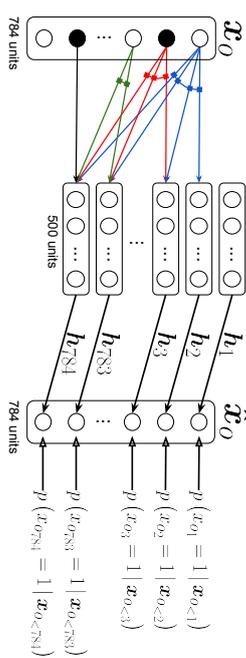


Figure 1: Illustration of a NADE model. In this example, in the input layer, units with value 0 are shown in black while units with value 1 are shown in white. The dashed border represents a layer pre-activation. The outputs $\hat{\mathbf{x}}_0$ give predictive probabilities for each dimension of a vector \mathbf{x}_0 , given elements earlier in some ordering. There is no path of connections between an output and the value being predicted, or elements of \mathbf{x}_0 later in the ordering. Arrows connected together correspond to connections with shared (tied) parameters.

computing all conditionals

$$p(\mathbf{x}_{o_d} | \mathbf{x}_{o_{<d}}) = \sum_{\mathbf{x}_{o_{>d}} \in \{0,1\}^{D-d}} \sum_{\mathbf{h} \in \{0,1\}^H} \exp\{-E(\mathbf{x}, \mathbf{h})\} / Z(\mathbf{x}_{o_{<d}}) \quad (9)$$

$$Z(\mathbf{x}_{o_{<d}}) = \sum_{\mathbf{x}_{o_{>d}} \in \{0,1\}^{D-d+1}} \sum_{\mathbf{h} \in \{0,1\}^H} \exp\{-E(\mathbf{x}, \mathbf{h})\} \quad (10)$$

is intractable. However, these could be approximated using mean-field variational inference. Specifically, consider the conditional over x_{o_d} , $\mathbf{x}_{o_{>d}}$ and \mathbf{h} instead:

$$p(x_{o_d}, \mathbf{x}_{o_{>d}}, \mathbf{h} | \mathbf{x}_{o_{<d}}) = \exp\{-E(\mathbf{x}, \mathbf{h})\} / Z(\mathbf{x}_{o_{<d}}). \quad (11)$$

A mean-field approach could first approximate this conditional with a factorized distribution

$$q(x_{o_d}, \mathbf{x}_{o_{>d}}, \mathbf{h} | \mathbf{x}_{o_{<d}}) = \mu_d^i(d)^{x_{o_d}} (1 - \mu_d^i(d))^{1-x_{o_d}} \prod_{j>d} \mu_j(d)^{x_{o_j}} (1 - \mu_j(d))^{1-x_{o_j}} \prod_k \tau_k(d)^{h_k} (1 - \tau_k(d))^{1-h_k}, \quad (12)$$

where $\mu_j(d)$ is the marginal probability of x_{o_j} being equal to 1, given $\mathbf{x}_{o_{<d}}$. Similarly, $\tau_k(d)$ is the marginal for hidden variable h_k . The dependence on d comes from conditioning on $\mathbf{x}_{o_{<d}}$, that is on the first $d-1$ dimensions of \mathbf{x} in ordering o .

For some d , a mean-field approximation is obtained by finding the parameters $\mu_j(d)$ for $j \in \{d, \dots, D\}$ and $\tau_k(d)$ for $k \in \{1, \dots, H\}$ which minimize the KL divergence between $q(x_{o_d}, \mathbf{x}_{o_{>d}}, \mathbf{h} | \mathbf{x}_{o_{<d}})$ and $p(x_{o_d}, \mathbf{x}_{o_{>d}}, \mathbf{h} | \mathbf{x}_{o_{<d}})$. This is usually done by finding message

Algorithm 1 Computation of $p(\mathbf{x})$ and learning gradients for NADE.

Input: training observation vector \mathbf{x} and ordering o of the input dimensions.

Output: $p(\mathbf{x})$ and gradients of $-\log p(\mathbf{x})$ on parameters.

```

# Computing  $p(\mathbf{x})$ 
 $\mathbf{a}_1 \leftarrow \mathbf{c}$ 
 $p(\mathbf{x}) \leftarrow 1$ 
for  $d$  from 1 to  $D$  do
   $\mathbf{h}_d \leftarrow \text{sigm}(\mathbf{a}_d)$ 
   $p(x_{o_d} = 1 | \mathbf{x}_{o_{<d}}) \leftarrow \text{sigm}(\mathbf{V}_{o_d} \mathbf{h}_d + b_{o_d})$ 
   $p(\mathbf{x}) \leftarrow p(\mathbf{x}) (p(x_{o_d} = 1 | \mathbf{x}_{o_{<d}}))^{x_{o_d}} + (1 - p(x_{o_d} = 1 | \mathbf{x}_{o_{<d}}))^{1-x_{o_d}}$ 
   $\mathbf{a}_{d+1} \leftarrow \mathbf{a}_d + \mathbf{W}_{\cdot, o_d} x_{o_d}$ 
end for

```

```

# Computing gradients of  $-\log p(\mathbf{x})$ 
 $\delta \mathbf{c} \leftarrow 0$ 
 $\delta \mathbf{c} \leftarrow 0$ 
for  $d$  from  $D$  to 1 do
   $\delta b_{o_d} \leftarrow (p(x_{o_d} = 1 | \mathbf{x}_{o_{<d}}) - x_{o_d})$ 
   $\delta \mathbf{V}_{o_d} \leftarrow (p(x_{o_d} = 1 | \mathbf{x}_{o_{<d}}) - x_{o_d}) \mathbf{h}_d^\top$ 
   $\delta \mathbf{h}_d \leftarrow (p(x_{o_d} = 1 | \mathbf{x}_{o_{<d}}) - x_{o_d}) \mathbf{V}_{o_d}$ 
   $\delta \mathbf{c} \leftarrow \delta \mathbf{c} + \delta \mathbf{h}_d \odot \mathbf{h}_d \odot (1 - \mathbf{h}_d)$ 
   $\delta \mathbf{W}_{\cdot, o_d} \leftarrow \delta \mathbf{a}_d x_{o_d}$ 
   $\delta \mathbf{a}_{d-1} \leftarrow \delta \mathbf{a}_d + \delta \mathbf{h}_d \odot \mathbf{h}_d \odot (1 - \mathbf{h}_d)$ 
end for
return  $p(\mathbf{x})$ ,  $\delta \mathbf{b}$ ,  $\delta \mathbf{V}$ ,  $\delta \mathbf{c}$ ,  $\delta \mathbf{W}$ 

```

passing updates that each set the derivatives of the KL divergence to 0 for some of the parameters of $q(x_{o_d}, \mathbf{x}_{o_{>d}}, \mathbf{h} | \mathbf{x}_{o_{<d}})$ given others.

For some d , let us fix $\mu_j(d) = x_{o_d}$ for $j < d$, leaving only $\mu_j(d)$ for $j > d$ to be found. The KL-divergence develops as follows:

$$\begin{aligned}
& \text{KL}(q(x_{o_d}, \mathbf{x}_{o_{>d}}, \mathbf{h} | \mathbf{x}_{o_{<d}}) \| p(x_{o_d}, \mathbf{x}_{o_{>d}}, \mathbf{h} | \mathbf{x}_{o_{<d}})) \\
&= - \sum_{x_{o_d}, \mathbf{x}_{o_{>d}}, \mathbf{h}} q(x_{o_d}, \mathbf{x}_{o_{>d}}, \mathbf{h} | \mathbf{x}_{o_{<d}}) \log p(x_{o_d}, \mathbf{x}_{o_{>d}}, \mathbf{h} | \mathbf{x}_{o_{<d}}) \\
&+ \sum_{x_{o_d}, \mathbf{x}_{o_{>d}}, \mathbf{h}} q(x_{o_d}, \mathbf{x}_{o_{>d}}, \mathbf{h} | \mathbf{x}_{o_{<d}}) \log q(x_{o_d}, \mathbf{x}_{o_{>d}}, \mathbf{h} | \mathbf{x}_{o_{<d}}) \\
&= \log Z(\mathbf{x}_{o_{<d}}) - \sum_j \sum_k \tau_k(d) W_{k, o_j} \mu_j(d) - \sum_j b_{o_j} \mu_j(d) - \sum_k c_k \tau_k(d) \\
&+ \sum_{j \geq d} (\mu_j(d) \log \mu_j(d) + (1 - \mu_j(d)) \log(1 - \mu_j(d))) \\
&+ \sum_k (\tau_k(d) \log \tau_k(d) + (1 - \tau_k(d)) \log(1 - \tau_k(d))).
\end{aligned}$$

Then, we can take the derivative with respect to $\tau_k(d)$ and set it to 0, to obtain:

$$\begin{aligned}
0 &= \frac{\partial \text{KL}(q(x_{o_d}, \mathbf{x}_{o_{>d}}, \mathbf{h} | \mathbf{x}_{o_{<d}}) \| p(x_{o_d}, \mathbf{x}_{o_{>d}}, \mathbf{h} | \mathbf{x}_{o_{<d}}))}{\partial \tau_k(d)} \\
0 &= -c_k - \sum_j W_{k, o_j} \mu_j(d) + \log \left(\frac{\tau_k(d)}{1 - \tau_k(d)} \right) \\
\frac{\tau_k(d)}{1 - \tau_k(d)} &= \exp \left\{ c_k + \sum_j W_{k, o_j} \mu_j(d) \right\} \\
\tau_k(d) &= \frac{\exp \left\{ c_k + \sum_j W_{k, o_j} \mu_j(d) \right\}}{1 + \exp \left\{ c_k + \sum_j W_{k, o_j} \mu_j(d) \right\}} \\
\tau_k(d) &= \text{sigm} \left(c_k + \sum_{j \geq d} W_{k, o_j} \mu_j(d) + \sum_{j < d} W_{k, o_j} x_{o_j} \right).
\end{aligned} \tag{13}$$

where in the last step we have used the fact that $\mu_j(d) = x_{o_j}$ for $j < d$. Equation 14 would correspond to the message passing updates of the hidden unit marginals $\tau_k(d)$ given the marginals of input $\mu_j(d)$.

Similarly, we can set the derivative with respect to $\mu_j(d)$ for $j \geq d$ to 0 and obtain:

$$\begin{aligned}
0 &= \frac{\partial \text{KL}(q(x_{o_d}, \mathbf{x}_{o_{>d}}, \mathbf{h} | \mathbf{x}_{o_{<d}}) \| p(x_{o_d}, \mathbf{x}_{o_{>d}}, \mathbf{h} | \mathbf{x}_{o_{<d}}))}{\partial \mu_j(d)} \\
0 &= -b_{o_j} - \sum_k \tau_k(d) W_{k, o_j} + \log \left(\frac{\mu_j(d)}{1 - \mu_j(d)} \right) \\
\frac{\mu_j(d)}{1 - \mu_j(d)} &= \exp \left\{ b_{o_j} + \sum_k \tau_k(d) W_{k, o_j} \right\} \\
\mu_j(d) &= \frac{\exp \left\{ b_{o_j} + \sum_k \tau_k(d) W_{k, o_j} \right\}}{1 + \exp \left\{ b_{o_j} + \sum_k \tau_k(d) W_{k, o_j} \right\}} \\
\mu_j(d) &= \text{sigm} \left(b_{o_j} + \sum_k \tau_k(d) W_{k, o_j} \right).
\end{aligned} \tag{15}$$

Equation 15 would correspond to the message passing updates of the input marginals $\mu_j(d)$ given the hidden layer marginals $\tau_k(d)$. The complete mean-field algorithm would thus alternate between applying the updates of Equations 14 and 15, right to left.

We now notice that Equation 14 corresponds to NADE's hidden layer computation (Equation 3) where $\mu_j(d) = 0 \forall j \geq d$. Also, Equation 15 corresponds to NADE's output layer computation (Equation 2) where $j = d$, $\tau_k(d) = h_{d,k}$ and $\mathbf{W}^\top = \mathbf{V}$. Thus, in short, NADE's forward pass is equivalent to applying a single pass of mean-field inference to approximate all the conditionals $p(\mathbf{x}_{o_d} | \mathbf{x}_{o_{<d}})$ of an RBM, where initially $\mu_j(d) = 0$ and where a separate matrix \mathbf{V} is used for the hidden-to-input messages. A generalization of NADE based on this connection to mean field inference has been further explored by Raiko et al. (2014).

3. NADE for Non-Binary Observations

So far we have only considered the case of binary observations x_i . However, the framework of NADE naturally extends to distributions over other types of observations.

In the next section, we discuss the case of real-valued observations, which is one of the most general cases of non-binary observations and provides an illustrative example of the technical considerations one faces when extending NADE to new observations.

3.1 RNADE: Real-Valued NADE

A NADE model for real-valued data could be obtained by applying the derivations shown in Section 2.1 to the Gaussian-RBM (Welling et al., 2005). The resulting neural network would output the mean of a Gaussian with fixed variance for each of the conditionals in Equation 1. Such a model is not competitive with mixture models, for example on perceptual data sets (Uria, 2015). However, we can explore alternative models by making the neural network for each conditional distribution output the parameters of a distribution that’s not a fixed-variance Gaussian.

In particular, a mixture of one-dimensional Gaussians for each autoregressive conditional provides a flexible model. Given enough components, a mixture of Gaussians can model any continuous distribution to arbitrary precision. The resulting model can be interpreted as a sequence of mixture density networks (Bishop, 1994) with shared parameters. We call this model RNADE-MoG. In RNADE-MoG, each of the conditionals is modeled by a mixture of Gaussians:

$$p(\pi_{o_d} | \mathbf{x}_{o_{<d}}) = \sum_{c=1}^C \pi_{o_d,c} \mathcal{N}(\pi_{o_d}; \mu_{o_d,c}, \sigma_{o_d,c}^2), \quad (16)$$

where the parameters are set by the outputs of a neural network:

$$\pi_{o_d,c} = \frac{\exp \left\{ z_{o_d,c}^{(\pi)} \right\}}{\sum_{c=1}^C \exp \left\{ z_{o_d,c}^{(\pi)} \right\}} \quad (17)$$

$$\mu_{o_d,c} = z_{o_d,c}^{(\mu)} \quad (18)$$

$$\sigma_{o_d,c} = \exp \left\{ z_{o_d,c}^{(\sigma)} \right\} \quad (19)$$

$$z_{o_d,c}^{(\pi)} = b_{o_d,c}^{(\pi)} + \sum_{k=1}^H V_{o_d,k,c}^{(\pi)} h_{d,k} \quad (20)$$

$$z_{o_d,c}^{(\mu)} = b_{o_d,c}^{(\mu)} + \sum_{k=1}^H V_{o_d,k,c}^{(\mu)} h_{d,k} \quad (21)$$

$$z_{o_d,c}^{(\sigma)} = b_{o_d,c}^{(\sigma)} + \sum_{k=1}^H V_{o_d,k,c}^{(\sigma)} h_{d,k} \quad (22)$$

Parameter sharing conveys the same computational and statistical advantages as it does in the binary NADE.

Different one dimensional conditional forms may be preferred, for example due to limited data set size or domain knowledge about the form of the conditional distributions. Other choices, like single variable-variance Gaussians, sinh-arcsinh distributions, and mixtures of Laplace distributions, have been examined by Uria (2015).

Training an RNADE can still be done by stochastic gradient descent on the parameters of the model with respect to the negative log-density of the training set. It was found empirically (Uria et al., 2013) that stochastic gradient descent leads to better parameter configurations when the gradient of the mean $\left(\frac{\partial \mu}{\partial \pi_{o_d,c}} \right)$ was multiplied by the standard deviation $(\sigma_{o_d,c})$.

4. Orderless and Deep NADE

The fixed ordering of the variables in a NADE model makes the exact calculation of arbitrary conditional probabilities computationally intractable. Only a small subset of conditional distributions, those where the conditioned variables are at the beginning of the ordering and marginalized variables at the end, are computationally tractable.

Another limitation of NADE is that a naive extension to a deep version, with multiple layers of hidden units, is computationally expensive. Deep neural networks (Bengio, 2009; LeCun et al., 2015) are at the core of state-of-the-art models for supervised tasks like image recognition (Krizhevsky et al., 2012) and speech recognition (Dahl et al., 2013). The same inductive bias should also provide better unsupervised models. However, extending the NADE framework to network architectures with several hidden layers, by introducing extra non-linear calculations between Equations 3 and 2, increases its complexity to cubic in the number of units per layer. Specifically, the cost becomes $O(DH^2L)$, where L stands for the number of hidden layers and can be assumed to be a small constant, D is the number of variables modeled, and H is the number of hidden units, which we assumed to be of the same order as D . This increase in complexity is caused by no longer being able to share hidden layer computations across the conditionals in Equation 1, after the non-linearity in the first layer.

In this section we describe an order-agnostic training procedure, DeepNADE (Uria et al., 2014), which will address both of the issues above. This procedure trains a single deep neural network that can assign a conditional distribution to any variable given any subset of the others. This network can then provide the conditionals in Equation 1 for any ordering of the input observations. Therefore, the network defines a factorial number of different models with shared parameters, one for each of the $D!$ orderings of the inputs. At test time, given an inference task, the most convenient ordering of variables can be used. The models for different orderings will not be consistent with each other: they will assign different probabilities to a given test vector. However, we can use the models’ differences to our advantage by creating ensembles of NADE models (Section 4.1), which results in better estimators than any single NADE. Moreover, the training complexity of our procedure increases linearly with the number of hidden layers $O(H^2L)$, while remaining quadratic in the size of the network’s layers.

We first describe the model for an L -layer neural network modeling binary variables. A conditional distribution is obtained directly from a hidden unit in the final layer:

$$p(x_{o_d} = 1 \mid \mathbf{x}_{o_{<d}}, \boldsymbol{\theta}, o_{<d}, o_d) = \mathbf{h}_{o_d}^{(L)}. \quad (23)$$

This hidden unit is computed from previous layers, all of which can only depend on the $\mathbf{x}_{o_{<d}}$ variables that are currently being conditioned on. We remove the other variables from the computation using a binary mask,

$$\mathbf{m}_{o_{<d}} = [1_{1 \in o_{<d}}, 1_{2 \in o_{<d}}, \dots, 1_{D \in o_{<d}}], \quad (24)$$

which is element-wise multiplied with the inputs before computing the remaining layers as in a standard neural network:

$$\mathbf{h}^{(0)} = \mathbf{x} \odot \mathbf{m}_{o_{<d}} \quad (25)$$

$$\mathbf{a}^{(\ell)} = \mathbf{W}^{(\ell)} \mathbf{h}^{(\ell-1)} + \mathbf{b}^{(\ell)} \quad (26)$$

$$\mathbf{h}^{(\ell)} = \sigma(\mathbf{a}^{(\ell)}) \quad (27)$$

$$\mathbf{h}^{(L)} = \text{sigm}(\mathbf{a}^{(L)}). \quad (28)$$

The network is specified by a free choice of the activation function $\sigma(\cdot)$, and learnable parameters $\mathbf{W}^{(\ell)} \in \mathbb{R}^{H^{(\ell)} \times H^{(\ell-1)}}$ and $\mathbf{b}^{(\ell)} \in \mathbb{R}^{H^{(\ell)}}$, where $H^{(\ell)}$ is the number of units in the ℓ -th layer. As layer zero is the masked input, $H^{(0)} = D$. The final L -th layer needs to be able to provide predictions for any element (Equation 23) and so also has D units.

To train a DeepNADE, the ordering of the variables is treated as a stochastic variable with a uniform distribution. Moreover, since we wish DeepNADE to provide good predictions for any ordering, we optimize the expected likelihood over the ordering of variables:

$$\mathcal{J}(\boldsymbol{\theta}) = \mathbb{E}_{o \in D!} [-\log p(\mathbf{X} \mid \boldsymbol{\theta}, o)] \propto \mathbb{E}_{o \in D!} \mathbb{E}_{\mathbf{x} \in \mathcal{X}} [-\log p(\mathbf{x} \mid \boldsymbol{\theta}, o)], \quad (29)$$

where we've made the dependence on the ordering o and the network's parameters $\boldsymbol{\theta}$ explicit, $D!$ stands for the set of all orderings (the permutations of D elements) and \mathbf{x} is a uniformly sampled data point from the training set \mathcal{X} . Using NADE's expression for the density of a data point in Equation 1 we have

$$\mathcal{J}(\boldsymbol{\theta}) = \mathbb{E}_{o \in D!} \mathbb{E}_{\mathbf{x} \in \mathcal{X}} \sum_{d=1}^D -\log p(x_{o_d} \mid \mathbf{x}_{o_{<d}}, \boldsymbol{\theta}, o), \quad (30)$$

where d indexes the elements in the ordering, o , of the variables. By moving the expectation over orderings inside the sum over the elements of the ordering, the ordering can be split in three parts: $o_{<d}$ (the indices of the $d-1$ first dimensions in the ordering), o_d (the index of the d -th variable) and $o_{>d}$ (the indices of the remaining dimensions). Therefore, the loss function can be rewritten as:

$$\mathcal{J}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x} \in \mathcal{X}} \sum_{d=1}^D \mathbb{E}_{o_{<d}} \mathbb{E}_{o_d} \mathbb{E}_{o_{>d}} [-\log p(x_{o_d} \mid \mathbf{x}_{o_{<d}}, \boldsymbol{\theta}, o_{<d}, o_d, o_{>d})]. \quad (31)$$

The value of each of these terms does not depend on $o_{>d}$. Therefore, it can be simplified as:

$$\mathcal{J}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x} \in \mathcal{X}} \sum_{d=1}^D \mathbb{E}_{o_{<d}} \mathbb{E}_{o_d} [-\log p(x_{o_d} \mid \mathbf{x}_{o_{<d}}, \boldsymbol{\theta}, o_{<d}, o_d)]. \quad (32)$$

In practice, this loss function will have a very high number of terms and will have to be approximated by sampling \mathbf{x} , d and $o_{<d}$. The innermost expectation over values of o_d can be calculated cheaply, because all of the neural network computations depend only on the masked input $\mathbf{x}_{o_{<d}}$, and can be reused for each possible o_d . Assuming all orderings are equally probable, we will estimate $\mathcal{J}(\boldsymbol{\theta})$ by:

$$\hat{\mathcal{J}}(\boldsymbol{\theta}) = \frac{D}{D-d+1} \sum_{o_d} -\log p(x_{o_d} \mid \mathbf{x}_{o_{<d}}, \boldsymbol{\theta}, o_{<d}, o_d), \quad (33)$$

which is an unbiased estimator of Equation 29. Therefore, training can be done by descent on the gradient of $\hat{\mathcal{J}}(\boldsymbol{\theta})$.

For binary observations, we use the cross-entropy scaled by a factor of $\frac{D}{D-d+1}$ as the training loss which corresponds to minimizing $\hat{\mathcal{J}}$:

$$\mathcal{J}(\mathbf{x}) = \frac{D}{D-d+1} \mathbf{m}_{o_{>d}}^\top (\mathbf{x} \odot \log(\mathbf{h}^{(L)}) + (1-\mathbf{x}) \odot \log(1-\mathbf{h}^{(L)})). \quad (34)$$

Differentiating this cost involves backpropagating the gradients of the cross-entropy only from the outputs in $o_{>d}$ and rescaling them by $\frac{D}{D-d+1}$.

The resulting training procedure resembles that of a denoising autoencoder (Vincent et al., 2008). Like the autoencoder, D outputs are used to predict D inputs corrupted by a random masking process ($\mathbf{m}_{o_{<d}}$ in Equation 25). A single forward pass can compute $\mathbf{h}_{o_{>d}}^{(L)}$, which provides a prediction $p(x_{o_d} = 1 \mid \mathbf{x}_{o_{<d}}, \boldsymbol{\theta}, o_{<d}, o_d)$ for every masked variable, which could be used next in an ordering starting with $o_{<d}$. Unlike the autoencoder, the outputs for variables corresponding to those provided in the input (not masked out) are ignored.

In this order-agnostic framework, missing variables and zero-valued observations are indistinguishable by the network. This shortcoming can be alleviated by concatenating the inputs to the network (masked variables $\mathbf{x} \odot \mathbf{m}_{o_{<d}}$) with the mask $\mathbf{m}_{o_{<d}}$. Therefore we advise substituting the input described in Equation 25 with

$$\mathbf{h}^{(0)} = \text{concat}(\mathbf{x} \odot \mathbf{m}_{o_{<d}}, \mathbf{m}_{o_{<d}}). \quad (35)$$

We found this modification to be important in order to obtain competitive statistical performance (see Table 3). The resulting neural network is illustrated in Figure 2.

4.1 Ensembles of NADE Models

As mentioned, the DeepNADE parameter fitting procedure effectively produces a factorial number of different NADE models, one for each ordering of the variables. These models will not, in general, assign the same probability to any particular data point. This disagreement is undesirable if we require consistent inferences for different inference problems, as it will preclude the use of the most convenient ordering of variables for each inference task.

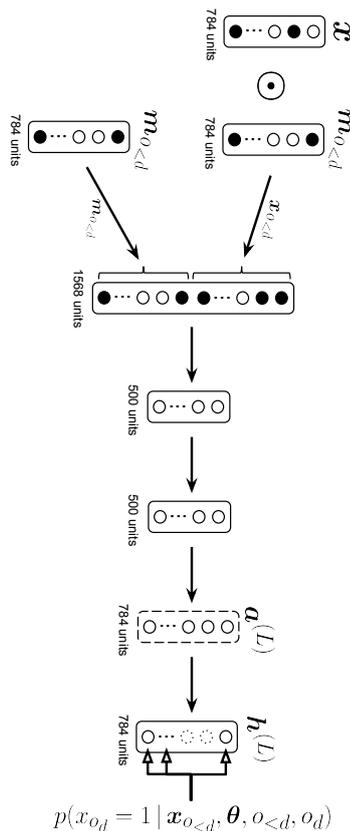


Figure 2: Illustration of a DeepNADe model with two hidden layers. The dashed border represents a layer pre-activation. Units with value 0 are shown in black while units with value 1 are shown in white. A mask $m_{o_{<d}}$ specifies a subset of variables to condition on. A conditional or predictive probability of the remaining variables is given in the final layer. The output units with a corresponding input mask of value 1 (shown with dotted contour) are not involved in DeepNADe’s training loss (Equation 34).

However, it is possible to use this variability across the different orderings to our advantage by combining several models. A usual approach to improve on a particular estimator is to construct an ensemble of multiple, strong but different estimators, e.g. using bagging (Breiman and Cutler, 1994) or stacking (Smyth and Wolpert, 1999). The DeepNADe training procedure suggests a way of generating ensembles of NADe models: take a set of uniformly distributed orderings $\{o^{(k)}\}_{k=1}^K$ over the input variables and use the average probability $\frac{1}{K} \sum_{k=1}^K p(\mathbf{x} | \theta, o^{(k)})$ as an estimator.

The use of an ensemble increases the test-time cost of density estimation linearly with the number of orderings used. The complexity of sampling does not change however: after one of the K orderings is chosen at random, the single corresponding NADe is sampled. Importantly, the cost of training also remains the same, unlike other ensemble methods such as bagging. Furthermore, the number of components can be chosen after training and even adapted to a computational budget on the fly.

5. ConvNADe: Convolutional NADe

One drawback of NADe (and its variants so far) is the lack of a mechanism for truly exploiting the high-dimensional structure of the data. For example, when using NADe on binarized MNIST, we first need to flatten the 2D images before providing them to the model as a vector. As the spatial topology is not provided to the network, it can’t use this information to share parameters and may learn less quickly.

Recently, convolutional neural networks (CNN) have achieved state-of-the-art performance on many supervised tasks related to images (Krizhevsky et al. (2012)). Briefly, CNNs are composed of convolutional layers, each one having multiple learnable filters. The outputs of a convolutional layer are feature maps and are obtained by the convolution on the input image (or previous feature maps) of a linear filter, followed by the addition of a bias and the application of a non-linear activation function. Thanks to the convolution, spatial structure in the input is preserved and can be exploited. Moreover, as per the definition of a convolution the same filter is reused across all sub-regions of the entire image (or previous feature maps), yielding a parameter sharing that is natural and sensible for images.

The success of CNNs raises the question: can we exploit the spatial topology of the inputs while keeping NADe’s autoregressive property? It turns out we can, simply by replacing the fully connected hidden layers of a DeepNADe model with convolutional layers. We thus refer to this variant as Convolutional NADe (ConvNADe).

First we establish some notation that we will use throughout this section. Without loss of generality, let the input $\mathbf{X} \in \{0, 1\}^{N \times N \times N}$ be a square binary image of size $N \times N$ and every convolution filter $\mathbf{W}_{ij}^{(\ell)} \in \mathbb{R}^{N_W^{(\ell)} \times N_W^{(\ell)}}$ connecting two feature maps $\mathbf{H}_i^{(\ell-1)}$ and $\mathbf{H}_j^{(\ell)}$ also be square with their size $N_W^{(\ell)}$ varying for each layer ℓ . We also define the following mask $\mathbf{M}_{o_{<d}} \in \{0, 1\}^{N \times N \times N}$, which is 1 for the locations of the first $d - 1$ pixels in the ordering o . Formally, Equation 26 is modified to use convolutions instead of dot products. Specifically for an L -layer convolutional neural network that preserves the input shape (explained below) we have

$$p(x_{o_d} = 1 | \mathbf{x}_{o_{<d}}, \theta, o_{>d}, o_d) = \text{vec} \left(\mathbf{H}_1^{(L)} \right)_{o_d}, \quad (36)$$

with

$$\mathbf{H}_1^{(0)} = \mathbf{X} \odot \mathbf{M}_{o_{<d}} \quad (37)$$

$$\mathbf{A}_j^{(\ell)} = h_j^{(\ell)} + \sum_{i=1}^{H^{(\ell-1)}} \mathbf{H}_i^{(\ell-1)} \otimes \mathbf{W}_{ij}^{(\ell)} \quad (38)$$

$$\mathbf{H}_j^{(\ell)} = \sigma \left(\mathbf{A}_j^{(\ell)} \right) \quad (39)$$

$$\mathbf{H}_j^{(L)} = \text{sigm} \left(\mathbf{A}_j^{(L)} \right), \quad (40)$$

where $H^{(\ell)}$ is the number of feature maps output by the ℓ -th layer and $\mathbf{h}^{(\ell)} \in \mathbb{R}^{H^{(\ell)}}$, $\mathbf{W}^{(\ell)} \in \mathbb{R}^{H^{(\ell-1)} \times H^{(\ell)} \times N_W^{(\ell)} \times N_W^{(\ell)}}$, with \odot denoting the element-wise multiplication, $\sigma(\cdot)$ being any activation function and $\text{vec}(\mathbf{X}) \rightarrow \mathbf{x}$ is the concatenation of every row in \mathbf{X} . Note that $H^{(0)}$ corresponds to the number of channels the input images have.

For notational convenience, we use \otimes to denote both “valid” convolutions and “full” convolutions, instead of introducing bulky notations to differentiate these cases. The “valid” convolutions only apply a filter to complete patches of the image, resulting in a smaller image (its shape is decreased to $N \times N - N_W^{(\ell)} + 1$). Alternatively, “full” convolutions zero-pad the contour of the image before applying the convolution, thus expanding the image (its shape is increased to $N \times N + N_W^{(\ell)} - 1$). Which one is used should be self-explanatory depending on the context. Note that we only use convolutions with a stride of 1.

Moreover, in order for ConvNADE to output conditional probabilities as shown in Equation 36, the output layer must have only one feature map $\mathbf{H}_1^{(L)}$, whose dimension matches the dimension of the input \mathbf{X} . This can be achieved by carefully combining layers that use either “valid” or “full” convolutions.

To explore different model architectures respecting that constraint, we opted for the following strategy. Given a network, we ensured the first half of its layers was using “valid” convolutions while the other half would use “full” convolutions. In addition to that, we made sure the network was symmetric with respect to its filter shapes (i.e. the filter shape used in layer ℓ matched the one used in layer $L - \ell$).

For completeness, we wish to mention that ConvNADE can also include pooling and upsampling layers, but we did not see much improvement when using them. In fact, recent research suggests that these types of layers are not essential to obtain state-of-the-art results (Springenberg et al., 2015).

The flexibility of DeepNADE allows us to easily combine both convolutional and fully connected layers. To create such hybrid models, we used the simple strategy of having two separate networks, with their last layer fused together at the end. The ‘convnet’ part is only composed of convolutional layers whereas the ‘fullnet’ part is only composed of fully connected layers. The forward pass of both networks follows respectively Equations 37–39 and Equations 25–27. Note that in the ‘fullnet’ network case, \mathbf{x} corresponds to the input image having been flattened.

In the end, the output layer \mathbf{g} of the hybrid model corresponds to the aggregation of the last layer pre-activation of both ‘convnet’ and ‘fullnet’ networks. The conditionals are slightly modified as follows:

$$p(\alpha_{o_d} = 1 | \mathbf{x}_{o_{<d}}, \boldsymbol{\theta}, o_{<d}, o_d) = \mathbf{g}_{o_d} \quad (41)$$

$$\mathbf{g} = \mathbf{sigm} \left(\mathbf{vec} \left(\mathbf{A}_1^{(L)} \right) + \mathbf{a}^{(L)} \right). \quad (42)$$

The same training procedure as for DeepNADE model can also be used for ConvNADE. For binary observations, the training loss is similar to Equation 34, with $\mathbf{h}^{(L)}$ being substituted for \mathbf{g} as defined in Equation 42.

As for the DeepNADE model, we found that providing the mask $\mathbf{M}_{o_{<d}}$ as an input to the model improves performance (see Table 4). For the ‘convnet’ part, the mask was provided as an additional channel to the input layer. For the ‘fullnet’ part, the inputs were concatenated with the mask as shown in Equation 35.

The final architecture is shown in Figure 3. In our experiments, we found that this type of hybrid model works better than only using convolutional layers (see Table 4). Certainly, more complex architectures could be employed but this is a topic left for future work.

6. Related Work

As we mentioned earlier, the development of NADE and its extensions was motivated by the question of whether a tractable distribution estimator could be designed to match a powerful but intractable model such as the restricted Boltzmann machine.

The original inspiration came from the autoregressive approach taken by fully visible sigmoid belief networks (FVSBN), which were shown by Frey et al. (1996) to be surprisingly

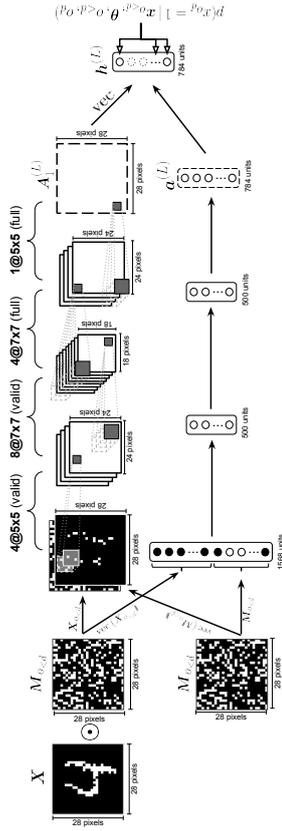


Figure 3: Illustration of a ConvNADE that combines a convolutional neural network with three hidden layers and a fully connected feed-forward neural network with two hidden layers. The dashed border represents a layer pre-activation. Units with a dotted contour are not valid conditionals since they depend on themselves i.e. they were given in the input.

competitive, despite the simplicity of the distribution family for its conditionals. Bengio and Bengio (2000) later proposed using more powerful conditionals, modeled as single layer neural networks. Moreover, they proposed connecting the output of each d^{th} conditional to all of the hidden layers of the $d - 1$ neural networks for the preceding conditionals. More recently, Germain et al. (2015) generalized this model by deriving a simple procedure for making it deep and orderless (akin to DeepNADE, in Section 4). We compare with all of these approaches in Section 7.1.

There exists, of course, more classical and non-autoregressive approaches to tractable distribution estimation, such as mixture models and Chow-Liu trees (Chow and Liu, 1968). We compare with these as well in Section 7.1.

This work also relates directly to the recently growing literature on generative neural networks. In addition to the autoregressive approach described in this paper, there exists three other types of such models: directed generative networks, undirected generative networks and hybrid networks.

Work on directed generative networks dates back to the original work on sigmoid belief networks (Neal, 1992) and the Helmholtz machine (Hinton et al., 1995; Dayan et al., 1995). Helmholtz machines are equivalent to a multilayer sigmoid belief network, with each using binary stochastic units. Originally they were trained using Gibbs sampling and gradient descent (Neal, 1992), or with the so-called wake sleep algorithm (Hinton et al., 1995). More recently, many alternative directed models and training procedures have been proposed. Kingma and Welling (2014); Rezende et al. (2014) proposed the variational autoencoder (VAE), where the model is the same as the Helmholtz machine, but with real-valued (usually Gaussian) stochastic units. Importantly, Kingma and Welling (2014) identified a reparameterization trick making it possible to train the VAE in a way that resembles the training of an autoencoder. This approach falls in the family of stochastic variational inference methods, where the encoder network corresponds to the approximate variational

posterior. The VAE optimizes a bound on the likelihood which is estimated using a single sample from the variational posterior, though recent work has shown that a better bound can be obtained using an importance sampling approach (Burda et al., 2016). Gregor et al. (2015) later exploited the VAE approach to develop DRAM, a directed generative model for images based on a read-write attentional mechanism. Goodfellow et al. (2014) proposed an adversarial approach to training directed generative networks, that relies on a discriminator network simultaneously trained to distinguish between data and model samples. Generative networks trained this way are referred to as Generative Adversarial Networks (GAN). While the VAE optimizes a bound of the likelihood (which is the KL divergence between the empirical and model distributions), it can be shown that the earliest versions of GANs optimize the Jensen–Shannon (JS) divergence between the empirical and model distributions. Li et al. (2015) instead propose a training objective derived from Maximum Mean Discrepancy (MMD; Gretton et al., 2007). Recently, the directed generative model approach has been very successfully applied to model images (Denton et al., 2015; Sohl-Dickstein et al., 2011).

The undirected paradigm has also been explored extensively for developing powerful generative networks. These include the restricted Boltzmann machine (Smolensky, 1986; Freund and Haussler, 1992) and its multilayer extension, the deep Boltzmann machine (Salakhutdinov and Hinton, 2009), which dominate the literature on undirected neural networks. Salakhutdinov and Murray (2008) provided one of the first quantitative evidence of the generative modeling power of RBMs, which motivated the original parameterization for NADE (Larochelle and Murray, 2011). Efforts to train better undirected models can vary in nature. One has been to develop alternative objectives to maximum likelihood. The proposal of Contrastive Divergence (CD; Hinton, 2002) was instrumental in the popularization of the RBM. Other proposals include pseudo-likelihood (Besag, 1975; Marlin et al., 2010), score matching (Hyvärinen, 2005; Hyvärinen, 2007a,b), noise contrastive estimation (Gutmann and Hyvärinen, 2010) and probability flow minimization (Sohl-Dickstein et al., 2011). Another line of development has been to optimize likelihood using Robbins–Morro stochastic approximation (Younes, 1989), also known as Persistent CD (Telesman, 2008), and develop good MCMC samplers for deep undirected models (Salakhutdinov, 2009, 2010; Desjardins et al., 2010; Cho et al., 2010). Work has also been directed towards proposing improved update rules or parameterization of the model’s energy function (Telesman and Hinton, 2009; Cho et al., 2013; Montavon and Müller, 2012) as well as improved approximate inference of the hidden layers (Salakhutdinov and Larochelle, 2010). The work of Ngiem et al. (2011) also proposed an undirected model that distinguishes itself from deep Boltzmann machines by having deterministic hidden units, instead of stochastic.

Finally, hybrids of directed and undirected networks are also possible, though much less common. The most notable case is the Deep Belief Network (DBN; Hinton et al., 2006), which corresponds to a sigmoid belief network for which the prior over its top hidden layer is an RBM (whose hidden layer counts as an additional hidden layer). The DBN revived interest in RBMs, as they were required to successfully initialize the DBN.

NADE thus substantially differs from this literature focusing on directed and undirected models, benefiting from a few properties that these approaches lack. Mainly, NADE does not rely on latent stochastic hidden units, making it possible to tractably compute its associated data likelihood for some given ordering. This in turn makes it possible to efficiently produce

Name	# Inputs	Train	Valid.	Test
Adult	123	5000	1414	26147
Connect4	126	16000	4000	47557
DNA	180	1400	600	1186
Mushrooms	112	2000	500	5624
NIPS-0-12	500	400	100	1240
OCR-letters	128	32152	10000	10000
RCV1	150	40000	10000	150000
Web	300	14000	3188	32561

Table 1: Statistics on the binary vector data sets of Section 7.1.

exact samples from the model (unlike in undirected models) and get an unbiased gradient for maximum likelihood training (unlike in directed graphical models).

7. Results

In this section, we evaluate the performance of our different NADE models on a variety of data sets. The code to reproduce the experiments of the paper is available on GitHub¹. Our implementation is done using Theano (Team et al., 2016).

7.1. Binary Vectors Data Sets

We start by evaluating the performance of NADE models on a set of benchmark data sets where the observations correspond to binary vectors. These data sets were mostly taken from the LIBSVM data sets web site², except for OCR-letters³ and NIPS-0-12⁴. Code to download these data sets is available here: <http://info.usherbrooke.ca/hlarochelle/code/nae.tar.gz>. Table 1 summarizes the main statistics for these data sets.

For these experiments, we only consider tractable distribution estimators, where we can evaluate $p(\mathbf{x})$ on test items exactly. We consider the following baselines:

- **MoB**: A mixture of multivariate Bernoullis, trained using the EM algorithm. The number of mixture components was chosen from {32, 64, 128, 256, 512, 1024} based on validation set performance, and early stopping was used to determine the number of EM iterations.
- **RBM**: A restricted Boltzmann machine made tractable by using only 23 hidden units, trained by contrastive divergence with up to 25 steps of Gibbs sampling. The validation set performance was used to select the learning rate from {0.005, 0.0005, 0.00005}, and the number of iterations over the training set from {100, 500, 1000}.

1. <http://github.com/MarcCote/NADE>

2. <http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>

3. <http://ai.stanford.edu/~btaskar/ocr/>

4. <http://www.cs.nyu.edu/~trovets/data.html>

- **FVSBN**: Fully visible sigmoid belief network, that models each conditional $p(x_{o_d} | \mathbf{x}_{\rho_{<d}})$ with logistic regression. The ordering of inputs was selected randomly. Training was by stochastic gradient descent. The validation set was used for early stopping, as well as for choosing the base learning rate $\eta \in \{0.05, 0.005, 0.0005\}$, and a decreasing schedule constant γ from $\{0, 0.001, 0.000001\}$ for the learning rate schedule $\eta/(1 + \gamma t)$ for the t^{th} update.

- **Chow-Liu**: A Chow-Liu tree is a graph over the observed variables, where the distribution of each variable, except the root, depends on a single parent node. There is an $O(D^2)$ fitting algorithm to find the maximum likelihood tree and conditional distributions (Chow and Liu, 1968). We adapted an implementation provided by Harmeling and Williams (2011), who found Chow-Liu to be a strong baseline.

The maximum likelihood parameters are not defined when conditioning on events that haven't occurred in the training set. Moreover, conditional probabilities of zero are possible, which could give infinitely bad test set performance. We re-estimated the conditional probabilities on the Chow-Liu tree using Lidstone or "add- α " smoothing:

$$p(x_d = 1 | x_{\text{parent}} = z) = \frac{\text{count}(x_d = 1 | x_{\text{parent}} = z) + \alpha}{\text{count}(x_{\text{parent}} = z) + 2\alpha}, \quad (43)$$

selecting α for each data set from $\{10^{-20}, 0.001, 0.01, 0.1\}$ based on performance on the validation set.

- **MADE** (Germain et al., 2015): Generalization of the neural network approach of Bengio and Bengio (2000), to multiple layers. We consider a version using a single (fixed) input ordering and another trained on multiple orderings from which an ensemble was constructed (which was inspired from the order-agnostic approach of Section 4) that we refer to as MADE-E. See Germain et al. (2015) for more details.

We compare these baselines with the two following NADE variants:

- **NADE (fixed order)**: Single layer NADE model, trained on a single (fixed) randomly generated order, as described in Section 2. The sigmoid activation function was used for the hidden layer, of size 500. Much like for FVSBN, training relied on stochastic gradient descent and the validation set was used for early stopping, as well as for choosing the learning rate from $\{0.05, 0.005, 0.0005\}$, and the decreasing schedule constant γ from $\{0, 0.001, 0.000001\}$.

- **NADE-E**: Single layer NADE trained according to the order-agnostic procedure described in Section 4. The rectified linear activation function was used for the hidden layer, also of size 500. Minibatch gradient descent was used for training, with minibatches of size 100. The initial learning rate, chosen among $\{0.016, 0.004, 0.001, 0.00025, 0.0000675\}$, was linearly decayed to zero over the course of 100,000 parameter updates. Early stopping was used, using Equation 34 to get a stochastic estimate of the validation set average log-likelihood. An ensemble using 16 orderings was used to compute the test-time log-likelihood.

Model	Adult	Connect4	DNA	Mushrooms	NIPS-0-12	OCR-letters	RCV1	Web
MoB	-20.44	-23.41	-98.19	-14.46	-290.02	-40.56	-47.59	-30.16
RBM	-16.26	-22.66	-96.74	-15.15	-277.37	-43.05	-48.88	-29.38
FVSBN	-13.17	-12.39	-83.64	-10.27	-276.88	-39.30	-49.84	-29.35
Chow-Liu	-18.51	-20.57	-87.72	-20.99	-281.01	-48.87	-55.60	-33.92
MADE	-13.12	-11.90	-83.63	-9.68	-280.25	-28.34	-47.10	-28.53
MADE-E	-13.13	-11.90	-79.66	-9.69	-277.28	-30.04	-46.74	-28.25
NADE	-13.19	-11.99	-84.81	-9.81	-273.08	-27.22	-46.66	-28.39
NADE-E	-13.19	-12.58	-82.31	-9.69	-272.39	-27.32	-46.12	-27.87

Table 2: Average log-likelihood performance of tractable distribution baselines and NADE models, on binary vector data sets. The best result is shown in bold, along with any other result with an overlapping confidence interval.

Table 2 presents the results. We observe that NADE restricted to a fixed ordering of the inputs achieves very competitive performance compared to the baselines. However, the order-agnostic version of NADE is overall the best method, being among the top performing model for 5 data sets out of 8.

The performance of fixed-order NADE is surprisingly robust to variations of the chosen input ordering. The standard deviation on the average log-likelihood when varying the ordering was small: on Mushrooms, DNA and NIPS-0-12, we observed standard deviations of 0.045, 0.05 and 0.15, respectively. However, models with different orders can do well on different test examples, which explains why ensembling can still help.

7.2 Binary Image Data Set

We now consider the case of an image data set, constructed by binarizing the MNIST digit data set. Each image has been stochastically binarized according to their pixel intensity as generated by Salakhutdinov and Murray (2008). This benchmark has been a popular choice for the evaluation of generative neural network models. Here, we investigate two questions:

1. How does NADE compare to intractable generative models?
2. Does the use of a convolutional architecture improve the performance of NADE?

For these experiments, in addition to the baselines already described in Section 7.1, we consider the following:

- **DARN** (Gregor et al., 2014): This deep generative autoencoder has two hidden layers, one deterministic and one with binary stochastic units. Both layers have 500 units (denoted as $n_h = 500$). Adaptive weight noise (adaNoise) was either used or not to avoid the need for early stopping (Graves, 2011). Evaluation of exact test probabilities is intractable for large latent representations. Hence, Monte Carlo was used to approximate the expected description length, which corresponds to an upper bound on the negative log-likelihood.

Model	$-\log p$	\approx
MoBertoullis K=10	168.95	
MoBertoullis K=500	137.64	
Chow-Lin tree	134.99	
MADDE 2hl (32 masks)	86.64	
RBM (500 h, 25 CD steps)		86.34
DBN 2hl		84.55
DARN $n_h = 500$		84.71
DARN $n_h = 500$ (adaNoise)		84.13
NADE (fixed order)	88.33	
DeepNADE 1hl (no input masks)	99.37	
DeepNADE 2hl (no input masks)	95.33	
DeepNADE 1hl	92.17	
DeepNADE 2hl	89.17	
DeepNADE 3hl	89.38	
DeepNADE 4hl	89.60	
EoNADE 1hl (2 orderings)	90.69	
EoNADE 1hl (128 orderings)	87.71	
EoNADE 2hl (2 orderings)	87.96	
EoNADE 2hl (128 orderings)	85.10	

Table 3: Negative log-likelihood test results of models ignorant of the 2D topology on the binarized MNIST data set.

- **DRAW** (Gregor et al., 2015): Similar to a variational autoencoder where both the encoder and the decoder are LSTMs, guided (or not) by an attention mechanism. In this model, both LSTMs (encoder and decoder) are composed of 256 recurrent hidden units and always perform 64 timesteps. When the attention mechanism is enabled, patches (2×2 pixels) are provided as inputs to the encoder instead of the whole image and the decoder also produces patches (5×5 pixels) instead of a whole image.
- **Pixel RNN** (Oord et al., 2016): NADE-like model for natural images that is based on convolutional and LSTM hidden units. This model has 7 hidden layers, each composed of 16 units. Oord et al. (2016) proposed a novel two-dimensional LSTM, named Diagonal BiLSTM, which is used in this model. Unlike our ConvNADE, the ordering is fixed before training and at test time, and corresponds to a scan of the image in a diagonal fashion starting from a corner at the top and reaching the opposite corner at the bottom.

We compare these baselines with some NADE variants. The performance of a basic (fixed-order, single hidden layer) NADE model is provided in Table 3 and samples are illustrated in Figure 4. More importantly, we will focus on whether the following variants achieve better test set performance:

- **DeepNADE**: Multiple layers (1hl, 2hl, 3hl or 4hl) trained according to the order-agnostic procedure described in Section 4. Information about which inputs are masked

was either provided or not (no input masks) to the model. The rectified linear activation function was used for all hidden layers. Minibatch gradient descent was used for training, with minibatches of size 1000. Training consisted of 200 iterations of 1000 parameter updates. Each hidden layer was pre-trained according to Algorithm 2. We report an average of the average test log-likelihoods over ten different random orderings.

- **EoNADE**: This variant is similar to DeepNADE except for the log-likelihood on the test set, which is instead computed from an ensemble that averages predictive probabilities over 2 or 128 orderings. To clarify, the DeepNADE results report the typical performance of one ordering, by averaging results after taking the log, and so do not combine the predictions of the models like EoNADE does.

- **ConvNADE**: Multiple convolutional layers trained according to the order-agnostic procedure described in Section 4. The exact architecture is shown in Figure 5(a). Information about which inputs are masked was either provided or not (no input masks). The rectified linear activation function was used for all hidden layers. The Adam optimizer (Kingma and Ba, 2015) was used with a learning rate of 10^{-4} . Early stopping was used with a look ahead of 10 epochs, using Equation 34 to get a stochastic estimate of the validation set average log-likelihood. An ensemble using 128 orderings was used to compute the log-likelihood on the test set.

- **ConvNADE + DeepNADE**: This variant is similar to ConvNADE except for the aggregation of a separate DeepNADE model at the end of the network. The exact architecture is shown in Figure 5(b). The training procedure is the same as with ConvNADE.

Algorithm 2 Pre-training of a NADE with n hidden layers on data set X .

```

procedure PRETRAIN( $n, X$ )
  if  $n = 1$  then
    return RANDOM-ONE-HIDDEN-LAYER-NADE
  else
     $\text{nade} \leftarrow$  PRETRAIN( $n - 1, X$ )
     $\text{nade} \leftarrow$  REMOVE-OUTPUT-LAYER( $\text{nade}$ )
     $\text{nade} \leftarrow$  ADD-A-NEW-HIDDEN-LAYER( $\text{nade}$ )
     $\text{nade} \leftarrow$  ADD-A-NEW-OUTPUT-LAYER( $\text{nade}$ )
     $\text{nade} \leftarrow$  TRAIN-ALL( $\text{nade}, X, \text{iters}=20$ )
    return  $\text{nade}$ 
   $\triangleright$  Train for 20 iterations.
  end if
end procedure

```

Table 3 presents the results obtained by models ignorant of the 2D topology, such as the basic NADE model. Addressing the first question, we observe that the order-agnostic version of NADE with two hidden layers is competitive with intractable generative models. Moreover, examples of the ability of DeepNADE to solve inference tasks by marginalization and conditional sampling are shown in Figure 6.

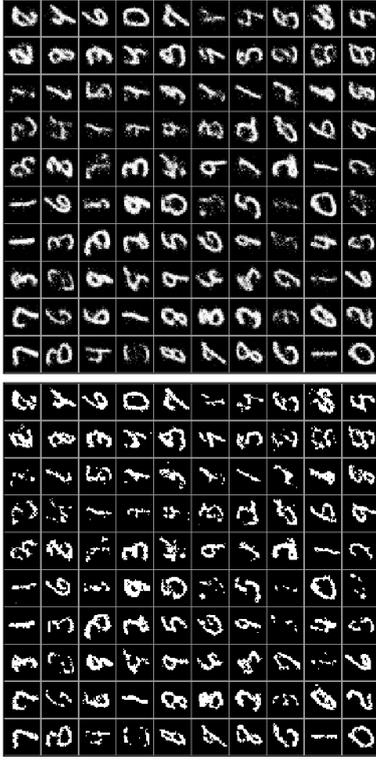


Figure 4: **Left:** samples from NADE trained on binarized MNIST. **Right:** probabilities from which each pixel was sampled. Ancestral sampling was used with the same fixed ordering used during training.

Model	$-\log p$	\leq
DRAW (without attention)	87.40	
DRAW	80.97	
Pixel RNN	79.20	
ConvNADE+DeepNADE (no input masks)	85.25	
ConvNADE	81.30	
ConvNADE+DeepNADE	80.82	

Table 4: Negative log-likelihood test results of models exploiting 2D topology on the binarized MNIST data set.

Now, addressing the second question, we can see from Table 4 that convolutions do improve the performance of NADE. Moreover, we observe that providing information about which inputs are masked is essential to obtaining good results. We can also see that combining convolutional and fully-connected layers helps. Even though ConvNADE+DeepNADE performs slightly worse than Pixel RNN, we note that our proposed approach is order-agnostic, whereas Pixel RNN requires a fixed ordering. Figure 7 shows samples obtained from the ConvNADE+DeepNADE model using ancestral sampling on a random ordering.

7.3 Real-Valued Observations Data Sets

In this section, we compare the statistical performance of RNADE to mixtures of Gaussians (MoG) and factor analyzers (MFA), which are surprisingly strong baselines in some tasks (Tang et al., 2012; Zoran and Weiss, 2012).

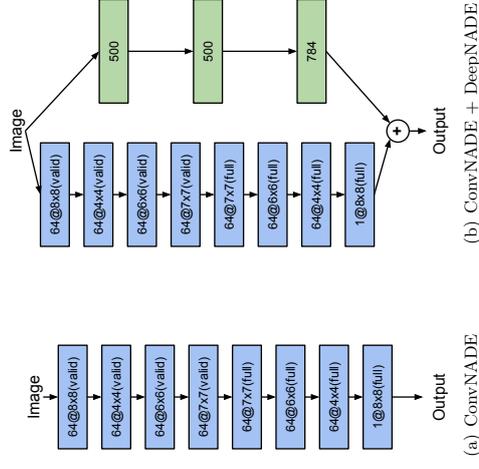


Figure 5: Network architectures for binarized MNIST. (a) ConvNADE with 8 convolutional layers (depicted in blue). The number of feature maps for a given layer is given by the number before the “@” symbol followed by the filter size and the type of convolution is specified in parentheses. (b) The same ConvNADE combined with a DeepNADE consisting of three fully-connected layers of respectively 500, 500 and 784 units.

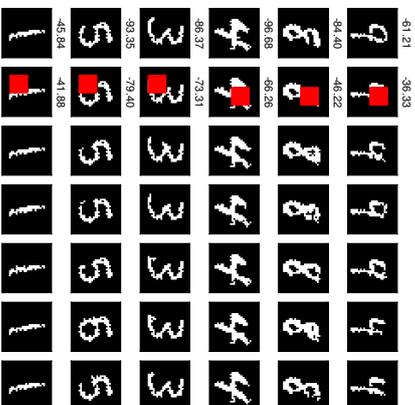


Figure 6: Example of marginalization and sampling. The first column shows five examples from the test set of the MNIST data set. The second column shows the density of these examples when a random 10×10 pixel region is marginalized. The right-most five columns show samples for the hollowed region. Both tasks can be done easily with a NADE where the pixels to marginalize are at the end of the ordering.

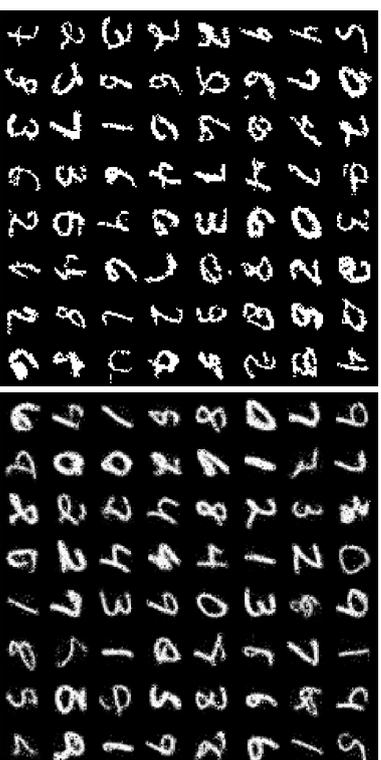


Figure 7: **Left:** samples from ConvNADE+DeepNADE trained on binarized MNIST. **Right:** probabilities from which each pixel was sampled. Ancestral sampling was used with a different random ordering for each sample.

7.3.1 LOW-DIMENSIONAL DATA

We start by considering three UCI data sets (Bache and Lichman, 2013), previously used to study the performance of other density estimators (Sliva et al., 2011; Tang et al., 2012), namely: *red wine*, *white wine* and *parkinsons*. These are low dimensional data sets (see Table 5) with hard thresholds and non-linear dependencies that make it difficult to fit mixtures of Gaussians or factor analyzers.

Following Tang et al. (2012), we eliminated discrete-valued attributes and an attribute from every pair with a Pearson correlation coefficient greater than 0.98. We normalized each dimension of the data by subtracting its training-subset sample mean and dividing by its standard deviation. All results are reported on the normalized data.

We use full-covariance Gaussians and mixtures of factor analyzers as baselines. Models were compared on their log-likelihood on held-out test data. Due to the small size of the data sets (see Table 5), we used 10-folds, using 90% of the data for training, and 10% for testing.

We chose the hyperparameter values for each model by doing per-fold cross-validation, using a ninth of the training data as validation data. Once the hyperparameter values have been chosen, we train each model using all the training data (including the validation data) and measure its performance on the 10% of held-out testing data. In order to avoid overfitting, we stopped the training after reaching a training likelihood higher than the one obtained on the best validation-wise iteration of the best validation run. Early stopping was important to avoid overfitting the RNADe models. It also improved the results of the MFAs, but to a lesser degree.

The MFA models were trained using the EM algorithm (Chahramani and Hinton, 1996; Verbeek, 2005). We cross-validated the number of components and factors. We also selected the number of factors from $2, 4, \dots, D$, where choosing D results in a mixture of Gaussians, and the number of components was chosen among $2, 4, \dots, 50$. Cross-validation selected fewer than 50 components in every case.

We report the performance of several RNADe models using different parametric forms for the one-dimensional conditionals: Gaussian with fixed variance (RNADe-FV), Gaussian with variable variance (RNADe-Gaussian), *sinh-arcsinh* distribution (RNADe-SAS), mixture of Gaussians (RNADe-MoG), and mixture of Laplace distributions (RNADe-MoL). All RNADe models were trained by stochastic gradient descent, using minibatches of size 100, for 500 epochs, each epoch comprising 10 minibatches. We fixed the number of hidden units to 50, and the non-linear activation function of the hidden units to ReLU. Three hyperparameters were cross-validated using grid-search: the number of components on each one-dimensional conditional (only applicable to the RNADe-MoG and RNADe-MoL models) was chosen from $\{2, 5, 10, 20\}$, the weight-decay (used only to regularize the input to hidden weights) from $\{2.0, 1.0, 0.1, 0.01, 0.001, 0\}$, and the learning rate from $\{0.1, 0.05, 0.025, 0.0125\}$. Learning rates were decreased linearly to reach 0 after the last epoch.

The results are shown in Table 6. RNADe with mixture of Gaussian conditionals was among the statistically significant group of best models on all data sets. As shown in Figure 8, RNADe-SAS and RNADe-MoG models are able to capture hard thresholds and heteroscedasticity.

	Red wine	White wine	Parkinsons
Dimensionality	11	11	15
Total number of data points	1599	4898	5875

Table 5: Dimensionality and size of the UCI data sets used in Section 7.3.1

Model	Red wine	White wine	Parkinsons
Gaussian	-13.18	-13.20	-10.85
MFA	-10.19	-10.73	-1.99
RNADE-FV	-12.29	-12.50	-8.87
RNADE-Gaussian	-11.99	-12.20	-3.47
RNADE-SAS	-9.86	-11.22	-3.07
RNADE-MoG	-9.36	-10.23	-0.90
RNADE-MoL	-9.46	-10.38	-2.63

Table 6: Average test set log-likelihoods per data point for seven models on three UCI data sets. Performances not in bold can be shown to be significantly worse than at least one of the results in bold as per a paired t -test on the ten mean-likelihoods (obtained from each data fold), with significance level 0.05.

7.3.2 NATURAL IMAGE PATCHES

We also measured the ability of RNADE to model small patches of natural images. Following the work of Zoran and Weiss (2011), we use 8-by-8-pixel patches of monochrome natural images, obtained from the BSDS300 data set (Martin et al., 2001; Figure 9 gives examples).

Pixels in this data set can take a finite number of brightness values ranging from 0 to 255. We added uniformly distributed noise between 0 and 1 to the brightness of each pixel. We then divided by 256, making the pixels take continuous values in the range $[0, 1]$. Adding noise prevents deceptively high-likelihood solutions that assign narrow high-density spikes around some of the possible discrete values.

We subtracted the mean pixel value from each patch. Effectively reducing the dimensionality of the data. Therefore we discarded the 64th (bottom-right) pixel, which would be perfectly predictable and models could fit arbitrarily high densities to it. All of the results in this section were obtained by fitting the pixels in a raster-scan order.

Experimental details follow. We trained our models by using patches randomly drawn from 180 images in the training subset of BSDS300. We used the remaining 20 images in the training subset as validation data. We used 1000 random patches from the validation subset to early-stop training of RNADE. We measured the performance of each model by their log-likelihood on one million patches drawn randomly from the test subset of 100 images not present in the training data. Given the larger scale of this data set, hyperparameters of the RNADE and MoG models were chosen manually using the performance of preliminary runs on the validation data, rather than by grid search.

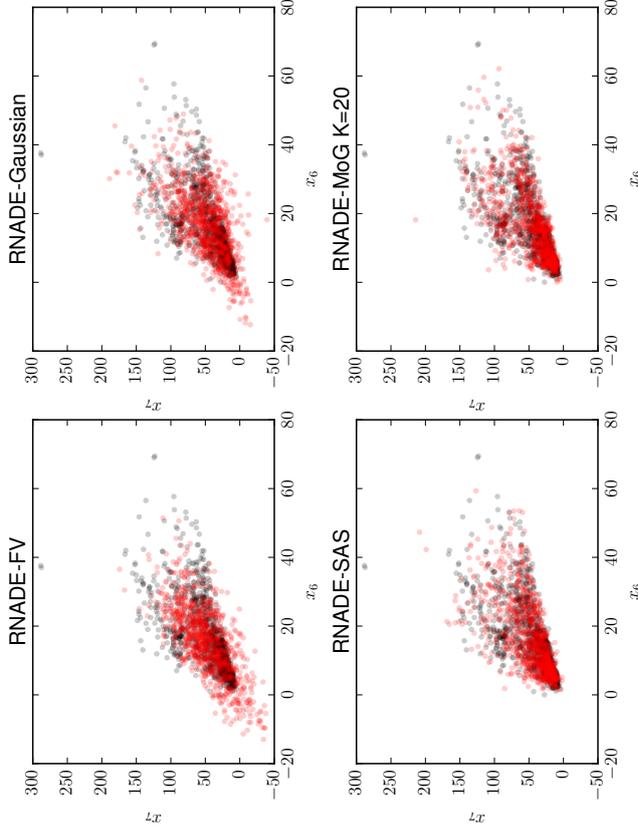


Figure 8: Scatter plot of dimensions x_7 vs x_6 of the *red wine* data set. A thousand data points from the data set are shown in black in all subfigures. As can be observed, this conditional distribution $p(x_7 | x_6)$ is heteroscedastic, skewed and has hard thresholds. In red, a thousand samples from four RNADE models with different one-dimensional conditional forms are shown. **Top-left:** In red, one thousand samples from a RNADE-FV model. **Top-right:** In red, one thousand samples from a RNADE-Gaussian model. **Bottom-left:** In red, one thousand samples from a RNADE-SAS (sinh-arcsinh distribution) model. **Bottom-right:** In red, one thousand samples from a RNADE-MoG model with 20 components per one-dimensional conditional. The RNADE-SAS and RNADE-MoG models successfully capture all the characteristics of the data.

All RNADe models reported use ReLU activations for the hidden units. The RNADe models were trained by stochastic gradient descent, using 25 data points per minibatch, for a total of 1,000 epochs, each comprising 1,000 minibatches. The learning rate was initialized to 0.001, and linearly decreased to reach 0 after the last epoch. Gradient momentum with factor 0.9 was used, but initiated after the first epoch. A weight decay rate of 0.001 was applied to the input-to-hidden weight matrix only. We found that multiplying the gradient of the mean output parameters by the standard deviation improves results of the models with mixture outputs⁵. RNADe training was early stopped but didn't show signs of overfitting. Even larger models might perform better.

The MoG models were trained using 1,000 iterations of minibatch EM. At each iteration 20,000 randomly sampled data points were used in an EM update. A step was taken from the previous parameters' value towards the parameters resulting from the M-step: $\theta_t = (1 - \eta)\theta_{t-1} + \eta\theta_{EM}$. The step size, η , was scheduled to start at 0.1 and linearly decreased to reach 0 after the last update. The training of the MoG was early-stopped and also showed no signs of overfitting.

The results are shown in Table 7. We report the average log-likelihood of each model for a million image patches from the test set. The ranking of RNADe models is maintained when ordered by validation likelihood: the model with best test-likelihood would have been chosen using cross-validation across all the RNADe models shown in the table. We also compared RNADe with a MoG trained by Zoran and Weiss (downloaded from Daniel Zoran's website) from which we removed the 64th row and column of each covariance matrix. There are two differences in the set-up of our experiments and those of Zoran and Weiss. First, we learned the means of the MoG components, while Zoran and Weiss (2011) fixed them to zero. Second, we held-out 20 images from the training set to do early-stopping and hyperparameter optimisation, while they used the 200 images for training.

The RNADe-FV model with fixed conditional variances obtained very low statistical performance. Adding an output parameter per dimension to have variable standard deviations made our models competitive with MoG with 100 full-covariance components. However, in order to obtain results superior to the mixture of Gaussians model trained by Zoran and Weiss, we had to use richer conditional distributions: one-dimensional mixtures of Gaussians (RNADe-MoG). On average, the best RNADe model obtained 3.3 nats per patch higher log-density than a MoG fitted with the same training data.

In Figure 9, we show one hundred examples from the test set, one hundred examples from Zoran and Weiss' mixture of Gaussians, and a hundred samples from our best RNADe-MoG model. Similar patterns can be observed in the three cases: uniform patches, edges, and locally smooth noisy patches.

7.3.3 SPEECH ACOUSTICS

We also measured the ability of RNADe to model small patches of speech spectrograms, extracted from the TIMIT data set (Garofolo et al., 1993). The patches contained 11 frames of 20 filter-banks plus energy; totalling 231 dimensions per data point. A good generative model of speech acoustics could be used, for example, in denoising, or speech detection tasks.

⁵ Empirically, we found this to work better than regular gradients and also better than multiplying by the variances, which would provide a step with the right units.

Model	Test log-likelihood
MoG $K=200$ (Zoran and Weiss, 2012) ^a	152.8
MoG $K=100$	144.7
MoG $K=200$	150.4
MoG $K=300$	150.4
RNADe-FV $h=512$	100.3
RNADe-Gaussian $h=512$	143.9
RNADe-Laplace $h=512$	145.9
RNADe-SAS ^b $h=512$	148.5
RNADe-MoG $K=2$ $h=512$	149.5
RNADe-MoG $K=2$ $h=1024$	150.3
RNADe-MoG $K=5$ $h=512$	152.4
RNADe-MoG $K=5$ $h=1024$	152.7
RNADe-MoG $K=10$ $h=512$	153.5
RNADe-MoG $K=10$ $h=1024$	153.7
RNADe-MoL $K=2$ $h=512$	149.3
RNADe-MoL $K=2$ $h=1024$	150.1
RNADe-MoL $K=5$ $h=512$	151.5
RNADe-MoL $K=5$ $h=1024$	151.4
RNADe-MoL $K=10$ $h=512$	152.3
RNADe-MoL $K=10$ $h=1024$	152.5

Table 7: Average per-example log-likelihood of several mixture of Gaussian and RNADe models on 8×8 pixel patches of natural images. These results are reported in nats and were calculated using one million patches. Standard errors due to the finite test sample size are lower than 0.1 nats in every case. h indicates the number of hidden units in the RNADe models, and K the number of one-dimensional components for each conditional in RNADe or the number of full-covariance components for MoG.

^a This model was trained using the full 200 images in the BSDS training data set, the rest of the models were trained using 180, reserving 20 for hyperparameter cross-validation and early-stopping.

^b Training an RNADe with sinh-arcsinh conditionals required the use of a starting learning rate 20 times smaller to avoid divergence during training. For this reason, this model was trained for 2000 epochs.

Model	Test LogL
MoG $N=50$	110.4
MoG $N=100$	112.0
MoG $N=200$	112.5
MoG $N=300$	112.5
RNADE-Gaussian	110.6
RNADE-Laplace	108.6
RNADE-SAS	119.2
RNADE-MoG $K=2$	121.1
RNADE-MoG $K=5$	124.3
RNADE-MoG $K=10$	127.8
RNADE-MoL $K=2$	116.3
RNADE-MoL $K=5$	120.5
RNADE-MoL $K=10$	123.3

Table 8: Log-likelihood of several MoG and RNADE models on the core-test set of TIMIT measured in nats. Standard errors due to the finite test sample size are lower than 0.4 nats in every case. RNADE obtained a higher (better) log-likelihood.

We fitted the models using the standard TIMIT training subset, which includes recordings from 605 speakers of American English. We compare RNADE with a mixture of Gaussians by measuring their log-likelihood on the complete TIMIT core-test data set: a held-out set of 25 speakers.

The RNADE models have 512 hidden units, ReLU activations, and a mixture of 20 one-dimensional Gaussian components per output. Given the large scale of this data set, hyperparameter choices were again made manually using validation data. The same training procedures for RNADE and mixture of Gaussians were used as for natural image patches.

The RNADE models were trained by stochastic gradient descent, with 25 data points per minibatch, for a total of 200 epochs, each comprising 1,000 minibatches. The learning rate was initialized to 0.001 and linearly decreased to reach 0 after the last epoch. Gradient momentum with momentum factor 0.9 was used, but initiated after the first epoch. A weight decay rate of 0.001 was applied to the input-to-hidden weight matrix only. Again, we found that multiplying the gradient of the mean output parameters by the standard deviation improved results. RNADE training was early stopped but didn't show signs of overfitting. As for the MoG model, it was trained exactly as in Section 7.3.2.

The results are shown in Table 8. The best RNADE (which would have been selected based on validation results) has 15 nats higher likelihood per test example than the best mixture of Gaussians. Examples from the test set, and samples from the MoG and RNADE-MoG models are shown in Figure 10. In contrast with the log-likelihood measure, there are no marked differences between the samples from each model. Both sets of samples look like blurred spectrograms, but RNADE seems to capture sharper formant structures (peaks of energy at the lower frequency bands characteristic of vowel sounds).

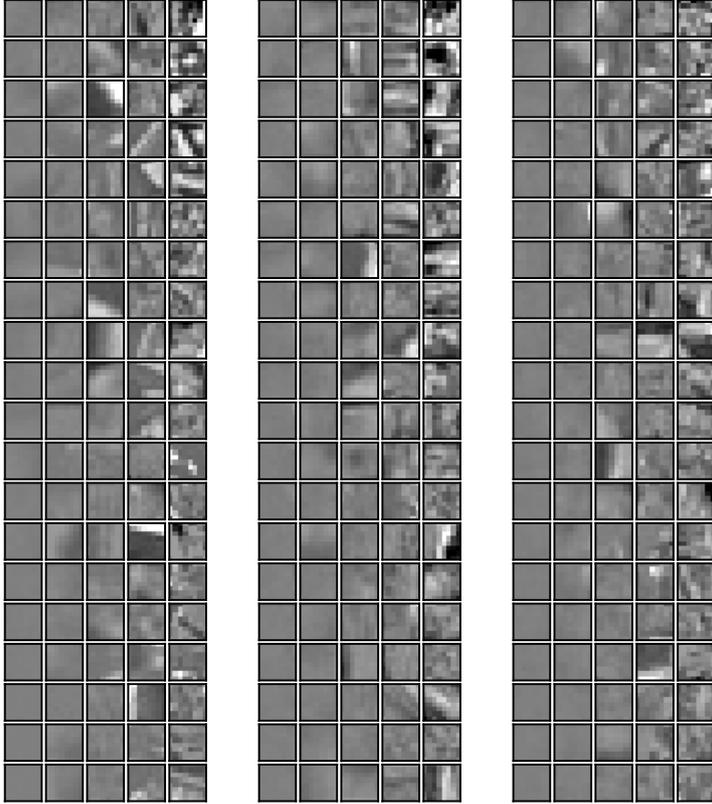


Figure 9: **Top:** 100 8×8 patches from the BSDS test set. **Center:** 100 samples from a mixture of Gaussians with 200 full-covariance components. **Bottom:** 100 samples from an RNADE with 1024 hidden units and 10 Gaussian components per conditional. All data and samples were drawn randomly and sorted by their density under the RNADE.

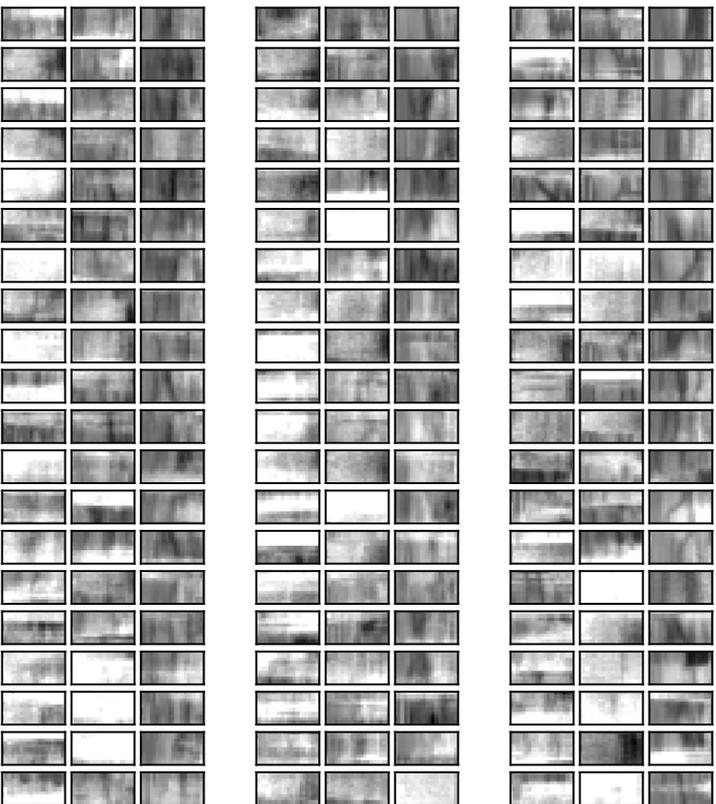


Figure 10: **Top:** 60 data points from the TIMT core-test set. **Center:** 60 samples from a MoG model with 200 components. **Bottom:** 60 samples from an RNADe with 10 Gaussian output components per dimension. For each data point displayed, time is shown on the horizontal axis, the bottom row displays the energy feature, while the others display the Mel filter bank features (in ascending frequency order from the bottom). All data and samples were drawn randomly and sorted by density under the RNADe model.

We’ve described the Neural Autoregressive Distribution Estimator, a tractable, flexible and competitive alternative to directed and undirected graphical models for unsupervised distribution estimation.

Since the publication of the first formulation of NADE (Larochelle and Murray, 2011), it has been extended to many more settings, other than those described in this paper. Larochelle and Lairly (2012); Zheng et al. (2015b) adapted NADE for topic modeling of documents and images, while Boulanger-Lewandowski et al. (2012) used NADE for modeling music sequential data. Theis and Bethge (2015) and Oord et al. (2016) proposed different NADE models for images than the one we presented, applied to natural images and based on convolutional and LSTM hidden units. Zheng et al. (2015a) used a NADE model to integrate an attention mechanism into an image classifier. Bornschein and Bengio (2015) showed that NADE could serve as a powerful prior over the latent state of directed graphical model. These are just a few examples of many possible ways one can leverage the flexibility and effectiveness of NADE models.

References

- Kevin Baehre and Moshé Lichman. UCI machine learning repository, 2013. <http://archive.ics.uci.edu/ml>.
- Yoshua Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.
- Yoshua Bengio and Samy Bengio. Modeling high-dimensional discrete data with multi-layer neural networks. In *Advances in Neural Information Processing Systems 12*, pages 400–406. MIT Press, 2000.
- Julian Besag. Statistical analysis of non-lattice data. *The Statistician*, 24(3):179–195, 1975.
- Christopher M. Bishop. Mixture density networks. Technical Report NCRG 4288, Neural Computing Research Group, Aston University, Birmingham, 1994.
- Jörg Bornschein and Yoshua Bengio. Reweighted wake-sleep. In *Proceedings of the 3rd International Conference on Learning Representations*. arXiv:1406.2751, 2015.
- Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1159–1166. Omnipress, 2012.
- Yuri Burda, Ruslan Salakhutdinov, and Roger Grosse. Importance weighted autoencoders. In *Proceedings of the 4th International Conference on Learning Representations*. arXiv:1509.00519v3, 2016.
- KyungHyun Cho, Tapas Rajko, and Alexander Ilin. Parallel tempering is efficient for learning restricted Boltzmann machines. In *Proceedings of the International Joint Conference on Neural Networks*. IEEE, 2010.

- KyungHyun Cho, Tapani Raiko, and Alexander Ilin. Enhanced gradient for training restricted Boltzmann machines. *Neural Computation*, 25:805–831, 2013.
- C.K. Chow and C.N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.
- George E. Dahl, Tara N. Sainath, and Geoffrey E. Hinton. Improving deep neural networks for LVCSR using rectified linear units and dropout. *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8609–8613, 2013.
- Peter Dayan, Geoffrey E. Hinton, Radford M. Neal, and Richard S. Zemel. The Helmholtz machine. *Neural Computation*, 7:889–904, 1995.
- Emily L. Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a Laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems 28*, pages 1486–1494. Curran Associates, Inc., 2015.
- Guillaume Desjardins, Aaron Courville, Yoshua Bengio, Pascal Vincent, and Olivier Delalleau. Tempered Markov chain Monte Carlo for training of restricted Boltzmann machine. *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, JMLR W&CP*, 9:145–152, 2010.
- Yoav Freund and David Haussler. Unsupervised learning of distributions on binary vectors using two layer networks. In *Advances in Neural Information Processing Systems 4*, pages 912–919. Morgan-Kaufmann, 1992.
- Brendan J. Frey, Geoffrey E. Hinton, and Peter Dayan. Does the wake-sleep algorithm learn good density estimators? In *Advances in Neural Information Processing Systems 8*, pages 661–670. MIT Press, 1996.
- J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, N. Dahlgren, and V. Zue. DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM. NIST, 1993.
- Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. MADE: Masked autoencoder for distribution estimation. *Proceedings of the 32nd International Conference on Machine Learning, JMLR W&CP*, 37:881–889, 2015.
- Zoubin Ghahramani and Geoffrey E. Hinton. The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, University of Toronto, 1996.
- Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pages 2672–2680, 2014.
- Alex Graves. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems 24*, pages 2348–2356. Curran Associates, Inc., 2011.
- Karol Gregor and Yann LeCun. Learning representations by maximizing compression. Technical report, arXiv:1108.1169, 2011.
- Karol Gregor, Andriy Mnih, and Daan Wierstra. Deep autoregressive networks. *Proceedings of the 31st International Conference on Machine Learning, JMLR W&CP*, 32:1242–1250, 2014.
- Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. DRAW: a recurrent neural network for image generation. *Proceedings of the 32nd International Conference on Machine Learning, JMLR W&CP*, 37:1462–1471, 2015.
- Arthur Gretton, Karsten M. Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J. Smola. A kernel method for the two-sample-problem. In *Advances in Neural Information Processing Systems 19*, pages 513–520. MIT Press, 2007.
- Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010.
- Stefan Harmeling and Christopher K.I. Williams. Greedy learning of binary latent trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(6):1087–1097, 2011.
- Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14:1771–1800, 2002.
- Geoffrey E. Hinton, Peter Dayan, Brendan J. Frey, and Radford M. Neal. The wake-sleep algorithm for unsupervised neural networks. *Science*, 268:1161–1158, 1995.
- Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006.
- Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.
- Aapo Hyvärinen. Some extensions of score matching. *Computational Statistics and Data Analysis*, 51:2499–2512, 2007a.
- Aapo Hyvärinen. Connections between score matching, contrastive divergence, and pseudo-likelihood for continuous-valued variables. *IEEE Transactions on Neural Networks*, 18:1529–1531, 2007b.
- Diederik P. Kingma and Jimmy Lei Ba. Adam: a method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*. arXiv:1412.6980v5, 2015.
- Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *Proceedings of the 2nd International Conference on Learning Representations*. arXiv:1312.6114v10, 2014.
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

- Hugo Larochelle and Stanislas Lauly. A neural autoregressive topic model. In *Advances in Neural Information Processing Systems 25*, pages 2708–2716. Curran Associates, Inc., 2012.
- Hugo Larochelle and Iain Murray. The neural autoregressive distribution estimator. *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, JMLR W&CP*, 15:29–37, 2011.
- Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- Yujia Li, Kevin Swersky, and Richard S. Zemel. Generative moment matching networks. *Proceedings of the 32nd International Conference on Machine Learning, JMLR W&CP*, 37:1718–1727, 2015.
- Banjamin Martin, Kevin Swersky, Bo Chen, and Nando de Freitas. Inductive principles for restricted Boltzmann machine learning. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010.
- D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *International Conference on Computer Vision*, volume 2, pages 416–423. IEEE, July 2001.
- Grégoire Montavon and Klaus-Robert Müller. Deep Boltzmann machines and the centering trick. In *Neural Networks: Tricks of the Trade, Second Edition*, pages 621–637. Springer, 2012.
- Radford M. Neal. Connectionist learning of belief networks. *Artificial Intelligence*, 56:71–113, 1992.
- Jiquan Ngiam, Zhenghao Chen, Pang Wei Koh, and Andrew Y. Ng. Learning deep energy models. In *Proceedings of the 28th International Conference on Machine Learning*, pages 1105–1112. Omnipress, 2011.
- Aïron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *Proceedings of the 33rd International Conference on Machine Learning, JMLR W&CP*, 2016. To appear. arXiv:1601.06759v2.
- Dirk Ormonoit and Volker Tresp. Improved Gaussian mixture density estimates using Bayesian penalty terms and network averaging. In *Advances in Neural Information Processing Systems 8*, pages 542–548. MIT Press, 1995.
- Tapani Raiko, Li Yao, Kyunghyun Cho, and Yoshua Bengio. Iterative neural autoregressive distribution estimator (NADE-k). In *Advances in Neural Information Processing Systems 27*, pages 325–333. Curran Associates, Inc., 2014.
- Daniio Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *Proceedings of the 31st International Conference on Machine Learning, JMLR W&CP*, 32:1278–1286, 2014.
- Ruslan Salakhutdinov. Learning in Markov random fields using tempered transitions. In *Advances in Neural Information Processing Systems 22*, pages 1598–1606. Curran Associates, Inc., 2009.
- Ruslan Salakhutdinov. Learning deep Boltzmann machines using adaptive MCMC. In *Proceedings of the 27th International Conference on Machine Learning*, pages 943–950. Omnipress, 2010.
- Ruslan Salakhutdinov and Geoffrey E. Hinton. Deep Boltzmann machines. *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, JMLR W&CP*, 5:448–455, 2009.
- Ruslan Salakhutdinov and Hugo Larochelle. Efficient learning of deep Boltzmann machines. *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, JMLR W&CP*, 9:693–700, 2010.
- Ruslan Salakhutdinov and Iain Murray. On the quantitative analysis of deep belief networks. In *Proceedings of the 25th International Conference on Machine Learning*, pages 872–879. Omnipress, 2008.
- Ricardo Silva, Charles Blundell, and Yee Whye Teh. Mixed cumulative distribution networks. *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, JMLR W&CP*, 15:670–678, 2011.
- Paul Smolensky. Information processing in dynamical systems: Foundations of harmony theory. In D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing: Volume 1: Foundations*, volume 1, chapter 6, pages 194–281. MIT Press, Cambridge, 1986.
- Paathraic Smyth and David Wolpert. Linearly combining density estimators via stacking. *Machine Learning*, 36(1-2):59–83, 1999.
- Jascha Sohl-Dickstein, Peter Battaglia, and Michael R. DeWeese. Minimum probability flow learning. In *Proceedings of the 28th International Conference on Machine Learning*, pages 905–912. Omnipress, 2011.
- Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: the all convolutional net. In *Proceedings of the 3rd International Conference on Learning Representations*. arXiv:1412.6806v3, 2015.
- Yichuan Tang, Ruslan Salakhutdinov, and Geoffrey E. Hinton. Deep mixtures of factor analysers. In *Proceedings of the 29th International Conference on Machine Learning*, pages 505–512. Omnipress, 2012.
- The Theano Development Team, Rami Al-Rfou, Guillaume Alain, Amjad Almahairi, Christof Angermueller, Dmitry Bahdanau, Nicolas Ballas, Frédéric Bastien, Justin Bayer, Anatoly Belikov, et al. Theano: A python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688*, 2016.

- Lucas Theis and Matthias Bethge. Generative image modeling using spatial lstms. In *Advances in Neural Information Processing Systems 28*, pages 1927–1935. Curran Associates, Inc., 2015.
- Tijmen Tieleman. Training restricted Boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1064–1071. Omnipress, 2008.
- Tijmen Tieleman and Geoffrey E. Hinton. Using fast weights to improve persistent contrastive divergence. In *Proceedings of the 26th International Conference on Machine Learning*, pages 1033–1040. Omnipress, 2009.
- Benigno Uria. *Connectionist multivariate density-estimation and its application to speech synthesis*. PhD thesis, The University of Edinburgh, 2015.
- Benigno Uria, Iain Murray, and Hugo Larochelle. RNADE: The real-valued neural autoregressive density-estimator. In *Advances in Neural Information Processing Systems 26*, pages 2175–2183. Curran Associates, Inc., 2013.
- Benigno Uria, Iain Murray, and Hugo Larochelle. A deep and tractable density estimator. *Proceedings of the 31st International Conference on Machine Learning, JMLR W&CP*, 32: 467–475, 2014.
- Jakob Verbeek. Mixture of factor analyzers Matlab implementation, 2005. <http://lear.inrialpes.fr/~verbeek/software.php>.
- Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1096–1103. Omnipress, 2008.
- Max Welling, Michal Rosen-Zvi, and Geoffrey E. Hinton. Exponential family harmoniums with an application to information retrieval. In *Advances in Neural Information Processing Systems 17*, pages 1481–1488. MIT Press, 2005.
- Laurent Younes. Parameter inference for imperfectly observed Gibbsian fields. *Probability Theory Related Fields*, 82:625–645, 1989.
- Yin Zheng, Richard S. Zemel, Yu-Jin Zhang, and Hugo Larochelle. A neural autoregressive approach to attention-based recognition. *International Journal of Computer Vision*, 113(1):67–79, 2015a.
- Yin Zheng, Yu-Jin Zhang, and Hugo Larochelle. A deep and autoregressive approach for topic modeling of multimodal data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(6):1056–1069, 2015b.
- Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *International Conference on Computer Vision*, pages 479–486. IEEE, 2011.
- Daniel Zoran and Yair Weiss. Natural images, Gaussian mixtures and dead leaves. In *Advances in Neural Information Processing Systems 25*, pages 1745–1753. Curran Associates, Inc., 2012.

ERRATA

On the Estimation of the Gradient Lines of a Density and the Consistency of the Mean-Shift Algorithm

Ery Arias-Castro

*Department of Mathematics
University of California, San Diego
La Jolla, CA 92093, USA*

EARIASCA@MATH.UCSB.EDU

David Mason

*Department of Applied Economics and Statistics
University of Delaware
Newark, DE 19717, USA*

DAVIDM@UDEL.EDU

Bruno Pelletier

*Département de Mathématiques
IRMAR – UMR CNRS 6625
Université Rennes II, France*

BRUNO.PELLETIER@UNIV-RENNES2.FR

Editor: Kevin Murphy

Our recent work (Arias-Castro et al., 2016) established the convergence of the mean shift algorithm under relatively general conditions. After the publication of this article, Prof. Jose E. Chacón — who has worked on the topic (Chacón and Monfort, 2013; Chacón and Duong, 2013) — alerted us of a mistake in the proof of the first part of our Theorem 1. The mistake is in the display following Eq. (32), where we applied the triangle inequality to the *squared* Euclidean distance, which of course is incorrect in general.

It turns out that the mistake has a simple and short fix, which we detail below. This relatively minor mistake would not warrant an errata, except that the same mistake has been made before by others also working on the convergence of the mean shift algorithm, including Comaniciu and Meer (2002), as revealed in (Li et al., 2007; Ghassabel, 2015). (Prof. Chacón also provided these last two references.)

Below is a slightly modified statement of our Theorem 1 with the additional assumption that the end point x^* of the flow line is an isolated local maximum, meaning that for all $\epsilon > 0$ small enough $B(x^*, \epsilon)$ contains no local maximum other than x^* . Since f is also assumed of class C^3 , this is equivalent to assuming that for all $\epsilon > 0$ small enough, $B(x^*, \epsilon)$ contains only one critical point of f , namely x^* , and $f(x^*) > f(x)$ for all $x \in B(x^*, \epsilon)$ such that $x \neq x^*$.

The equations numbered (xx) refers to the original paper, while equations numbered (R.xx) are new to this note.

Theorem 1 *Let f be a function of class C^3 . Let $(x(t) : t \geq 0)$ denote the flow line of f starting at x_0 and ending at an isolated local maximum x^* of f . Let (x_ℓ) be the sequence defined in (6) starting at x_0 . Then there exists $A = A(x_0, f) > 0$ such that, whenever*

$0 < a < A$,

$$\lim_{\ell \rightarrow +\infty} x_\ell = x^*.$$

Denote by $x_a(t)$ the following polygonal line

$$x_a(t) = x_{\ell-1} + (t/a - \ell + 1)(x_\ell - x_{\ell-1}), \quad \forall t \in [(\ell - 1)a, \ell a).$$

Assume $H_f(x^*)$ has all eigenvalues in $(-\underline{\nu}, -\underline{\nu})$ for some $0 < \underline{\nu} < \bar{\nu}$. Then, there exists a $C = C(x_0, f, \underline{\nu}, \bar{\nu}) > 0$ such that, for any $0 < a < A$,

$$\sup_{t \geq 0} \|x_a(t) - x(t)\| \leq C a^\delta, \quad \delta := \frac{\underline{\nu}}{\underline{\nu} + \bar{\nu}}.$$

The rest of this note is dedicated to a corrected proof of the first part of this theorem. Claims 1 and 2 refer to the first two claims in the original published proof of Theorem 1. (Note that in the proof of Claim 1, we can assume without loss of generality that $f(x) > 0$ over $B(x_0, 3r_0)$.) Also notice that x_0 is not a global minimum of f since x_0 is not a critical point of f and f is C^3 . In the published proof of Theorem 1, the second claim on page 15 should be removed.

Now replace the claim on page 16 by the following definition, observation and Claims A, B and C:

For any $\eta > 0$, denote by $\mathcal{C}(\eta)$ the connected component of $\mathcal{L}_f(f(x^*) - \eta)$ that contains x^* . Notice that since x^* is a local maximum, for all $\eta > 0$ small enough

$$\mathcal{C}(\eta) = \mathcal{C}(x^*, \eta), \tag{R.1}$$

where $\mathcal{C}(x^*, \eta)$ be the connected component of

$$\{y : f(x^*) - \eta \leq f(y) \leq f(x^*)\}$$

that contains x^* .

Claim A. *Let y^* be such that $f(y^*) = f(x^*)$, but $y^* \neq x^*$. For all $\eta > 0$ small enough $y^* \notin \mathcal{C}(x^*, \eta)$. Choose such a y^* . Since x^* is an isolated local maximum, for all $\epsilon > 0$ small enough, $y^* \notin \bar{B}(x^*, \epsilon)$ and for some $\eta_\epsilon > 0$, $f(y) < f(x^*) - \eta_\epsilon$ for all $y \in \bar{B}(x^*, \epsilon) - B(x^*, \epsilon/2)$. Note that $\eta_\epsilon > 0$ can be chosen as small as desired by choosing $\epsilon > 0$ small enough. Suppose $y^* \in \mathcal{C}(x^*, \eta_\epsilon)$, then since $\mathcal{C}(x^*, \eta_\epsilon)$ is connected and $x^* \in \mathcal{C}(x^*, \eta_\epsilon)$ there is a continuous path lying inside $\mathcal{C}(x^*, \eta_\epsilon)$ joining x^* and y^* . Such a path would have to pass through a point $y \in \mathcal{C}(x^*, \eta_\epsilon) \cap (\bar{B}(x^*, \epsilon) - B(x^*, \epsilon/2))$ for which $f(y) < f(x^*) - \eta_\epsilon$. This cannot happen, since $y \in \mathcal{C}(x^*, \eta_\epsilon)$ forces $f(x^*) - \eta_\epsilon \leq f(y)$. Hence for all for all $\eta > 0$ small enough $y^* \notin \mathcal{C}(x^*, \eta)$.*

Claim B. *For all $\eta > 0$ small enough $\mathcal{C}(x^*, \eta)$ contains only one critical point of f . Towards proving this we shall first show that for all $\epsilon > 0$ there exists an $\eta > 0$ such that*

$$\mathcal{C}(x^*, \eta) \subset \bar{B}(x^*, \epsilon). \tag{R.2}$$

To see this, for any $\eta > 0$ small, denote the contour set

$$c(\eta) = \{y : f(y) = f(x^*) - \eta, y \in \mathcal{C}(x^*, \eta)\}.$$

Note that each contour set $c(\eta)$ is closed and hence is compact. (To see this, by Claim 1, we may assume that $\mathcal{L}(f(x_0))$ is bounded and thus compact. Therefore, since $c(\eta) \subset \mathcal{C}(x^*, \eta) \subset \mathcal{L}(f(x_0))$, $c(\eta)$ is compact.) Since $c(2^{-k})$ is compact, for each $k \geq 1$ large enough there exists a $x_k \in c(2^{-k})$ such that

$$r_k := \sup \left\{ \|y - x^*\| : y \in c(2^{-k}) \right\} = \|x_k - x^*\|.$$

Observe that since $f(x_k) \rightarrow f(x^*)$ and x^* is isolated, necessarily by a compactness argument, $x_k \rightarrow x^*$. To see this, suppose that for some subsequence y_j of x_k we have $y_j \rightarrow y^*$. Necessarily $f(y^*) = f(x^*)$ and $y^* \in \mathcal{C}(x^*, \eta)$ for all $\eta > 0$. However by Claim A, necessarily $y^* = x^*$. Thus $x_k \rightarrow x^*$, which implies $r_k \rightarrow 0$. Noting that for all large k

$$\mathcal{C}(x^*, 2^{-k}) \subset \overline{B}(x^*, r_k),$$

we see that for all $\epsilon > 0$ there exists an $\eta > 0$ such that (R.2) holds. Therefore, since for all small enough $\epsilon > 0$, $\overline{B}(x^*, \epsilon)$ contains only one critical point of f , we get that for all $\eta > 0$ small enough $\mathcal{C}(x^*, \eta)$ contains only one critical point of f .

Claim C. (x_ℓ) converges to x^* . By Claim B and (R.1), there exists $\eta_0 > 0$ small enough such that $\mathcal{C}(\eta_0)$ contains no critical point of f other than x^* . Moreover since f is C^3 , there exists $\epsilon > 0$ such that $\overline{B}(x^*, \epsilon) \subset \mathcal{C}(\eta_0)$. Let ℓ_ϵ be such that $\|x(t_{\ell_\epsilon}) - x^*\| \leq \epsilon/2$ and let a_ϵ be such that

$$\left[e^{\ell_\epsilon a_\epsilon \kappa_2 \sqrt{a}} - 1 \right] \kappa_1 a_\epsilon = \epsilon/2.$$

Assume now that $a \leq A_1 \wedge a_\epsilon$, where A_1 is defined in (31). Then, by (33) and the triangle inequality,

$$\|x_{t_\ell} - x^*\| \leq \|x_{t_\ell} - x(t_{t_\ell})\| + \|x(t_{t_\ell}) - x^*\| \leq \epsilon.$$

Thus, x_{t_ℓ} belongs to $\overline{B}(x^*, \epsilon)$, and so to $\mathcal{C}(\eta_0)$. By Claim 2 the values of f are increasing along the polygonal curve x_{t_ℓ} , so x_{t_ℓ} belongs to $\mathcal{C}(\eta_0)$ for all $\ell \geq \ell_\epsilon$.

Since the sequence $(f(x_\ell)) : \ell \geq 0$ is increasing and bounded, it is convergent and since

$$f(x_{t+1}) - f(x_t) \geq \frac{\theta}{2} \|\nabla f(x_\ell)\|^2,$$

we deduce that

$$\lim_{\ell \rightarrow \infty} \|\nabla f(x_\ell)\| = 0.$$

Recall that by Claim 1 we can assume that $\mathcal{L}(f(x_0))$ is bounded in which case $\mathcal{C}(\eta_0)$ is compact. Then we conclude that (x_ℓ) is convergent with $x_\ell \rightarrow x^*$ by continuity of the gradient of f and the fact that x^* is the only critical point of f in $\mathcal{C}(\eta_0)$.

Acknowledgments

We are grateful to Jose E. Chacón for alerting us of the mistake and for referring us to the articles (Li et al., 2007; Ghassabeh, 2015).

References

- Ery Arias-Castro, David Mason, and Bruno Pelletier. On the estimation of the gradient lines of a density and the consistency of the mean-shift algorithm. *Journal of Machine Learning Research*, 17(43):1–28, 2016.
- José E Chacón and Tam Duong. Data-driven density derivative estimation, with applications to nonparametric clustering and bump hunting. *Electronic Journal of Statistics*, 7: 499–532, 2013.
- José E Chacón and Pablo Monfort. A comparison of bandwidth selectors for mean shift clustering. *arXiv preprint arXiv:1310.7855*, 2013.
- D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):1–18, 2002.
- Youness Aliyari Ghassabeh. A sufficient condition for the convergence of the mean shift algorithm with gaussian kernel. *Journal of Multivariate Analysis*, 135:1–10, 2015.
- Xiangru Li, Zhaoyi Hu, and Fuchao Wu. A note on the convergence of the mean shift. *Pattern Recognition*, 40(6):1756–1762, 2007.

Modelling Interactions in High-dimensional Data with Backtracking

Rajen D. Shah

Statistical Laboratory

University of Cambridge

Cambridge, CB3 0WB, UK

R.SHAH@STATSLAB.CAM.AC.UK

Editor: Sara van de Geer

Abstract

We study the problem of high-dimensional regression when there may be interacting variables. Approaches using sparsity-inducing penalty functions such as the Lasso can be useful for producing interpretable models. However, when the number of variables runs into the thousands, and so even two-way interactions number in the millions, these methods may become computationally infeasible. Typically variable screening based on model fits using only main effects must be performed first. One problem with screening is that important variables may be missed if they are only useful for prediction when certain interaction terms are also present in the model.

To tackle this issue, we introduce a new method we call Backtracking. It can be incorporated into many existing high-dimensional methods based on penalty functions, and works by building increasing sets of candidate interactions iteratively. Models fitted on the main effects and interactions selected early on in this process guide the selection of future interactions. By also making use of previous fits for computation, as well as performing calculations in parallel, the overall run-time of the algorithm can be greatly reduced.

The effectiveness of our method when applied to regression and classification problems is demonstrated on simulated and real data sets. In the case of using Backtracking with the Lasso, we also give some theoretical support for our procedure.

Keywords: high-dimensional data, interactions, Lasso, path algorithm

1. Introduction

In recent years, there has been a lot of progress in the field of high-dimensional regression. Much of the development has centred around the Lasso (Tibshirani, 1996), which given a vector of responses $\mathbf{Y} \in \mathbb{R}^n$ and design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$, solves

$$(\hat{\mu}, \hat{\beta}) := \arg \min_{(\mu, \beta) \in \mathbb{R} \times \mathbb{R}^p} \left\{ \frac{1}{2n} \|\mathbf{Y} - \mu \mathbf{1} - \mathbf{X}\beta\|_2^2 + \lambda \|\beta\|_1 \right\}, \quad (1)$$

where $\mathbf{1}$ is an n -vector of ones and the regularisation parameter λ controls the relative contribution of the penalty term to the objective. The many extensions of the Lasso allow most familiar models from classical (low-dimensional) statistics to now be fitted in situations where the number of variables p may be tens of thousands and even greatly exceed the number of observations n (see the monograph Bühlmann and van de Geer (2011b) and references therein).

However, despite the advances, fitting models with interactions remains a challenge. Two issues that arise are:

- (i) Since there are $p(p-1)/2$ possible first-order interactions, the main effects can be swamped by the vastly more numerous interaction terms and without proper regularisation, stand little chance of being selected in the final model (see Figure 1b).
- (ii) Monitoring the coefficients of all the interaction terms quickly becomes infeasible as p runs into the thousands.

1.1 Related Work

For situations where $p < 1000$ or thereabouts and the case of two-way interactions, a lot of work has been done in recent years to address this need. To tackle (i), many of the proposals use penalty functions and constraints designed to enforce that if an interaction term is in the fitted model, one or both main effects are also present (Lin and Zhang, 2006; Zhao et al., 2009; Yuan et al., 2009; Radchenko and James, 2010; Jenatton et al., 2011; Bach et al., 2012a,b; Bien et al., 2013; Lim and Hastie, 2015; Haris et al., 2015). See also Turlach (2004) and Yuan et al. (2007), which consider modifications of the LAR algorithm Efron et al. (2004) that impose this type of condition.

In the moderate-dimensional setting that these methods are designed for, the computational issue (ii) is just about manageable. However, when p is larger—the situation of interest in this paper—it typically becomes necessary to narrow the search for interactions. Comparatively little work has been done on fitting models with interactions to data of this sort of dimension. An exception is the method of Random Intersection Trees (Shah and Meinshausen, 2014), which does not explicitly restrict the search space of interactions. However this is designed for a classification setting with a binary predictor matrix and does not fit a model but rather tries to find interactions that are marginally informative.

One option is to screen for important variables and only consider interactions involving the selected set. Wu et al. (2010) and others take this approach: the Lasso is first used to select main effects; then interactions between the selected main effects are added to the design matrix, and the Lasso is run once more to give the final model.

The success of this method relies on all main effects involved in interactions being selected in the initial screening stage. However, this may well not happen. Certain interactions may need to be included in the model before some main effects can be selected. To address this issue, Bickel et al. (2010) propose a procedure involving sequential Lasso fits which, for some predefined number K , selects K variables from each fit and then adds all interactions between those variables as candidate variables for the following fit. The process continues until all interactions to be added are already present. However, it is not clear how one should choose K : a large K may result in a large number of spurious interactions being added at each stage, whereas a small K could cause the procedure to terminate before it has had a chance to include important interactions.

Rather than adding interactions in one or more distinct stages, when variables are selected in a greedy fashion, the set of candidate interactions can be updated after each selection. This dynamic updating of interactions available for selection is present in the popular MARS procedure of Friedman (1991). One potential problem with this approach

is that particularly in high-dimensional situations, overly greedy selection can sometimes produce unstable final models and predictive performance can suffer as a consequence.

The iFORT method of Hao and Zhang (2014) applies forward selection to a dynamically updated set of candidate interactions and main effects, for the purposes of variable screening. In this work, we propose a new method we call Backtracking, for incorporating a similar model building strategy to that of MARS and iFORT into methods based on sparsity-inducing penalty functions. Though greedy forward selection methods often work well, penalty function-based methods such as the Lasso can be more stable (see Efron et al. (2004)) and offer a useful alternative.

1.2 Outline of the Idea

When used with the Lasso, Backtracking begins by computing the Lasso solution path, decreasing λ from ∞ . A second solution path, P_2 , is then produced, where the design matrix contains all main effects, and also the interaction between the first two active variables in the initial path. Continuing iteratively, subsequent solution paths P_3, \dots, P_T are computed where the set of main effects and interactions in the design matrix for the k th path is determined based on the previous path P_{k-1} . Thus if in the third path, a key interaction was included and so variable selection was then more accurate, the selection of interactions for all future paths would benefit. In this way information is used as soon as it is available, rather than at discrete stages as with the method of Bickel et al. (2010). In addition, if all important interactions have already been included by P_3 , we have a solution path unhindered by the addition of further spurious interactions.

It may seem that a drawback of our proposed approach is that the computational cost of producing all T solution paths will usually be unacceptably large. However, computation of the full collection of solution paths is typically very fast. This is because rather than computing each of the solution paths from scratch, for each new solution path P_{k+1} , we first track along the previous path P_k to find where P_{k+1} departs from P_k . This is the origin of the name Backtracking. Typically, checking whether a given trial solution is on a solution path requires much less computation than calculating the solution path itself, and so this Backtracking step is rather quick. Furthermore, when the solution paths do separate, the tail portions of the paths can be computed in parallel.

An R (R Development Core Team, 2005) package for the method is available on the author's website.

1.3 Organisation of the Paper

The rest of the paper is organised as follows. In Section 2 we describe an example which provides some motivation for our Backtracking method. In Section 3 we develop our method in the context of the Lasso for the linear model. In Section 4, we describe how our method can be extended beyond the case of the Lasso for the linear model. In Section 5 we report the results of some simulation experiments and real data analyses that demonstrate the effectiveness of Backtracking. Finally, in Section 6, we present some theoretical results which aim to give a deeper understanding of the way in which Backtracking works. Proofs are collected in the appendix.

2. Motivation

In this section we introduce a toy example where approaches that select candidate interactions based on selected main effects will tend to perform poorly. We consider a linear model with interactions involving a design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ with $n = 200$, $p = 500$ and where

$$Y_i = \sum_{j=1}^6 \beta_j X_{ij} + \beta_7 X_{i1} X_{i2} + \beta_8 X_{i3} X_{i4} + \beta_9 X_{i5} X_{i6} + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2), \quad i = 1, \dots, n. \quad (2)$$

We take \mathbf{X} with i.i.d. rows having a distribution such that X_{i5} is uncorrelated with $\{X_{ij} : j \neq 5\}$. We then choose β_1, \dots, β_9 in such a way that X_{i5} is also uncorrelated with the response yet $\beta_5 \neq 0$. The precise construction is detailed in the appendix.

In order to select variable 5 using that Lasso, we would need to have already selected some important interactions. Thus if we first select important main effects using the Lasso, for example, it is very unlikely that variable 5 will be selected. Then if we add all two-way interactions between the selected variables and fit the Lasso once more, the interaction between variables 5 and 6 will not be included. Of course, one can again add interactions between selected variables and compute another Lasso fit, and then there is a chance the interaction will be selected. Thus it is very likely that at least three Lasso fits will be needed in order to select the right variables.

Figure 1a shows the result of applying the Lasso to data generated according to (2), σ chosen to give a signal-to-noise ratio (SNR) of 4, and

$$\boldsymbol{\beta} = (-1.25, -0.75, 0.75, -0.5, -2, 1.5, 2, 2, 1)^T.$$

As expected, we see variable 5 is nowhere to be seen and instead many unwanted variables are selected as λ is decreased. Figure 1b illustrates the effect of including all $p(p-1)/2$ possible interactions in the design matrix. Even in our rather moderate-dimensional situation, we are not able to recover the true signal. Though all the true interaction terms are selected, now neither variable 4 nor variable 5 are present in the solution paths and many false interactions are selected.

Although this example is rather contrived, it illustrates how sometimes the right interactions need to be augmented to the design matrix in order for certain variables to be selected. Even when interactions are only present if the corresponding main effects are too, main effects can be missed by a procedure that does not consider interactions. In fact, we can see the same phenomenon occurring when the design matrix has i.i.d. Gaussian entries (see Section 5.1). Thus multiple Lasso fits might be needed to have any chance of selecting the right model.

This raises the question of which tuning parameters to use in the multiple Lasso fits. One option, which we shall refer to as the iterated Lasso, is to select tuning parameters by cross-validation each time. A drawback of this approach, though, is that the number of interactions to add can be quite large if cross-validation chooses a large active set. This is often the case when the presence of interactions makes some important main effects hard to distinguish from noise variables in the initial Lasso fit. Then cross-validation may choose a low λ in order to try to select those variables, but this would result in many noise variables also being included in the active set.

We take an alternative approach here and include suspected interactions in the design matrix as soon as possible. That is, if we progress along the solution path from $\lambda = \infty$, and two variables enter the model, we immediately add their interaction to the design matrix and start computing the Lasso again. We could now disregard the original path, but there is little to lose, and possibly much to gain, in continuing the original path in parallel with the new one. We can then repeat this process, adding new interactions when necessary, and restarting the Lasso, whilst still continuing all previous paths in parallel. We show in the next section how computation can be made very fast since many of these solution paths will share the same initial portions.

3. Backtracking with the Lasso

In this section we introduce a version of the Backtracking algorithm applied to the Lasso (1). First, we present a naive version of the algorithm, which is easy to understand. Later in Section 3.2, we show that this algorithm performs a large number of unnecessary calculations, and we give a far more efficient version.

3.1 A Naive Algorithm

As well as a base regression procedure, the other key ingredient that Backtracking requires is a way of suggesting candidate interactions based on selected main effects, or more generally a way of suggesting higher-order interactions based on lower-order interactions. In order to discuss this and present our algorithm, we first introduce some notation concerning interactions.

Let \mathbf{X} be the original $n \times p$ design matrix, with no interactions. In order to consider interactions in our models, rather than indexing variables by a single number j , we use subsets of $\{1, \dots, p\}$. Thus by variable $\{1, 2\}$, we mean the interaction between variables 1 and 2, or in our new notation, variables $\{1\}$ and $\{2\}$. When referring to main effects $\{j\}$ however, we will often omit the braces. As we are using the Lasso as the base regression procedure here, interaction $\{1, 2\}$ will be the componentwise product of the first two columns of \mathbf{X} . We will write $\mathbf{X}_v \in \mathbb{R}^n$ for variable v .

The choice of whether and how to scale and centre interactions and main effects can be a rather delicate one, where domain knowledge may play a key role. In this work, we will centre all main effects, and scale them to have ℓ_2 -norm \sqrt{n} . The interactions will be created using these centred and scaled main effects, and they themselves will also be centred and scaled to have ℓ_2 -norm \sqrt{n} .

For C a set of subsets of $\{1, \dots, p\}$ we can form a modified design matrix \mathbf{X}_C , where the columns of \mathbf{X}_C are given by the variables in C , centred and scaled as described above. Thus C is the set of candidate variables available for selection when design matrix \mathbf{X}_C is used. This subsetting operation will always be taken to have been performed before any further operations on the matrix, so in particular \mathbf{X}_C^T means $(\mathbf{X}_C)^T$.

We will consider all associated vectors and matrices as indexed by variables, so we may speak of component $\{1, 2\}$ of β , denoted $\beta_{\{1,2\}}$, if β were multiplying a design matrix which included $\{1, 2\}$. Further, for any collection of variables A , we will write β_A for the subvector whose components are those indexed by A . To represent an arbitrary variable which may be an interaction, we shall often use v or u and reserve j to index main effects.

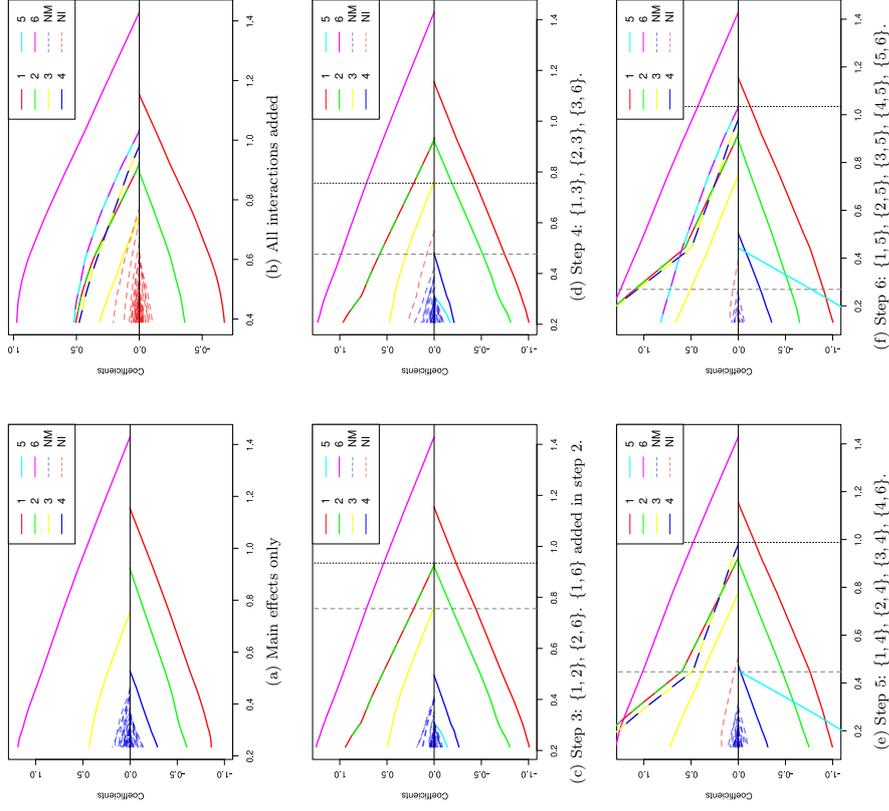


Figure 1: For data generated as described in Section 2, the coefficient paths against λ of the Lasso with main effects only, (a); the Lasso with all interactions added, (b); and Backtracking with $k = 3, \dots, 6$, ((c)-(d)); when applied to the example in Section 2. Below the Backtracking solution paths we give $C_k \setminus C_{k-1}$: the interactions which have been added in the current step. The solid red, green, yellow, blue, cyan and magenta lines trace the coefficients of variables 1, ..., 6 respectively, with the alternately coloured lines representing the corresponding interactions. The dotted blue and red coefficient paths indicate noise main effect ('NM') and interaction ('NI') terms respectively. Vertical dotted black and dashed grey lines give the values of λ_k^{start} and λ_k^{add} respectively.

We will often need to express the dependence of the Lasso solution $\hat{\beta}(1)$ on the tuning parameter λ and the design matrix used. We shall write $\hat{\beta}(\lambda, C)$ when \mathbf{X}_C is the design matrix. We will denote the set of active components of a solution β by $\mathcal{A}(\beta) = \{v : \hat{\beta}_v \neq 0\}$.

We now introduce a function \mathcal{I} that given a set of variables A , suggests a set of interactions to add to the design matrix. The choice of \mathcal{I} we use here is as follows:

$$\mathcal{I}(A) = \{v \subseteq \{1, \dots, p\} : \text{for all } u \subseteq v, u \neq \emptyset, u \in A\}.$$

In other words, $\mathcal{I}(A)$ is the set of variables not in A , all of whose corresponding lower order interactions are present in A . To ease notation, when A contains only main effects j_1, \dots, j_s , we will write $\mathcal{I}(j_1, \dots, j_s) = \mathcal{I}(A)$. For example, $\mathcal{I}(1, 2) = \{\{1, 2\}\}$, and $\mathcal{I}(1, 2, 3) = \{\{1, 2\}, \{2, 3\}, \{1, 3\}\}$. Note $\{1, 2, 3\} \notin \mathcal{I}(1, 2, 3)$ as the lower order interaction $\{1, 2\}$ of $\{1, 2, 3\}$ is not in $\{\{1\}, \{2\}, \{3\}\}$; for example. Other choices for \mathcal{I} can be made, and we discuss some further possibilities in Section 4.

Backtracking relies on a path algorithm for computing the Lasso on a grid of λ values $\lambda_1 > \dots > \lambda_T$. Several algorithms are available and coordinate descent methods (Friedman et al., 2010) appears to work well in practice.

We are now in a position to introduce a naive version of our Backtracking algorithm applied to the Lasso (Algorithm 1). We will assume that the response \mathbf{Y} is centred in addition to the design matrix; so no intercept term is necessary.

Algorithm 1 A naive version of Backtracking with the Lasso

Set T to be the (given) maximum number of candidate interaction sets to generate. Let the initial candidate set consist of just main effects: $C_1 = \{\{1\}, \dots, \{p\}\}$. Set the index for the candidate sets $k = 1$. Let $\lambda_1^{\text{start}} = \lambda_1$, the largest λ value on the grid. In the steps which follow, we maintain a record of the set of variables which have been non-zero at any point in the algorithm up to the current point (an ‘‘ever active set’’, A).

1. Compute the solution path of the Lasso with candidate set C_k from λ_k^{start} onwards until the ever active set A has $\mathcal{I}(A) \not\subseteq C_k$ (if the smallest λ value on the grid is reached then go to 5). Let the λ value where this occurs be λ_k^{add} . We will refer to this solution path as P_k .
2. Set $C_{k+1} = C_k \cup \mathcal{I}(A)$ so the next candidate set contains all interactions between variables in the ever active set.
3. Set $\lambda_{k+1}^{\text{start}} = \lambda_1$.
4. Increment k . If $k > T$ go to 5, otherwise go back to 1.
5. For each k complete the solution path P_k by continuing it until $\lambda = \lambda_T$. Computing these final pieces of the solution paths can be done in parallel.

The algorithm computes Lasso solution paths whose corresponding design matrices include interactions chosen based on previous paths. The quantity λ_k^{add} records the value of λ at which interaction terms were added to the set of candidates C_k . Here λ_k^{start} is a redundant quantity and can be replaced everywhere with λ_1 to give the same algorithm.

We include it at this stage though to aid with the presentation of an improved version of the algorithm where λ_k^{start} in general takes values other than λ_1 . We note that the final step of completing the solution paths can be carried out as the initial paths are being created, rather than once all initial paths have been created. Though here the algorithm can include three-way or even higher order interactions; it is straightforward to restrict the possible interactions to be added to first-order interactions, for example.

3.2 An Improved Algorithm

The process of performing multiple Lasso fits is computationally cumbersome, and an immediate gain in efficiency can be realised by noticing that the final collection of solution paths is in fact a tree of solutions: many of the solution paths computed will share the same initial portions.

To discuss this, we first recall the KKT conditions for the Lasso dictate that $\hat{\beta}$ is a solution to (1) when the design matrix is \mathbf{X}_C if and only if

$$\frac{1}{n} \mathbf{X}_C^T (\mathbf{Y} - \mathbf{X}_C \hat{\beta}) = \lambda \text{sgn}(\hat{\beta}_v) \quad \text{for } \hat{\beta}_v \neq 0 \tag{3}$$

$$\frac{1}{n} |\mathbf{X}_C^T (\mathbf{Y} - \mathbf{X}_C \hat{\beta})| \leq \lambda \quad \text{for } \hat{\beta}_v = 0. \tag{4}$$

Note the $\beta \mathbf{X}_C^T \mathbf{1}$ term vanishes as the columns of \mathbf{X}_C are centred.

We see that if for some λ

$$\frac{1}{n} \|\mathbf{X}_{C_{k+1} \setminus C_k}^T (\mathbf{Y} - \mathbf{X}_{C_k} \hat{\beta}(\lambda, C_k))\|_\infty \leq \lambda, \tag{5}$$

then

$$\hat{\beta}_{C_{k+1} \setminus C_k}(\lambda, C_{k+1}) = 0, \quad \hat{\beta}_{C_k}(\lambda, C_{k+1}) = \hat{\beta}(\lambda, C_k).$$

Thus given solution path P_k , we can attempt to find the smallest λ such that (5) holds. Up to that point then, path P_{k+1} will coincide with P_k and so those Lasso solutions need not be re-computed. Note that verifying (5) is a computationally simple task requiring only $O(|C_{k+1} \setminus C_k|n)$ operations.

Our final Backtracking algorithm therefore replaces step 3 of Algorithm 1 with the following:

- 3a. Find the smallest $\lambda \geq \lambda_T \geq \lambda_k^{\text{add}}$ such that (5) holds with $\lambda = \lambda$ and set this to be $\lambda_{k+1}^{\text{start}}$. If no such λ exists, set $\lambda_{k+1}^{\text{start}}$ to be λ_1 .

Figures 1c–1f show steps 3–6 (i.e. $k = 3, \dots, 6$) of Backtracking applied to the example described in Section 2. Note that Figure 1a is in fact step 1. Step 2 is not shown as the plot looks identical to that in Figure 1a. We see that when $k = 6$, we have a solution path where all the true variable and interaction terms are active before any noise variables enter the coefficient plots.

We can further speed up the algorithm by first checking if P_k coincides with P_{k+1} at λ_k^{add} . If not, we can perform a bisection search to find any point where P_k and P_{k+1} agree, but after which they disagree. This avoids checking (5) for every λ up to λ_k^{add} . We will work with the simpler version of Backtracking here using step 3a, but use this faster version in our implementation.

4. Further Applications of Backtracking

Our Backtracking algorithm has been presented in the context of the Lasso for the linear model. However, the real power of the idea is that it can be incorporated into any method that produces a path of increasingly complex sparse solutions by solving a family of convex optimisation problems parametrised by a tuning parameter. For the Backtracking step, the KKT conditions for these optimisation problems provide a way of checking whether a given trial solution is an optimum. As in the case of the Lasso, checking whether the KKT conditions are satisfied typically requires much less computational effort than computing a solution from scratch. Below we briefly sketch some applications of Backtracking to a few of the many possible methods with which it can be used.

4.1 Multinomial Regression

An example, which we apply to real data in Section 5.2, is multinomial regression with a group Lasso (Yuan and Lin, 2006) penalty. Consider m observations of a categorical response that takes J levels, and p associated covariates. Let \mathbf{Y} be the indicator response matrix, with ij th entry equal to 1 if the i th observation takes the j th level, and 0 otherwise. We model

$$\mathbb{P}(Y_{ij} = 1) := \Pi_{ij}(\boldsymbol{\mu}^*, \boldsymbol{\beta}^*; \mathbf{X}_{S^*}) := \frac{\exp(\mu_j^* + (\mathbf{X}_{S^*} \boldsymbol{\beta}_j^*)_i)}{\sum_{j'=1}^J \exp(\mu_{j'}^* + (\mathbf{X}_{S^*} \boldsymbol{\beta}_{j'}^*)_i)}.$$

Here $\boldsymbol{\mu}^*$ is a vector of intercept terms and $\boldsymbol{\beta}^*$ is a $|S^*| \times J$ matrix of coefficients; $\boldsymbol{\beta}_j^*$ denotes the j th column of $\boldsymbol{\beta}^*$. This model is over-parametrised, but regularisation still allows us to produce estimates of $\boldsymbol{\mu}^*$ and $\boldsymbol{\beta}^*$ and hence also of $\boldsymbol{\Pi}$ (see Friedman et al. (2010)). When our design matrix is \mathbf{X}_C , these estimates are given by $(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\beta}}) := \arg \min_{\boldsymbol{\mu}, \boldsymbol{\beta}} Q(\boldsymbol{\mu}, \boldsymbol{\beta}; \lambda)$ where

$$Q(\boldsymbol{\mu}, \boldsymbol{\beta}; \lambda) := \frac{1}{n} \sum_{j=1}^J \mathbf{Y}_j^T (\boldsymbol{\mu}_j \mathbf{1} + \mathbf{X}_C \boldsymbol{\beta}_j) - \frac{1}{n} \mathbf{1}^T \log \left(\sum_{j=1}^J \exp(\boldsymbol{\mu}_j \mathbf{1} + \mathbf{X}_C \boldsymbol{\beta}_j) \right) + \lambda \sum_{v \in C} \|(\boldsymbol{\beta}^T)_v\|_2.$$

The functions log and exp are to be understood as applied componentwise and the rows of $\boldsymbol{\beta}$ are indexed by elements of C . To derive the Backtracking step for this situation, we turn to the KKT conditions which characterise the minima of Q :

$$\begin{aligned} \frac{1}{n} (\mathbf{Y}^T - \boldsymbol{\Pi}^T(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\beta}}; \mathbf{X}_C)) \mathbf{1} &= \mathbf{0}, \\ \frac{1}{n} \{ \mathbf{Y}^T - \boldsymbol{\Pi}^T(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\beta}}; \mathbf{X}_C) \} \mathbf{X}_v &= -\lambda \frac{(\hat{\boldsymbol{\beta}}^T)_v}{\|(\hat{\boldsymbol{\beta}}^T)_v\|_2} \quad \text{for } (\hat{\boldsymbol{\beta}}^T)_v \neq \mathbf{0}, \\ \frac{1}{n} \| \{ \mathbf{Y}^T - \boldsymbol{\Pi}^T(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\beta}}; \mathbf{X}_C) \} \mathbf{X}_v \|_2 &\leq \lambda \quad \text{for } (\hat{\boldsymbol{\beta}}^T)_v = \mathbf{0}. \end{aligned}$$

Thus, analogously to (5), for $D \supseteq C$, $(\hat{\boldsymbol{\beta}}^T(\lambda, D))_{D \setminus C} = \mathbf{0}$ and $(\hat{\boldsymbol{\beta}}^T(\lambda, D))_C = \hat{\boldsymbol{\beta}}^T(\lambda, C)$ if and only if

$$\max_{v \in D \setminus C} \frac{1}{n} \| \{ \mathbf{Y}^T - \boldsymbol{\Pi}^T(\hat{\boldsymbol{\mu}}(\lambda, C), \hat{\boldsymbol{\beta}}(\lambda, C); \mathbf{X}_C) \} \mathbf{X}_v \|_2 \leq \lambda.$$

4.2 Structural Sparsity

Although in our Backtracking algorithm, interaction terms are only added as candidates for selection when all their lower order interactions and main effects are active, this hierarchy in the selection of candidates does not necessarily follow through to the final model: one can have first-order interactions present in the final model without one or more of their main effects, for example. One way to enforce the hierarchy constraint in the final model is to use a base procedure which obeys the constraint itself. Examples of such base procedures are provided by the Composite Absolute Penalties (CAP) family (Zhao et al., 2009).

Consider the linear regression setup with interactions. For simplicity we only describe Backtracking with first-order interactions. Let C be the candidate set and let $I = C \setminus C_1$ be the (first-order) interaction terms in C . In order to present the penalty, we borrow some notation from Combinatorics. Let $C_1^{(r)}$ denote the set of r -subsets of C_1 . For $A \subseteq C_1^{(r)}$ and $r \geq 1$, define

$$\begin{aligned} \partial_l(A) &= \{v \in C_1^{(r-1)} : v \subset u \text{ for some } u \in A\} \\ \partial_u(A) &= \{v \in C_1^{(r+1)} : v \subset u \text{ for some } u \in A\} \end{aligned}$$

These are known as the *lower shadow* and *upper shadow* respectively (Bollobás, 1986).

Our objective function Q is given by

$$Q(\boldsymbol{\mu}, \boldsymbol{\beta}) = \frac{1}{2n} \|\mathbf{Y} - \boldsymbol{\mu} \mathbf{1} - \mathbf{X}_C \boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}_{C_1 \setminus \partial_l(I)}\|_1 + \lambda \sum_{v \in \partial_l(I)} \|\boldsymbol{\beta}_{\{v\} \cup \partial_u(\{v\} \cap I)}\|_\gamma + \lambda \|\boldsymbol{\beta}_I\|_1,$$

where $\gamma > 1$. For example, if $C = \{\{1\}, \dots, \{4\}, \{1, 2\}, \{2, 3\}\}$, then omitting the factor of λ , the penalty terms in Q are

$$|\beta_4| + \|(\beta_1, \beta_{\{1,2\}})^T\|_\gamma + \|(\beta_2, \beta_{\{1,2\}}, \beta_{\{2,3\}})^T\|_\gamma + \|(\beta_3, \beta_{\{2,3\}})^T\|_\gamma + |\beta_{\{1,2\}}| + |\beta_{\{2,3\}}|.$$

The form of this penalty forces interactions to enter the active set only after or with their corresponding main effects.

The KKT conditions for this optimisation take a more complicated form than those for the Lasso. Nevertheless, checking they hold for a trial solution is an easier task than computing a solution.

4.3 Nonlinear Models

If a high-dimensional additive modelling method (Ravikumar et al., 2009; Meier et al., 2009) is used as the base procedure, it is possible to fit nonlinear models with interactions. Here each variable is a collection of basis functions, and to add an interaction between variables, one adds the tensor product of the two collections of basis functions, penalizing the new interaction basis functions appropriately. Structural sparsity approaches can also be used here. The VANISH method of Radchenko and James (2010) uses a CAP-type penalty in nonlinear regression, and this can be used as a base procedure in a similar way to that sketched above.

4.4 Introducing more Candidates

In our description of the Backtracking algorithm, we only introduce an interaction term when *all* of its lower order interactions and main effects are active. Another possibility, in the spirit of MARS (Friedman, 1991), is to add interaction terms when *any* of their lower order interactions or main effects are active. As at the k th step of Backtracking, there will be roughly kp extra candidates, an approach that can enforce the hierarchical constraint may be necessary to allow main effects to be selected from amongst the more numerous interaction candidates. The key point to note is that if the algorithm is terminated after T steps, we are having to deal with roughly at most Tp variables rather than $O(p^2)$, the latter coming from including all first-order interactions.

Another option proposed by a referee is to augment the initial set of candidates with interactions selected through a simple marginal screening step. If only pairwise interactions are considered here, then this would require $O(p^2\pi)$ operations. Though this would be infeasible for very large p , for moderate p this would allow important interactions whose corresponding main effects are not strong to be selected.

5. Numerical Results

In this section we evaluate the performance of Backtracking on both simulated and real data sets.

5.1 Simulations

Here we consider five numerical studies designed to demonstrate the effectiveness of Backtracking with the Lasso and also highlight some of the drawbacks of using the Lasso with main effects only, when interactions are present. In each of the five scenarios, we generated 200 design matrices with $n = 250$ observations and $p = 1000$ covariates. The rows of the design matrices were sampled independently from $N_p(\mathbf{0}, \Sigma)$ distributions. The covariance matrix Σ was chosen to be the identity in all scenarios except scenario 2, where

$$\Sigma_{ij} = 0.75^{-|i-j|-p/2+p/2}.$$

Thus in this case, the correlation between the components decays exponentially with the distance between them in $\mathbb{Z}/p\mathbb{Z}$.

We created the responses according to the linear model with interactions and set the intercept to 0:

$$\mathbf{Y} = \mathbf{X}_S^* \beta_S^* + \epsilon, \quad \epsilon_i \stackrel{\text{i.i.d.}}{\sim} N(0, \sigma^2). \quad (6)$$

The error variance σ^2 was chosen to achieve a signal-to-noise ratio (SNR) of either 2 or 3. The set of main effects in S^* , S_1^* , was $1, \dots, 10$. The subset of variables involved in interactions was $1, \dots, 6$. The set of first-order interactions in S^* chosen in the different scenarios, S_2^* , is displayed in Table 1, and we took $S^* = S_1^* \cup S_2^*$ so S^* contained no higher order interactions. In each simulation run, $\beta_{S_1^*}^*$ was fixed and given by

$$(2, -1.5, 1.25, -1, 1, -1, 1, 1, 1, 1)^T.$$

Scenario	S_2^*
1	\emptyset
2	\emptyset
3	$\{\{1, 2\}, \{3, 4\}, \{5, 6\}\}$
4	$\{\{1, 2\}, \{1, 3\}, \dots, \{1, 6\}\}$
5	$\mathcal{I}(1, 2, 3) \cup \mathcal{I}(4, 5, 6)$

Table 1: Simulation settings.

Each component of $\beta_{S_2^*}^*$ was chosen to be $\sqrt{\|\beta_{S_1^*}^*\|_2^2 / |S_1^*|}$. Thus the squared magnitude of the interactions was equal to average of the squared magnitudes of the main effects.

In all of the scenarios, we applied four methods: the Lasso using only the main effects; iterated Lasso fits; marginal screening for interactions followed by the Lasso; and the Lasso with Backtracking. Note that due to the size of p in these examples, most of the methods for finding interactions in lower-dimensional data discussed in Section 1, are computationally impractical here.

For the iterated Lasso fits, we repeated the following process. Given a design matrix, first fit the Lasso. Then apply 5-fold cross-validation to give a λ value and associated active set. Finally add all interactions between variables in this active set to the design matrix, ready for the next iteration. For computational feasibility, the procedure was terminated when the number of variables in the design matrix exceeded $p + 250 \times 249/2$.

With the marginal screening approach, we selected the $2p$ interactions with the largest marginal correlation with the response and added them to the design matrix. Then a regular Lasso was performed on the augmented matrix of predictors.

Additionally, in scenarios 3-5, we applied the Lasso with all main effects and only the true interactions. This theoretical Oracle approach provided a gold standard against which to test the performance of Backtracking.

We used the procedures mentioned to yield active sets on which we applied OLS to give a final estimator. To select the tuning parameters of the methods we used cross-validation randomly selection 5 folds but repeating this a total of 5 times to reduce the variance of the cross-validation scores. Thus for each λ value we obtained an estimate of the expected prediction error that was an average over the observed prediction errors on 25 (overlapping) validation sets of size $n/5 = 50$. Note that for both Backtracking and the iterated Lasso, this form of cross-validation chose not just a λ value but also a path rank. When using Backtracking, the size of the active set was restricted to 50 and the size of C_λ to $p + 50 \times 49/2 = 1225$, so T was at most 50.

In scenarios 1 and 2, the results of the methods were almost indistinguishable except that the screening approach performed far worse in scenario 1 where it tended to select several false interactions which in turn hampered the selection of main effects and resulted in a much larger prediction error.

The results of scenarios 3-5, where the signal contains interactions, are more interesting and given in Table 2. For each scenario, method and SNR level, we report 5 statistics. L_2 -sq² is the expected squared distance of the signal \mathbf{f}^* and our prediction functions $\hat{\mathbf{f}}$ based

on training data $(\mathbf{Y}_{\text{train}}, \mathbf{X}_{\text{train}})$, evaluated at a random independent test observation \mathbf{x}_{new} :

$$\mathbb{E}_{\mathbf{x}_{\text{new}}, \mathbf{Y}_{\text{train}}, \mathbf{X}_{\text{train}}} \|\mathbf{f}^*(\mathbf{x}_{\text{new}}) - \hat{\mathbf{f}}(\mathbf{x}_{\text{new}}; \mathbf{Y}_{\text{train}}, \mathbf{X}_{\text{train}})\|^2.$$

‘FP Main’ and ‘FP Inter’ are the numbers of noise main effects and noise interaction terms respectively, incorrectly included in the final active set. ‘FN Main’ and ‘FN Inter’ are the numbers of true main effects and interaction terms respectively, incorrectly excluded from the final active set.

For all the statistics presented, lower numbers are to be preferred. However, the higher number of false selections incurred by both Backtracking and the Oracle procedure compared to using the main effects only or iterated Lasso fits, is due to the model selection criterion being the expected prediction error. It should not be taken as an indication that the latter procedures are performing better in these cases.

Backtracking performs best out of the four methods compared here. Note that under all of the settings, iterated Lasso fits incorrectly selects more interaction terms than Backtracking. We see that the more careful way in which Backtracking adds candidate interactions, helps here. Unsurprisingly, fitting the Lasso on just the main effects performs rather poorly in terms of predictive performance. However, it also fails to select important main effects; Backtracking and Iterates have much lower main effect false negatives. The screening approach appears to perform worst here. This is partly because it is not making use of the fact that in all of the examples considered, the main effects involved in interactions are also informative. However, its poor performance is also due to the fact that too many false interactions are added to the design matrix after the screening stage. Reducing the number added may help to improve results, but choosing the number of interactions to include via cross-validation, for example, would be computationally costly, unless a Backtracking-type strategy of the sort introduced in this paper were used. We also note that for very large p , marginal screening of interactions would be infeasible due to the quadratic scaling in complexity with p .

5.2 Data Analyses

In this section, we look at the performance of Backtracking using two base procedures, the Lasso for the linear model and the Lasso for multinomial regression, on a regression and a classification data set. As competing methods, we consider simply using the base procedures (‘Main’), iterated Lasso fits (‘Iterated’), Lasso following marginal screening for interactions (‘Screening’), Random Forests (Breiman, 2001), hierNet (Bien et al., 2013) and MARS (Friedman, 1991) (implemented using Hastie et al. (2013)). Note that we do not view the latter two methods as competitors of Backtracking, as they are designed for use on lower dimensional data sets than Backtracking is capable of handling. However, it is still interesting to see how the methods perform on data of dimension that is perhaps approaching the upper end of what is easily manageable for methods such as hierNet and MARS, but at the lower end of what one might use Backtracking on.

Below we describe the data sets used which are both from the UCI machine learning repository (Asuncion and Newman, 2007).

Scenario	Statistic	SNR = 2			SNR = 3		
		Main	Iter-ate	Back-tracking	Main	Iter-ate	Back-tracking
3	L_2 -sq	6.95	1.40	12.87	1.21	0.82	1.68
	FP Main	3.18	2.43	0.01	2.89	3.19	3.19
	FN Main	1.26	0.38	7.24	0.24	0.14	0.14
	FP Inter	0.00	0.93	11.05	0.45	0.00	0.00
4	L_2 -sq	3.00	0.18	2.06	0.14	0.01	0.01
	FP Main	12.05	3.25	17.68	2.72	1.68	10.44
	FN Main	2.22	3.88	0.02	5.34	7.05	2.58
	FP Inter	3.12	0.90	8.13	0.61	0.26	1.77
5	L_2 -sq	0.00	2.50	12.33	0.77	0.00	0.00
	FP Main	5.00	0.66	4.07	0.51	0.08	5.00
	FN Main	14.12	5.08	19.96	4.52	2.14	12.84
	FP Inter	3.07	4.75	0.02	5.87	3.43	3.01
6	L_2 -sq	3.20	1.26	8.26	0.98	0.33	2.35
	FP Main	0.00	3.28	17.97	0.87	0.00	0.00
	FN Main	6.00	1.34	5.00	1.23	0.14	6.00
	FP Inter	6.00	0.00	0.00	0.00	0.00	0.00
7	L_2 -sq	6.00	1.34	5.00	1.23	0.14	6.00
	FP Main	12.84	1.56	16.99	1.17	0.44	12.84
	FN Main	3.01	0.05	3.23	3.77	0.00	3.01
	FP Inter	3.05	21.92	0.55	0.00	0.00	3.05
8	L_2 -sq	6.00	0.39	4.14	0.30	0.00	6.00
	FP Main	6.00	0.00	0.00	0.00	0.00	6.00
	FN Main	6.00	0.00	0.00	0.00	0.00	6.00
	FP Inter	6.00	0.00	0.00	0.00	0.00	6.00

Table 2: Simulation results.

5.2.1 COMMUNITIES AND CRIME

This data set available at <http://archive.ics.uci.edu/ml/datasets/Communities+and+Crime> contains crime statistics for the year 1995 obtained from FBI data, and national census data from 1990, for various towns and communities around the USA. We took violent crimes per capita as our response: violent crime being defined as murder, rape, robbery, or assault. The data set contains two different estimates of the populations of the communities: those from the 1990 census and those from the FBI database in 1995. The latter was used to calculate our desired response using the number of cases of violent crimes. However, in several cases, the FBI population data seemed suspect and we discarded all observations where the maximum of the ratios of the two available population estimates differed by more than 1.25. In addition, we removed all observations that were missing a response and several variables for which the majority of values were missing. This resulted in a data set with $n = 1903$ observations and $p = 101$ covariates. The response was scaled to have empirical variance 1.

5.2.2 ISOLET

This data set consists of $p = 617$ features based on the speech waveforms generated from utterances of each letter of the English alphabet. The task is to learn a classifier which can determine the letter spoken based on these features. The data set is available from <http://archive.ics.uci.edu/ml/datasets/ISOLET>; see Fänty and Cole (1991) for more background on the data. We consider classification on the notoriously challenging E-set consisting of the letters ‘B’, ‘C’, ‘D’, ‘E’, ‘G’, ‘P’, ‘T’, ‘V’ and ‘Z’ (pronounced ‘zee’). As there were 150 subjects and each spoke each letter twice, we have $n = 2700$ observations spread equally among 9 classes. The dimension of this data is such that MARS and hierNet could not be applied.

5.3 Methods and Results

For the Communities and crime data set, we used the Lasso for the linear model as the base regression procedure for Backtracking and Iterates. Since the per capita violent crime response was always non-negative, the positive part of the fitted values was taken. For Main, Backtracking, Iterates, Screening and hierNet, we employed 5-fold cross-validation with squared error loss to select tuning parameters. For MARS we used the default settings for pruning the final fits using generalised cross-validation. With Random Forests, we used the default settings on both data sets. For the classification example, penalised multinomial regression was used (see Section 4.1) as the base procedure for Backtracking and Iterates, and the deviance was used as the loss function for 5-fold cross-validation. In all of the methods except Random Forests, we only included first-order interactions. When using Backtracking, we also restricted the size of C_k to $p + 50 \times 49/2 = p + 1225$.

To evaluate the procedures, we randomly selected 2/3 for training and the remaining 1/3 was used for testing. This was repeated 200 times for each of the data sets. Note that we have specifically chosen data sets with n large as well as p large. This is to ensure that comparisons between the performances of the methods can be made with more accuracy. For the regression example, out-of-sample squared prediction error was used as a measure of error; for the classification example, we used out-of-sample misclassification error with 0–1 loss. The results are given in Table 3.

Random Forests has the lowest prediction error on the regression data set, with Backtracking not far behind, whilst Backtracking wins in the classification task, and in fact achieves strictly lower misclassification error than all the other methods on 90% of all test samples. Note that a direct comparison with Random Forests is perhaps unfair, as the latter is a black-box procedure whereas Backtracking is aiming for a more interpretable model.

MARS performs very poorly indeed on the regression data set. The enormous prediction error is caused by the fact that whenever observations corresponding to either New York or Los Angeles were in the test set, MARS predicted their responses to be far larger than they were. However, even with these observations removed, the instability of MARS meant that it was unable to give much better predictions than an intercept-only model.

HierNet performs well on this data set, though it is worth noting that we had to scale the interactions to have the same ℓ_2 -norm as the main effects to get such good results (the default scaling produced error rates worse than that of an intercept-only model). Backtracking does better here. One reason for this is that because the main effects are reasonably strong in this case, a low amount of penalisation works well. However, because with hierNet, the penalty on the interactions is coupled with the penalty on the main effects, the final model tended to include close to two hundred interaction terms. The Screening approach similarly suffers from including too many interactions and performs only a little better than a main effects only fit.

The way that Backtracking creates several solution paths with varying numbers of interaction terms means that it is possible to fit main effects and a few interactions using a low penalty without this low penalisation opening the door to many other interaction terms. The iterated Lasso approach also has this advantage, but as the number of interactions are increased in discrete stages, it can miss a candidate set with the right number of interactions that may be picked up by the more continuous model building process used

Method	Error	
	Communities and crime	ISOLET
Main	0.414 (6.5×10^{-3})	0.0641 (4.7×10^{-4})
Iterate	0.384 (5.9×10^{-3})	0.0641 (4.7×10^{-4})
Screening	0.390 (7.8×10^{-3})	-
Backtracking	0.365 (3.7×10^{-3})	0.0563 (4.5×10^{-4})
Random Forest	0.356 (2.4×10^{-3})	0.0837 (6.0×10^{-4})
hierNet	0.373 (4.7×10^{-3})	-
MARS	5580.586 (3.1×10^3)	-

Table 3. Real data analyses results. Average error rates over 200 training–testing splits are given, with standard deviations of the results divided by $\sqrt{200}$ in parentheses.

by Backtracking. This occurs in a rather extreme way with the ISOLET data set where, since in the first stage of the iterated Lasso, cross-validation selected far too many variables (> 250), the second and subsequent steps could not be performed. This is why the results are identical to using the main effects alone.

6. Theoretical Properties

Our goal in this section is to understand under what circumstances Backtracking with the Lasso can arrive at a set of candidates, C^* , that contains all of the true interactions, and only a few false interactions. On the event on which this occurs, we can then apply many of the existing results on the Lasso, to show that the solution path $\hat{\beta}(\lambda, C^*)$ has certain properties. As an example, in Section 6.2 we give sufficient conditions for the existence of a λ^* such that $\{v : \hat{\beta}_v(\lambda^*, C^*) \neq 0\}$ equals the true set of variables.

We work with the normal linear model with interactions,

$$Y = \mu^* \mathbf{1} + \mathbf{X}_S^* \beta_S^* + \epsilon, \tag{7}$$

where $\epsilon_l \stackrel{i.i.d.}{\sim} N(0, \sigma^2)$, and to ensure identifiability, \mathbf{X}_S^* has full column rank. We will assume that $S^* = S_1^* \cup S_2^*$, where S_1^* and S_2^* are main effects and two-way interactions respectively. Let the interacting main effects be I^* ; formally, I^* is the smallest set of main effects such that $I \setminus I^* \supseteq S_2^*$. Assume $I^* \subseteq S_1^*$ so interactions only involve important main effects. Let $s_l = |S_l^*|$, $l = 1, 2$ and set $s = s_1 + s_2$. Define $C^* = C_1 \cup I(S_1^*)$. Note that C^* contains S^* but not additional interactions from any variables from $C_1 \setminus S_1^*$.

Although the Backtracking algorithm was presented for a base path algorithm that computed solutions at only discrete values, for the following results, we need to imagine an idealised algorithm which computes the entire path of solutions. In addition, we will assume that we only allow first-order interactions in the Backtracking algorithm, and that $T \geq s_1$.

We first consider the special case where the design matrix is derived from a random matrix with i.i.d. multivariate normal rows, before describing a result for fixed design.

6.1 Random Normal Design

Let the random matrix \mathbf{Z} have independent rows distributed as $N_p(\mathbf{0}, \Sigma)$. Suppose that \mathbf{X}_{C_1} , the matrix of main effects, is formed by scaling and centring \mathbf{Z} . We consider an asymptotic regime where \mathbf{X} , \mathbf{f}^* , S^* , σ^2 and p can all change as $n \rightarrow \infty$, though we will suppress their dependence on n in the notation. Furthermore, for sets of indices $S, M \subseteq \{1, \dots, p\}$, let $\Sigma_{S,M} \in \mathbb{R}^{|S| \times |M|}$ denote the submatrix of Σ formed from those rows and columns of Σ indexed by S and M respectively. For any positive semi-definite matrix \mathbf{A} , we will let $c_{\min}(\mathbf{A})$ and $c_{\max}(\mathbf{A})$ denote its minimal and maximal eigenvalues respectively. For sequences a_n, b_n , by $a_n \succ b_n$ we mean $b_n = o(a_n)$. We make the following assumptions.

- A1. $c_{\min}(\Sigma_{S_1^*, S_1^*}) \geq c_* > 0$.
- A2. $\sup_{\tau \in \mathbb{R}^{S_1^*}, \|\tau\|_\infty \leq 1} \|\Sigma_{N, S_1^*}^{-1} \Sigma_{S_1^*, S_1^*}^{-1} \tau\|_\infty \leq \delta < 1$.
- A3. $s_1^4 \log(p)/n \rightarrow 0$ and $s_1^8 \log(s_1)^2/n \rightarrow 0$.
- A4. $\min_{j \in S^*} |\beta_j^*| \succ \frac{s_1(\sigma \sqrt{\log p} + \sqrt{s_1 + \log p})}{\sqrt{n}} + \frac{\sqrt{s_1^3 \log(s_1)}}{n^{1/3}}$.
- A5. $\|\beta_{S_2^*}^*\|_2$ is bounded as $n \rightarrow \infty$ and $c_{\max}(\Sigma_{S_1^*, S_1^*}) \leq c^* < \infty$.

A1 is a standard assumption in high-dimensional regression and is, for example, implied by the compatibility constant of Bühlmann and van de Geer (2011a) being bounded away from zero. A2 is closely related to irrepresentable conditions (see Meinshausen and Bühlmann (2006), Zhao and Yu (2006), Zou (2006), Bühlmann and van de Geer (2011a), Wainwright (2009)), which are used for proving variable selection consistency of the Lasso. Note that although here the signal may contain interactions, our irrepresentable-type condition only involves main effects.

A3 places restrictions on the rates at which s_1 and p can increase with n . The first condition involving $\log(p)$ is somewhat natural as $s_1^2 \log(p)/n \rightarrow 0$ would typically be required in order to show ℓ_1 estimation consistency of β where only s_1 main effects are present; here our effective number of variables is $s_1 \leq s \leq s_1^2$. The second condition restricts the size of s_1 more stringently but is nevertheless weaker than equivalent conditions in Hao and Zhang (2014).

A4 is a minimal signal strength condition. The term involving σ is the usual bound on the signal strength required in results on variable selection consistency with the Lasso when there are s_1^2 non-zero variables. Due to the presence of interactions, the terms not involving σ place additional restrictions on the sizes of non-zero components of β^* even when $\sigma = 0$. A5 ensures that the model is not too heavily misspecified in the initial stages of the algorithm, where we are regressing on only main effects.

The following theorem states that given the assumptions above, with probability tending to 1 we are guaranteed a candidate set will be produced by our algorithm which contains all true interactions and no interactions involving a noise variable.

Theorem 1 *Assuming A1–A5, the probability that there exists a k^* such that $C^* \supseteq C_{k^*} \supseteq S^*$ tends to 1 as $n \rightarrow \infty$.*

6.2 Fixed Design

The result for a random normal design above is based on a corresponding result for fixed design which we present here. In order for Backtracking not to add any interactions involving noise variables, to begin with, one pair of interacting signal variables must enter the solution path before any noise variables. Other interacting signal variables need only become active after the interaction between this first pair has become active. Thus we need that there is some ordering of the interacting variables where each variable only requires interactions between those variables earlier in the order to be present before it can become active. Variables early on in the order must have the ability to be selected when there is serious model misspecification as few interaction terms will be available for selection. Variables later in the order only need to have the ability to be selected when the model is approximately correct.

Note that a signal variable having a coefficient large in absolute value does not necessarily ensure that it becomes active before any noise variable. Indeed, in our example in Section 2, variable 5 did not enter the solution path at all when only main effects were present, but had the largest coefficient. Write \mathbf{f}^* for $\mathbf{X}_{S^*} \beta_{S^*}^*$, and for a set S such that \mathbf{X}_S has full column rank, define

$$\beta^S := (\mathbf{X}_S^T \mathbf{X}_S)^{-1} \mathbf{X}_S^T \mathbf{f}^*.$$

Intuitively what should matter are the sizes of the appropriate coefficients of β^S for suitable choices of S . In the next section, we give a sufficient condition based on β^S for a variable $v \in S$ to enter the solution path before any variable outside S .

6.2.1 THE ENTRY CONDITION

Let $\mathbf{P}^S = \mathbf{X}_S (\mathbf{X}_S^T \mathbf{X}_S)^{-1} \mathbf{X}_S^T$ denote orthogonal projection on to the space spanned by the columns of \mathbf{X}_S . Further, for any two candidate sets S, M that are sets of subsets of $\{1, \dots, p\}$, define

$$\hat{\Sigma}_{S,M} = \frac{1}{n} \mathbf{X}_{S,M}^T \mathbf{X}_{S,M}.$$

Now given a set of candidates, C , let $v \in S \subset C$ and write $M = C \setminus S$. For $\eta > 0$, we shall say that the $\text{Ent}(v, S, C; \eta)$ condition holds if, \mathbf{X}_S has full column rank, and the following holds,

$$\sup_{\tau_S \in \mathbb{R}^{|S|}, \|\tau_S\|_\infty \leq 1} \|\hat{\Sigma}_{M,S} \hat{\Sigma}_{S,S}^{-1} \tau_S\|_\infty < 1, \quad (8)$$

$$|\beta_v^S| > \max_{w \in M} \left\{ \frac{1}{n} |\mathbf{X}_w^T (\mathbf{I} - \mathbf{P}^S) \mathbf{f}^*| + 2\eta + \eta \right\} \|\hat{\Sigma}_{S,S}^{-1}\|_1. \quad (9)$$

In Lemma 4 given in the appendix, we show that this condition is sufficient for variable v to enter the active set before any variable in M , when the set of candidates is C and $\|\mathbf{X}_C^T \epsilon\|_\infty \leq \eta$. In addition, we show that v will remain in the active set at least until some variable from M enters the active set.

The second part of the entry condition (9) asserts that coefficient v of the regression of \mathbf{f}^* on \mathbf{X}_S must exceed a certain quantity that we now examine in more detail. The

$\frac{1}{n} \mathbf{X}_u^T (\mathbf{I} - \mathbf{P}_S) \mathbf{f}^*$ term is the sample covariance between \mathbf{X}_u , which is one of the columns of \mathbf{X}_M , and the residual from regressing \mathbf{f}^* on \mathbf{X}_S . Note that the more of S^* that S contains, the closer this will be to 0.

To understand the $\|(\hat{\Sigma}_{S,S}^{-1})_v\|_1$ term, without loss of generality take v as $\{1\}$ and write $\mathbf{b} = \hat{\Sigma}_{S^* \setminus \{v\}, \{v\}}$ and $\mathbf{D} = \hat{\Sigma}_{S^* \setminus \{v\}, S^* \setminus \{v\}}$. For any square matrix $\hat{\Sigma}$, let $c_{\min}(\hat{\Sigma})$ denote its minimal eigenvalue. Using the formula for the inverse of a block matrix and writing s for $|S|$, we have

$$\begin{aligned} \|(\hat{\Sigma}_{S,S}^{-1})_v\|_1 &= \left\| \begin{pmatrix} 1 + \mathbf{b}^T (\mathbf{D} - \mathbf{b} \mathbf{b}^T)^{-1} \mathbf{b} \\ -(\mathbf{D} - \mathbf{b} \mathbf{b}^T)^{-1} \mathbf{b} \end{pmatrix} \right\|_1 \\ &\leq 1 + \frac{\|\mathbf{b}\|_2^2 + \sqrt{s-1} \|\mathbf{b}\|_2}{c_{\min}(\hat{\Sigma}_{S,S})}. \end{aligned}$$

In the final line we have used the Cauchy-Schwarz inequality and the fact that if \mathbf{w}^* is a unit eigenvector of $\mathbf{D} - \mathbf{b} \mathbf{b}^T$ with minimal eigenvalue, then

$$c_{\min}(\mathbf{D} - \mathbf{b} \mathbf{b}^T) = \left\| \hat{\Sigma}_{S,S} \begin{pmatrix} -\mathbf{b}^T \mathbf{w}^* \\ \mathbf{w}^* \end{pmatrix} \right\|_2 \geq c_{\min}(\hat{\Sigma}_{S,S}) \sqrt{1 + \|\mathbf{b}^T \mathbf{w}^*\|^2} \geq c_{\min}(\hat{\Sigma}_{S,S}).$$

Thus when variable v is not too correlated with the other variables in S , and so $\|\mathbf{b}\|_2$ is small, $\|(\hat{\Sigma}_{S,S}^{-1})_v\|_1$ will not be too large. Even when this is not the case, we still have the bound

$$\|(\hat{\Sigma}_{S,S}^{-1})_v\|_1 \leq \frac{\sqrt{|S|}}{c_{\min}(\hat{\Sigma}_{S,S})}.$$

Turning now to the denominator, $\|\hat{\Sigma}_{S,S}^{-1} \hat{\Sigma}_{S^* \setminus \{v\}}\|_1$ is the ℓ_1 -norm of the coefficient of regression of \mathbf{X}_u on \mathbf{X}_S , and the maximum of this quantity over $u \in M$ gives the left-hand side of (8). Thus when u is highly correlated with many of the variables in S , $\|\hat{\Sigma}_{S,S}^{-1} \hat{\Sigma}_{S^* \setminus \{v\}}\|_1$ will be large. On the other hand, in this case one would expect $\|(\mathbf{I} - \mathbf{P}_S) \mathbf{X}_u\|_2$ to be small, and so to some extent the numerator and denominator compensate for each other.

6.2.2 STATEMENT OF RESULTS

Without loss of generality assume $I^* = \{1, \dots, |I^*|\}$. Also let $\mathcal{J} = \{\mathcal{I}(A) : A \subseteq S_1^*\}$. Our formal assumption corresponding to the discussion at the beginning of Section 6 is the following.

The entry order condition. There is some ordering of the variables in I^* , which without loss of generality we take to simply be $1, \dots, |I^*|$, such that for each $j \in I^*$, we have,

For all $A \in \mathcal{J}$ with $\mathcal{I}(1, \dots, j-1) \subseteq A \subseteq \mathcal{I}(S_1^*)$
 $\text{Ent}(j, S_1^* \cup B, C_1 \cup A; \eta)$ holds for some $A \cap S_2^* \subseteq B \subseteq A$.

Here

$$\eta = \eta(t; n, p, s_1, \sigma) = \sigma \sqrt{\frac{t^2 + 2 \log(p + s_1^2)}{n}}.$$

First we discuss the implications for variable 1. The condition ensures that whenever the candidate set is enlarged from C_1 to also include any set of interactions built from S_1 , variable 1 enters the active set before any variable outside $\mathcal{I}(S_1^*)$, and moreover, it remains in the active set at least until a variable outside $\mathcal{I}(S_1^*)$ enters.

For $j > 2$, we see that the enlarged candidate sets for which we require the entry conditions to hold, are fewer in number. Variable $|I^*|$ only requires the entry condition to hold for candidate sets that at least include $\mathcal{I}(1, \dots, |I^*| - 1)$ and thus include almost all of S^* . What this means is that we require some ‘strong’ interacting variables, for which when \mathbf{f}^* is regressed onto a variety of sets of variables containing them (some of which contain only a few of the true interaction variables), always have large coefficients. Given the existence of such strong variables, other interacting variables need only have large coefficients when \mathbf{f}^* is regressed onto sets containing them that also include many true interaction terms. Note that the equivalent result for the success of the strategy that simply adds interactions between selected main effects would essentially require all main effect involved in interactions to satisfy the conditions imposed on the variables 1 and 2 here. Going back to the example in Section 2, variable 5 has $|\beta_5^S| \approx 0$ for all $S \subseteq \{1, \dots, 6\}$, but $|\beta_5^S| > 0$ once $\{1, 2\} \in S$ or $\{3, 4\} \in S$.

Theorem 2 Assume the entry order condition holds. With probability at least $1 - \exp(-t^2/2)$, there exists a k^* such that $C^* \supseteq C_{k^*} \supseteq S^*$.

The following corollary establishes variable selection consistency under some additional conditions.

Corollary 3 Assume the entry order condition holds. Writing $N = C^* \setminus S^*$, further assume

$$\|\hat{\Sigma}_{N,S^*} \hat{\Sigma}_{S^*,S^*}^{-1} \text{sgn}(\beta_{S^*}^*)\|_\infty < 1;$$

and that for all $v \in S^*$,

$$|\beta_v^*| > \frac{\eta \left| \text{sgn}(\beta_{S^*}^*)^T (\hat{\Sigma}_{S^*,S^*}^{-1})_v \right|}{1 - \|\hat{\Sigma}_{N,S^*} \hat{\Sigma}_{S^*,S^*}^{-1} \text{sgn}(\beta_{S^*}^*)\|_\infty} + \xi,$$

where

$$\xi = \xi(t; n, s, \sigma, c_{\min}(\hat{\Sigma}_{S^*,S^*})) = \sigma \sqrt{\frac{t^2 + 2 \log(s)}{n c_{\min}(\hat{\Sigma}_{S^*,S^*})}}.$$

Then with probability at least $1 - 3 \exp(-t^2/2)$, there exist k^* and λ^* such that

$$\mathcal{A}(\hat{\beta}(\lambda^*, C_{k^*})) = S^*.$$

Note that if we were to simply apply the Lasso to the set of candidates $C^{\text{all}} := C_1 \cup \mathcal{I}(C_1)$ (i.e. all possible main effects and their first-order interactions), we would require an irreparable condition of the form

$$\|\hat{\Sigma}_{N^{\text{all}}, S^*} \hat{\Sigma}_{S^*, S^*}^{-1} \text{sgn}(\beta_{S^*}^*)\|_\infty < 1,$$

where $N^{\text{all}} = C^{\text{all}} \setminus S^*$. Thus we would need $O(p^2)$ inequalities to hold, rather than our $O(p)$. Of course, we had to introduce many additional assumptions to reach this stage and no set of assumptions is uniformly stronger or weaker than the other. However, our proposed method is computationally feasible.

7. Discussion

While several methods now exist for fitting interactions in moderate-dimensional situations where p is in the order of hundreds, the problem of fitting interactions when the data is of truly high dimension has received less attention.

Typically, the search for interactions must be restricted by first fitting a model using only main effects, and then including interactions between those selected main effects, as well as the original main effects, as candidates in a final fit. This approach has the drawbacks that important main effects may not be selected in the initial stage as they require certain interactions to be present in order for them to be useful for prediction. In addition, the initial model may contain too many main effects when, without the relevant interactions, the model selection procedure cannot find a good sparse approximation to the true model.

The Backtracking method proposed in this paper allows interactions to be added in a more natural gradual fashion, so there is a better chance of having a model which contains the right interactions. The method is computationally efficient, and our numerical results demonstrate its effectiveness for both variable selection and prediction.

From a theoretical point of view we have shown that when used with the Lasso, rather than requiring all main effects involved in interactions to be highly correlated with the signal, Backtracking only needs there to exist some ordering of these variables where those early on in the order are important for predicting the response by themselves. Variables later in the order only need to be helpful for predicting the response when interactions between variables early on in the order are present.

Though in this paper, we have largely focussed on Backtracking used with the Lasso, the method is very general and can be used with many procedures that involve sparsity-inducing penalty functions. These methods tend to be some of the most useful for dealing with high-dimensional data, as they can produce stable, interpretable models. Combined with Backtracking, the methods become much more flexible, and it would be very interesting to explore to what extent using non-linear base procedures could yield interpretable models with predictive power comparable to black-box procedures such as Random Forests (Breiman, 2001). In addition, we believe integrating Backtracking with some of the penalty-based methods for fitting interactions to moderate-dimensional data, will prove to be a fruitful direction for future research.

Acknowledgments

I am very grateful to Richard Samworth, for many helpful comments and suggestions.

Appendix A. Construction of \mathbf{X} in Section 2

First, consider (Z_{i1}, Z_{i2}, Z_{i3}) generated from a mean zero multivariate normal distribution with $\text{Var}(Z_{ij}) = 1$, $j = 1, 2, 3$, $\text{Cov}(Z_{i1}, Z_{i2}) = 0$ and $\text{Cov}(Z_{i1}, Z_{i3}) = \text{Cov}(Z_{i2}, Z_{i3}) = 1/2$. Independently generate R_{i1} and R_{i2} each of which takes only the values $\{-1, 1\}$, each with

probability $1/2$. We form the i th row of the design matrix as follows:

$$\begin{aligned} X_{i1} &= R_{i1} \text{sgn}(Z_{i1}) |Z_{i1}|^{1/4}, \\ X_{i2} &= R_{i1} |Z_{i1}|^{3/4}, \\ X_{i3} &= R_{i2} \text{sgn}(Z_{i2}) |Z_{i2}|^{1/4}, \\ X_{i4} &= R_{i2} |Z_{i2}|^{3/4}, \\ X_{i5} &= Z_{i3}. \end{aligned}$$

The remaining X_{ij} , $j = 6, \dots, p$ are independently generated from a standard normal distribution. Note that the random signs R_{i1} and R_{i2} ensure that X_{i5} is uncorrelated with each of X_{i1}, \dots, X_{i4} . Furthermore, the fact that $X_{i1} X_{i2} = Z_{i1}$ and $X_{i3} X_{i4} = Z_{i2}$, means that when $\beta_5 = -\frac{1}{2}(\beta_7 + \beta_8)$, X_{i5} is uncorrelated with the response.

Appendix B. Proofs of Theorem 2 and Corollary 3

In this subsection we use many ideas from Section B of Wainwright (2009) and Section 6 of Bühlmann and van de Geer (2011a).

Lemma 4 *Let $S \subseteq C$ be such that X_S has full column rank and let $M = C \setminus S$. On the event*

$$\Omega_{C,\eta} := \left\{ \frac{1}{n} \|\mathbf{X}_C^T \boldsymbol{\varepsilon}\|_\infty \leq \eta \right\},$$

the following hold:

(i) *If*

$$\lambda > \max_{u \in M} \left\{ \frac{\frac{1}{n} \|\mathbf{X}_u^T (\mathbf{I} - \mathbf{P}^S) \mathbf{f}^*\| + 2\eta}{1 - \|\tilde{\boldsymbol{\Sigma}}_{S,S}^{-1} \tilde{\boldsymbol{\Sigma}}_{S,\{u\}}\|_1} \right\}, \tag{10}$$

then the Lasso solution is unique and $\hat{\boldsymbol{\beta}}_M(\lambda, C) = \mathbf{0}$.

(ii) *If λ is such that for some Lasso solution $\hat{\boldsymbol{\beta}}_M(\lambda, C) = \mathbf{0}$, and for $v \in S$,*

$$|\beta_v^S| > \|(\tilde{\boldsymbol{\Sigma}}_{S,S}^{-1})_{v\cdot}\|_1 (\lambda + \eta),$$

then for all Lasso solutions, $\hat{\beta}_v(\lambda, C) \neq 0$.

(iii) *Let*

$$\lambda^{\text{ent}} = \sup\{\lambda : \lambda \geq 0 \text{ and for some Lasso solution } \hat{\boldsymbol{\beta}}_M(\lambda, C) \neq \mathbf{0}\},$$

where we take $\sup \emptyset = 0$. If for $v \in S$,

$$|\beta_v^S| > \max_{u \in M} \left\{ \frac{\frac{1}{n} \|\mathbf{X}_u^T (\mathbf{I} - \mathbf{P}^S) \mathbf{f}^*\| + 2\eta}{1 - \|\tilde{\boldsymbol{\Sigma}}_{S,S}^{-1} \tilde{\boldsymbol{\Sigma}}_{S,\{u\}}\|_1} + \eta \right\} \|(\tilde{\boldsymbol{\Sigma}}_{S,S}^{-1})_{v\cdot}\|_1,$$

there exists a $\lambda > \lambda^{\text{ent}}$ such that the solution $\hat{\boldsymbol{\beta}}(\lambda, C)$ is unique, and for all $v \in (\lambda^{\text{ent}}, \lambda]$ and all Lasso solutions $\hat{\boldsymbol{\beta}}(\lambda, C)$, we have $\hat{\beta}_v(\lambda, C) \neq 0$.

Proof We begin by proving (i). Suppressing the dependence of $\hat{\beta}$ on λ and C , we can write the KKT conditions ((3), (4)) as

$$\frac{1}{n} \mathbf{X}_C^T (\mathbf{Y} - \mathbf{X}_C \hat{\beta}) = \lambda \hat{\tau},$$

where $\hat{\tau}$ is an element of the subdifferential $\partial \|\hat{\beta}\|_1$ and thus satisfies

$$\|\hat{\tau}\|_\infty \leq 1, \quad (11)$$

$$\hat{\beta}_0 \neq 0 \Rightarrow \hat{\tau}_0 = \text{sgn}(\hat{\beta}_0). \quad (12)$$

By decomposing \mathbf{Y} as $\mathbf{P}^S \mathbf{f}^* + (\mathbf{I} - \mathbf{P}^S) \mathbf{f}^* + \epsilon$, \mathbf{X}_C as $(\mathbf{X}_S \mathbf{X}_M)$, and noting that $\mathbf{X}_S^T (\mathbf{I} - \mathbf{P}^S) = \mathbf{0}$, we can rewrite the KKT conditions in the following way:

$$\frac{1}{n} \mathbf{X}_S^T (\mathbf{P}^S \mathbf{f}^* - \mathbf{X}_S \hat{\beta}_S) + \frac{1}{n} \mathbf{X}_S^T \epsilon - \hat{\Sigma}_{S,M} \hat{\beta}_M = \lambda \hat{\tau}_S, \quad (13)$$

$$\frac{1}{n} \mathbf{X}_M^T (\mathbf{P}^S \mathbf{f}^* - \mathbf{X}_S \hat{\beta}_S) + \frac{1}{n} \mathbf{X}_M^T \{(\mathbf{I} - \mathbf{P}^S) \mathbf{f}^* + \epsilon\} - \hat{\Sigma}_{M,M} \hat{\beta}_M = \lambda \hat{\tau}_M. \quad (14)$$

Now let $\hat{\beta}_S$ be a solution to the restricted Lasso problem.

$$(\hat{\mu}, \hat{\beta}_S) = \arg \min_{\mu, \beta_S} \left\{ \frac{1}{2n} \|\mathbf{Y} - \mu \mathbf{I} - \mathbf{X}_S \beta_S\|^2 + \lambda \|\beta_S\|_1 \right\}.$$

The KKT conditions give that $\hat{\beta}_S$ satisfies

$$\frac{1}{n} \mathbf{X}_S^T (\mathbf{Y} - \mathbf{X}_S \hat{\beta}_S) = \lambda \hat{\tau}_S, \quad (15)$$

where $\hat{\tau}_S \in \partial \|\hat{\beta}_S\|_1$. We now claim that

$$(\hat{\beta}_S, \hat{\beta}_M) = (\hat{\beta}_S, \mathbf{0}) \quad (16)$$

$$(\hat{\tau}_S, \hat{\tau}_M) = \left(\hat{\tau}_S, \hat{\Sigma}_{M,S} \hat{\Sigma}_{S,S}^{-1} (\hat{\tau}_S - \frac{1}{n} \lambda^{-1} \mathbf{X}_S^T \epsilon) + \frac{1}{n} \lambda^{-1} \mathbf{X}_M^T \{(\mathbf{I} - \mathbf{P}^S) \mathbf{f}^* + \epsilon\} \right) \quad (17)$$

is the unique solution to (13), (14), (11) and (12). Indeed, as $\hat{\beta}_S$ solves the reduced Lasso problem, we must have that (13) and (12) are satisfied. Multiplying (13) by $\mathbf{X}_S \hat{\Sigma}_{S,S}^{-1}$, setting $\hat{\beta}_M = \mathbf{0}$ and rearranging gives us that

$$\mathbf{P}^S \mathbf{f}^* - \mathbf{X}_S \hat{\beta}_S = \mathbf{X}_S \hat{\Sigma}_{S,S}^{-1} (\lambda \hat{\tau}_S - \frac{1}{n} \mathbf{X}_S^T \epsilon), \quad (18)$$

and substituting this into (14) shows that our choice of $\hat{\tau}_M$ satisfies (14). It remains to check that we have $\|\hat{\tau}_M\|_\infty \leq 1$. In fact, we shall show that $\|\hat{\tau}_M\|_\infty < 1$. Since we are on Ω_{C_T} and $\|\hat{\tau}_S\|_\infty \leq 1$, for $u \in M$ we have

$$\begin{aligned} \lambda |\hat{\tau}_u| &\leq \|\hat{\Sigma}_{S,S}^{-1} \hat{\Sigma}_{S,\{u\}}\|_1 (\lambda \|\hat{\tau}_S\|_\infty + \|\frac{1}{n} \mathbf{X}_S^T \epsilon\|_\infty) + \frac{1}{n} |\mathbf{X}_M^T (\mathbf{I} - \mathbf{P}^S) \mathbf{f}^*| + \frac{1}{n} |\mathbf{X}_M^T \epsilon| \\ &< \lambda \|\hat{\Sigma}_{S,S}^{-1} \hat{\Sigma}_{S,\{u\}}\|_1 + \frac{1}{n} |\mathbf{X}_M^T (\mathbf{I} - \mathbf{P}^S) \mathbf{f}^*| + 2\eta \\ &< \lambda, \end{aligned}$$

where the final inequality follows from (10). We have shown that there exists a solution, $\hat{\beta}$, to the Lasso optimisation problem with $\hat{\beta}_M = \mathbf{0}$. The uniqueness of this solution follows from noting that $\|\hat{\tau}_M\|_\infty < 1$, \mathbf{X}_S has full column rank and appealing to Lemma 1 of Wainwright (2009).

For (ii), note that from (13), provided $\hat{\beta}_M = \mathbf{0}$, we have that

$$\hat{\beta}_S = \mathbf{g}^S - \hat{\Sigma}_{S,S}^{-1} (\lambda \hat{\tau}_S - \frac{1}{n} \mathbf{X}_S^T \epsilon).$$

But by assumption

$$|\beta_u^S| > \|\hat{\Sigma}_{S,S}^{-1}\|_1 (\lambda + \eta) \geq \left| \hat{\Sigma}_{S,S}^{-1} \right|_v (\lambda \hat{\tau}_S - \frac{1}{n} \mathbf{X}_S^T \epsilon),$$

whence $\hat{\beta}_u \neq 0$.

(iii) follows easily from (i) and (ii). ■

Proof of Theorem 2. In all that follows, we work on the event $\Omega_{C^* \cap T}$ defined in Lemma 4. Using standard bounds for the tails of Gaussian random variables and the union bound, it is easy to show that $\mathbb{P}(\Omega_1 \cap \Omega_{C^* \cap T}) \geq 1 - \exp(-t^2/2)$. Let $N = \{1, \dots, p\} \setminus S_1^*$.

Let T be the number of steps taken by the algorithm: this would typically be T , but may be smaller if a perfect fit is reached or if $p < T$ for example. Let C_k be the largest member of $\{C_1, \dots, C_T\}$ satisfying $C_k \subseteq C^*$. Such a C_k exists since $C_1 \subseteq C^*$.

Now suppose for a contradiction that $C_k \not\subseteq S^*$. Let j be such that

$$\mathcal{I}(1, \dots, j-1) \subseteq C_k,$$

with j maximal. Since $\mathcal{I}(1) = \emptyset$, such a j exists. Let $A = C_k \setminus C_1$. Note that $A \in \mathcal{J}$ and

$$\mathcal{I}(1, \dots, j-1) \subseteq A \subseteq C^* \setminus C_1 = \mathcal{I}(S_1^*).$$

By the entry order condition, we know that j will enter the active set before any variable in N , and before a perfect fit is reached. Thus $k+1 \leq \bar{T}$ and C_{k+1} contains only additional interactions not involving any variables from N , so $C_{k+1} \subseteq C^*$. ■

Proof of Corollary 3. Let $\Omega_{C^* \cap T}$ be defined as in Lemma 4. Also define the events

$$\begin{aligned} \Omega_1 &= \left\{ \frac{1}{n} \|\mathbf{X}_N^T (\mathbf{I} - \mathbf{P}^{S^*}) \epsilon\|_\infty \leq \eta \right\}, \\ \Omega_2 &= \left\{ \frac{1}{n} \|\hat{\Sigma}_{S^*, S^*}^{-1} \mathbf{X}_N^T \epsilon\|_\infty \leq \xi \right\} \end{aligned}$$

In all that follows, we work on the event $\Omega_1 \cap \Omega_2 \cap \Omega_{C^* \cap T}$. As $\mathbf{I} - \mathbf{P}^{S^*}$ is a projection,

$$\mathbb{P}\left(\frac{1}{n} \|\mathbf{X}_N^T (\mathbf{I} - \mathbf{P}^{S^*}) \epsilon\| \leq \eta\right) \geq \mathbb{P}\left(\frac{1}{n} \|\mathbf{X}_N^T \epsilon\| \leq \eta\right).$$

Further, $\frac{1}{n} \hat{\Sigma}_{S^*, S^*}^{-1} \mathbf{X}_N^T \epsilon \sim N^{|\mathcal{S}^*|}(\mathbf{0}, \frac{1}{n} \sigma^2 \hat{\Sigma}_{S^*, S^*}^{-1})$. Thus

$$\mathbb{P}(\Omega_2) \geq |\mathcal{S}^*| \mathbb{P}(|Z| \leq \xi)$$

where $Z \sim N(0, \sigma^2 / (nc_{\min}(\hat{\Sigma}_{S^*, S^*})))$. Note that

$$\mathbb{P}(\Omega_1 \cap \Omega_2 \cap \Omega_{C^*, \eta}) \geq 1 - \mathbb{P}(\Omega_{C^*, \eta}^c) - \mathbb{P}(\Omega_1^c) - \mathbb{P}(\Omega_2^c).$$

Using this, it is straightforward to show that $\mathbb{P}(\Omega_1 \cap \Omega_2 \cap \Omega_{C^*, \eta}) \geq 1 - 3 \exp(-t^2/2)$.

Since we are on $\Omega_{C^*, \eta}$, we can assume the existence of a k^* from Theorem 2. We now follow the proof of Lemma 4 taking $S = S^*$ and $M = C_{k^*} \setminus S^* \subseteq N$. The KKT conditions become

$$\hat{\Sigma}_{S^*, S^*}(\beta_{S^*}^* - \hat{\beta}_{S^*}^*) + \frac{1}{n} \mathbf{X}_{S^*}^T \boldsymbol{\epsilon} - \hat{\Sigma}_{S^*, M} \hat{\beta}_M = \lambda \hat{\tau}_{S^*}, \quad (19)$$

$$\hat{\Sigma}_{M, S^*}(\beta_{S^*}^* - \hat{\beta}_{S^*}^*) + \frac{1}{n} \mathbf{X}_M^T \boldsymbol{\epsilon} - \hat{\Sigma}_{M, M} \hat{\beta}_M = \lambda \hat{\tau}_M, \quad (20)$$

with $\hat{\tau}$ also satisfying (11) and (12) as before. Now let λ be such that

$$\frac{\eta}{1 - \|\hat{\Sigma}_{M, S^*} \hat{\Sigma}_{S^*, S^*}^{-1} \text{sgn}(\beta_{S^*}^*)\|_\infty} < \lambda < \min_{v \in S^*} \left\{ \left| \text{sgn}(\beta_{S^*}^*)^T (\hat{\Sigma}_{S^*, S^*}^{-1})^{-1} (\beta_{S^*}^* - \xi) \right| \right\}.$$

It is straightforward to check that

$$(\hat{\beta}_{S^*}^*, \hat{\beta}_M) = (\beta_{S^*}^* - \lambda \hat{\Sigma}_{S^*, S^*}^{-1} \text{sgn}(\beta_{S^*}^*) + \frac{1}{n} \hat{\Sigma}_{S^*, S^*}^{-1} \mathbf{X}_{S^*}^T \boldsymbol{\epsilon}, \mathbf{0})$$

$$(\hat{\tau}_{S^*}, \hat{\tau}_M) = \left(\text{sgn}(\beta_{S^*}^*), \hat{\Sigma}_{M, S^*} \hat{\Sigma}_{S^*, S^*}^{-1} \text{sgn}(\beta_{S^*}^*) + \frac{1}{n} \lambda^{-1} \mathbf{X}_M^T (\mathbf{I} - \mathbf{P}^{S^*}) \boldsymbol{\epsilon} \right)$$

is the unique solution to (19), (20), (11) and (12). \blacksquare

Appendix C. Proof of Theorem 1

In the following, we make use of notation defined in Section 6.2. In addition, for convenience we write $S = S_1^*$, $M = S \cup J^*$. Also, we will write main effects variables $\{j\}$ as simply j . For any matrix \mathbf{M} , $\|\mathbf{M}\|_\infty$ will denote $\max_{j,k} |M_{j,k}|$. First we collect together various results concerning $\hat{\Sigma}_{C^*, C^*}$.

Lemma 5 Consider the setup of Theorem 1. Let \mathbb{E}_n and Var_n denote empirical expectation and variance with respect to \mathbf{Z} so that, for example $\mathbb{E}_n z_j = \sum_{i=1}^n Z_{ij}/n$.

(i) Let \mathbf{D} be the diagonal matrix indexed by C^* used to scale transformations of \mathbf{Z} in order to create \mathbf{X}_{C^*} i.e. with entries such that $D_{jj}^2 = \text{Var}_n(z_j)$ and $D_{vv}^2 = \text{Var}_n(z_j - \mathbb{E}_n z_j)(z_k - \mathbb{E}_n z_k)$ when $v = \{j, k\}$. Then

$$\max_{j \in C_1} |D_{jj}^2 - 1| = O_P(\sqrt{\log(p)/n}) \quad (21)$$

$$\max_{\{j,k\} \in M} |D_{\{j,k\}}^2 - 1 - \Sigma_{jk}^2| = O_P(\sqrt{\log(s_1)n^{-1/4}}) \quad (22)$$

$$(ii) \quad \frac{1}{n} \|\mathbf{X}_{J^*}^T \mathbf{X}_S\|_\infty = O_P(\sqrt{\log(s_1)n^{-1/3}}) \quad (23)$$

$$c_{\min}(\hat{\Sigma}_{S,S}) \geq c_* - s_1 O_P(\sqrt{\log(s_1)/n}) \quad (24)$$

$$c_{\min}(\hat{\Sigma}_{M,M}) \geq c_*^2 + s_1^2 O_P(\sqrt{\log(s_1)n^{-1/4}}) \quad (25)$$

$$c_{\max}(\hat{\Sigma}_{J^*, J^*}) \leq 2c_*^2 + s_1^2 O_P(\sqrt{\log(s_1)n^{-1/4}}). \quad (26)$$

Proof We use bounds on the tails of products of normal random variables from Hao and Zhang (2014) (equation B.9). We have

$$\begin{aligned} \max_{j,k} |\text{Cov}_n(z_j, z_k) - \Sigma_{jk}| &= \max_{j,k} |\mathbb{E}_n(z_j z_k) - \mathbb{E}_n z_j \mathbb{E}_n z_k - \Sigma_{jk}| \\ &= O_P(\sqrt{\log(p)/n}). \end{aligned}$$

Also,

$$\begin{aligned} \max_{j,k,l,m \in S} |\text{Cov}_n(z_j - \mathbb{E}_n z_j)(z_k - \mathbb{E}_n z_k), (z_l - \mathbb{E}_n z_l)(z_m - \mathbb{E}_n z_m)) - \Sigma_{jl} \Sigma_{km} - \Sigma_{jm} \Sigma_{kl}| \\ = \max_{j,k,l,m \in S} |\mathbb{E}_n(z_j z_k z_l z_m) - \mathbb{E}_n(z_j z_k) \mathbb{E}_n(z_l z_m) - \Sigma_{jl} \Sigma_{km} - \Sigma_{jm} \Sigma_{kl}| + O_P(\sqrt{\log(s_1)/n}) \\ = O_P(\sqrt{\log(s_1)n^{-1/4}}). \end{aligned}$$

Now we consider (ii). We have

$$\begin{aligned} \frac{1}{n} \|\mathbf{X}_{J^*}^T \mathbf{X}_S\|_\infty &\leq \max_{v \in J^*} D_{kk}^{-1} \max_{k \in S} |\text{Cov}_n(z_j - \mathbb{E}_n z_j)(z_k - \mathbb{E}_n z_k), z_l| \\ &\leq O_P(\sqrt{\log(s_1)n^{-1/3}}), \end{aligned}$$

the rate being driven by the size of $\mathbb{E}_n(z_j z_k z_l)$. Also

$$\begin{aligned} c_{\min}(\hat{\Sigma}_{S,S}) &= \min_{\tau \in \mathbb{R}^{s_1}: \|\tau\|_2=1} \tau \{ \Sigma_{S,S} - (\Sigma_{S,S} - \hat{\Sigma}_{S,S}) \} \tau \\ &\geq c_{\min}(\Sigma_{S,S}) - \max_{\tau \in \mathbb{R}^{s_1}: \|\tau\|_2=1} \|\tau\|_2^2 \|\Sigma_{S,S} - \hat{\Sigma}_{S,S}\|_\infty \\ &= c_* - s_1 O_P(\sqrt{\log(s_1)/n}). \end{aligned}$$

Now let $\hat{\Sigma}$ be a matrix with entries indexed by M with

$$\hat{\Sigma}_{uv} = \Sigma_{jl} \Sigma_{km} + \Sigma_{jm} \Sigma_{kl}$$

when $u = \{j, k\}$ and $v = \{l, m\}$. Lemma A.4 of Hao and Zhang (2014) shows that $c_{\min}(\hat{\Sigma}) \geq 2c_{\min}(\Sigma_{S,S})^2$ and $c_{\max}(\hat{\Sigma}) \leq 2c_{\max}(\Sigma_{S,S})^2$. Thus we have

$$\begin{aligned} c_{\min}(\hat{\Sigma}_{M,M}) &= \min_{\tau \in \mathbb{R}^{|M|}: \|\mathbf{D}_{M,M} \tau\|_2=1} \tau \mathbf{D}_{M,M} \hat{\Sigma}_{M,M} \mathbf{D}_{M,M} \tau \\ &\geq \|\mathbf{D}_{M,M}\|_\infty^{-1} c_{\min}(\mathbf{D}_{M,M} \hat{\Sigma}_{M,M} \mathbf{D}_{M,M}) \\ &\geq \{1 + O_P(\sqrt{\log(s_1)n^{-1/4}})\} \{c_*^2 - s_1^2\} \|\hat{\Sigma} - \mathbf{D}_{M,M} \hat{\Sigma}_{M,M} \mathbf{D}_{M,M}\|_\infty + O_P(\sqrt{\log(s_1)n^{-1/3}}) \\ &\geq c_*^2 + s_1^2 O_P(\sqrt{\log(s_1)n^{-1/4}}). \end{aligned}$$

Similarly

$$\begin{aligned} c_{\max}(\hat{\Sigma}_{J^*, J^*}) &= \max_{\tau \in \mathbb{R}^{|J^*|}: \|\mathbf{D}_{J^*, J^*} \tau\|_2=1} \tau \mathbf{D}_{J^*, J^*} \hat{\Sigma}_{J^*, J^*} \mathbf{D}_{J^*, J^*} \tau \\ &\leq \{1 - O_P(\sqrt{\log(s_1)n^{-1/4}})\} c_{\max}(\mathbf{D}_{J^*, J^*} \hat{\Sigma}_{J^*, J^*} \mathbf{D}_{J^*, J^*}) \\ &\leq \{1 - O_P(\sqrt{\log(s_1)n^{-1/4}})\} \{2c_*^2 + s_1^2\} \|\hat{\Sigma} - \mathbf{D}_{J^*, J^*} \hat{\Sigma}_{J^*, J^*} \mathbf{D}_{J^*, J^*}\|_\infty \\ &\leq 2c_*^2 + s_1^2 O_P(\sqrt{\log(s_1)n^{-1/4}}). \end{aligned}$$

Lemma 6 Working with the assumptions of Theorem 1, we have

$$\max_{A \in \mathcal{J}} \|\beta_{S^A}^{S_{U^A}} - \beta_{S^A}^*\|_{\infty} \leq O_P(\sqrt{s_1^3 \log(s_1)} n^{-1/3}).$$

Proof For $A \in \mathcal{J}$ let $\Delta^A \in \mathbb{R}^{|\mathcal{S}_{U^A}|}$ with $\Delta_{S^A}^A = \beta_{S^A}^{S_{U^A}} - \beta_{S^A}^*$ and $\Delta_{A^c}^A = \beta_{S^A}^{S_{U^A}}$. Define $\mathbf{g}^* = \mathbf{X}_{S^A} \beta_{S^A}^*$. Note that

$$\mathbf{r}^* = \mathbf{X}_{S^A} \beta_{S^A}^* + \mathbf{g}^*,$$

so

$$\Delta^A = (\mathbf{X}_{S_{U^A}}^T \mathbf{X}_{S_{U^A}})^{-1} \mathbf{X}_{S_{U^A}} \mathbf{g}^*.$$

First we bound $\|\Delta_{A^c}^A\|_2$ in terms of $\|\mathbf{g}^*\|_2$. We have that

$$\|\mathbf{X}_{S_{U^A}} \Delta^A\|_2^2 = \|\mathbf{X}_{S^A} \Delta_{S^A}^A\|_2^2 + 2 \Delta_{S^A}^A{}^T \mathbf{X}_{S^A}^T \mathbf{X}_{A^c} \Delta_{A^c}^A + \|\mathbf{X}_{A^c} \Delta_{A^c}^A\|_2^2 \leq \|\mathbf{g}^*\|_2^2.$$

Thus

$$c_{\min}(\frac{1}{n} \mathbf{X}_{S^A}^T \mathbf{X}_{S^A}) \|\Delta_{S^A}^A\|_2^2 - 2\sqrt{|A|} \|S\| \frac{1}{n} \mathbf{X}_{S^A}^T \mathbf{X}_{A^c} \|\Delta_{A^c}^A\|_2 + c_{\min}(\frac{1}{n} \mathbf{X}_{A^c}^T \mathbf{X}_{A^c}) \|\Delta_{A^c}^A\|_2^2 - \frac{1}{n} \|\mathbf{g}^*\|_2^2 \leq 0.$$

Thinking of this as a quadratic in $\|\Delta_{A^c}^A\|_2$ and considering the discriminant yields

$$\|\Delta_{A^c}^A\|_2^2 \leq \frac{\frac{1}{n} c_{\min}(\frac{1}{n} \mathbf{X}_{S^A}^T \mathbf{X}_{S^A}) \|\mathbf{g}^*\|_2^2}{c_{\min}(\frac{1}{n} \mathbf{X}_{S^A}^T \mathbf{X}_{S^A}) c_{\min}(\frac{1}{n} \mathbf{X}_{A^c}^T \mathbf{X}_{A^c}) - \|\frac{1}{n} \mathbf{X}_{S^A}^T \mathbf{X}_{A^c}\|_{\infty}^2 |A| |S|}.$$

Thus by Lemma 5 (ii) and condition A2, $\max_{A \in \mathcal{J}} \|\Delta_{A^c}^A\|_2 = \frac{1}{\sqrt{n}} \|\mathbf{g}^*\|_2 O_P(1)$.

But

$$\frac{1}{\sqrt{n}} \|\mathbf{g}^*\|_2 \leq \sqrt{c_{\max}(\hat{\Sigma}_{J^*, J^*})} \|\beta_{S_{S^A}^*}\|_2 = O_P(1)$$

by Lemma 5 (ii) and A5, so $\max_{A \in \mathcal{J}} \|\Delta_{A^c}^A\|_2 = O_P(1)$.

Next observe that

$$\|\mathbf{X}_{S_{U^A}} \Delta^A - \mathbf{g}^*\|_2^2 \leq \|\mathbf{X}_{A^c} \Delta_{A^c}^A - \mathbf{g}^*\|_2^2,$$

so

$$\begin{aligned} \|\Delta_{S^A}^A\|_2^2 c_{\min}(\frac{1}{n} \mathbf{X}_{S^A}^T \mathbf{X}_{S^A}) &\leq \frac{1}{n} \|\mathbf{X}_{S^A} \Delta_{S^A}^A\|_2^2 \\ &\leq 2 \frac{1}{n} \Delta_{S^A}^A{}^T \mathbf{X}_{S^A}^T (\mathbf{X}_{A^c} \Delta_{A^c}^A - \mathbf{g}^*) \\ &\leq 2\sqrt{|A|} \|S\| \|\Delta_{S^A}^A\|_2 \|\frac{1}{n} \mathbf{X}_{S^A}^T \mathbf{X}_{A^c}\|_{\infty} \|\Delta_{A^c}^A\|_2 + 2 \|\Delta_{S^A}^A\|_2 \|\frac{1}{n} \mathbf{X}_{S^A}^T \mathbf{g}^*\|_2. \end{aligned}$$

Therefore

$$\|\Delta_{S^A}^A\|_{\infty} \leq 2\{c_{\min}(\frac{1}{n} \mathbf{X}_{S^A}^T \mathbf{X}_{S^A})\}^{-1} (\sqrt{|A|} \|S\| \|\frac{1}{n} \mathbf{X}_{S^A}^T \mathbf{X}_{A^c}\|_{\infty} \|\Delta_{A^c}^A\|_2 + \|\frac{1}{n} \mathbf{X}_{S^A}^T \mathbf{g}^*\|_2),$$

so

$$\max_{A \in \mathcal{J}} \|\Delta_{S^A}^A\|_{\infty} \leq 2\{c_{\min}(\frac{1}{n} \mathbf{X}_{S^A}^T \mathbf{X}_{S^A})\}^{-1} (\sqrt{|S|} \|J^*\| \|\frac{1}{n} \mathbf{X}_{S^A}^T \mathbf{X}_{J^*}\|_{\infty} O_P(1) + \|\frac{1}{n} \mathbf{X}_{S^A}^T \mathbf{g}^*\|_2).$$

Now

$$\begin{aligned} \|\frac{1}{n} \mathbf{X}_{S^A}^T \mathbf{g}^*\|_2 &\leq \sqrt{|S|} \|\frac{1}{n} \mathbf{X}_{S^A}^T \mathbf{X}_{S^A}\|_{\infty} \|\beta_{S_{S^A}^*}\|_1 \\ &\leq O_P(s_1 \sqrt{\log(s_1)} n^{-1/3}). \end{aligned}$$

Thus

$$\max_{A \in \mathcal{J}} \|\Delta_{S^A}^A\|_{\infty} \leq O_P(\sqrt{s_1^3 \log(s_1)} n^{-1/3}).$$

Proof of Theorem 1. In view of Theorem 2 and its proof, it is enough to show that with probability tending to 1, we have

$$\max_{A \in \mathcal{J}} \sup_{\tau \in \mathbb{R}^{p_1}} \|\hat{\Sigma}_{N, S_{U^A}} \hat{\Sigma}_{S_{U^A}, S_{U^A}}^{-1} \tau\|_{\infty} < 1, \quad (27)$$

$$\min_{j \in \mathcal{J}} \min_{A \in \mathcal{J}} |\beta_j^{S_{U^A}}| > \max_{A \in \mathcal{J}} \max_{j \in \mathcal{N}} \left\{ \frac{\frac{1}{n} \mathbf{X}_j^T (\mathbf{I} - \mathbf{P}^{S_{U^A}}) \mathbf{r}^*}{1 - \|\hat{\Sigma}_{S_{U^A}, S_{U^A}}^{-1} \hat{\Sigma}_{S_{U^A}, S_{U^A}}\|} + \frac{2 \frac{1}{n} \|\mathbf{X}_{C^*}^T \mathbf{e}\|_{\infty}}{n} + \frac{1}{n} \|\mathbf{X}_{C^*}^T \mathbf{e}\|_{\infty} \right\} \frac{\sqrt{|M|}}{c_{\min}(\hat{\Sigma}_{M, M})}. \quad (28)$$

First note that for $j \in N$, $\mathbf{Z}_j = \mathbf{Z}_S \Sigma_{S, S}^{-1} \Sigma_{S, j} + \mathbf{E}_j$ where \mathbf{E}_j is independent of \mathbf{Z}_S and $\mathbf{E}_j \sim N_n(\mathbf{0}, (1 - \Sigma_{j, S} \Sigma_{S, S}^{-1} \Sigma_{S, j}) \mathbf{I})$. Thus

$$\mathbf{X}_j D_j = \mathbf{X}_j \mathbf{D}_{S, S} \Sigma_{S, S}^{-1} \Sigma_{S, j} + \mathbf{E}_j - \mathbf{1} \bar{E}_j,$$

and

$$\max_{A \in \mathcal{J}} \|(\mathbf{X}_{S_{U^A}}^T \mathbf{X}_{S_{U^A}})^{-1} \mathbf{X}_{S_{U^A}}^T \mathbf{X}_j\|_1 \leq D_{kk}^{-1} \|\mathbf{D}_{S, S} \Sigma_{S, S}^{-1} \Sigma_{S, j}\|_1 + \max_{A \in \mathcal{J}} \|\hat{\Sigma}_{S_{U^A}, S_{U^A}}^{-1} \frac{1}{n} \mathbf{X}_{S_{U^A}}^T \mathbf{E}_j\|_1.$$

Now the second term above is at most

$$\max_{A \in \mathcal{J}} \max_{\tau \in \mathbb{R}^{|\mathcal{S}_{U^A}|}: \|\tau\|_2 \leq 1} \|\hat{\Sigma}_{S_{U^A}, S_{U^A}}^{-1} \frac{1}{n} \mathbf{X}_{S_{U^A}}^T \mathbf{E}_j\|_2.$$

But

$$\begin{aligned} \max_{A \in \mathcal{J}} \max_{\tau \in \mathbb{R}^{|\mathcal{S}_{U^A}|}: \|\tau\|_{\infty} \leq 1} \|\hat{\Sigma}_{S_{U^A}, S_{U^A}}^{-1} \tau\|_1 &\leq \frac{\sqrt{|M|}}{c_{\min}(\hat{\Sigma}_{M, M})} \\ &\leq \frac{c_2^2 + s_1^2 O_P(\sqrt{\log(s_1)} n^{-1/4})}{\sqrt{|M|}}. \end{aligned}$$

Also since for $v \in M$ and $j \in N$, $\mathbf{X}_v^T \mathbf{E}_j / n \sim N(0, 1)$ we have

$$\max_{j \in N} \|\frac{1}{n} \mathbf{X}_M^T \mathbf{E}_j\|_2^2 \leq |M| O_P(\log(p)/n).$$

Therefore

$$\max_{A \in \mathcal{J}} \sup_{\tau \in \mathbb{R}^{p_1}} \|\hat{\Sigma}_{N, S_{U^A}} \hat{\Sigma}_{S_{U^A}, S_{U^A}}^{-1} \tau\|_{\infty} \leq (1 + o_P(1)) \delta + \frac{s_1^2 o_P(1)}{c_2^2 + o_P(1)}.$$

This shows that (27) is satisfied with probability tending to 1.
Next

$$\max_{j \in N} \max_{A \in \mathcal{U}} \frac{1}{n} |\mathbf{X}_j^T (\mathbf{I} - \mathbf{P}^{S_{UA}}) \mathbf{f}^*| = \max_{j \in N} \max_{A \in \mathcal{U}} \frac{D_{-1}^{-1}}{n} |\mathbf{E}_j^T (\mathbf{I} - \mathbf{P}^{S_{UA}}) \mathbf{X}_A \beta_A^*|.$$

Since $\mathbf{E}_j^T (\mathbf{I} - \mathbf{P}^{S_{UA}}) \mathbf{X}_A \beta_A^* / n \sim N(0, \|(\mathbf{I} - \mathbf{P}^{S_{UA}}) \mathbf{X}_A \beta_A^*\|_2^2 / n^2)$ we have

$$\max_{j \in N} \max_{A \in \mathcal{U}} \frac{1}{n} |\mathbf{X}_j^T (\mathbf{I} - \mathbf{P}^{S_{UA}}) \mathbf{f}^*| \leq \sqrt{\frac{\log(2^{s_1} p)}{n}} \frac{1}{\sqrt{n}} \|\mathbf{X}_{S_2^*} \beta_{S_2^*}^*\|_2 O_P(1).$$

By (26) we have

$$\frac{1}{\sqrt{n}} \|\mathbf{X}_{S_2^*} \beta_{S_2^*}^*\|_2 \leq \{2c^* + s_1^* \sqrt{\log(s_1)} n^{-1/4} O_P(1)\} \|\beta_{S_2^*}\|_2.$$

Now using Lemma 6 we see that the difference between the LHS and RHS of (28) is at least

$$\min_{j \in \mathcal{P}} |\beta_j^*| - O_P \left(\sqrt{s_1^* \log(s_1)} n^{-1/3} \right) - \left(\frac{\sqrt{s_1 + \log p} + \sigma \sqrt{\log p} / \sqrt{n}}{1 - \delta + o_P(1)} + \sigma \sqrt{\frac{\log(p)}{n}} \right) s_1 O_P(1).$$

Thus A4 ensures that (28) holds with probability tending to 1. ■

References

- A. Asuncion and D. J. Newman. UCI Machine Learning Repository, 2007. URL <http://archive.ics.uci.edu/ml>.
- F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Structured sparsity through convex optimization. *Statistical Science*, 27:450–468, 2012a.
- F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4:1–106, 2012b.
- P. J. Bickel, Y. Ritov, and A. B. Tsybakov. Hierarchical selection of variables in sparse high-dimensional regression. *IMS Collections*, 6:56–69, 2010.
- J. Bien, J. Taylor, and R. Tibshirani. A lasso for hierarchical interactions. *Annals of Statistics*, 41(3):1111–1141, 2013.
- B. Bollobás. *Combinatorics*. Cambridge University Press, 1986.
- L. Breiman. Random Forests. *Machine Learning*, 45:5–32, 2001.
- P. Bühlmann and S. van de Geer. On the conditions used to prove oracle results for the lasso. *Electronic Journal of Statistics*, 3:1360–1392, 2011a.
- P. Bühlmann and S. van de Geer. *Statistics for high-dimensional data*. Springer, 2011b.
- B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least Angle Regression. *Annals of Statistics*, 32:407–451, 2004.
- M. Fandy and R. Cole. Spoken letter recognition. In R.P. Lippman, J. Moody, and D.S. Touretzky, editors, *Advances in Neural Information Processing Systems*, volume 13, San Mateo, CA, 1991. Morgan Kaufmann.
- J. Friedman. Multivariate adaptive regression splines. *Annals of Statistics*, 19:1–67, 1991.
- J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33:1–22, 2010.
- N. Hao and H. H. Zhang. Interaction screening for ultrahigh-dimensional data. *Journal of the American Statistical Association*, 109(507):1285–1301, 2014.
- A. Haris, D. Witten, and N. Simon. Convex modeling of interactions with strong heredity. *Journal of Computational and Graphical Statistics*, (just-accepted):1–35, 2015.
- T. Hastie, R. Tibshirani, F. Leisch, K. Hornik, and B. D. Ripley. *mda: Mixture and flexible discriminant analysis*, 2013. URL <http://CRAN.R-project.org/package=mda>. R package version 0.4–4.
- R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal Methods for Hierarchical Sparse Coding. *Journal of Machine Learning Research*, 12:2297–2334, 2011.
- M. Linn and T. Hastie. Learning interactions via hierarchical group-lasso regularization. *Journal of Computational and Graphical Statistics*, 24(3):627–654, 2015.
- Y. Lin and H. H. Zhang. Component selection and smoothing in multivariate nonparametric regression. *Annals of Statistics*, 35:2272–2297, 2006.
- L. Meier, S. van de Geer, and P. Bühlmann. High-dimensional additive modelling. *Annals of Statistics*, 37:3779–3821, 2009.
- N. Meinshausen and P. Bühlmann. High dimensional graphs and variable selection with the Lasso. *Annals of Statistics*, 34:1436–1462, 2006.
- R. Development Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2005. URL <http://www.R-project.org>. ISBN 3-900051-07-0.
- P. Radchenko and G. James. Variable selection using adaptive nonlinear interaction structures in high dimensions. *Journal of the American Statistical Association*, 105:1541–1553, 2010.
- P. Ravikumar, J. Lafferty, H. Liu, and L. Wasserman. Sparse additive models. *Journal of the Royal Statistical Society, Series B*, 71:1009–1030, 2009.
- R. D. Shah and N. Meinshausen. Random intersection trees. *The Journal of Machine Learning Research*, 15(1):629–654, 2014.

- R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.
- B. Turlach. Discussion of ‘Least angle regression’. *Annals of Statistics*, 32:481–490, 2004.
- M. J. Wainwright. Sharp thresholds for high-dimensional and noisy recovery of sparsity. *IEEE Transactions on Information Theory*, 55:2183–2202, 2009.
- J. Wu, B. Devlin, S. Ringquist, M. Trucco, and K. Roeder. Screen and clean: a tool for identifying interactions in genome-wide association studies. *Genetic Epidemiology*, 34: 275–285, 2010.
- M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006.
- M. Yuan, Y. R. Joseph, and Y. Lin. An efficient variable selection approach for analyzing designed experiments. *Technometrics*, 49:430–439, 2007.
- M. Yuan, R. Joseph, and H. Zou. Structured variable selection and estimation. *Annals of Applied Statistics*, 3:1738–1757, 2009.
- P. Zhao and B. Yu. On Model Selection Consistency of Lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006.
- P. Zhao, G. Rocha, and B. Yu. The composite absolute families penalty for grouped and hierarchical variable selection. *Annals of Statistics*, 37:3648–3497, 2009.
- H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101:1418–1429, 2006.

Choice of V for V -Fold Cross-Validation in Least-Squares Density Estimation

Sylvain Arlot

Laboratoire de Mathématiques d'Orsay

Univ. Paris-Sud, CNRS, Université Paris-Saclay
91405 Orsay, France

SYLVAIN.ARLOT@MATH.U-PSUD.FR

Matthieu Lerasle

CNRS

Univ. Nice Sophia Antipolis LJAD CNRS UMR 7351
06100 Nice France

MLERASLE@UNICE.FR

Editor: Xiaotong Shen

Abstract

This paper studies V -fold cross-validation for model selection in least-squares density estimation. The goal is to provide theoretical grounds for choosing V in order to minimize the least-squares loss of the selected estimator. We first prove a non-asymptotic oracle inequality for V -fold cross-validation and its bias-corrected version (V -fold penalization). In particular, this result implies that V -fold penalization is asymptotically optimal in the nonparametric case. Then, we compute the variance of V -fold cross-validation and related criteria, as well as the variance of key quantities for model selection performance. We show that these variances depend on V like $1 + 4/(V - 1)$, at least in some particular cases, suggesting that the performance increases much from $V = 2$ to $V = 5$ or 10 , and then is almost constant. Overall, this can explain the common advice to take $V = 5$ —at least in our setting and when the computational power is limited—, as supported by some simulation experiments. An oracle inequality and exact formulas for the variance are also proved for Monte-Carlo cross-validation, also known as repeated cross-validation, where the parameter V is replaced by the number B of random splits of the data.

Keywords: V -fold cross-validation, Monte-Carlo cross-validation, leave-one-out, leave- p -out, resampling penalties, density estimation, model selection, penalization

1. Introduction

Cross-validation methods are widely used in machine learning and statistics, for estimating the risk of a given statistical estimator (Stone, 1974; Allen, 1974; Geisser, 1975) and for selecting among a family of estimators. For instance, cross-validation can be used for model selection, where a collection of linear spaces is given (the models) and the problem is to choose the best least-squares estimator over one of these models. Cross-validation is also often used for choosing hyperparameters of a given learning algorithm. We refer to Arlot and Celisse (2010) for more references about cross-validation for model selection.

Model selection can target two different goals: (i) *estimation*, that is, minimizing the risk of the final estimator, which is the goal of AIC and related methods, or (ii) *identification*, that is, identifying the smallest true model in the family considered, assuming it exists and

it is unique, which is the goal of BIC for instance; see the survey by Arlot and Celisse (2010) for more details about this distinction. These two goals cannot be attained simultaneously in general (Yang, 2005).

We assume throughout the paper that the goal of model selection is estimation. We refer to Yang (2006, 2007) and Celisse (2014) for some results and references on cross-validation methods with an identification goal.

Then, a natural question arises: which cross-validation method should be used for minimizing the risk of the final estimator? For instance, a popular family of cross-validation methods is V -fold cross-validation (Geisser, 1975, often called k -fold cross-validation), which depends on an integer parameter V , and enjoys a smaller computational cost than other classical cross-validation methods. The question becomes (1) which V is optimal, and (2) can we do almost as well as the optimal V with a small computational cost, that is, a small V ? Answering the second question is particularly useful for practical applications where the computational power is limited.

Surprisingly, few theoretical results exist for answering these two questions, especially with a non-asymptotic point of view (Arlot and Celisse, 2010). In short, it is proved in least-squares regression that at first order, V -fold cross-validation is suboptimal for model selection (with an estimation goal) if V stays bounded, because V -fold cross-validation is biased (Arlot, 2008). When correcting for the bias (Burman, 1989; Arlot, 2008), we recover asymptotic optimality whatever V , but without any theoretical result distinguishing among values of V in second order terms in the risk bounds (Arlot, 2008).

Intuitively, if there is no bias, increasing V should reduce the variance of the V -fold cross-validation estimator of the risk, hence reduce the risk of the final estimator, as supported by some simulation experiments (Arlot, 2008, for instance). But variance computations for unbiased V -fold methods have only been made in the asymptotic framework for a fixed estimator, and they focus on risk estimation instead of model selection (Burman, 1989).

This paper aims at providing theoretical grounds for the choice of V by two means: a non-asymptotic oracle inequality valid for any V (Section 3) and exact variance computations shedding light on the influence of V on the variance (Section 5). In particular, we would like to understand why the common advice in the literature is to take $V = 5$ or 10 , based on simulation experiments (Breiman and Spector, 1992; Hastie et al., 2009, for instance).

The results of the paper are proved in the least-squares density estimation framework, because we can then benefit from explicit closed-form formulas and simplifications for the V -fold criteria. In particular, we show that V -fold cross-validation and all leave- p -out methods are particular cases of V -fold penalties in least-squares density estimation (Lemma 1).

The first main contribution of the paper (Theorem 5) is an oracle inequality with leading constant $1 + \varepsilon_n$, with $\varepsilon_n \rightarrow 0$ as $n \rightarrow \infty$ for unbiased V -fold methods, which holds for any value of V . To the best of our knowledge, Theorem 5 is the first non-asymptotic oracle inequality for V -fold methods enjoying such properties: the leading constant $1 + \varepsilon_n$ is new in density estimation, and the fact that it holds whatever the value of V had never been obtained in any framework. Theorem 5 relies on a new concentration inequality for the V -fold penalty (Proposition 4). Note that Theorem 5 implicitly assumes that the oracle loss is of order $n^{-\alpha}$ for some $\alpha \in (0, 1)$, that is, the setting is nonparametric; otherwise,

Theorem 5 may not imply the asymptotic optimality of V -fold penalization. Let us also emphasize that the leading constant is $1 + \varepsilon_n$ whatever V for unbiased V -fold methods, with ε_n independent from V in Theorem 5. So, second-order terms must be taken into account for understanding how the model selection performance depends on V . Section 4 proposes a heuristic for comparing these second order terms thanks to variance comparisons. This motivates our next result.

The second main contribution of the paper (Theorem 6) is the first non-asymptotic variance computation for V -fold criteria that allows to understand precisely how the *model selection performance* of V -fold cross-validation or penalization depends on V . Previous results only focused on the variance of the V -fold criterion (Bunman, 1989; Bengio and Grandvalet, 2005; Celisse, 2008, 2014; Celisse and Robin, 2008), which is not sufficient for our purpose, as explained in Section 4. In our setting, we can explain, partly from theoretical results, partly from a heuristic argument, why taking, say, $V > 10$ is not necessary for getting a performance close to the optimum, as supported by experiments on synthetic data in Section 6.

An oracle inequality and exact formulas for the variance are also proved for other cross-validation methods: Monte-Carlo cross-validation, also known as repeated cross-validation, where the parameter V is replaced by the number B of random splits of the data (Section 8.1), and hold-out penalization (Section 8.2).

Notation. For any integer $k \geq 1$, $[k]$ denotes $\{1, \dots, k\}$.

For any vector $\xi_{[n]} := (\xi_1, \dots, \xi_n)$ and any $B \subset [n]$, ξ_B denotes $(\xi_i)_{i \in B}$, $|B|$ denotes the cardinality of B and $B^c = [n] \setminus B$.

For any real numbers t, u , we define $t \vee u := \max\{t, u\}$, $u_+ := u \vee 0$ and $u_- := (-u) \vee 0$. All asymptotic results and notation $o(\cdot)$ or $\mathcal{O}(\cdot)$ are for the regime when the number n of observations tends to infinity.

2. Least-Squares Density Estimation and Definition of V -Fold Procedures

This section introduces the framework of the paper, the main procedures studied, and some useful notation.

2.1 General Statistical Framework

Let ξ_1, \dots, ξ_n be independent random variables taking value in a Polish space \mathcal{X} , with common distribution P and density s with respect to some known measure μ . Suppose that $s \in L^\infty(\mu)$, which implies that $s \in L^2(\mu)$. The goal is to estimate s from $\xi_{[n]} = (\xi_1, \dots, \xi_n)$, that is, to build an estimator $\widehat{s} = \widehat{s}(\xi_{[n]}) \in L^2(\mu)$ such that its loss $\|\widehat{s} - s\|^2$ is as small as possible, where for any $t \in L^2(\mu)$, $\|t\|^2 := \int_{\mathcal{X}} t^2 d\mu$.

Projection estimators are among the most classical estimators in this framework (see, for example, DeVore and Lorentz, 1993 and Massart, 2007). Given a separable linear subspace S_m of $L^2(\mu)$ (called a model), the projection estimator of s onto S_m is defined by

$$\widehat{s}_m := \operatorname{argmin}_{t \in S_m} \{\|t\|^2 - 2P_n(t)\}, \quad (1)$$

where P_n is the empirical measure; for any $t \in L^2(\mu)$, $P_n(t) = \int t dP_n = \frac{1}{n} \sum_{i=1}^n t(\xi_i)$. The quantity minimized in the definition of \widehat{s}_m is often called the empirical risk, and can be

denoted by

$$P_n \gamma(t) = \|t\|^2 - 2P_n(t) \quad \text{where } \forall x \in \mathcal{X}, \forall t \in L^2(\mu), \quad \gamma(t; x) = \|t\|^2 - 2t(x).$$

The function γ is called the least-squares contrast. Note that $S_m \subset L^1(P)$ since $s \in L^2(\mu)$.

2.2 Model Selection

When a finite collection of models $(S_m)_{m \in \mathcal{M}_n}$ is given, following Massart (2007), we want to choose from data one among the corresponding projection estimators $(\widehat{s}_m)_{m \in \mathcal{M}_n}$. The goal is to design a model selection procedure $\widehat{m} : \mathcal{X}^n \rightarrow \mathcal{M}_n$ so that the final estimator $\widehat{s} := \widehat{s}_{\widehat{m}}$ has a quadratic loss as small as possible, that is, comparable to the oracle loss $\inf_{m \in \mathcal{M}_n} \|\widehat{s}_m - s\|^2$. This goal is what is called the estimation goal in the Introduction. More precisely, we aim at proving that an oracle inequality of the form

$$\|\widehat{s}_{\widehat{m}} - s\|^2 \leq C_n \inf_{m \in \mathcal{M}_n} \{\|\widehat{s}_m - s\|^2\} + R_n$$

holds with a large probability. The procedure \widehat{m} is called asymptotically optimal when R_n is much smaller than the oracle loss and $C_n \rightarrow 1$, as $n \rightarrow +\infty$. In order to avoid trivial cases, we will always assume that $|\mathcal{M}_n| \geq 2$.

In this paper, we focus on model selection procedures of the form

$$\widehat{m} := \operatorname{argmin}_{m \in \mathcal{M}_n} \{\operatorname{crit}(m)\},$$

where $\operatorname{crit} : \mathcal{M}_n \rightarrow \mathbb{R}$ is some data-driven criterion. Since our goal is to satisfy an oracle inequality, an ideal criterion is

$$\operatorname{crit}_{\text{ideal}}(m) = \|\widehat{s}_m - s\|^2 - \|s\|^2 = -2P(\widehat{s}_m) + \|\widehat{s}_m\|^2 = P\gamma(\widehat{s}_m).$$

Penalization is a popular way of designing a model selection criterion (Barron et al., 1999; Massart, 2007)

$$\operatorname{crit}(m) = P_n \gamma(\widehat{s}_m) + \operatorname{pen}(m)$$

for some penalty function $\operatorname{pen} : \mathcal{M}_n \rightarrow \mathbb{R}$, possibly data-driven. From the ideal criterion $\operatorname{crit}_{\text{ideal}}$, we get the ideal penalty

$$\operatorname{pen}_{\text{ideal}}(m) := \operatorname{crit}_{\text{ideal}}(m) - P_n \gamma(\widehat{s}_m) = (P - P_n) \gamma(\widehat{s}_m) = 2(P_n - P)(\widehat{s}_m) \quad (2)$$

$$= 2(P_n - P)(\widehat{s}_m - s_m) + 2(P_n - P)(s_m) = 2\|\widehat{s}_m - s_m\|^2 + 2(P_n - P)(s_m),$$

$$\text{where } s_m := \operatorname{argmin}_{t \in S_m} \{\|t - s\|^2\}$$

is the orthogonal projection of s onto S_m in $L^2(\mu)$. Let us finally recall some useful and classical reformulations of the main term in the ideal penalty (2), that proves in particular the last equality in Eq. (2): If $\mathbb{B}_m = \{t \in S_m \text{ s.t. } \|t\| \leq 1\}$ and $(\psi_\lambda)_{\lambda \in \Lambda_m}$ denotes an orthonormal basis of S_m in $L^2(\mu)$, then

$$\begin{aligned} (P_n - P)(\widehat{s}_m - s_m) &= \sum_{\lambda \in \Lambda_m} [(P_n - P)(\psi_\lambda)]^2 \\ &= \|\widehat{s}_m - s_m\|^2 = \sup_{t \in \mathbb{B}_m} [(P_n - P)(t)]^2, \end{aligned} \quad (3)$$

where the last equality follows from Eq. (30) in Appendix A.

2.3 V -Fold Cross-Validation

A standard approach for model selection is cross-validation. We refer the reader to Arlot and Celisse (2010) for references and a complete survey on cross-validation for model selection. This section only provides the minimal definitions and notation necessary for the remainder of the paper.

For any subset $A \subset \llbracket n \rrbracket$, let

$$P_n^{(A)} := \frac{1}{|A|} \sum_{t \in A} \delta_t \quad \text{and} \quad \hat{s}_m^{(A)} := \operatorname{argmin}_{t \in S_m} \left\{ \|t\|^2 - 2P_n^{(A)}(t) \right\}.$$

The main idea of cross-validation is data splitting: some $T \subset \llbracket n \rrbracket$ is chosen, one first trains $\hat{s}_m(\cdot)$ with ξ_T , then test the trained estimator on the remaining data ξ_{T^c} . The hold-out criterion is the estimator of $\operatorname{crit}_{\text{id}}(m)$ obtained with this principle, that is,

$$\operatorname{crit}_{\text{HO}}(m, T) := P_n^{(T^c)} \gamma \left(\hat{s}_m^{(T)} \right) = -2P_n^{(T^c)} \left(\hat{s}_m^{(T)} \right) + \|\hat{s}_m^{(T)}\|^2, \quad (4)$$

and all cross-validation criteria are defined as averages of hold-out criteria with various subsets T .

Let $V \in \{2, \dots, n\}$ be a positive integer and let $\mathcal{B} = \mathcal{B}_{\llbracket V \rrbracket} = (\mathcal{B}_1, \dots, \mathcal{B}_V)$ be some partition of $\llbracket n \rrbracket$. The V -fold cross-validation criterion is defined by

$$\operatorname{crit}_{\text{VFCV}}(m, \mathcal{B}) := \frac{1}{V} \sum_{K=1}^V \operatorname{crit}_{\text{HO}}(m, \mathcal{B}_K^c).$$

Compared to the hold-out, one expects cross-validation to be less variable thanks to the averaging over V splits of the sample into $\xi_{\mathcal{B}_K}$ and $\xi_{\mathcal{B}_K^c}$.

Since $\operatorname{crit}_{\text{VFCV}}(m, \mathcal{B})$ is known to be a biased estimator of $\mathbb{E}[\operatorname{crit}_{\text{id}}(m)]$, Burman (1989) proposed the bias-corrected V -fold cross-validation criterion

$$\operatorname{crit}_{\text{corr,VFCV}}(m, \mathcal{B}) := \operatorname{crit}_{\text{VFCV}}(m, \mathcal{B}) + P_n \gamma \left(\hat{s}_m \right) - \frac{1}{V} \sum_{K=1}^V P_n \gamma \left(\hat{s}_m^{\mathcal{B}_K^c} \right).$$

In the particular case where $V = n$, this criterion is studied by Massart (2007, Section 7.2.1, p. 204–205) under the name cross-validation estimator.

2.4 Resampling-Based and V -Fold Penalties

Another approach for building general data-driven model selection criteria is penalization with a resampling-based estimator of the expectation of the ideal penalty, as proposed by Efron (1983) with the bootstrap and later generalized to all resampling schemes (Arlot, 2009). Let $W \sim \mathcal{W}$ be some random vector of \mathbb{R}^n independent from $\xi_{\llbracket n \rrbracket}$ with

$$\frac{1}{n} \sum_{i=1}^n W_i = 1,$$

and denote by $P_n^W = n^{-1} \sum_{i=1}^n W_i \delta_{\xi_i}$ the weighted empirical distribution of the sample. Then, the resampling-based penalty associated with \mathcal{W} is defined as

$$\operatorname{pen}_{\mathcal{W}}(m) := C_{\mathcal{W}} \mathbb{E}_{\mathcal{W}} \left[\left(P_n - P_n^W \right) \gamma \left(\hat{s}_m^W \right) \right], \quad (5)$$

where $\hat{s}_m^W \in \operatorname{argmin}_{\xi \in S_m} \{ P_n^W \gamma(t) \}$, $\mathbb{E}_{\mathcal{W}}[\cdot]$ denotes the expectation with respect to W only (that is, conditionally to the sample $\xi_{\llbracket n \rrbracket}$), and $C_{\mathcal{W}}$ is some positive constant. Resampling-based penalties have been studied recently in the least-squares density estimation framework (Lerasle, 2012), assuming that W is exchangeable, that is, its distribution is invariant by any permutation of its coordinates.

Since computing exactly $\operatorname{pen}_{\mathcal{W}}(m)$ has a large computational cost in general for exchangeable W , some non-exchangeable resampling schemes were introduced by Arlot (2008), inspired by V -fold cross-validation: given some partition $\mathcal{B} = \mathcal{B}_{\llbracket V \rrbracket}$ of $\llbracket n \rrbracket$, the weight vector W is defined by $W_i = (1 - \operatorname{Card}(\mathcal{B}_J)/n)^{-1} \mathbb{1}_{i \notin \mathcal{B}_J}$ for some random variable J with uniform distribution over $\llbracket V \rrbracket$. Then, $P_n^W = P_n^{\mathcal{B}_J^c}$ so that the associated resampling penalty, called V -fold penalty, is defined by

$$\begin{aligned} \operatorname{pen}_{\text{VF}}(m, \mathcal{B}, x) &:= \frac{x}{V} \sum_{K=1}^V \left[\left(P_n - P_n^{\mathcal{B}_K^c} \right) \gamma \left(\hat{s}_m^{\mathcal{B}_K^c} \right) \right] \\ &= \frac{2x}{V} \sum_{K=1}^V \left(P_n^{\mathcal{B}_K^c} - P_n \right) \left(\hat{s}_m^{\mathcal{B}_K^c} \right) \end{aligned} \quad (6)$$

where $x > 0$ is left free for flexibility, which is quite useful according to Lemma 1 below.

2.5 Links Between V -Fold Penalties, Resampling Penalties and (Corrected) V -Fold Cross-Validation

In this paper, we focus our study on V -fold penalties because Lemma 1 below shows that formula (6) covers all V -fold and resampling-based procedures mentioned in Sections 2.3 and 2.4.

First, when $V = n$, the only possible partition is $\mathcal{B}_{\text{LOO}} = \{\{1\}, \dots, \{n\}\}$, and the V -fold penalty is called the leave-one-out penalty $\operatorname{pen}_{\text{LOO}}(m, x) := \operatorname{pen}_{\text{VF}}(m, \mathcal{B}_{\text{LOO}}, x)$. The associated weight vector W is exchangeable, hence Eq. (6) leads to all exchangeable resampling penalties since they are all equal up to a deterministic multiplicative factor in the least-squares density estimation framework when $\sum_{i=1}^n W_i = n$, as proved by Lerasle (2012).

For V -fold methods, let us assume \mathcal{B} is a regular partition of $\llbracket n \rrbracket$, that is,

$$V = |\mathcal{B}| \geq 2 \text{ divides } n \quad \text{and} \quad \forall K \in \llbracket V \rrbracket, |\mathcal{B}_K| = \frac{n}{V}. \quad (\text{Reg})$$

Then, we get the following connection between V -fold penalization and cross-validation methods.

Lemma 1 *For least-squares density estimation with projection estimators, under assumption (Reg),*

$$\operatorname{crit}_{\text{corr,VFCV}}(m, \mathcal{B}) = P_n \gamma \left(\hat{s}_m \right) + \operatorname{pen}_{\text{VF}}(m, \mathcal{B}, V - 1) \quad (7)$$

$$\text{critVRCV}(m, \mathcal{B}) = P_n \gamma(\widehat{s}_m) + \text{pen}_{\text{VF}} \left(m, \mathcal{B}, V - \frac{1}{2} \right) \quad (8)$$

$$\text{critLPO}(m, p) = P_n \gamma(\widehat{s}_m) + \text{pen}_{\text{LPO}} \left(m, p, \frac{n}{p} - \frac{1}{2} \right) \quad (9)$$

$$\begin{aligned} &= P_n \gamma(\widehat{s}_m) + \text{pen}_{\text{LOO}} \left(m, (n-1) \frac{n/p-1/2}{n/p-1} \right) \\ &= P_n \gamma(\widehat{s}_m) + \text{pen}_{\text{VF}} \left(m, \mathcal{B}_{\text{LOO}}, (n-1) \frac{n/p-1/2}{n/p-1} \right) \end{aligned} \quad (10)$$

where for any $p \in \llbracket n-1 \rrbracket$, the leave- p -out cross-validation criterion is defined by

$$\text{critLPO}(m, p) := \frac{1}{|\mathcal{E}_p|} \sum_{A \in \mathcal{E}_p} P_n^{(A)} \gamma(\widehat{s}_m^{(A)}) \quad \text{with} \quad \mathcal{E}_p := \{A \subset \llbracket n \rrbracket \text{ s.t. } |A| = p\}$$

and the leave- p -out penalty is defined by

$$\forall x > 0, \quad \text{pen}_{\text{LPO}}(m, p, x) := \frac{x}{|\mathcal{E}_p|} \sum_{A \in \mathcal{E}_p} \left(P_n - P_n^{(A)} \right) \gamma(\widehat{s}_m^{(A)}) .$$

Lemma 1 is proved in Section A.1.

Remark 2 Eq. (7) was first proved by Arlot (2008) in a general framework that includes least-squares density estimation, assuming only **(Reg)**. Eq. (10) follows from Lemsle (2012, Lemma A.11) since pen_{LPO} belongs to the family of exchangeable resampling penalties, with weights $W_i := (1-p/n)^{-1} \mathbb{1}_{i \notin A}$ and A is randomly chosen uniformly over \mathcal{E}_p ; note that $\sum_{i=1}^n W_i = n$ for these weights. It can also be deduced from Proposition 3.1 by Celisse (2014), see Section A.1.

Remark 3 It is worth mentioning here the cross-validation estimators studied by Massart (2007, Chapter 7). First, the unbiased cross-validation criterion defined by Rademio (1982) is exactly $\text{crit}_{\text{corr,VRCV}}(m, \mathcal{B}_{\text{LOO}}$) (see also Massart, 2007, Section 7.2.1). Second, the penalized estimator of Massart (2007, Theorem 7.6) is the estimator selected by the penalty

$$\text{pen}_{\text{LOO}} \left(m, \frac{(1+\epsilon)^6(n-1)^2}{2[n-(1+\epsilon)^6]} \right)$$

for some $\epsilon > 0$ such that $(1+\epsilon)^6 < n$ (see Section A.1 for details).

So, in the least-squares density estimation framework and assuming only **(Reg)**, Lemma 1 shows that it is sufficient to study V -fold penalization with a free multiplicative factor x in front of the penalty for studying also V -fold cross-validation ($x = V-1/2$), corrected V -fold cross-validation ($x = V-1$), the leave- p -out ($V = n$ and $x = (n-1)(n/p-1/2)/(n/p-1)$) and all exchangeable resampling penalties. For any $C > 0$ and \mathcal{B} some partition of $\llbracket n \rrbracket$ into V pieces, taking $x = C(V-1)$, the V -fold penalization criterion is denoted by

$$C_{(C,\mathcal{B})}(m) := P_n \gamma(\widehat{s}_m) + \text{pen}_{\text{VF}}(m, \mathcal{B}, C(V-1)) . \quad (11)$$

A key quantity in our results is the bias $\mathbb{E}[C_{(C,\mathcal{B})}(m)] - \text{crit}_{\text{id}}(m)$. From Lemma 13 in Section A.2, we have

$$\mathbb{E}[\text{pen}_{\text{VF}}(m, \mathcal{B}, V-1)] = \mathbb{E}[\text{pen}_{\text{id}}(m)] = 2\mathbb{E}[\|\widehat{s}_m - s_m\|^2] , \quad (12)$$

so that for any $C > 0$,

$$\mathbb{E}[C_{(C,\mathcal{B})}(m)] - \text{crit}_{\text{id}}(m) = 2(C-1)\mathbb{E}[\|\widehat{s}_m - s_m\|^2] . \quad (13)$$

In Sections 3-7, we focus our study on V -fold methods, that is, we study the performance of the V -fold penalized estimators \widehat{s}_m , defined by

$$\widehat{m} = \widehat{m}(C_{(C,\mathcal{B})}) = \underset{m \in \mathcal{M}_n}{\text{argmin}} \{C_{(C,\mathcal{B})}(m)\} , \quad (14)$$

for all values of V and $C > 1/2$. Additional results on hold-out (penalization) are given in Section 8.2 to complete the picture.

3. Oracle Inequalities

In this section, we state our first main result, that is, a non-asymptotic oracle inequality satisfied by V -fold procedures. This result holds for any divisor $V \geq 2$ of n , any constant $x = C(V-1)$ in front of the penalty with $C > 1/2$, and provides an asymptotically optimal oracle inequality for the selected estimator when $C \rightarrow 1$ (assuming the setting is non parametric). In addition, as proved by Section 2.5, it implies oracle inequalities satisfied by leave- p -out procedures for all p .

3.1 Concentration of V -Fold Penalties

Concentration is the key property to establish oracle inequalities. Let us start with some new concentration results for V -fold penalties.

Proposition 4 Let $\xi_{\llbracket n \rrbracket}$ be i.i.d. real-valued random variables with density $s \in L^\infty(\mu)$, \mathcal{B} some partition of $\llbracket n \rrbracket$ into V pieces satisfying **(Reg)**, S_m a separable linear space of measurable functions and $(\psi_\lambda)_{\lambda \in \Lambda_m}$ an orthonormal basis of S_m . Define

$$\begin{aligned} \mathbb{B}_m &= \{t \in S_m \text{ s.t. } \|t\| \leq 1\} & \Psi_m &= \sum_{\lambda \in \Lambda_m} \psi_\lambda^2 = \sup_{t \in \mathbb{B}_m} t^2 & b_m &:= \|\sqrt{\Psi_m}\|_\infty \\ \mathcal{D}_m &:= P(\Psi_m) - \|s_m\|^2 = n\mathbb{E}[\|s_m - \widehat{s}_m\|^2] , \end{aligned}$$

where \widehat{s}_m is defined by Eq. (1), and for any $x, \epsilon > 0$,

$$\rho_1(m, \epsilon, s, x, n) := \frac{\|s\|_\infty x}{\epsilon n} + \frac{(b_m^2 + \|s\|^2)x^2}{\epsilon^3 n^2} .$$

Then, an absolute constant κ exists such that for any $x \geq 0$, with probability at least $1 - 8e^{-x}$, for any $\epsilon \in (0, 1]$, the following two inequalities hold true

$$\left| \text{pen}_{\text{VF}}(m, \mathcal{B}, V-1) - \frac{2\mathcal{D}_m}{n} \right| \leq \frac{\mathcal{D}_m}{n} + \kappa \rho_1(m, \epsilon, s, x, n) \quad (15)$$

$$\left| \text{pen}_{\text{VF}}(m, \mathcal{B}, V-1) - 2\|s_m - \widehat{s}_m\|^2 \right| \leq \frac{\mathcal{D}_m}{n} + \kappa \rho_1(m, \epsilon, s, x, n) . \quad (16)$$

Proposition 4 is proved in Section A.2. Eq. (15) gives the concentration of the V -fold penalty around its expectation $2\mathcal{D}_m/n = \mathbb{E}[\text{pen}_{\text{id}}(m)]$, see Eq. (12). Eq. (16) gives the concentration of the V -fold penalty around the ideal penalty, see Eq. (2). Optimizing over ϵ , the first order of the deviations of $\text{pen}_{\text{VF}}(m, \mathcal{B}, V-1)$ around $\text{pen}_{\text{id}}(m)$ is driven by $\sqrt{\mathcal{D}_m/n}$. The deviation term in Proposition 4 does not depend on V and cannot therefore help to discriminate between different values of this parameter.

3.2 Example: Histogram Models

Histograms on \mathbb{R} provide some classical examples of collections of models. Let \mathcal{X} be a measurable subset of \mathbb{R} , μ denote the Lebesgue measure on \mathcal{X} and m be some countable partition of \mathcal{X} such that $\mu(\lambda) > 0$ for any $\lambda \in m$. The histogram space S_m based on m is the linear span of the functions $(\psi_\lambda)_{\lambda \in \Lambda_m}$ where $\Lambda_m = m$ and for every $\lambda \in m$, $\psi_\lambda = \mu(\lambda)^{-1/2} \mathbb{1}_\lambda$. More precisely, we illustrate our results with the following examples.

Example 1 (Regular histograms on $\mathcal{X} = \mathbb{R}$)

$$\mathcal{M}_n = \{m_h, h \in \llbracket n \rrbracket\} \quad \text{where} \quad \forall h \in \llbracket n \rrbracket, \quad m_h = \left\{ \left[\frac{\lambda}{h}, \frac{\lambda+1}{h} \right], \lambda \in \mathbb{Z} \right\}.$$

In Example 1, defining $d_{m_h} = h$ for every $h \in \llbracket n \rrbracket$, for every $m \in \mathcal{M}_n$, $\mathcal{D}_m = d_m - \|s_m\|^2$ since Ψ_m is constant and equal to d_m . Therefore, Proposition 4 shows that $\text{pen}_{\text{VF}}(m, \mathcal{B}, V-1)$ is asymptotically equivalent to $\text{pen}_{\text{dim}}(m) := 2d_m/n$ when $d_m \rightarrow \infty$. Penalties of the form of pen_{dim} are classical and have been studied for instance by Barron et al. (1999).

Example 2 (k -rupture points on $\mathcal{X} = [0, 1]$)

$$\mathcal{M}_n = \left\{ m_{h_{\llbracket k+1 \rrbracket}, x_{\llbracket k+1 \rrbracket}} \text{ s.t. } x_1 < \dots < x_k \in \llbracket n-1 \rrbracket \text{ and } \forall i \in \llbracket k+1 \rrbracket, h_i \in \llbracket x_i - x_{i-1} \rrbracket \right\},$$

where $x_0 = 0$, $x_{k+1} = n$ and for any $x_1, \dots, x_k \in \llbracket n-1 \rrbracket$ such that $x_1 < \dots < x_k$ and any $h_{\llbracket k+1 \rrbracket} \in \mathbb{N}^{k+1}$, $m_{h_{\llbracket k+1 \rrbracket}, x_{\llbracket k+1 \rrbracket}}$ is defined as the union

$$\bigcup_{i \in \llbracket k+1 \rrbracket} \left\{ \left[\frac{x_{i-1}}{n} + \frac{(x_i - x_{i-1})(\lambda-1)}{nh_i}, \frac{x_{i-1}}{n} + \frac{(x_i - x_{i-1})\lambda}{nh_i} \right], \lambda \in \llbracket h_i \rrbracket \right\}.$$

In other words, $m_{h_{\llbracket k+1 \rrbracket}, x_{\llbracket k+1 \rrbracket}}$ splits $[0, 1]$ into $k+1$ pieces (at the x_i), and then splits the i -th piece into h_i pieces of equal size.

In Example 2, the function Ψ_m is constant on each interval $[x_{i-1}, x_i]$, equal to h_i , therefore,

$$\mathcal{D}_m = \sum_{i=1}^{k+1} h_i \mathbb{P}(\xi \in [x_{i-1}, x_i]) - \|s_m\|^2.$$

3.3 Oracle Inequality for V -Fold Procedures

In order to state the main result, we introduce the following hypotheses:

- A uniform bound on the L^∞ norm of the L^2 ball of the models

$$\forall m \in \mathcal{M}_n, \quad b_m \leq \sqrt{n} \quad (\text{H1})$$

where we recall that $b_m := \sup_{t \in \mathbb{B}_m} \|t\|_\infty$ and $\mathbb{B}_m := \{t \in S_m, \|t\| \leq 1\}$.

- The family of the projections of s is uniformly bounded.

$$\exists a > 0, \quad \forall m \in \mathcal{M}_n, \quad \|s_m\|_\infty \leq a, \quad (\text{H2})$$

- The collection of models is nested.

$$\forall (m, m') \in \mathcal{M}_n^2, \quad S_m \cup S_{m'} \in \{S_m, S_{m'}\} \quad (\text{H2}')$$

Hereafter, we define $A := a \vee \|s\|_\infty$ when (H2) holds and $A := \|s\|_\infty$ when (H2') holds. On histogram spaces, (H1) holds if and only if $\inf_{m \in \mathcal{M}_n} \inf_{\lambda \in m} \mu(\lambda) \geq n^{-1}$, and (H2) holds with $a = \|s\|_\infty$.

Theorem 5 Let $\xi_{\llbracket n \rrbracket}$ be i.i.d. real-valued random variables with common density $s \in L^\infty(\mu)$, \mathcal{B} some partition of $\llbracket n \rrbracket$ into V pieces satisfying (Reg) and $(S_m)_{m \in \mathcal{M}_n}$ be a collection of separable linear spaces satisfying (H1). Assume that either (H2) or (H2') holds true. Let $C \in (1/2, 2]$, $\delta := 2(C-1)$ and, for any $x, \epsilon > 0$,

$$\rho_2(\epsilon, s, x, n) := \frac{Ax}{\epsilon n} + \left(1 + \frac{\|s\|^2}{n}\right) \frac{x^2}{\epsilon^3 n} \quad \text{and} \quad x_n = x + \log |\mathcal{M}_n|.$$

For every $m \in \mathcal{M}_n$, let \hat{s}_m be the estimator defined by Eq. (1) and $\tilde{s} = \hat{s}_{\hat{m}}$ where

$$\hat{m} = \hat{m}(C, \mathcal{B})$$

is defined by Eq. (14). Then, an absolute constant κ exists such that, for any $x > 0$, with probability at least $1 - e^{-x}$, for any $\epsilon \in (0, 1]$,

$$\frac{1 - \delta - \epsilon}{1 + \delta + \epsilon} \|\tilde{s} - s\|^2 \leq \inf_{m \in \mathcal{M}_n} \|\hat{s}_m - s\|^2 + \kappa \rho_2(\epsilon, s, x_n, n). \quad (17)$$

Theorem 5 is proved in Section A.3.

Taking $\epsilon > 0$ small enough in Eq. (17), Theorem 5 proves that V -fold model selection procedures satisfy an oracle inequality with large probability. The remainder term can be bounded under the following classical hypothesis

$$\exists a' > 0, \quad \forall n \in \mathbb{N}^*, \quad |\mathcal{M}_n| \leq n^{a'}. \quad (\text{H3})$$

For instance, (H3) holds in Example 1 with $a' = 1$ and in Example 2 with $a' = k$. Under (H3), the remainder term in Eq. (17) is bounded by $L(\log n)^2 / (\epsilon^3 n)$ for some $L > 0$, which is much smaller than the oracle loss in the nonparametric case.

The leading constant in the oracle inequality (17) is $(1 + \delta_+)/ (1 - \delta_-) + o(1)$ by choosing $\epsilon = o(1)$, so the first-order behaviour of the upper bound on the loss is driven by δ . An asymptotic optimality result can be derived from Eq. (17) only if $\delta = o(1)$. The meaning of

$\delta = 2(C-1)$ is the amount of bias of the V -fold penalization criterion, as shown by Eq. (13). Given this interpretation of δ , the model selection literature suggests that no asymptotic optimality result can be obtained in general when $\delta \neq o(1)$ in the nonparametric case (see, for instance, Shao, 1997). Therefore, even if the leading constant $(1 + \delta_+)/ (1 - \delta_-)$ is only an upper bound, we conjecture that it cannot be taken as small as $1 + o(1)$ unless $\delta = o(1)$; such a result can be proved in our setting using similar arguments and assumptions as the ones of Arlot (2008) for instance.

For bias-corrected V -fold cross-validation, that is, $C = 1$ hence $\delta = 0$, Theorem 5 shows a first-order optimal non-asymptotic oracle inequality, since the leading constant $(1 + \epsilon)/(1 - \epsilon)$ can be taken equal to $1 + o(1)$, and the remainder term is small enough in the nonparametric case, under assumption (H3), for instance. Such a result valid with no upper bound on V had never been obtained before in any setting.

V -fold cross-validation is also analyzed by Theorem 5, since by Lemma 1 it corresponds to $C = 1 + 1/(2(V-1))$, hence $\delta = 1/(V-1)$. When V is fixed, the oracle inequality is asymptotically sub-optimal, which is consistent with the result proved in regression by Arlot (2008). On the contrary, if $\mathcal{B} = \mathcal{B}_n$ has V_n blocs, with $V_n \rightarrow \infty$, Theorem 5 implies under assumption (H3) the asymptotic optimality of V_n -fold cross-validation in the nonparametric case.

The bound obtained in Theorem 5 can be integrated and we get

$$\frac{1 - \delta_- - \epsilon}{1 + \delta_+ + \epsilon} \mathbb{E} \left[\|\tilde{s} - s\|^2 \right] \leq \mathbb{E} \left[\inf_{m \in \mathcal{M}_n} \|\hat{s}_m - s\|^2 \right] + \kappa' \rho_2 \left(\epsilon, s, \log(\mathcal{M}_n) \right)$$

for some absolute constant $\kappa' > 0$.

Assuming $C > 1/2$ is necessary, according to minimal penalty results proved by Lerasle (2012). Assuming $C \leq 2$ only simplifies the presentation; if $C > 2$, the same proof shows that Theorem 5 holds with κ replaced by $C\kappa$.

An oracle inequality similar to Theorem 5 holds in a more general setting, as proved in a previous version of this paper (Arlot and Lerasle, 2012, Theorem 1); we state a less general result here for simplifying the exposition, since it does not change the message of the paper. First, assumption (Reg) can be relaxed into assuming the partition \mathcal{B} is close to regular, that is,

$$\mathcal{B} \text{ is a partition of } \llbracket n \rrbracket \text{ of size } V \text{ and } \sup_{k \in \llbracket V \rrbracket} \left| \text{Card}(\mathcal{B}_k) - \frac{n}{V} \right| \leq 1, \quad (\text{Reg})$$

which can hold for any $V \in \llbracket n \rrbracket$. Second, data ξ_1, \dots, ξ_n can belong to a general Polish space \mathcal{X} , at the price of some additional technical assumption.

3.4 Comparison with Previous Works on V -Fold Procedures

Few non-asymptotic oracle inequalities have been proved for V -fold penalization or cross-validation procedures.

Concerning cross-validation, previous oracle inequalities are listed in the survey by Arlot and Celisse (2010). In the least-squares density estimation framework, oracle inequalities were proved by van der Laan et al. (2004) in the V -fold case, but compared the risk of the selected estimator with the risk of an oracle trained with $n(V-1)/V$ data. In comparison,

Theorem 5 considers the strongest possible oracle, that is, trained with n data. Optimal oracle inequalities were proved by Celisse (2014) for leave- p -out estimators with $p \ll n$, a case also treated in Theorem 5 by taking $V = n$ and $C = (n/p - 1/2)/(n/p - 1)$ as shown by Lemma 1. If $p \ll n$, $C \sim 1$, hence $\delta = o(1)$ and we recover the result of Celisse (2014).

Concerning V -fold penalization, previous results were either valid for $V = n$ only—by Massart (2007, Theorem 7.6) and Lerasle (2012) for least-squares density estimation, by Arlot (2009) for regression estimators—, or for V bounded when n tends to infinity—by Arlot (2008) for regression estimators. In comparison, Theorem 5 provides a result valid for all V , except for the assumption that V divides n , which can be removed (Arlot and Lerasle, 2012). In particular, the loss bound by Arlot (2008) deteriorates when V grows, while it remains stable in our result. Our result is therefore much closer to the typical behavior of the loss ratio $\|\tilde{s} - s\|^2 / \inf_{m \in \mathcal{M}_n} \|\hat{s}_m - s\|^2$ of V -fold penalization, which usually decreases as a function of V in simulation experiments, see Section 6 and the experiments by Arlot (2008), for instance.

Theorem 5 may not satisfactorily address the parametric setting, that is, when the collection $(\mathcal{S}_m)_{m \in \mathcal{M}_n}$ contains some fixed true model. In such a case, the usual way to obtain asymptotic optimality is to use a model selection procedure targeting identification, that is, taking $C \rightarrow +\infty$ when $n \rightarrow +\infty$. For instance, Celisse (2014, Theorem 3.3) shows that $\log(n) \ll C \ll n$ is a sufficient condition for such a result.

4. How to Compare Theoretically the Performances of Model Selection Procedures for Estimation?

The main goal of the paper is to compare the model selection performances of several (V -fold) cross-validation methods, when the goal is estimation, that is, minimizing the loss $\|\hat{s}_m - s\|^2$ of the final estimator. In this section, we discuss how such a comparison can be made on theoretical grounds, in a general setting.

For some data-driven function $\mathcal{C} : \mathcal{M}_n \rightarrow \mathbb{R}$, the goal is to understand how $\|\hat{s}_{\hat{m}(\mathcal{C})} - s\|^2$ depends on \mathcal{C} when the selected model is

$$\hat{m}(\mathcal{C}) \in \underset{m \in \mathcal{M}_n}{\text{argmin}} \{ \mathcal{C}(m) \}. \quad (18)$$

From now on, in this section, \mathcal{C} is assumed to be a cross-validation estimator of the risk, but the heuristic developed here applies to the general case.

Ideal comparison. Ideally, for proving that \mathcal{C}_1 is a better method than \mathcal{C}_2 in some setting, we would like to prove that

$$\|\hat{s}_{\hat{m}(\mathcal{C}_1)} - s\|^2 < (1 - \epsilon_n) \|\hat{s}_{\hat{m}(\mathcal{C}_2)} - s\|^2 \quad (19)$$

with a large probability, for some $\epsilon_n \geq 0$.

Previous works and their limits. When the goal is estimation, the classical way to analyze the performance of a model selection procedure is to prove an oracle inequality, that is, to *upper bound* (with a large probability or in expectation)

$$\|\hat{s}_{\hat{m}(\mathcal{C})} - s\|^2 - \inf_{m \in \mathcal{M}_n} \left\{ \|\hat{s}_m - s\|^2 \right\} \quad \text{or} \quad \mathfrak{R}_n(\mathcal{C}) := \frac{\|\hat{s}_{\hat{m}(\mathcal{C})} - s\|^2}{\inf_{m \in \mathcal{M}_n} \left\{ \|\hat{s}_m - s\|^2 \right\}}.$$

Alternatively, asymptotic results show that when n tends to infinity, $\mathfrak{R}_n(\mathcal{C}) \rightarrow 1$ (asymptotic optimality of \mathcal{C}) or $\mathfrak{R}_n(\mathcal{C}_1) \sim \mathfrak{R}_n(\mathcal{C}_2)$ (asymptotic equivalence of \mathcal{C}_1 and \mathcal{C}_2); see Arlot and Celisse (2010, Section 6) for a review of such results. Nevertheless, proving Eq. (19) requires a lower bound on $\mathfrak{R}_n(\mathcal{C})$ (asymptotic or not), which has been done only once for some cross-validation method, to the best of our knowledge. In some least-squares regression setting, V -fold cross-validation (\mathcal{C}^{VF}) performs (asymptotically) worse than all asymptotically optimal model selection procedures since $\mathfrak{R}_n(\mathcal{C}^{VF}) \geq \kappa(V) > 1$ with a large probability (Arlot, 2008).

The major limitation of all these previous results is that they can only compare \mathcal{C}_1 to \mathcal{C}_2 at first order, that is, according to $\lim_{n \rightarrow \infty} \mathfrak{R}_n(\mathcal{C}_1)/\mathfrak{R}_n(\mathcal{C}_2)$, which only depends on the bias of $\mathcal{C}_i(m)$ ($i = 1, 2$) as an estimator of $\mathbb{E}[\|\hat{s}_m - s\|^2]$; hence, on the asymptotic ratio between the training set size and the sample size (Arlot and Celisse, 2010, Section 6). For instance, the leave- p -out and the hold-out with a training set of size $(n - p)$ cannot be distinguished at first order, while the leave- p -out performs much better in practice, certainly because its “variance” is much smaller.

Beyond first-order. So, we must go beyond the first-order of $\mathfrak{R}_n(\mathcal{C})$ and take into account the variance of $\mathcal{C}(m)$. Nevertheless, proving a lower bound on $\mathfrak{R}_n(\mathcal{C})$ is already challenging at first order—probably the reason why only one has been proved up to now, in a specific setting only—so the challenge of computing a precise lower bound on the second order term of $\mathfrak{R}_n(\mathcal{C})$ seems too high for the present paper. We propose instead a heuristic showing that the variances of some quantities—depending on $(\mathcal{C}_i)_{i=1,2}$ and on \mathcal{M}_n —can be used as a proxy to a proper comparison of $\mathfrak{R}_n(\mathcal{C}_1)$ and $\mathfrak{R}_n(\mathcal{C}_2)$ at second order. Since we focus on second-order terms, from now on, we assume that \mathcal{C}_1 and \mathcal{C}_2 have the same bias, that is,

$$\forall m \in \mathcal{M}_n, \quad \mathbb{E}[\mathcal{C}_1(m)] = \mathbb{E}[\mathcal{C}_2(m)]. \quad (\text{SameBias})$$

In least-squares density estimation, given Lemma 1, this means that for $i \in \{1, 2\}$,

$$\mathcal{C}_i = \mathcal{C}_{(C, \mathcal{B}_i)}$$

as defined by Eq. (11), with different partitions \mathcal{B}_i satisfying **(Reg)** with different $V = V_i$, but the same constant $C > 0$; $C = 1$ corresponds to the unbiased case.

The variance of the cross-validation criteria is not the correct quantity to look at. If we were only comparing cross-validation methods $\mathcal{C}_1, \mathcal{C}_2$ as estimators of $\mathbb{E}[\|\hat{s}_m - s\|^2]$ for every single $m \in \mathcal{M}_n$, we could naturally compare them through their mean squared errors. Under assumption **(SameBias)**, this would mean to compare their variances. This can be done from Eq. (23) below, but it is not sufficient to solve our problem, since it is known that the best cross-validation estimator of the risk does not necessarily yield the best model selection procedure (Breiman and Spector, 1992). More precisely, the selected model $\hat{m}(\mathcal{C})$ defined by Eq. (18) is unchanged when $\mathcal{C}(m)$ is translated by any random quantity, but such a translation does change $\text{Var}(\mathcal{C}(m))$ and can make it as large as desired. For model selection, what really matters is that

$$\text{sign}(\mathcal{C}(m_1) - \mathcal{C}(m_2)) = \text{sign}(\|\hat{s}_{m_1} - s\|^2 - \|\hat{s}_{m_2} - s\|^2)$$

as often as possible for every $(m_1, m_2) \in \mathcal{M}_n^2$, and that most mistakes in the ranking of models occur when $\|\hat{s}_{m_1} - s\|^2 - \|\hat{s}_{m_2} - s\|^2$ is small, so that $\|\hat{s}_{\hat{m}(\mathcal{C})} - s\|^2$ cannot be much larger than $\inf_{m \in \mathcal{M}_n} \{\|\hat{s}_m - s\|^2\}$.

Heuristic. The heuristic we propose goes as follows. For simplicity, we assume that $m^* = \arg\min_{m \in \mathcal{M}_n} \mathbb{E}[\|\hat{s}_m - s\|^2]$ is uniquely defined. If the goal was identification, we could directly state that for any \mathcal{C} , the smaller is $\mathbb{P}(m = \hat{m}(\mathcal{C}))$ for all $m \neq m^*$, the better should be the performance of $\hat{m}(\mathcal{C})$. In this paper, our goal is estimation, but a similar claim can be conjectured by considering “all $m \in \mathcal{M}_n$ sufficiently far from m^* in terms of risk”, that is, all $m \in \mathcal{M}_n$ such that $\mathbb{E}[\|\hat{s}_m - s\|^2]$ is significantly worse than $\mathbb{E}[\|\hat{s}_{m^*} - s\|^2]$. Indeed, for any m “close to m^* ” in terms of risk, selecting m instead of m^* does not significantly change the performance of $\hat{m}(\mathcal{C})$; on the contrary, for any m “far from m^* ” in terms of risk, selecting m instead of m^* does increase significantly the risk $\mathbb{E}[\|\hat{s}_{\hat{m}(\mathcal{C})} - s\|^2]$.

Then, our idea is to find a proxy for $\mathbb{P}(m = \hat{m}(\mathcal{C}))$, that is, a quantity that should behave similarly as a function of \mathcal{C} and its “variance” properties. For all $m, m' \in \mathcal{M}_n$, let $\Delta_{\mathcal{C}}(m, m') := \mathcal{C}(m) - \mathcal{C}(m')$. \mathcal{N} some standard Gaussian random variable and, for all $t \in \mathbb{R}$, $\Phi(t) = \mathbb{P}(\mathcal{N} > t)$. Then, for every $m \in \mathcal{M}_n$

$$\begin{aligned} \mathbb{P}(\hat{m}(\mathcal{C}) = m) &= \mathbb{P}(\forall m' \neq m, \Delta_{\mathcal{C}}(m, m') < 0) \\ &\asymp \min_{m' \neq m} \mathbb{P}(\Delta_{\mathcal{C}}(m, m') < 0) \end{aligned} \quad (20)$$

$$\begin{aligned} &\approx \min_{m' \neq m} \mathbb{P}(\mathbb{E}[\Delta_{\mathcal{C}}(m, m')] + \mathcal{N}\sqrt{\text{Var}(\Delta_{\mathcal{C}}(m, m'))} < 0) \\ &= \overline{\Phi}(\text{SNR}_{\mathcal{C}}(m)) \quad \text{where} \quad \text{SNR}_{\mathcal{C}}(m) := \max_{m' \neq m} \frac{\mathbb{E}[\Delta_{\mathcal{C}}(m, m')]}{\sqrt{\text{Var}(\Delta_{\mathcal{C}}(m, m'))}}. \end{aligned} \quad (21)$$

So, if $\text{SNR}_{\mathcal{C}_1}(m) > \text{SNR}_{\mathcal{C}_2}(m)$ for all m “sufficiently far from m^* ”, \mathcal{C}_1 should be better than \mathcal{C}_2 . Assuming **(SameBias)** holds true and that

$$\{m^*\} = \arg\min_{m \in \mathcal{M}_n} \mathbb{E}[\mathcal{C}_1(m)] = \arg\min_{m \in \mathcal{M}_n} \mathbb{E}[\mathcal{C}_2(m)], \quad (\text{SameMin})$$

this leads to the following heuristic

$$\forall m \neq m', \quad \text{Var}(\Delta_{\mathcal{C}_1}(m, m')) < \text{Var}(\Delta_{\mathcal{C}_2}(m, m')) \Rightarrow \mathcal{C}_1 \text{ better than } \mathcal{C}_2. \quad (22)$$

Indeed, for every $m \neq m'$, assumption **(SameMin)** implies that $\text{SNR}_{\mathcal{C}_1}(m) > 0$ for $i = 1, 2$, hence we can restrict the max in the definition of $\text{SNR}_{\mathcal{C}_i}$ to all m' such that $\mathbb{E}[\Delta_{\mathcal{C}_i}(m, m')] > 0$. By assumption **(SameBias)**, the numerator in the definition of $\text{SNR}_{\mathcal{C}_i}$ does not depend on i , hence the ratio is maximal when the denominator is minimal, which leads to Eq. (22). Let us make some remarks.

- The quantity $\Delta_{\mathcal{C}}(m, m')$ appears in relative bounds (Catoni, 2007, Section 1.4) which can be used as a tool for model selection (Audibert, 2004).
- Assumptions **(SameBias)** and **(SameMin)** hold true in particular in the unbiased case, that is, when $\mathbb{E}[\mathcal{C}_i(m)] = \mathbb{E}[\|\hat{s}_m - s\|^2]$ for all $m \in \mathcal{M}_n$ and $i \in \{1, 2\}$.

- Assumption (**SameMin**) is necessary: Figure 3 shows an example where a larger variance corresponds to better performance under assumption (**SameBias**) alone.
- As noticed above, the heuristic (22) should apply when the goal is estimation *and* when the goal is identification, provided that (**SameBias**) and (**SameMin**) hold true. What should depend on the goal is the suitable amount of bias for $C_\alpha(m)$ as an estimator of the risk $\mathbb{E}\|\widehat{s}_m - s\|^2$.
- Approximation (20) is the strongest one. Clearly, inequality \leq holds true. The equality case occurs is for a very particular dependence setting; that is, when one among the events $(\{\Delta_C(m, m') < 0\})$, $m' \in \mathcal{M}_n$, is included into all the others. In general, the left-hand side is significantly smaller than the right-hand side; we conjecture that they vary similarly as a function of C .
- The Gaussian approximation (21) for $\Delta_C(m, m')$ does not hold exactly, but it seems reasonable to make it, at first order at least.
- The validity of approximations (20) and (21) is supported by the numerical experiments of Section 6.

In the heuristic (22), all (m, m') do not matter equally for explaining a quantitative difference in the performances of C . First, we can fix $m' = m^*$ since intuitively, the strongest candidate against any $m \neq m^*$ is m^* , which clearly holds in all our experiments, see Figures 18 and 24 in Section G of the Online Appendix. Second, as mentioned above, if m and m^* are very close, that is, $\|\widehat{s}_m - s\|^2 / \|\widehat{s}_{m^*} - s\|^2$ is smaller than the minimal order of magnitude we can expect for $\mathfrak{R}_n(C)$ with a data-driven C , taking m instead of m^* does not decrease the performance significantly. Third, if $\Phi(\text{SNR}_C(m))$ is very small, increasing it even by an order of magnitude will not affect the performance of $\widehat{m}(C)$ significantly; hence, all m such that, say, $\text{SNR}_C(m) \gg (\log(n))^\alpha$ for all $\alpha > 0$, can also be discarded. Overall, pairs (m, m') that really matter in (22) are pairs (m, m^*) that are at a “moderate distance”, in terms of $\mathbb{E}\|\widehat{s}_m - s\|^2 - \|\widehat{s}_{m^*} - s\|^2$.

5. Dependence on V of V -Fold Penalization and Cross-Validation

Let us now come back to the least-squares density estimation setting. Our goal is to compare the performance of cross-validation methods having the same bias, that is, according to Section 2.5, $\widehat{m}(C_{(C, \mathcal{B})})$ with the same constant C but different partitions \mathcal{B} , where $\widehat{m}(C_{(C, \mathcal{B})})$ is defined by Eq. (14).

Theorem 6 *Let $\xi_{[n]}$ be i.i.d. random variables with common density $s \in L^\infty(\mu)$, \mathcal{B} some partition of $[n]$ into V pieces satisfying (**Reg**), and $(\psi_\lambda)_{\lambda \in \Delta_{m_1}}$, $(\psi_\lambda)_{\lambda \in \Delta_{m_2}}$ two orthonormal families in $L^2(\mu)$. For any $m, m' \in \{m_1, m_2\}$, we define S_m the linear span of $(\psi_\lambda)_{\lambda \in \Delta_{m^*}}$, s_m the orthogonal projection of s onto S_m in $L^2(\mu)$, $\Psi_m := \sup_{t \in S_m} \text{s.t. } \|t\| \leq 1$, t^2 ,*

$$\beta(m, m') := \sum_{\lambda \in \Delta_m} \sum_{\lambda' \in \Delta_{m'}} \left(\mathbb{E} \left[(\psi_\lambda(\xi_1) - P_{\psi_\lambda})(\psi_{\lambda'}(\xi_1) - P_{\psi_{\lambda'}}) \right] \right)^2$$

and $\mathbf{B}(m_1, m_2) := \beta(m_1, m_1) + \beta(m_2, m_2) - 2\beta(m_1, m_2)$.

Then, for every $C > 0$,

$$\begin{aligned} \text{Var}(C_{(C, \mathcal{B})}(m_1)) &= \frac{2}{n^2} \left(1 + \frac{4C^2}{V-1} - \frac{2C-1}{n} \right) \beta(m_1, m_1) \\ &+ \frac{4}{n} \text{Var} \left(\left(1 + \frac{2C-1}{n} \right) s_{m_1}(\xi_1) - \frac{2C-1}{2n} \Psi_{m_1}(\xi_1) \right) \end{aligned} \quad (23)$$

$$\begin{aligned} \text{and } \text{Var}(C_{(C, \mathcal{B})}(m_1) - C_{(C, \mathcal{B})}(m_2)) &= \frac{2}{n^2} \left(1 + \frac{4C^2}{V-1} - \frac{2C-1}{n} \right) \mathbf{B}(m_1, m_2) \\ &+ \frac{4}{n} \text{Var} \left(\left(1 + \frac{2C-1}{n} \right) (s_{m_1} - s_{m_2})(\xi_1) - \frac{2C-1}{2n} (\Psi_{m_1} - \Psi_{m_2})(\xi_1) \right) \end{aligned} \quad (24)$$

where $C_{(C, \mathcal{B})}$ is defined by Eq. (11).

Theorem 6 is proved in Section A.4.

Unbiased case. When $C = 1$, Theorem 6 shows that

$$\text{Var}(C_{(1, \mathcal{B})}(m_1) - C_{(1, \mathcal{B})}(m_2)) = a + \left(1 + \frac{4}{V-1} - \frac{1}{n} \right) b$$

for some $a, b \geq 0$ depending on n, m_1, m_2 but not on V . If we admit that the heuristic (22) holds true, this implies that the model selection performance of bias-corrected V -fold cross-validation improves when V increases, but the improvement is at most in a second order term as soon as V is large. In particular, even if $a \ll b$, the improvement from $V = 2$ to 5 or 10 is much larger than from $V = 10$ to $V = n$, which can justify the commonly used principle that taking $V = 5$ or $V = 10$ is large enough.

Assuming in addition that S_{m_1} and S_{m_2} are regular histogram models (Example 1 in Section 3.2) with d_{m_1} that divides d_{m_2} , then, by Lemma 19 in Section B.2 of the Online Appendix,

$$\begin{aligned} a &= \frac{4}{n} \left(1 + \frac{1}{n} \right)^2 \text{Var}(s_{m_1}(\xi_1) - s_{m_2}(\xi_1)) \approx \mathcal{O} \left(\frac{1}{n} \|s_{m_1} - s_{m_2}\|^2 \right) \\ \text{and } b &= \frac{2}{n^2} \mathbf{B}(m_1, m_2) \asymp \|s_{m_2}\|^2 \frac{d_{m_2}}{n^2}. \end{aligned}$$

When d_{m_2}/n is at least as large as $\|s_{m_1} - s_{m_2}\|^2$, we obtain that the first-order term in the variance is of the form $\alpha + \beta/(V-1)$ where $\alpha, \beta > 0$ do not depend on V and are of the same order of magnitude, as supported by the numerical experiments of Section 6. Then, increasing V from 2 to n does reduce significantly the variance, by a constant multiplicative factor.

Let $C_{\text{id}}(m) := P_\gamma(\widehat{s}_m) + \mathbb{E}[\text{pen}_{\text{id}}(m)]$ be the criterion we could use if we knew the expectation of the ideal penalty. From Proposition 17 in Section B of the Online Appendix,

$$\begin{aligned} \text{Var}(C_{\text{id}}(m_1) - C_{\text{id}}(m_2)) &= \frac{2}{n^2} \left(1 - \frac{1}{n} \right) \mathbf{B}(m_1, m_2) \\ &+ \frac{4}{n} \text{Var} \left(\left(1 - \frac{1}{n} \right) (s_{m_1} - s_{m_2})(\xi_1) + \frac{1}{2n} (\Psi_{m_1} - \Psi_{m_2})(\xi_1) \right) \end{aligned}$$

which easily compares to formula (24) obtained for the V -fold criterion when $C = 1$. Up to smaller order terms, the difference lies in the first term, where $(1 + 4/(V - 1) - 1/n)$ is replaced by $(1 - 1/n)$ when using the expectation of the ideal penalty instead of a V -fold penalty. In other words, the leave-one-out penalty—that is, taking $V = n$ —behaves like the expectation of the ideal penalty.

We can also compare Eq. (23) with the asymptotic results obtained by Burman (1989), which imply that for any fixed model m_1

$$\text{Var}(\mathcal{C}_{(1,\mathcal{B})}(m_1) - P\gamma(\widehat{s}_{m_1})) = \frac{\gamma_0}{n} + \left(\frac{V}{V-1}\gamma_1 + \gamma_2\right) \frac{1}{n^2} + o\left(\frac{1}{n^2}\right)$$

with $\gamma_0, \gamma_1, \gamma_2$ that depend on m_1 and $\gamma_1 > 0$. Here, putting $C = 1$ in Eq. (23) yields a result with a similar flavour, valid for all $n \geq 1$, even if Eq. (23) computes the variance of a slightly different quantity.

Cross-validation criteria. V -fold cross-validation and the leave- p -out are also covered by Theorem 6, according to Lemma 1, respectively with $C = 1 + 1/(2(V - 1))$ and with $V = n$ and $C = 1 + 1/(2(n/p - 1))$. As in the unbiased case, increasing V decreases the variance, and if we admit that the heuristic (22) holds true, V -fold cross-validation performs almost as well as the leave- (n/V) -out as soon as V is larger than 5 or 10.

Similarly, the variances of the V -fold cross-validation and leave- p -out criteria, for instance, can be derived from Eq. (23). In the leave- p -out case, we recover formulas obtained by Celisse (2014) and Celisse and Robin (2008), with a different grouping of the variance components; Eq. (23) clearly emphasizes the influence of the bias—through $(C - 1)$ —on the variance. For V -fold cross-validation, we believe that Eq. (23) shows in a simpler way how the variance depends on V , compared to the result of Celisse and Robin (2008) which was focusing on the difference between V -fold cross-validation and the leave- (n/V) -out; here the difference can be written

$$\frac{8}{n^2} \left(\frac{1}{V-1} - \frac{1}{n-1} \right) \left(1 + \frac{1}{2(V-1)} \right)^2 \beta(m_1, m_1) .$$

A major novelty in Eq. (23) is also to cover a larger set of criteria, such as bias-corrected V -fold cross-validation. Note that $\text{Var}(\mathcal{C}_{(C,\mathcal{B})}(m_1))$ is generally much larger than

$$\text{Var}(\mathcal{C}_{(C,\mathcal{B})}(m_1) - \mathcal{C}_{(C,\mathcal{B})}(m_2)) ,$$

which illustrates again why computing the former quantity might not help for understanding the model selection properties of $\mathcal{C}_{(C,\mathcal{B})}$, as explained in Section 4. For instance, comparing Eq. (23) and (24), changing s_{m_1} into $s_{m_1} - s_{m_2}$ in the second term can reduce dramatically the variance when s_{m_1} and s_{m_2} are close, which happens for the pairs (m_1, m_2) that matter for model selection according to Section 4.

The variance of other criteria and their increments are computed in subsequent sections of the paper and in the Online Appendix: Monte-Carlo cross-validation (Theorem 10 in Section 8.1 and Theorem 24 in Section C.4) and hold-out penalization (Proposition 28 in Section D.2).

Remark 7 The term $\mathbf{B}(m_1, m_2)$ does not depend on the choice of particular bases of S_{m_1} and S_{m_2} ; as proved by Proposition 18 in Section B of the Online Appendix

$$\mathbf{B}(m_1, m_2) = n \text{Var}((\widehat{s}_{m_1} - \widehat{s}_{m_2})(\xi)) - (n+1) \text{Var}((s_{m_1} - s_{m_2})(\xi)) .$$

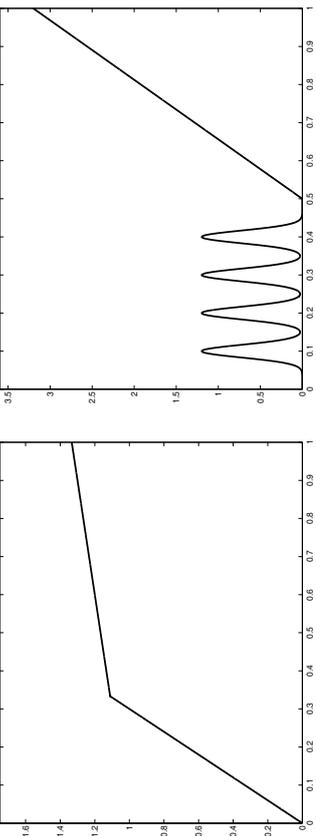


Figure 1: The two densities considered. Left: setting L. Right: setting S.

6. Simulation Study

This section illustrates the main theoretical results of the paper with some experiments on synthetic data.

6.1 Setting

In this section, we take $\mathcal{X} = [0, 1]$ and μ is the Lebesgue measure on \mathcal{X} . Two examples are considered for the target density s and for the collection of models $(S_m)_{m \in \mathcal{M}_n}$.

Two density functions s are considered, see Figure 1:

- Setting L: $s(x) = \frac{10x}{3} \mathbb{1}_{0 \leq x < 1/3} + (1 + \frac{x}{3}) \mathbb{1}_{1/3 \leq x < 1/2}$.
- Setting S: s is the mixture of the piecewise linear density $x \mapsto (8x - 4) \mathbb{1}_{1/2 \leq x < 1/2}$ (with weight 0.8) and four truncated Gaussian densities with means $(k/10)_{k=1, \dots, 4}$ and standard deviation $1/60$ (each with weight 0.05).

Two collections of models are considered, both leading to histogram estimators: for every $m \in \mathcal{M}_n$, S_m is the set of piecewise constant functions on some partition Λ_m of \mathcal{X} .

- “Regu” for regular histograms: $\mathcal{M}_n = \{1, \dots, n\}$ where for every $m \in \mathcal{M}_n$, Λ_m is the regular partition of $[0, 1]$ into m bins.

- “Dya2” for dyadic regular histograms with two bin sizes and a variable change-point:
$$\mathcal{M}_n = \bigcup_{k \in \{1, \dots, \tilde{n}\}} \{k\} \times \left\{ 0, \dots, \lfloor \log_2(k) \rfloor \right\} \times \left\{ 0, \dots, \lfloor \log_2(\tilde{n} - k) \rfloor \right\}$$

where $\tilde{n} = \lfloor n / \log(n) \rfloor$ and for every $(k, i, j) \in \mathcal{M}_n$, $\Lambda_{(k,i,j)}$ is the union of the regular partition of $[0, k/\tilde{n}]$ into 2^i pieces and the regular partition of $[k/\tilde{n}, 1]$ into 2^j pieces.

The difference between “Regu” and “Dya2” can be visualized on Figure 2, on which the corresponding oracle estimators \widehat{s}_{m^*} have been plotted for one sample in setting S, where

$$\widehat{m}^* \in \underset{m \in \mathcal{M}_n}{\text{argmin}} \|\widehat{s}_m - s\|^2 .$$

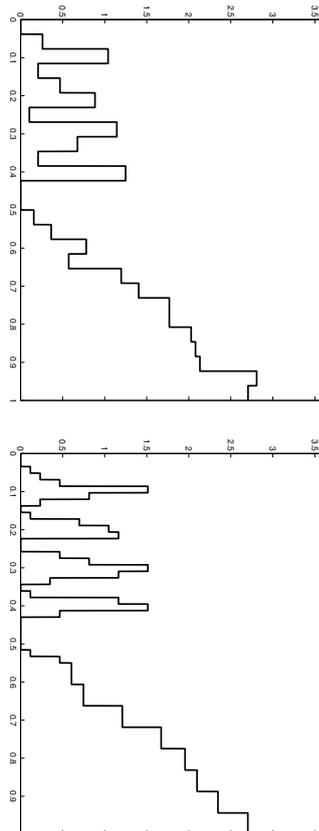


Figure 2: Oracle estimator for one sample of size $n = 500$, in setting S. Left: Regu. Right: Dya2.

Setting	Oracle(Regu)	Oracle(Dya2)	Best(Regu)	Best(Dya2)
L	13.4 ± 0.1	5.46 ± 0.02	25.8 ± 0.1	19.4 ± 0.1
S	62.4 ± 0.1	43.9 ± 0.1	100.9 ± 0.2	83.4 ± 0.2

Table 1: Comparison of Regu and Dya2: quadratic risks $\mathbb{E}[\|\hat{s}_m - s\|^2]$ of “Oracle” and “Best” estimators (multiplied by 10^3) with the two collections of models. “Best” means that \hat{m} is the data-driven procedure minimizing $\mathbb{E}[\|\hat{s}_m - s\|^2]$ among all the data-driven procedures we considered in our experiments (see Section 6.2). “Oracle” means that $\hat{m} \in \arg\min_{m \in \mathcal{M}_n} \|\hat{s}_m - s\|^2$ is the oracle model for each sample.

While “Regu” is one of the simplest and most classical collections for density estimation, the flexibility of “Dya2” allows to adapt to the variability of the smoothness of s . Intuitively, in settings L and S, the optimal bin size is smaller on $[0, 1/2]$ (where s is varying fastly) than on $[1/2, 1]$ (where $|s'|$ is much smaller).

Another point of comparison of Regu and Dya2 is given by Table 1, that reports values of the quadratic risks obtained depending on the collection of models considered. Table 1 shows that in settings L and S, the collection Dya2 helps reducing the quadratic risk by approximately 20% (when comparing the best data-driven procedures of our experiment), and even more when comparing oracle estimators (30% in setting S, 59% in setting L). Therefore, in settings L and S, it is worth considering more complex collections of models (such as Dya2) than regular histograms.

Let us finally remark that Dya2 does not reduce the quadratic risk in all settings as significantly as in settings L and S. We performed similar experiments with a few other density functions, sometimes leading to less important differences between Regu and Dya2 in terms of risk (results not shown). The oracle model was always better with Dya2, but in

two cases, the risk of the best data-driven procedure with Dya2 was larger than with Regu by 6 to 8%.

6.2 Procedures Compared

In each setting, we consider the following model selection procedures:

- pen_{dm} (Barron et al., 1999): penalization with $\text{pen}(m) = 2 \text{Card}(\Lambda_m)/n$.
- V -fold cross-validation with $V \in \{2, 5, 10, n\}$, see Section 2.3.
- V -fold penalties (with leading constant $x = V - 1$, that is, bias-corrected V -fold cross-validation), for $V \in \{2, 5, 10, n\}$, see Section 2.4.
- for comparison, penalization with $\mathbb{E}[\text{pen}_d(m)]$, that is, $\hat{m}(C_{1d})$.

Since it is often suggested to multiply the usual penalties by some factor larger than one (Arlot, 2008), we consider all penalties above multiplied by a factor $C \in [0, 10]$. Complete results can be found in Section G of the Online Appendix.

6.3 Model Selection Performances

In each setting, all procedures are compared on $N = 10000$ independent synthetic data sets of size $n = 500$. For measuring their respective model selection performances, for each procedure $\hat{m}(C)$ we estimate

$$C_{\text{or}}(C) := \mathbb{E}[\mathfrak{R}_n(C)] = \mathbb{E} \left[\frac{\|\hat{s}_{\hat{m}(C)} - s\|^2}{\inf_{m \in \mathcal{M}_n} \|\hat{s}_m - s\|^2} \right]$$

by the corresponding average over the N simulated data sets; $C_{\text{or}}(C)$ represents the constant that would appear in front of an oracle inequality. The uncertainty of estimation of $C_{\text{or}}(C)$ is measured by the empirical standard deviation of $\mathfrak{R}_n(C)$ divided by \sqrt{N} . The results are reported in Table 2 for settings L and S, with the collection Dya2.

Results for Regu are not reported here since dimensionality-based penalties are already known to work well with Regu (Lerasle, 2012), so V -fold methods cannot improve significantly their performance, with a larger computational cost. Complete results (including Regu, with $n = 100$ and $n = 500$) are given in Tables 3 and 4 in Section G of the Online Appendix, showing that the performances of pen_{dm} and V -fold methods indeed are very close.

Performance as a function of V . Let us first consider V -fold penalization. In both settings L and S, as suggested by our theoretical results, C_{or} decreases when V increases. The improvement is large when V goes from 2 to 5 (27% for L, 10% for S) and small when V goes from 5 to 10 and when V goes from 10 to $n = 500$ (each time, 8% for L, 2% for S). Since the main influence of V is on the variance of the V -fold penalty, these experiments support our interpretation of Theorem 6 in Section 5: increasing V helps much more from 2 to 5 or 10 than from 10 to n .

The picture is less clear for V -fold cross-validation, for which almost no difference is observed among $V \in \{2, 5, 10, n\}$ —less than 2%—, and C_{or} is minimized for $V \in \{5, 10\}$.

Procedure	L-Dya2	S-Dya2
pen _{dim}	8.27 ± 0.07	3.21 ± 0.01
pen2F	10.21 ± 0.08	2.39 ± 0.01
pen5F	7.47 ± 0.06	2.16 ± 0.01
pen10F	6.89 ± 0.06	2.11 ± 0.01
penLOO	6.35 ± 0.05	2.06 ± 0.01
2FCV	6.41 ± 0.05	2.05 ± 0.01
5FCV	6.27 ± 0.05	2.05 ± 0.01
10FCV	6.24 ± 0.05	2.05 ± 0.01
LOO	6.34 ± 0.05	2.06 ± 0.01
$\mathbb{E}[\text{pen}_{\text{id}}]$	6.52 ± 0.05	2.07 ± 0.01

Table 2: Estimated model selection performances, see text. ‘LOO’ is a shortcut for ‘leave-one-out’, that is, V -fold with $V = n = 500$.

Indeed, increasing V simultaneously decreases the bias and the variance of the V -fold cross-validation criterion, leading to various possible behaviours of C_{or} as a function of V , depending on the setting. The same phenomenon has been observed in regression (Arlot, 2008).

Overpenalization. In all settings considered in this paper, V -fold penalization performs much better when multiplying the penalty by $C > 1$, as illustrated by Figure 3. In particular, the best overpenalization factor for pen_{LOO} is $C_n^* \approx 2.5$ for L-Dya2 and $C_n^* \approx 1.4$ for S-Dya2, when $n = 500$. Such a phenomenon, which can also be observed in regression (Arlot, 2008), is related to the fact that some nonparametric model selection problems are “practically parametric”, using the terminology of Liu and Yang (2011), that is, BIC beats AIC and the optimal C is closer to $\log(n)/2$ than to 1. For instance, Figure 3 shows that L-Dya2 is practically parametric, while S-Dya2 is practically nonparametric since AIC beats BIC and the optimal C is close to 1.

Given an overpenalization factor C close to its optimal value C_n^* , V -fold penalization performs significantly better than V -fold cross-validation in settings S-Dya2 and L-Dya2 (Figure 3). Since V -fold cross-validation corresponds to taking

$$C = C^{\text{VF}}(V) := 1 + \frac{1}{2(V-1)}$$

according to Lemma 1, this mostly means that $C^{\text{VF}}(V)$ is not close to C_n^* in these settings. In addition, when $C \approx C_n^*$ is fixed, increasing V always improves the performance of V -fold penalization, as predicted by the heuristic of Section 4 and the theoretical results of Section 5. Let us emphasize that this fact does not depend on the parametricness of the setting: although the value of C_n^* is quite different for S-Dya2 and L-Dya2, in both cases, we observe qualitatively the same relationship between V and the performance of the procedure.

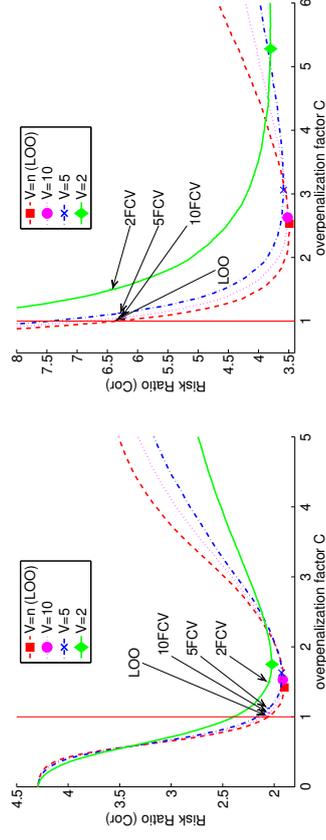


Figure 3: Overpenalization in settings S-Dya2 (left) and L-Dya2 (right), with $n = 500$ in both cases. Each plot represents the estimated model selection performance $C_{\text{or}}(\mathcal{C}(C, \mathcal{E}))$ of several penalization procedures, as a function of the overpenalization constant C ; unbiased risk estimation ($C = 1$) is materialized by a vertical red line. For each value of V , the estimated optimal value of C is shown on the graph; some arrows also show the performance of V -fold cross-validation, that is, $C = 1 + 1/2(V-1)$. Error bars are not shown for clarity; Table 2 shows their order of magnitude, which is smaller than visible differences in the above graph. The performance obtained with the penalty $\mathbb{E}[\text{pen}_{\text{id}}(m)]$ (not shown on the graph) is almost the same as with the leave-one-out penalty.

The results reported in Section G of the Online Appendix lead to similar conclusions in several other settings, as well as unshown results in a truly parametric setting, with a true model of dimension 2. Although a wider simulation study would be necessary to get general conclusions, this suggests at least that the heuristic of Section 4 and the theoretical results of Section 5 can be applied to both parametric and nonparametric settings.

Figure 3 also helps understanding how the performance of V -fold cross-validation depends on V in Table 2. Indeed, the performance of V -fold cross-validation for each value of V can be visualized on Figure 3 by taking the point of abscissa $C = C^{VF}(V)$ on the curve associated with V -fold penalization. Two phenomena are coupled when $C \leq C_n^*$, which always holds in our simulations for V -fold cross-validation since $\max_V C^{VF}(V) = 1.5$ and the estimated value of C_n^* is always larger. (i) The performance improves when V is fixed and C gets closer to C_n^* . (ii) The performance improves when C is fixed and V increases. Even if both phenomena (i) and (ii) seem quite universal, their coupling can result in various behaviours for V -fold cross-validation as a function of V , as shown by Table 3 in Section G of the Online Appendix for instance.

Other comments.

- `pendim` performs much worse than V -fold penalization (except $V = 2$ in setting L) with the collection `Dya2`. On the contrary, `pendim` does well with `Regu` (see Table 3 in Section G of the Online Appendix), but V -fold penalization then performs as well.
- In other settings considered in a preliminary phase of our experiments, for V -fold penalization, differences between $V = 2$ and $V = 5$ were sometimes smaller or not significant, but always with the same ordering (that is, the worse performance for $V = 2$ when C is fixed). In a few settings, for which the “change-point” in the smoothness of s was close to the median of sq_{L_s} , we found `pendim` among the best procedures with collection `Dya2`; then, V -fold penalization and cross-validation always had a performance very close to `pendim`. Both phenomena lead us to discard all settings for which there were no significant difference to comment.

6.4 Variance as a Function of V

We now illustrate the results of Section 5 about the variance of V -fold penalization and the heuristic of Section 4 about its influence on model selection. We focus on the unbiased case, that is, criteria $C_{(1,\mathcal{B})}$ with partitions \mathcal{B} satisfying **(Reg)**. Since the distribution of $(C_{(1,\mathcal{B})}(m))_{m \in \mathcal{M}_n}$ then only depends on $V = |\mathcal{B}|$, we write C_V instead of $C_{(1,\mathcal{B})}$ by abuse of notation. All results presented in this subsection have been obtained from $N = 10\,000$ independent samples in setting S with a sample size $n = 100$ and the collection `Regu`—for which models are naturally indexed by their dimension.

First, Figure 4 shows the variance of $\Delta_{C_V}(m, m^*) = C_V(m) - C_V(m^*)$ as a function of the dimension m of S_m , illustrating the conclusions of Theorem 6: the variance decreases when V increases. More precisely, the variance decrease is significant between $V = 2$ and $V = 5$, an order of magnitude smaller between $V = 5$ and $V = 10$ and between $V = 10$ and $V = n$, while the leave-one-out C_n is hard to distinguish from the ideal penalized criterion

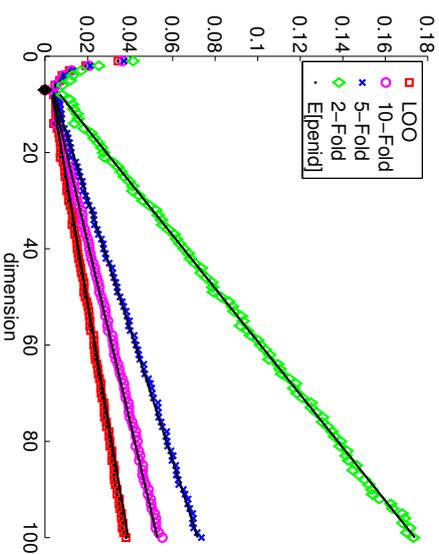


Figure 4: Illustration of the variance heuristic: $\text{Var}(\Delta c(m, m^*))$ as a function of m for five different C . Setting S-Regu, $n = 100$. The black diamond shows $m^* = 7$. The black lines show the linear approximation $n^{-2} [29(1 + \frac{0.81}{V-1}) + 3 \cdot 7(1 + \frac{3.8}{V-1})(m - m^*)]$ for $m > m^*$.

\mathcal{C}_{id} . On Figure 4, we can remark that for $m > m^*$

$$\text{Var}(\Delta_{C_V}(m, m^*)) \approx \frac{1}{m^2} \left[K_1 \left(1 + \frac{K_2}{V-1} \right) + K_3 \left(1 + \frac{K_4}{V-1} \right) (m - m^*) \right]$$

with $K_1 \approx 29$, $K_2 \approx 0.81$, $K_3 \approx 3.7$ and $K_4 \approx 3.8$. The shape of the dependence on V already appears in Theorem 6, the above formula clarifies the relative importance of the terms called a and b in Section 5, and their dependence on the dimension m of S_m . Remark that the same behaviour holds when $n = 500$ with very close values for K_3 and K_4 (see Figure 25 in Section G of the Online Appendix), as well as in setting L with $m = 100$ or $n = 500$ with $K_3 \approx 2.1$ and $K_4 \approx 4.2$ (see Figures 19 and 30 in Section G of the Online Appendix). The fact that K_4 is close to 4 in both settings supports that the term $1 + 4/(V-1)$ appearing Theorem 6 indeed drives how $\text{Var}(\Delta_{C_V}(m, m^*))$ depends on V .

Figures 5 and 6 respectively show $\mathbb{P}(\hat{m}(\mathcal{C}) = m)$ and its proxy $\bar{\Phi}(\text{SNR}_{\mathcal{C}}(m))$ as a function of m for $\mathcal{C} = C_V$ with $V \in \{2, 5, 10, n\}$ and for $\mathcal{C} = C_{\text{id}}$. First, we remark that both quantities behave similarly as a function of m and \mathcal{C} —see also Figure 16 in Section G of the Online Appendix—supporting empirically the heuristic of Section 4. The decrease of the variance observed on Figure 4 when V increases here translates into a better concentration of the distribution of $\hat{m}(C_V)$ around m^* , which can explain the performance improvement observed in Section 6.3. Figures 5–6 actually show how the decrease of the variance quantitatively influences the distribution of $\hat{m}(C_V)$: $\hat{m}(C_5)$ is significantly more concentrated than $\hat{m}(C_2)$, while the difference between $V = 10$ and $V = 5$ is much smaller and comparable to the difference between $V = n$ and $V = 10$; C_n is hard to distinguish from C_{id} . Similar experiments with $n = 500$ and in setting L are reported in Section G of the Online Appendix, leading to similar conclusions.

7. Fast Algorithm for Computing V -Fold Penalties for Least-Squares Density Estimation

Since the use of V -fold algorithms is motivated by computational reasons, it is important to discuss the actual computational cost of V -fold penalization and cross-validation as a function of V . In the least-squares density estimation framework, two approaches are possible: a naive one—valid for all other frameworks—, and a faster one—specific to least-squares density estimation. For clarifying the exposition, we assume in this section that **(Reg)** holds true—so, V divides n . The general algorithm for computing the V -fold penalized criterion and/or the V -fold cross-validation criterion consists in training the estimator with data sets $(\xi_i)_{i \notin B_j}$ for $j = 1, \dots, V$ and then testing each trained estimator on the data sets $(\xi_i)_{i \in B_j}$ and/or $(\xi_i)_{i \notin B_j}$. In the least-squares density estimation framework, for any model S_m given through an orthonormal family $(\psi_\lambda)_{\lambda \in \Lambda_m}$ of elements of $L^2(\mu)$, we get the “naive” algorithm described and analysed more precisely in Section E.1 of the Online Appendix, whose complexity is of order $nV \text{Card}(\Lambda_m)$.

Several simplifications occur in the least-squares density estimation framework, that allow to avoid a significant part of the computations made in the naive algorithm.

Algorithm 1

Input: \mathcal{B} some partition of $\{1, \dots, n\}$ satisfying **(Reg)**, $\xi_1, \dots, \xi_n \in \mathcal{X}$ and $(\psi_\lambda)_{\lambda \in \Lambda_m}$ a finite orthonormal family of $L^2(\mu)$.

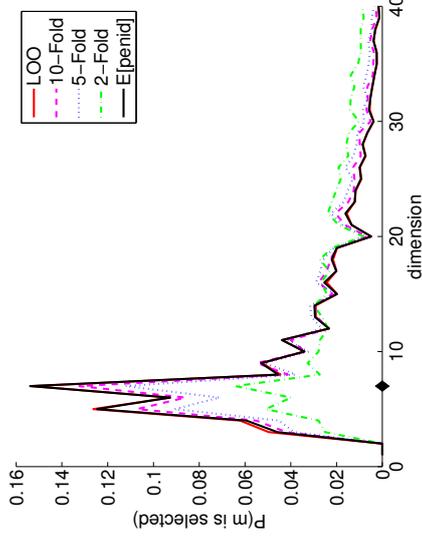


Figure 5: $\mathbb{P}(\hat{m}(\mathcal{C}) = m)$ as a function of m for five different \mathcal{C} . Setting S-Regu, $n = 100$. The black diamond shows $m^* = 7$.

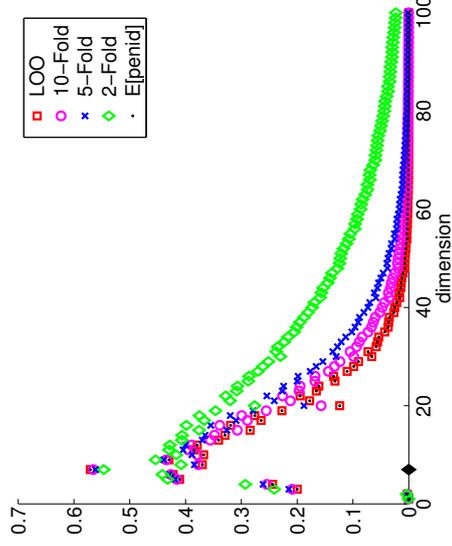


Figure 6: Illustration of the variance heuristic: $\bar{\Phi}(\text{SNR}_{\mathcal{C}}(m))$ as a function of m for five different \mathcal{C} . Setting S-Regu, $n = 100$. The black diamond shows $m^* = 7$.

1. For $i \in \{1, \dots, V\}$ and $\lambda \in \Delta_m$, compute $A_{i,\lambda} := \frac{1}{n} \sum_{j \in B_i} \psi_\lambda(\xi_j)$.
2. For $i, j \in \{1, \dots, V\}$, compute $C_{i,j} := \sum_{\lambda \in \Delta_m} A_{i,\lambda} A_{j,\lambda}$.
3. Compute $S := \sum_{1 \leq i, j \leq V} C_{i,j}$ and $\mathcal{T} := \text{tr}(C)$.

Output:

Empirical risk: $P_n \gamma(\widehat{s}_m) = \frac{-S}{V^2}$;

V -fold cross-validation criterion: $\text{crit}_{V\text{FCV}}(m) = \frac{\mathcal{T}}{V(V-1)} - \frac{S-\mathcal{T}}{(V-1)^2}$;

V -fold penalty: $\text{pen}_{V\text{F}}(m) = (\text{crit}_{V\text{FCV}}(m) - P_n \gamma(\widehat{s}_m)) \frac{V-1/2}{V-1}$.

To the best of our knowledge, Algorithm 1 is new, even for computing the V -fold cross-validation criterion. Its correctness and complexity are analyzed with the following proposition.

Proposition 8 *Algorithm 1 is correct and has a computational complexity of order*

$$(n + V^2) \text{Card}(\Delta_m) .$$

In the histogram case, that is, when Δ_m is a partition of \mathcal{X} and $\forall \lambda \in \Delta_m$, $\psi_\lambda = \mu(\lambda)^{-1/2} \mathbb{1}_\lambda$, the computational complexity of Algorithm 1 can be reduced to the order of $n + V^2 \text{Card}(\Delta_m)$.

Proposition 8 is proved in Section E.2 of the Online Appendix. It shows that Algorithm 1 is significantly faster than the “naive” algorithm, by a factor of order

$$\frac{nV}{n + V^2} = \left(\frac{1}{V} + \frac{V}{n} \right)^{-1} \ll 1 \quad \text{if} \quad 1 \ll V \ll n .$$

Note that closed-form formulas are available for the leave- p -out criterion in least-squares density estimation (Celisse, 2014), allowing to compute it with a complexity of order $n \text{Card}(\Delta_m)$ in general, and smaller in some particular cases—for instance, n for histograms.

8. Discussion

Before discussing how to choose V when using V -fold methods for model selection—our more generally for choosing among a given family of estimators—, we state some additional results and we discuss the model selection literature in least-squares density estimation.

8.1 Monte-Carlo Cross-Validation

Our analysis of V -fold procedures for model selection can be extended to some other cross-validation procedures. We here present results for Monte-Carlo cross-validation (MCCV, Picard and Cook, 1984), also known as repeated cross-validation, where B training samples of the same size $n - p$ are chosen independently and uniformly (see also Ahtol and Celisse, 2010, Section 4.3.2). Formally, we consider the criterion

$$\text{crit}_{\text{CV}}(m, (TK)_{1 \leq K \leq B}) := \frac{1}{B} \sum_{K=1}^B \text{crit}_{\text{HO}}(m, TK) , \quad (25)$$

where T_1, \dots, T_B are subsets of $[n]$ and we recall that the hold-out criterion is defined by Eq. (4). We make the following three assumptions throughout this subsection

$$\begin{aligned} \exists p \in [n-1], \quad \forall j \in [B], \quad |T_j| = n - p = n\tau_n, & \quad (\text{SameSize}) \\ (TK)_{1 \leq K \leq B} \text{ is independent from } D_n, & \quad (\text{Ind}) \\ T_1, \dots, T_B \text{ are independent with uniform distribution over } \mathcal{E}_{n-p}, & \quad (\text{MCCV}) \end{aligned}$$

where we recall that $\mathcal{E}_{n-p} = \{A \subset [n] \text{ s.t. } |A| = n - p\}$. Under these assumptions, we write $\text{crit}_{\text{MCCV}}(m)$ as a shortcut for $\text{crit}_{\text{CV}}(m, (TK)_{1 \leq K \leq B})$.

Similarly to Theorem 5, we prove in Section C.3 of the Online Appendix the following oracle inequality for MCCV.

Theorem 9 *Let $\xi_{[n]}$ be i.i.d. real-valued random variables with common density $s \in L^\infty(\mu)$, $(TK)_{1 \leq K \leq B}$ some sequence of subsets of $[n]$ satisfying (SameSize), (Ind) and (MCCV) and $(S_m)_{m \in \mathcal{M}_n}$ be a collection of separable linear spaces satisfying (H1). Assume that either (H2) or (H2') holds true. For every $m \in \mathcal{M}_n$, let \widehat{s}_m be the estimator defined by Eq. (1), and $\widehat{s} = \widehat{s}_{\widehat{m}}$ where*

$$\widehat{m} \in \underset{m \in \mathcal{M}_n}{\text{argmin}} \left\{ \text{crit}_{\text{CV}}(m, (TK)_{1 \leq K \leq B}) \right\}$$

and crit_{CV} is defined by Eq. (25). Let us define, for any $x, y, \epsilon > 0$, $x_n = x + \log |\mathcal{M}_n|$ and

$$\rho_3(\epsilon, x, y, n, \tau_n, B, A) := \frac{1}{n\tau_n^2} \left(1 + \frac{B \wedge (\log n + y)}{B(1 - \tau_n)} \right)^\alpha \left(\frac{Ax}{\tau_n \epsilon} + \frac{(A \vee 1)x^2}{\epsilon^3} \right)$$

with $\alpha = 1$ under assumption (H2) and $\alpha = 2$ under assumption (H2'). Then, an absolute constant $\kappa > 0$ exists such that, for any $x, y \geq 0$, with probability at least $1 - e^{-x} - e^{-y}$, for any $\epsilon \in (0, \kappa^{-1})$,

$$\left(1 - \frac{\epsilon}{\tau_n} \right) \|\widehat{s} - s\|^2 \leq \frac{1 + \epsilon}{\tau_n} \inf_{m \in \mathcal{M}_n} \left\{ \|\widehat{s}_m - s\|^2 \right\} + \kappa \rho_3(\epsilon, x_n, y, n, \tau_n, B, A) . \quad (26)$$

Theorem 9 actually is a corollary of a more general result (Theorem 23 in Section C.3 of the Online Appendix), which is valid without assumption (MCCV) and extends therefore our previous results on V -fold cross-validation).

Very few results exist in the literature about the model selection performance of MCCV with an estimation goal. Some asymptotic optimality result has been obtained by Burman (1990) for spline regression, and some oracle inequalities comparing the risk of the selected estimator with the risk of an oracle trained with $\tau_n n < n$ data have been proved by van der Laan and Dudoit (2003) in a general framework and by van der Laan et al. (2004) for density estimation with the Kullback-Leibler loss. In comparison, Theorem 9 provides a precise non-asymptotic comparison to an oracle trained with n data.

As in Theorem 5, the leading constant of the oracle inequality (26) is directly related to the bias, which is here quantified by $\tau_n^{-1} - 1 \geq 0$ instead of δ . The remainder term ρ_3 is also comparable to ρ_2 in Theorem 5: they differ by a factor between τ_n^{-2} (when B is large enough) and $\tau_n^{-2}(1 - \tau_n)^{-\alpha}$ (when B is small). In particular, let $V \geq 2$ and assume that

$p = n/V$ in Theorem 9, hence $\tau_n = 1 - V^{-1} \in [1/2, 1)$. Then, for the hold-out ($B = 1$), ρ_3 is larger than ρ_2 by a factor V^α with $\alpha \in \{1, 2\}$. For $B = V$, MCCV with $\tau_n = 1 - V^{-1}$ can be called ‘‘Monte-Carlo V -fold’’ (MCVF); then, with $y \approx \log n$, we loose a factor at most $\log n$ for MCVF compared to V -fold cross-validation. Finally, when B is large enough, that is, larger than $V \log n$, ρ_3 and ρ_2 are of the same order.

The above comparison of remainder terms suggests a hierarchy between several cross-validation methods with a common training sample size $n-p = n\tau_n$: from the (presumably) worse to the (presumably) best procedure, the hold-out, Monte-Carlo CV with $B = V$, V -fold CV, Monte-Carlo CV with B large and the leave- p -out. Nevertheless, upper bounds comparison can be misleading, so, following the heuristics (22) presented in Section 4, we compute below the variance of $\Delta\mathcal{L}(m, m')$ when C is a Monte-Carlo CV criterion.

Theorem 10 *We consider the setting and notation of Theorem 6, and we assume that (SameSize), (MCCV) and (Ind) hold true. We recall that $C^{\text{MCCV}}(m)$ is defined above at the beginning of Section 8.1. Then, for regular histogram models m_1, m_2 (Example 1 in Section 3.2), we have*

$$\begin{aligned} \text{Var}(C^{\text{MCCV}}(m_1) - C^{\text{MCCV}}(m_2)) &= C_1^{\text{MC}}(B, n, \tau_n) \frac{2}{n^2} \mathbf{B}(m_1, m_2) \\ &\quad + C_2^{\text{MC}}(B, n, \tau_n) \frac{4}{n} \text{Var}(s_{m_1}(\xi_1) - s_{m_2}(\xi_1)) \end{aligned} \quad (27)$$

where

$$\begin{aligned} C_1^{\text{MC}}(B, n, \tau_n) &= \frac{1}{B} \left(\frac{1}{\tau_n^2} + \frac{2}{\tau_n(1-\tau_n)} - \frac{1}{n\tau_n^3} \right) + \left(1 - \frac{1}{B} \right) \left[1 + \frac{1}{n-1} \left(\frac{1}{\tau_n} + 1 \right) - \frac{1}{n\tau_n^2} \right] \\ C_2^{\text{MC}}(B, n, \tau_n) &= \frac{1}{B} \left(\frac{1}{n^2\tau_n^3} + \frac{1}{1-\tau_n} \right) + \left(1 - \frac{1}{B} \right) \left(1 + \frac{1}{n\tau_n} \right) \end{aligned}$$

and we recall that $\tau_n = \lfloor T_K \rfloor / n = 1 - (p/n)$.

Theorem 10 is proved in Section C.4 of the Online Appendix, as a corollary of a more general result, called Theorem 24, which holds for all models m_1, m_2 —not only regular histograms—and provides a formula for the variance of the criterion itself—not its increments. Let us make a few comments.

Eq. (27) is similar to the formula obtained for bias-corrected V -fold and V -fold penalization, see Eq. (24) in Theorem 6. In the particular case of regular histogram models, Eq. (24) even fits the general form of Eq. (27), with constants $C_i^{\text{penVF}}(V, n, C)$ instead of $C_i^{\text{MC}}(B, n, \tau_n)$.

Assuming the heuristics of Section 4 is valid, for m_1, m_2 which matter for model selection, the two terms $2n^{-2}\mathbf{B}(m_1, m_2)$ and $4n^{-1}\text{Var}(s_{m_1}(\xi_1) - s_{m_2}(\xi_1))$ are of the same order of magnitude (see Section 5). Then, we can compare model selection performance of several cross-validation methods by comparing the values of the constants C_i only.

In order to get a variance of the same order of magnitude as the one of bias-corrected V -fold CV—that is, constants C_i of order 1—, MCCV requires to take τ_n far enough from 0 and 1, hence training and sample sets of comparable sizes, unless B is large enough.

Eq. (27) allows to compare the hold-out ($B = 1$) with the leave- p -out ($B \rightarrow +\infty$), for a given value $n\tau_n = n - p$ of the training sample size. Let us assume for simplicity that $n \rightarrow +\infty$ and $\tau_n \gg n^{-1/2}$. Then,

$$\begin{aligned} C_1^{\text{MC}}(1, n, \tau_n) &\sim \frac{1}{\tau_n^2} + \frac{2}{\tau_n(1-\tau_n)} > 11 \quad \text{and} \quad C_2^{\text{MC}}(1, n, \tau_n) \sim \frac{1}{1-\tau_n} \geq 1 \\ \text{whereas} \quad C_1^{\text{MC}}(\infty, n, \tau_n) &\rightarrow 1 \quad \text{and} \quad C_2^{\text{MC}}(\infty, n, \tau_n) \rightarrow 1 \end{aligned}$$

which shows an improvement at least by a constant factor in general. When τ_n tends to zero—leave-most-out—or 1—such as for the leave-one-out—, the improvement is by an order of magnitude. The fact that the leave- p -out has a smaller variance than the hold-out is not surprising at all—it holds in full generality, as a consequence of Jensen’s inequality—, but the exact quantification of the improvement given by Theorem 10 is new and can be useful in practice for choosing the number of splits B when using Monte-Carlo cross-validation.

Eq. (27) also allows to compare V -fold cross-validation, given by Theorem 6 with

$$C = 1 + \frac{1}{2(V-1)},$$

with MCCV with $B = V$ and $\tau_n = (V-1)/V$, which can be named ‘‘Monte-Carlo V -fold’’ cross-validation. The only difference between the two methods is that the V splits are chosen independently for ‘‘Monte-Carlo V -fold’’, whereas the usual V -fold makes a balanced use of each observation—putting it exactly $(V-1)$ times in the training set. Let us assume for simplicity that $n \rightarrow +\infty$ while $V = V_n$ can vary with n . Then, we have

$$\begin{aligned} C_1^{\text{MCCV}}(V_n, n) &:= C_1^{\text{MC}}\left(V_n, n, \frac{V_n-1}{V_n}\right) \sim 3 + \frac{2V_n+1}{V_n(V_n-1)} + \frac{1}{(V_n-1)^2} \\ C_1^{\text{VF}}(V_n, n) &:= C_1^{\text{penVF}}\left(V_n, n, 1 + \frac{1}{2(V_n-1)}\right) \sim 1 + \frac{4}{V_n-1} + \frac{4}{(V_n-1)^2} + \frac{1}{(V_n-1)^3} \end{aligned}$$

$$\text{hence} \quad \frac{C^{\text{MCCV}}(V_n, \infty)}{C_1^{\text{VF}}(V_n, \infty)} > 1 \text{ if } V_n \geq 3, \quad \frac{C_1^{\text{MCCV}}(V_n, n)}{C_1^{\text{VF}}(V_n, n)} \xrightarrow{n, V_n \rightarrow +\infty} 3,$$

$$C_2^{\text{MCCV}}(V_n, n) := C_2^{\text{MC}}\left(V_n, n, \frac{V_n-1}{V_n}\right) \sim 2 - \frac{1}{V_n} \in \left[\frac{3}{2}, 2\right]$$

$$\text{and} \quad C_2^{\text{VF}}(V_n, n) := C_2^{\text{penVF}}\left(V_n, n, 1 + \frac{1}{2(V_n-1)}\right) \rightarrow 1.$$

Overall, we get that V -fold cross-validation has a smaller variance than ‘‘Monte-Carlo V -fold’’ for $V \geq 3$, at least for n large enough, and that the improvement is by a constant factor between 3/2 and 3. Since increasing V cannot decrease the variance of (bias-corrected) VFVCV by more than a small constant factor, the above difference between two methods with the same computational complexity is quite important. This supports strongly the use of V -fold CV methods instead of ‘‘Monte-Carlo V -fold’’. Such an improvement was previously noticed in the asymptotic computations of Burman (1989); here we show that it holds in a non-asymptotic framework, where the models m_1, m_2 can depend on n .

8.2 Hold-Out Criteria

Our analysis of cross-validation procedures for model selection can also be extended to hold-out criteria. First, let us emphasize that the hold-out criterion defined by Eq. (4) corresponds to taking $B = 1$ in the results of Section 8.1, since choosing T uniformly over \mathcal{E}_{n-p} independently from D_n is equivalent to choosing some arbitrary T of size $n-p$ before seeing the data D_n .

Second, similarly to the definition of the hold-out criterion in Eq. (4), we can define the hold-out penalty by

$$\forall x \geq 0, \quad \text{pen}_{\text{HO}}(m, T, x) := 2x \left(P_n^{(T)} - P_n \right) \left(\widehat{s}_m^{(T)} - \widehat{s}_m \right), \quad (28)$$

that is, the hold-out estimator of $\mathbb{E}[2(P_n - P)(\widehat{s}_m - s_m)]$ which is equal to the expectation of the ideal penalty, see Eq. (2). We do not define pen_{HO} by Eq. (6) with $V = 1$ and $T = \mathcal{B}_1^c$ —that is, the hold-out estimator of $\mathbb{E}[(P - P_n)\gamma(\widehat{s}_m)]$, which amounts to removing the centering term $-\widehat{s}_m$ in Eq. (28)—because this would dramatically increase its variability. Note that adding such a term $-\widehat{s}_m$ in Eq. (6) does not change the value of the V -fold penalty under **(Reg)** since $\sum_{K=1}^V (P_n^{(\mathcal{B}_K^c)} - P_n) = 0$.

Denoting by $\tau_n = |T|/n$ as in Section 8.1, it comes from Lemma 26 in Section D.1 of the Online Appendix that

$$\mathbb{E}[\text{pen}_{\text{HO}}(m, T, x)] = x \frac{1 - \tau_n}{\tau_n} \mathbb{E}[\text{pen}_{\text{d}}(m)].$$

In the following, we choose $x = C\tau_n/(1 - \tau_n)$ so that $C = 1$ corresponds to the unbiased case, as in the previous sections for the V -fold penalty.

Remark 11 Since $P_n = \tau_n P_n^{(T)} + (1 - \tau_n) P_n^{(T^c)}$, by linearity of the estimator \widehat{s}_m ,

$$\text{pen}_{\text{HO}}(m, T, x) := 2x(1 - \tau_n)^2 \left(P_n^{(T)} - P_n^{(T^c)} \right) \left(\widehat{s}_m^{(T)} - \widehat{s}_m^{(T^c)} \right)$$

which is symmetric in T and T^c , hence $\text{pen}_{\text{HO}}(m, T^c, x) = \text{pen}_{\text{HO}}(m, T, x)$. In particular, if $|T| = n/2$, the 2-fold penalty computed on the partition $\mathcal{B} = \{T, T^c\}$ and the hold-out penalty coincide

$$\forall x > 0, \quad \text{pen}_{\text{VF}}(m, \{T, T^c\}, x) = \text{pen}_{\text{HO}}(m, T, x).$$

Theorem 12 Let $\xi_{[n]}$ be i.i.d. real-valued random variables, $s \in L^\infty(\mu)$ their common density, $T \subset [n]$ with $\tau_n = |T|/n \in (0, 1)$ and $(S_m)_{m \in \mathcal{M}_n}$ be a collection of separable linear spaces satisfying **(H1)**. Assume that either **(H2)** or **(H2')** holds true. Let $C \in (1/2, 2]$ and $\delta := 2(C - 1)$. For every $m \in \mathcal{M}_n$, let \widehat{s}_m be the projection estimator onto S_m defined by Eq. (1), and $\widehat{s}_{\text{HO}} = \widehat{s}_{\text{HO}}$ where

$$\widehat{m}_{\text{HO}} = \underset{m \in \mathcal{M}_n}{\text{argmin}} \left\{ P_n \gamma(\widehat{s}_m) + \text{pen}_{\text{HO}} \left(m, T, \frac{C\tau_n}{1 - \tau_n} \right) \right\}.$$

Then, an absolute constant κ exists such that, for any $x > 0$, defining $x_n = x + \log |\mathcal{M}_n|$, with probability at least $1 - e^{-x}$, for any $\epsilon \in (0, 1]$,

$$\frac{1 - \delta - \epsilon}{1 + \delta_+ + \epsilon} \|\widehat{s}_{\text{HO}} - s\|^2 \leq \inf_{m \in \mathcal{M}_n} \|\widehat{s}_m - s\|^2 + \kappa \left(\frac{Ax_n}{cn} + \frac{\tau_n^2 + (1 - \tau_n)^2 x_n^2}{\tau_n(1 - \tau_n)} \frac{x_n^2}{c^2 \tau_n} \right). \quad (29)$$

Theorem 12 is proved in Section D.1 of the Online Appendix.

Theorem 12 extends Theorem 5 to hold-out penalties, under similar assumptions. As in Theorem 5, δ quantifies the bias of the hold-out penalized criterion, and plays the same role in the leading constant of the oracle inequality (29).

We can compare the results obtained for hold-out and V -fold penalization in Theorems 5 and 12. For this comparison, let V be some divisor of n , $T \subset [n]$ such that $|T| = n - n/V$ and choose the same C so that both criteria have the same bias δ . Then, the only difference lies in the remainder term, the one in Eq. (29) is larger than the one of Eq. (17) in Theorem 5 by a factor of order V when V is large. These only are upper bounds, but at least they are consistent with the common intuition about the stabilizing effect of averaging over V folds. We can also compare the results obtained for hold-out penalization in Theorem 12 and for the hold-out criterion in Theorem 9. First, hold-out penalization gives a flexibility to choose an unbiased criterion and therefore to obtain asymptotically optimal oracle inequalities while hold-out criteria are always biased for fixed τ_n , hence a leading constant $\tau_n^{-1} > 1$ in the oracle inequality. The loss in the remainder term is also smaller in Eq. (29) than in Eq. (26) by a factor of order $\tau_n^{-1}(1 - \tau_n)^{-1}$ under assumption **(H2')**.

Similarly to Theorems 6 and 10, the variance terms can be computed for the hold-out penalty in order to understand separately the roles of the training sample size and of averaging over the V splits, in the V -fold criteria. Detailed results are given by Proposition 28 in Section D.2 of the Online Appendix.

8.3 Other Oracle Inequalities for Least-Squares Density Estimation

Although the primary topic of the paper is the study of V -fold procedures, let us compare briefly our results to other oracle inequalities that have been proved in the least-squares density estimation setting. For projection estimators, Massart (2007, Section 7.2) proves an oracle inequality for some penalization procedures, which are suboptimal since the leading constant C_n does not tend to 1 as n goes to $+\infty$. Oracle inequalities have also been proved for other estimators: blockwise Stein estimators (Rigollet, 2006), linear estimators (Goldenshluger and Lepski, 2011) and some T -estimators (Birgé, 2013). The models considered by Birgé (2013) are more general than ours, but the corresponding estimators are not computable in practice, and the oracle inequality by Birgé (2013) also has a suboptimal constant C_n . Some aggregation procedures also satisfy oracle inequalities (Rigollet and Tsybakov, 2007; Bunea et al., 2010). Overall, under our assumptions, none of these results imply strictly better bounds than ours.

Let us finally mention that Birgé and Rozenholc (2006) propose a precise evaluation of the penalty term in the case of regular histogram models and the log-likelihood contrast. Their final penalty is a function of the dimension, only slightly modified compared to $\text{pen}_{\text{d}}^{\text{lim}}$ performing very well on regular histograms. These performances are likely to become much worse on the collection Dyad2 presented in Section 6. This can be seen, for example, in Table 3 in Section G of the Online Appendix, where we present the performances of $\text{pen}_{\text{d}}^{\text{lim}}$ with different over-penalizing constants.

8.4 Conclusion on the Choice of V

This section summarizes the results of the paper in order to address the main question we would like to answer: How to choose a V -fold procedure for model selection?

Generality of the results. The results of the paper only hold for projection estimators in least-squares density estimation, but we conjecture that most of the statements below are valid much more generally. At least, they have been observed experimentally for projection estimators in least-squares regression (Arlot, 2008) and they are supported by theoretical results for kernel density estimators (Magalhães, 2015, Chapters 3–4). Nevertheless, it is reported in the literature that V -fold cross-validation can behave differently in other settings (Arlot and Celisse, 2010), so we must keep in mind that the statements below may not be universal.

Let us also recall that we focus here on model selection with an estimation goal, that is, minimizing the risk of the final estimator; see Yang (2006, 2007) and Celisse (2014) for results when the goal is identification.

Choice of a model selection procedure. Choosing among procedures of the form $\widehat{m}(\mathcal{C})$, as defined by Eq. (18), requires to take into account three quantities:

- *the bias of $\mathcal{C}(m)$* as an estimator of the risk of \widehat{s}_m for every $m \in \mathcal{M}_n$, or equivalently, the *overpenalization factor C* , which usually drives the performance at first order when $n \rightarrow +\infty$, as in Theorem 5. The simulation experiments of Section 6 also show that varying C can strongly change the performance of the procedure. In all settings considered in the paper, some C_n^* exists (the optimal overpenalization constant) such that the performance decreases for $C \in [0, C_n^*]$ and increases for $C > C_n^*$ (Figure 3). Note that C_n^* strongly depends on the setting, and can also vary with V when using V -fold penalization (in particular from $V = 2$ to $V \geq 5$). In the nonparametric case, when $n \rightarrow +\infty$, Theorem 5 shows that $C_n^* \sim 1$. On the contrary, in the parametric case, when $n \rightarrow +\infty$, it is known that a BIC-type penalty performs better, hence $C_n^* \rightarrow +\infty$. For a finite sample size, Section 6 and Liu and Yang (2011) show that some nonparametric settings can be “practically parametric”, that is, C_n^* can be much larger than 1.
- *the variance of increments $\mathcal{C}(m) - \mathcal{C}(m')$* drives the performance $\widehat{m}(\mathcal{C})$ at second order, according to the heuristic of Section 4, which suggests that this variance should be minimized, at least for a given “good enough” value of the overpenalization factor C .
- *the computational complexity of the procedure $\widehat{m}(\mathcal{C})$* , that we want to minimize—for a given statistical performance—or on which some upper bound is given—fixed budget.

V -fold cross-validation. The paper analyzes how the above three terms depend on V when $\mathcal{C} = C_{VFCV}^V$ is a V -fold cross-validation procedure, under assumption **(Reg)**. First, by Lemma 1, its overpenalization factor is $C^{VF}(V) = 1 + 1/[2(V - 1)] \in [1, 3/2]$, which decreases to 1 as V increases to $+\infty$. Second, by Theorem 6, its variance decreases as V increases. Theoretical and empirical arguments in Sections 5 and 6 show that the variance almost reaches its minimal value by taking, say, $V = 5$ or $V = 10$. Third, by Section 7,

its computational complexity is proportional to V in general; in the least-squares density estimation setting, it can be reduced to $(n + V^2) \text{Card}(\Lambda_m)$.

These three results can explain why the most common advices for choosing V in the literature (for instance Breiman and Spector, 1992; Hastie et al., 2009, Section 7.10.1) are between $V = 5$ and $V = 10$. Indeed, taking V larger does not reduce the variance significantly—with almost no impact on the risk of the final estimator—and it reduces the overpenalization factor although C_n^* is often larger than $C^{VF}(10) = 19/18$ or $C^{VF}(5) = 9/8$. So, if C_n^* is not much larger than $1 + 1/8$, which is likely to occur in many nonparametric settings, taking $V = 5$ or 10 can be close to be optimal.

Nevertheless, other situations can occur, for instance in (practically) parametric settings where C_n^* is much larger, possibly leading to the failure of the heuristic “ $5 \leq V \leq 10$ is almost optimal”. More generally, understanding precisely how C_{VFCV}^{VF} performs as a function of V seems to be a difficult question: V influences the performance in two opposite directions simultaneously, through the bias and the variance, so that various behaviours can result from this coupling of bias and variance, as shown in the simulation experiments.

V -fold penalization. Lemma 1 shows that a natural way to solve this difficulty is to consider instead a V -fold penalization procedure $\mathcal{C}_{(C,V)}^{\text{penVF}}$, with overpenalization factor $C > 0$. The value $C = C^{VF}(V)$ corresponds to V -fold cross-validation, but any other value of C can also be considered, making it easier to understand. Indeed, the overpenalization factor is directly given by C , while the variance and computational complexity of $\mathcal{C}_{(C,V)}^{\text{penVF}}$ vary with V —independently from C —exactly as for V -fold cross-validation. So, V should be taken as large as possible—depending on the maximal computational budget available—, while C should be taken as close as possible to C_n^* .

Compared to V -fold cross-validation, another interest of V -fold penalization is the improvement of the performance for a given computational cost, that is, a given value of V , because it is then possible to take C closer to C_n^* than $C^{VF}(V)$. This is especially true in (practically) parametric settings for which $C_n^* > 3/2 \geq C^{VF}(V)$ for all $V \geq 2$.

Data-driven overpenalization factor C . Although the paper shows that choosing well C is a key practical problem, making an optimal data-driven choice of C remains an open question which deserves to be studied, even independently from the analysis of cross-validation procedures. We postpone such a study to future works, but we can already make two suggestions. First, an external cross-validation loop can be used for choosing C , if the computational power is not a limitation. Second, a procedure built for choosing between AIC and BIC can be used in order to detect whether C should be close to 1 or significantly larger (see, for instance, Liu and Yang, 2011 and references therein).

Acknowledgments

The authors thank the two referees for their comments that allowed us to improve the paper. The authors thank gratefully Yannick Baraud and Guillaume Obozinski for precious comments on an earlier version of the paper, and Nelo Magalhães for his careful reading and helpful remarks on this earlier version. Let us also emphasize that this paper has changed a lot since its first version (Arlot and Lerasle, 2012), and that the most recent

results (Section 8.1, Section C of the Online Appendix) and some aspects of the proofs (for instance, the systematic use of $U_m(x; y)$ and $K_m(x; y)$) have been strongly influenced by our ongoing collaboration with Nello Magalhães (Chapters 3–4 Magalhães, 2015).

This work was done while the first author was financed by CNRS and member of the Sierra team in the Département d'Informatique de l'École normale supérieure (CNRS/ENS/INRIA UMR 8548), 45 rue d'Ulm, F-75230 Paris Cedex 05, France. The authors acknowledge the support of the French Agence Nationale de la Recherche (ANR) under reference ANR-09-JCJC-0027-01 (DETRECT project) and ANR 2011 BSO1 010 01 (projet Calibration). The first author also acknowledges the support of the GARGANTUA project funded by the Mastodon program of CNRS. The first author was partly supported by Institut des Hautes Études Scientifiques (IHES), Le Bois-Marie, 35, route de Chartres, 91440 Bures-Sur-Yvette, France) during the last days of writing of this paper.

Appendix A. Proofs

Before proving the main results stated in the paper, let us recall two simple results that we use repeatedly in the paper. First, if $(b_\lambda)_{\lambda \in \Lambda_m}$ is a family of real numbers such that $\sum_{\lambda \in \Lambda_m} b_\lambda^2 < \infty$, then

$$\sup_{\sum_{\lambda \in \Lambda_m} a_\lambda^2 \leq 1} \left(\sum_{\lambda \in \Lambda_m} a_\lambda b_\lambda \right)^2 = \sum_{\lambda \in \Lambda_m} b_\lambda^2. \quad (30)$$

The left-hand side is smaller than the right-hand side by Cauchy-Schwarz inequality, and considering $a_\lambda = b_\lambda / (\sum_{\lambda \in \Lambda_m} b_\lambda^2)^{1/2}$ shows that the converse inequality holds true. Second, for any probability distribution Q on \mathcal{X} ,

$$\sum_{\lambda \in \Lambda_m} (Q\psi_\lambda) \psi_\lambda \in \operatorname{argmin}_{t \in S_m} \{Q\gamma(t)\}, \quad (31)$$

a result which provides in particular a formula for \widehat{s}_m and for s_m , by taking $Q = P_n$ and $Q = P$, respectively.

A.1 Proof of Lemma 1

Let us first recall here the proof of Eq. (7)—coming from Arlot (2008)—for the sake of completeness. By **(Reg)**,

$$P_n - P_n^{(\beta_K)} = \frac{1}{V} (P_n^{(\beta_K)} - P_n^{(\beta_K)}) \quad \text{and} \quad P_n^{(\beta_K)} - P_n = \frac{V-1}{V} (P_n^{(\beta_K)} - P_n^{(\beta_K)}),$$

so that

$$\begin{aligned} G_{1,\beta}(m) &:= P_n \gamma(\widehat{s}_m) + \operatorname{pen}_{V,F}(m, \beta, V-1) \\ &= P_n \gamma(\widehat{s}_m) + \frac{V-1}{V^2} \sum_{k=1}^V [(P_n^{(\beta_K)} - P_n^{(\beta_K)}) \gamma(\widehat{s}_m^{(\beta_K)})] \\ &= P_n \gamma(\widehat{s}_m) + \frac{1}{V} \sum_{k=1}^V [(P_n^{(\beta_K)} - P_n) \gamma(\widehat{s}_m^{(\beta_K)})] \end{aligned}$$

Eq. (8) and (9) follow simultaneously from Eq. (35) below. Let \mathcal{E} be a set of subsets of $[n]$ such that

$$\forall A \in \mathcal{E}, \quad |A| = p \quad \text{and} \quad \frac{1}{|\mathcal{E}|} \sum_{A \in \mathcal{E}} P_n^{(A)} = P_n. \quad (32)$$

Let us consider the associated penalty

$$\operatorname{pen}_{\mathcal{E}}(m, C) = \frac{C}{|\mathcal{E}|} \sum_{A \in \mathcal{E}} (P_n - P_n^{(A)}) \gamma(\widehat{s}_m^{(A)}) = \frac{2C}{|\mathcal{E}|} \sum_{A \in \mathcal{E}} (P_n^{(A)} - P_n) (\widehat{s}_m^{(A)})$$

and the associated cross-validation criterion

$$\operatorname{crit}_{\mathcal{E}}(m) = \frac{1}{|\mathcal{E}|} \sum_{A \in \mathcal{E}} P_n^{(A)} \gamma(\widehat{s}_m^{(A)}).$$

When $\mathcal{E} = \mathcal{B}$, we get the V -fold penalty $\operatorname{pen}_V = \operatorname{pen}_{\mathcal{E}}$ and the V -fold cross-validation criterion $\operatorname{crit}_{V,F} = \operatorname{crit}_{\mathcal{E}}$, and Eq. (32) holds true with $p = n/V$ under assumption **(Reg)**. When $\mathcal{E} = \mathcal{E}_p := \{A \subset [n] \text{ s.t. } |A| = p\}$, Eq. (32) always holds true and we get the leave- p -out penalty $\operatorname{pen}_{LPO} = \operatorname{pen}_{\mathcal{E}}$ and the leave- p -out cross-validation criterion $\operatorname{crit}_{LPO} = \operatorname{crit}_{\mathcal{E}}$.

Let $(\psi_\lambda)_{\lambda \in \Lambda_m}$ be some orthonormal basis of S_m in $L^2(\mu)$. On the one hand, using Eq. (32), we get

$$\begin{aligned} \operatorname{pen}_{\mathcal{E}}(m, C) &= \frac{2C}{|\mathcal{E}|} \sum_{A \in \mathcal{E}} (P_n^{(A)} - P_n) (\widehat{s}_m^{(A)}) \\ &= \frac{2C}{|\mathcal{E}|} \sum_{A \in \mathcal{E}} \sum_{\lambda \in \Lambda_m} [(P_n^{(A)}(\psi_\lambda) - P_n(\psi_\lambda)) P_n^{(A)}(\psi_\lambda)] \\ &= \frac{2C}{|\mathcal{E}|} \sum_{\lambda \in \Lambda_m} \left[\sum_{A \in \mathcal{E}} (P_n^{(A)}(\psi_\lambda))^2 - P_n(\psi_\lambda) \sum_{A \in \mathcal{E}} P_n^{(A)}(\psi_\lambda) \right] \\ &= \frac{2C}{|\mathcal{E}|} \sum_{\lambda \in \Lambda_m, A \in \mathcal{E}} \left[(P_n^{(A)}(\psi_\lambda))^2 - (P_n(\psi_\lambda))^2 \right]. \end{aligned} \quad (33)$$

On the other hand, using that $P_n^{(A)} = \frac{n}{p} P_n - \frac{n-p}{p} P_n^{(A^c)}$ by Eq. (32),

$$\begin{aligned} \operatorname{crit}_{\mathcal{E}}(m) - P_n \gamma(\widehat{s}_m) &= \frac{1}{|\mathcal{E}|} \sum_{A \in \mathcal{E}} [P_n^{(A)} \gamma(\widehat{s}_m^{(A)}) - P_n \gamma(\widehat{s}_m)] \\ &= \frac{1}{|\mathcal{E}|} \sum_{A \in \mathcal{E}} [\|\widehat{s}_m^{(A^c)}\|^2 - 2P_n^{(A)}(\widehat{s}_m^{(A^c)}) - \|\widehat{s}_m\|^2 + 2P_n(\widehat{s}_m)] \\ &= \frac{1}{|\mathcal{E}|} \sum_{A \in \mathcal{E}} \sum_{\lambda \in \Lambda_m} \left[(P_n^{(A^c)}(\psi_\lambda))^2 - 2P_n^{(A)}(\psi_\lambda) P_n^{(A^c)}(\psi_\lambda) + (P_n(\psi_\lambda))^2 \right] \\ &= \frac{1}{|\mathcal{E}|} \sum_{\lambda \in \Lambda_m, A \in \mathcal{E}} \left[\left(\frac{2n}{p} - 1 \right) (P_n^{(A^c)}(\psi_\lambda))^2 - \frac{2n}{p} P_n(\psi_\lambda) P_n^{(A^c)}(\psi_\lambda) + (P_n(\psi_\lambda))^2 \right] \end{aligned}$$

$$= \binom{2n-1}{p} \frac{1}{|\mathcal{E}|} \sum_{\lambda \in \Lambda_m} \sum_{A \in \mathcal{E}} \left[\left(P_n^{(A^c)}(\psi_\lambda) \right)^2 - \left(P_n(\psi_\lambda) \right)^2 \right], \quad (34)$$

where we used again Eq. (32). Comparing Eq. (33) and (34) gives

$$\text{crit}_\mathcal{E}(m) = P_n\gamma(\widehat{s}_m) + \text{pen}_\mathcal{E}\left(m, \frac{n}{p} - \frac{1}{2}\right) \quad (35)$$

which implies Eq. (8) and (9). Eq. (10) follows by Lemma A.11 of Lerasle (2012). ■

We now prove the statements made in Remarks 2-3 below Lemma 1.

Proof of Remark 2 We first note that Eq. (10) can also be deduced from Celisse (2014, Proposition 2.1), which proves

$$\text{crit}_{\text{LPO}}(m, p) = \frac{1}{n(n-p)} \sum_{\lambda \in \Lambda_m} \left(\sum_{i=1}^n \psi_\lambda(\xi_i) \right)^2 - \frac{n-p+1}{n-1} \sum_{1 \leq i \neq j \leq n} \psi_\lambda(\xi_i) \psi_\lambda(\xi_j).$$

Elementary algebraic computations then show that

$$\begin{aligned} \text{crit}_{\text{LPO}}(m, p) - P_n\gamma(\widehat{s}_m) &= \frac{2n-p}{n^2(n-p)} \sum_{\lambda \in \Lambda_m} \left(\sum_{i=1}^n \psi_\lambda(\xi_i) \right)^2 - \frac{1}{n-1} \sum_{1 \leq i \neq j \leq n} \psi_\lambda(\xi_i) \psi_\lambda(\xi_j) \end{aligned} \quad (36)$$

hence for any $p, p' \in \llbracket n \rrbracket$,

$$\frac{n/p-1}{n/p-1/2} (\text{crit}_{\text{LPO}}(m, p) - P_n\gamma(\widehat{s}_m)) = \frac{n/p'-1}{n/p'-1/2} (\text{crit}_{\text{LPO}}(m, p') - P_n\gamma(\widehat{s}_m)).$$

In particular, when $p' = 1$, from Eq. (9), since $\text{pen}_{\text{LPO}}(m, 1, C) = \text{pen}_{\text{LOO}}(m, C)$,

$$\begin{aligned} \text{pen}_{\text{LPO}}\left(m, p, \frac{n}{p} - \frac{1}{2}\right) &= \frac{n/p-1/2}{n/p-1} \frac{n-1}{n-1/2} \text{pen}_{\text{LPO}}\left(m, 1, n - \frac{1}{2}\right) \\ &= \text{pen}_{\text{LOO}}\left(m, (n-1) \frac{n/p-1/2}{n/p-1}\right). \end{aligned}$$

■

Proof of Remark 3 Note first that the CV estimator of Massart (2007, Sec. 7.2.1, p. 204-205) is defined as the minimizer of

$$\begin{aligned} \|\widehat{s}_m\|^2 - \frac{2}{n(n-1)} \sum_{1 \leq i \neq j \leq n} \psi_\lambda(\xi_i) \psi_\lambda(\xi_j) \\ = P_n\gamma(\widehat{s}_m) + \frac{2}{n^2} \sum_{\lambda \in \Lambda_m} \left(\sum_{i=1}^n \psi_\lambda(\xi_i) \right)^2 - \frac{1}{n-1} \sum_{1 \leq i \neq j \leq n} \psi_\lambda(\xi_i) \psi_\lambda(\xi_j) \end{aligned} \quad (37)$$

On the other hand, from Eq. (36) and (9) with $p = 1$, we have

$$\text{pen}_{\text{LOO}}(m, n-1) = \frac{2}{n^2} \sum_{\lambda \in \Lambda_m} \left(\sum_{i=1}^n \psi_\lambda(\xi_i) \right)^2 - \frac{1}{n-1} \sum_{1 \leq i \neq j \leq n} \psi_\lambda(\xi_i) \psi_\lambda(\xi_j).$$

Hence, from Eq. (37), the CV estimator is the minimizer of $\text{crit}_{\text{cor, VFCV}}(m, \mathcal{B}_{\text{LOO}})$. Massart (2007, Theorem 7.6) studies the minimizers of the criterion

$$P_n\gamma(\widehat{s}_m) + \frac{C}{n^2} \sum_{i=1}^n \sum_{\lambda \in \Lambda_m} \psi_\lambda(\xi_i)^2, \quad (38)$$

where $C = (1 + \epsilon)^6$ for any $\epsilon > 0$. Let $\alpha = C/n$, so that $\alpha = (C - \alpha)/(n - 1)$. Then, the criterion (38) is equal to

$$\begin{aligned} (1-\alpha)P_n\gamma(\widehat{s}_m) + \frac{C-\alpha}{n^2} \sum_{\lambda \in \Lambda_m} \sum_{i=1}^n \psi_\lambda(\xi_i)^2 - \frac{\alpha}{n^2} \sum_{\lambda \in \Lambda_m} \sum_{1 \leq i \neq j \leq n} \psi_\lambda(\xi_i) \psi_\lambda(\xi_j) \\ = (1-\alpha)P_n\gamma(\widehat{s}_m) + \frac{C-\alpha}{n^2} \sum_{\lambda \in \Lambda_m} \left(\sum_{i=1}^n \psi_\lambda(\xi_i) \right)^2 - \frac{1}{n-1} \sum_{\lambda \in \Lambda_m} \sum_{1 \leq i \neq j \leq n} \psi_\lambda(\xi_i) \psi_\lambda(\xi_j) \\ = (1-\alpha) \left[P_n\gamma(\widehat{s}_m) + \frac{C-\alpha}{2(1-\alpha)} \text{pen}_{\text{LOO}}(m, n-1) \right] \\ = (1-\alpha) \left[P_n\gamma(\widehat{s}_m) + \text{pen}_{\text{LOO}}\left(m, \frac{C(n-1)^2}{2(n-C)}\right) \right]. \end{aligned}$$

■

A.2 Proof of Proposition 4

Note that the two formulas given for Ψ_m in the statement of Proposition 4 coincide by Eq. (30). The proof is decomposed into 3 lemmas.

Lemma 13 Let $\xi_{[n]}$ denote i.i.d. random variables taking value in a Polish space \mathcal{X} , $\mathcal{B}_{[V]}$ some partition of $\llbracket n \rrbracket$ satisfying **(Reg)**, S_m some separable linear subspace of $L^2(\mu)$ with orthonormal basis $(\psi_\lambda)_{\lambda \in \Lambda_m}$ and

$$U(m) := \frac{1}{n^2} \sum_{1 \leq k \neq k' \leq V} \sum_{i \in \mathcal{B}_k, j \in \mathcal{B}_{k'}} \sum_{\lambda \in \Lambda_m} (\psi_\lambda(\xi_i) - P\psi_\lambda)(\psi_\lambda(\xi_j) - P\psi_\lambda). \quad (39)$$

Then, the V -fold penalty is equal to

$$\text{pen}_{\text{VF}}(m, \mathcal{B}, C) = \frac{2C}{V-1} \|s_m - \widehat{s}_m\|^2 - \frac{2VC}{(V-1)^2} U(m) \quad (40)$$

$$\text{and } \mathbb{E} \left[\text{pen}_{\text{VF}}\left(m, \mathcal{B}, \frac{V-1}{2}\right) \right] = \mathbb{E}[\|s_m - \widehat{s}_m\|^2] = \frac{D_m}{2n}. \quad (41)$$

Proof Let $W_i = \frac{V}{V-1} \mathbb{1}_{i \notin \mathcal{B}_j}$, and use the formulation (5) of the V -fold penalty as a resampling penalty. Then,

$$\begin{aligned} \text{pen}_{V,F}(m, \mathcal{B}, C) &= C \mathbb{E}_W \left[(P_n - P_n^W) \left(\gamma(\hat{s}_m^W) \right) \right] \\ &= 2C \mathbb{E}_W \left[(P_n^W - P_n) (\hat{s}_m^W) \right] \\ &= 2C \mathbb{E}_W \left[(P_n^W - P_n) (\hat{s}_m^W - \hat{s}_m) \right] \quad \text{by (Reg)} \\ &= 2C \sum_{\lambda \in \Delta_m} \mathbb{E}_W \left[\left((P_n^W - P_n) (\psi_\lambda) \right)^2 \right] \\ &= 2C \sum_{\lambda \in \Delta_m} \mathbb{E}_W \left[\left((P_n^W - P_n) (\psi_\lambda - P\psi_\lambda) \right)^2 \right] \\ &= 2C \sum_{\lambda \in \Delta_m} \sum_{1 \leq i, j \leq n} e_{i,j}^{(V,F)} (\psi_\lambda(\xi_i) - P\psi_\lambda)(\psi_\lambda(\xi_j) - P\psi_\lambda) \end{aligned} \quad (42)$$

where $e_{i,j}^{(V,F)} := \mathbb{E}[(W_i - 1)(W_j - 1)]$. Since $\mathbb{E}[W_i] = 1$ by (Reg) and

$$W_i W_j = \left(\frac{V}{V-1} \right)^2 \mathbb{1}_{j \notin \{i_0, j_1\}} \quad \text{if } i \in \mathcal{B}_{i_0} \text{ and } j \in \mathcal{B}_{j_1},$$

we get that $e_{i,j}^{(V,F)} = (V-1)^{-1}$ if i and j belong to the same block and $e_{i,j}^{(V,F)} = -(V-1)^{-2}$ otherwise. So,

$$\begin{aligned} \text{pen}_{V,F}(m, \mathcal{B}, C) &= \frac{2C}{n^2(V-1)} \sum_{\lambda \in \Delta_m} \sum_{k=1}^V \sum_{(i,j) \in \mathcal{B}_k} (\psi_\lambda(\xi_i) - P\psi_\lambda)(\psi_\lambda(\xi_j) - P\psi_\lambda) - \frac{2C}{(V-1)^2} U(m) \\ &= \frac{2C}{V-1} \sum_{\lambda \in \Delta_m} \left((P_n - P)\psi_\lambda \right)^2 - \frac{2CV}{(V-1)^2} U(m) \end{aligned}$$

and Eq. (40) follows by Eq. (3). Eq. (41) directly follows from Eq. (40). \blacksquare

Lemma 14 Let $\xi_{[n]}$ be i.i.d. random variables taking values in a Polish space \mathcal{X} with common density $s \in L^\infty(\mu)$, S_m a separable linear subspace of $L^2(\mu)$ and denote by $(\psi_\lambda)_{\lambda \in \Delta_m}$ an orthonormal basis of S_m . Let $\mathcal{B}_m = \{t \in S_m \text{ s.t. } \|t\| \leq 1\}$, $\mathcal{D}_m = \sum_{\lambda \in \Delta_m} P(\psi_\lambda^2) - \|s_m\|^2$ and assume that $b_m = \sup_{t \in \mathcal{B}_m} \|t\|_\infty < \infty$. An absolute constant κ exists such that, for any $x > 0$, with probability larger than $1 - 2e^{-x}$, we have for every $\epsilon > 0$,

$$\left| \|s_m - \hat{s}_m\|^2 - \frac{\mathcal{D}_m}{n} \right| \leq \epsilon \frac{\mathcal{D}_m}{n} + \kappa \left(\frac{\|s\|_\infty x}{(\epsilon \wedge 1)n} + \frac{b_m^2 x^2}{(\epsilon \wedge 1)^3 n^2} \right).$$

Proof By Eq. (3), $\|s_m - \hat{s}_m\|^2 = \sup_{t \in \mathcal{B}_m} [(P_n - P)(t)]^2$ has expectation \mathcal{D}_m/n . In addition, for any $t \in \mathcal{B}_m$,

$$\text{Var}(t(\xi_1)) \leq \int_{\mathbb{R}} t^2 s \, d\mu \leq \|s\|_\infty \|t\|^2 \leq \|s\|_\infty, \quad (43)$$

which gives the conclusion thanks to a result by Lerasle (2011, Theorem 4.1 of the supplementary material), which is recalled in the Online Appendix (Proposition 29 in Section F). \blacksquare

Lemma 15 Assume that $\xi_{[n]}$ is a sequence of i.i.d. real-valued random variables with common density $s \in L^\infty(\mu)$ and $\mathcal{B}_{[n]}$ is some partition of $[n]$ satisfying (Reg). Let S_m denote a separable subspace of $L^2(\mu)$ with orthonormal basis $(\psi_\lambda)_{\lambda \in \Delta_m}$ such that

$$b_m := \sup_{t \in S_m, \|t\| \leq 1} \|t\|_\infty < +\infty.$$

Let $U(m)$ be the U -statistics defined by Eq. (39). Using the notations of Lemma 14, an absolute constant κ exists such that, with probability larger than $1 - 6e^{-x}$,

$$|U(m)| \leq \frac{3\sqrt{(V-1)}\|s\|_\infty \mathcal{D}_m x}{\sqrt{V}n} + \kappa \left(\frac{\|s\|_\infty x}{n} + \frac{(b_m^2 + \|s\|^2)x^2}{n^2} \right).$$

Hence, an absolute constant κ' exists such that, for any $x > 0$, with probability larger than $1 - 6e^{-x}$, for any $\theta \in (0, 1]$,

$$|U(m)| \leq \theta \frac{\mathcal{D}_m}{n} + \kappa' \left(\frac{\|s\|_\infty x}{\theta n} + \frac{(b_m^2 + \|s\|^2)x^2}{n^2} \right).$$

Proof For any $x, y \in \mathbb{R}$ and $i, j \in [n]$, let us define

$$U_m(x, y) = \sum_{\lambda \in \Delta_m} (\psi_\lambda(x) - P\psi_\lambda)(\psi_\lambda(y) - P\psi_\lambda)$$

and $g_{i,j}(x, y) = U_m(x, y) \mathbb{1}_{\{k, k' \in [V] \text{ s.t. } k \neq k', i \in \mathcal{B}_k, j \in \mathcal{B}_{k'}\}}$

so that $U(m) = \frac{2}{n^2} \sum_{i=2}^n \sum_{j=1}^{i-1} g_{i,j}(\xi_i, \xi_j) = \frac{2}{n^2} \sum_{k=2}^V \sum_{k'=1}^{k-1} \sum_{i \in \mathcal{B}_k, j \in \mathcal{B}_{k'}} U_m(\xi_i, \xi_j)$.

From Houdré and Reynaud-Bouret (2003, Theorem 3.4), an absolute constant κ exists such that, for any $x > 0$ and $\epsilon \in (0, 1]$,

$$\mathbb{P} \left(|U(m)| \geq \frac{1}{n^2} \left[(4 + \epsilon) \bar{A} \sqrt{x} + \kappa \left(\frac{\bar{B}x}{\epsilon} + \frac{C \bar{x}^{3/2}}{\epsilon^3} + \frac{\bar{D}x^2}{\epsilon^3} \right) \right] \right) \leq 6e^{-x}. \quad (44)$$

$$\bar{A}^2 = \sum_{i=2}^n \sum_{j=1}^{i-1} \mathbb{E}[g_{i,j}(\xi_i, \xi_j)^2],$$

$$\bar{B} = \sup \left\{ \mathbb{E} \left[\sum_{i=2}^n \sum_{j=1}^{i-1} a_i(\xi_i) b_j(\xi_j) g_{i,j}(\xi_i, \xi_j) \right] \right.$$

$$\left. \text{such that } \mathbb{E} \left[\sum_{i=1}^n a_i^2(\xi_i) \right] \leq 1 \text{ and } \mathbb{E} \left[\sum_{i=1}^n b_i^2(\xi_i) \right] \leq 1 \right\},$$

$$\overline{\mathcal{C}}^2 = \sup_{x \in \mathbb{R}} \left\{ \sum_{i=2}^n \mathbb{E}[g_{i,1}(\xi_i, x)^2] \right\} \quad \text{and} \quad \overline{D} = \sup_{x, y} |g_{i,j}(x, y)|.$$

It remains to upper bound these different terms for proving the first inequality, and the second inequality follows. First,

$$\begin{aligned} \mathbb{E}[U_m(\xi_1, \xi_2)^2] &= \sum_{\lambda \in \Lambda_m, \lambda' \in \Lambda_m} \mathbb{E} \left[(\psi_\lambda(\xi_1) - P\psi_\lambda)(\psi_{\lambda'}(\xi_1) - P\psi_{\lambda'}) \right]^2 \\ &= \sum_{\lambda \in \Lambda_m} \left(\sup_{\sum_{\lambda' \in \Lambda_m} a_{\lambda'}^2 \leq 1} \mathbb{E} \left[(\psi_\lambda(\xi_1) - P\psi_\lambda) \sum_{\lambda' \in \Lambda_m} a_{\lambda'} (\psi_{\lambda'}(\xi_1) - P\psi_{\lambda'}) \right) \right]^2 \\ &= \sum_{\lambda \in \Lambda_m} \left(\sup_{t \in \mathbb{B}_m} \mathbb{E} \left[(\psi_\lambda(\xi_1) - P\psi_\lambda)(t(\xi_1) - P(t)) \right] \right)^2 \\ &\leq \mathcal{D}_m \sup_{t \in \mathbb{B}_m} \mathbb{E} \left[(t(\xi_1) - P(t))^2 \right] \\ &\leq \|s\|_\infty \mathcal{D}_m \quad \text{by Eq. (43)} \end{aligned} \quad (45)$$

so that

$$\overline{A}^2 = \sum_{k=2}^V \sum_{k'=1}^{k-1} \sum_{t \in \mathbb{B}_{k,j} \in \mathcal{B}_{k'}} \mathbb{E} \left[U_m(\xi_i, \xi_j)^2 \right] \leq \frac{\eta^2(V-1)}{2V} \times \|s\|_\infty \mathcal{D}_m.$$

Second, let $a_1, \dots, a_n, b_1, \dots, b_n$ be functions in $L^2(\mu)$ such that

$$\mathbb{E} \left[\sum_{i=1}^n a_i^2(\xi_i) \right] \leq 1 \quad \text{and} \quad \mathbb{E} \left[\sum_{i=1}^n b_i^2(\xi_i) \right] \leq 1.$$

Using successively the independence of the ξ_i and that $\alpha\beta \leq (\alpha^2 + \beta^2)/2$ for every $\alpha, \beta \in \mathbb{R}$, for every $i \neq j$,

$$\begin{aligned} & \left| \mathbb{E} [a_i(\xi_i) b_j(\xi_j) U_m(\xi_i, \xi_j)] \right| \\ &= \left| \sum_{\lambda \in \Lambda_m} \mathbb{E} [a_i(\xi_i) (\psi_\lambda(\xi_i) - P\psi_\lambda)] \mathbb{E} [b_j(\xi_j) (\psi_\lambda(\xi_j) - P\psi_\lambda)] \right| \\ &\leq \frac{1}{2} \sum_{\lambda \in \Lambda_m} \left(\mathbb{E} [a_i(\xi_i) (\psi_\lambda(\xi_i) - P\psi_\lambda)]^2 + \mathbb{E} [b_j(\xi_j) (\psi_\lambda(\xi_j) - P\psi_\lambda)]^2 \right). \end{aligned} \quad (46)$$

Now, we have, for every $i \in \llbracket n \rrbracket$, using Eq. (30), Cauchy-Schwarz inequality and the fact that for every $t \in L^2(\mu)$, $\text{Var}(t(\xi_1)) \leq \|s\|_\infty \|t\|^2$,

$$\sum_{\lambda \in \Lambda_m} \mathbb{E} [a_i(\xi_i) (\psi_\lambda(\xi_i) - P\psi_\lambda)]^2 = \sup_{\sum_{\lambda \in \Lambda_m} t_\lambda^2 \leq 1} \left(\mathbb{E} \left[a_i(\xi_i) \sum_{\lambda \in \Lambda_m} t_\lambda \psi_\lambda(\xi_i) - P(t_\lambda \psi_\lambda) \right] \right)^2$$

$$\begin{aligned} &= \sup_{t \in \mathbb{B}_m} \left(\mathbb{E} [a_i(\xi_i) (t(\xi_i) - P(t))] \right)^2 \\ &\leq \mathbb{E} [a_i(\xi_i)^2] \sup_{t \in \mathbb{B}_m} \text{Var}(t(\xi_1)) \leq \mathbb{E} [a_i(\xi_i)^2] \|s\|_\infty. \end{aligned}$$

Plugging this bound in (46) yields

$$\left| \mathbb{E} [a_i(\xi_i) b_j(\xi_j) U_m(\xi_i, \xi_j)] \right| \leq \frac{\|s\|_\infty}{2} \left(\mathbb{E} [a_i(\xi_i)^2] + \mathbb{E} [b_j(\xi_j)^2] \right) \quad (47)$$

hence

$$\overline{B} \leq n \|s\|_\infty.$$

Third, for every $x, y \in \mathbb{R}$, let $g_x(y) = \sum_{\lambda \in \Lambda_m} (\psi_\lambda(x) - P\psi_\lambda) \psi_\lambda(y)$ so that

$$\begin{aligned} \|g_x\|^2 &= \sum_{\lambda \in \Lambda_m} (\psi_\lambda(x) - P\psi_\lambda)^2 \leq 2 \sum_{\lambda \in \Lambda_m} (\psi_\lambda(x))^2 + 2 \sum_{\lambda \in \Lambda_m} (P\psi_\lambda)^2 \\ &= 2\Psi_m(x)^2 + 2\|s_m\|^2 \leq 2(b_m^2 + \|s_m\|^2). \end{aligned}$$

Then,

$$\mathbb{E}[U_m(\xi_i, x)^2] = \text{Var}(g_x(\xi_1)) \leq \|g_x\|^2 \|s\|_\infty \leq 2(b_m^2 + \|s_m\|^2) \|s\|_\infty \quad (48)$$

and, using **(Reg)**, we get that

$$\overline{\mathcal{C}}^2 \leq \frac{2n(V-1)}{V} (b_m^2 + \|s_m\|^2) \|s\|_\infty.$$

Fourth, from Cauchy-Schwarz inequality, for every $x, y \in \mathcal{X}$,

$$U_m(x, y) \leq \sup_{x \in \mathbb{R}, \lambda \in \Lambda_m} (\psi_\lambda(x) - P\psi_\lambda)^2 \leq 2(b_m^2 + \|s_m\|^2). \quad (49)$$

Hence,

$$\overline{D} \leq 2(b_m^2 + \|s_m\|^2)$$

and we get the desired result. \blacksquare

Let us conclude the proof of Proposition 4. From Lemmas 13 and 15, an absolute constant κ exists such that, with probability larger than $1 - 6e^{-x}$, for every $\epsilon \in (0, 1]$,

$$\begin{aligned} & \left| \text{pen}_{\text{VF}}(m, V, V-1) - 2\|s_m - \widehat{s}_m\|^2 \right| \\ &= \frac{2V}{V-1} |U(m)| \leq \epsilon \frac{\mathcal{D}_m}{n} + \kappa \left(\frac{\|s\|_\infty x}{\epsilon n} + \frac{(b_m^2 + \|s\|^2) x^2}{n^2} \right). \end{aligned} \quad (50)$$

Using in addition Lemma 14, we get that an absolute constant κ' exists such that with probability larger than $1 - 8e^{-x}$, for every $\epsilon \in (0, 1]$, Eq. (50) holds true and

$$\left| \text{pen}_{\text{VF}}(m, V, V-1) - \frac{2\mathcal{D}_m}{n} \right| \leq \epsilon \frac{\mathcal{D}_m}{n} + \kappa \left(\frac{\|s\|_\infty x}{\epsilon n} + \frac{(b_m^2 \epsilon^{-3} + \|s\|^2) x^2}{n^2} \right),$$

which implies Eq. (15) and (16). \blacksquare

A.3 Proof of Theorem 5

By construction, the penalized estimator satisfies, for any $m \in \mathcal{M}_n$,

$$\begin{aligned} & \|\widehat{s}_{\widehat{m}} - s\|^2 - \left(\text{pen}_{\text{rd}}(\widehat{m}) - \text{pen}_{\text{VF}}(\widehat{m}, V, C(V-1)) \right) \\ & \leq \|\widehat{s}_{\widehat{m}} - s\|^2 + \left(\text{pen}_{\text{VF}}(m, V, C(V-1)) - \text{pen}_{\text{rd}}(m) \right). \end{aligned}$$

Now, by Eq. (2) and (3), $\text{pen}_{\text{rd}}(m) = 2\|\widehat{s}_m - s_{m'}\|^2 + 2(P_n - P)(s_m)$, hence

$$\begin{aligned} \|\widehat{s}_{\widehat{m}} - s\|^2 & \leq \|\widehat{s}_{\widehat{m}} - s\|^2 + \left[\text{pen}_{\text{VF}}(m, V, C(V-1)) - 2\|s_m - \widehat{s}_{\widehat{m}}\|^2 \right] \\ & \quad - \left[\text{pen}_{\text{VF}}(\widehat{m}, V, C(V-1)) - 2\|s_{\widehat{m}} - \widehat{s}_{\widehat{m}}\|^2 \right] + 2(P_n - P)(s_m - s_{\widehat{m}}) \\ & = \|\widehat{s}_{\widehat{m}} - s\|^2 + \left[\text{pen}_{\text{VF}}(m, V, C(V-1)) - 2C\|s_m - \widehat{s}_{\widehat{m}}\|^2 \right] \\ & \quad - \left[\text{pen}_{\text{VF}}(\widehat{m}, V, C(V-1)) - 2C\|s_{\widehat{m}} - \widehat{s}_{\widehat{m}}\|^2 \right] + 2(P_n - P)(s_m - s_{\widehat{m}}) \\ & \quad + 2(C-1) \left(\|\widehat{s}_{\widehat{m}} - s_m\|^2 - \|\widehat{s}_{\widehat{m}} - s_{\widehat{m}}\|^2 \right). \end{aligned} \quad (51)$$

Let $x > 0$ and $x_n = \log(|\mathcal{M}_n|) + x$. A union bound in Proposition 4 gives

$$\begin{aligned} \mathbb{P} \left(\exists m \in \mathcal{M}_n, \epsilon \in (0, 1] \text{ s.t. } \left| \text{pen}_{\text{VF}}(m, V, V-1) - 2\|s_m - \widehat{s}_m\|^2 \right| \right. \\ \left. > \epsilon \frac{D_m}{n} + \kappa \rho_1(m, \epsilon, s, x_n, n) \right) \leq 8 \sum_{m \in \mathcal{M}_n} e^{-x_n} = 8e^{-x} \sum_{m \in \mathcal{M}_n} \frac{1}{|\mathcal{M}_n|} = 8e^{-x} \end{aligned} \quad (52)$$

and a union bound in Lemma 14 gives

$$\mathbb{P} \left(\exists m \in \mathcal{M}_n, \epsilon \in (0, 1] \text{ s.t. } \left| \|\widehat{s}_m - s_m\|^2 - \frac{D_m}{n} \right| > \epsilon \frac{D_m}{n} + \kappa \rho_1(m, \epsilon, s, x_n, n) \right) \\ \leq 2 \sum_{m \in \mathcal{M}_n} e^{-x_n} = 2e^{-x}. \quad (53)$$

It remains to bound $2(P_n - P)(s_m - s_{m'})$ uniformly over m and m' in \mathcal{M}_n . In order to apply Bernstein's inequality, we first bound the variance and the sup norm of $s_m - s_{m'}$ for some $m, m' \in \mathcal{M}_n$. Since $s \in L^\infty(\mu)$,

$$\text{Var}((s_m - s_{m'})(\xi_1)) \leq \|s\|_\infty \|s_m - s_{m'}\|^2.$$

Under assumption (H2)

$$\|s_m - s_{m'}\|_\infty \leq \|s_m\|_\infty + \|s_{m'}\|_\infty \leq 2a.$$

Under assumption (H2), $s_m - s_{m'} \in \mathcal{S}_{m, m'}$ for some $m'' \in \{m, m'\}$, hence by (H1) we have

$$\|s_m - s_{m'}\|_\infty \leq b_{m''} \|s_m - s_{m'}\| \leq \sqrt{n} \|s_m - s_{m'}\|.$$

Therefore, by Bernstein's inequality, for any $x > 0$, for any m, m' , with probability larger than $1 - e^{-x}$, for any $\epsilon \in (0, 1]$,

$$\begin{aligned} (P_n - P)(s_m - s_{m'}) & \leq \sqrt{\frac{2x \text{Var}((s_m - s_{m'})(\xi_1))}{n}} + \frac{\|s_m - s_{m'}\|_\infty x}{3n} \\ & \leq \epsilon \|s_m - s_{m'}\|^2 + \frac{\kappa(Ax + x^2)}{\epsilon n}. \end{aligned}$$

for some absolute constant κ , where the last inequality is obtained by considering separately the cases (H2) and (H2'), and by using that for every $\alpha, \beta, \epsilon > 0$, $\alpha\beta \leq \epsilon\alpha^2 + (\beta^2)/4\epsilon$. A union bound gives that for any $x > 0$, with probability at least $1 - |\mathcal{M}_n|^2 e^{-x}$, for every $m, m' \in \mathcal{M}_n$ and every $\epsilon \in (0, 1]$,

$$(P_n - P)(s_m - s_{m'}) \leq \epsilon \|s_m - s_{m'}\|^2 + \frac{\kappa(Ax + x^2)}{\epsilon n} \quad (54)$$

for some absolute constant κ . Plugging Eq. (52), (53) and (54) into Eq. (51) and using that $C \in (1/2, 2]$ yields that, with probability $1 - (|\mathcal{M}_n|^2 + 10)e^{-x}$, for any $\epsilon \in (0, 1/2]$,

$$\begin{aligned} (1-4\epsilon)\|\widehat{s}_{\widehat{m}} - s\|^2 & \leq (1+4\epsilon)\|\widehat{s}_m - s\|^2 + (\delta_+ + 4\epsilon) \frac{D_m}{n} + (\delta_- + 3\epsilon) \frac{D_{\widehat{m}}}{n} \\ & \quad + \kappa \left(\rho_1(m, \epsilon, s, x, n) + \rho_1(\widehat{m}, \epsilon, s, x, n) + \frac{Ax + x^2}{\epsilon n} \right) \\ & \leq (1 + \delta_+ + 16\epsilon) \|\widehat{s}_m - s\|^2 + (\delta_- + 8\epsilon) \|\widehat{s}_{\widehat{m}} - s_{\widehat{m}}\|^2 \\ & \quad + \kappa' \left(\rho_1(m, \epsilon, s, x, n) + \rho_1(\widehat{m}, \epsilon, s, x, n) + \frac{Ax + x^2}{\epsilon n} \right) \end{aligned}$$

for some absolute constants $\kappa, \kappa' > 0$. Since $b_m \leq \sqrt{n}$ for all $m \in \mathcal{M}_n$, we get

$$2 \sup_{m \in \mathcal{M}_n} \rho_1(m, \epsilon, s, x, n) + \frac{Ax + x^2}{\epsilon n} \leq \frac{(2\|s\|_\infty + A)x}{\epsilon n} + \left(3 + \frac{2\|s\|^2}{n} \right) \frac{x^2}{\epsilon^3 n}$$

for every $\epsilon \in (0, 1]$. Hence, with probability larger than $1 - (|\mathcal{M}_n|^2 + 10)e^{-x}$, for any $\epsilon \in (0, 1]$,

$$\frac{1 - \delta_- - \epsilon}{1 + \delta_+ + \epsilon} \|\widehat{s}_{\widehat{m}} - s\|^2 \leq \|\widehat{s}_m - s\|^2 + \kappa \left[\frac{(\|s\|_\infty + A)x}{\epsilon n} + \left(1 + \frac{\|s\|^2}{n} \right) \frac{x^2}{\epsilon^3 n} \right] \quad (55)$$

for some absolute constant $\kappa > 0$. To conclude, we remark that Eq. (17) clearly holds true when $|\mathcal{M}_n| = 1$, so we can assume that $|\mathcal{M}_n| \geq 2$. Therefore, for every $x > 0$, Eq. (55) holds true with probability at least

$$1 - \left(|\mathcal{M}_n|^2 + 10 \right) e^{-x} \geq 1 - |\mathcal{M}_n|^4 e^{-x} \geq 1 - e^{-x+4 \log |\mathcal{M}_n|}.$$

So, if we replace x by $4x_n \geq x + 4 \log |\mathcal{M}_n|$ in Eq. (55), we get that Eq. (17) holds true with probability at least $1 - e^{-x}$ for some absolute constant $\kappa > 0$, slightly larger than the one appearing in Eq. (55). \blacksquare

A.4 Proof of Theorem 6

For every $x, y \in \mathcal{X}$ and $m \in \{m_1, m_2\}$, let $K_m(x, y) := \sum_{\lambda \in \Lambda_m} \psi_\lambda(x) \psi_\lambda(y)$ and remark that

$$\begin{aligned} U_m(x, y) &= \sum_{\lambda \in \Lambda_m} (\psi_\lambda(x) - P\psi_\lambda)(\psi_\lambda(y) - P\psi_\lambda) \\ &= K_m(x, y) - s_m(x) - s_m(y) + \|s_m\|^2. \end{aligned} \quad (56)$$

For every $x \in \mathcal{X}$, $K_m(x, x) = \Psi_m(x, x)$ by Eq. (30), $U_m(x, x) = \Psi_m(x, x) - 2s_m(x) + \|s_m\|^2$ and, by independence, for every $m, m' \in \{m_1, m_2\}$

$$\begin{aligned} \text{Cov}(U_m(\xi_1, \xi_2), U_{m'}(\xi_1, \xi_2)) &= \sum_{\lambda \in \Lambda_m, \lambda' \in \Lambda_{m'}} \mathbb{E} \left[(\psi_\lambda(\xi_1) - P\psi_\lambda)(\psi_{\lambda'}(\xi_2) - P\psi_{\lambda'}) (\psi_\lambda(\xi_1) - P\psi_\lambda)(\psi_{\lambda'}(\xi_2) - P\psi_{\lambda'}) \right] \\ &= \sum_{\lambda \in \Lambda_m, \lambda' \in \Lambda_{m'}} \mathbb{E} \left[(\psi_\lambda(\xi_1) - P\psi_\lambda)(\psi_{\lambda'}(\xi_1) - P\psi_{\lambda'}) \right]^2 = \beta(m, m'), \end{aligned}$$

hence, $\text{Var}(U_{m_1}(\xi_1, \xi_2) - U_{m_2}(\xi_1, \xi_2)) = \mathbf{B}(m_1, m_2)$. For every $m \in \{m_1, m_2\}$, by Eq. (56),

$$\begin{aligned} P_n \gamma(\widehat{s}_m) &= - \sum_{\lambda \in \Lambda_m} (P_n \psi_\lambda)^2 = - \frac{1}{n^2} \sum_{1 \leq i, j \leq n} K_m(\xi_i, \xi_j) \\ &= - \frac{1}{n^2} \sum_{1 \leq i, j \leq n} U_m(\xi_i, \xi_j) - \frac{2}{n} \sum_{i=1}^n s_m(\xi_i) + \|s_m\|^2. \end{aligned} \quad (57)$$

Moreover, by Eq. (42) in the proof of Lemma 13,

$$\text{pen}_{\text{VF}}(m, \mathbf{B}, C(V-1)) = \frac{2C}{n^2} \sum_{1 \leq i, j \leq n} E_{i,j}^{(\text{VF})} U_m(\xi_i, \xi_j)$$

where $\forall I, J \in \{1, \dots, V\}$, $\forall i \in B_I, \forall j \in B_J$, $E_{i,j}^{(\text{VF})} = 1 - \frac{V \mathbb{1}_{I \neq J}}{V-1} = (V-1)e_{i,j}^{(\text{VF})}$.

It follows that

$$C_{C, \mathbf{B}}(m) = \sum_{1 \leq i, j \leq n} \frac{2C E_{i,j}^{(\text{VF})} - 1}{n^2} U_m(\xi_i, \xi_j) + \sum_{i=1}^n \frac{-2s_m(\xi_i)}{n} + \|s_m\|^2. \quad (58)$$

Hence, up to the deterministic term $\|s_m\|^2$, $C_{C, \mathbf{B}}(m)$ has the form of a function \mathcal{C}_m defined in Lemma 16 below with

$$\bar{\omega}_{i,j} = \frac{2C E_{i,j}^{(\text{VF})} - 1}{n^2}, \quad f_m = \frac{-2s_m}{n} \quad \text{and} \quad \bar{\sigma}_i = 1.$$

It remains to evaluate the quantities appearing in Lemma 16 for these weights and function. First,

$$\sum_{i=1}^n E_{i,i}^{(\text{VF})} = n \quad \text{and} \quad \sum_{i=1}^n (E_{i,i}^{(\text{VF})})^2 = n.$$

Second, by **(Reg)**,

$$\begin{aligned} \sum_{1 \leq i \neq j \leq n} (E_{i,j}^{(\text{VF})}) &= n \left(\frac{n}{V} - 1 \right) + \frac{-1}{(V-1)} \times \frac{n^2(V-1)}{V} = -n \\ \text{and} \quad \sum_{1 \leq i \neq j \leq n} (E_{i,j}^{(\text{VF})})^2 &= n \left[\left(\frac{n}{V} - 1 \right) + \frac{n}{V(V-1)} \right] = \frac{n^2}{V-1} - n. \end{aligned}$$

It follows that

$$\begin{aligned} \sum_{1 \leq i \leq n} \bar{\omega}_{i,i}^2 &= \frac{(2C-1)^2}{n^3}, \quad \sum_{i=1}^n \bar{\omega}_{i,i} \bar{\sigma}_i = \frac{2C-1}{n} \\ \text{and} \quad \sum_{1 \leq i \neq j \leq n} \bar{\omega}_{i,j} \bar{\omega}_{j,i} &= \sum_{1 \leq i \neq j \leq n} \bar{\omega}_{i,j}^2 = \frac{1}{n^2} \left(1 + \frac{4C^2}{V-1} - \frac{(2C-1)^2}{n} \right). \end{aligned}$$

Hence, from Lemma 16, for every $m, m' \in \{m_1, m_2\}$,

$$\begin{aligned} \text{Cov}(C_{C, \mathbf{B}}(m), C_{C, \mathbf{B}}(m')) &= \frac{2}{n^2} \left(1 + \frac{4C^2}{V-1} - \frac{(2C-1)^2}{n} \right) \beta(m, m') \\ &\quad + \frac{(2C-1)^2}{n^3} \text{Cov}(U_m(\xi, \xi), U_{m'}(\xi, \xi)) + \frac{4}{n} \text{Cov}(s_m(\xi), s_{m'}(\xi)) \\ &\quad - \frac{2(2C-1)}{n^2} \left[\text{Cov}(U_m(\xi, \xi), s_{m'}(\xi)) + \text{Cov}(U_{m'}(\xi, \xi), s_m(\xi)) \right] \\ &= \frac{2}{n^2} \left(1 + \frac{4C^2}{V-1} - \frac{(2C-1)^2}{n} \right) \beta(m, m') \\ &\quad + \frac{1}{n} \text{Cov} \left(\frac{2C-1}{n} U_m(\xi, \xi) - 2s_m(\xi), \frac{2C-1}{n} U_{m'}(\xi, \xi) - 2s_{m'}(\xi) \right). \end{aligned}$$

Therefore,

$$\begin{aligned} \text{Var}(C_{C, \mathbf{B}}(m_1)) &= \frac{2}{n^2} \left(1 + \frac{4C^2}{V-1} - \frac{(2C-1)^2}{n} \right) \beta(m_1, m_1) \\ &\quad + \frac{1}{n} \text{Var} \left(\frac{2C-1}{n} U_{m_1}(\xi, \xi) - 2s_{m_1}(\xi) \right) \\ \text{and} \quad \text{Var}(C_{C, \mathbf{B}}(m_1) - C_{C, \mathbf{B}}(m_2)) &= \frac{2}{n^2} \left(1 + \frac{4C^2}{V-1} - \frac{(2C-1)^2}{n} \right) \mathbf{B}(m_1, m_2) \\ &\quad + \frac{1}{n} \text{Var} \left(2(s_{m_1} - s_{m_2})(\xi) - \frac{2C-1}{n} (U_{m_1}(\xi, \xi) - U_{m_2}(\xi, \xi)) \right) \\ &= \frac{2}{n^2} \left(1 + \frac{4C^2}{V-1} - \frac{(2C-1)^2}{n} \right) \text{Var}(U_{m_1}(\xi, \xi) - U_{m_2}(\xi, \xi_2)) \\ &\quad + \frac{4}{n} \text{Var} \left(\left(1 + \frac{2C-1}{n} \right) (s_{m_1} - s_{m_2})(\xi) - \frac{2C-1}{2n} (\Psi_{m_1}(\xi) - \Psi_{m_2}(\xi)) \right), \end{aligned}$$

which concludes the proof. \blacksquare

Lemma 16 Let $C_m = \sum_{1 \leq i, j \leq n} \bar{w}_{ij} U_m(\xi_i, \xi_j) + \sum_{i=1}^n \bar{\sigma}_i f_m(\xi_i)$, where U_m is defined by Eq. (56) and $f_m \in L^2(\mu)$. For every m, m' , we have

$$\begin{aligned} \text{Cov}(C_m, C_{m'}) &= \left(\sum_{1 \leq i \neq j \leq n} \bar{w}_{ij}^2 + \bar{w}_{ij} \bar{w}_{ji} \right) \text{Cov}(U_m(\xi_1, \xi_2), U_{m'}(\xi_1, \xi_2)) \\ &+ \left(\sum_{i=1}^n \bar{w}_{ii}^2 \right) \text{Cov}(U_m(\xi_1, \xi_1), U_{m'}(\xi_1, \xi_1)) \\ &+ \left(\sum_{i=1}^n \bar{w}_{ii} \bar{\sigma}_i \right) \left[\text{Cov}(U_m(\xi_1, \xi_1), f_{m'}(\xi_1)) + \text{Cov}(U_{m'}(\xi_1, \xi_1), f_m(\xi_1)) \right] \\ &+ \left(\sum_{i=1}^n \bar{\sigma}_i^2 \right) \text{Cov}(f_m(\xi_1), f_{m'}(\xi_1)). \end{aligned}$$

Proof We develop the covariance to get

$$\begin{aligned} \text{Cov}(C_m, C_{m'}) &= \sum_{1 \leq i, j, k, \ell \leq n} \bar{w}_{ij} \bar{w}_{k\ell} \text{Cov}(U_m(\xi_i, \xi_j), U_{m'}(\xi_k, \xi_\ell)) \\ &+ \sum_{1 \leq i, j, k \leq n} \bar{w}_{ij} \bar{\sigma}_k \text{Cov}(U_m(\xi_i, \xi_j), f_{m'}(\xi_k)) \\ &+ \sum_{1 \leq i, j, k \leq n} \bar{w}_{ij} \bar{\sigma}_k \text{Cov}(U_{m'}(\xi_i, \xi_j), f_m(\xi_k)) \\ &+ \sum_{1 \leq i, j \leq n} \bar{\sigma}_i \bar{\sigma}_j \text{Cov}(f_m(\xi_i), f_{m'}(\xi_j)). \end{aligned}$$

The proof is then concluded with the following remarks, which rely on the fact that the random variables $\xi_{[n]}$ are independent and identically distributed.

1. $\text{Cov}(f_m(\xi_j), f_{m'}(\xi_j)) = 0$ unless $i \neq j$, therefore

$$\sum_{1 \leq i, j \leq n} \bar{\sigma}_i \bar{\sigma}_j \text{Cov}(f_m(\xi_i), f_{m'}(\xi_j)) = \left(\sum_{i=1}^n \bar{\sigma}_i^2 \right) \text{Cov}(f_m(\xi_1), f_{m'}(\xi_1)).$$
2. By definition (56) of U_m , $\text{Cov}(U_m(\xi_i, \xi_j), f_{m'}(\xi_k)) = 0$ unless $i = j = k$, hence

$$\sum_{1 \leq i, j, k \leq n} \bar{w}_{ij} \bar{\sigma}_k \text{Cov}(U_m(\xi_i, \xi_j), f_{m'}(\xi_k)) = \left(\sum_{i=1}^n \bar{w}_{ii} \bar{\sigma}_i \right) \text{Cov}(U_m(\xi_1, \xi_1), f_{m'}(\xi_1)).$$
3. By definition (56) of U_m , $\text{Cov}(U_m(\xi_i, \xi_j), U_m(\xi_k, \xi_\ell)) = 0$ unless $i = j = k = \ell$ or $i = k \neq j = \ell$ or $i = \ell \neq j = k$. It follows that

$$\sum_{1 \leq i, j, k, \ell \leq n} \bar{w}_{ij} \bar{w}_{k\ell} \text{Cov}(U_m(\xi_i, \xi_j), U_{m'}(\xi_k, \xi_\ell))$$

$$\begin{aligned} &= \left(\sum_{1 \leq i \neq j \leq n} \bar{w}_{ij}^2 + \bar{w}_{ij} \bar{w}_{ji} \right) \text{Cov}(U_m(\xi_1, \xi_2), U_{m'}(\xi_1, \xi_2)) \\ &+ \left(\sum_{i=1}^n \bar{w}_{ii}^2 \right) \text{Cov}(U_m(\xi_1, \xi_1), U_{m'}(\xi_1, \xi_1)). \end{aligned}$$

■

References

- David M. Allen. The relationship between variable selection and data augmentation and a method for prediction. *Technometrics*, 16:125–127, 1974.
- Sylvain Arlot. V -fold cross-validation improved: V -fold penalization, February 2008. <http://arxiv.org/pdf/0802.0566v2.pdf>.
- Sylvain Arlot. Model selection by resampling penalization. *Electronic Journal of Statistics*, 3:557–624 (electronic), 2009.
- Sylvain Arlot and Alain Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, 2010.
- Sylvain Arlot and Matthieu Lerasle. V -fold cross-validation and V -fold penalization in least-squares density estimation, October 2012. <http://arxiv.org/pdf/1210.5830v1.pdf>.
- Jean-Yves Audibert. A better variance control for pac-bayesian classification. Technical Report 905b, Laboratoire de Probabilités et Modèles Aléatoires, 2004. Available electronically at <http://imagine.enpc.fr/publications/papers/04PMA-905Bis.pdf>.
- Andrew Barron, Lucien Birgé, and Pascal Massart. Risk bounds for model selection via penalization. *Probability Theory and Related Fields*, 113(3):301–413, 1999.
- Yoshua Bengio and Yves Grandvalet. Bias in estimating the variance of K -fold cross-validation. In *Statistical Modeling and Analysis for Complex Data Problems*, volume 1 of *GERAD 25th Anniversary Series*, pages 75–95. Springer, New York, 2005.
- Lucien Birgé. Model selection for density estimation with \mathbb{L}_2 -loss. *Probability Theory and Related Fields*, pages 1–42, 2013.
- Lucien Birgé and Yves Rozenholc. How many bins should be put in a regular histogram. *ESAIM: Probability and Statistics*, 10, 2006.
- Leo Breiman and Philip Spector. Submodel Selection and Evaluation in Regression. The X -Random Case. *International Statistical Review*, 60(3):291–319, 1992.
- Florentina Bunea, Alexandre B. Tsybakov, Marten H. Wegkamp, and Adrian Barbu. Spades and mixture models. *The Annals of Statistics*, 38(4):2525–2558, 2010.

- Prabir Burman. A comparative study of ordinary cross-validation, v -fold cross-validation and the repeated learning-testing methods. *Biometrika*, 76(3):503–514, 1989.
- Prabir Burman. Estimation of optimal transformations using v -fold cross validation and repeated learning-testing methods. *Sankhyā (Statistics)*. Series A, 52(3):314–345, 1990.
- Olivier Catoni. *Pac-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning*, volume 56 of *IMS Lecture Notes Monograph Series*. Institute of Mathematical Statistics, 2007.
- Alain Celisse. *Model Selection Via Cross-Validation in Density Estimation, Regression and Change-Points Detection*. PhD thesis, University Paris-Sud 11, December 2008. Available electronically at <http://tel.archives-ouvertes.fr/tel-00346320/>.
- Alain Celisse. Optimal cross-validation in density estimation with the L^2 -loss. *The Annals of Statistics*, 42(5):1879–1910, 10 2014.
- Alain Celisse and Stéphane Robin. Nonparametric density estimation by exact leave- p -out cross-validation. *Computational Statistics & Data Analysis*, 52(5):2350–2368, 2008.
- Ronald A. DeVore and George G. Lorentz. *Constructive Approximation*, volume 303 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1993.
- Bradley Efron. Estimating the error rate of a prediction rule: improvement on cross-validation. *Journal of the American Statistical Association*, 78(382):316–331, 1983.
- Seymour Geisser. The predictive sample reuse method with applications. *Journal of the American Statistical Association*, 70:320–328, 1975.
- Alexander Goldenshluger and Oleg Lepski. Bandwidth selection in kernel density estimation: oracle inequalities and adaptive minimax optimality. *The Annals of Statistics*, 39(3):1608–1632, 2011.
- Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, New York, second edition, 2009. Data Mining, Inference, and Prediction.
- Christian Houdré and Patricia Reynaud-Bouret. Exponential inequalities, with constants, for U -statistics of order two. In *Stochastic Inequalities and Applications*, volume 56 of *Progress in Probability*, pages 55–69. Birkhäuser, Basel, 2003.
- Matthieu Lerasle. Optimal model selection for stationary data under various mixing conditions. *The Annals of Statistics*, 39(4):1852–1877, 2011.
- Matthieu Lerasle. Optimal model selection in density estimation. *Annales de l'Institut Henri Poincaré. Probabilités et Statistiques*, 48(3):884–908, 2012.
- Wei Liu and Yuhong Yang. Parametric or nonparametric? A parametricness index for model selection. *The Annals of Statistics*, 39(4):2074–2102, 2011.
- Nelo Magalhães. *Cross-Validation and Penalization for Density Estimation*. PhD thesis, University Paris-Sud 11, May 2015. Available electronically at <http://tel.archives-ouvertes.fr/tel-01164581/>.
- Pascal Massart. *Concentration Inequalities and Model Selection*, volume 1896 of *Lecture Notes in Mathematics*. Springer, Berlin, 2007. Lectures from the 33rd Summer School on Probability Theory held in Saint-Flour, July 6–23, 2003.
- Richard R. Picard and R. Dennis Cook. Cross-validation of regression models. *Journal of the American Statistical Association*, 79(387):575–583, 1984.
- Philippe Rigollet. Adaptive density estimation using the blockwise Stein method. *Bernoulli*, 12(2):351–370, 2006.
- Philippe Rigollet and Alexander B. Tsybakov. Linear and convex aggregation of density estimators. *Mathematical Methods of Statistics*, 16(3):260–280, 2007.
- Mats Rudemo. Empirical choice of histograms and kernel density estimators. *Scandinavian Journal of Statistics. Theory and Applications*, 9(2):65–78, 1982.
- Jun Shao. An asymptotic theory for linear model selection. *Statistica Sinica*, 7(2):221–264, 1997. With comments and a rejoinder by the author.
- Mervyn Stone. Cross-validated choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B. Methodological*, 36:111–147, 1974.
- Mark J. van der Laan and Sandrine Dudoit. Unified cross-validation methodology for selection among estimators and a general cross-validated adaptive epsilon-net estimator: Finite sample oracle inequalities and examples. Working Paper 130, U.C. Berkeley Division of Biostatistics, November 2003. Available electronically at <http://www.bepress.com/ucbbiostat/paper130>.
- Mark J. van der Laan, Sandrine Dudoit, and Sunduz Keles. Asymptotic optimality of likelihood-based cross-validation. *Statistical Applications in Genetics and Molecular Biology*, 3:Art. 4, 27 pp. (electronic), 2004.
- Yuhong Yang. Can the strengths of AIC and BIC be shared? A conflict between model identification and regression estimation. *Biometrika*, 92(4):937–950, 2005.
- Yuhong Yang. Comparing learning methods for classification. *Statistica Sinica*, 16(2):635–657, 2006.
- Yuhong Yang. Consistency of cross validation for comparing regression procedures. *The Annals of Statistics*, 35(6):2450–2473, 2007.

Towards More Efficient SPSPD Matrix Approximation and CUR Matrix Decomposition

Shusen Wang

Department of Statistics
University of California at Berkeley
Berkeley, CA 94720, USA

SHUSEN@BERKELEY.EDU

Zhihua Zhang

School of Mathematical Sciences
Peking University
Beijing 100871, China

ZHZHANG@MATH.PKU.EDU.CN

Tong Zhang

Department of Statistics
Rutgers University
Piscataway, New Jersey 08854, USA

TZHANG@STAT.RUTGERS.EDU

Editor: Gert Lanckriet

Abstract

Symmetric positive semi-definite (SPSPD) matrix approximation methods have been extensively used to speed up large-scale eigenvalue computation and kernel learning methods. The standard sketch based method, which we call the prototype model, produces relatively accurate approximations, but is inefficient on large square matrices. The Nyström method is highly efficient, but can only achieve low accuracy. In this paper we propose a novel model that we call the *fast SPSPD matrix approximation model*. The fast model is nearly as efficient as the Nyström method and as accurate as the prototype model. We show that the fast model can potentially solve eigenvalue problems and kernel learning problems in linear time with respect to the matrix size n to achieve $1 + \epsilon$ relative-error, whereas both the prototype model and the Nyström method cost at least quadratic time to attain comparable error bound. Empirical comparisons among the prototype model, the Nyström method, and our fast model demonstrate the superiority of the fast model. We also contribute new understandings of the Nyström method. The Nyström method is a special instance of our fast model and is approximation to the prototype model. Our technique can be straightforwardly applied to make the CUR matrix decomposition more efficiently computed without much affecting the accuracy.

Keywords: Kernel approximation, matrix factorization, the Nyström method, CUR matrix decomposition

1. Introduction

With limited computational and storage resource, machine-precision inversion and decompositions of large and dense matrix are prohibitive. In the past decade matrix approximation techniques have been extensively studied by the theoretical computer science community

(Woodruff, 2014), the machine learning community (Mahoney, 2011), and the numerical linear algebra community (Halko et al., 2011).

In machine learning, many graph analysis techniques and kernel methods require expensive matrix computations on symmetric matrices. The truncated eigenvalue decomposition (that is to find a few eigenvectors corresponding to the greatest eigenvalues) is widely used in graph analysis such as spectral clustering, link prediction in social networks (Shin et al., 2012), graph matching (Patro and Kingsford, 2012), etc. Kernel methods (Schölkopf and Smola, 2002) such as kernel PCA and manifold learning require the truncated eigenvalue decomposition. Some other kernel methods such as Gaussian process regression/classification require solving $n \times n$ matrix inversion, where n is the number of training samples. The rank k ($k \ll n$) truncated eigenvalue decomposition (k -eigenvalue decomposition for short) of an $n \times n$ matrix costs time $\tilde{O}(n^2k)^1$; the matrix inversion costs time $\mathcal{O}(n^3)$. Thus, the standard matrix computation approaches are infeasible when n is large.

For kernel methods, we are typically given n data samples of dimension d , while the $n \times n$ kernel matrix \mathbf{K} is unknown beforehand and should be computed. This adds to the additional $\mathcal{O}(n^2d)$ time cost. When n and d are both large, computing the kernel matrix is prohibitively expensive. Thus, a good kernel approximation method should avoid the computation of the entire kernel matrix.

Typical SPSPD matrix approximation methods speed up matrix computation by efficiently forming a low-rank decomposition $\mathbf{K} \approx \mathbf{CUC}^T$ where $\mathbf{C} \in \mathbb{R}^{n \times c}$ is a sketch of \mathbf{K} (e.g., randomly sampled c columns of \mathbf{K}) and $\mathbf{U} \in \mathbb{R}^{c \times c}$ can be computed in different ways. With such a low-rank approximation at hand, it takes only $\mathcal{O}(nc^2)$ additional time to approximately compute the rank k ($k \leq c$) eigenvalue decomposition or the matrix inversion. Therefore, if \mathbf{C} and \mathbf{U} are obtained in linear time (w.r.t. n) and c is independent of n , then the aforementioned eigenvalue decomposition and matrix inversion can be approximately solved in linear time.

The Nyström method is perhaps the most widely used kernel approximation method. Let \mathbf{P} be an $n \times c$ sketching matrix such as uniform sampling (Williams and Seeger, 2001; Gittens, 2011), adaptive sampling (Kumar et al., 2012), leverage score sampling (Gittens and Mahoney, 2016), etc. The Nyström method computes \mathbf{C} by $\mathbf{C} = \mathbf{KP} \in \mathbb{R}^{n \times c}$ and $\mathbf{U} = (\mathbf{P}^T \mathbf{C})^\dagger \in \mathbb{R}^{c \times c}$. This way of computing \mathbf{U} is very efficient, but it incurs relatively large approximation error even if \mathbf{C} is a good sketch of \mathbf{K} . As a result, the Nyström method is reported to have low approximation accuracy in real-world applications (Dai et al., 2014; Hsieh et al., 2014; Si et al., 2014b). In fact, the Nyström is impossible to attain $1 + \epsilon$ bound relative to $\|\mathbf{K} - \mathbf{K}_k\|_F^2$ unless $c \geq \Omega(\sqrt{nk}/\epsilon)$ (Wang and Zhang, 2013). Here \mathbf{K}_k denotes the best rank- k approximation of \mathbf{K} . The requirement that c grows at least linearly with \sqrt{n} is a very pessimistic result. It implies that in order to attain $1 + \epsilon$ relative-error bound, the time cost of the Nyström method is of order $nc^2 = \Omega(n^2k/\epsilon)$ for solving the k -eigenvalue decomposition or matrix inversion, which is quadratic in n . Therefore, under the $1 + \epsilon$ relative-error requirement, the Nyström method is not a linear time method.

The main reason for the low accuracy of the Nyström method is due to the way that the \mathbf{U} matrix is calculated. In fact, much higher accuracy can be obtained if \mathbf{U} is calculated

1. The \tilde{O} notation hides the logarithm factors.

by solving the minimization problem $\min_{\mathbf{U}} \|\mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^T\|_F^2$, which is a standard way to approximate symmetric matrices (Halcko et al., 2011; Gittens and Mahoney, 2016; Wang and Zhang, 2013; Wang et al., 2016). This is the randomized SVD for symmetric matrices (Halcko et al., 2011). Wang et al. (2016) called this approxized the prototype model and provided an algorithm that samples $c = \mathcal{O}(k/\epsilon)$ columns of \mathbf{K} to form \mathbf{C} such that $\min_{\mathbf{U}} \|\mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^T\|_F^2 \leq (1+\epsilon)\|\mathbf{K} - \mathbf{K}_k\|_F^2$. Unlike the Nystrom method, the prototype model does not require c to grow with n . The downside of the prototype model is the high computational cost. It requires the full observation of \mathbf{K} and $\mathcal{O}(n^2c)$ time to compute \mathbf{U} . Therefore when applied to kernel approximation, the time cost cannot be less than $\mathcal{O}(n^2d + n^2c)$. To reduce the computational cost, this paper considers the problem of efficient calculation of \mathbf{U} with fixed \mathbf{C} while achieving an accuracy comparable to the prototype model.

More specifically, the key question we try to answer in this paper can be described as follows.

Question 1 For any fixed $n \times n$ symmetric matrix \mathbf{K} , target rank k , and parameter γ , assume that

A1 We are given a sketch matrix $\mathbf{C} \in \mathbb{R}^{n \times c}$ of \mathbf{K} , which is obtained in time $\text{Time}(\mathbf{C})$;

A2 The matrix \mathbf{C} is a good sketch of \mathbf{K} in that $\min_{\mathbf{U}} \|\mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^T\|_F^2 \leq (1+\gamma)\|\mathbf{K} - \mathbf{K}_k\|_F^2$.

Then we would like to know whether for an arbitrary ϵ , it is possible to compute \mathbf{C} and \mathbf{U} such that the following two requirements are satisfied:

R1 The matrix \mathbf{U} has the following error bound:

$$\|\mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^T\|_F^2 \leq (1+\epsilon)(1+\gamma)\|\mathbf{K} - \mathbf{K}_k\|_F^2.$$

R2 The procedure of computing \mathbf{C} and \mathbf{U} and approximately solving the aforementioned k -eigenvalue decomposition or the matrix inversion run in time $\mathcal{O}(n \cdot \text{poly}(k, \gamma^{-1}, \epsilon^{-1})) + \text{Time}(\mathbf{C})$.

Unfortunately, the following theorem shows that neither the Nystrom method nor the prototype model enjoys such desirable properties. We prove the theorem in Appendix B.

Theorem 1 Neither the Nystrom method nor the prototype model satisfies the two requirements in Question 1. To make requirement R1 hold, both the Nystrom method and the prototype model cost time no less than $\mathcal{O}(n^2 \cdot \text{poly}(k, \gamma^{-1}, \epsilon^{-1})) + \text{Time}(\mathbf{C})$ which is at least quadratic in n .

In this paper we give an affirmative answer to the above question. In particular, it has the following consequences. First, the overall approximation has high accuracy in the sense that $\|\mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^T\|_F^2$ is comparable to $\min_{\mathbf{U}} \|\mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^T\|_F^2$, and is thereby comparable to the best rank k approximation. Second, with \mathbf{C} at hand, the matrix \mathbf{U} is obtained efficiently (linear in n). Third, with \mathbf{C} and \mathbf{U} at hand, it takes extra time which is also linear in n to compute the aforementioned eigenvalue decomposition or linear system. Therefore, with a good \mathbf{C} , we can use linear time to obtain desired \mathbf{U} matrix such that the accuracy is comparable to the best possible low-rank approximation.

The CUR matrix decomposition (Mahoney and Drineas, 2009) is closely related to the prototype model and troubled by the same computational problem. The CUR matrix decomposition is an extension of the prototype model from symmetric matrices to general

matrices. Given any $m \times n$ fixed matrix \mathbf{A} , the CUR matrix decomposition selects c columns of \mathbf{A} to form $\mathbf{C} \in \mathbb{R}^{m \times c}$ and r rows of \mathbf{A} to form $\mathbf{R} \in \mathbb{R}^{r \times n}$, and computes matrix $\mathbf{U} \in \mathbb{R}^{c \times r}$ such that $\|\mathbf{A} - \mathbf{C}\mathbf{U}\mathbf{R}\|_F^2$ is small. Traditionally, it costs time

$$\mathcal{O}(mn \cdot \min\{c, r\})$$

to compute the optimal $\mathbf{U}^* = \mathbf{C}^\dagger \mathbf{A} \mathbf{R}^\dagger$ (Stewart, 1999; Wang and Zhang, 2013; Boutsidis and Woodruff, 2014). How to efficiently compute a high-quality \mathbf{U} matrix for CUR is unsolved.

1.1 Main Results

This work is motivated by an intrinsic connection between the Nystrom method and the prototype model. Based on a generalization of this observation, we propose the *fast SPSPD matrix approximation model* for approximating any symmetric matrix. We show that the fast model satisfies the requirements in Question 1. Given n data points of dimension d , the fast model computes \mathbf{C} and \mathbf{U}^{fast} and approximately solves the truncated eigenvalue decomposition or matrix inversion in time

$$\mathcal{O}(nc^3/\epsilon + nc^2d/\epsilon) + \text{Time}(\mathbf{C}).$$

Here $\text{Time}(\mathbf{C})$ is defined in Question 1.

The fast SPSPD matrix approximation model achieves the desired properties in Question 1 by solving $\min_{\mathbf{U}} \|\mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^T\|_F$ approximately rather than exactly while ensuring

$$\|\mathbf{K} - \mathbf{C}\mathbf{U}^{\text{fast}}\mathbf{C}^T\|_F^2 \leq (1+\epsilon) \min_{\mathbf{U}} \|\mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^T\|_F^2.$$

The time complexity for computing \mathbf{U}^{fast} is linear in n , which is far less than the time complexity $\mathcal{O}(n^2c)$ of the prototype model. Our method also avoids computing the entire kernel matrix \mathbf{K} ; instead, it computes a block of \mathbf{K} of size $\frac{\sqrt{nc}}{\epsilon} \times \frac{\sqrt{nc}}{\epsilon}$, which is substantially smaller than $n \times n$. The lower bound in Theorem 7 indicates that the \sqrt{n} factor here is optimal, but the dependence on c and ϵ are suboptimal and can be potentially improved.

This paper provides a new perspective on the Nystrom method. We show that, as well as our fast model, the Nystrom method is approximate solution to the problem $\min_{\mathbf{U}} \|\mathbf{C}\mathbf{U}\mathbf{C}^T - \mathbf{K}\|_F^2$. Unfortunately, the approximation is so rough that the quality of the Nystrom method is low.

Our method can also be applied to improve the CUR matrix decomposition of the general matrices which are not necessarily square. Given any matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{C} \in \mathbb{R}^{m \times c}$, and $\mathbf{R} \in \mathbb{R}^{r \times n}$, it costs time $\mathcal{O}(mn \cdot \min\{c, r\})$ to compute the matrix $\mathbf{U} = \mathbf{C}^\dagger \mathbf{A} \mathbf{R}^\dagger$. Applying our technique, the time cost drops to only

$$\mathcal{O}(c^2r\epsilon^{-1} \cdot \min\{m, n\} \cdot \min\{c, r\}),$$

while the approximation quality is nearly the same.

1.2 Paper Organization

The remainder of this paper is organized as follows. Section 2 defines the notation used in this paper. Section 3 introduces the related work of matrix sketching and SPSPD matrix

Table 1: A summary of the notation.

Notation	Description
n	number of data points
d	dimension of the data point
\mathbf{K}	$n \times n$ kernel matrix
\mathbf{P}, \mathbf{S}	sketching matrices
\mathbf{C}	$n \times c$ sketch computed by $\mathbf{C} = \mathbf{K}\mathbf{P}$
\mathbf{U}^*	$\mathbf{C}^T(\mathbf{C}^T)^T \in \mathbb{R}^{c \times c}$ —the \mathbf{U} matrix of the prototype model
\mathbf{U}^{NYS}	$(\mathbf{P}^T \mathbf{K})^\dagger \in \mathbb{R}^{c \times c}$ —the \mathbf{U} matrix of the Nyström method
\mathbf{U}^{fast}	$(\mathbf{S}^T \mathbf{C})^\dagger (\mathbf{S}^T \mathbf{K} \mathbf{S})(\mathbf{C}^T \mathbf{S})^\dagger \in \mathbb{R}^{c \times c}$ —the \mathbf{U} matrix of the fast model

approximation. Section 4 describes our fast model and analyze the time complexity and error bound. Section 5 applies the technique of the fast model to compute the CUR matrix decomposition more efficiently. Section 6 conducts empirical comparisons to show the effect of the \mathbf{U} matrix. The proofs of the theorems are in the appendix.

2. Notation

The notation used in this paper are defined as follows. Let $[n] = \{1, \dots, n\}$, \mathbf{I}_n be the $n \times n$ identity matrix, and $\mathbf{1}_n$ be the $n \times 1$ vector of all ones. We let $x \in y \pm z$ denote $y - z \leq x \leq y + z$. For an $m \times n$ matrix $\mathbf{A} = [A_{ij}]$, we let \mathbf{a}_i be its i -th row, $\mathbf{a}_{:j}$ be its j -th column, $\text{nnz}(\mathbf{A})$ be the number of nonzero entries of \mathbf{A} , $\|\mathbf{A}\|_F = (\sum_{i,j} A_{ij}^2)^{1/2}$ be its Frobenius norm, and $\|\mathbf{A}\|_2 = \max_{\mathbf{x} \neq 0} \|\mathbf{A}\mathbf{x}\|_2 / \|\mathbf{x}\|_2$ be its spectral norm.

Let $\rho = \text{rank}(\mathbf{A})$. The condensed singular value decomposition (SVD) of \mathbf{A} is defined

as

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_{i=1}^{\rho} \sigma_i \mathbf{u}_i \mathbf{v}_i^T$$

where $\sigma_1, \dots, \sigma_r$ are the positive singular values in the descending order. We also use $\sigma_i(\mathbf{A})$ to denote the i -th largest singular value of \mathbf{A} . Unless otherwise specified, in this paper “SVD” means the condensed SVD. Let $\mathbf{A}_k = \sum_{i=1}^k \sigma_i \mathbf{u}_i \mathbf{v}_i^T$ be the top k principal components of \mathbf{A} for any positive integer k less than ρ . In fact, \mathbf{A}_k is the closest to \mathbf{A} among all the rank k matrices. Let $\mathbf{A}^\dagger = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^T$ be the Moore-Penrose inverse of \mathbf{A} .

Assume that $\rho = \text{rank}(\mathbf{A}) < n$. The column leverage scores of \mathbf{A} are $l_i = \|\mathbf{v}_i\|_2^2$ for $i = 1$ to n . Obviously, $l_1 + \dots + l_n = \rho$. The row leverage scores and coherence are similarly $\frac{l_i}{\rho} \max_{j \in [n]} \|\mathbf{v}_j\|_2^2$. If $\rho = \text{rank}(\mathbf{A}) < m$, the row leverage scores and coherence are similarly defined. The row leverage scores are $\|\mathbf{u}_1\|_2^2, \dots, \|\mathbf{u}_m\|_2^2$ and the row coherence is $\mu(\mathbf{A}) = \frac{l_i}{\rho} \max_{i \in [m]} \|\mathbf{u}_i\|_2^2$.

We also list some frequently used notation in Table 1. Given the decomposition $\tilde{\mathbf{K}} = \mathbf{C}\mathbf{U}^T \approx \mathbf{K}$ which has rank at most c , it takes $\mathcal{O}(nc^2)$ time to compute the eigenvalue decomposition of $\tilde{\mathbf{K}}$ and $\mathcal{O}(nc^2)$ time to solve the linear system $(\tilde{\mathbf{K}} + \alpha \mathbf{I}_n)\mathbf{w} = \mathbf{y}$ to obtain \mathbf{w} (see Appendix A for more discussions). The truncated eigenvalue decomposition and linear system are the bottleneck of many kernel methods, and thus an accurate and efficient low-rank approximation can help to accelerate the computation of kernel learning.

3. Related Work

In Section 3.1 we introduce matrix sketching. In Section 3.2 we describe two SPSPD matrix approximation methods.

3.1 Matrix Sketching

Popular matrix sketching methods include uniform sampling, leverage score sampling (Drineas et al., 2006, 2008; Woodruff, 2014), Gaussian projection (Johnson and Lindenstrauss, 1984), subsampled randomized Hadamard transform (SRHT) (Drineas et al., 2011; Lu et al., 2013; Tropp, 2011), count sketch (Charikar et al., 2004; Clarkson and Woodruff, 2013; Meng and Mahoney, 2013; Nelson and Nguyen, 2013; Pham and Pagh, 2013; Thorup and Zhang, 2012; Weinberger et al., 2009), etc.

3.1.1 COLUMN SAMPLING

Let $p_1, \dots, p_n \in (0, 1)$ with $\sum_{i=1}^n p_i = 1$ be the sampling probabilities. Let each integer in $[n]$ be independently sampled with probabilities sp_1, \dots, sp_n , where $s \in [n]$ is integer. Assume that \tilde{s} integers are sampled from $[n]$. Let $i_1, \dots, i_{\tilde{s}}$ denote the selected integers, and let $\mathbb{E}[\tilde{s}] = s$. We scale each selected column by $\frac{1}{\sqrt{sp_{i_1}}}, \dots, \frac{1}{\sqrt{sp_{i_{\tilde{s}}}}}$, respectively. Uniform sampling means that the sampling probabilities are $p_1 = \dots = p_n = \frac{1}{n}$. Leverage score sampling means that the sampling probabilities are proportional to the leverage scores l_1, \dots, l_n of a certain matrix.

We can equivalently characterize column selection by the matrix $\mathbf{S} \in \mathbb{R}^{n \times \tilde{s}}$. Each column of \mathbf{S} has exactly one nonzero entry; let (i_j, j) be the position of the nonzero entry in the j -th column for $j \in [\tilde{s}]$. For $j = 1$ to \tilde{s} , we set

$$S_{i_j, j} = \frac{1}{\sqrt{sp_{i_j}}}. \quad (1)$$

The expectation $\mathbb{E}[\tilde{s}]$ equals to s , and $\tilde{s} = \Theta(s)$ with high probability. For the sake of simplicity and clarity, in the rest of this paper we will not distinguish \tilde{s} and s .

3.1.2 RANDOM PROJECTION

Let $\mathbf{G} \in \mathbb{R}^{n \times s}$ be a standard Gaussian matrix, namely each entry is sampled independently from $\mathcal{N}(0, 1)$. The matrix $\mathbf{S} = \frac{1}{\sqrt{s}}\mathbf{G}$ is a Gaussian projection matrix. Gaussian projection is also well known as the Johnson-Lindenstrauss (JL) transform (Johnson and Lindenstrauss, 1984); its theoretical property is well established. It takes $\mathcal{O}(mns)$ time to apply $\mathbf{S} \in \mathbb{R}^{n \times s}$ to any $m \times n$ dense matrix, which makes Gaussian projection inefficient.

The subsampled randomized Hadamard transform (SRHT) is usually a more efficient alternative of Gaussian projection. Let $\mathbf{H}_n \in \mathbb{R}^{n \times n}$ be the Walsh-Hadamard matrix with $+1$ and -1 entries, $\mathbf{D} \in \mathbb{R}^{n \times n}$ be a diagonal matrix with diagonal entries sampled uniformly from $\{+1, -1\}$, and $\mathbf{P} \in \mathbb{R}^{n \times s}$ be the uniform sampling matrix defined above. The matrix $\mathbf{S} = \frac{1}{\sqrt{n}}\mathbf{D}\mathbf{H}_n\mathbf{P} \in \mathbb{R}^{n \times s}$ is an SRHT matrix, and it can be applied to any $m \times n$ matrix in $\mathcal{O}(mn \log s)$ time.

Count sketch stems from the data stream literature (Charikar et al., 2004; Thorup and Zhang, 2012) and has been applied to speedup matrix computation. The count sketch

matrix $\mathbf{S} \in \mathbb{R}^{n \times s}$ can be applied to any matrix \mathbf{A} in $\mathcal{O}(\text{nnz}(\mathbf{A}))$ time where nnz denotes the number of non-zero entries. The readers can refer to (Woodruff, 2014) for detailed descriptions of count sketch.

3.1.3 THEORIES

The following lemma shows important properties of the matrix sketching methods. In the lemma, leverage score sampling means that the sampling probabilities are proportional to the row leverage scores of the column orthogonal matrix $\mathbf{U} \in \mathbb{R}^{n \times k}$. (Here \mathbf{U} is different from the notation elsewhere in the paper.) We prove the lemma in Appendix C.

Lemma 2 *Let $\mathbf{U} \in \mathbb{R}^{n \times k}$ be any fixed matrix with orthonormal columns and $\mathbf{B} \in \mathbb{R}^{n \times d}$ be any fixed matrix. Let $\mathbf{S} \in \mathbb{R}^{n \times s}$ be any sketching matrix considered in this section; the order of s (with the \mathcal{O} -notation omitted) is listed in Table 2. Then*

$$\begin{aligned} \mathbb{P}\left\{\|\mathbf{U}^T \mathbf{S} \mathbf{S}^T \mathbf{U} - \mathbf{I}_k\|_2 \geq \eta\right\} &\leq \delta_1 && \text{(Property 1),} \\ \mathbb{P}\left\{\|\mathbf{U}^T \mathbf{B} - \mathbf{U}^T \mathbf{S} \mathbf{S}^T \mathbf{B}\|_F^2 \geq \epsilon \|\mathbf{B}\|_F^2\right\} &\leq \delta_2 && \text{(Property 2),} \\ \mathbb{P}\left\{\|\mathbf{U}^T \mathbf{B} - \mathbf{U}^T \mathbf{S} \mathbf{S}^T \mathbf{B}\|_2^2 \geq \epsilon' \|\mathbf{B}\|_2^2 + \frac{\epsilon'}{k} \|\mathbf{B}\|_F^2\right\} &\leq \delta_3 && \text{(Property 3).} \end{aligned}$$

Table 2: The leverage score sampling is w.r.t. the row leverage scores of \mathbf{U} . For uniform sampling, the notation $\mu(\mathbf{U}) \in [1, \eta]$ is the row coherence of \mathbf{U} .

Sketching	Property 1	Property 2	Property 3
Leverage Sampling	$\frac{k}{\eta} \log \frac{k}{\delta_1}$	$\frac{k}{\delta_2}$	—
Uniform Sampling	$\frac{\mu(\mathbf{U})k}{\eta^2} \log \frac{k}{\delta_1}$	$\frac{\mu(\mathbf{U})}{\delta_2}$	—
Gaussian Projection	$\frac{k + \log(1/\delta_1)}{k \log(1/\delta_1)}$	$\frac{k}{\delta_2}$	—
SRHT	$\frac{k + \log(1/\delta_1)}{\eta^2} \log \frac{k}{\delta_1}$	$\frac{k + \log(1/\delta_1)}{\delta_2}$	$\frac{1}{\epsilon'} (k + \log \frac{d}{\delta_3})$
Count Sketch	$\frac{k}{\delta_1^2}$	$\frac{k}{\delta_2}$	$\frac{1}{\epsilon'} (k + \log \frac{nd}{k \delta_1}) \log \frac{d}{\delta_3}$

Property 1 is known as the subspace embedding property (Woodruff, 2014). It shows that all the singular values of $\mathbf{S}^T \mathbf{U}$ are close to one. Properties 2 and 3 show that sketching preserves the multiplication of a row orthogonal matrix and an arbitrary matrix.

For the SPSPD/CUR matrix approximation problems, the three properties are all we need to capture the randomness in the sketching methods. Leverage score sampling, uniform sampling, and count sketch do not enjoy Property 3, but it is fine—Frobenius norm (Property 2) will be used as a loose upper bound on the spectral norm (Property 3). Gaussian projection and SRHT satisfy all the three properties; when applied to the SPSPD/CUR problems, their error bounds are stronger than the leverage score sampling, uniform sampling, and count sketch.

3.2 SPSPD Matrix Approximation Models

We first describe the prototype model and the Nystrom method, which are most relevant to this work. We then introduce several other SPSPD matrix approximation methods.

3.2.1 MOST RELEVANT WORK

Given an $n \times n$ matrix \mathbf{K} and an $n \times c$ sketching matrix \mathbf{P} , we let $\mathbf{C} = \mathbf{K}\mathbf{P}$ and $\mathbf{W} = \mathbf{P}^T \mathbf{C} = \mathbf{P}^T \mathbf{K}\mathbf{P}$. The *prototype model* (Wang and Zhang, 2013) is defined by

$$\tilde{\mathbf{K}}_c^{\text{proto}} \triangleq \mathbf{C}\mathbf{U}^T \mathbf{C}^T = \mathbf{C}\mathbf{C}^t \mathbf{K}(\mathbf{C}^t)^T \mathbf{C}^T, \quad (2)$$

and the *Nystrom method* is defined by

$$\begin{aligned} \tilde{\mathbf{K}}_c^{\text{nys}} &\triangleq \mathbf{C}\mathbf{U}^{\text{nys}} \mathbf{C}^T = \mathbf{C}\mathbf{W}^t \mathbf{C}^T \\ &= \mathbf{C}(\mathbf{P}^T \mathbf{C})^t (\mathbf{P}^T \mathbf{K}\mathbf{P})(\mathbf{C}^T \mathbf{P})^t \mathbf{C}^T. \end{aligned} \quad (3)$$

The only difference between the two models is their \mathbf{U} matrices, and the difference leads to big difference in their approximation accuracies. Wang and Zhang (2013) provided a lower error bound of the Nystrom method, which shows that no algorithm can select less than $\Omega(\sqrt{nk/\epsilon})$ columns of \mathbf{K} to form \mathbf{C} such that

$$\|\mathbf{K} - \mathbf{C}\mathbf{U}^{\text{nys}} \mathbf{C}^T\|_F^2 \leq (1 + \epsilon) \|\mathbf{K} - \mathbf{K}_k\|_F^2.$$

In contrast, the prototype model can attain the $1 + \epsilon$ relative-error bound with $c = \mathcal{O}(k/\epsilon)$ (Wang et al., 2016), which is optimal up to a constant factor.

While we have mainly discussed the time complexity of kernel approximation in the previous sections, the memory cost is often a more important issue in large scale problems due to the limitation of computer memory. The Nystrom method and the prototype model require $\mathcal{O}(nc)$ memory to hold \mathbf{C} and \mathbf{U} to approximately solve the aforementioned eigenvalue decomposition or the linear system.² Therefore, we hope to make c as small as possible while achieving a low approximation error. There are two elements: (1) a good sketch $\mathbf{C} = \mathbf{K}\mathbf{P}$, and (2) a high-quality \mathbf{U} matrix. We focus on the latter in this paper.

3.2.2 LESS RELEVANT WORK

We note that there are many other kernel approximation approaches in the literature. However, these approaches do not directly address the issue we consider here, so they are complementary to our work. These studies are either less effective or inherently rely on the Nystrom method.

The Nystrom-like models such as MEKA (Si et al., 2014a) and the ensemble Nystrom method (Kumar et al., 2012) are reported to significantly outperform the Nystrom method in terms of approximation accuracy, but their key components are still the Nystrom method and the component can be replaced by any other methods such as the method studied in this work. The spectral shifting Nystrom method (Wang et al., 2014) also outperforms the

² The memory costs of the prototype model is $\mathcal{O}(nc + nd)$ rather than $\mathcal{O}(n^2)$. This is because we can hold the $n \times d$ data matrix and the $c \times n$ matrix \mathbf{C}^t in memory; compute a small block of \mathbf{K} each time; and then compute $\mathbf{C}^t \mathbf{K}$ block by block.

Nyström method in certain situations, but the spectral shifting strategy can be used for any other kernel approximation models beyond the prototype model. We do not compare with these methods in this paper because MEKA, the ensemble Nyström method, and the spectral shifting Nyström method can all be improved if we replace the underlying Nyström method or the prototype model by the new method developed here.

The column-based low-rank approximation model (Kumar et al., 2009) is another SPSPD matrix approximation approach different from the Nyström-like methods. Let $\mathbf{P} \in \mathbb{R}^{n \times c}$ be any sketching matrix and $\mathbf{C} = \mathbf{K}\mathbf{P}$. The column-based model approximates \mathbf{K} by $\mathbf{C}(\mathbf{C}^T\mathbf{C})^{-1/2}\mathbf{C}^T = (\mathbf{C}\mathbf{C}^T)^{1/2}$. Equivalently, it approximates \mathbf{K}^2 by

$$\mathbf{K}^T\mathbf{K} \approx \mathbf{C}\mathbf{C}^T = \mathbf{K}^T\mathbf{P}\mathbf{P}^T\mathbf{K}.$$

From Lemma 2 we can see that it is a typical sketch based approximation to the matrix multiplication. Unfortunately, the approximate matrix multiplication is effective only when \mathbf{K} has much more rows than columns, which is not true for the kernel matrix. The column-based model does not have good error bound and is not empirically as good as the Nyström method (Kumar et al., 2009).

The random feature mapping (Rahimi and Recht, 2007) is a family of kernel approximation methods. Each random feature mapping method is applicable to certain kernel rather than arbitrary SPSPD matrix. Furthermore, they are known to be noticeably less effective than the Nyström method (Yang et al., 2012).

4. The Fast SPSPD Matrix Approximation Model

In Section 4.1 we present the motivation behind the fast model. In Section 4.2 we provide an alternative perspective on our fast model and the Nyström method by formulating them as approximate solutions to an optimization problem. In Section 4.3 we analyze the error bound of the fast model. Theorem 3 is the main theorem, which shows that in terms of the Frobenius norm approximation, the fast model is almost as good as the prototype model. In Section 4.4 we describe the implementation of the fast model and analyze the time complexity. In Section 4.5 we give some implementation details that help to improve the approximation quality. In Section 4.6 we show that our fast model exactly recovers \mathbf{K} under certain conditions, and we provide a lower error bound of the fast model.

4.1 Motivation

Let $\mathbf{P} \in \mathbb{R}^{n \times c}$ be sketching matrix and $\mathbf{C} = \mathbf{K}\mathbf{P} \in \mathbb{R}^{n \times c}$. The fast SPSPD matrix approximation model is defined by

$$\tilde{\mathbf{K}}_{c,s}^{\text{fast}} \triangleq \mathbf{C}(\mathbf{S}^T\mathbf{C})^\dagger(\mathbf{S}^T\mathbf{K}\mathbf{S})(\mathbf{C}^T\mathbf{S})^\dagger\mathbf{C}^T,$$

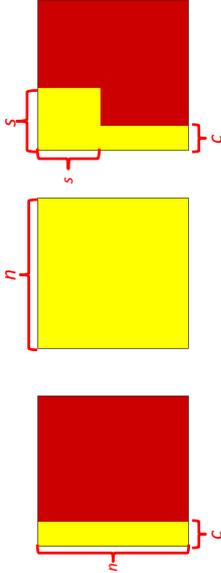
where \mathbf{S} is $n \times s$ sketching matrix.

From (2) and (3) we can see that the Nyström method is a special case of the fast model where \mathbf{S} is defined as \mathbf{P} and that the prototype model is a special case where \mathbf{S} is defined as \mathbf{I}_n .

The fast model allows us to trade off the accuracy and the computational cost—larger s leads to higher accuracy and higher time cost, and vice versa. Setting s as small as c

Table 3: Summary of the time cost of the models for computing the \mathbf{U} matrices and the number of entries of \mathbf{K} required to be observed in order to compute the \mathbf{U} matrices. As for the fast model, assume that \mathbf{S} is column selection matrix. The notation is defined previously in Table 1.

	Time	#Entries
Nyström	$\mathcal{O}(c^3)$	nc
Prototype	$\mathcal{O}(\min(\mathbf{K})c + nc^2)$	n^2
Fast	$\mathcal{O}(nc^2 + s^2c)$	$nc + (s-c)^2$



Nyström Prototype Fast

Figure 1: The yellow blocks denote the submatrices of \mathbf{K} that must be seen by the kernel approximation models. The Nyström method computes an $n \times c$ block of \mathbf{K} , provided that \mathbf{P} is column selection matrix; the prototype model computes the entire $n \times n$ matrix \mathbf{K} ; the fast model computes an $n \times c$ block and an $(s-c) \times (s-c)$ block of \mathbf{K} (due to the symmetry of \mathbf{K}), provided that \mathbf{P} and \mathbf{S} are column selection matrices.

sacrifices too much accuracy, whereas setting s as large as n is unnecessarily expensive. Later on, we will show that $s = \mathcal{O}(c\sqrt{n/\epsilon}) \ll n$ is a good choice. The setting $s \ll n$ makes the fast model much cheaper to compute than the prototype model. When applied to kernel methods, the fast model avoids computing the entire kernel matrix. We summarize the time complexities of the three matrix approximation methods in Table 3; the middle column lists the time cost for computing the \mathbf{U} matrices given \mathbf{C} and \mathbf{K} ; the right column lists the number of entry of \mathbf{K} which must be observed. We show a very intuitive comparison in Figure 1.

4.2 Optimization Perspective

With the sketch $\mathbf{C} = \mathbf{K}\mathbf{P} \in \mathbb{R}^{n \times c}$ at hand, we want to find the \mathbf{U} matrix such that $\mathbf{C}\mathbf{U}\mathbf{C}^T \approx \mathbf{K}$. It is very intuitive to solve the following problem to make the approximation

tight:

$$\mathbf{U}^* = \underset{\mathbf{U}}{\operatorname{argmin}} \|\mathbf{C}\mathbf{U}\mathbf{C}^T - \mathbf{K}\|_F^2 = \mathbf{C}^\dagger \mathbf{K} (\mathbf{C}^\dagger)^T. \quad (4)$$

This is the prototype model. Since solving this system is time expensive, we propose to draw a sketching matrix $\mathbf{S} \in \mathbb{R}^{n \times s}$ and solve the following problem instead:

$$\begin{aligned} \mathbf{U}^{\text{fast}} &= \underset{\mathbf{U}}{\operatorname{argmin}} \|\mathbf{S}^T (\mathbf{C}\mathbf{U}\mathbf{C}^T - \mathbf{K}) \mathbf{S}\|_F^2 \\ &= \underset{\mathbf{U}}{\operatorname{argmin}} \|(\mathbf{S}^T \mathbf{C}) \mathbf{U} (\mathbf{S}^T \mathbf{C})^T - \mathbf{S}^T \mathbf{K} \mathbf{S}\|_F^2 \\ &= (\mathbf{S}^T \mathbf{C})^\dagger (\mathbf{S}^T \mathbf{K} \mathbf{S}) (\mathbf{C}^T \mathbf{S})^\dagger, \end{aligned} \quad (5)$$

which results in the fast model. Similar ideas have been exploited to efficiently solve the least squares regression problem (Drineas et al., 2006, 2011; Clarkson and Woodruff, 2013), but their analysis can not be directly applied to the more complicated system (5).

This approximate linear system interpretation offers a new perspective on the Nystrom method. The \mathbf{U} matrix of the Nystrom method is in fact an approximate solution to the problem $\min_{\mathbf{U}} \|\mathbf{C}\mathbf{U}\mathbf{C}^T - \mathbf{K}\|_F^2$. The Nystrom method uses $\mathbf{S} = \mathbf{P}$ as the sketching matrix, which leads to the solution

$$\mathbf{U}^{\text{NYS}} = \underset{\mathbf{U}}{\operatorname{argmin}} \|\mathbf{P}^T (\mathbf{C}\mathbf{U}\mathbf{C}^T - \mathbf{K}) \mathbf{P}\|_F^2 = (\mathbf{P}^T \mathbf{K} \mathbf{P})^\dagger = \mathbf{W}^\dagger.$$

4.3 Error Analysis

Let \mathbf{U}^{fast} correspond to the fast model (5). Any of the five sketching methods in Lemma 2 can be used to compute \mathbf{U}^{fast} , although column selection is more useful than random projection in this application. In the following we show that \mathbf{U}^{fast} is nearly as good as \mathbf{U}^* in terms of the objective function value. The proof is in Appendix D.

Theorem 3 (Main Result) *Let \mathbf{K} be any $n \times n$ fixed symmetric matrix, \mathbf{C} be any $n \times c$ fixed matrix, $k_c = \operatorname{rank}(\mathbf{C})$, and \mathbf{U}^{fast} be the $c \times c$ matrix defined in (5). Let $\mathbf{S} \in \mathbb{R}^{n \times s}$ be any of the five sketching matrices defined in Table 4. Assume that $\epsilon^{-1} = o(n)$ or $\epsilon^{-1} = o(n/c)$. The inequality*

$$\|\mathbf{K} - \mathbf{C}\mathbf{U}^{\text{fast}}\mathbf{C}^T\|_F^2 \leq (1 + \epsilon) \min \|\mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^T\|_F^2 \quad (6)$$

holds with probability at least 0.8.

In the theorem, Gaussian projection and SRHT require smaller sketch size than the other three methods. It is because Gaussian projection and SRHT enjoys all of Properties 1, 2, 3 in Lemma 2, whereas leverage score sampling, uniform sampling, and count sketch does not enjoy Property 3.

Remark 4 Wang et al. (2016) showed that there exists an algorithm (though not linear-time algorithm) attaining the error bound

$$\|\mathbf{K} - \mathbf{C}\mathbf{C}^\dagger \mathbf{K} (\mathbf{C}^\dagger)^T \mathbf{C}^T\|_F^2 \leq (1 + \epsilon) \|\mathbf{K} - \mathbf{K}_k\|_F^2$$

Table 4: Leverage score sampling means sampling according to the row leverage scores of \mathbf{C} . For uniform sampling, the parameter $\mu(\mathbf{C}) \in [1, n]$ is the row coherence of \mathbf{C} .

Sketching	Order of s	Assumption	T_{sketch}	#Entries
Leverage Score Sampling	$c\sqrt{n}/\epsilon$	$\epsilon = o(n)$	$\mathcal{O}(nc^2 + s^2)$	$nc + (s - c)^2$
Uniform Sampling	$\mu(\mathbf{C})c\sqrt{n}/\epsilon$	$\epsilon = o(n)$	$\mathcal{O}(s^2)$	n^2
Gaussian Projection	$\sqrt{\frac{nc}{\epsilon}}(c + \log \frac{n}{\epsilon})$	$\epsilon = o(n/c)$	$\mathcal{O}(\operatorname{mnz}(\mathbf{K})s)$	n^2
SRHT	$\sqrt{\frac{nc}{\epsilon}}(c + \log n) \log(n)$	$\epsilon = o(n/c)$	$\mathcal{O}(n^2 \log s)$	n^2
Count Sketch	$c\sqrt{n}/\epsilon$	$\epsilon = o(n)$	$\mathcal{O}(\operatorname{mnz}(\mathbf{K}))$	n^2

Algorithm 1 The Fast SPSPD Matrix Approximation Model.

- 1: **Input:** an $n \times n$ symmetric matrix \mathbf{K} and the number of selected columns or target dimension of projection $c (< n)$.
- 2: Sketching: $\mathbf{C} = \mathbf{K}\mathbf{P}$ using an arbitrary $n \times c$ sketching matrix \mathbf{P} (not studied in this work);
- 3: Optional: replace \mathbf{C} by any orthonormal bases of the columns of \mathbf{C} ;
- 4: Compute another $n \times s$ sketching matrix \mathbf{S} , e.g. the leverage score sampling in Algorithm 2;
- 5: Compute the sketches $\mathbf{S}^T \mathbf{C} \in \mathbb{R}^{s \times c}$ and $\mathbf{S}^T \mathbf{K} \mathbf{S} \in \mathbb{R}^{s \times s}$;
- 6: Compute $\mathbf{U}^{\text{fast}} = (\mathbf{S}^T \mathbf{C})^\dagger (\mathbf{S}^T \mathbf{K} \mathbf{S}) (\mathbf{C}^T \mathbf{S})^\dagger \in \mathbb{R}^{c \times c}$;
- 7: **Output:** \mathbf{C} and \mathbf{U}^{fast} such that $\mathbf{K} \approx \mathbf{C}\mathbf{U}^{\text{fast}}\mathbf{C}^T$.

with high probability by sampling $c = \mathcal{O}(k/\epsilon)$ columns of \mathbf{K} to form \mathbf{C} . Let $\mathbf{C} \in \mathbb{R}^{n \times c}$ be formed by this algorithm and $\mathbf{S} \in \mathbb{R}^{n \times s}$ be the leverage score sampling matrix. With $c = \mathcal{O}(k/\epsilon)$ and $s = \mathcal{O}(n^{1/2} k \epsilon^{-3/2})$, the fast model satisfies

$$\|\mathbf{K} - \mathbf{C}\mathbf{U}^{\text{fast}}\mathbf{C}^T\|_F^2 \leq (1 + \epsilon) \|\mathbf{K} - \mathbf{K}_k\|_F^2$$

with high probability.

4.4 Algorithm and Time Complexity

We describe the whole procedure of the fast model in Algorithm 1, where $\mathbf{S} \in \mathbb{R}^{n \times s}$ can be one of the five sketching matrices described in Table 4. Given \mathbf{C} and (the whole or a part of) \mathbf{K} , it takes time

$$\mathcal{O}(s^2 c) + T_{\text{sketch}}$$

to compute \mathbf{U}^{fast} , where T_{sketch} is the time cost of forming the sketches $\mathbf{S}^T \mathbf{C}$ and $\mathbf{S}^T \mathbf{K} \mathbf{S}$ and is described in Table 4. In Table 4 we also show the number of entries of \mathbf{K} that must be observed. From Table 4 we can see that column selection is much more efficient than random projection, and column selection does not require the full observation of \mathbf{K} .

We are particularly interested in the column selection matrix \mathbf{S} corresponding to the row leverage scores of \mathbf{C} . The leverage score sampling described in Algorithm 2 can be efficiently performed. Using the leverage score sampling, it takes time $\mathcal{O}(nc^2/\epsilon)$ (excluding the time of computing $\mathbf{C} = \mathbf{K}\mathbf{P}$) to compute \mathbf{U}^{fast} . For the kernel approximation problem, suppose that we are given n data points of dimension d and that the kernel matrix \mathbf{K} is unknown beforehand. Then it takes $\mathcal{O}(nc^2 d/\epsilon)$ additional time to evaluate the kernel function values.

Algorithm 2 The Leverage Score Sampling Algorithm.

- 1: **Input:** an $n \times c$ matrix \mathbf{C} , an integer s .
- 2: Compute the condensed SVD of \mathbf{C} (by discarding the zero singular values) to obtain the orthonormal bases $\mathbf{U}_G \in \mathbb{R}^{n \times \rho}$, where $\rho = \text{rank}(\mathbf{C}) \leq c$;
- 3: Compute the sampling probabilities $p_i = s\ell_i/\rho$, where $\ell_i = \|\mathbf{e}_i^T \mathbf{U}_G\|_2^2$ is the i -th leverage score;
- 4: Initialize \mathbf{S} to be an matrices of size $n \times 0$;
- 5: **for** $i = 1$ to n **do**
- 6: With probability p_i , add $\sqrt{\frac{c}{s_i}} \mathbf{e}_i$ to be a new column of \mathbf{S} , where \mathbf{e}_i is the i -th standard basis;
- 7: **end for**
- 8: **Output:** \mathbf{S} , whose expected number of columns is s .

4.5 Implementation Details

In practice, the approximation accuracy and numerical stability can be significantly improved by the following techniques and tricks.

If \mathbf{P} and \mathbf{S} are both random sampling matrices, then empirically speaking, enforcing $\mathcal{P} \subset \mathcal{S}$ significantly improves the approximation accuracy. Here \mathcal{P} and \mathcal{S} are the subsets of $[n]$ selected by \mathbf{P} and \mathbf{S} , respectively. Instead of directly sampling s indices from $[n]$ by Algorithm 2, it is better to sample s indices from $[n] \setminus \mathcal{P}$ to form \mathcal{S}' and let $\mathcal{S} = \mathcal{S}' \cup \mathcal{P}$. In this way, $s + c$ columns are sampled. Whether the requirement $\mathcal{P} \subset \mathcal{S}$ improves the accuracy is unknown to us.

Corollary 5 *Theorem 3 still holds when we restrict $\mathcal{P} \subset \mathcal{S}$.*

Proof Let p_1, \dots, p_n be the original sampling probabilities without the restriction $\mathcal{P} \subset \mathcal{S}$. We define the modified sampling probabilities by

$$\tilde{p}_i = \begin{cases} 1 & \text{if } i \in \mathcal{P}; \\ p_i & \text{otherwise.} \end{cases}$$

The column sampling with restriction $\mathcal{P} \subset \mathcal{S}$ amounts to sampling columns according to $\tilde{p}_1, \dots, \tilde{p}_n$. Since $\tilde{p}_i \geq p_i$ for all $i \in [n]$, it follows from Remark 14 that the error bound will not get worse if p_i is replaced by \tilde{p}_i . ■

If \mathbf{S} is the leverage score sampling matrix, we find it better not to scale the entries of \mathbf{S} , although the scaling is necessary for theoretical analysis. According to our observation, the scaling sometimes makes the approximation numerically unstable.

4.6 Additional Properties

When \mathbf{K} is a low-rank matrix, the Nyström method and the prototype model are guaranteed to exactly recover \mathbf{K} (Kumar et al., 2009; Talwalkar and Rostamizadeh, 2010; Wang et al., 2016). We show in the following theorem that the fast model has the same property. We prove the theorem in Appendix E.

Theorem 6 (Exact Recovery) *Let \mathbf{K} be any $n \times n$ symmetric matrix, $\mathbf{P} \in \mathbb{R}^{n \times c}$ and $\mathbf{S} \in \mathbb{R}^{n \times s}$ be any sketching matrices, $\mathbf{C} = \mathbf{K}\mathbf{P}$, and $\mathbf{W} = \mathbf{P}^T \mathbf{C}$. Assume that $\text{rank}(\mathbf{S}^T \mathbf{C}) \geq \text{rank}(\mathbf{W})$. Then $\mathbf{K} = \mathbf{C}(\mathbf{S}^T \mathbf{C})^\dagger (\mathbf{S}^T \mathbf{K}\mathbf{S})(\mathbf{C}^T \mathbf{S})^\dagger \mathbf{C}^T$ if and only if $\text{rank}(\mathbf{K}) = \text{rank}(\mathbf{C})$.*

In the following we establish a lower error bound of the fast model, which implies that to attain the $1 + \epsilon$ Frobenius norm bound relative to the best rank k approximation, the fast model must satisfy

$$c \geq \Omega(k/\epsilon) \quad \text{and} \quad s \geq \Omega(\sqrt{nk/\epsilon}).$$

Notice that the theorem only holds for column selection matrices \mathbf{P} and \mathbf{S} . We prove the theorem in Appendix F.

Theorem 7 (Lower Bound) *Let $\mathbf{P} \in \mathbb{R}^{n \times c}$ and $\mathbf{S} \in \mathbb{R}^{n \times s}$ be any two column selection matrices such that $\mathcal{P} \subset \mathcal{S} \subset [n]$, where \mathcal{P} and \mathcal{S} are the index sets formed by \mathbf{P} and \mathbf{S} , respectively. There exists an $n \times n$ symmetric matrix \mathbf{K} such that*

$$\frac{\|\mathbf{K} - \tilde{\mathbf{K}}_{c,cs}^{\text{fast}}\|_F^2}{\|\mathbf{K} - \mathbf{K}_k\|_F^2} \geq \frac{n-c}{n-k} \left(1 + \frac{2k}{c}\right) + \frac{n-s}{n-k} \frac{k(n-s)}{s^2}, \quad (7)$$

where k is arbitrary positive integer smaller than n , $\mathbf{C} = \mathbf{K}\mathbf{P} \in \mathbb{R}^{n \times c}$, and

$$\tilde{\mathbf{K}}_{c,cs}^{\text{fast}} = \mathbf{C}(\mathbf{S}^T \mathbf{C})^\dagger (\mathbf{S}^T \mathbf{K}\mathbf{S})(\mathbf{C}^T \mathbf{S})^\dagger \mathbf{C}^T$$

is the fast model.

Interestingly, Theorem 7 matches the lower bounds of the Nyström method and the prototype model. When $s = c$, the right-hand side of (7) becomes $\Omega(1 + kn/c^2)$, which is the lower error bound of the Nyström method given by Wang and Zhang (2013). When $s = n$, the right-hand side of (7) becomes $\Omega(1 + k/c)$, which is the lower error bound of the prototype model given by Wang et al. (2016).

5. Extension to CUR Matrix Decomposition

In Section 5.1 we describe the CUR matrix decomposition and establish an improved error bound of CUR in Theorem 8. In Section 5.2 we use sketching to more efficiently compute the \mathbf{U} matrix of CUR. Theorem 8 and Theorem 9 together show that our fast CUR method satisfies $1 + \epsilon$ error bound relative to the best rank k approximation. In Section 5.3 we provide empirical results to intuitively illustrate the effectiveness of our fast CUR. In Section 5.4 we discuss the application of our results beyond the CUR decomposition.

5.1 The CUR Matrix Decomposition

Given any $m \times n$ matrix \mathbf{A} , the CUR matrix decomposition is computed by selecting c columns of \mathbf{A} to form $\mathbf{C} \in \mathbb{R}^{m \times c}$ and r rows of \mathbf{A} to form $\mathbf{R} \in \mathbb{R}^{r \times n}$ and computing the \mathbf{U} matrix such that $\|\mathbf{A} - \mathbf{CUR}\|_F^2$ is small. CUR preserves the sparsity and non-negativity properties of \mathbf{A} ; it is thus more attractive than SVD in certain applications (Mahoney and Drineas, 2009). In addition, with the CUR of \mathbf{A} at hand, the truncated SVD of \mathbf{A} can be very efficiently computed.

A standard way to finding the \mathbf{U} matrix is by minimizing $\|\mathbf{A} - \mathbf{CUR}\|_F^2$ to obtain the optimal \mathbf{U} matrix

$$\mathbf{U}^* = \underset{\mathbf{U}}{\text{argmin}} \|\mathbf{A} - \mathbf{CUR}\|_F^2 = \mathbf{C}^\dagger \mathbf{A} \mathbf{R}^\dagger, \quad (8)$$

which has been used by Stewart (1999); Wang and Zhang (2013); Boutsidis and Woodruff (2014). This approach costs time $\mathcal{O}(mc^2 + nr^2)$ to compute the Moore-Penrose inverse and $\mathcal{O}(m \cdot \min\{c, r\})$ to compute the matrix product. Therefore, even if \mathbf{C} and \mathbf{R} are uniformly sampled from \mathbf{A} , the time cost of CUR is $\mathcal{O}(m \cdot \min\{c, r\})$.

At present the strongest theoretical guarantee is by Boutsidis and Woodruff (2014). They use the adaptive sampling algorithm to select $c = \mathcal{O}(k/\epsilon)$ column and $r = \mathcal{O}(k/\epsilon)$ rows to form \mathbf{C} and \mathbf{R} , respectively, and form $\mathbf{U}^* = \mathbf{C}^\dagger \mathbf{A} \mathbf{R}^\dagger$. The approximation error is bounded by

$$\|\mathbf{A} - \mathbf{C}\mathbf{U}^*\mathbf{R}\|_F^2 \leq (1 + \epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

This result matches the theoretical lower bound up to a constant factor. Therefore this CUR algorithm is near optimal. We establish in Theorem 8 an improved error bound of the adaptive sampling based CUR algorithm, and the constants in the theorem are better than the those in (Boutsidis and Woodruff, 2014). Theorem 8 is obtained by following the idea of Boutsidis and Woodruff (2014) and slightly changing the proof of Wang and Zhang (2013). The proof is in Appendix G.

Theorem 8 *Let \mathbf{A} be any given $m \times n$ matrix, k be any positive integer less than m and n , and $\epsilon \in (0, 1)$ be an arbitrary error parameter. Let $\mathbf{C} \in \mathbb{R}^{m \times c}$ and $\mathbf{R} \in \mathbb{R}^{r \times n}$ be columns and rows of \mathbf{A} selected by the near-optimal column selection algorithm of Boutsidis et al. (2014). When c and r are both greater than $4ke^{-1}(1 + o(1))$, the following inequality holds:*

$$\|\mathbf{A} - \mathbf{C}\mathbf{C}^\dagger \mathbf{A} \mathbf{R}^\dagger \mathbf{R}\|_F^2 \leq (1 + \epsilon)\|\mathbf{A} - \mathbf{A}_k\|_F^2,$$

where the expectation is taken w.r.t. the random column and row selection.

5.2 Fast CUR Decomposition

Analogous to the fast SPSPD matrix approximation model, the CUR decomposition can be sped up while preserving its accuracy. Let $\mathbf{S}_C \in \mathbb{R}^{m \times s_c}$ and $\mathbf{S}_R \in \mathbb{R}^{n \times s_r}$ be any sketching matrices satisfying the approximate matrix multiplication properties. We propose to compute \mathbf{U} more efficiently by

$$\begin{aligned} \tilde{\mathbf{U}} &= \underset{\mathbf{U}}{\operatorname{argmin}} \|\mathbf{S}_C^T \mathbf{A} \mathbf{S}_R - (\mathbf{S}_C^T \mathbf{C}) \mathbf{U} (\mathbf{R} \mathbf{S}_R)\|_F^2 \\ &= \underbrace{(\mathbf{S}_C^T \mathbf{C})^\dagger (\mathbf{S}_C^T \mathbf{A} \mathbf{S}_R)}_{c \times s_c} \underbrace{(\mathbf{R} \mathbf{S}_R)}_{s_r \times s_r}^\dagger, \end{aligned} \quad (9)$$

which costs time

$$\mathcal{O}(s_r r^2 + s_c c^2 + s_c s_r \cdot \min\{c, r\}) + T_{\text{sketch}},$$

where T_{sketch} denotes the time for forming the sketches $\mathbf{S}_C^T \mathbf{A} \mathbf{S}_R$, $\mathbf{S}_C^T \mathbf{C}$, and $\mathbf{R} \mathbf{S}_R$. As for Gaussian projection, SRHT, and count sketch, T_{sketch} are respectively $\mathcal{O}(\min\{s_c, s_r\})$, $\mathcal{O}(m \log(\min\{s_c, s_r\}))$, and $\mathcal{O}(\min\{\mathbf{A}\})$. As for leverage score sampling and uniform sampling, T_{sketch} are respectively $\mathcal{O}(m c^2 + nr^2 + s_c s_r)$ and $\mathcal{O}(s_c s_r)$. Forming the sketches by column selection is more efficient than by random projection.

The following theorem shows that when s_c and s_r are sufficiently large, $\tilde{\mathbf{U}}$ is nearly as good as the best possible \mathbf{U} matrix. In the theorem, leverage score sampling means that \mathbf{S}_C and \mathbf{S}_R sample columns according to the row leverage scores of \mathbf{C} and \mathbf{R}^T , respectively. The proof is in Appendix H.

Theorem 9 *Let $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{C} \in \mathbb{R}^{m \times c}$, $\mathbf{R} \in \mathbb{R}^{r \times n}$ be any fixed matrices with $c \ll n$ and $r \ll m$. Let $q = \min\{m, n\}$ and $\hat{q} = \min\{m/c, n/r\}$. The sketching matrices $\mathbf{S}_C \in \mathbb{R}^{m \times s_c}$ and $\mathbf{S}_R \in \mathbb{R}^{n \times s_r}$ are described in Table 5. Assume that $\epsilon^{-1} = o(q)$ or $\epsilon^{-1} = o(\hat{q})$, as shown in the table. The matrix $\tilde{\mathbf{U}}$ is defined in (9). Then the inequality*

$$\|\mathbf{A} - \mathbf{C}\tilde{\mathbf{U}}\mathbf{R}\|_F^2 \leq (1 + \epsilon) \min\|\mathbf{A} - \mathbf{C}\mathbf{U}\mathbf{R}\|_F^2$$

holds with probability at least 0.7.

Table 5: Leverage score sampling means sampling according to the row leverage scores of \mathbf{C} and the column leverage scores of \mathbf{R} , respectively. For uniform sampling, the parameter $\mu(\mathbf{C})$ is the row coherence of \mathbf{C} and $\nu(\mathbf{R})$ is the column coherence of \mathbf{R} .

Sketching	Order of s_c	Order of s_r	Assumption
Leverage Score Sampling	$c\sqrt{q/\epsilon}$	$r\sqrt{q/\epsilon}$	$\epsilon^{-1} = o(q)$
Uniform Sampling	$\frac{\mu(\mathbf{C})c\sqrt{q/\epsilon}}{\epsilon}$	$\frac{\nu(\mathbf{R})r\sqrt{q/\epsilon}}{\epsilon}$	$\epsilon^{-1} = o(\hat{q})$
Gaussian Projection	$\sqrt{\frac{m}{\epsilon}}(c + \log \frac{m}{\epsilon})$	$\sqrt{\frac{n}{\epsilon}}(r + \log \frac{n}{\epsilon})$	$\epsilon^{-1} = o(q)$
SRHT	$\sqrt{\frac{m}{\epsilon}}(c + \log \frac{m}{\epsilon}) \log(m)$	$\sqrt{\frac{n}{\epsilon}}(r + \log \frac{n}{\epsilon}) \log(n)$	$\epsilon^{-1} = o(\hat{q})$
Count Sketch	$c\sqrt{q/\epsilon}$	$r\sqrt{q/\epsilon}$	$\epsilon^{-1} = o(q)$

As for leverage score sampling, uniform sampling, and count sketch, the sketch sizes $s_c = \mathcal{O}(c\sqrt{q/\epsilon})$ and $s_r = \mathcal{O}(r\sqrt{q/\epsilon})$ suffice, where $q = \min\{m, n\}$. As for Gaussian projection and SRHT, much smaller sketch sizes are required: $s_c = \tilde{\mathcal{O}}(\sqrt{mc/\epsilon})$ and $s_r = \tilde{\mathcal{O}}(\sqrt{nr/\epsilon})$ suffice. However, these random projection methods are inefficient choices in this application and only have theoretical interest. Only column sampling methods have linear time complexities. If \mathbf{S}_C and \mathbf{S}_R are leverage score sampling matrices (according to the row leverage scores of \mathbf{C} and \mathbf{R}^T , respectively), it follows from Theorem 9 that $\tilde{\mathbf{U}}$ with $1 + \epsilon$ bound can be computed in time

$$\mathcal{O}(s_r r^2 + s_c c^2 + s_c s_r \cdot \min\{c, r\}) + T_{\text{sketch}} = \mathcal{O}(\epsilon r \epsilon^{-1} \cdot \min\{m, n\} \cdot \min\{c, r\}),$$

which is linear in $\mathcal{O}(\min\{m, n\})$.

5.3 Empirical Comparisons

To intuitively demonstrate the effectiveness of our method, we conduct a simple experiment on a 1920 \times 1168 natural image obtained from the internet. We first uniformly sample $c = 100$ columns to form \mathbf{C} and $r = 100$ rows to form \mathbf{R} , and then compute the \mathbf{U} matrix by varying s_c and s_r . We show the image $\mathbf{A} = \mathbf{C}\mathbf{U}\mathbf{R}$ in Figure 2.

Figure 2(b) is obtained by computing the \mathbf{U} matrix according to (8), which is the best possible result when \mathbf{C} and \mathbf{R} are fixed. The \mathbf{U} matrix of Figure 2(c) is computed according to Drines et al. (2008):

$$\mathbf{U} = (\mathbf{P}_R^T \mathbf{A} \mathbf{P}_C)^\dagger,$$

where \mathbf{P}_C and \mathbf{P}_R are column selection matrices such that $\mathbf{C} = \mathbf{A}\mathbf{P}_C$ and $\mathbf{R} = \mathbf{P}_R^T\mathbf{A}$. This is equivalently to (9) by setting $\mathbf{S}_C = \mathbf{P}_R$ and $\mathbf{S}_R = \mathbf{P}_C$. Obviously, this setting leads to very poor quality. In Figures 2(c) and (d) the sketching matrices \mathbf{S}_C and \mathbf{S}_R are uniform sampling matrices. The figures show that when s_c and s_r are moderately greater than r and c , respectively, the approximation quality is significantly improved. Especially, when $s_c = 4r$ and $s_r = 4c$, the approximation quality is nearly as good as using the optimal \mathbf{U} matrix defined in (8).

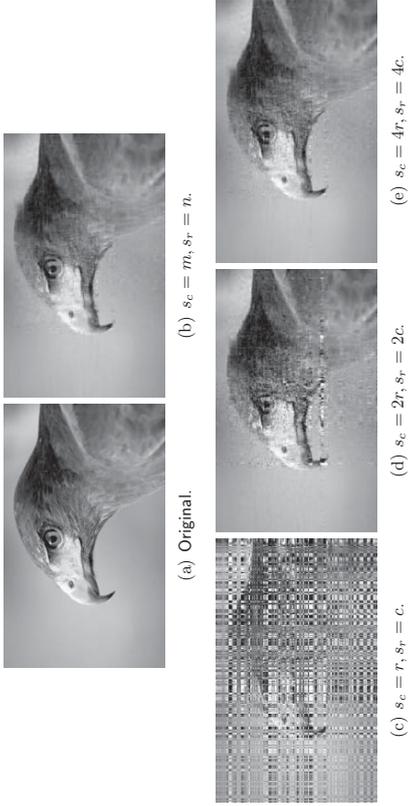


Figure 2: (a): the original 1920×1168 image. (b) to (e): CUR decomposition with $c = r = 100$ and different settings of s_c and s_r .

5.4 Discussions

We note that we are not the first to use row and column sampling to solve the CUR problem more efficiently, though we are the first to provide rigorous error analysis. Previous work has exploited similar ideas as heuristics to speed up computation and to avoid visiting every entry of \mathbf{A} . For example, the MEKA method (Si et al., 2014a) partitions the kernel matrix \mathbf{K} into b^2 blocks $\mathbf{K}^{(i,j)}$ ($i = 1, \dots, b$ and $j = 1, \dots, b$), and requires solving

$$\mathbf{L}^{(i,j)} = \operatorname{argmin}_{\mathbf{L}} \|\mathbf{W}^{(i)}\mathbf{L}\mathbf{W}^{(j)T} - \mathbf{K}^{(i,j)}\|_F^2$$

for all $i \in [b]$, $j \in [b]$, and $i \neq j$. Since $\mathbf{W}^{(i)}$ and $\mathbf{W}^{(j)}$ have much more rows than columns, Si et al. (2014a) proposed to approximately solve the linear system by uniformly sampling rows from $\mathbf{W}^{(i)}$ and $\mathbf{K}^{(i,i)}$ and columns from $(\mathbf{W}^{(j)})^T$ and $\mathbf{K}^{(i,j)}$, and they noted that this heuristic works pretty well. The basic ideas of our fast CUR and their MEKA are the same; their experiments demonstrate the effectiveness and efficiency of this approach, and

Table 6: A summary of the datasets for kernel approximation.

Dataset	Letters	PenDigit	Cpustall	Mushrooms	WineQuality
#Instance	15,000	10,992	8,192	8,124	4,898
#Attribute	16	16	12	112	12
σ (when $\eta = 0.90$)	0.400	0.101	0.075	1.141	0.314
σ (when $\eta = 0.99$)	0.590	0.178	0.180	1.960	0.486

our analysis answers why this approach is correct. This also implies that our algorithms and analysis may have broad applications and impacts beyond the CUR decomposition and SPSPD matrix approximation.

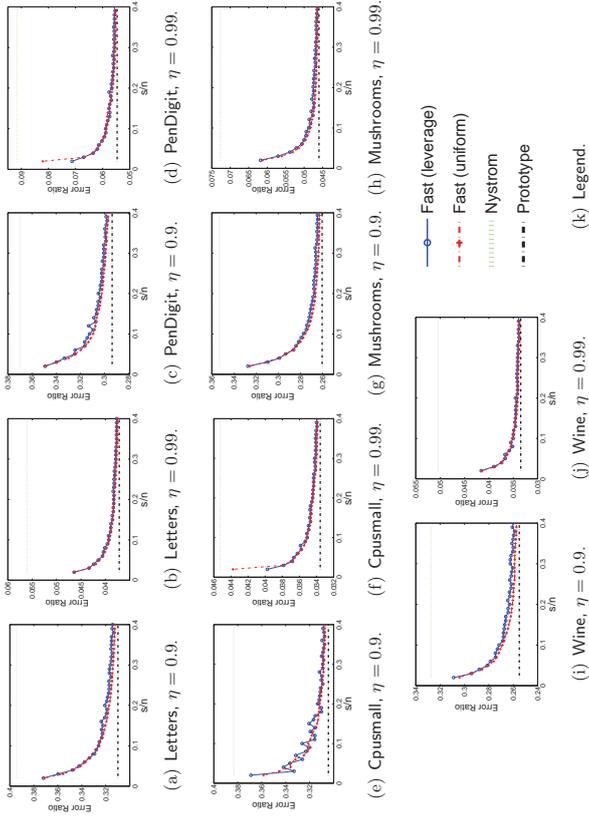


Figure 3: The plot of $\frac{\|\mathbf{K} - \mathbf{CUC}^T\|_F^2}{\|\mathbf{K}\|_F^2}$ against the approximation error $\|\mathbf{K} - \mathbf{CUC}^T\|_F^2 / \|\mathbf{K}\|_F^2$, where \mathbf{C} contains $c = \lfloor n/100 \rfloor$ column of $\mathbf{K} \in \mathbb{R}^{n \times n}$ selected by uniform sampling.

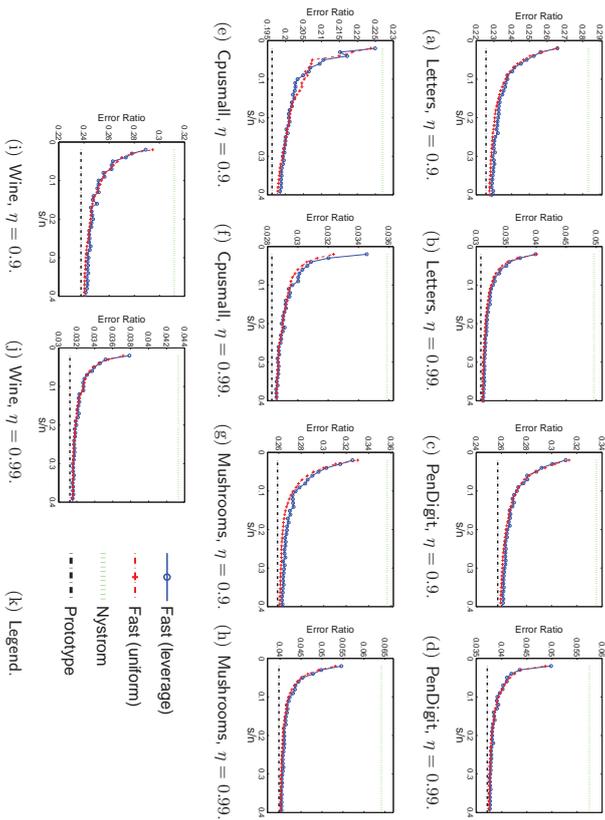


Figure 4: The plot of $\frac{\sigma}{n}$ against the approximation error $\|\mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^T\|_F^2 / \|\mathbf{K}\|_F^2$, where \mathbf{C} contains $c = \lceil n/100 \rceil$ column of $\mathbf{K} \in \mathbb{R}^{n \times n}$ selected by the uniform+adaptive² sampling algorithm (Wang et al., 2016).

6. Experiments

In this section we conduct several sets of illustrative experiments to show the effect of the \mathbf{U} matrix. We compare the three methods with different settings of c and s . We do not compare with other kernel approximation methods for the reasons stated in Section 3.2.2.

6.1 Setup

Let $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_n]$ be the $d \times n$ data matrix, and \mathbf{K} be the RBF kernel matrix with each entry computed by $K_{ij} = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2})$ where σ is the scaling parameter.

When comparing the kernel approximation error $\|\mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^T\|_F^2$, we set the scaling parameter σ in the following way. We let $k = \lceil n/100 \rceil$ and define

$$\eta = \frac{\|\mathbf{K}_k\|_F^2}{\|\mathbf{K}\|_F^2} = \frac{\sum_{i=1}^k \sigma^2(\mathbf{K})}{\sum_{i=1}^n \sigma^2(\mathbf{K})},$$

which indicate the importance of the top one percent singular values of \mathbf{K} . In general η grows with σ . We set σ such that $\eta = 0.9$ or 0.99 .

All the methods are implemented in MATLAB and run on a laptop with Intel i5 2.5GHz CPU and 8GB RAM. To compare the running time, we set MATLAB in the single thread mode.

6.2 Kernel Approximation Accuracy

We conduct experiments on several datasets available at the LIBSVM site. The datasets are summarized in Table 6. In this set of experiments, we study the effect of the \mathbf{U} matrices. We use two methods to form $\mathbf{C} \in \mathbb{R}^{n \times c}$: uniform sampling and the uniform+adaptive² sampling (Wang et al., 2016); we fix $c = \lceil n/100 \rceil$. For our fast model, we use two kinds of sketching matrices $\mathbf{S} \in \mathbb{R}^{n \times s}$: uniform sampling and leverage score sampling; we vary s from $2c$ to $40c$. We plot $\frac{\sigma}{n}$ against the approximation error $\|\mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^T\|_F^2 / \|\mathbf{K}\|_F^2$ in Figures 3 and 4. The Nystrom method and the prototype model are included for comparison.

Figures 3 and 4 show that the fast SPSSD matrix approximation model is significantly better than the Nystrom method when s is slightly larger than c , e.g., $s = 2c$. Recall that the prototype model is a special case of the fast model where $s = n$. We can see that the fast model is nearly as accurate as the prototype model when s is far smaller than n , e.g., $s = 0.2n$.

The results also show that using uniform sampling and leverage score sampling to generate \mathbf{S} does not make much difference. Thus, in practice, one can simply compute \mathbf{S} by uniform sampling.

By comparing the results in Figures 3 and 4, we can see that computing \mathbf{C} by uniform+adaptive² sampling is substantially better than uniform sampling. However, adaptive sampling requires the full observation of \mathbf{K} ; thus with uniform+adaptive² sampling, our fast model does not have much advantage over the prototype model in terms of time efficiency. Our main focus of this work is the \mathbf{U} matrix, so in the rest of the experiments we simply use uniform sampling to compute \mathbf{C} .

6.3 Approximate Kernel Principal Component Analysis

We apply the three methods to approximately compute kernel principal component analysis (KPCA), and contrast with the exact solution. The experiment setting follows Zhang and Kwok (2010). We fix k and vary c . For our fast model, we set $s = 2c$, $4c$, or $8c$. Since computing \mathbf{S} by uniform sampling or leverage score sampling yields the same empirical performance, we use only uniform sampling. Let $\mathbf{C}\mathbf{U}\mathbf{C}^T$ be the low-rank approximation formed by the three methods. Let $\tilde{\mathbf{V}}\tilde{\mathbf{A}}\tilde{\mathbf{V}}^T$ be the k -eigenvalue decomposition of $\mathbf{C}\mathbf{U}\mathbf{C}^T$.

6.3.1 QUALITY OF THE APPROXIMATE EIGENVECTORS

Let $\mathbf{U}_{k,k} \in \mathbb{R}^{n \times k}$ contain the top k eigenvectors of \mathbf{K} . In the first set of experiments, we measure the distance between $\mathbf{U}_{k,k}$ and the approximate eigenvectors $\tilde{\mathbf{V}}$ by

$$\text{Misalignment} = \frac{1}{k} \|\mathbf{U}_{k,k} - \tilde{\mathbf{V}}\tilde{\mathbf{V}}^T \mathbf{U}_{k,k}\|_F^2 \in [0, 1]. \quad (10)$$

Small misalignment indicates high approximation quality. We fix $k = 3$.

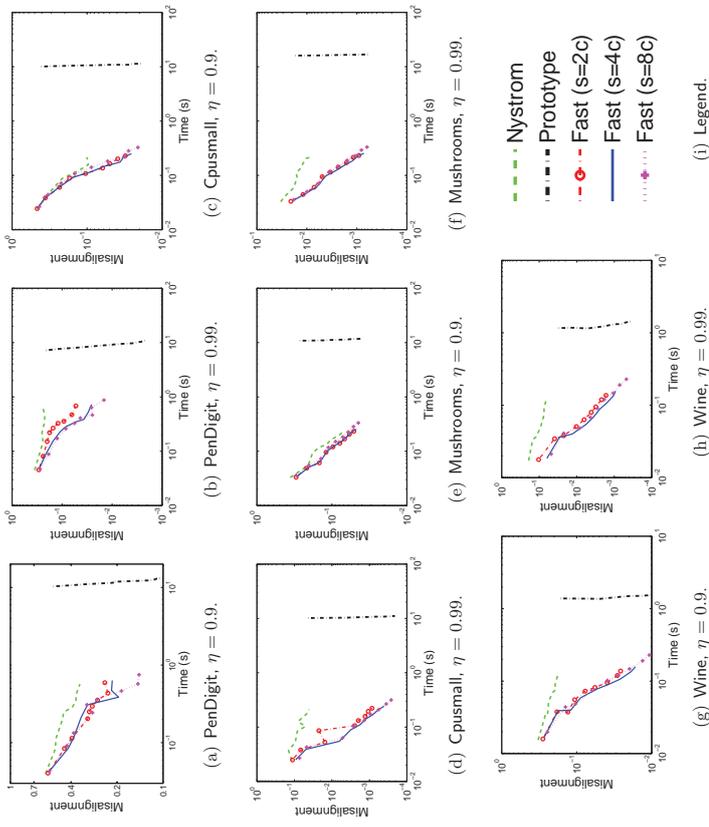


Figure 5: The plot of (log-scale) elapsed time against the (log-scale) misalignment defined in (10).

We conduct experiments on the datasets summarized in Table 6. We record the elapsed time of the entire procedure—computing (part of) the kernel matrix, computing \mathbf{C} and \mathbf{U} by the kernel approximation methods, computing the k -eigenvalue decomposition of \mathbf{CUC}^T . We plot the elapsed time against the misalignment defined in Figure 5. Results on the Letters dataset are not reported because the exact k -eigenvalue decomposition on MATLAB ran out of memory, making it impossible to calculate the misalignment.

At the end of Section 3.2.1 we have mentioned the importance of memory cost of the kernel approximation methods and that all three compared methods cost $\mathcal{O}(nc + nd)$ memory. Since n and d are fixed, we plot c against the misalignment in Figure 6 to show the memory efficiency.

The results show that using the same amount of time or memory, the misalignment incurred by the Nystrom method is usually tens of times higher than our fast model. The

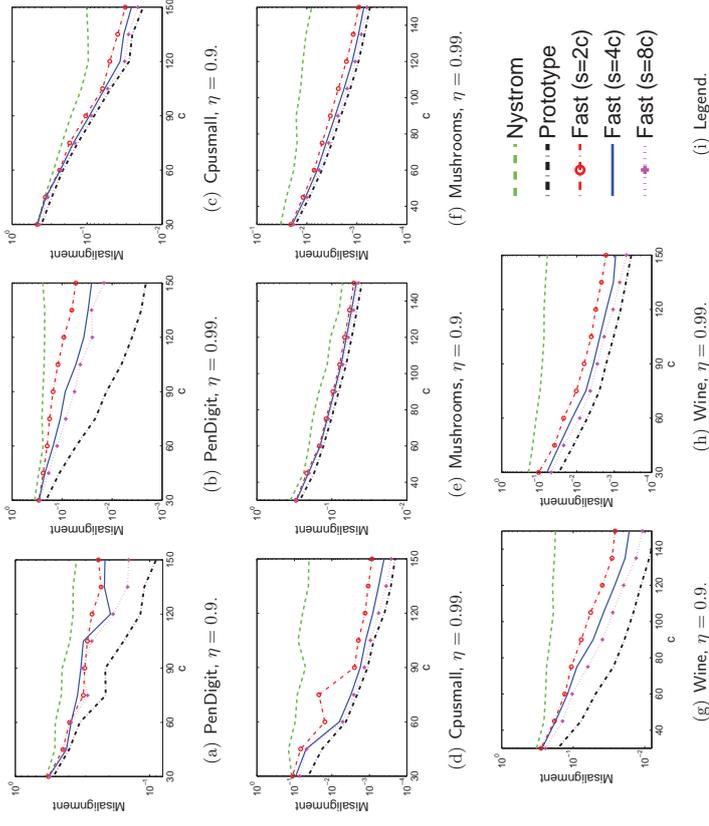
Figure 6: The plot of c against the (log-scale) misalignment defined in (10).

Table 7: A summary of the datasets for clustering and classification.

Dataset	MNIST	PenDigit	USPS	Mushrooms	Gisette	DNA
#Instance	60,000	10,992	9,298	8,124	7,000	2,000
#Attribute	780	16	256	112	5,000	180
#Class	10	10	10	2	2	3
Scaling Parameter σ	10	0.7	15	3	50	4

experiment also shows that with fixed c , the fast model is nearly as accurate as the prototype model when $s = 8c \ll n$.

6.3.2 QUALITY OF THE GENERALIZATION

In the second set of experiments, we test the generalization performance of the kernel approximation methods on classification tasks. The classification datasets are described in

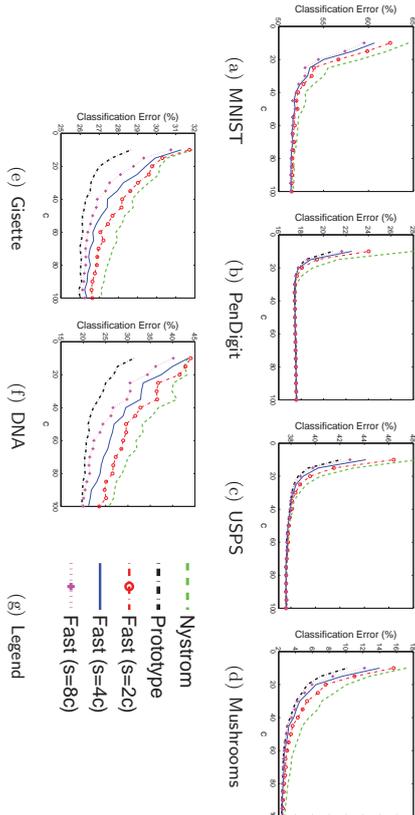
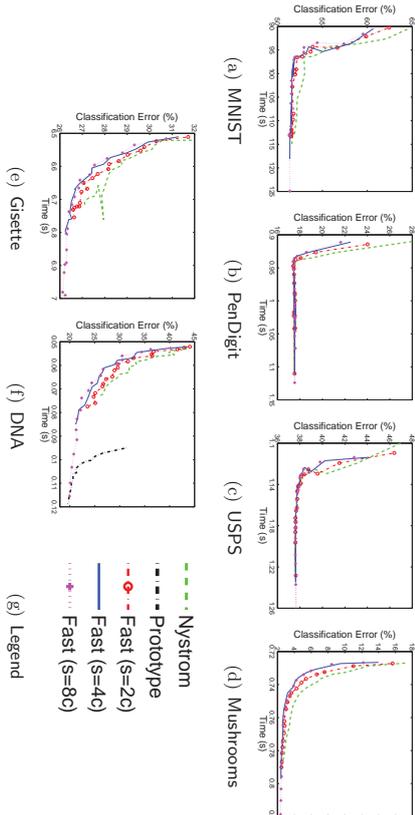
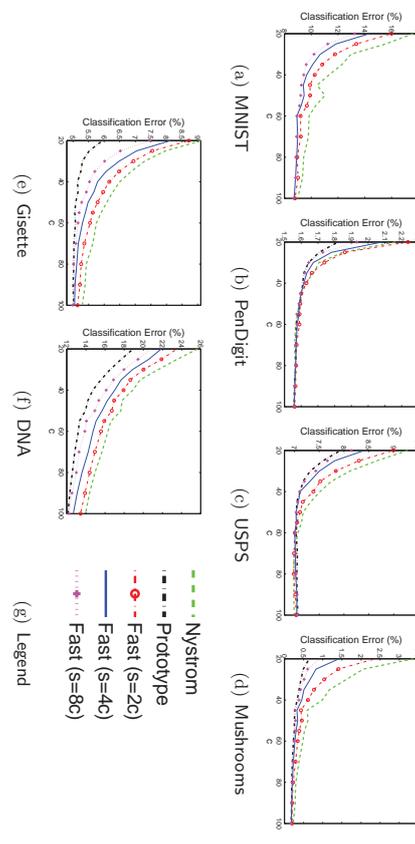
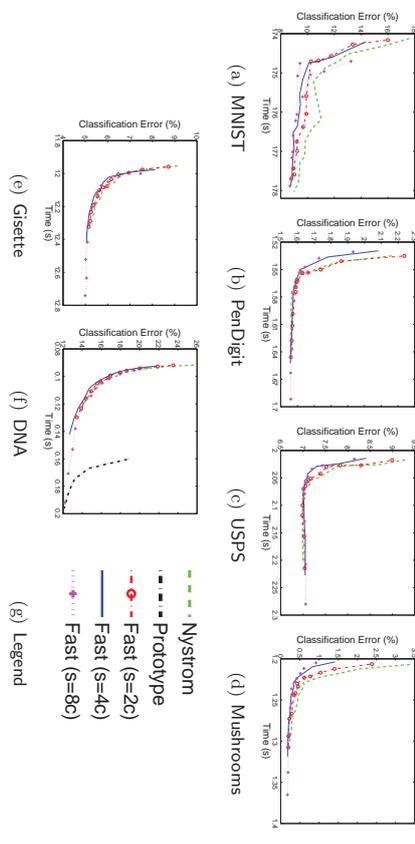
Figure 7: The plot of c against the classification error. Here $k = 3$.Figure 8: The plot of elapsed time against the classification error. Here $k = 3$.

Table 7. For each dataset, we randomly sample $n_1 = 50\%$ data points for training and the rest 50% for test. In this set of experiments, we set $k = 3$ and $k = 10$.

We let $\mathbf{K} \in \mathbb{R}^{n_1 \times n_1}$ be the RBF kernel matrix of the training data and $\mathbf{k}(\mathbf{x}) \in \mathbb{R}^{n_1}$ be defined by $[\mathbf{k}(\mathbf{x})]_i = \exp(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2\sigma^2})$, where \mathbf{x}_i is the i -th training data point. In the training step, we approximately compute the top k eigenvalues and eigenvectors, and denote

Figure 9: The plot of c against the classification error. Here $k = 10$.Figure 10: The plot of elapsed time against the classification error. Here $k = 10$.

$\tilde{\mathbf{A}} \in \mathbb{R}^{k \times k}$ and $\tilde{\mathbf{V}} \in \mathbb{R}^{n_1 \times k}$. The feature vector (extracted by KPCA) of the i -th training data point is the i -th column of $\tilde{\mathbf{A}}^{0.5} \tilde{\mathbf{V}}^T$. In the test step, the feature vector of test data \mathbf{x} is $\tilde{\mathbf{A}}^{-0.5} \tilde{\mathbf{V}}^T \mathbf{k}(\mathbf{x})$. Then we put the training labels and training and test features into the MATLAB K-nearest-neighbor classifier `knnclassify` to classify the test data. We fix the number of nearest neighbors to be 10. The scaling parameters of each dataset are listed in

Table 7. Since the kernel approximation methods are randomized, we repeat the training and test procedure 20 times and record the average elapsed time and average classification error.

We plot c against the classification error in Figures 7 and 9, and plot the elapsed time (excluding the time cost of KNN) against the classification error in Figures 8 and 10. Using the same amount of memory, the fast model is significantly better than the Nyström method, especially when c is small. Using the same amount of time, the fast model outperforms the Nyström method by one to two percent of classification error in many cases, and it is at least as good as the Nyström method in the rest cases. This set of experiments also indicate that the fast model with $s = 4c$ or $8c$ has the best empirical performance.

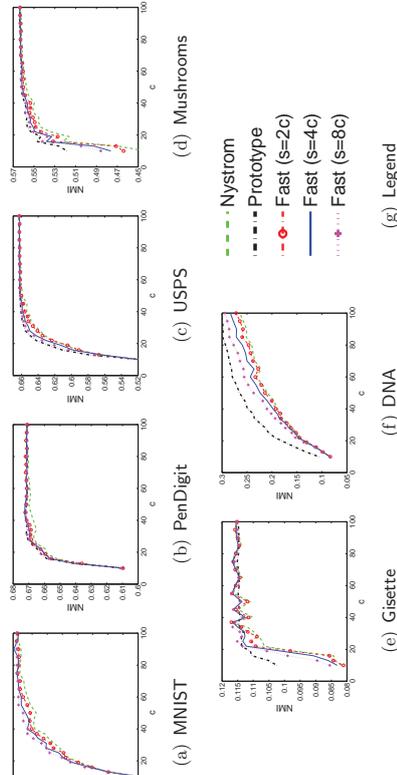


Figure 11: The plot of c against NMI.

6.4 Approximate Spectral Clustering

Following the work of Fowlkes et al. (2004), we evaluate the performance of the kernel approximation methods on the spectral clustering task. We conduct experiments on the datasets summarized in Table 7.

We describe the approximate spectral clustering in the following. The target is to cluster n data points into k classes. We use the RBF kernel matrix \mathbf{K} as the weigh matrix and let $\mathbf{CUC}^T \approx \mathbf{K}$ be the low-rank approximation. The degree matrix $\mathbf{D} = \text{diag}(\mathbf{d})$ is a diagonal matrix with $\mathbf{d} = \mathbf{CUC}^T \mathbf{1}_n$, and the normalized graph Laplacian is $\mathbf{L} = \mathbf{I}_n - \mathbf{D}^{-1/2}(\mathbf{CUC}^T)\mathbf{D}^{-1/2}$. The bottom k eigenvectors of \mathbf{L} are the top k eigenvectors of

$$\underbrace{(\mathbf{D}^{-1/2}\mathbf{C})}_{n \times c} \underbrace{\mathbf{U}}_{c \times c} \underbrace{(\mathbf{D}^{-1/2}\mathbf{C}^T)}_{c \times n},$$

which can be efficiently computed according to Appendix A. We denote the top k eigenvectors by $\tilde{\mathbf{V}} \in \mathbb{R}^{n \times k}$. We normalize the rows of $\tilde{\mathbf{V}}$ and take the normalized rows

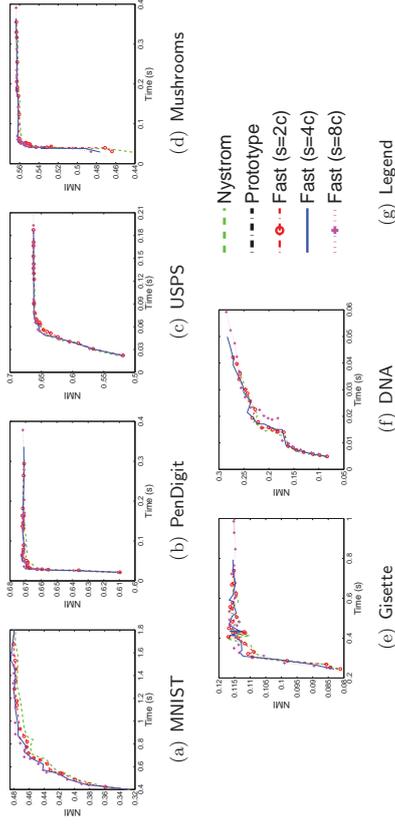


Figure 12: The plot of elapsed time against NMI.

of $\tilde{\mathbf{V}}$ as the input of the k -means clustering. Since the matrix approximation methods are randomized, we repeat this procedure 20 times and record the average elapsed time and the average normalized mutual information (NMI)³ of clustering.

We plot c against NMI in Figure 11 and the elapsed time (excluding the time cost of k -means) against NMI in Figure 12. Figure 11 shows that using the same amount of memory, the performance of the fast model is better than the Nyström method. Using the same amount of time, the fast model and the Nyström method have almost the same performance, and they are both better than the prototype model.

7. Concluding Remarks

In this paper we have studied the fast SPSPD matrix approximation model for approximating large-scale SPSPD matrix. We have shown that our fast model potentially costs time linear in n , while it is nearly as accurate as the best possible approximation. The fast model is theoretically better than the Nyström method and the prototype model because the latter two methods cost time quadratic in n to attain the same theoretical guarantee. Experiments show that our fast model is nearly as accurate as the prototype model and nearly as efficient as the Nyström method.

The technique of the fast model can be straightforwardly applied to speed up the CUR matrix decomposition, and theoretical analysis shows that the accuracy is almost unaffected. In this way, for any $m \times n$ large-scale matrix, the time cost of computing the \mathbf{U} matrix drops from $\mathcal{O}(mn)$ to $\mathcal{O}(\min\{m, n\})$.

3. NMI is a standard metric of clustering. NMI is between 0 and 1. Big NMI indicates good clustering performance.

Acknowledgements

We thank the anonymous reviewer for their helpful feedbacks. Shusen Wang acknowledges the support of Cray Inc., the Defense Advanced Research Projects Agency, the National Science Foundation, and the Baifu Scholarship. Zhifan Zhang acknowledges the support of National Natural Science Foundation of China (No. 61572017) and MISRA Collaborative Research Grant awards. Tong Zhang acknowledges NSF IIS-1250985, NSF IIS-1407939, and NIH R01AI116744.

Appendix A. Approximately Solving the Eigenvalue Decomposition and Matrix Inversion

In this section we show how to use the SPSPD matrix approximation methods to speed up eigenvalue decomposition and linear system. The two lemmas are well known results. We show them here for the sake of self-containing.

Lemma 10 (Approximate Eigenvalue Decomposition) *Given $\mathbf{C} \in \mathbb{R}^{n \times c}$ and $\mathbf{U} \in \mathbb{R}^{c \times c}$. Then the eigenvalue decomposition of $\tilde{\mathbf{K}} = \mathbf{C}\mathbf{U}\mathbf{C}^T$ can be computed in time $\mathcal{O}(nc^2)$.*

Proof It cost $\mathcal{O}(nc^2)$ time to compute the SVD

$$\mathbf{C} = \underbrace{\mathbf{U}_c}_{n \times c} \underbrace{\Sigma_c}_{c \times c} \underbrace{\mathbf{V}_c^T}_{c \times c}$$

and $\mathcal{O}(c^3)$ time to compute $\mathbf{Z} = (\Sigma_c \mathbf{V}_c^T) \mathbf{U} (\Sigma_c \mathbf{V}_c^T)^T \in \mathbb{R}^{c \times c}$. It costs $\mathcal{O}(c^3)$ time to compute the eigenvalue decomposition $\mathbf{Z} = \mathbf{V}_z \mathbf{A}_z \mathbf{V}_z^T$. Combining the results above, we obtain

$$\begin{aligned} \mathbf{C}\mathbf{U}\mathbf{C}^T &= (\mathbf{U}_c \Sigma_c \mathbf{V}_c^T) \mathbf{U} (\mathbf{U}_c \Sigma_c \mathbf{V}_c^T)^T \\ &= \mathbf{U}_c \mathbf{Z} \mathbf{U}_c^T = (\mathbf{U}_c \mathbf{V}_z \mathbf{V}_z^T) \mathbf{A}_z (\mathbf{U}_c \mathbf{V}_z)^T. \end{aligned}$$

It then cost time $\mathcal{O}(nc^2)$ to compute the matrix product $\mathbf{U}_c \mathbf{V}_z$. Since $(\mathbf{U}_c \mathbf{V}_z)$ has orthonormal columns and \mathbf{A}_z is diagonal matrix, the eigenvalue decomposition of $\mathbf{C}\mathbf{U}\mathbf{C}^T$ is solved. The total time cost is $\mathcal{O}(nc^2) + \mathcal{O}(c^3) = \mathcal{O}(nc^2)$. \blacksquare

Lemma 11 (Approximately Solving Matrix Inversion) *Given $\mathbf{C} \in \mathbb{R}^{n \times c}$, SPSPD matrix $\mathbf{U} \in \mathbb{R}^{c \times c}$, vector $\mathbf{y} \in \mathbb{R}^n$, and arbitrary positive real number α . Then it costs time $\mathcal{O}(nc^2)$ to solve the $n \times n$ linear system $(\mathbf{C}\mathbf{U}\mathbf{C}^T + \alpha \mathbf{I}_n) \mathbf{w} = \mathbf{y}$ to obtain $\mathbf{w} \in \mathbb{R}^n$.*

In addition, if the SVD of \mathbf{C} is given, then it takes only $\mathcal{O}(c^3 + nc)$ time to solve the linear system.

Proof Since the matrix $(\mathbf{C}\mathbf{U}\mathbf{C}^T + \alpha \mathbf{I}_n)$ is nonsingular when $\alpha > 0$ and \mathbf{U} is SPSPD, the solution is $\mathbf{w}^* = (\mathbf{C}\mathbf{U}\mathbf{C}^T + \alpha \mathbf{I}_n)^{-1} \mathbf{y}$. Instead of directly computing the matrix inversion, we can expand the matrix inversion by the Sherman-Morrison-Woodbury matrix identity and obtain

$$(\mathbf{C}\mathbf{U}\mathbf{C}^T + \alpha \mathbf{I}_n)^{-1} = \alpha^{-1} \mathbf{I}_n - \alpha^{-1} \mathbf{C} (\alpha \mathbf{U}^{-1} + \mathbf{C}^T \mathbf{C})^{-1} \mathbf{C}^T.$$

Thus the solution to the linear system is

$$\mathbf{w}^* = \alpha^{-1} \mathbf{y} - \underbrace{\alpha^{-1} \mathbf{C}}_{n \times c} \underbrace{(\alpha \mathbf{U}^{-1} + \mathbf{C}^T \mathbf{C})^{-1}}_{c \times c} \underbrace{\mathbf{C}^T}_{c \times n} \mathbf{y}.$$

Suppose we are given only \mathbf{C} and \mathbf{U} . The matrix multiplication $\mathbf{C}^T \mathbf{C}$ costs time $\mathcal{O}(nc^2)$, the matrix inversions cost time $\mathcal{O}(c^3)$, and multiplying matrix with vector costs time $\mathcal{O}(nc)$. Thus the total time cost is $\mathcal{O}(nc^2) + \mathcal{O}(c^3) + \mathcal{O}(nc) = \mathcal{O}(nc^2)$.

Suppose we are given \mathbf{U} and the SVD $\mathbf{C} = \mathbf{U}_c \Sigma_c \mathbf{V}_c^T$. The matrix product

$$\mathbf{C}^T \mathbf{C} = \mathbf{V}_c \Sigma_c \mathbf{U}_c^T \mathbf{U}_c \Sigma_c \mathbf{V}_c = \mathbf{V}_c \Sigma_c^2 \mathbf{V}_c$$

can be computed in time $\mathcal{O}(c^3)$. Thus the total time cost is merely $\mathcal{O}(c^3 + nc)$. \blacksquare

Appendix B. Proof of Theorem 1

The prototype model trivially satisfies requirement R1 with $\epsilon = 0$. However, it violates requirement R2 because computing the \mathbf{U} matrix by solving $\min_{\mathbf{U}} \|\mathbf{K} - \mathbf{C}\mathbf{U}\mathbf{C}^T\|_F^2$ costs time $\mathcal{O}(n^2c)$.

For the Nystrom method, we provide such an adversarial case that assumptions A1 and A2 can both be satisfied and that requirements R1 and R2 cannot hold simultaneously. The adversarial case is the block diagonal matrix

$$\mathbf{K} = \text{diag}(\underbrace{\mathbf{B}, \dots, \mathbf{B}}_{k \text{ blocks}}),$$

where

$$\mathbf{B} = (1-a) \mathbf{I}_p + \alpha \mathbf{1}_p \mathbf{1}_p^T, \quad a < 1, \quad \text{and } p = \frac{n}{k},$$

and let $a \rightarrow 1$. Wang et al. (2016) showed that sampling $c = 3k\gamma^{-1}(1 + o(1))$ columns of \mathbf{K} to form \mathbf{C} makes assumptions A1 and A2 in Question 1 be satisfied. This indicates that \mathbf{C} is a good sketch of \mathbf{K} . The problem is caused by the way the \mathbf{U}^{NYS} matrix is computed. Wang and Zhang (2013, Theorem 12) showed that to make requirement R1 in Question 1 satisfied, c must be greater than $\Omega(\sqrt{nk}/(\epsilon + \gamma))$. Thus it takes time $\mathcal{O}(nc^2) = \Omega(n^2k/(\epsilon + \gamma))$ to compute the rank- k eigenvalue decomposition of $\mathbf{C}\mathbf{U}^{\text{NYS}}\mathbf{C}^T$ or the linear system $(\mathbf{C}\mathbf{U}^{\text{NYS}}\mathbf{C}^T + \alpha \mathbf{I}_n) \mathbf{w} = \mathbf{y}$. Thus, requirement R2 is violated.

Appendix C. Proof of Lemma 2

Lemma 2 is a simplified version of Lemma 12. We prove Lemma 12 in the subsequent subsections. In the lemma, leverage score sampling means that the sampling probabilities are proportional to the row leverage scores of $\mathbf{U} \in \mathbb{R}^{n \times k}$. For uniform sampling, $\mu(\mathbf{U})$ is the row coherence of \mathbf{U} .

Lemma 12 Let $\mathbf{U} \in \mathbb{R}^{n \times k}$ be any fixed matrix with orthonormal columns and $\mathbf{B} \in \mathbb{R}^{n \times d}$ be any fixed matrix. Let $\mathbf{S} \in \mathbb{R}^{n \times s}$ be any sketching matrix described in Table 8. Then

$$\begin{aligned} \mathbb{P}\left\{\|\mathbf{U}^T \mathbf{S} \mathbf{S}^T \mathbf{U} - \mathbf{I}_k\|_2 \geq \eta\right\} &\leq \delta_1 && \text{(Property 1),} \\ \mathbb{P}\left\{\|\mathbf{U}^T \mathbf{B} - \mathbf{U}^T \mathbf{S} \mathbf{S}^T \mathbf{B}\|_F^2 \geq \epsilon \|\mathbf{B}\|_F^2\right\} &\leq \delta_2 && \text{(Property 2),} \\ \mathbb{P}\left\{\|\mathbf{U}^T \mathbf{B} - \mathbf{U}^T \mathbf{S} \mathbf{S}^T \mathbf{B}\|_2^2 \geq \epsilon' \|\mathbf{B}\|_2^2 + \frac{\epsilon'}{k} \|\mathbf{B}\|_F^2\right\} &\leq \delta_3 && \text{(Property 3).} \end{aligned}$$

Table 8: The sketch size s for satisfying the three properties. For SRHT, we define $\lambda = (1 + \sqrt{8k-1} \log(100n))^2$ and $\lambda' = (1 + \sqrt{4k-1} \log \frac{nd}{k\delta_1})^2$.

Sketching	Property 1	Property 2	Property 3
Leverage Sampling	$k \frac{6+2\eta}{3\eta^2} \log \frac{k}{\delta_1}$	$\frac{k}{\epsilon \delta_2}$	—
Uniform Sampling	$\mu(\mathbf{U}) k \frac{6+2\eta}{3\eta^2} \log \frac{k}{\delta_1}$	$\frac{\mu(\mathbf{U}) k}{\epsilon \delta_2}$	—
SRHT	$\lambda k \frac{6+2\eta}{3\eta^2} \log \frac{k}{\delta_1 - 0.01}$	$\frac{\lambda k}{\epsilon(\delta_2 - 0.01)}$	$\lambda k \frac{24+4\sqrt{2\lambda'}}{3\epsilon} \log \frac{2d}{\delta_3 - 0.01}$
Gaussian Projection	$\frac{9(\sqrt{k} + \sqrt{2 \log(2/\delta_1)})^2}{\eta^2}$	$\frac{18k}{\epsilon \delta_2}$	$\frac{36k}{\epsilon} \left(1 + \sqrt{k-1} \log \frac{2d}{k\delta_3}\right)^2$
Count Sketch	$\frac{k^2 + k}{\delta_1 \eta^2}$	$\frac{2k}{\epsilon \delta_2}$	—

C.1 Column Selection

In this subsection we prove Property 1 and Property 2 of leverage score sampling and uniform sampling. We cite the following lemma from (Wang et al., 2016); the lemma was firstly proved by the work Drineas et al. (2008); Gittens (2011); Woodruff (2014).

Lemma 13 Let $\mathbf{U} \in \mathbb{R}^{n \times k}$ be any fixed matrix with orthonormal columns. The column selection matrix $\mathbf{S} \in \mathbb{R}^{n \times s}$ samples s columns according to arbitrary probabilities p_1, p_2, \dots, p_n . Assume $\alpha \geq k$ and

$$\max_{i \in [n]} \frac{\|\mathbf{u}_i\|_2^2}{p_i} \leq \alpha.$$

If $s \geq \alpha \frac{6+2\eta}{3\eta^2} \log(k/\delta_1)$, it holds that

$$\mathbb{P}\left\{\|\mathbf{I}_k - \mathbf{U}^T \mathbf{S} \mathbf{S}^T \mathbf{U}\|_2 \geq \eta\right\} \leq \delta_1.$$

If $s \geq \frac{\alpha}{\epsilon \delta_2}$, it holds that

$$\mathbb{P}\left\{\|\mathbf{U} \mathbf{B} - \mathbf{U}^T \mathbf{S} \mathbf{S}^T \mathbf{B}\|_F^2 \geq \epsilon \|\mathbf{B}\|_F^2\right\} \leq \delta_2.$$

Leverage score sampling satisfies $\max_{i \in [n]} \frac{\|\mathbf{u}_i\|_2^2}{p_i} \leq k$. Uniform sampling satisfies $\max_{i \in [n]} \frac{\|\mathbf{u}_i\|_2^2}{p_i} \leq \mu(\mathbf{U})k$, where $\mu(\mathbf{U})$ is the row coherence of \mathbf{U} . Then Property 1 and Property 2 of the two column sampling methods follow from Lemma 13.

Remark 14 Let p_1, \dots, p_n be the sampling probabilities corresponding to the leverage score sampling or uniform sampling, and let $\tilde{p}_i \in [p_i, 1]$ for all $i \in [n]$ be arbitrary. For all $i \in [n]$, if the i -th column is sampled with probability \tilde{p}_i and scaled by $\frac{1}{\sqrt{\tilde{p}_i}}$ if it gets sampled, then Lemma 2 still holds. This can be easily seen from the proof of the above lemma (in (Wang et al., 2016)). Intuitively, it indicates that if we increase the sampling probabilities, the resulting error bound will not get worse.

C.2 Count Sketch

Count sketch stems from the data stream literature (Charikar et al., 2004; Thorup and Zhang, 2012). Theoretical guarantees were first shown by Weinberger et al. (2009); Pham and Pagh (2013); Clarkson and Woodruff (2013). Meng and Mahoney (2013); Nelson and Nguyen (2013) strengthened and simplified the proofs. Because the proof is involved, we will not show the proof here. The readers can refer to (Meng and Mahoney, 2013; Nelson and Nguyen, 2013; Woodruff, 2014) for the proof.

C.3 Property 1 and Property 2 of SRHT

The properties of SRHT were established in the previous work (Drineas et al., 2011; Lu et al., 2013; Tropp, 2011). Following (Tropp, 2011), we show a simple proof of the properties of SRHT. Our analysis is based on the following two key observations.

- The scaled Walsh-Hadamard matrix $\frac{1}{\sqrt{n}} \mathbf{H}_n$ and the diagonal matrix \mathbf{D} are both orthogonal, so $\frac{1}{\sqrt{n}} \mathbf{D} \mathbf{H}_n$ is also orthogonal. If \mathbf{U} has orthonormal columns, the matrix $\frac{1}{\sqrt{n}} (\mathbf{D} \mathbf{H}_n)^T \mathbf{U}$ has orthonormal columns.
- For any fixed matrix $\mathbf{U} \in \mathbb{R}^{n \times k}$ ($k \ll n$) with orthonormal columns, the matrix $\frac{1}{\sqrt{n}} (\mathbf{D} \mathbf{H}_n)^T \mathbf{U} \in \mathbb{R}^{n \times k}$ has low row coherence with high probability. Tropp (2011) showed that the row coherence of $\frac{1}{\sqrt{n}} (\mathbf{D} \mathbf{H}_n)^T \mathbf{U}$ satisfies

$$\mu \triangleq \frac{n}{k} \max_{i \in [n]} \left\| \left(\frac{1}{\sqrt{n}} (\mathbf{D} \mathbf{H}_n)^T \mathbf{U} \right)_i \right\|_2^2 \leq \left(1 + \sqrt{\frac{8 \log(n/\delta)}{k}} \right)^2$$

with probability at least $1 - \delta$. In other words, the randomized Hadamard transform flats out the leverage scores. Consequently uniform sampling can be safely applied to form a sketch.

In the following, we use the properties of uniform sampling and the bound on the coherence μ to analyze SRHT. Let $\mathbf{V} \triangleq \frac{1}{\sqrt{n}} (\mathbf{D} \mathbf{H}_n)^T \mathbf{U} \in \mathbb{R}^{n \times k}$, $\mathbf{B} \triangleq \frac{1}{\sqrt{n}} (\mathbf{D} \mathbf{H}_n)^T \mathbf{B} \in \mathbb{R}^{n \times d}$, and μ be the row coherence of \mathbf{V} . It holds that

$$\begin{aligned} \mathbf{V}^T \mathbf{V} &= \mathbf{U}^T \mathbf{U} = \mathbf{I}_k, & \mathbf{V}^T \mathbf{P} \mathbf{P}^T \mathbf{V} &= \mathbf{U}^T \mathbf{S} \mathbf{S}^T \mathbf{U}, \\ \mathbf{V}^T \mathbf{B} &= \mathbf{U}^T \mathbf{B}, & \mathbf{V}^T \mathbf{P} \mathbf{P}^T \mathbf{B} &= \mathbf{U}^T \mathbf{S} \mathbf{S}^T \mathbf{B}, & \|\mathbf{B}\|_F &= \|\mathbf{B}\|_F, \\ \mathbb{P}\left\{\mu > (1 + \sqrt{8k-1} \log(100n))\right\} &\leq 0.01. \end{aligned}$$

Therefore it suffices to prove that

$$\begin{aligned} & \mathbb{P}\left\{\|\mathbf{I}_k - \mathbf{V}^T \mathbf{P} \mathbf{P}^T \mathbf{V}\|_2 \geq \eta\right\} \leq \delta_1 - 0.01, \\ & \mathbb{P}\left\{\|\mathbf{V} \bar{\mathbf{B}} - \mathbf{V}^T \mathbf{P} \mathbf{P}^T \bar{\mathbf{B}}\|_F \geq \epsilon \|\bar{\mathbf{B}}\|_F\right\} \leq \delta_2 - 0.01. \end{aligned}$$

The above inequalities follows from the two properties of uniform sampling.

C.4 Property 1 and Property 2 of Gaussian Projection

The two properties of Gaussian projection can be found in (Woodruff, 2014). In the following we prove Property 1 in a much simpler way than (Woodruff, 2014).

The concentration of the singular values of standard Gaussian matrix is very well known. Let \mathbf{G} be an $n \times s$ ($n > s$) standard Gaussian matrix. For any fixed matrix $\mathbf{U} \in \mathbb{R}^{n \times k}$ with orthonormal columns, the matrix $\mathbf{N} = \mathbf{G}^T \mathbf{U} \in \mathbb{R}^{s \times k}$ is also standard Gaussian matrix. Vershynin (2010) showed that for every $t \geq 0$, the following holds with probability at least $1 - 2e^{-t^2/2}$:

$$\sqrt{s} - \sqrt{k} - t \leq \sigma_k(\mathbf{N}) \leq \sigma_1(\mathbf{N}) \leq \sqrt{s} + \sqrt{k} + t.$$

Therefore, for any $\eta \in (0, 1)$, if $s = 9\eta^{-2}(\sqrt{k} + \sqrt{2 \log(2/\delta_1)})^2$, then

$$\sigma_i(\mathbf{U}^T \mathbf{S} \mathbf{S}^T \mathbf{U}) = \sigma_i^2(\mathbf{S}^T \mathbf{U}) \in [1 \pm \eta] \quad \text{for all } i \in [n]$$

hold simultaneously with probability at least $1 - \delta_1$. Hence

$$\mathbb{P}\left\{\|\mathbf{I}_k - \mathbf{U}^T \mathbf{S} \mathbf{S}^T \mathbf{U}\|_2 \geq \eta\right\} \leq \delta_1.$$

This concludes Property 1 of Gaussian projection.

C.5 Property 3 of SRHT and Gaussian Projection

The following lemma is the main result of (Cohen et al., 2015). If a sketching method satisfies Property 1 for arbitrary column orthogonal matrix \mathbf{U} , then it satisfies Property 3 due to the following lemma. Notice that the lemma does not apply to the leverage score and uniform sampling because they depends on the leverage scores or matrix coherence of specific column orthogonal matrix \mathbf{U} . The lemma is inappropriate for count sketch because Property 1 of count sketch holds with constant probability rather than arbitrary high probability.

Lemma 15 *Let $\mathbf{A} \in \mathbb{R}^{n \times k}$ and $\mathbf{B} \in \mathbb{R}^{n \times d}$ be any fixed matrices and r be any fixed integer. Let $k \geq k$ and $d \geq d$ be the least integer divisible by r . Let $\mathbf{S} \in \mathbb{R}^{n \times s}$ be a certain data-independent sketching matrix satisfying*

$$\mathbb{P}\left\{\|\mathbf{U}^T \mathbf{S} \mathbf{S}^T \mathbf{U} - \mathbf{I}\|_2 \geq \eta\right\} \leq \frac{r^2 \delta_3}{kd}$$

for any fixed matrix $\mathbf{U} \in \mathbb{R}^{n \times 2r}$ with orthonormal columns. Then

$$\|\mathbf{A}^T \mathbf{S} \mathbf{S}^T \mathbf{B} - \mathbf{A}^T \mathbf{B}\|_2 \leq \eta \left(\|\mathbf{A}\|_2^2 + \frac{\|\mathbf{A}\|_F^2 - \|\mathbf{A}\|_2^2}{r} \right) \left(\|\mathbf{B}\|_2^2 + \frac{\|\mathbf{B}\|_F^2 - \|\mathbf{B}\|_2^2}{r} \right)$$

holds with probability at least $1 - \delta_3$.

SRHT and Gaussian projection enjoys Property 1 with high probability for arbitrary column orthogonal matrix \mathbf{U} . Thus Property 3 can be immediately obtained by applying the above lemma with the setting $r = k$.

Appendix D. Proof of Theorem 3

Let $\mathbf{K} \in \mathbb{R}^{n \times n}$ be any fixed SPSPD matrix, $\mathbf{C} \in \mathbb{R}^{n \times c}$ be any fixed matrix, $\mathbf{S} \in \mathbb{R}^{n \times s}$ be a sketching matrix, and

$$\begin{aligned} \mathbf{U}^* &= \underset{\mathbf{U}}{\operatorname{argmin}} \|\mathbf{K} - \mathbf{C} \mathbf{U} \mathbf{C}^T\|_F^2 = \mathbf{C}^\dagger \mathbf{K} (\mathbf{C}^T)^\dagger, \\ \tilde{\mathbf{U}} &= \underset{\mathbf{U}}{\operatorname{argmin}} \|\mathbf{S}^T (\mathbf{K} - \mathbf{C} \mathbf{U} \mathbf{C}^T) \mathbf{S}\|_F^2 = (\mathbf{S}^T \mathbf{C})^\dagger (\mathbf{S}^T \mathbf{K} \mathbf{S}) (\mathbf{C}^T \mathbf{S})^\dagger. \end{aligned}$$

Lemma 16 is a direct consequence of Lemma 24.

Lemma 16 *Let $\mathbf{K} \in \mathbb{R}^{n \times n}$ be any fixed SPSPD matrix, $\mathbf{C} \in \mathbb{R}^{n \times c}$ be any fixed matrix, and $\mathbf{C} = \mathbf{U} \mathbf{C} \mathbf{\Sigma} \mathbf{C}^T \mathbf{V}_C^T$ be the SVD. Assume that $\mathbf{S}^T \mathbf{U} \mathbf{C}$ has full column rank. Let \mathbf{U}^* and $\tilde{\mathbf{U}}$ be defined in the above. Then the following inequality holds:*

$$\|\mathbf{K} - \mathbf{C} \tilde{\mathbf{U}} \mathbf{C}^T\|_F^2 \leq \|\mathbf{A} - \mathbf{C} \mathbf{U}^* \mathbf{C}^T\|_F^2 + \left(2f\sqrt{h} + f^2\sqrt{g_2 g_F} \right)^2,$$

where $\alpha \in [0, 1]$ is arbitrary and

$$\begin{aligned} f &= \sigma_{\min}^{-1}(\mathbf{U}_C^T \mathbf{S} \mathbf{S}^T \mathbf{U}_C), \quad h = \|\mathbf{U}_C^T \mathbf{S} \mathbf{S}^T (\mathbf{K} - \mathbf{U}_C \mathbf{U}_C^T \mathbf{K})\|_F^2 \\ g_2 &= \|\mathbf{U}_C^T \mathbf{S} \mathbf{S}^T (\mathbf{I}_n - \mathbf{U}_C \mathbf{U}_C^T) \mathbf{K}^\alpha\|_2^2, \quad g_F = \|\mathbf{U}_C^T \mathbf{S} \mathbf{S}^T (\mathbf{I}_n - \mathbf{U}_C \mathbf{U}_C^T) \mathbf{K}^{1-\alpha}\|_F^2. \end{aligned}$$

The following lemma shows that $\tilde{\mathbf{X}}$ is nearly as good as \mathbf{X}^* in terms of objective function value if \mathbf{S} satisfies Assumption 1.

Assumption 1 *Let \mathbf{B} be any fixed matrix. Let $\mathbf{C} \in \mathbb{R}^{m \times c}$ and $\mathbf{G} = \mathbf{U}_C \mathbf{\Sigma}_C \mathbf{V}_C^T$ be the SVD.*

Assume that the sketching matrix $\mathbf{S} \in \mathbb{R}^{m \times s}$ satisfies

$$\begin{aligned} & \mathbb{P}\left\{\|\mathbf{U}_C \mathbf{S} \mathbf{S}^T \mathbf{U}_C - \mathbf{I}\|_2 \geq \frac{1}{10}\right\} \leq \delta_1 \\ & \mathbb{P}\left\{\|\mathbf{U}_C^T \mathbf{S} \mathbf{S}^T \mathbf{B} - \mathbf{U}_C^T \mathbf{B}\|_F \geq \epsilon \|\mathbf{B}\|_F\right\} \leq \delta_2 \end{aligned}$$

for any $\delta_1, \delta_2 \in (0, 1/3)$.

Lemma 17 *Let $\mathbf{K} \in \mathbb{R}^{n \times n}$ be any fixed SPSPD matrix, $\mathbf{C} \in \mathbb{R}^{n \times c}$ be any fixed matrix, and $\mathbf{C} = \mathbf{U}_C \mathbf{\Sigma}_C \mathbf{V}_C^T$ be the SVD. Let \mathbf{U}^* and $\tilde{\mathbf{U}}$ be defined in the above, respectively. Let $\mathbf{S} \in \mathbb{R}^{n \times s}$ be certain sketching matrix satisfying Assumption 1. Assume that $\epsilon^{-1} = o(n)$. Then*

$$\begin{aligned} & \|\mathbf{K} - \mathbf{C} \tilde{\mathbf{U}} \mathbf{C}^T\|_F^2 - \|\mathbf{K} - \mathbf{C} \mathbf{U}^* \mathbf{C}^T\|_F^2 \\ & \leq \left(\frac{20\sqrt{\epsilon}}{9} \|\mathbf{A} - \mathbf{C} \mathbf{U}^* \mathbf{C}^T\|_F + \frac{100\epsilon}{81} \|\mathbf{I}_n - \mathbf{U}_C \mathbf{U}_C^T\|_2 \|\mathbf{K}\|_* \right)^2 \\ & \leq 4\epsilon^2 \eta \|\mathbf{A} - \mathbf{C} \mathbf{U}^* \mathbf{C}^T\|_F^2. \end{aligned}$$

holds with probability at least $1 - \delta_1 - 2\delta_2$.

Proof Let f, h, g_2, g_F, α be defined in Lemma 16 and fix $\alpha = 1/2$. Under Assumption 1 it holds simultaneously with probability at least $1 - \delta_1 - 2\delta_2$ that

$$f \leq \frac{10}{9}, \quad h \leq \epsilon \|(\mathbf{I}_n - \mathbf{U}_C \mathbf{U}_C^T) \mathbf{K}\|_F, \quad g_2 \leq g_F \leq \epsilon \|(\mathbf{I}_n - \mathbf{U}_C \mathbf{U}_C^T) \mathbf{K}^{1/2}\|_F^2.$$

It follows that

$$\begin{aligned} g_2 \leq g_F &\leq \epsilon \cdot \text{tr} \left((\mathbf{I}_n - \mathbf{U}_C \mathbf{U}_C^T) \mathbf{K}^{1/2} (\mathbf{I}_n - \mathbf{U}_C \mathbf{U}_C^T) \right) \\ &\leq \epsilon \cdot \text{tr} \left((\mathbf{I}_n - \mathbf{U}_C \mathbf{U}_C^T) \mathbf{K} (\mathbf{I}_n - \mathbf{U}_C \mathbf{U}_C^T) \right) \\ &= \epsilon \|(\mathbf{I}_n - \mathbf{U}_C \mathbf{U}_C^T) \mathbf{K} (\mathbf{I}_n - \mathbf{U}_C \mathbf{U}_C^T)\|_* \\ &\leq \epsilon \|(\mathbf{I}_n - \mathbf{U}_C \mathbf{U}_C^T) \mathbf{K}\|_*. \end{aligned}$$

It follows from Lemma 16 and the assumption $\epsilon^{-1} = o(n)$ that

$$\begin{aligned} &\|\mathbf{K} - \mathbf{C}\tilde{\mathbf{U}}\mathbf{C}^T\|_F^2 - \|\mathbf{A} - \mathbf{C}\mathbf{U}^*\mathbf{C}^T\|_F^2 \\ &\leq \left(\frac{20\sqrt{\epsilon}}{9} \|\mathbf{A} - \mathbf{C}\mathbf{U}^*\mathbf{C}^T\|_F + \frac{10^2\epsilon}{g^2} \|(\mathbf{I}_n - \mathbf{U}_C \mathbf{U}_C^T) \mathbf{K}\|_* \right)^2 \\ &\leq \left(\frac{20\sqrt{\epsilon}}{9} \|\mathbf{A} - \mathbf{C}\mathbf{U}^*\mathbf{C}^T\|_F + \frac{10^2\epsilon\sqrt{n}}{g^2} \|(\mathbf{I}_n - \mathbf{U}_C \mathbf{U}_C^T) \mathbf{K}\|_F \right)^2 \\ &= \frac{10^4\epsilon^2 n}{9^4} (1 + o(1)) \|\mathbf{A} - \mathbf{C}\mathbf{U}^*\mathbf{C}^T\|_F^2, \end{aligned}$$

by which the lemma follows. \blacksquare

Under both Assumption 1 and Assumption 2, the error bound can be further improved. We show the improved bound in Lemma 18.

Assumption 2 Let \mathbf{B} be any fixed matrix. Let $\mathbf{C} \in \mathbb{R}^{n \times c}$, $k_c = \text{rank}(\mathbf{C})$, and $\mathbf{C} = \mathbf{U}_C \mathbf{\Sigma}_C \mathbf{V}_C^T$ be the SVD. Assume that the sketching matrix $\mathbf{S} \in \mathbb{R}^{m \times s}$ satisfies

$$\mathbb{P} \left\{ \|\mathbf{U}_C^T \mathbf{S} \mathbf{S}^T \mathbf{B} - \mathbf{U}_C^T \mathbf{B}\|_2 \geq \epsilon \|\mathbf{B}\|_2^2 + \frac{\epsilon}{k_c} \|\mathbf{B}\|_F^2 \right\} \leq \delta_3$$

for any $\delta_3 \in (0, 1/3)$.

Lemma 18 Let $\mathbf{K} \in \mathbb{R}^{n \times n}$ be any fixed SPSPD matrix, $\mathbf{C} \in \mathbb{R}^{n \times c}$ be any fixed matrix, $k_c = \text{rank}(\mathbf{C})$, and $\mathbf{C} = \mathbf{U}_C \mathbf{\Sigma}_C \mathbf{V}_C^T$ be the SVD. Let \mathbf{U}^* and $\tilde{\mathbf{U}}$ be defined in the beginning of this section. Let $\mathbf{S} \in \mathbb{R}^{n \times s}$ be certain sketching matrix satisfying both Assumption 1 and Assumption 2. Assume that $\epsilon = o(n/k_c)$. Then

$$\|\mathbf{K} - \mathbf{C}\tilde{\mathbf{U}}\mathbf{C}^T\|_F^2 \leq \|\mathbf{A} - \mathbf{C}\mathbf{U}^*\mathbf{C}^T\|_F^2 + 4\epsilon^2 n/k_c \|\mathbf{A} - \mathbf{C}\mathbf{U}^*\mathbf{C}^T\|_F^2$$

holds with probability at least $1 - \delta_1 - \delta_2 - \delta_3$.

Proof Let f, h, g_2, g_F, α be defined in Lemma 16 and fix $\alpha = 0$. Under Assumption 1 it holds simultaneously with probability at least $1 - \delta_1 - \delta_2$ that

$$f \leq \frac{10}{9}, \quad h = g_F \leq \epsilon \|(\mathbf{I}_n - \mathbf{U}_C \mathbf{U}_C^T) \mathbf{K}\|_F^2.$$

Under Assumption 2, it holds with probability at least $1 - \delta_3$ that

$$\begin{aligned} g_2 &= \|\mathbf{U}_C^T \mathbf{S} \mathbf{S}^T (\mathbf{I}_n - \mathbf{U}_C \mathbf{U}_C^T) + \underbrace{\mathbf{U}_C^T (\mathbf{I}_n - \mathbf{U}_C \mathbf{U}_C^T)}_{=0}\|_2^2 \\ &\leq \epsilon \|\mathbf{I}_n - \mathbf{U}_C \mathbf{U}_C^T\|_2^2 + \frac{\epsilon}{k_c} \|\mathbf{I}_n - \mathbf{U}_C \mathbf{U}_C^T\|_F^2 \leq \epsilon + \frac{\epsilon}{k_c} (n - k_c) = \frac{\epsilon n}{k_c}. \end{aligned}$$

It follows from Lemma 16 and the assumption $\epsilon^{-1} = o(n/k_c)$ that

$$\begin{aligned} &\|\mathbf{K} - \mathbf{C}\tilde{\mathbf{U}}\mathbf{C}^T\|_F^2 - \|\mathbf{A} - \mathbf{C}\mathbf{U}^*\mathbf{C}^T\|_F^2 \\ &\leq \left(\frac{20\sqrt{\epsilon}}{9} \|\mathbf{A} - \mathbf{C}\mathbf{U}^*\mathbf{C}^T\|_F + \frac{10^2\epsilon}{g^2} \sqrt{n/k_c} \|\mathbf{A} - \mathbf{C}\mathbf{U}^*\mathbf{C}^T\|_F \right)^2 \\ &\leq 4\epsilon^2 n/k_c \|\mathbf{A} - \mathbf{C}\mathbf{U}^*\mathbf{C}^T\|_F^2, \end{aligned}$$

by which the lemma follows. \blacksquare

Finally, we prove Theorem 3 using Lemma 17 and Lemma 18. Leverage score sampling, uniform sampling, and count sketch satisfy Assumption 1, and the bounds follow by setting $\epsilon = 0.5\sqrt{\epsilon'/n}$ and applying Lemma 17. For the three sketching methods, we set $\delta_1 = 0.01$ and $\delta_2 = 0.095$.

Gaussian projection and SRHT satisfy Assumption 1 and Assumption 2, and their bounds follow by setting $\epsilon = 0.5\sqrt{\epsilon'/k_c n}$ and applying Lemma 18. For Gaussian projection, we set $\delta_1 = 0.01$, $\delta_2 = 0.09$, and $\delta_3 = 0.1$. For SRHT, we set $\delta_1 = 0.02$, $\delta_2 = 0.08$, and $\delta_3 = 0.1$.

Appendix E. Proof of Theorem 6

Since $\mathbf{C} = \mathbf{K}\mathbf{P} \in \mathbb{R}^{n \times c}$, $\mathbf{W} = \mathbf{P}^T \mathbf{C} \in \mathbb{R}^{c \times c}$, and $\text{rank}(\mathbf{S}^T \mathbf{C}) \geq \text{rank}(\mathbf{W})$, we have that

$$\text{rank}(\mathbf{K}) \geq \text{rank}(\mathbf{C}) \geq \text{rank}(\mathbf{S}^T \mathbf{C}) \geq \text{rank}(\mathbf{W}). \quad (11)$$

If $\text{rank}(\mathbf{C}) = \text{rank}(\mathbf{K})$, there exists a matrix \mathbf{X} such that $\mathbf{K} = \mathbf{C}\mathbf{X}$. By left multiplying both sides by \mathbf{P}^T , it follows that

$$\mathbf{C}^T = \mathbf{P}^T \mathbf{K} = \mathbf{P}^T \mathbf{C}\mathbf{X} = \mathbf{W}\mathbf{X},$$

and thus $\text{rank}(\mathbf{W}) = \text{rank}(\mathbf{S}^T \mathbf{C}) = \text{rank}(\mathbf{C}) = \text{rank}(\mathbf{K})$. It follows from $\mathbf{K} = \mathbf{C}\mathbf{X}$ and $\mathbf{C} = \mathbf{X}^T \mathbf{W}$ that

$$\mathbf{K} = \mathbf{X}^T \mathbf{W}\mathbf{X}.$$

We let $\Phi = \mathbf{X}\mathbf{S}$, and it holds that

$$\begin{aligned}\tilde{\mathbf{K}}_{c,s}^{\text{fast}} &= \mathbf{C}(\mathbf{S}^T \mathbf{C})^\dagger (\mathbf{S}^T \mathbf{K} \mathbf{S}) (\mathbf{C}^T \mathbf{S})^\dagger \mathbf{C}^T \\ &= \mathbf{X}^T \mathbf{W} (\mathbf{S}^T \mathbf{X}^T \mathbf{W})^\dagger (\mathbf{S}^T \mathbf{X}^T \mathbf{W} \mathbf{X} \mathbf{S}) (\mathbf{W} \mathbf{X} \mathbf{S})^\dagger \mathbf{W} \mathbf{X} \\ &= \mathbf{X}^T \mathbf{W} (\Phi^T \mathbf{W})^\dagger (\Phi^T \mathbf{W} \Phi) (\mathbf{W} \Phi)^\dagger \mathbf{W} \mathbf{X}.\end{aligned}$$

Let $\text{rank}(\mathbf{W}) = \text{rank}(\mathbf{C}) = \text{rank}(\mathbf{S}^T \mathbf{C}) = \text{rank}(\mathbf{K}) = \rho$. Since \mathbf{W} is symmetric, we denote the rank- ρ eigenvalue decomposition of \mathbf{W} by

$$\mathbf{W} = \underbrace{\mathbf{U}}_{c \times \rho} \underbrace{\mathbf{A} \mathbf{W}}_{\rho \times \rho} \underbrace{\mathbf{U}_W^T}_{\rho \times c}.$$

Since $\mathbf{S}^T \mathbf{C} = \Phi^T \mathbf{W}$ and $\text{rank}(\mathbf{S}^T \mathbf{C}) = \text{rank}(\mathbf{W}) = \rho$, we have that $\text{rank}(\Phi^T \mathbf{W}) = \text{rank}(\mathbf{W}) = \rho$. The $n \times \rho$ matrix $\Phi^T \mathbf{U}_W$ must have full column rank, otherwise $\text{rank}(\Phi^T \mathbf{W}) < \rho$. Thus we have

$$(\Phi^T \mathbf{W})^\dagger = (\Phi^T \mathbf{U}_W \mathbf{A} \mathbf{W} \mathbf{U}_W^T)^\dagger = (\mathbf{A} \mathbf{W} \mathbf{U}_W^T)^\dagger (\Phi^T \mathbf{U}_W)^\dagger.$$

It follows that

$$\begin{aligned}\tilde{\mathbf{K}}_{c,s}^{\text{fast}} &= \mathbf{X}^T \mathbf{W} \underbrace{(\mathbf{A} \mathbf{W} \mathbf{U}_W^T)^\dagger}_{c \times \rho} \underbrace{(\Phi^T \mathbf{U}_W)^\dagger}_{\rho \times n} \underbrace{(\Phi^T \mathbf{U}_W)^\dagger}_{n \times \rho} \mathbf{A} \mathbf{W} (\mathbf{U}_W^T \Phi) (\mathbf{U}_W^T \Phi)^\dagger (\mathbf{U}_W \mathbf{A} \mathbf{W})^\dagger \mathbf{W} \mathbf{X} \\ &= \mathbf{X}^T \mathbf{U}_W \mathbf{A} \mathbf{W} \mathbf{U}_W \mathbf{X} = \mathbf{X}^T \mathbf{W} \mathbf{X} = \mathbf{K}.\end{aligned}$$

This shows that the fast model is exact. To this end, we have shown that if $\text{rank}(\mathbf{C}) = \text{rank}(\mathbf{K})$, then the fast model is exact.

Conversely, if the fast model is exact, that is, $\mathbf{K} = \mathbf{C}(\mathbf{S}^T \mathbf{C})^\dagger (\mathbf{S}^T \mathbf{K} \mathbf{S}) (\mathbf{C}^T \mathbf{S})^\dagger \mathbf{C}^T$, we have that $\text{rank}(\mathbf{K}) \leq \text{rank}(\mathbf{C})$. It follows from (11) that $\text{rank}(\mathbf{K}) = \text{rank}(\mathbf{C})$.

Appendix F. Proof of Theorem 7

We prove Theorem 7 by constructing an adversarial case. Theorem 7 is a direct consequence of the following theorem.

Theorem 19 *Let \mathbf{A} be the $n \times n$ symmetric matrix defined in Lemma 21 with $\alpha \rightarrow 1$ and k be any positive integer smaller than n . Let \mathcal{P} be any subset of $[n]$ with cardinality c and $\mathbf{C} \in \mathbb{R}^{n \times c}$ contain c columns of \mathbf{A} indexed by \mathcal{P} . Let \mathbf{S} be any $n \times s$ column selection matrix satisfying $\mathcal{P} \subset S$, where $S \subset [n]$ is the index set formed by \mathbf{S} . Then the following inequality holds:*

$$\frac{\|\mathbf{A} - \mathbf{C}(\mathbf{S}^T \mathbf{C})^\dagger (\mathbf{S}^T \mathbf{A} \mathbf{S}) (\mathbf{C}^T \mathbf{S})^\dagger \mathbf{C}^T\|_F^2}{\|\mathbf{A} - \mathbf{A}_k\|_F^2} \geq \frac{n-c}{n-k} \left(1 + \frac{2k}{c}\right) + \frac{n-s}{n-k} \frac{k(n-s)}{s^2}.$$

Proof Let \mathbf{A} and \mathbf{B} be defined in Lemma 21. We prove the theorem using Lemma 21 and Lemma 23. Let $n = pk$. Let \mathbf{C} consist of c column sampled from \mathbf{A} and $\tilde{\mathbf{C}}_i$ consist of c_i columns sampled from the i -th diagonal block of \mathbf{A} . Thus $\mathbf{C} = \text{diag}(\tilde{\mathbf{C}}_1, \dots, \tilde{\mathbf{C}}_k)$. Without loss of generality, we assume $\tilde{\mathbf{C}}_i$ consists of the first c_i columns of \mathbf{B} . Let $\tilde{\mathbf{S}} =$

$\text{diag}(\tilde{\mathbf{S}}_1, \dots, \tilde{\mathbf{S}}_k)$ be an $n \times s$ column selection matrix, where $\tilde{\mathbf{S}}_i$ is a $p \times s_i$ column selection matrix and $s_1 + \dots + s_k = s$. Then the \mathbf{U} matrix is computed by

$$\begin{aligned}\mathbf{U} &= (\mathbf{S}^T \mathbf{C})^\dagger (\mathbf{S}^T \mathbf{A} \mathbf{S}) (\mathbf{C}^T \mathbf{S})^\dagger \\ &= [\text{diag}(\tilde{\mathbf{S}}_1^T \tilde{\mathbf{C}}_1, \dots, \tilde{\mathbf{S}}_k^T \tilde{\mathbf{C}}_k)]^\dagger \text{diag}(\tilde{\mathbf{S}}_1^T \mathbf{B} \tilde{\mathbf{S}}_1, \dots, \tilde{\mathbf{S}}_k^T \mathbf{B} \tilde{\mathbf{S}}_k) [\text{diag}(\tilde{\mathbf{C}}_1^T \tilde{\mathbf{S}}_1, \dots, \tilde{\mathbf{C}}_k^T \tilde{\mathbf{S}}_k)]^\dagger \\ &= \text{diag}\left((\tilde{\mathbf{S}}_1^T \tilde{\mathbf{C}}_1)^\dagger (\tilde{\mathbf{S}}_1^T \mathbf{B} \tilde{\mathbf{S}}_1) (\tilde{\mathbf{C}}_1^T \tilde{\mathbf{S}}_1)^\dagger, \dots, (\tilde{\mathbf{S}}_k^T \tilde{\mathbf{C}}_k)^\dagger (\tilde{\mathbf{S}}_k^T \mathbf{B} \tilde{\mathbf{S}}_k) (\tilde{\mathbf{C}}_k^T \tilde{\mathbf{S}}_k)^\dagger\right).\end{aligned}$$

The approximation formed by the fast model is the block-diagonal matrix whose the i -th ($i \in [k]$) diagonal block is the $p \times p$ matrix

$$[\tilde{\mathbf{A}}_{c,s}^{\text{fast}}]_{ii} = \tilde{\mathbf{C}}_i (\tilde{\mathbf{S}}_i^T \tilde{\mathbf{C}}_i)^\dagger (\tilde{\mathbf{S}}_i^T \mathbf{B} \tilde{\mathbf{S}}_i) (\tilde{\mathbf{C}}_i^T \tilde{\mathbf{S}}_i)^\dagger \mathbf{C}_i^T.$$

It follows from Lemma 23 that for any $i \in [k]$,

$$\lim_{\alpha \rightarrow 1} \frac{\|\mathbf{B} - [\tilde{\mathbf{A}}_{c,s}^{\text{fast}}]_{ii}\|_F^2}{(1-\alpha)^2} = (p-c_i) \left(1 + \frac{2}{c_i}\right) + \frac{(p-s_i)^2}{s_i^2}.$$

Thus

$$\begin{aligned}\lim_{\alpha \rightarrow 1} \frac{\|\mathbf{A} - \tilde{\mathbf{A}}_{c,s}^{\text{fast}}\|_F^2}{(1-\alpha)^2} &= \lim_{\alpha \rightarrow 1} \sum_{i=1}^k \frac{\|\mathbf{B} - [\tilde{\mathbf{A}}_{c,s}^{\text{fast}}]_{ii}\|_F^2}{(1-\alpha)^2} \\ &= \sum_{i=1}^k (p-c_i) \left(1 + \frac{2}{c_i}\right) + \frac{(p-s_i)^2}{s_i^2} \\ &= \left(\sum_{i=1}^k p - c_i - 2\right) + \left(2p \sum_{i=1}^k \frac{1}{c_i}\right) + \left(p^2 \sum_{i=1}^k \frac{1}{s_i^2}\right) - \left(2p \sum_{i=1}^k \frac{1}{s_i}\right) + k \\ &\geq n-c-2k + \frac{2nk}{c} + \frac{kn^2}{s^2} - 2nk + k \\ &= (n-c) \left(1 + \frac{2k}{c}\right) + \frac{k(n-s)^2}{s^2}.\end{aligned}$$

Here the inequality follows by minimizing over c_1, \dots, c_k and s_1, \dots, s_k with constraints $\sum_i c_i = c$ and $\sum_i s_i = s$. Finally, it follows from Lemma 21 that

$$\lim_{\alpha \rightarrow 1} \frac{\|\mathbf{A} - \tilde{\mathbf{A}}_{c,s}^{\text{fast}}\|_F^2}{\|\mathbf{A} - \mathbf{A}_k\|_F^2} \geq \frac{n-c}{n-k} \left(1 + \frac{2k}{c}\right) + \frac{n-s}{n-k} \frac{k(n-s)}{s^2}.$$

■

F.1 Key Lemmas

Lemma 20 provides a useful tool for expanding the Moore-Penrose inverse of partitioned matrices.

Lemma 20 (Page 179 of Ben-Israel and Greville (2003)) Given a matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ of rank c which has a nonsingular $c \times c$ submatrix \mathbf{X}_{11} . By rearrangement of columns and rows by permutation matrices \mathbf{P} and \mathbf{Q} , the submatrix \mathbf{X}_{11} can be brought to the top left corner of \mathbf{X} , that is,

$$\mathbf{P}\mathbf{X}\mathbf{Q} = \begin{bmatrix} \mathbf{X}_{11} & \mathbf{X}_{12} \\ \mathbf{X}_{21} & \mathbf{X}_{22} \end{bmatrix}.$$

Then the Moore-Penrose inverse of \mathbf{X} is

$$\mathbf{X}^\dagger = \mathbf{Q} \begin{bmatrix} \mathbf{I}_c & \\ \mathbf{T}^T & \end{bmatrix} (\mathbf{I}_c + \mathbf{T}\mathbf{T}^T)^{-1} \mathbf{X}_{11}^{-1} (\mathbf{I}_c + \mathbf{H}^T\mathbf{H})^{-1} \begin{bmatrix} \mathbf{I}_c & \mathbf{H}^T \end{bmatrix} \mathbf{P},$$

where $\mathbf{T} = \mathbf{X}_{11}^{-1}\mathbf{X}_{12}$ and $\mathbf{H} = \mathbf{X}_{21}\mathbf{X}_{11}^{-1}$.

Lemmas 21 and 23 will be used to prove Theorem 19.

Lemma 21 (Lemma 19 of Wang and Zhang (2013)) Given n and k , we let \mathbf{B} be an $\frac{n}{k} \times \frac{n}{k}$ matrix whose diagonal entries equal to one and off-diagonal entries equal to $\alpha \in [0, 1)$. We let \mathbf{A} be an $n \times n$ block-diagonal matrix

$$\mathbf{A} = \text{diag}(\underbrace{\mathbf{B}, \dots, \mathbf{B}}_{k \text{ blocks}}). \quad (12)$$

Let \mathbf{A}_k be the best rank- k approximation to the matrix \mathbf{A} , then we have that

$$\|\mathbf{A} - \mathbf{A}_k\|_F^2 = (1 - \alpha)^2(n - k).$$

Lemma 22 The following equality holds for any nonzero real number a :

$$(a\mathbf{I}_c + b\mathbf{I}_c\mathbf{I}_c^T)^{-1} = a^{-1}\mathbf{I}_c - \frac{b}{a(a + bc)}\mathbf{I}_c\mathbf{I}_c^T.$$

Proof The lemma directly follows from the Sherman-Morrison-Woodbury matrix identity

$$(\mathbf{X} + \mathbf{Y}\mathbf{Z}\mathbf{R})^{-1} = \mathbf{X}^{-1} - \mathbf{X}^{-1}\mathbf{Y}(\mathbf{Z}^{-1} + \mathbf{R}\mathbf{X}^{-1}\mathbf{Y})^{-1}\mathbf{R}\mathbf{X}^{-1}.$$

■

Lemma 23 Let \mathbf{B} be any $n \times n$ matrix with diagonal entries equal to one and off-diagonal entries equal to α . Let $\mathbf{C} = \mathbf{B}\mathbf{P} \in \mathbb{R}^{n \times c}$; let $\tilde{\mathbf{B}} = \mathbf{C}(\mathbf{S}^T\mathbf{C})^\dagger(\mathbf{S}^T\mathbf{K}\mathbf{S})(\mathbf{C}^T\mathbf{S})^\dagger\mathbf{C}^T$ be the fast SPSD matrix approximation model of \mathbf{B} . Let \mathcal{P} and \mathcal{S} be the index sets formed by \mathbf{P} and \mathbf{S} , respectively. If $\mathcal{P} \subset \mathcal{S}$, the error incurred by the fast model satisfies

$$\lim_{\alpha \rightarrow 1} \frac{\|\mathbf{B} - \tilde{\mathbf{B}}\|_F^2}{(1 - \alpha)^2} \geq (n - c) \left(1 + \frac{2}{c}\right) + \frac{(n - s)^2}{s^2}.$$

Proof Let $\mathbf{B}_1 = \mathbf{S}^T\mathbf{B}\mathbf{S} \in \mathbb{R}^{s \times s}$ and $\mathbf{C}_1 = \mathbf{S}^T\mathbf{C} = \mathbf{S}^T\mathbf{B}\mathbf{P} \in \mathbb{R}^{s \times c}$. Without loss of generality, we assume that \mathbf{P} selects the first c columns and \mathbf{S} selects the first s columns. We partition \mathbf{B} and \mathbf{C} by:

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{B}_3^T \\ \mathbf{B}_3 & \mathbf{B}_2 \end{bmatrix} \quad \text{and} \quad \mathbf{C} = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{W} \\ \mathbf{C}_2 \end{bmatrix}.$$

We further partition $\mathbf{B}_1 \in \mathbb{R}^{s \times s}$ by

$$\mathbf{B}_1 = \begin{bmatrix} \mathbf{W} & \mathbf{C}_{12}^T \\ \mathbf{C}_{12} & \mathbf{B}_{12} \end{bmatrix},$$

where

$$\mathbf{C}_{12} = \alpha \mathbf{1}_{s-c}\mathbf{1}_c^T \quad \text{and} \quad \mathbf{B}_{12} = (1 - \alpha)\mathbf{I}_{s-c} + \alpha \mathbf{1}_{s-c}\mathbf{1}_{s-c}^T.$$

The \mathbf{U} matrix is computed by

$$\mathbf{U} = (\mathbf{S}^T\mathbf{C})^\dagger(\mathbf{S}^T\mathbf{B}\mathbf{S})(\mathbf{C}^T\mathbf{S})^\dagger = \mathbf{C}_1^\dagger\mathbf{B}_1(\mathbf{C}_1^\dagger)^T.$$

It is not hard to see that \mathbf{C}_1 contains the first c rows of \mathbf{B}_1 .

We expand the Moore-Penrose inverse of \mathbf{C}_1 by Lemma 20 and obtain

$$\mathbf{C}_1^\dagger = \mathbf{W}^{-1}(\mathbf{I}_c + \mathbf{H}^T\mathbf{H})^{-1} \begin{bmatrix} \mathbf{I}_c & \mathbf{H}^T \end{bmatrix},$$

where

$$\mathbf{W}^{-1} = \left((1 - \alpha)\mathbf{I}_c + \alpha \mathbf{1}_c\mathbf{1}_c^T \right)^{-1} = \frac{1}{1 - \alpha}\mathbf{I}_c - \frac{\alpha}{(1 - \alpha)(1 - \alpha + c\alpha)}\mathbf{1}_c\mathbf{1}_c^T$$

and

$$\mathbf{H} = \mathbf{C}_{12}\mathbf{W}^{-1} = \frac{\alpha}{1 - \alpha + c\alpha}\mathbf{1}_{s-c}\mathbf{1}_c^T.$$

It is easily verified that $\mathbf{H}^T\mathbf{H} = \left(\frac{\alpha}{1 - \alpha + c\alpha}\right)^2(s - c)\mathbf{1}_c\mathbf{1}_c^T$. It follows from Lemma 22 that

$$(\mathbf{I}_c + \mathbf{H}^T\mathbf{H})^{-1} = \mathbf{I}_c - \frac{(s - c)\alpha^2}{c(s - c)\alpha^2 + (1 - \alpha + c\alpha)^2}\mathbf{1}_c\mathbf{1}_c^T.$$

Then we obtain

$$\begin{aligned} \mathbf{C}_1^\dagger &= \mathbf{W}^{-1}(\mathbf{I}_c + \mathbf{H}^T\mathbf{H})^{-1} \begin{bmatrix} \mathbf{I}_c & \mathbf{H}^T \end{bmatrix} \\ &= \left(\frac{1}{1 - \alpha}\mathbf{I}_c + \gamma_1\mathbf{1}_c\mathbf{1}_c^T \right) \begin{bmatrix} \mathbf{I}_c & \mathbf{H}^T \end{bmatrix}, \end{aligned} \quad (13)$$

where

$$\begin{aligned} \gamma_1 &= c\gamma_2\gamma_3 - \gamma_2 - \frac{\gamma_3}{1 - \alpha}, \\ \gamma_2 &= \frac{\alpha}{(1 - \alpha)(1 - \alpha + c\alpha)}, \\ \gamma_3 &= \frac{(s - c)\alpha^2}{c(s - c)\alpha^2 + (1 - \alpha + c\alpha)^2}. \end{aligned}$$

Then

$$\begin{aligned} [\mathbf{I}_c; \mathbf{H}^T] \mathbf{B}_1 [\mathbf{I}_c; \mathbf{H}^T]^T &= \mathbf{W} + \mathbf{B}_{13}^T \mathbf{H} + \mathbf{H}^T \mathbf{B}_{13} + \mathbf{H}^T \mathbf{B}_{12} \mathbf{H} \\ &= (1-\alpha) \mathbf{I}_c + \gamma_4 \mathbf{1}_c \mathbf{1}_c^T, \end{aligned} \quad (14)$$

where

$$\gamma_4 = \frac{\alpha(3\alpha s - \alpha c - 2\alpha + \alpha^2 c - 3\alpha c^2 s + \alpha^2 + \alpha^2 s^2 + 1)}{(\alpha c - \alpha + 1)^2}.$$

It follows from (13) (14) that

$$\begin{aligned} \mathbf{U} &= \mathbf{C}_1^T \mathbf{B}_1 (\mathbf{C}_1^T)^T = \left(\frac{1}{1-\alpha} \mathbf{I}_c + \gamma_1 \mathbf{1}_c \mathbf{1}_c^T \right) \left((1-\alpha) \mathbf{I}_c + \gamma_4 \mathbf{1}_c \mathbf{1}_c^T \right) \left(\frac{1}{1-\alpha} \mathbf{I}_c + \gamma_1 \mathbf{1}_c \mathbf{1}_c^T \right) \\ &= \frac{1}{1-\alpha} \mathbf{I}_c + \gamma_5 \mathbf{1}_c \mathbf{1}_c^T, \end{aligned}$$

where

$$\gamma_5 = \gamma_1 + \left(c\gamma_1 + \frac{1}{1-\alpha} \right) \left(c\gamma_1 \gamma_4 + \gamma_1(1-\alpha) + \frac{\gamma_4}{1-\alpha} \right).$$

Then we have

$$\begin{aligned} \mathbf{W}\mathbf{U} &= \mathbf{I}_c + \gamma_6 \mathbf{1}_c \mathbf{1}_c^T, \\ \gamma_6 &= (1-\alpha + \alpha c) \gamma_5 + \frac{\alpha}{1-\alpha}. \end{aligned}$$

We partition the fast SPSPD matrix approximation model by

$$\tilde{\mathbf{B}} = \begin{bmatrix} \tilde{\mathbf{W}} & \tilde{\mathbf{B}}_{21}^T \\ \tilde{\mathbf{B}}_{21} & \tilde{\mathbf{B}}_{22} \end{bmatrix},$$

where

$$\begin{aligned} \mathbf{B}_{11} &= \mathbf{W}\mathbf{U}\mathbf{W} = (1-\alpha) \mathbf{I}_c + (\alpha + (1-\alpha + \alpha c) \gamma_6) \mathbf{1}_c \mathbf{1}_c^T, \\ \tilde{\mathbf{B}}_{21} &= \mathbf{W}\mathbf{U}(\alpha \mathbf{1}_c \mathbf{1}_{n-c}^T) = \alpha(1 + c\gamma_6) \mathbf{1}_c \mathbf{1}_{n-c}^T, \\ \tilde{\mathbf{B}}_{22} &= (\alpha \mathbf{1}_{n-c} \mathbf{1}_c^T) \mathbf{U}(\alpha \mathbf{1}_c \mathbf{1}_{n-c}^T) = \alpha^2 c \left(\frac{1}{1-\alpha} + \gamma_5 c \right) \mathbf{1}_c \mathbf{1}_{n-c}^T \end{aligned}$$

The approximate error is

$$\|\mathbf{B} - \tilde{\mathbf{B}}\|_F^2 = \|\mathbf{W} - \tilde{\mathbf{W}}\|_F^2 + 2\|\mathbf{B}_{21} - \tilde{\mathbf{B}}_{21}\|_F^2 + \|\mathbf{B}_{22} - \tilde{\mathbf{B}}_{22}\|_F^2,$$

where

$$\begin{aligned} \|\mathbf{W} - \tilde{\mathbf{W}}\|_F^2 &= \|(1-\alpha + \alpha c) \gamma_6 \mathbf{1}_c \mathbf{1}_c^T\|_F^2 = c^2(1-\alpha + \alpha c) \gamma_6^2, \\ \|\mathbf{B}_{21} - \tilde{\mathbf{B}}_{21}\|_F^2 &= \|\alpha c \gamma_6 \mathbf{1}_c \mathbf{1}_{n-c}^T\|_F^2 = \alpha^2 c^3 (n-c) \gamma_6^2, \\ \|\mathbf{B}_{22} - \tilde{\mathbf{B}}_{22}\|_F^2 &= \underbrace{(n-c)(n-c-1)\alpha^2 \left(\frac{\alpha c}{1-\alpha} + \alpha c^2 \gamma_5 - 1 \right)^2}_{\text{off-diagonal}} + \underbrace{(n-c) \left(\frac{\alpha^2 c}{1-\alpha} + \alpha^2 c^2 \gamma_5 - 1 \right)^2}_{\text{diagonal}}. \end{aligned}$$

We let

$$\eta \triangleq \frac{\|\mathbf{B} - \tilde{\mathbf{B}}\|_F^2}{(1-\alpha)^2},$$

which is a symbolic expression of α , n , s , and c . We then simplify the expression using MATLAB and substitute the α in η by 1, and we obtain

$$\lim_{\alpha \rightarrow 1} \eta = (n-c)(1+2/c) + (n-s)^2/s^2,$$

by which the lemma follows. \blacksquare

Appendix G. Proof of Theorem 8

We define the projection operation $\mathcal{P}_{C,k}(\mathbf{A}) = \mathbf{C}\mathbf{X}$ where \mathbf{X} is defined by

$$\mathbf{X} = \underset{\text{rank}(\mathbf{X}) \leq k}{\text{argmin}} \|\mathbf{A} - \mathbf{C}\mathbf{X}\|_F^2.$$

By sampling $c = 2k\epsilon^{-1}(1 + o(1))$ columns of \mathbf{A} by the near-optimal algorithm of Boutsidis et al. (2014) to form $\mathbf{C} \in \mathbb{R}^{m \times c}$, we have that

$$\|\mathbf{A} - \mathcal{P}_{C,k}(\mathbf{A})\|_F^2 \leq (1+\epsilon) \|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

Applying Lemma 3.11 of Boutsidis and Woodruff (2014), there exists a much smaller column orthogonal matrix $\mathbf{Z} \in \mathbb{R}^{m \times k}$ such that $\text{range}(\mathbf{Z}) \subset \text{range}(\mathbf{C})$ and

$$\|\mathbf{A} - \mathbf{C}\mathbf{C}^T \mathbf{A}\|_F^2 \leq \|\mathbf{A} - \mathbf{Z}\mathbf{Z}^T \mathbf{A}\|_F^2 \leq \|\mathbf{A} - \mathcal{P}_{C,k}(\mathbf{A})\|_F^2.$$

Notice that the algorithm does not compute \mathbf{Z} .

Let $\mathbf{R}_1^T \in \mathbb{R}^{n \times r_1}$ be columns of \mathbf{A}^T selected by the randomized dual-set sparsification algorithm of Boutsidis et al. (2014). When $r_1 = \mathcal{O}(k)$, it holds that

$$\|\mathbf{A} - \mathbf{R}_1 \mathbf{R}_1^T \mathbf{A}\|_F^2 \leq 2(1 + o(1)) \|\mathbf{A} - \mathbf{A}_k\|_F^2.$$

Let $\mathbf{R}_2^T \in \mathbb{R}^{n \times r_2}$ be columns of \mathbf{A}^T selected by adaptive sampling according to the residual $\mathbf{A}^T - \mathbf{R}_1^T (\mathbf{R}_1^T)^T \mathbf{A}^T$. Set $r_2 = 2k\epsilon^{-1}(1 + o(1))$. Let $\mathbf{R}^T = [\mathbf{R}_1^T, \mathbf{R}_2^T]$. By the adaptive sampling theorem of Wang and Zhang (2013), we obtain

$$\begin{aligned} \|\mathbf{A} - \mathbf{Z}\mathbf{Z}^T \mathbf{A}\mathbf{R}\mathbf{R}^T \mathbf{A}\|_F^2 &\leq \|\mathbf{A} - \mathbf{Z}\mathbf{Z}^T \mathbf{A}\|_F^2 + \frac{k}{r_2} \mathbb{E} \|\mathbf{A} - \mathbf{A}\mathbf{R}\mathbf{R}^T \mathbf{A}\|_F^2 \\ &\leq (1+\epsilon) \|\mathbf{K} - \mathbf{K}_k\|_F^2 + \epsilon \|\mathbf{K} - \mathbf{K}_k\|_F^2 \\ &\leq (1+2\epsilon) \|\mathbf{K} - \mathbf{K}_k\|_F^2. \end{aligned} \quad (15)$$

Obviously \mathbf{R}^T contains

$$r = r_1 + r_2 = 2k\epsilon^{-1}(1 + o(1))$$

columns of \mathbf{A}^T .

It remains to show $\|\mathbf{A} - \mathbf{C}\mathbf{C}^\dagger\mathbf{A}\mathbf{R}^\dagger\mathbf{R}\|_F^2 \leq \|\mathbf{A} - \mathbf{Z}\mathbf{Z}^T\mathbf{A}\mathbf{R}^\dagger\mathbf{R}\|_F^2$. Since the columns of \mathbf{Z} are contained in the column space of \mathbf{C} , for any matrix \mathbf{Y} the inequality $\|(\mathbf{I}_m - \mathbf{C}\mathbf{C}^\dagger)\mathbf{Y}\|_F^2 \leq \|(\mathbf{I}_m - \mathbf{Z}\mathbf{Z}^T)\mathbf{Y}\|_F^2$ holds. Then we obtain

$$\begin{aligned} \|\mathbf{A} - \mathbf{C}\mathbf{C}^\dagger\mathbf{A}\mathbf{R}^\dagger\mathbf{R}\|_F^2 &= \|\mathbf{A} - \mathbf{A}\mathbf{R}^\dagger\mathbf{R} + \mathbf{A}\mathbf{R}^\dagger\mathbf{R} - \mathbf{C}\mathbf{C}^\dagger\mathbf{A}\mathbf{R}^\dagger\mathbf{R}\|_F^2 \\ &= \|\mathbf{A}(\mathbf{I}_n - \mathbf{R}^\dagger\mathbf{R})\|_F^2 + \|(\mathbf{I}_m - \mathbf{C}\mathbf{C}^\dagger)\mathbf{A}\mathbf{R}^\dagger\mathbf{R}\|_F^2 \\ &\leq \|\mathbf{A}(\mathbf{I}_n - \mathbf{R}^\dagger\mathbf{R})\|_F^2 + \|(\mathbf{I}_m - \mathbf{Z}\mathbf{Z}^T)\mathbf{A}\mathbf{R}^\dagger\mathbf{R}\|_F^2 \\ &= \|\mathbf{A}(\mathbf{I}_n - \mathbf{R}^\dagger\mathbf{R}) + (\mathbf{I}_m - \mathbf{Z}\mathbf{Z}^T)\mathbf{A}\mathbf{R}^\dagger\mathbf{R}\|_F^2 \\ &= \|\mathbf{A} - \mathbf{Z}\mathbf{Z}^T\mathbf{A}\mathbf{R}^\dagger\mathbf{R}\|_F^2. \end{aligned} \quad (16)$$

The theorem follows from (15) and (16) and by setting $\epsilon' = 2\epsilon$.

Appendix H. Proof of Theorem 9

In Section H.1 we establish a key lemma to decompose the error incurred by the approximation. In Section H.2 we prove Theorem 9 using the key lemma.

H.1 Key Lemma

We establish the following lemma for decomposing the error of the approximate solution.

Lemma 24 *Let $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{C} \in \mathbb{R}^{m \times c}$, and $\mathbf{R} \in \mathbb{R}^{r \times n}$ be any fixed matrices, and $\mathbf{A} = \mathbf{U}_A \Sigma_A \mathbf{V}_A^T$, $\mathbf{C} = \mathbf{U}_C \Sigma_C \mathbf{V}_C^T$, $\mathbf{R} = \mathbf{U}_R \Sigma_R \mathbf{V}_R^T$ be the SVD. Assume that $\mathbf{S}_C^T \mathbf{U}_C$ and $\mathbf{S}_R^T \mathbf{V}_R$ have full column rank. Let \mathbf{U}^* and \mathbf{V} be defined in (8) and (9), respectively. Then the following inequalities hold:*

$$\begin{aligned} \|\mathbf{A} - \mathbf{C}\mathbf{U}\mathbf{R}\|_F^2 &\leq \|\mathbf{A} - \mathbf{C}\mathbf{U}^*\mathbf{R}\|_F^2 + \left(f_R \sqrt{h_R} + f_C \sqrt{h_C} + f_C f_R \sqrt{g_C g_R} \right)^2, \\ \|\mathbf{A} - \mathbf{C}\mathbf{U}\mathbf{R}\|_F^2 &\leq \|\mathbf{A} - \mathbf{C}\mathbf{U}^*\mathbf{R}\|_F^2 + \left(f_R \sqrt{h_R} + f_C \sqrt{h_C} + f_C f_R \sqrt{g_C g_R} \right)^2, \end{aligned}$$

where $\alpha \in [0, 1]$ is arbitrary, and

$$\begin{aligned} f_C &= \sigma_{\min}^{-1}(\mathbf{U}_C^T \mathbf{S}_C \mathbf{S}_C^T \mathbf{U}_C), & f_R &= \sigma_{\min}^{-1}(\mathbf{V}_R^T \mathbf{S}_R \mathbf{S}_R^T \mathbf{V}_R), \\ h_C &= \|\mathbf{U}_C^T \mathbf{S}_C \mathbf{S}_C^T (\mathbf{A} - \mathbf{U}_C \mathbf{U}_C^T \mathbf{A})\|_F^2, & h_R &= \|(\mathbf{A} - \mathbf{A}\mathbf{V}_R \mathbf{V}_R^T) \mathbf{S}_C \mathbf{S}_C^T \mathbf{V}_R\|_F^2, \\ g_C &= \|\mathbf{U}_C^T \mathbf{S}_C \mathbf{S}_C^T (\mathbf{I}_m - \mathbf{U}_C \mathbf{U}_C^T) \mathbf{U}_A \Sigma_A\|_F^2, & g_R &= \|\Sigma_A^{1-\alpha} \mathbf{V}_A (\mathbf{I}_n - \mathbf{V}_R \mathbf{V}_R^T) \mathbf{S}_R \mathbf{S}_R^T \mathbf{V}_R\|_F^2, \\ g'_C &= \|\mathbf{U}_C^T \mathbf{S}_C \mathbf{S}_C^T (\mathbf{I}_m - \mathbf{U}_C \mathbf{U}_C^T) \mathbf{U}_A \Sigma_A^\alpha\|_2^2, & g'_R &= \|\Sigma_A^{1-\alpha} \mathbf{V}_A (\mathbf{I}_n - \mathbf{V}_R \mathbf{V}_R^T) \mathbf{S}_R \mathbf{S}_R^T \mathbf{V}_R\|_2^2. \end{aligned}$$

Proof Let $k_C = \text{rank}(\mathbf{C}) \leq c$ and $k_R = \text{rank}(\mathbf{R}) \leq r$. Let $\mathbf{U}_C \in \mathbb{R}^{m \times k_C}$ be the left singular vectors of \mathbf{C} and $\mathbf{V}_R \in \mathbb{R}^{n \times k_R}$ be the right singular vectors of \mathbf{R} . Define \mathbf{Z}^* , $\tilde{\mathbf{Z}} \in \mathbb{R}^{k_C \times k_R}$ by

$$\mathbf{Z}^* = \mathbf{U}_C^T \mathbf{A} \mathbf{V}_R, \quad \tilde{\mathbf{Z}} = (\mathbf{S}_C^T \mathbf{U}_C)^\dagger (\mathbf{S}_C^T \mathbf{A} \mathbf{S}_R) (\mathbf{V}_R^T \mathbf{S}_R)^\dagger.$$

We have that $\mathbf{C}\mathbf{U}^*\mathbf{R} = \mathbf{C}\mathbf{C}^\dagger\mathbf{A}\mathbf{R}^\dagger\mathbf{R} = \mathbf{U}_C \mathbf{U}_C^T \mathbf{A} \mathbf{V}_R \mathbf{V}_R^T = \mathbf{U}_C \mathbf{Z}^* \mathbf{V}_R^T$. By definition, it holds that that

$$\begin{aligned} \tilde{\mathbf{U}} &= (\mathbf{S}_C^T \mathbf{C})^\dagger (\mathbf{S}_C^T \mathbf{A} \mathbf{S}_R) (\mathbf{R} \mathbf{S}_R)^\dagger \\ &= (\mathbf{S}_C^T \mathbf{U}_C \Sigma_C \mathbf{V}_C^T)^\dagger (\mathbf{S}_C^T \mathbf{A} \mathbf{S}_R) (\mathbf{U}_R \Sigma_R \mathbf{V}_R^T \mathbf{S}_R)^\dagger \\ &= (\Sigma_C \mathbf{V}_C^T)^\dagger (\mathbf{S}_C^T \mathbf{U}_C)^\dagger (\mathbf{S}_C^T \mathbf{A} \mathbf{S}_R) (\mathbf{V}_R^T \mathbf{S}_R)^\dagger (\mathbf{U}_R \Sigma_R)^\dagger \\ &= (\Sigma_C \mathbf{V}_C^T)^\dagger \tilde{\mathbf{Z}} (\mathbf{U}_R \Sigma_R)^\dagger, \end{aligned}$$

where the third equality follows from that $\mathbf{S}_C^T \mathbf{U}_C$ and $\mathbf{S}_R^T \mathbf{V}_R$ have full column rank and that $\Sigma_C \mathbf{V}_C^T$ and $\mathbf{V}_R^T \mathbf{S}_R$ have full row rank. It follows that

$$\mathbf{C}\tilde{\mathbf{U}}\mathbf{R} = \mathbf{U}_C \Sigma_C \mathbf{V}_C^T (\Sigma_C \mathbf{V}_C^T)^\dagger \tilde{\mathbf{Z}} (\mathbf{U}_R \Sigma_R)^\dagger \mathbf{U}_R \Sigma_R \mathbf{V}_R^T = \mathbf{U}_C \tilde{\mathbf{Z}} \mathbf{V}_R^T.$$

Since $\mathbf{C}\mathbf{U}^*\mathbf{R} = \mathbf{U}_C \mathbf{Z}^* \mathbf{V}_R^T$ and $\mathbf{C}\tilde{\mathbf{U}}\mathbf{R} = \mathbf{U}_C \tilde{\mathbf{Z}} \mathbf{V}_R^T$, it suffices to prove the two inequalities:

$$\begin{aligned} \|\mathbf{A} - \mathbf{U}_C \tilde{\mathbf{Z}} \mathbf{V}_R^T\|_F^2 &\leq \|\mathbf{A} - \mathbf{U}_C \mathbf{Z}^* \mathbf{V}_R^T\|_F^2 + \left(f_R \sqrt{h_R} + f_C \sqrt{h_C} + f_C f_R \sqrt{g_C g_R} \right)^2, \\ \|\mathbf{A} - \mathbf{U}_C \tilde{\mathbf{Z}} \mathbf{V}_R^T\|_F^2 &\leq \|\mathbf{A} - \mathbf{U}_C \mathbf{Z}^* \mathbf{V}_R^T\|_F^2 + \left(f_R \sqrt{h_R} + f_C \sqrt{h_C} + f_C f_R \sqrt{g_C g_R} \right)^2. \end{aligned} \quad (17)$$

The left-hand side can be expressed as

$$\begin{aligned} \|\mathbf{A} - \mathbf{U}_C \tilde{\mathbf{Z}} \mathbf{V}_R^T\|_F^2 &= \|(\mathbf{A} - \mathbf{U}_C \mathbf{Z}^* \mathbf{V}_R^T) + \mathbf{U}_C (\mathbf{Z}^* - \tilde{\mathbf{Z}}) \mathbf{V}_R^T\|_F^2 \\ &= \|(\mathbf{I}_m - \mathbf{U}_C \mathbf{U}_C^T) \mathbf{A} + \mathbf{U}_C \mathbf{U}_C^T (\mathbf{A} (\mathbf{I}_n - \mathbf{V}_R \mathbf{V}_R^T) + \mathbf{U}_C (\mathbf{Z}^* - \tilde{\mathbf{Z}}) \mathbf{V}_R^T)\|_F^2 \\ &= \|(\mathbf{I}_m - \mathbf{U}_C \mathbf{U}_C^T) \mathbf{A}\|_F^2 + \|\mathbf{U}_C \mathbf{U}_C^T \mathbf{A} (\mathbf{I}_n - \mathbf{V}_R \mathbf{V}_R^T) + \mathbf{U}_C (\mathbf{Z}^* - \tilde{\mathbf{Z}}) \mathbf{V}_R^T\|_F^2 \\ &= \|(\mathbf{I}_m - \mathbf{U}_C \mathbf{U}_C^T) \mathbf{A}\|_F^2 + \|\mathbf{U}_C \mathbf{U}_C^T \mathbf{A} (\mathbf{I}_n - \mathbf{V}_R \mathbf{V}_R^T)\|_F^2 + \|\mathbf{U}_C (\mathbf{Z}^* - \tilde{\mathbf{Z}}) \mathbf{V}_R^T\|_F^2 \\ &= \|(\mathbf{I}_m - \mathbf{U}_C \mathbf{U}_C^T) \mathbf{A} + \mathbf{U}_C \mathbf{U}_C^T \mathbf{A} (\mathbf{I}_n - \mathbf{V}_R \mathbf{V}_R^T)\|_F^2 + \|\mathbf{U}_C (\mathbf{Z}^* - \tilde{\mathbf{Z}}) \mathbf{V}_R^T\|_F^2 \\ &= \|\mathbf{A} - \mathbf{U}_C \mathbf{U}_C^T \mathbf{A} \mathbf{V}_R \mathbf{V}_R^T\|_F^2 + \|\mathbf{U}_C (\mathbf{Z}^* - \tilde{\mathbf{Z}}) \mathbf{V}_R^T\|_F^2. \end{aligned}$$

From (17) we can see that it suffices to prove the two inequalities:

$$\begin{aligned} \|\mathbf{Z}^* - \tilde{\mathbf{Z}}\|_F &\leq f_R \sqrt{h_R} + f_C \sqrt{h_C} + f_C f_R \sqrt{g_C g_R}, \\ \|\mathbf{Z}^* - \tilde{\mathbf{Z}}\|_F &\leq f_R \sqrt{h_R} + f_C \sqrt{h_C} + f_C f_R \sqrt{g_C g_R}. \end{aligned} \quad (18)$$

We left multiply both sides of $\tilde{\mathbf{Z}} = (\mathbf{S}_C^T \mathbf{U}_C)^\dagger (\mathbf{S}_C^T \mathbf{A} \mathbf{S}_R) (\mathbf{V}_R^T \mathbf{S}_R)^\dagger$ by $(\mathbf{S}_C^T \mathbf{U}_C)^T (\mathbf{S}_C^T \mathbf{U}_C)$ and right multiply by $(\mathbf{V}_R^T \mathbf{S}_R) (\mathbf{V}_R^T \mathbf{S}_R)^T$. We obtain

$$\begin{aligned} &(\mathbf{U}_C^T \mathbf{S}_C \mathbf{S}_C^T \mathbf{U}_C) \tilde{\mathbf{Z}} (\mathbf{V}_R^T \mathbf{S}_R \mathbf{S}_R^T \mathbf{V}_R) \\ &= (\mathbf{S}_C^T \mathbf{U}_C)^T (\mathbf{S}_C^T \mathbf{U}_C) (\mathbf{S}_C^T \mathbf{U}_C)^\dagger (\mathbf{S}_C^T \mathbf{A} \mathbf{S}_R) (\mathbf{V}_R^T \mathbf{S}_R) (\mathbf{V}_R^T \mathbf{S}_R) (\mathbf{V}_R^T \mathbf{S}_R)^T \\ &= (\mathbf{S}_C^T \mathbf{U}_C)^T (\mathbf{S}_C^T \mathbf{A} \mathbf{S}_R) (\mathbf{V}_R^T \mathbf{S}_R)^T \\ &= \mathbf{U}_C^T \mathbf{S}_C \mathbf{S}_C^T (\mathbf{A} - \mathbf{U}_C \mathbf{Z}^* \mathbf{V}_R^T) \mathbf{S}_R \mathbf{S}_R^T \mathbf{V}_R. \end{aligned}$$

Here the second equality follows from that $\mathbf{Y}^T \mathbf{Y} \mathbf{Y}^\dagger = \mathbf{Y}^T$ and $\mathbf{Y}^\dagger \mathbf{Y} \mathbf{Y}^T = \mathbf{Y}^\dagger$ for any \mathbf{Y} , and the last equality follows by defining $\mathbf{A}^\perp = \mathbf{A} - \mathbf{U}_C \mathbf{Z}^* \mathbf{V}_R^T$. It follows that

$$(\mathbf{U}_C^T \mathbf{S}_C \mathbf{S}_C^T \mathbf{U}_C) (\tilde{\mathbf{Z}} - \mathbf{Z}^*) (\mathbf{V}_R^T \mathbf{S}_R \mathbf{S}_R^T \mathbf{V}_R) = \mathbf{U}_C^T \mathbf{S}_C \mathbf{S}_C^T \mathbf{A}^\perp \mathbf{S}_R \mathbf{S}_R^T \mathbf{V}_R.$$

We decompose \mathbf{A}^\perp by

$$\begin{aligned}\mathbf{A}^\perp &= \mathbf{A} - \mathbf{U}_C \mathbf{U}_C^T \mathbf{A} + \mathbf{U}_C \mathbf{U}_C^T \mathbf{A} - \mathbf{U}_C \mathbf{U}_C^T \mathbf{A} \mathbf{V}_R \mathbf{V}_R^T \\ &= \mathbf{U}_C \mathbf{U}_C^T \mathbf{A} (\mathbf{I}_n - \mathbf{V}_R \mathbf{V}_R^T) + (\mathbf{I}_m - \mathbf{U}_C \mathbf{U}_C^T) \mathbf{A} \mathbf{V}_R \mathbf{V}_R^T + (\mathbf{I}_m - \mathbf{U}_C \mathbf{U}_C^T) \mathbf{A} (\mathbf{I}_n - \mathbf{V}_R \mathbf{V}_R^T).\end{aligned}$$

It follows that

$$\begin{aligned}(\mathbf{U}_C^T \mathbf{S}_C \mathbf{S}_C^T \mathbf{U}_C) (\tilde{\mathbf{Z}} - \mathbf{Z}^*) (\mathbf{V}_R^T \mathbf{S}_R \mathbf{S}_R^T \mathbf{V}_R) \\ = \mathbf{U}_C^T \mathbf{S}_C \mathbf{S}_C^T \mathbf{U}_C \mathbf{U}_C^T \mathbf{A} (\mathbf{I}_n - \mathbf{V}_R \mathbf{V}_R^T) \mathbf{S}_R \mathbf{S}_R^T \mathbf{V}_R \\ + \mathbf{U}_C^T \mathbf{S}_C \mathbf{S}_C^T (\mathbf{I}_m - \mathbf{U}_C \mathbf{U}_C^T) \mathbf{A} \mathbf{V}_R \mathbf{V}_R^T \mathbf{S}_R \mathbf{S}_R^T \mathbf{V}_R \\ + \mathbf{U}_C^T \mathbf{S}_C \mathbf{S}_C^T (\mathbf{I}_m - \mathbf{U}_C \mathbf{U}_C^T) \mathbf{A} (\mathbf{I}_n - \mathbf{V}_R \mathbf{V}_R^T) \mathbf{S}_R \mathbf{S}_R^T \mathbf{V}_R.\end{aligned}$$

and thus

$$\begin{aligned}\tilde{\mathbf{Z}} - \mathbf{Z}^* &= \mathbf{U}_C^T \mathbf{A} (\mathbf{I}_n - \mathbf{V}_R \mathbf{V}_R^T) \mathbf{S}_R \mathbf{S}_R^T \mathbf{V}_R (\mathbf{V}_R^T \mathbf{S}_R \mathbf{S}_R^T \mathbf{V}_R)^{-1} \\ &+ (\mathbf{U}_C^T \mathbf{S}_C \mathbf{S}_C^T \mathbf{U}_C)^{-1} \mathbf{U}_C^T \mathbf{S}_C \mathbf{S}_C^T (\mathbf{I}_m - \mathbf{U}_C \mathbf{U}_C^T) \mathbf{A} \mathbf{V}_R \\ &+ (\mathbf{U}_C^T \mathbf{S}_C \mathbf{S}_C^T \mathbf{U}_C)^{-1} \mathbf{U}_C^T \mathbf{S}_C \mathbf{S}_C^T (\mathbf{I}_m - \mathbf{U}_C \mathbf{U}_C^T) \mathbf{A} (\mathbf{I}_n - \mathbf{V}_R \mathbf{V}_R^T) \mathbf{S}_R \mathbf{S}_R^T \mathbf{V}_R (\mathbf{V}_R^T \mathbf{S}_R \mathbf{S}_R^T \mathbf{V}_R)^{-1}.\end{aligned}$$

It follows that

$$\begin{aligned}\|\tilde{\mathbf{Z}} - \mathbf{Z}^*\|_F &\leq \sigma_{\min}^{-1} (\mathbf{V}_R^T \mathbf{S}_R \mathbf{S}_R^T \mathbf{V}_R) \|\mathbf{A} (\mathbf{I}_n - \mathbf{V}_R \mathbf{V}_R^T) \mathbf{S}_R \mathbf{S}_R^T \mathbf{V}_R\|_F \\ &+ \sigma_{\min}^{-1} (\mathbf{U}_C^T \mathbf{S}_C \mathbf{S}_C^T \mathbf{U}_C) \|\mathbf{U}_C^T \mathbf{S}_C \mathbf{S}_C^T (\mathbf{I}_m - \mathbf{U}_C \mathbf{U}_C^T) \mathbf{A} \mathbf{V}_R\|_F \\ &+ \sigma_{\min}^{-1} (\mathbf{U}_C^T \mathbf{S}_C \mathbf{S}_C^T \mathbf{U}_C) \sigma_{\min}^{-1} (\mathbf{V}_R^T \mathbf{S}_R \mathbf{S}_R^T \mathbf{V}_R) \|\mathbf{U}_C^T \mathbf{S}_C \mathbf{S}_C^T (\mathbf{I}_m - \mathbf{U}_C \mathbf{U}_C^T) \mathbf{A} (\mathbf{I}_n - \mathbf{V}_R \mathbf{V}_R^T) \mathbf{S}_R \mathbf{S}_R^T \mathbf{V}_R\|_F.\end{aligned}$$

This proves (18) and thereby concludes the proof. \blacksquare

H.2 Proof of the Theorem

Assumption 3 assumes that the sketching matrices \mathbf{S}_C and \mathbf{S}_R satisfy the first two approximate matrix multiplication properties. Under the assumption, we obtain Lemma 25, which shows that $\tilde{\mathbf{U}}$ is nearly as good as \mathbf{U}^* in terms of objective function value.

Assumption 3 Let \mathbf{B} be any fixed matrix. Let $\mathbf{C} \in \mathbb{R}^{m \times c}$ and $\mathbf{C} = \mathbf{U}_C \mathbf{\Sigma}_C \mathbf{V}_C^T$ be the SVD.

Assume that a certain sketching matrix $\mathbf{S}_C \in \mathbb{R}^{m \times s_c}$ satisfies

$$\begin{aligned}\mathbb{P}\left\{\|\mathbf{U}_C \mathbf{S}_C \mathbf{S}_C^T \mathbf{U}_C - \mathbf{I}\|_2 \geq \frac{1}{10}\right\} &\leq \delta_1 \\ \mathbb{P}\left\{\|\mathbf{U}_C^T \mathbf{S}_C \mathbf{S}_C^T \mathbf{B} - \mathbf{U}_C^T \mathbf{B}\|_F^2 \geq \epsilon \|\mathbf{B}\|_F^2\right\} &\leq \delta_2\end{aligned}$$

for any $\delta_1, \delta_2 \in (0, 0.2)$. Let $\mathbf{R} \in \mathbb{R}^{r \times n}$ and $\mathbf{R} = \mathbf{U}_R \mathbf{\Sigma}_R \mathbf{V}_R^T$ be the SVD. Similarly, assume $\mathbf{S}_R \in \mathbb{R}^{n \times s_r}$ satisfies

$$\begin{aligned}\mathbb{P}\left\{\|\mathbf{V}_R^T \mathbf{S}_R \mathbf{S}_R^T \mathbf{V}_R - \mathbf{I}\|_2 \geq \frac{1}{10}\right\} &\leq \delta_1 \\ \mathbb{P}\left\{\|\mathbf{V}_R^T \mathbf{S}_R \mathbf{S}_R^T \mathbf{B} - \mathbf{V}_R^T \mathbf{B}\|_F^2 \geq \epsilon \|\mathbf{B}\|_F^2\right\} &\leq \delta_2.\end{aligned}$$

Lemma 25 Let $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{C} \in \mathbb{R}^{m \times c}$, and $\mathbf{R} \in \mathbb{R}^{r \times n}$ be any fixed matrices. Let \mathbf{U}^* and $\tilde{\mathbf{U}}$ be defined in (8) and (9), respectively. Let $k_c = \text{rank}(\mathbf{C})$, $k_r = \text{rank}(\mathbf{R})$, $q = \min\{m, n\}$, and $\epsilon \in (0, 1)$ be the error parameter. Assume that the sketching matrices \mathbf{S}_C and \mathbf{S}_R satisfy Assumption 3 and that $\epsilon^{-1} = o(q)$. Then

$$\|\mathbf{A} - \mathbf{C} \tilde{\mathbf{U}} \mathbf{R}\|_F^2 \leq (1 + 4\epsilon^2 q) \|\mathbf{A} - \mathbf{C} \mathbf{U}^* \mathbf{R}\|_F^2$$

holds with probability at least $1 - 2\delta_1 - 3\delta_2$.

Proof Let $f_C, f_R, h_C, h_R, g_C, g_R, g'_C, g'_R$ be defined Lemma 24. Under Assumption 3, we have that

$$\begin{aligned}f_C &\leq \frac{10}{9}, & h_C &\leq \epsilon \|\mathbf{A} - \mathbf{U}_C \mathbf{U}_C^T \mathbf{A}\|_F^2 \leq \epsilon \|\mathbf{A} - \mathbf{C} \mathbf{U}^* \mathbf{R}^T\|_F^2, \\ f_R &\leq \frac{10}{9}, & h_R &\leq \epsilon \|\mathbf{A} - \mathbf{A} \mathbf{V}_R \mathbf{V}_R^T\|_F^2 \leq \epsilon \|\mathbf{A} - \mathbf{C} \mathbf{U}^* \mathbf{R}^T\|_F^2,\end{aligned}$$

hold simultaneously with probability at least $1 - 2\delta_1 - 2\delta_2$.

We fix $\alpha = 1$, then $g_C = h_C$, and $g'_R \leq \|(\mathbf{I}_n - \mathbf{V}_R \mathbf{V}_R^T) \mathbf{S}_R \mathbf{S}_R^T \mathbf{V}_R\|_2^2$. Under Assumption 3, we have that

$$\begin{aligned}\sqrt{g'_R} &\leq \|(\mathbf{I}_n - \mathbf{V}_R \mathbf{V}_R^T) \mathbf{S}_R \mathbf{S}_R^T \mathbf{V}_R - (\mathbf{I}_n - \mathbf{V}_R \mathbf{V}_R^T) \mathbf{V}_R\|_F \\ &\leq \sqrt{\epsilon} \|(\mathbf{I}_n - \mathbf{V}_R \mathbf{V}_R^T)\|_F \leq \sqrt{\epsilon n}\end{aligned}$$

holds with probability at least $1 - \delta_2$. It follows from Lemma 24 that

$$\begin{aligned}\|\mathbf{A} - \mathbf{C} \tilde{\mathbf{U}} \mathbf{R}\|_F^2 - \|\mathbf{A} - \mathbf{C} \mathbf{U}^* \mathbf{R}\|_F^2 \\ \leq \left(f_R \sqrt{h_R} + f_C \sqrt{h_C} + f_C f_R \sqrt{g_C g'_R} \right)^2 \\ \leq \left(\frac{20}{9} \sqrt{\epsilon} \|\mathbf{A} - \mathbf{C} \mathbf{U}^* \mathbf{R}^T\|_F + \frac{10^2}{9^2} \epsilon \sqrt{n} \|\mathbf{A} - \mathbf{C} \mathbf{U}^* \mathbf{R}^T\|_F \right)^2 \\ = \frac{10^4}{9^4} \epsilon^2 n (1 + o(1)) \|\mathbf{A} - \mathbf{C} \mathbf{U}^* \mathbf{R}^T\|_F^2 \leq 4\epsilon^2 n \|\mathbf{A} - \mathbf{C} \mathbf{U}^* \mathbf{R}^T\|_F^2\end{aligned}$$

holds with probability at least $1 - 2\delta_1 - 3\delta_2$. Here the equality follows from that $\epsilon^{-1} = o(n)$.

Alternatively, if we fix $\alpha = 0$, we will obtain that

$$\|\mathbf{A} - \mathbf{C} \tilde{\mathbf{U}} \mathbf{R}\|_F^2 \leq \|\mathbf{A} - \mathbf{C} \mathbf{U}^* \mathbf{R}\|_F^2 + 4\epsilon^2 n \|\mathbf{A} - \mathbf{C} \mathbf{U}^* \mathbf{R}^T\|_F^2$$

with probability $1 - 2\delta_1 - 3\delta_2$. Therefore, if $n \leq m$, we fix $\alpha = 1$; otherwise we fix $\alpha = 0$. This concludes the proof. \blacksquare

In the following we further assume that the sketching matrices \mathbf{S}_C and \mathbf{S}_R satisfy the third approximate matrix multiplication property. Under Assumption 3 and Assumption 4, we obtain Lemma 26 which is stronger than Lemma 25.

Assumption 4 Let \mathbf{B} be any fixed matrix. Let $\mathbf{C} \in \mathbb{R}^{n \times c}$, $k_c = \text{rank}(\mathbf{C})$, and $\mathbf{C} = \mathbf{U}_C \mathbf{\Sigma}_C \mathbf{V}_C^T$ be the SVD. Assume that a certain sketching matrix $\mathbf{S}_C \in \mathbb{R}^{n \times s_c}$ satisfies

$$\mathbb{P}^f \left\{ \left\| \mathbf{U}_C^T \mathbf{S}_C \mathbf{S}_C^T \mathbf{B} - \mathbf{U}_C^T \mathbf{B} \right\|_2 \geq \epsilon \|\mathbf{B}\|_2 + \frac{\epsilon}{k_c} \|\mathbf{B}\|_F^2 \right\} \leq \delta_3$$

for any $\epsilon \in (0, 1)$ and $\delta_3 \in (0, 0.2)$. Let $\mathbf{R} \in \mathbb{R}^{r \times n}$, $k_r = \text{rank}(\mathbf{R})$, and $\mathbf{R} = \mathbf{U}_R \mathbf{\Sigma}_R \mathbf{V}_R^T$ be the SVD. Similarly, assume that $\mathbf{S}_R \in \mathbb{R}^{n \times s_r}$ satisfies

$$\mathbb{P}^f \left\{ \left\| \mathbf{V}_R^T \mathbf{S}_R \mathbf{S}_R^T \mathbf{B} - \mathbf{V}_R^T \mathbf{B} \right\|_2 \geq \epsilon \|\mathbf{B}\|_2 + \frac{\epsilon}{k_r} \|\mathbf{B}\|_F^2 \right\} \leq \delta_3.$$

Lemma 26 Let $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{C}, \mathbf{R}, \mathbf{U}^*, \tilde{\mathbf{U}}, k_c, k_r$ be defined in Lemma 25. Let $q = \min\{n, n\}$ and $\tilde{q} = \min\{m/k_c, n/k_r\}$. Assume that the sketching matrices \mathbf{S}_C and \mathbf{S}_R satisfy Assumption 3 and Assumption 4 and that $\epsilon^{-1} = o(\tilde{q})$. Then

$$\|\mathbf{A} - \mathbf{C}\tilde{\mathbf{X}}\mathbf{R}\|_F^2 \leq (1 + 4\epsilon^2 \tilde{q}) \|\mathbf{A} - \mathbf{C}\mathbf{X}^*\mathbf{R}\|_F^2$$

holds with probability at least $1 - 2\delta_1 - 2\delta_2 - \delta_3$.

Proof Let $f_C, f_R, h_C, h_R, g_C, g_R, g'_C, g'_R$ be defined Lemma 24. Under Assumption 3, we have shown in the proof of Lemma 25 that

$$\begin{aligned} f_C &\leq \frac{10}{9}, & h_C &\leq \epsilon \|\mathbf{A} - \mathbf{C}\mathbf{U}^*\mathbf{R}\|_F^2, \\ f_R &\leq \frac{10}{9}, & h_R &\leq \epsilon \|\mathbf{A} - \mathbf{C}\mathbf{U}^*\mathbf{R}\|_F^2, \end{aligned}$$

hold simultaneously with probability at least $1 - 2\delta_1 - 2\delta_2$.

We fix $\alpha = 1$, then $g_C = h_C$, and $g'_R \leq \|(\mathbf{I}_n - \mathbf{V}_R \mathbf{V}_R^T) \mathbf{S}_R \mathbf{S}_R^T \mathbf{V}_R\|_2^2$. Under Assumption 4, we have that

$$\begin{aligned} g'_R &\leq \|(\mathbf{I}_n - \mathbf{V}_R \mathbf{V}_R^T) \mathbf{S}_R \mathbf{S}_R^T \mathbf{V}_R - \underbrace{(\mathbf{I}_n - \mathbf{V}_R \mathbf{V}_R^T) \mathbf{V}_R}_= \|_2^2 \\ &\leq \|\mathbf{I}_n - \mathbf{V}_R \mathbf{V}_R^T\|_2^2 + \frac{\epsilon}{k_r} \|\mathbf{I}_n - \mathbf{V}_R \mathbf{V}_R^T\|_F^2 \leq \epsilon + \frac{\epsilon(n - k_r)}{k_r} = \frac{\epsilon n}{k_r} \end{aligned}$$

holds with probability at least $1 - \delta_3$. It follows from Lemma 24 that

$$\begin{aligned} \|\mathbf{A} - \mathbf{C}\tilde{\mathbf{U}}\mathbf{R}\|_F^2 - \|\mathbf{A} - \mathbf{C}\mathbf{U}^*\mathbf{R}\|_F^2 &\leq \left(f_R \sqrt{h_R} + f_C \sqrt{h_C} + f_C f_R \sqrt{g_C g_R} \right)^2 \\ &\leq \left(\frac{20}{9} \sqrt{\epsilon} \|\mathbf{A} - \mathbf{C}\mathbf{U}^*\mathbf{R}\|_F + \frac{10^2}{9^2} \epsilon \sqrt{n/k_r} \|\mathbf{A} - \mathbf{C}\mathbf{U}^*\mathbf{R}\|_F \right)^2 \\ &= \frac{10^4}{9^4} \epsilon^2 n k_r^{-1} (1 + o(1)) \|\mathbf{A} - \mathbf{C}\mathbf{U}^*\mathbf{R}\|_F^2 \leq 4\epsilon^2 n k_r^{-1} \|\mathbf{A} - \mathbf{C}\mathbf{U}^*\mathbf{R}\|_F^2 \end{aligned}$$

holds with probability at least $1 - 2\delta_1 - 2\delta_2 - \delta_3$. Here the equality follows from that $\epsilon^{-1} = o(n/k_r)$.

Analogously, by fixing $\alpha = 0$ and assuming $\epsilon^{-1} = o(m/k_c)$, we can show that

$$\|\mathbf{A} - \mathbf{C}\tilde{\mathbf{U}}\mathbf{R}\|_F^2 - \|\mathbf{A} - \mathbf{C}\mathbf{U}^*\mathbf{R}\|_F^2 \leq 4\epsilon^2 m k_c^{-1} \|\mathbf{A} - \mathbf{C}\mathbf{U}^*\mathbf{R}\|_F^2$$

holds with probability at least $1 - 2\delta_1 - 2\delta_2 - \delta_3$. This concludes the proof. \blacksquare

Finally, we prove Theorem 9 using Lemma 25 and Lemma 26.

For leverage score sampling, uniform sampling, and count sketch, Assumption 3 is satisfied. Then the bound follows by setting $\epsilon = 0.5\sqrt{\epsilon'}/\tilde{q}$ and applying Lemma 25. Here $q = \min\{m, n\}$. For the three sketching methods, we set $\delta_1 = 0.01$ and $\delta_2 = 0.093$.

For Gaussian projection and SRHT, Assumption 3 and Assumption 4 are satisfied. Then the bound follows by setting $\epsilon = 0.5\sqrt{\epsilon'}/\tilde{q}$ and applying Lemma 26. Here $\tilde{q} = \min\{m/k_c, n/k_r\}$. For Gaussian projection, we set $\delta_1 = 0.01$, $\delta_2 = 0.09$, and $\delta_3 = 0.1$. For SRHT, we set $\delta_1 = 0.02$, $\delta_2 = 0.08$, and $\delta_3 = 0.1$.

References

- Adi Ben-Israel and Thomas N.E. Greville. *Generalized Inverses: Theory and Applications. Second Edition*. Springer, 2003.
- Christos Boutsidis and David P. Woodruff. Optimal CUR matrix decompositions. *arXiv preprint arXiv:1405.7910*, 2014.
- Christos Boutsidis, Petros Drineas, and Malik Magdon-Ismael. Near-optimal column-based matrix reconstruction. *SIAM Journal on Computing*, 43(2):687-717, 2014.
- Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. *Theoretical Computer Science*, 312(1):3-15, 2004.
- Kenneth L. Clarkson and David P. Woodruff. Low rank approximation and regression in input sparsity time. In *Annual ACM Symposium on theory of computing (STOC)*. ACM, 2013.
- Michael B Cohen, Jelani Nelson, and David P Woodruff. Optimal approximate matrix product in terms of stable rank. *arXiv preprint arXiv:1507.02268*, 2015.
- Bo Dai, Bo Xie, Niao He, Yingyu Liang, Anant Raj, Maria-Florina F Balcan, and Le Song. Scalable kernel methods via doubly stochastic gradients. In *Neural Information Processing Systems (NIPS)*. 2014.
- Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Sampling algorithms for ℓ_2 regression and applications. In *the 17th Annual ACM-SIAM Symposium On Discrete Algorithms (SODA)*, 2006.
- Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 30(2):844-881, September 2008.

- Petros Drineas, Michael W. Mahoney, S. Muthukrishnan, and Tamás Sarlós. Faster least squares approximation. *Numerische Mathematik*, 117(2):219–249, 2011.
- Charles Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the Nystrom method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):214–225, 2004.
- Alex Gittens. The spectral norm error of the naive Nystrom extension. *arXiv preprint arXiv:1110.5305*, 2011.
- Alex Gittens and Michael W. Mahoney. Revisiting the Nystrom method for improved large-scale machine learning. *Journal of Machine Learning Research*, 17(117):1–65, 2016.
- Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.
- Cho-Jui Hsieh, Si Si, and Inderjit S Dhillon. Fast prediction for large-scale kernel machines. In *Neural Information Processing Systems (NIPS)*. 2014.
- William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary mathematics*, 26(189-206), 1984.
- Sanjiv Kumar, Mehryar Mohri, and Amreet Talwaker. On sampling-based approximate spectral decomposition. In *International Conference on Machine Learning (ICML)*, 2009.
- Sanjiv Kumar, Mehryar Mohri, and Amreet Talwaker. Sampling methods for the Nystrom method. *Journal of Machine Learning Research*, 13:981–1006, 2012.
- Yichao Lu, Paraneer Dhillon, Dean P Foster, and Lyle Ungar. Faster ridge regression via the subsampled randomized Hadamard transform. In *Neural Information Processing Systems (NIPS)*. 2013.
- Michael W. Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 3(2):123–224, 2011.
- Michael W. Mahoney and Petros Drineas. CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702, 2009.
- Xiangrui Meng and Michael W Mahoney. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *the 45th Annual ACM Symposium on Theory Of Computing (STOC)*, 2013.
- John Nelson and Huy L Ngyen. OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS)*, 2013.
- Rob Patro and Carl Kingsford. Global network alignment using multiscale spectral signatures. *Bioinformatics*, 28(23):3105–3114, 2012.
- Ninh Pham and Rasmus Pagh. Fast and scalable polynomial kernels via explicit feature maps. In *the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2013.
- Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Neural Information Processing Systems (NIPS)*, pages 1177–1184, 2007.
- Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.
- Donghyuk Shin, Si Si, and Inderjit S Dhillon. Multi-scale link prediction. In *International Conference on Information and Knowledge Management (CIKM)*. ACM, 2012.
- Si Si, Cho-Jui Hsieh, and Inderjit Dhillon. Memory efficient kernel approximation. In *International Conference on Machine Learning (ICML)*, pages 701–709, 2014a.
- Si Si, Donghyuk Shin, Inderjit S Dhillon, and Beresford N Parlett. Multi-scale spectral decomposition of massive graphs. In *Neural Information Processing Systems (NIPS)*. 2014b.
- G. W. Stewart. Four algorithms for the efficient computation of truncated pivoted QR approximations to a sparse matrix. *Numerische Mathematik*, 83(2):313–323, 1999.
- Amreet Talwaker and Afshin Rostamizadeh. Matrix coherence and the Nystrom method. *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2010.
- Mikkel Thorup and Yin Zhang. Tabulation-based 5-independent hashing with applications to linear probing and second moment estimation. *SIAM Journal on Computing*, 41(2): 293–331, April 2012. ISSN 0097-5397.
- Joel A Tropp. Improved analysis of the subsampled randomized hadamard transform. *Advances in Adaptive Data Analysis*, 3(01n02):115–126, 2011.
- Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.
- Shusen Wang and Zhihua Zhang. Improving CUR matrix decomposition and the Nystrom approximation via adaptive sampling. *Journal of Machine Learning Research*, 14:2729–2769, 2013.
- Shusen Wang, Chao Zhang, Hui Qian, and Zhihua Zhang. Improving the modified Nystrom method using spectral shifting. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, 2014.
- Shusen Wang, Luo Luo, and Zhihua Zhang. SPSPD matrix approximation via column selection: theories, algorithms, and extensions. *Journal of Machine Learning Research*, 17(49):1–49, 2016.
- Kilian Weinberger, Amrban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. Feature hashing for large scale multitask learning. In *International Conference on Machine Learning (ICML)*, 2009.

- Christopher Williams and Matthias Seeger. Using the Nystrom method to speed up kernel machines. In *Neural Information Processing Systems (NIPS)*, 2001.
- David P Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1-2):1–157, 2014.
- Tianbao Yang, Yi-Feng Li, Mehrdad Mahdavi, Rong Jin, and Zhi-Hua Zhou. Nystrom method vs random fourier features: A theoretical and empirical comparison. In *Neural Information Processing Systems (NIPS)*, 2012.
- Kai Zhang and James T. Kwok. Clustered Nystrom method for large scale manifold learning and dimension reduction. *IEEE Transactions on Neural Networks*, 21(10):1576–1587, 2010.

Multi-Objective Markov Decision Processes for Data-Driven Decision Support

Daniel J. Lizotte

*Department of Computer Science, Department of Epidemiology & Biostatistics
The University of Western Ontario
1151 Richmond Street
London, ON N6A 3K7
Canada*

DLIZOTTE@UWO.CA

Eric B. Laber

*Department of Statistics
North Carolina State University
Raleigh, NC 27695
USA*

LABER@STAT.NCSU.EDU

Editor: Benjamin M. Marlin, C. David Page, and Suchi Saria

Abstract

We present new methodology based on Multi-Objective Markov Decision Processes for developing sequential decision support systems from data. Our approach uses sequential decision-making data to provide support that is useful to many different decision-makers, each with different, potentially time-varying preference. To accomplish this, we develop an extension of fitted- Q iteration for multiple objectives that computes policies for all scalarization functions, i.e. preference functions, simultaneously from continuous-state, finite-horizon data. We identify and address several conceptual and computational challenges along the way, and we introduce a new solution concept that is appropriate when different actions have similar expected outcomes. Finally, we demonstrate an application of our method using data from the Clinical Antipsychotic Trials of Intervention Effectiveness and show that our approach offers decision-makers increased choice by a larger class of optimal policies.

Keywords: multi-objective optimization, reinforcement learning, Markov decision processes, clinical decision support, evidence-based medicine

1. Introduction

Markov Decision Processes (MDPs) (Bertsekas and Tsitsiklis, 1996) provide a framework for reasoning about the actions of an autonomous decision-making agent in an environment as it strives to achieve long-term success. Operating within this framework, reinforcement learning (RL) methods for finding optimal actions in MDPs hold great promise for using vast amounts of accumulating longitudinal data to help humans make better-informed decisions. Batch reinforcement learning methods, including fitted Q -learning (Ernst et al., 2005), A -learning (Blatt et al., 2004), and regret regression (Henderson et al., 2010), are already being used to aid decision-making in diverse areas including medicine (Alagoz et al., 2010; Shortreed et al., 2011; Burnside et al., 2012), ecology (Păduraru et al., 2012), intelligent

tutoring systems (Brunskill and Russell, 2011), and water reservoir control (Castelletti et al., 2010). Although headway has been made in these application areas, progress is hampered by the fact that many sequential decision *support* problems are not modelled well by MDPs.

One reason for this is that in most cases, human action selection is driven by multiple competing objectives; for example, a medical decision will be based not only on the effectiveness of a treatment, but also on its potential side-effects, cost, and other considerations. Because the relative importance of these objectives varies from user to user, the quality of a policy is not well captured by a universal single scalar “reward” or “value.” Multi-Objective Markov Decision Processes (MOMDPs) accommodate this by allowing vector-valued rewards (Rojers et al., 2013) and proposing an application-dependent *solution concept*. A solution concept is essentially a partial order on policies; the set of policies that are maximal according to the partial order are considered “optimal” and are indistinguishable under that solution concept. Depending on the application, a single policy may be selected from among these, or a set of policies may be presented in some way. Computing and presenting a set of policies is termed the *decision support* setting by Roijers et al. and is the setting we consider here.

2. Existing Methods and Our Contributions

Rojers et al. (2013) note that, “...there are currently no methods for learning multiple policies with non-linear [preferences] using a value-function approach.” We present a method that fills this gap, and that additionally uses value function approximation to accommodate continuous state features, thus allowing us to use the MOMDP framework to analyze continuous-valued sequential data. Previous work (Lizotte et al., 2012) on this problem computes a set of policies based on the assumptions that i) end-users have a “true reward function” that is linear in the objectives and ii) all future actions will be chosen optimally with respect to the same “true reward function” over time. Our new method relaxes both of these assumptions as it allows the decision-maker to revisit action selection at each decision point in light of new information, both about state and about their own preferences and priorities over different outcomes of interest. Therefore, the proposed method can accommodate changes in preference over time while still making optimal decisions according to our new solution concepts by introducing the *non-deterministic multi-objective fitted- Q* algorithm, which computes policies for all scalarization functions, i.e., preference functions, simultaneously from continuous-state, finite-horizon data. This allows us to present a greater variety of action choices by acknowledging that preference functions may be non-linear. We then present the vector-valued expected returns associated with the different policies in order to provide decision support without having to refer to any particular scalarization function. Showing the expected returns in the original reward space allows us to more easily understand the qualitative differences between action choices. Although decision support is important in many application areas, we are motivated by clinical decision-making; therefore we demonstrate the use of our algorithm using data from the Clinical Antipsychotic Trials of Intervention Effectiveness (CATIE).

Simplified versions of some of our ideas were presented in a shorter paper by Laber et al. (2014a), but we treat the problem in its full generality here. In particular, our work goes beyond “Set-Valued Dynamic Treatment Regimes” in four significant ways:

- We introduce a complete *non-deterministic fitted-Q* algorithm that is applicable to arbitrary numbers of actions and arbitrary time horizons. (Previous work was limited to binary actions and maximum two decision points.) This allows us to perform fitted-Q backups in general settings using multiple reward functions over continuous-valued state features.
- We prove that our algorithm finds all policies that are optimal for some scalarization function by considering a collection of policies at the next time step that is only polynomial in the data set size.
- We formalize a solution concept, *practical domination*, that is more flexible than Pareto domination for identifying whether an action is not desirable. A similar concept was introduced in previous work (Laber et al., 2014a), but we show that using practical domination, while useful, is problematic for more than two decision points because it does not induce a partial order on actions. However, we show that a modification of practical domination leads to a partial ordering for any number of actions or time points.
- We demonstrate the use of our algorithm on the Clinical Antipsychotic Trials of Intervention Effectiveness (CATIE) and we compare our approach quantitatively and qualitatively with a competing approach derived from previous work of Lizotte et al. (2010, 2012).

3. Motivation

Our work is motivated by a clear opportunity for reinforcement learning methods to provide novel ways of analyzing data to produce high-quality, evidence-based decision support. We briefly review some specific applications here where we believe our approach could be particularly relevant.

3.1 Intelligent Tutorial Systems

Brunskill and Russell (2011), and Rafferty et al. (2011) study the automatic construction of adaptive pedagogical strategies for intelligent tutoring systems. They employ POMDP models to capture the partially observable and sequential aspect of this problem, using hidden state to represent a student’s knowledge. Their approach uses time taken to learn all skills as a cost, i.e., negative reward, that drives teaching action selection. Chi et al. (2011) use an MDP formulation and use “Normalized Learning Gain,” a quantification of skill acquisition, as a reward; however, they do not explicitly consider time spent. The ability to consider both of these rewards simultaneously would empower the learner or the teacher to emphasize one or the other over the course of their interaction with the system. The method we present could offer a selection of teaching actions that are all optimal for different preferences over these rewards, and possibly others as well.

3.2 Computational Sustainability

Paduraru et al. (2012) identify an application within the domain of sustainable wildlife management where the MDP framework is particularly appropriate. They investigate the

efficacy of several off-policy methods for developing control policies for mallard duck populations. Their output, rather than providing autonomous control, is intended to provide decision support for public environmental policy-makers. They use “number of birds harvested per year” as the reward. However, in practical management plans, several outcomes may be of interest including minimum population size, program cost, and so on. Because formulating a (e.g. linear) trade-off among these rewards would be difficult, our method is relevant to this problem.

3.3 Treating Chronic Disease

Reinforcement learning has also been used as a means of analyzing sequential medical data to inform clinical decision-makers of the comparative effectiveness of different treatments (Shortreed et al., 2011). RL methods are suited to decision support for treating chronic illness where a good *policy* for choosing treatments over time is crucial for success. Indeed, optimal policies—known as “Dynamic Treatment Regimes” in statistics and the behavioral sciences—have been learned for the management of chronic conditions including attention deficit hyperactivity disorder (Laber et al., 2014b), HIV infection (Moodie et al., 2007), and smoking addiction (Strecher et al., 2006). They have also been applied to sequences of diagnostics as well, for example in breast cancer (Burnside et al., 2012). We present a case study in this domain in Section 7.

4. Background

We introduce a new approach for solving Multi-Objective Markov Decision Processes with the goal of providing data-driven decision support. Our approach uses non-deterministic policies to encode the set of all non-dominated policies. In this section, we review the most relevant existing literature on MOMDPs and NDPs.

4.1 Multi-objective Optimization and MOMDPs

The most basic definition of a Markov Decision Process is as a 4-tuple $\langle S, \mathcal{A}, P, R \rangle$ where S is a set of states, \mathcal{A} is a set of actions, $P(s, a, s') = \Pr(s'|s, a)$ gives the probability of a state transition given action and current state, and $R(s, a)$ is the immediate scalar reward obtained in state s when taking action a . One common goal of “solving” an MDP, if we assume a finite time horizon of T steps, is to find a policy $\pi : S \rightarrow \mathcal{A}$ that maximizes

$$V^\pi(s) = \mathbb{E}^\pi \left[\sum_{t=1}^T R(s_t, a_t) \mid s_1 = s \right]$$

pointwise for all states. In the preceding, \mathbb{E}^π indicates that the expectation is taken assuming the state-action trajectories are obtained by following policy π . Because in the finite-horizon setting that the optimal π is in general non-stationary (Bertsekas, 2007), we define π to be a sequence of functions π_t for $t \in \{1, \dots, T\}$, where $\pi_t : S_t \rightarrow \mathcal{A}_t$.

Like previous work by Lizotte et al. (2010, 2012) and by many others (Roijfers et al., 2013), we focus on the setting where the definition of an MDP is augmented by assuming a D -dimensional reward vector $\mathbf{R}(s_t, a_t)$ is observed at each time step. We define a finite-horizon MOMDP with finite time horizon T as a tuple of state spaces S_t , action spaces \mathcal{A}_t ,

state transition functions $P_t : \mathcal{S}_t \times \mathcal{A}_t \rightarrow \mathbb{P}(\mathcal{S}_{t+1})$ where $\mathbb{P}(\mathcal{S}_{t+1})$ is the space of probability measures on \mathcal{S}_{t+1} , and reward functions $\mathbf{R}_t : \mathcal{S}_t \times \mathcal{A}_t \rightarrow \mathbb{R}^D$ for $t \in \{1, \dots, T\}$. In keeping with the Markov assumption, both \mathbf{R}_t and P_t depend only on the current state and action. In this work we assume finite action sets, but we do *not* assume that state spaces are finite. The *value* of a policy π is then given by

$$\mathbf{V}^\pi(s) = \mathbb{E}^\pi \left[\sum_{t=1}^T \mathbf{R}^t(s_t, a_t) \mid s_1 = s \right] \quad (1)$$

which is the expected sum of (vector-valued) rewards we achieve by following policy π .

Just as “solving” an MDP is an optimization problem (i.e. we want the optimal value function or policy), “solving” a MOMDP is a *multi-objective optimization* (MOO) problem. Whereas in typical scalar optimization problems having a unique solution is viewed as typical or at least desirable, in the MOO setting, the most common goal is to produce a *set* of solutions that are *non-dominated*.

Definition 1 (Non-dominated a.k.a. Pareto optimal solutions) *Let \mathcal{X} be the set of all feasible inputs to a multi-objective optimization problem with objective $\mathbf{f}(x)$. Let \mathcal{Y} be the range of \mathbf{f} on \mathcal{X} . (In the RL context, one can think of \mathcal{X} as the set of all possible policies starting from a given state, and \mathcal{Y} as their corresponding values, which are vectors in this case.) A solution vector $\mathbf{y} \in \mathcal{Y}$ is non-dominated if $\nexists \mathbf{y}' \in \mathcal{Y}$ s.t. $\forall i, y'_i \geq y_i$ and $\exists j, y'_j > y_j$. A preimage $\mathbf{x} \in \mathcal{X}$ of such a \mathbf{y} is sometimes called an efficient solution, but we will also refer to such inputs as non-dominated.*

A common goal in MOO is to find *all* of the non-dominated solutions (Miettinen, 1999; Ehrgott, 2005). Some work on MOMDPs has this same goal (Perry and Weng, 2010). One approach to finding non-dominated solutions of a MOO problem is to solve a set of optimization problems that are *scalarized* versions of the MOO. A *scalarization function* ρ is chosen which maps vector-valued outcomes scalars and then one solves

$$\max_{x \in \mathcal{X}} \rho(\mathbf{f}(x)),$$

the scalar optimization defined by composing the vector-valued outcome function with the scalarization function. If a specific “correct” scalarization function is fixed and known, we can simply apply it to all outcome vectors and reduce our MOO problem to a scalar optimization problem (and arguably we never had a MOO problem to begin with.) Otherwise, we may seek solutions to scalarized problems for all ρ belonging to some function class. We assume that any ρ of interest is non-decreasing along each dimension, that is, it is always preferable to increase a reward dimension, all else being equal. It is well-known (Miettinen, 1999) that the set of Pareto-optimal solutions corresponds to the set of all solutions that are optimal for some scalarization function; we will use these two views of optimality as we construct our algorithm.

Previous work by Lizotte et al. (2012) uses dynamic programming to compute policies for all possible scalar rewards that are a convex combination of the basis rewards, using Q -functions learned by linear regression. Thus the output produces the optimal policies for all ρ such that $\rho(\mathbf{r}) = \mathbf{r}^\top \mathbf{w}$, $w_d > 0$, $\sum_d w_d = 1$. Each convex combination is interpreted

as a preference describing the relative importance of the the different basis rewards, and the method is used to show how preference relates to optimal action choice. This gives a new and potentially useful way of visualizing the connection between preference and action choice, but there are drawbacks to the approach. First, one must assume that the convex combination is fixed for all time points—that is, preferences do not change over time. This assumption enables dynamic programming to work, but is not reasonable for some applications, particularly in clinical decision-making where a patient’s first-hand experience with a treatment may influence subsequent preferences for symptom versus side-effect reduction. Second, the method is overly eager to eliminate actions. Consider two actions a_1 and a_2 that are extreme, e.g. a_1 has excellent efficacy but terrible side-effects and a_2 has no side-effects but poor efficacy. These could eliminate a third action a_3 that is moderately good according to both rewards. An example of this situation is illustrated in Figure 1 (a), which shows that the actions chosen by this method are restricted to the convex hull of the Pareto frontier, rather than the entire frontier. In this circumstance where a_3 is qualitatively very different from both a_1 and a_2 , we argue that a decision support system should suggest *all three* treatments and thereby allow the decision-maker to make the final choice based on her expertise. The third drawback of using this approach is that it is limited to ordinary least squares regression, which may not work well for data with non-Gaussian errors, e.g., a binary terminal reward.

Rather than use the method of Lizotte et al., we will instead base our method on assessing actions and policies using a *partial order* on their vector of Q -values. Perhaps the most common partial order on vectors comes from the notion of *Pareto-optimality* (Vamplew et al., 2011). For example, an action a is *Pareto-optimal* at a state st if $\forall (d, d') Q_{T|d}(st, a) \geq Q_{T|d'}(st, a')$. We will show in Section 5 that for $t < T$, the problem of deciding which actions are optimal is more complex, but we will still leverage the idea of a partial order. The problem of identifying Pareto-optimal policies is of significant interest in RL (Perry and Weng, 2010; Vamplew et al., 2009) and is closely related to what we wish to accomplish. Basing our work on the Pareto-optimal approach rather than on the previous work of Lizotte et al. avoids assuming that preferences are fixed over time, and it avoids the problem of “extreme” actions eliminating “moderate” ones. Furthermore, our approach works with a larger class of regression models, including ordinary least squares, the lasso, support vector regression, and logistic regression. While our Pareto-based approach makes these three improvements, using Pareto-optimality can *still* result in actions being eliminated unnecessarily; this is illustrated in Figure 1(b). We address this problem by introducing an alternative notion of domination in Section 6. Each of these contributions leads to increased action choice for the decision-maker by considering a larger class of preferences over reward vectors.

4.2 Non-Deterministic Policies

Milani Fard and Pineau (2011) describe non-deterministic policies for Markov Decision Processes (MDPs) with a finite state space and a single reward function. The term *non-deterministic* is used as in the study of *non-deterministic finite automata* and indicates that there are choices made as the system evolves about which we assume we have no

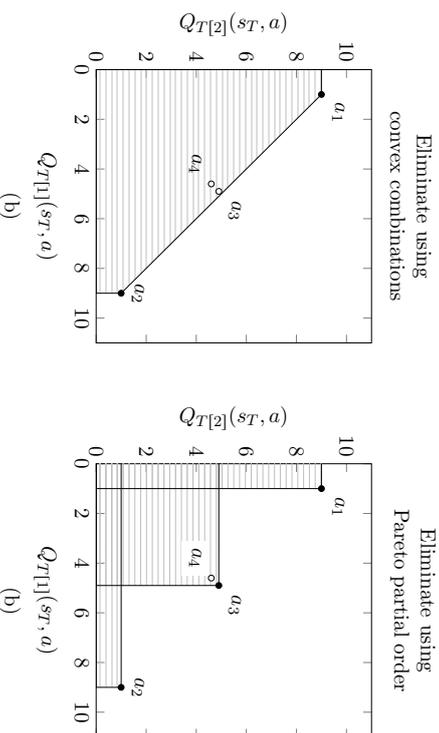


Figure 1: Comparison of existing approaches to eliminating actions at time T . The problems illustrated here have analogs for $t < T$ where the picture is more complicated. In this simple example, we suppose the vector-valued expected rewards ($Q_{T[1]}(s_T, a), Q_{T[2]}(s_T, a)$) are (1, 9), (9, 1), (4, 9, 4, 9), (4, 6, 4, 6) for actions a_1, a_2, a_3, a_4 , respectively. **Figure 1(a)**: Using the method of Lizotte et al. (2010, 2012) based on convex combinations of rewards, actions a_3 and a_4 would be eliminated, and we would have $\Pi_T(s_t) = \{a_1, a_2\}$. (Any action whose expected rewards fall in the shaded region would be eliminated.) However, we would prefer to at least include a_3 since it offers a more “moderate” outcome that may be important to some decision-makers. **Figure 1(b)**: Using the Pareto partial order, only action a_4 is eliminated, and we have $\Pi_T(s_T) = \{a_1, a_2, a_3\}$. However, we may prefer to include a_4 since its performance is very close to that of a_3 , and may be preferable for reasons we cannot infer from our data—e.g. cost, or allergy to a_3 .

information.¹ Given an MDP with *state space* S and an *action set* A , an NDP Π is a map from the state space to the set $2^A \setminus \{\emptyset\}$. Milani Fard and Pineau assume that a *user* operating the MDP will, at each timestep, choose an action from the set $\Pi(s)$. They are motivated by the same considerations that we are in the sense that they wish to provide choice to the user while still achieving good performance; thus, they only eliminate actions that are clearly sub-optimal. Because they consider only a single reward function, they can measure performance using the expected discounted infinite sum of future (scalar) rewards in the usual way, and they can produce an NDP Π that has near-optimal performance even if the user chooses the “worst” actions from $\Pi(s)$ in each state.

One can view the NDP as a compact way of expressing a set of policies that might be executed. Suppose that $\#A = |\Pi(s)|$, the number of actions provided by the NDP Π , is the same at all states. Then the number of policies that are *consistent* with Π , that is, the policies for which $\pi(s) \in \Pi(s)$, is $\#A^{|\mathcal{S}|}$. So the NDP Π is a compact encoding of an exponential number of policies. We will make use of this property to encode our policies. The two most important differences between our work and that of Milani Fard and Pineau are that our motivation for learning non-deterministic policies is driven explicitly by having more than one basis reward of interest, and that we use more general value function models rather than a tabular representation. Having multiple basis rewards combined with value function approximation leads us to a different, novel algorithm for learning NDPs.

5. Fitted- Q for MOMDPs

Our non-deterministic fitted- Q algorithm for multiple objectives uses finite-horizon, batch data. We present a version that uses linear value function approximation because this model is commonly used by statisticians working in clinical decision support (Strecher et al., 2006; Lizotte et al., 2010, 2012; Laber et al., 2014b), and because available data often contain continuous-valued features, e.g., symptom and side-effect levels, laboratory values, etc., and outcomes, e.g., symptom scores, body mass index. It is a flexible model because we will not restrict the state features one might use. For learning, we assume a batch of n data trajectories of the form

$$s^i_1, a^i_1, r^i_1[1], \dots, r^i_1[D]; s^i_2, a^i_2, r^i_2[1], \dots, r^i_2[D]; \dots; s^i_T, a^i_T, r^i_T[1], \dots, r^i_T[D] \text{ for } i = 1, \dots, n.$$

In the following exposition, we begin by specifying how the algorithm works for the last time point $t = T$. This would be the only step needed in a “non-sequential” decision problem. We then describe the steps analogous to the fitted- Q “backup” operation for earlier timepoints $t < T$, which are more complex.

5.1. Final time point, $t = T$

At time T , we define the approximate Q -function for reward dimension d as the linear least squares fit

$$\hat{Q}_{T[d]}(s_T, a_T) = \phi_T(s_T, a_T)^\top \hat{w}_{T[d]}, \quad \hat{w}_{T[d]} = \arg \min_w \sum_i \left(\phi_T(s^i_T, a^i_T)^\top w - r^i_T[a] \right)^2 \quad (2)$$

1. Note that *non-deterministic* does not mean “stochastic”; i.e., we do not suppose a known stationary random policy will be followed.

giving the estimated vector-valued expected reward function

$$\hat{\mathbf{Q}}_T(s_T, a_T) = (\hat{Q}_{T[1]}(s_T, a_T), \dots, \hat{Q}_{T[D]}(s_T, a_T))^T. \quad (3)$$

Here, $\phi_T(s_T, a_T)$ is a feature vector of state and action. As discussed by Lizotte et al. (2012), $\phi_T(s_T, a_T)$ would typically include: a constant component for the intercept, features describing s_T , dummy variables encoding the discrete action a_T , and the product of the dummy variables with the features describing s_T (Cook and Weisberg, 1999). One could also include other non-linear functions of s_T and a_T as features if desired. We present our method assuming that $\tilde{\mathbf{w}}_{T[d]}$ are found by least squares regression, but one could for example add an L_1 penalty, or use support vector regression (Hastie et al., 2001). Furthermore, unlike previous work by Lizotte et al. (2012), any Generalized Linear Model (GLM) with a monotonic increasing link function (e.g. logistic regression, Poisson regression, and so on) can also be used (Cook and Weisberg, 1999). Note that we can recover a ‘‘tabular’’ representation if the states are discrete and we assign mutually orthogonal feature vectors to each one.

Having obtained the $\hat{\mathbf{Q}}_T$ from (2), we construct an NDP Π_T that will give, for each state, the actions one might take at the last time point. For each state s_T at the last time point, each action a_T is associated with a *unique* vector-valued estimated expected reward given by $\mathbf{Q}_T(s_T, a_T)$. Thus, we decide which among these vectors is a desirable outcome, and include their associated actions in $\Pi_T(s_T)$. Our main focus will be to construct $\Pi_T(s_T)$ for each state based on the multi-objective criterion of Pareto optimality; however, an important advantage of our algorithm is that it can use other definitions of Π_T as well; we discuss an extension in Section 6. For example, different definitions of Π_T allow us to recover other varieties of Q -learning:

Scalar Fitted-Q: Defining

$$\Pi_T(s_T) = \{\arg \max_a \hat{Q}_{T[0]}(s_T, a)\}$$

gives standard fitted- Q applied to reward dimension 0.

Convex Pareto-optimal Fitted-Q: Defining

$$\Pi_T(s) = \{a : \exists \theta (\theta \geq \mathbf{0} \wedge \theta^T \mathbf{1} = 1 \wedge \forall a' \hat{\mathbf{Q}}_T(s_T, a)^T \theta \geq \hat{\mathbf{Q}}_T(s_T, a')^T \theta)\}$$

includes those actions whose expected reward is on the convex hull of the Pareto frontier; these are the actions that would be included by the previous method of Lizotte et al. (2012).

Pareto-optimal Fitted-Q: Defining

$$\Pi_T(s) = \{a : \nexists a' (\forall d \hat{Q}_{T[d]}(s_T, a) < \hat{Q}_{T[d]}(s_T, a'))\}$$

includes precisely those actions whose expected reward is on the (weak) Pareto frontier; this is a superset of those included by the method of Lizotte et al. (2012). It includes the actions that are optimal for some scalarization function, which is our action set of interest.

5.2 Earlier time points, $t < T$

For $t < T$, it is only possible to define the expected return of taking an action in a given state by also deciding which particular policy will be followed to choose future actions. In standard fitted- Q , for example, one assumes that the future policy is given by $\pi_t(s) = \arg \max_a \hat{Q}_j(s, a)$ for all $j > t$. In the non-deterministic setting, we may know that the future policy belongs to some set of possible policies derived from Π_j for $j > t$, but in general we do not know which among that set will be chosen; therefore, we explicitly include the dependence of $\hat{\mathbf{Q}}_t$ on the choice of future policies π_j , $t < j \leq T$:

$$\hat{\mathbf{Q}}_t(s_t, a_t; \pi_{t+1}, \dots, \pi_T) = \left\{ \hat{Q}_{t[1]}(s_t, a_t; \pi_{t+1}, \dots, \pi_T), \dots, \hat{Q}_{t[D]}(s_t, a_t; \pi_{t+1}, \dots, \pi_T) \right\}^T$$

where

$$\text{for } d = 1, \dots, D, \quad \hat{Q}_{t[d]}(s_t, a_t; \pi_{t+1}, \dots, \pi_T) = \phi_t(s_t, a_t)^T \tilde{\mathbf{w}}_{t[d]|\pi_{t+1}, \dots, \pi_T},$$

and

$$\tilde{\mathbf{w}}_{t[d]|\pi_{t+1}, \dots, \pi_T} = \arg \min_{\mathbf{w}} \sum_{i=1}^n \left[\phi_t(s_t^i, a_t^i)^T \mathbf{w} - \left\{ r_{t[d]}^i + \hat{Q}_{t+1[d]}(s_{t+1}^i, \pi_{t+1}(s_{t+1}^i); \pi_{t+2}, \dots, \pi_T) \right\} \right]^2. \quad (4)$$

We say an expected return is *achievable* if it can be obtained by taking some immediate action in the current state and following it with a fixed sequence of policies until we reach the last time point.

We use \mathcal{Q}_t to denote a set of partially-evaluated Q -functions; each member of \mathcal{Q}_t is a function of s_t and a_t only and assumes a particular fixed sequence π_{t+1}, \dots, π_T of future policies. Precisely which future policies should be considered is the subject of the next section. For the last time point, we define $\mathcal{Q}_T = \{\hat{\mathbf{Q}}_T\}$, the set containing the single (multivariate) Q -function for the last time point. Figure 2 is a visualization of an example \mathcal{Q}_{T-1} where each function in the set is evaluated at the same given state and for each of the five available actions, $\{\blacktriangleright, \blacksquare, \blacktriangleleft, \blacktriangleright\}$. Thus, each element of the example \mathcal{Q}_{T-1} corresponds to a collection of five markers on the plot, one for the expected return for each action, assuming we follow a particular π_T . The question of what collection of π_T we should consider is the subject of the next section.

5.3 Constructing Π_t from Π_{t+1}

We now describe the ‘‘backup’’ step that constructs Π_t and \mathcal{Q}_t from Π_{t+1} and \mathcal{Q}_{t+1} . A member of \mathcal{Q}_t is constructed from data using equation (4) by choosing two components: An element of \mathcal{Q}_{t+1} (with its implicit choice of π_{t+2} through π_T) and a policy π_{t+1} . When considering different possible π_{t+1} , we restrict our attention to policies that i) are *consistent* with Π_{t+1} , and ii) are *representable* using the approximation space chosen for $\hat{\mathbf{Q}}_{t+1}$. In the following, we define these notions of consistency and representability, argue that this subset of policies contains all those we need to consider, and show how the set of consistent and representable policies can be efficiently enumerated using mixed integer linear programming.

To construct \mathcal{Q}_t , we will only consider future policies that are consistent with the NDPs we have already learned for later time points. As described above, each $\Pi(s)$ contains each action for which some scalarization function (i.e. preference) prefers that action.

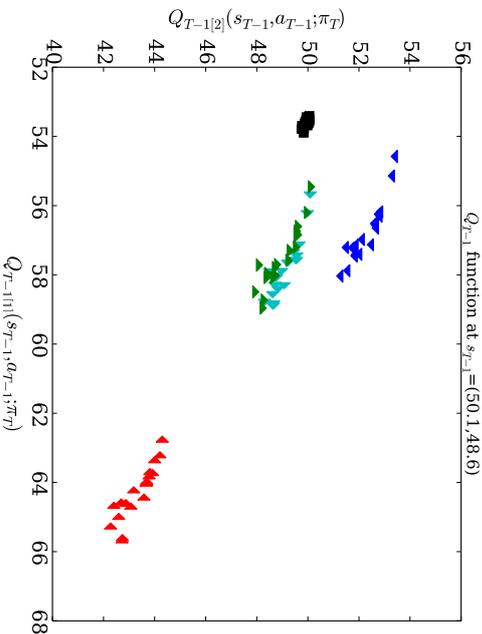


Figure 2: Partial visualization of the members of an example Q_{T-1} . We fix a state $s_{T-1} = (50.1, 48.6)$ in this example, and we plot $\hat{Q}_{T-1}(s_{T-1}, a_{T-1})$ for each $\hat{Q}_{T-1} \in Q_{T-1}$ and for each $a_{T-1} \in \{\blacktriangledown, \blacksquare, \blacktriangle, \blacklozenge, \blacktriangleright, \blacktriangleleft\}$. For example, the \blacktriangledown markers near the top of the plot correspond to expected returns for each $\hat{Q} \in Q_{T-1}$ that is achievable by taking the \blacktriangledown action at the current time point and then following a particular future policy. This example Q_{T-1} contains 20 \hat{Q}_{T-1} functions, each assuming a different π_T .

Definition 2 (Policy consistency) A policy π is consistent with an NDP Π , denoted $\pi \subset \Pi$, if and only if $\pi(s) \in \Pi(s) \forall s \in S$. We denote the set of all policies consistent with Π by $C(\Pi)$.

This restriction is analogous to fitted- Q in the scalar reward setting, where we estimate the current Q function assuming we will follow the greedy policy of the estimated optimal Q function at later time points. In our setting, there are likely to be multiple different policies whose values, pointwise at each state, are considered “optimal,” e.g. that are Pareto non-dominated. Although we cannot pare down the possible future policies to a single unique choice as in scalar fitted- Q , we can still make significant computational savings. Note that in the batch RL setting, two policies are distinguishable only if they differ in action choice on states observed in our data set. In the following, when we talk about the properties of policies, we mean in particular over the observed states in our data set. Where clarification is needed, we write S_t^n to mean the n states observed in our data set at time t . Note that $|C(\Pi_t)| = \sum_{s \in S_t^n} |\Pi_t(s_t)|$; the product of the cardinalities of the sets produced by Π_t over the observed data. Because $|\Pi_t(s_t)| \leq |\mathcal{A}|$, we have $|C(\Pi_t)| \leq |\mathcal{A}|^n$. If Π_t screens out enough actions from enough observed states, restriction to consistent policies can result in a much smaller Q_T . Unfortunately, in the worst case where $\forall s_t, \Pi_t(s_t) = \mathcal{A}_t$, we have $|C(\Pi_t)| = |\mathcal{A}|^n$,

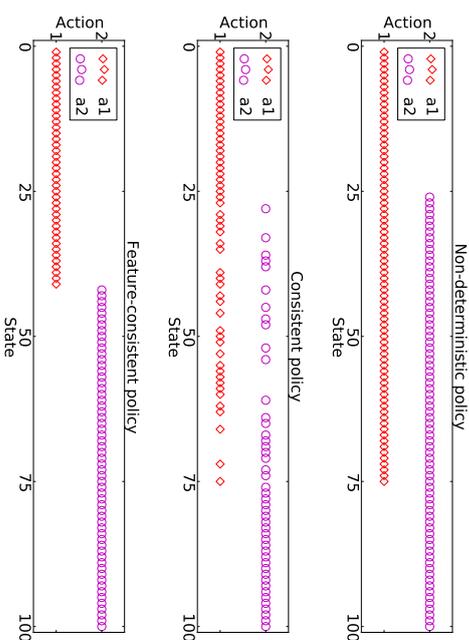


Figure 3: An NDP on a one-dimensional continuous state-space, a consistent policy, and a ϕ -consistent policy.

and if for some fraction η of the n trajectories ($0 < \eta \leq 1$) we have $|\Pi_t(s_t)| \geq 2$, then we have $|C(\Pi_t)| \in \Omega(2^\eta)$. Therefore in many interesting cases, computing a Q_t that includes even just the consistent future policies is computationally intractable.

We therefore impose a further restriction on possible future policies, again only eliminating policies we do not wish to consider. In scalar fitted- Q , the learned optimal policy is given by $\text{argmax}_a Q(s, a)$. If the learned Q -functions are linear in some feature space, then the learned optimal policy can be represented by a collection of linear separators that divide feature space into regions where different actions are chosen. This is true for *any* scalar reward signal. Therefore, in the scalar reward case for a given feature space, any future policy that cannot be represented in this way will never be considered when computing \hat{Q} for earlier timepoints no matter what the observed rewards are.

In NDP settings where $\dim \phi_t(s_t, a_t) \ll n$, most of the policies that are consistent with $\Pi_t(s_t)$ are not representable in the form $\pi(s_t) = \text{argmax}_a Q_t(s_t, a)$, and therefore would never be learned by fitted- Q iteration using *any* scalar reward signal. Figure 3 illustrates this. The top panel shows a non-deterministic policy on a one-dimensional continuous state-space with two possible actions. The middle panel shows a policy that is consistent with the NDP. Though it is consistent, this policy is a complex function of the 1D state and is difficult to justify if the state is a continuous patient measurement and the action is a treatment. Furthermore, there is *no* Q -function linear in the given feature space that produces this consistent policy as its greedy policy. In other words, given the feature space, there is *no scalar reward signal* that would cause us to learn this policy with fitted- Q and

linear regression. We therefore will “prune away” these consistent but un-representable policies in order to reduce the size of \mathcal{Q}_t by introducing the notion of *policy ϕ -consistency*.

Definition 3 (Policy ϕ -consistency) *Given a feature map $\phi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^p$, we say a policy π_t is ϕ -consistent with a non-deterministic policy Π_t over a data set with n trajectories, if and only if $\exists \mathbf{w} (\forall i \in 1, \dots, n \pi_t(s_t^i) \wedge \pi_t(s_t^i) = \arg\max_a \phi(s_t^i, a)^\top \mathbf{w})$. We write $\pi_t \sqsubseteq_\phi \Pi_t$, and we denote the set of all policies that are ϕ -consistent with Π_t by $\mathcal{C}_\phi(\Pi_t)$.*

A ϕ -consistent policy is an element of $\mathcal{C}(\Pi_t)$ that is the argmax policy for some (scalar) Q -function over the feature map ϕ . The form of such a policy is much like that of the function learned by a structured-output SVM (Tsochantaridis et al., 2005).

We now show that the number of ϕ -optimal policies for any given time point is polynomial in the data set size n .

Theorem 1 *Given a data set of size n , a feature map ϕ , and an action set \mathcal{A} , there are at most $O(n^{\dim(\phi)} \cdot |\mathcal{A}|^{2 \dim(\phi)})$ feature-consistent policies.*

Proof The space of ϕ -consistent policies is exactly analogous to the space of linear multiclass predictors with ϕ as their feature map. We therefore port two results from learning theory to analyze the number of ϕ -consistent policies in terms of the dimension of ϕ , the size of the data set n , and the size of the action set. The *Natarajan dimension* (Natarajan, 1989; Shalev-Shwartz and Ben-David, 2014) is an extension of VC-dimension to the multiclass setting. For a supervised learning data set of size n , k classes, and a hypothesis class \mathcal{H} with Natarajan dimension $\text{Ndim}(\mathcal{H})$, the number $|\mathcal{H}_n|$ of hypotheses restricted to the n datapoints is subject to the following upper bound due to Natarajan (1989):

$$|\mathcal{H}_n| \leq n^{\text{Ndim}(\mathcal{H})} \cdot k^{2 \cdot \text{Ndim}(\mathcal{H})}. \quad (5)$$

Furthermore, the hypothesis class given by

$$\mathcal{H}_\phi = \{x \mapsto \arg\max_i \phi(x, i)^\top \mathbf{w} : \mathbf{w} \in \mathbb{R}^{\dim \phi}\} \quad (6)$$

has Natarajan dimension $\text{Ndim}(\mathcal{H}_\phi) = \dim(\phi)$ (Shalev-Shwartz and Ben-David, 2014). ■ Combining Equations (5) and (6) and completes our proof.

Theorem 1 shows that for fixed $|\mathcal{A}|$ and $\dim(\phi)$ there are only polynomially many ϕ -consistent future policies, rather than a potentially exponential number of consistent policies as a function of n . Therefore, by considering only ϕ -consistent future policies, we can ensure that the size of \mathcal{Q}_{T-1} is polynomial in n . The restriction to ϕ -consistent policies applies to Q -functions based on Generalized Linear Models with monotonic increasing link functions (such as logistic regression) as well. Such models have output of the form $g(\phi(s_t, a)^\top \mathbf{w})$ for monotonic increasing g . For these models, $\arg\max_a g(\phi(s_t^i, a)^\top \mathbf{w}) = \arg\max_a \phi(s_t, a)^\top \mathbf{w}$, so all of our results and algorithms for ϕ -consistency immediately apply.

We note that even if we prune using ϕ -consistency, the number of policies is exponential in $\dim \phi$, the feature space. Hence, this approach is tractable only for relatively simple Q -models. In this work we demonstrate that it is practical in a proof-of-concept setting (the CATIE study) but we acknowledge this limitation and defer it to future work.

We now express $\mathcal{C}_\phi(\Pi)$ in a way that allows us to enumerate it using a Mixed Integer Program (MIP). To formulate the constraints describing $\mathcal{C}_\phi(\Pi)$, we take advantage of *indicator constraints*, a mathematical programming formalism offered by modern solvers; e.g. the CPLEX optimization software package as of version 10.0, which was released in 2006 (CPLEX). Each indicator constraint is associated with a binary variable, and is only enforced when that variable takes the value 1. To construct the MIP, we introduce $n \times |\mathcal{A}|$ indicator variables $\alpha_{i,j}$ that indicate whether $\pi(s^i) = j$ or not. We then impose the following constraints:

$$\forall i \in 1, \dots, n, j \in 1, \dots, |\mathcal{A}|, \alpha_{i,j} \in \{0, 1\} \quad (7)$$

$$\forall i \in 1, \dots, n, \sum_j \alpha_{i,j} = 1 \quad (8)$$

$$\forall i \in 1, \dots, n, \forall j \in 1, \dots, |\mathcal{A}|, \alpha_{i,j} = 1 \implies \forall k \neq j, (\phi(s^i, j) - \phi(s^i, k))^\top \mathbf{w} \geq 1. \quad (9)$$

Constraints (7) ensure that the indicator variables for the actions are binary. Constraints (8) ensure that, for each example in our data set, exactly one action indicator variable is on. The indicator constraints in (9) ensure that if the indicator for action j is on for the i th example, then weights must satisfy $j = \arg\max_a \phi(s^i, a)^\top \mathbf{w}$. Note that the margin condition (i.e., having the constraint be ≥ 1 rather than ≥ 0) avoids a degenerate solution with $w = \mathbf{0}$.

The above constraints ensure that any feasible $\alpha_{i,j}$ define a policy that can be represented as an argmax of linear functions over the given feature space. Imposing the additional constraint that the policy defined is consistent with a given NDP Π is now trivial:

$$\forall i \in 1, \dots, n, \sum_{j \in \Pi(s^i)} \alpha_{i,j} = 1. \quad (10)$$

Constraints (10) ensure that the indicator that turns on for the i th example in the data must be one that indicates an action that belongs to the set $\Pi(s^i)$.

Note that we have not specified an objective for this MIP; for the problem of generating ϕ -consistent policies, we are only interested in generating feasible solutions and interpreting the label variables as a potential future policy. Software such as CPLEX can enumerate all possible discrete feasible solutions to the constraints we have formulated. To do so, we give the constraints to the solver and ask for solutions given an objective that is identically zero. Note that if we instead minimized the quadratic objective $\|w\|^2$ subject to these constraints, we would recover the consistent policy with the largest margin between action choices in the feature space. The output would be equivalent to exact transductive learning of a hard-margin multiclass SVM using the actions as class labels (Tsochantaridis et al., 2005).

Given \mathcal{Q}_t , our final task is to define $\Pi_t(s_t)$ for all s_t . While \mathcal{Q}_T is a singleton, for $t < T$ this is not the case in general, and we must take this into account when defining $\Pi_t(s_t)$. We present two definitions for $\Pi_t(s_t)$ based on a strict partial order $<$. (For example $<$ may be the Pareto partial order.)

$$\begin{aligned} \Pi_t^>(s_t) &= \{a : \forall \hat{Q} \in \mathcal{Q}_t, (\nexists! a' \neq a, \hat{Q}' \in \mathcal{Q}_t) (\hat{Q}(s_t, a) < \hat{Q}'(s_t, a'))\} \\ \Pi_t^<(s_t) &= \{a : \exists \hat{Q} \in \mathcal{Q}_t (\nexists! a' \neq a, \hat{Q}' \in \mathcal{Q}_t) (\hat{Q}(s_t, a) < \hat{Q}'(s_t, a'))\}. \end{aligned}$$

Algorithm 1 Non-deterministic fitted- Q

```

Learn  $\hat{Q}_T = (\hat{Q}_{T[1]}, \dots, \hat{Q}_{T[D]})$ , set  $\mathcal{Q}_T = \{\hat{Q}_T\}$ 
for  $t = T-1, T-2, \dots, 1$  do
  for all  $s_t^i$  in the data do
    Generate  $\Pi_{\leq}^{\pm}(s_t^i)$  using  $\mathcal{Q}_{t+1}$ 
   $\mathcal{Q}_t \leftarrow \emptyset$ 
  for all  $\pi_t \in \mathcal{C}_\phi(\Pi_{\leq}^{\pm})$  do
    for all  $\hat{Q}_{t+1} \in \mathcal{Q}_{t+1}$  do
      Learn  $(\hat{Q}_{t[1]}(\cdot, \cdot, \pi_t, \dots), \dots, \hat{Q}_{t[D]}(\cdot, \cdot, \pi_t, \dots))$  using  $\hat{Q}_{t+1}$ , add to  $\mathcal{Q}_t$ 

```

Under Π_{\leq}^{\vee} , action a is included if for *all* fixed sequences of policies we might follow after choosing a , no other choice of current action and future policy is preferable according to \prec . Π_{\leq}^{\vee} is appealing in cases where we wish to guard against a naive decision maker choosing poor sequences of future actions. For the \mathcal{Q}_{T-1} shown in Figure 2, we would have $\Pi_{\leq}^{\vee}(s_{T-1}) = \{\blacktriangleright, \blacktriangleleft\}$. The \blacksquare action is obviously eliminated because any \blacktriangleright point dominates every single \blacksquare point. The \blacktriangleleft and \blacktriangleright actions eliminate each other: There are \blacktriangleleft points that are dominated by \blacktriangleright points, and \blacktriangleright points that are dominated by \blacktriangleleft points. Note that this illustrates how $\Pi_{\leq}^{\vee}(s_t)$ could be empty: if our example only contained the \blacktriangleright and \blacktriangleleft actions, we would have $\Pi_{\leq}^{\vee}(s_{T-1}) = \emptyset$. In practice we find that Π_{\leq}^{\vee} can be very restrictive; we therefore present Π_{\leq}^{\pm} as an alternative. Under Π_{\leq}^{\pm} , action a is included if there is *at least one* fixed future policy for which a is not dominated by a value achievable by another (a', \hat{Q}) pair. Note that $\Pi_{\leq}^{\pm} \supseteq \Pi_{\leq}^{\vee}$, and that because the relation $\mathbf{Q} < \mathbf{Q}'$ is a partial order on a finite set, there must exist at least one maximal element; therefore $\Pi_{\leq}^{\pm}(s_t) \neq \emptyset$. In the Figure 2 example, we have $\Pi_{\leq}^{\pm}(s_{T-1}) = \{\blacktriangleright, \blacktriangleleft, \blacktriangle, \blacklozenge\}$; note that \blacksquare is not included because there is always another action that can dominate it if we choose an appropriate future policy. In order to provide increased choice and to ensure we do not generate NDRs with empty action sets, we will use Π_{\leq}^{\pm} in our complete non-deterministic multiple-reward fitted- Q algorithm, but in our examples we will investigate the effect of choosing Π_{\leq}^{\vee} instead.

5.4 Time Complexity

Pseudocode is given in Algorithm 1. The time cost of Algorithm 1 is dominated by the construction of \mathcal{Q}_t , whose size may increase by a factor of $O(n^{\text{dim } \phi})$ at each timestep; therefore in the worst case $|\mathcal{Q}_1|$ is exponential in T . This can be mitigated somewhat by pruning \mathcal{Q}_t at each step, essentially removing from consideration future policy sequences that are dominated no matter what current action is chosen. Again, this pruning has no impact on solution quality because we are only eliminating future policy sequences that will never be executed. Despite the exponential dependence on T , we will show that our method can be successfully applied to real data in Section 7, and we defer the development of approximations to future work.

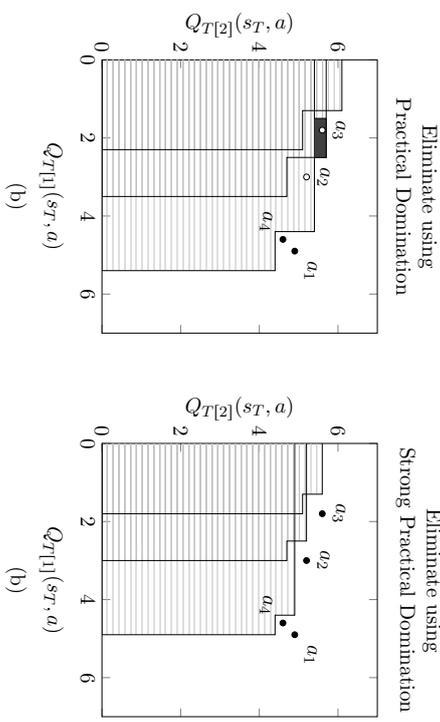


Figure 4: Comparison of rules for eliminating actions. In this simple example, we suppose the Q -vectors $(Q_{T[1]}(s_T, a), Q_{T[2]}(s_T, a))$ are $(4.9, 4.9)$, $(3, 5.2)$, $(1.8, 5.6)$, $(4.6, 4.6)$ for a_1, a_2, a_3, a_4 , respectively, and suppose $\Delta_1 = \Delta_2 = 0.5$. **Figure 4(a)**: Using the Practical Domination rule, action a_4 is not eliminated by a_3 because it is not much worse according to either basis reward, as judged by Δ_1 and Δ_2 . Action a_2 is eliminated because although it is slightly better than a_1 according to basis reward 2, it is much worse according to basis reward 1. Similarly, a_3 is eliminated by a_2 . Note the small solid rectangle to the left of a_2 : points in this region (including a_3) are dominated by a_2 , but not by a_1 . This illustrates the non-transitivity of the Practical Domination relation, and in turn shows that it is not a partial order. **Figure 4(b)**: Using Strong Practical Domination, which is a partial order, no actions are eliminated, and there are no regions of non-transitivity.

6. Practical domination

So far we have presented our algorithm assuming we will use Pareto dominance to define \prec . However, there are two ways in which Pareto dominance does not reflect the reasoning of a physician when she determines whether one action is superior to another. First, an action that has a *slightly* lower value along a single dimension, but is otherwise equivalent, will be Pareto-dominated (and eliminated) even if this difference is clinically meaningless. A physician with this knowledge would consider both actions in light of other “tie-breaking” factors not known to the RL policy, e.g., cost, allergies, etc. Second, an action that is slightly better for one reward but *much* worse for another would *not* be dominated, even though it may realistically be a very poor choice, and perhaps even unethical. Chatterjee et al. (2006) introduced ε -dominance which would partially address the first issue, but not the second. We wish to eliminate only actions that are “obviously” inferior while maintaining as much freedom of choice as possible. To accomplish this, we use the idea of *practical significance* (Kirk, 1996) to develop a definition of domination based on the idea that in real-world applications, small enough differences in expected reward simply do not matter. Differences that fall below a threshold of importance are termed “practically insignificant.”

We introduce two notions of domination that are modifications of Pareto domination. The first, *Practical Domination*, most accurately describes our intuition about the set of actions that should be recommended. However, we show that it has an undesirable non-transitivity property. We then describe an alternative strategy based on what we call *Strong Practical Domination*.

Definition 4 (Practical Domination) *We say that an action a_2 is practically dominated by a_1 at state s_T , and we write $a_2 \prec_p a_1$, if both of the following hold*

$$\forall d \in 1, \dots, D \quad Q_{T[d]}(s_T, a_2) \leq Q_{T[d]}(s_T, a_1) + \Delta_d, \quad (11)$$

$$\exists d \in 1, \dots, D \quad Q_{T[d]}(s_T, a_2) < Q_{T[d]}(s_T, a_1) - \Delta_d. \quad (12)$$

If either of the above do not hold, we write $a_2 \not\prec_p a_1$.

Intuitively, an action a_1 practically dominates a_2 if a_2 is “not practically better” than a_1 for any basis reward (property 11), and if a_2 is “practically worse” than a_1 for *some* basis reward (property 12). “Practically better” and “practically worse” are determined by the elicited differences $\Delta_d \geq 0$. Note that we could have Δ_d depend on the current state if that were appropriate for the application at hand; for simplicity we assume a uniform Δ_d . We might consider using the relation \prec_p as the ordering that produces our NDP according to one of the mappings from Section 5. Unfortunately, \prec_p is not transitive. Suppose that the Q -vectors $(Q_{T[1]}(s_T, a), Q_{T[2]}(s_T, a))$ are $(4.9, 4.9)$, $(3, 5.2)$, $(1.8, 5.6)$, $(4.6, 4.6)$ for a_1, a_2, a_3, a_4 , respectively, and suppose $\Delta_1 = \Delta_2 = 0.5$. Then $a_2 \prec_p a_1$ and $a_3 \prec_p a_2$ but $a_3 \not\prec_p a_1$. This non-transitivity causes undesirable behavior: if we consider only actions a_1 and a_3 , we get $\Pi_{\prec}^+(s_T) = \{a_1, a_3\}$. However, if we consider a_1, a_2 and a_3 , we get $\Pi_{\prec}^+(s_T) = \{a_1\}$! Thus by considering *more* actions, we get a *smaller* $\Pi_{\prec}^+(s_T)$. This is unacceptable in our domain, so we introduce an alternative.²

² Note that for binary actions, the non-transitivity is not an issue and that this is common in some medical applications (e.g. treatment vs. watchful waiting, high-intensity vs. low-intensity treatment, etc.)

Definition 5 (Strong Practical Domination) *We say an action a_2 is strongly practically dominated by a_1 at state s_T , and we write $a_2 \prec_{sp} a_1$, if both of the following hold.*

$$\forall d \in 1, \dots, D \quad Q_{T[d]}(s_T, a_2) \leq Q_{T[d]}(s_T, a_1) \quad (13)$$

$$\exists d \in 1, \dots, D \quad Q_{T[d]}(s_T, a_2) < Q_{T[d]}(s_T, a_1) - \Delta_d \quad (14)$$

If either of the above do not hold, we write $a_2 \not\prec_{sp} a_1$.

The relation \prec_{sp} is transitive, and will not cause the unintuitive results of \prec_p . However, it does not eliminate actions that are slightly better for one basis reward but much worse for another. (Note that $\exists d \in 1..D, Q_{T[d]}(s_T, a_2) > Q_{T[d]}(s_T, a_1) \implies a_2 \not\prec_{sp} a_1$.) We propose a compromise: we will use \prec_{sp} as our partial order for producing NDPs as in Section 5. However, if an action a would have been eliminated according to \prec_p but not according to \prec_{sp} , we may “warn” that it may be a bad choice. This has no impact on computation of Π and Q at earlier time points, but can warn the user that choosing a entails taking a practically significant loss on one basis reward to achieve a practically insignificant gain on another.

7. Empirical Example: CATIE

We illustrate the output of non-deterministic fitted- Q using data from the Clinical Antipsychotic Trials of Intervention Effectiveness (CATIE) study. The CATIE study was designed to compare sequences of antipsychotic drug treatments for the care of schizophrenia patients. The full study design is quite complex (Stroup and al, 2003; Swartz et al., 2003); we use a simplified subset of the CATIE data in order to more clearly illustrate the proposed methodology. CATIE was an 18-month study of $n = 1460$ patients that was divided into two main phases of treatment. Upon entry, most patients began “Phase 1,” and were randomized to one of five treatments³ with equal probability: olanzapine \blacklozenge , risperidone \blacklozenge , quetiapine \blacklozenge , ziprasidone \blacklozenge , or perphenazine \blacksquare . As time passed, patients were given the opportunity to discontinue their Phase 1 treatment and begin “Phase 2” on a new treatment. The possible Phase 2 treatments depended on the reason for discontinuing Phase 1 treatment. If the Phase 1 treatment was ineffective at reducing symptoms, then patients entered the “Efficacy” arm of Phase 2, and their Phase 2 treatment was chosen randomly as: {clozapine \blacklozenge } with probability 1/2, or uniformly randomly from the set {olanzapine \blacklozenge , risperidone \blacklozenge , quetiapine \blacklozenge } with probability 1/2. Because relatively few patients entered this arm, and because of the uneven action probabilities, it is reasonable to combine {olanzapine \blacklozenge , risperidone \blacklozenge , quetiapine \blacklozenge } into one “not-clozapine” action, and we will do so here. If the Phase 1 treatment produced unacceptable side-effects, they entered the “Tolerability” arm of Phase 2, and their Phase 2 treatment was chosen uniformly randomly from {olanzapine \blacklozenge , risperidone \blacklozenge , quetiapine \blacklozenge , ziprasidone \blacklozenge }.

The goal of analyzing CATIE is to develop a two-time point policy ($T = 2$), choosing the initial treatment at $t = 1$ and possibly a follow-up treatment at $t = 2$. From a methodological perspective, the $t = 1$ policy is most interesting as it requires the computation of Q_1 using Algorithm 1. Previous authors have used batch RL to analyze data from this study using

³ Throughout the text we will suffix each treatment name with its corresponding plot-marker.

a single basis reward (Shortreed et al., 2011) and examining convex combinations of basis rewards (Lizotte et al., 2012). In the following, we present the treatment recommendations of a non-deterministic fitted- Q analysis that considers both symptom relief and side-effects, and we compare with the output of a method by Lizotte et al. (2010, 2012). We begin by describing our basis rewards and our state spaces for $t = 2$ and $t = 1$, and we then present our results, paying particular attention to how much action choice is available using the different methods.

7.1 Basis Rewards

We will use ordinary least squares to learn Q functions for two basis rewards. For our first basis reward, we use the Positive and Negative Syndrome Scale (PANSS) which is a numerical representation of the severity of psychotic symptoms experienced by a patient (Kay et al., 1987). PANSS has been used in previous work on the CATIE study (Shortreed et al., 2011; Lizotte et al., 2012; Swartz et al., 2003), and is measured for each patient at the beginning of the study and at several times over the course of the study. Larger PANSS scores are worse, so for our first basis reward $r_{[1]}$ we use 100 minus the percentile of a patient’s PANSS at their exit from the study. We use the distribution of PANSS at intake as the reference distribution for the percentile.

For our second basis reward, we use Body Mass Index (BMI), a measure of obesity. Weight gain is an important and problematic side-effect of many antipsychotic drugs (Allison et al., 1999), and has been studied in the multiple-reward context (Lizotte et al., 2012). Because having a larger BMI is worse, for our second basis reward, $r_{[2]}$, we use 100 minus the percentile of a patient’s BMI at the end of the study, using the distribution of BMI at intake as the reference distribution.

7.2 State Space

For our state space, we use the patient’s most recently recorded PANSS score, which experts consider for decision making (Shortreed et al., 2011). We also include their most recent BMI, and several baseline characteristics.

Because the patients who entered Phase 2 had different possible action sets based on whether they entered the Tolerability or Efficacy arm, we learn separate Q -functions for these two cases. The feature vectors we use for Stage 2 Efficacy patients are given by

$$\phi^{\text{EFF}}(s_2, a_2) = [1, \text{1TD}, \text{1EX}, \text{1ST1}, \text{1ST2}, \text{1ST3}, \text{1ST4}, s_{2:P}, s_{2:B}, \\ \mathbf{1}_{a_2=\bullet}, s_{2:P} \cdot \mathbf{1}_{a_2=\bullet}, s_{2:B} \cdot \mathbf{1}_{a_2=\bullet}]^T.$$

Here, $s_{2:P}$ and $s_{2:B}$ are the PANSS and BMI percentiles at entry to Phase 2, respectively. Feature $\mathbf{1}_{a_2=\bullet}$ indicates that the action at the second stage was clozapine \bullet and not one of the other treatments. We also have other features that do not influence the optimal action choice but that are chosen by experts to reduce variance in the value estimates. 1TD indicates whether the patient has had tardive dyskinesia (a motor-control side-effect), 1EX indicates whether the patient has been recently hospitalized, and 1ST1 through 1ST4

indicate the “site type,” which is the type of facility at which the patient is being treated (e.g. hospital, specialist clinic, etc.)

For Phase 2 patients in the Tolerability arm, the possible actions are $\mathcal{A}_2^{\text{TOI}} = \{\blacktriangleleft, \blacktriangle, \blacktriangleright, \blacktriangleright\}$, and the feature vectors we use are given by

$$\phi^{\text{TOI}}(s_2, a_2) = [1, \text{1TD}, \text{1EX}, \text{1ST1}, \text{1ST2}, \text{1ST3}, \text{1ST4}, s_{2:P}, s_{2:B}, \\ \mathbf{1}_{a_2=\blacktriangleleft}, s_{2:P} \cdot \mathbf{1}_{a_2=\blacktriangleleft}, s_{2:B} \cdot \mathbf{1}_{a_2=\blacktriangleleft}, \mathbf{1}_{a_2=\blacktriangle}, s_{2:P} \cdot \mathbf{1}_{a_2=\blacktriangle}, s_{2:B} \cdot \mathbf{1}_{a_2=\blacktriangle}, \\ \mathbf{1}_{a_2=\blacktriangleright}, s_{2:P} \cdot \mathbf{1}_{a_2=\blacktriangleright}, s_{2:B} \cdot \mathbf{1}_{a_2=\blacktriangleright}, \mathbf{1}_{a_2=\blacktriangleright}, s_{2:P} \cdot \mathbf{1}_{a_2=\blacktriangleright}, s_{2:B} \cdot \mathbf{1}_{a_2=\blacktriangleright}]^T.$$

Here we have three indicator features for different treatments at Phase 2, $\mathbf{1}_{a_2=\blacktriangleleft}$, $\mathbf{1}_{a_2=\blacktriangle}$, with ziprasidone represented by turning all of these indicators off. Again we include the product of each of these indicators with the PANSS percentile s_2 . The remainder of the features are the same as for the Phase 2 Efficacy patients.

For Phase 1 patients, the possible actions are $\mathcal{A}_1 = \{\blacktriangleleft, \blacksquare, \blacktriangle, \blacktriangleright, \blacktriangleright\}$, and the feature vectors we use are given by

$$\phi^{\text{EFF}}(s_2, a_2) = [1, \text{1TD}, \text{1EX}, \text{1ST1}, \text{1ST2}, \text{1ST3}, \text{1ST4}, s_{1:P}, s_{1:B}, \\ \mathbf{1}_{a_2=\blacktriangleleft}, s_{1:P} \cdot \mathbf{1}_{a_2=\blacktriangleleft}, s_{1:B} \cdot \mathbf{1}_{a_2=\blacktriangleleft}, \mathbf{1}_{a_2=\blacksquare}, s_{1:P} \cdot \mathbf{1}_{a_2=\blacksquare}, s_{1:B} \cdot \mathbf{1}_{a_2=\blacksquare}, \\ \mathbf{1}_{a_2=\blacktriangle}, s_{1:P} \cdot \mathbf{1}_{a_2=\blacktriangle}, s_{1:B} \cdot \mathbf{1}_{a_2=\blacktriangle}, \mathbf{1}_{a_2=\blacktriangleright}, s_{1:P} \cdot \mathbf{1}_{a_2=\blacktriangleright}, s_{1:B} \cdot \mathbf{1}_{a_2=\blacktriangleright}, \\ \mathbf{1}_{a_2=\blacktriangleright}, s_{1:P} \cdot \mathbf{1}_{a_2=\blacktriangleright}, s_{1:B} \cdot \mathbf{1}_{a_2=\blacktriangleright}]^T.$$

We have four indicator features for different treatments at Phase 2, $\mathbf{1}_{a_1=\blacktriangleleft}$, $\mathbf{1}_{a_1=\blacksquare}$, and $\mathbf{1}_{a_1=\blacktriangle}$, with ziprasidone represented by turning all of these indicators off. We include the product of each of these indicators with the PANSS percentile s_1 at entry to the study, and the remainder of the features are the same as for the Phase 2 feature vectors. (These are collected before the study begins and are therefore available at Phase 1 as well.)

7.3 Results

The purpose of our empirical study is to demonstrate that our non-deterministic fitted- Q algorithm is feasible to use on real clinical trial data, and that it can offer increased choice over other approaches in a real-world setting. We will discuss several plots of different NDPs. Each point on a plot represents one value of s_1 in our data set, and at each point is placed a marker for each action recommended by an NDP⁵. To use the plots to make a decision for Phase 1, one would find the point on the plot corresponding to a current patient’s state, and see what actions are recommended for that state. One would then decide among them using expert knowledge, knowing that according to the data and the chosen solution concept, any of those actions would be optimal. Then the process would be repeated should the patient move on to Phase 2, using the corresponding plots for $T = 2$ (not shown.) It is important to note that the axes in Figures 5 through 8 represent *state*, even though the same features (measured after treatment) are also used as reward values.

One can think of all of the learned NDPs that we present in the following experiments as transformations of the raw trajectory data into recommended actions, made under different solution concepts. The choice of solution concept is subjective and tied an application at

4. See Section 4.2 of the paper by Shortreed et al. (2011) for an explanation of these kinds of features.

5. Note that Figure 2 is in fact a plot of the Q -function for Phase 1 at a state where (PANSS, BMI) = (50.1, 48.6), limited to a Q_t of size of 20 for clarity.

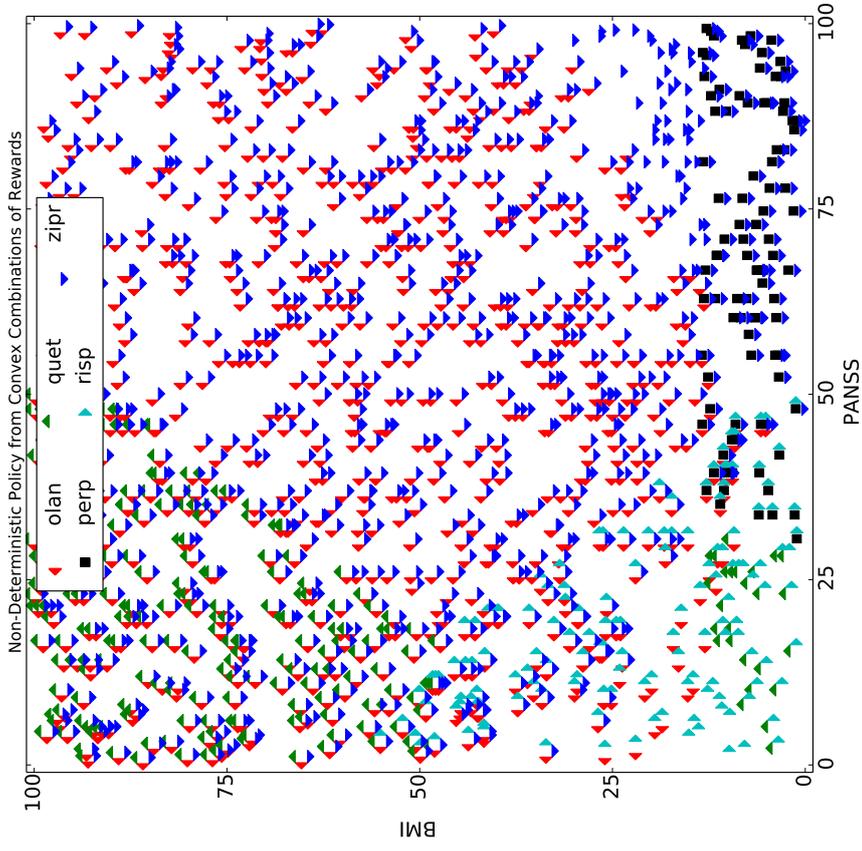


Figure 5: NDP produced by taking the union over actions recommended by Lizotte et al. (2010, 2012)

hand; hence we will not argue that one result is necessarily “better” than another, but rather illustrate some of the differences between them. Indeed, the ability to accommodate different solution concepts is a strength of our approach. That said, we argue that if two solution concepts are both acceptable for a given application, we should prefer the one that offers more action choice to the decision-maker. Therefore, as we discuss the appropriateness of different solution concepts for the CATIE data, and we will examine how the amount of action choice varies for different solution concepts.

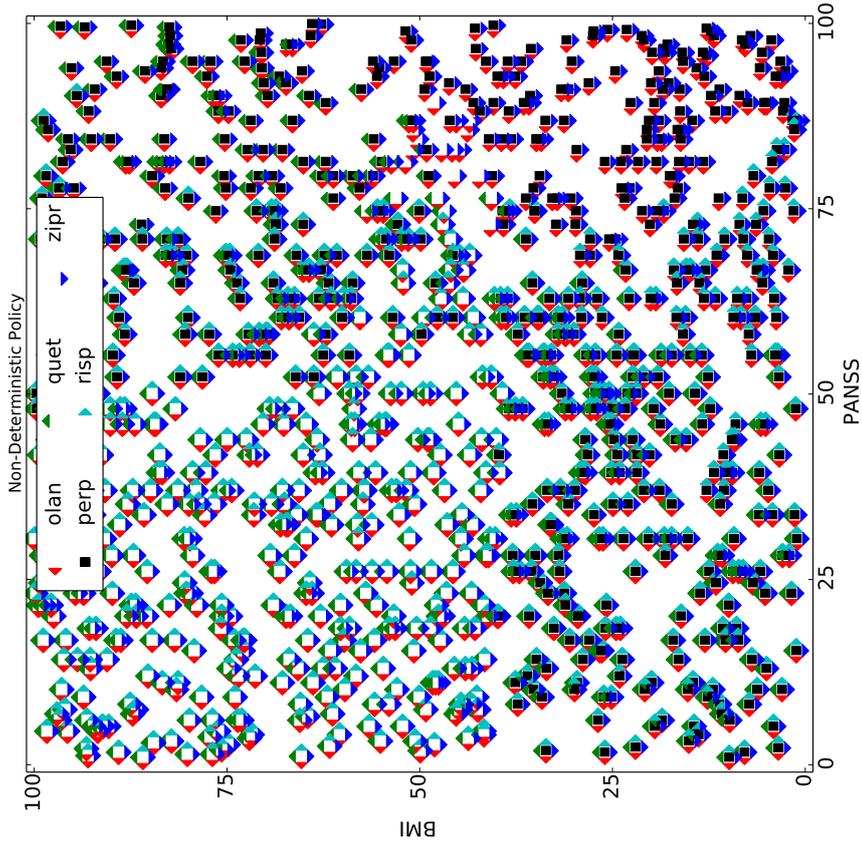


Figure 6: NDP produced by Π_{\leq}^3 with Pareto Domination.

Figure 5 serves as our baseline. It shows the NDP at Phase 1 produced using the convex combination technique of Lizotte et al. (2012), which assumes a linear scalarization function (equivalent to the convex Pareto partial order) and assumes that preferences are fixed over time. One can see that for a large part of the state space, only ziprasidone and olanzapine are recommended. This occurs because for much of the state space, ziprasidone and olanzapine have Q values similar to those in Figure 2: olanzapine performs better on PANSS than on BMI, and ziprasidone has the opposite effect. These two treatments tend to eliminate the more “moderate” actions by the mechanism we described in Figure 1. In

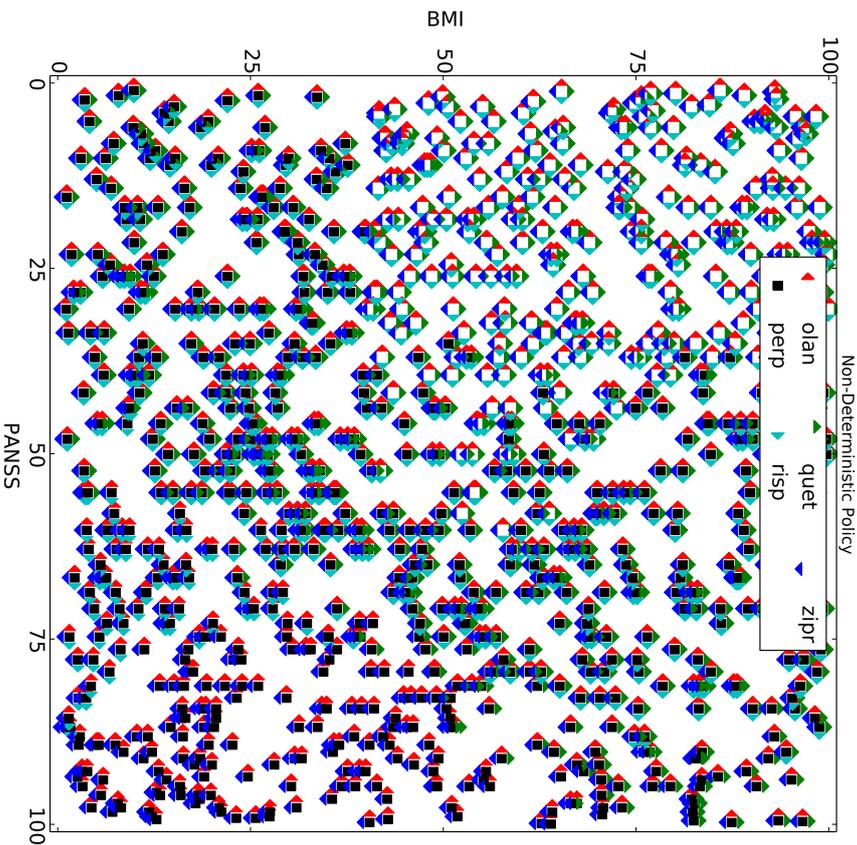


Figure 7: CATIE NDP for Phase 1 made using Π_{\leq}^E ; “warning” actions that would have been eliminated by Practical Domination but not by Strong Practical Domination have been removed.

this NDP, the mean number of choices per state is 2.26, and 100% of states have had one or more actions eliminated.

In our opinion, the convex Pareto domination solution criterion is overly eager to eliminate actions in this context, and the assumption of a fixed scalarization function is unrealistic. Figure 6 shows the NDP learned for Phase 1 using Algorithm 1 with Pareto domination and Π_{\leq}^E , which relaxes these two assumptions. As expected, the recommended action sets

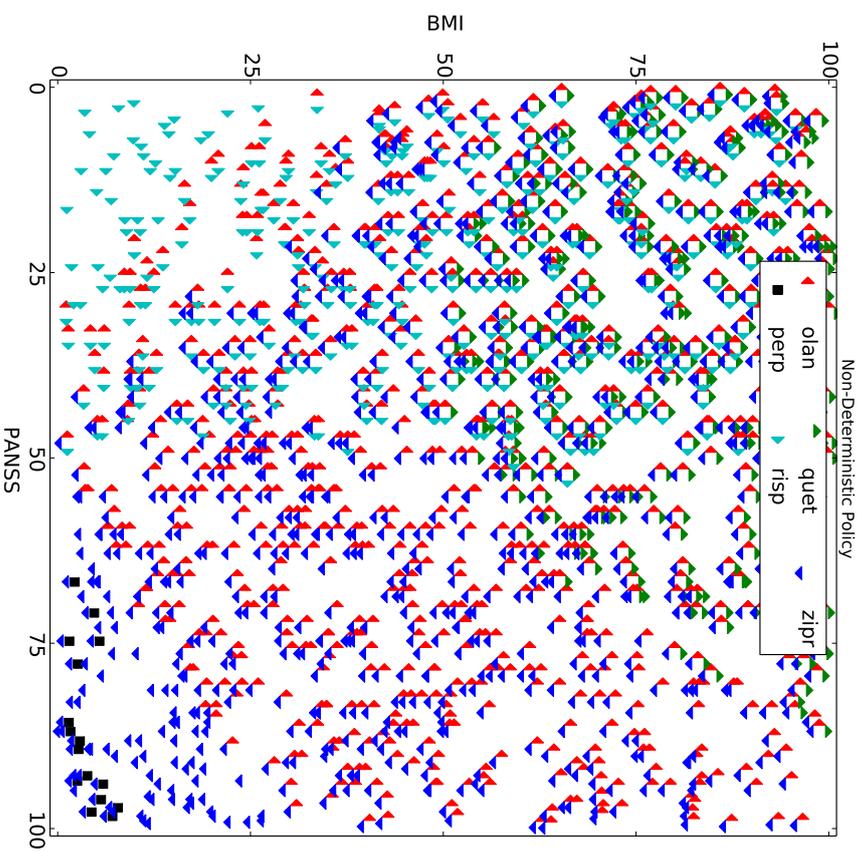


Figure 8: NDP produced by Π_{\leq}^V with Strong Practical Domination.

are larger. Despite the increased choice available, a user following these recommendations can still achieve a value on the Pareto frontier even if their preferences change in Phase 2. In this NDP, the mean number of choices per state is 4.14, and 68% of states have had at least one eliminated.

We now examine the actions that would be recommended if the decision-maker used the Strong Practical Domination solution concept. Figure 7 shows the NDP learned for Phase 1 using our algorithm with Strong Practical Domination ($\Delta_1 = \Delta_2 = 2.5$) and Π_{\leq}^E , and actions that receive a “warning” according to Practical Domination have been removed. In this example, choice is further increased by requiring an action to be practically better

than another action in order to dominate it, and although we have removed actions that were warned to have a bad trade-off—those that were slightly better for one reward but practically worse for another—we still provide increased choice over using the Pareto frontier alone. In this NDP, the mean number of choices per state is 4.30, and 55% of states have had one or more actions eliminated.

We now consider using the same solution concept but the more strict Π_{\leq}^V definition for constructing the NDP. Figure 8 shows the NDP learned for Phase 1 using our algorithm with Strong Practical Domination ($\Delta_1 = \Delta_2 = 2.5$) and Π_{\leq}^V . Again, an action must be practically better than another action in order to dominate it, which tends to increase action choices. However, recall that for Π_{\leq}^V we only recommend actions that are not dominated by another action for any future policy. Hence, these actions are extremely “safe” in the sense that they achieve an expected value on the \leq_{sp} -frontier as long as the user selects from our recommended actions in the future. In this NDP, the mean number of choices per state is 2.56, and 100% of states have had one or more actions eliminated. Hence, we have a trade-off here: Relative to Π_{\leq}^E , this approach reduces choice, yet increases safety; whether or not this is preferable will depend on the application at hand. That said, using Π_{\leq}^V in this way provides more choice than recommending actions based on convex Pareto optimality and a fixed future policy, while at the same time providing a guarantee that the recommended actions are safe choices even if preferences change.

Using ϕ -consistency to reduce the size of Q_t was critical for all of our analyses. In the Phase 2 Tolerability NDP there are over 10^{124} consistent policies but only 1213 ϕ -consistent policies, and in the Phase 2 Efficacy NDP there are 1048576 consistent policies but only 98 ϕ -consistent policies. Finding the ϕ -consistent policies took less than one minute on an Intel Core i7 at 3.4 GHz using Python and CPLEX.

8. Discussion

Our overarching goal is to expand the toolbox of data analysts by developing new, useful methods for producing decision support systems in very challenging settings. To have maximum impact, decision support must appropriately take into account the sequential aspects of the problem at hand and at the same time acknowledge the fact that different decision makers have different preferences. Working toward this goal, we have presented a suite of novel ideas for learning non-deterministic policies for MDPs with multiple objectives. We gave a formulation of fitted- Q iteration for multiple basis rewards, we discussed ways of producing an NDP from a set Q_t of Q -functions that depend on different future policies, we introduced the idea of ϕ -consistent policies to control computational complexity, and we introduced “practical domination” to help users express their preference over actions without explicitly eliciting a preference over basis rewards. Finally, we showed using clinical trial data how our method could be used, and we showed that the NDPs we are able to learn offer more optimal action choice than previous approaches.

One of our next steps will be to augment the definition of practical dominance to incorporate our estimation uncertainty in the Q -values. We will also investigate more aggressive “pruning” of the Q_t to control computational complexity—one could even consider using a single consistent policy per timestep, for example, by adding a margin-based objective to the MIP as described in Section 3.

Rather than restrict ourselves by trying to identify a single “best approach” for all decision support systems, we have developed an algorithm that is modular: One could substitute another notion of domination for the ones we proposed if another notion is more appropriate for a given problem domain. Regardless of this choice, our algorithm will suggest sets of actions that are optimal in the sense we have described. For some applications, Π_{\leq}^E may be appropriate; for other more conservative applications Π_{\leq}^V may be the only responsible choice. Note that we are not dictating how the output from the NDP is used; one could imagine an interface that accepted patient state information and displayed richer information based on Π_{\leq}^E , Π_{\leq}^V , and perhaps plots like Figure 2 to convey to the user what the pros and cons are for the different actions. Our contributions make a wide variety of new decision support systems possible.

Acknowledgments

We acknowledge support from the Natural Sciences and Engineering Research Council of Canada. Data used in the preparation of this article were obtained from the limited access data sets distributed from the NIH-supported “Clinical Antipsychotic Trials of Intervention Effectiveness in Schizophrenia” (CATIE-Sz). The study was supported by NIMH Contract N01MH90001 to the University of North Carolina at Chapel Hill. The ClinicalTrials.gov identifier is NCT00014001. This manuscript reflects the views of the authors and may not reflect the opinions or views of the CATIE-Sz Study Investigators or the NIH.

References

- O. Alagoz, H. Hsu, A. J. Schaefer, and M. S. Roberts. Markov decision processes: A tool for sequential decision making under uncertainty. *Medical decision making : an international journal of the Society for Medical Decision Making*, 30(4):474–483, 2010. ISSN 0272-989X.
- D. B. Allison, J. L. Mentore, M. Heo, L. P. Chandler, J. C. Cappelleri, M. C. Infante, and P. J. Weiden. Antipsychotic-induced weight gain: A comprehensive research synthesis. *American Journal of Psychiatry*, 156:1686–1696, November 1999.
- D. P. Bertsekas. *Dynamic Programming and Optimal Control, Vol. II*. Athena Scientific, 3rd edition, 2007. ISBN 1886529302, 9781886529304.
- D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*, chapter 2.1, page 12. Athena Scientific, 1996.
- D. Blatt, S. A. Murphy, and J. Zhu. A-learning for approximate planning. Technical Report 04-63, The Methodology Center, Penn. State University, 2004.
- E. Brunskill and S. J. Russell. Partially observable sequential decision making for problem selection in an intelligent tutoring system. In *Educational Data Mining (EDM)*, pages 327–328, 2011.
- E. S. Burnside, J. Chhatwal, and O. Alagoz. What Is the Optimal Threshold at Which to Recommend Breast Biopsy? *PLoS ONE*, 7(11):e48820, nov 2012. ISSN 1932-6203.

- A. Castelletti, S. Galati, M. Restelli, and R. Soncini-Sessa. Tree-based reinforcement learning for optimal water reservoir operation. *Water Resources Research*, 46, 2010.
- K. Chatterjee, R. Majumdar, and T. Henzinger. Markov decision processes with multiple objectives. In *STACS*, pages 325–336, 2006.
- M. Chi, K. VanLehn, D. Litman, and P. Jordan. An evaluation of pedagogical tutorial tactics for a natural language tutoring system: A reinforcement learning approach. *Int. J. Artif. Intell. Ed.*, 21(1-2):83–113, January 2011. ISSN 1560-4292.
- R. D. Cook and S. Weisberg. *Applied Regression Including Computing and Graphics*. Wiley, August 1999.
- CPLEX. ILOG CPLEX Optimizer. <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/>, 2012.
- M. Ehrgott. *Multicriteria Optimization*, chapter 3. Springer, second edition, 2005.
- D. Ernst, P. Geurts, and L. Wehenkel. Tree-Based Batch Mode Reinforcement Learning. *Journal of Machine Learning Research*, 6:503–556, 2005.
- T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.
- R. Henderson, P. Ansell, and D. Alshibani. Regret-regression for optimal dynamic treatment regimes. *Biometrics*, 66:1192–1201, 2010.
- S. R. Kay, A. Fiszbein, and L. A. Opler. The Positive and Negative Syndrome Scale (PANSS) for schizophrenia. *Schizophrenia Bulletin*, 13(2):261–276, 1987.
- R. E. Kirk. Practical significance: A concept whose time has come. *Educational and Psychological Measurement*, 56(5):746–759, October 1996.
- E. B. Laber, D. J. Lizotte, and B. Ferguson. Set-valued dynamic treatment regimes for competing outcomes. *Biometrics*, 70(1):53–61, 2014a.
- E. B. Laber, D. J. Lizotte, M. Qian, W. E. Pelham, and S. A. Murphy. Dynamic treatment regimes: technical challenges and applications. *Electronic Journal of Statistics*, 8(1):1225–1272, 2014b.
- D. J. Lizotte, M. Bowling, and S. A. Murphy. Efficient reinforcement learning with multiple reward functions for randomized clinical trial analysis. In *International Conference on Machine Learning (ICML)*, 2010.
- D. J. Lizotte, M. Bowling, and S. A. Murphy. Linear fitted-Q iteration with multiple reward functions. *Journal of Machine Learning Research*, 13:3253–3295, Nov 2012.
- K. M. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer, 1999.
- M. Milani Fard and J. Pineau. Non-deterministic policies in Markovian decision processes. *Journal of Artificial Intelligence Research*, 40:1–24, 2011.
- E. E. M. Moodie, T. S. Richardson, and D. A. Stephens. Demystifying optimal dynamic treatment regimes. *Biometrics*, 63(2):447–455, 2007.
- B. K. Natarajan. On learning sets and functions. *Machine Learning*, 4(1):67–97, 1989.
- P. Perry and P. Weng. On finding compromise solutions in multiobjective Markov decision processes. In *European Conference on Artificial Intelligence (ECAI)*, pages 969–970, 2010.
- C. Paduraru, D. Precup, J. Pineau, and G. Comanici. A study of off-policy learning in computational sustainability. In *European Workshop on Reinforcement Learning (EWRL)*, volume 24 of *JMLR Workshop and Conference Proceedings*, pages 89–102, 2012.
- A. N. Rafferty, E. Brunskill, T. L. Griffiths, and P. Shafto. Faster teaching by POMDP planning. In *International Conference on Artificial Intelligence in Education (AIED)*, pages 280–287, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 978-3-642-21868-2.
- D. M. Rojfers, P. Vamplew, S. Whiteson, and R. Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.
- S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning*. Cambridge University Press, 2014. Cambridge Books Online.
- S. Shortreed, E. B. Laber, D. J. Lizotte, T. S. Stroup, J. Pineau, and S. A. Murphy. Inferring sequential clinical decision-making through reinforcement learning: an empirical study. *Machine Learning*, 84(1-2):109–136, 2011.
- V. J. Strecher, S. Shiffman, and R. West. Moderators and mediators of a web-based computer-tailored smoking cessation program among nicotine patch users. *Nicotine & tobacco research*, 8(S. 1):S95, 2006.
- T. S. Stroup and al. The national institute of mental health clinical antipsychotic trials of intervention effectiveness (CATIE) project: Schizophrenia trial design and protocol development. *Schizophrenia Bulletin*, 29(1), 2003.
- M. S. Swartz, D. O. Perkins, T. S. Stroup, J. P. McEvoy, J. M. Nieri, and D. D. Haal. Assessing clinical and functional outcomes in the clinical antipsychotic of intervention effectiveness (CATIE) schizophrenia trial. *Schizophrenia Bulletin*, 29(1), 2003.
- I. Tsochantzidis, T. Joachims, T. Hofmann, and Y. Altmann. Large margin methods for structured and independent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.
- P. Vamplew, R. Dazeley, E. Barker, and A. Kelarev. Constructing stochastic mixture policies for episodic multiobjective reinforcement learning tasks. In *The 22nd Australasian Conf. on AI*, 2009.
- P. Vamplew, R. Dazeley, A. Berry, R. Issabekov, and E. Dekker. Empirical evaluation methods for multiobjective reinforcement learning algorithms. *Machine Learning*, 84:51–80, 2011.

Measuring Dependence Powerfully and Equitably

Yakir A. Reshef*†

School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138, USA

YAKIR@SEAS.HARVARD.EDU

David N. Reshef*

Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

DNRESHEF@MIT.EDU

Hilary K. Finucane

Department of Mathematics
Massachusetts Institute of Technology
Cambridge, MA 02139, USA.

HILARYF@MIT.EDU

Pardis C. Sabeti**

Department of Organismic and Evolutionary Biology
Harvard University
Cambridge, MA 02138, USA

PSABETI@OEB.HARVARD.EDU

Michael Mitzenmacher**

School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138, USA

MICHAELM@ECS.HARVARD.EDU

* Co-first author.

† To whom correspondence should be addressed.

** Co-last author.

Editor: Edo Airoldi

Abstract

Given a high-dimensional data set, we often wish to find the strongest relationships within it. A common strategy is to evaluate a measure of dependence on every variable pair and retain the highest-scoring pairs for follow-up. This strategy works well if the statistic used (a) has good power to detect non-trivial relationships, and (b) is *equitable*, meaning that for some measure of noise it assigns similar scores to equally noisy relationships regardless of relationship type (e.g. linear, exponential, periodic). In this paper, we define and theoretically characterize two new statistics that together yield an efficient approach for obtaining both power and equitability. To do this, we first introduce a new population measure of dependence and show three equivalent ways that it can be viewed, including as a canonical “smoothing” of mutual information. We then introduce an efficiently computable consistent estimator of our population measure of dependence, and we empirically establish its equitability on a large class of noisy functional relationships. This new statistic has better bias/variance properties and better runtime complexity than a previous heuristic approach. Next, we derive a second, related statistic whose computation is a trivial side-

product of our algorithm and whose goal is powerful independence testing rather than equitability. We prove that this statistic yields a consistent independence test and show in simulations that the test has good power against independence. Taken together, our results suggest that these two statistics are a valuable pair of tools for exploratory data analysis.

Keywords: maximal information coefficient, total information coefficient, equitability, statistical power, mutual information

1. Introduction

The growing dimensionality of today’s data sets has popularized the idea of *hypothesis-generating science*, whereby a data set is used not to test existing hypotheses but rather to help a researcher formulate new ones. A common approach among practitioners is to evaluate some statistic on many candidate variable pairs in a data set, sort the variable pairs from highest-scoring to lowest, and manually examine all the pairs above a threshold score (Storey and Tibshirani, 2003; Emilsson et al., 2008).

A popular class of statistics used for such analyses is *measures of dependence*, i.e., statistics whose population value is zero in cases of statistical independence and non-zero otherwise. Measures of dependence are attractive because they guarantee that asymptotically no non-trivial relationship will erroneously be declared trivial. In the setting of continuous-valued data, which is our focus, there is a long line of fruitful research on such statistics including, e.g., Hoeffding (1948); Rényi (1959); Breiman and Friedman (1985); Painski (2003); Székely et al. (2007); Gretton et al. (2005); Reshef et al. (2011); Gretton et al. (2012); Lopez-Paz et al. (2013); Heller et al. (2013); Jiang et al. (2015); Heller et al. (2016).

One way to measure the utility of a measure of dependence $\hat{\varphi}$ is *power against independence*, i.e., the power of independence testing based on $\hat{\varphi}$ to detect various types of non-trivial relationships. This is an important goal for data sets that have very few non-trivial relationships, or only very weak relationships that are difficult to detect. Often, however, the number of relationships declared statistically significant by a measure of dependence greatly exceeds the number of relationships that can then be explored further. For example, biological data sets often contain many non-trivial relationships, but further corroborating any one of them may take extensive manual lab work or a study on human or animal subjects. In this case, it is tempting to restrict follow-up to a few relationships with the highest values of $\hat{\varphi}$, but this can skew the direction of follow-up work: if $\hat{\varphi}$ systematically assigns higher scores to, say, linear relationships than to non-linear ones, relatively noisy linear relationships might crowd out strong non-linear relationships from the top-scoring set.

Motivated by this problem, we previously introduced a second way of assessing a measure of dependence, called *equitability* (Reshef et al., 2011). Informally, an equitable statistic is one that, for some measure of relationship strength, assigns similar scores to equally strong relationships regardless of relationship type. For instance, we may want our measure of dependence to also have the property that on noisy functional relationships it assigns similar scores to relationships with the same R^2 , i.e., the squared Pearson correlation in question the observed y-values and the x-values passed through the underlying function in question (Reshef et al., 2011). Or, alternatively, we may want the value of our statistic to tell us about the proportion of points coming from the deterministic component of a mixture containing part signal and part uniform noise (Ding and Li, 2013). Defining measures of dependence that achieve good equitability with respect to interesting measures of relationship strength

is a new and challenging problem, with a number of different formalizations. (See, e.g., Reshef et al., 2015b and Ding and Ji, 2013 cited above, as well as Kinney and Atrwal, 2014 along with associated technical comments Reshef et al., 2014 and Murrell et al., 2014.) A companion paper to this work (Reshef et al., 2015b) presents a general formalization that unifies these.

In this paper, we introduce and theoretically characterize two new measures of dependence that we empirically show to have good equitability with respect to R^2 and power against independence, respectively. We begin by introducing a new population measure of dependence called MIC_* . Given a pair of jointly distributed random variables (X, Y) , $MIC_*(X, Y)$ is the supremum, over all finite grids G imposed on the support of (X, Y) , of the mutual information of the discrete distribution induced by (X, Y) on the cells of G , subject to a regularization based on the resolution of G . We prove three results, each of which gives a different way that this population quantity can be viewed.

1. MIC_* is the population value of the maximal information coefficient (MIC), a statistic introduced in Reshef et al. (2011) that is empirically highly equitable with respect to R^2 on a large class of noisy functional relationships. Simple corollaries of this result simplify and strengthen many of the theoretical results proven in Reshef et al. (2011) about MIC.
2. MIC_* is a minimal smoothing of mutual information, in the sense that the regularization in the definition of MIC_* renders it uniformly continuous as a function of random variables with respect to statistical distance, and no “smaller” regularization achieves continuity. This result yields as a corollary that mutual information by itself is not continuous with respect to statistical distance.
3. MIC_* is the supremum of an infinite sequence defined in terms of optimal (one-dimensional) partitions of the marginal distributions of (X, Y) rather than optimal (two-dimensional) grids imposed on the joint distribution. This characterization greatly simplifies computation.

After proving these three results, we leverage them to introduce efficient algorithms both for approximating MIC_* in practice and for estimating it consistently from a finite sample. We first provide an efficient algorithm that in many cases allows for computation to arbitrary precision of the MIC_* of a pair of random variables whose joint density is known. We then introduce a statistic, called MIC_e , that we prove is a consistent estimator of MIC_* . In contrast to the MIC statistic from Reshef et al. (2011), for which no efficient algorithm is known and a heuristic algorithm is used in practice, MIC_e is efficiently computable. It has a better runtime complexity than the heuristic algorithm currently in use for computing the original MIC statistic, and is orders of magnitude faster in practice.

With a consistent and fast estimator for MIC_* in hand, we turn to empirical analysis of its performance. Specifically, we show through simulation that MIC_e has better bias/variance properties than the heuristic algorithm used in Reshef et al. (2011) for computing MIC, which has no theoretical convergence guarantees. Our analysis also reveals that the main parameter of MIC_e can be used to tune statistical performance toward either stronger or weaker relationships in general. After studying the bias/variance properties of MIC_e , we

then demonstrate via simulation that it outperforms currently available methods in terms of equitability with respect to R^2 on a broad set of noisy functional relationships. We show this performance advantage both on the set of functional relationships analyzed in Reshef et al. (2011) as well as on a large set of randomly chosen noisy functional relationships.

We choose in this paper to analyze equitability specifically with respect to R^2 , rather than some other notion of relationship strength, because R^2 on noisy functional relationships is a simple measure with broad familiarity and intuitive interpretation among practitioners. Of course, it is also important to develop measures of dependence that are equitable with respect to notions of relationship strength besides R^2 or on families of relationships besides noisy functional relationships; however, our focus here remains on the “simple” case of R^2 on noisy functional relationships.

Importantly, we note that although there are methods for directly estimating the R^2 of a noisy functional relationship via nonparametric regression (see, e.g., Cleveland and Devlin, 1988; Stone, 1977), those methods are not applicable in the context of equitability because they are not measures of dependence. That is, because non-parametric regression methods assume a functional form for the relationship in question, they can give trivial scores to non-functional relationships, even in the large-sample limit. (A simple example of this is a uniform distribution over a circle, whose regression function is constant.) In contrast, a *measure of dependence* is guaranteed never to make this “mistake”. A measure of dependence that is equitable with respect to R^2 can therefore be viewed either as an “upgraded” measure of dependence that also comes with some of the interpretability properties of non-parametric regression, or as an “upgraded” approximate non-parametric regression method that also has the robustness properties of a measure of dependence.

The main strength of MIC_e is equitability rather than power to reject a null hypothesis of independence. In some settings, though, it may be more important to focus on good power against independence. We therefore introduce here a statistic closely related to MIC_e called the total information coefficient and denoted TIC_e . We prove the consistency of testing for independence using TIC_e , and show via simulations that it achieves excellent power in practice, performing comparably to or better than current methods on an index suite of relationships from Simon and Tibshirani (2012). Because TIC_e arises naturally as a side-product of the computation of MIC_e , it is available “for free” once MIC_e has been computed. This leads us to propose a data analysis strategy consisting of first using TIC_e to filter out non-significant relationships, and then ranking the remaining ones using the simultaneously computed values of MIC_e .

In addition to the companion paper Reshef et al. (2015b), which focuses on the theory behind equitability, this paper is accompanied by a second companion work (Reshef et al., 2015a) that explores in detail the empirical performance of the methods introduced here. That paper compares MIC_e and TIC_e to several leading measures of dependence (Kruskowl et al., 2004; Székely and Rizzo, 2009; Heller et al., 2013, 2016; Gretton et al., 2005; Breiman and Friedman, 1985; Lopez-Paz et al., 2013) on a broad range of relationship types under many different sampling and noise models, finding that the equitability with respect to R^2 of MIC_e and the power of independence testing using TIC_e are both state-of-the-art on the relationships examined. It also shows that these methods can be computed very fast in practice.

Taken together, our results shed significant light on the theory behind the maximal information coefficient, and suggest that TIC_e and MIC_e are a useful pair of methods for data exploration. Specifically, they point to joint use of these two statistics to filter and then rank relationships as a fast, practical way to explore large data sets by measuring dependence both powerfully and equitably.

2. Preliminaries

We work extensively in this paper with grids and discrete distributions over their cells. Given a grid G and a point (x, y) , we define the function $\text{row}_G(y)$ to be the row of G containing y and we define $\text{col}_G(x)$ analogously. For a pair (X, Y) of jointly distributed random variables, we write $(X, Y)|_G$ to denote $(\text{col}_G(X), \text{row}_G(Y))$, and we use $I((X, Y)|_G)$ to denote the discrete mutual information (Cover and Thomas, 2006; Csiszár and Shields, 2004; Csiszár, 2008) between $\text{col}_G(X)$ and $\text{row}_G(Y)$. Given a finite sample D from the distribution of (X, Y) , we sometimes use D to refer both to the set of points in the sample as well as to a point chosen uniformly at random from D . In the latter case, it will then make sense to talk about, e.g., $D|_G$ and $I(D|_G)$.

For natural numbers k and ℓ , we use $G(k, \ell)$ to denote the set of all k -by- ℓ grids (possibly with empty rows/columns). A grid G is an equipartition of (X, Y) if all the rows of $(X, Y)|_G$ have the same probability mass, and all the columns do as well. We also use the term equipartition in the analogous way for one-dimensional partitions into just rows or columns. For a one-dimensional partition P into rows and a one-dimensional partition Q into columns, we write (P, Q) to refer to the grid constructed from these two partitions. When a partition P can be obtained from a partition P' by addition of separators alone, we write $P' \subset P$.

Finally, let us establish some notation for infinite matrices. We use m^∞ to denote the space of infinite matrices equipped with the supremum norm. Given a matrix $A \in m^\infty$, we often examine only the k, ℓ -th entries of A for which $k\ell \leq i$ for some i . Thus, for $i \in \mathbb{Z}^+$, we define the projection $r_i : m^\infty \rightarrow m^\infty$ via

$$r_i(A)_{k,\ell} = \begin{cases} A_{k,\ell} & k\ell \leq i \\ 0 & k\ell > i \end{cases}.$$

Unless noted otherwise, all logarithms are to base 2.

3. The Population Maximal Information Coefficient MIC_{*}

In this section, we define and characterize the population maximal information coefficient MIC_{*}. We begin by defining the population quantity MIC_{*}(X, Y) for a pair of jointly distributed random variables (X, Y) . We then show three different ways to characterize this population quantity: first, as the large-sample limit of the statistic MIC from Reshef et al. (2011); second, as a minimally smoothed version of mutual information; and third, as the supremum of an infinite sequence defined in terms of optimal one-dimensional partitions of the marginal distributions of (X, Y) . We conclude the section by showing how the third characterization leads to an efficient approach for approximating MIC_{*} in practice from the density of (X, Y) .

3.1 Defining MIC_{*}

The population maximal information coefficient can be defined in several equivalent ways, as we will see later. For now, we begin with the simplest definition.

Definition 1 Let (X, Y) be jointly distributed random variables. The population maximal information coefficient (MIC_{*}) of (X, Y) is defined by

$$\text{MIC}_*(X, Y) = \sup_G \frac{I((X, Y)|_G)}{\log \|G\|}$$

where $\|G\|$ denotes the minimum of the number of rows of G and the number of columns of G .

Given that $I(X, Y) = \sup_G I((X, Y)|_G)$ (see, e.g., Chapter 8 of Cover and Thomas 2006), this can be viewed as a regularized version of mutual information that penalizes complicated grids and ensures that the result falls between zero and one.

Before we continue, we state one simple equivalent definition of MIC_{*} that is useful for the results in this section. This definition views MIC_{*} as the supremum of a matrix called the population characteristic matrix, defined below.

Definition 2 Let (X, Y) be jointly distributed random variables. Let

$$I^*((X, Y), k, \ell) = \max_{G \in G(k, \ell)} I((X, Y)|_G).$$

The population characteristic matrix of (X, Y) , denoted by $M(X, Y)$, is defined by

$$M(X, Y)_{k,\ell} = \frac{I^*((X, Y), k, \ell)}{\log \min\{k, \ell\}}$$

for $k, \ell > 1$.

It is easy to see the following:

Proposition 3 Let (X, Y) be jointly distributed random variables. We have

$$\text{MIC}_*(X, Y) = \sup M(X, Y)$$

where $M(X, Y)$ is the population characteristic matrix of (X, Y) .

The population characteristic matrix is so named because just as MIC_{*}, the supremum of this matrix, captures a sense of relationship strength, other properties of this matrix correspond to different properties of relationships. For instance, later in this paper we introduce an additional property of the characteristic matrix, the total information coefficient, that is useful for testing for the presence or absence of a relationship rather than quantifying relationship strength.

3.2 First Alternate Characterization: MIC_* Is the Population Value of MIC

With MIC_* defined, we now state our first alternate characterization of it, as the large-sample limit of the statistic MIC introduced in Reshef et al. (2011). We begin by first reproducing a description of MIC from Reshef et al. (2011), via the two definitions below.

Definition 4 (Reshef et al., 2011) Let $D \subset \mathbb{R}^2$ be a set of ordered pairs. The sample characteristic matrix $\widehat{M}(D)$ of D is defined by

$$\widehat{M}(D)_{k,\ell} = \frac{I^*(D, k, \ell)}{\log \min\{k, \ell\}},$$

Definition 5 (Reshef et al., 2011) Let $D \subset \mathbb{R}^2$ be a set of n ordered pairs, and let $B : \mathbb{Z}^+ \rightarrow \mathbb{Z}^+$. We define

$$MIC_B(D) = \max_{k \leq B(n)} \widehat{M}(D)_{k,\ell}.$$

where the function $B(n)$ is specified by the user. In Reshef et al. (2011), it was suggested that $B(n)$ be chosen to be n^α for some constant α in the range of 0.5 to 0.8. (The statistics we introduce later will have an analogous parameter; see Section 4.4.1.)

We show the following result about convergence of functions of the sample characteristic matrix to their population counterparts, a consequence of which is the convergence of MIC to MIC_* . (In the theorem statement below, recall that m^∞ is the space of infinite matrices equipped with the supremum norm, and given a matrix A the projection τ_i zeros out all the entries $A_{k,\ell}$ for which $k\ell > i$.)

Theorem 6 Let $f : m^\infty \rightarrow \mathbb{R}$ be uniformly continuous, and assume that $f \circ \tau_i \rightarrow f$ pointwise. Then for every random variable (X, Y) , we have

$$(f \circ \tau_{B(n)}) (\widehat{M}(D_n)) \rightarrow f(M(X, Y))$$

in probability where D_n is a sample of size n from the distribution of (X, Y) , provided $\omega(1) < B(n) \leq O(n^{1-\varepsilon})$ for some $\varepsilon > 0$.

Proof See Appendix A. ■

Since the supremum of a matrix is uniformly continuous as a function on m^∞ and can be realized as the limit of maxima of larger and larger segments of the matrix, this theorem yields our claim about MIC_* as a corollary.

Corollary 7 MIC_B is a consistent estimator of MIC_* provided $\omega(1) < B(n) \leq O(n^{1-\varepsilon})$ for some $\varepsilon > 0$.

Though Theorem 6 is proven in Appendix A, we provide here some intuition for why it should hold as well as a description of the obstacles that must be overcome in the proof.

For concreteness, suppose f is the supremum function. To see why the theorem should hold, fix a random variable (X, Y) and let D be a sample of size n from its distribution. It is known that for a fixed grid G $I(D|_G)$ is a consistent estimator of $I(X, Y|_G)$ (Roulston,

1999; Paninski, 2003). We might therefore expect $I^*(D, k, \ell)$ to be a consistent estimator of $I^*(X, Y, k, \ell)$ as well. And if $I^*(D, k, \ell)$ is a consistent estimator of $I^*(X, Y, k, \ell)$, then we might expect the maximum of the sample characteristic matrix (which just consists of normalized I^* terms) to be a consistent estimator of the supremum of the true characteristic matrix.

These intuitions turn out to be true, but there are two reasons they are non-trivial to prove. First, consistency for I^* does not follow from abstract considerations since the supremum of an infinite set of estimators is not necessarily a consistent estimator of the supremum of the estimands.¹ Second, consistency of I^* alone does not suffice to show that the maximum of the sample characteristic matrix converges to MIC_* . In particular, if $B(n)$ grows too quickly, and the convergence of $I^*(D, k, \ell)$ to $I^*(X, Y, k, \ell)$ is slow, inflated values of MIC can result. To see this, notice that if $B(n) = \infty$ then $MIC = 1$ for uniformly generated noise at any finite sample size, even though each individual entry of the sample characteristic matrix converges to its true value eventually.

The technical heart of the proof is overcoming these obstacles by using the dependencies between the quantities $I(D|_G)$ for different grids G to not only show the consistency of $I^*(D, k, \ell)$ but then to quantify how quickly $I^*(D, k, \ell)$ converges to $I^*(X, Y, k, \ell)$.

3.3 Second Alternate Characterization: MIC_* Is a Minimally Smoothed Mutual Information

We now describe a second equivalent view of MIC_* . Recall that for a pair of jointly distributed random variables (X, Y) , we defined $MIC_*(X, Y)$ as

$$MIC_*(X, Y) = \sup_G \frac{I((X, Y)|_G)}{\log \|G\|}$$

where $\|G\|$ denotes the minimum of the number of rows of G and the number of columns of G . As we discussed in Section 3.1, the mutual information $I(X, Y)$ is also a supremum, namely

$$I(X, Y) = \sup_G I((X, Y)|_G).$$

and so MIC_* can be viewed as a regularized version of I . It is natural to ask whether the regularization in the definition of MIC_* has any smoothing effect on I . In this sub-section we show first that it does, in the sense that MIC_* is uniformly continuous as a function of random variables with respect to the metric of statistical distance,² and second that the regularization by $\log \|G\|$ is in some sense the minimal one necessary for achieving any sort of continuity. As a corollary, we obtain that I by itself is not continuous as a function of

1. If $\theta_1, \dots, \theta_k$ is a finite set of estimators, then a union bound shows that the random variable $(\theta_1(D), \dots, \theta_k(D))$ converges in probability to $(\theta_1, \dots, \theta_k)$ with respect to the supremum metric. The continuous mapping theorem then gives the desired result. However, if the set of estimators is infinite, the union bound cannot be employed. And indeed, if we let $\theta_1 = \dots = \theta_k = 0$, and let $\theta_i(D_n) = i/n$ deterministically, then each θ_i is a consistent estimator of θ_i , but since the set $\{\theta_i(D_n), \theta_{i+1}(D_n), \dots\} = \{1/n, 2/n, \dots\}$ is unbounded, $\sup_i \theta_i(D_n) = \infty$ for every n .

2. Recall that the statistical distance between random variables A and B is defined as $\sup_P |\mathbf{P}(A \in T) - \mathbf{P}(B \in T)|$. When A and B have probability density functions or probability mass functions, this equals one-half of the L_1 distance between those functions.

random variables with respect to the metric of statistical distance. This provides a view of MIC_* as a canonical smoothing of I that yields continuity.

Formally, let $\mathcal{P}(\mathbb{R}^2)$ denote the space of random variables supported on \mathbb{R}^2 equipped with the metric of statistical distance. Our first claim is that as a function defined on $\mathcal{P}(\mathbb{R}^2)$, MIC_* is uniformly continuous. We prove this claim by establishing a stronger result: the uniform continuity of the characteristic matrix $M(X, Y)$. Specifically, by showing that the family of maps corresponding to each individual entry of the characteristic matrix is uniformly equicontinuous, we obtain the following result.

Theorem 8 *The map from $\mathcal{P}(\mathbb{R}^2)$ to m^∞ defined by $(X, Y) \mapsto M(X, Y)$ is uniformly continuous.*

Proof See Appendix B. ■

Since the supremum is a uniformly continuous function on m^∞ , Theorem 8 yields the following corollary.

Corollary 9 *The map $(X, Y) \mapsto MIC_*(X, Y)$ is uniformly continuous.*

Similar corollaries exist for any uniformly continuous function of the characteristic matrix.

Interestingly, Theorem 8 relies crucially on the normalization in the definition of the characteristic matrix. This is not a coincidence: as the following proposition shows, any normalization that is meaningfully smaller than the one in the definition of the characteristic matrix will cause the matrix to contain a discontinuity as a function on $\mathcal{P}(\mathbb{R}^2)$.

Proposition 10 *For some function $N(k, \ell)$, let M^N be the characteristic matrix with normalization N , i.e.,*

$$M^N(X, Y)_{k, \ell} = \frac{I^*((X, Y), k, \ell)}{N(k, \ell)}.$$

If $N(k, \ell) = o(\log \min\{k, \ell\})$ along some infinite path in $\mathbb{N} \times \mathbb{N}$, then M^N and $\sup M^N$ are not continuous as functions of $\mathcal{P}([0, 1] \times [0, 1]) \subset \mathcal{P}(\mathbb{R}^2)$.

Proof See Appendix C. ■

The above proposition implies that the “smoothing” that MIC_* applies to mutual information is necessary in some sense. In particular, one corollary of the proposition is that mutual information with no smoothing will contain a discontinuity.

Corollary 11 *Mutual information is not continuous on $\mathcal{P}([0, 1] \times [0, 1]) \subset \mathcal{P}(\mathbb{R}^2)$.*

Proof Mutual information is the supremum of M^N with $N \equiv 1$. ■

The same result can also be shown for the squared Linfoot correlation (Speed, 2011; Linfoot, 1957), which equals $1 - 2^{-2I}$ where I represents mutual information. Thus, though the Linfoot correlation smooths the mutual information enough to cause it to lie in the unit interval, it does not smooth the mutual information sufficiently to cause it to be continuous.

As we remarked previously, these results, when contrasted with the uniform continuity of MIC_* , allow us to view the latter as a canonical “minimally smoothed” version of mutual information that is uniformly continuous. This view gives a meaningful interpretation to the normalization used in MIC_* . Understanding MIC_* as having smoothness properties not shared by mutual information also suggests that estimators of MIC_* may have better statistical properties than estimators of ordinary mutual information. This is consistent with a recent hardness-of-estimation result for mutual information in Ding and Li (2013) and is also borne out empirically in Reshef et al. (2015a).

3.4 Third Alternate Characterization: MIC_* Is the Supremum of the Boundary of the Characteristic Matrix

We now show the third alternate view of MIC_* : that it can be equivalently defined as the supremum over a boundary of the characteristic matrix rather than as a supremum over all of the entries of the matrix. This characterization of MIC_* will serve as the foundation both for our approach to approximating $MIC_*(X, Y)$ as well as the new estimator of MIC_* that we introduce later in this paper.

We begin by defining what we mean by the boundary of the characteristic matrix. Our definition rests on the following observation.

Proposition 12 *Let M be a population characteristic matrix. Then for $\ell \geq k$, $M_{k, \ell} \leq M_{k, \ell+1}$.*

Proof Let (X, Y) be the random variable in question. Since we can always let a row/column be empty, we know that $I^*((X, Y), k, \ell) \leq I^*((X, Y), k, \ell + 1)$. And since $\ell, \ell + 1 \geq k$, we know that $M_{k, \ell} = I^*((X, Y), k, \ell) / \log k \leq I^*((X, Y), k, \ell + 1) / \log k = M_{k, \ell+1}$. ■

Since the entries of the characteristic matrix are bounded, the monotone convergence theorem then gives the following corollary. In the corollary and henceforth, we let $M_{k, \uparrow} = \lim_{\ell \rightarrow \infty} M_{k, \ell}$ and define $M_{\uparrow, \ell}$ similarly.

Corollary 13 *Let M be a population characteristic matrix. Then $M_{k, \uparrow}$ exists, is finite, and equals $\sup_{\ell \geq k} M_{k, \ell}$. The same is true for $M_{\uparrow, \ell}$.*

The above corollary allows us to define the boundary of the characteristic matrix.

Definition 14 *Let M be a population characteristic matrix. The boundary of M is the set*

$$\partial M = \{M_{k, \uparrow} : 1 < k < \infty\} \cup \{M_{\uparrow, \ell} : 1 < \ell < \infty\}.$$

The theorem below then gives a relationship between the boundary of the characteristic matrix and MIC_* .

Theorem 15 *Let (X, Y) be a random variable. We have*

$$MIC_*(X, Y) = \sup \partial M(X, Y)$$

where $M(X, Y)$ is the population characteristic matrix of (X, Y) .

Proof The following argument shows that every entry of M is at most $\sup \partial M$: fix a pair (k, ℓ) and notice that either $k \leq \ell$, in which case $M_{k,\ell} \leq M_{k,\uparrow}$ or $\ell \leq k$, in which case $M_{k,\ell} \leq M_{\uparrow,\ell}$. Thus, $\text{MIC}_* \leq \sup\{M_{\uparrow,\ell}\} \cup \{M_{k,\uparrow}\} = \sup \partial M$.

On the other hand, Corollary 13 shows that each element of ∂M is a supremum over some elements of M . Therefore, $\sup \partial M$, being a supremum over suprema of elements of M , cannot exceed $\sup M = \text{MIC}_*$. ■

3.5 Approximating MIC_* in Practice

The importance of the characterization in Theorem 15 from the previous sub-section is computational. Specifically, elements of the boundary of the characteristic matrix can be expressed in terms of a maximization over (one-dimensional) partitions rather than (two-dimensional) grids, the former being much quicker to compute exactly. This is stated in the theorem below.

Theorem 16 *Let M be a population characteristic matrix. Then $M_{k,\uparrow}$ equals*

$$\max_{P \in \mathcal{P}(k)} \frac{I(X, Y|P)}{\log k}$$

where $\mathcal{P}(k)$ denotes the set of all partitions of size at most k .

Proof See Appendix D. ■

To formally state how this will help us from an algorithmic standpoint, we note that Theorems 15 and 16 above together give the following corollary.

Corollary 17 *Let (X, Y) be a random variable, and let \mathbb{P} be the set of finite-size partitions. Then*

$$\text{MIC}_*(X, Y) = \sup \left\{ \frac{I(X, Y|P)}{\log |P|} : P \in \mathbb{P} \right\} \bigcup \left\{ \frac{I(X|P, Y)}{\log |P|} : P \in \mathbb{P} \right\}$$

where $|P|$ is the number of bins in the partition P .

We can exploit the fact that the expressions in the above corollary involve maximization only over one-dimensional partitions rather than two-dimensional grids to give an algorithm for computing elements of the boundary of the characteristic matrix to arbitrary precision, and by extension an approach to approximating MIC_* in practice. To do so, we utilize as a subroutine a dynamic programming algorithm from Reshef et al. (2011) called OPTIMIZE-X-AXIS. Before continuing, we therefore give a brief overview of that algorithm.

Overview of OPTIMIZE-X-AXIS algorithm from Reshef et al. (2011). The OPTIMIZE-X-AXIS algorithm takes as input a set D of n data points, a fixed partition into columns³ \mathcal{Q} of size ℓ , a “master” partition into rows Π , and a number k . The algorithm returns, for

3. Despite its name, the OPTIMIZE-X-AXIS algorithm can be used to optimize a partition of either axis. In our description of the algorithm here, we choose to describe the algorithm as it would work for optimizing a partition of the y -axis rather than the x -axis. This is for notational coherence of this paper only.

$2 \leq i \leq k$, the partition into rows $P_i \subset \Pi$ that maximizes the mutual information of $D|_{(P_i, \mathcal{Q})}$ among all sub-partitions of Π of size at most i . The algorithm works by exploiting the fact that, conditioned on the location y of the top-most line of P_i , the optimization of the rest of P_i can be formulated as a sub-problem that depends only on the data points below y . The algorithm uses dynamic programming to store and reuse solutions to these subproblems, resulting in a runtime of $O(|\Pi|^2 k \ell)$. If a black-box algorithm is used to compute each required mutual information in time at most T , then the runtime of the algorithm can be shown to be $O(Tk|\Pi|)$.

The following theorem shows that the theory developed about the boundary of the characteristic matrix, together with OPTIMIZE-X-AXIS, yields an efficient algorithm for computing entries of the boundary to arbitrary precision.

Theorem 18 *Given a random variable (X, Y) , $M_{k,\uparrow}$ (resp. $M_{\uparrow,\ell}$) is computable to within an additive error of $O(k\epsilon \log(1/(k\epsilon))) + \epsilon$ (resp. $O(\ell \log(1/(\ell\epsilon))) + \epsilon$) in time $O(kT(E)/\epsilon)$ (resp. $O(\ell T(E)/\epsilon)$), where $T(E)$ is the time required to numerically compute the mutual information of a continuous distribution to within an additive error of ϵ .*

Proof See Appendix E. ■

The algorithm proposed in Theorem 18 gives us a polynomial-time method for computing any finite subset of the boundary ∂M of the population characteristic matrix $M(X, Y)$ of a random variable (X, Y) . Thus, if we have some k_0, ℓ_0 such that the maximum of the finite subset $\{M_{k,\uparrow}, M_{\uparrow,\ell} : k \leq k_0, \ell \leq \ell_0\}$ of ∂M will be ϵ -close to the supremum of the entire set ∂M , we can compute $\text{MIC}_*(X, Y)$ to within an error of ϵ . Though we usually do not have precise knowledge of k_0 and ℓ_0 , for many distributions it is often easy to make very conservative educated guesses for them, in which case this algorithm allows us to approximate $\text{MIC}_*(X, Y)$ very well in practice.

Being able to compute $\text{MIC}_*(X, Y)$ to arbitrary precision in some cases has two main advantages. The first advantage is that it allows us to assess in simulations the large-sample properties of MIC_* , independent of any estimator. This is done in the companion paper (Reshef et al., 2015a), which shows that MIC_* achieves high equitability with respect to R^2 on a set of noisy functional relationships thereby confirming that statistically efficient estimation of MIC_* is a worthwhile goal.

The second advantage is that we can empirically assess the bias, variance, and expected squared error of estimators of MIC_* by taking a distribution, computing MIC_* , and then comparing the result to estimators of it based on finite samples. In the next section, we introduce a new estimator MIC_ϵ of MIC_* and carry out such an analysis to compare its statistical properties to those of the statistic MIC from Reshef et al. (2011).

4. Estimating MIC_* with MIC_ϵ

As we have shown, MIC_* is the population value of the statistic MIC introduced in Reshef et al. (2011). However, though consistent, the statistic MIC is not known to be efficiently computable and in Reshef et al. (2011) a heuristic approximation algorithm called APPROX- MIC was computed instead. In this section, we leverage the theory we have developed here

to introduce a new estimator of MIC_* that is both consistent and efficiently computable. The new estimator, called MIC_e , has better runtime complexity even than the heuristic APPROX-MIC algorithm, and runs orders of magnitude faster in practice.

The estimator MIC_e is based on one of the alternate characterizations of MIC_* proven in the previous section. Namely, if MIC_* can be viewed as the supremum of the *boundary* of the characteristic matrix rather than of the entire matrix, then only the boundary of the matrix must be accurately estimated in order to estimate MIC_* . This has the advantage that, whereas computing individual entries of the sample characteristic matrix involves finding optimal (two-dimensional) grids, estimating entries of the boundary requires us only to find optimal (one-dimensional) partitions. While the former problem is computationally difficult, the latter can be solved using the dynamic programming algorithm from Reshef et al. (2011) that we also employed in Section 3.5 to compute MIC_* to arbitrary precision in the large-sample limit.

We formalize this idea via a new object called the *equicharacteristic matrix*, which we denote by $[M]$. The difference between $[M]$ and the characteristic matrix M is as follows: while the k, ℓ -th entry of M is computed from the maximal achievable mutual information using any k -by- ℓ grid, the k, ℓ -th entry of $[M]$ is computed from the maximal achievable mutual information using any k -by- ℓ grid that equipartitions the dimension with more rows/columns. (See Figure 1.) Despite this difference, as the equipartition in question gets finer and finer it becomes indistinguishable from an optimal partition of the same size. This intuition can be formalized to show that the boundary of $[M]$ equals the boundary of M , and therefore that $\sup[M] = \sup M = \text{MIC}_*$. It will then follow that estimating $[M]$ and taking the supremum—as we did with M in the case of MIC —yields a consistent estimate of MIC_* .

4.1 The Equicharacteristic Matrix

We now define the equicharacteristic matrix and show that its supremum is indeed MIC_* . To do so, we first define a version of I^* that equipartitions the dimension with more rows/columns. Note that in the definition, brackets are used to indicate the presence of an equipartition.

Definition 19 Let (X, Y) be jointly distributed random variables. Define

$$I^*((X, Y), k, [\ell]) = \max_{G \in \mathcal{G}(k, [\ell])} I((X, Y)|_G)$$

where $G(k, [\ell])$ is the set of k -by- ℓ grids whose y -axis partition is an equipartition of size ℓ . Define $I^*((X, Y), [k], \ell)$ analogously.

Define $I^{[*]}((X, Y), k, \ell)$ to equal $I^*((X, Y), k, [\ell])$ if $k < \ell$ and $I^*((X, Y), [k], \ell)$ otherwise.

We now define the equicharacteristic matrix in terms of $I^{[*]}$. In the definition below, we continue our convention of using brackets to denote the presence of equipartitions.

Definition 20 Let (X, Y) be jointly distributed random variables. The population equicharacteristic matrix of (X, Y) , denoted by $[M](X, Y)$, is defined by

$$[M](X, Y)_{k, \ell} = \frac{I^{[*]}((X, Y), k, \ell)}{\log \min\{k, \ell\}}$$

for $k, \ell > 1$.

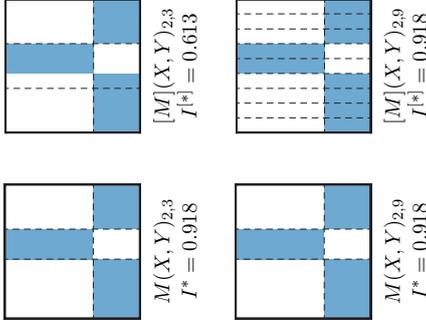


Figure 1: A schematic illustrating the difference between the characteristic matrix M and the equicharacteristic matrix $[M]$. (Top) When restricted to 2 rows and 3 columns, the characteristic matrix M is computed from the optimal 2-by-3 grid. In contrast, the equicharacteristic matrix $[M]$ still optimizes the smaller partition of size 2 but is restricted to have the larger partition be an equipartition of size 3. This results in a lower mutual information of 0.613. (Bottom) When 9 columns are allowed instead of 3, the grid found by the characteristic matrix does not change, since the grid with 3 columns was already optimal. However, now the equicharacteristic matrix uses an equipartition into columns of size 9, whose resolution is able to fully capture the dependence between X and Y .

The boundary of the equicharacteristic matrix can be defined via a limit in the same way as the characteristic matrix. We then have the following theorem.

Theorem 21 Let (X, Y) be jointly distributed random variables. Then $\partial[M] = \partial M$.

Proof See Appendix F. ■

Since every entry of the equicharacteristic matrix is dominated by some entry on its boundary, the equivalence of $\partial[M]$ and ∂M yields the following corollary as a simple consequence.

Corollary 22 Let (X, Y) be jointly distributed random variables. Then $\sup[M](X, Y) = \text{MIC}_*(X, Y)$.

4.2 The Estimator MIC_e

With the equicharacteristic matrix defined, we can now define our new estimator MIC_e in terms of the sample equicharacteristic matrix, analogously to the way we defined MIG in terms of the sample characteristic matrix.

Definition 23 Let $D \subset \mathbb{R}^2$ be a set of ordered pairs. The sample equicharacteristic matrix $[\widehat{M}](D)$ of D is defined by

$$[\widehat{M}](D)_{k,\ell} = \frac{J^*(D, k, \ell)}{\log \min\{k, \ell\}}.$$

Definition 24 Let $D \subset \mathbb{R}^2$ be a set of n ordered pairs, and let $B : \mathbb{Z}^+ \rightarrow \mathbb{Z}^+$. We define

$$MIC_{e,B}(D) = \max_{k,\ell \leq B(n)} [\widehat{M}](D)_{k,\ell}.$$

With the equivalence between the boundary of the characteristic matrix and that of the equicharacteristic matrix established, it is straightforward to show that MIC_e is a consistent estimator of MIC_* via arguments similar to those we applied in the case of MIG . (See Appendix G.) Specifically, we show the following theorem, an analogue of Theorem 6.

Theorem 25 Let $f : m^\infty \rightarrow \mathbb{R}$ be uniformly continuous, and assume that $f \circ r_i \rightarrow f$ pointwise. Then for every random variable (X, Y) , we have

$$(f \circ r_{B(n)})([\widehat{M}](D_n)) \rightarrow f([M](X, Y))$$

in probability where D_n is a sample of size n from the distribution of (X, Y) , provided $\omega(1) < B(n) \leq O(n^{1-\varepsilon})$ for some $\varepsilon > 0$.

By setting $f([M]) = \sup[M]$, we then obtain as a corollary the consistency of MIC_e .

Corollary 26 $MIC_{e,B}$ is a consistent estimator of MIC_* provided $\omega(1) < B(n) \leq O(n^{1-\varepsilon})$ for some $\varepsilon > 0$.

As with the statistic MIG , the statistic MIC_e requires the user to specify a function $B(n)$ to use. While the theory suggests that any function of the form $B(n) = n^\alpha$ suffices provided $0 < \alpha < 1$, different values of α may yield different finite-sample properties. We study the empirical performance of MIC_e for different choices of $B(n)$ in Section 4.4 and point the reader to specific recommendations for practical use in Section 4.4.1.

4.3 Computing MIC_e

Both MIG and MIC_e are consistent estimators of MIC_* . The difference between them is that while MIG can currently be computed efficiently only via a heuristic approximation, MIC_e can be computed exactly, very efficiently, via an approach similar to the one used for approximating MIG_* involving the OPTIMIZEAXIS subroutine. We now describe the details of this approach.

Recall that, given a fixed x -axis partition Q into ℓ columns, a set of n data points, a “master” y -axis partition Π , and a number k , the OPTIMIZEAXIS subroutine finds, for

every $2 \leq i \leq k$, a y -axis partition $P_i \subset \Pi$ of size at most i that maximizes the mutual information induced by the grid (P_i, Q) . The algorithm does this in time $O(|\Pi|^2 k \ell)$. (For more discussion of OPTIMIZEAXIS, see Section 3.5)

In the pair of theorems below, we show two ways that OPTIMIZEAXIS can be used to compute MIC_e efficiently. In the proofs of both theorems, we neglect issues of divisibility, e.g., we often write $B/2$ rather than $\lfloor B/2 \rfloor$. This does not affect the results.

Theorem 27 There exists an algorithm EQUICHAR that, given a sample D of size n and some $B \in \mathbb{Z}^+$, computes the portion $r_{B(n)}([\widehat{M}](D))$ of the sample equicharacteristic matrix in time $O(n^2 B^2)$, which equals $O(n^{4-2\varepsilon})$ for $B(n) = O(n^{1-\varepsilon})$ with $\varepsilon > 0$.

Proof We describe the algorithm and simultaneously bound its runtime. We do so only for the k, ℓ -th entries of $[\widehat{M}](D)$ satisfying $k \leq \ell, k, \ell \leq B$. This suffices, since by symmetry computing the rest of the required entries at most doubles the runtime.

To compute $[\widehat{M}](D)_{k,\ell}$ with $k \leq \ell$, we must fix an equipartition into ℓ columns on the x -axis and then find the optimal partition of the y -axis of size at most k . If we set the master partition Π of the OPTIMIZEAXIS algorithm to be an equipartition into rows of size n , then it performs precisely the required optimization. Moreover, for fixed ℓ it can carry out the optimization simultaneously for all of the pairs $\{(2, \ell), \dots, (B/\ell, \ell)\}$ in time $O(|\Pi|^2 (B/\ell) \ell) = O(n^2 B)$. For fixed ℓ , this set contains all the pairs (k, ℓ) satisfying $k \leq \ell, k \ell \leq B$. Therefore, to compute all the required entries of $[\widehat{M}](D)$ we need only apply this algorithm for each $\ell = 2, \dots, B/2$. Doing so gives a runtime of $O(n^2 B^2)$. ■

The algorithm above, while polynomial-time, is nonetheless not efficient enough for use in practice. However, a simple modification solves this problem without affecting the consistency of the resulting estimates. The modification hinges on the fact that OPTIMIZEAXIS can use master partitions Π besides the equipartition of size n that we used above. Specifically, setting Π in the above algorithm to be an equipartition into ck “chunks”, where k is the size of the largest optimal partition being sought, speeds up the computation significantly. This modification gives a slightly different statistic, but one that has all of the theoretical properties of MIC_e —namely, consistent estimation of MIC_* and efficient exact computation. These properties are formalized in the following theorem.

Theorem 28 Let (X, Y) be a pair of jointly distributed random variables, and let D_n be a sample of size n from the distribution of (X, Y) . For every $c \geq 1$, there exists a matrix $\{\widehat{M}\}^c(D_n)$ such that

1. The function

$$\widetilde{MIC}_{e,B}(\cdot) = \max_{k,\ell \leq B(n)} \{\widehat{M}\}^c(\cdot)_{k,\ell}$$

is a consistent estimator of MIC_* provided $\omega(1) < B(n) \leq O(n^{1-\varepsilon})$ for some $\varepsilon > 0$.

2. There exists an algorithm EQUICHARCLUMP for computing $\text{TR}(\{\widehat{M}\}^c(D_n))$ in time $O(n + B^{5/2})$, which equals $O(n + n^{5(1-\varepsilon)/2})$ when $B(n) = O(n^{1-\varepsilon})$.

Proof See Appendix H. ■

For an analysis of the effect of the parameter c in the above theorem on the results of the EQUICHARCLUMP algorithm, see Appendix H.3.

Setting $\varepsilon = 0.6$ in the above theorem yields the following corollary.

Corollary 29 MIC_* can be estimated consistently in linear time.

Of course, at low sample sizes, setting $\varepsilon = 0.6$ would be undesirable. However, our companion paper (Reshef et al., 2015a) shows empirically that at large sample sizes this strategy works very well on typical relationships.

We remark that the EQUICHARCLUMP algorithm given above is asymptotically faster even than the heuristic APPROX-MIC algorithm used to calculate MIC in practice, which runs in time $O(B(n)^4)$. As demonstrated in our companion paper (Reshef et al., 2015a), this difference translates into a substantial difference in runtimes for similar performance at a range of realistic sample sizes, ranging from a 30-fold speedup at $n = 500$ to over a 350-fold speedup at $n = 10,000$.

For readability, in the rest of this paper we do not distinguish between the two versions of MIC_e computed by the EQUICHAR and EQUICHARCLUMP algorithms described above. Wherever we present simulation data about MIC_e in simulations though, we use the version of the statistic computed by EQUICHARCLUMP.

4.4 Bias/Variance Characterization of MIC_e

The algorithm we presented in Section 3.5 for computing MIC_* to arbitrary precision in some cases allows us to examine the bias/variance properties of estimators of MIC_* . Here, we use it to examine the bias and variance of both MIC as computed by the heuristic APPROX-MIC algorithm from Reshef et al. (2011), and MIC_e as computed by the EQUICHARCLUMP algorithm given above. To do this, we performed a simulation analysis on the following set of relationships

$$\mathcal{Q} = \{(x + \varepsilon_\sigma, f(x) + \varepsilon'_\sigma) : x \in X_f, \varepsilon_\sigma, \varepsilon'_\sigma \sim \mathcal{N}(0, \sigma^2), f \in F, \sigma \in \mathbb{R}_{\geq 0}\}$$

where ε_σ and ε'_σ are i.i.d., F is the set of 16 functions analyzed in Reshef et al. (2011), and X_f is the set of n x-values that result in the points $(x_i, f(x_i))$ being equally spaced along the graph of f .

For each relationship $Z \in \mathcal{Q}$ that we examined, we used the algorithm from Theorem 18 with very conservative values of k_0 and l_0 to compute MIC_* . We then simulated 500 independent samples from Z , each of size $n = 500$, and computed both APPROX-MIC and MIC_e on each one to obtain estimates of the sampling distributions of the two statistics. From each of the two sampling distributions, we estimated the bias and variance of either statistic on Z . We then analyzed the bias, variance, and expected squared error of the two determinations (R^2) with respect to the generating function.

The results, presented in Figure 2, are interesting for two reasons. First, they demonstrate that for a typical usage parameter of $B(n) = n^{0.6}$, MIC_e performs substantially better than APPROX-MIC overall. Specifically, the median of the expected squared error of MIC_e

across the set F of functions is uniformly lower across R^2 values than that of APPROX-MIC. When average expected squared error is used instead of median, MIC_e still performs better on all but the strongest of relationships (R^2 above ~ 0.9). The superior performance of MIC_e is consistent with the fact that we have theoretical guarantees about its statistical properties whereas APPROX-MIC is a heuristic.

Second, the results show that different values of the exponent in $B(n) = n^b$ give good performance in different signal-to-noise regimes due to a bias-variance trade-off represented by this parameter. We expand on this phenomenon and discuss its implications for choosing α in practice below.

4.4.1 CHOOSING $B(n)$

Large values of α lead to increased expected error in lower-signal regimes (low R^2) through both a positive bias in those regimes and a general increase in variance that predominantly affects those regimes. On the other hand, small values of α lead to an increased expected error in higher-signal regimes (high R^2) by leading to a negative bias in those regimes and by shifting the variance of the estimator toward those regimes. In other words, lower values of α are better suited for detecting weaker signals, while higher values of α are better suited for distinguishing among stronger signals. This is consistent with the results seen in our companion paper (Reshef et al., 2015a), which show that low values of α cause MIC_e to yield better powered independence tests while high values of α cause MIC_e to have better equitability.

Reshef et al. (2015a) provides simple, empirical recommendations about appropriate values of α for different settings. Those recommendations are formulated by choosing a set of representative relationships (e.g., a set of noisy functional relationships), as well as a “ground truth” population quantity Φ (e.g., R^2) that can be used to quantify the strength of each of those relationships, and then assessing which values of α maximize the equitability of MIC_e with respect to Φ at a given sample size. This approach is applied to an analysis of real data from the World Health Organization in Reshef et al. (2015a), and the parameters chosen for that analysis are the ones used for all subsequent analyses in this paper.

We remark that if the goal of the user is only detection of non-trivial relationships rather than discovery of the strongest such relationships, α can also be chosen in a more straightforward manner: the user can subsample a small random set of relationships on which to compare the power of MIC_e for different values of α . Those relationships can then be discarded and the rest of the relationships analyzed with the optimal value of α . However, if the user’s primary goal is power against independence, the statistic TIC_e introduced in Section 5 of this paper should be used with this strategy rather than MIC_e .

4.5 Equitability of MIC_e

As mentioned previously, one of the main motivations for the introduction of MIC was equitability, the extent to which a measure of dependence usefully captures some notion of relationship strength on some set of standard relationships. We therefore carried out an empirical analysis of the equitability of MIC_e with respect to R^2 and compared its performance to distance correlation (Székely et al., 2007; Székely and Rizzo, 2009), mutual

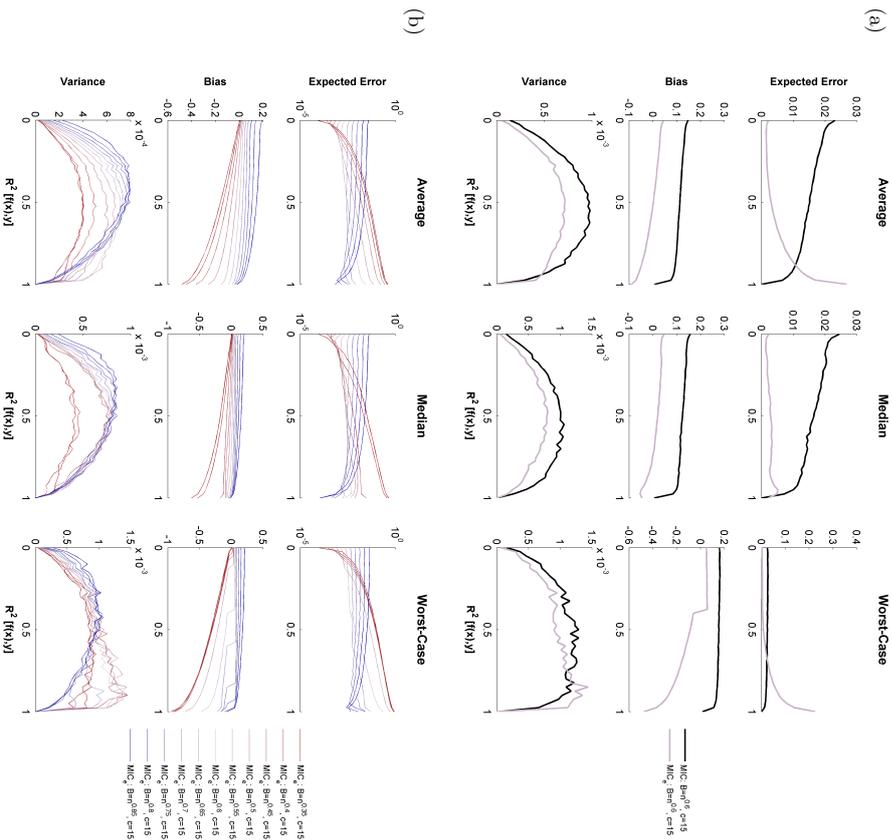


Figure 2: Bias/variance characterization of APPROX-MIC and MIC_C . Each plot shows expected squared error, bias, or variance across the set of noisy functional relationships described in Section 4.4 as a function of the R^2 of the relationships. The results are aggregated across the 16 function types analyzed by either the average, median, or worst result at every value of R^2 . (a) A comparison between MIC_C (light purple) and MIC as computed via the heuristic APPROX-MIC algorithm (black), at a typical usage parameter. (b) Performance of MIC_C with $B(n) = n^\alpha$ for various values of α .

information estimation (Kraskov et al., 2004), and maximal correlation estimation (Breiman and Friedman, 1985).

We began by assessing equitability on the set of relationships \mathcal{Q} defined above, a set that has been analyzed in previous work (Reshef et al., 2011, 2015a; Kinney and Atwal, 2014). The results, shown in Figure 3, confirm the superior equitability of the new estimator MIC_C on this set of relationships.

To assess equitability more objectively without relying on a manually curated set of functions, we then analyzed 160 random functions drawn from a Gaussian process distribution with a radial basis function kernel with one of eight possible bandwidths in the set $\{0.01, 0.025, 0.05, 0.1, 0.2, 0.25, 0.5, 1\}$ to represent a range of possible relationship complexities. The results, shown in Figure 4, show that MIC_C outperforms existing methods in terms of equitability with respect to R^2 on these functions as well. Appendix Figure J1 shows a version of this analysis under a different noise model that yields the same conclusion. We also examined the effect of outlier relationships on our results by repeatedly subsampling random subsets of 20 functions from this large set of relationships and measuring the equitability of each method on average over the subsets: results were similar.

One feature of the performance of MIC_C on these randomly chosen relationships that is demonstrated in Figure 4 is that it appears minimally sensitive to the bandwidth of the Gaussian process from which a given relationship is drawn. This puts it in contrast to, e.g., mutual information estimation, which shows a pronounced sensitivity to this parameter that prevents it from being highly equitable when relationships with different bandwidths are present in the same data set.

In our companion paper (Reshef et al., 2015a), we perform more in-depth analyses of the equitability with respect to R^2 of MIC_C , MIC, and the four measures of dependence described above as well as the Hilbert-Schmidt independence criterion (HSIC) (Gretton et al., 2005, 2007), the Heller-Heller-Gorfine (HHG) test (Heller et al., 2013), the data-derived partitions (DDP) test (Heller et al., 2016), and the randomized dependence coefficient (RDC) (Lopez-Paz et al., 2013). These analyses consider a range of sample sizes, noise models, marginal distributions, and parameter settings. They conclude that, in terms of equitability with respect to R^2 on the sets of noisy functional relationships studied, a) MIC_C uniformly outperforms MIC, and b) MIC_C outperforms all the methods tested in the large majority of settings examined. Appendix Figure II contains a reproduction of a representative equitability analysis from that paper for the reader’s reference.

5. The Total Information Coefficient

So far we have presented results about estimators of the population maximal information coefficient, a quantity for which equitability is the primary motivation. We now introduce and analyze a new measure of dependence coefficient, the *total information coefficient* (TIC). In contrast to the maximal information coefficient, the total information coefficient is designed not for equitability but rather as a test statistic for testing a null hypothesis of independence.

We begin by giving some intuition. Recall that the maximal information coefficient is the supremum of the characteristic matrix. While estimating the supremum of this matrix has many advantages, this estimation involves taking a maximum over many estimates of individual entries of the characteristic matrix. Since maxima of sets of random variables

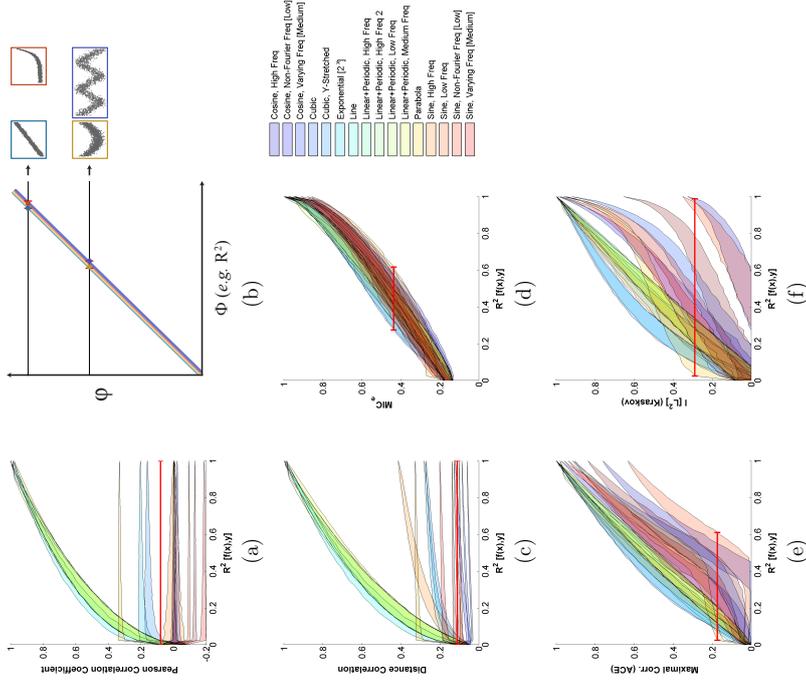


Figure 3: Equitability with respect to R^2 on a set of noisy functional relationships of (a) the Pearson correlation coefficient, (b) a hypothetical measure of dependence φ with perfect equitability, (c) distance correlation, (d) MIC_e , (e) maximal correlation estimation, and (f) mutual information estimation. For each relationship, a shaded region denotes estimated 5th and 95th percentile values of the sampling distribution of the statistic in question on that relationship at every R^2 . The resulting plot shows which values of R^2 correspond to a given value of each statistic. The red interval on each plot indicates the widest range of R^2 values corresponding to any one value of the statistic; the narrower the red interval, the higher the equitability. A red interval with width 0, as in (b), means that the statistic reflects only R^2 with no dependence on relationship type, as demonstrated by the pairs of thumbnails of relationships of different types with identical R^2 values.

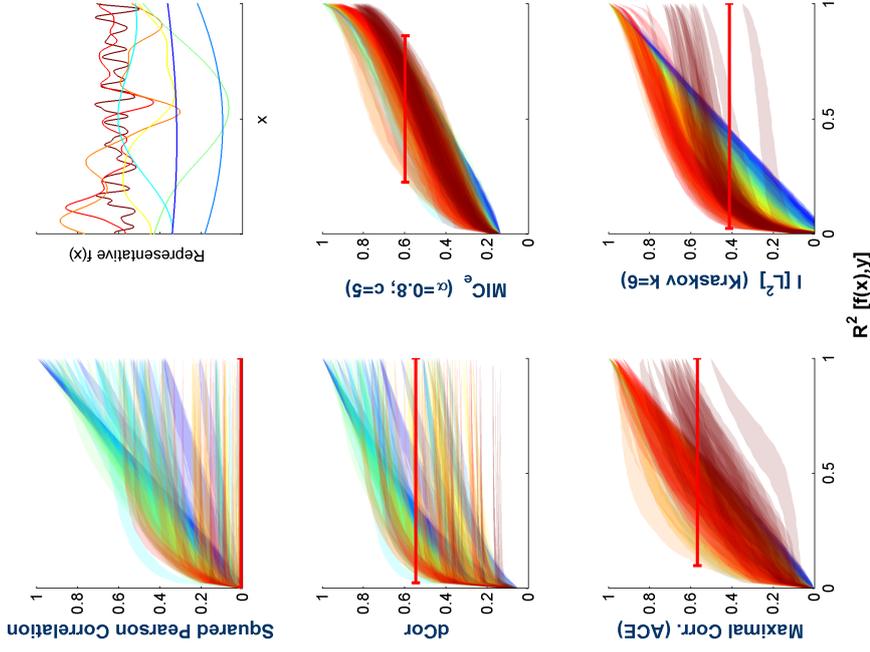


Figure 4: Equitability of methods examined on functions randomly drawn from a Gaussian process distribution. Each method is assessed as in Figure 3, with a red interval indicating the widest range of R^2 values corresponding to any one value of the statistic; the narrower the red interval, the higher the equitability. Each shaded region corresponds to one relationship, and the regions are colored by the bandwidth of the Gaussian process from which they were sampled. Sample relationships for each bandwidth are shown in the top right with matching colors.

tend to become large as the number of variables grows, one can imagine that this procedure may lead to an undesirable positive bias in the case of statistical independence, when the population characteristic matrix equals 0. This might be detrimental for independence testing, when the sampling distribution of a statistic under a null hypothesis of independence is crucial.

The intuition behind the total information coefficient is that if we instead consider a more stable property, such as the sum of the entries in the characteristic matrix, we might expect to obtain a statistic with a smaller bias in the case of independence and therefore better power. Stated differently, if our only goal is to distinguish any dependence at all from complete noise, then disregarding all of the sample characteristic matrix except for its maximal value may throw away useful signal, and the total information coefficient avoids this by summing all the entries.

We remark that in Reshef et al. (2011) it is suggested that other properties of the characteristic matrix may allow us to measure other aspects of a given relationship besides its strength, and several such properties were defined. The total information coefficient fits within this conceptual framework.

In this section we define the total information coefficient in the case of both the characteristic matrix (TIC) and the equicharacteristic matrix (TIC_e). We then prove that both TIC and TIC_e yield independence tests that are consistent against all dependent alternatives. (As in the case of MIC and MIC_e, TIC_e is more easily computable than TIC.) Finally, we present a simulation study of the power of independence testing based on TIC_e on an index set of relationships chosen in Simon and Tibshirani (2012), showing that TIC_e outperforms other common measures of dependence on many of the relationships and closely matches their performance on the rest.

5.1 Definition and Consistency of the Total Information Coefficient

We begin by defining the two versions of the total information coefficient. In the definition below, recall that \widehat{M} denotes a sample characteristic matrix whereas \widehat{M} denotes a sample equicharacteristic matrix.

Definition 30 Let $D \subset \mathbb{R}^2$ be a set of n ordered pairs, and let $B : \mathbb{Z}^+ \rightarrow \mathbb{Z}^+$. We define

$$TIC_B(D) = \sum_{k \leq B(n)} \widehat{M}(D)_{k,\ell}$$

$$TIC_{e,B}(D) = \sum_{k \leq B(n)} \widehat{M}(D)_{k,\ell}$$

and

To show that these two statistics lead to consistent independence tests, we must take a step back and analyze the behavior of the analogous population quantities.

Definition 31 For a matrix A and a positive number B , the B -partial sum of A , denoted by $S_B(A)$, is

$$S_B(A) = \sum_{k \leq B} A_{k,\ell}.$$

When A is an (equi)characteristic matrix, $S_B(A)$ is the sum over all entries corresponding to grids with at most B total cells. Thus, if $M(D)$ is a sample characteristic matrix of a sample D , $S_B(\widehat{M}(D)) = TIC_B(D)$, and the same holds for $S_B(\widehat{M}(D))$ and $TIC_{e,B}(D)$.

It is clear that if X and Y are statistically independent random variables, then both the characteristic matrix $M(X, Y)$ and the equicharacteristic matrix $[M](X, Y)$ are identically 0, so that $S_B(M(X, Y)) = S_B([M](X, Y)) = 0$ for all B . However, we are also interested in how these quantities behave when X and Y are dependent. The following pair of propositions helps us understand this. The first proposition shows a lower bound on the values of entries in both $M(X, Y)$ and $[M](X, Y)$. The second proposition translates this into an asymptotic characterization of how quickly $S_B(M)$ and $S_B([M])$ grow as functions of B . These two propositions are the technical heart of why the total information coefficient yields a consistent independence test.

Proposition 32 Let (X, Y) be a pair of jointly distributed random variables. If X and Y are statistically independent, then $M(X, Y) \equiv [M](X, Y) \equiv 0$. If not, then there exists some $a > 0$ and some integer $\ell_0 \geq 2$ such that

$$M(X, Y)_{k,\ell}, [M](X, Y)_{k,\ell} \geq \frac{a}{\log \min\{k, \ell\}}$$

either for all $k \geq \ell \geq \ell_0$, or for all $\ell \geq k \geq \ell_0$.

Proof See Appendix K.1

Proposition 33 Let (X, Y) be a pair of jointly distributed random variables. If X and Y are statistically independent, then $S_B(M(X, Y)) = S_B([M](X, Y)) = 0$ for all $B > 0$. If not, then $S_B(M(X, Y))$ and $S_B([M](X, Y))$ are both $\Omega(B \log \log B)$.

Proof See Appendix K.2

The propositions above, together with reasoning analogous to the convergence arguments presented earlier, can be used to show the main result of this section, namely that the statistics TIC and TIC_e yield consistent independence tests.

Theorem 34 The statistics TIC_B and $TIC_{e,B}$ yield consistent right-tailed tests of independence, provided $\omega(1) < B(n) \leq O(n^{1-\varepsilon})$ for some $\varepsilon > 0$.

Proof See Appendix K.3. ■

In practice, we often use the EQUICHARGLUMP algorithm (see Section 4.3) to compute the equicharacteristic matrix from which we calculate TIC_e . This algorithm does not compute the sample equicharacteristic matrix exactly. However, as in the case of MIC_e , the use of the algorithm does not affect the theoretical properties of the statistic. This is proven in Appendix H.

5.2 Power of Independence Tests Based on TIC_e

With the consistency of independence tests based on TIC and TIC_e established, we turn now to empirical evaluation of the power of independence testing based on TIC_e as computed using the EQUICHARCLUMP algorithm.

To evaluate the power of TIC_e -based tests, we reproduced the analysis performed in Simon and Tibshirani (2012). Namely, we considered the set of relationships they analyzed, defined by

$$\mathcal{Q} = \{(X, f(X) + \varepsilon) : X \sim \text{Unif}, f \in F, \varepsilon' \sim \mathcal{N}(0, \sigma^2), \sigma \in \mathbb{R}_{>0}\}.$$

where F is a set of functions specified in Simon and Tibshirani (2012). (NB: one of the relationships is a circle, which we treat as a union of two half-circles.)

For each relationship Z in this set that we examined, we simulated a null hypothesis of independence with the same marginal distributions, and generated 1,000 independent samples, each with a sample size of $n = 500$, from both Z and from the null distribution. These were used to estimate the power of the size- α right-tailed independence test based on each statistic being evaluated. Following Simon and Tibshirani, we compared TIC_e to the distance correlation (Székely et al., 2007; Székely and Rizzo, 2009), the original maximal information coefficient (Reshef et al., 2011) as approximated using APPROX-MIC, and to the Pearson correlation. (Though it is not a measure of dependence, the Pearson correlation was presumably included by Simon and Tibshirani as an intuitive benchmark for what is achievable under a linear model.) We also compared to MIC_e using identical parameters to those of TIC_e to examine whether the summation performed by TIC_e is better than maximization when all other things are equal. Note that we do not compare to methods of analyzing contingency tables, such as Pearson’s chi-squared test. This is because our data are real-valued rather than discrete, and so contingency-based methods are not applicable. However, when data are discrete, those methods can be very well powered.

The results of our analysis are presented in Figure 5. First, the figure shows that TIC_e compares quite favorably with distance correlation, a method considered to have state-of-the-art power (Simon and Tibshirani, 2012). Specifically, TIC_e uniformly outperforms distance correlation on 5 of the 8 relationship types examined, and performs comparably to it on the other three relationship types. We remark that distance correlation has many advantages over TIC_e , including the fact that it easily generalizes to higher-dimensional relationships and comes with an elegant and comprehensive theoretical framework.

The analysis also shows that TIC_e outperforms the original maximal information coefficient by a very large margin, and outperforms MIC_e as well, supporting the intuition that the summation performed by the former can indeed lead to substantial gains in power against independence over the maximization performed by the latter. (We note that in both Simon and Tibshirani’s analysis and in this one, the original maximal information coefficient was run with default parameters that were optimized for equitability rather than power against independence. When run with different parameters, its power improves substantially, though it still does not match the power of MIC_e . See Appendix Figure I2 and the discussion in Reshef et al., 2015a.)

Our companion paper (Reshef et al., 2015a) expands on this analysis, conducting an in-depth evaluation of the power against independence of the tests described above as well

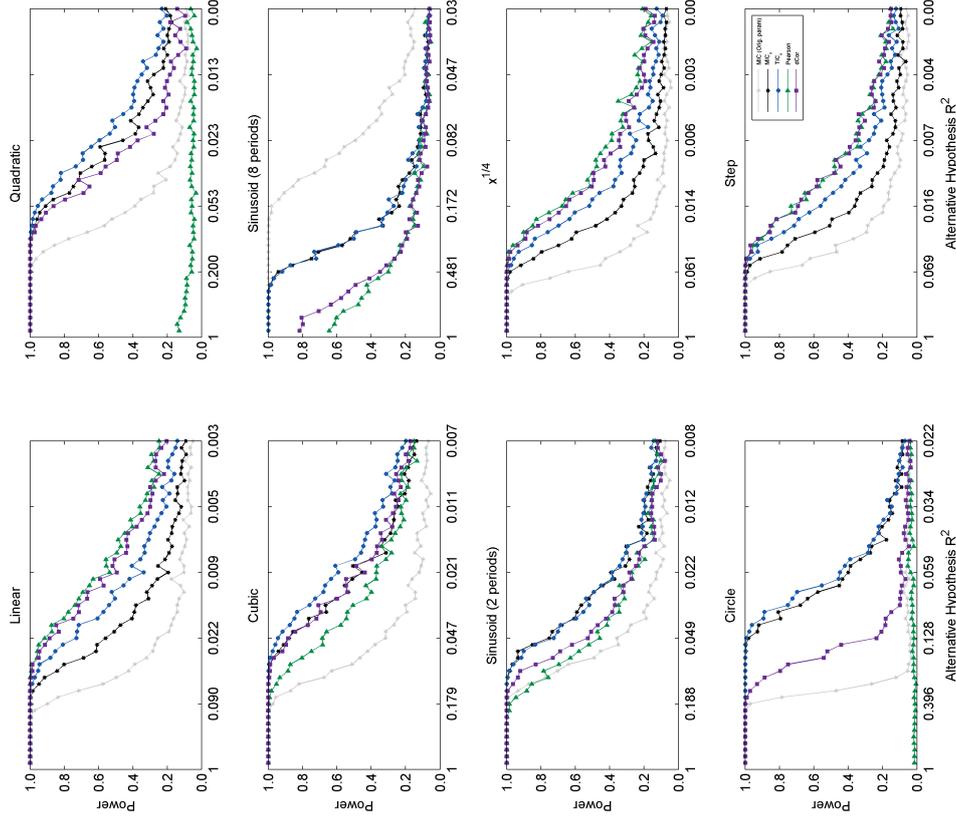


Figure 5: Comparison of power of independence testing based on TIC_e (blue) to MIC with default parameters (gray), MIC_e with the same parameters as TIC_e (black), distance correlation (purple), and the Pearson correlation coefficient (green) across several alternative hypothesis relationship types chosen by Simon and Tibshirani (2012). The relationships analyzed are described in Section 5.2.

as tests based on mutual information estimation (Kraskov et al., 2004), maximal correlation estimation (Breiman and Friedman, 1985), HSIC (Gretton et al., 2005, 2007), HHG (Heller et al., 2013), DDP (Heller et al., 2016), and RDC (Lopez-Paz et al., 2013). These analyses consider a range of sample sizes and parameter settings, as well as a variety of ways of quantifying power across different alternative hypothesis relationship types and noise levels. They conclude that in most settings TIC_e either outperforms all the methods tested or performs comparably to the best ones. Appendix Figure 12 contains a reproduction of one detailed set of power curves from the main analysis in that paper for the reader’s reference.

6. Conclusion

As high-dimensional data sets become increasingly common, data exploration requires not only statistics that can accurately detect a large number of non-trivial relationships in a data set, but also ones that can identify a smaller number of strongest relationships. The former property is achieved by measures of dependence that yield independence tests with high power; the latter is achieved by measures of dependence that are equitable with respect to some measure of relationship strength. In this paper, we introduced two related measures of dependence that achieve these two goals, respectively, through the following theoretical contributions.

- A new population measure of dependence, MIC_* , that we proved can be viewed in three different ways: as the population value of the maximal information coefficient (MIC) from Reshef et al. (2011), as a “minimal smoothing” of mutual information that makes it uniformly continuous, or as the supremum of an infinite sequence defined in terms of optimal partitions of one marginal at a time of a given joint distribution.
- An efficient approach for approximating the MIC_* of a given joint distribution.
- A statistic MIC_e that is a consistent estimator of MIC_* , is efficiently computable, and has good equitability with respect to R^2 both on a manually chosen set of noisy functional relationships as well as on a set of randomly chosen noisy functional relationships.
- The total information coefficient (TIC_e), a statistic that arises as a trivial side-product of the computation of MIC_e and yields a consistent and powerful independence test.

Though we presented here some empirical results for MIC_* , MIC_e , and TIC_e , our focus was on theoretical considerations; the performance of these methods is analyzed in detail in our companion paper (Reshef et al., 2015a). That paper shows that on a large set of noisy functional relationships with varying noise and sampling properties, the asymptotic equitability with respect to R^2 of MIC_* is quite high and the equitability with respect to R^2 of MIC_e is state-of-the-art. It also shows that the power of the independence test based on TIC_e is state-of-the-art across a wide variety of dependent alternative hypotheses. Finally, it demonstrates that the algorithms presented here allow for MIC_e and TIC_e to be computed simultaneously very quickly, enabling analysis of extremely large data sets using both statistics together.

Our contributions are of both theoretical and practical importance for several reasons. First, our characterization of MIC_* as the large-sample limit of MIC sheds light on the latter statistic. For example, while MIC is parametrized, MIC_* is not. Knowing that MIC converges in probability to MIC_* tells us that this parametrization is statistical only: it controls the bias/variance properties of the statistic, but not its asymptotic behavior.

Second, the normalization in the definition of MIC, while empirically seen to yield good performance, had previously not been theoretically understood. Our result that this normalization is the minimal smoothing necessary to make mutual information uniformly continuous provides for the first time a lens through which the normalization is canonical. In doing so, it constitutes an initial step toward understanding the role of the normalization in the performance of MIC_* and MIC. The uniform continuity of MIC_* and the lack of continuity of ordinary mutual information also suggest that estimation of the former may be easier in some sense than estimation of the latter. This is consonant with a recent result concerning difficulty of estimation of mutual information shown in Ding and Li (2013). It is also borne out empirically by the substantial finite-sample bias and variance observed in Reshef et al. (2015a) of the Kraskov mutual information estimator (Kraskov et al., 2004) compared to MIC_e .

Third, our alternate characterization of MIC_* in terms of one-dimensional optimization over partitions rather than two-dimensional optimization over grids enhances our understanding of how to efficiently compute it in the large-sample limit and estimate it from finite samples using MIC_e . This is a significant improvement over the previous state of affairs, in which the statistic MIC could only be approximated heuristically, with even the heuristic approximation being orders of magnitude slower than the results in this paper now allow.

Finally, the introduction of the total information coefficient provides evidence that the basic approach of considering the set of normalized mutual information values achievable by applying different grids to a joint distribution is of fundamental value in characterizing dependence. Interestingly, a statistic introduced in Heller et al. (2016) follows a similar approach by considering the (non-normalized) sum of the mutual information values achieved by all possible finite grids. Consistent with our demonstration here that an aggregative grid-based approach works well, that statistic also achieves excellent power. (TIC_e is compared to the statistic from Heller et al. 2016 in our companion paper, Reshef et al., 2015a.)

Taken together, our results point to joint use of the statistics MIC_e and TIC_e as a theoretically grounded, computationally efficient, and highly practical approach to data exploration. Specifically, since the two statistics can be computed simultaneously with little extra cost beyond that of computing either individually, we propose computing both of them on all variable pairs in a data set, using TIC_e to filter out non-significant associations, and then using MIC_e to rank the remaining variable pairs. Such a strategy would have the advantage of leveraging the state-of-the-art power of TIC_e to substantially reduce the multiple-testing burden on MIC_e , while utilizing the latter statistic’s state-of-the-art equitability to effectively rank relationships for follow-up by the practitioner.

Our results, while useful, nevertheless have limitations that warrant exploration in future work. First, for a sample D from the distribution of some random (X, Y) , all of the sample quantities we define here use the naive estimate $I(D|G)$ of the quantity $I(X, Y|G)$ for various grids G . There is a long and fruitful line of work on more sophisticated estimators of the discrete mutual information Paminski (2003) whose use instead of $I(D|G)$ could improve

the statistics introduced here. Second, our approach to approximating the MIC_* of a given joint density consists of computing a finite subset of an infinite set whose supremum we seek to calculate. However, the choice of how large a finite set we should compute in order to approximate the supremum to a given precision remains heuristic. Finally, though empirical characterization of the equitability of MIC_e on representative sets of relationships is important and promising, we are still missing a theoretical characterization of its equitability in the large-sample limit. A clear theoretical demarcation of the set of relationships on which MIC_* achieves good equitability with respect to R^2 , and an understanding of why that is, would greatly advance our understanding of both MIC_* and equitability.

Acknowledgments

We would like to acknowledge R. Adams, E. Airolidi, T. Broderick, A. Gelman, M. Gorfine, R. Heller, J. Huggins, T. Jaakkola, J. Mueller, J. Tenenbaum, and R. Tibshirani for constructive conversations and useful feedback. HKF was supported by the Fannie and John Hertz Foundation. MM was supported in part by NSF grants CCF-1563710, CCF-1535795, CCF-1320231, and CNS-1228598. DNR and YAR were supported by the Paul and Daisy Soros Foundation. YAR was supported by award Number T32GM007753 from the National Institute of General Medical Sciences, as well as the National Defense Science and Engineering Graduate Fellowship. PCS was supported by the Howard Hughes Medical Institute. The content of this paper is solely the responsibility of the authors and does not necessarily represent the official views of the National Institute of General Medical Sciences or the National Institutes of Health.

Appendix A. Proof of Theorem 6

This appendix is devoted to proving Theorem 6, restated below.

Theorem *Let $f : m^\infty \rightarrow \mathbb{R}$ be uniformly continuous, and assume that $f \circ r_i \rightarrow f$ pointwise. Then for every random variable (X, Y) , we have*

$$(f \circ r_{B(n)}) \left(\widetilde{M}(D_n) \right) \rightarrow f(M(X, Y))$$

in probability where D_n is a sample of size n from the distribution of (X, Y) , provided $\omega(1) < B(n) \leq O(n^{1-\varepsilon})$ for some $\varepsilon > 0$.

We prove the theorem by a sequence of lemmas that build on each other to bound the bias of $I^*(D, k, \ell)$. The general strategy is to capture the dependencies between different k -by- ℓ grids G by considering a “master grid” Γ that contains many more than $k\ell$ cells. Given this master grid, we first bound the difference between $I(D|_G)$ and $I((X, Y)|_G)$ only for sub-grids G of Γ . The bound is in terms of the difference between $D|_\Gamma$ and $(X, Y)|_\Gamma$. We then show that this bound can be extended without too much loss to all k -by- ℓ grids. This gives what we seek, because then the difference between $I(D|_G)$ and $I((X, Y)|_G)$ is uniformly bounded for all grids G in terms of the same random variable: $D|_\Gamma$. Once this is done, standard arguments give the consistency we seek. ■

In our argument we occasionally require technical facts about entropy and mutual information that are self-contained and unrelated to the central ideas. These lemmas are consolidated in Appendix L.

We begin by using one of these technical lemmas to prove a bound on the difference between $I(D|_G)$ and $I((X, Y)|_G)$ that is uniform over all grids G that are sub-grids of a much denser grid Γ . The common structure imposed by Γ will allow us to capture the dependence between the quantities $|I(D|_G) - I((X, Y)|_G)|$ for different grids G .

Lemma 35 *Let $\Pi = (\Pi_X, \Pi_Y)$ and $\Psi = (\Psi_X, \Psi_Y)$ be random variables distributed over the cells of a grid Γ , and let $(\pi_{i,j})$ and $(\psi_{i,j})$ be their respective distributions. Define*

$$\varepsilon_{i,j} = \frac{\psi_{i,j} - \pi_{i,j}}{\pi_{i,j}}.$$

Let G be a sub-grid of Γ with B cells. Then for every fixed $0 < a < 1$ we have

$$|I(\Psi|_G) - I(\Pi|_G)| \leq O \left((\log B) \sum_{i,j} |\varepsilon_{i,j}| \right)$$

when $|\varepsilon_{i,j}| \leq 1 - a$ for all i and j .

Proof Let $P = \Pi|_G$ and $Q = \Psi|_G$ be the random variables induced by Π and Ψ respectively on the cells of G . Using the fact that $I(X, Y) = H(X) + H(Y) - H(X, Y)$, we write

$$|I(Q) - I(P)| \leq |H(Q_X) - H(P_X)| + |H(Q_Y) - H(P_Y)| + |H(Q) - H(P)|$$

where Q_X and P_X denote the marginal distributions on the columns of G and Q_Y and P_Y denote the marginal distributions on the rows. We can bound each of the terms on the right-hand side of the equation above using a Taylor expansion argument given in Lemma 51, whose proof is found in Appendix L. Doing so gives

$$|I(Q) - I(P)| \leq (\ln B) \left(\sum_i O(|\varepsilon_{i,*}|) + \sum_j O(|\varepsilon_{*,j}|) + \sum_{i,j} O(|\varepsilon_{i,j}|) \right)$$

where

$$\varepsilon_{i,*} = \frac{\sum_j (\psi_{i,j} - \pi_{i,j})}{\sum_j \pi_{i,j}}$$

and $\varepsilon_{*,j}$ is defined analogously.

To obtain the result, we observe that

$$|\varepsilon_{i,*}| = \left| \frac{\sum_j \pi_{i,j} \varepsilon_{i,j}}{\sum_j \pi_{i,j}} \right| \leq \frac{\sum_j \pi_{i,j} |\varepsilon_{i,j}|}{\sum_j \pi_{i,j}} \leq \sum_j |\varepsilon_{i,j}|$$

since $\pi_{i,j} / \sum_j \pi_{i,j} \leq 1$, and the analogous bound holds for $|\varepsilon_{*,j}|$. ■

We now extend Lemma 35 to all grids with B cells rather than just those that are sub-grids of the master grid Γ . The proof of this lemma relies on an information-theoretic result proven in Appendix B that bounds the difference in mutual information between two distributions that can be obtained from each other by moving a small amount of probability mass.

Lemma 36 Let $\Pi = (\Pi_X, \Pi_Y)$ and $\Psi = (\Psi_X, \Psi_Y)$ be random variables, and let Γ be a grid. Define $\varepsilon_{i,j}$ on Π_Γ and Ψ_Γ as in Lemma 35. Let G be any grid with B cells, and let δ (resp. d) represent the total probability mass of Π_Γ (resp. Ψ_Γ) falling in cells of Γ that are not contained in individual cells of G . We have that

$$|I(\Psi|_G) - I(\Pi|_G)| \leq O\left(\left(\sum_{i,j} |\varepsilon_{i,j}| + \delta + d\right) \log B + \delta \log(1/\delta) + d \log(1/d)\right)$$

provided that the $|\varepsilon_{i,j}|$ are bounded away from 1 and that $d, \delta \leq 1/2$.

Proof In the proof below, we use the convention that for any two grids G and G' and any random variable Z , the expression $\Delta^Z(G, G')$ denotes $|I(Z|_G) - I(Z|_{G'})|$.

Consider the grid G' obtained by replacing every horizontal or vertical line in G that is not in Γ with a closest line in Γ . The grid G' is clearly a sub-grid of Γ . Moreover, $\Pi|_{G'}$ (resp. $\Psi|_{G'}$) can be obtained from $\Pi|_G$ (resp. $\Psi|_G$) by moving at most δ (resp. d) probability mass. This can be shown to imply that

$$\Delta^\Pi(G, G') \leq O(\delta \log(1/\delta) + \delta \log B) \quad \text{and} \quad \Delta^\Psi(G', G) \leq O(d \log(1/d) + d \log B).$$

The proof of this information-theoretic fact is self-contained and so we defer it to Proposition 40 in Appendix B, as it is more central to the arguments presented there.

With $\Delta^\Phi(G, G')$ and $\Delta^\Psi(G', G)$ bounded in terms of δ and d , we can bound $|I(\Psi|_G) - I(\Phi|_G)|$ using the triangle inequality by comparing it with

$$\Delta^\Pi(G, G') + |I(\Pi|_{G'}) - I(\Psi|_G)| + \Delta^\Psi(G', G)$$

and bounding the middle term using Lemma 35, since $G' \subset \Gamma$. ■

We now use the fact that the variables $\varepsilon_{i,j}$ defined in Lemma 35 are small with high probability to give a concrete bound on the bias of $I(D|_G)$ that is uniform over all k -by- ℓ grids G and that holds with high probability. It is useful at this point to recall that, given a distribution (X, Y) , an equipartition of (X, Y) is a grid G such that all the rows of $(X, Y)|_G$ have the same probability mass, and all the columns do as well.

Lemma 37 Let D_n be a sample of size n from the distribution of a pair (X, Y) of jointly distributed random variables. For any $\alpha \geq 0$, any $\varepsilon > 0$, and any integers $k, \ell > 1$, we have that for all n

$$|I(D_n|_G) - I((X, Y)|_G)| \leq O\left(\frac{\log(k\ell)}{C(n)^\alpha} + \frac{\log(k\ell n)}{n^{\varepsilon/4}}\right)$$

for every k -by- ℓ grid G with probability at least $1 - C(n)e^{-\Omega(n/C(n)^{1+2\alpha})}$, where $C(n) = k\ell n^{\varepsilon/2}$.

Proof Fix n , and let Γ be an equipartition of (X, Y) into $k\ell n^{\varepsilon/4}$ rows and $\ell n^{\varepsilon/4}$ columns. $C(n)$ is now the number of cells in Γ . Lemma 36, with $\Pi = (X, Y)$ and $\Psi = D$, shows that $|I(D|_G) - I((X, Y)|_G)|$ is at most

$$O\left(\left(\sum_{i,j} |\varepsilon_{i,j}| + \delta + d\right) \log(k\ell) + \delta \log(1/\delta) + d \log(1/d)\right)$$

provided the $\varepsilon_{i,j}$ have absolute value bounded away from 1, and provided that $d, \delta \leq 1/2$.

The remainder of the proof proceeds as follows. We first show that the $\varepsilon_{i,j}$ are small with high probability. This will both show that the lemma's requirement on the $\varepsilon_{i,j}$ holds and allow us to bound the sum in the inequality above. We will then use our bound on the $\varepsilon_{i,j}$ to bound d in terms of δ . Finally, we will bound δ using the fact that the number of rows and columns in Γ increases with n . This will give us that $d, \delta \leq 1/2$ and allow us to bound the rest of the terms in the expression above.

Bounding the $\varepsilon_{i,j}$: We bound the $\varepsilon_{i,j}$ using a multiplicative Chernoff bound. Let $\pi_{i,j}$ and $\psi_{i,j}$ represent the probability mass functions of $(X, Y)|_\Gamma$ and $D|_\Gamma$ respectively. We write

$$\begin{aligned} \mathbf{P}(|\varepsilon_{i,j}| \geq \delta) &= \mathbf{P}(\pi_{i,j}(1 - \delta) \leq \psi_{i,j} \leq \pi_{i,j}(1 + \delta)) \\ &\leq e^{-\Omega(n\pi_{i,j}\delta^2)} \end{aligned}$$

since $\psi_{i,j}$ is a sum of n i.i.d. Bernoulli random variables and $\mathbf{E}(\psi_{i,j}) = n\pi_{i,j}$. (See, e.g., Mitzenmacher and Upfal 2005.) Setting $\delta = \sqrt{\pi_{i,j}/C(n)^{1/2+\alpha}}$ yields

$$\mathbf{P}\left(|\varepsilon_{i,j}| \geq \frac{\sqrt{\pi_{i,j}}}{C(n)^{1/2+\alpha}}\right) \leq e^{-\Omega(n/C(n)^{1+2\alpha})}.$$

A union bound over the pairs (i, j) then gives that, with the desired probability, the above bound on $|\varepsilon_{i,j}|$ holds for all i, j .

Bounding $\sum |\varepsilon_{i,j}|$: The bound on the $\varepsilon_{i,j}$ implies that

$$\begin{aligned} \sum_i |\varepsilon_{i,j}| &\leq \frac{1}{C(n)^{1/2+\alpha}} \sum_{i,j} \sqrt{\pi_{i,j}} \\ &\leq \frac{1}{C(n)^{1/2+\alpha}} \sqrt{C(n)} \\ &\leq \frac{1}{C(n)^\alpha} \end{aligned}$$

where the second line follows from the fact that the function $\sum \sqrt{\pi_{i,j}}$ is symmetric and concave and therefore, when restricted to the hyperplane $\sum \pi_{i,j} = 1$, must achieve its maximum when $\pi_{i,j} = 1/C(n)$ for all i, j .

Bounding d in terms of δ : We use our bound on the $\varepsilon_{i,j}$ to bound d . We do so by observing that it implies

$$\psi_{i,j} \leq \pi_{i,j} \left(1 + \frac{\sqrt{\pi_{i,j}}}{C(n)^{1/2+\alpha}}\right) = \pi_{i,j} + \frac{\pi_{i,j}^{3/2}}{C(n)^{1/2+\alpha}} \leq \pi_{i,j} + \frac{\pi_{i,j}}{C(n)^{1/2+\alpha}} \leq 2\pi_{i,j}$$

since $\pi_{i,j} \leq 1$ and $C(n) \geq 1$.

The connection to d comes from the fact that for any column j of Γ , this means that

$$\psi_{i,*} = \sum_i \psi_{i,j} \leq 2 \sum_i \pi_{i,j} = 2\pi_{*,j}.$$

This also applies to the sums across rows. Since d is a sum of terms of the form $\psi_{i,*}$ and $\psi_{i,*}$ for j in some index set J and i in an index set I , and δ is a sum of terms of the form $\pi_{*,j}$ and $\pi_{i,*}$ with the same index sets, we therefore get that $d \leq 2\delta$.

Bounding δ and obtaining the result: To bound δ , we observe that because G has at most $\ell - 1$ vertical lines and $k - 1$ horizontal lines, we have

$$\delta \leq \frac{\ell}{\ell n^{\varepsilon/4}} + \frac{k}{k n^{\varepsilon/4}} \leq \frac{2}{n^{\varepsilon/4}}.$$

This bound on δ allows us to bound the terms involving d and δ by

$$\delta + d \leq O\left(\frac{1}{n^{\varepsilon/4}}\right), \quad \delta \log\left(\frac{1}{\delta}\right) + d \log\left(\frac{1}{d}\right) \leq O\left(\frac{\log n}{n^{\varepsilon/4}}\right).$$

Combining all of the bounds gives the desired result. \blacksquare

Our final lemma shows that as long as $B(n)$ doesn't grow too fast, the bound from the previous lemma yields a uniform bound on the entire sample characteristic matrix. This is done by specifying an error threshold for which Lemma 37 yields a bound that holds with high probability, and then invoking a union bound.

Lemma 38 *Let D_n be a sample of size n from the distribution of a pair (X, Y) of jointly distributed random variables. For every $B(n) = O(n^{1-\varepsilon})$, there exists an $a > 0$ such that for sufficiently large n ,*

$$\left| \widehat{M}(D_n)_{k,\ell} - M(X, Y)_{k,\ell} \right| \leq O\left(\frac{1}{n^a}\right)$$

holds for all $k\ell \leq B(n)$ with probability $P(n) = 1 - o(1)$, where $\widehat{M}(D_n)_{k,\ell}$ is the k, ℓ -th entry of the sample characteristic matrix and $M(X, Y)_{k,\ell}$ is the k, ℓ -th entry of the population characteristic matrix of (X, Y) .

Proof Fix k, ℓ , and any α satisfying $0 < \alpha < \varepsilon/(4 - 2\varepsilon)$. Lemma 37 implies that with high probability the difference $|\widehat{M}(D_n)_{k,\ell} - M_{k,\ell}|$ is at most

$$\begin{aligned} O\left(\frac{\log(k\ell)}{C(n)^\alpha} + \frac{\log(k\ell n)}{n^{\varepsilon/4}}\right) &\leq O\left(\frac{\log n}{C(n)^\alpha} + \frac{\log n}{n^{\varepsilon/4}}\right) \\ &\leq O\left(\frac{\log n}{n^{\alpha\varepsilon/2}} + \frac{\log n}{n^{\varepsilon/4}}\right) \end{aligned}$$

where the first inequality comes from $k\ell \leq B(n)$ and second is because $C(n) = k\ell n^{\varepsilon/2} \geq n^{\varepsilon/2}$. This bound is at most $O(1/n^a)$ for every $a < \min\{\alpha\varepsilon/2, \varepsilon/4\}$, as desired. It remains only to show that the bound holds with high probability across all $k\ell \leq B(n)$.

Lemma 37 states that the probability our bound holds for one fixed pair (k, ℓ) is at least

$$1 - C(n)e^{-\Omega(n/C(n)^{1+2\alpha})} \geq 1 - O(n)e^{-\Omega(n^u)}$$

for some positive u . This is because $C(n) \leq B(n)n^{\varepsilon/2} \leq O(n^{1-\varepsilon/2})$ for large n , and so our choice of α ensures that $C(n)^{1+2\alpha} = O(n^{1-u})$ for some $u > 0$.

We can then perform a union bound over all pairs $k\ell \leq B(n)$: since the number of such pairs can be bounded by a polynomial in n , we have that the desired condition is satisfied for all $k\ell \leq B(n)$ with probability approaching 1. \blacksquare

We are now ready to prove the main result.

Theorem *Let $f : m^\infty \rightarrow \mathbb{R}$ be uniformly continuous, and assume that $f \circ r_i \rightarrow f$ pointwise. Then for every random variable (X, Y) , we have*

$$(f \circ r_{B(n)})\left(\widehat{M}(D_n)\right) \rightarrow f(M(X, Y))$$

in probability where D_n is a sample of size n from the distribution of (X, Y) , provided $\omega(1) < B(n) \leq O(n^{1-\varepsilon})$ for some $\varepsilon > 0$.

Proof Let N denote $B(n)$, let $M_N = r_N(M)$, and let $\widehat{M}_N(D_n) = r_N(\widehat{M}(D_n))$. We begin by writing

$$\begin{aligned} \left| f\left(\widehat{M}_N(D_n)\right) - f(M) \right| &\leq \left| f\left(\widehat{M}_N(D_n)\right) - f(M_N) \right| + \left| f(M_N) - f(M) \right| \\ &= \left| f\left(\widehat{M}_N(D_n)\right) - f(M_N) \right| + \left| (f \circ r_N)(M) - f(M) \right| \end{aligned}$$

and observing that as $n \rightarrow \infty$, the second term vanishes by the pointwise convergence of $f \circ r_i$ and the fact that $B(n) > \omega(1)$. It therefore suffices to show that the first term converges to zero in probability. Since f is uniformly continuous, we can establish this via a simple adaptation of the continuous mapping theorem, which says that if the sequence of random variables $R_n \rightarrow R$ in probability, and g is continuous, then $g(R_n) \rightarrow g(R)$ in probability. We replace R with a second sequence, and replace continuity with uniform continuity.

Let $\|\cdot\|$ denote the supremum norm on m^∞ , and fix any $z > 0$. Then, for any $\delta > 0$, define

$$C_\delta = \{A \in m^\infty : \exists A' \in m^\infty \text{ s.t. } \|A - A'\| < \delta, \|f(A) - f(A')\| > z\}.$$

This is the set of matrices $A \in m^\infty$ for which it is possible to find, within a δ -neighborhood of A , a second matrix that f maps to more than z away from $f(A)$. Because f is uniformly continuous, there exists a δ^* sufficiently small so that $C_{\delta^*} = \emptyset$.

Suppose that $|f(\widehat{M}_N(D_n)) - f(M_N)| > z$. This means that either $\|\widehat{M}_N(D_n) - M_N\| > \delta^*$, or $M_N \in C_{\delta^*}$. The latter option is impossible since $C_{\delta^*} = \emptyset$, and Lemma 38 tells us that $\mathbf{P}\left(\|\widehat{M}_N(D_n) - M_N\| > \delta^*\right) \rightarrow 0$ as n grows. We therefore have that

$$\left| f\left(\widehat{M}_N(D_n)\right) - f(M_N) \right| \rightarrow 0$$

in probability, as desired. \blacksquare

Appendix B. Proof of Theorem 8

In this appendix we prove Theorem 8, reproduced below.

Theorem Let $\mathcal{P}(\mathbb{R}^2)$ denote the space of random variables supported on \mathbb{R}^2 equipped with the metric of statistical distance. The map from $\mathcal{P}(\mathbb{R}^2)$ to m^∞ defined by $(X, Y) \mapsto M(X, Y)$ is uniformly continuous.

The proposition below begins our argument with the simple observation that the family of maps consisting of applying any finite grid to some $(X, Y) \in \mathcal{P}(\mathbb{R}^2)$ is uniformly equicontinuous. The reason this holds is that $(X, Y)|_G$ is a deterministic function of (X, Y) , and deterministic functions cannot increase statistical distance.

Proposition 39 Let \mathbb{G} be the set of all finite grids. The family $\{(X, Y) \mapsto (X, Y)|_G : G \in \mathbb{G}\}$ is uniformly equicontinuous on $\mathcal{P}(\mathbb{R}^2)$.

Proof To establish uniform equicontinuity, we need to show that, given some $(X, Y) \in \mathcal{P}(\mathbb{R}^2)$ and some $\varepsilon > 0$, we can choose δ to satisfy the continuity condition in a way that does not depend on G or on (X, Y) . But because deterministic functions cannot increase statistical distance, we have that if $(X, Y), (X', Y') \in \mathcal{P}$ are at most ε apart then

$$\Delta((X, Y)|_G, (X', Y')|_G) \leq \Delta((X, Y), (X', Y')) = \varepsilon$$

where Δ denotes statistical distance. Choosing $\delta = \varepsilon$ therefore gives the result. \blacksquare

At this point it is tempting to try to use continuity properties of discrete mutual information to obtain uniform continuity of the characteristic matrix. And indeed, this strategy does yield that each *individual* entry of the characteristic matrix is a uniformly continuous function. However, to obtain continuity of the entire (infinite) characteristic matrix we need to make a statement about all grid resolutions simultaneously. This is not straightforward because mutual information is only uniformly continuous for a fixed grid resolution, and the family $\{(X, Y) \mapsto I((X, Y)|_G) : G \in \mathbb{G}\}$ is in fact not even equicontinuous.

The normalization in the definition of MIC_* is what allows us to establish the uniform continuity of the characteristic matrix despite this problem. To see why, suppose we have a distribution over a k -by- ℓ grid and we are allowed to move at most δ away in statistical distance for some small δ . The largest change in discrete mutual information that this can cause indeed increases as we increase k and ℓ . However, it turns out that we can bound the extent of this ‘‘non-uniformity’’: the proposition below shows that as we move away from a distribution, the discrete mutual information can change only proportionally to the amount of mass we move, with the proportionality constant bounded by $\log \min\{k, \ell\}$. Because $\log \min\{k, \ell\}$ is the quantity by which we regularize the entries of the characteristic matrix, this is exactly enough to make the normalized matrix continuous. This proposition is the technical heart of our continuity result. And as we show in Corollary 11 when we demonstrate the non-continuity of the non-normalized characteristic matrix mutual information, our bound is tight.

Proposition 40 Let $I_{k,\ell} : \mathcal{P}(\{1, \dots, k\} \times \{1, \dots, \ell\}) \rightarrow \mathbb{R}$ denote the discrete mutual information function on k -by- ℓ grids. For $0 < \delta \leq 1/4$, the maximal change in $I_{k,\ell}$ over any subset of $\mathcal{P}(\{1, \dots, k\} \times \{1, \dots, \ell\})$ of diameter δ (in statistical distance) is

$$O\left(\delta \log\left(\frac{1}{\delta}\right) + \delta \log \min\{k, \ell\}\right).$$

Proof Without loss of generality, assume $k \leq \ell$, so that $\log \min\{k, \ell\} = \log k$. Let (X, Y) and (X', Y') be two random variables distributed over $\{1, \dots, k\} \times \{1, \dots, \ell\}$ that are at most δ apart in statistical distance. Using $I(X, Y) = H(Y) - H(Y|X)$, we can express the difference between the mutual information of these two pairs of random variables as

$$|I(X, Y) - I(X', Y')| \leq |H(Y) - H(Y')| + |H(Y|X) - H(Y'|X')|.$$

We now use Lemma 55, which relates movement of probability mass to changes in entropy and is proven in Appendix L, to separately bound each of the terms on the right hand side. Straightforward application of the lemma to $|H(Y) - H(Y')|$ shows that it is at most $2H_b(2\delta) + 3\delta \log k$, where $H_b(\cdot)$ is the binary entropy function. Since $H_b(x) \leq O(x \log(1/x))$ for x small, this is $O(\delta \log(1/\delta) + \delta \log k)$.

Bounding the term with the conditional entropies is more involved. Let $p_x = \mathbf{P}(X = x)$, and let $p'_x = \mathbf{P}(X' = x)$. We have

$$\begin{aligned} |H(Y|X) - H(Y'|X')| &= \sum_x |p_x H(Y|X = x) - p'_x H(Y'|X' = x)| \\ &\leq \sum_x (p_x |H(Y|X = x) - H(Y'|X' = x)| + \\ &\quad |p'_x - p_x| H(Y'|X' = x)) \\ &= \sum_x p_x |H(Y|X = x) - H(Y'|X' = x)| + \sum_x |p'_x - p_x| \log k \\ &\leq \sum_x p_x |H(Y|X = x) - H(Y'|X' = x)| + \delta \log k \end{aligned} \tag{2}$$

where the last line is because $\sum_x |p_x - p'_x| \leq \delta$ and $H(Y'|X' = x) \leq \log k$.

Now let $\delta_{x,+}$ be the magnitude of all the probability mass entering any cell in column x , let $\delta_{x,-}$ be the magnitude of all the probability mass leaving any cell in column x , and let $\delta_x = \delta_{x,+} + \delta_{x,-}$. Using this notation, we can again apply Lemma 55 to obtain

$$\begin{aligned} \sum_x p_x |H(Y|X = x) - H(Y'|X' = x)| &\leq \sum_x p_x \left(2H_b\left(\frac{2\delta_x}{p_x}\right) + 3\delta_x \log k \right) \\ &= 2 \sum_x p_x H_b\left(\frac{2\delta_x}{p_x}\right) + 3 \sum_x \delta_x \log k \\ &\leq 2 \sum_x p_x H_b\left(\frac{2\delta_x}{p_x}\right) + 3\delta \log k \\ &\leq 2H_b(2\delta) + 3\delta \log k \end{aligned}$$

where the last line is by application of Lemma 52 from the appendix, which bounds weighted sums of binary entropies.

Combining this with Line (2) gives that

$$|H(Y|X) - H(Y'|X')| \leq 2H_\delta(2\delta) + 4\delta \log k$$

which, together with the bound on $|H(Y) - H(Y')|$ and the fact that $H_\delta(X) \leq O(x \log(1/x))$ for x small, gives the result. \blacksquare

Having bounded the extent to which variation in mutual information depends on grid resolution, we are now ready to show the uniform continuity of the characteristic matrix.

Theorem Let $\mathcal{P}(\mathbb{R}^2)$ denote the space of random variables supported on \mathbb{R}^2 equipped with the metric of statistical distance. The map from $\mathcal{P}(\mathbb{R}^2)$ to m^∞ defined by $(X, Y) \mapsto M(X, Y)$ is uniformly continuous.

Proof We complete the proof in three steps. First, we show that a certain family of functions F is uniformly equicontinuous. Second, we use this to show that a different family F' consisting of functions of the form $\sup_{g \in A} g$ with $A \subset F$ is uniformly equicontinuous. Finally, we argue that since the entries of $M(X, Y)$ consist of the functions in F' , this is sufficient to establish the result.

Define

$$F = \left\{ (X, Y) \mapsto \frac{I_{k,\ell}((X, Y)|_G)}{\log \min\{k, \ell\}} : k, \ell \in \mathbb{Z}_{>1}, G \in \mathcal{G}(k, \ell) \right\}.$$

F is uniformly equicontinuous by the following argument. Given some $\varepsilon > 0$, we know (Proposition 39) that for any (X', Y') in an ε -ball around (X, Y) , $(X', Y')|_G$ will remain within ε of $(X, Y)|_G$ for any G . Proposition 40 then tells us that if ε is sufficiently small then the distance between $I_{k,\ell}((X', Y')|_G)$ and $I_{k,\ell}((X, Y)|_G)$ will be at most

$$O(\varepsilon \log(1/\varepsilon) + \varepsilon \log \min\{k, \ell\}).$$

After the normalization, this becomes at most $O(\varepsilon(\log(1/\varepsilon) + 1))$, which goes to zero (uniformly with respect to (X, Y)) as ε approaches zero, as desired.

Next, define

$$F' = \{(X, Y) \mapsto M(X, Y)_{k,\ell} : k, \ell \in \mathbb{Z}_{>1}\}.$$

Each map in F' is of the form $\sup_{g \in A} g$ for some $A \subset F$. Therefore, for a given $\varepsilon > 0$, whatever δ establishes the uniform equicontinuity for F can be used to establish continuity of all the functions in F' . (To see this: $\sup_{g \in A} g$ can't increase by more than ε if no g increases by more than ε , and $\sup_{g \in A} g$ is also lower bounded by any of the g 's, so it can't decrease by more than ε either.) Since we can use the same δ for all of the maps in F' , they therefore form a uniformly equicontinuous family.

Finally, the δ provided by the uniform equicontinuity of F' also ensures that $M(X', Y')$ is within ε of $M(X, Y)$ in the supremum norm, thus giving the uniform continuity of $(X, Y) \mapsto M(X, Y)$. \blacksquare

Appendix C. Proof of Proposition 10

Theorem For some function $N(k, \ell)$, let M^N be the characteristic matrix with normalization N , i.e.,

$$M^N(X, Y) = \frac{I^*((X, Y), k, \ell)}{N(k, \ell)}.$$

If $N(k, \ell) = o(\log \min\{k, \ell\})$ along some infinite path in $\mathbb{N} \times \mathbb{N}$, then M^N and $\sup M^N$ are not continuous as functions of $\mathcal{P}([0, 1] \times [0, 1]) \subset \mathcal{P}(\mathbb{R}^2)$.

Proof Consider a random variable Z uniformly distributed on $[0, 1/2]^2$. Because Z exhibits statistical independence, $I^*(Z, k, \ell)$ is zero for all k, ℓ . Now define Z_ε to be uniformly distributed on $[0, 1/2]^2$ with probability $1 - \varepsilon$ and uniformly distributed on the line from $(1/2, 1/2)$ to $(1, 1)$ with probability ε .

We lower-bound $I^*(Z_\varepsilon, k, \ell)$. Without loss of generality suppose that $k \leq \ell$, and consider a grid that places all of $[0, 1/2]^2$ into one cell and uniformly partitions the set $[1/2, 1]^2$ into $k - 1$ rows and $k - 1$ columns. By considering just the rows/columns in the set $[1/2, 1]^2$ we see that this grid gives a mutual information of at least $\varepsilon \log(k - 1)$. Thus, we have that for all k, ℓ ,

$$I^*(Z_\varepsilon, k, \ell) \geq \varepsilon \log \min\{k - 1, \ell - 1\}.$$

This implies that the limit of $M^N(Z_\varepsilon)$ along P is ∞ , and so the distance between $M^N(Z)$ and $M^N(Z_\varepsilon)$ in the supremum norm is infinite. \blacksquare

Appendix D. Proof of Theorem 16

Theorem Let M be a population characteristic matrix. Then $M_{k,\uparrow}$ equals

$$\max_{P \in \mathcal{P}(k)} \frac{I(X, Y|_P)}{\log k}$$

where $\mathcal{P}(k)$ denotes the set of all partitions of size at most k .

Proof Define

$$M_{k,\uparrow}^* = \max_{P \in \mathcal{P}(k)} \frac{I(X, Y|_P)}{\log k}.$$

We wish to show that $M_{k,\uparrow}^*$ is in fact equal to $M_{k,\uparrow}$. To show that $M_{k,\uparrow} \leq M_{k,\uparrow}^*$ we observe that for every k -by- ℓ grid $G = (P, Q)$, where P is a partition into rows and Q is a partition into columns, the data processing inequality gives $I((X, Y)|_G) \leq I(X, Y|_P)$. Thus $M_{k,\ell} \leq M_{k,\uparrow}^*$ for $\ell \geq k$, implying that

$$M_{k,\uparrow} = \lim_{\ell \rightarrow \infty} M_{k,\ell} \leq M_{k,\uparrow}^*.$$

It remains to show that $M_{k,\uparrow}^* \leq M_{k,\uparrow}$. To do this, we let P be any partition into k rows, and we define Q_ℓ to be an equipartition into ℓ columns. We let

$$M_{k,\ell,P}^* = \frac{I(X|_{Q_\ell}, Y|_P)}{\log k}.$$

Since $M_{k,\ell}^* \leq M_{k,\ell}$ when $\ell \geq k$, we have that for all P

$$\frac{I(X, Y|P)}{\log k} = \lim_{\ell \rightarrow \infty} M_{k,\ell}^* \leq \lim_{\ell \rightarrow \infty} M_{k,\ell} = M_{k,\uparrow}$$

which gives that

$$M_{k,\uparrow}^* = \sup_P \frac{I(X, Y|P)}{\log k} \leq M_{k,\uparrow}$$

as desired. \blacksquare

Appendix E. Proof of Theorem 18

Theorem *Given a random variable (X, Y) , $M_{k,\uparrow}$ (resp. $M_{\uparrow,\ell}$) is computable to within an additive error of $O(k\varepsilon \log(1/(k\varepsilon))) + E$ (resp. $O(\ell\varepsilon \log(1/(\ell\varepsilon))) + E$) in time $O(kT(E)/\varepsilon)$ (resp. $O(\ell T(E)/\varepsilon)$), where $T(E)$ is the time required to numerically compute the mutual information of a continuous distribution to within an additive error of E .*

Proof Without loss of generality we prove the claim only for $M_{k,\uparrow}$. Given $0 < \varepsilon < 1$, we would like a partition into rows P of size at most k such that $I(X, Y|P)$ is maximized. We would like to use OPTIMIZEXAXIS for this purpose, but while our search problem is continuous, OPTIMIZEXAXIS can only perform a discrete search over sub-partitions of some master partition Π . We therefore set Π to be an equipartition into $1/\varepsilon$ rows and show that this gets us close enough to achieve the desired result.

With Π as described, the OPTIMIZEXAXIS provides in time $O(kT(E)/\varepsilon)$ a partition P_0 into at most k rows such that $I(X, Y|P_0)$ is maximized, subject to $P_0 \subset \Pi$, to within an additive error of E . To prove the claim then, we must show that the loss we incur by restricting to sub-partitions of Π costs us at most $O(k\varepsilon \log(1/(k\varepsilon)))$. In other words, we must show that

$$I(X, Y|P) - I(X, Y|P_0) \leq O(k\varepsilon)$$

where P is an optimal partition into rows. Note that we have omitted the absolute value above, since by the optimality of P , $I(X, Y|P) \geq I(X, Y|P_0)$ always.

We prove the desired bound by showing that there exists some $P' \subset \Pi$ such that the mutual information of $(X, Y|P')$ is $O(k\varepsilon \log(1/(k\varepsilon)))$ -close to that achieved with $(X, Y|P)$. Since $P' \subset \Pi$ gives us that $I(X, Y|P_0) \geq I(X, Y|P')$, we may then conclude that $I(X, Y|P) - I(X, Y|P_0)$ is at most $O(k\varepsilon \log(1/(k\varepsilon)))$.

We construct P' by simply replacing every horizontal line in P with a horizontal line in Π closest to it. Since there are at most $k - 1$ horizontal lines in P , and each such line is contained in a row of Π containing $1/\varepsilon$ probability mass, performing this operation moves at most $(k - 1)\varepsilon$ probability mass. In other words, the statistical distance between $(X, Y|P)$ and $(X, Y|P')$ is at most $(k - 1)\varepsilon \leq k\varepsilon$. Thus, for sufficiently small ε , Proposition 40, proven in Appendix B, can be used to show that

$$|I(X, Y|P) - I(X, Y|P')| \leq O\left(k\varepsilon \log\left(\frac{1}{k\varepsilon}\right) + k\varepsilon \log\left(\frac{1}{\varepsilon}\right)\right)$$

which yields the desired result. \blacksquare

Remark 41 *We do not explore here the details of the numerical integration associated with the above theorem, since the error introduced by the numerical integration is independent of the algorithm being proposed. However, standard numerical integration methods can be used to make this error arbitrarily small with an understood complexity tradeoff (see, e.g., Stoer and Bittenschi 1980).*

Appendix F. Proof of Theorem 21

Theorem *Let (X, Y) be jointly distributed random variables. Then $\partial[M] = \partial M$.*

Proof Without loss of generality, we show that $[M]_{k,\uparrow} = M_{k,\uparrow}$. Fix any partition into rows P . If Q_ℓ is an equipartition into ℓ columns then

$$\lim_{\ell \rightarrow \infty} I(X|_{Q_\ell}, Y|P) = I(X, Y|P),$$

because the continuous mutual information equals the limit of the discrete mutual information with increasingly fine partitions. (See, e.g., Chapter 8 of Cover and Thomas 2006 for a proof of this.) This means that, letting $P(k)$ denote the set of all partitions of size at most k , we have

$$[M]_{k,\uparrow} = \max_{P \in P(k)} \frac{I(X, Y|P)}{\log k} = M_{k,\uparrow}$$

where the second equality follows from Proposition 16. \blacksquare

Appendix G. Consistency of MIC_ε in Estimating MIC_{*}

The consistency of MIC_ε for estimating MIC_{*} can be established using the same technical lemmas that we used to show that MIC_ε \rightarrow MIC_{*}. Specifically, we can use Lemma 37, which bounds the difference, for all k -by- ℓ grids G , between the sample quantity $I(D_n|G)$ and the population quantity $I((X, Y)|G)$ with high probability, where D_n is a sample of size n from (X, Y) . That lemma yields the following fact about the sample equicharacteristic matrix, whose proof is similar to that of Lemma 38.

Lemma 42 *Let D_n be a sample of size n from the distribution of a pair (X, Y) of jointly distributed random variables. For every $B(n) = O(n^{1-\varepsilon})$, there exists an $a > 0$ such that for sufficiently large n ,*

$$\left| \widehat{[M]}(D_n)_{k,\ell} - [M](X, Y)_{k,\ell} \right| \leq O\left(\frac{1}{n^a}\right)$$

holds for all $k\ell \leq B(n)$ with probability $P(n) = 1 - o(1)$, where $\widehat{[M]}(D_n)_{k,\ell}$ is the k, ℓ -th entry of the sample equicharacteristic matrix and $[M](X, Y)_{k,\ell}$ is the k, ℓ -th entry of the population equicharacteristic matrix of (X, Y) .

In the case of MIC, we proceeded to apply abstract continuity considerations to obtain our consistency theorem (Theorem 6) from a result analogous to the above lemma. A similar argument shows us that, in the case of the equicharacteristic matrix as well, we can estimate a large class of functions of the matrix in the same way. This is stated formally in the theorem below. As before, we let m^∞ be the space of infinite matrices equipped with the supremum norm, and given a matrix A the projection r_i zeros out all the entries $A_{k,\ell}$ for which $k\ell > i$.

Theorem *Let $f : m^\infty \rightarrow \mathbb{R}$ be uniformly continuous, and assume that $f \circ r_i \rightarrow f$ pointwise. Then for every random variable (X, Y) , we have*

$$(f \circ r_{B(n)}) \left(\widehat{M}(D_n) \right) \rightarrow f(M(X, Y))$$

in probability where D_n is a sample of size n from the distribution of (X, Y) , provided $\omega(1) < B(n) \leq O(n^{1-\varepsilon})$ for some $\varepsilon > 0$.

Appendix H. The EQUICHARCLUMP Algorithm

In Theorem 28, we sketched an algorithm called EQUICHARCLUMP for approximating the sample equicharacteristic matrix that is more efficient than the naive computation. In this appendix, we describe the algorithm in detail, bound its runtime, and show that it indeed yields a consistent estimator of MIC_* from finite samples as well as a consistent independence test when used to compute the total information coefficient. We then present some empirical results characterizing the sensitivity of the algorithm to its speed-versus-optimality parameter c .

The results in this section can be summarized as follows: let (X, Y) be a pair of jointly distributed random variables, and let D_n be a sample of size n from the distribution of (X, Y) . For every $c \geq 1$, there exists a matrix $\{\widehat{M}\}^c(D_n)$ such that

1. There exists an algorithm EQUICHARCLUMP for computing $r_B(\{\widehat{M}\}^c(D_n))$ in time $O(n + B^{5/2})$, which equals $O(n + n^{5(1-\varepsilon)/2})$ when $B(n) = O(n^{1-\varepsilon})$.

2. The function

$$\widehat{\text{MIC}}_{e,B}(\cdot) = \max_{k\ell \leq B(n)} \{\widehat{M}\}^c(\cdot)_{k,\ell}$$

is a consistent estimator of MIC_* provided $\omega(1) < B(n) \leq O(n^{1-\varepsilon})$ for some $\varepsilon > 0$.

3. The function

$$\widehat{\text{TIC}}_{e,B}(\cdot) = \sum_{k\ell \leq B(n)} \{\widehat{M}\}^c(\cdot)_{k,\ell}$$

yields a consistent right-tailed test of independence provided $\omega(1) < B(n) \leq O(n^{1-\varepsilon})$ for some $\varepsilon > 0$

We will prove these results in order.

H.1 Algorithm Description and Analysis of Runtime

We begin by describing the algorithm and bounding its runtime simultaneously. As in the proof of Theorem 27, we bound the runtime required to approximately compute only the k, ℓ -th entries of $\{\widehat{M}\}^c(D_n)$ satisfying $k \leq \ell, k\ell \leq B$. To do this, we analyze two portions of $\{\widehat{M}\}^c(D_n)$ separately: we first consider the case $\ell \geq \sqrt{B}$, in which we must compute the entries corresponding to all the pairs $\{(2, \ell), \dots, (B/\ell, \ell)\}$. We then consider $\ell < \sqrt{B}$, in which case we need only compute the entries $\{(2, \ell), \dots, (\ell, \ell)\}$ since the additional pairs would all have $k > \ell$.

For the case of $\ell \geq \sqrt{B}$, as in the previous theorem we can simultaneously compute using OPTIMIZEXAXIS the entries corresponding to all the pairs $\{(2, \ell), \dots, (B/\ell, \ell)\}$ in time $O(|\Pi|^2 B/\ell) = O(|\Pi|^2 B)$, which equals $O(c^2 B^3/\ell^2)$ when we set Π to be an equipartition of size cB/ℓ . Doing this for $\ell = \sqrt{B}, \dots, B/2$ gives a contribution of the following order to the runtime.

$$\begin{aligned} O(c^2 B^3) \sum_{\ell=\sqrt{B}}^{B/2} \frac{1}{\ell^2} &= O(c^2 B^3) O\left(\frac{1}{\sqrt{B}}\right) \\ &= O(c^2 B^{5/2}) \end{aligned}$$

For the case of $\ell < \sqrt{B}$, we can simultaneously compute using OPTIMIZEXAXIS the entries corresponding to all the pairs $\{(2, \ell), \dots, (\ell, \ell)\}$ in time $O(|\Pi|^2 \ell^2)$ which equals $O(c^2 \ell^4) \leq O(c^2 B^2)$ when we set Π to be an equipartition of size $c\ell$. Summing over the $O(\sqrt{B})$ possible values of ℓ with $\ell < \sqrt{B}$ gives an upper bound of $O(c^2 B^{5/2})$.

H.2 Consistency

Let (X, Y) be a pair of jointly distributed random variables. For a sample D_n of size n from the distribution of (X, Y) and a speed-versus-optimality parameter $c \geq 1$, let $\{\widehat{M}\}^c(D_n)$ denote the matrix computed by EQUICHARCLUMP. (Notice the use of curly braces to differentiate this from the sample equicharacteristic matrix $[M]$.) We show here that $\max_{k\ell \leq B(n)} \{\widehat{M}\}^c(D_n)_{k,\ell}$ is a consistent estimator of $\text{MIC}_*(X, Y)$, and correspondingly that $\sum_{k\ell \leq B(n)} \{\widehat{M}\}^c(D_n)_{k,\ell}$ yields a consistent independence test.

The key to both consistency results is that, though in calculating the k, ℓ -th entry of $\{\widehat{M}\}^c(D_n)$ the algorithm only searches for optimal partitions that are sub-partitions of some equipartition, the size of the equipartition used always grows as n, k , and ℓ grow large. Therefore, in the limit this additional restriction does not hinder the optimization. We present this argument by introducing a population object called the *clumped equicharacteristic matrix*. We observe that this matrix is the limit of the EQUICHARCLUMP procedure as sample size grows, and then show that the supremum and partial sums of this matrix have the necessary properties.

Definition 43 *Let (X, Y) be jointly distributed random variables and fix some $c \geq 1$. Let*

$$I^{(c*)}((X, Y), k, \ell) = \max_G I((X, Y)|_G)$$

where the maximum is over k -by- ℓ grids whose larger partition is an equipartition and whose smaller partition must be contained in an equipartition of size $c \cdot \max\{k, \ell\}$. The clumped equicharacteristic matrix of (X, Y) , denoted by $\{M\}^c(X, Y)$, is defined by

$$\{M\}^c(X, Y)_{k, \ell} = \frac{I^{c(\ast)}((X, Y), k, \ell)}{\log \min\{k, \ell\}}$$

Notice that curly braces differentiate the quantities $I^{c(\ast)}$ and $\{M\}^c$ defined above from the corresponding equicharacteristic matrix quantities $I^{(\ast)}$ and $[M]$.

The following two results, which we state without proof, characterize the convergence of the output of EQUICHARCLUMP to the clumped equicharacteristic matrix. These lemmas can be shown using Lemma 37, which simultaneously bounds the difference, for all k -by- ℓ grids G , between the sample quantity $I(D_n|G)$ and the population quantity $I((X, Y)|G)$ with high probability over the sample D_n of size n from (X, Y) .

Lemma 44 *Let D_n be a sample of size n from the distribution of a pair (X, Y) of jointly distributed random variables. For every $B(n) = O(n^{1-\varepsilon})$, there exists an $a > 0$ such that for sufficiently large n ,*

$$\left| \widehat{[M]}^c(D_n)_{k, \ell} - \{M\}^c(X, Y)_{k, \ell} \right| \leq O\left(\frac{1}{n^a}\right)$$

holds for all $k, \ell \leq \sqrt{B(n)}$ with probability $P(n) = 1 - o(1)$, where $\widehat{[M]}^c(D_n)$ denotes the matrix computed by the EQUICHARCLUMP algorithm with parameter c on the sample D_n .

Notice that the error bound provided by the above lemma holds not for $k\ell \leq B(n)$ as in the analogous Lemma 38 and Lemma 42, but rather for the smaller region defined by $k, \ell \leq \sqrt{B(n)}$. However, though we do not have uniform convergence outside the region $k, \ell \leq \sqrt{B(n)}$, we do nevertheless have pointwise convergence there, as stated below.

Lemma 45 *Fix $k, \ell \geq 2$. Let D_n be a sample of size n from the distribution of a pair (X, Y) of jointly distributed random variables. For every $B(n) > \omega(1)$, we have that*

$$\widehat{[M]}^c(D_n)_{k, \ell} \rightarrow \{M\}^c(X, Y)_{k, \ell}$$

in probability as n grows, where $\widehat{[M]}^c(D_n)$ denotes the matrix computed by the EQUICHARCLUMP algorithm with parameter c on the sample D_n .

H.2.1 CONSISTENCY FOR ESTIMATING MIC_*

The consistency of $\widehat{[M]}^c(D_n)$ for estimating MIC_* follows from the following property of the clumped equicharacteristic matrix $\{M\}^c$, for which we state a proof sketch.

Proposition 46 *Let (X, Y) be a pair of jointly distributed random variables. Then we have $\sup\{[M]^c(X, Y) = \text{MIC}_*(X, Y)$.*

Proof (Sketch) Let $\{M\}^c = \{M\}^c(X, Y)$, and let $M = M(X, Y)$ be the characteristic matrix. Fix k , and consider the limit $\{M\}_{k, \ell}^c$ as ℓ grows. The grid chosen for the k, ℓ -th entry when $\ell > k$ will contain an equipartition P_ℓ of size ℓ on the x -axis, and a partition Q_ℓ of size k on the y -axis that is optimal subject to the restriction that Q_ℓ be contained in an equipartition of size $c\ell$. As ℓ grows large, the equipartition P_ℓ on the first axis will become finer and finer until in the limit $X|_{P_\ell} \rightarrow X$. And the partition Q_ℓ will be chosen from a finer and finer equipartition, so that in the limit it approaches an unconditionally optimal partition Q of size k . The convergence of Q_ℓ to the optimal partition Q of size k can be shown to be uniform using Proposition 40. This implies that

$$\{M\}_{k, \uparrow}^c = \lim_{\ell \rightarrow \infty} \{M\}_{k, \ell}^c = \max_{P \in \mathcal{P}(k)} \frac{I(X, Y|P)}{\log k}$$

where $\mathcal{P}(k)$ denotes the set of all partitions of size at most k . Therefore, the boundary $\partial\{M\}^c$ of $\{M\}^c$ equals the boundary ∂M of M . Since $\text{MIC}_*(X, Y) = \sup \partial M$ (Theorem 15), this implies that

$$\sup\{M\}^c \geq \sup \partial\{M\}^c = \sup \partial M = \text{MIC}_*(X, Y).$$

On the other hand, $\{M\}^c \leq M$ element-wise since the optimization for the k, ℓ -th entry of $\{M\}^c$ is performed over a subset of the grids searched for the k, ℓ -th entry of M . This means that $\sup\{M\}^c \leq \sup M = \text{MIC}_*(X, Y)$. ■

This fact, together with the pointwise convergence of $\widehat{[M]}^c(D_n)$ to $\{M\}^c$, suffices to establish the consistency we seek via standard continuity arguments, which we give in the abstract lemma below. The lemma applies to a double-indexed sequence indexed by i and j : in our argument, the index i corresponds to position in the equicharacteristic matrix, and the index j corresponds to sample size. The sequence A corresponds to the output of the EQUICHARCLUMP algorithm, the sequence a corresponds to the clumped equicharacteristic matrix, and the sequence B corresponds to the sample equicharacteristic matrix.

Lemma 47 *Let $\{A_{ij}\}_{i,j=1}^\infty$ and $\{B_{ij}\}_{i,j=1}^\infty$ be sequences of random variables, and let $\{a_j\}_{j=1}^\infty$ be a non-stochastic sequence. Assume that the following conditions hold*

1. $A_{ij} \leq B_{ij}$ almost surely
2. For every i , $A_{ij} \rightarrow a_i$ in probability
3. $B_j^i = \max_{s \leq j} B_{is}$ satisfies $B_j^i \rightarrow \sup\{a_i\}$ in probability

Then $A_j^i = \max_{s \leq j} A_{is}$ converges in probability to $\sup\{a_i\}$ as well.

Proof Let $a = \sup\{a_j\}$. We give the proof for the case that $a < \infty$. However, it is easily adapted to the infinite case. We must show that for every $\varepsilon > 0$ and every $0 < p \leq 1$, there exists some N such that $\mathbf{P}(|A_j^i - a| < \varepsilon) > p$ for all $j \geq N$. By the definition of a , we know that there exists some k such that $|a_k - a| < \varepsilon/2$. Also, by the convergence of A_{ij} to a_i , there exists some m such that $\mathbf{P}(|A_{kj} - a_k| < \varepsilon/2) > 1 - p$ for all $j \geq m$. Thus, with probability at least $1 - p$, we have

$$\begin{aligned} |A_{kj} - a| &\leq |A_{kj} - a_k| + |a_k - a| \\ &\leq \varepsilon \end{aligned}$$

for all $j \geq m$.

Next, we observe that since $A'_j \geq A_{k_j}$ for $j \geq k$, the above inequality implies that for $j \geq \max\{m, k\}$ we have $\mathbf{P}(A'_j > a - \varepsilon) > 1 - p$. It remains only to show that A'_j doesn't get too large, but this follows from the fact that $A'_j \leq B'_j$ and $B'_j \rightarrow a$ in probability. Specifically, we are guaranteed some $N \geq \max\{m, k\}$ such that $\mathbf{P}(B'_j < a + \varepsilon) > 1 - p$ for $j \geq N$. Since $B'_j < a + \varepsilon$ implies $A'_j < a + \varepsilon$, we have that $\mathbf{P}(|A'_j - a| < \varepsilon) > 1 - p$ for $j \geq N$, as desired. ■

Proposition 48 *The function*

$$\widetilde{\text{MIC}}_{e,B}(\cdot) = \max_{k\ell \leq B(n)} \{\widehat{M}\}^c(\cdot)_{k,\ell}$$

is a consistent estimator of MIC_* provided $\omega(1) < B(n) \leq O(n^{1-\varepsilon})$ for some $\varepsilon > 0$, where $\{\widehat{M}\}^c(\cdot)$ is the output of the `EQUICHARCLUMP` algorithm.

Proof Let (X, Y) be a pair of jointly distributed random variables, and let D_n be a sample of size n from the distribution of (X, Y) . Let $\{(k_i, \ell_i)\}_{i=1}^\infty \subset \mathbb{Z}^+ \times \mathbb{Z}^+$ be a sequence of coordinates with the property that for every number B there exists an index $q(B)$ such that $\{(k_i, \ell_i) : i \leq q(B)\} = \{(k, \ell) : k\ell \leq B\}$.

We define $B_{ij} = [\widehat{M}(D_j)_{k_i, \ell_i}]$, i.e., B_{ij} is the k_i, ℓ_i -th entry of the sample characteristic matrix evaluated on a sample of size j . We analogously define $A_{ij} = \{\widehat{M}\}^c(D_j)_{k_i, \ell_i}$, and we define $a_i = \{M\}^c(X, Y)_{k_i, \ell_i}$. We observe that by Proposition 46, $\sup a_i = \sup\{M\}^c(X, Y) = \text{MIC}_*$.

It is straightforward to see that $A_{ij} \leq B_{ij}$. Additionally, Lemma 45 shows that $A_{ij} \rightarrow a_i$ in probability, and Corollary 26, which states that MIC_e is a consistent estimator of MIC_* , shows that $B'_j = \max_{i \leq j} B_{ij} \rightarrow \text{MIC}_*(X, Y)$. In the notation of the lemma, it therefore follows that $A'_j = \max_{i \leq j} A_{ij}$ converges in probability to $\text{MIC}_*(X, Y)$ as well. But this means that the sub-sequence

$$A'_{q(B(n))} = \max_{i \leq q(B(n))} \{\widehat{M}\}^c(D_{q(B(n))})_{k_i, \ell_i} = \max_{k\ell \leq B(n)} \{\widehat{M}\}^c(D_{q(B(n))})_{k,\ell}$$

converges in probability to $\text{MIC}_*(X, Y)$, which implies the result since the sequence A'_j is monotone. ■

H.2.2. CONSISTENCY FOR TOTAL INFORMATION COEFFICIENT

Similarly to the consistency argument for MIC_* , we begin by exhibiting the relevant property of the population clumped equicharacteristic matrix.

Proposition 49 *Let (X, Y) be a pair of jointly distributed random variables. If X and Y are statistically independent, then $\{M\}^c(X, Y) \equiv 0$. If not, then there exists some $a > 0$ and some integer $\ell_0 \geq 2$ such that*

$$\{M\}^c(X, Y)_{k,\ell} \geq \frac{a}{\log \min\{k, \ell\}}$$

either for all $k \geq \ell \geq \ell_0$, or for all $\ell \geq k \geq \ell_0$.

Proof (Sketch) Let $\{M\}^c = \{\widehat{M}\}^c(X, Y)$. Under independence, every entry of $\{M\}^c$ is zero since $I((X, Y)|_G) = 0$ for any grid G . For the case of dependence, the argument is identical to that given in the proof of Proposition 32. Specifically, it can be shown that there exists some index ℓ_0 , taken without loss of generality to be a column index, and some $r > 0$ such that all but finitely many of the entries in the ℓ_0 -column are at least r . It can then be shown that for large k , the entries $(k, \ell_0), (k, \ell_0 + 1), \dots, (k, k)$ have non-decreasing values of $I^{(a)}$. This establishes the claim for $a = r \log \ell_0$. ■

We now show that the above result, together with the uniform convergence of $\{\widehat{M}\}^c(D_n)$ to $\{M\}^c(X, Y)$, implies the consistency we seek.

Proposition 50 *The function*

$$\widetilde{\text{TIC}}_{e,B}(\cdot) = \sum_{k\ell \leq B(n)} \{\widehat{M}\}^c(\cdot)_{k,\ell}$$

yields a consistent right-tailed test of independence provided $\omega(1) < B(n) \leq O(n^{1-\varepsilon})$ for some $\varepsilon > 0$, where $\{\widehat{M}\}^c(\cdot)$ is the output of the `EQUICHARCLUMP` algorithm.

Proof Let (X, Y) a pair of jointly distributed random variables, and let D_n be a sample of size n from the distribution of (X, Y) . It suffices to show consistency for any deterministic monotonic function of the statistic in question. We therefore choose to analyze $\widetilde{\text{TIC}}_{e,B}(D_n) \log(B(n))/B(n)$.

For the null hypothesis in which X and Y are independent, we observe that since $\{\widehat{M}\}^c(D_n) \leq [\widehat{M}(D_n)]$ element-wise, $0 \leq \widetilde{\text{TIC}}_{e,B}(D_n) \leq \text{TIC}_{e,B}(D_n)$ as well. Moreover, the argument given in Appendix K, which shows that $\text{TIC}_{e,B}(D_n)/B(n)$ converges to 0 in probability under the null hypothesis, can be adapted to show that $\widetilde{\text{TIC}}_{e,B}(D_n) \log(B(n))/B(n) \rightarrow 0$ as well. Thus, $\widetilde{\text{TIC}}_{e,B}(D_n) \log(B(n))/B(n)$ converges to zero in probability, as required.

For the case that X and Y are dependent, the proof is analogous to the argument given in Appendix K for TIC_e . The only difference is that Lemma 44, which guarantees the uniform convergence of $\{\widehat{M}\}^c(D_n)$ to $\{M\}^c(X, Y)$, applies only to the k, ℓ -th entries for which $k, \ell \leq \sqrt{B(n)}$, rather than the entries over which we are summing, which are those for which $k\ell \leq B(n)$. However, since we require only a lower bound on $\widetilde{\text{TIC}}_{e,B}(D_n)$, we may neglect these entries because

$$\widetilde{\text{TIC}}_{e,B}(D_n) = \sum_{k\ell \leq B(n)} \{\widehat{M}\}^c(D_n)_{k,\ell} \geq \sum_{k,\ell \leq \sqrt{B(n)}} \{\widehat{M}\}^c(D_n)_{k,\ell}.$$

It can then be shown, following the argument from Appendix K, that there exists some $a > 0$ depending only on B such that, with probability $1 - o(1)$,

$$\frac{\log B(n)}{B(n)} \left(\sum_{k,\ell \leq \sqrt{B(n)}} \{\widehat{M}\}^c(X, Y)_{k,\ell} - \widetilde{\text{TIC}}_{e,B}(D_n) \right) \leq O\left(\frac{\#_n \log B(n)}{B(n)n^a}\right) = O\left(\frac{\log B(n)}{n^a}\right)$$

where $\#_n = B(n)$ represents the number of pairs (k, ℓ) such that $k, \ell \leq \sqrt{B(n)}$. To obtain the result, we note that this means that

$$\frac{\log B(n)}{B(n)} \widehat{\text{TTC}}_{\epsilon, B}(D_n) \geq \frac{\log B(n)}{B(n)} \sum_{k, \ell \leq \sqrt{B(n)}} \{\widehat{M}^{\epsilon}(X, Y)_{k, \ell} - O\left(\frac{\log B(n)}{n^{\alpha}}\right)\}$$

and then invoke Proposition 49, which implies that for large n

$$\sum_{k, \ell \leq \sqrt{B(n)}} \{M\}^{\epsilon}(X, Y) \geq \Omega\left(\frac{B(n)}{\log B(n)}\right).$$

■

H.3 Empirical Characterization of the Performance of EQUICHARCLUMP

The EQUICHARCLUMP algorithm has a parameter c that controls the fineness of the equipartition whose sub-partitions are searched over by the algorithm. To gain an empirical understanding of the effect of c on performance, we computed MIC_{ϵ} on the set of relationships described in Section 4.4 using EQUICHARCLUMP with different values of c . For each relationship, we compared the average MIC_{ϵ} across all 500 independent samples from that relationship with different values of c . We performed this analysis at sample sizes of $n = 250$ (Figure H1), $n = 500$ (Figure H2), and 5,000 (Figure H3).

We summarize our findings as follows.

- At low ($n = 250$) and medium ($n = 500$) sample sizes, using $c = 1$ introduces a downward bias for more complex relationships when $B(n) = n^{0.6}$ is used but not when $B(n) = n^{0.8}$ is used. This makes sense since the low sample size and low setting of $B(n)$ mean that the algorithm is searching over grids with relatively few cells, and so setting $c = 1$ hinders its ability to find good grids in this limited search space. This bias is almost entirely alleviated by setting $c \geq 2$.
- At high sample size ($n = 5,000$), this effect is still observable but much reduced. This makes sense since when n is large, $B(n)$ is large as well, and so the number of cells allowed in the grids being searched over is already large regardless of the exponent α used in $B(n) = n^{\alpha}$. Thus, there is less need for the robustness provided by searching for an optimal grid.

Appendix I. Equitability and Power Analyses from Reshef et al. (2015a)

Figure I1 contains a representative equitability analysis from Reshef et al. (2015a). Figure I2 contains power curves from Reshef et al. (2015a) for a large set of leading methods.

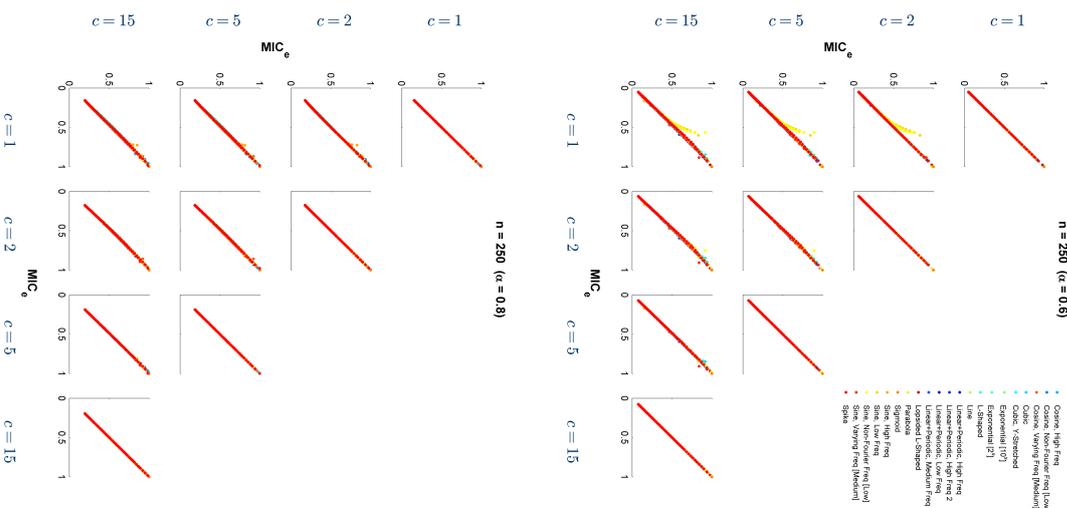


Figure H1: The effect of the parameter c on the performance of EQUICHARCLUMP, at $n = 250$. See Section H.3 for details.

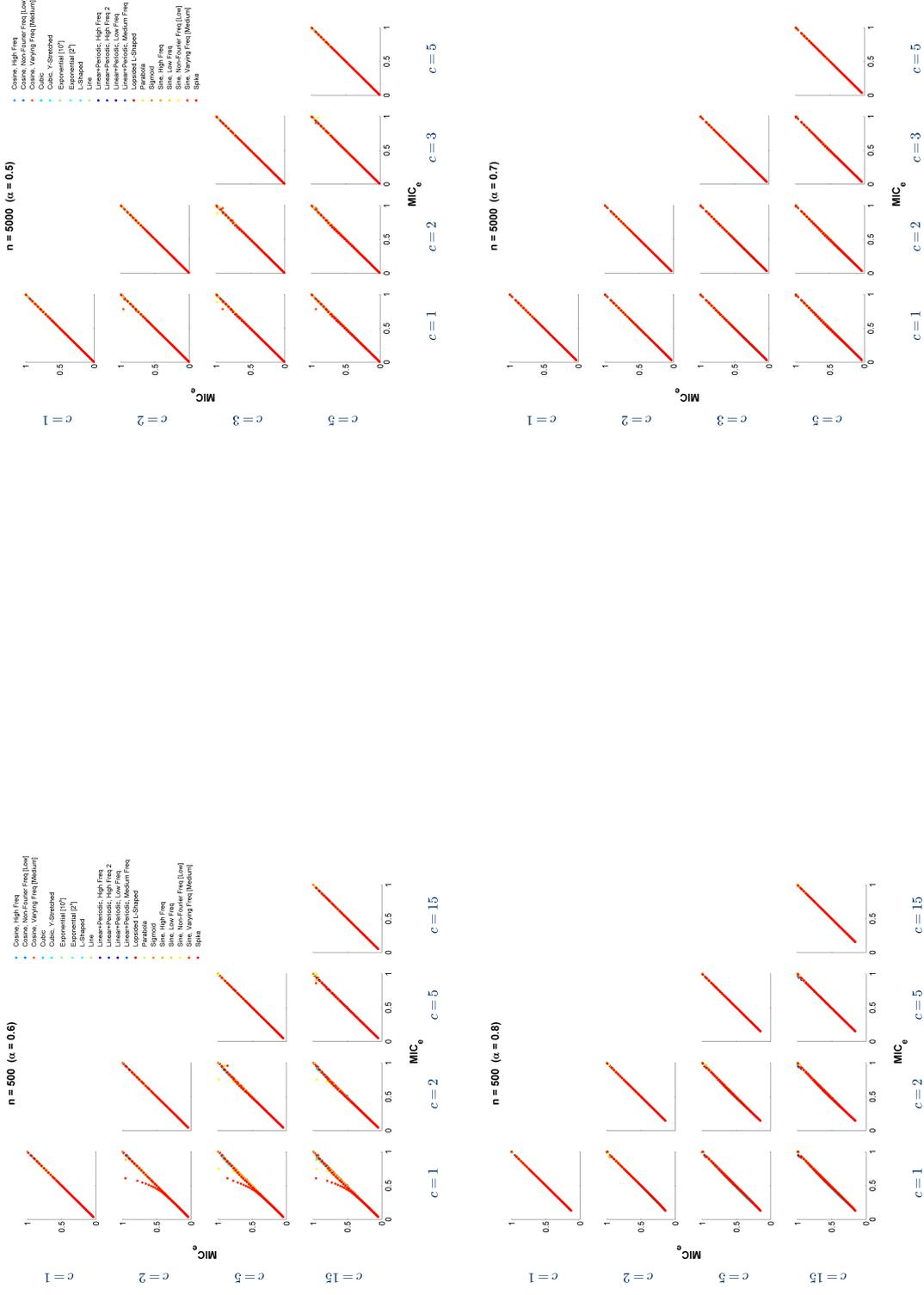


Figure H2: The effect of the parameter c on the performance of EQUICHARCLUMP, at $n = 500$. See Section H.3 for details.

Figure H3: The effect of the parameter c on the performance of EQUICHARCLUMP, at $n = 5,000$. See Section H.3 for details.

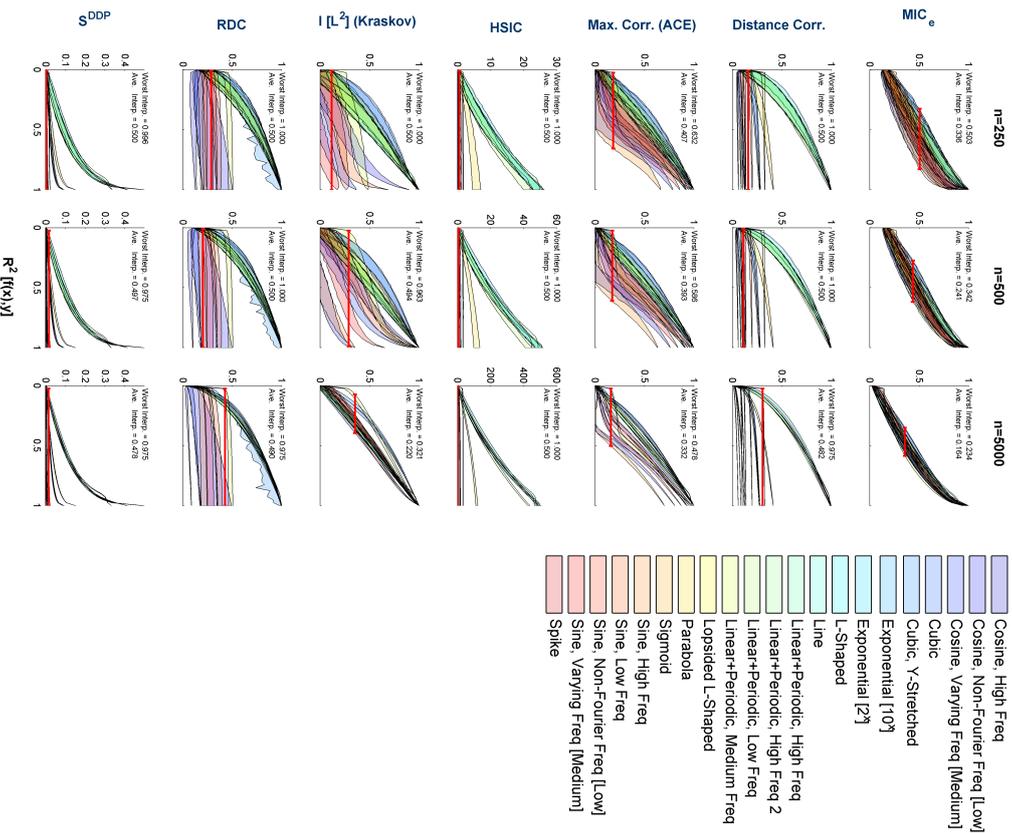


Figure 11: (Reproduced from Reshef et al., 2015a.) The equitability of measures of dependence on a set of noisy functional relationships, reproduced from Reshef et al. (2015a). *Narrower is more equitable.* The plots were constructed as in Figure 3. Mutual information, estimated using the Kraskov estimator, is represented using the squared Linf-foot correlation.

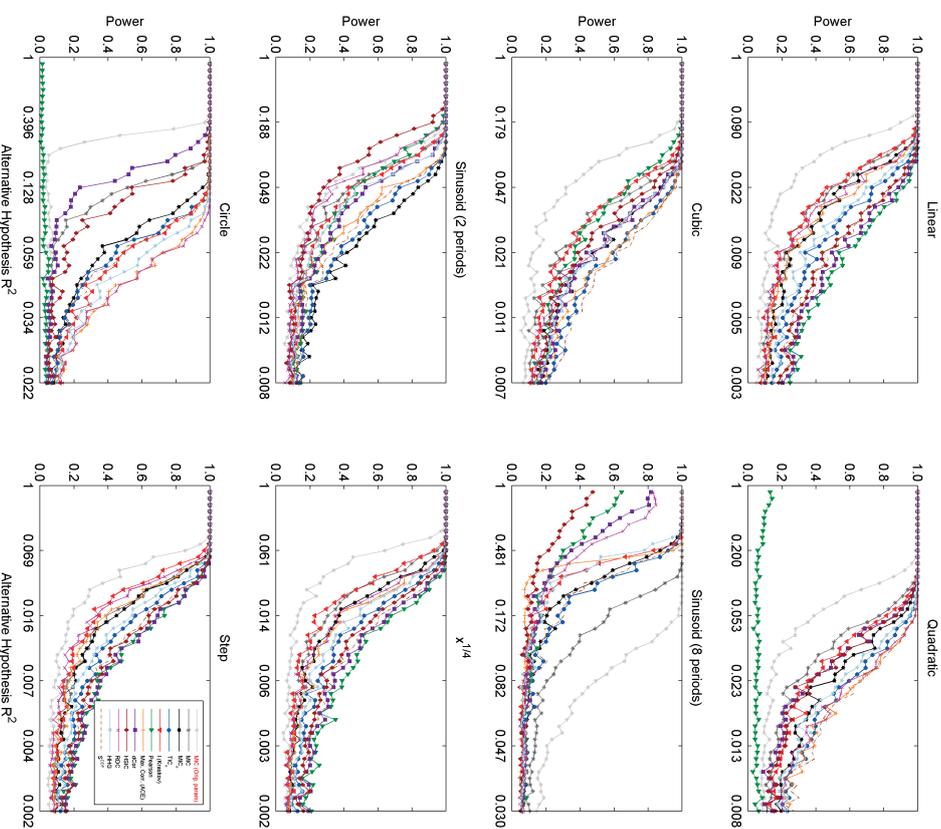


Figure 12: (Reproduced from Reshef et al., 2015a.) Power of independence testing using several leading measures of dependence, on the relationships chosen by Simon and Tibshirani (2012), at 50 noise levels with linearly increasing magnitude for each relationship and $n = 500$. To enable comparison of power regimes across relationships, the x-axis of each plot lists R^2 rather than noise magnitude.

Appendix J. Equitability Analysis of Randomly Chosen Functions with Additional Noise Model

Figure J1 contains a version of the main text Figure 4, but where noise has been added only to the dependent variable in each functional relationship, rather than to both the independent and dependent variables.

Appendix K. Consistency of Independence Testing Based on TIC_e

Here we prove Propositions 32 and 33 and then use those propositions to prove Theorem 34, which shows that TIC_e can be used for independence testing.

K.1 Proof of Proposition 32

Proposition *Let (X, Y) be a pair of jointly distributed random variables. If X and Y are statistically independent, then $M(X, Y) \equiv [M](X, Y) \equiv 0$. If not, then there exists some $a > 0$ and some integer $\ell_0 \geq 2$ such that*

$$M(X, Y)_{k,\ell}, [M](X, Y)_{k,\ell} \geq \frac{a}{\log \min\{k, \ell\}}$$

either for all $k \geq \ell \geq \ell_0$, or for all $\ell \geq k \geq \ell_0$.

Proof We give the proof only for $[M] = [M](X, Y)$, with the understanding that all parts of the argument are either identical or similar for $M(X, Y)$. When X and Y are independent, then for any grid $G, (X, Y)|_G$ exhibits independence as well. Therefore $I((X, Y)|_G) = 0$ for all grids G , and so every entry of $[M]$, being a supremum over such quantities, is 0.

For the case that X and Y are dependent, our strategy is to first find, without loss of generality, a column of $[M]$ almost all of whose values are bounded away from zero, and then argue that this suffices.

The dependence of X and Y implies that $\text{MIC}_*(X, Y) > 0$. By Corollary 22, which states that $\sup \partial[M] = \text{MIC}_*(X, Y)$, we therefore know that there is at least one non-zero element of the boundary of $[M]$, as defined in Definition 14. Without loss of generality, suppose that this element is $[M]_{r,\ell_0} = \lim_{k \rightarrow \infty} [M]_{k,\ell_0}$. The fact that this limit is strictly positive implies that there exists some $k_0 \geq \ell_0$ and some $r > 0$ such that $[M]_{k,\ell_0} \geq r$ for all $k \geq k_0$. That is, all but finitely many of the entries in the ℓ_0 -th column of $[M]$ are at least r .

We now show that the existence of such a column suffices to prove the claim. Fix some $k > k_0$ and note that this implies that $k > \ell_0$. We argue that for all ℓ in $\{\ell_0, \dots, k\}$, the desired condition holds. Since $k > \ell_0$, the term $I^{[\bullet]}((X, Y), k, \ell_0)$ in the definition of $[M]_{k,\ell_0}$ is a maximization over grids that have an equipartition of size k on one axis and an optimal partition of size ℓ_0 on the other. Since we allow empty rows/columns in the maximization, substituting any ℓ satisfying $\ell_0 \leq \ell \leq k$ therefore does not constrain the maximization in any way and so it cannot decrease $I^{[\bullet]}$. In other words, for ℓ satisfying $\ell_0 \leq \ell \leq k$, we have

$$I^{[\bullet]}((X, Y), k, \ell) \geq I^{[\bullet]}((X, Y), k, \ell_0).$$

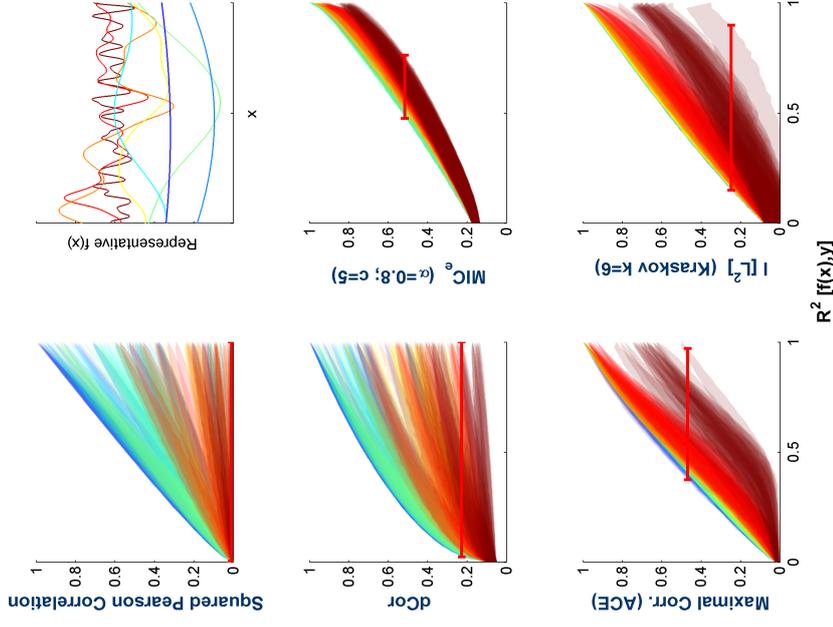


Figure J1: Equitability of methods examined on functions randomly drawn from a Gaussian process distribution, using a different noise model. This figure is identical to Figure 4, but with noise added only to the dependent variable in each relationship. Each method is assessed as in Figure 4, with a red interval indicating the widest range of R^2 values corresponding to any one value of the statistic; the narrower the red interval, the higher the equitability. Sample relationships for each Gaussian process bandwidth are shown in the top right with matching colors.

Since $k \geq \ell$, ℓ_0 , the normalizations in the definition of $[M]_{k,\ell}$ and $[M]_{k,\ell_0}$ are $\log \ell$ and $\log \ell_0$ respectively. Therefore, we have that

$$[M]_{k,\ell} \geq [M]_{k,\ell_0} \frac{\log \ell_0}{\log \ell} \geq \frac{r \log \ell_0}{\log \ell}$$

where the last inequality is because $k > k_0$. Setting $a = r \log \ell_0$ then gives the result. \blacksquare

K.2 Proof of Proposition 33

Proposition *Let (X, Y) be a pair of jointly distributed random variables. If X and Y are statistically independent, then $S_B(M(X, Y)) = S_B([M](X, Y)) = 0$ for all $B > 0$. If not, then $S_B(M(X, Y))$ and $S_B([M](X, Y))$ are both $\Omega(B \log \log B)$.*

Proof We give the argument for $M = M(X, Y)$ only, but the argument holds as stated for $[M](X, Y)$ as well.

The result follows from the guarantee given by the Proposition 32 above. In the case of independence, the proposition tells us that $M \equiv 0$, which immediately gives that $S_B(M) = 0$ for all $B > 0$. For the case of dependence, the proposition implies that there is some $a > 0$ and some integer $\ell_0 \geq 2$ such that, without loss of generality, $M_{k,\ell} \geq a/\log \ell$ for all $k \geq \ell \geq \ell_0$. We convert this into a lower bound on $S_B(M)$.

The key is to write the sum one column at a time, counting how many entries in each column both satisfy $k \geq \ell \geq \ell_0$ and $k\ell \leq B$. For any ℓ satisfying $\ell_0 \leq \ell \leq \sqrt{B}$, the entries $(\ell, \ell), \dots, (B/\ell, \ell)$ meet this criterion, and there are $B/\ell_0 - (\ell_0 - 1)$ of them. Moreover, since the guarantee of Proposition 32 tells us that all of these entries are at least $a/\log \ell$, we can lower-bound $S_B(M)$ as follows.

$$\begin{aligned} S_B(A) &\geq \sum_{\ell=\ell_0}^{\sqrt{B}} \frac{a}{\log \ell} \left(\frac{B}{\ell} - (\ell - 1) \right) \\ &= aB \sum_{\ell=\ell_0}^{\sqrt{B}} \frac{1}{\ell \log \ell} - a \sum_{\ell=\ell_0}^{\sqrt{B}} \frac{\ell - 1}{\log \ell} \\ &= a \left(B \sum_{\ell=\ell_0}^{\sqrt{B}} \frac{1}{\ell \log \ell} - O(B) \right) \\ &= \Omega(B \log \log B) \end{aligned}$$

where the second-to-last equality is because $(\ell - 1)/\log \ell \leq \ell$, and the last equality is because $\sum_{\ell=\ell_0}^{\sqrt{B}} 1/(\ell \log \ell)$ grows like $\log \log n$. \blacksquare

K.3 Proof of Theorem 34

Theorem *The statistics TTC_B and $\text{TTC}_{\alpha,B}$ yield consistent right-tailed tests of independence, provided $\omega(1) < B(n) \leq O(n^{1-\varepsilon})$ for some $\varepsilon > 0$.*

Proof We give the proof for TTC only; however, the argument holds as stated for TTC_ε as well.

Let (X, Y) be jointly distributed random variables, and let D_n be a sample of size n from the distribution of (X, Y) . Let $M = M(X, Y)$ be the characteristic matrix of (X, Y) and let $\widehat{M}(D_n)$ be the sample characteristic matrix. It suffices to establish the result for a deterministic monotonic function of $\text{TTC}_B(D_n)$. We therefore show convergence of $\text{TTC}_B(D_n)/B(n)$ to zero under the null hypothesis of independence and to ∞ under any alternative. Our general strategy for doing so is to translate known bounds on our error at estimating entries of M into bounds on the difference between $\text{TTC}_B(D_n)/B(n) = S_{B(n)}(\widehat{M}(D_n))/B(n)$ and $S_B(M)/B(n)$. We then obtain the result by invoking Proposition 33, which implies that $S_B(M)/B(n)$ is zero under the null hypothesis but grows without bound under the alternative.

We know from Lemma 38 (Lemma 42 for the equicharacteristic matrix) that there exists some $a > 0$ depending only on B such that

$$\left| \widehat{M}(D_n)_{k,\ell} - M_{k,\ell} \right| \leq O\left(\frac{1}{n^a}\right)$$

for all $k\ell \leq B(n)$ with probability $1 - o(1)$. This means that with probability $1 - o(1)$ we have

$$\frac{1}{B(n)} \left| \text{TTC}_B(D_n) - S_{B(n)}(M) \right| \leq O\left(\frac{\#_n}{B(n)n^a}\right)$$

where $\#_n$ is the number of pairs (k, ℓ) such that $k\ell \leq B(n)$. It can be shown by taking the integral of B/x with respect to x that $\#_n = O(B(n) \log B(n))$. Therefore, the error in the above bound is at most $O(\log B(n)/n^a) = O(1/\text{poly}(n))$ for our choice of $B(n)$.

We now use Proposition 33 to show that this bound gives the desired result. Under the null hypothesis of independence, the proposition says that $S_{B(n)}(M) = 0$ always, and so since B is a growing function the bound implies that $\text{TTC}_B(D_n)/B(n) \rightarrow 0$ in probability. Under the alternative hypothesis in which (X, Y) exhibit a dependence, the proposition implies that $S_{B(n)}(M)/B(n) > \omega(1)$. Since B is a growing function of n , this means that for any $\tau > 0$, the probability that $S_{B(n)}(M)/B(n) > \tau$ goes to 1 as n grows. In other words, $\text{TTC}_B(D_n)/B(n) \rightarrow \infty$ in probability. \blacksquare

Appendix L. Information-Theoretic Lemmas

Lemma 51 *Let Π and Ψ be random variables distributed over a discrete set of states Γ , and let (π_i) and (ψ_i) be their respective distributions. Let $P = f(\Pi)$ and $Q = f(\Psi)$ for some function f whose image is of size B . Define*

$$\varepsilon_i = \frac{\psi_i - \pi_i}{\pi_i}.$$

Then for every $0 < a < 1$ there exists some $A > 0$ such that

$$|H(Q) - H(P)| \leq (\log B) A \sum_i |\varepsilon_i|$$

when $|\varepsilon_i| \leq 1 - a$ for all i .

Proof We prove the claim with entropy measured in nats. A rescaling then gives the general result.

Let (p_i) and (q_i) be the distributions of P and Q respectively, and define

$$e_i = \frac{q_i - p_i}{p_i}$$

analogously to ε_i . Before proceeding, we observe that

$$e_i = \sum_{j \in f^{-1}(i)} \frac{\pi_j \varepsilon_j}{p_i}.$$

We now proceed with the argument. We have from [Roulston \(1999\)](#) that

$$|H(Q) - H(P)| \leq \left| \sum_i \left(e_i p_i (1 + \ln p_i) + \frac{1}{2} e_i^2 p_i + O(e_i^3) \right) \right| \quad (3)$$

$$\leq \sum_i |e_i p_i| + \sum_i |e_i p_i \ln p_i| + \frac{1}{2} \sum_i |e_i^2 p_i| + \left| \sum_i O(e_i^3) \right| \quad (4)$$

$$= \sum_i |e_i p_i \ln p_i| + \frac{1}{2} \sum_i |e_i^2 p_i| + \left| \sum_i O(e_i^3) \right| \quad (5)$$

where the final equality is because $\sum_i e_i p_i = \sum_i q_i - \sum_i p_i = 0$. We proceed by bounding each of the terms in Equation 5 separately.

To bound the first term, we write

$$\left| \sum_i e_i p_i \ln p_i \right| \leq - \sum_i |e_i| p_i \ln p_i.$$

We then note that $-\sum_i p_i \ln p_i \leq \ln B$, and since each of the summands has the same sign this means that $-p_i \ln p_i \leq \ln B$. We also observe that

$$|e_i| \leq \left| \sum_{j \in f^{-1}(i)} \frac{\pi_j \varepsilon_j}{p_i} \right| \leq \sum_j \frac{\pi_j}{p_i} |\varepsilon_j| \leq \sum_j |\varepsilon_j|$$

since $\pi_j/p_i \leq 1$. Together, these two facts give

$$\begin{aligned} - \sum_i |e_i| p_i \ln p_i &\leq (\ln B) \sum_i |e_i| \\ &\leq (\ln B) \sum_i |\varepsilon_i| \end{aligned}$$

The second inequality is because each e_i is a weighted average of a set of ε_i and each ε_i enters into the expression of exactly one e_i .

To bound the second term, we use the fact that $p_i \leq 1$ for all i , and so

$$\sum_i e_i^2 p_i \leq \sum_i e_i^2.$$

We then write

$$\begin{aligned} \sum_i e_i^2 &= \sum_i \left(\sum_{j \in f^{-1}(i)} \frac{\pi_j \varepsilon_j}{p_i} \right)^2 \\ &\leq \sum_i \sum_{j \in f^{-1}(i)} \frac{\pi_j \varepsilon_j^2}{p_i} \\ &\leq \sum_j \varepsilon_j^2 \\ &= \sum_j O(|\varepsilon_j|) \end{aligned}$$

where the second line is a consequence of the convexity of $f(x) = x^2$ and the third line is because the sets $f^{-1}(i)$ partition Γ .

To bound the third term, we write

$$\left| \sum_i O(e_i^3) \right| \leq \sum_i O(|e_i|^3)$$

and then proceed as we did with the second term, using the fact that $f(x) = x^3$ is convex for $x \geq 0$. This gives

$$\sum_i O(|e_i|^3) \leq \sum_i O(|\varepsilon_i|^3) = \sum_i O(|\varepsilon_i|)$$

completing the proof. \blacksquare

Lemma 52 Let $\{w_i\} \subset [0, 1]$ be a set of size n with $\sum_i w_i \leq 1$, and let $\{u_i\}$ be a set of n non-negative numbers satisfying $\sum_i u_i = a$ and $u_i \leq w_i$. Then

$$\sum_{i=1}^n u_i H_b \left(\frac{u_i}{w_i} \right) \leq H_b(a)$$

where H_b is the binary entropy function.

Proof Consider the random variable X taking values in $\{0, \dots, n\}$ that equals zero with probability $1 - \sum_j w_j$ and equals i with probability w_i for $0 < i \leq n$. Define the random variable Y taking values in $\{0, 1\}$ by

$$\mathbf{P}(Y = 0 | X = i) = \begin{cases} 0 & i = 0 \\ u_i/w_i & 0 < i \leq n \end{cases}.$$

The function we wish to bound equals $H(Y|X) \leq H(Y)$. We therefore observe that

$$\sum_{i=1}^n u_i H_b \left(\frac{u_i}{w_i} \right) \leq H(Y).$$

The result follows from the observation that

$$\mathbf{P}(Y=0) = \sum_i \mathbf{P}(X=i) \frac{u_i}{w_i} = \sum_i u_i \leq a.$$

■

Lemma 53 *Let X be a random variable distributed over k states, with $\mathbf{P}(X=x) = p_x$. Let $\alpha_x \geq 0$ be such that $\sum \alpha_x = \delta$, and define the random variable X' by $\mathbf{P}(X'=x) = (p_x + \alpha_x)/(1 + \delta)$. We have*

$$|H(X') - H(X)| \leq H_b(\delta) + \delta \log k$$

where H_b is the binary entropy function.

Proof Define a new random variable Z by

$$\mathbf{P}(Z=0|X'=x) = \frac{p_x}{p_x + \alpha_x}, \quad \mathbf{P}(Z=1|X'=x) = \frac{\alpha_x}{p_x + \alpha_x}.$$

We will use the fact that $H(X'|Z=0) = H(X)$ to obtain the required bound.

To upper bound $H(X') - H(X)$, we write

$$\begin{aligned} H(X') - H(X) &\leq H(X', Z) - H(X) \\ &= H(Z) + \mathbf{P}(Z=0)H(X'|Z=0) + \mathbf{P}(Z=1)H(X'|Z=1) - H(X) \\ &\leq H_b(\delta) + (1-\delta)H(X) + \delta H(X'|Z=1) - H(X) \\ &= H_b(\delta) - \delta H(X) + \delta \log k \\ &\leq H_b(\delta) + \delta \log k \end{aligned}$$

where in the fourth line we have used that $H(X'|Z=1) \leq \log k$.

To upper bound $H(X) - H(X')$, we write

$$\begin{aligned} H(X') + H(Z) &\geq H(X', Z) \\ &\geq \mathbf{P}(Z=0)H(X'|Z=0) \\ &= (1-\delta)H(X) \end{aligned}$$

which yields

$$H(X') \geq (1-\delta)H(X) - H_b(\delta)$$

since $H(Z) = H_b(\delta)$. Thus, we have

$$H(X) - H(X') \leq \delta H(X) + H_b(\delta) \leq \delta \log k + H_b(\delta). \quad \blacksquare$$

Lemma 54 *Let X be a random variable distributed over k states, with $\mathbf{P}(X=x) = p_x$. Let $\alpha_x \leq 0$ be such that $\sum |\alpha_x| = \delta$, and define the random variable X' by $\mathbf{P}(X'=x) = (p_x + \alpha_x)/(1 - \delta)$. We have*

$$|H(X') - H(X)| \leq H_b\left(\frac{\delta}{1-\delta}\right) + \frac{\delta}{1-\delta} \log k$$

where H_b is the binary entropy function. In particular, when $\delta \leq 1/3$ we have

$$|H(X') - H(X)| \leq H_b(2\delta) + 2\delta \log k.$$

Proof We observe that we can get from X' to X by adding $\delta/(1-\delta)$ probability mass and rescaling. The previous lemma then gives the result. ■

Lemma 55 *Let X be a random variable distributed over k states, with $\mathbf{P}(X=x) = p_x$. Let α_x be such that $\sum |\alpha_x| = \delta$, and define the random variable X' by $\mathbf{P}(X'=x) = (p_x + \alpha_x)/(1 - \sum \alpha_x)$. That is, X' is the result of changing the probability of state x by α_x and then re-normalizing to obtain a valid distribution. If $\delta \leq 1/4$, we have*

$$|H(X') - H(X)| \leq 2H_b(2\delta) + 3\delta \log k$$

where H_b is the binary entropy function.

Proof Let δ_+ be the total magnitude of all the positive α_x , and let δ_- be the total magnitude of all the negative α_x . We first add all the mass we're going to add, and apply the first of the previous two lemmas. Then we remove all the mass we are going to remove, and apply the second of the two previous lemmas. This yields a bound of

$$\begin{aligned} &H_b(\delta_+) + \delta_+ \log k + H_b\left(\frac{2\delta_-}{1+\delta_+}\right) + 2\frac{\delta_-}{1+\delta_+} \log k \\ &\leq H_b(\delta_+) + \delta_+ \log k + H_b(2\delta_-) + 2\delta_- \log k \\ &\leq H_b(2\delta) + \delta \log k + H_b(2\delta) + 2\delta \log k \\ &\leq 2H_b(2\delta) + 3\delta \log k \end{aligned}$$

where the first inequality is because $1 + \delta_+ \leq 1 + \delta < 2$ and $2\delta_- \leq 2\delta \leq 1/2$, and the second inequality is because $\delta_+ \leq \delta < 2\delta \leq 1/2$. ■

References

- Leo Breiman and Jerome H. Friedman. Estimating optimal transformations for multiple regression and correlation. *Journal of the American Statistical Association*, 80(391):580–598, 1985.
- William S. Cleveland and Susan J. Devlin. Locally weighted regression: an approach to regression analysis by local fitting. *Journal of the American Statistical Association*, 83(403):596–610, 1988.
- Thomas Cover and Joy Thomas. *Elements of Information Theory*. New York: John Wiley & Sons, Inc, 2006.
- Imre Csiszár. Axiomatic characterizations of information measures. *Entropy*, 10(3):261–273, 2008.
- Imre Csiszár and Paul C. Shields. Information theory and statistics: A tutorial. *Communications and Information Theory*, 1(4):417–528, 2004.
- A. Adam Ding and Yi Li. Copula correlation: An equitable dependence measure and extension of pearson’s correlation. *arXiv preprint arXiv:1312.7214*, 2013.
- Valur Emilsson, Gudmar Thorleifsson, Bin Zhang, Amy S Leonardson, Florian Zink, Jun Zhu, Sonia Carlson, Agnar Helgason, G Bragi Walters, Steinunn Gunnarsdottir, et al. Genetics of gene expression and its effect on disease. *Nature*, 452(7186):423–428, 2008.
- Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *Algorithmic learning theory*, pages 63–77. Springer, 2005.
- Arthur Gretton, Kenji Fukumizu, Choon H Teo, Le Song, Bernhard Schölkopf, and Alex J Smola. A kernel statistical test of independence. In *Advances in neural information processing systems*, pages 585–592, 2007.
- Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.
- Ruth Heller, Yair Heller, and Malka Gorfine. A consistent multivariate test of association based on ranks of distances. *Biometrika*, 100(2):503–510, 2013.
- Ruth Heller, Yair Heller, Shachar Kaufman, Barak Brill, and Malka Gorfine. Consistent distribution-free k -sample and independence tests for univariate random variables. *Journal of Machine Learning Research*, 17(29):1–54, 2016.
- Wassily Hoeffding. A non-parametric test of independence. *The Annals of Mathematical Statistics*, pages 546–557, 1948.
- Bo Jiang, Chao Ye, and Jun S Liu. Nonparametric k -sample tests via dynamic slicing. *Journal of the American Statistical Association*, 110(510):642–653, 2015.
- Justin B. Kinney and Gurinder S. Atwal. Equitability, mutual information, and the maximal information coefficient. *Proceedings of the National Academy of Sciences*, 2014. doi: 10.1073/pnas.1309933111.
- Alexander Kraskov, Harald Stogbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E*, 69, 2004.
- Edward H. Linfoot. An informational measure of correlation. *Information and Control*, 1(1):85–89, 1957.
- David Lopez-Paz, Philipp Hennig, and Bernhard Schölkopf. The randomized dependence coefficient. In *Advances in Neural Information Processing Systems*, pages 1–9, 2013.
- Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomized algorithms and probabilistic analysis*. Cambridge University Press, 2005.
- Ben Murrell, Daniel Murrell, and Hugh Murrell. R2-equitability is satisfiable. *Proceedings of the National Academy of Sciences*, 2014. doi: 10.1073/pnas.1403623111. URL <http://www.pnas.org/content/early/2014/04/29/1403623111.short>.
- Liam Paninski. Estimation of entropy and mutual information. *Neural computation*, 15(6):1191–1253, 2003.
- Alfred Rényi. On measures of dependence. *Acta mathematica hungarica*, 10(3):441–451, 1959.
- David N. Reshef, Yakir A. Reshef, Hilary K. Finucane, Sharon R. Grossman, Gilean McVean, Peter J. Turnbaugh, Eric S. Lander, Michael Mitzenmacher, and Pardis C. Sabeti. Detecting novel associations in large data sets. *Science*, 334(6062):1518–1524, 2011.
- David N. Reshef, Yakir A. Reshef, Michael Mitzenmacher, and Pardis C. Sabeti. Cleaning up the record on the maximal information coefficient and equitability. *Proceedings of the National Academy of Sciences*, 2014. doi: 10.1073/pnas.1408920111. URL <http://www.pnas.org/content/early/2014/08/07/1408920111.short>.
- David N. Reshef, Yakir A. Reshef, Pardis C. Sabeti, and Michael Mitzenmacher. An empirical study of leading measures of dependence. *arXiv preprint arXiv:1505.02214*, 2015a.
- Yakir A Reshef, David N Reshef, Pardis C Sabeti, and Michael Mitzenmacher. Equitability, interval estimation, and statistical power. *arXiv preprint arXiv:1505.02212*, 2015b.
- Mark S. Roulston. Estimating the errors on measured entropy and mutual information. *Physica D: Nonlinear Phenomena*, 125(3):285–294, 1999.
- Noah Simon and Robert Tibshirani. Comment on “Detecting novel associations in large data sets”. *Unpublished (available at http://www-stat.stanford.edu/~tibs/reshef/comment.pdf on 11 Nov. 2012)*, 2012.
- Terry Speed. A correlation for the 21st century. *Science*, 334(6062):1502–1503, 2011.
- Josef Stoer and Roland Bulirsch. *Introduction to Numerical Analysis*. Springer-Verlag, 1980.

MEASURING DEPENDENCE POWERFULLY AND EQUITABLY

- Charles J. Stone. Consistent nonparametric regression. *The annals of statistics*, pages 595–620, 1977.
- John D. Storey and Robert Tibshirani. Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences*, 100(16):9440–9445, 2003.
- Gábor J. Székely and Maria L. Rizzo. Brownian distance covariance. *The Annals of Applied Statistics*, 3(4):1236–1265, 2009.
- Gábor J. Székely, Maria L. Rizzo, Nail K. Bakirov, et al. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769–2794, 2007.

Neyman-Pearson Classification under High-Dimensional Settings

Anqi Zhao

Department of Statistics
Harvard University

ANQIZHAO@FAS.HARVARD.EDU

Yang Feng

Department of Statistics
Columbia University

YANGFENG@STAT.COLUMBIA.EDU

Lie Wang

Department of Mathematics
Massachusetts Institute of Technology

LIEWANG@MATH.MIT.EDU

Xin Tong

Department of Data Sciences and Operations
Marshall Business School
University of Southern California

XINT@MARSHALL.USC.EDU

Editor: Hui Zou

Abstract

Most existing binary classification methods target on the optimization of the overall classification risk and may fail to serve some real-world applications such as cancer diagnosis, where users are more concerned with the risk of misclassifying one specific class than the other. Neyman-Pearson (NP) paradigm was introduced in this context as a novel statistical framework for handling asymmetric type I/II error priorities. It seeks classifiers with a minimal type II error and a constrained type I error under a user specified level. This article is the first attempt to construct classifiers with guaranteed theoretical performance under the NP paradigm in high-dimensional settings. Based on the fundamental Neyman-Pearson Lemma, we used a plug-in approach to construct NP-type classifiers for Naive Bayes models. The proposed classifiers satisfy the NP oracle inequalities, which are natural NP paradigm counterparts of the oracle inequalities in classical binary classification. Besides their desirable theoretical properties, we also demonstrated their numerical advantages in prioritized error control via both simulation and real data studies.

Keywords: classification, high-dimension, Naive Bayes, Neyman-Pearson (NP) paradigm, NP oracle inequality, plug-in approach, screening

1. Introduction

Classification plays an important role in many aspects of our society. In medical research, identifying pathogenically distinct tumor types is central to advances in cancer treatments (Golub et al., 1999; Alderton, 2014). In cyber security, spam messages and virus make automatic categorical decisions a necessity. Binary classification is arguably the simplest and most important form of classification problems, and can serve as a building block for more complicated applications. We focus our attention on binary classification in this work.

A few common notations are introduced to facilitate our discussion. Let (X, Y) be a random pair where $X \in \mathcal{X} \subset \mathbb{R}^d$ is a vector of features and $Y \in \{0, 1\}$ indicates X 's class label. A classifier $\phi: \mathcal{X} \rightarrow \{0, 1\}$ is a mapping from \mathcal{X} to $\{0, 1\}$ that assigns X to one of the classes. A classification loss function is defined to assign a “cost” to each misclassified instance $\phi(X) \neq Y$, and the classification error is defined as the expectation of this loss function with respect to the joint distribution of (X, Y) . We will focus our discussion on the 0-1 loss function $\mathbb{I}\{\phi(X) \neq Y\}$ throughout the paper, where $\mathbb{I}(\cdot)$ denotes the indicator function. Denote by \mathbb{P} and \mathbb{E} the generic probability distribution and expectation, whose meaning depends on specific contexts. The classification error is $R(\phi) = \mathbb{E}\mathbb{I}\{\phi(X) \neq Y\} = \mathbb{P}\{\phi(X) \neq Y\}$. The law of total probability allows us to decompose it into a weighted average of type I error $R_0(\phi) = \mathbb{P}\{\phi(X) \neq Y | Y = 0\}$ and type II error $R_1(\phi) = \mathbb{P}\{\phi(X) \neq Y | Y = 1\}$ as

$$R(\phi) = \mathbb{P}(Y = 0)R_0(\phi) + \mathbb{P}(Y = 1)R_1(\phi). \quad (1.1)$$

With the advent of high-throughput technologies, classification tasks have experienced an exponential growth in the feature dimensions throughout the past decade. The fundamental challenge of “high dimension, low sample size” has motivated the development of a plethora of classification algorithms for various applications. While dependencies among features are usually considered a crucial characteristic of the data (Ackermann and Strimmer, 2009), and can effectively reduce classification errors under suitable models and relative data abundance (Shao et al., 2011; Cai and Liu, 2011; Fan et al., 2012; Mai et al., 2012; Witten and Tibshirani, 2012), independence rules, with their superb scalability, become a rule of thumb when the feature dimension grows faster than the sample size (Hastie et al., 2009; James et al., 2013). Despite Naive Bayes models’ reputation of being “simplistic” by ignoring all dependency structure among features, they lead to simple classifiers that have proven worthy on high-dimensional data with remarkably good performances in numerous real-life applications. Taking the classical model setting of two-class Gaussian with a common covariance matrix, Bickel and Levina (2004) showed the superior performance of Naive Bayes models over (naive implementation of) the Fisher linear discriminant rule under broad conditions in high-dimensional settings. Fan and Fan (2008) further established the necessity of feature selection for high-dimensional classification problems by showing that even independence rules can be as poor as random guessing due to noise accumulation. Featuring both independence rule and feature selection, the (sparse) Naive Bayes model remains a good choice for classification when the sample size is *fairly limited*.

1.1 Asymmetrical priorities on errors

Most existing binary classification methods target on the optimization of the overall risk (1.1) and may fail to serve the purpose when users’ relative priorities over type I/II errors differ significantly from those implied by the marginal probabilities of the two classes. A representative example of such scenario is the diagnosis of serious disease. Let 1 code the healthy class and 0 code the diseased class. Given that usually

$$\mathbb{P}(Y = 1) \gg \mathbb{P}(Y = 0),$$

minimizing the overall risk (1.1) might yield classifiers with small overall risk R (as a result of small R_1) yet large R_0 — a situation quite undesirable in practice given flagging a healthy

case incurs only extra cost of additional tests while failing to detect the disease endangers a life.

The neuroblastoma dataset introduced by Oberthner et al. (2006) provides a perfect illustration of such intuition. The dataset contains gene expression profiles on $d = 10707$ genes from 246 patients in a German neuroblastoma trial, among which 56 are high-risk (labeled as 0) and 190 are low-risk (labeled as 1). We randomly selected 41 '0's and 123 '1's as our training sample (such that the proportion of '0's is about the same as that in the entire dataset), and tested the resulting classifiers on the rest 15 '0's and 67 '1's. The average error rates of PSN² (to be proposed; implemented here at significance level 0.05), Gaussian Naive Bayes (nb), penalized logistic regression (pen-log), and Support Vector Machine (svm) over 1000 random splits are summarized in Table 1. All procedures except

Table 1: Average error rates over 1000 random splits for neuroblastoma dataset.

Error Type	PSN ²	nb	pen-log	svm
type I (0 as 1)	<u>.038</u>	.304	.529	.375
type II (1 as 0)	.761	.162	.103	.092

PSN² led to high type I errors, and are thus considered unsatisfactory given the more severe consequences of missing a diseased instance than vice versa.

One existing solution to asymmetric error control is *cost-sensitive learning*, which assigns two different costs as weights of the type I/II errors (Elkan, 2001; Lin et al., 2002; Zadrozny et al., 2003). Despite many merits and practical values of this framework, limitations arise in applications when there is no consensus over how much costs to be assigned to each class, or more fundamentally, whether it is morally acceptable to assign costs in the first place. Also, when users have a specific target for type I/II error control, cost-sensitive learning does not fit. Other methods aiming for small type I error include the Asymmetric Support Vector Machine (Wu et al., 2008), and the p -value for classification (Dünngeen et al., 2008). However, the former has no theoretical guarantee on errors, while the latter treats all classes as of equal importance.

1.2 Neyman-Pearson (NP) paradigm and NP oracle inequalities

Neyman-Pearson (NP) paradigm was introduced as a novel statistical framework for targeted type I/II error control. Assume type I error R_0 as the prioritized error type, this paradigm seeks to control R_0 under a user specified level α with R_1 as small as possible. The *oracle* is thus

$$\phi^* \in \operatorname{argmin}_{R_0(\phi) \leq \alpha} R_1(\phi), \quad (1.2)$$

where the *significance level* α reflects the level of conservativeness towards type I error. Given ϕ^* is unattainable in the learning paradigm, the best within our capability is to construct a data dependant classifier $\hat{\phi}$ that mimics it.

Despite its practical importance, NP classification has not received much attention in the statistics and machine learning communities. Cannon et al. (2002) initiated the theoretical treatment of NP classification. Under the same framework, Scott (2005) and Scott and

Nowak (2005) derived several results for traditional statistical learning such as PAC bounds or oracle inequalities. By combining type I and type II errors in sensible ways, Scott (2007) proposed a performance measure for NP classification. More recently, Blanchard et al. (2010) developed a general solution to semi-supervised novelty detection by reducing it to NP classification. Other related works include Cassesant and Chen (2003) and Han et al. (2008). A common issue with methods in this line of literature is that they all follow an empirical risk minimization (ERM) approach, and use some forms of relaxed empirical type I error constraint in the optimization program. As a result, all type I errors can only be proven to satisfy some relaxed upper bound. Take the framework set up by Cannon et al. (2002) for example. Given $\epsilon_0 > 0$, they proposed the program

$$\min_{\phi \in \mathcal{H}, R_0(\phi) \leq \alpha + \epsilon_0/2} \hat{R}_1(\phi),$$

where \mathcal{H} is a set of classifiers with finite Vapnik-Chervonenkis dimension, and \hat{R}_0, \hat{R}_1 are the empirical type I and type II errors respectively. It is shown that with high probability, the solution $\hat{\phi}$ to the above program satisfies simultaneously: i) the type I error $R_0(\hat{\phi})$ is bounded from above by $\alpha + \epsilon_0$, and ii) the type II error $R_1(\hat{\phi})$ is bounded from above by $R_1(\phi^*) + \epsilon_1$ for some $\epsilon_1 > 0$.

Rigollet and Tong (2011) is a significant departure from the previous NP classification literature. This paper argues that a good classifier $\hat{\phi}$ under the NP paradigm should respect the chosen significance level α , rather than some relaxation of it. More precisely, two **NP oracle inequalities** should be satisfied simultaneously with high probability:

- (I) the type I error constraint is respected, i.e., $R_0(\hat{\phi}) \leq \alpha$.
- (II) the excess type II error $R_1(\hat{\phi}) - R_1(\phi^*)$ diminishes with explicit rates (w.r.t. sample size).

Recall that, for a classifier \hat{h} , the classical oracle inequality insists that with high probability

$$\text{the excess risk } R(\hat{h}) - R(h^*) \text{ diminishes with explicit rates,} \quad (1.3)$$

where $h^*(x) = \mathbb{I}(g(x) \geq 1/2)$ is the Bayes classifier, in which $\eta(x) = \mathbb{E}[Y|X = x] = \mathbb{P}(Y = 1|X = x)$ is the regression function of Y on X (see Koltchinskii (2008) and references within). The two NP oracle inequalities defined above can be thought of as a generalization of (1.3) that provides a novel characterization of classifiers' theoretical performances under the NP paradigm.

Using a more stringent empirical type I error constraint (than the level α), Rigollet and Tong (2011) established NP oracle inequalities for its proposed classifiers under convex loss functions (as opposed to the indicator loss). They also proved an interesting negative result: under the binary loss, ERM approaches (convexification or not) cannot guarantee diminishing excess type II error as long as one insists type I error of the proposed classifier be bounded from above by α with high probability. This negative result motivated a plug-in approach to NP classification in Tong (2013).

1.3 Plug-in approaches

Plug-in methods in classical binary classification have been well studied in the literature, where the usual plug-in target is the Bayes classifier $\mathbb{I}(\eta(x) \geq 1/2)$. Earlier works gave rise to pessimism of the plug-in approach to classification. For example, under certain assumptions, Yang (1999) showed plug-in estimators cannot achieve excess risk with rates faster than $O(1/\sqrt{n})$, while direct methods can achieve rates up to $O(1/n)$ under *margin assumption* (Mammen and Tsybakov, 1999; Tsybakov, 2004; Tsybakov and van de Geer, 2005; Tarigan and van de Geer, 2006). However, it was shown in Audibert and Tsybakov (2007) that plug-in classifiers $\mathbb{I}(\hat{\eta}_n \geq 1/2)$ based on local polynomial estimators can achieve rates faster than $O(1/n)$, with a smoothness condition on η and the margin assumption.

The oracle classifier under the NP paradigm arises from its close connection to the Neyman-Pearson Lemma in statistical hypothesis testing. Hypothesis testing bears strong resemblance to binary classification if we assume the following model. Let P_1 and P_0 be two known probability distributions on $\mathcal{X} \subset \mathbb{R}^d$. Assume that $Y \sim \text{Bern}(\zeta)$ for some $\zeta \in (0, 1)$, and the conditional distribution of X given Y is P_Y . Given such a model, the goal of statistical hypothesis testing is to determine if we should reject the null hypothesis that X was generated from P_0 . To this end, we construct a randomized test $\phi : \mathcal{X} \rightarrow [0, 1]$ that rejects the null with probability $\phi(X)$. Two types of errors arise: type I error occurs when P_0 is rejected yet $X \sim P_0$, and type II error occurs when P_0 is not rejected yet $X \sim P_1$. The Neyman-Pearson paradigm in hypothesis testing amounts to choosing ϕ that solves the following constrained optimization problem

$$\text{maximize } \mathbb{E}[\phi(X)|Y = 1], \text{ subject to } \mathbb{E}[\phi(X)|Y = 0] \leq \alpha,$$

where $\alpha \in (0, 1)$ is the significance level of the test. A solution to this constrained optimization problem is called a *most powerful test* of level α . The Neyman-Pearson Lemma gives mild sufficient conditions for the existence of such a test.

Lemma 1.1 (Neyman-Pearson Lemma). *Let P_1 and P_0 be two probability measures with densities p and q respectively, and denote the density ratio as $r(x) = p(x)/q(x)$. For a given significance level α , let C_α be such that $P_0\{r(X) > C_\alpha\} \leq \alpha$ and $P_0\{r(X) \geq C_\alpha\} \geq \alpha$. Then, the most powerful test of level α is*

$$\phi^*(X) = \begin{cases} 1 & \text{if } r(X) > C_\alpha, \\ 0 & \text{if } r(X) < C_\alpha, \\ \frac{\alpha - P_0\{r(X) > C_\alpha\}}{P_0\{r(X) = C_\alpha\}} & \text{if } r(X) = C_\alpha. \end{cases}$$

Under mild continuity assumption, we take the NP oracle

$$\phi^*(x) = \phi_\alpha^*(x) = \mathbb{I}\{p(x)/q(x) \geq C_\alpha\} = \mathbb{I}\{r(x) \geq C_\alpha\}. \quad (1.4)$$

as our plug-in target for NP classification. With kernel density estimates \hat{p} , \hat{q} , and a proper estimate of the threshold level \hat{C}_α , Tong (2013) constructed a plug-in classifier $\mathbb{I}\{\hat{p}(x)/\hat{q}(x) \geq \hat{C}_\alpha\}$ that satisfies both NP oracle inequalities with high probability when the dimensionality is small, leaving the high-dimensional case an uncharted territory.

1.4 Contribution

In the big data era, NP classification framework faces the same curse of dimensionality as its classical counterpart. Despite its wide potential applications, this paper is the *first attempt* to construct performance-guaranteed classifiers under the NP paradigm in high-dimensional settings. Based on the Neyman-Pearson Lemma, we employ Naive Bayes models and propose a computationally feasible plug-in approach to construct classifiers that satisfy the NP oracle inequalities. We also improve the *detection condition*, a critical theoretical assumption first introduced in Tong (2013), for effective threshold level estimation that grounds the good NP properties of these classifiers. Necessity of the new detection condition is also discussed. Note that classifiers proposed in this work are not straightforward extensions of Tong (2013): kernel density estimation is now applied in combination with feature selection, and the threshold level is estimated in a more precise way by order statistics that require only moderate sample size — while Tong (2013) resorted to the Vapnik-Chervonenkis theory and required sample size much bigger than what is available in most high-dimensional applications.

The rest of the paper is organized as follows. Two screening based plug-in NP-type classifiers are presented in Section 2, where theoretical properties are also discussed. Performance of the proposed classifiers is demonstrated in Section 3 by both simulation studies and real data analysis. We conclude in Section 4 with a short discussion. The technical proofs are relegated to the Appendix.

2. Methods

In this section, we first introduce several notations and definitions, with a focus on the *detection condition*. Then we present the plug-in procedure, together with its theoretical properties.

2.1 Notations and definitions

We introduce here several notations adapted from Audibert and Tsybakov (2007). For $\beta > 0$, denote by $|\beta|$ the largest integer strictly less than β . For any $x, x' \in \mathbb{R}$ and any $|\beta|$ times continuously differentiable real-valued function $g(\cdot)$ on \mathbb{R} , we denote by g_x its Taylor polynomial of degree $|\beta|$ at point x . For $L > 0$, the $(\beta, L, [-1, 1])$ -Hölder class of functions, denoted by $\Sigma(\beta, L, [-1, 1])$, is the set of functions $g : [-1, 1] \rightarrow \mathbb{R}$ that are $|\beta|$ times continuously differentiable and satisfy, for any $x, x' \in [-1, 1]$, the inequality $|g(x') - g_x(x')| \leq L|x - x'|^\beta$. The $(\beta, L, [-1, 1])$ -Hölder class of density is defined as

$$\mathcal{P}_{\Sigma}(\beta, L, [-1, 1]) = \left\{ f : f \geq 0, \int f = 1, f \in \Sigma(\beta, L, [-1, 1]) \right\}.$$

We will use β -valid kernels (kernels of order β , Tsybakov (2009)) for all the kernel estimation throughout the theoretical discussion, the definition of which is as follows.

Definition 2.1. *Let $K(\cdot)$ be a real-valued function on \mathbb{R} with support $[-1, 1]$. The function $K(\cdot)$ is a β -valid kernel if it satisfies $\int K = 1$, $\int |K|^v < \infty$ for any $v \geq 1$, $\int |t|^\beta |K(t)| dt < \infty$, and in the case $|\beta| \geq 1$, it satisfies $\int t^l K(t) dt = 0$ for any $l \in \mathbb{N}$ such that $1 \leq l \leq \lfloor \beta \rfloor$.*

We assume that all the β -valid kernels considered in the theoretical part of this paper are constructed from Legendre polynomials, and are thus Lipschitz and bounded, satisfying the kernel conditions for the important technical Lemma A.6.

Definition 2.2 (margin assumption). *A function $f(\cdot)$ is said to satisfy margin assumption of order $\bar{\gamma}$ with respect to probability distribution P at the level C^* if there exists a positive constant M_0 , such that for any $\delta \geq 0$,*

$$P\{|f(X) - C^*| \leq \delta\} \leq M_0 \delta^{\bar{\gamma}}.$$

This assumption was first introduced in Polonik (1995). In the classical binary classification framework, Mannen and Tsybakov (1999) proposed a similar condition named “margin condition” by requiring most data to be away from the optimal decision boundary. In the classical classification paradigm, definition 2.2 reduces to the “margin condition” by taking $f = \eta$ and $C^* = 1/2$, with $\{x : |f(x) - C^*| = 0\} = \{x : \eta(x) = 1/2\}$ giving the decision boundary of the Bayes classifier. On the other hand, unlike the classical paradigm where the optimal threshold level is known and does not need an estimate, the optimal threshold level C_α in the NP paradigm is unknown and needs to be estimated, suggesting the necessity of having sufficient data around the decision boundary to detect it well. This concern motivated the following condition improved from Tong (2013).

Definition 2.3 (detection condition). *A function $f(\cdot)$ is said to satisfy detection condition of order $\bar{\gamma}$ with respect to P (i.e., $X \sim P$) at level (C^*, δ^*) if there exists a positive constant M_1 , such that for any $\delta \in (0, \delta^*)$,*

$$P\{C^* \leq f(X) \leq C^* + \delta\} \geq M_1 \delta^{\bar{\gamma}}.$$

A detection condition works as an opposite force to the margin assumption, and is basically an assumption on the lower bound of probability. Through we take here a power function of δ as the lower bound, so that it is simple and aesthetically similar to the margin assumption, any increasing function $u(\cdot)$ of δ on R^+ with $\lim_{\delta \rightarrow 0^+} u(\delta) = 0$ should be able to serve the purpose. The version of detection condition we would use to establish the NP inequalities for the (to be) proposed classifiers takes $f = r$, $C^* = C_\alpha$, and $P = P_0$ (recall that P_0 is the conditional distribution of X given $Y = 0$).

Now we argue why such a condition is *necessary* to achieve the NP oracle inequalities. Consider the simpler case where the density ratio r is known, and we only need a proper estimate of the threshold level \hat{C}_α . If there is nothing like the detection condition (Definition 2.3 involves a power function, but the idea is just to have any kind of lower bound), we would have, for some $\delta > 0$,

$$P_0\{C_\alpha \leq r(X) \leq C_\alpha + \delta\} = 0. \quad (2.1)$$

In getting the threshold estimate \hat{C}_α of $\hat{\phi}(x) = \mathbb{1}\{r(x) \geq \hat{C}_\alpha\}$, we can not distinguish any threshold level between C_α and $C_\alpha + \delta$. In particular, it is possible that

$$\hat{C}_\alpha > C_\alpha + \delta/2.$$

But then the excess type II error is bounded from below as follows

$$R_1(\hat{\phi}) - R_1(\phi^*) = P\{C_\alpha < r(X) < \hat{C}_\alpha\} > P\{C_\alpha < r(X) < C_\alpha + \delta/2\},$$

where the last quantity can be positive. Therefore, the second NP oracle inequality (diminishing excess type II error) does not hold for $\hat{\phi}$. Since some detection condition is necessary in this simpler case, it is certainly necessary in our real setup.

Note that Definition 2.3 is a significant improvement of the detection condition formulated in Tong (2013), which requires

$$P\{C^* - \delta \leq f(X) \leq C^*\} \wedge P\{C^* \leq f(X) \leq C^* + \delta\} \geq M_1 \delta^{\bar{\gamma}}.$$

We are able to drop the lower bound for the first piece due to an improved layout of the proofs. Intuitively, our new detection condition ensures an upper bound on \hat{C}_α . But we do not need an extra condition to get a lower bound of \hat{C}_α , because of the type I error bound requirement (see the proof of Proposition 2.4 for details). One example that satisfies the current condition but violate that in Tong (2013) is cited in the Appendix.

2.2 Neyman-Pearson plug-in procedure

Suppose the sampling scheme is fixed as follows.

Assumption 1. *Assume the training sample contains n i.i.d. observations $\mathbf{S}^1 = \{U_1, \dots, U_n\}$ from class 1 with density p , and m i.i.d. observations $\mathbf{S}^0 = \{V_1, \dots, V_m\}$ from class 0 with density q . Given fixed n_1, n_2, m_1, m_2 and m_3 such that $n_1 + n_2 = n$, $m_1 + m_2 + m_3 = m$, we further decompose \mathbf{S}^1 and \mathbf{S}^0 into independent subsamples as: $\mathbf{S}^1 = \mathbf{S}_1^1 \cup \mathbf{S}_2^1$, and $\mathbf{S}^0 = \mathbf{S}_1^0 \cup \mathbf{S}_2^0 \cup \mathbf{S}_3^0$, where $|\mathbf{S}_1^1| = n_1$, $|\mathbf{S}_2^1| = n_2$, $|\mathbf{S}_1^0| = m_1$, $|\mathbf{S}_2^0| = m_2$, $|\mathbf{S}_3^0| = m_3$.*

The sample splitting idea has been considered in the literature, such as in Meinshausen and Bühlmann (2010) and Robins et al. (2006). Given these samples, we introduce the following plug-in procedure.

Definition 2.4. Neyman-Pearson plug-in procedure

Step 1 Use \mathbf{S}_1^1 , \mathbf{S}_2^1 , \mathbf{S}_1^0 , and \mathbf{S}_2^0 to construct a density ratio estimate \hat{r} . The specific use of each subsample will be introduced in Section 2.4.

Step 2 Given \hat{r} , choose a threshold estimate \hat{C}_α from the set $\hat{r}(\mathbf{S}_3^0) = \{\hat{r}(V_{i+m_1+m_2})\}_{i=1}^{m_3}$.

Denote by $\hat{r}_{(k)}(\mathbf{S}_3^0)$ the k -th order statistic of $\hat{r}(\mathbf{S}_3^0)$, $k \in \{1, \dots, m_3\}$. The corresponding plug-in classifier by setting $\hat{C}_\alpha = \hat{r}_{(k)}(\mathbf{S}_3^0)$ is

$$\hat{\phi}_k(x) = \mathbb{1}\{\hat{r}(x) \geq \hat{r}_{(k)}(\mathbf{S}_3^0)\}. \quad (2.2)$$

A generic procedure for choosing the optimal k will be given in Section 2.3.

2.3 Threshold estimate \hat{C}_α

For any arbitrary density ratio estimate \hat{r} , we employ a proper order statistic $\hat{r}_{(k)}(\mathbf{S}_3^0)$ to estimate the threshold C_α and establish a probabilistic upper bound for the type I error of $\hat{\phi}_k$ for each $k \in \{1, \dots, m_3\}$.

Proposition 2.1. For any arbitrary density ratio estimate \hat{r} , let $\hat{\phi}_k(x) = \mathbb{I}\{\hat{r}(x) \geq \hat{r}_{(k)}(\mathcal{S}_3^0)\}$. It holds for any $\delta \in (0, 1)$ and $k \in \{1, \dots, m_3\}$ that

$$\mathbb{P}\{R_0(\hat{\phi}_k) > \delta\} \leq \text{Beta.cdf}_{k, m_3+1-k}(1 - \delta), \quad (2.3)$$

where $\text{Beta.cdf}_{k, m_3+1-k}(\cdot)$ is the CDF of $\text{Beta}(k, m_3+1-k)$. The inequality becomes equality when $F_{0,\hat{r}}(t) = F_0\{\hat{r}(X) \leq t\}$ is continuous almost surely.

In view of the above proposition, a sufficient condition for the classifier $\hat{\phi}_k$ to satisfy NP Oracle Inequality (I) at tolerance level $\delta_3 \in (0, 1)$ is thus

$$\text{Beta.cdf}_{k, m_3+1-k}(1 - \alpha) \leq \delta_3. \quad (2.4)$$

Despite the potential tightness of (2.3), we are not able to derive an explicit formula for the minimum k that satisfies (2.4). To get an explicit choice for k , we resort to concentration inequalities for an alternative.

Proposition 2.2. For any arbitrary density ratio estimate \hat{r} , let $\hat{\phi}_k(x) = \mathbb{I}\{\hat{r}(x) \geq \hat{r}_{(k)}(\mathcal{S}_3^0)\}$. It holds for any $\delta_3 \in (0, 1)$ and $k \in \{1, \dots, m_3\}$ that

$$\mathbb{P}\{R_0(\hat{\phi}_k) > g(\delta_3, m_3, k)\} \leq \delta_3, \quad (2.5)$$

where

$$g(\delta_3, m_3, k) = \frac{m_3 + 1 - k}{m_3 + 1} + \sqrt{\frac{k(m_3 + 1 - k)}{\delta_3(m_3 + 2)(m_3 + 1)^2}}. \quad (2.6)$$

Let $\mathcal{K} = \mathcal{K}(\alpha, \delta_3, m_3) = \{k \in \{1, \dots, m_3\} : g(\delta_3, m_3, k) \leq \alpha\}$. Proposition 2.2 implies that $k \in \mathcal{K}(\alpha, \delta_3, m_3)$ is a sufficient condition for the classifier $\hat{\phi}_k$ to satisfy NP Oracle Inequality (I). The next step is to characterize \mathcal{K} and choose some $k \in \mathcal{K}$, so that $\hat{\phi}_k$ has small excess type II error. Clearly, we would like to find the smallest element in \mathcal{K} .

Proposition 2.3. The minimum $k \in \{1, \dots, m_3 + 1\}$ that satisfies $g(\delta_3, m_3, k) \leq \alpha$ is

$$k_{\min}(\alpha, \delta_3, m_3) = \lceil (m_3 + 1)A_{\alpha, \delta_3}(m_3) \rceil, \quad (2.7)$$

where $\lceil z \rceil$ denotes the smallest integer larger than or equal to z , and

$$A_{\alpha, \delta_3}(m_3) = \frac{1 + 2\delta_3(m_3 + 2)(1 - \alpha) + \sqrt{1 + 4\delta_3(1 - \alpha)\alpha(m_3 + 2)}}{2\{\delta_3(m_3 + 2) + 1\}}.$$

Moreover,

1. $A_{\alpha, \delta_3}(m_3) \in (1 - \alpha, 1)$.
2. $\hat{r}_{(k_{\min}(\alpha, \delta_3, m_3))}(\mathcal{S}_3^0)$ is asymptotically the empirical $(1 - \alpha)$ -th quantile of $F_{0,\hat{r}}$ in the sense that

$$\lim_{m_3 \rightarrow \infty} \frac{k_{\min}(\alpha, \delta_3, m_3)}{m_3} = \lim_{m_3 \rightarrow \infty} A_{\alpha, \delta_3}(m_3) = 1 - \alpha.$$

3. For any $m_3 \geq 4/(\alpha\delta_3)$, we have $k_{\min}(\alpha, \delta_3, m_3) \leq m_3$, and thus

$$\mathcal{K}(\alpha, \delta_3, m_3) = \{k_{\min}(\alpha, \delta_3, m_3), k_{\min}(\alpha, \delta_3, m_3) + 1, \dots, m_3\}.$$

Introduce shorthand notations $k_{\min} = k_{\min}(\alpha, \delta_3, m_3)$, $\hat{r}_{(k)} = \hat{r}_{(k)}(\mathcal{S}_3^0)$, and $\hat{C}_\alpha = \hat{r}_{(\min\{k_{\min}, m_3\})}$. We will take

$$\hat{\phi}(x) = \mathbb{I}\{\hat{r}(x) \geq \hat{C}_\alpha\} = \begin{cases} \mathbb{I}\{\hat{r}(x) \geq \hat{r}_{(k_{\min})}\}, & \text{if } k_{\min} \leq m_3, \\ \mathbb{I}\{\hat{r}(x) \geq \hat{r}_{(m_3)}\}, & \text{if } k_{\min} = m_3 + 1 \end{cases} \quad (2.8)$$

as the default NP plug-in classifier for any arbitrary \hat{r} . An alternative threshold estimate that also guarantees type I error bound is derived in the Appendix C. Assume $m_3 \geq 4/(\alpha\delta_3)$ for the rest of the theoretical discussion. It follows from Proposition 2.3 that $k_{\min} \leq m_3$, and thus $\hat{C}_\alpha = \hat{r}_{(k_{\min})}$, $\hat{\phi} = \hat{\phi}_{(k_{\min})}$ with guaranteed type I error control.

Remark 2.1. Note that $\lim_{m_3 \rightarrow \infty} k_{\min}/\lceil m_3(1 - \alpha) \rceil = 1$. Thus, choosing the k_{\min} -th order statistic of $\hat{r}(\mathcal{S}_3^0)$ as the threshold can be viewed as a modification to the classical approach of estimating the $1 - \alpha$ quantile of $F_{0,\hat{r}}$ by the $\lceil m_3(1 - \alpha) \rceil$ -th order statistic of $\hat{r}(\mathcal{S}_3^0)$. Recall that the oracle C_α is actually the $1 - \alpha$ quantile of distribution $F_{0,r}$, so the intuition is that \hat{C}_α is asymptotically (when $m_3 \rightarrow \infty$) equivalent to the $1 - \alpha$ quantile of $F_{0,\hat{r}}$, which in turn converges (when $n_1, n_2, m_1, m_2 \rightarrow \infty$) to C_α as the $1 - \alpha$ quantile of $F_{0,r}$ under moderate conditions.

Lemma 2.1. Let $\alpha, \delta_3 \in (0, 1)$. In addition to Assumption 1, suppose \hat{r} be such that $F_{0,\hat{r}}$ is continuous almost surely. Then for any $\delta_4 \in (0, 1)$ and $m_3 \geq 4/(\alpha\delta_3)$, the distance between $R_0(\hat{\phi})$ ($\hat{\phi}$ as defined in (2.8)) and $R_0(\phi^*)$ can be bounded as

$$\mathbb{P}\{|R_0(\hat{\phi}) - R_0(\phi^*)| > \xi_{\alpha, \delta_3, m_3}(\delta_4)\} \leq \delta_4,$$

where

$$\xi_{\alpha, \delta_3, m_3}(\delta_4) = \sqrt{\frac{k_{\min}(m_3 + 1 - k_{\min})}{(m_3 + 2)(m_3 + 1)^2\delta_4}} + A_{\alpha, \delta_3}(m_3) - (1 - \alpha) + \frac{1}{m_3 + 1}. \quad (2.9)$$

If $m_3 \geq \max\{\delta_3^{-2}, \delta_4^{-2}\}$, we have $\xi_{\alpha, \delta_3, m_3}(\delta_4) \leq (5/2)m_3^{-1/4}$.

Proposition 2.4. Let $\alpha, \delta_3, \delta_4 \in (0, 1)$. In addition to assumptions of Lemma 2.1, assume that the density ratio r satisfies the margin assumption of order $\bar{\gamma}$ at level C_α (with constant M_0) and detection condition of order $\underline{\gamma}$ at level (C_α, δ^*) (with constant M_1), both with respect to distribution P_0 . If $m_3 \geq \max\{4/(\alpha\delta_3), \delta_3^{-2}, \delta_4^{-2}, (\frac{2}{5}M_1\delta^{*\bar{\gamma}})^{-4}\}$, the excess type II error of the classifier $\hat{\phi}$ defined in (2.8) satisfies with probability at least $1 - \delta_3 - \delta_4$,

$$\begin{aligned} & R_1(\hat{\phi}) - R_1(\phi^*) \\ & \leq 2M_0 \left[\left\{ \frac{|R_0(\hat{\phi}) - R_0(\phi^*)|}{M_1} \right\}^{1/2} + 2\|\hat{r} - r\|_\infty \right]^{1+\bar{\gamma}} + C_\alpha |R_0(\hat{\phi}) - R_0(\phi^*)| \\ & \leq 2M_0 \left[\left(\frac{2}{5}M_3^{1/4}M_1 \right)^{-1/2} + 2\|\hat{r} - r\|_\infty \right]^{1+\bar{\gamma}} + C_\alpha \left(\frac{2}{5}M_3^{1/4} \right)^{-1}. \end{aligned}$$

Given the above proposition, we can control the excess (type II error as long as the uniform deviation of density ratio estimate $\|\hat{r} - r\|_\infty$ is controlled. In the following subsection, we will introduce estimates \hat{r} and provide bounds for $\|\hat{r} - r\|_\infty$.

2.4 Density ratio estimate \hat{r}

Denote the marginal densities of class 1 and 0 as p_j and q_j ($j = 1, \dots, d$) respectively, Naive Bayes models for the density ratio take the form

$$r(x) = \prod_{j=1}^d \frac{p_j(x_j)}{q_j(x_j)}, \quad \text{where } x_j \text{ is the } j\text{-th component of } x.$$

The subsamples $S_1^j = \{U_{i_j}^{n_1}\}_{i_j=1}^{n_1}$, $S_2^j = \{U_{i_j+n_1}^{n_2}\}_{i_j=1}^{n_2}$, $S_0^j = \{V_i\}_{i=1}^{m_1}$ and $S_0^j = \{V_i+m_1\}_{i=1}^{m_2}$ are used to construct (nonparametric/parametric) estimators of p_j and q_j for $j = 1, \dots, d$.

Nonparametric estimate of the density ratio. For marginal densities p_j and q_j , we apply kernel estimates $\hat{p}_j(x_j) = \{(n_1 + m_2)h_1\}^{-1} \sum_{i=1}^{n_1+m_2} K\left(\frac{U_{i_j}-x_j}{h_1}\right)$, and $\hat{q}_j(x_j) = \{(m_1 + m_2)h_0\}^{-1} \sum_{i=1}^{m_1+m_2} K\left(\frac{V_{i_j}-x_j}{h_0}\right)$, where $K(\cdot)$ is the kernel function, h_1, h_0 are the bandwidths, and V_{i_j} and U_{i_j} denote the j -th component of V_i and U_i respectively. The resulting nonparametric estimate is

$$\hat{r}_N(x) = \prod_{j=1}^d \frac{\hat{p}_j(x_j)}{\hat{q}_j(x_j)}. \quad (2.10)$$

Parametric estimate of the density ratio. Assume the two-class Gaussian model $XY = 0 \sim \mathcal{N}(\mu^0, \Sigma)$ and $XY = 1 \sim \mathcal{N}(\mu^1, \Sigma)$, where $\Sigma = \text{diag}(\sigma_1^2, \dots, \sigma_d^2)$. We estimate μ^0, μ^1 and Σ using their sample versions $\hat{\mu}^0, \hat{\mu}^1$ and $\hat{\Sigma}$. Under this model, the density ratio function is given by

$$r_P(x) = \exp \left\{ (\mu^1 - \mu^0)' \Sigma^{-1} x + \frac{1}{2} (\mu^0)' \Sigma^{-1} \mu^0 - \frac{1}{2} (\mu^1)' \Sigma^{-1} \mu^1 \right\},$$

and the corresponding parametric estimate is

$$\hat{r}_P(x) = \exp \left\{ (\hat{\mu}^1 - \hat{\mu}^0)' \hat{\Sigma}^{-1} x + \frac{1}{2} (\hat{\mu}^0)' \hat{\Sigma}^{-1} \hat{\mu}^0 - \frac{1}{2} (\hat{\mu}^1)' \hat{\Sigma}^{-1} \hat{\mu}^1 \right\}. \quad (2.11)$$

2.5 Screening-based density ratio estimate and plug-in procedures

For ‘‘high dimension, low sample size’’ applications, complex models that take into account all features usually fail: even Naive Bayes models that ignore feature dependency might lead to poor performance due to noise accumulation (Fan and Fan, 2008). A common solution in these scenarios is to first study marginal relations between the response and each of the features (Fan and Lv, 2008; Li et al., 2012). By selecting the most important individual features, we greatly reduce the model size, and other models can be applied after this screening step. We now introduce screening based variants of \hat{r}_N and \hat{r}_P . Let F_j^0 and F_j^1 denote the CDFs of q_j and p_j respectively, for $j = 1, \dots, d$. Step 1 of Procedure 2.4

introduced in Section 2.1 is now decomposed into a screening substep and an estimation substep.

Nonparametric Screening-based NP Naive Bayes (NSN²) classifier

Step 1.1 Select features using S_0^j and S_1^j as follows:

$$\hat{\mathcal{A}}_\tau = \left\{ 1 \leq j \leq d : \|F_j^0 - F_j^1\|_\infty \geq \tau \right\}, \quad (2.12)$$

where $\tau > 0$ is some threshold level, and

$$\hat{F}_j^0(x_j) = \frac{1}{m_1} \sum_{i=1}^{m_1} \mathbb{I}(V_{i_j} \leq x_j), \quad \hat{F}_j^1(x_j) = \frac{1}{n_1} \sum_{i=1}^{n_1} \mathbb{I}(U_{i_j} \leq x_j) \quad (2.13)$$

are the empirical CDFs.

Step 1.2 Use S_0^j and S_1^j to construct kernel estimates of q_j and p_j for $j \in \hat{\mathcal{A}}_\tau$. The density ratio estimate is given by

$$\hat{r}_N^S(x) = \prod_{j \in \hat{\mathcal{A}}_\tau} \frac{\hat{p}_j(x_j)}{\hat{q}_j(x_j)}.$$

Step 2 Given \hat{r}_N^S , use S_0^j to get a threshold estimate $(\hat{r}_N^S)_{(k_{\min})}$ as in (2.8).

The resulting NSN² classifier is

$$\hat{\phi}_{\text{NSN}^2}^S(x) = \mathbb{I} \left\{ \hat{r}_N^S(x) \geq (\hat{r}_N^S)_{(k_{\min})} \right\}. \quad (2.14)$$

Parametric Screening-based NP Naive Bayes (PSN²) classifier

The PSN² procedure is similar to NSN², except the following two differences. In Step 1.1, features are now selected based on t -statistics (\mathcal{A}_τ represent the index set of the selected features). In Step 1.2, p_j, q_j for $j \in \hat{\mathcal{A}}_\tau$ follow two-class Gaussian model, and the resulting parametric screening-based density ratio estimate is

$$\hat{r}_P^S(x) = \prod_{j \in \hat{\mathcal{A}}_\tau} \frac{\hat{p}_j(x_j)}{\hat{q}_j(x_j)}.$$

The corresponding PSN² classifier is thus given by

$$\hat{\phi}_{\text{PSN}^2}^S(x) = \mathbb{I} \left\{ \hat{r}_P^S(x) \geq (\hat{r}_P^S)_{(k_{\min})} \right\}. \quad (2.15)$$

We assume the domains of all p_j and q_j to be $[-1, 1]$ for all the following theoretical discussion. We will prove NP oracle inequalities for $\hat{\phi}_{\text{NSN}^2}^S$, and those for $\hat{\phi}_{\text{PSN}^2}^S$ can be developed similarly. Recall that by Proposition 2.4, we need an upper bound for $\|\hat{r}_N^S - r\|_\infty$. Necessarily, performance of the screening step should be studied. To this end, we assume that only a small fraction of the d features have marginal differentiating power.

Assumption 2. *There exists a signal set $\mathcal{A} \subset \{1, \dots, d\}$ with size $|\mathcal{A}| = s \ll d$ such that $\inf_{j \in \mathcal{A}} \|F_j^0 - F_j^1\|_\infty \geq D$ for some positive constant D , and $F_j^0 = F_j^1$ for $j \notin \mathcal{A}$.*

The following proposition shows that Step 1.1 achieves exact recovery ($\hat{\mathcal{A}}_\tau = \mathcal{A}$) with high probability for some properly chosen τ .

Proposition 2.5 (exact recovery). *Let $\delta_1 \in (0, 1)$. In addition to Assumptions 1 and 2, suppose $n_1 \wedge m_1 \geq 8D^{-2} \log(4d/\delta_1)$. Then for any $\tau \in [\Delta_0, D - \Delta_0]$, where $\Delta_0 = \sqrt{\frac{\log(4d/\delta_1)}{2n_1}} + \sqrt{\frac{\log(4d/\delta_1)}{2m_1}}$, the screening substep Step 1.1 (2.12) satisfies*

$$\mathbb{P}(\hat{\mathcal{A}}_\tau = \mathcal{A}) \geq 1 - \delta_1.$$

Now we are ready to control the uniform deviation of density ratio estimate given in Step 1.2.

Assumption 3. *The marginal densities $p_j, q_j \in \mathcal{P}_{\Sigma}(\beta, L, [-1, 1])$ for all $j = 1, \dots, d$, and there exists $\underline{\mu} > 0$ such that $p_j, q_j \geq \underline{\mu}$ for all $j \in \mathcal{A}$. There exists some constant $\bar{C} > 0$, such that $\|r\|_\infty \leq \bar{C}$, and there is a uniform absolute upper bound for $\|p_j^{(l)}\|_\infty$ and $\|q_j^{(l)}\|_\infty$ for $j \in \mathcal{A}$ and $l \in [0, \lfloor \beta \rfloor]$. Moreover, the kernel K in the nonparametric density estimates is β -valid and L -Lipschitz.*

Smoothness conditions (Assumption 3) and the margin assumption were used together in the classical classification literature. However, it is not entirely obvious why Assumption 3 does not render the detection condition redundant. We refer interested readers to Appendix B for more detailed discussion.

Let C_j^1 and C_j^0 be the constants C in Lemma A.6 when applied to p_j and q_j respectively. Assumption 3 ensures the existence of absolute constants $C^1 \geq \sup_{j \in \mathcal{A}} C_j^1$ and $C^0 \geq \sup_{j \in \mathcal{A}} C_j^0$.

Proposition 2.6 (uniform deviation of density ratio estimate). *Under Assumptions 1 - 3, for any $\delta_1, \delta_2 \in (0, 1)$, if $n_1 \wedge m_1 \geq 8D^{-2} \log(4d/\delta_1)$, $\sqrt{\frac{\log(2m_2s/\delta_2)}{n_2h_0}} \leq \min(1, \underline{\mu}/C^1)$, $\sqrt{\frac{\log(2m_2s/\delta_2)}{m_2h_0}} \leq \min(1, \underline{\mu}/C^0)$, and the screening threshold τ is specified as in Proposition 2.5, we have*

$$\mathbb{P}(\|\hat{r}_N^S - r\|_\infty \leq T) \geq 1 - \delta_1 - \delta_2, \quad (2.16)$$

where $T = Be^B \|r\|_\infty$ with

$$B = s \left\{ \frac{C^1 \sqrt{\frac{\log(2m_2s/\delta_2)}{n_2h_1}}}{\underline{\mu} - C^1 \sqrt{\frac{\log(2m_2s/\delta_2)}{n_2h_1}}} + \frac{C^0 \sqrt{\frac{\log(2m_2s/\delta_2)}{m_2h_0}}}{\underline{\mu} - C^0 \sqrt{\frac{\log(2m_2s/\delta_2)}{m_2h_0}}} \right\}.$$

Moreover, assume that $n_2 \wedge m_2 \geq 1/\delta_2$, $|\mathcal{A}| = s \leq (n_2 \wedge m_2)^{\frac{\beta}{2(\beta+1)}}$, and the bandwidths $h_1 = (\log n_2/n_2)^{\frac{\beta}{2\beta+1}}$ and $h_0 = (\log m_2/m_2)^{\frac{1}{2\beta+1}}$, then there exists an absolute constant $C_2 > 0$ such that

$$\mathbb{P} \left[\|\hat{r}_N^S - r\|_\infty \leq C_2 s \left\{ \left(\frac{\log n_2}{n_2} \right)^{\frac{\beta}{2\beta+1}} + \left(\frac{\log m_2}{m_2} \right)^{\frac{\beta}{2\beta+1}} \right\} \geq 1 - \delta_1 - \delta_2 \right]$$

The condition $|\mathcal{A}| = s \leq (n_2 \wedge m_2)^{\frac{\beta}{2(\beta+1)}}$ in the above proposition ensures that the upper bound of the uniform deviation diminishes as sample sizes n_2, m_2 go to infinity. Now we are in a position to present the theorem finale of NSN².

Theorem 2.1 (NP Oracle Inequalities for $\hat{\phi}_{\text{NSN}^2}$). *In addition to Assumptions 1 - 3, assume the density ratio r satisfies the margin assumption of order $\bar{\gamma}$ at level C_α and detection condition of order $\underline{\gamma}$ at level (C_α, δ^*) , both with respect to P_0 . For any given $\delta_1, \delta_2, \delta_3, \delta_4 \in$*

(0, 1), let the NSN² classifier $\hat{\phi}_{\text{NSN}^2}$ be defined as in (2.14), with the screening threshold τ specified as in Proposition 2.5 and kernel bandwidths $h_1 = (\log n_2/n_2)^{\frac{\beta}{2\beta+1}}$ and $h_0 = (\log m_2/m_2)^{\frac{\beta}{2\beta+1}}$, and \hat{r}_N^S be such that $F_{0, \hat{r}_N^S} = F_0\{\hat{r}_N^S \leq t\}$ is continuous almost surely.

*For subsample sizes that satisfy $n_1 \wedge m_1 \geq 8D^{-2} \log(4d/\delta_1)$, $n_2 \wedge m_2 \geq \max\{\delta_2^{-1}, s^{\frac{2(\beta+1)}{\beta}}\}$, $\sqrt{\frac{\log(2m_2s/\delta_2)}{n_2h_1}} \leq \min(1, \underline{\mu}/C^1)$, $\sqrt{\frac{\log(2m_2s/\delta_2)}{m_2h_0}} \leq \min(1, \underline{\mu}/C^0)$, and $m_3 \geq \max\{4/(\alpha\delta_3), \delta_3^{-2}, \delta_4^{-2}, (\frac{2}{5}M_1\delta^{*2})^{-4}\}$, there exists an absolute constant $\tilde{C} > 0$ such that with probability at least $1 - \delta_1 - \delta_2 - \delta_3 - \delta_4$,*

$$(I) \quad R_0(\hat{\phi}_{\text{NSN}^2}) \leq \alpha, \\ (II) \quad R_1(\hat{\phi}_{\text{NSN}^2}) - R_1(\phi^*) \leq \tilde{C} \left\{ m_3^{-\left(\frac{1}{4} \wedge \frac{1-\beta}{2}\right)} + s^{1+\bar{\gamma}} \left(\frac{\log n_2}{n_2} \right)^{\frac{\beta(1+\beta)}{2\beta+1}} + s^{1+\bar{\gamma}} \left(\frac{\log m_2}{m_2} \right)^{\frac{\beta(1+\beta)}{2\beta+1}} \right\}.$$

Theorem 2.1 establishes the NP oracle inequalities for $\hat{\phi}_{\text{NSN}^2}$. To help understand the conditions of this theorem, recall that Assumption 1 is about sample splitting, Assumption 2 is on minimal signal strength for active feature set, Assumption 3 is on marginal densities and kernels in nonparametric estimates, and the margin assumption and detection condition describe the neighbourhood of the oracle decision boundary. Note that the subsample sizes n_1 and m_1 do not enter the upper bound for the excess type II error explicitly. Instead, we have size requirements on them so that the important features are kept with high probability $1 - \delta_1$ in the screening substep. The tolerance parameter δ_2 arises from the nonparametric estimation of densities, the parameter δ_3 is for the tolerance on violation of type I error bound, and δ_4 arises from controlling $|R_0(\hat{\phi}_{\text{NSN}^2}) - R_0(\phi^*)|$.

3. Numerical investigation

In this section, we analyze two simulated examples and two real datasets to demonstrate the performance of our newly proposed NSN² and PSN² classifiers, in comparison with their corresponding non-screening counterparts (denoted as NN² and PN² respectively) as well as three popular methods under the classical framework: Gaussian Naive Bayes (nb), penalized logistic regression (pen-log), and Support Vector Machine (svm). We use R package ‘‘e1071’’ for nb and svm, and the R package ‘‘glmnet’’ for pen-log. Note that for fair comparison, we also include a pre-screening step for nb and svm under the high-dimensional settings. To facilitate the presentation, we summarize the four Neyman-Pearson Naive Bayes classifiers in Table 2.

To train the classifiers in Table 2, we set $\alpha = 0.05$, $\delta_1 = 0.05$, and $\delta_3 = 0.05$ throughout this section unless specified otherwise. In Assumption 1, motivated by Proposition 2.5, we

Table 2: A summary of the four Neyman-Pearson Naive Bayes classifiers.

	Screening-based	Non-screening
Non-parametric	$\hat{\phi}_{\text{NSN}^2}^{\text{NS}}(x) = \mathbb{I}\left\{\hat{r}_{\text{N}}^{\text{NS}}(x) \geq (\hat{r}_{\text{N}}^{\text{NS}})_{(k_{\text{min}})}\right\}$	$\hat{\phi}_{\text{NN}^2}^{\text{NS}}(x) = \mathbb{I}\left\{\hat{r}_{\text{N}}^{\text{NS}}(x) \geq (\hat{r}_{\text{N}}^{\text{NS}})_{(k_{\text{min}})}\right\}$
Parametric	$\hat{\phi}_{\text{PSN}^2}^{\text{NS}}(x) = \mathbb{I}\left\{\hat{r}_{\text{P}}^{\text{NS}}(x) \geq (\hat{r}_{\text{P}}^{\text{NS}})_{(k_{\text{min}})}\right\}$	$\hat{\phi}_{\text{PN}^2}^{\text{NS}}(x) = \mathbb{I}\left\{\hat{r}_{\text{P}}^{\text{NS}}(x) \geq (\hat{r}_{\text{P}}^{\text{NS}})_{(k_{\text{min}})}\right\}$

take $m_1 = \min\{10 \log(4d/\delta_1), m/4\} \mathbb{I}(\text{screening})$, $n_1 = \min\{10 \log(4d/\delta_1), n/2\} \mathbb{I}(\text{screening})$, $m_2 = \lfloor m/2 \rfloor - m_1$, $n_2 = n - n_1$, and $m_3 = m - \lfloor m/2 \rfloor$.

Due to the absence of information with respect to the true p and q , the theoretical screening cutoff that achieves exact recovery is not feasible in practice. We resort to an empirical permutation-based approach (Fan et al., 2011) as a substitute. Specifically, the screening substep in NSN² is executed as follows:

1. Combine S_0^0 and S_1^1 into $\{(X_i, Y_i)\}_{i=1}^{m_1+n_1}$, where $X_i \in S_0^0 \cup S_1^1$, and Y_i is X_i 's class label.
2. Calculate the marginal D -statistic for each feature:

$$D_j = \|\hat{F}_j^0 - \hat{F}_j^1\|_{\infty}, \quad j = 1, 2, \dots, d,$$

where $\hat{F}_j^0(x) = \sum_{i:Y_i=0} \mathbb{I}(X_{i,j} \leq x_j)$ and $\hat{F}_j^1(x) = \sum_{i:Y_i=1} \mathbb{I}(X_{i,j} \leq x_j)$.

3. Let $\pi = \{\pi(1), \dots, \pi(m_1 + n_1)\}$ be a random permutation of $\{1, \dots, (m_1 + n_1)\}$. For $j = 1, \dots, d$, compute $D_j^{\text{null}} = \|\hat{F}_j^{0,\text{null}} - \hat{F}_j^{1,\text{null}}\|_{\infty}$, where $\hat{F}_j^{0,\text{null}}(x_j) = \sum_{i:\pi(i)=0} \mathbb{I}(X_{i,j} \leq x_j)$, $\hat{F}_j^{1,\text{null}}(x_j) = \sum_{i:\pi(i)=1} \mathbb{I}(X_{i,j} \leq x_j)$.

4. For some pre-specified $Q \in [0, 1]$, let $\omega(Q)$ be the Q -th quantile of $\{D_j^{\text{null}} : j = 1, \dots, d\}$ and select $\hat{\mathcal{A}} = \{j : D_j \geq \omega(Q)\}$. Here, Q is a tuning parameter that keeps the percentage of noise features that pass the screening around $1 - Q$.

The same permutation idea is applied to the screening substep of PSN². Q is set at 0.95 throughout this section.

3.1 Simulation

Samples in both simulated examples are generated from the model

$$p(x) = \prod_{j=1}^d p_j(x_j), \quad q(x) = \prod_{j=1}^d q_j(x_j)$$

at 3 different dimensions: $d \in \{10, 100, 1000\}$. Sparsity for $d = 100$ and 1000 is imposed by setting $p_j = q_j$ for all $j > 10$. Seven different training sample sizes: $m = n \in \{200, 400, 800, 1600, 3200, 6400, 12800\}$ are considered. The number of replications for each scenario is 1000. Test errors are estimated using the average of 1000 independent observations from each class for each replication.

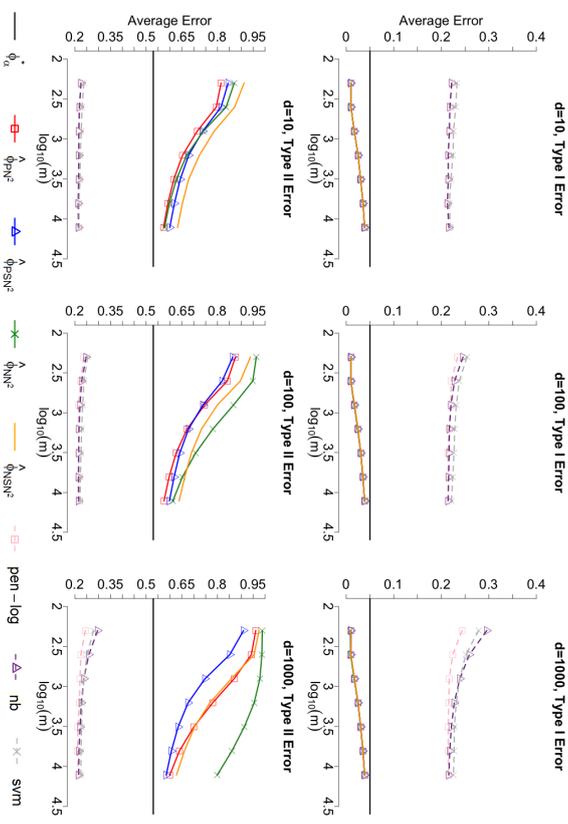
3.1.1 EXAMPLE 1: NORMALS WITH DIFFERENT MEANS

Assume the two-class conditional densities $p \sim \mathcal{N}(0.5(1_{10}^T, 0_{d-10}^T)^T, I_d)$ and $q \sim \mathcal{N}(0_d, I_d)$ where I_d is the identity matrix. At significance level $\alpha = 0.05$, the oracle type I/II risks are $R_0(\phi_{\alpha}^*) = 0.05$ and $R_1(\phi_{\alpha}^*) = 0.53$ respectively.

We first evaluate the screening performance of PSN² and NSN² with results presented in Table 3. Both t -statistic (in PSN²) and D -statistic (in NSN²) are able to pick up most of the true signals while keeping the false positive rates at around $1 - Q$.

 Table 3: Average screening performance summarized over 1000 independent replications at sample sizes $m = n = 400$ and $Q = 0.95$ with standard errors in parentheses.

d	# of selected features		# of missed signals		# of false positive	
	t -stat	D -stat	t -stat	D -stat	t -stat	D -stat
10	9.11 (1.14)	8.11 (1.63)	0.89 (1.14)	1.89 (1.63)	0 (0)	0 (0)
100	14.64 (3.46)	12.43 (3.38)	0.78 (0.90)	2.00 (1.39)	5.43 (3.17)	4.43 (2.77)
1000	59.99 (9.77)	58.82 (9.87)	0.48 (0.66)	1.14 (1.05)	50.47 (9.71)	49.96 (9.78)

 Figure 1: Average errors of $\hat{\phi}$'s over 1000 independent replications for each combination of (d, m, n) .


We then move on to evaluate the trend of type I and type II errors as the sample size increases in Figure 1. All the Neyman-Pearson based classifiers have type I error approaching α from below as sample size increases and they have similar type I errors at each sample size. However, nb, pen-log and svm all lead to a type I error larger than α .

By enlarging the second row of Figure 1, one would observe the differences in type II errors among PN^2 , PSN^2 , NN^2 , NSN^2 . In the case of $d = 10$ when all features are signals, PN^2 performs the best throughout all sample sizes since it assumes the correct model without the unnecessary screening substep. When sample size is small, PSN^2 outperforms NN^2 , but NN^2 gradually catches up on larger samples. In the case of $d = 100$, screening helps PSN^2 to take the lead at low sample sizes. The advantage of screening fades off as the sample size increases. In the case of $d = 1000$, PSN^2 dominates all other three classifiers throughout the sample size range we investigate.

Overall, the advantage of PSN^2 over NSN^2 , and PN^2 over NN^2 are uniform across all dimensions and sample sizes. This is consistent with the intuition that when the data are from a two-class Gaussian model, the parametric methods lead to more efficient estimators than nonparametric counterparts.

3.1.2 EXAMPLE 2: NORMAL VS. MIXTURE NORMAL

Normality assumption is violated in the second example. Assume $p \sim 0.5\mathcal{N}(a, \Sigma) + 0.5\mathcal{N}(-a, \Sigma)$ and $q \sim \mathcal{N}(0_d, I_d)$, where $a = \begin{pmatrix} 3 \\ -10 \\ 0 \end{pmatrix}$, $\Sigma = \begin{pmatrix} 10^{-1}I_{10} & 0 \\ 0 & I_{d-10} \end{pmatrix}$. At significance level $\alpha = 0.05$, the oracle type I/II risks are $R_0(\phi_\alpha^*) = 0.05$ and $R_1(\phi_\alpha^*) = 0.027$ respectively.

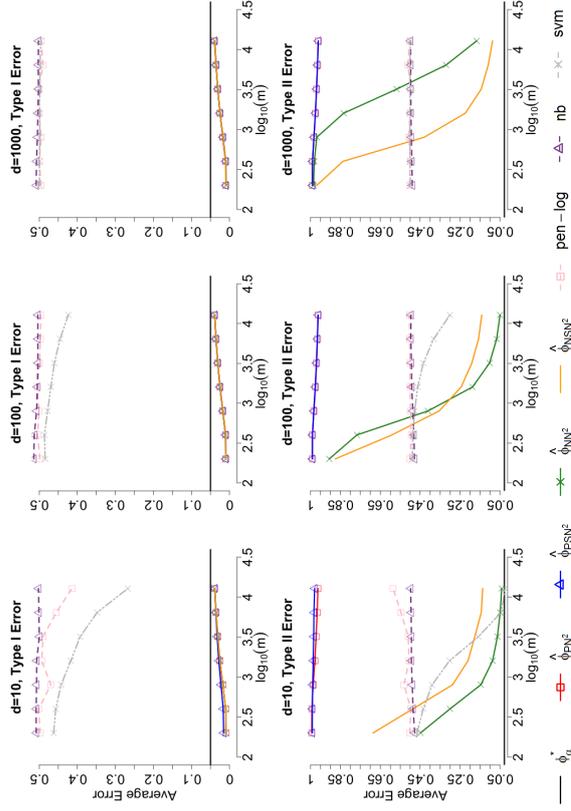
The performance of the screening substep of PSN^2 and NSN^2 is shown in Table 4. While both screening methods keep the false positive rates at around $1 - Q$, the parametric screening method (PSN^2) with t -statistic misses almost all signals. This is not surprising since t -statistics rank features by differences in means and the two groups have exactly the same marginal mean and variance across all dimensions.

Table 4: Average screening performance summarized over 1000 independent replications at sample sizes $m = n = 400$ and $Q = 0.95$ with standard errors in parentheses.

d	# of selected features		# of missed signals		# of false positive	
	t -stat	D -stat	t -stat	D -stat	t -stat	D -stat
10	1.76 (1.53)	8.13 (1.83)	8.24 (1.53)	1.87 (1.83)	0 (0)	0 (0)
100	5.93 (3.44)	11.96 (3.57)	9.38 (0.80)	2.34 (1.59)	5.31 (3.17)	4.29 (2.68)
1000	50.69 (9.60)	58.78 (9.87)	9.50 (0.69)	1.26 (1.04)	50.19 (9.51)	50.04 (9.62)

Figure 2 presents the average error rates. The same reason that causes the above fiasco of t -statistic screening reduces PSN^2 and PN^2 to nothing more than, if not less than, two unfair random coins with probability 0.05 of landing 1, while the behaviors of nb and pen-log bear more resemblance to that of fair random coins. This fundamental difference is due to that the classical framework aims to minimize the overall risk, and therefore tends to distribute errors evenly when the sample size for the two classes are about the same. The NSN^2 and NN^2 based on nonparametric assumptions, on the other hand, perform very

Figure 2: Average error rates of $\hat{\phi}^*$'s over 1000 independent replications for each combination of (d, m, n) . Error rates are computed as the average of 1000 independent testing data points from each class in each replication, and then average over replications.



well on non-normal data. Their difference in type II error performances are similar as in Example 1.

From the two simulation examples, it is clear that the screening-based NSN^2 and PSN^2 exhibit advantages over their non-screening counterparts under high-dimensional settings. When the normality assumption is violated, and the sample sizes are reasonably large for efficient kernel estimates, NSN^2 prevails over PSN^2 . As a rule of thumb, for high-dimensional classification problems that emphasize type I error control, we recommend NSN^2 if the sample size is relatively large and PSN^2 otherwise.

3.2 Real data analysis

In addition to the neuroblastoma dataset analyzed in the introduction, we now demonstrate the performance of PSN^2 and NSN^2 for targeted asymmetric error control on two additional real datasets.

3.2.1 P53 MUTANTS DATASET

The p53 mutants dataset (Danziger et al., 2006) contains $d = 5407$ attributes extracted from biophysical experiments for 16772 mutant p53 proteins, among which 143 are determined as “active” and the rest as “inactive” via in vivo assays.

All 143 active samples and the first 1500 inactive samples are included in our analysis. We treat the active class as class 0 and aimed to control the error of missing an active under $\alpha = 0.05$. This dataset is split into a training set with 100 observations from the active class and 1000 observations from the inactive class, and a testing set with the remaining observations. PSN² is used as the representative of our proposed methods, as the class 0 sample size is small for nonparametric methods. The average type I and type II errors over 1000 random splits are shown in Table 5. Compared with pen-log, nb and svm, PSN² performs much better in controlling the type I error.

Table 5: Average errors over 1000 random splits with standard errors in parentheses. $\alpha = 0.05$, $\delta_1 = 0.05$, $Q = 0.95$, and $\delta_3 = 0.1$.

	PSN ²	pen-log	nb	svm
type I	<u>.019 (.028)</u>	.162 (.060)	.054 (.035)	.275 (.189)
type II	.461 (.291)	.010 (.004)	.384 (.427)	.344 (.457)

3.2.2 EMAIL SPAM DATASET

Now, we consider an e-mail spam dataset available at <https://archive.ics.uci.edu/ml/datasets/Spambase>, which contains 4601 observations with 57 features, among which 2788 are class 0 (non-spam) and 1813 are class 1 (spam). We first standardize each feature and add 5000 synthetic features consisting of independent $\mathcal{N}(0, 1)$ variables to make the problem more challenging. The augmented data has $n = 4601$ observations with $d = 5057$ features. This augmented dataset is split into a training set with 1000 observations from each class and a testing set with the remaining observations. We use NSN² since the sample size is relatively large. The average type I and type II errors over 1000 random splits are shown in Table 6.

To evaluate the flexibility of NSN² in terms of prioritized error control, we also report the performance when the priority is switched to control the type II error below $\alpha = 0.05$. The results in Table 6 demonstrate that NSN² is able to control either type I or type II error depending on the specific need of the practitioner.

4. Discussion

The Neyman-Pearson classification framework is an important and interesting paradigm to explore beyond the Naive Bayes models considered in this work. For example, we can relax the independence assumption on PSN², and consider a general covariance matrix. Also, we can consider NP-type classifiers with decision boundaries involving feature interactions.

Table 6: Average errors over 1000 random splits with standard errors in parentheses. $\alpha = 0.05$, $\delta_1 = 0.05$, $Q = 0.95$, and $\delta_3 = 0.05$. The suffix after NSN² indicates the type of error it targets to control under α .

	NSN ² - R_0	NSN ² - R_1	pen-log	nb	svm
type I	<u>.019 (.007)</u>	.488 (.078)	.064 (.007)	.423 (.024)	.099 (.012)
type II	.439 (.057)	<u>.020 (.009)</u>	.133 (.015)	.058 (.010)	.174 (.016)

It is also worthwhile to study the non-probabilistic approaches under high-dimensional NP paradigm. Methods of potential interest include the k nearest neighbor (Weiss et al., 2010) and the centroid based classifiers (Tibshirani et al., 2002; Hall et al., 2010). However, the NP oracle inequalities are likely to be replaced by a new theoretical formulation for these methods.

A benefit of the present approach is that, for any given estimator \hat{r} , we have a uniform method to determine the proper threshold level in the plug-in classifiers. However, it would be interesting to develop new ways to estimate the threshold level C_α that is adaptive to the particular method used to approximate the density ratio r . Another future work is to study the theoretical properties of the permutation based choice of the threshold for the screening step.

Acknowledgements

The authors would like to thank the Editor and two anonymous referees for constructive comments. This research is partially supported by NSF CAREER grant DMS-1554804 (Feng), Lenfest Junior Faculty Development Grant from Columbia University (Feng), NSF grant DMS-1613338 (Tong), and NIH grant GM120507 (Tong).

Appendix A. Technical Lemmas and Proofs

Let $\text{Bin.cdf}_{n,p}(\cdot)$ denote the CDF of $\text{Bin}(n, p)$, and $\text{Beta.cdf}_{a,b}(\cdot)$ denote the CDF of $\text{Beta}(a, b)$. The following lemma proves a duality between the beta and binomial distributions.

Lemma A.1 (Beta-binomial duality). *For any $p \in [0, 1]$ and $k \in \{1, \dots, n\}$, it holds that*

$$1 - \text{Bin.cdf}_{n,p}(k-1) = \text{Beta.cdf}_{k,n+1-k}(p).$$

Proof of Lemma A.1. Let U_1, \dots, U_n be n i.i.d. $\text{Uniform}[0, 1]$. For any $p \in [0, 1]$, let $N_p = \sum_{i=1}^n \mathbb{1}\{U_i \leq p\}$ denote the number of U_i 's that are less or equal to p . Given

$$\mathbb{P}(\mathbb{1}\{U_i \leq p\} = 1) = \mathbb{P}(U_i \leq p) = p, \quad \mathbb{1}\{U_i \leq p\} \sim \text{Bern}(p) \quad \forall i,$$

we have $N_p \sim \text{Bin}(n, p)$, and therefore

$$\mathbb{P}(N_p \geq k) = 1 - \mathbb{P}(N_p \leq k-1) = 1 - \text{Bin.cdf}_{n,p}(k-1). \quad (\text{A.1})$$

On the other hand, let $U_{(k)}$ denote the k -th order statistic of $\{U_i\}_{i=1}^n$. It follows from the definition of order statistics that

$$\{N_p \geq k\} = \{\text{at least } k \text{ of } U_1, \dots, U_n \text{ are less or equal to } p\} = \{U_{(k)} \leq p\}. \quad (\text{A.2})$$

Combining (A.1) with (A.2) yields

$$1 - \text{Bin.cdf}_{n,p}(k-1) = \mathbb{P}(N_p \geq k) = \mathbb{P}(U_{(k)} \leq p) = \text{Beta.cdf}_{k,n+1-k}(p),$$

where the last equality follows from $U_{(k)} \sim \text{Beta}(k, n+1-k)$ ($k = 1, \dots, n$) as a direct implication of Rényi's representation. This completes the proof. \square

Lemma A.2. *Let Z be a random variable from CDF F . We have*

$$P_F\{F(Z) < \delta\} \leq \delta, \quad P_F\{F(Z) > \delta\} \geq 1 - \delta \quad \forall \delta \in [0, 1]. \quad (\text{A.3})$$

For continuous F , the inequality becomes equality as

$$P_F\{F(Z) < \delta\} = \delta, \quad P_F\{F(Z) > \delta\} = 1 - \delta \quad \forall \delta \in [0, 1]. \quad (\text{A.4})$$

Proof. Let $t_1 = \min\{t : F(t) \geq \delta\}$. Given the right continuity of F , it can be easily proved by contradiction that i) $F(t_1-) = F(t_1) = \delta$ if F is continuous at t_1 , and ii) $F(t_1-) < \delta \leq F(t_1)$ if F is discontinuous at t_1 . Thus,

$$P_F\{F(Z) < \delta\} = P_F(Z < t_1) = F(t_1-) \leq \delta.$$

Likewise, let $t_2 = \inf\{t : F(t) > \delta\}$. We have i) $F(t_2-) = F(t_2) = \delta$ if F is continuous at t_2 , and ii) $F(t_2-) < \delta \leq F(t_2)$ if F is discontinuous at t_2 . As a result,

$$P_F\{F(Z) > \delta\} = P_F\{Z \geq t_2\} = 1 - P_F\{Z < t_2\} \geq 1 - \delta.$$

This completes the proof. \square

Lemma A.3. *Let $S = \{Z_i\}_{i=1}^n$ be a set n i.i.d. random variables from distribution F , and let $Z_{(k)}$ denote its k -th order statistic ($k = 1, \dots, n$). For any $\delta \in (0, 1)$, the probability of a new, independent realization Z from F to be greater than $Z_{(k)}$ satisfies*

$$\begin{aligned} \mathbb{P}\{P_F(Z > Z_{(k)} | S) > \delta\} &\leq 1 - \text{Bin.cdf}_{n,1-\delta}(k-1), \\ \mathbb{P}\{P_F(Z > Z_{(k)} | S) < \delta\} &\geq 1 - \text{Bin.cdf}_{n,\delta}(n-k) = \text{Bin.cdf}_{n,1-\delta}(k-1). \end{aligned} \quad (\text{A.5}) \quad (\text{A.6})$$

The inequalities become equalities if F is continuous.

Proof of Lemma A.3. Rewrite the left-hand side of (A.5) as

$$\begin{aligned} \mathbb{P}\{P_F(Z > Z_{(k)} | S) > \delta\} &= \mathbb{P}\{1 - P_F(Z \leq Z_{(k)} | S) > \delta\} \\ &= \mathbb{P}\{1 - F(Z_{(k)}) > \delta\} = \mathbb{P}\{F(Z_{(k)}) < 1 - \delta\}. \end{aligned} \quad (\text{A.7})$$

To bound the probability of $\{F(Z_{(k)}) < 1 - \delta\}$, let $N_{1-\delta} = \sum_{i=1}^n \mathbb{1}_{\{F(Z_i) < 1-\delta\}}$ denote the number of $F(Z_i)$'s that are less than $1 - \delta$. It follows from $F(Z_{(1)}) \leq F(Z_{(2)}) \leq \dots \leq F(Z_{(n)})$ that

$$\begin{aligned} \{F(Z_{(k)}) < 1 - \delta\} &= \{F(Z_{(i)}) < 1 - \delta, i = 1, \dots, k\} = \{N_{1-\delta} \geq k\}, \\ \mathbb{P}\{F(Z_{(k)}) < 1 - \delta\} &= \mathbb{P}(N_{1-\delta} \geq k). \end{aligned} \quad (\text{A.8})$$

Let $\tau = P_F\{F(Z_1) < 1 - \delta\}$ denote the success probability of $N_{1-\delta}$ as a binomial. It follows from (A.3) that $\tau \leq 1 - \delta$. Given $\text{Bin.cdf}_{n,p}(k-1)$ being decreasing in p for any fixed n and k , we have

$$\mathbb{P}(N_{1-\delta} \geq k) = 1 - \text{Bin.cdf}_{n,\tau}(k-1) \leq 1 - \text{Bin.cdf}_{n,1-\delta}(k-1) \quad (\text{A.9})$$

as a result of The equalities hold for continuous F . Connecting (A.7), (A.8), and (A.9) together yields

$$\begin{aligned} \mathbb{P}\{P_F(Z > Z_{(k)} | S) > \delta\} &= \mathbb{P}\{F(Z_{(k)}) < 1 - \delta\} = \mathbb{P}(N_{1-\delta} \geq k) \\ &\leq 1 - \text{Bin.cdf}_{n,1-\delta}(k-1). \end{aligned}$$

Likewise, let $M_{1-\delta} = \sum_{i=1}^n \mathbb{1}_{\{F(Z_i) > 1-\delta\}}$ be a binomial random variable with size n and success rate $\tau' = P_F\{F(Z_i) > 1 - \delta\} \geq \delta$ that represents the number of $F(Z_i)$'s that are greater than $1 - \delta$. The left-hand side of (A.6) can be rewritten as

$$\begin{aligned} \mathbb{P}\{P_F(Z > Z_{(k)} | S) < \delta\} &= \mathbb{P}\{F(Z_{(k)}) > 1 - \delta\} \\ &= \mathbb{P}\{F(Z_{(i)}) > 1 - \delta, i = k, \dots, n\} = \mathbb{P}\{M_{1-\delta} \geq n + 1 - k\} \\ &= 1 - \mathbb{P}\{M_{1-\delta} \leq n - k\} = 1 - \text{Bin.cdf}_{n,\tau'}(n-k) \\ &\geq 1 - \text{Bin.cdf}_{n,\delta}(n-k). \end{aligned} \quad (\text{A.10})$$

This completes the proof. \square

Proof of Proposition 2.1. Letting $Z_i = \hat{r}_i$, $n = m_3$ in Lemma (A.3) yields

$$\mathbb{P}\{R_{t_0}(\hat{\phi}_k) > \delta\} \leq 1 - \text{Bin.cdf}_{m_3,1-\delta}(k-1).$$

This, together with Lemma A.1, completes the proof. \square

Lemma A.4. *For random variable $Z \sim \text{Beta}(a, b)$, and any $\epsilon > 0$, we have*

$$\mathbb{P}\{Z > (1 + \epsilon)\mathbb{E}Z\} < \mathbb{P}(|Z - \mathbb{E}Z| > \epsilon\mathbb{E}Z) < \frac{b\epsilon^{-2}}{(a+b+1)a}. \quad (\text{A.11})$$

Proof of Lemma A.4. By Chebyshev inequality,

$$\mathbb{P}(|Z - \mathbb{E}Z| > \epsilon\mathbb{E}Z) \leq \frac{\text{var}(Z)}{(\epsilon\mathbb{E}Z)^2} = \frac{ab}{(a+b)^2(a+b+1)} \left(\frac{\epsilon a}{a+b}\right)^{-2} = \frac{b\epsilon^{-2}}{(a+b+1)a}.$$

\square

Proof of Proposition 2.2. Let B be a realization from $\text{Beta}(k, m_3 + 1 - k)$. It follows from Proposition 2.1 that

$$\begin{aligned} \mathbb{P}\{R_0(\hat{\phi}_k) > g(\delta_3, m_3, k)\} &\leq \text{Beta.cdf}_{k, m_3+1-k}\{1 - g(\delta_3, m_3, k)\} \\ &= \mathbb{P}\{B \leq 1 - g(\delta_3, m_3, k)\} = \mathbb{P}\{1 - B \geq g(\delta_3, m_3, k)\} \end{aligned}$$

for any $k \in \{1, \dots, m_3\}$ and \hat{r} , with $1 - B \sim \text{Beta}(m_3 + 1 - k, k)$. Letting $a = m_3 + 1 - k$, $b = k$, and $\epsilon = k^{1/2}\{\delta_3(m_3 + 2)(m_3 + 1 - k)\}^{-1/2}$ in Lemma A.4 yields

$$\mathbb{P}\{R_0(\hat{\phi}_k) > g(\delta_3, m_3, k)\} \leq \delta_3,$$

where

$$g(\delta_3, m_3, k) = (1 + \epsilon) \left(\frac{m_3 + 1 - k}{m_3 + 1} \right) + \sqrt{\frac{k(m_3 + 1 - k)}{\delta_3(m_3 + 2)(m_3 + 1)^2}}.$$

This completes the proof. \square

Proof of Proposition 2.3. By some basic algebra we have

$$\begin{aligned} A_{\alpha, \delta_3}(m_3) - (1 - \alpha) &= \frac{-1 + 2\alpha + \sqrt{1 + 4\delta_3(1 - \alpha)\alpha(m_3 + 2)}}{2\{\delta_3(m_3 + 2) + 1\}} > 0, \\ A_{\alpha, \delta_3}(m_3) - 1 &= \frac{-1 - 2\delta_3(m_3 + 2)\alpha + \sqrt{1 + 4\delta_3(1 - \alpha)\alpha(m_3 + 2)}}{2\{\delta_3(m_3 + 2) + 1\}} < 0, \end{aligned}$$

and

$$\begin{aligned} g(\delta_3, m_3, k) &= \frac{m_3 + 1 - k}{m_3 + 1} + \sqrt{\frac{k(m_3 + 1 - k)}{\delta_3(m_3 + 2)(m_3 + 1)^2}} \leq \alpha \\ &\Leftrightarrow k - (1 - \alpha)(m_3 + 1) \geq 0, \\ &\Leftrightarrow \begin{cases} \{\delta_3(m_3 + 2) + 1\} \left(\frac{k}{m_3 + 1}\right)^2 - \{1 + 2\delta_3(m_3 + 2)(1 - \alpha)\} \left(\frac{k}{m_3 + 1}\right) \\ + \delta_3(m_3 + 2)(1 - \alpha)^2 \geq 0 \end{cases} \\ &\Leftrightarrow k \geq (m_3 + 1) \max\{1 - \alpha, A_{\alpha, \delta_3}(m_3)\} \\ &\Leftrightarrow k \geq (m_3 + 1) A_{\alpha, \delta_3}(m_3). \end{aligned}$$

Thus,

$$\begin{aligned} k_{\min}(\alpha, \delta_3, m_3) &= \lceil (m_3 + 1) A_{\alpha, \delta_3}(m_3) \rceil \\ &\in \lceil (m_3 + 1) A_{\alpha, \delta_3}(m_3), (m_3 + 1) A_{\alpha, \delta_3}(m_3) + 1 \rceil. \end{aligned}$$

Since $A_{\alpha, \delta_3}(m_3) \rightarrow 1 - \alpha$, as $m_3 \rightarrow \infty$, it follows from sandwich rule that

$$\lim_{m_3 \rightarrow \infty} \frac{k_{\min}(\alpha, \delta_3, m_3)}{m_3} = \lim_{m_3 \rightarrow \infty} A_{\alpha, \delta_3}(m_3) = 1 - \alpha.$$

We have $k_{\min}(\alpha, \delta_3, m_3) \in \mathcal{K}(\alpha, \delta_3, m_3)$ ($\Leftrightarrow k_{\min}(\alpha, \delta_3, m_3) \leq m_3$) as long as

$$(m_3 + 1) A_{\alpha, \delta_3}(m_3) + 1 \leq m_3 \quad \Leftrightarrow \quad (1 - \alpha \leq) \quad A_{\alpha, \delta_3}(m_3) \leq \frac{m_3 - 1}{m_3 + 1}. \quad (\text{A.12})$$

For any $\Delta \in (0, \alpha)$, a sufficient condition for (A.12) is

$$\frac{m_3 - 1}{m_3 + 1} \geq 1 - \Delta, \quad A_{\alpha, \delta_3}(m_3) \leq 1 - \Delta,$$

which can be further simplified as

$$m_3 \geq \frac{2}{\Delta} - 1, \quad m_3 \geq x^* - 2,$$

where

$$x^* = \frac{-2\Delta^2 - \alpha^2 + 2\alpha\Delta + \Delta + (1 - 2\alpha)\Delta + \alpha^2}{2(\alpha - \Delta)^2\delta_3} = \frac{\Delta(1 - \Delta)}{(\alpha - \Delta)^2\delta_3}$$

is the positive root of the quadratic equation

$$(\alpha - \Delta)^2\delta_3^2x^2 + \delta_3(2\Delta^2 + \alpha^2 - 2\alpha\Delta - \Delta)x - \Delta(1 - \Delta) = 0.$$

Thus, a sufficient condition for (A.12) is

$$m_3 \geq \max \left\{ \frac{\Delta(1 - \Delta)}{(\alpha - \Delta)^2\delta_3} - 2, \frac{2}{\Delta} - 1 \right\}.$$

Setting $\Delta = \alpha/2$ yields

$$\max \left\{ \frac{\Delta(1 - \Delta)}{(\alpha - \Delta)^2} - 2, \frac{2}{\Delta} - 1 \right\} = \max \left\{ \frac{2 - \alpha}{\alpha\delta_3} - 2, \frac{4}{\alpha} - 1 \right\} \leq \frac{4}{\alpha\delta_3}.$$

Therefore, $m_3 \geq 4/(\alpha\delta_3)$ guarantees (A.12) and $k_{\min}(\alpha, \delta_3, m_3) \in \mathcal{K}(\alpha, \delta_3, m_3)$. This completes the proof. \square

Proof of Lemma 2.1. Introduce shorthand notation let $A = A_{\alpha, \delta_3}(m_3)$ (defined in Proposition 2.3) and $\alpha_1 = (m_3 + 1 - k_{\min})/(m_3 + 1)$ for simplicity of exposition. For any $B_1, B_2 \in \mathbb{R}^+$, we have

$$\{|R_0(\hat{\phi}) - \alpha| > B_1 + B_2\} \subset \{|R_0(\hat{\phi}) - \alpha_1| > B_1\} \cup \{|\alpha_1 - \alpha| > B_2\},$$

and thus

$$\begin{aligned} &\mathbb{P}\{|R_0(\hat{\phi}) - \alpha| > B_1 + B_2 \mid \hat{r}\} \\ &\leq \mathbb{P}\{|R_0(\hat{\phi}) - \alpha_1| > B_1 \mid \hat{r}\} + \mathbb{P}\{|\alpha_1 - \alpha| > B_2 \mid \hat{r}\} \\ &\leq \frac{k_{\min}(m_3 + 1 - k_{\min})}{(m_3 + 2)(m_3 + 1)^2} B_1^{-2} + \mathbb{I}\{|\alpha_1 - \alpha| > B_2\}, \end{aligned} \quad (\text{A.13})$$

where the last inequality follows from applying Lemma A.4 to $R_0(\hat{\phi})$ which follows Beta($m_3 + 1 - k_{\min}, k_{\min}$) for $m_3 \geq 4/(\alpha\delta_3)$ and continuous $F_{\hat{r}}$ due to Lemma A.3. It follows from Proposition 2.3 that

$$|\alpha - \alpha_1| \leq A - (1 - \alpha) + \frac{1}{m_3 + 1}. \quad (\text{A.14})$$

Letting $B_1 = \sqrt{\frac{k_{\min}(m_3+1-k_{\min})}{(m_3+2)(m_3+1)^2\delta_4}}$ and $B_2 = A - (1 - \alpha) + \frac{1}{m_3+1}$ in (A.13) yields

$$\begin{aligned} & \mathbb{P}\{|R_0(\hat{\phi}) - \alpha| > \xi_{\alpha, \delta_3, m_3}(\delta_4) \mid \hat{r}\} \\ & \leq \delta_4 + \mathbb{P}\{|\alpha_1 - \alpha| > A - (1 - \alpha) + \frac{1}{m_3 + 1}\} \\ & = \delta_4 \end{aligned}$$

for any arbitrary \hat{r} . This, together with the independence between \mathcal{S}_3^0 and \hat{r} (as a function of $(\mathcal{S}_1^0, \mathcal{S}_1^1, \mathcal{S}_2^0, \mathcal{S}_2^1)$) yields

$$\mathbb{P}\{|R_0(\hat{\phi}) - \alpha| > \xi_{\alpha, \delta_3, m_3}(\delta_4)\} \leq \delta_4.$$

To establish an upper bound for $\xi_{\alpha, \delta_3, m_3}(\delta_4)$, note that

$$\begin{aligned} & \xi_{\alpha, \delta_3, m_3}(\delta_4) \\ & = \sqrt{\frac{k_{\min}(m_3+1-k_{\min})}{(m_3+2)(m_3+1)^2\delta_4}} + \frac{-1+2\alpha+\sqrt{1+4\delta_3(1-\alpha)\alpha(m_3+2)}}{2\{\delta_3(m_3+2)+1\}} + \frac{1}{m_3+1} \\ & \leq \sqrt{\frac{(m_3+1)^2/4}{(m_3+2)(m_3+1)^2\delta_4}} + \frac{1}{2\{\delta_3(m_3+2)+1\}} + \frac{\sqrt{1+\delta_3(m_3+2)}}{2\{\delta_3(m_3+2)+1\}} + \frac{1}{m_3+1} \\ & < \frac{1}{2\sqrt{m_3\delta_4}} + \frac{1}{2m_3\delta_3} + \frac{1}{2\sqrt{m_3\delta_3}} + \frac{1}{m_3}. \end{aligned}$$

When $m_3 \geq \max(\delta_3^{-2}, \delta_4^{-2})$, we have

$$\begin{aligned} \xi_{\alpha, \delta_3, m_3}(\delta_4) & < \frac{1}{2m_3^{1/4}} + \frac{1}{2m_3^{1/2}} + \frac{1}{2m_3^{3/4}} + \frac{1}{m_3} \\ & = \frac{1}{m_3^{1/4}} \left(1 + \frac{1}{2m_3^{1/4}} + \frac{1}{m_3^{3/4}} \right) < \frac{5/2}{m_3^{1/4}} = \left(\frac{2}{5} m_3^{1/4} \right)^{-1}. \end{aligned}$$

This completes the proof. \square

Proof of Proposition 2.4. Let $G^* = \{r < C_\alpha\}$ and $\widehat{G} = \{\hat{r} < \widehat{C}_\alpha\}$, the excess type II error can be decomposed as:

$$\begin{aligned} & P_1(\widehat{G}) - P_1(G^*) \\ & = \int_{\widehat{G}} dP_1 - \int_{G^*} dP_1 = \int_{\widehat{G}} \frac{p}{q} dP_0 - \int_{G^*} \frac{p}{q} dP_0 \\ & = \int_{\widehat{G}} (r - C_\alpha) dP_0 + C_\alpha P_0(\widehat{G}) - \int_{G^*} (r - C_\alpha) dP_0 - C_\alpha P_0(G^*) \\ & = \int_{\widehat{G} \setminus G^*} (r - C_\alpha) dP_0 - \int_{G^* \setminus \widehat{G}} (r - C_\alpha) dP_0 + C_\alpha \{P_0(\widehat{G}) - P_0(G^*)\} \\ & = \int_{\widehat{G} \setminus G^*} |r - C_\alpha| dP_0 + \int_{G^* \setminus \widehat{G}} |r - C_\alpha| dP_0 + C_\alpha \{R_0(\hat{\phi}^*) - R_0(\hat{\phi})\}. \end{aligned} \quad (\text{A.15})$$

It follows from Lemma 2.1 that when $m_3 \geq \max\{\frac{1}{\alpha\delta_3}, \delta_3^{-2}, \delta_4^{-2}, (\frac{2}{5}M_1\delta^{*2})^{-4}\}$,

$$\xi_{\alpha, \delta_3, m_3}(\delta_4) \leq \frac{5}{2} m_3^{-1/4} \leq M_1(\delta^*)^{2/3}, \quad \left\{ \frac{\xi_{\alpha, \delta_3, m_3}(\delta_4)}{M_1} \right\}^{1/2} \leq \delta^*.$$

Introduce shorthand notations $\Delta R_0 = |R_0(\hat{\phi}^*) - R_0(\hat{\phi})|$, $\mathcal{E}_0 = \{\Delta R_0 < \xi_{\alpha, \delta_3, m_3}(\delta_4)\}$, and $T = \|\hat{r} - r\|_\infty$. On the event \mathcal{E}_0 ,

$$\left(\frac{\Delta R_0}{M_1} \right)^{1/2} \leq \left\{ \frac{\xi_{\alpha, \delta_3, m_3}(\delta_4)}{M_1} \right\}^{1/2} \leq \delta^*.$$

By the detection condition, we have

$$\Delta R_0 \leq P_0\{C_\alpha < r(X) < C_\alpha + (\Delta R_0/M_1)^{1/2}\}.$$

Note that

$$\begin{aligned} & P_0\{r(X) \geq C_\alpha + (\Delta R_0/M_1)^{1/2}\} = R_0(\hat{\phi}^*) - P_0\{C_\alpha < r(X) < C_\alpha + (\Delta R_0/M_1)^{1/2}\} \\ & \leq R_0(\hat{\phi}^*) - \Delta R_0 \\ & \leq R_0(\hat{\phi}) = P_0\{\hat{r}(X) > \widehat{C}_\alpha\} \\ & \leq P_0\{r(X) + T \geq \widehat{C}_\alpha\} = P_0\{r(X) \geq \widehat{C}_\alpha - T\}. \end{aligned}$$

Thus, we have $\widehat{C}_\alpha \leq C_\alpha + (\Delta R_0/M_1)^{1/2} + T$, and

$$\begin{aligned} \widehat{G} \setminus G^* & = \{r \geq C_\alpha, \hat{r} < \widehat{C}_\alpha\} = \{r \geq C_\alpha, \hat{r} < C_\alpha + (\Delta R_0/M_1)^{1/2} + T\} \cap \{\hat{r} < \widehat{C}_\alpha\} \\ & = \{C_\alpha + (\Delta R_0/M_1)^{1/2} + 2T \geq r \geq C_\alpha, \hat{r} < C_\alpha + (\Delta R_0/M_1)^{1/2} + T\} \cap \{\hat{r} < \widehat{C}_\alpha\} \\ & \subset \{C_\alpha + (\Delta R_0/M_1)^{1/2} + 2T \geq r \geq C_\alpha\}. \end{aligned}$$

Therefore, the margin assumption implies

$$\begin{aligned} P_0(\widehat{G} \setminus G^*) & \leq P_0\{C_\alpha + (\Delta R_0/M_1)^{1/2} + 2T \geq r \geq C_\alpha\} \\ & \leq M_0\{(\Delta R_0/M_1)^{1/2} + 2T\}^{\bar{\gamma}}. \end{aligned}$$

Hence on the event \mathcal{E}_0 ,

$$\begin{aligned} \int_{\widehat{\mathcal{A}} \setminus G^*} |r - C_\alpha| dP_0 &\leq \{(\Delta R_0/M_1)^{1/2} + 2T\} P_0(\widehat{G} \setminus G^*) \\ &\leq M_0 \{(\Delta R_0/M_1)^{1/2} + 2T\}^{1+\bar{\gamma}}. \end{aligned}$$

We will bound $\int_{G^* \setminus \widehat{G}} |r - C_\alpha| dP_0$ on the event $\mathcal{E}_1 = \{R_0(\widehat{\phi}) \leq \alpha\}$. Note that

$$P_0(r \geq C_\alpha) = \alpha \geq R_0(\widehat{\phi}) = R_0(\hat{r} \geq \widehat{C}_\alpha) \geq P_0(r \geq \widehat{C}_\alpha + \|r - r\|_\infty) = P_0(r \geq \widehat{C}_\alpha + T).$$

The above chain implies that $\widehat{C}_\alpha \geq C_\alpha - T$. Therefore,

$$\begin{aligned} G^* \setminus \widehat{G} &= \{r < C_\alpha, \hat{r} \geq \widehat{C}_\alpha\} \\ &= \{r < C_\alpha, r \geq r - \hat{r} + \widehat{C}_\alpha\} \\ &\subset \{r < C_\alpha, r \geq \widehat{C}_\alpha - T\} \\ &\subset \{C_\alpha - 2T \leq r \leq C_\alpha\}. \end{aligned}$$

Hence on the event \mathcal{E}_1 ,

$$\int_{G^* \setminus \widehat{G}} |r - C_\alpha| dP_0 \leq 2T \cdot P_0(C_\alpha - 2T \leq r \leq C_\alpha) \leq M_0(2T)^{1+\bar{\gamma}},$$

where the last inequality follows from the margin assumption. Then it follows from (A.15) that on the event $\mathcal{E}_0 \cap \mathcal{E}_1$,

$$\begin{aligned} R_1(\widehat{\phi}) - R_1(\phi^*) &\leq M_0 \left[\left\{ \frac{|\Delta R|}{M_1} \right\}^{1/2} + 2T \right]^{1+\bar{\gamma}} + M_0(2T)^{1+\bar{\gamma}} + C_\alpha |R_0(\widehat{\phi}) - R_0(\phi^*)| \\ &\leq 2M_0 \left[\left\{ \frac{\xi_{\alpha, \delta_3, m_3}(\delta_4)}{M_1} \right\}^{1/2} + 2T \right]^{1+\bar{\gamma}} + C_\alpha \cdot \xi_{\alpha, \delta_3, m_3}(\delta_4). \end{aligned}$$

From Lemma 2.1, we know that the event \mathcal{E}_0 occurs with probability at least $1 - \delta_4$. By Proposition 2.2 and Proposition 2.3 we know event \mathcal{E}_1 occurs with probability at least $1 - \delta_3$, so $\mathcal{E}_0 \cap \mathcal{E}_1$ occurs with probability at least $1 - \delta_3 - \delta_4$. This completes the proof. \square

Proof of Proposition 2.5. Define event

$$\mathcal{E}_{\delta_1} = \bigcap_{j=1}^d \{ \|\widehat{F}_j^1 - F_j^1\|_\infty < \delta_1^1 \} \cap \{ \|\widehat{F}_j^0 - F_j^0\|_\infty < \delta_1^0 \},$$

where $\delta_1^1 = \sqrt{\frac{\log(4d/\delta_1)}{2m_1}}$ and $\delta_1^0 = \sqrt{\frac{\log(4d/\delta_1)}{2m_0}}$. On the event \mathcal{E}_{δ_1} , for any $j \in \mathcal{A}$,

$$\begin{aligned} \|\widehat{F}_j^0 - \widehat{F}_j^1\|_\infty &\geq \|\widehat{F}_j^0 - F_j^0\|_\infty - \|F_j^0 - \widehat{F}_j^0\|_\infty - \|F_j^1 - \widehat{F}_j^1\|_\infty \\ &\geq D - \|\widehat{F}_j^0 - F_j^0\|_\infty - \|F_j^1 - \widehat{F}_j^1\|_\infty \\ &> D - \delta_1^0 - \delta_1^1. \end{aligned}$$

For any $j \notin \mathcal{A}$,

$$\begin{aligned} \|\widehat{F}_j^0 - \widehat{F}_j^1\|_\infty &\leq \|\widehat{F}_j^0 - \widehat{F}_j^0\|_\infty + \|\widehat{F}_j^0 - F_j^0\|_\infty + \|F_j^1 - \widehat{F}_j^1\|_\infty \\ &= \|\widehat{F}_j^0 - F_j^0\|_\infty + \|F_j^1 - \widehat{F}_j^1\|_\infty \\ &< \delta_1^0 + \delta_1^1. \end{aligned}$$

Since $n_1 \geq 8D^{-2} \log(4d/\delta_1)$ and $m_1 \geq 8D^{-2} \log(4d/\delta_1)$, $\delta_1^0 + \delta_1^1 \leq D - \delta_1^0 - \delta_1^1$. As a result, on the event \mathcal{E}_{δ_1} , any $\tau \in [\delta_1^0 + \delta_1^1, D - \delta_1^0 - \delta_1^1]$ would lead to $\widehat{\mathcal{A}}_\tau = \mathcal{A}$. Therefore,

$$\begin{aligned} \mathbb{P}(\widehat{\mathcal{A}}_\tau = \mathcal{A}) &\geq \mathbb{P}(\mathcal{E}_{\delta_1}) \\ &\geq 1 - \sum_{j=1}^d \left\{ \mathbb{P}(\|\widehat{F}_j^1 - F_j^1\|_\infty \geq \delta_1^1) + \mathbb{P}(\|\widehat{F}_j^0 - F_j^0\|_\infty \geq \delta_1^0) \right\} \\ &\geq 1 - \delta_1, \end{aligned}$$

where the last inequality follows from applying Lemma A.5 to F_j^0 and F_j^1 for $j = 1, \dots, d$. This completes the proof. \square

Proof of Proposition 2.6. Define event

$$\mathcal{E} = \bigcap_{j \in \mathcal{A}} \{ \|\log \hat{r}_j - \log P_j\|_\infty < B_j^1 \} \cap \{ \|\log \hat{q}_j - \log Q_j\|_\infty < B_j^0 \},$$

where

$$B_j^1 = \frac{C_j^1 \sqrt{\frac{\log(2m_2s/\delta_2)}{n_2 \hat{n}_1}}}{n_2 \hat{n}_1}, \quad B_j^0 = \frac{C_j^0 \sqrt{\frac{\log(2m_2s/\delta_2)}{m_2 \hat{m}_0}}}{m_2 \hat{m}_0}.$$

Let $B^1 = \sup_{j \in \mathcal{A}} B_j^1$ and $B^0 = \sup_{j \in \mathcal{A}} B_j^0$, we have $B \geq s(B^0 + B^1)$. On the event $\{\widehat{\mathcal{A}}_\tau = \mathcal{A}\} \cap \mathcal{E}$, we have

$$\log \hat{r}_N^S(x) = \sum_{j \in \widehat{\mathcal{A}}} \log \frac{\hat{r}_j(x_j)}{\hat{q}_j(x_j)} = \sum_{j \in \mathcal{A}} \log \hat{r}_j(x_j) - \sum_{j \in \mathcal{A}} \log \hat{q}_j(x_j).$$

Therefore,

$$\begin{aligned} \|\log \hat{r}_N^S - \log r\|_\infty &= \left\| \sum_{j \in \mathcal{A}} \log \hat{r}_j - \sum_{j \in \mathcal{A}} \log \hat{q}_j - \sum_{j \in \mathcal{A}} \log P_j + \sum_{j \in \mathcal{A}} \log Q_j \right\|_\infty \\ &\leq \sum_{j \in \mathcal{A}} (\|\log \hat{r}_j - \log P_j\|_\infty + \|\log \hat{q}_j - \log Q_j\|_\infty) \\ &\leq \sum_{j \in \mathcal{A}} (B^1 + B^0) \leq B. \end{aligned}$$

On the event $\{\widehat{\mathcal{A}}_\tau = \mathcal{A}\} \cap \mathcal{E}$, it follows from Lagrange's mean value theorem that for any x , there exists some w_x between $\log \hat{r}_N^S(x)$ and $\log r(x)$ such that

$$\begin{aligned} |\hat{r}_N^S(x) - r(x)| &= |e^{\log \hat{r}_N^S(x)} - e^{\log r(x)}| = e^{w_x} |\log \hat{r}_N^S(x) - \log r(x)| \\ &\leq e^{\|\log r\|_\infty + B} B = B e^B \|r\|_\infty = T, \end{aligned}$$

where the last inequality follows from the fact that

$$w_x \leq \max(\log r(x), \log \hat{r}_N^S(x)) \leq \max(\|\log r\|_\infty, \|\log \hat{r}_N^S\|_\infty) \leq \|\log r\|_\infty + B.$$

Thus, $\|\hat{r}_N^S - r\|_\infty \leq T$, and we have

$$\begin{aligned} \mathbb{P}(\|\hat{r}_N^S - r\|_\infty \leq T) &\geq \mathbb{P}(\{\hat{\mathcal{A}}_T = \mathcal{A}\} \cap \mathcal{E}) \geq \mathbb{P}(\hat{\mathcal{A}}_T = \mathcal{A}) + \mathbb{P}(\mathcal{E}) - 1 \\ &= \mathbb{P}(\hat{\mathcal{A}}_T = \mathcal{A}) - \mathbb{P}(\mathcal{E}^c). \end{aligned} \quad (\text{A.16})$$

By Proposition 2.5, we have

$$\mathbb{P}(\hat{\mathcal{A}}_T = \mathcal{A}) \geq 1 - \delta_1. \quad (\text{A.17})$$

Also, it follows from Lemma A.6 that

$$\mathbb{P}(\|\log \hat{p}_j - \log p_j\|_\infty > B_j^1) \vee \mathbb{P}(\|\log \hat{q}_j - \log q_j\|_\infty > B_j^0) \leq \delta_2/(2s).$$

Therefore,

$$\mathbb{P}(\mathcal{E}^c) \leq (2s)\delta_2/(2s) = \delta_2. \quad (\text{A.18})$$

Plugging (A.17) and (A.18) back to (A.16) yields (2.16). Moreover, because $s \leq n_2 \wedge m_2$, it follows from Lemma A.6 that there exists some $C_2 > 0$, such that

$$B \leq C_2 s \left\{ \left(\frac{\log n_2}{n_2} \right)^{\frac{\beta}{2\beta+1}} + \left(\frac{\log m_2}{m_2} \right)^{\frac{\beta}{2\beta+1}} \right\}.$$

Moreover, since $s \leq (n_2 \wedge m_2)^{\frac{\beta}{2\beta+1}}$, the above bound implies that B is bounded from above by some absolute constant. Also note that $\|r\|_\infty$ is bounded from above, so there exists an absolute constant $C_2 > 0$, such that

$$T = Be^B \|r\|_\infty \leq C_2 s \left\{ \left(\frac{\log n_2}{n_2} \right)^{\frac{\beta}{2\beta+1}} + \left(\frac{\log m_2}{m_2} \right)^{\frac{\beta}{2\beta+1}} \right\}.$$

This completes the proof. \square

Proof of Theorem 2.1. Combining Propositions 2.2, 2.3, 2.4 and 2.6,

$$\mathbb{P}\left(R_0(\hat{\phi}_{\text{NSN}^2}) \leq \alpha, R_1(\hat{\phi}_{\text{NSN}^2}) \leq R_1(\phi^*) + W\right) \geq 1 - \delta_1 - \delta_2 - \delta_3 - \delta_4,$$

where

$$\begin{aligned} W = & 2M_0 \left[\left(\frac{2}{5} m_3^{1/4} M_1 \right)^{-1/2} + 2C_2 s \left\{ \left(\frac{\log n_2}{n_2} \right)^{\frac{\beta}{2\beta+1}} + \left(\frac{\log m_2}{m_2} \right)^{\frac{\beta}{2\beta+1}} \right\} \right]^{1+\gamma} \\ & + C_\alpha \left(\frac{2}{5} m_3^{1/4} \right)^{-1}. \end{aligned}$$

This completes the proof. \square

Lemma A.5 (Dvoretzky-Kiefer-Wolfowitz inequality (Dvoretzky et al., 1956)). Let X_1, X_2, \dots, X_n be real-valued i.i.d. random variables with cdf $F(\cdot)$, and let $\hat{F}_n(x) = n^{-1} \sum_{i=1}^n \mathbb{1}(X_i \leq x)$. For any $t > 0$, it holds that

$$\mathbb{P}(\|\hat{F}_n - F\|_\infty \geq t) \leq 2e^{-2nt^2}.$$

Or, for any given $\delta \in (0, 1)$,

$$\mathbb{P}(\|\hat{F}_n - F\|_\infty \geq \sqrt{\frac{\log(2/\delta)}{2n}}) \leq \delta. \quad (\text{A.19})$$

Lemma A.6. Given a density function $p \in \mathcal{P}_\Sigma(\beta, L, [-1, 1])$, construct its kernel estimate $\hat{p}(x) = (nh)^{-1} \sum_{i=1}^n K\left(\frac{X_i - x}{h}\right)$ from i.i.d. sample $\{X_i\}_{i=1}^n$, where the kernel K is β -valid and L -Lipschitz, and the bandwidth $h = (\log n/n)^{\frac{1}{2\beta+1}}$. For any $\delta \in (0, 1)$, as long as the sample size n is such that $\sqrt{\frac{\log(n/\delta)}{nh}} < \min(1, \underline{\mu}/C)$, where $C = \sqrt{48c_1} + 32c_2 + 2Lc_3 + L + L + \tilde{C} \sum_{1 \leq |\beta| \leq \frac{1}{h}} \frac{1}{h}$, in which $c_1 = \|p\|_\infty \|K\|_2^2$, $c_2 = \|K\|_\infty + \|p\|_\infty + \int |K| |t|^\beta dt$, $c_3 = \int |K| |t|^\beta dt$, and \tilde{C} is such that $\tilde{C} \geq \sup_{1 \leq |\beta| \leq \frac{1}{h}} \sup_{x \in [-1, 1]} |p^{(\beta)}(x)|$, and $\underline{\mu}(> 0)$ is a lower bound of p , we have

$$\mathbb{P}(\|\log \hat{p} - \log p\|_\infty \geq U) \leq \delta, \quad (\text{A.20})$$

where $U = \frac{C\sqrt{\frac{\log(n/\delta)}{nh}}}{\underline{\mu} - C\sqrt{\frac{\log(n/\delta)}{nh}}}$. When $n \geq 1/\delta$, we have $U \leq C_1 (\log n/n)^{\frac{\beta}{2\beta+1}}$ for some absolute constant C_1 .

Proof. Let $\mathcal{E}_1 = \{\|\hat{p} - p\|_\infty \leq C\sqrt{\frac{\log(n/\delta)}{nh}}\}$. On the event \mathcal{E}_1 , since $\sqrt{\frac{\log(n/\delta)}{nh}} < \min(1, \underline{\mu}/C)$, we have

$$\min(p(x_0), \hat{p}(x_0)) \geq \min(p(x_0), p(x_0) - \|\hat{p} - p\|_\infty) \geq \underline{\mu} - \|\hat{p} - p\|_\infty > 0.$$

It then follows from Lagrange's mean value theorem that for any fixed x_0 , there exists some w_{x_0} between $\hat{p}(x_0)$ and $p(x_0)$,

$$\begin{aligned} |\log \hat{p}(x_0) - \log p(x_0)| &= w_{x_0}^{-1} |\hat{p}(x_0) - p(x_0)| \\ &\leq [\min\{\hat{p}(x_0), p(x_0)\}]^{-1} |\hat{p}(x_0) - p(x_0)| \leq \frac{\|\hat{p} - p\|_\infty}{\underline{\mu} - \|\hat{p} - p\|_\infty}. \end{aligned}$$

As a result, it holds on event \mathcal{E}_1 that

$$\|\log \hat{p} - \log p\|_\infty \leq \frac{C\sqrt{\frac{\log(n/\delta)}{nh}}}{\underline{\mu} - C\sqrt{\frac{\log(n/\delta)}{nh}}} = U,$$

and

$$\mathbb{P}(\|\log \hat{p} - \log p\|_\infty \leq U) \geq \mathbb{P}(\|\hat{p} - p\|_\infty \leq C\sqrt{\frac{\log(n/\delta)}{nh}}) \geq 1 - \delta,$$

where the last inequality follows from Lemma A.1 in Tong (2013) (the special case of $d = 1$). Finally when $n \geq 1/\delta$, we have $U = \frac{C\sqrt{\frac{\log(n/\delta)}{nh}}}{\underline{\mu} - C\sqrt{\frac{\log(n/\delta)}{nh}}} \leq C_1 (\log n/n)^{\frac{\beta}{2\beta+1}}$ for some absolute constant C_1 . This completes the proof. \square

Appendix B. About detection condition and Assumption 3

We show that it is possible for densities satisfying Assumption 3 to violate a generalized version of the detection condition defined in Definition 2.3. While the generalized detection condition applies to general (P, f, C^*) as the original one, we narrow its definition to (P_0, r, C_α) which we actually use in the main text.

Definition B.1 (Generalized detection condition). *Let $u(\cdot)$ be a strictly increasing differentiable function on \mathbb{R}^+ with $\lim_{x \rightarrow 0+} u(x) = 0$, a function $r(\cdot)$ is said to satisfy the generalized detection condition with respect to P_0 and $u(\cdot)$ at level (C_α, δ^*) if for any $\delta \in (0, \delta^*)$,*

$$P_0\{C_\alpha \leq r(X) \leq C_\alpha + \delta\} \geq u(\delta). \quad (\text{B.1})$$

The following conditions suffice to make (B.1) fail

$$P_0\{C_\alpha \leq r(X) \leq C_\alpha + k^{-1}\} < u(k^{-1}), \quad k = 1, 2, \dots \quad (\text{B.2})$$

A 1-dimensional toy example that satisfies Assumption 3 and (B.2) (thus violating the generalized detection condition) is given as follows. Assume P_0 and P_1 have the same support $[-1, 1]$. Given $u(\cdot)$ as a strictly increasing differentiable function on \mathbb{R}^+ with $\lim_{x \rightarrow 0+} u(x) = 0$, let $q(x) = \alpha$ for all $x \in [0, 1]$, and set $p(x)$ accordingly such that

$$r(x) = \frac{p(x)}{q(x)} = \begin{cases} 2u^{-1}(1) + 2u^{-1}(\alpha x), & x \in (0, 1], \\ 2u^{-1}(1), & x = 0, \\ 2u^{-1}(1) - v(x), & x \in [-1, 0), \end{cases} \quad (\text{B.3})$$

where $v(\cdot)$ is some positive differentiable function that makes $r(\cdot)$ differentiable at $x = 0$. It follows from (B.3) that $\{x \in [-1, 1] : r(x) \geq 2u^{-1}(1)\} = [0, 1]$, and identity

$$P_0\{r(X) \geq 2u^{-1}(1)\} = \int_{\{x \in [-1, 1] : r(x) \geq 2u^{-1}(1)\}} q(x) dx = \int_{[0, 1]} q(x) dx = \alpha$$

implies $C_\alpha = 2u^{-1}(1)$. As a result, for any $k \in \{1, 2, \dots\}$ we have

$$\{C_\alpha \leq r(X) \leq C_\alpha + k^{-1}\} = \{X \in [0, 1], 2u^{-1}(\alpha X) \leq k^{-1}\} = \{X \in [0, \alpha^{-1}u(0.5k^{-1})]\},$$

and

$$\begin{aligned} P_0\{C_\alpha \leq r(X) \leq C_\alpha + k^{-1}\} &= P_0\{X \in [0, \alpha^{-1}u(0.5k^{-1})]\} = \int_0^{\alpha^{-1}u(0.5k^{-1})} q(x) dx \\ &= \alpha \cdot \alpha^{-1}u(0.5k^{-1}) = u(0.5k^{-1}) < u(k^{-1}) \end{aligned}$$

satisfies (B.2). Note that the above construction makes no assumption about the behavior of $q(\cdot)$ and $p(\cdot)$ on $[-1, 0)$ except the normalization constraints $\int_{[-1, 1]} p dx = \int_{[-1, 1]} q dx = 1$ and $r(\cdot)$ being differentiable on $[-1, 1]$. Thus, there exist p, q , and r that satisfy Assumption 3.

Appendix C. An alternative threshold estimate

This part contains an alternative estimate of threshold C_α that guarantees type I error bound. Based on Chernoff inequality, the following Proposition gives an alternative version of Proposition 2.2. First, we introduce two technical lemmas.

Lemma C.1. *If $G_k \sim \text{Gamma}(k, 1)$, $k > 0$, then for any $\tau \in (0, k)$, we have*

$$\mathbf{P}\{G_k \geq k + \tau\} \leq e^{-\tau^2/(4k)}, \quad \mathbf{P}\{G_k \leq k - \tau\} \leq e^{-\tau^2/(2k)} \leq e^{-\tau^2/(4k)}.$$

Proof of Lemma C.1. For any $\epsilon \in (0, 1)$ and $t \in (0, 1)$, it follows from Chernoff inequality that

$$\mathbf{P}\{G_k \geq (1 + \epsilon)k\} = \mathbf{P}\{e^{G_k} \geq e^{t(1+\epsilon)k}\} \leq \frac{\mathbf{E}(e^{tG_k})}{e^{t(1+\epsilon)k}} = (1 - t)^{-k} e^{-t(1+\epsilon)k}. \quad (\text{C.1})$$

Letting $t = \arg \min_{x \in (0, 1)} (1 - x)^{-k} e^{-x(1+\epsilon)k} = \epsilon/(1 + \epsilon)$ in (C.1) yields

$$\mathbf{P}\{G_k \geq (1 + \epsilon)k\} \leq (1 + \epsilon)^k e^{-\epsilon k} = e^{k(\log(1+\epsilon) - \epsilon)} \leq e^{-k\epsilon^2/4},$$

Likewise, for any $\epsilon \in (0, 1)$ and $s < 0$,

$$\mathbf{P}\{G_k \leq (1 - \epsilon)k\} = \mathbf{P}\{e^{sG_k} \geq e^{s(1-\epsilon)k}\} = (1 - s)^{-k} e^{-s(1-\epsilon)k}. \quad (\text{C.2})$$

Letting $s = \arg \min_{x < 0} (1 - x)^{-k} e^{-x(1-\epsilon)k} = -\epsilon/(1 - \epsilon)$ in (C.2) yields

$$\mathbf{P}\{G_k \leq (1 - \epsilon)k\} \leq (1 - \epsilon)^k e^{\epsilon k} = e^{k(\log(1-\epsilon) + \epsilon)} \leq e^{-k\epsilon^2/2},$$

where the last inequality follows from Taylor expansion

$$\log(1 - \epsilon) + \epsilon = \sum_{i=1}^{\infty} \frac{\epsilon^i}{i} - \epsilon = \sum_{i=2}^{\infty} \frac{\epsilon^i}{i} > \frac{\epsilon^2}{2}, \quad \forall 0 < \epsilon < 1.$$

Take $\epsilon = \tau/k$, the conclusion of the lemma follows. \square

Lemma C.2. *Let $B \sim \text{Beta}(a, b)$, and $\mu = \mathbf{E}(B) = a/(a + b)$. For any $t \in (0, 1 - \mu)$,*

$$\mathbf{P}\{B > \mu + t\} \leq 2 \exp \left[-4^{-1} \left\{ \frac{(a + b)t}{\sqrt{b(\mu + t) + \sqrt{a(1 - \mu - t)}}} \right\}^2 \right].$$

Proof of Lemma C.2. By properties of beta distribution, we can represent B as

$$B = \frac{G_a}{G_a + G_b}, \quad \text{where } G_a \sim \Gamma(a, 1), G_b \sim \Gamma(b, 1) \text{ are independent.}$$

For any $t > 0$ and constant C such that $a(1 - \mu - t) < C < b(\mu + t)$, we have

$$\begin{aligned} \mathbb{P}(B \leq \mu + t) &= \mathbb{P}\{(1 - \mu - t)G_a \leq (\mu + t)G_b\} \geq \mathbb{P}\{(1 - \mu - t)G_a \leq C \leq (\mu + t)G_b\} \\ &= \mathbb{P}\{(1 - \mu - t)G_a \leq C\} \mathbb{P}\{C \leq (\mu + t)G_b\} \\ &= \mathbb{P}\left(G_a \leq \frac{C}{1 - \mu - t}\right) \mathbb{P}\left(G_b \geq \frac{C}{\mu + t}\right) \\ &= \left\{1 - \mathbb{P}\left(G_a > \frac{C}{1 - \mu - t}\right)\right\} \left\{1 - \mathbb{P}\left(G_b > \frac{C}{\mu + t}\right)\right\} \\ &\geq 1 - \mathbb{P}\left(G_a > \frac{C}{1 - \mu - t}\right) - \mathbb{P}\left(G_b > \frac{C}{\mu + t}\right), \end{aligned} \quad (\text{C.3})$$

where by Lemma C.1

$$\begin{aligned} \mathbb{P}\left(G_a > \frac{C}{1 - \mu - t}\right) &\leq \mathbb{P}\left\{G_a > a + \left(\frac{C}{1 - \mu - t} - a\right)\right\} \leq e^{-\left(\frac{C}{1 - \mu - t} - a\right)^2 (4a)^{-1}}, \\ \mathbb{P}\left(G_b < \frac{C}{\mu + t}\right) &\leq \mathbb{P}\left\{G_b < b - \left(b - \frac{C}{\mu + t}\right)\right\} \leq e^{-\left(b - \frac{C}{\mu + t}\right)^2 (4b)^{-1}}. \end{aligned} \quad (\text{C.4})$$

Letting

$$C = \frac{(1 - \mu - t)(\mu + t)(a\sqrt{b} + \sqrt{ab})}{\sqrt{b}(\mu + t) + \sqrt{a}(1 - \mu - t)}$$

in (C.3) such that the two exponents in (C.4) equal

$$\left(\frac{C}{1 - \mu - t} - a\right)^2 (4a)^{-1} = \left(b - \frac{C}{\mu + t}\right)^2 (4b)^{-1} = 4^{-1} \left\{ \frac{(a + b)t}{\sqrt{b}(\mu + t) + \sqrt{a}(1 - \mu - t)} \right\}^2$$

yields

$$\begin{aligned} \mathbb{P}(B > \mu + t) &= 1 - \mathbb{P}(B \leq \mu + t) \leq \mathbb{P}\left(G_a > \frac{C}{1 - \mu - t}\right) + \mathbb{P}\left(G_b > \frac{C}{\mu + t}\right) \\ &\leq e^{-\left(\frac{C}{1 - \mu - t} - a\right)^2 (4a)^{-1}} + e^{-\left(b - \frac{C}{\mu + t}\right)^2 (4b)^{-1}} \\ &= 2 \exp \left[-4^{-1} \left\{ \frac{(a + b)t}{\sqrt{b}(\mu + t) + \sqrt{a}(1 - \mu - t)} \right\}^2 \right]. \end{aligned}$$

This completes the proof. \square

Proposition C.1. Let $\hat{r}(\cdot)$ be any estimate of the density ratio function. For any $\delta_3 \in (0, 1)$ and $k \in \{1, \dots, m_3\}$, the type I error of classifier $\hat{\phi}_k$ defined in (2.1) satisfies

$$\mathbb{P}\left\{R_0(\hat{\phi}_k) > h(\delta_3, m_3, k)\right\} \leq \delta_3,$$

where

$$h(\delta_3, m_3, k) = \frac{m_3 + 1 - k + 2\sqrt{\log(2/\delta_3)}\sqrt{m_3 - k + 1}}{m_3 + 1 + 2\sqrt{\log(2/\delta_3)}\left(\sqrt{m_3 - k + 1} - \sqrt{k}\right)}.$$

Proof. Let B be a realization from $\text{Beta}(k, m_3 + 1 - k)$. It follows from Proposition 2.1 that

$$\begin{aligned} \mathbb{P}\{R_0(\hat{\phi}_k) > h(\delta_3, m_3, k)\} &\leq \text{Beta.cdf}_{k, m_3 + 1 - k}\{1 - h(\delta_3, m_3, k)\} \\ &= \mathbb{P}\{B \leq 1 - h(\delta_3, m_3, k)\} = \mathbb{P}\{1 - B \geq h(\delta_3, m_3, k)\} \end{aligned}$$

for any $k \in \{1, \dots, m_3\}$ and \hat{r} , with $1 - B \sim \text{Beta}(m_3 + 1 - k, k)$. Letting $a = m_3 + 1 - k$, $b = k$, and

$$t = \frac{2\sqrt{\log(2/\delta_3)}\left\{(m_3 + 1 - k)\sqrt{k} + k\sqrt{m_3 + 1 - k}\right\}}{(m_3 + 1)\left\{m_3 + 1 + 2\sqrt{\log(2/\delta_3)}\left(\sqrt{m_3 + 1 - k} - \sqrt{k}\right)\right\}}.$$

in Lemma C.2 yields

$$\mathbb{P}\left\{R_0(\hat{\phi}_k) > h(\delta_3, m_3, k)\right\} \leq \delta_3.$$

This completes the proof. \square

Proposition C.1 implies that $h(\delta_3, m_3, k) \leq \alpha$ is a sufficient condition for the classifier $\hat{\phi}_k$ (defined in (2.2)) to satisfy NP Oracle Inequality (1) ($k = 1, \dots, m_3$). Let $\mathcal{K}_{\text{chem}} = \{k \in \{1, \dots, m_3\} : h(\delta_3, m_3, k) \leq \alpha\}$. Similar to Proposition 2.3 we can prove $\mathcal{K}_{\text{chem}}$ to be non-empty as long as m_3 is greater than some threshold.

Numerical investigation shows that for most combinations of (α, δ_3, m_3) with non-empty \mathcal{K} and $\mathcal{K}_{\text{chem}}$, $k_{\min} = \min_k \mathcal{K}$ as defined in (2.7) is better than $k_{\text{chem}} = \min_k \mathcal{K}_{\text{chem}}$ in the sense that $\hat{\phi}_{k_{\min}}$ has a lower type II error than $\hat{\phi}_{k_{\text{chem}}}$ as a result of $k_{\min} < k_{\text{chem}}$. Specifically, for each $\delta_3 \in \{0.01 \cdot \iota\}_{\iota=1}^{10}$, the number of $\{k_{\text{chem}} < k_{\min}\}$ out of 100 combinations of $(\alpha, m_3) \in \{0.01 \cdot \iota\}_{\iota=1}^{10} \times \{100 \cdot \iota\}_{\iota=1}^{10}$ is reported as follows. Only when δ_3 gets very close to 0 is k_{chem} preferred to k_{\min} .

δ_3	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
$\#\{k_{\text{chem}} < k_{\min}\}$	83	70	49	4	0	0	0	0	0	0

Appendix D. An example violating detection condition in Tong (2013)

Definition D.1. A function $f(\cdot)$ is said to satisfy 'detection condition Tong (2013)' of order γ with respect to P (i.e., $X \sim P$) at level (C^*, δ^*) if there exists a positive constant M , such that for any $\delta \in (0, \delta^*)$,

$$P\{C^* \leq f(X) \leq C^* + \delta\} \geq M\delta^\gamma, \quad P\{C^* - \delta \leq f(X) \leq C^*\} \geq M\delta^\gamma.$$

Our target is to construct an $f(\cdot) = p(\cdot)/q(\cdot)$ function, such that

1. there exists some positive constant M_1 such that

$$P_0\{C^* \leq f(X) \leq C^* + \delta\} \geq M_1\delta^\gamma$$

for any $\delta \in (0, \delta^*)$;

2. there exists no M_2 such that

$$P_0\{C^* - \delta \leq f(X) \leq C^*\} \geq M_2 \delta^2$$

for any $\delta \in (0, \delta^*)$.

Let

$$q(x) = \begin{cases} \frac{5}{8}, & x \in [-1, 0), \\ \frac{5}{8} - \frac{x}{2}, & x \in [0, 1], \end{cases}$$

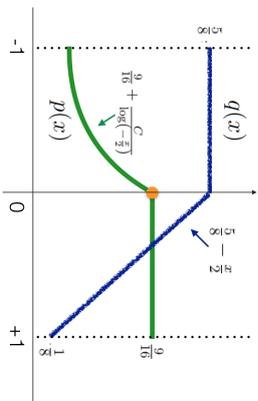
$$p(x) = \begin{cases} \frac{9}{16} + \frac{C}{\log(-\frac{x}{2})}, & x \in [-1, 0), \\ \frac{9}{16}, & x \in [0, 1]. \end{cases}$$

where

$$C = -\frac{1}{8} \left(\int_{-1}^0 \frac{1}{\log(-\frac{x}{2})} dx \right)^{-1} \approx 0.165.$$

such that

$$\int_{-1}^1 p(x) dx = \int_{-1}^1 \left[\frac{9}{16} + I_{(x \in [-1, 0])} \cdot \frac{C}{\log(-\frac{x}{2})} \right] dx = 1.$$



$$f(x) = \frac{p(x)}{q(x)} = \begin{cases} \frac{9}{10} + \frac{8C}{5 \log(-\frac{x}{2})}, & x \in [-1, 0) \\ \frac{9}{10 - 8x}, & x \in [0, 1] \end{cases}, \tag{D.1}$$

We have

Let $C^* = \frac{9}{10}$ and we have $\{x : f(x) \leq C^*\} \subset [-1, 0]$ and thus

$$P_0(C^* - \delta \leq f(X) \leq C^*) = P_0(-2e^{-\frac{8C}{5\delta}} \leq X \leq 0) = \frac{5}{8} \cdot 2e^{-\frac{8C}{5\delta}} = \frac{5e^{-\frac{8C}{5\delta}}}{4}.$$

For any constant γ , we have

$$\lim_{\delta \rightarrow 0^+} \frac{e^{-\frac{8C}{5\delta}}}{\delta^\gamma} = \lim_{\kappa = \delta^{-1} \rightarrow \infty} e^{-\frac{8C}{5} \kappa} \kappa^\gamma = 0.$$

Therefore, no such M_2 could possibly satisfy (b).

On the other hand, assume $\delta < 1$ without loss of generality. It follows from $\{x : f(x) \geq C^*\} \subset [0, 1]$ and $\frac{25\delta}{18+20\delta} \geq \frac{\delta}{2}$ that

$$\begin{aligned} P_0(C^* \leq f(X) \leq C^* + \delta) &= P_0\left(C^* \leq \frac{9}{10 - 8X} \leq C^* + \delta\right) \\ &= P_0\left(0 \leq X \leq \frac{25\delta}{18 + 20\delta}\right) \geq P_0\left(0 \leq X \leq \frac{\delta}{2}\right) = \int_0^{\delta/2} \left(\frac{5-x}{8} - \frac{x}{2}\right) dx \\ &= \left(\frac{5x}{8} - \frac{x^2}{4}\right) \Big|_0^{\delta/2} = \frac{5\delta}{16} - \frac{\delta^2}{16} \geq \frac{5\delta}{16} - \frac{\delta}{16} = \frac{\delta}{4}, \end{aligned}$$

satisfying (a) for constant $M = 1/4$ and $\bar{\gamma} = 1$.

This completes the construction of the density functions p and q such that the current detection condition is met but that in Tong (2013) is violated.

References

- ACKERMANN, M. and STRIMMER, K. (2009). A general modular framework for gene set enrichment analysis. *BMC Bioinformatics*, **10** 1471–2105.
- ALDERTON, G. K. (2014). Breast cancer: Breast cancer classification. *Nature Reviews Cancer*, **14** 155–155.
- AUDIBERT, J. and TSYBAKOV, A. (2007). Fast learning rates for plug-in classifiers under the margin condition. *The Annals of Statistics*, **35** 608–633.
- BICKEL, P. J. and LEVINA, E. (2004). Some theory for Fisher’s linear discriminant function, ‘naïve Bayes’, and some alternatives when there are many more variables than observations. *Bernoulli*, **10** 989–1010.
- BLANCHARD, G., LEE, G. and SCOTT, C. (2010). Semi-supervised novelty detection. *Journal of Machine Learning Research*, **11** 2973–3009.
- CAR, T. and LIU, W. (2011). A direct estimation approach to sparse linear discriminant analysis. *Journal of the American Statistical Association*, **106** 1566–1577.
- CANNON, A., HOWSE, J., HUSH, D. and SCOVILLE, C. (2002). Learning with the neyman-pearson and min-max criteria. *Technical Report LA-UR-02-2951*.
- CASASENT, D. and CHEN, X. (2003). Radial basis function neural networks for nonlinear fisher discrimination and neyman-pearson classification. *Neural Networks*, **16** 529 – 535.

- DANZIGER, S. A., SWAMIDASS, S. J., ZENG, J., DEARTH, L. R., LU, Q., CHEN, J. H., CHENG, J., HOANG, V. P., SAIGO, H., LUO, R. ET AL. (2006). Functional census of mutation sequence spaces: the example of p53 cancer rescue mutants. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, **3** 114–125.
- DÜMBGEN, L., IGL, B. and MUNK, A. (2008). P-value for classification. *Electronic Journal of Statistics*, **2** 468–493.
- DVORETZKY, A., KIEFER, J. and WOLFOVITZ, J. (1956). Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *The Annals of Mathematical Statistics* 642–669.
- ELKAN, C. (2001). The foundations of cost-sensitive learning. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence* 973–978.
- FAN, J. and FAN, Y. (2008). High-dimensional classification using features annealed independence rules. *The Annals of Statistics*, **36** 2605–2637.
- FAN, J., FENG, Y. and SONG, R. (2011). Nonparametric independence screening in sparse ultra-high-dimensional additive models. *Journal of the American Statistical Association*, **106**.
- FAN, J., FENG, Y. and TONG, X. (2012). A road to classification in high dimensional space: the regularized optimal affine discriminant. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **74** 745–771.
- FAN, J. and LV, J. (2008). Sure independence screening for ultrahigh dimensional feature space (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **70** 849–911.
- GOLUB, T. R., SLONIM, D. K., TAMAYO, P., HUARD, C., GAASENBEEK, M., MESIROV, J. P., COLLIER, H., LOH, M. L., DOWNING, J. R., CALIGURI, M. A., BLOOMFIELD, C. D. and LANDER, E. S. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, **286** 531–537.
- HALL, P., PHAM, T. ET AL. (2010). Optimal properties of centroid-based classifiers for very high-dimensional data. *The Annals of Statistics*, **38** 1071–1093.
- HAN, M., CHEN, D. and SUN, Z. (2008). Analysis to Neyman-Pearson classification with convex loss function. *Analysis in Theory and Applications*, **24** 18–28.
- HASTIE, T., TIBSHIRANI, R. and FRIEDMAN, J. H. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd edition)*. Springer-Verlag Inc.
- JAMES, G., WITTEN, D., HASTIE, T. and TIBSHIRANI, R. (2013). *An introduction to statistical learning*. Springer.
- KOLTCHINSKI, V. (2008). *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems*.
- LI, R., ZHONG, W. and ZHU, L. (2012). Feature screening via distance correlation learning. *Journal of the American Statistical Association*, **107** 1129–1139.
- LIN, Y., LEE, Y. and WAHBA, G. (2002). Support vector machines for classification in nonstandard situations. *Machine learning*, **46** 191–202.
- MAI, Q., ZOU, H. and YUAN, M. (2012). A direct approach to sparse discriminant analysis in ultra-high dimensions. *Biometrika*, **99** 29–42.
- MAMMEN, E. and TSYBAKOV, A. (1999). Smooth discrimination analysis. *The Annals of Statistics*, **27** 1808–1829.
- MEINSHAUSEN, N. and BÜHLMANN, P. (2010). Stability selection (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **72** 417–473.
- OBERTHURER, A., BERTHOLD, F., WARNAT, P., HERO, B., KAHLERT, Y., SPITZ, R., ERNESTUS, K., KNIG, R., HAAS, S., EILS, R., SCHWAB, M., BRORS, B., WESTERMANN, F. and FISCHER, M. (2006). Customized oligonucleotide microarray gene expression-based classification of neuroblastoma patients outperforms current clinical risk stratification. *Journal of Clinical Oncology*, **24** 5070–5078.
- POLONIK, W. (1995). Measuring mass concentrations and estimating density contour clusters—an excess mass approach. *The Annals of Statistics*, **23** 855–881.
- RICOLLET, P. and TONG, X. (2011). Neyman-pearson classification, convexity and stochastic constraints. *Journal of Machine Learning Research*, **12** 2831–2855.
- ROBINS, J., VAN DER VAART, A. ET AL. (2006). Adaptive nonparametric confidence sets. *The Annals of Statistics*, **34** 229–253.
- SCOTT, C. (2005). Comparison and design of neyman-pearson classifiers. Unpublished.
- SCOTT, C. (2007). Performance measures for Neyman-Pearson classification. *IEEE Transactions on Information Theory*, **53** 2852–2863.
- SCOTT, C. and NOWAK, R. (2005). A neyman-pearson approach to statistical learning. *IEEE Transactions on Information Theory*, **51** 3806–3819.
- SHAO, J., WANG, Y., DENG, X. and WANG, S. (2011). Sparse linear discriminant analysis by thresholding for high dimensional data. *The Annals of Statistics*, **39** 1241–1265.
- TARIGAN, B. and VAN DE GEER, S. (2006). Classifiers of support vector machine type with l_1 complexity regularization. *Bernoulli*, **12** 1045–1076.
- TIBSHIRANI, R., HASTIE, T., NARASIMHAN, B. and CHU, G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences of the United States of America*, **99** 6567–6572.
- TONG, X. (2013). A plug-in approach to neyman-pearson classification. *Journal of Machine Learning Research*, **14** 3011–3040.

- TSYBAKOV, A. (2004). Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, **32** 135–166.
- TSYBAKOV, A. (2009). *Introduction to Nonparametric Estimation*. Springer.
- TSYBAKOV, A. and VAN DE GEER, S. (2005). Square root penalty: Adaptation to the margin in classification and in edge estimation. *The Annals of Statistics*, **33** 1203–1224.
- WEISS, S. M., INDURKHVA, N., ZHANG, T. and DAMERAV, F. (2010). *Text mining: predictive methods for analyzing unstructured information*. Springer.
- WITTEN, D. and TRSHIRANI, R. (2012). Penalized classification using fisher’s linear discriminant. *Journal of the Royal Statistical Society Series B*, **73** 753–772.
- WU, S., LIN, K., CHEN, C. and M., C. (2008). *Asymmetric support vector machines: low false-positive learning under the user tolerance*.
- YANG, Y. (1999). Minimax nonparametric classification-part i: rates of convergence. *IEEE Transaction Information Theory*, **45** 2271–2284.
- ZADROZNY, B., LANGFORD, J. and ABE, N. (2003). Cost-sensitive learning by cost-proportionate example weighting. *IEEE International Conference on Data Mining* 435.

A Statistical Perspective on Randomized Sketching for Ordinary Least-Squares

Garvesh Raskutti

Department of Statistics

University of Wisconsin–Madison

Madison, WI 53706, USA

RASKUTTI@STAT.WISC.EDU

Michael W. Mahoney

International Computer Science Institute and Department of Statistics

University of California at Berkeley

Berkeley, CA 94720, USA

MMAHONEY@STAT.BERKELEY.EDU

Editor: Mehryar Mohri

Abstract

We consider statistical as well as algorithmic aspects of solving large-scale least-squares (LS) problems using randomized sketching algorithms. For a LS problem with input data $(X, Y) \in \mathbb{R}^{n \times p} \times \mathbb{R}^n$, sketching algorithms use a “sketching matrix,” $S \in \mathbb{R}^{r \times n}$, where $r \ll n$. Then, rather than solving the LS problem using the full data (X, Y) , sketching algorithms solve the LS problem using only the “sketched data” (SX, SY) . Prior work has typically adopted an *algorithmic perspective*, in that it has made no statistical assumptions on the input X and Y , and instead it has been assumed that the data (X, Y) are fixed and worst-case (WC). Prior results show that, when using sketching matrices such as random projections and leverage-score sampling algorithms, with $p \lesssim r \ll n$, the WC error is the same as solving the original problem, up to a small constant. From a *statistical perspective*, we typically consider the mean-squared error performance of randomized sketching algorithms, when data (X, Y) are generated according to a statistical linear model $Y = X\beta + \epsilon$, where ϵ is a noise process. In this paper, we provide a rigorous comparison of both perspectives leading to insights on how they differ. To do this, we first develop a framework for assessing, in a unified manner, algorithmic and statistical aspects of randomized sketching methods. We then consider the statistical prediction efficiency (PE) and the statistical residual efficiency (RE) of the sketched LS estimator; and we use our framework to provide upper bounds for several types of random projection and random sampling sketching algorithms. Among other results, we show that the RE can be upper bounded when $p \lesssim r \ll n$ while the PE typically requires the sample size r to be substantially larger. Lower bounds developed in subsequent results show that our upper bounds on PE can not be improved.¹

Keywords: algorithmic leveraging, randomized linear algebra, sketching, random projection, statistical leverage, statistical efficiency

1. Introduction

Recent work in large-scale data analysis has focused on developing so-called sketching algorithms: given a data set and an objective function of interest, construct a small “sketch”

1. A preliminary version of this paper appeared as Raskutti and Mahoney (2014, 2015).

of the full data set, e.g., by using random sampling or random projection methods, and use that sketch as a surrogate to perform computations of interest for the full data set (see Mahoney (2011) for a review). Most effort in this area has adopted an *algorithmic perspective*, whereby one shows that, when the sketches are constructed appropriately, one can obtain answers that are approximately as good as the exact answer for the input data at hand, in less time than would be required to compute an exact answer for the data at hand. In statistics, however, one is often more interested in how well a procedure performs relative to an hypothesized model than how well it performs on the particular data set at hand. Thus an important question to consider is whether the insights from the algorithmic perspective of sketching carry over to the statistical setting.

Thus, in this paper, we develop a unified approach that considers both the *statistical perspective* as well as *algorithmic perspective* on recently-developed randomized sketching algorithms, and we provide bounds on two statistical objectives for several types of random projection and random sampling sketching algorithms.

1.1 Overview of the Problem

The problem we consider in this paper is the ordinary least-squares (LS or OLS) problem: given as input a matrix $X \in \mathbb{R}^{n \times p}$ of observed features or covariates and a vector $Y \in \mathbb{R}^n$ of observed responses, return as output a vector β_{OLS} that solves the following optimization problem:

$$\beta_{OLS} = \arg \min_{\beta \in \mathbb{R}^p} \|Y - X\beta\|_2^2. \quad (1)$$

We will assume that n and p are both very large, with $n \gg p$, and for simplicity we will assume $\text{rank}(X) = p$, e.g., to ensure a unique full-dimensional solution. The OLS solution, $\beta_{OLS} = (X^T X)^{-1} X^T Y$, has a number of well-known desirable statistical properties (Chatterjee and Hadi, 1988); and it is also well-known that the running time or computational complexity for this problem is $O(np^2)$ (Golub and Loan, 1996).² For many modern applications, however, n may be on the order of $10^6 - 10^9$ and p may be on the order of $10^3 - 10^4$, and thus computing the exact LS solution with traditional $O(np^2)$ methods can be computationally challenging. This, coupled with the observation that approximate answers often suffice for downstream applications, has led to a large body of work on developing fast approximation algorithms to the LS problem (Mahoney, 2011).

One very popular approach to reducing computation is to perform LS on a carefully-constructed “sketch” of the full data set. That is, rather than computing a LS estimator from Problem (1) from the full data (X, Y) , generate “sketched data” (SX, SY) where $S \in \mathbb{R}^{r \times n}$, with $r \ll n$, is a “sketching matrix,” and then compute a LS estimator from the following sketched problem:

$$\beta_S \in \arg \min_{\beta \in \mathbb{R}^p} \|SY - SX\beta\|_2^2. \quad (2)$$

2. That is, $O(np^2)$ time suffices to compute the LS solution from Problem (1) for arbitrary or worst-case input, with, e.g., the Cholesky Decomposition on the normal equations, with a QR decomposition, or with the Singular Value Decomposition (Golub and Loan, 1996).

Once the sketching operation has been performed, the additional computational complexity of β_S is $O(nr^2)$, i.e., simply call a traditional LS solver on the sketched problem. Thus, when using a sketching algorithm, two criteria are important: first, ensure the accuracy of the sketched LS estimator is comparable to, e.g., not much worse, than the performance of the original LS estimator; and second, ensure that computing and applying the sketching matrix S is not too computationally intensive, e.g., that is faster than solving the original problem exactly.

1.2 Prior Results

Random sampling and random projections provide two approaches to construct sketching matrices S that satisfy both of these criteria and that have received attention recently in the computer science community. Very loosely speaking, a random projection matrix S is a dense matrix,³ where each entry is a mean-zero bounded-variance Gaussian or Rademacher random variable, although other constructions based on randomized Hadamard transformations are also of interest; and a random sampling matrix S is a very sparse matrix that has exactly 1 non-zero entry (which typically equals one multiplied by a rescaling factor) in each row, where that one non-zero can be chosen uniformly, non-uniformly based on hypotheses about the data, or non-uniformly based on empirical statistics of the data such as the leverage scores of the matrix X . In particular, note that a sketch constructed from an $r \times n$ random projection matrix S consists of r linear combinations of most or all of the rows of (X, Y) , and a sketch constructed from a random sampling matrix S consists of r typically-rescaled rows of (X, Y) . Random projection algorithms have received a great deal of attention more generally, largely due to their connections with the Johnson-Lindenstrauss lemma (Johnson and Lindenstrauss, 1984) and its extensions; and random sampling algorithms have received a great deal of attention, largely due to their applications in large-scale data analysis applications (Mahoney and Drineas, 2009). A detailed overview of random projection and random sampling algorithms for matrix problems may be found in the recent monograph of Mahoney (2011). Here, we briefly summarize the most relevant aspects of the theory.

In terms of running time guarantees, the running time bottleneck for random projection algorithms for the LS problem is the application of the projection to the input data, i.e., actually performing the matrix-matrix multiplication to implement the projection and compute the sketch. By using fast Hadamard-based random projections, however, Drineas et al. (2011) developed a random projection algorithm that runs on arbitrary or worst-case input in $o(nr^2)$ time. (See Drineas et al. (2011) for a precise statement of the running time.) As for random sampling, it is trivial to implement uniform random sampling, but it is very easy to show examples of input data on which uniform sampling performs very poorly. On the other hand, Drineas et al. (2006b, 2012) have shown that if the random sampling is performed with respect to nonuniform importance sampling probabilities that depend on the *empirical statistical leverage scores* of the input matrix X , i.e., the diagonal entries of the *hat matrix* $H = X(X^T X)^{-1} X^T$, then one obtains a random sampling algorithm that achieves much better results for arbitrary or worst-case input.

3. The reader should, however, be aware of recently-developed input-sparsity time random projection methods (Clarkson and Woodruff, 2013; Meng and Mahoney, 2013; Nelson and Hry, 2013).

Leverage scores have a long history in robust statistics and experimental design. In

the robust statistics community, samples with high leverage scores are typically flagged as potential outliers (see, e.g., Chatterjee and Hadi (2006, 1988); Hampel et al. (1986); Hoaglin and Weisich (1978); Huber and Ronchetti (1981)). In the experimental design community, samples with high leverage have been shown to improve overall efficiency, provided that the underlying statistical model is accurate (see, e.g., Royall (1970); Zaslavsky et al. (2008)). This should be contrasted with their use in theoretical computer science. From the algorithmic perspective of worst-case analysis, that was adopted by Drineas et al. (2011) and Drineas et al. (2012), samples with high leverage tend to contain the most important information for subsampling/sketching, and thus it is beneficial for worst-case analysis to bias the random sample to include samples with large statistical leverage scores or to rotate to a random basis where the leverage scores are approximately uniformized.

The running-time bottleneck for this leverage-based random sampling algorithm is the computation of the leverage scores of the input data; and the obvious well-known algorithm for this involves $O(nr^2)$ time to perform a QR decomposition to compute an orthogonal basis for X (Golub and Loan, 1996). By using fast Hadamard-based random projections, however, Drineas et al. (2012) showed that one can compute approximate QR decompositions and thus approximate leverage scores in $o(nr^2)$ time, and (based on previous work (Drineas et al., 2006b)) this immediately implies a leverage-based random sampling algorithm that runs on arbitrary or worst-case input in $o(nr^2)$ time (Drineas et al., 2012). Readers interested in the practical performance of these randomized algorithms should consult BENDENPIK (Avron et al., 2010) or ISRN (Meng et al., 2014).

In terms of accuracy guarantees, both Drineas et al. (2011) and Drineas et al. (2012) prove that their respective random projection and leverage-based random sampling LS sketching algorithms each achieve the following worst-case (WC) error guarantee: for any arbitrary (X, Y) ,

$$\|Y - X\beta_S\|_2^2 \leq (1 + \kappa) \|Y - X\beta_{OLS}\|_2^2, \quad (3)$$

with high probability for some pre-specified error parameter $\kappa \in (0, 1)$.⁴ This $1 + \kappa$ relative-error guarantee⁵ is extremely strong, and it is applicable to arbitrary or worst-case input. That is, whereas in statistics one typically assumes a model, e.g., a standard linear model on Y ,

$$Y = X\beta + \epsilon, \quad (4)$$

where $\beta \in \mathbb{R}^p$ is the true parameter and $\epsilon \in \mathbb{R}^n$ is a standardized noise vector, with $\mathbb{E}[\epsilon] = 0$ and $\mathbb{E}[\epsilon\epsilon^T] = I_{n \times n}$, in Drineas et al. (2011) and Drineas et al. (2012) no statistical model is assumed on X and Y , and thus the running time and quality-of-approximation bounds apply to any arbitrary (X, Y) input data.

1.3 Our Approach and Main Results

In this paper, we adopt a statistical perspective on these randomized sketching algorithms, and we address the following fundamental questions. First, under a standard linear model, e.g., as given in Eqn. (4), what properties of a sketching matrix S are sufficient to ensure

4. The quantity $\|\beta_S - \beta_{OLS}\|_2^2$ is also bounded by Drineas et al. (2011) and Drineas et al. (2012).

5. The nonstandard parameter κ is used here for the error parameter since ϵ is used below to refer to the noise or error process.

low statistical error, e.g., mean-squared, error? Second, how do existing random projection algorithms and leverage-based random sampling algorithms perform by this statistical measure? Third, how does this relate to the properties of a sketching matrix S that are sufficient to ensure low worst-case error, e.g., of the form of Eqn. (3), as has been established previously in Drineas et al. (2011, 2012); Mahoney (2011)? We address these related questions in a number of steps.

In Section 2, we will present a framework for evaluating the algorithmic and statistical properties of randomized sketching methods in a unified manner; and we will show that providing worst-case error bounds of the form of Eqn. (3) and providing bounds on two related statistical objectives boil down to controlling different structural properties of how the sketching matrix S interacts with the left singular subspace of the design matrix. In particular, we will consider the oblique projection matrix, $\Pi_S^U = U(SU)^\dagger S$, where $(\cdot)^\dagger$ denotes the Moore-Penrose pseudo-inverse of a matrix and U is the left singular matrix of X . This framework will allow us to draw a comparison between the worst-case error and two related statistical efficiency criteria, the statistical prediction efficiency (PE) (which is based on the prediction error $\mathbb{E}\|X(\hat{\beta} - \beta)\|_2^2$) and which is given in Eqn. (7) below) and the statistical residual efficiency (RE) (which is based on residual error $\mathbb{E}\|Y - X\hat{\beta}\|_2^2$) and which is given in Eqn. (8) below); and it will allow us to provide sufficient conditions that any sketching matrix S must satisfy in order to achieve performance guarantees for these two statistical objectives.

In Section 3, we will present our main theoretical results, which consist of bounds for these two statistical quantities for variants of random sampling and random projection sketching algorithms. In particular, we provide upper bounds on the PE and RE (as well as the worst-case WC) for four sketching schemes: (1) an approximate leverage-based random sampling algorithm, as is analyzed by Drineas et al. (2012); (2) a variant of leverage-based random sampling, where the random samples are *not* re-scaled prior to their inclusion in the sketch, as is considered by Ma et al. (2014, 2015); (3) a vanilla random projection algorithm, where S is a random matrix containing i.i.d. Gaussian or Rademacher random variables, as is popular in statistics and scientific computing; and (4) a random projection algorithm, where S is a random Hadamard-based random projection, as analyzed in Boutsidis and Gittens (2013). For sketching schemes (1), (3), and (4), our upper bounds for each of the two measures of statistical efficiency are identical up to constants; and they show that the RE scales as $1 + \frac{p}{r}$, while the PE scales as $\frac{p}{r}$. In particular, this means that it is possible to obtain good bounds for the RE when $p \lesssim r \ll n$ (in a manner similar to the sampling complexity of the WC bounds); but in order to obtain even near-constant bounds for PE, r must be at least of constant order compared to n . We then present a lower bound developed in subsequent work by Pilanci and Wainwright (2014) which shows that under general conditions on S , our upper bound of $\frac{p}{r}$ for PE can not be improved. For the sketching scheme (2), we show, on the other hand, that under the strong assumption that there are k “large” leverage scores and the remaining $n - k$ are “small,” then the WC scales as $1 + \frac{p}{r}$, the RE scales as $1 + \frac{2pk}{r}$, and the PE scales as $\frac{k}{r}$. That is, sharper bounds are possible for leverage-score sampling without re-scaling in the statistical setting, but much stronger assumptions are needed on the input data.

In Section 4, we will supplement our theoretical results by presenting our main empirical results, which consist of an evaluation of the complementary properties of random sampling

versus random projection methods. Our empirical results support our theoretical results, and they also show that for r larger than p but much closer to p than n , projection-based methods tend to out-perform sampling-based methods, while for r significantly larger than p , our leverage-based sampling methods perform slightly better. In Section 5, we will provide a brief discussion and conclusion and we provide proofs of our main results in the Appendix.

1.4 Additional Related Work

Very recently Ma et al. (2014) considered statistical aspects of leverage-based sampling algorithms (called *algorithmic leveraging* in Ma et al. (2014)). Assuming a standard linear model on Y of the form of Eqn. (4), the authors developed first-order Taylor approximations to the statistical relative efficiency of different estimators computed with leverage-based sampling algorithms, and they verified the quality of those approximations with computations on real and synthetic data. Taken as a whole, their results suggest that, if one is interested in the statistical performance of these randomized sketching algorithms, then there are nontrivial trade-offs that are not taken into account by standard worst-case analysis. Their approach, however, does not immediately apply to random projections or other more general sketching matrices. Further, the realm of applicability of the first-order Taylor approximation was not precisely quantified, and they left open the question of structural characterizations of random sketching matrices that were sufficient to ensure good statistical properties on the sketched data. We address these issues in this paper.

After the appearance of the original technical report version of this paper (Raskutti and Mahoney, 2014), we were made aware of subsequent work by Pilanci and Wainwright (2014), who also consider a statistical perspective on sketching. Amongst other results, they develop a lower bound which confirms that using a single randomized sketching matrix S can not achieve a PE better than $\frac{p}{r}$. This lower bound complements our upper bounds developed in this paper. Their main focus is to use this insight to develop an iterative sketching scheme which yields bounds on the PE when an $r \times n$ sketch is applied repeatedly.

2. General Framework and Structural Results

In this section, we develop a framework that allows us to view the algorithmic and statistical perspectives on LS problems from a common perspective. We then use this framework to show that existing worst-case bounds as well as our novel statistical bounds for the mean-squared errors can be expressed in terms of different structural conditions on how the sketching matrix S interacts with the data (X, Y) .

2.1 A Statistical-Algorithmic Framework

Recall that we are given as input a data set, $(X, Y) \in \mathbb{R}^{n \times p} \times \mathbb{R}^n$, and the objective function of interest is the standard LS objective, as given in Eqn. (1). Since we are assuming, without loss of generality, that $\text{rank}(X) = p$, we have that

$$\beta_{OLS} = X^\dagger Y = (X^T X)^{-1} X^T Y, \quad (5)$$

where $(\cdot)^{\dagger}$ denotes the Moore-Penrose pseudo-inverse of a matrix, and where the second equality follows since $\text{rank}(X) = p$.

To present our framework and objectives, let $S \in \mathbb{R}^{r \times n}$ denote an *arbitrary* sketching matrix. That is, although we will be most interested in sketches constructed from random sampling or random projection operations, for now we let S be *any* $r \times n$ matrix. Then, we are interested in analyzing the performance of objectives characterizing the quality of a “sketched” LS objective, as given in Eqn (2), where again we are interested in solutions of the form

$$\beta_S = (SX)^{\dagger}SY. \quad (6)$$

(We emphasize that this does *not* in general equal $((SX)^T SX)^{-1}(SX)^T SY$, since the inverse will *not* exist if the sketching process does not preserve rank.) Our goal here is to compare the performance of β_S to β_{OLS} . We will do so by considering three related performance criteria, two of a statistical flavor, and one of a more algorithmic or worst-case flavor.

From a statistical perspective, it is common to assume a standard linear model on Y ,

$$Y = X\beta + \epsilon,$$

where we remind the reader that $\beta \in \mathbb{R}^p$ is the true parameter and $\epsilon \in \mathbb{R}^n$ is a standardized noise vector, with $\mathbb{E}[\epsilon] = 0$ and $\mathbb{E}[\epsilon\epsilon^T] = I_{n \times n}$. From this statistical perspective, we will consider the following two criteria.

- The first statistical criterion we consider is the *prediction efficiency* (PE), defined as follows:

$$C_{PE}(S) = \frac{\mathbb{E}[\|X(\beta - \beta_S)\|_2^2]}{\mathbb{E}[\|X(\beta - \beta_{OLS})\|_2^2]}, \quad (7)$$

where the expectation $\mathbb{E}[\cdot]$ is taken over the random noise ϵ .

- The second statistical criterion we consider is the *residual efficiency* (RE), defined as follows:

$$C_{RE}(S) = \frac{\mathbb{E}[\|Y - X\beta_S\|_2^2]}{\mathbb{E}[\|Y - X\beta_{OLS}\|_2^2]}, \quad (8)$$

where, again, the expectation $\mathbb{E}[\cdot]$ is taken over the random noise ϵ .

Recall that the standard relative statistical efficiency for two estimators β_1 and β_2 is defined as $\text{eff}(\beta_1, \beta_2) = \frac{\text{Var}(\beta_2)}{\text{Var}(\beta_1)}$, where $\text{Var}(\cdot)$ denotes the variance of the estimator (see e.g., Lehmann (1998)). For the PE, we have replaced the variance of each estimator by the mean-squared prediction error. For the RE, we use the term residual since for any estimator $\hat{\beta}$, $Y - X\hat{\beta}$ are the residuals for estimating Y .

From an algorithmic perspective, there is no noise process ϵ . Instead, X and Y are arbitrary, and β is simply computed from Eqn (5). To draw a parallel with the usual statistical generative process, however, and to understand better the relationship between various objectives, consider “defining” Y in terms of X by the following “linear model”:

$$Y = X\beta + \epsilon,$$

where $\beta \in \mathbb{R}^p$ and $\epsilon \in \mathbb{R}^n$. Importantly, β and ϵ here represent different quantities than in the usual statistical setting. Rather than ϵ representing a noise process and β representing

a “true parameter” that is observed through a noisy Y , here in the algorithmic setting, we will take advantage of the rank-nullity theorem in linear algebra to relate X and Y .⁶ To define a “worst case model” $Y = X\beta + \epsilon$ for the algorithmic setting, one can view the “noise” process ϵ to consist of any vector that lies in the null-space of X^T . Then, since the choice of $\beta \in \mathbb{R}^p$ is arbitrary, one can construct any arbitrary or worst-case input data Y . From this algorithmic case, we will consider the following criterion.

- The algorithmic criterion we consider is the *worst-case* (WC) error, defined as follows:

$$C_{WC}(S) = \sup_Y \frac{\|Y - X\beta_S\|_2^2}{\|Y - X\beta_{OLS}\|_2^2}. \quad (9)$$

This criterion is worst-case since we take a supremum Y , and it is the performance criterion that is analyzed in Drineas et al. (2011) and Drineas et al. (2012), as bounded in Eqn. (3).

Writing Y as $X\beta + \epsilon$, where $X^T\epsilon = 0$, the WC error can be re-expressed as:

$$C_{WC}(S) = \sup_{Y=X\beta+\epsilon, X^T\epsilon=0} \frac{\|Y - X\beta_S\|_2^2}{\|Y - X\beta_{OLS}\|_2^2}.$$

Hence, in the worst-case algorithmic setup, we take a supremum over ϵ , where $X^T\epsilon = 0$, whereas in the statistical setup, we take an expectation over ϵ where $\mathbb{E}[\epsilon] = 0$.

Before proceeding, several other comments about this algorithmic-statistical framework and our objectives are worth mentioning.

- The most important distinction between the algorithmic approach and the statistical approach is how the data is assumed to be generated. For the statistical approach, (X, Y) are assumed to be generated by a standard Gaussian linear model and the goal is to estimate a true parameter β while for the algorithmic approach (X, Y) are not assumed to follow any statistical model and the goal is to do prediction on Y rather than estimate a true parameter β . Since ordinary least-squares is often run in the context of solving a statistical inference problem, we believe this distinction is important and focus in this article more on the implications for the statistical perspective.
- From the perspective of our two linear models, we have that $\beta_{OLS} = \beta + (X^T X)^{-1} X^T \epsilon$. In the statistical setting, since $\mathbb{E}[\epsilon\epsilon^T] = I_{n \times n}$, it follows that β_{OLS} is a random variable with $\mathbb{E}[\beta_{OLS}] = \beta$ and $\mathbb{E}[(\beta - \beta_{OLS})(\beta - \beta_{OLS})^T] = (X^T X)^{-1}$. In the algorithmic setting, on the other hand, since $X^T\epsilon = 0$, it follows that $\beta_{OLS} = \beta$.
- $C_{RE}(S)$ is a statistical analogue of the worst-case algorithmic objective $C_{WC}(S)$, since both consider the ratio of the metrics $\frac{\|Y - X\beta_S\|_2^2}{\|Y - X\beta_{OLS}\|_2^2}$. The difference is that a sup over Y in the algorithmic setting is replaced by an expectation over noise ϵ in the statistical

⁶ The rank-nullity theorem asserts that given any matrix $X \in \mathbb{R}^{n \times p}$ and vector $Y \in \mathbb{R}^n$, there exists a unique decomposition $Y = X\beta + \epsilon$, where β is the projection of Y on to the range space of X^T and $\epsilon = Y - X\beta$ lies in the null-space of X^T (Meyer, 2000).

setting. A natural question is whether there is an algorithmic analogue of $C_{PE}(S)$. Such a performance metric would be:

$$\sup_Y \frac{\|X(\beta - \beta_S)\|_2^2}{\|X(\beta - \beta_{OLS})\|_2^2}, \quad (10)$$

where β is the projection of Y on to the range space of X^T . However, since $\beta_{OLS} = \beta + (X^T X)^{-1} X^T \epsilon$ and since $X^T \epsilon = 0$, $\beta_{OLS} = \beta$ in the algorithmic setting, the denominator of Eqn. (10) equals zero, and thus the objective in Eqn. (10) is not well-defined. The ‘‘difficulty’’ of computing or approximating this objective parallels our results below that show that approximating $C_{PE}(S)$ is much more challenging (in terms of the number of samples needed) than approximating $C_{RE}(S)$.

- In the algorithmic setting, the sketching matrix S and the objective $C_{WC}(S)$ can depend on X and Y in any arbitrary way, but in the following we consider only sketching matrices that are either independent of both X and Y or depend only on X (e.g., via the statistical leverage scores of X). In the statistical setting, S is allowed to depend on X , but not on Y , as any dependence of S on Y might introduce correlation between the sketching matrix and the noise variable ϵ . Removing this restriction is of interest, especially since one can obtain WC bounds of the form Eqn. (3) by constructing S by randomly sampling according to an importance sampling distribution that depends on the *influence scores*—essentially the leverage scores of the matrix X augmented with $-Y$ as an additional column—of the (X, Y) pair.

- Both $C_{PE}(S)$ and $C_{RE}(S)$ are qualitatively related to quantities analyzed by Ma et al. (2014, 2015). In addition, $C_{WC}(S)$ is qualitatively similar to $\text{Cov}(\hat{\beta})Y$ in Ma et al. (2014, 2015), since in the algorithmic setting Y is treated as fixed; and $C_{RE}(S)$ is qualitatively similar to $\text{Cov}(\hat{\beta})$ in Ma et al. (2014, 2015), since in the statistical setting Y is treated as random and coming from a linear model. That being said, the metrics and results we present in this paper are not directly comparable to those of Ma et al. (2014, 2015) since, e.g., they had a slightly different setup than we have here, and since they used a first-order Taylor approximation while we do not.

2.2 Structural Results on Sketching Matrices

We are now ready to develop structural conditions characterizing how the sketching matrix S interacts with the data matrix X that will allow us to provide upper bounds for the quantities $C_{WC}(S)$, $C_{PE}(S)$, and $C_{RE}(S)$. To do so, recall that given the data matrix X , we can express the singular value decomposition of X as $X = U\Sigma V^T$, where $U \in \mathbb{R}^{n \times p}$ is an orthogonal matrix, i.e., $U^T U = I_{p \times p}$. In addition, we can define the *oblique projection* matrix

$$\Pi_S^U := U(SU)^\dagger S. \quad (11)$$

Note that if $\text{rank}(SX) = p$, then Π_S^U can be expressed as $\Pi_S^U = U(U^T S^T S U)^{-1} U^T S^T S$, since $U^T S^T S U$ is invertible. Importantly however, depending on the properties of X and how S is constructed, it can easily happen that $\text{rank}(SX) < p$, even if $\text{rank}(X) = p$.

Given this setup, we can now state the following lemma, the proof of which may be found in Section A.1. This lemma characterizes how $C_{WC}(S)$, $C_{PE}(S)$, and $C_{RE}(S)$ depend on different structural properties of Π_S^U and SU .

Lemma 1 *For the algorithmic setting,*

$$C_{WC}(S) = 1 + \sup_{\delta \in \mathbb{R}^p, U^T \epsilon = 0} \left[\frac{\|(I_{p \times p} - (SU)^\dagger (SU) \delta)\|_2^2}{\|\epsilon\|_2^2} + \frac{\|\Pi_S^U \epsilon\|_2^2}{\|\epsilon\|_2^2} \right].$$

For the statistical setting,

$$C_{PE}(S) = \frac{p}{\|(I_{p \times p} - (SU)^\dagger (SU) \Sigma V^T \beta)\|_2^2} + \frac{\|\Pi_S^U \beta\|_2^2}{p},$$

and

$$C_{RE}(S) = 1 + \frac{\|(I_{p \times p} - (SU)^\dagger (SU) \Sigma V^T \beta)\|_2^2}{n - p} + \frac{\|\Pi_S^U \beta\|_2^2}{n - p} = 1 + \frac{C_{PE}(S) - 1}{n/p - 1}.$$

Several points are worth making about Lemma 1.

- For all 3 criteria, the term which involves $(SU)^\dagger SU$ is a ‘‘bias’’ term that is non-zero in the case that $\text{rank}(SU) < p$. For $C_{PE}(S)$ and $C_{RE}(S)$, the term corresponds exactly to the statistical bias; and if $\text{rank}(SU) = p$, meaning that S is a *rank-preserving* sketching matrix, then the bias term equals 0, since $(SU)^\dagger SU = I_{p \times p}$. In practice, if r is chosen smaller than p or larger than but very close to p , it may happen that $\text{rank}(SU) < p$, in which case this bias is incurred.
- The final equality $C_{RE}(S) = 1 + \frac{C_{PE}(S) - 1}{n/p - 1}$ shows that in general it is much more difficult (in terms of the number of samples needed) to obtain bounds on $C_{PE}(S)$ than $C_{RE}(S)$ —since $C_{RE}(S)$ re-scales $C_{PE}(S)$ by p/n , which is much less than 1. This will be reflected in the main results below, where the scaling of $C_{RE}(S)$ will be a factor of p/n smaller than $C_{PE}(S)$. In general, it is significantly more difficult to bound $C_{PE}(S)$, since $\|X(\beta - \beta_{OLS})\|_2^2$ is p , whereas $\|Y - X\beta_{OLS}\|_2^2$ is $n - p$, and so there is much less margin for error in approximating $C_{PE}(S)$.
- In the algorithmic or worst-case setting, $\sup_{\epsilon \in \mathbb{R}^p, \langle \epsilon, 0 \rangle} \frac{\|\Pi_S^U \epsilon\|_2^2}{\|\epsilon\|_2^2}$ is the relevant quantity, whereas in the statistical setting $\|\Pi_S^U\|_F^2$ is the relevant quantity. The Frobenius norm enters in the statistical setting because we are taking an average over homoscedastic noise, and so the ℓ_2 norm of the eigenvalues of Π_S^U need to be controlled. On the other hand, in the algorithmic or worst-case setting, the worst direction in the null-space of U^T needs to be controlled, and thus the spectral norm enters.

3. Main Theoretical Results

In this section, we provide upper bounds for $C_{WC}(S)$, $C_{PE}(S)$, and $C_{RE}(S)$, where S correspond to random sampling and random projection matrices. In particular, we provide upper bounds for 4 sketching matrices: (1) a vanilla leverage-based random sampling algorithm

from Drineas et al. (2012): (2) a variant of leverage-based random sampling, where the random samples are *not* re-scaled prior to their inclusion in the sketch; (3) a vanilla random projection algorithm, where S is a random matrix containing i.i.d. sub-Gaussian random variables; and (4) a random projection algorithm, where S is a random Hadamard-based random projection, as analyzed in Boutsidis and Gittens (2013).

3.1 Random Sampling Methods

Here, we consider random sampling algorithms. To do so, first define a random sampling matrix $\tilde{S} \in \mathbb{R}^n$ as follows: $\tilde{S}_{ij} \in \{0, 1\}$ for all (i, j) and $\sum_{j=1}^n \tilde{S}_{ij} = 1$, where each row has an independent multinomial distribution with probabilities $(p_i)_{i=1}^n$. The matrix of cross-leverage scores is defined as $L = UU^T \in \mathbb{R}^{n \times n}$, and $\ell_i = L_{ii}$ denotes the leverage score corresponding to the i^{th} sample. Note that the leverage scores satisfy $\sum_{i=1}^n \ell_i = \text{trace}(L) = p$ and $0 \leq \ell_i \leq 1$.

The sampling probability distribution we consider $(p_i)_{i=1}^n$ is of the form $p_i = (1 - \theta)\frac{\ell_i}{p} + \theta q_i$, where $\{q_i\}_{i=1}^n$ satisfies $0 \leq q_i \leq 1$ and $\sum_{i=1}^n q_i = 1$ is an arbitrary probability distribution, and $0 \leq \theta < 1$. In other words, it is a convex combination of a leverage-based distribution and another arbitrary distribution. Note that for $\theta = 0$, the probabilities are proportional to the leverage scores, whereas for $\theta = 1$, the probabilities follow $\{\ell_i\}_{i=1}^n$.

We consider two sampling matrices, one where the random sampling matrix is re-scaled, as in Drineas et al. (2011), and one in which no re-scaling takes place. In particular, let $S_{NR} = \tilde{S}$ denote the random sampling matrix (where the subscript NR denotes the fact that no re-scaling takes place). The re-scaled sampling matrix is $SR \in \mathbb{R}^{n \times n} = \tilde{S}W$, where $W \in \mathbb{R}^{n \times n}$ is a diagonal re-scaling matrix, where $[W]_{jj} = \sqrt{\frac{1}{\tau p_j}}$ and $W_{ji} = 0$ for $j \neq i$. The quantity $\frac{1}{p_j}$ is the re-scaling factor. In this case, we have the following result, the proof of which may be found in Section B.1.

Theorem 1 For $S = S_R$, there exists constants C and C' such that if $\tau \geq \frac{Cp}{1-\theta} \log\left(\frac{C'p}{1-\theta}\right)$, $\text{rank}(S_R U) = p$ and:

$$\begin{aligned} C_{WC}(S_R) &\leq 1 + 12\frac{p}{r} \\ C_{PE}(S_R) &\leq 44\frac{p}{r} \\ C_{RE}(S_R) &\leq 1 + 44\frac{p}{r}, \end{aligned}$$

with probability at least 0.7 .

Several things are worth noting about this result. First, note that both $C_{WC}(S_R) - 1$ and $C_{RE}(S_R) - 1$ scale as $\frac{p}{r}$; thus, it is possible to obtain high-quality performance guarantees for ordinary least squares, as long as $\frac{p}{r} \rightarrow 0$, e.g., if r is only slightly larger than p . On the other hand, $C_{PE}(S_R)$ scales as $\frac{p}{r}$, meaning r needs to be close to n to provide similar performance guarantees. Next, note that all of the upper bounds apply to any data matrix X , without assuming any additional structure on X . Finally, note that when $\theta = 1$, which corresponds to sampling the rows based on $\{\ell_i\}_{i=1}^n$, all the upper bounds are ∞ . Our simulations also reveal that uniform sampling generally performs more poorly than leverage-score based approaches under the linear models we consider.

An important practical point is the following: the distribution $\{q_i\}_{i=1}^n$ does *not* enter the results. This allows us to consider different distributions. An obvious choice is uniform, i.e., $q_i = \frac{1}{n}$ (see e.g., Ma et al. (2014, 2015)). Another important example is that of *approximate* leverage-score sampling, as developed in Drineas et al. (2012). (The running time of the main algorithm of Drineas et al. (2012) is $o(np^2)$, and thus this reduces computation compared with the use of exact leverage scores, which take $O(np^2)$ time to compute). Let $(\ell_i)_{i=1}^n$ denote the approximate leverage scores developed by the procedure in Drineas et al. (2012). Based on Theorem 2 in Drineas et al. (2012), $|\ell_i - \tilde{\ell}_i| \leq \epsilon$ where $0 < \epsilon < 1$ for r appropriately chosen. Now, using $p_i = \frac{\tilde{\ell}_i}{p}$, p_i can be re-expressed as $p_i = (1 - \epsilon)\frac{\ell_i}{p} + \epsilon q_i$ where $(q_i)_{i=1}^n$ is a distribution (unknown since we only have a bound on the approximate leverage scores). Hence, the performance bounds achieved by approximate leveraging are analogous to those achieved by adding ϵ multiplied by a uniform or other arbitrary distribution.

Next, we consider the leverage-score estimator without re-scaling S_{NR} . In order to develop nontrivial bounds on $C_{WC}(S_{NR})$, $C_{PE}(S_{NR})$, and $C_{RE}(S_{NR})$, we need to make a strong assumption on the leverage-score distribution on X . To do so, we define the following.

Definition 1 (k-heavy hitter leverage distribution) A sequence of leverage scores $(\ell_i)_{i=1}^n$ is a k -heavy hitter leverage score distribution if there exist constants $c, C > 0$ such that for $1 \leq i \leq k$, $\frac{c}{p} \leq \ell_i \leq \frac{Cp}{k}$ and for the remaining $n - k$ leverage scores, $\sum_{i=k+1}^n \ell_i \leq \frac{3}{4}$.

The interpretation of a k -heavy hitter leverage distribution is one in which only k samples in X contain the majority of the leverage score mass. In the simulations below, we provide examples of synthetic matrices X where the majority of the mass is in the largest leverage scores. The parameter k acts as a measure of non-uniformity, in that the smaller the k , the more non-uniform are the leverage scores. The k -heavy hitter leverage distribution allows us to model highly non-uniform leverage scores. In this case, we have the following result, the proof of which may be found in Section B.2.

Theorem 2 For $S = S_{NR}$, with $\theta = 0$ and assuming a k -heavy hitter leverage distribution and, there exist constants c_1 and $r \geq c_1 p \log(c_2 p)$, such that $\text{rank}(S_{NR}) = p$ and:

$$\begin{aligned} C_{WC}(S_{NR}) &\leq 1 + \frac{44C^2 p}{c^2 r} \\ C_{PE}(S_{NR}) &\leq \frac{44C^4 k}{c^2 r} \\ C_{RE}(S_{NR}) &\leq 1 + \frac{44C^4 pk}{c^2 mp}, \end{aligned}$$

with probability at least 0.6 .

Notice that when $k \ll n$, bounds in Theorem 2 on $C_{PE}(S_{NR})$ and $C_{RE}(S_{NR})$ are significantly sharper than bounds in Theorem 1 on $C_{PE}(S_R)$ and $C_{RE}(S_R)$. Hence not re-scaling has the potential to provide sharper bound in the statistical setting. However a stronger assumption on X is needed for this result.

3.2 Random Projection Methods

Here, we consider two random projection algorithms, one based on a sub-Gaussian projection matrix and the other based on a Hadamard projection matrix. To do so, define $[S_{SGP}]_{ij} = \frac{1}{\sqrt{p}} X_{ij}$, where $(X_{ij})_{1 \leq i \leq r, 1 \leq j \leq n}$ are i.i.d. sub-Gaussian random variables with $\mathbb{E}[X_{ij}] = 0$, variance $\mathbb{E}[X_{ij}^2] = \sigma^2$ and sub-Gaussian parameter 1. In this case, we have the following result, the proof of which may be found in Section B.3.

Theorem 3 *For any matrix X , there exists a constant c such that if $r \geq c \log n$, then with probability greater than 0.7, it holds that $\text{rank}(S_{SGP}) = p$ and that:*

$$\begin{aligned} C_{WC}(S_{SGP}) &\leq 1 + 11 \frac{p}{r} \\ C_{PE}(S_{SGP}) &\leq 44 \left(1 + \frac{n}{r}\right) \\ C_{RE}(S_{SGP}) &\leq 1 + 44 \frac{p}{r}. \end{aligned}$$

Notice that the bounds in Theorem 3 for S_{SGP} are equivalent to the bounds in Theorem 1 for S_R , except that r is required only to be larger than $O(\log n)$ rather than $O(p \log p)$. Hence for smaller values of p , random sub-Gaussian projections are more stable than leverage-score sampling based approaches. This reflects the fact that to a first-order approximation, leverage-score sampling performs as well as performing a smooth projection.

Next, we consider the randomized Hadamard projection matrix. In particular, $S_{Had} = S_{ni,f} H D$, where $H \in \mathbb{R}^{n \times n}$ is the standard Hadamard matrix (see e.g., Hedayat and Wallis (1978)), $S_{ni,f} \in \mathbb{R}^{r \times n}$ is an $r \times n$ uniform sampling matrix, and $D \in \mathbb{R}^{n \times n}$ is a diagonal matrix with random equiprobable ± 1 entries. In this case, we have the following result, the proof of which may be found in Section B.4.

Theorem 4 *For any matrix X , there exists a constant c such that if $r \geq cp \log n (\log p + \log \log n)$, then with probability greater than 0.8, it holds that $\text{rank}(S_{Had}) = p$ and that:*

$$\begin{aligned} C_{WC}(S_{Had}) &\leq 1 + 40 \log(np) \frac{p}{r} \\ C_{RE}(S_{Had}) &\leq 40 \log(np) \left(1 + \frac{n}{r}\right) \\ C_{PE}(S_{Had}) &\leq 1 + 40 \log(np) \left(1 + \frac{p}{r}\right). \end{aligned}$$

Notice that the bounds in Theorem 4 for S_{Had} are equivalent to the bounds in Theorem 1 for S_R , up to a constant and $\log(np)$ factor. As discussed in Drineas et al. (2011), the Hadamard transformation makes the leverage scores of X approximately uniform (up to a $\log(np)$ factor), which is why the performance is similar to the sub-Gaussian projection (which also tends to make the leverage scores of X approximately uniform). We suspect that the additional $\log(np)$ factor is an artifact of the analysis since we use an entry-wise concentration bound; using more sophisticated techniques, we believe that the $\log(np)$ can be removed. The probabilities of 0.6, 0.7, and 0.8 for which the upper bounds hold in the four Theorems above is an artifact of the concentration bounds used in the proof and can be improved at the expense of weaker constants (e.g. 40) in front of the efficiency bounds. As

we show in the next section, the upper bound of $\frac{n}{r}$ on $C_{PE}(S)$ for $S = S_R, S_{SGP}$ and S_{Had} can not be improved up to constant while for $S = S_{NR}$ the upper bound of $\frac{k}{r}$ can not be improved.

3.3 Lower Bounds

Subject to the dissemination of the original version of this paper (Raskutti and Mahoney, 2014), Pilanci and Wainwright (2014) amongst other results develop lower bounds on the numerator in $C_{PE}(S)$. This proves that our upper bounds on $C_{PE}(S)$ can not be improved. We re-state Theorem 1 (Example 1) in Pilanci and Wainwright (2014) in a way that makes it most comparable to our results.

Theorem 5 (Theorem 1 in Pilanci and Wainwright (2014)) *For any sketching matrix satisfying $\|\mathbb{E}[S^T(SS^T)^{-1}S]\|_{op} \leq \eta \frac{n}{r}$, any estimator based on (SX, SY) satisfies the lower bound with probability greater than 1/2:*

$$C_{PE}(S) \geq \frac{n}{128\eta r}.$$

Pilanci and Wainwright (2014) show that for $S = S_R, S_{SGP}$ and S_{Had} , $\|\mathbb{E}[S^T(SS^T)^{-1}S]\|_{op} \leq c \frac{n}{r}$ where c is a constant and hence $\eta = c$ and the lower bound matches our upper bounds up to constant. On the other hand, for $S = S_{NR}$, it is straightforward to show that $\|\mathbb{E}[S^T(SS^T)^{-1}S]\|_{op} \leq c \frac{k}{r}$ for some constant c and hence $\eta = c \frac{k}{r}$ and the lower bound scales as $\frac{k}{r}$, to match the upper bound on $C_{PE}(S_{NR})$ from Theorem 2. This is why we are able to prove a tighter upper bound when the matrix X has highly non-uniform leverage scores.

Importantly, this proves that $C_{PE}(S)$ is a quantity that is more challenging to control than $C_{RE}(S)$ and $C_{WC}(S)$ when only a single sketch is used. Using this insight, Pilanci and Wainwright (2014) show that by using a particular iterative Hessian sketch, $C_{PE}(S)$ can be controlled up to constant. In addition to providing a lower bound on the PE using a sketching matrix just once, Pilanci and Wainwright (2014) also develop a new iterative sketching scheme where sketching matrices are used repeatedly can reduce the PE significantly. Once again the probability of 0.5 can be improved by making the constant of $\frac{1}{128}$ less tight.

Finally in prior related work, Lu and Foster (2014); Lu et al. (2013) show that the rate $1 + \frac{p}{r}$ may be achieved for the PE using the estimator $\hat{\beta} = ((SX)^T(SX))^{-1}X^TY$. This estimator is related to the ridge regression estimator since sketches or random projections are applied only in the computation of the X^TX matrix and not X^TY . Since both X^TY and $(SX)^T(SX)$ have small dimension, this estimator has significant computational benefits. However this estimator does not violate the lower bound in Pilanci and Wainwright (2014) since it is not based on the sketches (SX, SY) but instead uses (SX, X^TY) .

4. Empirical Results

In this section, we present the results of an empirical evaluation, illustrating the results of our theory. We will compare the following 6 sketching matrices.

- (1) $S = S_R$ - random leverage-score sampling with re-scaling.

- (2) $S = S_{NVR}$ - random leverage-score sampling without re-scaling.
- (3) $S = S_{Unif}$ - random uniform sampling (each sample drawn independently with probability $1/n$).
- (4) $S = S_{SIR}$ - random leverage-score sampling with re-scaling and with $\theta = 0.1$.
- (5) $S = S_{GCP}$ - Gaussian projection matrices.
- (6) $S = S_{Had}$ - Hadamard projections.

To compare the methods and see how they perform on inputs with different leverage scores, we generate test matrices using a method outlined in Ma et al. (2014, 2015). Set $n = 1024$ (to ensure, for simplicity, an integer power of 2 for the Hadamard transform) and $p = 50$, and let the number of samples drawn with replacement, r , be varied. X is then generated based on a t -distribution with different choices of ν to reflect different uniformity of leverage scores. Each row of X is selected independently with distribution $X_i \sim t_p(\Sigma)$, where Σ corresponds to an auto-regressive model with ν the degrees of freedom. The 3 values of ν presented here are $\nu = 1$ (highly non-uniform), $\nu = 2$ (moderately non-uniform), and $\nu = 10$ (very uniform). See Figure 1 for a plot to see how ν influences the uniformity of the leverage scores. For each setting, the simulation is repeated 100 times in order to average

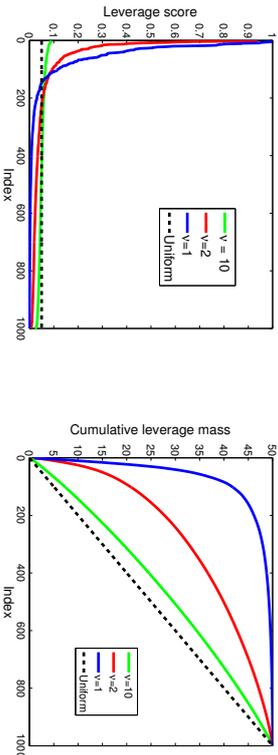


Figure 1: Ordered leverage scores for different values of ν (a) and cumulative sum of ordered leverage scores for different values of ν (b).

over both the randomness in the sampling, and in the statistical setting, the randomness over y .

Note that a natural comparison can be drawn between the parameter ν and the parameter k in the k -heavy hitter definition. If we want to find the value k such that 90% of the leverage mass is captured, for $\nu = 1$, $k \approx 100$, for $\nu = 2$, $k \approx 700$ and for $\nu = 10$, $k \approx 900$, according to Figure 1 (b). Hence the smaller ν , the smaller k since the leverage-scores are more non-uniform.

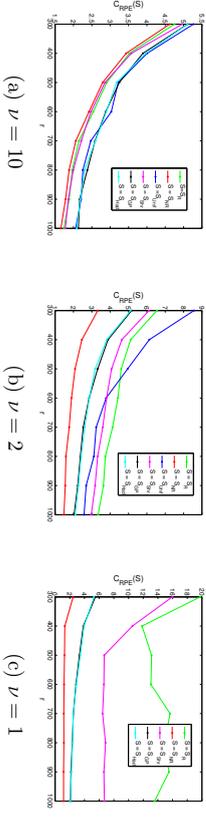


Figure 2: Relative prediction efficiency $CPE(S)$ for large r .

r	S_R	S_{NR}	S_{Unif}	S_{SIR}	S_{GCP}	S_{Had}
80	39.8	38.7	37.3	39.8	35.5	40.0
90	27.8	27.2	27.2	27.2	26.1	27.4
100	22.4	22.1	22.7	22.2	21.3	23.1
200	8.33	7.88	8.51	7.62	8.20	8.14

(a) $\nu = 10$

r	S_R	S_{NR}	S_{Unif}	S_{SIR}	S_{GCP}	S_{Had}
80	70.7	60.1	1.05×10^2	73.3	36.5	39.8
90	45.6	34.6	66.2	44.7	26.7	28.2
100	35.1	25.9	52.8	33.8	22.1	22.5
200	9.82	5.54	15.3	9.17	7.59	7.81

(b) $\nu = 2$

r	S_R	S_{NR}	S_{Unif}	S_{SIR}	S_{GCP}	S_{Had}
80	4.4×10^4	3.1×10^4	7.0×10^3	1.4×10^4	34.2	40.0
90	1.5×10^4	7.0×10^3	5.2×10^3	1.0×10^4	26.0	28.7
100	1.8×10^4	3.6×10^3	3.9×10^3	3.4×10^3	22.7	24.8
200	2.0×10^2	34.0	5.2×10^2	3.6×10^2	7.94	7.84

(c) $\nu = 1$

Figure 3: Relative prediction efficiency $CPE(S)$ for small r .

Overall, S_{NR} , S_R , and S_{SIR} compare very favorably to S_{Unif} , which is consistent with Theorem 2, since samples with higher leverage scores tend to reduce the mean-squared

error. Furthermore, S_R (which recall involves re-scaling) only increases the mean-squared error, which is again consistent with the theoretical results. The effects are more apparent as the leverage score distribution is more non-uniform (i.e., for $\nu = 1$).

The theoretical upper bound in Theorems 1-4 suggests that $C_{PE}(S)$ is of the order $\frac{n}{r}$, independent of the leverage scores of X , for $S = S_R$ as well as $S = S_{Had}$ and S_{GP} . On the other hand, the simulations suggest that for highly non-uniform leverage scores, $C_{PE}(S_R)$ is higher than when the leverage scores are uniform, whereas for $S = S_{Had}$ and S_{GP} , the non-uniformity of the leverage scores does not significantly affect the bounds. The reason that S_{Had} and S_{GP} are not significantly affected by the leverage-score distribution is that the Hadamard and Gaussian projection has the effect of making the leverage scores of any matrix uniform Drineas et al. (2011). The reason for the apparent disparity when $S = S_R$ is that the theoretical bounds use Markov's inequality which is a crude concentration bound. We suspect that a more refined analysis involving the bounded difference inequality would reflect that non-uniform leverage scores result in a larger $C_{PE}(S_R)$.

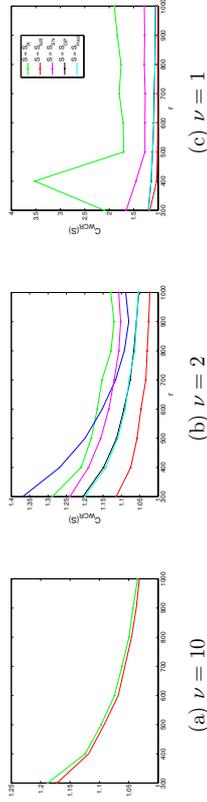


Figure 4: Worst-case relative error $C_{WC}(S)$ for large r .

Finally, Figures 4 and 5 provide a comparison of the worst-case relative error $C_{WCE}(S)$ for large and small ($r > 200$ and $r \leq 200$, respectively) values of r . Observe that, in general, $C_{WC}(S)$ are much closer to 1 than $C_{PE}(S)$ for all choices of S . This reflects the scaling of $\frac{2}{n}$ difference between the bounds. Interestingly, Figures 4 and 5 indicates that S_{NR} still tends to out-perform S_R in general, however the difference is not as significant as in the statistical setting.

5. Discussion and Conclusion

In this paper, we developed a framework for analyzing algorithmic and statistical criteria for general sketching matrices $S \in \mathbb{R}^{r \times n}$ applied to the least-squares objective. As our analysis makes clear, our framework reveals that the algorithmic and statistical criteria depend on different properties of the oblique projection matrix $\Pi_S^U = U(SU)^\dagger U$, where U is the left singular matrix for X . In particular, the algorithmic criteria (WC) depends on the quantity $\sup_{U^T \epsilon=0} \frac{\|\Pi_S^U \epsilon\|_2}{\|\epsilon\|_2}$, since in that case the data may be arbitrary and worst-case, whereas the two statistical criteria (RE and PE) depends on $\|\Pi_S^U\|_F$, since in that case the data follow a linear model with homogenous noise variance.

r	S_R	S_{NR}	S_{Unif}	S_{Shr}	S_{GP}	S_{Had}
80	2.82	2.78	2.94	2.82	2.74	2.89
90	2.34	2.32	2.40	2.37	2.24	2.33
100	2.04	2.01	2.09	2.06	2.02	2.03
200	1.33	1.31	1.36	1.32	1.34	1.34

(a) $\nu = 10$

r	S_R	S_{NR}	S_{Unif}	S_{Shr}	S_{GP}	S_{Had}
80	4.46	3.69	5.71	4.33	2.81	2.99
90	3.25	2.85	4.29	3.18	2.27	2.28
100	2.70	2.20	3.52	2.61	2.06	2.10
200	1.43	1.22	1.70	1.42	1.34	1.36

(b) $\nu = 2$

r	S_R	S_{NR}	S_{Unif}	S_{Shr}	S_{GP}	S_{Had}
80	6.0×10^9	6.0×10^{-0}	2.1×10^5	6.9×10^2	2.64	2.85
90	1.7×10^5	3.7×10^{-4}	2.4×10^5	5.0×10^2	2.35	2.35
100	9.1×10^4	4.5×10^{-4}	1.3×10^5	1.8×10^2	2.07	2.12
200	2.1×10^2	70.0	1.6×10^4	18.0	1.34	1.35

(c) $\nu = 1$

Figure 5: Worst-case relative error $C_{WC}(S)$ for large r .

Using our framework, we develop upper bounds for 3 performance criteria applied to 4 sketching schemes. Our upper bounds reveal that in the regime where $p < r \ll n$, our sketching schemes achieve optimal performance up to constants, in terms of WC and RE. On the other hand, the PE scales as $\frac{n}{r}$ meaning r needs to be close to (or greater than) n for good performance. Subsequent lower bounds in Pilanci and Wainwright (2014) show that this upper bound can not be improved, but subsequent work by Pilanci and Wainwright (2014) as well as Lu and Foster (2014); Lu et al. (2013) provide alternate more sophisticated sketching approaches to deal with these challenges. Our simulation results reveal that for when r is very close to p , projection-based approaches tend to out-perform sampling-based approaches since projection-based approaches tend to be more stable in that regime.

There are numerous ways in which the framework and results from this paper can be extended. Firstly, there is a large literature that presents a number of different approaches to sketching. Since our framework provides general conditions to assess the statistical and algorithmic performance for sketching matrices, a natural and straightforward extension would be to use our framework to compare other sketching matrices. Another natural extension is to determine whether aspects of the framework can be adapted to other statistical models and problems of interest (e.g., generalized linear models, covariance estimation, PCA, etc.). Finally, another important direction is to compare the stability and robustness properties of different sketching matrices. Our current analysis assumes a known linear model, and it is unclear how the sketching matrices behave under model mis-specification.

Acknowledgement. We would like to thank the Statistical and Applied Mathematical Sciences Institute and the members of its various working groups for helpful discussions.

Appendix A. Auxiliary Results

In this section, we provide proofs of Lemma 1 and an intermediate result we will later use to prove the main theorems.

A.1. Proof of Lemma 1

Recall that $X = U\Sigma V^T$, where $U \in \mathbb{R}^{n \times p}$, $\Sigma \in \mathbb{R}^{p \times p}$ and $V \in \mathbb{R}^{p \times p}$ denote the left singular matrix, diagonal singular value matrix and right singular matrix respectively.

First we show that $\|Y - X\beta_{OLS}\|_2^2 = \|\epsilon\|_2^2$. To do so, observe that

$$\|Y - X\beta_{OLS}\|_2^2 = \|Y - U\Sigma V^T\beta_{OLS}\|_2^2$$

and set $\delta_{OLS} = \Sigma V^T\beta_{OLS}$. It follows that $\delta_{OLS} = U^T Y$. Hence

$$\|Y - X\beta_{OLS}\|_2^2 = \|Y - \Pi_U Y\|_2^2$$

where $\Pi_U = UU^T$. For every $Y \in \mathbb{R}^n$, there exists a unique $\delta \in \mathbb{R}^p$ and $\epsilon \in \mathbb{R}^n$ such that $U^T \epsilon = 0$ and $Y = U\delta + \epsilon$. Hence

$$\|Y - X\beta_{OLS}\|_2^2 = \|(I_{n \times n} - \Pi_U)\epsilon\|_2^2 = \|\epsilon\|_2^2,$$

where the final equality holds since $\Pi_U \epsilon = 0$.

Now we analyze $\|Y - X\beta_S\|_2^2$. Observe that

$$\|Y - X\beta_S\|_2^2 = \|Y - \Pi_U^S Y\|_2^2,$$

where $\Pi_U^S = U(SU)^T S$. Since $Y = U\delta + \epsilon$, it follows that

$$\begin{aligned} \|Y - X\beta_S\|_2^2 &= \|U(I_{p \times p} - (SU)^T(SU)\delta + (I_{n \times n} - \Pi_U^S)\epsilon)\|_2^2 \\ &= \|(I_{p \times p} - (SU)^T(SU)\delta)\|_2^2 + \|(I_{n \times n} - \Pi_U^S)\epsilon\|_2^2 \\ &= \|(I_{p \times p} - (SU)^T(SU)\delta)\|_2^2 + \|\epsilon\|_2^2 + \|\Pi_U^S \epsilon\|_2^2. \end{aligned}$$

Therefore for all Y :

$$C_{WC}(S) = \frac{\|Y - X\beta_S\|_2^2}{\|Y - X\beta_{OLS}\|_2^2} = 1 + \frac{\|(I_{p \times p} - (SU)^T(SU)\delta)\|_2^2 + \|\Pi_U^S \epsilon\|_2^2}{\|\epsilon\|_2^2},$$

where $U^T \epsilon = 0$. Taking a supremum over Y and consequently over ϵ and δ completes the proof for $C_{WC}(S)$.

Now we turn to the proof for $C_{PE}(S)$. First note that

$$\mathbb{E}\|X(\beta_{OLS} - \beta)\|_2^2 = \mathbb{E}\|UU^T Y - U\Sigma V^T \beta\|_2^2.$$

Under the linear model $Y = U\Sigma V^T \beta + \epsilon$,

$$\mathbb{E}\|X(\beta_{OLS} - \beta)\|_2^2 = \mathbb{E}\|\Pi_U \epsilon\|_2^2.$$

Since $\mathbb{E}[\epsilon^T] = I_{n \times n}$, it follows that

$$\mathbb{E}\|X(\beta_{OLS} - \beta)\|_2^2 = \mathbb{E}\|\Pi_U \epsilon\|_2^2 = \|\Pi_U\|_F^2 = p.$$

For β_S , we have that

$$\begin{aligned} \mathbb{E}\|X(\beta_S - \beta)\|_2^2 &= \mathbb{E}\|\Pi_U^S Y - U\Sigma V^T \beta\|_2^2 = \mathbb{E}\|(U(I - (SU)^T(SU)\Sigma V^T \beta + \Pi_U^S \epsilon)\|_2^2 \\ &= \mathbb{E}\|(I_{p \times p} - (SU)^T(SU)\Sigma V^T \beta)\|_2^2 + \mathbb{E}\|\Pi_U^S \epsilon\|_2^2 \\ &= \mathbb{E}\|(I_{p \times p} - (SU)^T(SU)\Sigma V^T \beta)\|_2^2 + \|\Pi_U^S\|_F^2. \end{aligned}$$

Hence $C_{PE}(S) = 1/p(\|(I_{p \times p} - (SU)^T(SU)\Sigma V^T \beta)\|_2^2 + \|\Pi_U^S\|_F^2)$ as stated.

For $C_{RE}(S)$, the mean-squared error for δ_{OLS} and δ_S are

$$\begin{aligned} \mathbb{E}\|Y - X\beta_{OLS}\|_2^2 &= \mathbb{E}\|(I - \Pi_U^T)\epsilon\|_2^2 \\ &= \|I - \Pi_U^T\|_F^2 = n - p, \end{aligned}$$

and

$$\begin{aligned} \mathbb{E}\|Y - X\beta_S\|_2^2 &= \mathbb{E}\|(I_{p \times p} - (SU)^T(SU)\Sigma V^T \beta)\|_2^2 + \mathbb{E}\|(I - \Pi_U^S)\epsilon\|_2^2 \\ &= \mathbb{E}\|(I_{p \times p} - (SU)^T(SU)\Sigma V^T \beta)\|_2^2 + \text{trace}((I - \Pi_S)^T(I - \Pi_S)) \\ &= \mathbb{E}\|(I_{p \times p} - (SU)^T(SU)\Sigma V^T \beta)\|_2^2 + \text{trace}(I) - 2\text{trace}(\Pi_S) + \|\Pi_S\|_F^2 \\ &= \mathbb{E}\|(I_{p \times p} - (SU)^T(SU)\Sigma V^T \beta)\|_2^2 + n - 2p + \|\Pi_S\|_F^2 \\ &= \mathbb{E}\|(I_{p \times p} - (SU)^T(SU)\Sigma V^T \beta)\|_2^2 + n - p + \|\Pi_S\|_F^2 - p. \end{aligned}$$

Hence,

$$\begin{aligned} C_{RE}(S) &= \frac{n - p + \|(I_{p \times p} - (SU)^T(SU)\Sigma V^T \beta)\|_2^2 + \|\Pi_S\|_F^2 - p}{n - p} \\ &= 1 + \frac{\|(I_{p \times p} - (SU)^T(SU)\Sigma V^T \beta)\|_2^2 + \|\Pi_S\|_F^2 - p}{n - p} \\ &= 1 + \frac{C_{PE}(S) - 1}{n/p - 1}. \end{aligned}$$

A.2. Intermediate Result

In order to provide a convenient way to parameterize our upper bounds for $C_{WC}(S)$, $C_{PE}(S)$, and $C_{RE}(S)$, we introduce the following three structural conditions on S . Let $\hat{\sigma}_{\min}(A)$ denote the minimum *non-zero* singular value of a matrix A .

- The first condition is that there exists an $\alpha(S) > 0$ such that

$$\hat{\sigma}_{\min}(SU) \geq \alpha(S). \quad (12)$$

- The second condition is that there exists a $\beta(S)$ such that

$$\sup_{\epsilon, U^T \epsilon = 0} \frac{\|U^T S^T S \epsilon\|_2}{\|\epsilon\|_2} \leq \beta(S). \quad (13)$$

- The third condition is that there exists a $\gamma(S)$ such that

$$\|U^T S^T S\|_F \leq \gamma(S). \quad (14)$$

Note that the structural conditions defined by $\alpha(S)$ and $\beta(S)$ have been defined previously as Eqn. (8) and Eqn. (9) in Drineas et al. (2011).

Given these quantities, we can state the following lemma, the proof of which may be found in Section A.2. This lemma provides upper bounds for $C_{WC}(S)$, $C_{PE}(S)$, and $C_{RE}(S)$ in terms of the parameters $\alpha(S)$, $\beta(S)$, and $\gamma(S)$.

Lemma 2 For $\alpha(S)$ and $\beta(S)$, as defined in Eqn. (12) and (13),

$$C_{WC}(S) \leq 1 + \sup_{\delta \in \mathbb{R}^p, U^T \epsilon = 0} \frac{\|(I_{p \times p} - (SU)^\dagger(SU))\delta\|_2}{\|\epsilon\|_2} + \frac{\beta^2(S)}{\alpha^4(S)}.$$

For $\alpha(S)$ and $\gamma(S)$, as defined in Eqn. (12) and (14),

$$C_{PE}(S) \leq \frac{\|(I_{p \times p} - (SU)^\dagger(SU))\Sigma V^T \beta\|_2}{p} + \frac{\gamma^2(S)}{\alpha^4(S)}.$$

Furthermore,

$$C_{RE}(S) \leq 1 + \frac{p}{n} \left[\frac{\|(I_{p \times p} - (SU)^\dagger(SU))\Sigma V^T \beta\|_2}{p} + \frac{\gamma^2(S)}{\alpha^4(S)} \right].$$

Again, the terms involving $(SU)^\dagger S U$ are a ‘‘bias’’ that equal zero for rank-preserving sketching matrices. In addition, we emphasize that the results of Lemma 1 and Lemma 2 hold for arbitrary sketching matrices S . In Appendix A.3, we bound $\alpha(S)$, $\beta(S)$ and $\gamma(S)$ for several different randomized sketching matrices, and this will permit us to obtain bounds on $C_{WC}(S)$, $C_{PE}(S)$, and $C_{RE}(S)$. For the sketching matrices we analyze, we prove that the bias term is 0 with high probability.

A.3 Proof of Lemma 2

Note that $\Pi_S^U = U(SU)^\dagger S$. Let $\text{rank}(SU) = k < p$, and the singular value decomposition is $SU = \tilde{U}\tilde{\Sigma}\tilde{V}^T$, where $\tilde{\Sigma} \in \mathbb{R}^{k \times k}$ is a diagonal matrix with non-zero singular values of SU . Then,

$$\begin{aligned} \frac{\|\Pi_S^U \epsilon\|_2}{\|\epsilon\|_2} &= \frac{\|U(SU)^\dagger S \epsilon\|_2}{\|\epsilon\|_2} = \frac{\|(SU)^\dagger S \epsilon\|_2}{\|\epsilon\|_2} \\ &= \frac{\|\tilde{V}\tilde{\Sigma}^{-1}\tilde{U}^T S \epsilon\|_2}{\|\epsilon\|_2} \\ &= \frac{\|\tilde{V}\tilde{\Sigma}^{-2}\tilde{V}^T \tilde{V}\tilde{\Sigma}\tilde{U}^T S \epsilon\|_2}{\|\epsilon\|_2}, \end{aligned}$$

where we have ignored the bias term which remains unchanged. Note that $\tilde{V}\tilde{\Sigma}^{-2}\tilde{V}^T \succeq \alpha^{-2}(S)I_{p \times p}$ and $\tilde{V}\tilde{\Sigma}\tilde{U}^T = (SU)^T = U^T S^T$. Hence,

$$\begin{aligned} \frac{\|\Pi_S^U \epsilon\|_2}{\|\epsilon\|_2} &= \frac{\|\tilde{V}\tilde{\Sigma}^{-2}\tilde{V}^T \tilde{V}\tilde{\Sigma}\tilde{U}^T S \epsilon\|_2}{\|\epsilon\|_2} \\ &\leq \frac{\|U^T S^T S \epsilon\|_2}{\alpha^4(S)\|\epsilon\|_2} \\ &= \frac{\beta^2(S)}{\alpha^4(S)\|\epsilon\|_2}, \end{aligned}$$

and the upper bound on $C_{WC}(S)$ follows.

Similarly,

$$\|\Pi_S^U\|_F^2 = \|U(SU)^\dagger S\|_F^2 = \|(SU)^\dagger S\|_F^2 = \|\tilde{V}\tilde{\Sigma}^{-2}\tilde{V}^T \tilde{V}\tilde{\Sigma}\tilde{U}^T S\|_F^2 \leq \alpha^{-4}(S)\|U^T S^T S\|_F^2$$

and the upper bound on $C_{PE}(S)$ follows.

Appendix B. Proof of Main Theorems

The proof techniques for all of four theorems are similar, in that we use the intermediate result Lemma 2 and bound the expectations of $\alpha(S)$, $\beta(S)$, and $\gamma(S)$ for each S , then apply Markov’s inequality to develop high probability bounds.

B.1 Proof of Theorem 1

First we bound $\alpha^2(S_R)$ by using existing results in Drineas et al. (2011). In particular applying Theorem 4 in Drineas et al. (2011) with $\beta = 1 - \theta$, $A = U^T$, $\epsilon = \frac{1}{\sqrt{2}}$ and $\delta = 0.1$ provides the desired lower bound on $\alpha(S_R)$ and ensures that the ‘‘bias’’ term in Lemma 2 is 0 since $\text{rank}(S_R U) = p$.

To upper bound $\beta(S_R)$, we first upper bound its expectation, then apply Markov’s inequality. Using the result of Table 1 (second row) of Drineas et al. (2006a) with $\beta = 1 - \theta$:

$$\mathbb{E}[\|U^T S_R^T S_R \epsilon\|_2^2] \leq \frac{1}{(1-\theta)^r} \|U^T\|_F^2 \|\epsilon\|_2^2 = \frac{p}{(1-\theta)^r} \|\epsilon\|_2^2.$$

Applying Markov’s inequality,

$$\|U^T S_R^T S_R \epsilon\|_2^2 \leq \frac{11p}{(1-\theta)^r} \|\epsilon\|_2^2,$$

with probability at least 0.9.

Finally we bound $\gamma(S_R)$:

$$\begin{aligned} \frac{1}{p} \mathbb{E}[\|U^T S_R^T S_R\|_F^2] &= \frac{1}{p} [\text{trace}(U^T (S_R^T S_R)^2 U)] \\ &= \frac{1}{p} \sum_{j=1}^p \sum_{i=1}^n \sum_{k=1}^n U_{ij} U_{kj} [(S_R^T S_R)^2]_{ki} \\ &= \frac{1}{p} \sum_{j=1}^p \sum_{i=1}^n U_{ij}^2 [S_R^T S_R]_{ii}^2 \\ &= \frac{1}{p} \sum_{i=1}^n \ell_i |S_R^T S_R|_{ii}^2, \end{aligned}$$

where the second last equality follows since $[S_R^T S_R]_{ki}^2 = 0$ for $k \neq i$ and the final equality follows since $\ell_i = \sum_{j=1}^p U_{ij}^2$. First we upper bound $\mathbb{E}[\gamma(S_R)]$ and then apply Markov's inequality. Recall that $|S_R|_{ki} = \frac{1}{\sqrt{r}} \sigma_{ki}$ where $\mathbb{P}(\sigma_{ki} = \pm 1) = p_i$. Then,

$$\begin{aligned} \frac{1}{p} \sum_{i=1}^n \ell_i \mathbb{E}[(S_R^T S_R)_{ii}^2] &= \frac{1}{r^2 p} \sum_{i=1}^n \ell_i \sum_{m=1}^r \sum_{t=1}^r \mathbb{E}[\sigma_{mi}^2 \sigma_{ti}^2] \\ &= \frac{1}{r^2 p} \sum_{i=1}^n \ell_i \sum_{m=1}^r \sum_{t=1}^r \mathbb{E}[\sigma_{mi} \sigma_{ti}] \\ &= \frac{1}{r^2 p} \sum_{i=1}^n \ell_i \sum_{m=1}^r [(r^2 - r)p_i^2 + r p_i] \\ &= \frac{1}{r^2 p} \sum_{i=1}^n (\ell_i (r^2 - r) + r \frac{\ell_i}{p_i}) \\ &= 1 - \frac{1}{r} + \frac{1}{r p} \sum_{i=1}^n \frac{\ell_i}{p_i}. \end{aligned}$$

Substituting $p_i = (1 - \theta) \frac{\ell_i}{p} + \theta q_i$ completes the upper bound on $\mathbb{E}[\gamma(S_R)]$:

$$\begin{aligned} 1 - \frac{1}{r} + \frac{1}{r p} \sum_{i=1}^n \frac{\ell_i}{p_i} &= 1 - \frac{1}{r} + \frac{1}{r p} \sum_{i=1}^n \frac{\ell_i}{(1 - \theta) \frac{\ell_i}{p} + \theta q_i} \\ &\leq 1 - \frac{1}{r} + \frac{1}{r} \sum_{i=1}^n \frac{1}{1 - \theta} \\ &\leq 1 - \frac{1}{r} + \frac{1}{r} \frac{n}{(1 - \theta)^r} \\ &\leq 1 + \frac{1}{(1 - \theta)^r}. \end{aligned}$$

Using Markov's inequality,

$$\mathbb{P}(|\gamma(S_R) - \mathbb{E}[\gamma(S_R)]| \geq 10 \mathbb{E}[\gamma(S_R)]) \leq 0.1,$$

and consequently $\gamma(S_R) \leq 11(1 + \frac{n}{(1 - \theta)^r})$ with probability greater than 0.9. The final probability of 0.7 arises since we simultaneously require all three bounds to hold which hold with probability $0.9^3 > 0.7$. Applying Lemma 2 in combination with our high probability bounds for $\alpha(S_R)$, $\beta(S_R)$ and $\gamma(S_R)$ completes the proof for Theorem 1.

B.2 Proof of Theorem 2

Define $\bar{S} = \sqrt{\frac{k}{n}} S_{NR}$ where S_{NR} is the sampling matrix without re-scaling. Recall the k -heavy litter leverage-score assumption. Since $\sum_{i=k+1}^n \ell_i \leq \frac{p}{10r}$, $\sum_{i=k+1}^n p_i \leq \frac{1}{10r}$ (recall $p_i = \frac{\ell_i}{p}$). Hence the probability that a sample only contains the k samples with high leverage score is:

$$\left(1 - \frac{1}{10r}\right)^r \geq 1 - \frac{1}{10} = 0.9.$$

For the remainder of the proof, we condition on the event \mathcal{A} that only the rows with the k largest leverage scores are selected. Let $\tilde{U} \in \mathbb{R}^{k \times p}$ be the sub-matrix of U corresponding to the top k leverage scores.

Let $W = \mathbb{E}[\tilde{S}^T \tilde{S}] \in \mathbb{R}^{k \times k}$. Since $\frac{k}{n} \leq p_i \leq \frac{c}{k}$ for all $1 \leq i \leq k$, $c I_{k \times k} \preceq W \preceq C I_{k \times k}$. Furthermore since $\sum_{i=k+1}^n \ell_i \leq \frac{p}{10r}$, $0.9 I_{p \times p} \preceq U^T U \preceq I_{p \times p}$.

First we lower bound $\alpha^2(S_{NR})$. Applying Theorem 4 in Drineas et al. (2011) with $\beta = C$, $A = \tilde{U}^T W^{1/2}$, $\epsilon = \frac{c}{2}$ and $\delta = 0.1$ ensures that as long as $r \geq c' p \log(p)$ for sufficiently large c' ,

$$\|\tilde{U}^T W \tilde{U} - \tilde{U}^T \tilde{S}^T \tilde{S} \tilde{U}\|_{op} \leq \frac{c}{2C},$$

with probability at least 0.9. Since $\tilde{U}^T W \tilde{U} \succeq \frac{3c}{4}$, $\tilde{U}^T \tilde{S}^T \tilde{S} \tilde{U} \succeq \frac{c}{4}$. Therefore with probability at least 0.9,

$$\alpha^2(S_{NR}) \geq \frac{c^r}{4C^r}.$$

Next we bound $\beta(S_{NR})$. Since $\bar{S} = \sqrt{\frac{k}{n}} S_{NR}$, if we condition on \mathcal{A} , only the leading k leverage scores are selected and let $\tilde{U} \in \mathbb{R}^{k \times p}$ be the sub-matrix of U corresponding to the top k leverage scores. Using the result of Table 1 (second row) of Drineas et al. (2006a) with $\beta = 1$:

$$\mathbb{E}[\|U^T S_{NR}^T S_{NR} \epsilon\|_2^2] = \frac{r^2}{k^2} \mathbb{E}[\|U^T \tilde{S}^T \tilde{S} \epsilon\|_2^2] = \frac{r^2}{k^2} \mathbb{E}[\|\tilde{U}^T \tilde{S}^T \tilde{S} \epsilon\|_2^2] \leq \frac{r^2}{k^2} \|\tilde{U}^T\|_2 \|\epsilon\|_2 \leq \frac{p r^2}{k^2} \|\epsilon\|_2^2.$$

Applying Markov's inequality,

$$\|U^T S_{NR}^T S_{NR} \epsilon\|_2^2 \leq \frac{11 p r^2}{k^2} \|\epsilon\|_2^2.$$

with probability at least 0.9 which completes the upper bound for $\beta(S_{NR})$.

Finally we bound $\gamma(S_{NR})$:

$$\begin{aligned} \text{trace}(\bar{U}^T (\bar{S}^T \bar{S})^2 \bar{U})/p &= \frac{1}{p} \sum_{i=1}^n [\bar{S}^T \bar{S}]_{ii}^2 \sum_{j=1}^r \bar{U}_{ij}^2 \\ &= \frac{1}{p} \sum_{i=1}^k \ell_i [\bar{S}^T \bar{S}]_{ii}^2 \\ &\leq \frac{Ck}{r} \sum_{i=1}^k \frac{1}{r} \left(\sum_{m=1}^r \sigma_{mi}^2 \right), \end{aligned}$$

where the last step follows since $\ell_i \leq \frac{Cp}{k}$ for $1 \leq i \leq k$ and 0 otherwise. Now taking expectations:

$$\begin{aligned} \frac{1}{r} \mathbb{E} \left[\sum_{i=1}^k \left(\sum_{m=1}^r \sigma_{mi}^2 \right)^2 \right] &= \frac{1}{r} \sum_{i=1}^k \sum_{\ell=1}^r \sum_{m=1}^r \mathbb{E}[\sigma_{\ell i} \sigma_{m i}] \\ &\leq \frac{C^2}{r} \sum_{i=1}^k \left(\frac{r^2}{k^2} - \frac{r}{k} + \frac{r}{k} \right) \\ &= C^2 \left(\frac{r-1}{k} + 1 \right) \\ &\leq C^2 \left(\frac{r}{k} + 1 \right). \end{aligned}$$

Since $S_{NR} = \sqrt{\frac{k}{r}} \bar{S}$, $\mathbb{E}[\text{trace}(U^T (S_{NR}^T S_{NR})^2 U)/p] \leq C^2(1 + \frac{k}{r})$. Applying Markov's inequality,

$$\text{trace}(U^T (S_{NR}^T S_{NR})^2 U)/p \leq 11C^2(1 + \frac{k}{r}),$$

with probability at least 0.9. The probability of 0.6 arises since $0.9^4 > 0.6$. Again, using Lemma 2 completes the proof for Theorem 2.

B.3 Proof of Theorem 3

First we bound the smallest singular value of $S_{SGP}(U)$. Using standard results for bounds on the eigenvalues of sub-Gaussian matrices (see Proposition 2.4 in Rudelson and Vershynin (2009)), each entry of $A \in \mathbb{R}^{r \times n}$ is an i.i.d. zero-mean sub-Gaussian matrix:

$$\mathbb{P} \left(\inf_{\|x\|_2=1} \frac{1}{\sqrt{r}} \|Ax\|_2 \leq \frac{1}{\sqrt{2}} \right) \leq n \exp(-cr).$$

Hence, provided $cr \geq 2 \log n$,

$$\alpha(S_{SGP}) \geq \frac{1}{\sqrt{2}},$$

with probability greater than $1 - c \exp(-cr)$.

Next we bound $\beta(S_{SGP})$. Since $U^T \epsilon = 0$, $\|U^T S_{SGP}^T S_{SGP} \epsilon\|_2^2 = \|U^T (S_{SGP}^T S_{SGP} - I_{n \times n}) \epsilon\|_2^2$. Therefore

$$\begin{aligned} \|U^T S_{SGP}^T S_{SGP} \epsilon\|_2^2 &= \sum_{j=1}^p \left(\sum_{i=1}^n \sum_{k=1}^n U_{ij} (S_{SGP}^T S_{SGP} - I_{n \times n})_{ik} \epsilon_k \right)^2 \\ &= \sum_{j=1}^p \sum_{i=1}^n \sum_{k=1}^n \sum_{m=1}^n U_{ij} U_{mj} (S_{SGP}^T S_{SGP} - I_{n \times n})_{ik} (S_{SGP}^T S_{SGP} - I_{n \times n})_{m\ell} \epsilon_k \epsilon_\ell. \end{aligned}$$

First we bound $\mathbb{E}[\|U^T S_{SGP}^T S_{SGP} \epsilon\|_2^2]$.

$$\mathbb{E}[\|U^T S_{SGP}^T S_{SGP} \epsilon\|_2^2] = \sum_{j=1}^p \sum_{i=1}^n \sum_{k=1}^n \sum_{m=1}^n U_{ij} U_{mj} \mathbb{E}[(S_{SGP}^T S_{SGP} - I_{n \times n})_{ik} (S_{SGP}^T S_{SGP} - I_{n \times n})_{m\ell}] \epsilon_k \epsilon_\ell.$$

Recall that $S_{SGP} \in \mathbb{R}^{r \times n}$, $[S_{SGP}]_{si} = \frac{\lambda_{si}}{\sqrt{r}}$ where X_{si} are i.i.d. sub-Gaussian random variables with mean 0 and sub-Gaussian parameter 1. Hence

$$\mathbb{E}[(S_{SGP}^T S_{SGP} - I_{n \times n})_{ik} (S_{SGP}^T S_{SGP} - I_{n \times n})_{m\ell}] = \frac{1}{r} (\mathbb{I}(i=k) \mathbb{I}(\ell=m) + \mathbb{E}[X_i X_j X_k X_m]),$$

where X_i, X_k, X_ℓ and X_m are i.i.d. sub-Gaussian random variables. Therefore

$$\mathbb{E}[\|U^T S_{SGP}^T S_{SGP} \epsilon\|_2^2] = \frac{1}{r} \sum_{j=1}^p \sum_{i=1}^n \sum_{k=1}^n \sum_{m=1}^n U_{ij} U_{mj} (\mathbb{I}(i=k) \mathbb{I}(\ell=m) + \mathbb{E}[X_i X_k X_\ell X_m]) \epsilon_k \epsilon_\ell.$$

First note that $\mathbb{I}(i=k) \mathbb{I}(\ell=m) + \mathbb{E}[X_i X_k X_\ell X_m] = 0$ unless $i=k$ and $\ell=m$, or $i=\ell$ and $k=\ell$ or any other combination of two pairs of variables have the same index. When $i=k=\ell=m$, $\mathbb{I}(i=k) \mathbb{I}(\ell=m) + \mathbb{E}[X_i X_k X_\ell X_m] = 1 + \mathbb{E}[X_i^4] \leq 2$, since for sub-Gaussian random variables with parameter 1, $\mathbb{E}[X_i^4] \leq 1$ and

$$\begin{aligned} \frac{1}{r} \sum_{j=1}^p \sum_{i=1}^n \sum_{k=1}^n \sum_{m=1}^n U_{ij} U_{mj} (\mathbb{I}(i=k) \mathbb{I}(\ell=m) + \mathbb{E}[X_i X_k X_\ell X_m]) \epsilon_k \epsilon_\ell &= \frac{1}{r} \sum_{j=1}^p \sum_{i=1}^n U_{ij}^2 \epsilon_i^2 (1 + \mathbb{E}[X_i^4]) \\ &\leq \frac{2}{r} \sum_{j=1}^p \sum_{i=1}^n U_{ij}^2 \epsilon_i^2 \\ &\leq \frac{2}{r} \sum_{j=1}^p U_{ij}^2 \|\epsilon\|_2^2 \\ &= \frac{2p}{r} \|\epsilon\|_2^2. \end{aligned}$$

When $i=k$ and $\ell=m$ but $k \neq \ell$, $\mathbb{I}(i=k) \mathbb{I}(\ell=m) + \mathbb{E}[X_i X_k X_\ell X_m] = 2$ and

$$\sum_{j=1}^p \sum_{i=1}^n \sum_{k=1}^n \sum_{m=1}^n U_{ij} U_{mj} (\mathbb{I}(i=k) \mathbb{I}(\ell=m) + \mathbb{E}[X_i X_k X_\ell X_m]) \epsilon_k \epsilon_\ell = 2 \sum_{j=1}^p \sum_{i=1}^n \sum_{k=1}^n U_{ij} U_{mj} \epsilon_i \epsilon_m = 0,$$

since $U^T \epsilon = 0$. Using similar logic when the two pairs of variables are not identical, the sum is 0, and hence

$$\mathbb{E} \|U^T S_{SGP}^T S_{SGP} \epsilon\|_2^2 \leq \frac{2p}{r} \|\epsilon\|_2^2,$$

for all ϵ such that $U^T \epsilon = 0$. Applying Markov's inequality,

$$\|U^T S_{SGP}^T S_{SGP} \epsilon\|_2^2 \leq \frac{22p}{r} \|\epsilon\|_2^2,$$

with probability greater than 0.9. Therefore $\beta(S_{SGP}) \leq \sqrt{\frac{22p}{r}}$ with probability at least 0.9.

Now we bound $\gamma(S_{SGP}) = \|U^T S_{SGP}^T S_{SGP}\|_F^2 = \text{trace}(U^T (S_{SGP}^T S_{SGP})^2 U)/p$:

$$\begin{aligned} \text{trace}(U^T (S_{SGP}^T S_{SGP})^2 U)/p &= \frac{1}{p} \sum_{j=1}^p \sum_{i=1}^n \sum_{k=1}^n [(S_{SGP}^T S_{SGP})^2]_{ik} U_j U_i U_k \\ &= \frac{1}{pr^2} \sum_{j=1}^p \sum_{i=1}^n \sum_{k=1}^n \sum_{v=1}^r \sum_{m=1}^r U_j U_{ij} U_{kj} \sum_{v=1}^r \sum_{m=1}^r U_{ij} U_{kv} X_{mv} X_{mi} X_{lv} X_{lk} \end{aligned}$$

First we bound the expectation:

$$\begin{aligned} \mathbb{E}[\text{trace}(U^T (S_{SGP}^T S_{SGP})^2 U)/p] &= \frac{1}{pr^2} \sum_{j=1}^p \sum_{i=1}^n \sum_{k=1}^n \sum_{v=1}^r \sum_{m=1}^r \mathbb{E}[X_{mv} X_{mi} X_{lv} X_{lk}] \\ &= \frac{1}{pr^2} \sum_{j=1}^p \sum_{i=1}^n \sum_{k=1}^n \sum_{v=1}^r (\sigma^2 - r) U_{ij} U_{kj} \mathbb{E}[X_{iv} X_{kv}] + r \mathbb{E}[X_v^2 X_i X_k] \\ &= \frac{1}{pr^2} \sum_{j=1}^p \sum_{i=1}^n (\sigma^2 - r) U_{ij}^2 (\mathbb{E}[X_i^2])^2 + \frac{1}{pr^2} \sum_{j=1}^p \sum_{i=1}^n \sum_{v=1}^r r U_{ij}^2 \mathbb{E}[X_v^2 X_i^2] \\ &= \frac{1}{pr^2} \sum_{j=1}^p \sum_{i=1}^n (\sigma^2 - r) U_{ij}^2 \sigma^4 + \frac{1}{pr^2} \sum_{j=1}^p \sum_{i=1}^n \sum_{v=1}^r r U_{ij}^2 \mathbb{E}[X_v^2 X_i^2] \\ &= 1 - \frac{1}{r} + \frac{\mu_4}{4r} + \frac{n-1}{r} \\ &\leq 1 + \frac{3}{r} + \frac{n-2}{r} \\ &= 1 + \frac{(n+1)}{r}. \end{aligned}$$

Applying Markov's inequality,

$$\gamma(S_{SGP}) \leq 11 \left(1 + \frac{(n+1)}{r}\right),$$

with probability at least 0.9. This completes the proof for Theorem 3.

B.4 Proof of Theorem 4

For $S = S_{Had}$ we use existing results in Drineas et al. (2011) to lower bound $\alpha^2(S_{Had})$ and $\beta(S_{Had})$ and then upper bound $\gamma(S_{Had})$. Using Lemma 4 in Drineas et al. (2011) provides the desired lower bound on $\alpha^2(S_{Had})$.

To upper bound $\beta(S_{Had})$, we use Lemma 5 in Drineas et al. (2011) which states that:

$$\|U^T S_{Had}^T S_{Had} \epsilon\|_2^2 \leq \frac{20d \log(40nd) \|\epsilon\|_2^2}{r},$$

with probability at least 0.9.

Finally to bound $\gamma(S_{Had})$ recall that $S_{Had} = S_{Unif} H D$, where S_{Unif} is the uniform sampling matrix, D is a diagonal matrix with ± 1 entries and H is the Hadamard matrix:

$$\begin{aligned} \frac{1}{p} \|U^T S_{Had}^T S_{Had} \epsilon\|_2^2 &= \frac{1}{p} [\text{trace}(U^T (S_{Had}^T S_{Had})^2 U)] \\ &= \frac{1}{p} [\text{trace}(U^T (D^T H^T S_{Unif}^T S_{Unif} H D)^2 U)] \\ &= \frac{1}{p} [\text{trace}(U^T D^T H^T S_{Unif}^T S_{Unif} H D D^T H^T S_{Unif}^T S_{Unif} H D U)] \\ &= \frac{1}{p} [\text{trace}(U^T D^T H^T (S_{Unif}^T S_{Unif})^2 H D U)] \\ &= \frac{1}{p} \sum_{j=1}^p \sum_{i=1}^n \sum_{k=1}^n [H D U]_{ij} [H D U]_{kj} [(S_{Unif}^T S_{Unif})^2]_{ki} \\ &= \frac{1}{p} \sum_{j=1}^p \sum_{i=1}^n [H D U]_{ij}^2 [S_{Unif}^T S_{Unif}]_{ii}^2. \end{aligned}$$

Using Lemma 3 in Drineas et al. (2011), with probability greater than 0.95,

$$\frac{1}{p} \sum_{j=1}^p [H D U]_{ij}^2 \leq \frac{2 \log(40np)}{n}.$$

In addition, we have that

$$\begin{aligned} \frac{1}{p} \|U^T S_{Had}^T S_{Had} \epsilon\|_2^2 &= \frac{1}{p} \sum_{j=1}^p \sum_{i=1}^n [H D U]_{ij}^2 [S_{Unif}^T S_{Unif}]_{ii}^2 \\ &\leq \frac{2 \log(40np)}{n} \sum_{i=1}^n [S_{Unif}^T S_{Unif}]_{ii}^2. \end{aligned}$$

Now we bound $\mathbb{E} \sum_{i=1}^n [S_{Unif}^T S_{Unif}]_{ii}^2$.

$$\begin{aligned}
 \sum_{i=1}^n \mathbb{E}(|S_{\text{Unif}}^T S_{\text{Unif}} f_{\text{ii}}|^2) &= \frac{n^2}{r^2} \sum_{i=1}^n \sum_{m=1}^r \sum_{\ell=1}^r \mathbb{E}[\sigma_m^2 \sigma_\ell^2] \\
 &= \frac{n^2}{r^2} \sum_{i=1}^n \sum_{m=1}^r \sum_{\ell=1}^r \mathbb{E}[\sigma_m \sigma_\ell] \\
 &= \frac{n^2}{r^2} \sum_{i=1}^n \left[\frac{r^2 - r}{n^2} + \frac{r}{n} \right] \\
 &= \frac{n^2}{r^2} \left[\frac{r^2 - r}{n} + r \right] \\
 &= n - \frac{n}{r} + \frac{n^2}{r} \\
 &\leq n + \frac{n^2}{r}.
 \end{aligned}$$

Using Markov's inequality, with probability greater than 0.9,

$$\sum_{i=1}^n |S_{\text{Unif}}^T S_{\text{Unif}} f_{\text{ii}}|^2 \leq 10 \left(n + \frac{n^2}{r} \right),$$

which completes the proof.

References

H. Avron, P. Maymounkov, and S. Toledo. Blendenpik: Supercharging LAPACK's least-squares solver. *SIAM Journal on Scientific Computing*, 32:1217–1236, 2010.

C. Boutsidis and A. Gittens. Improved matrix algorithms via the subsampled randomized Hadamard transform. *SIAM Journal of Matrix Analysis and Applications*, 34:1301–1340, 2013.

S. Chatterjee and A. S. Hadi. Influential observations, high leverage point, and outliers in linear regression. *Statistical Science*, 1(3):379–416, 2006.

S. Chatterjee and A.S. Hadi. *Sensitivity Analysis in Linear Regression*. John Wiley & Sons, New York, 1988.

K. L. Clarkson and D. P. Woodruff. Low rank approximation and regression in input sparsity time. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 81–90, 2013.

P. Drineas, R. Kannan, and M. W. Mahoney. Fast Monte Carlo algorithms for matrices I: approximating matrix multiplication. *SIAM J. Comput.*, 36:132–157, 2006a.

P. Drineas, M.W. Mahoney, and S. Muthukrishnan. Sampling algorithms for ℓ_2 regression and applications. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1127–1136, 2006b.

P. Drineas, M. W. Mahoney, S. Muthukrishnan, and T. Sarlos. Faster least squares approximation. *Numerical Mathematics*, 117:219–249, 2011.

P. Drineas, M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff. Fast approximation of matrix coherence and statistical leverage. *Journal of Machine Learning Research*, 13: 3475–3506, 2012.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, 1996.

F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. John Wiley & Sons, New York, 1986.

A. Hedayat and W. D. Wallis. Hadamard matrices and their applications. *Annals of Statistics*, 6(6):1184–1238, 1978.

D.C. Hoaglin and R.E. Welsch. The hat matrix in regression and ANOVA. *The American Statistician*, 32(1):17–22, 1978.

P. J. Huber and E. M. Ronchetti. *Robust Statistics*. John Wiley & Sons, New York, 1981.

W. B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. volume 26, pages 189–206, 1984.

E. L. Lehmann. *Elements of Large-Sample Theory*. Springer Verlag, New York, 1998.

Y. Lu and D. P. Foster. Fast ridge regression with randomized principal component analysis and gradient descent. In *Proceedings of NIPS 2014*, 2014.

Y. Lu, P. S. Dhillon, D. P. Foster, and L. Ungar. Faster ridge regression via the subsampled randomized Hadamard transform. In *Proceedings of NIPS 2013*, 2013.

P. Ma, M. W. Mahoney, and B. Yu. A statistical perspective on algorithmic leveraging. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

P. Ma, M. W. Mahoney, and B. Yu. A statistical perspective on algorithmic leveraging. *Journal of Machine Learning Research*, 16:861–911, 2015.

M. W. Mahoney. *Randomized algorithms for matrices and data*. Foundations and Trends in Machine Learning. NOW Publishers, Boston, 2011. Also available at: arXiv:1104.5557.

M. W. Mahoney and P. Drineas. CUR matrix decompositions for improved data analysis. *Proc. Natl. Acad. Sci. USA*, 106(3):697–702, 2009.

X. Meng and M. W. Mahoney. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, pages 91–100, 2013.

X. Meng, M. A. Saunders, and M. W. Mahoney. LSRN: A parallel iterative solver for strongly over- or under-determined systems. *SIAM Journal on Scientific Computing*, 36(2):C95–C118, 2014.

- Carl D. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000.
- J. Nelson and N. L. Huy. OSNAP: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science*, pages 117–126, 2013.
- M. Piantoni and M. J. Wainwright. Iterative Hessian sketch: Fast and accurate solution approximation for constrained least-squares. Technical report, 2014. Preprint: arXiv:1411.0347.
- G. Raskutti and M. W. Mahoney. A statistical perspective on randomized sketching for ordinary least-squares. Technical report, 2014. Preprint: arXiv:1406.5986.
- G. Raskutti and M. W. Mahoney. Statistical and algorithmic perspectives on randomized sketching for ordinary least-squares. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.
- R. M. Royall. On finite population sampling theory under certain linear regression models. *Biometrika*, 57:377–387, 1970.
- M. Rudelson and R. Vershynin. Smallest singular value of a random rectangular matrix. *Communications on Pure and Applied Mathematics*, 62:1707–1739, 2009.
- A. M. Zaslavsky, H. Zheng, and J. Adams. Optimal sample allocation for design-consistent regression in a cancer services survey when design variables are known for aggregates. *Survey Methodology*, 34(1):65–78, 2008.

Learning Planar Ising Models

Jason K. Johnson
Numerica
Ft. Collins, CO, USA

Diane Oyen
Michael Chertkov
Los Alamos National Laboratory
Los Alamos, NM, USA

Praneeth Netrapalli
Microsoft Research
Cambridge, MA, USA

JASON.JOHNSON@NUMERICA.US

DOYEN@LANL.GOV
CHERTKOV@LANL.GOV

PRANEETH@MICROSOFT.COM

Editor: Manfred Opper

Abstract

Inference and learning of graphical models are both well-studied problems in statistics and machine learning that have found many applications in science and engineering. However, exact inference is intractable in general graphical models, which suggests the problem of seeking the best approximation to a collection of random variables within some tractable family of graphical models. In this paper, we focus on the class of planar Ising models, for which exact inference is tractable using techniques of statistical physics. Based on these techniques and recent methods for planarity testing and planar embedding, we propose a greedy algorithm for learning the best planar Ising model to approximate an arbitrary collection of binary random variables (possibly from sample data). Given the set of all pairwise correlations among variables, we select a planar graph and optimal planar Ising model defined on this graph to best approximate that set of correlations. We demonstrate our method in simulations and for two applications: modeling senate voting records and identifying geo-chemical depth trends from Mars rover data.

Keywords: Ising models, graphical models

1. Introduction

Graphical models are widely used to represent the statistical relations among a set of random variables (Lauritzen, 1996; MacKay, 2003). Nodes of the graph correspond to random variables and edges of the graph represent statistical interactions among the variables. The problems of inference and learning on graphical models arise in many practical applications. The problem of inference is to deduce certain statistical properties (such as marginal probabilities, modes etc.) of a given set of random variables whose graphical model is known. Inference has wide applications in areas such as error correcting codes, statistical physics and so on. The problem of learning on the other hand is to deduce the graphical model of a set of random variables given statistics (possibly from samples) of the random variables. Learning is also a widely encountered problem in areas such as biology, neuroscience and so on (Barabasi and Oltvai, 2004; Smith et al., 2011).

The *Ising model*, a class of binary-variable graphical models with pairwise interactions, has been studied by physicists as a simple model of order-disorder transitions in magnetic materials (Onsager, 1944). Remarkably, it was found that in the special case of an Ising model with zero-mean $\{-1, +1\}$ binary random variables and pairwise interactions defined on a planar graph, calculation of the partition function (which is closely tied to inference) is tractable, essentially reducing to calculation of a matrix determinant (Kac and Ward, 1952; Sherman, 1960; Kasteleyn, 1963; Fisher, 1966). Planar graph inference methods have been used in machine learning for efficient inference on planar graphs (Schraudolph and Kamenetsky, 2008; Gómez et al., 2010), in approximating inference for general graphs with planar graph decomposition (Jaakkola and T., 2007); and applied to problems such as computer vision (Batra et al., 2010; Yarkony et al., 2012) and financial forecasting (Pozzi et al., 2013), usually as an approximation method for problems with non-binary data.

We address the problem of approximating a collection of binary random variables (given their pairwise marginal distributions) by a zero-mean planar Ising model. We also consider the related problem of selecting a non-zero mean Ising model defined on an outer-planar graph (these models are also tractable, being essentially equivalent to a zero-field model on a related planar graph).

There has been a great deal of work on learning graphical models. Much of these have focused on learning over the class of thin graphical models (Deshpande et al., 2001; Bach and Jordan, 2001; Karger and Srebro, 2001; Shahaf et al., 2009) for which inference is tractable by converting the model to a junction tree. The simplest case of this is learning tree models (treewidth one graphs) for which it is tractable to find the best tree model by reduction to a max-weight spanning tree problem (Chow and Liu, 1968). However, the problem of finding the best bounded-treewidth model is NP-hard for treewidths greater than two (Karger and Srebro, 2001), and so heuristic methods are used to select the graph structure (Deshpande et al., 2001; Karger and Srebro, 2001). Another popular method is to use convex optimization of the log-likelihood penalized by the ℓ_1 norm of parameters of the graphical model so as to promote sparsity (Banerjee et al., 2008; Lee et al., 2006). To go beyond low-treewidth graphs, such methods either focus on Gaussian graphical models or adopt a tractable approximation of the likelihood. Other methods learn only the graph structure itself (Ravikumar et al., 2010; Abbeel et al., 2006) and are often able to demonstrate asymptotic correctness of this estimate under appropriate conditions.

In contrast to existing approaches, this paper explores planarity as an alternative restriction on the model class, instead of low treewidth, to make learning tractable while maintaining a tractable model for inference, unlike unrestricted regularized approaches, while still providing a solution in which the number of edges learned is linear in the number of variables.

2. Preliminaries

We develop our notation and briefly review the necessary background theory on graph estimation, Ising models and exact inference in planar graphs.

2.1 Divergence and Likelihood

A graph represents a joint probability distribution over a collection of variables. Suppose we want to calculate how well a probability distribution Q approximates another probability distribution P (on the same sample space \mathcal{X}). For any two probability distributions P and Q on some sample space \mathcal{X} , we denote by $D(P, Q)$ the *Kullback-Leibler divergence* (or *relative entropy*) between P and Q as $D(P, Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)}$. The *log-likelihood function* is defined as $LL(P, Q) = \sum_{x \in \mathcal{X}} P(x) \log Q(x)$. The probability distribution in a family \mathcal{F} that maximizes the log-likelihood of a probability distribution P is called the *maximum-likelihood estimate* of P in \mathcal{F} , and this is equivalent to the *minimum-divergence projection* of P to \mathcal{F} , so that $P_{\mathcal{F}} = \arg \max_{Q \in \mathcal{F}} LL(P, Q) = \arg \min_{Q \in \mathcal{F}} D(P, Q)$.

2.2 Graphical Models and The Ising Model

We will be dealing with binary random variables throughout the paper. We write $P(x)$ to denote the probability distribution of a collection of random variables $x = (x_1, \dots, x_n)$. Unless otherwise stated, we work with undirected graphs $G = (V, E)$ with vertex (or node) set V and edges $\{i, j\} \in E \subset \binom{V}{2}$. For vertices $i, j \in V$ we write $G + ij$ to denote the graph $(V, E \cup \{i, j\})$. A *pairwise graphical model* is a probability distribution $P(x) = P(x_1, \dots, x_n)$ that is defined on a graph $G = (V, E)$ with vertices $V = \{1, \dots, n\}$ as

$$P(x) \propto \prod_{i \in V} \psi_i(x_i) \prod_{\{i, j\} \in E} \psi_{ij}(x_i, x_j) \\ \propto \exp \left\{ \sum_{i \in V} f_i(x_i) + \sum_{\{i, j\} \in E} f_{ij}(x_i, x_j) \right\}, \quad (1)$$

where $\psi_i, \psi_{ij} \geq 0$ are non-negative node and edge compatibility functions. For positive ψ 's, we may also represent $P(x)$ as a Gibbs distribution with potentials $f_i = \log \psi_i$ and $f_{ij} = \log \psi_{ij}$.

Definition 1 An Ising model on binary random variables $x = (x_1, \dots, x_n)$ and graph $G = (V, E)$ is the probability distribution defined by

$$P(x) = \frac{1}{Z(\theta)} \exp \left\{ \sum_{i \in V} \theta_i x_i + \sum_{\{i, j\} \in E} \theta_{ij} x_i x_j \right\}, \\ Z(\theta) = \sum_x \exp \left\{ \sum_{i \in V} \theta_i x_i + \sum_{\{i, j\} \in E} \theta_{ij} x_i x_j \right\},$$

where $x_i \in \{-1, 1\}$. The partition function $Z(\theta)$ serves to normalize the probability distribution.

Formally, this defines an *exponential family* $P_\theta(x) = \exp\{\theta^T \phi(x) - \Phi(\theta)\}$ (Barndorff-Nielsen, 1979; Wainwright and Jordan, 2008) based on sufficient statistics $\phi_i(x) = x_i, i \in V$ and $\phi_{ij}(x) = x_i x_j, \{i, j\} \in E$, parameters $\theta_i, i \in V$ and $\theta_{ij}, \{i, j\} \in E$ and moment

parameters $(\mu_i = \mathbb{E}[x_i], i \in V)$ and $(\mu_{ij} = \mathbb{E}[x_i x_j], \{i, j\} \in E)$. The function $\Phi(\theta) = \log Z(\theta)$ is a convex function of θ and has the moment generating properties: $\nabla \Phi(\theta) = \mathbb{E}_\theta[\phi(x)] = \mu$ and $\nabla^2 \Phi(\theta) = \mathbb{E}_\theta[(\phi(x) - \mu)(\phi(x) - \mu)^T]$.

In fact, any pairwise graphical model among binary variables can be represented as an Ising model:

$$\theta_i = \frac{1}{2} \sum_{x_i} x_i f_i(x_i) + \frac{1}{4} \sum_{\{i, j\} \in E} \sum_{x_i, x_j} x_i f_{ij}(x_i, x_j), \\ \theta_{ij} = \frac{1}{4} \sum_{x_i, x_j} x_i x_j f_{ij}(x_i, x_j).$$

The moments can be computed as: $\mu_i = \sum_{x_i} x_i P(x_i)$ and $\mu_{ij} = \sum_{x_i, x_j} x_i x_j P(x_i, x_j)$. Inversely, the marginals are computed by:

$$P(x_i) = \frac{1}{2} (1 + \mu_i x_i), \\ P(x_i, x_j) = \frac{1}{4} (1 + \mu_i x_i + \mu_j x_j + \mu_{ij} x_i x_j).$$

We will be especially concerned with the following sub-family of Ising models:

Definition 2 An Ising model is said to be zero-field if $\theta_i = 0$ for all $i \in V$. It is zero-mean if $\mu_i = 0$ ($P(x_i = \pm 1) = \frac{1}{2}$) for all $i \in V$.

The Ising model is zero-field if and only if it is zero-mean. Although the zero-field assumption appears very restrictive, a general Ising model can be represented as a zero-field model by adding one auxiliary variable node connected to every other node of the graph (Jaakkola and T., 2007). The parameters and moments of the two models are then related as follows:

Proposition 1 Consider the Ising model on $G = (V, E)$ with $V = \{1, \dots, n\}$, parameters $\{\theta_i\}$ and $\{\theta_{ij}\}$, moments $\{\mu_i\}$ and $\{\mu_{ij}\}$ and partition function Z . Let $\widehat{G} = (\widehat{V}, \widehat{E})$ denote the extended graph based on nodes $\widehat{V} = V \cup \{n+1\}$ with edges $\widehat{E} = E \cup \{\{i, n+1\}, i \in V\}$. We define a zero-field Ising model on \widehat{G} with parameters $\{\theta_{ij}\}$, moments $\{\widehat{\mu}_{ij}\}$ and partition function \widehat{Z} . If we set the parameters according to

$$\widehat{\theta}_{ij} = \begin{cases} \theta_i & \text{if } j = n+1 \\ \theta_{ij} & \text{otherwise} \end{cases}, \\ \widehat{\mu}_{ij} = \begin{cases} \mu_i & \text{if } j = n+1 \\ \mu_{ij} & \text{otherwise} \end{cases}.$$

then $\widehat{Z} = 2Z$ and

Thus, inference on the corresponding zero-field Ising model on the extended graph \widehat{G} is equivalent to inference on the (non-zero-field) Ising model defined on G . Proof given in Appendix A.

2.3 Inference for Planar Ising Models

The motivation for our paper is the following result on tractability of inference for the planar zero-field Ising model.

Definition 3 *A graph is planar if it may be embedded in the plane without any edge crossings.*

Moreover, it is known that any planar graph can be embedded such that all edges are drawn as straight lines.

Theorem 1 (Kac and Ward, 1952; Sherman, 1960; Loeb, 2010) *Let G be a planar graph with specified straight-line embedding in the plane and let $\phi_{i,j,k} \in [-\pi, +\pi]$ denote the clockwise rotation between the directed edges (i,j) and (j,k) . We define the matrix $W \in \mathbb{C}^{2|E| \times 2|E|}$ indexed by directed edges of the graph as follows: $W = AD$ where D is the diagonal matrix with $D_{i,j,i} = \tanh \theta_{ij} \triangleq w_{ij}$ and*

$$A_{i,j,kl} = \begin{cases} \exp(\frac{1}{2}\sqrt{-1}\phi_{ijl}), & j = k \text{ and } i \neq l \\ 0, & \text{otherwise.} \end{cases}$$

Then, the partition function of the zero-field planar Ising model is given by the Kac-Ward determinant formula:

$$Z = 2^n \left(\prod_{\{i,j\} \in E} \cosh \theta_{ij} \right) \det(I - W)^{\frac{1}{2}}.$$

Another related method for computing the Ising model partition function is based on counting perfect matchings of planar graphs (Kasteleyn, 1963; Fisher, 1966). Thus, calculating the partition function reduces to calculating the determinant of a matrix; therefore, using the generalized nested dissection algorithm to exploit sparsity of the matrix, the complexity of these calculations is $O(n^{3/2})$ (Lipton et al., 1979; Lipton and Tarjan, 1979; Galluccio et al., 2000). Thus, inference of the zero-field planar Ising model is tractable and scales well with problem size.

The gradient and Hessian of the log-partition function $\Phi(\theta) = \log Z(\theta)$ can also be calculated efficiently from the Kac-Ward determinant formula. Derivatives of $\Phi(\theta)$ recover the moment parameters of the exponential family model as $\nabla \Phi(\theta) = \mathbb{E}_\theta[\phi] = \mu$ (Barndorff-Nielsen, 1979; Wainwright and Jordan, 2008). Thus, inference of moments (and node and edge marginals) is tractable for the zero-field planar Ising model.

Proposition 2 *Let $\mu = \nabla \Phi(\theta)$, $H = \nabla^2 \Phi(\theta)$. Let $S = (I - W)^{-1}A$ and $T = (I + P)(S \circ S^T)(I + P^T)$ where A and W are defined as in Theorem 1, \circ denotes the element-wise product and P is the permutation matrix swapping the indices of the directed edges (i,j) and (j,i) . Then,*

$$\mu_{ij} = w_{ij} - \frac{1}{2}(1 - w_{ij}^2)(S_{ij,ij} + S_{ji,ji}) \quad ij = kl \\ H_{i,j,kl} = \begin{cases} 1 - \mu_{ij}^2, & ij = kl \\ -\frac{1}{2}(1 - w_{ij}^2)T_{i,j,kl}(1 - w_{kl}^2), & \text{otherwise.} \end{cases}$$

Calculating the full matrix S requires $O(n^3)$ calculations. However, to compute just the moments μ only the diagonal elements of S are needed. Then, using the generalized nested dissection method, inference of moments (edge-wise marginals) of the zero-field Ising model can be achieved with complexity $O(n^{3/2})$. Computing the full Hessian is more expensive, requiring $O(n^3)$ calculations.

2.3.1 INFERENCE FOR OUTER-PLANAR GRAPHICAL MODELS

We emphasize that the above calculations require both a planar graph G and a zero-field Ising model. Using the graphical transformation of Proposition 1, the latter zero-field condition may be relaxed but at the expense of adding an auxiliary node connected to all the other nodes. In general planar graphs G , the new graph \widehat{G} may not be planar and hence may not admit tractable inference calculations. However, for the subset of planar graphs where this transformation does preserve planarity inference is still tractable.

Definition 4 *A graph G is said to be outer-planar if there exists an embedding of G in the plane where all the nodes are on the outer face.*

In other words, the graph G is outer-planar if the extended graph \widehat{G} (defined by Proposition 1) is planar. Then, from Proposition 1 and Theorem 1 it follows that:

Proposition 3 (Jaakkola and T., 2007) *The partition function and moments of any outer-planar Ising graphical model (not necessarily zero-field) can be calculated efficiently. Hence, inference is tractable for any binary-variable graphical model with pairwise interactions defined on an outer-planar graph.*

This motivates the problem of learning outer-planar graphical models for a collection of (possibly non-zero mean) binary random variables.

3. Learning Planar Ising Models

This section addresses the main goals of the paper, which are two-fold:

1. Solving for the maximum-likelihood Ising model on a given planar graph to best approximate a collection of zero-mean random variables.
2. Selecting heuristically the planar graph to obtain the best approximation.

We address these problems in the following two subsections. The solution of the first problem is an integral part of our approach to the second. Both solutions are easily adapted to the context of learning outer-planar graphical models of (possibly non-zero mean) binary random variables.

3.1 Maximum-Likelihood Parameter Estimation

Maximum-likelihood estimation over an exponential family is a convex optimization problem based on the log-partition function $\Phi(\theta)$. In the case of the zero-field Ising model defined on a given planar graph it is tractable to compute $\Phi(\theta)$ via a matrix determinant described in

Theorem 1. Thus, we obtain an unconstrained, tractable, convex optimization problem for the maximum-likelihood zero-field Ising model on the planar graph G to best approximate a probability distribution $P(x)$:

$$\max_{\theta} \{\mu^T \theta - \Phi(\theta)\} = \max_{\theta \in \mathbb{R}^{|E|}} \left\{ \sum_{ij} (\mu_{ij} \theta_{ij} - \log \cosh \theta_{ij}) - \frac{1}{2} \log \det(I - W(\theta)) \right\}.$$

Here, $\mu_{ij} = \mathbb{E}_P[x_i x_j]$ for all edges $\{i, j\} \in G$ and the matrix $W(\theta)$ is as defined in Theorem 1. If P represents the empirical distribution of a set of independent identically-distributed (iid) samples $\{x^{(s)}, s = 1, \dots, S\}$, then $\{\mu_{ij}\}$ are the corresponding empirical moments $\mu_{ij} = \frac{1}{S} \sum_s x_i^{(s)} x_j^{(s)}$.

3.1.1 NEWTON'S METHOD

We solve this unconstrained convex optimization problem using Newton's method with step-size chosen by back-tracking line search (Boyd and Vandenberghe, 2004). This produces a sequence of estimates $\theta^{(t)}$ calculated as follows:

$$\theta^{(t+1)} = \theta^{(t)} + \lambda_t H(\theta^{(t)})^{-1} (\mu(\theta^{(t)}) - \mu),$$

where $\mu(\theta^{(t)})$ and $H(\theta^{(t)})$ are calculated using Proposition 2 and $\lambda_t \in (0, 1]$ is a step-size parameter chosen by backtracking line search (see Boyd and Vandenberghe (2004): Chapter 9, Section 2 for details). The per iteration complexity of this optimization is $O(n^3)$ using explicit computation of the Hessian at each iteration. This complexity can be offset somewhat by only re-computing the Hessian a few times (reusing the same Hessian for a number of iterations), to take advantage of the fact that the gradient computation only requires $O(n^{\frac{3}{2}})$ calculations. As Newton's method has quadratic convergence, the number of iterations required to achieve a high-accuracy solution is typically 8-16 iterations (essentially independent of problem size). We estimate the computational complexity of solving this convex optimization problem as roughly $O(n^3)$.

3.2 Greedy Planar Graph Selection

We now consider the problem of selection of the planar graph G to best approximate a probability distribution $P(x)$ with pairwise moments $\mu_{ij} = \mathbb{E}_P[x_i x_j]$ given for all $i, j \in V$. Formally, we seek the planar graph that maximizes the log-likelihood (minimizes the divergence) relative to P :

$$\hat{G} = \arg \max_{G \in \mathcal{P}_V} LL(P, P_G) = \arg \max_{Q \in \mathcal{F}_G} LL(P, Q),$$

where \mathcal{P}_V is the set of planar graphs on the vertex set V , \mathcal{F}_G denotes the family of zero-field Ising models defined on graph G and $P_G = \arg \max_{Q \in \mathcal{F}_G} LL(P, Q)$ is the maximum-likelihood (minimum-divergence) approximation to P over this family.

We obtain a heuristic solution to this graph selection problem using the following greedy edge-selection procedure. The input to the algorithm is a probability distribution P (which could be empirical) on n binary $\{-1, 1\}$ random variables. In fact, it is sufficient to summarize P by its pairwise correlations $\mu_{ij} = \mathbb{E}_P[x_i x_j]$ on all pairs $i, j \in V$. The output is a

maximal planar graph G and the maximum-likelihood approximation θ_G to P in the family of zero-field Ising models defined on this graph. Note that a maximal planar graph is a planar graph for which no new edge can be added that would maintain planarity. Planar graphs are inherently sparse. All maximal planar graphs with $n > 2$ have $3n - 6$ edges¹.

Algorithm 1 GreedyPlanarGraphSelect(P)

```

1:  $G = \emptyset, \theta_G = 0$ 
2: for  $k = 1 : 3n - 6$  do
    $\triangleright$  Add edges until maximal planar graph reached
3:    $\Delta = \{\{i, j\} \subset V \mid \{i, j\} \notin G, G + ij \in \mathcal{P}_V\}$ 
    $\triangleright$  Set of edges that preserve planarity
4:    $\hat{\mu}_\Delta = \{\hat{\mu}_{ij} = \mathbb{E}_{P_G}[x_i x_j], \{i, j\} \in \Delta\}$ 
    $\triangleright$  Compute pairwise correlations
5:    $G \leftarrow G \cup \arg \max_{e \in \Delta} D(P_{e_i}, P_e)$ 
    $\triangleright$  Select edge that maximizes gain in log-likelihood
6:    $\theta_G = \text{PlanarIsing}(G, P)$ 
    $\triangleright$  Compute maximum-likelihood parameters for  $G$ 
7: end for

```

The algorithm starts with an empty graph and then sequentially adds edges to the graph one at a time so as to greedily increase the log-likelihood (decrease the divergence) relative to P as much as possible at each step. Here is a more detailed description of the algorithm along with estimates of the computational complexity of each step:

- *Line 3.* First, we enumerate the set Δ of all edges one might add (individually) to the graph while preserving planarity. This is accomplished by an $O(n^3)$ algorithm in which we iterate over all pairs $\{i, j\} \notin G$ and for each such pair we form the graph $G + ij$ and test planarity of this graph using known $O(n)$ algorithms (Chrobak and Payne, 1995).
- *Line 4.* Next, we perform tractable inference calculations with respect to the Ising model on G to calculate the pairwise correlations $\hat{\mu}_{ij}$ for all pairs $\{i, j\} \in \Delta$. This is accomplished using $O(n^{\frac{3}{2}})$ inference calculations on augmented versions of the graph G . For computational efficiency instead of calculating the addition of each proposed edge individually, moments of proposed edges are calculated in batches as follows. For each inference calculation we add as many edges to G from Δ as possible (setting $\theta = 0$ on these edges) while preserving planarity and then calculate all the edge-wise moments of this graph using Proposition 2 (including the zero-edges). This requires at most $O(n)$ iterations to cover all pairs of Δ , so the worst-case complexity to compute all required pairwise moments is $O(n^{\frac{5}{2}})$.
- *Line 5.* Once we have these moments, which specify the corresponding pairwise marginals of the current Ising model, we compare these moments (pairwise marginals) to those of the input distribution P by evaluating the pairwise KL-divergence between the Ising model and P . As seen by the following proposition, this gives us a lower-

¹ Euler's formula states that for a planar graph, $n - e + f = 2$, where f is the number of faces and e is the number of edges (Ritchson, 2012). The faces of a planar graph are simple polygons, so each face has at least 3 edges and each edge is part of at most 2 faces, $2e \leq 3f$. Combining these two properties, gives $e \leq 3n - 6$ for planar graphs. A maximal planar graph is triangulated, meaning that all faces have three sides, and therefore $e = 3n - 6$ (if any face has more than 3 sides, an edge can be added as a chord of that face polygon without breaking planarity).

bound on the improvement obtained by adding edge $\{i, j\}$ (see Appendix A for proof):

Proposition 4 Let P_G and P_{G+ij} be projections of P on G and $G + ij$ respectively.

Then,

$$D(P, P_G) - D(P, P_{G+ij}) \geq D(P(x_i, x_j), P_G(x_i, x_j)),$$

where $P(x_i, x_j)$ and $P_G(x_i, x_j)$ represent the marginal distributions on x_i, x_j of probabilities P and P_G respectively.

Thus, we greedily select the next edge $\{i, j\}$ to add so as to maximize this lower-bound on the improvement measured by the increase on log-likelihood (this being equal to the decrease in KL-divergence).

- *Line 6.* Finally, we calculate the new maximum-likelihood parameters θ_G on the new graph $G \leftarrow G + ij$. This involves solving the convex optimization problem discussed in the preceding subsection, which requires $O(n^3)$ complexity. This step is necessary in order to subsequently calculate the pairwise moments $\tilde{\mu}$ which guide further edge-selection steps, and also to provide the final estimate.

We continue adding one edge at a time until a maximal planar graph (with $3n - 6$ edges) is obtained. Thus, the total complexity of our greedy algorithm for planar graph selection is $O(n^4)$.

3.2.1 NON-MAXIMAL PLANAR GRAPHS

Since adding an edge always improves the log-likelihood, the greedy algorithm always outputs a maximal planar graph. However, this might lead to over-fitting of the data especially when the input probability distribution is an empirical distribution. Note that at $3n - 6$ edges, the maximal planar graph is sparse and our empirical work indicates that over-fitting is often not an issue. In the case that over-fitting is a concern, we could terminate the algorithm when adding an edge to the graph would only improve the log-likelihood by less than some threshold γ . A data-driven search can be performed for a suitable value of this threshold so as to minimize some estimate of the generalization, such as in cross validation methods (Zhang, 1993). Or, one could use some heuristic value for γ based on the number of samples such as Akaike's information criterion (AIC) or Schwarz's Bayesian information criterion (BIC) (Akaike, 1974; Schwarz, 1978).

3.2.2 OUTER-PLANAR GRAPHS AND NON-ZERO MEANS

The greedy algorithm returns a zero-field Ising model (which has zero mean for all the random variables) defined on a planar graph. If the actual random variables are non-zero mean, this may not be desirable. For this case we may prefer to exactly model the means of each random variable but still retain tractability by restricting the greedy learning algorithm to select outer-planar graphs. This model faithfully represents the marginals of each random variable but at the cost of modeling fewer pairwise interactions among the variables.

This is equivalent to the following procedure. First, given the sample moments $\{\mu_i\}$ and $\{\mu_{ij}\}$ we convert these to an equivalent set of zero-mean moments $\hat{\mu}$ on the extended

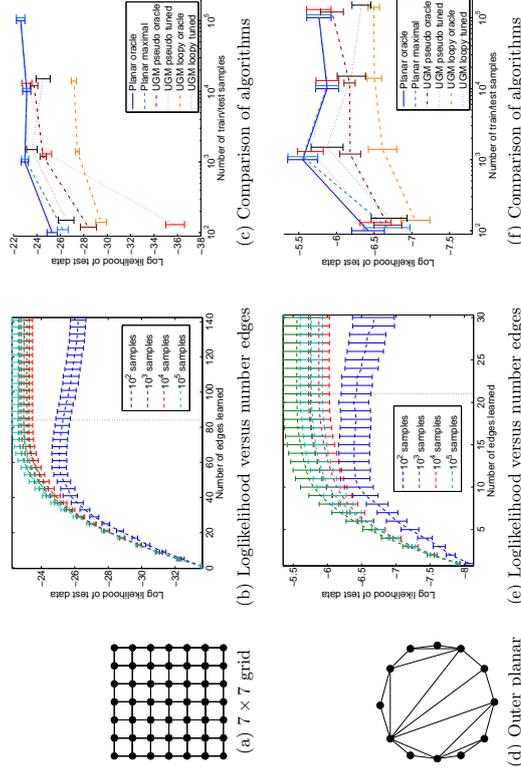


Figure 1: Results on known models, (top row): 7×7 grid; and (bottom row): outer planar. Left column (a,d): true graph. Middle column (b,e): likelihood of learned planar graphs as edges are added; the true number of edges is marked with a vertical dashed line. Right column (c,f): likelihood of test data for various algorithms; x-axis values are perturbed horizontally so that overlapping errorbars are visible.

vertex set $\hat{V} = V \cup \{n + 1\}$ according to Proposition 1. Then, we select a zero-mean planar Ising model for these moments using our greedy algorithm. However, to fit the means of each of the original n variables, we initialize this graph to include all the edges $\{i, n + 1\}$ for all $i \in V$ (requiring that these are present in our final estimate of the graph \hat{G}). After this initialization step, we use the same greedy edge-selection procedure as before. This yields the graph \hat{G} and parameters $\theta_{\hat{G}}$. Lastly, we convert back to a (non-zero field) Ising model on the subgraph of \hat{G} defined on nodes V , as prescribed by Proposition 1. The resulting graph G and parameters θ_G is our heuristic solution for the maximum-likelihood outer-planar Ising model.

We remark that it is not essential to choose between the zero-field planar Ising model and the outer-planar Ising model. The greedy algorithm may instead select something in between—a partial outer-planar Ising model where only nodes of the outer-face are allowed to have non-zero means. This is accomplished simply by omitting the initialization step of adding edges $\{i, n + 1\}$ for all $i \in V$.

4. Experiments

We present the results of experiments evaluating our algorithm on known models with simulated data to evaluate the correctness of the learned models. We generate two styles of known Ising models: a 7×7 grid ($n = 49$) with zero-field; and a 12-node outer planar model where nodes have non-zero mean; shown in Figures 1a and 1d. The edge parameters are chosen uniformly randomly between -1 and 1 with the condition that the absolute value be greater than a threshold (chosen to be 0.05) so as to avoid edges with negligible interactions. We use Gibbs sampling to obtain samples from this model and calculate empirical moments from these samples which are then passed as input to our algorithm. We run 10 trials of randomly generated edge parameters and data samples. Though our algorithm can run on graphs with many more nodes, we choose small examples here to illustrate the result effectively. On the outer planar model, we ensure that the first moments of all the nodes are satisfied by starting our algorithm with the auxiliary node connected to all other nodes.

As the planar learning algorithm adds edges to the model, the likelihood of the training data is guaranteed to increase. We assess how adding edges affects the likelihood of out-of-sample test data. Figures 1b and 1e demonstrate that likelihood on test sets generally increases as edges are added up to the maximal planar graph. The true number of edges in each synthetic graph is marked with a vertical dotted line. On the smallest data sets (100 samples) the out-of-sample performance begins to degrade, a sign of over-fitting the training data; yet the likelihood of the maximal graph is not significantly worse than the best likelihood obtained (with fewer edges).

We also compare against a Markov random field (MRF) learning algorithm for binary data (Schmidt et al., 2008), as implemented in the undirected graphical model learning Matlab package, UGMLearn². UGM is not restricted to learning planar graphs. The objective is optimized via projected gradient descent. We try two versions of the objective function, one using pseudo-likelihood and the other using loopy belief propagation for inference. UGM employs a regularization parameter which we set using two different methods. First, we used the *tuning* method on validation data as detailed in Schmidt et al. (2008). That is, we split the data into two parts, train on half the data using 7 different values for the parameter; measure the data likelihood of the other half of the data and vice-versa, then select the parameter value that maximizes the validation data likelihood across both folds. The learned model is trained on the full training data with the tuned regularization parameter value. The second method for setting the regularization parameter we call the *oracle* method, where we select the learned model at the true number of edges, k , in our known models. For UGM, we set the regularization parameter via linear search until k edges are learned. The likelihood of test data for the learned UGM model is calculated exactly when $n \leq 25$; in this case, the outer planar example. For larger graphs, e.g. the grid example, the likelihood of test data for UGM is approximated via loopy belief propagation, which we observed to converge well therefore providing a reasonable estimate.

We compare the likelihood of test data from the various learned models in Figures 1c and 1f. For comparison, we selected the maximal planar graph that our algorithm learns, **Planar maximal**; as well as the planar graph learned if the algorithm was stopped when the true number of edges are learned, **Planar oracle**. We compare against UGM **pseudo**

tuned and UGM **loopy tuned**, both of which tune the regularization parameter on validation data; but the former uses pseudo-likelihood in learning and the latter uses loopy belief propagation. The tuning method is the most common way of selecting the regularization parameter, but tends to produce relatively dense graphs. For fair comparison, we also show the likelihood of UGM **pseudo oracle** and UGM **loopy oracle**; that is, the model with the known true number of edges.

Figures 1c and 1f show that our greedy planar Ising model learning algorithm is at least as accurate and often better than the UGM learning algorithms on these inputs. As mentioned earlier, we see that **Planar maximal** and **Planar oracle** fit test data nearly equally well. On the outer planar model, UGM **pseudo tuned** performs nearly as well as our planar algorithm, yet on the larger grid model it performs quite poorly at the smaller sample sizes. UGM **loopy tuned** performs more consistently close to our planar algorithm, but it seems that loopy belief propagation performs worse at large sample sizes.

On the largest data set (10^5 samples) of the 7×7 grid model, UGM was aborted after running for 40 hours without reaching convergence on a single run, and so results are not available.

5. Applications

We apply our planar graph learning algorithm to real-world data in which there is no guarantee that the data is generated according to our model assumptions. The first application models voting patterns of the United States senate, while the second application models geological layers in rocks on Mars. We compare our learned planar graphs against the non-planar graph learning algorithm UGM, as described in the previous section. For the UGM learned graphs, the likelihood of test data is approximated via loopy belief propagation, which we observed to converge well therefore providing a reasonable estimate. Quantitative comparisons indicate that the learned planar models better predict held-out test data with sparser graphs than the non-planar graphs. We also discuss qualitative comparisons of the learned graphs.

5.1 Modeling Correlations of Senator Voting

We consider an interesting application of our algorithm to model correlations of senator voting following Banerjee et al. (2008). We use senator voting data from the years 2009 and 2010 to calculate correlations in the voting patterns among senators. A *Yea* vote is treated as $+1$ and a *Nay* vote is treated as -1 . We also treat non-votes as -1 , but only consider senators who voted in at least $\frac{2}{3}$ of the votes per year to limit bias. The data includes $n = 108$ variables and 645 samples. To accommodate the non-zero mean data we add an auxiliary node and allow the algorithm to select the connections between it and other nodes. We run a 10-fold cross-validation, training on 90% of the data and measuring likelihood on the held-out 10% of data. Figure 3 shows that the likelihood of test data increases as edges are added. We also show the likelihood of cross-validation test data for the UGM **pseudo** and UGM **loopy** algorithms for two different methods of choosing the value of the regularization parameter: (1) the value that produces the same number of edges as the maximal planar graph (at 318 edges); and (2) the value selected by tuning with validation data (at a variable number of edges, typically a dense graph). The likelihood of the sparse UGM models are

² <http://www.cs.nbc.ca/~mmp/hyk/Software/LICRF>

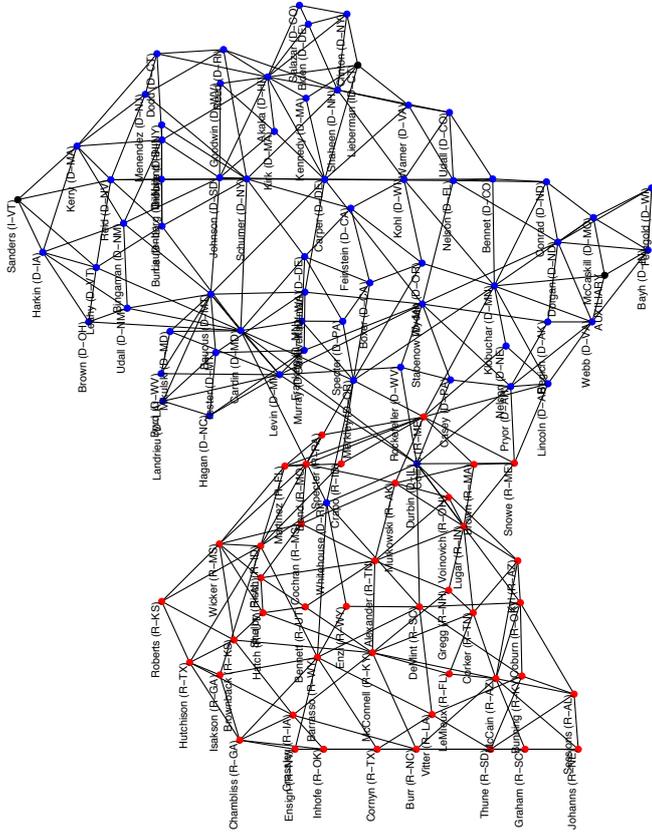


Figure 2: Senator voting results: Learned planar graphical model representing the senator voting pattern. Blue nodes represent Democrats, red nodes represent Republicans and black nodes represents Independents. We use a force-directed graph drawing algorithm (Fruchterman and Reingold, 1991).

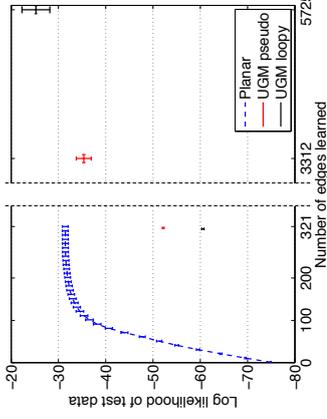


Figure 3: Senator voting results: Comparison of algorithms by likelihood of holdout data versus the number of edges in the learned graph. Note the break in the x-axis, due to tuned UGM learning dense graphs. On the tuned UGM models, we indicate standard error on number of edges learned.

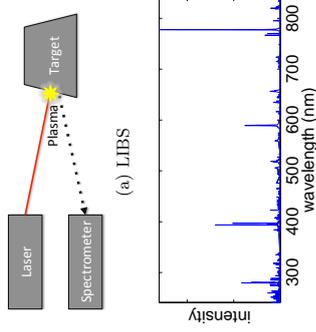


Figure 4: The ChemCam LIBS instrument fires a laser at a target, creating a plasma. Depending on its chemical composition, the plasma gives off different wavelengths of light which are measured by the spectrometer, producing an observed spectrum. Each shot ablates the target surface, leaving a small pit. Typically, sequences of 30 or more shots are fired in several locations on a single target.

5.2 Discovering Depth Trends in Rocks on Mars from Sample Correlations

Our second real-world data set consists of geological observations from the Mars rover *Curiosity*. We are interested in identifying correlations in chemical composition among spatially-related rock samples as taken from the Mars rover *Curiosity*. The ChemCam instrument onboard *Curiosity* collects observations of the chemical composition of rock targets using Laser-Induced Breakdown Spectrometry (LIBS) (Wiens et al., 2012). With each laser

significantly worse than the planar model. Only the UGM 1copy algorithm at a very dense (nearly fully connected) graph has better fit to test data.

The maximal planar graph learned from the full data set, shown in Figure 2, conveys many facts that are already known to us. For instance, the graph shows Sanders with edges only to Democrats which makes sense because he caucuses with Democrats. Same is the case with Lieberman. The graph also shows the senate minority leader McConnell well connected to other Republicans though the same is not true of the senate majority leader Reid. The learned UGM models can be seen in Appendix B, and they show that the non-planar models are qualitatively different, learning one or two densely connected components.

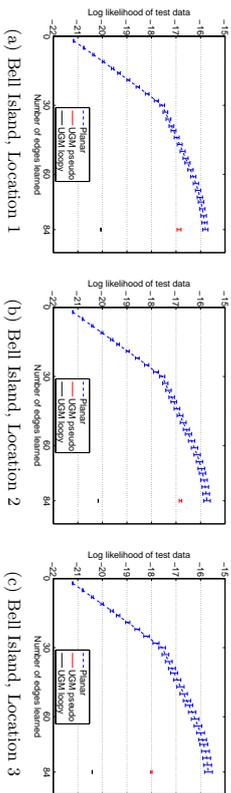


Figure 5: Comparison of algorithms on ChemCam data. Likelihood of holdout data versus the number of edges in the learned graph.

shot, the rock surface is ablated and therefore ChemCam produces a sequence of samples at increasing depth, potentially revealing compositional trends such as coatings, weathering rinds and thin stratigraphic layers that could give clues about the past atmospheric and aqueous conditions of Mars (Lanza et al., 2014).

We expect spatial correlations to exist among observations (samples), yet do not expect that the correlations will necessarily correspond to a fixed (known) grid; therefore, our planar model is a reasonable assumption to make. To test this, we compare against the non-planar UGM algorithm.

As shown in Figure 4, each LIBS shot produces a spectral observation consisting of 5810 wavelength bands between 224nm and 840nm. The spectral response is given as a table of intensity values for each wavelength band for each shot. A typical sequence of shots includes 30 - 150 shots on a fixed location. We model the correlations of rock chemistry among these shots, as measured by the set of wavelength bands that show non-zero response (above a noise threshold) in the observed spectra. More precisely, each spectrum is normalized, then thresholded so that 50% of the values are +1. To investigate shot-to-shot correlations, shots are the nodes in the graph while the 5810 wavelength bands are treated as samples. We add an auxiliary node and allow the algorithm to select the connections between it and other nodes. For comparison, we run a 10-fold cross validation using our planar model, the UGM model with pseudolikelihood and loopy³ belief propagation. Model selection for UGM is done by tuning with validation data and by comparing at the same number of edges learned by the planar algorithm.

We look at 30-shot depth sequences taken at one location at a time to find depth trends of interest. The rock is named Bell Island, and there are three locations that we investigated³. The graphs learned by our planar algorithm are quantitatively better than those learned by UGM, as shown in Figure 5. We compare the log-likelihood of 10-fold cross validation data for Planar, UGM pseudo and UGM loopy. With the number of edges in the graph fixed at $3n - 6$ (the number of edges in a maximal planar graph), we see that Planar achieves a significantly better fit to holdout data than UGM with either objective function. When we use the standard tuning method for UGM, it selects dense graphs that

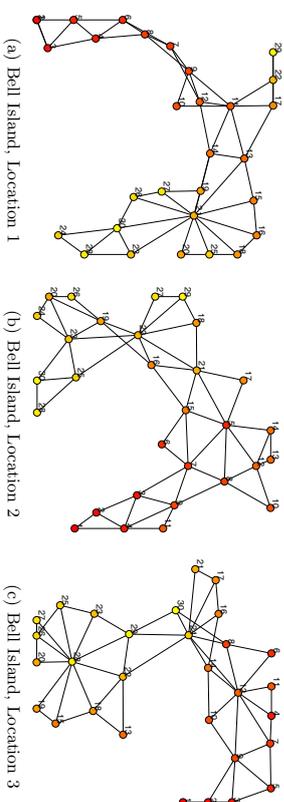


Figure 6: Learned planar graphs from ChemCam data. Nodes are numbered by LIBS laser shot number and color-coded by number starting with dark red at shot 1 and fading to light yellow at shot 30. We use a force-directed graph drawing algorithm (Fruchterman and Reingold, 1991).

are nearly fully-connected. While these tuned graphs learned by UGM do fit the data with higher log-likelihood, they provide no insight into the nature of depth trends.

Figure 6 shows the planar graphs learned from the Bell Island ChemCam data. Bell Island, Location 1 shows an interesting pattern of the first 10 or so shots being related to each other in ascending order, while the last 20 shots are more arbitrarily dependent. This pattern is consistent with the observation that the rock is covered in a layer of dust. As the laser ablates through the dust, each shot is conditionally dependent on the next shot. Once the rock itself is being sampled, the composition is more homogeneous. At Locations 2 and 3, there is less dust cover, and so there is less of a "tail" formed by the first few shots than seen at Location 1. However, in all of these graphs, we see that the graphs generally link earlier shots with early shots and later shots with late shots, despite not being given this ordering information. This indicates that the chemical composition of the rock is changing with depth.

6. Conclusion and Future Work

We provide a greedy heuristic to obtain the maximum-likelihood planar Ising model approximation to a collection of binary random variables with known pairwise marginals. The algorithm is simple to implement with the help of known methods for tractable exact inference in planar Ising models, efficient methods for planarity testing and embedding of planar graphs. While limiting the search to planar graphs, our learning model provides key advantages over arbitrary (non-planar) graph learning. Namely, the planar graph is sparse without necessitating a regularization term or tuning parameter. Also, the learned planar graph can be used for efficient inference of hidden values in future partially observed data. Further validating this approach, our empirical results on synthetic data and on real data indicate that our planar graph learning finds solutions that it are competitive with arbitrary (non-planar) graph learning in terms of fitting the distribution well.

³ NASA data is archived and available at <http://pds-geosciences.wustl.edu/missions/msl/chemcam.htm>.

Directions for further work are suggested by the methods and results of this paper. Firstly, we know that the greedy algorithm is not guaranteed to find the best planar graph. In Appendix C, we provide an enlightening counterexample in which the combination of the planarity restriction and greedy method prevent the correct model from being learned, because a strong indirect correlation exists that masks the correct combination of weaker direct correlations. That counterexample suggests strategies one might consider to further refine the estimate. One strategy would be to allow the greedy algorithm to prune edges which turn out to be less important once later edges are added. It would also be feasible to implement a multi-step greedy look-ahead search technique for selection of which edge to add (or prune) next.

Currently, our framework only allows learning planar graphical models on the set of observed random variables and requires that all variables are observed in each sample. One could imagine extensions of our approach to handle missing samples or to try to identify hidden variables that were not seen in the data. This concept offers another avenue to achieve a better fit to data that is not well-approximated by a planar graph among just the set of observed nodes, but might be well-approximated as the marginal distribution of a planar model with more nodes.

Acknowledgments

This work is approved for unlimited release as LA-UR-15-20740.

Appendix A. Proofs

Proof [Proposition 1] Let the probability distributions corresponding to G and \widehat{G} be P and \widehat{P} respectively and the corresponding expectations be \mathbb{E} and $\widehat{\mathbb{E}}$ respectively. For the partition function, we have that

$$\begin{aligned} \widehat{Z} &= \sum_{x_{\widehat{V}}} \exp \left(\sum_{\{i,j\} \in \widehat{E}} \widehat{\theta}_{ij} x_i x_j \right) \\ &= \sum_{x_{\widehat{V}}} \exp \left(x_{n+1} \sum_{i \in V} \theta_i x_i + \sum_{\{i,j\} \in E} \theta_{ij} x_i x_j \right) \\ &= \sum_{x_V} \exp \left(\sum_{i \in V} \theta_i x_i + \sum_{\{i,j\} \in E} \theta_{ij} x_i x_j \right) + \sum_{x_V} \exp \left(- \sum_{i \in V} \theta_i x_i + \sum_{\{i,j\} \in E} \theta_{ij} x_i x_j \right) \\ &= 2 \sum_{x_V} \exp \left(\sum_{i \in V} \theta_i x_i + \sum_{\{i,j\} \in E} \theta_{ij} x_i x_j \right) \\ &= 2Z, \end{aligned}$$

where the fourth equality follows from the symmetry between -1 and 1 in an Ising model. For the second part, since \widehat{P} is zero-field, we have that

$$\widehat{\mathbb{E}}[x_i] = 0 \quad \forall i \in \widehat{V}.$$

Now consider any $\{i, j\} \in E$. If x_{n+1} is fixed to a value of 1 , then the model is the same as the original one on V and we have

$$\widehat{\mathbb{E}}[x_i x_j \mid x_{n+1} = 1] = \mathbb{E}[x_i x_j] \quad \forall \{i, j\} \in E.$$

By symmetry (between -1 and 1) in the model, the same is true for $x_{n+1} = -1$ and so

$$\begin{aligned} \widehat{\mathbb{E}}[x_i x_j] &= \widehat{\mathbb{E}}[x_i x_j \mid x_{n+1} = 1] \widehat{P}(x_{n+1} = 1) + \widehat{\mathbb{E}}[x_i x_j \mid x_{n+1} = -1] \widehat{P}(x_{n+1} = -1) \\ &= \mathbb{E}[x_i x_j]. \end{aligned}$$

Fixing x_{n+1} to a value of 1 , we have

$$\widehat{\mathbb{E}}[x_i \mid x_{n+1} = 1] = \mathbb{E}[x_i] \quad \forall i \in V,$$

and by symmetry

$$\widehat{\mathbb{E}}[x_i \mid x_{n+1} = -1] = -\mathbb{E}[x_i] \quad \forall i \in V.$$

Combining the two equations above, we have

$$\begin{aligned} \widehat{\mathbb{E}}[x_i x_{n+1}] &= \widehat{\mathbb{E}}[x_i \mid x_{n+1} = 1] \widehat{P}(x_{n+1} = 1) + \widehat{\mathbb{E}}[-x_i \mid x_{n+1} = -1] \widehat{P}(x_{n+1} = -1) \\ &= \mathbb{E}[x_i]. \end{aligned}$$

■

Proof [Proposition 2] From Theorem 1, we see that the log partition function can be written as

$$\Phi(\theta) = n \log 2 + \sum_{\{i,j\} \in E} \log \cosh \theta_{ij} + \frac{1}{2} \log \det(I - AD),$$

where A and D are as given in Theorem 1. For the derivatives, we have

$$\begin{aligned} \frac{\partial \Phi(\theta)}{\partial \theta_{ij}} &= \tanh \theta_{ij} + \frac{1}{2} \text{Tr} \left((I - AD)^{-1} \frac{\partial (I - AD)}{\partial \theta_{ij}} \right) \\ &= \tanh \theta_{ij} - \frac{1}{2} \text{Tr} \left((I - AD)^{-1} AD'_{ij} \right) \\ &= w_{ij} - \frac{1}{2} (1 - w_{ij})^2 (S_{ij,ij} + S_{j,i,j}), \end{aligned}$$

where D'_{ij} is the derivative of the matrix D with respect to θ_{ij} . The first equality follows from the chain rule and the fact that $\nabla \ln |K| = K^{-1}$ for any matrix K . Please refer to Boyd and Vandenberghe (2004) for details.

For the Hessian, we have

$$\begin{aligned} \frac{\partial^2 \Phi(\theta)}{\partial \theta_{ij}^2} &= \frac{1}{Z(\theta)} \frac{\partial^2 Z(\theta)}{\partial \theta_{ij}^2} - \frac{1}{Z(\theta)^2} \left(\frac{\partial Z(\theta)}{\partial \theta_{ij}} \right)^2 \\ &= 1 - \mu_{ij}^2. \end{aligned}$$

For $\{i, j\} \neq \{k, l\}$, following Boyd and Vandenberghe (2004), we have

$$\begin{aligned} \frac{\partial^2 \Phi(\theta)}{\partial \theta_{ij} \partial \theta_{kl}} &= -\frac{1}{2} \text{Tr} \left(SD'_{ij} SD'_{kl} \right) \\ &= -\frac{1}{2} (1 - w_{ij}^2) (S_{ij,kl} S_{kl,ij} + S_{j,i,kl} S_{kl,ji} + S_{ij,kl} S_{kl,ij} + S_{j,i,kl} S_{kl,ji}) (1 - w_{kl}^2). \end{aligned}$$

On the other hand, we also have

$$\begin{aligned} T_{ijkl} &= e_{ij}^T(I + P)(S \circ S^T)(I + P)e_{kl} \\ &= (e_{ij} + e_{ji})^T(S \circ S^T)(e_{kl} + e_{lk}) \\ &= (S \circ S^T)_{ij,kl} + (S \circ S^T)_{ji,lk} + (S \circ S^T)_{j,i,kl} + (S \circ S^T)_{j,i,lk} \\ &= S_{ijkl}S_{kl,ij} + S_{ji,kl}S_{kl,ji} + S_{j,i,kl}S_{kl,ij} + S_{j,i,lk}S_{lk,ji} \end{aligned}$$

where e_{ij} is the unit vector with 1 in the ij^{th} position and 0 everywhere else. Using the above two equations, we obtain

$$H_{ijkl} = -\frac{1}{2}(1 - u_{ij}^2)T_{ijkl}(1 - u_{kl}^2).$$

Proof [Proposition 4] The proof follows from the following steps of inequalities. The Pythagorean law of information projection (Amari et al., 1992) gives

$$D(P, P_G) = D(P, P_{G+ij}) + D(P_{G+ij}, P_G).$$

The conditional rule of relative entropy (Cover and Thomas, 2006) gives

$$D(P_{G+ij}, P_G) = D(P_{G+ij}(x_i, x_j), P_G(x_i, x_j)) + D(P_{G+ij}(x_{V-ij}|x_i, x_j), P_G(x_{V-ij}|x_i, x_j)),$$

where $P_{G+ij}(x_i, x_j)$ and $P_G(x_i, x_j)$ represent the marginal distributions on x_i, x_j of probabilities P_{G+ij} and P_G respectively. Information inequality (Cover and Thomas, 2006) gives us:

$$D(P_{G+ij}(x_{V-ij}|x_i, x_j), P_G(x_{V-ij}|x_i, x_j)) \geq 0.$$

Plugging the above two properties into the first equation leads to the inequality

$$D(P, P_G) \geq D(P, P_{G+ij}) + D(P_{G+ij}(x_i, x_j), P_G(x_i, x_j)).$$

Finally, the property of information projection to $G + ij$ (Wainwright and Jordan, 2008) gives us $D(P_{G+ij}(x_i, x_j), P_G(x_i, x_j)) \geq D(P(x_i, x_j), P_G(x_i, x_j))$ leading to

$$D(P, P_G) \geq D(P, P_{G+ij}) + D(P(x_i, x_j), P_G(x_i, x_j)).$$

■

Appendix B. Applications: UGM Learned Models

For comparison to our planar learning algorithm, we provide the results of using the UGM MRF learning algorithm on the senate voting data and the ChemCam data. For all figures, we use a force-directed graph drawing algorithm (Fruchterman and Reingold, 1991). Figure 7 presents the graph learned using pseudolikelihood, UGM pseudo, from the full data set with the regularization parameter set to obtain the same number of edges as learned in the planar case ($3n - 6$ edges).

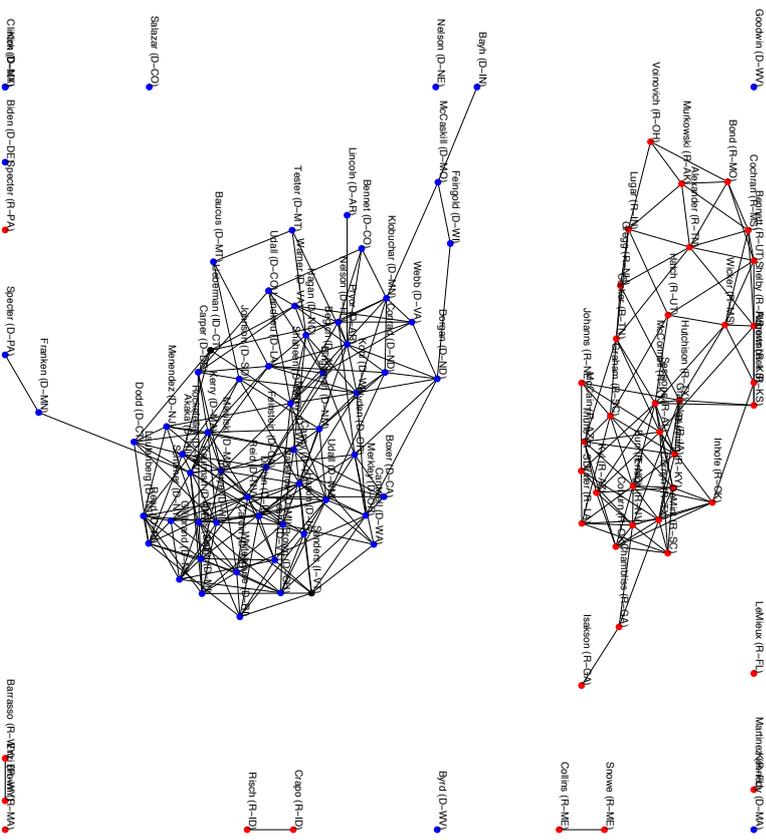


Figure 7: Senate voting graph learned by UGM pseudo with 318 edges. Blue nodes represent Democrats, red nodes represent Republicans and black nodes represent Independents.

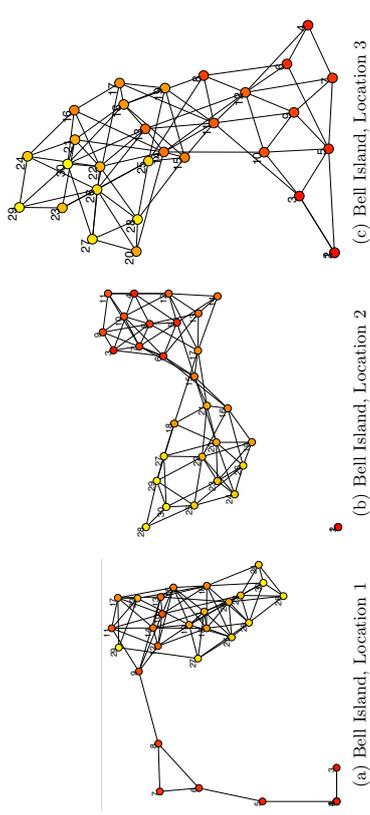


Figure 9: Learned UGM Pseudo graphs from ChemCam data. Nodes are numbered by LIBS laser shot number and color-coded by number starting with dark red at shot 1 and fading to light yellow at shot 30. We use a force-directed graph drawing algorithm (Fruchterman and Reingold, 1991).

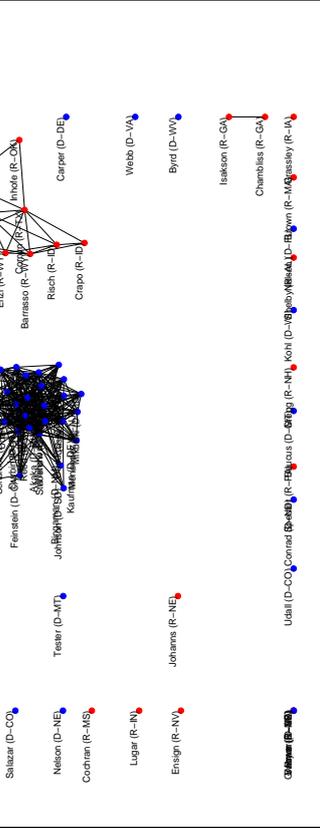


Figure 8: Senate voting graph learned by UGM. Loopy graph with 318 edges. Blue nodes represent Democrats, red nodes represent Republicans and black nodes represent Independents.

UGM graphs learned from the senate voting data are given in Figures 7 – 8. Figure 7 presents the graph learned using pseudolikelihood, UGM pseudo, from the full data set with the regularization parameter set to obtain 318 edges. Figure 8 presents the graph learned using loopy belief propagation, UGM loopy, from the full data set with the regularization parameter set to obtain 318 edges. The graphs learned using the tuning method are not displayed because they are nearly fully-connected graphs providing little visual information.

UGM graphs learned from the ChemCam data from the Bell Island target are given in Figures 9 – 10. Figure 9 presents the graph learned using pseudolikelihood, UGM pseudo, from the full data set with the regularization parameter set to obtain 84 edges. Figure 10 presents the graph learned using loopy belief propagation, UGM loopy, from the full data set with the regularization parameter set to obtain 84 edges. Graphs learned from this data using the tuning method to select the number of edges are nearly fully connected, and therefore provide little visual information.

Appendix C. Discussion: Counter Example

The result presented in Figure 11 illustrates the fact that our algorithm does not always recover the exact structure even when the underlying graph is planar and the algorithm is given exact moments as inputs. This counterexample gives insight into how the greedy algorithm works. The basic idea is that graphical models can have nodes which are not neighbors but are more correlated than some other nodes which are neighbors. If the spurious edges corresponding to these highly correlated nodes are added early on in the algorithm, then the actual edges may have to be left out because of the planarity restriction.

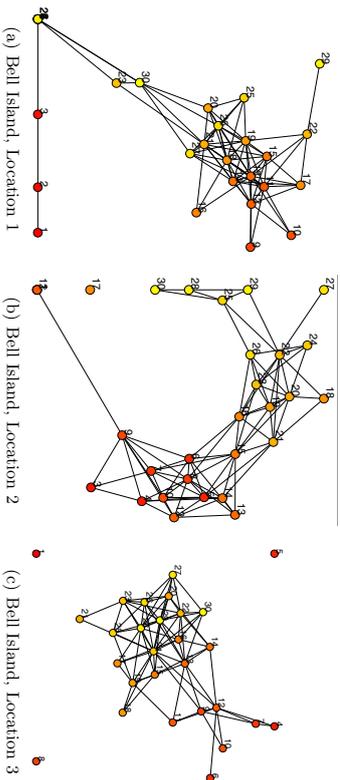


Figure 10: Learned UGM Loopy graphs from ChemCam data. Nodes are numbered by LIBS laser shot number and color-coded by number starting with dark red at shot 1 and fading to light yellow at shot 30. We use a force-directed graph drawing algorithm (Fruchterman and Reingold, 1991).

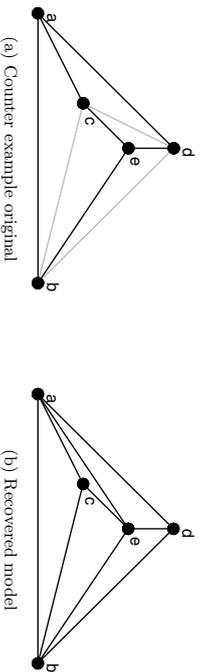


Figure 11: Example graphical models. (a) Counter example. (b) The recovered graphical model has one spurious edge $\{a, e\}$ and one missing edge $\{c, d\}$.

We define a zero-field Ising model on the graph in Figure 11a with the edge parameters as follows: $\theta_{bc} = \theta_{cd} = \theta_{bd} = 0.1$ and $\theta_{ij} = 1$ for all the other edges. Figure 11a shows the edge parameters in the graph pictorially using the intensity of the edges - the higher the intensity of an edge, higher the corresponding edge parameter. With these edge parameters, the correlation between nodes a and e is greater than the correlation between any other pair of nodes. This leads to the edge between a and e to be the first edge added in the algorithm. However, since $K=5$ (the complete graph on 5 nodes) is not planar, one of the actual edges is missed in the output graph as shown in Figure 11b.

References

- P. Abbeel, D. Koller, and A. Y. Ng. Learning factor graphs in polynomial time and sample complexity. *Journal of Machine Learning Research*, 7, 2006.
- H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6), 1974.
- S. Amari, K. Kurata, and H. Nagaoka. Information geometry of Boltzmann machines. *IEEE Transactions on Neural Networks*, 3(2), 1992.
- F. Bach and M. Jordan. Thin junction trees. In *Advances in Neural Information Processing Systems (NIPS)*, 2001.
- O. Banerjee, L. El Ghaoui, and A. d’Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine Learning Research*, 9, 2008.
- A. Barabasi and Z. Oltvai. Network biology: Understanding the cell’s functional organization. *Nature Reviews Genetics*, 5(2):101–113, 2004.
- O. Barndorff-Nielsen. Information and exponential families in statistical theory. *Bulletin of the American Mathematical Society*, 1979.
- D. Batra, A. Gallagher, D. Parikh, and T. Chen. Beyond trees: MRF inference via outer-planar decomposition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2496–2503. IEEE, 2010.
- S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14, 1968.
- M. Chrobak and T. Payne. A linear-time algorithm for drawing a planar graph on a grid. *Information Processing Letters*, 54(4), 1995.
- T. Cover and J. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.
- A. Deshpande, M. Garofalakis, and M. Jordan. Efficient stepwise selection in decomposable models. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2001.

- M. Fisher. On the dimer solution of planar Ising models. *Journal of Mathematical Physics*, 7(10), 1966.
- T. Fruchterman and E. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991.
- A. Galluccio, M. Loeb, and J. Vondrak. New algorithm for the Ising problem: Partition function for finite lattice graphs. *Physical Review Letters*, 84(26), 2000.
- V. Gómez, H. Kappen, and M. Chertkov. Approximate inference on planar graphs using loop calculus and belief propagation. *The Journal of Machine Learning Research*, 11: 1273–1296, 2010.
- A. Jaakkola and Globerson T. Approximate inference using planar graph decomposition. *Advances in Neural Information Processing Systems (NIPS)*, 19:473, 2007.
- M. Kac and J. Ward. A combinatorial solution of the two-dimensional Ising model. *Physical Review*, 88(6), 1952.
- D. Karger and N. Srebro. Learning Markov networks: Maximum bounded tree-width graphs. In *ACM-SIAM Symposium on Discrete Algorithms*, 2001.
- P. Kasteleyn. Dimer statistics and phase transitions. *Journal of Mathematical Physics*, 4(2), 1963.
- N. Lanza, A. Ollila, A. Cousin, R. Wiens, S. Clegg, N. Mangold, N. Bridges, D. Cooper, M. Schmidt, J. Berger, et al. Understanding the signature of rock coatings in laser-induced breakdown spectroscopy data. *Icarus*, 2014.
- S. Lauritzen. *Graphical Models*. Oxford University Press, 1996.
- S. Lee, V. Ganapathi, and D. Koller. Efficient structure learning of Markov networks using ℓ_1 -regularization. In *Advances in Neural Information Processing Systems (NIPS)*, 2006.
- R. Lipton and R. Tarjan. A separator theorem for planar graphs. *SIAM Journal of Applied Math*, 36(2), 1979.
- R. Lipton, D. Rose, and R. Tarjan. Generalized nested dissection. *SIAM Journal of Numerical Analysis*, 16(2), 1979.
- M. Loeb. *Discrete Mathematics in Statistical Physics*. Vieweg + Teubner, 2010.
- D. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.
- L. Onsager. Crystal statistics. I. A Two-dimensional model with an order-disorder transition. *Physical Review*, 65(3-4), 1944.
- F. Pozzi, T. Di Matteo, and T. Aste. Spread of risk across financial markets: Better to invest in the peripheries. *Scientific Reports*, 3, 2013.
- P. Ravikumar, M. Wainwright, and J. Lafferty. High-dimensional graphical model selection using ℓ_1 -regularized logistic regression. *Annals of Statistics*, 38(3), 2010.
- D. Richeson. *Euler's Gem: The Polyhedron Formula and the Birth of Topology*. Princeton University Press, 2012.
- M. Schmidt, K. Murphy, G. Fung, and R. Rosales. Structure learning in random fields for heart motion abnormality detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.
- N. Schraudolph and D. Kamenetsky. Efficient exact inference in planar Ising models. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1417–1424, 2008.
- G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2), 1978.
- D. Shahaf, A. Checketka, and C. Guestrin. Learning thin junction trees via graph cuts. In *Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009.
- S. Sherman. Combinatorial aspects of the Ising model for ferromagnetism. I. A Conjecture of Feynman on paths and graphs. *Journal of Mathematical Physics*, 1(3), 1960.
- S. Smith, K. Miller, G. Salimi-Khooshdidi, M. Webster, C. Beckmann, T. Nichols, J. Ramsey, and M. Woolrich. Network modelling methods for fMRI. *NeuroImage*, 54(2):875–891, 2011.
- M. Wainwright and M. Jordan. *Graphical Models, Exponential Families, and Variational Inference*. Now Publishers Inc., 2008.
- R. Wiens, S. Maurice, B. Barraclough, M. Saccoccio, W. Barkley, J. Bell III, S. Bender, J. Bernardin, D. Blaney, J. Blank, et al. The ChemCam instrument suite on the Mars Science Laboratory (MSL) rover: Body unit and combined system tests. *Space Science Reviews*, 170(1-4):167–227, 2012.
- J. Yarkony, A. Ihler, and C. Fowlkes. Fast planar correlation clustering for image segmentation. In *European Conference on Computer Vision (ECCV)*, pages 568–581. Springer, 2012.
- P. Zhang. Model selection via multifold cross validation. *Annals of Statistics*, 21(1), 1993.

Newton-Stein Method: An Optimization Method for GLMs via Stein's Lemma

Murat A. Erdogdu

*Department of Statistics
Stanford University*

Stanford, CA 94305-4065, USA

ERDOGDU@STANFORD.EDU

Editor: Qiang Lin

Abstract

We consider the problem of efficiently computing the maximum likelihood estimator in *Generalized Linear Models* (GLMs) when the number of observations is much larger than the number of coefficients ($n \gg p \gg 1$). In this regime, optimization algorithms can immensely benefit from approximate second order information. We propose an alternative way of constructing the curvature information by formulating it as an estimation problem and applying a *Stein-type lemma*, which allows further improvements through sub-sampling and eigenvalue thresholding. Our algorithm enjoys fast convergence rates, resembling that of second order methods, with modest per-iteration cost. We provide its convergence analysis for the general case where the rows of the design matrix are samples from a sub-Gaussian distribution. We show that the convergence has two phases, a quadratic phase followed by a linear phase. Finally, we empirically demonstrate that our algorithm achieves the highest performance compared to various optimization algorithms on several data sets.

Keywords: Optimization, Generalized Linear Models, Newton's method, Sub-sampling

1. Introduction

Generalized Linear Models (GLMs) play a crucial role in numerous statistical and machine learning problems. GLMs formulate the natural parameter in exponential families as a linear model and provide a miscellaneous framework for statistical methodology and supervised learning tasks. Celebrated examples include linear, logistic, multinomial regressions and applications to graphical models (Nelder and Baker, 1972; McCullagh and Nelder, 1989; Koller and Friedman, 2009).

In this paper, we focus on how to solve the maximum likelihood problem efficiently in the GLM setting when the number of observations n is much larger than the dimension of the coefficient vector p , i.e., $n \gg p \gg 1$. GLM optimization task is typically expressed as a minimization problem where the objective function is the negative log-likelihood that is denoted by $\ell(\beta)$ where $\beta \in \mathbb{R}^p$ is the coefficient vector. Many optimization algorithms are available for such minimization problems (Bishop, 1995; Boyd and Vandenberghe, 2004; Nesterov, 2004). However, only a few uses the special structure of GLMs. In this paper, we consider updates that are specifically designed for GLMs, which are of the form

$$\beta \leftarrow \beta - \gamma \mathbf{Q} \nabla_{\beta} \ell(\beta), \quad (1)$$

where γ is the step size and \mathbf{Q} is a scaling matrix which provides curvature information.

For the updates of the form Equation 1, the performance of the algorithm is mainly determined by the scaling matrix \mathbf{Q} . Classical *Newton's method* and *natural gradient descent* can be recovered by simply taking \mathbf{Q} to be the inverse Hessian and the inverse Fisher's information at the current iterate, respectively (Amari, 1998; Nesterov, 2004). Second order methods may achieve quadratic convergence rate, yet they suffer from excessive cost of computing the scaling matrix at every iteration. On the other hand, if we take \mathbf{Q} to be the identity matrix, we recover the standard *gradient descent* which has a linear convergence rate. Although the convergence rate of gradient descent is considered slow compared to that of second order methods such as Newton's method, modest per-iteration cost makes it practical for large-scale optimization.

The trade-off between convergence rate and per-iteration cost has been extensively studied (Bishop, 1995; Boyd and Vandenberghe, 2004; Nesterov, 2004). In $n \gg p \gg 1$ regime, the main objective is to construct a scaling matrix \mathbf{Q} that is computationally feasible which also provides sufficient curvature information. For this purpose, several Quasi-Newton methods have been proposed (Bishop, 1995; Nesterov, 2004). Updates given by Quasi-Newton methods satisfy an equation which is often called the *Quasi-Newton relation*. A well-known member of this class of algorithms is the *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) algorithm (Broyden, 1970; Fletcher, 1970; Goldfarb, 1970; Shanno, 1970).

In this paper, we propose a Newton-type algorithm that utilizes the special structure of GLMs by relying on a Stein-type lemma (Stein, 1981). It attains fast convergence rates with low per-iteration cost. We call our algorithm *Newton-Stein* method which we abbreviate as *NewSt*. Our contributions can be summarized as follows:

- We recast the problem of constructing a scaling matrix as an estimation problem and apply a Stein-type lemma along with the sub-sampling technique to form a computationally feasible \mathbf{Q} .
- Newton-Stein method allows further improvements through eigenvalue shrinkage, eigenvalue thresholding, sub-sampling and various other techniques that are available for covariance estimation.
- Excessive per-iteration cost of $\mathcal{O}(np^2 + p^3)$ of Newton's method is replaced by $\mathcal{O}(np + p^2)$ per-iteration cost and a one-time $\mathcal{O}(|S|p^2)$ cost, where $|S|$ is the sub-sample size.
- Assuming that the rows of the design matrix are i.i.d. and have bounded support (or sub-Gaussian), and denoting the iterates of Newton-Stein method by $\{\beta^t\}$, we prove a bound of the form

$$\|\beta^{t+1} - \beta_*\|_2 \leq \tau_1 \|\beta^t - \beta_*\|_2 + \tau_2 \|\beta^t - \beta_*\|_2^2, \quad (2)$$

where β_* is the true minimizer and τ_1, τ_2 are the convergence coefficients. The above bound implies that the local convergence starts with a quadratic phase and transitions into linear as the iterate gets closer to the true minimizer. We further establish a global convergence result of Newton-Stein method coupled with a line search algorithm.

- We demonstrate the performance of Newton-Stein method on real and synthetic data sets by comparing it to commonly used optimization algorithms.

The rest of the paper is organized as follows: Section 1.1 surveys the related work and Section 1.2 introduces the notations we use throughout the paper. Section 2 briefly discusses the GLM framework and its relevant properties. In Section 3, we introduce Newton-Stein method, develop its intuition, and discuss the computational aspects. Section 4 covers the theoretical results and in Section 4.4 we discuss how to choose the algorithm parameters. Section 5 provides the empirical results where we compare the proposed algorithm with several other methods on four data sets. Finally, in Section 6, we conclude with a brief discussion along with a few future research directions.

1.1 Related Work

There are numerous optimization techniques that can be used to find the maximum likelihood estimator in GLMs. For moderate values of n and p , the classical second order methods such as Newton’s method (also referred to as Newton-Raphson) are commonly used. In large-scale problems, data dimensionality is the main factor while determining the optimization method, which typically falls into one of two major categories: online and batch methods. Online methods use a gradient (or sub-gradient) of a single, randomly selected observation to update the current iterate (Robbins and Monro, 1951). Their per-iteration cost is independent of n , but the convergence rate might be extremely slow. There are several extensions of the classical stochastic descent algorithms, providing significant improvement and improved stability (Botton, 2010; Duchi et al., 2011; Schmidt et al., 2013; Kojte et al., 2015).

On the other hand, batch algorithms enjoy faster convergence rates, though their per-iteration cost may be prohibitive. In particular, second order methods enjoy quadratic convergence, but constructing the Hessian matrix generally requires excessive amount of computation. To remedy this issue, recent research is focused on designing an approximate and cost-efficient scaling matrix. This idea lies at the core of Quasi-Newton methods such as BFGS (Bishop, 1995; Nesterov, 2004).

Another approach to construct an approximate Hessian makes use of sub-sampling techniques (Martens, 2010; Byrd et al., 2011; Vinyals and Povey, 2011; Erdogdu and Montanari, 2015; Roosta-Khorasani and Mahoney, 2016a,b). Many contemporary learning methods rely on sub-sampling as it is simple and it provides significant boost over the first order methods. Further improvements through conjugate gradient methods and Krylov sub-spaces are available. Sub-sampling can also be used to obtain an approximate solution, with certain large deviation guarantees (Dhillon et al., 2013).

There are many composite variants of the aforementioned methods, that mostly combine two or more techniques. Well-known composite algorithms are the combinations of sub-sampling and Quasi-Newton (Schraudolph et al., 2007; Byrd et al., 2016), stochastic and deterministic gradient descent (Friedlander and Schmidt, 2012), natural gradient and Newton’s method (Le Roux and Fitzgibbon, 2010), natural gradient and low-rank approximation (Le Roux et al., 2008), sub-sampling and eigenvalue thresholding (Erdogdu and Montanari, 2015).

Lastly, algorithms that specialize on certain types of GLMs include coordinate descent methods for the penalized GLMs (Friedman et al., 2010), trust region Newton-type methods (Lin et al., 2008), and approximation methods (Erdogdu et al., 2016b,a).

1.2 Notation

Let $[n] = \{1, 2, \dots, n\}$ and denote by $|S|$, the size of a set S . The gradient and the Hessian of f with respect to β are denoted by $\nabla_{\beta} f$ and $\nabla_{\beta}^2 f$, respectively. The j -th derivative of a function $f(w)$ is denoted by $f^{(j)}(w)$. For a vector x and a symmetric matrix \mathbf{X} , $\|x\|_2$ and $\|\mathbf{X}\|_2$ denote the ℓ_2 and spectral norms of x and \mathbf{X} , respectively. $\|x\|_{\ell_2}$ denotes the sub-Gaussian norm, which will be defined later. S^{p-1} denotes the p -dimensional sphere. \mathcal{P}_C denotes the projections onto the set C , and $B_p(R) \subset \mathbb{R}^p$ denotes the p -dimensional ball of radius R . For a random variable x and density f , $x \sim f$ means that the distribution of x follows the density f . Multivariate Gaussian density with mean $\mu \in \mathbb{R}^p$ and covariance $\Sigma \in \mathbb{R}^{p \times p}$ is denoted as $N_p(\mu, \Sigma)$. For random variables x, y , $d(x, y)$ and $\mathcal{D}(x, y)$ denote probability metrics (will be explicitly defined) measuring the distance between the distributions of x and y . $\mathcal{N}(\cdot, \cdot)$ and T_{ϵ} denote the bracketing number and ϵ -net.

2. Generalized Linear Models

Distribution of a random variable $y \in \mathbb{R}$ belongs to an exponential family with natural parameter $\eta \in \mathbb{R}$ if its density can be written as

$$f(y|\eta) = e^{\eta y - \phi(\eta)} h(y),$$

where ϕ is the *cumulant generating function* and h is the *carrier density*. Let y_1, y_2, \dots, y_n be independent observations such that $\forall i \in [n], y_i \sim f(y_i|\eta_i)$. Denoting $\eta = (\eta_1, \dots, \eta_n)^T$, the joint likelihood can be written as

$$f(y_1, y_2, \dots, y_n|\eta) = \exp \left\{ \sum_{i=1}^n [y_i \eta_i - \phi(\eta_i)] \right\} \prod_{i=1}^n h(y_i). \quad (3)$$

We consider the problem of learning the maximum likelihood estimator in the above exponential family framework, where the vector $\eta \in \mathbb{R}^n$ is modeled through the linear relation, $\eta = \mathbf{X}\beta$,

for some design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ with rows $x_i \in \mathbb{R}^p$, and a coefficient vector $\beta \in \mathbb{R}^p$. This formulation is known as *Generalized Linear Models* (GLMs) with canonical links. The cumulant generating function ϕ determines the class of GLMs, i.e., for ordinary least squares (OLS) $\phi(z) = z^2/2$, for logistic regression (LR) $\phi(z) = \log(1 + e^z)$, and for Poisson regression (PR) $\phi(z) = e^z$.

Finding the maximum likelihood estimator in the above formulation is equivalent to minimizing the negative log-likelihood function $\ell(\beta)$,

$$\ell(\beta) = \frac{1}{n} \sum_{i=1}^n [\phi(x_i, \beta) - y_i \langle x_i, \beta \rangle], \quad (4)$$

where $\langle x, \beta \rangle$ is the inner product between the vectors x and β . The relation to OLS and LR can be seen much easier by plugging in the corresponding $\phi(z)$ in Equation 4. The gradient

and the Hessian of $\ell(\beta)$ can be written as:

$$\nabla_{\beta}^2 \ell(\beta) = \frac{1}{n} \sum_{i=1}^n \begin{bmatrix} \phi^{(1)}(x_i, \beta) x_i - y_i x_i \\ \phi^{(2)}(x_i, \beta) x_i x_i^T \end{bmatrix}, \quad \nabla_{\beta}^2 \ell(\beta) = \frac{1}{n} \sum_{i=1}^n \phi^{(2)}(x_i, \beta) x_i x_i^T. \quad (5)$$

For a sequence of scaling matrices $\{\mathbf{Q}^t\}_{t>0} \in \mathbb{R}^{p \times p}$, we consider iterations of the form

$$\hat{\beta}^{t+1} = \hat{\beta}^t - \gamma_t \mathbf{Q}^t \nabla_{\beta} \ell(\hat{\beta}^t)$$

where γ_t is the step size. The above iteration is our main focus, but with a new approach on how to compute the sequence of matrices $\{\mathbf{Q}^t\}_{t>0}$. We will formulate the problem of finding a scalable \mathbf{Q}^t as an estimation problem and apply a Stein-type lemma that provides us with a computationally efficient update rule.

3. Newton-Stein Method

Classical Newton-Raphson (or simply Newton's) method is the standard approach for training GLMs for moderately large data sets. However, its per-iteration cost makes it impractical for large-scale optimization. The main bottleneck is the computation of the Hessian matrix that requires $\mathcal{O}(np^2)$ flops which is prohibitive when $n \gg p \gg 1$. Numerous methods have been proposed to achieve the fast convergence rate of Newton's method while keeping the per-iteration cost manageable. To this end, a popular approach is to construct a scaling matrix \mathbf{Q}^t , which approximates the inverse Hessian at every iteration t .

The task of constructing an approximate Hessian can be viewed as an estimation problem. Assuming that the rows of \mathbf{X} are i.i.d. random vectors, the Hessian of the negative log-likelihood of GLMs with a cumulant generating function ϕ has the following sample average form

$$[\mathbf{Q}^t]^{-1} = \frac{1}{n} \sum_{i=1}^n x_i x_i^T \phi^{(2)}(x_i, \beta) \approx \mathbb{E}[x x^T \phi^{(2)}(x, \beta)].$$

We observe that $[\mathbf{Q}^t]^{-1}$ is just a sum of i.i.d. matrices. Hence, the true Hessian is nothing but a sample mean estimator to its expectation. Another natural estimator would be the sub-sampled Hessian method which is extensively studied by Martens, 2010; Byrd et al., 2011; Erdogdu and Montanari, 2015; Roosta-Khorasani and Mahoney, 2016a. Therefore, our goal is to propose an estimator for the population level Hessian that is also computationally efficient. Since n is large, the proposed estimator will be close to the true Hessian.

We use the following Stein-type lemma to find a more efficient estimator to the expectation of the Hessian.

Lemma 1 (Stein-type lemma) *Assume that $x \sim \mathbf{N}_p(0, \Sigma)$ and $\beta \in \mathbb{R}^p$ is a constant vector. Then for any function $f: \mathbb{R} \rightarrow \mathbb{R}$ that is twice "weakly" differentiable, we have*

$$\mathbb{E}[x x^T f(x, \beta)] = \mathbb{E}[f(x, \beta)] \Sigma + \mathbb{E}\left[f^{(2)}(x, \beta)\right] \Sigma \beta \beta^T \Sigma. \quad (6)$$

Proof The proof will follow from integration by parts. Let $g(x|\Sigma)$ denote the density of a multivariate normal random variable x with mean 0 and covariance Σ . We recall the basic

Algorithm 1 Newton-Stein Method

Input: $\hat{\beta}^0, |S|, \epsilon, \{\gamma_t\}_{t \geq 0}$.

1. Estimate the covariance using a random sub-sample $S \subset [n]$:

$$\hat{\Sigma}_S = \frac{1}{|S|} \sum_{i \in S} x_i x_i^T.$$

2. **while** $\|\hat{\beta}^{t+1} - \hat{\beta}^t\|_2 > \epsilon$ **do**

$$\hat{\mu}_2(\hat{\beta}^t) = \frac{1}{n} \sum_{i=1}^n \phi^{(2)}(x_i, \hat{\beta}^t), \quad \hat{\mu}_4(\hat{\beta}^t) = \frac{1}{n} \sum_{i=1}^n \phi^{(4)}(x_i, \hat{\beta}^t),$$

$$\mathbf{Q}^t = \frac{1}{\hat{\mu}_2(\hat{\beta}^t)} \left[\hat{\Sigma}_S^{-1} - \frac{\hat{\beta}^t [\hat{\beta}^t]^T}{\hat{\mu}_2(\hat{\beta}^t) / \hat{\mu}_4(\hat{\beta}^t) + \langle \hat{\Sigma}_S \hat{\beta}^t, \hat{\beta}^t \rangle} \right],$$

$$\hat{\beta}^{t+1} = \hat{\beta}^t - \gamma_t \mathbf{Q}^t \nabla_{\beta} \ell(\hat{\beta}^t),$$

$t \leftarrow t + 1$.

3. **end while**

Output: $\hat{\beta}^t$.

identity $xg(x|\Sigma)dx = -\Sigma dg(x|\Sigma)$ and write

$$\begin{aligned} \mathbb{E}[x x^T f(x, \beta)] &= \int x x^T f(x, \beta) g(x) dx, \\ &= \Sigma \left\{ \int f(x, \beta) g(x|\Sigma) dx + \int \beta x^T f^{(1)}(x, \beta) g(x|\Sigma) dx \right\}, \\ &= \Sigma \left\{ \mathbb{E}[f(x, \beta)] + \int \beta \beta^T f^{(2)}(x, \beta) g(x|\Sigma) dx \right\}, \\ &= \mathbb{E}[f(x, \beta)] \Sigma + \mathbb{E}\left[f^{(2)}(x, \beta)\right] \Sigma \beta \beta^T \Sigma. \end{aligned}$$

■

The right hand side of Equation 6 is a rank-1 update to the first term. Hence, its inverse can be computed with $\mathcal{O}(p^2)$ cost. Quantities that change at each iteration are the ones that depend on β , i.e.,

$$\mu_2(\beta) = \mathbb{E}[\phi^{(2)}(x, \beta)], \quad \text{and} \quad \mu_4(\beta) = \mathbb{E}[\phi^{(4)}(x, \beta)].$$

Note that $\mu_2(\beta)$ and $\mu_4(\beta)$ are scalar quantities and they can be estimated by their corresponding sample means $\hat{\mu}_2(\beta)$ and $\hat{\mu}_4(\beta)$ (explicitly defined at Step 2 of Algorithm 1) respectively, with only $\mathcal{O}(np)$ computation.

To complete the estimation task suggested by Equation 6, we need an estimator for the covariance matrix Σ . A natural estimator is the sample mean where, we only use a sub-sample of the indices $S \subset [n]$ so that the cost is reduced to $\mathcal{O}(|S|p^2)$ from $\mathcal{O}(np^2)$. Sub-sampling based sample mean estimator is denoted by $\hat{\Sigma}_S = \frac{1}{|S|} \sum_{i \in S} x_i x_i^T$, which is

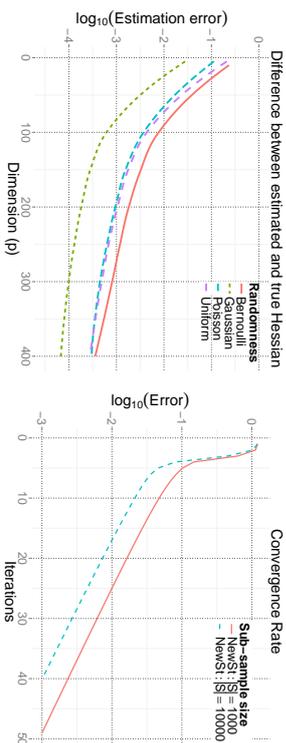


Figure 1: The left plot demonstrates the accuracy of proposed Hessian estimation over different distributions. Number of observations is set to be $n = \mathcal{O}(p \log(p))$. The right plot shows the phase transition in the convergence rate of Newton-Stein method (NewSt). Convergence starts with a quadratic rate and transitions into linear. Plots are obtained using *Covertype* data set.

widely used in large-scale problems (Vershynin, 2010). We highlight the fact that Lemma 1 replaces $\mathcal{O}(np^2)$ per-iteration cost of Newton’s method with a one-time cost of $\mathcal{O}(np^2)$. We further use sub-sampling to reduce this one-time cost to $\mathcal{O}(|S|p^2)$, and obtain the following Hessian estimator at β

$$\left[\mathbf{Q}^q \right]^{-1} = \underbrace{f_{i_2}(\beta)}_{\in \mathbb{R}^{p \times p}} \widehat{\Sigma}_S + \underbrace{f_{i_4}(\beta)}_{\in \mathbb{R}} \underbrace{\widehat{\Sigma}_S \beta \beta^T \widehat{\Sigma}_S}_{\text{rank-1 update}} \underbrace{\widehat{\Sigma}_S}_{\in \mathbb{R}^{p \times p}} \quad (7)$$

We emphasize that any covariance estimation method can be applied in the first step of the algorithm. There are various estimation techniques most of which rely on the concept of *shrinkage* (Cai et al., 2010; Donoho et al., 2013). This is because, important curvature information is generally contained in the largest few spectral features (Erdogdu and Montanari, 2015). In particular, for a given threshold r , we suggest to use the largest r eigenvalues of the sub-sampled covariance estimator $\widehat{\Sigma}_S$, and setting rest of them to $(r+1)$ -th eigenvalue. This operation helps denoising and provides additional computational benefits when inverting the covariance estimator (Erdogdu and Montanari, 2015).

Inverting the constructed Hessian estimator can make use of the low-rank structure. First, notice that the updates in Equation 7 are based on rank-1 matrix additions. Hence, we can simply apply Sherman–Morrison inversion formula to Equation 7 and obtain an explicit equation for the scaling matrix \mathbf{Q}^t (Step 2 of Algorithm 1). This formulation would impose another inverse operation on the covariance estimator. We emphasize that this operation is performed once. Therefore, instead of $\mathcal{O}(p^3)$ per-iteration cost of Newton’s method due to inversion, Newton-Stein method (NewSt) requires $\mathcal{O}(p^2)$ per-iteration and a one-time cost of $\mathcal{O}(p^3)$. Assuming that Newton-Stein and Newton methods converge in T_1 and T_2 iterations respectively, the overall complexity of Newton-Stein is $\mathcal{O}(npT_1 + p^2T_1 + (|S| + p)p^2) \approx$

$\mathcal{O}(npT_1 + p^2T_1 + |S|p^2)$ whereas that of Newton is $\mathcal{O}(np^2T_2 + p^3T_2)$. We show both empirically and theoretically that the quantities T_1 and T_2 are close to each other.

The convergence rate of Newton-Stein method has two phases. Convergence starts quadratically and transitions into linear rate when it gets close to the true minimizer. The phase transition behavior can be observed through the right plot in Figure 1. This is a consequence of the bound provided in Equation 2, which is the main result of our theorems on the local convergence (given in Section 4).

Even though Lemma 1 assumes that the covariates are multivariate Gaussian random vectors, in Section 4, the only assumption we make on the covariates is either bounded support or sub-Gaussianity, both of which cover a wide class of random variables including Bernoulli, elliptical distributions, bounded variables etc. The left plot of Figure 1 shows that the estimation is accurate for many distributions. This is a consequence of the fact that the proposed estimator in Equation 7 relies on the distribution of x only through inner products of the form $\langle x, v \rangle$, which in turn results in an approximate normal distribution due to the central limit theorem. To provide more intuition, we explain this through *zero-biased transformations* which is a general version of Stein’s lemma for arbitrary distributions (Goldstein and Reinert, 1997).

Definition 2 Let z be a random variable with mean 0 and variance σ^2 . Then, there exists a random variable z^* that satisfies $\mathbb{E}[zf(z)] = \sigma^2 \mathbb{E}[f'(z^*)]$, for all differentiable functions f . The distribution of z^* is said to be the *z-zero-bias distribution*.

The normal distribution is the unique distribution whose zero-bias transformation is itself (i.e. the normal distribution is a fixed point of the operation mapping the distribution of z to that of z^*). The distribution of z^* is referred to as *z-zero-bias distribution* and is entirely determined by the distribution of z . Properties such as existence can be found, for example, in Chen et al., 2010.

To provide some intuition behind the usefulness of Lemma 1 even for arbitrary distributions, we use zero-bias transformations. For simplicity, assume that the covariate vector x has i.i.d. entries from an arbitrary distribution with mean 0, and variance 1. Then the zero-bias transformation applied twice to the entry (i, j) of matrix $\mathbb{E}[xx^T f(\langle x, \beta \rangle)]$ yields

$$\mathbb{E}[x_i x_j f(\langle x, \beta \rangle)] = \begin{cases} \mathbb{E}[f(\beta_i x_i^* + \sum_{k \neq i} x_k \beta_k)] + \beta_i^2 \mathbb{E}[f^{(2)}(\beta_i x_i^{**} + \sum_{k \neq i} x_k \beta_k)] & \text{if } i = j, \\ \beta_i \beta_j \mathbb{E}[f^{(2)}(\beta_i x_i^* + \beta_j x_j^* + \sum_{k \neq i, j} x_k \beta_k)] & \text{if } i \neq j, \end{cases}$$

where x_i^* and x_i^{**} have x_i -zero-bias and x_i^* -zero-bias distributions, respectively. For each entry (i, j) at most two summands of $\langle x, \beta \rangle = \sum_k x_k \beta_k$ change their distributions. Therefore, if β is well spread and p is sufficiently large, the sums inside the expectations will behave similar to the inner product $\langle x, \beta \rangle$. Correspondingly, the above equations will be close to their Gaussian counterpart as given in Equation 6.

4. Theoretical Results

We start by introducing the terms that will appear in the theorems. Then we will provide two technical results on bounded and sub-Gaussian covariates. The proofs of the theorems are technical and provided in Appendix.

4.1 Preliminaries

Hessian estimation described in the previous section relies on a Gaussian approximation. For theoretical purposes, we use the following probability metric to quantify the gap between the distribution of x_i 's and that of a normal vector.

Definition 3 Given a family of functions \mathcal{H} , and random vectors $x, y \in \mathbb{R}^p$, for \mathcal{H} and any $h \in \mathcal{H}$, define

$$d_{\mathcal{H}}(x, y) = \sup_{h \in \mathcal{H}} d_h(x, y) \quad \text{where} \quad d_h(x, y) = |\mathbb{E}[h(x)] - \mathbb{E}[h(y)]|.$$

Many probability metrics can be expressed as above by choosing a suitable function class \mathcal{H} . Examples include *Total Variation* (TV), *Kolmogorov* and *Wasserstein* metrics (Gibbs and Su, 2002; Chen et al., 2010). Based on the second and the fourth derivatives of the cumulant generating function, we define the following function classes:

$$\begin{aligned} \mathcal{H}_1 &= \left\{ h(x) = \phi^{(2)}(\langle x, \beta \rangle) : \beta \in \mathcal{C} \right\}, & \mathcal{H}_2 &= \left\{ h(x) = \phi^{(4)}(\langle x, \beta \rangle) : \beta \in \mathcal{C} \right\}, \\ \mathcal{H}_3 &= \left\{ h(x) = \langle v, x \rangle^2 \phi^{(2)}(\langle x, \beta \rangle) : \beta \in \mathcal{C}, \|v\|_2 = 1 \right\}, \end{aligned}$$

where $\mathcal{C} \in \mathbb{R}^p$ is a closed, convex set that is bounded by the radius R . Exact calculation of such probability metrics are often difficult. The general approach is to upper bound the distance by a more intuitive metric. In our case, we observe that $d_{\mathcal{H}_j}(x, y)$ for $j = 1, 2, 3$, can be easily upper bounded by $d_{\text{TV}}(x, y)$ up to a scaling constant, when the covariates have bounded support.

In our theoretical results, we rely on projected updates onto a closed convex set \mathcal{C} , which are of the form

$$\hat{\beta}^{t+1} = \mathcal{P}_{\mathcal{C}}^t \left(\hat{\beta}^t - \gamma \mathbf{Q}^t \nabla_{\beta} \ell(\hat{\beta}^t) \right)$$

where the projection is defined as $\mathcal{P}_{\mathcal{C}}^t(\beta) = \arg \min_{w \in \mathcal{C}} \frac{1}{2} \|w - \beta\|_{\mathbf{Q}^{t-1}}^2$, with \mathcal{C} bounded by R . This is a special case of proximal Newton-type algorithms and further generalization is straightforward (See Lee et al., 2014). We will further assume that the covariance matrix has full rank and its smallest eigenvalue is lower bounded by a positive constant.

4.2 Bounded Covariates

We have the following per-step bound for the iterates generated by the Newton-Stein method, when the covariates are supported on a ball.

Theorem 4 (Local convergence) Assume that the covariates x_1, x_2, \dots, x_n are i.i.d. random vectors supported on a ball of radius \sqrt{K} with

$$\mathbb{E}[x_i] = 0 \quad \text{and} \quad \mathbb{E}[x_i x_i^T] = \Sigma.$$

Further assume that the cumulant generating function ϕ has bounded 2nd-5th derivatives and that the set \mathcal{C} is bounded by R . For $\{\hat{\beta}^t\}_{t>0}$ given by the Newton-Stein method for $\gamma = 1$, define the event

$$\mathcal{E} = \left\{ \inf_{\|u\|_2=1} \left| \mu_2(\hat{\beta}^t)(u, \Sigma u) + \mu_4(\hat{\beta}^t)(u, \Sigma \hat{\beta}^t) \right| > 2\kappa^{-1} \quad \forall t, \quad \beta_* \in \mathcal{C} \right\} \quad (8)$$

for some positive constant κ , and the optimal value β_* . If $n, |S|$ and p are sufficiently large, then there exist constants c, c_1, c_2 depending on the radii K, R , $\mathbb{P}(\mathcal{E})$ and the bounds on $\phi^{(2)}$ and $\phi^{(4)}$ such that conditioned on the event \mathcal{E} , with probability at least $1 - c/p^2$, we have

$$\|\hat{\beta}^{t+1} - \beta_*\|_2 \leq \tau_1 \|\hat{\beta}^t - \beta_*\|_2 + \tau_2 \|\hat{\beta}^t - \beta_*\|_2^2, \quad (9)$$

where the coefficients τ_1 and τ_2 are deterministic constants defined as

$$\tau_1 = \kappa \mathfrak{D}(x, z) + c_1 \kappa \sqrt{\frac{p}{\min\{p/\log(p)|S|, n/\log(n)\}}}, \quad \tau_2 = c_2 \kappa, \quad (10)$$

and $\mathfrak{D}(x, z)$ is defined as

$$\mathfrak{D}(x, z) = \|\Sigma\|_2 d_{\mathcal{H}_1}(x, z) + \|\Sigma\|_2^2 R^2 d_{\mathcal{H}_2}(x, z) + d_{\mathcal{H}_3}(x, z), \quad (11)$$

for a multivariate Gaussian random variable z with the same mean and covariance as x_i 's.

The bound in Equation 9 holds with high probability, and the coefficients τ_1 and τ_2 are deterministic constants which will describe the convergence behavior of the Newton-Stein method. Observe that the coefficient τ_1 is sum of two terms: $\mathfrak{D}(x, z)$ measures how accurate the Hessian estimation is, and the second term depends on the sub-sampling size $|S|$ and the data dimensions n, p .

Theorem 4 shows that the convergence of Newton-Stein method can be upper bounded by a compositely converging sequence, that is, the squared term will dominate at first providing us with a quadratic rate, then the convergence will transition into a linear phase as the iterate gets close to the optimal value. The coefficients τ_1 and τ_2 govern the linear and quadratic terms, respectively. The effect of sub-sampling appears in the coefficient of linear term. In theory, there is a threshold for the sub-sampling size $|S|$, namely $\mathcal{O}(n/\log(n))$, beyond which further sub-sampling has no effect. The transition point between the quadratic and the linear phases is determined by the sub-sampling size and the properties of the data. The phase transition behavior can be observed through the right plot in Figure 1.

Using the above theorem, we state the following corollary.

Corollary 5 Assume that the assumptions of Theorem 4 hold. For a constant $\delta \geq \mathbb{P}(\mathcal{E}^C)$, and a tolerance ϵ satisfying

$$\epsilon \geq 20R \{c/p^2 + \delta\},$$

and for an iterate satisfying $\mathbb{E}[\|\hat{\beta}^t - \beta_*\|_2] > \epsilon$, the following inequality holds for the iterates of Newton-Stein method,

$$\mathbb{E} \left[\|\hat{\beta}^{t+1} - \beta_*\|_2 \right] \leq \bar{\tau}_1 \mathbb{E} \left[\|\hat{\beta}^t - \beta_*\|_2 \right] + \tau_2 \mathbb{E} \left[\|\hat{\beta}^t - \beta_*\|_2^2 \right],$$

where $\bar{\tau}_1 = \tau_1 + 0.1$ and τ_1, τ_2 are as in Theorem 4.

The bound stated in the above corollary is an analogue of composite convergence (given in Equation 9) in expectation. Note that our results make strong assumptions on the derivatives of the cumulant generating function ϕ . We emphasize that these assumptions

are valid for linear and logistic regressions. An example that does not fit in our scheme is *Poisson regression* with $\phi(z) = e^z$. However, we observed empirically that the algorithm still provides significant improvement.

The following theorem characterizes the local convergence behavior of a compositely converging sequence.

Theorem 6 *Assume that the assumptions of Theorem 4 hold with $\tau_1 < 1$ and for $\vartheta = \|\beta^0 - \beta_*\|_2$ define the interval $\Xi = \left(\frac{\tau_1\vartheta}{1-\tau_2\vartheta}, \vartheta\right)$. Conditioned on the event $\mathcal{E} \cap \{\vartheta < (1 - \tau_1)/\tau_2\}$, there exists a constant c such that with probability at least $1 - c/p^2$, the number of iterations to reach a tolerance of ϵ cannot exceed*

$$\inf_{\xi \in \Xi} \mathcal{J}(\xi) := \log_2 \left(\frac{\log(\tau_1 + \tau_2\xi)}{\log((\tau_1/\xi + \tau_2)(1 - \tau_1)/\tau_2)} + \frac{\log(\epsilon/\xi)}{\log(\tau_1 + \tau_2\xi)} \right), \quad (12)$$

where the constants τ_1 and τ_2 are as in Theorem 4.

The expression in Equation 12 has two terms: the first one is due to the quadratic phase whereas the second one is due to the linear phase. To obtain the properties of local convergence, a locality constraint is required. We note that $\tau_1 < 1$ is a necessary assumption, which is satisfied for sufficiently large n and $|S|$.

In the following, we establish the global convergence of the Newton-Stein method coupled with a backtracking line search—which is explicitly given in Section 4.4.

Theorem 7 (Global Convergence) *Assume that the assumptions of Theorem 4 hold and at each step, the step size τ_t of the Newton-Stein method is determined by the backtracking line search with parameters a and b . Then conditioned on the event \mathcal{E} , there exists a constant c such that with probability at least $1 - c/p^2$, the sequence of iterates $\{\beta^t\}_{t>0}$ generated by the Newton-Stein method converges globally.*

4.3 Sub-Gaussian Covariates

In this section, we carry our analysis to the more general case, where the covariates are sub-Gaussian vectors.

Theorem 8 (Local convergence) *Assume that x_1, x_2, \dots, x_n are i.i.d. sub-Gaussian random vectors with sub-Gaussian norm K such that*

$$\mathbb{E}[x_1] = 0, \quad \mathbb{E}[\|x\|_2] = \mu \quad \text{and} \quad \mathbb{E}[xx^T] = \Sigma.$$

Further assume that the cumulant generating function ϕ is uniformly bounded and has bounded 3rd-5th derivatives and that C is bounded by R . For $\{\beta^t\}_{t>0}$ given by the Newton-Stein method and the event \mathcal{E} in Equation 8, if we have $n, |S|$ and p sufficiently large and $n^{0.2}/\log(n) \gtrsim p$,

then there exist constants c_1, c_2, c_3, c_4 depending on the eigenvalues of Σ , the radius R , μ , $\mathbb{P}(\mathcal{E})$ and the bounds on $\phi^{(2)}$ and $|\phi^{(4)}|$ such that conditioned on the event \mathcal{E} , with probability at least $1 - c_1 e^{-c_2 p}$, the bound given in Equation 9 holds for constants

$$\tau_1 = \kappa \mathfrak{D}(x, z) + c_3 \kappa \sqrt{\frac{p}{\min\{|S|, n^{0.2}/\log(n)\}}}, \quad \tau_2 = c_4 \kappa p^{1.5}, \quad (13)$$

where $\mathfrak{D}(x, z)$ defined as in Equation 11.

The above theorem is more restrictive than Theorem 4. We require n to be much larger than the dimension p . Also note that a factor of $p^{1.5}$ appears in the coefficient of the quadratic term. We also notice that the threshold for the sub-sample size reduces to $n^{0.2}/\log(n)$. We have the following analogue of Corollary 5.

Corollary 9 *Assume that the assumptions of Theorem 8 hold. For a constant $\delta \geq \mathbb{P}(\mathcal{E}^c)$, and a tolerance ϵ satisfying*

$$\epsilon \geq 20R\sqrt{c_1 e^{-c_2 p} + \delta},$$

and for an iterate satisfying $\mathbb{E}[\|\beta^t - \beta_*\|_2] > \epsilon$, the iterates of Newton-Stein method will satisfy,

$$\mathbb{E}[\|\beta^{t+1} - \beta_*\|_2] \leq \bar{\tau}_1 \mathbb{E}[\|\beta^t - \beta_*\|_2] + \tau_2 \mathbb{E}[\|\beta^t - \beta_*\|_2^2],$$

where $\bar{\tau}_1 = \tau_1 + 0.1$ and τ_1, τ_2 are as in Theorem 8.

When the covariates are in fact multivariate normal, we have $\mathfrak{D}(x, z) = 0$ which implies that the coefficient τ_1 is smaller. Correspondingly, the quadratic phase lasts longer providing better performance.

We conclude this section by noting that the global convergence properties of the sub-Gaussian case is very similar to the previous section where we had bounded covariates.

4.4 Algorithm Parameters

Newton-Stein method takes two input parameters and for those, we suggest near-optimal choices based on our theoretical results. We further discuss the choice of a covariance estimation method which provides additional improvements to the proposed algorithm.

- *Sub-sample size:* Newton-Stein method uses a subset of indices to approximate the covariance matrix Σ . Corollary 5.50 of Vershynin, 2010 proves that a sample size of $\mathcal{O}(p)$ is sufficient for sub-Gaussian covariates and that of $\mathcal{O}(p \log(p))$ is sufficient for arbitrary distributions supported in some ball to estimate a covariance matrix by its sample mean estimator. In the regime we consider, $n \gg p$, we suggest to use a sample size of $|S| = \mathcal{O}(p \log(p))$ for this task.

- *Covariance estimation method:* Many methods have been suggested to improve the estimation of the covariance matrix and almost all of them rely on the concept of *shrinkage* (Cai et al., 2010; Donoho et al., 2013). Therefore, we suggest to use a thresholding based approach suggested by Erdogdu and Montanari, 2015. For a given threshold r , we take the largest r eigenvalues of the sub-sampled covariance estimator, setting rest of them to $(r + 1)$ -th eigenvalue. Eigenvalue thresholding can be considered as a shrinkage operation which will retain only the important second order information. Choosing the rank threshold r can be simply done on the sample mean estimator of Σ . After obtaining the sub-sampled estimate of the mean, one can either plot the spectrum and choose manually or use an optimal technique from Donoho

et al., 2013. The suggested method requires a single time $\mathcal{O}(np^2)$ computation and reduces the cost of inversion from $\mathcal{O}(p^3)$ to $\mathcal{O}(np^2)$. We highlight that the Newton-Stein method was originally presented with the eigenvalue thresholding in an early version of this paper (Erdogdu, 2015).

- *Step size*: Step size choices for the Newton-Stein method are quite similar to those of Newton-type methods (i.e., see Boyd and Vandenberghe, 2004). In the *damped phase*, one should use a line search algorithm such as *backtracking* with parameters $a \in (0, 0.5)$ and $b \in (0, 1)$. Defining the modified gradient (or composite gradient Lee et al., 2014) $D_\gamma(\hat{\beta}^t) = \frac{1}{\gamma} \{ \hat{\beta}^t - \mathcal{P}_C^t(\hat{\beta}^t - \gamma \mathbf{Q}^t \nabla \ell(\hat{\beta}^t)) \}$, we compute the step size via

$$\gamma = \bar{\gamma}; \quad \text{while: } \ell(\hat{\beta}^t - \gamma D_\gamma(\hat{\beta}^t)) > \ell(\hat{\beta}^t) - a\gamma \langle \nabla \ell(\hat{\beta}^t), D_\gamma(\hat{\beta}^t) \rangle, \quad \gamma \leftarrow \gamma b.$$

The above line search algorithm leads to global convergence with high probability as stated in Theorem 7.

The step choice for the local phase depends on the use of eigenvalue thresholding. If no shrinkage method is applied, line search algorithm should be initialized with $\bar{\gamma} = 1$. If a shrinkage method (e.g. eigenvalue thresholding) is applied, then choosing a larger local step size may provide faster convergence. If the data follows the r -spiked model, the optimal step size will be close to 1 if there is no sub-sampling. However, due to fluctuations resulting from sub-sampling, starting with $\bar{\gamma} = 1.2$ will provide faster local rates. This case has been explicitly studied in a preliminary version of this work (Erdogdu, 2015). A heuristic derivation and a detailed discussion can also be found in Section E in the Appendix.

5. Experiments

In this section, we validate the performance of Newton-Stein method through extensive numerical studies. We experimented on two commonly used GLM optimization problems, namely, *Logistic Regression* (LR) and *Linear Regression* (OLS). LR minimizes Equation 4 for the logistic function $\phi(z) = \log(1 + e^z)$, whereas OLS minimizes the same equation for $\phi(z) = z^2/2$. In the following, we briefly describe the algorithms that are used in the experiments:

- *Newton's Method* (NM) uses the inverse Hessian evaluated at the current iterate, and may achieve local quadratic convergence. NM steps require $\mathcal{O}(np^2 + p^3)$ computation which makes it impractical for large-scale data sets.
- *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) forms a curvature matrix by cultivating the information from the iterates and the gradients at each iteration. Under certain assumptions, the convergence rate is locally super-linear and the per-iteration cost is comparable to that of first order methods.
- *Limited Memory BFGS* (L-BFGS) is similar to BFGS, and uses only the recent few iterates to construct the curvature matrix, gaining significant performance in terms of memory usage.

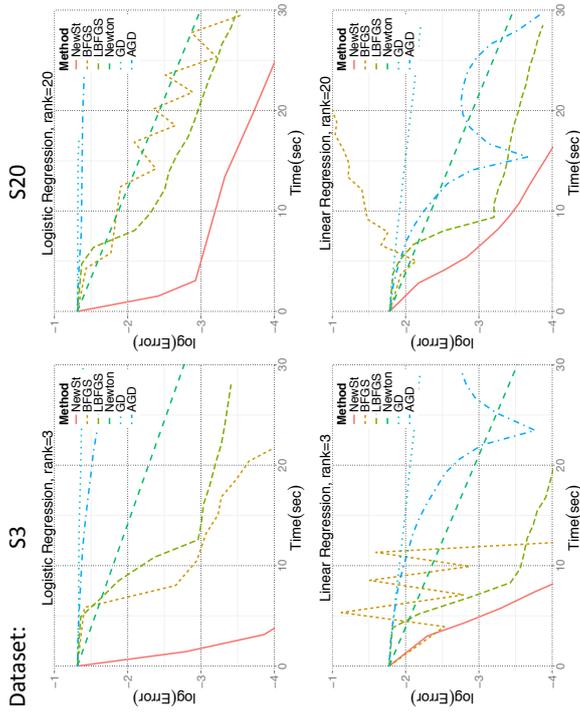


Figure 2: Performance of various optimization methods on two different simulated data sets. Red straight line represents the Newton-Stein method (NewSt). y and x axes denote $\log_{10}(\|\hat{\beta}^t - \beta_*\|_2)$ and time elapsed in seconds, respectively.

- *Gradient Descent* (GD) update is proportional to the negative of the full gradient evaluated at the current iterate. Under smoothness assumptions, GD achieves a locally linear convergence rate, with $\mathcal{O}(np)$ per-iteration cost.
- *Accelerated Gradient Descent* (AGD) is proposed by Nesterov (Nesterov, 1983), which improves over the gradient descent by using a momentum term. Performance of AGD strongly depends of the smoothness of the function.

For all the algorithms, we use a constant step size that provides the fastest convergence. We use the Newton-Stein method with eigenvalue thresholding as described in Section 4.4. The parameters such as sub-sample size $|S|$, and rank r are selected by following the guidelines described in Section 4.4. The rank threshold r (which is an input to the eigenvalue thresholding) is specified at the title of each plot.

5.1 Simulations With Synthetic Data Sets

Synthetic data sets, S3, S10, and S20 are generated through a multivariate Gaussian distribution where the covariance matrix follows r -spiked model, i.e., $r = 3$ for S3 and $r = 20$

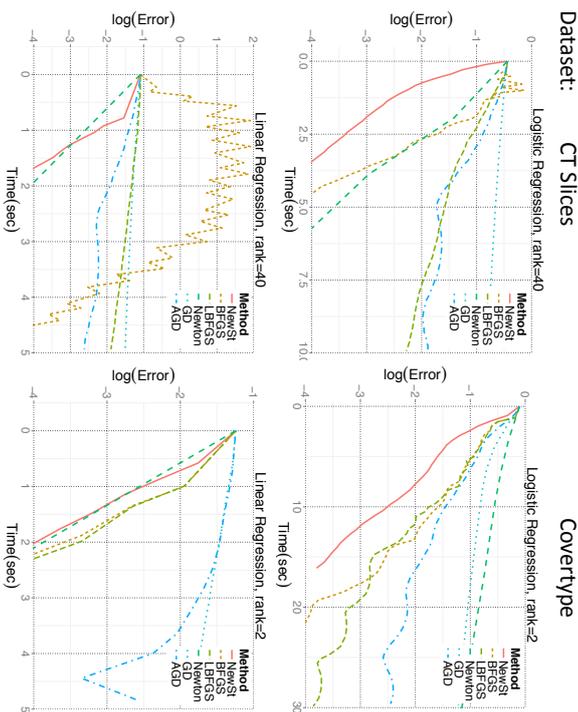


Figure 3: Performance of various optimization methods on two different real data sets obtained from Lichman, 2013. Red straight line represents the Newton-Stein method (NewSt). y and x axes denote $\log_{10}(\|\beta^t - \beta_*\|_2)$ and time elapsed in seconds, respectively.

for S20. To generate the covariance matrix, we first generate a random orthogonal matrix, say M . Next, we generate a diagonal matrix Λ that contains the eigenvalues, i.e., the first r diagonal entries are chosen to be large, and rest of them are equal to 1. Then, we let $\Sigma = M\Lambda M^T$. For dimensions of the data sets, see Table 2. We also emphasize that the data dimensions are chosen so that Newton’s method still does well.

The simulation results are summarized in Figure 2. Further details regarding the experiments can be found in Table 1. We observe that Newton-Stein method (NewSt) provides a significant improvement over the classical techniques.

Observe that the convergence rate of NewSt has a clear phase transition point in the top left plot in Figure 2. As argued earlier, this point depends on various factors including sub-sampling size $|S|$ and data dimensions n, p , the rank threshold r and structure of the covariance matrix. The prediction of the phase transition point is an interesting line of research. However, our convergence guarantees are conservative and we believe that they cannot be used for this purpose.

5.2 Experiments With Real Data Sets

We experimented on two real data sets where the data sets are downloaded from UCI repository (Lichman, 2013). Both data sets satisfy $n \gg p$, but we highlight the difference between the proportions of dimensions n/p . See Table 2 for details.

We observe that Newton-Stein method performs better than classical methods on real data sets as well. More specifically, the methods that come closer to NewSt is Newton’s method for moderate n and p and BFGS when n is large.

The optimal step-size for Newton-Stein method will typically be larger than 1 which is mainly due to eigenvalue thresholding operation. This feature is desirable if one is able to obtain a large step-size that provides convergence. In such cases, the convergence is likely to be faster, yet more unstable compared to the smaller step size choices. We observed that similar to other second order algorithms, Newton-Stein method is also susceptible to the step size selection. If the data is not well-conditioned, and the sub-sample size is not sufficiently large, algorithm might have poor performance. This is mainly because the sub-sampling operation is performed only once at the beginning. Therefore, it might be good in practice to sub-sample once in every few iterations.

DATA SET	S3				S20			
	LR	LS	LR	LS	LR	LS	LR	LS
TYPE	TIME(SEC)	ITER	TIME(SEC)	ITER	TIME(SEC)	ITER	TIME(SEC)	ITER
NEWSt	10.637	2	8.763	4	23.158	4	16.475	10
BFGS	22.885	8	13.149	6	40.258	17	54.294	37
LBFGS	46.763	19	19.952	11	51.888	26	33.107	20
NEWTON	55.328	2	38.150	1	47.955	2	39.328	1
GD	865.119	493	155.155	100	1204.01	245	145.987	100
AGD	169.473	82	65.396	42	182.031	83	56.257	38

DATA SET	CT SLICES				COVERTYPE			
	LR	LS	LR	LS	LR	LS	LR	LS
TYPE	TIME(SEC)	ITER	TIME(SEC)	ITER	TIME(SEC)	ITER	TIME(SEC)	ITER
NEWSt	4.191	32	1.799	11	16.113	31	2.080	5
BFGS	4.638	35	4.525	37	21.916	48	2.238	3
LBFGS	26.838	217	22.679	180	30.765	69	2.321	3
NEWTON	5.730	3	1.937	1	122.158	40	2.164	1
GD	96.142	1156	61.526	721	194.473	446	22.738	60
AGD	96.142	880	45.864	518	80.874	186	32.563	77

TABLE 1: DETAILS OF THE EXPERIMENTS PRESENTED IN FIGURES 2 AND 3.

Data set	n	p	Reference, UCI repo (Lichman, 2013)
CT slices	53500	386	Graf et al., 2011
Covertype	581012	54	Blackard and Dean, 1999
S3	500000	300	3-spiked model, (Donoho et al., 2013)
S10	500000	300	10-spiked model, (Donoho et al., 2013)
S20	500000	300	20-spiked model, (Donoho et al., 2013)

Table 2: Data sets used in the experiments.

5.3 Analysis of Number of Iterations

We provide additional plots to better understand the convergence behavior of the algorithms. Plots in Figure 4 show the decrease in $\log_{10}(\|\hat{\beta}^t - \beta_0\|_2)$ error over iterations (instead of time elapsed).

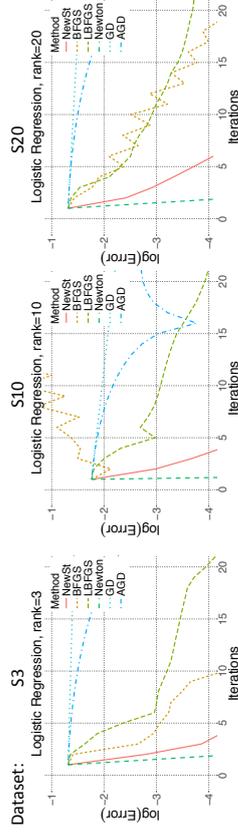


Figure 4: Figure shows the convergence behavior over the number of iterations. y and x axes denote $\log_{10}(\|\hat{\beta}^t - \beta_*\|_2)$ and the number iterations, respectively.

We observe from the plots that Newton’s method enjoys the fastest convergence rate as expected. The one that is closest to Newton’s method is the Newton-Stein method. This is because the Hessian estimator used by Newton-Stein method better approximates the true Hessian as opposed to Quasi-Newton methods. We emphasize that x axes in Figure 4 denote the number of iterations whereas in figures shown previously in this section x axes were the time elapsed.

6. Discussion

In this paper, we proposed an efficient algorithm for training GLMs. We call our algorithm Newton-Stein method (NewSt) as it takes a Newton-type step at each iteration relying on a Stein-type lemma. The algorithm requires a one time $\mathcal{O}(|S|p^2)$ cost to estimate the covariance structure and $\mathcal{O}(np)$ per-iteration cost to form the update equations. We observe that the convergence of Newton-Stein method has a phase transition from quadratic rate to linear rate. This observation is justified theoretically along with several other guarantees for the bounded as well as the sub-Gaussian covariates such as per-step convergence bounds, conditions for local rates and global convergence with line search, etc. Parameter selection guidelines of Newton-Stein method are based on our theoretical results. Our experiments show that Newton-Stein method provides significant improvement over the classical optimization methods.

Relaxing some of the theoretical constraints is an interesting line of research. In particular, strong assumptions on the cumulant generating functions might be loosened. Another interesting direction is to determine when the phase transition point occurs, which would provide a better understanding of the effects of sub-sampling and eigenvalue thresholding.

Acknowledgments

The author is grateful to Mohsen Bayati and Andrea Montanari for stimulating conversations on the topic of this work. The author would like to thank Bhaswar B. Bhattacharya and Qingyuan Zhao for carefully reading this article and providing valuable feedback.

Appendix A. Preliminary Concentration Inequalities

In this section, we provide several concentration bounds that will be useful throughout the proofs. We start by defining a special class of random variables.

Definition 10 (Sub-Gaussian) A random variable $x \in \mathbb{R}$ is called sub-Gaussian if it satisfies

$$\mathbb{E}[|x|^m]^{1/m} \leq K\sqrt{m}, \quad m \geq 1,$$

for some finite constant K . The smallest such K is the sub-Gaussian norm of x and it is denoted by $\|x\|_{\psi_2}$. Similarly, a random vector $y \in \mathbb{R}^p$ is called sub-Gaussian if there exists a constant $K' > 0$ such that

$$\sup_{v \in S^{p-1}} \|(y, v)\|_{\psi_2} \leq K'.$$

Definition 11 (Sub-exponential) A random variable $x \in \mathbb{R}$ is called sub-exponential if it satisfies

$$\mathbb{E}[|x|^m]^{1/m} \leq Km, \quad m \geq 1,$$

for some finite constant K . The smallest such K is the sub-exponential norm of x and it is denoted by $\|x\|_{\psi_1}$. Similarly, a random vector $y \in \mathbb{R}^p$ is called sub-exponential if there exists a constant $K' > 0$ such that

$$\sup_{v \in S^{p-1}} \|(y, v)\|_{\psi_1} \leq K'.$$

We state the following Lemmas from Vershynin, 2010 for the convenience of the reader (i.e., See Theorem 5.39 and the following remark for sub-Gaussian distributions, and Theorem 5.44 for distributions with arbitrary support):

Lemma 12 (Vershynin, 2010) Let S be an index set and $x_i \in \mathbb{R}^p$ for $i \in S$ be i.i.d. sub-Gaussian random vectors with

$$\mathbb{E}[x_i] = 0, \quad \mathbb{E}[x_i x_i^T] = \Sigma, \quad \|x_i\|_{\psi_2} \leq K.$$

There exists constants c, C depending only on the sub-Gaussian norm K such that with probability $1 - 2e^{-c^2}$,

$$\|\hat{\Sigma}_S - \Sigma\|_2 \leq \max(\delta, \delta^2) \quad \text{where} \quad \delta = C\sqrt{\frac{p}{|S|} + \frac{t}{\sqrt{|S|}}}.$$

Remark 13 We are interested in the case where $\delta < 1$, hence the right hand side becomes $\max(\delta, \delta^2) = \delta$. In most cases, we will simply let $t = \sqrt{p}$ and obtain a bound of order $\sqrt{p}/|S|$ on the right hand side. For this, we need $|S| = \mathcal{O}(C^2 p)$ which is a reasonable assumption in the regime we consider.

The following lemma is an analogue of Lemma 12 for covariates sampled from arbitrary distributions with bounded support.

Lemma 14 (Vershynin, 2010) *Let S be an index set and $x_i \in \mathbb{R}^p$ for $i \in S$ be i.i.d. random vectors with*

$$\mathbb{E}[x_i] = 0, \quad \mathbb{E}[x_i x_i^T] = \Sigma, \quad \|x_i\|_2 \leq \sqrt{K} \text{ a.s.}$$

Then, for some absolute constant c , with probability $1 - pe^{-ct^2}$, we have

$$\|\widehat{\Sigma}_S - \Sigma\|_2 \leq \max\left(\|\Sigma\|_2^{1/2} \delta, \delta^2\right) \quad \text{where} \quad \delta = t \sqrt{\frac{K}{|S|}}.$$

Remark 15 *We will choose $t = \sqrt{3 \log(p)/c}$ which will provide us with a probability of $1 - 1/p^2$. Therefore, if the sample size is sufficiently large, i.e.,*

$$|S| \geq \frac{3K \log(p)}{c \|\Sigma\|_2} = \mathcal{O}(K \log(p) / \|\Sigma\|_2),$$

we can estimate the true covariance matrix quite well for arbitrary distributions with bounded support. In particular, with probability $1 - 1/p^2$, we obtain

$$\|\widehat{\Sigma}_S - \Sigma\|_2 \leq c' \sqrt{\frac{\log(p)}{|S|}}, \tag{14}$$

where $c' = \sqrt{3K \|\Sigma\|_2 / c}$.

In the following, we will focus on empirical processes and obtain uniform bounds for proposed Hessian approximation. To that extent, we provide a few basic definitions which will be useful later in the proofs. For a more detailed discussion on the machinery used throughout the next section, we refer reader to Van der Vaart, 2000.

Definition 16 *On a metric space (X, d) , for $\epsilon > 0$, $T_\epsilon \subset X$ is called an ϵ -net over X if $\forall x \in X, \exists t \in T_\epsilon$ such that $d(x, t) \leq \epsilon$.*

In the following, we will use L_1 distance between two functions f and g , namely $d(f, g) = \int |f - g|$. Note that the same distance definition can be carried to random variables as they are simply real measurable functions. The integral takes the form of expectation.

Definition 17 *Given a function class \mathcal{F} , and any two functions l and u (not necessarily in \mathcal{F}), the bracket $[l, u]$ is the set of all $f \in \mathcal{F}$ such that $l \leq f \leq u$. A bracket satisfying $l \leq u$ and $\int |u - l| \leq \epsilon$ is called an ϵ -bracket in L_1 . The bracketing number $N_{[]}(\epsilon, \mathcal{F}, L_1)$ is the minimum number of different ϵ -brackets needed to cover \mathcal{F} .*

The preliminary tools presented in this section will be utilized to obtain the concentration results in Section B.

Appendix B. Main Lemmas

B.1 Concentration of Covariates With Bounded Support

Lemma 18 *Let $x_i \in \mathbb{R}^p$, for $i = 1, 2, \dots, n$, be i.i.d. random vectors supported on a ball of radius \sqrt{K} , with mean 0, and covariance matrix Σ . Further, let $f : \mathbb{R} \rightarrow \mathbb{R}$ be a uniformly bounded function such that for some $B > 0$, we have $\|f\|_\infty < B$ and f is Lipschitz continuous with constant L . Then, for sufficiently large n , there exist constants c_1, c_2, c_3 such that*

$$\mathbb{P}\left(\sup_{\beta \in B_p(R)} \left| \frac{1}{n} \sum_{i=1}^n f(x_i, \beta) - \mathbb{E}[f(x, \beta)] \right| > c_1 \sqrt{\frac{p \log(n)}{n}}\right) \leq c_2 e^{-c_3 p},$$

where the constants depend only on the bound B .

Proof We start by using the Lipschitz property of the function f , i.e., $\forall \beta, \beta' \in B_p(R)$,

$$\begin{aligned} |f(x, \beta) - f(x, \beta')| &\leq L \|x\|_2 \|\beta - \beta'\|_2, \\ &\leq L \sqrt{K} \|\beta - \beta'\|_2, \end{aligned}$$

where the first inequality follows from Cauchy-Schwartz. Now let T_Δ be a Δ -net over $B_p(R)$. Then $\forall \beta \in B_p(R)$, $\exists \beta' \in T_\Delta$ such that the right hand side of the above inequality is smaller than $\Delta L \sqrt{K}$. Then, we can write

$$\left| \frac{1}{n} \sum_{i=1}^n f(x_i, \beta) - \mathbb{E}[f(x, \beta)] \right| \leq \left| \frac{1}{n} \sum_{i=1}^n f(x_i, \beta') - \mathbb{E}[f(x, \beta')] \right| + 2\Delta L \sqrt{K}. \tag{14}$$

By choosing

$$\Delta = \frac{\epsilon}{4L\sqrt{K}},$$

and taking supremum over the corresponding β sets on both sides, we obtain the following inequality

$$\sup_{\beta \in B_p(R)} \left| \frac{1}{n} \sum_{i=1}^n f(x_i, \beta) - \mathbb{E}[f(x, \beta)] \right| \leq \max_{\beta \in T_\Delta} \left| \frac{1}{n} \sum_{i=1}^n f(x_i, \beta) - \mathbb{E}[f(x, \beta)] \right| + \frac{\epsilon}{2}.$$

Now, since we have $\|f\|_\infty \leq B$ and for a fixed β and $i = 1, 2, \dots, n$, the random variables $f(x_i, \beta)$ are i.i.d., by the Hoeffding's concentration inequality, we have

$$\mathbb{P}\left(\left| \frac{1}{n} \sum_{i=1}^n f(x_i, \beta) - \mathbb{E}[f(x, \beta)] \right| > \epsilon/2\right) \leq 2 \exp\left(-\frac{n\epsilon^2}{8B^2}\right).$$

Combining Equation 14 with the above result and a union bound, we easily obtain

$$\begin{aligned} &\mathbb{P}\left(\sup_{\beta \in B_p(R)} \left| \frac{1}{n} \sum_{i=1}^n f(x_i, \beta) - \mathbb{E}[f(x, \beta)] \right| > \epsilon\right) \\ &\leq \mathbb{P}\left(\max_{\beta \in T_\Delta} \left| \frac{1}{n} \sum_{i=1}^n f(x_i, \beta) - \mathbb{E}[f(x, \beta)] \right| > \epsilon/2\right) \leq 2|T_\Delta| \exp\left(-\frac{n\epsilon^2}{8B^2}\right), \end{aligned}$$

where $\Delta = \epsilon/4L\sqrt{K}$.

Next, we apply Lemma 33 and obtain that

$$|T_\Delta| \leq \left(\frac{R\sqrt{p}}{\Delta} \right)^p = \left(\frac{R\sqrt{p}}{\epsilon/4L\sqrt{K}} \right)^p.$$

We require that the probability of the desired event is bounded by a quantity that attains an exponential decay with rate $\mathcal{O}(p)$. This can be attained if

$$\epsilon^2 \geq \frac{8B^2p}{n} \log(4eLR\sqrt{K}\sqrt{p}/\epsilon).$$

Assuming that n is sufficiently large, and using Lemma 34 with $a = 8B^2p/n$ and $b = 4eLR\sqrt{K}p$, we obtain that ϵ should be

$$\epsilon = \sqrt{\frac{4B^2p}{n} \log\left(\frac{30L^2R^2Kn}{B^2}\right)} = \mathcal{O}\left(\sqrt{\frac{p \log(n)}{n}}\right).$$

When $n > 30L^2R^2K/B^2$, we obtain

$$\mathbb{P}\left(\sup_{\beta \in B_p(R)} \left| \frac{1}{n} \sum_{i=1}^n f(\langle x_i, \beta \rangle) - \mathbb{E}[f(\langle x, \beta \rangle)] \right| > 3B\sqrt{\frac{p \log(n)}{n}}\right) \leq 2e^{-p}.$$

■

In the following, we state similar bounds on functions of the following form

$$x \rightarrow f(\langle x, \beta \rangle) \langle x, v \rangle^2,$$

which appear in the summation that form the Hessian matrix.

Lemma 19 *Let $x_i \in \mathbb{R}^p$, for $i = 1, \dots, n$, be i.i.d. random vectors supported on a ball of radius \sqrt{K} , with mean 0, and covariance matrix Σ . Also let $f : \mathbb{R} \rightarrow \mathbb{R}$ be a uniformly bounded function such that for some $B > 0$, we have $\|f\|_\infty < B$ and f is Lipschitz continuous with constant L . Then, for $v \in S^{p-1}$ and sufficiently large n , there exist constants c_1, c_2, c_3 such that*

$$\mathbb{P}\left(\sup_{\beta \in B_p(R)} \left| \frac{1}{n} \sum_{i=1}^n f(\langle x_i, \beta \rangle) \langle x_i, v \rangle^2 - \mathbb{E}[f(\langle x, \beta \rangle) \langle x, v \rangle^2] \right| > c_1 \sqrt{\frac{p \log(n)}{n}}\right) \leq c_2 e^{-c_3 p},$$

where the constants depend only on the bound B and the radius \sqrt{K} .

Proof As in the proof of Lemma 18, we start by using the Lipschitz property of the function f , i.e., $\forall \beta, \beta' \in B_p(R)$,

$$\begin{aligned} \|f(\langle x, \beta \rangle) \langle x, v \rangle^2 - f(\langle x, \beta' \rangle) \langle x, v \rangle^2\|_2 &\leq L \|x\|_2^2 \|\beta - \beta'\|_2, \\ &\leq LK^{1.5} \|\beta - \beta'\|_2. \end{aligned}$$

For a net T_Δ , $\forall \beta \in B_p(R)$, $\exists \beta' \in T_\Delta$ such that right hand side of the above inequality is smaller than $\Delta LK^{1.5}$. Then, we can write

$$\begin{aligned} \left| \frac{1}{n} \sum_{i=1}^n f(\langle x_i, \beta \rangle) \langle x_i, v \rangle^2 - \mathbb{E}[f(\langle x, \beta \rangle) \langle x, v \rangle^2] \right| &\leq \left| \frac{1}{n} \sum_{i=1}^n f(\langle x_i, \beta' \rangle) \langle x_i, v \rangle^2 - \mathbb{E}[f(\langle x, \beta' \rangle) \langle x, v \rangle^2] \right| \\ &\quad + 2\Delta LK^{1.5}. \end{aligned} \quad (15)$$

This time, we choose

$$\Delta = \frac{\epsilon}{4LK^{1.5}},$$

and take the supremum over the corresponding feasible β -sets on both sides,

$$\begin{aligned} \sup_{\beta \in B_p(R)} \left| \frac{1}{n} \sum_{i=1}^n f(\langle x_i, \beta \rangle) \langle x_i, v \rangle^2 - \mathbb{E}[f(\langle x, \beta \rangle) \langle x, v \rangle^2] \right| \\ \leq \max_{\beta \in T_\Delta} \left| \frac{1}{n} \sum_{i=1}^n f(\langle x_i, \beta \rangle) \langle x_i, v \rangle^2 - \mathbb{E}[f(\langle x, \beta \rangle) \langle x, v \rangle^2] \right| + \frac{\epsilon}{2}. \end{aligned}$$

Now, since we have $\|f\|_\infty \leq B$ and for fixed β and v , $i = 1, 2, \dots, n$, $f(\langle x_i, \beta \rangle) \langle x_i, v \rangle^2$ are i.i.d. random variables. By the Hoeffding's concentration inequality, we write

$$\mathbb{P}\left(\left| \frac{1}{n} \sum_{i=1}^n f(\langle x_i, \beta \rangle) \langle x_i, v \rangle^2 - \mathbb{E}[f(\langle x, \beta \rangle) \langle x, v \rangle^2] \right| > \epsilon/2\right) \leq 2 \exp\left(-\frac{n\epsilon^2}{8B^2K^2}\right).$$

Using Equation 15 and the above result combined with the union bound, we easily obtain

$$\begin{aligned} \mathbb{P}\left(\sup_{\beta \in B_p(R)} \left| \frac{1}{n} \sum_{i=1}^n f(\langle x_i, \beta \rangle) \langle x_i, v \rangle^2 - \mathbb{E}[f(\langle x, \beta \rangle) \langle x, v \rangle^2] \right| > \epsilon\right) \\ \leq \mathbb{P}\left(\max_{\beta \in T_\Delta} \left| \frac{1}{n} \sum_{i=1}^n f(\langle x_i, \beta \rangle) \langle x_i, v \rangle^2 - \mathbb{E}[f(\langle x, \beta \rangle) \langle x, v \rangle^2] \right| > \epsilon/2\right) \\ \leq 2|T_\Delta| \exp\left(-\frac{n\epsilon^2}{8B^2K^2}\right), \end{aligned}$$

where $\Delta = \epsilon/4LK^{1.5}$. Using Lemma 33, we have

$$|T_\Delta| \leq \left(\frac{R\sqrt{p}}{\Delta} \right)^p = \left(\frac{R\sqrt{p}}{\epsilon/4LK^{1.5}} \right)^p.$$

As before, we require that the right hand side of above inequality gets a decay with rate $\mathcal{O}(p)$. Using Lemma 34 with $a = 8B^2K^2p/n$ and $b = 100LRK^{1.5}\sqrt{p}$, we obtain that ϵ should be

$$\epsilon = \sqrt{\frac{4B^2K^2p}{n} \log\left(\frac{50^2L^2R^2Kn}{B^2}\right)} = \mathcal{O}\left(\sqrt{\frac{p \log(n)}{n}}\right).$$

When $n > 50LKR^{1/2}/B$, we obtain

$$\mathbb{P}\left(\sup_{\beta \in B_p(R)} \left| \frac{1}{n} \sum_{i=1}^n f(\langle x_i, \beta \rangle) \langle x_i, v \rangle^2 - \mathbb{E}[f(\langle x, \beta \rangle) \langle x, v \rangle^2] \right| > 4BK \sqrt{\frac{p \log(n)}{n}}\right) \leq 2e^{-32p}.$$

The rate $-3.2p$ will be important later. \blacksquare

B.2 Concentration of Sub-Gaussian Covariates

In this section, we derive the analogues of the Lemmas 18 and 19 for sub-Gaussian covariates. Note that the Lemmas in this section are more general in the sense that they also cover the case where the covariates have bounded support. As a result, the resulting convergence coefficients are worse compared to the previous section.

Lemma 20 *Let $x_i \in \mathbb{R}^p$, for $i = 1, \dots, n$, be i.i.d. sub-Gaussian random vectors with mean 0, covariance matrix Σ and sub-Gaussian norm K . Also let $f : \mathbb{R} \rightarrow \mathbb{R}$ be a uniformly bounded function such that for some $B > 0$, we have $\|f\|_\infty < B$ and f is Lipschitz continuous with constant L . Then, there exists absolute constants c_1, c_2, c_3 such that*

$$\mathbb{P}\left(\sup_{\beta \in B_n(R)} \left| \frac{1}{n} \sum_{i=1}^n f(\langle x_i, \beta \rangle) - \mathbb{E}[f(\langle x, \beta \rangle)] \right| > c_1 \sqrt{\frac{p \log(n)}{n}}\right) \leq c_2 e^{-c_3 p},$$

where the constants depend only on the eigenvalues of Σ , bound B and radius R and sub-Gaussian norm K .

Proof We start by defining the brackets of the form

$$\begin{aligned} l_\beta(x) &= f(\langle x, \beta \rangle) - \frac{\|x\|_2}{4\mathbb{E}\|x\|_2}, \\ u_\beta(x) &= f(\langle x, \beta \rangle) + \frac{\|x\|_2}{4\mathbb{E}\|x\|_2}. \end{aligned}$$

Observe that the size of bracket $[l_\beta, u_\beta]$ is $\epsilon/2$, i.e., $\mathbb{E}[u_\beta - l_\beta] = \epsilon/2$. Now let T_Δ be a Δ -net over $B_p(R)$ where we use $\Delta = \epsilon/(4L\mathbb{E}\|x\|_2)$. Then $\forall \beta \in B_p(R)$, $\exists \beta' \in T_\Delta$ such that $f(\langle \cdot, \beta \rangle)$ falls into the bracket $[l_{\beta'}, u_{\beta'}]$. This can be seen by writing out the Lipschitz property of the function f . That is,

$$\begin{aligned} |f(\langle x, \beta \rangle) - f(\langle x, \beta' \rangle)| &\leq L\|x\|_2 \|\beta - \beta'\|_2, \\ &\leq \Delta L \|x\|_2, \end{aligned}$$

where the first inequality follows from Cauchy-Schwarz. Therefore, we conclude that

$$M_{\lceil \epsilon/2, \mathcal{F}, L, 1 \rceil} \leq |T_\Delta|$$

for the function class $\mathcal{F} = \{f(\langle \cdot, \beta \rangle) : \beta \in B_p(R)\}$. We further have $\forall \beta \in B_p(R)$, $\exists \beta' \in T_\Delta$ such that

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n f(\langle x_i, \beta \rangle) - \mathbb{E}[f(\langle x, \beta \rangle)] &\leq \frac{1}{n} \sum_{i=1}^n u_{\beta'}(x_i) - \mathbb{E}[u_{\beta'}(x)] + \frac{\epsilon}{2}, \\ \frac{1}{n} \sum_{i=1}^n f(\langle x_i, \beta \rangle) - \mathbb{E}[f(\langle x, \beta \rangle)] &\geq \frac{1}{n} \sum_{i=1}^n l_{\beta'}(x_i) - \mathbb{E}[l_{\beta'}(x)] - \frac{\epsilon}{2}. \end{aligned}$$

Using the above inequalities, we have, $\forall \beta \in B_p(R)$, $\exists \beta' \in T_\Delta$

$$\begin{aligned} &\left\{ \left[\frac{1}{n} \sum_{i=1}^n u_{\beta'}(x_i) - \mathbb{E}[u_{\beta'}(x)] \right] > \epsilon/2 \right\} \cup \left\{ \left[-\frac{1}{n} \sum_{i=1}^n l_{\beta'}(x_i) + \mathbb{E}[l_{\beta'}(x)] \right] > \epsilon/2 \right\} \supset \\ &\left\{ \frac{1}{n} \sum_{i=1}^n f(\langle x_i, \beta \rangle) - \mathbb{E}[f(\langle x, \beta \rangle)] > \epsilon \right\}. \end{aligned}$$

By the union bound, we obtain

$$\begin{aligned} &\mathbb{P}\left(\max_{\beta \in T_\Delta} \left[\frac{1}{n} \sum_{i=1}^n u_{\beta'}(x_i) - \mathbb{E}[u_{\beta'}(x)] \right] > \epsilon/2\right) + \mathbb{P}\left(\max_{\beta \in T_\Delta} \left[-\frac{1}{n} \sum_{i=1}^n l_{\beta'}(x_i) + \mathbb{E}[l_{\beta'}(x)] \right] > \epsilon/2\right) \\ &\geq \mathbb{P}\left(\sup_{\beta \in B_p(R)} \left| \frac{1}{n} \sum_{i=1}^n f(\langle x_i, \beta \rangle) - \mathbb{E}[f(\langle x, \beta \rangle)] \right| > \epsilon\right). \end{aligned} \quad (16)$$

In order to complete the proof, we need concentration inequalities for u_β and l_β . We state the following lemma.

Lemma 21 *There exists a constant C depending on the eigenvalues of Σ and B such that, for each $\beta \in B_p(R)$ and for some $0 < \epsilon < 1$, we have*

$$\begin{aligned} \mathbb{P}\left(\frac{1}{n} \sum_{i=1}^n u_\beta(x_i) - \mathbb{E}[u_\beta(x)] > \epsilon/2\right) &\leq 2e^{-Cn\epsilon^2}, \\ \mathbb{P}\left(\frac{1}{n} \sum_{i=1}^n l_\beta(x_i) - \mathbb{E}[l_\beta(x)] > \epsilon/2\right) &\leq 2e^{-Cn\epsilon^2}, \end{aligned}$$

where

$$C = \frac{c}{\left(B + \frac{\sqrt{2}K}{4\mu/\sqrt{p}}\right)^2}$$

for an absolute constant c .

Remark 22 *Note that $\mu = \mathbb{E}\|x\|_2 = \mathcal{O}(\sqrt{p})$ and hence $\mu/\sqrt{p} = \mathcal{O}(1)$.*

Proof By the relation between sub-Gaussian and sub-exponential norms, we have

$$\begin{aligned} \| |x| \|_2^2 \|_{\psi_2} &\leq \| |x| \|_2^2 \|_{\psi_1} \leq \sum_{i=1}^p \| |x_i| \|_{\psi_1}^2, \\ &\leq 2 \sum_{i=1}^p \| |x_i| \|_{\psi_2}^2, \\ &\leq 2K^2 p. \end{aligned} \quad (17)$$

Therefore $\| |x| \|_2 - \mathbb{E} \| |x| \|_2$ is a centered sub-Gaussian random variable with sub-Gaussian norm bounded above by $2K\sqrt{2p}$. We have,

$$\mathbb{E} \| |x| \|_2 = \mu.$$

Note that μ is actually of order \sqrt{p} . Assuming that the left hand side of the above equality is equal to $\sqrt{p}K'$ for some constant $K' > 0$, we can conclude that the random variable $u_\beta(x) = f(\langle x, \beta \rangle) + \epsilon \frac{\| |x| \|_2}{4\mathbb{E} \| |x| \|_2}$ is also sub-Gaussian with

$$\begin{aligned} \| u_\beta(x) \|_{\psi_2} &\leq B + \frac{\epsilon}{4\mathbb{E} \| |x| \|_2} \| |x| \|_2 \|_{\psi_2} \\ &\leq B + \frac{K\sqrt{2p}}{4\sqrt{p}K'} K\sqrt{2p} \\ &\leq B + C' \end{aligned}$$

where $C' = \sqrt{2}K/4K'$ is a constant and we also assumed $\epsilon < 1$. Now, define the function

$$g_\beta(x) = u_\beta(x) - \mathbb{E}[u_\beta(x)].$$

Note that $g_\beta(x)$ is a centered sub-Gaussian random variable with sub-Gaussian norm

$$\| g_\beta(x) \|_{\psi_2} \leq 2B + 2C'.$$

Then, by the Hoeffding-type inequality for the sub-Gaussian random variables, we obtain

$$\mathbb{P} \left(\left| \frac{1}{n} \sum_{i=1}^n g_\beta(x_i) \right| > \epsilon/2 \right) \leq 2e^{-c\epsilon^2/(B+C')^2}$$

where c is an absolute constant. The same argument also holds for $I_\beta(x)$. \blacksquare

Using the above lemma with the union bound over the set T_Δ , we can write

$$\mathbb{P} \left(\sup_{\beta \in B_n(R)} \left| \frac{1}{n} \sum_{i=1}^n f(\langle x_i, \beta \rangle) - \mathbb{E}[f(\langle x, \beta \rangle)] \right| > \epsilon \right) \leq 4|T_\Delta| e^{-Cn\epsilon^2}.$$

Since we can also write, by Lemma 33

$$\begin{aligned} |T_\Delta| &\leq \left(\frac{R\sqrt{p}}{\Delta} \right)^p \leq \left(\frac{4RL\mathbb{E} \| |x| \|_2 \sqrt{p}}{\epsilon} \right)^p, \\ &\leq \left(\frac{4\sqrt{2}RLKp}{\epsilon} \right)^p, \end{aligned}$$

and we observe that, for the constant $c' = 4\sqrt{2}RLK$,

$$\begin{aligned} \mathbb{P} \left(\sup_{\beta \in B_n(R)} \left| \frac{1}{n} \sum_{i=1}^n f(\langle x_i, \beta \rangle) - \mathbb{E}[f(\langle x, \beta \rangle)] \right| > \epsilon \right) &\leq 4 \left(\frac{4\sqrt{2}RLKp}{\epsilon} \right)^p e^{-Cn\epsilon^2}, \\ &= 4 \exp \{ p \log(c'p/\epsilon) - Cn\epsilon^2 \}. \end{aligned}$$

We will obtain an exponential decay of order p on the right hand side. For some constant h depending on n and p , if we choose $\epsilon = hp$, we need

$$h^2 \geq \frac{1}{Cnp} \log(c'/h).$$

By the Lemma 34, choosing $h^2 = \log(2c^2Cnp)/(2Cnp)$, we satisfy the above requirement. Note that for n large enough, the condition of the lemma is easily satisfied. Hence, for

$$\epsilon^2 = \frac{p \log(2c^2Cnp)}{2Cn} = \mathcal{O} \left(\frac{p \log(n)}{n} \right),$$

we obtain that there exists constants c_1, c_2, c_3 such that

$$\mathbb{P} \left(\sup_{\beta \in B_n(R)} \left| \frac{1}{n} \sum_{i=1}^n f(\langle x_i, \beta \rangle) - \mathbb{E}[f(\langle x, \beta \rangle)] \right| > c_1 \sqrt{\frac{p \log(n)}{n}} \right) \leq c_2 e^{-c_3 p},$$

where

$$\begin{aligned} c_1 &= \frac{3 \left(B + \frac{\sqrt{2}K}{4\sqrt{\text{Tr}(\Sigma)/p-16K^2}} \right)^2}{2c}, \\ c_2 &= 4, \\ c_3 &= \frac{1}{2} \log(7) \leq \frac{1}{2} \log(\log(64R^2L^2K^2C) + 6\log(p)). \end{aligned}$$

when $p > e$ and $64R^2L^2K^2C > e$. \blacksquare

In the following, we state the concentration results on the unbounded functions of the form

$$x \rightarrow f(\langle x, \beta \rangle) \langle x, v \rangle.$$

Functions of this type form the summands of the Hessian matrix in GLMs.

Lemma 23 *Let x_i , for $i = 1, \dots, n$, be i.i.d sub-Gaussian random variables with mean 0, covariance matrix Σ and sub-Gaussian norm K . Also let $f: \mathbb{R} \rightarrow \mathbb{R}$ be a uniformly bounded function such that for some $B > 0$, we have $\|f\|_\infty < B$ and f is Lipschitz continuous with constant L . Further, let $v \in \mathbb{R}^p$ such that $\|v\|_2 = 1$. Then, for n, p sufficiently large satisfying*

$$n^{0.2}/\log(n) \gtrsim p,$$

there exist constants c_1, c_2 depending on L, B, R and the eigenvalues of Σ such that, we have

$$\mathbb{P} \left(\sup_{\beta \in B_p(R)} \left| \frac{1}{n} \sum_{i=1}^n f(\langle x_i, \beta \rangle) \langle x_i, v \rangle^2 - \mathbb{E}[f(\langle x, \beta \rangle) \langle x, v \rangle^2] \right| > c_1 \sqrt{\frac{p}{n^{0.2} \log(n)}} \right) \leq c_2 e^{-p}.$$

Proof We define the brackets of the form

$$\begin{aligned} l_\beta(x) &= f(\langle x, \beta \rangle) \langle x, v \rangle^2 - \epsilon \frac{\|x\|_2^3}{4L\mathbb{E}[\|x\|_2^3]}, \\ u_\beta(x) &= f(\langle x, \beta \rangle) \langle x, v \rangle^2 + \epsilon \frac{\|x\|_2^3}{4\mathbb{E}[\|x\|_2^3]}, \end{aligned} \quad (18)$$

and we observe that the bracket $[l_\beta, u_\beta]$ has size $\epsilon/2$ in L_1 , that is,

$$\mathbb{E}[\|u_\beta(x) - l_\beta(x)\|] = \epsilon/2.$$

Next, for the following constant

$$\Delta = \frac{\epsilon}{4L\mathbb{E}[\|x\|_2^3]},$$

we define a Δ -net over $B_p(R)$ and call it \mathcal{T}_Δ . Then, $\forall \beta \in B_p(R)$, $\exists \beta' \in \mathcal{T}_\Delta$ such that $f(\langle \cdot, \beta \rangle) \langle \cdot, v \rangle^2$ belongs to the bracket $[l_{\beta'}, u_{\beta'}]$. This can be seen by writing the Lipschitz continuity of the function f , i.e.,

$$\begin{aligned} |f(\langle x, \beta \rangle) \langle x, v \rangle^2 - f(\langle x, \beta' \rangle) \langle x, v \rangle^2| &= \langle x, v \rangle^2 \{ |f(\langle x, \beta \rangle) - f(\langle x, \beta' \rangle)| \}, \\ &\leq L \|x\|_2^2 \|v\|_2^2 \langle x, \beta - \beta' \rangle, \\ &\leq L \|x\|_2^3 \|\beta - \beta'\|_2, \\ &\leq \Delta L \|x\|_2^3, \end{aligned}$$

where we used Cauchy-Schwartz to obtain the above inequalities. Hence, we may conclude that for the bracketing functions given in Equation 18, the corresponding bracketing number of the function class

$$\mathcal{F} = \{f(\langle \cdot, \beta \rangle) \langle \cdot, v \rangle^2 : \beta \in B_p(R)\}$$

is bounded above by the covering number of the ball of radius R for the given scale $\Delta = \epsilon/(4L\mathbb{E}[\|x\|_2^3])$, i.e.,

$$N_{[]}(\epsilon/2, \mathcal{F}, L_1) \leq |\mathcal{T}_\Delta|.$$

Next, we will upper bound the target probability using the bracketing functions u_β, l_β . We have $\forall \beta \in B_p(R)$, $\exists \beta' \in \mathcal{T}_\Delta$ such that

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n f(\langle x_i, \beta \rangle) \langle x_i, v \rangle^2 - \mathbb{E}[f(\langle x, \beta \rangle) \langle x, v \rangle^2] &\leq \frac{1}{n} \sum_{i=1}^n u_{\beta'}(x_i) - \mathbb{E}[u_{\beta'}(x)] + \frac{\epsilon}{2}, \\ \frac{1}{n} \sum_{i=1}^n f(\langle x_i, \beta \rangle) \langle x_i, v \rangle^2 - \mathbb{E}[f(\langle x, \beta \rangle) \langle x, v \rangle^2] &\geq \frac{1}{n} \sum_{i=1}^n l_{\beta'}(x_i) - \mathbb{E}[l_{\beta'}(x)] - \frac{\epsilon}{2}. \end{aligned}$$

Using the above inequalities, $\forall \beta \in B_p(R)$, $\exists \beta' \in \mathcal{T}_\Delta$, we can write

$$\begin{aligned} &\left\{ \left[\frac{1}{n} \sum_{i=1}^n u_{\beta'}(x_i) - \mathbb{E}[u_{\beta'}(x)] \right] > \epsilon/2 \right\} \cup \left\{ \left[-\frac{1}{n} \sum_{i=1}^n l_{\beta'}(x_i) + \mathbb{E}[l_{\beta'}(x)] \right] > \epsilon/2 \right\} \supset \\ &\left\{ \left[\frac{1}{n} \sum_{i=1}^n f(\langle x_i, \beta \rangle) \langle x_i, v \rangle^2 - \mathbb{E}[f(\langle x, \beta \rangle) \langle x, v \rangle^2] \right] > \epsilon \right\}. \end{aligned}$$

Hence, by the union bound, we obtain

$$\begin{aligned} &\mathbb{P} \left(\max_{\beta \in \mathcal{T}_\Delta} \left[\frac{1}{n} \sum_{i=1}^n u_\beta(x_i) - \mathbb{E}[u_\beta(x)] \right] > \epsilon/2 \right) + \mathbb{P} \left(\max_{\beta \in \mathcal{T}_\Delta} \left[-\frac{1}{n} \sum_{i=1}^n l_\beta(x_i) + \mathbb{E}[l_\beta(x)] \right] > \epsilon/2 \right) \\ &\geq \mathbb{P} \left(\sup_{\beta \in B_p(R)} \left[\frac{1}{n} \sum_{i=1}^n f(\langle x_i, \beta \rangle) \langle x_i, v \rangle^2 - \mathbb{E}[f(\langle x, \beta \rangle) \langle x, v \rangle^2] \right] > \epsilon \right). \end{aligned} \quad (19)$$

In order to complete the proof, we need one-sided concentration inequalities for u_β and l_β . Handling these functions is somewhat tedious since $\|x\|_2^3$ terms do not concentrate nicely. We state the following lemma.

Lemma 24 *For given $\alpha, \epsilon > 0$, and n sufficiently large such that, $\nu(n^\alpha, p, \epsilon, B, K, \Sigma) < \epsilon/4$ where*

$$\begin{aligned} \nu(n^\alpha, p, \epsilon, B, K, \Sigma) &= 2 \left(n^\alpha + \frac{6BK^2p}{c} \right) \exp \left(-\frac{n^\alpha}{6BK^2p} \right) + 2 \left\{ n^\alpha + \frac{3K^2p}{c\text{Tr}(\Sigma)} e^{2/3} \right. \\ &\quad \left. + \frac{3K^4p^2}{c^2\text{Tr}(\Sigma)^2} e^{4/3} n^{-\alpha/3} \right\} \exp \left(-\frac{\text{Tr}(\Sigma)(n^\alpha/\epsilon)^{2/3}}{2K^2p} \right). \end{aligned}$$

Then, there exists constants c', c'', c''' depending on the eigenvalues of Σ , B and K such that $\forall \beta$, we have,

$$\begin{aligned} &\mathbb{P} \left(\frac{1}{n} \sum_{i=1}^n u_\beta(x_i) - \mathbb{E}[u_\beta(x)] > \epsilon/2 \right) \leq 2 \exp(-c'n^\alpha/p) \\ &\quad + 2 \exp(-c''n^{2\alpha/3}\epsilon^{-2/3}) + \exp(-c'''n^{1-2\alpha}\epsilon^2), \end{aligned}$$

and

$$\begin{aligned} &\mathbb{P} \left(-\frac{1}{n} \sum_{i=1}^n l_\beta(x_i) + \mathbb{E}[l_\beta(x)] > \epsilon/2 \right) \leq 2 \exp(-c'n^\alpha/p) + \\ &\quad 2 \exp(-c''n^{2\alpha/3}\epsilon^{-2/3}) + \exp(-c'''n^{1-2\alpha}\epsilon^2). \end{aligned}$$

Proof For the sake of simplicity, we define the functions

$$\begin{aligned}\tilde{u}_\beta(w) &= u_\beta(w) - \mathbb{E}[u_\beta(x)], \\ \tilde{l}_\beta(w) &= l_\beta(w) - \mathbb{E}[l_\beta(x)].\end{aligned}$$

We will derive the result for the upper bracket, \tilde{u} , and skip the proof for the lower bracket \tilde{l} as it follows from the same steps. We write,

$$\begin{aligned}\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^n \tilde{u}_\beta(x_i) > \epsilon/2\right) &\leq \mathbb{P}\left(\frac{1}{n}\sum_{i=1}^n \tilde{u}_\beta(x_i) > \epsilon/2, \max_{1 \leq i \leq n} |\tilde{u}_\beta(x_i)| < n^\alpha\right) \\ &\quad + \mathbb{P}\left(\max_{1 \leq i \leq n} |\tilde{u}_\beta(x_i)| \geq n^\alpha\right).\end{aligned}\tag{20}$$

We need to bound the right hand side of the above equation. For the second term, since $\tilde{u}_\beta(x_i)$'s are i.i.d. centered random variables, we have

$$\begin{aligned}\mathbb{P}\left(\max_{1 \leq i \leq n} |\tilde{u}_\beta(x_i)| \geq n^\alpha\right) &= 1 - \mathbb{P}\left(\max_{1 \leq i \leq n} |\tilde{u}_\beta(x_i)| < n^\alpha\right), \\ &= 1 - \mathbb{P}(|\tilde{u}_\beta(x)| < n^\alpha)^n, \\ &= 1 - (1 - \mathbb{P}(\tilde{u}_\beta(x) \geq n^\alpha))^n, \\ &\leq n\mathbb{P}(|\tilde{u}_\beta(x)| \geq n^\alpha).\end{aligned}$$

Also, note that

$$\begin{aligned}|\tilde{u}_\beta(x)| &\leq \mathcal{B}\|x\|_2^2 + \epsilon \frac{\|x\|_3^2}{4\mathbb{E}[\|x\|_2^2]} + \mathbb{E}[u_\beta(x)], \\ &\leq \mathcal{B}\|x\|_2^2 + \epsilon \frac{\|x\|_3^2}{4\mathbb{E}[\|x\|_2^2]} + \mathcal{B}\lambda_{\max}(\Sigma) + \epsilon/4.\end{aligned}$$

Therefore, if $t > 3\mathcal{B}\lambda_{\max}(\Sigma)$ and for ϵ small, we can write

$$\{\tilde{u}_\beta(x) > t\} \subset \{\mathcal{B}\|x\|_2^2 > t/3\} \cup \left\{\epsilon \frac{\|x\|_3^2}{4\mathbb{E}[\|x\|_2^2]} > t/3\right\}.\tag{21}$$

Since x is a sub-Gaussian random variable with $\|x\|_{\psi_2} = K$, we have

$$K = \sup_{w \in S^{p-1}} \langle w, x \rangle \| \psi_2 = \|x\|_{\psi_2}.$$

Using this and the relation between sub-Gaussian and sub-exponential norms as in Equation 17, we have $\|x\|_2^2 \leq 2K^2p$. This provides the following tail bound for $\|x\|_2$,

$$\mathbb{P}(\|x\|_2 > s) \leq 2 \exp\left(-\frac{cs^2}{2pK^2}\right),\tag{22}$$

where c is an absolute constant. Using the above tail bound, we can write,

$$\mathbb{P}\left(\|x\|_2^2 > \frac{1}{3B}t\right) \leq 2 \exp\left(-\frac{t}{6BK^2p}\right).$$

For the next term in Equation 21, we need a lower bound for $\mathbb{E}[\|x\|_2^2]$. We use a modified version of the Hölder's inequality and obtain

$$\mathbb{E}[\|x\|_2^3] \geq \mathbb{E}[\|x\|_2^2]^{3/2} = \text{Tr}(\Sigma)^{3/2}.$$

Using the above inequality, we can write

$$\begin{aligned}\mathbb{P}\left(\frac{\epsilon\|x\|_3^2}{4\mathbb{E}[\|x\|_2^2]} > t/3\right) &\leq \mathbb{P}\left(\|x\|_3^2 > \frac{4}{3\epsilon}\text{Tr}(\Sigma)^{3/2}t\right), \\ &= \mathbb{P}\left(\|x\|_2 > \left(\frac{4t}{3\epsilon}\right)^{1/3}\text{Tr}(\Sigma)^{1/2}\right), \\ &\leq 2 \exp\left(-c\frac{\text{Tr}(\Sigma)(t/\epsilon)^{2/3}}{2K^2p}\right),\end{aligned}$$

where c is the same absolute constant as in Equation 22.

Now for $\alpha > 0$ such that $t = n^\alpha > 3\mathcal{B}\lambda_{\max}(\Sigma)$ (we will justify this assumption for a particular choice of α later), we combine the above results,

$$\mathbb{P}(\tilde{u}_\beta(x) > t) \leq 2 \exp\left(-\frac{t}{6BK^2p}\right) + 2 \exp\left(-c\frac{\text{Tr}(\Sigma)(t/\epsilon)^{2/3}}{2K^2p}\right).\tag{23}$$

Next, we focus on the first term in Equation 20. Let $\mu = \mathbb{E}[\tilde{u}_\beta(x)\mathbb{1}_{\{\tilde{u}_\beta(x) < n^\alpha\}}]$, and write

$$\begin{aligned}\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^n \tilde{u}_\beta(x_i) > \frac{\epsilon}{2}; \max_{1 \leq i \leq n} |\tilde{u}_\beta(x_i)| < n^\alpha\right) &\leq \mathbb{P}\left(\frac{1}{n}\sum_{i=1}^n \tilde{u}_\beta(x_i)\mathbb{1}_{\{\tilde{u}_\beta(x_i) < n^\alpha\}} > \frac{\epsilon}{2}\right), \\ &= \mathbb{P}\left(\frac{1}{n}\sum_{i=1}^n \tilde{u}_\beta(x_i)\mathbb{1}_{\{\tilde{u}_\beta(x_i) < n^\alpha\}} - \mu > \frac{\epsilon}{2} - \mu\right) \\ &\leq \exp\left\{-\frac{n^{1-2\alpha}}{2}\left(\frac{\epsilon}{2} - \mu\right)^2\right\},\end{aligned}$$

where we used the Hoeffding's concentration inequality for the bounded random variables. Further, note that

$$0 = \mathbb{E}[\tilde{u}_\beta(x)] = \mu + \mathbb{E}\left[\tilde{u}_\beta(x)\mathbb{1}_{\{\tilde{u}_\beta(x) > n^\alpha\}}\right].$$

By Lemma 30, we can write

$$|\mu| = \left|\mathbb{E}\left[\tilde{u}_\beta(x)\mathbb{1}_{\{\tilde{u}_\beta(x) > n^\alpha\}}\right]\right| \leq n^\alpha \mathbb{P}(\tilde{u}_\beta(x) > n^\alpha) + \int_{n^\alpha}^{\infty} \mathbb{P}(|\tilde{u}_\beta(x)| > t)dt.$$

The first term on the right hand side can be easily bounded by using Equation 23, i.e.,

$$n^\alpha \mathbb{P}(|\bar{u}_\beta(x)| > n^\gamma) \leq 2n^\alpha \exp\left(-c \frac{n^\alpha}{6BK^2p}\right) + 2n^\alpha \exp\left(-c \frac{\text{Tr}(\Sigma)(n^\alpha/\epsilon^{2/3})}{2K^2p}\right).$$

For the second term, using Equation 23 once again, we obtain

$$\begin{aligned} \int_{n^\alpha}^\infty \mathbb{P}(|\bar{u}_\beta(x)| > t) dt &\leq 2 \int_{n^\alpha}^\infty \exp\left(-c \frac{t}{6BK^2p}\right) dt + 2 \int_{n^\alpha}^\infty \exp\left(-c \frac{\text{Tr}(\Sigma)(t/\epsilon^{2/3})}{2K^2p}\right) dt, \\ &= \frac{12BK^2p}{c} \exp\left(-c \frac{n^\alpha}{6BK^2p}\right) + 2 \int_{n^\alpha}^\infty \exp\left(-c \frac{\text{Tr}(\Sigma)(t/\epsilon^{2/3})}{2K^2p}\right) dt. \end{aligned}$$

Next, we apply Lemma 31 to bound the second term on the right hand side. That is, we have

$$\begin{aligned} &\int_{n^\alpha}^\infty \exp\left(-c \frac{\text{Tr}(\Sigma)(t/\epsilon^{2/3})}{2K^2p}\right) dt \\ &\leq \left\{ \frac{3K^2p}{c \text{Tr}(\Sigma)} n^{\alpha/3} \epsilon^{2/3} + \frac{3K^4p^2}{c^2 \text{Tr}(\Sigma)^2} \epsilon^{4/3} n^{-\alpha/3} \right\} \exp\left(-c \frac{\text{Tr}(\Sigma)(n^\alpha/\epsilon^{2/3})}{2K^2p}\right). \end{aligned}$$

Combining the above results, we can write

$$\begin{aligned} |\mu| &\leq 2 \left(n^\alpha + \frac{6BK^2p}{c} \right) \exp\left(-c \frac{n^\alpha}{6BK^2p}\right) \\ &\quad + 2 \left\{ n^\alpha + \frac{3K^2p}{c \text{Tr}(\Sigma)} n^{\alpha/3} \epsilon^{2/3} + \frac{3K^4p^2}{c^2 \text{Tr}(\Sigma)^2} \epsilon^{4/3} n^{-\alpha/3} \right\} \exp\left(-c \frac{\text{Tr}(\Sigma)(n^\alpha/\epsilon^{2/3})}{2K^2p}\right), \\ &=: \nu(n^\alpha, p, \epsilon, B, K, \Sigma). \end{aligned}$$

Notice that, the upper bound on $|\mu|$, namely $\nu(n^\alpha, p, \epsilon, B, K, \Sigma)$, is close to 0 when n is large. This is because of exponentially decaying functions that dominates the other terms. We assume that n is sufficiently large that the upper bound for $|\mu|$ is less than $\epsilon/4$. For the value of α , we will choose $\alpha = 0.4$ later in the proof.

Applying this bounds in Equation 20, we obtain

$$\begin{aligned} \mathbb{P}\left(\frac{1}{n} \sum_{i=1}^n u_\beta(x_i) > \epsilon/2\right) &\leq 2 \exp\left(-c \frac{n^\alpha}{6BK^2p}\right) \\ &\quad + 2 \exp\left(-c \frac{\text{Tr}(\Sigma)(n^\alpha/\epsilon^{2/3})}{2K^2p}\right) + \exp\left(-\frac{n^{1-2\alpha} \epsilon^2}{32}\right), \\ &= 2 \exp\left(-c' n^\alpha/p\right) + 2 \exp\left(-c'' n^{2\alpha/3} \epsilon^{-2/3}\right) + \exp\left(-c''' n^{1-2\alpha} \epsilon^2\right), \end{aligned}$$

where

$$\begin{aligned} c' &= \frac{c}{6BK^2}, \\ c'' &= \frac{c \text{Tr}(\Sigma)/p}{2K^2} \geq \frac{c \lambda_{\min}(\Sigma)}{2K^2}, \\ c''' &= \frac{1}{32}. \end{aligned}$$

Hence, the proof is completed for the upper bracket.

The proof for the lower brackets $l_\beta(x)$ follows from exactly the same steps and omitted here. \blacksquare

Applying the above lemma on Equation 19, for $\alpha > 0$, we obtain

$$\begin{aligned} &\mathbb{P}\left(\sup_{\beta \in B_n(n)} \left| \frac{1}{n} \sum_{i=1}^n f(x_i, \beta) \langle x_i, v \rangle^2 - \mathbb{E}[f(x, \beta) \langle x, v \rangle^2] \right| > \epsilon\right) \\ &\leq 4|T_\Delta| \exp(-c' n^\alpha/p) + 4|T_\Delta| \exp(-c'' n^{2\alpha/3} \epsilon^{-2/3}) + 2|T_\Delta| \exp(-c''' n^{1-2\alpha} \epsilon^2). \end{aligned} \quad (24)$$

Observe that we can write, by Lemma 33

$$|T_\Delta| \leq \left(\frac{R\sqrt{p}}{\Delta}\right)^p = \left(\frac{4\sqrt{p}RL\mathbb{E}\|x\|_2^3}{\epsilon}\right)^p.$$

Also, recall that $\|x\|_2$ was a sub-Gaussian random variable with $\| \|x\|_2 \|_{\psi_2} \leq K\sqrt{2p}$. Using the definition of sub-Gaussian norm, we have

$$\frac{1}{\sqrt{3}} \mathbb{E}\|x\|_2^{3/2} \leq \| \|x\|_2 \|_{\psi_2} \leq \sqrt{2p}K, \implies \mathbb{E}\|x\|_2^3 \leq 15K^3 p^{3/2}.$$

Therefore, we have $\mathbb{E}\|x\|_2^3 = \mathcal{O}(p^{3/2})$ (recall that we had a lower bound of the same order). We define a constant K' , and as ϵ is small, we have

$$|T_\Delta| \leq \left(\frac{60RLK^3 p^2}{\epsilon}\right)^p = \left(\frac{K' p^2}{\epsilon}\right)^p,$$

where we let $K' = 60RLK^3$. We will show that each term on the right hand side of Equation 24 decays exponentially with a rate of order p . For the first term, for $s > 0$, we write

$$\begin{aligned} |T_\Delta| \exp(-c' n^\alpha/p) &= \exp(-c' n^\alpha/p + p \log(K') + 2p \log(p) + p \log(\epsilon^{-1})), \\ &\leq \exp(-c' n^\alpha/p + 2p \log(K' p/\epsilon)). \end{aligned} \quad (25)$$

Similarly for the second and third terms, we write

$$\begin{aligned} |T_\Delta| \exp(-c'' n^{2\alpha/3} \epsilon^{-2/3}) &\leq \exp(-c'' n^{2\alpha/3} \epsilon^{-2/3} + 2p \log(K' p/\epsilon)), \\ |T_\Delta| \exp(-c''' n^{1-2\alpha} \epsilon^2) &\leq \exp(-c''' n^{1-2\alpha} \epsilon^2 + 2p \log(K' p/\epsilon)). \end{aligned} \quad (26)$$

We will seek values for ϵ and α to obtain an exponential decay with rate p on the right sides of Equations 25 and 26. That is, we need

$$c'n^\alpha/p \geq 2p \log(K''/p/\epsilon), \quad (27)$$

$$c''n^{2\alpha/3} \geq 2p \log(K''/p/\epsilon)e^{2/3},$$

$$c''n^{1-2\alpha}e^2 \geq 2p \log(K''/p/\epsilon),$$

where $K'' = eK'$.

We apply Lemma 34 for the last inequality in Equation 27. That is,

$$\begin{aligned} e^2 &= \frac{p}{c''n^{1-2\alpha}} \log\left(\frac{c''n^{1/2}pn^{1-2\alpha}}{c''n^{1-2\alpha}}\right), \\ &= \mathcal{O}\left(\frac{p}{n^{1-2\alpha}} \log(n)\right). \end{aligned} \quad (28)$$

where we assume that n is sufficiently large. The above statement holds for $\alpha < 1/2$.

In the following, we choose $\alpha = 0.4$ and use the assumption that

$$n^{0.2}/\log(n) \gtrsim p, \quad (29)$$

which provides $\epsilon < 1$. Note that this choice of α also justifies the assumption used to derive Equation 23. One can easily check that $\alpha = 0.4$ implies that the first and the second statements in Equation 27 are satisfied for sufficiently large n .

It remains to check whether $\nu(n^\alpha, p, \epsilon, B, K, \Sigma) < \epsilon/4$ (in Lemma 24) for this particular choice of α and ϵ . It suffices to consider only the dominant terms in the definition of ν . We use the assumption on n, p and write

$$\begin{aligned} \nu(n^{0.4}, p, \epsilon, B, K, \Sigma) &\lesssim n^{0.4} \exp\left(-\frac{cn^{0.4}}{6BK^2p}\right) + n^{0.4} \exp\left(-\frac{c\text{Tr}(\Sigma)/p}{2K^2}n^{0.8/3}\right), \\ &\lesssim n^{0.4} \exp\left(-\frac{c}{6BK^2}n^{0.2}\right) + n^{0.4} \exp\left(-\frac{c\lambda_{\min}(\Sigma)}{2K^2}n^{0.8/3}\right). \end{aligned} \quad (30)$$

For n sufficiently large, due to exponential decay in $n^{0.2}$, the above quantity can be made arbitrarily small. Hence, for some constants c_1, c_2 , we obtain

$$\mathbb{P}\left(\sup_{\beta \in B_p(\hat{\beta})} \left| \frac{1}{n} \sum_{i=1}^n f(x_i, \beta) \langle x_i, v \rangle^2 - \mathbb{E}[f(x, \beta)] \langle x, v \rangle^2 \right| > c_1 \sqrt{\frac{p}{n^{0.2}} \log(n)}\right) \leq c_2 e^{-p}. \quad \blacksquare$$

Appendix C. Proofs of Theorems 4 and 8

We will provide the proofs of Theorems 4 and 8 in parallel as they follow from similar steps. The only difference is the application of the lemmas that are provided in the previous

sections. On the event \mathcal{E} , we write,

$$\begin{aligned} \hat{\beta}^t - \beta_* - \gamma \mathbf{Q}^t \nabla_{\beta} \ell(\hat{\beta}^t) &= \hat{\beta}^t - \beta_* - \gamma \mathbf{Q}^t \int_0^1 \nabla_{\beta}^2 \ell(\beta_* + \xi(\hat{\beta}^t - \beta_*)) d\xi (\hat{\beta}^t - \beta_*), \\ &= \left(I - \gamma \mathbf{Q}^t \int_0^1 \nabla_{\beta}^2 \ell(\beta_* + \xi(\hat{\beta}^t - \beta_*)) d\xi \right) (\hat{\beta}^t - \beta_*). \end{aligned} \quad (31)$$

In the following, we will work on the event that $\hat{\Sigma}_S$ is invertible and that $[\mathbf{Q}^j]^{-1}$ is positive definite. We later show that conditioned on \mathcal{E} , this event holds with very high probability when $|S|$ is sufficiently large.

We use the nonexpensiveness of the projection \mathcal{P}_C^t , i.e., for any $u, u' \in \mathbb{R}^{\mathcal{V}}$ and $v = \mathcal{P}_C^t(u)$, $v' = \mathcal{P}_C^t(u')$ we have $\langle u - u', [\mathbf{Q}^j]^{-1}(u - u') \rangle \geq \langle v - v', [\mathbf{Q}^j]^{-1}(v - v') \rangle$. This simply means that the projection decreases the distance. Therefore, we can write

$$\begin{aligned} \|\hat{\beta}^{t+1} - \beta_*\|_{\mathbf{Q}^{t-1}} &\leq \|\hat{\beta}^t - \beta_* - \gamma \mathbf{Q}^t \nabla_{\beta} \ell(\hat{\beta}^t)\|_{\mathbf{Q}^{t-1}} \\ &\leq \|[\mathbf{Q}^j]^{-1/2} - \gamma [\mathbf{Q}^j]^{1/2} \int_0^1 \nabla_{\beta}^2 \ell(\beta_* + \xi(\hat{\beta}^t - \beta_*)) d\xi\|_2 \|\hat{\beta}^t - \beta_*\|_2. \end{aligned} \quad (32)$$

The coefficient of $\|\hat{\beta}^t - \beta_*\|_2$ in Equation 32 determines the convergence behavior of the algorithm. Switching back to ℓ_2 norm, we obtain an upper bounded of the form

$$\|\hat{\beta}^{t+1} - \beta_*\|_2 \leq \|\mathbf{Q}^j\|_2 \|[\mathbf{Q}^j]^{-1} - \int_0^1 \nabla_{\beta}^2 \ell(\beta_* + \xi(\hat{\beta}^t - \beta_*)) d\xi\|_2 \|\hat{\beta}^t - \beta_*\|_2,$$

where we have set step size $\gamma = 1$. First, we will bound the second term on the right hand side. We define the following,

$$\mathfrak{E}(\beta) = \mathbb{E}\left[\phi^{(2)}(\langle x, \beta \rangle)\right] \Sigma + \mathbb{E}\left[\phi^{(4)}(\langle x, \beta \rangle)\right] \Sigma \beta \beta^T \Sigma.$$

Note that for a function f and fixed β , $\mathbb{E}[f(\langle x, \beta \rangle)] = h(\beta)$ is a function of β . With a slight abuse of notation, we write $\mathbb{E}[f(\langle x, \hat{\beta} \rangle)] = h(\hat{\beta})$ as a random variable. We have

$$\begin{aligned} \|[\mathbf{Q}^j]^{-1} - \int_0^1 \nabla_{\beta}^2 \ell(\beta_* + \xi(\hat{\beta}^t - \beta_*)) d\xi\|_2 &\leq \|[\mathbf{Q}^j]^{-1} - \mathfrak{E}(\hat{\beta}^t)\|_2 \\ &+ \|\mathbb{E}[xx^T \phi^{(2)}(\langle x, \hat{\beta}^t \rangle)] - \mathfrak{E}(\hat{\beta}^t)\|_2 \\ &+ \left\| \int_0^1 \nabla_{\beta}^2 \ell(\beta_* + \xi(\hat{\beta}^t - \beta_*)) d\xi - \mathbb{E}\left[xx^T \int_0^1 \phi^{(2)}(\langle x, \beta_* + \xi(\hat{\beta}^t - \beta_*) \rangle) d\xi\right] \right\|_2 \\ &+ \|\mathbb{E}[xx^T \phi^{(2)}(\langle x, \hat{\beta}^t \rangle)] - \mathbb{E}\left[xx^T \int_0^1 \phi^{(2)}(\langle x, \beta_* + \xi(\hat{\beta}^t - \beta_*) \rangle) d\xi\right] \|_2. \end{aligned} \quad (33)$$

For the first term on the right hand side, we state the following lemma.

Lemma 25 *When the covariates are sub-Gaussian, there exist constants C_1, C_2 such that, with probability at least $1 - C_1/p^2$,*

$$\|\mathbf{Q}^{\dagger-1} - \mathbf{e}(\beta^t)\|_2 \leq C_2 \sqrt{\frac{p}{\min\{|S|p/\log(p), n/\log(n)\}}}.$$

Similarly, when the covariates are sampled from a distribution with bounded support, there exist constants C_1^t, C_2^t, C_3^t such that, with probability $1 - C_1^t e^{-C_2^t p}$,

$$\|\mathbf{Q}^{\dagger-1} - \mathbf{e}(\beta^t)\|_2 \leq C_3^t \sqrt{\frac{p}{\min\{|S|, n/\log(n)\}}},$$

where the constants depend on K, B and the radius R .

Proof In the following, we will only provide the proof for the bounded support case. The proof for the sub-Gaussian covariates follows from the same steps, by only replacing Lemma 14 with Lemma 12, and Lemma 18 with Lemma 20.

Using a uniform bound on the feasible set, we write

$$\begin{aligned} & \|\mathbf{Q}^{\dagger-1} - \mathbf{e}(\beta^t)\|_2 \\ & \leq \sup_{\beta \in \mathcal{C}^t} \|\hat{\mu}_2(\beta) \widehat{\Sigma}_S + \hat{\mu}_4(\beta) \widehat{\Sigma}_S \beta \widehat{\Sigma}_S \beta^T - \mathbb{E}[\phi^{(2)}(\langle x, \beta \rangle)] \Sigma - \mathbb{E}[\phi^{(4)}(\langle x, \beta \rangle)] \Sigma \beta \beta^T \Sigma\|_2. \end{aligned}$$

We will find an upper bound for the quantity inside the supremum. By denoting the expectations of $\hat{\mu}_2(\beta)$ and $\hat{\mu}_4(\beta)$, with $\mu_2(\beta)$ and $\mu_4(\beta)$ respectively, we write

$$\begin{aligned} & \|\hat{\mu}_2(\beta) \widehat{\Sigma}_S + \hat{\mu}_4(\beta) \widehat{\Sigma}_S \beta \widehat{\Sigma}_S \beta^T - \mathbb{E}[\phi^{(2)}(\langle x, \beta \rangle)] \Sigma - \mathbb{E}[\phi^{(4)}(\langle x, \beta \rangle)] \Sigma \beta \beta^T \Sigma\|_2 \\ & \leq \|\hat{\mu}_2(\beta) \widehat{\Sigma}_S - \mu_2(\beta) \Sigma\|_2 + \|\hat{\mu}_4(\beta) \widehat{\Sigma}_S \beta \widehat{\Sigma}_S \beta^T - \mu_4(\beta) \Sigma \beta \beta^T \Sigma\|_2. \end{aligned}$$

For the first term on the right hand side, we have

$$\begin{aligned} & \|\hat{\mu}_2(\beta) \widehat{\Sigma}_S - \mu_2(\beta) \Sigma\|_2 \leq |\hat{\mu}_2(\beta)| \|\widehat{\Sigma}_S - \Sigma\|_2 + \|\Sigma\|_2 |\hat{\mu}_2(\beta) - \mu_2(\beta)|, \\ & \leq B_2 \|\widehat{\Sigma}_S - \Sigma\|_2 + K |\hat{\mu}_2(\beta) - \mu_2(\beta)|. \end{aligned}$$

By the Lemmas 14 and 18, for an absolute constant c , we have with probability $1 - 1/p^2$,

$$\begin{aligned} & \sup_{\beta \in \mathcal{C}^t} \|\hat{\mu}_2(\beta) \zeta_r(\widehat{\Sigma}_S) - \mu_2(\beta) \Sigma\|_2 \leq B_2 c \sqrt{K \|\Sigma\|_2} \sqrt{\frac{\log(p)}{|S|}} + 3B_2 K \sqrt{\frac{p \log(n)}{n}}, \\ & \leq 3cB_2 K \sqrt{\frac{p}{\min\{p/\log(p), |S|, n/\log(n)\}}}, \\ & = \mathcal{O}\left(\sqrt{\frac{p}{\min\{p/\log(p), |S|, n/\log(n)\}}}\right). \end{aligned}$$

For the second term, we have

$$\begin{aligned} & \|\hat{\mu}_4(\beta) \widehat{\Sigma}_S \beta \widehat{\Sigma}_S \beta^T - \mu_4(\beta) \Sigma \beta \beta^T \Sigma\|_2 \\ & \leq |\hat{\mu}_4(\beta)| \|\widehat{\Sigma}_S \beta \beta^T \widehat{\Sigma}_S - \Sigma \beta \beta^T \Sigma\|_2 + |\hat{\mu}_4(\beta) - \mu_4(\beta)| \|\Sigma \beta \beta^T \Sigma\|_2, \\ & \leq B_4 R^2 \left\{ \|\widehat{\Sigma}_S\|_2 + \|\Sigma\|_2 \right\} \|\widehat{\Sigma}_S - \Sigma\|_2 + R^2 \|\Sigma\|_2^2 |\hat{\mu}_4(\beta) - \mu_4(\beta)|, \\ & \leq B_4 R^2 \left\{ \|\widehat{\Sigma}_S\|_2 + K \right\} \|\widehat{\Sigma}_S - \Sigma\|_2 + R^2 K^2 |\hat{\mu}_4(\beta) - \mu_4(\beta)|. \end{aligned}$$

Again, by the Lemmas 14 and 18, for an absolute constant c , we have with probability $1 - 1/p^2$,

$$\begin{aligned} & B_4 R^2 \left\{ \|\widehat{\Sigma}_S\|_2 + K \right\} \|\widehat{\Sigma}_S - \Sigma\|_2 \leq cK B_4 R^2 \left\{ 2K + cK \sqrt{\frac{\log(p)}{|S|}} \right\} \sqrt{\frac{\log(p)}{|S|}}, \\ & \leq 2cK^2 B_4 R^2 \sqrt{\frac{\log(p)}{|S|}} + c^2 K^2 B_4 R^2 \frac{\log(p)}{|S|}, \\ & \leq 2cK^2 B_4 R^2 \left(1 + c_1 \sqrt{\frac{\log(p)}{|S|}} \right) \sqrt{\frac{\log(p)}{|S|}}, \\ & \leq 4cK^2 B_4 R^2 \sqrt{\frac{\log(p)}{|S|}}, \\ & = \mathcal{O}\left(\sqrt{\frac{\log(p)}{|S|}}\right), \end{aligned}$$

for sufficiently large $|S|$, i.e., $|S| \geq c^2 \log(p)$.

Further, by Lemma 18, we have with probability $1 - 2e^{-p}$,

$$\sup_{\beta \in \mathcal{C}^t} |\hat{\mu}_4(\beta) - \mu_4(\beta)| \leq 3B_4 \sqrt{\frac{p \log(n)}{n}} = \mathcal{O}\left(\sqrt{\frac{p \log(n)}{n}}\right).$$

Combining the above results, for sufficiently large $p, |S|$, we have with probability at least $1 - 1/p^2 - 2e^{-p}$,

$$\begin{aligned} & \sup_{\beta \in \mathcal{C}^t} \|\hat{\mu}_2(\beta) \zeta_r(\widehat{\Sigma}_S) - \mu_2(\beta) \Sigma\|_2 + \sup_{\beta \in \mathcal{C}^t} \|\hat{\mu}_4(\beta) \widehat{\Sigma}_S \beta \widehat{\Sigma}_S \beta^T - \mu_4(\beta) \Sigma \beta \beta^T \Sigma\|_2 \\ & \leq 3B_2 K c \sqrt{\frac{p}{\min\{p/\log(p), |S|, n/\log(n)\}}} + 4cK^2 B_4 R^2 \sqrt{\frac{\log(p)}{|S|}} + 3B_4 R^2 K^2 \sqrt{\frac{p \log(n)}{n}}, \end{aligned}$$

$$\begin{aligned}
 &\leq 3B_2Kc\sqrt{\frac{p}{\min\{p/\log(p)|S|, n/\log(n)\}}} + 4cK^2B_4R^2\sqrt{\frac{p}{\min\{p/\log(p)|S|, n/\log(n)\}}}, \\
 &\leq CK\max\{B_2, B_4KR^2\}\sqrt{\frac{p}{\min\{p/\log(p)|S|, n/\log(n)\}}}, \\
 &= \mathcal{O}\left(\sqrt{\frac{p}{\min\{|S|p/\log(p), n/\log(n)\}}}\right).
 \end{aligned}$$

Hence, for some constants C_1, C_2 , with probability $1 - C_1/p^2$, we have

$$\|\mathbf{Q}^{\dagger-1} - \mathbf{e}(\hat{\beta}^t)\|_2 \leq C_2\sqrt{\frac{p}{\min\{|S|p/\log(p), n/\log(n)\}}},$$

where the constants depend on $K, B = \max\{B_2, B_4\}$ and the radius R . \blacksquare

Lemma 26 *The bias term can be upper bounded by*

$$\|\mathbb{E}[xx^T\phi^{(2)}(\langle x, \hat{\beta}^t \rangle)] - \mathbf{e}(\hat{\beta}^t)\|_2 \leq d_{H_3}(x, z) + \|\Sigma\|_2 d_{H_1}(x, z) + \|\Sigma\|_2^2 R^2 d_{H_2}(x, z),$$

for both sub-Gaussian and bounded support cases.

Proof For a random variable $z \sim N_p(0, \Sigma)$, by the triangle inequality, we write

$$\begin{aligned}
 &\|\mathbb{E}[xx^T\phi^{(2)}(\langle x, \hat{\beta}^t \rangle)] - \mathbf{e}(\hat{\beta}^t)\|_2 \\
 &\leq \|\mathbb{E}[xx^T\phi^{(2)}(\langle x, \hat{\beta}^t \rangle)] - \mathbb{E}[zz^T\phi^{(2)}(\langle z, \hat{\beta}^t \rangle)]\|_2 + \|\mathbb{E}[zz^T\phi^{(2)}(\langle z, \hat{\beta}^t \rangle)] - \mathbf{e}(\hat{\beta}^t)\|_2
 \end{aligned}$$

For the first term on the right hand side, we have

$$\begin{aligned}
 &\|\mathbb{E}[xx^T\phi^{(2)}(\langle x, \hat{\beta}^t \rangle)] - \mathbb{E}[zz^T\phi^{(2)}(\langle z, \hat{\beta}^t \rangle)]\|_2 \\
 &\leq \sup_{\beta \in \mathcal{C}} \sup_{\|v\|_2=1} \left| \mathbb{E}\left[\langle v, x \rangle^2 \phi^{(2)}(\langle x, \beta \rangle)\right] - \mathbb{E}\left[\langle v, z \rangle^2 \phi^{(2)}(\langle z, \beta \rangle)\right] \right|, \\
 &\leq d_{H_3}(x, z).
 \end{aligned}$$

For the second term, we write

$$\begin{aligned}
 &\|\mathbb{E}[zz^T\phi^{(2)}(\langle z, \hat{\beta}^t \rangle)] - \mathbf{e}(\hat{\beta}^t)\|_2 \\
 &\leq \sup_{\beta \in \mathcal{C}} \|\mathbb{E}[zz^T\phi^{(2)}(\langle z, \beta \rangle)] - \mathbb{E}[\phi^{(2)}(\langle x, \beta \rangle)]\Sigma + \mathbb{E}\left[\phi^{(4)}(\langle x, \beta \rangle)\right]\Sigma\beta\beta^T\Sigma\|_2, \\
 &\leq \sup_{\beta \in \mathcal{C}} \|\mathbb{E}[\phi^{(2)}(\langle z, \beta \rangle)]\Sigma + \mathbb{E}\left[\phi^{(4)}(\langle z, \beta \rangle)\right]\Sigma\beta\beta^T\Sigma \\
 &\quad - \mathbb{E}[\phi^{(2)}(\langle x, \beta \rangle)]\Sigma - \mathbb{E}\left[\phi^{(4)}(\langle x, \beta \rangle)\right]\Sigma\beta\beta^T\Sigma\|_2, \\
 &\leq \sup_{\beta \in \mathcal{C}} \|\mathbb{E}[\phi^{(2)}(\langle z, \beta \rangle)]\Sigma - \mathbb{E}[\phi^{(2)}(\langle x, \beta \rangle)]\Sigma\|_2, \\
 &\quad + \sup_{\beta \in \mathcal{C}} \|\mathbb{E}\left[\phi^{(4)}(\langle z, \beta \rangle)\right]\Sigma\beta\beta^T\Sigma - \mathbb{E}\left[\phi^{(4)}(\langle x, \beta \rangle)\right]\Sigma\beta\beta^T\Sigma\|_2, \\
 &\leq \|\Sigma\|_2 \sup_{\beta \in \mathcal{C}} \|\mathbb{E}[\phi^{(2)}(\langle z, \beta \rangle)] - \mathbb{E}[\phi^{(2)}(\langle x, \beta \rangle)]\| \\
 &\quad + \|\Sigma\|_2^2 R^2 \sup_{\beta \in \mathcal{C}} \|\mathbb{E}[\phi^{(4)}(\langle z, \beta \rangle)] - \mathbb{E}[\phi^{(4)}(\langle x, \beta \rangle)]\|, \\
 &\leq \|\Sigma\|_2 d_{H_1}(x, z) + \|\Sigma\|_2^2 R^2 d_{H_2}(x, z).
 \end{aligned}$$

Hence, we conclude that

$$\|\mathbb{E}[xx^T\phi^{(2)}(\langle x, \hat{\beta}^t \rangle)] - \mathbf{e}(\hat{\beta}^t)\|_2 \leq d_{H_3}(x, z) + \|\Sigma\|_2 d_{H_1}(x, z) + \|\Sigma\|_2^2 R^2 d_{H_2}(x, z).$$

\blacksquare

Lemma 27 *There exists constants c_1, c_2, c_3 depending on the eigenvalues of Σ, B, L and R such that, with probability at least $1 - c_2e^{-c_3p}$*

$$\left\| \frac{1}{n} \sum_{i=1}^n x_i x_i^T \int_0^1 \phi^{(2)}(\langle x_i, \beta_* + \xi(\hat{\beta}^t - \beta_*) \rangle) d\xi - \mathbb{E}\left[xx^T \int_0^1 \phi^{(2)}(\langle x, \beta_* + \xi(\hat{\beta}^t - \beta_*) \rangle) d\xi\right] \right\|_2 \leq \delta,$$

where $\delta = c_1 \sqrt{\frac{p}{n \log(n)}}$ for sub-Gaussian covariates, and $\delta = c_1 \sqrt{\frac{p}{n} \log(n)}$ for covariates with bounded support.

Proof We provide the proof for bounded support case. The proof for sub-Gaussian case can be carried by replacing Lemma 19 with Lemma 23.

By the Fubini's theorem, we have

$$\begin{aligned}
& \left\| \frac{1}{n} \sum_{i=1}^n x_i x_i^T \int_0^1 \phi^{(2)}(\langle x_i, \beta_* + \xi(\beta^t - \beta_*) \rangle) d\xi - \mathbb{E} \left[x x^T \int_0^1 \phi^{(2)}(\langle x, \beta_* + \xi(\beta^t - \beta_*) \rangle) d\xi \right] \right\|_2, \\
&= \left\| \int_0^1 \left\{ \frac{1}{n} \sum_{i=1}^n x_i x_i^T \phi^{(2)}(\langle x_i, \beta_* + \xi(\beta^t - \beta_*) \rangle) - \mathbb{E} \left[x x^T \phi^{(2)}(\langle x, \beta_* + \xi(\beta^t - \beta_*) \rangle) \right] \right\} d\xi \right\|_2, \\
&\leq \int_0^1 \left\| \left\{ \frac{1}{n} \sum_{i=1}^n x_i x_i^T \phi^{(2)}(\langle x_i, \beta_* + \xi(\beta^t - \beta_*) \rangle) - \mathbb{E} \left[x x^T \phi^{(2)}(\langle x, \beta_* + \xi(\beta^t - \beta_*) \rangle) \right] \right\} \right\|_2 d\xi, \\
&\leq \sup_{\beta \in \mathcal{C}} \left\| \frac{1}{n} \sum_{i=1}^n x_i x_i^T \phi^{(2)}(\langle x_i, \beta \rangle) - \mathbb{E} \left[x x^T \phi^{(2)}(\langle x, \beta \rangle) \right] \right\|_2.
\end{aligned}$$

Using the properties of operator norm, the above bound can be written as

$$\begin{aligned}
& \sup_{\beta \in \mathcal{C}} \left\| \frac{1}{n} \sum_{i=1}^n x_i x_i^T \phi^{(2)}(\langle x_i, \beta \rangle) - \mathbb{E} \left[x x^T \phi^{(2)}(\langle x, \beta \rangle) \right] \right\|_2 \\
&= \sup_{\beta \in \mathcal{C}} \sup_{v \in S^{p-1}} \left\| \frac{1}{n} \sum_{i=1}^n \phi^{(2)}(\langle x_i, \beta \rangle) \langle x_i, v \rangle^2 - \mathbb{E} \left[\phi^{(2)}(\langle x, \beta \rangle) \langle x, v \rangle^2 \right] \right\|,
\end{aligned}$$

where S^{p-1} denotes the p -dimensional unit sphere.

For $\Delta = 0.25$, let T_Δ be an Δ -net over S^{p-1} . Using Lemma 32, we obtain

$$\begin{aligned}
& \mathbb{P} \left(\sup_{\beta \in \mathcal{C}} \sup_{v \in S^{p-1}} \left\| \frac{1}{n} \sum_{i=1}^n \phi^{(2)}(\langle x_i, \beta \rangle) \langle x_i, v \rangle^2 - \mathbb{E} \left[\phi^{(2)}(\langle x, \beta \rangle) \langle x, v \rangle^2 \right] \right\| > \epsilon \right), \\
&\leq \mathbb{P} \left(\sup_{\beta \in \mathcal{C}} \sup_{v \in T_\Delta} \left\| \frac{1}{n} \sum_{i=1}^n \phi^{(2)}(\langle x_i, \beta \rangle) \langle x_i, v \rangle^2 - \mathbb{E} \left[\phi^{(2)}(\langle x, \beta \rangle) \langle x, v \rangle^2 \right] \right\| > \epsilon/2 \right), \\
&\leq |T_\Delta| \mathbb{P} \left(\sup_{\beta \in \mathcal{C}} \left\| \frac{1}{n} \sum_{i=1}^n \phi^{(2)}(\langle x_i, \beta \rangle) \langle x_i, v \rangle^2 - \mathbb{E} \left[\phi^{(2)}(\langle x, \beta \rangle) \langle x, v \rangle^2 \right] \right\| > \epsilon/2 \right), \\
&= 9^p \mathbb{P} \left(\sup_{\beta \in \mathcal{C}} \left\| \frac{1}{n} \sum_{i=1}^n \phi^{(2)}(\langle x_i, \beta \rangle) \langle x_i, v \rangle^2 - \mathbb{E} \left[\phi^{(2)}(\langle x, \beta \rangle) \langle x, v \rangle^2 \right] \right\| > \epsilon/2 \right).
\end{aligned}$$

By applying Lemma 19 to the last line above, we obtain

$$\mathbb{P} \left(\sup_{\beta \in \mathcal{C}} \left\| \frac{1}{n} \sum_{i=1}^n \phi^{(2)}(\langle x_i, \beta \rangle) \langle x_i, v \rangle^2 - \mathbb{E} \left[\phi^{(2)}(\langle x, \beta \rangle) \langle x, v \rangle^2 \right] \right\| > 4B_2 K \sqrt{\frac{p}{n} \log(n)} \right) \leq 2e^{-3.2p}.$$

Notice that $3.2 - \log(9) > 1$. Therefore, by choosing n large enough, on the set \mathcal{E} , we obtain that with probability at least $1 - 2e^{-p}$

$$\sup_{\beta \in \mathcal{C}} \left\| \frac{1}{n} \sum_{i=1}^n x_i x_i^T \phi^{(2)}(\langle x_i, \beta \rangle) - \mathbb{E} \left[x x^T \phi^{(2)}(\langle x, \beta \rangle) \right] \right\|_2 \leq 8B_2 K \sqrt{\frac{p}{n} \log(n)}.$$

■

Lemma 28 *There exists a constant C depending on K and L such that,*

$$\left\| \mathbb{E} \left[x x^T \phi^{(2)}(\langle x, \hat{\beta}^t \rangle) \right] - \mathbb{E} \left[x x^T \int_0^1 \phi^{(2)}(\langle x, \beta_* + \xi(\hat{\beta}^t - \beta_*) \rangle) d\xi \right] \right\|_2 \leq \tilde{C} \|\hat{\beta}^t - \beta_*\|_2,$$

where $\tilde{C} = C$ for the bounded support case and $\tilde{C} = Cp^{1.5}$ for the sub-Gaussian case.

Proof By the Fubini's theorem, we write

$$\begin{aligned}
& \left\| \mathbb{E} \left[x x^T \phi^{(2)}(\langle x, \hat{\beta}^t \rangle) \right] - \mathbb{E} \left[x x^T \int_0^1 \phi^{(2)}(\langle x, \beta_* + \xi(\hat{\beta}^t - \beta_*) \rangle) d\xi \right] \right\|_2, \\
&= \left\| \int_0^1 \mathbb{E} \left[x x^T \left\{ \phi^{(2)}(\langle x, \hat{\beta}^t \rangle) - \phi^{(2)}(\langle x, \beta_* + \xi(\hat{\beta}^t - \beta_*) \rangle) \right\} \right] d\xi \right\|_2.
\end{aligned}$$

Moving the integration out, right hand side of the above equation is smaller than

$$\begin{aligned}
& \int_0^1 \left\| \mathbb{E} \left[x x^T \left\{ \phi^{(2)}(\langle x, \hat{\beta}^t \rangle) - \phi^{(2)}(\langle x, \beta_* + \xi(\hat{\beta}^t - \beta_*) \rangle) \right\} \right] \right\|_2 d\xi, \\
&\leq \int_0^1 \left\| \mathbb{E} \left[x x^T L \langle x, (1 - \xi)(\hat{\beta}^t - \beta_*) \rangle \right] \right\|_2 d\xi, \\
&\leq \mathbb{E} \left[\|x\|_2^2 \|\hat{\beta}^t - \beta_*\|_2 \right] L \int_0^1 (1 - \xi) d\xi, \\
&= \frac{L \mathbb{E} \left[\|x\|_2^3 \right]}{2} \|\hat{\beta}^t - \beta_*\|_2.
\end{aligned}$$

We observe that, when the covariates are supported in the ball of radius \sqrt{K} , we have $\frac{\mathbb{E} \left[\|x\|_2^3 \right]}{\mathbb{E} \left[\|x\|_2^2 \right]} \leq K^{3/2}$. When they are sub-Gaussian random variables with norm K , we have $\frac{\mathbb{E} \left[\|x\|_2^3 \right]}{\mathbb{E} \left[\|x\|_2^2 \right]} \leq K^3 6^{1.5} p^{1.5}$. ■

By combining the above results, for bounded covariates we obtain

$$\begin{aligned}
& \left\| \mathbf{Q}^t \right\|^{-1} - \int_0^1 \frac{\nabla^2 \phi(\beta_* + \xi(\hat{\beta}^t - \beta_*)) d\xi}{\mathbb{E} \left[\phi^{(2)}(\langle x, \hat{\beta}^t \rangle) \right]} \\
&\leq \mathfrak{Q}(x, z) + c_1 \sqrt{\frac{p}{\min\{|S|p/\log(p), n/\log(n)\}}} + c_2 \|\hat{\beta}^t - \beta_*\|_2,
\end{aligned}$$

and for sub-Gaussian covariates, we obtain

$$\begin{aligned} & \left\| [\mathbf{Q}^j]^{-1} - \int_0^1 \nabla_{\beta}^2 \ell(\beta_* + \xi(\hat{\beta}^t - \beta_*)) d\xi \right\|_2 \\ & \leq \mathfrak{D}(x, z) + c_1 \sqrt{\frac{p}{\min\{|S|, n^{0.2}/\log(n)\}}} + c_2 p^{1.5} \|\hat{\beta}^t - \beta_*\|_2, \end{aligned}$$

where

$$\mathfrak{D}(x, z) = d_{H_2}(x, z) + \|\Sigma\|_2 d_{H_4}(x, z) + \|\Sigma\|_2^2 R^2 d_{H_2}(x, z).$$

In the following, we will derive an upper bound for $\|\mathbf{Q}^t\|_2$, which is equivalent to proving the positive definiteness of $[\mathbf{Q}^j]^{-1}$ and finding a lower bound for $\|[\mathbf{Q}^j]^{-1}\|_2$. The sub-Gaussian case is more restrictive than the bounded support case. Therefore we derive the bound for the sub-Gaussian case. We have

$$\begin{aligned} \lambda_{\min}([\mathbf{Q}^j]^{-1}) &= \inf_{\|u\|_2=1} \left\{ \hat{\mu}_2(\hat{\beta}^t) \langle u, \widehat{\Sigma}_S u \rangle + \hat{\mu}_4(\hat{\beta}^t) \langle u, \widehat{\Sigma}_S \hat{\beta}^t \rangle^2 \right\}, \\ &\geq \inf_{\|u\|_2=1} \left\{ \hat{\mu}_2(\hat{\beta}^t) \langle u, \Sigma u \rangle + \hat{\mu}_4(\hat{\beta}^t) \langle u, \Sigma \hat{\beta}^t \rangle^2 \right\} \\ &\quad - B_2 \|\widehat{\Sigma}_S - \Sigma\|_2 - B_4 R^2 \|\widehat{\Sigma}_S - \Sigma\|_2 \|\widehat{\Sigma}_S + \Sigma\|_2. \end{aligned}$$

On the event \mathcal{E} , the first term on the right hand side is lower bounded by κ^{-1} . For the other terms, we use Lemma 12 and write

$$\begin{aligned} \lambda_{\min}([\mathbf{Q}^j]^{-1}) &\leq 2\kappa^{-1} - \|\widehat{\Sigma}_S - \Sigma\|_2 \left\{ B_2 + B_4 R^2 \|\widehat{\Sigma}_S - \Sigma\|_2 + 2B_4 R^2 \|\Sigma\|_2 \right\}, \\ &\leq 2\kappa^{-1} - C \sqrt{\frac{p}{|S|}} \left\{ B_2 + B_4 R^2 C \sqrt{\frac{p}{|S|}} + 2B_4 R^2 \|\Sigma\|_2 \right\} \end{aligned}$$

with probability $1 - 2e^{-cp}$. When $|S| > 4pC^2 \max\{1, 2C(B_2 + 3B_4 R^2 \lambda_{\max}(\Sigma))\kappa\}^2$, with probability $1 - 2e^{-cp}$, we obtain

$$\lambda_{\min}([\mathbf{Q}^j]^{-1}) \geq \kappa^{-1}.$$

This proves that, with high probability, on the event \mathcal{E} , $[\mathbf{Q}^j]^{-1}$ is positive definite and consequently we obtain

$$\|\mathbf{Q}^j\|_2 \leq \kappa.$$

Finally, we take into account the conditioning on the event \mathcal{E} . Since we worked on the event \mathcal{E} , the probability of a desired outcome is at least $\mathbb{P}(\mathcal{E}) - \delta$, where δ is either c/p^2 or ce^{-p} depending on the distribution of the covariates. Hence, conditioned on the event \mathcal{E} , the probability becomes $1 - \delta/\mathbb{P}(\mathcal{E})$, which completes the proof.

C.1 Proof of Corollaries 5 and 9

In the following, we provide the proof for Corollary 5. The proof for Corollary 9 follows from the exact same steps.

The statement of Theorem 4 holds on the probability space with a probability lower bounded by $\mathbb{P}(\mathcal{E}) - c/p^2$ for some constant c (See previous section). Let \mathcal{Q} denote this set, on which the statement of the theorem holds without the conditioning on the event \mathcal{E} . Note that $\mathcal{Q} \subset \mathcal{E}$ and we also have

$$\mathbb{P}(\mathcal{E}) \geq \mathbb{P}(\mathcal{Q}) \geq \mathbb{P}(\mathcal{E}) - c/p^2. \quad (34)$$

This suggests that the difference between \mathcal{Q} and \mathcal{E} is small. By taking expectations on both sides over the set \mathcal{Q} , we obtain,

$$\begin{aligned} \mathbb{E} \left[\|\hat{\beta}^{t+1} - \beta_*\|_2; \mathcal{Q} \right] &\leq \kappa \left\{ \mathfrak{D}(x, z) + c_1 \sqrt{\frac{p}{\min\{p/\log(p)|S|, n/\log(n)\}}} \right\} \mathbb{E} \left[\|\hat{\beta}^t - \beta_*\|_2 \right] \\ &\quad + \kappa c_2 \mathbb{E} \left[\|\hat{\beta}^t - \beta_*\|_2^2 \right] \end{aligned}$$

where we used

$$\mathbb{E} \left[\|\hat{\beta}^t - \beta_*\|_2; \mathcal{Q} \right] \leq \mathbb{E} \left[\|\hat{\beta}^t - \beta_*\|_2^l \right], \quad l = 1, 2.$$

Similarly for the iterate $\hat{\beta}^{t+1}$, we write

$$\begin{aligned} \mathbb{E} \left[\|\hat{\beta}^{t+1} - \beta_*\|_2 \right] &= \mathbb{E} \left[\|\hat{\beta}^{t+1} - \beta_*\|_2; \mathcal{Q} \right] + \mathbb{E} \left[\|\hat{\beta}^{t+1} - \beta_*\|_2; \mathcal{Q}^c \right], \\ &\leq \mathbb{E} \left[\|\hat{\beta}^{t+1} - \beta_*\|_2; \mathcal{Q} \right] + 2R\mathbb{P}(\mathcal{Q}^c), \\ &\leq \mathbb{E} \left[\|\hat{\beta}^{t+1} - \beta_*\|_2; \mathcal{Q} \right] + 2R \left(\mathbb{P}(\mathcal{E}^c) + \frac{c}{p^2} \right), \\ &\leq \mathbb{E} \left[\|\hat{\beta}^{t+1} - \beta_*\|_2; \mathcal{Q} \right] + \frac{\epsilon}{10}, \\ &\leq \mathbb{E} \left[\|\hat{\beta}^{t+1} - \beta_*\|_2; \mathcal{Q} \right] + \frac{\mathbb{E} \left[\|\hat{\beta}^t - \beta_*\|_2 \right]}{10}. \end{aligned}$$

Combining these two inequalities, we obtain

$$\begin{aligned} \mathbb{E} \left[\|\hat{\beta}^{t+1} - \beta_*\|_2 \right] &\leq \left\{ 0.1 + \kappa \mathfrak{D}(x, z) + c_1 \kappa \sqrt{\frac{p}{\min\{p/\log(p)|S|, n/\log(n)\}}} \right\} \mathbb{E} \left[\|\hat{\beta}^t - \beta_*\|_2 \right] \\ &\quad + c_2 \kappa \mathbb{E} \left[\|\hat{\beta}^t - \beta_*\|_2^2 \right]. \end{aligned}$$

Hence the proof follows.

C.2 Proof of Theorem 6

The iterates generated by the Newton-Stein method satisfy the following inequality;

$$\|\beta^{t+1} - \beta_*\|_2 \leq \left(\tau_1 + \tau_2 \|\beta^t - \beta_*\|_2 \right) \|\beta^t - \beta_*\|_2,$$

on the event \mathcal{Q} where \mathcal{Q} is defined in the previous section. We have observed that $\mathbb{P}(\mathcal{Q}) \geq \mathbb{P}(\mathcal{E}) - c/p^2$ in Equation 34. Since the coefficients τ_1 and τ_2 are obtained by uniform bounds on the feasible set, the above inequality holds for every t on \mathcal{Q} . On the event we consider, $\mathcal{Q} \cap \{\theta < (1 - \tau_1)/\tau_2\}$, the starting point satisfies the following

$$\tau_1 + \tau_2 \|\beta^0 - \beta_*\|_2 < 1, \quad (35)$$

which implies that the sequence of iterates converges. Let $\xi \in (c, \theta)$ and t_ξ be the last iteration that $\|\beta^t - \beta_*\|_2 > \xi$. Then, for $t > t_\xi$

$$\begin{aligned} \|\beta^{t+1} - \beta_*\|_2 &\leq \left(\tau_1 + \tau_2 \|\beta^t - \beta_*\|_2 \right) \|\beta^t - \beta_*\|_2, \\ &\leq (\tau_1 + \tau_2 \xi) \|\beta^t - \beta_*\|_2. \end{aligned}$$

This convergence behavior describes a linear rate and requires at most

$$\frac{\log(\epsilon/\xi)}{\log(\tau_1 + \tau_2 \xi)}$$

iterations to reach a tolerance of ϵ . For $t \leq t_\xi$, we have

$$\begin{aligned} \|\beta^{t+1} - \beta_*\|_2 &\leq \left(\tau_1 + \tau_2 \|\beta^t - \beta_*\|_2 \right) \|\beta^t - \beta_*\|_2, \\ &\leq (\tau_1/\xi + \tau_2) \|\beta^t - \beta_*\|_2^2. \end{aligned}$$

This describes a quadratic rate and the number of iterations to reach a tolerance of ξ can be upper bounded by

$$\log_2 \left(\frac{\log(\xi(\tau_1/\xi + \tau_2))}{\log(\tau_1/\xi + \tau_2) \|\beta^0 - \beta_*\|_2} \right) \leq \log_2 \left(\frac{\log(\tau_1 + \tau_2 \xi)}{\log(\tau_1/\xi + \tau_2)(1 - \tau_1)/\tau_2} \right).$$

Therefore, the overall number of iterations to reach a tolerance of ϵ is upper bounded by

$$\log_2 \left(\frac{\log(\tau_1 + \tau_2 \xi)}{\log((\tau_1/\xi + \tau_2)(1 - \tau_1)/\tau_2)} \right) + \frac{\log(\epsilon/\xi)}{\log(\tau_1 + \tau_2 \xi)}$$

which is a function of ξ . Therefore, we take the minimum over the feasible set and conclude that on $\mathcal{E} \cap \{\theta < (1 - \tau_1)/\tau_2\}$, the number of iterations to reach a tolerance of ϵ is upper bounded by $\inf_{\xi} \mathcal{J}(\xi)$ with a bad event probability of c/p^2 . By conditioning on the event $\mathcal{E} \cap \{\theta < (1 - \tau_1)/\tau_2\}$, we conclude that with probability at least $1 - c'/p^2$, the statement of the theorem holds for $c' = c/\mathbb{P}(\mathcal{E} \cap \{\theta < (1 - \tau_1)/\tau_2\})$.

Appendix D. Proof of Theorem 7

We have the following projected updates

$$\beta^{t+1} = \mathcal{P}_{\mathcal{C}} \left(\beta^t - \gamma_t \mathbf{Q}^t \nabla \ell(\beta^t); \mathbf{Q}^t \right) = \beta^t - \gamma_t D_{\eta_t}(\beta^t),$$

where we define

$$D_{\gamma}(\beta^t) = \frac{1}{\gamma} \left(\beta^t - \mathcal{P}_{\mathcal{C}}(\beta^t - \gamma \mathbf{Q}^t \nabla \ell(\beta^t); \mathbf{Q}^t) \right).$$

For simplicity, we only consider the projection onto a convex set, i.e.,

$$\begin{aligned} \mathcal{P}_{\mathcal{C}}^{\ell}(\beta^t) &= \mathcal{P}_{\mathcal{C}}(\beta^t, \mathbf{Q}^t) = \underset{w \in \mathcal{C}}{\operatorname{argmin}} \frac{1}{2} \|w - \beta^t\|_{\mathbf{Q}^{t-1}}^2, \\ &= \underset{w \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2} \|w - \beta^t\|_{\mathbf{Q}^{t-1}}^2 + \mathbb{I}_{\mathcal{C}}(w), \end{aligned} \quad (36)$$

where $\mathbb{I}_{\mathcal{C}}(w)$ is the indicator function for the convex set \mathcal{C} , i.e.

$$\mathbb{I}_{\mathcal{C}}(w) = \begin{cases} 0 & \text{if } w \in \mathcal{C}, \\ \infty & \text{otherwise.} \end{cases}$$

We note that other projection methods (such as proximal mappings) are also applicable to our update rule.

Defining the decrement $\lambda^t = \langle \nabla \ell(\beta^t), D_{\gamma}(\beta^t) \rangle$, we consider the following form of backtracking line search with update parameters $a \in (0, 0.5)$ and $b \in (0, 1)$:

$$\gamma = \bar{\gamma}; \quad \text{while: } \ell(\beta^t - \gamma D_{\gamma}(\beta^t)) > \ell(\beta^t) - a\gamma\lambda^t, \quad \gamma \leftarrow \gamma b.$$

Depending on the projection choice, there are various other search methods that can be applied. Before we move on to the convergence analysis, we first establish some properties of the modified gradient D_{γ} .

For a given point $w \in \mathcal{C}$, the sub-differential of the indicator function is the normal cone. This together with Equation 36 implies that

$$\beta^t - \gamma \mathbf{Q}^t \nabla \ell(\beta^t) - \mathcal{P}_{\mathcal{C}}^{\ell}(\beta^t - \gamma \mathbf{Q}^t \nabla \ell(\beta^t)) \in \mathbf{Q}^t \partial \mathbb{I}_{\mathcal{C}}(\mathcal{P}_{\mathcal{C}}^{\ell}(\beta^t - \gamma \mathbf{Q}^t \nabla \ell(\beta^t))),$$

which in turn implies

$$\gamma [\mathbf{Q}^t]^{-1} \left\{ D_{\gamma}(\beta^t) - \mathbf{Q}^t \nabla \ell(\beta^t), \right\} \in \partial \mathbb{I}_{\mathcal{C}}(\mathcal{P}_{\mathcal{C}}^{\ell}(\beta^t - \gamma \mathbf{Q}^t \nabla \ell(\beta^t))),$$

and correspondingly for any $\beta \in \mathcal{C}$

$$\langle [\mathbf{Q}^t]^{-1} D_{\gamma}(\beta^t) - \nabla \ell(\beta^t), \mathcal{P}_{\mathcal{C}}^{\ell}(\beta^t - \gamma \mathbf{Q}^t \nabla \ell(\beta^t)) - \beta \rangle \geq 0.$$

For $\beta = \beta^t \in \mathcal{C}$, this yields

$$\kappa^{-1} \|D_{\gamma}(\beta^t)\|_2^2 \leq \langle D_{\gamma}(\beta^t), [\mathbf{Q}^t]^{-1} D_{\gamma}(\beta^t) \rangle \leq \langle \nabla \ell(\beta^t), D_{\eta_t}(\beta^t) \rangle, \quad (37)$$

with probability at least $P(\mathcal{E}) - c/p^2$. Also note that the Hessian of the GLM problem can be upper bounded by

$$\left\| \frac{1}{n} \sum_{i=1}^n x_i x_i^T \phi^{(2)}(\langle x_i, \hat{\beta}^t \rangle) \right\|_2 \leq B_2 \left\| \frac{1}{n} \sum_{i=1}^n x_i x_i^T \right\|_2 \leq B_2 K.$$

Now we move to the convergence analysis. For a step size γ , by the convexity of the negative log-likelihood, we can write almost surely

$$\begin{aligned} \ell(\hat{\beta}^t - \gamma D_\gamma(\hat{\beta}^t)) &\leq \ell(\hat{\beta}^t) - \gamma \langle \nabla \ell(\hat{\beta}^t), D_\gamma(\hat{\beta}^t) \rangle + \frac{\gamma^2 B_2 K}{2} \|D_\gamma(\hat{\beta}^t)\|_2^2, \\ &\leq \ell(\hat{\beta}^t) - \gamma \langle \nabla \ell(\hat{\beta}^t), D_\gamma(\hat{\beta}^t) \rangle \left\{ 1 - \frac{\gamma}{2} B_2 K \kappa \right\} \end{aligned}$$

and notice that the exit condition for the backtracking line search algorithm is satisfied when $\gamma \leq (\kappa B_2 K)^{-1}$. Hence, the line search returns a step size satisfying

$$\gamma_t \geq \min\{\bar{\gamma}, b/(\kappa B_2 K)\}.$$

Using the line search condition, we have

$$\ell(\hat{\beta}^t - \gamma_t D_{\gamma_t}(\hat{\beta}^t)) - \ell(\hat{\beta}^t) \leq -a\gamma_t \lambda^t,$$

with probability at least $\mathbb{P}(\mathcal{E}) - c/p^2$ which implies that the sequence $\{\ell(\hat{\beta}^t)\}_t$ is decreasing. We note that this event is independent of the iteration number due to uniform positive definite condition given in \mathcal{E} . Since ℓ is continuous and \mathcal{C} is closed, ℓ is a closed function. Hence, the sequence $\{\ell(\hat{\beta}^t)\}_t$ must converge to a limit. This implies that $a\gamma_t \lambda^t \rightarrow 0$. But we have $a > 0$ and $\gamma_t > \min\{\bar{\gamma}, b/(\kappa B_2 K)\} > 0$. Therefore, we conclude that $\lambda^t \rightarrow 0$. Using the inequality provided in Equation 37, we conclude that $\|D_\gamma(\hat{\beta}^t)\|_2$ converges to 0 which implies that the algorithm converges with probability at least $1 - \frac{c}{\mathbb{P}(\mathcal{E})} p^{-2}$, where in the last step we conditioned on \mathcal{E} .

Appendix E. Local Step Size Selection

This section provides a heuristic calculation for choosing a local step size when eigenvalue thresholding is applied to the Newton-Stein method. We carry our analysis from Equation 32. The optimal local step size would be

$$\gamma_* = \operatorname{argmin}_\gamma \left\| I - \gamma \mathbf{Q}^t \int_0^1 \nabla_\beta^2 \ell(\beta_* + \xi(\hat{\beta}^t - \beta_*)) d\xi \right\|_2.$$

Defining the following matrix,

$$\nabla_\beta^3 \tilde{\ell}(\hat{\beta}^t) = \int_0^1 \nabla_\beta^2 \ell(\beta_* + \xi(\hat{\beta}^t - \beta_*)) d\xi,$$

and we write the governing term as

$$\left\| I - \gamma \mathbf{Q}^t \nabla_\beta^3 \tilde{\ell}(\hat{\beta}^t) \right\|_2.$$

The above function is piecewise linear in γ and it can be minimized by setting

$$\gamma_* = \frac{2}{\lambda_1 \left(\mathbf{Q}^t \nabla_\beta^2 \tilde{\ell}(\hat{\beta}^t) \right) + \lambda_p \left(\mathbf{Q}^t \nabla_\beta^2 \tilde{\ell}(\hat{\beta}^t) \right)}.$$

Since we don't have access to the optimal value β_* , we cannot determine the exact value of $\nabla_\beta^2 \tilde{\ell}(\hat{\beta}^t)$. Hence, we will assume that $\nabla_\beta^2 \tilde{\ell}(\hat{\beta}^t)$ and the current estimate are close.

In the regime $n \gg p$, and by our construction of the scaling matrix \mathbf{Q}^t , we have

$$\mathbf{Q}^t \approx \left[\mathbb{E}[xx^T \phi^{(2)}(\langle x, \hat{\beta}^t \rangle)] \right]^{-1} \quad \text{and} \quad \nabla_\beta^2 \ell(\hat{\beta}^t) \approx \mathbb{E}[xx^T \phi^{(2)}(\langle x, \hat{\beta}^t \rangle)].$$

The crucial observation is that the eigenvalue thresholding suggested in Erdogdu and Montanari, 2015 estimates the smallest eigenvalue with $(r+1)$ -th eigenvalue (say $\hat{\sigma}^2$) which overestimates true value (say σ^2) in general. Even though, the largest eigenvalue of $\mathbf{Q}^t \nabla_\beta^2 \tilde{\ell}(\hat{\beta}^t)$ will be close to 1, the smallest value will be $\sigma^2/\hat{\sigma}^2$. This will make the optimal step size larger than 1. Hence, we suggest

$$\gamma = \frac{2}{1 + \sigma^2/\hat{\sigma}^2},$$

if σ^2 were known. We also have, by the Weyl's inequality,

$$|\sigma^2 - \sigma^2| \leq \left\| \hat{\Sigma} - \Sigma \right\|_2 \leq C \sqrt{\frac{p}{|S|}},$$

with high probability. Whenever r is less than $p/2$, we suggest to use

$$\gamma = \frac{2}{1 + \frac{\sigma^2 - C(\sqrt{p/|S|})}{\hat{\sigma}^2}},$$

if σ^2 is unknown.

Appendix F. Useful Lemmas

Lemma 29 Let Γ denote the Gamma function. Then, for $r \in (0, 1)$, we have

$$z^{1-r} < \frac{\Gamma(z+1)}{\Gamma(z+r)} < (1+z)^{1-r}.$$

Lemma 30 Let Z be a random variable with a density function f and cumulative distribution function F . If $F^C = 1 - F$, then,

$$\mathbb{E}[|Z|_{\{|Z|>t\}}] \leq t \mathbb{P}(|Z| > t) + \int_t^\infty \mathbb{P}(|Z| > z) dz.$$

Proof We write,

$$\mathbb{E}[|Z|_{\{|Z|>t\}}] = \int_{-\infty}^\infty z f(z) dz + \int_{-\infty}^{-t} z f(z) dz.$$

Using integration by parts, we obtain

$$\begin{aligned} \int z f(z) dz &= -z F^C(z) + \int F^C(z) dz, \\ &= z F(z) - \int F(z) dz. \end{aligned}$$

Since $\lim_{z \rightarrow \infty} z F^C(z) = \lim_{z \rightarrow -\infty} z F(z) = 0$, we have

$$\begin{aligned} \int_t^{\infty} z f(z) dz &= t F^C(t) + \int_t^{\infty} F^C(z) dz, \\ \int_{-\infty}^{-t} z f(z) dz &= -t F(-t) - \int_{-\infty}^{-t} F(z) dz, \\ &= -t F(-t) - \int_t^{\infty} F(-z) dz. \end{aligned}$$

Hence, we obtain the following bound,

$$\begin{aligned} |\mathbb{E}[Z \mathbb{1}_{\{|Z|>t\}}]| &= \left| t F^C(t) + \int_t^{\infty} F^C(z) dz - t F(-t) - \int_t^{\infty} F(-z) dz \right|, \\ &\leq t (F^C(t) + F(-t)) + \left(\int_t^{\infty} F^C(z) + F(-z) dz \right), \\ &\leq t \mathbb{P}(|Z| > t) + \int_t^{\infty} \mathbb{P}(|Z| > z) dz. \end{aligned}$$

■

Lemma 31 For positive constants c_1, c_2 , we have

$$\int_{c_1}^{\infty} e^{-c_2 t^{2/3}} dt \leq \left\{ \frac{3c_1^{1/3}}{2c_2} + \frac{3}{4c_2^2 c_1^{1/3}} \right\} e^{-c_2 c_1^{2/3}}$$

Proof By the change of variables $t^{2/3} = x^2$, we get

$$\int_{c_1}^{\infty} e^{-c_2 t^{2/3}} dt = 3 \int_{c_1^{1/3}}^{\infty} x^2 e^{-c_2 x^2} dx.$$

Next, we notice that

$$d e^{-c_2 x^2} = -2c_2 x e^{-c_2 x^2} dx.$$

Hence, using the integration by parts, we have

$$\int_{c_1}^{\infty} e^{-c_2 t^{2/3}} dt = \frac{3}{2c_2} \left\{ c_1^{1/3} e^{-c_2 c_1^{2/3}} + \int_{c_1^{1/3}}^{\infty} e^{-c_2 x^2} dx \right\}.$$

We will find an upper bound on the second term. Using the change of variables, $x = y + c_1^{1/3}$, we obtain

$$\begin{aligned} \int_{c_1^{1/3}}^{\infty} e^{-c_2 x^2} dx &= \int_0^{\infty} e^{-c_2 (y + c_1^{1/3})^2} dy, \\ &\leq e^{-c_2 c_1^{2/3}} \int_0^{\infty} e^{-2c_2 y c_1^{1/3}} dy, \\ &= \frac{e^{-c_2 c_1^{2/3}}}{2c_2 c_1^{1/3}}. \end{aligned}$$

Combining the above results, we complete the proof. ■

Lemma 32 (Vershynin, 2010) Let X be a symmetric $p \times p$ matrix, and let T_ϵ be an ϵ -net over S^{p-1} . Then,

$$\|X\|_2 \leq \frac{1}{1 - 2\epsilon} \sup_{v \in T_\epsilon} |(Xv, v)|.$$

Lemma 33 Let $B_p(R) \subset \mathbb{R}^p$ be the ball of radius R centered at the origin and T_ϵ be an ϵ -net over $B_p(R)$. Then,

$$|T_\epsilon| \leq \left(\frac{R\sqrt{p}}{\epsilon} \right)^p.$$

Proof A similar proof appears in (Van der Vaart, 2000). The set $B_p(R)$ can be contained in a p -dimensional cube of size $2R$. Consider a grid over this cube with mesh width $2\epsilon/\sqrt{p}$. Then $B_p(R)$ can be covered with at most $(2R/(2\epsilon/\sqrt{p}))^p$ many cubes of edge length $2\epsilon/\sqrt{p}$. If one takes the projection of the centers of such cubes onto $B_p(R)$ and considers the circumscribed balls of radius ϵ , we may conclude that $B_p(R)$ can be covered with at most

$$\left(\frac{2R}{2\epsilon/\sqrt{p}} \right)^p$$

many balls of radius ϵ . ■

Lemma 34 For $a, b > 0$, and ϵ satisfying

$$\epsilon = \left\{ \frac{a}{2} \log \left(\frac{2b^2}{a} \right) \right\}^{1/2} \quad \text{and} \quad \frac{2}{a} b^2 > \epsilon,$$

we have $\epsilon^2 \geq a \log(b/\epsilon)$. Moreover, the gap in the inequality can be written as

$$\epsilon^2 - a \log(b/\epsilon) = \frac{a}{2} \log \log \left(\frac{2b^2}{a} \right).$$

Proof Since $a, b > 0$ and $x \rightarrow e^x$ is a monotone increasing function, the above inequality condition is equivalent to

$$\frac{2e^2}{a} e^{\frac{2x^2}{a}} \geq \frac{2b^2}{a}.$$

Now, we use the function $f(w) = we^w$ for $w > 0$ (in fact this function is well-known by the name Lambert W function). f is continuous and invertible on $[0, \infty)$. Note that f^{-1} is also a continuous and increasing function for $w > 0$. Therefore, we have

$$\epsilon^2 \geq \frac{a}{2} f^{-1} \left(\frac{2b^2}{a} \right)$$

Observe that the smallest possible value for ϵ would be simply the square root of $a f^{-1}(2b^2/a)/2$. For simplicity, we will obtain a more interpretable expression for ϵ . By the definition of f^{-1} , we have

$$\log(f^{-1}(y)) + f^{-1}(y) = \log(y).$$

Since the condition on a and b enforces $f^{-1}(y)$ to be larger than 1, we obtain the simple inequality that

$$f^{-1}(y) \leq \log(y).$$

Using the above inequality, if ϵ satisfies

$$\epsilon^2 = \frac{a}{2} \log \left(\frac{2b^2}{a} \right),$$

we obtain the desired inequality. ■

References

- Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2): 251–276, 1998.
- Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., NY, USA, 1995. ISBN 0198538642.
- Jock A Blackard and Denis J Dean. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and electronics in agriculture*, pages 131–151, 1999.
- Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.
- Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.
- Charles G Broyden. The convergence of a class of double-rank minimization algorithms 2. the new algorithm. *IMA Journal of Applied Mathematics*, 6(3):222–231, 1970.
- Richard H Byrd, Gillian M Chin, Will Neveitt, and Jorge Nocedal. On the use of stochastic hessian information in optimization methods for machine learning. *SIAM Journal on Optimization*, 21(3):977–995, 2011.
- Richard H Byrd, SL Hansen, Jorge Nocedal, and Yoram Singer. A stochastic quasi-newton method for large-scale optimization. *SIAM Journal on Optimization*, 26(2):1008–1031, 2016.
- Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.
- Louis HY Chen, Larry Goldstein, and Qi-Man Shao. *Normal approximation by Stein's method*. Springer Science, 2010.
- Paramveer Dhillion, Yichao Lu, Dean P Foster, and Lyle Ungar. New subsampling algorithms for fast least squares regression. In *Advances in Neural Information Processing Systems 26*, pages 360–368. Curran Associates, Inc., 2013.
- David L Donoho, Matan Gavish, and Iain M Johnstone. Optimal shrinkage of eigenvalues in the spiked covariance model. *arXiv preprint arXiv:1311.0851*, 2013.
- John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul): 2121–2159, 2011.
- Murat A Erdogdu. Newton-Stein Method: A Second Order Method for GLMs via Stein's Lemma. In *Advances in Neural Information Processing Systems 28*, pages 1216–1224. Curran Associates, Inc., 2015.
- Murat A Erdogdu and Andrea Montanari. Convergence rates of sub-sampled newton methods. In *Advances in Neural Information Processing Systems 28*, pages 3034–3042. Curran Associates, Inc., 2015.
- Murat A Erdogdu, Mohsen Bayati, and Lee H Dicker. Scalable approximations for generalized linear problems. *arXiv preprint arXiv:1611.06686*, 2016a.
- Murat A Erdogdu, Lee H Dicker, and Mohsen Bayati. Scaled least squares estimator for glm's in large-scale problems. In *Advances In Neural Information Processing Systems*, pages 3324–3332, 2016b.
- Roger Fletcher. A new approach to variable metric algorithms. *The computer journal*, 13(3):317–322, 1970.
- Michael P Friedlander and Mark Schmidt. Hybrid deterministic-stochastic methods for data fitting. *SIAM Journal on Scientific Computing*, 34(3):A1380–A1405, 2012.
- Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

- Alison I Gibbs and Francis Edward Su. On choosing and bounding probability metrics. *International statistical review*, 70(3):419–435, 2002.
- Donald Goldfarb. A family of variable-metric methods derived by variational means. *Mathematics of computation*, 24(109):23–26, 1970.
- Larry Goldstein and Gesine Reinert. Stein’s method and the zero bias transformation with application to simple random sampling. *The Annals of Applied Probability*, 7(4):935–952, 1997.
- Franz Graf, Hans-Peter Kriegel, Matthias Schubert, Sebastian Polsterl, and Alexander Cavallaro. 2d image registration in ct images using radial image descriptors. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 607–614. Springer, 2011.
- Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
- Ritesh Kolte, Murat A Erdogdu, and Ayfer Ozgur. Accelerating svrg via second-order information. In *NIPS Workshop on Optimization for Machine Learning*, 2015.
- Nicolas Le Roux and Andrew W Fitzgibbon. A fast natural newton method. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 623–630, 2010.
- Nicolas Le Roux, Manzagol Pierre-antoine, and Yoshua Bengio. Topnnonoute online natural gradient algorithm. In J.C. Platt, D. Koller, Y. Singer, and S.T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 849–856. Curran Associates, Inc., 2008.
- Jason D Lee, Yuekai Sun, and Michael A Saunders. Proximal newton-type methods for minimizing composite functions. *SIAM Journal on Optimization*, 24(3):1420–1443, 2014.
- Moshe Lichman. UCI machine learning repository, 2013. URL <http://archive.ics.uci.edu/ml>.
- Chih-Jan Lin, Rudy C Weng, and S Sathiya Keerthi. Trust region newton method for logistic regression. *Journal of Machine Learning Research*, 9(Apr):627–650, 2008.
- James Martens. Deep learning via hessian-free optimization. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 735–742, 2010.
- Peter McCullagh and John A Nelder. *Generalized linear models*, volume 2. Chapman and Hall London, 1989.
- John A Nelder and R. Jacob Baker. *Generalized linear models*. Wiley Online Library, 1972.
- Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$. In *Doklady AN SSSR*, volume 269, pages 543–547, 1983.
- Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer, 2004.
- Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.
- Farbod Roosta-Khorasani and Michael W Mahoney. Sub-sampled newton methods i: Globally convergent algorithms. *arXiv preprint arXiv:1601.04737*, 2016a.
- Farbod Roosta-Khorasani and Michael W Mahoney. Sub-sampled newton methods ii: Local convergence rates. *arXiv preprint arXiv:1601.04738*, 2016b.
- Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *arXiv preprint arXiv:1309.2388*, 2013.
- Nicol N Sra and Jiahui Yu, Simon Ginters, et al. A stochastic quasi-newton method for online convex optimization. In *International Conference on Artificial Intelligence and Statistics*, volume 7, pages 436–443, 2007.
- David F Shanno. Conditioning of quasi-newton methods for function minimization. *Mathematics of computation*, 24(111):647–656, 1970.
- Charles M Stein. Estimation of the mean of a multivariate normal distribution. *Annals of Statistics*, pages 1135–1151, 1981.
- Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.
- Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv:1011.3027*, 2010.
- Oriol Vinyals and Daniel Powey. Krylov subspace descent for deep learning. *arXiv preprint arXiv:1111.4259*, 2011.

Bayesian Decision Process for Cost-Efficient Dynamic Ranking via Crowdsourcing

Xi Chen*

Stern School of Business

New York University

New York, New York, 10012, USA

XCHEN3@STERN.NYU.EDU

Kevin Jiao

Stern School of Business

New York University

New York, New York, 10012, USA

JJIAO@STERN.NYU.EDU

Qihang Lin

Tippie College of Business

University of Iowa

Iowa City, Iowa, 52242, USA

QIANG-LIN@UIOWA.EDU

Editor: Qiang Lin

Abstract

Rank aggregation based on pairwise comparisons over a set of items has a wide range of applications. Although considerable research has been devoted to the development of rank aggregation algorithms, one basic question is how to efficiently collect a large amount of high-quality pairwise comparisons for the ranking purpose. Because of the advent of many crowdsourcing services, a crowd of workers are often hired to conduct pairwise comparisons with a small monetary reward for each pair they compare. Since different workers have different levels of reliability and different pairs have different levels of ambiguity, it is desirable to wisely allocate the limited budget for comparisons among the pairs of items and workers so that the global ranking can be accurately inferred from the comparison results. To this end, we model the active sampling problem in *crowdsourced ranking* as a Bayesian Markov decision process, which dynamically selects item pairs and workers to improve the ranking accuracy under a budget constraint. We further develop a computationally efficient sampling policy based on knowledge gradient as well as a moment matching technique for posterior approximation. Experimental evaluations on both synthetic and real data show that the proposed policy achieves high ranking accuracy with a lower labeling cost.

Keywords: crowdsourced ranking, Bayesian, Markov decision process, dynamic programming, knowledge gradient, moment matching

1. Introduction

Inferring the ranking over a set of items, such as documents, images, movies, or URL links, is an important learning problem with many applications in areas like web search, recommendation systems, online games, etc. An interesting problem related to rank inference is estimating a score for each item based on a certain criterion that the items can be ranked,

such as the score of relevance or the score of quality. Typically, both the ranking and the scores of items can be inferred from a collection of high-quality labels on the items. There are mainly two different types of labels. The label of the first type is associated with each individual item in order to characterize the property of the item itself, for example, a binary or an ordinal score (e.g., 5-point grade). The label of the second type is instead associated with a subset of items that reveal their relative properties, for example, a partial ranking that covers only this subset. Labels of both types can be obtained by soliciting the knowledge of human workers, depending on whether the worker is employed to evaluate a single item or to compare a subset of items according to a given criterion. In practice, a binary score usually cannot fully distinguish all items and ordinal scores from different workers are often inconsistent due to the difference in their understandings of the grades in the ordinal scoring scheme. Therefore, the second type of labels has been more widely adopted, which can effectively reduce the impact of misunderstanding among workers and is more appropriate for ranking fine-grained items with a large number of graduations (e.g., in our real data experiment on accessing reading difficulty of an article into one of twelve American grade levels). Moreover, empirical evidences show that the ranking accuracy of a human worker typically decreases when he or she has to compare many items at a time. For this reason, in this paper, we only consider the relative comparisons over *pairs* of items and the label from a human worker indicates which item is preferred to the other.

The traditional approach of conducting pairwise comparisons by a small group of experts is usually time consuming and expensive. It fails to meet the growing need of labeled data for ranking tasks. Because of the advent of online crowdsourcing services (Howe, 2006) such as Amazon Mechanical Turk, a more efficient and more economic approach has emerged: a large amount of unlabeled pairs of items are posted to a crowdsourcing platform, where a crowd of workers are hired to perform pairwise comparisons and provide labels of the assigned pairs. Given the labels from crowd workers, we can infer a global ranking over all items. We refer to the process of collecting pairwise labels and ranking items as *crowdsourced ranking*.

Despite its availability and scalability, challenges remain in crowdsourced ranking. A certain amount of monetary reward is paid to a worker for each pair of items he or she compares while there is usually only a fixed amount of budget available, limiting the total number of pairwise labels we can collect. Hence, there is a need for a budget-efficient decision process for allocating the budget over item pairs and workers. In particular, on crowdsourcing platforms, there are unreliable workers who submit their answers quickly but carelessly in order to obtain more monetary reward with less effort. Hence, the comparison results provided by crowd workers often contain non-negligible noise. As a remedy, multiple workers are hired to compare the same pair of items independently in the hope that the correct ranking can be recovered, and that the unreliable workers can be identified by comparing their answers with the rest of workers. However, each pairwise comparison will incur a pre-specified monetary cost. Without a careful control, such a repetitive labeling strategy often results in too many labels on the same pair by different workers, leading to a high cost. Furthermore, because of the diversity of their backgrounds and expertise, workers do not always agree with each other in the results of pairwise comparisons, especially when the two items in comparison are competitive to each other. We refer to such a competitive pair as an *ambiguous pair* since the ordering of them is more difficult to be determined.

*. The authors are listed in alphabetical order

Presumably, a greater budget should be spent on ambiguous pairs, but identifying ambiguous pairs under the budget constraint itself is a challenging problem, which requires some effective learning scheme. Given the trade-off between the labeling cost and the quality of ranking results, there are two fundamental challenges in crowdsourced ranking:

1. Given the inconsistent pairwise labels from crowd workers with different reliability, how to aggregate these labels into a global ranking over items.
2. With both unreliable workers and ambiguous pairs initially unidentified, how to incorporate a learning scheme with an efficient sampling procedure (over both pairs of items and workers) under the budget constraint to achieve the highest ranking accuracy.

To address these challenges, we need to first model the reliability of workers and the ambiguity of item pairs and analyze how they influence the pairwise label. To this end, we adopt a combination of the Bradley-Terry-Luce ranking model (Bradley and Terry, 1952; Luce, 1959) for modeling the comparison results and the Dawid-Skene model (Dawid and Skene, 1979) for workers' reliability. The reason why we adopt the Bradley-Terry-Luce model is that learning such a model will not only provide a ranking over items but also give a score to each item, which can be useful in many applications (e.g., providing player's rating in chess games). We measure the quality of the ranking inferred from the collected labels using the *Kendall's tau rank correlation coefficient* (Kendall's tau for short) with respect to the underlying true ranking.

Under such a model and a quality measure, we propose a dynamic sampling and ranking procedure which addresses the aforementioned two challenges in a unified framework. In particular, we first introduce the priors' latent true scores and workers' reliability and formulate the crowdsourced ranking problem into a finite-horizon Bayesian *Markov decision problem* (MDP), whose state variables correspond to the posterior distributions given the observed labels. Here, the number of stages is determined by the total budget, i.e., the total number of pairs that can be requested for labeling. As the budget level increases, the size of the state space grows at an exponential rate, which makes the exact solving of such a MDP problem intractable. To address the computational difficulty, we propose an efficient sampling strategy called *approximated knowledge gradient* (AKG) policy based on the popular knowledge gradient policy (Powell, 2010; Frazier, 2009; Frazier et al., 2008; Ryzhov et al., 2012). The proposed policy dynamically chooses the next pair of items and the worker that together lead to a maximum expected improvement in Kendall's tau rank correlation coefficient. Finally, to determine the global ranking that maximizes the expected Kendall's tau, one needs to solve a maximum linear ordering problem (Grötschel et al., 1984) which is a NP-hard problem (and in fact, APX-hard (approximable-hard) (Mishra and Slicker, 2004)). To address this challenge, we propose a moment matching technique to approximate the posteriors in parametric forms so that the linear ordering problem under the approximated posterior can be easily solved by a simple sorting procedure.

The rest of the paper is organized as follows. In Section 2, we review the related literature. In Section 3, we introduce the model and the proposed policy under the simplified case where all workers are homogeneous and perfectly reliable. In Section 4, we extend our policy to the case where the crowd workers have heterogeneous reliability. In Section 5, we

present numerical results on both simulated and real datasets, followed by conclusions in Section 6. The detailed proofs and derivations are provided in the appendix.

2. Related Work

The dataset of partial rankings over items can be generated from a variety of sources including crowdsourcing services (Shah et al., 2016b), online competition games (e.g., Microsoft's TrueSkill system (Herbrich et al., 2007)), and online users' activities such as browsing, clicking and transactions that reveal certain preferences. Learning a global ranking of a large set of items by aggregating a collection of partial rankings/preferences has been an active research area for the past ten years (see, e.g., Gleich and Lin (2011); Negahban et al. (2012); Yi et al. (2013); Shah et al. (2016a,b); Rajkumar and Agarwal (2014); Lu and Boutlier (2014); Volkovs and Zemel (2014)). However, most work on rank aggregation considers a static estimation problem — inferring a global ranking based on a pre-existing dataset. The problem we consider here is related to but significantly different from these works because we model crowdsourced ranking as a dynamic procedure where the inference of ranking and collection of data proceed concurrently and influence each other.

The crowdsourced ranking problem we considered has a close connection with the dynamic sorting problem using noisy pairwise comparisons, which has been studied by several authors (Ailon, 2012; Braverman and Mossel, 2008; Radinsky and Ailon, 2011; Wauthier et al., 2013; Jameson and Nowak, 2011). However, these papers assume the noise of pairwise comparison results has the same distribution for all pairs, which is not reasonable in crowdsourced ranking because workers usually rank significantly different items more correctly than they do for similar items. The approaches proposed by Pfeiffer et al. (2012) and Qian et al. (2015) assume that the labeling noise depends on the latent qualities or features of the items. However, their approaches do not model the reliability of workers in the decision process. In contrast, our approach allows a label's noise to depend not only on the items themselves, but also on the reliability of the worker who provides the label. The ranking model adopted in this paper, which combines the Bradley-Terry-Luce model and the Dawid-Skene model, was originally proposed in (Chen et al., 2013), which also considers a similar problem of Bayesian statistical decision-making for crowdsourced ranking. However, the sampling strategy developed in Chen et al. (2013), which prioritizes the pair of items and the worker with the highest information gain, is a simple heuristic without a well-defined objective function to be optimized. In contrast, our work chooses the expected Kendall's tau as the objective function to maximize, which guides the development of the knowledge gradient policy.

In addition to crowdsourced ranking, the problem of crowdsourced categorical labeling/classification has been extensively studied in the past five years. Most work aims at solving a static problem, which infers the categorical labels and workers' reliability based on a static problem (see, e.g., Dawid and Skene (1979); Raykar et al. (2010); Weinder et al. (2010); Whitehill et al. (2009)); Lin et al. (2012); Gao and Zhou (2013); Zhang et al. (2014)). Recently, some research has been devoted to dynamic sampling in crowdsourced classification (Karger et al., 2013b,a; Bachrach et al., 2012; Ertekin et al., 2012; Kamar et al., 2012; Ho et al., 2013; Chen et al., 2015). In particular, both Kamar et al. (2012) and Chen et al. (2015) utilized the Markov decision process to model the budget allocation (i.e.,

sampling over items and workers) process. Since we also adopt a Bayesian Markov decision process with a variant of knowledge gradient policy, the spirit of our method is similar to that in Chen et al. (2015). However, since the statistical model for a ranking problem is fundamentally different from that of a classification problem, the Markov decision process in this paper is significantly different from the one introduced by Chen et al. (2015) in many aspects such as the objective function, stage-wise rewards, transition probabilities, optimal policy, etc. For example, the policy by Chen et al. (2015) is designed to maximize the expected classification accuracy while our policy aims at maximizing the expected Kendall's tau with respect to the true ranking. In fact, even for a static problem with a given set of collected data, inferring the ranking with the maximum expected Kendall's tau is equivalent to a NP-hard maximum linear ordering problem while classifying items with a maximum expected accuracy can be done in closed-form by Bayesian decision rule. In this paper, we avoid this computational challenge by exploiting the structure of the expected Kendall's tau and approximating the posteriors using moment matching. We also note that, although one can view the problem of ranking K items as a problem of classifying $K(K-1)/2$ pairs (each pair is treated as an item in Chen et al. (2015)), such an approach increases the size of the problem and ignores the dependency between pairwise labels.

In addition, it is worth to note that the problem we consider here is different from the typical tasks in machine-learned ranking or learning to rank (Liu, 2009; Acharya, 2013) where some feature information is available for each item and training data is used to calibrate some statistical models for ranking new items. In contrast to these problems, the feature information is not necessary in our crowdsourced ranking problem. Moreover, besides being applied to ranking items directly, our methods can be utilized to collect training labels for learning to rank problems. According to the type of training data utilized, statistical ranking methods can be classified into three categories (Liu, 2009; Acharya, 2013): pointwise method, pairwise method and listwise method. The pointwise methods (Li et al., 2008; Cooper et al., 1992; Crammer and Singer, 2001) learn a ranking model based on the data of scores or ratings of items. The pairwise methods (Freund et al., 2003; Burges et al., 2005; Zheng et al., 2008; Cao et al., 2006) and the listwise methods (Xu and Li, 2007; Cao et al., 2007; Taylor et al., 2008; Kuo et al., 2009) learn a ranking model using pairwise comparison results or partial rankings over a subset of items. For the pairwise or listwise methods, the crowdsourced ranking technique we proposed can be used as an upstream procedure that provides high-quality pairwise/listwise comparison data which helps increase the accuracy of the models in the aforementioned papers.

3. Crowdsourced Ranking by Homogeneous Workers

In this section, we first consider a simplified setting where workers are homogeneous (we will clarify the meaning of "homogeneous workers" shortly). In Section 4, we further extend the developed method for homogeneous workers to heterogeneous workers with different levels of reliability.

3.1 Model Setup

We assume that there are K items (denoted by $\{1, \dots, K\}$) to be ranked and each item i has an *unknown* latent score $\theta_i > 0$ for $i = 1, 2, \dots, K$. Let $\theta = (\theta_1, \theta_2, \dots, \theta_K)^T$, where each

latent score θ_i models the intensity of preference to item i under some criterion. A *ranking* over K items $\{1, 2, \dots, K\}$ is a permutation/one-to-one mapping $\pi : \{1, 2, \dots, K\} \rightarrow \{1, 2, \dots, K\}$ and $\pi(i)$ is the *rank* of item i under π . We follow the convention that $\theta_i > \theta_j$ means item i is preferred to item j and thus item i should have a higher rank than item j . Therefore, the underlying *true ranking* π^* over K items is determined by the ranking of their latent scores, i.e.,

$$\pi^*(i) > \pi^*(j) \quad \text{if and only if} \quad \theta_i > \theta_j. \quad (1)$$

We note that the latent scores naturally provide a characterization of *ambiguity for a pair of items*: when the values of θ_i and θ_j are closer, the pair of item i and j is more ambiguous in the sense that the true ordering of them is less obvious.

The way we explore the ranking of θ_i 's is through the collection of workers' preferences on different pairs of items. Specifically, we will present only two items at a time to a worker, who will be asked to compare these two items according to the given ranking criterion. Each worker will not be asked to compare the same pair more than once. The results of comparisons will be collected over time and become our historical data, based on which, our task is to infer the true ranking π^* .

In this section, we consider a basic setup where the crowd workers are assumed to be *homogeneous*, meaning that the probabilistic outcomes of their comparisons are only affected by the ambiguities of pairs. More specifically, suppose a worker is randomly selected from the crowd to compare a pair of items i and j with $i < j$ and the comparison result is denoted by a random variable Y_{ij} :

$$Y_{ij} = \begin{cases} 1 & \text{if item } i \text{ is preferred to item } j \text{ by the randomly selected worker} \\ -1 & \text{if item } j \text{ is preferred to item } i \text{ by the randomly selected worker.} \end{cases} \quad (2)$$

The setting of homogeneous workers means the probability distribution of Y_{ij} takes the following form

$$\Pr(Y_{ij} = 1) = \frac{\theta_i}{\theta_i + \theta_j} \quad \text{and} \quad \Pr(Y_{ij} = -1) = \frac{\theta_j}{\theta_i + \theta_j} \quad \text{for } i, j = 1, 2, \dots, K. \quad (3)$$

The probabilistic model we used in (3) is the well-known Bradley-Terry-Luce (BTL) model (Bradley and Terry, 1952; Luce, 1959). We choose this model for the distribution of Y_{ij} because it admits a simple structure and well fits our framework of dynamic sampling. Furthermore, our method developed for the BTL model can be easily extended to the case of heterogeneous workers which will be studied in Section 4.

It is worthwhile to mention that other comparison models can potentially be implemented here. Considering a simplified version of the Thurstone model (Thurstone, 1927) in which each object i has a score following $N(\theta_i, 1)$, then we have

$$\Pr(Y_{ij} = 1) = \Phi\left(\frac{\theta_i - \theta_j}{\sqrt{2}}\right) \quad \text{and} \quad \Pr(Y_{ij} = -1) = \Phi\left(\frac{\theta_j - \theta_i}{\sqrt{2}}\right).$$

The problem can still be formulated using a Bayesian decision process framework. However, there are several reasons why the BTL model is favored in this paper. First of all, moment

matching under the Thurstone model does not have closed-form solutions and hence we must rely on numerical scheme to compute the first and second moments of the posterior. Second, using moment matching approach, because the posterior is an n -dimensional multivariate Gaussian distribution, we need to update $n(n+1)/2$ parameters (the number of mean parameters plus the number of off-diagonal elements of the covariance matrix) during each iteration of the algorithm whereas with Dirichlet posterior there are only n parameters. Last but not least, with Thurstone model the ranking is no longer a simple sorting of parameters, which is a feature of the BTL model as shown in Theorem 2.

Since each worker can compare the same pair at most once, we assume the size of the crowd workers is large enough so that the distribution of Y_{ij} stays the same after sampling workers without replacement. Note that we can assume $\sum_{i=1}^K \theta_i = 1$ without loss of generality since the distribution of Y_{ij} in (3) remains unchanged if we multiply each θ_i by the same positive constant. The probability $\frac{\theta_i}{\theta_i + \theta_j}$ in (3) can also be interpreted as the percentage of workers in the crowd who prefer item i to item j .

Since the probabilistic model (3) does not incorporate or reveal the quality of each worker in the comparison result, in the subsequent study of this section, we only need to focus on how to dynamically select pairs of items to compare. The worker will be selected randomly from the crowd. A dynamic choice over workers will be incorporated into our method in Section 4 where the performance of workers is modeled heterogeneously.

3.2 Bayesian Decision Process

In a typical crowdsourcing marketplace, a monetary cost must be paid to a worker every time this worker completes a task such as comparing a pair of items. We assume the cost for each comparison is one unit and the total budget available is T units so that at most T pairs (repetition allowed) can be compared in total. Since comparing different pairs will generate different historical data and reveal different information about the true ranking, it is critical to dynamically determine the right sequence of pairs to compare in order to maximize the final ranking accuracy, especially when the budget T is small.

In the traditional offline setting, one needs to determine T pairs at a time beforehand and request the comparisons on those pairs in a batch. The potential problem of such a static approach is that the budget T is not spent in an efficient way to discover the true ranking. In fact, the distribution in (3) implies that, when two items have similar latent scores, workers will provide highly inconsistent preferences and it is hard to reach an agreement on such a pair. In this case, the comparison results will be very noisy and one needs to spend more budget on this pair in order to rank them correctly. In contrast, when two items have significantly different latent scores, workers will provide consistent answers so that the additional information we can obtain is little from repeatedly comparing the same two items. In this case, one might want to reduce the budget on such a pair. Unfortunately, without any prior knowledge of the latent scores, it is impossible to decide how much budget should be spent on each pair before observing some comparison results.

In order to efficiently allocate the limited total budget over all pairs, we consider a dynamic crowdsourced ranking policy (Algorithm 1) where only one pair of items is selected and presented to a worker at each time based on historical comparison results. This online

method allows the budget to be adaptively shifted towards the ambiguous pairs so that the final ranking accuracy can be improved.

In particular, given the total budget T , the dynamic decision process consists of T stages and, in stage $t = 0, 1, \dots, T-1$, a pair of items (i_t, j_t) with $i_t < j_t$ is presented to a randomly selected worker and we receive the comparison result $Y_{i_t j_t}$ defined in (2) and (3). The historical comparison results up to stage t can be summarized by a $K \times K$ matrix M^t with its entry¹ M^t_{ij} equal to the number of times item i is preferred to item j up to stage t . For each stage t where the pair (i_t, j_t) is compared, we define Δ^t to be a sparse $K \times K$ matrix with only one non-zero element: $\Delta^t_{i_t j_t} = 1$ if $Y_{i_t j_t} = 1$ and $\Delta^t_{j_t i_t} = 1$ if $Y_{i_t j_t} = -1$. By its definition, M^t can be updated iteratively as follows

$$M^0 = \mathbf{0}, \quad M^{t+1} = M^t + \Delta^t \quad \text{for } t = 0, 1, \dots, T-1, \quad (4)$$

where $\mathbf{0}$ denotes the $K \times K$ all-zero matrix.

We denote an *adaptive dynamic budget allocation/sampling policy* by $\mathcal{A} = \{(i_t, j_t)\}_{t=0,1,\dots,T-1}$ where $(i_t, j_t) = (i_t(M^t), j_t(M^t))$ depends on the previous comparison results through M^t . Our goal is to find the best \mathcal{A} so that the inferred ranking based on all the historical comparisons (represented by M^T) achieves the highest accuracy.

To measure the accuracy of an inferred ranking π , we adopt the popular evaluation criterion — normalized *Kendall's tau rank correlation coefficient* (Kendall, 1938) between π and π^* (Kendall's tau for short):

$$\begin{aligned} \tau(\pi, \pi^*) &\equiv \frac{|\{(i, j) : i < j, (\pi(i) - \pi(j))(\pi^*(i) - \pi^*(j)) > 0\}|}{K(K-1)/2} \\ &= \frac{2}{K(K-1)} \sum_{i \neq j} \mathbf{1}_{\{\pi(i) > \pi(j)\}} \mathbf{1}_{\{\theta_i > \theta_j\}}, \end{aligned} \quad (5)$$

where $\mathbf{1}_{\{\cdot\}}$ denotes the indicator function. Here, the numerator counts the number of pairs that π and π^* agree with each other and the denominator is the total number of pairs over K items. Hence, $\tau(\pi, \pi^*) \in [0, 1]$ and represents the percentage of agreements between π and π^* . The ranking accuracy of π is higher when $\tau(\pi, \pi^*)$ is closer to one and $\pi = \pi^*$ if and only if $\tau(\pi, \pi^*) = 1$.

However, we cannot infer a ranking based on the collected data by directly maximizing $\tau(\pi, \pi^*)$ because π^* and θ are unknown. To address this challenge, we adopt a Bayesian framework by proposing a prior distribution on θ and infer a ranking π that maximizes the posterior expectation of $\tau(\pi, \pi^*)$. Recall that the vector of latent scores θ is assumed to lie in the simplex

$$\Delta \equiv \left\{ \theta \in \mathbb{R}^K \mid \sum_{i=1}^K \theta_i = 1, \theta_i > 0 \right\}. \quad (6)$$

It is natural to assume that θ is drawn from a *Dirichlet prior distribution* parameterized by $\alpha^0 = (\alpha_1^0, \dots, \alpha_K^0)^T$ with $\alpha_i^0 > 0$ for all i (note that Dirichlet distribution of order K is supported on Δ). Namely,

$$\theta \sim \text{Dir}(\alpha^0) = \frac{1}{B(\alpha^0)} \prod_{i=1}^K g_i^{\alpha_i-1},$$

1. In this paper, the notation A_{ij} represents the entry in the i -th row and j -th column of matrix A .

where $B(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^K \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^K \alpha_i)}$ and $\Gamma(x) \equiv \int_0^\infty \lambda^{x-1} e^{-\lambda} d\lambda$ is the gamma function. Given the comparison data M^t up to stage t and the probability distribution of each comparison result in (3), the density function of the posterior distribution of $\boldsymbol{\theta}$ takes the following form,

$$p(\boldsymbol{\theta}|M^t, \boldsymbol{\alpha}^0) = \frac{1}{H(M^t, \boldsymbol{\alpha}^0)} \prod_{i \neq j} \binom{\theta_i}{\theta_i + \theta_j} \prod_i \theta_i^{\alpha_i^0 - 1} = \frac{1}{H(M^t, \boldsymbol{\alpha}^0)} \prod_{i < j} (\theta_i + \theta_j)^{M_{ij}^t + M_{ji}^t}, \quad (7)$$

where $\boldsymbol{\beta}^t = (\beta_1^t, \beta_2^t, \dots, \beta_K^t)^T$ with $\beta_i^t \equiv \sum_{j \neq i} M_{ij}^t$, i.e., the number of times item i is preferred to another item up to stage t , and

$$H(M^t, \boldsymbol{\alpha}^0) \equiv \int_{\Delta} \frac{\prod_{i=1}^K \theta_i^{\beta_i^t + \alpha_i^0 - 1}}{\prod_{i < j} (\theta_i + \theta_j)^{M_{ij}^t + M_{ji}^t}} d\boldsymbol{\theta},$$

is the normalization constant.

With this posterior distribution in place and with M^t at any stage t , we can infer a ranking $\hat{\pi}_t$ to maximize the posterior expected ranking accuracy measured by its Kendall's tau with respect to π^* , namely, to find

$$\hat{\pi}_t \in \arg \max_{\pi} \mathbb{E} [\tau(\pi, \pi^*) | M^t, \boldsymbol{\alpha}^0], \quad (8)$$

where the expectation is taken with respect to the posterior distribution $p(\boldsymbol{\theta} | M^t, \boldsymbol{\alpha}^0)$ in (7). We denote the corresponding maximum posterior expected accuracy by $h(M^t)$, i.e.,

$$h(M^t) \equiv \max_{\pi} \mathbb{E} [\tau(\pi, \pi^*) | M^t, \boldsymbol{\alpha}^0], \quad (9)$$

where the dependence of h on the prior $\boldsymbol{\alpha}^0$ is suppressed for notational simplicity. We are interested in finding a dynamic budget allocation policy $\mathcal{A} = \{(i_t, j_t)\}_{t=0,1,\dots,T-1}$ that maximizes $h(M^T)$, i.e., the final expected ranking accuracy when the budget is exhausted. This problem can be stated as

$$\max_{\mathcal{A}} \mathbb{E}^{\mathcal{A}} [h(M^T) | \boldsymbol{\alpha}^0], \quad (10)$$

where $\mathbb{E}^{\mathcal{A}}$ represents the expectation over the sample paths (i.e., the sampled pairs and outcomes) generated by the policy \mathcal{A} .

The maximization problem in (10) can be formulated as a T -stage Bayesian Markov decision process (MDP), where the *state variable* is the posterior distribution in (7) or simply the matrix M^t . The *state space* at each stage t denoted by \mathcal{S}^t takes the form of

$$\mathcal{S}^t = \left\{ M^t \in \mathbb{Z}_{\geq 0}^{K \times K} : \sum_{i,j} M_{ij}^t = t \right\}, \quad (11)$$

where $\mathbb{Z}_{>0}$ denotes the set of non-negative integers. The state variable makes a transition according to (4) given the observed comparison result Y_{i_t, j_t} , where the sampled pair (i_t, j_t) is determined by the policy \mathcal{A} . The expected transition probabilities take the form of,

$$\mathbb{E} [\Pr(Y_{ij} = 1) | M^t, \boldsymbol{\alpha}^0] = \mathbb{E} \left[\frac{\theta_i}{\theta_i + \theta_j} | M^t, \boldsymbol{\alpha}^0 \right] \quad (12)$$

$$\mathbb{E} [\Pr(Y_{ij} = -1) | M^t, \boldsymbol{\alpha}^0] = \mathbb{E} \left[\frac{\theta_j}{\theta_i + \theta_j} | M^t, \boldsymbol{\alpha}^0 \right] \quad (13)$$

for $1 \leq i < j \leq K$ and the expectation is taken over the posterior of $\boldsymbol{\theta}$ in (7). To complete the definition of our Bayesian MDP for crowdsourced ranking, we still need to define the *stage-wise reward*. To this end, we rewrite $h(M^T)$ in (10) as a telescopic sum,

$$h(M^T) = \sum_{t=0,1,\dots,T-1} R(M^t, i_t, j_t, Y_{i_t, j_t}); \quad R(M^t, i_t, j_t, Y_{i_t, j_t}) \equiv h(M^{t+1}) - h(M^t), \quad (14)$$

and note that $R(M^t, i_t, j_t, Y_{i_t, j_t}) = h(M^{t+1}) - h(M^t)$ only depends on $M^t, i_t, j_t, Y_{i_t, j_t}$. Given (14), the maximization problem (10) is equivalent to

$$\begin{aligned} \max_{\mathcal{A}} \mathbb{E}^{\mathcal{A}} \left[h(M^0) + \sum_{t=0}^{T-1} R(M^t, i_t, j_t, Y_{i_t, j_t}) | \boldsymbol{\alpha}^0 \right] \\ = h(M^0) + \max_{\mathcal{A}} \mathbb{E}^{\mathcal{A}} \left[\sum_{t=0}^{T-1} \mathbb{E} [R(M^t, i_t, j_t, Y_{i_t, j_t}) | M^t, \boldsymbol{\alpha}^0] | \boldsymbol{\alpha}^0 \right]. \end{aligned} \quad (15)$$

From (15), it is clear that $R(M^t, i_t, j_t, Y_{i_t, j_t})$ is the *stage-wise reward*, which can be interpreted as the improvement of the expected ranking accuracy after receiving the comparison result Y_{i_t, j_t} at stage t for $t = 0, 1, \dots, T-1$.

Given the Bayesian MDP in place, we can apply the dynamic programming (DP) algorithm (a.k.a. backward induction) (Puterman, 2005) to compute the optimal policy. Although DP finds the optimal policy, its computation is intractable because:

1. The sophisticated form of the posterior distribution in (7) makes it difficult to evaluate the posterior expected ranking accuracy $\mathbb{E} [\tau(\pi, \pi^*) | M^t, \boldsymbol{\alpha}^0]$ in (9) and the expected transition probabilities in (12) and (13).
2. The maximization problem (9) for solving the optimal posterior expected ranking accuracy is essentially a linear ordering problem (Grötschel et al., 1984), which is NP-hard in general (see Section 3.3 for more details).
3. The size of the state space \mathcal{S}^t grows exponentially in t according to (11), which is known as the curse of dimensionality that prevents us from solving (15) exactly with the standard techniques such as value iteration, policy iteration and linear programming.

To address these challenges, we propose an approximated knowledge gradient policy (AKG) in the next Section.

3.3 Approximated Knowledge Gradient Policy

In this section, we describe an approximated policy to solve (10), which is computationally efficient and still provides an inferred ranking with high quality. The proposed approximation policy belongs to the family of *knowledge gradient* (KG) policies (Gupta and Miescke, 1996; Frazier et al., 2008; Powell, 2010; Ryzhov et al., 2012), which is essentially a single-step look-ahead policy. In our problem, the KG policy will sample the next pair of items with the highest expected stage-wise reward in each stage, i.e., choosing the pair (i_t, j_t) such that

$$\begin{aligned}
 (i_t, j_t) &\in \arg \max_{i < j} \mathbb{E} [R(M^t, i_t, j_t, Y_{i_t j_t}) | M^t, \alpha^0] \\
 &= \arg \max_{i < j} \mathbb{E} [\Pr(Y_{ij} = 1) | M^t, \alpha^0] R(M^t, i_t, j_t, 1) \\
 &\quad + \mathbb{E} [\Pr(Y_{ij} = -1) | M^t, \alpha^0] R(M^t, i_t, j_t, -1).
 \end{aligned} \tag{16}$$

Despite its simplicity and wide applicability, the implementation of the KG policy for our problem in (16) is still computationally intractable since we have to evaluate the expected stage-wise reward $\mathbb{E}[R(M^t, i_t, j_t, Y_{i_t j_t}) | M^t, \alpha^0]$, where two main challenges will arise.

First, we have to evaluate the transition probabilities (12) and (13) as well as the ranking accuracy (9), which can be written as

$$\begin{aligned}
 h(M^t) &= \max_{\pi} \mathbb{E} [\tau(\pi, \pi^*) | M^t, \alpha^0] \\
 &= \max_{\pi} \frac{2 \sum_{i \neq j} \mathbb{E} [\mathbf{1}_{\{\pi(i) > \pi(j)\}} \mathbf{1}_{\{\theta_i > \theta_j\}} | M^t, \alpha^0]}{K(K-1)} \\
 &= \max_{\pi} \frac{2 \sum_{i \neq j} \mathbf{1}_{\{\pi(i) > \pi(j)\}} \Pr(\theta_i > \theta_j | M^t, \alpha^0)}{K(K-1)}.
 \end{aligned} \tag{17}$$

However, due to the complicated structure of the posterior distribution $p(\theta | M^t, \alpha^0)$ in (7), the expected transition probabilities (12) and (13) and the posterior probability $\Pr(\theta_i > \theta_j | M^t, \alpha^0)$ in (17) do not admit a closed form so that one needs to use multidimensional numerical integral or sampling techniques to compute their values. Note that for each stage t , we need to evaluate (12), (13) and $\Pr(\theta_i > \theta_j | M^t, \alpha^0)$ for all $K(K-1)/2$ pairs. When these quantities cannot be easily computed, the overall computational cost will be extremely expensive.

Second, even if the posterior probabilities $\Pr(\theta_i > \theta_j | M^t, \alpha^0)$ for all pairs are given, the maximization problem (17) with respect to a global ranking π is still very challenging. In fact, this problem is equivalent to the *maximum linear ordering problem (MAX-LOP)* described as follows. Let $G = (V, E, w)$ be a completed directed graph defined on a set V of K nodes, where the edge set E contains the directed arcs between all pairs of nodes and $w(i, j)$ refers to the weight associated with the arc from node i to node j . A tournament D is a sub-graph of G such that, for any pair of nodes i and j , D contains either the arc from i to j or the arc from j to i but not both. The MAX-LOP aims to find an acyclic tournament D with a maximum total weight on its arcs. If we interpret the arc from node i and node j as the preference of node i to node j under a ranking criterion, each acyclic tournament in G corresponds one-to-one to a global ranking of the nodes. Hence, MAX-LOP is equivalent to finding a ranking π such that the total weight $\sum_{\pi(i) > \pi(j)} w(i, j)$ is maximized. In problem (17), the nodes correspond to the K items and the weight $w(i, j) = \Pr(\theta_i > \theta_j | M^t, \alpha^0)$. Unfortunately, the MAX-LOP is known to be a NP-hard problem and in fact, APX (approximable)-complete and thus no PTAS (Polynomial Time Approximation Scheme) under P \neq NP (Mishra and Sikdar, 2004).

Given these two challenges, evaluating $\mathbb{E}[R(M^t, i_t, j_t, Y_{i_t j_t}) | M^t, \alpha^0]$ and solving (16) repeatedly at each stage are computationally intractable. To address this problem, we propose an *approximated knowledge gradient (AKG)* policy, which first replaces the stage-wise reward (14) by an approximated but computable reward and then chooses the pair that

maximizes this approximated reward. Our approximation scheme starts with approximating the posterior distribution $p(\theta | M^t, \alpha^0)$ in (7) recursively using a sequence of Dirichlet distributions $\text{Dir}(\alpha^t)$ for $t = 1, 2, \dots, T$ based on *moment matching*. One key benefit of such an approximation is that, at each stage t , the approximated posterior distribution of θ is still a Dirichlet distribution so that the NP-hard MAX-LOP problem in (17) will admit a simple solution via a sorting procedure (see Theorem 2).

Although there exist other methods for posterior approximation, these methods cannot be implemented as efficiently as moment matching in our application. For example, some methods such as variational inference (e.g., Beal, 2003; Paisley et al., 2012) minimize the KL-divergence between the exact posterior and the variational posterior, which requires an iterative optimization algorithm as a subroutine. Other methods like Gibbs sampler are computationally expensive in our case because the full conditional distribution does not have a closed form to allow easy sampling. In contrast, the proposed (algorithmic) moment matching admits a closed-form solution for approximating the posterior, which is computationally very efficient, and further provides a Dirichlet distribution as the approximated posterior, which facilitates solving the MAX-LOP. We note that the close-form update is critical for online crowdsourcing applications to reduce the computation time between two stages. In practice, since the crowd workers want to maximize their return in a short period of time, they may quit the current task if we let them wait for too long before we determine the next pair. Finally, we note that, although providing the theoretical guarantee for such an iterative approximation is hard in the Bayesian setup, we empirically show that the resulting AKG policy will generate a final ranking of a high accuracy with the limited budget.

Now we formally introduce the posterior approximation and AKG policy. Suppose $\theta \sim \text{Dir}(\alpha)$ for some parameters $\alpha \in \mathbb{R}^K$. We consider a basic case where only one comparison result Y_{ij} for a pair (i, j) with $i < j$ has been observed. In this case, we approximate the posterior $p(\theta | Y_{ij}, \alpha)$ by another Dirichlet distribution $\text{Dir}(\alpha')$ such that

$$\mathbb{E} [\theta_k | \theta \sim \text{Dir}(\alpha')] = \mathbb{E} [\theta_k | Y_{ij}, \alpha] \text{ for } k = 1, 2, \dots, K \tag{18}$$

$$\mathbb{E} \left[\sum_{k=1}^K \theta_k^2 \theta \sim \text{Dir}(\alpha') \right] = \mathbb{E} \left[\sum_{k=1}^K \theta_k^2 | Y_{ij}, \alpha \right]. \tag{19}$$

This system of equations has the following explicit characterization.

Proposition 1 *Suppose $\theta \sim \text{Dir}(\alpha)$ and Y_{ij} is the only comparison result for $i < j$. Let $\alpha_0 = \sum_{k=1}^K \alpha_k$ and $\alpha'_0 = \sum_{k=1}^K \alpha'_k$. The equations (18) and (19) can be represented as*

$$\begin{cases} \frac{\alpha'_i}{\alpha'_0} = \frac{(\alpha_i + \frac{1+Y_{ij}}{2})(\alpha_i + \alpha_j)}{\alpha_0(\alpha_i + \alpha_j + 1)} \\ \frac{\alpha'_j}{\alpha'_0} = \frac{(\alpha_j + \frac{1-Y_{ij}}{2})(\alpha_i + \alpha_j)}{\alpha_0(\alpha_i + \alpha_j + 1)} \\ \frac{\alpha'_k}{\alpha'_0} = \frac{\alpha_k}{\alpha_0} \text{ for } k \neq i, j \\ \sum_{k=1}^K \frac{\alpha'_k(\alpha'_k + 1)}{\alpha'_0(\alpha'_0 + 1)} = \frac{(\alpha_i + \frac{1+Y_{ij}}{2})(\alpha_i + \frac{3+Y_{ij}}{2})(\alpha_i + \alpha_j)}{\alpha_0(\alpha_0 + 1)(\alpha_i + \alpha_j + 2)} + \sum_{k \neq i, j} \frac{\alpha_k(\alpha_k + 1)}{\alpha_0(\alpha_0 + 1)}. \end{cases} \tag{20}$$

The proof of Proposition 1 is provided in the Appendix. We denote any α' that satisfies (18) and (19), and thus (20), by

$$\alpha' = \mathbf{MM}(\alpha, i, j, Y_{ij}). \quad (21)$$

Note that, given α, i, j and Y_{ij} , the right-hand sides of (20) are all constants so that we can solve $\alpha' = \mathbf{MM}(\alpha, i, j, Y_{ij})$ in a closed form. In fact, we denote the constants on the right hand sides of (20) as C_i, C_j, C_k (for $k \neq i, j$) and D , respectively. It is easy to show that $\sum_{k=1}^K C_k = 1$. The first three equalities in (20) imply that $\alpha'_k = C_k \alpha'_0$ for $k = 1, 2, \dots, K$ so that the fourth equality in (20) can be represented as $\sum_{k=1}^K C_k (\alpha'_k \alpha'_0 + 1) = D(\alpha'_0 + 1)$. Solving α'_0 from this equation leads to a closed-form for $\alpha' = \mathbf{MM}(\alpha, i, j, Y_{ij})$ as follows

$$\alpha'_0 = \frac{D-1}{\sum_{k=1}^K C_k^2 - D} \quad \text{and} \quad \alpha'_k = C_k \alpha'_0 \quad \text{for } k = 1, 2, \dots, K. \quad (22)$$

Although the above approximation scheme is established for only one comparison result, it produces a Dirichlet distribution $\text{Dir}(\alpha')$ which has the same type as the prior distribution $\text{Dir}(\alpha)$. Therefore, as more comparison results are generated sequentially, we can apply this approximation scheme iteratively after each comparison result. In particular, given a policy $\mathcal{A} = \{(i, j_t)\}_{t=0,1,\dots,T-1}$ with $i_t < j_t$ and the comparison results $\{Y_{i_t j_t}\}_{t=0,1,\dots,T-1}$, we define α^t recursively as

$$\alpha^{t+1} = \mathbf{MM}(\alpha^t, i_t, j_t, Y_{i_t j_t}) \quad (23)$$

for $t = 1, 2, \dots, T$. By doing so, we approximate the posterior distribution $p(\theta|M^t, \alpha^0)$ by the Dirichlet distribution $\text{Dir}(\alpha^t)$ for $t = 1, 2, \dots, T$.

With $p(\theta|M^t, \alpha^0)$ approximated by $\text{Dir}(\alpha^t)$, we can mitigate the two challenges mentioned at the beginning of this subsection. First, we can approximate (12) and (13) as

$$\mathbb{E}[\text{Pr}(Y_{ij} = 1)|M^t, \alpha^0] \approx \mathbb{E}\left[\frac{\theta_i}{\theta_i + \theta_j} | \theta \sim \text{Dir}(\alpha^t)\right] = \frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} \quad (24)$$

$$\mathbb{E}[\text{Pr}(Y_{ij} = -1)|M^t, \alpha^0] \approx \mathbb{E}\left[\frac{\theta_i}{\theta_i + \theta_j} | \theta \sim \text{Dir}(\alpha^t)\right] = \frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} \quad (25)$$

and approximate $\text{Pr}(\theta_i > \theta_j | M^t, \alpha^0)$ in (7) as

$$\text{Pr}(\theta_i > \theta_j | M^t, \alpha^0) \approx \text{Pr}(\theta_i > \theta_j | \theta \sim \text{Dir}(\alpha^t)) = \int_{\frac{1}{2}}^1 t^{\alpha_i^t - 1} (1-t)^{\alpha_j^t - 1} dt = I_{\frac{1}{2}}(\alpha_i^t, \alpha_j^t), \quad (26)$$

where $I_x(a, b) = \frac{B(x; a, b)}{B(a, b)}$ is known as the *regularized incomplete beta function* with $B(x; a, b) = \int_0^x \lambda^{a-1} (1-\lambda)^{b-1} d\lambda$ and $B(a, b) = \int_0^1 \lambda^{a-1} (1-\lambda)^{b-1} d\lambda$. Note that the approximated quantities in (24), (25) and (26) are much easier to compute than the original ones.

More importantly, the approximation (26) simplifies the NP-hard MAX-LOP in (17):

$$\max_{\pi} \mathbb{E}[\tau(\pi, \pi^*) | M^t, \alpha^0] \approx \max_{\pi} \mathbb{E}[\tau(\pi, \pi^*) | \theta \sim \text{Dir}(\alpha^t)].$$

The right-hand side is still a MAX-LOP but has a special structure so that it can be solved easily by a simple sorting procedure. In particular, the following theorem shows that when $\theta \sim \text{Dir}(\alpha)$, the optimal ranking in (16) can be obtained by sorting the components of α .

Theorem 2 Suppose $\theta \sim \text{Dir}(\alpha)$. We have

$$\begin{aligned} \Pi_{\alpha} &\equiv \{\pi | \pi \text{ is a ranking of } \{1, 2, \dots, K\} \text{ such that } \pi(i) > \pi(j) \text{ only if } \alpha_i \geq \alpha_j \text{ for all } i, j\} \\ &= \arg \max_{\pi} \mathbb{E}[\tau(\pi, \pi^*) | \theta \sim \text{Dir}(\alpha)] \end{aligned} \quad (27)$$

Proof We first show that $\arg \max_{\pi} \mathbb{E}[\tau(\pi, \pi^*) | \theta \sim \text{Dir}(\alpha)] \subset \Pi_{\alpha}$. Suppose $\hat{\pi}$ is the optimal solution of (27) where $\hat{\pi}(j) > \hat{\pi}(i)$ for a pair i and j with $\alpha_i > \alpha_j$. We put all items in a row with their ranks given by $\hat{\pi}$ decreasing from the left to the right and obtain a pattern like

$$X \cdots X_j \underbrace{X \cdots X}_S i X \cdots X,$$

where X represents some item different from i and j and S represents the set of items ranked between i and j . We will show that the objective value of (27) can be increased by switching the ranks of i and j .

Recall that the expected accuracy of $\hat{\pi}$ can be represented as

$$\begin{aligned} \mathbb{E}[\tau(\hat{\pi}, \pi^*) | \theta \sim \text{Dir}(\alpha)] &= \frac{2}{K(K-1)} \sum_{i' \neq j'} \mathbf{1}_{\hat{\pi}(i') > \hat{\pi}(j')} \text{Pr}(\theta_{i'} > \theta_{j'} | \theta \sim \text{Dir}(\alpha)) \quad (28) \\ &= \frac{2}{K(K-1)} \left[I_{\frac{1}{2}}(\alpha_i, \alpha_j) + \sum_{s \in S} I_{\frac{1}{2}}(\alpha_s, \alpha_j) + \sum_{s \in S} I_{\frac{1}{2}}(\alpha_i, \alpha_s) + C \right], \end{aligned}$$

where C is the summation of the remaining terms like $I_{\frac{1}{2}}(\alpha_{i'}, \alpha_{j'})$ which have either at least one of i' and j' not in $S \cup \{i, j\}$ or both i' and j' in S .

Note that switching the ranks of i and j does not change the values of the terms in C . In fact, after such a switch, we obtain a new ranking $\hat{\pi}'$ whose objective value in (27) is

$$\mathbb{E}[\tau(\hat{\pi}', \pi^*) | \theta \sim \text{Dir}(\alpha)] = \frac{2}{K(K-1)} \left[I_{\frac{1}{2}}(\alpha_j, \alpha_i) + \sum_{s \in S} I_{\frac{1}{2}}(\alpha_j, \alpha_s) + \sum_{s \in S} I_{\frac{1}{2}}(\alpha_s, \alpha_i) + C \right].$$

Using the fact that $I_{\frac{1}{2}}(a, b)$ is monotonically decreasing in a and monotonically increasing in b and noticing that $\alpha_i > \alpha_j$, we have

$$I_{\frac{1}{2}}(\alpha_j, \alpha_i) + \sum_{s \in S} I_{\frac{1}{2}}(\alpha_j, \alpha_s) + \sum_{s \in S} I_{\frac{1}{2}}(\alpha_s, \alpha_i) > I_{\frac{1}{2}}(\alpha_i, \alpha_j) + \sum_{s \in S} I_{\frac{1}{2}}(\alpha_s, \alpha_j) + \sum_{s \in S} I_{\frac{1}{2}}(\alpha_s, \alpha_s),$$

which implies $\mathbb{E}[\tau(\hat{\pi}', \pi^*) | \theta \sim \text{Dir}(\alpha)] > \mathbb{E}[\tau(\hat{\pi}, \pi^*) | \theta \sim \text{Dir}(\alpha)]$, contradicting with the optimality of $\hat{\pi}$. Hence, we can have $\hat{\pi}(i) > \hat{\pi}(j)$ only if $\alpha_i \geq \alpha_j$, meaning that $\hat{\pi} \in \Pi_{\alpha}$.

We then show $\arg \max_{\pi} \mathbb{E}[\tau(\pi, \pi^*) | \theta \sim \text{Dir}(\alpha)] = \Pi_{\alpha}$ by showing that $\mathbb{E}[\tau(\pi, \pi^*) | \theta \sim \text{Dir}(\alpha)]$ has the same value for any $\pi \in \Pi_{\alpha}$. Suppose $\hat{\pi}$ and $\hat{\pi}'$ both belong to Π_{α} and there exists a pair i and j with $i \neq j$ such that $\hat{\pi}(i) > \hat{\pi}(j)$ and $\hat{\pi}'(j) > \hat{\pi}'(i)$. By the definition of Π_{α} , we have $\alpha_i = \alpha_j$ so that

$$\text{Pr}(\theta_i > \theta_j | \theta \sim \text{Dir}(\alpha)) = I_{\frac{1}{2}}(\alpha_j, \alpha_i) = \frac{1}{2} = I_{\frac{1}{2}}(\alpha_i, \alpha_j) = \text{Pr}(\theta_j > \theta_i | \theta \sim \text{Dir}(\alpha)).$$

This means

$$\begin{aligned} & \mathbf{1}_{\hat{\pi}(i) > \hat{\pi}(j)} \Pr(\theta_i > \theta_j | \boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha})) + \mathbf{1}_{\hat{\pi}(j) > \hat{\pi}(i)} \Pr(\theta_j > \theta_i | \boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha})) \\ &= \mathbf{1}_{\hat{\pi}(i) > \hat{\pi}(j)} \Pr(\theta_i > \theta_j | \boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha})) + \mathbf{1}_{\hat{\pi}(j) > \hat{\pi}(i)} \Pr(\theta_j > \theta_i | \boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha})) \end{aligned}$$

for any pair i and j so that $\mathbb{E}[\tau(\hat{\pi}, \pi^*) | \boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha})] = \mathbb{E}[\tau(\hat{\pi}', \pi^*) | \boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha})]$ by the formulation (28), which completes the proof. \blacksquare

Given a parameter vector $\boldsymbol{\alpha}$, we denote any ranking in $\Pi_{\boldsymbol{\alpha}}$ by $\pi_{\boldsymbol{\alpha}}$. Using moment matching and Theorem 2, we can approximate the stage-wise reward $R(M^t, i, j, Y_{ij}^t)$ by

$$\begin{aligned} R(M^t, i, j, Y_{ij}^t) &= h(M^{t+1}) - h(M^t) \\ &= \max_{\pi} \mathbb{E}[\tau(\pi, \pi^*) | M^{t+1}, \boldsymbol{\alpha}^0] - \max_{\pi} \mathbb{E}[\tau(\pi, \pi^*) | M^t, \boldsymbol{\alpha}^0] \\ &\approx \max_{\pi} \mathbb{E}[\tau(\pi, \pi^*) | \boldsymbol{\theta} \sim \text{Dir}(\hat{\boldsymbol{\alpha}})] - \max_{\pi} \mathbb{E}[\tau(\pi, \pi^*) | \boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha}^j)] \\ &= \mathbb{E}[\tau(\pi_{\hat{\boldsymbol{\alpha}}}, \pi^*) | \boldsymbol{\theta} \sim \text{Dir}(\hat{\boldsymbol{\alpha}})] - \mathbb{E}[\tau(\pi_{\boldsymbol{\alpha}}, \pi^*) | \boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha}^j)] \\ &= \frac{2}{K(K-1)} \left(\sum_{i', j': \pi_{\hat{\boldsymbol{\alpha}}}(\hat{i}) > \pi_{\hat{\boldsymbol{\alpha}}}(\hat{j})} I_{\frac{1}{2}}(\hat{\alpha}_{i'}, \hat{\alpha}_{j'}) - \sum_{i', j': \pi_{\boldsymbol{\alpha}^j}(\hat{i}) > \pi_{\boldsymbol{\alpha}^j}(\hat{j})} I_{\frac{1}{2}}(\alpha_{i'}^j, \alpha_{j'}^j) \right) \\ &\equiv \tilde{R}(\boldsymbol{\alpha}^t, i, j, Y_{ij}^t) \end{aligned} \quad (29)$$

where $\hat{\boldsymbol{\alpha}} = \text{MM}(\boldsymbol{\alpha}^t, i, j, Y_{ij}^t)$, the third equality is from Theorem 2 and the fourth equality is due to (26). Putting (16), (24), (25), and (29) together, we can approximate the expected stage-wise reward $\mathbb{E}[R(M^t, i, j, Y_{ij}^t) | M^t, \boldsymbol{\alpha}^0]$ as

$$\begin{aligned} & \mathbb{E}[R(M^t, i, j, Y_{ij}^t) | M^t, \boldsymbol{\alpha}^0] \\ &= \mathbb{E}[\Pr(Y_{ij}^t = 1) | M^t, \boldsymbol{\alpha}^0] R(\boldsymbol{\alpha}^t, i, j, 1) + \mathbb{E}[\Pr(Y_{ij}^t = -1) | M^t, \boldsymbol{\alpha}^0] R(M^t, i, j, -1) \\ &\approx \frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} \tilde{R}(\boldsymbol{\alpha}^t, i, j, 1) + \frac{\alpha_j^t}{\alpha_i^t + \alpha_j^t} \tilde{R}(\boldsymbol{\alpha}^t, i, j, -1). \end{aligned} \quad (30)$$

The proposed AKG policy will choose the pair (i_t, j_t) that maximizes the approximated expected stage-wise reward in (30). As a summary, we describe the AKG policy as Algorithm 1.

It is noteworthy that it is easy to implement a *batch version* of Algorithm 1. In fact, the AKG policy in Algorithm 1 is known as an *index policy* where the right-hand side of (31), which calculates the marginal improvement on the ranking accuracy, can be treated as the index for each pair of items. The AKG policy selects the pair with the highest index at each stage. In the batch version, instead of selecting only one pair, one heuristics is to select the top B pairs and distribute to workers simultaneously, where B is a pre-defined batch size. Such a batch implementation can reduce the waiting time of crowd workers and thus accelerate the ranking procedure. Moreover, the AKG policy can be combined with some other batch optimization techniques (Wu and Prazler, 2016) to determine the optimal set of pairs to evaluate next.

Algorithm 1 Approximated Knowledge Gradient Policy with Homogeneous Workers

Initialization: Choose $\boldsymbol{\alpha}^0$ for the prior distribution. Let M^0 be a $K \times K$ all-zero matrix. For $t = 0, \dots, T - 1$ do

- 1: For each pair (i, j) with $i < j$, compute $\tilde{R}(\boldsymbol{\alpha}^t, i, j, 1)$ and $\tilde{R}(\boldsymbol{\alpha}^t, i, j, -1)$ according to (29).
- 2: Select (i_t, j_t) such that

$$(i_t, j_t) \in \arg \max_{i < j} \left[\frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} \tilde{R}(\boldsymbol{\alpha}^t, i, j, 1) + \frac{\alpha_j^t}{\alpha_i^t + \alpha_j^t} \tilde{R}(\boldsymbol{\alpha}^t, i, j, -1) \right] \quad (31)$$

and present item i_t and item j_t to a randomly selected worker and receive the comparison result $Y_{i_t j_t}^t$.

- 3: According to (21) and (22), compute

$$\boldsymbol{\alpha}^{t+1} = \text{MM}(\boldsymbol{\alpha}^t, i_t, j_t, Y_{i_t j_t}^t) \quad (32)$$

End For

Return: The aggregated ranking $\pi_{\boldsymbol{\alpha}^T}$ obtained by sorting the components of $\boldsymbol{\alpha}^T$.

4. Crowdsourced Ranking by Heterogeneous Workers

In the previous section, we considered the setting of homogeneous workers, where the comparison results are determined only by the intrinsic latent scores of items but not by the characteristics of workers. However, on crowdsourcing platforms, the quality of the workers varies a lot. Some workers are less reliable or lack of the domain knowledge; some workers are spammers, who either do not actually take a look at the assigned pairs or are robots pretending to be human workers, and thus provide random comparison results in order to quickly receive payment; some workers may be poorly informed (or even malicious), misunderstand the ranking criteria and thus always flip the comparison results. To identify the reliability of a worker, one can assign the same pair of items to multiple workers and hope to identify the unreliable ones whose labels are often different from the majority. However, the abuse of this strategy will result in hiring too many workers and lead to a quick growth of the monetary cost. In order to maximize the accuracy of the final ranking under the limited amount of budget, it is critical to balance the budget spent on estimating the reliability of the workers and learning the true ranking of the items. To formalize such trade-off, we incorporate the reliability of each worker to our previous Bayesian MDP and generalize the AKG policy to the heterogeneity of workers.

4.1 Model Setup

Similar to the previous setting, we assume that each item i has an unknown latent score $\theta_i > 0$ for $i = 1, 2, \dots, K$ which determines its true ranking π^* (see (1)) and $\boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha}^0)$. In the setting of heterogeneous workers, we assume that there are M crowd workers in total, denoted by $w = 1, 2, \dots, M$. If a pair of items i and j with $i < j$ is presented to the worker

w , we denote the returned comparison result by a random variable Y_{ij}^w such that

$$Y_{ij}^w = \begin{cases} 1 & \text{if item } i \text{ is preferred to item } j \text{ by worker } w \\ -1 & \text{if item } j \text{ is preferred to item } i \text{ by worker } w. \end{cases} \quad (33)$$

To model the reliability for workers, we introduce M latent parameters $\boldsymbol{\rho} = (\rho_1, \rho_2, \dots, \rho_M)^T$ of reliability with $\rho_w \in [0, 1]$ for worker w and assume Y_{ij}^w has the following distribution

$$\Pr(Y_{ij}^w = 1) = \rho_w \frac{\theta_i}{\theta_i + \theta_j} + (1 - \rho_w) \frac{\theta_j}{\theta_i + \theta_j} \quad (34)$$

$$\Pr(Y_{ij}^w = -1) = \rho_w \frac{\theta_j}{\theta_i + \theta_j} + (1 - \rho_w) \frac{\theta_i}{\theta_i + \theta_j} \quad (35)$$

for $1 \leq i < j \leq K$ and $w = 1, 2, \dots, M$. This model can be viewed as a combination of David-Skene model for categorical labeling tasks (David and Skene, 1979; Raykar et al., 2010; Karger et al., 2013a) and Bradley-Terry-Luce (BTL) model, which was first introduced in Chen et al. (2013). Such a mixture of BTL model is flexible and capable of modeling various types of workers. When $\rho_w = 1$, the distribution in (34) and (35) reduces to (3), and we refer to worker w with $\rho_w = 1$ as a ‘‘fully reliable’’ worker². Therefore, the reliability parameter ρ_w can be interpreted as the probability that worker w behaves as a random fully reliable workers in the previous section, namely, the one whose preference over a pair i and j follows a distribution in accordance with the BTL model (3). The worker with ρ_w closer to 1 is considered to be more reliable while a worker with ρ_w closer to 0 tends to be a poorly informed (or malicious) one who intentionally gives answers opposite to the majority (truth). Also, a worker is known as a spammer if the associated ρ_w is near 0.5 since this worker prefers i or j in any pair i and j with an equal probability regardless of their latent scores.

The reliability of each worker is unknown for the ranking task, which needs to be gradually identified during the comparison process. In the Bayesian framework, since the reliability parameter ρ_w is supported on $[0, 1]$, it can be naturally modeled to follow a Beta prior distribution, i.e., $\rho_w \sim \text{Beta}(\mu_w^0, \nu_w^0)$, for $w = 1, 2, \dots, M$, where $\boldsymbol{\mu}^0 = (\mu_1^0, \mu_2^0, \dots, \mu_M^0)$ and $\boldsymbol{\nu}^0 = (\nu_1^0, \nu_2^0, \dots, \nu_M^0)$ are positive parameters.

4.2 Bayesian Decision Process

In this section, we model the sequential decision problem with a finite budget of T in the setting of heterogeneous workers. Since the workers now have different levels of reliability, we can no longer randomly select a worker from the crowd in each stage. Instead, we need to adaptively determine not only which pair of items to be compared but also who should perform this comparison task according to the historical results so that the budget can be gradually shifted towards more reliable workers.

2. We note that the full reliability does not imply that the worker is capable of identifying the latent scores of items and always give the correct comparison result, i.e., preferring the item with a higher latent score. Instead, being fully reliable only means the worker tries her best to provide the preference after a careful consideration, and the inconsistency of comparisons among workers is mainly because the intrinsic ambiguity of the pair of items.

Suppose a pair of items (i_t, j_t) with $i_t < j_t$ is compared by a worker w_t in stage t and the comparison result is $Y_{i_t j_t}^{w_t}$ defined in (33). The historical comparison results up to stage t can be summarized by a $K \times K \times M$ tensor \mathbf{M}^t , which is updated iteratively as follows. In particular, at each stage t , we define $\boldsymbol{\Delta}^t$ to be a sparse $K \times K \times M$ tensor with only non-zero element: if $Y_{i_t j_t}^{w_t} = 1$, $\boldsymbol{\Delta}_{i_t j_t}^{w_t} = 1$ and if $Y_{i_t j_t}^{w_t} = -1$, $\boldsymbol{\Delta}_{j_t i_t}^{w_t} = 1$. Let

$$\mathbf{M}^0 = \mathbf{0}, \quad \mathbf{M}^{t+1} = \mathbf{M}^t + \boldsymbol{\Delta}^t \quad \text{for } t = 0, 1, \dots, T - 1, \quad (36)$$

where $\mathbf{0}$ is a $K \times K \times M$ all-zero tensor. In contrast to the matrix M^t in (4), each element in the tensor \mathbf{M}^t takes the value either zero or one because each worker is not allowed to compare the same pair more than once. The dynamic budget allocation policy is denoted by $\mathcal{A} = \{(i_t, j_t, w_t)\}_{t=0,1,\dots,T-1}$ where $(i_t, j_t, w_t) = (i_t(\mathbf{M}^t), j_t(\mathbf{M}^t), w_t(\mathbf{M}^t))$ depends on the previous comparison results through \mathbf{M}^t . The posterior distributions of $\boldsymbol{\theta}$ and $\boldsymbol{\rho}$ in stage t are denoted by $p(\boldsymbol{\theta}|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0)$ and $p(\boldsymbol{\rho}|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0)$, respectively.

Similar to the homogeneous worker setup, we adopt the Kendall’s tau (5) to measure the ranking accuracy. At each stage t , we denote the maximum posterior expected ranking accuracy by (with a slight abuse of notation)

$$\begin{aligned} h(\mathbf{M}^t) &\equiv \max_{\pi} \mathbb{E}[\tau(\pi, \pi^*)|\mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0] \\ &= \max_{\pi} \frac{2 \sum_{i \neq j} \mathbf{1}_{\pi(i) > \pi(j)} \Pr(\theta_i > \theta_j | \mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0)}{K(K-1)}. \end{aligned} \quad (37)$$

The maximizer in (37) is the optimal ranking inferred from the historical comparison results up to the stage t . Our goal is to search for the optimal policy \mathcal{A} that maximizes the final expected ranking accuracy $h(\mathbf{M}^T)$, i.e.,

$$\max_{\mathcal{A}} \mathbb{E}^{\mathcal{A}} [h(\mathbf{M}^T) | \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0]. \quad (38)$$

This maximization problem can be further reformulated in a telescopic sum

$$h(\mathbf{M}^0) + \max_{\mathcal{A}} \mathbb{E} \left[\sum_{t=0}^{T-1} \mathbb{E} \left[R(\mathbf{M}^t, i_t, j_t, w_t, Y_{i_t j_t}^{w_t}) | \mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0 \right] \middle| \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0 \right], \quad (39)$$

where

$$R(\mathbf{M}^t, i_t, j_t, w_t, Y_{i_t j_t}^{w_t}) \equiv h(\mathbf{M}^{t+1}) - h(\mathbf{M}^t), \quad (40)$$

is the *stage-wise reward* depending on $\mathbf{M}^t, i_t, j_t, w_t$ and $Y_{i_t j_t}^{w_t}$. It can be interpreted as the improvement of the expected ranking accuracy after receiving the comparison result at stage t . The *state variable* of the MDP (38) or (39) is the tensor \mathbf{M}^t which evolves according to (36) and the state space at each t is

$$\mathcal{S}^t = \left\{ \mathbf{M} \in \{0, 1\}^{K \times K \times M} : \sum_{i,j,w} \mathbf{M}_{jw} = t \right\}.$$

The expected transition probabilities of MDP (38) are

$$\mathbb{E} [\Pr(Y_{ij}^w = 1) | \mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0] = \mathbb{E} \left[\rho_w \frac{\theta_i}{\theta_i + \theta_j} + (1 - \rho_w) \frac{\theta_j}{\theta_i + \theta_j} | \mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0 \right] \quad (41)$$

$$\mathbb{E} [\Pr(Y_{ij}^w = -1) | \mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0] = \mathbb{E} \left[\rho_w \frac{\theta_j}{\theta_i + \theta_j} + (1 - \rho_w) \frac{\theta_i}{\theta_i + \theta_j} | \mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0 \right] \quad (42)$$

for $i, j = 1, 2, \dots, K$ and $w = 1, 2, \dots, M$. So far, we have modeled the sequential budget allocation in the heterogeneous worker setting as a Bayesian MDP. Due to the similar reasons that have been explained in Section 3.2, although the dynamic programming can be directly applied to solve the Bayesian MDP and obtain the optimal policy, it is computationally intractable. In fact, the Bayesian MDP (39) is even more challenging to solve than that for the homogeneous worker setting due to a much larger state space after introducing the reliability of workers. In the next subsection, we will propose a computationally efficient approximated knowledge gradient policy for (39).

4.3 Approximated Knowledge Gradient Policy

To solve the Bayesian MDP (39), we still consider the family of knowledge gradient (KG) policies. In our problem, the KG policy will select the pair of items and the worker that together give the highest expected stage-wise reward. In particular, at the t -stage, the KG policy for (39) will choose the pair (i_t, j_t) and the worker w_t such that

$$\begin{aligned} (i_t, j_t, w_t) &\in \arg \max_{i < j, w} \mathbb{E} [R(\mathbf{M}^t, i, j, w, Y_{ij}^w) | \mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0] \\ &= \arg \max_{i < j, w} \left\{ \mathbb{E} [\Pr(Y_{ij}^w = 1) | \mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0] R(\mathbf{M}^t, i, j, w, 1) \right. \\ &\quad \left. + \mathbb{E} [\Pr(Y_{ij}^w = -1) | \mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0] R(\mathbf{M}^t, i, j, w, -1) \right\}. \end{aligned} \quad (43)$$

To implement the KG policy (43), we encounter the same difficulties as when we implemented (16). Specifically, since the posterior distributions $p(\boldsymbol{\theta} | \mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0)$ and $p(\boldsymbol{\rho} | \mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0)$ are sophisticated and the MAX-LOP problem (37) is NP-hard, we cannot efficiently evaluate the stage-wise reward (40) and the transition probabilities (41) and (42). To obtain a computationally efficient policy, we follow the techniques in Section 3.3 to approximate the posterior distributions $p(\boldsymbol{\theta} | \mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0)$ and $p(\boldsymbol{\rho} | \mathbf{M}^t, \boldsymbol{\alpha}^0, \boldsymbol{\mu}^0, \boldsymbol{\nu}^0)$ recursively using a sequence of Dirichlet distributions $\text{Dir}(\boldsymbol{\alpha}^t)$ and a sequence of beta distributions $\text{Beta}(\mu_w^t, \nu_w^t)$, respectively, for $w = 1, 2, \dots, M$ and $t = 1, 2, \dots, T$. The parameters $\boldsymbol{\alpha}^t(\alpha_1^t, \alpha_2^t, \dots, \alpha_K^t)$, $\boldsymbol{\mu}^t = (\mu_1^t, \mu_2^t, \dots, \mu_M^t)$ and $\boldsymbol{\nu}^t = (\nu_1^t, \nu_2^t, \dots, \nu_M^t)$ will be chosen recursively based on a moment matching.

Suppose $\boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha})$ for some parameter vector $\boldsymbol{\alpha} \in \mathbb{R}^K$ and $\rho_w \sim \text{Beta}(\mu_w, \nu_w)$ for each w with $\boldsymbol{\mu} = (\mu_1, \mu_2, \dots, \mu_M)$ and $\boldsymbol{\nu} = (\nu_1, \nu_2, \dots, \nu_M)$. We consider a basic scenario where only one comparison result Y_{ij}^w from worker w for a pair (i, j) has been observed. We can approximate $p(\boldsymbol{\theta} | Y_{ij}^w, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\nu})$ by a Dirichlet distribution $\text{Dir}(\boldsymbol{\alpha}')$ and $p(\rho_w | Y_{ij}^w, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\nu})$ by

$$\mathbb{E} [\theta_k | \boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha}')] = \mathbb{E} [\theta_k | Y_{ij}^w, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\nu}] \text{ for } k = 1, 2, \dots, K \quad (44)$$

$$\mathbb{E} \left[\sum_{k=1}^K \theta_k^2 | \boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha}') \right] = \mathbb{E} \left[\sum_{k=1}^K \theta_k^2 | Y_{ij}^w, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\nu} \right] \quad (45)$$

$$\mathbb{E} [\rho_w | \rho_w \sim \text{Beta}(\mu_w', \nu_w')] = \mathbb{E} [\rho_w | Y_{ij}^w, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\nu}] \quad (46)$$

$$\mathbb{E} [\rho_w^2 + (1 - \rho_w)^2 | \rho_w \sim \text{Beta}(\mu_w', \nu_w')] = \mathbb{E} [\rho_w^2 + (1 - \rho_w)^2 | Y_{ij}^w, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\nu}]. \quad (47)$$

Note that we do not need to approximate $p(\rho_w | Y_{ij}^w, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\nu})$ for $w' \neq w$ since the worker w' has not performed any comparison so that $p(\rho_w | Y_{ij}^w, \boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\nu})$ is still the prior distribution $\text{Beta}(\mu_w, \nu_w)$. This system of equations has the following explicit characterization.

Proposition 3 Suppose $\boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha})$ and $\rho_w \sim \text{Beta}(\mu_w, \nu_w)$ for worker w and Y_{ij}^w is the only comparison result. Let $\alpha_0 = \sum_{k=1}^K \alpha_k$ and $\alpha'_k = \sum_{k=1}^K \alpha'_k$. The equations (44), (45), (46) and (47) can be represented as

$$\begin{cases} \frac{\alpha'_k}{\alpha_0} = \eta_{ij} w \frac{(\alpha_0+1)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_i+\alpha_j+1)} + (1 - \eta_{ij} w) \frac{\alpha_0(\alpha_i+\alpha_j)}{\alpha_0(\alpha_i+\alpha_j+1)} \\ \frac{\alpha'_j}{\alpha_0} = \eta_{ij} w \frac{\alpha_0(\alpha_i+\alpha_j)}{\alpha_0(\alpha_i+\alpha_j+1)} + (1 - \eta_{ij} w) \frac{\alpha_0(\alpha_i+\alpha_j)}{\alpha_0(\alpha_i+\alpha_j+1)} \\ \frac{\alpha'_k}{\alpha_0} = \frac{\alpha_k}{\alpha_0} \text{ for } k \neq i, j \\ \frac{\alpha'_k}{\alpha_0} = \frac{(\alpha_0+1)(\alpha_i+2)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_0+1)(\alpha_i+\alpha_j+2)} + (1 - \eta_{ij} w) \frac{\alpha_0(\alpha_0+1)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_0+1)(\alpha_i+2)(\alpha_i+\alpha_j)} \\ \quad + \eta_{ij} w \frac{\alpha_0(\alpha_0+1)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_0+1)(\alpha_i+\alpha_j+2)} + (1 - \eta_{ij} w) \frac{\alpha_0(\alpha_0+1)(\alpha_i+\alpha_j)}{\alpha_0(\alpha_0+1)(\alpha_i+2)(\alpha_i+\alpha_j)} \\ \quad + \sum_{k \neq i, j} \frac{\alpha_k(\alpha_k+1)}{\alpha_0(\alpha_k+1)} \\ \frac{\mu_w + \nu_w}{(\mu_w + \nu_w + 1)} = \eta_{ij} w \frac{\mu_w + (1 - Y_{ij}^w)/2}{(\mu_w + \nu_w + 1)(\mu_w + \nu_w + 2)} + (1 - \eta_{ij} w) \frac{\mu_w + (1 - Y_{ij}^w)/2}{(\mu_w + \nu_w + 1)(\mu_w + \nu_w + 2)} \\ \frac{\mu_w + \nu_w}{(\mu_w + \nu_w + 1)} = \eta_{ij} w \frac{\mu_w + \nu_w + 1}{(\mu_w + \nu_w + 1)(\mu_w + \nu_w + 2)} + (1 - \eta_{ij} w) \frac{\mu_w + \nu_w + 1}{(\mu_w + \nu_w + 1)(\mu_w + \nu_w + 2)} \\ \frac{\mu_w + \nu_w}{(\mu_w + \nu_w + 1)} = \eta_{ij} w \frac{\mu_w + \nu_w + 1}{(\mu_w + \nu_w + 1)(\mu_w + \nu_w + 2)} + (1 - \eta_{ij} w) \frac{\mu_w + \nu_w + 1}{(\mu_w + \nu_w + 1)(\mu_w + \nu_w + 2)} \\ \quad + \eta_{ij} w \frac{(\nu_w + 1 - Y_{ij}^w)/2}{(\nu_w + 1 + Y_{ij}^w)/2} + (1 - \eta_{ij} w) \frac{(\nu_w + 1 - Y_{ij}^w)/2}{(\nu_w + 1 + Y_{ij}^w)/2} \\ \quad + \eta_{ij} w \frac{(\nu_w + 1 - Y_{ij}^w)/2}{(\nu_w + 1 + Y_{ij}^w)/2} + (1 - \eta_{ij} w) \frac{(\nu_w + 1 - Y_{ij}^w)/2}{(\nu_w + 1 + Y_{ij}^w)/2} \\ \quad + (1 - \eta_{ij} w) \frac{(\nu_w + 1 + Y_{ij}^w)/2}{(\nu_w + 1 + Y_{ij}^w)/2}. \end{cases} \quad (48)$$

where $\eta_{ij} w = \frac{[(1+Y_{ij}^w)\mu_w + (1-Y_{ij}^w)\nu_w]\alpha_0}{[(1+Y_{ij}^w)\mu_w + (1-Y_{ij}^w)\nu_w]\alpha_0 + [(1+Y_{ij}^w)\mu_w + (1-Y_{ij}^w)\nu_w]\alpha_0}$.

The proof of Proposition 3 is given in Appendix. We denote any $\boldsymbol{\alpha}'$, μ_w' and ν_w' that satisfy (44), (45), (46) and (47), and thus (48), by

$$\boldsymbol{\alpha}' = \text{MM}_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}, i, j, w, Y_{ij}^w) \quad \text{and} \quad (\mu_w', \nu_w') = \text{MM}_{\mu\nu}(\boldsymbol{\alpha}, i, j, w, Y_{ij}^w). \quad (49)$$

Although the equations in Proposition 3 are more complicated than those in Proposition 1, the right-hand sides of (48) are still constants for any given $i, j, w, Y_{ij}^w, \boldsymbol{\alpha}, \mu_w$ and ν_w so that both $\boldsymbol{\alpha}' = \text{MM}_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}, i, j, w, Y_{ij}^w)$ and $(\mu_w', \nu_w') = \text{MM}_{\mu\nu}(\boldsymbol{\alpha}, i, j, w, Y_{ij}^w)$ can be solved in a closed form. In fact, we denote the constants on the right-hand sides of (20) as C_i, C_j, C_k (for $k \neq i, j$), D , E and F , respectively. It is easy to see that $\sum_{k=1}^K C_k = 1$. By the

same derivation for (22), we obtain the following closed form for $\alpha^l = \mathbf{MM}_\alpha(\alpha, i, j, w, Y_{ij}^w)$

$$\alpha_0^l = \frac{D-1}{\sum_{k=1}^K C_k^2 - D} \quad \text{and} \quad \alpha_k^l = C_k \alpha_0^l \quad \text{for } k = 1, 2, \dots, K, \quad (50)$$

which takes the same form as (22) but with the constants C_k for $k = 1, 2, \dots, K$ defined differently (which involve the information of worker w , i.e., μ_w and ν_w). Similarly, solving μ_w^l and ν_w^l from the last two equations in (48), we obtain the following closed form for $(\mu_w^l, \nu_w^l) = \mathbf{MM}_{\mu\nu}(\alpha, i, j, w, Y_{ij}^w)$

$$\mu_w^l = \frac{(F-1)E}{E^2 + (1-E)^2 - F} \quad \text{and} \quad \nu_w^l = \frac{(F-1)(1-E)}{E^2 + (1-E)^2 - F}. \quad (51)$$

Although the approximate scheme above is derived when there is only one comparison result, it generates a Dirichlet distribution $\text{Dir}(\alpha^l)$ for θ and a Beta distribution $\text{Beta}(\mu_w^l, \nu_w^l)$ for ρ_w and does not change the Beta distribution $\text{Beta}(\mu_{w'}^l, \nu_{w'}^l)$ for $w' \neq w$. The fact that the approximated posteriors take the same form as the priors suggests that we can apply this approximation scheme iteratively to approximate $p(\theta|\mathbf{M}^t, \alpha^0, \mu^0, \nu^0)$ and $p(\rho|\mathbf{M}^t, \alpha^0, \mu^0, \nu^0)$ for any given policy $\mathcal{A} = \{(i_t, j_t, w_t)\}_{t=0,1,\dots,T-1}$. In particular, let α^t, μ^t and ν^t be the sequences of parameters generated recursively as follows

$$\alpha^{t+1} = \mathbf{MM}_\alpha(\alpha^t, i_t, j_t, w_t, Y_{i_t j_t}^{w_t}) \quad (52)$$

$$(\mu_w^{t+1}, \nu_w^{t+1}) = \begin{cases} \mathbf{MM}_{\mu\nu}(\alpha^t, i_t, j_t, w_t, Y_{i_t j_t}^{w_t}) & \text{if } w = w_t \\ (\mu_w^t, \nu_w^t) & \text{if } w \neq w_t \end{cases} \quad (53)$$

for $t = 1, 2, \dots, T$. The posterior distributions $p(\theta|\mathbf{M}^t, \alpha^0, \mu^0, \nu^0)$ and $p(\rho|\mathbf{M}^t, \alpha^0, \mu^0, \nu^0)$ can be approximated by $\text{Dir}(\alpha^t)$ and $\Pi_{w=1,\dots,M} \text{Beta}(\mu_w^t, \nu_w^t)$, respectively.

Following the same strategy as in (24) and (25), we can approximate (41) and (42) as

$$\begin{aligned} & \mathbb{E} [\text{Pr}(Y_{ij}^w = 1) | \mathbf{M}^t, \alpha^0, \mu^0, \nu^0] \\ & \approx \mathbb{E} \left[\frac{\rho_w}{\theta_i + \theta_j} + (1 - \rho_w) \frac{\theta_j}{\theta_i + \theta_j} \middle| \theta \sim \text{Dir}(\alpha^t), \rho_w \sim \text{Beta}(\mu_w^t, \nu_w^t) \right] \\ & = \frac{\mu_w^t}{\mu_w^t + \nu_w^t} \frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} + \frac{\nu_w^t}{\mu_w^t + \nu_w^t} \frac{\alpha_j^t}{\alpha_i^t + \alpha_j^t} \end{aligned} \quad (54)$$

and

$$\begin{aligned} & \mathbb{E} [\text{Pr}(Y_{ij}^w = -1) | \mathbf{M}^t, \alpha^0, \mu^0, \nu^0] \\ & \approx \mathbb{E} \left[\frac{\rho_w}{\theta_i + \theta_j} + (1 - \rho_w) \frac{\theta_i}{\theta_i + \theta_j} \middle| \theta \sim \text{Dir}(\alpha^t), \rho_w \sim \text{Beta}(\mu_w^t, \nu_w^t) \right] \\ & = \frac{\mu_w^t}{\mu_w^t + \nu_w^t} \frac{\alpha_j^t}{\alpha_i^t + \alpha_j^t} + \frac{\nu_w^t}{\mu_w^t + \nu_w^t} \frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} \end{aligned} \quad (55)$$

and approximate $\text{Pr}(\theta_i > \theta_j | \mathbf{M}^t, \alpha^0, \mu^0, \nu^0)$ in (37) as

$$\text{Pr}(\theta_i > \theta_j | \mathbf{M}^t, \alpha^0, \mu^0, \nu^0) \approx \text{Pr}(\theta_i > \theta_j | \theta \sim \text{Dir}(\alpha^t)) = I_{\frac{1}{2}}(\alpha_j^t, \alpha_i^t). \quad (56)$$

The approximation (56) helps to simplify the NP-hard MAX-LOP in (37) as

$$\max_{\pi} \mathbb{E} [\tau(\pi, \pi^*) | \mathbf{M}^t, \alpha^0, \mu^0, \nu^0] \approx \max_{\pi} \mathbb{E} [\tau(\pi, \pi^*) | \theta \sim \text{Dir}(\alpha^t)],$$

where the right-hand side can be solved easily by sorting of the components of α^t according to Theorem 2.

Similar to (29), the stage-wise reward is approximated as

$$\begin{aligned} R(\mathbf{M}^t, i, j, w, Y_{ij}^w) &= \max_{\pi} \mathbb{E} [\tau(\pi, \pi^*) | \mathbf{M}^{t+1}, \alpha^0, \mu^0, \nu^0] - \max_{\pi} \mathbb{E} [\tau(\pi, \pi^*) | \mathbf{M}^t, \alpha^0, \mu^0, \nu^0] \\ &\approx \max_{\pi} \mathbb{E} [\tau(\pi, \pi^*) | \theta \sim \text{Dir}(\hat{\alpha})] - \max_{\pi} \mathbb{E} [\tau(\pi, \pi^*) | \theta \sim \text{Dir}(\alpha^t)] \\ &= \frac{2}{K(K-1)} \left(\sum_{\pi_{\alpha}(i') > \pi_{\alpha}(j')} I_{\frac{1}{2}}(\hat{\alpha}_{j'}, \hat{\alpha}_{i'}) - \sum_{\pi_{\alpha}(i') > \pi_{\alpha}(j')} I_{\frac{1}{2}}(\alpha_{j'}^t, \alpha_{i'}^t) \right) \\ &\equiv \tilde{R}(\alpha^t, i, j, w, Y_{ij}^w) \end{aligned} \quad (57)$$

where $\hat{\alpha} = \mathbf{MM}_\alpha(\alpha^t, i, j, w, Y_{ij}^w)$. Putting (54), (55), (56) and (57) together, we can approximate the expected stage-wise reward $\mathbb{E} [\tilde{R}(\mathbf{M}^t, i, j, w, Y_{ij}^w) | \mathbf{M}^t, \alpha^0, \mu^0, \nu^0]$ as

$$\begin{aligned} & \mathbb{E} [R(\mathbf{M}^t, i, j, w, Y_{ij}^w) | \mathbf{M}^t, \alpha^0, \mu^0, \nu^0] \\ &= \mathbb{E} [\text{Pr}(Y_{ij}^w = 1) | \mathbf{M}^t, \alpha^0, \mu^0, \nu^0] R(\alpha^t, i, j, w, 1) \\ &+ \mathbb{E} [\text{Pr}(Y_{ij}^w = -1) | \mathbf{M}^t, \alpha^0, \mu^0, \nu^0] R(\alpha^t, i, j, w, -1) \\ &\approx \left(\frac{\mu_w^t}{\mu_w^t + \nu_w^t} \frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} + \frac{\nu_w^t}{\mu_w^t + \nu_w^t} \frac{\alpha_j^t}{\alpha_i^t + \alpha_j^t} \right) \tilde{R}(\alpha^t, i, j, w, 1) \\ &+ \left(\frac{\mu_w^t}{\mu_w^t + \nu_w^t} \frac{\alpha_j^t}{\alpha_i^t + \alpha_j^t} + \frac{\nu_w^t}{\mu_w^t + \nu_w^t} \frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} \right) \tilde{R}(\alpha^t, i, j, w, -1). \end{aligned} \quad (58)$$

When the workers have various levels of reliability, our AKG policy will choose the pair (i_t, j_t) and present it to worker w_t so that (58) is maximized. The AKG policy for the setting of heterogeneous workers is formally presented as Algorithm 2. Note that when $\rho_w = 1$ for all w , we do not need to solve (46) and (47) anymore and thus the rest of the problem reduces to the homogeneous setting.

5. Experiment

In this section, we conduct empirical studies using both simulated and real data. We compare the proposed AKG algorithms to some existing methods in terms of ranking accuracy versus different levels of budget as well as computation time. We also show some interesting properties of the proposed AKG policies, e.g., how budget will be allocated over pairs of items with different levels of ambiguity and workers with different levels of reliability. The ranking accuracy is evaluated using the Kendall's tau as defined in (5).

Algorithm 2 Approximated Knowledge Gradient Policy with Heterogeneous Workers

Initialization: Choose α^0 , μ^0 and ν^0 for the prior distributions.

For $t = 0, \dots, T - 1$ **do**

- 1: For each pair (i, j) with $i < j$, compute $\tilde{R}(\alpha^t, i, j, w, 1)$ and $\tilde{R}(\alpha^t, i, j, w, -1)$ according to (57).
- 2: Select (i_t, j_t, w_t) such that

$$(i_t, j_t) \in \underset{i < j, w}{\operatorname{argmax}} \left[\left(\frac{\mu_w^t}{\mu_w^t + \nu_w^t} \frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} + \frac{\nu_w^t}{\mu_w^t + \nu_w^t} \frac{\alpha_j^t}{\alpha_i^t + \alpha_j^t} \right) \tilde{R}(\alpha^t, i, j, w, 1) \right. \\ \left. + \left(\frac{\mu_w^t}{\mu_w^t + \nu_w^t} \frac{\alpha_j^t}{\alpha_i^t + \alpha_j^t} + \frac{\nu_w^t}{\mu_w^t + \nu_w^t} \frac{\alpha_i^t}{\alpha_i^t + \alpha_j^t} \right) \tilde{R}(\alpha^t, i, j, w, -1) \right] \quad (59)$$

and present item i_t and item j_t to worker w_t and receive the comparison result $Y_{i_t, j_t}^{w_t}$.

- 3: According to (49), (50) and (51), compute

$$\alpha^{t+1} = \operatorname{MM}_\alpha(\alpha^t, i_t, j_t, w_t, Y_{i_t, j_t}^{w_t}) \quad (60)$$

$$\begin{pmatrix} \mu_w^{t+1} \\ \nu_w^{t+1} \end{pmatrix} = \begin{cases} \operatorname{MM}_{\mu\nu}(\alpha^t, i_t, j_t, w_t, Y_{i_t, j_t}^{w_t}) & \text{if } w = w_t \\ \begin{pmatrix} \mu_w^t \\ \nu_w^t \end{pmatrix} & \text{if } w \neq w_t \end{cases} \quad (61)$$

End For

Return: The aggregated ranking π_{α^T} obtained by sorting the components of α^T .

5.1 Simulated Study under the Homogeneous Workers Setting

In this section, we assume that all workers are fully reliable and investigate the performance of the AKG policy (Algorithm 1). Two scenarios are designed: 10 items with a total budget of 100, and 100 items with a total budget of 1000. Each scenario consists of 100 independent trials and the average ranking accuracy is reported. For each trial, the latent item scores θ is sampled uniformly from the simplex in (6), which determines the true ranking π^* . Given θ , the comparison results are generated according to the Bradley-Terry-Luce model (3). We compare several different methods, including the proposed AKG, random sampling (uniformly random sampling), distance-based sampling, adaptive polling (Peiffer et al., 2012) and rank centrality with uniform sampling or knowledge gradient sampling (Negahban et al., 2012). The details of the methods are provided as follows.

1. **AKG** (see Algorithm 1): We set the prior of θ to be the uniform distribution on the simplex (i.e., α^0 is set to be an all-one vector).

2. **Random Sampling:** The random sampling algorithm is similar to Algorithm 1 in terms of the posterior approximation (by moment matching) and rank inference (by sorting the approximated posterior parameters α^t) after receiving each label. The only difference is that this algorithm replaces Step 2 of Algorithm 1 by a random sampling policy, which selects (i_t, j_t) randomly at each stage. We also choose the uniform distribution on the simplex as the prior.

3. **Distance-Based Sampling:** This algorithm is also the same as Algorithm 1 in terms of the posterior approximation. However, in the sampling phase, this algorithm simply selects the pair of items (i_t, j_t) with the closest posterior parameters α_i^t and α_j^t . We choose the uniform distribution on the simplex as the prior.

4. **Adaptive Polling:** This is a greedy policy proposed by Peiffer et al. (2012), which chooses the pair of items to maximize the KL-divergence between the posterior and prior. The initial $K \times K$ matrix M used in adaptive polling is set to 0 on the diagonal and 0.15 everywhere else.

5. **Rank Centrality:** This is a static rank aggregation algorithm recently proposed by Negahban et al. (2012). We combine it with both the random sampling policy and the knowledge gradient policy. Specifically, for **Centrality + RS**, we randomly select a pair of items at each stage and infer the true ranking using rank centrality. For **Centrality + KG**, we select the next pair of items using AKG policy, but estimate the ranking using rank centrality.

It is worthwhile to point out that we are able to compute the optimal policy exactly only up to the 4-item case, which is not interesting from the ranking perspective and thus is left out from the experiment.

As we can see from Figure 1, the AKG policy has higher accuracy than other methods at all budget levels. Note that the average accuracy of AKG surpasses the level of 70% with only 20 pairs in the case of 10 items. In general, random sampling has similar performance as AKG at the beginning, but eventually AKG will outperform random sampling as it will spend more budget on the ambiguous pairs. This will be verified in the next experiment. Meanwhile, if we combine rank centrality with knowledge gradient sampling, the performance of the algorithm can be boosted significantly. Furthermore, the curves of ranking accuracy of AKG are in general monotonically increasing and have fewer ‘‘bumps’’ than other algorithms. This implies that the sequence of posterior parameters α^t is quite stable when the budget level becomes larger. We also note that due to the high computational cost of adaptive polling, it takes extremely long time when the number of items is 100 and thus we omit its performance in Figure 1b.

It is worthwhile to note that AKG runs significantly faster than the adaptive polling method. It enjoys the advantage of closed-form updating rule during each iteration/stage without using a numerical algorithm as a subroutine, which is a good feature for online applications. In contrast, adaptive polling is much slower because it requires inverting a $K \times K$ matrix for all $O(K^2)$ possible pairs and all possible comparison results in each iteration. Table 1 gives the computation time of a *single iteration* for both AKG and adaptive polling. Note that in the 25-item case, the computation time for adaptive polling of a single iteration has already exceeded 40 minutes. Therefore, we omit to present the computation time of adaptive polling when the number of items is 100 in Table 1 since each iteration/stage would take hours to run.

Next, we study the allocation of labeling budget over pairs of items with different levels of ambiguity when using the AKG policy. Again, we consider two scenarios: $K = 10, T = 100$ and $K = 100, T = 1000$, each with 100 independent trials. We report the averaged labeling frequency of each pair. The results are presented in Figure 2 in the form of heat maps.

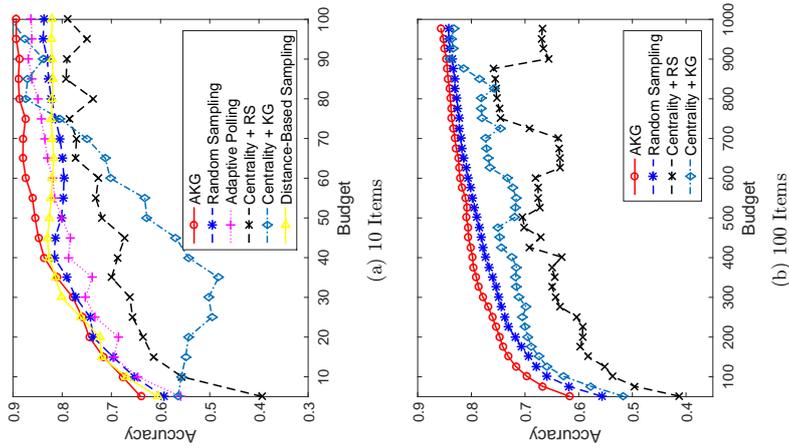


Figure 1: Performance comparison under the homogeneous workers setting. The x -axis is the budget level and y -axis is the averaged ranking accuracy.

Table 1: Comparison in computation time under the homogeneous workers setting.

No. of Items	AKG	Adaptive Polling
10	0.023 sec	20 sec
25	0.75 sec	42 min
100	22 sec	-

In Figure 2, each small block represents a pair of items. Items are sorted based on their true latent scores, from lowest to highest along both y -axis and x -axis, so that the item pairs along the back-diagonal are more ambiguous than those around the corner. Figure

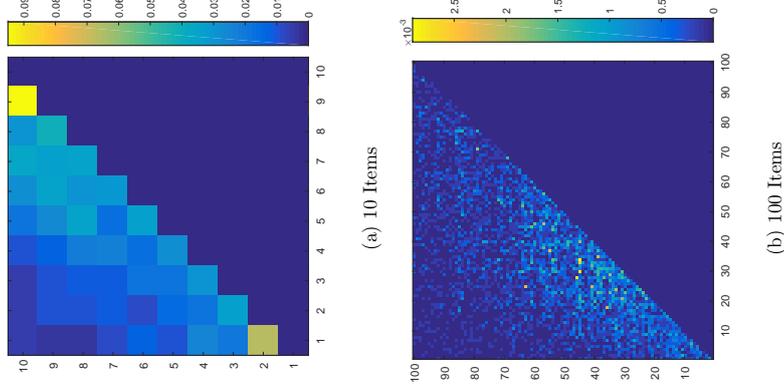


Figure 2: Heat map of labeling frequency for item pairs with different levels of ambiguity

2 presents the normalized number of comparisons over different pairs in total T stages. It can be seen from Figure 2 that the back-diagonal pairs in general have higher labeling frequency than other pairs. Some adjacent pairs are labeled 10 times more frequently than the distant pairs. To further demonstrate this property, we design a scenario in which out of 10 items, the two best items and the two worst items have very close true scores respectively. Although the main goal of the algorithm is to achieve higher ranking accuracy, we are still curious to see whether our policy can spend the budget on these two pairs. As we can see from Figure 3, it is clear that the algorithm concentrates on the 1-2 pair and the 9-10 pair. This implies that our policy can identify and explore more ambiguous pairs to improve the learning of the true ranks.

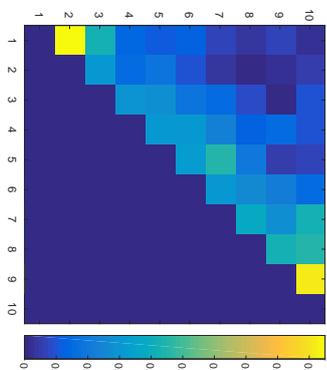
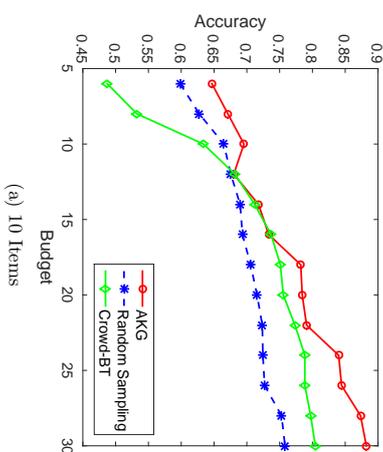


Figure 3: Heat map of labeling frequency for pairs with very close scores

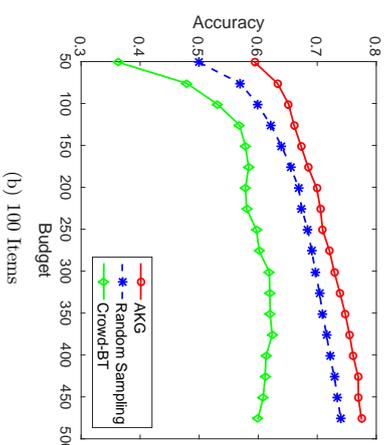
5.2 Simulated Study under the Heterogeneous Workers Setting

In this section we bring worker quality ρ_w into consideration, which is assumed to be drawn from the Beta(4,1) distribution. We choose the Beta(4,1) to generate ρ_w since the average reliability measure of workers in this case is $4/5 = 80\%$. This assumption is in line with the practice in that there are usually more reliable workers than unreliable ones. Similar to the homogeneous worker setting, we consider two scenarios: 10 items with 10 heterogeneous workers ($K = 10, M = 10$); 100 items with 50 heterogeneous workers ($K = 100, M = 50$) and we note that each worker is allowed to label any pair at most once. We compare the following three methods.

1. **AKG** (see Algorithm 2): We set the prior of θ to be the uniform distribution on the simplex (i.e., α^0 is set to be an all-one vector) and choose $\mu_w^0 = 4, \nu_w^0 = 1$ for each worker $w = 1, 2, \dots, M$.
2. **Random Sampling**: It is implemented simply by replacing Step 2 of Algorithm 2 by a random sampling policy, which selects a triplet $\{item\ i, item\ j, worker\ w\}$ uniformly randomly at each stage. The choices of priors are the same as in AKG. Like the AKG method, the random sampling algorithm also maintains a Dirichlet distribution for the scores of items and a beta distribution for the reliability parameter of each worker using moment matching.
3. **Crowd-BT**: This is an adaptive algorithm recently proposed by Chen et al. (2013), which chooses the triplet $\{item\ i, item\ j, worker\ w\}$ at each iteration to maximize the information gain. This can be viewed as an extension of the adaptive polling (Peiffer et al., 2012) by incorporating the workers' reliability. Unlike adaptive polling which computes the relative entropy for each pair exactly, Crowd-BT uses moment matching to approximate the posterior and hence runs significantly faster than adaptive polling. The parameter γ , which balances the exploitation-exploration trade-off in Chen et al. (2013), is set to 1 in this experiment.



(a) 10 Items



(b) 100 Items

Figure 4: Performance comparison under the heterogeneous workers setting. The x -axis is the budget level and y -axis is the averaged ranking accuracy.

The comparison results are presented in Figure 4, where AKG outperforms the other two methods, especially when the budget level is low. The performance of random sampling is comparable to AKG at the beginning. As we gather more information, AKG can learn the reliability of workers so that the budget will be gradually shifted towards those reliable workers (as shown later in Figure 7). In fact, it can be seen from Figure 4 that the ranking accuracy of AKG increases more quickly than that of other methods. In this experiment, even if there is a small amount of budget (e.g. $T = K$), the AKG policy is still able to achieve reasonably good performance. We notice that in the 100-item case Crowd-BT is beaten by random sampling. The main reason is that when the reliability of workers varies and the pool is large, it is difficult to balance exploration and exploitation for Crowd-BT,

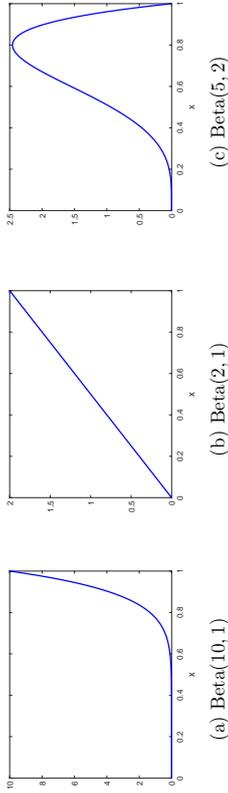


Figure 5: Density plots of different Beta distributions for generating ρ_w

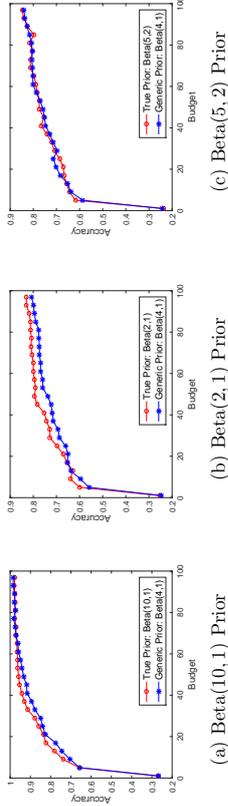


Figure 6: Comparisons between AKG using Beta(4,1) prior and AKG using the true generating distribution as prior.

Table 2: Computation time under the heterogeneous workers setting.

No. of Items	No. of Workers	AKG
10	10	0.038 sec
25	20	0.82 sec
100	50	41 sec

which has already been acknowledged in Chen et al. (2013). Similar to the previous setting, we also give the table of the computation time of a *single iteration* for AKG in Table 2. As we can see from the table, even with another dimension of uncertainty — the reliability of workers, AKG is still quite fast, and thus is suitable for online implementation.

In order to investigate how sensitive the prior for workers' reliability ρ_w is, we generate the workers' true reliability parameters from three different distributions, Beta(10, 1), Beta(2, 1), and Beta(5, 2), and compare the performances of AKG between using the true generating distribution as the prior and using the generic Beta(4, 1) as the prior. The results are plotted in Figure 6. As one can see from Figure 6, using the true generating distribution and generic Beta(4, 1) prior lead to very similar performance in all three cases. Although there are some small differences between the two groups of curves, they are not significant as to the overall performance of the algorithm. This result shows that when there is no

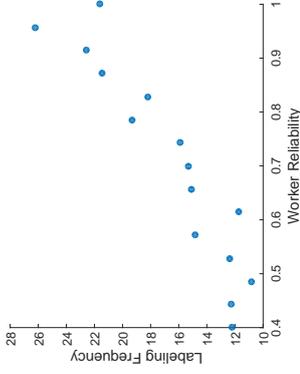


Figure 7: Averaged number of comparisons (a.k.a., labeling frequency) made by workers with different levels of reliability ρ_w .

exact information on the quality of all workers, Beta(4, 1) is a reasonable prior for workers' reliability and the proposed AKG policy is quite robust to the prior distribution in use.

Finally, we investigate whether good workers are indeed assigned more comparison tasks by our AKG policy in the setting of heterogeneous workers. In particular, we consider $k = 10$ items and $M = 15$ workers with the workers' true reliability parameters $\rho_w, w = 1, 2, \dots, M$ ranging from 0.4 to 1 with an equal space in between. This crowd of workers is fixed and the total budget in each trial $T = 250$. We report the averaged number of pairs assigned to workers with different levels of reliability in Figure 7. As one can see from Figure 7, there is a clear trend that more reliable workers receive more pairs on average.

5.3 Real Data Study

We now apply the proposed AKG policy (Algorithm 2) to a real dataset on reading difficulty levels (Collins-Thompson and Callan, 2004). The dataset comprises $K = 491$ different paragraphs, each assigned an integer-valued true reading difficulty score ranging from 1, 2, ..., 12. Here, a higher score means the paragraph is more difficult to read. A total number of $M = 217$ different workers from Canada and the United States performed the comparison tasks on an online crowdsourcing platform called CrowdFlower³. Each worker was presented a pair of paragraphs every time and the worker identified which paragraph is more difficult to read. To overcome the issue of an imbalanced judgemental pool, each worker was allowed to compare at most 40 different pairs. There are 7,898 pairwise comparison results available in this dataset. Using these pairwise labels, we apply the AKG policy to recover the ranking by difficulty of these 491 paragraphs. We note that since the underlying truth is given as a difficulty level (1–12) for each paragraph (denoted by s_i for $i = 1, \dots, K$) instead of a global ranking, we measure the accuracy of a ranking π as

$$\frac{2}{K(K-1)} \sum_{i \neq j} \mathbf{1}_{(\pi(i) > \pi(j))} \mathbf{1}_{(s_i \geq s_j)}.$$

3. <http://www.crowdflower.com/>

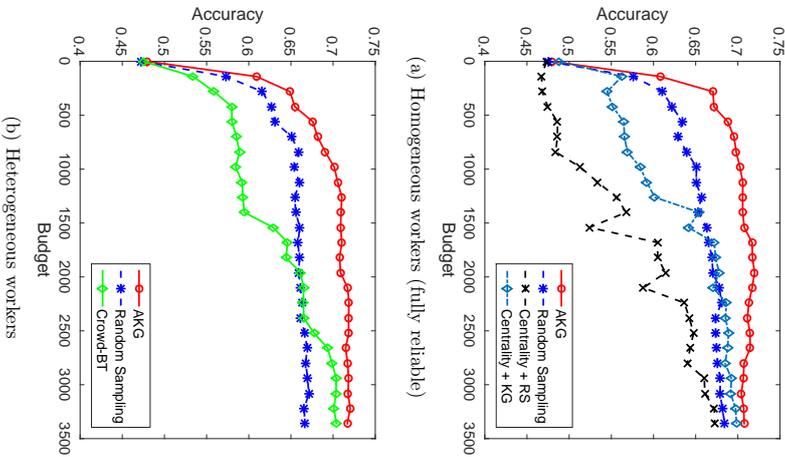


Figure 8: Performance comparison on the real dataset

In the above definition of ranking accuracy, when two paragraphs have the same reading difficulty level, any ranking between this pair will be treated as correct. It is also worth noting that, in the knowledge gradient step in (59), it is possible that the selected triplet (i_t, j_t, w_t) does not exist in the dataset (i.e., the worker w_t did not compare i_t and j_t in this data). Hence, in our implementation of AKG, we select the triplet in the dataset that maximizes the right-hand side of (59). We set the prior of θ to be the uniform distribution on the simplex. This dataset also comes with a rating for each worker which measures the long-run performance of this worker on CrowdFlower. A higher rating implies a higher reliability of the worker. This dataset shows the averaged workers' rating is above 0.75. Thus, we still use Beta(4,1) as the prior on workers' reliability.

We run experiments in two different settings. The first one assumes that all workers are homogeneous and fully reliable. In this setting, we only need to select the next pair of paragraphs to compare but can randomly choose a worker to perform the comparison task. In this case, four algorithms are implemented (AKG policy (Algorithm 1), random sampling, rank centrality with the random sampling policy, and rank centrality with the knowledge gradient policy) and we report the averaged accuracy over 100 independent trials in Figure 8a to minimize the sampling effect of randomly selecting the next worker. The second experiment incorporates the heterogeneous reliability of workers so that the algorithms have to select both the pair to compare and the worker to perform the comparison task. In this case, three algorithms, AKG policy (Algorithm 2), random sampling and Crowd-BT, are implemented and the result is shown in Figure 8b. As one can see from these two plots, AKG outperforms the other methods in both settings, especially when the amount of budget is relatively low. As the budget level increases, the performance of Crowd-BT and rank centrality will eventually improve and achieve a similar accuracy as AKG.

6. Conclusion

In this paper, we address the dynamic budget allocation problem in crowdsourced ranking. Using the Kendall's tau with respect to the true ranking as the measure of ranking accuracy, we formulate the problem of maximizing expected Kendall's tau by sequential comparisons into a Bayesian Markov decision process. To further address the computational challenges (especially, solving the NP-hard MAX-TOP) involved in the decision process, we propose an approximated knowledge gradient policy, which is not only computationally efficient but also achieves good performance as shown in the experimental sections.

We note that although this paper focuses on the Bradley-Terry-Luce model (Bradley and Terry, 1952; Luce, 1959), it will be interesting to study the dynamic sampling in crowdsourced ranking for other ranking models such as permutation-based models (e.g., Mallows (Mallows, 1957) and CPS (Qin et al., 2010) models) or stochastically transitive models (Fishburn, 1973; Shah et al., 2016b)). Meanwhile, theoretical bounds on posterior approximation errors are difficult to obtain and error propagation does exist during each iteration of the algorithm. In our future analysis we would like to quantify this error. Another interesting future direction is to incorporate the feature information of each item into the probabilistic model of the pairwise comparison results and develop a dynamic sampling policy that can further improve the ranking accuracy via modeling the feature information.

Acknowledgments

Xi Chen would like to acknowledge support for this project from the Google Faculty Research Award.

Appendix

In this section, we provide detailed proofs of some propositions in the paper.

Proof (of Proposition 1)

We will only show that (18) and (19) can be represented as (20) when $Y_{ij} = 1$. The proof for $Y_{ij} = -1$ is similar.

It is known that $\mathbb{E}[\theta_k|\boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha}')] = \frac{\alpha'_k}{\alpha'_0}$ and $\mathbb{E}[\theta_k^2|\boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha}')] = \frac{\alpha'_k(\alpha'_k+1)}{\alpha'_0(\alpha'_0+1)}$ for $k = 1, 2, \dots, K$, which characterize the left-hand sides of (18) and (19).

With elementary calculus, we can show

$$\Pr(Y_{ij} = 1|\boldsymbol{\alpha}) = \int_{\Delta} \frac{\theta_i}{\theta_i + \theta_j} \frac{1}{\text{B}(\boldsymbol{\alpha})} \prod_{k=1}^K \theta_k^{\alpha_k-1} d\boldsymbol{\theta} = \frac{\alpha_i}{\alpha_i + \alpha_j} \quad (62)$$

so that

$$p(\boldsymbol{\theta}|Y_{ij} = 1, \boldsymbol{\alpha}) = \frac{p(\boldsymbol{\theta}, Y_{ij} = 1|\boldsymbol{\alpha})}{\Pr(Y_{ij} = 1|\boldsymbol{\alpha})} = \frac{\alpha_i + \alpha_j}{\alpha_i} \frac{\theta_i}{\theta_i + \theta_j} \frac{1}{\text{B}(\boldsymbol{\alpha})} \prod_{k=1}^K \theta_k^{\alpha_k-1}. \quad (63)$$

Let $\boldsymbol{\beta} = (\beta_1, \dots, \beta_K)$ with $\beta_i = \alpha_i + 1$ and $\beta_k = \alpha_k$ for $k \neq i$. Then, we can show that

$$\begin{aligned} \mathbb{E}[\theta_i|Y_{ij} = 1, \boldsymbol{\alpha}] &= \frac{\alpha_i + \alpha_j}{\alpha_i} \left[\int_{\Delta} \frac{\theta_i^2}{\theta_i + \theta_j} \frac{1}{\text{B}(\boldsymbol{\alpha})} \prod_{k=1}^K \theta_k^{\alpha_k-1} d\boldsymbol{\theta} \right] \\ &= \frac{\alpha_i + \alpha_j}{\alpha_0} \left[\int_{\Delta} \frac{\theta_i}{\theta_i + \theta_j} \frac{1}{\text{B}(\boldsymbol{\beta})} \prod_{k=1}^K \theta_k^{\beta_k-1} d\boldsymbol{\theta} \right] \\ &= \frac{\alpha_i + 1}{\alpha_0} \frac{\alpha_i + \alpha_j}{\alpha_i + \alpha_j + 1}, \end{aligned} \quad (64)$$

where the first and the third equalities are due to (62) and (63) and the second equality is by the definition of $\boldsymbol{\beta}$ and the property $\Gamma(x+1) = x\Gamma(x)$ of Gamma function. Using a similar argument, we can show that

$$\mathbb{E}[\theta_j|Y_{ij} = 1, \boldsymbol{\alpha}] = \frac{\alpha_j}{\alpha_0} \frac{\alpha_i + \alpha_j}{\alpha_i + \alpha_j + 1} \quad (65)$$

$$\mathbb{E}[\theta_k|Y_{ij} = 1, \boldsymbol{\alpha}] = \frac{\alpha_k}{\alpha_0} \quad \text{for } k \neq i, j \quad (66)$$

$$\mathbb{E}[\theta_i^2|Y_{ij} = 1, \boldsymbol{\alpha}] = \frac{\alpha_i + 1}{\alpha_0} \frac{\alpha_i + 2}{\alpha_0 + 1} \frac{\alpha_i + \alpha_j}{\alpha_i + \alpha_j + 2} \quad (67)$$

$$\mathbb{E}[\theta_j^2|Y_{ij} = 1, \boldsymbol{\alpha}] = \frac{\alpha_j}{\alpha_0} \frac{\alpha_j + 1}{\alpha_0 + 1} \frac{\alpha_i + \alpha_j}{\alpha_i + \alpha_j + 2} \quad (68)$$

$$\mathbb{E}[\theta_k^2|Y_{ij} = 1, \boldsymbol{\alpha}] = \frac{\alpha_k}{\alpha_0} \frac{\alpha_k + 1}{\alpha_0 + 1} \quad \text{for } k \neq i, j. \quad (69)$$

Note that, when $Y_{ij} = 1$, the right-hand sides of (18) and (19) can be represented as the right-hand side of (20) using (64)~(69) \blacksquare

Proof (of Proposition 3)

We will only show the conclusion when $Y_{ij}^w = 1$. The proof for $Y_{ij}^w = -1$ is similar.

When $Y_{ij}^w = 1$, we have $\eta_{ijw} = \frac{\mu_w \alpha_i}{\mu_w \alpha_i + \nu_w \alpha_j}$. We will first show (44) and (45) can be represented as the first four equations in (48). Since $\mathbb{E}[\theta_k|\boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha}')] = \frac{\alpha'_k}{\alpha'_0}$ and $\mathbb{E}[\theta_k^2|\boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha}')] = \frac{\alpha'_k(\alpha'_k+1)}{\alpha'_0(\alpha'_0+1)}$ for $k = 1, 2, \dots, K$, the left-hand sides of the first four equations in (48) and those of (44) and (45) are identical.

With (62) and some basic properties of the Beta distribution, we can show

$$\begin{aligned} \Pr(Y_{ij}^w = 1|\boldsymbol{\alpha}, \mu_w, \nu_w) &= \mathbb{E} \left[\frac{\rho_w}{\theta_i + \theta_j} + (1 - \rho_w) \frac{\theta_j}{\theta_i + \theta_j} \middle| \boldsymbol{\theta} \sim \text{Dir}(\boldsymbol{\alpha}), \rho_w \sim \text{Beta}(\mu_w, \nu_w) \right] \\ &= \frac{\mu_w}{\mu_w + \nu_w} \frac{\alpha_i}{\alpha_i + \alpha_j} + \frac{\nu_w}{\mu_w + \nu_w} \frac{\alpha_j}{\alpha_i + \alpha_j} \end{aligned} \quad (70)$$

so that

$$\begin{aligned} p(\boldsymbol{\theta}, \rho_w|Y_{ij}^w = 1, \boldsymbol{\alpha}, \mu_w, \nu_w) &= \frac{\theta_i}{\left(\frac{\rho_w}{\theta_i + \theta_j} + (1 - \rho_w) \frac{\theta_j}{\theta_i + \theta_j} \right)} \frac{1}{\text{B}(\boldsymbol{\alpha}) \text{B}(\mu_w, \nu_w)} \prod_{k=1}^K \theta_k^{\alpha_k-1} \rho_w^{\mu_w-1} (1 - \rho_w)^{\nu_w-1} \\ &= \frac{\frac{\mu_w}{\mu_w + \nu_w} \frac{\alpha_i}{\alpha_i + \alpha_j} + \frac{\nu_w}{\mu_w + \nu_w} \frac{\alpha_j}{\alpha_i + \alpha_j}}{\frac{\mu_w}{\mu_w + \nu_w} \frac{\alpha_i}{\alpha_i + \alpha_j} + (1 - \rho_w) \frac{\theta_j}{\theta_i + \theta_j}} \frac{1}{\text{B}(\boldsymbol{\alpha}) \text{B}(\mu_w, \nu_w)} \prod_{k=1}^K \theta_k^{\alpha_k-1} \rho_w^{\mu_w-1} (1 - \rho_w)^{\nu_w-1}. \end{aligned} \quad (71)$$

The equations (70) and (71), together with (64), imply

$$\begin{aligned} \mathbb{E}[\theta_i|Y_{ij}^w = 1, \boldsymbol{\alpha}, \mu_w, \nu_w] &= \frac{\int_0^1 \int_{\Delta} \frac{\theta_i^2}{\left(\frac{\rho_w}{\theta_i + \theta_j} + (1 - \rho_w) \frac{\theta_j}{\theta_i + \theta_j} \right)} \frac{1}{\text{B}(\boldsymbol{\alpha}) \text{B}(\mu_w, \nu_w)} \prod_{k=1}^K \theta_k^{\alpha_k-1} \rho_w^{\mu_w-1} (1 - \rho_w)^{\nu_w-1} d\boldsymbol{\theta} d\rho_w}{\int_{\Delta} \left(\frac{\mu_w}{\mu_w + \nu_w} \frac{\theta_i^2}{\theta_i + \theta_j} + \frac{\nu_w}{\mu_w + \nu_w} \frac{\theta_j}{\theta_i + \theta_j} \right) \frac{1}{\text{B}(\boldsymbol{\alpha})} \prod_{k=1}^K \theta_k^{\alpha_k-1} d\boldsymbol{\theta}} \\ &= \frac{\frac{\mu_w}{\mu_w + \nu_w} \frac{\alpha_i}{\alpha_i + \alpha_j} + \frac{\nu_w}{\mu_w + \nu_w} \frac{\alpha_j}{\alpha_i + \alpha_j}}{\frac{\mu_w}{\mu_w + \nu_w} \frac{\alpha_i}{\alpha_i + \alpha_j} + \frac{\nu_w}{\mu_w + \nu_w} \frac{\alpha_j}{\alpha_i + \alpha_j}} \frac{\frac{\mu_w}{\mu_w + \nu_w} \frac{\alpha_i}{\alpha_i + \alpha_j}}{\frac{\mu_w}{\mu_w + \nu_w} \frac{\alpha_i}{\alpha_i + \alpha_j} + \frac{\nu_w}{\mu_w + \nu_w} \frac{\alpha_j}{\alpha_i + \alpha_j}} \\ &= \frac{\frac{\mu_w}{\mu_w + \nu_w} \frac{\alpha_i}{\alpha_i + \alpha_j} + \frac{\nu_w}{\mu_w + \nu_w} \frac{\alpha_j}{\alpha_i + \alpha_j}}{\frac{\mu_w}{\mu_w + \nu_w} \frac{\alpha_i}{\alpha_i + \alpha_j} + \frac{\nu_w}{\mu_w + \nu_w} \frac{\alpha_j}{\alpha_i + \alpha_j}} \frac{\frac{\mu_w}{\mu_w + \nu_w} \frac{\alpha_i}{\alpha_i + \alpha_j}}{\frac{\mu_w}{\mu_w + \nu_w} \frac{\alpha_i}{\alpha_i + \alpha_j} + \frac{\nu_w}{\mu_w + \nu_w} \frac{\alpha_j}{\alpha_i + \alpha_j}} \\ &= \eta_{ijw} \frac{\alpha_0(\alpha_i + \alpha_j + 1)}{\alpha_0(\alpha_i + \alpha_j) + 1} + (1 - \eta_{ijw}) \frac{\alpha_i(\alpha_i + \alpha_j)}{\alpha_0(\alpha_i + \alpha_j) + 1}. \end{aligned} \quad (72)$$

Using a similar argument, we can show that

$$\mathbb{E}[\theta_j^w | Y_{ij}^{vw} = 1, \alpha, \mu_w, \nu_w] = \eta_{ij}^{vw} \frac{\alpha_j(\alpha_i + \alpha_j)}{\alpha_0(\alpha_i + \alpha_j + 1)} + (1 - \eta_{ij}^{vw}) \frac{(\alpha_j + 1)(\alpha_i + \alpha_j)}{\alpha_0(\alpha_i + \alpha_j + 1)} \quad (73)$$

$$\mathbb{E}[\theta_k | Y_{ij}^{vw} = 1, \alpha, \mu_w, \nu_w] = \frac{\alpha_k}{\alpha_0} \quad \text{for } k \neq i, j \quad (74)$$

$$\mathbb{E}[\theta_i^w | Y_{ij}^{vw} = 1, \alpha, \mu_w, \nu_w] = \frac{\eta_{ij}^{vw}(\alpha_i + 1)(\alpha_i + \alpha_j)}{\alpha_0(\alpha_0 + 1)(\alpha_i + \alpha_j + 2)} + \frac{(1 - \eta_{ij}^{vw})\alpha_i(\alpha_i + 1)(\alpha_i + \alpha_j)}{\alpha_0(\alpha_0 + 1)(\alpha_i + \alpha_j + 2)} \quad (75)$$

$$\mathbb{E}[\theta_j^w | Y_{ij}^{vw} = 1, \alpha, \mu_w, \nu_w] = \frac{\eta_{ij}^{vw}\alpha_j(\alpha_j + 1)(\alpha_i + \alpha_j)}{\alpha_0(\alpha_0 + 1)(\alpha_i + \alpha_j + 2)} + \frac{(1 - \eta_{ij}^{vw})(\alpha_j + 1)(\alpha_i + \alpha_j + 2)(\alpha_i + \alpha_j)}{\alpha_0(\alpha_0 + 1)(\alpha_i + \alpha_j + 2)} \quad (76)$$

$$\mathbb{E}[\theta_k^w | Y_{ij}^{vw} = 1, \alpha, \mu_w, \nu_w] = \frac{\alpha_k}{\alpha_0} \frac{\alpha_k + 1}{\alpha_0 + 1} \quad \text{for } k \neq i, j. \quad (77)$$

In the next, we will show (46) and (47) can be represented as the last two equations in (48). When $Y_{ij}^{vw} = 1$, the last two equations in (48) become

$$\begin{cases} \frac{\mu_w}{\eta_{ij}^{vw} \mu_w + \nu_w + 1} + (1 - \eta_{ij}^{vw}) \frac{\mu_w}{\mu_w + \nu_w + 1} = \eta_{ij}^{vw} k \frac{(\mu_w + 1)(\mu_w + 2)}{(\mu_w + \nu_w + 1)(\mu_w + \nu_w + 2)} + (1 - \eta_{ij}^{vw}) \frac{(\mu_w)(\mu_w + 1)}{(\nu_w + 1)(\nu_w + 2)} \\ \frac{\mu_w}{(\mu_w + \nu_w + 1)(\nu_w + 1)} + (1 - \eta_{ij}^{vw}) \frac{\mu_w}{(\mu_w + \nu_w + 1)(\nu_w + 2)} = \eta_{ij}^{vw} k \frac{(\mu_w + 1)(\mu_w + 2)}{(\mu_w + \nu_w + 1)(\mu_w + \nu_w + 2)} + (1 - \eta_{ij}^{vw}) \frac{(\mu_w + 1)(\nu_w + 2)}{(\nu_w + 1)(\nu_w + 2)}. \end{cases} \quad (78)$$

It is known that $\mathbb{E}[\rho_w | \rho_w \sim \text{Beta}(\mu_w', \nu_w')] = \frac{\mu_w'}{\mu_w' + \nu_w'}$, $\mathbb{E}[\rho_w^2 | \rho_w \sim \text{Beta}(\mu_w', \nu_w')] = \frac{\mu_w'(\mu_w' + 1)}{(\mu_w' + \nu_w')(\mu_w' + \nu_w' + 1)}$ and $\mathbb{E}[(1 - \rho_w)^2 | \rho_w \sim \text{Beta}(\mu_w', \nu_w')] = \frac{\nu_w'(\nu_w' + 1)}{(\mu_w' + \nu_w')(\mu_w' + \nu_w' + 1)}$, indicating that the left-hand sides of (46) and (47) match those of (78).

To characterize the right-hand sides of (46) and (47), we first derive from (71) that

$$\begin{aligned} & \mathbb{E}[\rho_w | Y_{ij}^{vw}, \alpha, \mu, \nu] \\ &= \int_0^1 \int \Delta \left(\frac{\theta_i}{\theta_i + \theta_j} + \rho_w(1 - \rho_w) \frac{\theta_j}{\theta_i + \theta_j} \right) \frac{1}{\text{B}(\alpha)\text{B}(\mu_w, \nu_w)} \prod_{k=1}^K \theta_k^{\alpha_k - 1} \rho_w^{\alpha_w - 1} (1 - \rho_w)^{\nu_w - 1} d\theta d\rho_w \\ &= \frac{\int_0^1 \frac{\mu_w}{\mu_w + \nu_w} \frac{\alpha_i}{\alpha_i + \alpha_j} + \frac{\nu_w}{\mu_w + \nu_w} \frac{\alpha_j}{\alpha_i + \alpha_j}}{\frac{\mu_w}{\mu_w + \nu_w} \frac{\alpha_i}{\alpha_i + \alpha_j} + \frac{\nu_w}{\mu_w + \nu_w} \frac{\alpha_j}{\alpha_i + \alpha_j}} \\ &= \frac{\int_0^1 \left(\frac{\alpha_i}{\alpha_i + \alpha_j} \rho_w^2 + \frac{\alpha_j}{\alpha_i + \alpha_j} \rho_w(1 - \rho_w) \right) \frac{1}{\text{B}(\mu_w, \nu_w)} \rho_w^{\mu_w - 1} (1 - \rho_w)^{\nu_w - 1} d\rho_w}{\frac{\mu_w}{\mu_w + \nu_w} \frac{\alpha_i}{\alpha_i + \alpha_j} + \frac{\nu_w}{\mu_w + \nu_w} \frac{\alpha_j}{\alpha_i + \alpha_j}} \\ &= \frac{\frac{\mu_w}{\mu_w + \nu_w} \frac{\mu_w + 1}{\mu_w + \nu_w + 1} \frac{\alpha_i}{\alpha_i + \alpha_j}}{\frac{\mu_w}{\mu_w + \nu_w} \frac{\alpha_i}{\alpha_i + \alpha_j} + \frac{\nu_w}{\mu_w + \nu_w} \frac{\alpha_j}{\alpha_i + \alpha_j}} + \frac{\frac{\mu_w}{\mu_w + \nu_w} \frac{\nu_w}{\mu_w + \nu_w + 1} \frac{\alpha_j}{\alpha_i + \alpha_j}}{\frac{\mu_w}{\mu_w + \nu_w} \frac{\alpha_i}{\alpha_i + \alpha_j} + \frac{\nu_w}{\mu_w + \nu_w} \frac{\alpha_j}{\alpha_i + \alpha_j}} \\ &= \frac{\eta_{ij}^{vw}}{\mu_w + \nu_w + 1} + (1 - \eta_{ij}^{vw}) \frac{\mu_w}{\mu_w + \nu_w + 1}. \end{aligned} \quad (79)$$

Following a similar procedure, we can show

$$\begin{aligned} & \mathbb{E}[\rho_w^2 + (1 - \rho_w)^2 | \alpha_i \succ_w \alpha_j, \theta \sim \text{Dir}(\alpha), \rho_w \sim \text{Beta}(\mu_w, \nu_w)] \\ &= \frac{\frac{\mu_w}{\mu_w + \nu_w} \frac{\mu_w + 1}{\mu_w + \nu_w + 1} \frac{\mu_w + 2}{\mu_w + \nu_w + 2} \frac{\alpha_i}{\alpha_i + \alpha_j} + \frac{\frac{\nu_w}{\mu_w + \nu_w} \frac{\nu_w + 1}{\mu_w + \nu_w + 1} \frac{\nu_w}{\mu_w + \nu_w + 2} \frac{\alpha_i + \alpha_j}{\alpha_i}}{\frac{\mu_w}{\mu_w + \nu_w} \frac{\alpha_i}{\alpha_i + \alpha_j} + \frac{\nu_w}{\mu_w + \nu_w} \frac{\alpha_j}{\alpha_i + \alpha_j}} + \frac{\frac{\mu_w}{\mu_w + \nu_w} \frac{\mu_w + 1}{\mu_w + \nu_w + 1} \frac{\nu_w}{\mu_w + \nu_w + 2} \frac{\alpha_i + \alpha_j}{\alpha_i}}{\frac{\mu_w}{\mu_w + \nu_w} \frac{\alpha_i}{\alpha_i + \alpha_j} + \frac{\nu_w}{\mu_w + \nu_w} \frac{\alpha_j}{\alpha_i + \alpha_j}} \\ &+ \frac{\frac{\nu_w}{\mu_w + \nu_w} \frac{\nu_w + 1}{\mu_w + \nu_w + 1} \frac{\mu_w}{\mu_w + \nu_w + 2} \frac{\alpha_i + \alpha_j}{\alpha_i}}{\frac{\mu_w}{\mu_w + \nu_w} \frac{\alpha_i}{\alpha_i + \alpha_j} + \frac{\nu_w}{\mu_w + \nu_w} \frac{\alpha_j}{\alpha_i + \alpha_j}} + \frac{\frac{\mu_w}{\mu_w + \nu_w} \frac{\mu_w + 1}{\mu_w + \nu_w + 1} \frac{\mu_w + \nu_w + 2}{\mu_w + \nu_w + 2} \frac{\alpha_i + \alpha_j}{\alpha_i}}{\frac{\mu_w}{\mu_w + \nu_w} \frac{\alpha_i}{\alpha_i + \alpha_j} + \frac{\nu_w}{\mu_w + \nu_w} \frac{\alpha_j}{\alpha_i + \alpha_j}} \\ &= \frac{\eta_{ij}^{vw}}{(\mu_w + 1)(\mu_w + 2)} + (1 - \eta_{ij}^{vw}) \frac{(\mu_w)(\mu_w + 1)}{(\mu_w + 1)(\nu_w + 2)} \\ &+ \eta_{ij}^{vw} \frac{(\mu_w + \nu_w + 1)(\mu_w + \nu_w + 2)}{(\nu_w)(\nu_w + 1)} + (1 - \eta_{ij}^{vw}) \frac{(\mu_w + \nu_w + 1)(\mu_w + \nu_w + 2)}{(\nu_w + 1)(\nu_w + 2)}. \end{aligned} \quad (80)$$

Putting (79) and (80) together, we have shown that the right-hand sides of (78) are exactly the right-hand sides of (70) and (71), which completes the proof. ■

References

- S. Acharyya. *Learning to rank in supervised and unsupervised settings using convexity and monotonicity*. PhD thesis, Electrical and Computer Engineering, The University of Texas at Austin, 2013.
- N. Allou. An active learning algorithm for ranking from pairwise preferences with an almost optimal query complexity. *Journal of Machine Learning Research*, 13(1):137–164, 2012.
- Y. Bachrach, T. Minka, J. Guiver, and T. Graepel. How to grade a test without knowing the answers - a Bayesian graphical model for adaptive crowdsourcing and aptitude testing. In *International Conference on Machine Learning (ICML)*, 2012.
- M. J. Beal. *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, Gatsby Computational Neuroscience Unit, University College London, 2003.
- R. A. Bradley and M. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324345, 1952.
- M. Braverman and E. Mossel. Noisy sorting without resampling. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2008.
- C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *International Conference on Machine Learning (ICML)*, 2005.
- Y. Cao, J. Xu, T.-Y. Liu, H. Li, Y. Huang, and H.-W. Hon. Adapting ranking svm to document retrieval. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006.

- Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In *International Conference on Machine Learning (ICML)*, 2007.
- X. Chen, P. N. Bennett, K. Collins-Thompson, and E. Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *ACM International Conference on Web Search and Data Mining (WSDM)*, 2013.
- X. Chen, Q. Lin, and D. Zhou. Statistical decision making for optimal budget allocation in crowd labelling. *Journal of Machine Learning Research*, 16:1–46, 2015.
- K. Collins-Thompson and J. Callan. A language modeling approach to predicting reading difficulty. In *HLT*, 2004.
- W. S. Cooper, F. C. Gey, and D. P. Dabney. Probabilistic retrieval based on staged logistic regression. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1992.
- K. Crammer and Y. Singer. Pranking with ranking. In *Advances in Neural Information Processing Systems (NIPS)*, 2001.
- A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society Series C*, 28:20–28, 1979.
- S. Ertekin, H. Hirsh, and C. Rudin. Wisely using a budget for crowdsourcing. Technical report, MIT, 2012.
- P. C. Fishburn. Binary choice probabilities: on the varieties of stochastic transitivity. *Journal of Mathematical Psychology*, 10(4):327 – 352, 1973.
- P. Frazier. *Knowledge-Gradient Methods for Statistical Learning*. PhD thesis, Princeton University, 2009.
- P. Frazier, W. B. Powell, and S. Dayanik. A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, 47(5):2410–2439, 2008.
- Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4(11):933–969, 2003.
- C. Gao and D. Zhou. Minimax optimal convergence rates for estimating ground truth from crowdsourced labels. arXiv:1310.5764, 2013.
- D. Gleich and L. h. Lim. Rank aggregation via nuclear norm minimization. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2011.
- M. Grötschel, M. Jtinger, and G. Reinelt. A cutting plane algorithm for the linear ordering problem. *Operations Research*, 32(6):1195–1220, 1984.
- S. S. Gupta and K. J. Miescke. Bayesian look ahead one-stage sampling allocations for selection of the best population. *Journal of Statistical Planning and Inference*, 54(2): 229–244, 1996.
- R. Herbrich, T. Minka, and T. Graepel. Trueskill (TM): a bayesian skill rating system. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.
- C. Ho, S. Jabbari, and J. W. Vaughan. Adaptive task assignment for crowdsourced classification. In *International Conference on Machine Learning (ICML)*, 2013.
- J. Howe. The rise of crowdsourcing. *Wired*, 2006.
- K. G. Jamieson and R. Nowak. Active ranking using pairwise comparisons. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.
- E. Kamar, S. Hacker, and E. Horvitz. Combining human and machine intelligence in large-scale crowdsourcing. In *International Conference on Autonomous Agents and Multiagent System*, 2012.
- D. Karger, S. Oh, and D. Shah. Budget-optimal task allocation for reliable crowdsourcing systems. *Operations Research*, 62(1):1–24, 2013a.
- D. R. Karger, S. Oh, and D. Shah. Efficient crowdsourcing for multi-class labeling. *ACM SIGMETRICS Performance Evaluation Review*, 41(1):81–92, 2013b.
- M. Kendall. A new measure of rank correlation. *Biometrika*, 30:81–89, 1938.
- J.-W. Kuo, P.-J. Cheng, and H.-M. Wang. Learning to rank from Bayesian decision inference. In *ACM Conference on Information and Knowledge Management*, 2009.
- P. Li, C. Burges, and Q. Wu. Learning to rank using classification and gradient boosting. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.
- Q. Liu, J. Peng, and A. Ihler. Variational inference for crowdsourcing. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.
- T. Liu. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*, 3:225–331, 2009.
- T. Lu and C. Boutilier. Effective sampling and learning for mallows models with pairwise preference data. *Journal of Machine Learning Research*, 15(1):3783–3829, 2014.
- R. Luce. *Individual choice behavior: a theoretical analysis*. Wiley, 1959.
- C. L. Mallows. Non-null ranking models. *Biometrika*, 44:114–130, 1957.
- S. Mishra and K. Sikdar. On approximability of linear ordering and related np-optimization problems on graphs. *Discrete Applied Mathematics*, 136(2–3):249–269, 2004.
- S. Negalban, S. Oh, and D. Shah. Rank centrality: ranking from pair-wise comparisons. arXiv:1209.1688, 2012.
- J. Paisley, D. Blei, and M. Jordan. Variational bayesian inference with stochastic search. In *International Conference on Machine Learning (ICML)*, 2012.

- T. Pfeiffer, X. A. Gao, Y. Chen, A. Mao, and D. G. Rand. Adaptive polling for information aggregation. In *AAAI Conference on Artificial Intelligence*, 2012.
- W. B. Powell. *The Knowledge Gradient for Optimal Learning*. Wiley Encyclopedia for Operations Research and Management Science, 2010.
- M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, 2005.
- L. Qian, J. Gao, and H. V. Jagadish. Learning user preferences by adaptive pairwise comparison. *Proceedings of the VLDB Endowment*, 8(11):1322–1333, 2015.
- T. Qin, X. Geng, and T. Y. Lin. A new probabilistic model for rank aggregation. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.
- K. Radnisky and N. Ailon. Ranking from pairs and triplets: Information quality, evaluation methods and query complexity. In *ACM International Conference on Web Search and Data Mining*, 2011.
- A. Rajkumar and S. Agarwal. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In *International Conference on Machine Learning (ICML)*, 2014.
- V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(4):1297–1322, 2010.
- I. O. Ryzhov, W. B. Powell, and P. I. Frazier. The knowledge gradient algorithm for a general class of online learning problems. *Operations Research*, 60(1):180–195, 2012.
- N. B. Shah, S. Balakrishnan, J. Bradley, A. Parekh, K. Ramchandran, and M. J. Wainwright. Estimation from pairwise comparisons: Sharp minimax bounds with topology dependence. *Journal of Machine Learning Research*, 17, 2016a.
- N. B. Shah, S. Balakrishnan, A. Guntuboyina, and M. J. Wainwright. Stochastically transitive models for pairwise comparisons: Statistical and computational issues. In *International Conference on Machine Learning (ICML)*, 2016b.
- M. Taylor, J. Guiver, S. Robertson, and T. Minka. Softrank: Optimising non-smooth rank metrics. In *ACM International Conference on Web Search and Data Mining (WSDM)*, 2008.
- L. L. Thurstone. The method of paired comparisons for social values. *Journal of Abnormal and Social Psychology*, 21:384–400, 1927.
- M. N. Volkovs and R. S. Zemel. New learning methods for supervised and unsupervised preference aggregation. *Journal of Machine Learning Research*, 15(1):1135–1176, 2014.
- F. Wauthier, M. Jordan, and N. Jojic. Efficient ranking from pairwise comparisons. In *International Conference on Machine Learning (ICML)*, 2013.
- P. Weïnder, S. Branson, S. Belongie, and P. Perona. The multidimensional wisdom of crowds. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.
- J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. R. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.
- J. Wu and P. I. Frazier. The parallel knowledge gradient method for batch bayesian optimization. arXiv:1606.04414, 2016.
- J. Xu and H. Li. AdRank: A boosting algorithm for information retrieval. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2007.
- J. Yi, R. Jin, S. Jain, and A. K. Jain. Inferring users’ preferences from crowdsourced pairwise comparisons: A matrix completion approach. In *Conference on Human Computation and Crowdsourcing (HCOMP)*, 2013.
- Y. Zhang, X. Chen, D. Zhou, and M. I. Jordan. Spectral methods meet em: A provably optimal algorithm for crowdsourcing. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.
- Z. Zheng, H. Zha, T. Zhang, O. Chapelle, K. Chen, and G. Sun. A general boosting method and its application to learning ranking functions for web search. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

Multi-scale Classification using Localized Spatial Depth

Subhajit Dutta

*Department of Mathematics and Statistics
Indian Institute of Technology
Kanpur 208016, India.*

DUTTAS@IITK.AC.IN

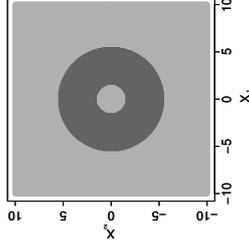
Soham Sarkar

*Theoretical Statistics and Mathematics Unit
Indian Statistical Institute
203, B. T. Road, Kolkata 700108, India.*

SOHAMSARKAR1991@GMAIL.COM

AKGHOSH@ISICAL.AC.IN

(a) Example **E1**



(b) Example **E2**

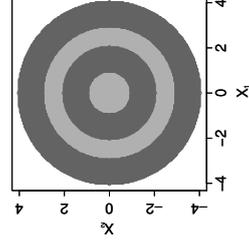


Figure 1: Bayes class boundaries in \mathbb{R}^2 .

Editor: Jie Peng

Abstract

In this article, we develop and investigate a new classifier based on features extracted using spatial depth. Our construction is based on fitting a generalized additive model to posterior probabilities of different competing classes. To cope with possible multi-modal as well as non-elliptic nature of the population distribution, we also develop a localized version of spatial depth and use that with varying degrees of localization to build the classifier. Final classification is done by aggregating several posterior probability estimates, each of which is obtained using this localized spatial depth with a fixed scale of localization. The proposed classifier can be conveniently used even when the dimension of the data is larger than the sample size, and its good discriminatory power for such data has been established using theoretical as well as numerical results.

Keywords: Bayes classifier, elliptic distributions, generalized additive models, HDLSS asymptotics, uniform strong consistency, weighted aggregation of posteriors.

1. Introduction

In a supervised classification problem with J competing classes, we have n_j labeled observations $\mathbf{x}_{j1}, \dots, \mathbf{x}_{jn_j}$ from the j -th class ($1 \leq j \leq J$). We use this training sample consisting of $n = \sum_{j=1}^J n_j$ observations to construct a decision rule for classifying an unlabeled observation \mathbf{x} to one of these J classes. If π_j , f_j and $p(j|\cdot)$ denote the prior probability, the probability density function and the posterior probability of the j -th class, respectively, then the *Bayes classifier* assigns \mathbf{x} to the class j_0 , where $j_0 = \operatorname{argmax}_{1 \leq j \leq J} p(j|\mathbf{x}) = \operatorname{argmax}_{1 \leq j \leq J} \pi_j f_j(\mathbf{x})$. However, the f_j 's or the $p(j|\cdot)$'s are usually unknown in practice, and one needs to estimate them from the training sample. Popular parametric classifiers like linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA) (see, e.g., Hastie et al., 2009) are motivated by parametric model assumptions on the f_j 's. So, they may lead to poor classification when these assumptions fail to hold, and the class boundaries of the Bayes classifier have complex geometry. On the other hand, nonparametric classifiers like those based on k -nearest neighbors (k -NN) (see, e.g., Cover and Hart, 1967) and kernel

density estimates (KDE) (see, e.g., Scott, 2015) are more flexible and free from such model assumptions. But, they suffer from the curse of dimensionality and are often not suitable for high-dimensional data.

To demonstrate this, let us consider two examples denoted by **E1** and **E2**. **E1** involves a classification problem with two classes in \mathbb{R}^d , where the distribution of the first class is an equal mixture of $N_d(\mathbf{0}_d, \mathbf{I}_d)$ and $N_d(\mathbf{0}_d, 10\mathbf{I}_d)$, and that of the second class is $N_d(\mathbf{0}_d, 5\mathbf{I}_d)$. Here N_d denotes the d -variate normal distribution, $\mathbf{0}_d = (0, \dots, 0)^T \in \mathbb{R}^d$ and \mathbf{I}_d is the $d \times d$ identity matrix. In **E2**, each class distribution is an equal mixture of two uniform distributions. For the first (respectively, the second) class, it is a mixture of $U_d(0, 1)$ and $U_d(2, 3)$ (respectively, $U_d(1, 2)$ and $U_d(3, 4)$), where $U_d(r_1, r_2)$ denotes the uniform distribution over the region $\{\mathbf{x} \in \mathbb{R}^d : r_1 \leq \|\mathbf{x}\| \leq r_2\}$ with $0 \leq r_1 < r_2 < \infty$ and $\|\cdot\|$ being the Euclidean norm. Figure 1 shows the class boundaries of the Bayes classifier for these two examples when $d = 2$ and $\pi_1 = \pi_2 = 1/2$. The regions colored grey (respectively, black) correspond to observations classified to the first (respectively, the second) class by the Bayes classifier. It is clear that classifiers like LDA and QDA, or any other classifier with linear or quadratic class boundaries will deviate significantly from the Bayes classifier in both examples. A natural question then is how standard nonparametric classifiers like those based on k -NN and KDE perform in such examples.

Figure 2 shows the average misclassification rates of these two classifiers along with the Bayes risks for different values of d . These classifiers were trained on a sample of size 100 from each class, and the misclassification rates were computed based on 250 independent observations from each class. This procedure was repeated 500 times to calculate the average misclassification rates. Smoothing parameters associated with k -NN and KDE (i.e., the number of neighbors k in k -NN and the bandwidth in KDE) were chosen by minimizing leave-one-out cross-validation estimates of misclassification rates (see, e.g., Hastie et al., 2009). Figure 2 shows that in **E1**, the Bayes risk decreases to zero as d grows. Since the class distributions in **E2** have disjoint supports, the Bayes risk is zero for all values of d . But in both examples, the misclassification rates of these two nonparametric classifiers increased to almost 50% as d increased.

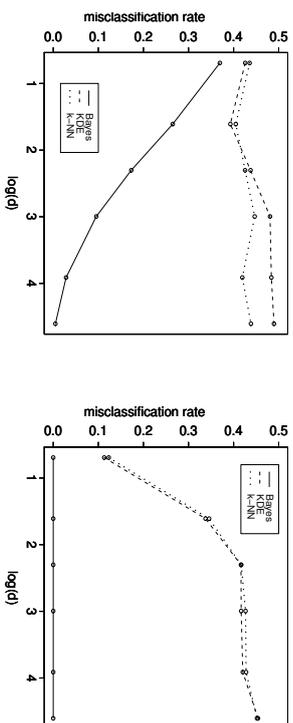


Figure 2: Average misclassification rates of nonparametric classifiers and the Bayes classifier for $d = 2, 5, 10, 20, 50$ and 100 .

These two examples clearly show the necessity to develop new classifiers to cope with such situations. We use the idea of data depth for this purpose. Over the last three decades, data depth (see, e.g., Liu et al., 1999; Zuo and Serfling, 2000) has emerged as a powerful tool for multivariate data analysis with applications in many areas including supervised and unsupervised classification (see, e.g., Jornsten, 2004; Ghosh and Chaudhuri, 2005a,b; Hoberg and Mosler, 2006; Xia et al., 2008; Dutta and Ghosh, 2012; Li et al., 2012; Lange et al., 2014; Paillardaveine and Van Bever, 2015). Spatial depth (also known as the L_1 depth) is a popular notion of data depth that was introduced and studied by Vardi and Zhang (2000) and Serfling (2002). The *spatial depth* (SPD) of an observation $\mathbf{x} \in \mathbb{R}^d$ with respect to (w.r.t.) a distribution function F on \mathbb{R}^d is defined as $\text{SPD}(\mathbf{x}, F) = 1 - \frac{\|E_F[u(\mathbf{x} - \mathbf{X})]\|}{\|\mathbf{x}\|}$, where $\mathbf{X} \sim F$, and $u(\cdot)$ is the multivariate sign function given by $u(\mathbf{x}) = \|\mathbf{x}\|^{-1}\mathbf{x}$ if $\mathbf{x} \neq \mathbf{0}_d \in \mathbb{R}^d$, and $u(\mathbf{0}_d) = \mathbf{0}_d$. This version of SPD is invariant w.r.t. location shift, orthogonal, and homogeneous scale transformations. SPD is often computed on the standardized version of \mathbf{X} as well. In that case, it is defined as

$$\text{SPD}(\mathbf{x}, F) = 1 - \frac{\|E_F[u(\Sigma^{-1/2}(\mathbf{x} - \mathbf{X}))]\|}{\|\mathbf{x}\|},$$

where Σ is a scatter matrix associated with F . One can check that if Σ has the affine equivariance property (see, e.g., Zuo and Serfling, 2000), this version of SPD is affine invariant. To differentiate between these two versions of SPD, we will denote them by SPD° and SPD^* , respectively. If $\Sigma = \lambda \mathbf{I}_d$ for some $\lambda > 0$ (e.g., if F is spherically symmetric, see Pang et al., 1990), then SPD° and SPD^* coincide. Throughout this article, the term SPD will be used in a generic sense.

Like other depth functions, SPD provides a center-outward ordering of multivariate data. An observation has higher (respectively, lower) depth if it lies close to (respectively, away from) the center of the distribution. In other words, given an observation \mathbf{x} and a

pair of probability distributions F_1 and F_2 , if $\text{SPD}(\mathbf{x}, F_1)$ is larger than $\text{SPD}(\mathbf{x}, F_2)$, one would expect \mathbf{x} to come from F_1 instead of F_2 . Based on this simple idea, the *maximum depth classifier* was developed by Jornsten (2004); Ghosh and Chaudhuri (2005b). For a J class problem involving distributions F_1, \dots, F_J , the maximum depth classifier based on SPD assigns an observation \mathbf{x} to the j_0 -th class, where $j_0 = \text{argmax}_{1 \leq j \leq J} \text{SPD}(\mathbf{x}, F_j)$.

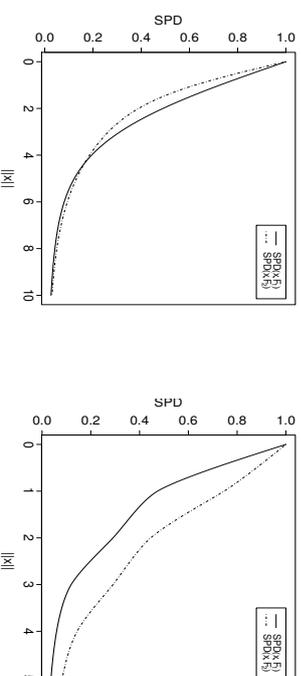


Figure 3: $\text{SPD}(\mathbf{x}, F_1)$ and $\text{SPD}(\mathbf{x}, F_2)$ for different values of $\|\mathbf{x}\|$ when $\mathbf{x} \in \mathbb{R}^2$.

In **E1** and **E2**, since the class distributions are spherically symmetric, SPD^* coincides with SPD° , and they become a monotonically decreasing function of the Euclidean norm of \mathbf{x} (see Lemma 7). In Figure 3, we have plotted $\text{SPD}(\mathbf{x}, F_1)$ and $\text{SPD}(\mathbf{x}, F_2)$ for different values of $\|\mathbf{x}\|$ in **E1** and **E2**, where F_1 and F_2 are the distributions of the two classes and $\mathbf{x} \in \mathbb{R}^2$. It is transparent from Figure 3 that the maximum depth classifier based on SPD will fail in both examples. In **E1**, for all values of $\|\mathbf{x}\|$ smaller (respectively, greater) than a constant close to 4, the observations will be classified to the first (respectively, the second) class by the maximum SPD classifier. On the other hand, this classifier will classify all observations to the second class in **E2**. Most of the popular depth functions turn out to be monotonically decreasing functions of the Euclidean norm in the case of a spherically symmetric distribution. So, the maximum depth classifiers based on those depth functions will have similar problems as well.

In Section 2, we develop a modified classifier based on SPD to overcome this limitation of maximum depth classifiers. In the literature, most of the modified depth based classifiers are developed mainly for two class problems (see, e.g., Ghosh and Chaudhuri, 2005b; Dutta and Ghosh, 2012; Li et al., 2012; Lange et al., 2014). For classification problems involving $J (> 2)$ classes, one usually solves $\binom{J}{2}$ binary classification problems taking one pair of classes at a time and then uses either majority voting (see, e.g., Friedman, 1996) or pairwise coupling (see, e.g., Hastie and Tibshirani, 1998) to make the final classification. Unlike those existing methods, our proposed classifier directly addresses the J class problem.

Almost all existing depth based classifiers require ellipticity of class distributions to achieve Bayes optimality. To cope with possible multi-modal as well as non-elliptic population distributions, we construct a localized version of spatial depth (LSPD) in Section 3. In Section 4, we develop a multi-scale classifier based on LSPD. Relevant theoretical results on SPD, LSPD and the resulting classifiers are studied in these sections. In Sections 5 and 6, some simulated and benchmark data sets are analyzed to demonstrate the usefulness of these proposed classifiers. An advantage of SPD over other depth functions is its computational simplicity. Classifiers based on SPD and LSPD can be constructed even when the dimension exceeds the sample size. We deal with such high dimension, low sample size (HDLSS) cases in Section 7, and show that both classifiers turn out to be optimal under a fairly general framework. Several high-dimensional data sets are also analyzed to evaluate their empirical performance. All proofs and mathematical details are given in Appendix A.

2. Bayes Optimality of a Classifier Based on Spatial Depth

Let us assume that f_1, \dots, f_J are density functions of J elliptically symmetric distributions (Fang et al., 1990) on \mathbb{R}^d , where $f_j(\mathbf{x}) = |\Sigma_j|^{-1/2} g_j(|\Sigma_j^{-1/2}(\mathbf{x} - \boldsymbol{\mu}_j)|)$ for $1 \leq j \leq J$. Here $\boldsymbol{\mu}_j \in \mathbb{R}^d$, Σ_j is a $d \times d$ symmetric and positive definite matrix, and $g_j(|\mathbf{t}|)$ is a probability density function of a spherically symmetric distribution on \mathbb{R}^d for $1 \leq j \leq J$. For such classification problems involving general elliptic populations with equal or unequal priors, the next theorem establishes the Bayes optimality of a classifier, which is based on $\mathbf{z}^*(\mathbf{x}) = (z_1^*(\mathbf{x}), \dots, z_J^*(\mathbf{x}))^T = (\text{SPD}^*(\mathbf{x}, F_1), \dots, \text{SPD}^*(\mathbf{x}, F_J))^T$.

Theorem 1 *If the densities of J competing classes are elliptically symmetric, the posterior probabilities of these classes satisfy the logistic regression model given by*

$$p(j|\mathbf{x}) = \tilde{p}(j|\mathbf{z}^*(\mathbf{x})) = \frac{\exp(\Phi_j(\mathbf{z}^*(\mathbf{x})))}{[1 + \sum_{k=1}^{(J-1)} \exp(\Phi_k(\mathbf{z}^*(\mathbf{x})))]} \text{ for } 1 \leq j \leq (J-1) \quad (1)$$

$$\text{and } p(J|\mathbf{x}) = \tilde{p}(J|\mathbf{z}^*(\mathbf{x})) = \frac{1}{[1 + \sum_{k=1}^{(J-1)} \exp(\Phi_k(\mathbf{z}^*(\mathbf{x})))]} \quad (2)$$

Here $\Phi_j(\mathbf{z}^*(\mathbf{x})) = \varphi_{j1}(z_1^*(\mathbf{x})) + \dots + \varphi_{jJ}(z_J^*(\mathbf{x}))$, and $\varphi_{j\cdot}$ s are appropriate real-valued functions of π_j and f_j for $1 \leq j \leq J$. Consequently, the Bayes rule assigns an observation \mathbf{x} to the class j_0 , where $j_0 = \arg\max_{1 \leq j \leq J} \tilde{p}(j|\mathbf{z}^*(\mathbf{x}))$.

Theorem 1 shows that the Bayes classifier is based on a nonparametric multinomial additive logistic regression model for the posterior probabilities, which is a special case of generalized additive models (GAM) (Hastie and Tibshirani, 1990). If the prior probabilities of J classes are equal, and f_1, \dots, f_J are all elliptic and unimodal differing only in their locations, this Bayes classifier reduces to the maximum depth classifier (Ghosh and Chaudhuri, 2005b) (see Remark 8 after the proof of Theorem 1 in Appendix A). A special case of Theorem 1 with $\Sigma_j = \lambda_j \mathbf{I}_d$, where $\lambda_j > 0$ for $1 \leq j \leq J$ is stated below.

Corollary 2 *If the densities of J competing classes are spherically symmetric (i.e., $f_j(\mathbf{x}) = g_j(\|\mathbf{x} - \boldsymbol{\mu}_j\|)$ for $1 \leq j \leq J$), then the posterior probabilities of these classes satisfy the logistic regression model given in Theorem 1 with $\mathbf{z}^*(\mathbf{x})$ replaced by $\mathbf{z}^\circ(\mathbf{x}) = (\text{SPD}^\circ(\mathbf{x}, F_1), \dots, \text{SPD}^\circ(\mathbf{x}, F_J))^T$.*

For any fixed i and j , one can calculate the J -dimensional vector $\mathbf{z}^\circ(\mathbf{x}_{ji})$ (or, $\mathbf{z}^*(\mathbf{x}_{ji})$), where \mathbf{x}_{ji} is the i -th labeled observation from the j -th class for $1 \leq i \leq n_j$ and $1 \leq j \leq J$. These $\mathbf{z}^\circ(\mathbf{x}_{ji})$ s (or, $\mathbf{z}^*(\mathbf{x}_{ji})$ s) can be viewed as realizations of the vector of covariates in a non-parametric multinomial additive logistic regression model, where the response corresponds to the class label that belongs to $\{1, \dots, J\}$. Now, a classifier based on SPD can be constructed by fitting a GAM with the logistic link function. This procedure can be viewed as a multinomial logistic regression in the J -dimensional depth plot. Lange et al. (2014); Li et al. (2012); Mozharovskiy et al. (2015) used such plots for nonparametric classification. Recently, Cuesta-Albertos et al. also considered GAM to construct a depth based classifier for functional data. In practice, we use a random sample $\mathbf{x}_1, \dots, \mathbf{x}_n$ generated from F to compute the empirical versions of SPD° and SPD^* , which are given by

$$\text{SPD}^\circ(\mathbf{x}, F_n) = 1 - \left\| \frac{1}{n} \sum_{i=1}^n u(\mathbf{x} - \mathbf{x}_i) \right\| \quad \text{and} \quad \text{SPD}^*(\mathbf{x}, F_n) = 1 - \left\| \frac{1}{n} \sum_{i=1}^n u(\widehat{\Sigma}^{-1/2}(\mathbf{x} - \mathbf{x}_i)) \right\|,$$

respectively, where $\widehat{\Sigma}$ is an estimate of Σ , and F_n is the empirical distribution of the data $\mathbf{x}_1, \dots, \mathbf{x}_n$. Clearly, SPD^* is affine invariant if $\widehat{\Sigma}$ has the affine equivariance property. The resulting classifier worked quite well in examples E1 and E2, and we shall see the numerical results later in Section 5.1.

3. Extraction of Small Scale Distributional Features by Localization of Spatial Depth

Under elliptic symmetry, the density function of a class can be expressed as a function of SPD^* , and hence the depth contours coincide with the density contours. This is the main mathematical argument used in the proof of Theorem 1. For non-elliptic distributions, where the density function cannot be expressed as a function of SPD, such mathematical arguments are no longer valid. Consider an equal mixture of $N_d(\mathbf{0}_d, 0.25\mathbf{I}_d)$, $N_d(2\mathbf{1}_d, 0.25\mathbf{I}_d)$ and $N_d(4\mathbf{1}_d, 0.25\mathbf{I}_d)$, where $\mathbf{1}_d = (1, \dots, 1)^T$ denotes a d -dimensional vector with all elements equal to 1. We have plotted the density contours in Figure 4(a) and SPD° contours in Figure 4(b) when $d = 2$. In this trimodal distribution, the SPD° contours failed to match the density contours. As a second example, we consider a d -dimensional distribution with independent components, where the i -th component is exponential with the scale parameter $d/(d-i+1)$ for $1 \leq i \leq d$. Figures 5(a) and 5(b) show the density contours and the SPD° contours, respectively, when $d = 2$. Even in this example, SPD° and density contours differed significantly. We observed a similar picture for contours based on SPD^* as well.

To cope with this issue, we suggest a *localization* of SPD. Note that $\text{SPD}^\circ(\mathbf{x}, F) = 1 - \|E_F[u(\mathbf{x} - \mathbf{X})]\|$ is constructed by assigning the same weight to each unit vector $u(\mathbf{x} - \mathbf{X})$ and ignoring the significance of the distance between \mathbf{x} and \mathbf{X} . By introducing a weight function, which takes account of this distance, one can extract important features related to the local geometry of the data. To capture these local features, we use a kernel function $K(\cdot)$ and define

$$\Gamma_h^\circ(\mathbf{x}, F) = E_F[K_h(\mathbf{t})] - \|E_F[K_h(\mathbf{t})u(\mathbf{t})]\|,$$

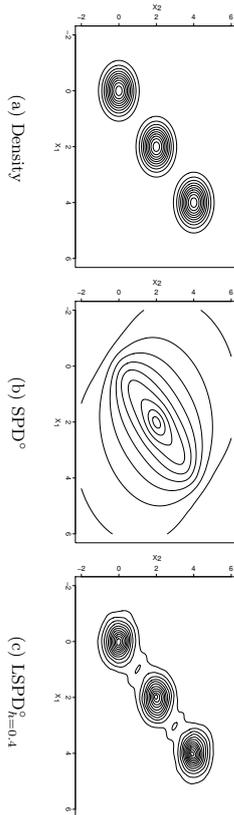


Figure 4: Contours of density; SPD $^\circ$ and LSPD $^\circ_h$ (with $h = 0.4$) functions for a symmetric, trimodal density function.

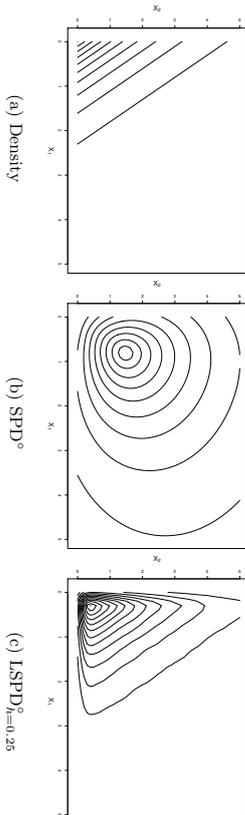


Figure 5: Contours of density; SPD $^\circ$ and LSPD $^\circ_h$ (with $h = 0.25$) functions for the density function $f(x_1, x_2) = 0.5 \exp\{-(x_1 + 0.5x_2)\} I\{x_1 > 0, x_2 > 0\}$.

where $\mathbf{t} = (\mathbf{x} - \mathbf{X})$ and $K_h(\mathbf{t}) = h^{-d}K(\mathbf{t}/h)$. For our theoretical investigation, we will assume K to be a continuous probability density function on \mathbb{R}^d that satisfies the following properties:

- (K1) $K(\mathbf{t}) = g_0(\|\mathbf{t}\|)$, where g_0 is a decreasing function with $g_0(0) < \infty$ and $g_0(\|\mathbf{t}\|) \rightarrow 0$ as $\|\mathbf{t}\| \rightarrow \infty$,
- (K2) $K(\mathbf{t})$ has bounded first derivatives, and
- (K3) $\int_{\mathbb{R}^d} \|t\|K(\mathbf{t})d\mathbf{t} < \infty$.

The Gaussian kernel $K(\mathbf{t}) = (\sqrt{2\pi})^{-d} \exp(-\|\mathbf{t}\|^2/2)$ is a possible choice. It is desirable that localized spatial depth (LSPD) approximates the class density, or a monotone function of it for small values of h . This will ensure that the class densities and hence the class posterior probabilities become functions of LSPD as $h \rightarrow 0$. On the other hand, one should expect that as $h \rightarrow \infty$, LSPD should tend to SPD, or a monotone function of it. However,

$\Gamma_h^\circ(\mathbf{x}, F) \rightarrow 0$ as $h \rightarrow \infty$. So, we re-scale $\Gamma_h^\circ(\mathbf{x}, F)$ by an appropriate factor of h to define LSPD $^\circ$ as follows:

$$\text{LSPD}_h^\circ(\mathbf{x}, F) = \begin{cases} \Gamma_h^\circ(\mathbf{x}, F) & \text{if } h \leq 1, \\ h^d \Gamma_h^\circ(\mathbf{x}, F) & \text{if } h > 1. \end{cases} \quad (3)$$

LSPD $^\circ_h$ defined in this way is a continuous function of h . For $d = 2$, Figures 4(c) and 5(c) show that unlike SPD $^\circ$ contours, LSPD $^\circ_h$ contours matched the density contours in both examples. Using $\mathbf{t} = \Sigma^{-1/2}(\mathbf{x} - \mathbf{X})$ in the definition of $\Gamma_h^\circ(\mathbf{x}, F)$, one gets $\Gamma_h^*(\mathbf{x}, F)$, and LSPD $^\circ_h$ is defined using $\Gamma_h^*(\mathbf{x}, F)$ in the same way. Clearly, LSPD $^\circ_h$ is affine invariant if Σ is affine equivariant. When $\Sigma = \lambda \mathbf{I}_d$, we obtain $\Gamma_h^*(\mathbf{x}, F) = \lambda^{d/2} \Gamma_h^\circ(\mathbf{x}, F)$ with $h' = h\sqrt{\lambda}$, and using this expression, one can derive the relation between LSPD $^\circ_h$ and LSPD $^\circ_{h'}$. The vector $\mathbf{z}_h^*(\mathbf{x}) = (\text{LSPD}_h^*(\mathbf{x}, F_1), \dots, \text{LSPD}_h^*(\mathbf{x}, F_J))^T$ has the desired behavior as shown in Theorem 3.

Theorem 3 *If f_1, \dots, f_J are continuous density functions with bounded first derivatives, and Σ_j is the scatter matrix corresponding to the j -th class ($1 \leq j \leq J$), then*

- (a) $\mathbf{z}_h^*(\mathbf{x}) \rightarrow (\sum_{j=1}^J |f_j(\mathbf{x})|, \dots, \sum_{j=1}^J |f_j(\mathbf{x})|^{1/2} f_j(\mathbf{x}))^T$ as $h \rightarrow 0$, and
- (b) $\mathbf{z}_h^*(\mathbf{x}) \rightarrow (K(\mathbf{0})\text{SPD}^*(\mathbf{x}, F_1), \dots, K(\mathbf{0})\text{SPD}^*(\mathbf{x}, F_J))^T$ as $h \rightarrow \infty$.

Now, we construct a classifier by plugging in LSPD $^\circ_h$ instead of SPD in the GAM framework discussed in equations (1) and (2) of Section 2. Consider the following model for the posterior probabilities:

$$p(j|\mathbf{x}) = \frac{\exp(\Phi_j(\mathbf{z}_h^*(\mathbf{x})))}{[1 + \sum_{k=1}^{(J-1)} \exp(\Phi_k(\mathbf{z}_h^*(\mathbf{x})))]}, \quad \text{for } 1 \leq j \leq (J-1), \quad (4)$$

$$\text{and } p(J|\mathbf{x}) = \frac{1}{[1 + \sum_{k=1}^{(J-1)} \exp(\Phi_k(\mathbf{z}_h^*(\mathbf{x})))]}. \quad (5)$$

The main implication of part (a) of Theorem 3 is that the classifier constructed using GAM and $\mathbf{z}_h^*(\mathbf{x})$ as the covariate tends to the Bayes classifier in a general nonparametric setup as $h \rightarrow 0$. On the other hand, part (b) of Theorem 3 implies that for elliptic class distributions, the same classifier tends to the Bayes classifier when $h \rightarrow \infty$. When we fit a GAM, the unknown functions Φ_j s are estimated nonparametrically. Flexibility of such nonparametric estimates also takes care of the unknown constants $|\Sigma_j|^{1/2}$ for $1 \leq j \leq J$ and $K(\mathbf{0})$ in the expressions of the limiting values of $\mathbf{z}_h^*(\mathbf{x})$ in parts (a) and (b) of Theorem 3, respectively. A special case of Theorem 3 follows by taking $\Sigma_j = \lambda_j \mathbf{I}_d$ with $\lambda_j > 0$ for all $1 \leq j \leq J$.

Corollary 4 *If f_1, \dots, f_J are continuous density functions with bounded first derivatives, then*

- (a) $\mathbf{z}_h^*(\mathbf{x}) = (\text{LSPD}_h^\circ(\mathbf{x}, F_1), \dots, \text{LSPD}_h^\circ(\mathbf{x}, F_J))^T \rightarrow (f_1(\mathbf{x}), \dots, f_J(\mathbf{x}))^T$ as $h \rightarrow 0$, and
- (b) $\mathbf{z}_h^*(\mathbf{x}) \rightarrow (K(\mathbf{0})\text{SPD}^\circ(\mathbf{x}, F_1), \dots, K(\mathbf{0})\text{SPD}^\circ(\mathbf{x}, F_J))^T$ as $h \rightarrow \infty$.

If $\mathbf{x}_1, \dots, \mathbf{x}_n$ is a random sample of size n from F , the empirical version of $\Gamma_h^\circ(\mathbf{x}, F)$ is given by

$$\Gamma_h^\circ(\mathbf{x}, F_n) = \frac{1}{n} \sum_{i=1}^n K_h(\mathbf{t}_i) - \left\| \frac{1}{n} \sum_{i=1}^n K_h(\mathbf{t}_i) u(\mathbf{t}_i) \right\|,$$

where $\mathbf{t}_i = (\mathbf{x} - \mathbf{x}_i)$ for $1 \leq i \leq n$. Then $\text{LSPD}_h^o(\mathbf{x}, F_n)$ is defined using (3) with $\Gamma_h^o(\mathbf{x}, F)$ replaced by $\Gamma_h^o(\mathbf{x}, F_n)$. Similarly, we obtain $\Gamma_h^*(\mathbf{x}, F_n)$ and $\text{LSPD}_h^*(\mathbf{x}, F_n)$ by using $\mathbf{t}_i = \widehat{\Sigma}^{-1/2}(\mathbf{x} - \mathbf{x}_i)$ in the expression stated above. Here $\widehat{\Sigma}$ is an estimate of Σ , and F_n is the empirical distribution of the data $\mathbf{x}_1, \dots, \mathbf{x}_n$.

We know that $\sup_{\mathbf{x} \in \mathbb{R}^d} |\text{LSPD}_h^o(\mathbf{x}, F_n) - \text{LSPD}_h^o(\mathbf{x}, F)|$ goes to 0 almost surely (a.s.) as n goes to infinity (see Gao, 2003). Theorem 5 establishes a similar a.s. uniform convergence of $\text{LSPD}_h^o(\mathbf{x}, F_n)$ to its population counterpart $\text{LSPD}_h^o(\mathbf{x}, F)$ for a fixed value of h .

Theorem 5 *Assume the density corresponding to the distribution function F to be bounded. Then, for any fixed $h > 0$, $\sup_{\mathbf{x} \in \mathbb{R}^d} |\text{LSPD}_h^o(\mathbf{x}, F_n) - \text{LSPD}_h^o(\mathbf{x}, F)| \xrightarrow{a.s.} 0$ as $n \rightarrow \infty$.*

From the proof of Theorem 5 (see Appendix A), it is easy to check that this a.s. uniform convergence also holds when $h \rightarrow \infty$. Under additional moment conditions on F , we obtain this convergence when $h \rightarrow 0$ in such a way that $nh^{2d}/\log n \rightarrow \infty$ as $n \rightarrow \infty$ (see Remarks 9 and 10 after the proof of Theorem 5 in Appendix A).

The fact that LSPD tends to a constant multiple of the probability density function as $h \rightarrow 0$ is a crucial requirement for limiting Bayes optimality of classifiers based on this local depth function. Agostinelli and Romanazzi (2010) proposed localized versions of simplicial depth and half-space depth, but the relationship between the local depth and the probability density function was established only for $d = 1$. A depth function based on inter-point distances was developed by Lok and Lee (2011) to capture multi-modality in a data set. Chen et al. (2009) defined kernelized spatial depth using a reproducing kernel Hilbert space. Hu et al. (2011) also considered a generalized notion of Mahalanobis depth in reproducing kernel Hilbert spaces. However, there is no result connecting them to the probability density function. In fact, the kernelized spatial depth function becomes degenerate at the value $(1 - 1/\sqrt{2})$ as the tuning parameter goes to zero. Consequently, it becomes non-informative for small values of the tuning parameter. It will be appropriate to note here that none of the preceding authors used their proposed depth functions for constructing classifiers.

Recently, Paindaveine and Van Bever (2013, 2015) proposed a notion of local depth and used it for supervised classification along with other applications. Their version of local depth does not relate to the underlying density function either. At this point, one should note that convergence of local depth function to the underlying density function is an advantageous property for classification. However, this may not always be a desirable property for other applications of data depth (see Paindaveine and Van Bever, 2013, for a detailed discussion).

4. Multi-scale Classification using Localized Spatial Depth

When the class distributions are elliptic, part (b) of Theorem 3 implies that LSPD_h with large values of h will lead to good classifiers. These large values may not be appropriate for non-elliptic class distributions, but part (a) of Theorem 3 implies that LSPD_h with small values of h will lead to good classifiers for general nonparametric models for class densities. However, the empirical version of LSPD_h with small h and the resulting classifier may have their statistical limitations for high-dimensional data.

We now consider two examples to demonstrate the above points. The first example (we call it **E3**) involves two multivariate normal distributions $N_d(\mathbf{0}_d, \mathbf{I}_d)$ and $N_d(\mathbf{1}_d, 4\mathbf{I}_d)$. In the second example (we call it **E4**), both the competing classes have trimodal distributions. The first class has the same density as in Figure 4(a) (i.e., an equal mixture of $N_d(\mathbf{0}_d, 0.25\mathbf{I}_d)$, $N_d(2\mathbf{1}_d, 0.25\mathbf{I}_d)$ and $N_d(4\mathbf{1}_d, 0.25\mathbf{I}_d)$), while the second class is an equal mixture of $N_d(\mathbf{1}_d, 0.25\mathbf{I}_d)$, $N_d(3\mathbf{1}_d, 0.25\mathbf{I}_d)$ and $N_d(5\mathbf{1}_d, 0.25\mathbf{I}_d)$. In each of these examples, we considered $d = 5$ and generated a training sample of size 100 from each class. The misclassification rate for the classifier based on LSPD_h^o was computed based on a test sample of size 500 (250 observations from each class). This procedure was repeated 100 times to calculate the average misclassification rates for different values of h . Small values of h extracted local distributional features and yielded low misclassification rates in **E4** (see Figure 6(b)). However, those small values of h led to relatively higher misclassification rates in **E3**, while the underlying global elliptic structure was captured well by the proposed classifier for larger values of h (see Figure 6(a)). This provides a strong motivation for adapting a multi-scale approach in constructing the final classifier so that one can harness the strength of different classifiers corresponding to different scales of localization. One would expect that when aggregated judiciously, the multi-scale classifier will lead to an improved misclassification rate. Usefulness of the multi-scale approach in combining different classifiers has been discussed in the classification literature (see, e.g., Kittler et al., 1998; Dzeroski and Zenko, 2004; Ghosh et al., 2005, 2006).

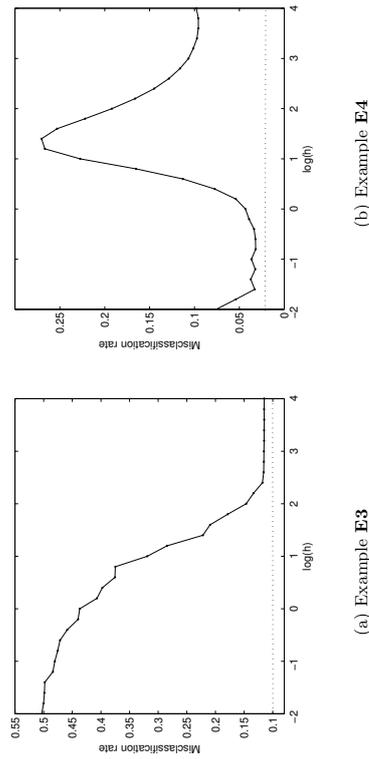


Figure 6: Misclassification rates of the Bayes classifier (indicated by dotted lines) and the classifier based on LSPD_h^o (indicated by solid curves) in examples **E3** and **E4** for varying choices of h .

A popular way of aggregation is to consider a weighted average of the estimated posterior probabilities computed for different values of h . There are various proposals for the choice of the weight function in the literature. Following Ghosh et al. (2005, 2006), we compute $\widehat{\Delta}_h$, a cross-validation estimate of the misclassification rate of the classifier based on LSPD_h^o .

(or, LSPD_h^*) and use

$$W(h) \propto \exp \left[\frac{1}{2} \frac{(\hat{\Delta}_h - \hat{\Delta}_0)^2}{\hat{\Delta}_0(1 - \hat{\Delta}_0)/n} \right]$$

as the weight function, where $\hat{\Delta}_0 = \min_h \hat{\Delta}_h$. The exponential function helps to appropriately weigh up (respectively, weigh down) the promising (respectively, the unsatisfactory) classifier resulting from different choices of the smoothing parameter h . We compute $\int W(h) \bar{g}(h) \bar{p}(j) \mathbf{z}_h^*(\mathbf{x}) dh$ for the j -th class ($1 \leq j \leq J$), where a probability density function \bar{g} is used to make the integral finite. Here $\bar{p}(j) \mathbf{z}_h^*(\mathbf{x})$ is as defined in equations (4) and (5) of Section 3. If we use very small values of h to classify a test case, then the kernel function used in LSPD_h will put almost zero weights on all observations. Clearly, those small values of h will not be useful for classification. On the other hand, LSPD_h behaves like SPD for large values of h . So, after a certain threshold value, increasing the value of h will not provide any additional information about the distributional features. Therefore, one needs to find suitable lower and upper limits of h to compute the weighted posterior probabilities of different classes. Following Ghosh et al. (2006), we compute the pairwise distances (standardized pairwise distances in the case of LSPD_h^*) among the observations in a class and compute the quantiles of these distances. Let $\lambda_{j,\alpha}$ denote the α -th quantile ($0 < \alpha < 1$) of the pairwise distances for the j -th class with $1 \leq j \leq J$. We use $h_L = \min_j \{\lambda_{j,0.05}\} / 3$ as the lower limit of h , and $h_U = 2^r h_L$ as the upper limit of h . Here r is the smallest integer for which we have $\|\mathbf{z}_h^*(\mathbf{x}_{j_1}) - \mathbf{z}^*(\mathbf{x}_{j_1})\| / \|\mathbf{z}^*(\mathbf{x}_{j_1})\| < 0.05$ (or, $\|\mathbf{z}_h^*(\mathbf{x}_{j_1}) - \mathbf{z}^*(\mathbf{x}_{j_1})\| / \|\mathbf{z}^*(\mathbf{x}_{j_1})\| < 0.05$ in case of LSPD_h^*) for $1 \leq i \leq n_j$ and $1 \leq j \leq J$. Our final classifier, which we call the LSPD classifier, assigns an observation \mathbf{x} to the class j_0 , where

$$j_0 = \operatorname{argmax}_{1 \leq j \leq J} \int_{h_L}^{h_U} W(h) \bar{g}(h) \bar{p}(j) \mathbf{z}_h^*(\mathbf{x}) dh.$$

One can choose \bar{g} to be the uniform distribution on the interval $[h_L, h_U]$. Since we are dealing with a scale parameter h , we take the uniform distribution in the logarithmic scale. In practice, we generate M independent observations h_1, \dots, h_M from the distribution \bar{g} . For any given $1 \leq j \leq J$ and \mathbf{x} , $\int_{h_L}^{h_U} W(h) \bar{g}(h) \bar{p}(j) \mathbf{z}_h^*(\mathbf{x}) dh$ is approximated by the average $\sum_{i=1}^M W(h_i) \bar{p}(j) \mathbf{z}_{h_i}^*(\mathbf{x}) / M$.

5. Analysis of Simulated Data Sets

We have analyzed several data sets simulated from elliptic as well as non-elliptic distributions in \mathbb{R}^5 . In each example, taking an equal number of observations from each of the two competing classes, we generated training and test sets of sizes 200 and 500, respectively. This procedure was repeated 500 times, and the average test set misclassification rates of different classifiers are reported in Tables 1 and 2 along with their corresponding standard errors. To facilitate comparison, the corresponding Bayes risks are reported as well. In all the tables in this article, the best misclassification rate in a data set is indicated by ‘*’. The other figures in bold (if any) are the misclassification rates whose differences from the best misclassification rate were found to be statistically insignificant at the 5% level when the usual large sample test for equality of proportions was used.

For the classifiers based on SPD and LSPD , we wrote our own R codes and they are available at the link goo.gl/E5fmld6. Throughout this article, we have used 50 different values of h for multi-scale classification based on LSPD , and the weight function is computed using 5-fold cross-validation method. In this section and in Section 6, we have used SPD^* and LSPD_h^* for classification with the usual sample covariance matrix of the j -th class as $\hat{\Sigma}_j$ for $1 \leq j \leq J$. Any other choice of $\hat{\Sigma}_j$ has been mentioned at appropriate places.

We compared our proposed classifiers with a pool of classifiers that include parametric classifiers like LDA and QDA, and nonparametric classifiers like those based on k -NN (with the Euclidean metric as the distance function) and KDE (with the Gaussian kernel). For k -NN and KDE, we have used the pooled sample covariance matrix for standardization. Tables 1 and 2 show misclassification rates for the multi-scale versions of k -NN (Ghosh et al., 2005) and KDE (Ghosh et al., 2006) based on the same weight function described in Section 4. For the multi-scale method based on KDE, we have considered 50 equi-spaced values of the bandwidth in the range suggested by Ghosh et al. (2006). For the multi-scale version of k -NN, we considered all possible values of k (see Ghosh et al., 2005, for more details). These multi-scale versions usually had better performance than their single scale analogs with the smoothing parameters chosen by the method of cross-validation.

We also considered support vector machines (SVM) (Hastie et al., 2009) based on the linear kernel (i.e., $K(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle$) and the radial basis function (RBF) kernel (i.e., $K_r(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2)$) to facilitate comparison. We used the codes available at the R library `e1071` (Dimitriadou et al., 2011). For the RBF kernel, it has been suggested in the literature to use $\gamma = 1/d$ (see <http://www.csie.ntu.edu.tw/~cjlin1/libsvm/>). However, for our numerical work, we considered $\gamma = i/10d$ for $1 \leq i \leq 50$. We also used 25 different values for the box constraint in the interval $[0.1, 100]$, which were equi-spaced in the logarithmic scale. Misclassification rates were computed for these different choices of the tuning parameters, and the best result is reported in the tables for both classifiers.

Misclassification rates are also reported for classification tree (TREE), and a boosted version of TREE known as random forest (RF) (see, e.g., Hastie et al., 2009). For the implementation of TREE and RF, we used the R codes available in the libraries `tree` (Ripley, 2011) and `randomForest` (Liaw and Wiener, 2002), respectively. For classification tree, the deviance function was used as a measure of impurity, and the maximum height of the tree was restricted to 31. Nodes with less than 5 observations were never considered for splitting. We have combined the results of 500 trees in RF, where each tree was generated based on 63.2% randomly chosen observations from the training sample. At any stage, only a random subset of $\lfloor \sqrt{d} \rfloor$ out of d variables were considered for splitting. Here $\lfloor t \rfloor$ denotes the largest integer less than, or equal to t .

In addition, we also compared the performance of our classifiers with two depth based classification methods: the classifier based on depth-depth (DD) plot (Li et al., 2012) and the maximum depth classifier based on local depth (LD) (Paindaveine and Van Bever, 2013). The DD classifier fits a polynomial on the depth values corresponding to the two competing classes to construct a separating surface. Three notions of depth were used: Mahalanobis depth, half-space depth and projection depth, where the last two depths were computed based on 500 random projections. For each of these depth functions, we used polynomials of degrees 1, 2 and 3. The best result obtained among all these nine possibilities is reported in Tables 1 and 2. For the maximum LD classifier, we used the R library `DepthProc`

(Kosiorowski and Zawadzki, 2016) and considered the best result obtained for different choices of depth and a range of values for the localization parameter. The misclassification rates of the maximum LD classifier was higher than those of the DD classifier in almost all cases, and we do not report those results in this article.

5.1 Examples Involving Elliptic Distributions

Recall examples **E1** and **E2** in Section 2, and example **E3** in Section 4 involving elliptic class distributions. In **E1**, the DD classifier led to the lowest misclassification rate closely followed by SPD and LSPD classifiers (see Table 1), but it did not perform well in **E2**. In this example, SPD and LSPD classifiers significantly outperformed all their competitors. Since the class distributions were elliptic, the SPD classifier had a slight edge over the LSPD classifier in these examples. In view of normality of the class distributions, QDA was expected to have the best performance in **E3**. The DD classifier ranked second here, while SPD and LSPD classifiers performed satisfactorily. In all these examples, the Bayes classifier had non-linear class boundaries. So, LDA and SVM with the linear kernel did not perform well. The performance of SVM with the RBF kernel was relatively better, and it had competitive misclassification rates in **E3**. In all these examples, nonparametric classifiers based on k -NN and KDE yielded much higher misclassification rates compared to SPD and LSPD classifiers.

Table 1: Misclassification rates (in %) of different classifiers in elliptic data sets.

Ex	Bayes risk	LDA	QDA	SVM (linear)	SVM (RBF)	k -NN	KDE	TREE	RF	DD	SPD	LSPD
E1	26.50	50.22 (0.11)	51.58 (0.19)	45.46 (0.12)	33.03 (0.13)	39.99 (0.12)	39.16 (0.12)	36.90 (0.13)	31.32 (0.11)	27.92 * (0.12)	28.32 (0.10)	28.54 (0.11)
E2	0.00	47.43 (0.11)	42.08 (0.12)	43.92 (0.11)	34.06 (0.12)	36.98 (0.13)	34.29 (0.15)	39.10 (0.13)	34.26 (0.11)	26.68 (0.13)	8.23 * (0.11)	8.26 (0.10)
E3	10.14	21.56 (0.09)	11.09 * (0.07)	22.09 (0.09)	11.74 (0.07)	17.86 (0.09)	16.95 (0.08)	19.18 (0.13)	13.77 (0.08)	11.37 (0.08)	11.49 (0.07)	11.64 (0.07)

5.2 Examples Involving Non-elliptic Distributions

We now consider some examples involving non-elliptic class distributions. Recall the trimodal example **E4** discussed in Section 4. In this example, when the classifiers based on k -NN and KDE were used after standardizing the data set by the pooled sample covariance matrix, they yielded misclassification rates higher than 40%. For KDE, we used a common bandwidth in all directions after standardization. This led to the use of a large bandwidth in the principal component direction $\frac{1}{\sqrt{d}}\mathbf{1}_d$ (this can be observed from Figure 4(a)). Since the difference between the posterior probabilities of the two classes changes its sign frequently along this direction, use of this large bandwidth makes it difficult to discriminate between the two competing classes. In the k -NN classifier, this standardization leads to the use of a neighborhood which was also elongated along the direction $\frac{1}{\sqrt{d}}\mathbf{1}_d$, and this affected the performance of this classifier. So, we did not standardize the data for these two classifiers, and they outperformed all other classifiers considered here (see Table 2). Classifiers based on SPD* and LSPD* also had poor performance because of this issue with standard-

ization. So we used classifiers based on SPD° and LSPD° in this example. The LSPD° classifier had the third best performance. SVM with the RBF kernel also performed well. All other classifiers had relatively higher misclassification rates. The DD classifier, LDA, QDA and SVM with the linear kernel all misclassified more than 25% of the observations.

The next example (we call it **E5**) is with exponential distributions, where the component variables are independently distributed in both classes. The i -th variable in the first (respectively, the second) class is exponential with scale parameter $d/(d-i+1)$ (respectively, $d/2i$) for $1 \leq i \leq d$. Further, the second class has a location shift, such that the difference between the mean vectors of the two classes is $\frac{1}{d}\mathbf{1}_d$. Recall that Figure 5(a) shows the density contours of the first class when $d=2$. In this example, the RF classifier had the best performance followed by TREE. Here all the measurement variables were independent, and there was significant separation between the two classes in some of the co-ordinate directions. This is one of the main reasons behind the superior performance of both TREE and RF. Classifiers based on DD, SPD* and LSPD* also performed quite well, and their misclassification rates were significantly lower than all other classifiers. The two linear classifiers performed poorly, but QDA had a reasonably good performance in this example. Good performance of QDA was not surprising as the two competing classes are unimodal, while they differ widely in their dispersion structures.

Table 2: Misclassification rates (in %) of different classifiers in non-elliptic data sets.

Ex	Bayes risk	LDA	QDA	SVM (linear)	SVM (RBF)	k -NN	KDE	TREE	RF	DD	SPD	LSPD
E4	2.10	40.45 (0.12)	42.41 (0.11)	36.16 (0.12)	3.28 (0.04)	2.70 * (0.03)	2.75 (0.03)	15.52 (0.10)	4.98 (0.07)	30.14 (0.12)	10.07 (0.10)	3.25 (0.04)
E5	2.04	41.17 (0.15)	5.97 (0.05)	32.14 (0.34)	7.12 (0.07)	9.55 (0.08)	9.32 (0.07)	4.82 (0.08)	2.04 * (0.03)	5.92 (0.05)	5.53 (0.06)	5.42 (0.06)
E6	13.16	49.67 (0.12)	25.77 (0.12)	47.77 (0.15)	29.33 (0.11)	27.44 (0.11)	27.59 (0.14)	38.39 (0.14)	29.73 (0.11)	28.86 (0.14)	24.15 (0.10)	24.09 * (0.10)
E7	19.96	50.78 (0.23)	50.48 (0.22)	49.77 (0.07)	46.01 (0.23)	35.29 (0.22)	38.88 (0.24)	34.45 (0.13)	27.62 (0.11)	26.48 * (0.12)	38.39 (0.20)	40.64 (0.28)

In example **E6**, each class is an equal mixture of four elliptic distributions. The first class constitutes of $N_d(\mathbf{1}_d, S_{0.6}), t_{3,d}(\beta_d, S_{0.7}), N_d(-\mathbf{1}_d, S_{0.8})$ and $t_{3,d}(-\beta_d, S_{0.9})$, while the second class is an equal mixture of $t_{3,d}(\mathbf{1}_d, S_{-0.9}), N_d(\beta_d, 3S_{-0.8}), t_{3,d}(-\mathbf{1}_d, S_{-0.7})$ and $N_d(-\beta_d, 3S_{-0.6})$. Here $t_{3,d}(\mu, \Sigma)$ denotes the d -variate t distribution with 3 degrees of freedom (df), location parameter μ and scatter matrix Σ . The vector β_d is a d -dimensional vector with the i -th element equal to $(-1)^{i+1}$ for $1 \leq i \leq d$ and the matrix $S_\alpha = ((\alpha^{|i-j|}))_{d \times d}$ for $\alpha \in (-1, 1)$ and $1 \leq i, j \leq d$. This example has a complex structure for the class distributions, and both SPD and LSPD classifiers significantly outperformed all their competitors. As the Bayes classifier was far from being linear, LDA and linear SVM did not have satisfactory performance.

Finally, we consider a classification problem between a Cauchy distribution and a skewed Cauchy distribution (Azzalini, 2014) (we call it **E7**). The Cauchy distribution had location parameter $\mathbf{1}_d$ and scatter matrix $0.5\mathbf{I}_d + 0.5\mathbf{1}_d\mathbf{1}_d^T$; while the skewed Cauchy distribution had location parameter $\mathbf{0}_d$, scatter matrix \mathbf{I}_d and asymmetry vector $\mathbf{1}_d$. The DD classifier and RF performed better than other classifiers, but SPD* and LSPD* classifiers yielded

relatively higher misclassification rates. Both half-space depth and projection depth used in the DD classifier are robust against outliers generated from heavy-tailed distributions, while the moment based estimates used in both SPD* and LSPD* are non-robust. So, it is better to use robust estimates of Σ_j s here. When we used MCD estimates based on 75% of the observations (Rousseeuw and Van Driessen, 1999), the misclassification rates of SPD* and LSPD* classifiers dropped to 31.90% and 32.05%, respectively, with corresponding standard errors of 0.18% and 0.20%.

All these examples clearly demonstrate that the LSPD classifier performs as good as (if not better) popular nonparametric classifiers for non-elliptic, or multi-modal data. This adjustment of the LSPD classifier is automatic in view of the multi-scale approach developed in Section 4.

5.3 Computing Time for SPD and LSPD Classifiers

For a training sample of size n , computation of $\mathbf{z}(\mathbf{x};x_i)$ for $1 \leq i \leq n_j$ and $1 \leq j \leq J$ requires $O(n^2)$ calculations. Fitting a GAM involves an iterative algorithm, and it is quite difficult to calculate its exact computational complexity. Each iteration requires computations of the order $O(n^2)$ (Wood, 2006). So, the algorithm takes no more than $O(n^2)$ computations to fit a GAM for a finite number of iterations. For the multi-scale classifier based on LSPD, we need to repeat this procedure for M different values of h and then compute the weight function $W(h)$ based on V -fold cross-validation. The overall order of computation remains $O(n^2)$ although the associated constant increases linearly with d , J , M and V . However, one should note that these are offline calculations. Both SPD and LSPD classifiers require $O(n)$ calculations to classify a test case.

Throughout this article, we have used $M = 50$ and $V = 5$ and the R library VGAM (Yee, 2008) was used to fit GAM. In a single iteration, the average CPU time to determine the weight function $W(h)$ based on cross-validation for the LSPD classifier was 21.83 seconds, while 0.55 seconds were required to fit a GAM using the full training data. The average CPU time to classify the 500 test observations was about 0.01 seconds. All the calculations were done on a desktop computer with an Intel i7 (2.2 GHz) processor having 8 GB RAM.

6. Analysis of Benchmark Data Sets

We have analyzed seven benchmark data sets for further evaluation of our proposed classifiers. The biomedical data set is taken from the CMU data archive (<http://lib.stat.cmu.edu/datasets/>). In this data set, we ignored the observations with missing values. The diabetes data set is available in the R library mlc1st (also analyzed in Reaven and Miller, 1979). All other data are taken from the UCI machine learning repository (<http://archive.ics.uci.edu/ml/>). Descriptions of these data sets are available at these sources. Satellite image (satimage) data set has specific training and test samples. For this data set, we report misclassification rates of different classifiers based on this fixed test set. If a classifier had misclassification rate ϵ , its standard error was computed as $\sqrt{\epsilon(1-\epsilon)/(\text{size of the test set})}$. For all other data sets, we formed the training and the test sets by randomly partitioning the data, and this random partitioning was repeated 500 times. Average test set misclassification rates of different classifiers were computed over these 500 partitions, and they are reported in Table 3 along with their corresponding stan-

dard errors. Sizes of training and test sets in each partition are also reported in this table. For all classifiers, we used the same tuning procedures as described in Section 5. Codes for the DD classifier are available only for two class problems. In biomedical and Parkinson's data sets, the DD classifier yielded misclassification rates of 12.54% and 14.48%, respectively, with corresponding standard errors of 0.18% and 0.15%. We also used the maximum LD classifier on these real data sets. However, its performance was not satisfactory for most data sets and we do not report those misclassification rates in Table 3.

In biomedical and vehicle data, covariance matrices of the competing classes were different. So, QDA led to significant improvement over LDA, and its misclassification rates were close to the best rate. In both these data sets, the competing classes were nearly elliptic (this can be verified using the diagnostic plots suggested by Li et al., 1997). The SPD classifier utilized this ellipticity of the class distributions to outperform the nonparametric classifiers. The LSPD classifier competed well with the SPD classifier in biomedical data. But, the evidence of ellipticity was much stronger in vehicle data and LSPD had a slightly higher misclassification rate. In diabetes data also, the three competing classes had widely varying covariance structures. As expected, QDA performed better than LDA. Since the class distributions were not elliptic, the SPD classifier yielded a higher misclassification rate than the LSPD classifier, while both TREE and RF outperformed all other classifiers in this data set.

Table 3: Descriptions of the real data sets, and misclassification rates (in %) of different classifiers.

Data set	Biomed	Parkinson's	Diabetes	Wine	Waveform	Vehicle	Satimage
d	4	22	3	13	21	18	36
J	2	2	3	3	3	4	6
Train	100	97	73	100	300	423	4435
Test	94	98	72	78	501	423	2000

Data set	LDA	QDA	SVM (linear)	SVM (RBF)	k-NN	KDE	TREE	RF	SPD	LSPD
Biomed	15.66 (0.14)	12.57 (0.13)	21.90 (0.13)	12.76 (0.13)	17.74 (0.15)	16.67 (0.14)	17.69 (0.18)	13.23 (0.21)	12.53 (0.21)	12.49 * (0.15)
Parkinson's	30.93 (0.12)	xxxx (0.12)	14.83 (0.12)	13.29 (0.10)	14.42 (0.16)	11.24 * (0.12)	16.63 (0.20)	11.68 (0.15)	15.44 (0.15)	14.23 (0.11)
Diabetes	13.86 (0.16)	8.51 (0.13)	10.20 (0.19)	14.93 (0.15)	11.20 (0.13)	11.96 (0.14)	3.78 * (0.09)	4.29 (0.10)	9.36 (0.15)	7.93 (0.14)
Wine	2.00 (0.06)	2.46 (0.09)	3.64 (0.09)	1.86 (0.06)	1.98 (0.05)	1.40 * (0.22)	10.99 (0.07)	2.12 (0.06)	2.34 (0.08)	1.85 (0.07)
Waveform	19.74 (0.15)	20.78 (0.15)	18.89 (0.07)	16.28 (0.11)	21.23 (0.11)	21.04 (0.11)	28.81 (0.12)	16.45 (0.08)	15.12 * (0.06)	15.36 (0.06)
Vehicle	22.49 (0.07)	16.38 (0.07)	20.39 (0.07)	25.37 (0.08)	21.80 (0.08)	21.21 (0.07)	31.41 (0.10)	25.52 (0.07)	16.35 * (0.08)	17.15 (0.08)
Satimage	16.02 (0.82)	14.11 (0.78)	12.95 (0.75)	8.97 (0.64)	18.00 (0.86)	21.40 (0.92)	18.60 (0.87)	8.24 * (0.61)	12.58 (0.74)	12.58 (0.74)

xxxx: QDA could not be used because of singularity of the estimated class dispersion matrices.

In Parkinson's data, we could not use QDA because of singularity of the estimated class dispersion matrices. So, we used the pooled sample covariance matrix for computation

of SPD* and LSPD*. In this data set, all the nonparametric classifiers had significantly lower misclassification rates than LDA, and the classifier based on KDE had the lowest misclassification rate. The performance of the LSPD classifier was also competitive. Since the underlying distributions were non-elliptic, LSPD outperformed the SPD classifier. We observed a similar phenomena in wine data as well. The sample covariance matrices of different classes were nearly singular, and we used the pooled sample covariance matrix for computing SPD* and LSPD*. The classifier based on KDE yielded the lowest misclassification rate, while the LSPD classifier had the second best performance. Although the data dimension was quite high in both data sets, all the competing classes had low intrinsic dimensions (can be estimated using the method described by Levina and Bickel, 2004). So, nonparametric methods like KDE were not affected much by the curse of dimensionality. TREE was the only classifier with a somewhat higher misclassification rate.

In waveform data, the competing class distributions were nearly elliptic and the SPD classifier was expected to perform well. The LSPD classifier is quite flexible, and it yielded a competitive misclassification rate. The class distributions were not normal (can be checked using the method proposed in Royston, 1983) for this data, and did not have low intrinsic dimensions. As a result, LDA, QDA and the nonparametric classifiers had relatively higher misclassification rates.

In satimage data, recall that the results are based on a single training and a single test set. So, the standard errors of the misclassification rates were high for all classifiers, and it is quite difficult to compare the performance of different classifiers. Both RF and SVM with the RBF kernel had lower misclassification rates than other classifiers, while the classifiers based on SPD and LSPD had the next best performance.

7. Classification of High-dimensional Data

A serious practical limitation of many existing depth based classifiers is their computational complexity in high dimensions, and this makes such classifiers impossible to use even for moderately large dimensional data. Besides, depth functions that are based on random simplices formed by the data points (see, e.g., Liu et al., 1999; Zuo and Serfling, 2000) cannot be defined in a meaningful way if the dimension of the data exceeds the sample size. Tukey's half-space depth and projection depth both become degenerate at zero for such high-dimensional data (see, e.g., Dutta et al., 2011). Classification of high-dimensional data presents a substantial challenge to many nonparametric classification tools as well. We have seen in examples **E1** and **E2** (recall Figure 2) that nonparametric classifiers like those based on k -NN and KDE can yield poor performance when the data dimension is large. Some limitations of SVM for classification of high-dimensional data has been noted by Marron et al. (2007); Dutta and Ghosh (2016).

One of our primary motivations behind using SPD is its computational tractability (especially when the dimension is large). If the dimension exceeds the sample size, then the sample covariance matrices become singular, and we cannot use these estimates to define the empirical versions of SPD* and LSPD*. So, we use classifiers based on SPD $^\circ$ and LSPD $^\circ$. We now assume the following regularity conditions to investigate the behavior of these classifiers for such high-dimensional data.

(C) Consider two independent random vectors $\mathbf{X}_1 = (X_1^{(1)}, \dots, X_1^{(d)})^T \sim F_j$ and $\mathbf{X}_2 = (X_2^{(1)}, \dots, X_2^{(d)})^T \sim F_i$ for $1 \leq j, i \leq J$.

Further, assume that

(C1) $a_j = \lim_{d \rightarrow \infty} d^{-1} \sum_{k=1}^d E(X_1^{(k)})^2$ exists, and $d^{-1} \sum_{k=1}^d (X_1^{(k)})^2 \xrightarrow{a.s.} a_j$ as $d \rightarrow \infty$,

(C2) $b_{ji} = \lim_{d \rightarrow \infty} d^{-1} \sum_{k=1}^d E(X_1^{(k)} X_2^{(k)})$ exists, and $d^{-1} \sum_{k=1}^d X_1^{(k)} X_2^{(k)} \xrightarrow{a.s.} b_{ji}$ as $d \rightarrow \infty$.

It is not difficult to verify that for $\mathbf{X}_1 \sim F_j$ ($1 \leq j \leq J$), if we assume that the sequence of variables $\{X_1^{(k)} - E(X_1^{(k)}) : k = 1, 2, \dots\}$ centered at their means are independent with uniformly bounded eighth moments (see Theorem 1 (2) in Jung and Marron, 2009, p. 4110), or they are m -dependent processes with some appropriate conditions (see Theorem 2 in de Jong, 1995, p. 350), then the convergence results in (C1) and (C2) hold. Also, if the observations are generated from discrete time ARMA processes, all these conditions are satisfied. Stationarity of such time series is not required here. These assumptions continue to hold if the sequences $\{(X_1^{(k)})^2 - E(X_1^{(k)})^2 : k = 1, 2, \dots\}$ and $\{X_1^{(k)} X_2^{(k)} - E(X_1^{(k)} X_2^{(k)}) : k = 1, 2, \dots\}$, where $\mathbf{X}_1 \sim F_j$ and $\mathbf{X}_2 \sim F_i$ for all $1 \leq j, i \leq J$, are *mixingales* satisfying some appropriate conditions (see, e.g., Theorem 2 in de Jong, 1995, p. 350).

Define $\sigma_j^2 = a_j - b_{jj}$ and $\nu_{ji} = b_{jj} - 2b_{ji} + b_{ii}$. For the random vector $\mathbf{X}_1 \sim F_j$, σ_j^2 is the limit of $d^{-1} \sum_{k=1}^d \text{Var}(X_1^{(k)})$ as $d \rightarrow \infty$. If we consider a second independent random vector $\mathbf{X}_2 \sim F_i$ with $i \neq j$, then ν_{ji} is the limit of $d^{-1} \sum_{k=1}^d \{E(X_1^{(k)}) - E(X_2^{(k)})\}^2$ as $d \rightarrow \infty$. Hall et al. (2005) assumed a similar set of conditions to study the performance of support vector machines (SVM) with the linear kernel and the 1-NN classifier as the data dimension grows to infinity. Similar conditions on observation vectors were also considered by Jung and Marron (2009) to study consistency of principal components of the empirical covariance matrix for high-dimensional data. Under (C1) and (C2), the following theorem describes the behavior of $\mathbf{z}^\circ(\mathbf{x}) = (\text{SPD}^\circ(\mathbf{x}, F_1), \dots, \text{SPD}^\circ(\mathbf{x}, F_J))^T$ and $\mathbf{z}_h^\circ(\mathbf{x}) = (\text{LSPD}_h^\circ(\mathbf{x}, F_1), \dots, \text{LSPD}_h^\circ(\mathbf{x}, F_J))^T$ as d grows to infinity.

Theorem 6 Suppose that the conditions (C1)-(C2) hold, and $\mathbf{X} \sim F_j$ for $1 \leq j \leq J$.

(a) $\mathbf{z}^\circ(\mathbf{X}) \xrightarrow{a.s.} (c_{j1}, \dots, c_{jJ})^T = \mathbf{c}_j$ as $d \rightarrow \infty$, where $c_{jj} = 1 - \sqrt{\frac{1}{2}}$ and $c_{ji} = 1 - \sqrt{\frac{\sigma_j^2 + \nu_{ji}}{\sigma_j^2 + \sigma_i^2 + \nu_{ji}}}$ for $1 \leq j \neq i \leq J$.

(b) Assume that $h \rightarrow \infty$ and $d \rightarrow \infty$ in such a way that $\sqrt{d}/h \rightarrow 0$ or $A_0(> 0)$. Then, $\mathbf{z}_h^\circ(\mathbf{X}) \xrightarrow{a.s.} g_0(\mathbf{0}; \mathbf{c}_j)$ or $\mathbf{c}_j' = (g_0(\epsilon_{j1}, A_0), c_{j1}, \dots, g_0(\epsilon_{jJ}, A_0), c_{jJ})^T$ depending on whether $\sqrt{d}/h \rightarrow 0$ or A_0 , respectively. Here $K(\mathbf{t}) = g_0(\|\mathbf{t}\|)$, $\epsilon_{jj} = \sqrt{2}\sigma_j$ and $\epsilon_{ji} = \sqrt{\sigma_j^2 + \sigma_i^2 + \nu_{ji}}$ for $j \neq i$.

(c) Assume that $h > 1$, and $\sqrt{d}/h \rightarrow \infty$ as $d \rightarrow \infty$. Then, $\mathbf{z}_h^\circ(\mathbf{X}) \xrightarrow{a.s.} \mathbf{0}_J$.

The \mathbf{c}_j 's as well as the \mathbf{c}_j 's in the statement of Theorem 6 are *distinct* for all $1 \leq j \leq J$ whenever either $\sigma_j^2 \neq \sigma_i^2$ or $\nu_{ji} \neq 0$ for all $1 \leq j \neq i \leq J$ (see Lemma 11 in Appendix A). In such a case, part (a) of Theorem 6 implies that for large d , $\mathbf{z}^\circ(\mathbf{x})$ becomes degenerate at points depending on the class distributions. So, $\mathbf{z}^\circ(\mathbf{x})$ has good discriminatory power, and our classifier based on SPD $^\circ$ can discriminate well among the J populations. Further, it follows from part (b) that when both d and h grow to infinity in such a way that $\sqrt{d}/h \rightarrow 0$ or to a positive constant, $\mathbf{z}_h^\circ(\mathbf{x})$ has good discriminatory power and the classifier based on LSPD $^\circ$ can yield low misclassification probability. However, part (c) shows that if \sqrt{d} grows

at a rate faster than h_n , $\mathbf{z}_n^*(\mathbf{x})$ converges to the same value $\mathbf{0}_J$ and it becomes non-informative. Consequently, the classifier based on $\text{LSPD}_{h_n}^*$ will lead to a high misclassification probability in this case.

To evaluate the performance of our depth based classifiers for high-dimensional data, we considered examples **E1-E7** with $d = 200$. In each example, we generated 20 observations from each class to constitute the training sample, while 250 observations from each class were used to form the test set. We generated 500 training and test sets, and the average test set misclassification rates of the different classifiers along with their corresponding standard errors are reported in Table 4. The Bayes risks were *almost zero* in all these examples, and we have not stated them in Table 4. We did not standardize the data for KDE and k -NN. QDA could not be used in these examples, and we used \mathbf{I}_d instead of the pooled sample covariance matrix for LDA. When the competing classes have equal priors (which is the case in simulated examples), this leads to the Euclidean distance based classifier which classifies an observation to the class having the nearest centroid.

As we have mentioned before, we use SPD° and LSPD° for classification of these high-dimensional data sets. For a single iteration, the LSPD classifier required an average CPU time of 8.82 seconds to compute the weight function $W(h)$, 0.39 seconds for fitting GAM using the full training data, and 0.06 seconds for classification of 500 test cases.

Table 4: Misclassification rates (in %) of different classifiers in simulated data sets.

Example	LDA [†]	SVM (linear)	SVM (RBF)	k -NN	KDE	TREE	RF	SPD [°]	LSPD [°]
E1	50.93 (0.13)	47.57 (0.09)	28.97 (0.38)	49.71 (0.06)	49.99 (0.15)	45.72 (0.14)	41.95 (0.03)	0.27 * (0.03)	0.31 (0.03)
E2	45.84 (0.08)	45.69 (0.07)	32.70 (0.18)	49.96 (0.01)	49.92 (0.12)	43.70 (0.03)	39.36 (0.03)	0.08 * (0.03)	0.09 (0.03)
E3	0.20 (0.01)	0.29 (0.01)	0.00 * (0.00)	49.99 (0.01)	49.98 (0.12)	27.46 (0.17)	0.28 (0.00)	0.00 * (0.00)	0.00 * (0.00)
E4	34.87 (0.26)	44.28 (0.15)	10.43 (0.43)	0.19 (0.08)	38.55 (0.08)	23.57 (0.24)	0.68 (0.45)	0.13 * (0.12)	0.06 (0.06)
E5	40.83 (0.07)	44.61 (0.11)	13.69 (0.15)	49.98 (0.01)	49.93 (0.03)	18.93 (0.00)	0.00 * (0.00)	0.84 (0.04)	0.80 (0.04)
E6	50.11 (0.12)	48.16 (0.14)	31.03 (0.26)	46.52 (0.18)	47.03 (0.14)	48.20 (0.17)	45.00 (0.20)	30.98 (0.20)	29.76 * (0.19)
E7	44.74 (0.45)	35.06 (0.24)	31.82 (0.48)	18.92 * (0.22)	22.91 (0.32)	36.82 (0.19)	22.36 (0.19)	26.33 (0.26)	25.96 (0.27)

[†] \mathbf{I}_d was used instead of the pooled sample covariance matrix.

In the first five examples, the two competing classes had separation between either in their locations and/or scales. So, good performance of the SPD° and LSPD° classifiers was expected in view of Theorem 6 and Lemma 11 (see Appendix A). In **E1** and **E2**, recall that the component distributions of the two classes differed only in scales. The SPD° and LSPD° classifiers performed well in these examples, and the former had an edge due to ellipticity of the class distributions. Surprisingly, all other classifiers failed to extract this separability information properly, and had misclassification rates higher than 25%. Since the Bayes class boundaries were highly nonlinear in these two examples, poor performance of linear SVM and LDA was quite expected. Dutta and Ghosh (2016) showed that when one component

distribution from the first class and one from the second class differ only in their scales, the k -NN classifier gives a decision in favor of the distribution with a smaller spread (also see Hall et al., 2005). This was the main reason behind the poor performance of the k -NN classifier. Similar arguments can be given for the poor performance of the classifier based on KDE. In these two examples, splitting based on a single variable failed to yield significant reduction in the impurity function (one can see this in Figure 1). So, TREE and RF had relatively higher misclassification rates. In **E3**, the two Gaussian distributions differed in their locations and scales. Barring TREE, k -NN and the classifier based on KDE, all other classifiers yielded misclassification rates close to zero. Since the scale difference between the two classes dominates the location difference, such a poor performance of the classifier based on KDE and k -NN was expected (see the results in Hall et al., 2005; Dutta and Ghosh, 2016). The same explanation holds for **E5** as well. These nonparametric classifiers yielded excellent performance in **E4**, where the component distribution differ only in their locations. However, TREE and RF failed to have satisfactory performance here. Splitting based on linear combinations of the variables may be helpful in **E4** (see Figure 4).

Examples **E6** and **E7** were difficult to deal with. Unlike **E1-E5**, none of the classifiers could achieve misclassification rates close to zero in these two examples. Conditions (C1) and (C2) do not hold here, and Theorem 6 is not applicable. The LSPD classifier had the best performance in **E6** (just like the case with $d = 5$ in Section 5). SVM with the RBF kernel and the SPD classifier also led to competitive misclassification rates. Their performance was much better than all other classifiers. In **E7**, the linear classifiers and SVM with the RBF kernel could not perform well. This is also consistent with what we observed in Section 5. Barring TREE, all other classifiers yielded competitive performances in this example. Among them the k -NN classifier led to the lowest misclassification rate.

We also analyzed two high-dimensional benchmark data sets, namely, lightning-2 data and colon data (Alon et al., 1999). The first data set is from the UCR time series classification archive (http://www.cs.ucr.edu/~eamonn/time_series_data/), while the other one is taken from the R library rda. In each case, we formed 500 training and test sets by randomly partitioning each data into two almost equal parts. The average test set misclassification rates of different classifiers are reported in Table 5.

Table 5: Misclassification rates (in %) of different classifiers in real data sets.

Data set	d	J	Sample size [†]	LDA [†]	SVM (linear)	SVM (RBF)	k -NN	KDE	TREE	RF	SPD	LSPD
Lightning-2	637	2	Train	60	31.86	35.64	28.73	29.89	28.11	33.69	22.08 *	27.70
			Test	61	(0.25)	(0.35)	(0.32)	(0.20)	(0.30)	(0.34)	(0.30)	(0.30)
Colon	2000	2	Train	31	14.47	16.38	21.48	22.47	23.20	28.78	19.28	19.66
			Test	31	(0.21)	(0.23)	(0.25)	(0.27)	(0.28)	(0.35)	(0.24)	(0.31)

[†] \mathbf{I}_d was used instead of the pooled sample covariance matrix.

Lightning-2 data consist of observations that are realizations of a time series. In this data set, RF had the best performance followed by the LSPD classifier. The SPD classifier also worked well and yielded the third best performance. The class distributions for this data set turn out to be non-elliptic (can be verified using the method proposed by Li et al., 1997) with low intrinsic dimensions (Levina and Bickel, 2004). As a consequence, the classifier based on KDE and k -NN yielded reasonably good performances.

Colon data contain micro-array expression levels of 2000 genes for ‘normal’ and ‘colon cancer’ tissues. There was a good linear separation among the observations from the two competing classes, and the linear classifiers lead to low misclassification rates. Among the other classifiers, the LSPD classifier yielded the minimum misclassification rate closely followed by RF and the SPD classifier. These three classifiers were less affected by the curse of dimensionality.

In these high-dimensional benchmark data sets, the data had low intrinsic dimensions due to high correlation among the measurement variables (Levina and Bickel, 2004). Moreover, data from the competing classes differed mainly in their locations. As a consequence, though the proposed LSPD classifier had a good overall performance, its superiority over the nonparametric methods was not as prominent as it was in the simulated examples.

Acknowledgments

The authors are grateful to Prof. Probal Chaudhuri for his valuable contributions to this manuscript. They are also thankful to the Action Editor and two anonymous reviewers for providing them with several helpful comments. The first author would like to thank Prof. Thomas W. Yee for his help with VGAM, and Prof. Jun Li for sharing R codes of the DD classifier.

Appendix A. Proofs and Mathematical Details

Lemma 7 *If F has a spherically symmetric density $f(\mathbf{x}) = g(\|\mathbf{x}\|)$ on \mathbb{R}^d with $d > 1$, then $\|E_F[u(\mathbf{x} - \mathbf{X})]\|$ is a non-negative monotonically increasing function of $\|\mathbf{x}\|$.*

Proof of Lemma 7 : In view of spherical symmetry of $f(\mathbf{x})$, $S(\mathbf{x}) = \|E_F[u(\mathbf{x} - \mathbf{X})]\|$ is invariant under orthogonal transformations of \mathbf{x} . Consequently, $S(\mathbf{x}) = \eta(\|\mathbf{x}\|)$ for some non-negative function η . Consider now \mathbf{x}_1 and \mathbf{x}_2 such that $\|\mathbf{x}_1\| < \|\mathbf{x}_2\|$. Using spherical symmetry of $f(\mathbf{x})$, without loss of generality, we can assume $\mathbf{x}_i = (t_i, 0, \dots, 0)^T$ for $i = 1, 2$ such that $|t_1| < |t_2|$. For any $\mathbf{x} = (t, 0, \dots, 0)^T$, we have

$$S(\mathbf{x}) = \left| E_F \left[\frac{(t - X_1)}{\sqrt{(t - X_1)^2 + X_2^2 + \dots + X_d^2}} \right] \right|,$$

due to spherical symmetry of $f(\mathbf{x})$. For any $\mathbf{x} \in \mathbb{R}^d$ with $d > 1$, $E_F[\|\mathbf{x} - \mathbf{X}\|]$ is a strictly convex function of \mathbf{x} in this case. Consequently, it is a strictly convex function of t . Observe now that $S(\mathbf{x})$ with this choice of \mathbf{x} is the absolute value of the derivative of $E_F[\|\mathbf{x} - \mathbf{X}\|]$ w.r.t. t . This derivative is a symmetric function of t that vanishes at $t = 0$. Hence, $S(\mathbf{x})$ is an increasing function of $|t|$, and this proves that $\eta(\|\mathbf{x}_1\|) < \eta(\|\mathbf{x}_2\|)$. ■

Proof of Theorem 1 : If the population distribution $f_j(\mathbf{x})$ is elliptically symmetric, we have $f_j(\mathbf{x}) = |\Sigma_j|^{-1/2} g_j(\delta(\mathbf{x}, F_j))$, where $\delta(\mathbf{x}, F_j) = \|\Sigma_j^{-1/2}(\mathbf{x} - \mu_j)\|$ is the Mahalanobis distance for $1 \leq j \leq J$. Since $\text{SPD}^*(\mathbf{x}, F_j) = 1 - \|E[u(\Sigma_j^{-1/2}(\mathbf{x} - \mu_j))]\|$ is affine invariant, it is a function of $\delta(\mathbf{x}, F_j)$. Again, as $\Sigma_j^{-1/2}(\mathbf{X} - \mu_j)$ has a spherically symmetric distribution

with its center at the origin, from Lemma 7 it follows that $\text{SPD}^*(\mathbf{x}, F_j)$ is a monotonically decreasing function of $\delta(\mathbf{x}, F_j)$. Therefore, $\delta(\mathbf{x}, F_j)$ is also a function of $\text{SPD}^*(\mathbf{x}, F_j)$ and using this fact $f_j(\mathbf{x})$ can also be expressed as

$$f_j(\mathbf{x}) = \psi_j(\text{SPD}^*(\mathbf{x}, F_j)) \text{ for all } 1 \leq j \leq J,$$

where ψ_j is an appropriate real-valued function that depends on g_j . Now, one can check that

$$\log \left[\frac{p(j|\mathbf{x})}{p(J|\mathbf{x})} \right] = \log(\pi_j/\pi_J) + \log \psi_j(\text{SPD}^*(\mathbf{x}, F_j)) - \log \psi_J(\text{SPD}^*(\mathbf{x}, F_J)).$$

for $1 \leq j \leq (J-1)$. Now, if we define $\varphi_{jj}(z) = \log \pi_j + \log \psi_j(z)$ and $\varphi_{jJ}(z) = 0$ for $1 \leq j \neq i \leq (J-1)$; and $\varphi_{1J}(z) = \dots = \varphi_{(J-1)J}(z) = -\log \pi_J - \log \psi_J(z)$, then the proof is complete. ■

Remark 8 *If $f_j(\mathbf{x})$ is unimodal, $\psi_j(z)$ is monotonically increasing for $1 \leq j \leq J$. Moreover, if the distributions differ only in their locations, then the ψ_j s are same for all classes. In that case, $f_j(\mathbf{x}) > f_i(\mathbf{x}) \Leftrightarrow \delta(\mathbf{x}, F_j) < \delta(\mathbf{x}, F_i) \Leftrightarrow \text{SPD}^*(\mathbf{x}, F_j) > \text{SPD}^*(\mathbf{x}, F_i)$ for $1 \leq i \neq j \leq J$, and hence the classifier turns out to be the maximum SPD classifier.*

Proof of Theorem 3(a) : Let $h < 1$. For any fixed $\mathbf{x} \in \mathbb{R}^d$ and the distribution function F_j , we have $\text{LSPD}_h^*(\mathbf{x}, F_j) = E_{F_j}[K_h(\mathbf{t}u(\mathbf{t}))]$, where $\mathbf{t} = \Sigma_j^{-1/2}(\mathbf{x} - \mathbf{X})$ for $1 \leq j \leq J$. For the first term in the expression of $\text{LSPD}_h^*(\mathbf{x}, F_j)$ above, we have

$$E_{F_j}[K_h(\mathbf{t})] = \int_{\mathbb{R}^d} \frac{1}{h^d} K_h(\Sigma_j^{-1/2}(\mathbf{x} - \mathbf{v})) f_j(\mathbf{v}) d\mathbf{v} = |\Sigma_j|^{1/2} \int_{\mathbb{R}^d} K(\mathbf{y}) f_j(\mathbf{x} - h\Sigma_j^{1/2}\mathbf{y}) d\mathbf{y},$$

where $\mathbf{y} = h^{-1}\Sigma_j^{-1/2}(\mathbf{x} - \mathbf{v})$. So, using Taylor’s expansion of $f_j(\mathbf{x})$, we get

$$E_{F_j}[K_h(\mathbf{t})] = |\Sigma_j|^{1/2} f_j(\mathbf{x}) - h|\Sigma_j|^{1/2} \int_{\mathbb{R}^d} K(\mathbf{y}) (\Sigma_j^{1/2}\mathbf{y})^T \nabla f_j(\xi) d\mathbf{y},$$

where ξ lies on the line joining \mathbf{x} and $(\mathbf{x} - h\Sigma_j^{1/2}\mathbf{v})$. Using the Cauchy-Schwarz inequality, one gets $|E_{F_j}[K_h(\mathbf{t})] - |\Sigma_j|^{1/2} f_j(\mathbf{x})| \leq h|\Sigma_j|^{1/2} \lambda_j^{1/2} M_j^c M_K$, where $M_j^c = \sup_{\mathbf{x} \in \mathbb{R}^d} \|\nabla f_j(\mathbf{x})\|$, $M_K = \int \|\mathbf{y}\| K(\mathbf{y}) d\mathbf{y}$, and λ_j is the largest eigenvalue of Σ_j . This implies $|E_{F_j}[K_h(\mathbf{t})] - |\Sigma_j|^{1/2} f_j(\mathbf{x})| \rightarrow 0$ as $h \rightarrow 0$ for $1 \leq j \leq J$.

For the second term in the expression of $\text{LSPD}_h^*(\mathbf{x}, F_j)$, a similar argument yields

$$\begin{aligned} E_{F_j}[K_h(\mathbf{t})u(\mathbf{t})] &= |\Sigma_j|^{1/2} \int_{\mathbb{R}^d} K(\mathbf{y}) u(\mathbf{y}) f_j(\mathbf{x} - h\Sigma_j^{1/2}\mathbf{y}) d\mathbf{y} \\ &= -h|\Sigma_j|^{1/2} \int_{\mathbb{R}^d} K(\mathbf{y}) u(\mathbf{y}) (\Sigma_j^{1/2}\mathbf{y})^T \nabla f_j(\xi) d\mathbf{y} \text{ (as } \int_{\mathbb{R}^d} K(\mathbf{y}) u(\mathbf{y}) d\mathbf{y} = \mathbf{0}). \end{aligned}$$

Now, $\|E_{F_j}[K_h(\mathbf{t})u(\mathbf{t})]\| \leq h|\Sigma_j|^{1/2} \lambda_j^{1/2} M_j^c M_K \rightarrow 0$ and this implies that $\text{LSPD}_h^*(\mathbf{x}, F_j) \rightarrow |\Sigma_j|^{1/2} f_j(\mathbf{x})$, as $h \rightarrow 0$. Consequently, we have $\mathbf{z}_h^*(\mathbf{x}) \rightarrow (|\Sigma_1|^{1/2} f_1(\mathbf{x}), \dots, |\Sigma_J|^{1/2} f_J(\mathbf{x}))^T$ as $h \rightarrow 0$. ■

Proof of Theorem 3(b) : Here we consider the case $h > 1$. Take any fixed $\mathbf{x} \in \mathbb{R}^d$ and a j with $1 \leq j \leq J$. For any fixed \mathbf{t} , since $K(\mathbf{t}/h) \rightarrow K(\mathbf{0})$ as $h \rightarrow \infty$ and K is bounded, using Dominated Convergence Theorem (DCT), one can show that $\text{LSPD}_h^*(\mathbf{x}, F_j) \rightarrow K(\mathbf{0})\text{SPD}^*(\mathbf{x}, F_j)$ as $h \rightarrow \infty$. So, $\mathbf{z}_h^*(\mathbf{x}) \rightarrow (K(\mathbf{0})\text{SPD}^*(\mathbf{x}, F_1), \dots, K(\mathbf{0})\text{SPD}^*(\mathbf{x}, F_J))^T$ as $h \rightarrow \infty$. ■

Proof of Theorem 5 : Define the sets $B_n = \{\mathbf{x} = (x_1, \dots, x_d) : \|\mathbf{x}\| \leq \sqrt{dn}\}$, and $A_n = \{\mathbf{x} : n^2x_i$ is an integer and $|x_i| \leq n$ for all $1 \leq i \leq d\}$. Clearly $A_n \subset B_n \subset \mathbb{R}^d$, the set B_n is a closed ball and the set A_n has cardinality $(2n^2 + 1)^d$. We will prove almost sure (a.s.) uniform convergence on three disjoint sets: (i) A_n , (ii) $B_n \setminus A_n$ and (iii) B_n^c . Consider any fixed $h \in (0, 1]$. Recall that for this choice of h , $\text{LSPD}_h^\circ(\mathbf{x}, F)$ (see equation (3)) and $\text{LSPD}_h^\circ(\mathbf{x}, F_n)$ are defined as follows:

$$\begin{aligned} \text{LSPD}_h^\circ(\mathbf{x}, F_n) &= \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{\mathbf{x} - \mathbf{X}_i}{h}\right) - \left\| \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{\mathbf{x} - \mathbf{X}_i}{h}\right) u(\mathbf{x} - \mathbf{X}_i) \right\|, \text{ and} \\ \text{LSPD}_h^\circ(\mathbf{x}, F) &= \frac{1}{h^d} E\left[K\left(\frac{\mathbf{x} - \mathbf{X}}{h}\right)\right] - \frac{1}{h^d} \|E\left[K\left(\frac{\mathbf{x} - \mathbf{X}}{h}\right) u(\mathbf{x} - \mathbf{X})\right]\|. \end{aligned}$$

(i) Define $\mathbf{Z}_i = K(h^{-1}(\mathbf{x} - \mathbf{X}_i))u(\mathbf{x} - \mathbf{X}_i) - E[K(h^{-1}(\mathbf{x} - \mathbf{X}))u(\mathbf{x} - \mathbf{X})]$ for $1 \leq i \leq n$. Note that \mathbf{Z}_i 's are independent and identically distributed (i.i.d.) with $E(\mathbf{Z}_i) = \mathbf{0}$ and $\|\mathbf{Z}_i\| \leq 2K(\mathbf{0})$. Fix an $\epsilon > 0$. Using the exponential inequality for sums of i.i.d. random vectors (see Yurinskii, 1976, p. 491), we obtain $P\left(\|n^{-1} \sum_{i=1}^n \mathbf{Z}_i\| \geq \epsilon\right) \leq 2e^{-C_0 n \epsilon^2}$. Here C_0 is a positive constant that depends on $K(\mathbf{0})$ and ϵ . This now implies that

$$\begin{aligned} P\left(\left\| \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{\mathbf{x} - \mathbf{X}_i}{h}\right) u(\mathbf{x} - \mathbf{X}_i) \right\| - \left\| \frac{1}{h^d} E\left[K\left(\frac{\mathbf{x} - \mathbf{X}}{h}\right) u(\mathbf{x} - \mathbf{X})\right] \right\| \geq \epsilon\right) \\ \leq P\left(\left\| \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{\mathbf{x} - \mathbf{X}_i}{h}\right) u(\mathbf{x} - \mathbf{X}_i) - \frac{1}{h^d} E\left[K\left(\frac{\mathbf{x} - \mathbf{X}}{h}\right) u(\mathbf{x} - \mathbf{X})\right] \right\| \geq \epsilon\right) \\ = P\left(\left\| \frac{1}{n} \sum_{i=1}^n \mathbf{Z}_i \right\| \geq h^d \epsilon\right) \leq 2e^{-C_0 n h^{2d} \epsilon^2}. \end{aligned} \quad (6)$$

For a fixed value of h , $\sum_{i=1}^n K(h^{-1}(\mathbf{x} - \mathbf{X}_i))$ is a sum of i.i.d. bounded random variables. Using Bernstein's inequality, we obtain

$$\begin{aligned} P\left(\left| \frac{1}{n} \sum_{i=1}^n K\left(\frac{\mathbf{x} - \mathbf{X}_i}{h}\right) - E\left[K\left(\frac{\mathbf{x} - \mathbf{X}}{h}\right)\right] \right| \geq \epsilon\right) &\leq 2e^{-C_1 n \epsilon^2}, \\ P\left(\left| \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{\mathbf{x} - \mathbf{X}_i}{h}\right) - \frac{1}{h^d} E\left[K\left(\frac{\mathbf{x} - \mathbf{X}}{h}\right)\right] \right| \geq \epsilon\right) &\leq 2e^{-C_1 n h^{2d} \epsilon^2}. \end{aligned} \quad (7)$$

for some suitable positive constant C_1 . This implies

$$\begin{aligned} \text{Combining (6) and (7), we get } P(|\text{LSPD}^\circ(\mathbf{x}, F_n) - \text{LSPD}^\circ(\mathbf{x}, F)| \geq \epsilon) &\leq C_3 e^{-C_4 n h^{2d} \epsilon^2} \text{ for} \\ \text{some suitable constants } C_3 \text{ and } C_4. \text{ Since the cardinality of } A_n \text{ is } (2n^2 + 1)^d, \text{ we have} \\ P\left(\sup_{\mathbf{x} \in A_n} |\text{LSPD}^\circ(\mathbf{x}, F_n) - \text{LSPD}^\circ(\mathbf{x}, F)| \geq \epsilon\right) &\leq C_3 (2n^2 + 1)^d e^{-C_4 n h^{2d} \epsilon^2}. \end{aligned} \quad (8)$$

Now, $\sum_{n \geq 1} (2n^2 + 1)^d e^{-C_4 n h^{2d} \epsilon^2} < \infty$. So, an application of Borel-Cantelli lemma implies that $\sup_{\mathbf{x} \in A_n} |\text{LSPD}_h^\circ(\mathbf{x}, F_n) - \text{LSPD}_h^\circ(\mathbf{x}, F)| \xrightarrow{\text{a.s.}} 0$ as $n \rightarrow \infty$.

(ii) Consider the set $B_n \setminus A_n$. Given any \mathbf{x} in $B_n \setminus A_n$, there exists $\mathbf{y} \in A_n$ such that $\|\mathbf{x} - \mathbf{y}\| \leq \sqrt{2}/n^2$. First we will show that $|\text{LSPD}^\circ(\mathbf{y}, F_n) - \text{LSPD}^\circ(\mathbf{x}, F_n)| \xrightarrow{\text{a.s.}} 0$ as $n \rightarrow \infty$. Using the mean-value theorem, one obtains

$$\left| \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{\mathbf{x} - \mathbf{X}_i}{h}\right) - \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{\mathbf{y} - \mathbf{X}_i}{h}\right) \right| \leq \frac{1}{nh^{d+1}} \sum_{i=1}^n \left| (\mathbf{x} - \mathbf{y})^T \nabla K\left(\frac{\boldsymbol{\xi} - \mathbf{X}_i}{h}\right) \right|,$$

where $\boldsymbol{\xi}$ lies on the line joining \mathbf{x} and \mathbf{y} . Note that the right hand side is less than $\frac{M'_K \sqrt{2}}{h^{d+1} n^2}$, and $M'_K = \sup_{\mathbf{t}} \|\nabla K(\mathbf{t})\|$. This upper bound is free of \mathbf{x} , and goes to 0 as $n \rightarrow \infty$. Now,

$$\begin{aligned} &\left\| \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{\mathbf{x} - \mathbf{X}_i}{h}\right) u(\mathbf{x} - \mathbf{X}_i) \right\| - \left\| \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{\mathbf{y} - \mathbf{X}_i}{h}\right) u(\mathbf{y} - \mathbf{X}_i) \right\| \\ &\leq \left\| \frac{1}{nh^d} \sum_{i=1}^n \left[K\left(\frac{\mathbf{x} - \mathbf{X}_i}{h}\right) u(\mathbf{x} - \mathbf{X}_i) - K\left(\frac{\mathbf{y} - \mathbf{X}_i}{h}\right) u(\mathbf{y} - \mathbf{X}_i) \right] \right\| \\ &\leq \frac{1}{nh^d} \sum_{i=1}^n \left[K\left(\frac{\mathbf{x} - \mathbf{X}_i}{h}\right) - K\left(\frac{\mathbf{y} - \mathbf{X}_i}{h}\right) \right] + K(\mathbf{0}) \left\| \frac{1}{nh^d} \sum_{i=1}^n [u(\mathbf{x} - \mathbf{X}_i) - u(\mathbf{y} - \mathbf{X}_i)] \right\|. \end{aligned} \quad (9)$$

We have proved above that the first part converges to 0 in a.s. sense.

For the second part, consider a ball of radius $1/n$ around \mathbf{x} (say, $B(\mathbf{x}, 1/n)$). Now,

$$\begin{aligned} \left\| \frac{1}{nh^d} \sum_{i=1}^n [u(\mathbf{x} - \mathbf{X}_i) - u(\mathbf{y} - \mathbf{X}_i)] \right\| &\leq \left\| \frac{2}{nh^d} \sum_{i=1}^n I[\mathbf{X}_i \in B(\mathbf{x}, 1/n)] \right\| + \frac{2n}{h^d} \|\mathbf{x} - \mathbf{y}\| \\ &\leq \frac{2}{h^d} \left\| \frac{1}{n} \sum_{i=1}^n I[\mathbf{X}_i \in B(\mathbf{x}, 1/n)] - P[\mathbf{X}_1 \in B(\mathbf{x}, 1/n)] \right\| \\ &\quad + \frac{2}{h^d} P[\mathbf{X}_1 \in B(\mathbf{x}, 1/n)] + \frac{2n\sqrt{2}}{n^2 h^d}. \end{aligned}$$

Note that $I[\mathbf{X}_i \in B(\mathbf{x}, 1/n)]$'s are i.i.d. bounded random variables with expectation $P[\mathbf{X} \in B(\mathbf{x}, 1/n)]$. Therefore, a.s. convergence of the first term follows from Bernstein's inequality. Since $P[\mathbf{X} \in B(\mathbf{x}, 1/n)] \leq M_f n^{-d}$ (where $M_f = \sup_{\mathbf{x}} f(\mathbf{x}) < \infty$), the second term converges to 0. For any fixed h , the third term also converges to 0 as $n \rightarrow \infty$. So, we have $|\text{LSPD}_h^\circ(\mathbf{x}, F_n) - \text{LSPD}_h^\circ(\mathbf{y}, F_n)| \xrightarrow{\text{a.s.}} 0$ as $n \rightarrow \infty$.

Similarly, one can prove that $|\text{LSPD}_h^\circ(\mathbf{x}, F) - \text{LSPD}_h^\circ(\mathbf{y}, F)| \xrightarrow{\text{a.s.}} 0$ as $n \rightarrow \infty$. In the arguments above, all the bounds are free from \mathbf{x} and \mathbf{y} . We have also proved that $\sup_{\mathbf{y} \in A_n} |\text{LSPD}_h^\circ(\mathbf{y}, F_n) - \text{LSPD}_h^\circ(\mathbf{y}, F)| \xrightarrow{\text{a.s.}} 0$ as $n \rightarrow \infty$. Combining all these results, we have $\sup_{\mathbf{x} \in B_n \setminus A_n} |\text{LSPD}_h^\circ(\mathbf{x}, F_n) - \text{LSPD}_h^\circ(\mathbf{x}, F)| \xrightarrow{\text{a.s.}} 0$ as $n \rightarrow \infty$.

(iii) Now, consider the region outside B_n (i.e., the set B_n^c). First note that

$$\sup_{\mathbf{x} \in B_n^c} |\text{LSPD}_h^\circ(\mathbf{x}, F_n) - \text{LSPD}_h^\circ(\mathbf{x}, F)| \leq \sup_{\mathbf{x} \in B_n^c} \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{\mathbf{x} - \mathbf{X}_i}{h}\right) + \sup_{\mathbf{x} \in B_n^c} \frac{1}{h^d} E\left[K\left(\frac{\mathbf{x} - \mathbf{X}}{h}\right)\right].$$

We will show that both of these terms become sufficiently small as $n \rightarrow \infty$.

Fix an $\epsilon > 0$. We can choose two constants M_1 and M_2 such that $P(\|\mathbf{X}\| \geq M_1) \leq h^d \epsilon / 2K(\mathbf{0})$ and $K(\mathbf{t}) \leq h^d \epsilon / 2$ when $\|\mathbf{t}\| \geq M_2$. Now, one can check that

$$\frac{1}{h^d} E \left[K \left(\frac{\mathbf{x} - \mathbf{X}}{h} \right) \right] \leq \frac{1}{h^d} E \left[K \left(\frac{\mathbf{x} - \mathbf{X}}{h} \right) I(\|\mathbf{X}\| \leq M_1) \right] + \frac{1}{h^d} K(\mathbf{0}) P(\|\mathbf{X}\| > M_1).$$

If $\mathbf{x} \in B_n^c$ and $\|\mathbf{X}\| \leq M_1$, then $h^{-1} \|\mathbf{x} - \mathbf{X}\| \geq h^{-1} |\sqrt{dn} - M_1|$. Choose n large enough so that $|\sqrt{dn} - M_1| \geq M_2 h$, and this implies $K(h^{-1}(\mathbf{x} - \mathbf{X})) \leq h^d \epsilon / 2$. So, we obtain

$$\begin{aligned} \frac{1}{h^d} E \left[K \left(\frac{\mathbf{x} - \mathbf{X}}{h} \right) \right] &\leq \frac{\epsilon}{2} + \frac{1}{h^d} K(\mathbf{0}) P(\|\mathbf{X}\| > M_1) \leq \epsilon, \text{ and} \\ \frac{1}{nh^d} \sum_{i=1}^n K \left(\frac{\mathbf{x} - \mathbf{X}_i}{h} \right) &\leq \frac{\epsilon}{2} + \frac{1}{h^d} K(\mathbf{0}) \frac{1}{n} \sum_{i=1}^n I(\|\mathbf{X}_i\| > M_1) \end{aligned}$$

$$\leq \epsilon + \frac{1}{h^d} K(\mathbf{0}) \left| \frac{1}{n} \sum_{i=1}^n I(\|\mathbf{X}_i\| > M_1) - P(\|\mathbf{X}\| > M_1) \right|.$$

The Glivenko-Cantelli theorem implies that the last term on the right hand side converges to 0 as $n \rightarrow \infty$. So, we have $\sup_{\mathbf{x} \in B_n^c} |\text{LSPD}_h^{\circ}(\mathbf{x}, F_n) - \text{LSPD}_h^{\circ}(\mathbf{x}, F)| \xrightarrow{a.s.} 0$ as $n \rightarrow \infty$.

Combining the arguments in parts (i), (ii) and (iii) and for a fixed $h \in (0, 1]$, we get $\sup_{\mathbf{x}} |\text{LSPD}_h^{\circ}(\mathbf{x}, F_n) - \text{LSPD}_h^{\circ}(\mathbf{x}, F)| \xrightarrow{a.s.} 0$ as $n \rightarrow \infty$. If we have $h > 1$, then this convergence result can be proved in a similar way. For this case, recall that the definition of $\text{LSPD}^{\circ}(\mathbf{x}, F)$ does not involve the h^d term in the denominator (see equation (3)). ■

Remark 9 Following the proof of Theorem 5, it is easy to check that a.s. convergence holds when h diverges to infinity with n .

Remark 10 The result continues to hold when $h \rightarrow 0$ as well. However, for a.s. convergence in part (i) (to use the Borel-Cantelli lemma) we require $nh^{2d} / \log n \rightarrow \infty$ as $n \rightarrow \infty$. In part (iii), we need M_1 and M_2 to vary with n . Assume the first moment of the density corresponding to F to be finite, and $\int \|\mathbf{t}\| K(\mathbf{t}) dt < \infty$ (which implies that $\|\mathbf{t}\| K(\mathbf{t}) \rightarrow 0$ as $\|\mathbf{t}\| \rightarrow \infty$). Also, assume that $nh^{2d} / \log n \rightarrow \infty$ as $n \rightarrow \infty$. We can now choose $M_1 = M_2 = \sqrt{n}$ to ensure that both $P(\|\mathbf{X}\| \geq M_1) \leq h^d \epsilon / 2K(\mathbf{0})$ and $K(\mathbf{t}) \leq h^d \epsilon / 2$ for $\|\mathbf{t}\| \geq M_2$ hold for a sufficiently large n .

Proof of Theorem 6(a) : Consider two independent random vectors $\mathbf{X} = (X^{(1)}, \dots, X^{(d)})^T \sim F_j$ and $\mathbf{X}_1 = (X_1^{(1)}, \dots, X_1^{(d)})^T \sim F_j$, where $1 \leq j \leq J$. It follows from (C1) and (C2) that $\|\mathbf{X} - \mathbf{X}_1\| / \sqrt{d} \xrightarrow{a.s.} \sqrt{2\sigma_j^2}$ as $d \rightarrow \infty$. So, for almost every realization \mathbf{x} of $\mathbf{X} \sim F_j$,

$$\|\mathbf{x} - \mathbf{X}_1\| / \sqrt{d} \xrightarrow{a.s.} \sqrt{2\sigma_j^2} \text{ as } d \rightarrow \infty. \quad (10)$$

Next, consider two independent random vectors $\mathbf{X} \sim F_j$ and $\mathbf{X}_1 \sim F_i$ for $1 \leq i \neq j \leq J$. Using (C1) and (C2), we get $\|\mathbf{X} - \mathbf{X}_1\| / \sqrt{d} \xrightarrow{a.s.} \sqrt{\sigma_j^2 + \sigma_i^2 + \nu_{ji}}$ as $d \rightarrow \infty$. Consequently, for almost every realization \mathbf{x} of $\mathbf{X} \sim F_j$

$$\|\mathbf{x} - \mathbf{X}_1\| / \sqrt{d} \xrightarrow{a.s.} \sqrt{\sigma_j^2 + \sigma_i^2 + \nu_{ji}} \text{ as } d \rightarrow \infty. \quad (11)$$

Let us next consider $\langle \mathbf{x} - \mathbf{X}_1, \mathbf{x} - \mathbf{X}_2 \rangle$, where $\mathbf{X} \sim F_j$, $\mathbf{X}_1, \mathbf{X}_2 \sim F_i$ are independent random vectors, and $\langle \cdot, \cdot \rangle$ denotes the inner product in \mathbb{R}^d . Therefore, for almost every realization \mathbf{x} of \mathbf{X} , arguments similar to those used in (10) and (11) yield

$$\frac{\langle \mathbf{x} - \mathbf{X}_1, \mathbf{x} - \mathbf{X}_2 \rangle}{d} \xrightarrow{a.s.} \sigma_j^2 \text{ as } d \rightarrow \infty \text{ if } 1 \leq i = j \leq J, \text{ and} \quad (12)$$

$$\frac{\langle \mathbf{x} - \mathbf{X}_1, \mathbf{x} - \mathbf{X}_2 \rangle}{d} \xrightarrow{a.s.} \sigma_j^2 + \nu_{ji} \text{ as } d \rightarrow \infty \text{ if } 1 \leq i \neq j \leq J. \quad (13)$$

Observe now that $\|E_{F_j}[u(\mathbf{x} - \mathbf{X})]\|^2 = \langle E_{F_j}[u(\mathbf{x} - \mathbf{X}_1)], E_{F_j}[u(\mathbf{x} - \mathbf{X}_2)] \rangle = E_{F_j}[u(\mathbf{x} - \mathbf{X}_1), u(\mathbf{x} - \mathbf{X}_2)]$, where $\mathbf{X}_1, \mathbf{X}_2 \sim F_j$ are independent random vectors for $1 \leq j \leq J$.

Since we are dealing with expectations of random vectors with bounded norm, a simple application of DCT implies that for almost every realization \mathbf{x} of $\mathbf{X} \sim F_j$ ($1 \leq j \leq J$), as $d \rightarrow \infty$,

$$\text{SPD}^{\circ}(\mathbf{x}, F_j) \xrightarrow{a.s.} 1 - \sqrt{\frac{1}{2}} \text{ and } \text{SPD}^{\circ}(\mathbf{x}, F_i) \xrightarrow{a.s.} 1 - \sqrt{\frac{\sigma_j^2 + \nu_{ji}}{\sigma_j^2 + \sigma_i^2 + \nu_{ji}}} \text{ for } i \neq j. \quad (14)$$

Thus, for $\mathbf{X} \sim F_j$, we get $z^{\circ}(\mathbf{X}) = (\text{SPD}^{\circ}(\mathbf{X}, F_1), \dots, \text{SPD}^{\circ}(\mathbf{X}, F_J))^T \xrightarrow{a.s.} \mathbf{c}_j$ as $d \rightarrow \infty$. ■

Proof of Theorem 6(b) : Recall that for $h > 1$, $\text{LSPD}_h^{\circ}(\mathbf{x}, F) = E_{F_j}[h^d K_h(\mathbf{t})] - \|E_{F_j}[h^d K_h(\mathbf{t})u(\mathbf{t})]\|$. Since we have assumed \mathbf{X} s to be standardized, here we get $h^d K_h(\mathbf{t}) = K(\mathbf{x} - \mathbf{X})/h = g_0(\|\mathbf{x} - \mathbf{X}\|/h)$. Let $\mathbf{X} \sim F_j$ and $\mathbf{X}_i \sim F_i$ with $1 \leq i, j \leq J$. Using (10) and (11) above, and the continuity of g_0 , for almost every realization \mathbf{x} of $\mathbf{X} \sim F_j$, one obtains the following

$$g_0 \left(\frac{\|\mathbf{x} - \mathbf{X}_i\| \sqrt{d}}{h} \right) \xrightarrow{a.s.} g_0(0) \text{ or } g_0(\epsilon_{ji} A_0),$$

depending on whether $\sqrt{d}/h \rightarrow 0$ or A_0 . The proof follows from an application of DCT, and the arguments used in the proof of Theorem 6(a). ■

Proof of Theorem 6(c) : Since $g_0(s) \rightarrow 0$ as $s \rightarrow \infty$, using the same argument as used in the proof of Theorem 6(b), for $\mathbf{X}_i \sim F_i$ and almost every realization \mathbf{x} of $\mathbf{X} \sim F_j$, we have

$$g_0 \left(\frac{\|\mathbf{x} - \mathbf{X}_i\| \sqrt{d}}{h} \right) \xrightarrow{a.s.} 0 \text{ as } \sqrt{d}/h \rightarrow \infty.$$

The proof now follows from a simple application of DCT. ■

Lemma 11 Recall \mathbf{c}_j and \mathbf{c}'_j for $1 \leq j \leq J$ defined in Theorem 6(a) and (b), respectively. For any $1 \leq j \neq i \leq J$, $\mathbf{c}_j = \mathbf{c}_i$ if and only if $\sigma_j = \sigma_i$ and $\nu_{ji} = \nu_{ij} = 0$. Similarly, $\mathbf{c}'_j = \mathbf{c}'_i$ if and only if $\sigma_j = \sigma_i$ and $\nu_{ji} = \nu_{ij} = 0$.

Proof of Lemma 11 : The 'if' part is easy to check in both cases. So, it is enough to prove the 'only if' part and that too for the case of $J = 2$. If $\mathbf{c}_1 = (c_{11}, c_{12})^T$ and $\mathbf{c}_2 = (c_{21}, c_{22})^T$ are equal, then we have

$$\frac{\sigma_1^2 + \nu_{12}}{\sigma_1^2 + \sigma_2^2 + \nu_{12}} = 1/2 \text{ and } \frac{\sigma_2^2 + \nu_{12}}{\sigma_1^2 + \sigma_2^2 + \nu_{12}} = 1/2.$$

These two equations hold simultaneously only if $\sigma_1^2 = \sigma_2^2$ and $\nu_12 = \nu_21 = 0$.

Consider the case $c_1^1 = c_2^2$. Recall that $c_{11}^1 = g_0(A_0\sqrt{2\sigma_1})/c_{11}$, $c_{22}^2 = g_0(A_0\sqrt{2\sigma_2})/c_{22}$, $c_{12}^2 = g_0(A_0\sqrt{\sigma_1^2 + \sigma_2^2 + \nu_{12}})/c_{12}$ and $c_{21}^1 = g_0(A_0\sqrt{\sigma_2^2 + \sigma_1^2 + \nu_{21}})/c_{21}$. If possible, assume that $\sigma_1 > \sigma_2$. This implies that $A_0\sqrt{\sigma_1^2 + \sigma_2^2 + \nu_{12}} > A_0\sqrt{2\sigma_1}$ and hence we obtain

$$g_0(A_0\sqrt{2\sigma_1}) > g_0(A_0\sqrt{\sigma_1^2 + \sigma_2^2 + \nu_{12}}) \quad (\text{since } g_0 \text{ is monotonically decreasing}). \quad (15)$$

Also, if $\sigma_1 > \sigma_2$, we must have

$$1/2 < \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} < \frac{\sigma_1^2 + \nu_{12}}{\sigma_1^2 + \sigma_2^2 + \nu_{12}} < 1 \Leftrightarrow 1 - \sqrt{1/2} > 1 - \sqrt{\frac{\sigma_1^2 + \nu_{12}}{\sigma_1^2 + \sigma_2^2 + \nu_{12}}}. \quad (16)$$

Combining (15) and (16), we have $c_{11}^1 > c_{21}^1$, and this implies $c_1^1 \neq c_2^2$. Similarly, if $\sigma_1 < \sigma_2$, we get $c_{12}^2 > c_{22}^2$ and hence $c_1^1 \neq c_2^2$. Again, if $\sigma_1 = \sigma_2$ but $\nu_{12} = \nu_{21} > 0$, similar arguments lead to $c_1^1 \neq c_2^2$. This completes the proof. ■

References

- C. Agostinelli and M. Romanazzi. Local depth. *Journal of Statistical Planning and Inference*, **141**:817–830, 2010.
- U. Alon, N. Barkai, D. A. Notterman, K. Gish, D. Mack, and A. J. Leine. Broad pattern of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences, USA*, **96**:6745–6750, 1999.
- A. Azzalini. *The Skew-Normal and Related Families*. 2014. Cambridge University Press, Cambridge.
- Y. Chen, X. Dang, H. Peng, and H. L. Bart Jr. Outlier detection with the kernelized spatial depth function. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **31**:288–305, 2009.
- T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, **13**:21–27, 1967.
- J. A. Cuesta-Albertos, M. Febrero-Bande, and M. Oviedo de la Fuente. The DD^C-classifier in the functional setting. *Test*, forthcoming.
- R. M. de Jong. Laws of large numbers for dependent heterogeneous processes. *Econometric Theory*, **11**:347–358, 1995.
- E. Dimitriadou, K. Hornik, F. Leisch, D. Meyer, and A. Weingessel. e1071: Misc functions of the department of statistics (e1071), TU Wien. R package version 1.5-27, 2011. <http://CRAN.R-project.org/package=e1071>.
- S. Dutta and A. K. Ghosh. On robust classification using projection depth. *Annals of the Institute of Statistical Mathematics*, **64**:657–676, 2012.
- S. Dutta and A. K. Ghosh. On some transformations of high dimension, low sample size data for nearest neighbor classification. *Machine Learning*, **102**:57–83, 2016.
- S. Dutta, A. K. Ghosh, and P. Chaudhuri. Some intriguing properties of Tukey’s half-space depth. *Bernoulli*, **17**:1420–1434, 2011.
- S. Dzeroski and B. Zenko. Is combining classifiers better than selecting the best one? *Machine Learning*, **54**:255–273, 2004.
- K. T. Fang, S. Kotz, and K. W. Ng. *Symmetric Multivariate and Related Distributions*. 1990. Chapman & Hall, London.
- J. Friedman. Another approach to polychotomous classification. Technical report, Dept. of Statistics, Stanford University, 1996. <http://old.cba.uu.edu/~mhardt/poly.pdf>.
- Y. Gao. Data depth based on spatial rank. *Statistics and Probability Letters*, **65**:217–225, 2003.
- A. K. Ghosh and P. Chaudhuri. On data depth and distribution free discriminant analysis using separating surfaces. *Bernoulli*, **11**:1–27, 2005a.
- A. K. Ghosh and P. Chaudhuri. On maximum depth and related classifiers. *Scandinavian Journal of Statistics*, **32**:328–350, 2005b.
- A. K. Ghosh, P. Chaudhuri, and C. A. Murthy. On visualization and aggregation of nearest neighbor classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**:1592–1602, 2005.
- A. K. Ghosh, P. Chaudhuri, and D. Sen Gupta. Classification using kernel density estimates : Multiscale analysis and visualization. *Technometrics*, **48**:120–132, 2006.
- P. Hall, J. S. Marron, and A. Neeman. Geometric representation of high dimension low sample size data. *Journal of the Royal Statistical Society: Series B*, **67**:427–444, 2005.
- T. Hastie and R. Tibshirani. *Generalized Additive Models*. 1990. Chapman & Hall, London.
- T. Hastie and R. Tibshirani. Classification by pairwise coupling. *Annals of Statistics*, **26**:451–471, 1998.
- T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2009. Springer, New York.
- R. Hoberg and K. Mösler. Data analysis and classification with the zonoid depth. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, **72**:45–59, 2006.
- Y. Hu, Y. Wang, Y. Wu, Q. Li, and C. Hou. Generalized Mahalanobis depth in the reproducing kernel Hilbert space. *Statistical Papers*, **52**:511–522, 2011.
- R. Jorntsen. Clustering and classification based on the L₁ data depth. *Journal of Multivariate Analysis*, **90**:67–89, 2004.

- S. Jung and J. S. Marron. PCA consistency in high dimension, low sample size context. *Annals of Statistics*, **37**:4104–4130, 2009.
- J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20**:226–239, 1998.
- D. Kosiorowski and Z. Zawadzki. *DepthProc: An R Package for Robust Exploration of Multidimensional Economic Phenomena*, 2016.
- T. Lange, K. Mosler, and P. Mozharovskiy. Fast nonparametric classification based on data depth. *Statistical Papers*, **55**:49–69, 2014.
- E. Levina and P. J. Bickel. Maximum likelihood estimation of intrinsic dimension. *Advances in Neural Information Processing Systems (NIPS)*, **17**:777–784, 2004. MIT Press, Cambridge, MA.
- J. Li, J. A. Cuesta-Albertos, and R. Liu. Nonparametric classification procedures based on DD-plot. *Journal of the American Statistical Association*, **107**:737–753, 2012.
- R. Z. Li, K. T. Fang, and L. X. Zhu. Some Q-Q probability plots to test spherical and elliptic symmetry. *Journal of Computational and Graphical Statistics*, **6**:435–450, 1997.
- A. Liaw and M. Wiener. Classification and regression by random forest. *R News*, **2**:18–22, 2002.
- R. Liu, J. Parelius, and K. Singh. Multivariate analysis of data depth : Descriptive statistics and inference. *Annals of Statistics*, **27**:783–858, 1999.
- W. S. Lok and S. M. S. Lee. A new statistical depth function with applications to multimodal data. *Journal of Nonparametric Statistics*, **23**:617–631, 2011.
- J. S. Marron, M. J. Todd, and J. Ahn. Distance weighted discrimination. *Journal of the American Statistical Association*, **102**:1267–1271, 2007.
- P. Mozharovskiy, K. Mosler, and T. Lange. Classifying real-world data with the DD^α procedure. *Advances in Data Analysis and Classification*, **9**:287–314, 2015.
- D. Paindaveine and G. Van Bever. From depth to local depth : a focus on centrality. *Journal of the American Statistical Association*, **105**:1105–1119, 2013.
- D. Paindaveine and G. Van Bever. Nonparametrically consistent depth-based classifiers. *Bernoulli*, **21**:62–82, 2015.
- G. M. Reaven and R. G. Miller. An attempt to define the nature of chemical diabetes using a multidimensional analysis. *Diabetologia*, **16**:17–24, 1979.
- B. Ripley. tree: Classification and regression trees. R package version 1.0-29, 2011.
- P. J. Rousseeuw and K. Van Driessen. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, **41**:212–223, 1999.
- J. P. Royston. Some techniques for assessing multivariate normality based on the Shapiro-Wilk W. *Technometrics*, **32**:121–133, 1983.
- D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. 2015. Wiley, Hoboken, New Jersey.
- R. Serfling. A depth function and a scale curve based on spatial quantiles. In *Statistics and Data Analysis based on L_1 -Norm and Related Methods (Y. Dodge ed.)*, pages 25–38, 2002. Birkhaeuser.
- Y. Vardi and C. H. Zhang. The multivariate L_1 -median and associated data depth. *Proceedings of the National Academy of Sciences, USA*, **97**:1423–1426, 2000.
- S. N. Wood. *Generalized Additive Models: An Introduction with R*. 2006. Chapman & Hall/CRC, Boca Raton, FL.
- C. Xia, L. Lin, and G. Yang. An extended projection data depth and its applications to discrimination. *Communication in Statistics - Theory and Methods*, **37**:2276–2290, 2008. Thomas W. Yee. The VGAM package. *R News*, **8**(2):28–39, 2008.
- V. V. Yurinskii. Exponential inequalities for sums of random vectors. *Journal of Multivariate Analysis*, **6**:473–499, 1976.
- Y. Zuo and R. Serfling. General notions of statistical depth function. *Annals of Statistics*, **28**:461–482, 2000.

On Bayes Risk Lower Bounds

Xi Chen

*Stern School of Business
New York University
New York, NY 10012, USA*

XCHEN3@STERN.NYU.EDU

Adityanand Guntuboyina

*Department of Statistics
University of California
Berkeley, CA 94720, USA*

ADITYA@STAT.BERKELEY.EDU

Yuchen Zhang

*Computer Science Department
Stanford University
Stanford, CA 94305, USA*

ZHANGYUC@CS.STANFORD.EDU

Editor: Edo Airoldi

Abstract

This paper provides a general technique for lower bounding the Bayes risk of statistical estimation, applicable to arbitrary loss functions and arbitrary prior distributions. A lower bound on the Bayes risk not only serves as a lower bound on the minimax risk, but also characterizes the fundamental limit of any estimator given the prior knowledge. Our bounds are based on the notion of f -informativity (Csiszár, 1972), which is a function of the underlying class of probability measures and the prior. Application of our bounds requires upper bounds on the f -informativity, thus we derive new upper bounds on f -informativity which often lead to tight Bayes risk lower bounds. Our technique leads to generalizations of a variety of classical minimax bounds (e.g., generalized Fano's inequality). Our Bayes risk lower bounds can be directly applied to several concrete estimation problems, including Gaussian location models, generalized linear models, and principal component analysis for spiked covariance models. To further demonstrate the applications of our Bayes risk lower bounds to machine learning problems, we present two new theoretical results: (1) a precise characterization of the minimax risk of learning spherical Gaussian mixture models under the smoothed analysis framework, and (2) lower bounds for the Bayes risk under a natural prior for both the prediction and estimation errors for high-dimensional sparse linear regression under an improper learning setting.

Keywords: Bayes risk, Minimax risk, f -divergence, f -informativity, Fano's inequality, Smoothed analysis

1. Introduction

Consider a standard setting where we observe data points X taking values in a sample space \mathcal{X} . The distribution of X depends on an unknown parameter $\theta \in \Theta$ and is denoted by P_θ . The goal is to compute an estimate of θ based on the observed samples. Formally, we denote the estimator by $\mathfrak{d}(X)$, where $\mathfrak{d} : \mathcal{X} \rightarrow \Theta$ is a mapping from the sample space to the parameter space. The risk of the estimator is defined by $\mathbb{E}_\theta L(\theta, \mathfrak{d}(X))$ where $L :$

$\Theta \times \mathcal{A} \mapsto [0, \infty)$ is a non-negative loss function. This framework applies to a broad scope of machine learning problems. Taking sparse linear regression as a concrete example, the data X represents the design matrix and the response vector; the parameter space is the set of sparse vectors; the loss function can be chosen as a squared loss.

Given an estimation problem, we are interested in the lowest possible risk achievable by any estimator, which will be useful in justifying the potential of improving existing algorithms. The classical notion of optimality is formalized by the so-called *minimax risk*. More specifically, we assume that the statistician chooses an optimal estimator \mathfrak{d} , then the adversary chooses the worst parameter θ by knowing the choice of \mathfrak{d} . The minimax risk is defined as:

$$R_{\text{minimax}}(L; \Theta) := \inf_{\mathfrak{d}} \sup_{\theta \in \Theta} \mathbb{E}_\theta L(\theta, \mathfrak{d}(X)). \quad (1)$$

The minimax risk has been determined up to multiplicative constants for many important problems. Examples include sparse linear regression (Raskutti et al., 2011), classification (Yang, 1999), additive models over kernel classes (Raskutti et al., 2012), and crowdsourcing (Zhang et al., 2016).

The assumption that the adversary is capable of choosing a worst-case parameter is sometimes over-pessimistic. In practice, the parameter that incurs a worst-case risk may appear with very small probability. To capture the hardness of the problem with this prior knowledge, it is reasonable to assume that the true parameter is sampled from an underlying prior distribution w . In this case, we are interested in the *Bayes risk* of the problem. That is, the lowest possible risk when the true parameter is sampled from the prior distribution:

$$R_{\text{Bayes}}(w, L; \Theta) := \inf_{\mathfrak{d}} \int_{\Theta} \mathbb{E}_\theta L(\theta, \mathfrak{d}(X)) w(d\theta). \quad (2)$$

If the prior distribution w is known to the learner, then the Bayes estimator attains the Bayes risk (Berger, 2013). But in general, the Bayes estimator is computationally hard to evaluate, and the Bayes risk has no closed-form expression. It is thus unclear what is the fundamental limit of estimators when the prior knowledge is available.

In this paper, we present a technique for establishing lower bounds on the Bayes risk for a general prior distribution w . When the lower bound matches the risk of any existing algorithm, it captures the convergence rate of the Bayes risk. The Bayes risk lower bounds are useful for three main reasons:

1. They provide an idea of the difficulty of the problem under a specific prior w .
2. They automatically provide lower bounds for the minimax risk and, because the minimax regret is always larger than or equal to the minimax risk (see, for example, Rakhlin et al. (2013)), they also yield lower bounds for the minimax regret.
3. As we will show, they have an important application in establishing the minimax lower bound under the *smoothed analysis framework*.

Throughout this paper, when the loss function L and the parameter space Θ are clear from the context, we simply denote the Bayes risk by $R_{\text{Bayes}}(w)$. When the prior w is also clear, the notation is further simplified to R .

1.1 Our Main Results

In order to give the reader a flavor of the kind of results proved in this paper, let us consider Fano’s classical inequality (Han and Verdú, 1994; Cover and Thomas, 2006; Yu, 1997) which is one of the most widely used Bayes risk lower bounds in statistics and information theory. The standard version of Fano’s inequality applies to the case when $\Theta = \mathcal{A} = \{1, \dots, N\}$ for some positive integer N with the indicator loss $L(\theta, a) := \mathbb{I}\{\theta \neq a\}$ (\mathbb{I} stands for the zero-one valued indicator function) and the prior w being the discrete uniform distribution on Θ . In this setting, Fano’s inequality states that

$$R_{\text{Bayes}}(w) \geq 1 - \frac{I(w; \mathcal{P}) + \log 2}{\log N} \quad (3)$$

where $I(w; \mathcal{P})$ is the mutual information between the random variables $\theta \sim w$ and X with $X|\theta \sim P_\theta$ (note that this mutual information only depends on w and $\mathcal{P} = \{P_\theta : \theta \in \Theta\}$ which is why we denote it by $I(w; \mathcal{P})$). Fano’s inequality implies that when $I(w; \mathcal{P})$ is large i.e., when the information that X has about θ is large, then the risk of estimation is small.

A natural question regarding Fano’s inequality, which does not seem to have been asked until very recently, is the following: does there exist an analogue of (3) when w is not necessarily the uniform prior and/or when Θ and \mathcal{A} are arbitrary sets, and/or when the loss function is not necessarily $\mathbb{I}\{\theta \neq a\}$? An interesting result in this direction is the following inequality which has been recently proved by Duchi and Wainwright (2013) who termed it the continuum Fano inequality. This inequality applies to the case when $\Theta = \mathcal{A}$ is a subset of Euclidean space with finite strictly positive Lebesgue measure, $L(\theta, a) = \mathbb{I}\{\|\theta - a\|_2 \geq \epsilon\}$ for a fixed $\epsilon > 0$ ($\|\cdot\|_2$ is the usual Euclidean metric) and the prior w being the uniform probability measure (i.e., normalized Lebesgue measure) on Θ . In this setting, Duchi and Wainwright (2013) proved that

$$R_{\text{Bayes}}(w) \geq 1 + \frac{I(w; \mathcal{P}) + \log 2}{\log(\sup_{a \in \mathcal{A}} w\{\theta \in \Theta : \|\theta - a\|_2 < \epsilon\})}. \quad (4)$$

It turns out that there is a very clean connection between inequalities (3) and (4). Indeed, both these inequalities are special instances of the following inequality:

$$R_{\text{Bayes}}(w) \geq 1 + \frac{I(w; \mathcal{P}) + \log 2}{\log(\sup_{a \in \mathcal{A}} w\{\theta \in \Theta : L(\theta, a) = 0\})} \quad (5)$$

Indeed, the term $w\{\theta \in \Theta : L(\theta, a) = 0\}$ equal to $1/N$ in the setting of (3) and it is equal to $w\{\theta \in \Theta : \|\theta - a\|_2 < \epsilon\}$ in the setting of (4).

Since both (3) and (4) are special instances of (5), one might reasonably conjecture that proving that inequality (5) might hold more generally. In Section 3, we give an affirmative answer by proving that inequality (5) holds for any zero-one valued loss function L and any prior w . No assumptions on Θ , \mathcal{A} and w are needed. We refer to this result as *generalized Fano’s inequality*. Our proof of (5) is quite succinct and is based on the data processing inequality (Cover and Thomas, 2006; Liese, 2012) for Kullback-Leibler (KL) divergence. The use of the data processing inequality for proving Fano-type inequalities was introduced by Gushchin (2003).

The data processing inequality is not only available for the KL divergence. It can be generalized to any divergence belonging to a general family known as f -divergences (Csiszár, 1963; Ali and Silvey, 1966). This family includes the KL divergence, chi-squared divergence, squared Hellinger distance, total variation distance and power divergences as special cases. The usefulness of f -divergences in machine learning has been illustrated in Reid and Williamson (2011); Garcia-Garcia and Williamson (2012); Reid and Williamson (2009).

For every f -divergence, one can define a quantity called f -informativity (Csiszár, 1972) which plays the same role as the mutual information for KL divergence. The precise definitions of f -divergences and f -informativities are given in Section 2. Utilizing the data processing inequality for f -divergence, we prove general Bayes risk lower bounds which hold for every zero-one valued loss L and for arbitrary Θ , \mathcal{A} and w (Theorem 2). The generalized Fano’s inequality (5) is a special case by choosing the f -divergence to be KL. The proposed Bayes risk lower bounds can also be specialized to other f -divergences and have a variety of interesting connections to existing lower bounds in the literature such as Le Cam’s inequality, Assouad’s lemma (see Theorem 2.12 in Tsybakov (2010)), Birgé-Gushchin inequality (Gushchin, 2003; Birgé, 2005). These results are provided in Section 3.

In Section 4, we deal with nonnegative valued loss functions L which are not necessarily zero-one valued. Basically, we use the standard method of lower bounding the general loss function L by a zero-one valued function and then use our results from Section 3 for lower bounding the Bayes risk. This technique, in conjunction with the generalized Fano’s inequality, gives the following lower bound (proved in Corollary 12)

$$R_{\text{Bayes}}(w, L; \Theta) \geq \frac{1}{2} \sup_{a \in \mathcal{A}} \left\{ t > 0 : \sup_{a \in \mathcal{A}} w\{\theta : L(\theta, a) < t\} \leq \frac{1}{4} e^{-2I(w; \mathcal{P})} \right\}. \quad (6)$$

A special case of the above inequality has appeared previously in Zhang (2006, Theorem 6.1) (please refer to Remark 13 for a detailed explanation of the connection between inequality (6) and (Zhang, 2006, Theorem 6.1)).

We also prove analogues of the above inequality for different f divergences. Specifically, using our f -divergence inequalities from Section 3, we prove, in Theorem 9, the following inequality which holds for every f divergence:

$$R_{\text{Bayes}}(w, L; \Theta) \geq \frac{1}{2} \sup_{a \in \mathcal{A}} \left\{ t > 0 : \sup_{a \in \mathcal{A}} w\{\theta : L(\theta, a) < t\} < 1 - u_f(I(w; \mathcal{P})) \right\} \quad (7)$$

where $I_f(w; \mathcal{P})$ represents the f -informativity and $u_f(\cdot)$ is a non-decreasing $[0, 1]$ -valued function that depends only on f . This function $u_f(\cdot)$ (see its definition from (31)) can be explicitly computed for many f -divergences of interest, which gives useful lower bounds in terms of f -informativity. For example, for the case of KL divergence and chi-squared divergence, inequality (7) gives the lower bound in (6) and the following inequality respectively,

$$R_{\text{Bayes}}(w, L; \Theta) \geq \frac{1}{2} \sup_{a \in \mathcal{A}} \left\{ t > 0 : \sup_{a \in \mathcal{A}} w\{\theta : L(\theta, a) < t\} \leq \frac{1}{4(1 + I_{\chi^2}(w; \mathcal{P}))} \right\}. \quad (8)$$

where $I_{\chi^2}(w; \mathcal{P})$ is the chi-squared informativity.

Intuitively, inequality (7) shows that the Bayes risk is lower bounded by half of the largest possible t such that the maximum prior mass of any t -radius “ball” ($w\{\theta : L(\theta, a) < t\}$) is

less than some function of f -informativity. To apply (7), one needs to obtain upper bounds on the following two quantities:

1. The “small ball probability” $\sup_{\theta \in \mathcal{A}} w\{\theta : L(\theta, a) < t\}$, which does not depend of the family of probability measures \mathcal{P} .
2. The f -informativity $I_f(w, \mathcal{P})$, which does not depend on the loss function L .

We note that a nice feature of (7) is that L and \mathcal{P} play separately roles. One may first obtain an upper bound I_f^{up} for the f -informativity $I_f(w, \mathcal{P})$, then choose t so that the small ball probability $w\{\theta : L(\theta, a) < t\}$ can be bounded from above by $1 - u_f(I_f^{\text{up}})$. The Bayes risk will be bounded from below by $t/2$. It is noteworthy that the terminology “small ball probability” was used by Xu and Raginsky (2014) (this paper proved information-theoretic lower bounds on the minimum time in a distributed function computation problem).

We do not have a general guideline for bounding the small ball probability. It needs to be dealt with case by case based on the prior and the loss function. But for upper bounding the f -informativity, we offer a general recipe in Section 5 for a subclass of divergences of interest (power divergences for $\alpha \notin [0, 1)$), which covers the chi-squared divergence as one of the most important divergences in our applications. These bounds generalize results of Hausser and Opper (1997) and Yang and Barron (1999) for mutual information to f -informativities involving power divergences. As an illustration of our techniques (inequality (7) combined with the f -informativity upper bounds), we apply them to a concrete estimation problem in Section 5. We further apply our results to several popular machine learning and statistics problems (e.g., generalized linear model, spiked covariance model, and Gaussian model with general loss) in Appendix C.

In Section 6 and Section 7, we present non-trivial applications of our Bayes risk lower bounds to two learning problems: the first one is a unsupervised learning problem, while the second one is a supervised learning problem. Section 6 studies smoothed analysis for learning mixtures of spherical Gaussians with uniform weights. Although learning mixtures of Gaussians is a computationally hard problem, it has been shown recently by Hsu and Kakade (2013) that under the assumptions that the Gaussian means are linearly independent, it can be learnt in polynomial time by a spectral method. We perform a smoothed analysis on a variant of the algorithm (Hsu and Kakade, 2013), showing that the linear independence assumption can be replaced by perturbing the true parameters by a small random noise. The method described in Section 6 achieves a better convergence rate than the original algorithm of Hsu and Kakade (2013). Furthermore, we apply the Bayes risk lower bound techniques to show that the algorithm’s convergence rate is unimprovable, even under smoothed analysis (i.e. when the true parameters are randomly perturbed). Section 6 highlights the usefulness of our techniques in proving lower bounds for smoothed analysis, which appears to be challenging using traditional techniques of the minimax theory.

In Section 7, we consider the high-dimensional sparse linear regression problem and we provide Bayes risk lower bounds for both prediction error and estimation error under a natural prior on the regression parameter belonging to the set of k -sparse vectors. Although lower bounds for sparse linear regression have been well-studied (see, e.g., Raskutti et al. (2011); Zhang et al. (2014) and references therein), these bounds only focus on the minimax or the worst-case scenario and thus are too pessimistic in practice. Indeed, the parameters

that usually attain these minimax lower bounds have zero probability under any continuous prior, so that their average effects might be negligible. The fundamental limits of sparse linear regression under a realistic prior is, to the best of the our knowledge, unknown. The developed tool of lower bounding Bayes risks can be directly applied to characterize these limits. Moreover, our Bayes risk lower bound is flexible in the sense that by tuning the variance of the prior of non-zero elements of θ , it provides a wide spectrum of lower bounds. For one particular choice of the variance, our Bayes risk lower bounds match the minimax risk lower bounds. This gives a natural *least favorable prior* for sparse linear regression, while the known least favorable prior in Raskutti et al. (2011) is a non-constructive discrete prior over a packing set of the parameter space that cannot be sampled from. We also work under the *improper learning* setting where we allow non-sparse estimators for the true regression vector (even though the true regression vector is assumed to be sparse).

1.2 Related Works

Before finishing this introduction section, we briefly describe related work on Bayes risk lower bounds. There are a few results dealing with special cases of finite dimensional estimation problems under (weighted/truncated) quadratic losses. The first results of this kind were established by Van Trees (1968), and Borovkov and Sakhanienko (1980) with extensions by Brown and Gajek (1990); Brown (1993); Gill and Levit (1995); Sato and Akahira (1996); Takada (1999). A few additional papers dealt with even more specialized problems e.g., Gaussian white noise model (Brown and Liu, 1993), scale models (Gajek and Kaluszka, 1994) and estimating Gaussian variance (Vidakovi and DasGupta, 1995). Most of these results are based on the van Trees inequality (see Gill and Levit (1995) and Theorem 2.13 in Tsybakov (2010)). Although the van Trees inequality usually leads to sharp constant in the Bayes risk lower bounds, it only applies to weighted quadratic loss functions (as its proof relies on Cauchy-Schwarz inequality) and requires the underlying Fisher information to be easily computable, which limits its applicability. There is also a vast body of literature on minimax lower bounds (see, e.g., Tsybakov (2010)) which can be viewed as Bayes risk lower bounds for certain priors. These priors are usually discrete and specially constructed so that the lower bounds do not apply to more general (continuous) priors. Another related area of work involves finding lower bounds on posterior contraction rates (see, e.g., Castillo (2008)).

1.3 Outline of the Paper

The rest of the paper is organized in the following way. In Section 2, we describe notations and review preliminaries such as f -divergences, f -informativity, data processing inequality, etc. Section 3 deals with inequalities for zero-one valued loss functions. These inequalities have many connections to existing lower bound techniques. Section 4 deals with nonnegative loss functions and we provide inequality (7) and its special cases. Section 5 presents upper bounds on the f -informativity for power divergences for $\alpha \notin [0, 1)$. Some examples are given in this section. Section 6 studies smoothed analysis for learning mixtures of spherical Gaussians with uniform weights using our technique. We conclude the paper in Section 1.3. Due to space constraints, we have relegated some proofs and additional examples and results to the appendix.

2. Preliminaries and Notations

We first review the notions of f -divergence (Csiszár, 1963; Ali and Silvey, 1966) and f -informativity (Csiszár, 1972). Let \mathcal{C} denote the class of all convex functions $f : (0, \infty) \rightarrow \mathbb{R}$ which satisfy $f(1) = 0$. Because of convexity, the limits $f(0) := \lim_{x \downarrow 0} f(x)$ and $f'(\infty) := \lim_{x \uparrow \infty} f(x)/x$ exist (even though they may be $+\infty$) for each $f \in \mathcal{C}$. Each function $f \in \mathcal{C}$ defines a divergence between probability measures which is referred to as f -divergence. For two probability measures P and Q on a sample space having densities p and q with respect to a common measure μ , the f -divergence $D_f(P\|Q)$ between P and Q is defined as follows:

$$D_f(P\|Q) := \int f\left(\frac{p}{q}\right) q d\mu + f'(\infty)P\{q=0\}. \quad (9)$$

We note that the convention $0 \cdot \infty = 0$ is adopted here so that $f'(\infty)P\{q=0\} = 0$ when $f'(\infty) = \infty$ and $P\{q=0\} = 0$. Note that $D_f(P\|Q) = +\infty$ when $f'(\infty) = +\infty$ and $P\{q=0\} > 0$. Also note that $f(1) = 0$ implies that $D_f(P\|Q) = 0$ when $P = Q$.

Certain divergences are commonly used because they can be easily computed or bounded when P and Q are product measures. These divergences are the power divergences corresponding to the functions f_α defined by

$$f_\alpha(x) = \begin{cases} x^\alpha - 1 & \text{for } \alpha \notin [0, 1]; \\ 1 - x^\alpha & \text{for } \alpha \in (0, 1); \\ x \log x & \text{for } \alpha = 1; \\ -\log x & \text{for } \alpha = 0. \end{cases}$$

Popular examples of power divergences include:

1) Kullback-Leibler (KL) divergence: $\alpha = 1$, $D_H(P\|Q) = \int p \log(p/q) d\mu$ if P is absolutely continuous with respect to Q (and it is infinite if P is not absolutely continuous with respect to Q). Following the conventional notation, we denote the KL divergence by $D(P\|Q)$ (instead of $D_H(P\|Q)$).

2) Chi-squared divergence: $\alpha = 2$, $D_H(P\|Q) = \int (p^2/q) d\mu - 1$ if P is absolutely continuous with respect to Q (and it is infinite if P is not absolutely continuous with respect to Q). We denote the chi-squared divergence by $\chi^2(P\|Q)$ following the conventional notation.

3) When $\alpha = 1/2$, one has $D_{H_{1/2}}(P\|Q) = 1 - \int \sqrt{pq} d\mu$ which is a half of the squared Hellinger distance. That is, $D_{H_{1/2}}(P\|Q) = H^2(P\|Q)/2$, where $H^2(P\|Q) = \int (\sqrt{p} - \sqrt{q})^2 d\mu$ is the squared Hellinger distance between P and Q .

The total variation distance $\|P - Q\|_{TV}$ is another f -divergence (with $f(x) = |x - 1|/2$) but not a power divergence.

One of the most important properties of f -divergences is the ‘‘data processing inequality’’ (Csiszár (1972) and Liese (2012, Theorem 3.1)) which states the following: let \mathcal{X} and \mathcal{Y} be two measurable spaces and let $\Gamma : \mathcal{X} \rightarrow \mathcal{Y}$ be a measurable function. For every $f \in \mathcal{C}$ and every pair of probability measures P and Q on \mathcal{X} , we have

$$D_f(P\Gamma^{-1}\|Q\Gamma^{-1}) \leq D_f(P\|Q), \quad (10)$$

where $P\Gamma^{-1}$ and $Q\Gamma^{-1}$ denote the *induced measures* of Γ on \mathcal{Y} , i.e., for any measurable set B on the space \mathcal{Y} , $P\Gamma^{-1}(B) := P(\Gamma^{-1}(B))$, $Q\Gamma^{-1}(B) := Q(\Gamma^{-1}(B))$ (see the definition of induced measure from Definition 2.2.1. in Athreya and Lahiri (2006)).

Next, we introduce the notion of f -informativity (Csiszár, 1972). Let $\mathcal{P} = \{P_\theta : \theta \in \Theta\}$ be a family of probability measures on a space \mathcal{X} and w be a probability measure on Θ . For each $f \in \mathcal{C}$, the f -informativity, $I_f(w; \mathcal{P})$, is defined as

$$I_f(w; \mathcal{P}) = \inf_Q \int D_f(P_\theta\|Q) w(d\theta), \quad (11)$$

where the infimum is taken over all possible probability measures Q on \mathcal{X} . When $f(x) = x \log x$ (so that the corresponding f -divergence is the KL divergence^(e)), the f -informativity is equal to the mutual information and is denoted by $I(w; \mathcal{P})$. We denote the informativity corresponding to the power divergence D_{f_α} by $I_{f_\alpha}(w; \mathcal{P})$. For the special case $\alpha = 2$, we use the more suggestive notation $I_{\chi^2}(w; \mathcal{P})$. The informativity corresponding to the total variation distance will be denoted by $I_{TV}(w; \mathcal{P})$.

Additional notations and definitions are described as follows. Recall the Bayes risk (2) and the minimax risk (1). When the loss function L and parameter space Θ are clear from the context, we drop the dependence on L and Θ . When the prior w is also clear from the context, we denote the Bayes risk by R and the minimax risk by $R_{\min\max}$. We need certain notation for covering numbers. For a given f -divergence and a subset $S \subset \Theta$, let $M_f(\epsilon; S)$ denote any upper bound on the smallest number M for which there exist probability measures Q_1, \dots, Q_M that form an ϵ^2 -cover of $\{P_\theta, \theta \in S\}$ under the f -divergence i.e.,

$$\sup_{\theta \in S} \min_{1 \leq j \leq M} D_f(P_\theta\|Q_j) \leq \epsilon^2. \quad (12)$$

We write the covering number as $M_{KL}(\epsilon; S)$ when $f(x) = x \log x$ and $M_{\chi^2}(\epsilon; S)$ when $f(x) = x^2 - 1$. We write $M_A(\epsilon; S)$ when $f = f_\alpha$ for other $\alpha \in \mathbb{R}$. We note that $\log M_f(\epsilon; S)$ is an upper bound on the metric entropy. The quantity $M_f(\epsilon; S)$ can be infinite if S is arbitrary. For a vector $x = (x_1, \dots, x_d)$ and a real number $p \geq 1$, denote by $\|x\|_p$ the l_p -norm of x . In particular, $\|x\|_2$ denotes the Euclidean norm of x . $\mathbb{I}(A)$ denotes the indicator function which takes value 1 when A is true and 0 otherwise. We use C, c , etc. to denote generic constants whose values might change from place to place.

3. Bayes Risk Lower Bounds for Zero-one Valued Loss Functions and Their Applications

In this section, we consider zero-one loss functions L and present a principled approach to derive Bayes risk lower bounds involving f -informativity for every $f \in \mathcal{C}$. Our results hold for any given prior w and zero-one loss L . By specializing the f -divergence to KL divergence, we obtain the generalized Fano’s inequality (5). When specializing to other f -divergences, our bounds lead to some classical minimax bounds of Le Cam and Assouad (Assouad, 1983), more recent minimax results of Gushchin (2003); Birgé (2005) and also results in Tsybakov (2010, Chapter 2). Bayes risk lower bounds for general nonnegative loss functions will be presented in the next section.

We need additional notations to state the main results of this section. For each $f \in \mathcal{C}$, let $\phi_f : [0, 1]^2 \rightarrow \mathbb{R}$ be the function defined in the following way: for $a, b \in [0, 1]^2$, $\phi_f(a, b)$ is the

f -divergence between the two probability measures P and Q on $\{0, 1\}$ given by $P\{1\} = a$ and $Q\{1\} = b$. By the definition (9), it is easy to see that $\phi_f(a, b)$ has the following expression (recall that $f'(\infty) := \lim_{x \rightarrow \infty} f(x)/x$):

$$\phi_f(a, b) = \begin{cases} bf\left(\frac{a}{b}\right) + (1-b)f\left(\frac{1-a}{1-b}\right) & \text{for } 0 < b < 1; \\ f(1-a) + af'(\infty) & \text{for } b = 0; \\ f(a) + (1-a)f'(\infty) & \text{for } b = 1. \end{cases} \quad (13)$$

The convexity of f implies monotonicity and convexity properties of ϕ_f , which is stated in the following lemma.

Lemma 1 *For each $f \in \mathcal{C}$, for every fixed b , the map $g(a) : a \mapsto \phi_f(a, b)$ is non-increasing for $a \in [0, b]$ and $g(a)$ is convex and continuous in a . Further, for every fixed a , the map $h(b) : b \mapsto \phi_f(a, b)$ is non-decreasing for $b \in [a, 1]$.*

We also define the quantity

$$R_0 := \inf_{a \in \mathcal{A}} \int_{\Theta} L(\theta, a) w(d\theta), \quad (14)$$

where the decision a does not depend on data X . Note that R_0 represents the Bayes risk with respect to w in the “no data” problem i.e., when one only has information on Θ , \mathcal{A} , L and the prior w but not the data X . For simplicity, our notation for R_0 suppresses its dependence on w . Because the loss function is zero-one valued so that $L(\theta, a) = 1 - \mathbb{I}(L(\theta, a) = 0)$, the quantity R_0 has the following alternative expression:

$$R_0 = 1 - \sup_{a \in \mathcal{A}} w(B(a)), \quad (15)$$

where

$$B(a) := \{\theta \in \Theta : L(\theta, a) = 0\}, \quad (16)$$

and $w(B(a))$ is the prior mass of the “ball” $B(a)$. It will be important in the sequel to observe that the Bayes risk, $R_{\text{Bayes}}(w)$ is bounded from above by R_0 . This is obvious because the risk with some data cannot be greater than the risk in the no data problem (which can be viewed as an application of the data processing inequality). Formally, if $\mathcal{D} = \{\emptyset : \exists a \in \mathcal{A} \text{ such that } \mathfrak{d}(x) = a \forall x \in \mathcal{X}\}$ is the class of the constant decision rules, then $R_0 = \inf_{\mathfrak{d} \in \mathcal{D}} \int_{\Theta} \mathbb{E}_{\theta} L(\theta, \mathfrak{d}(X)) w(d\theta) \geq R_{\text{Bayes}}(w)$. Because $0 \leq R_{\text{Bayes}}(w) \leq R_0$, we have $R_{\text{Bayes}}(w) = 0$ when $R_0 = 0$. We shall therefore assume throughout this section that $R_0 > 0$.

The main result of this section is presented next. It provides an implicit lower bound for the Bayes risk in terms of R_0 and the f -informativity $I_f(w, \mathcal{P})$ for every $f \in \mathcal{C}$. The only assumption is that L is zero-one valued and we do not assume the existence of the Bayes decision rule.

Theorem 2 *Suppose that the loss function L is zero-one valued. For any $f \in \mathcal{C}$, we have*

$$I_f(w, \mathcal{P}) \geq \phi_f(R_{\text{Bayes}}(w), R_0) \quad (17)$$

where ϕ_f and R_0 are defined (13) and (14) respectively.

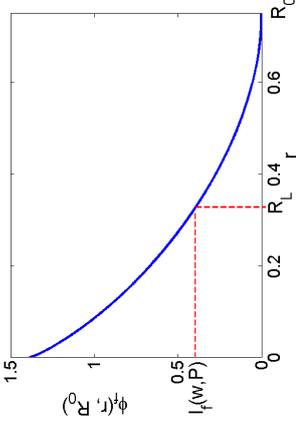


Figure 1: Illustration on why (17) leads to a lower bound on $R_{\text{Bayes}}(w)$. Recall that $R \leq R_0$ and $r \mapsto \phi_f(r, R_0)$ is non-increasing in r for $r \in [0, R_0]$. Given $I_f(w, \mathcal{P})$ as an upper bound of $\phi_f(R_{\text{Bayes}}(w), R_0)$, we have $R_{\text{Bayes}}(w) \geq R_L = g^{-1}(I_f(w, \mathcal{P}))$ and thus R_L serves as a Bayes risk lower bound.

Before we prove Theorem 2, we first show that the inequality (17) indeed provides an implicit lower bound for the Bayes risk $R := R_{\text{Bayes}}(w)$ since $R \leq R_0$ and $r \mapsto \phi_f(r, R_0)$ is non-increasing in r for $r \in [0, R_0]$ (Lemma 1). Therefore, let $g(r) := \phi_f(r, R_0)$. We have

$$R_{\text{Bayes}}(w) \geq g^{-1}(I_f(w, \mathcal{P})), \quad (18)$$

where $g^{-1}(x) := \inf\{0 \leq r \leq R_0, g(r) \leq x\}$ is the generalized inverse function of the non-increasing $g(r)$. As an illustration, we plot $\phi_f(r, R_0)$ for $f(x) = x \log x$ and the corresponding Bayes risk lower bound $g^{-1}(I_f(w, \mathcal{P}))$ in Figure 1. The lower bound (18) can be immediately applied to obtain Bayes risk lower bounds when the f -divergence in (17) is chi-squared divergence, total variation distance, or Hellinger distance (see Corollary 7). However, for the KL divergence, there is no simple form of $g^{-1}(x)$. To obtain the corresponding Bayes risk lower bound, we can invert (17) by utilizing the convexity of $g(r)$, which will give a generalized Fano’s inequality (see Corollary 5). In particular, since $r \mapsto \phi_f(r, R_0)$ is convex (see Lemma 1),

$$\phi_f(R, R_0) \geq \phi_f(r, R_0) + \phi_f'(r-, R_0)(R - r) \quad \text{for every } 0 < r \leq R_0$$

where $\phi_f'(r-, R_0)$ denotes the left derivative of $x \mapsto \phi_f(x, R_0)$ at $x = r$. The monotonicity of $\phi_f(r, R_0)$ in r (Lemma 1) gives $\phi_f'(r-, R_0) \leq 0$ and we thus have,

$$R \geq r + \frac{\phi_f(R, R_0) - \phi_f(r, R_0)}{\phi_f'(r-, R_0)} \quad \text{for every } 0 < r \leq R_0.$$

Inequality (17) $I_f(w, \mathcal{P}) \geq \phi_f(R, R_0)$ can now be used to deduce that (note that $\phi_f'(r-, R_0) \leq 0$)

$$R \geq r + \frac{I_f(w, \mathcal{P}) - \phi_f(r, R_0)}{\phi_f'(r-, R_0)} \quad \text{for every } 0 < r \leq R_0. \quad (19)$$

The inequalities (18) and (19) provide general approaches to convert (17) to an explicit lower bound on R .

Theorem 2 is new, but its special case $\Theta = \mathcal{A} = \{1, \dots, N\}$, $L(\theta, a) := \mathbb{I}(\theta \neq a)$ and the uniform prior w is known (see Gushchin (2003) and Guntuboyina (2011b)). In such a discrete setting, $w(B(a)) = 1/N$ for any $a \in \mathcal{A}$ and thus $R_0 = 1 - 1/N$. The proof of Theorem 2 heavily relies on the following lemma, which is a consequence of the data processing inequality for f -divergences (see (10) in Section 2).

Lemma 3 *Suppose that the loss function L is zero-one valued. For every $f \in \mathcal{C}$, every probability measure Q on \mathcal{X} and every decision rule \mathfrak{d} , we have*

$$\int_{\Theta} D_f(P_\theta \| Q) w(d\theta) \geq \phi_f(R^\theta, R_Q^{\mathfrak{d}}) \quad (20)$$

where

$$R^\theta := \int_{\Theta} \mathbb{E}_\theta L(\theta, \mathfrak{d}(X)) w(d\theta), \quad R_Q^{\mathfrak{d}} := \int_{\mathcal{X}} \int_{\Theta} L(\theta, \mathfrak{d}(x)) w(d\theta) Q(dx). \quad (21)$$

We note that Lemma 3 is of independent interest, which can be applied to establish minimax lower bound as shown in the following remark.

Proof [Proof of Lemma 3]

Let \mathbb{P} denote the joint distribution of θ and X under the prior w i.e., $\theta \sim w$ and $X|\theta \sim P_\theta$. For any decision rule \mathfrak{d} , R^θ in (21) can be written as $R^\theta = \mathbb{E}_{\mathbb{P}} L(\theta, \mathfrak{d}(X))$. Let \mathbb{Q} denote the joint distribution of θ and X under which they are independently distributed according to $\theta \sim w$ and $X \sim Q$ respectively. The quantity $R_Q^{\mathfrak{d}}$ in (21) can then be written as $R_Q^{\mathfrak{d}} = \mathbb{E}_{\mathbb{Q}} L(\theta, \mathfrak{d}(X))$.

Because the loss function is zero-one valued, the function $\Gamma(\theta, x) := L(\theta, \mathfrak{d}(x))$ maps $\Theta \times \mathcal{X}$ into $\{0, 1\}$. Our strategy is to fix $f \in \mathcal{C}$ and apply the data processing inequality (10) to the probability measures \mathbb{P} , \mathbb{Q} and the mapping Γ . This gives

$$D_f(\mathbb{P} \| \mathbb{Q}) \geq D_f(\mathbb{P}^{-1} \| \mathbb{Q}^{-1}), \quad (22)$$

where \mathbb{P}^{-1} and \mathbb{Q}^{-1} are induced measures on the space $\{0, 1\}$ of Γ . In other words, since L is zero-one valued, both \mathbb{P}^{-1} and \mathbb{Q}^{-1} are two-point distributions on $\{0, 1\}$ with

$$\mathbb{P}^{-1}\{1\} = \int_{\Gamma} d\mathbb{P} = \mathbb{E}_{\mathbb{P}} L(\theta, \mathfrak{d}(X)) = R^\theta, \quad \mathbb{Q}^{-1}\{1\} = \int_{\Gamma} d\mathbb{Q} = R_Q^{\mathfrak{d}}.$$

By the definition of the function $\phi_f(\cdot, \cdot)$, it follows that $D_f(\mathbb{P}^{-1} \| \mathbb{Q}^{-1}) = \phi_f(R^\theta, R_Q^{\mathfrak{d}})$. It is also easy to see $D_f(\mathbb{P} \| \mathbb{Q}) = \int_{\Theta} D_f(P_\theta \| Q) w(d\theta)$. Combining this equation with inequality (22) establishes inequality (20). ■

With Lemma 3 in place, we are ready to prove Theorem 2.

Proof [Proof of Theorem 2]

We write R as a shorthand notation of $R_{\text{Bayes}}(w)$. By the definition (11) of $I_f(w, \mathcal{P})$, it suffices to prove that

$$\int D_f(P_\theta \| Q) w(d\theta) \geq \phi_f(R, R_0) \quad (23)$$

for every probability measure Q .

Notice that $R \leq R_0$. If $R = R_0$, then the right hand side of (17) is zero and hence the inequality immediately holds. Assume that $R < R_0$. Let $\epsilon > 0$ be small enough so that $R + \epsilon < R_0$. Let \mathfrak{d} denote any decision rule for which $R \leq R^\theta < R + \epsilon$ and note that such a rule exists since $R = \inf_{\theta \in \mathcal{A}} R^\theta$. It is easy to see that

$$R_Q^{\mathfrak{d}} = \int_{\mathcal{X}} \int_{\Theta} L(\theta, \mathfrak{d}(x)) w(d\theta) Q(dx) \geq \int_{\mathcal{X}} \left(\inf_{a \in \mathcal{A}} \int_{\Theta} L(\theta, a) w(d\theta) \right) Q(dx) = R_0.$$

We thus have $R \leq R^\theta < R + \epsilon < R_0 \leq R_Q^{\mathfrak{d}}$. By Lemma 3, we have

$$\int_{\Theta} D_f(P_\theta \| Q) w(d\theta) \geq \phi_f(R^\theta, R_Q^{\mathfrak{d}}).$$

Because $x \mapsto \phi_f(x, R_Q^{\mathfrak{d}})$ is non-increasing on $x \in [0, R_Q^{\mathfrak{d}}]$, we have

$$\phi_f(R^\theta, R_Q^{\mathfrak{d}}) \geq \phi_f(R + \epsilon, R_Q^{\mathfrak{d}}).$$

Because $x \mapsto \phi_f(R + \epsilon, x)$ is non-decreasing on $x \in [R + \epsilon, 1]$, we have

$$\phi_f(R + \epsilon, R_Q^{\mathfrak{d}}) \geq \phi_f(R + \epsilon, R_0).$$

Combining the above three inequalities, we have

$$\int_{\Theta} D_f(P_\theta \| Q) w(d\theta) \geq \phi_f(R^\theta, R_Q^{\mathfrak{d}}) \geq \phi_f(R + \epsilon, R_Q^{\mathfrak{d}}) \geq \phi_f(R + \epsilon, R_0).$$

The proof of (23) completes by letting $\epsilon \downarrow 0$ and using the continuity of $\phi_f(\cdot, R_0)$ (continuity was noted in Lemma 1). This completes the proof of Theorem 2. ■

Remark 4 *Lemma 3 can also be used to derive minimax lower bounds in a different way. For example, when the minimax decision rule \mathfrak{d} exists (e.g., for finite space Θ and \mathcal{A} (Ferguson, 1967)), we have $R^\theta \leq R_{\text{minimax}}$. If the probability measure Q is chosen so that $R_{\text{minimax}} \leq R_Q^{\mathfrak{d}}$, then, by Lemma 1, the right hand side of (17) can be lower bounded by replacing R^θ with R_{minimax} which yields*

$$\int_{\Theta} D_f(P_\theta \| Q) w(d\theta) \geq \phi_f(R_{\text{minimax}}, R_Q^{\mathfrak{d}}). \quad (24)$$

Similarly, this inequality can be converted to an explicit lower bound on minimax risk. We will show an application of this inequality in deriving Brygé-Gushchin inequality (Gushchin, 2003; Brygé, 2005) in Section 3.3.

3.1 Generalized Fano's Inequality

In the next result, we derive the generalized Fano's inequality (5) using Theorem 2. The inequality proved here is in fact slightly stronger than (5); see Remark 6 for the clarification.

Corollary 5 (Generalized Fano's inequality) For any given prior w and zero-one loss L , we have

$$R_{\text{Bayes}}(w, L; \Theta) \geq 1 + \frac{I(w, \mathcal{P}) + \log(1 + R_0)}{\log(\sup_{a \in \mathcal{A}} w(B(a)))}, \quad (25)$$

where $B(a)$ is defined in (16).

Proof [Proof of Corollary 5]

We simply apply (19) to $f(x) = x \log x$ and $r = R_0/(1 + R_0)$, it can then be checked that

$$\phi_f(r; R_0) = -\log(1 + R_0) - \frac{1}{1 + R_0} \log(1 - R_0), \quad \phi_f'(r-, R_0) = \log(1 - R_0),$$

Inequality (19) then gives

$$R \geq 1 + \frac{I(w, \mathcal{P}) + \log(1 + R_0)}{\log(1 - R_0)}$$

which proves (25). ■

Remark 6 This inequality is slightly stronger than (5) because $R_0 \leq 1$ (thus $\log(1 + R_0) \leq \log 2$). For example, when $\Theta = \mathcal{A} = \{0, 1\}$, $L(\theta, a) := \mathbb{I}\{\theta \neq a\}$ and $w\{0\} = w\{1\} = 1/2$, the inequality (5) leads to a trivial bound since the right hand side of (5) is negative. However, since $R_0 = 1/2$, the inequality (25) still provides a useful lower bound when $I(w, \mathcal{P})$ is strictly smaller than $\log 2 - \log(3/2)$.

As mentioned in the introduction, the classical Fano inequality (3) and the recent continuum Fano inequality (4) are both special cases (restricted to uniform priors) of Corollary 5. The proof of (4) given in Duchi and Wainwright (2013) is rather complicated with a stronger assumption and a discretization-approximation argument. Our proof based on Theorem 2 is much simpler. Lemma 3 also has its independent interest. Using Lemma 3, we are able to recover another recently proposed variant of Fano's inequality in Braun and Pokutta (2014, Proposition 2.2). Details of this argument are provided in Appendix A.2.

3.2 Specialization of Theorem 2 to Different f -Divergences and Their Applications

In addition to the generalized Fano's inequality, Theorem 2 allows us to derive a class of lower bounds on Bayes risk for zero-one losses by plugging other f -divergences. In the next corollary, we consider some widely used f -divergences and provide the corresponding Bayes risk lower bounds by inverting (17) in Theorem 2.

Corollary 7 Let L be zero-one valued, w be any prior on Θ and $R = R_{\text{Bayes}}(w, L, \Theta)$. We then have the following inequalities

(i) *Chi-squared divergence:*

$$R \geq R_0 - \sqrt{R_0(1 - R_0)I_{\chi^2}(w, \mathcal{P})}. \quad (26)$$

(ii) *Total variation distance:*

$$R \geq R_0 - I_{\text{TV}}(w, \mathcal{P}). \quad (27)$$

(iii) *Hellinger distance:*

$$R \geq R_0 - (2R_0 - 1) \frac{h^2}{2} - \sqrt{R_0(1 - R_0)h^2(2 - h^2)}. \quad (28)$$

provided $h^2 \leq 2R_0$. Here $h^2 = \int_{\Theta} \int_{\Theta} H^2(P_{\theta} \| P_{\theta'}) w(d\theta) w(d\theta')$.

See Appendix A.3 for the proof of the corollary. The special case of Corollary 7 for $\Theta = \mathcal{A} = \{1, \dots, N\}$, $L(\theta, a) = \mathbb{I}\{\theta \neq a\}$ and w being the uniform prior has been discovered previously in Guntuboyina (2011b). It is clear from Corollary 7 that the choice of f -divergence will affect the tightness of the lower bound for R . In Appendix A.5, we provide a qualitative comparison of the lower bounds (25), (26) and (28). In particular, we show that in the discrete setting with $\Theta = \mathcal{A} = \{1, \dots, N\}$, the lower bounds induced by the KL divergence and the chi-squared divergence are much stronger than the bounds given by the Hellinger distance. Therefore, in most applications in this paper, we shall only use the bounds involving the KL divergence and the chi-squared divergence.

Corollary 7 can be used to recover classical inequalities of Le Cam (for two point hypotheses) and Assouad (Theorem 2.12 in Tsybakov (2010) with both total variation distance and Hellinger distance) and Theorem 2.15 in Tsybakov (2010) that involves fuzzy hypotheses. The details are presented in Appendix A.4.

3.3 Birgé-Gushchin's Inequality

In this section, we expand (24) to obtain a minimax risk lower bound due to Gushchin (2003) and Birgé (2005), which presents an improvement of the classical Fano's inequality when specializing to KL divergence.

Proposition 8 (Gushchin, 2003; Birgé, 2005) Consider the finite parameter and action space $\Theta = \mathcal{A} = \{\theta_0, \theta_1, \dots, \theta_N\}$ and the zero-one valued indicator loss $L(\theta, a) = \mathbb{I}\{\theta \neq a\}$, for any f -divergence,

$$\phi_f(R_{\text{minimax}}, 1 - R_{\text{minimax}}/N) \leq \min_{0 \leq j \leq N} \frac{1}{N} \sum_{i \neq j} D_f(P_{\theta_i} \| P_{\theta_j}). \quad (29)$$

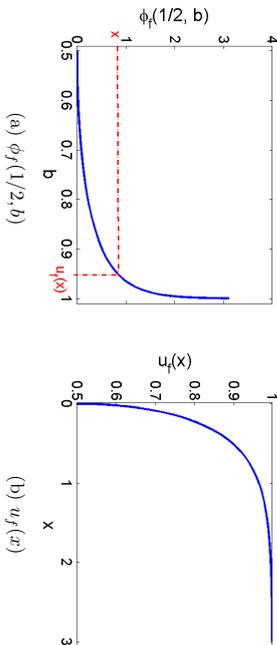
Proof [Proof of Proposition 8]

To prove Proposition 8, it is enough to prove that $\frac{1}{N} \sum_{i \neq j} D_f(P_{\theta_i} \| P_{\theta_j}) \geq \phi_f(R_{\text{minimax}}, 1 - R_{\text{minimax}}/N)$ for every $j \in \{0, \dots, N\}$. Without loss of generality, we assume that $j = 0$. We apply (20) with the uniform distribution on $\Theta \setminus \{\theta_0\} = \{\theta_1, \dots, \theta_N\}$ as w , $Q = P_{\theta_0}$ and the minimax rule for the problem as \mathfrak{d} . Because \mathfrak{d} is the minimax rule, $R^{\mathfrak{d}} \leq R_{\text{minimax}}$. Also

$$R_Q^{\mathfrak{d}} = \frac{1}{N} \sum_{i=1}^N \mathbb{E}_{\theta_0} L(\theta_i, \mathfrak{d}(X)) = \frac{1}{N} \mathbb{E}_{\theta_0} \sum_{i=1}^N \mathbb{I}\{\theta_i \neq \mathfrak{d}(X)\}.$$

It is easy to verify that $\sum_{i=1}^N \mathbb{I}\{\theta_i \neq \mathfrak{d}(X)\} = N - \mathbb{I}\{\theta_0 \neq \mathfrak{d}(X)\}$. We thus have $R_Q^{\mathfrak{d}} = 1 - \mathbb{E}_{\theta_0} L(\theta_0, \mathfrak{d}(X))/N$. Because \mathfrak{d} is minimax, $\mathbb{E}_{\theta_0} L(\theta_0, \mathfrak{d}(X)) \leq R_{\text{minimax}}$ and thus

$$R_Q^{\mathfrak{d}} \geq 1 - R_{\text{minimax}}/N. \quad (30)$$

Figure 2: Illustration of $\phi_f(1/2, b)$ and $u_f(x)$ for $f(x) = x \log x$.

On the other hand, we have $R_{\min\max} \leq N/(N+1)$. To see this, note that the minimax risk is upper bounded by the maximum risk of a random decision rule, which chooses among the $N+1$ hypotheses uniformly at random. For this random decision rule, its risk is $\frac{N}{N+1}$ no matter what the true hypothesis is. Thus, $\frac{N}{N+1}$ is an upper bound on the minimax risk. We thus have, from (30), that $R_0^2 \geq 1 - R_{\min\max}/N \geq R_{\min\max}$. We can thus apply (24) to obtain

$$\frac{1}{N} \sum_{i=1}^N D_f(P_{\theta_i} \| P_{\theta_0}) \geq \phi_f(R_{\min\max}, 1 - R_{\min\max}/N),$$

which completes the proof Proposition 8. \blacksquare

4. Bayes Risk Lower Bounds for Nonnegative Loss Functions

In the previous section, we discussed Bayes risk lower bounds for zero-one valued loss functions. We deal with general nonnegative loss functions in this section. The main result of this section, Theorem 9, provides lower bounds for $R_{\text{Bayes}}(w, L; \Theta)$ for any given loss L and prior w . To state this result, we need the following notion. Fix $f \in \mathcal{C}$ and recall the definition of ϕ_f in (13). We define $u_f : [0, \infty) \rightarrow [1/2, 1]$ by

$$u_f(x) := \inf \{1/2 \leq b \leq 1 : \phi_f(1/2, b) > x\} \quad (31)$$

and if $\phi_f(1/2, b) \leq x$ for every $b \in [1/2, 1]$, then we take $u_f(x)$ to be 1. By Lemma 1, it is easy to see that $u_f(x)$ is a non-decreasing function of x . For example, for KL-divergence with $f(x) = x \log x$, we have $\phi_f(1/2, b) = \frac{1}{2} \log \frac{1-b}{1-b}$ and $u_f(x) = \frac{1}{2} + \frac{1}{2} \sqrt{1 - e^{-2x}}$ (see Figure 2). We are now ready to state the main theorem of this paper.

Theorem 9 For every $\Theta, \mathcal{A}, L, w$ and $f \in \mathcal{C}$, we have

$$R_{\text{Bayes}}(w, L; \Theta) \geq \frac{1}{2} \sup \left\{ t > 0 : \sup_{a \in \mathcal{A}} w(B_t(a, L)) < 1 - u_f(I_f(w, \mathcal{P})) \right\}, \quad (32)$$

where

$$B_t(a, L) := \{\theta \in \Theta : L(\theta, a) < t\} \quad \text{for } a \in \mathcal{A} \text{ and } t > 0. \quad (33)$$

Proof [Proof of Theorem 9]

Fix $\Theta, \mathcal{A}, L, w$ and f . Let $I := I_f(w, \mathcal{P})$ be a shorthand notation. Suppose $t > 0$ is such that

$$\sup_{a \in \mathcal{A}} w(B_t(a, L)) < 1 - u_f(I). \quad (34)$$

We prove below that $R_{\text{Bayes}}(w, L; \Theta) \geq t/2$ and this would complete the proof. Let L_t denote the zero-one valued loss function $L_t(\theta, a) := \mathbb{1}\{L(\theta, a) \geq t\}$. It is obvious that $L \geq tL_t$ and hence the proof will be complete if we establish that $R_{\text{Bayes}}(w, L_t; \Theta) \geq 1/2$. Let $R := R_{\text{Bayes}}(w, L_t; \Theta)$ for a shorthand notation.

Because L_t is a zero-one valued loss function, Theorem 2 gives

$$I \geq \phi_f(R, R_0) \quad \text{where } R_0 = 1 - \sup_{a \in \mathcal{A}} w(B_t(a, L)). \quad (35)$$

By (34), it then follows that $R_0 > u_f(I)$. By definition of $u_f(\cdot)$, it is clear that there exists $b^* \in [1/2, R_0)$ such that $\phi(1/2, b^*) > I$ (this in particular implies that $R_0 \geq 1/2$). Lemma 1 implies that $b \mapsto \phi_f(1/2, b)$ is non-decreasing for $b \in [1/2, 1]$, which yields $\phi_f(1/2, b^*) \leq \phi_f(1/2, R_0)$. The above two inequalities imply $I < \phi_f(1/2, R_0)$. Combining this inequality with (35), we have

$$\phi_f(1/2, R_0) > I \geq \phi_f(I, R_0).$$

Lemma 1 shows that $a \mapsto \phi_f(a, R_0)$ is non-increasing for $a \in [0, R_0]$. Thus, we have $R \geq 1/2$. \blacksquare

We further note that because $u_f(x)$ is non-decreasing in x , one can replace $I_f(w, \mathcal{P})$ in (32) by any upper bound I_f^{ub} i.e., for any $I_f^{\text{ub}} \geq I_f(w, \mathcal{P})$, we have

$$R_{\text{Bayes}}(w, L; \Theta) \geq \frac{1}{2} \sup \left\{ t > 0 : \sup_{a \in \mathcal{A}} w(B_t(a, L)) < 1 - u_f(I_f^{\text{ub}}) \right\}. \quad (36)$$

This is useful since $I_f(w, \mathcal{P})$ is often difficult to calculate exactly. When $f(x) = x \log x$, Hausler and Oppet (1997) provided a useful upper bound on the mutual information $I(w, \mathcal{P})$. We describe this result in Section 5 where we also extend it to power divergences f_a for $a \notin [0, 1]$ (which covers the case of chi-squared divergence).

Remark 10 From the proof of Theorem 9, it can be observed that the constant $1/2$ in the right hand side of (32) and in the definition of $u_f(\cdot)$ can be replaced by any $c \in (0, 1]$. This gives the sharper lower bound:

$$R_{\text{Bayes}}(w, L; \Theta) \geq \sup_{c \in (0, 1]} \left(c \sup \left\{ t > 0 : \sup_{a \in \mathcal{A}} w(B_t(a, L)) < 1 - u_{f,c}(I_f(w, \mathcal{P})) \right\} \right),$$

where $u_{f,c}(x) = \inf\{c \leq b \leq 1 : \phi_f(c, b) \geq x\}$. Since obtaining exact constants is not our main concern, the inequality (32) is usually sufficient to provide Bayes risk lower bounds with correct dependence on the model and prior.

Remark 11 We note that the lower bound presented in Theorem 9 might not be tight for some spectral priors, e.g., when the prior w has extremely large density in some small region of the parameter space. We call such regions with unbounded density as spikes in the prior distribution. As a concrete example, let $\Theta = \mathcal{A}$ be a subset of a finite dimensional Euclidean space containing the origin with L being the Euclidean distance and let w denote the mixture of the uniform priors over the balls $B_1(0, L)$ and $B_\epsilon(0, L)$ for some very small $0 < \epsilon \ll 1$. In this case, the mixture component $B_\epsilon(0, L)$ is a spike. If ϵ is very small, then the term $\sup_{a \in \mathcal{A}} w(B_\epsilon(a, L))$ might be too big for Theorem 9 to establish a tight lower bound.

Even in such extreme cases, the tight lower bound can be salvaged by partitioning the parameter space Θ into finite or countably many disjoint subsets $\Theta_i, i \geq 0$ and to apply Theorem 9 to w restricted to each Θ_i . To illustrate this technique, suppose that w has a Lebesgue density φ that is bounded from above. Let φ_{\max} denote the supremum of φ . We partition the parameter space Θ into disjoint subsets $\Theta_0, \Theta_1, \dots$ with

$$\Theta_i := \{\theta \in \Theta : 2^{-(i+1)}\varphi_{\max} < \varphi(\theta) \leq 2^{-i}\varphi_{\max}\}. \quad (37)$$

Then, we apply Theorem 9 to w restricted to each Θ_i . More specifically, let w_i denote the probability measure w restricted to Θ_i i.e., $w_i(S) := w(S \cap \Theta_i)/w(\Theta_i)$ for any measurable set $S \subseteq \Theta_i$. we have

$$R_{\text{Bayes}}(w, L; \Theta) \geq \sum_i w(\Theta_i) R_{\text{Bayes}}(w_i, L; \Theta_i), \quad (38)$$

where $R_{\text{Bayes}}(w_i, L; \Theta_i) = \inf_{\delta} \int_{\Theta_i} \mathbb{E}_{\theta} L(\theta, \delta(X)) w_i(d\theta)$. To see this, for any decision rule δ , we have $\mathbb{R}^{\delta}(w, L; \Theta) = \sum_{i=1}^{\infty} w(\Theta_i) \mathbb{R}^{\delta}(w_i, L; \Theta_i)$; then take infimum over all possible δ on both sides,

$$\begin{aligned} R_{\text{Bayes}}(w, L; \Theta) &= \inf_{\delta} \mathbb{R}^{\delta}(w, L; \Theta) \\ &\geq \sum_{i=1}^{\infty} w(\Theta_i) \inf_{\delta} \mathbb{R}^{\delta}(w_i, L; \Theta_i) = \sum_{i=1}^{\infty} w(\Theta_i) R_{\text{Bayes}}(w_i, L; \Theta_i) \end{aligned}$$

One can lower bound each Bayes risk $R_{\text{Bayes}}(w_i, L; \Theta_i)$ for all i using Theorem 9. Since the density of w_i differs by a factor at most 2, the spiking prior problem will no longer exist while applying Theorem 9 for w_i . We also note that another useful application of such a partitioning technique is presented in Corollary 17.

Now take the concrete example of the mixture of the uniform priors over $B_1 := B_1(0, L)$ and $B_\epsilon := B_\epsilon(0, L)$. It is clear from (37) that $\Theta_0 = B_\epsilon$ and $\Theta_k = B_1 \setminus B_\epsilon$ for some $k > 0$ and the rest of Θ_i 's are empty sets. Applying (38), we have

$$\begin{aligned} R_{\text{Bayes}}(w, L; \Theta) &\geq w(B_\epsilon) R_{\text{Bayes}}(w_1, L; B_\epsilon) + w(B_1 \setminus B_\epsilon) R_{\text{Bayes}}(w_2, L; B_1 \setminus B_\epsilon) \\ &\geq w(B_1 \setminus B_\epsilon) R_{\text{Bayes}}(w_2, L; B_1 \setminus B_\epsilon) \end{aligned}$$

Note that $w(B_1 \setminus B_\epsilon)$ is lower bounded by a universal constant. Then we can lower bound $R_{\text{Bayes}}(w_2, L; B_1 \setminus B_\epsilon)$ using Theorem 9 and obtain a tight lower bound up to a constant factor that is independent of ϵ (see an example of deriving Bayes risk lower bound for estimating the mean of a Gaussian model with uniform prior on a ball in Section 5).

For specific $f \in \mathcal{C}$, the right hand side of (36) can be explicitly evaluated as shown in the next corollary.

Corollary 12 Fix $\Theta, \mathcal{A}, L, w$ and \mathcal{P} . The Bayes risk $R_{\text{Bayes}}(w, L; \Theta)$ satisfies each of the following inequalities (the quantity I_f^{up} represents an upper bound on the corresponding f -informativity):

(i) KL divergence:

$$R_{\text{Bayes}}(w, L; \Theta) \geq \frac{1}{2} \sup_{a \in \mathcal{A}} \left\{ t > 0 : \sup_{a \in \mathcal{A}} w(B_t(a, L)) < \frac{1}{4} e^{-2I_f^{\text{up}}} \right\}. \quad (39)$$

(ii) Chi-squared divergence:

$$R_{\text{Bayes}}(w, L; \Theta) \geq \frac{1}{2} \sup_{a \in \mathcal{A}} \left\{ t > 0 : \sup_{a \in \mathcal{A}} w(B_t(a, L)) < \frac{1}{4(1+I_f^{\text{up}})} \right\}. \quad (40)$$

(iii) Total variation distance:

$$R_{\text{Bayes}}(w, L; \Theta) \geq \frac{1}{2} \sup_{a \in \mathcal{A}} \left\{ t > 0 : \sup_{a \in \mathcal{A}} w(B_t(a, L)) < \frac{1}{2} - I_f^{\text{up}} \right\}. \quad (41)$$

(iv) Hellinger distance: If $I_f^{\text{up}} < 1 - 1/\sqrt{2}$, then we have

$$R_{\text{Bayes}}(w, L; \Theta) \geq \frac{1}{2} \sup_{a \in \mathcal{A}} \left\{ t > 0 : \sup_{a \in \mathcal{A}} w(B_t(a, L)) < \frac{1}{2} - (1 - I_f^{\text{up}}) \sqrt{I_f^{\text{up}}(2 - I_f^{\text{up}})} \right\}. \quad (42)$$

Proof [Proof of Corollary 12]

Inequality (39) involving KL divergence: Suppose $f(x) = x \log x$ so that $D_f(P||Q) = D(P||Q)$ equals the KL divergence. Then the function $u_f(x)$ in (31) has the expression for all $x > 0$,

$$u_f(x) = \inf \left\{ 1/2 \leq b \leq 1 : b(1-b) < e^{-2x}/4 \right\} = \frac{1}{2} + \frac{1}{2} \sqrt{1 - e^{-2x}}.$$

The elementary inequality $\sqrt{1-a} \leq 1 - a/2$ gives for all $x > 0$,

$$u_f(x) \leq 1 - \frac{1}{4} e^{-2x}.$$

Inequality (32) reduces to the desired inequality (39):

$$R_{\text{Bayes}}(w, L; \Theta) \geq \frac{1}{2} \sup_{a \in \mathcal{A}} \left\{ t > 0 : \sup_{a \in \mathcal{A}} w(B_t(a, L)) < \frac{1}{4} e^{-2I_f^{\text{up}}} \right\}.$$

The proof of the Bayes risk lower bounds for the other three f -divergences are similar and thus we only present the form of $u_f(x)$. Inequality (40) involves chi-squared divergence with $f(x) = x^2 - 1$. Therefore, we have for all $x > 0$,

$$u_f(x) = \inf \left\{ 1/2 \leq b \leq 1 : \frac{(1-2b)^2}{4b(1-b)} > x \right\} = \frac{1}{2} + \frac{1}{2} \sqrt{\frac{x}{1+x}} \leq 1 - \frac{1}{4(1+x)}.$$

Inequality (41) involves total variation distance with $f(x) = |x - 1|/2$. Then

$$u_f(x) = \inf \{1/2 \leq b \leq 1 : |1 - 2b| > 2x\} = \frac{1}{2} + x.$$

Inequality (42) involves Hellinger divergence with $f(x) = 1 - \sqrt{x}$ and thus

$$\begin{aligned} u_f(x) &= \inf \left\{ 1/2 \leq b \leq 1 : 1 - \sqrt{b/2} - \sqrt{(1-b)/2} > x \right\} \\ &= \begin{cases} 1 & \text{if } x \geq 1 - 1/\sqrt{2} \\ \frac{1}{2} + (1-x)\sqrt{x(2-x)} & \text{if } x < 1 - 1/\sqrt{2}. \end{cases} \end{aligned}$$

■

Remark 13 A special case of Corollary 12(i) appeared as Zhang (2006, Theorem 6.1). To see that Zhang (2006, Theorem 6.1) is indeed a special case of (39), note first that (39) is equivalent to

$$R_{\text{Bayes}}(w, L; \Theta) \geq \frac{1}{2} \sup \left\{ t > 0 : \inf_{a \in \mathcal{A}} \frac{1}{w(B_t(a, L))} > 2I^w + \log 4 \right\}. \quad (43)$$

Here I^w is any upper bound on the mutual information. One such upper bound on the mutual information is

$$I^w = \int_{\Theta} \int_{\Theta} D(P_{\theta} \| P_{\xi}) w(d\xi) w(d\theta) \quad (44)$$

That I^w is an upper bound on the mutual information can be seen for example by using concavity of the logarithm (46) when the family $\{Q_{\xi}, \xi \in \Xi\}$ is chosen to be the same as $\{P_{\theta}, \theta \in \Theta\}$. Using (44) in (43), we obtain

$$R_{\text{Bayes}}(w, L; \Theta) \geq \frac{1}{2} \sup \left\{ t > 0 : \inf_{a \in \mathcal{A}} \frac{1}{w(B_t(a, L))} > 2 \int_{\Theta} \int_{\Theta} D(P_{\theta} \| P_{\xi}) w(d\xi) w(d\theta) + \log 4 \right\}.$$

If we now specialize to the setting when the probability measures $\{P_{\theta}, \theta \in \Theta\}$ are all n -fold product measures i.e., when each P_{θ} is of the form $\mathfrak{P}_{\theta}^{\otimes n}$ for some class of probabilities $\{\mathfrak{P}_{\theta}, \theta \in \Theta\}$, then the inequality becomes

$$R_{\text{Bayes}}(w, L; \Theta) \geq \frac{1}{2} \sup \left\{ t > 0 : \inf_{a \in \mathcal{A}} \frac{1}{w(B_t(a, L))} > 2n \int_{\Theta} \int_{\Theta} D(\mathfrak{P}_{\theta} \| \mathfrak{P}_{\xi}) w(d\xi) w(d\theta) + \log 4 \right\}.$$

This inequality is precisely Zhang (2006, Theorem 6.1).

5. Upper Bounds on f -informativity and Examples

Application of Theorem 9 requires upper bounds on the f -informativity $I_f(w; \mathcal{P})$. This is the subject of this section. We focus on the power divergence f_{α} for $\alpha \geq 1$ which includes the KL divergence and chi-squared divergence as special cases. Recall that in the comment/paragraph below Corollary 7 (see also Section A.5 in the appendix), we provided

motivation for restricting our attention to such divergences as opposed to e.g., Hellinger distance.

We assume that there is a measure μ on \mathcal{X} that dominates P_{θ} for every $\theta \in \Theta$. None of our results depend on the choice of the dominating measure μ .

When the f -informativity is the mutual information, Haussler and Opper (1997) have proved useful upper bounds which we briefly review here. Let P and $\{Q_{\xi}, \xi \in \Xi\}$ be probability measures on \mathcal{X} having densities p and $\{q_{\xi}, \xi \in \Xi\}$ respectively with respect to μ . Let ν be an arbitrary probability measure on Ξ and \bar{Q} be the probability measure on \mathcal{X} having density $\bar{q} = \int_{\Xi} q_{\xi} \nu(d\xi)$ with respect to μ . Haussler and Opper (1997) proved the following inequality

$$D(P \| \bar{Q}) \leq -\log \left(\int_{\Xi} \exp(-D(P \| Q_{\xi})) \nu(d\xi) \right). \quad (45)$$

Now given a class of probability measures $\{P_{\theta}, \theta \in \Theta\}$, applying the above inequality for each P_{θ} and integrating the resulting inequalities with respect to a probability measure w on Θ , Haussler and Opper (1997, Theorem 2) obtained the following mutual information upper bound:

$$I(w, \mathcal{P}) \leq -\int_{\Theta} \log \left(\int_{\Xi} \exp(-D(P_{\theta} \| Q_{\xi})) \nu(d\xi) \right) w(d\theta). \quad (46)$$

In the special case when $\Xi = \{1, \dots, M\}$ and ν is the uniform probability measure on Ξ , we have $\bar{Q} = (Q_1 + \dots + Q_M)/M$ and inequality (45) then becomes $D(P \| \bar{Q}) \leq -\log \left(\frac{1}{M} \sum_{j=1}^M \exp(-D(P \| Q_j)) \right)$. Because $\sum_{j=1}^M \exp(-D(P \| Q_j)) \geq \exp(-\min_j D(P \| Q_j))$, we obtain

$$D(P \| \bar{Q}) \leq \log M + \min_{1 \leq j \leq M} D(P \| Q_j).$$

Inequality (46) can be further simplified to

$$I(w, \mathcal{P}) \leq \log M + \int_{\Theta} \min_{1 \leq j \leq M} D(P_{\theta} \| Q_j) w(d\theta). \quad (47)$$

This inequality can be used to give an upper bound for f -informativity in terms of the KL covering numbers. Recall the definition of $M_{KL}(\epsilon, \Theta)$ from (12). Applying (47) to any fixed $\epsilon > 0$ and choosing $\{Q_1, \dots, Q_M\}$ to be an ϵ^2 -covering, we have

$$I(w, \mathcal{P}) \leq \inf_{\epsilon > 0} (\log M_{KL}(\epsilon, \Theta) + \epsilon^2). \quad (48)$$

When w is the uniform prior on a finite subset of Θ , the above inequality has been proved by Yang and Barron (1999, Page 1571). If $M_{KL}(\epsilon, \Theta)$ is infinity for all ϵ , then (48) gives ∞ as the upper bound on $I(w, \mathcal{P})$ and thus (39) will lead to a trivial lower bound 0 for R_{Bayes} . In such a case, one may find a subset $\tilde{\Theta} \subset \Theta$ for which $M_{KL}(\epsilon, \tilde{\Theta})$ is bounded and contains most prior mass. If \tilde{w} denotes the prior w restricted in $\tilde{\Theta}$, then it is easy to see that $R_{\text{Bayes}}(w, L; \Theta) \geq w(\tilde{\Theta}) R_{\text{Bayes}}(\tilde{w}, L; \tilde{\Theta})$. Then we can use (39) and (48) to lower bound $R_{\text{Bayes}}(w, L; \Theta)$.

In the next theorem, we extend inequalities (45) and (46) to power divergences corresponding to f_{α} for $\alpha \notin [0, 1]$. We also note that in Appendix B.2, we demonstrate the tightness of the bound (49) in Theorem 14 by a simple example.

Theorem 14 Fix $\alpha \notin [0, 1]$ and let $f_\alpha \in \mathcal{C}$ be as defined in Section 2. Under the setting of inequalities (45) and (46), we have

$$D_{f_\alpha}(P\|\bar{Q}) \leq \left[\int_{\Xi} (D_{f_\alpha}(P\|Q_\xi) + 1)^{1/(1-\alpha)} \nu(d\xi) \right]^{1-\alpha} - 1. \quad (49)$$

and

$$I_{f_\alpha}(w, \mathcal{P}) \leq \int_{\Theta} \left[\int_{\Xi} (D_{f_\alpha}(P_\theta\|Q_\xi) + 1)^{1/(1-\alpha)} \nu(d\xi) \right]^{1-\alpha} w(d\theta) - 1. \quad (50)$$

To prove Theorem 14, the following lemma is critical (the proof of this lemma in given in Appendix B.1).

Lemma 15 Fix $r < 1$. Let μ be a probability measure on the space T and let $S := \{u : T \rightarrow \mathbb{R}_+ : u \in L_r^+(T)\}$. Then the map $f : S \rightarrow \mathbb{R}$ defined by $f(u) := \left(\int_T u(t)^r \mu(dt) \right)^{1/r}$ is concave in u .

Note that the discrete version of Lemma 15 states that $f(u) = \left(\sum_{i=1}^M u_i^r / M \right)^{1/r}$ is a concave function of $u \in \mathbb{R}_+^M$ when $r < 1$.

In fact, since we will apply this lemma to prove Theorem 14 with $r = \frac{1}{1-\alpha}$, the condition $r < 1$ in Lemma 15 translates into $\alpha \notin [0, 1]$ in Theorem 14. We are now ready to prove Theorem 14.

Proof [Proof of Theorem 14]

By the identity that $D_{f_\alpha}(P\|Q) = D_{f_{1-\alpha}}(Q\|P)$, we have

$$\begin{aligned} D_{f_\alpha}(P\|\bar{Q}) &= D_{f_{1-\alpha}}(\bar{Q}\|P) = \int_{\mathcal{X}} p \left(\int_{\Xi} \frac{q\xi}{p} \nu(d\xi) d\mu \right)^{1-\alpha} - 1 \\ &= \int_{\mathcal{X}} p \left(\int_{\Xi} \left[\frac{q\xi}{p} \right]^{1-\alpha} \nu(d\xi) d\mu \right)^{1-\alpha} - 1 \end{aligned}$$

Let $u(\xi, x) = \left(\frac{q\xi}{p} \right)^{1-\alpha}$. Since $\frac{1}{1-\alpha} < 1$ when $\alpha \notin [0, 1]$, Lemma 15 implies that $u(\xi, x) \mapsto \left(\int_{\Xi} u(\xi, x)^{1/(1-\alpha)} \nu(d\xi) \right)^{1-\alpha}$ is concave in u . Applying Jensen's inequality,

$$\begin{aligned} D_{f_\alpha}(P\|\bar{Q}) &\leq \left(\int_{\Xi} \left[\int_{\mathcal{X}} p \left(\frac{q\xi}{p} \right)^{1-\alpha} d\mu \right]^{1/(1-\alpha)} \nu(d\xi) \right)^{1-\alpha} - 1 \\ &= \left(\int_{\Xi} [D_{f_{1-\alpha}}(Q_\xi\|P)]^{1/(1-\alpha)} \nu(d\xi) \right)^{1-\alpha} - 1. \end{aligned}$$

This completes the proof of (49) because $D_{f_{1-\alpha}}(Q_\xi\|P) = D_{f_\alpha}(P\|Q_\xi)$. The proof of (50) follows by applying (49) for $P = P_\theta$ and then integrating the resulting bound with respect to $w(d\theta)$. \blacksquare

For $\alpha > 1$, one can deduce an upper bound analogous to (48) for the f_α -informativity which is described in the next corollary. Recall the notion of the covering numbers $M_\alpha(\epsilon, \Theta)$ from Section 2.

Corollary 16 For every $\alpha > 1$, we have

$$I_{f_\alpha}(w, \mathcal{P}) \leq \inf_{\epsilon > 0} (1 + \epsilon^2) M_\alpha(\epsilon, \Theta)^{\alpha-1} - 1. \quad (51)$$

In particular, when D_{f_α} is the chi-square divergence, Corollary 16 implies

$$I_{\chi^2}(w, \mathcal{P}) \leq \inf_{\epsilon > 0} (1 + \epsilon^2) M_{\chi^2}(\epsilon, \Theta) - 1. \quad (52)$$

Note that Corollary 16 gives trivial bound when $M_\alpha(\epsilon, \Theta)$ equals ∞ for all $\epsilon > 0$. This can be handled in a way similar to that outlined in the discussion after (48).

Proof [Proof of Corollary 16]

Let Q_1, \dots, Q_M be probability measures on \mathcal{X} and fix $\theta \in \Theta$. Inequality (49) applied to $P = P_\theta$, $\Xi := \{1, \dots, M\}$ and the uniform probability measure on Ξ as ν gives

$$D_{f_\alpha}(P_\theta\|\bar{Q}) \leq M^{\alpha-1} \left[\sum_{j=1}^M (1 + D_{f_\alpha}(P_\theta\|Q_j))^{1/(1-\alpha)} \right]^{1-\alpha} - 1$$

We now use (note that $\alpha > 1$)

$$\begin{aligned} \sum_{j=1}^M (1 + D_{f_\alpha}(P_\theta\|Q_j))^{1/(1-\alpha)} &\geq \max_{1 \leq j \leq M} (1 + D_{f_\alpha}(P_\theta\|Q_j))^{1/(1-\alpha)} \\ &= (1 + \min_{1 \leq j \leq M} D_{f_\alpha}(P_\theta\|Q_j))^{1/(1-\alpha)}. \end{aligned}$$

This gives

$$D_{f_\alpha}(P_\theta\|\bar{Q}) \leq M^{\alpha-1} \left(1 + \min_{1 \leq j \leq M} D_{f_\alpha}(P_\theta\|Q_j) \right) - 1.$$

We now fix $\epsilon > 0$ and apply the above with $\{Q_1, \dots, Q_M\}$ taken to be an ϵ^2 -cover of Θ under the f_α -divergence. We then obtain

$$D_{f_\alpha}(P_\theta\|\bar{Q}) \leq \inf_{\epsilon > 0} (1 + \epsilon^2) M_\alpha(\epsilon, \Theta)^{\alpha-1} - 1.$$

The proof is complete by integrating the above inequality with respect to $w(d\theta)$. \blacksquare

We now turn to applications of the Bayes risk lower bounds in Corollary 12 and the informativity upper bounds in this section. We present a toy example here and postpone more complicated examples (e.g., generalized linear model, spiked covariance model, Gaussian model with general prior and loss) to Appendix C.

Example 1 (Gaussian model with uniform priors on large balls) Fix $d \geq 1$. Suppose $\Theta = \mathcal{A} \subset \mathbb{R}^d$ and let $L(\theta, a) := \|\theta - a\|_2^2$. For each $\theta \in \mathbb{R}^d$, let P_θ denote the Gaussian distribution with mean θ and covariance matrix $\sigma^2 I_{d \times d}$ ($\sigma^2 > 0$ is a constant). Let w be the uniform distribution on the closed ball of radius Γ centered at the origin. Let $\Gamma \geq \sigma\sqrt{d}$.

We will show below how to obtain the tight Bayes risk lower bound using Corollary 12 along with the f -informativity upper bound in Corollary 16.

We can assume that Θ (and \mathcal{A}) is the closed ball of radius Γ centered at the origin as w puts zero probability outside this ball. We use the inequality (40) induced by the chi-squared divergence. To establish the lower bound, we need to upper bound $\sup_{a \in \mathcal{A}} w(B_1(a, L))$ and the chi-squared informativity. The former can be easily controlled because $\sup_{a \in \mathcal{A}} w(B_1(a, L)) \leq (\sqrt{L}/\Gamma)^d$. For the latter, we use (52), which requires an upper bound on $M_{\chi^2}(\epsilon, \Theta)$. Note that $\chi^2(P_\theta \| P_\theta) = \exp(\|\theta - \theta'\|_2^2 / \sigma^2) - 1$ for $\theta, \theta' \in \Theta$. As a consequence, $\chi^2(P_\theta \| P_\theta) \leq e^2$ if and only if $\|\theta - \theta'\|_2 \leq e' := \sigma \sqrt{\log(1 + e^2)}$. Therefore, by a standard volumetric argument, we have

$$M_{\chi^2}(\epsilon, \Theta) \leq \left(\frac{\Gamma + e'/2}{e'/2} \right)^d \leq \left(\frac{3\Gamma}{e'} \right)^d = \left(\frac{3\Gamma}{\sigma \sqrt{\log(1 + e^2)}} \right)^d$$

provided $e' \leq \Gamma$. In particular, if we take $\epsilon := \sqrt{e^d} - 1$, then $e' = \sigma \sqrt{d} \leq \Gamma$, we will obtain $M_{\chi^2}(\epsilon, \Theta) \leq (3\Gamma/(\sigma \sqrt{d}))^d$. Inequality (52) then gives $I_{\chi^2}(w, \mathcal{P}) \leq \left(\frac{3e\Gamma}{\sigma \sqrt{d}} \right)^d - 1$. Let I_f^{up} be the right hand side. If we choose $t = cd\sigma^2$ for a sufficiently small constant $c > 0$, then we have $\sup_{a \in \mathcal{A}} w(B_1(a, L)) < \frac{1}{4}(1 + I_f^{\text{up}})^{-1}$. Inequality (40) then gives

$$R_{\text{Bayes}}(w, L; \Theta) \geq cd\sigma^2. \quad (53)$$

This lower bound is tight due to the trivial upper bound $R_{\text{Bayes}}(w, L; \Theta) \leq d \min(\sigma^2, \Gamma^2)$ since $R_{\text{Bayes}}(w, L; \Theta)$ is smaller than the risk of the constant estimator 0 as well as the trivial estimator of the observation itself.

This example allows us to compare the bound given by Theorem 9 for different $f \in \mathcal{C}$. We argue below that using KL divergence and applying (39) along with inequality (48) for controlling the mutual information will not yield a tight lower bound for this example. In other words, the same strategy that works for $f(x) = x^2 - 1$ does not work for $f(x) = x \log x$. To see this, notice that $D(P_\theta \| P_\theta) = \|\theta - \theta'\|_2^2 / \sigma^2$ for $\theta, \theta' \in \Theta$. As a result, $D(P_\theta \| P_\theta) \leq e^2$ if and only if $\|\theta - \theta'\|_2 \leq \sqrt{2e}\sigma$. The same volumetric argument again gives $M_{KL}(\epsilon, \Theta) \leq \left(\frac{3\Gamma}{\sqrt{2e}\sigma} \right)^d$ provided $\sqrt{2e}\sigma \leq \Gamma$. The bound (48) implies that the mutual information $I(w, \mathcal{P})$ is bounded by

$$I(w, \mathcal{P}) \leq \int_{0 < \delta \leq \Gamma / (\sqrt{2e}\sigma)} d \log \left(\frac{3\Gamma}{\sqrt{2e}\sigma} \right) + e^2 = d \log \left(\frac{3\Gamma}{\sigma \sqrt{d}} \right) + \frac{d}{2}.$$

Let I_f^{up} be the right hand side above. The maximum $t > 0$ for which $(\sqrt{t}/\Gamma)^d < \frac{1}{4} \exp(-2I_f^{\text{up}})$ is on the order of $d^2 \sigma^4 / \Gamma^2$. This means that inequality (39) implies a weaker lower bound $\Omega(d^2 \sigma^4 / \Gamma^2)$, which is suboptimal when $d\sigma^2$ is small or when Γ is large. This is in contrast with the optimal bound (53).

In the above example, a direct application of Theorem 9 with $f(x) = x \log x$ does not produce a tight lower bound. This is mainly because, when the prior is over a large parameter space (e.g., a ball of a constant radius), the upper bound of mutual information over the entire parameter space Θ in (48) could be too loose. This can be corrected by partitioning the

parameter space Θ into small hypercubes, and applying our bounds for the prior restricted to each hypercube separately so that the mutual information inside the partition can be appropriately upper bounded using (48). This is another illustration of the idea described in Remark 11. We first describe this method in a more general setting in the following corollary and then apply it to the setting of Example 1. We use the following notation. For measurable subsets S of a Euclidean space, $\text{Vol}(S)$ denotes the volume (Lebesgue measure) of S .

Corollary 17 *Let $\Theta = \mathcal{A} \subseteq \mathbb{R}^d$. Suppose that the prior w has a Lebesgue density f_w that is positive over Θ . For each $\theta \in \Theta$ and $\delta > 0$, let*

$$r_\delta(\theta) := \sup \left\{ \frac{f_w(\theta_1)}{f_w(\theta_2)} : \theta_i \in \Theta \text{ and } \|\theta_i - \theta\|_2 \leq \sqrt{d}\delta \text{ for } i = 1, 2 \right\}.$$

Suppose also the existence of $A > 0$ such that $D(P_{\theta_1} \| P_{\theta_2}) \leq A \|\theta_1 - \theta_2\|_2^2$ for all $\theta_1, \theta_2 \in \Theta$ and the existence of $V > 0$ (which may depend on d) and $p > 0$ such that $\sup_{a \in \mathcal{A}} \text{Vol}(B_1(a, L)) \leq V\mu^{d/p}$ for every $t > 0$. Then

$$R_{\text{Bayes}}(w, L; \Theta) \geq \frac{1}{2} \sup_{0 < \delta \leq A^{-1/2}} \left[e^{-2p} p^p (8V)^{-p/d} \int_{\Theta} \left(\frac{1}{r_\delta(\theta)} \right)^{p/d} w(d\theta) \right]. \quad (54)$$

The proof of Corollary is quite technically involved and thus is deferred to Appendix B.3.

We demonstrate below that this corollary yields the correct rate in Example 1. More examples (e.g., estimation problem in generalized linear model, spiked covariance model, and Gaussian model with a general loss) are given in Appendix C.

Example 2 (Gaussian model with uniform priors on large balls (continued)) *Consider the same setting as in Example 1. Because $D(P_\theta \| P_\theta) = \|\theta - \theta'\|_2^2 / (2\sigma^2)$, we can take $A = (2\sigma^2)^{-1}$ in Corollary 17. Moreover, because $L(\theta, a) = \|\theta - a\|_2^2$, it is easy to see that $\sup_{a \in \mathcal{A}} \text{Vol}(B_1(a, L)) \leq t^{d/2} \text{Vol}(B)$ which means that we can take $p = 2$ and $V = \text{Vol}(B)$ in Corollary 17 where B is the unit ball in \mathbb{R}^d . Finally, because w is the uniform prior, we have $r_\delta(\theta) = 1$ for all $\theta \in \Theta$. Corollary 17 therefore gives*

$$R_{\text{Bayes}}(w, L; \Theta) \geq \frac{1}{2} \sup_{0 < \delta \leq \sqrt{\delta}\sigma} \left(e^{-4} 8^{-2/d} \delta^2 \text{Vol}(B)^{-2/d} \right).$$

This matches the tight lower bound (53) by noting that $\text{Vol}(B)^{1/d} \asymp d^{-1/2}$.

6. Smoothed Analysis for Spherical Gaussian Mixture Models with Uniform Weights

Smoothed analysis is a useful technique for analyzing algorithms that fail in the worst case but succeed with high probability in the average case. For parameter estimation problems, smoothed analysis assumes that the parameter to be estimated is randomly perturbed

by a small noise, and the data is generated with respect to the perturbed parameter as well. Under this setting, if the set of “bad” parameters that fail the estimator has zero measure, then the estimator will succeed almost surely after the perturbation. Smoothed analysis has been successfully applied to analyze linear programming (Blum and Dumagan, 2002; Duneagan et al., 2011; Hsu and Kakade, 2013; Spielman and Teng, 2003), integer programming (Röglin and Vöcking, 2007), binary search trees (Manthey and Reischuk, 2007), and other combinatorial problems (Banderier et al., 2003). See the paper by Spielman and Teng (2003) for a survey of existing works.

In this section, we use smoothed analysis to study an important problem in statistical estimation: learning mixture of spherical Gaussians. The problem of computing the maximum log-likelihood estimator is NP-hard (Arora and Kannan, 2005). However, if the true parameters are perturbed by a random noise, then we demonstrate that a variant of the polynomial-time algorithm proposed by Hsu and Kakade (2013) succeeds in estimating the Gaussian means. We present an upper bound on the algorithm’s mean-squared error using smoothed analysis, which achieves a better rate than the original algorithm of Hsu and Kakade (2013). Furthermore, we apply the Bayes risk lower bound developed in this paper to show that, the mean squared-error achieved by this algorithm is unimprovable, even under smoothed analysis. To the best of our knowledge, the lower bound cannot be established by traditional information-theoretic techniques for lower bounding minimax risks.

6.1 Learning Mixture of Gaussians

We study estimating the parameter of a Gaussian mixture model (GMM). The parameter of a GMM is a d -by- k matrix $\theta := (\theta_1, \dots, \theta_k)$. Each $\theta_i \in \mathbb{R}^d$ represents the mean of the i -th mixture component. We assume that the number of components k is much less than the dimensionality d . Suppose that n i.i.d. instances $\{x_i\}_{i=1}^n$ are sampled from the GMM with each $x_i \in \mathbb{R}^d$. Equivalently, it is generated by the following procedure: First, an integer z_i is uniformly sampled from $\{1, \dots, k\}$. This integer is called the *membership* of the i -th instance¹. Then, the vector x_i is drawn from the spherical Gaussian distribution $N(\theta_{z_i}; I_{d \times d})$. The goal is to estimate the parameters θ .

Information theoretically, the GMM model is learnable if the Gaussian means are well separated. Let D represent the minimum distance between two distinct component means. Venupala and Wang (2004) show that, as long as $D > C$ for C being a sufficiently large constant, the estimation error on θ scales as $\mathcal{O}(n^{-1/2})$. However, the algorithm achieving this rate has $\mathcal{O}(k^k)$ time complexity. When the mutual distance D is large enough, there are poly(n, d, k)-time algorithms to estimate the model parameters. In particular, Dasgupta (1999) presents an algorithm for $D = \Omega(\sqrt{d})$. Arora and Kannan (2005) and Dasgupta and Schulman (2000) present algorithms for $D = \Omega(d^{1/4})$. Venupala and Wang (2004) reduce this distance lower bound to $\Omega(k^{1/4})$. However, designing poly(n, d, k)-time algorithm for $\Omega(1)$ -separated GMMs is a long-standing open problem.

Hsu and Kakade (2013) proposed a method that does not need the well-separation condition. The only assumption is that $\{\theta_1, \dots, \theta_k\}$ are linearly independent. Let $\sigma_{\min} > 0$ be the smallest singular value of the matrix θ . Their algorithm runs in poly(n, d, k)-time

1. For simplicity, we focus on the case when all mixture components have equal weights, but our argument can be easily generalized to the case of non-uniform weights.

and achieves the following bound for estimator $\hat{\theta}$:

$$\|\hat{\theta} - \theta\|_F^2 = \mathcal{O}\left(\frac{\text{poly}(d, k, 1/\sigma_{\min}) \log(1/\delta)}{n}\right) \quad \text{with probability at least } 1 - \delta. \quad (55)$$

Here, $\|\cdot\|_F$ denotes the matrix Frobenius norm. In general, we cannot guarantee that $\sigma_{\min} > 0$. However, if we add a small perturbation on the true component means, then the assumption is satisfied almost surely. More precisely, we assume that there is a matrix $\theta^* \in \mathbb{R}^{d \times k}$ so that each entry of matrix θ is sampled from $\theta_{ij} \sim N(\theta_{ij}^*; \rho^2)$. The following lemma lower bounds the smallest singular value.

Lemma 18 (Ge et al. (2015), Lemma G.16) *Let $\theta^* \in \mathbb{R}^{d \times k}$ and suppose that $d \geq 3k$. If all entries of θ^* are independently perturbed by $N(0, \rho^2)$ to yield matrix θ . For any $\epsilon > 0$, with probability at least $1 - c_1(c_2\epsilon)^d$, the smallest singular value of matrix θ is lower bounded by:*

$$\sigma_{\min} > \epsilon\rho\sqrt{d}.$$

Here, c_1, c_2 are universal constants.

We choose $\epsilon, \rho \sim n^{-c}$ for a sufficiently small $c > 0$, then the perturbation diminishes to zero, and if $\sigma_{\min} > \epsilon\rho\sqrt{d}$ holds, then the right-hand side of equation (55) converges to zero at a polynomial rate as $n \rightarrow \infty$. Lemma 18 implies that the probability of this event is at least $1 - \mathcal{O}(n^{-cd})$. Thus, with high probability, the estimator $\hat{\theta}$ is consistent under the smoothed analysis.

The convergence rate of the estimator $\hat{\theta}$ can be improved if we add a mild assumption that $D = \tilde{O}(\sqrt{\log(nk)})$. Although the main focus of the paper is on lower bounds, the upper bound result on the estimation of $\hat{\theta}$ in learning mixture of Gaussians is of its independent interest. To obtain the upper bound on $\mathbb{E}\|\hat{\theta} - \theta\|_F^2$, we first establish the following lemma:

Lemma 19 *Let the mutual distance satisfy $D \geq c_1\sqrt{\log(nk/\delta)} \geq 3$ for a sufficiently large constant c_1 . With probability at least $1 - \delta$, the inequality $\|x_i - \theta_j\|_2 - \|x_i - \theta_{z_i}\|_2 \geq c_2(d \log(nk/\delta))^{-1/2}$ holds for a constant $c_2 > 0$, for any $i \in [n]$ and any $j \in [k] \setminus \{z_i\}$.*

The proof of this technical lemma is relegated to Appendix D. Lemma 19 shows that with high probability, the distance of a random sample to its true component mean is significantly less than the distance to any other means. Let $\hat{\theta}_j$ represent the j -th column of $\hat{\theta}$. When the sample size n is sufficiently large, the method of Hsu and Kakade (2013) guarantees that $\|\hat{\theta}_j - \theta_j\|_2 < o(d \log(nk/\delta))^{-1/2}$ for any $j \in [k]$. Thus, Lemma 19 implies that the distance of x_i to $\hat{\theta}_{z_i}$ is smaller than the distance to any other estimated centers. As a consequence, we may recover the membership of instances by computing the center that is the closest to them.

$$\hat{z}_i = \arg \min_{j \in [k]} \|x_i - \hat{\theta}_j\|_2.$$

According to Lemma 19, with high probability we have $\hat{z}_i = z_i$ for any $i \in [n]$. Given the membership, we refine the mean estimates by:

$$\hat{\theta}_j \leftarrow \frac{\sum_{i: \hat{z}_i=j} x_i}{|\{i: \hat{z}_i=j\}|}.$$

Since the membership is uniformly assigned, with high probability the sample size of the j -th Gaussian component is lower bounded by $\frac{2n}{k}$. Thus, with high probability the squared error of $\hat{\theta}_j$ will be upper bounded by $\mathcal{O}(dk/n)$. Since there are k components, the overall squared error is bounded by $\mathcal{O}(dk^2/n)$. Putting pieces together, we have an upper bound on the mean-squared error of parameter estimation.

Proposition 20 *Suppose that $d \geq 3k$ and n is greater than a fixed polynomial function of $(d, k, 1/\rho)$. Let the true parameter θ be ρ -perturbed from an arbitrary matrix $\theta^* \in \mathbb{R}^{d \times k}$. In addition, assume that the distances between the columns of θ^* are at least $D = c\sqrt{\log(nk)}$ for some universal constant c . Then there is a universal constant C such that the estimator $\hat{\theta}$ described above achieves mean-square error:*

$$\mathbb{E}[\|\hat{\theta} - \theta\|_F^2] \leq \frac{Cdk^2}{n}.$$

6.2 Minimax Risk of Smoothed Analysis

In this section, we formalize the notion of minimax risk under smoothed analysis. Similar to the classical statistical setting, the *minimax risk under smoothed analysis* can be defined in a game theoretic way. The learner first chooses an estimator $\hat{\theta}$, then the adversary chooses a parameter θ^* from the parameter space Θ , which is randomly perturbed to form the true parameter θ . The data X is generated with respect to θ . Under this random perturbation framework, the minimax risk is defined as:

$$R_{\text{minimax}} := \inf_{\hat{\theta}} \sup_{\theta^* \in \Theta} \mathbb{E}_{\theta} [L(\hat{\theta}(X), \theta)] \quad (56)$$

where $L(\cdot, \cdot)$ is the loss function. In our GMM application, the parameters are the means of mixture components. The parameter space is the set of means whose mutual distances are lower bounded by D . The true parameter is generated by a random Gaussian perturbation with variance ρ^2 . The loss is the Frobenius norm of the difference of matrices.

We note that the minimax risk (56) differs from the classical notion of minimax risk in that the adversary is not able to explicitly choose the true parameter θ . Instead, the true parameter is sampled from a prior distribution parameterized by θ^* . This Bayes nature makes it hard to lower bound the minimax risk (56) using the traditional Le Cam's or the Fano's method. In particular, both the Le Cam's method and the Fano's method lower bound the minimax risk by assuming a uniform prior over a carefully constructed discrete set. However, in our GMM setting, the prior distribution of parameter θ is always continuous.

Our Bayes risk lower bound naturally fits into the setting of smoothed analysis. Let w^* be an arbitrary prior distribution over θ^* . Since θ is perturbed from θ^* , the prior w^* induces a prior w over θ . It is sufficient to see that the Bayes risk with respect to w is a lower bound on the minimax risk (56). Thus, it suffices to lower bound the Bayes risk:

$$R_{\text{Bayes}}(w, L; \Theta) := \inf_{\hat{\theta}} \mathbb{E}_{\theta \sim w} [L(\hat{\theta}(X), \theta)].$$

For the GMM example, we construct the prior distribution w^* as follow: the j -th column of θ^* , namely the vector $\theta_j^* \in \mathbb{R}^d$, is sampled from the normal distribution $\mathcal{N}(D_{e_j}, I_{d \times d})$,

where e_j is the unit vector of the j -th coordinate. As a consequence, the prior distribution w samples the j -th column of θ from the normal distribution $\mathcal{N}(D_{e_j}; (1 + \rho^2)I_{d \times d})$.

In the GMM setting, the membership variables z_i are unknown to the estimator. If we assume that the memberships are given to the estimator, it makes the problem easier so that the associated Bayes risk is a smaller than or equal to the original Bayes risk. Since we want to derive a lower bound, we make the assumption that the memberships are given, then partition the instances into k disjoint subsets according to their memberships. Let the j -th subset S_j be defined as $S_j := \{x_i : z_i = j\}$. Conditioning on the memberships, the distributions of $\{(\theta_j, S_j)\}_{j=1}^k$ are mutually independent. Thus, we have

$$R_{\text{Bayes}}(w, L; \Theta) \geq \sum_{j=1}^k \inf_{\hat{\theta}_j} \mathbb{E}_{\theta_j \sim w_j} [L(\hat{\theta}_j(S_j), \theta_j)] \geq \sum_{j=1}^k \mathbb{E} \left[\inf_{\hat{\theta}_j} \mathbb{E}_{\theta_j \sim w_j} [L(\hat{\theta}_j(S_j), \theta_j)] | n_j \right] \quad (57)$$

where w_j is the prior distribution $\mathcal{N}(D_{e_j}; (1 + \rho^2)I_{d \times d})$ and n_j is the cardinality of S_j . We focus on the inner term on the right-hand side, namely $\inf_{\hat{\theta}_j} \mathbb{E}_{\theta_j \sim w_j} [L(\hat{\theta}_j(S_j), \theta_j)] | n_j$, and find that it is the Bayes risk of Gaussian mean estimation with n_j i.i.d. samples, with the true parameter θ_j satisfying a Gaussian prior w_j . This Bayes risk can be easily lower bounded by the techniques that we develop in this paper.

Lemma 21 *Suppose that the standard deviation of normal perturbation $\rho \leq 1$ and $n_j \geq 1$. For a universal constant c , the Bayes risk is lower bounded by*

$$R_{\text{Bayes}}(w_j, n_j) := \inf_{\hat{\theta}_j} \mathbb{E}_{\theta_j \sim w_j} [L(\hat{\theta}_j(S_j), \theta_j)] | n_j \geq \frac{cd}{\theta_j}.$$

Proof [Proof of Lemma 21]

We denote the distribution of instances in S_j by P_{θ_j} and let \mathcal{P} be the set of such distributions. Since the support of w_j is \mathbb{R}^d , we start by defining a prior whose support is an Euclidean ball of radius $\Gamma := \sqrt{2d}$. Let \bar{w} be the truncated prior satisfying:

$$\bar{w}(x) = \begin{cases} w_j(x)/c_1 & \text{if } \|x - D_{e_j}\|_2 \leq \Gamma \\ 0 & \text{otherwise.} \end{cases}$$

The normalization factor c_1 is equal to the total mass of w in the ball $\{x : \|x - D_{e_j}\|_2 \leq \Gamma\}$. It is straightforward to verify that the radius Γ is sufficiently large so that c_1 is lower bounded by a universal constant. The prior \bar{w} can be viewed as restricting the original prior in a finite radius. According to Remark 11, we may lower bound the Bayes risk by

$$R_{\text{Bayes}}(w_j, n_j) \geq c_1 \cdot R_{\text{Bayes}}(\bar{w}, n_j).$$

Thus, it suffices to lower bound the second term on the right-hand side.

We follow the similar steps of Example 1 to establish the lower bound. We start by upper bounding the terms $\sup_{\theta \in \mathcal{A}} \bar{w}(B_\Gamma(\theta, L))$ and the chi-squared informativity $I_{\chi^2}(\bar{w}; \mathcal{P})$. Using definition of the multivariate normal distribution, it is easy to see that

$$\sup_{\theta \in \mathcal{A}} \bar{w}(B_\Gamma(\theta, L)) = \bar{w}(B_\Gamma(D_{e_j}, L)) \leq \frac{V(\sqrt{L})}{c_1(2\pi(1 + \rho^2))^{d/2}}$$

where $V(\sqrt{t})$ represents the volume of the Euclidean ball of radius \sqrt{t} . Thus, there is a universal constant c_2 such that $\sup_{a \in A} \bar{w}(B_t(a, L)) \leq (c_2 \sqrt{t}/\Gamma)^d$. On the other hand, we follow the same steps of Example 1 to upper bound the chi-square informativity. Note that our setup has n_j i.i.d. observations, but in Example 1 there is only one observation. In this generalized setup, the chi-square distance $\chi^2(P_{\theta_0} \| P_{\theta'})$ is equal to $\exp(n_j \|\theta - \theta'\|_2^2 / \sigma^2) - 1$. Plugging this formula into the argument of Example 1, we obtain the upper bound $I_{\chi^2}(\bar{w}, \mathcal{P}) \leq (3e\Gamma \sqrt{n_j/d})^d - 1$.

Let I_{pr}^* be the obtained informativity upper bound. If we choose $t = cd/n_j$ for a sufficiently small constant $c > 0$, then we have $\sup_{a \in A} \bar{w}(B_t(a, L)) < \frac{1}{4}(1 + I_{\text{pr}}^*)^{-1}$. Corollary 12 then gives $R_{\text{Bayes}}(\bar{w}, n_j) \geq cd/n_j$. ■

Combining inequality (57) and Lemma 21, we have

$$R_{\text{Bayes}}(w, L; \Theta) \geq \sum_{j=1}^k \frac{cdk}{2n} \mathbb{P}(n_j \leq 2n/k).$$

Recall that every n_j satisfies a binomial distribution $B(n, 1/k)$, which has median $\lfloor n/k \rfloor$ or $\lceil n/k \rceil$, thus the probability $\mathbb{P}(n_j \leq 2n/k)$ will be at least $1/2$. It implies that the Bayes risk is lower bounded by $\Omega(dk^2/n)$. Putting pieces together, we have the following lower bound on the minimax risk.

Proposition 22 *Assume that the standard deviation of normal perturbation $\rho \leq 1$, then for some universal constant c the minimax risk of smoothed analysis is lower bounded by $R_{\text{minimax}} \geq c \frac{dk^2}{n}$.*

Comparing proposition 20 and proposition 22, we find that both the upper bound and the lower bound are tight. More precisely, under the assumptions of proposition 20, the minimax risk of smoothed analysis is precisely on the order of dk^2/n .

7. Bayes Risk Lower Bounds for Sparse Linear Regression

Linear regression is a canonical problem in machine learning and statistics. For a fixed design matrix $X \in \mathbb{R}^{n \times d}$ and an unknown parameter $\theta \in \mathbb{R}^d$, the learner observes a noise-corrupted response vector $y = X\theta + \varepsilon$, where ε satisfies an isotropic normal distribution $N(0, \sigma^2 I_{d \times d})$. The goal is to take the response vector as input and find an estimator $\hat{\theta} \in \mathbb{R}^d$ for the true parameter θ . The risk is measured either by the estimation error $I_{\text{est}}(\hat{\theta}, \hat{\theta}) := \|\hat{\theta} - \theta\|_2^2$, or by the prediction error $I_{\text{pre}}(\hat{\theta}, \hat{\theta}) := \|X\hat{\theta} - X\theta\|_2^2$. Both errors will be studied in this section.

For high-dimensional linear regression, the dimension d can be much greater than the sample size n . In order to prevent over-fitting, one needs to impose structural assumptions on the true parameter, for example, assuming that the number of non-zero entries in vector θ is at most k ($k \ll d$). Formally, we use $\mathbb{B}_0(k)$ to represent the set of k -sparse vectors in \mathbb{R}^d , and assume that $\theta \in \mathbb{B}_0(k)$. Under this setting, we want to compute an estimator $\hat{\theta} \in \mathbb{R}^d$ to minimize the estimation error or the prediction error. Note that the estimator $\hat{\theta}$ does not need to be k -sparse. Hence, our theoretical framework includes *improper learners* which are allowed to output non-sparse estimates whenever they achieve small risks.

The minimax risks of sparse linear regression have been well-studied. Under the same problem setting, Raskutti et al. (2011) proved information theoretic lower bounds on both the estimation error and the prediction error. Certain lower bounds have also been proved under the computation tractability constraint Zhang et al. (2014), or proved for the family of regularized M-estimators Zhang et al. (2015). All these lower bounds handle the worst-case scenario — given an arbitrary estimator, they prove the existence of a parameter θ that attains the lower bound. This setting might be too pessimistic in practice. The goal of this section is to study the Bayes risk of sparse linear regression under a natural prior, whose construction is described in the next subsection.

7.1 Prior Definition and Assumptions

We define a prior over k -sparse d -dimensional vectors for the true parameter $\theta \in \mathbb{R}^d$, referred to as distribution w , as follows:

1. Uniformly sample a subset of k indices from the integer set $\{1, 2, \dots, d\}$, naming this subset by K .
2. For every index $i \in K$, the coordinate θ_i is generated by sampling from the normal distribution $N(0, \tau^2)$. For any $i \notin K$, define $\theta_i := 0$.

Given an index set K , we use θ_K as a shorthand notation to denote the coordinates of the vector $\theta \in \mathbb{R}^d$ whose indices belong to the set K . Similarly, we use θ_{-K} to denote the subvector whose indices are not in K . Then the second step of the above generative process can be rephrased as generating $\theta_K \sim N(0, \tau^2 I_{k \times k})$ and defining $\theta_{-K} = 0$. It is clear that the sampled θ belongs to the k -sparse ℓ_0 -ball $\mathbb{B}_0(k) := \{\theta \in \mathbb{R}^d : \|\theta\|_0 \leq k\}$.

One may consider variants of the prior defined above. For example, one can assume that the number of non-zero entries of the vector θ is not exactly equal to k , but random sampled from a Poisson distribution with mean k . One may also redefine the prior of non-zero entries to be a non-Gaussian distribution. However, these variants don't add essential technical challenge to the analysis, thus we focus on the prior w as a concrete example for illustrating the general idea.

We make an additional assumption on the design matrix X that is important for characterizing the minimax risk (see, e.g. Raskutti et al., 2011), and in this section, we study their effects on the Bayes risk. Specifically, the design matrix X satisfies the *sparse eigenvalue conditions* with parameter (κ_u, κ_ℓ) if:

$$\kappa_\ell \|\beta\|_2 \leq \frac{\|X\beta\|_2}{\sqrt{n}} \leq \kappa_u \|\beta\|_2 \quad \text{for any } (2k)\text{-sparse vector } \beta \in \mathbb{R}^d. \quad (58)$$

Here, both κ_u and κ_ℓ are positive constants. As a concrete example, if entries of the matrix X are i.i.d. sampled from a normal distribution, then the matrix is called a *Gaussian random design*. This type of matrices have been extensively studied for sparse linear regression (Candes et al., 2006; Guédon et al., 2008), and proved to satisfy condition (58) with $\kappa_u/\kappa_\ell = \mathcal{O}(1)$ (Raskutti et al., 2010). For the rest of this section, we assume that the design matrix X satisfies the condition (58).

7.2 Bayes Risk Lower Bounds

For sparse linear regression, we denote the parameter space and action space by $\Theta = \mathbb{B}_0(k)$ and $\mathcal{A} = \mathbb{R}^d$, respectively. We present a Bayes risk lower bound with respect to the prior distribution defined in Section 7.1, then demonstrate its consequences.

Theorem 23 *Assume that the design matrix X satisfies the sparse eigenvalue condition (58), and that $d > k^3$. There are universal constants $c', c'' > 0$ such that for any $\tau > 0$, we have Bayes risk lower bounds: $R_{\text{Bayes}}(w, L_{\text{est}}; \Theta) \geq c' T(\tau)$ and $R_{\text{Bayes}}(w, L_{\text{pre}}; \Theta) \geq c'' \kappa_u^2 T(\tau)$, where $T(\tau)$ is a term defined by*

$$T(\tau) := k\tau^2 \max \left\{ \frac{1}{1 + \kappa_u^2 \sigma^2 \tau^2 / \sigma^2}, \exp \left(-\frac{4\kappa_u^2 n}{\sigma^2} \left[\tau^2 - \frac{\sigma^2 \log(d/k)}{16\kappa_u^2 n} \right]_+ \right) \right\}. \quad (59)$$

The proof of Theorem 23 follows the general strategy that we sketched in earlier sections: first, we bound the mutual informativity using the techniques described in Section 5, then we upper bound the probability $\sup_{a \in \mathcal{A}} w(B_t(a, L))$ for a specific scalar $t > 0$. Combining the two upper bounds with Corollary 12 establishes the theorem. See Appendix E for the proof. We make a few important remarks of this result in the below.

Estimation versus prediction By Theorem 23, the lower bounds on the estimator error and the prediction error differ by a factor κ_u^2 . As a consequence, if we multiply a constant to the design matrix, then the term κ_u^2 will also be scaled. If the scalar is very small, then the lower bound on the prediction error will be close to zero, but the lower on the estimation error won't. These are the right scaling for both risks. Indeed, when the design matrix converges to an all-zero matrix, the true parameters will be hard to identify, but the constant estimator $\hat{\theta} \equiv 0$ will be able to achieve a small prediction error.

Comparison with minimax risk lower bounds It is worth comparing Theorem 23 with the well-studied minimax risk lower bound. Under the sparse eigenvalue condition (58), Raskutti et al. (2011) proved the follow minimax risk lower bound:

$$\inf_{\hat{\theta}} \max_{\theta \in \mathbb{B}_0(k)} \mathbb{E}[L_{\text{est}}(\hat{\theta}, \hat{\theta})] \geq c' \frac{\sigma^2 k \log(d/k)}{\kappa_u^2 n} \quad \text{and} \quad \inf_{\hat{\theta}} \max_{\theta \in \mathbb{B}_0(k)} \mathbb{E}[L_{\text{pre}}(\hat{\theta}, \hat{\theta})] \geq c'' \frac{\kappa_u^2 \sigma^2 k \log(d/k)}{\kappa_u^2 n}, \quad (60)$$

where c' and c'' are universal constants. These bounds are matched by Theorem 23. In particular, if we assume $d > k^3$ and consider the prior distribution with variance:

$$\tau^2 = \begin{pmatrix} \tau_u^2 \\ \tau_s^2 \end{pmatrix} := \begin{pmatrix} \sigma^2 \log(d/k) \\ 16\kappa_u^2 n \end{pmatrix}, \quad (61)$$

then expression (59) implies $T(\tau) = k\tau^2$, and as a consequence, we have

$$R_{\text{Bayes}}(w, L_{\text{est}}; \Theta) \geq c' \frac{\sigma^2 k \log(d/k)}{\kappa_u^2 n} \quad \text{and} \quad R_{\text{Bayes}}(w, L_{\text{pre}}; \Theta) \geq c'' \frac{\kappa_u^2 \sigma^2 k \log(d/k)}{\kappa_u^2 n}, \quad (62)$$

where c' and c'' are universal constants. The minimax risk lower bounds (60) and the Bayes risk lower bounds (62) thus match by a universal constant factor. Therefore, using our

technique, we can directly obtain this classical minimax result on sparse linear regression. It is worth noting that the lower bounds of Raskutti et al. (2011) were proved by constructing a uniform prior over a discrete packing set over the parameter space. The existence of the proper packing set was proved in a non-constructive, worst-case fashion, which might be too pessimistic in practice. In contrast, our lower bound was established for a realistic and flexible prior which admits a simple closed-form definition and allows for different levels of variance. The theorem also shows that the prior w with the variance level (61) is in fact a *least favorable prior* for sparse linear regression.

Bayes risk on the spectrum of priors Besides the least-favorable setting (61), let us consider the Bayes risk under other choices of the parameter τ^2 . When $\tau^2 < \tau_u^2$, Theorem 23 implies

$$R_{\text{Bayes}}(w, L_{\text{est}}; \Theta) \geq c' k\tau^2 \quad \text{and} \quad R_{\text{Bayes}}(w, L_{\text{pre}}; \Theta) \geq c'' \kappa_u^2 k\tau^2. \quad (63)$$

When $\tau^2 \rightarrow +\infty$, Theorem 23 implies

$$R_{\text{Bayes}}(w, L_{\text{est}}; \Theta) \geq c' \frac{k\sigma^2}{\kappa_u^2 n} \quad \text{and} \quad R_{\text{Bayes}}(w, L_{\text{pre}}; \Theta) \geq c'' \frac{\kappa_u^2 k\sigma^2}{\kappa_u^2 n}. \quad (64)$$

In both cases, the Bayes risk lower bounds can be significantly smaller than the minimax risk. We argue that these lower bounds are essentially tight under specific assumptions. That is, when taking the prior information into account, we can indeed achieve better rates than the minimax rate.

First, notice that the upper bound:

$$\mathbb{E}_{\theta \sim w}[L_{\text{est}}(\hat{\theta}, \hat{\theta})] \leq k\tau^2 \quad \text{and} \quad \mathbb{E}_{\theta \sim w}[L_{\text{pre}}(\hat{\theta}, \hat{\theta})] \leq \kappa_u^2 k\tau^2.$$

can always be achieved using the constant estimator $\hat{\theta} \equiv 0$. It means that for the case of $\tau^2 < \tau_u^2$, the lower bounds (63) are tight under the assumption $\kappa_u/\kappa_\ell = \mathcal{O}(1)$.

For the case of $\tau^2 \rightarrow +\infty$, we consider the ℓ_0 -norm constrained estimator:

$$\hat{\theta} := \arg \inf_{\beta \in \mathbb{B}_0(k)} \|X\beta - y\|_2^2. \quad (65)$$

Whenever $\kappa_u/\kappa_\ell = \mathcal{O}(1)$, Raskutti et al. (2011) showed that the estimator (65) achieves an error bound $\|\hat{\theta} - \theta\|_2^2 \leq c \frac{k \log(d)}{\kappa_u^2 n}$ with high probability for a constant $c > 0$. Suppose that $\tau^2 = C \frac{k \log(d)}{n}$ with a scaling factor $C > c$. For any $i \in K$, the expectation of θ_i^2 is equal to τ^2 , so that the probability of $\theta_i^2 \leq c \frac{k \log(d)}{n}$ is bounded by $\mathcal{O}(c/C)$. It means that by choosing a large enough C (specifically, choosing $C \gg ck$), the lower bound $\theta_i^2 > c \frac{k \log(d)}{n}$ will hold for every $i \in K$ with a probability close to 1. Combining this fact with the bound $\|\hat{\theta} - \theta\|_2^2 \leq c \frac{k \log(d)}{n}$, we find that the support of $\hat{\theta}$ must agree with K , so that the estimator must satisfy:

$$\hat{\theta}_K = \arg \inf_{\beta \in \mathbb{R}^k} \|X_K \beta - y\|_2^2 \quad \text{and} \quad \hat{\theta}_{-K} = 0,$$

where X_K is a submatrix of X consisting of columns indexed by K . In other words, the vector $\hat{\theta}_K$ is the least-square estimator for a k -dimensional linear regression problem. For estimators taking this form, both the estimation error and the prediction error are known to match the lower bound (64) with high probability.

8. Conclusions

In this paper, we presented lower bounds for the Bayes risk in abstract decision-theoretic problems. Our bounds are quite general and only require upper bounds on $\sup_{a \in \mathcal{A}} w(B_t(a, L))$ and the f -informativity $I_f(w, \mathcal{P})$ for their application. Because of the generality, the bounds are not always tight however. For example, the bounds involve $\sup_{a \in \mathcal{A}} w(B_t(a, L))$ and this quantity becomes large when the prior w has a spike. In such situations, our main Bayes risk lower bound in Theorem 9 will not be tight. In specific examples, this looseness can be remedied by adhoc fixes such as the one described in Remark 11. Obtaining tight lower bounds for the Bayes risk in the generality considered in this paper is a challenging open problem.

Acknowledgments

Adityanand Guntuboyina is supported by NSF Grant DMS-1309356. The authors would like to thank Michael I. Jordan and Sivaraman Balakrishnan for helpful discussions.

Appendix A. Proofs and Additional Results for Section 3 on Bayes Risk Lower Bound for Zero-one Loss

A.1 Proof of Lemma 1

Recall the expression (13) of $\phi_f(a, b)$. We first fix b and show that $g(a) : a \mapsto \phi_f(a, b)$ is a non-increasing for $a \in [0, b]$. There is nothing to prove if $b = 0$ so let us assume that $b > 0$. We will consider the cases $0 < b < 1$ and $b = 1$ separately. For $0 < b < 1$, note that for every $a \in (0, b]$, we have,

$$g'_L(a) = f'_L\left(\frac{a}{b}\right) - f'_R\left(\frac{1-a}{1-b}\right),$$

where g'_L and f'_L represent left derivatives and f'_R represents right derivative (note that f'_L and f'_R exist because of the convexity of f). Because $\frac{a}{b} \leq \frac{1-a}{1-b}$ for every $0 \leq a \leq b$ and f is convex, we see that

$$g'_L(a) \leq f'_R\left(\frac{a}{b}\right) - f'_R\left(\frac{1-a}{1-b}\right) \leq 0$$

for every $a \in (0, b]$ which implies that $g(a)$ is non-increasing on $[0, b]$.

When $b = 1$, we have $g'_L(a) = f'_L(a) - f'(\infty)$ which is always ≤ 0 because f is convex (note that $f'(\infty) = \lim_{x \uparrow \infty} f(x)/x = \lim_{x \uparrow \infty} (f(x) - f(1))/(x - 1)$).

The convexity and continuity of g follow from the convexity of f and the expression for ϕ_f .

Next, we fix a and show that $h(b) : b \mapsto \phi_f(a, b)$ is non-decreasing for $b \in [a, 1]$. For every $b \in [a, 1)$, we have,

$$h'_R(b) = f\left(\frac{a}{b}\right) - \frac{a}{b}f'_L\left(\frac{a}{b}\right) - f\left(\frac{1-a}{1-b}\right) + \frac{1-a}{1-b}f'_R\left(\frac{1-a}{1-b}\right), \quad (66)$$

where h'_R represents the right derivative of h . By the convexity of f ,

$$f\left(\frac{a}{b}\right) - f\left(\frac{1-a}{1-b}\right) \geq f'_R\left(\frac{1-a}{1-b}\right) \left(\frac{a}{b} - \frac{1-a}{1-b}\right). \quad (67)$$

Combining (66) with (67), we obtain that,

$$h'_R(b) \geq \frac{a}{b} \left(f'_R\left(\frac{1-a}{1-b}\right) - f'_L\left(\frac{a}{b}\right) \right) \geq \frac{a}{b} \left(f'_L\left(\frac{1-a}{1-b}\right) - f'_L\left(\frac{a}{b}\right) \right) \geq 0,$$

where the last inequality is because that $\frac{a}{b} \leq \frac{1-a}{1-b}$ for every $0 \leq a \leq b$ and f is convex. The non-negativity of $h'_R(b)$ implies that $h(b)$ is non-decreasing on $[a, 1]$.

A.2 A Variant of Fano's Inequality from Braun and Pokutta (2014)

One of the main results in Braun and Pokutta (2014) (Proposition 2.2) establishes the following variant of Fano's inequality. Consider the setting of Lemma 3. In particular, recall the quantities R^0 and $R^0_{\mathcal{Q}}$ from (21) and also the sets $\mathcal{B}(a)$, $a \in \mathcal{A}$ from (16). (Braun and Pokutta, 2014, Proposition 2.2) proved the following: for any decision rule \mathfrak{D} ,

$$R^0 \geq \frac{-I(w, \mathcal{P}) - H(R^0)}{\log[(1 - w_{\min})/w_{\max}]}, \quad (68)$$

where $H(x) := -x \log x - (1-x) \log(1-x)$, $w_{\min} := \inf_{a \in \mathcal{A}} w(B(a))$ and $w_{\max} := \sup_{a \in \mathcal{A}} w(B(a))$.

Below we provide a proof of this inequality using Lemma 3. The proof given in Braun and Pokutta (2014) is quite different proof. Using (20) from Lemma 3 with $f(x) = x \log x$, we have for any decision rule

$$\int_{\Theta} D_f(P_\theta \| Q) w(d\theta) \geq R^3 \log \frac{R^3}{R_0^3} + (1 - R^3) \log \frac{1 - R^3}{1 - R_0^3}.$$

We can rewrite this as

$$\int_{\Theta} D_f(P_\theta \| Q) w(d\theta) \geq -H(R^3) - R^3 \log R_0^3 - (1 - R^3) \log(1 - R_0^3) \quad (69)$$

where $H(x) := -x \log x - (1-x) \log(1-x)$. Since L in Lemma 3 is zero-one valued,

$$R_0^3 = 1 - \mathbb{E}_Q w(B(\mathfrak{q}(X))) \quad (70)$$

where \mathbb{E}_Q denotes expectation taken under $X \sim Q$ and $B(\mathfrak{q}(X))$ is defined in (16). As a result, we have

$$1 - \max_{a \in \mathcal{A}} w(B(a)) \leq R_0^3 \leq 1 - \min_{a \in \mathcal{A}} w(B(a)). \quad (71)$$

Using the bounds in (71) on the right hand side of (69), we deduce

$$\int_{\Theta} D_f(P_\theta \| Q) w(d\theta) \geq -H(R^3) - R^3 \log(1 - w_{\min}) - (1 - R^3) \log w_{\max}.$$

where $w_{\min} := \inf_{a \in \mathcal{A}} w(B(a))$ and $w_{\max} := \sup_{a \in \mathcal{A}} w(B(a))$ for notational simplicity. Taking the infimum on the left hand side above over all probability measures Q , we obtain

$$I(w, \mathcal{P}) \geq -H(R^3) - R^3 \log(1 - w_{\min}) - (1 - R^3) \log(w_{\max}).$$

Provided $w_{\min} + w_{\max} < 1$, one can rewrite the above inequality as (68). This completes the proof of (68).

A.3 Proof of Corollary 7

1. **Proof of inequality (26):** Applying Theorem 2 with $f(x) = x^2 - 1$, we obtain

$$I_{\chi^2}(w, \mathcal{P}) \geq \frac{(R_0 - R)^2}{R_0(1 - R_0)}$$

Because $R \leq R_0$, we can invert the above to obtain (26).

2. **Proof of inequality (27):** Theorem 2 with $f(x) = |x - 1|/2$ gives

$$I_{TV}(w, \mathcal{P}) \geq \frac{R_0}{2} \left| \frac{R}{R_0} - 1 \right| + \frac{1 - R_0}{2} \left| \frac{1 - R}{1 - R_0} - 1 \right| = R_0 - R,$$

where the last equality uses the fact that $R \leq R_0$. Inverting the above inequality, we obtain (27).

3. **Proof of inequality (28):** Theorem 2 with $f(x) = f_{1/2}(x) = 1 - \sqrt{x}$ gives

$$I_{f_{1/2}}(w, \mathcal{P}) \geq 1 - \sqrt{RR_0} - \sqrt{(1 - R)(1 - R_0)}. \quad (72)$$

Assume that P_θ has density p_θ with respect to a common dominating measure μ . We shall show below that

$$I_{f_{1/2}}(w, \mathcal{P}) = 1 - \sqrt{\int_{\mathcal{X}} u^2 d\mu} \quad \text{where } u := \int_{\Theta} \sqrt{p_\theta} w(d\theta). \quad (73)$$

To see this, fix a probability measure Q that has a density q with respect to μ . We can then write

$$\int_{\Theta} D_{f_{1/2}}(P_\theta \| Q) w(d\theta) = 1 - \int_{\mathcal{X}} \sqrt{q} \left(\int_{\Theta} \sqrt{p_\theta} w(d\theta) \right) d\mu = 1 - \int_{\mathcal{X}} \sqrt{q u^2} d\mu$$

It follows then from the Cauchy-Schwarz inequality that

$$\int_{\Theta} D_{f_{1/2}}(P_\theta \| Q) w(d\theta) = 1 - \int_{\mathcal{X}} \sqrt{q u^2} d\mu \geq 1 - \sqrt{\int_{\mathcal{X}} u^2 d\mu},$$

with equality holding when q is proportional to u^2 . This proves (73). We now see that

$$\int_{\mathcal{X}} u^2 d\mu = \int_{\Theta} \int_{\Theta} \int_{\mathcal{X}} \sqrt{p_\theta} \sqrt{p_{\theta'}} d\mu w(d\theta) w(d\theta') = 1 - \frac{1}{2} h^2 \quad (74)$$

where h^2 is defined as

$$h^2 = \int_{\Theta} \int_{\Theta} H^2(P_\theta \| P_{\theta'}) w(d\theta) w(d\theta'). \quad (75)$$

This, together with (72) and (73), gives the inequality

$$\sqrt{RR_0} + \sqrt{(1 - R)(1 - R_0)} \geq \sqrt{1 - \frac{h^2}{2}} \quad (76)$$

Now under the assumption $h^2 \leq 2R_0$, the right hand side of the inequality (76) lies between $\sqrt{1 - R_0}$ and 1. On the other hand, it can be checked that, as a function in R , the left hand side of (76) is strictly increasing from $\sqrt{1 - R_0}$ (at $R = 0$) to 1 at $(R = R_0)$. Therefore, from (76), we know that $R \geq \hat{R}$ where $\hat{R} \in [0, R_0]$ is the solution to the equation obtained by replacing the inequality (76) with an equality. One can solve this equation and obtain two solutions. One of two solutions can be discarded by the fact that $R \leq R_0$. The other solution is given by:

$$\hat{R} = R_0 - (2R_0 - 1) \frac{h^2}{2} - \sqrt{R_0(1 - R_0) \sqrt{h^2(2 - h^2)}}$$

and thus we have $R \geq \hat{R}$ which proves inequality (28).

We note that the lower bound on R in (28) only holds under the condition $h^2 \leq 2R_0$. When $h^2 > 2R_0$, inequality (28) holds for every $R \in [0, R_{Q^*}]$ and thus cannot provide a non-trivial lower bound on R . As an example, when $\Theta = \mathcal{A} = \{1, \dots, N\}$, $L(\theta, a) = \mathbb{I}\{\theta \neq a\}$ and w is the uniform prior on Θ , it is easy to see that $R_0 = 1 - (1/N)$ and

$$h^2 = \frac{1}{N^2} \sum_{\theta \neq \theta'} H^2(P_\theta \| P_{\theta'}) \leq 2 \frac{N(N-1)}{N^2} = 2R_{Q^*}. \quad (77)$$

Inequality (28) therefore is equivalent to

$$R \geq 1 - \frac{1}{N} - \frac{N-2h^2}{N} - \frac{\sqrt{N-1}}{N} \sqrt{h^2(2-h^2)}.$$

This recovers the result in Example II.6 in Guntuboyina (2011b).

A.4 Derivations of Le Cam's Inequality (Two Hypotheses) and Assouad's Lemma and other Results from Corollary 7

To demonstrate the application of Corollary 7, we apply it to derive the two hypotheses version of Le Cam's inequality (with total variation distance) and Assouad's lemma (see Theorem 2.12 in (Tsybakov, 2010)).

The simplest version of the Le Cam's inequality, the so-called two-point argument, is an easy corollary of (27). Indeed, applying (27) with $\Theta = \mathcal{A} = \{\theta_0, \theta_1\}$, $L(\theta, a) = \mathbb{I}\{\theta \neq a\}$ and $w\{0\} = w\{1\} = 1/2$ (and note that $R_0 = 1/2$), we obtain that for any distribution Q on \mathcal{X} ,

$$\frac{1}{2} (\|P_{\theta_0} - Q\|_{TV} + \|P_{\theta_1} - Q\|_{TV}) \geq I_{TV}(w, \mathcal{P}) \geq 1/2 - R.$$

Taking $Q = (P_{\theta_0} + P_{\theta_1})/2$, we obtain Le Cam's inequality:

$$R_{\min} \geq \frac{1}{2} (1 - \|P_{\theta_0} - P_{\theta_1}\|_{TV}). \quad (78)$$

The more involved Le Cam's inequality considers $\Theta = \mathcal{A} = \Theta_0 \cup \Theta_1$ for two disjoint subsets Θ_0 and Θ_1 and loss function $L(\theta, a) = \mathbb{I}\{\theta \in \Theta_1, a \in \Theta_2\} + \mathbb{I}\{\theta \in \Theta_2, a \in \Theta_1\}$. The inequality states that for every pair of probability measures w_0 and w_1 concentrated on Θ_0 and Θ_1 respectively,

$$R_{\min} \geq \frac{1}{2} (1 - \|m_0 - m_1\|_{TV}) \quad (79)$$

where m_0 and m_1 are marginal densities given by $m_\tau(x) = \int p_\theta(x) w_\tau(d\theta)$ for $\tau = 0, 1$. To prove (79), consider the prior $w = (w_0 + w_1)/2$. Under this prior, the problem is easily converted to the previous binary testing problem. In particular, the data generating process under the prior w can be viewed as first sampling $\tau \sim \text{Uniform}\{0, 1\}$ and then $X \sim m_\tau$. The decision $a \in \mathcal{A}$ can be converted into the binary decision $\hat{\tau} = \mathbb{I}\{a \in \Theta_1\}$. The loss function is $L(\tau, \hat{\tau}) = \mathbb{I}\{\tau \neq \hat{\tau}\}$. The Bayes risk under the prior w can be re-written as,

$$R_{\text{Bayes}}(w, L; \Theta) = \frac{1}{2} \inf_{\tau} \sum_{\tau=0,1} \int_{\mathcal{X}} \mathbb{I}(\tau \neq \hat{\tau}(x)) m_\tau(x) \mu(dx), \quad (80)$$

which has the same form as the Bayes risk in the earlier binary testing problem. Applying the same argument as for proving (78), we obtain the lower bound on the Bayes risk in (80), $R_{\text{Bayes}}(w, L; \Theta) \geq \frac{1}{2} (1 - \|m_0 - m_1\|_{TV})$, which further implies (79).

Another classical minimax inequality involving the total variation distance is Assouad's inequality (Assouad, 1983) which states that if $\Theta = \mathcal{A} = \{0, 1\}^d$ and the loss function L is defined by the Hamming distance, i.e., $L(\theta, a) = \sum_{i=1}^d \mathbb{I}\{\theta_i \neq a_i\}$, then

$$R_{\min} \geq \frac{d}{2} \min_{L(\theta, \theta')=1} (1 - \|P_\theta - P_{\theta'}\|_{TV}). \quad (81)$$

This inequality is also a consequence of (27): let w be the uniform probability measure on Θ and $L_1(\theta, a) = \mathbb{I}\{\theta_1 \neq a_1\}$. Under w , the marginal distribution of the first coordinate is $w_1\{0\} = w_1\{1\} = 1/2$. Let $m_\tau(x) := \sum_{\theta: \theta_1=\tau} p_\theta(x)/2^{d-1}$ for $\tau \in \{0, 1\}$ be the corresponding marginal density of X and let $Q(x) = \frac{1}{2} (m_0(x) + m_1(x))$. Applying the same argument as for proving (78), we obtain that the minimax risk for the zero-one valued loss function $L_1(\theta, a)$ is bounded below by $\frac{1}{2} (1 - \|m_0 - m_1\|_{TV}) \geq \frac{1}{2} \min_{L(\theta, \theta')=1} (1 - \|P_\theta - P_{\theta'}\|_{TV})$. Repeating this argument for $L_i(\theta, a) := \mathbb{I}\{\theta_i \neq a_i\}$ for $i = 2, \dots, d$ and adding up the resulting bounds, we obtain (81).

By using Le Cam's inequality (see, e.g., Lemma 2.3 in (Tsybakov, 2010)) which states that:

$$\|P_\theta - P_{\theta'}\|_{TV} \leq \sqrt{H^2(P_\theta \| P_{\theta'}) \left(1 - \frac{1}{4} H^2(P_\theta \| P_{\theta'})\right)},$$

the inequality in (81) further implies the Hellinger distance version of Assouad's inequality in the book Tsybakov (2010, Theorem 2.12), i.e.,

$$R_{\min} \geq \frac{d}{2} \min_{L(\theta, \theta')=1} \left\{ 1 - \sqrt{H^2(P_\theta \| P_{\theta'}) \left(1 - \frac{1}{4} H^2(P_\theta \| P_{\theta'})\right)} \right\}. \quad (82)$$

A.5 Comparison of the Bounds for Different Divergences

We provide some qualitative comparisons of Bayes risk lower bounds given by Theorem 2 for different power divergences. In particular, let us consider the discrete setting where $\Theta = \mathcal{A} = \{\theta_1, \dots, \theta_N\}$, $L(\theta, a) = \mathbb{I}\{\theta \neq a\}$, and w is the discrete uniform. Note that in such a "multiple testing problem" setup, R_0 is equal to $1 - (1/N)$. We take N sufficiently large so that R_0 is close to 1. To establish minimax lower bounds, a typical approach is to reduce the estimation problem to a multiple hypotheses testing problem in the aforementioned setup, then try to prove that the Bayes risk $R \geq c > 0$ (see Section 2.2. in Tsybakov (2010)). Without loss of generality, we take $c = 1/2$ and we shall see how the three inequalities (25), (26) and (28) work to establish $R \geq 1/2$.

Let us start with (25) corresponding to KL divergence, which is equivalent to the classical Fano's inequality (3) in the discrete setting. To establish $R \geq 1/2$, the following condition should hold:

$$I(w, \mathcal{P}) \leq \frac{1}{2} \log \left(\frac{N}{4} \right). \quad (83)$$

We remark that $I(w, \mathcal{P})$ is at most $\log N$ even if every the pairwise KL divergence $D(P_\theta \| P_{\theta'})$ equals ∞ for $i \neq j$. This fact will be clear from the inequality (47) from Section 5 (let $M = N$

and $Q_j = P_{\theta_j}$ for $1 \leq j \leq M$). The upper bound on $I(w, \mathcal{P})$ in (47) further provides a sufficient condition to verify (83).

Now we turn to (26) corresponding to the chi-squared divergence. Since $R_0 = 1 - (1/N)$, inequality (26) implies a sufficient condition for $R \geq 1/2$:

$$I_{\chi^2}(w, \mathcal{P}) \leq \frac{N^2}{N-1} \left(\frac{1}{2} - \frac{1}{N} \right)^2. \quad (84)$$

When N is large, the above condition is equivalent to $I_{\chi^2}(w, \mathcal{P}) \leq N/4$. Note that the maximum possible value of $I_{\chi^2}(w, \mathcal{P})$ in this discrete setting is $N-1$ (even when $\chi^2(P_{\theta_i} \| P_{\theta_j}) = \infty$ for every $i \neq j$) and this follows from our upper bounds on f -informativity for a class of power divergences in (50) (see Section 5).

The conditions (83) and (84) don't imply each other. The chi-squared divergence is always greater than the KL divergence (see Lemma 2.7 in Tsybakov (2010)), but the upper bound required by (84) is also weaker than that required by (83). For both divergences, constructing more hypotheses (i.e., choosing $N > 2$) is often helpful for showing $R \geq 1/2$.

For the Hellinger distance (inequality (28)), we claim that it gives no more useful bounds than those obtained by a simple two point argument. To see this, since $R_0 = 1 - (1/N)$, inequality (28) implies

$$R \geq 1 - \frac{1}{N} - \frac{N-2}{2} \frac{h^2}{N} - \frac{\sqrt{N-1}}{2} \sqrt{h^2(2-h^2)}$$

where $h^2 = \sum_{i,j} H^2(P_{\theta_i} \| P_{\theta_j})/N^2$. When N is large, the above inequality reduces to effectively $R \geq 1 - (h^2/2)$. Therefore a sufficient condition for $R \geq 1/2$ is $h^2 \leq 1$, which is equivalent to,

$$\frac{1}{N(N-1)/2} \sum_{i < j} H^2(P_{\theta_i} \| P_{\theta_j}) \leq \frac{N}{N-1}.$$

When N is large, the above displayed condition implies the existence of $i < j$ for which $H^2(P_{\theta_i} \| P_{\theta_j}) \leq 1$. Let \tilde{w} denote the prior $\tilde{w}^i = \tilde{w}^j = 1/2$. It is easy to see that the Bayes risk for \tilde{w} equals $R_{\text{Bayes}}(\tilde{w}) = \frac{1}{2} (1 - \|P_{\theta_i} - P_{\theta_j}\|_{TV})$. By Le Cam's inequality (see Lemma 2.3 in Tsybakov (2010)), we have,

$$R_{\text{Bayes}}(\tilde{w}) \geq \frac{1}{2} \left(1 - H(P_{\theta_i} \| P_{\theta_j}) \sqrt{1 - \frac{H^2(P_{\theta_i} \| P_{\theta_j})}{4}} \right)$$

Since $H(P_{\theta_i} \| P_{\theta_j}) \leq 1$, it is easy to verify from the above that $R_{\text{Bayes}}(\tilde{w}) \geq 1/8$. Therefore in this discrete setting, if inequality (28) implies $R_{\text{Bayes}}(\tilde{w}) \geq 1/2$, then there is a much simpler two point prior \tilde{w} for which $R_{\text{Bayes}}(\tilde{w}) \geq 1/8$. It shows that for Hellinger distance, considering $N > 2$ hypotheses is not more useful than using a pair of hypotheses. The reason is that the Hellinger informativity can be written as an expression involving pairwise Hellinger distances. In particular, it can be seen from the proof of inequality (28) that

$$I_{H^{1/2}}(w, \mathcal{P}) = 1 - \left(1 - \frac{1}{2N^2} \sum_{i,j} H^2(P_{\theta_i} \| P_{\theta_j}) \right)^{1/2}.$$

In contrast, the mutual information, $I(w, \mathcal{P})$, cannot be written in terms of $D(P_{\theta_i} \| P_{\theta_j})$ for $i \neq j$ (recall that $I(w, \mathcal{P})$ is always at most $\log N$ even when $D(P_{\theta_i} \| P_{\theta_j}) = \infty$ for all $i \neq j$). The same holds for $I_{\chi^2}(w, \mathcal{P})$ as well (which is always at most $N-1$ even if $\chi^2(P_{\theta_i} \| P_{\theta_j}) = \infty$ for all $i \neq j$).

If the eventual goal of obtaining Bayes risk lower bounds is to obtain lower bounds up to multiplicative constants on the minimax risk, then the bound in (28) gives no more useful bounds than those obtained by the simple two point argument. In this sense, inequality (28) induced by Hellinger distance is not as useful as inequalities (25) and (26). In fact, the Hellinger distance is seldom used in lower bounding minimax risk involving many hypotheses (for example, none of the minimax rates in the examples of Tsybakov (2010) involving multiple hypotheses testing are established via Hellinger distance).

Appendix B. Proofs and Additional Results for Section 5 on Upper Bounds on f -informativity

B.1 Proof of Lemma 15

Let $\phi(t) \equiv t^r$ with $\phi'(t) = r t^{r-1}$ and $\phi''(t) = r(r-1)t^{r-2}$ and $\varphi(t) = t^{1/r}$ with $\varphi'(t) = \frac{1}{r} t^{(1-r)/r}$. Then

$$f(w) = \varphi \left(\int_T \phi(u(t)) \mu(dt) \right).$$

To prove the concavity of $f(w)$, considering the scalar function

$$h(s) = \varphi \left(\int_T \phi(u(t) + sv(t)) \mu(dt) \right), \quad (85)$$

for arbitrary $u, v \in L^r_{\mu}(T)$. We notice that concavity of f is equivalent to concavity at zero for all functions of the form h , and we therefore only have to show that $h''(0) \leq 0$. Let $g(s) = \int_T \phi(u(t) + sv(t)) \mu(dt)$,

$$\begin{aligned} h'(s) &= \varphi'(g(s)) \int_T \phi'(u(t) + sv(t)) v(t) \mu(dt) \\ h''(s) &= \varphi''(g(s)) \left(\int_T \phi'(u(t) + sv(t)) v(t) \mu(dt) \right)^2 \\ &\quad + \varphi'(g(s)) \int_T \phi''(u(t) + sv(t)) v^2(t) \mu(dt) \end{aligned}$$

By plugging in the definitions of $\phi(t)$, $\varphi(t)$, $g(s)$ and setting $s = 0$, we have

$$h''(0) = \frac{1-r}{f(w)} \left(\left(\int_T (f'(u))^{1-r} \int_T u(t)^{r-1} v(t) \mu(dt) \right)^2 - f(w)^{2-r} \int_T u(t)^{r-2} v^2(t) \mu(dt) \right)$$

Applying the Cauchy-Schwarz inequality

$$\left(\int_T a(t) b(t) \mu(dt) \right)^2 \leq \left(\int_T a(t)^2 \mu(dt) \right) \left(\int_T b(t)^2 \mu(dt) \right)$$

with $a(t) = \left(\frac{f(u)}{u(t)} \right)^{-r/r}$ and $b(t) = v(t) \left(\frac{f(u)}{u(t)} \right)^{1-r/r}$ and noticing that $r < 1$, we have $h''(0) \leq 0$, which completes the proof.

B.2 Example Demonstrating the Effectiveness of Theorem 14

In this example, we show the tightness of the upper bound in (49) in terms of chi-squared divergence ($\alpha = 2$). In particular, let the distribution P be the n -fold product of $N(0, 1)$ and Q_ξ be the n -fold product of $N(\xi, 1)$ where $\xi \sim N(0, 1)$. It is straightforward to show that the marginal distribution \bar{Q} is a n -dimensional Gaussian distribution with mean $\mathbf{0}$ and covariance matrix $I_n + \mathbf{1}_n \mathbf{1}_n^T$, where $\mathbf{1}_n$ denotes the n -dimensional all one vector and I_n the $n \times n$ identity matrix.

Since $\chi^2(P\|Q_\xi) = \exp(n\xi^2) - 1$, the right hand side of (49) equals to $\sqrt{2n+1} - 1$. The term $\chi^2(P\|\bar{Q})$ on the left hand side of (49) is difficult to evaluate. However, we can lower bound $\chi^2(P\|\bar{Q})$ using the following standard inequality $\exp(D(P\|\bar{Q})) - 1 \leq \chi^2(P\|\bar{Q})$ (see Lemma 2.7 in Tsybakov (2010)). By the closed-form expression for KL divergence between two multivariate Gaussian distributions, we have $D(P\|\bar{Q}) = \frac{1}{2}(\log(n+1) - n/(n+1))$ and thus

$$e^{-1/2}\sqrt{n+1} - 1 \leq \exp(D(P\|\bar{Q})) - 1 \leq \chi^2(P\|\bar{Q})$$

As we can see, the upper bound $\sqrt{2n+1} - 1$ in (49) is quite tight and $\chi^2(P\|\bar{Q})$ is on the order of \sqrt{n} .

B.3 Proof of Corollary 17

Fix $0 < \delta \leq A^{-1/2}$. Partition the entire parameter space Θ into small hypercubes each with side length δ . For each such hypercube S and let π_S denote the probability measure w conditioned to be in S i.e., $\pi_S(C) := w(C)/w(S)$ for measurable set $C \subseteq S$.

For every decision rule $\mathfrak{d}(X)$, clearly

$$\int_{\Theta} \mathbb{E}_{\Theta} L(\theta, \mathfrak{d}(X)) w(d\theta) = \sum_S w(S) \int_S \mathbb{E}_{\Theta} L(\theta, \mathfrak{d}(X)) d\pi_S(\theta)$$

where the sum above is over all hypercubes S in the partition. This implies therefore that

$$R_{\text{Bayes}}(w, L; \Theta) \geq \sum_S w(S) R_{\text{Bayes}}(\pi_S, L; S).$$

The proof will therefore be completed if we show that

$$R_{\text{Bayes}}(\pi_S, L; S) \geq \frac{1}{2} e^{-2pA\delta^2} \delta^{2p} V^{-p/d} \int_S \left(\frac{1}{r_\delta(\bar{\theta})} \right)^{p/d} \pi_S(d\theta) \quad (86)$$

for every fixed hypercube S . So let us fix S and, for notational simplicity, let $\pi := \pi_S$. We will use (39) to prove a lower bound on $R_{\text{Bayes}}(\pi_S, L; S)$. Note first that

$$\begin{aligned} \inf_Q \int_S D(P_\theta\|Q)\pi(d\theta) &\leq \int_S \int_S D(P_\theta\|P_{\theta'})\pi(d\theta)\pi(d\theta') \\ &\leq A \max_{\theta \in S, \theta' \in S} \|\theta - \theta'\|_2^2 \leq Ad\delta^2 =: I_f^{\text{up}}. \end{aligned} \quad (87)$$

Also, letting f_w^{\max} and f_w^{\min} be the maximum and minimum values of f_w in S , we have

$$\sup_{a \in S} \pi(B_L(a, L)) \leq \frac{f_w^{\max}}{w(S)} \text{Vol}(B_L(a, L)) \leq \frac{f_w^{\max} V_\ell^d / p}{f_w^{\min} \delta^d}.$$

Let $\tilde{\theta}$ be an arbitrary point in the set S . Since S has diameter $\sqrt{d}\delta$, the set $\{\theta : \|\theta - \tilde{\theta}\|_2 \leq \sqrt{d}\delta\}$ contains S . We obtain from the definition of $r_\delta(\theta)$ that $f_w^{\max} / f_w^{\min} \leq r_\delta(\tilde{\theta})$ so that

$$\sup_{a \in S} \pi(B_L(a, L)) \leq r_\delta(\tilde{\theta}) V \delta^{-d/p}.$$

Thus, by (87), the choice

$$t = e^{-2pA\delta^2} \delta^{2p} \left(\frac{1}{8V r_\delta(\tilde{\theta})} \right)^{p/d},$$

leads to $\sup_{a \in S} \pi(B_L(a, L)) < \frac{1}{4} e^{-2I_f^{\text{up}}}$. Employing (39), we deduce

$$R_{\text{Bayes}}(\pi, L; S) \geq \frac{1}{2} e^{-2pA\delta^2} \delta^{2p} \left(\frac{1}{8V r_\delta(\tilde{\theta})} \right)^{p/d} \geq \frac{1}{2} e^{-2p} \delta^{2p} \left(\frac{1}{8V r_\delta(\tilde{\theta})} \right)^{p/d}$$

where we used the fact that $\delta^2 \leq 1/A$. Because $\tilde{\theta} \in S$ is arbitrary, we can write

$$\begin{aligned} R_{\text{Bayes}}(\pi, L; S) &\geq \frac{1}{2} e^{-2p} \delta^{2p} (8V)^{-p/d} \sup_{\tilde{\theta} \in S} \left(\frac{1}{r_\delta(\tilde{\theta})} \right)^{p/d} \\ &\geq \frac{1}{2} e^{-2p} \delta^{2p} (8V)^{-p/d} \int_S \left(\frac{1}{r_\delta(\theta)} \right)^{p/d} \pi(d\theta). \end{aligned}$$

This proves (86).

Appendix C. More Examples on Bayes Risk Lower Bounds

In this section, we provide more examples on the applications of derived Bayes risk lower bound in Theorem 9 and Corollary 12. For the clarity of the presentation, in each example, we will first present the Bayes risk lower bound and then provide the proof.

C.1 Generalized Linear Model

Fix $d \geq 1$ and let $\Theta = \mathcal{A} = \mathbb{R}^d$ with $L(\theta, a) = \|\theta - a\|_2^2$ for a fixed $p > 0$. Also fix $n \geq 1$ and an $n \times d$ matrix X whose rows are written as x_1^T, \dots, x_n^T . As in the last example, λ_{\max} denotes the maximum eigenvalue of $X^T X/n$.

For $\theta \in \Theta$, let P_θ denote the joint distribution of independent random variables Y_1, \dots, Y_n where Y_i has the density

$$\exp \left[\frac{y\beta_i - b(\beta_i)}{a(\phi)} + c(y, \phi) \right] \quad \text{for } y \in \mathbb{R} \quad (88)$$

with $\beta_i = x_i^T \theta$ for $i = 1, \dots, n$. The parameter ϕ is taken to be a constant and the functions $a(\cdot), c(\cdot, \cdot)$ and $b(\cdot)$ are assumed to be known. We assume the existence of a constant $K > 0$ such that $b''(\beta) \leq K$ for all β where $b''(\cdot)$ is the second derivative of $b(\cdot)$. This assumption indeed holds for many generalized linear models (e.g., binomial, Gaussian) and we will discuss the case (i.e., Poisson) where this assumption fails at the end of this example.

Let w denote the Gaussian prior with mean zero and covariance matrix $\tau^2 I_d$. Using Corollary 17, we can prove that

$$R_{\text{Bayes}}(w, L; \Theta) \geq C \left[d \min \left(\frac{a(\phi)}{nK}, \tau^2 \right) \right]^{1/p/2} \quad (89)$$

for a constant C that depends only on p . Let us illustrate this lower bound by considering a simple case of $p = 2$. We note that the term $\frac{a(\phi)}{nK}$ is the well-known minimax risk of generalized linear model under the squared loss. The parameter τ characterizes the strength of the prior information. In fact, since $\tau^2 I$ is the variance of the Gaussian prior distribution, a small value of τ provides strong prior information that each θ_j should be concentrated around 0. When τ is large, i.e., with less prior information, the lower bound of the Bayes risk in (89) is the same as the minimax risk up to a constant factor. On the other hand, when τ is small, i.e., with strong prior information, the lower bound of the Bayes risk becomes $d\tau^2$, which is smaller than the minimax risk.

The proof of (89) will involve Corollary 17 for which we need to determine A, V and $r_\delta(\theta)$. As before, it is easy to check that $V = \text{Vol}(B)$. To determine A , fix a pair θ_1, θ_2 and, letting $\beta_i^{(j)} = x_i^T \theta_j$ for $j = 1, 2$ and $i = 1, \dots, n$, observe that

$$D(P_{\theta_1} \| P_{\theta_2}) = \frac{1}{a(\phi)} \sum_{i=1}^n \left(b(\beta_i^{(1)}) (\beta_i^{(1)} - \beta_i^{(2)}) - (b(\beta_i^{(1)}) - b(\beta_i^{(2)})) \right)$$

By the second order Taylor expansion of $b(\beta_i^{(2)})$ at the point $\beta_i^{(1)}$, we obtain

$$D(P_{\theta_1} \| P_{\theta_2}) = \frac{1}{a(\phi)} \sum_{i=1}^n \frac{b''(\tilde{\beta}_i)}{2} (\beta_i^{(1)} - \beta_i^{(2)})^2$$

where $\tilde{\beta}_i$ lies between $\min(\beta_i^{(1)}, \beta_i^{(2)})$ and $\max(\beta_i^{(1)}, \beta_i^{(2)})$. Now because of our assumption that $b''(\cdot)$ is bounded from above by K , we get

$$\begin{aligned} D(P_{\theta_1} \| P_{\theta_2}) &\leq \frac{K}{2a(\phi)} \|\beta^{(1)} - \beta^{(2)}\|_2^2 = \frac{K}{2a(\phi)} (\theta_1 - \theta_2)^T X^T X (\theta_1 - \theta_2) \\ &\leq \frac{nK\lambda_{\max}}{2a(\phi)} \|\theta_1 - \theta_2\|_2^2. \end{aligned}$$

We can thus take $A = nK\lambda_{\max}/(2a(\phi))$ in Corollary 17. Next we control $r_\delta(\theta)$. For given θ and δ ,

$$r_\delta(\theta) = \sup \left\{ \exp \left(-\frac{1}{2\tau^2} (\|\theta_1\|_2^2 - \|\theta_2\|_2^2) \right) : \|\theta_1 - \theta_2\|_2 \leq \sqrt{d}\delta \right\}.$$

For θ_1, θ_2 with $\|\theta_1 - \theta_2\|_2 \leq \sqrt{d}\delta$, $i = 1, 2$, we have

$$\begin{aligned} \|\theta_1\|_2^2 - \|\theta_2\|_2^2 &= \|\theta_1 - \theta_2\|_2^2 + 2\theta_1^T (\theta_1 - \theta_2) - \|\theta_2 - \theta_1\|_2^2 - 2\theta_2^T (\theta_2 - \theta_1) \\ &\leq \|\theta_1 - \theta_2\|_2^2 - \|\theta_2 - \theta_1\|_2^2 + 2\|\theta_1\|_2 (\|\theta_1 - \theta_2\|_2 + \|\theta_2 - \theta_1\|_2) \\ &\leq d\delta^2 + 4\sqrt{d}\delta \|\theta\|_2. \end{aligned}$$

As a result $r_\delta(\theta)^{-p/d} \geq \exp(-p\delta^2/(2\tau^2)) \exp(-2p\delta \|\theta\|_2 / (\tau^2 \sqrt{d}))$ and hence

$$\begin{aligned} \int_{\Theta} \left(\frac{1}{r_\delta(\theta)} \right)^{p/d} w(d\theta) &\geq \exp \left(-\frac{p\delta^2}{2\tau^2} \right) \int_{\Theta} \exp \left(-\frac{2p\delta \|\theta\|_2}{\tau \sqrt{d}} \right) w(d\theta) \\ &\geq \exp \left(-\frac{p\delta^2}{2\tau^2} - \frac{4p\delta}{\tau} \right) \int_{\Theta} \mathbb{I} \{ \|\theta\|_2 < 2\tau\sqrt{d} \} w(d\theta). \end{aligned}$$

By Chebyshev's inequality, we have

$$\int_{\Theta} \mathbb{I} \{ \|\theta\|_2 \geq 2\tau\sqrt{d} \} w(d\theta) \leq \frac{1}{4\tau^2 d} \int_{\Theta} \|\theta\|_2^2 w(d\theta) = \frac{1}{4} \quad (90)$$

Consequently,

$$\int_{\Theta} \left(\frac{1}{r_\delta(\theta)} \right)^{p/d} w(d\theta) \geq \frac{3}{4} \exp \left(-\left(\frac{p\delta^2}{2\tau^2} + \frac{4p\delta}{\tau} \right) \right). \quad (91)$$

Corollary 17 therefore gives

$$R_{\text{Bayes}}(w, L; \Theta) \geq \frac{3}{8} e^{-2p} (8V)^{-p/d} \delta^p \exp \left(-\frac{p\delta^2}{2\tau^2} - \frac{4p\delta}{\tau} \right) \quad \text{whenever } \delta^2 \leq 1/4.$$

We make the choice

$$\delta^2 := \min(1/4, \tau^2) = \min \left(\frac{2a(\phi)}{nK\lambda_{\max}}, \tau^2 \right)$$

which implies that the exponential term in the right hand side of (91) is bounded from below by $\exp(-9p/2)$. We thus have

$$R_{\text{Bayes}}(w, L; \Theta) \geq \frac{3}{8} e^{-13p/2} (8V)^{-p/d} \left[\min \left(\frac{2a(\phi)}{nK\lambda_{\max}}, \tau^2 \right) \right]^{1/p/2}.$$

The inequality (89) now follows because $V^{1/d} \geq d^{-1/2}$.

The assumption that $b''(\beta) \leq K$ which was used for the proof of (89) holds under some widely used densities of Y_i in (88). For Gaussian distribution in (88), we have $b(\beta) = \frac{\beta^2}{2}$ so that $b''(\beta) = 1$ for $\beta \in \mathbb{R}$. For binomial distribution, $b(\beta) = \log(1 + \exp(\beta))$ and $b''(\beta) = \frac{\exp(\beta)}{(1 + \exp(\beta))^2} \leq \frac{1}{4}$ for all $\beta \in \mathbb{R}$. However, for Poisson distribution, $b(\beta) = \exp(\beta)$ and thus $b''(\beta) = \exp(\beta)$ is unbounded on \mathbb{R} . To address this issue, we restrict the prior to the subset $\Theta = \{\theta \in \Theta : \|\theta\|_2 \leq 2\tau\sqrt{d}\}$ and define the re-scaled prior distribution π on $\tilde{\Theta}$ as $\pi(S) = w(S)/w(\tilde{\Theta})$ for any measurable set $S \subseteq \tilde{\Theta}$. Let $B = \max_{i=1, \dots, n} \|x_i\|_2$. For any $\beta = x_i^T \theta$ for some $i = 1, \dots, n$ and $\theta \in \tilde{\Theta}$, we have $b''(\beta) \leq \exp(2\tau\sqrt{d}B) := K$. We note that such a restriction of the parameter space will not affect the order of the Bayes risk lower bound. In particular, since now $b''(\beta) \leq K$ when $\theta \in \tilde{\Theta}$, applying the same argument, we obtain the lower bound on $R_{\text{Bayes}}(\pi, L; \tilde{\Theta})$. By (90), we have $w(\tilde{\Theta}) \geq 3/4$ and the lower bound on $R_{\text{Bayes}}(w, L; \Theta)$ can be easily established by noticing that $R_{\text{Bayes}}(w, L; \Theta) \geq w(\tilde{\Theta}) R_{\text{Bayes}}(\pi, L; \tilde{\Theta}) \geq \frac{3}{4} R_{\text{Bayes}}(\pi, L; \tilde{\Theta})$.

C.2 Spiked Covariance Model

Fix $\Theta = \mathcal{A} = B$ where B is the unit Euclidean closed ball of radius one and let $L_t(\theta, a) := \|\theta - a\|_F^2$ for a fixed $p > 0$. Also fix $n \geq d/2$. For $\theta \in \Theta$, let P_θ denote the joint distribution of independent and identically distributed observations X_1, \dots, X_n satisfying the Gaussian distribution with zero mean and covariance matrix $\Sigma_\theta := I_d + \theta\theta^T$. This is the problem of estimating the principal component for a rank-one spiked covariance model. Let w denote the uniform distribution on B . We shall prove that

$$R_{\text{Bayes}}(w, L; \Theta) \geq C \left[\min \left(\frac{1}{2}, \frac{d}{n} \right) \right]^{p/2} \quad (92)$$

where C only depends on p .

The proof is based on the application of (32) with $f(x) = x^2 - 1$, i.e., on inequality (40). For this, we need to bound the term $\sup_{a \in \mathcal{A}} w(B_t(a, L))$ and the f -informativity corresponding to the chi-squared divergence. It is easy to see that $\sup_{a \in \mathcal{A}} w(B_t(a, L)) \leq t^{d/p}$.

For the f -informativity, we will use the bound (51) with $\alpha = 2$ which requires bounding $M_{\chi^2}(\epsilon, \Theta)$. According to (Guntuboyina, 2011a, Theorem 4.6.1), for two Gaussian distributions with mean zero and covariance matrices Σ_1 and Σ_2 such that $2\Sigma_1^{-1} - \Sigma_2^{-1}$ is positive definite and $\|\Sigma_1 - \Sigma_2\|_F^2 \leq \frac{1}{2}\lambda_{\min}(\Sigma_2)$, we have

$$\chi^2(N_d(0, \Sigma_1) \| N_d(0, \Sigma_2)) \leq \exp \left(\frac{\|\Sigma_1 - \Sigma_2\|_F^2}{\lambda_{\min}(\Sigma_2)^2} \right) - 1. \quad (93)$$

Here $\|\cdot\|_F$ denotes the Frobenius norm defined as $\|A\|_F^2 := \sum_{i,j} a_{ij}^2$ where $A = (a_{ij})$ and λ_{\min} denotes the smallest eigenvalue.

Using this result, we get that for $\theta_1, \theta_2 \in \Theta$ (note that $\lambda_{\min}(\Sigma_\theta) = 1$ for all θ),

$$\chi^2(P_{\theta_1} \| P_{\theta_2}) \leq \exp(n \|\Sigma_{\theta_1} - \Sigma_{\theta_2}\|_F^2) - 1, \quad (94)$$

provided

$$2\Sigma_{\theta_1}^{-1} - \Sigma_{\theta_2}^{-1} \text{ is positive definite and } \|\Sigma_{\theta_1} - \Sigma_{\theta_2}\|_F^2 \leq 1/2. \quad (95)$$

In the sequel, whenever we employ (94), the conditions (95) hold. But, for ease of presentation, instead of verifying (95) for every application of (94), we will simply assume (94) and verify the necessary conditions at the end of the proof. Assuming (94), we see that $\chi^2(P_{\theta_1} \| P_{\theta_2}) \leq \epsilon^2$ provided $\|\Sigma_{\theta_1} - \Sigma_{\theta_2}\|_F^2 \leq \log(1 + \epsilon^2)/n$. Now for $\theta_1, \theta_2 \in \Theta$

$$\begin{aligned} \|\Sigma_{\theta_1} - \Sigma_{\theta_2}\|_F^2 &= \|\theta_1\theta_1^T - \theta_2\theta_2^T\|_F^2 = \|\theta_1\theta_1^T - \theta_1\theta_2^T + \theta_1\theta_2^T - \theta_2\theta_2^T\|_F^2 \\ &\leq 2(\|\theta_1\|_2^2 + \|\theta_2\|_2^2) \|\theta_1 - \theta_2\|_2^2 \leq 4\|\theta_1 - \theta_2\|_2^2. \end{aligned}$$

It follows therefore that the ϵ^2 -covering number in the chi-squared divergence can be bounded from above by the $\sqrt{\log(1 + \epsilon^2)}/(2\sqrt{n})$ -covering number of B under the usual Euclidean norm. Consequently

$$M_{\chi^2}(\epsilon, \Theta) \leq \left(\frac{36n}{\log(1 + \epsilon^2)} \right)^{d/2} \text{ provided } \log(1 + \epsilon^2) \leq 4n.$$

We now set ϵ to satisfy $\log(1 + \epsilon^2) = \min(n/2, d)$ so that Corollary 16 gives

$$\begin{aligned} I_{\chi^2}(w, \mathcal{P}) &\leq M_{\chi^2}(\epsilon)(1 + \epsilon^2) - 1 \\ &\leq \exp \left(\min \left(\frac{n}{2}, d \right) \right) \left[36 \max \left(2, \frac{n}{d} \right) \right]^{d/2} - 1 =: I_f^{\text{up}}. \end{aligned}$$

It follows that $\sup_{a \in \mathcal{A}} w(B_t(a, L)) < \frac{1}{4}(1 + I_f^{\text{up}})^{-1}$ provided $t = (4(1 + I_f^{\text{up}}))^{-p/d}$. Inequality (40) then proves

$$R_{\text{Bayes}}(w, L; \Theta) \geq \frac{1}{2} \left(4(1 + I_f^{\text{up}}) \right)^{-p/d} \geq \frac{1}{2} (24\epsilon)^{-p} \left[\min \left(\frac{1}{2}, \frac{d}{n} \right) \right]^{p/2}$$

which implies (92).

It remains to justify the conditions (95) when we used (94). It should be clear that for this, we only need to verify (95) when

$$\|\Sigma_{\theta_1} - \Sigma_{\theta_2}\|_F^2 \leq \frac{\log(1 + \epsilon^2)}{n} = \min \left(\frac{1}{2}, \frac{d}{n} \right). \quad (96)$$

We only need to check that $2\Sigma_{\theta_1}^{-1} - \Sigma_{\theta_2}^{-1}$ is positive definite under the above condition. For this, observe that by Weyl's inequality,

$$\lambda_{\min} \left(2\Sigma_{\theta_1}^{-1} - \Sigma_{\theta_2}^{-1} \right) \geq \lambda_{\min} \left(2\Sigma_{\theta_1}^{-1} \right) - \lambda_{\max} \left(\Sigma_{\theta_2}^{-1} \right) = \frac{2}{1 + \|\theta_1\|_2^2} - 1 \geq 0.$$

This implies that $2\Sigma_{\theta_1}^{-1} - \Sigma_{\theta_2}^{-1}$ is positive semi-definite and $\|\theta_1\|_2 = 1$ is a necessary condition for $\lambda_{\min} \left(2\Sigma_{\theta_1}^{-1} - \Sigma_{\theta_2}^{-1} \right) = 0$. Under the condition that $\|\theta_1\|_2 = 1$, by Sherman-Morrison formula,

$$2\Sigma_{\theta_2}^{-1} - \Sigma_{\theta_1}^{-1} = I_d - \theta_1\theta_1^T + \frac{\theta_2\theta_2^T}{1 + \theta_2^T\theta_2}.$$

It is then easy to check that $\lambda_{\min} \left(2\Sigma_{\theta_1}^{-1} - \Sigma_{\theta_2}^{-1} \right) = 0$ only if θ_2 is orthogonal to θ_1 . However, when $\|\theta_1\|_2 = 1$ and θ_2 is orthogonal to θ_1 , $\|\Sigma_{\theta_1} - \Sigma_{\theta_2}\|_F^2 = \|\theta_1\|_2^2 + \|\theta_2\|_2^2 > 1$, which contradicts (96). Therefore $2\Sigma_{\theta_1}^{-1} - \Sigma_{\theta_2}^{-1}$ is positive definite and this completes the proof of (92).

C.3 Gaussian Model with General Loss

In this example, we consider Gaussian location model with continuous prior with a bounded Lebesgue density and general loss functions. Here, we do not specify the form of the prior and loss. We only present this example to illustrate applications of Theorem 9 and Corollary 12. Our main bound is inequality (97). This bound however might be suboptimal for specific priors w because we do not use knowledge about the specific form of w . However, when the specific form of w is available, the argument can often be easily modified to improve inequality (97). We provide examples of this at the end of this subsection.

C.3.1 GAUSSIAN MODEL WITH SQUARED LOSS

Fix $d \geq 1$. Suppose $\Theta = \mathcal{A} = \mathbb{R}^d$ and let $L(\theta; a) := \|\theta - a\|_2^2$ where $\|\cdot\|_2$ is the usual Euclidean norm on \mathbb{R}^d . For each $\theta \in \mathbb{R}^d$, let P_θ denote the Gaussian distribution with mean θ and covariance matrix $\sigma^2 I_d$ ($\sigma^2 > 0$ is a constant). For every prior w on \mathbb{R}^d with a Lebesgue density bounded by $W > 0$, we have

$$R_{\text{Bayes}}(w; L; \Theta) \gtrsim \frac{d\sigma^4 W^{-2/d}}{(\sigma^2 + V)^2} \quad (97)$$

where

$$V := \min_{s \in \mathbb{R}^d} \int_{\Theta} \frac{1}{d} \sum_{i=1}^d (\theta_i - s_i)^2 w(d\theta). \quad (98)$$

To prove (97), we shall apply (32) with $f(x) = x \log x$, i.e., we apply (39). The resulting f -informativity (a.k.a mutual information) can be bounded in the following way. Because $I(w; \mathcal{P}) \leq \int D(P_\theta \| Q) w(d\theta)$ for every Q . In particular, we take Q to be the Gaussian distribution with mean t and covariance matrix $(\sigma^2 + V)I_d$, where $t = \operatorname{argmin}_{s \in \mathbb{R}^d} \int_{\Theta} \frac{1}{d} \sum_{i=1}^d (\theta_i - s_i)^2 w(d\theta)$, i.e., $t_i = \int_{\Theta} \theta_i w(d\theta)$ is ‘‘center’’ of the prior. Then, we obtain

$$I(w; \mathcal{P}) \leq \int_{\Theta} D(N(\theta, \sigma^2 I_d) \| N(t, (\sigma^2 + V)I_d)) w(d\theta).$$

Using the standard formula for the KL divergence between two Gaussians, we deduce that

$$I(w; \mathcal{P}) \leq \frac{1}{2} \int_{\Theta} \left[\frac{\sum_{i=1}^d ((\theta_i - t_i)^2 - V)}{\sigma^2 + V} + d \log \frac{\sigma^2 + V}{\sigma^2} \right] w(d\theta)$$

which by (98) implies that

$$I(w; \mathcal{P}) \leq \frac{d}{2} \log \frac{\sigma^2 + V}{\sigma^2}. \quad (99)$$

Let J_f^w denote the right hand side above. To apply (39), we also need an upper bound on $\sup_{a \in \mathcal{A}} w(B_1(a, L))$. Because of the assumption that the Lebesgue density of w is bounded from above by W , we get

$$\sup_{a \in \mathcal{A}} w(B_1(a, L)) \leq W r^{d/2} \operatorname{Vol}(B) \quad (100)$$

where B is the Euclidean ball with unit radius. Thus the choice

$$t = cW^{-2/d} \operatorname{Vol}(B)^{-2/d} \frac{\sigma^4}{(\sigma^2 + V)^2},$$

for a small enough universal positive constant c , ensures $\sup_{a \in \mathcal{A}} w\{B_1(a)\} < \frac{1}{4} e^{-2J_f^w}$ (recall that J_f^w is the right hand side of (99)). Consequently, inequality (39) implies that $R_{\text{Bayes}} \geq t/2$. The proof of (97) is now completed using the standard fact: $\operatorname{Vol}(B)^{1/d} \asymp d^{-1/2}$.

However, since the form of the prior w is unspecified in this example, the simple upper bound on $\sup_{a \in \mathcal{A}} w(B_1(a, L))$ in (100) could be loose. But this can be easily fixed when

the concrete form of the prior is available. For example, for a spiked model with a large W (see an example of mixture prior in Remark 11 in the main text), the lower bound in (97) could be sub-optimal but can be easily tightened using the proposed chaining technique in Remark 11 in the main text. For another example, let w be the uniform prior on the hyper-rectangle $H = [-\epsilon, \epsilon] \times [-1, 1]^{d-1}$ for some very small ϵ . Here inequality (100) is equivalent to

$$\sup_{a \in \mathcal{A}} w(B_1(a, L)) \leq W r^{d/2} \operatorname{Vol}(B).$$

When $\epsilon \rightarrow 0$, we have $W \rightarrow \infty$ so that the upper bound is fairly loose. However, since H is the support of w , we can also use the following upper bound:

$$\sup_{a \in \mathcal{A}} w(B_1(a, L)) \leq W r^{d/2} \operatorname{Vol}(B \cap H).$$

When $\epsilon \rightarrow 0$, we have $W \rightarrow \infty$ but $\operatorname{Vol}(B \cap H) \rightarrow 0$. In particular, the product limit $\lim_{\epsilon \rightarrow 0} W \operatorname{Vol}(B \cap H) \rightarrow 0$ is finite. It converges to the maximum value of $w(B_1(a, L))$ where w is restricted in a $(d-1)$ -dimensional subspace of \mathbb{R}^d . Once we replace inequality (100) by the above upper bound, the associated Bayes risk lower bound will be tight.

C.3.2 GAUSSIAN MODEL WITH GENERAL LOSS

Consider the same setup as in the previous example but now allow the loss function to be $L(\theta, a) = \|\theta - a\|^2$ for an arbitrary norm $\|\cdot\|$ (not necessarily the Euclidean norm) on \mathbb{R}^d . In this case, we obtain the following Bayes risk lower bound:

$$R_{\text{Bayes}}(w; L; \Theta) \gtrsim \frac{\sigma^4 W^{-2/d}}{(\sigma^2 + V)^2} \frac{d^2}{(\mathbb{E}\|Z\|_*)^2}. \quad (101)$$

where Z is a standard Gaussian vector and $\|\cdot\|_*$ is the dual norm corresponding to $\|\cdot\|$ defined by $\|x\|_* := \sup\{x \cdot y : \|y\| \leq 1\}$. The quantities W and V are as defined in the previous example.

The proof of (101) is largely similar to that of (97). We use (39) along with (99) for controlling $I(w; \mathcal{P})$. To control $\sup_{a \in \mathcal{A}} w(B_1(a, L))$, we again use the fact that the Lebesgue density of w is bounded from above by W to obtain

$$\sup_{a \in \mathcal{A}} w(B_1(a, L)) \leq W \operatorname{Vol} \left\{ \theta \in \mathbb{R}^d : \|\theta\| < \sqrt{t} \right\}. \quad (102)$$

To deal with the volume term above, we use Urysohn’s inequality to obtain an upper bound in terms of the volume of the unit Euclidean unit ball B . The original reference for Urysohn’s inequality is Urysohn (1924) but it has been recently used in a statistical context by Ma and Wu (2015). Urysohn’s inequality gives

$$\left(\frac{\operatorname{Vol} \{ \theta \in \mathbb{R}^d : \|\theta\| < \sqrt{t} \}}{\operatorname{Vol}(B)} \right)^{\frac{1}{d}} \leq \frac{\sqrt{t}}{\sqrt{d}} \mathbb{E}\|Z\|_* \quad \text{with } Z \sim N(0, I_d). \quad (103)$$

Inequalities (102) and (103) together give

$$\sup_{a \in \mathcal{A}} w(B_1(a, L)) \leq W r^{d/2} \operatorname{Vol}(B) \left(\frac{\mathbb{E}\|Z\|_*}{\sqrt{d}} \right)^d.$$

The choice

$$t = c \text{Vol}(B)^{-2/d} \frac{W^{-2/d} \sigma^4}{(\sigma^2 + V)^2 (\mathbb{E}\|Z\|_*)^2} d$$

for a small enough universal positive constant c ensures $\sup_{a \in A} w\{B_t(a)\} < \frac{1}{4} e^{-2I_r}$ (I_r^{up} is the right hand side of (99)). The proof of (101) is then completed by noting that $\text{Vol}(B)^{1/d} \asymp d^{-1/2}$.

Appendix D. Proof of Lemma 19 in Section 6

Consider the i -th instance $x_i \in \mathbb{R}^d$ sampled from $N(\theta_{z_i}; I_{d \times d})$, where z_i is the membership. Note that for any $j \in [k] \setminus \{z_i\}$, the distance between θ_{z_i} and θ_j is lower bounded by D . We have

$$\|x_i - \theta_j\|_2^2 - \|x_i - \theta_{z_i}\|_2^2 = \|\theta_j - \theta_{z_i}\|_2^2 - 2\langle \theta_j - \theta_{z_i}, x_i - \theta_{z_i} \rangle. \quad (104)$$

The random variable $\langle \theta_j - \theta_{z_i}, x_i - \theta_{z_i} \rangle$ satisfies distribution $N(0; \|\theta_j - \theta_{z_i}\|_2^2)$. Let Φ be the CDF of the standard normal distribution. Then with probability $\Phi(\frac{\|\theta_j - \theta_{z_i}\|_2 - 1}{2})$, we have

$$\langle \theta_j - \theta_{z_i}, x_i - \theta_{z_i} \rangle \leq \|\theta_j - \theta_{z_i}\|_2 \cdot \frac{\|\theta_j - \theta_{z_i}\|_2 - 1}{2}. \quad (105)$$

Combining (104) and (105), we have

$$\|x_i - \theta_j\|_2^2 - \|x_i - \theta_{z_i}\|_2^2 \geq \|\theta_j - \theta_{z_i}\|_2. \quad (106)$$

On the other hand, the triangular inequality implies

$$\|x_i - \theta_j\|_2 + \|x_i - \theta_{z_i}\|_2 \leq 2\|x_i - \theta_{z_i}\|_2 + \|\theta_j - \theta_{z_i}\|_2. \quad (107)$$

The random variable $\|x_i - \theta_{z_i}\|_2^2$ satisfies a chi-square distribution with d degrees of freedom. It is upper bounded by βd with probability at least $1 - \exp(-\frac{d}{2}(1 - \beta + \log \beta))$ for any $\beta > 1$ (Dasgupta and Gupta, 2003). Putting (106) and (107) together, we have

$$\|x_i - \theta_j\|_2 - \|x_i - \theta_{z_i}\|_2 = \frac{\|x_i - \theta_j\|_2^2 - \|x_i - \theta_{z_i}\|_2^2}{\|x_i - \theta_j\|_2 + \|x_i - \theta_{z_i}\|_2} \geq \frac{\|\theta_j - \theta_{z_i}\|_2}{2\|\theta_j - \theta_{z_i}\|_2 + \sqrt{\beta d}} \geq \frac{3}{6 + \sqrt{\beta d}}$$

with probability at least $\Phi(\frac{D-1}{2}) - \exp(-\frac{d}{2}(1 - \beta + \log \beta))$. By choosing $D = c\sqrt{\log(nk/\delta)}$ and $\beta = c\log(nk/\delta)/d$ for a sufficiently large constant c , this probability is lower bounded by $1 - \delta/(nk)$. Applying union bound, the inequality holds for any (i, j) pair with probability at least $1 - \delta$.

Appendix E. Proof of Theorem 23 in Section 7

We start with a simplified case where the random index set K is given to the estimator. Knowing this information makes the problem easier, and makes the Bayes risk lower. In addition, it reduces the d -dimensional regression problem to a k -dimensional problem where a closed-form of the Bayes risk can be derived, which establishes the following lower bound:

Claim 1 For any $\tau > 0$, the Bayes risk is lower bounded by:

$$R_{\text{Bayes}}(w, L_{\text{est}}; \Theta) \geq \frac{1}{1 + \kappa_n^2 \tau^2 n / \sigma^2} \cdot k \tau^2, \quad \text{and} \quad R_{\text{Bayes}}(w, L_{\text{pre}}; \Theta) \geq \frac{1}{1 + \kappa_n^2 \tau^2 n / \sigma^2} \cdot \kappa_n^2 k \tau^2.$$

See Section E.1 for the proof.

For the rest of this proof, we establish stronger lower bounds using the fact that the index set K is unknown. It is easy to verify that for any random variable X sampled from $N(0, 1)$, the probability of $|X| \geq 1/2$ is greater than $1/2$. Consider a subset of the parameter space Θ :

$$\bar{\Theta} := \left\{ \theta \in \Theta : \|\theta\|_2^2 \leq 2k\tau^2 \text{ and } \sum_{i=1}^d \mathbb{1}[\theta_i] \geq \tau/2 \geq k/2 \right\}. \quad (108)$$

For a random vector θ sampled from the prior distribution w , the quantity $\|\theta\|_2^2/\tau^2$ satisfies a chi-square distribution with k degrees of freedom. For any $k \geq 1$, the event $\|\theta\|_2^2 \leq 2k\tau^2$ happens with probability at least 0.84. Given an index set K , for any $i \in K$ the random variable $\mathbb{1}[\theta_i] \geq \tau/2$ satisfies the Bernoulli distribution with parameter greater than $1/2$, so that the event $\sum_{i=1}^d \mathbb{1}[\theta_i] \geq \tau/2$ happens with probability at least $1/2$. Combining these two lower bounds and applying union bound, we obtain $w(\bar{\Theta}) \geq 1/2 - (1 - 0.84) > 1/4$. As a consequence, if we define a distribution \bar{w} over the subset $\bar{\Theta}$ by $\bar{w}(A) := w(A \cap \bar{\Theta})/w(\bar{\Theta})$, then Remark 11 implies that

$$R_{\text{Bayes}}(w, L; \Theta) \geq w(\bar{\Theta}) \cdot R_{\text{Bayes}}(\bar{w}, L; \bar{\Theta}) \geq \frac{1}{4} R_{\text{Bayes}}(\bar{w}, L; \bar{\Theta}). \quad (109)$$

Hence it suffices to focus on the Bayes risk for the marginal prior \bar{w} .

Let the action space $\mathcal{A} := \mathbb{R}^d$ and let the loss function be either the estimation error L_{est} or the prediction error L_{pre} . In order to lower bound the Bayes risk, it suffices to bound the chi-square informativity $I_{\chi^2}(\bar{w}, \mathcal{P})$ and the quantity $\sup_{a \in \mathcal{A}} \bar{w}(B_t(a, L))$, then applying Corollary 12. We begin with an upper bound on the chi-square informativity.

Claim 2 For any $\tau > 0$, the chi-square informativity is bounded by:

$$I_{\chi^2}(\bar{w}, \mathcal{P}) + 1 \leq \exp(2\kappa_n^2 \tau^2 k n / \sigma^2) \quad (110)$$

See Section E.2 for the proof.

Next, we upper bound the quantity $\sup_{a \in \mathcal{A}} \bar{w}(B_t(a, L))$. We begin by claiming a property of all Euclidean balls of small enough radius.

Claim 3 For any point $a \in \mathbb{R}^d$, let $B(a, r)$ be the Euclidean ball of radius r centering at a . If $r \leq \frac{1}{8}\sqrt{k}\tau$, then there is a universal constant $c > 0$ such that

$$\sup_{a \in \mathcal{A}} \bar{w}(B(a, r)) \leq \frac{c^k}{(d/k^2)^{k/4}} \left(\frac{\tau}{\sqrt{k}\tau} \right)^k.$$

See Section E.3 for the proof.

Lower bound on estimation error For the estimation error, we obtain by Claim 3 that for any $t \leq \frac{1}{64}k\tau^2$, the following upper bound holds:

$$\sup_{a \in \mathcal{A}} \bar{w}(B_t(a, L_{\text{est}})) = \sup_{a \in \mathcal{A}} \bar{w}(B(a, \sqrt{t})) \leq \frac{c^k}{(d/k^2)^{k/4}} \left(\frac{\sqrt{t}}{\sqrt{k\tau}} \right)^k. \quad (111)$$

Combining Claim 3 with inequality (111), and applying inequality (40) in Corollary 12, we obtain the lower bound:

$$R_{\text{Bayes}}(\bar{w}, L_{\text{est}}; \Theta) \geq \frac{1}{2} \sup \left\{ 0 < t \leq \frac{k\tau^2}{64} : \left(\frac{\sqrt{t}}{\sqrt{k\tau}} \right)^k \leq \frac{(d/k^2)^{k/4}}{c^k} \cdot \frac{1}{4} \exp(-2\kappa_n^2 \tau^2 kn / \sigma^2) \right\}.$$

The right-hand side is lower bounded by any scalar t satisfying:

$$t \leq \frac{1}{64} k\tau^2 \quad \text{and} \quad \frac{\sqrt{t}}{\sqrt{k\tau}} \leq \frac{(d/k^2)^{1/4}}{c} \cdot \frac{1}{4^{1/k}} \exp(-2\kappa_n^2 \tau^2 n / \sigma^2)$$

It implies that for some universal constant $c' > 0$, we have:

$$\begin{aligned} R_{\text{Bayes}}(\bar{w}, L_{\text{est}}; \Theta) &\geq c' k\tau^2 \min \left\{ 1, \exp\left(\frac{1}{2} \log(d/k^2) - 4\kappa_n^2 \tau^2 n / \sigma^2\right) \right\} \\ &= c' k\tau^2 \exp \left(\min \left\{ 0, \frac{1}{2} \log(d/k^2) - \frac{4\kappa_n^2 \tau^2 n}{\sigma^2} \right\} \right) \\ &= c' k\tau^2 \exp \left(-\frac{4\kappa_n^2 n}{\sigma^2} \left[\tau^2 - \frac{\sigma^2 \log(d/k^2)}{8\kappa_n^2 n} \right]_+ \right) \\ &\geq c' k\tau^2 \exp \left(-\frac{4\kappa_n^2 n}{\sigma^2} \left[\tau^2 - \frac{\sigma^2 \log(d/k)}{16\kappa_n^2 n} \right]_+ \right), \end{aligned} \quad (112)$$

where the last inequality uses the assumption $d > k^3$ and its implication $\log(d/k^2) > \frac{1}{2} \log(d/k)$.

Combining inequality (112) with Claim 1 yields the lower bound:

$$R_{\text{Bayes}}(w, L_{\text{est}}; \Theta) \geq c' k\tau^2 \max \left\{ \frac{1}{1 + \kappa_n^2 \tau^2 n / \sigma^2}, \exp \left(-\frac{4\kappa_n^2 n}{\sigma^2} \left[\tau^2 - \frac{\sigma^2 \log(d/k)}{16\kappa_n^2 n} \right]_+ \right) \right\},$$

which completes the proof.

Lower bound on prediction error For the prediction error, we consider an arbitrary vector $a \in \mathbb{R}^d$ and an arbitrary scalar t satisfying $\sqrt{t} \leq \frac{n}{16} \sqrt{k\tau}$. Let θ' be the vector in Θ which minimizes the term $\frac{1}{n} \|X(a - \theta')\|_2^2$. If the inequality $\frac{1}{n} \|X(a - \theta')\|_2^2 > t$ is true, then we have

$$\sup_{a \in \mathcal{A}} \bar{w}(B_t(a, L_{\text{pre}})) = 0. \quad (113)$$

Otherwise, we assume that $\frac{1}{n} \|X(a - \theta')\|_2^2 \leq t$. Then for any vector $\theta \in \Theta$ satisfying $\frac{1}{n} \|X(\theta - a)\|_2^2 \leq t$, we have the upper bound

$$\frac{1}{n} \|X(\theta - \theta')\|_2^2 \leq (n^{-1/2} \|X(\theta - a)\|_2 + n^{-1/2} \|X(a - \theta')\|_2)^2 \leq (\sqrt{t} + \sqrt{t})^2 \leq 4t.$$

It means that $B_t(a, L_{\text{pre}}) \subseteq B_{4t}(\theta', L_{\text{pre}})$. Since the vector θ' is k -sparse, the sparse eigenvalue condition implies that for any vector $\theta \in \Theta$, if $L_{\text{pre}}(\theta, \theta') \leq 4t$, then $\|\theta - \theta'\|_2 \leq \frac{2\sqrt{t}}{\kappa_\ell}$, so that $B_{4t}(\theta', L_{\text{pre}}) \subseteq B(\theta', \frac{2\sqrt{t}}{\kappa_\ell})$. Using Claim 3, we have

$$\sup_{a \in \mathcal{A}} \bar{w}(B_t(a, L_{\text{pre}})) \leq \sup_{a \in \mathcal{A}} \bar{w}(B(\theta', \frac{2\sqrt{t}}{\kappa_\ell})) \leq \sup_{a \in \mathcal{A}} \bar{w}(B(a, L_{\text{est}})) \leq \frac{c^k}{(d/k^2)^{k/4}} \left(\frac{2\sqrt{t}}{\kappa_\ell \sqrt{k\tau}} \right)^k. \quad (114)$$

Combining equation (113) and inequality (114) we obtain

$$\sup_{a \in \mathcal{A}} \bar{w}(B_t(a, L_{\text{pre}})) \leq \frac{c^k}{(d/k^2)^{k/4}} \left(\frac{2\sqrt{t}}{\kappa_\ell \sqrt{k\tau}} \right)^k \quad \text{for any } \sqrt{t} \leq \frac{\kappa_\ell}{16} \sqrt{k\tau}. \quad (115)$$

Comparing inequalities (111) and (115), we find that they differ by a factor of $(2/\kappa_\ell)^k$. Thus, following the same steps for deriving Inequality (112), we can find a universal constant $c'' > 0$ such that:

$$R_{\text{Bayes}}(\bar{w}, L_{\text{pre}}; \Theta) \geq c'' \kappa_\ell^2 k\tau^2 \exp \left(-\frac{4\kappa_n^2 n}{\sigma^2} \left[\tau^2 - \frac{\sigma^2 \log(d/k)}{16\kappa_n^2 n} \right]_+ \right) \quad (116)$$

Combining inequality (116) with Claim 1 yields:

$$R_{\text{Bayes}}(w, L_{\text{pre}}; \Theta) \geq c'' \kappa_\ell^2 k\tau^2 \max \left\{ \frac{1}{1 + \kappa_n^2 \tau^2 n / \sigma^2}, \exp \left(-\frac{4\kappa_n^2 n}{\sigma^2} \left[\tau^2 - \frac{\sigma^2 \log(d/k)}{16\kappa_n^2 n} \right]_+ \right) \right\},$$

which completes the proof.

E.1 Proof of Claim 1

The Bayes risk of the original problem is lower bounded by that of the following simplified problem: estimating θ when the index set K is known, and without loss of generality, we assume that $K = [k]$. For this case, let X' be the submatrix consisting of the first k columns of matrix X , and let θ' be the subvectors consisting of the first k coordinate of vectors θ . Given the response vector y , the posterior distribution of θ' is equal to

$$p(\theta' | y) \propto p(\theta) p(y | \theta) = N(\theta'; 0, \tau^{-2} I) N(y; X\theta', \sigma^2 I) \propto N(\theta'; \Sigma^{-1}(X')^\top y, \sigma^2 \Sigma^{-1}),$$

where $\Sigma := (X')^\top X' + \frac{\sigma^2}{\tau^2} I$ is a shorthand notation. As a consequence, the Bayes estimator $\hat{\theta}$ is given by $\hat{\theta}_K = \Sigma^{-1}(X')^\top y$ and $\hat{\theta}_{-K} = 0$. The Bayes risk on the estimation error is lower bounded by:

$$R_{\text{Bayes}}(w, L_{\text{est}}; \Theta) = \mathbb{E}[\|\Sigma^{-1}(X')^\top y - \theta'\|_2^2] = \sigma^2 \text{tr}(\Sigma^{-1}) \geq \frac{\sigma^2}{\kappa_n^2 \tau^2 n + \sigma^2} \cdot k\tau^2,$$

where the last inequality uses the sparse eigenvalue condition — it guarantees that all eigenvalues of the matrix Σ are less than or equal to $n\kappa_n^2 + \sigma^2/\tau^2$.

The Bayes estimator for minimizing the prediction error is also given by $\hat{\theta}_K = \Sigma^{-1}(X')^\top y$ and $\hat{\theta}_{-K} = 0$. Thus, the Bayes risk is lower bounded by:

$$\begin{aligned} R_{\text{Bayes}}(w, L_{\text{pre}}; \Theta) &= \frac{1}{n} \mathbb{E} \left[\|X'(\Sigma^{-1}(X')^\top y - \theta')\|_2^2 \right] = \frac{\sigma^2}{n} \text{tr}(X' \Sigma^{-1}(X')^\top) \\ &\geq \frac{\sigma^2}{k_1^2 \tau^2 n + \sigma^2} \cdot k_1^2 k \tau^2, \end{aligned}$$

where the last inequality uses the sparse eigenvalue condition — it guarantees that all eigenvalues of the matrix $X' \Sigma^{-1}(X')^\top$ are greater than or equal to $\frac{nk_1^2}{nk_1^2 + \sigma^2} \tau^2$.

E.2 Proof of Claim 2

Corollary 16 shows that the chi-square informativity can be bounded using the covering number $M_{\chi^2}(\epsilon, \Theta)$. Consider the zero vector $\theta_0 := 0$ and an arbitrary vector $\theta \in \Theta$. Their response vectors are generated from $P_{\theta_0} := N(0, \sigma^2 I)$ and $P_\theta := N(X\theta, \sigma^2 I)$, so that the chi-square divergence between P_{θ_0} and P_θ is equal to $\chi^2(P_{\theta_0}, P_\theta) = \exp(\|X\theta\|_2^2 / \sigma^2) - 1$. By the sparse eigenvalue condition and the fact that $\|\theta\|_2^2 \leq 2k\tau^2$, we have

$$\chi^2(P_{\theta_0}, P_\theta) \leq \exp(\kappa_n^2 \|\theta\|_2^2 / \sigma^2) - 1 \leq \exp(2\kappa_n^2 \tau^2 kn / \sigma^2) - 1.$$

It means that if we choose $\epsilon^2 = \exp(2\kappa_n^2 \tau^2 kn / \sigma^2) - 1$, then $M_{\chi^2}(\epsilon, \Theta) = 1$, so that the chi-square informativity is bounded by

$$I_{\chi^2}(\bar{w}, \mathcal{P}) + 1 \leq (1 + \epsilon^2) M_{\chi^2}(\epsilon, \Theta) = \exp(2\kappa_n^2 \tau^2 kn / \sigma^2). \quad (117)$$

E.3 Proof of Claim 3

Consider an arbitrary vector $a \in \mathbb{R}^d$, and let I_a be the set of indices defined by:

$$I_a = \{i \in [d] : |a_i| \geq \tau/4\}.$$

If $|I_a| > 2k$, then for any $\theta \in \bar{\Theta}$, there are at least $k+1$ coordinates such that $\theta_i = 0$ but $|a_i| \geq \tau/4$. It means that $\|a - \theta\|_2 > \frac{1}{4}\sqrt{k}\tau$. Since $r \leq \frac{1}{8}\sqrt{k}\tau$, we have $\bar{w}(B(a, r)) = 0$.

Otherwise, we assume that $|I_a| \leq 2k$. Given an index set K , let w_K and \bar{w}_K be the conditional version of the prior distribution w and \bar{w} , conditioning on the fact that the k -sparse index set is K . Recall that for any θ in the support of \bar{w}_K , there are at least $k/2$ coordinates such that $|\theta_i| \geq \tau/2$. If $|I_a \cap K| < k/4$, then there are at least $k/4$ coordinates such that $|\theta_i| \geq \tau/2$ but $|a_i| < \tau/4$. It means that $\|a - \theta\|_2 > \frac{3}{8}\sqrt{k}\tau$ for any θ in the support of \bar{w}_K , and as a consequence, we have $\bar{w}_K(B(a, r)) = 0$.

Thus, a necessary condition for $\bar{w}_K(B(a, r)) > 0$ to hold is $|I_a \cap K| \geq k/4$. Given $|I_a| \leq 2k$, the number of index set K satisfying this constraint is bounded by $\binom{2k}{k/4} \binom{d-k/4}{3k/4}$. To prove this bound, notice that every set K satisfying $|I_a \cap K| \geq k/4$ can be generated by the following two-step procedure: first, generate $k/4$ element from I_a ; second, generate the remaining $3k/4$ elements from the remaining $d - k/4$ integers of $\{1, \dots, d\}$. There are totally $\binom{2k}{k/4} \binom{d-k/4}{3k/4}$ ways of generating the set. We note that the same K can have multiple ways to generate, so that the above combinatorial number is a strict upper bound on the number of sets.

For any set K satisfying the above constraint, we have:

$$\bar{w}_K(B(a, r)) \leq 4w_K(B(a, r)) \leq 4w_K(B(0, r)), \quad (118)$$

where the last equation holds because w_K represents an isotropic normal distribution in \mathbb{R}^k , so that the maximum probability is achieved by centering at the origin. The right-hand side of inequality (118) the probability a k -dimension normal random variable $X \sim N(0, \tau^2 I_{k \times k})$ satisfying $\|X\|_2 \leq r$. As we showed in the proof of Lemma 21, this probability is bounded by $(\frac{c\tau}{\sqrt{k}\tau})^k$ for a universal constant $c > 0$. Putting pieces together, we have

$$\sup_{a \in A} \bar{w}(B(a, r)) \leq \frac{\binom{2k}{k/4} \binom{d-k/4}{3k/4}^k}{\binom{d}{k}} \cdot 4 \left(\frac{c\tau}{\sqrt{k}\tau} \right)^k.$$

By the definition of the combinatorial numbers, we have:

$$\begin{aligned} \frac{\binom{2k}{k/4} \binom{d-k/4}{3k/4}^k}{\binom{d}{k}} &= \binom{2k}{k/4} \frac{k!(d-k/4)!}{(3k/4)! d!} \leq \binom{2k}{k/4} \frac{k!}{(3k/4)!} \frac{1}{d^{k/4}} \\ &\leq \frac{(2k)^{k/2}}{d^{k/4}} = \left(\frac{d}{4k^2} \right)^{-k/4}. \end{aligned}$$

Combining the two upper bounds above completes the proof.

References

- S. M. Ali and S. D. Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society. Series B*, 28(1):131–142, 1966.
- S. Arora and R. Kannan. Learning mixtures of separated nonspherical Gaussians. *The Annals of Applied Probability*, 15(1A):69–92, 2005.
- P. Assouad. Deux remarques sur l'estimation. *Comptes Rendus de L'Academie des Sciences de Paris*, 296:1021–1024, 1983.
- K. B. Athreya and S. N. Lahiri. *Measure Theory and Probability Theory*. Springer, 2006.
- C. Banderier, R. Beier, and K. Mehlhorn. Smoothed analysis of three combinatorial problems. In *Mathematical Foundations of Computer Science 2003*, pages 198–207. Springer, 2003.
- J. O. Berger. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013.
- L. Birgé. A new lower bound for multiple hypothesis testing. *IEEE Trans. Inform. Theory*, 51(4):1611–1615, 2005.
- A. Blum and J. Dunagan. Smoothed analysis of the perceptron algorithm for linear programming. In *Proceedings of the ACM-SIAM symposium on Discrete algorithms (SODA)*, 2002.

- B. Z. Borovkov and A. U. Sakhanenko. On estimates of the expected quadratic risk. *Probl. Math. Statist.*, 1:185–195, 1980.
- G. Braun and S. Pokutta. A general Fano inequality. Preprint; available at <http://www.pokutta.com/Homepage/Publications.html>, 2014.
- L. D. Brown. An information inequality for the Bayes risk under truncated squared error loss. *Multivariate Analysis: Future Directions*, pages 85–94, 1993.
- L. D. Brown and L. Gajek. Information inequalities for the Bayes risk. *The Annals of Stat.*, 18(4):1578–1594, 1990.
- L. D. Brown and R. C. Liu. Bounds on the Bayes and minimax risk for signal parameter estimation. *IEEE Trans. Inform. Theory*, 39(4):1386–1394, 1993.
- E. J. Candès, J. K. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8):1207–1223, 2006.
- I. Castillo. Lower bounds for posterior rates with gaussian process priors. *Electronic Journal of Statistics*, 2:1281–1299, 2008.
- T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 2nd edition, 2006.
- I. Csizsár. Eine informationstheoretische ungleichung und ihre anwendung auf den beweis der endodizität von markoffschen ketten. *Publ. Math. Inst. Hungar. Acad. Sci., Series A*, 8:84–108, 1963.
- I. Csizsár. A class of measures of informativity of observation channels. *Periodica Mathematica Hungarica*, 2 (1–4):191–213, 1972.
- S. Dasgupta. Learning mixtures of gaussians. In *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, 1999.
- S. Dasgupta and A. Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.
- S. Dasgupta and L. J. Schulman. A two-round variant of EM for gaussian mixtures. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2000.
- J. C. Duchi and M. J. Wainwright. Distance-based and continuous Fano inequalities with applications to statistical estimation. Technical report, UC Berkeley, 2013.
- J. Dunning, D. A Spielman, and S. H. Teng. Smoothed analysis of condition numbers and complexity implications for linear programming. *Mathematical Programming*, 126(2):315–350, 2011.
- T. S. Ferguson. *Mathematical Statistics: A Decision Theoretic Approach*. Academic Press, 1967.
- L. Gajek and M. Kaluszcza. Lower bounds for the asymptotic Bayes risk in the scale model (with an application to the second-order minimax estimation). *The Annals of Statistics*, 22(4):1831–1839, 1994.
- D. Garcia-Garcia and R. C. Williamson. Divergences and risks for multiclass experiments. In *Proceedings of the Annual Conference on Learning Theory (COLT)*, 2012.
- R. Ge, Q. Q. Huang, and S. M. Kakade. Learning mixtures of Gaussians in high dimensions. *arXiv preprint arXiv:1503.00424*, 2015.
- R. D. Gill and B. Y. Levit. Applications of the van Trees inequality: a Bayesian Cramér-Rao bound. *Bernoulli*, 1(1-2):59–79, 03 1995.
- O. Guédon, S. Mendelson, A. Pajor, and N. Tomczak-Jaegermann. Majorizing measures and proportional subsets of bounded orthonormal systems. *Revista matemática iberoamericana*, 24(3):1075–1095, 2008.
- A. Guntuboyina. *Minimax Lower Bounds*. PhD thesis, Yale University, 2011a.
- A. Guntuboyina. Lower bounds for the minimax risk using f -divergences, and applications. *IEEE Transactions on Information Theory*, 57:2386–2399, 2011b.
- A. A. Gushchin. On Fano’s lemma and similar inequalities for the minimax risk. *Theor. Probability and Math. Statist.*, 67:29–41, 2003.
- T. S. Han and S. Verdú. Generalizing the fano inequality. *IEEE Trans. Inform. Theory*, 40:1247–1251, 1994.
- D. Haussler and M. Opper. Mutual information, metric entropy and cumulative relative entropy risk. *The Annals of Statistics*, 25(6):2451–2492, 1997.
- D. Hsu and S. M. Kakade. Learning mixtures of spherical Gaussians: moment methods and spectral decompositions. In *Proceedings of the Conference on Innovations in Theoretical Computer Science*, 2013.
- F. Liese. Phi-divergences, sufficiency, Bayes sufficiency, and deficiency. *Kybernetika*, 48(4):690–713, 2012.
- Z. Ma and Y. Wu. Volume ratio, sparsity, and minimaxity under unitarily invariant norms. *IEEE Trans. Inform. Theory*, 61(12):6939–6956, 2015.
- B. Mauthey and R. Reischuk. Smoothed analysis of binary search trees. *Theoretical Computer Science*, 378(3):292–315, 2007.
- A. Rakhlin, K. Sridharan, and A. B. Tsybakov. Empirical entropy; minimax regret and minimax risk. *arXiv preprint arXiv:1308.1147*, 2013.
- G. Raskutti, M. J. Wainwright, and B. Yu. Restricted eigenvalue properties for correlated Gaussian designs. *Journal of Machine Learning Research*, 11:2241–2259, 2010.

- G. Raskutti, M. J. Wainwright, and B. Yu. Minimax rates of estimation for high-dimensional linear regression over ℓ_q -balls. *IEEE Trans. Inform. Theory*, 57(10):6976–6994, 2011.
- G. Raskutti, M. J. Wainwright, and B. Yu. Minimax-optimal rates for sparse additive models over kernel classes via convex programming. *Journal of Machine Learning Research*, 13(1):389–427, 2012.
- M. D. Reid and R. C. Williamson. Generalised Pinsker inequalities. *arXiv preprint arXiv:0906.1244*, 2009.
- M. D. Reid and R. C. Williamson. Information, divergence and risk for binary experiments. *Journal of Machine Learning Research*, 12:731–817, 2011.
- H. Röglin and B. Vöcking. Smoothed analysis of integer programming. *Mathematical programming*, 110(1):21–56, 2007.
- M. Sato and M. Akahira. An information inequalities for the Bayes risk. *The Annals of Statistics*, 24(5):2288–2295, 1996.
- D. A. Spielman and S. H. Teng. Smoothed analysis. In *Algorithms and data structures*, pages 256–270. Springer, 2003.
- Y. Takada. Lower bounds on the Bayes risk for statistical precision problem. *Communications in Statistics - Theory and Methods*, 28:693–703, 1999.
- A. B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2010.
- P. S. Urysohn. Mean width and volume of convex bodies in n -dimensional space. *Mat. Sbornik*, 31:477–486, 1924.
- H. Van Trees. *Detection, Estimation and Modulation Theory*. Wiley, 1968.
- S. Vempala and G. Wang. A spectral algorithm for learning mixture models. *Journal of Computer and System Sciences*, 68(4):841–860, 2004.
- B. Vidakovi and A. DasGupta. Lower bounds on Bayes risk for estimating a normal variable: with applications. *The Canadian Journal of Statistics*, 23(3):269–282, 1995.
- A. Xu and M. Raginsky. A new information-theoretic lower bound for distributed function computation. In *Proceedings of IEEE International Symposium on Information Theory*, 2014.
- Y. H. Yang. Minimax nonparametric classification. I. rates of convergence. *IEEE Trans. Inform. Theory*, 45(7):2271–2284, 1999.
- Y. H. Yang and A. Barron. Information-theoretic determination of minimax rates of convergence. *The Annals of Statistics*, 27(5):1564–1599, 1999.
- B. Yu. Assouad, Fano, and Le Cam. In *Festschrift for Lucien Le Cam*, pages 423–435. Springer, 1997.
- T. Zhang. Information-theoretic upper and lower bounds for statistical estimation. *IEEE Trans. Inform. Theory*, 52(4):1307–1321, 2006.
- Y. Zhang, M. J. Wainwright, and M. I. Jordan. Lower bounds on the performance of polynomial-time algorithms for sparse linear regression. In *Proceedings of the Conference on Learning Theory*, 2014.
- Y. Zhang, M. J. Wainwright, and M. I. Jordan. Optimal prediction for sparse linear models? lower bounds for coordinate-separable M-estimators. *arXiv preprint arXiv:1503.03188*, 2015.
- Y. Zhang, X. Chen, D. Zhou, and M. I. Jordan. Spectral methods meet EM: A provably optimal algorithm for crowdsourcing. *Journal of Machine Learning Research*, 17(102):1–44, 2016.

Weak Convergence Properties of Constrained Emphatic Temporal-difference Learning with Constant and Slowly Diminishing Stepsize

Huizhen Yu

*Reinforcement Learning and Artificial Intelligence Group
Department of Computing Science, University of Alberta
Edmonton, AB, T6G 2E8, Canada*

JANEX.HZYU@GMAIL.COM

Editor: Shie Mannor

Abstract

We consider the emphatic temporal-difference (TD) algorithm, ETD(λ), for learning the value functions of stationary policies in a discounted, finite state and action Markov decision process. The ETD(λ) algorithm was recently proposed by Sutton, Mahmood, and White (2016) to solve a long-standing divergence problem of the standard TD algorithm when it is applied to off-policy training, where data from an exploratory policy are used to evaluate other policies of interest. The almost sure convergence of ETD(λ) has been proved in our recent work under general off-policy training conditions, but for a narrow range of diminishing stepsize. In this paper we present convergence results for constrained versions of ETD(λ) with constant stepsize and with diminishing stepsize from a broad range. Our results characterize the asymptotic behavior of the trajectory of iterates produced by those algorithms, and are derived by combining key properties of ETD(λ) with powerful convergence theorems from the weak convergence methods in stochastic approximation theory. For the case of constant stepsize, in addition to analyzing the behavior of the algorithms in the limit as the stepsize parameter approaches zero, we also analyze their behavior for a fixed stepsize and bound the deviations of their averaged iterates from the desired solution. These results are obtained by exploiting the weak Feller property of the Markov chains associated with the algorithms, and by using ergodic theorems for weak Feller Markov chains, in conjunction with the convergence results we get from the weak convergence methods. Besides ETD(λ), our analysis also applies to the off-policy TD(λ) algorithm, when the divergence issue is avoided by setting λ sufficiently large. It yields, for that case, new results on the asymptotic convergence properties of constrained off-policy TD(λ) with constant or slowly diminishing stepsize.

Keywords: Markov decision processes, approximate policy evaluation, reinforcement learning, temporal-difference methods, importance sampling, stochastic approximation, convergence

1. Introduction

We consider discounted finite state and action Markov decision processes (MDPs) and the problem of learning an approximate value function for a given policy from *off-policy* data, that is, from data due to a different policy. The first policy is called the *target policy* and the second the *behavior policy*. The case of *on-policy* learning, where the target and behavior policies are the same, has been well-studied and widely applied (see e.g.,

Sutton, 1988; Tsitsiklis and Van Roy, 1997; and the books Bertsekas and Tsitsiklis, 1996; Sutton and Barto, 1998). Off-policy learning provides additional flexibilities and is useful in many contexts. For example, one may want to avoid executing the target policy before estimating the potential risk for safety concerns, or one may want to learn value functions for many target policies in parallel from one exploratory behavior. These require off-policy learning. In addition, insofar as value functions (with respect to different reward/cost assignments) reflect statistical properties of future outcomes, off-policy learning can be used by an autonomous agent to build an experience-based internal model of the world in artificial intelligence applications (Sutton, 2009). Algorithms for off-policy learning are thus not only useful as model-free computational methods for solving MDPs, but can also potentially be a step toward the goal of making autonomous agents capable of learning over a long life-time, facing a sequence of diverse tasks.

In this paper we focus on a new off-policy learning algorithm proposed recently by Sutton, Mahmood, and White (2016): the emphatic temporal-difference (TD) learning algorithm, or ETD(λ). The algorithm is similar to the standard TD(λ) algorithm with linear function approximation (Sutton, 1988), but uses a novel scheme to resolve a long-standing divergence problem in TD(λ) when applied to off-policy data. Regarding the divergence problem, while TD(λ) was proved to converge for the on-policy case (Tsitsiklis and Van Roy, 1997), it was known quite early that the algorithm can diverge in other cases (Baird, 1995; Tsitsiklis and Van Roy, 1997).¹ The difficulty is intrinsic to sampling states according to an arbitrary distribution. Since then alternative algorithms without convergence issues have been sought for off-policy learning. In particular, in the off-policy LSTD(λ) algorithm (Bertsekas and Yu, 2009; Yu, 2012), which is an extension of the on-policy least-squares version of TD(λ) proposed by Bradtke and Barto (1996) and Boyan (1999), with higher computational complexity than TD(λ), the linear equation associated with TD(λ) is estimated from data and then solved.² In the gradient-TD algorithms (Sutton et al., 2008, 2009; Maei, 2011) and the proximal gradient-TD algorithms (Liu et al., 2009; Mahadevan and Liu, 2012; see also Mahadevan et al., 2014; Liu et al., 2015), the difficulty in TD(λ) is overcome by reformulating the approximate policy evaluation problem TD(λ) attempts to solve as optimization problems and then tackle them with optimization techniques. (See the surveys Geist and Scherrer, 2014 and Dann et al., 2014 for other algorithm examples.)

Compared to the algorithms just mentioned, ETD(λ) is closer to the standard TD(λ) algorithm and addresses the issue in TD(λ) more directly. It introduces a novel weighting scheme to re-weight the states when forming the eligibility traces in TD(λ), so that the weights reflect the occupation frequencies of the target policy rather than the behavior policy. An important result of this weighting scheme is that under natural conditions on the function approximation architecture, the average dynamics of ETD(λ) can be described by an affine function involving a negative definite matrix (Sutton et al., 2016; Yu, 2015a),³

1. For related discussions, see also Bertsekas and Tsitsiklis (1996); Sutton and Barto (1998); and Sutton et al. (2016).
2. An efficient algorithm for solving the estimated equations is the one given by Yao and Liu (2008) based on the line search method. It can also be applied to finding approximate solutions under additional penalty terms suggested by Pires and Szepesvári (2012).
3. Sutton et al. (2016) work with the negation of the matrix that we associate with ETD(λ) in this paper. The negative definiteness property we discuss here corresponds to the positive definiteness property discussed in their work.

which provides a desired stability property, similar to the case of convergent on-policy TD algorithms.

The almost sure convergence of ETD(λ), under general off-policy training conditions, has been shown in our recent work (Yu, 2015a) for diminishing stepsize. That result, however, requires the stepsize to diminish at the rate of $O(1/t)$, with t being the time index of the iterate sequence. This range of stepsize is too narrow for applications. In practice, algorithms tend to make progress too slowly if the stepsize becomes too small, and the environment may be non-stationary, so it is often preferred to use a much larger stepsize or constant stepsize.

The purpose of this paper is to provide an analysis of ETD(λ) for a broad range of stepsizes. Specifically, we consider constant stepsize and stepsize that can decrease at a rate much slower than $O(1/t)$. We will maintain general off-policy training conditions, without placing restrictions on the behavior policy. However, we will consider constrained versions of ETD(λ), which constrain the iterates to be in a bounded set, and a mode of convergence that is weaker than almost sure convergence. Constraining the ETD(λ) iterates is not only needed in analysis, but also a means to control the variances of the iterates, which is important in practice since off-policy learning algorithms generally have high variances. Almost sure convergence is no longer guaranteed for algorithms using large stepsizes; hence we analyze their behavior with respect to a weaker convergence mode.

We study a simple, basic version of constrained ETD(λ) and several variations of it, some of which are biased but can mitigate the variance issue better. To give an overview of our results, we shall refer to the first algorithm as the unbiased algorithm, and its biased variations as the biased variants. Two groups of results will be given to characterize the asymptotic behavior of the trajectory of iterates produced by these algorithms. The first group of results are derived by combining key properties of ETD(λ) with powerful convergence theorems from the weak convergence methods in stochastic approximation theory. The results show (roughly speaking) that:

- (i) In the case of diminishing stepsize, under mild conditions, the trajectory of iterates produced by the unbiased algorithm eventually spends nearly all its time in an arbitrarily small neighborhood of the desired solution, with an arbitrarily high probability (Theorem 4); and the trajectory produced by the biased algorithms has a similar behavior, when the algorithmic parameters are set to make the biases sufficiently small (Theorem 6). These results entail the convergence in mean to the desired solution for the unbiased algorithm (Corollary 2), and the convergence in probability to some vicinity of the desired solution for the biased variants.

- (ii) In the case of constant stepsize, imagine that we run the algorithms for all stepsizes; then conclusions similar to those in (i) hold in the limit as the stepsize parameter approaches zero (Theorems 5 and 7). In particular, a smaller stepsize parameter results in an increasingly longer segment of the trajectory to spend, with an increasing probability, nearly all its time in some neighborhood of the desired solution. The size of the neighborhood can be made arbitrarily small as the stepsize parameter approaches zero and, in the case of the biased variants, also as their biases are reduced.

The next group of results are for the constant-stepsize case and complement the results in (ii) by focusing on the asymptotic behavior of the algorithms for a fixed stepsize. Among others, they show (roughly speaking) that:

- (iii) For any given stepsize parameter, asymptotically, the expected maximal deviation of multiple consecutive averaged iterates from the desired solution can be bounded in terms of the masses that the invariant probability measures of certain associated Markov chains assign to a small neighborhood of the desired solution. Those probability masses approach one when the stepsize parameter approaches zero and, in the case of the biased variants, also when their biases are sufficiently small (Theorems 8 and 9).

- (iv) For a perturbed version of the unbiased algorithm and its biased variants, the maximal deviation of averaged iterates from the desired solution, under a given stepsize parameter, can be bounded almost surely in terms of those probability masses mentioned in (iii), for each initial condition (Theorems 10 and 11).

To derive the first group of results, we use powerful convergence theorems from the weak convergence methods in stochastic approximation theory (Kushner and Clark, 1978; Kushner and Schwartz, 1984; Kushner and Yin, 2003). This theory builds on the ordinary differential equation (ODE) based proof method, treats the trajectory of iterates as a whole, and studies its asymptotic behavior through the continuous-time processes corresponding to left-shifted and interpolated iterates. The probability distributions of these continuous-time interpolated processes are analyzed (as probability measures on a function space) by the weak convergence methods, leading to a characterization of their limiting distributions, from which asymptotic properties of the trajectory of iterates can be obtained.

Most of our efforts in the first part of our analysis are to prove that the constrained ETD(λ) algorithms satisfy the conditions required by the general convergence theorems just mentioned. We prove this by using key properties of ETD(λ) iterates, most importantly, the ergodicity and uniform integrability properties of the trace iterates, and the convergence of certain averaged processes which, intuitively speaking, describe the averaged dynamics of ETD(λ). Some of these properties were established earlier in our work (2015a) when analyzing the almost sure convergence of ETD(λ). Building upon that work, we prove the remaining properties needed in the analysis.

To derive the second group of results, we exploit the fact that in the case of constant stepsize, the iterates together with other random variables involved in the algorithms form weak Feller Markov chains, and such Markov chains have nice ergodicity properties. We use ergodic theorems for weak Feller Markov chains (Meyn, 1989; Meyn and Tweedie, 2009), together with the properties of ETD(λ) iterates and the convergence results we get from the weak convergence methods, in this second part of our analysis.

Besides ETD(λ), the analysis we give in the paper also applies to off-policy TD(λ), when the divergence issue mentioned earlier is avoided by setting λ sufficiently close to 1. The reason is that in that case the off-policy TD(λ) iterates have the same properties as the ones used in our analysis of ETD(λ) and therefore, the same conclusions hold for constrained versions of off-policy TD(λ), regarding their asymptotic convergence properties for constant or slowly diminishing stepsize (these results are new, to our knowledge). Similarly, our analysis also applies directly to the ETD(λ, β) algorithm, a variation of ETD(λ) recently proposed by Hallak et al. (2016).

Regarding practical performance of the algorithms, the biased ETD variant algorithms are much more robust than the unbiased algorithm despite the latter's superior asymptotic convergence properties. (This is not a surprise, for the biased algorithms are in fact defined

by using a well-known robustifying approach from stochastic approximation theory.) Their behavior is demonstrated by experiments in (Mahmood et al., 2015; Yu, 2016). In particular, the report (Yu, 2016) is our companion note for this paper and includes several simulation results to illustrate some of the theorems we give here regarding the behavior of multiple consecutive iterates of the biased algorithms.

The paper is organized as follows. In Section 2 we provide the background for the ETD(λ) algorithm. In Section 3 we present our convergence results on constrained ETD(λ) and several variants of it, and we give the proofs in Section 4. We conclude the paper in Section 5 with a brief discussion on direct applications of our convergence results to the off-policy TD(λ) algorithm and the ETD(λ, β) algorithm, as well as to ETD(λ) under relaxed conditions, followed by a discussion on several open issues. In Appendix A we include the key properties of the ETD(λ) trace iterates that are used in the analysis.

2. Preliminaries

In this section we describe the policy evaluation problem in the off-policy case, the ETD(λ) algorithm and its constrained version. We also review the results from our prior work (2015a) that are needed in this paper.

2.1 Off-policy Policy Evaluation

Let $\mathcal{S} = \{1, \dots, N\}$ be a finite set of states, and let \mathcal{A} be a finite set of actions. Without loss of generality we assume that for all states, every action in \mathcal{A} can be applied. If $a \in \mathcal{A}$ is applied at state $s \in \mathcal{S}$, the system moves to state s' with probability $p(s' | s, a)$ and yields a random reward with mean $r(s, a, s')$ and bounded variance, according to a probability distribution $q(\cdot | s, a, s')$. These are the parameters of the MDP model we consider; they are unknown to the learning algorithms to be introduced.

A *stationary policy* is a time-invariant decision rule that specifies the probability of taking an action at each state. When actions are taken according to such a policy, the states and actions (S_t, A_t) at times $t \geq 0$ form a (time-homogeneous) Markov chain on the space $\mathcal{S} \times \mathcal{A}$, with the marginal state process $\{S_t\}$ being also a Markov chain.

Let π and π^o be two given stationary policies, with $\pi(a | s)$ and $\pi^o(a | s)$ denoting the probability of taking action a at state s under π and π^o , respectively. While the system evolves under the policy π^o , generating a stream of state transitions and rewards, we wish to use these observations to evaluate the performance of the policy π , with respect to a discounted reward criterion, the definition of which will be given shortly. Here π is the target policy and π^o the behavior policy. It is allowed that $\pi^o \neq \pi$ (the off-policy case), provided that at each state, all actions taken by π can also be taken by π^o (cf. Assumption 1(ii) below).

Let $\gamma(s) \in [0, 1]$, $s \in \mathcal{S}$, be state-dependent discount factors, with $\gamma(s) < 1$ for at least one state. We measure the performance of π in terms of the expected discounted total rewards attained under π as follows: for each state $s \in \mathcal{S}$,

$$v_\pi(s) := \mathbb{E}^\pi \left[R_0 + \sum_{t=1}^{\infty} \gamma(S_1) \gamma(S_2) \cdots \gamma(S_t) \cdot R_t \mid S_0 = s \right], \quad (1)$$

where R_t is the random reward received at time t , and \mathbb{E}^π denotes expectation with respect to the probability distribution of the states, actions and rewards, (S_t, A_t, R_t) , $t \geq 0$, generated under the policy π . The function v_π on \mathcal{S} is called the *value function* of π . The special case of γ being a constant less than 1 corresponds to the γ -discounted reward criterion: $v_\pi(s) = \mathbb{E}^\pi [\sum_{j=0}^{\infty} \gamma^j R_t \mid S_0 = s]$. In the general case, by letting γ depend on the state, the formulation is able to also cover certain undiscounted total reward MDPs with termination;⁴ however, for v_π to be well-defined (i.e., to have the right-hand side of Equation 1 well-defined for each state), a condition on the target policy is needed, which is stated below and will be assumed throughout the paper.

Let P_π denote the transition matrix of the Markov chain on \mathcal{S} induced by π . Let Γ denote the $N \times N$ diagonal matrix with diagonal entries $\gamma(s)$, $s \in \mathcal{S}$.

Assumption 1 (conditions on the target and behavior policies)

- (i) *The target policy π is such that $(I - P_\pi \Gamma)^{-1}$ exists.*
- (ii) *The behavior policy π^o induces an irreducible Markov chain on \mathcal{S} , and moreover, for all $(s, a) \in \mathcal{S} \times \mathcal{A}$, $\pi^o(a | s) > 0$ if $\pi(a | s) > 0$.*

Under Assumption 1(i), the value function v_π in (1) is well-defined, and furthermore, v_π satisfies uniquely the Bellman equation⁵

$$v_\pi = r_\pi + P_\pi \Gamma v_\pi, \quad \text{i.e.,} \quad v_\pi = (I - P_\pi \Gamma)^{-1} r_\pi,$$

where r_π is the expected one-stage reward function under π (i.e., $r_\pi(s) = \mathbb{E}^\pi [R_0 \mid S_0 = s]$ for $s \in \mathcal{S}$).

2.2 The ETD(λ) Algorithm

Like the standard TD(λ) algorithm (Sutton, 1988; Tsitsiklis and Van Roy, 1997), the ETD(λ) algorithm (Sutton et al., 2016) approximates the value function v_π by a function of the form $v(s) = \phi(s)^\top \theta$, $s \in \mathcal{S}$, using a parameter vector $\theta \in \mathbb{R}^n$ and n -dimensional feature representations $\phi(s)$ for the states. (Here $\phi(s)$ is a column vector and $^\top$ stands for transpose.) In matrix notation, denote by Φ the $N \times n$ matrix with $\phi(s)$, $s \in \mathcal{S}$, as its rows. Then the columns of Φ span the subspace of approximate value functions, and the approximation problem is to find in that subspace a function $v = \Phi \theta \approx v_\pi$.

We focus on a general form of the ETD(λ) algorithm, which uses state-dependent λ values specified by a function $\lambda : \mathcal{S} \rightarrow [0, 1]$. Inputs to the algorithm are the states, actions and rewards, $\{(S_t, A_t, R_t)\}$, generated under the behavior policy π^o , where R_t is the random reward received upon the transition from state S_t to S_{t+1} with action A_t . The algorithm can access the following functions, in addition to the features $\phi(s)$:

4. We may view $v_\pi(s)$ as the expected (undiscounted) total rewards attained under π starting from the state s and up to a random termination time $\tau \geq 1$ that depends on the states in a Markovian way. In particular, if at time $t \geq 1$, the state is s and termination has not occurred yet, the probability of $\tau = t$ (terminating at time t) is $1 - \gamma(s)$. Then $v_\pi(s)$ can be equivalently written as $v_\pi(s) = \mathbb{E}^\pi [\sum_{j=0}^{\tau-1} R_t \mid S_0 = s]$.
5. One can verify this Bellman equation directly. It also follows from the standard MDP theory, as by definition v_π here can be related to a value function in a discounted MDP where the discount factors depend on state transitions, similar to discounted semi-Markov decision processes (see e.g., Puterman, 1994).

- (i) the state-dependent discount factor $\gamma(s)$ that defines v_π , as described earlier;
- (ii) $\lambda : \mathcal{S} \rightarrow [0, 1]$, which determines the single or multi-step Bellman equation for the algorithm (cf. the subsequent Equations 6-7 and Footnote 7);
- (iii) $\rho : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}_+$ given by $\rho(s, a) = \pi(a | s) / \pi^o(a | s)$ (with $0/0 = 0$), which gives the likelihood ratios for action probabilities that can be used to compensate for sampling states and actions according to the behavior policy π^o instead of the target policy π ;
- (iv) $i : \mathcal{S} \rightarrow \mathbb{R}_+$, which gives the algorithm additional flexibility to weight states according to the degree of “interest” indicated by $i(s)$.

The algorithm also uses a sequence $\alpha_t > 0, t \geq 0$, as stepsize parameters. We shall consider only deterministic $\{\alpha_t\}$.

To simplify notation, let

$$\rho_t = \rho(S_t, A_t), \quad \gamma_t = \gamma(S_t), \quad \lambda_t = \lambda(S_t).$$

ETD(λ) calculates recursively $\theta_t \in \mathbb{R}^n$, $t \geq 0$, according to

$$\theta_{t+1} = \theta_t + \alpha_t e_t \cdot \rho_t (R_t + \gamma_{t+1} \phi(S_{t+1})^\top \theta_t - \phi(S_t)^\top \theta_t), \quad (2)$$

where $e_t \in \mathbb{R}^n$, called the “eligibility trace,” is calculated together with two nonnegative scalar iterates (F_t, M_t) according to⁶

$$F_t = \gamma_t \rho_{t-1} F_{t-1} + i(S_t), \quad (3)$$

$$M_t = \lambda_t i(S_t) + (1 - \lambda_t) F_t, \quad (4)$$

$$e_t = \lambda_t \gamma_t \rho_{t-1} e_{t-1} + M_t \phi(S_t). \quad (5)$$

For $t = 0$, (e_0, F_0, θ_0) are given as an initial condition of the algorithm.

We recognize that the iteration (2) has the same form as TD(λ), but the trace e_t is calculated differently, involving an “emphasis” weight M_t on the state S_t , which itself evolves along with the iterate F_t , called the “follow-on” trace. If M_t is always set to 1 regardless of F_t and $i(\cdot)$, then the iteration (2) reduces to the off-policy TD(λ) algorithm in the case where γ and λ are constants.

2.3 Associated Bellman Equations and Approximation and Convergence Properties of ETD(λ)

Let Λ denote the diagonal matrix with diagonal entries $\lambda(s), s \in \mathcal{S}$. Associated with ETD(λ) is a generalized multistep Bellman equation of which v_π is the unique solution (Sutton, 1995):⁷

$$v = r_{\pi, \gamma}^\lambda + P_{\pi, \gamma}^\lambda v. \quad (6)$$

6. The definition (5) we use here differs slightly from the original definition of e_t used by Sutton et al. (2016), but the two are equivalent and (5) appears to be more convenient for our analysis.

7. For the details of this Bellman equation, we refer the readers to the early work (Sutton, 1995; Sutton and Barto, 1998) and the recent work (Sutton et al., 2016). We remark that similar to the standard one-step Bellman equation, which is a recursive relation that expresses v_π in terms of the expected one-stage reward and the expected total future rewards given by v_π itself, one can use the strong Markov property to derive other recursive relations satisfied by v_π , in which the expected one-stage reward is replaced by the expected rewards attained by π up to some random stopping time. This gives rise to a general class of Bellman equations, of which (6) is one example. Earlier works on using such equations in TD

Here $P_{\pi, \gamma}^\lambda$ is an $N \times N$ substochastic matrix, $r_{\pi, \gamma}^\lambda \in \mathbb{R}^N$ is a vector of expected discounted total rewards attained by π up to some random time depending on the function λ , and they can be expressed in terms of P_π and τ_π as

$$P_{\pi, \gamma}^\lambda = I - (I - P_\pi \Gamma \Lambda)^{-1} (I - P_\pi \Gamma), \quad r_{\pi, \gamma}^\lambda = (I - P_\pi \Gamma \Lambda)^{-1} \tau_\pi. \quad (7)$$

ETD(λ) aims to solve a projected version of the Bellman equation (6) (Sutton et al., 2016), which takes the following forms in the space of approximate value functions and in the space of the θ -parameters, respectively:

$$v = \Pi (r_{\pi, \gamma}^\lambda + P_{\pi, \gamma}^\lambda v), \quad v \in \text{column-space}(\Phi), \quad \iff \quad C\theta + b = 0, \quad \theta \in \mathbb{R}^n. \quad (8)$$

Here Π is a projection onto the approximation subspace with respect to a weighted Euclidean norm or seminorm, under a condition on the approximation architecture that will be explained shortly. The weights that define this norm also define the diagonal entries M_{ss} , $s \in \mathcal{S}$, of a diagonal matrix \bar{M} , which are given by

$$\text{diag}(\bar{M}) = d_{\pi^o, i}^\top (I - P_{\pi, \gamma}^\lambda)^{-1}, \quad \text{with } d_{\pi^o, i} \in \mathbb{R}^N, \quad d_{\pi^o, i}(s) = d_{\pi^o}(s) \cdot i(s), \quad s \in \mathcal{S}, \quad (9)$$

where $d_{\pi^o}(s) > 0$ denotes the steady state probability of state s for the behavior policy π^o , under Assumption 1(ii). For the corresponding linear equation in the θ -space in (8),

$$C = -\Phi^\top \bar{M} (I - P_{\pi, \gamma}^\lambda) \Phi, \quad b = \Phi^\top \bar{M} r_{\pi, \gamma}^\lambda. \quad (10)$$

From the expression (9) of the diagonal matrix \bar{M} , the most important difference between the earlier TD algorithms and ETD(λ) can be seen. For on-policy TD(λ), in stead of (9), the target matrix \bar{M} is determined by the steady state probabilities of the states under the target policy π under an ergodicity assumption (Tsitisklis and Van Roy, 1997), and for off-policy TD(λ), it is determined by the steady state probabilities $d_{\pi^o}(s)$ under the behavior policy π^o . Here, due to the emphatic weighting scheme (3)-(5), the diagonals of \bar{M} given by (9) reflect the occupation frequencies (with respect to $P_{\pi, \gamma}^\lambda$) of the target policy rather than the behavior policy.

Let $\|\cdot\|$ denote the (unweighted) Euclidean norm. The matrix C is said to be *negative definite* if there exists $c > 0$ such that $\theta^\top C \theta \leq -c \|\theta\|^2$ for all $\theta \in \mathbb{R}^n$; and *negative semidefinite* if in the preceding inequality $c = 0$. A salient property of ETD(λ) is that the matrix C is always negative semidefinite (Sutton et al., 2016), and under natural and mild conditions, C is negative definite. This is proved in our work (2015a) and summarized below.

Call those states s with $M_{ss} > 0$ *emphatized states* (define this set of states to be empty if \bar{M} given by Equation 9 is ill-defined, a case we will not encounter).

Assumption 2 (condition on the approximation architecture)

The set of feature vectors of *emphatized states*, $\{\phi(s) \mid s \in \mathcal{S}, M_{ss} > 0\}$, contains n linearly independent vectors.

Learning include the paper (Sutton, 1995) and Chap. 5.3 of the book (Bertsekas and Tsitsiklis, 1996). Recently, Ueno et al. (2011) considered an even broader class of Bellman equations using the concept of estimating equations from statistics, and Yu and Bertsekas (2012) focused on a special class of generalized Bellman equations and discussed their potential advantages from an approximation viewpoint. But an in-depth study of the application of such equations is still lacking currently. Because generalized Bellman equations offer flexible ways to address the bias vs. variance problem in learning the value functions of a policy, they are especially important and deserve further study, in our opinion.

Theorem 1 (Yu, 2015a, Prop. C.2) *Under Assumption 1, the matrix C is negative definite if and only if Assumption 2 holds.*

Assumption 2, which implies the linear independence of the columns of Φ , is satisfied in particular if the set of feature vectors, $\{\phi(s) \mid s \in \mathcal{S}, i(s) > 0\}$, contains n linearly independent vectors, since states with positive interest $i(s)$ are among the emphasized states.⁸ So this assumption can be easily satisfied in reinforcement learning without model knowledge.⁹

In view of Theorem 1, under Assumptions 1-2, the equation $C\theta + b = 0$ has a unique solution θ^* ; equivalently, $\Phi\theta^*$ is the unique solution to the projected Bellman equation (7):

$$\Phi\theta^* = \Pi(v_{\pi,\gamma}^\lambda + P_{\pi,\gamma}^\lambda \Phi\theta^*),$$

where Π is a well-defined projection operator that projects a vector in \mathbb{R}^N onto the approximation subspace with respect to the seminorm on \mathbb{R}^N given by

$$\sqrt{\sum_{s \in \mathcal{S}} \bar{M}_{ss} \cdot v(s)^2}, \quad \forall v \in \mathbb{R}^N$$

(which is a norm if $\bar{M}_{ss} > 0$ for all $s \in \mathcal{S}$). The relation between the approximate value function $v = \Phi\theta^*$ and the desired value function v_π , in particular, the approximation error, can be characterized by using the oblique projection viewpoint (Scherrer, 2010) for projected Bellman equations.¹⁰

The almost sure convergence of ETD(λ) to θ^* is proved in (Yu, 2015a, Theorem 2.2) under Assumptions 1 and 2, for diminishing stepsize satisfying $\alpha_t = O(1/t)$ and $\frac{\alpha_t - \alpha_{t+1}}{\alpha_t} = O(1/t)$. Despite this convergence guarantee, the stepsize range is too narrow for applications, as we discussed in the introduction. In this paper we will focus on constrained ETD(λ) algorithms that restrict the θ -iterates in a bounded set, but can operate with much larger stepsizes and also suffer less from the issue of high variance in off-policy learning. We will analyze their behavior under Assumptions 1 and 2, although our analysis extends to the case without Assumption 2 (see the discussion in Section 5.1).

8. This follows from the definition (9) of the diagonals \bar{M}_{ss} . Since $(I - P_{\pi,\gamma}^\lambda)^{-1} = I + \sum_{k=1}^{\infty} (P_{\pi,\gamma}^\lambda)^k \geq I$, we have $\text{diag}(M) = d_{\pi,\gamma}^{-1}(I - P_{\pi,\gamma}^\lambda)^{-1} \geq d_{\pi,\gamma}^{-1}$. Hence $i(s) > 0$ implies $\bar{M}_{ss} \geq d_{\pi,\gamma}^{-1} \cdot i(s) > 0$.

9. There is another way to verify Assumption 2 without calculating M . Suppose ETD(λ) starts from a state S_0 with $i(S_0) > 0$. Then it can be shown that if $S_t = s$ and $M_t > 0$, we must have $M_{ss} > 0$. This means that as soon as we find among states S_t with emphasis weights $M_t > 0$ n states that have linearly independent feature vectors, we can be sure that Assumption 2 is satisfied.

10. Briefly speaking, Scherrer (2010) showed that the solutions of projected Bellman equations are oblique projections of v_π on the approximation subspace. An oblique projection is defined by two nonorthogonal subspaces of equal dimensions and is the projection onto the first subspace orthogonally to the second (Saad, 2003). In the special case of ETD(λ), the first of these two subspaces is the approximation subspace $\{v \in \mathbb{R}^N \mid v = \Phi\theta \text{ for some } \theta \in \mathbb{R}^n\}$, and the second is the image of the approximation subspace under the linear transformation $(I - P_{\pi,\gamma}^\lambda)^\top M$. Essentially it is the angle between the two subspaces that determines the approximation bias $\Phi\theta^* - \Pi v_\pi$ in the worst case, for a worst-case choice of $r_{\pi,\gamma}^\lambda$. (For details, see also Yu and Bertsekas 2012, Sec. 2.2.) Recently, for the case of constant λ, i and γ , Hallak et al. (2016) derived bounds on the approximation bias that are based on contraction arguments and are comparable to the bound for on-policy TD(λ) (Tsitiklis and Van Roy, 1997). These bounds lie above the bounds given by the oblique projection view (cf. Yu and Bertsekas, 2010; Yu and Bertsekas, 2012, Sec. 2.2); however, they are expressed in terms of λ and γ , so they give us explicit numbers instead of analytical expressions to bound the approximation bias.

2.4 Constrained ETD(λ), Averaged Processes and Mean ODE

We consider first a constrained version of ETD(λ) that simply scales the θ -iterates, if necessary, to keep them bounded:

$$\theta_{t+1} = \Pi_B(\theta_t + \alpha_t \tilde{e}_t \cdot \rho_t (R_t + \gamma_{t+1} \phi(S_{t+1})^\top \theta_t - \phi(S_t)^\top \theta_t)), \quad (11)$$

where Π_B is the Euclidean projection onto a closed ball $B \subset \mathbb{R}^n$ at the origin with radius r_B : $B = \{\theta \in \mathbb{R}^n \mid \|\theta\| \leq r_B\}$. Under Assumptions 1 and 2, when the radius r_B is sufficiently large (greater than the threshold given in Lemma 1 below), from any given initial (e_0, F_0, θ_0) , the algorithm (11) converges almost surely to θ^* , for diminishing stepsize $\alpha_t = O(1/t)$ (Yu, 2015a, Theorem 4.1).

Our interest in this paper is to apply (11) with a much larger range of stepsizes, in particular, constant stepsize or stepsize that diminishes much more slowly than $O(1/t)$. In Sections 3 and 4, we will analyze the algorithm (11) and its two variants for such stepsizes. To prepare for the analysis, in the rest of this section, we review several results from our prior work (2015a) that will be needed.

First, we discuss about the ‘‘mean ODE’’ that we wish to associate with (11). It is the projected ODE

$$\dot{x} = \bar{h}(x) + z, \quad z \in -\mathcal{N}_B(x), \quad (12)$$

where the function \bar{h} is the left-hand side of the equation $Cx + b = 0$ we want to solve:

$$\bar{h}(x) = Cx + b; \quad (13)$$

$\mathcal{N}_B(x)$ is the normal cone of B at x (i.e., $\mathcal{N}_B(x) = \{0\}$ for x in the interior of B and $\mathcal{N}_B(x) = \{ax \mid a \geq 0\}$ for x on the boundary of B); and z is the boundary reflection term that cancels out the component of $\bar{h}(x)$ in $\mathcal{N}_B(x)$ (i.e., $z = -y$ where y is the projection of $\bar{h}(x)$ on $\mathcal{N}_B(x)$), and it is the ‘‘minimal force’’ needed to keep the solution $x(\cdot)$ of (12) in B (Kushner and Yin, 2003, Chap. 4.3).

The negative definiteness of the matrix C ensures that when the radius of B is sufficiently large, the boundary reflection term is zero for all $x \in B$ and the projected ODE (12) has no stationary points other than θ^* (see Yu 2015a, Sec. 4.1 for a simple proof):

Lemma 1 *Let $c > 0$ be such that $x^\top Cx \leq -c|x|^2$ for all $x \in \mathbb{R}^n$. Suppose B has a radius $r_B > |b|/c$. Then θ^* lies in the interior of B ; a solution $x(\tau), \tau \in [0, \infty)$, to the projected ODE (12) for an initial condition $x(0) \in B$ coincides with the unique solution to $\dot{x} = \bar{h}(x)$, with the boundary reflection term being $z(\cdot) \equiv 0$; and the only solution $x(\tau), \tau \in (-\infty, +\infty)$, of (12) in B is $x(\cdot) \equiv \theta^*$.*

Informally speaking, suppose we have proved that (12) is the mean ODE for the algorithm (11) under stepsizes of our interest. Then applying powerful convergence theorems from stochastic approximation theory (Kushner and Yin, 2003), we can assert that the iterates θ_t will eventually ‘‘follow closely’’ a solution of the mean ODE. This together with the solution property of the mean ODE given in Lemma 1 will then give us a characterization of the asymptotic behavior of the algorithm (11) for a constraint set B with sufficiently large radius.

Several properties of the ETD(λ) iterates will be important in proving that (12) is indeed the mean ODE for (11) and reflects its average dynamics. We now discuss two such properties (other key properties will be given in Appendix A). They concern the ergodicity of the Markov chain $\{(S_t, A_t, e_t, F_t)\}$ on the joint space of states, actions and traces, and the convergence of certain averaged sequences associated with the algorithm (11). They will also be useful in analyzing variants of (11).

Let $Z_t = (S_t, A_t, e_t, F_t)$, $t \geq 0$. It was shown in (Yu, 2015a) that under Assumption 1, $\{Z_t\}$ is a weak Feller Markov chain¹¹ on the infinite state space $S \times \mathcal{A} \times \mathbb{R}^{r+1}$ and is ergodic. Specifically, on a metric space, a sequence of probability measures $\{\mu_t\}$ is said to *converge weakly* to a probability measure μ if for any bounded continuous function f , $\int f d\mu_t \rightarrow \int f d\mu$ as $t \rightarrow \infty$ (Dudley, 2002, Chap. 9.3). We are interested in the weak convergence of the occupation probability measures of the process $\{Z_t\}$, where for each initial condition $Z_0 = z$, the *occupation probability measures* $\mu_{z,t}$, $t \geq 0$, are defined by $\mu_{z,t}(D) = \frac{1}{t+1} \sum_{k=0}^t \mathbb{1}(Z_k \in D)$ for any Borel subset D of $S \times \mathcal{A} \times \mathbb{R}^{r+1}$, with $\mathbb{1}(\cdot)$ denoting the indicator function.

Theorem 2 (ergodicity of $\{Z_t\}$; Yu, 2015a, Theorem 3.2) *Under Assumption 1, the Markov chain $\{Z_t\}$ has a unique invariant probability measure ζ , and for each initial condition $Z_0 = z$, the sequence $\{\mu_{z,t}\}$ of occupation probability measures converges weakly to ζ , almost surely.*

Let \mathbb{E}_ζ denote expectation with respect to the stationary process $\{Z_t\}$ with ζ as its initial distribution. By the definition of weak convergence, the weak convergence of $\{\mu_{z,t}\}$ given in Theorem 2 implies that for each given initial condition of Z_0 , the averages $\frac{1}{t} \sum_{k=0}^{t-1} f(Z_k)$ converge almost surely to $\mathbb{E}_\zeta\{f(Z_0)\}$ for any bounded continuous function f .¹² To study the average dynamics of the algorithm (11), however, we need to also consider unbounded functions. In particular, the function related to both (11) and the unconstrained ETD(λ) is $h : \mathbb{R}^n \times \Xi \rightarrow \mathbb{R}^n$,

$$h(\theta, \xi) = e \cdot \rho(s, a) (r(s, a, s') + \gamma(s') \phi(s')^\top \theta - \phi(s)^\top \theta), \quad (14)$$

where

$$\xi = (e, F, s, a, s') \in \Xi := \mathbb{R}^{n+1} \times S \times \mathcal{A} \times S.$$

Writing ξ_t for the traces and transition at time t : $\xi_t = (e_t, F_t, S_t, A_t, S_{t+1})$, we can express the recursion (11) equivalently as

$$\theta_{t+1} = \Pi_B(\theta_t + \alpha_t h(\theta_t, \xi_t) + \alpha_t e_t \cdot \tilde{\omega}_{t+1}), \quad (15)$$

where $\tilde{\omega}_{t+1} = \rho_t(R_t - r(S_t, A_t, S_{t+1}))$ is the noise part of the observed reward.

The convergence to $h(\theta)$ of the averaged sequence $\frac{1}{t} \sum_{k=0}^{t-1} h(\theta, \xi_k)$, with θ held fixed and t going to infinity, will be needed to prove that (12) is the mean ODE of (11). Since

11. See Section 4.3.1 or the book by Meyn and Tweedie (2009, Chap. 6) for the definition and properties of weak Feller Markov chains.

12. With the usual discrete topology for the finite space $S \times \mathcal{A}$ and the usual topology for the Euclidean space \mathbb{R}^{n+1} , the space $S \times \mathcal{A} \times \mathbb{R}^{n+1}$ equipped with the product topology is metrizable. A continuous function $f(s, a, e, F)$ on this space is a function that is continuous in (e, F) for each $(s, a) \in S \times \mathcal{A}$.

$\bar{h}(\theta) = C\theta + b$, this convergence for each fixed θ can be identified with the convergence of the matrix and vector iterates calculated by ELSTD(λ)—the least-squares version of ETD(λ)—to approximate the left-hand side of the equation $C\theta + b = 0$. It was proved in our work (2015a) as a special case of the convergence of averaged sequences for a larger set of functions including $h(\theta, \cdot)$. Since this general result will be needed in analyzing variants of (11), we give its formulation here.

Throughout the rest of the paper, we let $\|\cdot\|$ denote the infinity norm of a Euclidean space, and we use this notation for both vectors and matrices (viewed as vectors). For \mathbb{R}^m -valued random variables X_t , we say $\{X_t\}$ converges to a random variable X in mean if $\mathbb{E}\|X_t - X\| \rightarrow 0$ as $t \rightarrow \infty$.

Consider a vector-valued function $g : \Xi \rightarrow \mathbb{R}^m$ such that with $\xi = (e, F, s, a, s')$, $g(\xi)$ is Lipschitz continuous in (e, F) uniformly in (s, a, s') . That is, there exists a finite constant L_g such that for any $(e, F)_t, (\hat{e}, \hat{F}) \in \mathbb{R}^{r+1}$,

$$\|g(e, F, s, a, s') - g(\hat{e}, \hat{F}, s, a, s')\| \leq L_g \|(e, F) - (\hat{e}, \hat{F})\|, \quad \forall (s, a, s') \in S \times \mathcal{A} \times S. \quad (16)$$

For each $\theta \in \mathbb{R}^n$, the function $h(\theta, \cdot)$ in (14) is a special case of g . The convergence of the averaged sequence $\frac{1}{t} \sum_{k=0}^{t-1} g(\xi_k)$ is given in the theorem below; the part on convergence in mean will be used frequently later in this paper. The convergence of $\frac{1}{t} \sum_{k=0}^{t-1} h(\theta, \xi_k)$ then follows as a special case.

Theorem 3 (convergence of averaged sequences; Yu, 2015a, Theorems 3.1-3.3)

Let g be a vector-valued function satisfying the Lipschitz condition (16). Then under Assumption 1, $\mathbb{E}_\zeta[\|g(\xi_0)\|] < \infty$ and for any given initial $(e_0, F_0) \in \mathbb{R}^{r+1}$, as $t \rightarrow \infty$, $\frac{1}{t} \sum_{k=0}^{t-1} g(\xi_k)$ converges to $\bar{g} = \mathbb{E}_\zeta[g(\xi_0)]$ in mean and almost surely.

Corollary 1 (Yu, 2015a, Theorem 2.1) *Under Assumption 1, for the functions \bar{h}, h given in (13), (14) respectively, the following hold: For each $\theta \in \mathbb{R}^n$, $\mathbb{E}_\zeta[\|h(\theta, \xi_0)\|] < \infty$ and $\bar{h}(\theta) = \mathbb{E}_\zeta[h(\theta, \xi_0)]$; and for any given initial $(e_0, F_0) \in \mathbb{R}^{r+1}$, as $t \rightarrow \infty$, $\frac{1}{t} \sum_{k=0}^{t-1} h(\theta, \xi_k)$ converges to $\bar{h}(\theta)$ in mean and almost surely.*

3. Convergence Results for Constrained ETD(λ)

In this section we present the convergence properties of the constrained ETD(λ) algorithm (11) and several variants of it, for constant stepsize and for stepsize that diminishes slowly. We will explain briefly how the results are obtained, leaving the detailed analyses to Section 4. The first set of results about the algorithm (11) will be given first in Section 3.1, followed by similar results in Section 3.2 for two variant algorithms that have biases but can mitigate the variance issue in off-policy learning better. These results are obtained through applying two general convergence theorems from (Kushner and Yin, 2003), which concern weak convergence of stochastic approximation algorithms for diminishing and constant stepsize. Finally, the constant-stepsize case will be analyzed further in Section 3.3, in order to refine some results of Sections 3.1-3.2 so that the asymptotic behavior of the algorithms for a fixed stepsize can be characterized explicitly. In that subsection, besides the three algorithms just mentioned, we will also discuss another variant algorithm with perturbation.

Regarding notation, recall that $\mathbb{1}(\cdot)$ is the indicator function, $|\cdot|$ stands for the usual (unweighted) Euclidean norm and $\|\cdot\|$ the infinity norm for \mathbb{R}^m . We denote by $N_\delta(D)$ the δ -neighborhood of a set $D \subset \mathbb{R}^m$: $N_\delta(D) = \{x \in \mathbb{R}^m \mid \inf_{y \in D} |x - y| \leq \delta\}$, and we write $N_\delta(\theta^*)$ for the δ -neighborhood of θ^* . For the iteration index t , the notation $t \in [k_1, k_2]$ or $t \in [k_1, k_2]$ will be used to mean that the range of t is the set of integers in the interval $[k_1, k_2]$ or $[k_1, k_2)$. More definitions and notation will be introduced later where they are needed.

3.1 Main Results

We consider first the algorithm (11) for diminishing stepsize. Let the stepsize change slowly in the following sense.

Assumption 3 (condition on diminishing stepsize) *The (deterministic) nonnegative sequence $\{\alpha_t\}$ satisfies that $\sum_{t \geq 0} \alpha_t = \infty$, $\alpha_t \rightarrow 0$ as $t \rightarrow \infty$, and for some sequence of integers $m_t \rightarrow \infty$,*

$$\lim_{t \rightarrow \infty} \sup_{0 \leq j \leq m_t} \left| \frac{\alpha_{t+j} - 1}{\alpha_t} \right| = 0. \quad (17)$$

The condition (17) is the condition A.8.2.8 in (Kushner and Yin, 2003, Chap. 8) and allows stepsizes much larger than $O(1/t)$. We can have $\alpha_t = O(t^{-\beta})$, $\beta \in (0, 1]$, and even larger stepsizes are possible. For example, partition the time interval $[0, \infty)$ into increasingly longer intervals $I_k, k \geq 0$, and set α_t to be constant within each interval I_k . Then the condition (17) can be fulfilled by letting the constants for each I_k decrease as $O(k^{-\beta})$, $\beta \in (0, 1]$.

We now state the convergence result. For any $T > 0$, let $m(k, T) = \min\{t \geq k \mid \sum_{j=k}^t \alpha_j > T\}$. If we draw a continuous timeline and put each iteration of the algorithm at a specific moment, with the stepsize α_j being the length of time between iterations j and $j+1$, then $m(k, T)$ is the latest iteration before time T has elapsed since the k -th iteration. If $\alpha_t = O(t^{-\beta})$, $\beta \in (0, 1]$, for example, then for fixed T , there are $O(k^\beta)$ iterates between the k -th and $m(k, T)$ -th iteration.

Recall that Assumption 1, Assumption 2, and Lemma 1 are given in Sections 2.1, 2.3, and 2.4, respectively.

Theorem 4 (convergence of constrained ETD with diminishing stepsize)

Suppose Assumptions 1-2 hold and the radius of B exceeds the threshold given in Lemma 1. Let $\{\theta_t\}$ be generated by the algorithm (11) with stepsize $\{\alpha_t\}$ satisfying Assumption 3, from any given initial condition (ϵ_0, F_0) . Then there exists a sequence $T_k \rightarrow \infty$ such that for any $\delta > 0$,

$$\limsup_{k \rightarrow \infty} \mathbf{P} \left(\theta_t \notin N_\delta(\theta^*), \text{ some } t \in [k, m(k, T_k)] \right) = 0.$$

This theorem implies $\theta_t \rightarrow \theta^*$ in probability. Since $\{\theta_t\}$ is bounded, by (Dudley, 2002, Theorem 10.3.6), θ_t must also converge to θ^* in mean:

Corollary 2 (convergence in mean) *In the setting of Theorem 4, $\mathbb{E}[\|\theta_t - \theta^*\|] \rightarrow 0$ as $t \rightarrow \infty$.*

Another important note is that the conclusion of Theorem 4 is much stronger than that $\theta_t \rightarrow \theta^*$ in probability. Here as $k \rightarrow \infty$, we consider an increasingly longer segment $[k, m(k, T_k)]$ of iterates, and are able to conclude that the probability of that *entire segment* being inside an arbitrarily small neighborhood of θ^* approaches 1. This is the power of the weak convergence methods (Kushner and Clark, 1978; Kushner and Shwartz, 1984; Kushner and Yin, 2003), by which our conclusion is obtained.

In the case of constant stepsize, we consider all the trajectories that can be produced by the algorithm (11) using some constant stepsize, and we ask what the properties of these trajectories are in the limit as the stepsize parameter approaches 0. Here there is a common timeline used in relating trajectories generated with different stepsizes (and it comes from the ODE-based analysis): we imagine again a continuous timeline, along which we put the iterations at moments that are evenly separated in time by α , if the stepsize parameter is α . The scalars T, T_α in the theorem below represent amounts of time with respect to this continuous timeline.

Theorem 5 (convergence of constrained ETD with constant stepsize)

Suppose Assumptions 1-2 hold and the radius of B exceeds the threshold given in Lemma 1. For each $\alpha > 0$, let $\{\theta_t^\alpha\}$ be generated by the algorithm (11) with constant stepsize α , from any given initial condition (ϵ_0, F_0) . Let $\{k_\alpha \mid \alpha > 0\}$ be any sequence of nonnegative integers that are nondecreasing as $\alpha \rightarrow 0$. Then the following hold:

- (i) For any $\delta > 0$,
- $$\lim_{T \rightarrow \infty} \lim_{\alpha \rightarrow 0} \frac{1}{T/\alpha} \sum_{t=k_\alpha}^{k_\alpha + \lceil T/\alpha \rceil} \mathbb{1}(\theta_t^\alpha \in N_\delta(\theta^*)) = 1 \quad \text{in probability.}$$
- (ii) Let $\alpha k_\alpha \rightarrow \infty$ as $\alpha \rightarrow 0$. Then there exists a sequence $\{T_\alpha \mid \alpha > 0\}$ with $T_\alpha \rightarrow \infty$ as $\alpha \rightarrow 0$, such that for any $\delta > 0$,
- $$\limsup_{\alpha \rightarrow 0} \mathbf{P} \left(\theta_t^\alpha \notin N_\delta(\theta^*), \text{ some } t \in [k_\alpha, k_\alpha + T_\alpha/\alpha] \right) = 0.$$

Part (ii) above is similar to Theorem 4. Here as $\alpha \rightarrow 0$, an increasingly longer segment $[k_\alpha, k_\alpha + T_\alpha/\alpha]$ of the tail of the trajectory $\{\theta_t^\alpha\}$ is considered, and it is concluded that the probability of that *entire segment* being inside an arbitrarily small neighborhood of θ^* approaches 1. Part (i) above, roughly speaking, says that as α diminishes, within the segment $[k_\alpha, k_\alpha + T/\alpha]$, the fraction of iterates θ_t^α that lie in a small δ -neighborhood of θ^* approaches 1 for sufficiently large T .

We give the proofs of Theorems 4-5 in Section 4.1. As mentioned earlier, most of our efforts will be to use the properties of ETD iterates to show that the conditions of two general convergence theorems from stochastic approximation theory (Kushner and Yin, 2003, Theorems 8.2.2, 8.2.3) are satisfied by the algorithm (11). After that we can specialize the conclusions of those theorems to obtain Theorems 4-5. Specifically, after furnishing their conditions, applying (Kushner and Yin, 2003, Theorems 8.2.2, 8.2.3) will give us directly the desired conclusions in Theorems 4-5 with $N_\delta(L_B)$ in place of $N_\delta(\theta^*)$, where $N_\delta(L_B)$ is the δ -neighborhood of the *limit set* L_B for the projected ODE (12). This limit set is defined as follows:

$$L_B := \bigcap_{\tau > 0} \overline{\bigcup_{x(0) \in B} \{x(\tau), \tau \geq \tau\}}$$

where $x(\tau)$ is a solution of the projected ODE (12) with initial condition $x(0)$, the union is over all the solutions with initial $x(0) \in B$, and D for a set D denotes taking the closure of D . It can be shown that $L_B = \{\theta^*\}$ under our assumptions, so Theorems 4-5 will then follow as special cases of (Kushner and Yin, 2003, Theorems 8.2.2, 8.2.3).

Remark 1 (on weak convergence methods) The theorems from the book (Kushner and Yin, 2003) which we will apply are based on the weak convergence methods. While it is beyond the scope of this paper to explain these powerful methods, let us mention here a few basic facts about them to elucidate the origin of the convergence theorems we gave above. In the Framework of (Kushner and Yin, 2003), one studies a trajectory of iterates produced by an algorithm by working with continuous-time processes that are piecewise constant or linear interpolations of the iterates. (Often one also left-shifts a trajectory of iterates to bring the “asymptotic part” of the trajectory closer to the origin of the continuous time axis.) In the case of our problem, for example, for diminishing stepsize, these continuous-time processes are $x^k(\tau)$, $\tau \in [0, \infty)$, indexed by $k \geq 0$, where for each k , x^k is a piecewise constant interpolation of θ_{k+t} , $t \geq 0$, given by $x^k(\tau) = \theta_k$ for $\tau \in [0, \alpha_k)$ and $x^k(\tau) = \theta_{k+t}$ for $\tau \in [\sum_{m=0}^{t-1} \alpha_{k+m}, \sum_{m=0}^t \alpha_{k+m})$, $t \geq 1$. Similarly, for constant stepsize, the continuous-time processes involved are $x^\alpha(\tau)$, $\tau \in [0, \infty)$, indexed by $\alpha > 0$, and for each α , x^α is a piecewise constant interpolation of $\theta_{k+\alpha t}$, $t \geq 0$, given by $x^\alpha(\tau) = \theta_{k+\alpha t}$ for $\tau \in [t\alpha, (t+1)\alpha)$. The behavior of the sequence $\{x^k\}$ or $\{x^\alpha\}$ as $k \rightarrow \infty$ or $\alpha \rightarrow 0$, tells us the asymptotic properties of the algorithm as the number of iterations grows to infinity or as the stepsize parameter approaches 0. With the weak convergence methods, one considers the probability distributions of the continuous-time processes in such sequences, and analyze the convergence of these probability distributions and their limiting distributions along any subsequences. Here each continuous-time process takes values in a space of vector-valued functions on $[0, \infty)$ or $(-\infty, \infty)$ that are right-continuous and have left-hand limits, and this function space equipped with an appropriate metric, known as the Skorohod metric, is a complete separable metric space (Kushner and Yin, 2003, p. 238-240). On this space, one analyzes the weak convergence of the probability distributions of the continuous-time processes. Under certain conditions on the algorithm, the general conclusions from (Kushner and Yin, 2003, Theorems 8.2.2, 8.2.3) are that any subsequence of these probability distributions contains a further subsequence which is convergent, and that all the limiting probability distributions must assign the full measure 1 to the set of solutions of the mean ODE associated with the algorithm. This general weak convergence property then yields various conclusions about the asymptotic behavior of the algorithm and its relation with the mean ODE solutions. When further combined with the solution properties of the mean ODE, it leads to specific results such as the theorems we give in this section. ■

3.2 Two Variants of Constrained ETTD(λ) with Biases

We now consider two simple variants of (11). They constrain the ETTD iterates even more, at a price of introducing biases in this process, so that unlike (11), they can no longer get to θ^* arbitrarily closely. Instead they aim at a small neighborhood of θ^* , the size of which depends on how they modify the ETTD iterates. On the other hand, because the trace iterates $\{(e_t, F_t)\}$ can have unbounded variances and are also naturally unbounded in

common off-policy situations (see discussions in Yu, 2012, Prop. 3.1 and Footnote 3, p. 3320-3322 and Yu, 2015a, Remark A.1, p. 23), these variant algorithms have the advantage that they make the θ -iterates more robust against the drastic changes that can occur to the trace iterates. Indeed our definition of the variant algorithms below follows a well-known approach to “robustifying” algorithms in stochastic approximation theory (see discussions in Kushner and Yin, 2003, p. 23 and p. 141).

The two variant algorithms are defined as follows. For each $K > 0$, let $\psi_K : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be a bounded Lipschitz continuous function such that

$$\|\psi_K(x)\| \leq \|x\| \quad \forall x \in \mathbb{R}^n, \quad \text{and} \quad \psi_K(x) = x \quad \text{if } \|x\| \leq K. \quad (18)$$

(For instance, let $\psi_K(x) = \pi x/|x|$ if $|x| \geq \bar{r}$ and $\psi_K(x) = x$ otherwise, for $\bar{r} = \sqrt{\pi}K$; or let $\psi_K(x)$ be the result of truncating each component of x to be within $[-K, K]$.) For the first variant of the algorithm (11), we replace e_t in (11) by $\psi_K(e_t)$:

$$\theta_{t+1} = \Pi_B \left(\theta_t + \alpha_t \psi_K(e_t) \cdot \rho_t (R_t + \gamma_{t+1} \phi(S_{t+1})^\top \theta_t - \phi(S_t)^\top \theta_t) \right). \quad (19)$$

For the second variant, we apply ψ_K to bound the entire increment in (11) before it is multiplied by the stepsize α_t and added to θ_t :

$$\theta_{t+1} = \Pi_B \left(\theta_t + \alpha_t \psi_K(Y_t) \right), \quad \text{where } Y_t = e_t \cdot \rho_t (R_t + \gamma_{t+1} \phi(S_{t+1})^\top \theta_t - \phi(S_t)^\top \theta_t). \quad (20)$$

As will be proved later, these two algorithms are associated with mean ODEs of the form,

$$\dot{x} = \bar{h}_K(x) + z, \quad z \in -N_B(x), \quad (21)$$

where $\bar{h}_K : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is determined by each algorithm and deviates from the function $h(x) = Cx + b$ due to the alterations introduced by ψ_K . This ODE is similar to the projected ODE (12), except that since \bar{h}_K is an approximation of \bar{h} , θ^* is no longer a stable or stationary point for the mean ODE (21). The two variant algorithms thus have a bias in their θ -iterates, and the bias can be made smaller by choosing a larger K . This is reflected in the two convergence theorems given below. They are similar to the previous two theorems for the algorithm (11), except that now given a desired small neighborhood of θ^* , a sufficiently large K needs to be used in order for the θ -iterates to reach that neighborhood of θ^* and exhibit properties similar to those shown in the previous case.

Theorem 6 (constrained ETTD variants with diminishing stepsize)

In the setting of Theorem 4, let $\{\theta_t\}$ be generated instead by the algorithm (19) or (20), with a bounded Lipschitz continuous function ψ_K satisfying (18), and with stepsize $\{\alpha_t\}$ satisfying Assumption 3. Then for each $\delta > 0$, there exists $K_\delta > 0$ such that if $K \geq K_\delta$, then it holds for some sequence $T_k \rightarrow \infty$ that

$$\limsup_{k \rightarrow \infty} \mathbf{P} \left(\theta_t \notin N_\delta(\theta^*), \text{ some } t \in [k, m(k, T_k)] \right) = 0.$$

Theorem 7 (constrained ETTD variants with constant stepsize)

In the setting of Theorem 5, let $\{\theta_t\}$ be generated instead by the algorithm (19) or (20), with a bounded Lipschitz continuous function ψ_K satisfying (18) and with constant stepsize $\alpha > 0$. Let $\{K_\alpha | \alpha > 0\}$ be any sequence of nonnegative integers that are nondecreasing as $\alpha \rightarrow 0$. Then for each $\delta > 0$, there exists $K_\delta > 0$ such that the following hold if $K \geq K_\delta$:

- (i)
$$\lim_{T \rightarrow \infty} \lim_{\alpha \rightarrow 0} \frac{1}{T/\alpha} \sum_{t=k_\alpha}^{k_\alpha + \lceil T/\alpha \rceil} \mathbb{1}(\theta_t^\alpha \in N_\delta(\theta^*)) = 1 \quad \text{in probability.}$$
- (ii) Let $\alpha k_\alpha \rightarrow \infty$ as $\alpha \rightarrow 0$. Then there exists a sequence $\{T_\alpha \mid \alpha > 0\}$ with $T_\alpha \rightarrow \infty$ as $\alpha \rightarrow 0$, such that
- $$\limsup_{\alpha \rightarrow 0} \mathbf{P} \left(\theta_t^\alpha \notin N_\delta(\theta^*), \text{ some } t \in [k_\alpha, k_\alpha + T_\alpha/\alpha] \right) = 0.$$

We give the proofs of the above two theorems in Section 4.2. Because the proofs are similar for the two variant algorithms, we include in this paper only the proofs for the first variant—the proofs for the second variant can be found in the arXiv version of this paper (Yu, 2015b).

The proof arguments are largely the same as those that we will use first in Section 4.1 to prove Theorems 4-5 for the algorithm (11). Indeed, for all the three algorithms, the main proof step is the same, which is to apply the general conclusions of (Kushner and Yin, 2003, Theorems 8.2.2, 8.2.3) to establish the connection between the iterates of an algorithm and the solutions of an associated mean ODE, and this step does not concern what the solutions of the ODE are actually. (For the two variant algorithms, verifying that the conditions of Theorems 8.2.2, 8.2.3 in Kushner and Yin, 2003 are met is, in fact, easier because various functions involved in the analysis become bounded due to the use of the bounded function ψ_K .) For the two variant algorithms, the result of this step is that the same conclusions given in Theorems 4-5 hold with $N_\delta(L_B)$ in place of $N_\delta(\theta^*)$, where L_B is the limit set of the projected mean ODE (21) associated with each variant algorithm. To attain Theorems 6-7, we then combine this with the fact that by choosing K sufficiently large, one can make the limit set $L_B \subset N_\delta(\theta^*)$ for an arbitrarily small δ .

3.3 More about the Constant-stepsize Case

For the constant-stepsize case, the results given in Theorems 5 and 7 bear similarities to their counterparts for the diminishing stepsize case given in Theorems 4 and 6. However, they characterize the behavior of the iterates in the limit as the stepsize parameter approaches 0, and deal with only a finite segment of the iterates for each stepsize (although in their part (ii) both the segment's length $T_\alpha/\alpha \rightarrow \infty$ and its starting position $k_\alpha \rightarrow \infty$ as $\alpha \rightarrow 0$). So unlike in the diminishing stepsize case, these results do not tell us explicitly about the behavior of θ_t^α for a fixed stepsize α as we take t to infinity.

The purpose of the present subsection is to analyze further the case of a fixed stepsize just mentioned. We observe that for a fixed stepsize α , the iterates θ_t^α together with $Z_t = (S_t, A_t, \epsilon_t, F_t)$ form a weak Feller Markov chain $\{(Z_t, \theta_t^\alpha)\}$ (see Lemma 4, Section 4.3.1). Thus we can apply several ergodic theorems for weak Feller Markov chains (Meyn, 1989; Meyn and Tweedie, 2009) to analyze the constant-stepsize case and combine the implications from these theorems with the results we obtained previously using the weak convergence methods from stochastic approximation theory.

We now present our results using this approach. Let \mathcal{M}_α denote the set of invariant probability measures of the Markov chain $\{(Z_t, \theta_t^\alpha)\}$. This set depends on the particular

algorithm used to generate the θ -iterates, but we shall use the notation \mathcal{M}_α for all the algorithms we discuss here, for notational simplicity. We know that $\{Z_t\}$ has a unique invariant probability measure (Theorem 2, Section 2.4), but it need not be so for the Markov chain $\{(Z_t, \theta_t^\alpha)\}$ when $\{\theta_t^\alpha\}$ is generated by the algorithm (11) or its two variants. The set \mathcal{M}_α can therefore have multiple elements (it is nonempty; see Prop. 6, Section 4.3.2). We denote by \mathcal{M}_α the set that consists of the marginal of μ on B (the space of the θ 's), for all the invariant probability measures $\mu \in \mathcal{M}_\alpha$.

As in the previous analysis, we are interested in the behavior of multiple consecutive θ -iterates. In order to characterize that, we consider for each $m \geq 1$, the Markov chain

$$\left\{ ((Z_t, \theta_t^\alpha), (Z_{t+1}, \theta_{t+1}^\alpha), \dots, (Z_{t+m-1}, \theta_{t+m-1}^\alpha)) \right\}_{t \geq 0}$$

(i.e., each state now consists of m consecutive states of the chain $\{(Z_t, \theta_t^\alpha)\}$). We shall refer to this chain as the *m-step version* of $\{(Z_t, \theta_t^\alpha)\}$. Similar to \mathcal{M}_α , denote by \mathcal{M}_α^m the set of invariant probability measures of the m -step version of $\{(Z_t, \theta_t^\alpha)\}$, and correspondingly define \mathcal{M}_α^m to be the set of marginals of μ on B^m for all $\mu \in \mathcal{M}_\alpha^m$. The set \mathcal{M}_α^m is, of course, determined by \mathcal{M}_α , since each invariant probability measure in \mathcal{M}_α^m is just the m -dimensional distribution of a stationary Markov chain $\{(Z_t, \theta_t^\alpha)\}$.

Our first result, given in Theorem 8 below, says that for the algorithm (11), as the stepsize α approaches zero, the invariant probability measures in \mathcal{M}_α^m will concentrate their masses on an arbitrarily small neighborhood of $(\theta^*, \dots, \theta^*)$ (m copies of θ^*). Moreover, for a fixed stepsize, as the number of iterations grows to infinity, the expected maximal deviation of the m consecutive averaged iterates from θ^* can be bounded in terms of the masses those invariant probability measures assign to the vicinities of $(\theta^*, \dots, \theta^*)$. Here by averaged iterates, we mean

$$\bar{\theta}_t^\alpha = \frac{1}{t} \sum_{k=0}^{t-1} \theta_k^\alpha, \quad \forall t \geq 1, \quad (22)$$

and we shall refer to $\{\bar{\theta}_t^\alpha\}$ as the *averaged sequence* corresponding to $\{\theta_t^\alpha\}$. This iterative averaging is also known as ‘‘Polyak-averaging’’ when it is applied to accelerate the convergence of the θ -iterates (see Polyak and Juditsky, 1992; Kushner and Yin, 2003, Chap. 10; and the references therein). This is not the role of the averaging operation here, however. The purpose here is to bring to bear the ergodic theorems for weak Feller Markov chains, in particular, the weak convergence of certain averaged probability measures or occupation probability measures to the invariant probability measures of the m -step version of $\{(Z_t, \theta_t^\alpha)\}$. (For the details see Section 4.3, where the proofs of the results of this subsection will be given.) It can also be seen that for a sequence $\{\beta_t\}$ with $\beta_t \in [0, 1], \beta_t \rightarrow 0$ as $t \rightarrow \infty$, if we drop a fraction β_t of the terms in (22) when averaging the θ 's at each time t , the resulting differences in the averaged iterates $\bar{\theta}_t^\alpha$ are asymptotically negligible. Therefore, although our results below will be stated for (22), they apply to a variety of averaging schemes.

Recall that $N_\delta(\theta^*)$ denotes the closed δ -neighborhood of θ^* . In what follows, $N'_\delta(\theta^*)$ denotes the open δ -neighborhood of θ^* , i.e., the open ball around θ^* with radius δ . We write $[N_\delta(\theta^*)]^m$ or $[N'_\delta(\theta^*)]^m$ for the Cartesian product of m copies of $N_\delta(\theta^*)$ or $N'_\delta(\theta^*)$. Recall also that r_B is the radius of the constraint set B .

Theorem 8 *In the setting of Theorem 5, let $\{\theta_t^c\}$ be generated by the algorithm (11) with constant stepsize $\alpha > 0$, and let $\{\theta_t^c\}$ be the corresponding averaged sequence. Then the following hold for any $\delta > 0$ and $m \geq 1$:*

- (i) $\liminf_{\alpha \rightarrow 0} \inf_{\mu \in \mathcal{M}_{\alpha}^m} \mu([N_{\delta}(\theta^*)]^m) = 1$, and more strongly, with $m_{\alpha} = \lfloor \frac{m}{\alpha} \rfloor$,
- $$\liminf_{\alpha \rightarrow 0} \inf_{\mu \in \mathcal{M}_{m_{\alpha}}^m} \mu([N_{\delta}(\theta^*)]^{m_{\alpha}}) = 1.$$
- (ii) *For each stepsize α and any initial condition of (e_0, F_0, θ_0^c) ,*

$$\limsup_{k \rightarrow \infty} \mathbb{E} \left[\sup_{k \leq k < k+m} |\bar{\theta}_k^c - \theta^*| \right] \leq \delta \kappa_{\alpha, m} + 2r_B (1 - \kappa_{\alpha, m}),$$

where $\kappa_{\alpha, m} = \inf_{\mu \in \mathcal{M}_{\alpha}^m} \mu([N'_{\delta}(\theta^*)]^m)$.

Note that in part (ii) above, $\kappa_{\alpha, m} \rightarrow 1$ as $\alpha \rightarrow 0$ by part (i). Note also that for $m = 1$, the conclusions from the preceding theorem take the simplest form:

$$\liminf_{\alpha \rightarrow 0} \inf_{\mu \in \mathcal{M}_{\alpha}} \mu(N_{\delta}(\theta^*)) = 1,$$

$$\limsup_{t \rightarrow \infty} \mathbb{E} [|\bar{\theta}_t^c - \theta^*|] \leq \delta \kappa_{\alpha} + 2r_B (1 - \kappa_{\alpha}), \quad \text{for } \kappa_{\alpha} = \inf_{\mu \in \mathcal{M}_{\alpha}} \mu(N'_{\delta}(\theta^*)).$$

The conclusions for $m > 1$ are, however, much stronger. They also suggest that in practice, instead of simply choosing the last iterate of the algorithm as its final output at the end of its run, one can base that choice on the behavior of multiple consecutive $\bar{\theta}_k^c$ during the run.

For the two variant algorithms (19) and (20), we have a similar result given in Theorem 9 below. Here the neighborhood of $(\theta^*, \dots, \theta^*)$ around which the masses of the invariant probability measures are concentrated, depends not only on the stepsize α but also on the biases of these algorithms. The proofs of Theorems 8-9 are given in Section 4.3.2.

Theorem 9 *In the setting of Theorem 5, let $\{\theta_t^c\}$ be generated instead by the algorithm (19) or (20), with constant stepsize $\alpha > 0$ and with a bounded Lipschitz continuous function ψ_K satisfying (18). Let $\{\theta_t^c\}$ be the corresponding averaged sequence. Then the following hold:*

- (i) *For any given $\delta > 0$, there exists $K_{\delta} > 0$ such that for all $K \geq K_{\delta}$,*
- $$\liminf_{\alpha \rightarrow 0} \inf_{\mu \in \mathcal{M}_{\alpha}^m} \mu([N_{\delta}(\theta^*)]^m) = 1, \quad \forall m \geq 1,$$
- and more strongly, with $m_{\alpha} = \lfloor \frac{m}{\alpha} \rfloor$,
- $$\liminf_{\alpha \rightarrow 0} \inf_{\mu \in \mathcal{M}_{m_{\alpha}}^m} \mu([N_{\delta}(\theta^*)]^{m_{\alpha}}) = 1, \quad \forall m \geq 1.$$

- (ii) *Regardless of the choice of K , given any $\delta > 0$, $m \geq 1$ and stepsize α , for each initial condition of (e_0, F_0, θ_0^c) ,*

$$\limsup_{k \rightarrow \infty} \mathbb{E} \left[\sup_{k \leq k < k+m} |\bar{\theta}_k^c - \theta^*| \right] \leq \delta \kappa_{\alpha, m} + 2r_B (1 - \kappa_{\alpha, m}),$$

where $\kappa_{\alpha, m} = \inf_{\mu \in \mathcal{M}_{\alpha}^m} \mu([N'_{\delta}(\theta^*)]^m)$.

Finally, we consider a simple modification of the preceding algorithms, for which the conclusions of Theorems 8(ii) and 9(ii) can be strengthened. This is our motivation for introducing the modification, but we shall postpone the discussion till Remark 2 at the end of this subsection.

For any of the algorithms (11), (19) or (20), if the original recursion under a constant stepsize α can be written as

$$\theta_{t+1}^c = \Pi_B(\theta_t^c + \alpha Y_t^c),$$

we now modify this recursion formula by adding a perturbation term $\alpha \Delta_{\theta_t^c}$ as follows. Let

$$\theta_{t+1}^c = \Pi_B(\theta_t^c + \alpha Y_t^c + \alpha \Delta_{\theta_t^c}^c), \quad (23)$$

where for each $\alpha > 0$, $\Delta_{\theta_t^c}^c, t \geq 0$, are \mathbb{R}^n -valued random variables such that¹³

- (i) they are independent of each other and also independent of the process $\{Z_t\}$;
- (ii) they are identically distributed with zero mean and finite variance, where the variance can be bounded uniformly for all α ; and

- (iii) they have a positive continuous density function with respect to the Lebesgue measure. Below we refer to (23) as the perturbed version of the algorithm (11), (19) or (20).

Theorem 10 *In the setting of Theorem 5, let $\{\theta_t^c\}$ be generated instead by the perturbed version (23) of the algorithm (11) for a constant stepsize $\alpha > 0$, and let $\{\bar{\theta}_t^c\}$ be the corresponding averaged sequence. Then the conclusions of Theorems 5 and 8 hold. Furthermore, let the stepsize α be given. Then the Markov chain $\{(Z_t, \bar{\theta}_t^c)\}$ has a unique invariant probability measure $\bar{\mu}_{\alpha}$, and for any $\delta > 0$, $m \geq 1$ and initial condition of (e_0, F_0, θ_0^c) , almost surely,*

$$\liminf_{t \rightarrow \infty} \frac{1}{t} \sum_{k=0}^{t-1} \mathbb{1} \left(\sup_{k \leq j < k+m} |\theta_j^c - \theta^*| < \delta \right) \geq \bar{\mu}_{\alpha}^{(m)}([N'_{\delta}(\theta^*)]^m)$$

and

$$\limsup_{t \rightarrow \infty} |\bar{\theta}_t^c - \theta^*| \leq \delta \kappa_{\alpha} + 2r_B (1 - \kappa_{\alpha}), \quad \text{with } \kappa_{\alpha} = \bar{\mu}_{\alpha}(N'_{\delta}(\theta^*)),$$

where $\bar{\mu}_{\alpha}^{(m)}$ is the unique element in $\mathcal{M}_{\alpha}^{(m)}$, and $\bar{\mu}_{\alpha}$ is the marginal of $\bar{\mu}_{\alpha}$ on B .

Theorem 11 *In the setting of Theorem 5, let $\{\theta_t^c\}$ be generated instead by the perturbed version (23) of the algorithm (19) or (20), with a constant stepsize $\alpha > 0$ and with a bounded Lipschitz continuous function ψ_K satisfying (18). Let $\{\bar{\theta}_t^c\}$ be the corresponding averaged sequence. Then the conclusions of Theorems 7 and 9 hold. Furthermore, for any given stepsize α , the conclusions of the second part of Theorem 10 also hold.*

Note that in the second part of Theorem 10, both $\bar{\mu}_{\alpha}^{(m)}([N'_{\delta}(\theta^*)]^m)$ and κ_{α} approach 1 as $\alpha \rightarrow 0$, since by the first part of the theorem, the conclusions of Theorem 8 hold. For the second part of Theorem 11, the same is true provided that K is sufficiently large (so that $N_{\delta}(LB) \subset N_{\delta}(\theta^*)$ where LB is the limit set of the ODE associated with the algorithm),

13. We adopt these conditions for simplicity. They are not the weakest possible for our purpose, and our proof techniques can be applied to other types of perturbations as well. For related discussions, see Remark 2 at the end of this section, as well as Remark 3 and the discussion before Prop. 8 in Section 4.3.3.

and this can be seen from the conclusions of Theorem 9(i), which holds for the perturbed version (23) of the two variant algorithms, as the first part of Theorem 11 says. The proofs of Theorems 10-11 are given in Section 4.3.3.

Remark 2 (on the role of perturbation) At first sight it may seem counter-productive to add noise to the θ -iterates in the algorithm (23). Our motivation for such random perturbations of the θ -iterates is that this can ensure that the Markov chain $\{(Z_t, \theta_t^*)\}$ has a unique invariant probability measure (see Prop. 9, Section 4.3.3). The uniqueness allows us to invoke a result of Meyn (1989) on the convergence of the occupation probability measures of a weak Feller Markov chain, so that we can bound the deviation of the averaged iterates from θ^* not only in an expected sense as before, but also for almost all sample paths under each initial condition, as in the second part of Theorems 10-11. For the unperturbed algorithms, we can only prove such pathwise bounds on $\limsup_{t \rightarrow \infty} \|\bar{\theta}_t^* - \theta^*\|$ for a subset of the initial conditions of (Z_0, θ_0^*) . A more detailed discussion of this is given in Remark 3 at the end of Section 4.3.3, after the proofs of the preceding theorems.

Regarding other effects of the perturbation, intuitively, larger noise terms may help the Markov chain “mix” faster, but they can also result in less probability mass $\bar{\mu}_\alpha(N_\delta^*(\theta^*))$ around θ^* than in the case without perturbation. What is a suitable amount of noise to add to achieve a desired balance? We do not yet have an answer. It seems reasonable to us to let the magnitude of the variance of the perturbation terms $\Delta_{\theta,t}^\alpha$ be approximately $\alpha^{2\epsilon}$ for some $\epsilon \in (0, 1]$, so that a typical perturbation $\alpha \Delta_{\theta,t}^\alpha$ is at a smaller scale relative to the “signal part” αY_t^α in an iteration. Further investigation is needed. ■

4. Proofs for Section 3

We now prove the theorems given in the preceding section. We shall use KY as an abbreviation for the book (Kushner and Yin, 2003), which we will refer to frequently below.

4.1 Proofs for Theorems 4 and 5

In this subsection we prove Theorems 4 and 5 (Section 3.1) on convergence properties of the constrained ETD(λ) algorithm (11). We will apply two theorems from (KY), Theorems 8.2.2 and 8.2.3, which concern weak convergence of stochastic approximation algorithms for constant and diminishing stepsize, respectively. This requires us to show that the conditions of those theorems are satisfied by our algorithm. The major conditions concern the uniform integrability, tightness, and convergence in mean of certain sequences of random variables involved in the algorithm. Our proofs will rely on many properties of the ETD iterates that we have established in (2015a) when analyzing the almost sure convergence of the algorithm.

4.1.1 CONDITIONS TO VERIFY

We need some definitions and notation, before describing the conditions required. For some index set \mathcal{K} , let $\{X_k\}_{k \in \mathcal{K}}$ be a set of random variables taking values in a metric space \mathbf{X} (in our context \mathbf{X} will be \mathbb{R}^m or Ξ). The set $\{X_k\}_{k \in \mathcal{K}}$ is said to be *tight* or *bounded in probability*, if there exists for each $\delta > 0$ a compact set $D_\delta \subset \mathbf{X}$ such that

$$\inf_{k \in \mathcal{K}} \mathbf{P}(X_k \in D_\delta) \geq 1 - \delta.$$

For \mathbb{R}^m -valued X_k , the set $\{X_k\}_{k \in \mathcal{K}}$ is said to be *uniformly integrable* (u.i.) if

$$\lim_{a \rightarrow \infty} \sup_{k \in \mathcal{K}} \mathbb{E}[\|X_k\| \mathbb{1}(\|X_k\| \geq a)] = 0.$$

To analyze the constrained ETD(λ) algorithm (11), which is given by

$$\theta_{t+1} = \Pi_B(\theta_t + \alpha_t Y_t), \quad \text{where } Y_t := e_t \cdot \rho_t (R_t + \gamma_{t+1} \phi(S_{t+1})^\top \theta_t - \phi(S_t)^\top \theta_t),$$

let E_t denote expectation conditioned on \mathcal{F}_t , the sigma-algebra generated by $\theta_m, \xi_m, m \leq t$, where we recall $\xi_m = (\epsilon_m, F_m, S_m, A_m, S_{m+1})$ and its space $\mathbb{R}^{m+1} \times \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ is denoted by Ξ . By writing $Y_t = E_t[Y_t] + (Y_t - E_t[Y_t])$, we have the equivalent form of (11) given in (15):

$$\theta_{t+1} = \Pi_B(\theta_t + \alpha_t h(\theta_t, \xi_t) + \alpha_t e_t \cdot \tilde{\omega}_{t+1}).$$

In other words, $h(\theta_t, \xi_t) = E_t[Y_t]$ and $e_t \cdot \tilde{\omega}_{t+1} = Y_t - E_t[Y_t]$, a noise term that satisfies $E_t[e_t \cdot \tilde{\omega}_{t+1}] = 0$.

This algorithm belongs to the class of stochastic approximation algorithms with “exogenous noises” studied in the book (KY) – the term “exogenous noises” reflects the fact that the evolution of $\{\xi_t\}$ is not driven by the θ -iterates. Theorems 4 and 5 will follow as special cases from Theorems 8.2.3 and 8.2.2 of (KY, Chap. 8), respectively, if we can show that the algorithm (11) satisfies the following conditions.

Conditions for the case of diminishing stepsize:

- (i) The sequence $\{Y_t\} = \{h(\theta_t, \xi_t) + e_t \cdot \tilde{\omega}_{t+1}\}$ is u.i. (This corresponds to the condition A.8.2.1 in KY.)
- (ii) The function $h(\theta, \xi)$ is continuous in θ uniformly in $\xi \in D$, for each compact set $D \subset \Xi$. (This corresponds to the condition A.8.2.3 in KY.)
- (iii) The sequence $\{\xi_t\}$ is tight. (This corresponds to the condition A.8.2.4 in KY.)
- (iv) The sequence $\{h(\theta_t, \xi_t)\}$ is u.i., and so is $\{h(\theta, \xi_t)\}$ for each fixed $\theta \in B$. (This corresponds to the condition A.8.2.5 in KY.)
- (v) There is a continuous function $\bar{h}(\cdot)$ such that for each $\theta \in B$ and each compact set $D \subset \Xi$,

$$\lim_{k \rightarrow \infty, t \rightarrow \infty} \frac{1}{k} \sum_{m=t}^{t+k-1} E_t [h(\theta, \xi_m) - \bar{h}(\theta)] \mathbb{1}(\xi_t \in D) = 0 \quad \text{in mean,}$$

where k and t are taken to ∞ in any way possible. In other words, if we denote the average on the left-hand side by $\bar{X}_{k,t}$, then the requirement “ $\lim_{k \rightarrow \infty, t \rightarrow \infty} \bar{X}_{k,t} = 0$ in mean” means that along any subsequences $k_j \rightarrow \infty, t_j \rightarrow \infty$, we must have $\lim_{j \rightarrow \infty} E[\|\bar{X}_{k_j, t_j}\|] = 0$. (This condition corresponds to the condition A.8.2.7 in KY.)

For the case of constant stepsize, we consider the iterates that could be generated by the algorithm for all stepsizes. To distinguish between the iterates associated with different stepsizes, in the conditions below, the superscript α is attached to the variables involved in the algorithm with stepsize α , and similarly, the conditional expectation E_t is denoted by E_t^α instead.

Conditions for the case of constant stepsize:

In addition to the condition (ii) above (which corresponds to the condition A.8.1.6 in KY for the case of constant stepsize), the following conditions are required.

- (i') The set $\{Y_t^\alpha \mid t \geq 0, \alpha > 0\} := \{h(\theta_t^\alpha, \xi_t^\alpha) + e_t^\alpha \cdot \tilde{\omega}_{t+1}^\alpha \mid t \geq 0, \alpha > 0\}$ is u.i. (This corresponds to the condition A.8.1.1 in KY.)
- (iii') The set $\{\xi_t^\alpha \mid t \geq 0, \alpha > 0\}$ is tight. (This corresponds to the condition A.8.1.7 in KY.)
- (iv') The set $\{h(\theta_t^\alpha, \xi_t^\alpha) \mid t \geq 0, \alpha > 0\}$ is u.i., in addition to the uniform integrability of $\{h(\theta_t, \xi_t^\alpha) \mid t \geq 0, \alpha > 0\}$ for each $\theta \in B$. (This corresponds to the condition A.8.1.8 in KY.)
- (v') There is a continuous function $\bar{h}(\cdot)$ such that for each $\theta \in B$ and each compact set $D \subset \Xi$,

$$\lim_{k \rightarrow \infty, t \rightarrow \infty, \alpha \rightarrow 0} \frac{1}{k} \sum_{m=d}^{t+k-1} \mathbb{E}_t^\alpha [h(\theta, \xi_m^\alpha) - \bar{h}(\theta)] \mathbb{1}(\xi_t^\alpha \in D) = 0 \quad \text{in mean,}$$

where α is taken to 0 and k, t are taken to ∞ in any way possible. (This condition corresponds to the condition A.8.1.9 in KY, and it is in fact stronger than the latter condition but is satisfied by our algorithms as we will show.)

The preceding conditions allow ξ_t^α and θ_t^α to be generated under different initial conditions for different α . While we will need this generality later in Section 4.3, here we will focus on a common initial condition for all stepsizes, for simplicity. Then, the preceding conditions for the constant-stepsize case are essentially the same as those for the diminishing stepsize case, because except for the θ -iterates, all the other variables (such as ξ_t and $\tilde{\omega}_t$) involved in the algorithm have identical probability distributions for all stepsizes α and are not affected by the θ -iterates. For this reason, in the proofs below, except for the θ -iterates, we simply omit the superscript α for other variables in the case of constant stepsize, and to verify the two sets of conditions above, we shall treat the case of diminishing stepsize and the case of constant stepsize simultaneously.

As mentioned in Section 2.4, these conditions are to ensure that the projected ODE (12), $\dot{x} = \bar{h}(x) + z, z \in -N_B(x)$, is the mean ODE for the algorithm (11) and reflects its average dynamics. Among the proofs for these conditions given next, the proof for the convergence in mean condition (v) and (v') will be the most involved.

4.1.2 PROOFS

The condition (ii) is clearly satisfied. In what follows, we prove that the rest of the conditions are satisfied as well. We start with the tightness conditions (iii) and (iii'), as they are immediately implied by a property of the trace iterates we already know. We then tackle the uniform integrability conditions (i), (i'), (iv) and (iv'), before we address the convergence in mean required in (v) and (v'). The proofs build upon several key properties of the ETD iterates we have established in (2015a) and recounted in Section 2.4 and Appendix A.

First, we show that the tightness conditions (iii) and (iii') are satisfied. This is implied by the following property of traces: $\sup_{t \geq 0} \mathbb{E}[\|(e_t, F_t)\|] < \infty$ for any given initial condition (e_0, F_0) (see Prop. 11, Appendix A).

Proposition 1 *Under Assumption 1, for each given initial $(e_0, F_0) \in \mathbb{R}^{n+1}$, $\{(e_t, F_t)\}$ is tight and hence $\{\xi_t\}$ is tight.*

Proof By Prop. 11, $c := \sup_{t \geq 0} \mathbb{E}[\|(e_t, F_t)\|] < \infty$. Then, by the Markov inequality, for $a > 0$, $\sup_{t \geq 0} \mathbf{P}[\|(e_t, F_t)\| \geq a] \leq c/a \rightarrow 0$ as $a \rightarrow \infty$. This implies that $\{(e_t, F_t)\}$ is tight. Since the space $\mathcal{S} \times \mathcal{A} \times \mathcal{S}$ is finite and $\xi_t = (e_t, F_t, S_t, A_t, S_{t+1})$, $\{\xi_t\}$ is also tight. ■

We now handle the uniform integrability conditions (i), (i'), (iv) and (iv'). The uniform integrability of the trace sequence $\{e_t\}$, as we will prove, is important here.

Proposition 2 *Under Assumption 1, for each given initial $(e_0, F_0) \in \mathbb{R}^{n+1}$, the following sets of random variables are u.i.:*

- (i) $\{e_t\}$;
- (ii) $\{h(\theta, \xi_t)\}$ for each fixed $\theta \in B$;
- (iii) $\{h(\theta_t, \xi_t)\}$ in the case of diminishing stepsize; and $\{h(\theta_t^\alpha, \xi_t) \mid t \geq 0, \alpha > 0\}$ in the case of constant stepsize;
- (iv) $\{h(\theta_t, \xi_t) + e_t \tilde{\omega}_{t+1}\}$ in the case of diminishing stepsize; and $\{h(\theta_t^\alpha, \xi_t) + e_t \tilde{\omega}_{t+1} \mid t \geq 0, \alpha > 0\}$ in the case of constant stepsize.

The proof of Prop. 2 will use facts about u.i. sequences of random variables given in the lemma below. This lemma basically follows from the definition of uniform integrability. We therefore omit its proof, which can be found in the arXiv version of this paper (Yu, 2015b).

Lemma 2 *Let $X_k, Y_k, k \in \mathcal{K}$ (some index set) be real-valued random variables with X_k and Y_k defined on a common probability space for each k .*

- (i) *If $\{X_k\}_{k \in \mathcal{K}}, \{Y_k\}_{k \in \mathcal{K}}$ are u.i., then $\{X_k + Y_k\}_{k \in \mathcal{K}}$ is u.i.*
- (ii) *If $\{X_k\}_{k \in \mathcal{K}}$ is u.i. and for all k , $|Y_k| \leq |X_k|$ a.s., then $\{Y_k\}_{k \in \mathcal{K}}$ is u.i.*
- (iii) *If $\{X_k\}_{k \in \mathcal{K}}, \{Y_k\}_{k \in \mathcal{K}}$ are u.i. and for some $c \geq 0$, $\mathbb{E}[\|Y_k\| \mid X_k] \leq c$ a.s. for all k , then $\{X_k Y_k\}_{k \in \mathcal{K}}$ is u.i.*

We now proceed to prove Prop. 2. The proof will involve auxiliary variables, which we call truncated traces. They are defined similarly to the trace iterates (e_t, F_t) , but instead of depending on all the past states and actions, they only depend on a certain number of the most recent states and actions. Specifically, for each integer $K \geq 1$, we define truncated traces $(\tilde{e}_{t,K}, \tilde{F}_{t,K})$ as follows:

$$(\tilde{e}_{t,K}, \tilde{F}_{t,K}) = (e_t, F_t) \quad \text{for } t \leq K,$$

and for $t \geq K + 1$, with the shorthand $\beta_t := \rho_{t-1} \gamma^t \lambda_t$,

$$\tilde{F}_{t,K} = \sum_{k=d-K}^t i(S_k) \cdot (\rho_k \gamma_{k+1} \cdots \rho_{t-1} \gamma^t), \quad (24)$$

$$\tilde{M}_{t,K} = \lambda_t i(S_t) + (1 - \lambda_t) \tilde{F}_{t,K}, \quad (25)$$

$$\tilde{e}_{t,K} = \sum_{k=d-K}^t \tilde{M}_{k,K} \cdot \phi(S_k) \cdot (\beta_{k+1} \cdots \beta_t). \quad (26)$$

Note that when $t \geq 2K + 1$, the traces $(\tilde{e}_{t,K}, \tilde{F}_{t,K})$ no longer depend on the initial (e_0, F_0) ; being functions of the states and actions between time $t - 2K$ and t only, they lie in a bounded set determined by K , since the state and action spaces are finite. For $t = 0, \dots, 2K$, $(\tilde{e}_{t,K}, \tilde{F}_{t,K})$ also lie in a bounded set, which is determined by K and the initial (e_0, F_0) . We will use these bounded truncated traces to approximate the original traces $\{(e_t, F_t)\}$ in the analysis.

An important approximation property, given in Prop. 13 (Appendix A), is that for each K and any initial (e_0, F_0) from a given bounded set E ,

$$\sup_{t \geq 0} \mathbb{E} \left[\left\| (e_t, F_t) - (\tilde{e}_{t,K}, \tilde{F}_{t,K}) \right\| \right] \leq L_K,$$

where L_K is a finite constant that depends on K and E and decreases monotonically to 0 as K increases:

$$L_K \downarrow 0 \quad \text{as } K \rightarrow \infty.$$

We will use this property in the following analysis.

Proof of Prop. 2 First, we prove $\{e_t\}$ is u.i. We then use this to show the uniform integrability of the other sets required in parts (ii)-(iv).

(i) To prove $\{e_t\}$ is u.i., we shall exploit its relation with the truncated traces, $\tilde{e}_{t,K}$, $t \geq 0$ for integers $K \geq 1$. Note that since the state and action spaces are finite, the truncated traces $\{\tilde{e}_{t,K}\}$ lie in a bounded set (this set depends on K and the initial (e_0, F_0)), so there exists a constant a_K such that $\|\tilde{e}_{t,K}\| \leq a_K$ for all t . This fact will greatly simplify the analysis. Let us first fix K and consider $a \geq a_K$. Denote $\bar{a} = a - a_K \geq 0$. Then

$$\begin{aligned} \mathbb{E} \|e_t\| \mathbb{1}(\|e_t\| \geq a) &\leq \mathbb{E} \|e_t\| \mathbb{1}(\|e_t - \tilde{e}_{t,K}\| \geq \bar{a}) \\ &\leq \mathbb{E} \|e_t - \tilde{e}_{t,K}\| \mathbb{1}(\|e_t - \tilde{e}_{t,K}\| \geq \bar{a}) + \mathbb{E} \|\tilde{e}_{t,K}\| \mathbb{1}(\|e_t - \tilde{e}_{t,K}\| \geq \bar{a}) \\ &\leq \mathbb{E} \|e_t - \tilde{e}_{t,K}\| \mathbb{1}(\|e_t - \tilde{e}_{t,K}\| \geq \bar{a}) + a_K \mathbb{1}(\|e_t - \tilde{e}_{t,K}\| \geq \bar{a}). \end{aligned} \quad (27)$$

For the second term on the right-hand side, we can bound its expectation by

$$\mathbb{E} [a_K \mathbb{1}(\|e_t - \tilde{e}_{t,K}\| \geq \bar{a})] = a_K \mathbf{P}(\|e_t - \tilde{e}_{t,K}\| \geq \bar{a}) \leq a_K \cdot L_K / \bar{a}, \quad \forall t, \quad (28)$$

where in the last inequality L_K is a constant that depends on K (and the initial (e_0, F_0)) and has the property that $L_K \downarrow 0$ as $K \rightarrow \infty$, and this inequality is derived by bounding the Markov inequality $\mathbf{P}(\|e_t - \tilde{e}_{t,K}\| \geq \bar{a}) \leq \mathbb{E} \|e_t - \tilde{e}_{t,K}\| / \bar{a}$ with Prop. 13, which bounds $\sup_{t \geq 0} \mathbb{E} \|e_t - \tilde{e}_{t,K}\|$ by L_K . Similarly, for the first term on the right-hand side of (27), using Prop. 13, we can bound its expectation by L_K :

$$\mathbb{E} [\|e_t - \tilde{e}_{t,K}\| \mathbb{1}(\|e_t - \tilde{e}_{t,K}\| \geq \bar{a})] \leq \mathbb{E} \|e_t - \tilde{e}_{t,K}\| \leq L_K, \quad \forall t. \quad (29)$$

From (27)-(29) it follows that

$$\sup_{t \geq 0} \mathbb{E} [\|e_t\| \mathbb{1}(\|e_t\| \geq a)] \leq L_K + a_K \cdot L_K / (a - a_K),$$

so for fixed K , by taking $a \rightarrow \infty$, we obtain

$$\limsup_{a \rightarrow \infty} \sup_{t \geq 0} \mathbb{E} [\|e_t\| \mathbb{1}(\|e_t\| \geq a)] \leq L_K.$$

Since $L_K \downarrow 0$ as $K \rightarrow \infty$ (Prop. 13), this implies $\lim_{a \rightarrow \infty} \sup_{t \geq 0} \mathbb{E} [\|e_t\| \mathbb{1}(\|e_t\| \geq a)] = 0$, which proves the uniform integrability of $\{e_t\}$.

(ii) We now prove for each θ , $\{h(\theta, \xi_t)\}$ is u.i. Since the state and action spaces are finite and θ is given, using the expression of $h(\theta, \xi_t)$, we can bound it as $\|h(\theta, \xi_t)\| \leq L \|e_t\|$ for some constant L . As just proved, $\{e_t\}$ is u.i. (equivalently $\{\|e_t\|\}$ is u.i.) and thus $\{L \|e_t\|\}$ is u.i., so by Lemma 2(ii), $\{h(\theta, \xi_t)\}$ is u.i. (since this is by definition equivalent to $\{\|h(\theta, \xi_t)\|\}$ being u.i., which is true by Lemma 2(ii)).

(iii) The uniform integrability of $\{h(\theta_t, \xi_t)\}$ in the case of diminishing stepsize or $\{h(\theta_t^\alpha, \xi_t)\}$ $t \geq 0, \alpha > 0$ in the case of constant stepsize follows from the same argument given for (ii) above, because θ_t or θ_t^α for all $t \geq 0$ and $\alpha > 0$ lie in the bounded set B by the definition of the constrained ETD(λ) algorithm.

(iv) Consider first the case of diminishing stepsize. We prove that $\{h(\theta_t, \xi_t) + e_t \tilde{\omega}_{t+1}\}$ is u.i. (recall $\tilde{\omega}_{t+1} = \rho_t (R_t - r(S_t, A_t, S_{t+1}))$ is the noise part of the observed reward). Since we already showed that $\{h(\theta_t, \xi_t)\}$ is u.i., by Lemma 2(i), it is sufficient to prove that $\{e_t \tilde{\omega}_{t+1}\}$ is u.i. Now $\{e_t\}$ is u.i. by part (i). Since the random rewards R_t in our model have bounded variances, the noise variables $\tilde{\omega}_{t+1}, t \geq 0$, also have bounded variances. This implies that $\{\tilde{\omega}_{t+1}\}$ is u.i. (Billingsley, 1968, p. 32) and that $\mathbb{E}[\|\tilde{\omega}_{t+1}\| | e_t] < c$ for some constant c (independent of t). It then follows from Lemma 2(iii) that $\{e_t \tilde{\omega}_{t+1}\}$ is u.i., and hence $\{h(\theta_t, \xi_t) + e_t \tilde{\omega}_{t+1}\}$ is u.i.

Similarly, in the case of constant stepsize, it follows from Lemma 2(i) that the set $\{h(\theta^\alpha, \xi_t) + e_t \tilde{\omega}_{t+1} | t \geq 0, \alpha > 0\}$ is u.i., because $\{h(\theta^\alpha, \xi_t) | t \geq 0, \alpha > 0\}$ is u.i. by part (iii) proved earlier and $\{e_t \tilde{\omega}_{t+1}\}$ is u.i. as we just proved. ■

Finally, we handle the conditions (v) and (v') stated in Section 4.1.1. The two conditions are the same condition in the case here, because they concern each fixed θ , whereas $\{\xi_t\}$ is not affected by the stepsize and the θ -iterates. So we can focus just on the condition (v) in presenting the proof, for notational simplicity. For the algorithm (11), the continuous function \bar{h} required in the condition is the function $\bar{h}(\theta) = C\theta + b$ associated with the desired mean ODE (12). We now prove the required convergence in mean by using the properties of trace iterates and the convergence results given in Theorem 3 and Corollary 1 (Section 2.4).

Proposition 3 *Let Assumption 1 hold. For each $\theta \in B$ and each compact set $D \subset \Xi$,*

$$\lim_{k \rightarrow \infty} \frac{1}{k} \sum_{m=i}^{i+k-1} \mathbb{E}_t [h(\theta, \xi_m) - \bar{h}(\theta)] \mathbb{1}(\xi_t \in D) = 0 \quad \text{in mean.}$$

Proof Denote $X_{k,t} = \frac{1}{k} \sum_{m=t}^{t+k-1} (h(\theta, \xi_m) - \bar{h}(\theta)) \mathbb{1}(\xi_t \in D)$. Since $\mathbb{E}[\|X_{k,t}\|] \leq \mathbb{E}[\|X_{k,t}\|]$, to prove $\lim_{k,t} \mathbb{E}[\|X_{k,t}\|] = 0$ (here and in what follows we simply write " k, t " under a limit symbol for " $k \rightarrow \infty, t \rightarrow \infty$ "), it is sufficient to prove $\lim_{k,t} \mathbb{E}[\|X_{k,t}\|] = 0$, that is, to prove

$$\lim_{k,t} \frac{1}{k} \sum_{m=i}^{i+k-1} (h(\theta, \xi_m) - \bar{h}(\theta)) \mathbb{1}(\xi_t \in D) = 0 \quad \text{in mean.} \quad (30)$$

Furthermore, since $\limsup_{k,t} \mathbb{E}[\|X_{k,t}\| \mathbb{1}(\xi_t \in D)]$ is upper-bounded by

$$\sum_{(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}} \limsup_{k,t} \mathbb{E}[\|X_{k,t}\| \mathbb{1}(\xi_t \in D, (S_t, A_t, S_{t+1}) = (s, a, s'))],$$

it is sufficient in the proof to consider only those compact sets D of the form $D = E \times \{(s, a, s')\}$, for each compact set $E \subset \mathbb{R}^{n+1}$ and each $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$. Henceforth, let us fix a compact set E together with a triplet (s, a, s') as the set D under consideration, and for this set D , we proceed to prove (30).

To show (30), what we need to show is that for two arbitrary subsequences of integers $k_j \rightarrow \infty, t_j \rightarrow \infty$,

$$\lim_{j \rightarrow \infty} \frac{1}{k_j} \sum_{m=t_j}^{t_j+k_j-1} (h(\theta, \xi_m) - \bar{h}(\theta)) \mathbb{1}(\xi_{t_j} \in D) = 0 \quad \text{in mean.} \quad (31)$$

To this end, we first define auxiliary trace variables to decompose each difference term $h(\theta, \xi_m) - \bar{h}(\theta)$ into two difference terms as follows:

- (a) Fix a point $(\bar{e}, \bar{F}) \in E$.
- (b) For each $j \geq 1$, define a sequence of trace pairs, $(\bar{e}_m^j, \bar{F}_m^j)$, $m \geq t_j$, by using the same recursion (3)-(5) that defines the traces $\{e_t, F_t\}$, based on the same trajectory $\{(S_t, A_t)\}$, but starting at time $m = t_j$ with the initial $(\bar{e}_{t_j}^j, \bar{F}_{t_j}^j) = (\bar{e}, \bar{F})$.

Denote $\xi_m^j = (e_m^j, F_m^j, S_m, A_m, S_{m+1})$ for $m \geq t_j$; it differs from ξ_m only in the two trace components. Next, for each m , we write $h(\theta, \xi_m) - h(\theta) = (h(\theta, \xi_m^j) - h(\theta)) + (h(\theta, \xi_m) - h(\theta, \xi_m^j))$ and correspondingly, we write

$$\frac{1}{k_j} \sum_{m=t_j}^{t_j+k_j-1} (h(\theta, \xi_m) - \bar{h}(\theta)) = \frac{1}{k_j} \sum_{m=t_j}^{t_j+k_j-1} (h(\theta, \xi_m^j) - \bar{h}(\theta)) + \frac{1}{k_j} \sum_{m=t_j}^{t_j+k_j-1} (h(\theta, \xi_m) - h(\theta, \xi_m^j)).$$

We see that for (31) to hold, it is sufficient that

$$\lim_{j \rightarrow \infty} \frac{1}{k_j} \sum_{m=t_j}^{t_j+k_j-1} (h(\theta, \xi_m^j) - \bar{h}(\theta)) \mathbb{1}(\xi_{t_j} \in D) = 0 \quad \text{in mean,} \quad (32)$$

and

$$\lim_{j \rightarrow \infty} \frac{1}{k_j} \sum_{m=t_j}^{t_j+k_j-1} (h(\theta, \xi_m) - h(\theta, \xi_m^j)) \mathbb{1}(\xi_{t_j} \in D) = 0 \quad \text{in mean.} \quad (33)$$

Let us now prove these two statements.

Proof of (32): Since the set $D = E \times \{(s, a, s')\}$ and $\mathbb{1}(\xi_{t_j} \in D) \leq \mathbb{1}((S_{t_j}, A_{t_j}, S_{t_j+1}) = (s, a, s'))$, we can remove ξ_{t_j} from consideration and show instead

$$\lim_{j \rightarrow \infty} \frac{1}{k_j} \sum_{m=t_j}^{t_j+k_j-1} (h(\theta, \xi_m^j) - \bar{h}(\theta)) \mathbb{1}((S_{t_j}, A_{t_j}, S_{t_j+1}) = (s, a, s')) = 0 \quad \text{in mean,} \quad (34)$$

which will imply (32). By definition ξ_m^j , $m \geq t_j$, are generated from the initial trace pairs (\bar{e}, \bar{F}) and initial transition $(S_{t_j}, A_{t_j}, S_{t_j+1})$ at time $m = t_j$. So if $(S_{t_j}, A_{t_j}, S_{t_j+1}) = (s, a, s')$, then conditioned on this transition at t_j , the sequence $\{\xi_m^j, m \geq t_j\}$ has the same probability distribution as a sequence $\hat{\xi}_m, m \geq 0$, where $\hat{\xi}_m = (\hat{e}_m, \hat{F}_m, \hat{S}_m, \hat{A}_m, \hat{S}_{m+1})$ is generated from the initial condition $\hat{\xi}_0 = (\bar{e}, \bar{F}, s, a, s')$ by the same recursion (3)-(5) and a trajectory $\{(S_m, A_m)\}$ of states and actions under the behavior policy. This shows that

$$\begin{aligned} & \mathbb{E} \left[\left\| \frac{1}{k_j} \sum_{m=t_j}^{t_j+k_j-1} (h(\theta, \xi_m^j) - \bar{h}(\theta)) \mathbb{1}((S_{t_j}, A_{t_j}, S_{t_j+1}) = (s, a, s')) \right\| \right] \\ & \leq \mathbb{E} \left[\left\| \frac{1}{k_j} \sum_{m=0}^{k_j-1} (h(\theta, \hat{\xi}_m) - \bar{h}(\theta)) \right\| \right], \end{aligned}$$

from which we see that the convergence in mean stated by (34) holds if we have

$$\lim_{k \rightarrow \infty} \frac{1}{k} \sum_{m=0}^{k-1} (h(\theta, \hat{\xi}_m) - \bar{h}(\theta)) = 0 \quad \text{in mean.} \quad (35)$$

Now since for each θ , the function $h(\theta, \cdot)$ is Lipschitz continuous in e uniformly in the other arguments, (35) holds by Theorem 3 and its implication Corollary 1 (Section 2.4). Consequently, (34) holds, and this implies (32).

Proof of (33): Using the expression of h and the finiteness of the state and action spaces, we can bound the difference $h(\theta, \xi_m) - h(\theta, \xi_m^j)$ by

$$\|h(\theta, \xi_m) - h(\theta, \xi_m^j)\| \leq c \cdot \|e_m - e_m^j\|$$

for some constant c (independent of m, j). Let us show

$$\lim_{j \rightarrow \infty} \frac{1}{k_j} \sum_{m=t_j}^{t_j+k_j-1} \|e_m - e_m^j\| \mathbb{1}(\xi_{t_j} \in D) = 0 \quad \text{in mean,} \quad (36)$$

which will imply (33).

To prove (36), similarly to the preceding proof, we first decompose each difference term $e_m - e_m^j$ in (36) into several difference terms, by using truncated traces $\{\tilde{e}_{m,K}, \tilde{F}_{m,K}\}$ and $\{\tilde{e}_{m,K}^j, \tilde{F}_{m,K}^j \mid m \geq t_j\}$, $j \geq 1, K \geq 1$, which we now introduce. Specifically, for each $K \geq 1$, $\{\tilde{e}_{m,K}, \tilde{F}_{m,K}\}$ are defined by (24)-(26). For each $j \geq 1$ and $K \geq 1$, the truncated traces $\{\tilde{e}_{m,K}^j, \tilde{F}_{m,K}^j \mid m \geq t_j\}$ are also defined by (24)-(26), except that the initial time is set to be t_j (instead of 0) and for $m \leq t_j + K$, $(\tilde{e}_{m,K}^j, \tilde{F}_{m,K}^j)$ is set to be $(\tilde{e}_{m,K}^j, \tilde{F}_{m,K}^j)$ (instead of (e_m, F_m)).

Let us fix K for now. We bound the difference $e_m - e_m^j$ by the sum of three difference terms as

$$\|e_m - e_m^j\| \leq \|e_m - \tilde{e}_{m,K}\| + \|\tilde{e}_{m,K} - \tilde{e}_{m,K}^j\| + \|\tilde{e}_{m,K} - \tilde{e}_{m,K}^j\|, \quad (37)$$

and correspondingly, we consider the following three sequences of variables, as j tends to ∞ :

$$\frac{1}{k_j} \sum_{m=t_j}^{t_j+k_j-1} \|e_m - \tilde{e}_{m,K}\| \mathbb{1}(\xi_{t_j} \in D), \quad \frac{1}{k_j} \sum_{m=t_j}^{t_j+k_j-1} \|e_m^j - \tilde{e}_{m,K}^j\|, \quad (38)$$

and

$$\frac{1}{k_j} \sum_{m=t_j}^{t_j+k_j-1} \|\tilde{e}_{m,K} - \tilde{e}_{m,K}^j\| \mathbb{1}(\xi_{t_j} \in D). \quad (39)$$

In what follows, we will bound their expected values as $j \rightarrow \infty$ and then take $K \rightarrow \infty$; this will lead to (36).

The analyses for the two sequences in (38) are similar. Recall $D = E \times \{(s, a, s')\}$, so $\xi_{t_j} \in D$ implies $(e_{t_j}, F_{t_j}) \in E$. Since the set E is bounded, if $(e_{t_j}, F_{t_j}) \in E$, then we can use Prop. 13 (Appendix A) to bound the expectation of $\|e_m - \tilde{e}_{m,K}\|$ for $m \geq t_j$ conditioned on \mathcal{F}_{t_j} , and this gives us the bound

$$\sup_{m \geq t_j} \mathbb{E}_{t_j} [\|e_m - \tilde{e}_{m,K}\|] \mathbb{1}(\xi_{t_j} \in D) \leq L_K$$

where L_K is a constant that depends on K and the set E , and has the property that $L_K \downarrow 0$ as $K \rightarrow \infty$. From this bound, we obtain

$$\mathbb{E} \left[\frac{1}{k_j} \sum_{m=t_j}^{t_j+k_j-1} \|e_m - \tilde{e}_{m,K}\| \mathbb{1}(\xi_{t_j} \in D) \right] \leq L_K, \quad \forall j \geq 1. \quad (40)$$

Similarly, for the second sequence in (38), by Prop. 13 we have

$$\mathbb{E} \left[\frac{1}{k_j} \sum_{m=t_j}^{t_j+k_j-1} \|e_m^j - \tilde{e}_{m,K}^j\| \right] \leq L_K, \quad \forall j \geq 1, \quad (41)$$

where L_K is some constant that can be chosen to be the same constant in (40) (because the point (\tilde{e}, \tilde{F}) , which is the initial trace pair for (e_m^j, F_m^j) at time $m = t_j$, lies in E).

Consider now the sequence in (39). As discussed after the definition (24)-(26) of truncated traces, because of truncation, these traces lie in a bounded set determined by K and the set in which the initial trace pair lies. Therefore, there exists a finite constant c_K which depends on K and E , such that for all $m \geq t_j$,

$$\|\tilde{e}_{m,K}^j\| \leq c_K, \quad \text{and} \quad \|\tilde{e}_{m,K}\| \leq c_K \text{ if } (e_{t_j}, F_{t_j}) \in E.$$

Also by their definition, once m is sufficiently large, the truncated traces do not depend on the initial trace pairs; in particular,

$$\tilde{e}_{m,K}^j = \tilde{e}_{m,K}, \quad \forall m \geq t_j + 2K + 1.$$

From these two arguments it follows that

$$\mathbb{E} \left[\frac{1}{k_j} \sum_{m=t_j}^{t_j+k_j-1} \|\tilde{e}_{m,K} - \tilde{e}_{m,K}^j\| \mathbb{1}(\xi_{t_j} \in D) \right] \leq \frac{(2K+1) \cdot 2c_K}{k_j} \rightarrow 0 \text{ as } j \rightarrow \infty. \quad (42)$$

Finally, combining (40)-(42) with (37), we obtain

$$\begin{aligned} & \limsup_{j \rightarrow \infty} \mathbb{E} \left[\frac{1}{k_j} \sum_{m=t_j}^{t_j+k_j-1} \|e_m - e_m^j\| \mathbb{1}(\xi_{t_j} \in D) \right] \\ & \leq \limsup_{j \rightarrow \infty} \mathbb{E} \left[\frac{1}{k_j} \sum_{m=t_j}^{t_j+k_j-1} \|e_m - \tilde{e}_{m,K}\| \mathbb{1}(\xi_{t_j} \in D) \right] + \limsup_{j \rightarrow \infty} \mathbb{E} \left[\frac{1}{k_j} \sum_{m=t_j}^{t_j+k_j-1} \|e_m^j - \tilde{e}_{m,K}^j\| \right] \\ & \quad + \lim_{j \rightarrow \infty} \mathbb{E} \left[\frac{1}{k_j} \sum_{m=t_j}^{t_j+k_j-1} \|\tilde{e}_{m,K} - \tilde{e}_{m,K}^j\| \mathbb{1}(\xi_{t_j} \in D) \right] \\ & \leq 2L_K. \end{aligned}$$

Since $L_K \downarrow 0$ as $K \rightarrow \infty$ (Prop. 13, Appendix A), by taking $K \rightarrow \infty$, we obtain

$$\lim_{j \rightarrow \infty} \mathbb{E} \left[\frac{1}{k_j} \sum_{m=t_j}^{t_j+k_j-1} \|e_m - e_m^j\| \mathbb{1}(\xi_{t_j} \in D) \right] = 0.$$

This proves (36), which implies (33). \blacksquare

With Props. 1-3, we have furnished all the conditions required in order to apply (KY, Theorems 8.2.2, 8.2.3) to the constrained ETD algorithm (11), so we can now specialize the conclusions of these two theorems to our problem. In particular, they tell us that the projected ODE (12) is the mean ODE for (11), and furthermore, by (KY, Theorem 8.2.3) (respectively, KY, Theorem 8.2.2), the conclusions of Theorem 4 (respectively, Theorem 5) hold with $N_\delta(L_B)$ in place of $N_\delta(\theta^*)$, where $N_\delta(L_B)$ is the δ -neighborhood of the limit set L_B for the projected ODE (12). Recall that this limit set is given by

$$L_B = \bigcap_{\tau > 0} \overline{\cup_{x(0) \in B} \{x(\tau), \tau \geq \bar{\tau}\}}$$

where $x(\tau)$ is a solution of the projected ODE (12) with initial condition $x(0)$, the union is over all the solutions with initial $x(0) \in B$, and \bar{D} for a set D denotes the closure of D .

Now when the matrix C is negative definite (as implied by Assumptions 1-2) and when the radius of B exceeds the threshold given in Lemma 1, by the latter lemma, the solutions $x(\tau), \tau \in [0, \infty)$, of the ODE (12) coincide with the solutions of $\dot{x} = \bar{h}(x) = Cx + b$ for all initial $x(0) \in B$. Then from the negative definiteness of C (Theorem 1, Section 2.3), it follows that as $\tau \rightarrow \infty$, $x(\tau) \rightarrow \theta^*$ uniformly in the initial condition, and consequently, $L_B = \{\theta^*\}$.¹⁴ Thus $N_\delta(L_B) = N_\delta(\theta^*)$ and we obtain Theorems 4 and 5.

¹⁴The details for this statement are as follows. Since \bar{h} is bounded on B and the boundary reflection term $z(\cdot) \equiv 0$ under our assumptions (Lemma 1, Section 2.4), a solution $x(\cdot)$ of (12) is Lipschitz continuous on $[0, \infty)$. We calculate $V(\tau)$ for the Lyapunov function $V(\tau) = |x(\tau) - \theta^*|^2$. By the negative definiteness of the matrix C , for some $c > 0$, $x^T C x \leq -c|x|^2$ for all $x \in \mathbb{R}^n$. Then, since $\bar{h}(x) = Cx + b = C(x - \theta^*)$, we have $\dot{V}(\tau) = 2 \langle x(\tau) - \theta^*, \bar{h}(x(\tau)) \rangle \leq -2c|x(\tau) - \theta^*|^2$, and hence for any $\delta > 0$, there exists $\epsilon > 0$ such that $\dot{V}(\tau) \leq -\epsilon$ if $V(\tau) = |x(\tau) - \theta^*|^2 \geq \delta^2$. This together with the continuity of the solution $x(\cdot)$ implies that for any $x(0) \in B$, within time $\bar{\tau} = r_B^2/\epsilon$, the trajectory $x(\tau)$ must reach $N_\delta(\theta^*)$ and stay in that set thereafter. By the definition of the limit set and the arbitrariness of δ , this implies $L_B = \{\theta^*\}$.

4.2 Proofs for Theorems 6 and 7

In this subsection we prove the part of Theorems 6-7 for the first variant of the constrained ETD(λ) algorithm given in (19), Section 3.2. The proof for the second variant algorithm (20) is similar and can be found in the arXiv version of this paper (Yu, 2015b). Like in the previous subsection, we will apply (KY, Theorems 8.2.2, 8.2.3) and show that the required conditions are met. Using the properties of the mean ODE of the variant algorithm, we will then specialize the conclusions of those theorems to obtain the desired results.

Consider the first variant algorithm (19):

$$\theta_{t+1} = \Pi_B \left(\theta_t + \alpha_t \psi_K(e_t) \cdot \rho_t (R_t + \gamma_{t+1} \phi(S_{t+1})^\top \theta_t - \phi(S_t)^\top \theta_t) \right).$$

We define a function $h_K : \mathbb{R}^n \times \Xi \rightarrow \mathbb{R}^n$ by

$$h_K(\theta, \xi) = \psi_K(e) \cdot \rho(s, a) (\tau(s, a, s') + \gamma(s') \phi(s')^\top \theta - \phi(s)^\top \theta), \quad \text{for } \xi = (e, F, s, a, s'), \quad (43)$$

and write (19) equivalently as

$$\theta_{t+1} = \Pi_B \left(\theta_t + \alpha_t h_K(\theta_t, \xi_t) + \alpha_t \psi_K(e_t) \cdot \tilde{\omega}_{t+1} \right)$$

with $\tilde{\omega}_{t+1} = \rho_t (R_t - \tau(S_t, A_t, S_{t+1}))$ as before. Note that $\mathbb{E}_t[\psi_K(e) \tilde{\omega}_{t+1}] = 0$, and the algorithm is similar to the algorithm (11)—equivalently (15)—except that we have h_K and $\psi_K(e_t)$ in place of h and e_t , respectively.

We note two properties of the function h_K . They follow from direct calculations and will be useful in our analysis shortly:

- (a) Using the Lipschitz continuity of the function ψ_K (cf. Equation 18, Section 3.2), we have that for each $\theta \in \mathbb{R}^n$, there exists a finite $c > 0$ such that with $\xi = (e, F, s, a, s')$ and $\xi' = (e', F', s, a, s')$,

$$\|h_K(\theta, \xi) - h_K(\theta, \xi')\| \leq c \|e - e'\|, \quad \forall (s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}. \quad (44)$$

Thus $h_K(\theta, \cdot)$ is Lipschitz continuous in (e, F) uniformly in (s, a, s') .

- (b) Since the set B is bounded, we can bound the difference $h_K(\theta, \xi) - h(\theta, \xi)$ for all θ in B as follows. For some finite constant $c > 0$,

$$\|h_K(\theta, \xi) - h(\theta, \xi)\| \leq c \|\psi_K(e) - e\| \leq 2c \|e\| \cdot \mathbb{1}(\|e\| \geq K), \quad \forall \theta \in B, \quad (45)$$

where the last inequality follows from the property (18) of ψ_K :

$$\|\psi_K(x)\| \leq \|x\| \quad \forall x \in \mathbb{R}^n, \quad \text{and} \quad \psi_K(x) = x \quad \text{if } \|x\| \leq K.$$

We now apply (KY, Theorems 8.2.2, 8.2.3) to obtain the desired conclusions in Theorems 6-7 for the algorithm (19). This requires us to show that the conditions (i)-(v) and (i')-(v') given in Section 4.1.1 are still satisfied when we replace e_t by $\psi_K(e_t)$ and h by h_K . The uniform integrability conditions (i), (i'), (iv) and (iv') require the following sets to be u.i.: $\{h_K(\theta_t, \xi_t) + \psi_K(e_t) \cdot \tilde{\omega}_{t+1}\}$ and $\{h_K(\theta_t^c, \xi_t) + \psi_K(e_t) \cdot \tilde{\omega}_{t+1} \mid t \geq 0, \alpha > 0\}$, $\{h_K(\theta_t, \xi_t)\}$ and $\{h_K(\theta_t^c, \xi_t) \mid t \geq 0, \alpha > 0\}$, and $\{h_K(\theta, \xi)\}$ for each θ . These conditions are evidently

satisfied, in view of the boundedness of the functions ψ_K and $h_K(\theta, \cdot)$ for each θ , the boundedness of the θ -iterates due to constraints, and the finite variances of $\{\tilde{\omega}_t\}$. The condition (ii) on the continuity of $h_K(\cdot, \xi)$ uniformly in $\xi \in D$, for each compact set $D \subset \Xi$, is also clearly satisfied, whereas the condition (iii) (equivalently (iii')) on the tightness of $\{\xi_t\}$ was already verified earlier in Prop. 1 (Section 4.1.2).

What remains is the condition (v) (which is equivalent to (v')), for the same reason as discussed immediately before Prop. 3, Section 4.1.2). It requires the existence of a continuous function $h_K : \mathbb{R}^n \rightarrow \mathbb{R}^n$ such that for each $\theta \in B$ and each compact set $D \subset \Xi$,

$$\lim_{k \rightarrow \infty, t \rightarrow \infty} \frac{1}{k} \sum_{m=t}^{t+k-1} \mathbb{E}_t[h_K(\theta, \xi_m) - \bar{h}_K(\theta)] \mathbb{1}(\xi \in D) = 0 \quad \text{in mean.} \quad (46)$$

If this condition is satisfied as well, then the mean ODE for the algorithm (19) is given by

$$\dot{x} = \bar{h}_K(x) + z, \quad z \in -\mathcal{N}_B(x). \quad (47)$$

To furnish the condition (v), we first identify the function $\bar{h}_K(\theta)$ to be $\mathbb{E}_\zeta[h_K(\theta, \xi_0)]$, the expectation of $h_K(\theta, \xi_0)$ under the stationary distribution of the process $\{Z_t\}$ with the invariant probability measure ζ as its initial distribution. We relate the functions $h_K, K > 0$, to \bar{h} in the proposition below, and we will use it to characterize the bias of the algorithm (19) later.

Proposition 4 *Let Assumption 1 hold. Consider the setting of the algorithm (19), and for each $\theta \in \mathbb{R}^n$, let $\bar{h}_K(\theta) = \mathbb{E}_\zeta[h_K(\theta, \xi_0)]$. Then the function \bar{h}_K is Lipschitz continuous on \mathbb{R}^n , and*

$$\sup_{\theta \in B} \|\bar{h}_K(\theta) - \bar{h}(\theta)\| \rightarrow 0 \quad \text{as } K \rightarrow \infty. \quad (48)$$

Proof For each θ , the function $h_K(\theta, \cdot)$ is by definition bounded. Under Assumption 1, the Markov chain $\{(S_t, A_t, e_t, F_t)\}$ has a unique invariant probability measure ζ (Theorem 2, Section 2.4). Therefore, $h_K(\theta)$ is well-defined and finite. Let $c_1 = \sup_{e \in \mathbb{R}^n} \|\psi_K(e)\| < \infty$ (since ψ_K is bounded). For any θ, θ', ξ , using the definition of h_K , a direct calculation shows that for some $c_2 > 0$, $\|h_K(\theta, \xi) - h_K(\theta', \xi)\| \leq c_1 c_2 \|\theta - \theta'\|$ for all $\xi \in \Xi$, from which it follows that

$$\|\bar{h}_K(\theta) - \bar{h}_K(\theta')\| \leq \mathbb{E}_\zeta[\|h_K(\theta, \xi_0) - h_K(\theta', \xi_0)\|] \leq c_1 c_2 \|\theta - \theta'\|.$$

This shows that \bar{h}_K is Lipschitz continuous. We now prove (48). Since $\bar{h}_K(\theta) = \mathbb{E}_\zeta[h_K(\theta, \xi_0)]$ by definition and $h(\theta) = \mathbb{E}_\zeta[h(\theta, \xi_0)]$ by Corollary 1 (Section 2.4), it is sufficient to prove the following statement, which entails (48):

$$\sup_{\theta \in B} \mathbb{E}_\zeta[\|h_K(\theta, \xi_0) - h(\theta, \xi_0)\|] \rightarrow 0 \quad \text{as } K \rightarrow \infty. \quad (49)$$

By (45), for some constant $c > 0$,

$$\|h_K(\theta, \xi_0) - h(\theta, \xi_0)\| \leq 2c \|e_0\| \cdot \mathbb{1}(\|e_0\| \geq K), \quad \forall \theta \in B,$$

and therefore,

$$\sup_{\theta \in B} \mathbb{E}_\zeta \left[\left| \langle h_K(\theta, \xi_0) - h(\theta, \xi_0) \rangle \right| \right] \leq 2c \mathbb{E}_\zeta \left[\|\epsilon_0\| \cdot \mathbb{1}(\|\epsilon_0\| \geq K) \right].$$

By Theorem 3 (Section 2.4), $\mathbb{E}_\zeta \|\epsilon_0\| < \infty$ and hence $\mathbb{E}_\zeta \|\epsilon_0\| \cdot \mathbb{1}(\|\epsilon_0\| \geq K) \rightarrow 0$ as $K \rightarrow \infty$. Together with the preceding inequality, this implies (49), which in turn implies (48). \blacksquare

We now show that the convergence in mean required in (46) is satisfied.

Proposition 5 *Under Assumption 1, the conclusion of Prop. 3 (Section 4.1.2) holds in the setting of the algorithm (19), with the functions h_K and \bar{h}_K in place of h and \bar{h} , respectively.*

Proof The same arguments given in the proof of Prop. 3 apply here, with the functions h_K, \bar{h}_K in place of h, \bar{h} , respectively. Only two details are worth noting here. The proof relies on the Lipschitz continuity property of h_K given in (44). As mentioned earlier, this property implies that for each θ , with $\xi = (e, F, s, a, s')$, $h_K(\theta, \xi)$ is Lipschitz continuous in (e, F) uniformly in (s, a, s') , so we can apply Theorem 3 to conclude that (35) and hence (32) hold in this case (for h_K, \bar{h}_K instead of h, \bar{h}). The property (44) also allows us to obtain (33) in this case, by exactly the same proof given earlier. \blacksquare

Thus we have furnished all the conditions required by (KY, Theorems 8.2.2, 8.2.3). As in the case of the algorithm (11), by these two theorems, the assertions of Theorems 4-5 hold for the variant algorithm (19) with $N_\delta(L_B)$ in place of $N_\delta(\theta^*)$, where L_B is the limit set of the projected mean ODE associated with (19):

$$\dot{x} = \bar{h}_K(x) + z, \quad z \in -\mathcal{N}_B(x).$$

To finish the proof for Theorems 6-7, it is now sufficient to show that for any given $\delta > 0$, we can choose a number K_δ large enough so that $L_B \subset N_\delta(\theta^*)$ for all $K \geq K_\delta$. We prove this below, using Prop. 4. Note that the set L_B reflects the bias of the constrained algorithm (19), so what we are showing now is that this bias decreases as K increases.

Lemma 3 *Let Assumptions 1-2 hold, and let the radius of the set B exceed the threshold given in Lemma 1. Then for all K sufficiently large, given any initial condition $x(0) \in B$, a solution to the projected ODE (47) coincides with the unique solution to $\dot{x} = \bar{h}_K(x)$, with the boundary reflection term being $z(\cdot) \equiv 0$. Given $\delta > 0$, there exists K_δ such that for $K \geq K_\delta$, the limit set L_B of (47) satisfies $L_B \subset N_\delta(\theta^*)$.*

Proof Under Assumptions 1-2, the matrix C is negative definite (Theorem 1, Section 2.3), and when the radius of the set B exceeds the threshold given in Lemma 1, there exists a constant $\epsilon > 0$ such that for all boundary points x of B , $\langle x, \bar{h}(x) \rangle < -\epsilon$. At such points x , the normal cone $\mathcal{N}_B(x) = \{ax \mid a \geq 0\}$, and

$$\langle x, \bar{h}_K(x) \rangle = \langle x, \bar{h}(x) \rangle + \langle x, \bar{h}_K(x) - \bar{h}(x) \rangle < -\epsilon + \langle x, \bar{h}_K(x) - \bar{h}(x) \rangle.$$

By (48) in Prop. 4, $\langle x, \bar{h}_K(x) - \bar{h}(x) \rangle \rightarrow 0$ uniformly on B as $K \rightarrow \infty$. Thus when K is sufficiently large, at all boundary points x of B , $\langle x, \bar{h}_K(x) \rangle < 0$; i.e., $\bar{h}_K(x)$ points inside B

and the boundary reflection term $z = 0$. It then follows that for such K , given an initial condition $x(0) \in B$, a solution to (47) coincides with the unique solution to $\dot{x} = \bar{h}_K(x)$, where the uniqueness is ensured by the Lipschitz continuity of \bar{h}_K proved in Prop. 4 (cf. Borkar, 2008, Chap. 11.2).

To prove the second statement concerning the limit set of the projected ODE, let K be large enough so that the conclusion of the first part holds. Let $x(\tau), \tau \in [0, \infty)$, be the solution of (47) for a given initial $x(0) \in B$. Since \bar{h}_K is bounded on B , $x(\cdot)$ is Lipschitz continuous on $[0, \infty)$. Let $V(\tau) = |x(\tau) - \theta^*|^2$, and we calculate $\dot{V}(\tau)$. Since for all x , $\bar{h}_K(x) = Cx + b = C(x - \theta^*)$ and $x^\top Cx \leq -c|x|^2$ for some $c > 0$ by the negative definiteness of C , a direct calculation shows that

$$\begin{aligned} \dot{V}(\tau) &= 2 \langle x(\tau) - \theta^*, \bar{h}_K(x(\tau)) \rangle \\ &= 2 \langle x(\tau) - \theta^*, \bar{h}(x(\tau)) \rangle + 2 \langle x(\tau) - \theta^*, \bar{h}_K(x(\tau)) - \bar{h}(x(\tau)) \rangle \\ &\leq -2c|x(\tau) - \theta^*|^2 + 2|x(\tau) - \theta^*| \cdot |\bar{h}_K(x(\tau)) - \bar{h}(x(\tau))|. \end{aligned}$$

By (48) in Prop. 4, $\sup_{x \in B} |\bar{h}_K(x) - \bar{h}(x)| \rightarrow 0$ as $K \rightarrow \infty$. It then follows that for any $\delta > 0$, there exist $\epsilon > 0$ and $K_\delta > 0$ such that for all $K \geq K_\delta$, $\dot{V}(\tau) \leq -\epsilon$ if $V(\tau) = |x(\tau) - \theta^*|^2 \geq \delta^2$. This together with the continuity of the solution $x(\cdot)$ shows that for any $x(0) \in B$, within time $\bar{\tau} = r_B^2/\epsilon$ (where r_B is the radius of B), the trajectory $x(\tau)$ must reach $N_\delta(\theta^*)$ and stay in that set thereafter. Consequently, for all $K \geq K_\delta$, the limit set $L_B = \bigcap_{\tau \geq 0} \bigcup_{x(0) \in B} \{x(\tau), \tau \geq \bar{\tau}\} \subset N_\delta(\theta^*)$. \blacksquare

This completes the proofs of Theorems 6 and 7 for the first variant algorithm (19).

4.3 Further Analysis of the Constant-stepsize Case

We now consider again the case of constant stepsize, and prove Theorems 8-11 given in Section 3.3. The proofs will be based on combining the results we obtained earlier by using stochastic approximation theory, with the ergodic theorems of weak Feller Markov chains. As before the proofs will also rely on the key properties of the ETD iterates.

4.3.1 WEAK FELLER MARKOV CHAINS

We shall focus on Markov chains on complete separable metric spaces. For such a Markov chain $\{X_t\}$ with state space \mathbf{X} , let $P(\cdot, \cdot)$ denote its transition kernel, that is, $P : \mathbf{X} \times \mathcal{B}(\mathbf{X}) \rightarrow [0, 1]$,

$$P(x, D) = \mathbf{P}_x(X_1 \in D), \quad \forall x \in \mathbf{X}, D \in \mathcal{B}(\mathbf{X}),$$

where $\mathcal{B}(\mathbf{X})$ denotes the Borel sigma-algebra on \mathbf{X} , and \mathbf{P}_x denotes the probability distribution of $\{X_t\}$ conditioned on $X_0 = x$. Multiple-step transition kernels will also be needed. For $t \geq 1$, the t -step transition kernel $P^{(t)}(\cdot, \cdot) : \mathbf{X} \times \mathcal{B}(\mathbf{X}) \rightarrow [0, 1]$ is given by

$$P^{(t)}(x, D) = \mathbf{P}_x(X_t \in D), \quad \forall x \in \mathbf{X}, D \in \mathcal{B}(\mathbf{X}),$$

and for $t = 0$, P^0 is defined as $P^0(x, \cdot) = \delta_x$, the Dirac measure that assigns probability 1 to the point x , for each $x \in \mathbf{X}$. Define averaged probability measures $\bar{P}_k(x, \cdot)$ for $k \geq 1$ and

$x \in \mathbf{X}$, as

$$\bar{P}_k(x, \cdot) = \frac{1}{k} \sum_{t=0}^{k-1} P^t(x, \cdot).$$

The Markov chain $\{X_t\}$ has the *weak Feller* property if for every bounded continuous function f on \mathbf{X} ,

$$Pf(x) := \int f(y)P(x, dy) = \mathbb{E}[f(X_1) \mid X_0 = x]$$

is a continuous function of x (Meyn and Tweedie, 2009, Prop. 6.1.1). Weak Feller Markov chains have nice properties. In our analysis, we will use in particular several properties relating to the invariant probability measures of these chains and convergence of certain probability measures to the invariant probability measures.

Recall that if μ and μ_t , $t \geq 0$, are probability measures on \mathbf{X} , $\{\mu_t\}$ is said to converge weakly to μ if $\int f d\mu_t \rightarrow \int f d\mu$ for every bounded continuous function f on \mathbf{X} . For $\{\mu_t\}$ that is not necessarily convergent, we shall call the limiting probability measure of any of its convergent subsequence, in the sense of weak convergence, a *weak limit* of $\{\mu_t\}$. For an (arbitrary) index set \mathcal{K} , a set of probability measures $\{\mu_k\}_{k \in \mathcal{K}}$ on \mathbf{X} is said to be *tight* if for every $\delta > 0$, there exists a compact set $D_\delta \subset \mathbf{X}$ such that $\mu_k(D_\delta) \geq 1 - \delta$ for all $k \in \mathcal{K}$. An important fact is that on a complete separable metric space, any tight sequence of probability measures has a further subsequence that converges weakly to some probability measure (Dudley, 2002, Theorem 11.5.4).

For weak Feller Markov chains, their averaged probability measures $\{\bar{P}_k(x, \cdot)\}_{k \geq 1}$ are known to have the following property; see e.g., the proof of Lemma 4.1 in (Meyn, 1989). It will be needed in our proofs of Theorems 8-9.

Lemma 4 *Let $\{X_t\}$ be a weak Feller Markov chain with transition kernel $P(\cdot, \cdot)$ on a metric space \mathbf{X} . For each $x \in \mathbf{X}$, any weak limit of $\{\bar{P}_k(x, \cdot)\}_{k \geq 1}$ is an invariant probability measure of $\{X_t\}$.*

Recall that the occupation probability measures of $\{X_t\}$, denoted $\{\mu_{x,t}\}$ for each initial condition $x \in \mathbf{X}$, are defined as follows:

$$\mu_{x,t}(D) := \frac{1}{t} \sum_{k=0}^{t-1} \mathbb{1}(X_k \in D), \quad \forall D \in \mathcal{B}(\mathbf{X}),$$

where the chain $\{X_t\}$ starts from $X_0 = x$, and each $\mu_{x,t}$ is a random variable taking values in the space of probability measures on \mathbf{X} . Let “ $\mathbf{P}_{x\text{-a.s.}}$ ” stand for “almost surely with respect to \mathbf{P}_x ”. The next lemma concerns the convergence of occupation probability measures of a weak Feller Markov chain. It is a result of Meyn (1989) and will be needed in our proofs of Theorems 10-11.

Lemma 5 (Meyn, 1989, Prop. 4.2) *Let $\{X_t\}$ be a weak Feller Markov chain with transition kernel $P(\cdot, \cdot)$ on a complete separable metric space \mathbf{X} . Suppose that*

- (i) $\{X_t\}$ has a unique invariant probability measure μ ;
- (ii) for each compact set $E \subset \mathbf{X}$, the set $\{\bar{P}_k(x, \cdot) \mid x \in E, k \geq 1\}$ is tight; and

- (iii) for all initial conditions $x \in \mathbf{X}$, there exists a sequence of compact sets $E_k \uparrow \mathbf{X}$ (that is $E_k \subset E_{k+1}$ for all k and $\cup_k E_k = \mathbf{X}$) such that

$$\lim_{k \rightarrow \infty} \liminf_{t \rightarrow \infty} \mu_{x,t}(E_k) = 1, \quad \mathbf{P}_{x\text{-a.s.}}$$

Then, for each initial condition $x \in \mathbf{X}$, the sequence $\{\mu_{x,t}\}$ of occupation probability measures converges weakly to μ , \mathbf{P}_x -almost surely.

The condition (iii) above is equivalent to that the sequence $\{\mu_{x,t}\}$ of occupation probability measures is almost surely tight for each initial condition.

4.3.2 PROOFS OF THEOREMS 8 AND 9

In this subsection we prove Theorem 8 for the algorithm (11) and Theorem 9 for its two variants (19) and (20). We also show that the conclusions of Theorems 8-9 hold for the perturbed version (23) of these algorithms as well. The proof arguments are largely the same for all the algorithms we consider here. So except where noted otherwise, it will be taken for granted through out this subsection that $\{\theta_t^\alpha\}$ is generated by either of the six algorithms just mentioned for a constant stepsize $\alpha > 0$.

We start with some preliminary analysis given in the next two lemmas. Recall $Z_t = (S_t, A_t, e_t, F_t)$ and $\{Z_t\}$ is a weak Feller Markov chain on $\mathcal{Z} := S \times \mathcal{A} \times \mathbb{R}^{n+1}$ (Yu, 2015a, Sec. 3.1), and its evolution is not affected by the θ -iterates. We consider the Markov chain $\{(Z_t, \theta_t^\alpha)\}$ on the state space $\mathcal{Z} \times B$ (note that this is a complete separable metric space). This chain also has the weak Feller property:

Lemma 6 *Let Assumption 1(ii) hold. The process $\{(Z_t, \theta_t^\alpha)\}$ is a weak Feller Markov chain.*

The proof of the preceding lemma is a straightforward verification using the definition of the weak Feller property. It is included in the arXiv version of this paper (Yu, 2015b) but omitted here due to space limit.

In order to study the behavior of multiple consecutive θ -iterates, we consider for $m \geq 1$, the m -step version of $\{(Z_t, \theta_t^\alpha)\}$, that is, the Markov chain $\{X_t\}$ on $(\mathcal{Z} \times B)^m$ where each state X_t consists of m consecutive states of the original chain $\{(Z_t, \theta_t^\alpha)\}$:

$$X_t = ((Z_t, \theta_t^\alpha), \dots, (Z_{t+m-1}, \theta_{t+m-1}^\alpha)).$$

Similarly to Lemma 6, it is straightforward to show that the m -step version of a weak Feller Markov chain is a weak Feller chain as well. Thus the m -step version of $\{(Z_t, \theta_t^\alpha)\}$ is also a weak Feller Markov chain, and we can apply the ergodic theorems for weak Feller Markov chains to analyze it. In particular, in this subsection we will use Lemma 4 to prove Theorems 8-9; in the next subsection we will also use Lemma 5.

In analyzing the m -step version of $\{(Z_t, \theta_t^\alpha)\}$, sometimes it will be more convenient for us to take as its initial condition the condition of just (Z_0, θ_0^α) —instead of $(Z_0, \theta_0^\alpha), \dots, (Z_{m-1}^\alpha, \theta_{m-1}^\alpha)$ —and to work with the following objects that are essentially equivalent to the averaged probability measures $\{P_k(x, \cdot)\}$ and the occupation probability measures $\{\mu_{x,t}\}$ defined earlier for a general Markov chain $\{X_t\}$. Specifically, with $\{X_t\}$ denoting the m -step

version of $\{(Z_t, \theta_t^z)\}$, for each $(z, \theta) \in \mathcal{Z} \times B$, we define probability measures $\tilde{P}_{(z, \theta)}^{(m, k)}$, $k \geq 1$, on the space $\mathbf{X} = (\mathcal{Z} \times B)^m$, by

$$\tilde{P}_{(z, \theta)}^{(m, k)}(D) := \frac{1}{k} \sum_{t=0}^{k-1} \mathbf{P}_{(z, \theta)}(X_t \in D), \quad \forall D \in \mathcal{B}(\mathbf{X}). \quad (50)$$

Similarly, we define occupation probability measures $\{\mu_{(z, \theta), t}^{(m)}\}$ for each $(z, \theta) \in \mathcal{Z} \times B$ by

$$\mu_{(z, \theta), t}^{(m)}(D) := \frac{1}{t} \sum_{k=0}^{t-1} \mathbb{1}(X_k \in D), \quad \forall D \in \mathcal{B}(\mathbf{X}), \quad (51)$$

where the initial $(Z_0, \theta_0^z) = (z, \theta)$. Compared with the definitions of $\{\tilde{P}_k(x, \cdot)\}$ and $\{\mu_{x, t}\}$ for $\{X_t\}$, apparently, all the previous conclusions given in Section 4.3.1 for $\{\tilde{P}_k(x, \cdot)\}$ and $\{\mu_{x, t}\}$ hold for $\{\tilde{P}_{(z, \theta)}^{(m, k)}\}$ and $\{\mu_{(z, \theta), t}^{(m)}\}$ as well; therefore we can use the objects $\{\tilde{P}_{(z, \theta)}^{(m, k)}\}$ and $\{\mu_{(z, \theta), t}^{(m)}\}$, and $\{\mu_{x, t}\}$, interchangeably in our analysis.

Lemma 7 *Let Assumption 1 hold. For $m \geq 1$, let $\{X_t\}$ be the m -step version of $\{(Z_t, \theta_t^z)\}$ on $\mathbf{X} = (\mathcal{Z} \times B)^m$, with transition kernel $P(\cdot, \cdot)$. Then $\{X_t\}$ satisfies the conditions (ii)-(iii) of Lemma 5.*

Proof To show that the condition (ii) of Lemma 5 is satisfied, fix a compact set $E \subset \mathbf{X}$ and let us first show that the set $\{P^t(x, \cdot) \mid x \in E, t \geq 0\}$ is tight. Since the set B is compact and the state and action spaces are finite, of concern here is just the tightness of the marginals of these probability measures on the space of the trace components $(e_t, F_t, \dots, e_{t+m-1}, F_{t+m-1})$ of the state X_t . By Prop. 11 (Appendix A), for all initial conditions of (e_0, F_0) in a given bounded subset of \mathbb{R}^{n+1} , $\sup_{t \geq 0} \mathbb{E}[\|(e_t, F_t)\|] \leq L$ for a constant L (that depends on the subset). So for the set E , applying the Markov inequality together with the union bound, we have that there exists a constant $L > 0$ such that for all $x \in E$ and $a > 0$, $\mathbf{P}_x(\sup_{k \leq t < k+m} \|(e_k, F_k)\| \geq a) \leq mL/a$ for all $k \geq 0$. Now for any given $\delta > 0$, let a be large enough so that $mL/a < \delta$ and let D_a be the closed ball in \mathbb{R}^{n+1} centered at the origin with radius a . Then for the compact set $D = (S \times A \times D_a \times B)^m$, we have $\mathbf{P}^k(x, D) = \mathbf{P}_x(\sup_{k \leq t < k+m} \|(e_t, F_t)\| \leq a) \geq 1 - \delta$ for all $x \in E$ and all $k \geq 0$. This shows that the set $\{P^t(x, \cdot) \mid x \in E, t \geq 0\}$ is tight. Consequently, the averages of the probability measures in this set must also form a tight set; in particular, the set $\{P_k(x, \cdot) \mid x \in E, k \geq 1\}$ must be tight. Hence $\{X_t\}$ satisfies the condition (ii) of Lemma 5.

Consider now the condition (iii) of Lemma 5. For positive integers k , let E_k in that condition be the compact set $(S \times A \times D_k \times B)^m$, where D_k is the closed ball of radius k in \mathbb{R}^{n+1} centered at the origin. We wish to show that for each initial condition $x \in \mathbf{X}$,

$$\lim_{k \rightarrow \infty} \liminf_{t \rightarrow \infty} \mu_{x, t}(E_k) = 1, \quad \mathbf{P}_{x-a.s.} \quad (52)$$

Since the θ_t -iterates do not affect the evolution of Z_t , they can be neglected in the proof. It is sufficient to consider instead the m -step version of $\{Z_t\}$ and show that for the compact sets $\tilde{E}_k = (S \times A \times D_k)^m$, it holds for any initial condition $z \in \mathcal{Z}$ of Z_0 that

where $\{\hat{\mu}_{z, t}^{(m)}\}$ are the occupation probability measures of the m -step version of $\{Z_t\}$, defined analogously to (51) with (Z_t, \dots, Z_{t+m-1}) in place of X_t .

To prove (52), consider $\{Z_t\}$ first and its occupation probability measures $\{\mu_{z, t}\}$ for each initial condition $Z_0 = z \in \mathcal{Z}$. By Theorem 2 (Section 2.4), \mathbf{P}_z -almost surely, $\{\mu_{z, t}\}$ converges weakly to ζ (the unique invariant probability measure of $\{Z_t\}$). So by (Dudley, 2002, Theorem 11.1.1), for the open set $\tilde{D}_k = S \times A \times D_k^c$, where D_k^c denotes the interior of D_k (i.e., D_k^c is the open ball with radius k), almost surely,

$$\liminf_{t \rightarrow \infty} \hat{\mu}_{z, t}(\tilde{D}_k) \geq \zeta(\tilde{D}_k), \quad \text{and hence} \quad \lim_{k \rightarrow \infty} \liminf_{t \rightarrow \infty} \hat{\mu}_{z, t}(\tilde{D}_k) = 1. \quad (53)$$

Now for the m -step version of $\{Z_t\}$, with $[\tilde{D}_k]^m$ denoting the Cartesian product of m copies of \tilde{D}_k , we have

$$\hat{\mu}_{z, t}^{(m)}([\tilde{D}_k]^m) := \frac{1}{t} \sum_{j=0}^{t-1} \mathbb{1}(Z_{j+j'} \in \tilde{D}_k), \quad 0 \leq j' < m) \geq 1 - \sum_{j'=0}^{m-1} \frac{1}{t} \sum_{j=0}^{t-1} \mathbb{1}(Z_{j+j'} \notin \tilde{D}_k). \quad (54)$$

For each $j' < m$, by the definition of $\hat{\mu}_{z, t}$, we have $\limsup_{t \rightarrow \infty} \frac{1}{t} \sum_{j=0}^{t-1} \mathbb{1}(Z_{j+j'} \notin \tilde{D}_k) = \limsup_{t \rightarrow \infty} \hat{\mu}_{z, t}(\tilde{D}_k^c)$, where \tilde{D}_k^c denotes the complement of \tilde{D}_k in $S \times A \times \mathbb{R}^{n+1}$. By (53), $\lim_{k \rightarrow \infty} \limsup_{t \rightarrow \infty} \hat{\mu}_{z, t}(\tilde{D}_k^c) = 0$ almost surely. Hence for each $j' < m$, we have $\lim_{k \rightarrow \infty} \limsup_{t \rightarrow \infty} \frac{1}{t} \sum_{j=0}^{t-1} \mathbb{1}(Z_{j+j'} \notin \tilde{D}_k) = 0$ almost surely. We then obtain from (54), by taking the limits as $t \rightarrow \infty$ and $k \rightarrow \infty$, that

$$\liminf_{k \rightarrow \infty} \liminf_{t \rightarrow \infty} \hat{\mu}_{z, t}^{(m)}([\tilde{D}_k]^m) \geq 1 - \sum_{j'=0}^{m-1} \limsup_{k \rightarrow \infty} \limsup_{t \rightarrow \infty} \frac{1}{t} \sum_{j=0}^{t-1} \mathbb{1}(Z_{j+j'} \notin \tilde{D}_k) = 1$$

almost surely. The desired equality (52) then follows, since $[\tilde{D}_k]^m \subset \tilde{E}_k$. \blacksquare

Recall that \mathcal{M}_a^m is the set of invariant probability measures of the m -step version of $\{(Z_t, \theta_t^z)\}$. By Lemma 7 the latter Markov chain satisfies the condition (ii) of Lemma 5, and this implies that the set $\{\tilde{P}_{(z, \theta)}^{(m, k)}\}_{k \geq 1}$ is tight for each initial condition $(Z_0, \theta_0^z) = (z, \theta)$. Recall that any subsequence of a tight sequence has a further convergent subsequence (Dudley, 2002, Theorem 11.5.4). For $\{\tilde{P}_{(z, \theta)}^{(m, k)}\}_{k \geq 1}$, all the weak limits (i.e., the limits of its convergent subsequences) must be invariant probability measures in \mathcal{M}_a^m , by the property of weak Feller Markov chains given in Lemma 4:

Proposition 6 *Under Assumption 1, consider the m -step version of $\{(Z_t, \theta_t^z)\}$ for $m \geq 1$. For each $(z, \theta) \in \mathcal{Z} \times B$, the sequence $\{\tilde{P}_{(z, \theta)}^{(m, k)}\}_{k \geq 1}$ of probability measures is tight, and any weak limit of this sequence is an invariant probability measure of the m -step version of $\{(Z_t, \theta_t^z)\}$. (Thus $\mathcal{M}_a^m \neq \emptyset$.)*

We are now ready to prove Theorems 8-9. The idea is to use the conclusions on the θ_t -iterates that we can obtain by applying (KY, Theorem 8.2.2), to infer the concentration of the mass around a small neighborhood of $(\theta^*, \dots, \theta^*)$ (m copies of θ^*) for the marginals

of all the invariant probability measures in the set \mathcal{M}_α^m , when α is sufficiently small. This can then be combined with Prop. 6 above to prove the desired conclusions on the θ -iterates for a given stepsize.

Recall that \mathcal{M}_α is the set of invariant probability measures of $\{(Z_t, \theta_t^\alpha)\}$. Recall also that \mathcal{M}_α^m denotes the set of marginals of the invariant probability measures in \mathcal{M}_α^m , on the space of the θ 's.

Proposition 7 *In the setting of Theorem 5, for each $\alpha > 0$, let $\{\theta_t^\alpha\}$ be generated instead by the algorithm (11) or its perturbed version (23), with constant stepsize α and under the condition that the initial (Z_0, θ_0^α) is distributed according to some invariant probability measure in \mathcal{M}_α . Then the conclusions of Theorem 5 continue to hold.*

Proof The proof arguments are the same as those for Theorem 5 given in Section 4.1. We only need to show that the conditions (ii) and (i')-(v') given in Section 4.1.1 for applying (KY, Theorem 8.2.2) are still satisfied under our present assumptions.

For the algorithm (11), the only difference from the previous assumptions in Theorem 5 is that here for each stepsize α , the initial (Z_0, θ_0^α) has a distribution $\mu_\alpha \in \mathcal{M}_\alpha$. The condition (ii) does not depend on such initial conditions, so it continues to hold. For the other conditions, note that since $\{Z_t\}$ has a unique invariant probability measure ζ (Theorem 2), regardless of the choice of μ_α , for all α , $\{Z_t\}$ is stationary and has the same distribution. Then the tightness condition (iii') trivially holds because as $\{\xi_t\}$ is also stationary and unaffected by the stepsize, each ξ_t^α in (iii') has the same distribution. Similarly, since $\{e_t\}$ is stationary and unaffected by the stepsize, and each e_t has the same distribution with the mean of $\|e_t\|$ given by $\mathbb{E}_\zeta[\|e_t\|] < \infty$ (Theorem 3), we obtain that $\{e_t\}$ is u.i. From this the uniform integrability required in the conditions (i') and (iv') follows as a consequence, as shown in the proof of Prop. 2(ii)-(iv). Lastly, the convergence in mean condition (v') continues to hold (by the same proof given for Prop. 3). This is because $\{\xi_t\}$ has the same distribution regardless of the stepsize, and because the condition (v') is for each compact set D and concerns tails of a trajectory starting at instants t with $\xi_t \in D$, which renders any initial condition on Z_0 ineffective. Thus all the required conditions are met, and we obtain the same conclusions on the θ -iterates as given in Theorem 5.

For the perturbed version (23) of the algorithm (11), the only difference to (11) under the present assumptions is the perturbation variables $\Delta_{\theta_t^\alpha}^m$ involved in each iteration. But by definition these variables have conditional zero mean: $\mathbb{E}_{\xi_t^\alpha}[\Delta_{\theta_t^\alpha}^m] = 0$, so the only condition in which they appear is the uniform integrability condition (i'): $\{Y_t^\alpha \mid t \geq 0, \alpha > 0\}$ is u.i., where Y_t^α is now given by $Y_t^\alpha = h(\theta_t^\alpha, \xi_t) + e_t \cdot \tilde{\omega}_{t+1} + \Delta_{\theta_t^\alpha}^m$. By definition $\Delta_{\theta_t^\alpha}^m$ for all α and t have bounded variance, and hence $\{\Delta_{\theta_t^\alpha}^m\}$ is u.i. (Billingsley, 1968, p. 32). The set $h(\theta_t^\alpha, \xi_t) + e_t \cdot \tilde{\omega}_{t+1} \mid t \geq 0, \alpha > 0\}$ is u.i., which follows from the u.i. of $\{e_t\}$, as we just verified in the case of the algorithm (11). Therefore, by Lemma 2(i), $\{Y_t^\alpha \mid t \geq 0, \alpha > 0\}$ is u.i. and the condition (i') is satisfied. Since the perturbed version (23) meets all the required conditions, and shares with (11) the same mean ODE, the same conclusions given in Theorem 5 hold for this algorithm as well. ■

We now prove Theorem 8 for the algorithm (11). We prove its part (i) and part (ii) separately, as the arguments are different. Our proofs below also apply to the perturbed

version (23) of the algorithm (11), and together with the preceding proposition, they establish the first part of Theorem 10 (which says that the conclusions of both Theorem 5 and Theorem 8 hold for the perturbed algorithm).

Proof of Theorem 8(i) Proof by contradiction. Consider the statement of Theorem 8(i):

$$\forall \delta > 0, \quad \liminf_{\alpha \rightarrow 0} \inf_{\mu \in \mathcal{M}_{\alpha}^{m_\alpha}} \mu([N_\delta(\theta^*)]^{m_\alpha}) = 1, \quad \text{where } m_\alpha = \lceil \frac{m}{\alpha} \rceil.$$

Suppose it is not true. Then there exist $\delta, \epsilon > 0$, $m \geq 1$, a sequence $\alpha_k \rightarrow 0$, and a sequence $\mu_{\alpha_k} \in \mathcal{M}_{\alpha_k}^{m_{\alpha_k}}$, where $m_{\alpha_k} = m_{\alpha_k}$, such that

$$\mu_{\alpha_k}([N_\delta(\theta^*)]^{m_{\alpha_k}}) \leq 1 - \epsilon, \quad \forall k \geq 0. \quad (55)$$

Each μ_{α_k} corresponds to an invariant probability measure of $\{(Z_t, \theta_t^{\alpha_k})\}$ in \mathcal{M}_{α_k} , which we denote by $\hat{\mu}_{\alpha_k}$. For each $k \geq 0$, generate the iterates $\{\theta_t^{\alpha_k}\}$ using $\hat{\mu}_{\alpha_k}$ as the initial distribution of $(Z_0, \theta_0^{\alpha_k})$. For other values of α , generate the iterates $\{\theta_t^\alpha\}$ using some $\hat{\mu}_\alpha \in \mathcal{M}_\alpha$ as the initial distribution of (Z_0, θ_0^α) . By Prop. 7, the conclusions of Theorem 5 hold:

$$\limsup_{\alpha \rightarrow 0} \mathbf{P}\left(\theta_t^\alpha \notin N_\delta(\theta^*), \text{ some } t \in [k_\alpha, k_\alpha + T_\alpha/\alpha]\right) = 0,$$

where $T_\alpha \rightarrow \infty$ as $\alpha \rightarrow 0$, and this implies for the given m ,

$$\limsup_{\alpha \rightarrow 0} \mathbf{P}\left(\theta_t^\alpha \notin N_\delta(\theta^*), \text{ some } t \in [k_\alpha, k_\alpha + \lceil \frac{m}{\alpha} \rceil]\right) = 0. \quad (56)$$

But for each $\alpha > 0$, the process $\{(Z_t, \theta_t^\alpha)\}$ with the initial distribution $\hat{\mu}_\alpha$ is stationary, so the probability in the left-hand side of (56) is just $1 - \mu_\alpha([N_\delta(\theta^*)]^{m_\alpha})$, for the marginal probability measure $\mu_\alpha \in \mathcal{M}_{\alpha}^{m_\alpha}$ that corresponds to the invariant probability measure $\hat{\mu}_\alpha$. Therefore, by (56), $\liminf_{\alpha \rightarrow 0} \mu_\alpha([N_\delta(\theta^*)]^{m_\alpha}) = 1$. On the other hand, by (55), $\liminf_{\alpha \rightarrow 0} \mu_{\alpha_k}([N_\delta(\theta^*)]^{m_{\alpha_k}}) \leq \liminf_{k \rightarrow \infty} \mu_{\alpha_k}([N_\delta(\theta^*)]^{m_{\alpha_k}}) < 1$, a contradiction. Thus the statement of Theorem 8(i) recaptured at the beginning of this proof must hold.

This also proves the other statement of Theorem 8(i), $\liminf_{\alpha \rightarrow 0} \inf_{\mu \in \mathcal{M}_\alpha^m} \mu([N_\delta(\theta^*)]^{m_\alpha}) = 1$, because for $\alpha < 1$, by the correspondences between those invariant probability measures in \mathcal{M}_α^m and those in $\mathcal{M}_{\alpha}^{m_\alpha}$, $\inf_{\mu \in \mathcal{M}_\alpha^m} \mu([N_\delta(\theta^*)]^{m_\alpha}) \geq \inf_{\mu \in \mathcal{M}_\alpha^{m_\alpha}} \mu([N_\delta(\theta^*)]^{m_\alpha})$. This completes the proof. ■

Proof of Theorem 8(ii) We suppress the superscript α of θ_t^α in the proof. The statement is trivially true if $\delta \geq 2r_B$, so consider the case $\delta < 2r_B$. Let $(z, \theta) \in Z \times B$ be the initial condition of (Z_0, θ_0) . By convexity of the Euclidean norm, $|\theta_t - \theta^*| \leq \frac{t}{\alpha} \sum_{j=0}^{t-1} |\theta_j - \theta^*|$, and therefore, for all $k \geq 1$,

$$\sup_{k \leq t < k+m} |\bar{\theta}_t - \theta^*| \leq \frac{1}{k} \sum_{j=0}^{k-1} \sup_{j \leq t < j+m} |\theta_t - \theta^*|, \quad (57)$$

and

$$\mathbb{E} \left[\sup_{k \leq t < k+m} |\bar{\theta}_t - \theta^*| \right] \leq \frac{1}{k} \sum_{j=0}^{k-1} \mathbb{E} \left[\sup_{j \leq t < j+m} |\theta_t - \theta^*| \right]. \quad (58)$$

With $N'_\delta(\theta^*)$ denoting the open δ -neighborhood of θ^* , we have

$$\begin{aligned}
& \frac{1}{k} \sum_{j=0}^{k-1} \mathbb{E} \left[\sup_{j \leq t < j+m} |\theta_t - \theta^*| \right] \\
& \leq \frac{1}{k} \sum_{j=0}^{k-1} \mathbb{E} \left[\left(\sup_{j \leq t < j+m} |\theta_t - \theta^*| \cdot \mathbb{1}(\theta_t \in N'_\delta(\theta^*), j \leq t < j+m) \right) \right. \\
& \quad \left. + \frac{1}{k} \sum_{j=0}^{k-1} \mathbb{E} \left[\left(\sup_{j \leq t < j+m} |\theta_t - \theta^*| \right) \cdot \mathbb{1}(\theta_t \notin N'_\delta(\theta^*), \text{some } t \in [j, j+m]) \right) \right] \\
& \leq \delta \cdot \bar{P}_{(z, \theta)}^{(m, k)}(D_\delta) + 2r_B \cdot (1 - \bar{P}_{(z, \theta)}^{(m, k)}(D_\delta)), \tag{59}
\end{aligned}$$

where $D_\delta = \{(z^1, \theta^1, \dots, z^m, \theta^m) \in (\mathcal{Z} \times B)^m \mid \sup_{1 \leq i \leq m} |\theta^i - \theta^*| < \delta\}$, and the second inequality follows from the definition (50) of the averaged probability measure $\bar{P}_{(z, \theta)}^{(m, k)}$.

By Prop. 6, $\{\bar{P}_{(z, \theta)}^{(m, k)}\}_{k \geq 1}$ is tight and all its weak limits are in \mathcal{M}_α^m , the set of invariant probability measure of the m -step version of $\{(Z_t, \theta_t)\}$. There is also the fact that on a metric space, if a sequence of probability measures μ_k converges to some probability measure μ weakly, then $\liminf_{k \rightarrow \infty} \mu_k(D) \geq \mu(D)$ for any open set D (Dudley, 2002, Theorem 11.1.1). From these two arguments we have that for the set D_δ , which is open with respect to the topology on $(\mathcal{Z} \times B)^m$,

$$\liminf_{k \rightarrow \infty} \bar{P}_{(z, \theta)}^{(m, k)}(D_\delta) \geq \inf_{\mu \in \mathcal{M}_\alpha^m} \mu(D_\delta) = \inf_{\mu \in \mathcal{M}_\alpha^m} \mu([N'_\delta(\theta^*)]^m) =: \kappa_{\alpha, m}. \tag{60}$$

Combining the three inequalities (58)-(60), and using also the relation $\delta < 2r_B$, we obtain

$$\limsup_{k \rightarrow \infty} \mathbb{E} \left[\sup_{k \leq t < k+m} |\bar{\theta}_t - \theta^*| \right] \leq \delta \kappa_{\alpha, m} + 2r_B(1 - \kappa_{\alpha, m}).$$

This complete the proof. \blacksquare

We prove Theorem 9 in exactly the same way as we proved Theorem 8, so we omit the details and only outline the proof here. First, for the variant algorithms (19) and (20) as well as their perturbed version (23), we consider fixed K and ψ_K . Similar to Prop. 7, we show that if for each stepsize α , the initial (Z_0, θ_0^α) is distributed according to some invariant probability measure in \mathcal{M}_α , then the algorithms continue to satisfy the conditions given in Section 4.1.1, so we can apply (KY, Theorem 8.2.2) to assert that the conclusions of Theorem 5 continue to hold with $N_\delta(\theta^*)$ replaced by the limit set $N_\delta(L_B)$ of the mean ODE associated with each algorithm. (Recall Theorem 7 is also obtained in this way.) Subsequently, with $N_\delta(L_B)$ in place of $N_\delta(\theta^*)$ again, and with K and ψ_K still held fixed, we use the same proof for Theorem 8(i) to obtain that for any $\delta > 0$ and $m \geq 1$,

$$\liminf_{\alpha \rightarrow 0} \inf_{\mu \in \mathcal{M}_{m\alpha}^m} \mu([N_\delta(L_B)]^{m\alpha}) = 1, \quad \text{where } m_\alpha = \lceil \frac{m}{\alpha} \rceil.$$

Finally, we combine this with the fact that given any $\delta > 0$, the limit set $N_\delta(L_B) \subset N_\delta(\theta^*)$ for all K sufficiently large (see Lemma 3 in Section 4.2, which holds for (19) and (20), as well as their perturbed version (23) since the latter has the same mean ODE as the original algorithm). Theorem 9(i) then follows: given $\delta > 0$, for all K sufficiently large,

$$\liminf_{\alpha \rightarrow 0} \inf_{\mu \in \mathcal{M}_{m\alpha}^m} \mu([N_\delta(\theta^*)]^{m\alpha}) = 1.$$

The proof for Theorem 9(ii) is exactly the same as that for Theorem 8(ii) given earlier. In particular, this proof relies solely on the weak Feller property of the Markov chain $\{(Z_t, \theta_t^*)\}$ and the convergence property of the averaged probability measures of the m -step version of $\{(Z_t, \theta_t^*)\}$, all of which have shown to hold for the algorithms (19) and (20) and their perturbed version (23) in this subsection.

The preceding arguments also show that the first part of Theorem 11 holds; that is, the conclusions of Theorem 7 and Theorem 9 hold for the perturbed version (23) of the algorithm (19) or (20) as well.

4.3.3 PROOFS OF THEOREMS 10 AND 11

In this subsection we establish completely Theorems 10 and 11 regarding the perturbed version (23) of the algorithms (11), (19) and (20). We have already proved the first part of both of these theorems in the previous subsection. Below we tackle their second part, which, as we recall, is stronger than the corresponding part of Theorems 8 and 9 in that for a fixed stepsize α , the deviation of the averaged iterates $\{\bar{\theta}_t^*\}$ from θ^* in the limit as $t \rightarrow \infty$ is now characterized not in an expected sense but for almost all sample paths.

To simplify the presentation, except where noted otherwise, it will be taken for granted throughout this subsection that $\{\theta_t^*\}$ is generated by the perturbed version (23) of any of the three algorithms (11), (19) and (20). Recall that when updating θ_t^* to θ_{t+1}^* , the perturbed algorithm (23) adds the perturbation term $\alpha \Delta_{\theta, t}$ to the iterate before the projection Π_B , where $\Delta_{\theta, t} \geq 0$, are assumed to be i.i.d. \mathbb{R}^m -valued random variables that have zero mean and bounded variances and have a positive continuous density function with respect to the Lebesgue measure. (Here and in what follows, we omit the superscript α of the noise terms $\Delta_{\theta, t}$ since we deal with a fixed stepsize α in this part of the analysis.) As mentioned in Section 3.3, these conditions are not as weak as possible. Indeed, the purpose of the perturbation is to make the invariant probability measure of $\{(Z_t, \theta_t^*)\}$ unique so that we can invoke the ergodic theorem for weak Feller Markov chains given in Lemma 5, Section 4.3.1. Therefore, any conditions that can guarantee the uniqueness of the invariant probability measure can be used. In the present paper, for simplicity, we focus on the conditions we assumed earlier on $\Delta_{\theta, t}$, and prove the uniqueness just mentioned under these conditions, although our proof arguments can be useful for weaker conditions as well.

Proposition 8 *Under Assumption 1, $\{(Z_t, \theta_t^*)\}$ has a unique invariant probability measure.*

The next two lemmas are the intermediate steps to prove Prop. 8. We need the notion of a stochastic kernel, of which the transition kernel of a Markov chain is one example. For two topological spaces \mathbf{X} and \mathbf{Y} , a function $Q: \mathcal{B}(\mathbf{X}) \times \mathbf{Y} \rightarrow [0, 1]$ is a (Borel measurable) stochastic kernel on \mathbf{X} given \mathbf{Y} , if for each $y \in \mathbf{Y}$, $Q(\cdot | y)$ is a probability measure

on $\mathcal{B}(\mathbf{X})$ and for each $D \in \mathcal{B}(\mathbf{X})$, $Q(D | y)$ is a Borel measurable function on \mathbf{Y} . For the algorithms we consider, the iteration that generates $(Z_{t+1}, \theta_{t+1}^\alpha)$ from (Z_t, θ_t^α) can be equivalently described in terms of stochastic kernels. In particular, the transition from Z_t to Z_{t+1} is described by the transition kernel of the Markov chain $\{Z_t\}$, and the probability distribution of θ_{t+1}^α given θ_t^α and $\xi_t = (e_t, F_t, S_t, A_t, S_{t+1})$ is described by another stochastic kernel, which will be our focus in the analysis below.

Lemma 8 *Let Assumption 1(ii) hold. Let $Q(d\theta' | \xi, \theta)$ be the stochastic kernel (on B given $\Xi \times B$) that describes the probability distribution of θ_{t+1}^α given $\xi_t = \xi, \theta_t^\alpha = \theta$. Then for each bounded set $E \subset \Xi$, there exist $\beta \in (0, 1]$ and a probability measure Q_1 on B such that*

$$Q(d\theta' | \xi, \theta) \geq \beta Q_1(d\theta'), \quad \forall \xi \in E, \theta \in B. \quad (61)$$

Proof We consider only the case where $\{\theta_t^\alpha\}$ is generated by the perturbed version of the algorithm (11); the proof for the perturbed version of the two other algorithms (19) and (20) follows exactly the same arguments. In the proof below we use the notation that for a scalar c and a set $D \subset \mathbb{R}^n$, the set $cD = \{cx | x \in D\}$.

By the definitions of the algorithms (11) and (23), for $\xi = (e, F, s, a, s') \in \Xi$ and $\theta \in B$, we can express $Q(\cdot | \xi, \theta)$ as

$$Q(D | \xi, \theta) = \int \mathbb{1}(\Pi_B(\theta + \alpha f(\xi, \theta, r) + \alpha \Delta) \in D) p(d\Delta) q(dr | s, a, s'), \quad \forall D \in \mathcal{B}(B), \quad (62)$$

where $f(\xi, \theta, r) = e \cdot \rho(s, a)(r + \gamma(s')) \phi(s')^\top \theta - \phi(s)^\top \theta$, and $p(\cdot)$ is the common distribution of the perturbation variables $\Delta \theta_t$. Let $\bar{r} > 0$ be large enough so that for some $c > 0$, $q([- \bar{r}, \bar{r}] | \bar{s}, \bar{a}, \bar{s}') \geq c$ for all $(\bar{s}, \bar{a}, \bar{s}') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$. Let E be an arbitrary bounded subset of Ξ . For all $\xi \in E$, $\theta \in B$ and $r \in [- \bar{r}, \bar{r}]$, since E and B are bounded, $g(\xi, \theta, r) := (\theta + \alpha f(\xi, \theta, r))/\alpha$ lies in a compact subset of \mathbb{R}^n , which we denote by D . Let $\epsilon \in (0, r_B/\alpha]$ and let D_ϵ be the ϵ -neighborhood of D . By our assumption on the perturbation variables involved in the algorithm (23), $p(\cdot)$ has a positive continuous density function with respect to the Lebesgue measure $\ell(\cdot)$. Therefore, there exists some $c' > 0$ such that for any Borel subset D of the compact set $-D_\epsilon := \{-x | x \in D_\epsilon\}$, $p(D) \geq c' \ell(D)$.

Now consider an arbitrary $\xi \in E$, $\theta \in B$, and $r \in [- \bar{r}, \bar{r}]$. We have $y := g(\xi, \theta, r) \in D$. Let $B_\epsilon(-y)$ be the ϵ -neighborhood of $-y$, and let B_ϵ denote the closed ball in \mathbb{R}^n centered at the origin with radius ϵ . If $\Delta \in B_\epsilon(-y)$, then $\theta + \alpha f(\xi, \theta, r) + \alpha \Delta = \alpha y + \alpha \Delta \in \alpha B_\epsilon \subset B$ (since $\alpha \epsilon \leq r_B$). Therefore, for any $D \in \mathcal{B}(B)$,

$$\begin{aligned} \int \mathbb{1}(\Pi_B(\theta + \alpha f(\xi, \theta, r) + \alpha \Delta) \in D) p(d\Delta) &\geq \int_{B_\epsilon(-y)} \mathbb{1}(\alpha y + \alpha \Delta \in D) p(d\Delta) \\ &\geq c' \int_{B_\epsilon(-y)} \mathbb{1}(\alpha y + \alpha \Delta \in D) \ell(d\Delta) \\ &= c' \ell(\tfrac{1}{\alpha} D \cap B_\epsilon), \end{aligned} \quad (63)$$

where in the second inequality we used the fact that $B_\epsilon(-y) \subset -D_\epsilon$ and restricted to $B(-D_\epsilon)$, $p(d\Delta) \geq c' \ell(d\Delta)$, as discussed earlier.

To finish the proof, define the probability measure Q_1 on B by $Q_1(D) = \ell(\frac{1}{\alpha} D \cap B_\epsilon)/\ell(B_\epsilon)$ for all $D \in \mathcal{B}(B)$. Then for all $\xi \in E$ and $\theta \in B$, using (62) and (63) and our choice of \bar{r} , we have

$$Q(D | \xi, \theta) \geq \int_{[- \bar{r}, \bar{r}]} c' \ell(B_\epsilon) \cdot Q_1(D) q(dr | s, a, s') \geq c \cdot c' \ell(B_\epsilon) \cdot Q_1(D), \quad D \in \mathcal{B}(B),$$

and the desired inequality (61) then follows by letting $\beta = cc' \ell(B_\epsilon) > 0$ (we must have $\beta \leq 1$ since we can choose $D = B$ in the inequality above). \blacksquare

We will use the preceding result in the proof of the next lemma.

Lemma 9 *Let Assumption 1 hold. Let $\{\mu_{x,t}\}$ be the sequence of occupation probability measures of $\{(Z_t, \theta_t^\alpha)\}$ for each initial condition $x \in \mathcal{Z} \times B$. Suppose that for some $x = (z, \theta) \in \mathcal{Z} \times B$ and $\mu \in \mathcal{M}_\alpha$, $\{\mu_{x,t}\}$ converges weakly to μ , \mathbf{P}_x -almost surely. Then for each $\theta' \in B$ and $x' = (z', \theta')$, $\{\mu_{x',t}\}$ also converges weakly to μ , $\mathbf{P}_{x'}$ -almost surely.*

Proof We use a coupling argument to prove the statement. In the proof, we suppress the superscript α of θ_t^α . Let $\{X_t\}$ denote the process $\{(Z_t, \theta_t)\}$ with initial condition $x = (z, \theta)$, and let $\{X'_t\}$ denote the process $\{(Z_t, \theta_t)\}$ with initial condition $x' = (z', \theta')$, for an arbitrary $\theta' \in B$. In what follows, we first define a sequence

$$\{(Z_t, \tilde{\theta}_t, \hat{\theta}_t)\} \quad \text{with } (Z_0, \tilde{\theta}_0, \hat{\theta}_0) = (z, \theta, \theta'),$$

in such a way that the two marginal processes $\{(Z_t, \tilde{\theta}_t)\}$ and $\{(Z_t, \hat{\theta}_t)\}$ have the same probability distributions as $\{X_t\}$ and $\{X'_t\}$, respectively. We then relate the occupation probability measures $\{\mu_{x,t}\}$, $\{\mu_{x',t}\}$ to those of the marginal processes, $\{\tilde{\mu}_{x,t}\}$, $\{\hat{\mu}_{x',t}\}$, which are defined as

$$\tilde{\mu}_{x,t}(D) = \frac{1}{t} \sum_{k=0}^{t-1} \mathbb{1}((Z_k, \tilde{\theta}_k) \in D), \quad \hat{\mu}_{x',t}(D) = \frac{1}{t} \sum_{k=0}^{t-1} \mathbb{1}((Z_k, \hat{\theta}_k) \in D), \quad \forall D \in \mathcal{B}(\mathcal{Z} \times B).$$

We now define $\{(Z_t, \tilde{\theta}_t, \hat{\theta}_t)\}$. First, let $\{Z_t\}$ be generated as before with $Z_0 = z$. Denote $\xi_t = (e_t, F_t, S_t, A_t, S_{t+1})$ as before, and let Q be the stochastic kernel that describes the evolution of θ_{t+1} given (ξ_t, θ_t) . By Lemma 7, the occupation probability measures of $\{Z_t\}$ is almost surely tight for each initial condition. This implies the existence of a compact set $\bar{E} \subset \mathbb{R}^{n+1}$ such that for the compact set $E = \bar{E} \times \mathcal{S} \times \mathcal{A} \times \mathcal{S} \subset \Xi$, the sequence $\{\xi_t\}$ visits E infinitely often with probability one. For this set E , by Lemma 8, there exist some $\beta \in (0, 1]$ and probability measure Q_1 on B such that $Q(\cdot | \xi, \tilde{\theta}) \geq \beta Q_1(\cdot)$ for all $\xi \in E$ and $\tilde{\theta} \in B$. Therefore, on $E \times B$, we can write $Q(\cdot | \xi, \tilde{\theta})$ as the convex combination of Q_1 and another stochastic kernel Q_0 as follows:

$$Q(\cdot | \xi, \tilde{\theta}) = \beta Q_1(\cdot) + (1 - \beta) Q_0(\cdot | \xi, \tilde{\theta}), \quad \forall \xi \in E, \tilde{\theta} \in B, \quad (64)$$

where $Q_0(\cdot | \xi, \tilde{\theta}) = [Q(\cdot | \xi, \tilde{\theta}) - \beta Q_1(\cdot)] / (1 - \beta)$ and Q_0 is a stochastic kernel on B given $E \times B$.

Next, independently of $\{Z_t\}$, generate a sequence $\{Y_t\}_{t \geq 1}$ of i.i.d., $\{0, 1\}$ -valued random variables such that $Y_t = 1$ with probability β and $Y_t = 0$ with probability $1 - \beta$. Set $Y_0 = 0$. Let

$$t_Y = \min\{t \geq 1 \mid Y_t = 1, \xi_{t-1} \in E\}.$$

Then $t_Y < \infty$ with probability one. (Since $\{\xi_t\}$ visits E infinitely often and the process $\{Y_t\}$ is independent of $\{\xi_t\}$, this follows easily from applying the Borel-Cantelli lemma to $\{(\xi_{t_k}, Y_{t_k+1})\}_{k \geq 1}$, where t_k is when the k -th visit to E by $\{\xi_t\}$ occurs.)

Now for each $t \geq 0$, let us define the pair $(\tilde{\theta}_{t+1}, \tilde{\theta}'_{t+1})$ according to the following rule, based on the values of $(\xi_0, \tilde{\theta}_0, \tilde{\theta}'_0), \dots, (\xi_t, \tilde{\theta}_t, \tilde{\theta}'_t)$ and (Y_0, \dots, Y_{t+1}) :

- (i) In the case $t < t_Y$ and $\xi_t \notin E$, generate $\tilde{\theta}_{t+1}$ and $\tilde{\theta}'_{t+1}$ according to $Q(\cdot \mid \xi_t, \tilde{\theta}_t)$ and $Q(\cdot \mid \xi_t, \tilde{\theta}'_t)$ respectively.
- (ii) In the case $t < t_Y$ and $\xi_t \in E$, if $Y_{t+1} = 0$, generate $\tilde{\theta}_{t+1}$ and $\tilde{\theta}'_{t+1}$ according to $Q_0(\cdot \mid \xi_t, \tilde{\theta}_t)$ and $Q_0(\cdot \mid \xi_t, \tilde{\theta}'_t)$ respectively; if $Y_{t+1} = 1$, generate $\tilde{\theta}_{t+1}$ according to $Q_1(\cdot)$ and let $\tilde{\theta}'_{t+1} = \tilde{\theta}_{t+1}$.
- (iii) In the case $t \geq t_Y$, generate $\tilde{\theta}_{t+1}$ according to $Q(\cdot \mid \xi_t, \tilde{\theta}_t)$ and let $\tilde{\theta}'_{t+1} = \tilde{\theta}_{t+1}$.

In view of (64), it can be verified directly by induction on t that the marginal process $\{(Z_t, \tilde{\theta}_t)\}$ (resp. $\{(Z_t, \tilde{\theta}'_t)\}$) in the preceding construction has the same probability distribution as $\{X_t\}$ (resp. $\{X'_t\}$). This implies that $\{\mu_{x,t}\}$ (resp. $\{\mu_{x',t}\}$) converges weakly to μ with probability one if and only if $\{\tilde{\mu}_{x,t}\}$ (resp. $\{\tilde{\mu}_{x',t}\}$) converges weakly to μ with probability one. On the other hand, by construction $\tilde{\theta}_t = \tilde{\theta}'_t$ for $t \geq t_Y$, where $t_Y < \infty$ with probability one, so except on a null set, $\{\tilde{\mu}_{x,t}\}$ and $\{\tilde{\mu}_{x',t}\}$ have the same weak limits. Combining these two arguments with the assumption that $\{\mu_{x,t}\}$ converges weakly to μ with probability one, it follows that the three sequences $\{\mu_{x,t}\}$, $\{\mu_{x',t}\}$, and $\{\tilde{\mu}_{x',t}\}$ must all converge weakly to μ with probability one. ■

Proof of Prop. 8 We suppress the superscript α of θ_t^α in the proof. Let $\{X_t\} = \{(Z_t, \theta_t)\}$. By Prop. 6, the set \mathcal{M}_α of invariant probability measures of $\{X_t\}$ is nonempty. Recall also that since the evolution of $\{Z_t\}$ is not affected by the θ -iterates, the marginal of any $\mu \in \mathcal{M}_\alpha$ on the space \mathcal{Z} must equal ζ , the unique invariant probability measure of $\{Z_t\}$ (Theorem 2).

Suppose $\{X_t\}$ has multiple invariant probability measures; i.e., there exist $\mu, \mu' \in \mathcal{M}_\alpha$ with $\mu \neq \mu'$. Then by (Dudley, 2002, Theorem 11.3.2) there exists a bounded continuous function f on $\mathcal{Z} \times B$ such that

$$\int f d\mu \neq \int f d\mu'. \quad (65)$$

On the other hand, since μ is an invariant probability measure of $\{X_t\}$, applying a strong law of large numbers for stationary processes (Doob, 1953, Chap. X, Theorem 2.1; see also Meyn and Tweedie, 2009, Lemma 17.1.1 and Theorem 17.1.2) to the stationary Markov chain $\{X_t\}$ with initial distribution μ , we have that there exist a set $D_1 \subset \mathcal{Z} \times B$ with $\mu(D_1) = 1$ and a measurable function g_f on $\mathcal{Z} \times B$ such that

- (i) for each $x \in D_1$, with the initial condition $X_0 = x$, $\lim_{t \rightarrow \infty} \frac{1}{t} \sum_{k=0}^{t-1} f(X_k) = g_f(x)$, $\mathbf{P}_{x^{\text{-a.s.}}}$;
- (ii) $\mathbb{E}_\mu[g_f(X_0)] = \mathbb{E}_\mu[f(X_0)]$ (i.e., $\int g_f d\mu = \int f d\mu$).

The same is true for the invariant probability measure μ' : there exist a set $D_2 \subset \mathcal{Z} \times B$ with $\mu'(D_2) = 1$ and a measurable function $g'_f(x)$ such that

- (i) for each $x \in D_2$, with the initial condition $X_0 = x$, $\lim_{t \rightarrow \infty} \frac{1}{t} \sum_{k=0}^{t-1} f(X_k) = g'_f(x)$, $\mathbf{P}_{x^{\text{-a.s.}}}$;
- (ii) $\mathbb{E}_{\mu'}[g'_f(X_0)] = \mathbb{E}_{\mu'}[f(X_0)]$ (i.e., $\int g'_f d\mu' = \int f d\mu'$).

Also, since $\{X_t\}$ is a weak Feller Markov chain (Lemma 6), by (Meyn, 1989, Prop. 4.1), for a set of initial conditions x with μ -measure 1, the occupation probability measures $\{\mu_{x,t}\}$ of $\{X_t\}$ converge weakly, $\mathbf{P}_{x^{\text{-almost surely}}}$, to some (nonrandom) $\tilde{\mu}_x \in \mathcal{M}_\alpha$ that depends on the initial x . The same is true for μ' . So by excluding from D_1 a μ -null set and from D_2 a μ' -null set if necessary, we can assume that the sets D_1, D_2 above also satisfy that for each $x \in D_1 \cup D_2$, the occupation probability measures $\{\mu_{x,t}\}$ converge weakly to an invariant probability measure $\tilde{\mu}_x$ almost surely. Then since $\frac{1}{t} \sum_{k=0}^{t-1} f(X_k)$ is the same as $\int f d\mu_{x,t}$ for $X_0 = x$, we have, by the weak convergence of $\{\mu_{x,t}\}$ just discussed, that

$$g_f(x) = \int f d\tilde{\mu}_x \quad \text{for each } x \in D_1, \quad g'_f(x) = \int f d\tilde{\mu}_x \quad \text{for each } x \in D_2. \quad (66)$$

Certainly we must have $g_f(x) = g'_f(x)$ on $D_1 \cap D_2$. We now relate the values of these two functions at points that share the same z -component. In particular, let $\text{proj}(D_1)$ denote the projection of D_1 on \mathcal{Z} : $\text{proj}(D_1) = \{z \in \mathcal{Z} \mid \exists \theta \text{ with } (z, \theta) \in D_1\}$, and let $D_{1,z}$ be the vertical section of D_1 at z : $D_{1,z} = \{\theta \mid (z, \theta) \in D_1\}$. Define $\text{proj}(D_2)$ and $D_{2,z}$ similarly. If $x = (z, \theta) \in D_1 \cup D_2$ and $x' = (z, \theta') \in D_1 \cup D_2$, then in view of Lemma 9 and the weak convergence of $\{\mu_{x,t}\}$ and $\{\mu_{x',t}\}$, we must have $\tilde{\mu}_x = \tilde{\mu}_{x'}$. Consequently, by (66), for each $z \in \text{proj}(D_1)$, $g_f(z, \cdot)$ is constant on $D_{1,z}$; for each $z \in \text{proj}(D_2)$, $g'_f(z, \cdot)$ is constant on $D_{2,z}$; and for each $z \in \text{proj}(D_1) \cap \text{proj}(D_2)$, the constants that $g_f(z, \cdot)$, $g'_f(z, \cdot)$ take on $D_{1,z}, D_{2,z}$, respectively, are the same.

We now show $\int f d\mu = \int f d\mu'$ to contradict (65) and finish the proof. Since $\mu(D_1) = \mu'(D_2) = 1$ and by Theorem 2 (Section 2.4) μ, μ' have the same marginal distribution on \mathcal{Z} , which is ζ , there exists a Borel set $E \subset \text{proj}(D_1) \cap \text{proj}(D_2)$ with $\zeta(E) = 1$. Consider the sets $(E \times B) \cap D_1$ and $(E \times B) \cap D_2$, which have μ -measure 1 and μ' -measure 1, respectively. By (Dudley, 2002, Prop. 10.2.8), we can decompose μ, μ' into the marginal ζ on \mathcal{Z} and the conditional distributions $\mu(d\theta \mid z), \mu'(d\theta \mid z)$ for $z \in E$. Then

$$1 = \mu((E \times B) \cap D_1) = \int_E \mu(d\theta \mid z) \zeta(dz), \quad 1 = \mu'((E \times B) \cap D_2) = \int_E \int_{D_{2,z}} \mu'(d\theta \mid z) \zeta(dz),$$

where the equality for the iterated integral in each relation follows from (Dudley, 2002, Theorem 10.2.1(ii)). These relations imply that for some set $E_0 \subset E$ with $\zeta(E_0) = 0$,

$$\int_{D_{1,z}} \mu(d\theta \mid z) = \int_{D_{2,z}} \mu'(d\theta \mid z) = 1, \quad \forall z \in E \setminus E_0. \quad (67)$$

We now calculate $\int g_f d\mu$ and $\int g'_f d\mu'$. We have

$$\int g_f d\mu = \int_{(E \times B) \cap D_1} g_f d\mu = \int_E \int_{D_{1,z}} g_f(z, \theta) \mu(d\theta \mid z) \zeta(dz), \quad (68)$$

$$\int g'_f d\mu' = \int_{(E \times B) \cap D_2} g'_f d\mu' = \int_E \int_{D_{2,z}} g'_f(z, \theta) \mu'(d\theta \mid z) \zeta(dz), \quad (69)$$

where the equality for the iterated integral in each relation also follows from (Dudley, 2002, Theorem 10.2.1(ii)). As discussed earlier, for each $z \in E \subset \text{proj}(D_1) \cap \text{proj}(D_2)$, the two constant functions, $g_f(z, \cdot)$ on $D_{1,z}$ and $g_f(z, \cdot)$ on $D_{2,z}$, have the same value. Using this together with (67), we conclude that

$$\int_{D_{1,z}} g_f(z, \theta) \mu(d\theta | z) = \int_{D_{2,z}} g_f(z, \theta) \mu'(d\theta | z), \quad \forall z \in E \setminus E_0. \quad (70)$$

Since $\zeta(E_0) = 0$, we obtain from (68)-(70) that $\int g_f d\mu = \int g_f' d\mu'$. But $\int g_f d\mu = \int f d\mu$ and $\int g_f' d\mu' = \int f' d\mu'$ (as we obtained at the beginning of the proof), so $\int f d\mu = \int f' d\mu'$, a contradiction to (65). This proves that $\{X_t\}$ must have a unique invariant probability measure. ■

Proposition 8 implies that for every $m \geq 1$, the m -step version of $\{(Z_t, \theta_t^*)\}$ has a unique invariant probability measure. This together with Lemma 7 (Section 4.3.2) furnishes the conditions (A1)-(A3) of (Meyn, 1989, Prop. 4.2) for weak Feller Markov chains (these conditions are the conditions (i)-(iii) of our Lemma 5). We can therefore apply the conclusions of (Meyn, 1989, Prop. 4.2) (see Lemma 5 in our Section 4.3.1) to the m -step version of $\{(Z_t, \theta_t^*)\}$ here, and the result is the following proposition:

Proposition 9 *Under Assumption 1, for each $m \geq 1$, the m -step version of $\{(Z_t, \theta_t^*)\}$ has a unique invariant probability measure $\mu^{(m)}$, and the occupation probability measures $\mu_{(z,\theta),t}^{(m)}$, $t \geq 1$, as defined by (51), converge weakly to $\mu^{(m)}$ almost surely, for each initial condition $(z, \theta) \in \mathcal{Z} \times B$ of (Z_0, θ_0^*) .*

With Prop. 9 we can proceed to prove the second part of Theorems 10 and 11. Given that we have already established their first part in the previous subsection, the arguments for their second part are the same for both theorems and are given below. The proof is similar to that for Theorem 8(ii) in Section 4.3.2, except that here, instead of working with the averaged probability measures $\{P_{(z,\theta)}^{(m,k)}\}$, Prop. 9 allows us to work with the occupation probability measures.

Proof of the second part of both Theorem 10 and Theorem 11 We suppress the superscript α of θ_α^* in the proof. By Prop. 9, $\{(Z_t, \theta_t)\}$ has a unique invariant probability measure μ_α , and its m -step version has a corresponding unique invariant probability measure $\mu_\alpha^{(m)}$. We prove first the statement that for each initial condition $(z, \theta) \in \mathcal{Z} \times B$, almost surely,

$$\liminf_{t \rightarrow \infty} \frac{1}{t} \sum_{k=0}^{t-1} \mathbb{1} \left(\sup_{k \leq j < k+m} |\theta_j - \theta^*| < \delta \right) \geq \mu_\alpha^{(m)}([N_\delta^*(\theta^*)]^m), \quad (71)$$

where $\bar{\mu}_\alpha^{(m)}$ is the unique element in $\bar{\mathcal{M}}_\alpha^m$, and $N_\delta^*(\theta^*)$ is the open δ -neighborhood of θ^* . For each t , by the definition (51) of the occupation probability measure $\mu_{(z,\theta),t}^{(m)}$, the average in the left-hand side above is the same as $\mu_{(z,\theta),t}^{(m)}(D_\delta)$, where $D_\delta = \{z^1, \theta^1, \dots, z^m, \theta^m\} \in (\mathcal{Z} \times B)^m \mid \sup_{1 \leq j \leq m} |\theta^j - \theta^*| < \delta\}$. By Prop. 9, $\mathbf{P}_{(z,\theta)}$ -almost surely, $\{\mu_{(z,\theta),t}^{(m)}\}$ converges

weakly to $\mu_\alpha^{(m)}$, and therefore, except on a null set of sample paths, we have by (Dudley, 2002, Theorem 11.1.1) that for the open set D_δ ,

$$\liminf_{t \rightarrow \infty} \mu_{(z,\theta),t}^{(m)}(D_\delta) \geq \mu_\alpha^{(m)}(D_\delta) = \bar{\mu}_\alpha^{(m)}([N_\delta^*(\theta^*)]^m).$$

This proves (71).

We now prove the statement that for each initial condition $(z, \theta) \in \mathcal{Z} \times B$, almost surely,

$$\limsup_{t \rightarrow \infty} |\bar{\theta}_t - \theta^*| \leq \delta \kappa_\alpha + 2r_B (1 - \kappa_\alpha), \quad \text{where } \kappa_\alpha = \bar{\mu}_\alpha(N_\delta^*(\theta^*)), \quad (72)$$

and $\bar{\mu}_\alpha$ is the marginal of μ_α on B . The statement is trivially true if $\delta \geq 2r_B$, so consider the case $\delta < 2r_B$. Fix an initial condition $(z, \theta) \in \mathcal{Z} \times B$ for (Z_0, θ_0) , and let $\{\mu_{(z,\theta),t}\}$ be the corresponding occupation probability measures of $\{(Z_t, \theta_t)\}$. For the averaged sequence $\{\bar{\theta}_t\}$, by convexity of the norm $|\cdot|$,

$$|\bar{\theta}_t - \theta^*| \leq \frac{1}{t} \sum_{k=0}^{t-1} |\theta_k - \theta^*|. \quad (73)$$

We have

$$\begin{aligned} \frac{1}{t} \sum_{k=0}^{t-1} |\theta_k - \theta^*| &\leq \frac{1}{t} \sum_{k=0}^{t-1} |\theta_k - \theta^*| \cdot \mathbb{1}(\theta_k \in N_\delta^*(\theta^*)) + \frac{1}{t} \sum_{k=0}^{t-1} |\theta_k - \theta^*| \cdot \mathbb{1}(\theta_k \notin N_\delta^*(\theta^*)) \\ &\leq \delta \cdot \mu_{(z,\theta),t}(D_\delta) + 2r_B \cdot (1 - \mu_{(z,\theta),t}(D_\delta)), \end{aligned} \quad (74)$$

where $D_\delta = \{z^1, \theta^1\} \in \mathcal{Z} \times B \mid |\theta^1 - \theta^*| < \delta\}$. By Prop. 9, $\mathbf{P}_{(z,\theta)}$ -almost surely, $\{\mu_{(z,\theta),t}\}$ converges weakly to μ_α . Therefore, except on a null set of sample paths, we have by (Dudley, 2002, Theorem 11.1.1) that for the open set D_δ ,

$$\liminf_{t \rightarrow \infty} \mu_{(z,\theta),t}(D_\delta) \geq \mu_\alpha(D_\delta) = \bar{\mu}_\alpha(N_\delta^*(\theta^*)). \quad (75)$$

Combining the three inequalities (73)-(75), and using also the relation $\delta < 2r_B$, we obtain that (72) holds almost surely for each initial condition $(z, \theta) \in \mathcal{Z} \times B$. ■

Remark 3 (on the role of perturbation again) As mentioned before Prop. 8, our purpose of perturbing the constrained ETD algorithms is to guarantee that the Markov chain $\{(Z_t, \theta_t^*)\}$ has a unique invariant probability measure. Without the perturbation, this cannot be ensured, so we cannot apply the ergodic theorem given in Lemma 5 to exploit the convergence of occupation probability measures, as we did in the preceding proof, even though $\{(Z_t, \theta_t^*)\}$ satisfies the remaining two conditions required by that ergodic theorem (cf. Lemma 7, Section 4.3.2).

In connection with this discussion, let us clarify a point. We know that the occupation probability measures of $\{Z_t\}$ converge weakly to its unique invariant probability measure ζ almost surely for each initial condition of Z_0 (Theorem 2). But this fact alone cannot rule out the possibility that $\{(Z_t, \theta_t^*)\}$ has multiple invariant probability measures and that its occupation probability measures do not converge for some initial condition (z, θ) .

Finally, another property of weak Feller Markov chains and its implication for our problem are worth noting here. By (Meyn, 1989, Prop. 4.1), for a weak Feller Markov chain $\{X_t\}$, provided that an invariant probability measure μ exists, we have that for a set of initial conditions x with μ -measure 1, the occupation probability measures $\{\mu_{x,t}\}$ converge weakly, \mathbf{P}_x -almost surely, to an invariant probability measure μ_x that depends on the initial condition. Thus, for the unperturbed algorithms (11), (19) and (20), despite the possibility of $\{(Z_t, \theta_t^\alpha)\}$ having multiple invariant probability measures, the preceding proof can be applied to those initial conditions from which the occupation probability measures converge almost surely. In particular, this argument leads to the following conclusion. In the case of the algorithm (11), (19) or (20), under the same conditions as in Theorem 8 or 9, it holds for any invariant probability measure μ of $\{(Z_t, \theta_t^\alpha)\}$ that for each initial condition (z, θ) from some set of initial conditions with μ -measure 1,

$$\limsup_{t \rightarrow \infty} |\bar{\theta}_t^\alpha - \theta^*| \leq \delta \kappa_\alpha + 2r_B(1 - \kappa_\alpha) \quad \mathbf{P}_{(z, \theta)\text{-a.s.}},$$

where $\kappa_\alpha = \inf_{\mu \in \bar{\mathcal{M}}_\alpha} \mu(N_\delta^c(\theta^*))$. The limitation of this result, however, is that the set of initial conditions involved is unknown and can be small. ■

5. Discussion

In this section we discuss direct applications of our convergence results to ETD(λ) under relaxed conditions and to two other algorithms, the off-policy TD(λ) algorithm and the ETD(λ, β) algorithm (Hallak et al., 2016). We then discuss several open issues to conclude the paper.

5.1 The Case without Assumption 2

Let Assumption 1 hold. Recall from Section 2.3 that ETD(λ) aims to solve the equation $C\theta + b = 0$, where

$$b = \Phi^\top \bar{M} r_{\pi, \gamma}^\lambda, \quad C = -\Phi^\top G \Phi \quad \text{with} \quad G = \bar{M}(I - P_{\pi, \gamma}^\lambda).$$

In this paper we have focused on the case where Assumption 2 holds and C is negative definite (Theorem 1, Section 2.3). If Assumption 2 does not hold, then either there are less than n emphasized states (i.e., states s with $M_{ss} > 0$), or the feature vectors of emphasized states are not rich enough to contain n linearly independent vectors. In either case the function approximation capacity is not fully utilized. It is hence desirable to fulfill Assumption 2 by adding more states with positive interest weights $i(s)$ or by enriching the feature representation.

Nevertheless, suppose Assumption 2 does not hold (in which case C is negative semidefinite as shown by Sutton et al., 2016). This essentially has no effects on the convergence properties of the constrained or unconstrained ETD(λ) algorithms, because of the emphatic weighting scheme (3)-(5), as we explain now.

Let there be at least one state s with interest weight $i(s) > 0$ (the case is vacuous otherwise). Partition the state space into the set of emphasized states and the set of non-emphasized states:

$$\mathcal{J}_1 = \{s \in S \mid \bar{M}_{ss} > 0\}, \quad \mathcal{J}_0 = \{s \in S \mid \bar{M}_{ss} = 0\}.$$

Corresponding to the partition, by rearranging the indices of states if necessary, we can write

$$\Phi = \begin{bmatrix} \Phi_1 \\ \Phi_0 \end{bmatrix}, \quad r_{\pi, \gamma}^\lambda = \begin{bmatrix} r_1 \\ r_0 \end{bmatrix}, \quad \bar{M} = \begin{bmatrix} \bar{M} & 0_{|\mathcal{J}_1| \times |\mathcal{J}_0|} \\ 0_{|\mathcal{J}_0| \times |\mathcal{J}_1|} & 0_{|\mathcal{J}_0| \times |\mathcal{J}_0|} \end{bmatrix},$$

where $0_{m \times m'}$ denotes an $m \times m'$ zero matrix, \bar{M} is a diagonal matrix with \bar{M}_{ss} , $s \in \mathcal{J}_1$, as its diagonals. Let \hat{Q} be the sub-matrix of $P_{\pi, \gamma}^\lambda$ that consists of the entries whose row/column indices are in \mathcal{J}_1 . For the equation $C\theta + b = 0$, clearly $b = \Phi_1^\top \bar{M} r_1$. Consider now the matrix C . It is shown in the proof of Prop. C.2 in (Yu, 2015a) that G has a block-diagonal structure with respect to the partition $\{\mathcal{J}_1, \mathcal{J}_0\}$,

$$G = \begin{bmatrix} \hat{G} & 0_{|\mathcal{J}_1| \times |\mathcal{J}_0|} \\ 0_{|\mathcal{J}_0| \times |\mathcal{J}_1|} & 0_{|\mathcal{J}_0| \times |\mathcal{J}_0|} \end{bmatrix},$$

where the block corresponding to \mathcal{J}_0 is a zero matrix as shown above, and the block \hat{G} corresponding to \mathcal{J}_1 is a positive definite matrix given by

$$\hat{G} = \hat{M}(I - \hat{Q}), \quad (76)$$

and \hat{M} can be expressed explicitly as

$$\text{diag}(\hat{M}) = d_{\pi, i}^\top (I - \hat{Q})^{-1}, \quad d_{\pi, i}^\top \in \mathbb{R}^{|\mathcal{J}_1|}, \quad d_{\pi, i}^\top(s) = d_{\pi^o}(s) \cdot i(s), \quad s \in \mathcal{J}_1. \quad (77)$$

Thus the matrix C has a special structure:

Theorem 12 (structure of the matrix C ; Yu, 2015a, Appendix C.2, p. 41-44) *Let Assumption 1 hold, and let $i(s) > 0$ for at least one state $s \in S$. Then*

$$C = -\Phi_1^\top \hat{G} \Phi_1, \quad \text{where } \hat{G} = \hat{M}(I - \hat{Q}) \text{ is positive definite.}$$

Let $\text{range}(A)$ denote the range space of a matrix A . By the positive definiteness of the matrix \hat{G} given in the preceding theorem, the negative semidefinite matrix C possesses the following properties (we omit the straightforward proof):

Proposition 10 *Let Assumption 1 hold, and let $i(s) > 0$ for at least one state $s \in S$. Then the matrix C satisfies that*

- (i) $\text{range}(C) = \text{range}(C^\top) = \text{span}\{\phi(s) \mid s \in \mathcal{J}_1\}$; and
- (ii) there exists $c > 0$ such that for all $x \in \text{span}\{\phi(s) \mid s \in \mathcal{J}_1\}$, $x^\top C x \leq -c \|x\|^2$.

Two observations then follow immediately:

- (i) Since $b = \Phi_1^\top \bar{M} r_1 \in \text{span}\{\phi(s) \mid s \in \mathcal{J}_1\}$, Prop. 10(i) shows that the equation $C\theta + b = 0$ admits a solution, and a unique one in $\text{span}\{\phi(s) \mid s \in \mathcal{J}_1\}$, which we denote by θ^* .¹⁵

15. From the structures of G , $P_{\pi, \gamma}^\lambda$, \hat{Q} and \hat{M} shown in (Yu, 2015a, Appendix C.2, p. 41-44), which give rise to (76)-(77), we also have the following facts. The approximate value function $v = \Phi_1 \theta^*$ for the emphasized states \mathcal{J}_1 is the unique solution of the projected Bellman equation $v = \Pi(r_1 + Qv)$, where Π is the projection onto the column space of Φ_1 with respect to the weighted Euclidean norm on $\mathbb{R}^{|\mathcal{J}_1|}$ defined by the weights \bar{M}_{ss} , $s \in \mathcal{J}_1$ (the diagonals of \bar{M}). The equation $v = r_1 + Qv$ is indeed a generalized Bellman equation for the emphasized states only, and has $v_{\pi}(s)$, $s \in \mathcal{J}_1$, as its unique solution. Then for the emphasized states, the relation between the approximate value function $\Phi_1 \theta^*$ and v_{π} on \mathcal{J}_1 , in particular the approximation error, can again be characterized using the oblique projection viewpoint (Scherer, 2010), similar to the case with Assumption 2 discussed in Section 2.3.

(ii) Prop. 10(ii) shows that C acts like a negative definite matrix on the space of feature vectors, $\text{span}\{\phi(s)|s \in \mathcal{J}_1\}$, that the ETD(λ) algorithms naturally operate on.¹⁶

We remark that for an arbitrary negative semidefinite matrix C , neither of these conclusions holds. They hold here as direct consequences of the positive definiteness of the matrix \tilde{G} that underlies C , and this positive definiteness property is due to the emphatic weighting scheme (3)-(5) employed by ETD(λ).

Now let us discuss the behavior of the constrained ETD(λ) algorithms starting from some state S_0 of interest (i.e., $i(S_0) > 0$), in the absence of Assumption 2. Recall that earlier we did not need Assumption 2 when applying the two general convergence theorems from (Kushner and Yin, 2003), and we used the negative definiteness of C implied by this assumption only near the end of our proofs to get the solution properties of the mean ODE associated with each algorithm. In the absence of Assumption 2, for the unperturbed algorithms (11), (19) and (20), we can simply restrict attention to the subspace $\text{span}\{\phi(s)|s \in \mathcal{J}_1\}$ and use the property in Prop. 10(ii) in lieu of negative definiteness. After all, the θ -iterates of these algorithms always lie in the span of the feature vectors if the initial $\theta_0, e_0 \in \text{span}\{\phi(s)|s \in \mathcal{J}_1\}$ and in the case of the two biased algorithms (19) and (20), if the function $\psi_K(x)$ does not change the direction of x . On the subspace $\text{span}\{\phi(s)|s \in \mathcal{J}_1\}$, in view of Prop. 10(ii), the function $\theta - \theta^{*12}$ serves again as a Lyapunov function for analyzing the ODE solutions in exactly the same way as before. Thus, in the absence of Assumption 2, for the algorithms (11), (19) and (20) that set θ_0, e_0 and ψ_K as just described, and for $rB > |b|/c$ where c is as in Prop. 10(ii), the conclusions of Theorems 4-9 in Section 3 continue to hold with $N_\delta(\theta^*)$ or $N'_\delta(\theta^*)$ replaced by $N_\delta(\theta^*) \cap \text{span}\{\phi(s)|s \in \mathcal{J}_1\}$ or $N'_\delta(\theta^*) \cap \text{span}\{\phi(s)|s \in \mathcal{J}_1\}$.

The same is true for the almost sure convergence of the unconstrained ETD(λ) algorithm (2) under diminishing stepsize: with $i(S_0) > 0$ and $\theta_0, e_0 \in \text{span}\{\phi(s)|s \in \mathcal{J}_1\}$, the conclusion of (Yu, 2015a, Theorem 2.2) continues to hold in the absence of Assumption 2; that is, for $\alpha_t = O(1/t)$ with $\frac{\alpha_t}{\alpha_{t-1}} = O(1/t)$, $\theta_t \xrightarrow{\text{a.s.}} \theta^*$.

It can be seen now that without Assumption 2, complications can only arise through initializing the algorithms outside the desired subspace. We discussed such situations in the arXiv version of this paper (Yu, 2015b, Sec. 5.1), but we shall omit them here in part because it does not seem natural to initialize θ_0, e_0 with a component perpendicular to $\text{span}\{\phi(s)|s \in \mathcal{J}_1\}$ in the first place.

As a final note, in the absence of Assumption 2, any solution $\bar{\theta}$ of $C\bar{\theta} + b = 0$ gives *the same approximate value function for emphasized states*, but the approximate values $\Phi_{\bar{\theta}}\bar{\theta}$ for non-emphasized states in \mathcal{J}_0 are *different* for different solutions $\bar{\theta}$. Thus one needs to be cautious in using the approximate values $\Phi_{\bar{\theta}}\bar{\theta}$. They correspond to different extrapolations from the approximate values $\Phi_{\theta^*}\theta^*$ for the emphasized states, whereas $\Phi_{\bar{\theta}}\bar{\theta}$ is not defined to take into account approximation errors for those states in \mathcal{J}_0 , although its approximation error for emphasized states can be well characterized (cf. Footnote 15).

16. Start ETD(λ) from a state S_0 with $i(S_0) > 0$. It can be verified that the emphatic weighting scheme dictates that if $S_t \in \mathcal{J}_0$, then the emphasis weight M_t for that state must be zero. Consequently, e_t is a linear combination of the features of the emphasized states and the initial e_0 . So when $e_0 \in \text{span}\{\phi(s)|s \in \mathcal{J}_1\}$, $e_t \in \text{span}\{\phi(s)|s \in \mathcal{J}_1\}$ always, and if in addition $\theta_0 \in \text{span}\{\phi(s)|s \in \mathcal{J}_1\}$, then $\theta_t \in \text{span}\{\phi(s)|s \in \mathcal{J}_1\}$ always. This is very similar to the case of TD(λ) with possibly linearly dependent features discussed in (Tsitisklis and Van Roy, 1997).

5.2 Off-policy TD(λ) and ETD(λ, β)

Applying TD(λ) to off-policy learning by using importance sampling techniques was first proposed in (Precup et al., 2000, 2001), and the focus there was on episodic data. The analysis we gave in this paper applies directly to the (non-episodic) off-policy TD(λ) algorithm studied in (Bertsekas and Yu, 2009; Yu, 2012; Dann et al., 2014), when its divergence issue is avoided by setting λ sufficiently large. Specifically, we consider constant $\gamma \in [0, 1)$ and constant $\lambda \in [0, 1]$, and an infinitely long trajectory generated by the behavior policy as before. The algorithm is the same as TD(λ) except for incorporating the importance sampling weight p_t :¹⁷

$$\theta_{t+1} = \theta_t + \alpha_t e_t \cdot p_t (R_t + \gamma \phi(S_{t+1})^\top \theta_t - \phi(S_t)^\top \theta_t),$$

where

$$e_t = \lambda \gamma^{t-1} e_{t-1} + \phi(S_t).$$

The constrained versions of the algorithm are defined similarly to those for ETD(λ).

Under Assumption 1(ii), the associated projected Bellman equation is the same as that for on-policy TD(λ) (Tsitisklis and Van Roy, 1997) except that the projection norm is the weighted Euclidean norm with weights given by the steady state probabilities $d_{\pi^o}(s)$, $s \in \mathcal{S}$. Assuming Φ has full column rank, the corresponding equation in the θ -space, $C\theta + b = 0$, has the desired property that the matrix C is negative definite, if λ is sufficiently large (in particular if $\lambda = 1$) (Bertsekas and Yu, 2009). For that case, the conclusions given in this paper for constrained ETD(λ) all hold for the corresponding versions of off-policy TD(λ). (Similarly, for the case of C being negative semidefinite due to Φ having rank less than n , the discussion given in the previous subsection for ETD(λ) also applies.) The reason is that besides the property of C , the other properties of the iterates that we used in our analysis, which are given in Section 2 and Appendix A, all hold for off-policy TD(λ). In fact, some of these properties were first derived for off-policy LSTD(λ) and TD(λ) in (Yu, 2012) and extended later in (Yu, 2015a) to ETD(λ).

For the same reason, the convergence analyses we gave in (2015a) and this paper for ETD also apply to a variation of the ETD algorithm, ETD(λ, β), proposed recently by Hallak et al. (2016), when the parameter β is set in an appropriate range.

5.3 Open Issues

A major difficulty in applying off-policy TD learning, especially with $\lambda > 0$, is the high variances of the iterates. For ETD(λ), off-policy TD(λ) and their least-squares versions, because of the growing variances of products of the importance sampling weights $p_t p_{t+1} \dots$ along a trajectory, and because of the amplifying effects these weights can have on the traces, the variances of the traces iterates can grow unboundedly with time, severely affecting the behavior of the algorithms in practice. (The problem of growing variances when applying

17. It is not necessary to multiply the term $\phi(S_t)^\top \theta_t$ by p_t , and that version of the algorithm was the one given in (Bertsekas and Yu, 2009; Yu, 2012). The experimental results in (Dann et al., 2014) suggest to us that each version can have less variance than the other in some occasions, however. As far as convergence analysis is concerned, the two versions are essentially the same and the analyses given in (Yu, 2012, 2015a) and this paper indeed apply simultaneously to both versions of the algorithm.

importance sampling to simulate Markov systems was also known earlier and discussed in prior works; see e.g., Glynn and Iglehart, 1989; Randhawa and Juneja, 2004.) The two biased constrained algorithms discussed in this paper were motivated by the need to mitigate the variance problem, and their robust behavior has been observed in our experiments (Mahmood et al., 2015; Yu, 2016). However, beyond simply constraining the iterates, more variance reduction techniques are needed, such as control variates (Randhawa and Juneja, 2004; Ahamed et al., 2006) and weighted importance sampling (Precup et al., 2000, 2001; Mahmood et al., 2014; Mahmood and Sutton, 2015). To overcome the variance problem in off-policy learning, further research is required.

Regarding convergence analysis of ETD(λ), the results we gave in (2015a) and this paper concern only the convergence properties and not the rates of convergence. For on-policy TD(λ) and LSTD(λ), convergence rate analyses are available (Konda, 2002, Chap. 6). Such analyses in the off-policy case will give us better understanding of the asymptotic behavior of the off-policy algorithms. Finally, besides asymptotic behavior of the algorithms, their finite-time or finite-sample properties (such as those considered by Munos and Szepesvári, 2008; Antos et al., 2008; Lazaric et al., 2012; Liu et al., 2015), and their large deviations properties are also worth studying.

Acknowledgments

I thank Professors Richard Sutton and Csaba Szepesvári for helpful discussions, and I thank the anonymous reviewers for their helpful feedback. This research was supported by a grant from Alberta Innovates—Technology Futures.

Appendix A. Key Properties of Trace Iterates

In this appendix we list four key properties of trace iterates $\{(e_t, F_t)\}$ generated by the ETD(λ) algorithm. Three of them were derived in (Yu, 2015a, Appendix A), and used in the convergence analysis of ETD(λ) in both (Yu, 2015a) and the present paper.

As discussed in Section 3.2, $\{(e_t, F_t)\}$ can have unbounded variances and is naturally unbounded in common off-policy situations. However, as the proposition below shows, $\{(e_t, F_t)\}$ is bounded in a stochastic sense.

Proposition 11 *Under Assumption 1, given a bounded set $E \subset \mathbb{R}^{n+1}$, there exists a constant $L < \infty$ such that if the initial $(e_0, F_0) \in E$, then $\sup_{t \geq 0} \mathbb{E}[\|(e_t, F_t)\|] < L$.*

The preceding proposition is the same as (Yu, 2015a, Prop. A.1) except that the conclusion is for all the initial (e_0, F_0) from the set E , instead of a fixed initial (e_0, F_0) . By making explicit the dependence of the constant L on the initial (e_0, F_0) , the same proof of (Yu, 2015a, Prop. A.1) (which is a relatively straightforward calculation) applies to the preceding proposition.

We note that Prop. 11 does not imply the *uniform integrability* of $\{(e_t, F_t)\}$ —this stronger property does hold for the trace iterates, as we proved in Prop. 2(i), Section 4.1.2. (The latter and its proof focus on $\{e_t\}$ only, but the same argument applies to $\{(e_t, F_t)\}$.)

The next proposition concerns the change in the trace iterates due to the change in its initial condition. It is the same as (Yu, 2015a, Prop. A.2); its proof is more involved than the proofs of the two other properties of the trace iterates and uses, among others, a theorem for nonnegative random processes (Neveu, 1975). We did not use this proposition directly in the analysis of the present paper, but it is important in establishing that the Markov chain $\{Z_t\}$ has a unique invariant probability measure (Theorem 2, Section 2.4), which the results of the present paper rely on. In addition, it is helpful for understanding the behavior of the trace iterates.

Let (\hat{e}_t, \hat{F}_t) , $t \geq 1$, be defined by the same recursion (3)-(5) that defines (e_t, F_t) , using the same state and action random variables $\{(S_t, A_t)\}$, but with a different initial condition (\hat{e}_0, \hat{F}_0) . We write a zero vector in any Euclidean space as $\mathbf{0}$.

Proposition 12 *Under Assumption 1, for any two given initial conditions (e_0, F_0) and (\hat{e}_0, \hat{F}_0) ,*

$$F_t - \hat{F}_t \xrightarrow{a.s.} \mathbf{0}, \quad e_t - \hat{e}_t \xrightarrow{a.s.} \mathbf{0}.$$

The third proposition below concerns approximating the trace iterates (e_t, F_t) by truncated traces that depend on a fixed number of the most recent states and actions only. First, let us express the traces (e_t, F_t) , by using their definitions (cf. Equations 3-5), as

$$F_t = F_0 \cdot (\rho_0 \gamma_1 \cdots \rho_{t-1} \gamma_t) + \sum_{k=1}^t i(S_k) \cdot (\rho_k \gamma_{k+1} \cdots \rho_{t-1} \gamma_t), \quad (78)$$

$$e_t = e_0 \cdot (\beta_1 \cdots \beta_t) + \sum_{k=1}^t M_k \cdot \phi(S_k) \cdot (\beta_{k+1} \cdots \beta_t), \quad (79)$$

where $\beta_k = \rho_{k-1} \gamma_k \lambda_k$ and

$$M_k = \lambda_k i(S_k) + (1 - \lambda_k) F_k.$$

For each integer $K \geq 1$, the truncated traces $(\tilde{e}_{t,K}, \tilde{F}_{t,K})$ are defined by limiting the summations in (78)-(79) to be over $K+1$ terms only as follows:

$$(\tilde{e}_{t,K}, \tilde{F}_{t,K}) = (e_t, F_t) \quad \text{for } t \leq K,$$

and for $t \geq K+1$,

$$\tilde{F}_{t,K} = \sum_{k=t-K}^t i(S_k) \cdot (\rho_k \gamma_{k+1} \cdots \rho_{t-1} \gamma_t), \quad (80)$$

$$\tilde{M}_{t,K} = \lambda_t i(S_t) + (1 - \lambda_t) \tilde{F}_{t,K}, \quad (81)$$

$$\tilde{e}_{t,K} = \sum_{k=t-K}^t \tilde{M}_{k,K} \cdot \phi(S_k) \cdot (\beta_{k+1} \cdots \beta_t). \quad (82)$$

We have the following approximation property for truncated traces, in which the notation “ $L_K \downarrow 0$ ” means that L_K decreases monotonically to 0 as $K \rightarrow \infty$.

Proposition 13 *Let Assumption 1 hold. Given a bounded set $E \subset \mathbb{R}^{n+1}$, there exist constants $L_K, K \geq 1$, with $L_K \downarrow 0$ as $K \rightarrow \infty$, such that if the initial $(e_0, F_0) \in E$, then*

$$\sup_{t \geq 0} E \left[\left\| (e_t, F_t) - (\bar{e}_{t,K}, \bar{F}_{t,K}) \right\| \right] \leq L_K.$$

The preceding proposition is the same as (Yu, 2015a, Prop. A.3(i)), except that the initial (e_0, F_0) can be from a bounded set E instead of being fixed. The proof given in (Yu, 2015a) applies here as well, similar to the case of Prop. 11. This proposition about truncated traces was used in (Yu, 2015a) to obtain the convergence in mean given in Theorem 3 (Section 2.4) and allowed us to work with simple finite-space Markov chains, instead of working with the infinite-space Markov chain $\{Z_t\}$ directly, in that proof. In the present paper, it has expedited our proofs of Props. 2-3 (Section 4.1.2) regarding the uniform integrability and convergence in mean conditions for constrained ETD(λ).

Finally, the uniform integrability of $\{(e_t, F_t)\}$ (proved in Prop. 2(i) in this paper, as already mentioned) is important both for convergence analysis and for understanding the behavior of the trace iterates.

References

- T. P. Ahamed, V. S. Borkar, and S. Juneja. Adaptive importance sampling technique for Markov chains using stochastic approximation. *Operations Research*, 54:489–504, 2006.
- A. Antos, C. Szepesvári, and R. Munos. Learning near-optimal policies with Bellman residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71:89–129, 2008.
- L. C. Baird. Residual algorithms: Reinforcement learning with function approximation. In *The 13th International Conference on Machine Learning (ICML)*, 1995.
- D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA, 1996.
- D. P. Bertsekas and H. Yu. Projected equation methods for approximate solution of large linear systems. *Journal of Computational and Applied Mathematics*, 227(1):27–50, 2009.
- P. Billingsley. *Convergence of Probability Measures*. John Wiley & Sons, New York, 1968.
- V. S. Borkar. *Stochastic Approximation: A Dynamic Viewpoint*. Cambridge University Press, Cambridge, 2008.
- J. A. Boyan. Least-squares temporal difference learning. In *The 16th International Conference on Machine Learning (ICML)*, 1999.
- S. J. Bradtke and A. G. Barto. Linear least-squares algorithms for temporal difference learning. *Machine Learning*, 22(2):33–57, 1996.
- C. Dann, G. Neumann, and J. Peters. Policy evaluation with temporal differences: A survey and comparison. *Journal of Machine Learning Research*, 15:809–883, 2014.
- J. L. Doob. *Stochastic Processes*. John Wiley & Sons, New York, 1953.
- R. M. Dudley. *Real Analysis and Probability*. Cambridge University Press, Cambridge, 2002.
- M. Geist and B. Scherrer. Off-policy learning with eligibility traces: A survey. *Journal of Machine Learning Research*, 15:289–333, 2014.
- P. W. Glynn and D. L. Iglehart. Importance sampling for stochastic simulations. *Management Science*, 35:1367–1392, 1989.
- A. Hallak, A. Tamar, R. Munos, and S. Mannor. Generalized emphatic temporal difference learning: Bias-variance analysis. In *The 30th AAAI Conference on Artificial Intelligence*, 2016.
- V. R. Konda. *Actor-Critic Algorithms*. PhD thesis, MIT, 2002.
- H. J. Kushner and D. S. Clark. *Stochastic Approximation Methods for Constrained and Unconstrained Systems*. Springer-Verlag, New York, 1978.
- H. J. Kushner and A. Schwartz. Weak convergence and asymptotic properties of adaptive filters with constant gains. *IEEE Transactions on Information Theory*, 30:177–182, 1984.
- H. J. Kushner and G. G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer-Verlag, New York, 2nd edition, 2003.
- A. Lazaric, M. Ghavamzadeh, and R. Munos. Finite-sample analysis of least-squares policy iteration. *Journal of Machine Learning Research*, 13:3041–3074, 2012.
- B. Liu, S. Mahadevan, and J. Liu. Regularized off-policy TD-learning. In *Advances in Neural Information Processing Systems (NIPS)* 22, 2009.
- B. Liu, J. Liu, M. Ghavamzadeh, S. Mahadevan, and M. Petrik. Finite-sample analysis of proximal gradient TD algorithms. In *The 31st Conference on Uncertainty in Artificial Intelligence (UAI)*, 2015.
- H. R. Maei. *Gradient Temporal-Difference Learning Algorithms*. PhD thesis, University of Alberta, 2011.
- S. Mahadevan and B. Liu. Sparse Q-learning with mirror descent. In *The 28th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2012.
- S. Mahadevan, B. Liu, P. Thomas, W. Dabney, S. Giguere, N. Jacek, I. Gemp, and J. Liu. Proximal reinforcement learning: A new theory of sequential decision making in primal-dual spaces, 2014. arXiv:1405.6757.
- A. R. Mahmood and R. S. Sutton. Off-policy learning based on weighted importance sampling with linear computational complexity. In *The 31st Conference on Uncertainty in Artificial Intelligence (UAI)*, 2015.

- A. R. Mahmood, H. van Hasselt, and R. S. Sutton. Weighted importance sampling for off-policy learning with linear function approximation. In *Advances in Neural Information Processing Systems (NIPS)* 27, 2014.
- A. R. Mahmood, H. Yu, M. White, and R. S. Sutton. Emphatic temporal-difference learning. In *European Workshops on Reinforcement Learning (EWRLL)*, 2015.
- S. Meyn. Ergodic theorems for discrete time stochastic systems using a stochastic Lyapunov function. *SIAM Journal on Control and Optimization*, 27:1409–1439, 1989.
- S. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Cambridge University Press, Cambridge, 2nd edition, 2009.
- R. Munos and C. Szepesvári. Finite time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9:815–857, 2008.
- J. Neveu. *Discrete-Parameter Martingales*. North-Holland, Amsterdam, 1975.
- B. A. Pires and C. Szepesvári. Statistical linear estimation with penalized estimators: An application to reinforcement learning. In *The 29th International Conference on Machine Learning (ICML)*, 2012.
- B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30:838–855, 1992.
- D. Precup, R. S. Sutton, and S. Singh. Eligibility traces for off-policy policy evaluation. In *The 17th International Conference on Machine Learning (ICML)*, 2000.
- D. Precup, R. S. Sutton, and S. Daggupta. Off-policy temporal-difference learning with function approximation. In *The 18th International Conference on Machine Learning (ICML)*, 2001.
- M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, New York, 1994.
- R. S. Randhawa and S. Juneja. Combining importance sampling and temporal difference control variates to simulate Markov chains. *ACM Transactions on Modeling and Computer Simulation*, 14(1):1–30, 2004.
- Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, 2nd edition, 2003.
- B. Scherrer. Should one compute the temporal difference fix point or minimize the Bellman residual? The unified oblique projection view. In *The 27th International Conference on Machine Learning (ICML)*, 2010.
- R. S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 1988.
- R. S. Sutton. TD models: Modeling the world at a mixture of time scales. In *The 12th International Conference on Machine Learning (ICML)*, 1995.
- R. S. Sutton. The grand challenge of predictive empirical abstract knowledge. In *IJCAI Workshop on Grand Challenges for Reasoning from Experiences*, 2009.
- R. S. Sutton and A. G. Barto. *Reinforcement Learning*. MIT Press, Cambridge, MA, 1998.
- R. S. Sutton, C. Szepesvári, and H. Maei. A convergent $O(n)$ algorithm for off-policy temporal-difference learning with linear function approximation. In *Advances in Neural Information Processing Systems (NIPS)* 21, 2008.
- R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *The 26th International Conference on Machine Learning (ICML)*, 2009.
- R. S. Sutton, A. R. Mahmood, and M. White. An emphatic approach to the problem of off-policy temporal-difference learning. *Journal of Machine Learning Research*, 17(73):1–29, 2016.
- J. N. Tsitsiklis and B. Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, 1997.
- T. Ueno, S. Maeda, M. Kawanabe, and S. Ishii. Generalized TD learning. *Journal of Machine Learning Research*, 12:1977–2020, 2011.
- H. S. Yao and Z. Q. Liu. Preconditioned temporal difference learning. In *The 25th International Conference on Machine Learning (ICML)*, 2008.
- H. Yu. Least squares temporal difference methods: An analysis under general conditions. *SIAM Journal on Control and Optimization*, 50:3310–3343, 2012.
- H. Yu. On convergence of emphatic temporal-difference learning, 2015a. <http://arxiv.org/abs/1506.02582>; a shorter version appeared in *The 28th Annual Conference on Learning Theory (COLT)*, 2015.
- H. Yu. Weak convergence properties of constrained emphatic temporal-difference learning with constant and slowly diminishing stepsize, 2015b. <http://arxiv.org/abs/1511.07471>.
- H. Yu. Some simulation results for emphatic temporal-difference learning algorithms, 2016. <http://arxiv.org/abs/1605.02099>.
- H. Yu and D. P. Bertsekas. Error bounds for approximations from projected linear equations. *Mathematics of Operations Research*, 35(2):306–329, 2010.
- H. Yu and D. P. Bertsekas. Weighted Bellman equations and their applications in approximate dynamic programming. LIDS Technical Report 2876, MIT, 2012.

RLScore: Regularized Least-Squares Learners

Tapio Pahikkala
Antti Airola

*Department of Information Technology
20014 University of Turku
Finland*

TAPIO.PAHIKKALA@UTU.FI
ANTTI.AIROLA@UTU.FI

Editor: Alexandre Gramfort

Abstract

RLScore is a Python open source module for kernel based machine learning. The library provides implementations of several regularized least-squares (RLS) type of learners. RLS methods for regression and classification, ranking, greedy feature selection, multi-task and zero-shot learning, and unsupervised classification are included. Matrix algebra based computational short-cuts are used to ensure efficiency of both training and cross-validation.

A simple API and extensive tutorials allow for easy use of RLScore.

Keywords: cross-validation, feature selection, kernel methods, Kronecker product kernel, pair-input learning, python, regularized least-squares

1. Introduction

RLScore implements learning algorithms based on minimizing the regularized risk functional

$$\operatorname{argmin}_{f \in \mathcal{H}} R(f) + \lambda \|f\|^2,$$

where f is the learned predictor, \mathcal{H} a reproducing kernel Hilbert space of functions, $R(f)$ the empirical risk, $\|f\|^2$ the regularizer, and $\lambda > 0$ a regularization parameter.

Regularized least-squares¹ (RLS) is the classical method resulting from the choice $R(f) = \sum_{i=1}^n (f(\mathbf{x}_i) - y_i)^2$. The method admits a closed form solution, leading to efficient algorithms for leave-one-out cross-validation (LOO), multi-target learning, and fast selection of regularization parameter (Rifkin and Lippert, 2007). For example, the LOO predictions can be obtained essentially for free as the sideproduct of computations needed for training the method once. Previously, these methods have been implemented in libraries such as GURLS (Tacchetti et al., 2013) and Python scikit-learn (Pedregosa et al., 2011).

In the recent years, research in RLS methods has led to the development of a large variety of new efficient algorithms, that analogously to the classical RLS methods offer unique computational benefits both for training and model selection. These include leave-pair-out and leave-group-out cross-validation, methods for feature selection, ranking and unsupervised classification, as well as pair-input learning methods with applications to interaction prediction, cold start recommendations and zero-shot learning. RLScore is a Python module that provides a simple high-level interface to a library of highly optimized implementations of these methods.

1. aka kernel ridge regression, least-squares support vector machine

2. Implemented Algorithms

RLScore implements a large variety of fast holdout and CV algorithms. A fast leave-group-out (LGO) CV (Pahikkala et al., 2012b), where folds containing multiple instances are left out, is provided, complementing the classical fast RLS LOO algorithm (also included) (Rifkin and Lippert, 2007). The approach allows implementing fast K-fold CV, and more importantly, implementing CV for non i.i.d. data with natural group structure. Typical examples include leave-query-out CV for learning to rank, leave-sentence-out or leave-document out CV in text mining, leave-image-out CV for object recognition etc. Further, a leave-pair-out (LPO) algorithm (Pahikkala et al., 2009), that corresponds to leaving each combination of two instances (or a subset of these) from the data out in turn, is provided. LPO can be used to compute an almost unbiased estimate of area under ROC curve (Airola et al., 2011) and its generalization, the pairwise ranking accuracy.

RankRLS method implements efficient algorithms for both minimizing pairwise ranking losses and computing cross-validation estimates for ranking. The method has been shown to be highly competitive compared to ranking support vector machines (Pahikkala et al., 2009). Unsupervised variants of RLS classification inspired by the maximum margin clustering approach have also been developed (Pahikkala et al., 2012a).

Greedy RLS extends the basic RLS to learning sparse linear models in linear time, combining fast update formulas for feature addition and LOO with a greedy search (Naula et al., 2014). The computational short cuts allow scaling the approach to genome wide studies with hundreds of thousands of features. The method produced the winning submission of sub-challenge 3 of 2014 Broad-DREAM Gene Essentiality Prediction Challenge due to its ability to select a minimal accurate subset of features for multi-task learning problems.

RLS methods allow also fast learning from pair-input data. Applications include protein-protein and drug-target interaction prediction (Pahikkala et al., 2015), forecasting winners of two-player games, collaborative filtering, learning to rank for information retrieval etc. When making predictions for new pairs unseen in training set, the setting has natural applications in transfer and zero-shot learning. In the kernel methods framework this can be expressed as a learning problem where objects from two domains have their own kernel functions, and their joint kernel is the Kronecker product kernel. Efficient training and cross-validation algorithms for this setting have been recently derived (see e.g. Pahikkala et al. 2013; Stock et al. 2016).

3. Software Package

RLScore is implemented as a Python module that depends on NumPy (van der Walt et al., 2011) for basic data structures and linear algebra, SciPy (Jones et al., 2001-) for sparse matrices and optimization methods, and Cython (Behnel et al., 2011) for implementing low-level routines in C-language. The aim of the software is to provide high quality implementations of algorithms developed by the authors that combine efficient training with automated performance evaluation and model selection methods.

RLScore implements a modular design, where data representation, learning algorithms, and prediction are separated from each other where possible. The most basic kernel-based learning methods operate on a singular value decomposition of the data produced by an

adapter object. The hypothesis space used depends on the adapter, choices include both linear and kernel feature spaces, as well as Nyström type of reduced-set approximation. After training, the adapter creates a suitable type of linear or kernel predictor. The predictor object can be used or saved to disk independently of the algorithm used to train it.

The API design has been influenced by common Python data analysis environments such as NumPy, SciPy and scikit-learn, making it easy to combine RLScore with existing data analysis pipelines. The most fundamental classes in RLScore are learner objects in module `rlscore.learner`. At initialization, a learner is trained, and function `predict` is used for prediction. The predictor object can also be directly accessed and used independently of the learner. The majority of the learners also implement fast holdout and cross-validation functions, and support kernels, and fast multi-target learning. Unit tests are used to verify the implementations. Extensive tutorials describe how RLScore can be used to solve different types of problems. Listing 1 presents a simple demonstration of the interface.

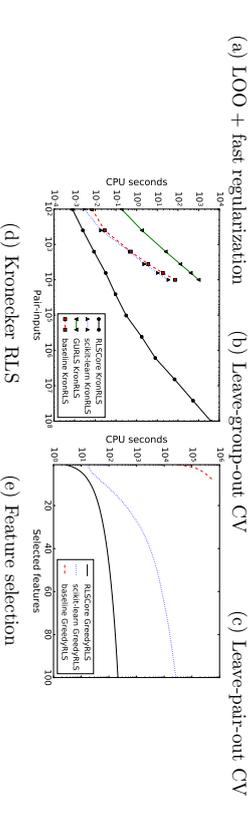
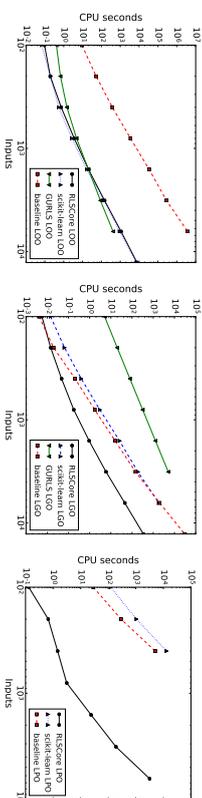
Listing 1: feature selection with greedy RLS algorithm

```
import numpy as np
from rlscore.learner import GreedyRLS
from scipy.stats import kendalltau

#regression problem with 3 important features
X = np.random.randn(100, 20)
y = X[:, 0] + X[:, 2] - X[:, 5] + 0.1*np.random.randn(100)
#select 3 features with greedy RLS
rls = GreedyRLS(X[:50], y[:50], regparam=1, subsetsize=3)
#Did we select the right features?
print(rls.selected)
#Compute test set predictions
p = rls.predict(X[50:])
print(kendalltau(y[50:], p))
```

4. Benchmarks

Here we demonstrate the advantages of RLScore solvers on five benchmark tasks. Each of the considered tasks can be expressed either as a single or a sequence of RLS problems with closed form solutions. The *baseline* method solves each resulting system $(\mathbf{K} + \lambda \mathbf{D})\mathbf{A} = \mathbf{Y}$ or $(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})\mathbf{W} = \mathbf{X}^T \mathbf{Y}$ with Python `numpy.linalg.solve` that calls the LAPACK `gesv` routine. Further we compare to two existing RLS solvers implemented in Python `scikit-learn` (version 0.18) (Pedregosa et al., 2011) and the MATLAB GURLS package (Tacheteli et al., 2013). The RLScore algorithms produce exactly the same results as the compared methods, but make use of a number of computational short-cuts resulting in substantial increases in efficiency. GURLS results were not included for LPO and feature selection as the runtimes were impractically long. Benchmark codes for comparing RLScore and scikit-learn RLS implementations are included in the RLScore code repository.



- (a) Leave-one-out CV and regularization parameter selection (parameter grid $\{2^{-15}, \dots, 2^{15}\}$, linear kernel, equal number of instances and features). Both scikit-learn and GURLS also implement fast LOO and regularization.
- (b) Leave-group-out CV, 10 instances per fold, Gaussian kernel, 500 features.
- (c) Leave-pair-out CV, Gaussian kernel, 500 features.
- (d) Kronecker product kernel $\mathbf{K} \otimes \mathbf{G}$ is a popular choice in pair-input learning. **KronRLS** allows learning with the kernel without explicitly forming the pairwise kernel matrix. We generate two kernel matrices of size $n \times n$, the label vector \mathbf{Y} contains n^2 entries, one label for each pair. Baseline explicitly constructs the $n^2 \times n^2$ -sized kernel matrix.
- (e) Learning sparse models. We consider greedy forward selection, where on each iteration one selects the feature whose addition provides the lowest RLS LOO error. **GreedyRLS** implements this procedure in linear time, with scikit-learn we use the fast LOO algorithm, baseline is a pure wrapper implementation. Data matrix \mathbf{X} contains 10000 instances and 1000 features, and the number of outputs in \mathbf{Y} is 10.

RLScore scales to orders of magnitude larger problem sizes than the baselines on all but the LOO experiment. With the exception of LOO, none of the considered fast algorithms are available in other software implementations. RLScore contains also a large variety of other methods, with new ones being added with each release.

Acknowledgments

We would like to acknowledge the support from the Academy of Finland (grants 134020 and 289903) and the co-authors who participated in developing the implemented algorithms.

References

- A. Airola, Tapio Pahikkala, Willem Waegeman, Bernard De Baets, and Tapio Salakoski. An experimental comparison of cross-validation techniques for estimating the area under the ROC curve. *Computational Statistics & Data Analysis*, 55(4):1828–1844, 2011.
- S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D.S. Seljebotn, and K. Smith. Cython: The best of both worlds. *Computing in Science Engineering*, 13(2):31–39, 2011.
- E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. URL <http://www.scipy.org/>.
- P. Naula, A. Airola, T. Salakoski, and T. Pahikkala. Multi-label learning under feature extraction budgets. *Pattern Recognition Letters*, 40:56–65, 2014.
- T. Pahikkala, E. Tsivtsivadze, A. Airola, J. Järvinen, and J. Boberg. An efficient algorithm for learning to rank from preference graphs. *Machine Learning*, 75(1):129–165, 2009.
- T. Pahikkala, A. Airola, F. Gieseke, and O. Kramer. Unsupervised multi-class regularized least-squares classification. In *IEEE International Conference on Data Mining*, pages 585–594. IEEE Computer Society, 2012a.
- T. Pahikkala, H. Suominen, and J. Boberg. Efficient cross-validation for kernelized least-squares regression with sparse basis expansions. *Machine Learning*, 87(3):381–407, 2012b.
- T. Pahikkala, A. Airola, M. Stock, B. De Baets, and W. Waegeman. Efficient regularized least-squares algorithms for conditional ranking on relational data. *Machine Learning*, 93(2–3):321–356, 2013.
- Tapio Pahikkala, Antti Airola, Sami Pietilä, Sushil Shakyaawar, Agnieszka Sz wajda, Jing Tang, and Tero Aittokallio. Toward more realistic drug-target interaction predictions. *Briefings in Bioinformatics*, 16(2):325–337, 2015.
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- R. Rifkin and R. Lippert. Notes on regularized least squares. Technical Report MIT-CSAIL-TR-2007-025, Massachusetts Institute of Technology, Cambridge, USA, 2007.
- Michiel Stock, Tapio Pahikkala, Antti Airola, Bernard De Baets, and Willem Waegeman. Efficient pairwise learning using kernel ridge regression: an exact two-step method. *CoRR*, abs/1606.04275, 2016.
- A. Tacchetti, P. Mallapragada, M. Santoro, and L. Rosasco. GURLS: A least squares library for supervised learning. *Journal of Machine Learning Research*, 14(1):3201–3205, 2013.
- S. van der Walt, S. C. Colbert, and G. Varoquaux. The numpy array: A structure for efficient numerical computation. *Computing in Science Engineering*, 13(2):22–30, 2011.

Stability and Generalization in Structured Prediction

Ben London

University of Maryland

Bert Huang

Virginia Tech

Lise Gettoor

University of California, Santa Cruz

BLONDON@CS.UMD.EDU

BHUANG@VT.EDU

GETTOOR@SOE.UCSF.EDU

Editor: John Shawe-Taylor

Abstract

Structured prediction models have been found to learn effectively from a few large examples—sometimes even just one. Despite empirical evidence, canonical learning theory cannot guarantee generalization in this setting because the error bounds decrease as a function of the number of examples. We therefore propose new PAC-Bayesian generalization bounds for structured prediction that decrease as a function of both the number of examples and the size of each example. Our analysis hinges on the stability of joint inference and the smoothness of the data distribution. We apply our bounds to several common learning scenarios, including max-margin and soft-max training of Markov random fields. Under certain conditions, the resulting error bounds can be far more optimistic than previous results and can even guarantee generalization from a single large example.

Keywords: structured prediction, learning theory, PAC-Bayes, generalization bounds

1. Introduction

Many important applications of machine learning require making multiple interdependent predictions whose dependence relationships form a graph. In some cases, the number of inputs and outputs can be enormous. For instance, in natural language processing, a document may contain thousands of words to be assigned a part-of-speech tag; in computer vision, a digital image may contain millions of pixels to be segmented; and in social network analysis, a relational graph may contain millions of users to be categorized. Obtaining fully annotated examples can be time-consuming and expensive, due to the number of variables. It is therefore common to train a structured predictor on far fewer examples than are used in the unstructured setting. In the extreme (yet not atypical) case, the training set consists of a single example, with large internal structure. A central question in statistical learning theory is *generalization*; that is, whether the expected error at test time will be reasonably close to the empirical error measured during training. Canonical learning-theoretic results for structured prediction (e.g., Taskar et al., 2004; Bartlett et al., 2005; McAllester, 2007) only guarantee generalization when the number of training examples is high. Yet, this pessimism contradicts a wealth of experimental results (e.g., Taskar et al., 2002; Tsochantzidis et al., 2005), which indicate that training on a few large examples is sufficient. In this work,

we address the question of when generalization is possible in this setting. We derive new generalization bounds for structured prediction that are far more optimistic than previous results. When sufficient conditions hold, our bounds guarantee generalization from a few large examples—even just one.

The intuition behind our analysis is motivated by a common practice known alternately as *templating* or *parameter-tying*. At a high level, templating shares parameters across substructures (e.g., nodes, edges, etc.) with identical local structure. (Templating is explained in detail in Section 2.2.2.) Originally proposed for relational learning as a way of dealing with non-uniformly-structured examples, templating has an additional benefit in that it effectively limits the complexity of the hypothesis class by reducing the number of parameters to be learned. Each instance of a substructure within an example acts as a kind of “micro example” of a template. Since each example may contain many micro examples, it is plausible that generalization could occur from even a single example.

Part of the difficulty when formalizing this intuition is that the micro examples are interdependent. Like all statistical arguments, generalization bounds must show that the empirical error concentrates around the expected error, and analyzing the concentration of functions of dependent random variables is nontrivial. Moreover, inference in a structured predictor is typically formulated as a global optimization over all outputs simultaneously. Due to model-induced dependencies, changes to one input may affect many of the outputs, which affects the loss differently than in binary or multiclass prediction. Thus, this problem cannot be viewed as simply learning from interdependent data, which has been studied extensively (e.g., Usumier et al., 2006; Mohri and Rostamizadeh, 2010; Ralaivola et al., 2010).

We therefore have two obstacles: the dependence in the data distribution and the dependence induced by the predictor. We characterize the former dependence using concepts from measure concentration theory (Kontorovich and Ramanan, 2008), and we view the latter dependence through the lens of algorithmic *stability*. Unlike previous literature (e.g., Bousquet and Elisseeff, 2002), we are not interested in the stability of the learning algorithm; rather, we examine the stability of inference (more specifically, a functional of the predictions) with respect to perturbations of the input. In prior work (London et al., 2013, 2014), we used the term *collective stability* to describe the stability of the predictions, guaranteeing collective stability for predictors whose inference objectives are strongly convex. In this work, we propose a form of stability that generalizes collective stability by analyzing the loss function directly. Our new definition accommodates a broader range of loss functions and predictors, and eliminates our previous reliance on strong convexity. Moreover, we support functions that are *locally* stable over some subset of their domain, and random functions that are stable with high probability.

This probabilistic notion of stability lends itself nicely to the *PAC-Bayes* framework, in which prediction proceeds by drawing a random hypothesis from a distribution on the hypothesis class. For this and other technical reasons, we use PAC-Bayesian analysis to derive our generalization bounds. When certain conditions are met by the distributions on the data and hypothesis class, our bounds can be as tight as $\tilde{O}(1/\sqrt{mn})$, where m is the number of examples, and n is the size of each example. Note that this expression decreases as either m or n increase. This rate is much tighter than previous results, which only

guarantee $\tilde{O}(1/\sqrt{m})$. From our bounds, we conclude that it is indeed possible to generalize from a few large examples—potentially even just one.

1.1 Related Work

One of the earliest explorations of generalization in structured prediction is by Collins (2001), who developed risk bounds for language parsers using various classical tools, such as the Vapnik-Chervonenkis dimension and margin theory. In Taskar et al.’s (2004) landmark paper on max-margin Markov networks, the authors use covering numbers to derive risk bounds for their proposed class of models. Bartlett et al. (2005) improved this result using PAC-Bayesian analysis.¹ McAllester (2007; 2011) provided a comprehensive PAC-Bayesian study of various structured losses and learning algorithms. Recently, Hazan et al. (2013) proposed a PAC-Bayes bound with a form often attributed to Catoni (2007), which can be minimized directly using gradient descent. Gigante et al. (2013) used PAC-Bayesian analysis to derive risk bounds for the kernel regression approach to structured prediction. In a similar vein as the above literature, yet taking a significantly different approach, Bradley and Gnestrin (2012) derived finite sample complexity bounds for learning conditional random fields using the composite likelihood estimator.

All of the above works have approached the problem from the traditional viewpoint, that the generalization error should decrease proportionally to the number of examples. In a previous publication (London et al., 2013), we proposed the first bounds that decrease with both the number of examples and the size of each example (given suitably weak dependence within each example). We later refined these results using PAC-Bayesian analysis (London et al., 2014). Our current work builds upon this foundation to derive similarly optimistic generalization bounds, while accommodating a broader range of loss functions and hypothesis classes.

From a certain perspective, our work fits into a large body of literature on learning from various types of interdependent data. Most of this is devoted to “unstructured” prediction. Usui et al. (2006) and Ralavola et al. (2010) used concepts from graph coloring to analyze generalization in learning problems that induce a dependency graph, such as bipartite ranking. In this case, the training data contains dependencies, but prediction is localized to each input-output pair. Similarly, Mohri and Rostamizadeh (2009, 2010) derived risk bounds for ϕ -mixing and β -mixing temporal data, using an “independent blocking” technique due to Yu (1994). The hypotheses they consider predict each time step independently, which makes independent blocking possible. Since we are interested in hypotheses (and loss functions) that perform joint inference, which may not decompose over the outputs, we cannot employ techniques such as graph coloring and independent blocking.

A related area of research is learning to forecast time series data. In this setting, the goal is to predict the next (or, some future) value in the series, given a moving window of previous observations. The generalization error of time series forecasting has been studied extensively by McDonald et al. (e.g., 2012) in the β -mixing regime. Similarly, Alquier and Wintenburger (2012) derived oracle inequalities for ϕ -mixing conditions.

¹ PAC-Bayesian analysis is often accredited to McAllester (1998, 1999), and has been refined by a number of authors (e.g., Herbrich and Graepel, 2001; Langford and Shawe-Taylor, 2002; Seeger, 2002; Ambrožek and Catoni, 2006; Catoni, 2007; Germain et al., 2009; Lever et al., 2010; Seldin et al., 2012).

The idea of learning from one example is related to the “one-network” learning paradigm, in which data is generated by a (possibly infinite) random field, with certain labels observed for training. The underlying model is estimated from the partially observed network, and the learned model is used to predict the missing labels, typically with some form of joint inference. Xiang and Neville (2011) examined maximum likelihood and pseudo-likelihood estimation in this setting, proving that are asymptotically consistent. Note that this is a *transductive* setting, in that the network data is fixed (i.e., realized), so the learned hypothesis is not expected to generalize to other network data. In contrast, we analyze *inductive* learning, wherein the model is applied to future draws from a distribution over network data.

Connections between stability and generalization have been explored in various forms. Bousquet and Elisseeff (2002) proposed the stability of a learning algorithm as a tool for analyzing generalization error. Wainwright (2006) analyzed the stability of marginal probabilities in variational inference, identifying the relationship between stability and strong convexity (similar to our work in London et al., 2013, 2014). He used this result to show that an *inconsistent* estimator, which uses approximate inference during training, can asymptotically yield lower regret (relative to the optimal Bayes least squares estimator) than using the *true* model with approximate inference. Honorio (2011) showed that the Bayes error rate of various graphical models is related to the stability of their log-likelihood functions with respect to changes in the model parameters.

1.2 Our Contributions

Our primary contribution is a new PAC-Bayesian analysis of structured prediction, producing generalization bounds that decrease when either the number of examples, m , or the size of each example, n , increase. Under suitable conditions, our bounds can be as tight as $\tilde{O}(1/\sqrt{mn})$. Our results apply to any composition of loss function and hypothesis class that satisfies our local stability conditions, which includes a broad range of modeling regimes used in practice. We also propose a novel view of PAC-Bayesian “derandomization,” based on the principle of stability, which provides a general proof technique for converting a generalization bound for a randomized structured predictor into a bound for a deterministic structured predictor.

As part of our analysis, we derive a new bound on the moment-generating function of a locally stable functional. The tightness of this bound (hence, our generalization bounds) hinges on a measure of the aggregate dependence between the random variables within each example. Our bounds are meaningful when the dependence is sub-logarithmic in the number of variables. We provide two examples of stochastic processes for which this condition holds. These results, and their implications for measure concentration, are of independent interest.

We apply our PAC-Bayes bounds to several common learning scenarios, including *margin* and *soft-margin* training of (conditional) Markov random fields. To demonstrate the benefit of local stability analysis, we also consider a specific generative process that induces unbounded stability in certain predictors, given certain inputs. These examples suggest several factors to be considered when modeling structured data, in order to obtain the best generalization rate: (1) templating is crucial; (2) the norm of the parameters contributes to the stability of inference, and should be controlled via regularization; and (3) limiting

local interactions in the model can improve stability, hence, generalization. All of these considerations can be summarized by the classic tension between representational power and overfitting, applied to the structured setting. Most importantly, these examples confirm that generalization from limited training examples is indeed possible for many structured prediction techniques used in practice.

1.3 Organization

The remainder of this paper is organized as follows. Section 2 introduces the notation used throughout the paper and reviews some background in structured prediction, templated Markov random fields, generalization error and PAC-Bayesian analysis. In Section 3, we propose general properties that characterize the local stability of a generic functional (e.g., the composition of a loss function and hypothesis). In Section 4, we introduce the statistical quantities and inequalities used in our analysis, as well as some examples of “nice” dependence conditions. Section 5 presents our main results: new PAC-Bayes bounds for structured prediction. We also propose a general proof technique for derandomizing the bounds using stability. In Section 6, we apply our bounds to a number of common learning scenarios. Specifically, we examine learning templated Markov random fields in the max-margin and soft-max frameworks, under various assumptions about the data distribution. Section 7 concludes our study with a discussion of the results and their implications for practitioners of structured prediction.

2. Preliminaries

This section introduces the notation and background used in this paper. We begin with notational conventions. We then formally define structured prediction and review some background on templated Markov random fields, a general class of probabilistic graphical models commonly used in structured prediction. Finally, we review the concept of generalization and discuss the PAC-Bayes framework, which we use to state our main results.

2.1 Notational Conventions

Let $\mathcal{X} \subseteq \mathbb{R}^k$ denote a domain of observations, and let \mathcal{Y} denote a finite set of discrete labels. Let $\mathcal{Z} \triangleq \mathcal{X} \times \mathcal{Y}$ denote the cross product of the two, representing input-output pairs.

Let $\mathbf{Z} \triangleq (Z_i)_{i=1}^n$ denote a set of n random variables, with joint distribution \mathbb{D} on a sample space \mathcal{Z}^n . We denote *realizations* of \mathbf{Z} by $\mathbf{z} \in \mathcal{Z}^n$. We use $\Pr_{\mathbf{Z} \sim \mathbb{D}}\{\cdot\}$ to denote the probability of an event over realizations of \mathbf{Z} , distributed according to \mathbb{D} . Similarly, we use $\mathbb{E}_{\mathbf{Z} \sim \mathbb{D}}[\cdot]$ to specify an expectation over \mathbf{Z} . When it is clear from context which variable(s) and distribution the probability (or expectation) is taken over, we may omit the subscript notation. We will occasionally employ the shorthand $\mathbb{D}(\mathcal{S})$ to denote the measure of a subset $\mathcal{S} \subseteq \mathcal{Z}^n$ under \mathbb{D} ; i.e., $\mathbb{D}(\mathcal{S}) = \Pr_{\mathbf{Z} \sim \mathbb{D}}\{\mathbf{Z} \in \mathcal{S}\}$. With a slight abuse of notation, which should be clear from context, we also use $\mathbb{D}(\mathbf{Z}_{i,j} | E)$ to denote the distribution of some subset of the variables, (Z_i, \dots, Z_j) , conditioned on an event, E .

For a graph $G \triangleq (\mathcal{V}, \mathcal{E})$, with nodes \mathcal{V} and edges \mathcal{E} , we use $|G| \triangleq |\mathcal{V}| + |\mathcal{E}|$ to denote the total number of nodes and edges in G .

2.2 Structured Prediction

At its core, *structured prediction* (sometimes referred to as *structured output prediction* or *structured learning*) is about learning concepts that have a natural internal structure. In the framework we consider, each example of a concept contains n interdependent random variables, $\mathbf{Z} \triangleq (Z_i)_{i=1}^n$, with joint distribution \mathbb{D} . Each $Z_i \triangleq (X_i, Y_i)$ is an input-output pair, taking values in $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$.² Each example is associated with an implicit dependency graph, $G \triangleq (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} \triangleq \{1, \dots, n\}$ indexes \mathbf{Z} , and \mathcal{E} captures the dependencies in \mathbf{Z} . Unless otherwise stated, assume that the edge structure is given *a priori*. The edge structure may be obvious from context, or may be inferred beforehand. To simplify our analysis, we assume that each example uses the same structure.

The prediction task is to infer $\mathbf{Y} \triangleq (Y_i)_{i=1}^n$, conditioned on $\mathbf{X} \triangleq (X_i)_{i=1}^n$. A *hypothesis*, h , maps \mathcal{X}^n to \mathcal{Y}^n , using some internal parametric representation that incorporates the structure of the problem (an example of which is given in Section 2.2.1). We are interested in hypotheses that perform joint reasoning over all variables simultaneously. We therefore assume that computing $h(\mathbf{X})$ implicitly involves a global optimization that does not decompose over the outputs, due to dependencies.

2.2.1 MARKOV RANDOM FIELDS

To better understand structured prediction, it will help to consider a specific hypothesis class. One popular class is that of *Markov random fields* (MRFs), a broad family of undirected graphical models that generalizes many models used in practice, such as relational Markov networks (Taskar et al., 2002), conditional random fields (Lafferty et al., 2001), and Markov logic networks (Richardson and Domingos, 2006). In this section, we review some background on MRFs.

Recall that each example is associated with a dependency graph, $G \triangleq (\mathcal{V}, \mathcal{E})$. We assume that the edge set is undirected. This does not limit the applicability of our analysis, since there exists a straight-forward conversion from directed models (Koller and Friedman, 2009). The parameters of an MRF are organized according to the *cliques* (i.e., complete subgraphs), \mathcal{C} , contained in G . For each clique, $c \in \mathcal{C}$, we associate a real-valued *potential* function, $\theta_c(\mathbf{y} | \mathbf{x}; \mathbf{w})$, parameterized by a vector of weights, $\mathbf{w} \in \mathbb{R}^d$, for some $d \geq 1$. This function indicates the score for \mathbf{Y}_c being in state \mathbf{y}_c , conditioned on the observation $\mathbf{X} = \mathbf{x}$. The potentials define a log-linear conditional probability distribution,

$$p(\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x}; \mathbf{w}) \triangleq \exp \left(\sum_{c \in \mathcal{C}} \theta_c(\mathbf{y} | \mathbf{x}; \mathbf{w}) - \Phi(\mathbf{x}; \mathbf{w}) \right),$$

where

$$\Phi(\mathbf{x}; \mathbf{w}) \triangleq \ln \sum_{\mathbf{y} \in \mathcal{Y}^n} \exp \left(\sum_{c \in \mathcal{C}} \theta_c(\mathbf{y} | \mathbf{x}; \mathbf{w}) \right)$$

is a normalizing function known as the *log-partition*.

For convenience, we represent the label space, \mathcal{Y} , by the set of $|\mathcal{Y}|$ standard basis (i.e., “one-hot”) vectors, $\mathbf{e}_1, \dots, \mathbf{e}_{|\mathcal{Y}|}$. Thus, the joint state of a clique, c , is represented by a

2. To minimize bookkeeping, we have assumed a one-to-one correspondence between input and output variables, and that the Z_i variables have identical domains, but these assumptions can be relaxed.

vector, $\mathbf{y}_c = \bigotimes_{i \in \mathcal{C}} y_i$, of length $|\mathcal{C}| \triangleq |\mathcal{Y}|^{|\mathcal{C}|}$. With a slight abuse of notation, we overload the potential functions so that $\theta_c(\mathbf{x}; \mathbf{w}) \in \mathbb{R}^{|\mathcal{C}|}$ denotes a vector of potentials, and

$$\theta_c(\mathbf{y} | \mathbf{x}; \mathbf{w}) = \theta_c(\mathbf{x}; \mathbf{w}) \cdot \mathbf{y}_c.$$

Thus, with

$$\theta(\mathbf{x}; \mathbf{w}) \triangleq (\theta_c(\mathbf{x}; \mathbf{w}))_{c \in \mathcal{C}} \quad \text{and} \quad \hat{\mathbf{y}} \triangleq (\hat{y}_c)_{c \in \mathcal{C}},$$

we have that

$$\sum_{c \in \mathcal{C}} \theta_c(\mathbf{y} | \mathbf{x}; \mathbf{w}) = \theta(\mathbf{x}; \mathbf{w}) \cdot \hat{\mathbf{y}}.$$

We refer to $\hat{\mathbf{y}}$ as the *full* representation of \mathbf{y} .

The canonical inference problems for MRFs are *maximum a posteriori* (MAP) inference, which computes the mode of the distribution,

$$\arg \max_{\mathbf{y} \in \mathcal{Y}^n} p(\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x}; \mathbf{w}),$$

and *marginal* inference, which computes the marginal distribution of a subset of the variables. In general, both tasks are intractable—MAP inference is NP-hard and marginal inference is $\#\text{-P-hard}$ (Roth, 1996)—though there are some useful special cases for which inference is tractable, and many approximation algorithms for the general case. In this work, we assume that an efficient (approximate) inference algorithm is given.

2.2.2 TEMPLATING

An important property of the above construction is that the same vector of weights, \mathbf{w} , is used to parameterize all of the potential functions. One could imagine that \mathbf{w} contains a unique subvector, \mathbf{w}_c , for every clique. However, one could also bin the cliques by a set of *templates*—such as singletons (nodes), pairs (edges) or triangles (hyperedges)—then use the same weights for each template. This technique is alternatively referred to as *templating* or *parameter-typing*.

With templating, one can define general inductive rules to reason about datasets of arbitrary size and structure. Because of this flexibility, templating is used in many *relational* models, such as relational Markov networks (Taskar et al., 2002), relational dependency networks (Neville and Jensen, 2004), and Markov logic networks (Richardson and Domingos, 2006).

A templated model implicitly assumes that all *groundings* (i.e., instances) of a template should be modeled identically, meaning location within the graph is irrelevant. A non-templated model is location-aware and therefore has higher representational power. However, without templating, the dimensionality of \mathbf{w} scales with the number of cliques; whereas, with templating, the dimensionality of \mathbf{w} is constant. Thus, we find the classic tension between representational power and overfitting. To mitigate overfitting, one must restrict model complexity. Yet, too little expressivity will hamper predictive performance. This consideration is critical to the application of our generalization bounds.

In practice, templated models typically consist of unary and pairwise templates. We refer to these as pairwise models. Higher-order templates (i.e., cliques of three or more)

can capture certain inductive rules that pairwise models cannot. For example, for a binary relation r , the transitive closure $r(A, B) \wedge r(B, C) \implies r(A, C)$ requires triadic templates. Rules like this are sometimes used for link prediction and entity resolution. Of course, this additional expressivity comes at a cost, as will become apparent later.

2.2.3 DEFINING THE POTENTIAL FUNCTIONS

In many applications of MRFs, the potentials are defined as multilinear functions of $(\mathbf{w}, \mathbf{x}, \mathbf{y})$. For example, assuming each node i has local observations $x_i \in \mathcal{X}$ and label $y_i \in \mathcal{Y}$, we can define a vector of local *features*,

$$f_i(\mathbf{x}, \mathbf{y}) \triangleq x_i \otimes y_i,$$

using the Kronecker product (since y_i is a standard basis vector). Similarly, for each edge $\{i, j\} \in \mathcal{E}$, let

$$f_{ij}(\mathbf{x}, \mathbf{y}) \triangleq \frac{1}{2} \begin{bmatrix} x_i \\ x_j \end{bmatrix} \otimes (y_i \otimes y_j).$$

Here, we have defined the edge features using a concatenation of the local observations, though this need not be the case. In general, the edge features can be arbitrary functions of the observations, such as kernels or similarity functions. Or, we could eschew the observations altogether and just use $y_i \otimes y_j$, which is typical in practice.

The potential functions are then defined as weighted feature functions. For the following, we will assume that the weights are templated, as described in Section 2.2.2. For each node, we associate a set of singleton weights, $\mathbf{w}_s \in \mathbb{R}^{d_s}$, and for each edge, a set of pairwise weights, $\mathbf{w}_p \in \mathbb{R}^{d_p}$, where d_s and d_p denote the respective lengths of the node and edge features. Then,

$$\theta_i(\mathbf{y} | \mathbf{x}; \mathbf{w}) \triangleq \mathbf{w}_s \cdot f_i(\mathbf{x}, \mathbf{y}) \quad \text{and} \quad \theta_{ij}(\mathbf{y} | \mathbf{x}; \mathbf{w}) \triangleq \mathbf{w}_p \cdot f_{ij}(\mathbf{x}, \mathbf{y});$$

and, with

$$\mathbf{w} \triangleq \begin{bmatrix} \mathbf{w}_s \\ \mathbf{w}_p \end{bmatrix} \quad \text{and} \quad \mathbf{f}(\mathbf{x}, \mathbf{y}) \triangleq \begin{bmatrix} \sum_{i \in \mathcal{V}} f_i(\mathbf{x}, \mathbf{y}) \\ \sum_{\{i, j\} \in \mathcal{E}} f_{ij}(\mathbf{x}, \mathbf{y}) \end{bmatrix},$$

we have that

$$\theta(\mathbf{x}; \mathbf{w}) \cdot \hat{\mathbf{y}} = \mathbf{w} \cdot \mathbf{f}(\mathbf{x}, \mathbf{y}).$$

In Section 6, we apply our generalization bounds to the above construction of a templated MRF, consisting of singleton and pairwise linear potentials (with or without edge features).

2.3 Learning and Generalization

Given a set of m training examples, $\hat{\mathbf{Z}} \triangleq (\mathbf{Z}^{(i)})_{i=1}^m$, drawn independently and identically from \mathbb{D} , the goal of learning is to produce a hypothesis from a specified class, denoted $\mathcal{H} \subseteq \{h : \mathcal{X}^n \rightarrow \mathcal{Y}^n\}$. We do not assume that the data is generated according to some target concept in \mathcal{H} , so \mathcal{H} may be misspecified.

Hypotheses are evaluated using a *loss function* of the form $L : \mathcal{H} \times \mathcal{X}^n \rightarrow \mathbb{R}_+$, which may have access to the internal representation of the hypothesis. For a given loss function,

L , let $\bar{L}(h) \triangleq \mathbb{E}_{\mathbf{Z} \sim \mathbb{D}}[L(h, \mathbf{Z})]$ denote the expected loss over realizations of an example. This quantity, known as the *risk*, corresponds to the error h will incur on future predictions. Let

$$\hat{L}(h, \hat{\mathbf{Z}}) \triangleq \frac{1}{m} \sum_{l=1}^m L(h, \mathbf{Z}^{(l)})$$

denote the average loss on the training set, $\hat{\mathbf{Z}}$. Most learning algorithms minimize (an upper bound on) $\hat{L}(h, \hat{\mathbf{Z}})$, since it is an *empirical* estimate of the risk.

The goal of our analysis is to upper-bound the difference of the expected and empirical risks, $\bar{L}(h) - \hat{L}(h, \hat{\mathbf{Z}})$ —which we refer to as the *generalization error*³—thereby yielding an upper bound on the risk. As is typically done in generalization analysis, we show that, with high probability over draws of a training set, the generalization error is upper-bounded by a function of certain properties of the domain, hypothesis class and learning algorithm, which decreases as the (effective) size of the training set increases. Note that small generalization error does not necessarily imply small risk, since the empirical risk may be large. Nonetheless, small generalization error implies that the empirical risk will be a good estimate of the risk, thus motivating empirical risk minimization.

2.3.1 PAC-BAYES

PAC-Bayes is a framework for analyzing the risk of a randomized predictor. One begins by fixing a *prior* distribution, \mathbb{P} , on the hypothesis space, \mathcal{H} . Then, given some training data, one constructs a *posterior* distribution, \mathbb{Q} , the parameters of which are typically learned from the training data. For example, when \mathcal{H} is a subset of Euclidean space, a common PAC-Bayesian construction is a standard multivariate Gaussian prior with an isotropic Gaussian posterior, centered at the learned hypothesis. To make a prediction on an input, \mathbf{x} , one draws a hypothesis, $h \in \mathcal{H}$, according to \mathbb{Q} , then computes $h(\mathbf{x})$.

Since prediction is randomized, the risk quantities are defined over draws of h , which we denote by

$$\hat{L}(\mathbb{Q}, \hat{\mathbf{Z}}) \triangleq \mathbb{E}_{h \sim \mathbb{Q}}[\hat{L}(h, \hat{\mathbf{Z}})] \quad \text{and} \quad \bar{L}(\mathbb{Q}) \triangleq \mathbb{E}_{h \sim \mathbb{Q}}[\bar{L}(h)].$$

The goal of PAC-Bayesian analysis is to upper-bound some measure of discrepancy between these quantities. The discrepancy is sometimes defined as the KL divergence between error rates, or the squared difference. In this work, we upper-bound the difference, $\bar{L}(\mathbb{Q}) - \hat{L}(\mathbb{Q}, \hat{\mathbf{Z}})$, which is the PAC-Bayesian analog of the generalization error.

3. Stability

A key component of our analysis is the *stability* of the loss function. In this section, we introduce some definitions of stability and relate them to other forms found in the literature. Broadly speaking, stability ensures that changes to the input result in proportional changes in the output. In structured prediction, where inference is typically a global optimization over many interdependent variables, changing any single observation may affect many of the inferred values. The structured loss functions we consider *implicitly* require some form

³ Our definition of generalization error differs from some literature, in which the term is used to refer to the expected loss.

of joint inference; therefore, their stability is nontrivial. In this chapter, we introduce some definitions of stability and relate them to other forms found in the literature.

The following definitions will make use of the *Hamming distance*. For vectors $\mathbf{z}, \mathbf{z}' \in \mathcal{Z}^n$, denote their Hamming distance by

$$D_H(\mathbf{z}, \mathbf{z}') \triangleq \sum_{i=1}^n \mathbf{1}\{z_i \neq z'_i\}.$$

3.1 Uniform and Local Stability

Throughout this section, let $\mathcal{F} \triangleq \{\varphi : \mathcal{Z}^n \rightarrow \mathbb{R}\}$ denote an arbitrary class of functionals; e.g., \mathcal{F} could be a structured loss function composed with a class of hypotheses. The definitions in this section describe notions of stability that hold either *uniformly* over the domain of \mathcal{F} (for each $\varphi \in \mathcal{F}$), or *locally* over some subset of the domain (for some subset of \mathcal{F}).

Definition 1. We say that a function $\varphi \in \mathcal{F}$ is *β -uniformly stable* if, for any inputs $\mathbf{z}, \mathbf{z}' \in \mathcal{Z}^n$,

$$|\varphi(\mathbf{z}) - \varphi(\mathbf{z}')| \leq \beta D_H(\mathbf{z}, \mathbf{z}'). \quad (1)$$

Similarly, the class \mathcal{F} is *β -uniformly stable* if every $\varphi \in \mathcal{F}$ is β -uniformly stable.

Equation 1 means that the change in the output should be proportional to the Hamming distance between the inputs. Put differently, a uniformly stable function is Lipschitz under the Hamming norm.

Uniform stability over the entire domain can be a strong requirement. Sometimes, stability only holds for a certain subset of inputs, such as points contained in a Euclidean ball of a certain radius. We refer to the set of inputs for which stability holds as the “good” set; all other inputs are “bad.” The precise meaning of good and bad depends on the hypothesis class. Given some delineation of good and bad, we obtain the following localized notion of stability.

Definition 2. For a subset $\mathcal{B}_Z \subseteq \mathcal{Z}^n$, we say that a function $\varphi \in \mathcal{F}$ is *(β, \mathcal{B}_Z) -locally stable* if Equation 1 holds for all $\mathbf{z}, \mathbf{z}' \notin \mathcal{B}_Z$. The class \mathcal{F} is *(β, \mathcal{B}_Z) -locally stable* if every $\varphi \in \mathcal{F}$ is (β, \mathcal{B}_Z) -locally stable.

Definition 2 has an alternate probabilistic interpretation. If \mathbb{D} is a distribution on \mathcal{Z}^n , then Equation 1 holds with some probability over draws of $\mathbf{z}, \mathbf{z}' \sim \mathbb{D}$. If the bad set \mathcal{B}_Z has measure $\mathbb{D}(\mathcal{B}_Z) \leq \nu$, then (β, \mathcal{B}_Z) -local stability is similar to, though slightly weaker than, the *strongly difference-bounded* property proposed by Kutin (2002). If φ is strongly difference-bounded, then Equation 1 must hold for any $\mathbf{z} \notin \mathcal{B}_Z$ and $\mathbf{z}' \in \mathcal{Z}^n$ (which could be in \mathcal{B}_Z). All functions that are strongly difference-bounded are locally stable, but the converse is not true.

The notion of probabilistic stability can be extended to distributions on the function class. For any stability parameter β (and bad inputs \mathcal{B}_Z), the function class is partitioned into functions that satisfy Equation 1, and those that do not. Therefore, for any distribution \mathbb{Q} on \mathcal{F} , uniform (or local) stability holds with some probability over draws of $\varphi \sim \mathbb{Q}$. This idea motivates the following definition.

Definition 3. Fix some $\beta \geq 0$ and $\mathcal{B}_Z \subseteq \mathcal{Z}^n$, and let $\mathcal{B}_F \subseteq \mathcal{F}$ denote the subset of functions that are not (β, \mathcal{B}_Z) -locally stable. We say that a distribution \mathbb{Q} on \mathcal{F} is $(\beta, \mathcal{B}_Z, \eta)$ -locally stable if $\mathbb{Q}(\mathcal{B}_F) \leq \eta$.

Note the taxonomical relationship between these definitions. Definition 1 is the strongest condition, since it implies Definitions 2 and 3. Clearly, if \mathcal{F} is β -uniformly stable, then it is (β, \emptyset) -locally and $(\beta, \emptyset, 0)$ -locally stable. Definition 2 extends Definition 1 by accommodating broader domains. Definition 3 extends this even further, by accommodating classes in which only some functions satisfy local stability.

Definition 3 is particularly interesting in the PAC-Bayes framework, in which a predictor is selected at random according to a (learned) posterior distribution. With prior knowledge of the hypothesis class (and data distribution), a posterior can be constructed so as to place low mass on predictors that do not satisfy uniform or local stability. As we show in Section 6, this technique lets us relax certain restrictions on the hypothesis class.

Stability measures the change in the output relative to the change in the inputs. A related property is that the change in the output is bounded—i.e., the function has bounded range.

Definition 4. We say that $\varphi \in \mathcal{F}$ is α -uniformly range-bounded if, for any $\mathbf{z}, \mathbf{z}' \in \mathcal{Z}^n$,

$$|\varphi(\mathbf{z}) - \varphi(\mathbf{z}')| \leq \alpha.$$

Range-boundedness is implied by stability, but the range constant, α , may be smaller than the upper bound implied by stability. Our analysis uses range-boundedness as a fall-back property when “good” stability does not hold.

3.2 Connections to Other Notions of Stability

In the learning theory literature, the word “stability” has traditionally been associated with a learning algorithm, rather than an inference algorithm. A learning algorithm is said to be stable with respect to a loss function if the loss of a learned hypothesis varies by a bounded amount upon replacing (or deleting) examples from the training set. This property has been used to derive generalization bounds (e.g., Bousquet and Elisseeff, 2002), similar to the way we use stability of inference. The key idea is that stability enables concentration of measure, which is central to generalization. That said, learning stability is distinct from inference stability, and neither property implies the other. Indeed, a learning algorithm might return hypotheses with drastically different losses for slightly different training sets, even if each hypothesis, composed with the loss function, is uniformly stable. Likewise, a stable learning algorithm might produce hypotheses with unstable loss.

Our definition of stability should also be contrasted with *sensitivity analysis*. Since the terms are often used interchangeably, we distinguish the two as follows: stability measures the amount of change induced in the *output* of a function upon perturbing its input within a certain range, and sensitivity analysis measures the amount of perturbation one can apply to the *input* such that its output remains within a certain range. By these definitions, one is the dual of the other. In the context of probabilistic inference, sensitivity analysis has been used to determine the maximum amount one can perturb the model parameters (or evidence) such that the likelihood of a query stays within a given tolerance, or such that

the most likely assignment does not change (Chan and Darwiche, 2005, 2006). Stability measures *how much* the likelihood or most likely assignment changes.

Our first generalization bounds for structured prediction (London et al., 2013) crucially relied on a property we referred to as *uniform collective stability*. A class of vector-valued functions, $\mathcal{G} \triangleq \{g : \mathcal{Z}^n \rightarrow \mathbb{R}^N\}$, has β -uniform collective stability if, for any $g \in \mathcal{G}$, and any $\mathbf{z}, \mathbf{z}' \in \mathcal{Z}^n$,

$$\|g(\mathbf{z}) - g(\mathbf{z}')\|_1 \leq \beta D_{\text{H}}(\mathbf{z}, \mathbf{z}').$$

We later relaxed this requirement to various non-uniform definitions of collective stability (London et al., 2014). Because collective stability implicitly involves the maximizing argument of a high-dimensional global optimization (i.e., a vector of predictions), we restricted our previous analyses to predictors with strongly convex inference objectives. Strong convexity let us bound the collective stability of a predictor; hence, the stability of its *output* composed with an *admissible*⁴ loss function. Our new definitions involve the output of a *functional* (i.e., a scalar-valued function of multiple inputs), which essentially means that we are interested in the stability of the loss, instead of the collective stability of the predictions. In our new analysis, the loss function has access to the model and may use it for inference. However, the loss function may not require the same inference used for prediction. (For example, the losses considered in Section 6 use the maximum of the inference objective instead of the maximizing argument.) This framework lets us analyze a broad range of structured losses, without requiring strongly convex inference. Further, it can be shown that any predictor with “good” collective stability (such as one with a strongly convex inference objective), composed with an admissible loss function, satisfies our new definitions of stability. Therefore, our new definitions are strictly more general than collective stability.

4. Statistical Tools

Reasoning about the concentration of functions of dependent random variables requires sophisticated statistical machinery. In this section, we review some supporting definitions and introduce a quantity to summarize the amount of dependence in the data distribution. We use this quantity in a new moment-generating function inequality for locally stable functions of dependent random variables. We then provide some example conditions under which dependence is suitably bounded, thereby supporting improved generalization bounds.

4.1 Quantifying Dependence

For probability measures \mathbb{P} and \mathbb{Q} on a σ -algebra (i.e., event space) Σ , the *total variation distance* is

$$\|\mathbb{P} - \mathbb{Q}\|_{\text{TV}} \triangleq \sup_{A \in \Sigma} |\mathbb{P}(A) - \mathbb{Q}(A)|.$$

As a special case, when the sample space, Ω , is finite,

$$\|\mathbb{P} - \mathbb{Q}\|_{\text{TV}} = \frac{1}{2} \sum_{\omega \in \Omega} |\mathbb{P}(\omega) - \mathbb{Q}(\omega)|.$$

4. See (London et al., 2013, 2014) for a precise definition of loss admissibility.

Let π be a permutation of $[n] \triangleq \{1, \dots, n\}$, where $\pi(i)$ denotes the i^{th} element in the sequence and $\pi(i : j)$ denotes a subsequence of elements i through j . Used to index variables $\mathbf{Z} \triangleq (Z_i)_{i=1}^n$, denote by $Z_{\pi(i)}$ the i^{th} variable in the permutation and $\mathbf{Z}_{\pi(i:j)}$ the subsequence $(Z_{\pi(i)}, \dots, Z_{\pi(j)})$.

Definition 5. We say that a sequence of permutations, $\boldsymbol{\pi} \triangleq (\pi_i)_{i=1}^n$, is a *filtration* if, for $i = 1, \dots, n-1$,

$$\pi_i(1 : i) = \pi_{i+1}(1 : i).$$

Let $\Pi(n)$ denote the set of all filtrations for a given n .

The following data structure quantifies the dependence between subsets of variables defined by a filtration.

Definition 6. Let $\mathbf{Z} \triangleq (Z_i)_{i=1}^n$ denote random variables with joint distribution \mathbb{D} on \mathcal{Z}^n . Fix a filtration, $\boldsymbol{\pi} \in \Pi(n)$, and a set of inputs, $\mathcal{B}_Z \subseteq \mathcal{Z}^n$. Let $\bar{\mathcal{B}}$ denote the event $\mathbf{Z} \notin \mathcal{B}_Z$. For $i \in [n]$, let $\mathcal{Z}_{\boldsymbol{\pi}, \bar{\mathcal{B}}}^i$ denote the subset of \mathcal{Z}^i such that, for every $\mathbf{z} \in \mathcal{Z}_{\boldsymbol{\pi}, \bar{\mathcal{B}}}^i$, $\mathbf{Z}_{\pi_i(1:i)} = \mathbf{z}$ is consistent with $\bar{\mathcal{B}}$. With a slight abuse of notation, for $\mathbf{z} \in \mathcal{Z}_{\boldsymbol{\pi}, \bar{\mathcal{B}}}^{i-1}$, let $\mathcal{Z}_{\boldsymbol{\pi}, \bar{\mathcal{B}}}^i(\mathbf{z})$ denote the subset of \mathcal{Z} such that, for any $z \in \mathcal{Z}_{\boldsymbol{\pi}, \bar{\mathcal{B}}}^i(\mathbf{z})$, $\mathbf{Z}_{\pi_i(1:i)} = (\mathbf{z}, z)$ is consistent with $\bar{\mathcal{B}}$. Then, for $i \in [n]$, $j > i$, $\mathbf{z} \in \mathcal{Z}_{\boldsymbol{\pi}, \bar{\mathcal{B}}}^{i-1}$ and $z, z' \in \mathcal{Z}_{\boldsymbol{\pi}, \bar{\mathcal{B}}}^i(\mathbf{z})$, we define the *ϑ -mixing coefficients*⁵ as

$$\vartheta_{ij}^{\boldsymbol{\pi}}(\mathbf{z}, z, z') \triangleq \|\mathbb{D}(\mathbf{Z}_{\pi_i(j:n)} | \bar{\mathcal{B}}, \mathbf{Z}_{\pi_i(1:i)} = (\mathbf{z}, z)) - \mathbb{D}(\mathbf{Z}_{\pi_i(j:n)} | \bar{\mathcal{B}}, \mathbf{Z}_{\pi_i(1:i)} = (\mathbf{z}, z'))\|_{\text{TV}}.$$

We use these to define the upper-triangular *dependence matrix*, $\boldsymbol{\Gamma}_{\boldsymbol{\pi}} \in \mathbb{R}^{n \times n}$, with entries

$$\gamma_{ij}^{\boldsymbol{\pi}} \triangleq \begin{cases} 1 & \text{for } i = j, \\ \sup_{\mathbf{z} \in \mathcal{Z}_{\boldsymbol{\pi}, \bar{\mathcal{B}}}^{i-1}, z, z' \in \mathcal{Z}_{\boldsymbol{\pi}, \bar{\mathcal{B}}}^i(\mathbf{z})} \vartheta_{ij}^{\boldsymbol{\pi}}(\mathbf{z}, z, z') & \text{for } i < j, \\ 0 & \text{for } i > j. \end{cases}$$

When $\mathcal{B}_Z = \emptyset$, we simply omit the subscript notation.

Each ϑ -mixing coefficient measures the influence of some variable, $Z_{\pi_i(i)}$, on some subset, $\mathbf{Z}_{\pi_i(j:n)}$, given some assignment to the variables, $\mathbf{Z}_{\pi_i(1:i-1)}$, that preceded $Z_{\pi_i(i)}$ in the filtration; $\gamma_{ij}^{\boldsymbol{\pi}}$ measures the maximal influence of $Z_{\pi_i(i)}$ conditioned on any assignment to $\mathbf{Z}_{\pi_i(1:i-1)}$. Thus, to summarize the amount of dependence in the data distribution, we use the induced matrix infinity norm of $\boldsymbol{\Gamma}_{\boldsymbol{\pi}}$, denoted

$$\|\boldsymbol{\Gamma}_{\boldsymbol{\pi}}\|_{\infty} \triangleq \max_{i \in [n]} \sum_{j=1}^n |\gamma_{ij}^{\boldsymbol{\pi}}|,$$

which effectively measures the maximal aggregate influence of any single variable, given the filtration. Observe that, if (Z_1, \dots, Z_n) are mutually independent, then $\boldsymbol{\Gamma}_{\boldsymbol{\pi}}^{\boldsymbol{\pi}}$ is the identity matrix and $\|\boldsymbol{\Gamma}_{\boldsymbol{\pi}}\|_{\infty} = 1$. At the other extreme, if (Z_1, \dots, Z_n) are deterministically dependent, then the top row of $\boldsymbol{\Gamma}_{\boldsymbol{\pi}}^{\boldsymbol{\pi}}$ is $\mathbf{1}$, so $\|\boldsymbol{\Gamma}_{\boldsymbol{\pi}}\|_{\infty} = n$.

⁵ The ϑ -mixing coefficients were introduced by Kontorovich and Ramanan (2008) as *η -mixing* and are related to the *maximal coupling coefficients* used by Chazottes et al. (2007).

Viewed through the lens of stability, $\|\boldsymbol{\Gamma}_{\boldsymbol{\pi}}\|_{\infty}$ can be interpreted as measuring the stability of the data distribution. From this perspective, distributions with strong, long-range dependencies (when $\|\boldsymbol{\Gamma}_{\boldsymbol{\pi}}\|_{\infty}$ is big) are unstable, whereas distributions with weak, local dependence (when $\|\boldsymbol{\Gamma}_{\boldsymbol{\pi}}\|_{\infty}$ is small) are stable. Intuitively, the same can be said for inference in MRFs; potentials that emphasize interactions between adjacent variables create long-range dependencies, which causes instability, whereas potentials that emphasize local signal make adjacent variables more independent, which promotes stability. Thus, dependence and stability are two sides of the same coin.

The filtration used to define $\boldsymbol{\Gamma}_{\boldsymbol{\pi}}^{\boldsymbol{\pi}}$ can have a strong impact on $\|\boldsymbol{\Gamma}_{\boldsymbol{\pi}}\|_{\infty}$. Since we do not assume that \mathbf{Z} corresponds to a temporal process, there may not be an obvious ordering of the variables. However, the types of stochastic processes we are interested in are typically endowed with a topology. If the topology is a graph, the filtration can be determined by traversing the graph. For instance, for a Markov tree process, Kontorovich (2012) ordered the variables via a breadth-first traversal from the root; for an Ising model on a lattice, Chazottes et al. (2007) ordered the variables with a spiraling traversal from the origin. (Both of these examples used a static permutation of the variables, not a filtration.) If the filtration is determined by graph traversal, the ϑ -mixing coefficients can be viewed as measuring the strength of dependence as a function of graph distance. Viewed as such, $\|\boldsymbol{\Gamma}_{\boldsymbol{\pi}}\|_{\infty}$ effectively captures the slowest decay of dependence along any traversal from a given set of traversals.

The aforementioned works (Kontorovich, 2012; Chazottes et al., 2007) showed that, for Markov trees and grids, under suitable contraction or temperature regimes, $\|\boldsymbol{\Gamma}_{\boldsymbol{\pi}}\|_{\infty}$ is bounded independently of n (i.e., $\|\boldsymbol{\Gamma}_{\boldsymbol{\pi}}\|_{\infty} = \mathcal{O}(1)$). By exploiting filtrations, we can show that the same holds for Markov random fields of any bounded-degree structure, provided the distribution exhibits suitable mixing. We discuss these conditions in Section 4.3.

4.2 A Moment-Generating Function Inequality for Local Stability

With the supporting definitions in mind, we now present a new moment-generating function inequality for locally stable functions of dependent random variables. The proof is provided in Appendix A.3.

Proposition 1. Let $\mathbf{Z} \triangleq (Z_i)_{i=1}^n$ denote random variables with joint distribution \mathbb{D} on \mathcal{Z}^n . Fix a set of “bad” inputs, $\mathcal{B}_Z \subseteq \mathcal{Z}^n$, and let $\bar{\mathcal{B}}$ denote the event $\mathbf{Z} \notin \mathcal{B}_Z$. Let $\varphi : \mathcal{Z}^n \rightarrow \mathbb{R}$ denote a measurable function with (β, \mathcal{B}_Z) -local stability. Then, for any $\tau \in \mathbb{R}$ and filtration $\boldsymbol{\pi} \in \Pi(n)$,

$$\mathbb{E}_{\mathbf{Z} \sim \mathbb{D}} \left[e^{\tau(\varphi(\mathbf{Z}) - \mathbb{E}[\varphi(\mathbf{Z}) | \bar{\mathcal{B}}])} | \bar{\mathcal{B}} \right] \leq \exp \left(\frac{\tau^2}{8} n \beta^2 \|\boldsymbol{\Gamma}_{\boldsymbol{\pi}}\|_{\infty}^2 \right).$$

This bound yields a novel concentration inequality for uniformly stable functions of dependent random variables, which we discuss in Appendix A.4. Though we will not use this corollary in our analysis, it may be of independent interest.

Proposition 1 builds on work by Samson (2000), Chazottes et al. (2007) and Kontorovich and Ramanan (2008). Our analysis differs from theirs in that we accommodate functions that are not uniformly stable. In this respect, our analysis is similar to that of Kutin (2002) and Vu (2002), though these works assume independence between variables. Because we

allow interdependence—as well as other technical challenges, related to our definitions of local stability—we do not use the same proof techniques as the aforementioned works.

4.3 Bounded Dependence Conditions

The infinity norm of the dependency matrix has a trivial upper bound, $\|\mathbf{\Gamma}_{\mathcal{F}}^{\pi}\|_{\infty} \leq n$. However, we are interested in bounds that are sub-logarithmic in (or, even better, independent of) n . In this section, we describe some general settings in which $\|\mathbf{\Gamma}_{\mathcal{F}}^{\pi}\|_{\infty}$ has a nontrivial upper bound.

For the remainder of this section, let $\mathbf{Z} \triangleq (Z_i)_{i=1}^n$ denote random variables with joint distribution \mathbb{D} on \mathcal{Z}^n . Assume that \mathbb{D} is associated with a graph, $G \triangleq (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} \triangleq [n]$ indexes \mathbf{Z} .

We use the following notion of distance-based dependence. For simplicity of exposition, we assume that $\mathcal{B}_{\mathbf{Z}} = \emptyset$, so we can omit \mathcal{B} from the following notation.

Definition 7. For any two subsets, $\mathcal{S}, \mathcal{T} \subseteq \mathcal{V}$, we define their graph distance, $D_G(\mathcal{S}, \mathcal{T})$, as the length of the shortest path from any node in \mathcal{S} to any node in \mathcal{T} . We then define the distance-based ϑ -mixing coefficients as

$$\vartheta(k) \triangleq \sup_{\substack{\mathcal{S} \subseteq \mathcal{V}, i \in \mathcal{S} \\ \mathcal{T} \subseteq \mathcal{V} \setminus \mathcal{S}; D_G(i, \mathcal{T}) \geq k \\ z \in \mathcal{Z}^{|\mathcal{S}|-1}, z, z' \in \mathcal{Z}}} \left| \mathbb{D}(\mathbf{Z}_{\mathcal{T}} | \mathbf{Z}_{\mathcal{S}} = \mathbf{z}, Z_i = z) - \mathbb{D}(\mathbf{Z}_{\mathcal{T}} | \mathbf{Z}_{\mathcal{S}} = \mathbf{z}, Z_i = z') \right|_{\mathcal{V}^{\mathcal{T}}},$$

where $\vartheta(0) \triangleq 1$.

The distance-based ϑ -mixing coefficients upper-bound the maximum influence exerted by any subset of the variables on any other subset that is separated by graph distance at least k . The sequence $(\vartheta(0), \vartheta(1), \vartheta(2), \dots)$ roughly measures how dependence decays with graph distance. Note that $\vartheta(k)$ uniformly upper-bounds $\vartheta_{i_j}^{\pi}$ when $D_G(\pi_i(i), \pi_i(j) : n) \geq k$. Therefore, for each upper-triangular entry of $\mathbf{\Gamma}^{\pi}$, we have that

$$\gamma_{i_j}^{\pi} \leq \vartheta(D_G(\pi_i(i), \pi_i(j) : n)).$$

Using the distance-based ϑ -mixing coefficients, we now show that, for certain Markov random fields, it is possible to upper-bound $\|\mathbf{\Gamma}_{\mathcal{F}}^{\pi}\|_{\infty}$ independently of n .

Proposition 2. Suppose \mathbb{D} is defined by an MRF, such that its graph, G , has maximum degree Δ_G . For any positive constant $\epsilon > 0$, if \mathbb{D} admits a distance-based ϑ -mixing sequence such that, for all $k \geq 1$, $\vartheta(k) \leq (\Delta_G + \epsilon)^{-k}$, then there exists a filtration π such that

$$\|\mathbf{\Gamma}^{\pi}\|_{\infty} \leq 1 + \Delta_G/\epsilon.$$

The proof is provided in Appendix A.5.

Uniformly geometric distance-based ϑ -mixing may seem like a restrictive condition. However, our analysis is overly pessimistic, in that it ignores the structure of the MRF beyond simply the maximum degree of the graph. Further, it does not take advantage of the actual conditional independencies present in the distribution. Nevertheless, there is a natural interpretation of the above conditions that follows from considering the mixing coefficients at distance 1: for the immediate neighbors of a node—i.e., its Markov blanket—its

ϑ -mixing coefficient must be less than $1/\Delta_G$. This loosely means that the combination of all incoming influence must be less than 1, implying that there is sufficiently strong influence from local observations.

Another important setting is when the graph is a chain. Chain-structured stochastic processes (usually temporal) under various mixing assumptions have been well-studied (see Bradley, 2005 for a comprehensive survey). It can be shown that any contracting Markov chain has $\|\mathbf{\Gamma}^{\pi}\|_{\infty} = O(1)$ (Kontorovich, 2012). Here, we provide an alternate condition, using distance-based ϑ -mixing, under which the dependency matrix of a Markov chain has suitably low norm. The key property of a chain graph is that the number of nodes at distance k from any starting node is constant. We can therefore relax the assumption of geometric decay used in the previous result.

Proposition 3. Suppose \mathbb{D} is an undirected Markov chain (i.e., chain-structured MRF) of length n . For any constants $\epsilon > 0$ and $p \geq 1$, if \mathbb{D} admits a distance-based ϑ -mixing sequence such that, for all $k \geq 1$, $\vartheta(k) \leq \epsilon k^{-p}$, then there exists a filtration, π , such that

$$\|\mathbf{\Gamma}^{\pi}\|_{\infty} \leq \begin{cases} 1 + \epsilon(1 + \ln(n-1)) & \text{if } p = 1, \\ 1 + \epsilon\zeta(p) & \text{if } p > 1, \end{cases}$$

where $\zeta(p) \triangleq \sum_{j=1}^{\infty} j^{-p}$ is the Riemann zeta function.

The proof is provided in Appendix A.6.

For $p > 1$, the Riemann function converges to a constant. For example, $\zeta(2) = \pi^2/6 \approx 1.645$. However, even $p = 1$ yields a sufficiently low growth rate. In the following section, we prove generalization bounds of the form $O(\|\mathbf{\Gamma}^{\pi}\|_{\infty}/\sqrt{mn})$, which still converges if $\|\mathbf{\Gamma}^{\pi}\|_{\infty} = O(\ln n)$, albeit at a slower rate.

5. PAC-Bayes Bounds

We now present some new PAC-Bayes generalization bounds using the stability definitions from Section 3. The first theorem is stated for a given stability parameter, β . We then generalize this result to hold for all β simultaneously, meaning β can depend on the posterior. We conclude this section with a general technique for derandomizing the bounds based on stability.

It will help to begin with a high-level sketch of the analysis, which we specialize to various settings in Sections 5.1 and 5.2. It will help to view the composition of the loss function, L , and the hypothesis class \mathcal{H} , as a family of functions, $L \circ \mathcal{H} = \{L(h, \cdot) : h \in \mathcal{H}\}$. If \mathbb{Q} is a distribution on \mathcal{H} , it is also a distribution on $L \circ \mathcal{H}$. Each member of $L \circ \mathcal{H}$ is a random function, determined by the draw of $h \sim \mathbb{Q}$. Further, when $L(h, \cdot)$ is composed with a training set $\tilde{\mathbf{Z}} \sim \mathbb{D}^m$ in $\hat{L}(h, \cdot)$, the generalization error, $\bar{L}(h) - \hat{L}(h, \tilde{\mathbf{Z}})$, becomes a centered random variable. Part of our analysis involves bounding the moment-generating function of this random variable, and to do so requires the notions of stability from Section 3. The stability of $L(h, \cdot)$ is determined by h , so the “bad” members of $L \circ \mathcal{H}$ are in fact the “bad” hypotheses (for the given loss function).

Let $\mathbf{Z} \triangleq (\mathbf{Z}^{(j)})_{j=1}^m$ denote a training set of m structured examples, distributed according to \mathbb{D}^m . Fix some $\beta \geq 0$ and a set of bad inputs $\mathcal{B}_{\mathbf{Z}}$, with measure $\nu \triangleq \mathbb{D}(\mathcal{B}_{\mathbf{Z}})$. Implicitly, the

pair (β, \mathcal{B}_Z) fixes a set of hypotheses $\mathcal{B}_H \subseteq \mathcal{H}$ for which $L(h, \cdot)$ does not satisfy Equation 1 with $\beta \triangleq \beta/n$ and \mathcal{B}_Z . For the time being, \mathcal{B}_H is independent of \mathcal{Q} . Fix a prior \mathbb{P} and posterior \mathcal{Q} on \mathcal{H} . (We will later consider all posteriors.) We define a convenience function,

$$\tilde{\phi}(h, \hat{\mathbf{Z}}) \triangleq \begin{cases} \mathbb{E}_{\mathbf{Z} \sim \mathbb{D}} [L(h, \mathbf{Z}) | \bar{\mathcal{B}}] - \hat{L}(h, \hat{\mathbf{Z}}) & \text{if } h \notin \mathcal{B}_H, \\ 0 & \text{otherwise,} \end{cases}$$

where $\bar{\mathcal{B}}$ denotes the event $\mathbf{Z} \notin \mathcal{B}_Z$. First, for any uniformly bounded random variable, with $|X| \leq b$, and some event, \bar{E} ,

$$\mathbb{E}[X] = \mathbb{E}[X \mathbb{1}\{\bar{E}\}] + \mathbb{E}[X \mathbb{1}\{\bar{E}^c\}] \leq b \mathbb{P}_{\mathbb{P}}\{\bar{E}\} + \mathbb{E}[X \mathbb{1}\{\bar{E}^c\}].$$

We use this identity to show that, if $L \circ \mathcal{H}$ is α -uniformly range-bounded, and \mathcal{Q} is $(\beta/n, \mathcal{B}_Z, \eta)$ -locally stable, then

$$\bar{L}(\mathcal{Q}) - \hat{L}(\mathcal{Q}, \hat{\mathbf{Z}}) \leq \alpha\eta + \alpha\nu + \mathbb{E}_{h \sim \mathcal{Q}} [\tilde{\phi}(h, \hat{\mathbf{Z}})].$$

To bound the $\mathbb{E}_{h \sim \mathcal{Q}} [\tilde{\phi}(h, \hat{\mathbf{Z}})]$, we use Donsker and Varadhan's (1975) *change of measure* inequality.

Lemma 1. For any measurable function $\varphi : \Omega \rightarrow \mathbb{R}$, and any two distributions, \mathbb{P} and \mathcal{Q} , on Ω ,

$$\mathbb{E}_{\omega \sim \mathbb{P}} [\varphi(\omega)] \leq D_{\text{kl}}(\mathbb{P} \parallel \mathcal{Q}) + \ln \mathbb{E}_{\omega \sim \mathcal{Q}} [e^{\varphi(\omega)}].$$

(McAllester (2003) provides a straightforward proof.) Using Lemma 1, for any free parameter $u \geq 0$, we have that

$$\mathbb{E}_{h \sim \mathcal{Q}} [\tilde{\phi}(h, \hat{\mathbf{Z}})] \leq \frac{1}{u} \left(D_{\text{kl}}(\mathcal{Q} \parallel \mathbb{P}) + \ln \mathbb{E}_{h \sim \mathbb{P}} [e^{u\tilde{\phi}(h, \hat{\mathbf{Z}})}] \right).$$

Combining the above inequalities yields

$$\bar{L}(\mathcal{Q}) - \hat{L}(\mathcal{Q}, \hat{\mathbf{Z}}) \leq \alpha\eta + \alpha\nu + \frac{1}{u} \left(D_{\text{kl}}(\mathcal{Q} \parallel \mathbb{P}) + \ln \mathbb{E}_{h \sim \mathbb{P}} [e^{u\tilde{\phi}(h, \hat{\mathbf{Z}})}] \right).$$

The remainder of the analysis concerns how to bound $\mathbb{E}_{h \sim \mathbb{P}} [e^{u\tilde{\phi}(h, \hat{\mathbf{Z}})}]$ and how to optimize u . For the first task, we combine Markov's inequality with the moment-generating function bound from Section 4.2. Optimizing u takes some care, since we would like the bounds to hold simultaneously for all posteriors. We therefore adopt a discretization technique (Seldin et al., 2012) that approximately optimizes the bound for all posteriors. We use a similar technique to obtain bounds that hold for all β .

5.1 Fixed Stability Bounds

In the following theorem, we derive a new PAC-Bayes bound for posteriors with local stability, with β fixed. Fixing β means that the set of “bad” hypotheses is determined by the characteristics of the hypothesis class independently of the posterior.

Theorem 1. Fix $m \geq 1$, $n \geq 1$, $\pi \in \Pi(\eta)$, $\delta \in (0, 1)$, $\alpha \geq 0$ and $\beta \geq 0$. Fix a distribution, \mathbb{D} , on \mathcal{Z}^n . Fix a set of bad inputs, \mathcal{B}_Z , with $\nu \triangleq \mathbb{D}(\mathcal{B}_Z)$. Let $\Gamma_{\frac{\pi}{\beta}}$ denote the dependency matrix induced by \mathbb{D} , π and \mathcal{B}_Z . Fix a prior, \mathbb{P} , on a hypothesis class, \mathcal{H} . Fix a loss function, L , such that $L \circ \mathcal{H}$ is α -uniformly range-bounded. Then, with probability at least $1 - \delta - mv$ over realizations of a training set, $\mathbf{Z} \triangleq (\mathbf{Z}^{(l)})_{l=1}^m$, drawn according to \mathbb{D}^n , the following hold: 1) for all $l \in [m]$, $\mathbf{Z}^{(l)} \notin \mathcal{B}_Z$; 2) for all $\eta \in [0, 1]$ and posteriors \mathcal{Q} with $(\beta/n, \mathcal{B}_Z, \eta)$ -local stability,

$$\bar{L}(\mathcal{Q}) \leq \hat{L}(\mathcal{Q}, \hat{\mathbf{Z}}) + \alpha(\eta + \nu) + 2\beta \|\Gamma_{\frac{\pi}{\beta}}\|_{\infty} \sqrt{\frac{D_{\text{kl}}(\mathcal{Q} \parallel \mathbb{P}) + \ln \frac{2}{\delta}}{2mn}}. \quad (2)$$

To interpret the bound, suppose $\alpha = O(1)$, $\beta = O(1)$, and that the data distribution is weakly dependent, with $\|\Gamma_{\frac{\pi}{\beta}}\|_{\infty} = O(1)$. We would then have that the generalization error decreases at a rate of $O(\eta + \nu + (mn)^{-1/2})$. Since η is a function of the posterior, we can reasonably assume that $\eta = O((mn)^{-1/2})$. (Section 6 provides examples of this.) However, while ν may be proportional to n , it is unreasonable to believe that ν will decrease with m , since \mathbb{D} is almost certainly agnostic to the number of training examples. Thus, Theorem 1 is interesting when either ν is negligible, or when m is a small constant.

It can be shown that any hypothesis class with *collective* stability, composed with a suitable loss function, satisfies the conditions of the bound. Thus, Theorem 1 is strictly more general than our prior PAC-Bayes bounds (London et al., 2014). Moreover, Theorem 1 easily applies to compositions with uniform stability, since $\mathcal{Q}(\mathcal{B}_H) = 0$ for all posteriors. This insight yields the following corollary.

Corollary 1. Suppose $L \circ \mathcal{H}$ is (β/n) -uniformly stable. Then, with probability at least $1 - \delta$ over realizations of \mathbf{Z} , for all \mathcal{Q} ,

$$\bar{L}(\mathcal{Q}) \leq \hat{L}(\mathcal{Q}, \hat{\mathbf{Z}}) + 2\beta \|\Gamma_{\frac{\pi}{\beta}}\|_{\infty} \sqrt{\frac{D_{\text{kl}}(\mathcal{Q} \parallel \mathbb{P}) + \ln \frac{2}{\delta}}{2mn}}. \quad (3)$$

As we show in Section 6.1.2, Corollary 1 is useful when the hypothesis class and instance space are uniformly bounded. Even when this property does not hold, we obtain an identical bound for all posteriors with $(\beta/n, \emptyset, 0)$ -local stability, meaning the support of the posterior is (β/n) -uniformly stable. However, this condition is less useful, since it is assumed that the posterior construction puts nonzero density on a learned hypothesis, which may not satisfy uniform stability for a fixed β .

It is worth noting that, if the hypothesis class does not use joint inference—for example, if a global prediction, $h(\mathbf{X})$, is in fact a set of independent, local predictions, $(h(X_1), \dots, h(X_n))$ —and the loss function decomposes over the labels, then uniform stability is trivially satisfied. In this case, Corollary 1 produces a PAC-Bayes bound for learning traditional predictors from interdependent data. If we further have that (Z_1, \dots, Z_n) are independent and identically distributed (i.i.d.)—for instance, if they represent “micro examples” drawn independently from some target distribution—then Corollary 1 reduces to standard PAC-Bayes bounds for learning from i.i.d. data (e.g., McAllester, 1999).

We now prove Theorem 1.

Proof (Theorem 1) We begin by defining two convenience functions,

$$\phi(h, \hat{\mathbf{Z}}) \triangleq \bar{L}(h) - \hat{L}(h, \hat{\mathbf{Z}}) \quad (4)$$

$$\text{and } \tilde{\phi}(h, \hat{\mathbf{Z}}) \triangleq \begin{cases} \mathbb{E}_{\mathbf{Z} \sim \mathbb{D}} [L(h, \mathbf{Z}) | \bar{\mathcal{B}}] - \hat{L}(h, \hat{\mathbf{Z}}) & \text{if } h \notin \mathcal{B}_{\mathcal{H}}, \\ 0 & \text{otherwise,} \end{cases} \quad (5)$$

If $L \circ \mathcal{H}$ is α -uniformly range-bounded (Definition 4), then, for any $h \in \mathcal{H}$,

$$\begin{aligned} \phi(h, \hat{\mathbf{Z}}) &= \frac{1}{m} \sum_{l=1}^m \bar{L}(h) - L(h, \mathbf{Z}^{(l)}) \\ &\leq \frac{1}{m} \sum_{l=1}^m \sup_{\mathbf{z} \in \mathcal{Z}^m} |L(h, \mathbf{z}) - L(h, \mathbf{Z}^{(l)})| \\ &\leq \frac{1}{m} \sum_{l=1}^m \alpha = \alpha. \end{aligned} \quad (6)$$

It follows that

$$\begin{aligned} \phi(h, \hat{\mathbf{Z}}) &= \mathbb{E}_{\mathbf{Z} \sim \mathbb{D}} [L(h, \mathbf{Z}) - \hat{L}(h, \hat{\mathbf{Z}})] \\ &= \mathbb{E}_{\mathbf{Z} \sim \mathbb{D}} \left[\left(L(h, \mathbf{Z}) - \hat{L}(h, \hat{\mathbf{Z}}) \right) \mathbb{1}\{\mathbf{Z} \notin \mathcal{B}_{\mathcal{Z}}\} \right] + \mathbb{E}_{\mathbf{Z} \sim \mathbb{D}} \left[\left(L(h, \mathbf{Z}) - \hat{L}(h, \hat{\mathbf{Z}}) \right) \mathbb{1}\{\mathbf{Z} \in \mathcal{B}_{\mathcal{Z}}\} \right] \\ &\leq \mathbb{E}_{\mathbf{Z} \sim \mathbb{D}} \left[\left(L(h, \mathbf{Z}) - \hat{L}(h, \hat{\mathbf{Z}}) \right) \mathbb{1}\{\mathbf{Z} \notin \mathcal{B}_{\mathcal{Z}}\} \right] + \alpha \mathbb{E}_{\mathbf{Z} \sim \mathbb{D}} [\mathbb{1}\{\mathbf{Z} \in \mathcal{B}_{\mathcal{Z}}\}] \\ &\leq \mathbb{E}_{\mathbf{Z} \sim \mathbb{D}} \left[\left(L(h, \mathbf{Z}) - \hat{L}(h, \hat{\mathbf{Z}}) \right) \mathbb{1}\{\mathbf{Z} \notin \mathcal{B}_{\mathcal{Z}}\} \right] + \alpha \nu \\ &= \Pr_{\mathbf{Z} \sim \mathbb{D}} \{ \mathbf{Z} \notin \mathcal{B}_{\mathcal{Z}} \} \left(\mathbb{E}_{\mathbf{Z} \sim \mathbb{D}} [L(h, \mathbf{Z}) | \bar{\mathcal{B}}] - \hat{L}(h, \hat{\mathbf{Z}}) \right) + \alpha \nu \\ &\leq \mathbb{E}_{\mathbf{Z} \sim \mathbb{D}} [L(h, \mathbf{Z}) | \bar{\mathcal{B}}] - \hat{L}(h, \hat{\mathbf{Z}}) + \alpha \nu. \end{aligned} \quad (7)$$

Moreover, for any posterior \mathbb{Q} with $(\beta/\eta, \mathcal{B}_{\mathcal{Z}} \cdot \eta)$ -local stability,

$$\begin{aligned} \bar{L}(\mathbb{Q}) - \hat{L}(\mathbb{Q}, \hat{\mathbf{Z}}) &= \mathbb{E}_{h \sim \mathbb{Q}} [\phi(h, \hat{\mathbf{Z}})] \\ &= \mathbb{E}_{h \sim \mathbb{Q}} [\phi(h, \hat{\mathbf{Z}}) \mathbb{1}\{h \in \mathcal{B}_{\mathcal{H}}\}] + \mathbb{E}_{h \sim \mathbb{Q}} [\phi(h, \hat{\mathbf{Z}}) \mathbb{1}\{h \notin \mathcal{B}_{\mathcal{H}}\}] \\ &\leq \alpha \mathbb{E}_{h \sim \mathbb{Q}} [\mathbb{1}\{h \in \mathcal{B}_{\mathcal{H}}\}] + \mathbb{E}_{h \sim \mathbb{Q}} [\phi(h, \hat{\mathbf{Z}}) \mathbb{1}\{h \notin \mathcal{B}_{\mathcal{H}}\}] \\ &\leq \alpha \eta + \mathbb{E}_{h \sim \mathbb{Q}} [\phi(h, \hat{\mathbf{Z}}) \mathbb{1}\{h \notin \mathcal{B}_{\mathcal{H}}\}] \\ &\leq \alpha \eta + \alpha \nu + \mathbb{E}_{h \sim \mathbb{Q}} [\tilde{\phi}(h, \hat{\mathbf{Z}})]. \end{aligned} \quad (8)$$

Then, for any $u \in \mathbb{R}$, using Lemma 1, we have that

$$\begin{aligned} \bar{L}(\mathbb{Q}) - \hat{L}(\mathbb{Q}, \hat{\mathbf{Z}}) &\leq \alpha \eta + \alpha \nu + \frac{1}{u} \mathbb{E}_{h \sim \mathbb{Q}} [u \tilde{\phi}(h, \hat{\mathbf{Z}})] \\ &\leq \alpha \eta + \alpha \nu + \frac{1}{u} \left(D_{\text{KL}}(\mathbb{Q} \| \mathbb{P}) + \ln \mathbb{E}_{h \sim \mathbb{P}} [e^{u \tilde{\phi}(h, \hat{\mathbf{Z}})}] \right). \end{aligned} \quad (9)$$

Since u cannot depend on (η, \mathbb{Q}) , we define it in terms of fixed quantities. For $j = 0, 1, 2, \dots$, let $\delta_j \triangleq \delta 2^{-(j+1)}$, let

$$u_j \triangleq 2^j \sqrt{\frac{8mn \ln \frac{2}{\delta}}{\beta^2 \|\mathbf{T}_{\bar{\mathcal{B}}}\|^2}}, \quad (10)$$

and define an event,

$$E_j \triangleq \mathbb{1} \left\{ \mathbb{E}_{h \sim \mathbb{P}} [e^{u_j \tilde{\phi}(h, \hat{\mathbf{Z}})}] \geq \frac{1}{\delta_j} \exp \left(\frac{u_j^2 \beta^2 \|\mathbf{T}_{\bar{\mathcal{B}}}\|^2}{8mn} \right) \right\}. \quad (11)$$

Note that u_j and E_j are independent of (η, \mathbb{Q}) , since β (hence, $\mathcal{B}_{\mathcal{H}}$) is fixed. Let $E \triangleq \bigcup_{j=0}^{\infty} E_j$ denote the event that any E_j occurs. We also define an event

$$B \triangleq \bigcup_{l=1}^m \mathbb{1} \{ \mathbf{Z}^{(l)} \in \mathcal{B}_{\mathcal{Z}} \}, \quad (12)$$

which indicates that at least one of the training examples is ‘‘bad.’’ Using the law of total probability and the union bound, we then have that

$$\begin{aligned} \Pr_{\mathbf{Z} \sim \mathbb{D}^m} \{ B \cup E \} &= \Pr_{\mathbf{Z} \sim \mathbb{D}^m} \{ B \} + \Pr_{\mathbf{Z} \sim \mathbb{D}^m} \{ E \cap \neg B \} \\ &\leq \Pr_{\mathbf{Z} \sim \mathbb{D}^m} \{ B \} + \Pr_{\mathbf{Z} \sim \mathbb{D}^m} \{ E | \neg B \} \\ &\leq \sum_{l=1}^m \Pr_{\mathbf{Z}^{(l)} \sim \mathbb{D}} \{ \mathbf{Z}^{(l)} \in \mathcal{B}_{\mathcal{Z}} \} + \sum_{j=0}^{\infty} \Pr_{\mathbf{Z} \sim \mathbb{D}^m} \{ E_j | \neg B \} \\ &\leq m\nu + \sum_{j=0}^{\infty} \Pr_{\mathbf{Z} \sim \mathbb{D}^m} \{ E_j | \neg B \}. \end{aligned} \quad (13)$$

The last inequality follows from the definition of ν . Then, using Markov’s inequality, and rearranging the expectations, we have that

$$\Pr_{\mathbf{Z} \sim \mathbb{D}^m} \{ E_j | \neg B \} \leq \delta_j \exp \left(-\frac{u_j^2 \beta^2 \|\mathbf{T}_{\bar{\mathcal{B}}}\|^2}{8mn} \right) \mathbb{E}_{h \sim \mathbb{P}} \mathbb{E}_{\mathbf{Z} \sim \mathbb{D}^m} [e^{u_j \tilde{\phi}(h, \hat{\mathbf{Z}})} | \neg B]. \quad (14)$$

Let

$$\varphi(h, \mathbf{Z}) \triangleq \begin{cases} \frac{1}{m} (\mathbb{E}_{\mathbf{Z} \sim \mathbb{D}} [L(h, \mathbf{Z}) | \bar{\mathcal{B}}] - L(h, \mathbf{Z})) & \text{if } h \notin \mathcal{B}_{\mathcal{H}}, \\ 0 & \text{otherwise,} \end{cases} \quad (15)$$

and note that $\tilde{\varphi}(h, \mathbf{Z}) = \sum_{l=1}^m \varphi(h, \mathbf{Z}^{(l)})$. Then, since $\mathbf{Z}^{(1)}, \dots, \mathbf{Z}^{(m)}$ are independent and identically distributed, we can write the inner expectation over \mathbf{Z} as

$$\begin{aligned} \mathbb{E}_{\mathbf{Z} \sim \mathbb{D}^m} \left[e^{u_j \tilde{\varphi}(h, \mathbf{Z})} \mid \neg B \right] &= \prod_{l=1}^m \mathbb{E}_{\mathbf{Z}^{(l)} \sim \mathbb{D}} \left[e^{u_j \varphi(h, \mathbf{Z}^{(l)})} \mid \neg B \right] \\ &= \prod_{l=1}^m \mathbb{E}_{\mathbf{Z}^{(l)} \sim \mathbb{D}} \left[e^{u_j \varphi(h, \mathbf{Z}^{(l)})} \mid \mathbf{Z}^{(l)} \notin \mathcal{B}_{\mathcal{Z}} \right] \\ &= \prod_{l=1}^m \mathbb{E}_{\mathbf{Z}^{(l)} \sim \mathbb{D}} \left[e^{u_j \varphi(h, \mathbf{Z}^{(l)})} \mid \overline{\mathcal{B}} \right]. \end{aligned} \quad (16)$$

By construction, $\varphi(h, \cdot)$ outputs zero whenever $h \in \mathcal{B}_{\mathcal{H}}$. In these cases, $\varphi(h, \cdot)$ trivially satisfies uniform stability, which implies local stability. Further, if \mathbb{Q} is $(\beta/n, \mathcal{B}_{\mathcal{Z}}, \eta)$ -locally stable, then every $L(h, \cdot) : h \notin \mathcal{B}_{\mathcal{H}}$ is $(\beta/n, \mathcal{B}_{\mathcal{Z}})$ -locally stable, and it is easily verified that $\varphi(h, \cdot) : h \notin \mathcal{B}_{\mathcal{H}}$ is $(\beta/(mn), \mathcal{B}_{\mathcal{Z}})$ -locally stable. Thus, $\varphi(h, \cdot) : h \in \mathcal{H}$ is $(\beta/(mn), \mathcal{B}_{\mathcal{Z}})$ -locally stable. Since $\mathbb{E}_{\mathbf{Z} \sim \mathbb{D}}[\varphi(h, \mathbf{Z}) \mid \overline{\mathcal{B}}] = 0$, we therefore apply Proposition 1 and have, for all $h \in \mathcal{H}$,

$$\mathbb{E}_{\mathbf{Z}^{(l)} \sim \mathbb{D}} \left[e^{u_j \varphi(h, \mathbf{Z}^{(l)})} \mid \overline{\mathcal{B}} \right] \leq \exp \left(\frac{u_j^2 \beta^2 \|\Gamma_{\overline{\mathcal{B}}}^{\mathbb{T}}\|_{\infty}^2}{8m^2 n} \right). \quad (17)$$

Combining Equations 14, 16 and 17, we have that

$$\begin{aligned} \Pr_{\mathbf{Z} \sim \mathbb{D}^m} \{ E_j \mid \neg B \} &\leq \delta_j \exp \left(-\frac{u_j^2 \beta^2 \|\Gamma_{\overline{\mathcal{B}}}^{\mathbb{T}}\|_{\infty}^2}{8mn} \right) \mathbb{E}_{h \sim \mathbb{P}} \left[\prod_{l=1}^m \mathbb{E}_{\mathbf{Z}^{(l)} \sim \mathbb{D}} \left[e^{u_j \varphi(h, \mathbf{Z}^{(l)})} \mid \overline{\mathcal{B}} \right] \right] \\ &\leq \delta_j \exp \left(-\frac{u_j^2 \beta^2 \|\Gamma_{\overline{\mathcal{B}}}^{\mathbb{T}}\|_{\infty}^2}{8mn} \right) \mathbb{E}_{h \sim \mathbb{P}} \left[\prod_{l=1}^m \exp \left(\frac{u_j^2 \beta^2 \|\Gamma_{\overline{\mathcal{B}}}^{\mathbb{T}}\|_{\infty}^2}{8m^2 n} \right) \right] \\ &= \delta_j \exp \left(-\frac{u_j^2 \beta^2 \|\Gamma_{\overline{\mathcal{B}}}^{\mathbb{T}}\|_{\infty}^2}{8mn} \right) \exp \left(\frac{u_j^2 \beta^2 \|\Gamma_{\overline{\mathcal{B}}}^{\mathbb{T}}\|_{\infty}^2}{8mn} \right) = \delta_j. \end{aligned} \quad (18)$$

Then, combining Equations 13 and 18, and using the geometric series identity, we have that

$$\Pr_{\mathbf{Z} \sim \mathbb{D}^m} \{ B \cup E \} \leq m\nu + \sum_{j=0}^{\infty} \delta_j = m\nu + \delta \sum_{j=0}^{\infty} 2^{-(j+1)} = m\nu + \delta.$$

Thus, with probability at least $1 - \delta - m\nu$ over realizations of $\tilde{\mathbf{Z}}$, every $l \in [m]$ satisfies $\mathbf{Z}^{(l)} \notin \mathcal{B}_{\mathcal{Z}}$, and every u_j satisfies

$$\mathbb{E}_{h \sim \mathbb{P}} \left[e^{u_j \tilde{\varphi}(h, \mathbf{Z})} \right] \leq \frac{1}{\delta_j} \exp \left(\frac{u_j^2 \beta^2 \|\Gamma_{\overline{\mathcal{B}}}^{\mathbb{T}}\|_{\infty}^2}{8mn} \right). \quad (19)$$

We now show how to select j for any particular posterior \mathbb{Q} . Let

$$j^* \triangleq \left\lfloor \frac{1}{2 \ln 2} \ln \left(\frac{D_{\text{kl}}(\mathbb{Q} \parallel \mathbb{P})}{\ln(2/\delta)} + 1 \right) \right\rfloor, \quad (20)$$

and note that $j^* \geq 0$. For all $v \in \mathbb{R}$, we have that $v - 1 \leq \lfloor v \rfloor \leq v$, and $2^{\ln v} = v^{\ln 2}$. We apply these identities to Equation 20 to show that

$$\frac{1}{2} \sqrt{\frac{D_{\text{kl}}(\mathbb{Q} \parallel \mathbb{P})}{\ln(2/\delta)} + 1} \leq 2^{j^*} \leq \sqrt{\frac{D_{\text{kl}}(\mathbb{Q} \parallel \mathbb{P})}{\ln(2/\delta)} + 1},$$

$$\sqrt{\frac{2mn \left(D_{\text{kl}}(\mathbb{Q} \parallel \mathbb{P}) + \ln \frac{2}{\delta} \right)}{\beta^2 \|\Gamma_{\overline{\mathcal{B}}}^{\mathbb{T}}\|_{\infty}^2}} \leq u_{j^*} \leq \sqrt{\frac{8mn \left(D_{\text{kl}}(\mathbb{Q} \parallel \mathbb{P}) + \ln \frac{2}{\delta} \right)}{\beta^2 \|\Gamma_{\overline{\mathcal{B}}}^{\mathbb{T}}\|_{\infty}^2}}. \quad (21)$$

implying

Further, by definition of δ_{j^*} ,

$$\begin{aligned} D_{\text{kl}}(\mathbb{Q} \parallel \mathbb{P}) + \ln \frac{1}{\delta_{j^*}} &= D_{\text{kl}}(\mathbb{Q} \parallel \mathbb{P}) + \ln \frac{2}{\delta} + j^* \ln 2 \\ &\leq D_{\text{kl}}(\mathbb{Q} \parallel \mathbb{P}) + \ln \frac{2}{\delta} + \frac{\ln 2}{2 \ln 2} \ln \left(\frac{D_{\text{kl}}(\mathbb{Q} \parallel \mathbb{P})}{\ln(2/\delta)} + 1 \right) \\ &= D_{\text{kl}}(\mathbb{Q} \parallel \mathbb{P}) + \ln \frac{2}{\delta} + \frac{1}{2} \ln \left(D_{\text{kl}}(\mathbb{Q} \parallel \mathbb{P}) + \ln \frac{2}{\delta} \right) - \frac{1}{2} \ln \ln \frac{2}{\delta} \\ &\leq D_{\text{kl}}(\mathbb{Q} \parallel \mathbb{P}) + \ln \frac{2}{\delta} + \frac{1}{2} \left(D_{\text{kl}}(\mathbb{Q} \parallel \mathbb{P}) + \ln \frac{2}{\delta} \right). \end{aligned} \quad (22)$$

The last inequality uses the identity $v - \ln \ln(2/\delta) \leq v + 1 \leq e^v$, for all $v \in \mathbb{R}$ and $\delta \in (0, 1)$. It can be shown that this bound is approximately optimal, in that it is at most twice what it would be for a fixed posterior.

Putting it all together, we now have that, with probability at least $1 - \delta - m\nu$, the approximately optimal (u_{j^*}, δ_{j^*}) for any posterior \mathbb{Q} satisfies

$$\begin{aligned} \tilde{L}(\mathbb{Q}) - \hat{L}(\mathbb{Q}, \tilde{\mathbf{Z}}) &\leq \alpha(\eta + \nu) + \frac{1}{u_{j^*}} \left(D_{\text{kl}}(\mathbb{Q} \parallel \mathbb{P}) + \ln \mathbb{E}_{h \sim \mathbb{P}} \left[e^{u_{j^*} \tilde{\varphi}(h, \tilde{\mathbf{Z}})} \right] \right) \\ &\leq \alpha(\eta + \nu) + \frac{1}{u_{j^*}} \left(D_{\text{kl}}(\mathbb{Q} \parallel \mathbb{P}) + \ln \frac{1}{\delta_{j^*}} + \frac{u_{j^*}^2 \beta^2 \|\Gamma_{\overline{\mathcal{B}}}^{\mathbb{T}}\|_{\infty}^2}{8mn} \right) \\ &\leq \alpha(\eta + \nu) + \frac{3 \left(D_{\text{kl}}(\mathbb{Q} \parallel \mathbb{P}) + \ln \frac{2}{\delta} \right)}{2u_{j^*}} + \frac{u_{j^*} \beta^2 \|\Gamma_{\overline{\mathcal{B}}}^{\mathbb{T}}\|_{\infty}^2}{8mn} \\ &\leq \alpha(\eta + \nu) + 2\beta \|\Gamma_{\overline{\mathcal{B}}}^{\mathbb{T}}\|_{\infty} \sqrt{\frac{D_{\text{kl}}(\mathbb{Q} \parallel \mathbb{P}) + \ln \frac{2}{\delta}}{2mn}}. \end{aligned}$$

The first inequality substitutes u_{j^*} into Equation 9; the second uses Equation 19; the third is from Equation 22; and the last uses the lower and upper bounds from Equation 21. ■

5.2 Posterior-Dependent Stability

In Theorem 1, we required β to be fixed *a priori*, meaning we required the user to pre-specify a desired stability. In this section, we prove bounds that hold for all $\beta \geq 1$ simultaneously,

meaning the value of β can depend on the learned posterior. (The requirement of nonnegativity is not restrictive, since stability with $\beta \leq 1$ implies stability with $\beta = 1$.)

Theorem 2. Fix $m \geq 1$, $n \geq 1$, $\boldsymbol{\pi} \in \Pi(n)$, $\delta \in (0, 1)$ and $\alpha \geq 0$. Fix a distribution, \mathbb{D} , on \mathcal{Z}^n . Fix a set of bad inputs, \mathcal{B}_Z , with $\nu \triangleq \mathbb{D}(\mathcal{B}_Z)$. Let $\mathbf{I}_{\frac{\boldsymbol{\pi}}{m}}$ denote the dependency matrix induced by \mathbb{D} , $\boldsymbol{\pi}$ and \mathcal{B}_Z . Fix a prior, \mathbb{P} , on a hypothesis class, \mathcal{H} . Fix a loss function, L , such that $L \circ \mathcal{H}$ is α -uniformly range-bounded. Then, with probability at least $1 - \delta - m\nu$ over realizations of $\hat{\mathbf{Z}} \triangleq (\mathbf{Z}^{(l)})_{l=1}^m$, drawn according to \mathbb{D}^m , the following hold: 1) for all $l \in [m]$, $\mathbf{Z}^{(l)} \notin \mathcal{B}_Z$; 2) for all $\beta \geq 1$, $\eta \in [0, 1]$ and posteriors \mathbb{Q} with $(\beta/n, \mathcal{B}_Z, \eta)$ -local stability,

$$\bar{L}(\mathbb{Q}) \leq \hat{L}(\mathbb{Q}, \hat{\mathbf{Z}}) + \alpha(\eta + \nu) + 4\beta \left\| \mathbf{I}_{\frac{\boldsymbol{\pi}}{m}} \right\|_{\infty} \sqrt{\frac{D_{\text{KL}}(\mathbb{Q} \parallel \mathbb{P}) + \ln \frac{4}{\delta} + \ln \beta}{2mn}}. \quad (23)$$

The proof is similar to that of Theorem 1, so we defer it to Appendix B.1.

Theorem 2 immediately yields the following corollary by taking $\mathcal{B}_Z \triangleq \emptyset$.

Corollary 2. With probability at least $1 - \delta$ over realizations of $\hat{\mathbf{Z}}$, for all $\beta \geq 1$, $\eta \in [0, 1]$ and \mathbb{Q} with $(\beta/n, \emptyset, \eta)$ -local stability,

$$\bar{L}(\mathbb{Q}) \leq \hat{L}(\mathbb{Q}, \hat{\mathbf{Z}}) + \alpha\eta + 4\beta \left\| \mathbf{I}_{\boldsymbol{\pi}} \right\|_{\infty} \sqrt{\frac{D_{\text{KL}}(\mathbb{Q} \parallel \mathbb{P}) + \ln \frac{4}{\delta} + \ln \beta}{2mn}}. \quad (24)$$

In Section 6, we apply this corollary to unbounded hypothesis classes, with bounded instance spaces. Corollary 2 trivially implies a bound for posteriors with $(\beta/n, \emptyset, 0)$ -local stability, such as those with bounded support on an unbounded hypothesis class, where β may depend on a learned model.

5.3 Derandomizing the Loss using Stability

PAC-Bayes bounds are stated in terms of a randomized predictor. Yet, in practice, one is usually interested in the loss of a learned, deterministic predictor. Given a properly constructed posterior distribution, it is possible to convert a PAC-Bayes bound to a generalization bound for the learned hypothesis. There are various ways to go about this for unstructured hypotheses; however, many of these methods fail for structured predictors, since the output is not simply a scalar, but a high-dimensional vector. In this section, we present a generic technique for derandomizing PAC-Bayes bounds for structured prediction based on the idea of stability. An attractive feature of this technique is that it obviates margin-based arguments, which often require a free-parameter for the margin.

We first define a specialized notion of local stability that measures the difference in loss induced by perturbing a given hypothesis. For the following, we view the posterior \mathbb{Q} as a function that, given a hypothesis $h \in \mathcal{H}$, returns a distribution \mathbb{Q}_h on \mathcal{H} .

Definition 8. Fix a hypothesis class, \mathcal{H} , a set of inputs, $\mathcal{B}_Z \subseteq \mathcal{Z}^n$, a loss function, L , and a posterior, \mathbb{Q} . We say that the pair (L, \mathbb{Q}) has $(\lambda, \mathcal{B}_Z, \eta)$ -local stability if, for any $h \in \mathcal{H}$ and $\mathbf{z} \notin \mathcal{B}_Z$, there exists a set $\mathcal{B}_{\mathcal{H}}(h, \mathbf{z}) \subseteq \mathcal{H}$ such that $\mathbb{Q}_h(\mathcal{B}_{\mathcal{H}}(h, \mathbf{z})) \leq \eta$ and, for all $h' \notin \mathcal{B}_{\mathcal{H}}(h, \mathbf{z})$,

$$|L(h, \mathbf{z}) - L(h', \mathbf{z})| \leq \lambda. \quad (25)$$

This form of stability is a slightly weaker condition than the previous definitions, in that each input, (h, \mathbf{z}) , has its own “bad” set, $\mathcal{B}_{\mathcal{H}}(h, \mathbf{z})$. This distinction means that “badness” is relative, whereas, in Definitions 2 and 3, it is absolute.

Proposition 4. Fix a hypothesis class, \mathcal{H} , a set of inputs, $\mathcal{B}_Z \subseteq \mathcal{Z}^n$, with $\nu \triangleq \mathbb{D}(\mathcal{B}_Z)$, and a loss function, L , such that, for any $\mathbf{z} \in \mathcal{Z}^n$, $L(\cdot, \mathbf{z})$ is α -uniformly range-bounded. Let \mathbb{Q} denote a posterior function on \mathcal{H} . If (L, \mathbb{Q}) has $(\lambda, \mathcal{B}_Z, \eta)$ -local stability, then, for all $h \in \mathcal{H}$,

$$|\bar{L}(h) - \bar{L}(\mathbb{Q}_h)| \leq \alpha(\eta + \nu) + \lambda, \quad (26)$$

and, for all $\hat{\mathbf{z}} \triangleq (\mathbf{z}^{(l)})_{l=1}^m$ such that, $\forall l \in [m]$, $\mathbf{z}^{(l)} \notin \mathcal{B}_Z$,

$$|\hat{L}(h, \hat{\mathbf{z}}) - \hat{L}(\mathbb{Q}_h, \hat{\mathbf{z}})| \leq \alpha\eta + \lambda. \quad (27)$$

Proof Define a convenience function

$$\varphi(h, h', \mathbf{z}) \triangleq |L(h, \mathbf{z}) - L(h', \mathbf{z})|.$$

For any $\mathbf{z} \notin \mathcal{B}_Z$, using the range-boundedness and stability assumptions, we have that

$$\begin{aligned} & \mathbb{E}_{h' \sim \mathbb{Q}_h} [\varphi(h, h', \mathbf{z})] \\ &= \mathbb{E}_{h' \sim \mathbb{Q}_h} [\varphi(h, h', \mathbf{z}) \mathbb{1}\{h' \in \mathcal{B}_{\mathcal{H}}(h, \mathbf{z})\}] + \mathbb{E}_{h' \sim \mathbb{Q}_h} [\varphi(h, h', \mathbf{z}) \mathbb{1}\{h' \notin \mathcal{B}_{\mathcal{H}}(h, \mathbf{z})\}] \\ &\leq \alpha\eta + \lambda. \end{aligned}$$

Therefore, if $\forall l \in [m]$, $\mathbf{z}^{(l)} \notin \mathcal{B}_Z$, by linearity of expectation and the triangle inequality,

$$\begin{aligned} |\hat{L}(h, \hat{\mathbf{z}}) - \hat{L}(\mathbb{Q}_h, \hat{\mathbf{z}})| &= \left| \frac{1}{m} \sum_{l=1}^m \mathbb{E}_{\mathbf{z}^{(l)} \sim \mathbb{Q}_h} [L(h, \mathbf{z}^{(l)}) - L(h', \mathbf{z}^{(l)})] \right| \\ &\leq \frac{1}{m} \sum_{l=1}^m \mathbb{E}_{h' \sim \mathbb{Q}_h} [\varphi(h, h', \mathbf{z}^{(l)})] \\ &\leq \alpha\eta + \lambda. \end{aligned}$$

thus proving Equation 27. Furthermore,

$$\begin{aligned} |\bar{L}(h) - \bar{L}(\mathbb{Q}_h)| &= \left| \mathbb{E}_{\mathbf{z} \sim \mathbb{D}} \mathbb{E}_{h' \sim \mathbb{Q}_h} [L(h, \mathbf{Z}) - L(h', \mathbf{Z})] \right| \\ &\leq \mathbb{E}_{\mathbf{z} \sim \mathbb{D}} \mathbb{E}_{h' \sim \mathbb{Q}_h} [\varphi(h, h', \mathbf{Z})] \\ &= \mathbb{E}_{\mathbf{z} \sim \mathbb{D}} \mathbb{E}_{h' \sim \mathbb{Q}_h} [\varphi(h, h', \mathbf{Z}) \mathbb{1}\{\mathbf{Z} \in \mathcal{B}_Z\}] + \mathbb{E}_{\mathbf{z} \sim \mathbb{D}} \mathbb{E}_{h' \sim \mathbb{Q}_h} [\varphi(h, h', \mathbf{Z}) \mathbb{1}\{\mathbf{Z} \notin \mathcal{B}_Z\}] \\ &\leq \alpha\nu + \alpha\eta + \lambda, \end{aligned}$$

which proves Equation 26. \blacksquare

Proposition 4 can easily be combined with the PAC-Bayes bounds from the previous sections to obtain derandomized generalization bounds. We analyze some examples in Section 6.

5.3.1 NORMED VECTOR SPACES

When the hypothesis class is a normed vector space (as is the case in all of the examples in Section 6), Definition 8 can be decomposed into properties of the loss function and posterior separately.

Definition 9. Fix a hypothesis class, \mathcal{H} , equipped with a norm, $\|\cdot\|$. Fix a set of inputs, $\mathcal{B}_Z \subseteq \mathcal{Z}^n$. We say that a loss function, L , has (λ, \mathcal{B}_Z) -local hypothesis stability if, for all $h, h' \in \mathcal{H}$ and $\mathbf{z} \notin \mathcal{B}_Z$,

$$|L(h, \mathbf{z}) - L(h', \mathbf{z})| \leq \lambda \|h - h'\|.$$

Definition 10. Fix a hypothesis class, \mathcal{H} , equipped with a norm, $\|\cdot\|$. We say that a posterior, \mathbb{Q} , has (β, η) -local hypothesis stability if, for any $h \in \mathcal{H}$, there exists a set $\mathcal{B}_H(h) \subseteq \mathcal{H}$ such that $\mathbb{Q}_h(\mathcal{B}_H(h)) \leq \eta$ and, for all $h' \notin \mathcal{B}_H(h)$, $\|h - h'\| \leq \beta$.

When both of these properties hold, we have the following.

Proposition 5. Fix a hypothesis class, \mathcal{H} , equipped with a norm, $\|\cdot\|$. Fix a set of inputs, $\mathcal{B}_Z \subseteq \mathcal{Z}^n$. If a loss function, L , has (λ, \mathcal{B}_Z) -local hypothesis stability, and a posterior, \mathbb{Q} , has (β, η) -local hypothesis stability, then (L, \mathbb{Q}) has $(\lambda\beta, \mathcal{B}_Z, \eta)$ -local stability.

The proof is provided in Appendix B.2.

6. Example Applications

To illustrate how various learning algorithms and modeling decisions affect the generalization error, we now apply our PAC-Bayes bounds to the class of pairwise MRFs with templated, linear potentials (described in Section 2.2.3). We derive generalization bounds for two popular training regimes, *max-margin* and *soft-max* learning, under various assumptions about the instance space and feature functions. The bounds in this section are stated in terms of a deterministic predictor, meaning we use the PAC-Bayes framework as an analytic tool only. That said, one could easily adapt our analysis to obtain bounds for a randomized predictor by skipping the derandomization step.

6.1 Max-Margin Learning

For classification tasks, the goal is to output the labeling that is closest to the true labeling, by some measure of closeness. This is usually measured by the *Hamming loss*,

$$L_H(h, \mathbf{x}, \mathbf{y}) \triangleq \frac{1}{n} D_H(\mathbf{y}, h(\mathbf{x})).$$

The Hamming loss can be considered the structured equivalent of the θ -1 loss. Unfortunately, the Hamming loss is not convex, making it difficult to minimize directly. Thus, many learning algorithms minimize a convex upper bound.

One such method is *max-margin* learning. Max-margin learning aims to find the “simplest” model that scores the correct outputs higher than all incorrect outputs by a specified margin. Though typically formulated as a quadratic program, the learning objective can also be stated as minimizing a *hinge loss*, with model regularization.

Structured predictors learned with a max-margin objective are alternatively referred to as *max-margin Markov networks* (Taskar et al., 2004) or *StructSVM* (Tschantzaris et al., 2005), depending on the form of the hinge loss. In this section, we consider the former formulation, defining the structured hinge loss as

$$L_h(h, \mathbf{x}, \mathbf{y}) \triangleq \frac{1}{n} \left(\max_{\mathbf{y}' \in \mathcal{Y}^n} D_H(\mathbf{y}, \mathbf{y}') + h(\mathbf{x}, \mathbf{y}') - h(\mathbf{x}, \mathbf{y}) \right), \quad (28)$$

where

$$h(\mathbf{x}, \mathbf{y}) \triangleq \boldsymbol{\theta}(\mathbf{x}; \mathbf{w}) \cdot \hat{\mathbf{y}} \quad (29)$$

is the unnormalized log-likelihood. The Hamming distance, $D_H(\mathbf{y}, \mathbf{y}')$, implies that the margin, $h(\mathbf{x}, \mathbf{y}) - h(\mathbf{x}, \mathbf{y}')$, should scale linearly with the distance between \mathbf{y} and \mathbf{y}' .

In theory, the structured hinge loss can be defined with any distance function; though, in practice, the Hamming distance is commonly used. One attractive property of the Hamming distance is that, when

$$h(\mathbf{x}) \triangleq \arg \max_{\mathbf{y} \in \mathcal{Y}^n} h(\mathbf{x}, \mathbf{y}) = \arg \max_{\mathbf{y} \in \mathcal{Y}^n} p(\mathbf{Y} = \mathbf{y} \mid \mathbf{X} = \mathbf{x}; \mathbf{w}) \quad (30)$$

(i.e., MAP inference), the hinge loss upper-bounds the Hamming loss. Another benefit is that it decomposes along the unary cliques. Indeed, with $\delta(\mathbf{y}) \triangleq \begin{bmatrix} 1-\mathbf{y} \\ \mathbf{0} \end{bmatrix}$ (i.e., one minus the unary clique states, then zero-padded to be the same length as $\hat{\mathbf{y}}$), observe that $D_H(\mathbf{y}, \mathbf{y}') = \delta(\mathbf{y}) \cdot \hat{\mathbf{y}}'$. This identity yields a convenient equivalence:

$$L_h(h, \mathbf{x}, \mathbf{y}) = \frac{1}{n} \left(\max_{\mathbf{y}' \in \mathcal{Y}^n} (\boldsymbol{\theta}(\mathbf{x}; \mathbf{w}) + \delta(\mathbf{y})) \cdot \hat{\mathbf{y}}' - \boldsymbol{\theta}(\mathbf{x}; \mathbf{w}) \cdot \hat{\mathbf{y}} \right).$$

The term $\boldsymbol{\theta}(\mathbf{x}; \mathbf{w}) \cdot \hat{\mathbf{y}}$ is constant with respect to \mathbf{y}' , and is thus irrelevant to the maximization. Therefore, letting

$$\hat{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}; \mathbf{w}) \triangleq \boldsymbol{\theta}(\mathbf{x}; \mathbf{w}) + \delta(\mathbf{y}), \quad (31)$$

computing the hinge loss is equivalent to performing *loss-augmented* MAP inference with $\hat{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{y}; \mathbf{w})$. Provided inference can be computed efficiently with the given class of models, so too can the hinge loss.⁶

6.1.1 STRUCTURED RAMP LOSS

Applying our generalization bounds requires a uniformly range-bounded loss function. Since the hinge loss is not uniformly range-bounded for certain hypothesis classes, we therefore introduce the structured *ramp loss*:

$$L_r(h, \mathbf{x}, \mathbf{y}) \triangleq \frac{1}{n} \left(\max_{\mathbf{y}' \in \mathcal{Y}^n} D_H(\mathbf{y}, \mathbf{y}') + h(\mathbf{x}, \mathbf{y}') - \max_{\mathbf{y}'' \in \mathcal{Y}^n} h(\mathbf{x}, \mathbf{y}'') \right),$$

where $h(\mathbf{x}, \mathbf{y})$ is defined in Equation 29. The ramp loss is 1-uniformly range-bounded. Further, when $h(\mathbf{x})$ performs MAP inference (Equation 30),

$$L_H(h, \mathbf{x}, \mathbf{y}) \leq L_r(h, \mathbf{x}, \mathbf{y}) \leq L_H(h, \mathbf{x}, \mathbf{y}). \quad (32)$$

6. The results in this section are easily extended to approximate MAP inference algorithms, such as linear programming relaxations. The bounds are the same, but the semantics of the loss functions change, since approximate MAP solutions might be fractional.

Thus, we can analyze the generalization properties of the ramp loss to obtain bounds for the difference of the expected Hanning loss and empirical hinge loss. To distinguish quantities of different loss functions, we will use a subscript notation: e.g., \bar{L}_π is the expected Hanning loss, and \hat{L}_h is the empirical hinge loss.

Using the templated, linear potentials defined in Section 2.2.3, we obtain two technical lemmas for the structured ramp loss. Proofs are provided in Appendices C.1 and C.2.

Lemma 2. Fix any $p, q \geq 1$ such that $1/p + 1/q = 1$. Fix a graph, $G \triangleq (\mathcal{V}, \mathcal{E})$, with maximum degree Δ_G . Assume that $\sup_{x \in \mathcal{X}} \|x\|_p \leq R$. Then, for any MRF h with weights \mathbf{w} , and any $\mathbf{z}, \mathbf{z}' \in \mathcal{Z}^n$, where $\mathbf{z} = (\mathbf{x}, \mathbf{y})$ and $\mathbf{z}' = (\mathbf{x}', \mathbf{y}')$,

$$|L_r(h, \mathbf{z}) - L_r(h, \mathbf{z}')| \leq \frac{1}{n} \left((2\Delta_G + 4)R \|\mathbf{w}\|_q + 1 \right) D_h(\mathbf{z}, \mathbf{z}'). \quad (33)$$

Further, if the model does not use edge observations (i.e., $f_{ij}(\mathbf{x}, \mathbf{y}) \triangleq y_i \otimes y_j$), then

$$|L_r(h, \mathbf{z}) - L_r(h, \mathbf{z}')| \leq \frac{1}{n} \left(4R \|\mathbf{w}\|_q + 1 \right) D_h(\mathbf{z}, \mathbf{z}'). \quad (34)$$

Lemma 3. Fix any $p, q \geq 1$ such that $1/p + 1/q = 1$. Fix a graph, $G \triangleq (\mathcal{V}, \mathcal{E})$. Assume that $\sup_{x \in \mathcal{X}} \|x\|_p \leq R$. Then, for any example $\mathbf{z} \in \mathcal{Z}^n$, and any two MRFs, h, h' with weights \mathbf{w}, \mathbf{w}' ,

$$|L_r(h, \mathbf{z}) - L_r(h', \mathbf{z})| \leq \frac{2 \|G\|_R}{n} \|\mathbf{w} - \mathbf{w}'\|_q.$$

Lemma 3 implies that L_r has $(2 \|G\|_R/n, \emptyset)$ -local hypothesis stability.

6.1.2 GENERALIZATION BOUNDS FOR MAX-MARGIN LEARNING

We now apply our PAC-Bayes bounds to the class of max-margin Markov networks that perform MAP inference, with the templated, linear potentials defined in Section 2.2.3. We denote this class by \mathcal{H}_{M3N} . As a warm-up, we first assume that both the observations and weights are uniformly bounded by the 2-norm unit ball. By Lemma 2, this means that the ramp loss satisfies uniform stability, meaning we can apply Corollary 1.

Example 1. Fix any $m \geq 1$, $n \geq 1$, $\pi \in \Pi(n)$ and $\delta \in (0, 1)$. Fix a graph, $G \triangleq (\mathcal{V}, \mathcal{E})$, with maximum degree Δ_G . Assume that $\sup_{x \in \mathcal{X}} \|x\|_2 \leq 1$. Then, with probability at least $1 - \delta$ over realizations of $\hat{\mathbf{Z}} \triangleq (\mathbf{Z}^{(i)})_{i=1}^m$, for all $h \in \mathcal{H}_{\text{M3N}}$ with $\|\mathbf{w}\|_2 \leq 1$,

$$\bar{L}_\pi(h) \leq \hat{L}_h(h, \hat{\mathbf{Z}}) + \frac{4}{mn} + (4\Delta_G + 10) \|\Gamma^\pi\|_\infty \sqrt{\frac{d \ln(2m |G|) + \ln \frac{4}{\delta}}{2mn}}.$$

The proof is given in Appendix C.3. Note that, with the bounded degree assumption, $|G| \leq n\Delta_G = O(n)$.

We now relax the assumption that the hypothesis class is bounded. One approach is to apply a covering argument directly to Example 1. However, it is interesting to see how other prior/posterior constructions behave. Of particular interest are Gaussian constructions, which correspond to 2-norm regularization. Since the support of a Gaussian is unbounded, this construction requires a non-uniform notion of stability. The following example illustrates how to use posterior-dependent, local stability.

Example 2. Fix any $m \geq 1$, $n \geq 1$, $\pi \in \Pi(n)$ and $\delta \in (0, 1)$. Fix a graph, $G \triangleq (\mathcal{V}, \mathcal{E})$, with maximum degree Δ_G . Assume that $\sup_{x \in \mathcal{X}} \|x\|_2 \leq 1$. Then, with probability at least $1 - \delta$ over realizations of $\hat{\mathbf{Z}} \triangleq (\mathbf{Z}^{(i)})_{i=1}^m$, for all $h \in \mathcal{H}_{\text{M3N}}$,

$$\bar{L}_\pi(h) \leq \hat{L}_h(h, \hat{\mathbf{Z}}) + \frac{7}{mn} + 4\beta_h \|\Gamma^\pi\|_\infty \sqrt{\frac{\frac{1}{2} \|\mathbf{w}\|_2^2 + \frac{d}{2} \ln(2d(m |G|)^2 \ln(2dm)) + \ln \frac{4\beta_h}{\delta}}{2mn}},$$

where

$$\beta_h \triangleq (2\Delta_G + 4) \left(\|\mathbf{w}\|_2 + \frac{1}{m |G|} \right) + 1.$$

Example 2 is only slightly worse than Example 1, incurring a $O(\ln \ln(mn))$ term for the Gaussian construction. Both bounds guarantee generalization when either m or n is large.

The proof of Example 2 uses a concentration inequality for vectors of Gaussian random variables, the proof of which is given Appendix C.4.

Lemma 4. Let $\mathbf{X} \triangleq (X_i)_{i=1}^d$ be independent Gaussian random variables, with mean vector $\boldsymbol{\mu} \triangleq (\mu_1, \dots, \mu_d)$ and variance σ^2 . Then, for any $p \geq 1$ and $\epsilon > 0$,

$$\Pr \left\{ \|\mathbf{X} - \boldsymbol{\mu}\|_p \geq \epsilon \right\} \leq 2d \exp \left(-\frac{\epsilon^2}{2\sigma^2 q^{2/p}} \right).$$

For $p = 2$ and small σ^2 , this bound can be significantly sharper than Chebyshev's inequality.

Proof (Example 2) Define the prior, \mathbb{P} , as a standard multivariate Gaussian, with zero mean and unit variance. More precisely, let

$$p(h) \triangleq (2\pi)^{-d/2} e^{-\frac{1}{2} \|\mathbf{w}\|_2^2}$$

denote the density of \mathbb{P} . Given a (learned) hypothesis, h , we construct the posterior, \mathbb{Q}_h , as an isotropic Gaussian, centered at \mathbf{w} , with variance

$$\sigma^2 \triangleq (2d(m |G|)^2 \ln(2dm))^{-1}$$

in all dimensions. Its density is

$$q_h(h) \triangleq (2\pi\sigma^2)^{-d/2} e^{-\frac{1}{2\sigma^2} \|\mathbf{w}' - \mathbf{w}\|_2^2}.$$

Note that the support of both distributions is \mathbb{R}^d , which is unbounded.

Our proof technique involves four steps. First, we upper-bound the KL divergence between \mathbb{Q}_h and \mathbb{P} . Then, we identify a β_h and η such that \mathbb{Q}_h is $(\beta_h/n, \emptyset, \eta)$ -locally stable. Combining the first two steps with Corollary 2 yields a PAC-Bayes bound for the randomized predictor. The final step is to derandomize this bound using Proposition 4.

The KL divergence between Gaussians is well known. Thus, it is easily verified that

$$\begin{aligned} D_{\text{KL}}(\mathbb{Q}_h \| \mathbb{P}) &= \frac{1}{2} \left[d(\sigma^2 - 1) + \|\mathbf{w}\|_2^2 - d \ln \sigma^2 \right] \\ &= \frac{1}{2} \left[d \left(\frac{1}{2d(m |G|)^2 \ln(2dm)} - 1 \right) + \|\mathbf{w}\|_2^2 + d \ln(2d(m |G|)^2 \ln(2dm)) \right] \\ &\leq \frac{1}{2} \left[\|\mathbf{w}\|_2^2 + d \ln(2d(m |G|)^2 \ln(2dm)) \right]. \end{aligned}$$

The inequality follows from the fact that $\sigma^2 \leq 1$ for all $d \geq 1$, $m \geq 1$ and $n \geq 1$ (implying $|G| \geq 1$).

To determine the local stability of \mathbb{Q}_h , for any $h \in \mathcal{H}_{\text{M3N}}$, we define a ‘‘bad’’ set of hypotheses,

$$\mathcal{B}_{\mathcal{H}_{\text{M3N}}}(h) \triangleq \left\{ h' \in \mathcal{H}_{\text{M3N}} : \|\mathbf{w}' - \mathbf{w}\|_2 \geq \frac{1}{m|G|} \right\}.$$

Using Lemma 4,

$$\begin{aligned} \mathbb{Q}_h(\mathcal{B}_{\mathcal{H}_{\text{M3N}}}(h)) &= \Pr_{h' \sim \mathbb{Q}_h} \left\{ \|\mathbf{w}' - \mathbf{w}\|_2 \geq \frac{1}{m|G|} \right\} \\ &\leq 2d \exp \left(-\frac{2d(m|G|)^2 \ln(2dmn)}{2d(m|G|)^2} \right) \\ &= \frac{1}{mn}. \end{aligned} \quad (35)$$

Further, for every $h' \notin \mathcal{B}_{\mathcal{H}_{\text{M3N}}}(h)$,

$$\|\mathbf{w}'\|_2 - \|\mathbf{w}\|_2 \leq \|\mathbf{w}' - \mathbf{w}\|_2 \leq \frac{1}{m|G|}.$$

When combined with Lemma 2, with $R = 1$, we have that

$$\begin{aligned} |L_{\mathcal{R}}(h, \mathbf{z}) - L_{\mathcal{R}}(h', \mathbf{z}')| &\leq \frac{1}{n} ((2\Delta_G + 4) \|\mathbf{w}'\|_2 + 1) D_{\text{H}}(\mathbf{z}, \mathbf{z}') \\ &\leq \frac{1}{n} \left((2\Delta_G + 4) \left(\|\mathbf{w}\|_2 + \frac{1}{m|G|} \right) + 1 \right) D_{\text{H}}(\mathbf{z}, \mathbf{z}') \\ &= \frac{\beta_h}{n} D_{\text{H}}(\mathbf{z}, \mathbf{z}'). \end{aligned}$$

Thus, every \mathbb{Q}_h is $(\beta_h/n, \theta, 1/(mn))$ -locally stable.

Having established an upper bound on the KL divergence and local stability of all posteriors, we can now apply one of our PAC-Bayes bounds. Since the definition of β_h depends on the posterior via \mathbf{w} , we must use a bound from Section 5.2. In this case, there are no ‘‘bad’’ inputs, since the observations are bounded in the unit ball, so we can invoke Corollary 2. Recalling that the ramp loss is 1-uniformly difference bounded, we then have that, with probability at least $1 - \delta$, every $\mathbb{Q}_h : h \in \mathcal{H}_{\text{M3N}}$ satisfies

$$\begin{aligned} \bar{L}_{\mathcal{R}}(\mathbb{Q}_h) &\leq \hat{L}_{\mathcal{R}}(\mathbb{Q}_h, \hat{\mathbf{Z}}) + \frac{1}{mn} \\ &\quad + 4\beta_h \|\Gamma^\pi\|_\infty \sqrt{\frac{\frac{1}{2} \|\mathbf{w}\|_2^2 + \frac{d}{2} \ln(2d(m|G|)^2 \ln(2dmn)) + \ln \frac{4\beta_h}{\delta}}{2mn}}. \end{aligned} \quad (36)$$

We now derandomize the loss terms in Equation 36. Observe that \mathcal{H}_{M3N} is a normed vector space, since it consists of weight vectors in \mathbb{R}^d . In this case, we will use the 2-norm. By Equation 35, it is clear that \mathbb{Q} has $(1/(m|G|), 1/(mn))$ -local hypothesis stability (Definition 10), since every $h \in \mathcal{H}_{\text{M3N}}$ results in the same probability bound. Further, by Lemma 3, with $R = 1$,

$$|L_{\mathcal{R}}(h, \mathbf{z}) - L_{\mathcal{R}}(h', \mathbf{z})| \leq \frac{2|G|}{n} \|\mathbf{w} - \mathbf{w}'\|_2, \quad (37)$$

meaning $L_{\mathcal{R}}$ has $(2|G|/n, \theta)$ -local hypothesis stability (Definition 9). Therefore, by Proposition 5, $(L_{\mathcal{R}}, \mathbb{Q})$ has $(2/(mn), \theta, 1/(mn))$ -local stability. It then follows, via Proposition 4 and Equation 32, that

$$\bar{L}_{\text{H}}(h) \leq \bar{L}_{\mathcal{R}}(h) \leq \bar{L}_{\mathcal{R}}(\mathbb{Q}_h) + \frac{3}{mn}, \quad (38)$$

and

$$\hat{L}_{\mathcal{R}}(\mathbb{Q}_h, \hat{\mathbf{Z}}) \leq \hat{L}_{\mathcal{R}}(h, \hat{\mathbf{Z}}) + \frac{3}{mn} \leq \hat{L}_{\text{H}}(h, \hat{\mathbf{Z}}) + \frac{3}{mn}. \quad (39)$$

Combining Equations 36, 38 and 39 completes the proof. \blacksquare

6.2 Soft-Max Learning

A drawback of max-margin learning is that the learning objective is not differentiable everywhere, due to the hinge loss. Thus, researchers (Gimpel and Smith, 2010; Hazan and Urtasun, 2010) have proposed a smooth alternative, based on the *soft-max* function. This form of learning has been popularized for learning conditional random fields (CRFs).

The soft-max loss, for a given temperature parameter, $\epsilon \in [0, 1]$, is defined as

$$L_{\text{SM}}(h, \mathbf{x}, \mathbf{y}) \triangleq \frac{1}{n} (\Phi_\epsilon(\mathbf{x}, \mathbf{y}; \mathbf{w}) - h(\mathbf{x}, \mathbf{y})), \quad (40)$$

where $h(\mathbf{x}, \mathbf{y})$ is the unnormalized log-likelihood (Equation 29) and

$$\begin{aligned} \Phi_\epsilon(\mathbf{x}, \mathbf{y}; \mathbf{w}) &\triangleq \epsilon \ln \sum_{\mathbf{y}' \in \mathcal{Y}^n} \exp \left(\frac{1}{\epsilon} (D_{\text{H}}(\mathbf{y}, \mathbf{y}') + h(\mathbf{x}, \mathbf{y}')) \right) \\ &= \epsilon \ln \sum_{\mathbf{y}' \in \mathcal{Y}^n} \exp \left(\frac{1}{\epsilon} \hat{\theta}(\mathbf{x}, \mathbf{y}; \mathbf{w}) \cdot \mathbf{y}' \right). \end{aligned} \quad (41)$$

is the soft-max function. We purposefully overload the notation of the log-partition function due to its relationship to the soft-max. Observe that, for $\epsilon = 1$, the soft-max becomes the log-partition of the distribution induced by the loss-augmented potentials, and Equation 40 is the corresponding negative log-likelihood, scaled by $1/n$. Further, as $\epsilon \rightarrow 0$, the soft-max approaches the max operator and Equation 40 becomes the hinge loss (Equation 28).

The latter equivalence can be illustrated by convex conjugacy. This requires some additional notation. Let $\mu \in [0, 1]^{|Y|+|E|+|Y|^2}$ denote a vector of marginal probabilities for all cliques and clique states. Let \mathcal{M} denote the set of all consistent marginal vectors, often called the *marginal polytope*. For every $\mu \in \mathcal{M}$, there is a corresponding distribution, p_μ , such that $\mu_c \cdot \mathbf{y}_c = p_\mu(\mathbf{Y}_c = \mathbf{y}_c)$ for every clique, $c \in \mathcal{C}$, and clique state, \mathbf{y}_c . Let $\Phi^*(\mu)$ denote the *convex conjugate* of the log-partition, which, for $\mu \in \mathcal{M}$, is equal to the negative entropy of p_μ .⁷ With these definitions, the soft-max, like the log-partition, has the following variational form:

$$\begin{aligned} \Phi_\epsilon(\mathbf{x}, \mathbf{y}; \mathbf{w}) &= \max_{\mu \in \mathcal{M}} \hat{\theta}(\mathbf{x}, \mathbf{y}; \mathbf{w}) \cdot \mu - \epsilon \Phi^*(\mu) \\ &= \max_{\mu \in \mathcal{M}} (\hat{\theta}(\mathbf{x}, \mathbf{y}; \mathbf{w}) + \delta(\mathbf{y})) \cdot \mu - \epsilon \Phi^*(\mu). \end{aligned} \quad (42)$$

⁷ We omit some details of the conjugate function for simplicity of exposition. See Wainwright and Jordan (2008) for a precise definition.

This maximization is equivalent to marginal inference with loss-augmented potentials.⁸ Let μ_u denote the marginals of the unary cliques, and observe that

$$\delta(\mathbf{y}) \cdot \boldsymbol{\mu} = \frac{1}{2} \|\mathbf{y} - \boldsymbol{\mu}_u\|_1 \triangleq D_1(\mathbf{y}, \boldsymbol{\mu}). \quad (43)$$

With a slight abuse of notation, we define an alternate scoring function for marginals:

$$h_\epsilon(\mathbf{x}, \boldsymbol{\mu}) \triangleq \boldsymbol{\theta}(\mathbf{x}; \mathbf{w}) \cdot \boldsymbol{\mu} - \epsilon \Phi^*(\boldsymbol{\mu}). \quad (44)$$

Note that each full labeling, $\hat{\mathbf{y}}$, corresponds to a vertex of the marginal polytope, so $\hat{\mathbf{y}} \in \mathcal{M}$. Further, $h_\epsilon(\mathbf{x}, \hat{\mathbf{y}}) = h(\mathbf{x}, \mathbf{y})$, since $\Phi^*(\hat{\mathbf{y}}) = 0$. Thus, combining Equations 42 to 44, we have that the soft-max loss (Equation 40) is equivalent to

$$L_{sm}(h, \mathbf{x}, \mathbf{y}) = \frac{1}{n} \left(\max_{\boldsymbol{\mu} \in \mathcal{M}} D_1(\mathbf{y}, \boldsymbol{\mu}) + h_\epsilon(\mathbf{x}, \boldsymbol{\mu}) - h_\epsilon(\mathbf{x}, \hat{\mathbf{y}}) \right),$$

which resembles a smoothed hinge loss for $\epsilon \in (0, 1)$.

Like the regular hinge loss, $L_{sm}(h, \mathbf{x}, \mathbf{y})$ is not uniformly range-bounded for certain hypothesis classes, so it cannot be used with our PAC-Bayes bounds. However, we can use the ramp loss, with a slight modification:

$$L_{sr}(h, \mathbf{x}, \mathbf{y}) \triangleq \frac{1}{n} \left(\max_{\boldsymbol{\mu} \in \mathcal{M}} D_1(\mathbf{y}, \boldsymbol{\mu}) + h_\epsilon(\mathbf{x}, \boldsymbol{\mu}) - \max_{\boldsymbol{\mu}' \in \mathcal{M}} h_\epsilon(\mathbf{x}, \boldsymbol{\mu}') \right).$$

We have essentially just replaced the maxes over \mathcal{Z}^n with maxes over \mathcal{M} and used Equation 44 instead of Equation 29. We refer to this loss as the *soft ramp loss*. The stability properties of the regular ramp loss over to the soft ramp loss; it is straightforward to show that Lemmas 2 and 3 hold when $L_r(h, \mathbf{x}, \mathbf{y})$ is replaced with $L_{sr}(h, \mathbf{x}, \mathbf{y})$.⁹ The distance function, $D_1(\mathbf{y}, \boldsymbol{\mu})$, has a probabilistic interpretation:

$$D_1(\mathbf{y}, \boldsymbol{\mu}) = \sum_{i=1}^n 1 - p_{\boldsymbol{\mu}}(Y_i = y_i | \mathbf{X} = \mathbf{x}).$$

This identity motivates another loss function: with

$$h_\epsilon(\mathbf{x}) \triangleq \arg \max_{\boldsymbol{\mu} \in \mathcal{M}} h_\epsilon(\mathbf{x}, \boldsymbol{\mu}),$$

let

$$L_1(h, \mathbf{x}, \mathbf{y}) \triangleq \frac{1}{n} D_1(\mathbf{y}, h_\epsilon(\mathbf{x})) = \frac{1}{n} \sum_{i=1}^n 1 - p(Y_i = y_i | \mathbf{X} = \mathbf{x}; \mathbf{w}).$$

Note that

$$L_1(h, \mathbf{x}, \mathbf{y}) \leq L_{sr}(h, \mathbf{x}, \mathbf{y}) \leq L_{sm}(h, \mathbf{x}, \mathbf{y}).$$

⁸ Since marginal inference is often intractable, exact inference could be replaced with a tractable surrogate, such as the Bethe approximation.

⁹ The additional $\epsilon \Phi^*(\cdot)$ term in Equation 44 is canceled out in Equations 55, 56, 58 and 59.

Marginal inference, $h_\epsilon(\mathbf{x})$, can be decoded by selecting the labels with the highest marginal probabilities. This technique is sometimes referred to as *posterior decoding*. Conveniently, because the marginals sum to one, it can be shown that the Hamming loss of the posterior decoding is at most twice L_1 .

In the following example, we consider the class of soft-max CRFs, \mathcal{H}_{car} . For historical reasons, these models typically do not use edge observations, which is a common modeling decision in, e.g., sequence models. We therefore assume that the edge features are simply $f_{ij}(\mathbf{x}, \mathbf{y}) \triangleq y_i \otimes y_j$.

Example 3. Fix any $m \geq 1$, $n \geq 1$, $\boldsymbol{\pi} \in \Pi(n)$, $\delta \in (0, 1)$ and $G \triangleq (\mathcal{V}, \mathcal{E})$. Assume that $\sup_{x \in \mathcal{X}} \|x\|_2 \leq 1$. Then, with probability at least $1 - \delta$ over realizations of $\mathbf{Z} \triangleq (\mathbf{Z}^{(i)})_{i=1}^m$, for all $h \in \mathcal{H}_{\text{car}}$,

$$\bar{L}_1(h) \leq \hat{L}_{sm}(h, \mathbf{Z}) + \frac{T}{mn} + 4\beta_n \|\Gamma^\boldsymbol{\pi}\|_\infty \sqrt{\frac{\frac{1}{2} \|\mathbf{w}\|_2^2 + \frac{d}{2} \ln(2d(m|G|)^2 \ln(2dmn)) + \ln \frac{4\beta_n}{\delta}}{2mn}},$$

where

$$\beta_n \triangleq 4 \left(\|\mathbf{w}\|_2 + \frac{1}{m|G|} \right) + 1.$$

We omit the proof, since it is almost identical to Example 2. The key difference worth noting is that, since the model does not use edge observations, the graph's maximum degree does not appear in β_n .

6.3 Possibly Unbounded Domains

Until now, we have assumed that the observations are uniformly bounded in the unit ball. This assumption is common in the literature, but it does not quite match what happens in practice. Typically, one will rescale each dimension of the input space using the minimum and maximum values found in the training data. While this procedure guarantees a bound on the observations at training time, the bound may not hold at test time when one rescales by the limits estimated from the training set. This outcome would violate the preconditions of the stability guarantees used to prove the previous examples.

Now, suppose we knew that the observations were bounded with high probability. In the following example, we construct a hypothetical data distribution under which this assumption holds. We combine this with Theorem 2 to derive a variant of Example 2.

Example 4. Fix any $m \geq 1$, $n \geq 1$, $\boldsymbol{\pi} \in \Pi(n)$, $\delta \in (0, 1)$ and $G \triangleq (\mathcal{V}, \mathcal{E})$. Suppose the data generating process, \mathbb{D} , is defined as follows. For each $y \in \mathcal{Y}$, assume there is an associated isotropic Gaussian over $\mathcal{X} \subseteq \mathbb{R}^k$, with mean $\mu_y \in \mathcal{X} : \|\mu_y\|_2 \leq 1$ and variance $\sigma_y^2 \leq (2k \ln(2km^2))^{-1}$. First, \mathbf{Y} is sampled according to some arbitrary distribution, conditioned on G . Then, for each $i \in [n]$, conditioned on $Y_i = y_i$, a vector of observations, $x_i \in \mathcal{X}$, is sampled according to $(\mu_{y_i}, \sigma_{y_i}^2)$.

Note that, conditioned on the labels, (y_1, \dots, y_n) , the observations, (x_1, \dots, x_n) , are mutually independent. It therefore does not make sense to model edge observations, so we use $f_{ij}(\mathbf{x}, \mathbf{y}) \triangleq y_i \otimes y_j$. For the following, we abuse our previous notation and let \mathcal{H}_{max} denote the class of max-margin Markov networks that use these edge features.

Let $\mathcal{B}_Z \triangleq \{\exists i : \|X_i\|_2 \geq 2\}$ denote a set of “bad” inputs, and let $\Gamma_{\frac{\beta_h}{m}}$ denote the dependency matrix induced by \mathbb{D} , $\boldsymbol{\pi}$ and \mathcal{B}_Z . Then, with probability at least $1 - \delta - m/n$ over realizations of $\hat{\mathbf{Z}} \triangleq (\mathbf{Z}^{(i)})_{i=1}^m$, for all $h \in \mathcal{H}_{M,3N}$,

$$\bar{T}_n(h) \leq \hat{I}_n(h, \hat{\mathbf{Z}}) + \frac{11}{mn} + \frac{2}{n} + 4\beta_h \sqrt{\frac{\frac{1}{2} \|\mathbf{w}\|_2^2 + \frac{d}{2} \ln(2d(m|G|)^2 \ln(2dmn)) + \ln \frac{4\beta_h}{\delta}}{2mn}},$$

where

$$\beta_h \triangleq 8 \left(\|\mathbf{w}\|_2 + \frac{1}{m|G|} \right) + 1.$$

The proof is provided in Appendix C.5.

Note that the dominating term is $2/n$, meaning the bound is meaningful for large n and small m . This rate follows intuition, since one should not expect η to depend on the number of training examples; moreover, the probability of drawing a “bad” example should increase proportionally to the number of independent draws.

7. Discussion

We have proposed new PAC-Bayes bounds for structured prediction that can decrease with both the number of examples, m , and the size of each example, n , thus proving that generalization is indeed possible from a few large examples. Under suitable conditions, our bounds can be as tight as $\tilde{O}(1/\sqrt{mn})$. The bounds reveal the connection between generalization and the stability of a structured loss function, as well as the role of dependence in the generating distribution. The stability conditions used in this work generalize our previous work, thereby accommodating a broader range of structured loss functions, including max-margin and soft-max learning. We also provide bounds on the norm of the dependency matrix, which is a result that may be useful outside of this context.

The examples in Section 6 identify several take-aways for practitioners. Primarily, they indicate the importance of templating (or, parameter-tying). Observe that all of the bounds depend on d , the number of parameters¹⁰, via a term that is $\tilde{O}(d/n)$. Clearly, if d scales linearly with n , the number of nodes, then this term is bounded away from zero as $n \rightarrow \infty$. Consequently, one cannot hope to generalize from one example. Though we do not prove this formally, the intuition is fairly simple: if there is a different \mathbf{w}_i for each node i , and \mathbf{w}_{ij} for each edge $\{i, j\}$, then one example provides exactly one “micro example” from which one can estimate $\{\mathbf{w}_i\}_{i \in \mathcal{V}}$ and $\{\mathbf{w}_{ij}\}_{i, j \in \mathcal{E}}$. In this setting, our bounds become $\tilde{O}(1/\sqrt{mn})$, which is no better (and no worse) than previous bounds. Thus, templating is crucial to achieving the fast generalization rate.¹¹

Another observation is that Examples 2 to 4 depend on the norm of the weight vector, \mathbf{w} . Specifically, we used the 2-norm, for its relationship to Gaussian priors; though, one could substitute any norm, due to the equivalence of norms in finite dimension. Dependence

10. We believe that this dependence is unavoidable when derandomizing PAC-Bayes bounds for structured prediction. Evidence to support this conjecture is given by McAllester’s (2007) bound, which depends on the number of templates, and the number of parameters is roughly linear in the number of templates.

11. It may be possible to achieve a fast rate without templating if one imposes a sparsity assumption on the optimal weight vector, but it seems likely that the sparsity would depend on n .

on the norm of the weights is a standard feature of most generalization bounds. This term is commonly interpreted as a measure of hypothesis complexity. Weight regularization during training controls the norm of the weights, thereby effectively limiting the complexity of the learned model.

We also find that the structure of the model influences the bounds via Δ_G , the maximum degree of the graph, and $|G|$, the total number of nodes and edges. (Since the bounds are sub-logarithmic in G , and $\frac{1}{n} \ln |G| \leq \frac{2}{n} \ln n$, one could reasonably argue that Δ_G is the only important structural term.) It is important to note that the edges in the model need not necessarily correspond to concrete relationships in the data. For example, there are many ways to define the “influential” neighbors of a user in a social network, though the user may be connected to nearly everyone in the network; the adjacencies one models may be a subset of the true adjacencies. Therefore, Δ_G and $|G|$ are quantities that one can control; they become part of the trade-off between representational power and overfitting. In light of this trade-off, recall that the stability term, β_h , partially depends on whether one conditions on the observations in the edge features; as shown in Examples 3 and 4, β_h can be reduced to $\tilde{O}(\|\mathbf{w}\|_2)$ if one does not. On the other hand, if observations are modeled in the edge features, and $\Delta_G = O(\sqrt{n})$, then the bounds become $\tilde{O}(1/\sqrt{m})$. Thus, under this modeling assumption, controlling the maximum degree is critical.

Our improved generalization rate critically relies on the dependency matrix, $\Gamma_{\frac{\beta_h}{m}}$, having low infinity norm. If this condition does not hold—for instance, suppose every variable has some non-negligible dependence on every other variable, and $\|\Gamma_{\frac{\beta_h}{m}}\|_\infty = O(n)$ —then our bounds are no more optimistic than previous results and may in fact be slightly looser than some. However, if the dependence is sub-logarithmic, i.e., $\|\Gamma_{\frac{\beta_h}{m}}\|_\infty = O(\ln n)$, then our bounds are much more optimistic. In Section 4.3, we examined two settings in which this assumption holds; these settings can be characterized by the following conditions: strong local signal, bounded interactions (i.e., degree), and dependence that decays with graph distance. Since the data distribution is determined by nature, it is not a variable one can control. There may be situations in which the mixing coefficients can be estimated from data, as done by McDonald et al. (2011) for β -mixing time series. We leave this as a question for future research. Identifying weaker sufficient dependence conditions is also of interest.

There are several ways in which our analysis can be refined and extended. In Lemma 2, which we use to establish the stability of the ramp loss, we used a rather coarse application of Hölder’s inequality to isolate the influence of the weights. This technique ignores the relative magnitudes of the node and edge weights. Indeed, it may be the case that the edge weights are significantly lower than the node weights. A finer analysis of the weights could improve Equation 33 and might yield new insights for weight regularization. One could also abstract the desirable properties of the potential functions to accommodate a broader class than the linear potentials used in our examples. Finally, we conjecture that our bounds could be tightened by adapting Germain et al.’s (2009) analysis to bound $\phi^2(h, \hat{\mathbf{Z}}) \triangleq (\bar{I}(h) - \hat{I}(h, \hat{\mathbf{Z}}))^2$ instead of $\phi(h, \hat{\mathbf{Z}}) \triangleq \bar{I}(\mathbb{Q}) - \hat{I}(\mathbb{Q}, \hat{\mathbf{Z}})$. The primary challenge would be bounding the moment-generating function, $\mathbb{E}_{\mathbf{Z} \sim \mathbb{D}^m} [e^{u \phi^2(h, \hat{\mathbf{Z}})}]$, since our martingale-based method would not work. If successful, this analysis could yield bounds that tighten when the empirical loss is small.

Acknowledgments

This paper is dedicated to the memory of our friend and collaborator, Ben Taskar. This work was supported by the National Science Foundation (NSF), under grant number IIS1218488, and by the Intelligence Advanced Research Projects Activity (IARPA), via Department of Interior National Business Center (DoI/NBC) contract number D12PC00337. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NSF, IARPA, DoI/NBC, or the U.S. Government.

Appendix A. Proofs from Section 4

This appendix contains the deferred proofs from Section 6. We begin with some supplemental background in measure concentration. We then prove Proposition 1, and derive a concentration inequality implied by the result. We conclude with the proofs of Propositions 2 and 3.

A.1 The Method of Bounded Differences

Our proof of Proposition 1 follows McDiarmid’s *method of bounded differences* (McDiarmid, 1989), which uses a construction known as a *Doob martingale difference sequence*. Let $\varphi : \mathcal{Z}^n \rightarrow \mathbb{R}$ denote a measurable function. Let $\mathbf{Z} \triangleq (Z_i)_{i=1}^n$ denote a set of random variables with joint distribution \mathbb{D} , and let $\mu \triangleq \mathbb{E}[\varphi(\mathbf{Z})]$ denote the mean of φ . For $i \in [n]$, let

$$V_i \triangleq \mathbb{E}[\varphi(\mathbf{Z}) \mid \mathbf{Z}_{1:i}] - \mathbb{E}[\varphi(\mathbf{Z}) \mid \mathbf{Z}_{1:i-1}],$$

$$\sum_{i=1}^n V_i = \varphi(\mathbf{Z}) - \mu.$$

where $V_i \triangleq \mathbb{E}[\varphi(\mathbf{Z}) \mid \mathbf{Z}_i] - \mu$. The sequence (V_1, \dots, V_n) has the convenient property that

Therefore, using the law of total expectation, we have that, for any $\tau \in \mathbb{R}$,

$$\begin{aligned} \mathbb{E} \left[e^{\tau(\varphi(\mathbf{Z}) - \mu)} \right] &= \mathbb{E} \left[\prod_{i=1}^n e^{\tau V_i} \right] \\ &= \mathbb{E} \left[\left(\prod_{i=1}^{n-1} e^{\tau V_i} \right) \mathbb{E} \left[e^{\tau V_n} \mid \mathbf{Z}_{1:n-1} \right] \right] \\ &\leq \mathbb{E} \left[\prod_{i=1}^{n-1} e^{\tau V_i} \right] \sup_{\mathbf{z} \in \mathcal{Z}^{n-1}} \mathbb{E} \left[e^{\tau V_n} \mid \mathbf{Z}_{1:n-1} = \mathbf{z} \right] \\ &\vdots \\ &\leq \prod_{i=1}^n \sup_{\mathbf{z} \in \mathcal{Z}^{i-1}} \mathbb{E} \left[e^{\tau V_i} \mid \mathbf{Z}_{1:i-1} = \mathbf{z} \right]. \end{aligned} \quad (45)$$

Note that the order in which we condition on variables is arbitrary, and does not necessarily need to correspond to any spatio-temporal process. The important property is that the

sequence of σ -algebras generated by the conditioned variables are *nested* (McDiarmid (1998) called this a *filter*), which is guaranteed by the construction of (V_1, \dots, V_n) .

One can then use Hoeffding’s lemma (Hoeffding, 1963) to bound each term in the above product.

Lemma 5. *If ξ is a random variable, such that $\mathbb{E}[\xi] = 0$ and $a \leq \xi \leq b$ almost surely, then for any $\tau \in \mathbb{R}$,*

$$\mathbb{E} \left[e^{\tau \xi} \right] \leq \exp \left(\frac{\tau^2 (b-a)^2}{8} \right).$$

Clearly, $\mathbb{E}[V_i \mid \mathbf{Z}_{1:i-1}] = 0$. Thus, if, for all $i \in [n]$, there exists a value $c_i \geq 0$ such that

$$\sup_{\mathbf{z} \in \mathcal{Z}^{i-1}} \sup_{z \in \mathcal{Z}} (V_i - \inf_{z \in \mathcal{Z}} V_i) = \sup_{\mathbf{z} \in \mathcal{Z}^{i-1}} \mathbb{E}[\varphi(\mathbf{Z}) \mid \mathbf{Z}_{1:i} = (\mathbf{z}, z)] - \mathbb{E}[\varphi(\mathbf{Z}) \mid \mathbf{Z}_{1:i} = (\mathbf{z}, z')] \leq c_i,$$

then

$$\mathbb{E} \left[e^{\tau(\varphi(\mathbf{Z}) - \mu)} \right] \leq \prod_{i=1}^n \exp \left(\frac{\tau^2 c_i^2}{8} \right) = \exp \left(\frac{\tau^2}{8} \sum_{i=1}^n c_i^2 \right).$$

When Z_1, \dots, Z_n are mutually independent, and φ has β -uniformly stability, upper-bounding c_i is straightforward: it becomes complicated when we relax the independence assumption, or when φ is not uniformly stable. The following section addresses the former challenge.

A.2 Coupling

To analyze interdependent random variables, we use a theoretical construction known as *coupling*. For random variables Z_1 and Z_2 , with respective distributions \mathbb{D}_1 and \mathbb{D}_2 over a common sample space \mathcal{Z} , a coupling is any joint distribution \mathbb{D} over $\mathcal{Z} \times \mathcal{Z}$ such that the marginal distributions, $\mathbb{D}(Z_1)$ and $\mathbb{D}(Z_2)$, are equal to $\mathbb{D}_1(Z_1)$ and $\mathbb{D}_2(Z_2)$ respectively.

Using a construction due to Fiebig (1993), one can create a coupling of two sequences of random variables, such that the probability that any two corresponding variables are different is upper-bounded by the θ -mixing coefficients in Definition 6. The following is an adaptation of this result (due to Samson, 2000) for continuous domains.

Lemma 6. *Let $\mathbf{Z}^{(1)} \triangleq (Z_i^{(1)})_{i=1}^n$ and $\mathbf{Z}^{(2)} \triangleq (Z_i^{(2)})_{i=1}^n$ be random variables with respective distributions \mathbb{D}_1 and \mathbb{D}_2 over a sample space \mathcal{Z}^n . Then there exists a coupling \mathbb{D} , with marginal distributions $\mathbb{D}(\mathbf{Z}^{(1)}) = \mathbb{D}_1(\mathbf{Z}^{(1)})$ and $\mathbb{D}(\mathbf{Z}^{(2)}) = \mathbb{D}_2(\mathbf{Z}^{(2)})$, such that, for any $i \in [n]$,*

$$\Pr_{(\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}) \sim \mathbb{D}} \left\{ Z_i^{(1)} \neq Z_i^{(2)} \right\} \leq \left\| \mathbb{D}_1(\mathbf{Z}_{1:n}^{(1)}) - \mathbb{D}_2(\mathbf{Z}_{1:n}^{(2)}) \right\|_{TV},$$

where $\Pr_{(\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}) \sim \mathbb{D}} \{ Z_i^{(1)} \neq Z_i^{(2)} \}$ denotes the marginal probability that $Z_i^{(1)} \neq Z_i^{(2)}$ under \mathbb{D} .

Note that the requirement of strictly positive densities is not restrictive, since one can always construct a positive density from a simply nonnegative one. We defer to Samson (2000) for details.

We are now equipped with the tools to prove Proposition 1.

A.3 Proof of Proposition 1

Conditioned on $\bar{\mathbf{B}}$, every realization of \mathbf{Z} is in the “good” set. We define a Doob martingale difference sequence, using the filtration π :

$$V_i^\pi \triangleq \mathbb{E}[\varphi(\mathbf{Z}) | \bar{\mathbf{B}}, \mathbf{Z}_{\pi_i(1:i)}] - \mathbb{E}[\varphi(\mathbf{Z}) | \bar{\mathbf{B}}, \mathbf{Z}_{\pi_i(1:i-1)}],$$

where $V_1^\pi \triangleq \mathbb{E}[\varphi(\mathbf{Z}) | \bar{\mathbf{B}}, \mathbf{Z}_{\pi_1(1)}] - \mathbb{E}[\varphi(\mathbf{Z}) | \bar{\mathbf{B}}]$. Note that $\mathbb{E}[V_i^\pi | \bar{\mathbf{B}}] = 0$ and, for $\mathbf{Z} \notin \mathcal{B}_Z$,

$$\sum_{i=1}^n V_i^\pi = \varphi(\mathbf{Z}) - \mathbb{E}[\varphi(\mathbf{Z}) | \bar{\mathbf{B}}].$$

We therefore have, via Equation 45, that

$$\mathbb{E} \left[e^{\tau(\varphi(\mathbf{Z}) - \mathbb{E}[\varphi(\mathbf{Z}) | \bar{\mathbf{B}}])} | \bar{\mathbf{B}} \right] \leq \prod_{i=1}^n \sup_{\mathbf{z} \in \mathcal{Z}_{\pi_i}^{i-1}} \mathbb{E} \left[e^{\tau V_i^\pi} | \bar{\mathbf{B}}, \mathbf{Z}_{\pi_i(1:i-1)} = \mathbf{z} \right],$$

where the supremum over $\mathcal{Z}_{\pi_i}^{i-1}$ comes from the fact that the expectations are conditioned on $\bar{\mathbf{B}}$. Recall that each permutation in π has the same prefix, thus preserving the order of conditioned variables, and ensuring that the sequence of σ -algebras is nested.

What remains is to show that, for all $i \in [n]$,

$$\begin{aligned} & \sup_{\mathbf{z} \in \mathcal{Z}_{\pi_i}^{i-1}} \sup_{\mathbf{z}' \in \mathcal{Z}_{\pi_i}^i} (V_i^\pi) - \inf_{\mathbf{z}' \in \mathcal{Z}_{\pi_i}^i} (V_i^\pi) \\ &= \sup_{\mathbf{z} \in \mathcal{Z}_{\pi_i}^{i-1}} \mathbb{E}[\varphi(\mathbf{Z}) | \bar{\mathbf{B}}, \mathbf{Z}_{\pi_i(1:i)} = (\mathbf{z}, \mathbf{z})] - \mathbb{E}[\varphi(\mathbf{Z}) | \bar{\mathbf{B}}, \mathbf{Z}_{\pi_i(1:i)} = (\mathbf{z}, \mathbf{z}')] \end{aligned} \quad (46)$$

is bounded, so as to apply Lemma 5. (Again, the suprema over $\mathcal{Z}_{\pi_i}^i(\mathbf{z})$ stem from conditioning on $\bar{\mathbf{B}}$.) To do so, we will use the coupling construction from Lemma 6. Fix any $\mathbf{z} \in \mathcal{Z}_{\pi_i}^{i-1}$ and $\mathbf{z}', \mathbf{z}' \in \mathcal{Z}_{\pi_i}^i(\mathbf{z})$, and let $N \triangleq n - i$. Define random variables $\xi^{(1)} \triangleq (\xi_j^{(1)})_{j=1}^N$ and $\xi^{(2)} \triangleq (\xi_j^{(2)})_{j=1}^N$, with coupling distribution $\hat{\mathbb{D}}$ such that

$$\begin{aligned} \hat{\mathbb{D}}(\xi^{(1)}) &\triangleq \mathbb{D}(\mathbf{Z}_{\pi_i(i+1:n)} | \bar{\mathbf{B}}, \mathbf{Z}_{\pi_i(1:i)} = (\mathbf{z}, \mathbf{z})) \\ \text{and } \hat{\mathbb{D}}(\xi^{(2)}) &\triangleq \mathbb{D}(\mathbf{Z}_{\pi_i(i+1:n)} | \bar{\mathbf{B}}, \mathbf{Z}_{\pi_i(1:i)} = (\mathbf{z}, \mathbf{z}')). \end{aligned} \quad (47)$$

In other words, the marginal distributions of $\xi^{(1)}$ and $\xi^{(2)}$ are equal to the conditional distributions of $\mathbf{Z}_{\pi_i(i+1:n)}$ given $\bar{\mathbf{B}}$ and, respectively, $\mathbf{Z}_{\pi_i(1:i)} = (\mathbf{z}, \mathbf{z})$ or $\mathbf{Z}_{\pi_i(1:i)} = (\mathbf{z}, \mathbf{z}')$. Note that we have renumbered the coupled variables according to π_i . This does not affect the distribution, but it does affect how we later apply Lemma 6. Denote by π_i^{-1} the inverse of π_i (i.e., $\pi_i^{-1}(\pi_i(1:n)) = [n]$), and let

$$\psi(\mathbf{z}) = \varphi(\mathbf{z}_{\pi_i^{-1}(1:n)}).$$

Put simply, ψ inverts the permutation applied to its input, so as to ensure $\psi(\mathbf{z}_{\pi_i(1:n)}) = \varphi(\mathbf{z})$. For convenience, let

$$\Delta\psi \triangleq \psi(\mathbf{z}, \mathbf{z}, \xi^{(1)}) - \psi(\mathbf{z}, \mathbf{z}', \xi^{(2)})$$

denote the difference. Using these definitions, we have the following equivalence:

$$\mathbb{E}[\varphi(\mathbf{Z}) | \bar{\mathbf{B}}, \mathbf{Z}_{\pi_i(1:i)} = (\mathbf{z}, \mathbf{z})] - \mathbb{E}[\varphi(\mathbf{Z}) | \bar{\mathbf{B}}, \mathbf{Z}_{\pi_i(1:i)} = (\mathbf{z}, \mathbf{z}')] = \mathbb{E}[\psi(\mathbf{z}, \mathbf{z}, \xi^{(1)}) - \psi(\mathbf{z}, \mathbf{z}', \xi^{(2)})].$$

Because the expectations are conditioned on $\bar{\mathbf{B}}$, both realizations, $(\mathbf{z}, \mathbf{z}, \xi^{(1)})$ and $(\mathbf{z}, \mathbf{z}', \xi^{(2)})$, are “good,” in the sense that Equation 1 holds. We therefore have that

$$\begin{aligned} & \mathbb{E}[\psi(\mathbf{z}, \mathbf{z}, \xi^{(1)}) - \psi(\mathbf{z}, \mathbf{z}', \xi^{(2)})] \leq \beta \mathbb{E} \left[D_{\text{H}}((\mathbf{z}, \mathbf{z}, \xi^{(1)}), (\mathbf{z}, \mathbf{z}', \xi^{(2)})) \right] \\ & \leq \beta \left(1 + \mathbb{E} \left[\sum_{j=1}^N \mathbb{1}\{\xi_j^{(1)} \neq \xi_j^{(2)}\} \right] \right) \\ & = \beta \left(1 + \sum_{j=1}^N \Pr_{(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}) \sim \hat{\mathbb{D}}} \left\{ \xi_j^{(1)} \neq \xi_j^{(2)} \right\} \right). \end{aligned}$$

In the second inequality, we assumed that $\mathbf{z} \neq \mathbf{z}'$. Recall from Lemma 6 and Definition 6 that

$$\begin{aligned} & 1 + \sum_{j=1}^N \Pr_{(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}) \sim \hat{\mathbb{D}}} \left\{ \xi_j^{(1)} \neq \xi_j^{(2)} \right\} \\ & \leq 1 + \sum_{j=i+1}^n \|\mathbb{D}(\mathbf{Z}_{\pi_i(j:n)} | \bar{\mathbf{B}}, \mathbf{Z}_{\pi_i(1:i)} = (\mathbf{z}, \mathbf{z})) - \mathbb{D}(\mathbf{Z}_{\pi_i(j:n)} | \bar{\mathbf{B}}, \mathbf{Z}_{\pi_i(1:i)} = (\mathbf{z}, \mathbf{z}'))\|_{\text{TV}} \\ & = 1 + \sum_{j=i+1}^n \vartheta_{ij}^\pi(\mathbf{z}, \mathbf{z}, \mathbf{z}') \\ & \leq 1 + \sum_{j=i+1}^n \gamma_{ij}^\pi = \sum_{j=i}^n \gamma_{ij}^\pi. \end{aligned}$$

The above inequalities hold uniformly for all $\mathbf{z} \in \mathcal{Z}_{\pi_i}^{i-1}$ and $\mathbf{z}, \mathbf{z}' \in \mathcal{Z}_{\pi_i}^i(\mathbf{z})$; thus,

$$\sup_{\mathbf{z} \in \mathcal{Z}_{\pi_i}^{i-1}} \sup_{\mathbf{z}' \in \mathcal{Z}_{\pi_i}^i(\mathbf{z})} (V_i^\pi) - \inf_{\mathbf{z}' \in \mathcal{Z}_{\pi_i}^i(\mathbf{z})} (V_i^\pi) \leq \beta \sum_{j=i}^n \gamma_{ij}^\pi.$$

Then, since we have identified a uniform upper bound for Equation 46, we apply Lemma 5 and obtain

$$\begin{aligned} & \mathbb{E} \left[e^{\tau(\varphi(\mathbf{Z}) - \mathbb{E}[\varphi(\mathbf{Z}) | \bar{\mathbf{B}}])} | \bar{\mathbf{B}} \right] \leq \exp \left(\frac{\tau^2}{8} \sum_{i=1}^n \left(\beta \sum_{j=i}^n \gamma_{ij}^\pi \right)^2 \right) \\ & \leq \exp \left(\frac{\tau^2}{8} n \beta^2 \max_{i \in [n]} \left(\sum_{j=i}^n \gamma_{ij}^\pi \right)^2 \right) \\ & = \exp \left(\frac{\tau^2}{8} n \beta^2 \|\Gamma_{\bar{\mathbf{B}}}^\pi\|_\infty^2 \right), \end{aligned}$$

which completes the proof.

A.4 A New Concentration Inequality

Proposition 1, implies the following concentration inequality, which may be of independent interest.

Corollary 3. *Let $\mathbf{Z} \triangleq (Z_i)_{i=1}^n$ denote random variables with joint distribution \mathbb{D} on \mathcal{Z}^n , and let $\varphi : \mathcal{Z}^n \rightarrow \mathbb{R}$ denote a measurable function. If φ is β -uniformly stable, then, for any $\epsilon > 0$ and $\boldsymbol{\pi} \in \Pi(n)$,*

$$\Pr\{\varphi(\mathbf{Z}) - \mathbb{E}[\varphi(\mathbf{Z})] \geq \epsilon\} \leq \exp\left(\frac{-2\epsilon^2}{n\beta^2 \|\boldsymbol{\Gamma}^\boldsymbol{\pi}\|_\infty^2}\right).$$

Proof First, note that, for any $\tau \in \mathbb{R}$,

$$\Pr\{\varphi(\mathbf{Z}) - \mathbb{E}[\varphi(\mathbf{Z})] \geq \epsilon\} = \Pr\left\{e^{\tau(\varphi(\mathbf{Z}) - \mathbb{E}[\varphi(\mathbf{Z})])} \geq e^{\tau\epsilon}\right\},$$

due to the monotonicity of exponentiation. We then apply Markov's inequality and obtain

$$\Pr\left\{e^{\tau(\varphi(\mathbf{Z}) - \mathbb{E}[\varphi(\mathbf{Z})])} \geq e^{\tau\epsilon}\right\} \leq \frac{1}{e^{\tau\epsilon}} \mathbb{E}\left[e^{\tau(\varphi(\mathbf{Z}) - \mathbb{E}[\varphi(\mathbf{Z})])}\right].$$

Since φ has β -uniform stability, we can apply Proposition 1 by taking $\mathcal{B}\mathcal{Z} \triangleq \emptyset$. Thus,

$$\Pr\left\{e^{\tau(\varphi(\mathbf{Z}) - \mathbb{E}[\varphi(\mathbf{Z})])} \geq e^{\tau\epsilon}\right\} \leq \frac{1}{e^{\tau\epsilon}} \exp\left(\frac{\tau^2}{8} n \beta^2 \|\boldsymbol{\Gamma}^\boldsymbol{\pi}\|_\infty^2\right).$$

Optimizing with respect to τ , we take $\tau \triangleq \frac{4\epsilon}{n\beta^2 \|\boldsymbol{\Gamma}^\boldsymbol{\pi}\|_\infty^2}$ to complete the proof. \blacksquare

Corollary 3 extends some current state-of-the-art results (e.g., Kontorovich and Razmanan, 2008, Theorem 1.1) by supporting filtrations of the mixing coefficients. Further, when Z_1, \dots, Z_n are mutually independent (i.e., $\|\boldsymbol{\Gamma}^\boldsymbol{\pi}\|_\infty = 1$), we recover McDiarmid's inequality.

A.5 Proof of Proposition 2

We construct the filtration $\boldsymbol{\pi}$ recursively. We initialize π_1 using a breadth-first traversal of the graph, starting from any node. Then, for $i = 2, \dots, n$, we set $\pi_i(1 : i-1) \triangleq \pi_{i-1}(1 : i-1)$, and determine $\pi_i(i : n)$ using a breadth-first traversal over the induced subgraph of $\pi_{i-1}(i : n)$, starting from $\pi_{i-1}(i-1)$. This ensures that nodes closer to $\pi_i(i)$ appear earlier in the permutation, so that the higher mixing coefficients are not incurred for all $j = i+1, \dots, n$.

The degree of any node in this induced subgraph is at most the maximum degree of the whole graph, Δ_G , so the number of nodes at distance k from node $\pi_i(i)$ is at most Δ_G^k . Hence, the number of subsets, $\pi_i(j : n) : j > i$, at distance k from $\pi_i(i)$ is at most Δ_G^k . Therefore,

$$\sum_{j=i}^n \gamma_{ij}^\boldsymbol{\pi} \leq \sum_{k=0}^{\infty} \Delta_G^k \theta(k) \leq \sum_{k=0}^{\infty} \left(\frac{\Delta_G}{\Delta_G + \epsilon}\right)^k.$$

Since $\Delta_G/(\Delta_G + \epsilon) < 1$ for $\epsilon > 0$, this geometric series converges to

$$\frac{1}{1 - \Delta_G/(\Delta_G + \epsilon)} = 1 + \Delta_G/\epsilon,$$

which completes the proof.

A.6 Proof of Proposition 3

For a chain graph, we define each permutation uniformly as $\pi_i \triangleq [n]$. Each upper-triangular entry of $\boldsymbol{\Gamma}^\boldsymbol{\pi}$ then satisfies $\gamma_{ij}^\boldsymbol{\pi} \leq \theta(j-i)$. The number of unconditioned variables at distance $k = j-i$ is exactly one. Thus, for any row i ,

$$\sum_{j=i}^n \gamma_{ij}^\boldsymbol{\pi} \leq 1 + \sum_{k=1}^{n-i} \theta(k) \leq 1 + \epsilon \sum_{k=1}^{n-i} k^{-p}.$$

For $p = 1$, $(k^{-p})_{k=1}^{\infty}$ is a Harmonic series. Thus, the partial sum, $\sum_{k=1}^{n-i} k^{-p}$, is the $(n-i)$ th Harmonic number, which is upper-bounded by $\ln(n-i) + 1$, and maximized at row $i = 1$. For $p > 1$,

$$1 + \epsilon \sum_{k=1}^{n-i} k^{-p} \leq 1 + \epsilon \sum_{k=1}^{\infty} k^{-p} = 1 + \zeta(p),$$

by definition.

Appendix B. Proofs from Section 5

This appendix contains the deferred proofs from Section 5.

B.1 Proof of Theorem 2

For $i = 0, 1, 2, \dots$, let $\beta_i \triangleq 2^{i+1}$. Since Equation 2 falls with probability $\delta + m\nu$, we could simply invoke Theorem 1 for each β_i with $\delta_i \triangleq \beta_i^{-1}(\delta + m\nu)$. This approach would introduce an additional $O(\ln(m\nu)^{-1})$ term in the numerator of Equation 23. We therefore choose instead to cover β and u simultaneously. Accordingly, for $j = 0, 1, 2, \dots$, let

$$u_j \triangleq 2^j \sqrt{\frac{8mn \ln \frac{2\beta_j}{\delta}}{\beta_j^2 \|\boldsymbol{\Gamma}^\beta\|_\infty^2}}.$$

Each β_j defines a set of ‘‘bad’’ hypotheses, $\mathcal{B}_j^{u_j}$, which we use in Equation 5 to define a function $\tilde{\phi}_j$. Let $\tilde{\phi}_j \triangleq \delta \beta_j^{-1} 2^{-U+1}$, and define an event

$$E_{ij} \triangleq \mathbf{1} \left\{ \mathbb{E}_{h \sim \mathcal{D}^p} \left[e^{u_j \tilde{\phi}_j(h; \mathbf{Z})} \right] \geq \frac{1}{\delta_{ij}} \exp\left(\frac{u_j^2 \beta_j^2 \|\boldsymbol{\Gamma}^\beta\|_\infty^2}{8mn}\right) \right\}.$$

Note that none of the above depend on $(\beta, \eta, \mathcal{Q})$. Using the event B defined in Equation 12, we have, via Proposition 1, that

$$\Pr_{\mathcal{Z} \sim \mathcal{D}^m} \{E_{ij} | \neg B\} \leq \delta_{ij} \exp\left(-\frac{u_j^2 \beta_j^2 \|\boldsymbol{\Gamma}^\beta\|_\infty^2}{8mn}\right) \mathbb{E}_{h \sim \mathcal{D}^p} \mathbb{E}_{\mathcal{Z} \sim \mathcal{D}^m} \left[e^{u_j \tilde{\phi}_j(h; \mathbf{Z})} | \neg B \right] \leq \delta_{ij}.$$

Then, using the same reasoning as Equation 13, with $E \triangleq \bigcup_{j=0}^{\infty} E_{ij}$,

$$\begin{aligned} \Pr_{\mathbf{Z} \sim \mathbb{D}^{nm}} \{B \cup E\} &\leq m\nu + \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \Pr_{\mathbf{Z} \sim \mathbb{D}^{nm}} \{E_{ij} | \neg B\} \\ &\leq m\nu + \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \delta_{ij} \\ &= m\nu + \delta \sum_{i=0}^{\infty} \beta_i^{-1} \sum_{j=0}^{\infty} 2^{-(j+1)} \\ &= m\nu + \delta \sum_{i=0}^{\infty} 2^{-(i+1)} \sum_{j=0}^{\infty} 2^{-(j+1)} \\ &= m\nu + \delta. \end{aligned}$$

Therefore, with probability at least $1 - \delta - m\nu$, every $l \in [m]$ satisfies $\mathbf{Z}^{(l)} \notin \mathcal{B}_{\mathbf{Z}}$, and every (i, j) satisfies

$$\mathbb{E}_{h \sim \mathbb{P}} \left[e^{u_{ij} \tilde{\phi}_i(h; \mathbf{z})} \right] \leq \frac{1}{\delta_{ij}} \exp \left(\frac{u_{ij}^2 \beta_i^2 \|\Gamma_{\mathbb{F}}\|_{\infty}^2}{8mn} \right). \quad (48)$$

Observe that $(\beta/n, \mathcal{B}_{\mathbf{Z}}, \eta)$ -local stability implies $(\beta_i/n, \mathcal{B}_{\mathbf{Z}}, \eta)$ -local stability for all $\beta_j \geq \beta$. Therefore, for any particular $(\beta, \eta, \mathbb{Q})$ such that \mathbb{Q} is $(\beta/n, \mathcal{B}_{\mathbf{Z}}, \eta)$ -locally stable, we select $i^* \triangleq \lfloor (\ln 2)^{-1} \ln \beta \rfloor$. This ensures that $\beta \leq \beta_{i^*}$, so \mathbb{Q} also satisfies $(\beta_{i^*}/n, \mathcal{B}_{\mathbf{Z}}, \eta)$ -local stability. Then, letting

$$j^* \triangleq \left\lfloor \frac{1}{2 \ln 2} \ln \left(\frac{D_{\text{kl}}(\mathbb{Q} \|\mathbb{P})}{\ln(2\beta_{i^*}/\delta)} + 1 \right) \right\rfloor,$$

we have that

$$\frac{1}{2} \sqrt{\frac{8mn \left(D_{\text{kl}}(\mathbb{Q} \|\mathbb{P}) + \ln \frac{2\beta_{i^*}}{\delta} \right)}{\beta_{i^*}^2 \|\Gamma_{\mathbb{F}}\|_{\infty}^2}} \leq u_{i^*, j^*} \leq \sqrt{\frac{8mn \left(D_{\text{kl}}(\mathbb{Q} \|\mathbb{P}) + \ln \frac{2\beta_{i^*}}{\delta} \right)}{\beta_{i^*}^2 \|\Gamma_{\mathbb{F}}\|_{\infty}^2}}. \quad (49)$$

Moreover,

$$\begin{aligned} D_{\text{kl}}(\mathbb{Q} \|\mathbb{P}) + \ln \frac{1}{\delta_{i^*, j^*}} &\leq D_{\text{kl}}(\mathbb{Q} \|\mathbb{P}) + \ln \frac{2\beta_{i^*}}{\delta} + \frac{1}{2} \ln \left(\frac{D_{\text{kl}}(\mathbb{Q} \|\mathbb{P})}{\ln(2\beta_{i^*}/\delta)} + 1 \right) \\ &\leq D_{\text{kl}}(\mathbb{Q} \|\mathbb{P}) + \ln \frac{2\beta_{i^*}}{\delta} + \frac{1}{2} \left(D_{\text{kl}}(\mathbb{Q} \|\mathbb{P}) + \ln \frac{2\beta_{i^*}}{\delta} \right). \end{aligned} \quad (50)$$

Thus, with probability at least $1 - \delta - m\nu$,

$$\begin{aligned} \bar{L}(\mathbb{Q}) - \hat{L}(\mathbb{Q}; \hat{\mathbf{Z}}) &\leq \alpha(\eta + \nu) + \frac{1}{u_{i^*, j^*}} \left(D_{\text{kl}}(\mathbb{Q} \|\mathbb{P}) + \ln \mathbb{E}_{h \sim \mathbb{P}} \left[e^{u_{i^*, j^*} \tilde{\phi}_{i^*}(h; \hat{\mathbf{z}})} \right] \right) \\ &\leq \alpha(\eta + \nu) + \frac{1}{u_{i^*, j^*}} \left(D_{\text{kl}}(\mathbb{Q} \|\mathbb{P}) + \ln \frac{1}{\delta_{i^*, j^*}} + \frac{u_{i^*, j^*}^2 \beta_{i^*}^2 \|\Gamma_{\mathbb{F}}\|_{\infty}^2}{8mn} \right) \\ &\leq \alpha(\eta + \nu) + \frac{3 \left(D_{\text{kl}}(\mathbb{Q} \|\mathbb{P}) + \ln \frac{2\beta_{i^*}}{\delta} \right)}{2u_{i^*, j^*}} + \frac{8mn}{8mn} \\ &\leq \alpha(\eta + \nu) + 2\beta_{i^*} \sqrt{\frac{D_{\text{kl}}(\mathbb{Q} \|\mathbb{P}) + \ln \frac{2\beta_{i^*}}{\delta}}{2mn}}. \end{aligned}$$

The first inequality uses Equation 9; the second uses Equation 48; the third and fourth use Equations 49 and 50. Noting that $\beta_{i^*} \leq 2\beta$ completes the proof.

B.2 Proof of Proposition 5

Fix any $h \in \mathcal{H}$ and $\mathbf{z} \notin \mathcal{B}_{\mathbf{Z}}$. By Definition 10, there exists a set $\mathcal{B}_{\mathcal{H}}(h)$ with measure $\mathbb{Q}_h(\mathcal{B}_{\mathcal{H}}(h)) \leq \eta$. For any $\mathbf{z} \notin \mathcal{B}_{\mathbf{Z}}$, let $\mathcal{B}_{\mathcal{H}}(h, \mathbf{z}) \triangleq \mathcal{B}_{\mathcal{H}}(h)$, and note that $\mathbb{Q}_h(\mathcal{B}_{\mathcal{H}}(h, \mathbf{z})) \leq \eta$ as well. Further, for any $h' \notin \mathcal{B}_{\mathcal{H}}(h, \mathbf{z})$, $\|h - h'\| \leq \beta$. Thus, by Definition 9,

$$\|L(h, \mathbf{z}) - L(h', \mathbf{z})\| \leq \lambda \|h - h'\| \leq \lambda \beta,$$

which completes the proof.

Appendix C. Proofs from Section 6

This appendix contains the deferred proofs from Section 6. Certain proofs require the following technical lemmas, which apply to the linear feature functions defined in Section 2.2.3.

Lemma 7. Fix a graph, $G \triangleq (V, \mathcal{E})$, with maximum degree Δ_G . Suppose \mathcal{X} is uniformly bounded by the p -norm ball with radius R ; i.e., $\sup_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x}\|_p \leq R$. Then, for any $\mathbf{x}, \mathbf{x}' \in \mathcal{X}^n$ and $\mathbf{y} \in \mathcal{Y}^n$,

$$\|\mathbf{f}(\mathbf{x}, \mathbf{y}) - \mathbf{f}(\mathbf{x}', \mathbf{y})\|_p \leq (\Delta_G + 2)RD_{\mathcal{H}}(\mathbf{x}, \mathbf{x}'). \quad (51)$$

Further, if the model does not use edge observations (i.e., $f_{ij}(\mathbf{x}, \mathbf{y}) \triangleq y_i \otimes y_j$), then

$$\|\mathbf{f}(\mathbf{x}, \mathbf{y}) - \mathbf{f}(\mathbf{x}', \mathbf{y})\|_p \leq 2RD_{\mathcal{H}}(\mathbf{x}, \mathbf{x}'). \quad (52)$$

Proof We start by considering a pair, $\mathbf{x}, \mathbf{x}' \in \mathcal{X}^n : D_{\mathcal{H}}(\mathbf{x}, \mathbf{x}') = 1$, that differ at a single coordinate, corresponding to a node i . This means that the aggregate features differ at one local feature, and any edge involving i . Thus, using the triangle inequality, we have that

$$\begin{aligned} \|\mathbf{f}(\mathbf{x}, \mathbf{y}) - \mathbf{f}(\mathbf{x}', \mathbf{y})\|_p &= \left\| \left[\begin{array}{c} f_i(\mathbf{x}, \mathbf{y}) - f_i(\mathbf{x}', \mathbf{y}) \\ \sum_{j:(i,j) \in \mathcal{E}} f_{ij}(\mathbf{x}, \mathbf{y}) - f_{ij}(\mathbf{x}', \mathbf{y}) \end{array} \right] \right\|_p \\ &\leq \|f_i(\mathbf{x}, \mathbf{y}) - f_i(\mathbf{x}', \mathbf{y})\|_p + \sum_{j:(i,j) \in \mathcal{E}} \|f_{ij}(\mathbf{x}, \mathbf{y}) - f_{ij}(\mathbf{x}', \mathbf{y})\|_p. \end{aligned} \quad (53)$$

Note that the second term disappears when the model does not use edge observations.

Recall that the features are defined using a Kronecker product. For any vectors \mathbf{u}, \mathbf{v} , $\|\mathbf{u} \otimes \mathbf{v}\|_p = \|\mathbf{u}\|_p \|\mathbf{v}\|_p$. Using this identity, and the fact that each $g \in \mathcal{Y}$ has $\|g\|_1 = 1$, we have that

$$\begin{aligned} \|f_i(\mathbf{x}, \mathbf{y}) - f_i(\mathbf{x}', \mathbf{y})\|_p &= \|(x_i - x'_i) \otimes y_i\|_p \\ &= \|x_i - x'_i\|_p \|y_i\|_p \\ &\leq \left(\|x_i\|_p + \|x'_i\|_p \right) \times 1 \\ &\leq 2R, \end{aligned}$$

and

$$\begin{aligned} \|f_{ij}(\mathbf{x}, \mathbf{y}) - f_{ij}(\mathbf{x}', \mathbf{y})\|_p &= \left\| \frac{1}{2} \begin{bmatrix} x_i \\ x_j \end{bmatrix} - \begin{bmatrix} x'_i \\ x'_j \end{bmatrix} \right\|_p \otimes (y_i \otimes y_j) \Big\|_p \\ &= \frac{1}{2} \|x_i - x'_i\|_p \|y_i\|_p \|y_j\|_p \\ &\leq \frac{1}{2} \left(\|x_i\|_p + \|x'_i\|_p \right) \times 1 \times 1 \\ &\leq R. \end{aligned}$$

Combining these inequalities with Equation 53, and using the fact that i participates in at most Δ_G edges, we have that

$$\|\mathbf{f}(\mathbf{x}, \mathbf{y}) - \mathbf{f}(\mathbf{x}', \mathbf{y})\|_p \leq 2R + \sum_{j:\{i,j\} \in \mathcal{E}} R \leq (2 + \Delta_G)R.$$

For no edge observations, the righthand side is simply $2R$. Thus, since the bounds hold for any single coordinate perturbation, Equations 51 and 52 follow from the triangle inequality. ■

Lemma 8. For a graph, $G \triangleq (\mathcal{V}, \mathcal{E})$, and recall that $|G| \triangleq |\mathcal{V}| + |\mathcal{E}|$. Suppose \mathcal{X} is uniformly bounded by the p -norm ball with radius R ; i.e., $\sup_{\mathbf{x} \in \mathcal{X}} \|\mathbf{x}\|_p \leq R$. Then, for all $\mathbf{x} \in \mathcal{X}^n$ and $\mathbf{y} \in \mathcal{Y}^n$,

$$\|\mathbf{f}(\mathbf{x}, \mathbf{y})\|_p \leq |G| R.$$

Proof Invoking the triangle inequality, we have that

$$\begin{aligned} \|\mathbf{f}(\mathbf{x}, \mathbf{y})\|_p &= \left\| \left[\sum_{i \in \mathcal{V}} f_i(\mathbf{x}, \mathbf{y}) \right] \right\|_p \\ &\leq \sum_{i \in \mathcal{V}} \|f_i(\mathbf{x}, \mathbf{y})\|_p + \sum_{\{i,j\} \in \mathcal{E}} \|f_{ij}(\mathbf{x}, \mathbf{y})\|_p \\ &= \sum_{i \in \mathcal{V}} \|x_i \otimes y_i\|_p + \sum_{\{i,j\} \in \mathcal{E}} \left\| \frac{1}{2} \begin{bmatrix} x_i \\ x_j \end{bmatrix} \otimes (y_i \otimes y_j) \right\|_p \\ &= \sum_{i \in \mathcal{V}} \|x_i\|_p \|y_i\|_p + \sum_{\{i,j\} \in \mathcal{E}} \frac{1}{2} \left\| \begin{bmatrix} x_i \\ x_j \end{bmatrix} \right\|_p \|y_i\|_p \|y_j\|_p \\ &\leq \sum_{i \in \mathcal{V}} \|x_i\|_p \|y_i\|_p + \sum_{\{i,j\} \in \mathcal{E}} \frac{1}{2} \left(\|x_i\|_p + \|x_j\|_p \right) \|y_i\|_p \|y_j\|_p \\ &\leq \sum_{i \in \mathcal{V}} R \times 1 + \sum_{\{i,j\} \in \mathcal{E}} \frac{1}{2} (R + R) \times 1 \times 1 \\ &= (|\mathcal{V}| + |\mathcal{E}|)R = |G|R, \end{aligned}$$

which completes the proof. ■

Note that Lemmas 7 and 8 hold when discrete labels are replaced with marginals, since each clique's marginals sum to one. This adaptation enables the proof of Example 3.

C.1 Proof of Lemma 2

To simplify notation, let:

$$\begin{aligned} \mathbf{y}_1 &\triangleq \arg \max_{\mathbf{u} \in \mathcal{Y}^n} D_H(\mathbf{y}, \mathbf{u}) + h(\mathbf{x}, \mathbf{u}); & \mathbf{y}_2 &\triangleq \arg \max_{\mathbf{u} \in \mathcal{Y}^n} h(\mathbf{x}, \mathbf{u}); \\ \mathbf{y}'_1 &\triangleq \arg \max_{\mathbf{u} \in \mathcal{Y}^n} D_H(\mathbf{y}', \mathbf{u}) + h(\mathbf{x}', \mathbf{u}); & \mathbf{y}'_2 &\triangleq \arg \max_{\mathbf{u} \in \mathcal{Y}^n} h(\mathbf{x}', \mathbf{u}). \end{aligned}$$

Using this notation, we have that

$$\begin{aligned} n |L_n(h, \mathbf{z}) - L_n(h, \mathbf{z}')| &= |D_H(\mathbf{y}, \mathbf{y}_1) + h(\mathbf{x}, \mathbf{y}_1) - h(\mathbf{x}, \mathbf{y}_2) - (D_H(\mathbf{y}', \mathbf{y}'_1) + h(\mathbf{x}', \mathbf{y}'_1) - h(\mathbf{x}', \mathbf{y}'_2))| \\ &\leq |D_H(\mathbf{y}, \mathbf{y}_1) + h(\mathbf{x}, \mathbf{y}_1) - (D_H(\mathbf{y}', \mathbf{y}'_1) + h(\mathbf{x}', \mathbf{y}'_1))| + |h(\mathbf{x}, \mathbf{y}_2) - h(\mathbf{x}', \mathbf{y}'_2)|. \end{aligned} \quad (54)$$

using the triangle inequality.

Focusing on the second absolute difference, we can assume, without loss of generality, that $h(\mathbf{x}, \mathbf{y}_2) \geq h(\mathbf{x}', \mathbf{y}'_2)$, meaning

$$\begin{aligned} |h(\mathbf{x}, \mathbf{y}_2) - h(\mathbf{x}', \mathbf{y}'_2)| &= h(\mathbf{x}, \mathbf{y}_2) - h(\mathbf{x}', \mathbf{y}'_2) \\ &\leq h(\mathbf{x}, \mathbf{y}_2) - h(\mathbf{x}', \mathbf{y}_2) \\ &= \mathbf{w} \cdot (\mathbf{f}(\mathbf{x}, \mathbf{y}_2) - \mathbf{f}(\mathbf{x}', \mathbf{y}_2)) \\ &\leq \|\mathbf{w}\|_q \|\mathbf{f}(\mathbf{x}, \mathbf{y}_2) - \mathbf{f}(\mathbf{x}', \mathbf{y}_2)\|_p \\ &\leq \|\mathbf{w}\|_q (\Delta_G + 2)R D_H(\mathbf{x}, \mathbf{x}'). \end{aligned} \quad (55)$$

The first inequality uses the optimality of \mathbf{y}'_2 , implying $-h(\mathbf{x}', \mathbf{y}'_2) \leq -h(\mathbf{x}', \mathbf{y}_2)$; the second inequality uses Hölder's inequality; the third inequality uses Lemma 7 (Equation 51). Note that we obtain the same upper bound if we assume that $h(\mathbf{x}, \mathbf{y}_2) \leq h(\mathbf{x}', \mathbf{y}'_2)$, since we can reverse the terms inside the absolute value and proceed with \mathbf{y}'_2 instead of \mathbf{y}_2 .

We now return to the first absolute difference. To reduce clutter, it will help to use the loss-augmented potentials, $\tilde{\theta}(\mathbf{x}, \mathbf{y}; \mathbf{w})$, from Equation 31. Recall that $\delta(\mathbf{y})$ denotes the loss augmentation vector for \mathbf{y} . We then have that

$$|(D_{\mathbb{H}}(\mathbf{y}, \mathbf{y}_1) + h(\mathbf{x}, \mathbf{y}_1)) - (D_{\mathbb{H}}(\mathbf{y}', \mathbf{y}'_1) + h(\mathbf{x}', \mathbf{y}'_1))| = |\tilde{\theta}(\mathbf{x}, \mathbf{y}; \mathbf{w}) \cdot \hat{\mathbf{y}}_1 - \tilde{\theta}(\mathbf{x}', \mathbf{y}'; \mathbf{w}) \cdot \hat{\mathbf{y}}'_1|.$$

If we assume (without loss of generality) that $\tilde{\theta}(\mathbf{x}, \mathbf{y}; \mathbf{w}) \cdot \hat{\mathbf{y}}_1 \geq \tilde{\theta}(\mathbf{x}', \mathbf{y}'; \mathbf{w}) \cdot \hat{\mathbf{y}}'_1$, then

$$\begin{aligned} |\tilde{\theta}(\mathbf{x}, \mathbf{y}; \mathbf{w}) \cdot \hat{\mathbf{y}}_1 - \tilde{\theta}(\mathbf{x}', \mathbf{y}'; \mathbf{w}) \cdot \hat{\mathbf{y}}'_1| &= \tilde{\theta}(\mathbf{x}, \mathbf{y}; \mathbf{w}) \cdot \hat{\mathbf{y}}_1 - \tilde{\theta}(\mathbf{x}', \mathbf{y}'; \mathbf{w}) \cdot \hat{\mathbf{y}}'_1 \\ &\leq \tilde{\theta}(\mathbf{x}, \mathbf{y}; \mathbf{w}) \cdot \hat{\mathbf{y}}_1 - \tilde{\theta}(\mathbf{x}', \mathbf{y}'; \mathbf{w}) \cdot \hat{\mathbf{y}}_1 \\ &= (\tilde{\theta}(\mathbf{x}; \mathbf{w}) + \delta(\mathbf{y}) - \tilde{\theta}(\mathbf{x}'; \mathbf{w}) - \delta(\mathbf{y}')) \cdot \hat{\mathbf{y}}_1 \\ &= \mathbf{w} \cdot (\mathbf{f}(\mathbf{x}, \mathbf{y}_1) - \mathbf{f}(\mathbf{x}', \mathbf{y}'_1)) + (\delta(\mathbf{y}) - \delta(\mathbf{y}')) \cdot \hat{\mathbf{y}}_1 \\ &\leq \|\mathbf{w}\|_q (\Delta_G + 2)R D_{\mathbb{H}}(\mathbf{x}, \mathbf{x}') + (\delta(\mathbf{y}) - \delta(\mathbf{y}')) \cdot \hat{\mathbf{y}}_1 \\ &\leq \|\mathbf{w}\|_q (\Delta_G + 2)R D_{\mathbb{H}}(\mathbf{x}, \mathbf{x}') + D_{\mathbb{H}}(\mathbf{y}, \mathbf{y}'). \end{aligned} \quad (56)$$

The first inequality uses the optimality of \mathbf{y}'_1 ; the second inequality uses Hölder's inequality and Lemma 7 again; the last inequality uses the fact that

$$(\delta(\mathbf{y}) - \delta(\mathbf{y}')) \cdot \hat{\mathbf{y}}_1 = D_{\mathbb{H}}(\mathbf{y}, \mathbf{y}_1) - D_{\mathbb{H}}(\mathbf{y}', \mathbf{y}'_1) \leq D_{\mathbb{H}}(\mathbf{y}, \mathbf{y}').$$

The upper bound in Equation 56 also holds when $\tilde{\theta}(\mathbf{x}', \mathbf{y}'; \mathbf{w}) \cdot \hat{\mathbf{y}}'_1 \geq \tilde{\theta}(\mathbf{x}, \mathbf{y}; \mathbf{w}) \cdot \hat{\mathbf{y}}_1$.

Combining Equations 55 to 57, we then have that

$$\begin{aligned} n |L_{\mathcal{I}}(h, \mathbf{z}) - L_{\mathcal{I}}(h, \mathbf{z}')| &\leq 2(\Delta_G + 2)R \|\mathbf{w}\|_q D_{\mathbb{H}}(\mathbf{x}, \mathbf{x}') + D_{\mathbb{H}}(\mathbf{y}, \mathbf{y}') \\ &\leq 2(\Delta_G + 2)R \|\mathbf{w}\|_q D_{\mathbb{H}}(\mathbf{z}, \mathbf{z}') + D_{\mathbb{H}}(\mathbf{z}, \mathbf{z}'). \end{aligned}$$

Dividing both sides by n yields Equation 33. To obtain Equation 34, we use Lemma 7's Equation 52 in Equations 55 and 56, which reduces the term $(\Delta_G + 2)$ to just 2.

C.2 Proof of Lemma 3

The proof proceeds similarly to that of Lemma 2. Let

$$\begin{aligned} \mathbf{y}_1 &\triangleq \arg \max_{\mathbf{u} \in \mathcal{Y}^n} D_{\mathbb{H}}(\mathbf{y}, \mathbf{u}) + h(\mathbf{x}, \mathbf{u}); & \mathbf{y}_2 &\triangleq \arg \max_{\mathbf{u} \in \mathcal{Y}^n} h(\mathbf{x}, \mathbf{u}); \\ \mathbf{y}'_1 &\triangleq \arg \max_{\mathbf{u} \in \mathcal{Y}^n} D_{\mathbb{H}}(\mathbf{y}, \mathbf{u}) + h'(\mathbf{x}, \mathbf{u}); & \mathbf{y}'_2 &\triangleq \arg \max_{\mathbf{u} \in \mathcal{Y}^n} h'(\mathbf{x}, \mathbf{u}). \end{aligned}$$

Using this notation, we have that

$$\begin{aligned} n |L_{\mathcal{I}}(h, \mathbf{z}) - L_{\mathcal{I}}(h', \mathbf{z})| &\leq |(D_{\mathbb{H}}(\mathbf{y}, \mathbf{y}_1) + h(\mathbf{x}, \mathbf{y}_1)) - (D_{\mathbb{H}}(\mathbf{y}, \mathbf{y}'_1) + h'(\mathbf{x}, \mathbf{y}'_1))|, \quad (57) \end{aligned}$$

via the triangle inequality. Assuming $h(\mathbf{x}, \mathbf{y}_2) \geq h'(\mathbf{x}, \mathbf{y}'_2)$, we have that

$$\begin{aligned} |h(\mathbf{x}, \mathbf{y}_2) - h'(\mathbf{x}, \mathbf{y}'_2)| &= h(\mathbf{x}, \mathbf{y}_2) - h'(\mathbf{x}, \mathbf{y}'_2) \\ &\leq h(\mathbf{x}, \mathbf{y}_2) - h'(\mathbf{x}, \mathbf{y}_2) \\ &= (\mathbf{w} - \mathbf{w}') \cdot \mathbf{f}(\mathbf{x}, \mathbf{y}_2) \\ &\leq \|\mathbf{w} - \mathbf{w}'\|_q \|\mathbf{f}(\mathbf{x}, \mathbf{y}_2)\|_p \\ &\leq \|\mathbf{w} - \mathbf{w}'\|_q |G| R, \end{aligned} \quad (58)$$

via Lemma 8. Further, using the loss-augmented potentials, and assuming $\tilde{\theta}(\mathbf{x}, \mathbf{y}; \mathbf{w}) \cdot \hat{\mathbf{y}}_1 \geq \tilde{\theta}(\mathbf{x}, \mathbf{y}'; \mathbf{w}') \cdot \hat{\mathbf{y}}'_1$, we have that

$$\begin{aligned} |(D_{\mathbb{H}}(\mathbf{y}, \mathbf{y}_1) + h(\mathbf{x}, \mathbf{y}_1)) - (D_{\mathbb{H}}(\mathbf{y}, \mathbf{y}'_1) + h'(\mathbf{x}, \mathbf{y}'_1))| &= \tilde{\theta}(\mathbf{x}, \mathbf{y}; \mathbf{w}) \cdot \hat{\mathbf{y}}_1 - \tilde{\theta}(\mathbf{x}, \mathbf{y}'; \mathbf{w}') \cdot \hat{\mathbf{y}}'_1 \\ &\leq \tilde{\theta}(\mathbf{x}, \mathbf{y}; \mathbf{w}) \cdot \hat{\mathbf{y}}_1 - \tilde{\theta}(\mathbf{x}, \mathbf{y}'; \mathbf{w}') \cdot \hat{\mathbf{y}}_1 \\ &= (\tilde{\theta}(\mathbf{x}; \mathbf{w}) + \delta(\mathbf{y}) - \tilde{\theta}(\mathbf{x}'; \mathbf{w}') - \delta(\mathbf{y}')) \cdot \hat{\mathbf{y}}_1 \\ &= (\mathbf{w} - \mathbf{w}') \cdot \mathbf{f}(\mathbf{x}, \mathbf{y}_1) \\ &\leq \|\mathbf{w} - \mathbf{w}'\|_q \|\mathbf{f}(\mathbf{x}, \mathbf{y}_1)\|_p \\ &\leq \|\mathbf{w} - \mathbf{w}'\|_q |G| R. \end{aligned} \quad (59)$$

Combining the inequalities and dividing by n completes the proof.

C.3 Proof of Example 1

Since the weights are uniformly bounded, we define the prior, \mathbb{P} , as a uniform distribution on the d -dimensional unit ball. Given a (learned) hypothesis, h , with weights \mathbf{w} , we construct a posterior, \mathbb{Q}_h , as a uniform distribution on a d -dimensional ball with radius ϵ , centered at \mathbf{w} , and clipped at the boundary of the unit ball; i.e., its support is $\{\mathbf{w}' \in \mathbb{R}^d : \|\mathbf{w}' - \mathbf{w}\|_2 \leq \epsilon, \|\mathbf{w}'\|_2 \leq 1\}$. We let $\epsilon \triangleq (m|G|)^{-1}$, meaning the radius of the ball should decrease as the size of the training set increases.

For a uniform distribution, \mathbb{U} , with *support* $\text{supp}(\mathbb{U}) \subseteq \mathcal{H}$, we denote its *volume* by

$$\text{vol}(\mathbb{U}) \triangleq \int_{\mathcal{H}} \mathbb{1}\{h \in \text{supp}(\mathbb{U})\} dh.$$

The probability density function of \mathbb{U} is the inverse of its volume. The volume of \mathbb{P} is the volume of a unit ball, which is proportional to 1. Similarly, the volume of \mathbb{Q}_h is at least the volume of a d -dimensional ball with radius $\epsilon/2$ (due to the intersection with the unit ball), which is proportional to $(\epsilon/2)^d$.¹² Therefore, using p and q_h to denote their respective

¹² We withhold the precise definitions for simplicity of exposition. It will suffice to recognize their relative proportions, since the withheld constant depends only on d , and is thereby canceled out in the KL divergence.

densities, we have that

$$\begin{aligned} D_{\text{KL}}(\mathbb{Q}_h \|\mathbb{P}) &= \int_{\mathcal{H}} q_h(t') \ln \frac{q_h(t')}{p(t')} dt' \\ &= \int_{\mathcal{H}} q_h(t') \ln \frac{\text{vol}(\mathbb{P})}{\text{vol}(\mathbb{Q}_h)} dt' \\ &\leq \int_{\mathcal{H}} q_h(t') \ln(2/\epsilon)^d dt' \\ &= d \ln(2m |G|). \end{aligned}$$

By assumption, every allowable hypothesis has a weight vector \mathbf{w} with $\|\mathbf{w}\|_2 \leq 1$. We also assume that $\sup_{x \in \mathcal{X}} \|x\|_2 \leq 1$. Therefore, with $R = 1$ and $\beta \triangleq (2\Delta_G + 4) + 1$, Lemma 2 immediately proves that $L_r \circ \{h \in \mathcal{H}_{\text{KNS}} : \|\mathbf{w}\|_2 \leq 1\}$ is (β/n) -uniformly stable. Invoking Corollary 1, we then have that, with probability at least $1 - \delta$, every $\mathbb{Q}_h : \|\mathbf{w}\|_2 \leq 1$ satisfies

$$\bar{I}_r(\mathbb{Q}_h) \leq \hat{L}_r(\mathbb{Q}_h, \hat{\mathcal{Z}}) + 2((2\Delta_G + 4) + 1) \|\mathbf{r}^\top\|_\infty \sqrt{\frac{d \ln(2m |G|) + \ln \frac{2}{\delta}}{2mn}}. \quad (60)$$

By construction, every $h' \sim \mathbb{Q}_h$ satisfies $\|\mathbf{w}' - \mathbf{w}\|_2 \leq (m |G|)^{-1}$, so \mathbb{Q} has $(1/(m |G|), 0)$ -local hypothesis stability. As demonstrated in Equation 37, L_r has $(2 |G|/n, \emptyset)$ -local hypothesis stability. Thus, via Proposition 5, (L_r, \mathbb{Q}) has $(2/(mn), \emptyset, 0)$ -local stability. Then, via Proposition 4 and Equation 32, we have that

$$\bar{I}_n(h) \leq \bar{I}_r(h) \leq \bar{I}_r(\mathbb{Q}_h) + \frac{2}{mn}, \quad (61)$$

and

$$\hat{L}_r(\mathbb{Q}_h, \hat{\mathcal{Z}}) \leq \hat{L}_r(h, \hat{\mathcal{Z}}) + \frac{2}{mn} \leq \hat{L}_n(h, \hat{\mathcal{Z}}) + \frac{2}{mn}. \quad (62)$$

Combining Equations 60 to 62 completes the proof.

C.4 Proof of Lemma 4

We begin with a fundamental property of the normal distribution, which is used to prove the concentration inequality.

Fact 1. *If X is a Gaussian random variable, with mean μ and variance σ^2 , then, for any $\epsilon > 0$,*

$$\Pr\{|X - \mu| \geq \epsilon\} \leq 2 \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right). \quad (63)$$

Observe that, if $\|\mathbf{X} - \boldsymbol{\mu}\|_p \geq \epsilon$, then there must exist at least one coordinate $i \in [d]$ such that $|X_i - \mu_i| \geq \epsilon/d^{1/p}$; otherwise, we would have

$$\|\mathbf{X} - \boldsymbol{\mu}\|_p = \left(\sum_{i=1}^d |X_i - \mu_i|^p\right)^{1/p} < \left(d \left(\frac{\epsilon}{d^{1/p}}\right)^p\right)^{1/p} = \epsilon.$$

We therefore have that

$$\begin{aligned} \Pr\left\{\|\mathbf{X} - \boldsymbol{\mu}\|_p \geq \epsilon\right\} &\leq \Pr\left\{\exists i : |X_i - \mu_i| \geq \frac{\epsilon}{d^{1/p}}\right\} \\ &\leq \sum_{i=1}^d \Pr\left\{|X_i - \mu_i| \geq \frac{\epsilon}{d^{1/p}}\right\} \\ &\leq \sum_{i=1}^d 2 \exp\left(-\frac{\epsilon^2}{2\sigma^2 d^{2/p}}\right). \end{aligned}$$

The second inequality uses the union bound; the last uses Fact 1. Summing over $i = 1, \dots, d$ completes the proof.

C.5 Proof of Example 4

We first show that $\mathbb{D}(\mathcal{B}_{\mathcal{Z}}) \leq 1/n$. Then, the rest of the proof is a simple modification of the previous analyses.

Observe that, for any x and μ_y ,

$$\|x\|_2 - 1 \leq \|x\|_2 - \|\mu_y\|_2 \leq \|x - \mu_y\|_2.$$

So, if $\|x\|_2 \geq 2$, then $\|x - \mu_y\|_2 \geq 1$. Therefore, using the union bound, and Lemma 4, we can upper-bound the measure of $\mathcal{B}_{\mathcal{Z}}$ as follows:

$$\begin{aligned} \mathbb{D}(\mathcal{B}_{\mathcal{Z}}) &= \Pr_{\mathcal{Z} \sim \mathbb{D}}\{\exists i : \|X_i\|_2 \geq 2\} \\ &\leq \sup_{\mathbf{y} \in \mathcal{Y}^n} \Pr_{\mathbf{X} \sim \mathbb{D}}\{\exists i : \|X_i\|_2 \geq 2 \mid \mathbf{Y} = \mathbf{y}\} \\ &= \sup_{\mathbf{y} \in \mathcal{Y}^n} \sum_{i=1}^n \Pr_{\mathbf{X} \sim \mathbb{D}}\{\|X_i\|_2 \geq 2 \mid Y_i = y_i\} \\ &\leq \sup_{\mathbf{y} \in \mathcal{Y}^n} \sum_{i=1}^n \Pr_{X_i \sim \mathbb{D}}\{\|X_i - \mu_{y_i}\|_2 \geq 1 \mid Y_i = y_i\} \\ &\leq \sup_{\mathbf{y} \in \mathcal{Y}^n} \sum_{i=1}^n 2k \exp\left(-\frac{1}{2k\sigma_{g_i}^2}\right) \\ &\leq \sum_{i=1}^n 2k \exp\left(-\frac{2k \ln(2kn^2)}{2k}\right) = \frac{1}{n}. \end{aligned}$$

Conditioned on $\mathbf{Z} \notin \mathcal{B}_{\mathcal{Z}}$, we have that Lemmas 7 and 8 hold for $R = 2$; hence, so do Lemmas 2 and 3. With \mathbb{P}, \mathbb{Q}_h and $\mathcal{B}_{\mathcal{H}_{\text{KNS}}}(h)$ constructed identically to Example 2, this means that \mathbb{Q}_h is $(\beta_h/n, \mathcal{B}_{\mathcal{Z}}, 1/(mn))$ -locally stable. Further, L_r has $(4 |G|/n, \mathcal{B}_{\mathcal{Z}})$ -local hypothesis stability; and \mathbb{Q} has $(1/(m |G|), 1/(mn))$ -local hypothesis stability; by Proposition 5, this means that (L_r, \mathbb{Q}) has $(4/(mn), \mathcal{B}_{\mathcal{Z}}, 1/(mn))$ -local stability. Thus, invoking Theorem 2 and Proposition 4, with $\nu = 1/n$, we have that, with probability at least $1 - \delta - m/n$, all

$l \in [m]$ satisfy $\mathbf{Z}^{(l)} \notin \mathcal{B}_z$, and all $h \in \mathcal{H}_{\text{MN}}$ satisfy

$$\begin{aligned} \bar{L}_r(h) &\leq \hat{L}_r(\mathbb{Q}_h) + \frac{5}{mn} + \frac{1}{n} \\ &\leq \hat{L}_r(\mathbb{Q}_h, \hat{\mathbf{Z}}) + \frac{6}{mn} + \frac{2}{n} \\ &\quad + 4\beta_h \|\Gamma_{\mathbb{P}}\|_{\infty} \sqrt{\frac{\frac{1}{2} \|\mathbf{w}\|_2^2 + \frac{d}{2} \ln(2d(m|G|)^2 \ln(2dmm)) + \ln \frac{4\beta_h}{\delta}}{2mn}}. \end{aligned}$$

Further, since none of the training examples in the sample are “bad,” we also have that

$$\hat{L}_r(\mathbb{Q}_h, \hat{\mathbf{Z}}) \leq \hat{L}_r(h, \hat{\mathbf{Z}}) + \frac{5}{mn} \leq \hat{L}_h(h, \hat{\mathbf{Z}}) + \frac{5}{mn}.$$

Combining these inequalities completes the proof.

References

- P. Alquier and O. Wintenburger. Model selection for weakly dependent time series forecasting. *Bernoulli*, 18(3):883–913, 2012.
- A. Ambroladze, E. Parrado-Hernández, and J. Shawe-Taylor. Tighter PAC-Bayes bounds. In *Neural Information Processing Systems*, 2006.
- P. Bartlett, M. Collins, D. McAllester, and B. Taskar. Large margin methods for structured classification: Exponentiated gradient algorithms and PAC-Bayesian generalization bounds. Extended version of paper appearing in *Advances in Neural Information Processing Systems* 17, 2005.
- O. Bousquet and A. Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.
- J. Bradley and C. Guestrin. Sample complexity of composite likelihood. In *Artificial Intelligence and Statistics*, 2012.
- R. Bradley. Basic properties of strong mixing conditions: A survey and some open questions. *Probability Surveys*, 2(2):107–144, 2005.
- O. Catoni. *Pac-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning*, volume 56 of *Institute of Mathematical Statistics Lecture Notes – Monograph Series*. Institute of Mathematical Statistics, 2007.
- H. Chan and A. Darwiche. Sensitivity analysis in Markov networks. In *International Joint Conference on Artificial Intelligence*, 2005.
- H. Chan and A. Darwiche. On the robustness of most probable explanations. In *Uncertainty in Artificial Intelligence*, 2006.
- J. Chazottes, P. Collet, C. Külske, and F. Redig. Concentration inequalities for random fields via coupling. *Probability Theory and Related Fields*, 137:201–225, 2007.
- M. Collins. Parameter estimation for statistical parsing models: Theory and practice of distribution-free methods. In *International Conference on Parsing Technologies*, 2001.
- M. Donsker and S. Varadhan. Asymptotic evaluation of certain Markov process expectations for large time. *Communications on Pure and Applied Mathematics*, 28(1):1–47, 1975.
- D. Fiebig. Mixing properties of a class of Bernoulli processes. *Transactions of the American Mathematical Society*, 338:479–492, 1993.
- P. Germain, A. Lacasse, F. Laviolette, and M. Marchand. PAC-Bayesian learning of linear classifiers. In *International Conference on Machine Learning*, 2009.
- S. Giguère, F. Laviolette, M. Marchand, and K. Sylla. Risk bounds and learning algorithms for the regression approach to structured output prediction. In *International Conference on Machine Learning*, 2013.
- K. Gimpel and N. Smith. Softmax-margin CRFs: Training log-linear models with cost functions. In *Conference of the North American Chapter of the Association of Computational Linguistics*, 2010.
- T. Hazan and R. Urtasun. A primal-dual message-passing algorithm for approximated large scale structured prediction. In *Neural Information Processing Systems*, 2010.
- T. Hazan, S. Maji, J. Keshet, and T. Jaakkola. Learning efficient random maximum a posteriori predictors with non-decomposable loss functions. In *Neural Information Processing Systems*, 2013.
- R. Herbrich and T. Graepel. A PAC-Bayesian margin bound for linear classifiers: Why SVMs work. In *Neural Information Processing Systems*, 2001.
- W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.
- J. Honorio. Lipschitz parametrization of probabilistic graphical models. In *Uncertainty in Artificial Intelligence*, 2011.
- D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.
- A. Kontorovich. Obtaining measure concentration from Markov contraction. *Markov Processes and Related Fields*, 18:613–638, 2012.
- A. Kontorovich and K. Ramanan. Concentration inequalities for dependent random variables via the martingale method. *Annals of Probability*, 36(6):2126–2158, 2008.
- S. Kutin. Extensions to McDiarmid’s inequality when differences are bounded with high probability. Technical report, University of Chicago, 2002.
- J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Intl. Conference on Machine Learning*, 2001.

- J. Langford and J. Shawe-Taylor. PAC-Bayes and margins. In *Neural Information Processing Systems*, 2002.
- G. Lever, F. Laviolette, and J. Shawe-Taylor. Distribution-dependent PAC-Bayes priors. In *Conference on Algorithmic Learning Theory*, 2010.
- B. London, B. Huang, B. Taskar, and L. Gettoor. Collective stability in structured prediction: Generalization from one example. In *International Conference on Machine Learning*, 2013.
- B. London, B. Huang, B. Taskar, and L. Gettoor. PAC-Bayesian collective stability. In *Artificial Intelligence and Statistics*, 2014.
- D. McAllester. Some PAC-Bayesian theorems. In *Conference on Computational Learning Theory*, 1998.
- D. McAllester. PAC-Bayesian model averaging. In *Conference on Computational Learning Theory*, 1999.
- D. McAllester. Simplified PAC-Bayesian margin bounds. In *Conference on Computational Learning Theory*, 2003.
- D. McAllester. Generalization bounds and consistency for structured labeling. In G. Bakir, T. Hofmann, B. Schölkopf, A. Smola, B. Taskar, and S. Vishwanathan, editors, *Predicting Structured Data*. MIT Press, 2007.
- D. McAllester and J. Keshet. Generalization bounds and consistency for latent structural probit and ramp loss. In *Neural Information Processing Systems*, 2011.
- C. McDiarmid. On the method of bounded differences. In *Surveys in Combinatorics, volume 141 of London Mathematical Society Lecture Note Series*, pages 148–188. Cambridge University Press, 1989.
- C. McDiarmid. Concentration. *Probabilistic Methods for Algorithmic Discrete Mathematics*, pages 195–248, 1998.
- D. McDonald, C. Shalizi, and M. Schervish. Estimating β -mixing coefficients. In *Artificial Intelligence and Statistics*, 2011.
- D. McDonald, C. Shalizi, and M. Schervish. Time series forecasting: model evaluation and selection using nonparametric risk bounds. *CoRR*, abs/1212.0463, 2012.
- M. Mohri and A. Rostamizadeh. Rademacher complexity bounds for non-i.i.d. processes. In *Neural Information Processing Systems*, 2009.
- M. Mohri and A. Rostamizadeh. Stability bounds for stationary ϕ -mixing and β -mixing processes. *Journal of Machine Learning Research*, 11:789–814, 2010.
- J. Neville and D. Jensen. Dependency networks for relational data. In *International Conference on Data Mining*, 2004.
- L. Ralaivola, M. Szafrański, and G. Stempfel. Chromatic PAC-Bayes bounds for non-i.i.d. data: Applications to ranking and stationary β -mixing processes. *Journal of Machine Learning Research*, 11:1927–1956, 2010.
- M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.
- Dan Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1-2): 273–302, 1996.
- P. Samson. Concentration of measure inequalities for Markov chains and ϕ -mixing processes. *Annals of Probability*, 28(1):416–461, 2000.
- M. Seeger. PAC-Bayesian generalization error bounds for Gaussian process classification. *Journal of Machine Learning Research*, 3:233–269, 2002.
- Y. Seldin, F. Laviolette, N. Cesa-Bianchi, J. Shawe-Taylor, and P. Auer. PAC-Bayesian inequalities for martingales. *IEEE Transactions on Information Theory*, 58(12):7086–7093, 2012.
- B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Uncertainty in Artificial Intelligence*, 2002.
- B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In *Neural Information Processing Systems*, 2004.
- I. Tschantzaris, T. Joachims, T. Hofmann, and Y. Altmann. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.
- N. Usunier, M. Amni, and P. Gallinari. Generalization error bounds for classifiers trained with interdependent data. In *Neural Information Processing Systems*, 2006.
- V. Vu. Concentration of non-Lipschitz functions and applications. *Random Structures and Algorithms*, 20(3):262–316, 2002.
- M. Wainwright. Estimating the “wrong” graphical model: Benefits in the computation-limited setting. *Journal of Machine Learning Research*, 7:1829–1859, 2006.
- M. Wainwright and M. Jordan. *Graphical Models, Exponential Families, and Variational Inference*. Now Publishers Inc., 2008.
- R. Xiang and J. Neville. Relational learning with one network: An asymptotic analysis. In *Artificial Intelligence and Statistics*, 2011.
- B. Yu. Rates of convergence for empirical processes of stationary mixing sequences. *Annals of Probability*, 22(1):94–116, 1994.

Composite Multiclass Losses

Robert C. Williamson

Australian National University and Data61

BOB.WILLIAMSON@ANU.EDU.AU

Elodie Vernet

*Centre for Mathematical Sciences
University of Cambridge*

EV315@CAM.AC.UK

Mark D. Reid

Australian National University and Data61

MARK.REID@ANU.EDU.AU

Editor: Nicolas Vayatis

Abstract

We consider loss functions for multiclass prediction problems. We show when a multiclass loss can be expressed as a “proper composite loss”, which is the composition of a proper loss and a link function. We extend existing results for binary losses to multiclass losses. We subsume results on “classification calibration” by relating it to properness. We determine the stationarity condition, Bregman representation, order-sensitivity, and quasi-convexity of multiclass proper losses. We then characterise the existence and uniqueness of the composite representation for multiclass losses. We show how the composite representation is related to other core properties of a loss: mixability, admissibility and (strong) convexity of multiclass losses which we characterise in terms of the Hessian of the Bayes risk. We show that the simple integral representation for binary proper losses can not be extended to multiclass losses but offer concrete guidance regarding how to design different loss functions. The conclusion drawn from these results is that the proper composite representation is a natural and convenient tool for the design of multiclass loss functions.

Keywords: Proper losses, Multiclass losses, Link Functions, Convexity and quasi-convexity of losses, Margin losses, Classification calibration, Parametrisations and representations of loss functions, Admissibility, Mixability, Minimality, Superprediction set

1. Introduction

Machine learning is done for a purpose. The performance of a machine learning solution is judged by means of a loss function. Different choices of loss function will lead to different solutions. The theory of binary losses (i.e. losses suitable for binary prediction problems) is well understood. This paper extends that understanding to multiclass losses and aids the choice of a suitable loss function by exploring the parametrisations available and the implications of different choices. It does so by systematically exploring a decomposition of a multiclass loss into two components, one which affects the statistical performance, and one which affects the computational optimisation of models.

The problem setting is where one is given a bag $\{(x_i, y_i)\}_i$ of pairs of points x_i and their accompanying labels $y_i \in [n] := \{1, \dots, n\}$, drawn from a finite set of size n . The task can

be either predict a label for an unseen instance, or predict the probability that a label takes on a particular value. These two problems are called multiclass *classification* and *probability estimation* respectively.

Proper composite losses are the composition of a *proper loss* and an *invertible link* (both defined formally below). This representation makes the understanding of multiclass losses easier because, crucially, it separates two distinct concerns: the statistical and the computational. The statistical properties are controlled by the proper loss, while the link function is essentially just a parametrisation. Choice of a suitable link can help—for example, a nonconvex proper loss can be made convex (and thus more amenable to numerical optimisation) by choice of the appropriate link. For prediction purposes it is desirable to use an *admissible* loss (one where every possible prediction is uniquely optimal for some underlying distribution). It turns out that every proper composite loss is admissible; in fact proper composite losses satisfy a stronger adequacy property than admissibility.

We characterise when a multiclass loss has a proper composite representation and when such representations are unique. We consider integral representations (whereby the proper component can be expressed as a weighted combination of elementary proper losses). We show the surprising result that there is a fundamental difference between $n = 2$ and $n > 2$ in terms of the simplicity of the parametrisation of the class of elementary proper losses. It has been known for some time that proper losses are characterised by their conditional Bayes risks (or entropy functions). It has already been shown how important properties of a loss that control the performance of certain learning tasks can be expressed directly in terms of the Bayes risk. In this paper we extend results due to Reid and Williamson (2010) (for $n = 2$) to general n and characterise the convexity of a proper loss in terms of the associated Bayes risk.

We also illuminate the connection between classification and probability estimation by characterising the relationship between the crucial property that a loss should have for each of these: *classification calibrated* (which we first generalise to make sense in the more general setting we consider) and *properness*. We explain the relationship between these two concepts, which captures the idea behind the probing reduction from classification to class probability estimation.

We also show how the results of the paper can provide tools to help with the design of multiclass losses, putting this on firmer ground than in the past.

1.1 Previous Work

With some exceptions, existing work on multiclass loss functions attempts to work directly with $\ell: \mathcal{Y} \rightarrow \mathbb{R}^n$. As we shall show this conflates two separate concerns—the design of the *statistical* properties of the loss, those that affect statistical performance, with the aspects that affect the computational properties that control the ease with which empirical averages of the loss are minimized. The proper composite representation is not new—in hindsight the observation of Grünwald and Dawid (2004) that every loss induces a proper scoring rule is tantamount to the proper composite representation. Furthermore, its components (link functions and proper losses) have a long history. The novelty of the present work is to systematically use these two components as a canonical parametrisation of loss functions. Key differences between the present paper and previous work are tabulated in Table 1.

Attribute	Previous Work	Present Paper	Ref.
Structure and Semantics	None—just a function, possibly convex in parameters	Clear separation of concerns and meaning for λ and W . Gives meaning to predictions v as transformed probabilities.	Fig. 1
Classification versus probability estimation	Little insight in the multiclass case; confer recent works such as (Reid and Williamson, 2010, 2011; Narasimhan and Agarwal, 2013; Menon and Williamson, 2014) for the binary case	Clear connection via a characterisation relating classification calibrated, prediction calibrated and proper losses	§3
Effect of choice of loss function on performance	Margin based. Only a sufficient condition and only for statistical batch setting. Mixes up statistical fundamentals (\mathcal{L}) with parametrization (Ψ). Strong convexity for speed of convergence in online setting; cf. (Abemethy et al., 2009).	Mixability and Stochastic Mixability. Characterisation in online setting. Both online worst-case and statistical batch settings. Parametrisation Ψ automatically ignored.	§6.1
Admissibility	Not considered explicitly. Ensured however by assuming ℓ is convex.	All proper composite losses admissible. All continuous Bayes losses have a proper composite representation.	§6.2
Quasi-convexity and Minimality	Guaranteed by assuming ℓ is convex.	Quasi-convexity guaranteed for all continuous proper losses; minimality for all continuous proper composite losses.	§6.4
Convexifiability	No principled way to convexify a loss; can make convex surrogate approximations.	All continuous proper losses convexifiable (using the canonical link).	§6.4
Design principles and parameterisation	No guidance; choose ℓ or margin function ϕ , in which case symmetry imposed.	Principled: general asymmetric losses possible; parametrise via (Δ, Ψ) ; separation of concerns.	§8.3
Connections to divergences	Many to one for margin losses in binary case. (Nguyen et al., 2009)	Explicit 1:1 correspondence for binary and multiclass case (Reid and Williamson, 2011; García and Williamson, 2012).	§9

Table 1: Comparison of present paper to previous works on loss functions.

Proper losses are the natural losses to use for probability estimation. They have been studied in detail when $n = 2$ (the “binary case”) where there is a nice integral representation (Bijia et al., 2005; Gneiting and Raftery, 2007; Reid and Williamson, 2011), and characterization (Reid and Williamson, 2010) when differentiable. The proper composite representation for binary losses has proved very illuminating in the study of bipartite ranking problems (Menon and Williamson,

2014). Classification calibrated losses are an analog of proper losses for the problem of classification (Bartlett et al., 2006). The relationship between classification calibration and properness was determined by Reid and Williamson (2010) for $n = 2$. Most of these results have had no multiclass analogue until now. Whilst there is much work on classification problems, it is now widely understood that there are often advantages in being able to predict probabilities, rather than just labels (Bennett, 2003; Cohen and Goldszmidt, 2004).

The theory of loss functions makes it clear how one ideally chooses a loss—one takes account of one’s utility concerning various incorrect predictions (Kiefer, 1987), (Berger, 1985, Section 2.4). The practice rarely involves such a step, primarily, we conjecture, because there is no adequate understanding of the way one can parametrise losses effectively, especially in the multiclass case. There is little guidance in the literature concerning how to choose a loss function; typically heuristic arguments are used for the choice—confer e.g. (Ighodaro et al., 1982; Nayak and Naik, 1989). An early approach to multiclass losses is simply reduction to binary (Allwein et al., 2001; Beygelzimer et al., 2007; Crammer and Singer, 2001; Dietterich and Bakiri, 1995; Zadrozny and Elkan, 2002). Related approaches are pairwise coupling or Bradley-Terry models (Hastie and Tibshirani, 1998; Wu and Weng, 2004; Huang et al., 2006) where certain relationships are assumed to hold between the pairwise probabilities and the multivariate probability of interest.

The design of losses for multiclass prediction has received recent attention (Zhang, 2004; Hill and Doucet, 2007; Tewari and Bartlett, 2007; Liu, 2007; Santos-Rodríguez et al., 2009; Zou et al., 2008; Zhang et al., 2009) although none of these papers developed the connection to proper losses, and most restrict consideration to margin losses (which imply certain symmetry conditions). Zou et al (2005) proposed a multiclass generalisation of “admissible losses” (their name for classification calibration) for multiclass margin classification. Liu (2007) considered several multiclass generalisations of hinge loss (suitable for multiclass SVMs) and showed some of them were and others were not Fisher consistent, and when they were not it was shown how the training algorithm could be modified to make the losses behave consistently. Shi et al. (2010) have investigated the relationship between classification calibration of multiclass losses and losses for structure prediction, and have proposed an extension of classification calibration which they call parametric consistency, which attempts to take account of the function class used (classification calibration is, like all the results in this paper, concerned with behaviour *per point*, in practice one typically optimises over restricted classes of functions). Multiclass losses have also been considered in the development of multiclass boosting (e.g. Zhu et al., 2009; Mukherjee and Schapire, 2013; Wu and Lange, 2010).

1.2 Outline

The rest of the paper is organised as follows. In §2 we set up the problem formally and state some purely mathematical results we will need; §3 we relate properness, classification calibration, and the notion used by Tewari and Bartlett (2007) which we rename “prediction calibrated”; §4 we provide a novel characterization of multiclass properness; §5 we study composite proper losses (the composition of a proper loss with an invertible link) and characterise when a given loss has such a representation and when the representation is unique; §6 we develop a number of interesting implications of the representation and the characterisation results in terms of

mixability (§6.1), admissibility (§6.2) and convexity (§6.4), where we give a complete characterisation of the (strong) convexity of composite multiclass proper losses in terms of the Bayes risk; §7: we present a (somewhat surprising) negative result concerning the integral representation of proper multiclass losses; §8: we outline how the above results can aid in the design of proper losses, especially by use of a (new) multiclass extension of the ‘‘canonical link’’; finally, §9 summarises the key contributions and outlines some future directions.

2. Formal Setup

Suppose \mathcal{X} is some set and $\mathcal{Y} = [n] = \{1, \dots, n\}$ is a set of labels. (Throughout the paper n is an integer greater than or equal to 2.) We suppose we are given data $S = \{(x_i, y_i)\}_{y_i \in [n]}$ such that $y_i \in \mathcal{Y}$ is the label corresponding to $x_i \in \mathcal{X}$. These data follow a joint distribution $\mathbb{P}_{\mathcal{X}, \mathcal{Y}}$ on $\mathcal{X} \times [n]$. We denote by $\mathbb{E}_{\mathcal{X}, \mathcal{Y}}$ and $\mathbb{E}_{\mathcal{Y}, \mathcal{X}}$ respectively, the expectation and the conditional expectation with respect to $\mathbb{P}_{\mathcal{X}, \mathcal{Y}}$. Given a new observation x we want to predict the probability $p_i := \mathbb{P}(Y = i | X = x)$ of x belonging to class i , for $i \in [n]$. *Multiclass classification* requires the learner to predict the most likely class of x ; that is to find $\hat{y} \in \arg \max_{i \in [n]} p_i$.

A loss measures the quality of prediction. Let $\Delta^n := \{(p_1, \dots, p_n) : \sum_{i \in [n]} p_i = 1, \text{ and } 0 \leq p_i \leq 1, \forall i \in [n]\}$ denote the *n-simplex*. For multiclass probability estimation, $\ell: \Delta^n \rightarrow \mathbb{R}_+^n$. The *partial losses* ℓ_i are the components of $\ell(q) = (\ell_1(q), \dots, \ell_n(q))'$ and $\ell_i(q)$ is the loss incurred by predicting $q \in \Delta^n$ when $y = i$. A commonly used loss for probability estimation is the *log loss* ℓ^{\log} defined by $\ell_i^{\log}(q) := -\log q_i$ for $i \in [n]$. Other examples of multiclass losses we will refer to in this paper include the *square loss* $\ell^{\text{sq}}(q) := \sum_{j \in [n]} (\|j - q_j\|)^2$, the *absolute loss* $\ell_i^{\text{abs}}(q) := \sum_{j \in [n]} \|[i - j] - q_j\|$ and the *0-1 loss* $\ell_i^{01}(q) := \mathbb{1}_{\{j \in \arg \max_{j \in [n]} q_j\}}$. Here, $\mathbb{1}\{P\}$ denotes the function that is 1 when P is true and 0 otherwise.

Throughout the paper, A' denotes transpose of the matrix or vector A , except when applied to a real-valued function where it denotes derivative. We denote matrix multiplication of compatible matrices A and B by $A \cdot B$, so the inner product of two vectors $x, y \in \mathbb{R}^n$ is $x' \cdot y$. The *conditional risk* L associated with a loss ℓ is the function

$$L: \Delta^n \times \Delta^n \ni (p, q) \mapsto L(p, q) = \mathbb{E}_{Y \sim p} \ell_Y(q) = p' \cdot \ell(q) \in \mathbb{R}_{++},$$

where $Y \sim p$ means Y is drawn according to a multinomial distribution with parameter $p \in \Delta^n$. In a typical learning problem one will construct an estimate $q: \mathcal{X} \rightarrow \Delta^n$. The *full risk* is $\mathbb{L}(q) = \mathbb{E}_{\mathcal{X}, \mathcal{Y}} \mathbb{E}_{\mathcal{Y}, \mathcal{X}} \ell_Y(q(X))$. Minimizing $\mathbb{L}(q)$ over $q: \mathcal{X} \rightarrow \Delta^n$ is equivalent to minimizing $L(p(x), q(x))$ over $q(x) \in \Delta^n$ for all $x \in \mathcal{X}$ where $p(x) = (p_1(x), \dots, p_n(x))'$, and $p_i(x) = \mathbb{P}(Y = i | X = x)$. Thus it suffices to only consider the conditional risk; confer (Reid and Williamson, 2011).

If one is interested in estimating probabilities ($\ell: \Delta^n \rightarrow \mathbb{R}_+^n$) it is natural to require the associated conditional risk is minimised when estimating the true underlying probability. Such a loss is called *proper* (formally: if $L(p, p) \leq L(p, q)$, $\forall p, q \in \Delta^n$). It is *strictly proper* if the inequality is strict when $p \neq q$ (so it is uniquely minimised by predicting the correct probability). The *conditional Bayes risk* is defined by

$$\underline{L}: \Delta^n \ni p \mapsto \inf_{q \in \Delta^n} L(p, q).$$

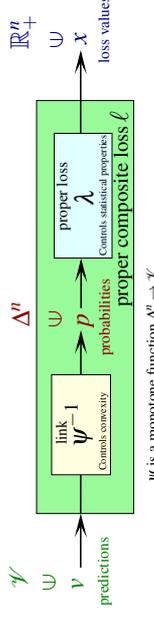


Figure 1: The idea of a proper composite loss.

This function is always concave (Gneiting and Raftery, 2007). If ℓ is proper, then $L(p) = L(p, p) = p' \cdot \ell(p)$. Strictly proper losses induce *Fisher consistent* estimators of probabilities: if ℓ is strictly proper, $p = \arg \min_q L(p, q)$. By considering when the derivatives $\frac{\partial}{\partial q_i} L(p, q)$ are zero it is straight-forward to show that, of the example losses introduced above, the log loss, square loss, and 0-1 loss are proper, while absolute loss is not. Furthermore, both log loss and square loss are strictly proper while 0-1 loss is proper but not strictly proper. Using the fact that, for proper losses, the Bayes risk $\underline{L}(p) = L(p, p)$ we see that $\underline{L}^{\log}(p) = -\sum_{i \in [n]} p_i \log p_i$ (i.e., Shannon entropy); $\underline{L}^{\text{sq}}(p) = 1 - \sum_{i \in [n]} p_i^2$; and $\underline{L}^{01}(p) = \min_i \{1 - p_i\}$.

The losses above are defined on the simplex Δ^n since the argument (a predictor) represents a probability vector. However it is sometimes desirable to use another set \mathcal{Y} of predictors. For example if one wishes to use linear predictors, their natural range is \mathbb{R}^n . One can consider losses $\ell: \mathcal{Y} \rightarrow \mathbb{R}_+^n$. Suppose there exists an invertible function $\psi: \Delta^n \rightarrow \mathcal{Y}$. Then ℓ can be written as a composition of a loss λ defined on the simplex with ψ^{-1} . That is, $\ell(v) = \lambda \psi(v) := \lambda(\psi^{-1}(v))$. Such a function $\lambda \psi$ is a *composite loss*. If λ is proper, we say ℓ is a *proper composite loss*, with *associated proper loss* λ and *link* ψ ; see Figure 1. Many commonly used multiclass losses are composite losses, even though they are not often expressed as such; see the example in §8.4.

Throughout the paper, ℓ is a general loss defined on \mathcal{Y} , where \mathcal{Y} may equal Δ^n , and λ is always a loss defined on Δ^n , which may be proper. For such a loss $\lambda: \Delta^n \rightarrow \mathbb{R}_+^n$, its corresponding conditional risk is denoted $\Lambda(p, q)$ and its conditional Bayes risk is $\underline{\Lambda}(p)$.

In order to differentiate the losses we project the n -simplex into a subset of \mathbb{R}^{n-1} . Let

$$\tilde{\Delta}^n := \left\{ (p_1, \dots, p_{n-1})' : p_i \geq 0, \forall i \in [n], \sum_{i=1}^{n-1} p_i \leq 1 \right\}$$

denote the ‘‘bottom’’ of the n -simplex. We denote by

$$\Pi_{\Delta}: \Delta^n \ni p = (p_1, \dots, p_n)' \mapsto \tilde{p} = (p_1, \dots, p_{n-1})' \in \tilde{\Delta}^n,$$

the projection of the Δ^n , and

$$\Pi_{\tilde{\Delta}}^{-1}: \tilde{\Delta}^n \ni \tilde{p} = (\tilde{p}_1, \dots, \tilde{p}_{n-1})' \mapsto p = (p_1, \dots, p_{n-1}, 1 - \sum_{i=1}^{n-1} \tilde{p}_i)' \in \Delta^n$$

its inverse. For convenience, we will often use $\tilde{n} := n - 1$ to denote the dimension of the set $\tilde{\Delta}^n$.

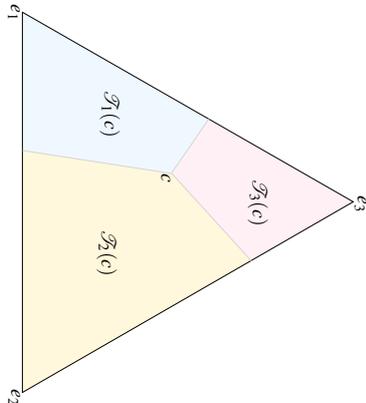


Figure 2: A partitioning of the 3-simplex by regions $\mathcal{F}_i(c)$, $i = 1, 2, 3$, where $c = (.35, .2, .45)$ as viewed from the direction $(1, 1, 1)$.

We use the following notation. The k th unit vector e_k is the n vector with all components zero except the k th which is 1. The n -vector $\mathbb{1}_n := (1, \dots, 1)^\top$. The (relative) interior of the simplex is $\hat{\Delta}^n := \{p_1, \dots, p_n\}$; $\Sigma_{i \in [n]} p_i = 1$, and $0 < p_i < 1$, $\forall i \in [n]$ and the boundary is $\partial \hat{\Delta}^n := \Delta^n \setminus \hat{\Delta}^n$. We also adopt notation from [Magnus and Neudecker \(1999\)](#). For the reader's convenience we list the essential notations and conventions in [Appendix A](#).

3. Relating Properness to Classification Calibration

Properness is an attractive property of a loss for the task of class probability estimation. However if one is merely interested in *classifying* (predicting $\hat{y} \in [n]$ given $x \in \mathcal{X}$) then it is stronger than one needs. In this section we relate *classification calibration* (the analog of properness for classification problems) to properness.

Suppose $c \in \hat{\Delta}^n$. We cover Δ^n with n subsets each representing one class:

$$\mathcal{F}_i(c) := \{p \in \Delta^n : \forall j \neq i \ p_i c_j \geq p_j c_i\}, \quad i \in [n].$$

Observe that for $i \neq j$, the sets $\mathcal{F}_i(c) := \{p \in \Delta^n : p_i c_j = p_j c_i\}$ are subsets of dimension $n-2$ through c and all e_k such that $k \neq i$ and $k \neq j$. These subsets partition \mathbb{R}^n into two parts. The set $\mathcal{F}_{ij}(c)$ is the intersection of Δ^n and the subspaces delimited by the precedent $(n-2)$ -subspace and in the same side as e_i . An example of this partition is shown graphically in [Figure 2](#). We will make use of the following properties of $\mathcal{F}_i(c)$.

Lemma 1 Suppose $c \in \hat{\Delta}^n$, $i \in [n]$. Then the following hold:

1. For all $p \in \Delta^n$, there exists i such that $p \in \mathcal{F}_i(c)$.
2. Suppose $p \in \Delta^n$. $\mathcal{F}_i(c) \cap \mathcal{F}_j(c) \subseteq \{p \in \Delta^n : p_i c_j = p_j c_i\}$, a subset of a subspace of dimension $n-2$.

3. Suppose $p \in \Delta^n$. If $p \in \cap_{i=1}^n \mathcal{F}_i(c)$ then $p = c$.
4. For all $p, q \in \Delta^n$, $p \neq q$, there exists $c \in \hat{\Delta}^n$, and $i \in [n]$ such that $p \in \mathcal{F}_i(c)$ and $q \notin \mathcal{F}_i(c)$.

The proof is deferred to [Appendix B.1](#).

Classification calibrated losses have been developed and studied under some different definitions and names ([Zhang, 2004](#); [Bartlett et al., 2006](#)). Below we generalise the notion of c -calibration which was proposed for $n = 2$ by [Reid and Williamson \(2010\)](#) and developed by [Scott \(2011, 2012\)](#) as a generalisation of the notion of classification calibration of [Bartlett et al. \(2006\)](#); confer also [Steinwart \(2007\)](#).

Definition 2 Suppose $\ell : \Delta^n \rightarrow \mathbb{R}_+^n$ is a loss and $c \in \hat{\Delta}^n$. We say ℓ is *c -calibrated at $p \in \Delta^n$* if for all $i \in [n]$ such that $p \notin \mathcal{F}_i(c)$ then $\forall q \in \mathcal{F}_i(c)$, $\underline{L}(p) < \underline{L}(p, q)$. We say that ℓ is *c -calibrated* if $\forall p \in \Delta^n$, ℓ is c -calibrated at p .

[Definition 2](#) means that if the probability vector q one predicts doesn't belong to the same subset (i.e. doesn't predict the same class) as the real probability vector p , then the loss might be larger than $\underline{L}(p)$.

Classification calibration in the sense used by [Bartlett et al. \(2006\)](#) corresponds to $\frac{1}{2}$ -calibrated losses when $n = 2$. If $c_{\text{mid}} := (\frac{1}{2}, \dots, \frac{1}{2})^\top$, c_{mid} -calibration induces Fisher-consistent estimates in the case of classification. Furthermore " ℓ is c_{mid} -calibrated and for all $i \in [n]$, and ℓ_i is continuous and bounded below" is equivalent to " ℓ is infinite sample consistent" as defined by [Zhang \(2004\)](#). This is because if ℓ is continuous and $\mathcal{F}_i(c)$ is closed, then $\forall q \in \mathcal{F}_i(c)$, $\underline{L}(p) < \underline{L}(p, q)$ if and only if $\underline{L}(p) < \inf_{q \in \mathcal{F}_i(c)} \underline{L}(p, q)$.

The following result generalises the correspondence between binary classification calibration and properness ([Reid and Williamson, 2010](#), [Theorem 16](#)) to multiclass losses ($n > 2$).

Proposition 3 A continuous loss $\ell : \Delta^n \rightarrow \mathbb{R}_+^n$ is strictly proper if and only if it is c -calibrated for all $c \in \hat{\Delta}^n$.

Proof (\Rightarrow) Suppose that ℓ is strictly proper. Then for all $c \in \hat{\Delta}^n$, for all $i \in [n]$ such that $p \notin \mathcal{F}_i(c)$ and for all $q \in \mathcal{F}_i(c)$ then $p \neq q$ and thus $\underline{L}(p) < \underline{L}(p, q)$ since ℓ is strictly proper.

(\Leftarrow) Suppose that ℓ is c -calibrated for all $c \in \hat{\Delta}^n$. Suppose $p, q \in \Delta^n$ and $p \neq q$. By [Lemma 1](#) (part 4) one can partition p and q into two different classes; there exists $c \in \hat{\Delta}^n$ and $i \in [n]$ such that $q \in \mathcal{F}_i(c)$ and $p \notin \mathcal{F}_i(c)$. Hence $\underline{L}(p) < \underline{L}(p, q)$ since ℓ is c -calibrated. Since ℓ is continuous and Δ^n is closed, the infimum in the definition of $\underline{L}(p)$ is attained. Since $\underline{L}(p) < \underline{L}(p, q)$ for all $q \neq p$, we conclude $\underline{L}(p) = \underline{L}(p, p)$. Thus ℓ is strictly proper. ■

In particular, a continuous strictly proper loss is c_{mid} -calibrated. Thus for any estimator \hat{q}_n of the conditional probability vector one constructs by minimizing the empirical average of a continuous strictly proper loss, one can build an estimator of the label (corresponding to the largest probability of \hat{q}_n) which is Fisher consistent for the problem of classification.

In the binary case, ℓ is classification calibrated if and only if the following implication holds ([Bartlett et al., 2006](#)):

$$\left(\underline{L}(f_n) \rightarrow \min_g \underline{L}(g) \right) \Rightarrow \left(\mathbb{P}_{\mathcal{X}, \mathcal{Y}}^{\mathbb{P}_{\mathcal{X}, \mathcal{Y}}}(\mathcal{Y} \neq f_n(\mathcal{X})) \rightarrow \min_g \mathbb{P}_{\mathcal{X}, \mathcal{Y}}^{\mathbb{P}_{\mathcal{X}, \mathcal{Y}}}(\mathcal{Y} \neq g(\mathcal{X})) \right). \quad (1)$$

Tewari and Bartlett (2007) have characterised when (1) holds in the multiclass case. Since there is no reason to assume the equivalence between classification calibration and (1) still holds for $n > 2$, we give different names for these two notions. We use *classification calibration* for the notion (Definition 2) linked to Fisher consistency and use *prediction calibrated* (defined below) for the notion of Tewari and Bartlett (equivalent to (1)).

Definition 4 Suppose $\ell: \mathcal{Y} \rightarrow \mathbb{R}_+^n$ is a loss. Let $\mathcal{G}_\ell := \text{co}(\{\ell(v) : v \in \mathcal{Y}\})$, the convex hull of the image of \mathcal{Y} . ℓ is said to be *prediction calibrated* if there exists a prediction function $\text{pred}: \mathbb{R}^n \rightarrow [n]$ such that

$$\forall p \in \Delta^n: \inf_{z \in \mathcal{G}_\ell: p_{\text{pred}(z)} \leq \max_{i \in [n]} p_i} p' \cdot z > \inf_{z \in \mathcal{G}_\ell} p' \cdot z = \underline{L}(p).$$

Suppose that $\ell: \Delta^n \rightarrow \mathbb{R}_+^n$ is such that ℓ is prediction calibrated and $\text{pred}(\ell(p)) \in \arg \max_i p_i$. Then ℓ is c_{mid} -calibrated almost everywhere.

By introducing a *reference link* Ψ (which corresponds to the actual link ψ if ℓ is a proper composite loss $\ell = \lambda \circ \Psi^{-1}$) we now show how the pred function can be canonically expressed in terms of $\arg \max_i p_i$.

Proposition 5 Suppose $\ell: \mathcal{Y} \rightarrow \mathbb{R}_+^n$ is a loss. Let $\tilde{\Psi}: \Delta^n \rightarrow \mathcal{Y}$ satisfy $\tilde{\Psi}(p) \in \arg \min_{v \in \mathcal{Y}} L(p, v)$ and $\lambda = \ell \circ \tilde{\Psi}$. Then λ is proper. If ℓ is prediction calibrated then $\text{pred}(\lambda(p)) \in \arg \max_{i \in [n]} p_i$.

Proof We show first that λ is proper. Let $p \in \Delta^n$. Then

$$\Lambda(p, p) = L(p, \tilde{\Psi}(p)) = L(p, \arg \min_v L(p, v)) = \min_v L(p, v) \leq \min_{q \in \Delta^n} \Lambda(p, q).$$

Thus λ is proper and $\underline{L}(p) = \Lambda(p)$. We now assume that ℓ is prediction calibrated. Suppose that $\text{pred}(z = \lambda(p)) \notin \arg \max_i p_i$. Then $p_{\text{pred}(\lambda(p))} < \max_i p_i$, thus $p' \cdot z = \Lambda(p, p) > \underline{L}(p) = \Lambda(p)$ which contradicts the properness of λ . ■

4. Characterizing Properness

We now present some simple (but new) consequences of properness in the multiclass case (Proposition 6). We also build some connections between the properness of multiclass losses and the properness of binary losses that can be derived from them via a restriction of the multiclass loss to a line connecting two points in the n -simplex (Proposition 7). Finally, we show that multiclass proper losses are effectively characterised by their Bayes risks (Proposition 8) and the continuity of losses is intimately tied to the differentiability of their Bayes risks (Proposition 9). An important implication of these last results is that we are able to study the class of multiclass proper losses by focusing our attention on concave functions defined over probabilities.

To state our propositions we need to introduce monotone functions, directional derivatives, and superdifferentials (cf. (Hiriart-Urruty and Lemaréchal, 2001)). We say $f: C \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ is *monotone* (resp. *strictly monotone*) on C when for all x and y in C ,

$$(f(x) - f(y))' \cdot (x - y) \geq 0 \quad \text{resp.} \quad (f(x) - f(y))' \cdot (x - y) > 0; \quad (2)$$

confer (Hiriart-Urruty and Lemaréchal, 2001; Rockafellar and Wets, 2004). If a function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is concave then $\lim_{t \downarrow 0} \frac{f(x+td) - f(x)}{t}$ exists, and is called the *directional derivative* of f at x in the direction d and is denoted $Df(x, d)$. By analogy with the usual definition of subdifferential for convex functions, we introduce the *superdifferential* $\partial f(x)$ for concave f at x is

$$\begin{aligned} \partial f(x) &:= \{s \in \mathbb{R}^n : s' \cdot y \geq Df(x, y), \forall y \in \mathbb{R}^n\} \\ &= \{s \in \mathbb{R}^n : f(y) \leq f(x) + s' \cdot (y - x), \forall y \in \mathbb{R}^n\}. \end{aligned}$$

Similarly, a vector $s \in \partial f(x)$ is called a *supergradient* of f at x .

Proposition 6 Suppose $\ell: \Delta^n \rightarrow \mathbb{R}_+^n$ is a loss. If ℓ is proper, then $-\ell$ is monotone on Δ^n . Furthermore, if ℓ is strictly proper then it is also invertible.

Proof For all $p, q \in \Delta^n$, $(\ell(p) - \ell(q))' \cdot (p - q) = p' \cdot \ell(p) - q' \cdot \ell(p) + q' \cdot \ell(q) - p' \cdot \ell(q) \leq 0$ since $p' \cdot \ell(p) \leq p' \cdot \ell(q)$. For the strictly proper case, we just have to check that ℓ is injective. By way of contradiction assume ℓ is not invertible. Then there exists $p \neq q$ such that $\ell(p) = \ell(q)$, which means $L(p, p) = L(p, q)$, contradicting the supposed strict properness of ℓ . ■

The following proposition presents several characterisations of multiclass properness. It shows how the characterisation of properness in the general (not necessarily differentiable) multiclass case can be reduced to the binary case. We also show this is equivalent to testing the properness condition for the loss on all possible line segments joining two distributions within the simplex. This latter characterisation can be viewed as a statement connecting ‘‘order sensitivity’’ and properness: the true class probability minimizes the risk and if the prediction moves away from the true class probability in a line then the risk increases. This property appears convenient for optimisation purposes: if one reaches a local minimum in the second argument of the risk and the loss is strictly proper then it is a global minimum. If the loss is proper, such a local minimum is a global minimum or a constant in an open set. But observe that typically one is minimising the full risk $\mathbb{L}(q(\cdot))$ over functions $q: \mathcal{X} \rightarrow \Delta^n$. We note that order sensitivity of ℓ does *not* imply this optimisation problem is well behaved; one needs convexity of $q \mapsto L(p, q)$ for all $p \in \Delta^n$ to ensure convexity of the functional optimisation problem; we characterise when that holds in section 6.4.

Proposition 7 Suppose $\ell: \Delta^n \rightarrow \mathbb{R}_+^n$ is a loss. We define the binary loss

$$\tilde{\ell}^{p,q}: [0, 1] \ni \eta \mapsto \begin{pmatrix} \tilde{\ell}_1^{p,q}(\eta) \\ -\tilde{\ell}_2^{p,q}(\eta) \end{pmatrix} = \begin{pmatrix} q' \cdot \ell(p + \eta(q - p)) \\ p' \cdot \ell(p + \eta(q - p)) \end{pmatrix}.$$

The following statements are equivalent:

1. ℓ is proper;
2. $\tilde{\ell}^{p,q}$ is proper for all $p, q \in \Delta^n$;
3. $\forall p, q \in \Delta^n, \forall 0 \leq h_1 \leq h_2, L(p, p + h_1(q - p)) \leq L(p, p + h_2(q - p))$; and

4. *there exists a concave function $f : \Delta^n \rightarrow \mathbb{R}$ and $\forall q \in \Delta^n$, there exists a supergradient $A(q) \in \partial f(q)$ such that $\forall p, q \in \Delta^n$, $p' \cdot \ell(q) = L(p, q) = f(q) + (p - q)' \cdot A(q)$.*

The proof is deferred to Appendix B.3.

Characterisation (2) shows that in order to check if a loss is proper one need only check the properness in each line. One could use the easy characterization of properness for differentiable binary losses ($\ell : [0, 1] \rightarrow \mathbb{R}_+^2$ is proper if and only if $\forall \eta \in [0, 1]$, $\frac{\ell(\eta)}{1-\eta} = \frac{\ell(\eta)}{\eta} \geq 0$, (Reid and Williamson, 2010)). However this needs to be checked for all lines defined by $p, q \in \partial \Delta^n$. The above result can also be seen as a generalisation of a result by Lambert (2010) who proved that properness is equivalent to the fact that the further your prediction is from reality, the larger the loss (hence the name “order sensitivity”); also confer the results on monotonicity due to Nau (1983). His result relied upon on the total order of \mathbb{R} . In the multiclass case, there does not exist such a total order. Yet, as the above result shows, one can compare two predictions if they are in the same line as the true real class probability.

Characterisation (4) is a restatement of the well known Bregman representation of proper losses: Cid-Suñer and Figueras-Vidal (2001) presented the differentiable case, and Gneiting and Raftery (2007, Theorem 3.2) the general case. This last property gives us the form of the proper losses associated with a given Bayes risk. Suppose $\underline{L} : \Delta^n \rightarrow \mathbb{R}_+$ is concave. The proper losses whose Bayes risk is equal to \underline{L} are

$$\ell : \Delta^n \ni q \mapsto \left(\underline{L}(q) + (e_i - q)' \cdot A(q) \right)_{i=1}^n \in \mathbb{R}_+^n, \forall A(q) \in \partial \underline{L}(q). \quad (3)$$

This result suggests that some information is lost by representing a proper loss via its Bayes risk (when the last is not differentiable). The next proposition elucidates this by showing that proper losses which have the same Bayes risk are equal almost everywhere.

Proposition 8 *Two proper losses $\ell^1, \ell^2 : \Delta^n \rightarrow \mathbb{R}_+^n$ have the same conditional Bayes risk function \underline{L} if and only if $\ell^1 = \ell^2$ almost everywhere. If \underline{L} is differentiable, $\ell^1 = \ell^2$ everywhere.*

Proof A concave function is differentiable almost everywhere (Hiriart-Urruty and Lemaréchal, 2001, theorem 4.2.3). Thus (3) proves that two proper losses ℓ^1 and ℓ^2 which have the same Bayes risk are equal almost everywhere. Suppose now that two proper losses are equal almost everywhere. Then their associated Bayes risks \underline{L}^1 and \underline{L}^2 are equal almost everywhere and continuous (since they are concave). If there exists p such that $\underline{L}^1(p) \neq \underline{L}^2(p)$, then since \underline{L}^1 and \underline{L}^2 are continuous, there exists $\varepsilon > 0$ such that $\forall q \in B(p, \varepsilon) \cap \Delta^n$, $\underline{L}^1(q) \neq \underline{L}^2(q)$, where $B(p, \varepsilon)$ is a ball of radius ε centred at p . Yet this contradicts the fact that \underline{L}^1 and \underline{L}^2 are equal almost everywhere. Hence the Bayes risks are equal everywhere. ■

While the previous proposition shows that losses are closely related to their Bayes risks the next proposition also shows how the continuity of a loss is related to the differentiability of its Bayes risk.

Proposition 9 *Suppose $\ell : \Delta^n \rightarrow \mathbb{R}_+^n$ is a proper loss. Then ℓ is continuous in Δ^n if and only if \underline{L} is differentiable on Δ^n ; ℓ is continuous at $p \in \Delta^n$ if and only if \underline{L} is differentiable at $p \in \Delta^n$.*

The proof of this result can be found in Appendix B.2. This type of relationship is further explored in Section 6.4 where the convexity of a composite loss is related to properties of its Bayes risk.

5. The Proper Composite Representation: Uniqueness and Existence

Many natural predictors have a range other than the simplex (for example those induced by linear functions). It is thus sometimes convenient to define a loss on some set \mathcal{Y} rather than Δ^n ; confer (Reid and Williamson, 2010). The link function explicates the result of Grünwald and Dawid (2004) that every decision problem induces a decision problem expressed in terms of proper losses; (see van Erven et al., 2011, section 6, for further explanation).

Traditionally (McCullagh and Nelder, 1989) links are defined only for binary problems (where one is using univariate probabilities). However there is scattered (but seemingly un-systematic) work on multivariate links (Glonek and McCullagh, 1995; Glonek, 1996), primarily from the perspective of probabilistic modelling (as opposed to the design of loss functions). Sometimes multivariate links are constructed from univariate links (Molenberghs and Lesaffre, 1999).

Composite losses (see the definition in §2) are a way of constructing losses on sets other than Δ^n : given a proper loss $\lambda : \Delta^n \rightarrow \mathbb{R}_+$ and an invertible link $\psi : \Delta^n \rightarrow \mathcal{Y}$, one defines $\lambda \circ \psi : \mathcal{Y} \rightarrow \mathbb{R}_+$ as $\lambda \circ \psi^{-1}$. We now consider the question: given a loss $\ell : \mathcal{Y} \rightarrow \mathbb{R}_+$, when does ℓ have a proper composite representation (whereby ℓ can be written as $\ell = \lambda \circ \psi^{-1}$), and is this representation unique? We first consider the binary case. Here the prediction space $\mathcal{Y} \subseteq \mathbb{R}$ is assumed to be either an interval or the entire real line.

5.1 The Binary Case

Our first result shows that if you can write a binary loss as a proper composite loss, the proper loss defined on the simplex is unique. Furthermore, as soon as the loss is not constant the link function is also unique. If the loss is constant on an interval, then you can choose any value of the link function on this interval which keeps the link function continuous and invertible and still obtain a composite proper loss. The proof can be found in Appendix B.4. As is common in the literature, we write the binary labels as $\{-1, +1\}$ and so the partial losses are ℓ_{-1} and ℓ_{+1} .

Proposition 10 *Suppose $\ell = \lambda \circ \psi^{-1} : \mathcal{Y} \rightarrow \mathbb{R}_+^2$ is a proper composite loss and that the proper loss λ is differentiable and the link function ψ is differentiable and invertible. Then the proper loss λ is unique. Furthermore ψ is unique if $\forall v_1, v_2 \in \mathcal{Y}$, $\exists v \in [v_1, v_2]$, $\ell_1(v) \neq 0$ or $\ell_{-1}(v) \neq 0$. If there exists $v_1, v_2 \in \mathcal{Y}$ such that $\ell_1(v) = \ell_{-1}(v) = 0 \forall v \in [v_1, v_2]$, one can choose any $\psi|_{[v_1, v_2]}$ such that ψ is differentiable, invertible and continuous in $[v_1, v_2]$ and obtain $\ell = \lambda \circ \psi^{-1}$, and ψ is uniquely defined where ℓ is invertible.*

We now determine necessary and sufficient conditions for a binary loss to be expressed as a proper composite loss. Once again, the proof is deferred to Section B.5.

Proposition 11 *Suppose $\ell : \mathcal{Y} \rightarrow \mathbb{R}_+^2$ is a differentiable binary loss such that $\forall v \in \mathcal{Y}$, $\ell_{-1}(v) \neq 0$ or $\ell_1(v) \neq 0$. Then ℓ can be expressed as a proper composite loss if and only if the following three conditions hold:*

1. ℓ_1 is decreasing (increasing);
2. ℓ_{-1} is increasing (decreasing); and
3. $f: \mathcal{Y} \ni v \mapsto \frac{\ell_1(v)}{\ell_{-1}(v)}$ is strictly increasing (decreasing) and continuous.

Observe that the last condition is always satisfied if both ℓ_1 and ℓ_{-1} are convex.

5.2 Binary Margin Losses

Suppose $\varphi: \mathbb{R} \rightarrow \mathbb{R}_+$ is a function. The loss $\ell_\varphi: \mathcal{Y} \ni v \mapsto (\ell_{-1}(v), \ell_1(v))' = (\varphi(-v), \varphi(v))' \in \mathbb{R}_+^2$ is called a binary *margin loss*. Binary margin losses are often used for classification problems. We will now show how the previous proposition applies to them.

Corollary 12 Suppose $\varphi: \mathbb{R} \rightarrow \mathbb{R}_+$ is differentiable and $\forall v \in \mathbb{R}, \varphi'(v) \neq 0$ or $\varphi'(-v) \neq 0$. Then ℓ_φ can be expressed as a proper composite loss if and only if $f: \mathbb{R} \ni v \mapsto -\frac{\varphi'(v)}{\varphi'(-v)}$ is strictly monotonic continuous and φ is monotonic.

If φ is convex or concave then f defined above is monotonic. However not all binary margin losses are composite proper losses. One can even build a smooth margin loss which cannot be expressed as a proper composite loss. Consider $\varphi(x) = \frac{1 - \arctan(\frac{x-1}{x})}{x}$. Then $f(v) = \frac{\varphi'(v)}{\varphi'(-v)} = \frac{2-2x+2}{x^2-2x+2}$ which is not invertible. This loss is illustrated in Figure 5, after some additional concepts are introduced.

5.3 The Multiclass Case

Uniqueness of the composite representation remains straightforward in the multiclass case.

Proposition 13 Suppose a loss $\ell: \mathcal{Y} \rightarrow \mathbb{R}_+^n$ has two proper composite representations $\ell = \lambda \circ \psi^{-1} = \mu \circ \phi^{-1}$ where λ and μ are proper losses with corresponding Bayes risks Δ and \underline{M} respectively, and ψ and ϕ are continuous invertible link-functions. Then $\lambda = \mu$ almost everywhere where.

If ℓ is continuous and has a composite representation, then the proper loss (in the decomposition) is unique ($\lambda = \mu$ everywhere).

If ℓ is invertible and has a composite representation, then the representation is unique.

Proof $\Delta(p) = \inf_q p' \cdot \lambda(q) = \inf_q p' \cdot \ell(\psi(q)) = \inf_v L(p, v)$ (since ψ is invertible)
 $= \inf_v L(p, v) = \inf_v L(p, \phi(q)) = \underline{M}(p)$.

Then λ and μ are two proper losses which have the same Bayes risk, so these two losses are equal almost everywhere.

If moreover ℓ is continuous, $\lambda = \ell \circ \psi$ and $\mu = \ell \circ \phi$ are continuous. So $\lambda = \mu$ everywhere. If moreover ℓ is invertible, $\psi = \lambda \circ \ell^{-1}$ and $\phi = \mu \circ \ell^{-1}$. So ψ and ϕ are also equal almost everywhere and as they are continuous, they are equal everywhere. So $\lambda = \ell \circ \psi = \ell \circ \phi = \mu$. ■

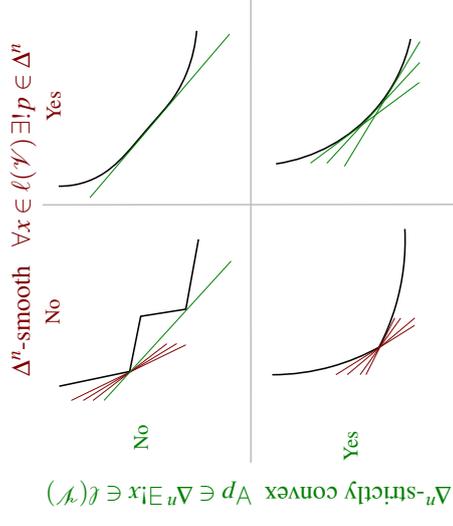


Figure 3: Illustration of Δ^n -smoothness and Δ^n -strict convexity. The hyperplanes witness the possession or non-possession of the respective properties.

Characterising the existence of a composite representation is more complex in the multiclass case. We need to introduce some definitions: We make use of a set of *hyperplanes* for $p \in \Delta^n$ and $\beta \in \mathbb{R}$,

$$h_p^\beta := \{x \in \mathbb{R}^n; x' \cdot p = \beta\}.$$

A hyperplane h_p^β *supports* a set A at $x \in A$ when $x \in h_p^\beta$ and for all $a \in A$, $a' \cdot p \geq \beta$ or for all $a \in A$, $a' \cdot p \leq \beta$. Given a loss $\ell: \mathcal{Y} \rightarrow \mathbb{R}_+^n$, the *loss image* $\ell(\mathcal{Y}) := \{\ell(v); v \in \mathcal{Y}\}$.

Definition 14 Let $\mathfrak{S}(p, x) := \{\ell(\mathcal{Y}) \text{ is supported by } h_p^\beta \text{ at } x \text{ for some } \beta \in \mathbb{R}\}$.

1. A loss image $\ell(\mathcal{Y})$ is Δ^n -strictly convex if for all $p \in \Delta^n$ there exists a unique $x \in \ell(\mathcal{Y})$ such that $\mathfrak{S}(p, x)$.
2. A loss image $\ell(\mathcal{Y})$ is Δ^n -smooth if for all $x \in \ell(\mathcal{Y})$ there exists a unique $p \in \Delta^n$ such that $\mathfrak{S}(p, x)$.

This definition is illustrated in Figure 3. Dropping the uniqueness requirement in these definitions would drastically change things: since we will require ℓ is continuous, $\ell(\mathcal{Y})$ is always closed. Since by assumption $\ell(\mathcal{Y}) \subset [0, \infty)^n$ every such loss satisfies the weakened version of Δ^n -strict convexity: for all $p \in \Delta^n$ there exists $x \in \ell(\mathcal{Y})$ such that $\mathfrak{S}(p, x)$. The weakened version of Δ^n -smoothness requires that for all $x \in \ell(\mathcal{Y})$ there exists $p \in \Delta^n$ such that $\mathfrak{S}(p, x)$ is

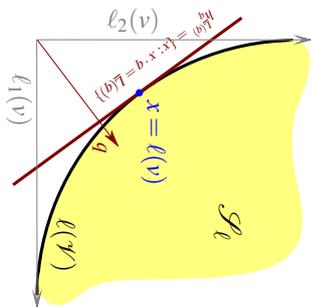


Figure 4: Illustration of geometry of loss functions. The locus of the vector valued loss ℓ is plotted as v varies over \mathcal{Y} . The superprediction set \mathcal{S}_ℓ is the region to the “north-east” of the loss image $\ell(\mathcal{Y})$. The hyperplane $h_q^{\ell(v)}$ has normal vector q and offset $\underline{L}(q)$. It supports \mathcal{S}_ℓ at the point $x = \ell(v)$ indicating the Bayes risk is achieved at v for the true probability q .

a convexity-like requirement. (Confer the following result (Schneider, 1993, Theorem 1.3.3): *Suppose A is closed set such that $A \neq \emptyset$ and through each boundary point of A there is a support plane to A ; then A is convex*)

The name “ Δ^n -strictly convex” is justified by the observation that replacing Δ^n by $B_{\mathbb{R}^n}$ (the ℓ_1^n unit ball) gives a natural definition of strict convexity of a general set in \mathbb{R}^n . We also observe that both Δ^n -strict convexity and Δ^n -smoothness are closely related to the curvature of the Bayes risk \underline{L} by way of the fact that the support function of the set $\ell(\mathcal{Y})$ (restricted to Δ^n) is the Bayes risk; confer (Williamson, 2014). Specifically, Δ^n -strict convexity is equivalent to the Hessian $H\underline{L}(p)$ being non-singular for all $p \in \Delta^n$ while Δ^n -smoothness is implied whenever $\underline{L}(p)$ is continuously differentiable.

Suppose $A, B \subset \mathbb{R}^n$. Then the *Minkowski sum* $A + B := \{a + b : a \in A, b \in B\}$.

Definition 15 Given a loss $\ell : \mathcal{Y} \rightarrow \mathbb{R}_+^n$, we denote by

$$\mathcal{S}_\ell := \ell(\mathcal{Y}) + [0, \infty)^n = \{x \in \mathbb{R}_+^n : \exists v \in \mathcal{Y}, \forall i \in [n], x_i \geq \ell_i(v)\}$$

the *superprediction set* of ℓ (Kahnishkan and Vyugin, 2008).

One can characterise the existence of proper composite representations in terms of properties the superprediction set. We start with an old result; confer (Dawid, 2007).

Proposition 16 Every continuous proper loss has a convex superprediction set.

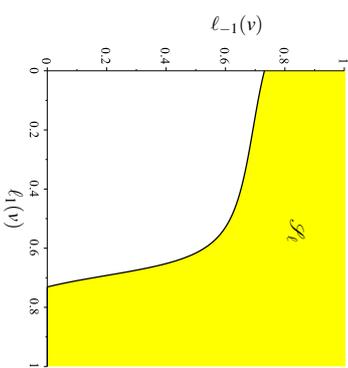


Figure 5: Superprediction set of a binary margin loss which is not a composite proper loss. See text following Corollary 12.

Proof Suppose ℓ is proper but \mathcal{S}_ℓ is not convex. Then there exists $x_0 \in \ell(\Delta^n)$ such that $\ell(\Delta^n)$ is not supported at x_0 by any hyperplane h_p with normal vector $p \in \Delta^n$. Let $q_0 \in \Delta^n$ be such that $\ell(q_0) = x_0$. Then there is a hyperplane h_{q_0} (with normal q_0) that supports $\ell(\Delta^n)$ at some $x_1 \neq x_0$. Thus $q_0^\top \ell(q)$ is minimised at q_1 and not minimised at q_0 and thus ℓ is can not be proper—a contradiction. ■

The geometry of continuous proper losses is illustrated (for $n = 2$) in Figure 4. The superprediction set of the margin loss discussed following Corollary 12 is not convex as can be seen in Figure 5.

Continuous proper losses are quasiconvex, canonically so, as the following result shows.

Proposition 17 Suppose $\ell : \Delta^n \rightarrow \mathbb{R}_+^n$ is a continuous proper loss. Then its superprediction set \mathcal{S}_ℓ is convex and, for all $p \in \Delta^n$, the function $f_p(q) := L(p, q) := p \cdot \ell(q)$ is quasi-convex. Conversely, suppose $f_p(q) := p \cdot \ell(q)$ is quasi-convex in q for all $p \in \Delta^n$. Then there is a unique convex set S such that $\mathcal{S}_\ell = S$ and ℓ is necessarily proper.

The proof is in Appendix B.6. Some (but not all) proper losses are in addition convex; this is studied in more detail in Section 6.4 below.

Working with \mathcal{S}_ℓ is problematic for characterising the existence of strictly proper composite representations (essentially because while for a strictly proper loss ℓ , $\ell(\Delta^n)$ is Δ^n -strictly convex, \mathcal{S}_ℓ is not strictly convex (because of the flat spots at the extremes—bounded losses have superprediction sets with flats parallel to the axes by construction)¹. We will thus characterise proper and strictly proper composite representations in terms of properties of $\ell(\mathcal{Y})$ rather than \mathcal{S}_ℓ .

¹ It turns out that by starting with the superprediction set, and defining the loss in terms of the (super-) gradient of the (concave) support function of the superprediction set, these difficulties can be avoided (Williamson, 2014).

Proposition 18 Suppose $\ell: \mathcal{Y} \rightarrow \mathbb{R}_+^n$ is a continuous loss. ℓ has a proper composite representation if and only if $\ell(\mathcal{Y})$ is Δ^n -smooth. Additionally, ℓ is strictly proper composite if and only if $\ell(\mathcal{Y})$ is also Δ^n -strictly convex.

The proof is in Section B.7.

6. Implications: Mixability, Admissibility, Minimality and Convexity

We now consider some of the implications that the proper composite representation has for several previously studied properties of loss functions.

6.1 Mixability

Mixability is a fundamental property of a loss function in the study of “prediction with expert advice.” In this setting learning takes place in fixed number of sequential rounds. Each round a learner is presented with predictions some finite number of experts. The learner then makes a prediction and the outcome for that round is revealed. The learner’s and experts’ predictions are assessed using some predefined loss function and the aim of the learner is to incur a total loss not much worse than the best expert – *i.e.*, the one with the smallest total loss. The difference between the learner’s total loss and that of the best expert is known as the *regret*. In his seminal work, [Vovk \(1995\)](#) showed that no matter how the experts behave, there exists a strategy for the learner (called the “aggregating algorithm”) that guarantees a regret bounded by $\frac{\ln K}{\eta}$ where K is the number of experts and η is a positive number called the *mixability constant* (defined below) that only depends on the loss. Losses for which this constant is defined are called *mixable*. Furthermore, this constant *characterises* when such a constant regret bound is possible. That is, if a loss is not mixable then there is no strategy the learner can use to guarantee a constant regret bound.

Formally, mixability of a loss ℓ is defined in terms of the convexity of a transformation of the loss’s superprediction set \mathcal{S}_ℓ (see Definition 15). We say that for $\eta > 0$ the η -*exponentiated superprediction set* is the image of $\mathcal{S}_\ell \subset \mathbb{R}^n$ under the mapping $E_\eta: \mathbb{R}^n \rightarrow \mathbb{R}_+^n$ defined by $E_\eta(x) := (e^{-\eta x_i})_{i=1}^n$. A loss ℓ is said to be η -*mixable* if its η -exponentiated superprediction set is convex. The *mixability* of ℓ is the smallest value of η for which ℓ is η -mixable. For further details, the reader is referred to papers by [Vovk \(1995\)](#), [Kalmishkan and Vyugin \(2008\)](#); [Vovk and Zhdanov \(2009\)](#).

Recently, [van Erven et al. \(2012b\)](#)² have shown that the mixability of a loss is related to the curvature of the loss’s Bayes risk relative to the curvature of the Bayes risk for log loss. The main result here builds on some of the insights from that work and shows that mixable losses (under mild conditions) always have proper composite representations.

For $\alpha \in (0, 1)$ we write $\tilde{\alpha} := 1 - \alpha$. For $x, y \in \mathbb{R}^n$, $x \leq y \Leftrightarrow (x_i \leq y_i, \forall i \in [n])$. We now give a necessary condition for mixability.

2. An extension of this notion of mixability can be related to a natural convex duality ([Reid et al., 2015](#)).

Lemma 19 Suppose $\ell: \mathcal{Y} \rightarrow \mathbb{R}_+^n$, $x_0 = \ell(v_0)$, $x_1 = \ell(v_1)$ with $x_0 \neq x_1$. For $\alpha \in (0, 1)$, define $x_\alpha := \tilde{\alpha}x_0 + \alpha x_1$ and $v_\alpha = \tilde{\alpha}v_0 + \alpha v_1$. If for some α

$$x_\alpha \leq \ell(v_\alpha) \quad (4)$$

then ℓ is not mixable.

Proof Pick some $\eta > 0$. Let $f_\eta(a) = e^{-\eta a}$ for $a \in \mathbb{R}$ so that for $x \in \mathbb{R}^n$ we have $E_\eta(x) = (f_\eta(x_i))_{i=1}^n$. Observe that the function f_η is strictly monotone decreasing ($a < b \Rightarrow f_\eta(a) > f_\eta(b)$) and strictly convex ($\tilde{\alpha}f_\eta(a) + \alpha f_\eta(b) > f_\eta(\tilde{\alpha}a + \alpha b)$). For $i \in [n]$ set $x_{0,i} = \ell_i(v_0)$ and $x_{1,i} = \ell_i(v_1)$. By assumption, we have

$$\tilde{\alpha}x_{0,i} + \alpha x_{1,i} \leq \ell_i(\tilde{\alpha}v_0 + \alpha v_1), \quad \forall i \in [n],$$

which by strict monotonicity

$$\Rightarrow f_\eta(\tilde{\alpha}x_{0,i} + \alpha x_{1,i}) \geq f_\eta(\ell_i(\tilde{\alpha}v_0 + \alpha v_1)), \quad \forall i \in [n],$$

and hence by strict convexity

$$\begin{aligned} &\Rightarrow \tilde{\alpha}f_\eta(x_{0,i}) + \alpha f_\eta(x_{1,i}) > f_\eta(\ell_i(\tilde{\alpha}v_0 + \alpha v_1)), \quad \forall i \in [n] \\ &\Leftrightarrow \tilde{\alpha}E_\eta(\ell(v_0)) + \alpha E_\eta(\ell(v_1)) > E_\eta(\ell(\tilde{\alpha}v_0 + \alpha v_1)) \end{aligned}$$

and thus ℓ is not mixable since we have witnessed the non-convexity of the η -exponentiated superprediction set for ℓ . ■

[van Erven et al. \(2012b\)](#) showed that (under some mild conditions) a proper loss λ and the composite loss λ^ψ obtained via the reference link $\tilde{\psi}$ (see Proposition 5) share the same mixability constant. We now show that mixable losses always have strictly proper composite representations.

Proposition 20 Suppose $\ell: \mathcal{Y} \rightarrow \mathbb{R}_+^n$ is a Δ^n -smooth continuous loss. If ℓ is mixable then ℓ has a strictly proper composite representation.

Proof We prove the contrapositive. Lack of a strictly proper composite representation is equivalent then to $\ell(\mathcal{Y})$ being not Δ^n -strictly convex. Suppose then that $\ell(\mathcal{Y})$ is indeed not Δ^n -strictly convex. There are two possibilities to consider:

1. There exists $p \in \Delta^n$ such that there is no $x \in \ell(\mathcal{Y})$ such that $\ell(\mathcal{Y})$ is supported by h_p^β at x for some $\beta \in \mathbb{R}$; or
2. There exists $p \in \Delta^n$ such that there exists $v_0, v_1 \in \mathcal{Y}$, $v_0 \neq v_1$, $x_0 = \ell(v_0)$, $x_1 = \ell(v_1)$, $\exists \beta \in \mathbb{R}$, h_p^β supports $\ell(\mathcal{Y})$ at x_1 and x_2 .

Since $\ell(\mathcal{Y}) \subset [0, \infty)^n$ and ℓ is continuous (and hence $\ell(\mathcal{Y})$ is closed), for all $p \in \Delta^n$ there always exists $x \in \ell(\mathcal{Y})$ such that h_p^β supports $\ell(\mathcal{Y})$ at x . Thus under the hypothesis, case 2 must always hold. Then by continuity of ℓ and the definition of a supporting hyperplane, there exists $\alpha \in (0, 1)$ such that (4) holds and so ℓ is not mixable. ■

6.2 Admissibility

The above results are strongly related to the classical notion of admissibility (Ferguson, 1967; Chernoff and Moses, 1986; Kiefer, 1987), which is particularly simple in our situation. We adapt the terminology of Ferguson (1967) to be consistent with elsewhere in the present paper:

Definition 21 Suppose $\ell : \mathcal{Y} \rightarrow \mathbb{R}_+^n$ is a loss. A prediction $v_1 \in \mathcal{Y}$ is better than $v_2 \in \mathcal{Y}$ if $\ell(v_1) \leq \ell(v_2)$ and for some $i \in [n]$, $\ell_i(v_1) < \ell_i(v_2)$. A prediction v_1 is **equivalent** to v_2 if $\ell(v_1) = \ell(v_2)$. A prediction $v \in \mathcal{Y}$ is **admissible** if there is no prediction better than v . If a prediction $v \in \mathcal{Y}$ is the Bayes-optimal for some distribution p , that is for all $v \in \mathcal{Y}$ there exists $p \in \Delta^n$ such that $v \in \arg\min_{v \in \mathcal{Y}} p' \cdot \ell(v)$, then we say v is **strongly admissible**.

Ferguson (1967, Theorem 1, page 60) states the following (which we present for invertible losses, so that $\ell(v_1) = \ell(v_2) \Rightarrow v_1 = v_2$).

Proposition 22 Suppose $\ell : \mathcal{Y} \rightarrow \mathbb{R}_+^n$ is invertible and $p \in \Delta^n$. If $v \in \mathcal{Y}$ is the unique prediction such that $L(p, v) = L(p)$, then v is admissible.

Proposition 18 then implies the following.

Corollary 23 Suppose $\ell : \mathcal{Y} \rightarrow \mathbb{R}_+^n$ is continuous and invertible. If ℓ has a strictly proper composite representation then all $v \in \mathcal{Y}$ are admissible and strongly admissible.

Proof If ℓ has a strictly proper composite representation, then $\ell(\mathcal{Y})$ is Δ^n -strictly convex and thus for all $p \in \Delta^n$ there exists a unique $x \in \ell(\mathcal{Y})$ such that $h_p^{L(p)}$ supports $\ell(\mathcal{Y})$ at x . Thus by Proposition 22, v such that $\ell(v) = x$ is an admissible prediction. Furthermore, since $\ell(\mathcal{Y})$ is Δ^n -smooth, this previous argument actually holds for all $v \in \mathcal{Y}$ and thus ℓ is admissible. Furthermore, it follows directly from the definition of Δ^n -smoothness that all v are strongly admissible. ■

Proposition 24 If $\ell : \mathcal{Y} \rightarrow \mathbb{R}_+^n$ is continuous and has a proper composite representation then every prediction is admissible.

Proof We will prove the contrapositive: Suppose a continuous loss $\ell : \mathcal{Y} \rightarrow \mathbb{R}_+^n$ is such that there exist $x_0, x_1 \in \ell(\mathcal{Y})$ with x_1 better than x_0 . Then ℓ can not have a proper composite representation. Observe that “ x_1 is better than x_0 ” is equivalent to

$$\begin{aligned} \forall i \in [n], \quad \ell_i'(x_0 - x_1) &\geq 0 \\ \exists i \in [n], \quad \ell_i'(x_0 - x_1) &> 0. \end{aligned}$$

Consider two mutually exclusive and exhaustive cases:

1. $\ell_i'(x_0 - x_1) > 0, \forall i \in [n]$. Then for all $p \in \Delta^n$, $p' \cdot (x_0 - x_1) > 0 \Rightarrow p' \cdot x_0 > p' \cdot x_1$ and thus $\ell(\mathcal{Y})$ can not be supported at x_0 by h_p^β for any $p \in \Delta^n$ and thus $\ell(\mathcal{Y})$ is not Δ^n -smooth.
2. Alternatively suppose

$$\ell_i'(x_0 - x_1) \begin{cases} = 0, & i \in I \subset [n] \\ > 0, & i \in [n] \setminus I \end{cases}$$
 with $1 \leq |I| < n$. Consider the two mutually exclusive subcases over $p \in \Delta^n$:

- (a) $p_i > 0$ for some $i \in [n] \setminus I$. Then $p' \cdot (x_0 - x_1) > 0$ and $\ell(\mathcal{Y})$ can not be supported at x_0 by h_p^β for any $\beta \in \mathbb{R}$.
- (b) $p_i = 0$ for all $i \in [n] \setminus I$. Then $p' \cdot (x_0 - x_1) = 0$ in which case $\ell(\mathcal{Y})$ is supported by h_p^β for some β at both x_0 and x_1 .

In either of these subcases, the Δ^n -smoothness condition is violated.

Thus in both cases we have shown $\ell(\mathcal{Y})$ can not be Δ^n -smooth and by Proposition 18 can not have a proper composite representation. ■

As can be seen in Figure 6, there can be no hope of a converse: mere admissibility of every prediction $x \in \ell(\mathcal{Y})$ can not imply that ℓ has a proper composite representation.

However strong admissibility of every prediction implies $\ell(\mathcal{Y})$ is Δ^n -smooth and so if ℓ is continuous, strong admissibility of every prediction implies (via Proposition 18) that ℓ has a proper composite representation.

The relationship between strict convexity of \mathcal{S}_ℓ and admissibility is not new (Brown, 1981); but the connection with our characterisation of composite proper losses is new.

We conclude that if ℓ is continuous and invertible and we desire that all predictions are admissible, then it suffices to only consider losses with a proper composite representation. Continuous invertible losses that do not have a proper composite representation are “redundant” in the sense that there are guaranteed to exist predictions that are not Bayes optimal for any true distribution.

6.3 Minimality

We say a loss $\ell : \mathcal{Y} \rightarrow \mathbb{R}_+^n$ is **minimax** if its conditional risk $L(p, v) = p' \cdot \ell(v)$ satisfies

$$\max_{p \in \Delta^n} \min_{v \in \mathcal{Y}} L(p, v) = \min_{v \in \mathcal{Y}} \max_{p \in \Delta^n} L(p, v). \quad (5)$$

Minimality of proper losses has been studied in a very general setting by Grünwald and Dawid (2004) who showed the connection between robust Bayes procedures and maximum entropy; confer classical results presented, for example, by Ferguson (1967). In this brief subsection we point out some simple implications of our earlier results. Setting $\mathcal{Y} = \Delta^n$, observe that for all proper losses $\lambda : \Delta^n \rightarrow \mathbb{R}_+^n$, $p \mapsto \Lambda(p, q) = p' \cdot \lambda(q)$ is linear for all $q \in \Delta^n$, and if λ is also continuous, by Proposition 17 $q \mapsto \Lambda(p, q)$ is quasi-convex for all $p \in \Delta^n$. It thus follows from the minimax theorem of Sion (1958) that all continuous proper losses satisfy

$$\max_{p \in \Delta^n} \min_{q \in \Delta^n} \Lambda(p, q) = \min_{q \in \Delta^n} \max_{p \in \Delta^n} \Lambda(p, q) \quad (6)$$

and are thus minimax.

Suppose $\ell = \lambda \circ \psi^{-1} : \mathcal{Y} \rightarrow \mathbb{R}_+^n$ is a proper composite loss, with conditional risk $L(p, v) = \Lambda(p, \psi^{-1}(v))$. Since ψ^{-1} is invertible,

$$\max_{p \in \Delta^n} \min_{v \in \mathcal{Y}} L(p, v) = \max_{p \in \Delta^n} \min_{q \in \Delta^n} \Lambda(p, q), \quad (7)$$

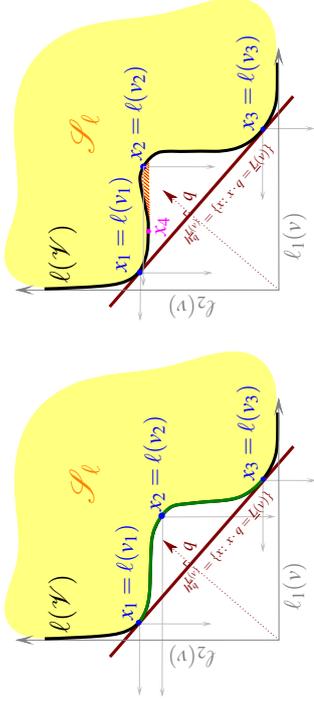


Figure 6: *Left:* Illustration of a continuous loss ℓ (which can be presumed invertible) with a non-convex superprediction set. For true probability q , v_1 and v_2 both are Bayes optimal since $q \cdot \ell(v_1) = q \cdot \ell(v_2) = \underline{\ell}(v)$; thus $h_{\underline{\ell}(v)}^{\Delta^N} = h_{\underline{\ell}(v_2)}^{\Delta^N}$ supports $\ell(\mathcal{Y})$ at both x_1 and x_3 . The point x_2 is never a member of a supporting hyperplane of $\ell(\mathcal{Y})$ and is thus never the Bayes optimal prediction for any q and so not strongly admissible. The green line indicates the set of predictions that are not strongly admissible—they will never be Bayes optimal for any $q \in \Delta^N$. Such predictions are, however, admissible, as can be seen by the grey translated negative orthants centred at x_1 , x_2 and x_3 (each orthant does not contain any other predictions “better than” them). All the other predictions whose image lies in the black line are both admissible and strongly admissible. The loss image $\ell(\mathcal{Y})$ is not Δ^N -smooth because there exist no $p \in \Delta^N$ that supports $\ell(\mathcal{Y})$ at x_2 . Hence by Proposition 18, ℓ can not have a proper composite representation. *Right:* Similar to the figure on the left, except there are now some predictions, such as x_2 , which are not admissible: x_4 is *better than* x_2 as can be seen since x_4 is contained in the interior of the shifted negative orthant centred at x_2 . Note in this case the boundary of the super-prediction set \mathcal{S}_ℓ does not equal $\ell(\mathcal{Y})$ (see the part of \mathcal{S}_ℓ cross-hatched in red). This loss can not have a proper composite representation by Proposition 24.

where by the relationship between q and v , $\arg \min_{v \in \mathcal{Y}} L(p, v) = \psi \left(\arg \min_{q \in \Delta^N} \Lambda(p, q) \right)$. Similarly,

$$\min_{v \in \mathcal{Y}} \max_{p \in \Delta^N} L(p, v) = \min_{q \in \Delta^N} \max_{p \in \Delta^N} \Lambda(p, q). \quad (8)$$

Since λ is proper, Λ satisfies (6) which combined with (7) and (8) proves the following.

Proposition 25 *Every continuous proper composite loss is minimax.*

Note that this alone does not imply that all continuous proper composite losses are quasi-convex, which would follow if ψ mapped convex sets to convex sets; however this can not be true in

general in \mathbb{R}^n because convexity preserving mappings must be affine (Webster, 1994, Theorem 7.3.7); confer (Meyer and Kay, 1973). Recall Proposition 17 showed the quasi-convexity of all proper losses.

Proposition 25 means that the use of the classical minimax theorem by Abernethy et al. (2009) in order to prove their main result for *convex* losses can be foregone; their result also holds for arbitrary continuous proper composite losses.

6.4 Convexity

In order to computationally optimise models with respect to a loss function it is convenient if the loss is convex. In this subsection we develop conditions for the convexity of multiclass composite proper losses. We assume throughout this section that the loss and link are twice differentiable. We start by proving some identities for their first and second derivatives.

6.4.1 TECHNICAL PRELIMINARIES

Suppose $\ell = \lambda \circ \psi^{-1}$ is composed of the proper loss $\lambda: \Delta^n \rightarrow \mathbb{R}_+^+$ and the inverse of the link $\psi: \Delta^n \rightarrow \mathcal{Y}$. In order to simplify the calculation of derivatives for the function $\ell: \mathcal{Y} \rightarrow \mathbb{R}_+^+$ we will assume the set \mathcal{Y} is a flat, $(n-1)$ -dimensional, convex subset of \mathbb{R}_+^+ . We do so since if \mathcal{Y} were some arbitrary manifold the extra definitions required to make sense of convexity (e.g., in terms of geodesics) and derivatives on manifolds would obscure the gist of the results below. Furthermore, little is lost either practically or theoretically by assuming a simple \mathcal{Y} . In practice, predictions are usually vectors in \mathbb{R}_+^+ , and in theory one could always choose a parametrisation of \mathcal{Y} in terms of some simpler space \mathcal{U} and redefine the link via composition with that parametrisation. Alternatively, since links must be invertible, a composite loss could be defined by a choice of loss and choice of *inverse link* $\psi^{-1}: \mathcal{Y} \rightarrow \Delta^n$ for a \mathcal{Y} assumed to be flat, etc.

Recalling the convention that $\bar{n} := n-1$, let $v \in \mathcal{Y}$ fixed but arbitrary with corresponding $\bar{p} = \psi^{-1}(v)$ where $\bar{\psi}(\bar{p}) := \psi(\bar{p}_1, \dots, \bar{p}_{\bar{n}}, p_n)$ with $p_n := \sum_{i=1}^{\bar{n}} \bar{p}_i$ is the induced function from $\bar{\Delta}^{\bar{n}}$ to \mathcal{Y} . By the chain rule and the inverse function theorem, the derivatives for each of the partial losses ℓ_i satisfy

$$D\ell_i(v) = D[\lambda_i(\bar{\psi}^{-1}(v))] = D\lambda_i(\bar{p}) \cdot [D\bar{\psi}(\bar{p})]^{-1}. \quad (9)$$

We use e_i^j to denote the i th n -dimensional unit vector, $e_i^i = (0, \dots, 0, 1, 0, \dots, 0)^T$ when $i \in [n]$, and define $e_i^i = 0_n$ when $i > n$. We can now write $D\lambda_i(\bar{p})$ in terms of the $n \times \bar{n}$ matrix $D\bar{\lambda}(\bar{p})$ using $D\lambda_i(\bar{p}) = (e_i^i)^T \cdot D\lambda(\bar{p}) = (D\bar{\lambda}(\bar{p})^T \cdot D\lambda_n(\bar{p}))^T$, where $\bar{\lambda}(\bar{p}) = (\lambda_1(\bar{p}), \dots, \lambda_{\bar{n}}(\bar{p}))^T$, and so

$$D\lambda_i(\bar{p}) = (e_i^i)^T \cdot D\lambda(\bar{p}) = (e_i^i)^T \cdot \begin{pmatrix} D\bar{\lambda}(\bar{p}) \\ D\lambda_n(\bar{p}) \end{pmatrix}. \quad (10)$$

Furthermore, since λ is proper, Lemma 6 of (van Erven et al., 2012b) means we can use the relationship between a proper loss and its projected Bayes risk $\bar{L} := \underline{L} \circ \Pi_{\Delta}^{-1}$ to write

$$D\bar{\lambda}(\bar{p}) = W(\bar{p}) \cdot H\bar{L}(\bar{p}) \quad (11)$$

$$D\lambda_n(\bar{p}) = y(\bar{p})^T \cdot D\bar{\lambda}(\bar{p}) \quad (12)$$

where $W(\tilde{\beta}) := I_n - \mathbb{1}_n \cdot \tilde{p}'$ and where $\gamma(\tilde{\beta}) := -\tilde{\beta}'/p_n(\tilde{\beta})$ and $p_n(\tilde{\beta}) := 1 - \sum_{i \in [n]} p_i$. Thus, combining (10–12) we have for all $i \in [n]$

$$\begin{aligned} D\lambda_i(\tilde{\beta}) &= (e_i^i)' \cdot W(\tilde{\beta}) \cdot H\tilde{L}(\tilde{\beta}) \\ &= ((e_i^i)' - (e_i^i)' \cdot \mathbb{1}_n \cdot \tilde{p}') \cdot H\tilde{L}(\tilde{\beta}) \\ &= (e_i^i - \tilde{p})' \cdot H\tilde{L}(\tilde{\beta}) \end{aligned} \quad (13)$$

and

$$\begin{aligned} D\lambda_n(\tilde{\beta}) &= \gamma(\tilde{\beta})' \cdot W(\tilde{\beta}) \cdot H\tilde{L}(\tilde{\beta}) \\ &= \frac{-1}{p_n(\tilde{\beta})} \tilde{p}' \cdot (I_n - \mathbb{1}_n \cdot \tilde{p}') \cdot H\tilde{L}(\tilde{\beta}) \\ &= \frac{-1}{p_n(\tilde{\beta})} (\tilde{p}' - (1 - p_n(\tilde{\beta}))\tilde{p}') \cdot H\tilde{L}(\tilde{\beta}) \\ &= -\tilde{p}' \cdot H\tilde{L}(\tilde{\beta}). \end{aligned} \quad (14)$$

Finally, noting that by definition $e_i^i = 0$, (14) and (13) can be merged and combined with (9) to obtain the following proposition.

Proposition 26 For all $i \in [n]$, $\tilde{p} \in \tilde{\Delta}^n$ (the relative interior of $\tilde{\Delta}^n$), and $v = \tilde{\psi}(\tilde{\beta})$,

$$D\ell_i(v) = -(e_i^i - \tilde{p})' \cdot \kappa(\tilde{\beta}) \quad (15)$$

where

$$\kappa(\tilde{\beta}) := -H\tilde{L}(\tilde{\beta}) \cdot [D\tilde{\psi}(\tilde{\beta})]^{-1}. \quad (16)$$

Using the definition of the Hessian $H\ell_i = D([D\ell_i]')$ and the product rule (31) gives

$$\begin{aligned} D[D\ell_i(v)] &= D_{v,i} \left[\underbrace{[D\tilde{\psi}(\tilde{\beta})]^{-1}}_{f(\tilde{\beta})} \cdot \underbrace{H\tilde{L}(\tilde{\beta}) \cdot (e_i^i - \tilde{p})}_{\kappa(\tilde{\beta})} \right] \\ &= \left((e_i^i - \tilde{p})' \otimes I_n \right) \cdot D_{v,i} [f(\tilde{\beta})] + (I_1 \otimes f(\tilde{\beta})) \cdot D(e_i^i - \tilde{\psi}^{-1}(v)) \\ &= \left((e_i^i - \tilde{p})' \otimes I_n \right) \cdot D_{v,i} \left[H\tilde{L}(\tilde{\beta}) \cdot [D\tilde{\psi}(\tilde{\beta})]^{-1} \right] - \left([D\tilde{\psi}(\tilde{\beta})]^{-1} \right)' \cdot H\tilde{L}(\tilde{\beta}) \cdot [D\tilde{\psi}(\tilde{\beta})]^{-1}, \end{aligned}$$

where $D_{v,i}$ is used to indicate that the derivative is with respect to v even when the terms inside the derivative are expressed using $\tilde{\beta}$. We have now established the following proposition.

Proposition 27 For all $i \in [n]$, $\tilde{p} \in \tilde{\Delta}^n$, and $v = \tilde{\psi}(\tilde{\beta})$,

$$H\ell_i(v) = -\left((e_i^i - \tilde{p})' \otimes I_n \right) \cdot D[\kappa(\tilde{\psi}^{-1}(v))] + (\kappa(\tilde{\beta})) \cdot [D\tilde{\psi}(\tilde{\beta})]^{-1},$$

where $\kappa(\tilde{\beta})$ is defined in (16).

The product $\kappa(\tilde{\beta}) := -H\tilde{L}(\tilde{\beta}) [D\tilde{\psi}(\tilde{\beta})]^{-1}$ that appears in both propositions above can be interpreted as the curvature of the Bayes risk function L relative to the rate of change of the link function $\tilde{\psi}$. When the link function is the identity $\tilde{\psi}(\tilde{\beta}) = \tilde{\beta}$ (i.e. when we have a proper loss directly) the expressions for the derivative and Hessian of each ℓ_i simplify to

$$\begin{aligned} D\ell_i(\tilde{\beta}) &= (e_i^i - \tilde{p})' \cdot H\tilde{L}(\tilde{\beta}) \\ H\ell_i(\tilde{\beta}) &= \left((e_i^i - \tilde{p})' \otimes I_n \right) \cdot D[H\tilde{L}(\tilde{\beta})] - H\tilde{L}(\tilde{\beta})'. \end{aligned} \quad (17) \quad (18)$$

The form of κ as the product of $H\tilde{L}$ and $D\tilde{\psi}$ suggests another simplification.

Definition 28 The canonical link function for a loss λ with Bayes risk \underline{L} is defined via

$$\tilde{\psi}_\lambda(\tilde{\beta}) := -D\underline{L}(\tilde{\beta})'. \quad (19)$$

We will show in section 8.1 that (19) is indeed guaranteed to be a legitimate link. The term κ simplifies to $\kappa(\tilde{\beta}) = I_n$ since $D\tilde{\psi}(\tilde{\beta}) = -D(D\underline{L}(\tilde{\beta})') = -H\underline{L}(\tilde{\beta})$. For this choice of link function, the first and second derivatives become considerably simpler.

Proposition 29 If $\lambda: \Delta^n \rightarrow \mathbb{R}_+^n$ is a proper loss and $\tilde{\psi}_\lambda$ is its associated canonical link then, for all $i \in [n]$, $\tilde{p} \in \tilde{\Delta}^n$, and $v = \tilde{\psi}_\lambda(\tilde{\beta})$, the composite loss $\ell = \lambda \circ \tilde{\psi}$ satisfies

$$\begin{aligned} D\ell_i(v) &= (e_i^i - \tilde{p}) \\ H\ell_i(v) &= [H\underline{L}(\tilde{\beta})]^{-1}. \end{aligned} \quad (20) \quad (21)$$

The simplified form of the Hessian above is established by noting that since $\kappa(\tilde{\beta}) = I_n$ we have $D[\kappa(\tilde{\psi}^{-1}(v))] = 0$ for all $v \in \mathcal{Y}$ in Proposition 27.

The above propositions hold for any number of classes n . It is instructive (both here and later in the paper) to examine the binary case where $n = 2$. In this case, Proposition 26 and Proposition 27 reduce to

$$\ell_1'(v) = -(1 - \tilde{p})\kappa(\tilde{\beta}) \quad ; \quad \ell_2'(v) = \tilde{p}\kappa(\tilde{\beta}) \quad (22)$$

$$\ell_1''(v) = \frac{-(1 - \tilde{p})\kappa(\tilde{\beta}) + \kappa(\tilde{\beta})}{\tilde{\psi}'(\tilde{\beta})} \quad (23)$$

$$\ell_2''(v) = \frac{\tilde{p}\kappa(\tilde{\beta}) + \kappa(\tilde{\beta})}{\tilde{\psi}'(\tilde{\beta})} \quad (24)$$

where $\kappa(\tilde{\beta}) = -\frac{\underline{L}''(\tilde{\beta})}{\tilde{\psi}'(\tilde{\beta})} \geq 0$ and so $\frac{d}{dv} \kappa(\tilde{\psi}^{-1}(v)) = \frac{\kappa'(\tilde{\beta})}{\tilde{\psi}'(\tilde{\beta})}$.

6.4.2 CONDITIONS FOR CONVEXITY OF MULTICLASS COMPOSITE PROPER LOSSES

We will now consider when multiclass proper losses are convex, and give a characterisation in terms of the corresponding Bayes risk which as we have seen is the natural way to parametrise a loss. The results below are the multiclass generalisation of the characterisation of convexity of binary composite losses (Reid and Williamson, 2010). In fact we obtain more general results

even in the binary case because here we consider *strongly* convex losses. We will also show how any non-convex proper loss can be made convex by suitable choice of a link function (the canonical link)³.

For a convex set $C \subseteq \mathbb{R}^n$, a loss $\ell: C \rightarrow \mathbb{R}_+^n$ is said to be *convex* if for all $p \in \Delta^n$, the map $C \ni v \mapsto L(p, v) = p' \cdot \ell(v)$ is convex. That is, a loss is convex if, under any distribution p over outcomes $i \in [n]$, the expected loss $\mathbb{E}_{p \sim p}[\ell_i(v)]$ is convex in v . It is easy to see that ℓ is convex if and only if $\ell_i: C \rightarrow \mathbb{R}_+$ is convex for all $i \in [n]$. (The ‘‘if’’ part follows since a sum of convex functions is convex; the ‘‘only if’’ follows by considering $p = e_i$, for $i \in [n]$.)

Definition 30 Suppose $C \subseteq \mathbb{R}^n$ is convex. A function $f: C \rightarrow \mathbb{R}$ is **strongly convex on C with modulus $c \geq 0$** if for all $x, x_0 \in C$, $\forall \alpha \in (0, 1)$,

$$f(\alpha x + (1 - \alpha)x_0) \leq \alpha f(x) + (1 - \alpha)f(x_0) - \frac{1}{2}c\alpha(1 - \alpha)\|x - x_0\|^2.$$

When $c = 0$ in the above definition, f is convex. The function f is strongly convex on C with modulus c if and only if $x \mapsto f(x) - \frac{c}{2}\|x\|^2$ is convex on C (Hiriart-Urruty and Lemaréchal, 2001, page 73). Therefore, the maps $v \mapsto \ell_i(v)$ are c -strongly convex if and only if $\text{H}\ell_i(v) \succeq cI_n$. By applying Proposition 27 we obtain the following characterisation of the c -strong convexity of the loss ℓ .

Proposition 31 A proper composite loss $\ell = \lambda \circ \psi^{-1}$ is strongly convex with modulus $c \geq 0$ if and only if for all $\tilde{p} \in \tilde{\Delta}^n$ and for all $i \in [n]$

$$\left((e_i^{\tilde{p}} - \tilde{p}) \otimes I_n \right) \cdot \text{D} \left(\kappa(\tilde{\psi}^{-1}(v)) \right) \preceq \kappa(\tilde{p}) \cdot [\text{D}\tilde{\psi}(\tilde{p})]^{-1} - cI_n. \quad (25)$$

We now consider the implications of Proposition 31 in two special cases: in the multiclass case with canonical link, and in the binary case with the identity link.

Recall that the canonical link $\tilde{\psi}$ is chosen so that $\tilde{\psi}(\tilde{p}) = -\text{D}\tilde{L}(\tilde{p})'$. This simplifies $\kappa(\tilde{p})$ to the identity matrix I_n so $\text{D}\kappa(\tilde{p}) = 0$. In this case the above proposition reduces to the following corollary.

Corollary 32 If $\ell = \lambda \circ \psi^{-1}$ is defined so that $\tilde{\psi} = -\text{D}\tilde{L}$ then each map $v \mapsto \ell_i(v)$ is c -strongly convex if and only if $[-\text{H}\tilde{L}(\tilde{p})]^{-1} \succeq cI_n$, or equivalently $-\text{H}\tilde{L}(\tilde{p}) \preceq \frac{1}{c}I_n$.

An immediate consequence of this result is obtained by observing that the definiteness constraint is always met when $c = 0$ since \tilde{L} is always a concave function. Thus, using a canonical link guarantees a proper composite loss is convex.

There is an upper definiteness condition analogous to that for strong convexity that has implications for rates of convergence in numerical optimisation. Boyd and Vandenberghe (2004, §9.1.2) show that if a twice differentiable function $f: \mathcal{B} \rightarrow \mathbb{R}$ satisfies

$$MI \succeq \text{H}f(x) \succeq mI$$

³ There are problems associated with the domain of definition of such link functions than need to be dealt with (Kamalanarayanan et al., 2015).

for all $x \in \mathcal{B} \subset \mathbb{R}^n$ then the value $\frac{M}{m}$ is an upper bound on the *condition number* of $\text{H}f$, that is, the ratio of maximum to minimum eigenvalue of $\text{H}f$. This value measures the eccentricity of the sublevel sets of f and controls the rate at which optima of f are approached.

Applying this result to the Hessian of a composite loss ℓ with a canonical link shows that the condition number bound is controlled by the Hessian of the Bayes risk of ℓ . Specifically, if the condition number is to be no more than M/m then $\frac{M}{m} \succeq -\text{H}\tilde{L}(\tilde{p}) \succeq \frac{1}{m}$ for all \tilde{p} . In the case that $M = m$ and the condition number is 1, the only Hessian that satisfies these conditions is $\text{H}\tilde{L}(\tilde{p}) = -I_n$ which is easily shown to be the Bayes risk for square loss. Thus, square loss is the only canonical composite loss for which a condition number of 1 is possible.

In the binary case, when $n = 2$, (23) and (24) and the positivity of $\tilde{\psi}'$ simplify (25) to the two conditions:

$$\left. \begin{aligned} (1 - \tilde{p})\kappa'(\tilde{p}) &\leq \kappa(\tilde{p}) - c\tilde{\psi}'(\tilde{p}) \\ -\tilde{p}\kappa'(\tilde{p}) &\leq \kappa(\tilde{p}) - c\tilde{\psi}'(\tilde{p}) \end{aligned} \right\}, \quad \forall \tilde{p} \in (0, 1).$$

Further assuming that $\tilde{\psi}$ is the identity link ($\tilde{\psi}(v) = v$) and letting $w(\tilde{p}) := -\tilde{L}''(\tilde{p})$ gives

$$\left. \begin{aligned} w'(\tilde{p}) &\leq \frac{1}{1-\tilde{p}}(w(\tilde{p}) - c) \\ w'(\tilde{p}) &\geq \frac{1}{\tilde{p}}(w(\tilde{p}) - c) \end{aligned} \right\}, \quad \forall \tilde{p} \in (0, 1) \\ \Leftrightarrow -\frac{1}{\tilde{p}} &\leq \frac{w'(\tilde{p})}{w(\tilde{p}) - c} \leq \frac{1}{1-\tilde{p}}, \quad \forall \tilde{p} \in (0, 1). \quad (26)$$

The last equivalence is achieved by dividing through by $w(\tilde{p}) - c$ which must necessarily be positive since if it were not the final pair of inequalities would imply $-\frac{1}{\tilde{p}} \geq \frac{1}{1-\tilde{p}}$, a contradiction given that $\tilde{p} \in [0, 1]$. Note that (26) reduces to (Reid and Williamson, 2010, Corollary 26) for $c = 0$.

Observe that if $g(\tilde{p}) := \log(w(\tilde{p}) - c)$ then $g'(\tilde{p}) = \frac{w'(\tilde{p})}{w(\tilde{p}) - c}$ is the middle term in (26). This allows a simplification of the inequality. Specifically, if we assume $w(\frac{1}{2}) = 1$ then

$$\left. \begin{aligned} -\frac{1}{\tilde{p}} &\leq g'(\tilde{p}) \leq \frac{1}{1-\tilde{p}}, \quad \forall \tilde{p} \in (0, 1) \\ \Rightarrow \int_{\frac{1}{2}}^q \frac{1}{\tilde{p}} d\tilde{p} &\leq \int_{\frac{1}{2}}^q g'(\tilde{p}) d\tilde{p} \leq \int_{\frac{1}{2}}^q \frac{1}{1-\tilde{p}} d\tilde{p}, \quad \forall q \in (0, 1) \\ \Leftrightarrow -\log(q) - \log(2) &\leq g(q) - \log(1 - c) \\ &\leq -\log(2) - \log(1 - q), \quad \forall q \in (0, 1) \\ \Leftrightarrow \frac{1}{2q} &\leq e^{g(q) - \log(1 - c)} \leq \frac{1}{2(1 - q)}, \quad \forall q \in (0, 1) \end{aligned} \right\} \quad (27)$$

which gives the following proposition purely in terms of $w(\tilde{p})$, rather than $w(\tilde{p})$ and its derivative.

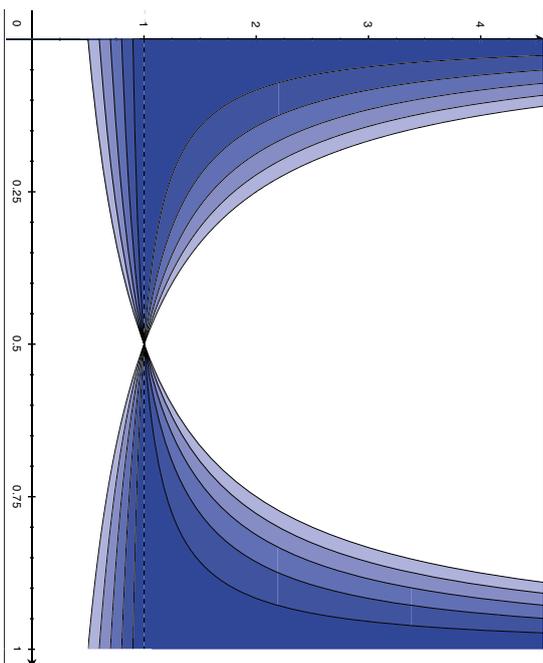


Figure 7: Graph of $w(\bar{p}) = -\underline{\ell}''(\bar{p})$ as a function of \bar{p} necessary for a suitably normalised binary proper loss to be strongly convex with modulus $c \in \{0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1\}$. The regions R_c are nested by subsethood so that $R_0 \supset R_{1/5} \supset R_{2/5} \supset R_{3/5} \supset R_{4/5} \supset R_1$, where R_1 is simply the dotted line (containing only the function $w(c) = 1$, $\forall c \in [0, 1]$, which is the weight function corresponding to squared loss). The palest shaded region corresponds to R_0 , the allowable range of $w(c)$ necessary for the corresponding proper loss to be convex, and the darkest corresponds to $R_{4/5}$.

Proposition 33 Let $w(\bar{p}) = -\underline{\ell}''(\bar{p})$ and assume $w(1/2) = 1$. A proper binary loss $\ell: \Delta^2 \rightarrow \mathbb{R}_+^2$ is strongly convex with modulus $c \in [0, 1]$ only if

$$\frac{1}{2\bar{p}} \leq \frac{w(\bar{p}) - c}{1 - c} \leq \frac{1}{2(1 - \bar{p})}, \quad \forall \bar{p} \in (0, 1), \quad (28)$$

where \leq denotes \leq for $\bar{p} \geq \frac{1}{2}$ and denotes \geq for $\bar{p} \leq \frac{1}{2}$.

When $c = 0$ (corresponding to ℓ being convex) this is equivalent to an expression by Reid and Williamson (2010, Equation 31), where it was incorrectly claimed this condition was also sufficient. Inequation 28 is illustrated in Figure 7.

The above proposition only gives a necessary condition for strong convexity. (In addition to w belonging to the specified region, $w(\bar{p})$ also needs to be suitably controlled). A sufficient condition is useful for designing strongly convex proper losses. Observe that if

$$w(\bar{p}) = \exp\left(\int_{1/2}^{\bar{p}} u(t)dt + K\right) + c$$

where $u: [0, 1] \rightarrow \mathbb{R}$ and $K, c \in \mathbb{R}$, then $\frac{\partial}{\partial \bar{p}} \log(w(\bar{p}) - c) = u(\bar{p})$. We require $w(1/2) = 1$ and so $\exp\left(\int_{1/2}^{1/2} u(t)dt + K\right) + c = 1$ and so $e^K = 1 - c$ and

$$w(\bar{p}) = (1 - c) \exp\left(\int_{1/2}^{\bar{p}} u(t)dt\right) + c \quad (29)$$

satisfies (26) if

$$-\frac{1}{\bar{p}} \leq u(\bar{p}) \leq \frac{1}{1 - \bar{p}}, \quad \forall \bar{p} \in (0, 1), \quad (30)$$

and hence the loss with weight function w is strongly convex with modulus c . Thus by choosing u to satisfy (30) and constructing w via (29) one can design strongly convex proper binary losses.

One can ask whether equation (25) can be simplified in the $n > 2$ case by using a matrix version of the logarithmic derivative trick in a manner similar to that used above when $n = 2$. Such a result does exist (Horn and Johnson, 1991, Section 6.6, 19) but it requires that $(\underline{H}\underline{\ell}(\bar{p}))^{-1}$ and $D(\underline{H}\underline{\ell}(\bar{p}))$ commute for all $\bar{p} \in \bar{\Delta}^n$, which is not generally the case.

7. Integral Representations of Proper Losses

Binary proper losses have an attractive integral representation that provides substantial insight and is a useful tool for both designing losses and understanding the implications of different choices of loss. Specifically, there exists a family of “extremal” loss functions (cost-weighted generalisations of the 0-1 loss) parametrised by $c \in [0, 1]$ and defined for all $\eta \in [0, 1]$ by $\underline{\ell}_{-1}(\eta) := c\|\eta \geq c\|$ and $\underline{\ell}_1 := (1 - c)\|\eta < c\|$. As shown by Buja et al. (2005) and Reid and Williamson (2011), given these extremal functions, any proper binary loss ℓ can be expressed as the weighted integral

$$\ell = \int_0^1 \rho^* w(c) dc + \text{constant}$$

with “weight function” $w(c) = -\underline{\ell}''(c)$. This representation is a special case of a representation from Choquet theory (Phelps, 2001; Simon, 2011) which characterises when every point in some set can be expressed as a weighted combination of the “extremal points” of the set. Although there is such a representation when $n > 2$, the difficulty is that the set of extremal points is much larger and this rules out the existence of a nice small set of “primitive” proper losses when $n > 2$, and consequently rules out an easy-to-work-with weight function parameterizing all possible multiclass losses in a manner analogous to the binary case. The rest of this section makes this statement precise.

A convex cone \mathcal{K} is a set of points closed under positive linear combinations. That is, $\mathcal{K} = \alpha\mathcal{K} + \beta\mathcal{K}$ for any $\alpha, \beta \geq 0$. A point $f \in \mathcal{K}$ is extremal if $f = \frac{1}{2}(g + h)$ for $g, h \in \mathcal{K}$

implies $\exists \alpha \in \mathbb{R}_+$ such that $g = \alpha f$. That is, f cannot be represented as a non-trivial combination of other points in \mathcal{X} . The set of extremal points for \mathcal{X} will be denoted $\text{ex } \mathcal{X}$. Suppose U is a bounded closed convex set in \mathbb{R}^d , and $\mathcal{X}_b(U)$ is the set of convex functions on U bounded by 1, then $\mathcal{X}_b(U)$ is compact with respect to the topology of uniform convergence. Bronshtein (1978, Theorem 2.2) showed that the extremal points of the convex cone $\mathcal{X}(U) = \{\alpha f + \beta g : f, g \in \mathcal{X}_b(U), \alpha, \beta \geq 0\}$ are dense (w.r.t. the topology of uniform convergence) in $\mathcal{X}(U)$ when $d > 1$. This means for any function $f \in \mathcal{X}(U)$ there is a sequence of functions (g^i) such that for all i $g^i \in \text{ex } \mathcal{X}(U)$ and $\lim_{i \rightarrow \infty} \|f - g^i\|_\infty = 0$, where $\|f\|_\infty := \sup_{u \in U} |f(u)|$. We use this result to show that the set of extremal Bayes risks is dense in the set of Bayes risks when $n > 2$.

In order to simplify our analysis, we restrict attention to fair proper losses. A loss is *fair* if each partial loss is zero on its corresponding vertex of the simplex ($\ell_i(e_i) = 0, \forall i \in [n]$). A proper loss is *fair* if and only if its Bayes risk is zero at each vertex of the simplex (in this case the Bayes risk is also called fair). One does not lose generality by studying fair proper losses since any proper loss is a sum of a fair proper loss and a constant vector.

The set of fair proper losses defined on Δ^n form a closed convex cone, denoted \mathcal{L}_n . The set of concave functions which are zero on all the vertices of the simplex Δ^n is denoted \mathcal{F}_n and is also a closed convex cone.

Proposition 34 *Suppose $n > 2$. Then for any fair proper loss $\ell \in \mathcal{L}_n$ there exists a sequence (ℓ^i) of extremal fair proper losses $(\ell^i \in \text{ex } \mathcal{L}_n)$ which converges almost everywhere to ℓ .*

The implication of this proposition is that the set of extremal multiclass proper losses is very large. Some intuition can be gleaned from Figure 8 from which it is apparent that there is a qualitative difference between the complexity of the set of extremal concave functions in one dimension (corresponding to $n = 2$) and higher dimensions ($n > 2$). The proof of Proposition 34 requires the following lemma which relies upon the correspondence between a proper loss and its Bayes risk (Proposition 8) and the fact that two continuous functions equal almost everywhere are equal everywhere.

Lemma 35 *If $\ell \in \text{ex } \mathcal{L}_n$ then its corresponding Bayes risk \underline{L} is extremal in \mathcal{F}_n . Conversely, if $\underline{L} \in \text{ex } \mathcal{F}_n$ then all the proper losses ℓ with Bayes risk equal to \underline{L} are extremal in \mathcal{L}_n .*

Proof We suppose that $\ell \in \text{ex } \mathcal{L}_n$ and denote its Bayes risk by $\underline{L}(p) = p' \cdot \ell(p)$. Let $\underline{F}, \underline{G} \in \mathcal{F}_n$ so that $\underline{L} = \frac{1}{2}(\underline{F} + \underline{G})$. Suppose f and g are proper losses whose Bayes risks are respectively equal to \underline{F} and \underline{G} , then $\forall p \in \Delta^n$ and almost everywhere in q (more precisely where $\underline{L}, \underline{F}$ and \underline{G} are differentiable), $\underline{L}(p, q) = \frac{1}{2}(\underline{G}(p, q) + \underline{F}(p, q))$. Then $\ell = \frac{1}{2}(g + f)$ almost everywhere, so there exists α such as $g = \alpha \ell$ almost everywhere, hence $\underline{G} = \alpha \underline{L}$ almost everywhere and then everywhere by continuity. So \underline{L} is extremal in \mathcal{F}_n .

Now suppose that the concave function \underline{L} is extremal and let ℓ be a proper loss whose Bayes risk is \underline{L} . Then $\underline{L}(p, q) = p' \cdot \ell(q) = \underline{L}(q) + (p - q)' \cdot A(q)$ where $A(q) \in \partial \underline{L}(q)$. Suppose that there exist $f, g \in \mathcal{L}_n$ so that $\ell = \frac{1}{2}(f + g)$ almost everywhere, and have associated Bayes risks \underline{F} and \underline{G} , respectively. Then $\underline{L}(p) = p' \cdot \ell(p) = p' \cdot \frac{1}{2}(f(p) + g(p)) = \frac{1}{2}(\underline{F} + \underline{G})$ almost everywhere so $\underline{L} = \frac{1}{2}(\underline{F} + \underline{G})$ everywhere by continuity. Since \underline{L} is extremal we must have $\underline{F} = \alpha \underline{L}$ and so $f = \alpha \ell$ where \underline{L} is differentiable (and so almost everywhere). Thus ℓ is extremal in \mathcal{L}_n . ■

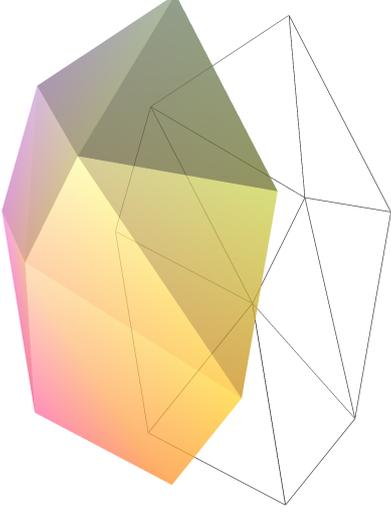


Figure 8: Complexity of extremal concave functions in two dimensions (corresponds to $n = 3$). The figure shows the graph of an extremal concave function in two dimensions. The lines below indicate where the slope changes. The pattern of these lines can be made arbitrarily complex. This illustrates the fact (Proposition 34) that the set of extremal concave functions is very large.

We also need a correspondence between the uniform convergence of a sequence of Bayes risk functions and the convergence of their associated proper losses.

Lemma 36 *Suppose $\underline{L}, \underline{L}^i \in \mathcal{F}_n$ for $i \in \mathbb{N}$ and suppose ℓ and $\ell^i, i \in \mathbb{N}$ are associated proper losses. Then (\underline{L}^i) converges uniformly to \underline{L} if and only if (ℓ^i) converges almost everywhere to ℓ .*

Proof We require two facts from convex analysis: confer (Hiriart-Urruty and Lemaréchal, 2001, Theorems B.3.1.4 and D.6.2.7). If a sequence (f^i) of convex functions f^i converges pointwise to f then: 1) the sequence converges uniformly on any compact domain; and 2) $\forall \varepsilon > 0, \partial f^i(x) \subset \partial f(x) + \mathcal{B}(0, \varepsilon)$ for i large enough. Then the reverse implication of the lemma is a direct consequence of the first result and the forward implication is obtained by considering the set $\{x : \forall n, \underline{L}^i$ and \underline{L} are differentiable at $x\}$ which is of measure 1. ■

Proof (Proposition 34) When $n > 2$ the simplex Δ^n is isomorphic to a subset of \mathbb{R}^d for $d > 1$. Since \mathcal{F}_n is a convex cone associated with the set of bounded concave functions (i.e., the fair Bayes risks), (Bronshtein, 1978, Theorem 2.2) guarantees (with an appropriate change from concavity to convexity) that $\text{ex } \mathcal{F}_n$ is dense in \mathcal{F}_n w.r.t. the topology of uniform convergence. Therefore, if $\ell \in \mathcal{L}_n$ there exists a sequence (f^i) with $f^i \in \text{ex } \mathcal{F}_n$ which converges uniformly to the Bayes risk \underline{L} of ℓ and so by Lemma 36 there is a corresponding sequence (ℓ^i) of fair proper

losses that converges almost everywhere to ℓ . Lemma 35 guarantees that each ℓ^i is extremal in \mathcal{L}_ρ since each $\ell^i \in \text{ex } \mathcal{F}_n$, and so we have shown there exists a sequence (θ^j) with $\theta^j \in \text{ex } \mathcal{L}_\rho$ which converges to an ℓ which was arbitrary. ■

8. Tools for Designing Losses

In this section we show how the results developed above could be used to design losses for particular purposes. There is no question about how this should be done “in principle” (Berger, 1985, Section 2.4). And even when not made explicit at the outset, *all* inference ultimately has an *implicit* loss function that captures what matters to the end user, even if the original purpose was to merely “gather information”, simply because in the end the “information” is acted upon (DeGroot, 1962). Until now, the lack of convenient and canonical parametrisations of multi-class loss functions has made the comparison of different loss functions, and their tuning for specific applications, difficult.

In subsection 8.1, we show how to construct a parametric family of valid link functions from a finite number of “base” links by effectively taking their convex combination. By composing each link with a fixed proper loss, this immediately allows for the specification of a family of losses with a fixed Bayes risk. This construction enables the creation of losses with a range of optimisation characteristics (ϵ , g , convexity, robustness) but a common statistical basis (ℓ, ϵ) , the same Bayes risk.

In subsection 8.2 we show how it is possible to build losses by building them up from constraints on their Bayes risk curves on the edges of the simplex. This allows a loss to be constructed by effectively specifying its behaviour on pairs of outcomes. We show how this observation can be used to create piecewise linear, proper losses for cost-sensitive misclassification.

Finally (subsection 8.3) we observe how link functions are in fact themselves very similar to loss functions, and (subsection 8.4) we present some examples of proper composite losses from the literature (where they were not expressed in the proper composite parametrisation)

8.1 Families of Losses with Fixed Bayes Risk

The theory developed above suggests that each choice of proper loss λ and link function ψ results in an overall loss function with properties (ϵ , g , convexity) that depend entirely on their relationship to each other. Given these two “knobs” for parameterising a loss function, we can begin to ask what kind of practical trade-offs are involved when selecting a composite loss as a surrogate loss for a particular problem.

We now propose a simple scheme for constructing families of losses with the same Bayes risk. This is achieved by fixing a choice of proper loss λ and creating a parameterised family (described below) of link functions ψ_α for parameters $\alpha \in A$. Since the Bayes risk is entirely determined by λ any composite loss $\lambda \circ \psi_\alpha^{-1}$ for $\alpha \in A$ will have Bayes risk $\underline{L}(\rho) = \rho^* \cdot \lambda(\rho)$. Thus, we are able to examine the effect different choices of composite loss can have on a problem *without changing the essential underlying problem*.⁴

4. Of course, this argument only holds in a point-wise analysis. That is, where choices for estimates $\hat{\mu}(\delta)$ can be made independently. Once a restricted hypothesis class for the functions ρ is introduced the choice of link can

In order to construct a parametric family of links we first choose some set of inverse link functions $\mathcal{F} = \{\psi_1^{-1}, \dots, \psi_B^{-1}\}$ with a common domain, that is, $\psi_b^{-1}: \mathcal{Y} \rightarrow \Delta^n$ for a convex n and \mathcal{Y} . This collection will be called the *basis set* of link functions. We then take the convex hull of \mathcal{F} to form a set of inverse link functions Ψ^{-1} . Each $\psi^{-1} \in \Psi^{-1}$ is then identified with the unique $\alpha \in A = \Delta^B$ such that $\sum_{b=1}^B \alpha_b \psi_b^{-1} = \psi^{-1}$. For this construction to be valid, it is necessary to show that every such $\psi^{-1} \in \Psi^{-1}$ is indeed an inverse link function, that is, it is invertible.

The following proposition shows that it suffices to assume that all of the basis functions are *strictly monotone* (see Equation 2).

Proposition 37 *Every function ψ^{-1} in the set $\Psi^{-1} = \text{co}(\mathcal{F})$ is invertible whenever each basis function in \mathcal{F} is strictly monotone.*

This result is a consequence of: 1) strict monotonicity being preserved under convex combination; and 2) strict monotonicity implies invertibility. The first claim is established by considering strictly monotone f and g and some $\alpha \in [0, 1]$ and noting that if $h = \alpha f + (1 - \alpha)g$ then $h(u) - h(v) = \alpha(f(u) - f(v)) + (1 - \alpha)(g(u) - g(v))$. A strictly monotone function f that is not invertible is impossible since if we have $(f(u) - f(v))(u - v) > 0$ for all u, v then a $u \neq v$ such that $f(u) = f(v)$ would lead to a contradiction.

Strictly monotone basis functions are easily obtained via canonical links for strictly proper losses. By definition, a canonical link satisfies $\psi = -D_{\underline{L}}$ for some Bayes risk function. Strict properness guarantees \underline{L} is strictly concave (van Erven et al., 2012b, Lemma 1). Kachurovskii’s theorem (Hiriart-Urruty and Lemaréchal, 2001, Theorem 4.1.4) states that the derivative of a function is (strictly) monotone if and only if the function is (strictly) convex. Since $f(f^{-1}(u)) = f(f^{-1}(v))(f^{-1}(u) - f^{-1}(v)) = (u - v)(f^{-1}(u) - f^{-1}(v))$ we see that strictly monotone functions have strictly monotone inverses and we have established the following proposition.

Proposition 38 *If λ is a strictly proper loss then its canonical link $\psi_\lambda = -D_{\underline{L}}$ has a strictly monotone inverse.*

This result means that a set of basis links can be defined via a choice of strictly concave Bayes risk functions. As an example, the class of Fisher-consistent margin losses proposed by Zou et al. (2008) provides a flexible starting point for designing sets of link functions as described above. They give explicit formulae for the inverse link for a composite loss defined by a choice of convex function $\phi: \mathbb{R} \rightarrow \mathbb{R}$. Specifically, if the loss for predicting $v \in \mathcal{Y} = \{v \in \mathbb{R}^n: \sum_i v_i = 0\}$ is given by $\ell(v) = \phi(v)$ then its inverse link is $\psi_\phi^{-1}(v) = \frac{1}{\sum_{i=1}^n \phi(v_i)}$ where $Z_\phi(v)$ normalises the vector to lie in Δ^n . Each choice of strictly convex ϕ gives a valid inverse link which can be used as a basis function.

8.2 Piecewise Linear Multiclass Losses

We now build a family of conditional Bayes risks. Suppose we are given $\frac{n(n-1)}{2}$ concave functions $\{\underline{L}^{i,j}: \Delta^2 \rightarrow \mathbb{R}\}_{1 \leq i < j \leq n}$ on Δ^2 , and we want to build a concave function \underline{L} on Δ^n affect the minimal achievable risk. The interaction between the hypothesis class and the loss function is complex (van Erven et al., 2015).

which is equal to one of the given functions on each edge of the simplex ($\forall 1 \leq i_1 < i_2 \leq n$, $\underline{L}(0, \dots, 0, p_{i_1}, 0, \dots, 0, p_{i_2}, 0, \dots, 0) = \underline{L}^{i_1, i_2}(p_{i_1}, p_{i_2})$). This is equivalent to choosing a binary loss function, knowing that the observation is in the class i_1 or i_2 . The result below gives one possible construction. (There exists an infinite number of solutions—one can simply add any concave function equal to zero in each edge).

Lemma 39 Suppose we have a family of concave functions $\{\underline{L}^{i_1, i_2} : \Delta^2 \rightarrow \mathbb{R}\}_{1 \leq i_1 < i_2 \leq n}$, then

$$\underline{L} : \Delta^n \ni p \mapsto \underline{L}((p_1, \dots, p_n)) = \sum_{1 \leq i_1 < i_2 \leq n} (p_{i_1} + p_{i_2}) \underline{L}^{i_1, i_2} \left(\left(\frac{p_{i_1}}{p_{i_1} + p_{i_2}}, \frac{p_{i_2}}{p_{i_1} + p_{i_2}} \right) \right)$$

is concave and $\forall 1 \leq i_1 < i_2 \leq n$, $\underline{L}((0, \dots, 0, p_{i_1}, 0, \dots, 0, p_{i_2}, 0, \dots, 0)) = \underline{L}^{i_1, i_2}((p_{i_1}, p_{i_2}))$.

Proof In order to show that \underline{L} is concave it suffices to show that for $g : \Delta^2 \rightarrow \mathbb{R}$ concave, $f : p \in \Delta^n \rightarrow f(p) = (p_1 + p_2)g\left(\frac{p_1}{p_1 + p_2}, \frac{p_2}{p_1 + p_2}\right)$ is concave, since a sum of concave functions is concave.

Let $\gamma \in [0, 1]$, $p, q \in \Delta^n$. Since g is concave, $\forall \alpha \in [0, 1]$, $\forall p, q \in \Delta^2$, $g\left(\frac{\alpha}{\alpha + (1-\alpha)}p + \frac{1-\alpha}{\alpha + (1-\alpha)}q\right) \geq \alpha g\left(\frac{p}{\alpha + (1-\alpha)}\right) + (1-\alpha)g\left(\frac{q}{\alpha + (1-\alpha)}\right)$. Then with $\alpha = \frac{\gamma(p_1 + p_2)}{\gamma(p_1 + p_2) + (1-\gamma)(q_1 + q_2)}$, we get $f(\gamma p + (1-\gamma)q) \geq \gamma f(p) + (1-\gamma)f(q)$.

Moreover, $\underline{L}((0, \dots, 0, p_{i_1}, 0, \dots, 0, p_{i_2}, 0, \dots, 0)) = \sum_{i_1, i_2} \underline{L}^{i_1, i_2}((p_{i_1} \times 0 + p_{i_2} \times 0))$

$+ (p_{i_1} + p_{i_2}) \underline{L}^{i_1, i_2} \left(\left(\frac{p_{i_1}}{p_{i_1} + p_{i_2}}, \frac{p_{i_2}}{p_{i_1} + p_{i_2}} \right) \right) = \underline{L}^{i_1, i_2}((p_{i_1}, p_{i_2}))$, ($p \in \Delta^n$, so $p_{i_1} + p_{i_2} = 1$).

Using this family of Bayes risks, one can build a family of proper losses. ■

Lemma 40 Suppose we have a family of binary proper losses $\ell^{i_1, i_2} : \Delta^2 \rightarrow \mathbb{R}^+$. Then

$$\ell : \Delta^n \ni p \mapsto \ell(p) = \left(\sum_{j=1}^{j-1} \ell^{i, j} \left(\frac{p_i}{p_i + p_j} \right) + \sum_{i=j+1}^n \ell^{i, j} \left(\frac{p_j}{p_i + p_j} \right) \right)_{j=1}^n \in \mathbb{R}^+$$

is a proper n -class loss such that

$$\ell_i((0, \dots, 0, p_{i_1}, 0, \dots, 0, p_{i_2}, 0, \dots, 0)) = \begin{cases} \ell^{i_1, i_2}(p_{i_1}) & i = i_1 \\ \ell^{i_1, i_2}(p_{i_2}) & i = i_2 \\ 0 & \text{otherwise} \end{cases}$$

Proof Use the correspondence between Bayes risk and proper losses and Lemma 39. ■

Observe that it is much easier to work at first with the Bayes risk and then using the correspondence between Bayes risks and proper losses.

We have already seen (Section 7) that it is not possible to parametrise all extremal concave functions in a tractable manner. However, for the sake of offering a range of knobs to the designer to design losses, it could often suffice to use a subset of extremal losses. These will all have polyhedral forms. A convex polytope is a compact convex intersection of a finite set of half-spaces and is therefore the convex hull of its vertices. Let $\{a_i\}$ be a finite family of affine functions defined on Δ^n . Now define the convex polyhedral function f by $f(x) := \max_i a_i(x)$. The set $K := \{P_i = \{x \in \Delta^n : f(x) = a_i(x)\}\}$ is a covering of Δ^n by polytopes. Bronshtein (1978, Theorem 2.1) shows that for f , P_i and K so defined, f is extremal if the following two conditions

are satisfied: 1) for all polytopes P_i in K and for every face F of P_i , $F \cap \Delta^n \neq \emptyset$ implies F has a vertex in Δ^n ; 2) every vertex of P_i in Δ^n belongs to n distinct polytopes of K . The set of all such f is dense in $\mathcal{K}(U)$.

Using this result it is straightforward to exhibit some sets of extremal fair Bayes risks $\{\underline{L}_c(p) : c \in \Delta^n\}$. Two examples are when

$$\underline{L}_c(p) = \sum_{i=1}^n \frac{p_i}{c_i} \prod_{j \neq i} \mathbb{1}_{\left\{ \frac{p_j}{c_j} \leq \frac{p_i}{c_i} \right\}}$$

or

$$\underline{L}_c(p) = \bigwedge_{i \in [n]} \frac{1-p_i}{c_i}.$$

Any convex combination of either of these families will be the Bayes risk of a proper fair multiclass loss. Thus the convex combination of the elementary losses induced by such $\underline{L}_c(p)$ will also be proper fair multiclass losses.

8.3 Parametrisation of Composite Losses

A composite loss $\ell = \lambda \circ \psi^{-1} : \mathcal{Y} \rightarrow \mathbb{R}^+$ is directly parametrised by the proper loss $\lambda : \Delta^n \rightarrow \mathbb{R}^+$ and the invertible link $\psi : \Delta^n \rightarrow \mathbb{R}^n$. However we have seen (Section 4) that proper losses λ are more nicely parametrised by their concave conditional Bayes risk $\Delta : \Delta^n \rightarrow \mathbb{R}$, which being scalar valued, are simpler objects to work with than λ . Although not every invertible function ψ can be written as the gradient of an analogous convex function $\Psi : \Delta^n \rightarrow \mathbb{R}$, by Kachurovskii's theorem (see Section 8.1) if for some $\Psi : \Delta^n \rightarrow \mathbb{R}$, $\psi = D\Psi$, then ψ is monotone (resp. strictly monotone) if and only if Ψ is convex (resp. strictly convex). A link ψ is a gradient if and only if $D\psi$ is symmetric (so that $H\Psi$ is symmetric) as a Hessian needs to be.

Thus if one were willing to restrict oneself to links such that $D\psi$ is symmetric, then a composite loss ℓ can be parametrised by (Δ, Ψ) , which are concave (resp. strictly convex) functions from Δ^n to \mathbb{R} . The parametrisation of ψ via Ψ allows the specification of the canonical link as that satisfying $\Psi = -\Delta$.

8.4 Examples from Related Work

In this subsection we look at some existing candidate multiclass losses from the perspective of proper composite representations. Not all such losses as the generalisation of hinge loss are so representable, a prominent example being those introduced by Crammer and Singer (2001).

Zou et al. (2008) presented multi-category losses of the form $\ell_j(f) = \phi_j(f_j)$ for f such that $\sum_j f_j = 0$ and $\phi_j(0) < 0$ and $\phi_j''(t) \geq 0$ so that we have Fisher consistency and the inverse link is $\frac{1/\phi_j(f_j)}{\sum_j 1/\phi_j(f_j)}$. As their examples of this class show, it is not always possible to write the link in closed form, even if the inverse link can be (e.g., logit loss $\phi(t) = \log(1+e^t)$).

The coherence functions of Zhang et al. (2009) are a separate class of surrogate functions that emphasise the margin of a prediction:

$$\ell_\zeta(v) = T \log \left[1 + \sum_{i \neq j} \exp(T^{-1}(1 + v_i - v_j)) \right].$$

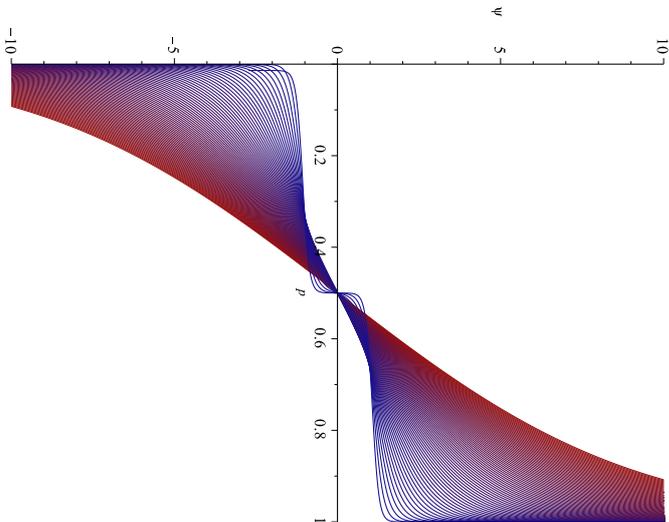


Figure 9: Illustration of the link function in the proper composite representation of the binary coherence loss for $T \in [0.1, 4]$. Blue corresponds to $T = 0.1$ and red to $T = 4$, with there being 80 equal increments of T plotted.

We will illustrate some aspects of the present paper with reference to this parametrised family of losses. For ease of calculation, we consider only $n = 2$ below, but the conclusions we draw below hold for $n > 2$ also. In the binary case, this corresponds to a parametric family of margin losses with margin function

$$\phi_T(z) := T \log \left(1 + \exp \left(\frac{1-z}{T} \right) \right),$$

and thus $\phi_T'(z) = \frac{e^{(1-z)/T}}{1+e^{(1-z)/T}}$ and one can check that $g_T(v) := -\frac{\phi_T'(v)}{\phi_T'(1-v)}$ is strictly monotone continuous and ϕ_T is monotone for all $T > 0$ and thus by Corollary 12 there is a strictly proper composite representation. Identifying $\ell_{-1}(v)$ with $\phi(-v)$ and $\ell_1(v)$ with $\phi(v)$ we can find for a

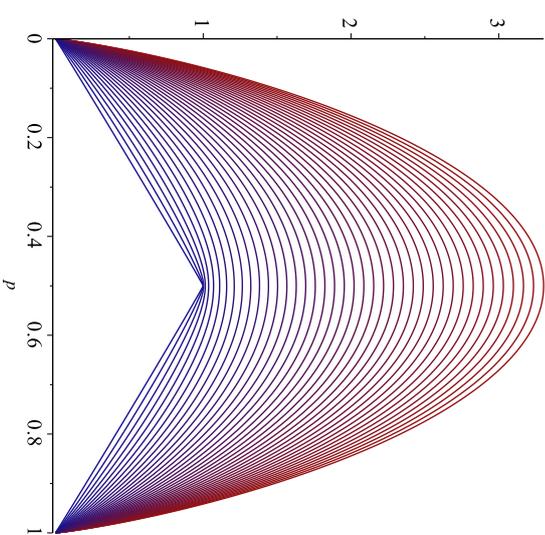


Figure 10: Illustration of the conditional Bayes risk corresponding to the binary coherence loss for $T \in [0.02, 4]$ with blue corresponding to $T = 0.02$ and red to $T = 4$.

given p the v that minimises $L(p, v)$ by solving

$$\frac{\partial}{\partial v} L(p, v) = (1-p)\phi_T'(-v) + p\phi_T'(v) = 0$$

and obtain $\frac{1-p}{p} = -g_T(v)$. Solving this for v (using Maple) we find the link function component of the proper composite representation:

$$\psi_T(p) = T \left(\ln \left(\frac{1}{2(1-p)} \left(2pe^{T^{-1}} - e^{T^{-1}} + \sqrt{4e^{2T^{-1}}p^2 - 4pe^{2T^{-1}} - 4p^2 + e^{2T^{-1}} + 4p} \right) \right) \right).$$

This is illustrated in Figure 9. Zou et al (2008, Theorem 1) effectively compute ψ_T^{-1} for general n . One can determine the proper component as

$$L_T(p) = p\phi_T(\psi_T(p)) + (1-p)\phi_T(-\psi_T(p))$$

which is plotted in Figure 10. One can glean further insight by considering the corresponding weight function $w_T^*(p) := -L_T''(p)$, which is plotted in Figure 11.

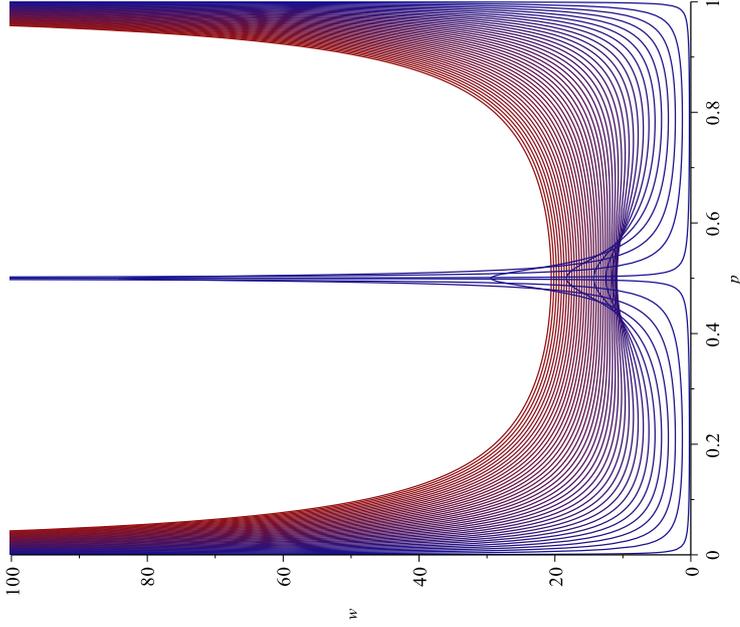


Figure 11: Weight function of the proper loss component of the proper composite representation of the binary coherence loss for $T \in [0.02, 4]$, where blue corresponds to $T = 0.02$ and red to $T = 4$.

The weight function view makes it clear how the proper component of the loss approaches 0-1 loss as $T \rightarrow 0$. (The weight function for 0/1 loss is $w(c) = \delta(c - \frac{1}{2})$; note the convergence in Figure 11 is not uniform, given the behaviour at 0 and 1.) Observe too that as well as the proper loss varying with T , the associated link also varies (in a complex way—see Figure 9). Thus not only is one varying the statistical properties of the loss (in a substantial way—when T is small, the weight is centered near $p = \frac{1}{2}$, whereas for large T , a series expansion of $w_T(p)$ shows that $w_T(p) \approx T \left(\frac{1}{p} + \frac{1}{1-p} \right) + 4$. An alternative to this class of losses, would be to fix a link, and then

vary the proper component. For a given model or hypothesis class \mathcal{F} this has the advantage that the effective hypothesis class $\{\psi^{-1} \circ f : f \in \mathcal{F}\}$ remains fixed, and only the (proper) loss varies. The (Δ, Ψ) parametrisation of section 8.3 offers a convenient way to do this.

9. Conclusions

We have systematically studied multiclass composite losses. The results of the paper (summarised below) show that this is an attractive parametrisation of multiclass losses. If one desires all predictions be strongly admissible, then there is nothing lost in using the proper composite representation. Since the link is only a reparametrisation, this means one still has the relationship between losses and divergences as described by [García-García and Williamson \(2012\)](#).

The proper composite representation leads to a desirable separation of concerns, where the inferential properties of the loss (such as its mixability) are governed by the proper loss, and the convexity (necessary for numerical optimisation) is controlled by the link function. It thus seems to be *the* best way to parametrise loss functions.

The key technical contributions of the paper are as follows.

- Relationship between prediction calibration and classification calibration, showing that the latter can be seen as a “pointwise” version of the former (Section 3);
- Characterisation of multiclass proper losses in terms of their binary restrictions (Proposition 7);
- Every (multiclass) proper loss is quasi-convex (Proposition 17);
- Characterisation of which binary margin losses have a proper composite representation (Corollary 12);
- Characterisation of when a multiclass loss has a proper composite representation and when the representation is unique (Section 5.3);
- Relationship between the proper composite representation, mixability and admissibility (Sections 6.1 and 6.2);
- Necessary conditions for strong convexity of multiclass proper losses in terms of their corresponding Bayes risks (Proposition 31);
- Canonical links always convexify proper losses, and outline how this can help in the design of losses (Proposition 32);
- The attractive (simply parametrised) integral representation for binary proper losses can *not* be extended to the multiclass case (Section 7);

These results suggest that in order to design losses for multiclass prediction problems it is helpful to use the composite representation, and design the proper part via the Bayes risk as suggested for the binary case by [Bajja et al. \(2005\)](#). The link function can be tuned to control the optimisation properties of the loss. Merely requiring the loss to be convex confounds two separate aspects of

a loss: the *shape* of $\ell(\gamma)$ which controls the predictive performance, and the *parameterization* of $\ell(\gamma)$ which affects the numerical optimisation of a loss.

There remain open questions. Perhaps the most practically important is the interaction between the loss and restricted hypothesis classes: typically one does not optimise conditionally, one optimises the full expected risk with respect to a restricted function class $\mathcal{F} \subset \mathcal{G}^{\mathcal{X}}$. The question of how knowledge of \mathcal{F} should influence the design of a loss remains open; some initial work along these lines is the notion of ‘‘stochastic mixability’’ (van Erven et al., 2012a, 2015).

Acknowledgments

The research reported here was performed whilst Elodie Vernet was a student at ENS Cachan and visiting ANU and NICTA, and was supported by the Australian Research Council and NICTA, which was funded by the Australian government through the ICT centre of excellence program. An earlier version of some of these results appeared in NIPS2011 and ICML2012. The work benefited from discussions with Jake Abernethy, Tim van Erven, Rafael Frongillo, and Dario García-García, and comments from the referees who we thank. Thanks also to Harish Guruprasad for identifying a flaw in an earlier proof of quasi-convexity of proper losses.

Appendix A. Matrix Calculus

If $A = [a_{ij}]$ is an $n \times m$ matrix, $\text{vec} A$ is the vector of columns of A stacked on top of each other. The *Kronecker product* of an $m \times n$ matrix A with a $p \times q$ matrix B is the $mp \times nq$ matrix

$$A \otimes B := \begin{pmatrix} A_{1,1}B & \cdots & A_{1,m}B \\ \vdots & \ddots & \vdots \\ A_{m,1}B & \cdots & A_{m,m}B \end{pmatrix}.$$

We use the following properties of Kronecker products (Magnus and Neudecker, 1999, Chapter 2): $(A \otimes B)(C \otimes D) = (AC \otimes BD)$ for all appropriately sized A, B, C, D , and $I_n \otimes A = A$.

If $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is differentiable at c then the *partial derivative* of f with respect to the j th coordinate at c is denoted $D_j f(c)$. The $m \times n$ matrix of partial derivatives of f is the *Jacobian* of f and denoted

$$(Df(c))_{ij} := D_j f(c) \quad \text{for } i \in [m], j \in [n].$$

If F is a matrix valued function $Df(X) := Df(\text{vec} X)$ where $f(X) := \text{vec} F(X)$.

We will require the *product rule* for matrix valued functions (Vetter, 1970; Fackler, 2005): Suppose $f : \mathbb{R}^n \rightarrow \mathbb{R}^{m \times p}$, $g : \mathbb{R}^n \rightarrow \mathbb{R}^{p \times q}$ so that $(f \times g) : \mathbb{R}^n \rightarrow \mathbb{R}^{m \times q}$. Then

$$D(f \times g)(x) = (g(x)' \otimes I_m) \cdot Df(x) + (I_q \otimes f(x)) \cdot Dg(x). \quad (31)$$

The *Hessian* at $x \in \mathcal{X} \subseteq \mathbb{R}^n$ of a real-valued function $f : \mathcal{X} \rightarrow \mathbb{R}$ is the $n \times n$ real, symmetric matrix of second derivatives at x

$$(Hf(x))_{jk} := D_{kj} f(x) = \frac{\partial^2 f}{\partial x_j \partial x_k}.$$

Note that the derivative D_{kj} is in row j , column k . It is easy to establish that the Jacobian of the transpose of the Jacobian of f is the Hessian of f . That is,

$$Hf(x) = D((Df(x))) \quad (32)$$

(Magnus and Neudecker, 1999, Chapter 10). If $\mathcal{X} \subset \mathbb{R}^n$ and $f : \mathcal{X} \rightarrow \mathbb{R}^m$ is a vector-valued function, then the Hessian of f at $x \in \mathcal{X}$ is the $mm \times n$ matrix that consists of the Hessians of the functions f_i stacked vertically:

$$Hf(x) := \begin{pmatrix} Hf_1(x) \\ \vdots \\ Hf_m(x) \end{pmatrix}.$$

Appendix B. Deferred Proofs

This appendix contains proofs of results in the main text that, due to their length or technicality, are better presented outside the flow of the main text.

B.1 Proof of Lemma 1

1. We prove this by contradiction. Suppose $p \in \Delta^n$ such that for all $i \in [n]$, $p \notin \mathcal{F}(c)$. Then

$$p \notin \mathcal{F}_1(c) \Rightarrow \exists j_2 \neq j_1 \text{ such that } \frac{p_{j_2}}{c_{j_1}} < \frac{p_{j_1}}{c_{j_2}}$$

$$p \notin \mathcal{F}_2(c) \Rightarrow \exists j_3 \neq j_2 \text{ such that } \frac{p_{j_3}}{c_{j_2}} < \frac{p_{j_2}}{c_{j_3}}$$

and hence by repeating this argument

$$p \notin \mathcal{F}_n(c) \Rightarrow \exists j_{n+1} \neq j_n \text{ such that } \frac{p_{j_{n+1}}}{c_{j_n}} < \frac{p_{j_n}}{c_{j_{n+1}}}.$$

Thus we have $n+1$ indices j_1, \dots, j_{n+1} belonging to $[n]$ and therefore one is repeated (jk and $\frac{p_k}{c_k} < \frac{p_k}{c_k}$ which is a contradiction).

2. Obvious.

3. If $p \in \cap_{i=1}^n \mathcal{F}_i(c)$, then for all $j \in [n]$, $c_j = \sum_i p_i c_i = \sum_i p_i c_i = p_j$. Thus $p = c$.

4. We prove this by contradiction. Suppose $p \neq q$ such that for all c if $p \in \mathcal{F}(c)$ then $q \in \mathcal{F}(c)$. Observe that $\forall j \in [n]$, $p \in \mathcal{F}_j(p)$, and so $q \in \cap_{j=1}^n \mathcal{F}_j(q)$, and hence $q = p$, a contradiction.

B.2 Proof of Proposition 9

Observe that

$$\partial \underline{L}(p) = \{(s', 0)' + \alpha \mathbb{1}, s \in \partial \underline{L}(p), \alpha \in \mathbb{R}\}. \quad (33)$$

Indeed $(\tilde{q} - \tilde{p})' \cdot s = (q - p)' \cdot ((s', 0)' + \alpha \mathbb{1})$.

(\Rightarrow) We first assume that L is differentiable at p . We use the following result (Hiriart-Urruty and Lemaréchal, 2001, page 203): *If f is a convex function, then $\forall \varepsilon > 0, \exists \delta > 0, \exists \tilde{\delta} > 0, y \in \mathcal{B}(x, \delta) \Rightarrow \partial f(y) \subset \partial f(x) + \mathcal{B}(0, \varepsilon)$.*

Assume $\varepsilon > 0$, then since \underline{L} is differentiable at \tilde{p} , $\exists \tilde{\delta} > 0$, such that

$$\forall \tilde{q} \in \mathcal{B}(\tilde{p}, \tilde{\delta}), \forall A(\tilde{q}) \in \partial \underline{L}(\tilde{q}), \|A(\tilde{q}) - D\underline{L}(\tilde{p})\| \leq \varepsilon. \quad (34)$$

Then there exists δ such that $q \in \mathcal{B}(p, \delta)$ implies $\tilde{p} \in \mathcal{B}(\tilde{p}, \tilde{\delta})$. Thus using (3) and (34), $\forall i \in [n]$, $\forall q \in \mathcal{B}(p, \delta)$, for $\alpha_1, \alpha_2 \in \mathbb{R}$,

$$\begin{aligned} \ell_i(q) - \ell_i(p) &= \underline{L}(q) + (e_i - q)' \cdot ((A(\tilde{q})', 0)' + \alpha_1 \mathbb{1}) - \\ &\quad (\underline{L}(p) + (e_i - p)' \cdot ((D\underline{L}(\tilde{p})', 0)' + \alpha_2 \mathbb{1})), \\ &= \underline{L}(q) - \underline{L}(p) + (\tilde{e}_i - \tilde{q})' \cdot A(\tilde{q}) - (\tilde{e}_i - \tilde{p})' \cdot D\underline{L}(\tilde{p}) + \gamma, \quad \forall A(\tilde{q}) \in \partial \underline{L}(\tilde{q}), \end{aligned}$$

where $A(\tilde{q}) \in \partial \underline{L}(\tilde{q})$, and

$$\begin{aligned} \gamma &= -(e_i - q)' \cdot \alpha_1 \mathbb{1} + (e_i - p)' \cdot \alpha_2 \mathbb{1} = -\alpha_1 + \alpha_1 q' \cdot \mathbb{1} + \alpha_2 - \alpha_2 p' \cdot \mathbb{1} \\ &= -\alpha_1 + \alpha_1 + \alpha_2 - \alpha_2 = 0 \end{aligned}$$

and so

$$\ell_i(q) - \ell_i(p) = \underline{L}(q) - \underline{L}(p) + (\tilde{e}_i - \tilde{q})' \cdot (A(\tilde{q}) - D\underline{L}(\tilde{p})) + (\tilde{p} - \tilde{q})' \cdot D\underline{L}(\tilde{p}).$$

By continuity of \underline{L} , $\|\underline{L}(q) - \underline{L}(p)\| < \varepsilon$ for small enough δ . Furthermore by (34), $\|A(\tilde{q}) - D\underline{L}(\tilde{p})\| \leq 0$ and $\|\tilde{p} - \tilde{q}\| \leq \varepsilon$. Hence $\|\ell_i(q) - \ell_i(p)\| \leq \varepsilon + \varepsilon + \delta$ which can be made arbitrarily small by suitable choice of ε . Thus ℓ_i is continuous for all $i \in [n]$ and so ℓ is continuous.

(\Rightarrow) Assume that \underline{L} is not differentiable at $p \in \Delta^n$. Thus there exists two different supergradients at p : $A(\tilde{p})$ and $B(\tilde{p})$. Assume that one of these supergradients, $A(\tilde{p})$, is the one associated to the loss ℓ in the sense that for all $i \in [n]$ $\ell_i(p) = \underline{L}(p) + (\tilde{e}_i - \tilde{p})' \cdot A(\tilde{p})$.

Suppose that $\forall i \in [n]$,

$$(e_i - p)' \cdot ((A(\tilde{p})', 0)' + \alpha_1 \mathbb{1}) \leq (e_i - p)' \cdot ((B(\tilde{p})', 0)' + \alpha_2 \mathbb{1}), \quad \alpha_1, \alpha_2 \in \mathbb{R}. \quad (35)$$

Thus

$$\begin{aligned} \sum_{i \in [n]} q_i (e_i - p)' \cdot ((A(\tilde{p})', 0)' + \alpha_1 \mathbb{1}) &\leq \sum_{i \in [n]} q_i (e_i - p)' \cdot ((B(\tilde{p})', 0)' + \alpha_2 \mathbb{1}), \quad \forall q \in \Delta^n, \alpha_1, \alpha_2 \in \mathbb{R} \\ \Leftrightarrow (q - p)' \cdot ((A(\tilde{p})', 0)' + \alpha_1 \mathbb{1}) &\leq (q - p)' \cdot ((B(\tilde{p})', 0)' + \alpha_2 \mathbb{1}), \quad \forall q \in \Delta^n, \alpha_1, \alpha_2 \in \mathbb{R} \\ \Leftrightarrow (\tilde{q} - \tilde{p})' \cdot A(\tilde{p}) &\leq (\tilde{q} - \tilde{p})' \cdot B(\tilde{p}), \quad \forall \tilde{q} \in \tilde{\Delta}^n. \end{aligned} \quad (36)$$

Since $p \in \Delta^n$ we can choose \tilde{q}_1 and $\tilde{q}_2 \in \tilde{\Delta}^n$ such that $\tilde{q}_1 - \tilde{p} = \tilde{p} - \tilde{q}_2$ and so the only way (36) can hold is if

$$(\tilde{q} - \tilde{p})' \cdot A(\tilde{p}) = (\tilde{q} - \tilde{p})' \cdot B(\tilde{p}).$$

Since $p \in \Delta^n$ is arbitrary, we obtain that $A(\tilde{p}) = B(\tilde{p})$, a contradiction and so (35) must be false. Thus there exists $i \in [n]$ such that

$$(e_i - p)' \cdot ((A(\tilde{p})', 0)' + \alpha_1 \mathbb{1}) > (e_i - p)' \cdot ((B(\tilde{p})', 0)' + \alpha_2 \mathbb{1}), \quad \alpha_1, \alpha_2 \in \mathbb{R}.$$

Thus

$$\exists i \in [n], (\tilde{e}_i - \tilde{p})' \cdot A(\tilde{p}) > (\tilde{e}_i - \tilde{p})' \cdot B(\tilde{p}). \quad (37)$$

Let $p_\eta := p + \eta(e_i - p)$ and denote by $C(\tilde{p}_\eta)$ the supergradient associated with ℓ at p_η (that is, $\ell_i(p_\eta) = \underline{L}(p_\eta) + (\tilde{e}_i - \tilde{p}_\eta)' \cdot C(\tilde{p}_\eta)$). By definition of the supergradient,

$$\underline{L}(p_\eta) \leq \underline{L}(p) + (\tilde{p}_\eta - \tilde{p})' \cdot B(\tilde{p}) \quad \text{and} \quad \underline{L}(p) \leq \underline{L}(p_\eta) + (\tilde{p} - \tilde{p}_\eta)' \cdot C(\tilde{p}_\eta).$$

Thus

$$\begin{aligned} \underline{L}(p_\eta) &\leq \underline{L}(p_\eta) + C(\tilde{p}_\eta)' \cdot (\tilde{p} - \tilde{p}_\eta) + B(\tilde{p})' \cdot (\tilde{p}_\eta - \tilde{p}) \\ \Rightarrow C(p_\eta)' \cdot (\tilde{p}_\eta - \tilde{p})' &\leq B(\tilde{p})' \cdot (\tilde{p}_\eta - \tilde{p})'. \end{aligned}$$

But by definition of p_η , $\tilde{p}_\eta - \tilde{p} = \tilde{p} + \eta(\tilde{e}_i - \tilde{p}) - \tilde{p} = \eta(\tilde{e}_i - \tilde{p})$. Thus for $\eta > 0$,

$$C(\tilde{p}_\eta)' \cdot (\tilde{e}_i - \tilde{p}) \leq B(\tilde{p})' \cdot (\tilde{p} - \tilde{e}_i). \quad (38)$$

and so

$$\text{Now } \ell_i(p_\eta) = \underline{L}(p_\eta) + (\tilde{e}_i - \tilde{p}_\eta)' \cdot C(\tilde{p}_\eta), \text{ Hence (38) implies}$$

$$\ell_i(p_\eta) \leq \underline{L}(p_\eta) + (\tilde{e}_i - \tilde{p})' \cdot B(\tilde{p}).$$

However $\lim_{\eta \searrow 0} p_\eta = p$ and by continuity of \underline{L} ,

$$\begin{aligned} \lim_{\eta \searrow 0} \underline{L}(p_\eta) + (\tilde{e}_i - \tilde{p})' \cdot B(\tilde{p}) &= \underline{L}(p) + (\tilde{e}_i - \tilde{p})' \cdot B(\tilde{p}) \\ &< \underline{L}(p) + (\tilde{e}_i - \tilde{p})' \cdot A(\tilde{p}) \\ &= \ell_i(p) \text{ by (37)}. \end{aligned}$$

Thus $\lim_{\eta \searrow 0} \ell_i(p_\eta) < \ell_i(p)$ and so ℓ_i is not continuous at p and thus ℓ is not continuous at p .

B.3 Proof of Proposition 7

The proof shows the equivalence of statements 1 and 2 and, separately the equivalence of 1 and 3 and 1 and 4.

1 \Rightarrow 2: Suppose that ℓ is proper and $p, q \in \partial \Delta^n$. Let $\tilde{L}^{p,q}$ denote the conditional risk associated with $\tilde{\ell}^{p,q}$. Then

$$\begin{aligned} \tilde{L}^{p,q}(\eta, \tilde{\eta}) &= (\eta q + (1 - \eta)p)' \cdot \ell(p + \eta(q - p)) = \underline{L}(p + \eta(q - p), p + \tilde{\eta}(q - p)) \\ &\geq \underline{L}(p + \eta(q - p), p + \eta(q - p)) = \tilde{L}^{p,q}(\eta, \eta). \end{aligned}$$

Hence $\tilde{\ell}^{p,q}$ is proper.

1 \Leftarrow 2: Suppose that $\tilde{p}^{p,q}$ is proper $\forall p, q \in \partial \Delta^n$. Suppose $p, q \in \Delta^n$. Then there exists \tilde{p} and $\tilde{q} \in \partial \Delta^n$ such that $p = \tilde{p} + \eta(\tilde{q} - \tilde{p})$ and $q = \tilde{p} + \hat{\eta}(\tilde{q} - \tilde{p})$, where $\eta, \hat{\eta} \in [0, 1]$ (the line passing through p and q cuts $\partial \Delta^n$ at \tilde{p} and \tilde{q} ; see Figure 12). Then

$$L(p, q) = \tilde{L}^{\tilde{p}, \tilde{q}}(\eta, \hat{\eta}) \geq \tilde{L}^{\tilde{p}, \tilde{q}}(\eta, \eta) = L(p, p).$$

Hence ℓ is proper.

One can easily prove that 3 \Rightarrow 1 by taking $h_1 = 0$.

For 3 \Leftarrow 1 we use a result of Lambert (2010, Proposition 1), which tells us a binary probability estimation loss ℓ_b is proper

if and only if $\forall \eta \leq \eta_1 \leq \eta_2$ or $\eta \geq \eta_2, L_b(\eta, \eta_1) \leq L_b(\eta, \eta_2)$ (the assumptions on the statistic are checked in the binary case with the statistic function $\Gamma: \Delta^2 \ni p \mapsto \mathbb{E}(p) \in [0, 1]$). We also know that if ℓ is proper then $\forall p, q \in \partial \Delta^n$, $\tilde{p}^{p,q}$ (introduced in Proposition 7) is proper. We assume that ℓ is proper, $\forall p, q \in \Delta^n$, $\forall 0 \leq h_1 \leq h_2$, we introduce the projections $\tilde{p}, \tilde{q} \in \partial \Delta^n$ of p and q , then there exists η and μ such that $p = \tilde{p} + \eta(\tilde{q} - \tilde{p})$ and $q = \tilde{p} + \mu(\tilde{q} - \tilde{p})$. We denote $\eta_1 = \eta + h_1(\mu - \eta)$ and $\eta_2 = \eta + h_2(\mu - \eta)$. Then the result of Lambert applied to $\tilde{p}^{p,q}$ gives us $L(p, p + h_1(q - p)) \leq L(p, p + h_2(q - p))$. One can adapt the proof in the case of strict properness.

1 \Rightarrow 4: If ℓ is proper, $p^i \cdot \ell(q) = q^i \cdot \ell(q) + (p - q)^i \cdot \ell(q) = \underline{L}(q) + (p - q)^i \cdot \ell(q)$. Thus $\forall q \in \Delta^n$ there exists $A(q)$ such as $L(p, q) = \underline{L}(q) + (p - q)^i \cdot A(q)$. Since ℓ is proper, $\forall p \in \Delta^n$, $0 \leq L(p, p) - L(p, q) = \underline{L}(q) - \underline{L}(p) + (p - q)^i \cdot A(q)$. Then $A(q)$ is a supergradient of $\underline{L} = f$ (which is concave) at q , and $p^i \cdot \ell(q) = f(q) + (p - q)^i \cdot A(q)$.

4 \Rightarrow 1: If there exists a function f concave and $\forall q \in \Delta^n$, there exists a supergradient $A(q) \in \partial f(q)$ such that $\forall p, q \in \Delta^n$, $p^i \cdot \ell(q) = f(q) + (p - q)^i \cdot A(q)$. Then, $L(p, p) - L(p, q) = f(p) - f(q) + (p - q)^i \cdot A(q) \geq 0$. Hence ℓ is proper.

B.4 Proof of Proposition 10

The proposition is a direct consequence of the characterization of differentiable binary proper losses (Reid and Williamson, 2010). A differentiable binary loss λ is proper if and only if $\frac{-\lambda'_1(\eta)}{1-\eta} = \frac{\lambda'_1(\eta)}{\eta} \geq 0, \forall \eta \in (0, 1)$.

Suppose the loss ℓ can be expressed as a proper composite loss: $\ell = \lambda \circ \psi^{-1}$ and so $\lambda = \ell \circ \psi$. Therefore for $y \in \{-1, 1\}$, $\lambda'_1(\eta) = \psi^i(\eta) \ell'_1(\psi(\eta))$. Then λ is proper and thus

$$\frac{-\lambda'_1(\eta)}{1-\eta} = \frac{\lambda'_1(\eta)}{\eta}, \quad \forall \eta \in (0, 1) \quad (39)$$

$$\Leftrightarrow \frac{\psi^i(\psi^{-1}(v))}{1-\psi^{-1}(v)} \ell'_1(v) = \frac{\psi^i(\psi^{-1}(v))}{\psi^{-1}(v)} \ell'_{-1}(v), \quad \forall v \in \mathcal{Y}$$

$$\Leftrightarrow \psi^i(\psi^{-1}(v)) = 0 \text{ or } \ell'_{-1}(v) = \ell'_1(v) = 0 \text{ or } \psi^{-1}(v) = \frac{\ell'_{-1}(v)}{\ell'_1(v)}, \quad \forall v \in \mathcal{Y}. \quad (40)$$

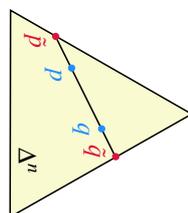


Figure 12: Illustration of proof of Proposition 7.

Since ψ is differentiable and invertible, ψ' cannot equal zero on an interval. By continuity, ψ^{-1} is uniquely defined on an interval I when $\forall v_1, v_2 \in I, \exists v \in [v_1, v_2], \ell'_1(v) \neq 0$ or $\ell'_{-1}(v) \neq 0$. If $I = \mathcal{Y}$ then ψ is unique and thus $\lambda = \ell \circ \psi$ is unique.

If $\ell'_1(v) = \ell'_{-1}(v) = 0, \forall v \in [v_1, v_2]$ then one can choose any $\psi|_{[v_1, v_2]}$ which is differentiable, invertible and such that ψ is continuous in v_1 and v_2 and as ℓ_1 and ℓ_{-1} are constant on $[v_1, v_2]$, $\lambda(\eta) = \ell(\psi(\eta))$ does not depend on ψ and so in any case λ is unique.

B.5 Proof of Proposition 11

The loss λ is proper if and only if (39) and $-\lambda'_1(\eta) \geq 0$ and $\lambda'_{-1}(\eta) \geq 0$. This is equivalent to there exists an invertible ψ such that (40) holds and

$$-\psi^i(\psi^{-1}(v)) \ell'_1(v) \geq 0 \text{ and } \psi^i(\psi^{-1}(v)) \ell'_{-1}(v) \geq 0, \quad \forall v \in \mathcal{Y}. \quad (41)$$

(\Rightarrow) Suppose ℓ has a composite representation with ψ strictly increasing and thus $\psi'(v) > 0$ for all $v \in \mathcal{Y}$ and thus $-\ell'_1(v) \geq 0$ and $\ell'_{-1}(v) \geq 0$. Hence ℓ_1 is decreasing and ℓ_{-1} is increasing. By hypothesis, $\ell'_{-1}(v) \neq 0$ or $\ell'_1(v) \neq 0$. Furthermore $\psi'(v)$ can not equal zero except at isolated points. Thus (40) implies $\psi^{-1}(v) = \frac{\ell'_{-1}(v)}{\ell'_1(v) - \ell'_1(v)}$ and thus f is strictly increasing. (If instead ψ was strictly decreasing, we can run the same argument to conclude ℓ_1 is increasing, ℓ_{-1} is decreasing and f is strictly decreasing.)

(\Leftarrow) Suppose ℓ_1 is decreasing, ℓ_{-1} is increasing and f is strictly increasing. By setting $\psi^{-1}(v) = \frac{1}{1-f(v)}$, ψ^{-1} is invertible and (41) holds. The other case is analogous.

B.6 Proof of Proposition 17

Fix an arbitrary $p \in \Delta^n$. The function f_p is quasi-convex if its α sublevel sets

$$F_p^\alpha := \{q \in \Delta^n : p^i \ell(q) \leq \alpha\}$$

are convex for all $\alpha \in \mathbb{R}$ (Greenberg and Pierskalla, 1971). Fix an arbitrary $\alpha > \underline{L}(p)$, and thus $F_p^\alpha \neq \emptyset$. Let

$$Q_p^\alpha := \{x \in \mathbb{R}^n : p^i x \leq \alpha\}$$

so $F_p^\alpha = \{q \in \Delta^n : \ell(q) \in Q_p^\alpha\}$. Denote by

$$H_p^\beta := \{x : x^i \cdot q = \beta\}$$

the hyperplane in direction $q \in \Delta^n$ with offset $\beta \in \mathbb{R}$ and by

$$H_p^\beta := \{x : x^i \cdot q \geq \beta\}$$

the corresponding half-space. Since ℓ is proper, \mathcal{S}_ℓ is supported at $x = \ell(q)$ by the hyperplane $H_q^{\underline{L}(q)}$ and furthermore since \mathcal{S}_ℓ is convex, $\mathcal{S}_\ell = \bigcap_{q \in \Delta^n} H_q^{\underline{L}(q)}$.

Let

$$V_p^\alpha := \bigcap_{x \in (\Delta^n) \cap Q_p^\alpha} H_x^{\ell^{-1}(x)} = \bigcap_{q \in F_p^\alpha} H_q^{\underline{L}(q)}$$

Observe that

$$\inf\{\beta : h_{\beta}^{\Delta} \cap \ell(\mathcal{Y}) \neq \emptyset\} = \inf\{p' \cdot \ell(v) : h_{p'}^{\Delta, \ell(v)} \cap \ell(\mathcal{Y}) \neq \emptyset, v \in \mathcal{Y}\}.$$

Thus $p' \cdot \lambda(p) = p' \cdot (\ell \circ \psi)(p) = \inf_{v \in \mathcal{Y}} p' \cdot \ell(v)$. By Δ^x -smoothness of $\ell(\mathcal{Y})$, for all $v \in \mathcal{Y}$ there exists a unique $p \in \Delta^n$ such that $\psi(p) = v$ and thus ψ is invertible. Hence $p' \cdot \lambda(p) = \inf_{q \in \Delta^n} p' \cdot \lambda(q)$ and thus λ is proper. Since $\ell(\mathcal{Y})$ is Δ^n -strictly convex there exists a unique point where $h_{\beta}^{\Delta(p)}$ supports $\ell(\mathcal{Y})$. Hence λ is strictly proper and we have shown that ℓ has a strictly proper composite representation.

Strictly proper composite $\ell \Rightarrow \Delta^x$ -strictly convex: Suppose ℓ has a strictly proper composite representation $\ell(v) = \lambda(\psi^{-1}(v))$. Pick $p \in \Delta^n$. By assumption, there exists $v \in \mathcal{Y}$ such that $\psi^{-1}(v) = p$. Since λ is strictly proper, there is a unique $q \in \Delta^n$ which minimises $q \mapsto p' \cdot \lambda(q)$. By invertibility of ψ , there thus exists a unique $v \in \mathcal{Y}$ that minimises $v \mapsto p' \cdot \ell(v)$ and so there is a unique x at which h_{β}^{Δ} supports $\ell(\mathcal{Y})$ for some $\beta \in \mathbb{R}$. Thus $\ell(\mathcal{Y})$ is Δ^n -strictly convex.

Now pick an arbitrary $v \in \mathcal{Y}$ which induces an arbitrary $x = \ell(v) \in \ell(\mathcal{Y})$. Let $p = \psi^{-1}(v)$. Then $h_{\beta}^{\Delta(p)}$ supports $\ell(\mathcal{Y})$ at x since λ is proper. Suppose there was another $q \neq p, q \in \Delta^n$ such that $h_{\beta}^{\Delta(q)}$ supports $\ell(\mathcal{Y})$ at x . But that would require that $v = \psi(q)$ which is impossible since $v = \psi(p)$ and ψ is invertible. Thus p is unique and $\ell(\mathcal{Y})$ is Δ^n -smooth.

References

- Jacob Abernethy, Alekh Agarwal, Peter L. Bartlett, and Alexander Rakhlin. A stochastic view of optimal regret through minimax duality. In *Proceedings of the 22nd Annual Conference on Learning Theory*, 2009.
- Erin L. Allwein, Robert E. Schapire, and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *The Journal of Machine Learning Research*, 1:113–141, 2001.
- Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, March 2006.
- Paul N. Bennett. Using asymmetric distributions to improve text classifier probability estimates. In *Proceedings of SIGIR'03*, pages 111–118, 2003.
- James O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer, New York, 1985.
- Alina Beygelzimer, John Langford, and Pradeep Ravikumar. Multiclass classification with filter trees. Preprint, June 2007. URL <http://hunch.net/~11/projects/reductions/mc-to-b/invertedTree.pdf>.
- Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- Elim Mikhailovich Bronshteyn. Extremal convex functions. *Siberian Mathematical Journal*, 19: 6–12, 1978.
- Lawrence D. Brown. A complete class theorem for statistical problems with finite sample spaces. *The Annals of Statistics*, 9(6):1289–1300, 1981.
- Andreas Buja, Werner Suetzle, and Yi Shen. Loss functions for binary class probability estimation and classification: Structure and applications. Technical report, University of Pennsylvania, November 2005. URL <http://www-stat.wharton.upenn.edu/~buja/a/PAPERS/paper-proper-scoring.pdf>.
- Hermann Chernoff and Lincoln E. Moses. *Elementary Decision Theory*. Dover, 1986.
- Jesús Cid-Sueiro and Anbal R. Figueiras-Vidal. On the structure of strict sense Bayesian cost functions and its applications. *IEEE Transactions on Neural Networks*, 12(3):445–455, May 2001.
- Ira Cohen and Moises Goldszmidt. Properties and benefits of calibrated classifiers. Technical Report HPL-2004-22(R.1), HP Laboratories, Palo Alto, July 2004. URL <http://www.hpl.hp.com/techreports/2004/HPL-2004-22R1.pdf>.
- Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.
- A. Philip Dawid. The geometry of proper scoring rules. *Annals of the Institute of Statistical Mathematics*, 59(1):77–93, March 2007.
- Morris H. DeGroot. Uncertainty, Information, and Sequential Experiments. *The Annals of Mathematical Statistics*, 33(2):404–419, 1962.
- Thomas G. Dietterich and Ghulun Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.
- Paul K. Fackler. Notes on matrix calculus. North Carolina State University, 2005. URL <http://www4.ncsu.edu/~pfackler/MatCalc.pdf>.
- Thomas S. Ferguson. *Mathematical Statistics: A Decision Theoretic Approach*. Academic Press, New York, 1967.
- Dario García-García and Robert C. Williamson. Divergences and risks for multiclass experiments. In *Conference on Learning Theory (JMLR: W&CP)*, volume 23, pages 28.1–28.20, 2012.
- Gary F.V. Glonek. A class of regression models for multivariate categorical responses. *Biometrika*, 83(1):15–28, 1996.
- Gary F.V. Glonek and Peter McCullagh. Multivariate logistic models. *Journal of the Royal Statistical Society, Series B*, 57(3):533–546, 1995.
- Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, March 2007.

- Harvey J. Greenberg and William P. Pierskalla. A review of quasi-convex functions. *Operations Research*, 19(7):1553–1570, November 1971.
- Peter D. Grünwald and A. Philip Dawid. Game theory, maximum entropy, minimum discrepancy and robust Bayesian decision theory. *The Annals of Statistics*, 32(4):1367–1433, 2004.
- Trevor Hastie and Robert Tibshirani. Classification by pairwise coupling. *The Annals of Mathematical Statistics*, 26(2):451–471, 1998.
- Simon I. Hill and Arnaud Doucet. A framework for kernel-based multi-category classification. *Journal of Artificial Intelligence Research*, 30:525–564, 2007.
- Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Fundamentals of Convex Analysis*. Springer, Berlin, 2001.
- Roger A. Horn and Charles A. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.
- Tzu-Kuo Huang, Ruby C. Weng, and Chih-Jen Lin. Generalized Bradley-Terry models and multi-class probability estimates. *Journal of Machine Learning Research*, 7:85–115, 2006.
- Ayodele Ighodaro, Thomas Santner, and Lawrence Brown. Admissibility and complete class results for the multinomial estimation problem with entropy and squared error loss. *Journal of Multivariate Analysis*, 12:469–479, 1982.
- Yuri Kalnishkan and Michael V. Vyugin. The weak aggregating algorithm and weak mixability. *Journal of Computer and System Sciences*, 74:1228–1244, 2008.
- Parameswaran Kamalaruban, Robert C. Williamson, and Xinhua Zhang. Exp-concavity of proper composite losses. In *JMLR Workshop and Conference Proceedings (Proceedings COLT 2015)*, volume 40, 2015.
- Jack Carl Kiefer. *Introduction to Statistical Inference*. Springer-Verlag, New York, 1987.
- Nicolas S. Lambert. Elicitation and evaluation of statistical forecasts. Technical report, Stanford University, March 2010. URL <http://www.stanford.edu/~nlambert/lambert-elicitation.pdf>.
- Yufeng Liu. Fisher consistency of multicategory support vector machines. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, pages 289–296, 2007.
- Jan R. Magnus and Heinz Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics (revised edition)*. John Wiley & Sons, 1999.
- Peter McCullagh and John A. Nelder. *Generalized Linear Models*. Chapman & Hall/CRC, 1989.
- Aditya K. Menon and Robert C. Williamson. Bipartite ranking: risk, optimality, and equivalences. *Journal of Machine Learning Research*, July 2014. URL <http://users.cecs.anu.edu.au/~williams/papers/P194.pdf>. Submitted.
- Walter Meyer and David C. Kay. A convexity structure admits but one real linearization of dimension greater than one. *Journal of the London Mathematical Society* (2), 7:124–130, 1973.
- Geert Molenberghs and Emmanuel Lesaffre. Marginal modelling of multivariate categorical data. *Statistics in Medicine*, 18:2237–2255, 1999.
- Indraneel Mukherjee and Robert E Schapire. A theory of multiclass boosting. *The Journal of Machine Learning Research*, 14(1):437–497, 2013.
- Harikrishna Narasimhan and Shivani Agarwal. On the relationship between binary classification, bipartite ranking, and binary class probability estimation. In *Advances in Neural Information Processing Systems*, pages 2913–2921, 2013.
- Robert F. Nau. Should scoring rules be ‘effective’? *Management Science*, 31(5):527–535, May 1985.
- Tapan K. Nayak and Dayanand N. Naik. Estimating multinomial cell probabilities under quadratic loss. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 38(1): 3–10, 1989.
- XuanLong Nguyen, Martin J. Wainwright, and Michael I. Jordan. On surrogate loss functions and f -divergences. *Annals of Statistics*, 37:876–904, 2009.
- Robert R. Phelps. *Lectures on Choquet’s Theorem*, volume 1757 of *Lecture Notes in Mathematics*. Springer, 2nd edition, 2001.
- Mark D. Reid and Robert C. Williamson. Composite binary losses. *Journal of Machine Learning Research*, 11:2387–2422, 2010.
- Mark D. Reid and Robert C. Williamson. Information, divergence and risk for binary experiments. *Journal of Machine Learning Research*, 12:731–817, March 2011.
- Mark D. Reid, Rafael M. Frongillo, Robert C. Williamson, and Nishant A. Mehta. Generalized mixability via entropic duality. *JMLR: Workshop and Conference Proceedings (COLT 2015)*, 40:1–22, 2015.
- R. Tyrrell Rockafellar and Roger J-B. Wets. *Variational Analysis*. Springer-Verlag, Berlin, 2004.
- Raúl Santos-Rodríguez, Alicia Guerrero-Curiñes, Rocío Alataz-Rodríguez, and Jesús Cid-Sueiro. Cost-sensitive learning based on Bregman divergences. *Machine Learning*, 76:271–285, 2009.
- Rolf Schneider. *Convex Bodies: The Brunn-Minkowski Theory*. Cambridge University Press, 1993.
- Clayton Scott. Surrogate losses and regret bounds for cost-sensitive classification with example-dependent costs. In *Proc. of the 28th International Conference on Machine Learning (ICML)*, 2011.

- Clayton Scott. Calibrated asymmetric surrogate losses. *Electronic Journal of Statistics*, 6:958–992, 2012.
- Qinfeng Shi, Mark Reid, and Tiberio Caetano. Conditional random fields and support vector machines: A hybrid approach. arXiv:1009.3346v1, September 2010. URL http://arxiv.org/PS_cache/arxiv/pdf/1009/1009.3346v1.pdf.
- Barry Simon. *Convexity: An Analytic Viewpoint*. Cambridge University Press, 2011.
- Maurice Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8(1):171–176, 1958.
- Ingo Steinwart. How to compare different loss functions. *Constructive Approximation*, 26: 225–287, 2007.
- Ambuj Tewari and Peter L. Bartlett. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 8:1007–1025, 2007.
- Frederick A. Valentine. *Convex Sets*. McGraw-Hill, New York, 1964.
- Tim van Erven, Mark D. Reid, and Robert C. Williamson. Mixability is Bayes risk curvature relative to log loss. In *Proceedings of the 24th Annual Conference on Learning Theory*, 2011.
- Tim van Erven, Peter Grünwald, Mark D Reid, and Robert C Williamson. Mixability in statistical learning. In *Advances in Neural Information Processing Systems*, pages 1691–1699, 2012a.
- Tim van Erven, Mark D. Reid, and Robert C. Williamson. Mixability is Bayes risk curvature relative to log loss. *Journal of Machine Learning Research*, 13:1639–1663, May 2012b.
- Tim van Erven, Peter D. Grünwald, Nishant A. Mehta, Mark D. Reid, and Robert C. Williamson. Fast rates in statistical and online learning. *Journal of Machine Learning Research*, 16:1793–1861, 2015.
- William J. Vetter. Derivative operations on matrices. *IEEE Transactions on Automatic Control*, 15(2):241–244, April 1970.
- Vobodya Vovk. A game of prediction with expert advice. In *Proceedings of the Eighth Annual Conference on Computational Learning Theory*, pages 51–60. ACM, 1995.
- Vobodya Vovk and Fedor Zhdanov. Prediction with expert advice for the Brier game. *Journal of Machine Learning Research*, 10:2445–2471, 2009.
- Roger Webster. *Convexity*. Oxford University Press, 1994.
- Robert C. Williamson. The geometry of losses. In *Conference on Learning Theory (JMLR: W&CP)*, volume 35, pages 1078–1108, 2014.
- Ting-Fan Wu and Ruby C. Weng. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5:975–1005, 2004.
- Tong Tong Wu and Kenneth Lange. Multicategory vertex discriminant analysis for high-dimensional data. *The Annals of Applied Statistics*, 4:1698–1721, 2010.
- Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of SIGKDD*, 2002.
- Tong Zhang. Statistical analysis of some multi-category large margin classification methods. *Journal of Machine Learning Research*, 5:1225–1251, 2004.
- Zhihua Zhang, Michael I. Jordan, Wu-Jun Li, and Di-Yan Yeung. Coherence functions for multicategory margin-based classification methods. In *Proceedings of the Twelfth Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009.
- Ji Zhu, Hui Zou, Saharon Rosset, and Trevor Hastie. Multi-class AdaBoost. *Statistics and its Interface*, 2:349–360, 2009.
- Hui Zou, Ji Zhu, and Trevor Hastie. The margin vector, admissible loss and multi-class margin-based classifiers. Preprint, 2005. URL <http://www-stat.stanford.edu/~hastie/Papers/margin.pdf>.
- Hui Zou, Ji Zhu, and Trevor Hastie. New multicategory boosting algorithms based on multicategory Fisher-consistent losses. *The Annals of Applied Statistics*, 2(4):1290–1306, 2008.

Learning Latent Variable Models by Pairwise Cluster Comparison: Part I – Theory and Overview

Nuaman Asbeh

Department of Industrial Engineering and Management
Ben-Gurion University of the Negev
Beer Sheva, 84105, Israel

ASBEH@POST.BGU.AC.IL

Boaz Lerner

Department of Industrial Engineering and Management
Ben-Gurion University of the Negev
Beer Sheva, 84105, Israel

BOAZ@BGU.AC.IL

Editors: Isabelle Guyon and Alexander Statnikov

Abstract

Identification of latent variables that govern a problem and the relationships among them, given measurements in the observed world, are important for causal discovery. This identification can be accomplished by analyzing the constraints imposed by the latents in the measurements. We introduce the concept of *pairwise cluster comparison* (PCC) to identify causal relationships from clusters of data points and provide a two-stage algorithm called *learning PCC* (LPCC) that learns a latent variable model (LVM) using PCC. First, LPCC learns exogenous latents and latent colliders, as well as their observed descendants, by using pairwise comparisons between data clusters in the measurement space that may explain latent causes. Since in this first stage LPCC cannot distinguish endogenous latent non-colliders from their exogenous ancestors, a second stage is needed to extract the former, with their observed children, from the latter. If the true graph has no serial connections, LPCC returns the true graph, and if the true graph has a serial connection, LPCC returns a pattern of the true graph. LPCC's most important advantage is that it is not limited to linear or latent-tree models and makes only mild assumptions about the distribution. The paper is divided in two parts: Part I (this paper) provides the necessary preliminaries, theoretical foundation to PCC, and an overview of LPCC; Part II formally introduces the LPCC algorithm and experimentally evaluates its merit in different synthetic and real domains. The code for the LPCC algorithm and data sets used in the experiments reported in Part II are available *online*.

Keywords: causal discovery, clustering, learning latent variable model, multiple indicator model, pure measurement model

1. Introduction

Latent (unmeasured, hidden, unrecorded) variables, as opposed to observed (measured, manifest, recorded) variables, cannot usually be observed directly in a domain but only inferred from other observed variables or indicators (Spirites, 2013). Concepts such as “quality of life,” “economic stability,” “gravitational fields,” and “psychological stress”

play a key role in scientific theories and models, and yet such entities are latent (Klee, 1997).

Sometimes, latent variables correspond to aspects of physical reality that could, in principle, be measured but may not be for practical reasons, for example, “quarks”. In this situation, the term hidden variables is commonly used, reflecting the fact that the variables are “really there”, but hidden. On the other hand, latent variables may not be physically real but instead correspond to abstract concepts such as “psychological stress” or “mental states”. The terms hypothetical variables or hypothetical constructs may be used in these situations.

Latent variable models (LVMs) represent latent variables and the causal relationships among them to explain observed variables that have been measured in the domain. These models are common and essential in diverse areas, such as in economics, social sciences, psychology, natural language processing, and machine learning. Thus, they have recently become the focus of an increasing number of studies. LVMs reduce dimensionality by aggregating (many) observed variables into a few latent variables, each of which represents a “concept” explaining some aspects of the domain that can be interpreted from the data. Latent variable modeling is a century-old enterprise in statistics. It originated with the work of Spearman (1904), who developed factor analytic models for continuous variables in the context of intelligence testing.

Learning an LVM exploits values of the measured variables as manifested in the data to make an inference about the causal relationships among the latent variables and to predict the value of these variables. Statistical methods for learning an LVM, such as factor analysis, are most commonly used to reveal the existence and influence of latent variables. Although these methods effectively reduce dimensionality and may fit the data reasonably well, the resulting models might not have any correspondence to real causal mechanisms (Silva et al., 2006). On the other hand, the focus of learning Bayesian networks (BNs) is on causal relations among observed variables, whereas the detection of latent variables and the interrelations among themselves and with the observed variables has received little attention. Learning an LVM using Inductive Causation* (IC*) (Pearl, 2000; Pearl and Verma, 1991) and Fast Causal Inference (FCI) (Spirites et al., 2000) returns partial ancestral graphs, which indicate for each link whether it is a (potential) manifestation of a hidden common cause for the two linked variables. The structural EM algorithm (Friedman, 1998) learns a structure using a fixed set of previously given latents. By searching for “structural signatures” of latents, the FindHidden algorithm (Elidan et al., 2000) detects substructures that suggest the presence of latents in the form of dense subnetworks. Elidan and Friedman (2001) give a fast algorithm for determining the cardinality – the number of possible states – of latent variables introduced this way. However, Silva et al. (2006) suspected that FindHidden cannot always find a pure measurement sub-model,¹ which is a flaw in causal analysis. Also, the recovery of latent trees of binary and Gaussian variables has been suggested (Pearl, 2000). Hierarchical latent class (HLC) models, which are rooted trees where the leaf nodes are observed while all other nodes are latent, were proposed for the clustering of categorical data (Zhang, 2004). Two greedy algorithms are suggested (Harmeling

¹A pure measurement model contains all graph variables and all and only edges directed from latent variables to observed variables, where each observed variable has only one latent parent and no observed parent.

and Williams, 2011) to expedite learning of both the structure of a binary HLC and the cardinalities of the latents. The BIN-G algorithm determines both the structure of the tree and the cardinality of the latent variables in a bottom-up fashion. The BIN-A algorithm first determines the tree structure using agglomerative hierarchical clustering and then determines the cardinality of the latent variables in the same manner as the BIN-G algorithm. Latent-tree models are also used to speed approximate inference in BNs trading the approximation accuracy with inferential complexity (Wang et al., 2008).

Models in which multiple latents may have multiple indicators (observed children), also known as multiple indicator models (MIMs) (Bartholomew et al., 2002; Spirites, 2013), are a very important subclass of structural equation models (SEM), which are widely used, together with BNs, in applied and social sciences to analyze causal relations (Pearl, 2000; Shimizu et al., 2011). For these models, and others that are not tree-constrained, most of the mentioned algorithms may lead to unsatisfactory results. This is one of the most difficult problems in machine learning and statistics since, in general, a joint distribution can be generated by an infinite number of different LVMs. However, an algorithm that fills the gap between learning latent-tree models and learning MIMs is BuildPureClusters (BPC; Silva et al., 2006). BPC searches for the set (an equivalence class) of MIMs that best matches the set of vanishing tetrad differences (Scheines et al., 1995), but is limited to linear models (Spirites, 2013).

In this study, we make another attempt in this direction and target the goal of Silva et al. (2006), but concentrate on the discrete case, rather than on the continuous case dealt with BPC. Towards this mission, we borrow ideas and principles of clustering and unsupervised learning. Interestingly, the same difficulty in learning MIMs is also faced in the domain of unsupervised learning that confronts similar questions such as: (1) How many clusters are there in the observed data? and (2) Which classes do the clusters really represent? Due to this similarity, our study suggests linking the two domains – learning a causal graphical model with latent variables and clustering analysis. We propose a concept and an algorithm that combine learning causal graphical models with clustering. According to the *pairwise cluster comparison* (PCC) concept, we compare pairwise clusters of data points representing instantiations of the observed variables to identify those pairs of clusters that exhibit major changes in the observed variables due to changes in their ancestor latent variables. Changes in a latent variable that are manifested in changes in the observed variables reveal this latent variable and its causal paths of influence in the domain. Using the *learning PCC* (LPCC) algorithm, we learn an LVM. We identify PCCs and use them to learn latent variables – exogenous and endogenous (the latter may be either colliders or non-colliders) – and their causal interrelationships as well as their children (latent variables and observed variables) and causal paths from latent variables to observed variables.

This paper is the first of two parts that introduce, describe, and evaluate LPCC. In this paper (Part I), we provide its foundations and theoretical infrastructure, from preliminaries to a broad overview of the PCC concept and LPCC algorithm. In the second paper (Part II), we formally introduce the two-stage LPCC algorithm, which implements the PCC concept, and evaluate LPCC, in comparison to state-of-the-art algorithms, using simulated and real-world data sets. The outline of the two papers is as follows:

Part I:

- **Section 2: Preliminaries to LVM learning** describes the assumptions of our approach and basic definitions of essential concepts of graphical models and SEM;
- **Section 3: Preliminaries to LPCC** formalizes our ideas and builds the theoretical basis for LPCC;
- **Section 4: Overview of LPCC** starts with an illustrative example and a broad description of the LPCC algorithm and then describes each step of LPCC in detail;
- **Section 5: Discussion and future research** summarizes and discusses the contribution of LPCC and suggests several new avenues of research;

• **Appendix A** provides proofs to all propositions, lemmas, and theorems for which the proof is either too detailed, lengthy, or impedes the flow of reading. All other proofs are given in the body of the paper;

• **Appendix B** sets a method to calculate a threshold in support of Section 4.4; and

• **Appendix C** supplies a detailed list of assumptions LPCC makes and the meaning of their violation.

Part II:

• **Section 2: The LPCC algorithm** introduces and formally describes a two-stage algorithm that implements the PCC concept;

• **Section 3: LPCC evaluation** evaluates LPCC, in comparison to state-of-the-art algorithms, using simulated and real-world data sets;

• **Section 4: Related works** compares LPCC to state-of-the-art LVM learning algorithms;

• **Section 5: Discussion** summarizes the theoretical advantages (from Part I) and the practical benefits (from this part) of using LPCC;

• **Appendix A** brings assumptions, definitions, propositions, and theorems from Part I that are essential to Part II;

• **Appendix B** supplies additional results for the experiments with the simulated data sets; and

• **Appendix C** provides PCC analysis for two example databases.

2. Preliminaries to LVM learning

The goal of our study is to reconstruct an LVM from i.i.d. data sampled from the observed variables in an unknown model. To accomplish this, we propose learning from pairwise cluster comparison using LPCC. First, we present the assumptions that LPCC makes and the constraints it applies on LVM and compare them to those required by other state-of-the-art methods.

Assumption 1 *The underlying model is a Bayesian network, $BN = \langle G, \Theta \rangle$, encoding a discrete joint probability distribution P for a set of random variables $V = LUO$, where $G = \langle V, E \rangle$ is a directed acyclic graph (DAG) whose nodes V correspond to latents L and observed variables O , and E is the set of edges between nodes in G . Θ is the set of parameters, i.e., the conditional probabilities of variables in V given their parents.*

Assumption 2 *No observed variable in O is an ancestor of any latent variable in L . This property is called the measurement assumption (Spirtes et al., 2000).*

Before we present additional assumptions about the learned LVM, we need Definitions 1–4 (following Silva et al., 2006), which are specific to LVM:

Definition 1 *A model satisfying Assumptions 1 and 2 is a latent variable model.*

Definition 2 *Given an LVM G with a variable set V , the subgraph containing all variables in V and all and only those edges directed into variables in O is called the measurement model of G .*

Definition 3 *Given an LVM G , the subgraph containing all and only G 's latent nodes and their respective edges is called the structural model of G .*

When each model variable is a linear function of its parents in the graph plus an additive error term of positive finite variance, the latent variable model is linear; this is also known as SEM. Great interest has been shown in linear LVMs and their applications in social science, econometrics, and psychometrics (Bollen, 1989), as well as in their learning (Silva et al., 2006). The motivation to use linear models usually comes from social and related sciences. For example,² researchers give subjects a questionnaire with questions like: "On a scale of 1 to 5, how much do you agree with the statement: 'I feel sad every day'." The answer is measured by an observed variable, and the linearity of the influence of an unknown cause (say depression in this case) on the answer (value) to the question is assumed. By using other questions, which researchers assume also measure depression, they expect to discover a latent depression variable that is a parent of several observed variables (each measuring a question), together indicating depression. It is common to require several questions/observed variables for the identification of each latent variable and to consider the information revealed through only a single question as noise, which cannot guarantee the identification of the latent. Researchers expect that other questions will be clustered by another latent variable that measures another aspect in the domain, and thus

²P. Spirtes, private communication.

questions of one cluster will be independent of questions of another cluster conditioned on the latent variables. They also attribute unconditional independence that is detected between observed variables of different clusters to errors in the learning algorithm.

Adding the linearity assumption to Assumptions 1 and 2 allows for the transformation of Definition 1 into that of a linear LVM. Since assuming linearity means linearity is assumed in the measurement model, a key to learning a linear LVM is learning the measurement model and only then the structural model. In learning the measurement model of MIM, the linearity assumption entails constraints on the covariance matrix of the observed variables and thereby eliminates learning co-variants (dependencies) between pairs of observed variables that "should" not be connected in the learned model (Silva et al., 2006; Spirtes, 2013). If, however, the linearity assumption does not hold, the algorithms suggested in Silva et al. (2006) may not find a model and would output a "can't tell" answer, which is, nevertheless, a better result than learning an incorrect model.

In this study, we dispense with the linearity assumption and apply the above concepts to learn not necessarily linear MIMs or latent-tree models. Our suggested algorithm, LPCC, is not limited by the linearity assumption and learns a model as long as it is MIM. In addition, we are interested in discrete LVMs.

Another important definition we need is:

Definition 4 *A pure measurement model is a measurement model in which each observed variable has only one latent parent and no observed parent.*

Assumption 3 *The measurement model of G is pure.*

As a principled way of testing conditional independence among latents, Silva et al. (2006) focus on MIMs, which are pure measurement models. Practically, these models have a smaller equivalence class of the latent structure than that of non-pure models and thus are easier to unambiguously learn. Consider, for example, that we are interested in learning the topic of a document (e.g., the first page in a newspaper) from anchor word (key phrase) distributions and that this document may cover several topics (e.g., politics, sports, and finance). Simplification of this topic modeling problem, following the representation of a topic using a latent variable in LVM, can be achieved by assuming and learning a pure measurement model representing a pure topic model for which each specific document covers only a single topic, which is reasonable in some cases, such as a sports or financial newspaper.

LPCC does not assume that the true measurement model is linear (which is a parametric assumption that, e.g., BPC makes), but rather assumes that the model is pure (a structural assumption). When the true causal model is pure, LPCC will identify it correctly (or find its pattern that represents the equivalence class of the true graph). When it is not pure, LPCC – similarly to BPC (Silva et al., 2006) – will learn a pure sub-model of the true model using two indicators for each latent (compared to three indicators per latent that are required by BPC). Part II of this paper presents several examples of real-world problems from different domains for which LPCC learns a pure (sometimes sub-) model, never less accurately than other methods.

That is, LPCC assumes that:

Assumption 4 *The true model G is MIM, in which each latent has at least two observed children and may have latent parents.*

Causal structure discovery – learning the number of latent variables in the model, their interconnections and connections to the observed variables, as well as the interconnections among the observed variables – is very difficult and thus requires making some assumptions about the problem. Particularly, MIMs, in which multiple observed variables are assumed to be affected by latent variables and perhaps by each other (Spirites, 2013), are reasonable models but have attracted scant attention in the machine-learning community. As Silva et al. (2006) pointed out, factor analysis, principal component analysis, and regression analysis adapted to learning LVMs are well understood but have not been proven, under any general assumptions, to learn the true causal LVM, calling for better learning methods. By assuming that the true model manifests local influence of each latent variable on at least a small number of observed variables, Silva et al. (2006) showed that learning the complete Markov equivalence class of MIM is feasible. Similar to Silva et al. (2006), we assume that the true model is MIM; thus, this is where we place our focus on learning. Note also that based on Assumptions 3 and 4, the observed variables in G are d-separated, given the latents.

3. Preliminaries to LPCC

Figure 1 sketches a range of MIMs, which all exhibit pure measurement models, from basic to more complex models. Compared to G1, which is a basic MIM of two unconnected latents, G2 shows a structural model that is characterized by a latent collider. Note that such an LVM cannot be learned by latent-tree algorithms such as in Zhang (2004). G3 and G4 demonstrate serial and diverging structural models, respectively, that together with G2 cover the three basic structural models. G5 and G6 manifest more complex structural models comprising a latent collider and a combination of serial and diverging connections. As the structural model becomes more complicated, the learning task becomes more challenging; hence, G1–G6 present a spectrum of such challenges to an LVM learning algorithm.³

In Section 3.1, we build the infrastructure to pairwise cluster comparison that relies on understanding the influence of the exogenous latent variables on the observed variables in the LVM. This influence is divided into major and minor effects that are introduced and explained in Section 3.2. In Section 3.3, we link this structural influence to data clustering and introduce the pairwise cluster comparison concept for learning an LVM.

3.1 The influence of exogenous latents on observed variables is fundamental to learning an LVM

We distinguish between observed (O) and latent (L) variables and between exogenous (EX) and endogenous (EN) variables. EX have zero in-degree, are autonomous, and unaffected

³In Part II of the paper, we compare LPCC with BPC and exploratory factor analysis using these six LVMs. Since BPC requires three indicators per latent to identify a latent, we determined from the beginning three indicators per latent for all true models to recover. Nevertheless, in Part II, we evaluate the learning algorithms for increasing numbers of indicators.

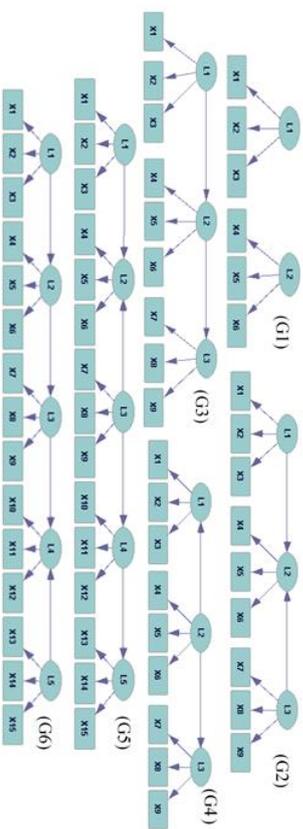


Figure 1: Example LVMs that are all MIMs. Each is based on a pure measurement model and a structural model of different complexity, posing a different challenge to a learning algorithm.

by the values of the other variables (e.g., L1 in all graphs but G4 in Figure 1), whereas EN are all non-exogenous variables in G (e.g., L2 in all graphs but G1 and G4, and X1 in all graphs in Figure 1). We identify three types of variables: (1) Exogenous latents, $EX \subset (L \cap NC)$ [all exogenous variables are latent non-colliders (NC)]; (2) Endogenous latents, $EL \subset (L \cap EN)$, which are divided into latent colliders $CCEL$ (e.g., L2 in G2 and G5; note that all latent colliders are endogenous) and latent non-colliders (in serial and diverging connections) $SEL \cap NC$ (e.g., L3 in G3, G4, and G6), thus $NC = (EX \cup S)$; and (3) Observed variables, $O \subset EN$, which are always endogenous and childless, that are divided into children of exogenous latents $OEX \subset O$ (e.g., X1 and X9 in G2), children of latent colliders $OCC \subset O$ (e.g., X4, X5, and X6 in G2), and children of endogenous latent non-colliders $OSEL \subset O$ (e.g., X4, X5, and X6 in G3). We denote value configurations of EX, EN (when we do not know whether the endogenous variables are latent or observed), EL, C, NC (when we do not know whether the non-collider variables are exogenous or endogenous), S, O, OEX, OC, and OS by $ex, en, el, c, nc, s, o, oex, oc,$ and os , respectively.

Since the underlying model is a BN, the joint probability over V , which is represented by the BN, is factored according to the local Markov assumption for G . That is, any variable in V is independent of its non-descendants in G conditioned on its parents in G :

$$P(\mathbf{V}) = \prod_{V_i \in V} P(V_i | \mathbf{Pa}_i) \quad (1)$$

where \mathbf{Pa}_i are the parents of V_i . It can be factorized under our assumptions as:

$$P(\mathbf{V}) = P(\mathbf{EX}, \mathbf{C}, \mathbf{S}, \mathbf{OEX}, \mathbf{OC}, \mathbf{OS}) = \prod_{EX_i \in \mathbf{EX}} P(EX_i) \prod_{C_j \in \mathbf{C}} P(C_j | \mathbf{Pa}_j) \prod_{S_t \in \mathbf{S}} P(S_t | Pa_t) \prod_{OEX_m \in \mathbf{OEX}} P(OEX_m | EX_m) \prod_{OC_k \in \mathbf{OC}} P(OC_k | C_k) \prod_{OS_v \in \mathbf{OS}} P(OS_v | S_v) \quad (2)$$

where \mathbf{Pa}_j are the latent parents of the latent collider C_j , Pa_t is the latent parent of the latent non-collider S_t (in other words, $\mathbf{Pa}_j, Pa_t \subset \mathbf{NC}$), $C_k \in \mathbf{C}$ and $S_v \in \mathbf{S}$ are the latent collider and latent non-collider parents of observed variables OC_k and OS_v , respectively, and $EX_m \in \mathbf{EX}$ is the exogenous latent parent of observed variable OEX_m .

In this paper, we claim and demonstrate that the influence of exogenous (latent) variables on observed variables is fundamental to learning an LVM and introduce LPCC that identifies and exploits this influence to learn an MIM. In this section, we prove that changes in values of the observed variables are due to changes in values of the exogenous variables and thus the identification of the former indicates the existence of the latter. To do that, we analyze the propagation of influence along paths connecting both variables, remembering that the paths may contain latent colliders and latent non-colliders. First, however, we should analyze paths among the latents and only then paths ending in their sinks (i.e., the observed variables). To prove that all changes in the graph, and specifically those measured in the observed variables, are the result of changes in the exogenous latent variables, we will need to first provide some definitions (following Spirtes et al., 2000; Pearl, 1988, 2000) of paths and some assumptions about the possible paths between latents in the structural model.

Definition 5 A path between two nodes V_1 and V_n in a graph \mathbf{G} is a sequence of nodes $\{V_1, \dots, V_n\}$, such that V_i and V_{i+1} are adjacent in \mathbf{G} , $1 \leq i < n$, i.e., $\{V_i, V_{i+1}\} \in \mathbf{E}$.

Note that a unique set of edges is associated with each given path. Paths are assumed to be simple by definition; in other words, no node appears in a path more than once, and an empty path consists of a single node.

Definition 6 A collider on a path $\{V_1, \dots, V_n\}$ is a node V_i , $1 < i < n$, such that V_{i-1} and V_{i+1} are parents of V_i .

Definition 7 A directed path T_{V_n} from V_1 to V_n in a graph \mathbf{G} is a path between these two nodes, such that for every pair of consecutive nodes V_i and V_{i+1} , $1 \leq i < n$ on the path, there is an edge from V_i into V_{i+1} in \mathbf{E} . V_1 is the source, and V_n is the sink of the path. A directed path has no colliders.

While BPC (Silva et al., 2006) needs to make a parametric assumption about the linearity of the model, LPCC makes assumptions about the model structure (Assumption 3 above and Assumption 5 below). This is also the approach of latent-tree algorithms (Zhang, 2004; Harmeling and Williams, 2011; Wang et al., 2008) that restrict the learned structure to a tree (note that LPCC is not limited to a tree because it allows latent variables

to be colliders). This shows a tradeoff between the structural and parametric assumptions that an algorithm for learning an LVM usually has to make; the fewer parametric assumptions the algorithm makes, the more structural assumptions it has to make and vice versa.

Assumption 5 A latent collider does not have any latent descendants (and thus cannot be a parent of another latent collider).

To distinguish between latent colliders and latent non-colliders, their observed children, and their connectivity patterns to their exogenous variables, we use Lemma 1. Latent colliders and their observed children are connected to several exogenous variables via several directed paths, whereas latent non-colliders and their observed children are connected only to a single exogenous variable via a single directed path. Use of these different connectivity patterns – from exogenous latents through endogenous latents (both colliders and non-colliders) to observed variables – simplifies (2) and the analysis of the influence of latents on observed variables.

Lemma 1

1. Each latent non-collider NC_t has only one exogenous latent ancestor EX_{NC_t} , and there is only one directed path T_{NC_t} from EX_{NC_t} (source) to NC_t (sink). (Note that we use the notation NC_t , rather than S_t , since the lemma applies to both exogenous and endogenous latent non-colliders.)
2. Each latent collider C_j is connected to a set of exogenous latent ancestors \mathbf{EX}_{C_j} via a set of directed paths T_{C_j} from \mathbf{EX}_{C_j} (sources) to C_j (sink).

Lemma 1 allows us to separate the influence of all exogenous variables to separate paths of influence, each from exogenous to observed variables. Proposition 1 quantifies the propagation of this influence along the paths through the joint probability distribution.

Proposition 1 The joint probability over \mathbf{V} due to value assignment \mathbf{ex} to exogenous set \mathbf{EX} is determined only by this assignment and the BN conditional probabilities.

Proof The first product in (2) for assignment \mathbf{ex} is of \mathbf{ex} 's priors. In the other five products, the probabilities are of endogenous latents or observed variables conditioned on their parents, which, based on Lemma 1, are either on the directed paths from \mathbf{EX} to the latents/observed variables or exogenous themselves. Either way, any assignment of endogenous latents or observed variables is a result of the assignment \mathbf{ex} to \mathbf{EX} that is mediated to the endogenous latents/observed variables by the BN probabilities:

$$P(\mathbf{V} | \mathbf{EX} = \mathbf{ex}) = P(\mathbf{EX}, \mathbf{C}, \mathbf{S}, \mathbf{OEX}, \mathbf{OC}, \mathbf{OS} | \mathbf{EX} = \mathbf{ex}) = \prod_{EX_i \in \mathbf{EX}} P(EX_i = ex_i) \prod_{C_j \in \mathbf{C}} P(C_j = c_j | \mathbf{Pa}_j = \mathbf{pa}_j) \prod_{S_t \in \mathbf{S}} P(S_t = nc_t | Pa_t = pa_t) \prod_{OEX_m \in \mathbf{OEX}} P(OEX_m = oe_{x_m} | EX_m = ex_m) \prod_{OC_k \in \mathbf{OC}} P(OC_k = oc_k | C_k = c_k) \prod_{OS_v \in \mathbf{OS}} P(OS_v = os_v | S_v = s_v) \quad (3)$$

where

- ex_i and ex_m are the values of EX_i and EX_m (the latter is the parent of the m th observed child of the exogenous latents), respectively, in the assignment \mathbf{ex} to EX ;
- $\mathbf{pa}_{-j}^{ex_{C_j}}$ is the configuration of C_j 's parents due to configuration \mathbf{ex}_{C_j} of C_j 's exogenous ancestors in \mathbf{ex} ;
- $pa_i^{ex_{S_i}}$ is the value of S_i 's parent due to the value ex_{S_i} of S_i 's exogenous ancestor in \mathbf{ex} ;
- $c_k^{ex_{C_k}}$ is the value of OC_k 's collider parent due to the configuration \mathbf{ex}_{C_k} of C_k 's exogenous ancestors in \mathbf{ex} ; and
- $s_v^{ex_{S_v}}$ is the value of OS_v 's non-collider parent due to the value ex_{S_v} of S_v 's exogenous ancestor in \mathbf{ex} .

Proposition 1 along with Lemma 1 are a key in our analysis because they show paths of hierarchical influence of latents on observed variables – from exogenous latents through endogenous latents (both colliders and non-colliders) to observed variables. Recognition and use of these paths of influence guides LPCC in learning LVMs.

To formalize our ideas, we introduce several concepts in Section 3.2. First, we define local influence on a single EN of its direct parents. Second, we use local influences and the BNMarkov property to generalize the influence of EX on EN . Third, exploiting the connectivity between the exogenous ancestors and their endogenous descendants, as described by Lemma 1, we focus on the influence of a specific (partial) set of exogenous variables on the values of their endogenous descendants. Analysis of the influence of all configurations \mathbf{exs} on all \mathbf{ens} and that of the configurations of specific exogenous ancestors in these \mathbf{exs} on their endogenous descendants enable learning the structure and parameters of the model and causal discovery. Finally, in Section 3.3, we show how these concepts can be exploited to learn an LVM from data clustering.

3.2 Major and minor effects and values

So far, we have analyzed the structural influences (path of hierarchies) of the latents on the observed variables. In this section, we complement this analysis with the parametric influences, which we divide into major and minor effects.

Definition 8 A local effect on an endogenous variable EN is the influence of a configuration of EN 's direct latent parents on any of EN 's values.

1. A major local effect is the largest local effect on EN_i , and it is identified by the maximal conditional probability of a specific value en_i of EN_i given a configuration \mathbf{pa}_i of EN_i 's latent parents \mathbf{Pa}_i , which is $MAE_{EN_i}(\mathbf{pa}_i) = \max_{en_i} P(EN_i = en_i | \mathbf{Pa}_i = \mathbf{pa}_i)$.
2. A minor local effect is any non-major local effect on EN_i , and it is identified by a conditional probability of any other value of EN_i given \mathbf{pa}_i , that is smaller than $MAE_{EN_i}(\mathbf{pa}_i)$. The minor local effect set, $MIESE_{EN_i}(\mathbf{pa}_i)$, comprises all such probabilities.

3. A major local value is the en_i corresponding to $MAE_{EN_i}(\mathbf{pa}_i)$, i.e., the most probable value of EN_i due to \mathbf{pa}_i , $MAV_{EN_i}(\mathbf{pa}_i) = \operatorname{argmax}_{en_i} P(EN_i = en_i | \mathbf{Pa}_i = \mathbf{pa}_i)$.
4. A minor local value is an en_i corresponding to a minor local effect, and $MIVSE_{EN_i}(\mathbf{pa}_i)$ is the set of all minor values that correspond to $MIESE_{EN_i}(\mathbf{pa}_i)$.

When EN_i is an observed variable or an endogenous latent non-collider, and thus has only a single parent Pa_i , the configuration \mathbf{pa}_i is actually the value pa_i of Pa_i .

So far, we have listed our assumptions about the structure of the model. Following is a parametric assumption:

Assumption 6 For every endogenous variable EN_i in G and every configuration \mathbf{pa}'_i of EN_i 's parents \mathbf{Pa}_i , there exists a certain value en'_i of EN_i , such that $P(EN_i = en'_i | \mathbf{Pa}_i = \mathbf{pa}'_i) > P(EN_i = en''_i | \mathbf{Pa}_i = \mathbf{pa}'_i)$ for every other value en''_i of EN_i . This assumption is related to the most probable explanation of a hypothesis given the data (Pearl, 1988).

Note that in the case that Assumption 6 is violated, in other words, if more than one value of EN_i gets the maximum probability value given a configuration of parents, LPCC still learns a model because the implementation will randomly choose a value that maximizes the probability as the most probable. However, the correctness of the algorithm is guaranteed only if all assumptions are valid; in other words, given the assumptions are valid, all causal claims made by the output graph are correct.

Proposition 2 The major local value $MAV_{EN_i}(\mathbf{pa}'_i)$ of an endogenous variable EN_i given a certain configuration of its parents \mathbf{pa}'_i is also certain.

Proof Assumption 6 guarantees that given a certain configuration \mathbf{pa}'_i of \mathbf{Pa}_i , there exists a certain value en'_i of EN_i , such that $P(EN_i = en'_i | \mathbf{Pa}_i = \mathbf{pa}'_i) > P(EN_i = en''_i | \mathbf{Pa}_i = \mathbf{pa}'_i)$ for every other value en''_i of EN_i . From the definition of a major local value, $MAV_{EN_i}(\mathbf{pa}'_i) = en'_i$. ■

We need one additional assumption about the model parameters that reflects parent-child influence in the causal model. Specifically, to identify parent-child relations, LPCC needs for each observed variable or endogenous latent non-collider to get different MAVs for different values of their latent parent. Similarly, LPCC needs a collider to get different values for each of its parents in at least two parent configurations in which this parent changes, whereas the other parents do not.

Assumption 7 First, for every EN_i that is an observed variable or an endogenous latent non-collider and for every two values pa'_i and pa''_i of Pa_i , $MAV_{EN_i}(pa'_i) \neq MAV_{EN_i}(pa''_i)$. Second, for every C_j that is a latent collider and for every $Pa_j \in \mathbf{Pa}_j$, there are at least two configurations \mathbf{pa}'_j and \mathbf{pa}''_j of \mathbf{Pa}_j in which only the value of Pa_j is different and $MAV_{C_j}(\mathbf{pa}'_j) \neq MAV_{C_j}(\mathbf{pa}''_j)$.

By aggregation over all local influences, we can now generalize these concepts through the BN parameters and Markov property from local influences on specific endogenous variables to influence on all endogenous variables in the graph.

Definition 9 An effect on EN is the influence of a configuration \mathbf{ex} of EX on EN . The effect is measured by a value configuration \mathbf{en} of EN due to \mathbf{ex} . A major effect (MAE) is the largest effect of \mathbf{ex} on EN and a minor effect (MIE) is any non-MAE effect of \mathbf{ex} on EN . Also, a major value configuration (MAV) is the configuration \mathbf{en} of EN corresponding to MAE (i.e., the most probable \mathbf{en} due to \mathbf{ex}), and a minor value configuration is a configuration \mathbf{en} corresponding to any MIE.

[Note the difference between a major effect, MAE, and a major local effect, MAE_{EN_j} , and between a major value configuration, MAV, and a major local value, MAV_{EN_j} (and similarly for the “minors”).]

Based on the proof of Proposition 1, we can quantify the effect of \mathbf{ex} on EN . For example, a major effect of \mathbf{ex} on EN can be factorized according to the product of major local effects on EN (weighted by the product of priors, $P(EX_j = ex_j)$):

$$\begin{aligned} MAE(\mathbf{ex}) &= \prod_{EX_i \in \mathbf{EX}} P(EX_i = ex_i) \prod_{C_j \in \mathbf{C}} MAE_{C_j}(\mathbf{pa}_j^{\mathbf{ex}_{C_j}}) \prod_{S_t \in \mathbf{S}} MAE_{S_t}(pa_t^{\mathbf{ex}_{S_t}}) \\ &= \prod_{OEX_m \in \mathbf{OEX}} MAE_{OEX_m}(ex_m) \prod_{OC_k \in \mathbf{OC}} MAE_{OC_k}(c_k^{\mathbf{ex}_{C_k}}) \prod_{OS_v \in \mathbf{OS}} MAE_{OS_v}(s_v^{\mathbf{ex}_{S_v}}) \\ &= \prod_{EX_i \in \mathbf{EX}} P(EX_i = ex_i) \prod_{C_j \in \mathbf{C}} \max_{c_j'} P(C_j = c_j' | \mathbf{pa}_j = \mathbf{pa}_j) \prod_{S_t \in \mathbf{S}} \max_{s_t'} P(S_t = s_t' | pa_t = pa_t) \\ &= \prod_{OEX_m \in \mathbf{OEX}} \max_{oc_k'} P(OEX_m = oc_k' | EX_m = ex_m) \prod_{OC_k \in \mathbf{OC}} \max_{oc_k'} P(OC_k = oc_k' | C_k = c_k) \\ &= \prod_{OS_v \in \mathbf{OS}} \max_{os_v'} P(OS_v = os_v' | S_v = s_v^{\mathbf{ex}_{S_v}}). \end{aligned} \quad (4)$$

A configuration \mathbf{en} of EN in which each variable in EN takes on the major local value is major or a MAV. Any effect in which at least one EN takes on a minor local effect is minor, and any configuration in which at least one EN takes on a minor local value is minor. We denote the set of all minor effects for \mathbf{ex} with $MIES(\mathbf{ex})$ (with correspondence to $MIES_{EN_j}$) and the set of all minor configurations with $MIVS(\mathbf{ex})$ (with correspondence to $MIVS_{EN_j}$).

Motivated by Lemma 1 and Proposition 1, we are interested in representing the influence on a subset of the endogenous variables of the subset of the exogenous variables that impact these endogenous variables. This partial representation of MAE will enable LPCC to recover the relationships between exogenous ancestors and only the descendants that are affected by these exogenous variables. To achieve this, we first extend the concept of effect to the concept of partial effect of specific exogenous variables and then quantify it. Later, we shall formalize all of this in Lemma 2.

Definition 10 A partial effect on a subset of endogenous variables $\mathbf{EN}' \subseteq \mathbf{EN}$ is the influence of a configuration \mathbf{ex} of \mathbf{EN}' 's exogenous ancestors $\mathbf{EX} \subseteq \mathbf{EX}$ on \mathbf{EN}' . We define a partial major effect $MAE_{\mathbf{EN}'}(\mathbf{ex})$ as the largest partial effect of \mathbf{ex} on \mathbf{EN}' and a partial minor effect $MIE_{\mathbf{EN}'}(\mathbf{ex})$ as any non- $MAE_{\mathbf{EN}'}(\mathbf{ex})$ partial effect of \mathbf{ex} on \mathbf{EN}' . A partial major value configuration $MAV_{\mathbf{EN}'}(\mathbf{ex})$ is the \mathbf{en}' of \mathbf{EN}' corresponding to $MAE_{\mathbf{EN}'}(\mathbf{ex})$; in other words, the most probable \mathbf{en}' due to \mathbf{ex} , and a partial minor value configuration is an \mathbf{en}' corresponding to any $MIE_{\mathbf{EN}'}(\mathbf{ex})$.

We are interested in representing the influence of exogenous variables on their observed descendants and all the variables in the directed paths connecting them. To do this, we separately analyze the (partial) effect of each exogenous variable on each observed variable for which the exogenous is its ancestor and all the latent variables along the path connecting these two. We distinguish between two cases (both are represented in Lemma 1): (1) Observed descendants in \mathbf{OEX} and \mathbf{OS} that are, respectively, children of exogenous latents and children of latent non-colliders that are linked to their exogenous ancestors, each via a single directed path; and (2) Observed descendants in \mathbf{OC} that are children of latent colliders and linked to their exogenous ancestors via a set of directed paths through their latent collider parents. Thus, we are interested in:

1. The partial effect of a value of exogenous ancestor EX_{NC_v} to non-collider NC_v on any configuration of the set of variables $\{TS_{NC_v} \setminus EX_{NC_v}, ON_{C_v}\}$, where ON_{C_v} is an observed child of latent non-collider NC_v , and TS_{NC_v} is the set of variables in the directed path (recall Definition 7) T_{NC_v} from EX_{NC_v} to NC_v . The corresponding $MAE_{\{TS_{NC_v} \setminus EX_{NC_v}, ON_{C_v}\}}(ex_{NC_v})$ and $MAV_{\{TS_{NC_v} \setminus EX_{NC_v}, ON_{C_v}\}}(ex_{NC_v})$ are partial major effect and partial major value configuration, respectively. For example, we may be interested in the partial effect of a value of $EX_{NC_v} = EX_{L5} = L3$ in $G5$ (Figure 1) on $\{TS_{NC_v} \setminus EX_{NC_v}, ON_{C_v}\} = \{TS_{L5} \setminus L3, X13\} = \{L4, L5, X13\}$. Note that we use here the notation NC_v since we are interested in both exogenous and endogenous latent non-colliders. When we are interested in the partial effect on an observed variable in \mathbf{OEX} , its exogenous ancestor (which is also its direct parent) is also the latent non-collider, NC_v , and the effect is not measured on any other variable but this observed variable. This is *Case 1*, which is analyzed below;

2. The partial effect of a configuration of exogenous variables \mathbf{EX}_{C_k} to collider C_k on any configuration of the set of variables $\{TS_{C_k} \setminus \mathbf{EX}_{C_k}, OC_k\}$, where OC_k is an observed child of latent collider C_k ,⁴ and TS_{C_k} is the set of variables in the set of directed paths \mathbf{T}_{C_k} from \mathbf{EX}_{C_k} to C_k . The corresponding $MAE_{\{TS_{C_k} \setminus \mathbf{EX}_{C_k}, OC_k\}}(\mathbf{ex}_{C_k})$ and $MAV_{\{TS_{C_k} \setminus \mathbf{EX}_{C_k}, OC_k\}}(\mathbf{ex}_{C_k})$ are partial major effect and partial major value configuration, respectively. For example, we may be interested in the partial effect of a configuration of $\mathbf{EX}_{C_k} = \mathbf{EX}_{L4} = \{L1, L5\}$ in $G6$ (Figure 1) on $\{TS_{C_k} \setminus \mathbf{EX}_{C_k}, OC_k\} = \{\{L1, L2, L3, L4\} \setminus \{L1, L5\}, X11\} = \{L2, L3, L4, X11\}$. This is *Case 2*, which is analyzed below.

⁴Throughout the paper, we use a child index also for its parent, e.g., OC_k 's parent is C_k , although generally, we use the index j for a collider, such as C_j .

Following, we provide detailed descriptions for these partial effects and partial values for observed children of latent non-colliders (Case 1) and observed children of latent colliders (Case 2) and formalize their properties in Propositions 3–7 to set the stage for Lemma 2.

Case 1: Observed children of latent non-colliders

If the latent non-collider NC_v is exogenous, $NC_v = EX_v$ and $ONC_v = OEX_v$, then, $\{TS_{NC_v} \setminus EX_{NC_v}, ONC_v\} = OEX_v$. Thus, the partial effect is simply the local effect, and the partial major effect is the major local effect $MAEOEX_v(ex_v)$. If the latent non-collider NC_v is endogenous, then $NC_v = S_v$ and $ONC_v = OS_v$. Then, all variables in $\{TS_{NC_v} \setminus EX_{NC_v}, OS_v\}$ are d-separated by EX_{NC_v} from $EX \setminus EX_{NC_v}$. For example, [L4, L5, X13] in G5 (Figure 1) are d-separated by L3 from L2 and its children. Thus, the effect of ex on the joint probability distribution (3) can be factored to the: a) joint probability over $EX = ex$; b) conditional probabilities of the influenced variables along a specific directed path that ends at OS_v on $EX_{NC_v} = ex_{NC_v}$ (note that the value ex_{NC_v} for all $S_i \in TS_{NC_v}$ is the same because $EX_{NC_v} = EX_v$ is the same exogenous ancestor of all latent non-colliders on the path to S_v); and c) conditional probabilities of all the remaining variables in the graph on $EX = ex$:

$$\begin{aligned} P(V|EX = ex) &= P(EX = ex)P(TS_{NC_v} \setminus EX_{NC_v}, OS_v | EX_{NC_v} = ex_{NC_v}) \\ P(V \setminus \{TS_{NC_v} \setminus EX_{NC_v}, OS_v\} | EX = ex) & \end{aligned} \quad (5)$$

in which the second factor corresponds to the partial effect of $EX_{NC_v} = ex_{NC_v}$ on $TS_{NC_v} \setminus EX_{NC_v}$ (the latent non-colliders on the path from EX_{NC_v} to S_v) and S_v 's observed child, OS_v , and the third factor corresponds to the influence of $EX = ex$ on all the other (latent and observed) variables in the graph. We can write the second factor describing the partial effect of the value ex_{NC_v} on the values of the variables $TS_{NC_v} \setminus EX_{NC_v}$ in the directed path from EX_{NC_v} to OS_v (including) as:

$$\begin{aligned} P(TS_{NC_v} \setminus EX_{NC_v}, OS_v | EX_{NC_v} = ex_{NC_v}) &= \\ \prod_{S_i \in \{TS_{NC_v} \setminus EX_{NC_v}\}} P(S_i = s_i | P_{a_i} = pa_i^{ex_{NC_v}}) P(OS_v = os_v | S_v = s_v^{ex_{NC_v}}) & \end{aligned} \quad (6)$$

The partial major effect in (4) for this directed path can be written as (note again that $ex_{NC_v} = ex_{S_v}$):

$$\begin{aligned} MAE_{TS_{NC_v} \setminus EX_{NC_v}, OS_v}(ex_{NC_v}) &= MAE_{TS_{NC_v} \setminus EX_{NC_v}}(ex_{NC_v}) \cdot MAE_{OS_v}(s_v^{ex_{NC_v}}) = \\ \prod_{S_i \in \{TS_{NC_v} \setminus EX_{NC_v}\}} MAE_{S_i}(pa_i^{ex_{NC_v}}) \cdot MAE_{OS_v}(s_v^{ex_{NC_v}}) & \end{aligned} \quad (7)$$

Proposition 3 The MAV $\{TS_{NC_v} \setminus EX_{NC_v}, ONC_v\}(ex_{NC_v})$ corresponding to

$MAE_{TS_{NC_v} \setminus EX_{NC_v}, ONC_v}(ex_{NC_v})$ is a certain value configuration for each certain value ex_{NC_v} .

(Note that here we use the notation NC_v rather than S_v since the proposition applies to both exogenous and endogenous latent non-colliders.)

Proposition 4 All corresponding values in $MAV_{TS_{NC_v} \setminus EX_{NC_v}, ONC_v}(ex'_{NC_v})$ and $MAV_{TS_{NC_v} \setminus EX_{NC_v}, ONC_v}(ex''_{NC_v})$ for two values ex'_{NC_v} and ex''_{NC_v} of EX_{NC_v} are different.

(Here also we use the notation NC_v , since the proposition applies to both exogenous and endogenous latent non-colliders.)

So far, we have analyzed the impact of an exogenous variable on a latent non-collider by ‘‘propagating’’ the exogenous (source) impact along the path to the latent non-collider (sink). Propositions 3 and 4, respectively, guarantee that a certain value of the exogenous variable is responsible for a certain value of the latent non-collider and different values of the exogenous are echoed through different values of the latent non-collider. Proposition 4 is based on the correspondence between changes in values of a latent non-collider and changes in values of its parent; a correspondence that is guaranteed by Assumption 7 (first part). Propositions 3 and 4, respectively, ensure the existence and uniqueness of the value a latent non-collider gets under the influence of an exogenous ancestor: one (Proposition 3) and only one (Proposition 4) value of the latent non-collider changes with a change in the value of the exogenous. We formalize this in the following Proposition 5.

Proposition 5 EX_{NC_v} changes values (i.e., has two values ex'_{NC_v} and ex''_{NC_v}) if and only if NC_v changes values in the two corresponding major value configurations:

$$MAV_{TS_{NC_v} \setminus EX_{NC_v}, ONC_v}(ex'_{NC_v}) \text{ and } MAV_{TS_{NC_v} \setminus EX_{NC_v}, ONC_v}(ex''_{NC_v}).$$

Case 2: Observed children of latent colliders

In the case of an observed variable OC_k that is a child of a latent collider C_k , all variables in $\{TS_{C_k} \setminus EX_{C_k}, OC_k\}$ are d-separated by EX_{C_k} from $EX \setminus EX_{C_k}$. Thus, the effect of ex on the joint probability distribution (3) can be factored (similarly to Case 1) to the: a) joint probability over $EX = ex$; b) conditional probabilities of the influenced variables along all directed paths that end at OC_k on $EX_{C_k} = ex_{C_k}$ (note that all variables along each directed path T_{C_k} are influenced by the same ex_{C_k}); and c) conditional probabilities of all the remaining variables in the graph on $EX = ex$:

$$\begin{aligned} P(V|EX = ex) &= P(EX = ex)P(TS_{C_k} \setminus EX_{C_k}, OC_k | EX_{C_k} = ex_{C_k}) \\ P(V \setminus \{TS_{C_k} \setminus EX_{C_k}, OC_k\} | EX = ex) & \end{aligned} \quad (8)$$

in which the second factor corresponds to the partial effect on $\{TS_{C_k} \setminus EX_{C_k}, OC_k\}$ of EX_{C_k} and the third factor corresponds to the partial effect on all variables other than $\{TS_{C_k} \setminus EX_{C_k}, OC_k\}$. We can decompose the second factor into a product of: a) a product over all directed paths into C_k of a product of partial effects over all variables (excluding C_k) in such a path; b) the partial effect on C_k ; and c) the partial effect on its child OC_k :

$$\begin{aligned} P(TS_{C_k} \setminus EX_{C_k}, OC_k | EX_{C_k} = ex_{C_k}) &= \\ \prod_{T_{C_k} \in \{TS_{C_k} \setminus EX_{C_k}\}} \prod_{S_i \in T_{C_k}} P(S_i = s_i | P_{a_i} = pa_i^{ex_{C_k}}) P(C_k = c_k | P_{a_k} = pa_k^{ex_{C_k}}) P(OC_k = oc_k | C_k = c_k^{ex_{C_k}}), & \end{aligned} \quad (9)$$

This factor can be rewritten as:

$$P(\mathbf{TS}_{C_k} \setminus \mathbf{EX}_{C_k}, \mathbf{OC}_k | \mathbf{EX}_{C_k} = \mathbf{ex}_{C_k}) = \prod_{\mathbf{TS}_{C_k} \in \mathbf{TS}_{C_k}} P(\{\mathbf{TS}_{C_k} \setminus \mathbf{EX}_{C_k}, C_k\} | \mathbf{EX}_{C_k} = \mathbf{ex}_{C_k}) P(C_k = c_k | \mathbf{pa}_k = \mathbf{pa}_k) P(\mathbf{OC}_k = oc_k | C_k = c_k, \mathbf{ex}_{C_k}). \quad (10)$$

It reflects the partial effects of a configuration \mathbf{ex}_{C_k} on the values of the variables in $\{\mathbf{TS}_{C_k} \setminus \mathbf{EX}_{C_k}\}$ and the values C_k and \mathbf{OC}_k get, and thus the partial major effect of the second factor can be represented as:

$$MAE_{(\mathbf{TS}_{C_k} \setminus \mathbf{EX}_{C_k}, \mathbf{OC}_k)}(\mathbf{ex}_{C_k}) = \prod_{\mathbf{TS}_{C_k} \in \mathbf{TS}_{C_k}} MAE_{(\mathbf{TS}_{C_k} \setminus \mathbf{EX}_{C_k}, C_k)}(\mathbf{ex}_{C_k}) MAE_{C_k}(\mathbf{pa}_k) MAE_{\mathbf{OC}_k}(c_k, \mathbf{ex}_{C_k}). \quad (11)$$

Proposition 6 The $MAE_{(\mathbf{TS}_{C_k} \setminus \mathbf{EX}_{C_k}, \mathbf{OC}_k)}(\mathbf{ex}_{C_k})$ corresponding to $MAE_{(\mathbf{TS}_{C_k} \setminus \mathbf{EX}_{C_k}, \mathbf{OC}_k)}(\mathbf{ex}_{C_k})$ is a certain value configuration for each certain value configuration \mathbf{ex}_{C_k} .

We wish to apply the same mechanism as in Case 1 to analyze the impact of more than a single exogenous ancestor on a latent collider, but here the impact is propagated toward the collider along more than a single path. To accomplish this, the following Proposition 7 analyzes the effect on a collider of each of its exogenous ancestors by considering the effect of such an exogenous on the corresponding collider's parent (using Proposition 5, similar to Case 1 for a latent non-collider) and then the effect of this parent on the collider itself (using the second part of Assumption 7).

Proposition 7 For every exogenous ancestor $EX_{C_k} \in \mathbf{EX}_{C_k}$ of a latent collider C_k , there are at least two configurations \mathbf{ex}'_{C_k} and \mathbf{ex}''_{C_k} of \mathbf{EX}_{C_k} in which only EX_{C_k} changes values when C_k changes values in the two corresponding major value configurations $MAV_{(\mathbf{TS}_{C_k} \setminus \mathbf{EX}_{C_k}, \mathbf{OC}_k)}(\mathbf{ex}'_{C_k})$ and $MAV_{(\mathbf{TS}_{C_k} \setminus \mathbf{EX}_{C_k}, \mathbf{OC}_k)}(\mathbf{ex}''_{C_k})$.

Lemma 2

1. A latent non-collider NC_v and its observed child ONC_v , both descendants of an exogenous variable EX_{NC_v} , change their values in any two major configurations if and only if EX_{NC_v} has changed its value in the corresponding two configurations of \mathbf{EX} .
2. A latent collider C_k and its observed child OC_k , both descendants of a set of exogenous variables \mathbf{EX}_{C_k} , change their values in any two major configurations only if at least one of the exogenous variables in \mathbf{EX}_{C_k} has changed its value in the corresponding two configurations of \mathbf{EX} .

3.3 PCC by clustering observational data

Practically, we use observational data that were generated from an unknown LVM and measured over the observed variables. Proposition 1 showed us that each configuration

of observed variables (which is part of a configuration of the endogenous variables) and their joint probability is a result of the assignment of a configuration \mathbf{ex} to the exogenous variables \mathbf{EX} . Therefore, we define:

Definition 11 An observed value configuration, observed major value configuration, and observed minor value configuration due to \mathbf{ex} are the parts in \mathbf{en} , MAV , and a minor value configuration, respectively, that correspond to the observed variables.

The following two propositions formalize the relationships between the observed major value configurations and the set of possible \mathbf{ex} .

Proposition 8 There is only a single observed major value configuration to each exogenous configuration \mathbf{ex} of \mathbf{EX} .

Proof Based on Lemma 2, different observed major value configurations can be obtained if and only if there is more than a single exogenous configuration \mathbf{ex} of \mathbf{EX} . Thus, an exogenous configuration \mathbf{ex} can only lead to a single observed major value configuration. ■

Proposition 9 There are different observed major value configurations to different exogenous configurations \mathbf{ex} .

Proof Assume for the sake of contradiction that two different value configurations \mathbf{ex}_1 and \mathbf{ex}_2 led to the same observed major value configuration. Because the two configurations are different, there is at least one exogenous variable EX' that has different values in \mathbf{ex}_1 and \mathbf{ex}_2 . Since based on Assumption 4, EX' has at least two observed children, then, based on Assumption 7, each of these children has different values in the two observed major value configurations due to the different value of EX' in \mathbf{ex}_1 and \mathbf{ex}_2 . This is contrary to our assumption that there is only one observed major value configuration. ■

Due to the probabilistic nature of \mathbf{BN} , each observed value configuration due to \mathbf{ex} may be represented by several data points. Clustering these data points may produce several clusters for each \mathbf{ex} and each cluster corresponds to another observed value configuration. Based on Propositions 8 and 9, one and only one of the clusters corresponds to each of the observed major value configurations, whereas the other clusters correspond to observed minor value configurations. We distinguish between these clusters using Definition 12.

Definition 12 The single cluster that corresponds to the observed major value configuration, and thus also represents the major effect $MAE(\mathbf{ex})$ due to configuration \mathbf{ex} of \mathbf{EX} , is the major cluster for \mathbf{ex} , and all the clusters that correspond to the observed minor value configurations due to minor effects in $MIES(\mathbf{ex})$ are minor clusters.

To resolve between different types of minor effects/clusters, we make two definitions.

Definition 13 A k -order minor effect is a minor effect in which exactly k endogenous variables in \mathbf{EN} correspond to minor local effects. An \mathbf{en} corresponding to a k -order minor effect is a k -order minor value configuration.

Definition 14 *Minor clusters that correspond to k -order minor effects are k -order minor clusters.*

Based on Proposition 9 and Definition 12, the set of all major clusters (corresponding to all observed major value configurations) reflects the effect of all possible **ex**s, and thus the number of major clusters is expected to be equal to the number of **EX** configurations. Therefore, the identification of all major clusters is a key to the discovery of exogenous variables and their causal interrelations. For this purpose, we introduce the concept of *pairwise cluster comparison* (PCC); PCC measures the differences between two clusters; each represents the response of LVM to another **ex**.

Definition 15 *Pairwise cluster comparison is a procedure by which pairs of clusters are compared, for example through a comparison of their centroids. The result of PCC between a pair of cluster centroids of dimension $|\mathbf{O}|$, where \mathbf{O} is the set of observed variables, can be represented by a binary vector of size $|\mathbf{O}|$ in which each element is 1 or 0 depending, respectively, on whether or not there is a difference between the corresponding elements in the compared centroids.*

When PCC is between clusters that represent observed major value configurations (i.e., PCC between major clusters), an element of 1 identifies an observed variable that has changed its value between the compared clusters due to a change in **ex**. Thus, the 1s in a major–major PCC provide evidence of causal relationships between **EX** and **O**. Practically, LPCC always identifies all observed variables that are represented by 1s together in all PCCs as the observed descendants of the same exogenous variable (Section 4.1). However, due to the probabilistic nature of BN and the existence of endogenous latents (mediating the connections from **EX** to **O**), some of the clusters are k -order minor clusters (in different orders), representing k -order minor configurations/effects. Minor clusters are more difficult to identify than major clusters because the latter reflect the major effects of **EX** on **EN** and, therefore, are considerably more populated by data points than the former. Nevertheless, minor clusters are important in causal discovery by LPCC even though a major–minor PCC cannot tell the effect of **EX** on **EN** because an observed variable in two compared (major and minor) clusters should not necessarily change its value as a result of a change in **ex**. Their importance is because a major cluster, which is a zero-order minor value configuration and thus has zero minor values, cannot indicate (when compared with another major cluster) the existence of minor values. On the contrary, PCC between major and minor clusters shows (through the number of 1s) the number of minor values represented in the minor cluster, and this is exploited by LPCC for identifying the endogenous latents and interrelations among them (Section 4.4). That is, PCC is the source to identify causal relationships in the unknown LVM; major–major PCCs are used for identifying the exogenous variables and their descendants; and major–minor PCCs are used for identifying the endogenous latents, their interrelations, and their observed children.

4. Overview of the LPCC concept⁵

Let us demonstrate the relations between clustering results and learning an LVM using LPCC through an example. G1 in Figure 1 shows a model having two exogenous variables,

⁵Preliminary versions of the PCC concept and LPCC algorithm are given in Asbeh and Lerner (2012).

L1 and L2, each having three children X1, X2, X3 and X4, X5, X6, respectively. ⁶ For the example, let us assume that all variables are binary,⁷ i.e., L1 and L2 have four possible **ex**s (L1L2 = 00, 01, 10, 11). First, we generated a synthetic data set of 1,000 patterns from G1 over the six observed variables. We used a uniform distribution over L1 and L2 and set the probabilities of an observed child, $X_i, i = 1, \dots, 6$, given its latent parent, $L_k, k = 1, 2$ (only if L_k is a direct parent of X_i , e.g., L1 and X1), to be $P(X_i = v | L_k = v) = 0.8, v = 0, 1$. Second, using the self-organizing map (SOM) (Kohonen, 1997), we clustered the data set and found 16 clusters, of which four were major (see Section 4.3 for details on how to identify major clusters). This meets our expectation of four major clusters corresponding to the four possible **ex**s. These clusters are presented in Table 1a by their centroids, which are the most prevalent patterns in the clusters, and in Table 1b by their PCCs. For example, PCC1,2, comparing clusters C1 and C2, shows that when moving from C1 to C2, only the values corresponding to variables X1, X2, and X3 have been changed (i.e., $\delta X_1 = \delta X_2 = \delta X_3 = 1$ in Table 1b). Lemma 2 guarantees that the three variables are descendants of the same **EX** that changed its value between two **ex**s represented by C1 and C2. PCC1,4, PCC2,3, and PCC3,4 reinforce this conclusion. Indeed, we know from the true graph, G1, that this **EX** is latent L1. A similar conclusion can be deduced about X4, X5, and X6 as descendants of an exogenous latent, which we know, based on the true graph, is L2.

Centroid	X1	X2	X3	X4	X5	X6
C1	0	0	1	1	1	1
C2	1	1	1	1	1	1
C3	0	0	0	0	0	0
C4	1	1	1	0	0	0

PCC	δX_1	δX_2	δX_3	δX_4	δX_5	δX_6
PCC1,2	1	1	1	0	0	0
PCC1,3	0	0	0	1	1	1
PCC1,4	1	1	1	1	1	1
PCC2,3	1	1	1	1	1	1
PCC2,4	0	0	0	1	1	1
PCC3,4	1	1	1	1	0	0

Table 1: (a) Centroids of major clusters for G1 and (b) PCCs between these major clusters

LPCC is fed by data that is sampled from the observed variables in the unknown model. LPCC clusters the data using SOM (although any other clustering algorithm is good as well) and selects an initial set of major clusters (Section 4.3). Then, LPCC learns LVM in two stages. In the first stage, LPCC first identifies exogenous latent variables and latent colliders (without distinguishing them yet) and their corresponding observed descendants (Section 4.1) before distinguishing them (Section 4.2). LPCC iteratively improves the selection of the major clusters (Section 4.3), and the entire stage is repeated until convergence. In the second stage, LPCC identifies endogenous latent non-colliders with their children. Because this stage cannot distinguish from the outset between latent non-colliders and their latent ancestors, LPCC also needs to apply a mechanism to split these two types of latent variables from each other and to find the links between them after the split (Section 4.4). A flowchart of the LPCC algorithm is given in Figure 2.

⁶We remind that we determined three indicators per latent in all true models we demonstrate their learning (Figure 1) because BPC requires three indicators per latent to identify that latent; which makes the experimental evaluation we did in Part II of the paper fair.

⁷This is only for demonstration purposes. Part II of the paper shows evaluation results also for ternary latent variables and observed variables of different dimensions.

4.1 Identification of exogenous latent variables and latent colliders and their descendants

Table 1b shows that $PCC1,2$ (and $PCC3,4$) provides evidence that $X1, X2$, and $X3$ may be descendants of the same exogenous latent ($L1$, as we know) that has changed its value between the two exs represented by $C1$ and $C2$ (and $C3$ and $C4$). Relying only on one PCC may be inadequate when concluding that these variables are descendants of the same exogenous latent because there may be other exogenous latents that have changed their values too. Table 1b shows that $PCC2,3$ (and $PCC1,4$) provides the same evidence about $X1, X2$, and $X3$. But, $PCC2,3$ and $PCC1,4$ also show that the values corresponding to $X4, X5$, and $X6$ have been changed together too, whereas these values did not change in $PCC1,2$ and $PCC3,4$. Does this mean that $X4, X5$, and $X6$ are also descendants of the same latent ancestor of $X1, X2$, and $X3$? If we combine the two pieces of evidence provided by, e.g., $PCC1,2$ and $PCC2,3$, we can answer this question with a “no”. This is because $X4, X5$, and $X6$ changed their values only in $PCC2,3$ but not in $PCC1,2$, and thus they cannot be descendants of $L1$. This insight strengthens the evidence that $X1, X2$, and $X3$ are the only descendants of $L1$. A similar analysis using $PCC1,3$ and $PCC2,4$ will identify that $X4, X5$, and $X6$ are descendants of another latent variable ($L2$, as we know). Therefore, we define:

Definition 16 A maximal set of observed (MSO) variables is the set of variables that always changes its values together in each major-major PCC in which at least one of the variables changes value.

That is, there is a particular interest in identifying the MSOs that always change their values together in each major-major PCC in which at least one of the variables changes value. For example, $X1$ (Table 1) changes its value in $PCC1,2, PCC1,4, PCC2,3$, and $PCC3,4$ and always together with $X2$ and $X3$ (and vice versa). Thus $\{X1, X2, X3\}$ (and similarly $\{X4, X5, X6\}$) is an MSO. Each MSO includes descendants of the same exogenous latent variable L , and after considering all PCCs, LPCC identifies an MSO for each exogenous latent variable.

Based on any identified MSO, LPCC introduces to the learned graph a new latent variable L together with all the observed variables that are included in this MSO as its children. At this stage, LPCC cannot yet distinguish between exogenous latents and latent colliders since the main goal at this stage is to identify latent variables. For now, LPCC focuses on the identification of the relations between the latents and the observed variables, but not on the identification of the interrelations between the latents. The latter task that is needed for distinguishing the latent colliders from the exogenous latents is performed in a further step (Section 4.2). Note, however, that the identification of endogenous latent non-colliders needs a different analysis that is based on major-minor PCCs and not on major-major PCCs, and thus it is described separately in Section 4.4.

The following Theorem 1 helps us formalize this identification step. For this theorem, we also need Definition 17 of equivalence relation/classes from set theory and Lemma 3, which is important by itself and for better understanding of LPCC, but also for proving Theorem 1.

Definition 17 A given binary relation (i.e., between two elements) \sim on a set A is said to be an equivalence relation if and only if it is reflexive ($a \sim a$), symmetric (if $a \sim b$ then $b \sim a$), and

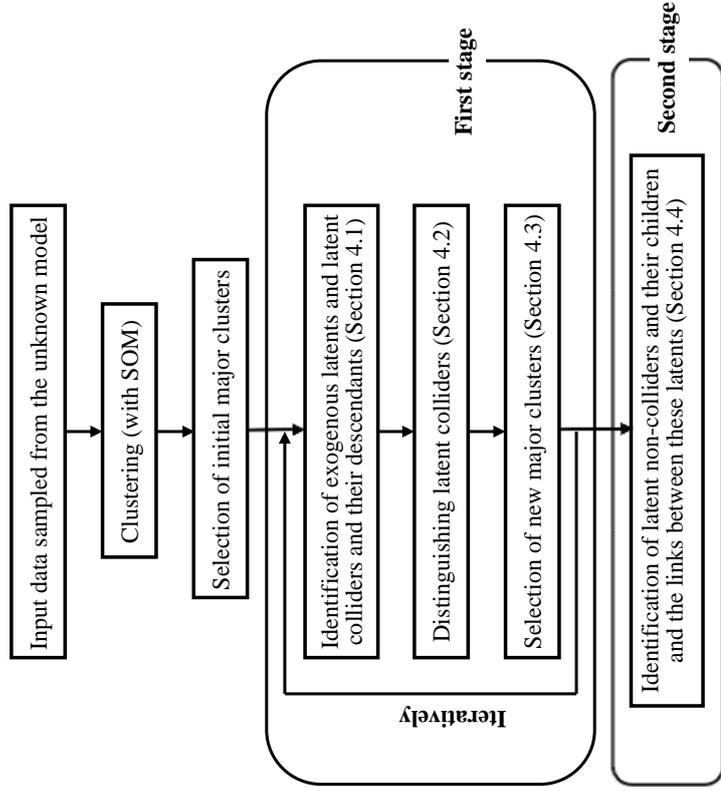


Figure 2: An overview of the LPCC algorithm.

transitive (if $a \sim b$ and $b \sim c$) for all a, b , and c in A . The equivalence class of a under \sim , denoted $[a]$, is defined as: $[a] = \{b \in A \mid b \sim a\}$ (Erderton, 1977).

Note that every two equivalence classes are either equal or disjoint. Therefore, the set of all equivalence classes of A forms a partition of A : every element of A belongs to one and only one equivalence class. It follows from the properties of an equivalence relation that: $a \sim b$ if and only if $|a| = |b|$. The following Lemma 3 is important since it shows that each **MSO** is an equivalence class, and thus **MSOs** corresponding to the learned latents are disjoint. At this stage, LPCC learns a set of at least two observed variables corresponding to a specific **MSO** for each latent where none of the observed variables is shared with other **MSOs** for other latents; in other words, a pure measurement model.

Lemma 3 *The relation “always changes together with” on the set O of all observed variables, such as “variable $O_j \in O$ always changes together with variable $O_j \in O$ in each PCC in which either O_i or O_j has changed” is an equivalence relation. Each equivalence class for this relation comprises an **MSO**.*

Proof All three conditions that are required for a binary relation to become equivalence are met:

1. O_j always changes with O_j (trivial).
2. If O_i always changes with O_j , then O_j always changes with O_i .
3. If O_j always changes with O_j , and O_j always changes with O_k , then O_i always changes with O_k .

Thus, the set of observed variables in a model can be represented by a set of equivalence classes for this relation, where each equivalence class includes all the variables that have the same equivalence relation, such as an **MSO**. ■

Theorem 1 *Variables of a particular **MSO** are children of a particular exogenous latent variable **EX** or its latent non-collider descendant or children of a particular latent collider **C**.*

Note that Theorem 1 guarantees that each of multiple latent variables (either an exogenous or any of its non-collider descendants or a collider) is identified by its own **MSO**, regardless of the latent cardinality.

4.2 Distinguishing latent collider variables

After identifying the exogenous latents and latent colliders together (Section 4.1), we need now to separate them. To demonstrate our concept for distinguishing latent colliders, we use graph G2 in Figure 1, which shows two exogenous latent variables, L1 and L3, that collide in one endogenous latent variable, L2. We assume that all latent variables are binary,⁸ and each has three binary observed children X1, X2, and X3 (L1), X4, X5, and X6

⁸See footnote 7.

(L2), and X7, X8, and X9 (L3). Having two exogenous binary variables, we expect to find four major clusters in the data generated from G2. Each cluster will correspond to one of the four possible **ex**s (L1L3= 00, 01, 10, 11). In this case, as for G1 that was analyzed in the introduction to Section 4, we expect the values of X1, X2, and X3 to change together in all the PCCs following a change in the value of L1, and the values of X7, X8, and X9 to change together in all the PCCs following a change in the value of L3. However, the values of X4, X5, and X6 will change together with those of X1, X2, and X3 in part of the PCCs and together with those of X7, X8, and X9 in the remaining PCCs, but always together in all of the PCCs. This will be evidence that X4, X5, and X6 are descendants of the same latent variable (L2, as we know), which is a collider of L1 and L3.

So far, LPCC learned latent variables but could not distinguish between exogenous latents and latent colliders (learning latent non-colliders will be described in Section 4.4). To learn that an already learned latent variable L is a collider for a set of other already learned (exogenous) latent ancestor variables **LACEX**, LPCC requires that: (1) The values of the children of L will change with the values of descendants of different latent variables in **LA** in different parts of major-major PCCs; and (2) The values of the children of L will not change in any PCC unless the values of descendants of at least one of the variables in **LA** change. This insures that L does not change independently of latents in **LA** that are L 's ancestors. We formalize this identification step in Theorem 2:

Theorem 2 *A latent variable L is a collider of a set of latent ancestors **LACEX** only if:*

1. *The values of the children of L change in different parts of some major-major PCCs each time with the values of descendants of another latent ancestor in **LA**; and*
2. *The values of the children of L do not change in any PCC unless the values of descendants of at least one of the variables in **LA** change too.*

4.3 Strategy for choosing major clusters

In this problem of unsupervised identification of latent variables given only observational data, LPCC has to deal with a lack of prior information regarding the distribution of each latent variable. Therefore, in its first iteration, LPCC assumes a uniform distribution over the latents and selects the major clusters based only on cluster size, which is the number of patterns clustered by the cluster. Clusters that are larger than the average cluster size are selected as majors. However, this initial selection may generate false negative errors (i.e., deciding a major cluster is minor). This may happen when a latent variable L has a skewed distribution over its values due to a low probability of L to take on any of its rare values. Then, the value configuration **ex** for which $L=v$, where v is a rare value, will be represented only by small clusters that could not be chosen as majors, although at least one of them should be major in representing v .

In addition, the initial selection may perform a false positive error (i.e., deciding a minor cluster is major), e.g., as a result of a very weak influence of L on any of its children (observed variables) X_i . In the discrete case, this weak influence can be represented as (almost) equal conditional probabilities of an observed variable to take on two different values $v_1 \neq v_2$ given the same value v of its latent parent, $P(X_i=v_1 \mid L=v) \cong P(X_i=v_2 \mid L=v)$. This may lead to splitting a data cluster that represents a configuration in which $L=v$ into

two clusters with almost the same size, and, when enough samples exist in both clusters, accepting both as major clusters instead of only one. For example, consider $G1$ in Figure 1, where all the variables are binary. Suppose that $P(X_2 = 0 | L1 = 0) = 0.6$ and $P(X_2 = 1 | L1 = 0) = 0.4$. This may split the cluster representing the configuration $L1L2=00$ into two clusters; in the first cluster $X_2 = 0$ and in the second cluster $X_2 = 1$. Due to the similar probabilities, both clusters may have approximately the same size, and if enough samples exist for $L1L2=00$ these two clusters may be larger than the average cluster size. Therefore, both may be accepted as major clusters in the initial selection. Recall that each ex should be represented by a single major cluster, which is the cluster that reflects the major effect of ex on the observed variables. In the example, only the cluster in which $X_2 = 0$ should be a major cluster, but due to the similar probabilities a false positive error could occur by also accepting the cluster in which $X_2 = 1$ as major.

To avoid these possible errors due to skewed data and circumstances that undermine identifiability, LPCC decides on major clusters iteratively. After learning a graph based on the initial selection of major clusters based on their sizes, it becomes possible to learn the cardinalities of the latent variables and consequently to find all possible exs (Section 4.1). Then, for each ex , we can select the most probable cluster given the data and use it as an update to the major cluster that represents this ex . Using an EM-style procedure (Dempster et al., 1977), the set of major clusters can be updated iteratively and probabilistically and augment LPCC to learn more accurate graphs (see Section 2.1 in Part II for more details). This process can be repeated until convergence to a final graph (Figure 2). Since the final graph depends on the initial graph, the iterative approach cannot guarantee finding the optimal model, but only improving the initial graph.

4.4 Identification of latent non-collider variables

So far (Section 4.1), based on major-major PCCs, all the endogenous latent non-colliders that are descendants of an exogenous variable EX were temporarily combined with EX , and all the observed children of these latent non-colliders were temporarily combined with the direct children of EX . Thus, to identify latent non-colliders, LPCC needs to split them from their previously learned ancestor together with their observed children. We suggest that this identification stage be based on major-minor PCCs (recall that latent colliders were already identified separately, as described in Section 4.2).

To exemplify this need, let us observe $G3$ in Figure 1, which shows a serial connection of three latent variables $L1$, $L2$, and $L3$. Assume each of the latents is binary and has three binary observed children. $L1$ is the only EX with two possible exs ($L1=0, 1$), and $L2$ and $L3$ are NCs; $L2$ is a child of $L1$ and a parent of $L3$. We synthetically generated a random data set of 1,000 patterns from $G3$ over the nine observed variables. We set the probabilities of: 1) $L1$ uniformly; 2) an observed child X_i , $i = 1, \dots, 9$, given its latent parent L_k , $k = 1, 2, 3$ (only if L_k is a direct parent of X_i , e.g., $L1$ and $X1$), as $P(X_i = v | L_k = v) = 0.8, v = 0, 1$; and 3) an endogenous latent L_j , $j = 2, 3$, given its latent parent L_k , $k = 1, 2$ (only if L_k is a direct parent of L_j , e.g., $L1$ and $L2$), as $P(L_j = v | L_k = v) = 0.8, v = 0, 1$. Table 2 presents the seventeen largest clusters using their centroids and sizes, from which $C1$ and $C2$ were selected as major clusters (initially, $C1-C6$ were selected, because they are larger than the average cluster size of 21, but then the iterative strategy described in Section 4.3 left only $C1$ and $C2$ as

major clusters). This meets our expectation of two major clusters corresponding to the two possible exs of $L1$. However, because all the elements in $PCC1,2$ are 1s (compare $C1$ and $C2$ in Table 2), the nine observed variables establish a single MSO and by Theorem 1 are considered descendants of the same exogenous variable. That is, the model $G0$ learned in the first phase of LPCC has only one exogenous latent variable (i.e., $L1$), and all of the nine observed descendants are learned as its direct children, which is contrary to $G3$. Since $L2$ and $L3$, which are latent non-colliders that are descendants of $L1$ in $G3$, were combined in $G0$ with $L1$, LPCC should split them from $L1$ along with their observed children in order to learn the true graph.

Thus, in the second phase, LPCC tests the assumption that $G0$ is true. If the assumption is rejected, LPCC infers that an exogenous latent EX has latent non-collider descendants, which were temporarily joined to EX in the first phase, and hence splits them from EX . To be able to reject the assumption about the correctness of $G0$, and thereby identify a possible split of an exogenous latent EX , we first define a first-order minor cluster (1-MC).

Centroid	X1	X2	X3	X4	X5	X6	X7	X8	X9	size
$C1$	1	1	1	1	1	1	1	1	1	49
$C2$	0	0	0	0	0	0	0	0	0	47
$C3$	1	1	1	1	1	1	1	1	0	28
$C4$	0	0	0	0	0	0	0	1	0	24
$C5$	0	1	0	0	0	0	0	0	0	22
$C6$	1	1	1	1	1	1	0	0	0	22
$C7$	0	0	1	0	0	0	0	0	0	21
$C8$	0	0	0	1	1	1	1	1	1	19
$C9$	0	0	0	0	0	0	1	1	1	18
$C10$	1	1	1	0	0	0	0	0	0	16
$C11$	0	0	0	1	0	0	0	0	0	14
$C12$	0	0	0	0	0	0	1	0	0	14
$C13$	1	0	1	1	1	1	1	1	1	14
$C14$	1	1	1	0	1	1	1	1	1	14
$C15$	1	0	0	0	0	0	0	0	0	13
$C16$	1	1	1	1	1	1	0	1	1	12
$C17$	0	0	0	0	0	1	0	0	0	12

Table 2: The seventeen largest clusters for $G3$ represented by their centroids and sizes

A 1-MC is a cluster that corresponds to a 1-order minor value configuration (Definitions 13 and 14), which exists when exactly one endogenous variable in EN (either latent or observed) has a minor local value (Definition 13) as a response to a value $ex \in ex$ that $EX \in EX$ has obtained. By analyzing, for each exogenous EX , PCCs between 1-MCs and the major clusters that identified EX , LPCC reveals the existence of the latent non-colliders that were previously combined with EX (Section 4.1). Following that, LPCC splits these non-colliders from EX . We will show that if only one observed variable changes in such PCCs (e.g., $X9$ in $PCC1,3$ in Table 3; $C1$ is major and $C3$ is 1-MC) as a response to ex , then the minor value in the 1-MC is of an observed descendant of EX . And, if two or more observed variables change in such PCCs (e.g., $X7-X9$ in $PCC1,6$ in Table 4; $C1$ is major and $C6$ is 1-MC) as a response to ex , then the minor value in the 1-MC is due to a minor value of a

latent non-collider descendant of EX . Thus, PCCs between 1-MCs and major clusters that show a change in the values of two or more observed variables provide evidence of the existence of an NC that should be split from its exogenous ancestor. Following, we describe how LPCC finds the set of 1-MCs. Then, we elaborate why and how the analysis of the PCCs between 1-MCs and major clusters is used to identify and split latent non-colliders from their exogenous ancestor.

PCC	$\delta X1$	$\delta X2$	$\delta X3$	$\delta X4$	$\delta X5$	$\delta X6$	$\delta X7$	$\delta X8$	$\delta X9$
$PCC1,3$	0	0	0	0	0	0	0	0	1
$PCC2,3$	1	1	1	1	1	1	1	1	0

Table 3: PCCs for $C3$ with $C1$ and $C2$ (Table 2) in learning $G3$

PCC	$\delta X1$	$\delta X2$	$\delta X3$	$\delta X4$	$\delta X5$	$\delta X6$	$\delta X7$	$\delta X8$	$\delta X9$
$PCC1,6$	0	0	0	0	0	0	1	1	1
$PCC2,6$	1	1	1	1	1	1	0	0	0
$PCC1,8$	1	1	1	0	0	0	0	0	0
$PCC2,8$	0	0	0	1	1	1	1	1	1
$PCC1,9$	1	1	1	1	1	1	0	0	0
$PCC2,9$	0	0	0	0	0	0	1	1	1
$PCC1,10$	0	0	0	1	1	1	1	1	1
$PCC2,10$	1	1	1	0	0	0	0	0	0

Table 4: All 25-PCCs for $G3$

To find the set of 1-MCs, LPCC first calculates a threshold on the maximal size of 2-order minor clusters (2-MCs). This threshold represents the maximal size of a minor cluster that corresponds to a 2-order minor value configuration, i.e., a minor cluster that represents exactly two endogenous variables in EN that have minor values (Definition 13). This threshold is an approximation for the maximal probability of having minor values as a response to any ex in exactly two descendants of EX , where all other descendants of EX in EN have major values. This approximation is derived from the product of the maximal minor local effects (Definition B.1 in Appendix B) of two observed descendants of EX and the maximal major local effects (Definition B.1) of the other observed descendants in EN (Appendix B). Thus, the sizes of all 1-MCs lie between the maximal size of a 2-MC (i.e., the threshold) and the minimal size of a major cluster (note that a major cluster is also a zero-order minor cluster corresponding to a zero-order minor value configuration). For example, based on the analysis above, $C2$ is the minimal major cluster in learning $G3$, and all the fifteen clusters (Table 2) that are smaller than $C2$ and larger than the threshold (calculated as 11), i.e., $C3$ - $C17$, are 1-MCs. Note that this procedure is separately applied to each $EX \in EX$. That is, for each EX , there is a different set of 1-MCs, each representing a single minor value of a descendant of EX and used to identify this descendant, whereas the other descendants of EX have major values.

Recall that every 1-MC corresponds to a 1-order minor value configuration that is due to exactly a single minor value of either an observed variable O or a latent non-collider NC , where both O and NC are descendants of EX in EN . The main difference between

these two cases is that in the former, the minor value in O is reflected only in this value, whereas in the latter, the minor value in NC may affect the values of all descendant latents of NC together with those of all the direct children (observed variables) of NC and its descendant latents. A minor value in O is identified based on the probability of this value conditioned on a certain value of O 's direct parent that is smaller than the maximal probability achieved for another value of O (i.e., the major value) conditioned on the same value of O 's direct parent. This happens for each value of the direct parent and does not require a change in EX to happen. From definition, a minor value in O in a 1-order minor value configuration can only happen when all EX 's descendants, except O , obtain major values. Although the mechanism of obtaining a minor value in a latent descendant NC of EX is similar to that in O , the impact of such a minor value is not locally restricted to NC , as for O , but it simultaneously affects all the descendants (latent and observed) of NC , which again, from definition, obtain major values.

We are only interested in the second case of minor values of NC , because their identification helps split this NC from its ancestor EX to which it was initially combined (Section 4.1). Since the observed variables in both cases are among EX 's descendants, which were already used to identify EX , it is a challenge to distinguish between them. Following, we analyze 1-MCs to identify these two cases and concentrate on the second case.

Case 1: A minor value of an observed variable

When comparing, for a specific EX , two centroids – one of a major cluster and the other of a 1-MC that corresponds to an observed minor value configuration (Definition 11) in which an observed variable O , which is a descendant of EX , has a minor value – we can observe that when:

1. EX changes values between two ex s that correspond to the compared clusters, all observed descendants of EX , except O , change values together, and when
2. EX does not change values between two ex s that correspond to the compared clusters, the only observed descendant of EX that changes value is O .

Thus, a PCC – between the centroid of such 1-MC and a centroid of any of the major clusters – that shows the same value for all but one (i.e., O) of the observed descendants of EX (i.e., either 1 if EX changes values in the corresponding ex s or 0 if it does not) identifies a minor value in O . For example, in Table 3, $PCC1,3$ and $PCC2,3$ of $C3$, which is a 1-MC, with the two major clusters $C1$ and $C2$ (Table 2) show the set of observed variables $X1$ - $X8$ that either do or do not change values together, whereas the single observed variable $X9$ acts contrarily. This is evidence that $C3$ is a 1-MC due to exactly a single minor value of an observed variable descendant ($X9$) of $L1$ in $G3$. Such an analysis helps LPCC ignore, on the one hand, observed descendants of $L1$ that cannot reflect minor values in $L1$'s latent (non-collider) descendants, and focus, on the other hand, on the latent descendants that should be split from $L1$, as part of Case 2.

Case 2: A minor value of a latent non-collider

The minor value of a latent non-collider NC , which is a descendant of EX , can be reflected only via the values of its observed descendants in an observed minor value configuration that is represented by a certain 1-MC. By definition, all of these observed descendants

have major values in this 1-order minor configuration since only NC has a minor value in this configuration. The major value of each of these observed descendants is certain given the minor value of NC (Proposition 2) and different from the certain major value it would have if NC had a major value (Assumption 7) instead of its minor value.

When comparing for a specific EX , two centroids – one of a major cluster and the other of a 1-MC that corresponds to an observed minor value configuration in which a latent non-collider NC , which is a descendant of EX , has a minor value – we can observe that when:

1. EX changes values between two exs that correspond to the compared clusters, all observed descendants of EX , but not observed descendants of NC , change values together, and when
2. EX does not change values between two exs that correspond to the compared clusters, the only observed descendants of EX that change values are those of NC .

Thus, a PCC – between the centroid of such 1-MC and a centroid of any of the major clusters – that shows two sets of two or more observed variables, each set having a different value, identifies a minor value in NC . The first set in such a PCC comprises the descendants of NC (with a value of 0 if EX changes values in the corresponding exs or 1 if it does not), and the second set comprises all other observed variables that are descendants of EX , but not NC (with a value of 1 if EX changes values in the corresponding exs or 0 if it does not). For example, $PCC1,6$ and $PCC2,6$ (Table 4) of $C6$, which is a 1-MC, with the two major clusters $C1$ and $C2$ (Table 2), show two sets of observed variables for $G3$. The first set consists of $X1-X6$ and the second of $X7-X9$. This is evidence that $C6$ is a 1-MC due to a minor value of a latent non-collider descendant of $L1$, and $L1$ should be split into two latents (each is responsible for one of the two sets). One latent (which we know is $L3$) is a parent of $X7$, $X8$, and $X9$, and the other latent is a parent of $X1-X6$ (which we will show is also split to $L1$ and $L2$, each with its three children).

Distinguishing between Case 1 and Case 2 gives us an instrument to identify latent non-colliders. We are interested in PCCs between 1-MCs and major clusters that show two sets of two or more elements corresponding to the observed variables. Variables in each set have the same value, which is different than that of the other set. Following, we infer that each set is of a different latent than the one that was expected to be sole. We denote such PCC by 2S-PCC (i.e., PCC of “two sets”) and the corresponding 1-MC by 2S-MC (Definition 18). Thus, to identify a latent non-collider that was combined to an exogenous latent EX , we consider only the 2S-PCCs; these PCCs are the result of comparing all the 2S-MCs among the 1-MCs for EX with the major clusters that revealed EX . Table 4 represents all 2S-PCCs for $G3$.

Definition 18 2S-PCC is PCC between 1-MC and a major cluster that shows two sets of two or more elements corresponding to the observed variables. Elements in each set have the same value, which is different than that of the other set. Accordingly, this 1-MC is defined as 2S-MC.

The following Theorem 3 helps formalize this identification step, but to prove this theorem, we first need Lemma 4. Recall that the challenge here is to identify a latent non-collider NC that is a descendant of an exogenous latent EX , but was wrongly combined with this exogenous ancestor. To face this challenge, we need to find a circumstance in which EX and NC are involved that is different than that which led to the inability to distinguish between them. NC could not be distinguished from EX when we analyzed major value configurations. But, although a major value configuration is the most probable configuration (Definition 9), minor value configurations are possible too – according to the probability tables of the latents, each given its direct parent – albeit less likely. A minor value configuration in which only NC takes a minor value (i.e., a first-order minor value configuration) is exactly what we need.⁹ This is because all NC 's latent ancestors, in the first-order minor value configuration, take the same major values they took in the major value configuration and thus influence their descendants the same. But, the minor value NC takes influences its (latent and observed) descendants differently than the major value NC took in the major value configuration. This influence is revealed in the different values the observed children of NC and its descendants take compared to the values they took when NC had a major value. Since the two value configurations are represented in two corresponding clusters – a major cluster and a 2S-MC for NC – the signature of NC can uniquely be detected by comparing the two clusters using 2S-PCC.¹⁰

Lemma 4 shows that it is possible to identify NC because: 1) Even when EX leads to major values in all NC 's ancestors (and in most cases also in NC), NC can still take a minor value; and 2) even when EX changes values, leading all NC 's ancestors to change values as well, NC can still keep the same (minor) value. Thereby, minor value configurations for NC demonstrate its autonomy, enabling its identification and its split from EX .

Lemma 4 Let a latent non-collider NC be a descendant of an exogenous latent variable EX . 2S-PCC is a PCC between a “two-set” first-order minor cluster 2S-MC due to a minor value in NC and a major cluster that identified EX . ex' and ex'' are two value configurations of EX that correspond to the compared clusters by 2S-PCC. When:

1. EX does not change values between ex' and ex'' , all the elements in 2S-PCC corresponding to the observed descendants of the latent ancestors of NC (including EX) show no change (i.e., are 0), whereas the elements corresponding to the observed descendants of NC show a change (i.e., are 1), and when
2. EX changes values between ex' and ex'' , all the elements in 2S-PCC corresponding to the observed descendants of the latent ancestors of NC (including EX) show a change (i.e., are 1), whereas the elements corresponding to the observed descendants of NC show no change (i.e., are 0).

⁹All other first-order minor value configurations (due to other latent variables, which are also EX 's descendants) or k -order minor value configurations (Definition 13) due to EX are irrelevant to the identification of NC , although the former – as will be shown in Theorem 3 – play a role in determining the direct observed children of NC among its observed descendants.

¹⁰Any 2S-PCC, which is detected for EX , will point to the NC that corresponds to the 2S-MC that is compared by this 2S-PCC.

Before moving to Theorem 3, let us illustrate the two cases discussed in Lemma 4 for G3. The “EX does not change values between \mathbf{ex} and \mathbf{ex}'' ” case can be demonstrated, for example, when comparing C1 and C6 (Table 2). In response to $EX(L1)=1$, NC’s (L3) parent (L2) takes a major value of 1 in both the value configurations of the latent variables in response to $\mathbf{ex}' = \mathbf{ex}''$.¹¹ Also, L3 takes a major value of 1 in the configuration that is represented by C1, which is one of the two major clusters. But L3, in response to the same configuration of its latent ancestors (L1 and L2), takes a minor value of 0 in the value configuration that is represented by the 2S-MC C6. By comparing C1 and C6, the corresponding 2S-PCC (i.e., $PCC1,6$; see Table 4) shows two sets of elements: the first of 0s that correspond to the observed variables X1–X6, which do not change values between the clusters, and the second of 1s that correspond to X7–X9, which do change values between the clusters. This is the evidence we are looking for that is needed to identify L3.

The “EX changes values between \mathbf{ex}' and \mathbf{ex}'' ” case can be demonstrated, for example, when comparing C1 and C9 (Table 2). In response to $EX(L1)=1$ and $EX(L1)=0$, NC’s (L3) parent (L2) takes a major value of 1 in response to L1=1 and a major value of 0 in response to L1=0. In the first instance, L3 takes a major value of 1 to create the major configuration that is represented by C1, and in the second instance, L3 takes a minor value of 1 in the value configuration that is represented by the 2S-MC C9 (and although the first value is major and second is minor, they are both 1). By comparing C1 and C9, the corresponding 2S-PCC shows two sets of elements, the first of 1s that correspond to the observed variables X1–X6, which changed values between the clusters, and the second of 0s that correspond to X7–X9, which did not change values between the clusters. This is additional support of the existence of L3. However, relying only on part of the 2S-PCCs may be inadequate to conclude on all possible splits. For example, $PCC1,8$ and $PCC2,8$ (Table 4) show that X1–X3 and X4–X9 are children of different latents, but do not suggest the split of X7–X9 as $PCC1,6$ and $PCC2,6$ do. Therefore, similarly to the MSO concept that was introduced for major-major PCCs to identify exogenous latents, it is necessary to introduce also for 2S-PCCs a maximal set of observed variables (2S-MSO) that always change their values together in all 2S-PCCs. We define:

Definition 19 A 2S-MSO is the maximal set of observed variables that always change their values together in all 2S-PCCs.

For example, X1 in Table 4 changes its value in $PCC2,6$, $PCC1,8$, $PCC1,9$, and $PCC2,10$ and always together with X2 and X3 (and the other way around). Thus, {X1, X2, X3} and similarly {X4, X5, X6} and {X7, X8, X9} are 2S-MSOs. Each 2S-MSO includes children of the same latent non-collider, which is a descendant of EX, or EX itself. After computing all 2S-PCCs for EX, LPCC detects 2S-MSOs for all these latent variables and thereby identifies all possible splits for EX. Note that compared to MSO (Section 4.1), which is identified in major-major PCCs to reveal exogenous latents, 2S-MSO is identified in PCCs between 2S-MCs and major clusters to reveal splits of latent non-colliders from the exogenous latent that was previously learned using these major clusters.

¹¹Note that the values the three latents take in the two-value configurations can only be inferred from the values their children (X1–X3 for L1, X4–X6 for L2, and X7–X9 for L3) take.

Theorem 3 Variables of a particular 2S-MSO are children of an exogenous latent variable EX or any of its descendant latent non-colliders NC.

After splitting the latent non-collider descendants from their exogenous latent ancestors EX, we need to identify the links between these latents. To identify these links, LPCC exploits the following Proposition 10 and Theorem 4. We will see that in the case of a serial connection, LPCC learns the undirected links among the latents, and in the case of a diverging connection, LPCC learns the directed links among the latents. That is, LPCC learns a pattern over the structural model of G, which represents a Markov equivalence class of models among the latents. In the special case where G has no serial connection, LPCC learns the true graph.

Proposition 10 In 2S-PCCs in which only the observed children of a single latent change, the latent is

1. EX or its leaf latent non-collider descendant, if the connection is serial; or
2. EX’s leaf latent non-collider descendant, if the connection is diverging.

Proof We already showed that at least a single 2S-PCC exists in the serial connection case in which only the observed children of EX change (Theorem 3). In addition, in the proof of Theorem 3 (Part II), we already showed that for any NC that is a latent non-collider descendant of EX, NC’s observed children change values in some 2S-PCCs with observed children of a latent non-collider descendant of NC and in the other 2S-PCCs with observed children of a latent non-collider ancestor of NC, but never alone. A special case in the proof of Theorem 3 is when NC is a leaf. Then, at least a single 2S-PCC exists in which only the children of NC change. ■

We will exemplify Proposition 10 using G3. Table 4 shows all the 2S-PCCs for G3 from which we can identify three 2S-MSOs: {X1, X2, X3}, {X4, X5, X6}, and {X7, X8, X9}. If we consider only 2S-PCCs due to C1 (the first major cluster), {X1, X2, X3} change alone in $PCC1,8$, and {X7, X8, X9} change alone in $PCC1,6$. By Proposition 10, these two 2S-MSOs are observed children of an exogenous latent variable EX and its leaf latent non-collider descendant. From knowing G3, we know that these two latents are L1 and L3. Note that if more than a single leaf of EX exists (i.e., in the case of a diverging connection emerging from EX), then for each such leaf, there is a 2S-PCC in which only the observed children of this leaf change alone. This will help LPCC to identify a diverging connection and determine EX as the source in all paths leading to the leaves (sinks). As a result, LPCC could identify the correct direction of the links among the latents.

Proposition 10 guarantees that if the connection is serial, we find the source (EX) and sink of the path between them (but not who is who). To identify the directionality between any two latent non-collider variables on the path between the source and sink, we will need more. To motivate the need, suppose that when learning G3, we already identified L1 as EX and L3 as EX’s leaf descendant (Proposition 10), and now we have to split L2 from L1 using the two major clusters, C1 and C2 (Table 2), which revealed L1, and identify the directionality among these three latent variables. Lemma 4 (first part) guarantees that the

observed children of a latent non-collider $NC1$, which is a child of another non-collider $NC2$ (both are descendants of EX), will change in all 2S-PCCs with the observed children of $NC2$ except in a single additional 2S-PCC due to a minor value of $NC1$. That is, $NC1$ is identified as a direct child of $NC2$ if the observed children of $NC1$ change in all 2S-PCCs (due to a specific major cluster and when EX does not change value), in which the children of $NC2$ change plus an additional 2S-PCC in which they change without the children of $NC2$.¹² In our case, this means that the observed children of $L3$, which is a child of $L2$, will change values in all 2S-PCCs in which the observed children of $L2$ change values, and also in an additional 2S-PCC, which is due to a minor value in $L3$. Indeed, $PCC1,10$ (Table 4), due to $C1$, shows that when EX does not change values and the observed children of $L3$, $\{X7,X8,X9\}$, change values, the observed children of $L2$, $\{X4,X5,X6\}$, also change values. In addition, $PCC1,6$, which is the result of comparing $C1$ and 2-MC $C6$ due to a minor value of $L3$, shows that $\{X7,X8,X9\}$ change values without $\{X4,X5,X6\}$ once. $PCC2,8$ and $PCC2,9$ demonstrate the same, when using major cluster $C2$ instead of $C1$ (and $C9$ is the 2-MC that reveals the minor value of $L3$). This provides an indication that $L3$ is a child of $L2$.

But, Proposition 10 cannot guarantee distinguishing between EX and its leaf latent non-collider descendant (hereby a “leaf”); hence, what if we mistakenly identified them? In the $G3$ example, this means we identified $L3$ as EX and $L1$ as EX 's leaf. Lemma 4 demonstrates an interplay between EX and NC (and all of its descendants) as presented in 2S-PCCs due to a minor value in NC ; when one of them changes, the other does not and vice versa. Because the leaf is one of NC 's descendants, Lemma 4 guarantees that the observed children of the leaf do not change if and only if EX changes value. That is, by the second part of Lemma 4, if EX changes, then the observed children of the leaf do not change. Thus, if we find 2S-PCCs that show that the observed children of the leaf do not change, then we have evidence that EX changed. This guarantees that the observed children of a latent non-collider $NC2$ (or EX itself), which is a parent of another non-collider $NC1$, will change in all 2S-PCCs with the observed children of $NC1$, except in a single additional 2S-PCC due to a minor value of $NC2$ (or if $NC2$ is EX). In our case, this means that the observed children of $L1$, which is $L2$'s parent, will change values in all 2S-PCCs in which the observed children of $L2$ change values, and also in an additional 2S-PCC. Indeed, $PCC1,9$ (Table 4), due to $C1$, shows that when the leaf does not change value and the observed children of $L1$, $\{X1,X2,X3\}$, change values, the observed children of $L2$, $\{X4,X5,X6\}$, also change values. In addition, $PCC1,8$ shows that $\{X1,X2,X3\}$ change values without $\{X4,X5,X6\}$ once. $PCC2,6$ and $PCC2,10$ demonstrate the same when using major cluster $C2$ instead of $C1$. This provides an indication that $L1$ is a child of $L2$, which is the opposite direction between the two in $G3$. That is, the interplay between EX and its leaf lets LPCC identify the directionality between latent non-colliders on the path between

¹²Note that Lemma 4 makes a clear distinction between NC 's ancestors (and their observed children) and NC 's descendants (and their observed children), when NC gets a minor value. That is, all NC 's ancestors follow EX (and change values or not with it) and all NC 's descendants follow its change of value. This change of NC “breaks” the influence of EX on the latents on the path emerging from EX and “starts” NC 's own influence on its latent descendants. And this is what is so important in finding the traces of minor values of endogenous latents through 2S-PCCs, that these traces identify the existence of the latents. Particularly, when EX does not change values and all its descendants get major values, the observed children of $NC1$ and $NC2$ will change together, and it is only a minor value that $NC1$ gets that can make a 2S-PCC in which $NC1$'s observed children change without those of $NC2$, and thereby indicate that $NC1$ is $NC2$'s child.

EX and the leaf, and in both directions. This means that LPCC can identify only the undirected links between the latents in the serial case.

In the diverging case, the children of EX never change alone, and every latent that its children change alone in some 2S-PCC is a leaf (Proposition 10). Therefore, by performing an analysis as for the serial case using 2S-PCCs in which the observed children of the leaf do not change for each leaf of the branches of the diverging connection, LPCC can identify the links among the latents in opposite directions on each branch. We formalize this by Theorem 4.

Theorem 4 *A latent non-collider $NC1$ is a direct child of another latent non-collider $NC2$ (both on the same path emerging in EX) only if:*

- *In all 2S-PCCs for which EX does not change, the observed children of $NC1$ always change with those of $NC2$ and also in a single 2S-PCC without the children of $NC2$; and*
- *In all 2S-PCCs for which a latent non-collider leaf descendant of EX does not change, the observed children of $NC2$ always change with those of $NC1$ and also in a single 2S-PCC without the children of $NC1$.*

LPCC uses Theorem 4 to identify the links between the split latents. In the serial connection, there are only two latents with observed children that change alone in some 2S-PCCs, that is, EX and its leaf latent non-collider descendant. However, LPCC cannot distinguish between them and thus finds all the links between these two latents as undirected. In the diverging connection, the observed children of EX never change alone (Proposition 10); thus, every latent with children that change alone in some 2S-PCCs can only be a leaf. Thereby, LPCC can identify the directed links among the latents repeatedly on each of the paths from EX to each of the leaves (Theorem 4). Still, LPCC needs to distinguish between the serial and diverging connections. In the case where the observed children of three or more latents change alone in some 2S-PCC, it is clear that it is a diverging connection. Then, LPCC treats these latents as leaves and returns directed paths from EX to each such leaf. However, in the case in which LPCC identifies that the observed children of exactly two latents change alone in some 2S-PCCs, it applies the analysis proposed in Theorem 4 to each of the latents. If it obtains the same path with opposite directions, then LPCC considers it as a serial connection and returns the undirected path; otherwise, it considers it as a diverging connection and returns the two directed paths from EX .

5. Discussion and Future Research

We introduced the PCC concept and LPCC algorithm for learning LVMs:

1. LPCC combines learning graphical models with data clustering by using the PCC concept to analyze clustering results of discrete variables for learning LVMs;
2. LPCC learns MIM, which is a large subclass of SEM. In MIM, multiple latent variables may have multiple indicators (observed children), and no observed variable may be an ancestor of any latent variable;

3. LPCC is not limited to latent-tree models, which are only a subclass of MIM, and does not make special assumptions, such as linearity, about the distribution;
 4. LPCC assumes that the measurement model of the true graph is pure, but, if the true graph is not pure, LPCC learns a pure sub-model of the true model, if one exists. LPCC's only assumption about the structural model is that a latent collider does not have any latent descendants (a detailed list of assumptions LPCC makes is given in Appendix O);
 5. LPCC is a two-stage algorithm. First, LPCC learns the exogenous latents and the latent colliders, as well as their observed descendants, by utilizing pairwise comparisons between data clusters in the measurement space that may explain latent causes. Second, LPCC learns the endogenous latent non-colliders and their children by splitting these latents from their previously learned latent ancestors;
 6. LPCC learns an equivalence class of the structural model of the true graph; and
 7. LPCC is formally expressed as an algorithm and evaluated using synthetic and real-world databases in Part II of the paper.
- A number of open problems invite further research including:
1. Extending LPCC to identify observed variables that are effects of other observed variables;
 2. Providing a formal analysis for the conditions of model identification and its sensitivity to parameterization. Learning by LPCC that an observed variable O is a descendant of a latent variable L depends on two factors. The first factor is the "graph distance", which means that the more edges that separate O from L , the less likely O would be grouped with other observed variables, descendants of L . The second factor is the conditional probabilities of an observed variable given its latent parent, which means that the stronger the probabilities are, the more likely the link will be identified by LPCC. Although the iterative strategy for choosing the major clusters (Section 4.3) improves the identification of observed children with weak associations with their latent parents, the final graph still depends on the initial graph. That is, the iterative approach alone cannot guarantee finding the optimal model. Future analysis should take into account both factors;
 3. Analyzing LPCC complexity. Future research should dive into this topic and decompose LPCC complexity to those of clustering, identification of major-major PCCs, and identification of major-minor PCCs. Assume a set $V = (\text{LUO})$ with a variable maximal cardinality, $k = \max(|V_i|)$, a number of exogenous variables, $|\text{EX}|$, and a number of major value configurations (major clusters), $|\text{ex}| = k^{|\text{EX}|}$. A preliminary analysis shows that LPCC complexity in identifying major-major PCCs is $O(k^{2|\text{EX}|})$. To compute the LPCC's complexity in identifying major-minor PCCs, we first have to identify 1-MC minor clusters (values), with complexity of $O((|V| - |\text{EX}|)k^{|\text{EX}|}(k - 1))$ due to $(k - 1)$ minor values for each of $k^{|\text{EX}|}$ parent configurations of $(|V| - |\text{EX}|)$ endogenous variables. Then, the complexity in identifying major-minor

PCCs is $O((|V| - |\text{EX}|)k^{2|\text{EX}|}(k - 1))$, and the total complexity in computing PCCs is $O((|V| - |\text{EX}|)k^{2|\text{EX}|})$, which is exponential in $|\text{EX}|$, but in most problems $|\text{EX}| \ll |V|$. However, a more elaborated analysis that also includes the complexity of clustering is desired.

4. Exploring the impact of clustering – as is manifested by the clustering algorithm and its parameters – on the LPCC results. In Part II, we show a problem in which the data structure is hierarchical, and a clustering algorithm that is more sophisticated than SOM, which is suggested in Section 4, is needed to preprocess the data used to learn an LVM that is meaningful to the domain. Exploring the requirements on clustering and any guidelines about the best approach to take for clustering is a direction of further research; and
5. Suggesting ways to use the graphical model to cluster data points. Although we have established and exploited a link between cluster analysis and learning an LVM, in this work, we only studied learning (reconstructing) the graphical model by analyzing clusters of observational data. Another very interesting line of future research is in the opposite direction, extending previous studies such as that in Zhang (2004). Because MIM models learned by LPCC are richer than HLC models (which are only a subset of MIM), such a line of research may enable accurate clustering of observational data generated by a model also having collider nodes.

Acknowledgments

The authors thank Ricardo Silva from UCL for his helpful comments and suggestions given to improve an earlier version of this paper. The authors also thank the two anonymous reviewers for their comments and suggestions that helped strengthen the paper and the special issue editors: Isabelle Guyon and Alexander Stahnikov. Numan Asbeh thanks the *Planning and Budgeting Committee* (PBC) of the *Israel Council for Higher Education* for its support by a scholarship for distinguished Ph.D. students.

Appendix A. Proofs of propositions, lemmas, and theorems

In this appendix, we give proofs of propositions, lemmas, and theorems for which the proof is too detailed, lengthy, or impedes the flow of reading. All other proofs are given in the body of the paper.

Lemma 1

1. Each latent non-collider NC_t has only one exogenous latent ancestor EX_{NC_t} , and there is only one directed path T_{NC_t} from EX_{NC_t} (source) to NC_t (sink). (Note that we use the notation NC_t , rather than S_t , since the Lemma applies to both exogenous and endogenous latent non-colliders.)
2. Each latent collider C_j is connected to a set of exogenous latent ancestors EX_{C_j} via a set of directed paths T_{C_j} from EX_{C_j} (sources) to C_j (sink).

Proof

1. If the latent non-collider is exogenous, $NC_t = EX_t$, then $EX_{NC_t} = EX_t$, and T_{NC_t} is the empty path consisting of EX_t . For example, $EX_{L3} = L3$ and $T_{L3} = L3$ in G2 and G5 in Figure 1. If, however, the latent non-collider is endogenous, $NC_t = S_t$, and we assume by contradiction that it has more than one exogenous latent ancestor and thus more than one directed path from each exogenous ancestor to S_t (and according to Assumption 5, none of the paths passes through a collider) that collide at S_t , then S_t is a collider. This is contrary to the assumption that NC_t is a non-collider. That is, EX_{S_t} is the only exogenous latent ancestor of S_t , and T_{S_t} is the only directed path from EX_{S_t} through S_t 's parent Pa_t to S_t . For example, $EX_{L5} = L3$ and $T_{L5} = \{L3, L4, L5\}$ in G5 (Figure 1).

[Note that if S_t has no endogenous latent non-collider ancestors, then $Pa_t = EX_{S_t}$, and T_{S_t} equals the ordered sequence $\{EX_{S_t}, S_t\}$, e.g., $EX_{L4} = L3$ and $T_{L4} = \{L3, L4\}$ in G5 (Figure 1).]

2. Under Assumption 5, any parent Pa_j of latent collider C_j could be either a latent non-collider or an exogenous latent; in other words, $Pa_j \subset (NC \cup EX)$. If Pa_j is a latent non-collider, then it is on the directed path T_{C_j} from EX_{C_j} to C_j ; and if Pa_j is an exogenous latent EX_{C_j} , then it is the source of a directed path T_{C_j} (or more than a single directed path) to C_j . $EX_{C_j} = \cup EX_{C_j}$ is the set of exogenous ancestors of C_j , and $T_{C_j} = \cup T_{C_j}$ is the set of directed paths from EX_{C_j} to C_j . For example, $EX_{L4} = \{L1, L5\}$ and $T_{L4} = \{\{L1, L2, L3, L4\}, \{L5, L4\}\}$ in G6 (Figure 1). ■

Proposition 3 The $MAV_{(TS_{NC_v} \setminus EX_{NC_v}, ONC_v)}(ex_{NC_v})$ corresponding to $MAE_{(TS_{NC_v} \setminus EX_{NC_v}, ONC_v)}(ex_{NC_v})$ is a certain value configuration for each certain value ex_{NC_v} .

(Note that here we use the notation NC_v rather than S_v since the proposition applies to both exogenous and endogenous latent non-colliders.)

Proof If the latent non-collider NC_v is exogenous, $NC_v = EX_v$ and $ONC_v = OEX_v$, then $\{TS_{NC_v} \setminus EX_{NC_v}, OEX_v\} = OEX_v$ and the partial major value is the local major value $MAV_{OEX_v}(ex_v)$, which by Proposition 2 is certain for a certain value ex_v .

If the latent non-collider NC_v is endogenous, $NC_v = S_v$ and $ONC_v = OS_v$, then we consider $\{TS_{S_v} \setminus EX_{S_v}, OS_v\}$, which is a set of ordered variables along the directed path T_{S_v} that ends in OS_v . The remainder of the proof is by induction:

Basis: Based on Proposition 2, $MAV_{S_1}(ex_{S_1})$, where S_1 is the first variable in $\{TS_{S_v} \setminus EX_{S_v}, OS_v\}$ and a direct child of EX_{S_v} , given a certain value ex_{S_1} , is also certain.

Step: If the major value of the i th variable, S_i , in the subset $\{TS_{S_v} \setminus EX_{S_v}, OS_v\}$, i.e., $MAV_{S_i}(pa_{S_i}^{ex_{S_i}})$, is certain for a certain value $pa_{S_i}^{ex_{S_i}}$, then the major value of the $(i+1)$ th variable, S_{i+1} , in the subset (which is S_i 's child), i.e., $MAV_{S_{i+1}}(pa_{S_{i+1}}^{ex_{S_i}})$, is by Proposition 2 certain too for a certain value $pa_{S_{i+1}}^{ex_{S_i}}$ (which is $MAV_{S_i}(pa_{S_i}^{ex_{S_i}})$). ■

Proposition 4 All corresponding values in $MAV_{(TS_{NC_v} \setminus EX_{NC_v}, ONC_v)}(ex'_{NC_v})$ and $MAV_{(TS_{NC_v} \setminus EX_{NC_v}, ONC_v)}(ex''_{NC_v})$ for two values ex'_{NC_v} and ex''_{NC_v} of EX_{NC_v} are different.

(Here also we use the notation NC_v , since the proposition applies to both exogenous and endogenous latent non-colliders.)

Proof If the latent non-collider NC_v is exogenous, $NC_v = EX_v$ and $ONC_v = OEX_v$, then $\{TS_{NC_v} \setminus EX_{NC_v}, OEX_v\} = OEX_v$, and, by Assumption 7, the corresponding $MAV_{OEX_v}(ex'_v)$ and $MAV_{OEX_v}(ex''_v)$ are different for two values ex'_v and ex''_v .

If the latent non-collider NC_v is endogenous, $NC_v = S_v$ and $ONC_v = OS_v$, then we consider $\{TS_{S_v} \setminus EX_{S_v}, OS_v\}$, which is a set of ordered variables along the directed path T_{S_v} that ends in OS_v . The remainder of the proof is by induction:

Basis: The major local values $MAV_{S_1}(ex'_{S_1})$ and $MAV_{S_1}(ex''_{S_1})$ of the first variable, S_1 , in $\{TS_{S_v} \setminus EX_{S_v}, OS_v\}$ (which is also a direct child of EX_{S_v}) and two values ex'_{S_1} and ex''_{S_1} of EX_{S_v} are different based on Assumption 7.

Step: If the major local values of the i th variable, S_i , in $\{TS_{S_v} \setminus EX_{S_v}, OS_v\}$ and two values ex'_i and ex''_i of EX_{S_v} , i.e., $MAV_{S_i}(ex'_i)$ and $MAV_{S_i}(ex''_i)$, are different, then the major local values of S_{i+1} (S_i 's child), and the two values $MAV_{S_i}(ex'_i)$ and $MAV_{S_i}(ex''_i)$, i.e., $MAV_{S_{i+1}}(pa_{S_{i+1}}^{ex'_i}) = MAV_{S_{i+1}}(MAV_{S_i}(ex'_i))$ and $MAV_{S_{i+1}}(pa_{S_{i+1}}^{ex''_i}) = MAV_{S_{i+1}}(MAV_{S_i}(ex''_i))$ are different too based on Assumption 7. ■

Proposition 5 EX_{NC_v} changes values (i.e., has two values ex'_{NC_v} and ex''_{NC_v}) if and only if NC_v changes values in the two corresponding major value configurations: $MAV_{(TS_{NC_v} \setminus EX_{NC_v}, ONC_v)}(ex'_{NC_v})$ and $MAV_{(TS_{NC_v} \setminus EX_{NC_v}, ONC_v)}(ex''_{NC_v})$.

Proof (“if”) Proposition 3 guarantees that NC_v has a certain value in $MAV_{(TS_{NC_v} \setminus EX_{NC_v}, ONC_v)}(ex_{NC_v})$ for a certain value ex_{NC_v} of EX_{NC_v} . Thus, if NC_v has

different values in two $MAV_{\{TS_{C_q} \setminus EX_{N_{C_q}}, ON_{C_q}\}}(ex_{N_{C_q}})$, then $EX_{N_{C_q}}$ should also have two corresponding values, say $ex'_{N_{C_q}}$ and $ex''_{N_{C_q}}$.

(“only if”) Proposition 4 guarantees that N_{C_q} will have different values in $MAV_{\{TS_{N_{C_q}} \setminus EX_{N_{C_q}}, ON_{C_q}\}}(ex'_{N_{C_q}})$ and $MAV_{\{TS_{N_{C_q}} \setminus EX_{N_{C_q}}, ON_{C_q}\}}(ex''_{N_{C_q}})$ for two values $ex'_{N_{C_q}}$ and $ex''_{N_{C_q}}$ of $EX_{N_{C_q}}$. Thus, if N_{C_q} has only a certain value in two

$MAV_{\{TS_{N_{C_q}} \setminus EX_{N_{C_q}}, ON_{C_q}\}}(ex_{N_{C_q}})$ then $EX_{N_{C_q}}$ should have also a certain value in the corresponding two $ex_{N_{C_q}}$. ■

Proposition 6 The $MAV_{\{TS_{C_q} \setminus EX_{C_q}, OC_k\}}(ex_{C_q})$ corresponding to $MAE_{\{TS_{C_q} \setminus EX_{C_q}, OC_k\}}(ex_{C_q})$ is a certain value configuration for each certain value configuration ex_{C_q} .

Proof $\{TS_{C_q} \setminus EX_{C_q}, OC_k\}$ comprises sets of variables $\{TS_{C_q} \setminus EX_{C_q}, OC_k\}$ along all directed paths through C_k that end at OC_k . We will divide each such set into to three subsets $\{TS_{C_q} \setminus EX_{C_q}, C_k\}$, C_k , and OC_k and consider a value configuration for ex_{C_q} for each subset separately. First, since no latent collider can be a child of a latent collider (Assumption 5), a value configuration for the subset $\{TS_{C_q} \setminus EX_{C_q}, C_k\}$ is considered to be identical to a value configuration for $\{TS_{N_{C_q}} \setminus EX_{N_{C_q}}\}$, and thus according to Proposition 3, is a certain value configuration for a certain value ex_{C_q} . Because $MAV_{\{TS_{C_q} \setminus EX_{C_q}, C_k\}}(ex_{C_q})$ is a certain value configuration for a certain ex_{C_q} for each directed path TS_{C_q} that is included in TS_{C_q} , the product of these value configurations, which corresponds to the product of $MAE_{\{TS_{C_q} \setminus EX_{C_q}, C_k\}}(ex_{C_q})$ in (11), is also certain. Second, since C_k 's parents $\mathbf{Pa}_k \subset \bigcup_{TS_{C_q} \in TS_{C_q}} \{TS_{C_q} \setminus EX_{C_q}, C_k\}$, $\mathbf{pa}_k^{ex_{C_q}}$ are certain value configurations. Thus, based on Proposition 2, $MAV_{C_q}(\mathbf{pa}_k^{ex_{C_q}})$ is also a certain value and similarly $MAV_{OC_k}(C_k^{ex_{C_q}})$ is certain, where $C_k^{ex_{C_q}} = MAV_{C_q}(\mathbf{pa}_k^{ex_{C_q}})$. Therefore, all variables in $\{TS_{C_q} \setminus EX_{C_q}, OC_k\}$ are certain in the major configuration for a certain value configuration ex_{C_q} . ■

Proposition 7 For every exogenous ancestor $EX_{C_q} \in \mathbf{EX}_{C_q}$ of a latent collider C_k , there are at least two configurations ex'_{C_q} and ex''_{C_q} of \mathbf{EX}_{C_q} in which only EX_{C_q} changes values when C_k changes values in the two corresponding major value configurations $MAV_{\{TS_{C_q} \setminus EX_{C_q}, OC_k\}}(ex'_{C_q})$ and $MAV_{\{TS_{C_q} \setminus EX_{C_q}, OC_k\}}(ex''_{C_q})$.

Proof We divide the proof into two parts. In the first part, we prove that for each exogenous ancestor of a latent collider, there are at least two MAVs in which only the collider's parent on the path from the exogenous to the collider (of all collider's parents) changes values together with the exogenous. We are aided in this part of the proof by Proposition 5 after considering the collider's parent as a latent non-collider. In the second part, using Assumption 7, we show that each such collider's parent changes values together with the collider in the same two MAVs in which the parent changes values together with the exogenous. Thereby, we prove that for each exogenous ancestor of a latent collider, there are at least two MAVs in which the collider changes values only with this exogenous ancestor.

For the first part, Proposition 5 guarantees that any exogenous ancestor EX_{C_q} of a parent $Pa_k \in \mathbf{Pa}_k$ of collider C_k (and thus $EX_{Pa_k} = EX_{C_q}$ and Pa_k is also a latent non-collider) changes its value if and only if Pa_k changes its value in two value configurations $MAV_{\{TS_{Pa_k} \setminus EX_{Pa_k}, OP_{Pa_k}\}}(ex_{Pa_k}')$. By the opposite of Proposition 5, any exogenous ancestor $EX_{C_q}^*$ of a parent $Pa_k^* \in \mathbf{Pa}_k \setminus Pa_k$ of C_k is certain if and only if Pa_k^* is certain in two value configurations $MAV_{\{TS_{Pa_k^*} \setminus EX_{Pa_k^*}, OP_{Pa_k^*}\}}(ex_{Pa_k^*}')$.

For the second part, we know by Assumption 7 (second part) that for every C_k that is a latent collider and for every $Pa_k \in \mathbf{Pa}_k$, there are at least two configurations \mathbf{pa}_k' and \mathbf{pa}_k'' of \mathbf{Pa}_k in which only the value of Pa_k is different and $MAV_{C_q}(\mathbf{pa}_k') \neq MAV_{C_q}(\mathbf{pa}_k'')$. That is, the collider (which is the only variable in MAV_{C_q}) changes values together with each of its parents in at least two parents' configurations.

Combining the two parts, we have proven that a collider changes values following a change in the value of each of its parents in at least two configurations of the parents, when the change of values of this parent is due to a change of values of its exogenous ancestor in two exogenous configurations. This means that the collider changes values with each of its exogenous ancestors in at least two exogenous configurations. That is, for two configurations ex'_{C_q} and ex''_{C_q} of \mathbf{EX}_{C_q} in which only EX_{C_q} changes values, there are at least two configurations \mathbf{pa}_k' and \mathbf{pa}_k'' of \mathbf{Pa}_k in which $Pa_k \in \mathbf{Pa}_k$ changes values in $MAV_{\{TS_{C_q} \setminus EX_{C_q}, OC_k\}}(ex'_{C_q})$ and $MAV_{\{TS_{C_q} \setminus EX_{C_q}, OC_k\}}(ex''_{C_q})$ with EX_{C_q} . Since these values of Pa_k in \mathbf{pa}_k' and \mathbf{pa}_k'' also change with values of C_k , C_k changes values with EX_{C_q} in ex'_{C_q} and ex''_{C_q} . Therefore, there are at least two configurations ex'_{C_q} and ex''_{C_q} of \mathbf{EX}_{C_q} in which only EX_{C_q} has changed values when C_k changes values in the two corresponding major value configurations $MAV_{\{TS_{C_q} \setminus EX_{C_q}, OC_k\}}(ex'_{C_q})$ and $MAV_{\{TS_{C_q} \setminus EX_{C_q}, OC_k\}}(ex''_{C_q})$. ■

Lemma 2

1. A latent non-collider N_{C_q} and its observed child ON_{C_q} , both descendants of an exogenous variable $EX_{N_{C_q}}$, change their values in any two major configurations if and only if $EX_{N_{C_q}}$ has changed its value in the corresponding two configurations of \mathbf{EX} .
2. A latent collider C_k and its observed child OC_k , both descendants of a set of exogenous variables \mathbf{EX}_{C_q} , change their values in any two major configurations only if at least one of the exogenous variables in \mathbf{EX}_{C_q} has changed its value in the corresponding two configurations of \mathbf{EX} .

Proof

1. First (“only if”), by Proposition 3, the major value configuration of a latent non-collider N_{C_q} and its observed child ON_{C_q} , both of which are descendants of an exogenous variable $EX_{N_{C_q}}$, are certain for any certain $ex_{N_{C_q}}$. That is, if N_{C_q} and ON_{C_q} changed their values in any two major configurations, it is only because $EX_{N_{C_q}}$ has changed its value in the corresponding two configurations of \mathbf{EX} . Second (“if”), by Proposition 4, the major value configurations of N_{C_q} and ON_{C_q} are changed if $EX_{N_{C_q}}$ has changed its value between two configurations of \mathbf{EX} .

2. By Proposition 6, the major value configuration of a collider C_k and its observed child OC_k , both of which are descendants of a set of exogenous variables EX_{C_k} , are certain for a certain ex_{C_k} . That is, if C_k and OC_k changed their values in any two major configurations, it is only because at least one of the variables in EX_{C_k} also changed its value in the corresponding two configurations of EX . ■

Theorem 1 *Variables of a particular MSO are children of a particular exogenous latent variable EX or its latent non-collider descendant or children of a particular latent collider C .*

Proof The proof is divided into two separate cases. In the first case, we show that the children of a particular exogenous latent variable or its non-collider descendant belong to the same **MSO**, and in the second case, we show that the children of a particular collider latent belong to the same **MSO**.

Case 1: MSO of observed children of an exogenous latent or its latent non-collider descendants

Let ONC_i ¹³ (in $OEX \cup OS$) be a set of observed variables that are children of an exogenous variable EX_i and any of its latent non-collider descendants (if they exist), and let OC_i be a set of observed variables that are children of latent colliders where each has EX_i as an exogenous ancestor with other exogenous variables. Note that OC_i may be empty, if EX_i does not have any collider descendants, but ONC_i is never empty because it includes at least OEX_i (Assumption 4). Because no observed child can be included in both OC_i and ONC_i , these sets are disjoint. Their union, $OV_i = ONC_i \cup OC_i$, includes all the observed variables that are affected by EX_i and thus should change their values when EX_i changes.

- First, by Lemma 2 (first part), any subset of variables in ONC_i (and thus also ONC_i itself, which is a maximal set) always changes together in all PCCs that correspond to a change in EX_i and never change together in any other PCC. These variables belong to the same **MSO** that represents EX_i .
- Second, let subset $OC_{i,j}$ of OC_i contain all variables that (1) have a shared exogenous ancestor EX_j (besides EX_i) and (2) change their values together in at least one PCC, which corresponds to a change only in the value of EX_j . By Lemma 2, the other variables in OV_i that are not descendants of EX_j do not change in that PCC. Thus, variables in $OC_{i,j}$ do not belong to the same **MSO** for which variables in ONC_i belong.

Consequently, variables in ONC_i will change together only in all PCCs that correspond to a change in EX_i , and therefore, will establish a maximal set of the variables $MSO_i = ONC_i$ that corresponds to all and only observed variables that are children of exogenous variable EX_i and its latent non-collider descendants.

¹³So far, observed variables had their own indices and their parents/ancestors also had these indices. In Theorem 1, the index is associated with the exogenous variable (Case 1) and the collider latent (Case 2), since these are the central subjects of interest here.

Case 2: MSO of observed children of a latent collider

In this case, it is important to note that different colliders and their children are affected by different sets of exogenous variables. Thus, we assume:

Assumption 8 *Latent colliders do not share exactly the same sets of exogenous ancestors.*

(In case Assumption 8 is violated, for example, if several latent colliders share exactly the same set of exogenous ancestors, LPCC does not identify the latent colliders as separate and learns one collider as the parent of all children of the latent colliders. Nevertheless, we believe this assumption is very realistic.)

Let OC_j be the set of the observed variables that are children of latent collider C_j that is a descendant of a set of exogenous variables EX_{C_j} . By Lemma 2, any variable in OC_j should not change in any PCC unless at least one of its exogenous ancestors changes. The sets of variables that should change together with variables in OC_j if any of the exogenous variables in EX_{C_j} change is represented by:

$$OV = \bigcup_{EX_i \in EX_{C_j}} OV_i = \bigcup_{EX_i \in EX_{C_j}} (ONC_i \cup OC_i) = \bigcup_{EX_i \in EX_{C_j}} (ONC_i \cup \{OC_i \setminus OC_j\}) \cup OC_j \quad (12)$$

where the union is over all exogenous ancestors EX_i of C_j . We separate the proof to include three sets of observed variables: 1) OC_j , which are children of C_j ; 2) ONC_i , which are children of an exogenous variable EX_i and any of its latent non-collider descendants; and 3) $\{OC_i \setminus OC_j\}$, which are children of latent colliders, other than C_j , that are descendants of EX_i .

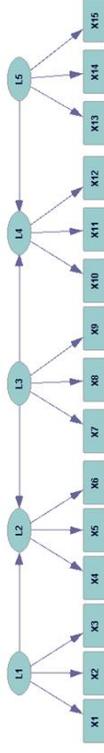


Figure 3: IVM with two latent colliders.

For example, for latent collider $C_j = L2$ in Figure 3, $EX_{L_2} = \{L1, L3\}$, $OC_{L_2} = \{X4, X5, X6\}$, $ONC_{L_1} = \{X1, X2, X3\}$, $ONC_{L_3} = \{X7, X8, X9\}$, and $\{OC_{L_4} \setminus OC_{L_2}\} = \{X10, X11, X12\}$.

Following, we analyze the three subsets of OV , specifically, OC_j , ONC_i , and $\{OC_i \setminus OC_j\}$, and show that only variables in OC_j (or any subset of OC_j) will always change together, whereas other variables in OV will not. We analyze the subsets ONC_i and $\{OC_i \setminus OC_j\}$ for each exogenous $EX_i \in EX_{C_j}$; thus, the analysis is also correct for their union (12).

- **OC_j**: By Lemma 2 (second part), any subset of variables in OC_j always changes together in all PCCs that correspond to a change in at least one exogenous variable in EX_{C_j} . In addition, none of the variables in OC_j has an exogenous ancestor that is not in EX_{C_j} ; therefore, no variable in OC_j ever changes in any PCC that corresponds to an exogenous variable that is not in EX_{C_j} . These variables belong to the same **MSO** that represents C_j .

- ONC_i : We previously showed in Case 1 that each ONC_i forms an MSO that corresponds to a single EX_i , and this is the only exogenous ancestor for ONC_i . By Lemma 3, an MSO is an equivalence class; therefore, no other variable in a subset of OV (including OC_j) can be added to ONC_i , and it will remain an MSO . Similarly, no subset of variables in ONC_i can be added to any subset of OV to obtain an MSO .

• $\{\text{OC}_i, \text{OC}_j\}$: Any subset of $\text{OC}_i \setminus \text{OC}_j$ does not change together with any subset of OC_j because (Assumption 8) for each variable OC_j in $\text{OC}_i \setminus \text{OC}_j$, there is an exogenous ancestor EX_j that is not an ancestor of variables in OC_i . Thus, by Proposition 7, OC_j changes its value in a PCC that corresponds to a change only in the value of EX_j , whereas the variables in OC_i , which are not descendants of EX_j , do not change in that PCC.

Consequently, all and only variables in OC_i (maximal subset of OC_i) compose MSO_i that changes together in all PCCs that correspond to a change in EX_{C_i} . ■

Theorem 2 *A latent variable L is a collider of a set of latent ancestors LACEX only if:*

1. *The values of the children of L change in different parts of some major–minor PCCs each time with the values of descendants of another latent ancestor in LA ; and*
2. *The values of the children of L do not change in any PCC unless the values of descendants of at least one of the variables in LA change too.*

Proof Recall that by this point (Section 4.1), latent variables that have already been learned are either exogenous or colliders. Thus, first, we show that a latent variable L that satisfies (2) has to be a collider for a set of latent ancestors LACEX by assuming by contradiction that L is not a collider but an exogenous variable. If L is an exogenous variable, then there exists at least a single major–minor PCC $_L$ that corresponds to two ex s in which only L changes its value. Thus, in PCC $_L$, only the values of descendants of L change, whereas descendants of other variables in any sub-set LACEX do not change. This is in contrast to (2).

Second, we show that if L satisfies (1), then LA is the set of L 's exogenous ancestors that collide in L . Let ONC_i (in $\text{OEX} \cup \text{OS}$) be the set of observed variables that are children of $\text{LA}_i \in \text{LA}$ or children of its latent non-collider descendants. Let OC_i be the set of children of latent colliders where each has LA_i as its ancestor with other exogenous variables in LA or not. $\text{OV}_i = \text{ONC}_i \cup \text{OC}_i$ includes all the observed variables that are affected by LA_i and thus may change their values when LA_i changes values. In addition, let OC_L be the set of children of L . We need to show that if L satisfies (1), then $\text{OC}_L \subset \text{OC}_i$ for each $\text{LA}_i \in \text{LA}$. Since LA_i is an ancestor of L , (1) ensures that there exists a PCC in which only the values of descendants of LA_i , including OC_L change, whereas the values of descendants of other variables in $\text{LA} \setminus \text{LA}_i$ do not change. Thus, $\text{OC}_L \subset \text{OV}_i$. However, none of the children in OC_L belongs to ONC_i ; otherwise, it would have already been identified (Theorem 1) as a descendant of LA_i . Thus, $\text{OC}_L \subset \text{OC}_i$. ■

Lemma 4 *Let a latent non-collider NC be a descendant of an exogenous latent variable EX . 2S-PCC is PCC between a “two-set” first-order minor cluster 2S-MC due to a minor value in NC and a major cluster that identified EX . ex' and ex'' are two value configurations of EX that correspond to the compared clusters by 2S-PCC. When:*

1. *EX does not change values between ex' and ex'' , all the elements in 2S-PCC corresponding to the observed descendants of the latent ancestors of NC (including EX) show no change (i.e., are 0), whereas the elements corresponding to the observed descendants of NC show a change (i.e., are 1), and when*
2. *EX changes values between ex' and ex'' , all the elements in 2S-PCC corresponding to the observed descendants of the latent ancestors of NC (including EX) show a change (i.e., are 1), whereas the elements corresponding to the observed descendants of NC show no change (i.e., are 0).*

Proof 2S-MC represents a 1-order minor configuration of EN in which only NC has a minor value, and all the other variables in EN have major values. Thus, when

- EX does not change values between ex' and ex'' (i.e., $\text{ex}' = \text{ex}''$), then
 1. the major value configuration of the latent ancestors of NC is the same for both ex s (Proposition 3), and for each such latent, each of its observed children has the same major local value (Proposition 2) for both ex s. Thus, all the observed children of the latent ancestors of NC do not change values in both clusters, and all the corresponding elements in 2S-PCC are 0; and
 2. NC may take either a major or minor value in response to $\text{ex}' (= \text{ex}'')$, depending on the probabilities of NC to take any of its values conditioned on the values NC 's direct parent takes. The result of the first case is a major cluster (NC and both its ancestors and descendants have major values) and that of the second case is 1-MC. Since all NC 's ancestors and descendants have major values, whereas NC has a minor value, this 1-MC is 2S-MC by definition. Using these two clusters, LPCC creates 2S-PCC. Since NC d-separates its descendants (both latents and observed) from its ancestors, the values of NC 's descendants are determined only by NC in a way similar to that which we used to prove Proposition 3. Since we are concerned with the case in which NC takes different values for ex' and ex'' , its descendants too have different values in the two corresponding configurations, and following Assumption 7, all of their observed children have different values in the corresponding observed configurations and clusters. Therefore, these children change their values between the clusters, as represented by 1s in the 2S-PCC.

- EX changes values between ex' and ex'' , then
 1. by Proposition 4, all the latent ancestors of NC have different values for ex' and ex'' , and by Assumption 7, all the observed children of these latents have

different values for \mathbf{ex}' and \mathbf{ex}'' . Thus, in any 2S-PCC between two clusters corresponding to \mathbf{ex}' and \mathbf{ex}'' , all the elements that correspond to the observed children of the latent ancestors of NC (including EX) show a change (i.e., are 1); and

2. NC does not change values between \mathbf{ex}' and \mathbf{ex}'' because if it did, then by Proposition 4, all of its latent descendants have different values for \mathbf{ex}' and \mathbf{ex}'' , and by Assumption 7, all of their observed children have different values in the two corresponding observed configurations. And following, in any 2S-PCC between two clusters corresponding to \mathbf{ex}' and \mathbf{ex}'' , all the elements that correspond to NC and its descendants would show a change (i.e., are 1). But, since as we already showed that all the observed children of the ancestors of NC are equal to 1 in these 2S-PCCs, it is contrary to the definition of a 2S-PCC that needs two sets of two or more elements of different values. Thus, NC cannot change values between \mathbf{ex}' and \mathbf{ex}'' . Following and by Proposition 3, all the latent descendants of NC have certain values for this certain value of NC in both configurations, and by Proposition 2, all the observed children of these latents have certain values in the corresponding observed configurations. Thus, all the elements in 2S-PCC that correspond to the observed children of NC and its descendants do not show a change (i.e., are 0).

Note that the proof implicitly assumes that NC is on a serial connection emerging from EX . In a diverging connection, all the latent variables that are on the paths other than the one that includes NC can be considered with NC 's ancestors because both the latents on the other paths and NC 's ancestors are d-separated (for these 2S-PCCs) by NC from its descendants. Thus, the analysis proposed above for a serial connection generalizes also to the diverging connection. ■

Theorem 3 *Variables of a particular 2S-MSO are children of an exogenous latent variable EX or any of its descendant latent non-colliders NC.*

Proof 1. Variables of 2S-MSO that are children of EX

We need, first, to prove that the children of EX always change values together and second, that no other observed child of another latent can always change value with them. First, Lemma 4 guarantees that the observed children of EX always change values together since a value change of EX between two \mathbf{exs} corresponds to the compared clusters in all 2S-PCCs of 2S-MCs with the major clusters for EX . The remainder of the proof is divided into two cases: 1) a serial connection and 2) a diverging connection. In case 1, there exists at least a single 2S-PCC in which only the observed children of EX change. This 2S-PCC is between a major cluster for EX and 2S-MC due to a minor value of the direct latent non-collider

child NC^{14} of EX (e.g., L2 is the direct latent non-collider child of L1 in G3).¹⁵ Thus, only the elements in 2S-PCC that correspond to the observed children of EX show a change and are equal to 1 (e.g., PCC2.10 in Table 4), which guarantees that the observed children of EX establish a 2S-MSO.

In case 2, the same analysis proposed in case 1 is repeated for each of the direct latent non-collider children of EX in each of the paths that emerges from EX . Let us use the same notation NC for each such direct child in each path in turn. In this case, not only do the observed children of EX change each time EX changes, but also the observed descendants of the other direct latent non-collider children of EX (in all paths except that which includes NC) change with EX . This shows that the observed children of EX change which in observed descendants of the direct latent non-collider children of EX (all but the descendants of NC), but never together with all of them (as at each time, another NC is excluded). This guarantees that the observed children of EX establish a 2S-MSO.

II. Variables of 2S-MSO that are children of EX's descendant NC

In a serial connection, we identify three possible situations in which either NC , its latent descendant, or its latent ancestor takes a minor value. In each of these situations, no other latent or observed variable can take a minor value because we focus the analysis on 2S-MC through the evaluation of 2S-PCC between this minor cluster and a major cluster for EX . For each of the three situations, EX may change its value or not, so we have to consider six cases:

1. 2S-MC is due to a minor value of any of NC 's latent non-collider descendants, $NC1$, and EX does not change value between two \mathbf{exs} that correspond to the compared clusters. Then, by Lemma 4 (first part), all of $NC1$'s observed descendants do change values, but all the observed children of $NC1$'s latent ancestors, including those of NC , do not change values.
2. 2S-MC is due to a minor value of any of NC 's latent non-collider descendants, $NC1$, and EX changes value between two \mathbf{exs} that correspond to the compared clusters. Then, by Lemma 4 (second part), all of $NC1$'s observed descendants do not change values, but all the observed children of $NC1$'s latent ancestors, including those of NC , change values.
3. 2S-MC is due to a minor value of NC , and EX does not change value between two \mathbf{exs} that correspond to the compared clusters. Then, by Lemma 4 (first part), all of NC 's

¹⁴A) We focus on the latent non-collider NC that is the direct child of EX since only a minor value that NC takes can d-separate EX and its observed children from NC 's observed children and the observed children of the remaining latent non-colliders, and partition the elements in the corresponding 2S-PCC into two sets in which the first consists of the observed children of EX and the second consists of the observed children of all EX 's latent descendants. B) In our circumstances, where at least a single latent non-collider has been combined with EX , the existence of such a latent variable is guaranteed. C) It is also guaranteed that the 1-MC due to the minor value of the direct latent child of EX is 2S-MC because it cannot be due to an observed variable (see Case 2 above).

¹⁵We assume that all possible 1-MCs, including the one corresponding to a minor value of the direct latent non-collider child NC of EX , are found. Practically, if we err in estimating the threshold on the maximal 2-MC (as described above and in Appendix B), we may miss this 1-MC, but this is an identification issue that does not affect the correctness of the theorem.

observed descendants do change values, but all the observed children of its ancestors do not.

4. 2S-MC is due to a minor value of NC , and EX changes value between two exs that correspond to the compared clusters. Then, by Lemma 4 (second part), all of NC 's observed descendants do not change values, but all the observed children of its ancestors do.
5. 2S-MC is due to a minor value of NC 's latent non-collider ancestor, $NC1$, and EX does not change value between two exs that correspond to the compared clusters. Then, by Lemma 4 (first part), all the observed children of $NC1$ and of its latent descendants, including those of NC , change values.
6. 2S-MC is due to a minor value of NC 's latent non-collider ancestor, $NC1$, and EX changes value between two exs that correspond to the compared clusters. Then, by Lemma 4 (second part), all the observed children of $NC1$ and of its latent descendants, including those of NC , do not change values.

That is, in all six cases, NC 's observed children change values together; in some 2S-PCCs they change values with observed children of a latent non-collider ancestor of NC and in some other 2S-PCCs with observed children of a latent non-collider descendant of NC . Thus, not only will the set of all the observed children of NC always change values together, but also no observed child of any of NC 's latent non-collider ancestors or descendants can be part of this set. This means that the set of observed children of NC is a maximal set of variables that always change together, i.e., **2S-MSO**.

Note that if NC does not have a latent non-collider descendant or ancestor, then Cases 1 and 2 and Cases 5 and 6, respectively, do not exist. In the special case where NC is a leaf (i.e., does not have a latent descendant), Case 3 guarantees that there exists at least a single 2S-PCC in which only the observed children of NC change.

In a diverging connection, all the latent variables that are on paths other than the one that includes NC can be considered with NC 's ancestors because NC d-separates them all from its descendants. Thus, the same analysis proposed in the serial case also holds in the diverging case. ■

Theorem 4 A latent non-collider $NC1$ is a direct child of another latent non-collider $NC2$ (both on the same path emerging in EX) only if:

- In all 2S-PCCs for which EX does not change, the observed children of $NC1$ always change with those of $NC2$ and also in a single 2S-PCC without the children of $NC2$; and
- In all 2S-PCCs for which a latent non-collider leaf descendant of EX does not change, the observed children of $NC2$ always change with those of $NC1$ and also in a single 2S-PCC without the children of $NC1$.

Proof Let $NC1$ and $NC2$ be latent non-collider descendants of EX (both on the same path emerging from EX), and $NC1$ be a direct child of $NC2$. A 2S-PCC may result from a 2S-MC

due to a minor value in: 1) a latent ancestor of $NC1$ (including $NC2$ itself), 2) $NC1$, or 3) a latent descendant of $NC1$. In the first type of such 2S-PCC (for which EX does not change), Lemma 4 (first part) guarantees that the children of $NC1$ and $NC2$ change together in (1) and do not change at all in (3), whereas in (2) only the observed children of $NC1$ change. Thus, the children of $NC1$ always change with the children of $NC2$, and in addition also in a single 2S-PCC in which the children of $NC2$ do not change.

In the second type of such 2S-PCC for which the observed children of the leaf latent non-collider descendant of EX do not change, Lemma 4 (second part) guarantees that EX changes value, and the children of $NC1$ and $NC2$ do not change at all in (1) and change together in (3), whereas in (2) only the observed children of $NC2$ change. Thus, the children of $NC2$ always change with the children of $NC1$, and in addition also in a single 2S-PCC in which the children of $NC1$ do not change. The same analysis is true for both a serial and diverging connection. ■

Appendix B. Setting a threshold for the maximal size of 2-order minor clusters (Section 4.4)

In this appendix, we describe the calculation of a 2-order minor cluster threshold (2MCT) on the maximal size of 2-order minor clusters (2-MCs) that were introduced in Section 4.4.

This threshold represents the maximal size of a minor cluster that corresponds to a 2-order minor value configuration (Definition 13), i.e., a minor cluster that represents exactly two endogenous variables in EN that have minor values. This threshold is separately calculated to each $EX_i \in EX$, when all endogenous variables in EN , except the two mentioned, have major values. This threshold is an approximation for the maximal probability of having minor values as a response to any ex in exactly two descendants of EX , where all other descendants of EX and the other exogenous variables in EX have major values. This approximation is derived from the product of the maximal minor local effects (Definition B.1) of two observed descendants of EX_i , the maximal major local effects (Definition B.1) of the other observed descendants of EX_i , the maximal major local effects of the descendants of the other exogenous variables in EX , and the maximal prior of all exogenous variables in EX . We define:

Definition B.1 A maximal major local effect on an observed child O_j of a latent parent Pa_i is the maximal major effect on O_j over all values pa_i' of Pa_i , such that $MaxMAE_i = \max_{pa_i'} MAE_i(pa_i')$. Similarly, a maximal minor local effect is the maximal minor effect over all values pa_i' of Pa_i , such that $MaxMIE_i = \max_{pa_i'} MIE_i(pa_i')$.

First, we find $MaxMAE_i$ and $MaxMIE_i$, which are the sorted vectors of $MaxMAE_i$ and $MaxMIE_i$ (Definition B.1) of all $O_j \in Ch_i$ (observed descendants of EX_i), respectively. These vectors include the maximal major local effects and the maximal minor local effects on the observed descendants of EX_i sorted from the highest to the lowest. Note that EX_i replaces the actual direct parent of an observed variable for calculating the maximal major and minor effects since the direct parent has not been identified and split yet from EX_i at this stage.

Using these maximal major and minor effects and their sorted vectors for EX_i , we can calculate the approximation of the threshold. First, the maximal probability of exactly two minor values among the descendants of EX_i can be approximated by:

$$\prod_{t=1}^2 \text{MaxMIEV}_i(t).$$

Second, the maximal probability of the other descendants of EX_i to have major values can be approximated by:

$$\prod_{t=1}^{|\text{Ch}_i|-2} \text{MaxMAEV}_i(t).$$

Third, the maximal probability of the other descendants of the other exogenous variables to have major values can be approximated by:

$$\prod_{EX_j \in \text{EX} \setminus \{EX_i, \text{Ch}_i\}} \text{MaxMAE}_t.$$

Fourth, the maximal prior of all the exogenous variables is represented by:

$$\prod_{EX_k \in \text{EX}} \text{max}_{ex'_k} P(EX_j = ex'_k).$$

Then, the threshold for the maximal size of 2-order minor clusters (measured by the number of patterns in such a cluster) for EX_i can be approximated by the product of all the above approximations multiplied by the data size N :

$$2MCT_i = N \prod_{t=1}^2 \text{MaxMIEV}_i(t) \prod_{t=1}^{|\text{Ch}_i|-2} \text{MaxMAEV}_i(t) \prod_{EX_j \in \text{EX} \setminus \{EX_i, \text{Ch}_i\}} \text{MaxMAE}_t \prod_{EX_k \in \text{EX}} \text{max}_{ex'_k} P(EX_j = ex'_k).$$

Appendix C. Assumptions LPCC makes and the meaning of their violation

Assumption	Essential?	If violated
Assumption 1 The underlying model is a Bayesian network, $\text{BN} = \langle G, \Theta \rangle$, encoding a discrete joint probability distribution P for a set of random variables $V = \text{LUO}$, where $G = \langle V, E \rangle$ is a directed acyclic graph (DAG) whose nodes V correspond to latents L and observed variables O , and E is the set of edges between nodes in G . Θ is the set of parameters, i.e., the conditional probabilities of variables in V given their parents. Assumption 2 No observed variable in O is an ancestor of any latent variable in L (the <i>measurement assumption</i> , Spirtes et al., 2000). Assumption 3 The measurement model of G is pure.	Yes [made also in similar algorithms, e.g., that of Silva et al. (2006) for continuous joint probability distributions].	If neither was investigated theoretically nor studied experimentally what LPCC returns, if the underlying model is not a BN (i.e., there are cycles in G), no latent variables exist in the domain, or an observed variable is an ancestor of a latent variable.
Assumption 4 The true model G is MIM, in which each latent has at least two observed children and may have latent parents.	No (only needed for the correctness of the learned model).	When the true causal model is pure, LPCC will identify it correctly (or find its pattern). However, when it is not pure, LPCC – similarly to BPC (Silva et al., 2006) – will learn a pure sub-model of the true model using two indicators for each latent (compared to three indicators per latent that are required by BPC). If a latent has only one observed child, LPCC will not identify this latent.
Assumption 5 A latent collider does not have any latent descendants (and thus cannot be a parent of another latent collider).	Yes [made also by Silva et al. (2006), which requires three indicators per latent].	If this assumption is violated, and a latent collider has latent descendants, but none of them is a collider, LPCC does not identify the latent descendants as separate and join them, along with their observed children, to the learned ancestor latent collider. The case in which this assumption is violated, and at least one of the latent descendants of the collider is a latent collider itself, needs further investigation.
Assumption 6 For every endogenous variable EN_i in G and every configuration pa'_i of EN_i 's parents Pa_i , there exists a certain value en'_i of EN_i , such that $P(EN_i = en'_i \text{Pa}_i = \text{pa}'_i) > P(EN_i = en'_i \text{Pa}_i = \text{pa}'_i)$ for every other value en''_i of EN_i . This assumption is related to the most probable explanation of a hypothesis given the data (Pearl, 1988).	No (only needed for the correctness of the learned model).	If more than one value of EN_i gets the maximal probability value given a configuration of parents, LPCC still learns a model because the implementation will randomly choose one of the values that maximize the probability as the most probable. However, the correctness of the algorithm will not be guaranteed.
Assumption 7 First, for every EN_i that is an observed variable or an endogenous latent non-collider and for every two values pa'_i and pa''_i of Pa_i , $\text{MAVEN}_i(\text{pa}'_i) \neq \text{MAVEN}_i(\text{pa}''_i)$. Second, for every C_j that is a latent collider and for every $\text{Pa}_j \in \text{Pa}_j$, there are at least two configurations pa'_j and pa''_j of Pa_j in which only the value of Pa_j is different and $\text{MAV}_{C_j}(\text{pa}'_j) \neq \text{MAV}_{C_j}(\text{pa}''_j)$.	Not essential but very reasonable.	Regarding the second part of the assumption first: if this assumption is violated, and a collider has the same major value for any value of one of its parents (while the values of the other parents are the same), then its correlation to this parent should be very weak, which challenges the existence of their connection in the domain, and of course, the ability of any learning algorithm to identify this connection. Although the first part of the assumption may be considered similarly (based on a parent-child correlation), it also invites further investigation.
Assumption 8 Latent colliders do not share exactly the same sets of exogenous ancestors.	Not essential but very reasonable.	If this assumption is violated, and several latent colliders share exactly the same set of exogenous ancestors, LPCC does not identify the latent colliders as separate and learns a single collider as the parent of all children of the latent colliders.

References

- N. Asbeh and B. Lerner. Learning latent variable models by pairwise cluster comparison. In *Proceedings of the 4th Asian Conference on Machine Learning, JMLR Workshop & Conference Proceedings*, pages 25:33–48, 2012.
- D. J. Bartholomew, F. Steele, I. Moustaki, and J. I. Galbraith. *The Analysis and Interpretation of Multivariate Data for Social Scientists (Texts in Statistical Science Series)*. Chapman & Hall/CRC Press, Boca Raton, Florida, USA, 2002.
- K. Bollen. *Structural Equation Models with Latent Variables*. John Wiley & Sons, New York, New York, 1989.
- A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society, B* 39:1–39, 1977.
- G. Elidan and N. Friedman. Learning the dimensionality of hidden variables. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*, pages 144–151, Seattle, Washington, 2001.
- G. Elidan, N. Lotter, N. Friedman, and D. Koller. Discovering hidden variables: A structure-based approach. In *Advances in Neural Information Processing Systems*, pages 13:479–485, 2000.
- H. B. Enderton. *Elements of Set Theory*. Academic Press, New York, New York, 1977.
- N. Friedman. The Bayesian structural EM algorithm. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 129–138, San Francisco, CA, 1998.
- S. Harneling and C. K. I. Williams. Greedy learning of binary latent trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:1087–1097, 2011.
- R. Klee. *Introduction to the Philosophy of Science: Cutting Nature at its Seams*. Oxford University Press, New York, New York, 1997.
- T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, New York, New York, 1997.
- J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann Press, San Mateo, California, 1988.
- J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, New York, New York, 2000.
- J. Pearl and T. Verma. A theory of inferred causation. In *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, pages 441–452, Cambridge, MA, 1991.
- R. Scheines, P. Spirtes, C. Glymour, C. Meek, and T. Richardson. The tetrad project: Constraint based aids to causal model specification. Technical report, Department of Philosophy, Carnegie-Mellon University, Pittsburgh, Pennsylvania, 1995.
- S. Shimizu, T. Inazumi, Y. Sogawa, A. Hyvarinen, Y. Kawahara, T. Washio, P. Hoyer, and K. Bollen. DirectedLINGAM: A direct method for learning a linear non-Gaussian structural equation model. *Journal of Machine Learning Research*, 12:1225–1248, 2011.
- R. Silva, R. Scheines, C. Glymour, and P. Spirtes. Learning the structure of linear latent variable models. *Journal of Machine Learning Research*, 7:191–246, 2006.
- C. Spearman. General intelligence objectively determined and measured. *American Journal of Psychology*, 15:201–293, 1904.
- P. Spirtes. Calculation of entailed rank constraints in partially non-linear and cyclic models. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence*, pages 606–615, Bellevue, Washington, 2013.
- P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction and Search*. MIT Press, New York, New York, 2nd edition, 2000.
- Y. Wang, N. L. Zhang, and T. Chen. Latent-tree models and approximate inference in Bayesian networks. *Journal of Artificial Intelligence Research*, 32:879–900, 2008.
- N. Zhang. Hierarchical latent class models for cluster analysis. *Journal of Machine Learning Research*, 5:697–723, 2004.

GenSVM: A Generalized Multiclass Support Vector Machine

Gerrit J.J. van den Burg
Patrick J.F. Groenen

*Econometric Institute
Erasmus University Rotterdam
P.O. Box 1738
3000 DR Rotterdam
The Netherlands*

BURG@ESE.EUR.NL
GROENEN@ESE.EUR.NL

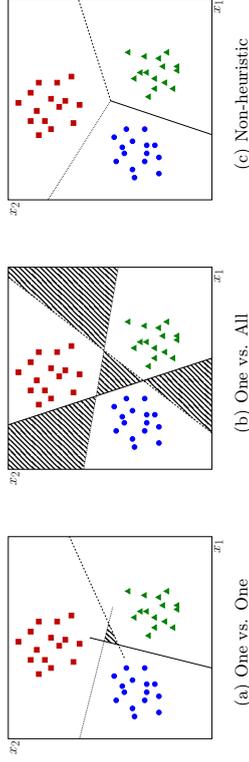


Figure 1: Illustration of ambiguity regions for common heuristic multiclass SVMs. In the shaded regions *ties* occur for which no classification rule has been explicitly trained. Figure (c) corresponds to an SVM where all classes are considered simultaneously, which eliminates any possible ties. Figures inspired by Statnikov et al. (2011).

Abstract

Traditional extensions of the binary support vector machine (SVM) to multiclass problems are either heuristics or require solving a large dual optimization problem. Here, a generalized multiclass SVM is proposed called GenSVM. In this method classification boundaries for a K -class problem are constructed in a $(K - 1)$ -dimensional space using a simplex encoding. Additionally, several different weightings of the misclassification errors are incorporated in the loss function, such that it generalizes three existing multiclass SVMs through a single optimization problem. An iterative majorization algorithm is derived that solves the optimization problem without the need of a dual formulation. This algorithm has the advantage that it can use warm starts during cross validation and during a grid search, which significantly speeds up the training phase. Rigorous numerical experiments compare linear GenSVM with seven existing multiclass SVMs on both small and large data sets. These comparisons show that the proposed method is competitive with existing methods in both predictive accuracy and training time, and that it significantly outperforms several existing methods on these criteria.

Keywords: support vector machines, SVM, multiclass classification, iterative majorization, MM algorithm, classifier comparison

1. Introduction

For binary classification, the support vector machine has shown to be very successful (Cortes and Vapnik, 1995). The SVM efficiently constructs linear or nonlinear classification boundaries and is able to yield a sparse solution through the so-called support vectors, that is, through those observations that are either not perfectly classified or are on the classification boundary. In addition, by regularizing the loss function the overfitting of the training data set is curbed. Due to its desirable characteristics several attempts have been made to extend the SVM to classification problems where the number of classes K is larger than two. Overall, these extensions differ considerably in the approach taken to include multiple classes. Three types of approaches for multiclass SVMs (MSVMs) can be distinguished.

First, there are heuristic approaches that use the binary SVM as an underlying classifier and decompose the K -class problem into multiple binary problems. The most commonly used heuristic is the one-vs-one (OvO) method where decision boundaries are constructed

between each pair of classes (Kreßel, 1999). OvO requires solving $K(K - 1)$ binary SVM problems, which can be substantial if the number of classes is large. An advantage of OvO is that the problems to be solved are smaller in size. On the other hand, the one-vs-all (OvA) heuristic constructs K classification boundaries, one separating each class from all the other classes (Vapnik, 1998). Although OvA requires fewer binary SVMs to be estimated, the complete data set is used for each classifier, which can create a high computational burden. Another heuristic approach is the directed acyclic graph (DAG) SVM proposed by Platt et al. (2000). DAGSVM is similar to the OvO approach except that the class prediction is done by successively voting away unlikely classes until only one remains. One problem with the OvO and OvA methods is that there are regions of the space for which class predictions are ambiguous, as illustrated in Figures 1a and 1b.

In practice, heuristic methods such as the OvO and OvA approaches are used more often than other multiclass SVM implementations. One of the reasons for this is that there are several software packages that efficiently solve the binary SVM, such as LibSVM (Chang and Lin, 2011). This package implements a variation of the sequential minimal optimization algorithm of Platt (1999). Implementations of other multiclass SVMs in high-level (statistical) programming languages are lacking, which reduces their use in practice.

The second type of extension of the binary SVM use error correcting codes. In these methods the problem is decomposed into multiple binary classification problems based on a constructed coding matrix that determines the grouping of the classes in a specific binary subproblem (Dietterich and Bakiri, 1995; Allwein et al., 2001; Crammer and Singer, 2002b). Error correcting code SVMs can thus be seen as a generalization of OvO and OvA. In Dietterich and Bakiri (1995) and Allwein et al. (2001), a coding matrix is constructed that determines which class instances are paired against each other for each binary SVM. Both approaches require that the coding matrix is determined beforehand. However, it is a priori

1. An exception to this is the method of Lee et al. (2004), for which an R implementation exists. See <http://www.stat.osu.edu/~ykLee/software.html>.

unclear how such a coding matrix should be chosen. In fact, as Crammer and Singer (2002b) show, finding the optimal coding matrix is an NP-complete problem.

The third type of approaches are those that optimize one loss function to estimate all class boundaries simultaneously, the so-called single machine approaches (Rifkin and Klautau, 2004). In the literature, such methods have been proposed by, among others, Weston and Watkins (1998), Bredeisen and Bennett (1999), Crammer and Singer (2002a), Lee et al. (2004), and Gherneur and Monfrini (2011). The method of Weston and Watkins (1998) yields a fairly large quadratic problem with a large number of slack variables; that is, $K - 1$ slack variables for each observation. The method of Crammer and Singer (2002a) reduces this number of slack variables by only penalizing the largest misclassification error. In addition, their method does not include a bias term in the decision boundaries, which is advantageous for solving the dual problem. Interestingly, this approach does not reduce parsimoniously to the binary SVM for $K = 2$. The method of Lee et al. (2004) uses a sum-to-zero constraint on the decision functions to reduce the dimensionality of the problem. This constraint effectively means that the solution of the multiclass SVM lies in a $(K - 1)$ -dimensional subspace of the full K dimensions considered. The size of the margins is reduced according to the number of classes, such that asymptotic convergence is obtained to the Bayes optimal decision boundary when the regularization term is ignored (Rifkin and Klautau, 2004). Finally, the method of Gherneur and Monfrini (2011) is a quadratic extension of the method developed by Lee et al. (2004). This extension keeps the sum-to-zero constraint on the decision functions, drops the nonnegativity constraint on the slack variables, and adds a quadratic function of the slack variables to the loss function. This means that at the optimum the slack variables are only positive on average, which differs from common SVM formulations.

The existing approaches to multiclass SVMs suffer from several problems. All current single machine multiclass extensions of the binary SVM rely on solving a potentially large dual optimization problem. This can be disadvantageous when a solution has to be found in a small amount of time, since iteratively improving the dual solution does not guarantee that the primal solution is improved as well. Thus, stopping early can lead to poor predictive performance. In addition, the dual of such single machine approaches should be solvable quickly in order to compete with existing heuristic approaches.

Almost all single machine approaches rely on misclassifications of the observed class with each of the other classes. By simply summing these misclassification errors (as in Lee et al., 2004) observations with multiple errors contribute more than those with a single misclassification do. Consequently, observations with multiple misclassifications have a stronger influence on the solution than those with a single misclassification, which is not a desirable property for a multiclass SVM, as it overemphasizes objects that are misclassified with respect to multiple classes. Here, it is argued that there is no reason to penalize certain misclassification regions more than others.

Single machine approaches are preferred for their ability to capture the multiclass classification problem in a single model. A parallel can be drawn here with multinomial regression and logistic regression. In this case, multinomial regression reduces exactly to the binary logistic regression method when $K = 2$, both techniques are single machine approaches, and many of the properties of logistic regression extend to multinomial regression. Therefore,

it can be considered natural to use a single machine approach for the multiclass SVM that reduces parsimoniously to the binary SVM when $K = 2$.

The idea of casting the multiclass SVM problem to $K - 1$ dimensions is appealing, since it reduces the dimensionality of the problem and is also present in other multiclass classification methods such as multinomial regression and linear discriminant analysis. However, the sum-to-zero constraint employed by Lee et al. (2004) creates an additional burden on the dual optimization problem (Dogan et al., 2011). Therefore, it would be desirable to cast the problem to $K - 1$ dimensions in another manner. Below a simplex encoding will be introduced to achieve this goal. The simplex encoding for multiclass SVMs has been proposed earlier by Hill and Doucet (2007) and Mroueh et al. (2012), although the method outlined below differs from these two approaches. Note that the simplex coding approach by Mroueh et al. (2012) was shown to be equivalent to that of Lee et al. (2004) by Avila Pires et al. (2013). An advantage of the simplex encoding is that in contrast to methods such as O^*O and O^*A , there are no regions of ambiguity in the prediction space (see Figure 1c). In addition, the low dimensional projection also has advantages for understanding the method, since it allows for a geometric interpretation. The geometric interpretation of existing single machine multiclass SVMs is often difficult since most are based on a dual optimization approach with little attention for a primal problem based on hinge errors.

A new flexible and general multiclass SVM is proposed, called GenSVM. This method uses the simplex encoding to formulate the multiclass SVM problem as a single optimization problem that reduces to the binary SVM when $K = 2$. By using a flexible hinge function and an ℓ_p norm of the errors the GenSVM loss function incorporates three existing multiclass SVMs that use the sum of the hinge errors, and extends these methods. In the linear version of GenSVM, $K - 1$ linear combinations of the features are estimated next to the bias terms. In the nonlinear version, kernels can be used in a similar manner as can be done for binary SVMs. The resulting GenSVM loss function is convex in the parameters to be estimated. For this loss function an iterative majorization (IM) algorithm will be derived with guaranteed descent to the global minimum. By solving the optimization problem in the primal it is possible to use warm starts during a hyperparameter grid search or during cross validation, which makes the resulting algorithm very competitive in total training time, even for large data sets.

To evaluate its performance, GenSVM is compared to seven of the multiclass SVMs described above on several small data sets and one large data set. The smaller data sets are used to assess the classification accuracy of GenSVM, whereas the large data set is used to verify feasibility of GenSVM for large data sets. Due to the computational cost of these rigorous experiments only comparisons of linear multiclass SVMs are performed, and experiments on nonlinear MSVMs are considered outside the scope of this paper. Existing comparisons of multiclass SVMs in the literature do not determine any statistically significant differences in performance between classifiers, and resort to tables of accuracy rates for the comparisons (for instance Hsu and Lin, 2002). Using suggestions from the benchmark literature predictive performance and training time of all classifiers is compared using performance profiles and rank tests. The rank tests are used to uncover statistically significant differences between classifiers.

This paper is organized as follows. Section 2 introduces the novel generalized multiclass SVM. In Section 3, features of the iterative majorization theory are reviewed and a number

of useful properties are highlighted. Section 4 derives the IM algorithm for GenSVM, and presents pseudocode for the algorithm. Extensions of GenSVM to nonlinear classification boundaries are discussed in Section 5. A numerical comparison of GenSVM with existing multiclass SVMs on empirical data sets is done in Section 6. Section 7 concludes the paper.

2. GenSVM

Before introducing GenSVM formally, consider a small illustrative example of a hypothetical data set of $n = 90$ objects with $K = 3$ classes and $m = 2$ attributes. Figure 2a shows the data set in the space of these two attributes x_1 and x_2 , with different classes denoted by different symbols. Figure 2b shows the $(K - 1)$ -dimensional simplex encoding of the data after an additional RBF kernel transformation has been applied and the mapping has been optimized to minimize misclassification errors. In this figure, the triangle shown in the center corresponds to a regular K -simplex in $K - 1$ dimensions, and the solid lines perpendicular to the faces of this simplex are the decision boundaries. This $(K - 1)$ -dimensional space will be referred to as the *simplex space* throughout this paper. The mapping from the input space to this simplex space is optimized by minimizing the misclassification errors, which are calculated by measuring the distance of an object to the decision boundaries in the simplex space. Prediction of a class label is also done in this simplex space, by finding the nearest simplex vertex for the object. Figure 2c illustrates the decision boundaries in the original space of the input attributes x_1 and x_2 . In Figures 2b and 2c, the support vectors can be identified as the objects that lie on or beyond the dashed margin lines of their associated class. Note that the use of the simplex encoding ensures that for every point in the predictor space a class is predicted, hence no ambiguity regions can exist in the GenSVM solution.

The misclassification errors are formally defined as follows. Let $\mathbf{x}_i \in \mathbb{R}^m$ be an object vector corresponding to m attributes, and let y_i denote the class label of object i with $y_i \in \{1, \dots, K\}$, for $i \in \{1, \dots, n\}$. Furthermore, let $\mathbf{W} \in \mathbb{R}^{m \times (K-1)}$ be a weight matrix, and define a translation vector $\mathbf{t} \in \mathbb{R}^{K-1}$ for the bias terms. Then, object i is represented in the $(K - 1)$ -dimensional simplex space by $\mathbf{s}'_i = \mathbf{x}'_i \mathbf{W} + \mathbf{t}$. Note that here the *linear* version of GenSVM is described, the nonlinear version is described in Section 5.

To obtain the misclassification error of an object, the corresponding simplex space vector \mathbf{s}'_i is projected on each of the decision boundaries that separate the true class of an object from another class. For the errors to be proportional with the distance to the decision boundaries, a regular K -simplex in \mathbb{R}^{K-1} is used with distance 1 between each pair of vertices. Let \mathbf{U}_K be the $K \times (K - 1)$ coordinate matrix of this simplex, where a row \mathbf{u}_k of \mathbf{U}_K gives the coordinates of a single vertex k . Then, it follows that with $k \in \{1, \dots, K\}$ and $l \in \{1, \dots, K - 1\}$ the elements of \mathbf{U}_K are given by

$$u_{kl} = \begin{cases} -\frac{1}{\sqrt{2(l^2+l)}} & \text{if } k \leq l \\ \frac{l}{\sqrt{2(l^2+l)}} & \text{if } k = l + 1 \\ 0 & \text{if } k > l + 1. \end{cases} \quad (1)$$

See Appendix A for a derivation of this expression. Figure 3 shows an illustration of how the misclassification errors are computed for a single object. Consider object A with true class

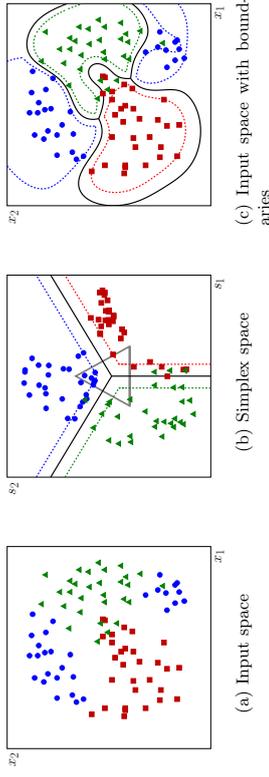


Figure 2: Illustration of GenSVM for a 2D data set with $K = 3$ classes. In (a) the original data is shown, with different symbols denoting different classes. Figure (b) shows the mapping of the data to the $(K - 1)$ -dimensional simplex space, after an additional RBF kernel mapping has been applied and the optimal solution has been determined. The decision boundaries in this space are fixed as the perpendicular bisectors of the faces of the simplex, which is shown as the gray triangle. Figure (c) shows the resulting boundaries mapped back to the original input space, as can be seen by comparing with (a). In Figures (b) and (c) the dashed lines show the margins of the SVM solution.

$y_A = 2$. It is clear that object A is misclassified as it is not located in the shaded area that has vertex \mathbf{u}_2 as the nearest vertex. The boundaries of the shaded area are given by the perpendicular bisectors of the edges of the simplex between vertices \mathbf{u}_2 and \mathbf{u}_1 and between vertices \mathbf{u}_2 and \mathbf{u}_3 , and form the decision boundaries for class 2. The error for object A is computed by determining the distance from the object to each of these decision boundaries. Let $q_A^{(21)}$ and $q_A^{(23)}$ denote these distances to the class boundaries, which are obtained by projecting $\mathbf{s}'_A = \mathbf{x}'_A \mathbf{W} + \mathbf{t}$ on $\mathbf{u}_2 - \mathbf{u}_1$ and $\mathbf{u}_2 - \mathbf{u}_3$ respectively, as illustrated in the figure. Generalizing this reasoning, scalars $q_i^{(kj)}$ can be defined to measure the projection distance of object i onto the boundary between class k and j in the simplex space, as

$$q_i^{(kj)} = (\mathbf{x}'_i \mathbf{W} + \mathbf{t}')(\mathbf{u}_k - \mathbf{u}_j). \quad (2)$$

It is required that the GenSVM loss function is both general and flexible, such that it can easily be tuned for the specific data set at hand. To achieve this, a loss function is constructed with a number of different weightings, each with a specific effect on the object distances $q_i^{(kj)}$. In the proposed loss function, flexibility is added through the use of the Huber hinge function instead of the absolute hinge function, and by using the ℓ_p norm of the hinge errors instead of the sum. The motivation for these choices follows.

As is customary for SVMs a hinge loss is used to ensure that instances that do not cross their class margin will yield zero error. Here, the flexible and continuous Huber hinge loss

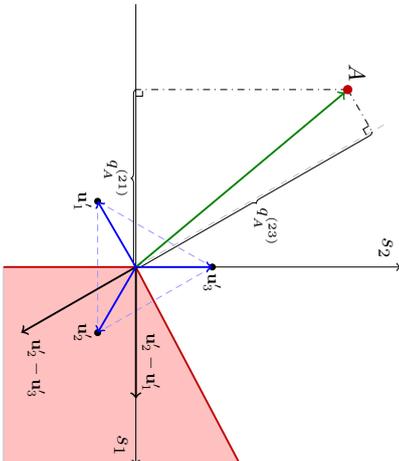


Figure 3: Graphical illustration of the calculation of distances $q_i^{(y_A j)}$ for an object A with $y_A = 2$ and $K = 3$. The figure shows the situation in the $(K - 1)$ -dimensional space. The distance $q_A^{(21)}$ is calculated by projecting $s'_A = X'_A \mathbf{W} + \mathbf{t}'$ on $\mathbf{u}_2 - \mathbf{u}_1$, and the distance $q_A^{(23)}$ is found by projecting s'_A on $\mathbf{u}_2 - \mathbf{u}_3$. The boundary between the class 1 and class 3 regions has been omitted for clarity, but lies along \mathbf{u}_2 .

is used (after the Huber error in robust statistics, see Huber, 1964), which is defined as

$$h(q) = \begin{cases} 1 - q - \frac{\kappa + 1}{2} & \text{if } q \leq -\kappa \\ \frac{1}{2(\kappa + 1)}(1 - q)^2 & \text{if } q \in (-\kappa, 1] \\ 0 & \text{if } q > 1, \end{cases} \quad (3)$$

with $\kappa > -1$. The Huber hinge loss has been independently introduced in Chapelle (2007), Rosset and Zhu (2007), and Groenen et al. (2008). This hinge error is zero when an instance is classified correctly with respect to its class margin. However, in contrast to the absolute hinge error, it is continuous due to a quadratic region in the interval $(-\kappa, 1]$. This quadratic region allows for a softer weighting of objects close to the decision boundary. Additionally, the smoothness of the Huber hinge error is a desirable property for the iterative majorization algorithm derived in Section 4.1. Note that the Huber hinge error approaches the absolute hinge for $\kappa \downarrow -1$, and the quadratic hinge for $\kappa \rightarrow \infty$.

The Huber hinge error is applied to each of the distances $q_i^{(y_A j)}$, for $j \neq y_i$. Thus, no error is counted when the object is correctly classified. For each of the objects, errors with respect to the other classes are summed using an ℓ_p norm to obtain the total object error

$$\left(\sum_{\substack{j=1 \\ j \neq y_i}}^K h^p \left(q_i^{(y_A j)} \right) \right)^{1/p}$$

The ℓ_p norm is added to provide a form of regularization on Huber weighted errors for instances that are misclassified with respect to multiple classes. As argued in the Introduction, simply summing misclassification errors can lead to overemphasizing of instances with multiple misclassification errors. By adding an ℓ_p norm of the hinge errors the influence of such instances on the loss function can be tuned. With the addition of the ℓ_p norm on the hinge errors it is possible to illustrate how GenSVM generalizes existing methods. For instance, with $p = 1$ and $\kappa \downarrow -1$, the loss function solves the same problem as the method of Lee et al. (2004). Next, for $p = 2$ and $\kappa \downarrow -1$ it resembles that of Guenieur and Monfimi (2011). Finally, for $p = \infty$ and $\kappa \downarrow -1$ the ℓ_p norm reduces to the max norm of the hinge errors, which corresponds to the method of Crammer and Singer (2002a). Note that in each case the value of κ can additionally be varied to include an even broader family of loss functions.

To illustrate the effects of p and κ on the total object error, refer to Figure 4. In Figures 4a and 4b, the value of p is set to $p = 1$ and $p = 2$ respectively, while maintaining the absolute hinge error using $\kappa = -0.95$. A reference point is plotted at a fixed position in the area of the simplex space where there is a nonzero error with respect to two classes. It can be seen from this reference point that the value of the combined error is higher when $p = 1$. With $p = 2$ the combined error at the reference point approximates the Euclidean distance to the margin, when $\kappa \downarrow -1$. Figures 4a, 4c, and 4d show the effect of varying κ . It can be seen that the error near the margin becomes more quadratic with increasing κ . In fact, as κ increases the error approaches the squared Euclidean distance to the margin, which can be used to obtain a quadratic hinge multiclass SVM. Both of these effects will become stronger when the number of classes increases, as increasingly more objects will have errors with respect to more than one class.

Next, let $\rho_i \geq 0$ denote optional object weights, which are introduced to allow flexibility in the way individual objects contribute to the total loss function. With these individual weights it is possible to correct for different group sizes, or to give additional weights to misclassifications of certain classes. When correcting for group sizes, the weights can be chosen as

$$\rho_i = \frac{n}{n_k K}, \quad i \in G_k, \quad (4)$$

where $G_k = \{i : y_i = k\}$ is the set of objects belonging to class k , and $n_k = |G_k|$. The complete GenSVM loss function combining all n objects can now be formulated as

$$L_{\text{MSVM}}(\mathbf{W}, \mathbf{t}) = \frac{1}{n} \sum_{k=1}^K \sum_{i \in G_k} \rho_i \left(\sum_{j \neq k} h^p \left(q_i^{(k j)} \right) \right)^{1/p} + \lambda \text{tr } \mathbf{W}' \mathbf{W}, \quad (5)$$

where $\lambda \text{tr } \mathbf{W}' \mathbf{W}$ is the penalty term to avoid overfitting, and $\lambda > 0$ is the regularization parameter. Note that for the case where $K = 2$, the above loss function reduces to the loss function for binary SVM given in Groenen et al. (2008), with Huber hinge errors.

The outline of a proof for the convexity of the loss function in (5) is given. First, note that the distances $q_i^{(k j)}$ in the loss function are affine in \mathbf{W} and \mathbf{t} . Hence, if the loss function is convex in $q_i^{(k j)}$ it is convex in \mathbf{W} and \mathbf{t} as well. Second, the Huber hinge function is trivially convex in $q_i^{(k j)}$, since each separate piece of the function is convex, and the Huber

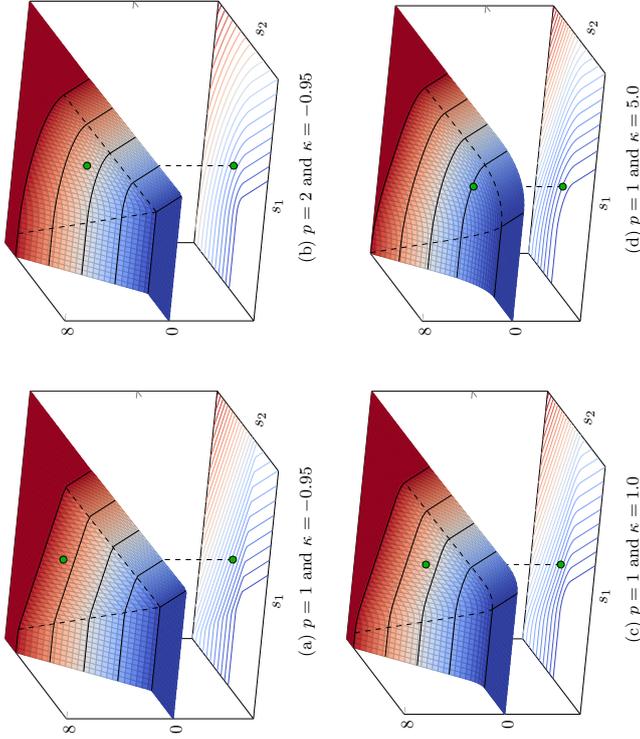


Figure 4: Illustration of the ℓ_p norm of the Huber weighted errors. Comparing figures (a) and (b) shows the effect of the ℓ_p norm. With $p = 1$ objects that have errors w.r.t. both classes are penalized more strongly than those with only one error, whereas with $p = 2$ this is not the case. Figures (a), (c), and (d) compare the effect of the κ parameter, with $p = 1$. This shows that with a large value of κ , the errors close to the boundary are weighted quadratically. Note that s_1 and s_2 indicate the dimensions of the simplex space.

hinge is continuous. Third, the ℓ_p norm is a convex function by the Minkowski inequality, and it is monotonically increasing by definition. Thus, it follows that the ℓ_p norm of the Huber weighted instance errors is convex (see for instance Rockafellar, 1997). Next, since it is required that the weights ρ_i are non-negative, the sum in the first term of (5) is a convex combination. Finally, the penalty term can also be shown to be convex, since $\text{tr } \mathbf{W}^* \mathbf{W}$ is the square of the Frobenius norm of \mathbf{W} , and it is required that $\lambda > 0$. Thus, it holds that the loss function in (5) is convex in \mathbf{W} and \mathbf{t} .

Predicting class labels in GenSVM can be done as follows. Let $(\mathbf{W}^*, \mathbf{t}^*)$ denote the parameters that minimize the loss function. Predicting the class label of an unseen sample \mathbf{x}'_{n+1} can then be done by first mapping it to the simplex space, using the optimal projection: $\mathbf{s}'_{n+1} = \mathbf{x}'_{n+1} \mathbf{W}^* + \mathbf{t}^*$. The predicted class label is then simply the label corresponding to

the nearest simplex vertex as measured by the squared Euclidean norm, or

$$\hat{y}_{n+1} = \arg \min_k \|\mathbf{s}'_{n+1} - \mathbf{u}'_k\|^2, \quad \text{for } k = 1, \dots, K. \quad (6)$$

3. Iterative Majorization

To minimize the loss function given in (5), an iterative majorization (IM) algorithm will be derived. Iterative majorization was first described by Weiszfeld (1937), however the first application of the algorithm in the context of a line search comes from Ortega and Rheinboldt (1970, p. 253–255). During the late 1970s, the method was independently developed by De Leeuw (1977) as part of the SMACOF algorithm for multidimensional scaling, and by Voss and Eckhardt (1980) as a general minimization method. For the reader unfamiliar with the iterative majorization algorithm a more detailed description has been included in Appendix B and further examples can be found in for instance Hunter and Lange (2004).

The asymptotic convergence rate of the IM algorithm is linear, which is less than that of the Newton-Raphson algorithm (De Leeuw, 1994). However, the largest improvements in the loss function will occur in the first few steps of the iterative majorization algorithm, where the asymptotic linear rate does not apply (Havel, 1991). This property will become very useful for GenSVM as it allows for a quick approximation to the exact SVM solution in few iterations.

There is no straightforward technique for deriving the majorization function for any given function. However, in the next section the derivation of the majorization function for the GenSVM loss function is presented using an “outside-in” approach. In this approach, each function that constitutes the loss function is majorized separately and the majorization functions are combined. Two properties of majorization functions that are useful for this derivation are now formally defined. In these expressions, \bar{x} is a supporting point, as defined in Appendix B.

P1. Let $f_1 : \mathcal{Y} \rightarrow \mathcal{Z}$, $f_2 : \mathcal{X} \rightarrow \mathcal{Y}$, and define $f = f_1 \circ f_2 : \mathcal{X} \rightarrow \mathcal{Z}$, such that for $x \in \mathcal{X}$, $f(x) = f_1(f_2(x))$. If $g_1 : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathcal{Z}$ is a majorization function of f_1 , then $g : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{Z}$ defined as $g = g_1 \circ f_2$ is a majorization function of f . Thus for $x, \bar{x} \in \mathcal{X}$ it holds that $g(x, \bar{x}) = g_1(f_2(x), f_2(\bar{x}))$ is a majorization function of $f(x)$ at \bar{x} .

P2. Let $f_i : \mathcal{X} \rightarrow \mathcal{Z}$ and define $f : \mathcal{X} \rightarrow \mathcal{Z}$ such that $f(x) = \sum_i a_i f_i(x)$ for $x \in \mathcal{X}$, with $a_i \geq 0$ for all i . If $g_i : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{Z}$ is a majorization function for f_i at a point $\bar{x} \in \mathcal{X}$, then $g : \mathcal{X} \times \mathcal{X} \rightarrow \mathcal{Z}$ given by $g(x, \bar{x}) = \sum_i a_i g_i(x, \bar{x})$ is a majorization function of f .

Proofs of these properties are omitted, as they follow directly from the requirements for a majorization function given in Appendix B. The first property allows for the use of the “outside-in” approach to majorization, as will be illustrated in the next section.

4. GenSVM Optimization and Implementation

In this section, a quadratic majorization function for GenSVM will be derived. Although it is possible to derive a majorization algorithm for general values of the ℓ_p norm parameter,²

² For a majorization algorithm of the ℓ_p norm with $p \geq 2$, see Groenen et al. (1999).

the following derivation will restrict this value to the interval $p \in [1, 2]$ since this simplifies the derivation and avoids the issue that quadratic majorization can become slow for $p > 2$. Pseudocode for the derived algorithm will be presented, as well as an analysis of the computational complexity of the algorithm. Finally, an important remark on the use of warm starts in the algorithm is given.

4.1 Majorization Derivation

To shorten the notation, define

$$\begin{aligned} \mathbf{V} &= [\mathbf{t} \ \mathbf{W}]^T, \\ \mathbf{z}'_i &= [1 \ \mathbf{x}'_i], \\ \delta_{kj} &= \mathbf{u}_k - \mathbf{u}_j, \end{aligned}$$

such that $q_i^{(kj)} = \mathbf{z}'_i \mathbf{V} \delta_{kj}$. With this notation it becomes sufficient to optimize the loss function with respect to \mathbf{V} . Formulated in this manner (5) becomes

$$L_{\text{MSVM}}(\mathbf{V}) = \frac{1}{n} \sum_{k=1}^K \sum_{i \in G_k} \rho_i \left(\sum_{j \neq k} h^p(q_i^{(kj)}) \right)^{1/p} + \lambda \text{tr} \mathbf{V}' \mathbf{J} \mathbf{V}, \quad (7)$$

where \mathbf{J} is an $m+1$ diagonal matrix with $J_{i,i} = 1$ for $i > 1$ and zero elsewhere. To derive a majorization function for this expression the ‘‘outside-in’’ approach will be used, together with the properties of majorization functions. In what follows, variables with a bar denote supporting points for the IM algorithm. The goal of the derivation is to find a quadratic majorization function in \mathbf{V} such that

$$L_{\text{MSVM}}(\mathbf{V}) \leq \text{tr} \mathbf{V}' \mathbf{Z}' \mathbf{A} \mathbf{Z}' \mathbf{V} - 2 \text{tr} \mathbf{V}' \mathbf{Z}' \mathbf{B} + C, \quad (8)$$

where \mathbf{A} , \mathbf{B} , and C are coefficients of the majorization depending on $\bar{\mathbf{V}}$. The matrix \mathbf{Z} is simply the $n \times (m+1)$ matrix with rows \mathbf{z}'_i .

Property P2 above means that the summation over instances in the loss function can be ignored for now. Moreover, the regularization term is quadratic in \mathbf{V} , and thus requires no majorization. The outermost function for which a majorization function has to be found is thus the ℓ_p norm of the Huber hinge errors. Hence it is possible to consider the function $f(\mathbf{x}) = \|\mathbf{x}\|_p$ for majorization. A majorization function for $f(\mathbf{x})$ can be constructed, but a discontinuity in the derivative at $\mathbf{x} = \mathbf{0}$ will remain (Tatsui and Morikawa, 2012).

To avoid the discontinuity in the derivative of the ℓ_p norm, the following inequality is needed (Hardy et al., 1934, eq. 2.10.3)

$$\left(\sum_{j \neq k} h^p(q_i^{(kj)}) \right)^{1/p} \leq \sum_{j \neq k} h(q_i^{(kj)}).$$

This inequality can be used as a majorization function only if equality holds at the supporting point

$$\left(\sum_{j \neq k} h^p(\bar{q}_i^{(kj)}) \right)^{1/p} = \sum_{j \neq k} h(\bar{q}_i^{(kj)}).$$

It is not difficult to see that this only holds if at most one of the $h(\bar{q}_i^{(kj)})$ errors is nonzero for $j \neq k$. Thus an indicator variable ε_i is introduced which is 1 if at most one of these errors is nonzero, and 0 otherwise. Then it follows that

$$\begin{aligned} L_{\text{MSVM}}(\mathbf{V}) &\leq \frac{1}{n} \sum_{k=1}^K \sum_{i \in G_k} \rho_i \left[\varepsilon_i \sum_{j \neq k} h(q_i^{(kj)}) + (1 - \varepsilon_i) \left(\sum_{j \neq k} h^p(q_i^{(kj)}) \right)^{1/p} \right] \\ &\quad + \lambda \text{tr} \mathbf{V}' \mathbf{J} \mathbf{V}. \end{aligned} \quad (9)$$

Now, the next function for which a majorization needs to be found is $f_1(x) = x^{1/p}$. From the inequality $a^\alpha b^\beta \leq \alpha a + \beta b$, with $\alpha + \beta = 1$ (Hardy et al., 1934, Theorem 37), a linear majorization inequality can be constructed for this function by substituting $a = x$, $b = \bar{x}$, $\alpha = 1/p$ and $\beta = 1 - 1/p$ (Groenen and Heiser, 1996). This yields

$$f_1(x) = x^{1/p} \leq \frac{1}{p} \bar{x}^{1/p-1} x + \left(1 - \frac{1}{p}\right) \bar{x}^{1/p} = g_1(x; \bar{x}).$$

Applying this majorization and using property P1 gives

$$\begin{aligned} \left(\sum_{j \neq k} h^p(q_i^{(kj)}) \right)^{1/p} &\leq \frac{1}{p} \left(\sum_{j \neq k} h^p(\bar{q}_i^{(kj)}) \right)^{1/p-1} \left(\sum_{j \neq k} h^p(q_i^{(kj)}) \right) \\ &\quad + \left(1 - \frac{1}{p}\right) \left(\sum_{j \neq k} h^p(\bar{q}_i^{(kj)}) \right)^{1/p}. \end{aligned}$$

Plugging this into (9) and collecting terms yields

$$\begin{aligned} L_{\text{MSVM}}(\mathbf{V}) &\leq \frac{1}{n} \sum_{k=1}^K \sum_{i \in G_k} \rho_i \left[\varepsilon_i \sum_{j \neq k} h(q_i^{(kj)}) + (1 - \varepsilon_i) \omega_i \sum_{j \neq k} h^p(q_i^{(kj)}) \right] \\ &\quad + \Gamma^{(1)} + \lambda \text{tr} \mathbf{V}' \mathbf{J} \mathbf{V}, \end{aligned} \quad (10)$$

with

$$\omega_i = \frac{1}{p} \left(\sum_{j \neq k} h^p(\bar{q}_i^{(kj)}) \right)^{1/p-1}.$$

The constant $\Gamma^{(1)}$ contains all terms that only depend on previous errors $\bar{q}_i^{(kj)}$. The next majorization step by the ‘‘outside-in’’ approach is to find a quadratic majorization function for $f_2(x) = h^p(x)$, of the form

$$f_2(x) = h^p(x) \leq a(\bar{x}, p)x^2 - 2b(\bar{x}, p)x + c(\bar{x}, p) = g_2(x; \bar{x}).$$

Since this derivation is mostly an algebraic exercise it has been moved to Appendix C. In the remainder of this derivation, $a_{i,jk}^{(p)}$ will be used to abbreviate $a(\bar{q}_i^{(kj)}, p)$, with similar

abbreviations for b and c . Using these majorizations and making the dependence on \mathbf{V} explicit by substituting $q_i^{(kj)} = \mathbf{z}_i^t \mathbf{V} \boldsymbol{\delta}_{kj}$ gives

$$\begin{aligned} L_{\text{MSVM}}(\mathbf{V}) &\leq \frac{1}{n} \sum_{k=1}^K \sum_{i \in G_k} \rho_i \varepsilon_i \sum_{j \neq k} \left[a_{ijk}^{(1)} \mathbf{z}_i^t \mathbf{V} \boldsymbol{\delta}_{kj} \boldsymbol{\delta}_{kj}' \mathbf{V}' \mathbf{z}_i - 2b_{ijk}^{(1)} \mathbf{z}_i^t \mathbf{V} \boldsymbol{\delta}_{kj} \right] \\ &\quad + \frac{1}{n} \sum_{k=1}^K \sum_{i \in G_k} \rho_i (1 - \varepsilon_i) \omega_i \sum_{j \neq k} \left[a_{ijk}^{(p)} \mathbf{z}_i^t \mathbf{V} \boldsymbol{\delta}_{kj} \boldsymbol{\delta}_{kj}' \mathbf{V}' \mathbf{z}_i - 2b_{ijk}^{(p)} \mathbf{z}_i^t \mathbf{V} \boldsymbol{\delta}_{kj} \right] \\ &\quad + \Gamma^{(2)} + \lambda \text{tr } \mathbf{V}' \mathbf{J} \mathbf{V}, \end{aligned} \quad (14)$$

where $\Gamma^{(2)}$ again contains all constant terms. Due to dependence on the matrix $\boldsymbol{\delta}_{ij} \boldsymbol{\delta}_{kj}'$, the above majorization function is not yet in the desired quadratic form of (8). However, since the maximum eigenvalue of $\boldsymbol{\delta}_{ij} \boldsymbol{\delta}_{kj}'$ is 1 by definition of the simplex coordinates, it follows that the matrix $\boldsymbol{\delta}_{ij} \boldsymbol{\delta}_{kj}' - \mathbf{I}$ is negative semidefinite. Hence, it can be shown that the inequality $\mathbf{z}_i^t (\mathbf{V} - \bar{\mathbf{V}}) (\boldsymbol{\delta}_{ij} \boldsymbol{\delta}_{kj}' - \mathbf{I}) (\mathbf{V} - \bar{\mathbf{V}})' \mathbf{z}_i \leq 0$ holds (Bijleveld and De Leeuw, 1991, Theorem 4). Rewriting this gives the majorization inequality

$$\mathbf{z}_i^t \mathbf{V} \boldsymbol{\delta}_{ij} \boldsymbol{\delta}_{kj}' \mathbf{V}' \mathbf{z}_i \leq \mathbf{z}_i^t \mathbf{V} \mathbf{V}' \mathbf{z}_i - 2\mathbf{z}_i^t \mathbf{V} (\mathbf{I} - \boldsymbol{\delta}_{ij} \boldsymbol{\delta}_{kj}') \bar{\mathbf{V}} \mathbf{z}_i + \mathbf{z}_i^t \bar{\mathbf{V}} (\mathbf{I} - \boldsymbol{\delta}_{ij} \boldsymbol{\delta}_{kj}') \bar{\mathbf{V}}' \mathbf{z}_i.$$

With this inequality the majorization inequality becomes

$$\begin{aligned} L_{\text{MSVM}}(\mathbf{V}) &\leq \frac{1}{n} \sum_{k=1}^K \sum_{i \in G_k} \rho_i \mathbf{z}_i^t \mathbf{V} (\mathbf{V}' - 2\bar{\mathbf{V}}') \mathbf{z}_i \sum_{j \neq k} \left[\varepsilon_i a_{ijk}^{(1)} + (1 - \varepsilon_i) \omega_i a_{ijk}^{(p)} \right] \\ &\quad - \frac{2}{n} \sum_{k=1}^K \sum_{i \in G_k} \rho_i \mathbf{z}_i^t \mathbf{V} \sum_{j \neq k} \left[\varepsilon_i \left(b_{ijk}^{(1)} - a_{ijk}^{(1)} \bar{q}_i^{(kj)} \right) \right. \\ &\quad \left. + (1 - \varepsilon_i) \omega_i \left(b_{ijk}^{(p)} - a_{ijk}^{(p)} \bar{q}_i^{(kj)} \right) \right] \boldsymbol{\delta}_{kj} \\ &\quad + \Gamma^{(3)} + \lambda \text{tr } \mathbf{V}' \mathbf{J} \mathbf{V}, \end{aligned} \quad (11)$$

where $\bar{q}_i^{(kj)} = \mathbf{z}_i^t \bar{\mathbf{V}} \boldsymbol{\delta}_{kj}$. This majorization function is quadratic in \mathbf{V} and can thus be used in the IM algorithm. To derive the first-order condition used in the update step of the IM algorithm (step 2 in Appendix B), matrix notation for the above expression is introduced. Let \mathbf{A} be an $n \times n$ diagonal matrix with elements α_i , and let \mathbf{B} be an $n \times (K - 1)$ matrix with rows $\boldsymbol{\beta}_i$, where

$$\alpha_i = \frac{1}{n} \sum_{j \neq k} \left[\varepsilon_i a_{ijk}^{(1)} + (1 - \varepsilon_i) \omega_i a_{ijk}^{(p)} \right], \quad (12)$$

$$\boldsymbol{\beta}_i = \frac{1}{n} \sum_{j \neq k} \left[\varepsilon_i \left(b_{ijk}^{(1)} - a_{ijk}^{(1)} \bar{q}_i^{(kj)} \right) + (1 - \varepsilon_i) \omega_i \left(b_{ijk}^{(p)} - a_{ijk}^{(p)} \bar{q}_i^{(kj)} \right) \right] \boldsymbol{\delta}_{kj}. \quad (13)$$

Then the majorization function of $L_{\text{MSVM}}(\mathbf{V})$ given in (11) can be written as

$$\begin{aligned} L_{\text{MSVM}}(\mathbf{V}) &\leq \text{tr} (\mathbf{V} - 2\bar{\mathbf{V}})' \mathbf{Z}' \mathbf{A} \mathbf{Z} \mathbf{V} - 2 \text{tr } \mathbf{B}' \mathbf{Z} \mathbf{V} + \Gamma^{(3)} + \lambda \text{tr } \mathbf{V}' \mathbf{J} \mathbf{V} \\ &= \text{tr } \mathbf{V}' (\mathbf{Z}' \mathbf{A} \mathbf{Z} + \lambda \mathbf{J}) \mathbf{V} - 2 \text{tr} (\bar{\mathbf{V}}' \mathbf{Z}' \mathbf{A} + \mathbf{B}') \mathbf{Z} \mathbf{V} + \Gamma^{(3)}. \end{aligned}$$

This majorization function has the desired functional form described in (8). Differentiation with respect to \mathbf{V} and equating to zero yields the linear system

$$(\mathbf{Z}' \mathbf{A} \mathbf{Z} + \lambda \mathbf{J}) \mathbf{V} = \mathbf{Z}' \mathbf{A} \bar{\mathbf{Z}} + \mathbf{Z}' \mathbf{B}. \quad (14)$$

The update \mathbf{V}^+ that solves this system can then be calculated efficiently by Gaussian elimination.

4.2 Algorithm Implementation and Complexity

Pseudocode for GenSVM is given in Algorithm 1. As can be seen, the algorithm simply updates all instance weights at each iteration, starting by determining the indicator variable ε_i . In practice, some calculations can be done efficiently for all instances by using matrix algebra. When step doubling (see Appendix B) is applied in the majorization algorithm, line 25 is replaced by $\mathbf{V} \leftarrow 2\mathbf{V}^+ - \bar{\mathbf{V}}$. In the implementation step doubling is applied after a burn-in of 50 iterations. The implementation used in the experiments described in Section 6 is written in C, using the ATLAS (Whaley and Dongarra, 1998) and LAPACK (Anderson et al., 1999) libraries. The source code for this C library is available under the open source GNU GPL license, through an online repository. A thorough description of the implementation is available in the package documentation.

The complexity of a single iteration of the IM algorithm is $O(n(m+1)^2)$ assuming that $n > m > K$. As noted earlier, the convergence rate of the general IM algorithm is linear. Computational complexity of standard SVM solvers that solve the dual problem through decomposition methods lies between $O(n^2)$ and $O(n^3)$ depending on the value of λ (Boftou and Lin, 2007). An efficient algorithm for the method of Crammer and Singer (2002a) developed by Keerthi et al. (2008) has a complexity of $O(n\bar{m}K)$ per iteration, where $\bar{m} \leq m$ is the average number of nonzero features per training instance. In the methods of Lee et al. (2004) and Weston and Watkins (1998), a quadratic programming problem with $n(K-1)$ dual variables needs to be solved, which is typically done using a standard solver. An analysis of the exact convergence of GenSVM, including the expected number of iterations needed to achieve convergence at a factor ϵ , is outside the scope of the current work and a subject for further research.

4.3 Smart Initialization

When training machine learning algorithms to determine the optimal hyperparameters, it is common to use cross validation (CV). With GenSVM it is possible to initialize the matrix $\bar{\mathbf{V}}$ such that the final result of a fold is used as the initial value for \mathbf{V}_0 for the next fold. This same technique can be used when searching for the optimal hyperparameter configuration in a grid search, by initializing the weight matrix with the outcome of the previous configuration. Such warm-start initialization greatly reduces the time needed to perform cross validation with GenSVM. It is important to note here that using warm starts is not easily possible with dual optimization approaches. Therefore, the ability to use warm starts can be seen as an advantage of solving the GenSVM optimization problem in the primal.

Algorithm 1: GenSVM Algorithm

```

Input:  $\mathbf{X}, \mathbf{y}, \rho, \beta, \kappa, \lambda, \epsilon$ 
Output:  $\mathbf{V}$ 
1  $K \leftarrow \max(\mathbf{Y})$ 
2  $t \leftarrow 1$ 
3  $\mathbf{Z} \leftarrow [\mathbf{1} \ \mathbf{X}]$ 
4 Let  $\bar{\mathbf{V}} \leftarrow \mathbf{V}_0$ 
5 Generate  $\mathbf{J}$  and  $\mathbf{U}_K$ 
6  $L_t = L_{\text{MSVM}}(\bar{\mathbf{V}})$ 
7  $L_{t-1} = (1 + 2\epsilon)L_t$ 
8 while  $(L_{t-1} - L_t)/L_t > \epsilon$  do
9   for  $i \leftarrow 1$  to  $n$  do
10    Compute  $\bar{q}_i^{(g,j)} = \mathbf{z}_i^T \bar{\mathbf{V}} \mathbf{g}_{g,j}$  for all  $j \neq g_i$ 
11    Compute  $h(\frac{\bar{q}_i^{(g,j)}}{q_i^{(g,j)}}$ ) for all  $j \neq g_i$  by (3)
12    if  $\epsilon_i = 1$  then
13      Compute  $q_{i,g_i}^{(1)}$  and  $b_{i,g_i}^{(1)}$  for all  $j \neq g_i$  according to Table 4 in Appendix C
14    else
15      Compute  $\omega_i$  following (10)
16      Compute  $q_{i,g_i}^{(p)}$  and  $b_{i,g_i}^{(p)}$  for all  $j \neq g_i$  according to Table 4 in Appendix C
17    end
18    Compute  $\alpha_i$  by (12)
19    Compute  $\beta_i$  by (13)
20  end
21  Construct  $\mathbf{A}$  from  $\alpha_i$ 
22  Construct  $\mathbf{B}$  from  $\beta_i$ 
23  Find  $\mathbf{V}^+$  that solves (14)
24   $\bar{\mathbf{V}} \leftarrow \mathbf{V}^+$ 
25   $\mathbf{V} \leftarrow \mathbf{V}^+$ 
26   $L_{t-1} \leftarrow L_t$ 
27   $L_t \leftarrow L_{\text{MSVM}}(\bar{\mathbf{V}})$ 
28   $t \leftarrow t + 1$ 
29 end

```

5. Nonlinearity

One possible method to include nonlinearity in a classifier is through the use of spline transformations (see for instance Hastie et al., 2009). With spline transformations each attribute vector \mathbf{x}_j is transformed to a spline basis \mathbf{N}_j , for $j = 1, \dots, m$. The transformed input matrix $\mathbf{N} = [\mathbf{N}_1, \dots, \mathbf{N}_m]$ is then of size $n \times l$, where l depends on the degree of the spline transformation and the number of interior knots chosen. An application of spline transformations to the binary SVM can be found in Groenen et al. (2007).

A more common way to include nonlinearity in machine learning methods is through the use of the kernel trick, attributed to Aizerman et al. (1964). With the kernel trick, the dot product of two instance vectors in the dual optimization problem is replaced by the dot product of the same vectors in a high dimensional feature space. Since no dot products appear in the primal formulation of GenSVM, a different method is used here.

By applying a preprocessing step on the kernel matrix, nonlinearity can be included using the same algorithm as the one presented for the linear case. Furthermore, predicting class labels requires a postprocessing step on the obtained matrix \mathbf{V}^* . A full derivation is given in Appendix D.

6. Experiments

To assess the performance of the proposed GenSVM classifier, a simulation study was done comparing GenSVM with seven existing multiclass SVMs on 13 small data sets. These experiments are used to precisely measure predictive accuracy and total training time using performance profiles and rank plots. To verify the feasibility of GenSVM for large data sets an additional simulation study is done. The results of this study are presented separately in Section 6.4. Due to the large number of data sets and methods involved, experiments were only done for the linear kernel. Experiments on nonlinear multiclass SVMs would require even more training time than for linear MSVMs and is considered outside the scope of this paper.

6.1 Setup

Implementations of the heuristic multiclass SVMs (OVO, OvA, and DAG) were included through LibSVM (v. 3.16, Chang and Lin, 2011). LibSVM is a popular library for binary SVMs with packages for many programming languages, it is written in C++ and implements a variation of the SMO algorithm of Platt (1999). The OvO and DAG methods are implemented in this package, and a C implementation of OvA using LibSVM was created for these experiments.³ For the single-machine approaches the MSVMpack package was used (v. 1.3, Laner and Guenieur, 2011), which is written in C. This package implements the methods of Weston and Watkins (W&W, 1998), Crammer and Singer (CKS, 2002a), Lee et al. (LLW, 2004), and Guenieur and Monfimi (MSVM², 2011). Finally, to verify if implementation differences are relevant for algorithm performance the LibLinear (Fan et al., 2008) implementation of the method by Crammer and Singer (2002a) is also included (denoted LL CKS). This implementation uses the optimization algorithm by Keerthi et al. (2008).

To compare the classification methods properly, it is desirable to remove any bias that could occur when using cross validation (Cawley and Talbot, 2010). Therefore, *nested* cross validation is used (Stone, 1974), as illustrated in Figure 5. In nested CV, a data set is randomly split in a number of *chunks*. Each of these chunks is kept apart from the remaining chunks once, while the remaining chunks are combined to form a single data set. A grid search is then applied to this combined data set to find the optimal hyperparameters with which to predict the test chunk. This process is then repeated for each of the chunks. The predictions of the test chunk will be unbiased since it was not included in the grid search. For this reason, it is argued that this approach is preferred over approaches that simply report maximum accuracy rates obtained during the grid search.

3. The LibSVM code used for DAGSVM is the same code as was used in Hsu and Lin (2002) and is available at <http://www.csie.ntu.edu.tw/~cjlin1/libsvmtools>.

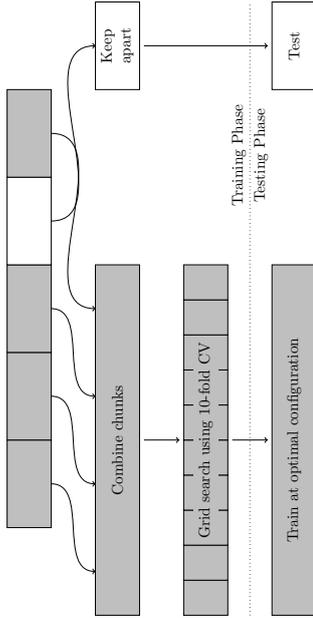


Figure 5: An illustration of nested cross validation. A data set is initially split in five *chunks*. Each chunk is kept apart once, while a grid search using 10-fold CV is applied to the combined data from the remaining 4 chunks. The optimal parameters obtained there are then used to train the model one last time, and predict the chunk that was kept apart.

For the experiments 13 data sets were selected from the UCI repository (Bache and Lichman, 2013). The selected data sets and their relevant statistics are shown in Table 1. All attributes were rescaled to the interval $[-1, 1]$. The **image segmentation** and **vowel1** data sets have a predetermined train and test set, and were therefore not used in the nested CV procedure. Instead, a grid search was done on the provided training set for each classifier, and the provided test set was predicted at the optimal hyperparameters obtained. For the data sets without a predetermined train/test split, nested CV was used with 5 initial chunks. Hence, $5 \cdot 11 + 2 = 57$ pairs of independent train and test data sets are obtained.

While running the grid search, it is desirable to remove any fluctuations that may result in an unfair comparison. Therefore, it was ensured that all methods had the same CV split of the training data for the same hyperparameter configuration (specifically, the value of the regularization parameter). In practice, it can occur that a specific CV split is advantageous for one classifier but not for others (either in time or performance). Thus, ideally the grid search would be repeated a number of times with different CV splits, to remove this variation. However, due to the size of the grid search this is considered to be infeasible. Finally, it should be noted here that during the grid search 10-fold cross validation was applied in a non-stratified manner, that is, without resampling of small classes.

The following settings were used in the numerical experiments. The regularization parameter was varied on a grid with $\lambda \in \{2^{-18}, 2^{-16}, \dots, 2^{18}\}$. For GenSVM the grid search was extended with the parameters $\kappa \in \{-0.9, 0.5, 5.0\}$ and $p \in \{1.0, 1.5, 2.0\}$. The stopping parameter for the GenSVM majorization algorithm was set at $\epsilon = 10^{-6}$ during the grid search in the training phase and at $\epsilon = 10^{-8}$ for the final model in the testing phase. In addition, two different weight specifications were used for GenSVM: the unit weights with $\theta_i = 1, \forall i$, as well as the group-size correction weights introduced in (4). Thus, the grid search consists of 342 configurations for GenSVM, and 19 configurations

Data set	Instances (n)	Features (m)	Classes (K)	min n_k	max n_k
breast tissue	106	9	6	14	22
iris	150	4	3	50	50
wine	178	13	3	48	71
image segmentation*	210/2100	18	7	30	30
glass	214	9	6	9	76
vertebral	310	6	3	60	150
ecoli	336	8	8	2	143
vowel*	528/462	10	11	48	48
balancescale	625	4	3	49	288
vehicle	846	18	4	199	218
contraception	1473	9	3	333	629
yeast	1484	8	10	5	463
car	1728	6	4	65	1210

Table 1: Data set summary statistics. Data sets with an asterisk have a predetermined test data set. For these data sets, the number of training instances is denoted for the train and test data sets respectively. The final two columns denote the size of the smallest and the largest class, respectively.

for the other methods. Since nested CV is used for most data sets, it is required to run 10-fold cross validation on a total of 28158 hyperparameter configurations. To enhance the reproducibility of these experiments, the exact predictions made by each classifier for each configuration were stored in a text file.

To run all computations in a reasonable amount of time, the computations were performed on the Dutch National LIISA Compute Cluster. A master-worker program was developed using the message passing interface in Python (Dalcin et al., 2005). This allows for efficient use of multiple nodes by successively sending out tasks to worker threads from a single master thread. Since the total training time of a classifier is also of interest, it was ensured that all computations were done on the exact same core type.⁴ Furthermore, training time was measured from within the C programs, to ensure that only the time needed for the cross validation routine was measured. The total computation time needed to obtain the presented results was about 152 days, using the LIISA Cluster this was done in five and a half wall-clock time.

During the training phase it showed that several of the single machine methods implemented through MSVMpack did not converge to an optimal solution within reasonable amount of time.⁵ Instead of limiting the maximum number of iterations of the method, MSVMpack was modified to stop after a maximum of 2 hours of training time per configuration. This results in 12 minutes of training time per cross validation fold. The solution found after this amount of training time was used for prediction during cross validation.

4. The specific type of core used is the Intel Xeon E5-2650 v2, with 16 threads at a clock speed of 2.6 GHz. At most 14 threads were used simultaneously, reserving one for the master thread and one for system processes.

5. The default MSVMpack settings were used with a chunk size of 4 for all methods.

Whenever training was stopped prematurely, this was recorded.⁶ Of the 57 training sets, 24 configurations had prematurely stopped training in one or more CV splits for the LLW method, versus 19 for W&W, 9 for MSVM², and 2 for C&S (MSVMpack). For the LibSVM methods, 13 optimal configurations for OvA reached the default maximum number of iterations in one or more CV folds, versus 9 for DAGSVM, and 3 for OvO. No early stopping was needed for GensVM or for LL C&S.

Determining the optimal hyperparameters requires a performance measure on the obtained predictions. For binary classifiers it is common to use either the hitrate or the area under the ROC curve as a measure of classifier performance. The hitrate only measures the percentage of correct predictions of a classifier and has the well known problem that no correction is made for group sizes. For instance, if 90% of the observations of a test set belong to one class, a classifier that always predicts this class has a high hitrate, regardless of its discriminatory power. Therefore, the adjusted Rand index (ARI) is used here as a performance measure (Hubert and Arabie, 1985). The ARI corrects for chance and can therefore more accurately measure discriminatory power of a classifier than the hitrate can. Using the ARI for evaluating supervised learning algorithms has previously been proposed by Santos and Embrechts (2009).

The optimal parameter configurations for each method on each data set were chosen such that the maximum predictive performance was obtained as measured with the ARI. If multiple configurations obtained the highest performance during the grid search, the configuration with the smallest training time was chosen. The results on the training data show that during cross validation GensVM achieved the highest classification accuracy on 41 out of 57 data sets, compared to 15 and 12 for DAG and OvO, respectively. However, these are results on the training data sets and therefore can contain considerable bias. To accurately assess the out-of-sample prediction accuracy the optimal hyperparameter configurations were determined for each of the 57 training sets, and the test sets were predicted with these parameters. To remove any variations due to random starts, building the classifier and predicting the test set was repeated 5 times for each classifier.

Below the simulation results on the small data sets will be evaluated using performance profiles and rank tests. Performance profiles offer a visual representation of classifier performance, while rank tests allow for identification of statistically significant differences between classifiers. For the sake of completeness tables of performance scores and computation times for each method on each data set are provided in Appendix E. To promote reproducibility of the empirical results, all the code used for the classifier comparisons and all the obtained results will be released through an online repository.

6.2 Performance Profiles

One way to get insight in the performance of different classification methods is through *performance profiles* (Dolan and Moré, 2002). A performance profile shows the empirical cumulative distribution function of a classifier on a performance metric.

6. For the classifiers implemented through LibSVM very long training times were only observed for the OvA method, however due to the nature of this method it is not trivial to stop the calculations after a certain amount of time. This behavior was observed in about 1% of all configurations tested on all data sets, and is therefore considered negligible. Also, for the LibSVM methods it was recorded whenever the maximum number of iterations was reached.

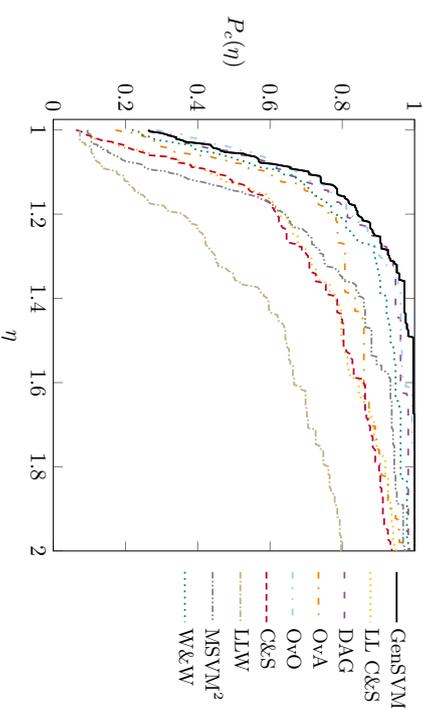


Figure 6: Performance profiles for classification accuracy created from all repetitions of the test set predictions. The methods OvA, C&S, LL C&S, MSVM², W&W, and LLW will always have a smaller probability of being within a factor η of the maximum performance than the GensVM, OvO, or DAG methods.

Let \mathcal{D} denote the set of data sets, and \mathcal{C} denote the set of classifiers. Further, let $p_{d,c}$ denote the performance of classifier $c \in \mathcal{C}$ on data set $d \in \mathcal{D}$ as measured by the ARI. Now define the performance ratio $v_{d,c}$ as the ratio between the best performance on data set d and the performance of classifier c on data set d , that is

$$v_{d,c} = \frac{\max\{p_{d,c} : c \in \mathcal{C}\}}{p_{d,c}}.$$

Thus the performance ratio is 1 for the best performing classifier on a data set and increases for classifiers with a lower performance. Then, the performance profile for classifier c is given by the function

$$P_c(\eta) = \frac{1}{N_D} |\{d \in \mathcal{D} : v_{d,c} \leq \eta\}|,$$

where $N_D = |\mathcal{D}|$ denotes the number of data sets. Thus, the performance profile estimates the probability that classifier c has a performance ratio below η . Note that $P_c(1)$ denotes the empirical probability that a classifier achieves the highest performance on a given data set.

Figure 6 shows the performance profile for classification accuracy. Estimates of $P_c(1)$ from Figure 6 show that there is a 28.42% probability that OvO achieves the optimal performance, versus 26.32% for both GensVM and DAGSVM. Note that this includes cases where each of these methods achieves the best performance. Figure 6 also shows that although there is a small difference in the probabilities of GensVM, OvO, and DAG within

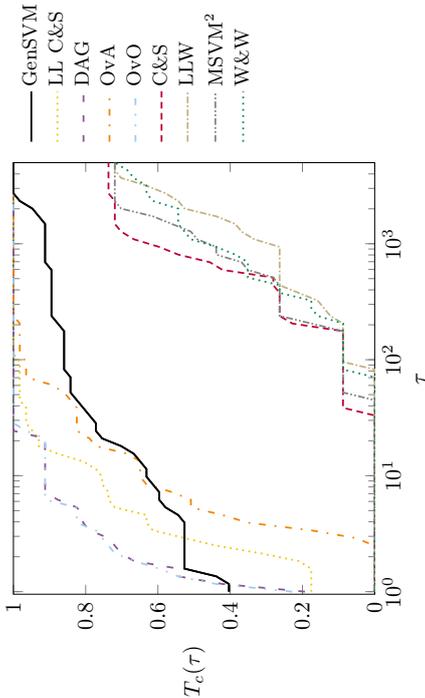


Figure 7: Performance profiles for training time. GenSVM has a priori about 40% chance of requiring the smallest time to perform the grid search on a given method. The methods implemented through MSVMpack always have a lower chance of being within a factor τ of the smallest training time than any of the other methods.

a factor of 1.08 of the best predictive performance, for $\eta \geq 1.08$ GenSVM almost always has the highest probability. It can also be concluded that since the performance profiles of the MSVMpack implementation and the LibLinear implementation of the method of Crammer and Singer (2002a) nearly always overlap, implementation differences have a negligible effect on the classification performance of this method. Finally, the figure shows that OVA and the methods of Lee et al. (2004), Crammer and Singer (2002a), Weston and Watkins (1998), and Guermeur and Monfrini (2011) always have a smaller probability of being within a given factor of the optimal performance than GenSVM, OVO, or DAG do.

Similarly, a performance profile can be constructed for the training time necessary to do the grid search. Let $t_{d,c}$ denote the total training time for classifier c on data set d . Next, define the performance ratio for time as

$$w_{d,c} = \frac{t_{d,c}}{\min\{t_{d,c} : c \in \mathcal{C}\}}.$$

Note that here the classifier with the smallest training time has preference. Therefore, comparison of classifier computation time is done with the lowest computation time achieved on a given data set d . Again, the ratio is 1 when the lowest training time is reached, and it increases for higher computation time. Hence, the performance profile for time is defined

$$T_c(\tau) = \frac{1}{N_D} |\{d \in \mathcal{D} : w_{d,c} \leq \tau\}|.$$

The performance profile for time estimates the probability that a classifier c has a time ratio below τ . Again, $T_c(1)$ denotes the fraction of data sets where classifier c achieved the smallest training time among all classifiers.

Figure 7 shows the performance profile for the time needed to do the grid search. Since large differences in training time were observed, a logarithmic scale is used for the horizontal axis. This performance profile clearly shows that all MSVMpack methods suffer from long computation times. The fastest methods are GenSVM, OVO, and DAG, followed by the LibLinear implementation of C&S. From the value of $T_c(1)$ it is found that GenSVM has the highest probability of being the fastest method for the total grid search, with a probability of 40.35%, versus 22.81% for OVO, 19.30% for DAG, and 17.54% for LibLinear C&S. The other methods never achieve the smallest grid search time. It is important to note here that the grid search for GenSVM is 18 times larger than that of the other methods. These results illustrate the incredible advantage GenSVM has over other methods by using warm starts in the grid search.

In addition to the performance profile, the average computation time per hyperparameter configuration was also examined. Here, GenSVM has an average training time of 0.97 seconds per configuration, versus 20.56 seconds for LibLinear C&S, 24.84 seconds for OVO, and 25.03 seconds for DAGSVM. This is a considerable difference, which can be explained again by the use of warm starts in GenSVM (see Section 4.3). When the total computation time per data set is averaged, it is found that GenSVM takes on average 331 seconds per data set, LibLinear C&S 391 seconds, OVO 472 seconds, and DAG 476 seconds. The difference between DAGSVM and OVO can be attributed to the prediction strategy used by DAGSVM. Thus it can be concluded that on average GenSVM is the fastest method during the grid search, despite the fact it has 18 times more hyperparameters to consider than the other methods.

6.3 Rank Tests

Following suggestions from Demšar (2006), ranks are used to investigate significant differences between classifiers. The benefit of using ranks instead of actual performance metrics is that ranks have meaning when averaged across different data sets, whereas average performance metrics do not. Ranks are calculated for the performance as measured by the ARI, the total training time needed to do the grid search, and the average time per hyperparameter configuration. When ties occur fractional ranks are used.

Figure 8 shows the average ranks for both classification performance and total and average training time for all classifiers. From Figure 8a it can be seen that GenSVM is in second place in terms of overall classification performance measured by the ARI. Only OVO has higher performance than GenSVM *on average*. Similarly, Figure 8b shows the average ranks for the total training time. Here, GenSVM is on average the fourth fastest method for the complete grid search. When looking at the rank plot for the average training time per hyperparameter configuration, it is clear that the warm starts used during training in GenSVM are very useful as it ranks as the fastest method on this metric, as shown in Figure 8c.

As Demšar (2006) suggests, the Friedman rank test can be used to find significant differences between classifiers (Friedman, 1937, 1940). If τ_{ed} denotes the fractional rank of

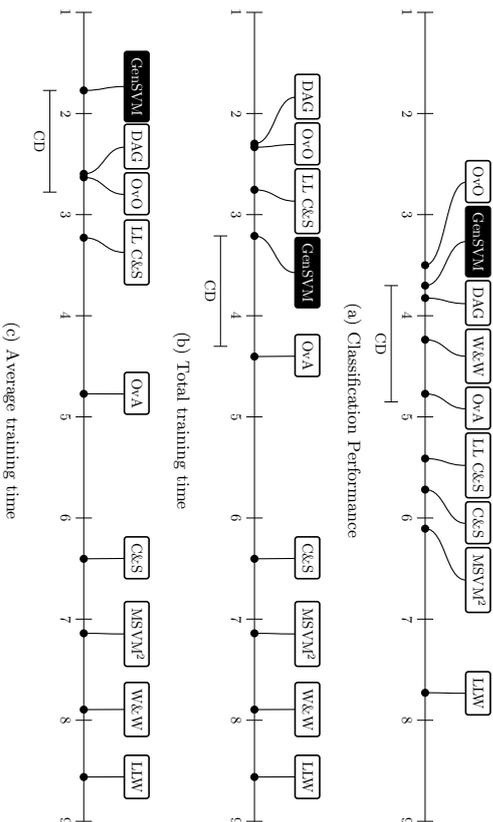


Figure 8. Figure (a) shows the average ranks for performance, (b) shows the average ranks for the total computation time needed for the grid search, and (c) shows the ranks for the average time per hyperparameter configuration. It can be seen that GenSVM obtains the second overall rank in predictive performance, fourth overall rank in total training time, and first overall rank in average training time. In all figures, CD shows the critical difference of Holm’s procedure. Classifiers beyond this CD differ significantly from GenSVM at the 5% significance level.

classifier c on data set d , then with N_C classifiers and N_D data sets the Friedman statistic is given by

$$\chi_F^2 = \frac{12N_D}{N_C(N_C + 1)} \left[\sum_c R_c^2 - \frac{N_C(N_C + 1)^2}{4} \right]. \quad (15)$$

Here, $R_c = 1/N_D \sum_d r_{cd}$ denotes the average rank of classifier c . This test statistic is distributed following the χ^2 distribution with $N_C - 1$ degrees of freedom. As Demsar (2006) notes, Iman and Davenport (1980) showed that the Friedman statistic is undesirably conservative and the F -statistic is to be used instead, which is given by

$$F_F = \frac{(N_D - 1)\chi_F^2}{N_D(N_C - 1) - \chi_F^2},$$

and is distributed following the F -distribution with $N_C - 1$ and $(N_C - 1)(N_D - 1)$ degrees of freedom. Under the null hypothesis of either test there is no significant difference in the performance of any of the algorithms.

When performing the Friedman test, it is found that with the ranks for classifier performance $\chi_F^2 = 116.3$ ($p < 10^{-16}$), and $F_F = 19.2$ ($p = 10^{-16}$). Hence, with both tests the

null hypothesis of equal classification accuracy can be rejected. Similarly, for training time the test statistics are $\chi_F^2 = 384.8$ ($p < 10^{-16}$) and $F_F = 302.4$ ($p \approx 10^{-16}$). Therefore, the null hypothesis of equal training time can also be rejected. When significant differences are found through the Friedman test, Demsar (2006) suggests to use Holm’s step-down procedure as a post-hoc test, to find which classifiers differ significantly from a chosen reference classifier (Holm, 1979). Here, GenSVM is used as a reference classifier, since comparing GenSVM with existing methods is the main focus of these experiments.

Holm’s procedure is based on testing whether the z -statistic comparing classifier i with classifier j is significant, while adjusting for the familywise error rate. Following Demsar (2006), this z -statistic is given by

$$z = (R_{\text{GenSVM}} - R_i) \sqrt{\frac{6N_D}{N_C(N_C + 1)}}, \quad (16)$$

where R_{GenSVM} is the average rank of GenSVM and R_i the average rank of another classifier, for $i = 1, \dots, N_C - 1$. Subsequently, the p -values computed from this statistic are sorted in increasing order, as $p_1 < p_2 < \dots < p_{N_C - 1}$. Then, the null hypothesis of equal classification accuracy can be rejected if $p_i < \alpha / (N_C - i)$. If for some i the null hypothesis cannot be rejected, all subsequent tests will also fail. By inverting this procedure, a critical difference (CD) can be computed that indicates the minimal difference between the reference classifier and the next best classifier.⁷ These critical differences are also illustrated in the rank plots in Figure 8.

Using Holm’s procedure, it is found that for predictive performance GenSVM significantly outperforms the method of Lee et al. (2004) ($p < 10^{-14}$), the method of Gernemur and Montfrini (2011) ($p = 10^{-6}$), the MSVMpack implementation of Grammer and Singer (2002a) ($p = 4 \cdot 10^{-5}$), and the Liblinear implementation of the same method ($p = 0.0004$) at the 5% significance level. Note that since this last method is included twice these test results are conservative. In terms of total training time, GenSVM is significantly faster than all methods implemented through MSVMpack (C&S, W&W, MSVM², and LLW) and OVA at the 5% significance level. Recall that the hyperparameter grid for GenSVM is 18 times larger than that of the other methods. When looking at average training time per hyperparameter configuration, GenSVM is significantly faster than all methods except OVO and DAG, at the 1% significance level.

6.4 Large Data sets

The above results focus on the predictive performance of GenSVM as compared to other multiclass SVM methods. To assess the practicality of GenSVM for large data sets additional simulations were done on three more data sets. The covtype data set ($n = 581016$, $m = 54$, $K = 7$) and the kddcup99 data set ($n = 494021$, $m = 116$, $K = 23$) were selected from the UCI repository (Bache and Lichman, 2013).⁸ Additionally, the *cars* data set ($n = 100968$, $m = 338$, $K = 8$) was retrieved from the Keel repository (Alcalá et al.,

7. This is done by taking the smallest value of $\alpha / (N_C - i)$ for which the null hypothesis is rejected, looking up the corresponding z -statistic, and inverting (16).

8. For kddcup99 the 10% training data set and the corrected test data set are used here, both available through the UCI repository.

Package	Method	Covtype	Fars	KDDCup-99
GenSVM	GenSVM	0.3571**	0.8102***	0.9758
LibLinear	L1R-L2L	0.3372	0.8080	0.9762
LibLinear	L2R-L1L (D)	0.3405	0.7995	0.9789
LibLinear	L2R-L2L	0.3383	0.8090**	0.9781
LibLinear	L2R-L2L (D)	0.3393	0.8085*	0.9744
LibLinear	C&S	0.3582***	0.8081	0.9758
LibSVM	DAG	0.8056	0.7872	0.9809***
LibSVM	OvA	0.8055	0.9804**	0.9800*
LibSVM	OvO	0.3432*	0.7996	0.9741
MSVMpack	C&S	0.3117	0.7846	0.9660
MSVMpack	LLW	0.3165	0.6567	0.9658
MSVMpack	MSVM ²	0.2848	0.7719	0.6446
MSVMpack	W&W			

Table 2: Overview of predictive performance on large data sets, as measured by the ARI. Asterisks are used to mark the three best performing methods for each data set, with three stars denoting the best performing method.

2010). For large data sets the LibLinear package (Fan et al., 2008) is often used, so the SVM methods from this package were added to the list of alternative methods.⁹

LibLinear includes five different SVM implementations: a coordinate descent algorithm for the ℓ_2 -regularized ℓ_1 -loss and ℓ_2 -loss dual problems (Hsieh et al., 2008), a coordinate descent algorithm for the ℓ_1 -regularized ℓ_2 -loss SVM (Yuan et al., 2010; Fan et al., 2008), a Newton method for the primal ℓ_2 -regularized ℓ_2 -loss SVM problem (Lin et al., 2008), and finally a sequential dual method for the multiclass SVM by Crammer and Singer (2002a) introduced by Keerthi et al. (2008). This last method was again included to facilitate a comparison between the implementations of LibLinear and MSVMpack. Note that with the exception of this last method all methods in LibLinear are binary SVMs that implement the one-vs-all strategy.

With the different variants of the linear multiclass SVMs included in LibLinear, a total of 13 methods were considered for these large data sets. Since training of the hyperparameters for each method leads to a high computational burden the nested CV procedure was replaced by a grid search using ten-fold CV on a training set of 80% of the data, followed by out-of-sample prediction on the remaining 20% using the final model. The `kddcup99` data set comes with a separate test data set of 292302 instances, so this was used for the out-of-sample predictions. The grid search on the training set used the same hyperparameter configurations as for the small data sets above, with 342 configurations for GenSVM and 19

9. Yet another interesting SVM approach to multiclass classification is the Pegasos method by Shalev-Shwartz et al. (2011). However, the LibLinear package includes five different approaches to SVM, including a fast solver for the method by Crammer and Singer (2002a), which makes it more convenient to include in the list of methods. Moreover, according to the LibLinear documentation (Fan et al., 2008): “LibLinear is competitive or even faster than state of the art linear classifiers such as Pegasos (Shalev-Shwartz et al., 2011) and SVM^{part} (Joachims, 2006)”.

Package	Method	Covtype		Fars		KDDCup-99	
		Total	Mean	Total	Mean	Total	Mean
GenSVM	GenSVM	166949	488	131174	384	1768303	5170
LibLinear	L1R-L2L	69469	3656	4199	221	34517	1817
LibLinear	L2R-L1L (D)	134908	7100	6995	368	16347	860
LibLinear	L2R-L2L	4168	219	746	39	3084	162
LibLinear	L2R-L2L (D)	159781	8410	7897	416	16974	893
LibLinear	C&S	166719	8775	124764	6567	5425	286
LibSVM	DAG	80410	40205	81557	8156	61111	3595
LibSVM	OvA	77335	77335	54965	18322	73871	12312
LibSVM	OvO	140826	46942	84580	8458	81023	4501
MSVMpack	C&S	350397	18442	351664	18509	365733	19249
MSVMpack	LLW	370790	19515	380943	20050	361329	19017
MSVMpack	MSVM2	370736	19512	346140	18218	353479	18604
MSVMpack	W&W	367245	19329	344880	18152	367685	19352

Table 3: Overview of training time for each of the large data sets. The average training time per hyperparameter configuration is also shown. All values are reported in seconds. For LibSVM the full grid search could never be completed, and results are averaged only over the finished configurations.

configurations for the other methods. The only difference was that for GenSVM $\epsilon = 10^{-9}$ was used when training the final model. To accelerate the GenSVM computations, support for sparse matrices was added.

Due to the large data set sizes, many methods had trouble converging within a reasonable amount of time. Therefore, total computation time was limited to five hours per hyperparameter configuration per method, both during CV and when training the final model. Where possible this limitation was included in the main optimization routine of each method, such that training was stopped when convergence was reached or when more than five hours had passed. Additionally, for all methods the CV procedure was stopped prematurely if more than five hours had passed after completion of a fold. In this case, cross validation performance is only measured for the folds that were completed. These computations were again performed on the Dutch National LISA Compute Cluster.

Table 2 shows the out-of-sample predictive performance of the different MSVMs on the large data sets. It can be seen that GenSVM is the best performing method on the `fars` data set and the second best method on the `covtype` data set, just after LL C&S. The LibSVM methods outperform the other methods on the `kddcup99` data set, with DAGSVM having the highest performance. No results are available for LibSVM for the `covtype` data set because convergence could not be reached within the five hour time limit during the test phase.

Results on the computation time are reported in Table 3. The ℓ_2 -regularized ℓ_2 -loss method by Lin et al. (2008) is clearly the fastest method. However, for the `covtype` data set GenSVM total training time is competitive with some of the other LibLinear methods, and outperforms these methods in terms of average training time. For the `fars` data set the

average training time of GenSVM is also competitive with some of the LibLinear methods, most notably the method by Crammer and Singer (2002a). The MSVMpack methods seem to be infeasible for such large data sets, as computations were stopped by the five hour time limit for almost all hyperparameter configurations. Early stopping was also needed for the LibLinear implementation of C&S on the `covtype` and `fars` data sets, and for the LibSVM methods on all data sets. For GenSVM, early stopping was only needed for the `kddcup99` data set, which explains the high total computation time there. Especially on these large data sets the advantage of using warm starts in GenSVM is visible: training time was less than 30 seconds in 30% of hyperparameters on `fars`, 23% on `covtype`, and 11% on `kddcup99`.

7. Discussion

A generalized multiclass support vector machine has been introduced, called GenSVM. The method is general in the sense that it subsumes three multiclass SVMs proposed in the literature and it is flexible due to several different weighting options. The simplex encoding of the multiclass classification problem used in GenSVM is intuitive and has an elegant geometrical interpretation. An iterative majorization algorithm has been derived to minimize the convex GenSVM loss function in the primal. This primal optimization approach has computational advantages due to the possibility to use warm starts, and because it can be easily understood. The ability to use warm starts contributes to small training time during cross validation in a grid search, and allows GenSVM to perform competitively on large data sets.

Rigorous computational tests of linear multiclass SVMs on small data sets show that GenSVM significantly outperforms three existing multiclass SVMs (four implementations) on predictive performance at the 5% significance level. On this metric, GenSVM is the second-best performing method overall and the best method among single-machine multiclass SVMs, although the difference with the method of Weston and Watkins (1998) could not be shown to be statistically significant. GenSVM outperforms five other methods on total training time and has the smallest total training time when averaged over all data sets, despite the fact that its grid of hyperparameters is 18 times larger than that of other methods. Due to the possibility of warm starts it also has the smallest average training time per hyperparameter and significantly outperforms all but two alternative methods in this regard at the 1% significance level. For the large data sets, it was found that GenSVM still achieves high classification accuracy and that total training time remains manageable due to the warm starts. In practice, the number of hyperparameters could be reduced if smaller training time is desired. Since GenSVM outperforms existing methods on a number of data sets and achieves fast training time it is a worthwhile addition to the collection of methods available to the practitioner.

In the comparison tests MSVMpack (Lauer and Guerneur, 2011) was used to access four single machine multiclass SVMs proposed in the literature. A big advantage of using this library is that it allows for a single straightforward C implementation, which greatly reduces the programming effort needed for the comparisons. However, as is noted in the MSVMpack documentation, slight differences exist between MSVMpack and method-specific implementations. For instance, on small data sets MSVMpack can be slower, due to working set

selection and shrinking procedures in other implementations. However, classification performance is comparable between MSVMpack and method-specific implementations, as was verified by adding the LibLinear implementation of the method of Crammer and Singer (2002a) to the list of alternative methods. Thus, we argue that the results for predictive accuracy presented above are accurate regardless of implementation, but small differences can exist for training time when other implementations for single machine MSVMs are used.

Another interesting conclusion that can be drawn from the experimental results is that the one-vs-all method never performs as good as one-vs-one, DAGSVM, or GenSVM. In fact, the profile plot in Figure 6 shows that OvA always has a smaller probability of obtaining the best classification performance as either of these three methods. These results are also reflected in the classification accuracy of the LibLinear methods on the large data set. In the literature, the paper by Rifkin and Klantau (2004) is often cited as evidence that OvA performs well (see for instance Keerthi et al., 2008). However, the simulation results in this paper suggest that OvA is in fact inferior to OvO, DAG, and GenSVM.

This paper was focused on linear multiclass SVMs. An obvious extension is to incorporate nonlinear multiclass SVMs through kernels. Due to the large number of data sets and the long training time the numerical experiments were limited to linear multiclass SVM. Nonlinear classification through kernels can be achieved by linear methods through a preprocessing step of an eigendecomposition on the kernel matrix, which is a process of the order $O(n^3)$. In this case, GenSVM will benefit from precomputing kernels before starting the grid search, or using a larger stopping criterion in the IM algorithm by increasing ϵ in Algorithm 1. In addition, approximations can be done by using rank approximated kernel matrices, such as the Nystrom method proposed by Williams and Seeger (2001). Such enhancements are considered topics for further research.

Finally, the potential of using GenSVM in an online setting is recognized. Since the solution can be found quickly when a warm-start is used, GenSVM may be useful in situations where new instances have to be predicted at a certain moment, and the true class label arrives later. Then, re-estimating the GenSVM solution can be done as soon as the true class label of an object arrives, and a previously known solution can be used as a warm start. It is expected that in this scenario only a few iterations of the IM algorithm are needed to arrive at a new optimal solution. This, too, is considered a subject for further research.

Acknowledgments

The computational experiments of this work were performed on the Dutch National LISA Compute Cluster, and supported by the Dutch National Science Foundation (NWO). The authors thank SURFsara (www.surfsara.nl) for the support in using the LISA cluster.

Appendix A. Simplex Coordinates

The simplex used in the formulation of the GenSVM loss function is a regular K -simplex in \mathbb{R}^{K-1} with distance 1 between each pair of vertices, which is centered at the origin. Since these requirements alone do not uniquely define the simplex coordinates in general, it will

be chosen such that at least one of the vertices lies on an axis. The 2-simplex in \mathbb{R}^1 is uniquely defined with the coordinates $-\frac{1}{2}$ and $+\frac{1}{2}$. Using these requirements, it is possible to define a recursive formula for \mathbf{U}_K , the simplex coordinate matrix of the K -simplex in \mathbb{R}^{K-1} as

$$\mathbf{U}_K = \begin{bmatrix} \mathbf{U}_{K-1} & \mathbf{1}t \\ \mathbf{0}' & s \end{bmatrix}, \quad \text{with} \quad \mathbf{U}_2 = \begin{bmatrix} -\frac{1}{2} \\ \frac{1}{2} \end{bmatrix}.$$

Note that the matrix \mathbf{U}_K has K rows and $K-1$ columns. Since the simplex is centered at zero it holds that the elements in each column sum to 0, implying that $s = -(K-1)t$. Denote by \mathbf{u}_i' the i -th row of \mathbf{U}_K and by $\tilde{\mathbf{u}}_i'$ the i -th row of \mathbf{U}_{K-1} , then it follows from the edge length requirement that

$$\|\mathbf{u}_i' - \mathbf{u}_K'\|^2 = \|\tilde{\mathbf{u}}_i' - \mathbf{0}' + t - s\|^2 = \|\tilde{\mathbf{u}}_i'\|^2 + (t-s)^2 = 1, \quad \forall i \neq K.$$

From the requirement of equal distance from each vertex to the origin it follows that

$$\begin{aligned} \|\mathbf{u}_i'\|^2 &= \|\mathbf{u}_K'\|^2, \\ \|\tilde{\mathbf{u}}_i'\|^2 + t^2 &= s^2, \quad \forall i \neq K. \end{aligned}$$

Combining these two expressions yields the equation $2s^2 - 2st - 1 = 0$. Substituting $s = -(K-1)t$ and choosing $s > 0$ and $t < 0$ gives

$$t = \frac{-1}{\sqrt{2K(K-1)}}, \quad s = \frac{K-1}{\sqrt{2K(K-1)}}.$$

Note that using $K=2$ in these expressions gives $t = -\frac{1}{2}$ and $s = \frac{1}{2}$, as expected. The recursive relationship defined above then reveals that the first $K-1$ elements in column $K-1$ of the matrix are equal to t , and the K -th element in column $K-1$ is equal to s . This can then be generalized for an element u_{kl} in row k and column l of \mathbf{U}_K , yielding the expression given in (1).

Appendix B. Details of Iterative Majorization

In this section a brief introduction to iterative majorization is given, following the description of Voss and Eckhardt (1980). The section concludes with a note on step doubling, a common technique to speed up quadratic majorization algorithms.

Given a continuous function $f: \mathcal{X} \rightarrow \mathbb{R}$ with $\mathcal{X} \subseteq \mathbb{R}^d$, construct a *majorization function* $g(x, \bar{x})$ such that

$$\begin{aligned} f(\bar{x}) &= g(\bar{x}, \bar{x}), \\ f(x) &\leq g(x, \bar{x}) \quad \text{for all } x \in \mathcal{X}, \end{aligned}$$

with $\bar{x} \in \mathcal{X}$ a so-called *supporting point*. In general, the majorization function is constructed such that its minimum can easily be found, for instance by choosing it to be quadratic in x . If $f(x)$ is differentiable at the supporting point, the above conditions imply $\nabla f(\bar{x}) = \nabla g(\bar{x}, \bar{x})$. The following procedure can now be used to find a stationary point of $f(x)$,

1. Let $\bar{x} = x_0$, with x_0 a random starting point.

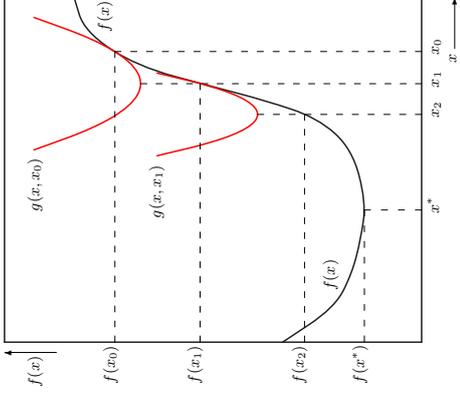


Figure 9: One-dimensional graphical illustration of the iterative majorization algorithm, adapted from De Leeuw (1988). The minimum of a majorization function $g(x, x_r)$ provides the supporting point for the next majorization function $g(x, x_{r+1})$. The sequence of supporting points $\{x_r\}$ converges towards the stationary point x^* if $f(x)$ is bounded from below, as is the case here.

2. Minimize $g(x, \bar{x})$ with respect to x , such that $x^+ = \arg \min g(x, \bar{x})$.
3. If $f(\bar{x}) - f(x^+) < \epsilon f(x^+)$ stop, otherwise let $\bar{x} = x^+$ and go to step 2.

In this algorithm ϵ is a small constant. Note that $f(x)$ must be bounded from below on \mathcal{X} for the algorithm to converge. In fact, the following *sandwich inequality* can be derived (De Leeuw, 1993)

$$f(x^+) \leq g(x^+, \bar{x}) \leq g(\bar{x}, \bar{x}) = f(\bar{x}).$$

This inequality shows that if $f(x)$ is bounded from below the iterative majorization algorithm achieves global convergence to a stationary point of the function (Voss and Eckhardt, 1980). The iterative majorization algorithm is illustrated in Figure 9, where the majorization functions are shown as a quadratic function. As can be seen from the illustration, the sequence of supporting points $\{x_r\}$ converges to the stationary point x^* of the function $f(x)$. In practical situations, this convergence is to a local minimum of $f(x)$.

For quadratic majorization the number of iterations can often be reduced by using a technique known as *step doubling* (De Leeuw and Heiser, 1980). Step doubling reduces the number of iterations by using $\bar{x} = x_{r+1} = 2x^+ - x_r$ as the next supporting point in Step 3 of the algorithm, instead of $\bar{x} = x_{r+1} = x^+$. Intuitively, step doubling can be understood as stepping over the minimum of the majorization function to the point lying directly

“opposite” the supporting point \bar{x} (see also Figure 9). Note that the guaranteed descent of the IM algorithm still holds when using step doubling, since $f(2x^+ - \bar{x}) \leq g(2x^+ - \bar{x}; \bar{x}) = g(\bar{x}; \bar{x}) = f(\bar{x})$. In practice, step doubling reduces the number of iterations by half. A caveat of using step doubling is that the distance to the stationary point can be increased if the initial point is far from this point. Therefore, in practical applications, a burn-in should be used before step doubling is applied.

Appendix C. Huber Hinge Majorization

In this appendix, the majorization function will be derived of the Huber hinge error raised to the power p . Thus, a quadratic function $g(x, \bar{x}) = ax^2 - 2bx + c$ is required, which is a majorization function of

$$f(x) = h^p(x) = \begin{cases} (1-x-\frac{\kappa+1}{2})^p & \text{if } x \leq -\kappa \\ \frac{1}{(2(\kappa+1))^p}(1-x)^{2p} & \text{if } x \in (-\kappa, 1] \\ 0 & \text{if } x > 1, \end{cases}$$

with $p \in [1, 2]$. Each piece of $f(x)$ provides a possible region for the supporting point \bar{x} . These regions will be treated separately, starting with $\bar{x} \in (-\kappa, 1]$.

Since the majorization function must touch $f(x)$ at the supporting point, we can solve $f(\bar{x}) = g(\bar{x}; \bar{x})$ and $f'(\bar{x}) = g'(\bar{x}; \bar{x})$ for b and c to find

$$b = a\bar{x} + \frac{p}{1-\bar{x}} \left(\frac{1-\bar{x}}{\sqrt{2(\kappa+1)}} \right)^{2p}, \quad (17)$$

$$c = a\bar{x}^2 + \left(1 + \frac{2p\bar{x}}{1-\bar{x}} \right) \left(\frac{1-\bar{x}}{\sqrt{2(\kappa+1)}} \right)^{2p}, \quad (18)$$

whenever $\bar{x} \in (-\kappa, 1]$. Note that since $p \in [1, 2]$ the function $f(x)$ can become proportional to a fourth power on the interval $x \in (-\kappa, 1]$. The upper bound of the second derivative of $f(x)$ on this interval is reached at $x = -\kappa$. Equating $f''(-\kappa)$ to $g''(-\kappa; \bar{x}) = 2a$ and solving for a yields

$$a = \frac{1}{4}p(2p-1) \left(\frac{\kappa+1}{2} \right)^{p-2}. \quad (19)$$

Figure 10a shows an illustration of the majorization function when $\bar{x} \in (-\kappa, 1]$.

For the interval $\bar{x} \leq -\kappa$ the following expressions are found for b and c using similar reasoning as above

$$b = a\bar{x} + \frac{1}{2}p \left(1 - \bar{x} - \frac{\kappa+1}{2} \right)^{p-1}, \quad (20)$$

$$c = a\bar{x}^2 + p\bar{x} \left(1 - \bar{x} - \frac{\kappa+1}{2} \right)^{p-1} + \left(1 - \bar{x} - \frac{\kappa+1}{2} \right)^p. \quad (21)$$

To obtain the largest possible majorization step it is desired that the minimum of the majorization function is located at $x \geq 1$, such that $g(x_{min}, \bar{x}) = 0$. This requirement yields

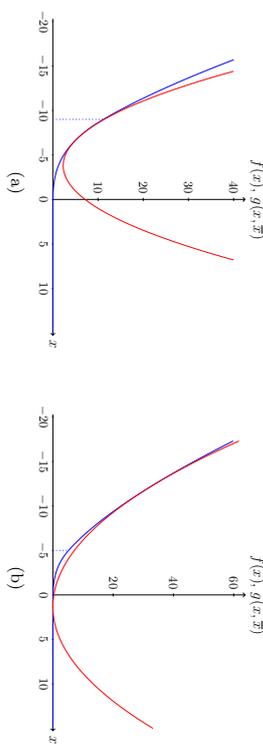


Figure 10: Graphical illustration of the majorization of the function $f(x) = h^p(x)$. Figure (a) shows the case where $\bar{x} \in (-\kappa, 1]$, whereas (b) shows the case where $\bar{x} \leq (p + \kappa - 1)/(p - 2)$. In both cases $p = 1.5$. It can be seen that in (b) the minimum of the majorization function lies at $x > 1$, such that the largest possible majorization step is obtained.

$c = b^2/a$, which gives

$$a = \frac{1}{4}p^2 \left(1 - \bar{x} - \frac{\kappa+1}{2} \right)^{p-2}. \quad (22)$$

Note however that due to the requirement that $f(x) \leq g(x; \bar{x})$ for all $x \in \mathbb{R}$, this majorization is not valid for all values of \bar{x} . Solving the requirement for the minimum of the majorization function, $g(x_{min}, \bar{x}) = 0$ for \bar{x} yields

$$\bar{x} \leq \frac{p + \kappa - 1}{p - 2}.$$

Thus, if \bar{x} satisfies this condition, (22) can be used for a , whereas for cases where $\bar{x} \in ((p + \kappa - 1)/(p - 2), -\kappa]$, the value of a given in (19) can be used. Figure 10b shows an illustration of the case where $\bar{x} \leq (p + \kappa - 1)/(p - 2)$.

Next, a majorization function for the interval $\bar{x} > 1$ is needed. Since it has been derived that for the interval $\bar{x} \leq (p + \kappa - 1)/(p - 2)$ the minimum of the majorization function lies at $x \geq 1$, symmetry arguments can be used to derive the majorization function for $\bar{x} > 1$, and ensure that it is also tangent at $x = (p\bar{x} + \kappa - 1)/(p - 2)$. This yields the coefficients

$$a = \frac{1}{4}p^2 \left(\frac{p}{p-2} \left(1 - \bar{x} - \frac{\kappa+1}{2} \right) \right)^{p-2}, \quad (23)$$

$$b = a \left(\frac{p\bar{x} + \kappa - 1}{p - 2} \right) + \frac{1}{2}p \left(\frac{p}{p-2} \left(1 - \bar{x} - \frac{\kappa+1}{2} \right) \right)^{p-1}, \quad (24)$$

$$c = a \left(\frac{p\bar{x} + \kappa - 1}{p - 2} \right)^2 + p \left(\frac{p\bar{x} + \kappa - 1}{p - 2} \right) \left(\frac{p}{p-2} \left(1 - \bar{x} - \frac{\kappa+1}{2} \right) \right)^{p-1} + \left(\frac{p}{p-2} \left(1 - \bar{x} - \frac{\kappa+1}{2} \right) \right)^p. \quad (25)$$

Region	a	b	c
$\bar{x} \leq \frac{p+\kappa-1}{p-2}$	(22)	(20)	(21)
$\bar{x} \in \left(\frac{p+\kappa-1}{p-2}, -\kappa \right]$	(19)	(20)	(21)
$\bar{x} \in (-\kappa, 1]$	(19)	(17)	(18)
$\bar{x} > 1, p \neq 2$	(23)	(24)	(25)
$\bar{x} > 1, p = 2$	(19)	$a\bar{x}$	$a\bar{x}^2$

Table 4: Overview of quadratic majorization coefficients for different pieces of $h^p(x)$, depending on \bar{x} .

Finally, observe that some of the above coefficients are invalid if $p = 2$. However, since the upper bound on the interval $\bar{x} \in (-\kappa, 1]$ given in (19) is still valid if $p = 2$, it is possible to do a separate derivation with this value for a to find for $\bar{x} > 1$, $b = a\bar{x}$ and $c = a\bar{x}^2$. For the other regions the previously derived coefficients still hold. Table 4 gives an overview of the various coefficients depending on the location of \bar{x} .

Appendix D. Kernels in GenSVM

To include kernels in GenSVM a preprocessing step is needed on the kernel matrix, and a preprocessing step is needed on the obtained parameters before doing class prediction. Let $k : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}^+$ denote a positive definite kernel satisfying Mercer's theorem, and let \mathcal{H}_k denote the corresponding reproducing kernel Hilbert space. Furthermore, define a feature mapping $\phi : \mathbb{R}^m \rightarrow \mathcal{H}_k$ as $\phi(\mathbf{x}) = k(\mathbf{x}, \cdot)$, such that by the reproducing property of k it holds that $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{H}_k}$.

Using this, the kernel matrix \mathbf{K} is defined as the $n \times n$ matrix with elements $k(\mathbf{x}_i, \mathbf{x}_j)$ on the i -th row and j -th column. Thus, if Φ denotes the $n \times l$ matrix with rows $\phi(\mathbf{x}_i)$ for $i = 1, \dots, n$ and $l \in [1, \infty]$, then $\mathbf{K} = \Phi\Phi'$. Note that it depends on the chosen kernel whether Φ is finite dimensional. However, the rank of Φ can still be determined through \mathbf{K} , since $r = \text{rank}(\Phi) = \text{rank}(\mathbf{K}) \leq \min(n, l)$.

Now, let the reduced singular value decomposition of Φ be given by

$$\Phi = \mathbf{P}\Sigma\mathbf{Q}',$$

where \mathbf{P} is $n \times r$, Σ is $r \times r$, and \mathbf{Q} is $l \times r$. Note that here, $\mathbf{P}'\mathbf{P} = \mathbf{I}_r$, $\mathbf{Q}'\mathbf{Q} = \mathbf{I}_r$, and Σ is diagonal. Under the mapping $\mathbf{X} \rightarrow \Phi$ it follows that the simplex space vectors become

$$\begin{aligned} \mathbf{S} &= \Phi\mathbf{W} + \mathbf{1}\mathbf{t}' \\ &= \mathbf{P}\Sigma\mathbf{Q}'\mathbf{W} + \mathbf{1}\mathbf{t}' \\ &= \mathbf{M}\mathbf{Q}'\mathbf{W} + \mathbf{1}\mathbf{t}'. \end{aligned}$$

Here \mathbf{W} is $l \times (K-1)$ to correspond to the dimensions of Φ , and the $n \times r$ matrix $\mathbf{M} = \mathbf{P}\Sigma$ has been introduced. In general \mathbf{W} cannot be determined, since l might be infinite. This problem can be solved as follows. Decompose \mathbf{W} in two parts, $\mathbf{W} = \mathbf{W}_1 + \mathbf{W}_2$, where \mathbf{W}_1 is in the linear space of \mathbf{Q} and \mathbf{W}_2 is orthogonal to that space, thus

$$\begin{aligned} \mathbf{W}_1 &= \mathbf{Q}\mathbf{Q}'\mathbf{W}, \\ \mathbf{W}_2 &= (\mathbf{I}_l - \mathbf{Q}\mathbf{Q}')\mathbf{W}. \end{aligned}$$

Then it follows that

$$\begin{aligned} \mathbf{S} &= \mathbf{M}\mathbf{Q}'\mathbf{W} + \mathbf{1}\mathbf{t}' \\ &= \mathbf{M}\mathbf{Q}'(\mathbf{W}_1 + \mathbf{W}_2) + \mathbf{1}\mathbf{t}' \\ &= \mathbf{M}\mathbf{Q}'\mathbf{W}_1 + (\mathbf{I}_l - \mathbf{Q}\mathbf{Q}')\mathbf{W} + \mathbf{1}\mathbf{t}' \\ &= \mathbf{M}\mathbf{Q}'\mathbf{W}_1 + \mathbf{M}(\mathbf{Q}' - \mathbf{Q}'\mathbf{Q}\mathbf{Q}')\mathbf{W} + \mathbf{1}\mathbf{t}' \\ &= \mathbf{M}\mathbf{Q}'\mathbf{W}_1 + \mathbf{M}(\mathbf{Q}' - \mathbf{Q}')\mathbf{W} + \mathbf{1}\mathbf{t}' \\ &= \mathbf{M}\mathbf{Q}'\mathbf{W}_1 + \mathbf{1}\mathbf{t}', \end{aligned}$$

where it has been used that $\mathbf{Q}'\mathbf{Q} = \mathbf{I}_r$. If the penalty term of the GenSVM loss function is considered, it is found that

$$P_\lambda(\mathbf{W}) = \lambda \text{tr } \mathbf{W}'\mathbf{W} = \lambda \text{tr } \mathbf{W}'_1\mathbf{W}_1 + \lambda \text{tr } \mathbf{W}'_2\mathbf{W}_2,$$

since

$$\begin{aligned} \mathbf{W}'_1\mathbf{W}_2 &= \mathbf{W}'\mathbf{Q}\mathbf{Q}'(\mathbf{I}_l - \mathbf{Q}\mathbf{Q}')\mathbf{W} \\ &= \mathbf{W}'\mathbf{Q}\mathbf{Q}'\mathbf{W} - \mathbf{W}'\mathbf{Q}\mathbf{Q}'\mathbf{W} \\ &= \mathbf{O}. \end{aligned}$$

Here again it has been used that $\mathbf{Q}'\mathbf{Q} = \mathbf{I}_r$, and \mathbf{O} is defined as a $(K-1) \times (K-1)$ dimensional matrix of zeroes. Note that the penalty term depends on \mathbf{W}_2 whereas the simplex vectors \mathbf{S} do not. Therefore, at the optimal solution it is required that \mathbf{W}_2 is zero, to minimize the loss function.

Since \mathbf{W}_1 is still $l \times (K-1)$ dimensional with l possibly infinite, consider the substitution $\mathbf{W}_1 = \mathbf{Q}\Omega$, with Ω an $r \times (K-1)$ matrix. The penalty term in terms of Ω then becomes

$$P_\lambda(\mathbf{W}_1) = \lambda \text{tr } \mathbf{W}'_1\mathbf{W}_1 = \lambda \text{tr } \Omega'\mathbf{Q}'\mathbf{Q}\Omega = \lambda \text{tr } \Omega'\Omega = P_\lambda(\Omega).$$

Note also that

$$\begin{aligned} \mathbf{S} &= \mathbf{M}\mathbf{Q}'\mathbf{W}_1 + \mathbf{1}\mathbf{t}' \\ &= \mathbf{M}\mathbf{Q}'\mathbf{Q}\Omega + \mathbf{1}\mathbf{t}' \\ &= \mathbf{M}\Omega + \mathbf{1}\mathbf{t}'. \end{aligned}$$

The question remains on how to determine the matrices \mathbf{P} and Σ , given that the matrix Φ cannot be determined explicitly. These matrices can be determined by the eigendecomposition of \mathbf{K} , where $\mathbf{K} = \mathbf{P}\Sigma^2\mathbf{P}'$. In the case where $r < n$, Σ^2 contains only the first r

eigenvalues of \mathbf{K} , and \mathbf{P} the corresponding r columns. Hence, if \mathbf{K} is not of full rank, a dimensionality reduction is achieved in Ω . The complexity of finding the eigendecomposition of the kernel matrix is $O(n^3)$.

Since the distances $q_i^{(k)}$ in the GenSVM loss function can be written as $q_i^{(k)} = \mathbf{s}_i' \delta_{k_j}$ it follows that the errors can again be calculated in this formulation. Finally, to predict the simplex space vectors of a test set \mathbf{X}_2 the following is used. Let Φ_2 denote the feature space mapping of \mathbf{X}_2 , then

$$\begin{aligned} \mathbf{S}_2 &= \Phi_2 \mathbf{W}_1 + \mathbf{1}t' \\ &= \Phi_2 \mathbf{Q} \mathbf{Q} + \mathbf{1}t' \\ &= \Phi_2 \mathbf{Q} \mathbf{Q} \mathbf{P}' \mathbf{P} \Sigma^{-1} \Omega + \mathbf{1}t' \\ &= \Phi_2 \Phi' \mathbf{P}' \Sigma^{-1} \Omega + \mathbf{1}t' \\ &= \mathbf{K}_2 \mathbf{P} \Sigma^{-1} \Omega + \mathbf{1}t' \\ &= \mathbf{K}_2 \mathbf{M} \Sigma^{-2} \Omega + \mathbf{1}t', \end{aligned}$$

where $\mathbf{K}_2 = \Phi_2 \Phi'$ is the kernel matrix between the test set and the training set, and it was used that $\Sigma \mathbf{P}' \mathbf{P} \Sigma^{-1} = \mathbf{I}_r$, and $\Phi' = \mathbf{Q} \Sigma \mathbf{P}'$ by definition.

With the above expressions for \mathbf{S} and $F_\lambda(\Omega)$, it is possible to derive the majorization function of the loss function for the nonlinear case. The first order conditions can then again be determined, which yields the following system

$$\begin{pmatrix} \mathbf{1}' \\ \mathbf{M}' \end{pmatrix} \mathbf{A} \begin{bmatrix} \mathbf{1} & \mathbf{M} \\ \mathbf{0} & \mathbf{I}_r \end{bmatrix} \begin{bmatrix} t \\ \Omega \end{bmatrix} = \begin{bmatrix} \mathbf{1}' \\ \mathbf{M}' \end{pmatrix} \mathbf{A} \begin{bmatrix} \mathbf{1} & \mathbf{M} \\ \Omega & \end{bmatrix} + \begin{bmatrix} \mathbf{1}' \\ \mathbf{M}' \end{bmatrix} \mathbf{B}. \quad (26)$$

This system is analogous to the system solved in linear GenSVM. In fact, it can be shown that by writing $\mathbf{Z} = [\mathbf{1} \ \mathbf{M}]$ and $\mathbf{V} = [t' \ \Omega]'$, this system is equivalent to (14). This property is very useful for the implementation of GenSVM, since nonlinearity can be included by simply adding a preprocessing and postprocessing step to the existing GenSVM algorithm.

Appendix E. Additional Simulation Results

Tables 5 and 6 respectively show the predictive accuracy rates and ARI scores on each data set averaged over each of the 5 test folds. For readability all scores are rounded to four decimal digits, however identifying the classifier with the highest score was done on the full precision scores. As can be seen, the choice of performance metric has an effect on which classification method has the highest classification performance. Regardless of the performance metric the tables show that MSVM² and W&W never achieve the maximum classification performance on a data set. Note that conclusions drawn from tables of performance scores are quite limited and the results presented in Section 6 provide more insight into the performance of the various classifiers.

Table 7 shows the computation time averaged over the five nested CV folds for each data set and each method. In the grid search GenSVM considered 342 hyperparameter configurations versus 19 configurations for the other methods. Despite this difference GenSVM outperformed the other methods on five data sets, DAG outperformed other methods on four data sets, OvO on two, and LibLinear C&S was fastest on the remaining two data

sets. To illustrate the effect of the larger grid search in GenSVM on the computation time, Table 8 shows the average computation time per hyperparameter configuration. This table shows that GenSVM is faster than other methods on nine out of thirteen data sets, which illustrates the influence of warm starts in the GenSVM grid search.

Data set	GenSVM	LI	C&S	DAG	OvA	OvO	C&S	LMW	MSVM ²	W&W
balance-scale	0.9168	0.8883	0.9168	0.8928	0.9168	0.8922	0.8701	0.8714	0.9008	
breast-tissue	0.7113	0.7005	0.6515	0.7455	0.6515	0.6944	0.5391	0.6663	0.5711	
car	0.8279	0.6185	0.8449	0.8131	0.8524	0.6489	0.7898	0.7855	0.8273	
contraception	0.5022	0.4773	0.5017	0.4739	0.5010	0.4699	0.4751	0.4964	0.4972	
ecoli	0.8630	0.8547	0.8629	0.8510	0.8659	0.8576	0.7450	0.8456	0.8098	
glass	0.6448	0.5813	0.6542	0.5746	0.6450	0.6342	0.4504	0.5988	0.6215	
image-seg	0.9169	0.9103	0.9162	0.9210	0.9219	0.9088	0.7741	0.8157	0.8962	
iris	0.9600	0.8893	0.9533	0.9467	0.9533	0.8813	0.7640	0.8320	0.9253	
vehicle	0.8016	0.7978	0.7955	0.7872	0.7990	0.7941	0.6870	0.7550	0.9933	
verbal	0.8323	0.8458	0.8355	0.8484	0.8419	0.8439	0.7890	0.8432	0.8426	
vowel	0.4762	0.4242	0.4957	0.3398	0.5065	0.4221	0.2273	0.3277	0.5017	
wine	0.9776	0.9841	0.9608	0.9775	0.9775	0.9775	0.9843	0.9775	0.9717	
yeast	0.5343	0.5841	0.5748	0.5175	0.5802	0.5818	0.4217	0.5500	0.5811	

Table 5: Predictive accuracy rates for each of the classification methods on all data sets. All numbers are out-of-sample prediction accuracies averaged over the 5 independent test folds. Maximum scores per data set are determined on the full precision scores and are underlined.

Data set	GenSVM	LI	C&S	DAG	OvA	OvO	C&S	LMW	MSVM ²	W&W
balance-scale	0.8042	0.7355	0.8042	0.7238	0.8042	0.7466	0.6634	0.6653	0.7698	
breast-tissue	0.5222	0.4964	0.4591	0.5723	0.4787	0.4755	0.4043	0.5585	0.4655	
car	0.5381	0.3290	0.5345	0.5238	0.5491	0.3131	0.4874	0.4808	0.5337	
contraception	0.0762	0.0532	0.0757	0.0535	0.0747	0.0525	0.0393	0.0688	0.0699	
ecoli	0.7668	0.7606	0.7755	0.7652	0.7776	0.7578	0.6236	0.7499	0.7385	
glass	0.2853	0.2478	0.2970	0.2346	0.2910	0.2792	0.1776	0.2494	0.2861	
image-seg	0.8318	0.8221	0.8280	0.8402	0.8390	0.8193	0.6810	0.6810	0.7991	
iris	0.8783	0.7057	0.8609	0.8384	0.8609	0.6879	0.5549	0.6057	0.7918	
vehicle	0.6162	0.6009	0.6057	0.5979	0.6057	0.5925	0.4933	0.5397	0.6121	
verbal	0.6649	0.6797	0.6606	0.6862	0.6778	0.6742	0.6480	0.6859	0.6836	
vowel	0.2472	0.2474	0.2895	0.1624	0.3218	0.2257	0.1425	0.2043	0.2767	
wine	0.9320	0.9585	0.8848	0.9378	0.9200	0.9362	0.9498	0.9352	0.9250	
yeast	0.2519	0.2501	0.2415	0.2419	0.2477	0.2534	0.1301	0.2235	0.2501	

Table 6: Predictive ARI scores for each of the classification methods on all data sets. All numbers are out-of-sample ARI scores averaged over the 5 independent test folds. Maximum scores per data set are determined on the full precision scores and are underlined.

References

- A. Aizerman, E.M. Braverman, and L.I. Rozoner. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25: 821–837, 1964.
- J. Alcalá, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera. Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, 17(2-3):255–287, 2010.
- E.L. Allwein, R.E. Schapire, and Y. Singer. Reducing multiclass to binary: a unifying approach for margin classifiers. *The Journal of Machine Learning Research*, 1:113–141, 2001.
- E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' guide*. SIAM, third edition, 1999.
- B. Ávila Pires, C. Szepesvari, and M. Ghavamzadeh. Cost-sensitive multiclass classification risk bounds. In *Proceedings of The 30th International Conference on Machine Learning*, pages 1391–1399, 2013.
- K. Bache and M. Lichman. UCI machine learning repository, 2013. URL <http://archive.ics.uci.edu/ml>.
- C.C.J.H. Bijleveld and J. De Leeuw. Fitting longitudinal reduced-rank regression models by alternating least squares. *Psychometrika*, 56(3):433–447, 1991.
- L. Bottou and C.-J. Lin. Support vector machine solvers. In L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, editors, *Large Scale Kernel Machines*, pages 301–320. MIT Press, Cambridge, MA., 2007.
- E.J. Breidensteiner and K.P. Bennett. Multicategory classification by support vector machines. *Computational Optimization and Applications*, 12(1):53–79, 1999.
- G.C. Cawley and N.L.C. Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *The Journal of Machine Learning Research*, 11:2079–2107, 2010.
- C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, 2011.
- O. Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19(5): 1155–1178, 2007.
- C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
- K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2:265–292, 2002a.

Data set	GenSVM	LL C&S	DAG	OvA	OvO	C&S	LLW	MSVM ²	W&W
balancscale	44.3	88.0	86.4	155.8	84.9	34549	73671	79092	35663
breasttissue	136.0	52.9	3.8	65.2	3.8	28782	74188	38625	81961
car	<u>251.2</u>	1239.1	1513.0	4165.2	1517.4	47408	95197	46978	85050
contraception	82.5	1128.5	1948.3	5079.1	1913.4	45163	88844	43402	40335
ecoli	603.0	88.9	34.7	183.2	34.8	28907	95989	39590	131571
glass	254.6	110.8	48.6	198.2	47.7	27938	89073	37194	108499
imageseq	558.2	67.1	2.4	151	3.2	32691	73300	48576	97218
iris	55.7	13.8	1.9	32.9	1.5	12822	47196	38060	77409
vehicle	<u>186.4</u>	376.8	307.9	1373.3	309.9	37605	49988	40665	43511
vertebral	23.5	66.6	24.4	63.1	24.3	24716	70798	36168	23888
vowel	1282.4	463.9	<u>83.7</u>	3900.2	86.1	36270	95036	49924	82990
wine	129.6	0.1	0.2	0.2	0.2	12854	70439	18389	41018
yeast	1643.3	<u>1181.7</u>	1434.6	4251.1	1423.6	44112	103240	56603	86802

Table 7: Computation time in seconds for each of the methods on all data sets. Values are averaged over the five nested CV splits. Minimum values per data set are underlined. Note that the size of the grid search is 18 times larger in GenSVM than in other methods.

Data set	GenSVM	LL C&S	DAG	OvA	OvO	C&S	LLW	MSVM ²	W&W
balancscale	<u>0.130</u>	4.632	4.546	8.199	4.468	1818	3877	4163	1877
breasttissue	0.398	2.785	0.201	3.434	0.201	1515	3905	2033	4314
car	<u>0.734</u>	63.217	79.629	219.221	79.863	2495	5010	2473	4476
contraception	0.241	59.396	102.544	267.319	100.704	2377	4676	2284	2123
ecoli	1.763	4.680	1.828	9.643	1.881	1521	5052	2084	6925
glass	<u>0.744</u>	5.832	2.559	10.432	2.511	1470	4688	1958	5710
imageseq	1.632	3.530	0.128	7.947	0.167	1721	3858	2557	5117
iris	0.163	0.725	0.101	1.729	0.081	675	2484	2003	4074
vehicle	<u>0.545</u>	19.830	16.206	72.282	16.308	1979	2631	2140	2290
vertebral	<u>0.069</u>	3.504	1.286	3.32	1.279	1301	3726	1904	1257
vowel	<u>3.750</u>	24.415	4.406	205.274	4.533	1909	5002	2628	4368
wine	0.379	<u>0.003</u>	0.010	0.011	0.010	677	3707	968	2159
yeast	4.805	62.197	75.506	223.74	74.925	2322	5434	2979	4569

Table 8: Average computation time in seconds per hyperparameter configuration for each of the methods on all data sets. Values are averaged over the five nested CV splits. Minimum values per data set are determined on the full precision values and are underlined.

- K. Crammer and Y. Singer. On the learnability and design of output codes for multiclass problems. *Machine Learning*, 47(2-3):201–233, 2002b.
- L. Dalcin, R. Paz, and M. Storti. MPI for Python. *Journal of Parallel and Distributed Computing*, 65(9):1108–1115, 2005.
- J. De Leeuw. Applications of convex analysis to multidimensional scaling. In J.R. Barra, F. Brodeau, G. Romier, and B. Van Cutsem, editors, *Recent Developments in Statistics*, pages 133–146. North Holland Publishing Company, Amsterdam, 1977.
- J. De Leeuw. Convergence of the majorization method for multidimensional scaling. *Journal of Classification*, 5(2):163–180, 1988.
- J. De Leeuw. Fitting distances by least squares. Technical Report 130, Los Angeles: Interdivisional Program in Statistics, UCLA, 1993.
- J. De Leeuw. Block-relaxation algorithms in statistics. In H.-H. Bock, W. Lenski, and M.M. Richter, editors, *Information Systems and Data Analysis*, pages 308–324. Springer Berlin Heidelberg, 1994.
- J. De Leeuw and W.J. Heiser. Multidimensional scaling with restrictions on the configuration. *Mathiomatic Analysis*, 5:501–522, 1980.
- J. Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.
- T.G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.
- U. Dogan, T. Glasmachers, and C. Igel. Fast training of multi-class support vector machines. Technical Report 03/2011, University of Copenhagen, Faculty of Science, 2011.
- E.D. Dolan and J.J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002.
- R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.
- M. Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701, 1937.
- M. Friedman. A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1):86–92, 1940.
- P.J.F. Groenen and W.J. Heiser. The tunneling method for global optimization in multidimensional scaling. *Psychometrika*, 61(3):529–550, 1996.
- P.J.F. Groenen, W.J. Heiser, and J.J. Meinman. Global optimization in least-squares multidimensional scaling by distance smoothing. *Journal of Classification*, 16(2):225–254, 1999.
- P.J.F. Groenen, G. Nalbantov, and J.C. Bioch. Nonlinear support vector machines through iterative majorization and L_{∞} -splines. In R. Decker and H.-J. Lenz, editors, *Advances in Data Analysis*, pages 149–161. Springer Berlin Heidelberg, 2007.
- P.J.F. Groenen, G. Nalbantov, and J.C. Bioch. SVM-Maj: a majorization approach to linear support vector machines with different hinge errors. *Advances in Data Analysis and Classification*, 2(1):17–43, 2008.
- Y. Guerneur and E. Montfili. A quadratic loss multi-class SVM for which a radius-margin bound applies. *Informatica*, 22(1):73–96, 2011.
- G.H. Hardy, J.E. Littlewood, and G. Pólya. *Inequalities*. Cambridge University Press, 1934.
- T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, New York, 2nd edition, 2009.
- T. F. Havel. An evaluation of computational strategies for use in the determination of protein structure from distance constraints obtained by nuclear magnetic resonance. *Progress in Biophysics and Molecular Biology*, 56(1):43–78, 1991.
- S.I. Hill and A. Doucet. A framework for kernel-based multi-category classification. *Journal of Artificial Intelligence Research*, 30:525–564, 2007.
- S. Hohn. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.
- C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S.S. Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear SVM. In *Proceedings of the 25th International Conference on Machine Learning*, pages 408–415, 2008.
- C.-W. Hsu and C.-J. Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002.
- P.J. Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.
- L. Hubert and P. Arable. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.
- D.R. Hunter and K. Lange. A tutorial on MM algorithms. *The American Statistician*, 58(1):30–37, 2004.
- R.L. Inman and J.M. Davenport. Approximations of the critical region of the Friedman statistic. *Communications in Statistics – Theory and Methods*, 9(6):571–595, 1980.
- T. Joachims. Training linear SVMs in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 217–226, 2006.
- S.S. Keerthi, S. Sundararajan, K.-W. Chang, C.-J. Hsieh, and C.-J. Lin. A sequential dual method for large scale multi-class linear SVMs. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 408–416, 2008.

- U.H.G. Krefel. Pairwise classification and support vector machines. In B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods*, pages 255–268. MIT Press, 1999.
- F. Lauer and Y. Guermur. MSVMPack: A multi-class support vector machine package. *The Journal of Machine Learning Research*, 12:2269–2272, 2011.
- Y. Lee, Y. Lin, and G. Wahba. Multicategory support vector machines: theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, 99(465):67–81, 2004.
- C.-J. Lin, R.C. Weng, and S.S. Keerthi. Trust region Newton method for logistic regression. *The Journal of Machine Learning Research*, 9:627–650, 2008.
- Y. Mroueh, T. Poggio, L. Rosasco, and J. Slotine. Multiclass learning with simplex coding. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2789–2797. Curran Associates, Inc., 2012.
- J.M. Ortega and W.C. Rheinboldt. *Iterative Solutions of Nonlinear Equations in Several Variables*. New York: Academic Press, 1970.
- J.C. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods*, pages 185–208. MIT press, 1999.
- J.C. Platt, N. Cristianini, and J. Shawe-Taylor. Large margin DAGs for multiclass classification. In S.A. Solla, T.K. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 547–553. MIT Press, 2000.
- R. Rifkin and A. Klautau. In defense of one-vs-all classification. *The Journal of Machine Learning Research*, 5:101–141, 2004.
- R.T. Rockafellar. *Convex Analysis*. Princeton University Press, 1997.
- S. Rosset and J. Zhu. Piecewise linear regularized solution paths. *The Annals of Statistics*, 35(3):1012–1030, 2007.
- J.M. Santos and M. Embrechts. On the use of the adjusted Rand index as a metric for evaluating supervised classification. In *Proceedings of the 19th International Conference on Artificial Neural Networks: Part II*, pages 175–184. Springer-Verlag, 2009.
- S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter. Pegasos: Primal Estimated sub-Gradient Solver for SVM. *Mathematical Programming*, 127(1):3–30, 2011.
- A. Statnikov, C.F. Aliferis, D.P. Hardin, and I. Guyon. *A Gentle Introduction to Support Vector Machines in Biomedicine: Theory and Methods*. World Scientific, 2011.
- M. Stone. Cross-validated choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):111–147, 1974.
- H. Tsutsu and Y. Morikawa. An l_p norm minimization using auxiliary function for compressed sensing. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 1, 2012.
- V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.
- H. Voss and U. Eckhardt. Linear convergence of generalized Weiszfeld’s method. *Computing*, 25(3):243–251, 1980.
- E. Weiszfeld. Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tohoku Mathematical Journal*, 43:355–386, 1937.
- J. Weston and C. Watkins. Multi-class support vector machines. Technical Report CSD-TR-98-04, University of London, Royal Holloway, Department of Computer Science, 1998.
- R.C. Whaley and J.J. Dongarra. Automatically tuned linear algebra software. In *Proceedings of the 1998 ACM/IEEE conference on Supercomputing*, pages 1–27. IEEE Computer Society, 1998.
- C.K.I. Williams and M. Seeger. Using the Nystrom method to speed up kernel machines. In T.K. Leen, T.G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001.
- G.-X. Yuan, K.-W. Chang, C.-J. Hsieh, and C.-J. Lin. A comparison of optimization methods and software for large-scale L1-regularized linear classification. *The Journal of Machine Learning Research*, 11:3183–3234, 2010.

Scalable Approximate Bayesian Inference for Outlier Detection under Informative Sampling

Terrance D. Savitsky

SAVITSKY.TERRANCE@BLS.GOV

U. S. Bureau of Labor Statistics

Office of Survey Methods Research

Washington, DC 20212, USA

Editor: Kevin Murphy

Abstract

Government surveys of business establishments receive a large volume of submissions where a small subset contain errors. Analysts need a fast-computing algorithm to flag this subset due to a short time window between collection and reporting. We offer a computationally-scalable optimization method based on non-parametric mixtures of hierarchical Dirichlet processes that allows discovery of multiple industry-indexed local partitions linked to a set of global cluster centers. Outliers are nominated as those clusters containing few observations. We extend an existing approach with a new “merge” step that reduces sensitivity to hyperparameter settings. Survey data are typically acquired under an informative sampling design where the probability of inclusion depends on the surveyed response such that the distribution for the observed sample is different from the population. We extend the derivation of a penalized objective function to use a pseudo-posterior that incorporates sampling weights that “undo” the informative design. We provide a simulation study to demonstrate that our approach produces unbiased estimation for the outlying cluster under informative sampling. The method is applied for outlier nomination for the Current Employment Statistics survey conducted by the Bureau of Labor Statistics.

©2016 Terrance D. Savitsky.

Key words: survey sampling, hierarchical dirichlet process, clustering, bayesian hierarchical models, optimization

1. Introduction

1.1 Outlier Detection and Informative Sampling

The U.S. Bureau of Labor Statistics (BLS) administers the Current Employment Statistics (CES) survey to over 350000 non-farm, public and private business establishments across the U.S. on a monthly basis, receiving approximately 270000 submitted responses in each month. Estimated total employment is published for local, state and national geographies in the U.S., as well as for domains defined by establishment size and industry categories within each geography. The BLS conducts a quality check to discover and correct establishment submission errors, particularly among those establishments whose entries are influential in the overall published domain-level estimates. Of the 270000 submissions, approximately 100000 – 150000 of those include employment changes from the prior to the current submission month. The CES maintains a short lag time of approximately 7 days between receipt of establishment submissions at the end of a month and subsequent publication of employment estimates for that month, such that investigations of submission data quality must be done quickly. The relatively large number of submissions with non-zero changes in employment levels, coupled with the rapid publication schedule, require use of quick-executing, automated data analysis tools that output a relatively small, outlying set of the submissions for further investigation and correction by BLS analysts.

The CES survey utilizes a stratified sampling design with strata constructed by combinations of state, broad industry grouping, and employment size (divided into 8 categories). Business establishments are sampled by their unique unemployment insurance (UI) tax identification numbers, which may contain a cluster of multiple individual sites. If a business establishment is selected for inclusion at the UI level, all of the associated sites in that cluster are also included. Stratum-indexed inclusion probabilities are set to be proportional to average employment size for member establishments of that stratum, which is

done because larger establishments compose a higher percentage of the published employment statistics. The correlation between establishment employment levels and inclusion probabilities induces informativeness into the sampling design, meaning that the probabilities of inclusion are correlated with the response variable of interest. The distribution for the resulting observed sample will be different from that for the population (because the sample emphasizes relatively larger establishments), such that inference made (e.g. about outliers) with the former distribution will be biased for the latter.

1.2 Methodologies for Outlier Detection

The Dirichlet process (DP) (Blackwell and MacQueen 1973) and more generally, species sampling formulations (Ishwaran and James 2003), induce a prior over partitions due to their almost surely discrete construction. Convolving a discrete distribution under a DP prior with a continuous likelihood in a mixture formulation is increasingly used for outlier detection (Quintana and Iglesias 2003), where one supposes that each observation is realized from one of multiple generation processes (Shotwell and Slate 2011). Each cluster collects an ellipse cloud of observations that are centered on the cluster mean.

In this article, we improve and extend sequential, penalized partition (clustering) algorithms of Kulis and Jordan (2011); Broderick et al. (2012), that each approximately estimate a maximum a posteriori (MAP) partition and associated cluster centers or means, to account for data acquired under an informative sampling design under which the distribution for the observed sample is different from that for the population (Bonny et al. 2012). We incorporate first order sampling weights into our model for partitions that “undo” the sampling design so that our application nominates outliers with respect to the population generating (mixture) distribution. The algorithm of Kulis and Jordan (2011) is of a class of approaches (see also Wang and Dunson 2011; Shotwell and Slate 2011) that each produce

an approximation for the MAP from a mixture of DPs posterior distribution (where the (unknown) mixing measure over the parameters that define the data likelihood is drawn from a DP).

We follow Kulis and Jordan (2011) and straightforwardly extend our formulation from a DP prior imposed on the mixing distribution to a hierarchical Dirichlet process (HDP) prior (Teh et al. 2006), which allows our estimation of a distinct “local” partition (set of clusters) for each subset of establishments in the CES binned to one of $J = 24$ industry groupings. The local clusters of the industry-indexed partitions may share cluster centers (means) from a global clustering partition. The hierarchical construction of local clusters that draw from or share global cluster means permits the estimation of a dependence structure among the local partitions and encourages a parsimonious discovery of global clusters as compared to running separate global clustering models on each industry group. We refer to this model as hierarchical clustering.

We apply our (sampling-weighted) global and hierarchical clustering models, derived from the MAP approximations, for the nomination of outliers based on identification of the subset of clusters that together collect relatively few observations (with “few” defined heuristically). Quintana and Iglesias (2003) implement a mixture of DPs, for the purpose of outlier detection, where a partition is sampled at each step of an MCMC under an algorithm that minimizes a loss function. The loss function penalizes the size of the partition (or total number of clusters discovered), which will tend to assign most observations into relatively few clusters, while only a small number of observations will lie outside these clusters, a useful feature for outlier detection. We extend small variance asymptotic result of Broderick et al. (2012) for our global and hierarchical clustering formulations to produce objective functions that penalize the number of estimated clusters. We further devise a new “merge” step among all pairs of discovered clusters in each iteration of our estimation algorithms.

Both the penalized likelihood and the merge step encourage parsimony in the number of clusters discovered.

We next derive our sampling-weighted global and hierarchical clustering algorithms in Section 2, followed by a simulation study that demonstrates the importance of correcting estimation for an informative sampling design for outlier detection in Section 3. The simulation study also demonstrates the efficacy of outlier detection performance for the sampling-weighted hierarchical formulation estimated on synthetic data generated in a nested fashion that is similar to the structure of CES survey data. We apply the sampling-weighted hierarchical algorithm to discover local, by-industry partitions and to nominate observations for outliers for our CES survey application in Section 4. We conclude the paper with a discussion in Section 5.

2. Estimation of Partitions from Asymptotic Non-parametric Mixtures

We begin by specifying a Dirichlet process mixtures of Gaussians model to which we apply small variance asymptotics to derive a penalized K – means optimization expression and an associated fast-computing estimation algorithm. We include the sampling weights in our mixture formulation that asymptotically corrects for an informative sampling design (Savitsky and Toth 2015). We continue and extend this MAP approximation approach to a hierarchical Dirichlet process mixtures of Gaussians model to achieve a hard, point estimate sampling-weighted hierarchical clustering algorithm that permits specification of a set of local, industry indexed, clusters linked to a common set of global clusters.

We implement both algorithms we present, below, in the `growClusters` package for **R** (R Core Team 2014), which is written in C++ for fast computation and available from the authors on request.

2.1 Asymptotic Sampling-weighted Mixtures of Dirichlet Processes

We specify a probability model for a mixture of K_{\max} allowed components to introduce Bayesian estimation of partitions of observations into clusters and the associated cluster centers. The mixtures of DPs will be achieved in the limit of this probability model as K_{\max} increases to infinity.

$$\alpha x_1^1 \mathbf{x}_i | s_i, \mathbf{M} = (\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_{K_{\max}})', \sigma^2, \tilde{w}_i \stackrel{\text{iid}}{\sim} \mathcal{N}_d(\boldsymbol{\mu}_{s_i}, \sigma^2 \mathbb{I}_d)^{w_i} \quad (1a)$$

$$s_i | \boldsymbol{\tau} \stackrel{\text{iid}}{\sim} \mathcal{M}(1, \tau_1, \dots, \tau_{K_{\max}}) \quad (1b)$$

$$\boldsymbol{\mu}_p | G_0 \stackrel{\text{iid}}{\sim} G_0 := \mathcal{N}_d(\mathbf{0}, \rho^2 \mathbb{I}_d) \quad (1c)$$

$$\tau_1, \dots, \tau_{K_{\max}} \sim \mathcal{D}(\alpha/K_{\max}, \dots, \alpha/K_{\max}), \quad (1d)$$

for $i = 1, \dots, n$ sample establishments and $p = 1, \dots, K_{\max}$ allowable clusters. Each cluster candidate, p , indexes a unique value for the associated cluster center, $\boldsymbol{\mu}_p$. The set of cluster assignments of observations together compose a partition of observations, where partitions are indexed by $\mathbf{s} = (s_1, \dots, s_n)$, for $s_i \in \{1, \dots, K\}$. The prior distributions for the cluster centers and assignments induce a random distribution prior, G , for drawing a mean value for each \mathbf{x}_i for $i = 1, \dots, n$, by sampling unique values in the support of G from G_0 and assigning the mean value (that we interpret as a cluster) for each observation by drawing from the set of unique values with probabilities, $\boldsymbol{\tau}$, which are, in turn, randomly drawn from a Dirichlet distribution. As $K_{\max} \uparrow \infty$ in Equation 1d, such that number of possible clusters are countably infinite, this construction converges to a (non-parametric) mixture of Dirichlet processes (Neal 2000). We also note that α influences the number of clusters discovered. For any given K_{\max} : larger values for α will produce draws for $\boldsymbol{\tau}$ that assign larger probabilities to more of the $p \in \{1, \dots, K_{\max}\}$ clusters, such that it is highly influential in the determination of the number of discovered clusters, $K < K_{\max}$.

The likelihood contribution for each observation, $i \in \{1, \dots, n\}$, in Equation 1a is uplifted by a sampling weight, $\tilde{w}_i = n \times w_i / \sum_{i=1}^n w_i$ for $w_i \propto 1/\pi_i$, constructed to be inversely proportional to the known inclusion probability in the sample. This formulation for the sampling weight assigns importance for that observation likelihood based on the amount of information in the finite population represented by that observation. The weights sum to the sample size, n , so that an observation with a low inclusion probability in the sample will have a weight greater than 1, meaning that it “represents” more than a single unit in a re-balanced sample. The weighting serves to “undo” the sampling design by re-balancing the information in the observed sample to approximate that in the population. The sum of the weights directly impacts the estimation of posterior uncertainty such that the summation to n expresses the asymptotic amount of information in the observed sample. This construction is known as a “pseudo” likelihood because the distribution for the population is approximated by weighting each observation back to the population. Savitsky and Toth (2015) provide three conditions that define a class of sampling designs under which the pseudo posterior (defined from the pseudo likelihood) is guaranteed to contract in L_1 on the unknown true generating distribution, P_0 . The first condition states that the inclusion probability, π_i , for each unit in the finite population, $i \in \{1, \dots, N\}$, must be strictly greater than 0. This condition ensures that no portion of the population may be systematically excluded from the sample. The second condition on the sampling design requires the second order inclusion dependence for any two units, $i, j \in \{1, \dots, N\}$, be globally bounded from above by a constant; for example, a two-stage design in which units within selected clusters are sampled without replacement is within this class to the extent that the number of population units within each cluster asymptotically increases. The third condition requires the sampling fraction, n/N , to converge to a constant as the population size limits to infinity.

Samples may be drawn from the joint posterior distribution under Equation 1 using Markov chain Monte Carlo (MCMC), though the computation will not scale well with increasing sample size as a partition is formed through sequential assignments of observations to clusters (including to possibly new, unpopulated clusters) on each MCMC iteration. We, instead, derive an optimization expression that allows for scalable approximation of the single MAP partition by taking the limit of the joint posterior distribution as the likelihood variance, $\sigma^2 \downarrow 0$.

We marginalize out τ from the joint prior, $f(\mathbf{s}, \boldsymbol{\tau} | \boldsymbol{\alpha}) = f(\mathbf{s} | \boldsymbol{\tau}) f(\boldsymbol{\tau} | \boldsymbol{\alpha})$, by using the Pólya urn scheme of Blackwell and MacQueen (1973), which produces,

$$f(s_1, \dots, s_n | \boldsymbol{\alpha}) = \alpha^K \frac{\Gamma(\alpha + 1)}{\Gamma(\alpha + n)} \prod_{p=1}^K (n_p - 1)!,$$

where $n_p = \sum_{i=1}^n \mathbf{1}(s_i = p)$ is the number of observations (e.g. establishments) assigned to cluster, p and K denotes the number of estimated clusters. We generally follow Broderick et al. (2012) and derive our optimization expression from the joint pseudo posterior distribution,

$$\begin{aligned} f(\mathbf{s}, \mathbf{M} | \mathbf{X}, \tilde{\mathbf{w}}) \propto f(\mathbf{X}, \mathbf{s}, \mathbf{M} | \tilde{\mathbf{w}}) &= \prod_{p=1}^K \prod_{i: s_i=p} \mathcal{N}_d(\mathbf{x}_i | \boldsymbol{\mu}_p, \sigma^2 \mathbb{I}_d)^{\tilde{w}_i} \\ &= \alpha^K \frac{\Gamma(\alpha + 1)}{\Gamma(\alpha + n)} \prod_{p=1}^K (n_p - 1)! \\ &\quad \prod_{p=1}^K \mathcal{N}_d(\boldsymbol{\mu}_p | \mathbf{0}, \rho^2 \mathbb{I}_d). \end{aligned} \quad (2)$$

Broderick et al. (2012) point out that if one limits $\sigma^2 \downarrow 0$ in Equation 2, each observation will be allocated to its own cluster since the prior allows a countably infinite number of clusters. To avoid this degenerate outcome, define a constant λ and set $\alpha = \exp(-\lambda / (2\sigma^2))$, which

produces $\alpha \downarrow 0$ as $\sigma^2 \downarrow 0$. This functional form for α achieves the result and avoids the degenerate outcome, where the λ hyperparameter controls the size of the partition as $\sigma^2 \downarrow 0$. Commonly-used approaches that also specify an approximate MAP of a mixture of DPs also restrict α ; Shotwell and Slate (2011) fix the value for α and Wang and Dunson (2011) restrict the values of α to a discrete grid. Plugging in this expression for α into Equation 2, and taking -1 times the logarithm of both sides results in,

$$\begin{aligned} -\log f(\mathbf{X}, \mathbf{s}, \mathbf{M}, \tilde{\mathbf{w}}) &= \sum_{p=1}^K \sum_{i: s_i=p} \left[\mathcal{O}(\log \sigma^2) + \frac{\tilde{w}_i}{2\sigma^2} \|\mathbf{x}_i - \boldsymbol{\mu}_p\|^2 \right] \\ &\quad + K \frac{\lambda}{2\sigma^2} + \mathcal{O}(1) \end{aligned} \quad (3)$$

where the expression in the first line derives from the log-kernel of a multivariate Gaussian distribution and $f(\sigma^2) = \mathcal{O}(h(\sigma^2))$ denotes that there exist constants, $c_1, c_2 > 0$, such that $|f(\sigma^2)| \leq c_1 |h(\sigma^2)|$, for $\sigma^2 < c_2$. Multiplying both sides of Equation 3 by $2\sigma^2$ and taking the limit as $\sigma^2 \downarrow 0$ results in the asymptotically equivalent sampling-weighted global clustering optimization problem,

$$\underset{\mathbf{K}, \mathbf{s}, \mathbf{M}}{\operatorname{argmin}} \sum_{p=1}^K \sum_{i: s_i=p} \tilde{w}_i \|\mathbf{x}_i - \boldsymbol{\mu}_p\|^2 + K\lambda, \quad (4)$$

whose solution is a MAP approximation for the sample-adjusted mixtures of DPs of Equation 1. We minimize a sampling-weighted distance in Equation 4, which will approximate a MAP partition and associated cluster centers with respect to the distribution for the *population*, which is our interest, rather than that for the realized sample.

Although we derive the sampling-weighted approximate MAP by letting the variance around a partition, σ^2 , limit to 0, this constructive derivation offers no comment on whether

the distribution over the space of partitions for a particular data set expresses a low or high variance. Our purpose is to extract a hard, point estimate that computationally scales to a large number of multivariate observations (that we employ for the purpose of nominating outliers). To the extent that computational considerations allow, one would obtain richer inference on both an asymptotically exact MAP of the marginal distribution over the space of partitions and an associated measure of uncertainty by using the fully Bayesian nonparametric mixture model.

Shotwell and Slate (2011) note that employing a MAP approximation (which selects the partition assigned the largest approximated value for the posterior mass) implicitly uses a 0 – 1 loss function. While other choices for the loss function may produce useful results, we prefer use of the MAP approximation as we would expect concentration of the estimated joint posterior distribution from the model of Equation 1 around the MAP due to the large sample sizes for our CES application.

2.2 Sampling-weighted Global Clustering Algorithm

We specify an estimation algorithm for MAP approximation, below, which is guaranteed to converge to a local optimum, since each step reduces (or doesn't increase) the objective function of Equation 4 and there are only a finite number of possible partitions. The algorithm operates, sequentially, on the observations and specifies successive assignment and cluster center estimation steps.

We introduce a new merge step, that tests all unique cluster pairs for merging and conducts a merge of any pair of clusters if such reduces the objective function, which reduces the sensitivity of the estimated partition to the specified values of the penalty parameter.

The observations are each weighted using the sampling weights in the computations of distances to cluster centers and the computation of cluster centers. Observations with higher weights contribute relatively more information to the computation of cluster centers. We demonstrate in the simulation study to follow that this procedure produces nearly unbiased population estimates for cluster centers, conditioned on the generation of the finite population from disjoint clusters. Since lower-weighted observations (less representative of the population from which they were drawn) contribute relatively less to the objective function, the weighting will also impact the number of clustered discovered and assignment of the multivariate observations to those clusters.

The algorithm for estimating a MAP partition and associated cluster centers is specified with,

1. **Input:** $n \times d$ data matrix, \mathbf{X} ; $n \times 1$ vector of sampling weights, \vec{w} , for observed units and cluster penalty parameter, λ .
2. **Output:** $K \times d$ matrix of cluster centers, $\mathbf{M} = (\mu_1, \dots, \mu_K)'$, and $n \times 1$ vector of cluster assignments, \mathbf{s} , where $s_i \in \{1, \dots, K\}$ for units, $i = 1, \dots, n$.
3. **Initializer:** $s_i = 1, \forall i$. $d \times 1$ cluster center, $\mu_1 = \sum_{i=1}^n (\mathbf{x}_i \vec{w}_i) / \sum_{i=1}^n \vec{w}_i$.
4. Repeat the following steps until convergence (when the decrease in energy is below

$$e \leftarrow \sum_{p=1}^K \sum_{i \in \{i: s_i=p\}} w_i \|\mathbf{x}_i - \mu_p\|^2 + \lambda K$$

a set threshold).

5. **Assign units to clusters for each unit, $i = 1, \dots, n$:**
 - (a) Compute distance, $d_{ip} = \vec{w}_i \|\mathbf{x}_i - \mu_p\|^2$ for $p = 1, \dots, K$.
 - (b) If $\min_p d_{ip} > \lambda$, create new cluster to which unit i is assigned; $K \leftarrow K + 1$, $s_i \leftarrow K$, $\mu_K \leftarrow \mathbf{x}_i$; else $s_i \leftarrow \operatorname{argmin}_p d_{ip}$.
6. **Re-compute centers for clusters, $p = 1, \dots, K$:** Let $\mathbf{S}_p = \{i : s_i = p\}$ and $\mu_p = \sum_{i \in \mathbf{S}_p} (\mathbf{x}_i \vec{w}_i) / \sum_{i \in \mathbf{S}_p} \vec{w}_i$.

7. Assess merge of unique cluster pairs, $(p \in (1, \dots, K), p' \in (1, \dots, K))$:

- (a) Perform test merge of each pair of clusters:
- (b) Set matrix of cluster centers for virtual step, $\mathbf{M}^* = \mathbf{M}$.
- (c) Compose index set of observations assigned to clusters p' or p ; $\mathbf{S}_p^* = \{i : s_i = p \text{ or } s_i = p'\}$.
- (d) Compute (weighted average) merged cluster centers, $\mu_p^* = \left(\sum_{i \in \mathbf{S}_p^*} \mathbf{x}_i \tilde{w}_i \right) / \sum_{i \in \mathbf{S}_p^*} \tilde{w}_i$ and set rows p and p' of \mathbf{M}^* equal to this value.
- (e) Compute energy under the tested move, $e^* \leftarrow \sum_{p=1}^K \sum_{i \in \mathbf{S}_p^*} \tilde{w}_i \|\mathbf{x}_i - \mu_p^*\|^2 + \lambda(K-1)$. If $e^* < e$, execute a merge of p and p' :
- (f) Re-assign units linked to p' to p , $\mathbf{s}_{i:s_i=p'} \leftarrow p$.
- (g) Re-set cluster labels, p in \mathbf{s} to be contiguous after removing p' , such that $p \leftarrow p-1, \forall p > p'$, leaving $s_i \in (1, \dots, K-1)$. Delete row, p' , from \mathbf{M}^* and set $\mathbf{M} \leftarrow \mathbf{M}^*$.

2.3 Asymptotic Sampling-weighted Mixtures of Hierarchical Dirichlet

Processes

The CES survey is administered to establishments across a diverse group of industries that comprise the U.S. economy. CES survey analysts examine month-over-month reporting changes in the average employment and production worker variables within each of 24 NAICS industry categories (outlined in Section 4 that analyzes a data set of CES responses). The by-industry focus reflects the experience that both reporting patterns and establishment processes tend to express more similarity within than between industries. Yet, there are also commonalities in the reporting processes and types of reporting errors committed among the establishments due to overlapping skill sets of reporting analysts, similarities in organization structures, as well as the use of a single BLS-suggested reporting process.

So we want to perform separate estimations of partitions within each industry group, but allow those partitions to express similarities or dependencies across industries. We next

use the result of Kulis and Jordan (2011) that generalizes the approximate MAP algorithm obtained from mixtures of Dirichlet processes to mixtures of hierarchical Dirichlet processes. The hierarchical Dirichlet process (HDP) of Teh et al. (2006) specifies a DP partition for each industry that we refer to as a local (to industry) partition or clustering. These local partitions draw their cluster center values from a set of ‘‘global’’ clusters, such that local clusters may be connected across industry partitions by their sharing of common global cluster centers (from a single global partition) in a hierarchical manner.

We next specify a hierarchical mixture model that will converge to a sampling-weighted mixtures of HDPs in the limit of the *allowable* maximum number of global and local clusters as a generalization of our earlier DP specification,

$$\mathbf{x}_i^{d \times 1} | s_i^j, \mathbf{M} = (\mu_1, \dots, \mu_{K_{\max}}), \sigma^2, \tilde{w}_i \stackrel{\text{ind}}{\sim} \mathcal{N}_d(\mu_{s_i^j}, \sigma^2 \mathbb{I}_d), \quad i = 1, \dots, n_j \quad (5a)$$

$$v_c^j | \tau \stackrel{\text{iid}}{\sim} \mathcal{M}(1, \tau_1, \dots, \tau_{K_{\max}}), \quad c = 1, \dots, L_{\max} \quad (5b)$$

$$z_i^j | \pi^j \stackrel{\text{iid}}{\sim} \mathcal{M}(1, \pi_1^j, \dots, \pi_{L_{\max}}^j) \quad (5c)$$

$$s_i^j = v_{z_i^j}^j \quad (5d)$$

$$\mu_p | G_0 \stackrel{\text{iid}}{\sim} G_0 := \mathcal{N}_d(\mathbf{0}, \rho^2 \mathbb{I}_d) \quad (5e)$$

$$\tau_1, \dots, \tau_{K_{\max}} \sim \mathcal{D}(\alpha / K_{\max}, \dots, \alpha / K_{\max}) \quad (5f)$$

$$\pi_1^j, \dots, \pi_{L_{\max}}^j \sim \mathcal{D}(\gamma / L_{\max}, \dots, \gamma / L_{\max}), \quad (5g)$$

for each industry partition, $j = 1, \dots, (J = 24)$, where K_{\max} denotes the allowable number of global clusters and L_{\max} denotes the maximum allowable number of local clusters over all J partitions. An $n_j \times d$ data matrix, \mathbf{X}^j , includes the $d \times 1$ responses, (\mathbf{x}_i^j) , for establishments in industry, j . The $L_{\max} \times 1$ vector, \mathbf{v}^j , indexes the global cluster assignment for each local cluster, $c \in (1, \dots, L)$ for industry, j . The $n_j \times 1$ vector, \mathbf{z}^j , assigns each individual establishment, $i \in (1, \dots, n_j)$, to a local cluster, $c \in 1, \dots, L_{\max}$. So, z_i^j indexes

the local cluster assignment for establishment, i , in industry, j and v_i^j indexes the global assignment for all of the establishments assigned to local cluster, c , in industry, j . We may chain together the local cluster assignment for observation, i , and the global cluster assignment for the local cluster containing i to produce the $n_j \times 1$ index, s^j , that holds the global cluster assignment for each establishment in industry, j . The vectors of probabilities for local cluster assignments of establishments, $(\pi^j)_{j=1,\dots,J}$ and for global cluster assignments of local clusters, τ , are random probability vectors drawn from Dirichlet distributions. As before, Equation 5 converges to a sampling-weighted mixture of HDDPs in the limit to infinity of K_{\max} and L_{\max} (Teh et al. 2006).

We achieve the following optimization algorithm by following a similar asymptotic derivation in the limit of $\sigma^2 \downarrow 0$, as earlier,

$$\arg \min_{K, S, M} \sum_{p=1}^K \sum_{j=1}^J \sum_{i: s_i^j=p} \bar{w}_i^j \|\mathbf{x}_i^j - \boldsymbol{\mu}_p\|^2 + K\lambda_K + L\lambda_L, \quad (6)$$

where $L = \sum_{j=1}^J L_j$ denotes the total number of estimated local clusters and L_j denotes the number local clusters estimated for data set, $j = 1, \dots, J$. As before, K denotes the number of estimated global clusters.

2.4 Sampling-weighted Hierarchical Clustering Algorithm

We sketch a summary overview of the MAP approximation algorithm derived from sampling-weighted mixtures of HDDPs. A more detailed, step-by-step, enumeration of the algorithm is offered in Appendix A. As with the sampling-weighted global clustering estimation, the algorithm will return a $K \times d$ matrix of global cluster centers, \mathbf{M} , and a vector for each industry, s^j , $j \in (1, \dots, J)$, that holds the global cluster assignments, $s_i^j \in (1, \dots, K)$, for the set of n_j observations in industry, j . Each vector s^j conveys additional information

about the local partitions of the hierarchical clustering. The estimation algorithm allows a distinct number of local clusters, L_j , to be discovered for each industry. The number of unique global cluster values, $p \in (1, \dots, K)$, that are assigned across all the establishments in industry j defines the structure - number of local clusters and establishments assigned to each cluster - of the local partition for that industry.

The algorithm contains an assignment step to directly assign each establishment, i , in industry, j , to a global cluster, as with the mixtures of DPs algorithm. For a new global and local cluster to be created, however, the minimum distance to the global cluster centers must be greater than the sum of local and global cluster penalty parameters, $\lambda_L + \lambda_K$. An observation is assigned to a global cluster by first assigning it to a local cluster that is, in turn, next assigned to that global cluster. A new local cluster will be created if an observation in an industry is assigned to a global cluster, p , that is not currently assigned to any local cluster for that industry. It is typical for only a subset of the K global clusters to be used in each local partition across the J industries. To the extent that the industry partitions share global clusters, then $K < \sum_{j=1}^J L_j$, and there will be a dependence among those partitions.

A second assignment step performs assignments to global clusters for *groups* of establishments in industry, j , who are assigned to the same local cluster, c . All of the establishments in local cluster, c , in industry, j , may have their assignments changed from global cluster, p , to global cluster, p' . This group assignment of establishments to global clusters contrasts with only changing global cluster assignments for establishments on an individual basis. This is a feature of the HDP and helps mitigate order dependence in the estimation of the MAP approximation. We specify a new “shed” step (that is not included in the algorithm of Knulis and Jordan (2011)) to remove any previously defined

global clusters that may not be linked to any local clusters across the J industries after this second assignment step of local-to-global clusters.

We continue to employ a merge step that tests each unique pair of global clusters for merging. We find in runs performed on synthetic data that the merge step helps reduce (but does not eliminate) the sensitivity (in number of clusters formed) to specifications of (λ_K, λ_L) . The number of merges increase under lower values for these penalty parameters. If global cluster, p' , is merged into p and deleted, then so are the local clusters, \mathbf{c} , assigned to p' across the J industries and establishments that were previously assigned to each $c \in \mathbf{c}$ are now re-assigned to existing local clusters linked to global cluster, p .

3. Simulation Study

We conduct a two-part Monte Carlo simulation study that assesses the effectiveness to detect an outlying cluster with respect to the population from an informative sample, where an outlying cluster is defined to contain a small percentage of the total number of establishments. Both parts of the simulation study generate a synthetic finite ‘‘population’’ from which repeated samples are drawn under a sampling design where the probabilities of selecting each establishment are a function of their response values, such that we produce informative samples. The informative samples are presented to the partition estimation models. The first part of the simulation study compares the accuracy of outlier detection under inclusion of sampling weights, used to correct for informativeness, versus excluding the sampling weights (where both employ the global (hard) clustering model using the estimation algorithm specified in Section 2.2. The sampling weights are set to a vector of 1’s in the case that excludes sampling weights). The second part of the simulation study compares the outlier estimation performance between the sampling-weighted global clustering model, on the one hand, and the sampling-weighted hierarchical clustering model,

on the other hand, for data generated from local partitions that share global cluster centers. We first devise an approach to select the penalty parameters for our estimation models, which determine the number of clusters discovered, that we will need for our two-part study.

3.1 Selection of Penalty Parameters

It is difficult to find a principled way to specify the penalty parameter, λ , for the global clustering model or (λ_L, λ_K) , for the hierarchical clustering model. Their specification is critical because these penalty parameters determine the number of discovered clusters. Relatively larger values estimate fewer clusters. Both cross-validation and randomly separating the data into distinct training and test sets (in the case the analyst possesses a sufficient number of observations) tends towards assignment of each observation to its own cluster by selecting nearly 0 values for the penalty parameters as poor fit overwhelms the penalty.

We demonstrate this tendency with a simulation study using the sampling-weighted hierarchical clustering model. We generate a population with $J = 3$ data sets, each of size $N_j = 15000$, with the local partition structure in each data set composed of $L_j = 5$, $j = 1, \dots, J$ clusters randomly selecting from $K = 7$ global cluster centers. Establishments are randomly assigned to the $L_j = 5$ local clusters for each data set under a skewed distribution of $(0.6, 0.25, 0.1, 0.025, 0.025)$ to roughly mimic the structure we find in the CES data set analysed in Section 4. Data observations are generated from a multivariate Gaussian,

$$\mathbf{x}_i^j \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}_d \left(\boldsymbol{\mu}_j, \sigma_j^2 \mathbb{I}_d \right),$$

centered on the local cluster mean of each observation and σ is set equal to 1.5 times the average over the $J = 3$ data sets, $N_j = 15000$ establishments and $d = 15$ dimensions of the assignment-weighted centers, $\mathbf{M}_{s,1:d}$, the mean values of \mathbf{X}_j .

We next take a sample of $n_j = 2250$ from each of the $J = 3$ data sets under a fixed sample size, proportional-to-size design where inclusion probabilities are set to be proportional to the variances of the $(d = 15) \times 1$, $\{\mathbf{x}_i\}$, inducing informativeness. The taking of an informative sample addresses the class of data in which we are interested and mimics our motivating GES data application. Observations are next randomly allocated into two sets of equal size; one used to train the model and the other to evaluate the resultant energy. The training sample is presented to the sampling-weighted hierarchical clustering model. Figure 1 displays the sampling-weighted hierarchical objective function (energy) values, evaluated on the test set, over a grid of global and local cluster penalty parameters. We observe that the energy doesn't reach a balanced optimum, but decreases along with values of the penalty parameters. The rate of decrease declines for these synthetic data, possibly suggesting the identification of an "elbow" for selecting penalty parameters, though there is little-to-no sensitivity in the values for the global penalty parameter, λ_K , which prevents precise identification of a value. The rate of decrease in energy on the GES data does not decline, however. We see a similar result under 10-fold cross-validation.

As an alternative, we employ penalty parameter selection statistics that combine measures of cohesion within clusters and separation between clusters to select the number of clusters. We devise a sampling-weighted version of the Calinski Harabasz (C) criterion, which is based on within cluster sum-of-squares (WSS) and between clusters sum-of-

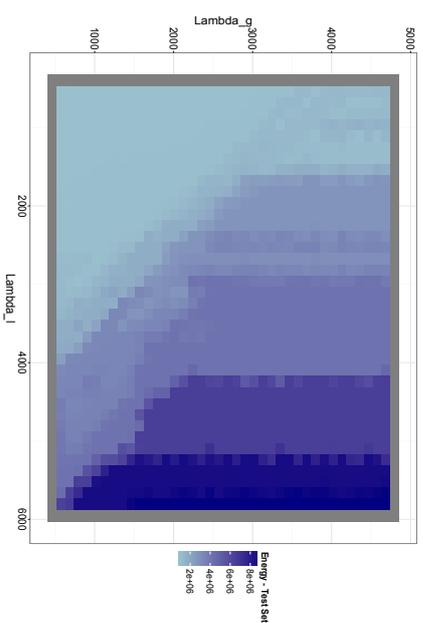


Figure 1: Heat map of (energy) values under the sampling-weighted hierarchical clustering model over grid of global and local cluster penalty parameters. The vertical axis represents the global cluster penalty, λ_K , and the horizontal axis, the local cluster penalty, λ_L . The energy is measured from assignment to a test set not used for training the model. The darker colors represent higher energy (optimization function) values.

squares ($BGSS$),

$$WGSS = \sum_{p=1}^K \sum_{i: s_i^k = k} \tilde{w}_i \|\mathbf{x}_i - \boldsymbol{\mu}_p\|^2$$

$$BGSS = \sum_{p=1}^K n_p \|\boldsymbol{\mu}_p - \boldsymbol{\mu}^G\|^2,$$

where $\mathbf{s}^v = (\mathbf{s}^1, \dots, \mathbf{s}^v)'$ stacks the set of J_i $\{\mathbf{s}^j\}$ into a vector, which retains information on the local partitions (based on the co-clustering relationships). The weighted total number of establishments linked to each cluster, $p \in (1, \dots, K)$ is denoted by $n_p = \sum_{i: s_i = p} \tilde{w}_i$ and $\boldsymbol{\mu}^G = \frac{\sum_{i=1}^n \tilde{w}_i \mathbf{x}_i}{\sum_{i=1}^n \tilde{w}_i}$. The C criterion is then formed as, $C = \frac{n-K}{K-1} \frac{WGSS}{BGSS}$, where K is determined by the number of unique values in \mathbf{s}^v , which is equal to the number of rows in the $K \times d$, $\mathbf{M} = (\boldsymbol{\mu}'_1, \dots, \boldsymbol{\mu}'_K)$ (and d denotes the dimension of each observation, \mathbf{x}_i). Larger values for C are preferred.

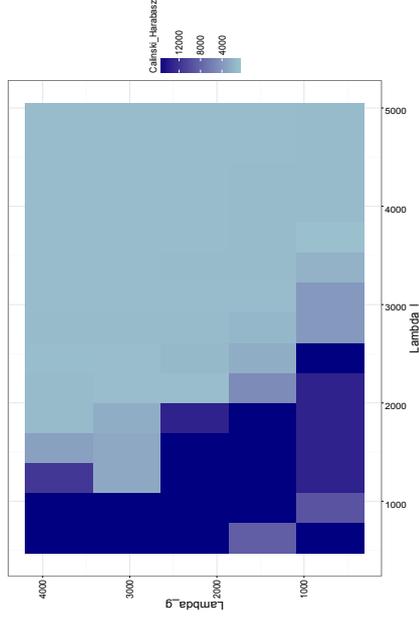


Figure 2: Heat map of Calinski Harabasz (C) index values under the mixtures of HDP model over grid of global and local cluster penalty parameters. The vertical axis represents the global cluster penalty, λ_K , and the horizontal axis, the local cluster penalty, λ_L . The darker colors represent higher C values. Global and local cluster penalty parameters with higher C values are preferred.

Our proposed procedure estimates a clustering on the sampled data over a grid of (local, global) penalty parameter values, where we compute the C statistic for each (λ_L, λ_K) in the grid, selecting that clustering which maximizes the C statistic. (Our `growclusters` package uses the farthest-first algorithm to convert user-specified minimum and maximum number of global and local clusters to the associated penalty parameters. The farthest first is a rough heuristic, so we are conservative in the specified range of $1 \geq \lambda_K \leq 30$, $1 \geq \lambda_L \leq 20$). We perform estimation using the sampling-weighted hierarchical clustering algorithm over a 5×15 grid of global and local clustering penalty parameters, respectively; on the full sample of $n_j = 2250$ for the $J = 3$ data sets. Figure 2 suggests selection of penalty parameters from the upper left-quadrant of the grid. We chose the values of $(\lambda_L = 1232, \lambda_K = 2254)$ that maximized the C index for our sampling-weighted hierarchical clustering model. Figure 3 presents the resulting estimated distribution of the n_j (informative sample) observations within clusters of each local partition (that index the $J = 3$ datasets), where the support of each local distribution indicates to which global cluster center each local cluster is linked. We note that the correct number ($K = 7$) of global clusters and local clusters ($L_j = 5$) is returned and that the skewed distribution of establishments within each local partition mimics that of the population (which is estimated from our informative sample). We use the Rand statistic $\in (0, 1)$, which measures the concordance of pairwise clustering assignments, to compare the true partition assigned in the the population versus the estimated partition, where the latter is estimated from the observed informative sample, rather than the population, itself. The computed Rand value is 1, indicating perfect agreement between the true and estimated partitions in their pairwise clustering assignments. (See Shotwell (2013) for a discussion of alternative statistics, including the Rand, for comparing the similarity of two partitions).

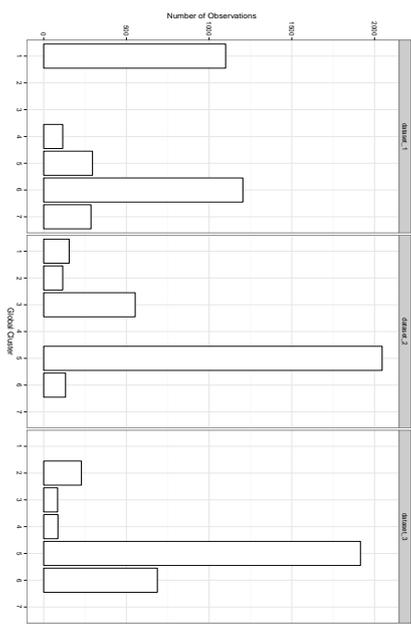


Figure 3: Estimated distribution of establishments in $J = 3$ local cluster partitions. The support of each distribution indicates the global cluster to which each local cluster is linked.

We additionally compute a sampling-weighted version of the silhouette index, $h_i = \frac{a_i - b_i}{\max(a_i, b_i)}$, where a_i is the average distance to observations co-clustered with i and b_i is the average distance to other observations in the nearest cluster to that holding i . The total index, $h^K = \frac{\sum_{i=1}^n w_i h_i}{\sum_{i=1}^n w_i}$, $\in (0, 1)$ and, like the C index, prefers partitions where the clusters are compact and well-separated. These two indices select the same penalty parameter values from the grid of values evaluated on both the simulation and CES data. We prefer our sampling-weighted C since the computation is more scalable to a large number of observations. The C index was used to select the penalty parameters for all implementations of the mixtures of DPs and HDDPs models.

Lastly, our procedure for selecting (λ_L, λ_K) by evaluating the Calinski Harabasz statistic over a grid generally selects the same clustering under inclusion or exclusion of the merge move on synthetic datasets. Figure 4 examines the numbers of merges that take place on

each run on the grid of penalty parameter values in the case we include the merge step, and reveals that the number of merges increases at relatively low values of the penalty parameters. In this case, and in others we tested on synthetic data, the merges took place outside of the range of nearly optimum penalty parameter values, though one may imagine the possibility of merges taking place in this range on a real dataset. There would be a reduction in the sensitivity for the number of clusters estimated to the values of the penalty parameter, which may have the effect of increasing the sub-space of nearly optimal penalty parameters (that produce the same clustering results). While we recommend and employ our selection procedure for penalty parameters in our simulations and application to follow, further simulations (not shown) demonstrate that inclusion of the merge step produces estimated clusters that are consistently closer to the truth as compared to excluding the merge step when the penalty parameters are specified by a heuristic procedure, like the farthest first algorithm employed in Kulis and Jordan (2011).

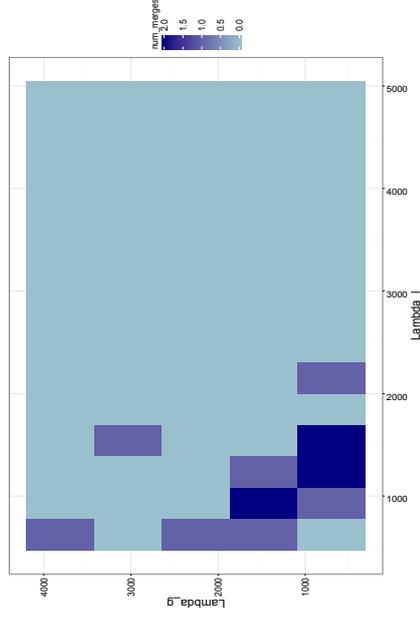


Figure 4: Heat map of the number of merges that took place under the mixtures of HDP estimation model over grid of global and local cluster penalty parameters. The vertical axis represents the global cluster penalty, λ_K , and the horizontal axis, the local cluster penalty, λ_L . The darker colors represent higher number of merges.

Our selection procedure that computes the Calinski Harabasz statistic over a range of global and local cluster penalty parameters defined on a grid and selects that clustering which produces the maximum value of this statistic will be used for every estimation run of the global and hierarchical clustering algorithms in the sequel.

3.2 Outlier Estimation Under Informative Sampling

Our next simulation study compares the outlier detection power when accounting for, versus ignoring, the informativeness in the observed sample by generating data under a simple, non-hierarchical global partition. We construct a set of $K = 5$ global cluster centers, each of dimensions, $d = 15$. Each cluster center expresses a distinct pattern over the $d = 15$ dimensions,

$$\stackrel{(d=15) \times 1}{\boldsymbol{\mu}_1} = (1, 1.5, 2.0, \dots, 7.5, 8)$$

$$\boldsymbol{\mu}_2 = (8, 7.5, \dots, 1)$$

$$\boldsymbol{\mu}_3 = (1, \dots, 7, 8, 7, \dots, 1)$$

$\boldsymbol{\mu}_4 =$ Sampling from $(1, \dots, 8)$ under equal probability with replacement, $d = 15$ times

$\boldsymbol{\mu}_5 =$ Sampling from $(-2, \dots, 6)$ under equal probability with replacement, $d = 15$ times,

to generate $\mathbf{M} = (\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_5)'$. Without loss of generality, the last cluster, $\boldsymbol{\mu}_5$, is defined as the outlier cluster and includes negative values, though there is overlap with the support of the other four clusters for values > 0 . We create the $(N = 25000) \times 1$ cluster assignment vector, \mathbf{s} , by randomly assigning establishments in the (finite) population to clusters (with equal probabilities) such that the first 4 clusters are assigned equal numbers of observations, while the last (outlying) cluster (with mean $\boldsymbol{\mu}_5$ is assigned 150 observations, according with our earlier definition of an outlying cluster as having relatively few assigned observations).

We then generate our $N \times d$ matrix of population response values, \mathbf{X} , from the multivariate Gaussian distribution,

$$\stackrel{d \times 1}{\mathbf{x}_i} \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}_d(\boldsymbol{\mu}_s, \sigma^2 \mathbb{I}_d),$$

where the standard deviation, σ , is set equal to 1.5 times the average over the $N = 25000$ establishments and $d = 15$ dimensions of the assignment-weighted matrix, $\mathbf{M}_{\mathbf{s},1:d}$, the mean values of \mathbf{X} . We may think of this process for generating a finite population as taking a census of all establishments in the population. Establishments who report values in any of the first 4 clusters are drawn from the true population generating distribution, while those establishments who report values from cluster 5 commit errors such that the reported values are from a different (shifted) distribution that generates the population of errors. Our inferential interest is to uncover outlying values with respect to the population, rather than the sample, as an observation may not be outlying relative to the sample, but may relative to the population (or vice versa).

We assign the population establishments, evenly, to one of $H = 10$ strata, so there are $N_h = 2500$ establishments assigned to each stratum. Our sampling design employs simple random sampling of establishments within each of the H strata. The sample size taken from each stratum is set proportionally to the average of the by-establishment variances of the $(d = 15) \times 1$, $\{\mathbf{x}_i\}$ for establishments, $i = 1, \dots, N_h$, assigned to each stratum, $h = 1, \dots, H$. Each generated sample produces the by-stratum sample sizes, $\mathbf{n} = (45, 90, 136, 181, 227, 272, 318, 363, 409, 459)$, ordered according to variance quantile, from left-to-right, for a total sample size of $n = \sum_{h=1}^H n_h = 2500$. This sampling design assigns higher probabilities to larger variance strata (and all establishments in each stratum have an equal probability of selection), which is often done, in practice, because there is expected to be more information in these strata.

The population of establishments are assigned to clusters based on the mean values of \mathbf{x}_i , not the associated variances (as the variances in each cluster are roughly equal). We conversely use the variance of each \mathbf{x}_i rather than the mean, in order to construct our sampling design with the goal to produce sampling inclusion probabilities that are nearly independent from the probabilities of assignment to the population clusters. So our model formulation doesn't (inadvertently) parameterize the sampling design, which is the most general set-up.

We generate a population and subsequently draw $B = 100$ samples under our informative single stage, stratified sampling design with inclusion probabilities of each stratum proportional to the average of the variances of member establishment response values (across the $d = 15$ dimensions), as described above. We run our sampling-weighted global clustering algorithm on each sample under two alternative configurations: 1. excluding the sampling weights, so that we do not correct for the informative sampling design; 2. including the sampling weights, such that we estimate outliers and cluster centers with respect to the clustering parameters for the population, asymptotically, conditioned on the generation of the finite population from disjoint clusters. We include two methods as comparators; firstly, we utilize the model-based clustering (MBC) algorithm of Fraley and Raftery (2002) that defines a finite mixture of Gaussians model of similar form as our (penalized) mixtures of DPs construction. Fraley and Raftery (2002) employ the EM algorithm to solve their mixture model (initialized by a hierarchical agglomeration algorithm) and select the number of clusters, K , using the Bayesian information criterion (BIC); secondly, we include the trimmed K-means method of Fritz et al. (2012), which intends to robustify the K-means algorithm by removing outlying data points that may induce mis-estimation of the partition. The authors note that these outlying points may be used to nominate outliers. Both implementations exclude employment of sampling weights to correct for informative

sampling. We also considered to include the algorithm of Shotwell and Slate (2011), but it did not computationally scale to the size of data we contemplate for our CES application.

The left-hand set of box plots in Figure 5 displays the distributions (within 95% confidence intervals under repeated sampling) of the true positive rate for identifying outlying observations, constructed as the number of true outliers discovered divided by the total number of true outliers, estimated on each Monte Carlo iteration for our three comparator models. The right-hand set of box plots display the false positive rate, which we define as the number of false discoveries divided by the total number of observations nominated as outliers. The inclusion of false positives permits assessment of the efficiency to detect outliers. Each set of box plots compares estimation under the global clustering algorithm *including* sampling weights, on the one hand, to a version of the global clustering algorithm, the MBC, and trimmed K-means, on the other hand, that all *exclude* the sampling weights. Outliers were detected in each simulation iteration, $b = 1, \dots, B$, based on selecting those clusters whose total observations (among the selected clusters) cumulatively summed to less than $C = 1.1$ times the total number of true outliers in the informative sample, which is how we would select outlying clusters on a real dataset where we don't know the truth. (We experimented with different values of $C \in [1, 1.5]$ and realized the same comparative results, as presented below).

Figure 5 reveals that failure to account for the informative sampling design induces a deterioration in outlier detection accuracy.

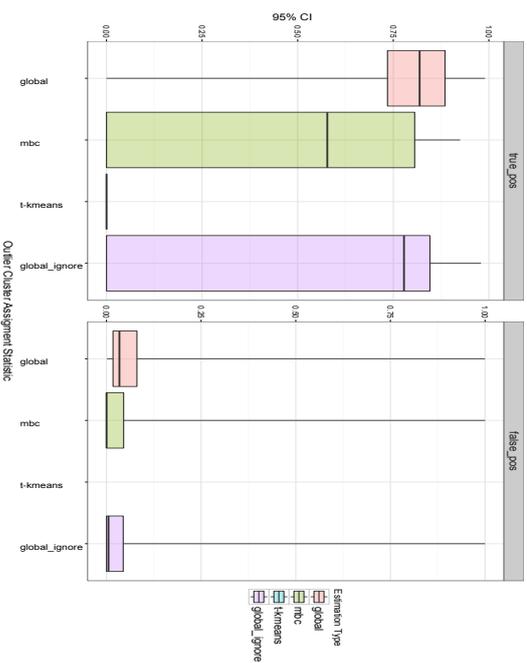


Figure 5: Accuracy of outlier detection under informative sampling. The left-hand plot panel presents the distributions of the true positive rates, and the right-hand panel presents the distribution for the false positive rates, both within 95% confidence intervals estimated from $B = 100$ Monte Carlo draws of informative samples. Each sample is of size, $n = 2500$, from a population of size, $N = 25000$, with a population cluster of outliers of size, $N_5 = 250$. The left-hand box plot within each plot panel is estimated from the sampling-weighted global clustering algorithm that accounts for the informative sampling design by including sampling weights, while the middle two box plots represent the model-based clustering (MBC) and trimmed K-means algorithms, respectively, that ignore the informative sampling design as does the right-hand box plot that represents the global clustering algorithm without inclusion of sampling weights.

Figure 6 presents the distribution over the number of discovered clusters for each of

the three comparator models: 1. global clustering model, *including* sampling weights; 2. MBC, *excluding* sampling weights; 3. Trimmed K-means, *excluding* sampling weights; 4. global clustering model, *excluding* sampling weights. The dashed line at $K = 5$ clusters is the correct generating value. While the the models excluding the sampling weights (except for the trimmed K-means) estimate a higher number of clusters, such is not the primary reason for their reduced outlier detection accuracy, as we observe from Figure 5 that the false positive rates are slightly lower for these two models compared to the model that includes the sampling weights. The reduced accuracy is primarily driven by biased estimation of the $d \times 1$ cluster centers, $\{\mu_{p_j}\}_{j=1,\dots,K}$, (relative to the population), whose estimation is performed together with assignment to clusters in a single iteration of the algorithm. We examine this bias in the next simulation study.

The trimmed K-means does relatively well in capturing the number of true clusters, absent the outlying cluster. The trimming is not isolating these points as outliers, however, but collapsing them into a larger cluster; hence, the trimmed k-means does not nominate any outliers.

3.3 Comparison of Outlier Estimation between Hierarchical and Global Clustering under Informative Sampling

We now generate a set of local partitions hierarchically linked to a collection of global cluster centers. We generate $J = 8$ local populations, $(\mathbf{X}^j)_{j=1,\dots,J}$, each of size $N_j = 25000$, and associated local partitions, $(\mathbf{s}^j)_{j=1,\dots,J}$, where each local partition contains, $L_j = 2$ clusters, including one that is composed of 150 outliers. The set of J local partitions randomly select their local cluster centers from the same $K = 5$ global cluster centers that we earlier introduced, which induces a dependence structure among the set of local partitions. We conduct $B = 100$ Monte Carlo draws, where we take an informative sample within each of the $J = 8$ datasets on each draw, using the same stratified sampling informative design, described above. Estimation is conducted on the set of J informative samples produced in each draw using both the sampling-weighted global clustering algorithm and the hierarchical clustering algorithm outlined in Section 2.4. Our goal is to uncover the global cluster center for the outlier cluster and the set of observations in each local dataset that are assigned to the outlier cluster. We concatenate the set of informative samples generated over the J datasets when conducting estimation using the global clustering algorithm because our primary interest is in outlier detection, rather than inference on the local partitions. (The performance of the global clustering algorithm relative to the hierarchical clustering algorithm is worse than shown in the case we separately estimate a global clustering on each dataset).

Figure 7 is of the same format as Figure 5, with two panels displaying distributions over true positive and false positive rates, respectively, for our choice of models. The set of five box plots compare the following models: 1. hierarchical clustering model, including the sampling weights to both estimate global partitions linked to a global partition and control for informative sampling; 2. global clustering model, including sampling weights to

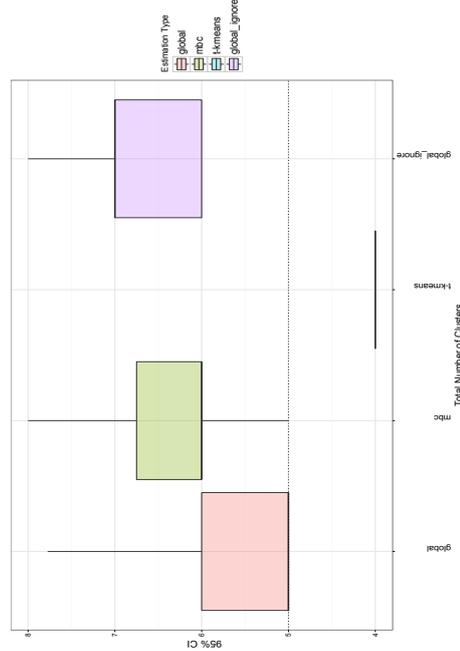


Figure 6: Comparison of distributions for number of estimated clusters, K , between the global clustering model including the sampling weights in the left-hand box plot, to the MBC in the middle box plot and the global clustering model in the right-hand box plot, where both exclude the sampling weights.

control for informative sampling; 3. model-based clustering, excluding sampling weights; 4. trimmed K-means robust clustering, excluding sampling weights; 5. global clustering model, excluding sampling weights. Results for the hierarchical clustering model, shown in the left-hand box plot of each panel, outperforms the global clustering model, where both include sampling weights, because the hierarchical model additionally estimates the dependent set of local clusters. The hierarchical clustering algorithm appears to do a better job of borrowing estimation information among the $J = 8$ local partitions.

Figure 8 presents distributions for each dimension, $d = 1, \dots, 15$, of the outlier cluster center, μ_5 , under four of the five comparison models (excluding the trimmed K-means) in the same order as displayed in Figure 7. The sampling-weighted hierarchical clustering model produces perfectly *unbiased* estimates for the population values of the global outlier cluster center, while the other 3 models (excluding the trimmed K-means) induce bias in proportion to their outlier detection performances. Unbiased estimation of the outlier cluster center as compared to the other clusters is important to detect an outlying cluster because the cluster centers encode the degree of separation of the outlying cluster(s) from the others. We see that the true positive detection rates across the methods shown in Figure 7 are roughly proportional to the levels of bias estimated in the dimensions of the outlying cluster center displayed in Figure 8. Since the trimmed K-means does not nominate any outliers, we cannot compute a outlying cluster center under this method.

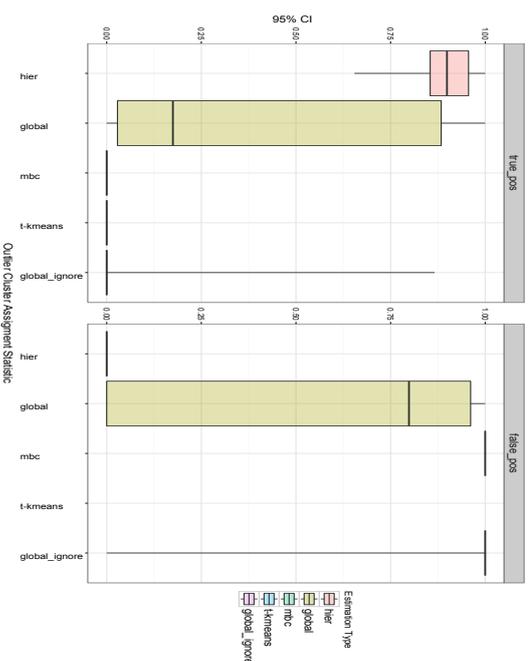


Figure 7: Accuracy of outlier detection under informative sampling for global vs hierarchical clustering model: The left-hand panel presents the distributions of the true positive rate, and the right-hand panel presents the distributions for the false positive rate, both within 95% confidence intervals estimated from $B = 100$ Monte Carlo draws of informative samples from each of $J = 8$ local populations. Each sample is of size, $n_j = 2500$, $j = 1, \dots, 8$, from a population of size, $N_j = 25000$, with a cluster of outliers of size, $N_{j,5} = 125$. The box plots represent the following models, from left-to-right: 1. hierarchical clustering model, including the sampling weights to both estimate global partitions linked to a global partition and control for informative sampling; 2. global clustering model, including sampling weights to control for informative sampling; 3. model-based clustering excluding sampling weights; 4. trimmed K-means, excluding sampling weights; 5. global clustering model excluding sampling weights.

We also examined the effect of excluding the merge move for the hierarchical clustering algorithm in this simulation study. While excluding the merge move induced an over-estimation of the the number of clusters (6 instead of the true 5), the outlier detection accuracy wasn't notably impacted, likely because employment of the C algorithm for selection of the number of local and global clustering penalty parameters protected against a large magnitude misestimation of the number of global and local clusters. We find a larger discrepancy between employment or not of the merge move for a *single* run of the clustering algorithm where the penalty parameters are set in an ad hoc fashion (e.g., through use of the farthest-first algorithm as in Kulis and Jordan (2011), a procedure that we do not recommend).

4. Application

We apply our hierarchical clustering algorithm to a data set of one month changes in CES survey responses for estimation of industry-indexed (local) partitions that may express a dependence structure across industries by potentially sharing global clusters. Outliers are nominated by selecting all establishment observations in any local cluster that holds a small percentage of observations (e.g. < 1%). Our CES survey application focuses on the set of 108017 establishments whose reported employment statistics differ between November and December, 2009 as a typical illustration. We are interested to flag those establishments whose responses express unusually large changes in employment statistics between November to December, relative to their November employment levels. So we normalize the observed statistics to,

$$\delta_{ijt} = \frac{x_{ijt}}{x_{ij(t-1)}}, j = 1, \dots, d \tag{7}$$

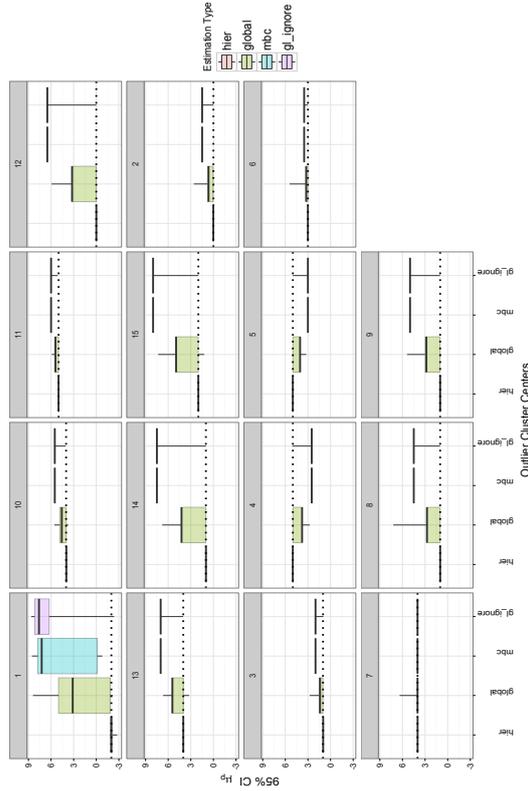


Figure 8: Comparison of estimation bias for each of $d = 15$ dimensions of the outlier global cluster mean, μ_5 , estimated from $J = 8$ local partitions under the following models, from left-to-right: 1. hierarchical clustering model, including the sampling weights to both estimate global partitions linked to a global partition and control for informative sampling; 2. global clustering model, including sampling weights to control for informative sampling; 3. model-based clustering, excluding sampling weights; 4. global clustering model, excluding sampling weights. The dashed line in each panel is the true value the center for each dimension, $d = 1, \dots, 15$

which has support on the positive real line that we, in turn, cluster. The distribution of δ_{jt} is highly right-skewed for all dimensions, $j \in \{1, \dots, d\}$, so we use a logarithm transform of the statistic to perform clustering in order to conform to the mixtures of Gaussians assumptions in Equation 5a. Alternatively, we could replace the squared Euclidean distance in Equation 6 with the Bregman divergence (which is uniquely specified for exponential family distributions) and replace the mixtures of Gaussian distributions with Exponential distributions, as suggested by Jiang et al. (2012) (which produces the same results on these data).

We focus on four employment variables of particular importance to CES (that also express high response rates): 1. “ae”, which is the total employment for each reporting establishment; 2. “pw”, which is production worker employment for each establishment; 3. “npr”, which is the total payroll dollars expended for employees of each establishment; 4. “nhr”, which is the average weekly hours worked for each establishment. So the dimension of our CES survey application is $d = 4$.

We select penalty parameters from the range, $(9 \leq \lambda_L \leq 829, 17 \leq \lambda_K \leq 187)$, on a 15×20 grid, using the Calinski Harabasz and silhouette statistics, which both choose $(\lambda_L = 19, \lambda_K = 145)$ from this grid. The selected penalty parameters did not change when including or excluding the merge step. Nevertheless, the number of clusters estimated without inclusion of the merge step was $K = 10$, while $K = 9$ was estimated when including the merge step. The resulting inference on the nomination of outliers, however, was unchanged so that we prefer the parsimonious model (in terms of number of clusters selected). The selected penalty parameters and resulting estimated clustering also did not change when we employed a finer grid. Results from inclusion of the merge step are presented in Table 1, where we discover $K = 9$ global clusters shared among the $J = 23$ local partitions, and each local partition holds between $L_j = 2 - 9$ clusters. Each column of Table 1 presents

the results for one of the $K = 9$ global clusters, from left-to-right in descending order of the number of establishments assigned to each global cluster. The first four rows (labeled “ae_ratio”, “pw_ratio”, “npr_ratio”, “nhr_ratio”) present the estimated global cluster centers, μ_{g_i} , for change ratio in ae, pw, npr, and nhr respectively, after reversing the logarithm transform. The fourth through final rows present the by-industry, local partitions, s_i^j , and their links to the global cluster centers. Scanning each of the columns, we see that all of the $K = 9$ global clusters are shared among the local partitions, indicating a high degree of dependence among them.

The fifth row (labeled “ae_avg”) averages the total employment, ae, over all establishments in all industries linked to the associated global cluster as reported for the month of November. The second column from the right represents a cluster whose centers indicate unusually large increases in reported employment and payroll from November-to-December. The establishments linked to this cluster are of generally small-sized, with an average of 20 reported employees in November. Establishments with a relatively small number of employees will receive high sampling weights due to their low inclusion probabilities, such that their high magnitude shifts in reported employment levels may be influential in the estimation of the sampling-weighted total employment estimates for November and December. It might, however, be expected that the retail (and associated wholesale) hiring might dramatically increase in anticipation of holiday shopping. So we would nominate the remaining 470 establishments linked to this global cluster as outliers for further analysis investigation by the BLS. Previous analysis conducted within the BLS suggests that smaller establishments generally tend to commit submission errors at a higher rate. The result excluding the merge step splits this cluster into two in a manner that doesn’t change inference about outlier nominations.

The smallest-size (right-most) cluster has a mean of 0.01 in the monthly change ratio for the variable “npr”, indicating an unusually large magnitude decrease in the number of employees (and payroll dollars) reported from November-to-December. This cluster contains 113 relatively large-sized establishments with an average of 278 reported employees in November. The moderate-to-large size of establishments in this cluster will tend to receive smaller sampling weights, however, (because they have higher sample inclusion probabilities), and so are less influential. So BLS would generally place a lower priority for investigation on this cluster than the previous discussed. The small number of establishments in this cluster, coupled with the large magnitude decreases in reported employment variables, however, would likely prompt an investigation of the submissions for all establishments in this cluster. The two Retail industries show 17 establishments in this cluster of establishments expressing a large decrease in employment. The seasonal hiring in this industry, mentioned above, may suggest to place an especially high priority on investigation of these establishments.

5. Discussion

The BLS seeks to maintain a short lead time for the CES survey between receipt of monthly establishment responses and publication of estimates. Retaining estimation quality requires investigation of establishment responses for errors, which may prove influential in published estimates (by region and industry, for example). BLS analysts, therefore, require an automated, fast-computing tool that nominates as outliers for further investigation a small subset of the over 100000 establishments whose responses reflect month-over-month changes in employment levels.

We have extended the MAP approximation algorithm of Kulis and Jordan (2011) and Broderick et al. (2012) for estimation of partitions among a set of observations to appli-

Table 1: CES Mixtures of HDPs: Global cluster centers and local partitions

Variable Values, by cluster										
	ac_ratio	0.961	0.98	1.157	0.829	0.657	2.008	0.156	19.212	0.093
	pw_ratio	0.951	0.973	1.199	0.797	0.596	2.586	0.189	11.95	0.146
	npr_ratio	0.939	0.963	1.283	0.423	0.465	4.106	0.143	16.617	0.010
	nhr_ratio	0.944	0.979	1.213	0.639	0.547	2.469	0.182	20.37	0.109
Average Employment Count, by cluster										
ac_avg (November)	168	205	104	154	125	182	258	20	278	
Units in Super Sector Clusters										
Agriculture	0	81	0	0	16	0	0	0	0	0
Mining	379	0	118	0	105	0	20	10	3	
Utilities	0	405	0	0	0	0	4	2	1	
Construction	2977	0	1111	0	1008	174	125	34	9	
Manufacturing(31)	1015	0	254	0	0	0	28	6	6	
Manufacturing(32)	1229	0	272	0	131	30	19	5	3	
Manufacturing(33)	0	2117	494	169	0	46	34	16	3	
Wholesale	1828	0	486	0	161	47	26	23	3	
Retail(44)	0	14648	4484	915	0	160	168	33	7	
Retail(45)	0	9381	5243	996	0	132	58	17	10	
Transportation(48)	1266	0	346	155	0	41	26	16	4	
Transportation(49)	0	1149	703	101	0	64	39	11	14	
Information	1852	0	456	198	0	30	22	30	1	
Finance	4947	0	1894	476	0	138	51	111	6	
Real Estate	1409	0	981	0	159	0	18	15	0	
Professional Services	2999	0	1151	349	0	149	70	47	5	
Management of Companies	0	881	185	64	0	0	6	6	1	
Waste Mgmt	3674	0	1301	0	602	144	103	40	11	
Education	666	0	229	0	0	23	19	6	0	
Health Care	0	6437	1068	368	0	126	63	23	14	
Arts-Entertainment	953	0	308	0	213	0	42	25	2	
Accommodation	11757	0	2541	822	0	181	100	49	7	
Other Services	1499	0	433	0	158	51	11	18	3	
	38450	35099	24058	4613	2553	1536	1052	543	113	108017

cations where the observations were acquired under an informative sampling design. We replace the usual likelihood in estimation of the joint posterior over partitions and associated cluster centers with a pseudo-likelihood that incorporates the first order sampling weights that serve to undo the sampling design. The resulting estimated parameters of the approximate MAP objective function are asymptotically unbiased with respect to the population, conditioned on the generation of the finite population from disjoint clusters. Our simulation study demonstrated that failure to correct for informative sampling reduces the outlier detection accuracy by inducing biased estimation of the cluster centers.

Our use of sampling-weighted partition estimation algorithms focuses on outlier detection, so we incorporated a new merge step, which may increase robustness against estimation of local optima and encourage discovery of clusters containing large numbers of observations, which may be a feature for outlier detection where we nominate as outliers those observations in clusters containing relatively few observations. We additionally constructed the sampling-weighted C statistic that our simulation study demonstrated was very effective for selection of the local and global cluster penalty parameters, (λ_L, λ_G) , that together determine the numbers of local and global clusters.

The sampling-weighted hierarchical clustering algorithm permitted our estimation of industry-indexed local clusters, which fits well into the BLS view that industry groupings tend to collect establishments with similar employment patterns and reporting processes. We saw that all of the estimated $K = 9$ global clusters for percentage change in employment from November to December, 2009 were shared among the local by-industry clusters, which served to both sharpen estimation of the local partitions and global cluster centers and to discover small global clusters of potentially outlying observations.

Acknowledgments

The authors wish to thank Julie Gershunskaya, a Mathematical Statistician colleague at the Bureau of Labor Statistics, for her clever insights and thoughtful comments that improved the preparation of this paper.

References

- D. Blackwell and J. B. MacQueen. Ferguson distributions via Polya urn schemes. *The Annals of Statistics*, 1:353–355, 1973.
- Daniel Bonny, F. Jay Breidt, and Francois Coquet. Uniform convergence of the empirical cumulative distribution function under informative selection from a finite population. *Bernoulli*, 18(4):1361–1385, 11 2012. doi: 10.3150/11-BEJ369. URL <http://dx.doi.org/10.3150/11-BEJ369>.
- T. Broderick, B. Kulis, and M. I. Jordan. MAD-Bayes: MAP-based Asymptotic Derivations from Bayes. *ArXiv e-prints*, December 2012.
- Chris Fraley and Adrian E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631, 2002.
- Heinrich Fritz, Luis A. García-Escudero, and Agustín Mayo-Iscar. tclust: An R package for a trimming approach to cluster analysis. *Journal of Statistical Software*, 47(12):1–26, 2012. URL <http://www.jstatsoft.org/v47/i12/>.
- Hemant Ishwaran and Lancelot F. James. Generalized weighted Chinese restaurant processes for species sampling mixture models. *Statistica Sinica*, 13(4):1211–1235, 2003.
- Ke Jiang, Brian Kulis, and Michael I. Jordan. Small-variance asymptotics for exponential family dirichlet process mixture models. In F. Pereira, C.J.C. Burges, L. Berton, and

- K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 3158–3166. Curran Associates, Inc., 2012. URL <http://papers.nips.cc/paper/4853-small-variance-asymptotics-for-exponential-family-dirichlet-process-mixture-models.pdf>.
- Brian Kulis and Michael I. Jordan. Revisiting k-means: New algorithms via bayesian nonparametrics. *CoRR*, abs/1111.0352, 2011. URL <http://arxiv.org/abs/1111.0352>.
- R. M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.
- Fernando A. Quintana and Pilar L. Iglesias. Bayesian clustering and product partition models. *Journal of the Royal Statistical Society, Series B: Statistical Methodology*, 65(2):557–574, 2003.
- R Core Team.** *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014. URL <http://www.R-project.org/>.
- Terrance Savitsky and Daniell Toth. Convergence of pseudo posterior distributions under informative sampling. 2015. URL <http://arxiv.org/abs/1507.07050>.
- Matthew S. Shotwell. Profdpn: An R package for MAP estimation in a class of conjugate product partition models. *Journal of Statistical Software*, 53(8):1–18, 2013.
- Matthew S. Shotwell and Elizabeth H. Slate. Bayesian outlier detection with Dirichlet process mixtures. *Bayesian Analysis*, 6(4):665–690, 2011.
- Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.
- Liaoning Wang and D. B. Dunson. Fast bayesian inference in dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 20:196–216, 2011.

Appendix A. Hierarchical Clustering Algorithm

Loop over algorithm blocks, A.2 and A.3 until convergence.

Algorithm A.1: INITIALIZE LOCAL AND GLOBAL CLUSTER OBJECTS

Input: Set of $n_j \times d$ data matrices, $\{\mathbf{X}^j\}_{j=1,\dots,J}$, where each holds $1 \times d$ observations for n_j units in local dataset, j . Set of $n_j \times 1$ vectors, $\{\mathbf{w}^j\}$ of sampling weights for n_j units in local dataset, j . λ_L , local cluster penalty parameter. λ_K , global cluster penalty parameter.

Output: $K \times d$, $\mathbf{M} = (\mu_1, \dots, \mu_K)$, a matrix of $1 \times d$ global cluster centers for $p = 1, \dots, K$ global clusters. A set of $n_j \times 1$ vectors, $\{s^j\}_{j=1,\dots,J}$, where $s^j \in \{1, 2, \dots, K\}$, linking units in each dataset, \mathbf{X}^j , to a global cluster, $p \in \{1, \dots, K\}$. The number of unique values in s^j , denoted by L_j , specifies the local partition structure for dataset, \mathbf{X}^j .

- 1 **Initialize:** Initialize number of global clusters, $K = 1$.
- 2 **for** $j \leftarrow 1$ **to** J **do**
 - 3 $L_j \times 1$, v^j records link of each local cluster, $c \in \{1, \dots, L_j\}$, to global cluster, $p \in \{1, \dots, K\}$. Initialize $L_j = 1$ and $v_{L_j}^j = 1$.
 - 4 $n_j \times 1$, \mathbf{z}^j , records link of each unit, i , in dataset, j , to a local cluster, $c \in \{1, \dots, L_j\}$. Initialize $z_i^j = 1$, $\forall i$.
 - 5 $n_j \times 1$, \mathbf{s}^j records link of each unit, i , in dataset, j to a global cluster, $p \in \{1, \dots, K\}$. Initialize $s_i^j \leftarrow v_{z_i^j}^j$, $\forall i$.
- 6 Set $d \times 1$ cluster center, $\mu_1 \leftarrow \left(\sum_{j=1}^J \sum_{i=1}^{n_j} \tilde{w}_i^j \mathbf{x}_i^j \right) / \left(\sum_{j=1}^J \sum_{i=1}^{n_j} \tilde{w}_i^j \right)$.
- 7 Compute energy,

$$e \leftarrow \sum_{p=1}^K \sum_{j=1}^J \sum_{i \in \{i: s_i^j = p\}} \tilde{w}_i^j \left\| \mathbf{X}_i^j - \mu_p \right\|^2 + \lambda_K K + \lambda_L L$$
, where $L = \sum_{j=1}^J L_j$.

45

Algorithm A.2: BUILD LOCAL AND GLOBAL CLUSTERS

1 Assignment of units to global clusters.

- 2 **for** $j \leftarrow 1$ **to** J **and** $i \leftarrow 1$ **to** n_j **do**
 - 3 Compute distance metric, $d_{ijp} = \tilde{w}_i^j \left\| \mathbf{X}_{i,1:d}^j - \mu_p \right\|^2$ for $p = 1, \dots, K$.
 - 4 For those global clusters, p , not linked to any local cluster in j , $\{p : v_p^j \neq p, \forall c \in \{1, \dots, L_j\}\}$,
 - 5 $d_{ijp} \leftarrow d_{ijp} + \lambda_L$ (since must add local cluster if assign i to p).
 - 6 **if** $\min_p d_{ijp} > \lambda_L + \lambda_K$ **then**
 - 7 Create new local cluster linked to new global cluster.
 - 8 $L_j \leftarrow L_j + 1$, $z_i^j = L_j$
 - 9 $K \leftarrow K + 1$, $\mu_K \leftarrow \mathbf{X}_i^j$, $v_{L_j}^j \leftarrow K$
 - 10 **else**
 - 11 Let $\hat{p} \leftarrow \operatorname{argmin}_p d_{ijp}$.
 - 12 **if** $v_i^j = \hat{p}$ for some $c \in \{1, \dots, L_j\}$ **then**
 - 13 | Assign unit i in j to local cluster c , $z_i^j \leftarrow c$ and $v_c^j = \hat{p}$.
 - 14 **else**
 - 15 | Create new local cluster for unit i and link to global cluster, \hat{p} .
 - 16 | $L_j \leftarrow L_j + 1$ and $z_i^j = L_j$ and $v_{L_j}^j = \hat{p}$.

17 Re-assignment of (all units in) local clusters to global clusters.

- 18 **for** local partitions, $j \leftarrow 1$ **to** J **and** local cluster, $c \leftarrow 1$ **to** L_j **do**
 - 19 Let $\mathbf{z}_c^j = \{i : \mathbf{z}^j = c\}$ and $1 \times d$, $\mu_{jc} = \left(\sum_{i \in \mathbf{z}_c^j} \tilde{w}_i^j \mathbf{x}_i^j \right) / \left(\sum_{i \in \mathbf{z}_c^j} \tilde{w}_i^j \right)$.
 - 20 Compute sum of local-to-global cluster distances:

$$d_{jcp} = \sum_{i \in \mathbf{z}_c^j} \tilde{w}_i^j \left\| \mathbf{X}_i^j - \mu_p \right\|^2$$
, for $p = 1, \dots, K$.
 - 21 **if** $\min_p d_{jcp} > \lambda_K + \sum_{i \in \mathbf{z}_c^j} \tilde{w}_i^j \left\| \mathbf{X}_i^j - \mu_{jc} \right\|^2$ **then**
 - 22 | Set $K \leftarrow K + 1$, $v_c^j = K$, and $\mu_K = \mu_{jc}$
 - 23 **else** $v_c^j \leftarrow \operatorname{argmin}_p d_{jcp}$
- 25 **Shed global clusters no longer assigned to any local clusters.**
- 26 **for** $p \leftarrow 1$ **to** K **global clusters do**
 - 27 **if** $v^j \neq p$, $\forall j \in \{1, \dots, J\}$ **then**
 - 28 | Recode global cluster labels in v^j such that $p' \leftarrow p - 1$, $\forall p' > p$
 - 29 Set $K \leftarrow K - 1$
 - 30 Delete cluster center for p , $\mathbf{M} \leftarrow \mathbf{M}_{-p,1:d}$.
 - 31 **for** local partitions, $j \leftarrow 1$ **to** J **and** units, $i \leftarrow 1$ **to** n_j **do**
 - 32 | $s_i^j \leftarrow v_{z_i^j}^j$.
- 33 **Re-compute global cluster centers**
- 34 **for** $p \leftarrow 1$ **to** K **global clusters do**
 - 35 **for** $j \leftarrow 1$ **to** J **datasets do**
 - 36 | Compute units in j assigned to global cluster p , $\mathbf{S}_{jp} = \{i : s_i^j = p\}$;
 - 37 | Compute $\mu_p = \left(\sum_{j=1}^J \sum_{i \in \mathbf{S}_{jp}} \mathbf{x}_i^j \tilde{w}_i^j \right) / \sum_{j=1}^J \sum_{i \in \mathbf{S}_{jp}} \tilde{w}_i^j$.

46

Algorithm A.3: MERGE GLOBAL CLUSTERS

```

1  Compute energy of current state,  $e \leftarrow \sum_{p=1}^K \sum_{j=1}^J \sum_{i \in (i: s_i^j = p)} \tilde{w}_i^j \left\| \mathbf{x}_i^j - \boldsymbol{\mu}_p \right\|^2 + \lambda_K K + \lambda_L L$ 
2  for  $p \leftarrow 2$  to  $K$  and  $p' \leftarrow 1$  to  $(p-1)$  do
3      Perform test merge for each pair of global clusters.
4      Set matrix of cluster centers for virtual step,  $\mathbf{M}^* \leftarrow \mathbf{M}$ .
5      for local partitions,  $j \leftarrow 1$  to  $J$  do
6          Let  $\mathbf{S}_{j,p'}^* = \{i : s_i^j = p'\}$ 
7          if  $|\mathbf{S}_{j,p'}^*| > 0$  then
8              [ There are some units in  $\mathbf{s}^j$  assigned to global cluster,  $p'$ , Re-assign units linked
9                  to  $p$  to  $p$ .  $\mathbf{S}_{j,p}^* = \{i : s_i^j = p \text{ or } s_i^j = p'\}$  ]
10             Compute merged cluster centers,  $\boldsymbol{\mu}_{p'}^* = \left( \sum_{j=1}^J \sum_{i \in \mathbf{S}_{j,p}^*} \mathbf{x}_i^j \tilde{w}_i^j \right) / \sum_{j=1}^J \sum_{i \in \mathbf{S}_{j,p}^*} \tilde{w}_i^j$ 
11             Set  $\mathbf{M}_{p',1:d}^* \leftarrow \mathbf{M}_{p,1:d}^*$ 
12             Compute number of local clusters shed if merge global cluster,  $(p', p)$ .
13             for local partitions,  $j \leftarrow 1$  to  $J$  do
14                 [ Local clusters linked to  $p', p$ :  $\mathbf{c}_{j,p'} = \{c : \mathbf{v}^j = p'\}$  and  $\mathbf{c}_{j,p} = \{c : \mathbf{v}^j = p\}$ ,
15                      $L_{j,p'} = |\mathbf{c}_{j,p'}|$  and  $L_{j,p} = |\mathbf{c}_{j,p}|$  ]
16             Reduced number of local clusters,  $L^* = L - \sum_{j=1}^J L_{j,p}$ , and global clusters,  $K^* \leftarrow K - 1$ .
17             Compute energy under the test merge,
18              $e^* \leftarrow \sum_{p=1}^K \sum_{j=1}^J \sum_{i \in \mathbf{S}_{j,p}^*} \tilde{w}_i^j \left\| \mathbf{x}_i^j - \boldsymbol{\mu}_{p'}^* \right\|^2 + \lambda_K K^* + \lambda_L L^*$ 
19             if  $e^* < e$  then
20                 Execute merge of global clusters,  $p$  and  $p'$ 
21                 for local partitions,  $j \leftarrow 1$  to  $J$  do
22                     if  $L_{j,p'} > 0$  then
23                         [ Cluster  $p'$  is linked to local partition,  $j$ .
24                             if  $L_{j,p} = 0$  then  $\mathbf{v}_{\mathbf{c}_{j,p'}}^j \leftarrow p$ 
25                             else
26                                 [ Reassign local clusters linked to  $p'$  to  $p$ .
27                                      $\mathbf{z}_{\mathbf{c}_{p'}^j}^j = \{i : \mathbf{z}^j \in \mathbf{c}_{j,p'}\}$  and  $\mathbf{z}_{\mathbf{c}_{p'}^j}^j \leftarrow \mathbf{c}_{j,p}$ .
28                                     Remove now empty local clusters linked to  $p'$ ,  $\mathbf{v}^j \leftarrow \mathbf{v}_{-\mathbf{c}_{p'}^j}^j$ .
29                                     Recode  $\mathbf{z}^j$  local cluster assignments so labels are contiguous by setting
30                                          $c \leftarrow c - 1, \forall c > \mathbf{c}_{j,p'}$ .
31                                     Recode global cluster labels in  $\mathbf{v}^j$  such that  $p \leftarrow p - 1, \forall p > p'$  ]
32                                 Remove global cluster center for  $p'$ .
33                                  $\mathbf{M}^* \leftarrow \mathbf{M}_{-p',1:d}^*$ 
34                                 for units,  $i \leftarrow 1$  to  $n_j$  do
35                                     [  $s_i^j \leftarrow \mathbf{v}_{z_i^j}^j$  ]
36                                  $\mathbf{M} \leftarrow \mathbf{M}^*$ 

```

Approximate Newton Methods for Policy Search in Markov Decision Processes

Thomas Furmston

*Department of Computer Science
University College London
London, WC1E 6BT*

T.FURMSTON@CS.UCL.AC.UK

Guy Lever

*Department of Computer Science
University College London
London, WC1E 6BT*

G.LEVER@CS.UCL.AC.UK

David Barber

*Department of Computer Science
University College London
London, WC1E 6BT*

D.BARBER@CS.UCL.AC.UK

Editor: Joelle Pineau

Abstract

Approximate Newton methods are standard optimization tools which aim to maintain the benefits of Newton's method, such as a fast rate of convergence, while alleviating its drawbacks, such as computationally expensive calculation or estimation of the inverse Hessian. In this work we investigate approximate Newton methods for policy optimization in Markov decision processes (MDPs). We first analyse the structure of the Hessian of the total expected reward, which is a standard objective function for MDPs. We show that, like the gradient, the Hessian exhibits useful structure in the context of MDPs and we use this analysis to motivate two Gauss-Newton methods for MDPs. Like the Gauss-Newton method for non-linear least squares, these methods drop certain terms in the Hessian. The approximate Hessians possess desirable properties, such as negative definiteness, and we demonstrate several important performance guarantees including guaranteed ascent directions, invariance to affine transformation of the parameter space and convergence guarantees. We finally provide a unifying perspective of key policy search algorithms, demonstrating that our second Gauss-Newton algorithm is closely related to both the EM algorithm and natural gradient ascent applied to MDPs, but performs significantly better in practice on a range of challenging domains.

Keywords: Markov decision processes, reinforcement learning, Newton method, function approximation

1. Introduction

Markov decision processes (MDPs) are the standard model for optimal control in a fully observable environment (Bertsekas, 2010). Strong empirical results have been obtained in numerous challenging real-world optimal control problems using the MDP framework. This includes problems of non-linear control (Stengel, 1993; Li and Todorov, 2004; Todorov and

Tassa, 2009; Deisenroth and Rasmussen, 2011; Rawlik et al., 2012; Spall and Cristion, 1998; Levine and Koltun, 2013b; Schulman et al., 2015; Heess et al., 2015; Lillicrap et al., 2016), robotic applications (Kober and Peters, 2011; Kohl and Stone, 2004; Vlassis et al., 2009), biological movement systems (Li, 2006), traffic management (Richter et al., 2007; Srinivasan et al., 2006), helicopter flight control (Abbeel et al., 2007), elevator scheduling (Crites and Barto, 1995) and numerous games, including chess (Veness et al., 2009), go (Silver et al., 2016; Gelly and Silver, 2008), backgammon (Tesauro, 1994) and Atari 2600 video games (Mnih et al., 2015; Schulman et al., 2015).

It is well known that the global optimum of a MDP with finite state and action sets can be obtained through methods based on dynamic programming, such as value iteration (Bellman, 1957) and policy iteration (Howard, 1960). However, these techniques are known to suffer from the curse of dimensionality, which makes them infeasible for many real-world problems of interest. As a result, most research in the reinforcement learning and control theory literature has focused on obtaining approximate or locally optimal solutions. There exists a broad spectrum of such techniques, including approximate dynamic programming methods (Bertsekas, 2010; Mnih et al., 2015), tree search methods (Russell and Norvig, 2009; Kocsis and Szepesvári, 2006; Browne et al., 2012; Silver et al., 2016), local trajectory-optimization techniques, such as differential dynamic programming (Jacobson and Mayne, 1970) and iLQG (Li and Todorov, 2006), and policy search methods (Williams, 1992; Baxter and Bartlett, 2001; Sutton et al., 2000; Marbach and Tsitsiklis, 2001; Kakade, 2002; Kober and Peters, 2011; Levine and Koltun, 2013b; Silver et al., 2014; Schulman et al., 2015; Heess et al., 2015; Lillicrap et al., 2016).

The focus of this paper is on policy search methods, which are a family of algorithms that have proven extremely popular in recent years, and which have numerous desirable properties that make them attractive in practice. Policy search algorithms are typically specialized applications of techniques from numerical optimization (Nocedal and Wright, 2006; Dempster et al., 1977). As such, the controller is given a differentiable parameterisation and an objective function is defined in terms of this parameterisation. Local information about the objective function, such as the gradient, is then used to update the parameters of the controller in an incremental manner until the algorithm converges to a local optimum of the objective function. There are several benefits to such an approach: the smooth updates of the control parameters endow these algorithms with very general convergence guarantees; as performance is improved at each iteration (or at least on average in stochastic policy search methods) these algorithms have good anytime performance properties; it is not necessary to approximate the value function, which is typically a difficult function to approximate—instant it is only necessary to approximate a low-dimensional projection of the value function, an observation which has led to the emergence of so called *actor-critic* methods (Konda and Tsitsiklis, 2003, 1999; Bhatnagar et al., 2008, 2009); policy search methods are easily extendable to models for optimal control in a partially observable environment, such as the Finite State Controllers (Meuleau et al., 1999; Toussaint et al., 2006).

In (stochastic) gradient ascent (Williams, 1992; Baxter and Bartlett, 2001; Sutton et al., 2000) the control parameters are updated by moving in the direction of the gradient of an objective function. While gradient ascent has enjoyed some success, it suffers from serious issues that can hinder its performance: specifically, it is not scale invariant (Nocedal and

Wright, 2006) and the search direction is often poorly-scaled, i.e., the variation of the objective function differs dramatically along the different components of the gradient. Poor scaling of the gradient leads to a poor rate of convergence (Noceval and Wright, 2006). It also makes the construction of a good step size sequence a difficult problem, which is an important issue in stochastic methods.¹ Poor scaling is a well known problem with gradient ascent and alternative numerical optimization techniques have been considered in the policy search literature. Two approaches that have proven to be particularly popular are expectation maximization (Dempster et al., 1977) and natural gradient ascent (Amari, 1997, 1998; Amari et al., 1992), which have both been successfully applied to various challenging MDPs (see the works of Dayan and Hinton (1997); Kober and Peters (2009); Toussaint et al. (2011); Levine and Kolthun (2013a) and Kakade (2002); Bagnell and Schneider (2003) respectively).

An avenue of research that has received less attention is the application of Newton’s method to Markov decision processes. Although such an extension of the *GPOMDP* algorithm is provided in the work of Baxter and Bartlett (2001), they give no empirical results in either that article or the accompanying paper of empirical comparisons (Baxter et al., 2001). There has since been only a limited amount of research into using the second order information contained in the Hessian during the parameter update. To the best of our knowledge only two attempts have been made: in the work of Schraudolph et al. (2006) an on-line estimate of a Hessian-vector product is used to adapt the step size sequence in an on-line manner; in the work of Ngo et al. (2011), Bayesian policy gradient methods (Ghahramanah and Engel, 2007) are extended to Newton’s method. There are several reasons for this lack of interest. Firstly, in many problems the construction and inversion of the Hessian is too computationally expensive to be feasible. Additionally, the objective function of a MDP is typically not concave, and so the Hessian is not guaranteed to be negative-definite. As a result, the search direction of Newton’s method may not be an ascent direction, and hence a parameter update could actually lower the objective. Additionally, the variance of sample-based estimators of the Hessian will be larger than that of estimators of the gradient. This is an important point because the variance of gradient estimates can be a problematic issue and various methods, such as baselines (Weaver and Tao, 2001; Greensmith et al., 2004), exist to reduce the variance.

Many of these problems are not particular to Markov decision processes, but are general longstanding issues that plague Newton’s method. Various methods have been developed in the optimization literature to alleviate these issues, while also maintaining desirable properties of Newton’s method. For instance, quasi-Newton methods were designed to efficiently mimic Newton’s method using only evaluations of the gradient obtained during previous iterations of the algorithm. These methods have low computational costs, a super-linear rate of convergence and have proven to be extremely effective in practice. See the work of Nocedal and Wright (2006) for an introduction to quasi-Newton methods. Alternatively, the well-known Gauss-Newton method is a popular approach that aims to efficiently mimic Newton’s method. The Gauss-Newton method is particular to non-linear least squares objective functions, for which the Hessian has a particular structure. Due to this structure there exist certain terms in the Hessian that can be used as a useful proxy for the Hessian

itself, with the resulting algorithm having various desirable properties. For instance, the preconditioning matrix used in the Gauss-Newton method is guaranteed to be positive-semidefinite, so that the non-linear least squares objective is guaranteed not to increase for a sufficiently small step size.

While a straightforward application of quasi-Newton methods will not typically be possible for MDPs,² in this paper we consider whether an analogue to the Gauss-Newton method exists, so that the benefits of such methods can be applied to MDPs. The specific contributions are as follows:

- In Section 3, we present an analysis of the Hessian of the total expected reward, which is a standard objective function for MDPs. Our starting point is a derivation of the Hessian for the total expected reward (Theorem 3) and we analyse the behavior of individual terms of the Hessian to provide insight into constructing efficient approximate Newton methods for policy optimization. In particular, we provide conditions under which certain terms become negligible near local optima.

- Motivated by this analysis, in Section 4 we provide two Gauss-Newton type methods for policy optimization in MDPs. These methods retain certain terms of our Hessian decomposition in the preconditioner in a gradient-based policy search algorithm. The first method discards terms which are difficult to approximate and which, for appropriately selected classes of controller, will become negligible near local optima. The second method further discards an additional term which is not guaranteed to be negative semi-definite. We provide an analysis of our Gauss-Newton methods and give several important performance guarantees for the second Gauss-Newton method: We demonstrate that the preconditioning matrix is negative-semidefinite when the controller is log-concave in the control parameters (detailing some widely used controllers for which this condition holds) guaranteeing that the search direction is an ascent direction; We show that the method is invariant to affine transformations of the parameter space and thus does not suffer the significant drawback of gradient ascent. We provide a convergence analysis, demonstrating linear convergence to local optima, in terms of the step size of the update.

Our methods apply to finite and continuous state and action sets. For simplicity of exposition we have presented results for the finite case to avoid measurability considerations. Some of our experiments have continuous state and action sets. Similarly our method is suitable for unknown transition dynamics, but can also be trivially used in a model-based approach with a known or estimated dynamics model.

- In Section 5 we present a unifying perspective for several policy search methods. In particular we relate the search direction of our second Gauss-Newton algorithm to that of expectation maximization (which provides new insights into the latter algorithm when used for policy search), and we also discuss its relationship to the natural gradient algorithm.

2. In quasi-Newton methods, to ensure an increase in the objective function, it is necessary to satisfy the secant condition (Nocedal and Wright, 2006). This condition is satisfied when the objective is concave/convex or the strong Wolfe conditions are met during a line search. For this reason, stochastic applications of quasi-Newton methods has been restricted to convex/concave objective functions (Schraudolph et al., 2007).

1. This is because line search techniques lose much of their desirability in stochastic numerical optimization algorithms, due to variance in the evaluations.

- In Section 6 we present experiments demonstrating state-of-the-art performance on challenging domains including Tetris and robotic arm applications.

2. Preliminaries and Background

In Section 2.1 we introduce Markov decision processes, along with some standard terminology relating to these models that will be required throughout the paper. In Section 2.2 we introduce policy search methods and provide an overview of the literature.

2.1 Markov Decision Processes

In a Markov decision process an *agent*, or *controller*, sequentially interacts with an environment by selecting actions (based on the current state of the environment) after which the system transitions to a new state (often in a stochastic manner) and the agent receives a scalar reward (dependent upon the selected action and the state of the environment). The optimality of an agent’s behavior is measured in terms of the total (discounted) reward the agent can expect to receive, so that optimal control is obtained when this quantity is maximized.

Formally a MDP is described by the tuple $(\mathcal{S}, \mathcal{A}, D, P, R)$, in which \mathcal{S} and \mathcal{A} are sets, known respectively as the state and action space, D is the initial state distribution, which is a distribution over the state space, P encodes the transition dynamics using a set of conditional distributions over the state space, $\{P(\cdot|s, a)\}_{(s, a) \in \mathcal{S} \times \mathcal{A}}$, and $R : \mathcal{S} \times \mathcal{A} \rightarrow [0, R_{\max}]$ is the (possibly stochastic³) reward function, which is assumed to be bounded and non-negative.⁴ We use the notation s_t and a_t to denote the random variable of the state and action of the t^{th} time step, $t \in \mathbb{N}$, respectively. The state at the initial time step is determined by the initial state distribution, $s_1 \sim D(\cdot)$. At any given time step, $t \in \mathbb{N}$, and given the state of the environment, the agent selects an action, $a_t \sim \pi(\cdot|s_t)$, according to the *policy* π . The next state is determined according to the transition dynamics, $s_{t+1} \sim P(\cdot|a_t, s_t)$. At each time step the agent receives a scalar reward, which is determined by the reward function.

In this paper we consider the total expected reward, which is a standard objective function in the reinforcement learning literature (Bertsekas, 2010). We shall consider the infinite horizon discounted reward framework. In this framework there is a discount factor, $\gamma \in [0, 1)$, and the objective function takes the form,

$$U(\pi) := \sum_{t=1}^{\infty} \mathbb{E}_{s_t, a_t \sim p_t} \left[\gamma^{t-1} R(s_t, a_t); \pi \right]. \quad (1)$$

3. For notational convenience we shall consider a deterministic reward function in this paper. The extension to the case of a stochastic reward function can be done by considering the expectation over the reward function where appropriate.
4. Given either an episodic finite horizon or a discounted infinite horizon MDP in which the reward function is bounded but not necessarily non-negative, one can construct an auxiliary MDP that has a bounded non-negative reward function and has the same optimal policies as the original MDP. This can be achieved, for example, by setting the reward function of the auxiliary MDP as $R_{\text{auxiliary}}(s, a) = R(s, a) - \min_{(s, a) \in \mathcal{S} \times \mathcal{A}} R(s, a)$, which only requires knowledge of the lower bound of the reward function in the original MDP. The same is not true, however, of absorbing state MDPs with a discount factor of 1.

We use the semi-colon to identify parameters of the distribution, rather than conditioning variables. The distribution of s_t and a_t , which we denote by p_t , is given by the marginal at time t of the joint distribution over $(s_{1:t}, a_{1:t})$, with $s_{1:t} = (s_1, s_2, \dots, s_t)$, $a_{1:t} = (a_1, a_2, \dots, a_t)$, which is given by,

$$p(s_{1:t}, a_{1:t}; \pi) := \pi(a_t|s_t) \left\{ \prod_{\tau=1}^{t-1} P(s_{\tau+1}|s_{\tau}, a_{\tau}) \times \pi(a_{\tau}|s_{\tau}) \right\} D(s_1). \quad (2)$$

We use the notation $\xi_t := (s_1, a_1, s_2, a_2, \dots, s_t, a_t)$ to denote trajectories through the state-action space of length, $t \in \mathbb{N}$. We use ξ to denote trajectories that are of infinite length, and use Ξ to denote the space of all such trajectories. Given a trajectory, $\xi \in \Xi$, we use the notation $\bar{R}(\xi)$ to denote the total discounted reward of the trajectory, so that $\bar{R}(\xi) = \sum_{t=1}^{\infty} \gamma^{t-1} R(s_t, a_t)$. Similarly, we use the notation $p(\xi; \pi)$ to denote the probability of generating the trajectory ξ under the policy π .

We now introduce several functions that are of central importance. The state value function w.r.t. policy π is defined as the total expected future reward given the current state,

$$V_{\pi}(s) := \sum_{s_1, a_1 \sim p_1}^{\infty} \left[\gamma^{t-1} R(s_t, a_t) \mid s_1 = s; \pi \right].$$

It can be seen that, $U(\pi) = \mathbb{E}_{s \sim D} [V_{\pi}(s)]$. The state value function can also be written as the solution of the following fixed-point equation, $V_{\pi}(s) = \mathbb{E}_{a \sim \pi(\cdot|s)} \left[R(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s, a)} [V_{\pi}(s')] \right]$, which is known as the Bellman equation (Bertsekas, 2010). The state-action value function w.r.t. policy π is given by,

$$Q_{\pi}(s, a) := R(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s, a)} [V_{\pi}(s')], \quad (3)$$

and gives the value of performing an action, in a given state, and then following the policy. Note that, $V_{\pi}(s) = \sum_{a \in \mathcal{A}} \pi(a|s) Q_{\pi}(s, a)$. Finally, the advantage function, $A_{\pi}(s, a) := Q_{\pi}(s, a) - V_{\pi}(s)$, gives the relative advantage of an action in relation to the other actions available in that state. It can be seen that, $\sum_{a \in \mathcal{A}} \pi(a|s) A_{\pi}(s, a) = 0$, for each $s \in \mathcal{S}$.

2.2 Policy Search Methods

In policy search methods the policy is given some differentiable parametric form, denoted $\pi(a|s; \mathbf{w})$, with policy parameters, $\mathbf{w} \in \mathcal{W} \subset \mathbb{R}^n$, $n \in \mathbb{N}$. (We also use the notation, $\pi_{\mathbf{w}}(a|s) \equiv \pi(a|s; \mathbf{w})$, where appropriate.) Local information, such as the gradient of the objective function, is then used to update the policy in an incremental manner until the algorithm converges to a local optimum of the objective function. We overload notation and write the objective function directly in terms of the parameter vector, i.e.,

$$U(\mathbf{w}) := U(\pi_{\mathbf{w}}), \quad \forall \mathbf{w} \in \mathcal{W}, \quad (4)$$

while the trajectory distribution is written in the form $p(a_{1:t}, s_{1:t}; \mathbf{w}) = p(a_{1:t}, s_{1:t}; \pi_{\mathbf{w}})$. Similarly, $V(s; \mathbf{w})$, $Q(s, a; \mathbf{w})$ and $A(s, a; \mathbf{w})$ denote respectively the state value function,

state-action value function and the advantage function in terms of the parameter vector \mathbf{w} . We introduce the function,

$$p_\gamma(s, a; \mathbf{w}) := \sum_{t=1}^{\infty} \gamma^{t-1} p_t(s_t = s, a_t = a; \mathbf{w}). \quad (5)$$

Note that $\sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} p_\gamma(s, a; \mathbf{w}) = (1 - \gamma)^{-1}$. Additionally, the objective function can be written in the form $U(\mathbf{w}) = \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} p_\gamma(s, a; \mathbf{w}) R(s, a)$.

We shall consider two forms of policy search algorithm in this paper: gradient-based optimization methods and methods based on iteratively optimizing a lower-bound on the objective function. In gradient-based methods the update of the policy parameters takes the form,

$$\mathbf{w}^{\text{new}} = \mathbf{w} + \alpha \mathcal{M}(\mathbf{w}) \frac{\partial}{\partial \mathbf{w}} U(\mathbf{w}), \quad (6)$$

where $\alpha \in \mathbb{R}^+$ is a step size parameter and $\mathcal{M}(\mathbf{w})$ is some preconditioning matrix that possibly depends on $\mathbf{w} \in \mathcal{W}$. If U is smooth, $\mathcal{M}(\mathbf{w})$ is positive-definite and α is sufficiently small then such an update will increase the total expected reward. If the preconditioning matrix is always positive-definite, the step size sequence is appropriately selected and U is Lipschitz, which is the case when $\|\frac{\partial}{\partial \mathbf{w}} \log \pi(a|s; \mathbf{w})\|_2$ is uniformly bounded by $M \in \mathbb{R}$ for all $\mathbf{w} \in \mathcal{W}$ and $(a, s) \in \mathcal{A} \times \mathcal{S}$, then iteratively updating the policy parameters according to (6) will result in the policy parameters converging to a local optimum of U . This generic gradient-based policy search algorithm is given in Algorithm 1. Gradient-based methods vary in the form of the preconditioning matrix used in the parameter update. The choice of the preconditioning matrix determines various aspects of the resulting algorithm, such as the computational complexity; the rate at which the algorithm converges to a local optimum and invariance properties of the parameter update. Typically the gradient, $\frac{\partial}{\partial \mathbf{w}} U(\mathbf{w})$, and the preconditioner, $\mathcal{M}(\mathbf{w})$, will not be known exactly and must be approximated by collecting data from the system. In the context of reinforcement learning, the expectation maximization (EM) algorithm searches for the optimal policy by iteratively optimizing a lower bound on the objective function. While EM does not have an update of the form given in (6) we shall see in Section 5.2 that the algorithm is closely related to such an update. We now review specific policy search methods.

2.2.1 GRADIENT ASCENT

Gradient ascent corresponds to the choice $\mathcal{M}(\mathbf{w}) = I_n$, where I_n denotes the $n \times n$ identity matrix, so that the parameter update takes the form:

Policy search update using gradient ascent

$$\mathbf{w}^{\text{new}} = \mathbf{w} + \alpha \frac{\partial}{\partial \mathbf{w}} U(\mathbf{w}). \quad (7)$$

The gradient, $\frac{\partial}{\partial \mathbf{w}} U(\mathbf{w})$, can be written in a relatively simple form using the following theorem:

<p>Algorithm 1: Generic gradient-based policy search algorithm</p> <p>Input: Initial vector of policy parameters, $\mathbf{w}_0 \in \mathcal{W}$, and a step size sequence, $(\alpha_k)_{k=0}^{\infty}$ with $\alpha_k \in \mathbb{R}^+$ for $k \in \mathbb{N}$.</p> <p>Set iteration counter, $k \leftarrow 0$.</p> <p>repeat</p> <p> Either calculate or estimate the gradient of the objective, $\frac{\partial}{\partial \mathbf{w}} U(\mathbf{w}) _{\mathbf{w}=\mathbf{w}_k}$, and the preconditioner, $\mathcal{M}(\mathbf{w}_k)$, at the current point in the parameter space.</p> <p> Update policy parameters, $\mathbf{w}_{k+1} = \mathbf{w}_k + \alpha_k \mathcal{M}(\mathbf{w}_k) \frac{\partial}{\partial \mathbf{w}} U(\mathbf{w}) _{\mathbf{w}=\mathbf{w}_k}$.</p> <p> Update iteration counter, $k \leftarrow k + 1$.</p> <p>until Convergence of the policy parameters;</p> <p>return \mathbf{w}_k</p>

Theorem 1 (Policy Gradient Theorem (Sutton et al., 2000)). *Suppose we are given a Markov decision process with objective (1) and Markovian trajectory distribution (2). For any given parameter vector, $\mathbf{w} \in \mathcal{W}$, the gradient of (4) takes the form,*

$$\frac{\partial}{\partial \mathbf{w}} U(\mathbf{w}) = \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} p_\gamma(s, a; \mathbf{w}) Q(s, a; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a|s; \mathbf{w}). \quad (8)$$

Proof. This is a well-known result that can be found in Sutton et al. (2000). A derivation of (8) is provided in Section A.1 in the Appendix. \square

It is not possible to calculate the gradient exactly for many real-world MDPs of interest. For instance, in discrete domains the size of the state-action space may be too large for enumeration over these sets to be feasible. Alternatively, in continuous domains the presence of non-linearities in the transition dynamics makes the calculation of the occupancy marginals an intractable problem. Additionally, it can be the case that P and R are unknown in practice. In such cases it can be preferable to directly estimate the gradient using samples obtained from the environment, rather than building a model of the MDP. Various techniques have been proposed in the literature to estimate the gradient, including the method of finite-differences (Kiefer and Wolfowitz, 1952; Kohl and Stone, 2004; Tadrake and Zhang, 2005), simultaneous perturbation methods (Spall, 1992; Spall and Criston, 1998; Srinivasan et al., 2006) and likelihood-ratio methods (Glynn, 1986, 1990; Williams, 1992; Baxter and Bartlett, 2001; Konda and Tsitsiklis, 2003, 1999; Sutton et al., 2000; Bhatnagar et al., 2009; Kober and Peters, 2011). Likelihood-ratio methods, which originated in the statistics literature and were later applied to MDPs, are now the prominent method for estimating the gradient. There are numerous such methods in the literature, including Monte-Carlo methods (Williams, 1992; Baxter and Bartlett, 2001) and actor-critic methods (Konda and Tsitsiklis, 2003, 1999; Sutton et al., 2000; Bhatnagar et al., 2009; Kober and Peters, 2011). Gradient ascent is known to perform poorly on objective functions that are poorly-scaled, that is, if changes to some parameters produce much larger variations to the function than changes in other parameters. In this case gradient ascent zig-zags along the ridges of the

objective in the parameter space (see e.g., the work of Nocedal and Wright, 2006). It can be difficult to gauge an appropriate scale for the steps sizes in poorly-scaled problems and the robustness of optimization algorithms to poor scaling is of significant practical importance.

2.2.2 NATURAL GRADIENT ASCENT

Natural gradient ascent techniques originated in the neural network and blind source separation literature (Amari, 1997, 1998; Amari et al., 1996, 1992), and were introduced into the policy search literature in the work of Kakade (2002). To address the issue of poor scaling, natural gradient methods take the perspective that the parameter space should be viewed with a manifold structure in which the distance between two points on the manifold captures the discrepancy between the distribution over trajectories parameterized by the two corresponding parameter vectors. In natural gradient ascent $\mathcal{M}(\mathbf{w}) = G^{-1}(\mathbf{w})$ in (6), with $G(\mathbf{w})$ a suitable metric tensor for the manifold, so that the parameter update takes the form:

Policy search update using natural gradient ascent

$$\mathbf{w}^{\text{new}} = \mathbf{w} + \alpha G^{-1}(\mathbf{w}) \frac{\partial}{\partial \mathbf{w}} U(\mathbf{w}). \quad (9)$$

In the case of Markov decision processes a standard choice for $G(\mathbf{w})$ is the Fisher information matrix of the policy distribution, averaged over the state distribution, which takes the form,

$$G(\mathbf{w}) = \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} p_\gamma(s, a; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a|s; \mathbf{w}) \frac{\partial^\top}{\partial \mathbf{w}} \log \pi(a|s; \mathbf{w}). \quad (10)$$

It was shown in the work of Bagnell and Schneider (2003) that (10) corresponds to the Fisher information matrix of the distribution over trajectories in the state-action space, given π . The use of the Fisher information matrix can be viewed as imposing a local norm on the parameter space which is derived from a second order approximation to the KL-divergence between induced trajectory distributions (Coolen et al., 2005). For brevity we refer to this choice of $G(\mathbf{w})$ as *the* natural gradient algorithm. When the policy distribution satisfies the Fisher regularity conditions (see Lehmann and Casella, 1998, Lemma 5.3) there is an alternate, equivalent, form of the Fisher information matrix given by,

$$G(\mathbf{w}) = - \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} p_\gamma(s, a; \mathbf{w}) \frac{\partial^2}{\partial \mathbf{w}^2} \log \pi(a|s; \mathbf{w}). \quad (11)$$

Equation (11) can be derived by differentiating the identity, $\sum_{a \in \mathcal{A}} \pi(a|s; \mathbf{w}) \frac{\partial}{\partial w_i} \log \pi(a|s; \mathbf{w}) = \sum_{a \in \mathcal{A}} \frac{\partial}{\partial w_i} \pi(a|s; \mathbf{w}) = 0$, with respect to w_j , and taking expectation over $s \in \mathcal{S}$. Other metric tensors have also been considered in the policy search literature, such as in the work of Morimura et al. (2008).

There are several desirable properties of the natural gradient approach: the Fisher information matrix is always positive-semidefinite, regardless of the policy parameterisation;

the search direction is invariant to the parameterisation of the policy, (Bagnell and Schneider, 2003; Peters and Schaal, 2008). Additionally, the natural gradient update direction can be obtained by regressing the state-action value function, or the advantage function, using a compatible function approximator (Sutton et al., 2000), and minimizing a square loss weighted by the (discounted) trajectory distribution (Kakade, 2002). Furthermore, natural gradient ascent has been shown to perform well in some difficult MDP environments, including Tetris (Kakade, 2002) and several challenging robotics problems (Peters and Schaal, 2008). However, theoretically, the rate of convergence of natural gradient ascent is the same as gradient ascent, i.e., linear, although, it has been noted to be substantially faster in practice.

2.2.3 EXPECTATION MAXIMIZATION

An alternative optimization procedure that has been the focus of much research in the planning and reinforcement learning communities is the EM-algorithm (Dayan and Hinton, 1997; Toussaint et al., 2006, 2011; Kober and Peters, 2009, 2011; Hoffman et al., 2009; Furmston and Barber, 2009, 2010; Levine and Koltun, 2013a). The EM-algorithm is a powerful optimization technique popular in the statistics and machine learning community (see e.g., the works of Dempster et al., 1977; Little and Rubin, 2002; Neal and Hinton, 1999) that has been successfully applied to a large number of problems. See the work of Barber (2012) for a general overview of some of the applications of the algorithm in the machine learning literature. Among the strengths of the algorithm are its guarantee of increasing the objective function at each iteration, its often simple update equations, and its generalization to highly intractable models through variational Bayes approximations (Saul et al., 1996).

Given the advantages of the EM-algorithm it is natural to extend the algorithm to the MDP framework. Several derivations of the EM-algorithm for MDPs exist (Kober and Peters, 2011; Toussaint et al., 2011). For reference, we state the lower-bound upon which the algorithm is based in the following theorem. The proof is based on an application of Jensen's inequality and can be found in the work of Kober and Peters (2011).

Theorem 2. *Suppose we are given a Markov decision process with objective (1) and Markovian trajectory distribution (2). Given any distribution, q , over the space of trajectories, Ξ , then the following bound holds,*

$$\log U(\mathbf{w}) \geq H(q(\xi)) + \mathbb{E}_{\xi \sim q(\cdot)} \left[\log(p(\xi; \mathbf{w}) \bar{R}(\xi)) \right], \quad \forall \mathbf{w} \in \mathcal{W}. \quad (12)$$

The distribution, q , in Theorem 2 is often referred to as the variational distribution. An EM-algorithm is obtained through coordinate-wise optimization of (12) with respect to the variational distribution (the E-step) and the policy parameters (the M-step). In the E-step the lower-bound is optimized when $q(\xi) \propto p(\xi; \mathbf{w}') \bar{R}(\xi)$, in which \mathbf{w}' are the current policy parameters. In the M-step the lower-bound is optimized with respect to \mathbf{w} , which, given $q(\xi) \propto p(\xi; \mathbf{w}') \bar{R}(\xi)$ and the Markovian structure of $\log p(\xi; \mathbf{w})$, is equivalent to optimizing the function,

$$\mathcal{Q}(\mathbf{w}, \mathbf{w}') = \sum_{(s, a) \in \mathcal{S} \times \mathcal{A}} p_\gamma(s, a; \mathbf{w}') Q(s, a; \mathbf{w}') \log \pi(a|s; \mathbf{w}), \quad (13)$$

with respect to the first parameter, \mathbf{w} . The E-step and M-step are iterated in this manner until the policy parameters converge to a local optimum of the objective function.

Alternative bound optimisation techniques exist in the policy search literature. For instance, the trust region policy optimisation algorithm (Schulman et al., 2015) is based on a generalisation of a lower-bound of the total expected reward given in the work of Kakade and Langford (2002). While the lower-bound is intractable in large-scale MDPs, the introduction of several heuristics enables the authors to obtain strong empirical results in both non-linear control problems and Atari games.

3. The Hessian of the Total Expected Discounted Return

In this section we provide an analysis of the Hessian of the total expected reward of a MDP. This analysis will then be used in Section 4 to propose Gauss-Newton type methods for MDPs. In Section 3.1 we provide a novel representation of the Hessian of the total expected reward, in Section 3.2 we detail the definiteness properties of certain terms in the Hessian and in Section 3.3 we analyse the behaviour of individual terms of the Hessian in the vicinity of a local optimum.

3.1 The Policy Hessian Theorem

There is a standard expansion of the Hessian of the total expected reward in the policy search literature (Baxter and Bartlett, 2001; Kakade, 2001, 2002) that, as with the gradient, takes a relatively simple form. This is summarized in the following result.

Theorem 3 (Policy Hessian Theorem). *Suppose we are given a Markov decision process with objective (1) and Markovian trajectory distribution (2). For any given parameter vector, $\mathbf{w} \in \mathcal{W}$, the Hessian of (4) takes the form,*

$$\mathcal{H}(\mathbf{w}) = \mathcal{H}_1(\mathbf{w}) + \mathcal{H}_2(\mathbf{w}) + \mathcal{H}_{12}(\mathbf{w}) + \mathcal{H}_{12}^\top(\mathbf{w}), \quad (14)$$

in which the matrices $\mathcal{H}_1(\mathbf{w})$, $\mathcal{H}_2(\mathbf{w})$ and $\mathcal{H}_{12}(\mathbf{w})$ can be written in the form,

$$\mathcal{H}_1(\mathbf{w}) := \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} p_\gamma(s, a; \mathbf{w}) Q(s, a; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a|s; \mathbf{w}) \frac{\partial^\top}{\partial \mathbf{w}} \log \pi(a|s; \mathbf{w}), \quad (15)$$

$$\mathcal{H}_2(\mathbf{w}) := \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} p_\gamma(s, a; \mathbf{w}) Q(s, a; \mathbf{w}) \frac{\partial^2}{\partial \mathbf{w}^2} \log \pi(a|s; \mathbf{w}), \quad (16)$$

$$\mathcal{H}_{12}(\mathbf{w}) := \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} p_\gamma(s, a; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a|s; \mathbf{w}) \frac{\partial^\top}{\partial \mathbf{w}} Q(s, a; \mathbf{w}). \quad (17)$$

Proof. A derivation for a sample-based estimator of the Hessian can be found in Baxter and Bartlett (2001). For ease of reference a derivation of (14) is provided in Section A.1 in the Appendix. \square

We remark that $\mathcal{H}_1(\mathbf{w})$ and $\mathcal{H}_2(\mathbf{w})$ are relatively simple to estimate, in the same manner as estimating the policy gradient. The term $\mathcal{H}_{12}(\mathbf{w})$ is more difficult to estimate since it

contains terms involving the gradient, $\frac{\partial^\top}{\partial \mathbf{w}} Q(s, a; \mathbf{w})$, resulting in a double sum over state-actions.

Below we will present a novel form for the Hessian of the total expected reward, with attention given to the term $\mathcal{H}_1(\mathbf{w}) + \mathcal{H}_2(\mathbf{w})$ in (14), which will require the following notion of a parameterisation with constant curvature.

Definition 1. *A policy parameterisation is said to have constant curvature with respect to the action space, if for each $(s, a) \in \mathcal{S} \times \mathcal{A}$ the Hessian of the log-policy, $\frac{\partial^2}{\partial \mathbf{w}^2} \log \pi(a|s; \mathbf{w})$, does not depend upon the action, i.e.,*

$$\frac{\partial^2}{\partial \mathbf{w}^2} \log \pi(a|s; \mathbf{w}) = \frac{\partial^2}{\partial \mathbf{w}^2} \log \pi(a'|s; \mathbf{w}), \quad \forall a, a' \in \mathcal{A}.$$

A common class of policy which satisfies the property of Definition 1 is, $\pi(a|s; \mathbf{w}) \propto \exp(\mathbf{w}^\top \phi(a, s))$, in which $\phi(a, s)$ is a vector of features that depends on the state-action pair, $(a, s) \in \mathcal{A} \times \mathcal{S}$. Under this parameterisation,

$$\frac{\partial^2}{\partial \mathbf{w}^2} \log \pi(a|s; \mathbf{w}) = -\text{COV}_{a' \sim \pi(\cdot|s; \mathbf{w})}(\phi(a', s), \phi(a', s)),$$

which does not depend on, $a \in \mathcal{A}$. In the case when the action space is continuous, then the policy parameterisation $\pi(a|s; \mathbf{w}; \Sigma) \propto \exp(-\frac{1}{2}(a - \mathbf{w}^\top \phi(s))^\top \Sigma^{-1}(a - \mathbf{w}^\top \phi(s)))$, in which $\phi : \mathcal{S} \rightarrow \mathbb{R}^n$ is a given feature map, satisfies the properties of Definition 1 with respect to the mean parameters, $\mathbf{w} \in \mathcal{W}$.

We now present a novel decomposition of the Hessian for Markov decision processes.

Theorem 4. *Suppose we are given a Markov decision process with objective (1) and Markovian trajectory distribution (2). For any given parameter vector, $\mathbf{w} \in \mathcal{W}$, the Hessian of (4) takes the form,*

$$\mathcal{H}(\mathbf{w}) = \mathcal{A}_1(\mathbf{w}) + \mathcal{A}_2(\mathbf{w}) + \mathcal{H}_{12}(\mathbf{w}) + \mathcal{H}_{12}^\top(\mathbf{w}), \quad (18)$$

with,

$$\mathcal{A}_1(\mathbf{w}) := \sum_{(s, a) \in \mathcal{S} \times \mathcal{A}} p_\gamma(s, a; \mathbf{w}) A(s, a; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a|s; \mathbf{w}) \frac{\partial^\top}{\partial \mathbf{w}} \log \pi(a|s; \mathbf{w}),$$

$$\mathcal{A}_2(\mathbf{w}) := \sum_{(s, a) \in \mathcal{S} \times \mathcal{A}} p_\gamma(s, a; \mathbf{w}) A(s, a; \mathbf{w}) \frac{\partial^2}{\partial \mathbf{w}^2} \log \pi(a|s; \mathbf{w}).$$

When the policy parameterization has constant curvature with respect to the action space, then the Hessian takes the form,

$$\mathcal{H}(\mathbf{w}) = \mathcal{A}_1(\mathbf{w}) + \mathcal{H}_{12}(\mathbf{w}) + \mathcal{H}_{12}^\top(\mathbf{w}). \quad (19)$$

Proof. See Section A.2 in the Appendix. \square

We now present an analysis of the terms of the policy Hessian, simplifying the expansion and demonstrating conditions under which certain terms disappear. The analysis will be used to motivate our Gauss-Newton methods in Section 4.

3.2 Analysis of the Policy Hessian

An interesting comparison can be made between the expansions (14) and (18, 19) in terms of the definiteness properties of the component matrices. As the state-action value function is non-negative over the entire state-action space, it can be seen that $\mathcal{H}_1(\mathbf{w})$ is positive-semidefinite for all $\mathbf{w} \in \mathcal{W}$. Similarly, it can be shown that under certain common policy parameterisations $\mathcal{H}_2(\mathbf{w})$ is negative-semidefinite over the entire parameter space. This is summarized in the following theorem.

Theorem 5. *The matrix $\mathcal{H}_2(\mathbf{w})$ is negative-semidefinite for all $\mathbf{w} \in \mathcal{W}$ if: 1) the policy is log-concave with respect to the policy parameters; or 2) the policy parameterisation has constant curvature with respect to the action space.*

Proof. See Section A.3 in the Appendix. \square

It can be seen, therefore, that when the policy parameterisation satisfies the properties of Theorem 5 the expansion (14) gives $\mathcal{H}(\mathbf{w})$ in terms of a positive-semidefinite term, $\mathcal{H}_1(\mathbf{w})$, a negative-semidefinite term, $\mathcal{H}_2(\mathbf{w})$, and a remainder term, $\mathcal{H}_{12}(\mathbf{w}) + \mathcal{H}_{12}^\top(\mathbf{w})$. In Section 3.3 we shall show that this remainder term becomes negligible around a local optimum when given a sufficiently rich policy parameterisation, in a sense that we introduce in Definition 2. In contrast to the state-action value function, the advantage function takes both positive and negative values over the state-action space. As a result, the matrices $\mathcal{A}_1(\mathbf{w})$ and $\mathcal{A}_2(\mathbf{w})$ in (18, 19) can be indefinite over parts of the parameter space.

3.3 Analysis in Vicinity of a Local Optimum

In this section we consider the term $\mathcal{H}_{12}(\mathbf{w}) + \mathcal{H}_{12}^\top(\mathbf{w})$, which is both difficult to estimate and not guaranteed to be negative definite. In particular, we shall consider the conditions under which this term becomes either negligible or vanishes completely at a local optimum. We start by noting that,

$$\begin{aligned} \mathcal{H}_{12}(\mathbf{w}) &= \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} p_\gamma(s,a;\mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a|s;\mathbf{w}) \frac{\partial^\top}{\partial \mathbf{w}} \left(R(s,a) + \gamma \sum_{s'} p(s'|a,s) V(s';\mathbf{w}) \right), \\ &= \gamma \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} p_\gamma(s,a;\mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a|s;\mathbf{w}) \sum_{s'} p(s'|a,s) \frac{\partial^\top}{\partial \mathbf{w}} V(s';\mathbf{w}). \end{aligned} \quad (20)$$

Our approach is to obtain a bound of, $\frac{\partial}{\partial w_i} V(s;\mathbf{w})|_{\mathbf{w}=\mathbf{w}^*}$, for all $s' \in \mathcal{S}$ and $i \in \{1, \dots, n\}$, when $\mathbf{w}^* \in \mathcal{W}$, is a local optimum of (4). Given such a bound it is then possible to obtain a bound on the term, $\mathcal{H}_{12}(\mathbf{w}) + \mathcal{H}_{12}^\top(\mathbf{w})$, at a local optimum. In the limit the bound gives conditions under which, $\frac{\partial}{\partial \mathbf{w}} V(s';\mathbf{w}) = \mathbf{0}$, for all $s' \in \mathcal{S}$, at a local optimum, thus giving sufficient conditions under which the term, $\mathcal{H}_{12}(\mathbf{w}) + \mathcal{H}_{12}^\top(\mathbf{w})$, vanishes at a local optimum. We start by introducing the notion of a ϵ -value-consistent policy class, with $\epsilon \in \mathbb{R}$, $\epsilon \geq 0$.

Definition 2. *Given $\epsilon \in \mathbb{R}$, with $\epsilon \geq 0$, then a policy parameterisation is said to be ϵ -value-consistent w.r.t. a Markov decision process if whenever $\frac{\partial}{\partial w_i} V(\hat{s};\mathbf{w}) \neq 0$ for some $\hat{s} \in \mathcal{S}$, $\mathbf{w} \in \mathcal{W}$ and $i \in \{1, \dots, n\}$, then $\forall s \in \mathcal{S}$ it holds that either,*

$$\text{sign} \left(\frac{\partial}{\partial w_i} V(s;\mathbf{w}) \right) = \text{sign} \left(\frac{\partial}{\partial w_i} V(\hat{s};\mathbf{w}) \right), \quad (21)$$

or

$$\left| \frac{\partial}{\partial w_i} V(s;\mathbf{w}) \right| \leq \epsilon. \quad (22)$$

Furthermore, for any state, $s \in \mathcal{S}$, for which, $\frac{\partial}{\partial w_i} V(s;\mathbf{w}) = 0$, it also holds that $\frac{\partial}{\partial w_i} \pi(a|s;\mathbf{w}) = 0$, $\forall a \in \mathcal{A}$. A policy parameterisation is said to be value-consistent if it is θ -value-consistent.

This property of a policy class captures the notion that when maximally improving the value in one state, then the amount that the value of another state can decrease is bounded. When a policy class is value-consistent, then changing a parameter to maximally improve the value in one state, does not worsen the value in another state. i.e., when a policy class is value-consistent, there is no trade-off between improving the value in different states.

Example. To illustrate the concept of a value-consistent policy parameterisation we now consider two simple maze navigation MDPs, one with a value-consistent policy parameterisation, and one with a policy parameterisation that is not value-consistent. The two MDPs are displayed in Figure 1. Walls of the maze are solid lines, while the dotted lines indicate state boundaries and are passable. The agent starts, with equal probability, in one of the states marked with an ‘S’. The agent receives a positive reward for reaching the goal state, which is marked with a ‘G’, and is then reset to one of the start states. All other state-action pairs return a reward of zero. The discount factor in both MDPs is 0.95. There are four possible actions (up, down, left, right) in each state, and the optimal policy is to move, with probability one, in the direction indicated by the arrow. We define the mapping, $o : \mathcal{S} \rightarrow \{0, 1\}^4$, which indicates the presence of a wall on each of the four state boundaries. We use the notation, $\mathcal{O} := \{o(s)|s \in \mathcal{S}\}$. We consider the policy parameterisation, $\pi(a|s;\mathbf{w}) \propto \exp(\mathbf{w}^\top \phi(a,s))$, in which, $\phi : \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}^{3|\mathcal{O}|}$, is a feature map. We consider the feature map,

$$\phi(a,s) = \begin{cases} \mathbf{0} & \text{if } a = \text{up}, \\ \mathbf{e}_{i(o)+3j(s)} & \text{otherwise,} \end{cases}$$

in which, $i : \mathcal{A} \setminus \{\text{up}\} \rightarrow \{1, 2, 3\}$, is an index function over the actions, $\mathcal{A} \setminus \{\text{up}\}$, and $j : \mathcal{S} \rightarrow \{0, 1, \dots, |\mathcal{O}| - 1\}$, is an index function over the aliased states, i.e., $j(s) = j(s)$ iff $o(s) = o(s')$. We use the notation, $\mathbf{e}_i \in \mathbb{R}^{3|\mathcal{O}|}$, $i \in \{1, \dots, 3|\mathcal{O}|\}$, to denote the i^{th} standard basis vector. Perceptual aliasing (Whitehead, 1992) occurs in both MDPs under this policy parameterisation, with states 2, 3 & 4 aliased in the hallway problem, and states 4, 5 & 6 aliased in McCallum’s grid. In the hallway problem all of the aliased states have the same optimal action, and the value of these states all increase/decrease in unison. Hence, it can be seen that the policy parameterisation is value-consistent for the hallway problem. In McCallum’s grid, however, the optimal action for states 4 & 6 is to move upwards, while in state 5 it is to move downwards. In this example increasing the probability of moving downwards in state 5 will also increase the probability of moving downwards in states 4 & 6. There is a point, therefore, at which increasing the probability of moving downwards in state 5 will decrease the value of states 4 & 6. Thus this policy parameterisation is not value-consistent for McCallum’s grid. Numerical inspection of the parameter space indicates that this policy parameterisation is ϵ -value-consistent, with $\epsilon \approx 0.2$.

We now show that tabular policies—i.e., policies such that, for each state $s \in \mathcal{S}$, the conditional distribution $\pi(a|s;\mathbf{w}_s)$ is parameterized by a separate parameter vector $\mathbf{w}_s \in \mathbb{R}^{n_s}$ for some $n_s \in \mathbb{N}$ —are value-consistent, regardless of the given Markov decision process.

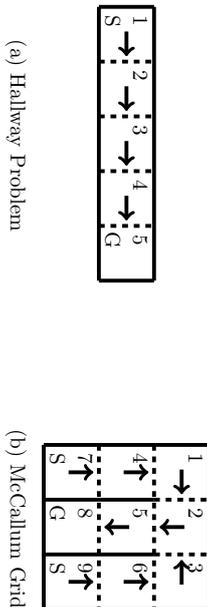


Figure 1: (a) The hallway problem. Under the feature map, ϕ , states 2, 3 and 4 map to the same feature, and the optimal policy is identical on these states. (b) McCallum’s grid. Under the feature map, ϕ , states 4, 5 and 6 map to the same feature, but now the optimal policy differs among these states.

Theorem 6. *Suppose that a given Markov decision process has a tabular policy parameterisation, then the policy parameterisation is value-consistent.*

Proof. See Section A.4 in the Appendix. \square

We now show that given an ϵ -value-consistent policy parameterisation and a local optimum, $\mathbf{w}^* \in \mathcal{W}$, the terms, $\frac{\partial}{\partial w_i} V(s; \mathbf{w}) \big|_{\mathbf{w}=\mathbf{w}^*}$, are bounded in magnitude by ϵ_i for all $s' \in \mathcal{S}$ and $i \in \{1, \dots, n\}$.

Theorem 7. *Suppose that $\mathbf{w}^* \in \mathcal{W}$ is a local optimum of the differentiable objective function, $U(\mathbf{w}) = \mathbb{E}_{s \sim p(\cdot)} [V(s; \mathbf{w})]$. Suppose that the Markov chain induced by \mathbf{w}^* is ergodic. Suppose that the policy parameterisation is ϵ -value-consistent, $\epsilon \in \mathbb{R}$, $\epsilon \geq 0$, w.r.t. the given Markov decision process. Then*

$$\left| \frac{\partial}{\partial w_i} V(s; \mathbf{w}) \bigg|_{\mathbf{w}=\mathbf{w}^*} \right| \leq \epsilon, \quad \forall s \in \mathcal{S}. \quad (23)$$

Proof. See Appendix A.5 \square

If a policy parameterisation is ϵ -value-consistent and the parameterisation satisfies the bound, $\left| \frac{\partial}{\partial w} \log \pi(a|s; \mathbf{w}) \right| \leq M$, $M \in \mathbb{R}$, for all $(s, a) \in \mathcal{S} \times \mathcal{A}$ and $\mathbf{w} \in \mathcal{W}$, then it follows from Theorem 7 that the terms of the matrix, $\mathcal{H}_{12}(\mathbf{w}^*) + \mathcal{H}_{12}^\top(\mathbf{w}^*)$, are bounded by $2e\gamma M/(1 - \gamma)$. A further corollary of Theorem 7 is that when a policy class is value-consistent the term, $\mathcal{H}_{12}(\mathbf{w}) + \mathcal{H}_{12}^\top(\mathbf{w})$, vanishes near local optima. Furthermore, when we have the additional condition that the gradient of the state value function is continuous in \mathbf{w} (at $\mathbf{w} = \mathbf{w}^*$) then $\mathcal{H}_{12}(\mathbf{w}) + \mathcal{H}_{12}^\top(\mathbf{w}) \rightarrow \mathbf{0}$ as $\mathbf{w} \rightarrow \mathbf{w}^*$. This condition will be satisfied if, for example, the policy is continuously differentiable w.r.t. the policy parameters.

Example (continued). Returning to the MDPs given in Figure 1, we now empirically observe the behaviour of the term $\mathcal{H}_{12}(\mathbf{w}) + \mathcal{H}_{12}^\top(\mathbf{w})$ as the policy approaches a local optimum of the objective function. Figure 2 gives the magnitude of $\mathcal{H}_{12}(\mathbf{w}) + \mathcal{H}_{12}^\top(\mathbf{w})$, in terms of the spectral norm, in relation to the distance from the local optimum. In correspondence with

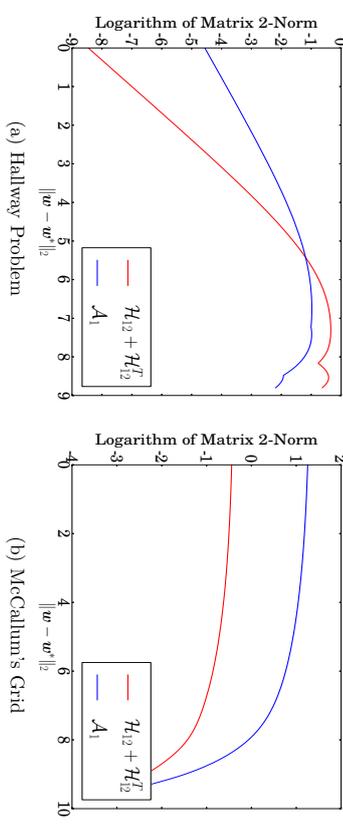


Figure 2: Graphical illustration of the logarithm of the spectral norm of $\mathcal{H}_{12}(\mathbf{w}) + \mathcal{H}_{12}^\top(\mathbf{w})$ and $\mathcal{A}_1(\mathbf{w})$ in terms of $\|\mathbf{w} - \mathbf{w}^*\|_2$ for the hallway problem (a) and McCallum’s grid (b). For the given policy parameterisation, $\mathcal{H}(\mathbf{w}) = \mathcal{A}_1(\mathbf{w}) + \mathcal{H}_{12}(\mathbf{w}) + \mathcal{H}_{12}^\top(\mathbf{w})$, so the plot displays the two components of the Hessian as the policy converges to a local optimum. As expected, in the hallway problem, $\mathcal{H}_{12}(\mathbf{w}) + \mathcal{H}_{12}^\top(\mathbf{w}) \rightarrow \mathbf{0}$ as $\mathbf{w} \rightarrow \mathbf{w}^*$. Conversely, in McCallum’s grid, $\mathcal{H}_{12}(\mathbf{w}) + \mathcal{H}_{12}^\top(\mathbf{w}) \not\rightarrow \mathbf{0}$ as $\mathbf{w} \rightarrow \mathbf{w}^*$, but the term, $\mathcal{H}_{12}(\mathbf{w}) + \mathcal{H}_{12}^\top(\mathbf{w})$, is still dominated by, $\mathcal{A}_1(\mathbf{w})$, in the vicinity of a local optima. In this example the magnitude of $\mathcal{A}_1(\mathbf{w})$ is roughly five times greater than that of $\mathcal{H}_{12}(\mathbf{w}) + \mathcal{H}_{12}^\top(\mathbf{w})$ when $\|\mathbf{w} - \mathbf{w}^*\|_2 \approx 0.0045$.

the theory, $\mathcal{H}_{12}(\mathbf{w}) + \mathcal{H}_{12}^\top(\mathbf{w}) \rightarrow \mathbf{0}$ as $\mathbf{w} \rightarrow \mathbf{w}^*$ in the hallway problem, while this is not the case in McCallum’s grid. This simple example illustrates the fact that if the feature representation is well chosen, in the sense that it is value-consistent, the term $\mathcal{H}_{12}(\mathbf{w}) + \mathcal{H}_{12}^\top(\mathbf{w})$ vanishes in the vicinity of a local optimum.

4. Gauss-Newton Methods for Markov Decision Processes

In this section we propose several Gauss-Newton type methods for MDPs, motivated by the analysis of Section 3. The algorithms are outlined in Section 4.1, and key performance analysis is provided in Section 4.2.

4.1 The Gauss-Newton Methods

The first Gauss-Newton method we propose drops the Hessian terms which are difficult to estimate, but are expected to be negligible in the vicinity of local optima. Specifically, it was shown in Section 3.3 that if the policy parameterisation is value-consistent with a given MDP, then $\mathcal{H}_{12}(\mathbf{w}) + \mathcal{H}_{12}^\top(\mathbf{w}) \rightarrow \mathbf{0}$ as \mathbf{w} converges towards a local optimum of the objective function. In such cases $\mathcal{A}_1(\mathbf{w}) + \mathcal{A}_2(\mathbf{w})$, as defined in Theorem 4, will be a good approximation to the Hessian in the vicinity of a local optimum. For this reason, the

first Gauss-Newton method that we propose for MDPs is to precondition the gradient with $\mathcal{M}(\mathbf{w}) = -(\mathcal{A}_1(\mathbf{w}) + \mathcal{A}_2(\mathbf{w}))^{-1}$ in (6), so that the update is of the form:

Policy search update using the first Gauss-Newton method

$$\mathbf{w}^{\text{new}} = \mathbf{w} - \alpha(\mathcal{A}_1(\mathbf{w}) + \mathcal{A}_2(\mathbf{w}))^{-1} \frac{\partial}{\partial \mathbf{w}} U(\mathbf{w}). \quad (24)$$

When the policy parameterisation has constant curvature with respect to the action space $\mathcal{A}_2(\mathbf{w}) = 0$ it is sufficient to calculate just $\mathcal{A}_1^{-1}(\mathbf{w})$.

The second Gauss-Newton method we propose removes further terms from the Hessian which are not guaranteed to be negative-semidefinite. As was seen in Section 3.1, when the policy parameterisation satisfies the properties of Theorem 5 then $\mathcal{H}_2(\mathbf{w})$ is negative-semidefinite over the entire parameter space.⁵ Recall that in (6) it is necessary that $\mathcal{M}(\mathbf{w})$ is positive-definite (in Newton’s method this corresponds to requiring the Hessian to be negative-definite) to ensure an increase of the objective function. That $\mathcal{H}_2(\mathbf{w})$ is negative-semidefinite over the entire parameter space is therefore a highly desirable property of a preconditioning matrix, and for this reason the second Gauss-Newton method that we propose for MDPs is to precondition the gradient with $\mathcal{M}(\mathbf{w}) = -\mathcal{H}_2^{-1}(\mathbf{w})$ in (6), so that the update is of the form:

Policy search update using the second Gauss-Newton method

$$\mathbf{w}^{\text{new}} = \mathbf{w} - \alpha \mathcal{H}_2^{-1}(\mathbf{w}) \frac{\partial}{\partial \mathbf{w}} U(\mathbf{w}). \quad (25)$$

We shall see that the second Gauss-Newton method has important performance guarantees including: a guaranteed ascent direction; linear convergence to a local optimum under a step size which does not depend upon unknown quantities; invariance to affine transformations of the parameter space; and efficient estimation procedures for the preconditioning matrix. We will also show in Section 5 that the second Gauss-Newton method is closely related to both the EM and natural gradient algorithms.

We shall also consider a diagonal form of the approximation for both forms of Gauss-Newton methods. Denoting the diagonal matrix formed from the diagonal elements of $\mathcal{A}_1(\mathbf{w}) + \mathcal{A}_2(\mathbf{w})$ and $\mathcal{H}_2(\mathbf{w})$ by $\mathcal{D}_{\mathcal{A}_1 + \mathcal{A}_2}(\mathbf{w})$ and $\mathcal{D}_{\mathcal{H}_2}(\mathbf{w})$, respectively, then we shall consider the methods that use $\mathcal{M}(\mathbf{w}) = -\mathcal{D}_{\mathcal{A}_1 + \mathcal{A}_2}^{-1}(\mathbf{w})$ and $\mathcal{M}(\mathbf{w}) = -\mathcal{D}_{\mathcal{H}_2}^{-1}(\mathbf{w})$ in (6). We call these methods the diagonal first and second Gauss-Newton methods, respectively. This

5. That the preconditioning matrix is negative-semidefinite and not negative-definite is a standard problem with many optimisation techniques that require the inversion of a preconditioning matrix. This includes natural gradient ascent when the Fisher information matrix is used to precondition the gradient (Thomas, 2014). Various approaches can be taken with regard to this problem: Add a ridge term to the preconditioning matrix; Minimize $\|\mathcal{H}_2(\mathbf{w})\mathbf{p} + \frac{\partial}{\partial \mathbf{w}} U(\mathbf{w})\|^2$ with respect to \mathbf{p} by gradient descent, and use \mathbf{p} as the search direction; Precondition the gradient with the pseudoinverse $-\mathcal{H}_2^{\dagger}(\mathbf{w})$.

diagonalization amounts to performing the approximate Newton methods on each parameter independently, but simultaneously.

4.1.1 ESTIMATION OF THE PRECONDITIONERS AND THE GAUSS-NEWTON UPDATE DIRECTION

It is possible to extend typical techniques used to estimate the policy gradient to estimate the preconditioner for the Gauss-Newton method, by including either the Hessian of the log-policy, the outer product of the derivative of the log-policy, or the respective diagonal terms. As an example, in Section B.1 of the Appendix we detail the extension of the recurrent state formulation of gradient evaluation in the average reward framework (Williams, 1992) to the second Gauss-Newton method. We use this extension in the Tetris experiment that we consider in Section 6. Given n_s sampled state-action pairs, the complexity of this extension scales as $\mathcal{O}(n_s n^2)$ for the second Gauss-Newton method, while it scales as $\mathcal{O}(n_s n)$ for the diagonal version of the algorithm. We provide more details of situations in which the inversion of the preconditioning matrices can be performed more efficiently in Section B.2 of the Appendix.

4.2 Performance Guarantees and Analysis

4.2.1 ASCENT DIRECTIONS

In general the objective (4) is not concave, which means that the Hessian will not be negative-definite over the entire parameter space. In such cases Newton’s method can actually lower the objective and this is an undesirable aspect of Newton’s method. We now consider ascent directions for the Gauss-Newton methods, and in particular demonstrate that the proposed second Gauss-Newton method guarantees an ascent direction in typical settings.

Ascent directions for the first Gauss-Newton method: As mentioned previously, the matrix $\mathcal{A}_1(\mathbf{w}) + \mathcal{A}_2(\mathbf{w})$ will typically be indefinite, and so a straightforward application of the first Gauss-Newton method will not necessarily result in an increase in the objective function. There are, however, standard correction techniques that one could consider to ensure that an increase in the objective function is obtained, such as adding a ridge term to the preconditioning matrix. A survey of such correction techniques can be found in Boyd and Vandenberghe (2004).

Ascent directions for the second Gauss-Newton method: It was seen in Theorem 5 that $\mathcal{H}_2(\mathbf{w})$ will be negative-semidefinite over the entire parameter space if either the policy is log-concave with respect to the policy parameters, or the policy has constant curvature with respect to the action space. It follows that in such cases an increase of the objective function will be obtained when using the second Gauss-Newton method with a sufficiently small step-size. Additionally, the diagonal terms of a negative-semidefinite matrix are non-positive, so that $\mathcal{D}_{\mathcal{H}_2}(\mathbf{w})$ is negative-semidefinite whenever $\mathcal{H}_2(\mathbf{w})$ is negative-semidefinite, and thus similar performance guarantees exist for the diagonal version of the second Gauss-Newton algorithm.

To motivate this result we now briefly consider some widely used policies that are either log-concave or blockwise log-concave. Firstly, consider the linear softmax policy parameterisation, $\pi(a|s; \mathbf{w}) \propto \exp \mathbf{w}^T \phi(a, s)$, in which $\phi(a, s) \in \mathbb{R}^n$ is a feature vector. This

policy is widely used in discrete systems and is log-concave in \mathbf{w} , which can be seen from the fact that $\log \pi(a|s; \mathbf{w})$ is the sum of a linear term and a negative log-sum-exp term, both of which are concave (Boyd and Vandenberghe, 2004). In systems with a continuous state-action space a common choice of controller is $\pi(a|s; K, \Sigma) = \mathcal{N}(a|K\phi(s), \Sigma)$, in which $\phi(s) \in \mathbb{R}^n$ is a feature vector. This controller is not jointly log-concave in K and Σ , but it is blockwise log-concave in K and Σ^{-1} . In terms of K the log-policy is quadratic and the coefficient matrix of the quadratic term is negative-semidefinite. In terms of Σ^{-1} the log-policy consists of a linear term and a log-determinant term, both of which are concave.

4.2.2 AFFINE INVARIANCE

An undesirable aspect of gradient ascent is that its performance is dependent on the choice of basis used to represent the parameter space. An important and desirable property of Newton’s method is that it is invariant to non-singular affine transformations of the parameter space (Boyd and Vandenberghe, 2004). The proposed approximate Newton methods have various invariance properties, and these properties are summarized in the following theorem.

Theorem 8. *The first and second Gauss-Newton methods are invariant to (non-singular) affine transformations of the parameter space. The diagonal versions of these algorithms are invariant to (non-singular) rescalings of the parameter space.*

Proof. See Section A.6 in the Appendix. \square

4.2.3 CONVERGENCE ANALYSIS

We now provide a local convergence analysis of the Gauss-Newton framework. We shall focus on the full Gauss-Newton methods, with the analysis of the diagonal Gauss-Newton method following similarly. Additionally, we shall focus on the case in which a constant step size is considered throughout, which is denoted by $\alpha \in \mathbb{R}^+$. We say that an algorithm converges linearly to a limit L at a rate $r \in (0, 1)$ if $\lim_{k \rightarrow \infty} \frac{|l(\mathbf{w}_{k+1}) - L|}{|l(\mathbf{w}_k) - L|} = r$. If $r = 0$ then the algorithm converges super-linearly. We denote the parameter update function of the first and second Gauss-Newton methods by G_1 and G_2 , respectively, so that $G_1(\mathbf{w}) = \mathbf{w} - \alpha(\mathcal{A}_1(\mathbf{w}) + \mathcal{A}_2(\mathbf{w}))^{-1} \frac{\partial}{\partial \mathbf{w}} U(\mathbf{w})$ and $G_2(\mathbf{w}) = \mathbf{w} - \alpha \mathcal{H}_2^{-1}(\mathbf{w}) \frac{\partial}{\partial \mathbf{w}} U(\mathbf{w})$. Given a matrix, $A \in L(\mathbb{R}^n)$ we denote the spectral radius of A by $\rho(A) = \max_i |\lambda_i|$, where $\{\lambda_i\}_{i=1}^n$ are the eigenvalues of A . Throughout this section we shall use $\nabla G(\mathbf{w}^*)$ to denote $\frac{\partial}{\partial \mathbf{w}} G(\mathbf{w})|_{\mathbf{w}=\mathbf{w}^*}$.

Theorem 9 (Convergence analysis for the first Gauss-Newton method). *Suppose that $\mathbf{w}^* \in \mathcal{W}$ is such that $\frac{\partial}{\partial \mathbf{w}} U(\mathbf{w})|_{\mathbf{w}=\mathbf{w}^*} = \mathbf{0}$ and $\mathcal{A}_1(\mathbf{w}^*) + \mathcal{A}_2(\mathbf{w}^*)$ is invertible, then G_1 is Fréchet differentiable at \mathbf{w}^* and $\nabla G_1(\mathbf{w}^*)$ takes the form,*

$$\nabla G_1(\mathbf{w}^*) = I - \alpha(\mathcal{A}_1(\mathbf{w}^*) + \mathcal{A}_2(\mathbf{w}^*))^{-1} \mathcal{H}(\mathbf{w}^*). \quad (26)$$

If $\mathcal{H}(\mathbf{w}^)$ and $\mathcal{A}_1(\mathbf{w}^*) + \mathcal{A}_2(\mathbf{w}^*)$ are negative-definite, and the step size is in the range,*

$$\alpha \in (0, 2/\rho((\mathcal{A}_1(\mathbf{w}^*) + \mathcal{A}_2(\mathbf{w}^*))^{-1} \mathcal{H}(\mathbf{w}^*))) \quad (27)$$

then \mathbf{w}^ is a point of attraction of the first Gauss-Newton method, the convergence is at least linear and the rate is given by $\rho(\nabla G_1(\mathbf{w}^*)) < 1$. When the policy parameterisation is*

value-consistent with respect to the given Markov decision process, then (26) simplifies to,

$$\nabla G_1(\mathbf{w}^*) = (1 - \alpha)I, \quad (28)$$

and whenever $\alpha \in (0, 2)$ then \mathbf{w}^ is a point of attraction of the first Gauss-Newton method, and the convergence to \mathbf{w}^* is linear if $\alpha \neq 1$ with a rate given by $\rho(\nabla G_1(\mathbf{w}^*)) < 1$, and convergence is super-linear when $\alpha = 1$.*

Proof. See Section A.7 in the Appendix. \square

Theorem 10 (Convergence analysis for the second Gauss-Newton method). *Suppose that $\mathbf{w}^* \in \mathcal{W}$ is such that $\frac{\partial}{\partial \mathbf{w}} U(\mathbf{w})|_{\mathbf{w}=\mathbf{w}^*} = \mathbf{0}$ and $\mathcal{H}_2(\mathbf{w}^*)$ is invertible, then G_2 is Fréchet differentiable at \mathbf{w}^* and $\nabla G_2(\mathbf{w}^*)$ takes the form,*

$$\nabla G_2(\mathbf{w}^*) = I - \alpha \mathcal{H}_2^{-1}(\mathbf{w}^*) \mathcal{H}(\mathbf{w}^*). \quad (29)$$

If $\mathcal{H}(\mathbf{w}^)$ is negative-definite and the step size is in the range,*

$$\alpha \in (0, 2/\rho(\mathcal{H}_2^{-1}(\mathbf{w}^*) \mathcal{H}(\mathbf{w}^*))) \quad (30)$$

then \mathbf{w}^ is a point of attraction of the second Gauss-Newton method, convergence to \mathbf{w}^* is at least linear and the rate is given by $\rho(\nabla G_2(\mathbf{w}^*)) < 1$. Furthermore, $\alpha \in (0, 2)$ implies condition (30). When the policy parameterisation is value-consistent with respect to the given Markov decision process, then (29) simplifies to,*

$$\nabla G_2(\mathbf{w}^*) = I - \alpha \mathcal{H}_2^{-1}(\mathbf{w}^*) \mathcal{A}_1(\mathbf{w}^*). \quad (31)$$

Proof. See Section A.7 in the Appendix. \square

The conditions of Theorem 10 look analogous to those of Theorem 9, but they differ in important ways: in Theorem 10 it is not necessary to assume that the preconditioning matrix is negative-definite and the sets in (27) will not be known in practice, whereas the condition $\alpha \in (0, 2)$ in Theorem 10 is more practical, i.e., for the second Gauss-Newton method convergence is guaranteed for a constant step size which is easily selected and does not depend upon unknown quantities.

It will be seen in Section 5.2 that the second Gauss-Newton method has a close relationship to the EM-algorithm. For this reason we postpone additional discussion about the rate of convergence of the second Gauss-Newton method until then.

5. Relation to Existing Policy Search Methods

In this section we consider the relationship between the second Gauss-Newton method and existing policy search methods. In Section 5.1 we examine its relation to natural gradient ascent and in Section 5.2 to the EM-algorithm.

5.1 Natural Gradient Ascent and the Second Gauss-Newton Method

Comparing the form of the Fisher information matrix given in (11) with \mathcal{H}_2 (16) it can be seen that there is a close relationship between natural gradient ascent and the second Gauss-Newton method: in \mathcal{H}_2 there is an additional weighting of the integrand from the state-action value function. Hence, \mathcal{H}_2 incorporates information about the reward structure of the objective function that is not present in the Fisher information matrix.

We now consider how this additional weighting affects the search direction for natural gradient ascent and the Gauss-Newton approach. Given a norm on the parameter space, $\|\cdot\|$, the steepest ascent direction at $\mathbf{w} \in \mathcal{W}$ with respect to that norm is given by,

$$\hat{\mathbf{p}} = \operatorname{argsup}_{\{\mathbf{p} \mid \|\mathbf{p}\|=1\}} \lim_{\alpha \rightarrow 0} \frac{U(\mathbf{w} + \alpha \mathbf{p}) - U(\mathbf{w})}{\alpha}.$$

Natural gradient ascent is obtained by considering the (local) norm $\|\cdot\|_{G(\mathbf{w})}$ given by $\|\mathbf{w} - \mathbf{w}'\|_{G(\mathbf{w})}^2 := (\mathbf{w} - \mathbf{w}')^\top G(\mathbf{w})(\mathbf{w} - \mathbf{w}')$, with $G(\mathbf{w})$ as in (10). The natural gradient method allows less movement in the directions that have high norm which, as can be seen from the form of (10), are those directions that induce large changes to the policy over the parts of the state-action space that are likely to be visited under the current policy parameters. More movement is allowed in directions that either induce a small change in the policy, or induce large changes to the policy, but only in parts of the state-action space that are unlikely to be visited under the current policy parameters. In a similar manner the second Gauss-Newton method can be obtained by considering the (local) norm $\|\cdot\|_{\mathcal{H}_2(\mathbf{w})}$, given by $\|\mathbf{w} - \mathbf{w}'\|_{\mathcal{H}_2(\mathbf{w})}^2 := -(\mathbf{w} - \mathbf{w}')^\top \mathcal{H}_2(\mathbf{w})(\mathbf{w} - \mathbf{w}')$ so that each term in (11) is additionally weighted by the state-action value function, $Q(s, a; \mathbf{w})$. Thus, the directions which have high norm are those in which the policy is rapidly changing in state-action pairs that are not only likely to be visited under the current policy, but also have high value. Thus the second Gauss-Newton method updates the parameters more conservatively if the behaviour in high value states is affected. Conversely, directions which induce a change only in state-action pairs of low value have low norm, and larger increments can be made in those directions.

5.2 Expectation Maximization and the Second Gauss-Newton Method

It has previously been noted (Kober and Peters, 2011) that the parameter update of gradient ascent and the EM-algorithm can be related through the function Q defined in (13). In particular, the gradient (8) evaluated at \mathbf{w}_k can be written in terms of Q as follows,

$$\frac{\partial}{\partial \mathbf{w}} U(\mathbf{w})|_{\mathbf{w}=\mathbf{w}_k} = \frac{\partial}{\partial \mathbf{w}} Q(\mathbf{w}, \mathbf{w}_k)|_{\mathbf{w}=\mathbf{w}_k},$$

while the parameter update of the EM-algorithm is given by $\mathbf{w}_{k+1} = \operatorname{argmax}_{\mathbf{w} \in \mathcal{W}} Q(\mathbf{w}, \mathbf{w}_k)$. In other words, gradient ascent moves in the direction that most rapidly increases Q with respect to the first variable, while the EM-algorithm maximizes Q with respect to the first variable. While this relationship is true, it is also quite a negative result. It states that in situations in which it is not possible to explicitly maximize Q with respect to its first variable, then the alternative, in terms of the EM-algorithm, is a generalized EM-algorithm, which is equivalent to gradient ascent. Given that the EM-algorithm is typically used to

overcome the negative aspects of gradient ascent, this is an undesirable alternative. It is possible to find the optimum of (13) numerically, but this is also undesirable as it results in a double-loop algorithm that could be computationally expensive. Finally, this result provides no insight into the behaviour of the EM-algorithm, in terms of the direction of its parameter update, when the maximization over \mathbf{w} in (13) can be performed explicitly.

We now demonstrate that the step-direction of the EM-algorithm has an underlying relationship with the second of our proposed Gauss-Newton methods. In particular, we show that under suitable regularity conditions the direction of the EM-update, $\mathbf{w}_{k+1} - \mathbf{w}_k$, is the same, up to first order, as the direction of the second Gauss-Newton method.

Theorem 11. *Suppose we are given a Markov decision process with objective (1) and Markovian trajectory distribution (2). Consider the parameter update (M-step) of expectation maximization at the k^{th} iteration of the algorithm, i.e., $\mathbf{w}_{k+1} = \operatorname{argmax}_{\mathbf{w} \in \mathcal{W}} Q(\mathbf{w}, \mathbf{w}_k)$. Provided that $Q(\mathbf{w}, \mathbf{w}_k)$ is twice continuously differentiable in the first parameter we have that,*

$$\mathbf{w}_{k+1} - \mathbf{w}_k = -\mathcal{H}_2^{-1}(\mathbf{w}_k) \frac{\partial}{\partial \mathbf{w}} U(\mathbf{w})|_{\mathbf{w}=\mathbf{w}_k} + \mathcal{O}(\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2). \quad (32)$$

Additionally, in the case where the log-policy is quadratic the relation to the approximate Newton method is exact, i.e., the second term on the r.h.s. of (32) is zero.

Proof. See Section A.8 in the Appendix. \square

Given a sequence of parameter vectors, $(\mathbf{w}_k)_{k=1}^\infty$, generated through an application of the EM-algorithm, then $\lim_{k \rightarrow \infty} \|\mathbf{w}_{k+1} - \mathbf{w}_k\| = 0$. This means that the rate of convergence of the EM-algorithm will be the same as that of the second Gauss-Newton method when considering a constant step size of one. We formalize this intuition and provide the convergence properties of the EM-algorithm when applied to Markov decision processes in the following theorem. This is, to our knowledge, the first formal derivation of the convergence properties for this application of the EM-algorithm.

Theorem 12. *Suppose that the sequence, $(\mathbf{w}_k)_{k \in \mathbb{N}}$, is generated by an application of the EM-algorithm, where the sequence converges to \mathbf{w}^* . Denote the update operation of the EM-algorithm by G_{EM} , so that $\mathbf{w}_{k+1} = G_{\text{EM}}(\mathbf{w}_k)$. Using $\nabla G_{\text{EM}}(\mathbf{w}^*)$ to denote $\frac{\partial}{\partial \mathbf{w}} G_{\text{EM}}(\mathbf{w})|_{\mathbf{w}=\mathbf{w}^*}$, then,*

$$\nabla G_{\text{EM}}(\mathbf{w}^*) = I - \mathcal{H}_2^{-1}(\mathbf{w}^*) \mathcal{H}(\mathbf{w}^*).$$

When the policy parameterisation is value-consistent with respect to the given Markov decision process this simplifies to $\nabla G_{\text{EM}}(\mathbf{w}^) = I - \mathcal{H}_2^{-1}(\mathbf{w}^*) \mathcal{A}_1(\mathbf{w}^*)$. When the Hessian, $\mathcal{H}(\mathbf{w}^*)$, is negative-definite then $\rho(\nabla G_{\text{EM}}(\mathbf{w}^*)) < 1$ and \mathbf{w}^* is a local point of attraction for the EM-algorithm.*

Proof. See Section A.9 in the Appendix. \square

6. Experiments

In this section we provide an empirical evaluation of the Gauss-Newton methods on a variety of domains. We summarize the experimental results here. For reproducibility, more details can be found in Appendix C.

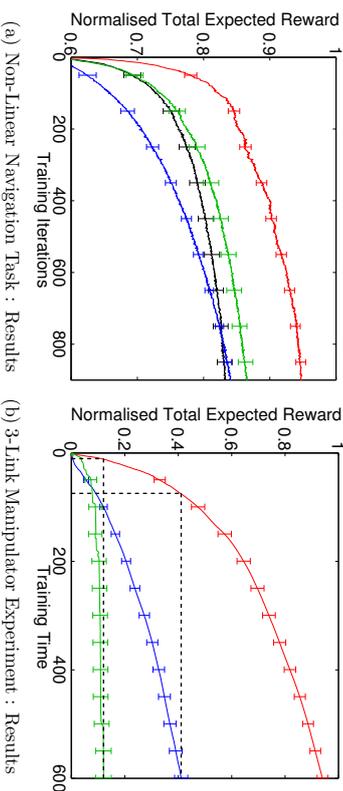


Figure 3: (a) Results from the non-linear navigation task, with the results for gradient ascent (black), expectation maximization (blue), natural gradient ascent (green) and the second Gauss-Newton method (red). (b) Normalized total expected reward plotted against training time (in seconds) for the 3-link rigid manipulator. The plot shows the results expectation maximization (blue), the second Gauss-Newton method (red) and natural gradient ascent (green).

6.1 Non-Linear Navigation Experiment

The first domain that we consider is the synthetic two-dimensional non-linear MDP considered in the work of Vlassis et al. (2009). In this experiment we consider gradient ascent, natural gradient ascent, expectation maximisation and the second Gauss-Newton method. Details of the domain and the experiment settings are given in Section C.1. The experiment was repeated 100 times and the results of the experiment are given in Figure 3a, which gives the mean and standard error of the results. The step size sequences of gradient ascent, natural gradient ascent and the Gauss-Newton method were all tuned for performance and the results shown were obtained from the best step size sequence for each algorithm.

6.2 N-link Rigid Manipulator Experiment

The N-link rigid robot arm manipulator is a standard continuous model, consisting of an end effector connected to an N-linked rigid body (Khali, 2001). A typical continuous control problem for such systems is to apply appropriate torque forces to the joints of the manipulator so as to move the end effector into a desired position. More details on the settings of the domain used in this experiment can be found in Section C.2. We consider a policy of the form,

$$\pi(a|s; \mathbf{w}) = \mathcal{N}(a|Ks + \mathbf{m}, \sigma^2 I), \quad (33)$$

with $\mathbf{w} = (K, \mathbf{m}, \sigma)$ and $s \in \mathbb{R}^{n_s}$, $a \in \mathbb{R}^{n_a}$, for some $n_s, n_a \in \mathbb{N}$. We consider a 3-link rigid manipulator, which results in a parameter space with 22 dimensions.

In this experiment we compare gradient ascent, natural gradient ascent, expectation maximization and the second Gauss-Newton method. The step size sequences of gradient ascent, natural gradient ascent and the Gauss-Newton method were all tuned for performance. Details of the experiment settings and the procedure used to tune the step size sequences are described in Section C.2. We repeated the experiment 100 times, each time with a different random initialisation of the system. The final results, obtained using the best step size sequence for each algorithm, are given in Figure 3b. We omit the result of gradient ascent as we were unable to obtain any meaningful results for this domain with this algorithm. In this experiment the maximal value of the objective function varied dramatically depending on the random initialization of the system. To account for this variation the results from each run of the experiment are normalized by the maximal value achieved between the algorithms in that run. This means that the results displayed are the percentages of reward received in comparison to the best results among the algorithms considered in the experiment. The second Gauss-Newton method significantly outperforms all of the comparison algorithms. In the experiment the Gauss-Newton method only took around 50 seconds to obtain the same performance as 300 seconds of training with expectation maximization. Furthermore expectation maximization was only able to obtain 40% of the performance of the Gauss-Newton method, while natural gradient ascent was only able to obtain around 15% of the performance. The step direction of expectation maximization is very similar to the search direction of the second Gauss-Newton method in this problem. In fact, given that the log-policy is quadratic in the mean parameters, they are the same for the mean parameters. The difference in performance between the Gauss-Newton method and expectation maximization is largely explained by the tuning of the step size in the Gauss-Newton method, compared to the constant step size of 1.0 in expectation maximization.

6.3 Tetris Experiment

In this experiment we consider the Tetris domain, which is a popular computer game designed by Alexey Pajitnov in 1985. Firstly, we compare the performance of the full and diagonal second Gauss-Newton methods to other policy search methods. We model the policy using a linear softmax parameterisation. We used the same set of features as used in the works of Bertsekas and Ioffe (1996) & Kakade (2002). Under this parameterisation it is not possible to obtain the explicit maximum over \mathbf{w} in (13), so a straightforward application of the EM-algorithm is not possible in this problem. We therefore compare the diagonal and full versions of the second Gauss-Newton method with steepest and natural gradient ascent. Due to computational costs we consider a 10×10 board in this experiment, which results in a state space with roughly 7×2^{100} states (Bertsekas and Ioffe, 1996). We ran 100 repetitions of the experiment, each consisting of 100 training iterations, and the mean and standard error of the results are given in Figure 4a. It can be seen that the full Gauss-Newton method outperforms all of the other methods, while the performance of the diagonal Gauss-Newton method is comparable to natural gradient ascent.

We also ran several training runs of the full approximate Newton method on the full-sized 20×10 board and were able to obtain a score in the region of 14,000 completed lines, which was obtained after roughly 40 training iterations. An approximate dynamic programming based method has previously been applied to the Tetris domain in the work of Bertsekas and

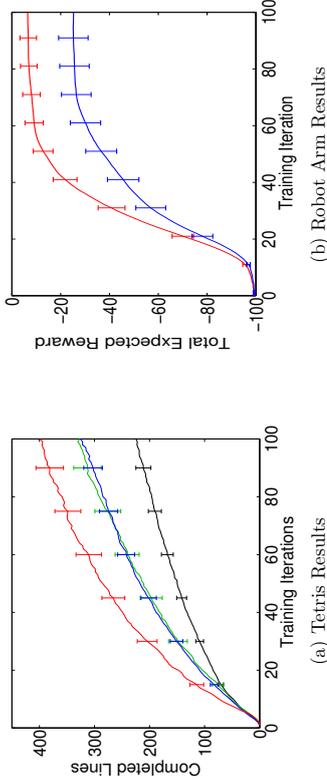


Figure 4: (a) Results from the Tetris experiment, with results for gradient ascent (black), natural gradient ascent (green), the diagonal Gauss-Newton method (blue) and the Gauss-Newton method (red). (b) Results from the robot arm experiment, with results for the second Gauss-Newton method (red) and the EM-algorithm (blue).

Ioffe (1996). The same set of features were used and a score of roughly 4,500 completed lines was obtained after around 6 training iterations, after which the solution then deteriorated. More recently a modified policy iteration approach (Gabillon et al., 2013) was able to obtain significantly better performance in the game of Tetris, completing approximately 51 million lines in a 20×10 board. However, these results were obtained through an entirely different set of features, and analysis of the results in the work of (Gabillon et al., 2013) indicates that this difference in features makes a substantial difference in performance. On a 10×10 board using the same features as in the work of Bertsekas and Ioffe (1996) the approach was able to complete approximately 500 lines on average.

6.4 Robot Arm Experiment

In the final experiment we consider a robotic arm application. We use the Simulation Lab (Schaal, 2006) environment, which provides a physically realistic engine of a Barrett WAMTM robot arm. We consider the ball-in-a-cup domain (Kober and Peters, 2009), which is a challenging motor skill problem that is based on the traditional children’s game. In this domain a small cup is attached to the end effector of the robot arm. A ball is attached to the cup through a piece of string. At the beginning of the task the robot arm is stationary and the ball is hanging below the cup in a stationary position. The aim of the task is for the robot arm to learn an appropriate set of joint movements to first swing the ball above the cup and then to catch the ball in the cup when the ball is in its downward trajectory. More details of the domain and the experiment settings are provided in Section C.4.

In this experiment we compare gradient ascent, natural gradient ascent, expectation maximization, the first Gauss-Newton method and the second Gauss-Newton method. We

repeated the experiment 50 times and the results are given in Figure 4b. We were unable to successfully learn to catch the ball in the cup using either gradient ascent, natural gradient ascent or the first Gauss-Newton method. For this reason the results for these algorithms are omitted. It can be seen that the second Gauss-Newton method significantly outperforms the EM-algorithm in this domain. Out of the 50 runs of the experiment, the second Gauss-Newton method was successfully able to learn to catch the ball in the cup 45 times. The EM-algorithm successfully learnt the task 36 times. We note that in this experiment the policy took the form, $\pi(a; \mathbf{w}) = \mathcal{N}(a | \boldsymbol{\mu}, (L L^*)^{-1})$, with, $\mathbf{w} = (\boldsymbol{\mu}, L)$. (More details of the policy parameterisation can be found in Section C.4.) A fixed step size of 1.0 was used in the second Gauss-Newton method, which means that, as the log-policy is quadratic in $\boldsymbol{\mu}$, the update of $\boldsymbol{\mu}$ in the second Gauss-Newton method and the EM-algorithm were the same. The difference in performance can therefore be attributed to the difference in the updates of L between the two algorithms.

7. Conclusions

Approximate Newton methods, such as quasi-Newton methods and the Gauss-Newton method, are standard optimization techniques. These methods aim to maintain the benefits of Newton’s method, while alleviating its shortcomings. In this paper we have considered approximate Newton methods in the context of policy optimization in MDPs. The first contribution of this paper was to provide a novel analysis of the Hessian of the total expected reward, which is a standard objective function for policy optimization. This included providing a novel form for the Hessian, as well as detailing the positive/negative semidefiniteness properties of certain terms in the Hessian. Furthermore, we have shown that when the policy parameterisation is sufficiently rich, in the sense that it is ϵ -value-consistent with an appropriately small value of ϵ , then the remaining terms in the Hessian become negligible in the vicinity of a local optimum. Motivated by this analysis we introduced two Gauss-Newton Methods for MDPs. Like the Gauss-Newton method for non-linear least squares, these methods involve approximating the Hessian by ignoring certain terms in the Hessian. The approximate Hessians possess desirable properties, such as negative-semidefiniteness, and we demonstrated several important performance guarantees including guaranteed ascent directions, invariance to affine transformation of the parameter space, and convergence guarantees. We also demonstrated our second Gauss-Newton algorithm is closely related to both the EM-algorithm and natural gradient ascent applied to MDPs, providing novel insights into both of these algorithms. We have compared the proposed Gauss-Newton methods with other techniques in the policy search literature over a range of challenging domains, including Tetris and a robotic arm application. We found that the second Gauss-Newton method performed significantly better than other methods in all of the domains that we considered.

We have provided a convergence analysis of the two proposed Gauss-Newton methods for the setting in which the gradient and the preconditioning matrices can be calculated exactly. An interesting piece of future work is to extend this analysis to the stochastic setting, in which these quantities are estimated from samples of the MDP, either through a Monte-Carlo approach or in a stochastic approximation framework.

Acknowledgements

We would like to thank Peter Dayan, David Silver, Nicolas Heess for helpful discussions on this work and Gerhard Neumann and Christian Daniel for their assistance in the robot arm experiment. We also thank the anonymous reviewers for their suggested improvements. This work was supported by the European Community Seventh Framework Programme (FP7/2007-2013) under grant agreement 270327 (ComPLACS), and by the EPSRC under grant agreement EP/M1006093/1 (C-PLACID).

Appendix A. Proofs

A.1 Proofs of Theorems 1 and 3

We begin with an auxiliary Lemma.

Lemma 1. *Suppose we are given a Markov decision process with objective (1) and Markovian trajectory distribution (2). For any given parameter vector, $\mathbf{w} \in \mathcal{W}$, the following identities hold,*

$$\frac{\partial}{\partial \mathbf{w}} V(s; \mathbf{w}) = \sum_{t=1}^{\infty} \sum_{s_t \in \mathcal{S}} \sum_{a_t \in \mathcal{A}} \gamma^{t-1} p(s_t, a_t | s_1 = s; \mathbf{w}) Q(s_t, a_t; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a_t | s_t; \mathbf{w}) \quad (34)$$

$$\frac{\partial}{\partial \mathbf{w}} Q(s, a; \mathbf{w}) = \sum_{t=2}^{\infty} \sum_{s_t \in \mathcal{S}} \sum_{a_t \in \mathcal{A}} \gamma^{t-1} p(s_t, a_t | s_1 = s, a_1 = a; \mathbf{w}) Q(s_t, a_t; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a_t | s_t; \mathbf{w}). \quad (35)$$

Proof. We start by writing the state value function in the form

$$V(s; \mathbf{w}) = \sum_{t=1}^{\infty} \sum_{s_{1:t}} \sum_{a_{1:t}} \gamma^{t-1} p(s_{1:t}, a_{1:t} | s_1 = s; \mathbf{w}) R(s_t, a_t), \quad (36)$$

so that,

$$\frac{\partial}{\partial \mathbf{w}} V(s; \mathbf{w}) = \sum_{t=1}^{\infty} \sum_{s_{1:t}} \sum_{a_{1:t}} \gamma^{t-1} p(s_{1:t}, a_{1:t} | s_1 = s; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log p(s_{1:t}, a_{1:t} | s_1 = s; \mathbf{w}) R(s_t, a_t).$$

Using the fact that

$$\frac{\partial}{\partial \mathbf{w}} \log p(s_{1:t}, a_{1:t} | s_1 = s; \mathbf{w}) = \sum_{\tau=1}^t \frac{\partial}{\partial \mathbf{w}} \log \pi(a_{\tau} | s_{\tau}; \mathbf{w}), \quad (37)$$

we have that,

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}} V(s; \mathbf{w}) &= \sum_{t=1}^{\infty} \sum_{s_t, a_t} \sum_{\tau=1}^t \sum_{s_{1:\tau}, a_{1:\tau}} \gamma^{t-1} p(s_{\tau}, a_{\tau}, s_t, a_t | s_1 = s; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a_{\tau} | s_{\tau}; \mathbf{w}) R(s_t, a_t) \\ &= \sum_{\tau=1}^{\infty} \sum_{s_{\tau}, a_{\tau}} \gamma^{\tau-1} p(s_{\tau}, a_{\tau} | s_1 = s; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a_{\tau} | s_{\tau}; \mathbf{w}) \sum_{t=\tau}^{\infty} \sum_{s_{t:\tau}, a_{t:\tau}} \gamma^{t-\tau} p(s_t, a_t | s_{\tau}, a_{\tau}; \mathbf{w}) R(s_t, a_t) \\ &= \sum_{\tau=1}^{\infty} \sum_{s_{\tau}, a_{\tau}} \gamma^{\tau-1} p(s_{\tau}, a_{\tau} | s_1 = s; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a_{\tau} | s_{\tau}; \mathbf{w}) Q(s_{\tau}, a_{\tau}; \mathbf{w}). \end{aligned} \quad (38)$$

where in the second line we swapped the order of summation and the third line follows from the definition (3). Identity (35) now follows by applying (3):

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}} Q(s, a; \mathbf{w}) &= \gamma \sum_{s'} P(s'|s, a) \frac{\partial}{\partial \mathbf{w}} V(s'; \mathbf{w}) \\ &= \gamma \sum_{s'} P(s'|s, a) \sum_{t=2}^{\infty} \sum_{s_t \in \mathcal{S}} \sum_{a_t \in \mathcal{A}} \gamma^{t-2} p(s_t, a_t | s_2 = s'; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a_t | s_t; \mathbf{w}) Q(s_t, a_t; \mathbf{w}) \\ &= \sum_{t=2}^{\infty} \sum_{s_t \in \mathcal{S}} \sum_{a_t \in \mathcal{A}} \gamma^{t-1} p(s_t, a_t | s_1 = s, a_1 = a; \mathbf{w}) Q(s_t, a_t; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a_t | s_t; \mathbf{w}). \end{aligned}$$

□

Theorem 1. *Proof.* Theorem 1 follows immediately from Lemma 1 by taking the expectation over s_1 w.r.t. the start state distribution p_1 and using the definition (5) of the discounted trajectory distribution. □

Theorem 3. *Proof.* Starting from $U(\mathbf{w}) = \sum_{t=1}^{\infty} \sum_{s_{1:t}, a_{1:t}} \gamma^{t-1} p(s_{1:t}, a_{1:t}; \mathbf{w}) R(s_t, a_t)$, the Hessian of (4) takes the form

$$\begin{aligned} \frac{\partial^2}{\partial \mathbf{w}^2} U(\mathbf{w}) &= \sum_{t=1}^{\infty} \sum_{s_{1:t}, a_{1:t}} \gamma^{t-1} p(s_{1:t}, a_{1:t}; \mathbf{w}) \frac{\partial^2}{\partial \mathbf{w}^2} \log p(s_{1:t}, a_{1:t}; \mathbf{w}) R(s_t, a_t) \\ &\quad + \sum_{t=1}^{\infty} \sum_{s_{1:t}, a_{1:t}} \gamma^{t-1} p(s_{1:t}, a_{1:t}; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log p(s_{1:t}, a_{1:t}; \mathbf{w}) \frac{\partial^{\top}}{\partial \mathbf{w}} \log p(s_{1:t}, a_{1:t}; \mathbf{w}) R(s_t, a_t). \end{aligned} \quad (39)$$

Using the fact that $\frac{\partial^2}{\partial \mathbf{w}^2} \log p(s_{1:t}, a_{1:t} | s_1 = s; \mathbf{w}) = \sum_{\tau=1}^t \frac{\partial^2}{\partial \mathbf{w}^2} \log \pi(a_{\tau} | s_{\tau}; \mathbf{w})$ we will show that the first term in (39) is equal to $\mathcal{H}_2(\mathbf{w})$ as defined in (16):

$$\begin{aligned} &\sum_{t=1}^{\infty} \sum_{s_{1:t}, a_{1:t}} \gamma^{t-1} p(s_{1:t}, a_{1:t}; \mathbf{w}) \frac{\partial^2}{\partial \mathbf{w}^2} \log p(s_{1:t}, a_{1:t}; \mathbf{w}) R(s_t, a_t) \\ &= \sum_{t=1}^{\infty} \sum_{s_{1:t}, a_{1:t}} \gamma^{t-1} p(s_{1:t}, a_{1:t}; \mathbf{w}) \sum_{\tau=1}^t \frac{\partial^2}{\partial \mathbf{w}^2} \log \pi(a_{\tau} | s_{\tau}; \mathbf{w}) R(s_t, a_t) \\ &= \sum_{\tau=1}^{\infty} \gamma^{\tau-1} \sum_{s_{\tau}, a_{\tau}} p(s_{\tau}, a_{\tau}; \mathbf{w}) \frac{\partial^2}{\partial \mathbf{w}^2} \log \pi(a_{\tau} | s_{\tau}; \mathbf{w}) \sum_{t=\tau}^{\infty} \gamma^{t-\tau} \sum_{s_t, a_t} p(s_t, a_t | s_{\tau}, a_{\tau}; \mathbf{w}) R(s_t, a_t) \\ &= \sum_{\tau=1}^{\infty} \gamma^{\tau-1} \sum_{s_{\tau}, a_{\tau}} p(s_{\tau}, a_{\tau}; \mathbf{w}) \frac{\partial^2}{\partial \mathbf{w}^2} \log \pi(a_{\tau} | s_{\tau}; \mathbf{w}) Q(s_{\tau}, a_{\tau}; \mathbf{w}). \end{aligned}$$

$$= \mathcal{H}_2(\mathbf{w})$$

where in the third line we swapped the order of summation.

Using (37) we can write the second term in (39) as,

$$\begin{aligned} &\sum_{t=1}^{\infty} \sum_{s_{1:t}, a_{1:t}} \gamma^{t-1} p(s_{1:t}, a_{1:t}; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log p(s_{1:t}, a_{1:t}; \mathbf{w}) \frac{\partial^{\top}}{\partial \mathbf{w}} \log p(s_t, a_t; \mathbf{w}) R(s_t, a_t) \\ &= \sum_{t=1}^{\infty} \sum_{\tau=1}^t \sum_{s_{1:t}, a_{1:t}} \gamma^{t-1} p(s_{1:t}, a_{1:t}; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a_{\tau} | s_{\tau}; \mathbf{w}) \frac{\partial^{\top}}{\partial \mathbf{w}} \log \pi(a_{\tau} | s_{\tau}; \mathbf{w}) R(s_t, a_t) \\ &\quad + \sum_{t=1}^{\infty} \sum_{\substack{\tau_1 \neq \tau_2 \\ \tau_1, \tau_2=1}}^t \sum_{s_{1:t}, a_{1:t}} \gamma^{t-1} p(s_{1:t}, a_{1:t}; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a_{\tau_1} | s_{\tau_1}; \mathbf{w}) \frac{\partial^{\top}}{\partial \mathbf{w}} \log \pi(a_{\tau_2} | s_{\tau_2}; \mathbf{w}) R(s_t, a_t). \end{aligned} \quad (40)$$

By swapping the order of summation and following analogous calculations to those above, it can be shown that the first term in (40) is equal to $\mathcal{H}_1(\mathbf{w})$ as defined in (15). It remains to show that the second term in (40) is given by $\mathcal{H}_{12}(\mathbf{w}) + \mathcal{H}_{12}^{\top}(\mathbf{w})$, with $\mathcal{H}_{12}(\mathbf{w})$ as given in (17). Splitting the second term in (40) into two terms,

$$\begin{aligned} &\sum_{t=1}^{\infty} \sum_{\substack{\tau_1, \tau_2=1 \\ \tau_1 \neq \tau_2}}^t \sum_{s_{1:t}, a_{1:t}} \gamma^{t-1} p(s_{1:t}, a_{1:t}; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a_{\tau_1} | s_{\tau_1}; \mathbf{w}) \frac{\partial^{\top}}{\partial \mathbf{w}} \log \pi(a_{\tau_2} | s_{\tau_2}; \mathbf{w}) R(s_t, a_t) \\ &= \sum_{t=1}^{\infty} \sum_{\tau_2=1}^t \sum_{\tau_1=1}^{\tau_2-1} \sum_{s_{1:t}, a_{1:t}} \gamma^{t-1} p(s_{1:t}, a_{1:t}; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a_{\tau_1} | s_{\tau_1}; \mathbf{w}) \frac{\partial^{\top}}{\partial \mathbf{w}} \log \pi(a_{\tau_2} | s_{\tau_2}; \mathbf{w}) R(s_t, a_t) \\ &\quad + \sum_{t=1}^{\infty} \sum_{\tau_1=1}^t \sum_{\tau_2=1}^{\tau_1-1} \sum_{s_{1:t}, a_{1:t}} \gamma^{t-1} p(s_{1:t}, a_{1:t}; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a_{\tau_1} | s_{\tau_1}; \mathbf{w}) \frac{\partial^{\top}}{\partial \mathbf{w}} \log \pi(a_{\tau_2} | s_{\tau_2}; \mathbf{w}) R(s_t, a_t), \end{aligned} \quad (41)$$

we will show that the first term is equal to $\mathcal{H}_{12}(\mathbf{w})$. Given this, it immediately follows that the second term is equal to $\mathcal{H}_{12}^{\top}(\mathbf{w})$. Using the Markov property of the transition dynamics and the policy it follows that the first term in (41) is given by,

$$\begin{aligned} &\sum_{t=1}^{\infty} \sum_{\tau_2=1}^t \sum_{\tau_1=1}^{\tau_2-1} \sum_{s_{1:t}, a_{1:t}} \gamma^{\tau_1-1} p(s_{\tau_1}, a_{\tau_1}; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a_{\tau_1} | s_{\tau_1}; \mathbf{w}) \\ &\quad \times \sum_{s_{\tau_2}, a_{\tau_2}} \gamma^{\tau_2-\tau_1} p(s_{\tau_2}, a_{\tau_2} | s_{\tau_1}, a_{\tau_1}; \mathbf{w}) \frac{\partial^{\top}}{\partial \mathbf{w}} \log \pi(a_{\tau_2} | s_{\tau_2}; \mathbf{w}) \sum_{s_t, a_t} \gamma^{t-\tau_2} p(s_t, a_t | s_{\tau_2}, a_{\tau_2}; \mathbf{w}) R(s_t, a_t). \end{aligned}$$

Rearranging the summation over t , τ_1 and τ_2 this can be rewritten in the form,

$$\begin{aligned} & \sum_{\tau_1=1}^{\infty} \sum_{s_{\tau_1}, a_{\tau_1}} \gamma^{\tau_1-1} p(s_{\tau_1}, a_{\tau_1}; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a_{\tau_1} | s_{\tau_1}; \mathbf{w}) \\ & \times \left\{ \sum_{\tau_2=\tau_1+1}^{\infty} \sum_{s_{\tau_2}, a_{\tau_2}} \gamma^{\tau_2-\tau_1} p(s_{\tau_2}, a_{\tau_2} | s_{\tau_1}, a_{\tau_1}; \mathbf{w}) \frac{\partial^{\top}}{\partial \mathbf{w}} \log \pi(a_{\tau_2} | s_{\tau_2}; \mathbf{w}) \right. \\ & \left. \sum_{t=\tau_2}^{\infty} \sum_{s_t, a_t} \gamma^{t-\tau_2} p(s_t, a_t | s_{\tau_2}, a_{\tau_2}; \mathbf{w}) R(s_t, a_t) \right\} \\ & = \sum_{\tau_1=1}^{\infty} \sum_{s_{\tau_1}, a_{\tau_1}} \gamma^{\tau_1-1} p(s_{\tau_1}, a_{\tau_1}; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a_{\tau_1} | s_{\tau_1}; \mathbf{w}) \\ & \times \sum_{\tau_2=\tau_1+1}^{\infty} \sum_{s_{\tau_2}, a_{\tau_2}} \gamma^{\tau_2-\tau_1} p(s_{\tau_2}, a_{\tau_2} | s_{\tau_1}, a_{\tau_1}; \mathbf{w}) \frac{\partial^{\top}}{\partial \mathbf{w}} \log \pi(a_{\tau_2} | s_{\tau_2}; \mathbf{w}) Q(s_{\tau_2}, a_{\tau_2}; \mathbf{w}) \\ & = \sum_{\tau_1=1}^{\infty} \sum_{s_{\tau_1}, a_{\tau_1}} \gamma^{\tau_1-1} p(s_{\tau_1}, a_{\tau_1}; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a_{\tau_1} | s_{\tau_1}; \mathbf{w}) \frac{\partial^{\top}}{\partial \mathbf{w}} Q(s_{\tau_1}, a_{\tau_1}; \mathbf{w}) \\ & = \mathcal{H}_{12}(\mathbf{w}) \end{aligned}$$

Where the penultimate line follows from (35). This completes the proof. \square

A.2 Proof of Theorem 4

Recalling that the state-action value function takes the form, $Q(s, a; \mathbf{w}) = V(s; \mathbf{w}) + A(s, a; \mathbf{w})$, the matrices $\mathcal{H}_1(\mathbf{w})$ and $\mathcal{H}_2(\mathbf{w})$ can be written in the following forms,

$$\mathcal{H}_1(\mathbf{w}) = \mathcal{A}_1(\mathbf{w}) + \mathcal{V}_1(\mathbf{w}), \quad \mathcal{H}_2(\mathbf{w}) = \mathcal{A}_2(\mathbf{w}) + \mathcal{V}_2(\mathbf{w}), \quad (42)$$

where,

$$\begin{aligned} \mathcal{A}_1(\mathbf{w}) &= \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} p_{\gamma}(s, a; \mathbf{w}) A(s, a; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a | s; \mathbf{w}) \frac{\partial^{\top}}{\partial \mathbf{w}} \log \pi(a | s; \mathbf{w}) \\ \mathcal{A}_2(\mathbf{w}) &= \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} p_{\gamma}(s, a; \mathbf{w}) A(s, a; \mathbf{w}) \frac{\partial^2}{\partial \mathbf{w}^2} \log \pi(a | s; \mathbf{w}) \\ \mathcal{V}_1(\mathbf{w}) &= \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} p_{\gamma}(s, a; \mathbf{w}) V(s, a; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a | s; \mathbf{w}) \frac{\partial^{\top}}{\partial \mathbf{w}} \log \pi(a | s; \mathbf{w}) \\ \mathcal{V}_2(\mathbf{w}) &= \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} p_{\gamma}(s, a; \mathbf{w}) V(s, a; \mathbf{w}) \frac{\partial^2}{\partial \mathbf{w}^2} \log \pi(a | s; \mathbf{w}). \end{aligned}$$

We begin with the following auxiliary lemmas.

Lemma 2. *Suppose we are given a Markov decision process with objective (1) and Markovian trajectory distribution (2). Provided that the policy satisfies the Fisher regularity conditions, then for any given parameter vector, $\mathbf{w} \in \mathcal{W}$, the matrices $\mathcal{V}_1(\mathbf{w})$ and $\mathcal{V}_2(\mathbf{w})$ satisfy the following relation $\mathcal{V}_1(\mathbf{w}) = -\mathcal{V}_2(\mathbf{w})$.*

Proof. As the policy satisfies the Fisher regularity conditions, then for any state, $s \in \mathcal{S}$, the following relation holds

$$\sum_{a \in \mathcal{A}} \pi(a | s; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a | s; \mathbf{w}) \frac{\partial^{\top}}{\partial \mathbf{w}} \log \pi(a | s; \mathbf{w}) = - \sum_{a \in \mathcal{A}} \pi(a | s; \mathbf{w}) \frac{\partial^2}{\partial \mathbf{w}^2} \log \pi(a | s; \mathbf{w}).$$

This means that $\mathcal{V}_1(\mathbf{w})$ can be written in the form

$$\begin{aligned} \mathcal{V}_1(\mathbf{w}) &= \sum_{s \in \mathcal{S}} p_{\gamma}(s; \mathbf{w}) V(s; \mathbf{w}) \sum_{a \in \mathcal{A}} \pi(a | s; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a | s; \mathbf{w}) \frac{\partial^{\top}}{\partial \mathbf{w}} \log \pi(a | s; \mathbf{w}), \\ &= - \sum_{s \in \mathcal{S}} p_{\gamma}(s; \mathbf{w}) V(s; \mathbf{w}) \sum_{a \in \mathcal{A}} \pi(a | s; \mathbf{w}) \frac{\partial^2}{\partial \mathbf{w}^2} \log \pi(a | s; \mathbf{w}) = -\mathcal{V}_2(\mathbf{w}), \end{aligned}$$

which completes the proof. \square

Lemma 3. *Suppose we are given a Markov decision process with objective (1) and Markovian trajectory distribution (2). If the policy parameterisation has constant curvature with respect to the action space, then $\mathcal{A}_2(\mathbf{w}) = \mathbf{0}$.*

Proof. When a policy parameterisation has constant curvature with respect to the action space, then we use, $\mathcal{H}_{\pi}(s, \mathbf{w})$, to denote $\frac{\partial^2}{\partial \mathbf{w}^2} \log \pi(a | s; \mathbf{w})$, for each $a \in \mathcal{A}$. Recalling Definition 2, the matrix $\mathcal{A}_2(\mathbf{w})$ takes the form,

$$\begin{aligned} \mathcal{A}_2(\mathbf{w}) &= \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} p_{\gamma}(s, a; \mathbf{w}) A(s, a; \mathbf{w}) \frac{\partial^2}{\partial \mathbf{w}^2} \log \pi(a | s; \mathbf{w}), \\ &= \sum_{s \in \mathcal{S}} p_{\gamma}(s; \mathbf{w}) \mathcal{H}_{\pi}(s, \mathbf{w}) \sum_{a \in \mathcal{A}} \pi(a | s; \mathbf{w}) A(s, a; \mathbf{w}). \end{aligned}$$

The relation $\mathcal{A}_2(\mathbf{w}) = \mathbf{0}$ follows because $\sum_{a \in \mathcal{A}} \pi(a | s; \mathbf{w}) A(s, a; \mathbf{w}) = 0$, for all $s \in \mathcal{S}$. \square

Lemmas 2 & 3, along with the relation (42), directly imply the result of Theorem 4.

A.3 Proof of Theorem 5 and Definiteness Results

Theorem 5. *Proof.* The first result follows from the fact that when the policy is log-concave with respect to the policy parameters, then $\mathcal{H}_2(\mathbf{w})$ is a non-negative mixture of negative-definite matrices, which again is negative-definite (Boyd and Vandenberghe, 2004).

The second result follows because when the policy parameterisation has constant curvature with respect to the action space, then by Lemma 3 in Section A.2 $\mathcal{A}_2(\mathbf{w}) = \mathbf{0}$, so that $\mathcal{H}_2(\mathbf{w}) = \mathcal{A}_2(\mathbf{w}) + \mathcal{V}_2(\mathbf{w}) = \mathcal{V}_2(\mathbf{w}) = -\mathcal{V}_1(\mathbf{w})$, with

$$\begin{aligned} \mathcal{V}_1(\mathbf{w}) &= \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} p_{\gamma}(s, a; \mathbf{w}) V(s, a; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log \pi(a | s; \mathbf{w}) \frac{\partial^{\top}}{\partial \mathbf{w}} \log \pi(a | s; \mathbf{w}) \\ \mathcal{V}_2(\mathbf{w}) &= \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} p_{\gamma}(s, a; \mathbf{w}) V(s, a; \mathbf{w}) \frac{\partial^2}{\partial \mathbf{w}^2} \log \pi(a | s; \mathbf{w}). \end{aligned}$$

The result now follows because $-\mathcal{V}_1(\mathbf{w})$ is negative-semidefinite for all $\mathbf{w} \in \mathcal{W}$. \square

Lemma 4. For any $\mathbf{w} \in \mathcal{W}$ the matrix $\mathcal{H}_{11}(\mathbf{w}) = \mathcal{H}_1(\mathbf{w}) + \mathcal{H}_{12}(\mathbf{w}) + \mathcal{H}_{12}^\top(\mathbf{w})$ is positive-semidefinite.

Proof. This follows immediately from the form of $\mathcal{H}_1(\mathbf{w}) + \mathcal{H}_{12}(\mathbf{w}) + \mathcal{H}_{12}^\top(\mathbf{w})$ given by (40) in Theorem 3, which is positive-semidefinite since the reward function is assumed to be non-negative. \square

A.4 Proof of Theorem 6

We first prove an auxiliary lemma about the gradient of the state value function in the case of a tabular policy. As we are considering a tabular policy we have a separate parameter vector \mathbf{w}_s for each state $s \in \mathcal{S}$. We denote the parameter vector of the entire policy by \mathbf{w} , in which this is given by the concatenation of the parameter vectors of the different states. The dimension of \mathbf{w} is given by $n = \sum_{s \in \mathcal{S}} n_s$. In order to show that tabular policies are value-consistent we start by relating the gradient of $V(\hat{s}; \mathbf{w})$ to the gradient of $V(\bar{s}; \mathbf{w})$, where the gradient is taken with respect to the policy parameters of state \bar{s} , while the policy parameters of the remaining states are held fixed.

Lemma 5. Suppose we are given a Markov decision process with a tabular policy such that $V(s; \mathbf{w})$ is differentiable for each $s \in \mathcal{S}$. Given $\bar{s}, \hat{s} \in \mathcal{S}$, such that $\bar{s} \neq \hat{s}$, then we have that

$$\frac{\partial}{\partial \mathbf{w}_{\bar{s}}} V(\hat{s}; \mathbf{w}) = p_{\text{hit}}(\hat{s} \rightarrow \bar{s}) \frac{\partial}{\partial \mathbf{w}_{\bar{s}}} V(\bar{s}; \mathbf{w}), \quad (43)$$

where the notation $\frac{\partial}{\partial \mathbf{w}_{\bar{s}}} V(\hat{s}; \mathbf{w})$ is used to denote the gradient of the state value function w.r.t. the policy parameter of state \bar{s} , with the policy parameters of all other states considered fixed. The term $p_{\text{hit}}(\hat{s} \rightarrow \bar{s})$ in (43) is given by

$$p_{\text{hit}}(\hat{s} \rightarrow \bar{s}) = \sum_{t=2}^{\infty} \gamma^{t-1} p(s_t = \bar{s} | s_1 = \hat{s}, s_\tau \neq \bar{s}, \tau = 1, \dots, t-1; \mathbf{w}).$$

Furthermore, when Markov chain induced by the policy parameters is ergodic then $p_{\text{hit}} > 0$.

Proof. Given the equality $V(s; \mathbf{w}) = \sum_{a \in \mathcal{A}} \pi(a|s; \mathbf{w}) Q(s, a; \mathbf{w})$, we have that

$$\frac{\partial}{\partial \mathbf{w}_{\bar{s}}} V(\hat{s}; \mathbf{w}) = \sum_{a \in \mathcal{A}} \left(\frac{\partial}{\partial \mathbf{w}_{\bar{s}}} \pi(a|\hat{s}; \mathbf{w}) Q(\hat{s}, a; \mathbf{w}) + \pi(a|\hat{s}; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}_{\bar{s}}} Q(\hat{s}, a; \mathbf{w}) \right).$$

As the policy is tabular and $\hat{s} \neq \bar{s}$ we have that $\frac{\partial}{\partial \mathbf{w}_{\bar{s}}} \pi(a|\hat{s}; \mathbf{w}) = 0$, so that this simplifies to

$$\frac{\partial}{\partial \mathbf{w}_{\bar{s}}} V(\hat{s}; \mathbf{w}) = \sum_{a \in \mathcal{A}} \pi(a|\hat{s}; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}_{\bar{s}}} Q(\hat{s}, a; \mathbf{w}).$$

Using the fact that $Q(s, a; \mathbf{w}) = R(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s; a) V(s'; \mathbf{w})$, we have

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}_{\bar{s}}} V(\hat{s}; \mathbf{w}) &= \gamma \sum_{s' \in \mathcal{S}} p(s'|\hat{s}; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}_{\bar{s}}} V(s'; \mathbf{w}) \\ &= \gamma p(\bar{s}|\hat{s}; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}_{\bar{s}}} V(\bar{s}; \mathbf{w}) + \gamma \sum_{\substack{s' \in \mathcal{S} \\ s' \neq \bar{s}}} p(s'|\hat{s}; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}_{\bar{s}}} V(s'; \mathbf{w}). \end{aligned} \quad (44)$$

Applying equation (44) recursively gives

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}_{\bar{s}}} V(\hat{s}; \mathbf{w}) &= \sum_{t=2}^{\infty} \gamma^{t-1} p(s_t = \bar{s} | s_1 = \hat{s}, s_\tau \neq \bar{s}, \tau = 1, \dots, t-1; \mathbf{w}) \frac{\partial}{\partial \mathbf{w}_{\bar{s}}} V(\bar{s}; \mathbf{w}) \\ &= p_{\text{hit}}(\hat{s} \rightarrow \bar{s}) \frac{\partial}{\partial \mathbf{w}_{\bar{s}}} V(\bar{s}; \mathbf{w}), \end{aligned} \quad (45)$$

which completes the proof. The probability, $p(s_t = \bar{s} | s_1 = \hat{s}, s_\tau \neq \bar{s}, \tau = 1, \dots, t-1; \mathbf{w})$, is equivalent to the probability that the first hitting time (of hitting state \bar{s} when starting in state \hat{s}) is equal to t . The strict inequality, $p_{\text{hit}}(\hat{s} \rightarrow \bar{s}) > 0$, follows from the ergodicity of the Markov chain induced by \mathbf{w} . \square

We are now ready to prove Theorem 6.

Theorem 6. *Proof.* Suppose that there exists $i \in \{1, \dots, n\}$, $\mathbf{w} \in \mathcal{W}$ and $\hat{s} \in \mathcal{S}$ such that $\frac{\partial}{\partial w_i} V(\hat{s}; \mathbf{w}) \neq 0$, for some $\hat{s} \in \mathcal{S}$. As the policy parameterisation is tabular, then the i^{th} component of \mathbf{w} corresponds to a policy parameter for a particular state, $\bar{s} \in \mathcal{S}$. From Lemma 5 it follows that

$$\frac{\partial}{\partial w_i} V(s; \mathbf{w}) = p_{\text{hit}}(s \rightarrow \bar{s}) \frac{\partial}{\partial w_i} V(\bar{s}; \mathbf{w}),$$

for all $s \in \mathcal{S}$. It follows that for states, $s \in \mathcal{S}$, for which $p_{\text{hit}}(s \rightarrow \bar{s}) > 0$ that we have

$$\text{sign} \left(\frac{\partial}{\partial w_i} V(s; \mathbf{w}) \right) = \text{sign} \left(\frac{\partial}{\partial w_i} V(\bar{s}; \mathbf{w}) \right),$$

while in states for which $p_{\text{hit}}(s \rightarrow \bar{s}) = 0$ we have $\text{sign} \left(\frac{\partial}{\partial w_i} V(s; \mathbf{w}) \right) = 0$.

It remains to show that for states in which $p_{\text{hit}}(s \rightarrow \bar{s}) = 0$ that $\text{sign} \left(\frac{\partial}{\partial w_i} \pi(a|s; \mathbf{w}) \right) = 0$, $\forall a \in \mathcal{A}$. This property follows immediately from the fact that the policy parameterisation is tabular and $p_{\text{hit}}(\bar{s} \rightarrow \bar{s}) \neq 0$. \square

A.5 Proof of Theorem 7

Lemma 6. Given a Markov decision process and a policy parameterisation that is ϵ -value-consistent, if there exists $i \in \{1, \dots, n\}$ and $\hat{s} \in \mathcal{S}$ such that,

$$\left| \frac{\partial}{\partial w_i} V(\hat{s}; \mathbf{w}) \Big|_{\mathbf{w}=\mathbf{w}^*} \right| > \epsilon, \quad (46)$$

then for each $s \in \mathcal{S}$,

$$\text{sign} \left(\frac{\partial}{\partial w_i} V(s; \mathbf{w}) \right) = \text{sign} \left(\frac{\partial}{\partial w_i} V(\hat{s}; \mathbf{w}) \right).$$

Proof. In order to obtain a contradiction suppose that there exists $s \in \mathcal{S}$ such that,

$$\text{sign} \left(\frac{\partial}{\partial w_i} V(s; \mathbf{w}) \right) \neq \text{sign} \left(\frac{\partial}{\partial w_i} V(\hat{s}; \mathbf{w}) \right). \quad (47)$$

By definition 2 it follows that for all $s' \in S$,

$$\text{sign} \left(\frac{\partial}{\partial u_i} V(s'; \mathbf{w}) \right) = \text{sign} \left(\frac{\partial}{\partial u_i} V(s; \mathbf{w}) \right), \quad (48)$$

or

$$\left| \frac{\partial}{\partial u_i} V(s'; \mathbf{w}) \right| \leq \epsilon. \quad (49)$$

From (47) it follows that,

$$\left| \frac{\partial}{\partial u_i} V(\hat{s}; \mathbf{w}) \right| \leq \epsilon. \quad (50)$$

This is a contradiction of (46), which completes the proof. \square

Theorem 7. Proof. In order to obtain a contradiction suppose that there exists $i \in \{1, \dots, n\}$ and $\hat{s} \in S$ such that,

$$\left| \frac{\partial}{\partial u_i} V(\hat{s}; \mathbf{w}) \Big|_{\mathbf{w}=\mathbf{w}^*} \right| > \epsilon. \quad (51)$$

We suppose that $\frac{\partial}{\partial u_i} V(\hat{s}; \mathbf{w}) \Big|_{\mathbf{w}=\mathbf{w}^*} > \epsilon$ (an identical argument can be used for the case $\frac{\partial}{\partial u_i} V(\hat{s}; \mathbf{w}) \Big|_{\mathbf{w}=\mathbf{w}^*} < -\epsilon$). As the policy parameterisation is ϵ -value-consistent it follows from lemma 6 that, for each $s \in S$,

$$\frac{\partial}{\partial u_i} V(s; \mathbf{w}) \Big|_{\mathbf{w}=\mathbf{w}^*} \geq 0. \quad (52)$$

In order to obtain a contradiction we will show that there is no $s \in S$ for which (52) holds with equality. Given this property a contradiction is obtained because it follows that

$$\frac{\partial}{\partial u_i} U(\mathbf{w}) \Big|_{\mathbf{w}=\mathbf{w}^*} = \mathbb{E}_{p_i(s)} \left[\frac{\partial}{\partial u_i} V(s; \mathbf{w}) \Big|_{\mathbf{w}=\mathbf{w}^*} \right] > 0,$$

contradicting the fact that \mathbf{w}^* is a local optimum of the objective function. Introducing the notation

$$\begin{aligned} S_{=} &= \{s \in S \mid \frac{\partial}{\partial u_i} V(s; \mathbf{w}) \Big|_{\mathbf{w}=\mathbf{w}^*} = 0\}, \\ S_{>} &= \{s \in S \mid \frac{\partial}{\partial u_i} V(s; \mathbf{w}) \Big|_{\mathbf{w}=\mathbf{w}^*} > 0\}, \end{aligned}$$

we wish to show that $S_{=} = \emptyset$. In particular, for a contradiction, suppose that $S_{=} \neq \emptyset$. This means, given the ergodicity of the Markov chain induced by \mathbf{w}^* and the fact that $S_{>} \neq \emptyset$, that there exists $s \in S_{=}$ and $s' \in S_{>}$ such that $p(s'|s; \mathbf{w}^*) = \sum_{a \in \mathcal{A}} p(s'|s, a) \pi(a|s; \mathbf{w}^*) > 0$. We now consider the form of $\frac{\partial}{\partial u_i} V(s; \mathbf{w}) \Big|_{\mathbf{w}=\mathbf{w}^*}$. In particular, we have

$$\begin{aligned} \frac{\partial}{\partial u_i} V(s; \mathbf{w}) &= \sum_{a \in \mathcal{A}} \frac{\partial}{\partial u_i} \pi(a|s; \mathbf{w}) \left(R(a, s) + \gamma \sum_{s_{\text{next}} \in S} p(s_{\text{next}}|s, a) V(s_{\text{next}}; \mathbf{w}) \right) \\ &+ \gamma \sum_{a \in \mathcal{A}} \pi(a|s; \mathbf{w}) \sum_{s_{\text{next}} \in S} p(s_{\text{next}}|s, a) \frac{\partial}{\partial u_i} V(s_{\text{next}}; \mathbf{w}). \end{aligned}$$

As $s \in S_{=}$, we have by value consistency that $\frac{\partial}{\partial u_i} \pi(a|s; \mathbf{w}) \Big|_{\mathbf{w}=\mathbf{w}^*} = 0$. This means that

$$\frac{\partial}{\partial u_i} V(s; \mathbf{w}) \Big|_{\mathbf{w}=\mathbf{w}^*} = \gamma \sum_{a \in \mathcal{A}} \pi(a|s; \mathbf{w}) \sum_{s_{\text{next}} \in S} p(s_{\text{next}}|s, a) \frac{\partial}{\partial u_i} V(s_{\text{next}}; \mathbf{w}) \Big|_{\mathbf{w}=\mathbf{w}^*} > 0.$$

The inequality follows from the fact that $p(s'|s; \mathbf{w}^*) > 0$, for some $s' \in S_{>}$. This is a contradiction of the fact that $s \in S_{=}$, so it follows that $S_{=} = \emptyset$ and for all $s \in S$ we have $\frac{\partial}{\partial u_i} V(s; \mathbf{w}) \Big|_{\mathbf{w}=\mathbf{w}^*} > 0$, which completes the proof. \square

A.6 Proof of Theorem 8

Theorem 8. Proof. A optimisation method is said to affine invariant if, given any objective function (for which the optimisation technique is applicable), $f: \mathcal{W} \rightarrow \mathbb{R}$, and non-singular affine mapping, $T \in \mathbb{R}^{n \times n}$, the update of the objective $f(\mathbf{w}) = f(T\mathbf{w})$ is related to the update of the original objective through the same affine mapping, i.e., $\mathbf{v} + \Delta \mathbf{v}_{\text{step}} = T(\mathbf{w} + \Delta \mathbf{w}_{\text{step}})$, in which $\mathbf{v} = T\mathbf{w}$ and $\Delta \mathbf{v}_{\text{step}}$ and $\Delta \mathbf{w}_{\text{step}}$ denote the respective steps in the parameter space.

We shall consider the second Gauss-Newton method, with the result for the diagonal approximate Newton method following similarly. Given a non-singular affine transformation, $T \in \mathbb{R}^{n \times n}$, define the objective, $\tilde{U}(\mathbf{w}) = U(T\mathbf{w}) = U(\mathbf{v})$, with $\mathbf{v} = T\mathbf{w}$, and denote the approximate Hessian of $\tilde{U}(\mathbf{w})$ by $\tilde{\mathcal{H}}_2(\mathbf{w})$. Given $\mathbf{w} \in \mathcal{W}$, then it is sufficient to show that,

$$T\mathbf{w}_{\text{new}} = T \left(\mathbf{w} - \alpha \tilde{\mathcal{H}}_2^{-1}(\mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \tilde{U}(\mathbf{w}) \right) = \mathbf{v} - \alpha \mathcal{H}_2^{-1}(\mathbf{v}) \frac{\partial}{\partial \mathbf{v}} U(\mathbf{v}) = \mathbf{v}_{\text{new}}, \quad \forall \alpha \in \mathbb{R}^+.$$

Following calculations analogous to those in Section A.1 it can be shown that,

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}} \tilde{U}(\mathbf{w}) &= \sum_{s, a} p_\gamma(s, a; T\mathbf{w}) Q(s, a; T\mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log p(a|s; T\mathbf{w}), \\ \tilde{\mathcal{H}}_2(\mathbf{w}) &= \sum_{s, a} p_\gamma(s, a; T\mathbf{w}) Q(s, a; T\mathbf{w}) \frac{\partial^2}{\partial \mathbf{w}^2} \log p(a|s; T\mathbf{w}). \end{aligned}$$

Using the relations

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}} \log \pi(a|s; T\mathbf{w}) &= T^{-\top} \frac{\partial}{\partial \mathbf{v}} \log \pi(a|s; \mathbf{v}), \\ \frac{\partial^2}{\partial \mathbf{w}^2} \log \pi(a|s; T\mathbf{w}) &= T^{-\top} \frac{\partial^2}{\partial \mathbf{v}^2} \log \pi(a|s; \mathbf{v}) T, \end{aligned}$$

it follows that

$$\begin{aligned} \frac{\partial}{\partial \mathbf{w}} \tilde{U}(\mathbf{w}) &= T^{-\top} \frac{\partial}{\partial \mathbf{v}} U(\mathbf{v}), \\ \tilde{\mathcal{H}}_2(\mathbf{w}) &= T^{-\top} \mathcal{H}_2(\mathbf{v}) T. \end{aligned}$$

From this we have, for any $\alpha \in \mathbb{R}^+$, that

$$T\mathbf{w}_{\text{new}} = T \left(\mathbf{w} - \alpha \tilde{\mathcal{H}}_2^{-1}(\mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \tilde{U}(\mathbf{w}) \right) = \mathbf{v} - \alpha \mathcal{H}_2^{-1}(\mathbf{v}) \frac{\partial}{\partial \mathbf{v}} U(\mathbf{v}) = \mathbf{v}_{\text{new}}, \quad \forall \alpha \in \mathbb{R}^+.$$

which completes the proof. \square

A.7 Proofs of Theorems 9 and 10

We begin by stating a well-known tool for analysis of convergence of iterative optimization methods. Given an iterative optimization method, defined through a mapping $G: \mathcal{W} \rightarrow \mathbb{R}^n$, where $\mathcal{W} \subseteq \mathbb{R}^n$, the local convergence at a point $\mathbf{w}^* \in \mathcal{W}$ is determined by the spectral radius of the Jacobian of G at \mathbf{w}^* , $\nabla G(\mathbf{w}^*)$. This is formalized through the well-known Ostrowski's Theorem, a formal proof of which can be found in the work of Ortega and Rheinboldt (1970).

Lemma 7 (Ostrowski's Theorem). *Suppose that we have a mapping $G: \mathcal{W} \rightarrow \mathbb{R}^n$, where $\mathcal{W} \subseteq \mathbb{R}^n$, such that $\mathbf{w}^* \in \text{int}(\mathcal{W})$ is a fixed-point of G and, furthermore, G is Fréchet differentiable at \mathbf{w}^* . If the spectral radius of $\nabla G(\mathbf{w}^*)$ satisfies $\rho(\nabla G(\mathbf{w}^*)) < 1$, then \mathbf{w}^* is a point of attraction of G . Furthermore, if $\rho(\nabla G(\mathbf{w}^*)) > 0$, then the convergence towards \mathbf{w}^* is linear and the rate is given by $\rho(\nabla G(\mathbf{w}^*))$.*

We now prove Theorems 9 and 10.

Theorem 9 (Convergence analysis for the first Gauss-Newton method). *Proof.* A formal proof that G_1 is Fréchet differentiable can be found in Section 10.2.1 of Ortega and Rheinboldt (1970). We now demonstrate the form of $\nabla G_1(\mathbf{w}^*)$. For simplicity we shall assume that $(\mathcal{A}_1(\mathbf{w}^*) + \mathcal{A}_2(\mathbf{w}^*))^{-1}$ is differentiable. This is not a necessary condition, and a proof that does not make this assumption can be found in Section 10.2.1 of Ortega and Rheinboldt (1970). We have that,

$$G_1(\mathbf{w}) = \mathbf{w} - \alpha(\mathcal{A}_1(\mathbf{w}) + \mathcal{A}_2(\mathbf{w}))^{-1} \frac{\partial^\top}{\partial \mathbf{w}} U(\mathbf{w}),$$

so that $\nabla G_1(\mathbf{w})$ is given by

$$\nabla G_1(\mathbf{w}) = I - \alpha \frac{\partial}{\partial \mathbf{w}} (\mathcal{A}_1(\mathbf{w}) + \mathcal{A}_2(\mathbf{w}))^{-1} \frac{\partial^\top}{\partial \mathbf{w}} U(\mathbf{w}) - \alpha (\mathcal{A}_1(\mathbf{w}) + \mathcal{A}_2(\mathbf{w}))^{-1} \frac{\partial^2}{\partial \mathbf{w}^2} U(\mathbf{w}).$$

The fact that $\frac{\partial^2}{\partial \mathbf{w}^2} U(\mathbf{w})|_{\mathbf{w}=\mathbf{w}^*} = \mathbf{0}$ means that

$$\nabla G_1(\mathbf{w}^*) = I - \alpha(\mathcal{A}_1(\mathbf{w}^*) + \mathcal{A}_2(\mathbf{w}^*))^{-1} \mathcal{H}(\mathbf{w}^*).$$

As $\mathcal{H}(\mathbf{w}^*)$ and $\mathcal{A}_1(\mathbf{w}^*) + \mathcal{A}_2(\mathbf{w}^*)$ are negative-definite, it follows that the eigenvalues of $(\mathcal{A}_1(\mathbf{w}^*) + \mathcal{A}_2(\mathbf{w}^*))^{-1} \mathcal{H}(\mathbf{w}^*)$ are positive. Hence,

$$\rho(\nabla G_1(\mathbf{w}^*)) = \max\{|1 - \alpha\lambda_{\min}|, |1 - \alpha\lambda_{\max}|\}, \quad (53)$$

with λ_{\min} and λ_{\max} respectively denoting the minimal and maximal eigenvalues of $(\mathcal{A}_1(\mathbf{w}^*) + \mathcal{A}_2(\mathbf{w}^*))^{-1} \mathcal{H}(\mathbf{w}^*)$. Hence, $\rho(\nabla G_1(\mathbf{w}^*)) < 1$ provided that $\alpha \in (0, 2\lambda_{\max}^{-1})$, or, written in terms of the spectral radius, $\alpha \in (0, 2/\rho(\mathcal{A}_1(\mathbf{w}^*) + \mathcal{A}_2(\mathbf{w}^*))^{-1} \mathcal{H}(\mathbf{w}^*))$.

When the policy parameterisation is value-consistent with respect to the given MDP, then from Theorem 7 $\mathcal{H}_{12}(\mathbf{w}^*) + \mathcal{H}_{22}(\mathbf{w}^*) = \mathbf{0}$, so that $\mathcal{H}(\mathbf{w}^*) = \mathcal{A}_1(\mathbf{w}^*) + \mathcal{A}_2(\mathbf{w}^*)$. It then follows that $\nabla G_1(\mathbf{w}^*) = (1 - \alpha)I$. Convergence for this case follows in the same manner. \square

Theorem 10 (Convergence analysis for the second Gauss-Newton method). *Proof.* The formulas (29) and (31) follow as in the proof of Theorem 9. Using the same approach as in Theorem 9, it can be shown that $\rho(\nabla G_2(\mathbf{w}^*)) < 1$ provided that, $\alpha \in (0, 2/\rho(\mathcal{H}_2(\mathbf{w}^*)^{-1} \mathcal{H}(\mathbf{w}^*)))$. As $\mathcal{H}(\mathbf{w}^*)$ and $\mathcal{H}_2(\mathbf{w}^*)$ are negative-definite the eigenvalues of $\mathcal{H}_2(\mathbf{w}^*)^{-1} \mathcal{H}(\mathbf{w}^*)$ are positive. Furthermore, as $\mathcal{H}(\mathbf{w}^*) = \mathcal{H}_{11}(\mathbf{w}^*) + \mathcal{H}_2(\mathbf{w}^*)$, and, by Lemma 4, $\mathcal{H}_{11}(\mathbf{w}^*)$ is positive-semidefinite, it follows that the eigenvalues of $\mathcal{H}_2(\mathbf{w}^*)^{-1} \mathcal{H}(\mathbf{w}^*)$ all lie in the range $(0, 1]$. This means that $\alpha \in (0, 2)$ is sufficient to ensure that $\rho(\nabla G_2(\mathbf{w}^*)) < 1$. \square

A.8 Proof of Theorem 11

Theorem 11. *Proof.* We use the notation $\nabla^{10} \mathcal{Q}(\mathbf{w}_j, \mathbf{w}_k)$ to denote the derivative with respect to the first variable of \mathcal{Q} , evaluated at $(\mathbf{w}_j, \mathbf{w}_k)$, and similarly $\nabla^{20} \mathcal{Q}(\mathbf{w}_j, \mathbf{w}_k)$ for the second derivative and $\nabla^{01} \mathcal{Q}(\mathbf{w}_j, \mathbf{w}_k)$ for the derivative with respect to the second variable etc. The idea of the proof is simple and consists of performing a Taylor expansion of $\nabla^{10} \mathcal{Q}(\mathbf{w}, \mathbf{w}_k)$. As \mathcal{Q} is assumed to be twice continuously differentiable in the first component this Taylor expansion is possible and gives

$$\nabla^{10} \mathcal{Q}(\mathbf{w}_{k+1}, \mathbf{w}_k) = \nabla^{10} \mathcal{Q}(\mathbf{w}_k, \mathbf{w}_k) + \nabla^{20} \mathcal{Q}(\mathbf{w}_k, \mathbf{w}_k)(\mathbf{w}_{k+1} - \mathbf{w}_k) + \mathcal{O}(\|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2). \quad (54)$$

As $\mathbf{w}_{k+1} = \text{argmax}_{\mathbf{w} \in \mathcal{W}} \mathcal{Q}(\mathbf{w}, \mathbf{w}_k)$ it follows that $\nabla^{10} \mathcal{Q}(\mathbf{w}_{k+1}, \mathbf{w}_k) = 0$. This means that, upon ignoring higher order terms in $\mathbf{w}_{k+1} - \mathbf{w}_k$, the Taylor expansion (54) can be rewritten into the form

$$\mathbf{w}_{k+1} - \mathbf{w}_k = -\nabla^{20} \mathcal{Q}(\mathbf{w}_k, \mathbf{w}_k)^{-1} \nabla^{10} \mathcal{Q}(\mathbf{w}_k, \mathbf{w}_k). \quad (55)$$

The proof is completed by observing that

$$\nabla^{10} \mathcal{Q}(\mathbf{w}_k, \mathbf{w}_k) = \frac{\partial}{\partial \mathbf{w}} U(\mathbf{w})|_{\mathbf{w}=\mathbf{w}_k}, \quad \nabla^{20} \mathcal{Q}(\mathbf{w}_k, \mathbf{w}_k) = \mathcal{H}_2(\mathbf{w}_k).$$

The second statement follows because in the case where the log-policy is quadratic the higher order terms in the Taylor expansion vanish. \square

A.9 Proof of Theorem 12

Theorem 12. *Proof.* In the EM-algorithm the update of the policy parameters takes the form

$$G_{\text{EM}}(\mathbf{w}_k) = \text{argmax}_{\mathbf{w} \in \mathcal{W}} \mathcal{Q}(\mathbf{w}, \mathbf{w}_k),$$

where the function $\mathcal{Q}(\mathbf{w}, \mathbf{w}')$ is given by

$$\mathcal{Q}(\mathbf{w}, \mathbf{w}') = \sum_{(s, \theta) \in S \times \Lambda} p_\gamma(s, \theta; \mathbf{w}') Q(s, \theta; \mathbf{w}') \left[\log \pi(\theta|s; \mathbf{w}') \right].$$

Note that \mathcal{Q} is a two parameter function, where the first parameter occurs inside the bracket, while the second parameter occurs outside the bracket. Also note that $\mathcal{Q}(\mathbf{w}, \mathbf{w}')$ satisfies

the following identities

$$\begin{aligned}\nabla^{10}\mathcal{Q}(\mathbf{w}, \mathbf{w}') &= \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} p_\gamma(s, a; \mathbf{w}') Q(s, a; \mathbf{w}') \left[\frac{\partial}{\partial \mathbf{w}} \log \pi(a|s; \mathbf{w}') \right], \\ \nabla^{20}\mathcal{Q}(\mathbf{w}, \mathbf{w}') &= \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} p_\gamma(s, a; \mathbf{w}') Q(s, a; \mathbf{w}') \left[\frac{\partial^2}{\partial \mathbf{w}^2} \log \pi(a|s; \mathbf{w}') \right], \\ \nabla^{11}\mathcal{Q}(\mathbf{w}, \mathbf{w}') &= \sum_{(s,a) \in \mathcal{S} \times \mathcal{A}} \frac{\partial}{\partial \mathbf{w}'} \left(p_\gamma(s, a; \mathbf{w}') Q(s, a; \mathbf{w}') \right) \frac{\partial^T}{\partial \mathbf{w}} \log \pi(a|s; \mathbf{w}').\end{aligned}$$

Here we have used the notation ∇^{ij} to denote the i^{th} derivative with respect to the first parameter and the j^{th} derivative with respect to the second parameter. Note that when we set $\mathbf{w} = \mathbf{w}'$ in the first two of these terms we have $\nabla^{10}\mathcal{Q}(\mathbf{w}, \mathbf{w}) = \frac{\partial}{\partial \mathbf{w}} U(\mathbf{w})$, $\nabla^{20}\mathcal{Q}(\mathbf{w}, \mathbf{w}) = \mathcal{H}_2(\mathbf{w})$. A key identity that we need for the proof is that $\nabla^{11}\mathcal{Q}(\mathbf{w}, \mathbf{w}) = \mathcal{H}_1(\mathbf{w}) + \mathcal{H}_{12}(\mathbf{w}) + \mathcal{H}_{12}^\top(\mathbf{w})$. This follows from the observation that $\frac{\partial}{\partial \mathbf{w}} U(\mathbf{w}) = \nabla^{10}\mathcal{Q}(\mathbf{w}, \mathbf{w})$, so that

$$\frac{\partial^2}{\partial \mathbf{w}^2} U(\mathbf{w}) = \frac{\partial}{\partial \mathbf{w}} \left(\nabla^{10}\mathcal{Q}(\mathbf{w}, \mathbf{w}) \right) = \nabla^{20}\mathcal{Q}(\mathbf{w}, \mathbf{w}) + \nabla^{11}\mathcal{Q}(\mathbf{w}, \mathbf{w}),$$

so that

$$\begin{aligned}\mathcal{H}_1(\mathbf{w}) + \mathcal{H}_{12}(\mathbf{w}) + \mathcal{H}_{12}^\top(\mathbf{w}) &= \mathcal{H}(\mathbf{w}) - \mathcal{H}_2(\mathbf{w}) = \nabla^{20}\mathcal{Q}(\mathbf{w}, \mathbf{w}) + \nabla^{11}\mathcal{Q}(\mathbf{w}, \mathbf{w}) - \nabla^{20}\mathcal{Q}(\mathbf{w}, \mathbf{w}), \\ &= \nabla^{11}\mathcal{Q}(\mathbf{w}, \mathbf{w}),\end{aligned}$$

as claimed.

Now, to calculate the matrix $\nabla^{GEM}(\mathbf{w}^*)$ we perform a Taylor series expansion of $\nabla^{10}\mathcal{Q}(\mathbf{w}, \mathbf{w}')$ in both parameters around the point $(\mathbf{w}^*, \mathbf{w}^*)$, and evaluated at $(\mathbf{w}_{k+1}, \mathbf{w}_k)$, which gives

$$\begin{aligned}\nabla^{10}\mathcal{Q}(\mathbf{w}_{k+1}, \mathbf{w}_k) &= \nabla^{10}\mathcal{Q}(\mathbf{w}^*, \mathbf{w}^*) + \nabla^{20}\mathcal{Q}(\mathbf{w}^*, \mathbf{w}^*)(\mathbf{w}_{k+1} - \mathbf{w}^*) \\ &\quad + \nabla^{11}\mathcal{Q}(\mathbf{w}^*, \mathbf{w}^*)(\mathbf{w}_k - \mathbf{w}^*) + \dots\end{aligned}$$

As \mathbf{w}^* is a local optimum of $U(\mathbf{w})$ we have that $\nabla^{10}\mathcal{Q}(\mathbf{w}^*, \mathbf{w}^*) = \mathbf{0}$. Furthermore, as the sequence $\{\mathbf{w}_k\}_{k \in \mathbb{N}}$ was generated by the EM-algorithm, we have, for each $k \in \mathbb{N}$, that $\mathbf{w}_{k+1} = \text{argmax}_{\mathbf{w} \in \mathcal{V}} \mathcal{Q}(\mathbf{w}, \mathbf{w}_k)$, which implies that $\nabla^{10}\mathcal{Q}(\mathbf{w}_{k+1}, \mathbf{w}_k) = \mathbf{0}$. Finally, as $\nabla^{20}\mathcal{Q}(\mathbf{w}^*, \mathbf{w}^*) = \mathcal{H}_2(\mathbf{w}^*)$ and $\nabla^{11}\mathcal{Q}(\mathbf{w}^*, \mathbf{w}^*) = \mathcal{H}_1(\mathbf{w}^*)$ we have

$$\mathbf{0} = \mathcal{H}_2(\mathbf{w}^*)(\mathbf{w}_{k+1} - \mathbf{w}^*) + (\mathcal{H}_1(\mathbf{w}^*) + \mathcal{H}_{12}(\mathbf{w}^*) + \mathcal{H}_{12}^\top(\mathbf{w}^*))(\mathbf{w}_k - \mathbf{w}^*) + \dots$$

Using the fact that $\mathbf{w}_{k+1} = GEM(\mathbf{w}_k)$ and $\mathbf{w}^* = GEM(\mathbf{w}^*)$, taking the limit $k \rightarrow \infty$ gives

$$\mathbf{0} = \mathcal{H}_2(\mathbf{w}^*) \nabla^{GEM}(\mathbf{w}^*) + \mathcal{H}_1(\mathbf{w}^*) + \mathcal{H}_{12}(\mathbf{w}^*) + \mathcal{H}_{12}^\top(\mathbf{w}^*),$$

so that

$$\nabla^{GEM}(\mathbf{w}^*) = -\mathcal{H}_2^{-1}(\mathbf{w}^*)(\mathcal{H}_1(\mathbf{w}^*) + \mathcal{H}_{12}(\mathbf{w}^*) + \mathcal{H}_{12}^\top(\mathbf{w}^*)) = I - \mathcal{H}_2^{-1}(\mathbf{w}^*)\mathcal{H}(\mathbf{w}^*).$$

In the case where the policy parameterisation value-consistent with respect to the given MDP then we have $\mathcal{H}_{12}(\mathbf{w}^*) + \mathcal{H}_{12}(\mathbf{w}^*)^\top = \mathbf{0}$, so that $\nabla^{GEM}(\mathbf{w}^*) = I - \mathcal{H}_2^{-1}(\mathbf{w}^*)\mathcal{A}_1(\mathbf{w}^*)$. The rest of the proof follows from the result in Theorem 10 when considering $\alpha = 1$. \square

Appendix B. Further Details for Estimation of Preconditioners and the Gauss-Newton Update Direction

B.1 Recurrent State Search Direction Evaluation for Second Gauss-Newton Method

In the work of Williams (1992) a sampling algorithm was provided for estimating the gradient of an infinite horizon MDP with average rewards. This algorithm makes use of a recurrent state, which we denote by \mathbf{s}^* . In Algorithm 2 we detail a straightforward extension of this algorithm to the estimation the approximate Hessian, $\mathcal{H}_2(\mathbf{w})$, in this MDP framework. The analogous algorithm for the estimation of the diagonal matrix, $\mathcal{D}_2(\mathbf{w})$, follows similarly. In Algorithm 2 we make use of an *eligibility trace* for both the gradient and the approximate Hessian, which we denote by Φ^1 and Φ^2 respectively. The estimates (up to a positive scalar) of the gradient and the approximate Hessian are denoted by Δ^1 and Δ^2 respectively.

B.2 Inversion of Preconditioning Matrices

A computational bottleneck of Newton's method is the inversion of the Hessian matrix, which scales with $\mathcal{O}(n^3)$. In a standard application of Newton's method this inversion is performed during each iteration, and in large parameter systems this becomes prohibitively costly. We now consider the inversion of the preconditioning matrix in proposed Gauss-Newton methods.

Firstly, in the diagonal forms of the Gauss-Newton methods the preconditioning matrix is diagonal, so that the inversion of this matrix is trivial and scales linearly in the number of parameters. In general the preconditioning matrix of the full Gauss-Newton methods will have no form of sparsity, and so no computational savings will be possible when inverting the preconditioning matrix. There is, however, a source of sparsity that allows for the efficient inversion of \mathcal{H}_2 in certain cases of interest. In particular, any product structure (with respect to the control parameters) in the model of the agent's behaviour will lead to sparsity in \mathcal{H}_2 . For example, in partially observable Markov decision processes in which the behaviour of the agent is modeled through a finite state controller (Meuleau et al., 1999) there are three functions that are to be optimized, the initial belief distribution, the belief transition dynamics and the policy. In this case the dynamics of the system are given by,

$$p(\mathbf{s}' | \mathbf{s}, \mathbf{o}', \mathbf{b}' | \mathbf{s}, \mathbf{o}, \mathbf{b}, \mathbf{a}; \mathbf{v}, \mathbf{w}) = p(\mathbf{s}' | \mathbf{s}, \mathbf{o}) p(\mathbf{o}' | \mathbf{s}, \mathbf{o}; \mathbf{v}) \pi(\mathbf{a}' | \mathbf{b}', \mathbf{o}'; \mathbf{w}),$$

in which $\mathbf{o} \in \mathcal{O}$ is an observation from a finite observation space, \mathcal{O} , and $\mathbf{b} \in \mathcal{B}$ is the belief state from a finite belief space, \mathcal{B} . The initial belief is given by the initial belief distribution, $p(\mathbf{b} | \mathbf{o}; \mathbf{u})$. The parameters to be optimized in this system are \mathbf{u} , \mathbf{v} and \mathbf{w} . It can be seen that in this system $\mathcal{H}_2(\mathbf{u}, \mathbf{v}, \mathbf{w})$ is block-diagonal (across the parameters \mathbf{u} , \mathbf{v} and \mathbf{w}) and the matrix inversion can be performed more efficiently by inverting each of the block matrices individually. By contrast, the Hessian $\mathcal{H}(\mathbf{u}, \mathbf{v}, \mathbf{w})$ does not exhibit any such sparsity properties.

<p>Algorithm 2: Recurrent state sampling algorithm to estimate the search direction of the second Gauss-Newton method. The algorithm is applicable to Markov decision processes with an infinite planning horizon and average rewards.</p> <p>Input: Policy parameter, $\mathbf{w} \in \mathcal{W}$, Number of restarts, $N \in \mathbb{N}$.</p> <p>Sample a state from the initial state distribution: $s_1 \sim p_1(\cdot)$.</p> <p>for $i = 1, \dots, N$ do</p> <p style="padding-left: 2em;">Given the current state, sample an action from the policy: $a_t \sim \pi(\cdot s_t; \mathbf{w})$.</p> <p>if $s_t \neq s^*$, then</p> <p style="padding-left: 2em;">update the eligibility traces: $\Phi^1 \leftarrow \Phi^1 + \frac{\partial}{\partial \mathbf{w}} \log \pi(a_t s_t; \mathbf{w}) \quad \Phi^2 \leftarrow \Phi^2 + \frac{\partial^2}{\partial \mathbf{w}^2} \log \pi(a_t s_t; \mathbf{w})$</p> <p>else</p> <p style="padding-left: 2em;">reset the eligibility traces: $\Phi^1 = \mathbf{0}, \quad \Phi^2 = \mathbf{0}$.</p> <p>end</p> <p>Update the estimates of the $\frac{\partial}{\partial \mathbf{w}} U(\mathbf{w})$ and $\mathcal{H}_2(\mathbf{w})$: $\Delta^1 \leftarrow \Delta^1 + R(a_t, s_t) \Phi^1, \quad \Delta^2 \leftarrow \Delta^2 + R(a_t, s_t) \Phi^2$.</p> <p>Sample state from the transition dynamics: $s_{t+1} \sim p(\cdot a_t, s_t)$.</p> <p>Update time-step, $t \leftarrow t + 1$.</p> <p>end</p> <p>return Δ^1 and Δ^2, which, up to a positive multiplicative constant, are estimates of $\frac{\partial}{\partial \mathbf{w}} U(\mathbf{w})$ and $\mathcal{H}_2(\mathbf{w})$.</p>
--

Appendix C. Experiments

C.1 Non-Linear Navigation Experiment

The state-space of the problem is two-dimensional, $\mathbf{s} = (s^1, s^2)$, in which s^1 is the agent's position and s^2 is the agent's velocity. The control is one-dimensional and the dynamics of

the system is given as follows,

$$\begin{aligned} s_{t+1}^1 &= s_t^1 + \frac{1}{1 + e^{-u_t}} - 0.5 + \kappa, \\ s_{t+1}^2 &= s_t^2 - 0.1 s_{t+1}^1 + \kappa, \end{aligned}$$

with κ a zero-mean Gaussian random variable with standard deviation $\sigma_\kappa = 0.02$. The agent starts in the state $\mathbf{s} = (0, 1)$, with the addition of Gaussian noise with standard deviation 0.001, and the objective is for the agent to reach the target state, $\mathbf{s}_{\text{target}} = (0, 0)$. We use the same policy as in Vlassis et al. (2009), which is given by $a_t = (\mathbf{w} + \boldsymbol{\epsilon}_t)^\top \mathbf{s}_t$, with control parameters, \mathbf{w} , and $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \sigma_\epsilon^2 I)$. The objective function is non-trivial for $\mathbf{w} \in [0, 60] \times [-8, 0]$. In the experiment the initial control parameters were sampled from the region $\mathbf{w}_0 \in [0, 60] \times [-8, 0]$. In all algorithms 50 trajectories were sampled during each training iteration and used to estimate the search direction. We consider a finite planning horizon, $H = 80$.

C.2 N -link Rigid Manipulator Experiment

The state of the system is given by $\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^N$, where $\mathbf{q}, \dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ denote the angles, velocities and accelerations of the joints respectively, while the control variables are the torques applied to the joints $\boldsymbol{\tau} \in \mathbb{R}^N$. The nonlinear state equations of the system are given by (Spong et al., 2005),

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\dot{\mathbf{q}}, \mathbf{q})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}, \quad (56)$$

where $M(\mathbf{q})$ is the inertia matrix, $C(\dot{\mathbf{q}}, \mathbf{q})$ denotes the Coriolis and centripetal forces and $\mathbf{g}(\mathbf{q})$ is the gravitational force. While this system is highly nonlinear it is possible to define an appropriate control function $\hat{\boldsymbol{\tau}}(\mathbf{q}, \dot{\mathbf{q}})$ that results in linear dynamics in a different state-action space. This technique is known as feedback linearisation (Khalil, 2001), and in the case of an N -link rigid manipulator recasts the torque action space into the acceleration action space. This means that the state of the system is now given by \mathbf{q} and $\ddot{\mathbf{q}}$, while the control is $\mathbf{a} = \ddot{\mathbf{q}}$. Ordinarily in such problems the reward would be a function of the generalized co-ordinates of the end effector, which results in a non-trivial reward function in terms of $\mathbf{q}, \dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$. This can be accounted for by modelling the reward function as a mixture of Gaussians (Hoffman et al., 2009), but for simplicity we consider the simpler problem where the reward is a function of $\mathbf{q}, \dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ directly.

We consider the finite horizon undiscounted problem in this section, so that the gradient of the objective function takes the form

$$\frac{\partial}{\partial \mathbf{w}} U(\mathbf{w}) = \int \int ds da \frac{\partial}{\partial \mathbf{w}} \log \pi(a | s; \mathbf{w}) \sum_{t=1}^H p_t(s, a; \mathbf{w}) Q(s, a, t; \mathbf{w}),$$

with the preconditioning matrices of natural gradient ascent and the Gauss-Newton methods taking analogous forms. It can be shown that for the policy parameterisation given in (33) the derivative of $\log \pi(a | s; \mathbf{w})$ is quadratic in (s, a) , for all $(s, a) \in \mathcal{S} \times \mathcal{A}$. This means that to calculate the search directions of gradient ascent, natural gradient ascent, expectation maximization and the Gauss-Newton methods it is sufficient to calculate the

first two moments of $p_t(s, a; \mathbf{w})Q(s, a, t; \mathbf{w})$ w.r.t. (s, a) , for each $t \in \{1, \dots, H\}$. These calculations can be done using the methods presented in the work of Furnston (2012).

For all algorithms that required the specification of a step size we ran the experiment over a collection of step size sequences and use the optimal step size sequence in the final experiment. In both steepest gradient ascent and natural gradient ascent we considered the following fixed step sizes: 0.001, 0.01, 0.1, 1, 10, 20, 30, 100 and 250. We were unable to obtain any reasonable results with steepest gradient ascent with any of these fixed step sizes, for which reason the results are omitted. In natural gradient ascent we found 30 to be the best step size of those considered. In the Gauss-Newton method we considered the following fixed step sizes: 10, 20, 30, 100 and 250 and found that the fixed step size of 30 gave consistently good results without overstepping in the parameter space. The smaller step sizes obtained better results than expectation maximization, but less than the fixed step size of 30. The larger step sizes often found superior results, but would sometimes overstep in the parameter space. For these reasons we used the fixed step size of 30 in the final experiment.

C.3 Tetris Experiment

In Tetris there exists a board, which is typically a 20×10 grid, which is empty at the beginning of a game. During each stage of the game a four block piece, called a tetrozoid, appears at the top of the board and begins to fall down the board. While the tetrozoid is moving the player is allowed to rotate the tetrozoid and to move it left or right. The tetrozoid stops moving once it reaches either the bottom of the board or a previously positioned tetrozoid. In this manner the board begins to fill up with tetrozoid pieces. There are seven different variations of tetrozoid, as shown in Figure 5a. When a horizontal line of the board is completely filled with (pieces of) tetrozoids the line is removed from the board and the player receives a score of one. The game terminates when the player is not able to fully place a tetrozoid on the board due to insufficient space remaining on the board. An example configuration of the board during a game of Tetris is given in Figure 5b. More details on the game of Tetris can be found in the work of Falvey (2003). As in other applications of Tetris in the reinforcement learning literature (Kakade, 2002; Bertsekas and Ioffe, 1996) we consider a simplified version of the game in which the current tetrozoid remains above the board until the player decides upon a desired rotation and column position for the tetrozoid.

We use the same procedure to evaluate the search direction for all the algorithms in the experiment. Irrespective of the policy, a game of Tetris is guaranteed to terminate after a finite number of turns (Bertsekas and Ioffe, 1996). We therefore model each game as an absorbing state MDP. The reward at each time step is equal to the number of lines deleted. We use a recurrent state approach (Williams, 1992) to estimate the gradient, using the empty board as a recurrent state. (Since a new game starts with an empty board this state is recurrent.⁶) We use analogous versions of this recurrent state approach for natural gradient ascent, the diagonal Gauss-Newton method and the full Gauss-Newton

6. This is actually an approximation because it does not take into account that the state is given by the configuration of the board and the current piece, so this particular ‘recurrent state’ ignores the current piece. Empirically we found that this approximation gave better results, presumably due to reduced variance in the estimands, and there is no reason to believe that it is unfairly biasing the comparison between the various parametric policy search methods.

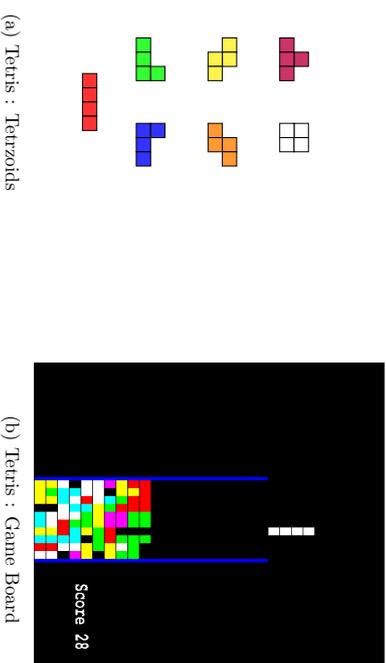


Figure 5: A graphical illustration of the game of tetris with (a) the collection of possible pieces, or tetrozoids, of which there are seven (b) a possible configuration of the board, which in this example is of height 20 and width 10.

method. As in the work of Kakade (2002), we use the sample trajectories obtained during the gradient evaluation to estimate the Fisher information matrix. During each training iteration an approximation of the search direction is obtained by sampling 1000 games, using the current policy to sample the games. It is computationally very expensive to perform experiments on the Tetris domain. When performing the experiment we found that it would be prohibitively expensive to perform an extensive sweep over different step size sequences for all of the different algorithms. For this reason we decided to implement a simple line search in this domain. Given the current approximate search direction we use the following basic line search method to obtain a step size: For every step size in a given finite set of step sizes sample a set number of games and then return the step size with the maximal score over these games. In practice, in order to reduce the susceptibility to random noise, we used the same simulator seed for each possible step size in the set. In this line search procedure we sampled 1000 games for each of the possible step sizes. We use the same set of step sizes

$$\{0.1, 0.5, 1.0, 2.0, 4.0, 8.0, 16.0, 32.0, 64.0, 128.0\}.$$

in all of the different training algorithms in the experiment. To reduce the amount of noise in the results we use the same set of simulator seeds in the search direction evaluation for each of the algorithms considered in the experiment. In particular, we generate a $n_{\text{experiments}} \times n_{\text{iterations}}$ matrix of simulator seeds, with $n_{\text{experiments}}$ the number of repetitions of the experiment and $n_{\text{iterations}}$ the number of training iterations in each experiment. We use this one matrix of simulator seeds in all of the different training algorithms, with the element in the j^{th} column and i^{th} row corresponding to the simulator seed of the j^{th} training iteration of the i^{th} experiment. In a similar manner, the set of simulator seeds we use for the line search procedure is the same for all of the different training algorithms. Finally, to

make the line search consistent among all of the different training algorithms we normalize the search direction and use the resulting unit vector in the line search procedure.

C.4 Robot Arm Experiment

The domain in the robot arm experiment is episodic, with each episode 20 seconds in length. The state of the domain is given by the angles and velocities of the seven joints in the robot arm, along with the Cartesian coordinates of the ball. The action is given by the joint accelerations of the robot arm. We denote the position of the cup and the ball by $(x_c, y_c, z_c) \in \mathbb{R}^3$ and $(x_b, y_b, z_b) \in \mathbb{R}^3$ respectively. The reward function is given by,

$$r(x_c, y_c, x_b, y_b, t) = \begin{cases} -20((x_c - x_b)^2 + (y_c - y_b)^2) & \text{if } t = t_c, \\ 0 & \text{if } t \neq t_c, \end{cases}$$

in which t_c is the moment the ball crosses the z -plane (level with the cup) in a downward direction. If no such t_c exists then the reward of the episode is given by -100 .

We use the motor primitive framework (Jipspeert et al., 2002, 2003; Schaal et al., 2007; Kober and Peters, 2011) in this domain, applying a separate motor primitive to each dimension of the action space. Each motor primitive consists of a parametrized curve that models the desired action sequence (for the respective dimension of the action space) through the course of the episode. Given this collection of motor primitives the control engine within the simulator tries to follow the desired action sequence as closely as possible while also satisfying the constraints on the system, such as the physical constraints on the torques that can safely be applied without damaging the robot arm. As in the work of Kober and Peters (2011) we use dynamic motor primitives, using 10 shape parameters for each of the individual motor primitives. The robot arm has 7 joints, so that there are 70 motor primitive parameters in total. We optimize the parameters of the motor primitives by considering the MDP induced by this motor primitive framework. The action space corresponds to the space of possible motor primitives, so that $\mathcal{A} = \mathbb{R}^{70}$. There is no state space in this MDP and the planning horizon is 1, so that this MDP is effectively a bandit problem. The reward of an action is equal to the total reward of the episode induced by the motor primitive. We consider a policy of the form,

$$\pi(a; \mathbf{w}) = \mathcal{N}(a | \boldsymbol{\mu}, (LL^*)^{-1}),$$

with $\mathbf{w} = (\boldsymbol{\mu}, L)$, $\boldsymbol{\mu}$ the mean of the Gaussian and LL^* the Cholesky decomposition of the precision matrix. We consider a diagonal precision matrix, which results in a total of 140 policy parameters.

In this experiment we compare gradient ascent, natural gradient ascent, expectation maximization, the first Gauss-Newton method and the second Gauss-Newton method. As the planning horizon is of length 1 it follows that $\mathcal{H}_{12}(\mathbf{w}) = \mathbf{0}$, $\forall \mathbf{w} \in \mathcal{W}$, so that the first Gauss-Newton method coincides with Newton's method for this MDP. The policy is block-wise log-concave in $\boldsymbol{\mu}$ and L , but not jointly log-concave in $\boldsymbol{\mu}$ and L . As a result we construct block diagonal forms of the preconditioning matrices for the first and second Gauss-Newton methods, with a separate block for $\boldsymbol{\mu}$ and L . Additionally, since the planning horizon is of length 1 it is possible to calculate the Fisher information exactly in this domain. For gradient ascent and natural gradient ascent we considered several different step size sequences. Each

sequence considered had a constant step size throughout, and the sequences differed in the size of this step size. We considered step sizes of length 1, 0.1, 0.01 and 0.001. For both Gauss-Newton methods we considered a fixed step size of one throughout training (i.e., no tuning of the step size sequence was performed for either the first or the second Gauss-Newton methods). As in the work of Kober and Peters (2009) the initial value of $\boldsymbol{\mu}$ is set so that the trajectory of the robot arm mimics that of a given human demonstration. The diagonal elements of the precision matrix are initialized to 0.01. During each training iteration we sampled 15 actions from the policy and used the episodes generated from these samples to estimate the search direction. To deal with this low number of samples we used the 'effective' sample size up to 150. Finally, we used the reward/fitness shaping approach of Wierstra et al. (2014) in all the algorithms considered, using the same shaping function as in Wierstra et al. (2014). In each run of the experiment we performed 100 updates of the policy parameters.

References

- P. Abbeel, A. Coates, M. Quigley, and A. Ng. An application of reinforcement learning to aerobatic helicopter flight. *NIPS*, 19:1–8, 2007.
- S. Amari. Neural learning in structured parameter spaces - natural Riemannian gradient. *NIPS*, 9:127–133, 1997.
- S. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10:251–276, 1998.
- S. Amari, K. Kurata, and H. Nagaoaka. Information geometry of Boltzmann machines. *IEEE Transactions on Neural Networks*, 3(2):260–271, 1992.
- S. Amari, A. Cichocki, and H. Yang. A new learning algorithm for blind signal separation. *NIPS*, 8:757–763, 1996.
- J. Bagnell and J. Schneider. Covariant policy search. *IJCAI*, 18:1019–1024, 2003.
- D. Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.
- J. Baxter and P. Bartlett. Infinite horizon policy gradient estimation. *Journal of Artificial Intelligence Research*, 15:319–350, 2001.
- J. Baxter, P. Bartlett, and L. Weaver. Experiments with infinite horizon policy gradient estimation. *Journal of Artificial Intelligence Research*, 15:351–381, 2001.
- R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.
- D. P. Bertsekas. Approximate policy iteration: a survey and some new methods. Research report, Massachusetts Institute of Technology, 2010.
- D. P. Bertsekas and S. Ioffe. Temporal differences-based policy iteration and applications in neuro-dynamic programming. Research Report LIDS-P-2349, Massachusetts Institute of Technology, 1996.

- S. Bhatnagar, R. Sutton, M. Ghavamzadeh, and M. Lee. Incremental natural actor-critic algorithms. *NIPS*, 20:105–112, 2008.
- S. Bhatnagar, R. Sutton, M. Ghavamzadeh, and L. Mark. Natural actor-critic algorithms. *Automatica*, 45:2471–2482, 2009.
- S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- C. Browne, E. Powley, D. Whitehouse, S. Lucas, P. Cowling, P. Rohlfshagen, S. Tavenier, D. Perez, S. Samothrakis, and S. Colton. A survey of Monte Carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4:1–43, 2012.
- A. Coolen, R. Kuehn, and P. Sollich. *Theory of Neural Information Processing Systems*. OUP Oxford, 2005.
- R. Crites and A. Barto. Improving elevator performance using reinforcement learning. *NIPS*, 8:1017–1023, 1995.
- P. Dayan and G. E. Hinton. Using expectation-maximization for reinforcement learning. *Neural Computation*, 9:271–278, 1997.
- M. P. Deisenroth and C. E. Rasmussen. PILCO: A model-based and data-efficient approach to policy search. *ICML*, 28, 2011.
- A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.
- C. Fahy. Tetris AI, computers play Tetris http://colinfahy.com/tetris/tetris_en.html, 2003.
- T. Furnston. *Applications of Probabilistic Inference to Planning & Reinforcement Learning*. PhD thesis, University College London, 2012.
- T. Furnston and D. Barber. Solving deterministic policy (PO)MPPs using expectation-maximisation and antifreeze. *ECML*, 1:50–65, 2009. Workshop on Learning and data Mining for Robotics.
- T. Furnston and D. Barber. Variational methods for reinforcement learning. *AISTATS*, 9: 241–248, 2010.
- V. Gabillon, M. Ghavamzadeh, and B. Scherrer. Approximate dynamic programming finally performs well in the game of Tetris. *NIPS*, 26, 2013.
- S. Gelly and D. Silver. Achieving master level play in 9 x 9 computer Go. *AAAI*, 23: 1537–1540, 2008.
- M. Ghavamzadeh and Y. Engel. Bayesian policy gradient algorithms. *NIPS*, 19:457–464, 2007.
- P. W. Glynn. Stochastic approximation for Monte-Carlo optimisation. *Proceedings of the 1986 ACM Winter Simulation Conference*, 18:356–365, 1986.
- P. W. Glynn. Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM*, 33:97–84, 1990.
- E. Greensmith, P. Bartlett, and J. Baxter. Variance reduction techniques for gradient based estimates in reinforcement learning. *Journal of Machine Learning Research*, 5:1471–1530, 2004.
- N. Heess, G. Wayne, D. Silver, T. Lillicrap, T. Erez, and Y. Tassa. Learning continuous control policies by stochastic value gradients. *NIPS*, 27:2926–2934, 2015.
- M. Hoffman, N. de Freitas, A. Doucet, and J. Peters. An expectation maximization algorithm for continuous Markov decision processes with arbitrary rewards. *AISTATS*, 12(5): 232–239, 2009.
- R. A. Howard. *Dynamic Programming and Markov Processes*. M.I.T. Press, 1960.
- A. Ijspeert, J. Nakanishi, and S. Schaal. Motor imitation with nonlinear dynamical systems in humanoid robots. *IEEE International Conference on Robotic and Automation*, pages 1398–1403, 2002.
- A. Ijspeert, J. Nakanishi, and S. Schaal. Learning attractor landscapes for learning motor primitives. *NIPS*, 15:1547–1554, 2003.
- D. Jacobson and D. Mayne. *Differential Dynamic Programming*. Elsevier, 1970.
- S. Kakade. Optimizing average reward using discounted rewards. *COLT*, 14:605–615, 2001.
- S. Kakade. A natural policy gradient. *NIPS*, 14, 2002.
- S. Kakade and J. Langford. Approximately optimal approximate reinforcement learning. *ICML*, 2:267–274, 2002.
- H. Khalil. *Nonlinear Systems*. Prentice Hall, 2001.
- J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *Annals of Mathematical Statistics*, 23:462–466, 1952.
- J. Köber and J. Peters. Policy search for motor primitives in robotics. *NIPS*, 21:849–856, 2009.
- J. Köber and J. Peters. Policy search for motor primitives in robotics. *Machine Learning*, 84(1-2):171–203, 2011.
- L. Kocsis and C. Szepesvári. Bandit based Monte-Carlo planning. *ECML*, 17:282–293, 2006.
- N. Kohl and P. Stone. Policy gradient reinforcement learning for fast quadrupedal locomotion. *IEEE International Conference on Robotics and Automation*, 2004.

- V. Konda and J. Tsitsiklis. Actor-critic algorithms. *NIPS*, 11:1008–1014, 1999.
- V. R. Konda and J. N. Tsitsiklis. On actor-critic algorithms. *SIAM J. Control Optim.*, 42(4):1143–1166, 2003.
- E. L. Lehmann and G. Casella. *Theory of Point Estimation*. Springer, 1998.
- S. Levine and V. Koltun. Variational policy search via trajectory optimization. *NIPS*, 27:207–215, 2013a.
- S. Levine and V. Koltun. Guided policy search. *ICML*, 30, 2013b.
- W. Li. *Optimal Control for Biological Movement Systems*. PhD thesis, University of San Diego, 2006.
- W. Li and E. Todorov. Iterative linear quadratic regulator design for nonlinear biological movement systems. *International Conference on Informatics in Control, Automation and Robotics*, 1, 2004.
- W. Li and E. Todorov. Iterative optimal control and estimation design for nonlinear stochastic systems. *IEEE Conference on Decision and Control*, 45, 2006.
- T. Lillicrap, J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *ICLR*, 4, 2016.
- R. Little and D. Rubin. *Statistical Analysis with Missing Data*. Wiley-Blackwell, 2002.
- P. Marbach and J. Tsitsiklis. Simulation-based optimisation of Markov reward processes. *IEEE Transactions on Automatic Control*, 46(2):191–209, 2001.
- N. Meuleau, L. Peshkin, K. Kim, and L. Kaelbling. Learning finite-state controllers for partially observable environments. *UAI*, 15:427–436, 1999.
- V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabnis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015.
- T. Morimura, E. Uchibe, J. Yoshimoto, and K. Doya. A new natural policy gradient by stationary distribution metric. *ECML*, 19:82–97, 2008.
- R. Neal and G. Hinton. A view of the EM algorithm that justifies incremental, sparse and other variants. *Learning in Graphical Models*, pages 355–368, 1999.
- A. Ngo, Y. Hwanjo, and C. TaeChoong. Hessian matrix distribution for Bayesian policy gradient reinforcement learning. *Information Sciences*, 181:1671–1685, 2011.
- J. Nocedal and S. Wright. *Numerical Optimisation*. Springer, 2006.
- J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, first edition, 1970.
- FURMSTON, LEVER AND BARBER
- J. Peters and S. Schaal. Natural actor-critic. *Neurocomputing*, 71(7-9):1180–1190, 2008.
- K. Rawlik, M. Toussaint, and S. Vijayakumar. On stochastic optimal control and reinforcement learning by approximate inference. *International Conference on Robotics Science and Systems*, 2012.
- S. Richter, D. Aberdeen, and J. Yu. Natural actor-critic for road traffic optimisation. *NIPS*, 19:1169–1176, 2007.
- S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2009.
- L. Saul, T. Jaakkola, and M. Jordan. Mean field theory for sigmoid belief networks. *Journal of Artificial Intelligence Research*, 4:61–76, 1996.
- S. Schaal. The SL simulation and real-time control software package. Technical report, University of Southern California, 2006.
- S. Schaal, P. Mohajeri, and A. J. Ijspeert. Dynamics systems vs. optimal control - a unifying view. *Progress in Brain Research*, 165(1):425–445, 2007.
- N. Schraudolph, J. Yu, and D. Aberdeen. Fast online policy gradient learning with SMD gain vector adaptation. *NIPS*, 18:1185–1192, 2006.
- N. Schraudolph, J. Yu, and S. Gunter. A stochastic quasi-Newton method for online convex optimization. *AISTATS*, 11:433–440, 2007.
- J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. *ICML*, 32:1889–1897, 2015.
- D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller. Deterministic policy gradient algorithms. *ICML*, 31:387–395, 2014.
- D. Silver, A. Huang, C. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabnis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503, 2016.
- J. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37:332–341, 1992.
- J. Spall and J. Cristion. Model-free control of nonlinear stochastic systems with discrete-time measurements. *IEEE Transactions on Automatic Control*, 43:1198–1210, 1998.
- M. Spong, S. Hutchinson, and M. Vidyasagar. *Robot Modelling and Control*. John Wiley & Sons, 2005.
- D. Srinivasan, M. C. Choy, and R. L. Cheu. Neural networks for real-time traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 7:261–272, 2006.
- R. Stengel. *Optimal Control and Estimation*. Dover, 1993.

- R. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. *NIPS*, 13, 2000.
- R. Tedrake and T. Zhang. Learning to walk in 20 minutes. *Proceedings of the Fourteenth Yale Workshop on Adaptive and Learning Systems*, 2005.
- G. Tesoro, TD-Gammon, a self-teaching backgammon program achieves master-level play. *Neural Computation*, 6:215–219, 1994.
- P. S. Thomas. Genga: A generalization of natural gradient ascent with positive and negative convergence results. *ICML*, 20:1575–1583, 2014.
- E. Todorov and Y. Tassa. Iterative local dynamic programming. *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, pages 90–95, 2009.
- M. Toussaint, S. Harmeling, and A. Storkey. Probabilistic inference for solving (PO)MDPs. Research Report EDL-INF-RR-0934, University of Edinburgh, School of Informatics, 2006.
- M. Toussaint, A. Storkey, and S. Harmeling. *Bayesian Time Series Models*, chapter Expectation-maximization methods for solving (PO)MDPs and optimal control problems. Cambridge University Press, 2011.
- J. Veness, D. Silver, A. Blair, and W. Uther. Bootstrapping from game tree search. *NIPS*, 19:1937–1945, 2009.
- N. Vlassis, M. Toussaint, G. Kontes, and S. Piperidis. Learning model-free robot control by a Monte-Carlo EM algorithm. *Autonomous Robots*, 27(2):123–130, 2009.
- L. Weaver and N. Tao. The optimal reward baseline for gradient based reinforcement learning. *UAI*, 17(29), 2001.
- S. Whitehead. *Reinforcement Learning for Adaptive Control of Perception and Action*. PhD thesis, University of Rochester, 1992.
- D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, J. Peters, and J. Schmidhuber. Natural evolution strategies. *Journal of Machine Learning Research*, 15:949–980, 2014.
- R. Williams. Simple statistical gradient following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

Structure-Leveraged Methods in Breast Cancer Risk Prediction

Jun Fan

*Department of Statistics
University of Wisconsin-Madison
1300 University Avenue, Madison, WI 53706, United States*

JUNFAN@STAT.WISC.EDU

Yirong Wu

*Department of Radiology
University of Wisconsin-Madison
600 Highland Avenue, Madison, WI 53792, United States*

YWU@UWHEALTH.ORG

Ming Yuan

*Department of Statistics
University of Wisconsin-Madison
1300 University Avenue, Madison, WI 53706, United States*

MYUAN@STAT.WISC.EDU

David Page

*Department of Biostatistics and Medical Informatics
University of Wisconsin-Madison
600 Highland Avenue, Madison, WI 53792, United States*

PAGE@BIostat.WISC.EDU

Jie Liu

*Department of Genome Sciences
University of Washington-Seattle
3720 15th Avenue, Seattle, WA 98105, United States*

LIUJ@UW.EDU

Irene M. Ong

*Department of Biostatistics and Medical Informatics
University of Wisconsin-Madison
600 Highland Avenue, Madison, WI 53792, United States*

ONG@CS.WISC.EDU

Peggy Peissig

*Marshfield Clinic Research Foundation
1000 North Oak Avenue, Marshfield, WI 54449, United States*

PEISSIG.PEGGY@MCRF.MFLDCLIN.EDU

Elizabeth Burnside

*Department of Radiology
University of Wisconsin-Madison
600 Highland Avenue, Madison, WI 53792, United States*

EBURNSIDE@UWHEALTH.ORG

Editor: Benjamin M. Marlin and Suchi Saria

Abstract

Predicting breast cancer risk has long been a goal of medical research in the pursuit of precision medicine. The goal of this study is to develop novel penalized methods to improve breast cancer risk prediction by leveraging structure information in electronic health

records. We conducted a retrospective case-control study, garnering 49 mammography descriptors and 77 high-frequency/low-penetrance single-nucleotide polymorphisms (SNPs) from an existing personalized medicine data repository. Structured mammography reports and breast imaging features have long been part of a standard electronic health record (EHR), and genetic markers likely will be in the near future. Lasso and its variants are widely used approaches to integrated learning and feature selection, and our methodological contribution is to incorporate the dependence structure among the features into these approaches. More specifically, we propose a new methodology by combining group penalty and l^p ($1 \leq p \leq 2$) fusion penalty to improve breast cancer risk prediction, taking into account structure information in mammography descriptors and SNPs. We demonstrate that our method provides benefits that are both statistically significant and potentially significant to people's lives.

Keywords: structure information, breast cancer risk prediction, mammography descriptors, genetic variants, personalized medicine

1. Introduction

Breast cancer is the most common non-skin malignancy affecting women, with approximately 1.67 million cases diagnosed annually worldwide (Ferlay et al., 2013). If an individual's risk of breast cancer could be predicted, then screening, prevention, and treatment strategies could be targeted toward those women to maximize survival benefit and minimize harm. Risk prediction models are important tools to improve breast cancer care by leveraging multi-dimensional electronic health data. Traditional breast cancer risk prediction models use demographic risk factors to estimate breast cancer risk, but they demonstrate only limited discriminatory power. In clinical practice, mammography is the most common breast cancer screening test, and the only imaging modality supported by randomized trials demonstrating reduction in mortality rate. However, its effectiveness is not universally accepted (Freedman et al., 2004). Recent advances in genome-wide association studies (GWAS) have revitalized the quest for genetic variants (single-nucleotide polymorphisms—SNPs) in risk prediction. However, the optimism of these studies has been tempered by disappointment and caution (Gail, 2008, 2009; Wacholder et al., 2010).

Although many breast cancer risk prediction models have been developed, current applications of these models are inadequate in the following respects: (1) due to the rare occurrence of breast cancer, many seemingly 'large' studies have small effective sample size to adequately model a large number of variables; (2) even for large studies, investigators often fail to systematically model risk factor interactions to avoid overly complicated models which are hard to interpret; and (3) they do not take available structure information into consideration. For example, there are five descriptors for mass margins in mammogram: circumscribed, microlobulated, obscured, indistinct, and spiculated, with an order of increasing probability of malignancy. However, few models utilize this structure information (group structure and dependence structure) to improve predictive performance. The quest for novel breast cancer risk prediction models is motivated to address these shortcomings.

In this paper, we propose to develop novel penalized methods to improve breast cancer risk prediction by incorporating unique structure information embedded in electronic health record data. Regularization is a common technique used in regression and classification problems. The lasso (Tibshirani, 1996) is one of the most popular penalized method and

SNP	Chr	SNP	Chr
rs616488	1	rs11814448	10
rs11249433	1	rs7072776	10
rs1550623	2	rs7904519	10
rs16857609	2	rs2981582	10
rs2016394	2	rs10995190	10
rs4849887	2	rs2380205	10
rs1045485	2	rs2981579	10
rs13387042	2	rs704010	10
rs17468277	2	rs11820646	11
rs4666451	2	rs3903072	11
rs12493607	3	rs3817198	11
rs6762644	3	rs2107425	11
rs4973768	3	rs614367	11
rs6828523	4	rs12422552	12
rs9790517	4	rs17356907	12
rs10472076	5	rs6220	12
rs1353747	5	rs10771399	12
rs1432679	5	rs1292011	12
rs10941679	5	rs11571833	13
rs889312	5	rs2236007	14
rs30099	5	rs2588809	14
rs981782	5	rs941764	14
rs10069690	5	rs999737	14
rs11242675	6	rs13329835	16
rs204247	6	rs17817449	16
rs2046210	6	rs3803662	16
rs2180341	6	rs12443621	16
rs17530068	6	rs8051542	16
rs3757318	6	rs6504950	17
rs720475	7	rs1436904	18
rs11780156	8	rs527616	18
rs2943559	8	rs3760982	19
rs6472903	8	rs4808801	19
rs9693444	8	rs8170	19
rs13281615	8	rs2284378	20
rs10759243	9	rs2823093	21
rs1011970	9	rs132390	22
rs865686	9	rs6001930	22
rs11199914	10		

Table 2: The 77 SNPs identified to be associated with breast cancer

2.1.3 GENETIC VARIANTS

We decided to focus on high-frequency/low-penetrance SNPs that affect breast cancer risk as opposed to low frequency SNPs with high penetrance or intermediate penetrance. We consolidated a list of 77 common genetic variants (Table 2) which were identified by recent large-scale GWAS studies or used to generate published predictive models (Liu et al., 2014). The list included 41 SNPs identified by COGS through a meta-analysis of 9 GWAS studies (Michailidou et al., 2013). Recently, a similar set of 77 breast cancer-associated SNPs is also studied for risk prediction (Mavaddat et al., 2015).

2.2 Logistic Regression

Assume that we have independent and identical distributed subjects $\{(x_i, y_i)\}_{i=1}^n$, where the explanatory variable $X \in \mathcal{R}^d$ and the binary response variable $Y \in \{-1, 1\}$. Note that the conditional probability $\eta(x) = \mathbb{P}(Y = 1|X = x)$ plays an important role in the classification problem. Denote $x_i = (x_{i1}, \dots, x_{id})^T$, and linear logistic regression model is defined by

$$\log \frac{\eta(x_i)}{1 - \eta(x_i)} = x_i^T \beta, \quad i = 1, \dots, n,$$

where $\beta = (\beta_1, \dots, \beta_d)^T$ is the slope parameter. And the logistic regression estimator $\hat{\beta}$ is given by the minimizer of the negative log-likelihood function

$$L(\beta) = \frac{1}{n} \sum_{i=1}^n \log(1 + \exp(-y_i \cdot x_i^T \beta)). \quad (1)$$

With $\hat{\beta}$, we then estimate the conditional probability $\eta(x_i)$ by

$$\hat{\eta}(x_i) = \frac{\exp(x_i^T \hat{\beta})}{1 + \exp(x_i^T \hat{\beta})} = \frac{1}{1 + \exp(-x_i^T \hat{\beta})}.$$

Then we should predict $y_i = 1$ if $\hat{\eta}(x_i) \geq 0.5$ and $y_i = -1$ if $\hat{\eta}(x_i) < 0.5$.

2.3 Group Penalty and ℓ^p Fusion Penalty

Note that there exist natural group structure and dependence structure in mammography features (Figure 1), which allows us to include the structure information into our risk prediction models directly. For genetic variants, group structures also exist (Liu et al., 2012, 2013). In this paper, we apply hierarchical clustering to cluster the 77 SNPs based on their dissimilarity matrix obtained by computing Spearman's correlation or Hamming distance among them. More details are provided in Section 2.5.

Suppose that d features are divided into G groups with d_g the number of features in group g . Define $\beta_g \in \mathbb{R}^{d_g}$ to be the corresponding coefficient vector in group g . The group lasso logistic regression (Meier et al., 2008) is defined as the following optimization problem

$$\min_{\beta \in \mathbb{R}^d} \left\{ L(\beta) + \lambda_1 \sum_{g=1}^G \sqrt{d_g} \|\beta_g\|_2 \right\},$$

where $L(\beta)$ is defined by (1) and $\lambda_1 \geq 0$ is the tuning parameter. It includes lasso as a special case with $G = d$.

The fact that there exist dependence structure within each mammography feature group and each SNP group encourages us to propose the following novel method by combining group lasso logistic regression and ℓ^{p^*} fusion penalty.

$$\min_{\beta \in \mathbb{R}^d} \left\{ L(\beta) + \sum_{g=1}^G \left(\lambda_1 \sqrt{d_g} \|\beta_g\|_2 + \lambda_2 \|D_g \beta_g\|_{p^*} \right) \right\}, \quad (2)$$

where D_g is a $(d_g - 1) \times d_g$ sparse matrix with only $D[i, j] = 1$ and $D[i, i+1] = -1$, $\lambda_2 \geq 0$ is the tuning parameter, and $1 \leq p \leq 2$ is the shrinkage parameter.

Moreover, if the within-group dependence structures are different for groups $\{1, \dots, G_1\}$ and $\{G_1 + 1, \dots, G\}$, we can split the ℓ^{p^*} fusion penalty into two parts as

$$\min_{\beta \in \mathbb{R}^d} \left\{ L(\beta) + \lambda_1 \sum_{g=1}^G \sqrt{d_g} \|\beta_g\|_2 + \lambda_2 \left(\sum_{g=1}^{G_1} \|D_g \beta_g\|_{p_1}^{p_1} + \sum_{g=G_1+1}^G \|D_g \beta_g\|_{p_2}^{p_2} \right) \right\}, \quad (3)$$

where $1 \leq p_1, p_2 \leq 2$ are selected based on cross validation.

The novelty of our method compared to previous works is three-fold: First, it includes within-group fusion penalty in the model and makes the coefficients of features in the same group close to each other, which reflects the dependence structure of features and improves the risk prediction; Second, in breast cancer risk prediction, we find that the dependence structures are different for mammography features and SNPs, which are actually two different views of the same data. And the utilization of method (3) will improve the predictive performance further; At last, we find that genetic variants improve risk prediction on mammography features, which provides some insight regarding personalized breast cancer diagnosis.

2.4 Computational Algorithms

Many algorithms have been proposed in the literatures to solve the logistic regression with fused lasso regularization (Lin, 2015; Yu et al., 2015). In this subsection we adopt the fast iterative shrinkage thresholding algorithm (Beck and Teboulle, 2009) to solve (2) as

$$\beta^{k+1} = \arg \min_{\beta \in \mathbb{R}^d} L(\beta^k) + \langle \beta - \beta^k, \nabla L(\beta^k) \rangle + \frac{\tau}{2} \|\beta - \beta^k\|_2^2 + \sum_{g=1}^G \left(\lambda_1 \sqrt{d_g} \|\beta_g\|_2 + \lambda_2 \|D_g \beta_g\|_{p^*} \right)$$

with $\beta = (\beta_1, \dots, \beta_d)^T$ and $\tau > 0$ the Lipschitz constant of $L(\cdot)$.

And the iteration step is equivalent to solving

$$\min_{\beta \in \mathbb{R}^d} \left\{ \frac{1}{2} \|\beta - (\beta^k - \frac{1}{\tau} \nabla L(\beta^k))\|_2^2 + \sum_{g=1}^G \left(\frac{\lambda_1 \sqrt{d_g}}{\tau} \|\beta_g\|_2 + \frac{\lambda_2}{\tau} \|D_g \beta_g\|_{p^*} \right) \right\}. \quad (4)$$

Therefore, it suffices to solve the following optimization problem within each group

$$\min_{\beta_g \in \mathbb{R}^{d_g}} \left\{ \frac{1}{2} \|\beta_g - z\|_2^2 + \rho_1 \|\beta_g\|_2 + \rho_2 \|D_g \beta_g\|_{p^*} \right\}, \quad (5)$$

where $z = \beta_g^k - \frac{1}{\tau} \nabla L(\beta_g^k)$, $\rho_1 = \frac{\lambda_1 \sqrt{d_g}}{\tau}$ and $\rho_2 = \frac{\lambda_2}{\tau}$.

The proximity operator (Polson et al., 2015) of a function f is defined as

$$P_f(z) = \arg \min_t \left\{ \frac{1}{2} \|t - z\|^2 + \lambda f(t) \right\}.$$

- For $f(t) = |t|$ and $z \in \mathbb{R}$, $P_f(z) := S_1(z, \lambda) = \text{sign}(z) \max\{|z| - \lambda, 0\}$, which is also called soft threshold operator.
- For $f(t) = |t|^p$ with $1 < p \leq 2$ and $z \in \mathbb{R}$, $P_f(z) := S_p(z, \lambda) = \text{sign}(z) \xi$, where ξ is the unique nonnegative solution to $\xi + p \lambda \xi^{p-1} = |z|$. In particular, we have $S_2(z, \lambda) = \frac{z + \sqrt{z^2 + 8\lambda^2}}{2\sqrt{2}}$, $S_{3/2}(z, \lambda) = z + 9\lambda^2 \text{sign}(z) (1 - \sqrt{1 + 16|z|/(9\lambda^2)})/8$ and $S_{4/3}(z, \lambda) = z + \frac{4\lambda}{3\sqrt{3}} ((\chi - z)^{1/3} - (\chi + z)^{1/3})$ with $\chi = \sqrt{z^2 + 256\lambda^3/729}$.
- For $f(t) = \|t\|_2$ and $z \in \mathbb{R}^d$, $P_f(z) := S_{2,1}(z, \lambda) = \max\{1 - \frac{\lambda}{\|z\|_2}, 0\} * z$.

With the help of these proximity operators and Bregman splitting algorithm (Ye and Xie, 2011), we can solve (5) by iteratively solving the following procedures:

$$\begin{cases} \beta_g^{k+1} = \arg \min_{\beta_g} \frac{1}{2} \|\beta_g - z\|_2^2 + \langle u^k, \beta_g - a^k \rangle + \langle v^k, D_g \beta_g - b^k \rangle \\ \quad + \frac{\mu}{2} \|\beta_g - a^k\|_2^2 + \frac{\mu}{2} \|D_g \beta_g - b^k\|_2^2 \\ a^{k+1} = \arg \min_{a} \rho_1 \|a\|_2 + \langle u^k, \beta^{k+1} - a \rangle + \frac{\mu}{2} \|\beta^{k+1} - a\|_2^2 \\ b^{k+1} = \arg \min_b \rho_2 \|b\|_{p^*} + \langle v^k, D_g \beta^{k+1} - b \rangle + \frac{\mu}{2} \|D_g \beta^{k+1} - b\|_2^2 \\ u^{k+1} = u^k + \mu (\beta^{k+1} - a^{k+1}) \\ v^{k+1} = v^k + \mu (D_g \beta^{k+1} - b^{k+1}) \end{cases}$$

where μ acts like a step size in this algorithm.

Remark 1 The minimization over β , a and b can all be solved in closed form.

- $\beta^{k+1} = [\mu + 1]I + \mu D_g^T D_g]^{-1} [z + \mu(a^k - u^k/\mu) + \mu D_g^T (b^k - v^k/\mu)]$
- $a^{k+1} = S_{2,1}(\beta^{k+1} + u^k/\mu, \rho_1/\mu)$
- $b^{k+1} = S_{p^*}(\beta^{k+1} + v^k/\mu, \rho_2/\mu)$

Note that $(\mu + 1)I + \mu D_g^T D_g$ is a triagonal positive definite matrix.

Remark 2 For $p = 1$, we can solve (5) more efficiently by the algorithm proposed in Zhou et al. (2012) based on the fact

$$P_{\|\cdot\|_2 + \|D_g(\cdot)\|_1} = P_{\|\cdot\|_2} \circ P_{\|D_g(\cdot)\|_1}.$$

However, we cannot show this equation for $1 < p \leq 2$.

Remark 3 For $p = 2$, since $\|\cdot\|_2^2$ is Lipschitz continuous, we can rewrite (4) as

$$\min_{\beta \in \mathbb{R}^d} \left\{ \frac{1}{2} \left\| \beta - \left(\beta^k - \frac{1}{\bar{\tau}} (\nabla L(\beta^k) + 2\lambda_2 \sum_{g=1}^G D_g^T D_g \beta^k) \right) \right\|_2^2 + \sum_{g=1}^G \frac{\lambda_1 \sqrt{d_g}}{\bar{\tau}} \|\beta_g\|_2 \right\},$$

where $\bar{\tau}$ is the Lipschitz constant of $L(\beta) + \lambda_2 \sum_{g=1}^G \|D_g \beta_g\|_2^2$. Then we can solve it efficiently via the proximity operator of $\|\cdot\|_2$.

2.5 Study Design and Statistical Analysis

We apply the ℓ^p fused group lasso logistic regression algorithm to the Marshfield breast cancer data set. There are 11 groups for 49 mammography features (Figure 1). For SNPs, we compute the Hamming distances (Wang et al., 2015) of 77 SNPs to get the dissimilarity matrix and then apply hierarchical clustering to obtain 10 groups.

We built three prediction models based on different sets of risk factors: the Mammo model developed by using mammography features only, the SNP77 model developed by using 77 SNPs only, and the Combined model developed by using both mammography features and 77 SNPs. We furthermore apply five methods for each model: logistic regression (LR), lasso in logistic regression (LR+Lasso), ℓ^p fused lasso logistic regression (LR+fusedLasso), group lasso logistic regression (LR+groupLasso), and ℓ^p fused group lasso logistic regression (LR+Structure).

The ℓ^p fused group lasso logistic regression method has several parameters. For the tuning parameters λ_1 and λ_2 , we let them vary among a given set of values, and the shrinkage parameter p (or p_1 and p_2) among $\{1, 4/3, 3/2, 2\}$. Each combination of these parameters is evaluated using stratified 5-fold cross-validation, and AUC (the area under the receiver operating characteristic (ROC) curve) is used as the performance measure. All 738 samples are randomly partitioned into five equal sized folds with approximately equal proportions of cases and controls. In each iteration (totally five iterations), four folds are used as training set and the rest one as validation set to compute AUC. And the parameters with the best average AUC are selected. At last we repeat this process ten times and report the average AUC. We obtain p-value by performing two-tailed two-sample t-test when we compare AUCs.

3. Experimental Results

In this section, we demonstrate the performance of the ℓ^p fused group lasso logistic regression method from three aspects: the significant improvement of AUCs by considering the structure information, the predictive performance under different p (or p_1 and p_2), and the detected important mammography features and SNPs.

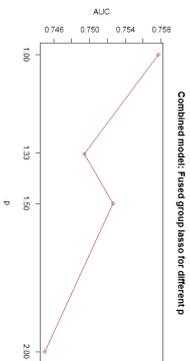
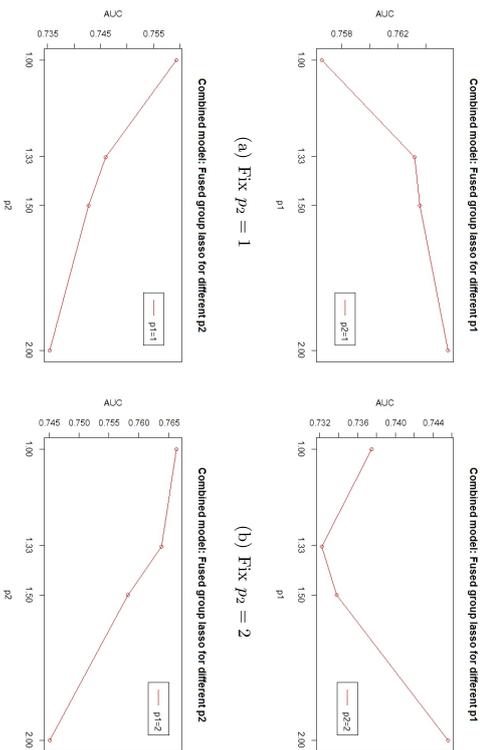
3.1 Performance of Fused Group Lasso

The result is summarized in Table 3.

Models/Methods	LR	Lasso	fusedLasso	groupLasso	Structure	p-value
Mammo	0.700	0.710	0.710	0.716	0.723	< 0.001
SNP77	0.590	0.598	0.676	0.614	0.684	< 0.001
Combined	0.697	0.721	0.754	0.727	0.766	< 0.001

Table 3: Predictive performance of three prediction models by using five different methods. The p-values represent the differences between AUCs of LR and LR+Structure.

- 1) The fifth column describes the predictive performance of the three prediction models by considering structure information in the logistic regression method. We find that the predictive performance of the three prediction models has been improved respectively, compared to those described in the first column. For each prediction model, the difference of the predictive performance is significant between LR+Structure and LR (p-value < 0.001), which demonstrates that breast cancer prediction models utilizing structure information can improve risk prediction significantly. We also find that mammography descriptors demonstrate a significantly higher predictive performance than 77 SNPs in terms of AUC (0.723 vs. 0.684, p-value < 0.001). The Combined model demonstrates significant improvement of the prediction performance, compared to the Mammo model (0.766 vs. 0.723, p-value < 0.001).
- 2) The first column describes the predictive performance of the three prediction models by using the logistic regression method. Mammography descriptors demonstrate a significantly higher predictive performance than 77 SNPs in terms of AUC (0.700 vs. 0.590, p-value < 0.001). We find that the difference of predictive performance between the Combined model and the Mammo model is negligible (0.697 vs. 0.700, p-value=0.277).
- 3) The second column describes the predictive performance of the three prediction models by using lasso in the logistic regression method. The predictive performance of the three prediction models has been improved, compared to those without lasso (using logistic regression method only). Mammography descriptors still demonstrate a significantly higher predictive performance than 77 SNPs in terms of AUC (0.710 vs. 0.598, p-value < 0.001). However, the Combined model demonstrates modest improvement of prediction performance, compared to the Mammo model (0.721 vs. 0.710, p-value=0.0057).
- 4) The third and fourth columns describe the predictive performance of the three prediction models by considering group structure or dependence structure in the logistic regression method. For the SNP77 model, fused lasso demonstrates a significantly higher performance than group lasso in terms of AUC (0.676 vs. 0.614, p-value < 0.001). For the Mammo model, group lasso plays a more important role than fused lasso (0.716 vs. 0.710, p-value=0.0073). Moreover, both fused lasso and group lasso demonstrate improved prediction performance compared to lasso.

Figure 2: The AUCs under different values of p by using method (2).Figure 3: The AUCs under different values of p_1 and p_2 by using method (3).

3.2 Performance under Different Values of p

Figure 2 and Figure 3 describe the the pattern of predictive performance for p' fused group lasso logistic regression over the shrinkage parameter p (or p_1 and p_2) in terms of AUC.

1) The Combined model demonstrates a higher predictive performance for $p = 1$ compared to $p = 2$ in terms of AUC (0.757 vs. 0.745, p-value < 0.001), see Figure 2.

2) Figure 3 describes the prediction performance of method (3) under different values of p_1 for mammography descriptors and p_2 for 77 SNPs.

FAN, WU, YUAN, PAGE, LIU, ONG, PEISSIG, AND BURNSIDE

- Fix $p_2 = 1$ or $p_2 = 2$, the fused group lasso with $p_1 = 2$ demonstrates higher predictive performance compared to $p_1 = 1$, see Figure 3(a) and 3(b).
- Fix $p_1 = 1$ or $p_1 = 2$, the predictive performance of the fused group lasso logistic regression decreases as p_2 increases, see Figure 3(c) and 3(d).
- The fused group lasso logistic regression with $p_1 = 2$ and $p_2 = 1$ demonstrates higher predictive performance than $p_1 = p_2 = 1$ (0.766 vs. 0.757, p-value=0.0053) and $p_1 = p_2 = 2$ (0.766 vs. 0.745, p-value < 0.001).

3.3 Important Features Detected by Fused Group Lasso

To take into account both group and dependence structure information in mammography features and SNPs, two penalty terms (group penalty and fusion penalty) are introduced into the logistic regression model. The idea of group penalty is to force the coefficients of features in the same group to be all zero or nonzero in order to achieve the goal of selecting features within a group simultaneously. The idea of fusion penalty is to shrink the successive difference of coefficients of features in the same group in order to take advantage of the dependence structure information. Applying fusion penalty with $p = 1$ tends to result in zero successive difference of coefficients, while $p = 2$ tends to small but nonzero successive difference of coefficients.

From a feature selection point of view, we can get the order of feature groups selected by fused group lasso via choosing the tuning parameters appropriately. We list below the feature groups selected from high to low in terms of predictive performance.

- 1) For mammography descriptors, the following features are predictive of malignancy (from most to least): ‘‘Mass Size’’, ‘‘Mass Margins’’, ‘‘Mass Shape’’, ‘‘Architectural Distortion’’ and ‘‘Mass Palpability’’, consistent with the literature (BI-RADS, 2014).
- 2) For 77 SNPs, three groups are selected in order, see Table 4.

Feature Group	SNPs
Group 1	rs2016394, rs1432679, rs13281615, rs4666451 rs981782, rs1292011, rs1436904, rs527616
Group 2	rs11249433, rs13387042, rs4973768, rs10069690 rs7904519, rs8051542, rs3760982
Group 3	rs2981579, rs2981582

Table 4: SNP groups selected by fused group lasso.

Remark 4 It verifies that ‘‘Mass size’’, ‘‘Mass Margins’’ and ‘‘Mass Shape’’ are the most important mammography descriptors in breast cancer diagnosis. These results are consistent with previous studies about comparing the importance of mammography features and SNPs in breast cancer risk prediction (Wu et al., 2013, 2014).

4. Discussion and Conclusions

This study demonstrates that models utilizing the novel combination of clinically relevant structure and ℓ^p fused group lasso logistic regression can improve breast cancer risk prediction significantly. Our study also shows that both mammography features and SNPs contribute to this improvement.

The structure information of the mammography features is derived from the BI-RADS lexicon, which is used widely in breast imaging practice. Thus, our model would likely be generalizable to other practices. On the other hand, we extracted the structure information of SNPs by computing Hamming distances (Wang et al., 2015). This method may not perform as well in small sample sizes, which may affect our results perhaps making our predictive performance results conservative.

Our methods for SNPs may not take advantage of biological knowledge that currently exists. For example, it may be possible to utilize the biological information available in HapMap (which encodes linkage disequilibrium) to more accurately emulate the patterns or dependence structure of SNPs, as in (Liu et al., 2012). Furthermore, we realize that taking into account more complicated structure information such as graph or tree structure (Sun and Wang, 2012) may further improve predictive performance of risk prediction models. We leave these promising directions for future work.

In conclusion, our results demonstrate that including structure information in the computational methods we test improves breast cancer risk prediction. Our models use diverse breast cancer risk factors including demographics, genetics, and imaging and leverage structure found in a standardized lexicon that is universally captured in electronic health records (EHRs) throughout the US. This information will increasingly be combined in complex ways. Merging imaging features, clinical notes and genetic data with models that accurately predict disease risk has the potential to provide powerful knowledge to practicing physicians.

Acknowledgments

The authors acknowledge the support of the Wisconsin Genomics Initiative, NCI grant R01CA127379-01 and its ARRA supplement R01CA127379-03S1, NIGMS grant R01GM097618-01, NLM grant R01LM011028-01, NIEHS grant 5R01ES017400-03, NIH grant 1U54AI117924-01, NIH grant K24CA194251 and NSF FRG grant DMS-1265202. We also acknowledge support from the Clinical and Translational Science Award (CTSA) program through the NIH National Center for Advancing Translational Sciences (NCATS) grant UL1TR000427 and the University of Wisconsin Carbone Cancer Center Cancer Support Grant P30CA014520. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH. We thank the anonymous reviewers for their valuable comments and suggestions.

References

A. Beck and M. Tebouille. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183-202, 2009.

Breast Imaging Reporting And Data System (BI-RADS). 5th ed. Reston VA: American College of Radiology, 2014.

E. S. Burnside, J. Liu, Y. Wu, A. A. Onitilo, C. A. McCarty, C. D. Page, P. Peissig, A. Trentham-Dietz, T. Kitchner, J. Fan, and M. Yuan. Comparing Mammography Abnormality Features and Genetic Variants in the Prediction of Breast Cancer in Women Recommended for Breast Biopsy. *Academic Radiology*, 23(1):62-9, 2016.

J. Ferlay, I. Soerjomataram, M. Ervik, et al. *GLOBOCAN 2012 cancer incidence and mortality worldwide: IARC cancerbase No. 11*. Lyon, France: International Agency for Research on Cancer, 2013.

D. Freedman, D. Petitti, and J. Robins. On the efficacy of screening for breast cancer. *Int J Epidemiol.*, 33(1):43-55, 2004.

M. Gail. Discriminatory accuracy from single-nucleotide polymorphisms in models to predict breast cancer risk. *J Natl Cancer Inst.*, 100(14):1037-41, 2008.

M. Gail. Value of adding single-nucleotide polymorphism genotypes to a breast cancer risk model. *J Natl Cancer Inst.*, 101(13):959-63, 2009.

T. Lin, S. Ma, and S. Zhang. An extragradient-based alternating direction method for convex minimization. *Found Comput Math*, DOI 10.1007/s10208-015-9282-8, 2015.

J. Liu, J. Huang, S. Ma, and K. Wang. Incorporating group correlations in genome-wide association studies using smoothed group lasso. *Biostatistics*, 14:205-219, 2013.

J. Liu, C. D. Page, P. L. Peissig, et al. New genetic variants improve personalized breast cancer diagnosis. *AMIA Summit on Translational Bioinformatics (AMIA-TBI)*, 2014.

J. Liu, C. Zhang, C. McCarty, P. L. Peissig, E. S. Burnside, and D. Page. Graphical-model based multiple testing under dependence, with applications to genome-wide association studies. In *Proceedings of the 28th conference on uncertainty in artificial intelligence*, 2012.

S. Ma, X. Song, and J. Huang. Supervised group Lasso with applications to microarray data analysis. *BMC bioinformatics*, 8:60, 2007.

N. Mavaddat, et al. Prediction of breast cancer risk based on profiling with common genetic variants. *J Natl Cancer Inst.*, 107(5):djv036, 2015.

C. McCarty, R. Wilke, P. Giampietro, S. Westbrook, and M. Caldwell. Marshfield Clinic Personalized Medicine Research Project (PMRP): design, methods and recruitment for a large population-based biobank. *Personalized Med.*, 2(1):49-79, 2005.

L. Meier, S. Van De Geer, and P. Bohlmann. The Group Lasso for logistic regression. *J. R. Statist. Soc. B*, 70:53-71, 2008.

K. Michailidou, P. Hall, A. Gonzalez-Neira, et al. Large-scale genotyping identifies 41 new loci associated with breast cancer risk. *Nat Genet.*, 45(4):353-61, 2013.

- H. Nassif, R. Woods, E. S. Burnside, M. Ayvaci, J. Shavlik, C. D. Page. Information extraction for clinical data mining: a mammography case study. *IEEE International Conference on Data Mining Workshops*, 2009.
- N. G. Polson, J. G. Scott, and B. T. Willard. Proximal Algorithms in Statistics and Machine Learning. *Statistical Science*, 30(4):559-581, 2015.
- H. Sun and S. Wang. Penalized logistic regression for high-dimensional DNA methylation data with case-control studies. *Bioinformatics*, 28(10):1368-1375, 2012.
- R. Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Statist. Soc. B*, 58:267-288, 1996.
- R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *J. R. Statist. Soc. B*, 67:91-108, 2005.
- R. Tibshirani and P. Wang. Spatial smoothing and hot spot detection for CGH data using the fused lasso. *Biostatistics*, 9(1):18-29, 2008.
- S. Wacholder, P. Hartge, R. Prentice, et al. Performance of common genetic variants in breast-cancer risk models. *N Engl J Med.*, 362(11):986-93, 2010.
- C. Wang, W. H. Kao, and C. K. Hsiao. Using Hamming distance as information for SNP-sets clustering and testing in disease association studies. *PLoS One*, 10(8), 2015.
- Y. Wu, O. Alagoz, M. Ayvaci, A. M. del Rio, D. J. Vanness, R. Woods, and E. S. Burnside. A comprehensive methodology for determining the most informative mammographic features. *J Digit Imaging*, 26(5):941-947, 2013.
- Y. Wu, J. Liu, C. D. Page, P. L. Peisig, C. A. McCarty, A. A. Onitilo, and E. S. Burnside. Comparing the Value of Mammographic Features and Genetic Variants in Breast Cancer Risk Prediction. *AMIA Annu Symp Proc.*, 1228-1237, 2014.
- G. Ye and X. Xie. Split Bregman method for large scale fused Lasso. *Computational Statistics and Data Analysis*, 55(4):1552-1569, 2011.
- D. Yu, S. Lee, W. Lee, S. Kim, J. Lim, and S. Kwon. Classification of spectral data using fused lasso logistic regression. *Chemometrics and Intelligent Laboratory Systems*, 142:70-77, 2015.
- M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J. R. Statist. Soc. B*, 68:49-67, 2006.
- J. Zhou, J. Liu, V. A. Narayan, and J. Ye. Modeling disease progression via fused sparse group lasso. In *KDD*, pages 1095-1103, 2012.

Gains and Losses are Fundamentally Different in Regret Minimization: The Sparse Case

Joon Kwon

Institut de mathématiques de Jussieu
Université Pierre-et-Marie-Curie
4, place Jussieu
75252 Paris Cedex 05, France

JOON.KWON@ENS-LYON.ORG

Vianney Perchet

Centre de recherche en économie et statistique
École nationale de la statistique et de l'administration économique
3, avenue Pierre Larousse
92245 Malakoff Cedex, France

VIANNEY.PERCHET@NORMALESUP.ORG

Editor: Alexander Rakhlin

Abstract

We demonstrate that, in the classical non-stochastic regret minimization problem with d decisions, gains and losses to be respectively maximized or minimized are fundamentally different. Indeed, by considering the additional sparsity assumption (at each stage, at most s decisions incur a nonzero outcome), we derive optimal regret bounds of different orders. Specifically, with gains, we obtain an optimal regret guarantee after T stages of order $\sqrt{T \log s}$, so the classical dependency in the dimension is replaced by the sparsity size.

With losses, we provide matching upper and lower bounds of order $\sqrt{T s \log(d)/d}$, which is decreasing in d . Eventually, we also study the bandit setting, and obtain an upper bound of order $\sqrt{T s \log(d/s)}$ when outcomes are losses. This bound is proven to be optimal up to the logarithmic factor $\sqrt{\log(d/s)}$.

Keywords: regret minimization, bandit, sparsity

1. Introduction

We consider the classical problem of regret minimization (Hannan, 1957) that has been well developed during the last decade (Cesa-Bianchi and Lugosi, 2006; Rakhlin and Tewari, 2008; Bubeck, 2011; Shalev-Shwartz, 2011; Hazan, 2012; Bubeck and Cesa-Bianchi, 2012). We recall that in this sequential decision problem, a decision maker (or agent, player, algorithm, strategy, policy, depending on the context) chooses at each stage a decision in a finite set (that we write as $[d] := \{1, \dots, d\}$) and obtains as an *outcome* a real number in $[0, 1]$. We specifically chose the word *outcome*, as opposed to *gain* or *loss*, as our results show that there exists a fundamental discrepancy between these two concepts.

The criterion used to evaluate the policy of the decision maker is the *regret*, i.e., the difference between the cumulative performance of the best stationary policy (that always picks a given action $i \in [d]$) and the cumulative performance of the policy of the decision maker.

We focus here on the *non-stochastic* framework, where no assumption (apart from boundedness) is made on the sequence of possible outcomes. In particular, they are not i.i.d. and we can even assume, as usual, that they depend on the past choices of the decision maker. This broad setup, sometimes referred to as *individual sequences* (since a policy must be good against *any* sequence of possible outcomes) incorporates prediction with expert advice (Cesa-Bianchi and Lugosi, 2006), data with time-evolving laws, etc. Perhaps the most fundamental results in this setup are the upper bound of order $\sqrt{T \log d}$ achieved by the Exponential Weight Algorithm (Littlestone and Warmuth, 1994; Vovk, 1990; Cesa-Bianchi, 1997; Auer et al., 2002) and the asymptotic lower bound of the same order (Cesa-Bianchi et al., 1997). This general bound is the same whether outcomes are gains in $[0, 1]$ (in which case, the objective is to maximize the cumulative sum of gains) or losses in $[0, 1]$ (where the decision maker aims at minimizing the cumulative sum). Indeed, a loss ℓ can easily be turned into gain g by defining $g := 1 - \ell$, the regret being invariant under this transformation.

This idea does not apply anymore with structural assumption. For instance, consider the framework where the outcomes are limited to *s-sparse vectors*, i.e. vectors that have at most s nonzero coordinates. The coordinates which are nonzero may change arbitrarily over time. In this framework, the aforementioned transformation does not preserve the sparsity assumption. Indeed, if (ℓ_1, \dots, ℓ_d) is a s -sparse loss vector, the corresponding gain vector $(1 - \ell_1, \dots, 1 - \ell_d)$ may even have full support. Consequently, results for loss vectors do not apply directly to sparse gains, and vice versa. It turns out that both setups are fundamentally different.

The sparsity assumption is actually quite natural in learning and have also received some attention in online learning (Gerchinovitz, 2013; Carpenter and Munos, 2012; Abbasi-Yadkori et al., 2012; Djolonga et al., 2013). In the case of gains, it reflects the fact that the problem has some hidden structure and that many options are irrelevant. For instance, in the canonical click-through-rate example, a website displays an ad and gets rewarded if the user clicks on it; we can safely assume that there are only a small number of ads on which a user would click.

The sparse scenario can also be seen through the scope of prediction with experts. Given a finite set of expert, we call the *winner of a stage* the expert with the highest revenue (or the smallest loss); ties are broken arbitrarily. And the objective would be to win as many stages as possible. The s -sparse setting would represent the case where s experts are designated as winners (or, non-loser) at each stage.

In the case of losses, the sparsity assumption is motivated by situations where rare failures might happen at each stage, and the decision maker wants to avoid them. For instance, in network routing problems, it could be assumed that only a small number of paths would lose packets as a result of a single, rare, server failure. Or a learner could have access to a finite number of classification algorithms that perform ideally most of the time; unfortunately, some of them makes mistakes on some examples and the learner would like to prevent that. The general setup is therefore a number of algorithms/experts/actions that mostly perform well (i.e., find the correct path, classify correctly, optimize correctly some target function, etc.); however, at each time instance, there are rare mistakes/accidents and the objective would be to find the action/algorithm that has the smallest number (or probability in the stochastic case) of failures.

	Full information		Bandit	
Upper bound	Gains	Losses	Gains	Losses
Lower bound	$\sqrt{T \log s}$	$\sqrt{T s \log \frac{d}{d}}$	$\sqrt{T} d$	$\sqrt{T s \log \frac{d}{s}}$
			$\sqrt{T} s$	$\sqrt{T} s$

Figure 1: Summary of upper and lower bounds.

1.1 Summary of Results

We investigate regret minimization scenarios both when outcomes are gains on the one hand, and losses on the other hand. We recall that our objectives are to prove that they are fundamentally different by exhibiting rates of convergence of different order.

When outcomes are gains, we construct an algorithm based on the Online Mirror Descent family (Shalev-Shwartz, 2007, 2011; Bubeck, 2011). By choosing a regularizer based on the ℓ^p norm, and then tuning the parameter p as a function of s , we get in Theorem 2 a regret bound of order $\sqrt{T \log s}$, which has the interesting property of being independent of the number of decisions d . This bound is trivially optimal, up to the constant.

If outcomes are losses instead of gains, although the previous analysis remains valid, a much better bound can be obtained. We build upon a regret bound for the Exponential Weight Algorithm (Littlestone and Warmuth, 1994; Freund and Schapire, 1997) and we manage to get in Theorem 4 a regret bound of order $\sqrt{T s \log \frac{d}{d}}$, which is *decreasing* in d , for a given s . A nontrivial matching lower bound is established in Theorem 6.

Both of these algorithms need to be tuned as a function of s . In Theorem 9 and Theorem 10, we construct algorithms which essentially achieve the same regret bounds without prior knowledge of s , by adapting over time to the sparsity level of past outcome vectors, using an adapted version of the doubling trick.

Finally, we investigate the bandit setting, where the only feedback available to the decision maker is the outcome of his decisions (and, not the outcome of all possible decisions). In the case of losses we obtain in Theorem 11 an upper bound of order $\sqrt{T s \log(d/s)}$, using the Greedy Online Mirror Descent family of algorithms (Audibert and Bubeck, 2009; Audibert et al., 2013; Bubeck, 2011). This bound is proven to be optimal up to a logarithmic factor, as Theorem 13 establishes a lower bound of order $\sqrt{T} s$.

The rates of convergence achieved by our algorithms are summarized in Figure 1.

1.2 General Model and Notation

We recall the classical non-stochastic regret minimization problem. At each time instance $t \geq 1$, the decision maker chooses a decision d_t in the finite set $[d] = \{1, \dots, d\}$, possibly at random, according to $x_t \in \Delta_d$, where

$$\Delta_d = \left\{ x = (x^{(1)}, \dots, x^{(d)}) \in \mathbb{R}_+^d \mid \sum_{i=1}^d x^{(i)} = 1 \right\}$$

is the set of probability distributions over $[d]$. Nature then reveals an outcome vector $\omega_t \in [0, 1]^d$ and the decision maker receives $\omega_t^{(d_t)} \in [0, 1]$. As outcomes are bounded, we can easily replace $\omega_t^{(d_t)}$ by its expectation that we denote by $\langle \omega_t, x_t \rangle$. Indeed, Hoeffding-Azuma concentration inequality will imply that all the results we will state in expectation hold with high probability.

Given a time horizon $T \geq 1$, the objective of the decision maker is to minimize his regret, whose definition depends on whether outcomes are *gains* or *losses*. In the case of gains (resp. losses), the notation ω_t is then changed to g_t (resp. ℓ_t) and the regret is:

$$R_T = \max_{i \in [d]} \sum_{t=1}^T g_t^{(i)} - \sum_{t=1}^T \langle g_t, x_t \rangle \quad (\text{resp. } R_T = \sum_{t=1}^T \langle \ell_t, x_t \rangle - \min_{k \in [d]} \sum_{t=1}^T \ell_t^{(k)}).$$

In both cases, the well-known Exponential Weight Algorithm guarantees a bound on the regret of order $\sqrt{T \log d}$. Moreover, this bound cannot be improved in general as it matches a lower bound.

We shall consider an additional structural assumption on the outcomes, namely that ω_t is s -sparse in the sense that $\|\omega_t\|_0 \leq s$, i.e., the number of nonzero components of ω_t is less than s , where s is a fixed known parameter. The set of components which are nonzero is not fixed nor known, and may change arbitrarily over time.

We aim at proving that it is then possible to drastically improve the previously mentioned guarantee of order $\sqrt{T \log d}$ and that losses and gains are two fundamentally different settings with minimax regrets of different orders.

2. When Outcomes are Gains to be Maximized

2.1 Online Mirror Descent Algorithms

We quickly present the general Online Mirror Descent algorithm (Shalev-Shwartz, 2011; Bubeck, 2011; Bubeck and Cesa-Bianchi, 2012; Kwon and Mertikopoulos, 2014) and state the regret bound it incurs; it will be used as a key element in Theorem 2.

A convex function $h : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{+\infty\}$ is called a *regularizer* on Δ_d if h is strictly convex and continuous on its domain Δ_h , and $h(x) = +\infty$ outside Δ_h . Denote $\delta_h = \max_{\Delta_d} h - \min_{\Delta_h} h$ and $h^* : \mathbb{R}^d \rightarrow \mathbb{R}^d$ the Legendre-Fenchel transform of h :

$$h^*(y) = \sup_{x \in \mathbb{R}^d} \{ \langle y, x \rangle - h(x) \}, \quad y \in \mathbb{R}^d,$$

which is differentiable since h is strictly convex. For all $y \in \mathbb{R}^d$, it holds that $\nabla h^*(y) \in \Delta_d$. Let $\eta \in \mathbb{R}$ be a parameter to be tuned. The Online Mirror Descent Algorithm associated with the regularizer h and parameter η is defined by:

$$x_t = \nabla h^* \left(\eta \sum_{k=1}^{t-1} \omega_k \right), \quad t \geq 1,$$

where $\omega_t \in [0, 1]^d$ denote the vector of outcomes and x_t the probability distribution chosen at stage t . The specific choice $h(x) = \sum_{i=1}^d x^{(i)} \log x^{(i)}$ for $x = (x^{(1)}, \dots, x^{(d)}) \in \Delta_d$ (and

$h(x) = +\infty$ otherwise) gives the celebrated Exponential Weight Algorithm, which can be written explicitly, component by component:

$$x_t^{(i)} = \frac{\exp\left(\eta \sum_{k=1}^{t-1} \omega_k^{(i)}\right)}{\sum_{j=1}^d \exp\left(\eta \sum_{k=1}^{t-1} \omega_k^{(j)}\right)}, \quad t \geq 1, i \in [d].$$

The following general regret guarantee for strongly convex regularizers is expressed in terms of the dual norm $\|\cdot\|_*$ of $\|\cdot\|$. Similar statements have appeared in e.g. (Shalev-Shwartz, 2011, Theorem 2.21), (Bubeck and Cesa-Bianchi, 2012, Theorem 5.6) and (Kwon and Mertikopoulos, 2014, Theorem 5.1).

Theorem 1 *Let $K > 0$ and assume h to be K -strongly convex with respect to a norm $\|\cdot\|$. Then, for any sequence of outcome vectors $(\omega_t)_{t \geq 1}$ in \mathbb{R}^d , the Online Mirror Descent strategy associated with h and η (with $\eta > 0$ in cases of gains and $\eta < 0$ in cases of losses) guarantees, for $T \geq 1$, the following regret bound:*

$$R_T \leq \frac{\delta_h}{|\eta|} + \frac{|\eta|}{2K} \sum_{t=1}^T \|\omega_t\|_*^2.$$

2.2 Upper Bound on the Regret

We first assume $s \geq 2$. Let $p \in (1, 2]$ and define the following regularizer:

$$h_p(x) = \begin{cases} \frac{1}{2} \|x\|_p^2 & \text{if } x \in \Delta_d \\ +\infty & \text{otherwise.} \end{cases}$$

One can easily check that h_p is indeed a regularizer on Δ_d and that $\delta_{h_p} \leq 1/2$. Moreover, it is $(p-1)$ -strongly convex with respect to $\|\cdot\|_p$: see (Bubeck, 2011, Lemma 5.7) or (Kakade et al., 2012, Lemma 9).

We can now state our first result, the general upper bound on regret when outcomes are s -sparse gains.

Theorem 2 *Let $\eta > 0$ and $s \geq 3$. Against all sequences of s -sparse gain vectors g_t , i.e., $g_t \in [0, 1]^d$ and $\|g_t\|_0 \leq s$, the Online Mirror Descent algorithm associated with regularizer h_p and parameter η guarantees:*

$$R_T \leq \frac{1}{2\eta} + \frac{\eta T s^2/q}{2(p-1)},$$

where $1/p + 1/q = 1$. In particular, the choices $\eta = \sqrt{(p-1)/Ts^2/q}$ and $p = 1 + (2 \log s - 1)^{-1}$ give:

$$R_T \leq \sqrt{2eT} \log s.$$

Proof h_p being $(p-1)$ -strongly convex with respect to $\|\cdot\|_p$, and $\|\cdot\|_q$ being the dual norm of $\|\cdot\|_p$, Theorem 1 gives:

$$R_T \leq \frac{\delta_{h_p}}{\eta} + \frac{\eta}{2(p-1)} \sum_{t=1}^T \|g_t\|_q^2.$$

For each $t \geq 1$, the norm of g_t can be bounded as follows:

$$\|g_t\|_q^2 = \left(\sum_{i=1}^d |g_t^{(i)}| \right)^{2/q} \leq \left(\sum_{s \text{ terms}} |g_t^{(i)}|^q \right)^{2/q} \leq s^{2/q},$$

which yields

$$R_T \leq \frac{1}{2\eta} + \frac{\eta T s^{2/q}}{2(p-1)}.$$

We can now balance both terms by choosing $\eta = \sqrt{(p-1)/(T s^{2/q})}$ and get:

$$R_T \leq \sqrt{\frac{T s^{2/q}}{p-1}}.$$

Finally, since $s \geq 3$, we have $2 \log s > 1$ and we set $p = 1 + (2 \log s - 1)^{-1} \in (1, 2]$, which gives:

$$\frac{1}{q} = 1 - \frac{1}{p} = \frac{p-1}{p} = \frac{(2 \log s - 1)^{-1}}{1 + (2 \log s - 1)^{-1}} = \frac{1}{2 \log s},$$

and thus:

$$R_T \leq \sqrt{\frac{T s^{2/q}}{p-1}} = \sqrt{2T \log s e^{2 \log s/q}} = \sqrt{2eT} \log s. \quad \blacksquare$$

We emphasize the fact that we obtain, up to a multiplicative constant, the exact same rate as when the decision maker only has a set of s decisions.

Theorem 2 was restricted to $s \geq 3$ to simplify the analysis. In the cases $s = 1, 2$, we can easily derive a bound of respectively \sqrt{T} and $\sqrt{2T}$ using the same regularizer with $p = 2$.

2.3 Matching Lower Bound

For $s \in [d]$ and $T \geq 1$, we denote $v_T^{g,s,d}$ the minimax regret of the T -stage decision problem with outcome vectors restricted to s -sparse gains:

$$v_T^{g,s,d} = \min_{\text{strat.}} \max_{(g_t)_t} R_T$$

where the minimum is taken over all possible policies of the decision maker, and the maximum over all sequences of s -sparse gains vectors.

To establish a lower bound in the present setting, we can assume that only the s first coordinates of g_t may be positive (for all $t \geq 1$) and that the decision maker is aware of that. Therefore he has no interest in assigning positive probabilities to any decision but the first s ones. Indeed, for any mixed action x_t , the decision maker can construct alternative mixed action $x'_t = (x_t^{(1)}, \dots, x_t^{(s)}, 0, \dots, 0)$ which obviously give a higher payoff:

$$\langle g_t, x_t \rangle \leq \langle g_t, x'_t \rangle$$

and therefore a lower regret:

$$\max_{t \in [d]} \sum_{t=1}^T g_t^{(t)} - \sum_{t=1}^T \langle g_t, x_t^* \rangle \leq \max_{t \in [d]} \sum_{t=1}^T g_t^{(t)} - \sum_{t=1}^T \langle g_t, x_t \rangle.$$

Therefore, we can restrict the strategies of the decision maker to those which assign positive probability to the s first components only. That setup, which is simpler for the decision maker than the original one, is obviously equivalent to the basic regret minimization problem with only s decisions. Therefore, the classical lower bound (Cesa-Bianchi et al., 1997, Theorem 3.2.3) holds and we obtain the following:

Theorem 3

$$\liminf_{s \rightarrow +\infty} \liminf_{T \rightarrow +\infty} \frac{v_T^{g^*, d}}{\sqrt{T} \log s} \geq \frac{\sqrt{2}}{2}.$$

The same lower bound, up to the multiplicative constant actually holds non asymptotically, see (Cesa-Bianchi and Lugosi, 2006, Theorem 3.6).

An immediate consequence of Theorem 3 is that the regret bound derived in Theorem 2 is asymptotically minimax optimal, up to a multiplicative constant.

3. When Outcomes are Losses to be Minimized

3.1 Upper Bound on the Regret

We now consider the case of losses, and the regularizer shall no longer depend on s (as with gains), as we will always use the Exponential Weight Algorithm. Instead, it is the parameter η that will be tuned as a function of s .

Theorem 4 *Let $s \geq 1$. For any sequence of s -sparse loss vectors $(\ell_t)_{t \geq 1}$, i.e., $\ell_t \in [0, 1]^d$ and $\|\ell_t\|_0 \leq s$, the Exponential Weight Algorithm with parameter $-\eta$ where*

$$\eta := \log \left(1 + \sqrt{2d \log d / sT} \right) > 0$$

guarantees, for $T \geq 1$:

$$R_T \leq \sqrt{\frac{2sT \log d}{d}} + \log d.$$

We build upon the following regret bound for losses which is written in terms of the performance of the best action. It is often called *improvement for small losses*: see e.g. (Littstone and Warmuth, 1994) or (Cesa-Bianchi and Lugosi, 2006, Theorem 2.4).

Theorem 5 *Let $\eta > 0$. For any sequence of loss vectors $(\ell_t)_{t \geq 1}$ in $[0, 1]^d$, the Exponential Weight Algorithm with parameter $-\eta$ guarantees, for all $T \geq 1$:*

$$R_T \leq \frac{\log d}{1 - e^{-\eta}} + \left(\frac{\eta}{1 - e^{-\eta}} - 1 \right) L_T^*,$$

where $L_T^* = \min_{t \in [d]} \sum_{t=1}^T \ell_t^{(t)}$ is the loss of the best stationary decision.

Proof Let $T \geq 1$ and $L_T^* = \min_{t \in [d]} \sum_{t=1}^T \ell_t^{(t)}$ be the loss of the best stationary policy. First note that since the loss vectors ℓ_t are s -sparse, we have $s \geq \sum_{t=1}^d \ell_t^{(t)}$. By summing over $1 \leq t \leq T$:

$$sT \geq \sum_{t=1}^T \sum_{t=1}^d \ell_t^{(t)} = \sum_{t=1}^d \left(\sum_{t=1}^T \ell_t^{(t)} \right) \geq d \left(\min_{t \in [d]} \sum_{t=1}^T \ell_t^{(t)} \right) = dL_T^*,$$

and therefore, we have $L_T^* \leq Ts/d$.

Then, by using the inequality $\eta \leq (e^\eta - e^{-\eta})/2$, the bound from Theorem 5 becomes:

$$R_T \leq \frac{\log d}{1 - e^{-\eta}} + \left(\frac{e^\eta - e^{-\eta}}{2(1 - e^{-\eta})} - 1 \right) L_T^*.$$

The factor of L_T^* in the second term can be transformed as follows:

$$\frac{e^\eta - e^{-\eta}}{2(1 - e^{-\eta})} - 1 = \frac{(1 + e^{-\eta})(e^\eta - e^{-\eta})}{2(1 - e^{-2\eta})} - 1 = \frac{(1 + e^{-\eta})e^\eta}{2} - 1 = \frac{e^\eta - 1}{2},$$

and therefore the bound on the regret becomes:

$$R_T \leq \frac{\log d}{1 - e^{-\eta}} + \frac{e^\eta - 1}{2} L_T^* \leq \frac{\log d}{1 - e^{-\eta}} + \frac{(e^\eta - 1)Ts}{2d},$$

where we have been able to use the upper-bound on L_T^* since $\frac{e^\eta - 1}{2} \geq 0$. Along with the choice $\eta = \log(1 + \sqrt{2d \log d / Ts})$ and standard computations, this yields:

$$R_T \leq \sqrt{\frac{2Ts \log d}{d}} + \log d. \quad \blacksquare$$

Interestingly, the bound from Theorem 4 shows that $\sqrt{2sT \log d / d}$, the dominating term of the regret bound, is *decreasing* when the number of decisions d increases. This is due to the sparsity assumptions (as the regret increases with s , the maximal number of decision with positive losses). Indeed, when s is fixed and d increases, more and more decisions are optimal at each stage, a proportion $1 - s/d$ to be precise. As a consequence, it becomes *easier* to find an optimal decisions when d increases. However, this intuition will turn out not to be valid in the bandit framework.

On the other hand, if the proportion s/d of positive losses remains constant then the regret bound achieved is of the same order as in the usual case.

3.2 Matching Lower Bound

When outcomes are losses, the argument from Section 2.3 does not allow to derive a lower bound. Indeed, if we assume that only the first s coordinates of the loss vectors ℓ_t can be positive, and that the decision maker knows it, then he just has to take at each stage the decision $d_t = d$ which incurs a loss of 0. As a consequence, he trivially has a regret

$R_T = 0$. Choosing at random, but once and for all, a fixed subset of s coordinates does not provide any interesting lower bound either. Instead, the key idea of the following result is to choose at random and at each stage the s coordinates associated with positive losses. And we therefore use the following classical probabilistic argument. Assume that we have found a probability distribution on $(\ell_t)_t$ such that the expected regret can be bounded from below by a quantity which does not depend on the strategy of the decision maker. This would imply that for any algorithm, there exists a sequence of $(\ell_t)_t$ such that the regret is greater than the same quantity.

In the following statement, $v_T^{s,d}$ stands for the minimax regret in the case where outcomes are losses.

Theorem 6 For all $s \geq 1$,

$$\liminf_{d \rightarrow +\infty} \liminf_{T \rightarrow +\infty} \frac{v_T^{s,d}}{\sqrt{T^s \log d}} \geq \frac{\sqrt{2}}{2}.$$

The main consequences of this theorem are that the algorithm described in Theorem 4 is asymptotically minimax optimal (up to a multiplicative constant) and that gains and losses are fundamentally different from the point of view of regret minimization.

Proof We define the sequence of i.i.d. loss vectors ℓ_t ($t \geq 1$) as follows. First, we draw a set $I_t \subset [d]$ of cardinality s uniformly among the $\binom{d}{s}$ possibilities. Then, if $i \in I_t$ set $\ell_t^{(i)} = 1$ with probability $1/2$ and $\ell_t^{(i)} = 0$ with probability $1/2$, independently for each component. If $i \notin I_t$, we set $\ell_t^{(i)} = 0$.

As a consequence, we always have that ℓ_t is s -sparse. Moreover, for each $t \geq 1$ and each coordinate $i \in [d]$, $\ell_t^{(i)}$ satisfies:

$$\mathbb{P}[\ell_t^{(i)} = 1] = \frac{s}{2d} \quad \text{and} \quad \mathbb{P}[\ell_t^{(i)} = 0] = 1 - \frac{s}{2d},$$

thus $\mathbb{E}[\ell_t^{(i)}] = s/2d$. Therefore we obtain that for any algorithm $(x_t)_{t \geq 1}$, $\mathbb{E}[\langle \ell_t, x_t \rangle] = s/2d$. This yields that

$$\begin{aligned} \mathbb{E} \left[\frac{R_T}{\sqrt{T}} \right] &= \mathbb{E} \left[\frac{1}{\sqrt{T}} \left(\sum_{t=1}^T \langle \ell_t, x_t \rangle - \min_{i \in [d]} \sum_{t=1}^T \ell_t^{(i)} \right) \right] \\ &= \mathbb{E} \left[\frac{\max_{i \in [d]} \frac{1}{\sqrt{T}} \sum_{t=1}^T (s - \ell_t^{(i)})}{\sqrt{T}} \right] \\ &= \mathbb{E} \left[\frac{\max_{i \in [d]} \frac{1}{\sqrt{T}} \sum_{t=1}^T X_t^{(i)}}{\sqrt{T}} \right], \end{aligned}$$

where $t \geq 1$, we have defined the random vector X_t by $X_t^{(i)} = s/2d - \ell_t^{(i)}$ for all $i \in [d]$. For $t \geq 1$, the X_t are i.i.d. zero-mean random vectors with values in $[-1, 1]^d$. We can therefore apply the comparison Lemma 8 to get:

$$\liminf_{T \rightarrow +\infty} \mathbb{E} \left[\frac{R_T}{\sqrt{T}} \right] = \liminf_{T \rightarrow +\infty} \mathbb{E} \left[\max_{i \in [d]} \frac{1}{\sqrt{T}} \sum_{t=1}^T X_t^{(i)} \right] \geq \mathbb{E} \left[\max_{i \in [d]} Z^{(i)} \right],$$

where $Z \sim \mathcal{N}(0, \Sigma)$ with $\Sigma = (\text{cov}(X_1^{(i)}, X_1^{(j)}))_{i,j}$.

We now make appeal to Slepian's lemma, recalled in Proposition 7 below. Therefore, we introduce the Gaussian vector $W \sim \mathcal{N}(0, \tilde{\Sigma})$ where

$$\tilde{\Sigma} = \text{diag} \left(\text{Var } X_1^{(1)}, \dots, \text{Var } X_1^{(1)} \right).$$

As a consequence, the first two hypotheses of Proposition 7 follow from the definitions of Z and W . Let $i \neq j$, then

$$\mathbb{E} \left[Z^{(i)} Z^{(j)} \right] = \text{cov}(Z^{(i)}, Z^{(j)}) = \text{cov}(\ell_1^{(i)}, \ell_1^{(j)}) = \mathbb{E} \left[\ell_1^{(i)} \ell_1^{(j)} \right] - \mathbb{E} \left[\ell_1^{(i)} \right] \mathbb{E} \left[\ell_1^{(j)} \right].$$

By definition of ℓ_1 , $\ell_1^{(i)} \ell_1^{(j)} = 1$ if and only if $\ell_1^{(i)} = \ell_1^{(j)} = 1$ and $\ell_1^{(i)} \ell_1^{(j)} = 0$ otherwise. Therefore, using the random subset I_1 that appears in the definition of ℓ_1 :

$$\begin{aligned} \mathbb{E} \left[Z^{(i)} Z^{(j)} \right] &= \mathbb{P} \left[\ell_1^{(i)} = \ell_1^{(j)} = 1 \right] - \left(\frac{s}{2d} \right)^2 \\ &= \mathbb{P} \left[\ell_1^{(i)} = \ell_1^{(j)} = 1 \mid \{i, j\} \subset I_1 \right] \mathbb{P} \left[\{i, j\} \subset I_1 \right] - \left(\frac{s}{2d} \right)^2 \\ &= \frac{1}{4} \cdot \frac{\binom{d-2}{s-2}}{\binom{d}{s}} - \left(\frac{s}{2d} \right)^2 \\ &= \frac{1}{4} \left(\frac{s(s-1)}{d(d-1)} - \frac{s^2}{d^2} \right) \leq 0, \end{aligned}$$

and since $\mathbb{E} \left[W^{(i)} W^{(j)} \right] = 0$ by independence, the third hypothesis of Slepian's lemma is also satisfied. It yields that, for all $\theta \in \mathbb{R}$:

$$\begin{aligned} \mathbb{P} \left[\max_{i \in [d]} Z^{(i)} \leq \theta \right] &= \mathbb{P} \left[Z^{(1)} \leq \theta, \dots, Z^{(d)} \leq \theta \right] \\ &\leq \mathbb{P} \left[W^{(1)} \leq \theta, \dots, W^{(d)} \leq \theta \right] = \mathbb{P} \left[\max_{i \in [d]} W^{(i)} \leq \theta \right]. \end{aligned}$$

This inequality between two cumulative distribution functions implies the reverse inequality on expectations:

$$\mathbb{E} \left[\max_{i \in [d]} Z^{(i)} \right] \geq \mathbb{E} \left[\max_{i \in [d]} W^{(i)} \right].$$

The components of the Gaussian vector W being independent, and of same variance $\text{Var } \ell_1^{(1)}$, we have

$$\mathbb{E} \left[\max_{i \in [d]} W^{(i)} \right] = \kappa_d \sqrt{\text{Var } \ell_1^{(1)}} = \kappa_d \sqrt{\frac{s}{2d} \left(1 - \frac{s}{2d} \right)} \geq \kappa_d \sqrt{\frac{s}{4d}},$$

where κ_d is the expectation of the maximum of d Gaussian variables. Combining everything gives:

$$\liminf_{T \rightarrow +\infty} \frac{v_T^{s,d}}{\sqrt{T}} \geq \liminf_{T \rightarrow +\infty} \mathbb{E} \left[\frac{R_T}{\sqrt{T}} \right] \geq \mathbb{E} \left[\max_{i \in [d]} Z^{(i)} \right] \geq \mathbb{E} \left[\max_{i \in [d]} W^{(i)} \right] \geq \kappa_d \sqrt{\frac{s}{4d}}.$$

And for large d , since κ_d is equivalent to $\sqrt{2 \log d}$ (see e.g. Galambos, 1978),

$$\liminf_{d \rightarrow +\infty} \liminf_{T \rightarrow +\infty} \frac{\kappa_{T, \kappa, d}^{\kappa, \kappa, d}}{\sqrt{T \frac{\kappa}{d} \log d}} \geq \sqrt{2}.$$

■

Proposition 7 (Slepian (1962)) *Let $Z = (Z^{(1)}, \dots, Z^{(d)})$ and $W = (W^{(1)}, \dots, W^{(d)})$ be Gaussian random vectors in \mathbb{R}^d satisfying:*

$$(i) \mathbb{E}[Z] = \mathbb{E}[W] = 0;$$

$$(ii) \mathbb{E}[Z^{(i)}]^2 = \mathbb{E}[W^{(i)}]^2 \text{ for } i \in [d];$$

$$(iii) \mathbb{E}[Z^{(i)}Z^{(j)}] \leq \mathbb{E}[W^{(i)}W^{(j)}] \text{ for } i \neq j \in [d].$$

Then, for all real numbers $\theta_1, \dots, \theta_d$, we have:

$$\mathbb{P}\left[Z^{(1)} \leq \theta_1, \dots, Z^{(d)} \leq \theta_d\right] \leq \mathbb{P}\left[W^{(1)} \leq \theta_1, \dots, W^{(d)} \leq \theta_d\right].$$

The following lemma is an extension of e.g. (Cesa-Bianchi and Lugosi, 2006, Lemma A.11) to random vectors with correlated components.

Lemma 8 (Comparison lemma) *For $t \geq 1$, let $(X_t)_{t \geq 1}$ be i.i.d. zero-mean random vectors in $[-1, 1]^d$, Σ be the covariance matrix of X_t and $Z \sim \mathcal{N}(0, \Sigma)$. Then,*

$$\liminf_{T \rightarrow +\infty} \mathbb{E}\left[\max_{i \in [d]} \frac{1}{\sqrt{T}} \sum_{t=1}^T X_t^{(i)}\right] \geq \mathbb{E}\left[\max_{i \in [d]} Z^{(i)}\right].$$

Proof Denote

$$Y_T = \max_{i \in [d]} \frac{1}{\sqrt{T}} \sum_{t=1}^T X_t^{(i)}.$$

Let $A \leq 0$ and consider the function $\phi_A : \mathbb{R} \rightarrow \mathbb{R}$ defined by $\phi_A(x) = \max(x, A)$.

$$\begin{aligned} \mathbb{E}[Y_T] &= \mathbb{E}[Y_T \cdot \mathbb{1}_{\{Y_T \geq A\}}] + \mathbb{E}[Y_T \cdot \mathbb{1}_{\{Y_T < A\}}] \\ &= \mathbb{E}[\phi_A(Y_T) \cdot \mathbb{1}_{\{Y_T \geq A\}}] + \mathbb{E}[Y_T \cdot \mathbb{1}_{\{Y_T < A\}}] \\ &= \mathbb{E}[\phi_A(Y_T)] - \mathbb{E}[\phi_A(Y_T) \cdot \mathbb{1}_{\{Y_T < A\}}] + \mathbb{E}[Y_T \cdot \mathbb{1}_{\{Y_T < A\}}] \\ &= \mathbb{E}[\phi_A(Y_T)] - \mathbb{E}[(A - Y_T) \cdot \mathbb{1}_{\{A - Y_T > 0\}}]. \end{aligned}$$

Let us estimate the second term. Denote $Z_T = (A - Y_T) \cdot \mathbb{1}_{\{A - Y_T > 0\}}$. We clearly have, for all $u > 0$, $\mathbb{P}[Z_T > u] = \mathbb{P}[A - Y_T > u]$. And Z_T being nonnegative, we can write:

$$\begin{aligned} 0 &\leq \mathbb{E}[(A - Y_T) \cdot \mathbb{1}_{\{A - Y_T > 0\}}] = \mathbb{E}[Z_T] \\ &= \int_0^{+\infty} \mathbb{P}[Z_T > u] \, du \\ &= \int_0^{+\infty} \mathbb{P}[A - Y_T > u] \, du \\ &= \int_{-A}^{+\infty} \mathbb{P}[Y_T < -u] \, du \\ &= \int_{-A}^{+\infty} \mathbb{P}\left[\max_{i \in [d]} \frac{1}{\sqrt{T}} \sum_{t=1}^T X_t^{(i)} < u\right] \, du \\ &\leq \int_{-A}^{+\infty} \mathbb{P}\left[\sum_{t=1}^T X_t^{(1)} < u\sqrt{T}\right] \, du. \end{aligned}$$

For $u > 0$, using Hoeffding's inequality together with the assumptions $\mathbb{E}[X_t^{(1)}] = 0$ and $X_t^{(1)} \in [-1, 1]$, we can bound the last integrand:

$$\mathbb{P}\left[\sum_{t=1}^T X_t^{(1)} < u\sqrt{T}\right] \leq e^{-u^2/2},$$

Which gives:

$$0 \leq \mathbb{E}[(A - Y_T) \cdot \mathbb{1}_{\{A - Y_T > 0\}}] \leq \int_{-A}^{+\infty} e^{-u^2/2} \, du \leq \frac{e^{-A^2/2}}{-A}.$$

Therefore:

$$\mathbb{E}[Y_T] \geq \mathbb{E}[\phi_A(Y_T)] + \frac{e^{-A^2/2}}{A}.$$

We now take the liminf on both sides as $t \rightarrow +\infty$. The left-hand side is the quantity that appears in the statement. We now focus on the second term of the right-hand side. The central limit theorem gives the following convergence in distribution:

$$\frac{1}{\sqrt{T}} \sum_{t=1}^T X_t \xrightarrow[T \rightarrow +\infty]{\mathcal{L}} X.$$

The application $(x^{(1)}, \dots, x^{(d)}) \mapsto \max_{i \in [d]} x^{(i)}$ being continuous, we can apply the continuous mapping theorem:

$$Y_T = \max_{i \in [d]} \frac{1}{\sqrt{T}} \sum_{t=1}^T X_t^{(i)} \xrightarrow[T \rightarrow +\infty]{\mathcal{L}} \max_{i \in [d]} X^{(i)}.$$

This convergence in distribution allows the use of the portmanteau lemma: ϕ_A being lower semi-continuous and bounded from below, we have:

$$\liminf_{t \rightarrow +\infty} \mathbb{E}[\phi_A(Y_T)] \geq \mathbb{E} \left[\phi_A \left(\max_{i \in [d]} X^{(i)} \right) \right],$$

and thus:

$$\liminf_{t \rightarrow +\infty} \mathbb{E}[Y_T] \geq \mathbb{E} \left[\phi_A \left(\max_{i \in [d]} X^{(i)} \right) \right] + \frac{e^{-A^2/2}}{A}.$$

We would now like to take the limit as $A \rightarrow -\infty$. By definition of ϕ_A , for $A \leq 0$, we have the following domination:

$$\left| \phi_A \left(\max_{i \in [d]} X^{(i)} \right) \right| \leq \left| \max_{i \in [d]} X^{(i)} \right| \leq \max_{i \in [d]} |X^{(i)}| \leq \sum_{i=1}^d |X^{(i)}|,$$

where each $X^{(i)}$ is L^1 since it is a normal random variable. We can therefore apply the dominated convergence theorem as $A \rightarrow -\infty$:

$$\mathbb{E} \left[\phi_A \left(\max_{i \in [d]} X^{(i)} \right) \right] \xrightarrow{A \rightarrow -\infty} \mathbb{E} \left[\max_{i \in [d]} X^{(i)} \right],$$

and eventually, we get the stated result: \blacksquare

$$\liminf_{t \rightarrow +\infty} \mathbb{E}[Y_T] \geq \mathbb{E} \left[\max_{i \in [d]} X^{(i)} \right].$$

4. When the Sparsity Level s is Unknown

We no longer assume in this section that the decision maker have the knowledge of the sparsity level s . We modify our algorithms to be adaptive over the sparsity level of the observed gain/loss vectors. The algorithms are proved to essentially achieve the same regret bounds as in the case where s is known. The constructions follow the same ideas behind the classical doubling trick. However, the latter cannot be directly applied here: the usual doubling trick involves time intervals whose lengths are always the same, whereas we here need to make the lengths on the sparsity levels of the payoff vectors.

Specifically, let $T \geq 1$ be the number of rounds and s^* the highest sparsity level of the gain/loss vectors chosen by Nature up to time T . In the following, we construct algorithms which achieve regret bounds of order $\sqrt{T \log s^*}$ and $\sqrt{T s^* \log d}$ for gains and losses respectively, without prior knowledge of s^* .

4.1 For Losses

Let $(\ell_t)_{t \geq 1}$ be the sequence of loss vectors in $[0, 1]^d$ chosen by Nature, and $T \geq 1$ the number of rounds. We denote $s^* = \max_{1 \leq t \leq T} \|\ell_t\|_0$ the higher sparsity level of the loss vectors up

to time T . The goal is to construct an algorithm which achieves a regret bound of order $\sqrt{T s^* \log d}$ without any prior knowledge about the sparsity level of the loss vectors.

The time instances $\{1, \dots, T\}$ will be divided into several time intervals. On each of those, the previous loss vectors will be left aside, and a new instance of the Exponential Weight Algorithm with a specific parameter will be run. Let $M = \lceil \log_2 s^* \rceil$ and $\tau(0) = 0$. Then, for $1 \leq m < M$ we define

$$\tau(m) = \min \{1 \leq t \leq T \mid \|\ell_t\|_0 > 2^m\} \quad \text{and} \quad \tau(M) = T.$$

In other words, $\tau(m)$ is the first time instance at which the sparsity level of the loss vector exceeds 2^m . $(\tau(m))_{1 \leq m \leq M}$ is thus a nondecreasing sequence. We can then define the time intervals $I(m)$ as follows. For $1 \leq m \leq M$, let

$$I(m) = \begin{cases} \{\tau(m-1) + 1, \dots, \tau(m)\} & \text{if } \tau(m-1) < \tau(m) \\ \emptyset & \text{if } \tau(m-1) = \tau(m). \end{cases}$$

The sets $(I(m))_{1 \leq m \leq M}$ clearly form a partition of $\{1, \dots, T\}$ (some of the intervals may be empty). For $1 \leq t \leq T$, we define $m_t = \min \{m \geq 1 \mid \tau(m) \geq t\}$ which implies $t \in I(m_t)$. In other words, m_t is the index of the only interval t belongs to.

Let $C > 0$ be a constant to be chosen later and for $1 \leq m \leq M$, let

$$\eta(m) = \log \left(1 + C \sqrt{\frac{d \log d}{2^m T}} \right)$$

be the parameter of the Exponential Weight Algorithm to be used on interval $I(m)$. In this section, h will be entropic regularizer on the simplex $h(x) = \sum_{i=1}^d x^{(i)} \log x^{(i)}$, so that $y \mapsto \nabla h^*(y)$ is the *logit map* used in the Exponential Weight Algorithm. We can then define the played actions to be:

$$x_t = \nabla h^* \left(-\eta(m_t) \sum_{\substack{t' < t \\ t' \in I(m_t)}} \ell_{t'} \right), \quad t = 1, \dots, T.$$

Theorem 9 *The above algorithm with $C = 2^{3/4}(\sqrt{2} + 1)^{1/2}$ guarantees*

$$R_T \leq 4 \sqrt{\frac{T s^* \log d}{d}} + \frac{\lceil \log s^* \rceil \log d}{2} + 5 s^* \sqrt{\frac{\log d}{dT}}.$$

Proof Let $1 \leq m \leq M$. On time interval $I(m)$, the Exponential Weight Algorithm is run with parameter $\eta(m)$ against loss vectors in $[0, 1]^d$. Therefore, the following regret bound

Algorithm 1: For losses in full information without prior knowledge about sparsity
input: $T \geq 1$, $d \geq 1$ integers, and $C > 0$.

```

 $\eta \leftarrow \log(1 + C\sqrt{d\log d/2T});$ 
 $m \leftarrow 1;$ 
for  $i \leftarrow 1$  to  $d$  do
   $w^{(i)} \leftarrow 1/d;$ 
end
for  $t \leftarrow 1$  to  $T$  do
  draw and play decision  $i$  with probability  $w^{(i)}/\sum_{j=1}^d w^{(j)}$ ;
  observe loss vector  $\ell_t$ ;
  if  $\|\ell_t\|_0 \leq 2^m$  then
    for  $i \leftarrow 1$  to  $d$  do
       $w^{(i)} \leftarrow w^{(i)}e^{-\eta\ell_t^{(i)}};$ 
    end
  else
     $m \leftarrow \lceil \log_2 \|\ell_t\|_0 \rceil;$ 
     $\eta \leftarrow \log(1 + C\sqrt{d\log d/2^m T});$ 
    for  $i \leftarrow 1$  to  $d$  do
       $w^{(i)} \leftarrow 1/d;$ 
    end
  end
end

```

derived in the proof of Theorem 4 applies:

$$\begin{aligned}
R(m) &:= \sum_{t \in I(m)} \langle \ell_t, x_t \rangle - \min_{i \in [d]} \sum_{t \in I(m)} \ell_t^{(i)} \\
&\leq \frac{\log d}{1 - e^{-\eta(m)}} + \frac{e^{\eta(m)} - 1}{2} \min_{i \in [d]} \sum_{t \in I(m)} \ell_t^{(i)} \\
&= \frac{1}{C} \sqrt{2^m T \log d} + \frac{\log d}{C} + \frac{C}{2} \sqrt{\frac{d \log d}{2^m T}} \cdot \min_{i \in [d]} \sum_{t \in I(m)} \ell_t^{(i)}.
\end{aligned}$$

We now bound the ‘‘best loss’’ quantity from above, using the fact that ℓ_t is 2^m -sparse for $t \in I(m) \setminus \{\tau(m)\}$ and that $\ell_{\tau(m)}$ is s^* -sparse:

$$\begin{aligned}
\sum_{i=1}^d \sum_{t \in I(m)} \ell_t^{(i)} &= \sum_{t \in I(m)} \sum_{i=1}^d \ell_t^{(i)} = \sum_{t \in I(m)} \sum_{i=1}^d \rho_t^{(i)} + \sum_{i=1}^d \rho_{\tau(m)}^{(i)} \\
&\leq (\tau(m) - \tau(m-1))2^m + s^*,
\end{aligned}$$

which implies:

$$\min_{i \in [d]} \sum_{t \in I(m)} \ell_t^{(i)} \leq \frac{(\tau(m) - \tau(m-1))2^m + s^*}{d}.$$

Therefore, the regret on interval $I(m)$, which we will denote $R(m)$, is bounded by:

$$\begin{aligned}
R(m) &:= \sum_{t \in I(m)} \langle \ell_t, x_t \rangle - \min_{i \in [d]} \sum_{t \in I(m)} \ell_t^{(i)} \\
&\leq \frac{1}{C} \sqrt{\frac{2^m T \log d}{d}} + \frac{\log d}{C} + \frac{C}{2} \sqrt{\frac{2^m \log d}{dT}} (\tau(m) - \tau(m-1)) + \frac{C}{2} \sqrt{\frac{\log d}{2^m d T}} s^* \\
&\leq \frac{1}{C} \sqrt{\frac{2^m T \log d}{d}} + \frac{\log d}{C} + \frac{C}{2} \sqrt{\frac{2s^* \log d}{dT}} (\tau(m) - \tau(m-1)) + \frac{C}{2} \sqrt{\frac{\log d}{2^m d T}} s^*,
\end{aligned}$$

where we used $2^m \leq 2^M \leq 2^{\lceil \log_2 s^* \rceil} \leq 2^{\log_2 s^* + 1} = 2s^*$ for the third term of the last line.

We now turn the whole regret R_T from 1 to T . Since $(I(m))_{1 \leq m \leq M}$ is a partition of $\{1, \dots, T\}$, we obtain

$$\begin{aligned}
R_T &= \sum_{t=1}^T \langle \ell_t, x_t \rangle - \min_{i \in [d]} \sum_{t=1}^T \ell_t^{(i)} \\
&\leq \sum_{m=1}^M \sum_{t \in I(m)} \langle \ell_t, x_t \rangle - \sum_{m=1}^M \min_{i \in [d]} \sum_{t \in I(m)} \ell_t^{(i)} \\
&= \sum_{m=1}^M R(m) \\
&\leq \frac{1}{C} \sqrt{\frac{T \log d}{d}} \sum_{m=1}^M \sqrt{2^m} + C \sqrt{\frac{s^* T \log d}{2d}} + \frac{M \log d}{C} + \frac{C}{2} \sqrt{\frac{\log d}{dT}} s^* \sum_{m=1}^M 2^{-m/2}.
\end{aligned}$$

The sum in the first term above can be bounded as follows

$$\sum_{m=1}^M \sqrt{2^m} \leq \sum_{m=1}^M \sqrt{2^m} = \sqrt{2} \frac{\sqrt{2^M} - 1}{\sqrt{2} - 1} \leq \sqrt{2} \frac{\sqrt{2} \sqrt{2^M} - 1}{\sqrt{2} - 1} = 2 \frac{\sqrt{s^*} - 1}{\sqrt{2} - 1} = 2(\sqrt{2} + 1)\sqrt{s^*},$$

whereas the sum in the last term can be bounded by $\sqrt{2} + 1$. Eventually, the choice $C = 2^{3/4}(\sqrt{2} + 1)^{1/2}$ gives:

$$R_T \leq 2^{5/4}(\sqrt{2} + 1)^{1/2} \sqrt{\frac{T s^* \log d}{d}} + \frac{\lceil \log s^* \rceil \log d}{2^{3/4}(\sqrt{2} + 1)^{1/2}} + 2^{1/4}(\sqrt{2} + 1)^{3/2} s^* \sqrt{\frac{\log d}{dT}},$$

and the statement follows from numerical computation of the constant factors. \blacksquare

4.2 For Gains

The construction is similar to the case of losses, but the time intervals are slightly different. Let $(g_t)_{t \geq 1}$ be the sequence of gain vectors in $[0, 1]^d$ chosen by Nature. We assume $s^* \geq 2$ and set $M = \lceil \log_2 \log_2 s^* \rceil$ and $\tau(0) = 0$. For $1 \leq m \leq M$ we define

$$\tau(m) = \min\{1 \leq t \leq T \mid \|g_t\|_0 > 2^{2^m}\} \quad \text{and} \quad \tau(M) = T.$$

We now define the time intervals $I(m)$. For $1 \leq m \leq M$,

$$I(m) = \begin{cases} \{\tau(m-1) + 1, \dots, \tau(m)\} & \text{if } \tau(m-1) < \tau(m) \\ \emptyset & \text{if } \tau(m-1) = \tau(m). \end{cases}$$

Therefore, for $1 \leq m \leq M$ and $t < \tau(m)$, we have $\|g_t\|_0 \leq 2^{2^m}$. For $1 \leq t \leq T$, we denote $m_t = \min\{m \geq 1 \mid \tau(m) \geq t\}$. Let $C > 0$ be a constant to be chosen later and for $1 \leq m \leq M$, let

$$\begin{aligned} p(m) &= 1 + \frac{1}{\log 2 \cdot 2^{m+1} - 1}, \\ q(m) &= \left(1 - \frac{1}{p(m)}\right)^{-1}, \\ \eta(m) &= C \sqrt{\frac{p(m) - 1}{T 2^{2^{m+1}/q(m)}}}. \end{aligned}$$

As in Section 2.2, for $p \in (1, 2]$, we denote h_p the regularizer on the simplex defined by:

$$h_p(x) = \begin{cases} \frac{1}{2} \|x\|_p^2 & \text{if } x \in \Delta_d \\ +\infty & \text{otherwise.} \end{cases}$$

The algorithm is then defined by:

$$x_t = \nabla h_{p(m_t)}^* \left(\eta(m_t) \sum_{\substack{t' < t \\ t' \in I(m_t)}} g_{t'} \right), \quad t = 1, \dots, T.$$

Algorithm 2: For gains in full information without prior knowledge about sparsity.

input: $T \geq 1$, $d \geq 1$ integers, and $C > 0$.

$p \leftarrow 1 + (4 \log 2 - 1)^{-1}$;

$q \leftarrow (1 - 1/p)^{-1}$;

$\eta \leftarrow C \sqrt{(p-1)/2^{4/q} T}$;

$m \leftarrow 1$;

$y \leftarrow (0, \dots, 0) \in \mathbb{R}^d$;

for $t \leftarrow 1$ **to** T **do**

draw and play decision $i \sim \nabla h_p^*(\eta \cdot y)$;

observe gain vector g_t ;

if $\|g_t\|_0 \leq 2^{2^m}$ **then**

| $y \leftarrow y + g_t$;

else

| $m \leftarrow \lceil \log_2 \log_2 \|g_t\|_0 \rceil$;

| $p \leftarrow 1 + (\log 2 \cdot 2^{m+1} - 1)^{-1}$;

| $q \leftarrow (1 - 1/p)^{-1}$;

| $\eta \leftarrow C \sqrt{(p-1)/2^{2^{m+1}/q} T}$;

| $y \leftarrow (0, \dots, 0)$;

end

end

Theorem 10 *The above algorithm with $C = (e\sqrt{2}(\sqrt{2} + 1))^{1/2}$ guarantees*

$$R_T \leq 7\sqrt{T} \log s^* + \frac{4s^*}{\sqrt{T}}.$$

Proof Let $1 \leq m \leq M$. On time interval $I(m)$, the algorithm boils down to an Online Mirror Descent algorithm with regularizer $h_{p(m)}$ and parameter $\eta(m)$. Therefore, using Theorem 1, the regret on this interval is bounded as follows.

$$\begin{aligned} R(m) &:= \max_{t \in I(m)} \sum_{t' \in I(m)} g_{t'} - \sum_{t \in I(m)} \langle g_t, x_t \rangle \\ &\leq \frac{1}{2\eta(m)} + \frac{\eta(m)}{2(p(m) - 1)} \sum_{t \in I(m)} \|g_t\|_{q(m)}^2 \\ &= \frac{1}{2\eta(m)} + \frac{\eta(m)}{2(p(m) - 1)} \left(\sum_{\substack{t \in I(m) \\ t < \tau(m)}} \|g_t\|_{q(m)}^2 + \|g_{\tau(m)}\|_{q(m)}^2 \right). \end{aligned}$$

g_t being 2^{2^m} -sparse for $t < \tau(m)$ and $g_{\tau(m)}$ being s^* -sparse, the $q(m)$ -norms can therefore be bounded from above as follows:

$$\|g_t\|_{q(m)}^2 \leq 2^{2^{m+1}/q(m)} \quad \text{and} \quad \|g_{\tau(m)}\|_{q(m)}^2 \leq (s^*)^{2/q(m)}.$$

The bound on $R(m)$ then becomes

$$\begin{aligned} R(m) &\leq \frac{1}{2\eta(m)} + \frac{\eta(m)(\tau(m) - \tau(m-1))2^{2^{m+1}/q(m)}}{2(p(m)-1)} + \frac{\eta(m)(s^*)^{2/q(m)}}{2(p(m)-1)} \\ &= \frac{1}{2C} \sqrt{T} e(\log 2 \cdot 2^{m+1} - 1) + \frac{C}{2} \sqrt{\frac{e(\log 2 \cdot 2^{m+1} - 1)}{T}} (\tau(m) - \tau(m-1)) \\ &\quad + \frac{C}{2} (s^*)^{1/(\log 2 \cdot 2^m)} \sqrt{\frac{e(\log 2 \cdot 2^{m+1} - 1)}{T}} \\ &\leq \frac{1}{2C} \sqrt{T} e \log 2 \cdot 2^{m+1} + C \sqrt{\frac{e \log s^*}{T}} (\tau(m) - \tau(m-1)) \\ &\quad + \frac{C}{2} s^* \sqrt{\frac{e \log 2 \cdot 2^{m+1}}{T}}, \end{aligned}$$

where for the second term of the last expression we used:

$$\begin{aligned} \log 2 \cdot 2^{m+1} - 1 &\leq \log 2 \cdot 2^{M+1} = \log 2 \cdot \exp(\log 2 (\lceil \log 2 \log_2 s^* \rceil + 1)) \\ &\leq \log 2 \cdot \exp(\log 2 (\log_2 \log_2 s^* + 2)) \\ &= \log 2 \cdot e^{2 \log 2} \exp(\log 2 \cdot \log_2 \log_2 s^*) \\ &= 4 \log 2 \cdot \exp(\log \log_2 s^*) \\ &= 4 \log 2 \cdot \log_2 s^* \\ &= 4 \log s^*. \end{aligned}$$

Then, the whole regret R_T is bounded by the sum of the regrets on each interval:

$$\begin{aligned} R_T &\leq \sum_{m=1}^M R(m) \leq \frac{1}{2C} \sqrt{T} e \log 2 \sum_{m=1}^M \sqrt{2^{m+1}} + C \sqrt{\frac{e \log s^*}{T}} \sum_{m=1}^M (\tau(m) - \tau(m-1)) \\ &\quad + \frac{C s^*}{2} \sqrt{\frac{e \log 2}{T}} \sum_{m=1}^M 2^{-(m+1)/2}. \end{aligned}$$

The second sum is equal to T and the third sum is bounded from above by $(\sqrt{2} + 1)/\sqrt{2}$. Let us bound the first sum from above:

$$\begin{aligned} \sqrt{\log 2} \sum_{m=1}^M \sqrt{2^{m+1}} &= 2 \sqrt{\log 2} \frac{2^{M/2} - 1}{\sqrt{2} - 1} \\ &\leq 2(\sqrt{2} + 1) \sqrt{\log 2} \cdot \exp\left(\frac{\log 2}{2} (\log_2 \log_2 s^* + 1)\right) \\ &= 2(\sqrt{2} + 1) \sqrt{\log 2} \cdot \sqrt{2} e^{\log \log_2 s^*} \\ &= 2\sqrt{2}(\sqrt{2} + 1) \sqrt{\log 2 \log_2 s^*} \\ &= 2\sqrt{2}(\sqrt{2} + 1) \sqrt{\log s^*}. \end{aligned}$$

Therefore,

$$R_T \leq \frac{\sqrt{2}(\sqrt{2} + 1)}{C} \sqrt{T} e \log s^* + C \sqrt{T} e \log s^* + \frac{C(\sqrt{2} + 1)s^*}{2} \sqrt{\frac{e \log 2}{2T}}.$$

Choosing $C = (e\sqrt{2}(\sqrt{2} + 1))^{1/2}$ balances the first two term and gives:

$$\begin{aligned} R_T &\leq 2(e\sqrt{2}(\sqrt{2} + 1))^{1/2} \sqrt{T} \log s^* + 2^{-5/4} e \sqrt{\log 2} (\sqrt{2} + 1)^{3/2} \frac{s^*}{\sqrt{T}} \\ &\leq \tau \sqrt{T} \log s^* + \frac{4s^*}{\sqrt{T}}. \end{aligned}$$

■

5. The Bandit Setting

We now turn to the bandit framework—see for instance (Bubeck and Cesa-Bianchi, 2012) for a recent survey. Recall that the minimax regret (Audibert and Bubeck, 2009) in the basic bandit framework (without sparsity) is of order \sqrt{Td} . In the case of losses, we manage to take advantage of the sparsity assumption and obtain in Theorem 11 an upper bound of order $\sqrt{T s \log \frac{d}{s}}$, and a lower bound of order $\sqrt{T s}$ in Theorem 13. This establishes the order of the minimax regret up to a logarithmic factor. In the case of gains, the argument from Section 2.3 can be adapted to get a lower bound of order \sqrt{sT} ; but the upper bound techniques from losses do not seem to work: this difficulty is discussed below in Remark 12.

For simplicity, we shall assume that the sequence of outcome vectors $(\omega_t)_{t \geq 1}$ is chosen before stage 1 by the environment, which is called *oblivious* in that case. We refer to (Bubeck and Cesa-Bianchi, 2012, Section 3) for a detailed discussion on the difference between oblivious and non-oblivious opponent, and between regret and pseudo-regret.

As before, at stage t , the decision maker chooses $x_t \in \Delta_d$ and draws decision $d_t \in [d]$ according to x_t . The main difference with the previous framework is that the decision maker only observes his own outcome $\omega_t^{d_t}$ before choosing the next decision d_{t+1} .

5.1 Upper Bounds on the Regret with Sparse Losses

We shall focus in this section on s -sparse losses. The algorithm we consider belongs to the family of Greedy Online Mirror Descent. We follow (Bubeck and Cesa-Bianchi, 2012, Section 5) and refer to it for the detailed and rigorous construction. Let $F_q(x)$ be the Legendre function associated with the potential $\psi(x) = (-x)^{-q}$ ($q > 1$), i.e.,

$$F_q(x) = -\frac{q}{q-1} \sum_{i=1}^d (x^i)^{1-1/q}.$$

The algorithm, which depends on a parameter $\eta > 0$ to be fixed later, is defined as follows. Set $x_1 = (\frac{1}{d}, \dots, \frac{1}{d}) \in \Delta_d$. For all $t \geq 1$, we define the estimator $\hat{\ell}_t$ of ℓ_t as usual:

$$\hat{\ell}_t^{(i)} = \mathbb{1}_{\{d_t=i\}} \frac{\ell_t^{(i)}}{x_t^{(i)}}, \quad i \in [d],$$

which is then used to compute

$$z_{t+1} = \nabla F_q^* (\nabla F_q(x_t) - \eta \hat{\ell}_t) \quad \text{and} \quad x_{t+1} = \arg \min_{x \in \Delta_d} D_{F_q}(x, z_{t+1}),$$

where $D_{F_q} : \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}$ is the Bregman divergence associated with F_q :

$$D_{F_q}(x', x) = F_q(x') - F_q(x) - \langle \nabla F_q(x), x' - x \rangle.$$

Theorem 11 *Let $\eta > 0$ and $q > 1$. For any sequence of s -sparse loss vectors, the above strategy with parameter η guarantees, for $T \geq 1$:*

$$R_T \leq q \left(\frac{d^{1/q}}{\eta(q-1)} + \frac{\eta T s^{1-1/q}}{2} \right).$$

In particular, if $d/s \geq e^2$, the choices

$$\eta = \sqrt{\frac{2d^{1/q}}{(q-1)T s^{1-1/q}}} \quad \text{and} \quad q = \log(d/s)$$

yield the following regret bound:

$$R_T \leq 2\sqrt{e} \sqrt{Ts \log \frac{d}{s}}.$$

Proof The general regret bound for Greedy Online Mirror Descent (Bubeck and Cesa-Bianchi, 2012, Theorem 5.10) gives:

$$R_T \leq \frac{\max_{x \in \Delta_d} F(x) - F(x_1)}{\eta} + \frac{\eta}{2} \sum_{t=1}^T \sum_{i=1}^d \mathbb{E} \left[\frac{(\hat{\ell}_t^{(i)})^2}{(\psi^{-1})(x_t^{(i)})} \right],$$

with $(\psi^{-1})(x) = (q x^{1+1/q})^{-1}$. Let us bound the first term.

$$\frac{1}{\eta} \max_{x \in \Delta_d} F(x) - F(x_1) \leq \frac{1}{\eta} \frac{q}{q-1} \left(0 + d(1/d)^{1-1/q} \right) = \frac{qd^{1/q}}{\eta(q-1)}.$$

We turn to the second term. Let $1 \leq t \leq T$.

$$\begin{aligned} \sum_{i=1}^d \mathbb{E} \left[\frac{(\hat{\ell}_t^{(i)})^2}{(\psi^{-1})(x_t^{(i)})} \right] &= q \sum_{i=1}^d \mathbb{E} \left[\frac{(\hat{\ell}_t^{(i)})^2 (x_t^{(i)})^{1+1/q}}{(x_t^{(i)})^2} \right] \\ &= q \sum_{i=1}^d \mathbb{E} \left[\mathbb{1}_{\{d_t=i\}} \left[\frac{(\hat{\ell}_t^{(i)})^2 (x_t^{(i)})^{1+1/q}}{(x_t^{(i)})^2} \right] \right] \\ &= q \sum_{i=1}^d \mathbb{E} \left[\frac{(\hat{\ell}_t^{(i)})^2 (x_k^{(i)})^{1/q}}{(x_k^{(i)})^2} \right] \\ &= q \mathbb{E} \left[\sum_{[s \text{ terms}]_s} \frac{(\hat{\ell}_t^{(i)})^2 (x_t^{(i)})^{1/q}}{(x_t^{(i)})^2} \right] \\ &\leq qs(1/s)^{1/q} = qs^{1-1/q}, \end{aligned}$$

where we used the assumption that ℓ_t has at most s nonzero components, and the fact that $x_t \in \Delta_d$. The first regret bound is thus proven. By choosing $\eta = \sqrt{\frac{2s^{1-1/q}}{(q-1)Td^{1/q}}}$, we balance both terms and get:

$$R_T \leq 2q \sqrt{\frac{Td^{1/q}s^{1-1/q}}{2(q-1)}} = \sqrt{2q} \sqrt{Ts \left(\frac{d}{s} \right)^{1/q} \left(\frac{q}{q-1} \right)}.$$

If $d/s \geq e^2$ and $q = \log(d/s)$, then $q/(q-1) \leq 2$ and finally:

$$R_T \leq 2\sqrt{e} \sqrt{Ts \log \frac{d}{s}}. \quad \blacksquare$$

Remark 12 *The previous analysis cannot be carried in the case of gains because the bound from (Bubeck and Cesa-Bianchi, 2012, Theorem 5.10) that we use above only holds for nonnegative losses (and its proof strongly relies on this assumption). We are unaware of techniques which could provide a similar bound in the case of nonnegative gains.*

5.2 Matching Lower Bound

The following theorem establishes that the bound from Theorem 11 is optimal up to a logarithmic factor. We denote $\hat{v}_T^{s,d}$ the minimax regret in the bandit setting with losses.

Theorem 13 *For all $d \geq 2$, $s \in [d]$ and $T \geq d^2/4s$, the following lower bound holds:*

$$\hat{v}_T^{s,d} \geq \frac{1}{32} \sqrt{Ts}.$$

The intuition behind the proof is the following. Let us consider the case where $s = 1$ and assume that ℓ_t is a unit vector $e_{i_t} = (\mathbb{1}\{j = i_t\})_j$ where $\mathbb{P}(i_t = i) \simeq (1 + \varepsilon)/d$ for all $i \in [d]$, except one fixed coordinate i^* where $\mathbb{P}(i_t = i^*) \simeq 1/d - \varepsilon$.

Since $1/d$ goes to 0 as d increases, the Kullback-Leibler divergence between two Bernoulli of parameters $(1 + \varepsilon)/d$ and $1/d - \varepsilon$ is of order $d\varepsilon^2$. As a consequence, it would require approximately $1/d\varepsilon^2$ samples to distinguish between the two. The standard argument that one of the coordinates has not been chosen more than T/d times, yields that one should take $1/d\varepsilon^2 \simeq T/d$ so that the regret is of order $T\varepsilon$. This provides a lower bound of order \sqrt{T} . Similar arguments with $s > 1$ give a lower bound of order \sqrt{sT} .

We emphasize that one cannot simply assume that the s components with positive losses are chosen at the beginning once for all, and apply standard lower bound techniques. Indeed, with this additional information, the decision maker just has to choose, at each stage, a decision associated with a zero loss. His regret would then be uniformly bounded (or even possibly equal to zero).

5.3 Proof of Theorem 13

Let $d \geq 1$, $1 \leq s \leq d$, $T \geq 1$, and $\varepsilon \in (0, s/2d)$. Denote $\mathfrak{P}_s([d])$ the set of subsets of $[d]$ of cardinality s , δ_{ij} the Kronecker symbol, and $B(1, p)$ the Bernoulli distribution of parameter $p \in [0, 1]$. If P, Q are two probability distributions on the same set, $D(P \| Q)$ will denote the relative entropy of P and Q .

5.3.1 RANDOM s -SPARSE LOSS VECTORS ℓ_t AND ℓ_t^j

For $t \geq 1$, define the random s -sparse loss vectors $(\ell_t)_{t \geq 1}$ as follows. Draw Z uniformly from $[d]$. We will denote $\mathbb{P}_s[\cdot] = \mathbb{P}[\cdot | Z = i]$ and $\mathbb{E}_s[\cdot] = \mathbb{E}[\cdot | Z = i]$. Knowing $Z = i$, the random vectors ℓ_t are i.i.d and defined as follows. Draw I_t uniformly from $\mathfrak{P}_s([d])$. If $j \in I_t$, define $\ell_t^{(j)}$ such that:

$$\mathbb{P}_s[\ell_t^{(j)} = 1] = 1 - \mathbb{P}_s[\ell_t^{(j)} = 0] = \frac{1}{2} - \frac{\varepsilon d}{s} \delta_{ij}.$$

If $j \notin I_t$, set $\ell_t^{(j)} = 0$. Therefore, one can check that for each component $j \in [d]$ and all $t \geq 1$,

$$\mathbb{E}_s[\ell_t^{(j)}] = \frac{s}{2d} - \varepsilon \delta_{ij}.$$

For $t \geq 1$, define the i.i.d. random s -sparse loss vectors $(\ell_t^j)_{t \geq 1}$ as follows. Draw I_t uniformly from $\mathfrak{P}_s([d])$. Then if $j \in I_t$, set $(\ell_t^j)^{(i)}$ such that:

$$\mathbb{P}[(\ell_t^j)^{(i)} = 1] = \mathbb{P}[(\ell_t^j)^{(j)} = 0] = 1/2.$$

And if $j \notin I_t$, set $(\ell_t^j)^{(i)} = 0$. Therefore, one can check that for each component $j \in [d]$ and all $t \geq 1$,

$$\mathbb{E}[(\ell_t^j)^{(i)}] = \frac{s}{2d}.$$

By construction, ℓ_t and ℓ_t^j are indeed random s -sparse loss vectors.

5.3.2 A DETERMINISTIC STRATEGY σ FOR THE PLAYER

We assume given a deterministic strategy $\sigma = (\sigma_t)_{t \geq 1}$ for the player:

$$\sigma_t : ([d] \times [0, 1])^{t-1} \rightarrow [d].$$

Therefore,

$$d_t = \sigma_t(d_1, \omega_1^{(d_1)}, \dots, d_{t-1}, \omega_{t-1}^{(d_{t-1})}),$$

where d_t denotes the decision chosen by the strategy at stage t and ω_t the outcome vector of stage t . But since d_t is determined by previous decisions and outcomes, we can consider that σ_t only depends on the received outcomes:

$$\sigma_t : [0, 1]^{t-1} \rightarrow [d],$$

$$d_t = \sigma_t(\omega_1^{(d_1)}, \dots, \omega_{t-1}^{(d_{t-1})}).$$

We define d_t and d_t^j to be the (random) decisions played by deterministic strategy σ against the random loss vectors $(\ell_t)_{t \geq 1}$ and $(\ell_t^j)_{t \geq 1}$ respectively:

$$\begin{aligned} d_t &= \sigma_t(\ell_1^{(d_1)}, \dots, \ell_{t-1}^{(d_{t-1})}), \\ d_t^j &= \sigma_t((\ell_1^j)^{(d_1)}, \dots, (\ell_{t-1}^j)^{(d_{t-1})}). \end{aligned}$$

For $t \geq 1$ and $i \in [d]$, define $A_t^{(i)}$ to be the set of sequences of outcomes in $\{0, 1\}$ of the first $t-1$ stages for which strategy σ plays decision i at stage t :

$$A_t^{(i)} = \left\{ (u_1, \dots, u_{t-1}) \in \{0, 1\}^{t-1} \mid \sigma_t(u_1, \dots, u_{t-1}) = i \right\},$$

and $B_t^{(i)}$ the complement:

$$B_t^{(i)} = \{0, 1\}^{t-1} \setminus A_t^{(i)}.$$

Note that for a given $t \geq 1$, $(A_t^{(i)})_{i \in [d]}$ is a partition of $\{0, 1\}^{t-1}$ (with possibly some empty sets).

For $i \in [d]$, define $\tau_i(T)$ (resp. $\tau_i^j(T)$) to be the number of times decision i is played by strategy σ against loss vectors $(\ell_t)_{t \geq 1}$ (resp. against $(\ell_t^j)_{t \geq 1}$) between stages 1 and T :

$$\tau_i(T) = \sum_{t=1}^T \mathbb{1}_{\{d_t=i\}} \quad \text{and} \quad \tau_i^j(T) = \sum_{t=1}^T \mathbb{1}_{\{d_t^j=i\}}.$$

5.3.3 THE PROBABILITY DISTRIBUTIONS \mathbb{Q} AND \mathbb{Q}_i ($i \in [d]$) ON BINARY SEQUENCES

We consider binary sequences $\vec{u} = (u_1, \dots, u_T) \in \{0, 1\}^T$. We define \mathbb{Q} and \mathbb{Q}_i ($i \in [d]$) to be probability distributions on $\{0, 1\}^T$ as follows:

$$\begin{aligned} \mathbb{Q}_i[\vec{u}] &= \mathbb{P}_i \left[\ell_1^{(d_1)} = u_1, \dots, \ell_T^{(d_T)} = u_T \right], \\ \mathbb{Q}[\vec{u}] &= \mathbb{P} \left[(\ell_1^j)^{(d_1)} = u_1, \dots, (\ell_T^j)^{(d_T)} = u_T \right]. \end{aligned}$$

Fix $(u_1, \dots, u_{t-1}) \in \{0, 1\}^t$. The applications

$$u_t \mapsto \mathbb{Q}[u_t | u_1, \dots, u_{t-1}] \quad \text{and} \quad u_t \mapsto \mathbb{Q}_i[u_t | u_1, \dots, u_{t-1}],$$

are probability distributions on $\{0, 1\}$, which we now aim at identifying. The first one is Bernoulli of parameter $s/2d$. Indeed,

$$\begin{aligned} \mathbb{Q}[1 | u_1, \dots, u_{t-1}] &= \mathbb{P} \left[(\ell_t^j)^{(d_t)} = 1 \mid (\ell_1^j)^{(d_1)} = u_1, \dots, (\ell_{t-1}^j)^{(d_{t-1})} = u_{t-1} \right] \\ &= \mathbb{P} \left[(\ell_t^j)^{(d_t)} = 1 \right] \\ &= \mathbb{P} [d_t \in I_t] \mathbb{P} \left[(\ell_t^j)^{(d_t)} = 1 \mid d_t \in I_t \right] \\ &= \frac{s}{d} \times \frac{1}{2} \\ &= \frac{s}{2d}. \end{aligned}$$

where we used the independence of the random vectors $(\ell_t^i)_{i \geq 1}$ for the second inequality. We now turn to the second distribution, which depends on (u_1, \dots, u_{t-1}) . If $(u_1, \dots, u_{t-1}) \in A_t^{(i)}$, it is a Bernoulli of parameter $s/2d - \varepsilon$:

$$\begin{aligned} \mathbb{Q}_i[1 | u_1, \dots, u_{t-1}] &= \mathbb{P}_i \left[\ell_t^{(d_t)} = 1 \mid \ell_1^{(d_1)} = u_1, \dots, \ell_{t-1}^{(d_{t-1})} = u_{t-1} \right] \\ &= \mathbb{P}_i \left[\ell_t^{(i)} = 1 \mid \ell_1^{(d_1)} = u_1, \dots, \ell_{t-1}^{(d_{t-1})} = u_{t-1} \right] \\ &= \mathbb{P}_i \left[\ell_t^{(i)} = 1 \right] \\ &= \mathbb{P}_i [i \in I_t] \mathbb{P}_i \left[\ell_t^{(i)} = 1 \mid i \in I_t \right] \\ &= \frac{s}{d} \times \left(\frac{1}{2} - \frac{\varepsilon d}{s} \right) \\ &= \frac{s}{2d} - \varepsilon. \end{aligned}$$

where for the third inequality, we used the assumption that the random vectors $(\ell_t)_{t \geq 1}$ are independent under \mathbb{P}_i , i.e. knowing $Z = i$. On the other hand, if $(u_1, \dots, u_{t-1}) \in B_t^{(i)}$, we can prove similarly that the distribution is a Bernoulli of parameter $s/2d$.

5.3.4 COMPUTATION THE RELATIVE ENTROPY OF \mathbb{Q}_i AND \mathbb{Q}

We apply iteratively the chain rule to the relative entropy of $\mathbb{Q}[\vec{u}]$ and $\mathbb{Q}_i[\vec{u}]$. Using the short-hand $\mathbb{D}_i[\cdot] := D(\mathbb{Q}[\cdot] \parallel \mathbb{Q}_i[\cdot])$,

$$\begin{aligned} D(\mathbb{Q}[\vec{u}] \parallel \mathbb{Q}_i[\vec{u}]) &= \mathbb{D}_i[\vec{u}] \\ &= \mathbb{D}_i[u_1] + \mathbb{D}_i[u_2, \dots, u_T \mid u_1] \\ &= \mathbb{D}_i[u_1] + \mathbb{D}_i[u_2 \mid u_1] + \mathbb{D}_i[u_3, \dots, u_T \mid u_1, u_2] \\ &= \sum_{t=1}^T \mathbb{D}_i[u_t \mid u_1, \dots, u_{t-1}]. \end{aligned}$$

We now use the definition of the conditional relative entropy, and make the previously discussed Bernoulli distributions appear. For $1 \leq t \leq T$,

$$\begin{aligned} \mathbb{D}_i[u_t \mid u_1, \dots, u_{t-1}] &= \sum_{u_1, \dots, u_{t-1}} \mathbb{Q}[u_1, \dots, u_{t-1}] \\ &\quad \times \sum_{u_t} \mathbb{Q}[u_t \mid u_1, \dots, u_{t-1}] \log \frac{\mathbb{Q}[u_t \mid u_1, \dots, u_{t-1}]}{\mathbb{Q}_i[u_t \mid u_1, \dots, u_{t-1}]} \\ &= \frac{1}{2^{t-1}} \sum_{u_1, \dots, u_{t-1}} \sum_{u_t} \mathbb{Q}[u_t \mid u_1, \dots, u_{t-1}] \log \frac{\mathbb{Q}[u_t \mid u_1, \dots, u_{t-1}]}{\mathbb{Q}_i[u_t \mid u_1, \dots, u_{t-1}]} \\ &= \frac{1}{2^{t-1}} \sum_{(u_1, \dots, u_{t-1}) \in A_t^{(i)}} D\left(B\left(1, \frac{s}{2d}\right) \parallel B\left(1, \frac{s}{2d} - \varepsilon\right)\right) \\ &\quad + \frac{1}{2^{t-1}} \sum_{(u_1, \dots, u_{t-1}) \in B_t^{(i)}} D\left(B\left(1, \frac{s}{2d}\right) \parallel B\left(1, \frac{s}{2d}\right)\right) \\ &= \frac{1}{2^{t-1}} \sum_{(u_1, \dots, u_{t-1}) \in A_t^{(i)}} \mathbb{B}\left(\frac{s}{2d}, \varepsilon\right), \end{aligned}$$

where we used the short-hand $\mathbb{B}\left(\frac{s}{2d}, \varepsilon\right) := D\left(B\left(1, \frac{s}{2d}\right) \parallel B\left(1, \frac{s}{2d} - \varepsilon\right)\right)$. Eventually:

$$D(\mathbb{Q}[\vec{u}] \parallel \mathbb{Q}_i[\vec{u}]) = \mathbb{B}\left(\frac{s}{2d}, \varepsilon\right) \sum_{t=1}^T \frac{|A_t^{(i)}|}{2^{t-1}}.$$

5.3.5 UPPER BOUND ON $\frac{1}{d} \sum_{i=1}^d \mathbb{E}_i[\tau_i(T)]$ USING PINSKER'S INEQUALITY

In this step, we will make use of Pinsker's inequality to make the relative entropy appear.

Proposition 14 (Pinsker's inequality) *Let X be a finite set, and P, Q probability distributions on X . Then,*

$$\frac{1}{2} \sum_{x \in X} |P(x) - Q(x)| \leq \sqrt{\frac{1}{2} D(P \parallel Q)}.$$

Immediate consequence:

$$\sum_{\substack{x \in X \\ P(x) > Q(x)}} (P(x) - Q(x)) \leq \sqrt{\frac{1}{2} D(P \parallel Q)}.$$

Let $i \in [d]$. If $(u_1, \dots, u_T) \in \{0, 1\}^T$ is given, since the decisions d_t and d'_t are determined by the previous losses $\ell_t^{(d_t)}$ and $(\ell'_t)^{(d'_t)}$ respectively, we have in particular:

$$\mathbb{E}_i \left[\tau_i(T) \mid \ell_1^{(d_1)} = u_1, \dots, \ell_T^{(d_T)} = u_T \right] = \mathbb{E} \left[\tau'_i(T) \mid (\ell'_t)^{(d'_t)} = u_1, \dots, (\ell'_T)^{(d'_T)} = u_T \right].$$

Therefore,

$$\begin{aligned}
\mathbb{E}_t[\tau_t(T)] - \mathbb{E}[\tau_t'(T)] &= \sum_{\vec{u}} \mathbb{Q}_t[\vec{u}] \cdot \mathbb{E}_t[\tau_t(T) \mid \forall t, \ell_t^{(d)} = u_t] \\
&\quad - \sum_{\vec{u}} \mathbb{Q}[\vec{u}] \cdot \mathbb{E}[\tau_t'(T) \mid \forall t, (\ell_t^i)^{d_t} = u_t] \\
&= \sum_{\vec{u}} (\mathbb{Q}_t[\vec{u}] - \mathbb{Q}[\vec{u}]) \mathbb{E}_t[\tau_t(T) \mid \forall t, \ell_t^{(d)} = u_t] \\
&\leq \sum_{\mathbb{Q}_t[\vec{u}] > \mathbb{Q}[\vec{u}]} (\mathbb{Q}_t[\vec{u}] - \mathbb{Q}[\vec{u}]) \mathbb{E}_t[\tau_t(T) \mid \forall t, \ell_t^{(d)} = u_t] \\
&\leq T \sum_{\vec{u}} \frac{(\mathbb{Q}_t[\vec{u}] - \mathbb{Q}[\vec{u}])}{\mathbb{Q}_t[\vec{u}] > \mathbb{Q}[\vec{u}]} \\
&\leq T \sqrt{\frac{1}{2} D(\mathbb{Q}[\vec{u}] \parallel \mathbb{Q}_t[\vec{u}])} \\
&= T \sqrt{\frac{\mathbb{B}(s/2d, \varepsilon)}{2} \left[\sum_{t=1}^T \frac{|A_t^{(i)}|}{2^{t-1}} \right]},
\end{aligned}$$

where we used Pinsker's inequality in the fifth line. Moreover, we have:

$$\frac{1}{d} \sum_{i=1}^d \mathbb{E}[\tau_t'(T)] = \frac{1}{d} \mathbb{E} \left[\sum_{t=1}^T \sum_{i=1}^d \mathbb{1}_{\{d_t=i\}} \right] = \frac{1}{d} \mathbb{E} \left[\sum_{i=1}^T \mathbb{1} \right] = \frac{T}{d}.$$

Combining this with the previous inequality gives:

$$\begin{aligned}
\frac{1}{d} \sum_{i=1}^d \mathbb{E}_t[\tau_t(T)] &\leq \frac{1}{d} \sum_{i=1}^d \mathbb{E}[\tau_t'(T)] + T \sqrt{\frac{\mathbb{B}(s/2d, \varepsilon)}{2} \frac{1}{d} \sum_{i=1}^d \left[\sum_{t=1}^T \frac{|A_t^{(i)}|}{2^{t-1}} \right]} \\
&\leq \frac{T}{d} + T \sqrt{\frac{\mathbb{B}(s/2d, \varepsilon)}{2} \sqrt{\frac{1}{d} \sum_{t=1}^T \sum_{i=1}^d \frac{|A_t^{(i)}|}{2^{t-1}}}} \\
&= \frac{T}{d} + T \sqrt{\frac{\mathbb{B}(s/2d, \varepsilon)}{2} \frac{1}{d} \sum_{t=1}^T \frac{|\{0, 1\}^{t-1}|}{2^{t-1}}} \\
&= \frac{T}{d} + T \sqrt{\frac{\mathbb{B}(s/2d, \varepsilon)}{2} \sqrt{\frac{T}{T}}} \\
&= \frac{T}{d} + T^{3/2} \sqrt{\frac{\mathbb{B}(s/2d, \varepsilon)}{2d}}.
\end{aligned}$$

where we used Jensen for the second inequality, and for the third line, we remembered that $(A_t^{(i)})_{i \in [d]}$ is a partition of $\{0, 1\}^{t-1}$.

5.3.6 AN UPPER BOUND ON $\mathbb{B}(s/2d, \varepsilon)$ FOR SMALL ENOUGH ε

We first write $\mathbb{B}(s/2d, \varepsilon)$ explicitly.

$$\begin{aligned}
\mathbb{B}\left(\frac{s}{2d}, \varepsilon\right) &= D(B(1, s/2d) \parallel B(1, s/2d - \varepsilon)) \\
&= \frac{s}{2d} \log \frac{s/2d}{s/2d - \varepsilon} + \left(1 - \frac{s}{2d}\right) \log \frac{1 - s/2d}{1 - s/2d + \varepsilon} \\
&= -\frac{s}{2d} \log \left(1 - \frac{2d\varepsilon}{s}\right) + \left(\frac{s}{2d} - 1\right) \log \left(1 + \frac{\varepsilon}{1 - m/2d}\right).
\end{aligned}$$

We now bound the two logarithms from above using respectively the two following easy inequalities:

$$\begin{aligned}
-\log(1-x) &\leq x + x^2, \quad \text{for } x \in [0, 1/2] \\
-\log(1+x) &\leq -x + x^2, \quad \text{for } x \geq 0.
\end{aligned}$$

This gives:

$$\begin{aligned}
\mathbb{B}\left(\frac{s}{2d}, \varepsilon\right) &\leq \frac{s}{2d} \left(\frac{2d\varepsilon}{s} + \frac{4d^2\varepsilon^2}{s^2}\right) + \left(1 - \frac{s}{2d}\right) \left(-\frac{\varepsilon}{1 - s/2d} + \frac{\varepsilon^2}{(1 - s/2d)^2}\right) \\
&= \frac{4d^2\varepsilon^2}{s(2d-s)},
\end{aligned}$$

which holds for $2d\varepsilon/s \leq 1/2$, in other words, for $\varepsilon \leq s/4d$.

5.3.7 LOWER BOUND ON THE EXPECTATION OF THE REGRET OF σ AGAINST ℓ_t

We can now bound from below the expected regret incurred when playing σ against loss vectors $(\ell_t)_{t \geq 1}$. For $\varepsilon \leq s/4d$,

$$\begin{aligned}
R_T &= \mathbb{E} \left[\sum_{t=1}^T \ell_t^{(d_t)} - \min_{j \in [d]} \sum_{t=1}^T \ell_t^{(j)} \right] \\
&= \frac{1}{d} \sum_{t=1}^d \mathbb{E}_i \left[\sum_{t=1}^T \ell_t^{(d_t)} - \min_{j \in [d]} \sum_{t=1}^T \ell_t^{(j)} \right] \\
&\geq \frac{1}{d} \sum_{t=1}^d \left(\mathbb{E}_i \left[\sum_{t=1}^T \ell_t^{(d_t)} \right] - \min_{j \in [d]} \sum_{t=1}^T \mathbb{E}_i \left[\ell_t^{(j)} \right] \right) \\
&= \frac{1}{d} \sum_{t=1}^d \left(\mathbb{E}_i \left[\sum_{t=1}^T \ell_t^{(d_t)} \mid d_t \right] - T \min_{j \in [d]} \left(\frac{s}{2d} - \varepsilon \delta_{ij} \right) \right) \\
&= \frac{1}{d} \sum_{t=1}^d \left(\mathbb{E}_i \left[\sum_{t=1}^T \left(\frac{s}{2d} - \varepsilon \delta_{id_t} \right) \right] - T \left(\frac{s}{2d} - \varepsilon \right) \right) \\
&= \sum_{t=1}^d \varepsilon (T - \mathbb{E}_i [\tau_i(T)]) \\
&= \varepsilon \left(T - \frac{1}{d} \sum_i \mathbb{E}_i [\tau_i(T)] \right).
\end{aligned}$$

We now use the upper bound derived in Section 5.3.5.

$$\begin{aligned}
R_T &\geq \varepsilon \left(T - \frac{T}{d} - T^{3/2} \sqrt{\frac{\mathbb{B}(s/2d, \varepsilon)}{2d}} \right) \\
&\geq \varepsilon \left(T - \frac{T}{d} - T^{3/2} \varepsilon \sqrt{\frac{2d}{s(2d-s)}} \right) \\
&\geq \varepsilon \left(T - \frac{T}{d} - 2T^{3/2} \varepsilon \frac{1}{\sqrt{s}} \right),
\end{aligned}$$

where in the penultimate, we used the upper bound on $\mathbb{B}(s/2d, \varepsilon)$ that we established above, and in the last line, the fact that $s \leq d$. Let $C > 0$ and we choose $\varepsilon = C\sqrt{s}/T$. Then, for $\varepsilon \leq s/4d$,

$$\begin{aligned}
R_T &\geq \varepsilon T \left(1 - \frac{1}{d} - 2\varepsilon \sqrt{\frac{T}{s}} \right) \\
&= C\sqrt{sT} \left(1 - \frac{1}{d} \right) - 2\sqrt{sT}C^2 \\
&\geq \sqrt{sT} \left(\frac{C}{2} - 2C^2 \right),
\end{aligned}$$

where in the last line, we used the assumption $d \geq 2$. The choice $C = 1/8$ give:

$$R_T \geq \frac{1}{32} \sqrt{sT},$$

which holds for $\varepsilon = C\sqrt{s/T} \leq s/4d$ i.e. for $T \geq d^2/4s$.

The above inequality does not depend on σ . As it is a classic that a randomized strategy is equivalent to some random choice of deterministic strategies, this lower bound holds for any strategy of the player. In other words, for $T \geq d^2/4s$,

$$v_T^{\ell, s, d} \geq \frac{1}{32} \sqrt{sT}. \quad \blacksquare$$

5.4 Discussion

If the outcomes are not losses but gains, then there is an important discrepancy between the upper and lower bounds we obtain. Indeed, obtaining small losses regret bound as in the first displayed equation of the proof of Theorem 11 is still open. An idea for circumventing this issue would be to enforce exploration by perturbing x_t into $(1-\gamma)x_t + \gamma\mathcal{U}$ where \mathcal{U} is the uniform distribution over $[d]$, but usual computations show that the only obtainable upper bounds are of order of \sqrt{dT} . The aforementioned techniques used to bound the regret from below with losses would also work with gains, which would give a lower bound of order \sqrt{sT} . Therefore, finding the optimal dependency in the dimension and/or the sparsity level is still an open question in that specific case. We tend to believe that the upper bound could be improved: imagine the case $s = 1$, the restriction on the payoff vectors is huge, and we think that this could be taken advantage of. This would imply that there is no discrepancy between gains and losses, unlike the full information setting, which would be an interesting fact.

Acknowledgments

The authors are grateful to Guillaume Barraquand, Rida Laraki and Sylvain Sorin for helpful discussions and careful proofreading. V. Perchet is partially funded by the ANR grant ANR-13-JS01-0004-01; he also benefited from the support of the *FM/JH Program Gaspard Monge in optimization and operations research* (supported in part by EDF) and from the support of the CNRS through the PEPS projects.

References

- Yasin Abbasi-Yadkori, David Pal, and Csaba Szepesvri. Online-to-confidence-set conversions and application to sparse stochastic bandits. In *JMLR: Workshop and Conference Proceedings (AISTATS)*, volume 22, pages 1–9, 2012.
- Jean-Yves Audibert and Sébastien Bubeck. Minimax policies for adversarial and stochastic bandits. In *Proceedings of the 22nd Annual Conference on Learning Theory (COLT)*, pages 217–226, 2009.

- Jean-Yves Audibert, Sébastien Bubeck, and Gábor Lugosi. Regret in online combinatorial optimization. *Mathematics of Operations Research*, 39(1):31–45, 2013.
- Peter Auer, Nicolo Cesa-Bianchi, and Claudio Gentile. Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 64(1):48–75, 2002.
- Sébastien Bubeck. *Introduction to Online Optimization: Lecture Notes*. Princeton University, 2011.
- Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Machine Learning*, 5(1):1–122, 2012.
- Alexandra Carpentier and Rémi Munos. Bandit theory meets compressed sensing for high dimensional stochastic linear bandit. In *International Conference on Artificial Intelligence and Statistics*, pages 190–198, 2012.
- Nicolo Cesa-Bianchi. Analysis of two gradient-based algorithms for on-line regression. In *Proceedings of the Tenth Annual Conference on Computational Learning Theory (COLT)*, pages 163–170. ACM, 1997.
- Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.
- Nicolo Cesa-Bianchi, Yoav Freund, David Haussler, David P Helmbold, Robert E Schapire, and Manfred K Warmuth. How to use expert advice. *Journal of the ACM*, 44(3):427–485, 1997.
- Josip Dijolonga, Andreas Krause, and Volkan Cevher. High-dimensional gaussian process bandits. In *Advances in Neural Information Processing Systems (NIPS)*, volume 26, pages 1025–1033, 2013.
- Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.
- János Galambos. *The asymptotic theory of extreme order statistics*. John Wiley, New York, 1978.
- Sébastien Gerchinovitz. Sparsity regret bounds for individual sequences in online linear regression. *The Journal of Machine Learning Research*, 14(1):729–769, 2013.
- James Hannan. Approximation to bayes risk in repeated play. *Contributions to the Theory of Games*, 3(2):97–139, 1957.
- Eliad Hazan. The convex optimization approach to regret minimization. In S. Nowozin, S. Sra and S. Wright, editors, *Optimization for Machine Learning*, pages 287–303. MIT press, 2012.
- Sham M Kakade, Shai Shalev-Shwartz, and Anbuji Tewari. Regularization techniques for learning with matrices. *The Journal of Machine Learning Research*, 13(1):1865–1890, 2012.
- Joon Kwon and Panayotis Mertikopoulos. A continuous-time approach to online optimization. *arXiv preprint arXiv:1401.6956*, 2014.
- Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.
- Alexander Rakhlin and A Tewari. *Lecture notes on online learning*. University of Pennsylvania, 2008.
- Shai Shalev-Shwartz. *Online learning: Theory, algorithms, and applications*. PhD thesis, The Hebrew University of Jerusalem, 2007.
- Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.
- David Stepien. The one-sided barrier problem for gaussian noise. *Bell System Technical Journal*, 41(2):463–501, 1962.
- Volodimir G. Vovk. Aggregating strategies. In *Proceedings of the Third Workshop on Computational Learning Theory (COLT)*, pages 371–383. Morgan Kaufmann, 1990.

Linear Convergence of Randomized Feasible Descent Methods Under the Weak Strong Convexity Assumption

Chenxin Ma

Industrial and Systems Engineering
Lehigh University
Bethlehem, PA 18015, USA

CHM514@LEHIGH.EDU

Rachael Tappenden

Mathematics and Statistics
University of Canterbury
Christchurch 8041, New Zealand

RACHAEL.TAPPENDEN@CANTERBURY.AC.NZ

Martin Takáč

Industrial and Systems Engineering
Lehigh University
Bethlehem, PA 18015, USA

TAKAC.MT@GMAIL.COM

Editor: Leon Bottou

Abstract

In this paper we generalize the framework of the Feasible Descent Method (FDM) to a Randomized (R-FDM) and a Randomized Coordinate-wise Feasible Descent Method (RC-FDM) framework. We show that many machine learning algorithms, including the famous SDCA algorithm for optimizing the SVM dual problem, or the stochastic coordinate descent method for the LASSO problem, fits into the framework of RC-FDM. We prove linear convergence for both R-FDM and RC-FDM under the weak strong convexity assumption. Moreover, we show that the duality gap converges linearly for RC-FDM, which implies that the duality gap also converges linearly for SDCA applied to the SVM dual problem.

Keywords: feasible descent method, stochastic methods, iteration complexity, convergence theory, weak strong convexity

1. Introduction

In this paper we are interested in the following optimization problem

$$\min_{x \in X} f(x), \quad (1)$$

where the function f is smooth and convex, and $X \subseteq \mathbb{R}^n$ is a convex set. The Feasible Descent Method (FDM) (Luo and Tseng 1993; Necoara 2015; Wang and Lin 2014) is any algorithm, which produces a sequence of points $\{x_k\}_{k=0}^{\infty}$, where there exist constants $\beta \geq 0$, $\zeta > 0$ and $\omega_k \geq \bar{\omega} > 0$, such that for every iteration k , the following conditions are satisfied:

$$\begin{aligned} x_{k+1} &= \mathbf{Proj}_X(x_k - \omega_k \nabla f(x_k) + z_k), & (2) \\ \|z_k\| &\leq \beta \|x_k - x_{k+1}\|, & (3) \\ f(x_{k+1}) &\leq f(x_k) - \zeta \|x_k - x_{k+1}\|^2, & (4) \end{aligned}$$

where $\mathbf{Proj}_X(y) := \arg \min_{x \in X} \|x - y\|$ is the projection of y onto X .

As was shown in Luo and Tseng (1993), many first order algorithms, including steepest descent, the gradient projection algorithm, the extra gradient method, the proximal minimization algorithm and the cyclic coordinate descent method, fit into the framework of FDM. However, randomized first order algorithms are becoming increasingly popular in the optimization and machine learning literature, and the following question naturally arises:

“Can the framework of FDM be extended to a randomized setting?”

In this paper we give an affirmative answer to this question: we show that, indeed, a randomized version of FDM can be formulated and we will show that, for example, the inexact gradient projection algorithm (when the gradient is corrupted with random noise) or the stochastic coordinate descent method, fit into this new framework.

1.1 Assumptions and Notations

In this section we state the assumptions and introduce the notation that will be used in this paper. In particular, throughout this paper we will assume that f satisfies weak strong convexity, and that the gradient of f is Lipschitz.

Now we formalize the first assumption, which is that the function f enjoys weak strong convexity. Henceforth, we use \mathbb{R}_{++}^n to denote the set of vectors in \mathbb{R}^n , with (strictly) positive components, and we denote the i -th component of the vector x by $x^{(i)}$.

Assumption 1. We assume that there exists a positive vector $w \in \mathbb{R}_{++}^n$ such that the function $f(x)$ satisfies the weak strong convexity property on the set X , which is defined as

$$f(x) - f(\bar{x}) \geq \kappa_f \|x - \bar{x}\|_W^2, \quad \forall x \in X, \quad (5)$$

where $\kappa_f > 0$, $W = \mathbf{diag}(w)$, $\|x\|_W^2 = \sum_{i=1}^n w_i (x^{(i)})^2$, $f^* = \min_{x \in X} f(x)$, and

$$\bar{x} := \arg \min_{y \in X: f(y)=f^*} \|x - y\|_W. \quad (6)$$

Let us remark that, if f is smooth and has a Lipschitz continuous gradient, then Assumption 1 is weaker than the strong convexity assumption or the global error bound property, Necoara (2015). We now provide a few examples of functions that are used in machine learning, which are weakly strongly convex, but are not strongly convex.

1. In particular, let $x \in \mathbb{R}^n$ and let $c \in \mathbb{R}$. We have

$$\mathbf{shrink}_c(x) = \text{sign}(x) \max\{|x| - c, 0\}, \quad \text{and} \quad f(x) = \frac{1}{2} \|\mathbf{shrink}_c(x)\|_2^2,$$

where the shrinkage function is applied component-wise to vector x . Note that f is not strongly convex because $f(x) = 0$ for $x \in [-c, c]$ (which is the minimizer set). On the other hand, $f(x) = \frac{1}{2} \|x + c\|_2^2$ for $x \leq -c$ and $f(x) = \frac{1}{2} \|x - c\|_2^2$ for $x \geq c$. Thus, $f(x)$ is weakly strongly convex. See also Zhang and Yin (2013).

2. Another illustrative example of function which is weakly strongly convex but not strongly convex is $f(x) = \frac{1}{2} \|Ax - b\|^2$ with $A \in \mathbb{R}^{m \times n}$ such that $m < n$. If x^* is some optimal solution then $x^* + t$ is also optimal iff $t \in \text{null}(A)$. One can easily show that κ_f is related to the smallest non-negative singular value of matrix $A^T A$.

The second assumption we make regards the smoothness of f , and is defined precisely as follows.

Assumption 2. We assume that $f(x)$ has a coordinate-wise Lipschitz continuous gradient with constants L_i , i.e. $\forall x \in X$ and $\forall \delta \in \mathbb{R} : x + \delta e_i \in X$ the following inequality holds

$$|\nabla_i f(x) - \nabla_i f(x + \delta e_i)| \leq L_i |\delta|, \quad (7)$$

where e_i denotes the i -th column of the identity matrix $I \in \mathbb{R}^{n \times n}$.

As was shown in Richtárik and Takáč (2014), Assumption 2 implies that the function $f(x)$ has a Lipschitz continuous gradient with Lipschitz constant $L_f^W > 0$ with respect to the norm $\|\cdot\|_W$, i.e. $\forall x, y \in X$ we have

$$\|\nabla f(x) - \nabla f(y)\|_W^* \leq L_f^W \|x - y\|_W, \quad (8)$$

where $\|x\|_W^* = \sqrt{\sum_{i=1}^n \frac{1}{w_i} (x^{(i)})^2}$ is the dual norm to $\|\cdot\|_W$. Moreover, Richtárik and Takáč (2014) also showed that $L_f^W \leq \sum_{i=1}^n \frac{L_i}{w_i}$.

We define the projection operator onto the set X , with respect to the norm $\|\cdot\|_W$, as follows

$$\text{Proj}_X^W(x) = \arg \min_{y \in X} \|x - y\|_W^2 = \arg \min_{y \in X} \sum_{i=1}^n w_i (x^{(i)} - y^{(i)})^2. \quad (9)$$

1.2 Applications

In this section we discuss several problems that arise in the optimization and machine learning literature, which fit into the FDM framework that we analyze in this paper. We also provide details showing that, for each problem, the objective function satisfies the assumptions in Section 1.1. (A discussion regarding the value of the weak strong convexity parameter κ_f will be given in Section 4.)

The dual of SVM. Consider the classical linear SVM problem. The goal is, given n training points (a_i, y_i) , where $a_i \in \mathbb{R}^d$ are the features for point i and $y_i \in \{-1, +1\}$ is its label, find $w \in \mathbb{R}^d$ such that the regularized empirical loss function is minimized, i.e., solve the following optimization problem

$$\min_{w \in \mathbb{R}^d} \{\mathbf{P}(w) := \frac{1}{n} \sum_{i=1}^n \ell_i(w^T a_i) + \frac{\lambda}{2} \|w\|^2\}, \quad (10)$$

where $\lambda > 0$ is a regularization parameter, and, in the case of SVM, the function $\ell_i(w^T a_i) = \max\{0, 1 - y_i w^T a_i\}$ is the hinge loss. Clearly, the objective function (10) is not smooth. However, one can formulate the dual problem (Hsieh et al. 2008; Shalev-Shwartz and Zhang 2013; Takáč et al. 2013)

$$\min_{x \in \mathbb{R}^n, 0 \leq x^{(i)} \leq 1} \{f(x) := \frac{1}{2\lambda n^2} x^T Q x - \frac{1}{n} \mathbf{1}^T x\}, \quad (11)$$

where $Q_{i,j} = y_i y_j \langle a_i, a_j \rangle$, and $\mathbf{1}$ denotes the vector of all ones, which is smooth. The linear SVM problem (10) can now be solved via the dual problem (11). Note that (11) is

of the form (1), so our new FDM framework can be used to solve this important machine learning problem.

Lasso problem and least squares problem. Consider the following optimization problem

$$\min_{x \in \mathbb{R}^n} g(x) + \lambda \|x\|_1, \quad (12)$$

where $\lambda \geq 0$ and $g(x)$ is a smooth function with the special structure: $g(x) = h(Ax) + q^T x$, where $A \in \mathbb{R}^{m \times n}$ is some data matrix, $q \in \mathbb{R}^m$ is some vector and h is a strongly convex function. It is a simple exercise to show that, if we double the dimension of x to $[x^+, x^-]$, we can replace the term $\lambda \|x\|_1$ in (12) with $\lambda \mathbf{1}^T x^+ + \lambda \mathbf{1}^T x^-$ and impose the constraints $x^+, x^- \geq 0$. Then the Lasso problem (12) can be reformulated as a smooth optimization problem with simple box constraints.

ℓ_2 **regularized empirical loss minimization.** Many machine learning problems have the following structure (Chang et al. (2008))

$$\min_{x \in \mathbb{R}^n} f(x) = \frac{1}{n} \sum_{i=1}^n \ell_i(a_i^T x) + \frac{\lambda}{2} x^T x, \quad (13)$$

where $\lambda > 0$ is a regularization parameter and ℓ_i is a loss function. Because we assume that f must be smooth, the following commonly used loss functions fit our assumptions: the logistic loss function $\ell_i(a_i^T x) = \log(1 + \exp(-y_i a_i^T x))$; the squared loss function $\ell_i(a_i^T x) = (y_i - a_i^T x)^2$ and the squared hinge loss function $\ell_i(a_i^T x) = (\max\{0, 1 - y_i a_i^T x\})^2$. Hence, any machine learning problem of the form (13) (used with any of the mentioned loss functions) fits our randomized FDM framework.

1.3 Related work

Liu and Tseng (1993) are among the first to establish asymptotic linear convergence for a non-strongly convex problem under the local error bound property. They consider a class of feasible descent methods, which includes, for example, the cyclic coordinate descent method. The error bound measures how close the current solution is to the optimal solution set, with respect to the projected gradient. Recently, Wang and Lin (2014) proved that the feasible descent method enjoys a linear convergence rate (from the beginning, rather than only locally) under the global error bound property. Considering the class of smooth constrained optimization problems with the global error bound property, Necoara and Clipci (2016); Necoara and Nedelcu (2014a) showed a linear convergence rate for the parallel version of the stochastic coordinate descent method. Liu and Wright (2015) analyzed the asynchronous stochastic coordinate descent method (SCDM) under the weak strong convexity assumption. Very recently, Necoara (2015) showed that, if the objective function is smooth, then the class of problems with the global error bound property is a subset of the class of problems with the weak strong convexity property.

1.4 Contributions

In this section we list the most important contributions of this paper (not in order of their significance):

- **Randomized and Randomized Coordinate Feasible Descent Methods.** We extend the well known framework of Feasible Descent Methods (FDM) (Luo and Tseng 1993) to randomized and randomized coordinate FDM and show that the SCDM algorithm fits into our new proposed framework.

- **Linear Convergence Rate.** We show that any stochastic or deterministic algorithm, which fits our Randomized FDM (R-FDM) or Randomized Coordinate-FDM (RC-FDM) framework and satisfies our previously stated assumptions, converges linearly in expectation.

- **Linear Convergence of the Duality Gap for SDCA for SVM.** As a consequence of our analysis, we show that when SDCA is applied to the dual of the SVM problem, the duality gap converges linearly. Previously, linear convergence of the duality gap was only proven in case when the matrix Q in (11) is positive definite (Shalev-Shwartz and Zhang 2013; Takáč et al. 2015). However, our new linear convergence result holds, even when Q is singular.

- **Inexact Randomized Coordinate Descent.** By the nature of the FDM framework, *inexact* first order methods belong to the class of FDMs, (where inexact methods are methods that incorporate some kind of inexact information, for example, via inexact gradients, or via inexact updates). Our new randomized coordinate FDM framework includes inexact randomized coordinate descent methods. Therefore, another contribution of this work is that it provides a linear convergence rate for e.g. randomized coordinate descent with *inexact* computations of (partial) gradient, which was analyzed in various settings. (See, for example, Devolder et al. 2014; Bonettini 2011; Tappenden et al. 2016; Hua and Yamashita 2012; Necoara and Nedelcu 2014b.)

- **Flexibility and wide applicability.** Our randomized- and randomized coordinate-FDM framework is extremely *flexible*. It is a general framework that not only covers and unifies many existing algorithms, but any algorithm that fits our framework is also covered by the FDM convergence guarantees. Moreover, as demonstrated in Section 1.2, a very wide range of optimization and machine learning problems can be written in the form (1), and subsequently, they can be solved via the new FDM framework. Problems include the dual of SVM, the LASSO problem, and any ℓ_2 -regularized empirical loss functions where the loss function is smooth and separable. All such problems appear very frequently in the machine learning literature.

- **Parallel methods.** The RC-FDM framework is sufficiently general so as to include parallel randomized coordinate descent methods.

1.5 Paper Outline

In Section 2 we derive the Randomized (R-FDM) and the Randomized Coordinate (RC-FDM) Feasible Descent Methods. In Section 3 we derive the convergence rate for any method which fits into the R-FDM or RC-FDM framework and we compare our results with those in Liu and Wright (2015) for SCDM. In Section 4 we briefly review the global error bound property and using the result in Necoara (2015) we compare our convergence

results with Wang and Lin (2014). In Section 5 we show that the duality gap converges linearly for SDCA applied to the dual of the SVM problem, and in Section 6 we present a brief summary.

2. Randomized and Randomized Coordinate Feasible Descent Method

The framework of Feasible Descent Methods (FDM) broadly covers many algorithms that use first-order information (Luo and Tseng 1993) including gradient descent, cyclic coordinate descent and also the inexact gradient descent algorithm. We generalize the classical FDM framework to a randomized setting, which we call the Randomized Feasible Descent Method (R-FDM). Algorithms that use randomization have become extremely popular over the past few years, and the success, reliability, scalability, applicability and efficiency of such random algorithms is well documented. To the best of our knowledge this is the first time such a unifying R-FDM framework has been proposed and that a global linear convergence rate has been established under Assumptions 1 and 2. Further, we also show that the popular minibatch stochastic coordinate descent/ascent method, fits into the R-FDM framework.

Definition 3 (Randomized Feasible Descent Method (R-FDM)). *A sequence $\{x_k\}_{k=0}^{\infty}$ is generated by R-FDM if there exist $\beta \geq 0$, $\zeta > 0$ and $\{\omega_k\}_{k=0}^{\infty}$ with $\min_k \omega_k \geq \bar{\omega} > 0$ such that for every iteration k , the following conditions are satisfied,*

$$x_{k+1} = \mathbf{Proj}_X(x_k - \omega_k W^{-1}(\nabla f(x_k) - z_k)), \quad (14)$$

$$\mathbf{E}[\|z_k\|_W^2] \leq \beta^2 \mathbf{E}[\|x_k - x_{k+1}\|_W^2], \quad (15)$$

$$\mathbf{E}[f(x_{k+1})] \leq f(x_k) - \zeta \mathbf{E}[\|x_k - x_{k+1}\|_W^2], \quad (16)$$

where z_k is some random vector, which satisfies the Markov property, conditioned on x_k .

We will now compare the new Randomized FDM framework (Definition 3) with the original FDM ((2)–(4)), where, for simplicity of exposition, we will take $\|\cdot\|_W \equiv \|\cdot\|_2$ (i.e., $W = I$). Notice that the first step of R-FDM (14) is the same as the first step of FDM (2); it is a projected (gradient) descent type step. Note the role that z_k plays in (14); it captures any error/inexactness/noise in the update step, and it is clear to see that if $z_k = 0$ for all k , (i.e., no inexactness) then (14) is the same update as in a projected gradient descent method. Next, (15) gives an upper bound for $\|z_k\|_W^2$, which shows that, in order to guarantee convergence, the noise in (14) cannot be arbitrarily large, which intuitively makes sense. Finally, (16) guarantees that there is a reduction in the objective value, in expectation, after each iteration. The key difference between FDM and R-FDM is that for FDM, (3) and (4) hold deterministically (with a deterministic vector z_k), whereas for R-FDM (3) and (4) only need to hold *in expectation*. That is, for R-FDM, conditions (3) and (4) are replaced by conditions (15) and (16), where z_k is a random vector. Notice that (15) and (16) are weaker conditions than (3) and (4). That is, for FDM, (3) and (4) must hold at every iteration (i.e., they are deterministic), whereas for the R-FDM framework, the conditions (15) and (16) are equivalent to (3) and (4) holding *only on average*. The R-FDM framework is *extremely general*. It encapsulates algorithms that involve a (possibly noisy) projection step, has small enough noise *on average* and decreases the objective function *on average*, as the iterations progress.

Remark 4. We will see later (in the proof of convergence of R-FDM) that (15) can be related to the existence of constant $\eta > 0$ such that $\mathbf{E}[\|z_k\|_W^2] \leq \eta (f(x_k) - \mathbf{E}[f(x_{k+1})])^2$.

We will now demonstrate that (see Theorem 7), under an additional mild assumption, if the set $X = \mathbb{R}^n$, then SCDM (captured in Algorithm 1 with Option I.) is equivalent to R-FDM. We also remark that there is a need to modify R-FDM so that the stochastic coordinate descent method can be analyzed even when $X \neq \mathbb{R}^n$. However, first we describe SCDM and make the following assumption in order to establish the equivalence of SCDM with $X = \mathbb{R}^n$ and R-FDM.

Algorithm 1 Stochastic Coordinate Descent Method (SCDM)

- 1: **Input:** $f(x), \{\omega_k\}_{k=0}^{\infty}$: diagonal matrix $W \succ 0, x_0$.
 - 2: **Input:** $X = X_1 \times \dots \times X_n$, where $X_i = [a, b]$ with $-\infty \leq a < b \leq +\infty$
 - 3: **while** $k \geq 0$: **do**
 - 4: choose $i \in \{1, 2, \dots, n\}$ uniformly at random
 - 5: set $x_{k+1} = x_k$
 - 6: Option I: $x_{k+1} = \arg \min_{x^{(i)} \in X_i} f(x_k^{(1)}, x_k^{(2)}, \dots, x_k^{(i-1)}, x_k^{(i)}, x_k^{(i+1)}, \dots, x_k^{(n)})^T$
 - 7: Option II: $x_{k+1} = \text{Proj}_X^W(x_k - \omega_k W^{-1} \nabla_i f(x_k) e_i)$
 - 8: **end while**
-

Remark 5. For simplicity of exposition, Algorithm 1 is the serial form of SCDM, although a minibatch version of SCDM does exist. We will see in Definition 9 that our RC-FDM framework is flexible and general, because it also works in the parallel/minibatch case.

Assumption 6. The function f is coordinate-wise strongly convex with respect to the norm $\|\cdot\|_W$ with parameter $\gamma > 0$, if, for any $x \in X$ and any $i \in \{1, 2, \dots, n\}$ we have

$$f(x^{(1)}, \dots, x^{(i-1)}, \zeta, x^{(i+1)}, \dots, x^{(n)}) - f(x) + \nabla_i f(x)(x^{(i)} - \zeta) \geq \gamma \omega_i |\zeta - x^{(i)}|^2. \quad (17)$$

Note that Assumption 6 does not imply strong convexity of the function f . For example, (17) is satisfied for the Lasso problem or for the SVM dual problem whenever $\forall i: \|a_i\| > 0$, and neither of those problems is strongly convex.

Theorem 7. Let Assumptions 1, 2 and 6 hold. If $X = \mathbb{R}^n$ then the Stochastic Coordinate Descent Method (SCDM) (Algorithm 1 with Option I.) is equivalent to R-FDM with the parameters $\beta^2 = 2[(L_f^W)^2 + 1] + (n-1)r^2$, $\zeta = \gamma$ and $\omega_k = 1$, where $r^2 = \max_{\frac{L_f^2}{\omega_i^2}}$.

Let us comment that, if importance sampling is incorporated into SCDM, the convergence rate in Theorem 7 can be slightly improved, as the parameter r can be made to be $r^2 = \frac{1}{n} \sum_i \frac{L_i^2}{\omega_i^2}$, i.e., ‘average’ rather than ‘max’. However, it is typical to consider the case $\omega_i = L_i$ so that $r^2 = 1$ in Theorem 7 regardless.

The following remark compares the result of the above theorem with the cyclic rule.

Remark 8. It was shown in Luo and Tseng (1993) that for the cyclic coordinate descent method (which is not randomized and hence Equation 14-16 hold deterministically) we have

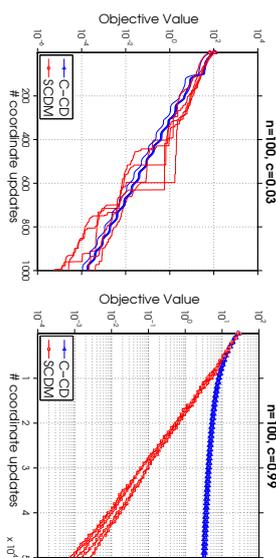


Figure 1: Number of coordinate updates v.s. objective gap for two methods.

$\omega_k^{\text{cyclic}} = 1$, $\zeta^{\text{cyclic}} = \gamma$ and $(\beta^{\text{cyclic}})^2 = (1 + \sqrt{n})L_f^W)^2 = 1 + 2\sqrt{n}L_f^W + n(L_f^W)^2$. For simplicity, let us assume that $W = \text{diag}(L_1, L_2, \dots, L_n)$. Then $r^2 = 1$ and $L_f^W \in [1, n]$. For the cyclic coordinate descent method and SCDM, ω_k and ζ are the same. However, if we consider the worst case (when $L_f^W = n$) we have that $\beta^2 \sim \mathcal{O}(n^2)$, whereas $(\beta^{\text{cyclic}})^2 \sim \mathcal{O}(n^3)$. Also note that one iteration of cyclic coordinate descent requires n coordinate updates, whereas SCDM updates just one coordinate, and therefore each iteration of SCDM is n times cheaper. In the other extreme, when $L_f^W = 1$ we have that both $\beta^2 \sim (\beta^{\text{cyclic}})^2 \sim \mathcal{O}(n)$, but again we recall that one iteration of SCDM is n times cheaper.

We present two experiments to support the discussion above. We apply both the cyclic coordinate descent method and SCDM to the problem

$$\min_{x \in \mathbb{R}^n} f(x) = x^T A x,$$

where the matrix $A \in \mathbb{R}^{n \times n}$ has ones on the diagonal and constant c elsewhere, Sun and Ye (2016). The optimal solution to this problem is $x = 0$. For the first experiment we set $n = 100$ and $c = 0.03$ ($L_f^W = 3.97 \approx 1$), and in the second experiment we keep n unchanged and set $c = 0.99$ ($L_f^W = 99.01 \approx n$). For each method we randomly select five starting points. From Figure 1, it is easy to see that when c is large (i.e., $L_f^W \approx n$) SCDM performs much better than cyclic coordinate descent. On the other hand, there is not such an obvious difference between the two methods when c is small. Thus, the difference in performance between the two methods depends upon the parameter L_f^W . Moreover, the case where c is small is more friendly for both methods, since they require far fewer coordinate updates to reach optimality, compared with the large c case. These results highlight and support Remark 8, regarding the theoretical gap between two methods.

It turns out that if $X \neq \mathbb{R}^n$ then SCDM does not fit the R-FDM framework because $\nabla_i f(x_k)$ cannot be bounded by $\|x_k - x_{k+1}\|_W$, as is shown in the proof of Theorem 7. Thus, there is a need to modify R-FDM such that the SCDM algorithm can be analyzed for bounded problems.

The natural modification to R-FDM, which would allow SCDM to fit the R-FDM framework is the following: at each iteration k we require that in (14), only a subset of coordinates of the vector x_k are updated. This can be achieved by the following method.

Definition 9. [Randomized Coordinate Feasible Descent Method (RC-FDM)] Let $X = X_1 \times \dots \times X_n$, where X_i are intervals. A sequence $\{x_k\}_{k=0}^{\infty}$ is generated by RC-FDM if there exists $\beta \geq 0$, $\zeta > 0$ and $\{\omega_k\}_{k=0}^{\infty}$ with $\min_k \omega_k \geq \bar{\omega} > 0$ such that for every iteration k , the following are satisfied

$$x_{k+1} = \mathbf{Proj}_X^W(x_k - \omega_k W^{-1}(\nabla f(x_k) - z_k)_{\mathcal{I}}), \quad (18)$$

$$(\|(z_k)_{\mathcal{I}}\|_W^*)^2 \leq \beta^2 \|x_k - x_{k+1}\|_W^2, \quad (19)$$

$$f(x_{k+1}) \leq f(x_k) - \zeta \|x_k - x_{k+1}\|_W^2, \quad (20)$$

where \mathcal{I} is a set of coordinates that are selected uniformly at random from the set $\{1, 2, \dots, n\}$ with $|\mathcal{I}| = \tau$, where $1 \leq \tau \leq n$, $x_{\mathcal{I}}$ is a vector whose elements $j \notin \mathcal{I}$ are set to 0 and z_k is some fixed vector at iteration k .

Now, we show that even if $X \neq \mathbb{R}^n$, SCDM fits the RC-FDM. Theorem 10 holds if Option I. is used in Algorithm 1 and Theorem 11 holds if Option II. is used.

Theorem 10. Let Assumptions 1, 2 and 6 hold. Let $\tau = 1$ for simplicity, if $X = X_1 \times \dots \times X_n$, where X_i are intervals then the Stochastic Coordinate Descent Method in Algorithm 1 with Option I. is RC-FDM with $\beta^2 = 2[(L_f^W)^2 + 1]$, $\zeta = \gamma$, and $\omega_k = 1$.

Theorem 11. Let Assumptions 1, 2 and 6 hold. Let $\tau = 1$ for simplicity, if $X = X_1 \times \dots \times X_n$, where X_i are intervals then the Stochastic Coordinate Descent Method in Algorithm 1 with Option II. is RC-FDM with $z_k = 0$, $\zeta = \gamma$, $\beta = 0$, $\omega_k = 1$, and $W = \text{diag}(L_1, L_2, \dots, L_n)$.

3. Convergence Analysis

Necoara (2015) proved a linear convergence rate for FDM under Assumptions 1 and 2. The following theorem shows that a linear convergence rate can also be established for R-FDM.

Theorem 12 (Linear Convergence of R-FDM). Let Assumptions 1 and 2 hold. If the sequence $\{x_k\}_{k=0}^{\infty}$ is produced by R-FDM, i.e. (14)-(16) are satisfied, then

$$\mathbf{E}[f(x_k) - f^*] \leq \left(\frac{c}{1+c}\right)^k (f(x_0) - f^*), \quad (21)$$

where

$$c = \frac{2}{\kappa_f \zeta} \left((L_f^W + \frac{1}{\bar{\omega}})^2 + \beta^2 \right). \quad (22)$$

The next theorem establishes a linear convergence rate for RC-FDM.

Theorem 13 (Linear Convergence of RC-FDM). Let $X = X_1 \times \dots \times X_n$, where X_i are intervals. Further, let Assumptions 1 and 2 hold, and let the sequence $\{x_k\}_{k=0}^{\infty}$ be produced by RC-FDM, i.e. (18)-(20) are satisfied. Then, for $z_k \neq 0$, there exists $c \in (0, 1)$ such that, for all k ,

$$\mathbf{E}[f(x_k) - f^*] \leq (1-c)^k (f(x_0) - f^*). \quad (23)$$

Moreover, if for all k we have $z_k \equiv 0$, and $\frac{L_i}{\omega_k} \geq \max_i \frac{L_i}{\omega_k}$, then $c = \frac{2\bar{\omega}\tau}{n(2\bar{\omega}+1)}$ with

$$\mathbf{E}[f(x_k) - f^*] \leq (1-c)^k \left(f(x_0) - f^* + \frac{\tau}{2\bar{\omega}} \|x_0 - \bar{x}_0\|_W^2 \right). \quad (24)$$

3.1 Comparison with the Results in Related Literature

In Theorem 13 we established a linear rate of convergence for RC-FDM for any z_k . We will now compare our result with the one presented in Liu and Wright (2015) for the projected coordinate gradient descent algorithm, and also with the result presented in Necoara (2015) for deterministic FDM. For this comparison we will assume that $\tau = 1$ (i.e., for serial RC-FDM), because the first paper only considers a serial algorithm, and for ease of comparison with the results in the second paper. (However, RC-FDM works for general $1 \leq \tau \leq n$.)

The projected coordinate gradient descent algorithm (Liu and Wright 2015) fits the RC-FDM framework. We also note that the result in Liu and Wright (2015) only holds for $z_k = 0$, so our result is more general. Further, even though the paper Liu and Wright (2015) considers an asynchronous implementation, where the update computed at iteration k is based on gradient information at a point up to ν iterations old, if $\nu = 0$ then their method fits into the RC-FDM framework. One of the benefits of our work is that more general norms can be used. So, for simplicity, and to match with the work in Liu and Wright (2015), let us assume that $L_i = 1$ for all i and we also choose $w_i = 1$ for all i . (This is the case e.g. for the SVM dual problem). The geometric rate in (24) in our work is then $1 - \frac{\kappa_f}{n(\kappa_f + \frac{1}{\bar{\omega}})}$ and from Theorem 4.1 in Liu and Wright (2015) for $\nu = 0$ we obtain that the geometric rate is $1 - \frac{\kappa_f}{n(\kappa_f + L_{\max})}$, where $L_{\max} \geq 1$ is such that

$$\|\nabla f(x) - \nabla f(x + \delta e_i)\|_{\infty} \leq L_{\max} |\delta|$$

holds $\forall x \in \mathbb{R}^n$, $\delta \in \mathbb{R}$ and $i \in \{1, 2, \dots, n\}$. Hence, in this case our convergence results are better because $\frac{1}{2} \leq 1 \leq L_{\max}$.

In Necoara (2015) the author provided a linear convergence rate for deterministic FDM. It is shown in Theorem 3.2 in Necoara (2015) that the coefficient of the linear rate is $1 - \frac{\zeta}{\zeta + \bar{\rho}}$ where $\bar{\rho} = \frac{1}{\kappa_f} (L_f + \frac{1}{\bar{\omega}} + \beta)^2$ whereas, in Theorem 13 of this work, from (21) we see that the coefficient is the same but with a different ρ . To be precise, in our case we have $\bar{\rho} = \frac{2}{\kappa_f} \left((L_f^W + \frac{1}{\bar{\omega}})^2 + \beta^2 \right)$. Our result can be better or worse than that in Necoara (2015), depending on the values of L_f^W , $\bar{\omega}$ and β , but our results holds for R-FDM, which is broader than FDM.

4. Global Error Bound Property

In this section we describe a class of problems that satisfies the Global Error Bound (GEB) property. We show that this implies the weak strong convexity property and we compare the convergence rate obtained in this paper with several results in the current literature derived for problems obeying the GEB. We begin by defining the projected gradient.

Definition 14 (Projected Gradient). For any $x \in \mathbb{R}^n$ let us define the projected gradient as follows,

$$\nabla^+ f(x) := x - \mathbf{Proj}_X^W(x - \nabla f(x)). \quad (25)$$

Note that projected gradient is zero at x if and only if x is an optimal solution of (1). Also, we will employ the projected gradient to define an error bound, which measures the distance between x and the optimal solution. Now, we are ready to define a global error bound as follows.

Definition 15 (Definition 6 in Wang and Lin 2014). *An optimization problem admits a global error bound if there is a constant $\eta_f \geq 0$ such that*

$$\|x - \bar{x}\| \leq \eta_f \|\nabla^+ f(x)\|_W^*, \quad \forall x \in X, \quad (26)$$

where \bar{x} and $\nabla^+ f(x)$ are defined in (6) and (25), respectively. A relaxed condition called the global error bound from the beginning is if the above inequality holds only for $x \in X$ such that $f(x) - f(\bar{x}) \leq M$, where M is a constant, and usually we have that $M = f(x_0) - f^*$.

Let us consider a special instance of (1) when X is a polyhedral set, i.e.

$$X = \{x \in \mathbb{R}^n : Bx \leq c\}, \quad (27)$$

and the function f has the following structure

$$f(x) = h(Ax) + q^T x, \quad (28)$$

where $B \in \mathbb{R}^{k \times n}$, $A \in \mathbb{R}^{d \times n}$, h is a σ_h strongly convex function and f satisfies Assumption 2. We also assume that there exists an optimal solution and hence the optimal solution set X^* is assumed to be non-empty, Wang and Lin (2014). It is easy to observe that if f is strongly convex, then (5) is trivially satisfied. Just recently, Necora (2015) showed that if (26) is satisfied, then (5) is satisfied with

$$\kappa_f = \frac{L_f^W}{2\eta_f^2}. \quad (29)$$

For problem (28) it was discussed in Wang and Lin (2014) that

$$\eta_f = \theta^2 (1 + L_f^W) \left(\frac{1 + 2\|\nabla h(A\bar{x})\|^2}{\sigma_h} + 4M \right) + 2\theta \|\nabla f(\bar{x})\|, \quad (30)$$

where θ is a constant from the Hoffman bound (Hoffman 1952; Li 1993; Robinson 1973) defined as follows

$$\theta := \sup_{u,v} \left\{ \begin{array}{l} \left\| \begin{pmatrix} u \\ v \end{pmatrix} \right\| \\ \left\| B^T u + \begin{pmatrix} A \\ q^T \end{pmatrix} v \right\| = 1, u \geq 0 \end{array} \right\} \quad \text{and the corresponding rows of } B, A \text{ to } u, v \text{'s} \\ \left. \begin{array}{l} \text{non-zero elements are linearly independent.} \end{array} \right\}. \quad (31)$$

Note that the constant θ can be very large; we will discuss this in Section 5.

Necora (2015) derived that, for problem (28), the weak strong convexity property (5) holds with

$$\kappa_f = \frac{\sigma_h}{2\theta^2}. \quad (32)$$

Note that κ_f given in (32) is $\mathcal{O}(\theta^{-2})$ whereas κ_f obtained from (29) is of the order θ^{-4} . Therefore we will compare our results using the latter estimates of κ_f .

5.1 Comparison with the Results in Related Literature

In Theorem 8 in Wang and Lin (2014), under the global error bound property, it is proven that FDM converges at a linear rate: $f(x_{k+1}) - f^* \leq (1 - \frac{1}{1+\zeta})(f(x_k) - f^*)$, with¹

$$\begin{aligned} \bar{c} &= \frac{1}{\zeta} (L_f^W + \frac{1}{\omega} + \beta)(1 + \eta_f(\frac{1}{\omega} + \beta)) = \frac{1}{\zeta} (L_f^W + \frac{1}{\omega} + \beta)(1 + \theta^2 \frac{1 + L_f^W}{\omega} (\frac{1}{\omega} + \beta)) \\ &\sim \mathcal{O} \left(\frac{\theta^2}{\zeta \sigma_h} (1 + L_f^W) (\frac{1}{\omega} + \beta)(L_f^W + \frac{1}{\omega} + \beta) \right). \end{aligned}$$

From Theorem 12 in this work, we have linear convergence of RC-FDM with the coefficient

$$c = \frac{2}{\kappa_f \zeta} \left((L_f^W + \frac{1}{\omega})^2 + \beta^2 \right) \stackrel{(32)}{=} \frac{4\theta^2}{\sigma_h \zeta} \left((L_f^W + \frac{1}{\omega})^2 + \beta^2 \right).$$

These coefficients are very similar, but FDM Wang and Lin (2014) covers only cyclic coordinate descent and not a randomized coordinate descent method (which is covered by Theorem 12).

5. Linear Convergence Rate of SDCA for Dual of SVM

In this section we show that the SDCA algorithm (which is SCDM applied to Equation 11) achieves a linear convergence rate for the duality gap. This improves upon the result obtained in Shalev-Shwartz and Zhang (2013); Takáč et al. (2015); Takáč et al. (2013), where only a sublinear rate was derived.

Assume, for simplicity, that in problem (10) for all $i \in \{1, 2, \dots, n\}$ it holds that $\|a_i\| \leq 1$. Then, from Takáč et al. (2015); Takáč et al. (2013), we have that for any $x \in \mathbb{R}^n$, $s \in [0, 1]$ and the function f defined in (11),

$$f(x) - f^* \geq sG(x) - s^2 \frac{\sigma^2}{2\lambda}, \quad (33)$$

where f^* denotes the optimal value of (11), $A = [a_1, a_2, \dots, a_n]$, $\sigma^2 = \frac{1}{n} \|X\| \in [\frac{1}{n}, 1]$ and $G(x)$ is the duality gap at the point x , which is defined as $G(x) := P(\frac{1}{\lambda n} Ax) + f(x)$.

We remark that SDCA for problem (11) is equivalent to RC-FDM, where the constants in (18)-(20) are: $z_k = 0$, $\beta^2 = 0$, $w_i = L_i = \frac{1}{\lambda n^2} \|a_i\|^2$, and $\omega_k = 1$. Hence, if we choose $x_0 = \mathbf{0}$ then from Theorem 13 we have that $\mathbf{E}[f(x_k) - f^*] \leq (1 - c)^k (f(\mathbf{0}) - f^* + \|x^*\|_2^2)$ with $c = \frac{2\kappa_f}{n(2\kappa_f + 1)}$.

Now, we see that rearranging (33) gives

$$G(x) \stackrel{(33)}{\leq} s \frac{\sigma^2}{2\lambda} + \frac{1}{s} (f(x) - f^*). \quad (34)$$

If we want to achieve $G(x) \leq \epsilon$ it is sufficient to choose both terms on right hand side of (34) to be $\leq \frac{\epsilon}{2}$. Hence, we can set $s = \min\{1, \frac{\epsilon}{2}\}$. All we have to do now is to choose k such that $f(x_k) - f^* \leq s \frac{\epsilon}{2}$. In the following theorem we establish linear convergence of the duality gap $G(x)$ for the SDCA algorithm.

1. In Wang and Lin (2014) it is shown that, in special cases (e.g. $X = \mathbb{R}^n$), (30) is $\eta_f = \theta^{2+L_f^W/\sigma_h}$.

Theorem 16. Let $s = \min\{1, \frac{1}{c}\sigma^2\}$ and let K be such that

$$K \geq n \left(1 + \frac{1}{2\kappa_f}\right) \log \frac{2(f(\mathbf{0}) - f^* + \|x^*\|_L^2)}{s\epsilon}.$$

Then if the SDCA algorithm is applied to problem (11) to produce $\{x_k\}_{k=0}^\infty$, then $\forall k \geq K$ we have that $\mathbf{E}[G(x_k)] \leq \epsilon$.

Let us now comment on the size of the parameter $\kappa_f \stackrel{(32)}{=} \frac{\sigma_f}{2\theta^2}$. In our case, X is the polyhedral set (27) defined by $B = (-I_n, I_n)^T$, and $c = (\mathbf{0}^T, \mathbf{1}^T)^T$, where $I_n \in \mathbb{R}^{n \times n}$ is the identity matrix. Because of this structure (31) simplifies to

$$\theta := \sup_{u, v} \left\{ \left\| \begin{pmatrix} u \\ v \end{pmatrix} \right\| \left\| \begin{pmatrix} I_n u + \begin{pmatrix} A \\ q \end{pmatrix}^T v \end{pmatrix} \right\| = 1 \right. \\ \left. \text{and the corresponding rows of } I_n, A \text{ to } u, v \text{'s} \right. \\ \left. \text{non-zero elements are linearly independent.} \right\}. \quad (35)$$

To show that θ can be very large, let us assume that two rows of the matrix A are highly correlated (in this case rows corresponds to features). We denote these two rows by A_1 and A_2 , and let us assume that $A_1 = A_2 + \delta e_1$. Then we can chose $v = (-\frac{1}{\delta}, \frac{1}{\delta}, 0, \dots, 0)^T$ and $u = \mathbf{0}$. This particular choice is feasible in optimization problem (35) and hence is imposing a lower-bound on θ : $\theta \geq \frac{\sqrt{2}}{|\delta|}$. Clearly, for small δ , this shows that θ can be arbitrarily large.

6. Summary

In this paper we have extended the framework of the feasible descent method FDM to a randomized, and a randomized coordinate, FDM framework. We have shown that many problems in the machine learning literature fit our problem structure, and subsequently, any algorithm that fits our FDM framework can be used to successfully solve them. We have proven a linear convergence rate (under the weak strong convexity assumption) for both methods, and we have shown that the convergence rates are similar to the deterministic/non-randomized FDM. We also showed that for the cyclic coordinate descent method, the coefficients in FDM are worse than, or similar to, the stochastic coordinate descent method (and hence the theory tells us that they converge at roughly the same speed), but each iteration of the stochastic coordinate descent method is n -times cheaper. We concluded the paper with a result showing that, for the SDCA algorithm applied to the dual of the linear SVM, the duality gap converges linearly.

Acknowledgments

Chenxim Ma and Martin Takáč were supported by National Science Foundation grant CCF-1618717.

Appendix A. Proof of Theorem 7

Let us define an auxiliary vector \tilde{x} such that

$$\tilde{x}^{(i)} = \arg \min_{x^{(i)} \in X_i} f((x_k^{(1)}, x_k^{(2)}, \dots, x_k^{(i-1)}, x_k^{(i)}, x_k^{(i+1)}, \dots, x_k^{(n)})^T). \quad (36)$$

Then we can see that if coordinate i is chosen during iteration k in Algorithm 1 then

$$x_{k+1}^{(i)} = \begin{cases} x_k^{(j)}, & \text{if } j \neq i, \\ \tilde{x}^{(i)}, & \text{otherwise.} \end{cases} \quad (37)$$

If coordinate i is chosen during iteration k , then the optimality conditions for Step 6 of Algorithm 1, give us that

$$x_{k+1}^{(i)} = \mathbf{Proj}_{X_i}^W \left(x_{k+1}^{(i)} - \frac{1}{w_i} \nabla_i f(x_{k+1}) \right). \quad (38)$$

Moreover, by (37), for $j \neq i$ we have that $x_k^{(j)} = x_{k+1}^{(j)}$ which is possible only if $z_k^{(j)} = \nabla_j f(x_k)$.

Note that x_{k+1} is a random variable, which depends on i and x_k only. Therefore, we can define a random z_k such that the i -th coordinate is

$$z_k^{(i)} = \nabla_i f(x_k) - \nabla_i f((x_k^{(1)}, x_k^{(2)}, \dots, x_k^{(i-1)}, \tilde{x}^{(i)}, x_k^{(i+1)}, \dots, x_k^{(n)})^T) + w_i(x_k^{(i)} - \tilde{x}^{(i)}) \quad (39)$$

and the j -th coordinate (for $j \neq i$) is defined as $z_k^{(j)} = \nabla_j f(x_k)$. It is easy to verify that for z_k defined above, condition (14) holds. Now, we will compute $\mathbf{E}[\|z_k\|_W^2]$. We have that if the i -th coordinate is chosen then

$$\begin{aligned} & \frac{1}{w_i} (z_k^{(i)})^2 \\ &= \frac{1}{w_i} \left(\nabla_i f(x_k) - \nabla_i f((x_k^{(1)}, x_k^{(2)}, \dots, x_k^{(i-1)}, \tilde{x}^{(i)}, x_k^{(i+1)}, \dots, x_k^{(n)})^T) + w_i(x_k^{(i)} - \tilde{x}^{(i)}) \right)^2 \\ &\leq \frac{2}{w_i} \left(\nabla_i f(x_k) - \nabla_i f((x_k^{(1)}, x_k^{(2)}, \dots, x_k^{(i-1)}, \tilde{x}^{(i)}, x_k^{(i+1)}, \dots, x_k^{(n)})^T) \right)^2 + 2w_i(x_k^{(i)} - \tilde{x}^{(i)})^2 \\ &\leq 2 \left(\|\nabla_i f(x_k) - \nabla_i f((x_k^{(1)}, x_k^{(2)}, \dots, x_k^{(i-1)}, \tilde{x}^{(i)}, x_k^{(i+1)}, \dots, x_k^{(n)})^T)\|_W \right)^2 + 2w_i(x_k^{(i)} - \tilde{x}^{(i)})^2 \\ &\stackrel{(8)}{\leq} 2(L_f^W \|x_k - (x_k^{(1)}, x_k^{(2)}, \dots, x_k^{(i-1)}, \tilde{x}^{(i)}, x_k^{(i+1)}, \dots, x_k^{(n)})^T\|_W)^2 + 2w_i(x_k^{(i)} - \tilde{x}^{(i)})^2 \\ &= 2(L_f^W)^2 w_i (x_k^{(i)} - \tilde{x}^{(i)})^2 + 2w_i(x_k^{(i)} - \tilde{x}^{(i)})^2 = 2[(L_f^W)^2 + 1]w_i(x_k^{(i)} - \tilde{x}^{(i)})^2, \end{aligned} \quad (40)$$

otherwise

$$\frac{1}{w_i} (z_k^{(i)})^2 = \frac{1}{w_i} (\nabla_i f(x_k))^2.$$

Hence, we obtain that

$$\mathbf{E}[\|z_k\|_W^2] \stackrel{(40)}{\leq} \sum_{i=1}^n \frac{1}{n} 2[(L_f^W)^2 + 1]w_i(x_k^{(i)} - \tilde{x}^{(i)})^2 + \frac{n-1}{n} \sum_{i=1}^n \frac{1}{w_i} (\nabla_i f(x_k))^2. \quad (41)$$

From the optimality condition of Step 6 of Algorithm 1, and the fact that $X_i = \mathbb{R}$, we know that for all i the following holds,

$$\nabla_i f(x_k^{(1)}, \dots, x_k^{(i-1)}, \bar{x}^{(i)}, x_k^{(i+1)}, \dots, x_k^{(n)}) = 0. \quad (42)$$

Therefore $\forall i$ we have

$$\begin{aligned} \frac{1}{w_i} (\nabla_i f(x_k))^2 &= \frac{1}{w_i} (\nabla_i f(x_k) - \nabla_i f(x_k^{(1)}, \dots, x_k^{(i-1)}, \bar{x}^{(i)}, x_k^{(i+1)}, \dots, x_k^{(n)}))^2 \\ &\stackrel{(7)}{\leq} \frac{1}{w_i} L_f^2 (\bar{x}^{(i)} - x_k^{(i)})^2 = \frac{1}{w_i^2} L_f^2 w_i (\bar{x}^{(i)} - x_k^{(i)})^2. \end{aligned}$$

If we denote by $r^2 = \max_i \frac{L_f^2}{w_i^2}$, then we obtain from (41)

$$\begin{aligned} \mathbf{E}[|z_k|_{W^*}^2] &\stackrel{(40)}{\leq} \sum_{i=1}^n \left[\frac{1}{n} 2(L_f^W)^2 + 1 \right] + \frac{n-1}{n} r^2 \sum_{i=1}^n w_i (x_k^{(i)} - \bar{x}^{(i)})^2 \\ &= \left[\frac{1}{n} 2(L_f^W)^2 + 1 \right] + \frac{n-1}{n} r^2 \sum_{i=1}^n w_i (x_k^{(i)} - \bar{x}^{(i)})^2 \\ &= (2[(L_f^W)^2 + 1] + (n-1)r^2) \frac{1}{n} \sum_{i=1}^n w_i (x_k^{(i)} - \bar{x}^{(i)})^2 \\ &= (2[(L_f^W)^2 + 1] + (n-1)r^2) \mathbf{E}[\|x_k - x_{k+1}\|_W^2] \end{aligned}$$

and we can conclude that (15) holds with $\beta^2 = 2[(L_f^W)^2 + 1] + (n-1)r^2$.

Now, it remains to show (16). From (36) we know that

$$\nabla_i f(x_k^{(1)}, x_k^{(2)}, \dots, x_k^{(i-1)}, \bar{x}^{(i)}, x_k^{(i+1)}, \dots, x_k^{(n)})^T (\bar{x}^{(i)} - x_k^{(i)}) \leq 0. \quad (43)$$

Therefore, from (17) with $\xi = x_k^{(i)}$ and $x = (x_k^{(1)}, \dots, x_k^{(i-1)}, \bar{x}^{(i)}, x_k^{(i+1)}, \dots, x_k^{(n)})^T \stackrel{(37)}{=} x_{k+1}$, we have that

$$f(x_k) - f(x_{k+1}) \geq \gamma w_i |x_k^{(i)} - x_{k+1}^{(i)}|^2 + \nabla_i f(x_{k+1})(x_k^{(i)} - x_{k+1}^{(i)}) \geq \gamma w_i |x_k^{(i)} - x_{k+1}^{(i)}|^2. \quad (44)$$

Therefore

$$f(x_k) - f(x_{k+1}) \stackrel{(44)}{\geq} \gamma w_i |x_k^{(i)} - x_{k+1}^{(i)}|^2 = \gamma \|x_k - x_{k+1}\|_W^2.$$

and by taking expectation on both sides of the above, (16) follows with $\zeta = \gamma$.

Appendix B. Proof of Theorem 10

The proof is very similar to the proof of Theorem 7. Let us define an auxiliary vector \bar{x} in the same way as in (36). Then we can see that if coordinate i is chosen during iteration k in Algorithm 1 then (37) holds, and the optimality conditions for Step 6 of Algorithm 1 imply that (38) holds.

Note that x_{k+1} is a random variable which depends on i and x_k only. Therefore, we can define z_k such that i -th coordinate is given by (39). It is easy to verify that for z_k defined in (39), the condition (18) holds. Now, let us compute $(\|z_k\|_{W^*})^2$. We have that

$$(\|z_k\|_{W^*})^2 = \frac{1}{w_i} (z_k^{(i)})^2 \stackrel{(40)}{\leq} 2(L_f^W)^2 + 1 + w_i (x_k^{(i)} - \bar{x}^{(i)})^2 \stackrel{(37)}{=} 2[(L_f^W)^2 + 1] \|x_k^{(i)} - x_{k+1}^{(i)}\|_W^2.$$

Therefore, we conclude that (19) holds with $\beta^2 = 2[(L_f^W)^2 + 1]$.

Now, it remains to show (20). Again from (36) we know that (43) holds. Therefore from (17) with $\xi = x_k^{(i)}$ and $x = (x_k^{(1)}, \dots, x_k^{(i-1)}, \bar{x}^{(i)}, x_k^{(i+1)}, \dots, x_k^{(n)})^T \stackrel{(37)}{=} x_{k+1}$ we have (44). Therefore $f(x_k) - f(x_{k+1}) \geq \gamma w_i |x_k^{(i)} - x_{k+1}^{(i)}|^2 = \gamma \|x_k - x_{k+1}\|_W^2$, so (20) holds with $\zeta = \gamma$.

Appendix C. Proof of Theorem 12

This proof is based on the proof of Theorem 3.2 in Neocora (2015). We can write the optimality conditions for x_{k+1} from (14) and using the definition of a projection given in (9). We have that $\forall x \in X$, the following inequality holds

$$\langle W(x_{k+1} - x_k + \omega_k W^{-1}(\nabla f(x_k) - z_k)), x - x_{k+1} \rangle \geq 0. \quad (45)$$

Now, using the convexity of f we obtain that

$$\begin{aligned} f(x_{k+1}) - f^* &= f(x_{k+1}) - f(\bar{x}_{k+1}) \leq \langle \nabla f(x_{k+1}), x_{k+1} - \bar{x}_{k+1} \rangle \\ &= \langle \nabla f(x_{k+1}) - \nabla f(x_k) + \nabla f(x_k), x_{k+1} - \bar{x}_{k+1} \rangle. \end{aligned} \quad (46)$$

Plugging $x = \bar{x}_{k+1}$ into (45) we obtain

$$\left\langle \frac{1}{\omega_k} W(x_{k+1} - x_k) - z_k, \bar{x}_{k+1} - x_{k+1} \right\rangle \geq \langle \nabla f(x_k), x_{k+1} - \bar{x}_{k+1} \rangle. \quad (47)$$

Plugging this into (46) gives us that

$$\begin{aligned} f(x_{k+1}) - f(\bar{x}_{k+1}) &\stackrel{(46), (47)}{\leq} \left\langle \nabla f(x_{k+1}) - \nabla f(x_k) - \frac{1}{\omega_k} W(x_{k+1} - x_k) + z_k, x_{k+1} - \bar{x}_{k+1} \right\rangle \\ &\stackrel{CS}{\leq} \|\nabla f(x_{k+1}) - \nabla f(x_k)\|_W^* \|x_{k+1} - \bar{x}_{k+1}\|_W \\ &\quad + \left\langle -\frac{1}{\omega_k} W(x_{k+1} - x_k) + z_k, x_{k+1} - \bar{x}_{k+1} \right\rangle \\ &\stackrel{(8)}{\leq} L_f^W \|x_{k+1} - x_k\|_W \|x_{k+1} - \bar{x}_{k+1}\|_W \\ &\quad + \left\langle -\frac{1}{\omega_k} W(x_{k+1} - x_k), x_{k+1} - \bar{x}_{k+1} \right\rangle + \langle z_k, x_{k+1} - \bar{x}_{k+1} \rangle \\ &\stackrel{CS}{\leq} L_f^W \|x_{k+1} - x_k\|_W \|x_{k+1} - \bar{x}_{k+1}\|_W \\ &= \left(L_f^W + \frac{1}{\omega_k} \right) \|x_{k+1} - x_k\|_W + \|z_k\|_W^* \|x_{k+1} - \bar{x}_{k+1}\|_W \\ &\stackrel{(5)}{\leq} \left(L_f^W + \frac{1}{\omega_k} \right) \|x_{k+1} - x_k\|_W + \|z_k\|_W^* \sqrt{\frac{1}{\kappa_f} (f(x_{k+1}) - f(\bar{x}_{k+1}))}. \end{aligned} \quad (48)$$

Therefore, we can conclude that

$$f(x_{k+1}) - f^* \stackrel{(48)}{\leq} \frac{1}{\kappa_f} \left((L_f^W + \frac{1}{\omega}) \|x_{k+1} - x_k\|_W + \|z_k\|_W^* \right)^2. \quad (49)$$

Taking the expectation of (49) with respect to the random vector z_k , we obtain

$$\begin{aligned} \mathbf{E}[f(x_{k+1}) - f(\bar{x}_{k+1})] &\stackrel{(49)}{\leq} \frac{1}{\kappa_f} \mathbf{E} \left[\left((L_f^W + \frac{1}{\omega}) \|x_{k+1} - x_k\|_W + \|z_k\|_W^* \right)^2 \right] \\ &\leq \frac{2}{\kappa_f} \left((L_f^W + \frac{1}{\omega})^2 \mathbf{E}[\|x_{k+1} - x_k\|_W^2] + \mathbf{E}[\|z_k\|_W^*]^2 \right) \\ &\stackrel{(15)}{\leq} \frac{2}{\kappa_f} \left((L_f^W + \frac{1}{\omega})^2 + \beta^2 \right) \mathbf{E}[\|x_k - x_{k+1}\|_W^2] \\ &\stackrel{(16)}{\leq} \frac{2}{\kappa_f} \left((L_f^W + \frac{1}{\omega})^2 + \beta^2 \right) \frac{1}{\zeta} (f(x_k) - \mathbf{E}[f(x_{k+1})]) \\ &= \underbrace{\frac{2}{\kappa_f} \left((L_f^W + \frac{1}{\omega})^2 + \beta^2 \right) \frac{1}{\zeta}}_c (f(x_k) - f(\bar{x}_k)) \\ &\quad + \mathbf{E}[f(\bar{x}_{k+1})] - \mathbf{E}[f(x_{k+1})]. \end{aligned} \quad (50)$$

Finally, from (50) we obtain that

$$\mathbf{E}[f(x_{k+1}) - f^*] = \mathbf{E}[f(x_{k+1}) - f(\bar{x}_{k+1})] \leq \frac{c}{1+c} (f(x_k) - f(\bar{x}_{k+1})) = \frac{c}{1+c} (f(x_k) - f^*),$$

and the result follows.

Appendix D. Proof of Theorem 13 if $z_k = 0$

Let us define an auxiliary vector \tilde{x} such that

$$\tilde{x}^{(j)} = \mathbf{Proj}_X^W(x_k - \omega_k W^{-1}(\nabla f(x_k) - z_k)_{[j]}). \quad (51)$$

Then we can see that if coordinates \mathcal{I} is chosen during iteration k in Algorithm 1 then

$$x_{k+1}^{(j)} = \begin{cases} x_k^{(j)}, & \text{if } j \notin \mathcal{I}, \\ \tilde{x}^{(j)}, & \text{otherwise.} \end{cases} \quad (52)$$

Therefore, let us estimate the expected value of f at a random point x_{k+1} , where the expectation is taken with respect to the selection of coordinates \mathcal{I} at iteration k . Let

$h \in \mathbb{R}^n$. Then if $\frac{1}{\omega_k} \geq \max_i \frac{L_i}{\omega_i}$ we have

$$\begin{aligned} \mathbf{E}[f(x_k + h_{[T]})] &\stackrel{(7)}{\leq} f(x_k) + \mathbf{E} \left[\langle \nabla f(x_k), h_{[T]} \rangle + \frac{L_T}{2\omega_T} \|h_{[T]}\|_W^2 \right] \\ &\leq f(x_k) + \mathbf{E} \left[\langle \nabla f(x_k), h_{[T]} \rangle + \frac{1}{2\omega_k} \|h_{[T]}\|_W^2 \right] \\ &\stackrel{(52)}{=} f(x_k) + \frac{T}{n} \left(\langle \nabla f(x_k), h \rangle + \frac{1}{2\omega_k} \|h\|_W^2 \right) \\ &= \frac{n-T}{n} f(x_k) + \frac{T}{n} \underbrace{\left(f(x_k) + \langle \nabla f(x_k) - z_k, h \rangle + \frac{1}{2\omega_k} \|h\|_W^2 + \langle z_k, h \rangle \right)}_{\mathcal{H}(h; x_k, z_k)}. \end{aligned} \quad (53)$$

Now, observe that

$$\begin{aligned} \tilde{x} &= x_k + \arg \min_{h: x+x_k \in X} \mathcal{H}(h; x_k, z_k) \\ &= x_k + \arg \min_{h \in \mathbb{R}^n} \{ \mathcal{H}(h; x_k, z_k) + \Phi_X(x + x_k) \} =: x_k + \hat{h}, \end{aligned} \quad (54)$$

where $\Phi_X(x)$ is the indicator function for the set X , i.e.

$$\Phi_X(x) = \begin{cases} 0, & \text{if } x \in X, \\ +\infty, & \text{otherwise.} \end{cases} \quad (55)$$

From the first order optimality conditions of (54) we have

$$\nabla f(x_k) - z_k + \frac{1}{\omega_k} W\hat{h} + s = 0, \quad (56)$$

where $s \in \partial\Phi(x_k + \hat{h})$. We can define a composite gradient mapping Lu and Xiao (2013); Nesterov (2013); Tappenden et al. (2015) as

$$g := -\frac{1}{\omega_k} W\hat{h}. \quad (57)$$

Therefore, we can observe that

$$-\nabla f(x_k) + z_k + g \stackrel{(56)}{\in} \partial\Phi(x_k + \hat{h}). \quad (58)$$

It is also easy to show that

$$\|\hat{h}\|_W^2 = \|\omega_k W^{-1}g\|_W^2 = \omega_k^2 (\|g\|_W^*)^2 \quad (59)$$

and

$$\langle g, \hat{h} \rangle = -\frac{1}{\omega_k} \|\hat{h}\|_W^2 \stackrel{(59)}{=} -\omega_k (\|g\|_W^*)^2. \quad (60)$$

Finally note that for any $y \in X$ we have

$$\begin{aligned} \|x_k + \hat{h} - y\|_W^2 &= \|x_k - y\|_W^2 + 2\omega_k \langle g, y - x_k \rangle + \|\hat{h}\|_W^2 \\ &\stackrel{(59)}{=} \|x_k - y\|_W^2 + 2\omega_k \langle g, y - x_k \rangle + \omega_k^2 (\|g\|_W^*)^2. \end{aligned} \quad (61)$$

Now, we are ready to bound $\mathcal{H}(h; x_k, z_k) + \Phi(x + h)$ for $h = \hat{h}$. We have

$$\begin{aligned} &\mathcal{H}(\hat{h}; x_k, z_k) + \Phi(x_k + \hat{h}) \\ &= f(x_k) + \langle \nabla f(x_k) - z_k, \hat{h} \rangle + \frac{1}{2\omega_k} \|\hat{h}\|_W^2 + \Phi(x_k + \hat{h}) \\ &\stackrel{(58)}{\leq} f(y) + \langle \nabla f(x_k), x_k - y \rangle + \langle \nabla f(x_k) - z_k, \hat{h} \rangle + \frac{1}{2\omega_k} \|\hat{h}\|_W^2 \\ &\quad + \Phi(y) + \langle -\nabla f(x_k) + z_k + g, x_k + \hat{h} - y \rangle \\ &= f(y) + \Phi(y) + \frac{1}{2\omega_k} \|\hat{h}\|_W^2 + \langle g, x_k - y \rangle + \langle z_k, x_k - y \rangle + \langle g, \hat{h} \rangle \\ &\stackrel{(60), (59)}{=} f(y) + \Phi(y) + \frac{1}{2} \omega_k (\|g\|_W^*)^2 + \langle g, x_k - y \rangle + \langle z_k, x_k - y \rangle - \omega_k (\|g\|_W^*)^2 \\ &= f(y) + \Phi(y) - \frac{1}{2} \omega_k (\|g\|_W^*)^2 + \langle g, x_k - y \rangle + \langle z_k, x_k - y \rangle \\ &\stackrel{(61)}{=} f(y) + \Phi(y) - \frac{1}{2\omega_k} \left(\|x_k + \hat{h} - y\|_W^2 - \|x_k - y\|_W^2 \right) + \langle z_k, x_k - y \rangle \\ &\stackrel{(32), (63)}{=} f(y) + \Phi(y) - \frac{1}{2\omega_k} \frac{n}{\tau} \left(\mathbf{E} \|x_{k+1} - y\|_W^2 - \|x_k - y\|_W^2 \right) + \langle z_k, x_k - y \rangle, \end{aligned}$$

where in the last step, we use

$$n \mathbf{E} \|x_{k+1} - y\|_W^2 = \tau \|x_k + \hat{h} - y\|_W^2 + (n - \tau) \|x_k - y\|_W^2.$$

Now, from (53) we conclude that $\forall y$ we have

$$\begin{aligned} \mathbf{E}[f(x_{k+1})] &\leq \frac{n - \tau}{n} f(x_k) + \frac{\tau}{n} \left(f(y) + \Phi(y) - \frac{n}{2\omega_k \tau} \mathbf{E} \|x_{k+1} - y\|_W^2 \right) \\ &\quad + \frac{n}{2\omega_k \tau} \|x_k - y\|_W^2 + \langle z_k, x_k + \hat{h} - y \rangle, \end{aligned}$$

which can be equivalently written as

$$\begin{aligned} \mathbf{E} \left[f(x_{k+1}) + \frac{1}{2\omega_k} \|x_{k+1} - y\|_W^2 \right] &\leq f(x_k) + \frac{1}{2\omega_k} \|x_k - y\|_W^2 \\ &\quad - \frac{\tau}{n} (f(x_k) - f(y) - \Phi(y)) + \frac{\tau}{n} \langle z_k, x_k + \hat{h} - y \rangle. \end{aligned}$$

If we choose $y = \bar{x}_k$ then the latter inequality reads as follows

$$\begin{aligned} \mathbf{E} \left[f(x_{k+1}) + \frac{1}{2\omega_k} \|x_{k+1} - \bar{x}_k\|_W^2 \right] &\leq f(x_k) + \frac{1}{2\omega_k} \|x_k - \bar{x}_k\|_W^2 \\ &\quad - \frac{\tau}{n} (f(x_k) - f^*) + \frac{\tau}{n} \langle z_k, x_k + \hat{h} - \bar{x}_k \rangle. \end{aligned}$$

From the definition of \bar{x} we obtain that $\|x_{k+1} - \bar{x}_{k+1}\|_W \leq \|x_{k+1} - \bar{x}_k\|_W$ and therefore

$$\begin{aligned} \mathbf{E} \left[f(x_{k+1}) - f^* + \frac{1}{2\omega_k} \|x_{k+1} - \bar{x}_{k+1}\|_W^2 \right] &\leq (1 - \frac{\tau}{n}) (f(x_k) - f^*) \\ &\quad + \frac{1}{2\omega_k} \|x_k - \bar{x}_k\|_W^2 + \frac{\tau}{n} \langle z_k, x_k + \hat{h} - \bar{x}_k \rangle. \end{aligned}$$

Let us assume that $\forall k : z_k = 0$. Then let us define $c = \frac{2\tau\omega^2}{n(2\delta+1)} \in (0, 1)$,

$$\mathbf{E} \left[f(x_{k+1}) - f^* + \frac{1}{2\omega_k} \|x_{k+1} - \bar{x}_{k+1}\|_W^2 \right] \leq (1 - c) \left(f(x_k) - f^* + \frac{1}{2\omega_k} \|x_k - \bar{x}_k\|_W^2 \right). \quad (62)$$

Therefore,

$$\begin{aligned} \mathbf{E}[f(x_k) - f^*] &\leq \mathbf{E} \left[f(x_k) - f^* + \frac{1}{2\omega_k} \|x_k - \bar{x}_k\|_W^2 \right] \\ &\stackrel{(62)}{\leq} (1 - c)^k \left(f(x_0) - f^* + \frac{1}{2\omega_0} \|x_0 - \bar{x}_0\|_W^2 \right). \end{aligned}$$

Appendix E. Proof of Theorem 13 if $z_k \neq 0$

The proof follows similar arguments to the proof of Theorem 13 when $z_k = 0$. Let us define an auxiliary vector \tilde{x} in the same way as in (51). Then we can see that if coordinates \mathcal{I} is chosen during iteration k in Algorithm 1 then (52) holds. Therefore, let us estimate the expected value of f at a random point x_{k+1} , where the expectation is taken with respect to the selection of coordinate i at iteration k . Let $h \in \mathbb{R}^n$. Then if $\frac{1}{\omega_k} \geq \max_{i \in \mathcal{I}_k} \frac{L_i}{\omega_i}$ we have that (53) holds. Now, observe that

$$\begin{aligned} \tilde{x} &= x_k + \arg \min_{h: x_k + h \in X} \mathcal{H}(h; x_k, z_k) \\ &= x_k + \arg \min_{h \in \mathbb{R}^n} \{ \mathcal{H}(h; x_k, z_k) + \Phi_X(h + x_k) \} =: x_k + \hat{h}, \end{aligned} \quad (63)$$

where $\Phi_X(x)$ is indicator function for set X , (55). Now, we have

$$\begin{aligned} \mathcal{H}(\hat{h}; x_k, z_k) &= \min_{h \in \mathbb{R}^n} \left\{ f(x_k) + \langle \nabla f(x_k) - z_k, h \rangle + \frac{1}{2\omega_k} \|h\|_W^2 + \Phi_X(h + x_k) \right\} \\ &= \min_{y \in \mathbb{R}^n} \left\{ f(x_k) + \langle \nabla f(x_k) - z_k, y - x_k \rangle + \frac{1}{2\omega_k} \|y - x_k\|_W^2 + \Phi_X(y) \right\} \\ &\leq \min_{\lambda \in [0, 1]} \left\{ f(\lambda \bar{x}_k + (1 - \lambda)x_k) + \langle -z_k, \lambda(\bar{x}_k - x_k) \rangle \right. \\ &\quad \left. + \frac{1}{2\omega_k} \|\lambda(\bar{x}_k - x_k)\|_W^2 + \Phi_X(\lambda(\bar{x}_k - x_k) + x_k) \right\} \\ &\leq \min_{\lambda \in [0, 1]} \left\{ \lambda f(x_k) + (1 - \lambda)f(x_k) + \lambda \|z_k\|_W^* \|\bar{x}_k - x_k\|_W + \frac{\lambda^2}{2\omega_k} \|\bar{x}_k - x_k\|_W^2 \right\}. \end{aligned}$$

Note that from (52) and (63) we have

$$\|\hat{h}\|_W^2 = \sum_{i=1}^n \|\hat{h}_i\|_i^2 = \frac{n}{\tau} \mathbf{E} \|x_{k+1} - x_k\|_W^2 \stackrel{(20)}{\leq} \frac{n}{\zeta \tau} \mathbf{E}[f(x_k) - f(x_{k+1})]. \quad (64)$$

Therefore, we conclude that

$$\begin{aligned} \mathbf{E}[f(x_{k+1}) - f^*] &\stackrel{(53), (19)}{\leq} \min_{\lambda \in [0,1]} \left\{ f(x_k) - f^* + \frac{\tau}{n} \left(\lambda (f(\bar{x}_k) - f(x_k)) + \lambda \|z_k\|_{W^*}^* \|\bar{x}_k - x_k\|_W \right) \right. \\ &\quad \left. + \frac{\lambda^2}{2\omega_k} \|\bar{x}_k - x_k\|_W^2 + \|z_k\|_W^2 \|\hat{h}\|_W \right\} \\ &\stackrel{(5)}{\leq} \min_{\lambda \in [0,1]} \left\{ f(x_k) - f^* + \frac{\tau}{n} \left(-\lambda (f(x_k) - f^*) + \lambda \|z_k\|_{W^*}^* \|\bar{x}_k - x_k\|_W \right) \right. \\ &\quad \left. + \frac{\lambda^2}{2\omega_k \kappa_f} (f(x_k) - f^*) + \|z_k\|_W \|\hat{h}\|_W \right\}. \end{aligned} \quad (65)$$

Now, let us denote by $\xi_k = f(x_k) - f^*$. Notice that

$$\left(\|z_k\|_{W^*}^* \right)^2 = \sum_{i=1}^n \left(\|z_k\|_{W^*}^* \right)^2 \stackrel{(19), (20)}{\leq} \frac{\beta^2}{n \zeta} (\xi_k - \mathbf{E}[\xi_{k+1}]) \quad (65)$$

where the expectation is with respect to the random choice i during the k -th iteration. Therefore we have

$$\begin{aligned} \mathbf{E}[\xi_{k+1}] &\leq \min_{\lambda \in [0,1]} \left\{ \xi_k + \frac{\tau}{n} \left(-\lambda \xi_k + \lambda \|z_k\|_{W^*}^* \|\bar{x}_k - x_k\|_W + \frac{\lambda^2}{2\omega_k \kappa_f} \xi_k + \|z_k\|_{W^*}^* \|\hat{h}\|_W \right) \right\} \\ &\stackrel{(65), (64)}{\leq} \min_{\lambda \in [0,1]} \left\{ \xi_k + \frac{\tau}{n} \left(-\lambda \xi_k + \lambda \|z_k\|_{W^*}^* \|\bar{x}_k - x_k\|_W \right) \right. \\ &\quad \left. + \frac{\lambda^2}{2\omega_k \kappa_f} \xi_k + \frac{n\beta}{\zeta \sqrt{\tau}} (\xi_k - \mathbf{E}[\xi_{k+1}]) \right\} \end{aligned}$$

which is equivalent to

$$\begin{aligned} \left(1 + \frac{\sqrt{\tau}\beta}{\zeta}\right) \mathbf{E}[\xi_{k+1}] &\leq \left(1 + \frac{\sqrt{\tau}\beta}{\zeta}\right) \xi_k + \min_{\lambda \in [0,1]} \left\{ -\frac{\tau}{n} \lambda \xi_k + \frac{\tau}{n} \lambda \|z_k\|_{W^*}^* \|\bar{x}_k - x_k\|_W + \frac{\tau}{n} \frac{\lambda^2}{2\omega_k \kappa_f} \xi_k \right\} \\ &\stackrel{(65), (5)}{\leq} \left(1 + \frac{\sqrt{\tau}\beta}{\zeta}\right) \xi_k + \min_{\lambda \in [0,1]} \left\{ -\frac{\tau}{n} \lambda \xi_k + \frac{\tau}{n} \lambda \sqrt{n \frac{\beta^2}{\zeta} (\xi_k - \mathbf{E}[\xi_{k+1}])} \sqrt{\frac{1}{\kappa_f^2} \xi_k + \frac{\tau}{n} \frac{\lambda^2}{2\omega_k \kappa_f} \xi_k} \right\}. \end{aligned}$$

Using the fact that $\forall a, b \in \mathbb{R}_+$, we have $\sqrt{ab} \leq \frac{1}{2}a + \frac{1}{2}b$ we obtain that

$$\begin{aligned} &\left(1 + \frac{\sqrt{\tau}\beta}{\zeta}\right) \mathbf{E}[\xi_{k+1}] \\ &\leq \left(1 + \frac{\sqrt{\tau}\beta}{\zeta}\right) \xi_k + \min_{\lambda \in [0,1]} \left\{ -\frac{\tau}{n} \lambda \xi_k + \sqrt{\frac{\beta^2}{\zeta} (\xi_k - \mathbf{E}[\xi_{k+1}])} \sqrt{\frac{\lambda^2}{n \kappa_f^2} \xi_k + \frac{\tau}{n} \frac{\lambda^2}{2\omega_k \kappa_f} \xi_k} \right\} \\ &\leq \left(1 + \frac{\sqrt{\tau}\beta}{\zeta}\right) \xi_k + \min_{\lambda \in [0,1]} \left\{ -\frac{\tau}{n} \lambda \xi_k + \frac{\beta^2}{2\zeta} (\xi_k - \mathbf{E}[\xi_{k+1}]) + \frac{1}{2} \frac{\lambda^2}{n \kappa_f} \xi_k + \frac{\tau}{n} \frac{\lambda^2}{2\omega_k \kappa_f} \xi_k \right\}. \end{aligned}$$

Therefore, we obtain

$$\left(1 + \frac{\sqrt{\tau}\beta}{\zeta} + \frac{\beta^2}{2\zeta}\right) \mathbf{E}[\xi_{k+1}] \leq \left(1 + \frac{\sqrt{\tau}\beta}{\zeta} + \frac{\beta^2}{2\zeta}\right) \xi_k + \frac{\tau}{n\omega_k \kappa_f} \min_{\lambda \in [0,1]} \left\{ -\lambda \omega_k \kappa_f + \frac{\tau}{2} (1 + \omega\tau) \right\} \xi_k. \quad (66)$$

The optimal λ^* that minimizes the above expression is

$$\lambda^* = \min \left\{ 1, \frac{\omega_k \kappa_f}{\omega\tau + 1} \right\}.$$

Consider now two cases:

- $\lambda^* < 1$. In this case

$$-\lambda^* \omega_k \kappa_f + \frac{(\lambda^*)^2}{2} (1 + \omega\tau) = -\frac{1}{2} \frac{(\omega_k \kappa_f)^2}{\omega\tau + 1}.$$

Combining this with (66) gives

$$\left(1 + \frac{\sqrt{\tau}\beta}{\zeta} + \frac{\beta^2}{2\zeta}\right) \mathbf{E}[\xi_{k+1}] \leq \left(1 + \frac{\sqrt{\tau}\beta}{\zeta} + \frac{\beta^2}{2\zeta} - \frac{1}{2n} \frac{\omega_k \kappa_f}{\omega\tau + 1}\right) \xi_k,$$

which is equivalent to

$$\mathbf{E}[\xi_{k+1}] \leq \left(1 - \frac{1}{2n} \frac{2\omega_k \kappa_f \zeta}{(\omega\tau + 1)(2\zeta + 2\beta\sqrt{\tau} + \beta)}\right) \xi_k.$$

- $\lambda^* = 1$. In this case $\frac{\omega_k \kappa_f}{\omega\tau + 1} \geq 1$ and hence

$$-\lambda^* \omega_k \kappa_f + \frac{(\lambda^*)^2}{2} (1 + \omega\tau) = -\omega_k \kappa_f + \frac{1}{2} (1 + \omega\tau) \leq -\omega_k \kappa_f + \frac{1}{2} \omega_k \kappa_f = -\frac{1}{2} \omega_k \kappa_f.$$

Therefore, from (66) we can conclude that

$$\mathbf{E}[\xi_{k+1}] \leq \left(1 - \frac{\zeta\tau}{n(2\zeta + 2\beta\sqrt{\tau} + 1 + \beta^2)}\right) \xi_k.$$

References

- Silvia Bonettini. Inexact block coordinate descent methods with application to non-negative matrix factorization. *IMA Journal of Numerical Analysis*, 31(4):1431–1452, 2011.
- Kai-Wei Chang, Cho-Jui Hsieh, and Chih-Jen Lin. Coordinate descent method for large-scale l2-loss linear support vector machines. *The Journal of Machine Learning Research*, 9:1369–1398, 2008.
- Olivier Devolder, François Glineur, and Yurii Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1-2):37–75, 2014.
- Alan J. Hoffman. On approximate solutions of systems of linear inequalities. *Journal of Research of the National Bureau of Standards*, 49(4):263–265, 1952.
- Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S Sathiya Keerthi, and Sellamannickam Sundararajan. A dual coordinate descent method for large-scale linear svm. In *International Conference on Machine Learning (ICML)*, 2008.

- Xiaogin Hua and Nobuo Yamashita. An inexact coordinate descent method for the weighted l_1 -regularized convex optimization problem. Technical report, Technical report, School of Mathematics and Physics, Kyoto University, Kyoto 606-8501, Japan, 2012.
- Wu Li. The sharp Lipschitz constants for feasible and optimal solutions of a perturbed linear program. *Linear algebra and its applications*, 187:15–40, 1993.
- Ji Lin and Stephen J. Wright. Asynchronous stochastic coordinate descent: Parallelism and convergence properties. *SIAM Journal on Optimization*, 25(1):35117376, 2015.
- Zhaosong Lu and Lin Xiao. On the complexity analysis of randomized block-coordinate descent methods. *Mathematical Programming*, pages 1–28, 2013.
- Zhi-Quan Luo and Paul Tseng. Error bounds and convergence analysis of feasible descent methods: a general approach. *Annals of Operations Research*, 46(1):157–178, 1993.
- Ion Necoara. Linear convergence of first order methods under weak nondegeneracy assumptions for convex programming. *arXiv:1504.06298*, 2015.
- Ion Necoara and Dragos Clipci. Parallel coordinate descent methods for composite minimization. *SIAM Journal on Optimization*, 26(1):19717226, 2016.
- Ion Necoara and Valentin Nedelcu. Distributed dual gradient methods and error bound conditions. *arXiv:1401.4398*, 2014a.
- Ion Necoara and Valentin Nedelcu. Rate analysis of inexact dual first-order methods application to dual decomposition. *Automatic Control, IEEE Transactions on*, 59(5):1232–1243, 2014b.
- Yurii Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.
- Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014.
- Stephen M. Robinson. Bounds for error in the solution set of a perturbed linear program. *Linear Algebra and its applications*, 6:69–81, 1973.
- Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss. *The Journal of Machine Learning Research*, 14(1):567–599, 2013.
- Ruoyu Sun and Yinyu Ye. Worst-case complexity of cyclic coordinate descent: $o(n^2)$ gap with randomized version. *arXiv:1604.07130*, 2016.
- Martin Takáč, Peter Richtárik, and Nathan Srebro. Distributed mini-batch SDCA. *arXiv:1507.08322*, 2015.
- Martin Takáč, Avleen Singh Bijral, Peter Richtárik, and Nathan Srebro. Mini-batch primal and dual methods for SVMs. *International Conference on Machine Learning (ICML)*, 2013.
- Rachael Tappenden, Martin Takáč, and Peter Richtárik. On the complexity of parallel coordinate descent. *arXiv:1503.03033*, 2015.
- Rachael Tappenden, Peter Richtárik, and Jacek Gondzio. Inexact coordinate descent: complexity and preconditioning. *Journal of Optimization Theory and Applications*, 170(1):14417176, 2016.
- Po-Wei Wang and Chih-Jen Lin. Iteration complexity of feasible descent methods for convex optimization. *The Journal of Machine Learning Research*, 15(1):1523–1548, 2014.
- H. Zhang and W. Yin. Gradient methods for convex minimization: better rates under weaker conditions. Technical report, CAM Report 13-17, UCLA, 2013.

A Practical Scheme and Fast Algorithm to Tune the Lasso With Optimality Guarantees

Michaël Chichignoud

*Seminar for Statistics
ETH Zürich*

MICHAEL.CHICHIGNOUD@GMAIL.COM

Johannes Lederer*

*Department of Statistics
and Department of Biostatistics
University of Washington*

LEDERERJ@UW.EDU

Martin J. Wainwright

*Department of Statistics
and Department of Electrical Engineering and Computer Sciences
University of California at Berkeley*

WAINWRIGHT@BERKELEY.EDU

Editor: Francis Bach

Abstract

We introduce a novel scheme for choosing the regularization parameter in high-dimensional linear regression with Lasso. This scheme, inspired by Lepski's method for bandwidth selection in non-parametric regression, is equipped with both optimal finite-sample guarantees and a fast algorithm. In particular, for any design matrix such that the Lasso has low sup-norm error under an "oracle choice" of the regularization parameter, we show that our method matches the oracle performance up to a small constant factor, and show that it can be implemented by performing simple tests along a single Lasso path. By applying the Lasso to simulated and real data, we find that our novel scheme can be faster and more accurate than standard schemes such as Cross-Validation.

Keywords: Lasso, regularization parameter, tuning parameter, high-dimensional regression, oracle inequalities

1. Introduction

Regularized estimators—among them the Lasso (Tibshirani, 1996), the Square-Root and the Scaled Lasso (Antoniadis, 2010; Belloni et al., 2011; Städler et al., 2010; Sun and Zhang, 2012), as well as estimators based on nonconvex penalties such as MCP (Zhang, 2010) and SCAD (Fan and Li, 2001)—all hinge on finding a "suitable" choice of tuning parameters. There are many possible methods for solving this so-called calibration problem, but for high-dimensional regression problems, there is not a single method that is computationally tractable and for which the non-asymptotic theory is well understood.

The focus of this paper is the calibration of the Lasso for sparse linear regression, where the tuning parameter needs to be adjusted to both the noise distribution and the

design matrix (van de Geer and Lederer, 2013; Hebiri and Lederer, 2013; Dalalyan et al., 2014). Calibration schemes for this setting are typically based on Cross-Validation (CV) or BIC-type criteria. However, CV-based procedures can be computationally intensive and are currently lacking in non-asymptotic theory for high-dimensional problems. BIC-type criteria, on the other hand, are computationally simpler but also lacking in non-asymptotic guarantees. Another approach is to replace the Lasso with Square-Root Lasso or TREX (Lederer and Müller, 2015); however, Square-Root Lasso still contains a tuning parameter that needs to be calibrated to certain aspects of the model, and the theory for TREX is currently fragmentary. For these reasons and given the extensive use of the Lasso in practice, understanding the calibration of Lasso is important.

In this paper, we introduce a new scheme for calibrating the Lasso in the supremum norm (ℓ_∞)-loss, which we refer to as *Adaptive Validation for ℓ_∞* (AV_∞). This method is based on tests that are inspired by Lepski's method for non-parametric regression (Lepski, 1990; Lepski et al., 1997); see also Chichignoud and Lederer (2014). In contrast to current schemes for the Lasso, our method is equipped with both optimal theoretical guarantees and a fast computational routine.

The remainder of this paper is organized as follows. In Section 2, we introduce the AV_∞ method. Our main theoretical results show that this method enjoys finite sample guarantees for the calibration of Lasso with respect to sup-norm loss (Theorem 3) and variable selection (Remark 4). In addition, we provide a simple and fast algorithm (Algorithm 1). In Section 3, we illustrate these features with applications to simulated data and to biological data. We conclude with a discussion in Section 4.

Notation: The indicator of events is denoted by $\mathbb{1}\{\cdot\} \in \{0, 1\}$, the cardinality of sets by $|\cdot|$, the sup-norm or maximum norm of vectors in \mathbb{R}^p vectors by $\|\cdot\|_\infty$, the number of non-zero entries by $\|\cdot\|_0$, the ℓ_1 - and ℓ_2 -norms by $\|\cdot\|_1$ and $\|\cdot\|_2$, respectively, and $[p] := \{1, \dots, p\}$. For given vector $\beta \in \mathbb{R}^p$ and subset A of $[p]$, $\beta_A \in \mathbb{R}^{|A|}$ and $\beta_{A^c} \in \mathbb{R}^{|A^c|}$ denote the components in A and in its complement A^c , respectively.

2. Background and Methodology

In this section, we introduce some background and then move onto a description of the AV_∞ method.

2.1 Framework

We study the calibration of the Lasso tuning parameter in high-dimensional linear regression models that can contain many predictors and allow for the possibility of correlated and heavy-tailed noise. More specifically, we assume that the data (Y, X) with outcome $Y \in \mathbb{R}^n$ and design matrix $X \in \mathbb{R}^{n \times p}$ is distributed according to a linear regression model

$$Y = X\beta^* + \varepsilon, \tag{Model}$$

where $\beta^* \in \mathbb{R}^p$ is the regression vector and $\varepsilon \in \mathbb{R}^n$ is a random noise vector. Our framework allows for p larger than n and requires that the noise variables ε satisfy only the second moment condition

$$\max_{i \in \{1, \dots, n\}} \mathbb{E}[\varepsilon_i^2] < \infty. \tag{1}$$

* Corresponding author. Postal address: Department of Statistics at University of Washington, Box 354322, Seattle, WA 98195

A standard approach for estimating β^* in such a model is by computing the l_1 -regularized least-squares estimate, known as the Lasso, and given by

$$\hat{\beta}_\lambda \in \underset{\beta \in \mathbb{R}^p}{\operatorname{arg\,min}} \left\{ \frac{\|Y - X\beta\|_2^2}{n} + \lambda \|\beta\|_1 \right\}. \quad (\text{Lasso})$$

Note that this equation actually defines a family of estimators indexed by the tuning parameter $\lambda > 0$, which determines the level of regularization.

Intuitively, the optimal choice of λ is dictated by a trade-off between bias and some form of variance control. Bias is induced by the shrinkage effect of the l_1 -regularizer, which acts even on non-zero coordinates of the regression vector. Thus, the bias grows as λ is increased. On the other hand, l_1 -regularization is useful in canceling out fluctuations in the score function, which for the linear regression model is given by $X^\top \varepsilon/n$. Thus, an optimal choice of λ is the smallest one that is large enough to control these fluctuations.

A large body of theoretical work (e.g., van de Geer and Bühlmann (2009); Büchel et al. (2009); Bühlmann and van de Geer (2011); Negahban et al. (2012)) has shown that an appropriate formalization of this intuition is based on the event

$$\mathcal{T}_\lambda := \left\{ \frac{\|X^\top \varepsilon\|_\infty}{n} \leq \frac{\lambda}{4} \right\}. \quad (2)$$

When this event holds, then as long as the design matrix X is “well-behaved”, it is possible to obtain bounds on the sup-norm error of the Lasso estimate. There are various ways of characterizing well-behaved design matrices; of most relevance for sup-norm error control are mutual incoherence conditions (Bunea, 2008; Lounici, 2008) as well as ℓ_∞ -restricted eigenvalues (Ye and Zhang, 2010). See van de Geer and Bühlmann (2009) and Section 2.3 for further discussion of these design conditions.

In order to bring sharp focus to the calibration problem, rather than focusing on any particular design condition, it is useful to instead work under the generic assumption that the Lasso sup-norm error is controlled under the event \mathcal{T}_λ defined in equation (2). More formally, we state:

Assumption 1 ($\ell_\infty(C)$) *There is a numerical constant C such that conditional on \mathcal{T}_λ , the Lasso ℓ_∞ -error is upper bounded as $\|\hat{\beta}_\lambda - \beta^*\|_\infty \leq C\lambda$.*

As mentioned above, there are many conditions on the design matrix X under which Assumption $\ell_\infty(C)$ is valid, and we consider a number of them in the sequel.

With this set-up in place, we can now focus specifically on how to choose the regularization parameter. Since we can handle only finitely many tuning parameters in practice, we restrict ourselves to the selection of a tuning parameter among a finite but arbitrarily large number of choices. It is easy to see that $\lambda_{\max} := 2\|X^\top Y\|_\infty/n$ is the smallest tuning parameter for which $\hat{\beta}_\lambda$ equals zero. Accordingly, for a given positive integer $N \in \mathbb{N}$, let us form the grid

$$0 < \lambda_1 < \dots < \lambda_N = \lambda_{\max},$$

denoted by $\Lambda := \{\lambda_1, \dots, \lambda_N\}$ for short. Assumption $\ell_\infty(C)$ guarantees that the sup-norm error is proportional to λ whenever the event \mathcal{T}_λ holds; consequently, for a given probability

of error $\delta \in (0, 1)$, it is natural to choose the smallest λ for which event \mathcal{T}_λ holds with probability at least $1 - \delta$, assuming that it is finite. This criterion can be formalized as follows:

Definition 1 (Oracle tuning parameter) *For any constant $\delta \in (0, 1)$, the oracle tuning parameter is given by*

$$\lambda_\delta^* := \arg \min_{\lambda \in \Lambda} \{\mathbb{P}(\mathcal{T}_\lambda) \geq 1 - \delta\}. \quad (3)$$

Note that by construction, if we solve the Lasso using the oracle choice λ_δ^* , and if the design matrix X fulfills Assumption $\ell_\infty(C)$, then the resulting estimate satisfies the bound $\|\hat{\beta}_{\lambda_\delta^*} - \beta^*\|_\infty \leq C\lambda_\delta^*$ with probability at least $1 - \delta$. Unfortunately, the oracle choice is inaccessible to us, since we cannot compute the probability of the event \mathcal{T}_λ based on the observed data. However, as we now describe, we can mimic this performance, up to a factor of three, using a simple data-dependent procedure.

2.2 Adaptive Calibration Scheme

Let us now describe a data-dependent scheme for choosing the regularization parameter, referred to as Adaptive Calibration for ℓ_∞ (AV_∞):

Definition 2 (AV_∞) *Under Assumption $\ell_\infty(C)$ and for a given constant $\bar{C} \geq C$, Adaptive Calibration for ℓ_∞ (AV_∞) selects the tuning parameter*

$$\hat{\lambda} := \min \left\{ \lambda \in \Lambda \mid \max_{\substack{X, X' \in \Lambda \\ X, X' \geq \lambda}} \left[\frac{\|\hat{\beta}_X - \hat{\beta}_{X'}\|_\infty}{\lambda + \lambda'} - \bar{C} \right] \leq 0 \right\}. \quad (4)$$

The definition is based on tests for sup-norm differences of Lasso estimates with different tuning parameters. We stress that Definition 2 requires neither prior knowledge about the regression vector nor about the noise.

The tests in Definition 2 can be formulated in terms of the binary random variables

$$\hat{\tau}_{\lambda_j} := \prod_{k=j}^N \mathbb{1} \left\{ \frac{\|\hat{\beta}_{\lambda_j} - \hat{\beta}_{\lambda_k}\|_\infty}{\lambda_j + \lambda_k} - \bar{C} \leq 0 \right\} \quad \text{for } j \in [N],$$

from the AV_∞ tuning parameter $\hat{\lambda}$ can be computed as follows:

Data: $\hat{\beta}_{\lambda_1}, \dots, \hat{\beta}_{\lambda_N}, \bar{C}$
Result: $\hat{\lambda} \in \Lambda$

Set initial index: $j \leftarrow N$

while $\hat{\tau}_{\lambda_{j-1}} \neq 0$ **and** $j > 1$ **do**
 | Update index: $j \leftarrow j - 1$
end

Set output: $\hat{\lambda} \leftarrow \lambda_j$

Algorithm 1: Algorithm for AV_∞ in Definition 2.

This algorithm can be readily implemented and only requires the computation of one Lasso solution path. In strong contrast, k -fold Cross-Validation requires the computation of k solution paths. Consequently, the Lasso with AV_∞ can be computed about k times faster than Lasso with k -fold Cross-Validation.

The following result guarantees that the Lasso with AV_∞ method achieves the sup-norm error up to a constant pre-factor:

Theorem 3 (Optimality of AV_∞) *Suppose that condition $\ell_\infty(C)$ holds and the AV_∞ method is implemented with parameter $\bar{C} \geq C$. Then for any $\delta \in (0, 1)$, the AV_∞ output pair $(\hat{\lambda}, \hat{\beta}_\lambda)$ given by the rule (4) satisfies the bounds*

$$\hat{\lambda} \leq \lambda_\delta^* \quad \text{and} \quad \|\hat{\beta}_\lambda - \beta^*\|_\infty \leq 3\bar{C}\lambda_\delta^* \quad (5)$$

with probability at least $1 - \delta$.

Remark 4 (Relevance for estimation and variable selection) *The ℓ_∞ -bound from equation (5) directly implies that the AV_∞ scheme is adaptively optimal for the estimation of the regression vector β^* in ℓ_∞ -loss. As another important feature, Theorem 3 entails strong variable selection guarantees. First, the ℓ_∞ -bound implies that AV_∞ recovers all non-zero entries of the regression vector β^* that are larger than $3\bar{C}\lambda_\delta^*$ in absolute value. Additionally, by virtue of the bound $\hat{\lambda} \leq \lambda_\delta^*$, thresholding $\hat{\beta}_\lambda$ by $3\bar{C}\hat{\lambda}$ leads to exact support recovery if all non-zero entries of β^* are larger than $6\bar{C}\lambda_\delta^*$ in absolute value. In strong contrast, standard calibration schemes are not equipped with comparable variable selection guarantees, and there is no theoretically sound guidance for how to threshold standard schemes.*

We prove Theorem 3 in Appendix A; here let us make a few remarks about its consequences. First, if we knew the oracle value λ_δ^* defined in equation (3), then under Assumption $\ell_\infty(C)$, the Lasso estimate $\hat{\beta}$ would satisfy the ℓ_∞ -bound $\|\hat{\beta} - \beta^*\|_\infty \leq C\lambda_\delta^*$. Consequently, when the AV_∞ method is implemented with parameter C , then its sup-norm error is optimal up to a factor of three. For standard calibration schemes, among them Cross-Validation, no comparable guarantees are available in the literature. In fact, we are not aware of any finite sample guarantees for standard calibration schemes.

We point out that Theorem 3—in contrast to asymptotic results or results with unspecified constants—provides explicit guarantees for arbitrary sample sizes. Moreover, Theorem 3 does not presume prior knowledge about the regression vector or the noise distribution and allows, in particular, for correlated, heavy-tailed noise. From the perspective of theoretical sharpness, the best choice for \bar{C} is $\bar{C} = C$. However, Theorem 3 shows that it also suffices to know an upper bound for C . We provide more details on choices of C and \bar{C} below.

We finally observe that the specific choice of the grid enters Theorem 3 only via the oracle. Indeed, for any choice of the grid, Theorem 3 ensures that $\hat{\lambda}$ performs as well as the oracle tuning parameter λ_δ^* , which is the “best” tuning parameter on the grid.

2.3 Conditions on the Design Matrix for ℓ_∞ -guarantees

Let us now describe some conditions on the design matrix X that are sufficient for Assumption $\ell_\infty(C)$. We stress that these are conditions to ensure that the Lasso satisfies

ℓ_∞ -bounds; importantly, our method itself does not impose any additional restrictions. We defer all proofs of the results stated here to Appendix B and, for simplicity, we assume in the following that the sample covariance $\hat{\Sigma} := X^\top X/n$ has been normalized such that $\hat{\Sigma}_{jj} = 1$ for all $j \in [p]$.

The significance of the event \mathcal{T}_λ lies in the following implication: when \mathcal{T}_λ holds, then it can be shown (e.g., Bickel et al. (2009); Bühlmann and van de Geer (2011); Negahban et al. (2012)) that the Lasso error $\hat{\Delta} := \hat{\beta}_\lambda - \beta^*$ must belong to the cone

$$\mathcal{C}(S) := \{\Delta \in \mathbb{R}^p \mid \|\Delta_{S^c}\|_1 \leq 2\|\Delta_S\|_1\}, \quad (6)$$

where S denotes the support of β^* , and S^c its complement. Accordingly, all known conditions involve controlling the behavior of the sample covariance matrix $\hat{\Sigma}$ for vectors lying within this cone.

The most directly stated sufficient condition is based on lower bounding the ℓ_∞ -restricted eigenvalue: there exists some $\gamma > 0$ such that

$$\|\hat{\Sigma}\Delta\|_\infty \geq \gamma\|\Delta\|_\infty \quad \text{for all } \Delta \in \mathcal{C}(S). \quad (7)$$

See van de Geer and Bühlmann (2009) for an overview of various conditions for the Lasso, and their relations. Based on (7), we prove in Appendix B.1 the following result:

Lemma 5 (ℓ_∞ -restricted eigenvalue) *Suppose that $\hat{\Sigma}$ satisfies the γ -RE condition (7) and that \mathcal{T}_λ holds. Then Assumption $\ell_\infty(C)$ is valid with $C = \frac{5}{4\gamma}$.*

Although this result is cleanly stated, the RE condition cannot be verified in practice, since it involves the unknown support set S . Accordingly, let us now state some sufficient and verifiable conditions for obtaining bounds on the restricted eigenvalues, and hence for verifying Assumption $\ell_\infty(C)$.

For a given integer $\tilde{s} \in [2, p]$ and scalar $\nu > 0$, let us say that the sample covariance $\hat{\Sigma}$ is diagonally dominant with parameters (\tilde{s}, ν) if

$$\max_{\substack{T \subset [p] \\ |T| = \tilde{s}}} \sum_{k \in T} |\hat{\Sigma}_{jk}| < \nu \quad \text{for all } j \in [p]. \quad (8)$$

In the context of this definition, the reader should recall that we have assumed that $\hat{\Sigma}_{jj} = 1$ for all $j \in [p]$. Note that this condition can be verified in polynomial-time, since the subset T achieving the maximum in row j can be obtained simply by sorting the entries $\{|\hat{\Sigma}_{jk}|, k \in [p] \setminus j\}$. The significance of this condition lies in the following result:

Lemma 6 (Diagonal dominance of order \tilde{s}) *Suppose that $\tilde{s} \geq 9|S|$ and $\hat{\Sigma}$ is \tilde{s} -order diagonally dominant with parameter $\nu \in [0, 1)$. Then under the event \mathcal{T}_λ , Assumption $\ell_\infty(C)$ is valid with $C = \frac{5}{4(1-\nu)}$.*

See Appendix B.2 for the proof.

It is worth noting that the diagonal dominance condition is weaker than the pairwise incoherence conditions that have been used in past work on sup-norm error (Lounici, 2008). The pairwise incoherence of the sample covariance is given by $\rho(\hat{\Sigma}) = \max_{j \neq k} |\hat{\Sigma}_{jk}|$. If the pairwise incoherence satisfies the bound $\rho(\hat{\Sigma}) \leq \nu/\tilde{s}$, then it follows that $\hat{\Sigma}$ is diagonally dominant with parameters (\tilde{s}, ν) .

By combining Lemma 6 with Theorem 3, we obtain the following corollary:

Corollary 7 Suppose that $\tilde{s} \geq 9|S|$ and $\tilde{\Sigma}$ is \tilde{s} -order diagonally dominant with parameter $\nu \in [0, 1)$. Then for any $\delta \in (0, 1)$, the AV_∞ method with $C = \frac{5}{4(1-\nu)}$ returns an estimate $\hat{\beta}_\lambda$ such that

$$\|\hat{\beta}_\lambda - \beta^*\|_\infty \leq \frac{15}{4(1-\nu)} \lambda_s^* \quad (9)$$

with probability at least $1 - \delta$.

Another sufficient condition for the sup-norm optimality of AV_∞ is a design compatibility condition due to van de Geer (2007). For each index $j \in [p]$, suppose that we define the deterministic vector

$$\eta^j \in \arg \min_{\substack{\beta \in \mathbb{R}^p \\ \beta_j = -1}} \left\{ \frac{\|X\beta\|_2^2}{n} + \sqrt{\frac{\log(p)}{n}} \|\beta\|_1 \right\}.$$

Note that this optimization problem defining the vector regression of the j th column of the design matrix on the set of all other columns, where we have imposed an l_1 -penalty with weight $\sqrt{\frac{\log(p)}{n}}$. We can then derive the following sup-norm bound for the Lasso.

Lemma 8 (Lasso bound under compatibility) Assume that X fulfills the compatibility condition

$$\min_{\|\beta^s\|_1 \leq 3\|\beta^s\|_1} \left\{ \frac{\sqrt{|S|} \|X\beta\|_2}{\sqrt{n} \|\beta^s\|_1} \right\} \geq t \quad (\text{Compatibility})$$

for a constant $t > 0$. Additionally, assume that

$$\sup_{j \in [p]} \frac{|S|}{t^2 \|\eta^j\|_1} \leq \frac{1}{\log n} \sqrt{\frac{n}{\log p}}.$$

Then under the event \mathcal{F}_λ , Assumption $\ell_\infty(C)$ is valid with

$$C := \left(\frac{3}{4} + \frac{1}{\log(n)} \right) \max_{j \in [p]} \frac{\|\eta^j\|_2 / n + \sqrt{\log(p)/n} \|\eta^j\|_1^{-j/2}}{\|\eta^j\|_1}.$$

This bound is a consequence of results in (van de Geer, 2014); the proof is deferred to Section B.3. We are now ready to state the optimality of AV_∞ with respect to this bound.

Corollary 9 (Optimality of AV_∞) Assume that the assumptions in Lemma 8 are met. Then for any constant $\delta > 0$, the following bound for Lasso AV_∞ with $\bar{C} = C$, and C as above, holds with probability at least $1 - \delta$:

$$\|\hat{\beta}_\lambda - \beta^*\|_\infty \leq 3C\lambda_s^* \quad (10)$$

This result demonstrates the optimality of AV_∞ for sup-norm loss under the compatibility condition.

Remark 10 (Constant \bar{C} in practice) The optimal choice is $\bar{C} = C$ in view of our theoretical results. The constant C (or an upper bound of it) can be readily computed, because it depends only on X (cf. Lemma 8) or on X and an upper bound on s (cf. Lemma 6). However, we propose the universal choice $C = 0.75$ for all practical purposes. Note that accurate support recovery and ℓ_∞ -estimation is possible only if the design is near orthogonal. A direct computation yields the bound $\|\beta_\lambda - \beta^*\|_\infty \leq C\lambda$ with $C = 0.75$ for orthogonal design. Letting $\alpha \rightarrow \infty$ in Theorem 1 due to Lounici (2008) yields the same bound with $C \approx 0.75$ for near orthogonal designs. This provides strong theoretical support for the choice $\bar{C} = 0.75$. The empirical evidence in Section 3 indicates that a further calibration is indeed not necessary.

3. Simulations

In this section, we perform experiments on both simulated and real data to demonstrate the practical performance of AV_∞ .

3.1 Simulated Data

We simulate data from linear regression models as in equation (Model) with $n = 200$ observations and $p \in \{300, 900\}$ parameters. More specifically, we sample each row of the design matrix $X \in \mathbb{R}^{n \times p}$ from a p -dimensional normal distribution with mean 0 and covariance matrix $(1 - \kappa)I + \kappa\mathbb{1}$, where I is the identity matrix, $\mathbb{1} := (1, \dots, 1)^T (1, \dots, 1)$ is the matrix of ones, and $\kappa \in \{0, 0.2, 0.4\}$ is the magnitude of the mutual correlations. For the entries of the noise $\varepsilon \in \mathbb{R}^n$, we take the one-dimensional normal distribution with mean 0 and variance 1. The entries of β^* are first set to 0 except for 6 uniformly at random chosen entries that are each set to 1 or -1 with equal probability. The entire vector β^* is then rescaled such that the signal-to-noise ratio $\|X\beta^*\|_2^2/n$ is equal to 5. We finally consider a grid of 100 tuning parameters $\lambda := \{\lambda_{\max}/1.3^0, \lambda_{\max}/1.3^1, \dots, \lambda_{\max}/1.3^{99}\}$ with $\lambda_{\max} := 2\|X^T Y\|_\infty/n$. We run 100 experiments for each set of parameters and report the corresponding means (thick, colored bars) and standard deviations (thin, black lines). All computations are conducted with the software R (R Core Team, 2013) and the glmnet package (Friedman et al., 2010). While we restrict the presentation to the parameter settings described, we found similar results over a wide range of settings.

We compare the sup-norm and variable selection performance of the following three procedures:

- Oracle: Lasso with the tuning parameter that minimizes the ℓ_∞ loss (this tuning parameter is unknown in practice);
- AV_∞ : Lasso with AV_∞ and $\bar{C} = 0.75$;
- Cross-Validation: Lasso with 10-fold Cross-Validation.

Our choice $\bar{C} = 0.75$ is motivated by a theorem due to Lounici (2008) in the regime $\alpha \rightarrow \infty$; see Remark 10 for details.

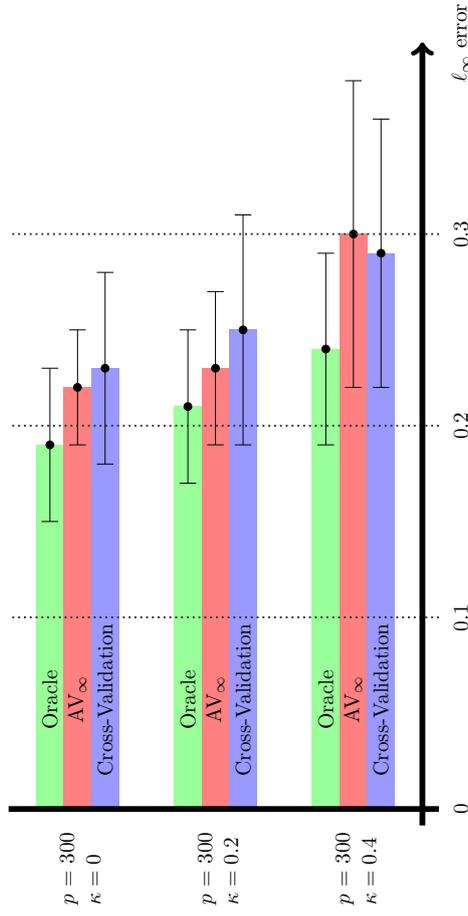


Figure 1: Sup-norm error $\|\hat{\beta}_\lambda - \beta^*\|_\infty$ of the Lasso with three different calibration schemes for the tuning parameter λ . Depicted are the results for three simulation settings that differ in the correlation level κ . The simulation settings and the calibration schemes are specified in the body of the text.

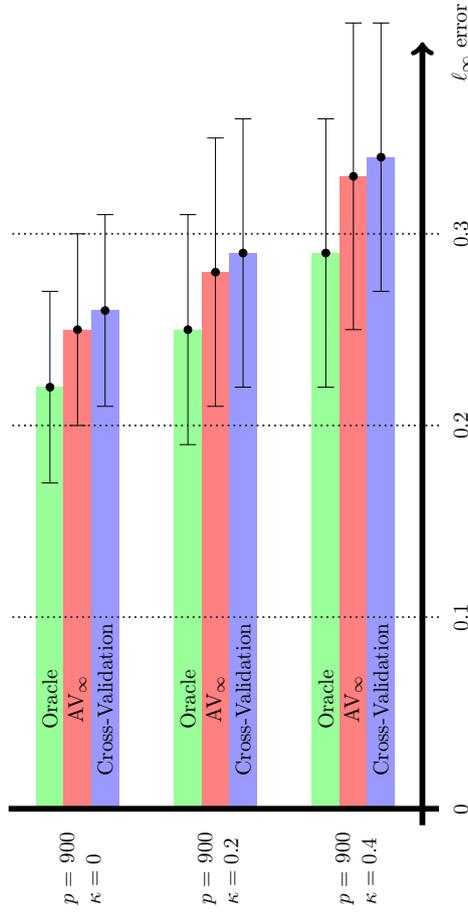


Figure 2: Sup-norm error $\|\hat{\beta}_\lambda - \beta^*\|_\infty$ of the Lasso with three different calibration schemes for the tuning parameter λ . Depicted are the results for three simulation settings that differ in the correlation level κ . The simulation settings and the calibration schemes are specified in the body of the text.

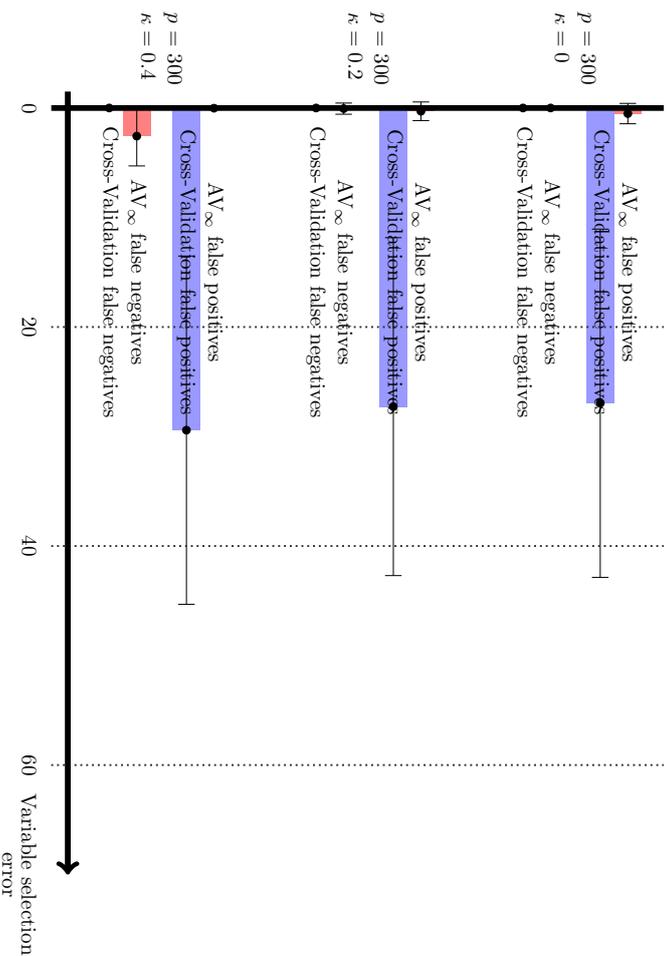


Figure 3: Number of false positives $\{|j : \beta_j^* = 0, (\hat{\beta}_\lambda)_j \neq 0\}$ and false negatives $\{|j : \beta_j^* \neq 0, (\hat{\beta}_\lambda)_j = 0\}$ of the Lasso with AV_∞ and Cross-Validation as calibration schemes for the tuning parameter λ . For AV_∞ , the safe threshold described after Theorem 3 is applied. The simulations settings correspond to those in Figure 1.

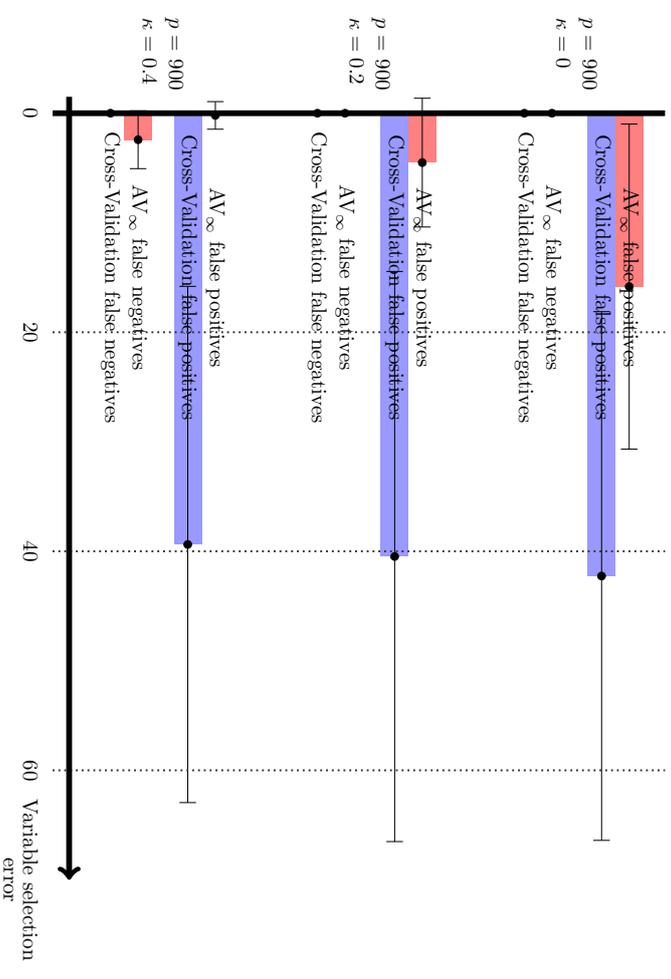


Figure 4: Number of false positives $\{|j : \beta_j^* = 0, (\hat{\beta}_\lambda)_j \neq 0\}$ and false negatives $\{|j : \beta_j^* \neq 0, (\hat{\beta}_\lambda)_j = 0\}$ of the Lasso with AV_∞ and Cross-Validation as calibration schemes for the tuning parameter λ . For AV_∞ , the safe threshold described after Theorem 3 is applied. The simulations settings correspond to those in Figure 2.

Sup-norm error: In Figures 1 and 2, we compare the ℓ_∞ error of the four procedures. We observe that AV_∞ outperforms Cross-Validation for most settings under consideration. We also mention that the same conclusions can be drawn if the normal distribution for the noise is replaced by other, possibly heavy-tailed distributions (for conciseness, we do not show the outputs).

Variable selection: In Figures 3 and 4, we compare the variable selection performance of AV_∞ and Cross-Validation. More specifically, we compare the number of false positives $\{|j : \beta_j^* = 0, (\hat{\beta}_\lambda)_j \neq 0\}$ and the number of false negatives $\{|j : \beta_j^* \neq 0, (\hat{\beta}_\lambda)_j = 0\}$. In contrast to Cross-Validation, AV_∞ allows for a safe threshold of size $3C\hat{\lambda}$ (recall the discussion after Theorem 3). Therefore, we report the results of Lasso with AV_∞ and an additional threshold of size $3C\hat{\lambda}$ applied to each component (that is, we consider the vector with entries $(\hat{\beta}_\lambda)_j \mathbb{1}\{|\hat{\beta}_\lambda|_j \geq 3C\hat{\lambda}\}$), and we report the results of Lasso with Cross-Validation (without threshold). We observe that, as compared to Cross-Validation, AV_∞ with subsequent thresholding can lead to a considerably smaller number of false positives, while keeping the number of false negatives on a low level. Note that one could perform a similar thresholding of the Cross-Validation solution, but unlike for AV_∞ , there is no theory to guide the choice of the threshold. This problem also applies to other standard calibration schemes.

Computational complexity: Cross-Validation with 10 folds requires the computation 10 Lasso paths, while AV_∞ requires the computation of only one Lasso path - or even less. AV_∞ is therefore about 10 times more efficient than 10-fold Cross-Validation.

Let us conclude with remarks on the scope of the simulations. First, many methods have been proposed for tuning the regularization parameter in the Lasso, including Cross-Validation, BIC and AIC-type criteria, Stability Selection (Meinshausen and Bühlmann, 2010), LinSelect (Baraud et al., 2014; Giraud et al., 2012), permutation approaches (Sabourin et al., 2015), and many more. On top of that, there are many modifications and extensions of the Lasso itself, including BoLasso (Bach, 2008), Square-Root/Scaled Lasso (Antoniadis, 2010; Belloni et al., 2011; Städler et al., 2010; Sun and Zhang, 2012), SCAD (Fan and Li, 2001), MCP (Zhang, 2010), and others. Detailed comparisons among the selection schemes and the methods can be found in the cited papers. We also refer to Leeb and Pötscher (2008) for theoretical insights about limitations of the methods.

In our simulations, we instead focus on the Lasso and, since we are not aware of guarantees similar to ours for any selection scheme, we compare to the most popular and most extensively studied selection scheme, Cross-Validation. This comparison shows that, beyond its theoretical properties and the easy and efficient implementation, AV_∞ is also a competitor in numerical experiments.

3.2 Riboflavin Production in *B. subtilis*

We now consider variable selection for a data set that describes the production of riboflavin (vitamin B₂) in *B. subtilis* (*Bacillus subtilis*), see (Bühlmann et al., 2014). The data set comprises the expressions of $p = 4088$ genes and the corresponding riboflavin production rates for $n = 71$ strains of *B. subtilis*. We apply AV_∞ and then impose the threshold $3C\hat{\lambda}$.

The resulting genes and the corresponding parameter values are given in the first column of Table 1. We see that these results commensurate with the results from previous

AV_∞	Stability Selection	B-TREX
YXLD_at -0.405	YXLD_at	YXLD_at
YOAB_at -0.420	YOAB_at	YOAB_at
YEBC_at -0.146	LYSC_at	YXLE_at
ARGF_at -0.313		
XHLB_at 0.278		

Table 1: Variable selection results for the riboflavin data set. The first column depicts the genes and the corresponding parameter values yielded by AV_∞ . The second and third column depict the genes returned by approaches based on Stability Selection and TREX.

approaches based on Stability Selection (Bühlmann et al., 2014) and TREX (Lederer and Müller, 2015), which are given in the third and fourth column.

4. Conclusions

We have introduced a novel method for sup-norm calibration, known as AV_∞ , that is equipped with finite sample guarantees for estimation in ℓ_∞ -loss and for variable selection. Moreover, we have shown that AV_∞ allows for simple and fast implementations. These properties make AV_∞ a competitive algorithm, as standard methods such as Cross-Validation are computationally more demanding and lack non-asymptotic guarantees.

In order to bring sharp focus to the issue, we have focused this paper exclusively on the calibration of the Lasso. However, we suspect that the methods and techniques developed here could be more generally applicable, for instance to problems with nonconvex penalties (e.g., SCAD, MCP). In particular, the paper (Loh and Wainwright, 2014) provides guarantees for ℓ_∞ -recovery using such nonconvex methods, which could be combined with our results. Another interesting direction for future work is the use of our methods for more general decomposable penalty functions (Negahban et al., 2012), including the nuclear norm that is often used in matrix estimation.

We also stress that our goals are ℓ_∞ -estimation and variable selection, which are feasible only under strict conditions on the design matrix. Other objectives, including prediction and ℓ_2 -estimation, can typically be achieved under less stringent conditions. However, the corresponding oracle inequalities contain quantities (such as the sparsity level) that are typically unknown in practice. Adaptations of our method to objectives beyond the ones considered here thus need further investigation. We refer to (Chételet et al., 2014) for ideas in this direction. However, there might be no approach that is uniformly optimal for all objectives, see also the papers (Yang, 2005; Zhao and Yu, 2006).

Finally, as pointed out by one of the reviewers, another field for further study is model misspecification. It would be interesting to see how robust the Lasso with the AV_∞ scheme is with respect to, for example, non-linearities in the model.

Acknowledgements

We thank Sara van de Geer and Sébastien Loustau for the inspiring discussions. We also thank the reviewers for the careful reading of the manuscript and the insightful comments. This work was partially supported by NSF Grant DMS-1107000, and Air Force Office of Scientific Research AFOSR-FA9550-14-1-0016 to MJW.

Appendix A. Proof of Theorem 1

Define the event $\mathcal{T}_\delta^* := \left\{ \frac{\|\mathbf{X}^\top \varepsilon\|_\infty}{n} \leq \frac{\lambda_\delta^*}{4} \right\}$ and note that $\mathbb{P}[\mathcal{T}_\delta^*] \geq 1 - \delta$ by our definition of the oracle tuning parameter in (3). Thus, it suffices to show that the two bounds hold conditioned on the event \mathcal{T}_δ^* .

Bound on $\hat{\lambda}$: To show that $\hat{\lambda} \leq \lambda_\delta^*$, we proceed by proof by contradiction. If $\hat{\lambda} > \lambda_\delta^*$, then the definition of the AV $_\infty$ method implies that there must exist two tuning parameters $\lambda', \lambda'' \geq \lambda_\delta^*$ such that

$$\|\hat{\beta}_{\lambda'} - \hat{\beta}_{\lambda''}\|_\infty > \bar{C}(\lambda' + \lambda''). \quad (11)$$

However, since \mathcal{T}_λ and $\mathcal{T}_{\lambda''}$ are both subsets of \mathcal{T}_δ^* , Assumption $\ell_\infty(C)$ implies that we must have the simultaneous inequalities $\|\hat{\beta}_{\lambda'} - \beta^*\|_\infty \leq C\lambda'$ and $\|\hat{\beta}_{\lambda''} - \beta^*\|_\infty \leq C\lambda''$. Combined with the triangle inequality, we find that

$$\|\hat{\beta}_{\lambda'} - \hat{\beta}_{\lambda''}\|_\infty \leq \|\hat{\beta}_{\lambda'} - \beta^*\|_\infty + \|\beta^* - \hat{\beta}_{\lambda''}\|_\infty \leq C(\lambda' + \lambda'').$$

Since $\bar{C} \geq C$, this upper bound contradicts our earlier conclusion (11) and, therefore, yields the desired claim.

Bound on the sup-norm error: On the event \mathcal{T}_δ^* , we have $\hat{\lambda} \leq \lambda_\delta^*$ and so the AV $_\infty$ definition implies that

$$\|\hat{\beta}_{\hat{\lambda}} - \hat{\beta}_{\lambda_\delta^*}\|_\infty \leq \bar{C}(\hat{\lambda} + \lambda_\delta^*) \leq 2\bar{C}\lambda_\delta^*.$$

Combined with the triangle inequality, we find that

$$\|\hat{\beta}_{\hat{\lambda}} - \beta^*\|_\infty \leq \|\hat{\beta}_{\hat{\lambda}} - \hat{\beta}_{\lambda_\delta^*}\|_\infty + \|\hat{\beta}_{\lambda_\delta^*} - \beta^*\|_\infty \leq 2\bar{C}\lambda_\delta^* + \|\hat{\beta}_{\lambda_\delta^*} - \beta^*\|_\infty.$$

Finally, under \mathcal{T}_δ^* and $\bar{C} \geq C$, Assumption $\ell_\infty(C)$ implies that $\|\hat{\beta}_{\lambda_\delta^*} - \beta^*\|_\infty \leq C\lambda_\delta^* \leq \bar{C}\lambda_\delta^*$, and combining the pieces completes the proof. \blacksquare

Appendix B. Remaining Proofs for Section 2

In this appendix, we provide the proofs of Lemmas 5, 6, and 8.

B.1 Proof of Lemma 5

By the first-order stationarity conditions for an optimum, the Lasso solution $\hat{\beta}_\lambda$ must satisfy the stationary condition $\frac{1}{n}\mathbf{X}^\top(\mathbf{X}\hat{\beta}_\lambda - Y) + \lambda\hat{z} = 0$, where $\hat{z} \in \mathbb{R}^p$ belongs to the sub-differential of the ℓ_1 -norm at $\hat{\beta}_\lambda$. Since $Y = \mathbf{X}\beta^* + \varepsilon$, we find that

$$\hat{\Sigma}(\hat{\beta}_\lambda - \beta^*) = -\lambda\hat{z} + \frac{\mathbf{X}^\top \varepsilon}{n}.$$

Taking the ℓ_∞ -norm of both sides and applying the triangle inequality yields

$$\|\hat{\Sigma}(\hat{\beta}_\lambda - \beta^*)\|_\infty \leq \lambda\|\hat{z}\|_\infty + \left\| \frac{\mathbf{X}^\top \varepsilon}{n} \right\|_\infty \leq \lambda + \frac{\lambda}{4} = \frac{5}{4}\lambda.$$

using the bound from event \mathcal{T}_λ , and the fact that $\|\hat{z}\|_\infty \leq 1$, by definition of the ℓ_1 -sub-differential. As noted previously, under the event \mathcal{T}_λ , the error vector $\Delta = \hat{\beta}_\lambda - \beta^*$ belongs to the cone $\mathcal{C}(S)$ in (6), so that the γ -RE condition can be applied so as to obtain the lower bound $\|\hat{\Sigma}(\hat{\beta}_\lambda - \beta^*)\|_\infty \geq \gamma\|\hat{\beta}_\lambda - \beta^*\|_\infty$. Combining the pieces concludes the proof. \blacksquare

B.2 Proof of Lemma 6

Since $\Delta \in \mathcal{C}(S)$, we have

$$\|\Delta\|_2^2 \leq 9\|\Delta_S\|_2^2 \leq 9|S|\|\Delta_S\|_2^2 \leq 9|S|\|\Delta\|_1\|\Delta\|_\infty,$$

which implies $\|\Delta\|_1 \leq 9|S|\|\Delta\|_\infty$. In view of Lemma 5, it thus suffices to prove the lower bound

$$\|\hat{\Sigma}\Delta\|_\infty \geq (1 - \nu)\|\Delta\|_\infty \quad \text{for all } \Delta \in A := \mathbb{B}_1(9|S|) \cap \mathbb{B}_\infty(1), \quad (12)$$

where we set $\mathbb{B}_d(r) := \{\beta \in \mathbb{R}^p : \|\beta\|_d \leq r\}$ for $d \in [0, \infty]$ and $r \geq 0$. We claim that

$$\underbrace{\mathbb{B}_1(9|S|) \cap \mathbb{B}_\infty(1)}_A \subseteq 2 \operatorname{cl} \operatorname{conv} \underbrace{\{\mathbb{B}_0(9|S|) \cap \mathbb{B}_\infty(1)\}}_B, \quad (13)$$

where $\operatorname{cl} \operatorname{conv}$ denotes the closed convex hull. Taking this as given for the moment, let us use it to prove the desired claim. We have

$$\begin{aligned} \max_{\Delta \in A} \frac{\|\hat{\Sigma} - I\Delta\|_\infty}{\|\Delta\|_\infty} &= \max_{\Delta \in A/2} \frac{\|\hat{\Sigma} - I\Delta\|_\infty}{\|\Delta\|_\infty} \leq \max_{\Delta \in B} \frac{\|\hat{\Sigma} - I\Delta\|_\infty}{\|\Delta\|_\infty} \leq \max_{j \in [p]} \max_{\substack{T=[9|S|] \\ T \subseteq [p]^j}} \sum_{k \in T} |\hat{\Sigma}_{jk}| \leq \nu \end{aligned} \quad (14)$$

using the diagonal dominance (8). Combined with the triangle inequality, the lower bound (12) follows.

It remains to prove the inclusion (13). Since both A and B are closed and convex, it suffices to prove that $\phi_A(\theta) \leq \phi_B(\theta)$ for all $\theta \in \mathbb{R}^p$, where $\phi_A(\theta) := \sup_{z \in A} \langle z, \theta \rangle$ and $\phi_B(\theta) := \sup_{z \in B} \langle z, \theta \rangle$ are the support functions. For a given vector $\theta \in \mathbb{R}^p$, let T be the

subset indexing its top $9|S|$ values in absolute value. By construction, we are guaranteed to have the bound $9|S|\|\theta_{T^c}\|_\infty \leq \|\theta_T\|_1$, and consequently

$$\begin{aligned} \sup_{z \in A} \langle (z_T, \theta_T) + (z_{T^c}, \theta_{T^c}) \rangle \phi_A(\theta) &\leq \sup_{z \in A} (\|z_T\|_\infty \|\theta_T\|_1 + \|z_{T^c}\|_1 \|\theta_{T^c}\|_\infty) \\ &\leq \|\theta_T\|_1 + 9|S| \|\theta_{T^c}\|_\infty \\ &\leq 2\|\theta_T\|_1. \end{aligned}$$

On the other hand, for this same subset T , we have $\phi_B(\theta) \geq \sup_{z \in B} \langle z_T, \theta_T \rangle = 2\|\theta_T\|_1$, which completes the proof. \blacksquare

B.3 Proof of Lemma 8

In order to prove Lemma 8, we use a somewhat simplified version of a recent result due to van de Geer (2014). So as to simplify notation, we first define the norms $\|a\|_j := |a_j|$ and $\|a\|_{-j} := \sum_{i \neq j} |a_i|$ for any vector a . We then have:

Lemma 11 (van de Geer (2014), Lemma 2.1) *Given any tuning parameter $\lambda > 0$, it holds that*

$$\|\hat{\beta}_\lambda - \beta^*\|_j \leq D_j \left(\frac{\|X^T \varepsilon\|_\infty}{n} + \frac{\sqrt{\log(p)} \|\hat{\beta}_\lambda - \beta^*\|_{-j}}{2\sqrt{n} \|\gamma^j\|_1} + \frac{\lambda}{2} \right)$$

where for each $j \in [p]$,

$$D_j := \frac{\|\gamma^j\|_1}{\|X\gamma^j\|_2^2/n + \sqrt{\log(p)/n} \|\gamma^j\|_{-j}/2}.$$

This result provides a specific bound for each coordinate of Lasso. Lemma 8 can then readily be proven using this result together with Theorem 6.1 from Bühlmann and van de Geer (2011). \blacksquare

Appendix C. Strong Correlations

In this paper, we assume that the correlations in design matrix are small, which is needed for precise ℓ_∞ -estimation and variable selection. In the interest of completeness, however, we add here two simulations where the correlations are large. Overall, we use the same settings as described in the main part of the paper, but we set $\kappa = 0.9$. The results are summarized in Figure 5 (note that the x-scale in the upper part of the figure is different from the scales of the corresponding plots in the main part of the paper). We find that AV_∞ misses about half of the pertinent variables but has almost no false positives. Cross-Validation, on the other hand, has less false negatives but selects many irrelevant variables. As expected, none of the methods, including the oracle, provide accurate ℓ_∞ -estimation.

References

A. Antoniadis. Comments on: ℓ_1 -penalization for mixture regression models. *Test*, 19(2): 257–258, 2010.

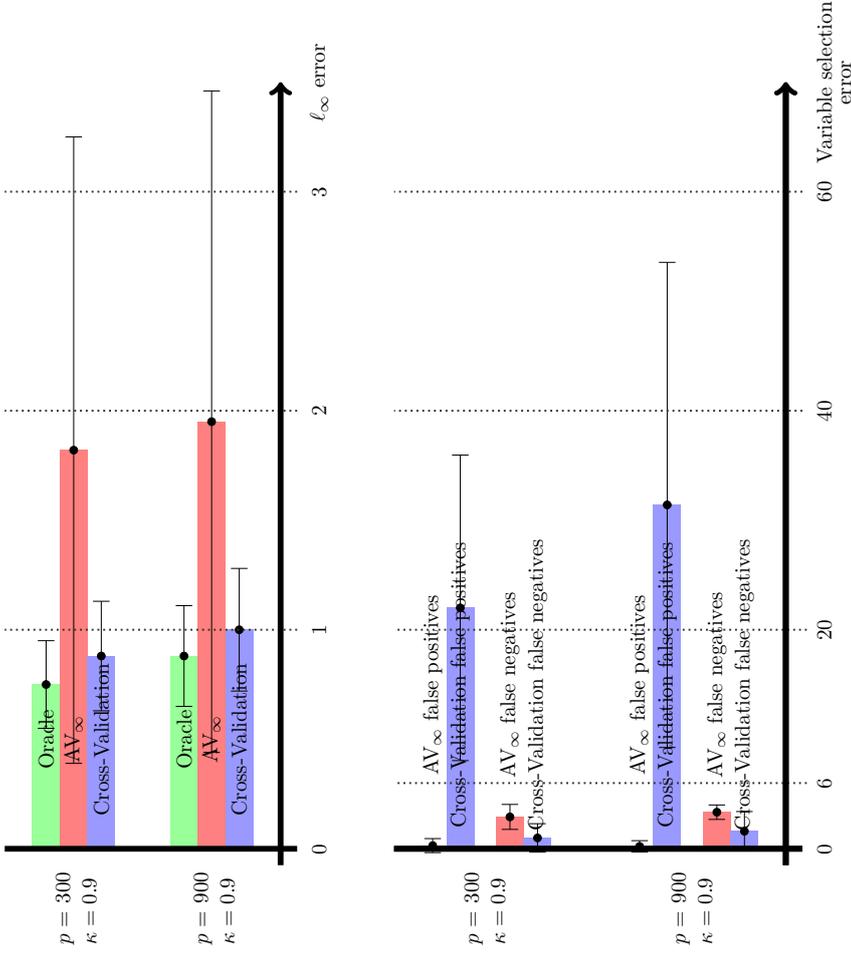


Figure 5: Sup-norm and variable selection errors of the Lasso with three/two different calibration schemes for the tuning parameter λ . Depicted are the results for two simulation settings that differ in the number of parameters p . The simulation settings and the calibration schemes are specified in the main part of the paper.

F. Bach. Bolasso: model consistent lasso estimation through the bootstrap. In *Proceedings of the 25th International Conference on Machine Learning*, pages 33–40, 2008.

Y. Baraud, C. Giraud, and S. Huet. Estimator selection in the gaussian setting. In *Ann. Inst. H. Poincaré Probab. Statist.*, volume 50, pages 1092–1119, 2014.

- A. Belloni, V. Chernozhukov, and L. Wang. Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika*, 98(4):791–806, 2011.
- P. Bickel, Y. Ritov, and A. Tsybakov. Simultaneous analysis of the Lasso and Dantzig selector. *Ann. Statist.*, 37(4):1705–1732, 2009.
- P. Bühlmann and S. van de Geer. *Statistics for high-dimensional data: Methods, theory and applications*. Springer Series in Statistics. Springer, 2011.
- P. Bühlmann, M. Kalisch, and L. Meier. High-dimensional statistics with a view toward applications in biology. *Annual Review of Statistics and Its Applications*, 1(1):255–278, 2014.
- F. Bunue. Honest variable selection in linear and logistic regression models via ℓ_1 and $\ell_1 + \ell_2$ penalization. *Electron. J. Stat.*, 2:1153–1194, 2008.
- D. Chételat, J. Lederer, and J. Salmon. Optimal two-step prediction in regression. *arXiv:1410.5014*, 2014.
- M. Chichignoud and J. Lederer. A robust, adaptive M-estimator for pointwise estimation in heteroscedastic regression. *Bernoulli*, 20(3):1560–1599, 2014.
- A. Dalalyan, M. Hebiri, and J. Lederer. On the Prediction Performance of the Lasso. *Bernoulli*, in press, 2014.
- J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Amer. Statist. Assoc.*, 96(456):1348–1360, 2001.
- J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.
- C. Giraud, S. Huet, and N. Verzelen. High-dimensional regression with unknown variance. *Statistical Science*, 27(4):500–518, 2012.
- M. Hebiri and J. Lederer. How correlations influence Lasso prediction. *IEEE Trans. Inform. Theory*, 59(3):1846–1854, 2013.
- J. Lederer and C. Müller. Don't fall for tuning parameters: Tuning-free variable selection in high dimensions with the trex. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.
- H. Leeb and B. Pötscher. Sparse estimators and the oracle property; or the return of Hodges' estimator. *J. Econometrics*, 142(1):201–211, 2008.
- O. Lepski. A problem of adaptive estimation in Gaussian white noise. *Teor. Veroyatnost. i Primenen.*, 35(3):459–470, 1990. ISSN 0040-361X.
- O. Lepski, E. Mammen, and V. Spokoiny. Optimal spatial adaptation to inhomogeneous smoothness: an approach based on kernel estimates with variable bandwidth selectors. *Ann. Statist.*, 25(3):929–947, 1997.
- P.-L. Loh and M. Wainwright. Support recovery without incoherence: A case for nonconvex regularization. *arXiv:1412.5632*, 2014.
- K. Lounici. Sup-norm convergence rate and sign concentration property of Lasso and Dantzig estimators. *Electron. J. Stat.*, 2:90–102, 2008.
- N. Meinshausen and P. Bühlmann. Stability selection. *J. R. Stat. Soc. Ser. B. Stat. Methodol.*, 72(4):417–473, 2010.
- S. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu. A unified framework for high-dimensional analysis of M -estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557, December 2012.
- R. Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013. <http://www.R-project.org/>.
- J. Sabourin, W. Valdar, and A. Nobel. A permutation approach for selecting the penalty parameter in penalized model selection. *Biometrics*, 71(4):1185–1194, 2015.
- N. Städler, P. Bühlmann, and S. van de Geer. ℓ_1 -penalization for mixture regression models. *Test*, 19(2):209–256, 2010.
- T. Sun and C.-H. Zhang. Scaled sparse linear regression. *Biometrika*, 99(4):879–898, 2012.
- R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B*, 58(1):267–288, 1996.
- S. van de Geer. The deterministic Lasso. 2007 *Proc. Amer. Math. Soc.* [CD-ROM], see also www.stat.math.ethz.ch/~geer/lasso.pdf, 2007.
- S. van de Geer. Worst possible sub-directions in high-dimensional models. *J. Multivariate Anal.*, in press, 2014.
- S. van de Geer and P. Bühlmann. On the conditions used to prove oracle results for the Lasso. *Electron. J. Stat.*, 3:1360–1392, 2009.
- S. van de Geer and J. Lederer. The Lasso, correlated design, and improved oracle inequalities. *IMS Collections*, 9:303–316, 2013.
- Y. Yang. Can the strengths of aic and bic be shared? A conflict between model identification and regression estimation. *Biometrika*, 92(4):937–950, 2005.
- F. Ye and C.-H. Zhang. Rate minimaxity of the lasso and dantzig selector for the lq loss in lr balls. *J. Mach. Learn. Res.*, 11:3519–3540, 2010.
- C.-H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *Ann. Statist.*, pages 894–942, 2010.
- P. Zhao and B. Yu. On model selection consistency of Lasso. *J. Mach. Learn. Res.*, 7:2541–2563, 2006.

A Characterization of Linkage-Based Hierarchical Clustering

Margareta Ackerman

*Department of Computer Science
San Jose State University
San Jose, CA*

MARGARETA.ACKERMAN@SJSU.EDU

Shai Ben-David

*D.R.C. School of Computer Science
University of Waterloo
Waterloo, ON*

SHAI@CS.UWATERLOO.CA

Editor: Marina Meila

Abstract

The class of linkage-based algorithms is perhaps the most popular class of hierarchical algorithms. We identify two properties of hierarchical algorithms, and prove that linkage-based algorithms are the only ones that satisfy both of these properties. Our characterization clearly delineates the difference between linkage-based algorithms and other hierarchical methods. We formulate an intuitive notion of locality of a hierarchical algorithm that distinguishes between linkage-based and “global” hierarchical algorithms like bisecting k -means, and prove that popular divisive hierarchical algorithms produce clusterings that cannot be produced by any linkage-based algorithm.

1. Introduction

Clustering is a fundamental and immensely useful task, with many important applications. There are many clustering algorithms, and these algorithms often produce different results on the same data. Faced with a concrete clustering task, a user needs to choose an appropriate algorithm. Currently, such decisions are often made in a very ad hoc, if not completely random, manner. Users are aware of the costs involved in employing different clustering algorithms, such as running times, memory requirements, and software purchasing costs. However, there is very little understanding of the differences in the outcomes that these algorithms may produce.

It has been proposed to address this challenge by identifying significant properties that distinguish between different clustering paradigms (see, for example, Ackerman et al. (2010b) and Fisher and Van Ness (1971)). By focusing on the input-output behaviour of algorithms, these properties shed light on essential differences between them (Ackerman et al. (2010b, 2012)). Users could then choose desirable properties based on domain expertise, and select an algorithm that satisfies these properties.

In this paper, we focus hierarchical algorithms, a prominent class of clustering algorithms. These algorithms output dendrograms, which the user can then traverse to obtain the desired clustering. Dendrograms provide a convenient method for exploring multiple

clusterings of the data. Notably, for some applications the dendrogram itself, not any clustering found in it, is the desired final outcome. One such application is found in the field of phylogeny, which aims to reconstruct the tree of life.

One popular class of hierarchical algorithms is linkage-based algorithms. These algorithms start with singleton clusters, and repeatedly merge pairs of clusters until a dendrogram is formed. This class includes commonly-used algorithms such as single-linkage, average-linkage, complete-linkage, and Ward’s method.

In this paper, we provide a property-based characterization of hierarchical linkage-based algorithms. We identify two properties of hierarchical algorithms that are satisfied by all linkage-based algorithms, and prove that at the same time no algorithm that is not linkage-based can satisfy both of these properties.

The popularity of linkage-based algorithms leads to a common misconception that linkage-based algorithms are synonymous with hierarchical algorithms. We show that even when the internal workings of algorithms are ignored, and the focus is placed solely on their input-output behaviour, there are natural hierarchical algorithms that are not linkage-based. We define a large class of divisive algorithms that includes the popular bisecting k -means algorithm, and show that no linkage-based algorithm can simulate the input-output behaviour of any algorithm in this class.

2. Previous Work

Our work falls within the larger framework of studying properties of clustering algorithms. Several authors study such properties from an axiomatic perspective. For instance, Wright (1973) proposes axioms of clustering functions in a weighted setting, where every domain element is assigned a positive real weight, and its weight may be distributed among multiple clusters. A recent, and influential, paper in this line of work is Kleinberg’s impossibility result (Kleinberg (2003)), where he proposes three axioms of partitional clustering functions and proves that no clustering function can simultaneously satisfy these properties.

Properties have been used to study different aspects of clustering. Ackerman and Ben-David (2008) consider properties satisfied by clustering quality measures, showing that properties analogous to Kleinberg’s axioms are consistent in this setting. Meila (2005) studies properties of criteria for comparing clusterings, functions that map pairs of clusterings to real numbers, and identifies properties that are sufficient to uniquely identify several such criteria. Puzicha et al. (2000) explore properties of clustering objective functions. They propose a few natural properties of clustering objective functions, and then focus on objective functions that arise by requiring functions to decompose into additive form.

Most relevant to our work are previous results distinguishing linkage-based algorithms based on their properties. Most of these results are concerned with the single-linkage algorithm. In the hierarchical clustering setting, Jardine and Sibson (1971) and Carlsson and Mémoli (2010) formulate a collection of properties that define single linkage.

Zadeh and Ben-David (2009) characterize single linkage in the partitional setting where instead of constructing a dendrogram, clusters are merged until a given number of clusters remain. Finally, Ackerman et al. (2010a) characterize linkage-based algorithms in the same partitional setting in terms of a few natural properties. These results enable a comparison

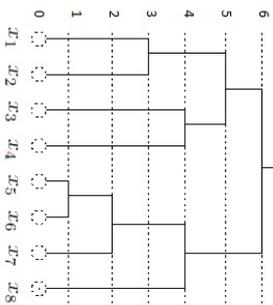


Figure 1: A dendrogram of domain set $\{x_1, \dots, x_8\}$. The horizontal lines represent levels and every leaf is associated with an element of the domain.

of the input-output behaviour of (a partitional variant of) linkage-based algorithms with other partitional algorithms.

In this paper, we characterize hierarchical linkage-based algorithms, which map data sets to dendrograms. Our characterization is independent of any stopping criterion. It enables the comparison of linkage-based algorithms to other hierarchical algorithms, and clearly delineates the differences between the input/output behaviour of linkage-based algorithms and other hierarchical methods.

3. Definitions

A *distance function* is a symmetric function $d : X \times X \rightarrow \mathbb{R}^+$, such that $d(x, x) = 0$ for all $x \in X$. The data sets that we consider are pairs (X, d) , where X is some finite domain set and d is a distance function over X . We say that a distance function d over X *extends* distance function d' over $X' \subseteq X$, denoted $d' \subseteq d$, if $d'(x, y) = d(x, y)$ for all $x, y \in X'$. Two distance functions d over X and d' over X' *agree* on a data set Y if $Y \subseteq X$, $Y \subseteq X'$, and $d(x, y) = d'(x, y)$ for all $x, y \in Y$.

A *k-clustering* $C = \{C_1, C_2, \dots, C_k\}$ of a data set X is a partition of X into k non-empty disjoint subsets of X (so, $\cup_i C_i = X$). A *clustering* of X is a k -clustering of X for some $1 \leq k \leq |X|$. For a clustering C , let $|C|$ denote the number of clusters in C . For $x, y \in X$ and clustering C of X , we write $x \sim_C y$ if x and y belong to the same cluster in C and $x \not\sim_C y$, otherwise.

Given a rooted tree T where the edges are oriented away from the root, let $V(T)$ denote the set of vertices in T , and $E(T)$ denote the set of edges in T . We use the standard interpretation of the terms leaf, descendant, parent, and child.

A dendrogram over a data set X is a binary rooted tree where the leaves correspond to elements of X . In addition, every node is assigned a level, using a level function (η); leaves are placed at level 0, parents have higher levels than their children, and no level is empty. See Figure 1 for an illustration. Formally,

Definition 1 (dendrogram) A dendrogram over (X, d) is a triple (T, M, η) where T is a binary rooted tree, $M : \text{leaves}(T) \rightarrow X$ is a bijection, and $\eta : V(T) \rightarrow \{0, \dots, h\}$ is onto (for some $h \in \mathbb{Z}^+ \cup \{0\}$) such that

1. For every leaf node $x \in V(T)$, $\eta(x) = 0$.
2. If $(x, y) \in E(T)$, then $\eta(x) > \eta(y)$.

Given a dendrogram $\mathcal{D} = (T, M, \eta)$ of X , we define a mapping from nodes to clusters $C : V(T) \rightarrow 2^X$ by $C(x) = \{M(y) \mid y \text{ is a leaf and a descendant of } x\}$. If $C(x) = A$, then we write $v(A) = x$. We think of $v(A)$ as the vertex (or node) in the tree that represents cluster A .

We say that $A \subseteq X$ is a cluster in \mathcal{D} if there exists a node $x \in V(T)$ so that $C(x) = A$. We say that a clustering $C = \{C_1, \dots, C_k\}$ of $X' \subseteq X$ is in \mathcal{D} if C_i is in \mathcal{D} for all $1 \leq i \leq k$. Note that a dendrogram may contain clusterings that do not partition the entire domain, and $\forall i \neq j$, $v(C_i)$ is not a descendant of $v(C_j)$, since $C_i \cap C_j = \emptyset$.

Definition 2 (sub-dendrogram) A sub-dendrogram of (T, M, η) rooted at $x \in V(T)$ is a dendrogram (T', M', η') where

1. T' is the subtree of T rooted at x ,
2. For every $y \in \text{leaves}(T')$, $M'(y) = M(y)$, and
3. For all $y, z \in V(T')$, $\eta'(y) < \eta'(z)$ if and only if $\eta(y) < \eta(z)$.

Definition 3 (Isomorphisms) A few notions of isomorphisms of structures are relevant to our discussion.

1. We say that (X, d) and (X', d') are isomorphic domains, denoted $(X, d) \cong_X (X', d')$, if there exists a bijection $\phi : X \rightarrow X'$ so that $d(x, y) = d'(\phi(x), \phi(y))$ for all $x, y \in X$.
2. We say that two clusterings (or partitions) C of some domain (X, d) and C' of some domain (X', d') are isomorphic clusterings, denoted $(C, d) \cong_C (C', d')$, if there exists a domain isomorphism $\phi : X \rightarrow X'$ so that $x \sim_C y$ if and only if $\phi(x) \sim_{C'} \phi(y)$.
3. We say that (T_1, η_1) and (T_2, η_2) are isomorphic trees, denoted $(T_1, \eta_1) \cong_T (T_2, \eta_2)$, if there exists a bijection $H : V(T_1) \rightarrow V(T_2)$ so that

- (a) for all $x, y \in V(T_1)$, $(x, y) \in E(T_1)$ if and only if $(H(x), H(y)) \in E(T_2)$, and
- (b) for all $x \in V(T_1)$, $\eta_1(x) = \eta_2(H(x))$.

4. We say that $\mathcal{D}_1 = (T_1, M_1, \eta_1)$ of (X, d) and $\mathcal{D}_2 = (T_2, M_2, \eta_2)$ of (X', d') are isomorphic dendrograms, denoted $\mathcal{D}_1 \cong_{\mathcal{D}} \mathcal{D}_2$, if there exists a domain isomorphism $\phi : X \rightarrow X'$ and a tree isomorphism $H : (T_1, \eta_1) \rightarrow (T_2, \eta_2)$ so that for all $x \in \text{leaves}(T_1)$, $\phi(M_1(x)) = M_2(H(x))$.

4. Hierarchical and Linkage-Based Algorithms

In the hierarchical clustering setting, linkage-based algorithms are hierarchical algorithms that can be simulated by repeatedly merging close clusters. In this section, we formally define hierarchical algorithms and linkage-based hierarchical algorithms.

4.1 Hierarchical Algorithms

In addition to outputting a dendrogram, we require that hierarchical clustering functions satisfy a few natural properties.

Definition 4 (Hierarchical clustering function) *A hierarchical clustering function F is a function that takes as input a pair (X, d) and outputs a dendrogram (T, M, η) . We require such a function, F , to satisfy the following:*

1. Representation Independence: *Whenever $(X, d) \cong_X (X', d')$, then $F(X, d) \cong_D F(X', d')$.*
2. Scale Invariance: *For any domain set X and any pair of distance functions d, d' over X , if there exists $c \in \mathbb{R}^+$ such that $d(a, b) = c \cdot d'(a, b)$ for all $a, b \in X$, then $F(X, d) = F(X, d')$.*
3. Richness: *For all data sets $\{(X_1, d_1), \dots, (X_k, d_k)\}$ where $X_i \cap X_j = \emptyset$ for all $i \neq j$, there exists a distance function \hat{d} over $\bigcup_{i=1}^k X_i$ that extends each of the d_i 's (for $i \leq k$), so that the clustering $\{X_1, \dots, X_k\}$ is in $F(\bigcup_{i=1}^k X_i, \hat{d})$.*

The last condition, richness, requires that by manipulating between-cluster distances every clustering can be produced by the algorithm. Intuitively, if we place the clusters sufficiently far apart, then the resulting clustering should be in the dendrogram.

In this work, we focus on distinguishing linkage-based algorithms from other hierarchical algorithms.

4.2 Linkage-Based Algorithms

The class of linkage-base algorithms includes some of the most popular hierarchical algorithms, such as single-linkage, average-linkage, complete-linkage, and Ward's method.

Every linkage-based algorithm has a linkage function that can be used to determine which clusters to merge at every step of the algorithm.

Definition 5 (Linkage Function) *A linkage function is a function*

$$\ell : \{(X_1, X_2, d) \mid d \text{ over } X_1 \cup X_2\} \rightarrow \mathbb{R}^+$$

such that,

1. ℓ is representation independent: *For all (X_1, X_2) and (X'_1, X'_2) , if $(\{X_1, X_2\}, d) \cong_C (\{X'_1, X'_2\}, d')$ then $\ell(X_1, X_2, d) = \ell(X'_1, X'_2, d')$.*
2. ℓ is monotonic: *For all (X_1, X_2, d) if d' is a distance function over $X_1 \cup X_2$ such that for all $x \sim_{\{X_1, X_2\}} y$, $d(x, y) = d'(x, y)$ and for all $x \not\sim_{\{X_1, X_2\}} y$, $d(x, y) \leq d'(x, y)$ then $\ell(X_1, X_2, d) \geq \ell(X_1, X_2, d')$.*

As in our characterization of partitional linkage-based algorithms, we assume that a linkage function has a countable range. Say, the set of non-negative algebraic real numbers. The following are the linkage-functions of some of the most popular linkage-based algorithms,

- **Single-linkage:** $\ell(A, B, d) = \min_{a \in A, b \in B} d(a, b)$
- **Average-linkage:** $\ell(A, B, d) = \sum_{a \in A, b \in B} d(a, b) / (|A| \cdot |B|)$
- **Complete-linkage:** $\ell(A, B, d) = \max_{a \in A, b \in B} d(a, b)$

For a dendrogram \mathcal{D} and clusters A and B in \mathcal{D} , if there exists x so that $\text{parent}(v(A)) = \text{parent}(v(B)) = x$, then let $\text{parent}(A, B) = x$, otherwise $\text{parent}(A, B) = \emptyset$.

We now define hierarchical linkage-based functions.

Definition 6 (Linkage-Based Function) *A hierarchical clustering function F is linkage-based if there exists a linkage function ℓ so that for all (X, d) , $F(X, d) = (T, M, \eta)$ where $\eta(\text{parent}(A, B)) = m$ if and only if $\ell(A, B)$ is minimal in $\{\ell(S, T) : S \cap T = \emptyset, \eta(S) < m, \eta(T) < m, \eta(\text{parent}(S)) \geq m, \eta(\text{parent}(T)) \geq m\}$.*

Note that the above definition implies that there exists a linkage function that can be used to simulate the output of F . We start by assigning every element of the domain to a leaf node. We then use the linkage function to identify the closest pair of nodes (with respect to the clusters that they represent), and repeatedly merge the closest pairs of nodes that do yet have parents, until only one such node remains.

4.3 Locality

We introduce a new property of hierarchical algorithms. Locality states that if we select a clustering from a dendrogram (a union of disjoint clusters that appear in the dendrogram), and run the hierarchical algorithm on the data underlying this clustering, we obtain a result that is consistent with the original dendrogram.

Definition 7 (Locality) *A hierarchical function F is local if for all X, d , and $X' \subseteq X$, whenever clustering $C = \{C_1, C_2, \dots, C_k\}$ of X' is in $F(X, d) = (T, M, \eta)$, then for all $1 \leq i \leq k$*

1. *Cluster C_i is in $F(X', d|_{X'}) = (T', M', \eta')$, and the sub-dendrogram of $F(X, d)$ rooted at $v(C_i)$ is also a sub-dendrogram of $F(X', d|_{X'})$ rooted at $v(C_i)$.*
2. *For all $x, y \in X'$, $\eta'(x) < \eta'(y)$ if and only if $\eta(x) < \eta(y)$.*

Locality is often a desirable property. Consider for example the field of phylogenetics, which aims to reconstruct the tree of life. If an algorithm clusters phylogenetic data correctly, then if we cluster any subset of the data, we should get results that are consistent with the original dendrogram.

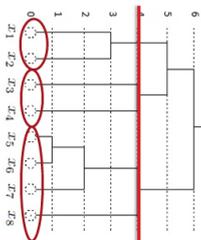


Figure 2: An example of an A-cut.

4.4 Outer Consistency

Clustering aims to group similar elements and separate dissimilar ones. These two requirements are often contradictory and algorithms vary in how they resolve this contradiction. Kleinberg (2003) proposed a formalization of these requirements in his “consistency” axiom for partitional clustering algorithms. Consistency requires that if within-cluster distances are decreased, and between-cluster distances are increased, then the output of a clustering function does not change.

Since then it was found that while many natural clustering functions fail consistency, most satisfy a relaxation, which requires that the output of an algorithm is not changed by increasing between-cluster distances (Ackerman et al. (2010b)). Given successfully clustered data, if points that are already assigned to different clusters are drawn even further apart, then it is natural to expect that, when clustering the resulting new data set, such points will not share the same cluster. Here we propose a variation of this requirement for the hierarchical clustering setting.

Given a dendrogram produced by a hierarchical algorithm, we select a clustering C from a dendrogram and pull apart the clusters in C (thus making the clustering C more pronounced). If we then run the algorithm on the resulting data, we can expect that the clustering C will occur in the new dendrogram. Outer consistency is a relaxation of the above property, making this requirement only on a subset of clusterings.

For a cluster A in a dendrogram \mathcal{D} , the A-cut of \mathcal{D} is a clustering in \mathcal{D} represented by nodes on the same level as $v(A)$ or directly below $v(A)$. For convenience, if node u is the root of the dendrogram, then assume its parent has infinite level, $\eta(\text{parent}(u)) = \infty$. Formally,

Definition 8 (A-cut) *Given a cluster A in a dendrogram $\mathcal{D} = (T, M, \eta)$, the A-cut of \mathcal{D} is $\text{cut}_A(\mathcal{D}) = \{C(u) \mid u \in V(T), \eta(\text{parent}(u)) > \eta(v(A)) \text{ and } \eta(u) \leq \eta(v(A))\}$.*

Note that for any cluster A in \mathcal{D} of (X, d) , the A-cut is a clustering of X , and A is one of the clusters in that clustering.

For example, consider the diagram in Figure 2. Let $A = \{x_3, x_4\}$. The horizontal line on level 4 of the dendrogram represents the intuitive notion of a cut. To obtain the corresponding clustering, we select all clusters represented by nodes on the line, and for

the remaining clusters, we choose clusters represented by nodes that lay directly below the horizontal cut. In this example, clusters $\{x_3, x_4\}$ and $\{x_5, x_6, x_7, x_8\}$ are represented by nodes directly on the line, and $\{x_1, x_2\}$ is a cluster represented by a node directly below the marked horizontal line.

Recall that a distance function d over X is (C, d) -outer-consistent if $d(x, y) = d(x, y)$ whenever $x \sim_C y$, and $d'(x, y) \geq d(x, y)$ whenever $x \not\sim_C y$.

Definition 9 (Outer-Consistency) *A hierarchical function F is outer consistent if for all (X, d) and any cluster A in $F(X, d)$, if d' is $(\text{cut}_A(F(X, d)), d)$ -outer-consistent then $\text{cut}_A(F(X, d)) = \text{cut}_A(F(X, d'))$.*

5. Main Result

The following is our characterization of linkage-based hierarchical algorithms.

Theorem 10 *A hierarchical function F is linkage-based if and only if F is outer consistent and local.*

We prove the result in the following subsections (one for each direction of the iff). In the last part of this section, we demonstrate the necessity of both properties.

5.1 All Local, Outer-Consistent Hierarchical Functions are Linkage-Based

Lemma 11 *If a hierarchical function F is outer-consistent and local, then F is linkage-based.*

We show that there exists a linkage function ℓ so that when ℓ is used in Definition 6 then for all (X, d) the output is $F(X, d)$. Due to the representation independence of F , one can assume w.l.o.g. that the domain sets over which F is defined are (finite) subsets of the set of natural numbers, \mathbb{N} .

Definition 12 (The (pseudo-) partial ordering $<_F$) *We consider triples of the form (A, B, d) , where $A \cap B = \emptyset$ and d is a distance function over $A \cup B$. Two triples, (A, B, d) and (A', B', d') are equivalent, denoted $(A, B, d) \cong (A', B', d')$ if they are isomorphic as clusterings, namely, if $\{(A, B), d\} \cong_C \{(A', B'), d'\}$.*

$<_F$ is a binary relation over equivalence classes of such triples, indicating that F merges a pair of clusters earlier than another pair of clusters. Formally, denoting \cong -equivalence classes by square brackets, we define it by: $[(A, B, d)] <_F [(A', B', d')]$ if

1. At most two sets in $\{A, B, A', B'\}$ are equal and no set is a strict subset of another.
2. The distance functions d and d' agree on $(A \cup B) \cap (A' \cup B')$.
3. There exists a distance function d^* over $X = A \cup B \cup A' \cup B'$ so that $F(X, d^*) = (T, M, \eta)$ such that
 - (a) d^* extends both d and d' ,

- (b) There exist $(x, y), (x, z) \in E(T)$ such that $C(x) = A \cup B$, $C(y) = A$, and $C(z) = B$
- (c) For all $D \in \{A', B'\}$, either $D \subseteq A \cup B$, or $D \in \text{cut}_{A \cup B} F(X, d^*)$.
- (d) $\eta(v(A')) < \eta(v(A \cup B))$ and $\eta(v(B')) < \eta(v(A \cup B))$.

Since we define hierarchical algorithms to be representation independent, we can just discuss triples, instead of their equivalence classes. For the sake of simplifying notation, we will omit the square brackets in the following discussion.

In the following lemma we show that if $(A, B, d) <_F (A', B', d')$, then $A' \cup B'$ cannot have a lower level than $A \cup B$.

Lemma 13 *Given a local and outer-consistent hierarchical function F , whenever*

- $(A_1, B_1, d_1) <_F (A_2, B_2, d_2)$, there is no data set (X, d) such that $A_1, B_1, A_2, B_2 \subseteq X$ and $\eta(v(A_2 \cup B_2)) \leq \eta(v(A_1 \cup B_1))$, where $F(X, d) = (T, M, \eta)$.

Proof By way of contradiction, assume that such (X, d) exists. Let $X' = A_1 \cup B_1 \cup A_2 \cup B_2$. Since $(A_1, B_1, d_1) <_F (A_2, B_2, d_2)$, there exists d' that satisfies the conditions of Definition 12.

Consider $F(X', d|X')$. By locality, the sub-dendrogram rooted at $v(A_1 \cup B_1)$ contains the same nodes in both $F(X', d|X')$ and $F(X, d)$, and similarly for the sub-dendrogram rooted at $v(A_2 \cup B_2)$. In addition, the relative level of nodes in these subtrees is the same.

Construct a distance function d^* over X' that is both $(\{A_1 \cup B_1, A_2 \cup B_2\}, d|X')$ -outer consistent and $(\{A_1 \cup B_2, A_2, B_2\}, d')$ -outer consistent as follows:

- $d^*(x, y) = \max(d(x, y), d'(x, y))$ whenever $x \in A_1 \cup B_1$ and $y \in A_2 \cup B_2$
- $d^*(x, y) = d_1(x, y)$ whenever $x, y \in A \cup B$
- $d^*(x, y) = d_2(x, y)$ whenever $x, y \in A' \cup B'$

Note that $\{A_1 \cup B_1, A_2 \cup B_2\}$ is an $(A_1 \cup B_1)$ -cut of $F(X', d|X')$. Therefore, by outer-consistency, $\text{cut}_{A_1 \cup B_1}(F(X', d^*)) = \{A_2 \cup B_2, A_1 \cup B_1\}$.

Since d' satisfies the conditions in Definition 12, $\text{cut}_{A_1 \cup B_1} F(X, d') = \{A_1 \cup B_1, A_2, B_2\}$. By outer-consistency we get that $\text{cut}_{A_1 \cup B_1}(F(X', d^*)) = \{A_2 \cup B_2, A_1, B_1\}$. Since these sets are all non-empty, this is a contradiction. ■

We now define equivalence with respect to $<_F$.

Definition 14 $[(\cong_F)]$ (A, B, d) and (A', B', d') are *F-equivalent*, denoted $[(A, B, d)] \cong_F [(A', B', d')]$, if either they are isomorphic as clusterings, $(\{A, B\}, d) \cong_C (\{A', B'\}, d')$ or

1. At most two sets in $\{A, B, A', B'\}$ are equal and no set is a strict subset of another.
2. The distance functions d and d' agree on $(A \cup B) \cap (A' \cup B')$.
3. There exists a distance function d^* over $X = A \cup B \cup A' \cup B'$ so that $F(A \cup B \cup A' \cup B', d^*) = (T, \eta)$ where
 - (a) d^* extends both d and d' ,

- (b) There exist $(x, y), (x, z) \in E(T)$ such that $C(x) = A \cup B$, and $C(y) = A$, and $C(z) = B$,
- (c) There exist $(x', y'), (x', z') \in E(T)$ such that $C(x') = A' \cup B'$, and $C(y') = A'$, and $C(z') = B'$, and
- (d) $\eta(x) = \eta(x')$

(A, B, d) is comparable with (C, D, d') if they are $<_F$ comparable or $(A, B, d) \cong_F (C, D, d')$.

Whenever two triples are F -equivalent, then they have the same $<_F$ or \cong_F relationship with all other triples.

Lemma 15 *Given a local, outer-consistent hierarchical function F , if $(A, B, d_1) \cong_F (C, D, d_2)$, then for any (E, F, d_3) , if (E, F, d_3) is comparable with both (A, B, d_1) and (C, D, d_2) then*

- if $(A, B, d_1) \cong_F (E, F, d_3)$ then $(C, D, d_2) \cong_F (E, F, d_3)$
- if $(A, B, d_1) <_F (E, F, d_3)$ then $(C, D, d_2) <_F (E, F, d_3)$

Proof Let $X = A \cup B \cup C \cup D \cup E \cup F$. By richness (condition 3 of Definition 4), there exists a distance function d that extends d_i for $i \in \{1, 2, 3\}$ so that $\{A \cup B, C \cup D, E \cup F\}$ is a clustering in $F(X, d)$. Assume that (E, F, d_3) is comparable with both (A, B, d_1) and (C, D, d_2) . By way of contradiction, assume that $(A, B, d_1) \cong_F (E, F, d_3)$ and $(C, D, d_2) <_F (E, F, d_3)$. Then by locality, in $F(X, d)$, $\eta(v(A \cup B)) = \eta(v(E \cup F))$.

Observe that by locality, since $(C, D, d_1) <_F (E, F, d_3)$, then $\eta(v(C \cup D)) < \eta(v(E \cup F))$ in $F(X, d)$. Therefore (again by locality) $\eta(v(A \cup B)) \neq \eta(v(C \cup D))$ in any data set that extends d_1 and d_2 , contradicting that $(A, B, d_1) \cong_F (C, D, d_2)$. ■

Note that $<_F$ is not transitive. In particular, if $(A, B, d_1) <_F (C, D, d_2)$ and $(C, D, d_2) <_F (E, F, d_3)$, it may be that (A, B, d_1) and (E, F, d_3) are incomparable. To show that $<_F$ can be extended to a partial ordering, we first prove the following ‘‘anti-cycle’’ property.

Lemma 16 *Given a hierarchical function F that is local and outer-consistent, there exists no finite sequence $(A_1, B_1, d_1) <_F \dots <_F (A_n, B_n, d_n) <_F (A_1, B_1, d_1)$.*

Proof Without loss of generality, assume that such a sequence exists. By richness, there exists a distance function d that extends each of the d_i where $\{A_1 \cup B_1, A_1 \cup B_2, \dots, A_n \cup B_n\}$ is a clustering in $F(\bigcup_i A_i \cup B_i, d) = (T, M, \eta)$.

Let i_0 be so that $\eta(v(A_{i_0} \cup B_{i_0})) \leq \eta(v(A_j \cup B_j))$ for all $j \neq i_0$. By the circular structure with respect to $<_F$, there exists j_0 so that $(A_{j_0}, B_{j_0}, d_{j_0}) <_F (A_{i_0}, B_{i_0}, d_{i_0})$. This contradicts Lemma 13. ■

We make use of the following general result.

Lemma 17 *For any cycle-free, anti-symmetric relation $P(\cdot, \cdot)$ over a finite or countable domain D there exists an embedding h into \mathbb{R}^+ so that for all $x, y \in D$, if $P(x, y)$ then $h(x) < h(y)$.*

Proof First we convert the relation P into a partial order by defining $a < b$ whenever there exists a sequence x_1, \dots, x_k so that $P(a, x_1), P(x_2, x_3), \dots, P(x_k, b)$. This is a partial ordering because P is antisymmetric and cycle-free. To map the partial order to the positive reals, we first enumerate the elements, which can be done because the domain is countable. The first element is then mapped to any value, $\phi(x_1)$. By induction, we assume that the first n elements are mapped in an order preserving manner. Let x_{r_1}, \dots, x_{r_k} be all the members of $\{x_1, \dots, x_n\}$ that are below x_{n+1} in the partial order. Let $r_1 = \max\{\phi(x_{r_1}), \dots, \phi(x_{r_k})\}$, and similarly let r_2 be the minimum among the images of all the members of $\{x_1, \dots, x_k\}$ that are above x_{n+1} in the partial order. Finally, let $\phi(x_{n+1})$ be any real number between r_1 and r_2 . It is easy to see that now ϕ maps $\{x_1, \dots, x_n, x_{n+1}\}$ in a way that respects the partial order. ■

Finally, we define our linkage function by embedding the \cong_F -equivalence classes into the positive real numbers in an order preserving way, as implied by applying Lemma 17 to $<_F$. Namely, $\ell_F : \{[A, B, d]\} : A \subseteq \mathbb{N}, B \subseteq \mathbb{N}, A \cap B = \emptyset$ and d is a distance function over $A \cup B \rightarrow \mathbb{R}^+$ so that $[A, B, d] <_F [A', B', d']$ implies $\ell_F([A, B, d]) < \ell_F([A', B', d'])$.

Lemma 18 *The function ℓ_F is a linkage function for any hierarchical function F that satisfies locality and outer-consistency.*

Proof Since ℓ_F is defined on \cong_F -equivalence classes, representation independence of hierarchical functions implies that ℓ_F satisfies condition 1 of Definition 5. The function ℓ_F satisfies condition 2 of Definition 5 by Lemma 19, whose proof follows. ■

Lemma 19 *Consider d_1 over $X_1 \cup X_2$ and d_2 that is $(\{X_1, X_2\}, d_1)$ -outer-consistent, then $(X_1, X_2, d_2) \not\prec_F (X_1, X_2, d_1)$, whenever F is local and outer-consistent.*

Proof Assume that there exist such d_1 and d_2 where $(X_1, X_2, d_2) <_F (X_1, X_2, d_1)$. Let d_3 over $X_1 \cup X_2$ be a distance function such that d_3 is $(\{X_1, X_2\}, d_1)$ -outer-consistent and d_2 is $(\{X_1, X_2\}, d_3)$ -outer-consistent. In particular, d_3 can be constructed as follows:

- $d_3(x, y) = \frac{d_1(x, y) + d_2(x, y)}{2}$ whenever $x \in X_1$ and $y \in X_2$
- $d_3(x, y) = d_1(x, y)$ whenever $x, y \in X_1$ or $x, y \in X_2$

Set $(X_1, X_2', d_2) \cong_F (X_1, X_2, d_2)$ and $(X_1'', X_2'', d_3) \cong_F (X_1, X_2, d_3)$.

Let $X = X_1 \cup X_2 \cup X_1' \cup X_2' \cup X_1'' \cup X_2''$. By richness, there exists a distance function d^* that extends d_i for all $1 \leq i \leq 3$ so that $\{X_1 \cup X_2, X_1' \cup X_2', X_1'' \cup X_2''\}$ is a clustering in $F(X, d^*)$.

Let $F(X, d^*) = (T, M, \eta)$. Since $(X_1', X_2', d_2) <_F (X_1, X_2, d_1)$, by locality and outer-consistency, we get that $\eta(v(X_1' \cup X_2')) < \eta(v(X_1 \cup X_2))$. We consider the level $(\eta$ value) of $v(X_1' \cup X_2')$ with respect to the levels of $v(X_1' \cup X_2')$ and $v(X_1 \cup X_2)$ in $F(X, d^*)$.

We now consider a few cases.

Case 1: $\eta(v(X_1' \cup X_2')) \leq \eta(v(X_1' \cup X_2'))$. Then there exists an outer-consistent change moving X_1 and X_2 further away from each other until $(X_1, X_2, d_1) = (X_1'', X_2'', d_3)$. Let d be the distance function that extends d_1 and d_2 which shows that $(X_1', X_2', d_2) <_F (X_1, X_2, d_1)$.

$cut_{X_1' \cup X_2'} F(X_1 \cup X_2 \cup X_1' \cup X_2', d) = \{X_1' \cup X_2', X_1, X_2\}$. We can apply outer consistency on $\{X_1' \cup X_2', X_1, X_2\}$ and move X_1 and X_2 away from each other until $\{X_1, X_2\}$ is isomorphic to $\{X_1'', X_2''\}$. By outer consistency, this modification should not affect the $(X_1 \cup X_2)$ -cut. Applying locality, we have two isomorphic data sets that produce different dendrograms, one in which the further pair $(\{X_1', X_2'\}$ with distance function d_2) is not below the medium pair $(\{X_1'', X_2''\}$ with distance function d_3), and the other in which the medium pair is above the furthest pair.

Case 2: $\eta(v(X_1'' \cup X_2'')) \geq \eta(v(X_1 \cup X_2))$. Since X_2'' is isomorphic to X_2 for all $i \in \{1, 2\}$, $\eta(v(X_2)) = \eta(v(X_2''))$ for all $i \in \{1, 2\}$. This gives us that in this case, $cut_{X_1 \cup X_2} F(X_1 \cup X_2 \cup X_1'' \cup X_2'') = \{X_1 \cup X_2, X_1'', X_2''\}$. We can therefore apply outer consistency and separate X_1'' and X_2'' until $\{X_1'', X_2''\}$ is isomorphic to $\{X_1' \cup X_2'\}$. So this gives us two isomorphic data sets, one in which the further pair is not below the closest pair, and the other in which the further pair is below the closest pair.

Case 3: $\eta(X_1 \cup X_2) < \eta(X_1'' \cup X_2'') < \eta(X_1' \cup X_2')$. Notice that $cut_{X_1' \cup X_2'} F(X_1 \cup X_2 \cup X_1' \cup X_2', d^*) = \{X_1' \cup X_2', X_1, X_2\}$. So outer-consistency applies when we increase the distance between X_1 and X_2 until $\{X_1, X_2\}$ is isomorphic to $\{X_1' \cup X_2'\}$. This gives us two isomorphic sets, one in which the medium pair is below the further pair, and another in which the medium pair is above the furthest pair. ■

The following Lemma concludes the proof that every local, outer-consistent hierarchical algorithm is linkage-based.

Lemma 20 *Given any hierarchical function F that satisfies locality and outer-consistency, let ℓ_F be the linkage function defined above. Let L_{ℓ_F} denote the linkage-based algorithm that ℓ_F defines. Then L_{ℓ_F} agrees with F on every input data set.*

Proof Let (X, d) be any data set. We prove that at every level s , the nodes at level s in $F(X, d)$ represent the same clusters as the nodes at level s in $L_{\ell_F}(X, d)$. In both $F(X, d) = (T, M, \eta)$ and $L_{\ell_F}(X, d) = (T', M', \eta')$, level 0 consists of $|X|$ nodes each representing a unique elements of X .

Assume the result holds below level k . We show that pairs of nodes that do not have parents below level k have minimal ℓ_F value only if they are merged at level k in $F(X, d)$. Consider $F(X, d)$ at level k . Since the dendrogram has no empty levels, let $x \in V(T)$ where $\eta(x) = k$. Let x_1 and x_2 be the children of x in $F(X, d)$. Since $\eta(x_1), \eta(x_2) < k$, these nodes also appear in $L_{\ell_F}(X, d)$ below level k , and neither node has a parent below level k .

If x is the only node in $F(X, d)$ above level $k - 1$, then it must also occur in $L_{\ell_F}(X, d)$. Otherwise, there exists a node $y_1 \in V(T)$, $y_1 \notin \{x_1, x_2\}$ so that $\eta(y_1) < k$ and $\eta(\text{parent}(y_1)) \geq k$. Let $X' = \mathcal{C}(x) \cup \mathcal{C}(y_1)$. By locality, $cut_{\mathcal{C}(x)} F(X', d|X') = \{\mathcal{C}(x), \mathcal{C}(y_1)\}$, y_1 is below x , and x_1 and x_2 are the children of x . Therefore, $(\mathcal{C}(x_1), \mathcal{C}(x_2), d) <_F (\mathcal{C}(x_1), \mathcal{C}(y_1), d)$ and $\ell_F(\mathcal{C}(x_1), \mathcal{C}(x_2), d) < \ell_F(\mathcal{C}(x_1), \mathcal{C}(y_1), d)$.

Assume that there exists $y_2 \in V(T)$, $y_2 \notin \{x_1, x_2, y_1\}$ so that $\eta(y_2) < k$ and $\eta(\text{parent}(y_2)) \geq k$. If $\text{parent}(y_2) = \text{parent}(y_2)$ and $\eta(\text{parent}(y_1)) = k$, then $(\mathcal{C}(x_1), \mathcal{C}(x_2), d) \cong_F (\mathcal{C}(y_1), \mathcal{C}(y_2), d)$ and so $\ell_F(\mathcal{C}(x_1), \mathcal{C}(x_2), d) = \ell_F(\mathcal{C}(y_1), \mathcal{C}(y_2), d)$.

Otherwise, let $X' = C(x) \cup C(y_1) \cup C(y_2)$. By richness, there exists a distance function d^* that extends $d(C(x))$ and $d(C(y_1) \cup C(y_2))$, so that $\{C(x), C(y_1) \cup C(y_2)\}$ is in $F(X', d^*)$. Note that by locality, the node $v(C(y_1) \cup C(y_2))$ has children $v(C(y_1))$ and $v(C(y_2))$ in $F(X', d^*)$. We can separate $C(x)$ from $C(y_1) \cup C(y_2)$ in both $F(X', d^*)$ and $F(X', d|X')$ until both are equal. Then by outer-consistency, $\text{cut}_{C(x)} F(X', d|X') = \{C(x), C(y_1), C(y_2)\}$ and by locality y_1 and y_2 are below x . Therefore, $(C(x_1), C(x_2), d) <_F (C(y_1), C(y_2), d)$ and so $\ell_F(C(x_1), C(x_2), d) < \ell_F(C(y_1), C(y_2), d)$. ■

5.2 All Linkage-Based Functions are Local and Outer-Consistent

Lemma 21 *Every linkage-based hierarchical clustering function is local.*

Proof Let $C = \{C_1, C_2, \dots, C_k\}$ be a clustering in $F(X, d) = (T, M, \eta)$. Let $X' = \cup_i C_i$. For all $X_1, X_2 \in X'$, $\ell(X_1, X_2, d) = \ell(X_1, X_2, d|X')$. Therefore, for all $1 \leq i \leq k$, the sub-dendrogram rooted at $v(C_i)$ in $F(X, d)$ also appears in $F(X, d')$, with the same relative levels. ■

Lemma 22 *Every linkage-based hierarchical clustering function is outer-consistent.*

Proof Let $C = \{C_1, C_2, \dots, C_k\}$ be a C_i -cut in $F(X, d)$ for some $1 \leq i \leq k$. Let d' be (C, d) -outer-consistent. Then for all $1 \leq i \leq k$, and all $X_1, X_2 \subseteq C_i$, $\ell(X_1, X_2, d) = \ell(X_1, X_2, d')$, while for all $X_1 \subseteq C_i, X_2 \subseteq C_j$, for any $i \neq j$, $\ell(X_1, X_2, d) \leq \ell(X_1, X_2, d')$ by monotonicity. Therefore, for all $1 \leq j \leq k$, the sub-dendrogram rooted at $v(C_j)$ in $F(X, d)$ also appears in $F(X, d')$. All nodes added after these sub-dendrograms are at a higher level than the level of $v(C_i)$. And since the C_i -cut is represented by nodes that occur on levels no higher than the level of $v(C_i)$, the C_i -cut in $F(X, d')$ is the same as the C_i -cut in $F(X, d)$. ■

5.3 Necessity of Both Properties

We now show that both the locality and outer-consistency properties are necessary for defining linkage-based algorithms. Neither property individually is sufficient for defining this family of algorithms. Our results above showing that all linkage-based algorithms are both local and outer-consistent already imply that a clustering function that satisfies one, but not both, of these requirements is not linkage-based. It remains to show that neither of these two properties implies the other. We do so by demonstrating the existence of a hierarchical function that satisfies locality but not outer-consistency, and one that satisfies outer-consistency but not locality.

Consider a hierarchical clustering function F that applies average-linkage on data sets with an even number of elements, and single-linkage on data sets consisting of an odd number of elements. Since both average-linkage and single-linkage are linkage-based algorithms, they are both outer-consistent. It follows that F is outer-consistent. However, this hierarchical clustering function fails locality, as it is easy to construct a data set with an even number of

elements where average-linkage detects an odd-sized cluster, for which single-linkage would produce a different dendrogram.

Now, consider the following function

$$\ell(X_1, X_2, d) = \frac{1}{\max_{x \in X_1, y \in X_2} d(x, y)}.$$

The function ℓ is not a linkage-function since it fails the monotonicity condition. The function ℓ also does not conform with the intended meaning of a linkage-function. For instance, $\ell(X_1, X_2, d)$ is smaller than $\ell(X'_1, X'_2, d')$ when *all* the distances between X_1 and X_2 are (arbitrarily) larger than any distance between X'_1 and X'_2 . If we then consider the hierarchical clustering function F that results by utilizing ℓ in a greedy fashion to construct a dendrogram (by repeatedly merging the closest clusters according to ℓ), then the function F is local by the same argument as the proof of Lemma 21. We now demonstrate that F is not outer-consistent. Consider a data set (X, d) such that for some $A \subset X$, the A -cut of $F(X, d)$ is a clustering with at least 3 clusters where every cluster consists of at least 2 elements. Then if we move two clusters sufficiently far away from each other and all other data, they will be merged by the algorithm before any of the other clusters are formed, and so the A -cut on the resulting data changes following an outer-consistent change. As such, F is not outer-consistent.

6. Divisive Algorithms

Our formalism provides a precise sense in which linkage-based algorithms make only local considerations, while many divisive algorithms inevitably take more global considerations into account. This fundamental distinction between these paradigms can be used to help select a suitable hierarchical algorithm for specific applications.

This distinction also implies that many divisive algorithms cannot be simulated by any linkage-based algorithm, showing that the class of hierarchical algorithms is strictly richer than the class of linkage-based algorithm (even when focusing only on the input-output behaviour of algorithms).

A 2-clustering function \mathcal{F} maps a data set (X, d) to a 2-partition of X . An \mathcal{F} -Divisive algorithm is a divisive algorithm that uses a 2-clustering function \mathcal{F} to decide how to split nodes. Formally,

Definition 23 (\mathcal{F} -Divisive) *A hierarchical clustering function is \mathcal{F} -Divisive with respect to a 2-clustering function \mathcal{F} , if for all (X, d) , $\mathcal{F}(X, d) = (T, M, \eta)$ such that for all $x \in V(T)/\text{leaves}(T)$ with children x_1 and x_2 , $\mathcal{F}(C(x)) = \{C(x_1), C(x_2)\}$.*

Note that Definition 23 does not place restrictions on the level function. This allows for some flexibility in the levels. Intuitively, it doesn't force an order on splitting nodes.

The following property represents clustering functions that utilize contextual information found in the remainder of the data set when partitioning a subset of the domain.

Definition 24 (Context sensitive) \mathcal{F} is context-sensitive if there exist x, y, z, w and distance functions d and d' , where d' extends d , such that $\mathcal{F}(\{x, y, z\}, d) = \{\{x\}, \{y, z\}\}$ and $\mathcal{F}(\{x, y, z, w\}, d') = \{\{x, y\}, \{z, w\}\}$.

Many 2-clustering functions, including k -means, min-sum, and min-diameter are context-sensitive (see Corollary 29, below). Natural divisive algorithms, such as bisecting k -means (k -means-Divisive), rely on context-sensitive 2-clustering functions.

Whenever a 2-clustering algorithm is context-sensitive, then the \mathcal{F} -divisive function is not local.

Theorem 25 *If \mathcal{F} is context-sensitive then the \mathcal{F} -divisive function is not local.*

Proof

Since \mathcal{F} is context-sensitive, there exists a distance functions $d \subset d'$ so that $\{x\}$ and $\{y, z\}$ are the children of the root in $\mathcal{F}(\{x, y, z\}, d)$, while in $\mathcal{F}(\{x, y, z, w\}, d')$, $\{x, y\}$ and $\{z, w\}$ are the children of the root and z and w are the children of $\{z, w\}$. Therefore, $\{\{x, y\}, \{z\}\}$ is clustering in $\mathcal{F}(\{x, y, z, w\}, d)$. But cluster $\{x, y\}$ is not in $\mathcal{F}(\{x, y, z\}, d)$, so the clustering $\{\{x, y\}, \{z\}\}$ is not in $\mathcal{F}(\{x, y, z\}, d)$, and so \mathcal{F} -divisive is not local. ■

Applying Theorem 10, we get:

Corollary 26 *If \mathcal{F} is context-sensitive, then the \mathcal{F} -divisive function is not linkage-based.*

We say that two hierarchical algorithms disagree if they may output dendrograms with different clusterings. Formally,

Definition 27 *Two hierarchical functions F_0 and F_1 disagree if there exists a data set (X, d) and a clustering C of X so that C is in $F_0(X, d)$ but not in $F_1(X, d)$, for some $i \in \{0, 1\}$.*

Theorem 28 *If \mathcal{F} is context-sensitive, then the \mathcal{F} -divisive function disagrees with every linkage-based function.*

Proof Let L be any linkage-based function. Since \mathcal{F} is context-sensitive, there exists distance functions $d \subset d'$ so that $\mathcal{F}(\{x, y, z\}, d) = \{\{x\}, \{y, z\}\}$ and $\mathcal{F}(\{x, y, z, w\}, d') = \{\{x, y\}, \{z, w\}\}$.

Assume that L and \mathcal{F} -divisive produce the same output on $(\{x, y, z, w\}, d')$. Therefore, since $\{\{x, y\}, \{z\}\}$ is a clustering in \mathcal{F} -divisive($\{x, y, z, w\}, d'$), it is also a clustering in $L(\{x, y, z, w\}, d')$. Since L is linkage-based, by Theorem 10, L is local. Therefore, $\{\{x, y\}, \{z\}\}$ is a clustering in $L(\{x, y, z\}, d)$. But it is not a clustering in \mathcal{F} -divisive($\{x, y, z\}, d$). ■

Corollary 29 *The divisive algorithms that are based on the following 2-clustering functions disagree with every linkage-based function: k -means, min-sum, min-diameter.*

Proof Set $x = 1$, $y = 3$, $z = 4$, and $w = 6$ to show that these 2-clustering functions are context-sensitive. The result follows by Theorem 28. ■

7. Conclusions

In this paper, we provide the first property-based characterization of hierarchical linkage-based clustering. Our characterization shows the existence of hierarchical methods that cannot be simulated by any linkage-based method, revealing inherent input-output differences between agglomeration and divisive hierarchical algorithms.

This work falls in the larger framework of property-based analysis of clustering algorithms, which aims to provide a better understanding of these techniques as well as aid users in the crucial task of algorithm selection. It is important to note that our characterization is not intended to demonstrate the superiority of linkage-based methods over other hierarchical techniques, but rather to enable users to make informed trade-offs when choosing algorithms. In particular, properties investigated in previous work should also be considered, while future work will continue to investigate important properties with the ultimate goal of providing users with a property-based taxonomy of popular clustering methods that would enable selecting suitable methods for a wide range of applications.

8. Acknowledgements

We would like to thank David Loker for several helpful discussions. We would also like to thank the anonymous referees whose comments and suggestions greatly improved this paper.

References

- M. Ackerman and S. Ben-David. Measures of clustering quality: A working set of axioms for clustering. In *Proceedings of Neural Information Processing Systems (NIPS)*, pages 121–128, 2008.
- M. Ackerman, S. Ben-David, and D. Loker. Characterization of linkage-based clustering. In *Proceedings of The 23rd Conference on Learning Theory*, pages 270–281, 2010a.
- M. Ackerman, S. Ben-David, and D. Loker. Towards property-based classification of clustering paradigms. *Lafferty et al.*, pages 10–18, 2010b.
- M. Ackerman, S. Ben-David, S. Branzai, and D. Loker. Weighted clustering. In *Association for the Advancement of Artificial Intelligence (AAAI)*, pages 858–863, 2012.
- G. Carlsson and F. Mémoli. Characterization, stability and convergence of hierarchical clustering methods. *The Journal of Machine Learning Research*, 11:1425–1470, 2010.
- L. Fisher and J.W. Van Ness. Admissible clustering procedures. *Biometrika*, 58(1):91–104, 1971.
- N. Jardine and R. Sibson. Mathematical taxonomy. *London*, 1971.
- J. Kleinberg. An impossibility theorem for clustering. *Proceedings of International Conferences on Advances in Neural Information Processing Systems*, pages 463–470, 2003.

- M. Meila. Comparing clusterings: an axiomatic view. In *Proceedings of the 22nd international conference on Machine learning*, pages 577–584. ACM, 2005.
- J. Puzicha, T. Hofmann, and J.M. Buhmann. A theory of proximity based clustering: Structure detection by optimization. *Pattern Recognition*, 33(4):617–634, 2000.
- W.E. Wright. A formalization of cluster analysis. *Pattern Recognition*, 5(3):273–282, 1973.
- R.B. Zadeh and S. Ben-David. A uniqueness theorem for clustering. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 639–646. AUAI Press, 2009.

Learning Latent Variable Models by Pairwise Cluster Comparison: Part II – Algorithm and Evaluation

Nuaman Asbeh

*Department of Industrial Engineering and Management
Ben-Gurion University of the Negev
Beer Sheva, 84105, Israel*

ASBEH@POST.BGU.AC.IL

Boaz Lerner

*Department of Industrial Engineering and Management
Ben-Gurion University of the Negev
Beer Sheva, 84105, Israel*

BOAZ@BGU.AC.IL

Editors: Isabelle Guyon and Alexander Statnikov

Abstract

It is important for causal discovery to identify any latent variables that govern a problem and the relationships among them, given measurements in the observed world. In Part I of this paper, we were interested in learning a discrete latent variable model (LVM) and introduced the concept of *pairwise cluster comparison* (PCC) to identify causal relationships from clusters of data points and an overview of a two-stage algorithm for *learning PCC* (LPCC). First, LPCC learns exogenous latent variables and latent colliders, as well as their observed descendants, by using pairwise comparisons between data clusters in the measurement space that may explain latent causes. Second, LPCC identifies endogenous latent non-colliders with their observed children. In Part I, we showed that if the true graph has no serial connections, then LPCC returns the true graph, and if the true graph has a serial connection, then LPCC returns a pattern of the true graph. In this paper (Part II), we formally introduce the LPCC algorithm that implements the PCC concept. In addition, we thoroughly evaluate LPCC using simulated and real-world data sets in comparison to state-of-the-art algorithms. Besides using three real-world data sets, which have already been tested in learning an LVM, we also evaluate the algorithms using data sets that represent two original problems. The first problem is identifying young drivers' involvement in road accidents, and the second is identifying cellular subpopulations of the immune system from mass cytometry. The results of our evaluation show that LPCC improves in accuracy with the sample size, can learn large LVMs, and is accurate in learning compared to state-of-the-art algorithms. The code for the LPCC algorithm and data sets used in the experiments reported here are available online.

Keywords: learning latent variable models, graphical models, clustering, pure measurement model

1. Introduction

We began Part I by describing the task of learning a latent variable model (LVM). We dispensed with the linearity assumption (for a child given its parents) and concentrated on the discrete case. In addition, we did not limit our analysis to learning latent-tree mod-

els and focused on multiple indicator models (MIMs) that are a very important subclass of structural equation models (SEM) – models that are widely used in applied and social sciences to analyze causal relations. By borrowing ideas from unsupervised learning, we could introduce the notion of pairwise cluster comparison (PCC). PCC compares pairwise clusters of data points representing instantiations of the observed variables to identify those pairs of clusters that exhibit changes in the observed variables due to changes in their ancestor latent variables. Changes in a latent variable that are manifested in changes in its descendant observed variables reveal this latent variable and its causal paths of influence in the domain. Learning PCC (LPCC) was introduced as a tool to transform data clusters into knowledge about latent variables – their number, types, cardinalities, and interrelations among themselves and with the observed variables – that is needed to learn an LVM.

Part I provided preliminaries and the theoretical support of LPCC. Several definitions and theorems that were already introduced also play an important role in Part II. To ease reading Part II, on the one hand, and to supply the necessary theoretical background, on the other hand, we have summarized these definitions, propositions, and theorems from Part I here in Appendix A. Following is a brief summary of the PCC concept and LPCC algorithm; the full details appear in Part I.

First in the LPCC algorithm is clustering of data that are sampled from the observed variables in the unknown model. Clustering in the current implementation is based on the self-organizing map (SOM) algorithm (Kohonen, 1997), although any other clustering algorithm that does not need a preliminary determination of the number of clusters may be suitable.¹ Second, LPCC selects an initial set of major clusters (Section 4.3 of Part I; Definition 12 in Appendix A²). Third, LPCC learns an LVM in two stages. In the first stage (Section 4.1 of Part I), LPCC analyzes PCCs³ (Definition 15) between two major clusters to find maximal sets of observed (**MSO** by Definition 16) variables that always change together. By Theorem 1, variables of a particular **MSO** are children of a particular exogenous latent variable or its latent non-collider descendant or children of a particular latent collider. This stage allows the identification of exogenous latent variables and latent colliders together and their corresponding observed descendants. Then (Section 4.2 of Part I), LPCC distinguishes the latent colliders from the exogenous latent variables using Theorem 2. To complete this stage, LPCC iteratively improves the selection of the major clusters (Section 4.3 of Part I), and the entire stage is repeated until convergence. In the second stage, LPCC identifies endogenous latent non-colliders with their children (Section 4.4 of Part I). Because distinguishing endogenous latent non-colliders from their exogenous ancestors could not be performed using major-major PCCs, in this stage LPCC needs to apply a mechanism to split these two types of latent variables from each other and then direct them using comparison of major clusters to (a special type of) minor clusters (2S-MC; Definition 14) that correspond to 2-order minor effects (Definition 13). For this task, LPCC

¹See for example Section 3.6, where we replaced SOM with hierarchical clustering.

²The definitions and theorems that are mentioned here are borrowed from Part I and are summarized in Appendix A.

³PCC is a procedure by which pairs of clusters are compared through a comparison of their centroids, and the result can be represented by a binary vector in which each element is 1 or 0 depending, respectively, on whether or not there is a difference between the corresponding elements in the compared clusters.

analyzes 2S-PCCs (Definition 18), which are PCCs between major and minor clusters that show two sets (this is the source of “2S” in the name 2S-PCC) of two or more elements in the PCC, and identifies 2S-MSOs (Definition 19), which are maximal sets of observed variables that always change their values together in all 2S-PCCs. Different 2S-MSOs due to an exogenous latent variable represent latent non-colliders that are descendants of this exogenous variable; hence, LPCC can distinguish between the two types of variables by analyzing 2S-MSOs (Theorem 3). To direct the edges between latent non-colliders on a path emerging in an exogenous latent, LPCC checks changes of several 2S-PCCs with respect to changes of the latent non-colliders’ exogenous ancestor. Theorem 4 guarantees that LPCC finds all diverging connections and represents all serial connections using a pattern of the true graph, which completes learning the LVM. A flowchart of the LPCC algorithm is given in Figure 1.

A main section of Part II is a formal description of the two-stage LPCC algorithm, which is founded on the PCC concept. Part II also provides an experimental evaluation of LPCC, in comparison to state-of-the-art algorithms, using simulated data sets (Section 3.1) and real-world data sets (Sections 3.2–3.6). The outline of the paper is as follows:

- **Section 2: The LPCC algorithm** introduces and formally describes a two-stage algorithm that implements the PCC concept;
- **Section 3: LPCC evaluation** evaluates LPCC, in comparison to state-of-the-art algorithms, using simulated data sets (Section 3.1) and real-world data sets (Sections 3.2–3.6);
- **Section 4: Related works** compares LPCC to state-of-the-art LVM learning algorithms;
- **Section 5: Discussion** summarizes the theoretical advantages (from Part I) and the practical benefits (from this part) of using LPCC;
- **Appendix A** provides essential assumptions, definitions, propositions, and theorems from Part I;
- **Appendix B** supplies additional results for the experiments with the simulated data sets (Section 3.1); and
- **Appendix C** provides PCC analysis for two example databases.

2. The LPCC algorithm

We introduced a two-stage algorithm, LPCC, that implements the PCC concept (Part I). The algorithm gets a data set \mathbf{D} over the observed variables \mathbf{O} and learns an LVM. In the first stage, LPCC learns the exogenous variables and the latent colliders as well as their descendants using the LEXC algorithm (Section 2.1). In the second stage, LPCC augments the graph learned by LEXC by learning the endogenous latent non-colliders and their children using the LNC algorithm (Section 2.2).

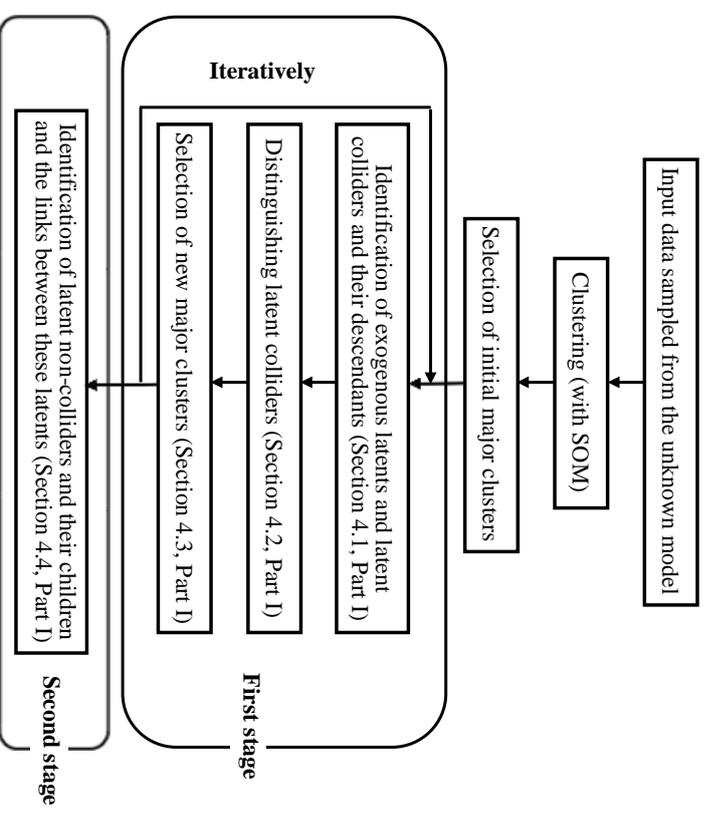


Figure 1: An overview of the LPCC algorithm as described in Part I.

2.1 Learning exogenous and latent colliders (LEXC)

LEXC (Algorithm 1) adapts an iterative approach and learns the initial graph in six steps. The first step is clustering \mathbf{D} using the self-organizing map (SOM) (Kohonen, 1997). We chose SOM because it does not require prior knowledge about the expected number of clusters, which is essential when targeting uncertainty in the number of latent variables in the model, but any other clustering algorithm that preserves this property can replace SOM. The result of the first step is a cluster set \mathbf{c} in which each cluster c is represented by its centroid.

In the second step, LEXC performs an initial selection of the major clusters set (Definition 12 in Appendix A), where a cluster in \mathbf{c} whose size (measured by the number of clustered patterns) is larger than the average cluster size in \mathbf{c} is selected as a major cluster (Section 4.3 of Part I). $\mathbf{MC} = \{\mathbf{MC}_j\}_{j=1}^n$ is a matrix that holds information about the major clusters, where each matrix row represents a centroid of one of the n major clusters (see, e.g., Table 2 in Part I).

In the third step, LEXC creates a matrix that represents all PCCs (Definition 15), derived from \mathbf{MC} . This matrix is $\mathbf{PCCM} = \{\mathbf{PCC}_{ij}\}_{i=1, j>1}^{n,n}$ where \mathbf{PCC}_{ij} is a Boolean vector representing the result of PCC between major clusters c_i and c_j having centroids \mathbf{MC}_i and \mathbf{MC}_j in \mathbf{MC} , respectively (see, e.g., Table 4 in Part I). The k -th element of \mathbf{PCC}_{ij} represents by “1” a change in value, if one exists, in the observed variable $O_k \in \mathbf{O}$ when comparing \mathbf{MC}_i and \mathbf{MC}_j (for example, Table 4 in Part I shows a change in element 7, corresponding to X_7 , of $\mathbf{PCC}_{2,9}$ between C2 and C9). We use the notation $\mathbf{PCC}_{ij} \rightarrow \delta O_k$ if the value of O_k has been changed and $\mathbf{PCC}_{ij} \rightarrow \neg\delta O_k$ otherwise.

In the fourth step, LEXC identifies exogenous latents and their descendants (Theorem 1) using a matrix \mathbf{MSOS} that holds all \mathbf{MSOs} (Definition 16) that always change their corresponding values together in all major-major PCCs in \mathbf{PCCM} . For each identified \mathbf{MSO}_i , LEXC adds a latent L_i to \mathbf{G} and to a latent set \mathbf{L} and also edges from L_i to each observed variable $O \in \mathbf{MSO}_i$. The observed children of latent $L_i \in \mathbf{L}$ in \mathbf{G} are \mathbf{Ch}_i .

In the fifth step, LEXC identifies in two phases, each corresponding to one condition in Theorem 2, the latent variables that are collider nodes in the graph along with their latent ancestors. In the first phase, LEXC considers for each latent variable $L_i \in \mathbf{L}$, a set of potential ancestors from the other latents in \mathbf{L} . We call them potential ancestors because another condition should be fulfilled in the second phase to turn them into actual ancestors. To simplify the notation, we represent the latent as an object and the set of potential ancestors as a field of this object, called \mathbf{PAS} (for potential ancestor set). For example, $L_i.\mathbf{PAS}$ represents that LEXC identifies a potential ancestor set \mathbf{PAS} to latent L_i . In addition, we use the notation $\mathbf{PCC}_{fg} \rightarrow \delta L_i$ if all of the variables in \mathbf{Ch}_i change their values in $\mathbf{PCC}_{fg} \in \mathbf{PCCM}$ and $\mathbf{PCC}_{fg} \rightarrow \neg\delta L_i$ otherwise. In the first phase of the fifth step, LEXC checks for each $L_j \in \mathbf{L}$ whether there exists a vector $\mathbf{PCC}_{fg} \in \mathbf{PCCM}$ in which L_j changes value together with $L_j \in \mathbf{L}$, but not with $L_k \in \mathbf{L}, \forall k \neq i, j$, and if so, it adds L_j to $L_i.\mathbf{PAS}$. At the end of this phase, the set $L_i.\mathbf{PAS}$ contains all of the latents in \mathbf{L} that change values with L_i in \mathbf{PCCM} . Still, this is not enough to decide that L_j is a collider of the variables in $L_i.\mathbf{PAS}$. An additional condition must be fulfilled, which is that L_j should never have changed in any $\mathbf{PCC}_{fg} \in \mathbf{PCCM}$ unless at least one of the variables in $L_i.\mathbf{PAS}$ has also changed in this \mathbf{PCC}_{fg} (Section 4.2 of Part I). The second phase of the fifth step checks this condition, and

Algorithm 1 LEXC

```

1: Input: Data set  $\mathbf{D}$  over the observed variables  $\mathbf{O}$ 
2: Output: Graph  $\mathbf{G}$  that includes the exogenous latent variables and the latent colliders and their descendants in LVM
3: Initialize:
4: Create an empty graph  $\mathbf{G}$  over the observed variables  $\mathbf{O}$ 
5:  $\mathbf{c} = \phi, \mathbf{MC} = \mathbf{0}, \mathbf{PCCM} = \mathbf{0}, \mathbf{L} = \phi, \mathbf{MSOS} = \phi$ 
6: % First step: perform clustering
7:  $\mathbf{c} \leftarrow$  perform clustering on  $\mathbf{D}$  and represent each cluster by its centroid
8: % Second step: select an initial set of major clusters
9: For each  $c_i \in \mathbf{c}$ 
10:   If the size of  $c_i$  is larger than the average cluster size in  $\mathbf{c}$ , then add  $c_i$  to  $\mathbf{MC}$ .
11: % Third step: create the  $\mathbf{PCCM}$  matrix
12: For each  $\mathbf{MC}_i, \mathbf{MC}_j \in \mathbf{MC}, j > i$ 
13:    $\mathbf{PCCM} \leftarrow$  compute  $\mathbf{PCC}_{ij}$ 
14: % Fourth step: identify latent variables and their observed children
15:  $\mathbf{MSOS} \leftarrow$  find all possible  $\mathbf{MSOs}$  using  $\mathbf{PCCM}$ 
16: For each  $\mathbf{MSO}_i \in \mathbf{MSOS}$ 
17:   Add a new latent variable  $L_i$  to  $\mathbf{G}$  and to  $\mathbf{L}$ 
18:   For each observed variable  $O \in \mathbf{MSO}_i$ 
19:     Add  $O$  and an edge  $L_i \rightarrow O$  to  $\mathbf{G}$ 
20: % Fifth step: identify latent collider variables and their parents
21: For each  $L_i \in \mathbf{L}$ 
22:   % First phase
23:    $L_i.\mathbf{PAS} = \phi$ 
24:   For each  $L_j \in \mathbf{L}, j \neq i$ 
25:     If  $\exists \mathbf{PCC}_{fg} \in \mathbf{PCCM}$  s.t.  $(\mathbf{PCC}_{fg} \rightarrow \delta L_i \wedge \mathbf{PCC}_{fg} \rightarrow \delta L_j \wedge \mathbf{PCC}_{fg} \rightarrow \neg\delta L_k, \forall k \neq i, j)$ , then
26:       Add  $L_j$  to  $L_i.\mathbf{PAS}$ 
27:   % Second phase
28:   if  $\forall \mathbf{PCC}_{fg} \in \mathbf{PCCM}$  s.t.  $\mathbf{PCC}_{fg} \rightarrow \delta L_i$ 
29:      $\exists \mathbf{PAS} \in L_i.\mathbf{PAS}$  s.t.  $\mathbf{PCC}_{fg} \rightarrow \delta \mathbf{PAS}$ 
30:     then  $\forall \mathbf{PAS} \in L_i.\mathbf{PAS}$ , add a new edge  $\mathbf{PAS} \rightarrow L_i$  to  $\mathbf{G}$ .
31: % Sixth step: search for a new set of major clusters
32:  $\mathbf{NMC} = \phi$ 
33: Find the cardinality of each  $L_i \in \mathbf{L}$ , then identify  $\mathbf{ex}$ 
34: For each  $\mathbf{ex} \in \mathbf{ex}$ 
35:   Find  $c^* = \mathit{argmax}_{c \in \mathbf{c}} P(c | \mathbf{ex})$  and add  $c^*$  to  $\mathbf{NMC}$ 
36:   If  $\mathbf{NMC} = \mathbf{MC}$ 
37:     Return  $\mathbf{G}$ 
38:   Else
39:      $\mathbf{MC} \leftarrow \mathbf{NMC}, \mathbf{PCCM} = \mathbf{0}, \mathbf{L} = \phi, \mathbf{G} \leftarrow$  empty graph over  $\mathbf{O}$ 
40:     Go to “Third step”

```

if fulfilled, it adds an edge from each variable in L_i , **PAS** to L_j to complete the identification of L_j as a collider.

In the sixth and last step, and to deal with possible false positive and false negative errors (Section 4.3 of Part I), LEXC searches for a new set of major clusters **NMC** based on the already learned graph and all the clusters that initially were identified by SOM. First, LEXC learns for each latent $L_i \in \mathbf{L}$ its cardinality, which is the number of different value configurations of L_i corresponding to all value configurations of \mathbf{Ch}_i in \mathbf{D} . Each such value configuration of observed children is due to a value l_i of L_i , and we denote it by $l_i \rightarrow \mathbf{ch}_i$. Then, LEXC finds the set of all possible **exs** (all possible configurations of all exogenous latents in \mathbf{L} , $L_i \in \mathbf{L} \cap \mathbf{EX}$). For each **ex**, LEXC finds the most probable cluster, $c^* = \mathit{argmax}_{c \in \mathbf{C}} P(c|\mathbf{ex})$, where the posterior probability $P(c|\mathbf{ex})$ for each $c \in \mathbf{C}$ is approximated by the ratio between c 's size and the size of \mathbf{D} . Thus, the cluster for which the values corresponding to the children of $L_i \in \mathbf{L} \cap \mathbf{EX}$, $l_i \rightarrow \mathbf{ch}_i$, are most probable due to l_i in **ex** is selected as the most probable to represent this **ex**. Each such cluster is added to **NMC**. If **NMC=MC**, **NMC** cannot improve the graph, and thus LEXC stops and returns the learned graph \mathbf{G} . Otherwise, LEXC reinitializes **MC** to be **NMC** and relearns a new graph.

2.2 Learning latent non-colliders (LNC)

Using the data set \mathbf{D} , LNC has to split the set of latent variables \mathbf{L} in graph \mathbf{G} , which was learned by LEXC, into exogenous latents and latent non-colliders. First, LNC (Algorithm 2) adds $|\mathbf{L}|$ elements to the end of each vector in \mathbf{D} and creates an incomplete data set **IND**. For a vector in **IND** for which values of the observed children for a specific latent $L_i \in \mathbf{L}$ take major values, the value of the latent can be reconstructed exactly, $l_i \rightarrow \mathbf{ch}_i$; however, when not all observed children take major values, this value of the latent cannot be reconstructed, and this is the reason why **IND** is incomplete. Second, using the EM algorithm (Lauritzen, 1995; Dempster et al., 1977) and **IND**, LNC learns (Section 4.4 of Part I) \mathbf{G} 's parameters and uses them to compute a threshold (Appendix B in Part I) on the maximal size of 2-MCs. This threshold is needed to find 1-order minor clusters (1-MCs; Definition 14). Note that after learning the parameters, the graph turns into a model, \mathbf{M}_0 . Third, for each exogenous latent $EX_i \in \mathbf{L} \cap \mathbf{EX}$ in turn, LNC tests if EX_i should be split (Section 4.4 of Part I). For this test, LNC needs first to find the set of 1-MCs for EX_i , and to compute all the PCCs between these clusters and the major clusters for EX_i . We denote the set of these PCCs by **PCCS**. Then, LNC finds all the PCCs in **PCCS** that are **2S-PCC** (Definition 18); these will be used to identify all possible **2S-MSOs** (Definition 19) and thus all possible latent non-collider descendants that should be split from EX_i (Theorem 3).

After identifying the latent non-colliders' descendants of EX_i and splitting them from EX_i , LNC finds the links between these latents (Section 4.4 of Part I). LNC first finds the set L' of all latents whose children change alone in some 2S-PCCs. These are the candidates to be EX_i or its leaves (Proposition 10). Then, for each $L \in L'$, LNC finds the 2S-PCCs in **2S-PCC** in which the observed children of L' do not change and are due to comparisons with the same major cluster. This set is denoted by **2S-PCC'**. Then, for every two latent non-collider descendants that were split from EX_i , LNC checks if there is a directed link

between them using Theorem 4. Note that, we assume by default that L is a leaf, so LNC does not need to redirect the links in the diverging connection case. After finding all the possible directed paths, LNC identifies if the connection is serial (in case $|\mathbf{L}'|$ is exactly two) and if so it makes the links on this path undirected; otherwise, the path is directed as part of a diverging connection. Finally, LNC returns a pattern \mathbf{G} , which represents a Markov equivalence class of the true graph.

Algorithm 2 LNC

```

1: Input: Data set  $\mathbf{D}$  over the observed variables  $\mathbf{O}$  and the graph  $\mathbf{G}$  learned by LEXC
2: Output: The final learned LVM  $\mathbf{G}$ 
3: Initialize: IND =  $\mathbf{0}$ , PCCS =  $\phi$ , 2S-PCC =  $\phi$ , 2S-MSOS =  $\phi$ , 2S-PCC' =  $\phi$ 
4: Create IND (see text)
5: Learn  $\mathbf{G}$ 's parameters using the EM algorithm to obtain an LVM,  $\mathbf{M}_0$ 
6: For each latent  $EX_i \in \mathbf{L}$ 
7:   Identify and split the latent non-collider descendants of  $EX_i$ 
8:   Find the set of 1-MCs according to  $\mathbf{M}_0$ 
9:   PCCS  $\leftarrow$  compute all PCCs between the 1-MCs and the major clusters for  $EX_i$ 
10:  2S-PCC  $\leftarrow$  find all 2S-PCCs in PCCS
11:  2S-MSOS  $\leftarrow$  find all possible 2S-MSOs using 2S-PCC
12:  For each 2S-MSO $_j \in \mathbf{2S-MSOS}$ 
13:    Add a latent non-collider  $NC_j$  to  $EX_i$ ,  $\mathbf{L}$ , and  $\mathbf{G}$ 
14:    For each observed variable  $O \in \mathbf{2S-MSO}_j$ 
15:      Split  $O$  from the children of  $EX_i$  and add an edge  $NC_j \rightarrow O$  to  $\mathbf{G}$ 
16:    Identify the links between the new latent non-colliders that were split from  $EX_i$ 
17:     $\mathbf{L}' \leftarrow$  all latents that were split (including  $EX_i$ ) and whose observed children change alone
    in some 2S-PCC
18:    For each  $L' \in \mathbf{L}'$  % assume by default  $L'$  is a leaf and apply Theorem 4
19:      2S-PCC'  $\leftarrow$  all 2S-PCCs in 2S-PCC in which the observed children of  $L'$  do not change
20:      For each two latent non-colliders  $NC_j, NC_k$ ,  $k \neq j$  that were split from  $EX_i$ :
21:        If
22:          1) the observed children of  $NC_k$  always change with those of  $NC_j$  in 2S-PCC'; and
23:          2) the observed children of  $NC_j$  change  $t$  times and the observed children of  $NC_k$ 
           change  $t+1$  times in 2S-PCC'
24:          Then add a directed edge from  $NC_k$  to  $NC_j$  to  $\mathbf{G}$ 
25:        % Identify if the connection is serial, and if so make the links in the path undirected
26:        If  $|\mathbf{L}'|=2$ 
27:          If there are two paths with the same latents but opposite directions, then make the edges
           between the latents undirected.
28:        Return  $\mathbf{G}$ 

```

3. LPCC Evaluation

We implemented the LPCC algorithm in Matlab, except for the SOM algorithm that was implemented using the SOM Toolbox (Vesanto et al., 2000). We evaluated LPCC using simulated data sets (Section 3.1) and five real-world data sets: data from the political action survey (Section 3.2), Holzinger and Swineford's data (Section 3.3), the HIV test data (Section 3.4), data of young driver (YD) involvement in road accidents (Section 3.5), and

a mass cytometry data set of the immune system (Section 3.6). In the case of the real-world data sets, we did not have an objective measure for evaluation; thus, we compared the LPCC output to hypothesized, theoretical models from the literature and to the outputs of four state-of-the-art learning algorithms. The first algorithm is FCI (Spirtes et al., 2000), and because we noticed (see below) for the political action survey and Holzinger and Swineford’s data sets that FCI is not suitable for learning MIM models, we did not use it for the other data sets. The second algorithm is for learning HLC models (Zhang, 2004), and since the theoretical models for all but the HIV data set are not latent-tree models, we used this algorithm only for the HIV data set. The third algorithm is exploratory factor analysis (EFA). Because the theoretical models for the political action survey and Holzinger and Swineford’s data set were already tested by confirmatory factor analysis [Joreskog, 2004; Arbuckle, 1997, p. 375]; and [Joreskog and Sorbom, 1989, p. 247]], we completed the examination of EFA also to the YD and mass cytometry data sets. The fourth algorithm, which is actually two algorithms, BuildPureClusters (BPC) and BuildSinglePureClusters (BSPC) of Silva (2005), is especially suitable for MIM models. BPC is Silva’s (2005) main algorithm; hence, we used it in all the evaluations. BPC assumes that the observed variables are continuous and normally distributed, whereas BSPC is a variant of BPC for discrete observed variables. We ran BPC using its implementation in the Tetrad IV package, which can take discrete data (as in all the data sets in this evaluation), as input and treat them as continuous.⁴ BPC learns LVM by testing Tetrad constraints at a given significance level (α). We used Wishart’s Tetrad test (Silva, 2005; Spirtes et al., 2000; Wishart, 1928), applying three significance levels of 0.01, 0.05 (Tetrad’s default), and 0.1. For the simulated data sets, we compared LPCC to EFA and BPC.

3.1 Evaluation using simulated data sets

We used Tetrad IV to construct the graphs G1, G2, G3, and G4 of Figure 2, once with binary and once with ternary variables. The priors on the exogenous latents were always distributed uniformly. We compared performances for three parameterization levels that differ by the conditional probabilities, $p_j=0.7, 0.75,$ and 0.8 , between a latent L_k and each of its children EN_i . For all graphs in the binary case, except L2 in G2, $P(EN_i = v | L_k = v) = p_j, v = 0$ or 1 . For all graphs in the ternary case, except L2 in G2, $P(EN_i = v | L_k = v) = p_j, P(EN_i \neq v | L_k = v) = (1 - p_j)/2, v = 0, 1,$ or 2 . Concerning L2 in G2, $P(L_2 = 0 | L_1, L_3 = 0, 0, 1, 1, 0) = P(L_2 = 1 | L_1, L_3 = 1) = p_j$ in the binary case and $P(L_2 = v | \max\{L_1, L_3\} = v) = p_j$ and $P(L_2 \neq v | \max\{L_1, L_3\} = v) = (1 - p_j)/2$ in the ternary case. Each such scheme imposes a different “parametric complexity” on the model and thereby affects the task of learning the latent model and the causal relations. That is, using $p_j=0.7$ poses a larger challenge to learning than $p_j=0.75$, which poses a larger challenge than $p_j=0.8$. For example for G3 and the binary case, the correlations between any latent and any of its children for the parametric settings $p_j=0.7, 0.75,$ and 0.8 are $0.4, 0.5,$

⁴Although all our data sets are discrete and BSPC is the suggested algorithm in Silva (2005) for discrete data, BSPC is neither published nor implemented in Tetrad IV, and is only mentioned in a complementary chapter in Silva (2005) as a variant of BPC suitable for discrete data. Since no concrete algorithm is suggested for BSPC, we used BPC as described above. However, for the political action survey, we could use the results for BSPC that are provided in Silva (2005). The Tetrad package is available at <http://www.phil.cmu.edu/projects/tetrad>.

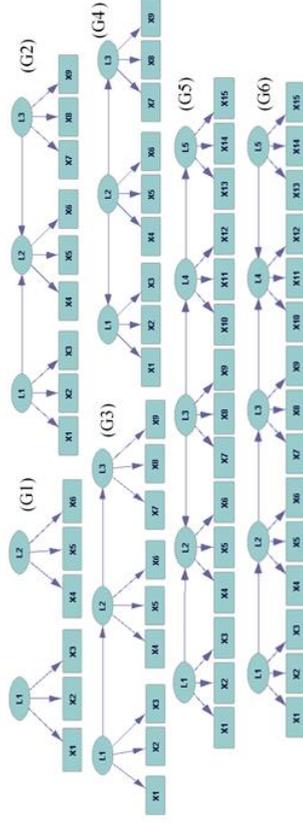


Figure 2: Example LVMs that are all MIMs. Each is based on a pure measurement model and a structural model of different complexity, posing a different challenge to a learning algorithm.

and 0.6 , respectively. Note that correlation of 0.4 is relatively low, providing a great challenge to the learning algorithms, and trying to learn an LVM for lower correlation values yields poor results by all algorithms.⁵ Tetrad IV was also used to draw data sets of 125, 250, 500, 750, 1000, and 2,000 samples for each test. Overall, we evaluated the LPCC algorithm using 144 synthetic data sets for four graphs (G1–G4), two types of variables, three parameterization levels, and six data set sizes.

In addition, we evaluated LPCC using the two large graphs in Figure 2, G5 and G6, which combine all types of links between the latents, such as serial, converging, and diverging. Each graph has five latents with three observed children each. Tetrad IV was used to draw data sets of 250, 500, 1000, and 2,000 samples, where all variables are binary and for two parametric settings $p_j=0.75$ and 0.8 . In all cases, we report on the structural hamming distance (SHD) (Tsamardinos et al., 2006) as a performance measure for learning the LVM structure. SHD is a global structural measure that accounts for all the possible learning errors: addition and deletion of an undirected edge, and addition, removal, and reversal of edge orientation.

Figures 3–5 show learning curves for SHD (the lower value is the better one) and increasing sample sizes for LPCC, BPC, and EFA. Figures 3 and 4 show SHD performance in learning G1–G4 with binary variables and ternary variables, respectively, and for two parametric settings, $p_j=0.7$ and 0.8 . Figure 5 shows performance in learning G5 and G6 with binary variables for two parametric settings $p_j=0.75$ and 0.8 (for $p_j=0.7$ the algorithms performed poorly and thus their results are excluded here). In addition, in Appendix B, we compare LPCC with BPC (Section B.1) and with EFA (Section B.2) in learning G1–G4 with binary and ternary variables for three parametric settings, $p_j=0.7, 0.75,$ and 0.8 . The graphs demonstrate the LPCC sensitivity to the parametric complexity – the

⁵For example, a common practice in EFA is that a correlation (loading) of at least 0.4 is needed in order to add a link between a latent variable and an observed variable.

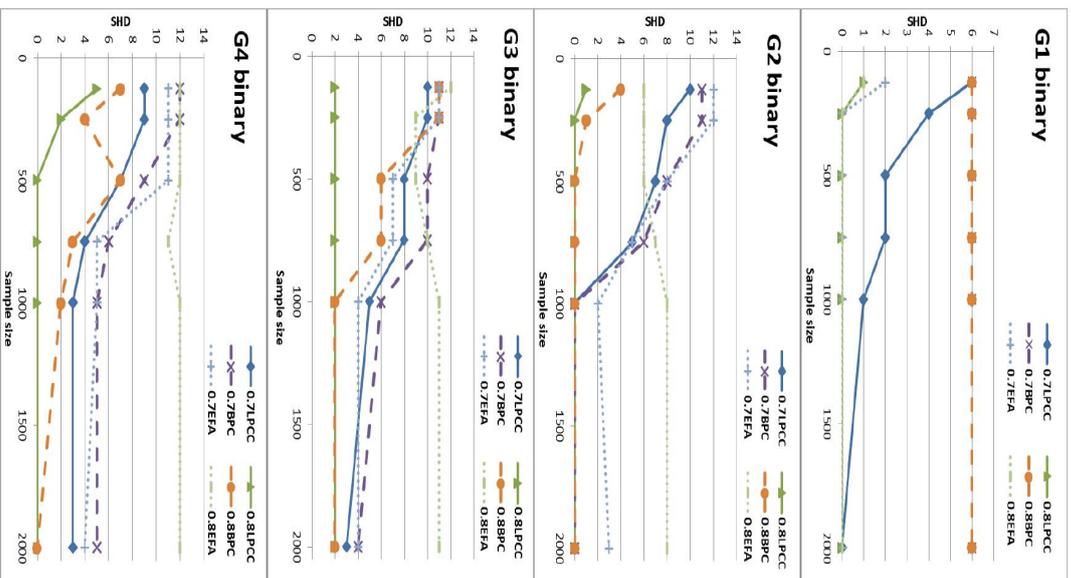


Figure 3: SHD learning curves of LPCC compared to those of BPC and EFA for G1–G4 of Figure 2 with binary variables, two parameterization levels, and increasing sample sizes. The lines of LPCC and EFA for a parametrization of 0.8 coincide for G1.

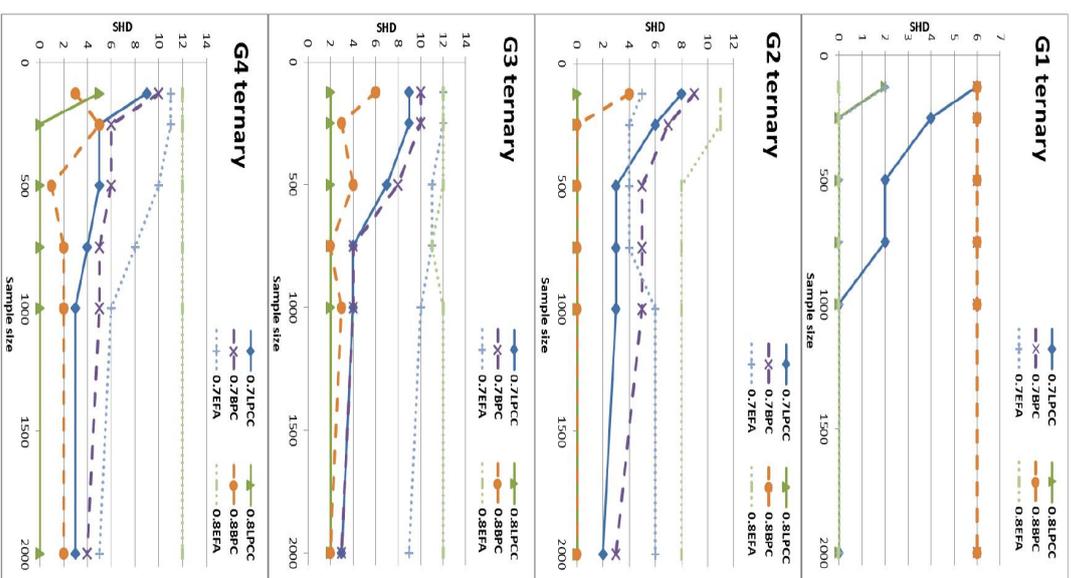


Figure 4: SHD learning curves of LPCC compared with those of BPC and EFA for G1–G4 of Figure 2 with ternary variables, two parameterization levels, and increasing sample sizes. The line of LPCC for a parametrization of 0.7 coincides with that of EFA for a parametrization of 0.7 for G1.

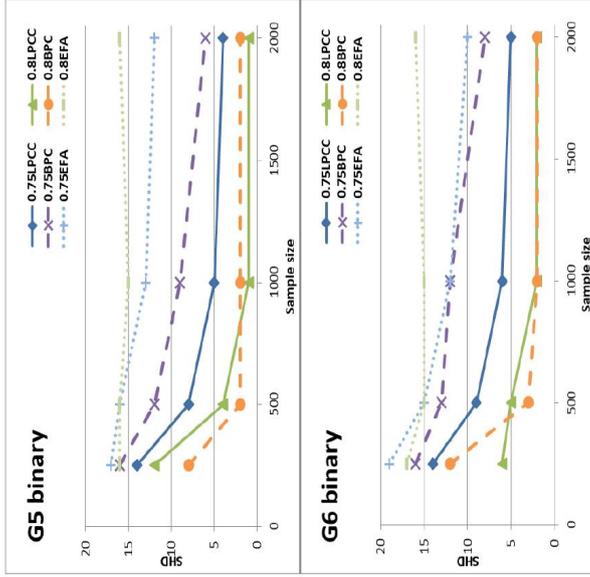


Figure 5: SHD learning curves of LPCC compared to those of BPC and EFA for G5 and G6 of Figure 2 with binary variables, two parameterization levels, and increasing sample sizes.

lower the complexity is, the faster learning is and the sooner the error vanishes – and the LPCC good asymptotic behavior, demonstrating accuracy improvement with the sample size. Generally, Figures 3–5 (and those in Appendix B) show superiority of LPCC over BPC and EFA; LPCC demonstrates higher accuracies (smaller errors) and a better asymptotic behavior than BPC and EFA.

Specifically in regard to EFA, the algorithm – contrary to what is expected from a learning algorithm (see LPCC and BPC) – fails as the experiment conditions improve and the learning task becomes easier (e.g., larger parameterization levels and/or data samples as in the graphs for G4 with binary variables and G2 with ternary variables). Larger parameterization levels increase the chances of EFA to learn links between latent variables and observed variables – some of them are not between a latent and its real child – to compensate for the algorithm’s inability to identify links among latents (as EFA assumes latents are uncorrelated). The increase in the sample size helps increase the confidence of EFA in learning these erroneous links (see the graphs for G2, G5, and G6 with binary variables). As Figures 3–5, together with the more detailed Figures 19 and 20 in Appendix B, demonstrate, EFA is inferior to LPCC for all parameterization levels and sample sizes

and all graphs but G1. Independent latent variables, as manifested in G1, is the ultimate prerequisite for a successful application of EFA, and indeed, EFA shows competitive (and sometimes, for small sample sizes, even slightly improved) performance to LPCC in learning G1.

Unlike LPCC, BPC is not suitable for learning models such as G1, where the latents are independent and each has fewer than four observed children. This is because BPC requires the variables in a Tetrad constraint to all be mutually dependent, where in the case of G1, there are at most three mutually dependent variables, so no Tetrad constraint can be tested, and no graph is learned (SHD=6 for missing all the six edges in G1). However, it is reasonable to assume that a practitioner would naturally analyze the data before trying BPC, and if they recognize that not all observed variables are correlated (e.g., X1 and X4 for G1), then they will not use BPC. As Figures 3–5, together with the more detailed Figures 17 and 18 in Appendix B, demonstrate, for most graphs, parameterization levels, and sample sizes (except for some cases with small sample sizes), LPCC is superior to BPC.

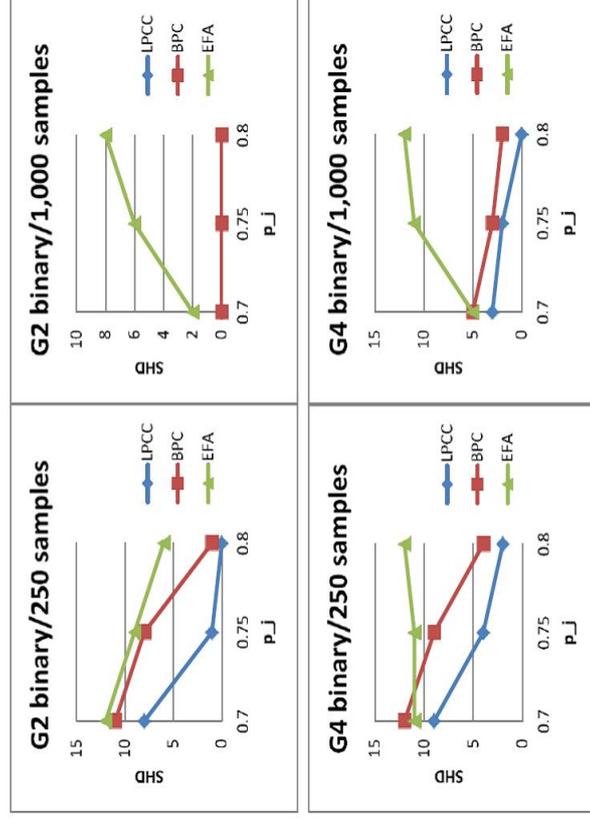


Figure 6: SHD of the LPCC, BPC, and EFA algorithms for increasing parameterization levels for four combinations of learned graphs (G2 and G4) and sample sizes (250 and 1,000 samples). Note that for G2/1,000 samples, both LPCC and BPC learn the structure perfectly for any parameterization level.

Another view of these results is manifested in Figure 6 that shows SHD values for the LPCC, BPC, and EFA algorithms for increasing parametrization levels for four combinations of learned graphs and sample sizes. Figure 6 shows that both LPCC and BPC improve performance, as expected, with increased levels of latent-observed variable correlation (ρ_j): LPCC never falls behind BPC, and its advantage over BPC is especially vivid for a small sample size. EFA, besides falling behind LPCC and BPC, also demonstrates worsening of performance with increasing the parametrization level, especially for large sample sizes, for the reasons provided above.

Finally, we expand the evaluation by examining the algorithms when the number of indicators a latent has increases. Figure 7 shows the SHD values of the LPCC, BPC, and EFA algorithms for increasing numbers of binary indicators per latent variable in G2, a parametrization level (ρ_j) of 0.75, and four sample sizes. The figure exhibits clear superiority of LPCC over BPC and EFA for almost all numbers of indicators and sample sizes. While LPCC hardly worsens its performance with the increase of complexity (number of indicators a latent has), both BPC and EFA are affected by this increase. Also worth mentioning is the difficulty these two latter algorithms have in learning an LVM for which latent variables have exactly two indicators, regardless of the sample size.

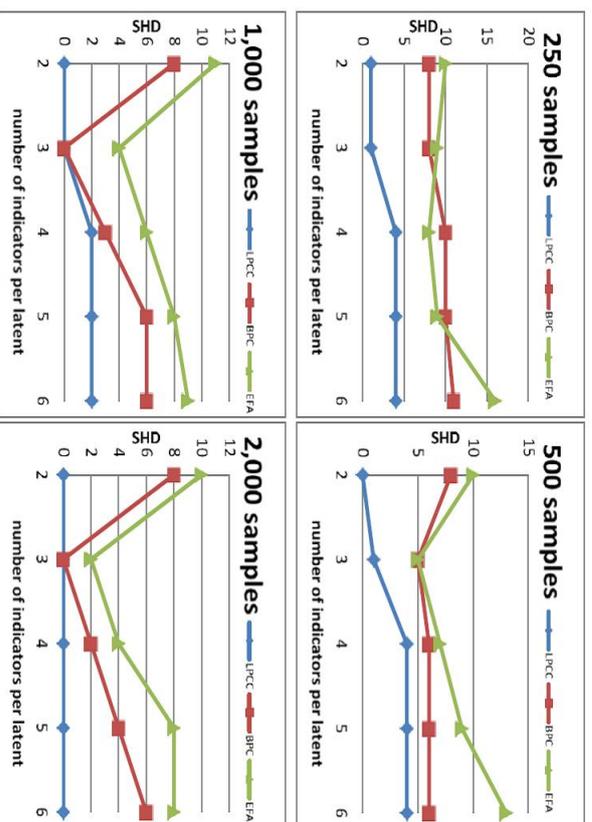


Figure 7: SHD values of the LPCC, BPC, and EFA algorithms for increasing numbers of binary indicators a latent variable has in G2, $\rho_j = 0.75$, and four sample sizes.

To understand the differences among the three algorithms in more detail, we analyze the errors they make. For example, when using 1,000 samples (the reference is the bottom-left graph in Figure 7). When the number of indicators a latent has is less than 4, LPCC learns the LVM perfectly, and when this number is greater, LPCC errs twice in missing an edge from a latent to one of its indicators. BPC cannot learn an LVM using two indicators per latent, and thus it misses all eight edges in G2 and returns an empty graph. It successfully learns the LVM when each latent has exactly three indicators, but then fails to direct the edges among the latent variables and misses at least a single edge between a latent and an indicator when the latent variables have more than three indicators each. For two indicators per latent, EFA detects only two factors and fails to connect them. It connects one factor to six indicators and the second factor to five indicators, and thereby errs in learning seven extra edges from latent variables to observed variables, missing two edges from the missing latent variable to two observed variables, and missing the two edges among the latent variables, which accounts for eleven errors in total. For three indicators per latent, EFA detects three indicators for two of the latents and five indicators for the other and misses the edges among the latents, which accounts for four errors in total. For four to six indicators per latent, EFA learns more extra edges between the latent and observed variables, together with missing the edges among the latents. This experiment vividly demonstrates the advantage of LPCC over BPC and EFA in that not only does LPCC detect edges between latent and observed variables more accurately, but it also detects latent-latent connections in all scenarios, which is impressive especially when the sample size is small and/or the number of indicators a latent has is large.

3.2 The political action survey data

We evaluated LPCC using a simplified political action survey data set over the following six variables (Joreskog, 2004):

- NOSAY: “People like me have no say in what the government does.”
- VOTING: “Voting is the only way that people like me can have any say about how the government runs things.”
- COMPLEX: “Sometimes politics and government seem so complicated that a person like me cannot really understand what is going on.”
- NOCARE: “I don’t think that public officials care much about what people like me think.”
- TOUCH: “Generally speaking, those we elect to Congress in Washington lose touch with people pretty quickly.”
- INTEREST: “Parties are only interested in people’s votes, but not in their opinions.”

These six variables represent the operational definition of political efficacy and correspond to questions to which the respondents have to give their degree of agreement on a discrete ordinal scale of four values. This data set is available as part of the LISREL software for latent variable analysis and contains the responses to these questions from a

sample of 1,076 United States respondents. A model consisting of two latents that correspond to a previously established theoretical trait of Efficacy and Responsiveness based on Joreskog (2004) is given in Figure 8a. VOTING is discarded by Joreskog for this particular data based on the argument that the question for VOTING is not clearly phrased.

Similar to the theoretical model, LPCC finds two latents (Figure 8b): One corresponds to NOSAY and VOTING and the other corresponds to NOCARE, TOUCH, and INTEREST (a detailed description of the PCC analysis that led to these results is in Appendix C). Compared with the theoretical model, LPCC misses the edge between Efficacy and NOCARE and the bidirectional edge between the latents. Both edges are not supposed to be discovered by LPCC or BSPC/BPC; the former because the algorithms learn a pure measurement model in which each observed variable has only one latent parent and the latter because no cycles are assumed. Nevertheless, compared with the theoretical model, LPCC makes no use of prior knowledge.

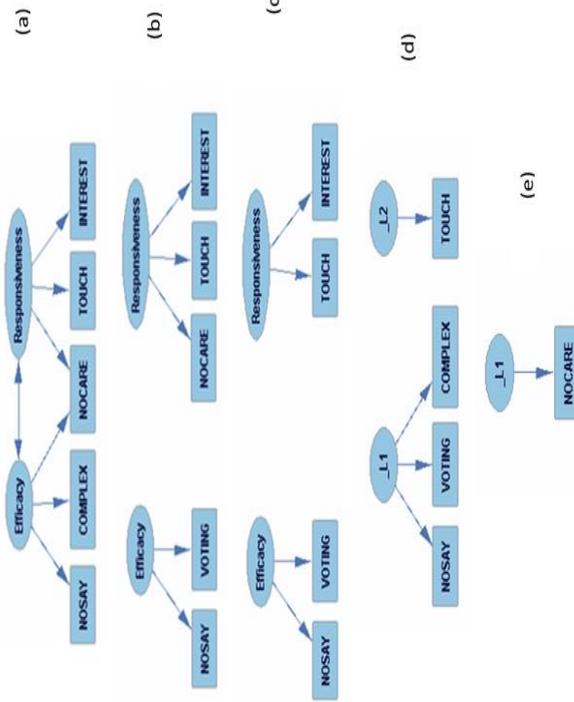


Figure 8: The political action survey: (a) A theoretical model (Joreskog, 2004) and five outputs of (b) LPCC, (c) BSPC, (d) BPC for $\alpha=0.01$ and 0.05, and (e) BPC for $\alpha=0.1$.

BSPC output (Figure 8c) is very similar to LPCC output, except for NOCARE, which was not identified by BSPC as a measure of Responsiveness, making the output obtained by LPCC closer to the theoretical model than that of BSPC. In addition, both algorithms

identify VOTING as a child of Efficacy (at the expense of COMPLEX), and thereby challenge the decision made in Joreskog (2004) to discard VOTING from the model. The outputs of the BPC algorithm (Figure 8d) for both $\alpha=0.01$ and $\alpha=0.05$ are poorer than those of LPCC and BSPC. BPC finds two latents. The first latent corresponds to NOSAY, VOTING, and COMPLEX with partial resemblance to the theoretical model (identifying NOSAY and COMPLEX as indicators of this latent) and partial resemblance to the outputs of LPCC and BPC (identifying NOSAY and VOTING as indicators of the latent). However, the second latent found by BPC corresponds only to TOUCH and misses INTEREST (identified in the theoretical model and by LPCC and BSPC as an indicator of Responsiveness) and NOCARE (that is identified in the theoretical model and by LPCC as an indicator of Responsiveness). The output of the BPC algorithm using $\alpha=0.1$ (Figure 8e) gives very little information about the problem as it finds only one latent that corresponds only to NOCARE. These last two figures show the sensitivity of BPC to the significance level, which is a parameter whose value should be determined beforehand. Note that the success of the LPCC and BSPC algorithms emphasizes the importance of such algorithms in learning discrete problems.

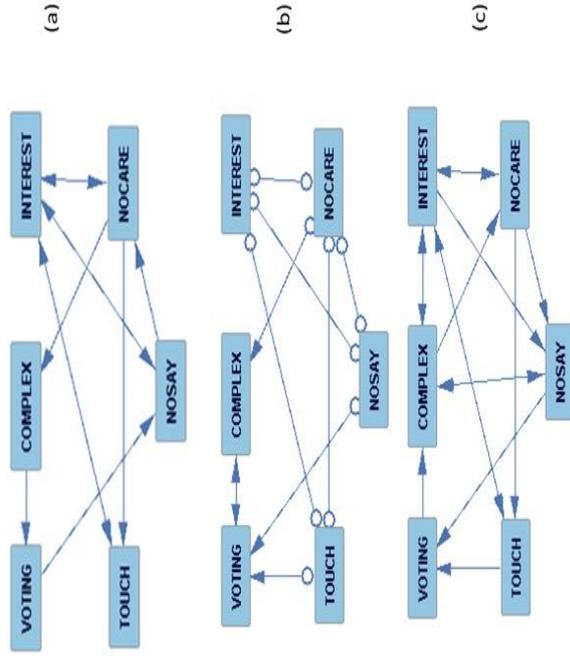


Figure 9: FCI outputs for the political action survey data set and significance levels of (a) 0.01, (b) 0.05, and (c) 0.1.

The outputs of the FCI algorithm using any of the above significance levels are not sufficient (Figure 9). For example, the FCI outputs that were learned using $\alpha=0.01$ (Figure 9a) and 0.05 (Figure 9b) show that NOSAY and INTEREST potentially have a latent common cause. However, these two variables are indicators of different latents in the theoretical model. These results are understandable because unlike LPCC, BPC, and BSPC, FCI is not suitable for learning MIM models such as the political action survey.

3.3 Holzinger and Swineford’s data

Holzinger and Swineford (1939) collected data from 26 psychological tests administered to 145 seventh- and eighth-grade children in the Grant-White School in Chicago, Illinois. In this evaluation, we use a subset of this data over only six variables representing the scores in six intelligence tests. The variables are: scores on a visual perception test (VisPerc), scores on a cube test (Cubes), scores on a lozenge test (Lozenges), scores on a paragraph comprehension test (ParComp), scores on a sentence completion test (SenComp), and scores on a word meaning test (WordMean). There are two hypothesized intelligence factors, which are spatial ability and verbal ability factors. The first three variables measure spatial ability and the latter three variables measure verbal ability. A confirmatory factor model that fits this data well was extracted from the Amos manual (Arbuckle 1997, p. 375; Joreskog and Sorbom 1989, p. 247) and is shown in Figure 10a.

We ran LPCC using a dichotomous (binary) presentation of the continuous data. For each variable, scores that were above the average score were recorded as 2, and scores below the average score were recorded as 1. Despite the small size of the data set and the loss of information due to the discretization process, LPCC found two latents (Figure 10b). The first latent corresponds to VisPerc and Lozenges, and the other latent corresponds to ParComp, SenComp, and WordMean (a detailed description of the PCC analysis is in Appendix C). Our model matches the theoretical model, except for missing one link between Spatial and Cubes (and the link between the latents that the model is not supposed to identify).

The outputs of the BPC algorithm using $\alpha=0.01$ and 0.05 (Figure 10c) were not good compared to the theoretical model and LPCC output. In both cases, BPC found only a single latent variable that corresponds to only four of the six indicators, specifically, VisPerc, Lozenges, Cubes, and WordMean. Notice that WordMean and the other three variables belong to two different latent variables in the theoretical model. However, for a significance level of 0.1 , BPC output (Figure 10d) is the closest of all models to the theoretical model. These results show the sensitivity of BPC to the significance level, which is a parameter that does not have a predetermined value. LPCC does not have this disadvantage. Note that the superiority of BPC for a significance level of 0.1 for Holzinger and Swineford’s data set is in contrast to the model inferiority for other significance levels (Figure 10e) and for the political action survey data set with any significance level.

The output of the FCI algorithm using a significance level of 0.01 or 0.05 (Figure 11a) indicates that ParComp, SenComp, and WordMean potentially have a latent common cause. In addition, Lozenges potentially has a latent common cause with VisPerc and with Cubes, but there is no link between VisPerc and Cubes. For α of 0.1 , the output of the FCI algorithm (Figure 11b) indicates that ParComp, SenComp, and WordMean potentially

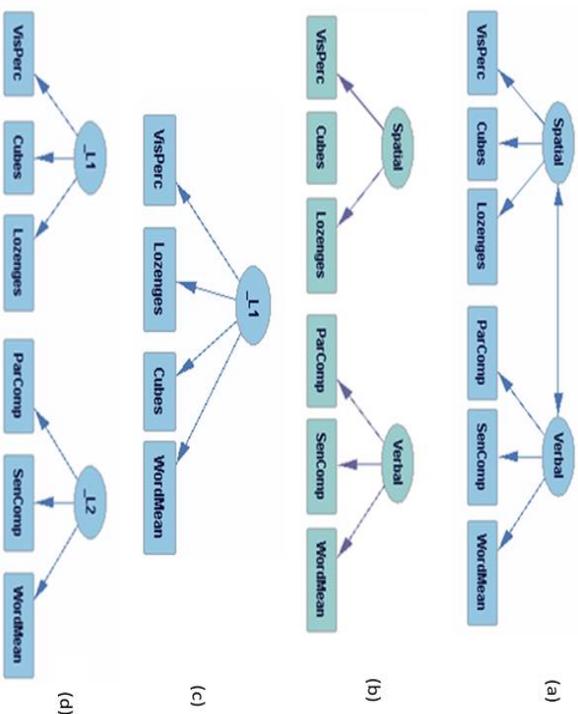


Figure 10: (a) A theoretical model for Holzinger and Swineford’s data set based on a confirmatory factor model that fits this data well and the outputs of (b) LPCC, (c) BPC for $\alpha=0.01$ and 0.05 , and (d) BPC for $\alpha=0.1$.

have a latent common cause, and Lozenges, VisPerc, and Cubes potentially have another latent common cause. This model matches the theoretical model (Figure 11a) except for the bidirectional edge between the latents.

3.4 The HIV test data

We also evaluated LPCC using the HIV test data (Zhang, 2004). This data set consists of results for 428 subjects of four diagnostic tests for the human immunodeficiency virus (HIV): “radioimmunoassay of antigen ag121” (A); “radioimmunoassay of HIV p24” (B); “radioimmunoassay of HIV gp120” (C); and “enzyme-linked immunosorbent assay” (D). A negative result is represented by 0 and a positive result by 1. LPCC learned a model identical to that in Zhang (2004) (Figure 12), where X1 and X2 are both binary latent variables. However, unlike the algorithm in Zhang (2004) that aims at learning tree-latent models like the one required for the HIV data, LPCC is not limited to latent-tree models. BPC returned an empty model for any conventional α .

of traffic offenses for the first six months after obtaining a driving license. YDs in the two databases were grouped according to the following classes:

DB1:

Accident and no offense: All YDs who had at least one accident and committed at least one offense in the three-month period after ADP (the “period”). There were 345 such drivers; hence, this number defined a group size.

Offense but no accident: 345 drivers who committed at least one offense, but had no accidents in the period.

Accident but no offense: 345 drivers who had at least one accident, but committed no offense in the period.

No accident and no offense: 345 drivers who did not have any accidents or commit any offenses in the period.

In total, there are 1,380 observations (YDs) for DB1.

DB2:

Accident and offense: All YDs who had at least one accident and committed at least one offense in the period (similar to this class in DB1).

No accident and no offense: 345 drivers who did not have any accidents or commit any offenses in the period (similar to this class in DB1).

In total, there are 690 observations (YDs) for DB2.

All observations in both databases are represented by thirteen observed variables that a previous study indicated as relevant to the explanation of YD involvement in road accidents and offenses (Lerner, 2012; Lerner and Meyer, 2012). In addition, we used four observed variables that indicate if YDs or their parents were involved in a road accident or an offense. A detailed description of all seventeen observed variables is given in Table 1.

We ran LPCC on DB1 and DB2 and compared its results to those of EFA and BPC. We ran BPC using a significance level (alpha) of 0.05 (Tetrad’s default). Exploratory factor analysis was applied in two phases. First, principal component analysis (PCA) was used for factor extraction, where the Kaiser criterion (Kaiser, 1960) was used for determining the number of factors. Any factor with an associated eigenvalue less than 1.0 was dropped because this value is equal to the information accounted for by an average single observed variable. Second, the factor model was rotated using varimax, which is an orthogonal rotation method of the factor axes to maximize the variance of the squared loadings of a factor on all the variables. Factor loadings, also called component loadings in PCA, are the correlation coefficients between the variables and factors, indicating how strongly the latter influence the former. Analogous to Pearson’s correlation coefficient, the squared factor loading is the percent of variance in that indicator variable explained by the factor. The varimax rotation method has the effect of differentiating the original variables by the extracted factors. Each factor tends to have either large or small loadings of any particular variable. A varimax solution yields results that make it as easy as possible to identify each variable with a single factor (with the highest loading on the variable). In confirmatory factor analysis (CFA), loadings should be 0.7 or above to confirm that independent variables identified a priori are represented by a particular factor, using the rationale that the

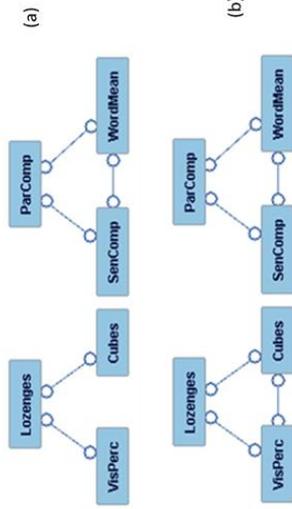


Figure 11: FCI outputs for Holzinger and Swineford’s data set and significance levels of (a) 0.01 and (b) 0.1.

3.5 Explanation of young drivers’ involvement in road accidents using LPCC

We produced two databases (DB1 and DB2) from a main database that includes all young drivers (between the ages of 18 and 24 years) who received their private-car driving licenses in Israel between 2002 and 2008. The main database includes more than 600,000 drivers and their parents who were involved in more than 600,000 road accidents and committed more than 2,000,000 traffic offenses in this period. We were interested in explaining young driver (YD) involvement in road accidents and offenses using LPCC and the databases. By “explaining”, we mean that we wanted to find the factors among all variables representing a driver, car, accident, offenses, and so forth and the interrelations that could explain YD involvement in road accidents and offenses. These factors could also contribute to prediction of YD involvement in road accidents and offenses with the highest accuracy. We concentrated on the first three months after the accompanied driving phase (ADP), which is a three-month driving phase in which a YD is accompanied by an experienced driver. We concentrated on the three months after ADP because: (1) this is the first solo experience of YDs, and it is when they commit most of their traffic offenses or are involved in most of their road accidents; and (2) we only had detailed monthly records

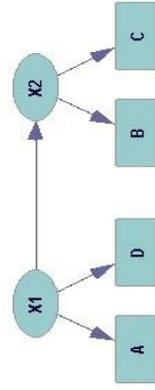


Figure 12: Model learned for HIV using LPCC.

0.7 level corresponds to about half of the variance in the indicator being explained by the factor. However, the 0.7 standard is high, and real-world data may not meet this criterion, which is why some researchers, including us in this study (particularly for exploratory purposes such as this case), use a lower level, where 0.4 is the common practice (Marly, 1994). In addition, we adapted Occam's razor parsimony principle (to explain the variance with the fewest possible factors) and required the variance explained criterion to be above 50%.

Number	Variable	Variable short name	Variable values
1	Age	gil	1 (17-18), 2 (19-20), 3 (21-22), 4 (23-24)
2	Gender	Min	1 (male), 2 (female)
3	Medical limitations	lim	1 (no), 2 (yes)
4	Father is allowed to drive	MurF	1 (yes), 2 (no)
5	Mother is allowed to drive	MurM	1 (yes), 2 (no)
6	Has a motorcycle license	of	1 (no), 2 (yes)
7	Received "Or Yarok" kit, as part of a graduated driver licensing program ⁶	or	1 (didn't receive), 2 (received)
8	Socioeconomic index	GF	1-4 (1-low, 4-high)
9	Ethnic group	KU	1 (Jew), 2 (non-Jew)
10	Father's marital status	RS	1 (single), 2 (married), 3 (divorced), 4 (widowed)
11	Mother's marital status	MS	1 (single), 2 (married), 3 (divorced), 4 (widowed)
12	Father's number of years of education	FED	1-4 (1-low, 4-high)
13	Mother's number of years of education	MED	1-4 (1-low, 4-high)
14	Offenses of YD	OFYD	1 (no), 2 (yes)
15	Accidents of YD	ACYD	1 (no), 2 (yes)
16	Offenses of parents	OFPA	1 (no), 2 (yes)
17	Accidents of parents	ACPA	1 (no), 2 (yes)

Table 1: Seventeen observed variables in DB1 and DB2

Results for DB1:

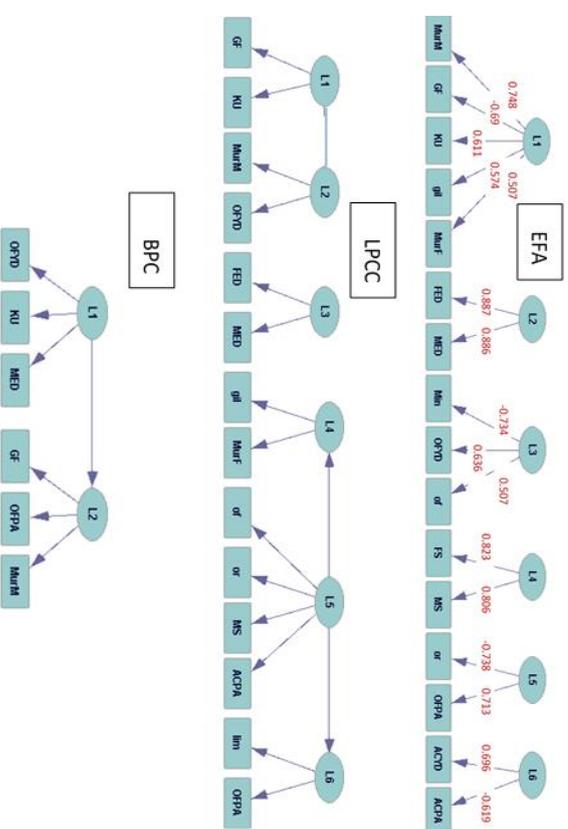


Figure 13: LVMs learned by EFA, LPCC, and BPC for DB1. Numbers for EFA represent the factor loadings, where plus and minus signs indicate positive and negative correlation, respectively.

Both LPCC and EFA found six latent variables (Figure 13). L1 in EFA is a parent of five observed variables, where each describes another aspect of the demographic or socioeconomic state of the YD or his/her family. The variables (see Table 1) *father is allowed to drive* (MurF), *mother is allowed to drive* (MurM), *age* (gil), and *ethnic group* (KU) are positively associated, whereas the variable *socioeconomic index* (GF) is negatively associated with L1. Based on the values of these variables, we can identify groups in the YDs' population, such as a group of YDs that are not Jewish, their parents are not allowed to drive, and their age and their socioeconomic status are low. L1 is not connected by EFA to other latent variables that relate to YD involvement in road accidents or offenses; thus, it does not contribute much to this study. However, L1 and L2 as learned by LPCC relate the socioeconomic state of the YD family (GF and KU are children of L1) with YD offenses (OFYD

⁶The Or Yarok (i.e., "green light" in Hebrew) kit includes documentation and accessories, such as CDs, with instructions, movies, and advice regarding safe driving. Granting the kit before licensure was found (Lerner, 2012) helpful in reducing young drivers' involvement in road accidents.

is a child of L2), thus LPCC links the socioeconomic state of YD with its involvement in traffic offenses. For example, according to LPCC, a YD who is not Jewish, with a low socioeconomic index, and whose mother is not allowed to drive is more likely to be involved in traffic offenses.

Latents L2 in EFA and L3 in LPCC describe the educational status of the YDs' parents and indicate a similar tendency between the parents' educational levels. However, both EFA and LPCC analyses do not link the educational status of the YD family to involvement in road accidents and traffic offenses.

L3 in EFA shows a negative relationship between a YD's gender (Min) or having a motorcycle license (of) and his/her tendency to commit road offenses. Male drivers tend to commit more traffic offenses than female drivers, and drivers who have a motorcycle license tend to commit more traffic offenses than drivers who do not have such a license. L4 in EFA shows the marital status of the YD parents, and not surprisingly, it indicates that the father's and mother's marital status is correlated. L5 in EFA shows that the parents of YDs who did not receive the "Or Yarok kit" tend to commit more traffic offenses. That is, the introduction of the kit in the family seems to also reduce involvement in road offenses of family members other than the YD. L6 in EFA shows a negative relationship between YD involvement in road accidents and the involvement of their parents in accidents. One explanation for this negative relationship could be that in a family in which one member was involved in an accident, other members tend to be more careful and thus decrease their involvement in accidents. Due to the independency assumption (between the factors) in EFA, L3, L4, L5, and L6 are not related, and EFA is not able to holistically represent relations among variables representing demographic and socioeconomic characteristics and road accidents and offenses of YDs and their parents.

LPCC describes involvements in road accidents and offenses in a more comprehensive way using a structure that is based on three latents with a diverging link from L5 to L4 and L6. L5 shows a relationship among the variables *motorcycle license*, *received "Or Yarok kit"*, *mother's marital status*, and *parents' involvement in accidents*. For example, it was found that there is a relation between receiving the "Or Yarok kit" and decreasing values of parents' involvement in accidents. However, we note that the variable parents' involvement in accidents is sparse in DB1, making its relationship with the other variables via L5 quite arguable. An interesting relationship found in the LPCC results is between parents' accidents (child of L5) and parents' offenses (child of L6). This relationship was missed by EFA since each variable in EFA is a child of a different latent that is independent of the other latent (due to EFA's orthogonality assumption).

BPC finds only two latents compared to six latents that are found by LPCC and EFA. A possible explanation for the low number of latents identified by BPC is that BPC requires that a latent have at least three observed children to be identified, whereas LPCC requires only two (see latents L1–L4 and L6 in the LPCC model, each having two observed children). L1 and L2 as learned by BPC relate the demographic-socioeconomic state of the YD family (KU and GF) with YD offenses, as the LPCC model did, but the two latents in the BPC model mix the indicators. It is more reasonable to believe that if L1 is a parent of L2 as BPC identified, then offenses of YDs (OFYD) should be a child of L2 and not of L1, and GF should be a child of L1 and not of L2 since socioeconomic status is expected to affect violent road behavior and not the opposite. BPC also relates OFYD with offenses

of their parents (OFFA), a relation that seems reasonable and is not identified by EFA and LPCC. However, the identification of OFYD as L1's child and OFFA as L2's child implies that it is the YD offenses that affect the offenses of their parents and not the opposite, as may be expected. Another relation that is identified only by BPC is between the mother's education level (MED) and both YDs' and their parents' offenses.

Results for DB2:

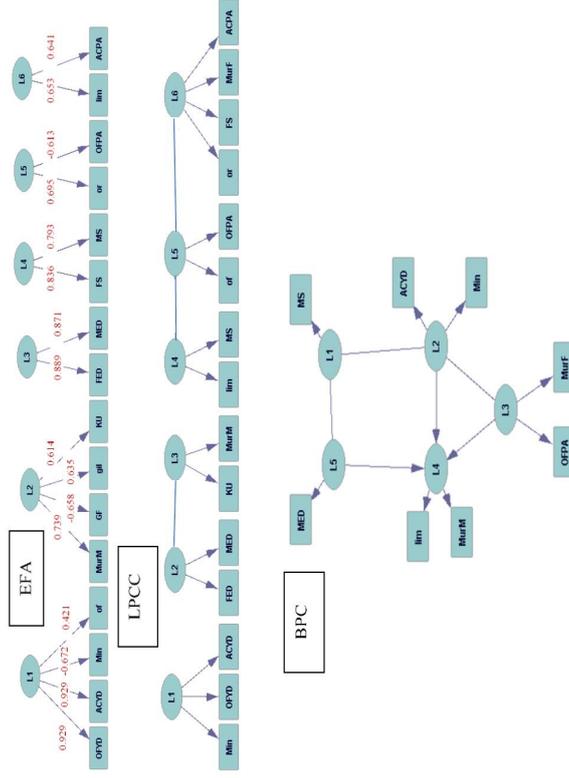


Figure 14: LVMs learned by EFA, LPCC, and BPC for DB2. Numbers for EFA represent the factor loadings, where plus and minus signs indicate positive and negative correlations, respectively.

Again, both the EFA and LPCC algorithms found six latent variables (Figure 14). L1 in EFA is interesting and very important because it links YD involvement in accidents and offenses with gender and motorcycle license. The loading coefficients show that male YDs, especially those who have a motorcycle license, are involved more in both accidents and offenses (in both cases with loadings of 0.929). L1 in LPCC shows a similar relation between gender and a YD's involvement in accidents and offenses, but without a relation to having a motorcycle license. Also in the EFA results, this last relation is quite weak, with a boundary loading value of 0.421, whereas the threshold for connecting an observed variable to a latent factor is 0.4 (which is a common practice in EFA). The relation found

ers between these cellular subpopulations. This observation is demonstrated clearly in the results obtained by BPC, EFA, and LPCC in Figures 15a, 15b, and 16a, respectively. LPCC and EFA managed to learn models with eight latents each and BPC a model with one latent, but none of the three latent variable models seems to be biologically meaningful, as judged by biological experts.

However, LPCC has an advantage over the other two algorithms because to advance learning an LVM, LPCC clusters the input data in its first stage. We exploit this advantage by improving clustering of the input to consider the domain specific properties. This act of clustering – that is natural to LPCC – coincides with the conventional clustering-based analysis of CyTOF data, which is mandatory to this domain because the data shows a hierarchical structure (as outlined below). Therefore, instead of using SOM, we initialized LPCC based on the clustering results obtained by Citrus (Bruggner et al., 2014). Citrus applies hierarchical clustering to the cell events; however, instead of cutting the hierarchy at an arbitrary height to identify the clusters, it uses a minimum cluster size threshold (we used a 1% threshold of the observations, i.e., 8,000 observations), for which only clusters larger than this threshold are selected. By selecting automatically and based on the data only large enough clusters, we preserve the requirement of LPCC of not determining the number of clusters arbitrarily and facilitate the avoidance of noisy clusters. In addition, we performed another purification procedure by selecting only clusters for which the ratio of the cluster marker entropy to the distribution marker entropy is smaller than a threshold.

Following this cluster purification procedure, LPCC found a five-latent variable model that is represented in Figure 16b. L1 is a parent of five markers that represent T cells (except for CD457 that may be expressed also by other leukocytes, and CD62L that is an activation marker that also can be expressed by T cells). L2 partially represents monocytes by having the markers CD11b and CD86 as its children; however, this representation is not perfect since it wrongly connects CD34, which is a phenotypic marker of stem cells. Thus, L2 may be representing monocytes that are antigen-representing cells excluding CD34. L3 and L4 are linked by a directed edge from L3 into L4 and together represent macrophage cells. Although both latents represent the same population of cells, LPCC correctly splits it into two latent variables since the children of L3 are expressed only by macrophages, but the children of L4 may also be expressed by monocytes, which are the macrophages' precursor. L5 is a parent of three markers that represent B cells together with another marker, IA-IE, which is an activation marker that can also be expressed by B cells. Still, the model learned by LPCC represents only sixteen of the 37 markers in this experiment. This may be explained by the high level of overlap and number of shared markers among the different subpopulations. Despite that, these results are encouraging and demonstrate a significant improvement compared to previous results obtained by BPC, EFA, and LPCC before cluster purification.

Analysis of cell sub-populations in the immune system naturally lends itself to the use of clustering methodologies, as immunologists traditionally resort to the classification of cells as belonging to cellular subpopulations. Cell subpopulations are usually defined by the stable expression of markers on said cells. Yet, not all markers capture protein expression that is stably expressed, and the expression of some proteins may be noisy or plastic,

⁷The cellular markers we use are well known in immunology (Janeway et al., 2001), and thus their description is avoided here for clarity of the demonstration.

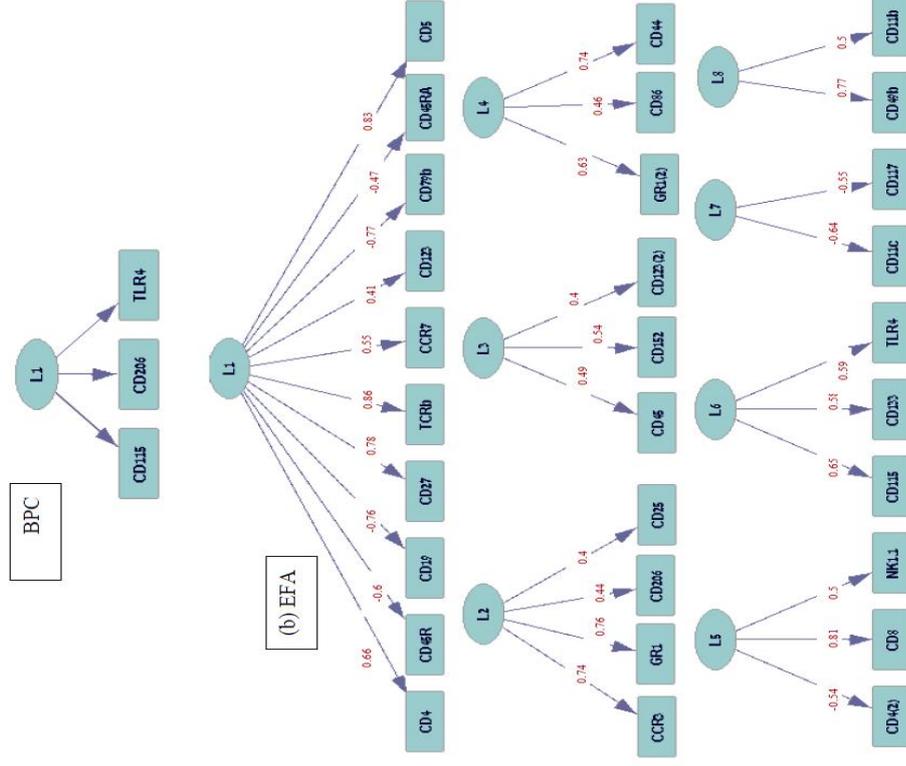


Figure 15: LVMs learned by (a) BPC and (b) EFA. Numbers for EFA represent the factor loadings, where plus and minus signs indicate positive and negative correlation, respectively.

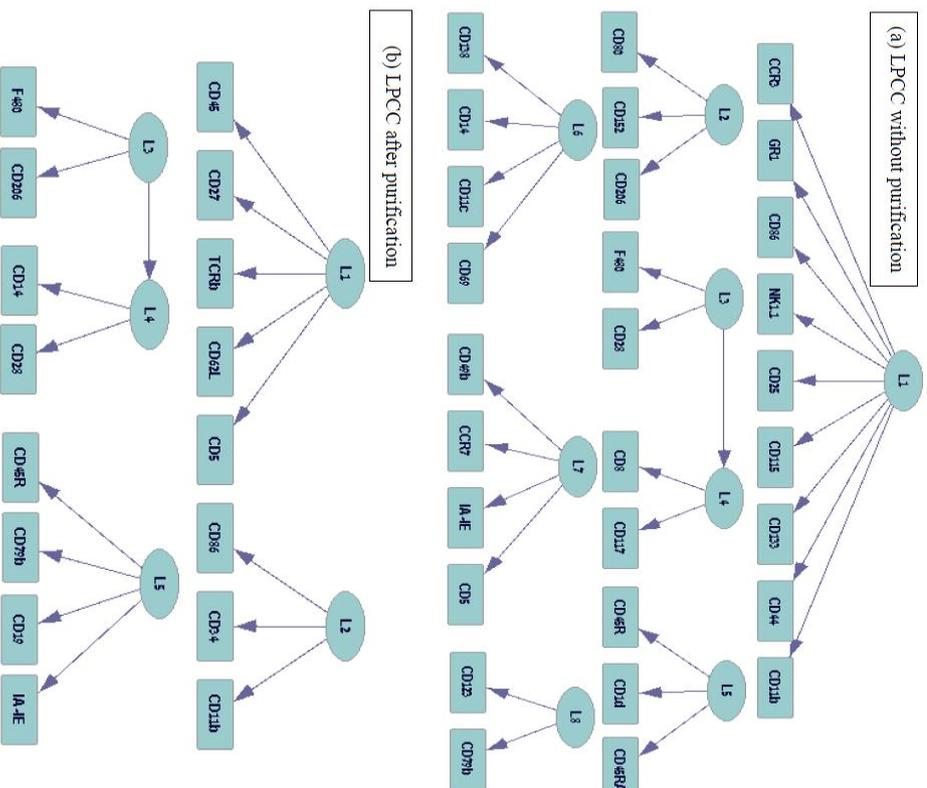


Figure 16: LVMs learned by LPCC (a) before and (b) after a purification procedure.

varying over time and conditions. With this in mind, clustering and noise filtration – two pre-processing steps of LPCC – provide great benefit in this context, yielding improved results. Specifically, the pre-processing step of clustering the data prior to LPCC provides an advantage as it compartmentalizes marker relationships to the context in which they matter, whereas the noise filtering step focuses the analysis to those markers whose relationship with one another may be meaningful.

4. Related Works

The traditional framework for discovering latent variables is factor analysis and its variants (e.g., see Bartholomew et al., 2002). This is, by far, the most common method used in several applied sciences (Glymour, 2002). However, a limitation of factor analysis is its level of subjectivity stemming from the many methodological decisions a researcher must make to complete an analysis, where the results of this analysis largely depend on the quality of these decisions (Henson and Roberts, 2006). Moreover, factor analysis and its variants provide only a limited ability in causal explanation (see Silva, 2005, and our evaluation section). Therefore, in this section, we will focus on related work in the framework of learning causal graphical models beyond the variants of factor analysis.

The main goal of heuristic methods such as those of Elidan et al. (2000) is the reduction of the number of parameters in a BN. The idea is to reduce the variance of the resulting density estimator, achieving better probabilistic predictions. For probabilistic modeling, the results described by Elidan et al. are a convincing demonstration of the suitability of their approach, which is intuitively sound. However, such heuristic methods provide neither formal interpretation of what the resulting structure is, nor explicit assumptions on how such latents should interact with the observed variables. Further, such heuristic methods do not provide an analysis of possible equivalence classes, and consequently, there is no search algorithm that can account for equivalence classes. Therefore, for a causality discovery under the assumption that multiple observed variables have hidden common causes, such as in MIM that is widely used in applied sciences, the results described by Elidan et al. are unsatisfying.

Unlike other algorithms (Pearl, 2000; Zhang, 2004; Harmeling and Williams, 2011; Wang et al., 2008), LPCC is suitable for learning MIM models and not just latent-tree models. This LPCC quality is shared by BPC (Silva et al., 2006). Both LPCC and BIN-A (Harmeling and Williams, 2011) apply clustering as a preprocessing step to learn latent models. But, LPCC applies clustering to the data points, whereas BIN-A clusters the variables using agglomerative hierarchical clustering, which is suitable to learn HLC models, as in Zhang (2004). LPCC provides a consistent and substantive analysis of data-point clustering using the PCC concept, and can learn all types of links between the latents; thus, unlike BIN-A, it is not limited to binary latent trees.

FCL (Spirites et al., 2000) is not comparable to LPCC in learning MIM models as illustrated for the political action survey and the Holzinger and Swinford databases (Sections 3.2 and 3.3). Compared to BPC and FCL, LPCC does not rely on statistical tests and pre-setting of a significance level for learning LVM.

Contrary to BPC, LPCC concentrates on the discrete case and dispenses with the linearity assumption. However, LPCC assumes that the measurement model is pure; still a

weaker assumption than the one latent-tree models make. Unlike LPCC, BPC is not suitable for learning models such as G1 in Figure 2, where the latents are independent and each has fewer than four observed children. This is because BPC requires the variables in a Tetrad constraint to all be mutually dependent, where in the case of G1, there are at most three mutually dependent variables, so no Tetrad constraint can be tested and no graph is learned (Section 3.1). In addition, BPC is not suitable for learning models such as the HIV model (Section 3.4), where each latent has only two indicators and BPC requires three indicators for a latent to be identified. This also explains the poor results of BPC on the YD databases compared to the LPCC results (Section 3.5).

When the attributes are categorical, cluster analysis is sometimes called latent class analysis (LCA) (Lazarsfeld and Henry, 1968; Goodman, 1974; Bartholomew and Knott, 1999), where data are assumed to be generated by a latent class model (LCM). An LCM consists of a class variable (latent) that represents the clusters to be identified and a number of other variables that represent attributes (observed variables) of objects.⁸ LCMs assume local independence; in other words, the observed variables are conditionally independent given the latent variable. A serious problem with the use of LCA, known as local dependence, is that the local independence assumption is often violated. To relax this strong assumption, Zhang (2004) proposed a richer, tree-structured latent variable model, specifically, the HLC model. The network structure is a rooted tree, and the leaves of the tree are the observed variables. HLC models were chosen for two reasons. First, the class of HLC models is significantly larger than the class of LCMs and can accommodate local dependence. Second, inference in an HLC model takes time that is linear in the model size (because it is a tree), which makes it computationally feasible to run EM. However, MIM models learned by LPCC are richer than HLC models that are only a subset of MIMs. Thus, LPCC may contribute to clustering analysis of data generated by richer models, while keeping the advantage of accommodating local dependence.

5. Discussion

In Part I, we introduced the PCC concept and LPCC algorithm for learning LVMs. We showed that LPCC: 1) Is not limited to latent-tree models, and does not make a linearity assumption about the distribution; 2) Learns MIMs; 3) Learns a MIM with no assumptions about the number of latent variables and their interrelations (except the assumption that a latent collider does not have any latent descendants; Assumption 5) and which observed variables are the children of which latents; and 4) Learns an equivalence class of the structural model of the true graph.

In Part II, we formally introduced the LPCC two-stage algorithm. First, LPCC learns the exogenous latents and the latent colliders, as well as their observed descendants, by utilizing pairwise comparisons between data clusters in the measurement space that may explain latent causes. Second, LPCC learns the endogenous latent non-colliders and their children by splitting these latents from their previously learned latent ancestors.

Using simulated and real-world data sets, we showed in Part II that LPCC improves accuracy with the sample size, can learn large LVMs, and has consistently good results

⁸This model has the same graphical structure as the naive-Bayes classifier, but because it is trained in an unsupervised manner (clustering), we refer to it as an LCM.

compared to models that are expert-based or learned by state-of-the-art algorithms. Using LPCC to identify possible causes of young drivers' involvement in road accidents, we found interesting relations among latent and observed variables and can provide illuminating insights into this important problem. Using LPCC to identify cell subpopulations in the immune system, we offer an LVM that makes sense to expert biologists in describing this challenging system. A criticism of LPCC may be its reliance on performing preliminary clustering to the data. Changes in the data used for clustering may affect the LPCC output. Yet, our experience shows that even if the clustering results change for different data samples drawn from the distribution, the same major and 1-order minor clusters are usually identified. In addition, as the biological example (Section 3.6) illustrates, when a structure is inherent to the data, clustering of the data first yields high benefit in learning an LVM later and improves results. Structured real-life problems are prevalent in many disciplines (Vazquez et al., 2004); hence, being a clustering-based LVM learning mechanism gives LPCC an advantage more than a disadvantage.

Finally, a number of open problems that invite further research were provided in the discussion of Part I.

Acknowledgments

Special thanks to the *Israel National Road Safety Authority* and *Ran Naor Foundation for the Advancement of Road Safety Research* for supporting the research to explain young drivers' involvement in road accidents (Section 3.5) and to Shai Shen-Orr, Elina Starostevsky, and Tania Dubovik from the Department of Immunology at the Technion for providing the mass-cytometry data set and helping in the evaluation of the results of its analysis (Section 3.6). The authors also thank the two anonymous reviewers for their comments and suggestions that helped improve the paper and the special issue editors: Isabelle Guyon and Alexander Statnikov. Nuaman Asbeh thanks the *Planning and Budgeting Committee* (PBC) of the *Israel Council for Higher Education* for its support through a scholarship for distinguished Ph.D. students.

Appendix A. Important assumptions, definitions, propositions, and theorems from Part I (numbers are taken from Part I)

Assumption 5 *A latent collider does not have any latent descendants (and thus cannot be a parent of another latent collider).*

Definition 12 *The single cluster that corresponds to the observed major value configuration, and thus also represents the major effect MAE (ex) due to configuration ex of EX, is the major cluster for ex, and all the clusters that correspond to the observed minor value configurations due to minor effects in MIES (ex) are minor clusters.*

Definition 13 *A k-order minor effect is a minor effect in which exactly k endogenous variables in EN correspond to minor local effects. An en corresponding to a k-order minor effect is a k-order minor value configuration.*

Definition 14 Minor clusters that correspond to k -order minor effects are k -order minor clusters.

Definition 15 Pairwise cluster comparison is a procedure by which pairs of clusters are compared, for example through a comparison of their centroids. The result of PCC between a pair of cluster centroids of dimension $|\mathbf{O}|$, where \mathbf{O} is the set of observed variables, can be represented by a binary vector of size $|\mathbf{O}|$ in which each element is 1 or 0 depending, respectively, on whether or not there is a difference between the corresponding elements in the compared centroids.

Definition 16 A maximal set of observed (MSO) variables is the set of variables that always changes its values together in each major-major PCC in which at least one of the variables changes value.

Definition 18 2S-PCC is PCC between 1-MC and a major cluster that shows two sets of two or more elements corresponding to the observed variables. Elements in each set have the same value, which is different than that of the other set. Accordingly, this 1-MC is defined as 2S-MC.

Definition 19 A 2S-MSO is the maximal set of observed variables that always change their values together in all 2S-PCCs.

Proposition 10 In 2S-PCCs in which only the observed children of a single latent change, the latent is

1. EX or its leaf latent non-collider descendant, if the connection is serial; or
2. EX's leaf latent non-collider descendant, if the connection is diverging.

Theorem 1 Variables of a particular MSO are children of a particular exogenous latent variable EX or its latent non-collider descendant or children of a particular latent collider C.

Theorem 2 A latent variable L is a collider of a set of latent ancestors LACEX only if:

1. The values of the children of L change in different parts of some major-major PCCs each time with the values of descendants of another latent ancestor in LA; and
2. The values of the children of L do not change in any PCC unless the values of descendants of at least one of the variables in LA change too.

Theorem 3 Variables of a particular 2S-MSO are children of an exogenous latent variable EX or any of its descendant latent non-colliders NC.

Theorem 4 A latent non-collider NC1 is a direct child of another latent non-collider NC2 (both on the same path emerging in EX) only if:

- In all 2S-PCCs for which EX does not change, the observed children of NC1 always change with those of NC2 and also in a single 2S-PCC without the children of NC2; and
- In all 2S-PCCs for which EX does not change, the observed children of NC1 always change, the observed children of NC2 always change with those of NC1 and also in a single 2S-PCC without the children of NC1.

Appendix B. Additional results for the simulated data (Section 3.1)

B.1 LPCC compared to BPC

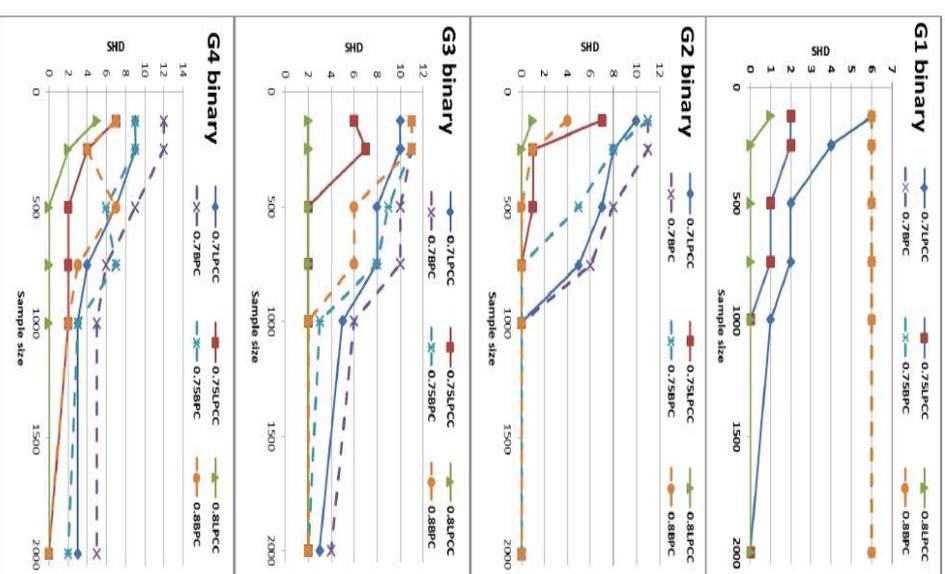


Figure 17: SHD learning curves of LPCC compared with those of BPC for G1–G4 of Figure 2 with binary variables, three parameterization levels, and increasing sample sizes.

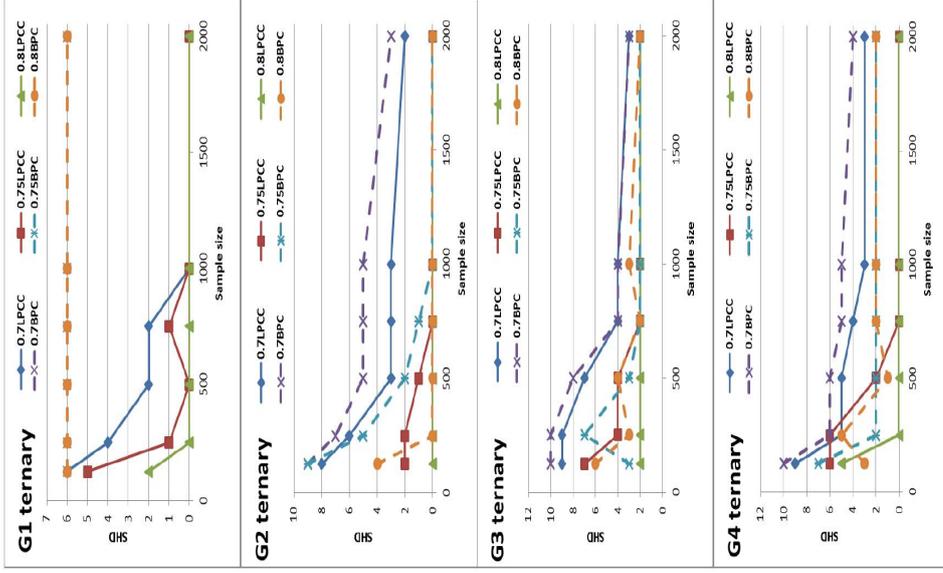


Figure 18: SHD learning curves of LPCC compared with those of BPC for G1–G4 of Figure 2 with ternary variables, three parameterization levels, and increasing sample sizes.

B.2 LPCC compared to EFA

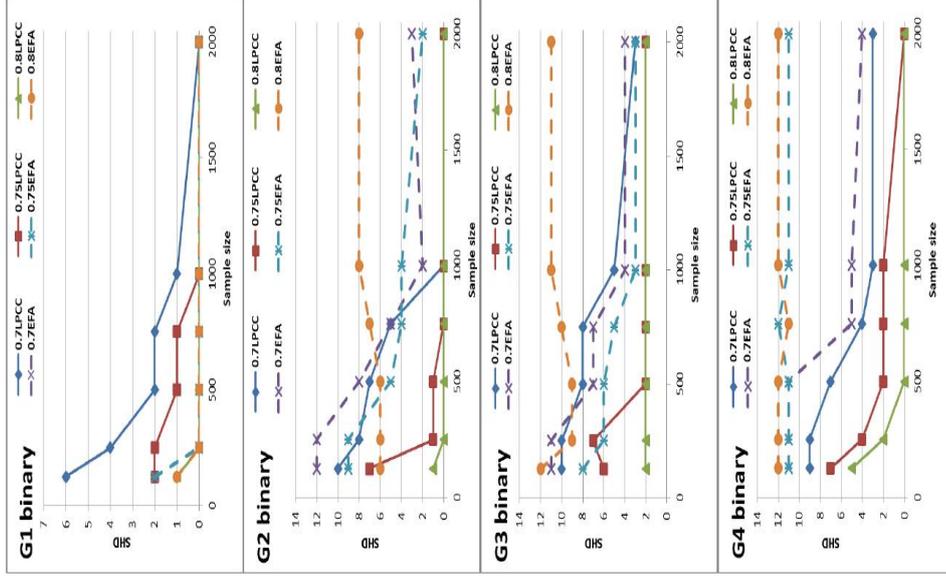


Figure 19: SHD learning curves of LPCC compared with those of EFA for G1–G4 of Figure 2 with binary variables, three parameterization levels, and increasing sample sizes. The lines of LPCC and EFA for a parametrization of 0.8 coincide for G1.

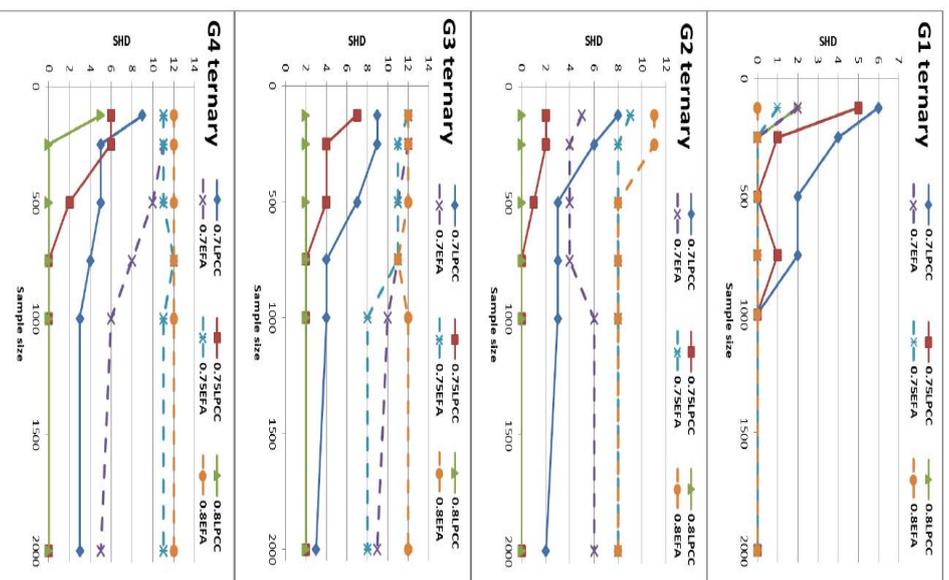


Figure 20: SHD learning curves of LPCC compared with those of EFA for G1–G4 of Figure 2 with ternary variables, three parameterization levels, and increasing sample sizes. The line of LPCC for a parameterization of 0.8 coincides with that of EFA for a parameterization of 0.7 for G1.

Appendix C. PCC analysis for two example databases

C.1 Results for the political action survey data (Section 3.2)

We applied clustering analysis to the political action survey data using SOM having 250 unit map size (similar results were obtained for SOMs having 125 and 500 unit map sizes). U-matrix visualization⁹ of the SOM result is given in Figure 21. As presented in Table 2, nine clusters were found, and since four clusters are larger than the average cluster size of 45, only four of the nine clusters are major. Table 3 shows PCCs between these four major clusters. Note that NOSAY and VOTING always change together in all PCCs in which either of them changes, and this is also the case for NOCARE, TOUCH, and INTEREST. Therefore, LPCC found two latents (Figure 8 b): One (Efficacy) corresponds to NOSAY and VOTING and the other (Responsiveness) corresponds to NOCARE, TOUCH, and INTEREST.

Centroid	NOSAY	VOTING	COMPLEX	NOCARE	TOUCH	INTEREST
C1(86)	3	3	2	3	3	3
C2(60)	2	2	2	2	2	2
C3(57)	3	3	2	2	2	2
C4(49)	3	3	3	3	3	3
C5(39)	3	2	2	2	2	2
C6(31)	3	3	2	3	2	2
C7(31)	3	2	3	3	3	3
C8(28)	1	1	1	1	1	1
C9(27)	3	2	2	3	3	3

Table 2: Nine clusters are represented by their centroids for the political action survey data. Cluster sizes are in parentheses. The first four clusters are major.

PCC	δ NOSAY	δ VOTING	δ COMPLEX	δ NOCARE	δ TOUCH	δ INTEREST
PCC1,2	1	1	0	1	1	1
PCC1,3	0	0	0	1	1	1
PCC1,4	0	0	1	0	0	0
PCC2,3	1	1	0	0	0	0
PCC2,4	1	1	1	1	1	1
PCC3,4	0	0	1	1	1	1

Table 3: PCCs between the four major clusters for the political action survey data.

⁹The U-matrix is a widely used visualization of SOM. It computes (for each unit in the SOM) the mean of the distance measures between neighbors. By plotting this data on a 2D map using a color scheme, we can visualize a landscape with walls (red areas) and valleys (blue areas). The walls separate different clusters; they represent extreme distances between neighboring units, whereas patterns mapped to units in the same valley are similar and belong to the same cluster (Ultsch et al., 1993).

C.2 Results for Holzinger and Swineford's data (Section 3.3)

We applied clustering analysis to the European Values Survey (Holzinger and Swineford's) data set using SOM having a 100 map size. Fifteen clusters were found (Table 4), and since the average cluster size is 7.4, four clusters are major. Based on major-major PCCs (Table 5), one learned latent (Spatial) corresponds to VisPerc and Lozenges (that always change together in all PCCs in which either of them changes) and the other latent (Verbal) corresponds to ParComp, SenComp, and WordMean (Figure 10). It is interesting to see that in half of the PCCs in which VisPerc and Lozenges change together, Cubes also changes. If Cubes would have changed in all PCCs in which VisPerc and Lozenges change together, then it would be found as Spatial's child, and the learned model would be exactly the theoretical model (Figure 10a). This did not happen, probably because the PCCs in which Cubes did not change with VisPerc and Lozenges relate to relatively small clusters.

Centroid	VisPerc	Cubes	Lozenges	ParComp	SenComp	WordMean
C1(19)	1	1	1	1	1	1
C2(18)	2	2	2	2	2	2
C3(9)	1	2	1	1	1	1
C4(8)	2	2	2	1	1	1
C5(7)	1	1	1	1	1	2
C6(7)	2	1	2	1	1	1
C7(7)	1	1	2	1	1	1
C8(6)	2	1	2	2	2	2
C9(6)	2	1	1	1	1	1
C10(5)	2	1	1	2	2	2
C11(5)	1	2	2	2	2	2
C12(4)	1	2	2	2	1	2
C13(4)	2	2	1	2	2	2
C14(3)	1	1	1	2	2	2
C15(3)	2	1	1	1	2	2

Table 4: Fifteen clusters are represented by their centroids for Holzinger and Swineford's data. Cluster sizes are in parentheses. The first four clusters are major.

PCC	$\delta VisPerc$	$\delta Cubes$	$\delta Lozenges$	$\delta ParComp$	$\delta SenComp$	$\delta WordMean$
PCC1,2	1	1	1	1	1	1
PCC1,3	0	1	0	0	0	0
PCC1,4	1	1	1	0	0	0
PCC2,3	1	0	1	1	1	1
PCC2,4	0	0	0	1	1	1
PCC3,4	1	0	1	0	0	0

Table 5: PCCs between the four major clusters for Holzinger and Swineford's data.

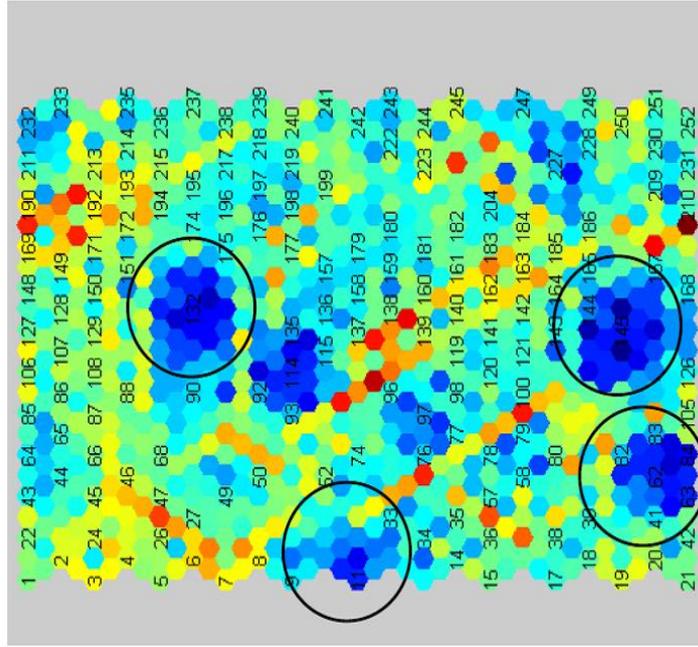


Figure 21: U-matrix visualization of a 250 unit map size SOM (numbers on the map represent the SOM units) obtained for the political action survey data. Vectors that were mapped to the same map unit belong to the same valley (blue area) and the same cluster. Of the nine clusters that were found (Table 2), four are major (circled) and the remaining are minor.

References

- J. L. Arbuckle. *Amos Users' Guide Version 3.6*. Small Waters Corporation, Chicago, IL, 1997.
- D. J. Bartholomew and M. Knott. *Latent Variable Models and Factor Analysis*. Kendall's Library of Statistics 7. Arnold Press, London, United Kingdom, 2nd edition, 1999.
- D. J. Bartholomew, F. Steele, I. Moustaki, and J. I. Galbraith. *The Analysis and Interpretation of Multivariate Data for Social Scientists (Texts in Statistical Science Series)*. Chapman & Hall/CRC Press, Boca Raton, Florida, USA, 2002.
- R. V. Bruggner, B. Bodenmiller, D. L. Dill, R. J. Tibshirani, and G. P. Nolan. Automated identification of stratifying signatures in cellular subpopulations. In *Proceedings of the National Academy of Sciences of the United States of America*, pages 111(126):E2770–E2777, 2014.
- A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society, B* 39:1–39, 1977.
- G. Eldan, N. Lotner, N. Friedman, and D. Koller. Discovering hidden variables: A structure-based approach. In *Advances in Neural Information Processing Systems*, pages 13:479–485, 2000.
- C. Glymour. *The Mind's Arrow: Bayes Nets and Graphical Causal Models in Psychology*. MIT Press, Cambridge, Massachusetts, 2002.
- L. A. Goodman. Exploratory latent structure analysis using both identifiable and unidentifiable models. *Biometrika*, 61:215–231, 1974.
- S. Harnelling and C. K. I. Williams. Greedy learning of binary latent trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:1087–1097, 2011.
- R. K. Henson and J. K. Roberts. Use of exploratory factor analysis in published research. Common errors and some comment on improved practice. *Educational and Psychological Measurement*, 66:393–416, 2006.
- K. J. Holzinger and F. Swinford. A study in factor analysis: The stability of a bifactor solution. Technical Report 48, Supplementary Educational Monographs, University of Chicago Press, Illinois, 1939.
- C.A. Janeway, P. Travers, M. Walport, and M. Schlomchick. *Immunobiology*. Garland Publishing, New York, 5th edition, 2001.
- K. Joreskog. Structural equation modeling with ordinal variables using LISREL. Technical report, Scientific Software International Inc, 2004.
- K. G. Joreskog and D. Sorbom. *Lisrel 7: A Guide to the Program and Applications*. SPSS, Chicago, 1989.
- H. F. Kaiser. The application of electronic computers to factor analysis. *Educational and Psychological Measurement*, 20:141–151, 1960.
- T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, New York, New York, 1997.
- S. L. Lauritzen. The EM algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis*, 19:191–201, 1995.
- P. F. Lazarsfeld and N. W. Henry. *Latent Structure Analysis*. Houghton Mifflin, Boston, Massachusetts, 1968.
- B. Lerner. Young drivers' crash involvement, involvement prediction, and evaluation of the impact of Or Yarok kit on the involvement using machine learning. Technical report, Ran Naor Institute for the Advancement of Road Safety Research, 2012. http://www.rannaort.org.il/Young_Novice_Drivers_Researches.
- B. Lerner and J. Meyer. Identification of factors that account for young drivers' crash involvement and involvement prediction using machine learning. Technical report, Israel National Road Safety Authority, 2012. <http://www.ResearchAndSurveys/Pages/YoungDrivers.aspx>.
- B. F. J. Manly. *Multivariate Statistical Methods. A Primer*. Chapman & Hall, London, United Kingdom, 1994.
- J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, New York, New York, 2000.
- R. Silva. *Automatic Discovery of Latent Variable Models*. PhD thesis, Carnegie Mellon University, 2005.
- R. Silva, R. Scheines, C. Glymour, and P. Spirtes. Learning the structure of linear latent variable models. *Journal of Machine Learning Research*, 7:191–246, 2006.
- P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction and Search*. MIT Press, New York, New York, 2nd edition, 2000.
- I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65:31–78, 2006.
- A. Ullsch, G. Guimarães, D. Korus, and H. Li. Knowledge extraction from artificial neural networks and applications. In *Proceedings of Transputer-Anwender-Treffen/World-Transputer-Congress*, pages 194–203, Aachen, Germany, 1993.
- A. Vazquez, R. Dobrin, D. Sergi, J.-P. Eckmann, Z.N. Oltvai, and A.-L. Barabasi. The topological relationship between the large-scale attributes and local interaction patterns of complex networks. In *Proceedings of the National Academy of Sciences of the United States of America*, pages 101(52):17940–17945, 2004.
- J. Vesanto, J. Himberg, E. Alhoniemi, and J. Parhankangas. SOM toolbox for Matlab 5. Technical Report A57, Helsinki University of Technology, Helsinki, Finland, 2000.

- Y. Wang, N. L. Zhang, and T. Chen. Latent-tree models and approximate inference in Bayesian networks. *Journal of Artificial Intelligence Research*, 32:879–900, 2008.
- J. Wishart. Sampling errors in the theory of two factors. *British Journal of Psychology*, 19: 180–187, 1928.
- N. Zhang. Hierarchical latent class models for cluster analysis. *Journal of Machine Learning Research*, 5:697–723, 2004.

Integrative Analysis using Coupled Latent Variable Models for Individualizing Prognoses

Peter Schulam

Suchi Saria

Department of Computer Science

Johns Hopkins University

Baltimore, MD 21218, USA

PSCHULAM@CS.JHU.EDU

SSARIA@CS.JHU.EDU

Editor: Benjamin M. Marlin and C. David Page

Abstract

Complex chronic diseases (e.g., autism, lupus, and Parkinson's) are remarkably heterogeneous across individuals. This heterogeneity makes treatment difficult for caregivers because they cannot accurately predict the way in which the disease will progress in order to guide treatment decisions. Therefore, tools that help to predict the trajectory of these complex chronic diseases can help to improve the quality of health care. To build such tools, we can leverage clinical markers that are collected at baseline when a patient first presents and *longitudinally* over time during follow-up visits. Because complex chronic diseases are typically systemic, the longitudinal markers often track disease progression in multiple organ systems. In this paper, our goal is to predict a function of time that models the future trajectory of a single target clinical marker tracking a disease process of interest. We want to make these predictions using the histories of many related clinical markers as input. Our proposed solution tackles several key challenges. First, we can easily handle irregularly and sparsely sampled markers, which are standard in clinical data. Second, the number of parameters and the computational complexity of learning our model grows linearly in the number of marker types included in the model. This makes our approach applicable to diseases where many different markers are recorded over time. Finally, our model accounts for latent factors influencing disease expression, whereas standard regression models rely on observed features alone to explain variability. Moreover, our approach can be applied dynamically in continuous-time and updates its predictions as soon as any new data is available. We apply our approach to the problem of predicting lung disease trajectories in scleroderma, a complex autoimmune disease. We show that our model improves over state-of-the-art baselines in predictive accuracy and we provide a qualitative analysis of our model's output. Finally, the variability of disease presentation in scleroderma makes clinical trial recruitment challenging. We show that a prognostic tool that integrates multiple types of routinely collected longitudinal data can be used to identify individuals at greatest risk of rapid progression and to target trial recruitment.

Keywords: gaussian processes, conditional random fields, prediction of functional targets, latent variable models, disease trajectories, precision medicine

1. Introduction

In complex chronic diseases (CCD) such as autism, lupus, and Parkinson's, the way the disease manifests may vary greatly across individuals. This makes treatment challenging

because caregivers cannot easily predict an individual's future trajectory to guide therapy decisions. For example, in scleroderma, an autoimmune disorder, lung disease is a common cause of morbidity and mortality (Varga et al., 2012), but there are no known biomarkers or precise algorithms for stratifying individuals into groups based on similar lung disease course. A tool that can provide accurate forecasts of disease progression can help clinicians to tailor treatments to each patient based on their most likely course.

To monitor disease progression, clinicians collect many clinical markers both at baseline when an individual first visits the clinic and *longitudinally* during routine follow-up visits. Many CCDs are systemic, and so the markers are designed to monitor the disease's impact across many organ systems. In scleroderma, individuals may be affected across six organ systems—the lungs, heart, skin, gastrointestinal tract, kidneys, and vasculature—to varying extents (Varga et al., 2012). Example clinical markers include PFVC (percent of predicted forced vital capacity), which is used to measure lung damage severity; TSS (total skin score), which is used to measure skin disease activity; and, PDLCO (percent of carbon monoxide diffused by the lung), used to measure vasculature health.

Our goal is to predict a function of time that models the future trajectory of a single target clinical marker tracking a disease process of interest. We want to make these predictions by leveraging baseline information and additional time-dependent clinical markers (henceforth referred to as auxiliary markers) as they are collected. This is the focal challenge of personalized medicine: integrative analysis of heterogeneous data from an individual's medical history to improve care (Collins and Varmus, 2015). So far, efforts in integrative analysis have focused on combining inferences from molecular data modalities (Rosenbloom et al., 2013). Our focus in this paper is on leveraging routinely recorded information from the electronic health record—both static and time-dependent—to make precise estimates of an individual's disease course.

A key challenge in this setting is that these data are collected during routine clinical visits and therefore they are sparse and irregularly sampled. Predicting an individual's future disease is commonly framed as a regression problem where the target clinical marker at a specific time in the future is modeled as a function of observed input features alone. These features are computed by generating summaries from the observed data (e.g., the last PFVC value or the trend in the PFVC over the last six months). However, training conditional models is less straightforward from data where varying numbers of repeated measurements are sampled per patient and across different markers. In this setting, others have focused on dynamical prediction (e.g., Rizopoulos and Ghosh, 2011; Proust-Lima et al., 2014) by fitting parametric models to the longitudinal data and using the resulting model parameters as features for prediction. But existing formulations do not scale to high-dimensional problems with many auxiliary markers.

Another key challenge in predicting disease trajectories in CCDs is that differences in trajectories across individuals may be largely due to factors that are not yet known. For example, different disease pathways or biological mechanisms (e.g., genetic mutations or autoimmune markers) may be driving different subtypes of the disease (Lewis et al., 2005; Lötvald et al., 2011; Doshi-Velez et al., 2014; Saria and Goldenberg, 2015), each associated with distinct disease trajectories (Schulam et al., 2015). But, in many diseases, our knowledge of these pathways is, at best, limited. In this setting, Schulam et al. (2015) use a latent variable model to infer subtypes—subgroups with similar trajectories—using

repeated measurements of clinical marker data in the electronic health record. Schulam and Saria (2015) extend these ideas and introduce a transfer learning framework for predicting individual-specific disease trajectories that accounts for subtypes and other latent factors causing heterogeneity in disease expression. These works, however, focus on modeling single marker trajectories. We build on Schulam and Saria (2015) in this paper.

1.1 Contributions

In this paper, we describe a scalable framework for predicting a target marker trajectory (i.e. a continuous-time function) that allows us to use multiple longitudinal clinical marker histories as inputs. Our approach makes it easy to handle irregular sampling patterns across markers. Because we use a discriminative training criterion that conditions on marker histories instead of jointly modeling them, the framework is not as sensitive to misspecified dependencies across marker types. Moreover, the number of parameters and computational complexity scales linearly with the number of markers, which makes it possible to apply our approach in high-dimensional settings where many different marker types are available. Finally, our approach aligns with the dynamical nature of clinical medicine; it can be used to make predictions using continuously growing marker histories. We apply our approach to the problem of predicting lung disease trajectories in scleroderma, a complex autoimmune disease. We show that our model improves over state-of-the-art baselines in predictive accuracy and we provide a qualitative analysis of our model’s output. Moreover, we demonstrate the clinical utility of our model by measuring performance on early detection of individuals who develop aggressive lung disease.

2. Related Work

Most predictive models used in medicine are cross-sectional—they use features from data measured up until the current time to predict a clinical marker or outcome at a fixed point in the future. As an example, consider the mortality prediction model by Lee et al. (2003), where logistic regression is used to integrate features into a prediction about the probability of death within 30 days for a given patient. To predict the outcome at multiple time points, it is common to fit separate models (e.g., Wang et al. 2012; Zhou et al. 2011). These models are trained to use features extracted from a fixed-size window, rather than a dynamically growing history. Moreover, they tackle heterogeneity in a limited way—any differences across individuals must be explained by observed features alone.

A common approach to dynamical prediction of trajectories is to use Markov models such as order- p autoregressive models (AR- p), HMMs, state space models, and dynamic Bayesian networks (e.g. Hassan and Nath 2005; Quinn et al. 2009; Murphy 2002). While such models naturally make dynamic predictions using the full history by forward-filtering, they typically assume discrete, regularly-spaced observation times.

To model an individual’s disease trajectory using sparse and irregularly sampled clinical markers, we draw heavily from ideas in the functional data analysis (FDA) literature (see e.g., Ramsay 2006). In FDA, sequences of measurements are assumed to be samples from an underlying continuous function. A common first-step in FDA is to project the irregular observations on to a functional basis, such as B-splines, and then analyze the time series in coefficient space. However, coefficient estimates can have high variance when a time series

has too few observations, which is common in clinical data. James and Sugar (2003) address this issue by modeling the parameters of individual trajectories as random variables with a low-rank parameterization of the mean and covariance. This work is closely related to ours, and the idea of sharing statistical strength across trajectories through a structured prior over individual-specific parameters is used broadly throughout trajectory analysis to account for sparsity. Gaussian processes (GPs) are also commonly used in FDA: they offer flexible nonparametric models of trajectories but can also help to counteract sparsity by sharing kernel hyperparameters across individuals—see Roberts et al. (2013) for a recent review of GPs applied to time series data. Recent work by Liu and Hauskrecht (2014) combines the advantages of Markov models (e.g. AR processes and state space models) and Gaussian processes to make predictions of clinical laboratory test results. To account for variability in collections of functions, a number of authors have proposed variants of GPs that account for variability in the mean function (e.g. Lázaro-Gredilla et al. 2012; Shi et al. 2012) and the covariance function (e.g. Shi et al. 2005). Another related line of work in the FDA literature is function-to-function regression (e.g., Oliva et al. 2015). In most approaches to function-to-function regression (FFR) the input and output are defined on fixed domains. In contrast, our problem requires updated predictions as the clinical history continues to grow; both the input and output domains are therefore constantly changing.

Most related to our work is that by Rizopoulos (2011), where the focus is on making dynamical predictions about a time-to-event outcome (e.g. time until death) using all previously observed values of a longitudinally recorded marker. As more data is collected, they dynamically update posterior distributions over individual-specific longitudinal model parameters (as is done in FDA), which serve as time-varying features for the time-to-event prediction. Proust-Lima et al. (2014) tackles the same task but uses a mixture of trajectories to model longitudinal data. As more observations are collected, the posterior over a set of classes is updated, each of which has a distinct set of time-to-event model parameters. These are both state-of-the-art models for the task of dynamical disease trajectory prediction; we will revisit them in our experimental section where we use the approaches as baselines. To scale these models to multivariate time series, however, requires careful specification of the joint model across different markers, which can be challenging in high-dimensional settings (e.g., Dürichen et al. 2015) and may be difficult to scale. For example, Rizopoulos and Ghosh (2011) use a random effects model with a full covariance matrix to describe dependencies across markers, which scales quadratically in the number of marker types (as opposed to linearly as is the case for C-LTM). Because C-LTM is discriminatively trained (we optimize the likelihood of future target trajectories given target and auxiliary marker histories), it is less sensitive to misspecification of the dependencies across markers.

3. Coupled Latent Trajectory Model

Our goal is to predict a *continuous function* modeling the future trajectory of a *target clinical marker* (e.g. PFVC) that tracks disease progression in a specific organ. To make our predictions, we will use a collection of baseline (i.e. static) markers measured when an individual first visits the clinic, the previously observed values of the target marker, and the previously observed values of a collection of *auxiliary clinical markers* tracking related organ systems. See Figure 6a-d for example applications. In these figures, the posterior

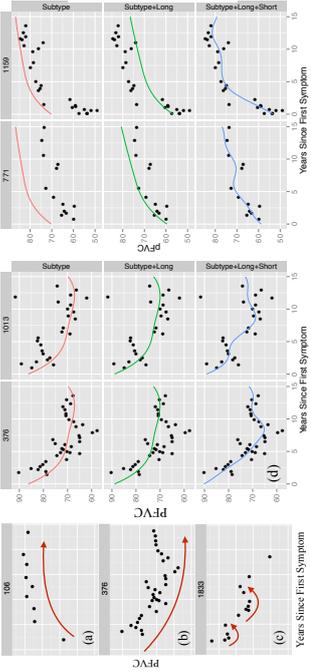


Figure 1: Plots (a-c) show example marker trajectories. Plot (d) shows four individuals with adjustments to a population and subpopulation fit (row 1). Row 2 makes an individual-specific long-term adjustment. Row 3 makes individual-specific short-term adjustments. To simplify, we only show mean functions; posterior uncertainty intervals are omitted.

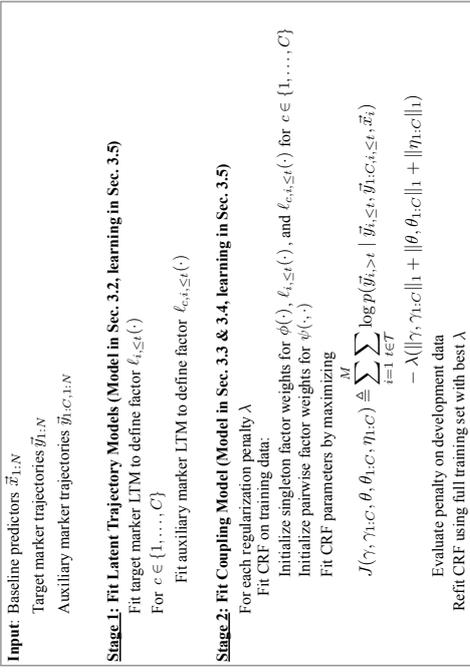


Figure 2: Two-stage procedure for fitting the Coupled Latent Trajectory Model (C-LTM).

distribution over the PFVC values (blue and green shaded regions) are conditioned upon baseline markers (e.g. gender and race), the observed PFVC values (black points), and auxiliary marker histories (e.g. TSS). We learn our model from a database of clinical histories of individuals, which are comprised of the individuals' baseline information and irregularly sampled trajectories of both the target and auxiliary markers. Formally, our

model will estimate the following conditional distribution (notation is described in the subsequent paragraph):

$$\mathcal{D}(i, t) \triangleq p(\mathbf{y}_i(\cdot) \mid \vec{y}_{i \leq t}, \vec{y}_{1:C,i \leq t}, \vec{x}_i). \quad (1)$$

Notation. For an individual i , we denote each target marker observation using y_{ij} and its measurement time using t_{ij} where $j \in \{1, \dots, N_i\}$. We use $\vec{y}_i \in \mathbb{R}^{N_i}$ and $\vec{t}_i \in \mathbb{R}^{N_i}$ to denote all of individual i 's marker values and measurement times respectively. We assume that the target marker observations are noisy observations of a latent continuous-time function (the trajectory), which we denote using $\mathbf{y}_i(\cdot)$. Each individual has baseline (static) information collected into a vector, which we denote using \vec{x}_i . We use C to denote the number of auxiliary marker types, N_{ci} to denote the number of observations of the c^{th} type, and use y_{cij} and t_{cij} to denote individual i 's j^{th} measurement of marker type c . We use $\vec{y}_{ci} \in \mathbb{R}^{N_{ci}}$ and $\vec{t}_{ci} \in \mathbb{R}^{N_{ci}}$ to denote the vector containing all of individual i 's c^{th} marker values and times respectively. We will also frequently need to refer to the vector of marker values observed up until a time t , which we denote using $\vec{y}_{i \leq t}$ ($\vec{y}_{ci \leq t}$ for auxiliary markers). Similarly, for markers observed after a time t , we use $\vec{y}_{i > t}$ ($\vec{y}_{ci > t}$ for auxiliary markers). The term $\vec{y}_{1:C,i \leq t}$ refers to all auxiliary markers measured on individual i up until time t .

At a high-level, we will model Eq. 1 by first assuming that each clinical marker trajectory (both target and auxiliary) can be d-separated (rendered conditionally independent) of all other marker types given a marker type-specific latent variable. We denote these latent variables using z_i for the target marker and z_{ci} for auxiliary marker c , and will describe them further later in this section. Under this assumption, we can write Eq. 1 as

$$\begin{aligned} \mathcal{D}(i, t) &= \sum_{z_i} p(\mathbf{y}(\cdot) \mid z_i, \vec{y}_{i \leq t}, \vec{x}_i) p(z_i \mid \vec{y}_{i \leq t}, \vec{y}_{1:C,i \leq t}, \vec{x}_i) \\ &\propto \sum_{z_i} p(\mathbf{y}(\cdot) \mid z_i, \vec{y}_{i \leq t}, \vec{x}_i) p(\vec{y}_{i \leq t} \mid z_i, \vec{x}_i) p(z_i \mid \vec{y}_{1:C,i \leq t}, \vec{x}_i) \\ &\propto \sum_{z_i} p(\mathbf{y}(\cdot) \mid z_i, \vec{y}_{i \leq t}, \vec{x}_i) \underbrace{p(\vec{y}_{i \leq t} \mid z_i, \vec{x}_i)}_{\text{LTM predictive, Section 3.2.2, Eq. 24}} \sum_{z_{1:C,i}} \underbrace{p(z_i, z_{1:C,i} \mid \vec{x}_i)}_{\text{Coupling Model, Section 3.3, Eq. 25}} \prod_{c=1}^C \underbrace{p(\vec{y}_{ci \leq t} \mid z_{ci}, \vec{x}_i)}_{\text{LTM likelihood, Section 3.2.1, Eq. 16}} \end{aligned} \quad (2)$$

We will learn this parameterization of $\mathcal{D}(i, t)$ in two stages. The models for the target and each of the auxiliary markers are learned independently during the first stage; using these, the LTM predictive and likelihood terms can be computed in Eq. 2. We treat the target and auxiliary markers as instances of the Latent Trajectory Model (LTM), which we review in Section 3.2. We emphasize, however, that any other generative model can be used if better suited to the domain. The coupling model is learned in the second stage, and is described in Section 3.3. We refer to the model created by combining these components as the Coupled Latent Trajectory Model (C-LTM), which we describe in Section 3.4. An overview of the procedure used to fit the C-LTM is shown in Figure 2.

3.1 Preliminaries

The Latent Trajectory Model (LTM) uses B-splines to model longitudinal trajectories, and our coupling model uses a conditional random field (CRF). We briefly introduce these two concepts, and point to resources where the interested reader can find additional details.

3.1.1 B-SPLINES

A common approach to fitting nonlinear functions of time while maintaining a linear dependence on model parameters is to use a basis expansion. Such an expansion defines some non-linear function $f(t)$ as a linear combination of other functions $\phi_1(t), \dots, \phi_d(t)$:

$$y = f(t | \beta) = \sum_{i=1}^d \beta_i \phi_i(t) = \Phi^\top(t) \bar{\beta}, \quad (3)$$

where ϕ_1, \dots, ϕ_d act as bases in some vector space of nonlinear functions and $\Phi(t) \in \mathbb{R}^d$ is the vector containing the values of the p basis functions evaluated at time t . The benefit of this formulation is that the function f is linear in the model parameters β , making it relatively easy to fit complex models. B-splines are a particular family of basis functions that we can use to parameterize nonlinear functions. Others include polynomial bases and radial basis functions. However, there are two advantages to using B-splines. First, each basis function is non-zero only over a compact interval of the real line, which improves statistical stability and also allows for computational speed ups that take advantage of sparse basis matrices (Gelman et al., 2014). This is in contrast to polynomials, where each basis takes non-zero values globally. The second advantage is that the family of functions parameterized by B-splines are not infinitely differentiable (in contrast to radial basis functions) and therefore not smooth (Gelman et al., 2014). This bias is often helpful in modeling functions from the real-world that arise from non-smooth processes. Because B-splines are linear in their parameters, we can use the well-developed machinery of linear regression for learning. See Ch. 20 in Gelman et al. (2014) or Ch. 5 in Friedman et al. (2001) for further details.

Penalized B-splines. In practice, the parameters of a B-spline model are fit using a penalized least squares criterion. The penalty is typically introduced in order to control the smoothness of the fit. For data \tilde{y} measured at times \tilde{t} with corresponding basis matrix $\Phi(\tilde{t}) = [\Phi(t_1), \dots, \Phi(t_n)]^\top$, we minimize the following objective:

$$J(\bar{\beta}) = \|\tilde{y} - \Phi(\tilde{t})\bar{\beta}\|_2^2 + \rho \bar{\beta}^\top \Omega \bar{\beta}, \quad (4)$$

where Ω is a first-order differences matrix as described by Eilers and Marx (1996). The penalized objective is still quadratic in $\bar{\beta}$ and so can be easily minimized.

3.1.2 CONDITIONAL RANDOM FIELDS

Conditional random fields (CRFs) provide a framework for modeling and learning the joint distribution of a collection of random variables conditioned on some set of observations (see e.g., Murphy 2012). The parameterization is identical to that of Markov random fields (MRF), but the factors that define the distribution can be functions of the observations (this allows the distribution to vary depending on the values of the observations). For some

output y , input x and parameters θ , the conditional probability is defined to be:

$$p(y | x, \theta) = \frac{1}{Z(x, \theta)} \prod_c \psi_c(y_c | x, \theta), \quad Z(x, \theta) \triangleq \sum_y \prod_c \psi_c(y_c | x, \theta), \quad (5)$$

where $\psi_c(y_c | x, \theta)$ is a non-negative factor that can be interpreted as scoring the configuration of the subset of variables y_c given the observations x and parameters θ . The term $Z(x, \theta)$ is called the *partition function* and ensures that the distribution is normalized. When we can write

$$\log \psi_c(y_c | x, \theta) = \theta_c^\top f_c(y_c, x) \iff \psi_c(y_c | x, \theta) = \exp \left\{ \theta_c^\top f_c(y_c, x) \right\}, \quad (6)$$

where f_c extracts some vector of features from the observations x and the target y_c , then we say that the CRF is a log-linear model. Log-linear models have a number of desirable properties, the most relevant to this work being the ease with which we can differentiate the log-likelihood with respect to model parameters. To compute the derivative with respect to θ_c (the parameters corresponding to the c^{th} factor) we have:

$$\frac{\partial \log p(y | x, \theta)}{\partial \theta_c} = f_c(y_c, x) - \frac{\partial \log Z(x, \theta)}{\partial \theta_c}. \quad (7)$$

To compute the partial derivative in the second term on the RHS, first note that

$$\begin{aligned} \frac{\partial Z(x, \theta)}{\partial \theta_c} &= \sum_{y'} \left(\prod_{d \neq c} \psi_d(y'_d | x, \theta_d) \right) \frac{\partial \psi_c(y'_c | x, \theta_c)}{\partial \theta_c} \\ &= \sum_{y'} \left(\prod_{d \neq c} \psi_d(y'_d | x, \theta_d) \right) \psi_c(y'_c | x, \theta_c) f_c(y'_c, x). \end{aligned} \quad (8) \quad (9)$$

This implies that the partial derivative of $\log Z(x, \theta)$ is simply:

$$\frac{\partial \log Z(x, \theta)}{\partial \theta_c} = \frac{1}{Z(x, \theta)} \frac{\partial Z(x, \theta)}{\partial \theta_c} = \mathbb{E}_y [f_c(y_c, x) | x] \quad (10)$$

This means that the gradient of the log-likelihood with respect to a set of parameters θ_c is the difference between the observed features $f_c(y_c, x)$ and their expectation under the current set of parameters θ . To learn the weights, we can apply gradient-based algorithms to optimize the likelihood of a set of observed training input-output pairs. In addition, a regularizer is often added to the objective to discourage complexity or induce sparsity. We will use these ideas in the derivation of our learning algorithm. See Ch. 19 in Murphy (2012) for further details.

3.2 Latent Trajectory Model

The Latent Trajectory Model (LTM) is a probabilistic model introduced by Schulam and Saria (2015) for obtaining individualized predictions of a clinical marker trajectory in populations with diverse disease expression. LTM posits that the measured markers are

to occur. The OU kernel is ideal for modeling such deviations as it is both mean-reverting and draws from the corresponding stochastic process are only first-order continuous, which eliminates long-range dependencies between deviations (Rasmussen and Williams, 2006). Applications in other domains may require different kernel structures motivated by properties of transient deviations in the trajectories.

Accounting for treatments. Several interventions are common in scleroderma, but none have been proven to significantly alter the long-term course of the disease. For example, steroids are commonly administered, but there have been no randomized controlled trials confirming its effects on patients with scleroderma-related lung disease—see, for example, Ch. 35 in Varga et al. (2012). Immunosuppressants are also commonly used to treat scleroderma-related lung disease, but the proven effects are modest and have only been demonstrated over the course of one year (Tashkin et al., 2006). We assume that these types of transient interventions are well-modeled by the individual-specific short-term component, and so we do not explicitly model the treatment effects of steroids or immunosuppressants in our data. Others have developed methods for estimating treatment effects from observational time series (e.g., Chib and Hamilton 2002; Kleinberg and Hirpsak 2011; Brodersen et al. 2015). More recently, see Xu et al. (2016) for an application using functional data. Treatment effects can be incorporated within the trajectory likelihood in diseases where treatments are suspected to alter long term trajectory. We leave this more general case as a direction for future work.

Missing data mechanism. The LTM assumes observations of the trajectory are missing at random (MAR). This implies that we can use maximum likelihood estimation without needing to incorporate additional information about the sampling model; see Appendix B. When the data are missing not at random, assumptions about the missing data mechanism should be explicated and incorporated within the individual marker models.

In summary, the latent, individual-specific factors in the model (z_i , \bar{b}_i , and f_i from Eq. 11B, 11C, and 11D respectively) each contribute to describe the observed trajectory at different granularities. These are all treated as random variables and marginalized out during learning to avoid overfitting. When making predictions, we can use an individual’s observed data to compute posterior distributions over these latent factors, which allows us to tailor predictions.

3.2.1 LTM LIKELIHOOD

Given parameters $\Theta = \{\Lambda, \pi_{1:G}, \bar{\beta}_{1:G}, \Sigma_b, a, \ell, \sigma^2\}$, we can compute the observed-data likelihood of a given clinical marker trajectory by marginalizing z_i , \bar{b}_i and f_i out of the joint distribution defined by our model:

$$p(\bar{y}_i | \bar{x}_i; \Theta) = \sum_{z_i=1}^G \underbrace{p(z_i | \Theta)}_{\text{Multinomial prior}} \int_{\mathbb{R}^{d_\ell}} \underbrace{p(\bar{b}_i | \Theta)}_{\text{Normal prior}} \int_{\mathbb{R}^{N_i}} \underbrace{p(f_i | \Theta)}_{\text{GP prior}} p(\bar{y}_i | z_i, \bar{b}_i, f_i, \bar{x}_i; \Theta) df_i d\bar{b}_i \quad (13)$$

$$= \sum_{z_i=1}^G \pi_{z_i} \mathcal{N}(\bar{y}_i | \Phi_p(\bar{t}_i) \Lambda \bar{x}_i + \Phi_z(\bar{t}_i) \bar{\beta}_{z_i}, K(\bar{t}_i, \bar{t}_i)). \quad (14)$$

Moving from Eq. 13 to Eq. 14, we evaluate the innermost integral using the fact that the GP prior over f_i is conjugate to Eq. 11 yielding a new multivariate normal (Rasmussen and Williams, 2006). To evaluate the next integral in Eq. 13, we again have that the normal prior over \bar{b}_i is conjugate to the multivariate normal obtained by marginalizing over f_i , which gives us the multivariate normal shown in Eq. 14 where the covariance function is defined as

$$K(t_1, t_2) = \Phi_\ell(t_1)^\top \Sigma_b \Phi_\ell(t_2) + \text{Kou}(t_1, t_2) + \sigma^2 \mathbb{1}(t_1 = t_2). \quad (15)$$

We see that the observed-data likelihood for individual i is defined by a mixture of multivariate normals where each subtype is associated with a class in the mixture. The mixing probabilities are defined by the multinomial over subtypes. The mean of the multivariate normal is defined by the population and subpopulation models, and the covariance is defined by the individual long-term and short-term components of the model. To obtain the LTM likelihood needed in Eq. 2, we will condition Eq. 14 on subtype z_i . This gives us the following expression:

$$p(\bar{y}_i | z_i, \bar{x}_i) \triangleq \mathcal{N}\left(\bar{y}_i | \Phi_p(\bar{t}_i) \Lambda \bar{x}_i + \Phi_z(\bar{t}_i) \bar{\beta}_{z_i}, K(\bar{t}_i, \bar{t}_i)\right). \quad (16)$$

3.2.2 LTM PREDICTIVE

As presented, the LTM can be easily applied to the task of disease activity trajectory prediction. Note that the LTM provides a posterior distribution over the trajectory using baseline markers and measurements of the target marker (e.g. PFVC) as they are recorded. It does not incorporate information from other time-varying markers such as TSS and PDLCO. Suppose we have estimates of the model parameters $\Theta = \{\Lambda, \pi_{1:G}, \bar{\beta}_{1:G}, \Sigma_b, a, \ell, \sigma^2\}$, then we can predict an individual’s future course by computing the posterior predictive distribution $p(\bar{y}_{i>t} | \bar{y}_{i \leq t}, \bar{x}_i)$, where $\bar{y}_{i>t}$ denotes marker values after time t and $\bar{y}_{i \leq t}$ denotes marker values observed prior to time t . To compute the expected marker value at time t_i^* , we evaluate the following expression:

$$\hat{y}(t_i^*) = \sum_{z_i=1}^G \int_{\mathbb{R}^{d_\ell}} \int_{\mathbb{R}^{N_i}} \underbrace{\mathbb{E}[y_i^* | z_i, \bar{b}_i, f_i]}_{\text{prediction given latent vars.}} p\left(z_i, \bar{b}_i, f_i | \bar{y}_{i \leq t}, x_{0:t}, \Theta\right) df_i d\bar{b}_i \quad (17)$$

$$= \mathbb{E}_{z_i, \bar{b}_i, f_i}^* \left[\Phi_p(t_i^*)^\top \Lambda \bar{x}_i + \Phi_z(t_i^*)^\top \bar{\beta}_{z_i} + \Phi_\ell(t_i^*)^\top \bar{b}_i + f_i(t_i^*) \right] \quad (18)$$

$$= \underbrace{\Phi_p(t_i^*)^\top \Lambda \bar{x}_i}_{\text{pop. prediction}} + \underbrace{\Phi_z(t_i^*)^\top \mathbb{E}_{z_i}^*[\bar{\beta}_{z_i}]}_{\text{subpop. prediction}} + \underbrace{\Phi_\ell(t_i^*)^\top \mathbb{E}_{\bar{b}_i}^*[\bar{b}_i]}_{\text{ind. long prediction}} + \underbrace{\mathbb{E}_{f_i}^*[f_i(t_i^*)]}_{\text{ind. short prediction}}, \quad (19)$$

where E^* denotes an expectation conditioned on $\bar{y}_{i \leq t}, x_i, \Theta$. We see that the prediction takes a natural form: we compute the value of the individual’s disease activity trajectory at the future time point by replacing the latent factors with their posterior expectations. Computing the population prediction is straightforward as all quantities are observed. To compute the subpopulation prediction, we need to compute the marginal posterior over z_i ,

then we can write $\mathcal{D}(i, t)$ (Eq. 1) as

$$\mathcal{D}(i, t) = \sum_{z_i} p(\mathbf{Y}(\cdot) | z_i, \bar{y}_i \leq t, \bar{x}_i) p(z_i | \mathcal{H}(i, t)). \quad (29)$$

Intuitively, we see that the predictive distribution under C-LTM is simply a weighted combination of the subtype-specific predictive distributions under LTM (Eq. 24). Moreover, the distribution $p(z_i | \mathcal{H}(i, t))$ is the marginal distribution over z_i in a conditional random field with structure similar to the coupling model (Eq. 25) but augmented with additional singleton factors defined by the LTM likelihood functions given the marker trajectory histories. The LTM likelihood factors in Eq. 27 are added into the model unchanged, but additional parameters $\{\gamma, \gamma_{1:C}\}$ can be included to reweight those terms (a similar idea is used in Raina et al. (2003)).¹ The factor graph for this conditional random field is shown in Figure 4. Note that the weight $p(z_i | \mathcal{H}(i, t))$ can be efficiently computed in time linear in the number of auxiliary markers using the junction tree algorithm.

The C-LTM offers a number of advantages for predictive modeling of disease trajectories in domains where many other related marker trajectories are available. First, it allows irregularly and sparsely sampled trajectories to be neatly summarized using modularized, single-marker generative models. These can capture important latent factors and account for marker-specific measurement models and noise processes. Second, we can discriminatively use auxiliary marker trajectory histories when modeling Eq. 1 instead of specifying a joint generative model, which sidesteps the challenges associated with correctly specifying dependencies between many different marker types. Finally, the model can be used in continuous time and it dynamically updates predictions as new observations arrive.

3.5 Learning the C-LTM

We have described two components of our approach: the Latent Trajectory Model (LTM) and the coupling model. When these components are combined as shown in Section 3.4, then we obtain the C-LTM. The C-LTM has two conceptually distinct sets of parameters. The first set are those belonging to the individually trained LTMs for each marker type. To learn these, we can use the EM algorithm described in Schulam and Sarria (2015). To learn the parameters for the C-LTM, we keep the single-marker model parameters fixed (e.g. those learned for the LTM), and use a standard gradient-based CRF learning algorithm (as described in Section 3.1.2) to optimize the penalized log-likelihood of example trajectory predictions. For completeness, we provide additional details for both stages in Appendix A.

3.5.1 SCALABILITY

The EM algorithm used to learn the parameters of the LTM poses no serious challenges to scalability. The primary computational burden lies in the E-step wherein sufficient statistics from all individuals are computed and collected. This is linear in the number of patient records being analyzed, but since the inference required to compute the sufficient statistics can be performed independently for each individual given the current parameter estimates, the E-step can be easily parallelized to offset slow learning due to large numbers

of patient records. For any given individual, the E-step is dominated by the inversion of the $N_i \times N_i$ covariance matrix. We do not expect this to be problematic, however, because clinical markers in chronic diseases are observed at a maximum rate of 12 times per year. Moreover, such diseases occur over periods on the order of tens of years. Therefore, the number of measurements will be at most on the order of 100-200.

Learning the parameters of the CRF requires a sweep through all $M|T|$ training instances in order to compute and aggregate the gradient at each iteration. The primary computational burden is computing the expected values of the features (Eq. 42), however, the tree-structured graphical model shown in Figure 4 allows the junction tree algorithm to run in time linear in the number of auxiliary markers. On a standard laptop, we are able to train the model on 772 patients (5,458 PFVC measurements) in 10-20 minutes.

Online inference for predicting a given individual’s future trajectory is also computationally straightforward. The key quantities are (1) the weights $p(z_i | \mathcal{H}(i, t))$ in Eq. 29, which are easily computed using the junction tree algorithm in time linear in the number of auxiliary markers, and (2) the subtype-specific predictive densities $p(\mathbf{Y}(\cdot) | z_i, \bar{y}_i \leq t, \bar{x}_i)$, which have the same computational complexity as the E-step in the LTM learning algorithm.

4. Experiments

We demonstrate our approach by building a tool for predicting lung disease trajectories for individuals with scleroderma. Lung disease is currently the leading cause of death among scleroderma patients, and is notoriously difficult to treat due to the lack of accurate predictors of decline and tremendous variability across individual trajectories (Allanore et al., 2015). Clinicians use percent of predicted forced vital capacity (PFVC) to track lung severity, which is expected to drop as the disease progresses. In addition, they collect demographic information and other clinical marker values that measure the impact of disease on the different organ systems involved in scleroderma.

Data description. We train and validate our model using data from the Johns Hopkins Scleroderma Center patient registry, one of largest collections of clinical scleroderma data in the world. Demographic information is collected during the patient’s first visit to the clinic. PFVC and other clinical markers are collected during routine visits thereafter. To select individuals from the registry, we used the following criteria. First, we include individuals who were seen at the clinic within two years of their earliest scleroderma-related symptom² (1,186 individuals). Second, we exclude all individuals with fewer than two PFVC measurements after first being seen by the clinic (398 individuals). Finally, we exclude individuals who received a lung transplant (16 individuals) because their natural trajectory is altered by the intervention. Transplants are rare so removing patients with transplants should not introduce significant bias. As mentioned earlier, there are no other known course-altering therapies for scleroderma.

Our final data set contains 772 individuals and a total of 5,458 PFVC measurements tracking individuals over a period of 20 years. The first, second, and third quartiles of the total number of PFVC measurements for an individual are 3, 5, and 9 respectively. The maximum number of PFVC measurements for one individual is 63. The first, second, and third quartiles of the measurement times are 1 year, 2.8 years, and 5.9 years. The first,

¹ When using a penalty, we can center the weights at 1 so that the default behavior is to leave the likelihood factors unchanged as in Eq. 27

² Date of first symptom is established during the first encounter by both the patient and clinician.

second, and third quartiles of elapsed time between measurements are 0.4 years, 0.7 years, and 1.10 years. The minimum and maximum elapsed time is 0.002 years and 16.4 years respectively.

The baseline demographic information includes gender and African American race, both of which have been shown to be associated with disease severity in scleroderma (Allanore et al., 2015). Antibody data are also collected at baseline, but since these are only available for a small subset of individuals, we do not include that data here. For time-dependent predictors, we include 5 auxiliary clinical markers. Three of the auxiliary markers are similar to PFVC in that they are continuous-valued test results used to measure the health of organ systems. We include: percent of predicted forced expiratory volume in one second (PFEV1), which measures the force with which air is expelled from the lungs; percent of predicted diffusing capacity (PDLCO), which measures the efficiency of oxygen diffusion from the lungs to the bloodstream; and total skin score (TSS), which is a cumulative measure of the thickness of the skin at various points on the body. In addition, we include 2 severity scores—clinical Likert-scaled judgements of organ damage severity: Raynaud’s phenomenon (RP) severity score, which measures the severity of damage to the extremities by issues related to the vasculature, and GI severity score that measures the severity of damage to the GI tract. For the interested reader, a more detailed discussion of these markers and their relationship to the disease can be found in Varga et al. (2012).

Experimental setup. For the 4 continuous-valued clinical markers (PFVC, PFEV1, PDLCO, TSS) we use the LTM and for the 2 severity scores (GI and RP) we use a simpler model that we will describe later. For the population model, we use constant functions (i.e. the basis expansion $\Phi_p(t)$ contains an intercept term whose coefficient is determined by baseline covariates). For the subpopulation B-splines, we set boundary knots at 0 and 25 years (the maximum observation time in our data set is 23 years), use two interior knots that divide the time period from 0-25 years into three equally spaced chunks, and use quadratics as the piecewise components. For the individual-specific long-term basis Φ_t , we use the same basis as the population model (constant functions).

We divide our data into 10 folds and use log-likelihood on the first fold for tuning hyperparameters. For PFVC, we select $G = 9$ subtypes using BIC. For the kernel hyperparameters $\Theta_1 = \{\Sigma_b, \alpha, \ell, \sigma^2\}$ we set $\Sigma_b \in \mathbb{R}$ to be 16.0, which corresponds to the variance of individual-specific intercepts. We set $\alpha = 6$, $\ell = 2$, and $\sigma^2 = 1$ using a grid search over values chosen using domain knowledge. Qualitatively, these make sense; we expect transient deviations to last around 2 years and to change PFVC by around ± 6 units. Finally, we penalize the expected log-likelihood with respect to $\beta_{1,G}$ as in Eq. 4 and set the weight $\rho = 0.01$, which was chosen based on the clinical interpretability of the learned subtype trajectories. The remaining 9 folds were used for our cross-validation experiments. The parameters of each trajectory model are estimated independently for each fold (e.g. the B-spline coefficients of the subtype trajectories). For the severity scores, which are Likert-scaled and not continuous, we use a simple naive Bayes generative model wherein the latent “class” is an indicator of whether the individual ever reaches a high severity level (a cutoff in the severity scale determined by clinical collaborators). Severity score observations are treated as iid draws from a class-specific multinomial distribution (i.e. the likelihood for these auxiliary markers is a multinomial distribution over severity scores). Finally, we estimate the parameters of the C-LTM by maximizing the objective in Eq. 33 augmented

with an $L1$ regularizer. We optimize the objective using the Orthant-Wise Limited-memory Quasi-Newton (OWL-QN) algorithm (Andrew and Gao, 2007). To generate training examples for the C-LTM, we use times $\mathcal{T} = \{1, 2, 4\}$ (the first three quintiles of observation times in our data) to fit three different models. We choose time points earlier in the disease course because this is when it is most valuable to leverage all available information. In our cross-validated experimental results below, we estimate the penalty of the $L1$ regularization term in each fold by splitting a portion of the training data into a development set. We sweep the penalty from 1.0×10^{-7} to 1.0×10^{-1} and choose based on development set performance.

Baselines. As a first baseline, we fit a regression model using static predictors only (features in \vec{x}_i). This is to compare against typical approaches in clinical prediction which rely only on observed features to predict disease progression (e.g. Khanna et al. 2011). The regression function is as follows, where $\Phi(t)$ is a B-spline basis:

$$\hat{y}(t) \mid \vec{x}_i = \Phi(t)^\top \left(\vec{\beta}_0 + \sum_{\sigma_{ij}} \vec{x}_{i,j} \vec{\beta}_j + \sum_{\sigma_{ij}, \sigma_{ik}} \vec{x}_{i,j} \vec{x}_{i,k} \vec{\beta}_{ij} \right). \quad (30)$$

The following baselines reflect state-of-the-art approaches for dynamical prediction. The focus for each of these models, as discussed in the related work section, is on dynamical prediction of single marker trajectories using the marker history and static measurements collected during the first visit. The second baseline, like Rizopoulos (2011) and Shi et al. (2012), defines a single mean function parameterized in the same way as the first baseline and models individual-specific variations using a GP with the same kernel as in Equation 15 (using hyper-parameters as above). The third baseline is a mixture of B-splines, which models subpopulations that can express different trajectory shapes (as in Proust-Lima et al. (2014)).³ Finally, we use the LTM (no coupling to auxiliary markers) as a baseline. All B-spline bases used in these baseline models are parameterized in the same way as the C-LTM (described above).

Evaluation. Prediction accuracy for all models is measured using the absolute error between the predicted and a smoothed version of the individual’s observed trajectory. We make predictions after one, two, and four years of follow-up, which are summarized using averages computed in the second year of follow-up ($t \in \{1, 2\}$), in the third and fourth year of follow-up ($t \in \{2, 4\}$), fifth to eighth year of follow-up ($t \in \{4, 8\}$), and beyond the eighth year of follow-up ($t \in \{8, 25\}$)⁴. Mean absolute errors (MAE) and standard errors are estimated using 9-fold CV⁵ at the level of individuals (i.e. all of an individual’s data is held-out). Significance tests are computed against baselines using a paired t-test with point-wise predictions aggregated across folds.

4.1 Results

In this section, we present four sets of results. The first two are qualitative, and demonstrate the advantages of the C-LTM over the baseline models using examples. In the first

3. For the B-spline mixture, we use the subtypes discovered by LTM as the mixture classes. Without accounting for individual-specific variability explicitly, we have found that fitting a B-spline mixture using EM recovers poor classes that do not capture important trajectory shapes in the data. For additional details, see Section 3 in the supplement of Schulam and Saria (2015).
4. After the eighth year, data becomes too sparse to further divide this time span.
5. Recall that the first of 10 folds is used for hyperparameter estimates.

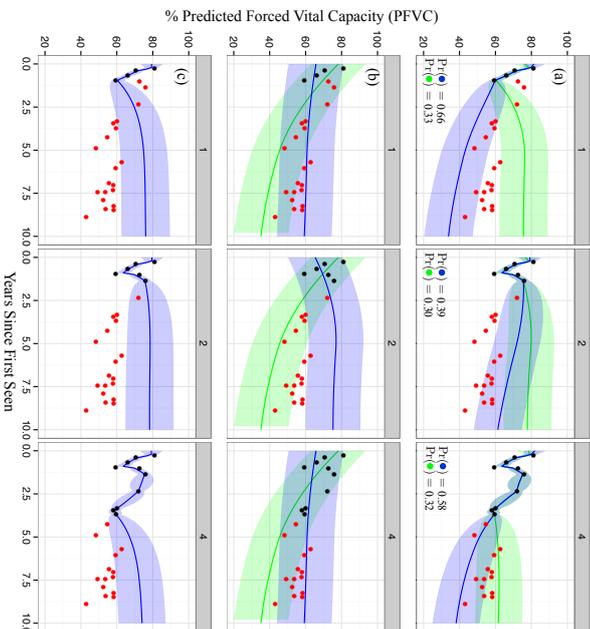


Figure 5: Examples of predictions made using 1, 2, and 4 years of data (moving across columns from left to right). Plot (a) shows dynamic predictions using C-LTM. Red markers are unobserved. Blue shows the trajectory predicted using the most likely subtype, and green shows the second most likely. Plot (b) shows dynamic predictions for the B-spline mixture baseline. Plot (c) shows the same for the B-spline + GP baseline.

qualitative analysis, we compare predictions made by C-LTM to those made by the B-spline mixture and the B-spline + GP. In the second qualitative analysis, we compare the C-LTM inferences with those from the LTM, which is a state-of-the-art single-marker model. The second two results are quantitative. The first compares predictive accuracies between the baseline models and the C-LTM. The second investigates clinical utility by using each model to predict a severity score that we use to detect individuals with aggressive lung disease.

4.1.1 VISUAL COMPARISON TO BASELINES

As an illustrative example to compare C-LTM with baselines, in Figures 5a, 5b, and 5c we show the dynamic predictions made using the C-LTM, the B-spline mixture, and the B-spline + GP baselines on a sample patient.⁶ For each model, we show 95% posterior

⁶ This patient was selected as an exemplar for the types of errors commonly made by the baseline models.

intervals for the future trajectory. For the C-LTM and B-spline mixture, the most likely subtype is shown in blue and the second most likely is in green. The B-spline mixture (Figure 5b) cannot explain individual-specific sources of variation (e.g. short-term deviations from the mixture mean) and so over-reacts to the slight rise in PFVC seen in the last two observed (black) measurements in the second panel (year 2). The B-spline + GP (Figure 5c) cannot capture long-term differences in trajectory means (e.g. due to subtypes) and so pulls back to the population mean over time even after four years of data suggest a declining trajectory. On the other hand, at year 1 the C-LTM (Figure 5a) maintains the hypothesis that the individual may decline or return to stability (correctly putting most weight on the former). After 2 years of data, the temporary recovery seems to have caused confidence in the declining trajectory to fall (going from 66% to 39%), but the top-weighted hypothesis is still correct. After 4 years of data, the model again becomes confident in the declining trajectory. Clinically, this robustness to short-term changes is important. After having seen the recovery between years 1 and 2, a clinician may become less immediately concerned with the individual's future lung disease, possibly delaying immunotherapy until a rapid decline becomes more evident. Note that the B-spline mixture, on the other hand, over-reacts to the recovery and predicts that the individual will continue to recover.

4.1.2 ANALYSIS OF EXAMPLE INFERENCES

In Figure 6a-d, we show the C-LTM's target and auxiliary marker inferences for four different patients. For the target marker (PFVC) and auxiliary markers (TSS, PDLCO, and PFEV1), we show the most likely (blue) and second most likely (green) subtype and their corresponding trajectories. For the RP and GI severity score markers, we show the most likely severity class (high versus low). The dashed lines indicate the threshold at which high and low are determined based on judgements by our clinical collaborators. For PFVC, PFEV1, and PDLCO lower values indicate more severe progression. For TSS, higher values indicate severe progression. In Figures 6e-h, we show the predictions made by LTM to visually compare against predictions made using the baseline markers and PFVC history only (i.e. that do not leverage information from auxiliary markers).

In Figure 6a, we see a 55-year-old woman who presents with mildly impaired lung function (approximately 65 PFVC), but seems to recover over the course of the first year to reach a PFVC above 75 (considered by clinicians to be relatively healthy). Using this information alone, one may suspect that she will not have future lung issues. Indeed, this is what LTM predicts as shown in Figure 6e. By examining her auxiliary markers, however, we see that the picture is less clear. In particular, PFEV1 (a clinical marker closely related to PFVC) both decreases and increases over that period. C-LTM infers a mildly declining trajectory for PFEV1. In addition, PDLCO is also noisy and overall low, which suggests that the blood is not efficiently absorbing oxygen. This can happen for a number of reasons, but active lung disease is one of them. Finally, we see that her initial skin score is quite high and C-LTM projects it to stay high for the next few years, which is associated with active lung disease. We see that C-LTM has successfully incorporated inferences about the future trends of the auxiliary markers and correctly predicts that this woman's PFVC will decline after this initial improvement.

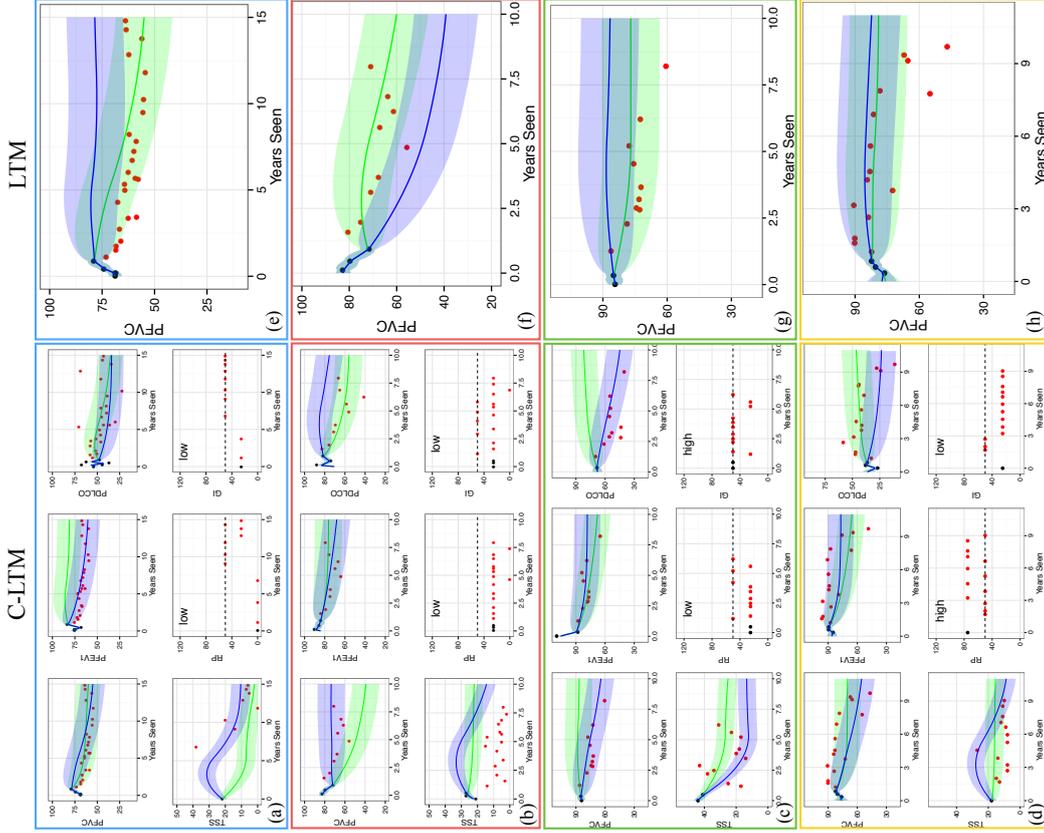


Figure 6: The predicted PFVC trajectory and the auxiliary markers are shown for two different patients. Red markers are unobserved. For the auxiliary markers TSS, PFEV1, and PDLCO we show the most likely (blue) and second most likely (green) subtype and their corresponding trajectories. For the RP and GI severity scores, we show the most likely severity class (high versus low). The dashed lines indicate the threshold at which high and low are determined clinically.

In Figure 6b, we see a 75 year-old white woman who presents with healthy lung function (approximately 85 PFVC), but is consistently declining over the course of the first year by nearly 15 PFVC. A clinical rule of thumb is that a drop in 10 PFVC over the course of a year warrants close monitoring for active lung disease. We see that LTM extrapolates this initial trend and predicts that this individual will continue to decline rapidly (Figure 6f). Just as in the previous example, however, the auxiliary markers paint a more complete picture of this individual. In the first few PFEV1 observations, we see that this decline is not quite as pronounced and the progression is predicted to be more mild. In PDLCO we see that oxygen is absorbed into the blood at healthy levels and also predicted to remain stable (although incorrectly in this case). Finally, C-LTM predicts that the RP and GI severity scores will remain low, which also supports the prediction that this woman will stabilize. Note that in this example C-LTM overestimates the course of PDLCO and TSS. Although the model still makes the correct prediction for PFVC in spite of this mistake, it highlights that the performance of our approach may be further improved with better auxiliary marker inferences. As research in systems biology yields new insights into modeling specific measurements more precisely, the modular architecture of C-LTM makes it possible to improve overall performance by incorporating improved versions of the target or auxiliary marker models.

In Figure 6c, we see a 76 year-old white woman that presents with healthy lung function (just under 90 PFVC), which also appears to be stable given the subsequent test result taken later that same year. The LTM predicts that this individual's most likely course is to remain stable. From the PFEV1 trajectory, however, we see that there was a large initial loss in PFEV1, which, together with the unusually high skin score (TSS) suggests that this woman's disease is active. The activity in the other organ systems allows the C-LTM to offset the stability seen in the first two PFVC measurements and correctly predict the consistently declining lung trajectory.

Finally, in Figure 6d, we see a 67 year-old African American man with mildly impaired lung function early in the disease course (around 75 PFVC) that seems to recover over the next one or two years to a healthier 85 PFVC. In Figure 6h, we see that the LTM predicts that a stable trajectory thereafter is likely. By considering other organ systems, however, we see that this man's blood-oxygen diffusion is severely limited early in the disease course (nearly 25% of the predicted DLCO). Moreover, we see that the this individual's Raynaud's phenomenon severity score is high early on and correctly predicted to remain that way. The low PDLCO and high RP severity score point to active vasculature disease, which is hypothesized to cause late deterioration in lung function. We see that C-LTM correctly uses this evidence to predict an accurate disease trajectory.

4.1.3 PREDICTIVE ACCURACY

In Table 1, we report performance of the C-LTM, LTM, and the three other baseline models. First, we note that the C-LTM statistically significantly outperforms the B-spline with baseline features for all predictions. This baseline makes static predictions using baseline information only, and cannot adapt to an individual as new data becomes available. Moreover, after an initial amount of data has been collected on an individual, C-LTM statistically significantly outperforms all other models. This is not surprising. When compared to the

Model	Predictions using 1 year of data				(8, 25)
	(1, 2)	(2, 4)	(4, 8)	(8, 25)	
B-spline with Baseline Feats.	13.17 (0.43)	14.07 (0.61)	14.34 (0.65)	14.12 (1.04)	
B-spline + GP	5.57 (0.24)	8.40 (0.19)	10.88 (0.42)	11.74 (0.76)	
B-spline Mixture	6.31 (0.22)	7.59 (0.36)	9.82 (0.46)	13.77 (0.55)	
LTM	5.70 (0.30)	8.02 (0.41)	11.17 (0.72)	13.93 (0.67)	
C-LTM	★◆◆◆5.12 (0.20)	★◆◆◆5.88 (0.27)	★◆◆◆9.95 (0.51)	★13.70 (1.08)	
Predictions using 2 years of data					
B-spline with Baseline Feats.	14.07 (0.61)	14.34 (0.65)	14.12 (1.04)	14.12 (1.04)	
B-spline + GP	6.51 (0.19)	9.79 (0.35)	10.95 (0.68)	12.19 (0.48)	
B-spline Mixture	6.17 (0.29)	8.34 (0.36)	10.11 (0.56)	10.89 (0.62)	
LTM	5.74 (0.29)	8.08 (0.37)	10.89 (0.62)	12.19 (0.48)	
C-LTM	★◆◆◆5.58 (0.34)	★◆◆7.99 (0.61)	★◆◆7.99 (0.61)	★◆◆11.27 (1.02)	
Predictions using 4 years of data					
B-spline with Baseline Feats.	14.34 (0.65)	14.34 (0.65)	14.12 (1.04)	14.12 (1.04)	
B-spline + GP	6.60 (0.24)	9.79 (0.35)	10.95 (0.68)	12.19 (0.48)	
B-spline Mixture	6.00 (0.37)	8.34 (0.36)	10.11 (0.56)	12.19 (0.48)	
LTM	4.88 (0.28)	8.08 (0.37)	10.89 (0.62)	12.19 (0.48)	
C-LTM	★◆◆5.04 (0.42)	★◆◆5.04 (0.42)	★◆◆8.07 (0.35)	★◆◆8.07 (0.35)	

Table 1: Mean absolute error of PFVC predictions for the B-spline with baseline features, the B-spline + GP, LTM, and C-LTM. Bold numbers indicate best performance across baseline models and proposed model. ★ indicates statistically significant improvement against the B-spline model with baseline features only using a paired t-test ($\alpha = 0.05$). ◆ indicates statistical significance compared against the B-spline + GP. ◆ indicates statistical significance compared against the B-spline mixture. ♠ indicates statistical significance compared against LTM.

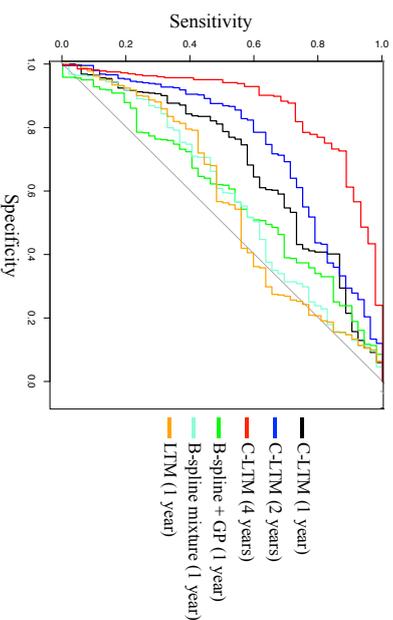
LTM, we see that C-LTM benefits from leveraging information from auxiliary markers. As more information is collected, both models are able to the individual and provide comparable predictions. The B-spline mixture is not able to personalize beyond capturing long-term trends across subpopulations, so we see that it becomes less competitive compared to both C-LTM and LTM as more data are collected. Finally, the B-spline + GP cannot capture long-term trends specific to subpopulations (as we saw in Section 4.1.1), and so we see that it does poorly when making predictions.

4.1.4 CLINICAL UTILITY

One may naturally wonder whether the observed improvements in MAE reported above translate to practical benefits in the clinic. In the examples shown in Figure 6, we have walked through cases where the model makes predictions that would seem unlikely if we were to consider PFVC alone. This suggests that the model can augment expert clinical judgement and may serve to protect against incorrect extrapolations. In this section, we further elaborate upon this intuition by studying clinical utility quantitatively. In particular, we compare how well the B-spline + GP, B-spline mixture, LTM, and C-LTM are able to detect individuals who will have rapidly declining lung function. It is notoriously difficult to predict which scleroderma patients will rapidly decline using only information from early in the disease course. In addition to improving prognoses, more accurate detection of rapidly declining lung function can help to improve the recruitment for clinical trials evaluating drugs for scleroderma-related lung disease. If we include many individuals in a study who

Model / Years of Data	1	2	4
B-spline + GP	0.59	0.63	0.74
B-spline mixture	0.58	0.63	0.76
LTM	0.57	0.71	0.84
C-LTM	0.68	0.75	0.87

(a) AUCs for detecting declining individuals.



(b) ROCs comparing B-spline + GP at 1 year, B-spline mixture at 1 year, LTM at 1 year, and C-LTM at years 1, 2, and 4.

Figure 7: Declining individual detection results.

are predicted to have active lung disease but do not, the results of the study are blurred because both arms of the trial may include many individuals without active lung disease.

To test how well these different models can detect individuals that will experience rapidly declining lung function, we use the predictions of future PFVC measurements to produce a score. The score is defined to be the difference between the individual's first PFVC measurement and the minimum predicted value in the future—this will be higher for individuals on whom a model predicts deteriorating lung function and lower for those predicted to be stable. To label an individual as declining, we require that they (1) have at least one observation within the first year of being seen by the clinic, (2) have 3 years between their first and last measurements, (3) have at least 4 PFVC measurements, and (4) have an initial PFVC measurement that is 20 PFVC higher than their last measurement. Requirements (2) and (3) are to ensure that the trajectory can be reliably annotated as declining or not. For each model, we make predictions at years 1, 2, and 4 and compute the score described above for each individual. We then compute the AUC for each model at each year. Table 7a displays the results of this experiment. We see that C-LTM achieves higher AUC at all years than the baseline models. Figure 7b displays the ROCs for the B-spline + GP (green), B-spline mixture (cyan), LTM (orange), and the proposed model (black) at year 1

and also includes the ROCs for the proposed model at years 2 (blue) and 4 (red) to visualize how performance improves as more data is added. Clinically, an AUC of 0.87 for predicting individuals with lung decline after—on average—four years of data is high and has not been shown previously.

5. Discussion

The goal of personalized (also called precision) medicine is to develop tools that help to tailor prognoses to the characteristics and unique medical history of the individual. In this paper, we describe an approach to personalized prognosis that uses an integrative analysis of multiple clinical marker histories from the individuals' medical records. Our approach combines single-marker latent variable models (the LTM) with a CRF coupling model to make more accurate predictions about the future trajectory of a target clinical marker.

The coupled model (C-LTM) has several advantages. First, the marker-specific LTMs account for marker trajectory shapes using components at the population, subpopulation, individual long-term, and individual short-term levels, which simultaneously allows for heterogeneity across and within individuals, and enables statistical strength to be shared across observations at different “resolutions” of the data. Within an individual marker model, the population and subpopulation components are learned offline, while estimates of the individual-specific parameters are refined over the course of the disease as data accrues for that individual. Second, our coupling model allows us to condition both the target and auxiliary marker histories to make predictions about the future target marker trajectory. We therefore use the marker-specific latent variable models to neatly summarize and extract information from the irregularly sampled and sparse, while simultaneously sidestepping the issue of jointly modeling both the target and auxiliary markers. The conditional formulation is less sensitive to misspecified dependencies between different marker types and can also be easily scaled to diseases with a large number of auxiliary markers. Finally, our model aligns with clinical practice; predictions are dynamically updated in continuous time as new marker observations are measured. We also note that our description of the method and the experimental results focus on predicting the trajectory of a single clinical marker, but multiple latent factor regression models can be easily fit so that many markers can be simultaneously predicted. Using this extension, we only need to maintain different CRF parameters; the latent variable models are shared since they are fit independently as a precursor to learning the CRF.

There are several shortcomings of the proposed approach that are promising directions for future research. First, the model implicitly assumes that the data generating process is noninformative (i.e. missing data is missing at random (Little and Rubin, 2014)). This is appropriate for clinical markers that are routinely collected, but additional machinery would be required to model markers whose missingness is informative. For example, in some cases, additional measurements may be made due to clinical suspicion caused by factors that are not clearly documented in the health record. Researchers have begun to explore more complex missing-data mechanisms for disease trajectory modeling (e.g., Lange et al. 2015; Coley et al. 2016), and it will be important to incorporate these ideas into the framework discussed here to integrate the full set of markers measured during a clinical visit. Another shortcoming is our focus on discrete latent factors of the auxiliary marker trajectories. Continuous-valued

latent factors may also be useful, but would make learning and inference in the latent factor CRF more challenging.

There are also several other immediate opportunities for improving the model. Auxiliary markers are integrated via separate marker-specific generative models. While we incorporated two different types of models—trajectory and maximum-severity based—both of which were data driven, existing and new clinical knowledge should be brought to bear to improve these models, which we expect will improve predictions of the target trajectories. Further, in this work, we focused on modeling the dependency of the target subtype on the auxiliary markers. In addition, estimates of the individual-specific long-term and short-term components may also benefit from conditioning on the auxiliary markers. Finally, the parameters for the pairwise potentials learned in our model may serve as a means for generating hypotheses about the co-evolution of organ-specific trajectories.

The ideas proposed here also open up other longer-term directions for future work. The proposed model does not account for the effects of treatment on an individual's long-term trajectory. In many chronic conditions, as is the case for scleroderma, drugs only provide short-term relief (accounted for in our model by the individual-specific adjustments). However, if treatments that alter long-term course are available and commonly prescribed, then these should be included within the model as an additional component that influences the trajectory. Learning these treatment effects from noisy electronic health record data (e.g., Xu et al. 2016) present an exciting and challenging direction for future work.

We have demonstrated our model by developing a prognostic tool for predicting lung disease trajectories in patients with scleroderma, an autoimmune disease. We showed that the proposed model makes more accurate predictions than state-of-the-art approaches. Accurate tools for prognosis can allow clinicians and patients to more actively manage their disease. While we have focused model development and evaluation on scleroderma, this work is broadly applicable to other complex diseases (Craig, 2008), many of which track disease activity using clinical scales of severity. The proposed model is most directly applicable to CCDs where heterogeneity in disease presentation is common. Examples of such diseases include lupus, multiple sclerosis, inflammatory bowel disease (IBD), chronic obstructive pulmonary disease (COPD), and asthma. Extensions of the proposed ideas, and the model, to these diseases offer an opportunity to address important open challenges in precision medicine.

Acknowledgments

We would like to thank Drs. Laura Hummers, Fredrick Wigley and Robert Wise who have provided extensive clinical guidance as well as the data set with which this study was conducted. We would also like to thank Dr. Colin Ligon for his generous support in chart reviewing patients; many of the key ideas in our work were motivated by these chart reviews. Finally, we would like to thank Zachary Barnes who helped deploy a previous iteration of this model as a tool in the clinic and shadowed clinicians while it was in use. Our work has benefited from the lessons he learned when observing the clinicians use this tool.

Appendix A. Learning the C-LTM: Details

In this section, we provide additional details on the learning algorithm for the C-LTM. Recall that this consists of two stages: (1) independently fitting the single-marker models (the LTM in our case), and (2) fitting the parameters of the coupling model. We describe both stages below.

A.1 Learning the LTM

To learn the parameters of the single-marker model $\Theta = \{\Lambda, \pi_{1:G}, \bar{\beta}_{1:G}, \Sigma_b, a, \ell, \sigma^2\}$, we maximize the observed-data log-likelihood of a training sample of M retrospectively observed trajectories (i.e. the probability of all individual's marker values \bar{y}_i given measurement times \bar{t}_i and features \bar{x}_i). Using the expression for the observed-data likelihood in Eq. 14, we have that the observed-data log-likelihood for all individuals in a training sample is

$$\mathcal{L}(\Theta) = \sum_{i=1}^M \log \left[\sum_{z_i=1}^G \pi_{z_i} \mathcal{N} \left(\bar{y}_i \mid \Phi_b(\bar{t}_i) \Lambda \bar{x}_i + \Phi_z(\bar{t}_i) \bar{\beta}_{z_i}, K(\bar{t}_i, \bar{t}_i) \right) \right]. \quad (31)$$

To maximize the observed-data log-likelihood with respect to Θ , we partition the parameters into two subsets. The first subset, $\Theta_1 = \{\Sigma_b, \alpha, \ell, \sigma^2\}$, contains values that parameterize the covariance function shown in Equation 15. As is often done when designing the kernel of a Gaussian process, we use a combination of domain knowledge to choose candidate values and model selection using observed-data log-likelihood as a criterion for choosing among candidates (Rasmussen and Williams, 2006). The second subset, $\Theta_2 = \{\Lambda, \pi_{1:G}, \bar{\beta}_{1:G}\}$, contains values that parameterize the mean of the multivariate normal distribution in Equation 14. We learn these parameters using expectation maximization (EM) to find a local maximum of the observed-data log-likelihood in Equation 31 (Dempster et al., 1977).

Expectation step. All parameters related to \bar{b}_i and \bar{f}_i are limited to the covariance kernel and are not optimized using EM. We therefore only need to consider the subtype indicators z_i as unobserved in the expectation step. Because z_i is discrete, its posterior is computed by normalizing the joint probability of z_i and \bar{y}_i . Let $\pi_{z_i}^*$ denote the posterior probability that individual i has subtype $g \in \{1, \dots, G\}$, then we have

$$\pi_{z_i}^* \propto \pi_g \mathcal{N} \left(\bar{y}_i \mid \Phi_b(\bar{t}_i) \Lambda \bar{x}_i + \Phi_z(\bar{t}_i) \bar{\beta}_g, K(\bar{t}_i, \bar{t}_i) \right). \quad (32)$$

Maximization step. In the maximization step, we optimize the marginal probability of the soft assignments under the multinomial model with respect to $\pi_{1:G}$. This amounts to collecting total “soft counts” computed in Eq. 32 for each subtype and renormalizing. To optimize the expected complete-data log-likelihood with respect to Λ and $\bar{\beta}_{1:G}$, we note that the mean of the multivariate normal for each individual is a linear function of these parameters. Holding Λ fixed, we can therefore solve for $\bar{\beta}_{1:G}$ in closed form and vice versa. We use a block coordinate ascent approach, alternating between solving for Λ and $\bar{\beta}_{1:G}$ until convergence. To control the smoothness of the subtypes we penalize the log-likelihood with respect to the subtype parameters $\bar{\beta}_{1:G}$ as in Eq. 4. Because the penalized expected complete-data log-likelihood is concave with respect to all parameters in Θ_2 , each maximization step is guaranteed to converge. The exact computations required to maximize the expected log-likelihood can be found in Schulam and Saria (2015) and its supplement.

A.2 Learning the Coupling Model

To learn the parameters of the latent-factor CRF regression, we directly maximize the conditional probability of future target markers given previously observed target markers, previously observed auxiliary markers, and static baseline covariates on a collection of examples extracted from retrospective data. Suppose we are given records containing the target marker, auxiliary markers, and baseline covariates for M individuals. We choose a collection of times \mathcal{T} that will be used to create training examples of history-future pairs. For example, we may choose $\mathcal{T} = \{1, 2\}$ because early management decisions are made using prognoses at years 1 and 2. We emphasize, however, that the model is *not* restricted to making predictions at years 1 and 2; it can make predictions at arbitrary times. The times \mathcal{T} are simply used to create training instances. We also note that it is possible to train specialized models for different time periods. For example, we may train one model for making predictions in the first 2 years and another for beyond 4 years. Given the M records and times \mathcal{T} , we define the objective:

$$J(\gamma, \gamma_{1:C}, \theta, \theta_{1:C}, \eta_{1:C}) = \sum_{i=1}^M \sum_{t \in \mathcal{T}} \log p(\bar{y}_{i,t} > \ell \mid \mathcal{H}(i, t)) \quad (33)$$

$$= \sum_{i=1}^M \sum_{t \in \mathcal{T}} \log \left(\sum_{z_i} \underbrace{p(\bar{y}_{i,t} > \ell \mid z_i)}_{(A)} \underbrace{p(z_i \mid \mathcal{H}(i, t))}_{(B)} \right), \quad (34)$$

where (A) is the subtype-specific multivariate normal likelihood in Eq. 14 and (B) is the conditional distribution over z_i shown in Eq. 28. To learn the parameters, we maximize this objective with respect to $\gamma, \gamma_{1:C}, \theta, \theta_{1:C}$, and $\eta_{1:C}$ using gradient-based methods (e.g. L-BFGS). In our experiments, we optimize a regularized version of the objective, but for simplicity this section discusses the computations required to compute the gradient of Eq. 33 only. Consider a single summand of Eq. 33

$$\log p(\bar{y}_{i,t} > \ell \mid \mathcal{H}(i, t)) = \log \left(\sum_{z_i} p(\bar{y}_{i,t} > \ell \mid z_i) p(z_i \mid \mathcal{H}(i, t)) \right). \quad (35)$$

To reiterate, the parameters of the density $p(\bar{y}_{i,t} > \ell \mid z_i)$ are assumed to have been learned in a separate step (e.g. using the EM algorithm presented above), and so we are only concerned with estimating the parameters of the singleton and pairwise factors in the CRF:

$$\gamma, \gamma_{1:C}, \theta, \theta_{1:C}, \eta_{1:C}.$$

Gradient of the objective. We derive the gradient for a single summand of the objective (Eq. 33), which are combined additively to form the full gradient used at each iteration. Although our model is log-linear over all latent variables z_i and $z_{1:C,i}$, Eq. 35 is not linear in the parameters because the random field does not directly estimate the conditional distribution over the future target clinical markers, but instead estimates the weights assigned to each configuration of the latent variables. We therefore have that the partial derivative of Eq. 35 with respect to any parameter θ_k is:

$$\frac{\partial \log p(\bar{y}_{i,t} > \ell \mid \mathcal{H}(i, t))}{\partial \theta_k} = \frac{\left(\sum_{z_i} p(\bar{y}_{i,t} > \ell \mid z_i) \frac{\partial p(z_i \mid \mathcal{H}(i, t))}{\partial \theta_k} \right)}{p(\bar{y}_{i,t} > \ell \mid \mathcal{H}(i, t))}. \quad (36)$$

To complete the expression for the partial derivative, we need to compute the partial derivative of the probability of a given target marker latent variable z with respect to the parameter θ_k . We have that:

$$\frac{\partial p(z | \mathcal{H}(i, t))}{\partial \theta_k} = \frac{\partial}{\partial \theta_k} \frac{Z'_{i,t}(z)}{Z_{i,t}} = \frac{1}{Z_{i,t}} \frac{\partial Z'_{i,t}(z)}{\partial \theta_k} + Z'_{i,t}(z) \frac{\partial Z_{i,t}^{-1}}{\partial \theta_k}. \quad (37)$$

We can now leverage identities from the theory of log-linear models to continue with the derivation. In particular, recall that log-linear models are in the exponential family of distributions. As a consequence, we can consider the parameters $\gamma, \gamma_{1:C}, \theta, \theta_{1:C}, \eta_{1:C}$ as the *natural parameters* of the distribution. The corresponding *sufficient statistics* are therefore the factors in the log-linear model:

$$\begin{aligned} T(z, z_{1:C}, \vec{x}_i) &= [l_{i,\leq t}(z), l_{1,i,\leq t}(z_1), \dots, l_{C,i,\leq t}(z_C), \\ & f_1^\top(z, \vec{x}_i), f_1^\top(z_1, \vec{x}_i), \dots, f_C^\top(z_C, \vec{x}_i), \\ & g_1^\top(z, z_1), \dots, g_C^\top(z, z_C)]^\top. \end{aligned} \quad (38) \quad (39)$$

An important property of exponential families is that the gradient of the log-normalizing-constant with respect to the natural parameters is simply the expected value of the sufficient statistics computed using the current value of the natural parameters. Note that both $Z'_{i,t}(z)$ and $Z_{i,t}$ are normalizing constants of exponential family distributions. In the case of $Z'_{i,t}$ this is trivial to see because it is the normalizing constant of our log-linear model. In the case of $Z_{i,t}(z)$ we see that it is the normalizing constant of a log-linear model over the auxiliary marker latent variables $z_{1:C}$ given *both* z and the clinical history $\mathcal{H}(i, t)$. We therefore have:

$$\begin{aligned} \frac{\partial \log Z'_{i,t}(z)}{\partial \theta_k} &= \mathbb{E}_{z_{1:C}} [T(z, z_{1:C}, \vec{x}_i)_k | z, \mathcal{H}(i, t)] \\ &\implies \frac{\partial Z'_{i,t}(z)}{\partial \theta_k} = Z'_{i,t}(z) \mathbb{E}_{z_{1:C}} [T(z, z_{1:C}, \vec{x}_i)_k | z, \mathcal{H}(i, t)], \quad (40) \\ \frac{\partial \log Z_{i,t}}{\partial \theta_k} &= \mathbb{E}_{z, z_{1:C}} [T(z, z_{1:C}, \vec{x}_i)_k | \mathcal{H}(i, t)] \\ &\implies \frac{\partial Z_{i,t}^{-1}}{\partial \theta_k} = -\frac{1}{Z_{i,t}} \mathbb{E}_{z, z_{1:C}} [T(z, z_{1:C}, \vec{x}_i)_k | \mathcal{H}(i, t)], \quad (41) \end{aligned}$$

where we have used $T(z, z_{1:C}, \vec{x}_i)_k$ to denote the feature (or sufficient statistic) corresponding to the parameter θ_k . By plugging these partial derivatives back into Eq. 37, we have

$$\begin{aligned} \frac{\partial}{\partial \theta_k} \frac{Z'_{i,t}(z)}{Z_{i,t}} &= \frac{Z'_{i,t}(z)}{Z_{i,t}} (\mathbb{E}_\Theta [T(z, z_{1:C}, \vec{x}_i)_k | z, \mathcal{H}(i, t)] - \mathbb{E}_\Theta [T(z, z_{1:C}, \vec{x}_i)_k | \mathcal{H}(i, t)]) \\ &= p(z | \mathcal{H}(i, t)) (\mathbb{E}_\Theta [T(z, z_{1:C}, \vec{x}_i)_k | z, \mathcal{H}(i, t)] - \mathbb{E}_\Theta [T(z, z_{1:C}, \vec{x}_i)_k | \mathcal{H}(i, t)]). \end{aligned} \quad (42) \quad (43)$$

In words, we see that the partial derivative with respect to a parameter θ_k is the expected value of its corresponding feature given that we have observed the target marker latent variable z and clinical history $\mathcal{H}(i, t)$ minus the expected value of the feature given only the

clinical history $\mathcal{H}(i, t)$. The difference is then weighted by the probability of observing the target marker latent variable given the clinical history. By plugging this expression back into Eq. 36, we arrive at the final expression for the partial derivative of a single summand with respect to θ_k :

$$\begin{aligned} \frac{\partial \log p(\vec{y}_{k,>t} | \mathcal{H}(i, t))}{\partial \theta_k} &= \sum_z \frac{p(\vec{y}_{k,>t} | z) p(z | \mathcal{H}(i, t))}{p(\vec{y}_{k,>t} | \mathcal{H}(i, t))} (\mathbb{E}_\Theta [T(z, z_{1:C}, \vec{x}_i)_k | z, \mathcal{H}(i, t)] - \mathbb{E}_\Theta [T(z, z_{1:C}, \vec{x}_i)_k | \mathcal{H}(i, t)]) \\ &= \sum_z p(z | \vec{y}_{k,>t}, \mathcal{H}(i, t)) (\mathbb{E}_\Theta [T(z, z_{1:C}, \vec{x}_i)_k | z, \mathcal{H}(i, t)] - \mathbb{E}_\Theta [T(z, z_{1:C}, \vec{x}_i)_k | \mathcal{H}(i, t)]). \end{aligned} \quad (44) \quad (45)$$

The partial derivative has a nice interpretation. Each summand has similar structure to the partial derivative of $p(z | \mathcal{H}(i, t))$ (Eq. 42), but the weight conditioned on only the clinical history has been replaced with a weight conditioned on both the clinical history *and* the future target marker trajectory. The partial derivatives of the summands of the objective in Eq. 33 are added together to obtain the partial derivative with respect to the objective. These partial derivatives are combined to form a gradient, which is easily plugged into existing first-order optimization routines. Optionally, the objective can be augmented with a regularizer to restrict the complexity of the model or to encourage a sparse solution to the learning problem. This concludes our discussion of the learning algorithm.

Appendix B. Missing Data for Continuous-Time Trajectories

Trajectories in continuous-time can be thought of as random *functions* $F(\cdot)$ (Gaussian processes are an example of a family of distributions over random functions). Although the function specifies infinitely many values, to learn continuous-time models we maximize the probability of a finite set of observations (or a penalized version of this objective). In *observational* health care data, we need to be careful that we do not bias our likelihood-based learners by unduly ignoring the dependence between the finite set of times at which we observe the trajectory and the trajectory's values at those times. For example, if the trajectory is more likely to be sampled when its value is low, then our model will learn that trajectories with high values are less likely than they actually are.

The aim of this section is to posit a set of assumptions about continuous trajectory observation times that are (1) substantively reasonable, and (2) justify the use of standard likelihood-based learning. At a high-level, we assume that trajectory observation times are functions of the previous observation times and the values of the trajectory sampled at those times. These assumptions are more formally encoded in the graphical model shown in Figure 8, which expresses dependencies for an individual with three trajectory observations. In the figure, $F(\cdot)$ denotes the full trajectory, $\{T_1, T_2, T_3\}$ are random variables denoting the times at which the trajectory is sampled, and $\{Y_1^*, Y_2^*, Y_3^*\}$ are the observed data. The conditional probability distribution of any Y_t^* given the trajectory and associated observation time is simply:

$$p(Y_t^* = y_t^* | T_t = t_i, F = f) = \mathbb{I}(f(t_i) = y_t^*). \quad (46)$$

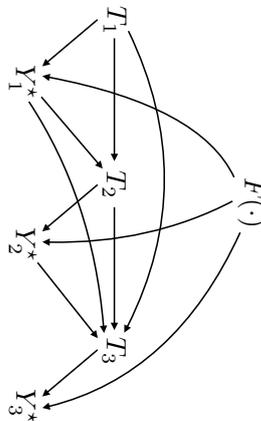


Figure 8: Example missing data mechanism in continuous-time.

These assumptions are reasonable in many healthcare settings. For example, in an ICU where a patient is constantly under supervision, we can reasonably assume that clinical marker measurements are made at times that depend on the previous observations (e.g. the individual is thought to be at risk and so measurements are taken more frequently) and on previous observation times (e.g. a measurement has not been recorded in a while, so we should collect a new one). In the outpatient setting, an individual with a particular disease that is being actively managed by a physician will have follow-up visits scheduled either routinely or more frequently if the physician is especially concerned. On the other hand, modeling the progression of a disease such as the flu using information from a general practitioner’s office may not satisfy our assumption because individual’s with less severe manifestations are less likely to visit.

Conditioned on these assumptions about the dependencies between the trajectory, observation times, and observed values, we want to justify likelihood-based learning. Suppose we have a trajectory model with parameters Θ that allows us to compute the probability of any finite set of trajectory values. For example, we can compute $p_{\Theta}(F(t_1) = y_1^*, F(t_2) = y_2^*, F(t_3) = y_3^*)$. The observed data, however, are the observation times and sampled values: $\{T_{1:n}, Y_{1:n}^*\}$. Proper likelihood-based learning requires that we maximize:

$$p(T_{1:n} = t_{1:n}, Y_{1:n}^* = y_{1:n}^*). \quad (47)$$

However, this expression is determined by both the observation time mechanism and the trajectory model. Our goal is to show that this can be factored into two terms: one that depends on the observed data and the observation time mechanism parameters, and the other that depends on the sampled trajectory values and the trajectory model parameters Θ . To do this, we first see that Equation 47 can be written as

$$\int p(F = f)p(T_{1:n} = t_{1:n}, Y_{1:n}^* = y_{1:n}^* | F = f)dF. \quad (48)$$

The integrand in Equation 48 can be now be factored further to obtain

$$p(F = f) \prod_{i=1}^n p(T_i = t_i | \mathcal{H}_i)p(Y_i^* = y_i^* | T_i = t_i, F = f), \quad (49)$$

where \mathcal{H}_i is defined to be the previous $i - 1$ observation times and sampled trajectory values. Note that the first term in the product of Equation 49 can be pulled out of the integral, allowing us to write Equation 48 as

$$\left[\prod_{i=1}^n p(T_i = t_i | \mathcal{H}_i) \right] \left[\int p(F = f) \prod_{i=1}^n p(Y_i^* = y_i^* | T_i = t_i, F = f)dF \right]. \quad (50)$$

The left-hand factor above depends only on the observation time mechanism and the observed data. Moreover, the right-hand factor depends only on the trajectory model and the sampled trajectory values, which we now show:

$$\begin{aligned} \int p(F = f) \prod_{i=1}^n p(Y_i^* = y_i^* | T_i = t_i, F = f)dF \\ &= \int p(F = f) \prod_{i=1}^n \mathbb{I}(f(t_i) = y_i^*)dF \\ &= \int p(F = f) \mathbb{I}(f(t_1) = y_1^*, \dots, f(t_n) = y_n^*)dF \\ &= p_{\Theta}(f(t_1) = y_1^*, \dots, f(t_n) = y_n^*). \end{aligned} \quad (51)$$

We therefore see that, given our observation time mechanism assumptions, maximizing the likelihood of the sampled trajectory values under our trajectory model is equivalent to maximizing the “proper” likelihood in Equation 47 with respect to the model parameters Θ . This result aligns with Theorems 7.1 and 8.1 found in Rubin’s original paper on missing data (Rubin, 1976).

References

- Y. Allanore, R. Simms, O. Distler, M. Trojanowska, J. Pope, C.P. Denton, and J. Varga. Systemic sclerosis. *Nature Reviews Disease Primers*, 2015.
- G. Andrew and J. Gao. Scalable training of ℓ_1 -regularized log-linear models. In *International Conference on Machine Learning (ICML)*, 2007.
- K.H. Brodersen, F. Gallusser, J. Koehler, N. Remy, and S.L. Scott. Inferring causal impact using bayesian structural time-series models. *The Annals of Applied Statistics*, 9(1): 247–274, 2015.
- S. Chib and B.H. Hamilton. Semiparametric bayes analysis of longitudinal data treatment models. *Journal of Econometrics*, 110(1):67–89, 2002.
- R.Y. Coley, A.J. Fisher, M. Mamawala, H.B. Carter, K.J. Pienta, and S.L. Zeger. A bayesian hierarchical model for prediction of latent health states from multiple data sources with application to active surveillance of prostate cancer. *Biometrics*, 2016.
- F.S. Collins and H. Varmus. A new initiative on precision medicine. *New England Journal of Medicine*, 372(9):793–795, 2015.

- J. Craig. Complex diseases: Research and applications. *Nature Education*, 1(1):184, 2008.
- A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (methodological)*, pages 1–38, 1977.
- F. Doshi-Velez, Y. Ge, and I. Kohane. Comorbidity clusters in autism spectrum disorders: an electronic health record time-series analysis. *Pediatrics*, 133(1):e54–e63, 2014.
- R. Dürichen, M.A.F. Pimentel, L. Clifton, A. Schweikard, and D.A. Clifton. Multitask gaussian processes for multivariate physiological time-series analysis. *Biomedical Engineering, IEEE Transactions on*, 62(1):314–322, 2015.
- P.H.C. Eilers and B.D. Marx. Flexible smoothing with b-splines and penalties. *Statistical Science*, pages 89–102, 1996.
- J. Friedman, T. Hastie, and R. Tibshirani. *The Elements of Statistical Learning*. Springer, 2001.
- A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin. *Bayesian Data Analysis*. Taylor & Francis, 2014.
- M.R. Hassan and B. Nath. Stock market forecasting using hidden markov model: a new approach. In *Intelligent Systems Design and Applications, 5th International Conference on*, pages 192–196. IEEE, 2005.
- G.M. James and C.A. Sugar. Clustering for sparsely sampled functional data. *Journal of the American Statistical Association*, 98(462):397–408, 2003.
- D. Khanna, C.H. Tseng, N. Farnmani, V. Steen, D.E. Furst, P.J. Clements, M.D. Roth, J. Goldin, R. Elashoff, J.R. Seibold, R. Saggat, and D.P. Tashkin. Clinical course of lung physiology in patients with scleroderma and interstitial lung disease: analysis of the scleroderma lung study placebo group. *Arthritis & Rheumatism*, 63(10):3078–3085, 2011.
- S. Kleinberg and G. Hripcsak. A review of causal inference for biomedical informatics. *Journal of Biomedical Informatics*, 44(6):1102–1112, 2011.
- J.M. Lange, R.A. Hubbard, L.Y.T. Inoue, and V.N. Minin. A joint model for multistate disease processes and random informative observation times, with applications to electronic medical records data. *Biometrics*, 71(1):90–101, 2015.
- M. Lázaro-Gredilla, S. Van Vaerenbergh, and N.D. Lawrence. Overlapping mixtures of gaussian processes for the data association problem. *Pattern Recognition*, 45(4):1386–1395, 2012.
- D.S. Lee, P.C. Austin, J.L. Rouleau, P.P. Liu, D. Naimark, and J.V. Tu. Predicting mortality among patients hospitalized for heart failure: derivation and validation of a clinical model. *Journal of the American Medical Association*, 290(19):2581–2587, 2003.
- S.J.G. Lewis, T. Foltynie, A.D. Blackwell, T.W. Robbins, A.M. Owen, and R.A. Barker. Heterogeneity of parkinsons disease in the early clinical stages using a data driven approach. *Journal of Neurology, Neurosurgery & Psychiatry*, 76(3):343–348, 2005.
- R.J.A. Little and D.B. Rubin. *Statistical Analysis with Missing Data*. John Wiley & Sons, 2014.
- Z. Liu and M. Hauskrecht. Clinical time series prediction: Toward a hierarchical dynamical system framework. *Artificial Intelligence in Medicine*, 2014.
- J. Lötvall, C.A. Akdis, L.B. Bacharier, L. Bjermer, T.B. Casale, A. Custovic, R.F. Lemanske, A.J. Wardlaw, S.E. Wenzel, and P.A. Greenberger. Asthma endotypes: a new approach to classification of disease entities within the asthma syndrome. *Journal of Allergy and Clinical Immunology*, 127(2):355–360, 2011.
- K.P. Murphy. *Dynamic bayesian networks: representation, inference and learning*. PhD thesis, University of California, Berkeley, 2002.
- K.P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.
- J.B. Oliva, W. Neiswanger, B. Póczos, E.P. Xing, H. Trac, S. Ho, and J.G. Schneider. Fast function to function regression. In *Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.
- C. Proust-Lima, M. Séne, J.M.G Taylor, and H. Jacquin-Gadda. Joint latent class models for longitudinal and time-to-event data: A review. *Statistical Methods in Medical Research*, 23(1):74–90, 2014.
- J.A. Quinn, C.K. Williams, and N. McIntosh. Factorial switching linear dynamical systems applied to physiological condition monitoring. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(9):1537–1551, 2009.
- R. Raina, Y. Shen, A. McCallum, and A.Y. Ng. Classification with hybrid generative/discriminative models. In *Advances in Neural Information Processing Systems (NIPS)*, 2003.
- J.O. Ramsay. *Functional Data Analysis*. Wiley Online Library, 2006.
- C.E. Rasmussen and C.K. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.
- D. Rizopoulos. Dynamic predictions and prospective accuracy in joint models for longitudinal and time-to-event data. *Biometrics*, 67(3):819–829, 2011.
- D. Rizopoulos and P. Ghosh. A bayesian semiparametric multivariate joint model for multiple longitudinal outcomes and a time-to-event. *Statistics in Medicine*, 30(12):1366–1380, 2011.
- S. Roberts, M. Osborne, M. Ebdon, S. Reece, N. Gibson, and S. Aigrain. Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984):20110550, 2013.

- K.R. Rosenbloom, C.A. Sloan, V.S. Malladi, T.R. Dreszer, K. Learned, V.M. Kirkup, M.C. Wong, M. Madden, R. Fang, S.G. Heitner, B.T. Lee, G.P. Barber, R.A. Harte, M. Diekhans, J.C. Long, S.P. Wilder, A.S. Zweig, D. Karolchik, R.M. Kuhn, D. Haussler, and W.J. Kent. Encode data in the UCSC genome browser: year 5 update. *Nucleic acids research*, 41(D1):D56–D63, 2013.
- D.B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.
- S. Saria and A. Goldenberg. Subtyping: What Is It and Its Role in Precision Medicine. *IEEE Intelligent Systems*, 30, 2015.
- P.F. Schuhman and S. Saria. A framework for individualizing predictions of disease trajectories by exploiting multi-resolution structure. In *Advances in Neural Information Processing Systems (NIPS)*, pages 748–756, 2015.
- P.F. Schuhman, F. Wigley, and S. Saria. Clustering longitudinal clinical marker trajectories from electronic health data: Applications to phenotyping and endotype discovery. In *Conference on Artificial Intelligence (AAAI)*, 2015.
- J.Q. Shi, R. Murray-Smith, and D.M. Titterton. Hierarchical gaussian process mixtures for regression. *Statistics and Computing*, 15(1):31–41, 2005.
- J.Q. Shi, B. Wang, E.J. Will, and R.M. West. Mixed-effects gaussian process functional regression models with application to dose–response curve prediction. *Statistics in Medicine*, 31(26):3165–3177, 2012.
- D.P. Tashkin et al. Cyclophosphamide versus placebo in scleroderma lung disease. *New England Journal of Medicine*, 354(25):2655–2666, 2006.
- J. Varga, C.P. Denton, and F.M. Wigley. *Scleroderma: From Pathogenesis to Comprehensive Management*. Springer Science & Business Media, 2012.
- H. Wang, F. Nie, H. Huang, J. Yan, S. Kim, S. Risacher, A. Saykin, and L. Shen. High-order multi-task feature learning to identify longitudinal phenotypic markers for alzheimer’s disease progression prediction. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1277–1285, 2012.
- Y. Xu, Y. Xu, and S. Saria. A bayesian nonparametric approach for estimating individualized treatment–response curves. *arXiv preprint arXiv:1608.05182*, 2016.
- J. Zhou, L. Yuan, J. Liu, and J. Ye. A multi-task learning formulation for predicting disease progression. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 814–822, 2011.

Structure-Leveraged Methods in Breast Cancer Risk Prediction

Jun Fan

*Department of Statistics
University of Wisconsin-Madison
1300 University Avenue, Madison, WI 53706, United States*

JUNFAN@STAT.WISC.EDU

Yirong Wu

*Department of Radiology
University of Wisconsin-Madison
600 Highland Avenue, Madison, WI 53792, United States*

YWU@UWHEALTH.ORG

Ming Yuan

*Department of Statistics
University of Wisconsin-Madison
1300 University Avenue, Madison, WI 53706, United States*

MYUAN@STAT.WISC.EDU

David Page

*Department of Biostatistics and Medical Informatics
University of Wisconsin-Madison
600 Highland Avenue, Madison, WI 53792, United States*

PAGE@BIostat.WISC.EDU

Jie Liu

*Department of Genome Sciences
University of Washington-Seattle
3720 15th Avenue, Seattle, WA 98105, United States*

LIUJ@UW.EDU

Irene M. Ong

*Department of Biostatistics and Medical Informatics
University of Wisconsin-Madison
600 Highland Avenue, Madison, WI 53792, United States*

ONG@CS.WISC.EDU

Peggy Peissig

*Marshfield Clinic Research Foundation
1000 North Oak Avenue, Marshfield, WI 54449, United States*

PEISSIG.PEGGY@MCRF.MFLDCLIN.EDU

Elizabeth Burnside

*Department of Radiology
University of Wisconsin-Madison
600 Highland Avenue, Madison, WI 53792, United States*

EBURNSIDE@UWHEALTH.ORG

Editor: Benjamin M. Marlin and Suchi Sarin

Abstract

Predicting breast cancer risk has long been a goal of medical research in the pursuit of precision medicine. The goal of this study is to develop novel penalized methods to improve breast cancer risk prediction by leveraging structure information in electronic health

records. We conducted a retrospective case-control study, garnering 49 mammography descriptors and 77 high-frequency/low-penetrance single-nucleotide polymorphisms (SNPs) from an existing personalized medicine data repository. Structured mammography reports and breast imaging features have long been part of a standard electronic health record (EHR), and genetic markers likely will be in the near future. Lasso and its variants are widely used approaches to integrated learning and feature selection, and our methodological contribution is to incorporate the dependence structure among the features into these approaches. More specifically, we propose a new methodology by combining group penalty and l^p ($1 \leq p \leq 2$) fusion penalty to improve breast cancer risk prediction, taking into account structure information in mammography descriptors and SNPs. We demonstrate that our method provides benefits that are both statistically significant and potentially significant to people's lives.

Keywords: structure information, breast cancer risk prediction, mammography descriptors, genetic variants, personalized medicine

1. Introduction

Breast cancer is the most common non-skin malignancy affecting women, with approximately 1.67 million cases diagnosed annually worldwide (Ferlay et al., 2013). If an individual's risk of breast cancer could be predicted, then screening, prevention, and treatment strategies could be targeted toward those women to maximize survival benefit and minimize harm. Risk prediction models are important tools to improve breast cancer care by leveraging multi-dimensional electronic health data. Traditional breast cancer risk prediction models use demographic risk factors to estimate breast cancer risk, but they demonstrate only limited discriminatory power. In clinical practice, mammography is the most common breast cancer screening test, and the only imaging modality supported by randomized trials demonstrating reduction in mortality rate. However, its effectiveness is not universally accepted (Freedman et al., 2004). Recent advances in genome-wide association studies (GWAS) have revitalized the quest for genetic variants (single-nucleotide polymorphisms—SNPs) in risk prediction. However, the optimism of these studies has been tempered by disappointment and caution (Gail, 2008, 2009; Wacholder et al., 2010).

Although many breast cancer risk prediction models have been developed, current applications of these models are inadequate in the following respects: (1) due to the rare occurrence of breast cancer, many seemingly 'large' studies have small effective sample size to adequately model a large number of variables; (2) even for large studies, investigators often fail to systematically model risk factor interactions to avoid overly complicated models which are hard to interpret; and (3) they do not take available structure information into consideration. For example, there are five descriptors for mass margins in mammogram: circumscribed, microlobulated, obscured, indistinct, and spiculated, with an order of increasing probability of malignancy. However, few models utilize this structure information (group structure and dependence structure) to improve predictive performance. The quest for novel breast cancer risk prediction models is motivated to address these shortcomings.

In this paper, we propose to develop novel penalized methods to improve breast cancer risk prediction by incorporating unique structure information embedded in electronic health record data. Regularization is a common technique used in regression and classification problems. The lasso (Tibshirani, 1996) is one of the most popular penalized method and

SNP	Chr	SNP	Chr
rs616488	1	rs11814448	10
rs11249433	1	rs7072776	10
rs1550623	2	rs7904519	10
rs16857609	2	rs2981582	10
rs2016394	2	rs10995190	10
rs4849887	2	rs2380205	10
rs1045485	2	rs2981579	10
rs13387042	2	rs704010	10
rs17468277	2	rs11820646	11
rs4666451	2	rs3903072	11
rs12493607	3	rs3817198	11
rs6762644	3	rs2107425	11
rs4973768	3	rs614367	11
rs6828523	4	rs12422552	12
rs9790517	4	rs17356907	12
rs10472076	5	rs6220	12
rs1353747	5	rs10771399	12
rs1432679	5	rs1292011	12
rs10941679	5	rs11571833	13
rs889312	5	rs2236007	14
rs30099	5	rs2588809	14
rs981782	5	rs941764	14
rs10069690	5	rs999737	14
rs11242675	6	rs13329835	16
rs204247	6	rs17817449	16
rs2046210	6	rs3803662	16
rs2180341	6	rs12443621	16
rs17530068	6	rs8051542	16
rs3757318	6	rs6504950	17
rs720475	7	rs1436904	18
rs11780156	8	rs527616	18
rs2943559	8	rs3760982	19
rs6472903	8	rs4808801	19
rs9693444	8	rs8170	19
rs13281615	8	rs2284378	20
rs10759243	9	rs2823093	21
rs1011970	9	rs132390	22
rs865686	9	rs6001930	22
rs11199914	10		

Table 2: The 77 SNPs identified to be associated with breast cancer

2.1.3 GENETIC VARIANTS

We decided to focus on high-frequency/low-penetrance SNPs that affect breast cancer risk as opposed to low frequency SNPs with high penetrance or intermediate penetrance. We consolidated a list of 77 common genetic variants (Table 2) which were identified by recent large-scale GWAS studies or used to generate published predictive models (Liu et al., 2014). The list included 41 SNPs identified by COGS through a meta-analysis of 9 GWAS studies (Michailidou et al., 2013). Recently, a similar set of 77 breast cancer-associated SNPs is also studied for risk prediction (Mavaddat et al., 2015).

2.2 Logistic Regression

Assume that we have independent and identical distributed subjects $\{(x_i, y_i)\}_{i=1}^n$, where the explanatory variable $X \in \mathcal{R}^d$ and the binary response variable $Y \in \{-1, 1\}$. Note that the conditional probability $\eta(x) = \mathbb{P}(Y = 1|X = x)$ plays an important role in the classification problem. Denote $x_i = (x_{i1}, \dots, x_{id})^T$, and linear logistic regression model is defined by

$$\log \frac{\eta(x_i)}{1 - \eta(x_i)} = x_i^T \beta, \quad i = 1, \dots, n,$$

where $\beta = (\beta_1, \dots, \beta_d)^T$ is the slope parameter. And the logistic regression estimator $\hat{\beta}$ is given by the minimizer of the negative log-likelihood function

$$L(\beta) = \frac{1}{n} \sum_{i=1}^n \log(1 + \exp(-y_i \cdot x_i^T \beta)). \quad (1)$$

With $\hat{\beta}$, we then estimate the conditional probability $\eta(x_i)$ by

$$\hat{\eta}(x_i) = \frac{\exp(x_i^T \hat{\beta})}{1 + \exp(x_i^T \hat{\beta})} = \frac{1}{1 + \exp(-x_i^T \hat{\beta})}.$$

Then we should predict $y_i = 1$ if $\hat{\eta}(x_i) \geq 0.5$ and $y_i = -1$ if $\hat{\eta}(x_i) < 0.5$.

2.3 Group Penalty and ℓ^p Fusion Penalty

Note that there exist natural group structure and dependence structure in mammography features (Figure 1), which allows us to include the structure information into our risk prediction models directly. For genetic variants, group structures also exist (Liu et al., 2012, 2013). In this paper, we apply hierarchical clustering to cluster the 77 SNPs based on their dissimilarity matrix obtained by computing Spearman's correlation or Hamming distance among them. More details are provided in Section 2.5.

Suppose that d features are divided into G groups with d_g the number of features in group g . Define $\beta_g \in \mathbb{R}^{d_g}$ to be the corresponding coefficient vector in group g . The group lasso logistic regression (Meier et al., 2008) is defined as the following optimization problem

$$\min_{\beta \in \mathbb{R}^d} \left\{ L(\beta) + \lambda_1 \sum_{g=1}^G \sqrt{d_g} \|\beta_g\|_2 \right\},$$

where $L(\beta)$ is defined by (1) and $\lambda_1 \geq 0$ is the tuning parameter. It includes lasso as a special case with $G = d$.

The fact that there exist dependence structure within each mammography feature group and each SNP group encourages us to propose the following novel method by combining group lasso logistic regression and ℓ^{p_1} fusion penalty.

$$\min_{\beta \in \mathbb{R}^d} \left\{ L(\beta) + \sum_{g=1}^G \left(\lambda_1 \sqrt{d_g} \|\beta_g\|_2 + \lambda_2 \|D_g \beta_g\|_{p_2} \right) \right\}, \quad (2)$$

where D_g is a $(d_g - 1) \times d_g$ sparse matrix with only $D[i, j] = 1$ and $D[i, i+1] = -1$, $\lambda_2 \geq 0$ is the tuning parameter, and $1 \leq p \leq 2$ is the shrinkage parameter.

Moreover, if the within-group dependence structures are different for groups $\{1, \dots, G_1\}$ and $\{G_1 + 1, \dots, G\}$, we can split the ℓ^{p_1} fusion penalty into two parts as

$$\min_{\beta \in \mathbb{R}^d} \left\{ L(\beta) + \lambda_1 \sum_{g=1}^G \sqrt{d_g} \|\beta_g\|_2 + \lambda_2 \left(\sum_{g=1}^{G_1} \|D_g \beta_g\|_{p_1} + \sum_{g=G_1+1}^G \|D_g \beta_g\|_{p_2} \right) \right\}, \quad (3)$$

where $1 \leq p_1, p_2 \leq 2$ are selected based on cross validation.

The novelty of our method compared to previous works is three-fold: First, it includes within-group fusion penalty in the model and makes the coefficients of features in the same group close to each other, which reflects the dependence structure of features and improves the risk prediction; Second, in breast cancer risk prediction, we find that the dependence structures are different for mammography features and SNPs, which are actually two different views of the same data. And the utilization of method (3) will improve the predictive performance further; At last, we find that genetic variants improve risk prediction on mammography features, which provides some insight regarding personalized breast cancer diagnosis.

2.4 Computational Algorithms

Many algorithms have been proposed in the literatures to solve the logistic regression with fused lasso regularization (Lin, 2015; Yu et al., 2015). In this subsection we adopt the fast iterative shrinkage thresholding algorithm (Beck and Teboulle, 2009) to solve (2) as

$$\beta^{k+1} = \arg \min_{\beta \in \mathbb{R}^d} L(\beta^k) + \langle \beta - \beta^k, \nabla L(\beta^k) \rangle + \frac{\tau}{2} \|\beta - \beta^k\|_2^2 + \sum_{g=1}^G \left(\lambda_1 \sqrt{d_g} \|\beta_g\|_2 + \lambda_2 \|D_g \beta_g\|_{p_2} \right)$$

with $\beta = (\beta_1, \dots, \beta_d)^T$ and $\tau > 0$ the Lipschitz constant of $L(\cdot)$.

And the iteration step is equivalent to solving

$$\min_{\beta \in \mathbb{R}^d} \left\{ \frac{1}{2} \|\beta - (\beta^k - \frac{1}{\tau} \nabla L(\beta^k))\|_2^2 + \sum_{g=1}^G \left(\frac{\lambda_1 \sqrt{d_g}}{\tau} \|\beta_g\|_2 + \frac{\lambda_2}{\tau} \|D_g \beta_g\|_{p_2} \right) \right\}. \quad (4)$$

Therefore, it suffices to solve the following optimization problem within each group

$$\min_{\beta_g \in \mathbb{R}^{d_g}} \left\{ \frac{1}{2} \|\beta_g - z\|_2^2 + \rho_1 \|\beta_g\|_2 + \rho_2 \|D_g \beta_g\|_{p_2} \right\}, \quad (5)$$

where $z = \beta_g^k - \frac{1}{\tau} \nabla L(\beta_g^k)$, $\rho_1 = \frac{\lambda_1 \sqrt{d_g}}{\tau}$ and $\rho_2 = \frac{\lambda_2}{\tau}$.

The proximity operator (Polson et al., 2015) of a function f is defined as

$$P_f(z) = \arg \min_t \left\{ \frac{1}{2} \|t - z\|^2 + \lambda f(t) \right\}.$$

- For $f(t) = |t|$ and $z \in \mathbb{R}$, $P_f(z) := S_1(z, \lambda) = \text{sign}(z) \max\{|z| - \lambda, 0\}$, which is also called soft threshold operator.
- For $f(t) = |t|^p$ with $1 < p \leq 2$ and $z \in \mathbb{R}$, $P_f(z) := S_p(z, \lambda) = \text{sign}(z) \xi$, where ξ is the unique nonnegative solution to $\xi + p \lambda \xi^{p-1} = |z|$. In particular, we have $S_2(z, \lambda) = \frac{z + \sqrt{z^2 + 8\lambda^2}}{2\sqrt{2}}$, $S_{3/2}(z, \lambda) = z + 9\lambda^2 \text{sign}(z) (1 - \sqrt{1 + 16|z|/(9\lambda^2)})/8$ and $S_{4/3}(z, \lambda) = z + \frac{4\lambda}{3\sqrt{3}} ((\chi - z)^{1/3} - (\chi + z)^{1/3})$ with $\chi = \sqrt{z^2 + 256\lambda^3/729}$.
- For $f(t) = \|t\|_2$ and $z \in \mathbb{R}^d$, $P_f(z) := S_{2,1}(z, \lambda) = \max\{1 - \frac{\lambda}{\|z\|_2}, 0\} * z$.

With the help of these proximity operators and Bregman splitting algorithm (Ye and Xie, 2011), we can solve (5) by iteratively solving the following procedures:

$$\begin{cases} \beta_g^{k+1} = \arg \min_{\beta_g} \frac{1}{2} \|\beta_g - z\|_2^2 + \langle u^k, \beta_g - a^k \rangle + \langle v^k, D_g \beta_g - b^k \rangle \\ \quad + \frac{\mu}{2} \|\beta_g - a^k\|_2^2 + \frac{\mu}{2} \|D_g \beta_g - b^k\|_2^2 \\ a^{k+1} = \arg \min_{a} \rho_1 \|a\|_2 + \langle u^k, \beta^{k+1} - a \rangle + \frac{\mu}{2} \|\beta^{k+1} - a\|_2^2 \\ b^{k+1} = \arg \min_b \rho_2 \|b\|_{p_2} + \langle v^k, D_g \beta^{k+1} - b \rangle + \frac{\mu}{2} \|D_g \beta^{k+1} - b\|_2^2 \\ u^{k+1} = u^k + \mu (\beta^{k+1} - a^{k+1}) \\ v^{k+1} = v^k + \mu (D_g \beta^{k+1} - b^{k+1}) \end{cases}$$

where μ acts like a step size in this algorithm.

Remark 1 The minimization over β , a and b can all be solved in closed form.

- $\beta^{k+1} = [(\mu + 1)I + \mu D_g^T D_g]^{-1} [z + \mu(a^k - u^k/\mu) + \mu D_g^T (b^k - v^k/\mu)]$
- $a^{k+1} = S_{2,1}(\beta^{k+1} + u^k/\mu, \rho_1/\mu)$
- $b^{k+1} = S_{p_2}(\beta^{k+1} + v^k/\mu, \rho_2/\mu)$

Note that $(\mu + 1)I + \mu D_g^T D_g$ is a triagonal positive definite matrix.

Remark 2 For $p = 1$, we can solve (5) more efficiently by the algorithm proposed in Zhou et al. (2012) based on the fact

$$P_{\|\cdot\|_2 + \|D_g(\cdot)\|_1} = P_{\|\cdot\|_2} \circ P_{\|D_g(\cdot)\|_1}.$$

However, we cannot show this equation for $1 < p \leq 2$.

Remark 3 For $p = 2$, since $\|\cdot\|_2^2$ is Lipschitz continuous, we can rewrite (4) as

$$\min_{\beta \in \mathbb{R}^d} \left\{ \frac{1}{2} \left\| \beta - \left(\beta^k - \frac{1}{\bar{\tau}} (\nabla L(\beta^k) + 2\lambda_2 \sum_{g=1}^G D_g^T D_g \beta^k) \right) \right\|_2^2 + \sum_{g=1}^G \frac{\lambda_1 \sqrt{d_g}}{\bar{\tau}} \|\beta_g\|_2 \right\},$$

where $\bar{\tau}$ is the Lipschitz constant of $L(\beta) + \lambda_2 \sum_{g=1}^G \|D_g \beta_g\|_2^2$. Then we can solve it efficiently via the proximity operator of $\|\cdot\|_2$.

2.5 Study Design and Statistical Analysis

We apply the ℓ^p fused group lasso logistic regression algorithm to the Marshfield breast cancer data set. There are 11 groups for 49 mammography features (Figure 1). For SNPs, we compute the Hamming distances (Wang et al., 2015) of 77 SNPs to get the dissimilarity matrix and then apply hierarchical clustering to obtain 10 groups.

We built three prediction models based on different sets of risk factors: the Mammo model developed by using mammography features only, the SNP77 model developed by using 77 SNPs only, and the Combined model developed by using both mammography features and 77 SNPs. We furthermore apply five methods for each model: logistic regression (LR), lasso in logistic regression (LR+Lasso), ℓ^p fused lasso logistic regression (LR+fusedLasso), group lasso logistic regression (LR+groupLasso), and ℓ^p fused group lasso logistic regression (LR+Structure).

The ℓ^p fused group lasso logistic regression method has several parameters. For the tuning parameters λ_1 and λ_2 , we let them vary among a given set of values, and the shrinkage parameter p (or p_1 and p_2) among $\{1, 4/3, 3/2, 2\}$. Each combination of these parameters is evaluated using stratified 5-fold cross-validation, and AUC (the area under the receiver operating characteristic (ROC) curve) is used as the performance measure. All 738 samples are randomly partitioned into five equal sized folds with approximately equal proportions of cases and controls. In each iteration (totally five iterations), four folds are used as training set and the rest one as validation set to compute AUC. And the parameters with the best average AUC are selected. At last we repeat this process ten times and report the average AUC. We obtain p-value by performing two-tailed two-sample t-test when we compare AUCs.

3. Experimental Results

In this section, we demonstrate the performance of the ℓ^p fused group lasso logistic regression method from three aspects: the significant improvement of AUCs by considering the structure information, the predictive performance under different p (or p_1 and p_2), and the detected important mammography features and SNPs.

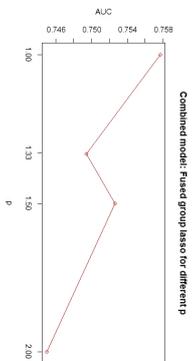
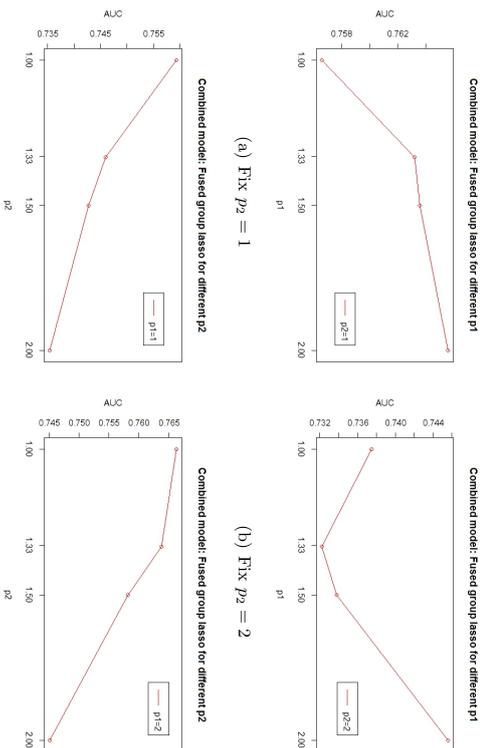
3.1 Performance of Fused Group Lasso

The result is summarized in Table 3.

Models/Methods	LR	Lasso	fusedLasso	groupLasso	Structure	p-value
Mammo	0.700	0.710	0.710	0.716	0.723	< 0.001
SNP77	0.590	0.598	0.676	0.614	0.684	< 0.001
Combined	0.697	0.721	0.754	0.727	0.766	< 0.001

Table 3: Predictive performance of three prediction models by using five different methods. The p-values represent the differences between AUCs of LR and LR+Structure.

- 1) The fifth column describes the predictive performance of the three prediction models by considering structure information in the logistic regression method. We find that the predictive performance of the three prediction models has been improved respectively, compared to those described in the first column. For each prediction model, the difference of the predictive performance is significant between LR+Structure and LR (p-value < 0.001), which demonstrates that breast cancer prediction models utilizing structure information can improve risk prediction significantly. We also find that mammography descriptors demonstrate a significantly higher predictive performance than 77 SNPs in terms of AUC (0.723 vs. 0.684, p-value < 0.001). The Combined model demonstrates significant improvement of the prediction performance, compared to the Mammo model (0.766 vs. 0.723, p-value < 0.001).
- 2) The first column describes the predictive performance of the three prediction models by using the logistic regression method. Mammography descriptors demonstrate a significantly higher predictive performance than 77 SNPs in terms of AUC (0.700 vs. 0.590, p-value < 0.001). We find that the difference of predictive performance between the Combined model and the Mammo model is negligible (0.697 vs. 0.700, p-value=0.277).
- 3) The second column describes the predictive performance of the three prediction models by using lasso in the logistic regression method. The predictive performance of the three prediction models has been improved, compared to those without lasso (using logistic regression method only). Mammography descriptors still demonstrate a significantly higher predictive performance than 77 SNPs in terms of AUC (0.710 vs. 0.598, p-value < 0.001). However, the Combined model demonstrates modest improvement of prediction performance, compared to the Mammo model (0.721 vs. 0.710, p-value=0.0057).
- 4) The third and fourth columns describe the predictive performance of the three prediction models by considering group structure or dependence structure in the logistic regression method. For the SNP77 model, fused lasso demonstrates a significantly higher performance than group lasso in terms of AUC (0.676 vs. 0.614, p-value < 0.001). For the Mammo model, group lasso plays a more important role than fused lasso (0.716 vs. 0.710, p-value=0.0073). Moreover, both fused lasso and group lasso demonstrate improved prediction performance compared to lasso.

Figure 2: The AUCs under different values of p by using method (2).Figure 3: The AUCs under different values of p_1 and p_2 by using method (3).

3.2 Performance under Different Values of p

Figure 2 and Figure 3 describe the the pattern of predictive performance for p' fused group lasso logistic regression over the shrinkage parameter p (or p_1 and p_2) in terms of AUC.

- 1) The Combined model demonstrates a higher predictive performance for $p = 1$ compared to $p = 2$ in terms of AUC (0.757 vs. 0.745, p -value < 0.001), see Figure 2.
- 2) Figure 3 describes the prediction performance of method (3) under different values of p_1 for mammography descriptors and p_2 for 77 SNPs.

FAN, WU, YUAN, PAGE, LIU, ONG, PEISSIG, AND BURNSIDE

- Fix $p_2 = 1$ or $p_2 = 2$, the fused group lasso with $p_1 = 2$ demonstrates higher predictive performance compared to $p_1 = 1$, see Figure 3(a) and 3(b).
- Fix $p_1 = 1$ or $p_1 = 2$, the predictive performance of the fused group lasso logistic regression decreases as p_2 increases, see Figure 3(c) and 3(d).
- The fused group lasso logistic regression with $p_1 = 2$ and $p_2 = 1$ demonstrates higher predictive performance than $p_1 = p_2 = 1$ (0.766 vs. 0.757, p -value=0.0053) and $p_1 = p_2 = 2$ (0.766 vs. 0.745, p -value < 0.001).

3.3 Important Features Detected by Fused Group Lasso

To take into account both group and dependence structure information in mammography features and SNPs, two penalty terms (group penalty and fusion penalty) are introduced into the logistic regression model. The idea of group penalty is to force the coefficients of features in the same group to be all zero or nonzero in order to achieve the goal of selecting features within a group simultaneously. The idea of fusion penalty is to shrink the successive difference of coefficients of features in the same group in order to take advantage of the dependence structure information. Applying fusion penalty with $p = 1$ tends to result in zero successive difference of coefficients, while $p = 2$ tends to small but nonzero successive difference of coefficients.

From a feature selection point of view, we can get the order of feature groups selected by fused group lasso via choosing the tuning parameters appropriately. We list below the feature groups selected from high to low in terms of predictive performance.

- 1) For mammography descriptors, the following features are predictive of malignancy (from most to least): ‘‘Mass Size’’, ‘‘Mass Margins’’, ‘‘Mass Shape’’, ‘‘Architectural Distortion’’ and ‘‘Mass Palpability’’, consistent with the literature (BI-RADS, 2014).
- 2) For 77 SNPs, three groups are selected in order, see Table 4.

Feature Group	SNPs
Group 1	rs2016394, rs1432679, rs13281615, rs4666451 rs981782, rs1292011, rs1436904, rs527616
Group 2	rs11249433, rs13387042, rs4973768, rs10069690 rs7904519, rs8051542, rs3760982
Group 3	rs2981579, rs2981582

Table 4: SNP groups selected by fused group lasso.

Remark 4 It verifies that ‘‘Mass size’’, ‘‘Mass Margins’’ and ‘‘Mass Shape’’ are the most important mammography descriptors in breast cancer diagnosis. These results are consistent with previous studies about comparing the importance of mammography features and SNPs in breast cancer risk prediction (Wu et al., 2013, 2014).

4. Discussion and Conclusions

This study demonstrates that models utilizing the novel combination of clinically relevant structure and ℓ^p fused group lasso logistic regression can improve breast cancer risk prediction significantly. Our study also shows that both mammography features and SNPs contribute to this improvement.

The structure information of the mammography features is derived from the BI-RADS lexicon, which is used widely in breast imaging practice. Thus, our model would likely be generalizable to other practices. On the other hand, we extracted the structure information of SNPs by computing Hamming distances (Wang et al., 2015). This method may not perform as well in small sample sizes, which may affect our results perhaps making our predictive performance results conservative.

Our methods for SNPs may not take advantage of biological knowledge that currently exists. For example, it may be possible to utilize the biological information available in HapMap (which encodes linkage disequilibrium) to more accurately emulate the patterns or dependence structure of SNPs, as in (Liu et al., 2012). Furthermore, we realize that taking into account more complicated structure information such as graph or tree structure (Sun and Wang, 2012) may further improve predictive performance of risk prediction models. We leave these promising directions for future work.

In conclusion, our results demonstrate that including structure information in the computational methods we test improves breast cancer risk prediction. Our models use diverse breast cancer risk factors including demographics, genetics, and imaging and leverage structure found in a standardized lexicon that is universally captured in electronic health records (EHRs) throughout the US. This information will increasingly be combined in complex ways. Merging imaging features, clinical notes and genetic data with models that accurately predict disease risk has the potential to provide powerful knowledge to practicing physicians.

Acknowledgments

The authors acknowledge the support of the Wisconsin Genomics Initiative, NCI grant R01CA127379-01 and its ARRA supplement R01CA127379-03S1, NIGMS grant R01GM097618-01, NLM grant R01LM011028-01, NIEHS grant 5R01ES017400-03, NIH grant 1U54AI117924-01, NIH grant K24CA194251 and NSF FRG grant DMS-1265202. We also acknowledge support from the Clinical and Translational Science Award (CTSA) program through the NIH National Center for Advancing Translational Sciences (NCATS) grant UL1TR000427 and the University of Wisconsin Carbone Cancer Center Cancer Support Grant P30CA014520. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH. We thank the anonymous reviewers for their valuable comments and suggestions.

References

A. Beck and M. Tebouille. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183-202, 2009.

- Breast Imaging Reporting And Data System (BI-RADS). 5th ed. Reston VA: American College of Radiology, 2014.
- E. S. Burnside, J. Liu, Y. Wu, A. A. Onitilo, C. A. McCarty, C. D. Page, P. Peissig, A. Trentham-Dietz, T. Kitchner, J. Fan, and M. Yuan. Comparing Mammography Abnormality Features and Genetic Variants in the Prediction of Breast Cancer in Women Recommended for Breast Biopsy. *Academic Radiology*, 23(1):62-9, 2016.
- J. Ferlay, I. Soerjomataram, M. Ervik, et al. *GLOBOCAN 2012 cancer incidence and mortality worldwide: IARC cancerbase No. 11*. Lyon, France: International Agency for Research on Cancer, 2013.
- D. Freedman, D. Petitti, and J. Robins. On the efficacy of screening for breast cancer. *Int J Epidemiol.*, 33(1):43-55, 2004.
- M. Gail. Discriminatory accuracy from single-nucleotide polymorphisms in models to predict breast cancer risk. *J Natl Cancer Inst.*, 100(14):1037-41, 2008.
- M. Gail. Value of adding single-nucleotide polymorphism genotypes to a breast cancer risk model. *J Natl Cancer Inst.*, 101(13):959-63, 2009.
- T. Lin, S. Ma, and S. Zhang. An extragradient-based alternating direction method for convex minimization. *Found Comput Math*, DOI 10.1007/s10208-015-9282-8, 2015.
- J. Liu, J. Huang, S. Ma, and K. Wang. Incorporating group correlations in genome-wide association studies using smoothed group lasso. *Biostatistics*, 14:205-219, 2013.
- J. Liu, C. D. Page, P. L. Peissig, et al. New genetic variants improve personalized breast cancer diagnosis. *AMIA Summit on Translational Bioinformatics (AMIA-TBI)*, 2014.
- J. Liu, C. Zhang, C. McCarty, P. L. Peissig, E. S. Burnside, and D. Page. Graphical-model based multiple testing under dependence, with applications to genome-wide association studies. In *Proceedings of the 28th conference on uncertainty in artificial intelligence*, 2012.
- S. Ma, X. Song, and J. Huang. Supervised group Lasso with applications to microarray data analysis. *BMC bioinformatics*, 8:60, 2007.
- N. Mavaddat, et al. Prediction of breast cancer risk based on profiling with common genetic variants. *J Natl Cancer Inst.*, 107(5):djv036, 2015.
- C. McCarty, R. Wilke, P. Giampietro, S. Westbrook, and M. Caldwell. Marshfield Clinic Personalized Medicine Research Project (PMRP): design, methods and recruitment for a large population-based biobank. *Personalized Med.*, 2(1):49-79, 2005.
- L. Meier, S. Van De Geer, and P. Böhmann. The Group Lasso for logistic regression. *J. R. Statist. Soc. B*, 70:53-71, 2008.
- K. Michailidou, P. Hall, A. Gonzalez-Neira, et al. Large-scale genotyping identifies 41 new loci associated with breast cancer risk. *Nat Genet.*, 45(4):353-61, 2013.

- H. Nassif, R. Woods, E. S. Burnside, M. Ayvaci, J. Shavlik, C. D. Page. Information extraction for clinical data mining: a mammography case study. *IEEE International Conference on Data Mining Workshops*, 2009.
- N. G. Polson, J. G. Scott, and B. T. Willard. Proximal Algorithms in Statistics and Machine Learning. *Statistical Science*, 30(4):559-581, 2015.
- H. Sun and S. Wang. Penalized logistic regression for high-dimensional DNA methylation data with case-control studies. *Bioinformatics*, 28(10):1368-1375, 2012.
- R. Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Statist. Soc. B*, 58:267-288, 1996.
- R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *J. R. Statist. Soc. B*, 67:91-108, 2005.
- R. Tibshirani and P. Wang. Spatial smoothing and hot spot detection for CGH data using the fused lasso. *Biostatistics*, 9(1):18-29, 2008.
- S. Wacholder, P. Hartge, R. Prentice, et al. Performance of common genetic variants in breast-cancer risk models. *N Engl J Med.*, 362(11):986-93, 2010.
- C. Wang, W. H. Kao, and C. K. Hsiao. Using Hamming distance as information for SNP-sets clustering and testing in disease association studies. *PLoS One*, 10(8), 2015.
- Y. Wu, O. Alagoz, M. Ayvaci, A. M. del Rio, D. J. Vanness, R. Woods, and E. S. Burnside. A comprehensive methodology for determining the most informative mammographic features. *J Digit Imaging*, 26(5):941-947, 2013.
- Y. Wu, J. Liu, C. D. Page, P. L. Peiszig, C. A. McCarty, A. A. Onitilo, and E. S. Burnside. Comparing the Value of Mammographic Features and Genetic Variants in Breast Cancer Risk Prediction. *AMIA Annu Symp Proc.*, 1228-1237, 2014.
- G. Ye and X. Xie. Split Bregman method for large scale fused Lasso. *Computational Statistics and Data Analysis*, 55(4):1552-1569, 2011.
- D. Yu, S. Lee, W. Lee, S. Kim, J. Lim, and S. Kwon. Classification of spectral data using fused lasso logistic regression. *Chemometrics and Intelligent Laboratory Systems*, 142:70-77, 2015.
- M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J. R. Statist. Soc. B*, 68:49-67, 2006.
- J. Zhou, J. Liu, V. A. Narayan, and J. Ye. Modeling disease progression via fused sparse group lasso. In *KDD*, pages 1095-1103, 2012.

An Error Bound for L_1 -norm Support Vector Machine Coefficients in Ultra-high Dimension

Bo Peng

Lan Wang

School of Statistics

University of Minnesota

Minneapolis, MN 55455, USA

PENG0199@UMIN.EDU

WANGX346@UMIN.EDU

Yichao Wu

Department of Statistics

North Carolina State University

Raleigh, NC 27695, USA

WU@STAT.NCSU.EDU

Editor: Jie Peng

Abstract

Comparing with the standard L_2 -norm support vector machine (SVM), the L_1 -norm SVM enjoys the nice property of simultaneously performing classification and feature selection. In this paper, we investigate the statistical performance of L_1 -norm SVM in ultra-high dimension, where the number of features p grows at an exponential rate of the sample size n . Different from existing theory for SVM which has been mainly focused on the generalization error rates and empirical risk, we study the asymptotic behavior of the coefficients of L_1 -norm SVM. Our analysis reveals that the estimated L_1 -norm SVM coefficients achieve near oracle rate, that is, with high probability, the L_2 error bound of the estimated L_1 -norm SVM coefficients is of order $O_p(\sqrt{q \log p/n})$, where q is the number of features with nonzero coefficients. Furthermore, we show that if the L_1 -norm SVM is used as an initial value for a recently proposed algorithm for solving non-convex penalized SVM (Zhang et al., 2016b), then in two iterative steps it is guaranteed to produce an estimator that possesses the oracle property in ultra-high dimension, which in particular implies that with probability approaching one the zero coefficients are estimated as exactly zero. Simulation studies demonstrate the fine performance of L_1 -norm SVM as a sparse classifier and its effectiveness to be utilized to solve non-convex penalized SVM problems in high dimension.

Keywords: feature selection, L_1 -norm SVM; non-convex penalty, oracle property, error bound, support vector machine, ultra-high dimension

1. Introduction

Support vector machine (SVM), originally introduced by Boser et al. (1992) and Vapnik (1995) and subsequently investigated by many others, is a popular and highly powerful technique for classification and has a solid mathematical foundation in statistical learning. In modern applications, we often face the challenge of classification at the presence of a very large number of redundant features. For example, in genomics it is of fundamental importance to build a classifier using a small number of genes from thousands of candidate genes for the purpose of disease diagnosis and drug discovery; in spam email classification,

it is desirable to build an accurate classifier using a relatively small number of words from a dictionary that contains a huge number of different words. For such applications, the standard L_2 -norm SVM suffers from some potential drawbacks. First, L_2 -norm SVM does not automatically build in dimension reduction and hence usually does not yield an interpretable sparse decision rule. Second, the generalization performance of L_2 -norm SVM can deteriorate by including many redundant features (e.g., Zhu et al., 2004).

The standard L_2 -norm SVM has the well known *hinge loss*+ L_2 *norm penalty* formulation. An effective way to perform simultaneous variable selection and classification using SVM is to replace the L_2 -norm penalty with the L_1 -norm penalty, which results in the L_1 -norm SVM. See the earlier work of Bradley and Mangasarian (1998) and Song et al. (2002). Important advancement on the methodology and theory of L_1 -norm SVM has been obtained in recent years, for example, Zhu et al. (2004) proposed a path-following algorithm and effectively demonstrated the advantages of L_1 -norm SVM in high-dimensional sparse scenario; Tarigan and van de Geer (2004) investigated the adaptivity of SVMs with L_1 penalty and derived its adaptive rates; Tarigan, Van De Geer, et al. (2006) obtained an oracle inequality involving both model complexity and margin for L_1 -norm SVM; Wang and Shen (2007) extended L_1 -norm SVM to multi-class classification problems; Zou (2007) proposed to use adaptive L_1 penalty with the SVM; and Wegkamp and Yuan (2011) considered L_1 -norm SVM with a built-in reject option.

The existing theory in the literature on SVM has been largely focused on the analysis of generalization error rate and empirical risk, see Greenshtein et al. (2006), Wang and Shen (2007), Van de Geer (2008), among others. These results neither contain nor directly imply the transparent error bound of the estimated coefficients of L_1 -norm SVM studied in this paper. Our work makes a significant departure from most of the existing literature and is motivated by the recent growing interest of understanding the statistical properties of the estimated SVM coefficients (also referred to as the weight vector). For a linear binary SVM, the decision function is a hyperplane that separates two classes. The coefficients of SVM describe this hyperplane which directly predicts which class a new observation point belongs to. Moreover, the magnitudes of the SVM coefficients provide critical information on the importance of the features and can be used for feature ranking (Chang and Lin, 2008; Guyon et al., 2002). Koo et al. (2008) derived a novel Bahadur type representation of the coefficients of the L_2 -norm SVM and established the asymptotic normality of the estimated coefficients when the number of features p is fixed. Park et al. (2012) studied the oracle properties of SCAD-penalized SVM coefficients, also for the fixed p case. The aforementioned worked has only considered small, fixed number of features. More recently, Zhang et al. (2016b) proposed a systemic framework for non-convex penalized SVM regarding variable selection consistency and oracle property in high dimension. Zhang et al. (2016a) investigated a consistent information criterion for tuning parameter selection for support vector machine in the diverging model space. Both of these two papers directly assume an appropriate initial value exists in the high-dimensional setting.

In this paper, we study the asymptotic behavior of the estimated L_1 -norm SVM coefficients and derive that the error bound is of near-oracle rate $O(\sqrt{q \log p/n})$, where q is the number of features with nonzero coefficients, n is the sample size, and the number of candidate features p can be of exponential order of n (i.e., the ultra-high dimensional case). Furthermore, in Section 4 we show that this sharp error bound helps greatly extend the

applicability of the recent algorithm and theory of high-dimensional non-convex-penalized SVM (Zhang et al., 2016b) by providing a statistically valid and computationally convenient initial value. The use of non-convex penalty function aims to further reduce the bias associated with the L_1 penalty and accurately identify the set of relevant features for classification. However, the presence of non-convex penalty results in computational complexity. Zhang et al. (2016b) proposed an algorithm and showed that given an appropriate initial value, in two iterative steps the algorithm is guaranteed to produce an estimator that possesses the oracle property in the ultra-high dimension and consequently with probability approaching one the zero coefficients are estimated as exactly zero. However, the availability of a qualified initial estimator is itself a challenging issue in high dimension. Zhang et al. (2016b) provided an initial estimator that would satisfy the requirement when $p = o(\sqrt{n})$. Our result shows that the L_1 -norm SVM can be a valid initial estimator under general conditions when p grows at an exponential rate of n , which completes the algorithm and theory of Zhang et al. (2016b).

The rest of the paper is organized as follows. In Section 2, we introduce the basics and computation of the L_1 -norm penalized support vector machine. Section 3 derives the near-oracle error bound for the estimated L_1 -norm SVM coefficients in the ultra-high dimension. Section 4 investigates the application of the result in Section 3 for non-convex penalized SVM in the ultra-high dimension. Section 5 demonstrates through Monte Carlo experiments the effectiveness of L_1 -norm SVM coefficients both as a sparse classifier and as an initial value for the non-convex penalized SVM algorithm. Technical proofs and additional notes are given in the appendices.

2. L_1 -norm support vector machine

We consider the classical binary classification problem. Let $\{Y_i, \mathbf{X}_i\}_{i=1}^n$ be a random sample from an unknown distribution $P(\mathbf{X}, Y)$. The response variable (class label) $Y_i \in \{1, -1\}$ has the marginal distribution: $P(Y_i = 1) = \pi_+$ and $P(Y_i = -1) = \pi_-$, where $\pi_+, \pi_- > 0$ and $\pi_+ + \pi_- = 1$. We write $\mathbf{X}_i = (X_{i0}, X_{i1}, \dots, X_{ip})^T = (X_{i0}, (\mathbf{X}_i^-)^T)^T$, where $X_{i0} = 1$ corresponds to the intercept term. Let f and g be the conditional density functions of \mathbf{X}_+ - given $Y_i = 1$ and \mathbf{X}_- given $Y_i = -1$, respectively. Moreover, in this paper we use the following notation for vector norms: for $\mathbf{x} = (x_1, \dots, x_k)^T \in \mathbb{R}^k$ and a positive integer m , we define $\|\mathbf{x}\|_m = \left(\sum_{i=1}^k |x_i|^m\right)^{1/m}$, $\|\mathbf{x}\|_\infty = \max\{|x_1|, \dots, |x_k|\}$ and $\|\mathbf{x}\|_0 = \sum_{i=1}^k I(x_i \neq 0)$.

The standard linear SVM can be expressed as the following regularization problem

$$\min_{\boldsymbol{\beta}} n^{-1} \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta})_+ + \lambda \|\boldsymbol{\beta}_-\|_2^2, \quad (1)$$

where $(1 - u)_+ = \max\{1 - u, 0\}$ is often called the hinge loss function, λ is a tuning parameter and $\boldsymbol{\beta} = (\beta_0, (\boldsymbol{\beta}_-)^T)^T$ with $\boldsymbol{\beta}_- = (\beta_1, \beta_2, \dots, \beta_p)^T$. Generally for a given vector \mathbf{e} , we use \mathbf{e}_- to denote the subvector with the first entry of \mathbf{e} omitted. Actually, optimization problem in (1) is known as the primal problem of the SVM, which can be efficiently solved by quadratic programming algorithms.

The L_1 -norm SVM replaces the L_2 penalty in (1) by the L_1 penalty. That is, we consider the objective function

$$l_n(\boldsymbol{\beta}, \lambda) = n^{-1} \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta})_+ + \lambda \|\boldsymbol{\beta}_-\|_1, \quad (2)$$

and define

$$\widehat{\boldsymbol{\beta}}(\lambda) = \arg \min_{\boldsymbol{\beta}} l_n(\boldsymbol{\beta}, \lambda). \quad (3)$$

For a given data point X_i , it is classified into class + (corresponding to $\widehat{Y}_i = 1$) if $\mathbf{X}_i^T \widehat{\boldsymbol{\beta}}(\lambda) > 0$ and into class - (corresponding to $\widehat{Y}_i = -1$) if $\mathbf{X}_i^T \widehat{\boldsymbol{\beta}}(\lambda) < 0$.

By introducing the slack variables, we can transform our optimization problem (3) as a linear programming problem (Zhu et al., 2004)

$$\begin{aligned} \min_{\xi, \zeta, \boldsymbol{\beta}} & \left(\frac{1}{n} \sum_{i=1}^n \xi_i + \lambda \sum_{j=1}^p \zeta_j \right) \\ \text{subject to} & \xi_i \geq 0, \quad i = 1, 2, \dots, n, \\ & \xi_i \geq 1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}, \quad i = 1, 2, \dots, n, \\ & \zeta_j \geq \beta_j, \zeta_j \geq -\beta_j, \quad j = 1, 2, \dots, p. \end{aligned} \quad (4)$$

Several R packages are available to solve such a standard linear programming problem, such as `lpSolve` and `linprog`.

3. An error bound of L_1 -norm SVM in ultra-high dimension

In this section, we will describe the near-oracle error bound for the estimated L_1 -norm SVM coefficients under the ultra-high dimensional setting. The choice of the tuning parameter λ will be studied to achieve this error bound.

3.1 Preliminaries

The key result of the paper is an error bound of $\|\widehat{\boldsymbol{\beta}}(\lambda) - \boldsymbol{\beta}^*\|_2$, where $\boldsymbol{\beta}^*$ is the minimizer of the population version of the hinge loss function, that is,

$$\boldsymbol{\beta}^* = \arg \min_{\boldsymbol{\beta}} L(\boldsymbol{\beta}), \quad (5)$$

where $L(\boldsymbol{\beta}) = E(1 - Y \mathbf{X}^T \boldsymbol{\beta})_+$. Lin (2002) suggested that there is a close connection between the minimizer of the population hinge loss function and the Bayes rule. The definition of $\boldsymbol{\beta}^*$ above is also used in Koo et al. (2008) and Park et al. (2012), both of which only considered the fixed p case. We are interested in the error bound of $\|\widehat{\boldsymbol{\beta}}(\lambda) - \boldsymbol{\beta}^*\|_2$ when $p \gg n$. In the ultra-high dimensional settings, it is often reasonable to assume that $\boldsymbol{\beta}^* = (\beta_0^*, \beta_1^*, \dots, \beta_p^*)^T$ is sparse in the sense that most of its components are exactly zero. We define the index set of active features as $T = \{1 \leq j \leq p : \beta_j^* \neq 0\}$. We denote the cardinality of T by $|T| = q$. To incorporate the intercept term, we also define $T_+ = T \cup \{0\}$.

Next, we introduce the gradient vector and Hessian matrix of the population hinge loss function $L(\beta)$. We define

$$S(\beta) = -E(I(1 - Y\mathbf{X}^T\beta \geq 0)Y\mathbf{X}) \quad (6)$$

as the $(p+1)$ -dimensional gradient vector and

$$H(\beta) = E(\delta(1 - Y\mathbf{X}^T\beta)\mathbf{X}\mathbf{X}^T) \quad (7)$$

as the $(p+1) \times (p+1)$ -dimensional Hessian matrix where $I(\cdot)$ is the indicator function and $\delta(\cdot)$ is the Dirac delta function. Section 6.1 in Koo et al. (2008) has explained more details and theoretical properties of $S(\beta)$ and $H(\beta)$ under certain conditions.

Throughout the paper, we assume the following regularity condition.

(A1) The densities f and g are continuous with common support $\mathcal{S} \subset \mathbb{R}^p$ and have finite second moments. In addition, there exists a constant $M > 0$ such that $|X_j| \leq M$, $j \in \{1, \dots, p\}$.

REMARK 1. Condition (A1) ensures that $H(\beta)$ is well defined and continuous in β . The bound of \mathbf{X}_- can be relaxed with further technical complexity. More details can be found in Park et al. (2012) and Koo et al. (2008).

3.2 The choice of the tuning parameter λ and a fact about $\hat{\beta}$

The estimated L_1 -norm SVM parameter $\hat{\beta}(\lambda)$ defined in (3) depends on the tuning parameter λ . We will first show that a universal choice

$$\lambda = c\sqrt{2A(\alpha)\log p/n}, \quad (8)$$

where c is some given constant, α is a small probability and $A(\alpha) > 0$ is a constant such that $4p^{-\frac{A(\alpha)}{M^2}+1} \leq \alpha$, can provide theoretical guarantee on the good performance of $\hat{\beta}(\lambda)$.

The above choice of λ is motivated by a principle in the setting of penalized least squares regression (Bickel et al., 2009), which advocates to choose the penalty level λ to dominate the subgradient of the loss function evaluated at the true value. Intuitively, the subgradient evaluated at β^* summarizes the estimation noise. See also the application of the same principle to choose the penalty level for quantile regression (Belloni and Chernozhukov, 2011; Wang, 2013). Another more technical motivation of this principle comes from the KKT condition in convex optimization theory. Let $\hat{\beta}$ be the oracle estimator (formally defined in Section 4) that minimizes the sample hinge loss function when the index set T is known in advance. Define the subgradient function

$$\hat{S}(\beta) = -n^{-1} \sum_{i=1}^n I(1 - Y_i \mathbf{X}_i^T \beta \geq 0) Y_i \mathbf{X}_i.$$

Then it follows from the argument as in Theorem 3.1 of Zhang et al. (2016b) that under some weak regularity conditions $\|\hat{S}(\beta)\|_\infty \leq \lambda$ with probability approaching one. It follows from Koo et al. (2008) that the oracle estimator $\hat{\beta}$ provides a consistent and asymptotically normal estimate of β^* .

Hence, in the ideal case where the population parameter β^* is known, an intuitive choice of λ is to set its value to be larger than the supremum norm of $S(\beta^*)$ with large probability, that is

$$P(\lambda \geq c\|\hat{S}(\beta^*)\|_\infty) \geq 1 - \alpha, \quad (9)$$

where $c > 1$ is some given constant and α is a small probability. Lemma 1 below shows that the choice of λ given in (8) satisfies this requirement.

Lemma 1 Assume that condition (A1) is satisfied. Suppose $\lambda = c\sqrt{2A(\alpha)\log p/n}$, we have

$$P(\lambda \geq c\|\hat{S}(\beta^*)\|_\infty) \geq 1 - \alpha$$

with α being a given small probability defined earlier in this section.

The proof of Lemma 1 is given in the Appendix A. The crux of the proof is to bound the tail probability of $\sum_{i=1}^n I(1 - Y_i \mathbf{X}_i^T \beta^* \geq 0) Y_i \mathbf{X}_i$ by applying Hoeffding's inequality and the union bound. Later in this section, we will show that this choice of λ warrants near-oracle rate performance of $\hat{\beta}(\lambda)$. Let $\mathbf{h} = \beta^* - \hat{\beta}(\lambda)$. We state below an interesting fact about \mathbf{h} .

Lemma 2 For $\lambda \geq c\|\hat{S}(\beta^*)\|_\infty$ and $\bar{C} = \frac{c-1}{c+1}$, we have

$$\mathbf{h} \in \Delta_{\bar{C}},$$

where

$$\Delta_{\bar{C}} = \{\gamma \in \mathbf{R}^{p+1} : \|\gamma_{T_+}\|_1 \geq \bar{C}\|\gamma_{T_+^c}\|_1, \text{ where } T_+ = T \cup \{0\}, T \subset \{1, 2, \dots, p\} \text{ and } |T| \leq q\},$$

with T_+^c denoting the complement of T_+ , and γ_{T_+} denoting the $(p+1)$ -dimensional vector that has the same coordinates as γ on T_+ and zero coordinates on T_+^c .

We call $\Delta_{\bar{C}}$ the restricted set. The proof of Lemma 2 is also given in Appendix A.

3.3 Regularity conditions

Let $\mathcal{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n)^T$ denote the feature design matrix. We define restricted eigenvalues as follows

$$\lambda_{\max} = \max_{\mathbf{d} \in \mathbb{R}^{p+1}; \|\mathbf{d}\|_0 \leq 2(q+1)} \frac{\mathbf{d}^T \mathcal{X}^T \lambda \mathbf{d}}{\eta \|\mathbf{d}\|_2^2} \quad (10)$$

and

$$\lambda_{\min}(H(\beta^*); q) = \min_{\mathbf{d} \in \Delta_{\bar{C}}} \frac{\mathbf{d}^T H(\beta^*) \mathbf{d}}{\|\mathbf{d}\|_2^2}. \quad (11)$$

These are similar to the sparse eigenvalue notion in Bickel, Ritov, and Tsybakov (2009) and Meinshausen and Yu (2009) for analyzing sparse least squares regression, see also Cai, Wang, and Xu (2010).

In addition to condition (A1) introduced in Section 2, we require the following regularity conditions for the main theory of this paper.

(A2) $q = O(n^{c_1})$ for some $0 \leq c_1 < 1/2$.

(A3) There exists a constant M_1 such that $\lambda_{\max} \leq M_1$ almost surely.

(A4) $\lambda_{\min}(H(\boldsymbol{\beta}^*); q) \geq M_2$, for some constant $M_2 > 0$.

(A5) $n^{(1-\alpha_2)/2} \min_{j \in \mathcal{I}} |\beta_j^*| \geq M_3$ for some constants $M_3 > 0$ and $2c_1 < \alpha_2 \leq 1$.

(A6) Denote the conditional density of $\mathbf{X}^T \boldsymbol{\beta}^*$ given $Y = +1$ and $Y = -1$ as f^* and g^* , respectively. It is assumed that f^* is uniformly bounded away from 0 and ∞ in a neighborhood of 1 and g^* is uniformly bounded away from 0 and ∞ in a neighborhood of -1 .

REMARK 2. Conditions (A2) and (A5) are very common in high dimensional literature. Basically, condition (A2) states that the number of nonzero variables cannot diverge at a rate larger than \sqrt{n} . Condition (A5) controls the decay rate of true parameter $\boldsymbol{\beta}^*$. Condition (A3) is not restrictive, see the relevant discussions in Meinshausen and Yu (2009). Condition (A4) requires the smallest restricted eigenvalue has a lower bound. This would be satisfied if $H(\boldsymbol{\beta}^*)$ is positive definite. We provide a thorough discussion of this condition in Appendix B, including an example that demonstrates the validity of this condition. Condition (A6) warrants that there is sufficient information around the non-differentiable point of the hinge loss, similarly to Condition (C3) in Wang, Wu, and Li (2012) for quantile regression.

3.4 An error bound of $\widehat{\beta}(\lambda)$ in ultra-high dimension

Before stating the main theorem, we first present an important lemma, which has to do with the empirical process behavior of the hinge loss function.

Lemma 3 Assume that conditions (A1)-(A3) are satisfied. For $\mathbf{h} \in \mathbb{R}^{p+1}$, let

$$\begin{aligned} B(\mathbf{h}) &= \frac{1}{n} \left| \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^* + Y_i \mathbf{X}_i^T \mathbf{h})_+ - \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^*)_+ \right. \\ &\quad \left. - E \left(\sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^* + Y_i \mathbf{X}_i^T \mathbf{h})_+ - \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^*)_+ \right) \right|. \end{aligned}$$

Assume $p > n$, then for all n sufficiently large

$$P \left(\sup_{\|\mathbf{h}\|_0 \leq q+1, \|\mathbf{h}\|_2 \neq 0} \frac{B(\mathbf{h})}{\|\mathbf{h}\|_2} \geq (1 + 2C_1 \sqrt{M_1}) \sqrt{\frac{2q \log p}{n}} \right) \leq 2p^{-2q(C_1^2-1)},$$

where $C_1 > 1$ is a constant.

Lemma 3 guarantees that $n^{-1} (\sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^* + Y_i \mathbf{X}_i^T \mathbf{h})_+ - \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^*)_+)$ is close to its expected value with high probability. This provides an important tool to handle the non-smoothness of the hinge loss function in proving the main theory, which is stated below.

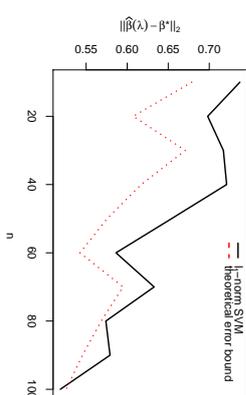


Figure 1: L_2 -norm estimation error comparison

Theorem 4 Suppose that conditions (A1)-(A6) hold, then the estimated L_1 -norm SVM coefficients vector $\widehat{\beta}(\lambda)$ satisfies

$$\|\widehat{\beta}(\lambda) - \boldsymbol{\beta}^*\|_2 \leq \sqrt{1 + \frac{1}{C}} \left(\frac{2\lambda \sqrt{q+1}}{M_2} + \frac{2C}{M_2} \sqrt{\frac{2q \log p}{n}} \left(\frac{5}{4} + \frac{1}{C} \right) \right)$$

with probability at least $1 - 2p^{-2q(C_1^2-1)}$, where C is a constant, C_1 is given in Lemma 3 and \widehat{C} is defined in Lemma 2.

From this theorem, we can easily capture the near-oracle property for l_1 penalized SVM estimator, such that with high probability,

$$\|\widehat{\beta}(\lambda) - \boldsymbol{\beta}^*\|_2 = O_p \left(\sqrt{\frac{q \log p}{n}} \right)$$

when $\lambda = c\sqrt{2A(\alpha) \log p/n}$. Actually, in the inequality of Theorem 4, the first term satisfies $\frac{\lambda \sqrt{q}}{M_2} = \frac{2}{M_2} \sqrt{\frac{2A(\alpha) q \log p}{n}}$ and it is also trivial to have the second term of the same order. Hence the near-oracle property of $\widehat{\beta}(\lambda)$ will hold given λ above.

To numerically evaluate the above error bound of the L_1 -norm SVM, we consider the simulation setting in Model 4 of Section 5.1. We choose $p = 0.1 * n^2$, $q = \lfloor n^{1/3} \rfloor$ and $\boldsymbol{\beta}^* = ((1.1, \dots, 1.1)_q, 0, \dots, 0)^T$, which allows p and q to vary with sample size n . Figure 1 depicts the average of $\|\widehat{\beta}(\lambda) - \boldsymbol{\beta}^*\|_2$ across 200 simulation runs for different values of n for L_1 -norm SVM and compares the curve with the theoretical error bound $(\sqrt{\frac{q \log p}{n}})$. We observe that these two curves display similar decreasing pattern and approach each other as n gets larger.

4. Application to non-convex penalized SVM in ultra-high dimension

In this section, we will step further to discuss the advantage of non-convex penalized SVM in ultra-high dimension. Similarly, the oracle property of non-convex penalized SVM coefficients will be investigated.

4.1 Why non-convex penalty?

Recently, several authors studied non-convex penalized SVM for simultaneous variable selection and classification, see Zhang et al. (2006), Becker et al. (2011), Park et al. (2012) and Zhang et al. (2016b). The idea is to replace the L_2 norm in standard SVM (1) by a non-convex penalty term in the form $\sum_{j=1}^p p_\lambda(|\beta_j|)$, where $p_\lambda(\cdot)$ is a symmetric penalty function with tuning parameter λ . Two commonly used non-convex penalty functions are the SCAD penalty and the MCP penalty. The SCAD penalty (Fan and Li, 2001) is defined by

$$p_\lambda(|\beta|) = \lambda|\beta|I(0 \leq |\beta| < \lambda) + \frac{a\lambda|\beta| - (\beta^2 + \lambda^2)/2}{a-1}I(\lambda \leq |\beta| \leq a\lambda) + \frac{(a+1)\lambda^2}{2}I(|\beta| > a\lambda)$$

for some $a > 2$. The MCP (Zhang, 2010) is defined by

$$p_\lambda(|\beta|) = \lambda(|\beta| - \frac{\beta^2}{2a\lambda})I(0 \leq |\beta| < a\lambda) + \frac{a\lambda^2}{2}I(|\beta| \geq a\lambda)$$

for some $a > 1$.

The motivation of using non-convex penalty function is to further reduce the bias resulted from L_1 penalty and accurately identify the set of relevant features \mathcal{T} . The use of non-convex penalty function was introduced in the setting of penalized least squares regression (Fan and Li, 2001; Zhang, 2010). These authors observed that L_1 penalized least squares regression requires stringent conditions, often not satisfied in real data analysis, to achieve variable selection consistency. The use of non-convex penalty function alleviates the bias caused by L_1 penalty which overpenalizes large coefficients, and leads to the so called *oracle property*. That is, under regularity conditions the resulted non-convex penalized estimator is able to estimate zero coefficients as exactly zero with probability approaching one, and estimate the nonzero coefficients as efficiently as if the set of relevant features is known in advance.

4.2 Oracle property in ultra-high dimension

The oracle property of non-convex penalized SVM coefficients is investigated by Park et al. (2012) for the case of fixed number of features and more recently by Zhang et al. (2016b) for the large p case. The oracle estimator of β^* is defined as

$$\tilde{\beta} = \arg \min_{\beta, \beta_{\mathcal{T}_+} = 0} \tilde{l}_n(\beta), \quad (12)$$

where $\tilde{l}_n(\beta) = n^{-1} \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \beta)_+$ is the sample hinge loss function and $\beta_{\mathcal{T}_+}$ denotes the vector containing the components of β with indices in \mathcal{T}_+ and others to be zero.

To solve the non-convex penalized SVM, we choose to use the local linear approximation (LLA) algorithm. The LLA algorithm starts with an initial value $\beta^{(0)}$. At each step t , we update the β to be $\beta^{(t)}$ by solving

$$\min_{\beta} \left\{ n^{-1} \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \beta)_+ + \sum_{j=1}^p p_\lambda^{(\epsilon-1)}(|\beta_j|) \right\}, \quad (13)$$

where $p_\lambda^{(\cdot)}$ denotes the derivative of the penalty function $p_\lambda(\cdot)$. Specifically, we have $p_\lambda^{(0)} = p_\lambda^{(0+)} = \lambda$.

Zhang et al. (2016b) showed that if an appropriate initial estimator exists, then under quite general regularity conditions, the LLA algorithm can identify the oracle estimator with probability approaching one in just two iterative steps (see their Theorem 3.4). This result provides a systematic framework for non-convex penalized SVM in high dimension. However it relies on the availability of a qualified initial value $\tilde{\beta}^{(0)} = (\tilde{\beta}_0^{(0)}, \tilde{\beta}_1^{(0)}, \dots, \tilde{\beta}_p^{(0)})^T$ that satisfies

$$P(|\tilde{\beta}_j^{(0)} - \beta_j^*| > \lambda, \text{ for some } 1 \leq j \leq p) \rightarrow 0 \text{ as } n \rightarrow \infty. \quad (14)$$

Yet the availability of such an appropriate initial value is itself a challenging problem in ultra-high dimension. Zhang et al. (2016b) showed that such an initial estimator is guaranteed when $p = o(\sqrt{n})$. The error bound we derived on L_1 -norm SVM ensures that a qualified initial value is indeed available under general conditions in ultra-high dimension and hence greatly extends the applicability of the result of Zhang et al. (2016b). In the following we restate Theorem 3.4 of Zhang et al. (2016b) for the ultra-high dimensional case.

Theorem 5 Assume $\tilde{\beta}(\lambda)$ is the solution to the L_1 -norm SVM with tuning parameter $\lambda = c\sqrt{2A(\alpha)} \log p/n$ defined above. Suppose that conditions (A1)-(A6) hold, then we have $P(|\tilde{\beta}_j(\lambda) - \beta_j^*| > \lambda, \text{ for some } 1 \leq j \leq p) \rightarrow 0$ as $n \rightarrow \infty$. Furthermore, the LLA algorithm initiated by $\tilde{\beta}(\lambda)$ finds the oracle estimator in two iterations with probability tending to 1, i.e., $P(\tilde{\beta}^{(nc)}(\lambda) = \tilde{\beta})$, where $\tilde{\beta}^{(nc)}(\lambda)$ is the solution for non-convex penalized SVM with given λ .

5. Simulation experiments

In this section, we will investigate the finite sample performance of the L_1 -norm SVM. We will also study its application to non-convex penalized SVM in high dimension.

5.1 Monte Carlo results for L_1 -norm SVM

We generate random data from each of the following four models.

- Model 1: $Pr(Y = 1) = Pr(Y = -1) = 0.5$, $\mathbf{X}_- | (Y = 1) \sim MN(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, $\mathbf{X}_- | (Y = -1) \sim MN(-\boldsymbol{\mu}, \boldsymbol{\Sigma})$, $q = 5$, $\boldsymbol{\mu} = (0.1, 0.2, 0.3, 0.4, 0.5, 0, \dots, 0)^T \in \mathbb{R}^p$, $\boldsymbol{\Sigma} = (\sigma_{ij})$ with diagonal entries equal to 1, nonzero entries $\sigma_{ij} = -0.2$ for $1 \leq i \neq j \leq q$ and other entries equal to 0. The Bayes rule is $\text{sign}(1.39X_1 + 1.47X_2 + 1.56X_3 + 1.65X_4 + 1.74X_5)$ with Bayes error 6.3%.

- **Model 2:** $Pr(Y = 1) = Pr(Y = -1) = 0.5$, $\mathbf{X}_-(Y = 1) \sim MN(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, $\mathbf{X}_-(Y = -1) \sim MN(-\boldsymbol{\mu}, \boldsymbol{\Sigma})$, $q = 5$, $\boldsymbol{\mu} = (0.1, 0.2, 0.3, 0.4, 0.5, 0, \dots, 0)^T \in \mathbb{R}^p$, $\boldsymbol{\Sigma} = (\sigma_{ij})$ with $\sigma_{ij} = -0.4^{|i-j|}$ for $1 \leq i, j \leq q$ and other entries equal to 0. The Bayes rule is $\text{sign}(3.09X_1 + 4.45X_2 + 5.06X_3 + 4.77X_4 + 3.58X_5)$ with Bayes error 0.6%.
- **Model 3:** model stays the same as Model 2, but $\boldsymbol{\Sigma} = (\sigma_{ij})$ with nonzero elements $\sigma_{ij} = -0.4^{|i-j|}$ for $1 \leq i, j \leq q$ and $\sigma_{ij} = 0.4^{|i-j|}$ for $q < i, j \leq p$. The Bayes rule is still $\text{sign}(3.09X_1 + 4.45X_2 + 5.06X_3 + 4.77X_4 + 3.58X_5)$ with Bayes error 0.6%.
- **Model 4:** $\mathbf{X}_- \sim MN(\mathbf{0}_p, \boldsymbol{\Sigma})$, $\boldsymbol{\Sigma} = (\sigma_{ij})$ with nonzero elements $\sigma_{ij} = 0.4^{|i-j|}$ for $1 \leq i, j \leq p$, $Pr(Y = 1|\mathbf{X}_-) = \Phi(\mathbf{X}_-^T \boldsymbol{\beta}^*)$, where $\Phi(\cdot)$ is the cumulative density function of the standard normal distribution, $\boldsymbol{\beta}_*^* = (1.1, 1.1, 1.1, 1.1, 0, \dots, 0)^T$ and $q = 4$. The Bayes rule is $\text{sign}(1.1X_1 + 1.1X_2 + 1.1X_3 + 1.1X_4)$ with Bayes error 10.4%.

Model 1 and Model 4 are identical to the ones in Zhang et al. (2016b). In particular, Model 1 focuses on a standard linear discriminate analysis setting. On the other hand, Model 4 is a typical probit regression case. Models 2 and 3 are designed with autoregressive covariance as correlation decaying off-diagonal-wise. We consider sample size $n = 100$ with $p = 1000$ and 1500, and $n = 200$ with $p = 1500$ and 2000. Similarly as in Cai, Liu, and Luo (2011), we use an independent tuning data set of size $2n$ to tune our λ by minimizing the prediction error using five-fold cross validation. The tuning range spans from 2^{-6} to 2 as equally-spaced sequence with 100 elements. For each simulation scenario, we conduct 200 runs. Then we generate an independent test data set of size n to report the estimated test error.

We evaluate the performance of L_1 -norm SVM by its testing misclassification error rate, estimator error and variable selection ability. In particular, we measure the estimation accuracy by two criteria: the L_2 estimation error $\|\hat{\boldsymbol{\beta}}(\lambda) - \boldsymbol{\beta}^*\|_2$ where Appendix B provides details on the calculation of $\boldsymbol{\beta}^*$ and the absolute value of the sample correlation between $\mathbf{X}^T \hat{\boldsymbol{\beta}}(\lambda)$ and $\mathbf{X}^T \boldsymbol{\beta}^*$. The absolute value of the sample correlation (AAC) is also used as accuracy measure in Cook et al. (2007). To summarize, we will report

- **Test error:** the misclassification error rate.
- L_2 **error:** $\|\hat{\boldsymbol{\beta}}(\lambda) - \boldsymbol{\beta}^*\|_2$.
- **AAC:** Absolute absolute correlation $\text{corr}(\mathbf{X}^T \hat{\boldsymbol{\beta}}(\lambda), \mathbf{X}^T \boldsymbol{\beta}^*)$.
- **Signal:** the average of number of nonzero regression coefficients $\hat{\beta}_i \neq 0$ with $i = 1, 2, 3, 4, 5$ for Model 1-3 and with $i = 1, 2, 3, 4$ for Model 4. This measures the ability of L_1 -norm SVM selecting relevant features.
- **Noise:** the average of number of nonzero regression coefficients $\hat{\beta}_i(\lambda) \neq 0$ with $i \notin \{1, 2, 3, 4, 5\}$ for Model 1-3 and with $i \notin \{1, 2, 3, 4\}$ for Model 4. This measures the ability of L_1 -norm SVM not selecting noise features.

Table 1 summarizes the simulation results for all four models. The numbers in the parentheses are the corresponding standard errors based on 200 replications. Overall, the L_1 -norm SVM performs satisfactorily for classification with relatively low error rates in all

Table 1: Simulation results for L_1 -norm SVMs

Model	n	p	Test error	L_2 error	AAC	Signal	Noise
Model 1	100	1000	0.17(0.06)	0.53(0.14)	0.89(0.03)	4.84(0.41)	38.20(5.50)
	100	1500	0.19(0.05)	0.59(0.14)	0.89(0.03)	4.75(0.47)	40.27(5.41)
	200	1500	0.10(0.003)	0.27(0.07)	0.96(0.02)	5.00(0.07)	19.80(4.12)
Model 2	100	2000	0.10(0.02)	0.27(0.06)	0.96(0.02)	5.00(0.00)	23.61(4.80)
	100	1000	0.06(0.04)	0.34(0.12)	0.95(0.02)	4.88(0.35)	21.25(4.22)
	100	1500	0.07(0.04)	0.39(0.12)	0.95(0.02)	4.79(0.41)	28.80(4.61)
Model 3	100	1500	0.02(0.01)	0.21(0.07)	0.97(0.01)	4.99(0.10)	5.41(2.25)
	200	2000	0.02(0.02)	0.22(0.07)	0.97(0.01)	4.99(0.10)	6.88(2.50)
	100	1000	0.06(0.05)	0.36(0.14)	0.95(0.02)	4.8(0.40)	19.93(3.87)
Model 4	100	1500	0.06(0.04)	0.37(0.13)	0.95(0.02)	4.83(0.40)	27.55(4.85)
	200	1500	0.02(0.02)	0.22(0.07)	0.97(0.02)	5.00(0.07)	5.18(2.19)
	200	2000	0.02(0.02)	0.20(0.08)	0.97(0.02)	5.00(0.07)	6.72(2.67)
Model 4	100	1000	0.16(0.04)	0.52(0.13)	0.94(0.03)	3.88(0.33)	12.87(3.65)
	100	1500	0.17(0.05)	0.55(0.14)	0.93(0.03)	3.81(0.42)	12.09(3.56)
	200	1500	0.13(0.03)	0.33(0.09)	0.97(0.01)	4.00(0.00)	11.12(3.53)
200	2000	0.15(0.03)	0.43(0.07)	0.94(0.02)	4.00(0.00)	48.34(7.71)	

the models. Actually, the error rates are all quite close to the Bayes errors. It is also successful in eliminating most of the irrelevant features. The performance improves with increased sample size. In terms of estimation accuracy, the L_2 error decreases as p decreases and n increases, which echoes the result in main theorem. We observe that AAC is greater than 0.9 in most cases, implying that the direction of $\hat{\boldsymbol{\beta}}(\lambda)$ matches that of the Bayes rule.

It is worth noting that the earlier literature have already performed thorough numerical analysis to compare the performance of L_1 -norm SVM with L_2 -norm SVM and logistic regression. For example, Zhu et al. (2004) observes that the performance of L_1 -norm SVM and L_2 -norm SVM is similar when there is no redundant features; however, the performance of L_2 -norm SVM can be adversely affected by the presence of redundant features. Rocha et al. (2009) numerically compared L_1 -norm SVM with logistic regression classifier and discovered that they are comparable but their relative finite-sample advantage depends on the sample size and design. See similar observation in Zou (2007), Zhang et al. (2016b), among others. Although L_1 -norm SVM can outperform regular L_2 -norm SVM when there are many redundant features, it shares the drawback of L_1 penalized least squares regression that it overpenalizes large coefficients and tends to have larger false positives (including more noise features) comparing with the non-convex penalized SVM, which will be investigated in Section 5.2.

5.2 Monte Carlo results for non-convex penalized SVM

In this subsection, we consider the same four models as in Section 5.1. Instead of the L_1 -norm SVM, we use it as the initial value for the non-convex penalized SVM algorithm proposed in Zhang et al. (2016b). We consider two popular choices of non-convex penalty

functions: SCAD penalty (with $a = 3.7$) and MCP penalty (with $a = 3$). As suggested in Zhang et al. (2016b), we used the recently developed high-dimensional BIC criterion to choose the tuning parameter for non-convex penalized SVMs. More specifically, the SVM-extended BIC is defined as

$$SVMIC_\gamma(T) = \sum_{i=1}^n 2\xi_i + \log(n)/|T| + 2\gamma \binom{p}{|T|}, \quad 0 \leq \gamma \leq 1,$$

where in practice we can set $\gamma = 0.5$ as suggested by Chen and Chen (2008) and choose the λ that minimizes the above $SVMIC_\gamma$ for non-convex penalized SVM.

Table 2: Simulation results for SCAD penalized SVM

Model	n	p	Test error	L_2 error	AAC	Signal	Noise
Model 1	100	1000	0.10(0.05)	0.25(0.17)	0.95(0.04)	4.88(0.38)	4.92(5.82)
	100	1500	0.12(0.06)	0.35(0.20)	0.93(0.05)	4.84(0.53)	9.31(8.89)
	200	1500	0.08(0.03)	0.15(0.10)	0.98(0.03)	4.99(0.12)	0.48(0.51)
	200	2000	0.07(0.02)	0.10(0.05)	0.99(0.01)	5.00(0.00)	0.66(0.80)
Model 2	100	1000	0.04(0.05)	0.25(0.17)	0.95(0.05)	4.73(0.51)	1.47(1.38)
	100	1500	0.05(0.05)	0.28(0.18)	0.94(0.05)	4.64(0.55)	1.42(1.38)
	200	1500	0.03(0.03)	0.19(0.10)	0.96(0.03)	4.91(0.29)	2.77(3.53)
	200	2000	0.02(0.01)	0.15(0.06)	0.98(0.02)	5.00(0.07)	1.40(1.81)
Model 3	100	1000	0.05(0.04)	0.30(0.16)	0.94(0.04)	4.53(0.58)	0.58(0.84)
	100	1500	0.04(0.04)	0.24(0.15)	0.95(0.04)	4.75(0.46)	1.08(1.15)
	200	1500	0.02(0.01)	0.14(0.06)	0.98(0.01)	4.99(0.10)	1.30(1.53)
	200	2000	0.02(0.01)	0.15(0.06)	0.98(0.02)	5.00(0.00)	1.32(1.83)
Model 4	100	1000	0.15(0.05)	0.51(0.20)	0.94(0.04)	3.50(0.59)	7.54(5.20)
	100	1500	0.17(0.05)	0.61(0.18)	0.93(0.04)	3.57(0.71)	8.86(6.37)
	200	1500	0.12(0.03)	0.19(0.10)	0.99(0.01)	3.98(0.14)	3.19(2.45)
	200	2000	0.14(0.03)	0.39(0.19)	0.97(0.03)	3.69(0.51)	0.95(1.07)

Tables 2 and 3 summarize the simulation results for SCAD and MCP penalty functions, respectively. We observe that the SCAD-penalized SVM and MCP-penalized MCP have similar performance, both demonstrating a clear advantage of selecting the relevant features and excluding irrelevant ones over L_1 -norm SVM. The Noise size decreases dramatically to less than 3 as the sample size gets larger. The Signal size is almost 5 when $n = 200$ for Model 1-3 and 4 for Model 4, implying the success of selecting the exact true model. We also observe that non-convex penalized SVM has uniformly smaller L_2 error and larger AAC than L_1 -norm SVM. This resonates with the observation in the literature that eliminating irrelevant features enhances classification performance. The Monte Carlo study confirms the effectiveness of the algorithm of Zhang et al. (2016b) for feature selection for SVM in high dimension when using L_1 -norm SVM as an initial value.

Table 3: Simulation results for MCP penalized SVM

Model	n	p	Test error	L_2 error	AAC	Signal	Noise
Model 1	100	1000	0.11(0.05)	0.28(0.17)	0.95(0.04)	4.87(0.42)	5.46(5.45)
	100	1500	0.13(0.07)	0.36(0.20)	0.93(0.05)	4.84(0.47)	9.00(8.49)
	200	1500	0.07(0.02)	0.11(0.07)	0.99(0.02)	4.99(0.10)	0.48(0.51)
	200	2000	0.07(0.02)	0.10(0.04)	0.99(0.01)	5.00(0.00)	0.83(0.83)
Model 2	100	1000	0.03(0.03)	0.20(0.12)	0.96(0.03)	4.84(0.38)	0.88(0.97)
	100	1500	0.11(0.10)	0.47(0.27)	0.89(0.08)	4.08(0.85)	3.56(2.65)
	200	1500	0.02(0.01)	0.14(0.05)	0.98(0.01)	5.00(0.00)	1.50(2.22)
	200	2000	0.02(0.01)	0.14(0.06)	0.98(0.02)	5.00(0.07)	1.38(1.80)
Model 3	100	1000	0.04(0.04)	0.26(0.15)	0.95(0.04)	4.67(0.54)	0.60(0.82)
	100	1500	0.04(0.04)	0.24(0.15)	0.95(0.04)	4.75(0.46)	1.01(1.07)
	200	1500	0.02(0.01)	0.14(0.06)	0.98(0.01)	5.00(0.07)	1.27(1.72)
	200	2000	0.02(0.01)	0.15(0.06)	0.98(0.02)	5.00(0.00)	1.47(2.04)
Model 4	100	1000	0.15(0.05)	0.50(0.20)	0.94(0.04)	3.66(0.52)	7.20(4.49)
	100	1500	0.17(0.05)	0.62(0.16)	0.92(0.04)	3.35(0.68)	4.96(3.58)
	200	1500	0.12(0.03)	0.20(0.12)	0.99(0.01)	3.98(0.12)	1.99(1.72)
	200	2000	0.13(0.03)	0.34(0.17)	0.97(0.02)	3.83(0.43)	0.86(0.80)

6. Conclusion and discussion

We investigate the statistical properties of L_1 -norm SVM coefficients in ultra-high dimension. We proved that L_1 -norm SVM coefficients achieve a near-oracle rate of estimation error. To deal with the non-smoothness of the hinge loss function, we employ empirical processes techniques to derive the theory. Furthermore, we showed that under some general regularity conditions, the L_1 -norm SVM provides an appropriate initial value for the recent algorithm developed by Zhang et al. (2016b) for non-convex penalized SVM in high dimension. Combined with the theory in that paper, we extended the applicability and validity of their result to the ultra-high dimension.

Our work is motivated by the importance of identifying individual features for SVM in analyzing high-dimensional data, which frequently arise in genomics and many other fields. We not only closed a theoretical gap on the estimation error bound on L_1 -SVM when $p \gg n$, but also verified that (Section 4) this leads to consistently identifying important features when combined with a two-step iterative algorithm in the ultra-high dimensional setting. Hence, we have guarantee for both algorithm convergence and theoretical performance. We believe such results are of direct interest to JMLR readers given the popularity of SVM in practice. Our work has substantial difference from the existing work in the literature. The existing theory on SVM has been largely focused on the analysis of generalization error rate and empirical risk. These results neither contain nor directly imply the transparent error bound of the estimated coefficients of L_1 -norm SVM studied in this paper. Furthermore, the techniques used in the paper for deriving the L_2 error bound when $p \gg n$ are completely different from those used in $p < n$ setting. Although our approach for deriving the L_2 -error bound is inspired by the recent work in the literature for Lasso. There is substantial

new technical challenge to deal with the nonsmooth Hinge loss function and requires more delicate application of empirical process techniques. Also, unlike Lasso, we do not require Gaussian or sub-Gaussian conditions in the technical derivation.

Acknowledgments

We would like to acknowledge support for this project from the National Science Foundation (NSF grant DMS-1308960) for Peng and Wang's research and the National Science Foundation (NSF grant DMS-1055210) and National Institutes of Health (NIH grants R01 CA149569 and P01 CA142538) for Wiri's research respectively.

Appendix A: Technical Proofs

Proof of Lemma 1. By the union bound, we have

$$\begin{aligned} P(c\sqrt{2A(\alpha)}\log p/n \leq c\|\widehat{S}(\boldsymbol{\beta}^*)\|_\infty) \\ \leq \sum_{j=0}^p P\left(\sqrt{2A(\alpha)}\log p/n \leq n^{-1}\sum_{i=1}^n I(1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^* \geq 0) Y_i X_{ij}\right). \end{aligned}$$

Notice that we have $S(\boldsymbol{\beta}^*) = 0$ because of minimizer $\boldsymbol{\beta}^*$ and the definition of gradient vector. Then, for each i and j , $E(Y_i X_{ij} I(1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^* \geq 0)) = 0$, by Hoeffding's inequality,

$$\begin{aligned} P\left(\sqrt{2A(\alpha)}\log p/n \leq n^{-1}\sum_{i=1}^n I(1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^* \geq 0) Y_i X_{ij}\right) \\ \leq 2\exp\left(-\frac{4A(\alpha)n\log p}{4n\Delta T^2}\right) = 2p^{-\frac{A(\alpha)}{\Delta T^2}}. \end{aligned}$$

Therefore $P(c\sqrt{2A(\alpha)}\log p/n \leq c\|\widehat{S}(\boldsymbol{\beta}^*)\|_\infty) \leq (p+1) \cdot 2p^{-\frac{A(\alpha)}{\Delta T^2}} \leq \alpha$.

Proof of Lemma 2. Since $\widehat{\boldsymbol{\beta}}$ minimizes $l_n(\boldsymbol{\beta})$, we have

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \widehat{\boldsymbol{\beta}}) + \lambda \|\widehat{\boldsymbol{\beta}}_-\|_1 &\leq \frac{1}{n} \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^*) + \lambda \|\boldsymbol{\beta}^*\|_1, \\ \frac{1}{n} \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^* + Y_i \mathbf{X}_i^T \mathbf{h}_+) - \frac{1}{n} \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^*) &\leq \lambda \|\boldsymbol{\beta}^*\|_1 - \lambda \|\widehat{\boldsymbol{\beta}}_-\|_1. \end{aligned}$$

Recalling $T = \{1 \leq j \leq p : \beta_j^* \neq 0\}$ and $T_+ = T \cup \{0\}$, we have

$$\begin{aligned} \|\boldsymbol{\beta}^*\|_1 - \|\widehat{\boldsymbol{\beta}}_-\|_1 &\leq \|\boldsymbol{\beta}_{T_+}^*\|_1 - \|\widehat{\boldsymbol{\beta}}_-\|_1 \\ &\leq \|\mathbf{h}_{T_+}\|_1 - \|\mathbf{h}_{T_+^c}\|_1. \end{aligned}$$

This implies

$$\frac{1}{n} \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^* + Y_i \mathbf{X}_i^T \mathbf{h}_+) - \frac{1}{n} \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^*) \leq \lambda (\|\mathbf{h}_{T_+}\|_1 - \|\mathbf{h}_{T_+^c}\|_1). \quad (15)$$

Since the subdifferential of $l_n(\boldsymbol{\beta})$ at the point of $\boldsymbol{\beta}^*$ is $\widehat{S}(\boldsymbol{\beta}^*)$ and recall the assumption $\lambda \geq c\|\widehat{S}(\boldsymbol{\beta}^*)\|_\infty$, we have

$$\begin{aligned} &\frac{1}{n} \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^* + Y_i \mathbf{X}_i^T \mathbf{h}_+) - \frac{1}{n} \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^*) \\ &\geq \widehat{S}^T(\boldsymbol{\beta}^*) \mathbf{h} \\ &\geq -\|\mathbf{h}\|_1 \cdot \|\widehat{S}(\boldsymbol{\beta}^*)\|_\infty \\ &\geq -\frac{\lambda}{c} (\|\mathbf{h}_{T_+}\|_1 + \|\mathbf{h}_{T_+^c}\|_1). \end{aligned}$$

Hence, we have

$$\begin{aligned} \lambda (\|\mathbf{h}_{T_+}\|_1 - \|\mathbf{h}_{T_+^c}\|_1) &\geq -\frac{\lambda}{c} (\|\mathbf{h}_{T_+}\|_1 + \|\mathbf{h}_{T_+^c}\|_1), \\ \|\mathbf{h}_{T_+}\|_1 &\geq C \|\mathbf{h}_{T_+^c}\|_1, \end{aligned}$$

where $C = \frac{c-1}{c+1}$. We have thus proved that $\mathbf{h} \in \Delta_C$.

Proof of Lemma 3. We first consider a fixed $\mathbf{h} \in \mathbb{R}^{p+1}$ such that $\|\mathbf{h}\|_0 \leq q+1$ and $\|\mathbf{h}\|_2 \neq 0$. Note that the Hinge loss function is Lipschitz continuous and we have

$$\frac{|(1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^* + Y_i \mathbf{X}_i^T \mathbf{h}_+) - (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^*)|}{\|\mathbf{h}\|_2} \leq \frac{|\mathbf{X}_i^T \mathbf{h}|}{\|\mathbf{h}\|_2}.$$

By Hoeffding's inequality, we have $\forall t > 0$,

$$P\left(\frac{B(\mathbf{h})}{\|\mathbf{h}\|_2} \geq \frac{t}{\sqrt{n}} |\mathcal{X}\| \right) \leq 2 \exp\left(-\frac{2nt^2}{4\|\mathcal{X}\mathbf{h}\|_2^2/\|\mathbf{h}\|_2^2}\right).$$

Hence by assumption (A3),

$$P\left(\frac{B(\mathbf{h})}{\|\mathbf{h}\|} \geq \frac{t}{\sqrt{n}} |\mathcal{X}\| \right) \leq 2 \exp\left(-\frac{t^2}{2\lambda_{\max}^2}\right) \leq 2 \exp\left(-\frac{t^2}{2M_1^2}\right).$$

Let $t = C\sqrt{2q\log p}$, where C is an arbitrary given positive constant. Then

$$P\left(\frac{B(\mathbf{h})}{\|\mathbf{h}\|} \geq C\sqrt{\frac{2q\log p}{n}}\right) \leq 2 \exp\left(-\frac{C^2 q \log p}{M_1^2}\right) \leq 2p^{-C^2 q/M_1} \leq 2p^{-C^2(q+1)/(2M_1)}.$$

Next we will derive an upper bound for $\sup_{\|\mathbf{h}\|_0 \leq q+1, \|\mathbf{h}\|_2 \neq 0} \frac{B(\mathbf{h})}{\|\mathbf{h}\|}$. We consider covering $\{\mathbf{h} \in \mathbb{R}^{p+1}, \|\mathbf{h}\|_0 \leq q+1\}$ with ϵ -balls such that for any \mathbf{h}_1 and \mathbf{h}_2 in the same ball we have $\left|\frac{\|\mathbf{h}_1\|}{\|\mathbf{h}_2\|} - \frac{\|\mathbf{h}_2\|}{\|\mathbf{h}_1\|}\right| \leq \epsilon$, where ϵ is a small positive number. The number of ϵ -balls that is required to cover a k -dimensional unit ball is bounded by $(3/\epsilon)^k$, see for example Rogers (1963) and Bourgain and Milman (1987). Since \mathbf{h} is a $(p+1)$ -dimensional vector with at most $q+1$ nonzero coordinates and $\mathbf{h}/\|\mathbf{h}\|_2$ has unit length in L_2 norm, the covering number we require is at most $(3p/\epsilon)^{q+1}$. Let N denote such an ϵ -net. By the union bound,

$$P\left(\sup_{\mathbf{h} \in N} \frac{B(\mathbf{h})}{\|\mathbf{h}\|} \geq C\sqrt{\frac{2q\log p}{n}}\right) \leq 2 \left(\frac{3p}{\epsilon}\right)^{q+1} p^{-C^2(q+1)/(2M_1)} = 2 \left(\frac{3}{\epsilon}\right)^{q+1} p^{-C^2/(2M_1)},$$

for any given positive constant C . Furthermore, for any $\mathbf{h}_1, \mathbf{h}_2 \in \mathbb{R}^{p+1}$ such that $\|\mathbf{h}_1\|_0 \leq q+1$, $\|\mathbf{h}_2\|_0 \leq q+1$, $\|\mathbf{h}_1\|_2 \neq 0$ and $\|\mathbf{h}_2\|_2 \neq 0$, we have

$$\begin{aligned} \left| \frac{B(\mathbf{h}_1)}{\|\mathbf{h}_1\|_2} - \frac{B(\mathbf{h}_2)}{\|\mathbf{h}_2\|_2} \right| &\leq \frac{2}{n} \|\mathcal{X}(\mathbf{h}_1/\|\mathbf{h}_1\|_2 - \mathbf{h}_2/\|\mathbf{h}_2\|_2)\|_1 \\ &\leq \frac{2}{\sqrt{n}} \|\mathcal{X}(\mathbf{h}_1/\|\mathbf{h}_1\|_2 - \mathbf{h}_2/\|\mathbf{h}_2\|_2)\|_2 \\ &\leq 2\sqrt{M_1}\epsilon. \end{aligned}$$

Therefore,

$$\sup_{\|\mathbf{h}_1\|_0 \leq q+1, \|\mathbf{h}_1\|_2 \neq 0} \frac{B(\mathbf{h})}{\|\mathbf{h}\|} \leq \sup_{\mathbf{h} \in \mathcal{N}} \frac{B(\mathbf{h})}{\|\mathbf{h}\|} + 2\sqrt{M_1}\epsilon.$$

Let $\epsilon = \sqrt{\frac{2q \log p}{2M_1 n}}$, we have

$$\begin{aligned} P \left(\sup_{\|\mathbf{h}_1\|_0 \leq q+1, \|\mathbf{h}_1\|_2 \neq 0} \frac{B(\mathbf{h})}{\|\mathbf{h}\|} \geq C \sqrt{\frac{2q \log p}{n}} \right) \\ &\leq P \left(\sup_{\mathbf{h} \in \mathcal{N}} \frac{B(\mathbf{h})}{\|\mathbf{h}\|} \geq (C-1) \sqrt{\frac{2q \log p}{n}} \right) \\ &\leq 2(2M_1 n)^{\frac{q+1}{2}} (3p)^{-(C-1)^2/(2M_1)} (q+1) \\ &\leq 2(\sqrt{2M_1 n} 3p)^{1-(C-1)^2/(2M_1)} (q+1). \end{aligned}$$

Since $p > n$, take $C = 1 + 2C_1\sqrt{M_1}$ for some $C_1 > 1$, then for all n sufficiently large,

$$P \left(\sup_{\|\mathbf{h}_1\|_0 \leq q+1, \|\mathbf{h}_1\|_2 \neq 0} \frac{B(\mathbf{h})}{\|\mathbf{h}\|} \geq (1 + 2C_1\sqrt{M_1}) \sqrt{\frac{2q \log p}{n}} \right) \leq 2p^{-2q(C_1^2-1)}.$$

Lemma 6 For any $x \in \mathbb{R}^n$,

$$\|x\|_2 - \frac{\|x\|_1}{\sqrt{n}} \leq \frac{\sqrt{n}}{4} \left(\max_{1 \leq i \leq n} |x_i| - \min_{1 \leq i \leq n} |x_i| \right).$$

Proof. This proof was given in Cai, Wang, and Xu (2010). We include it here for completeness and easy reference. It is obvious that the result holds when $|x_1| = |x_2| = \dots = |x_n|$. Without loss of generality, we now assume that $x_1 \geq x_2 \geq \dots \geq x_n \geq 0$ and not all x_i are equal. Let

$$f(x) = \|x\|_2 - \frac{\|x\|_1}{\sqrt{n}}.$$

Note that for any $i \in \{2, 3, \dots, n-1\}$

$$\frac{\partial f}{\partial x_i} = \frac{x_i}{\|x\|_2} - \frac{1}{\sqrt{n}}.$$

This implies that when $x_i \leq \frac{\|x\|_2}{\sqrt{n}}$, $f(x)$ is decreasing w.r.t x_i ; otherwise $f(x)$ is increasing w.r.t x_i . Hence, if we fix x_1 and x_n , when $f(x)$ achieves its maximum, x must be of the form that $x_1 = x_2 = \dots = x_k$ and $x_{k+1} = \dots = x_n$ for some $1 \leq k \leq n$. Now,

$$f(x) = \sqrt{k(x_1^2 - x_n^2) + nx_n^2} - \frac{k}{\sqrt{n}}(x_1 - x_n) - \sqrt{nx_n}.$$

Treat this as a function of k for $k \in (0, n)$.

$$g(x) = \sqrt{k(x_1^2 - x_n^2) + nx_n^2} - \frac{k}{\sqrt{n}}(x_1 - x_n) - \sqrt{nx_n}.$$

By taking the derivatives, it is easy to see that

$$\begin{aligned} g(k) &\leq g \left(n \frac{(\frac{x_1+x_n}{2})^2 - x_n^2}{x_1^2 - x_n^2} \right) \\ &= \sqrt{n}(x_1 - x_n) \left(\frac{1}{2} - \frac{x_1 + 3x_n}{4(x_1 + x_n)} \right). \end{aligned}$$

Since $\frac{1}{2} - \frac{x_1+3x_n}{4(x_1+x_n)} \geq \frac{1}{4}$, we have

$$\|x\|_2 \leq \frac{\|x\|_1}{\sqrt{n}} + \frac{\sqrt{n}}{4}(x_1 - x_n).$$

We can also see that the above inequality becomes an equality if and only if $x_{k+1} = \dots = x_n = 0$ and $k = \frac{n}{4}$.

Proof of Theorem 4. Let $\mathbf{h} = \boldsymbol{\beta}^* - \hat{\boldsymbol{\beta}}$, then it follows from Lemma 2 that $\mathbf{h} \in \Delta_{\mathcal{C}}$. Assume without loss of generality that $|h_0| \geq |h_1| \geq \dots \geq |h_p|$. Create a partition of $\{0, 1, 2, \dots, p\}$ as

$$S_0 = \{0, 1, 2, \dots, q\}, S_1 = \{q+1, q+2, \dots, 2q+1\}, S_2 = \{2q+2, 2q+3, \dots, 3q+2\}, \dots$$

where S_i , $i = 1, 2, \dots$, has cardinality $q+1$, except the last set which may have cardinality smaller than $q+1$. This partition leads to the following decomposition

$$\begin{aligned} &\frac{1}{n} \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^* + Y_i \mathbf{X}_i^T \mathbf{h})_+ - \frac{1}{n} \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^*)_+ \\ &= \frac{1}{n} \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^* + Y_i \mathbf{X}_i^T \sum_{k \geq 0} \mathbf{h}_{S_k})_+ - \frac{1}{n} \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^*)_+ \\ &= \sum_{j \geq 1} \frac{1}{n} \left(\sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^* + Y_i \mathbf{X}_i^T \sum_{k=0}^j \mathbf{h}_{S_k})_+ - \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^* + Y_i \mathbf{X}_i^T \sum_{k=0}^{j-1} \mathbf{h}_{S_k})_+ \right) \\ &\quad + \frac{1}{n} \left(\sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^* + Y_i \mathbf{X}_i^T \mathbf{h}_{S_0})_+ - \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^*)_+ \right), \end{aligned} \quad (16)$$

where the first equation follows from the definition of \mathbf{h}_{S_k} , $k \geq 0$; the second equation holds by observing that the intermediate terms cancel out each other. The purpose of the above decomposition is to obtain more accurate probability bounds by appealing to Lemma 3. This is made possible by noting that the j th term in the sum of the above decomposition has the increment indexed by \mathbf{h}_{S_j} , which has at most $q+1$ nonzero coordinates. Lemma 3 implies that uniformly for $j = 1, 2, \dots$, with probability at least $1 - 2p^{-2q}(C_1^2 - 1)$,

$$\begin{aligned} & \frac{1}{n} \left(\sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^* + Y_i \mathbf{X}_i^T \sum_{k=0}^j \mathbf{h}_{S_k})_+ - \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^* + Y_i \mathbf{X}_i^T \sum_{k=0}^{j-1} \mathbf{h}_{S_k})_+ \right) \\ & \geq \frac{1}{n} E \left(\sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^* + Y_i \mathbf{X}_i^T \sum_{k=0}^j \mathbf{h}_{S_k})_+ - \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^* + Y_i \mathbf{X}_i^T \sum_{k=0}^{j-1} \mathbf{h}_{S_k})_+ \right) \\ & \quad - C \sqrt{\frac{2q \log p}{n}} \|\mathbf{h}_{S_j}\|_2, \end{aligned}$$

where $C = 1 + 2C_1 \sqrt{M_T}$. Hence by (16), with probability at least $1 - 2p^{-2q}(C_1^2 - 1)$,

$$\frac{1}{n} \left(\sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^* + Y_i \mathbf{X}_i^T \mathbf{h})_+ - \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^*)_+ \right) \geq M(\mathbf{h}) - C \sqrt{\frac{2q \log p}{n}} \sum_{j \geq 0} \|\mathbf{h}_{S_j}\|_2 \quad (17)$$

where $M(\mathbf{h}) = \frac{1}{n} E(\sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^* + Y_i \mathbf{X}_i^T \mathbf{h})_+ - \sum_{i=1}^n (1 - Y_i \mathbf{X}_i^T \boldsymbol{\beta}^*)_+)$.

It is straightforward to show that $\|\mathbf{h}_{S_0}\|_1 \geq \|\mathbf{h}_{T^c}\|_1 \geq C \|\mathbf{h}_{T^c}\|_2 \geq C \|\mathbf{h}_{S_0}\|_1$. By Lemma 6, we have

$$\begin{aligned} \sum_{j \geq 1} \|\mathbf{h}_{S_j}\|_2 & \leq \sum_{j \geq 1} \frac{\|\mathbf{h}_{S_j}\|_1}{\sqrt{q+1}} + \frac{\sqrt{q+1}}{4} |h_q| \\ & \leq \frac{\|\mathbf{h}_{S_0^c}\|_1}{\sqrt{q+1}} + \frac{\|\mathbf{h}_{S_0}\|_1}{4\sqrt{q+1}} \\ & \leq \left(\frac{1}{\sqrt{q+1}C} + \frac{1}{4\sqrt{q+1}} \right) \|\mathbf{h}_{S_0}\|_1 \\ & \leq \left(\frac{1}{4} + \frac{1}{C} \right) \|\mathbf{h}_{S_0}\|_2. \end{aligned} \quad (18)$$

By the definition of \mathbf{h} , (15), (17) and (18), we have

$$\begin{aligned} M(\mathbf{h}) & \leq \lambda (\|\mathbf{h}_{T^c}\|_1 - \|\mathbf{h}_{T^c}\|_1) + \left(\frac{1}{4} + \frac{1}{C} \right) C \sqrt{\frac{2q \log p}{n}} \|\mathbf{h}_{S_0}\|_2 + C \sqrt{\frac{2q \log p}{n}} \|\mathbf{h}_{S_0}\|_2 \\ & \leq \lambda \sqrt{q+1} \|\mathbf{h}_{S_0}\|_2 + C \sqrt{\frac{2q \log p}{n}} \left(\frac{5}{4} + \frac{1}{C} \right) \|\mathbf{h}_{S_0}\|_2. \end{aligned} \quad (19)$$

Condition (A4) imply that

$$M(\mathbf{h}) = \frac{1}{2} \mathbf{h}^T H(\boldsymbol{\beta}^*) \mathbf{h} + o(\|\mathbf{h}\|_2^2) \geq \frac{1}{2} M_2 \|\mathbf{h}\|_2^2 + o(\|\mathbf{h}\|_2^2). \quad (20)$$

Combining (19) and (20), we have

$$\frac{1}{2} M_2 \|\mathbf{h}\|_2^2 + o(\|\mathbf{h}\|_2^2) \leq \lambda \sqrt{q+1} \|\mathbf{h}_{S_0}\|_2 + C \sqrt{\frac{2q \log p}{n}} \left(\frac{5}{4} + \frac{1}{C} \right) \|\mathbf{h}_{S_0}\|_2.$$

Note that $\|\mathbf{h}\|_2^2 = \|\mathbf{h}_{S_0}\|_2^2 + \sum_{j \geq 1} \|\mathbf{h}_{S_j}\|_2^2 \geq \|\mathbf{h}_{S_0}\|_2^2$, and

$$\sum_{j \geq 1} \|\mathbf{h}_{S_j}\|_2^2 \leq |h_q| \sum_{j \geq 1} \|\mathbf{h}_{S_j}\|_1 \leq \frac{1}{C} |h_q| \|\mathbf{h}_{S_0}\|_1 \leq \frac{1}{C} \|\mathbf{h}_{S_0}\|_2^2.$$

So $\|\mathbf{h}\|_2^2 \leq (1 + \frac{1}{C}) \|\mathbf{h}_{S_0}\|_2^2$. This implies $o(\|\mathbf{h}\|_2^2) = o(\|\mathbf{h}_{S_0}\|_2^2)$. To wrap up, we have

$$\|\mathbf{h}_{S_0}\|_2 + o(\|\mathbf{h}_{S_0}\|_2) \leq \frac{2\lambda \sqrt{q+1}}{M_2} + \frac{2C}{M_2} \sqrt{\frac{2q \log p}{n}} \left(\frac{5}{4} + \frac{1}{C} \right).$$

Hence,

$$\|\mathbf{h}\|_2 + o(\|\mathbf{h}\|_2) \leq \sqrt{1 + \frac{1}{C}} \left(\frac{2\lambda \sqrt{q+1}}{M_2} + \frac{2C}{M_2} \sqrt{\frac{2q \log p}{n}} \left(\frac{5}{4} + \frac{1}{C} \right) \right).$$

We therefore have

$$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2 \leq \sqrt{1 + \frac{1}{C}} \left(\frac{2\lambda \sqrt{q+1}}{M_2} + \frac{2C}{M_2} \sqrt{\frac{2q \log p}{n}} \left(\frac{5}{4} + \frac{1}{C} \right) \right)$$

with probability at least $1 - 2p^{-2q}(C_1^2 - 1)$.

Proof of Theorem 5. It follows by combining the result of Theorem 3.3 with that of Theorem 4 in of Zhang et al. (2016b).

Appendix B: Discussions of Condition (A4)

We note that Condition (A4) is satisfied if the smallest eigenvalues of $H(\boldsymbol{\beta}^*)$ has a positive lower bound. In the following, we provide a set of sufficient conditions to guarantee the positive definiteness of $H(\boldsymbol{\beta}^*)$.

(A1*) For some $1 \leq k \leq p$,

$$\int_S I(X_k \geq V_k^-) X_k g(\mathbf{X}) d\mathbf{X} < \int_S I(X_k \leq U_k^+) X_k f(\mathbf{X}) d\mathbf{X}$$

or

$$\int_S I(X_k \leq V_k^+) X_k g(\mathbf{X}) d\mathbf{X} > \int_S I(X_k \geq U_k^-) X_k f(\mathbf{X}) d\mathbf{X}$$

Here U_k^+ , V_k^+ $\in [-\infty, +\infty]$ are upper bounds such that $\int_S I(X_k \leq U_k^+) f(\mathbf{X}) d\mathbf{X} = \min(1, \frac{\pi_k^+}{\pi_k^-})$ and $\int_S I(X_k \leq V_k^+) g(\mathbf{X}) d\mathbf{X} = \min(1, \frac{\pi_k^+}{\pi_k^-})$. Similarly, lower bounds U_k^- , $V_k^- \in [-\infty, +\infty]$ and are defined as $\int_S I(X_k \geq U_k^-) f(\mathbf{X}) d\mathbf{X} = \min(1, \frac{\pi_k^-}{\pi_k^+})$ and $\int_S I(X_k \geq V_k^-) g(\mathbf{X}) d\mathbf{X} = \min(1, \frac{\pi_k^-}{\pi_k^+})$.

(A2*) For an orthogonal transformation A_j that maps $\frac{\beta^*}{\|\beta^*\|_2}$ to the j -th unit vector \mathbf{e}_j for some $j \in \{1, 2, 3, \dots, p\}$, there exists rectangles

$$D^+ = \{x \in M^+ : l_i \leq (A_j x)_i \leq v_i \text{ with } l_i < v_i \text{ for } i \neq j\}$$

and

$$D^- = \{x \in M^- : l_i \leq (A_j x)_i \leq v_i \text{ with } l_i < v_i \text{ for } i \neq j\}$$

such that $f(x) \geq B_1 > 0$ on D^+ , and $g(x) \geq B_2 > 0$ on D^- , where $M^+ = \{x \in \mathbf{R}^p : x^T \beta^+ + \beta_0^+ = 1\}$ and $M^- = \{x \in \mathbf{R}^p : x^T \beta^+ + \beta_0^+ = -1\}$.

Also with some technical modification, Condition (A1) in our paper can be further relaxed to

(A3*) The densities f and g are continuous with common support $\mathcal{S} \subset \mathbb{R}^p$ and have finite second moments.

As an interesting side result, Lemma 5 in Koo et al. (2008) showed that Condition (A4) holds under (A1*)-(A3*). Although their paper's results on the Bahadur representation of L_1 -norm SVM coefficients are restricted to the classical fixed p case, a careful examination of the derivation showed that this particular lemma holds irrespective of the dimension of p .

In the following, we demonstrate that Conditions (A1*)-(A3*) hold in a nontrivial example where we have two multivariate normal distributions in \mathcal{R}^p . The marginal distribution of Y is given by $\pi_+ = \pi_- = 1/2$. Let f and g be the density functions of \mathbf{X}_- given $Y = 1$ and -1 , respectively. Here, we assume f and g are multivariate normal densities with different mean vectors $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ and a common covariance matrix $\boldsymbol{\Sigma}$. This setup was also considered in Koo et al. (2008) but we will provide more details to show condition (A4) is satisfied in our high-dimensional settings. In particular, we will provide some details for deriving the analytic forms of β^* and $H(\beta^*)$, which complements the results in Koo et al. (2008).

For normal density functions f and g , it is straightforward to check Condition (A3*) is satisfied. While $U_k^+ = V_k^+ = +\infty$ and $U_k^- = V_k^- = -\infty$, Condition (A1*) also holds. Since D^+ and D^- are bounded rectangles in \mathbb{R}^p , the normal densities f and g are always bounded away from zero on D^+ and D^- . Thus (A2*) is satisfied. Denote the density and cumulative distribution function of standard normal distribution $N(0, 1)$ as ϕ and Φ , respectively. Then we have $S(\beta^*) = 0$, where $S(\cdot)$ is defined in (6), that is

$$E_f(I(1 - \mathbf{X}^T \beta^* \geq 0)) = E_g(I(1 + \mathbf{X}^T \beta^* \geq 0)) \quad (21)$$

and

$$E_f(I(1 - \mathbf{X}^T \beta^* \geq 0) \mathbf{X}_-) = E_g(I(1 + \mathbf{X}^T \beta^* \geq 0) \mathbf{X}_-) \quad (22)$$

For left hand of equation (21), we have $\mathbf{X}^T \beta^* \sim N(\boldsymbol{\mu}^T \beta^*, \beta^{*T} \boldsymbol{\Sigma} \beta^*)$, thus

$$E_f(I(1 - \mathbf{X}^T \beta^* \geq 0)) = P_f(1 - \beta_0^+ - \mathbf{X}^T \beta^* \geq 0) = \Phi(c_f), \quad (23)$$

where $c_f = \frac{1 - \beta_0^+ - \boldsymbol{\mu}^T \beta^*}{\|\boldsymbol{\Sigma}^{1/2} \beta^*\|_2}$. Similarly, $E_g(I(1 + \mathbf{X}^T \beta^* \geq 0)) = \Phi(c_g)$, where $c_g = \frac{1 + \beta_0^+ + \boldsymbol{\nu}^T \beta^*}{\|\boldsymbol{\Sigma}^{1/2} \beta^*\|_2}$.

To obtain an analytic expression of $E_f(I(1 - \mathbf{X}^T \beta^* \geq 0))$, we consider an orthogonal matrix \mathbf{P} that satisfies $\frac{\mathbf{P} \boldsymbol{\Sigma}^{1/2} \beta^*}{\|\boldsymbol{\Sigma}^{1/2} \beta^*\|_2} = (1, 0, 0, \dots, 0)^T$. Such a matrix \mathbf{P} can always be constructed. Actually, let $\mathbf{P} = (\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_p)^T$ and $\mathbf{P}_1 = \frac{\boldsymbol{\Sigma}^{1/2} \beta^*}{\|\boldsymbol{\Sigma}^{1/2} \beta^*\|_2}$. By using Gram-Schmidt process, we can generate other orthogonal vectors \mathbf{P}_i based on \mathbf{P}_1 with $i = 2, 3, \dots, p$. Since $\mathbf{P} \boldsymbol{\Sigma}^{-1/2} (\mathbf{X}_- - \boldsymbol{\mu}) = \mathbf{Z}$, a standard multivariate normal random vector, we have $I - \mathbf{X}^T \beta^* = c_f \|\boldsymbol{\Sigma}^{1/2} \beta^*\|_2 - \mathbf{Z}^T \mathbf{P} \boldsymbol{\Sigma}^{1/2} \beta^*$. Thus

$$\begin{aligned} E_f(I(1 - \mathbf{X}^T \beta^* \geq 0) \mathbf{X}_-) &= E_\phi(I(c_f - Z_1 \geq 0)(\boldsymbol{\Sigma}^{1/2} \mathbf{P}^T \mathbf{Z} + \boldsymbol{\mu})) \\ &= E_\phi(I(c_f - Z_1 \geq 0) \boldsymbol{\mu}) + E_\phi(I(c_f - Z_1 \geq 0) \boldsymbol{\Sigma}^{1/2} \mathbf{P}^T \mathbf{Z}). \end{aligned}$$

where ϕ is the joint probability density function of a p -dimensional standard multivariate normal distribution. We will compute the above expectation componentwise. Let $\boldsymbol{\Sigma}^{1/2} = \Lambda = (\Lambda_1, \Lambda_2, \dots, \Lambda_p)^T$. For $k = 1, \dots, p$, we have

$$E_f(I(1 - \mathbf{X}^T \beta^* \geq 0) X_k) = \mu_k \Phi(c_f) + E_\phi(I(c_f - Z_1 \geq 0) \Lambda_k^T \sum_{i=1}^p P_i Z_i)$$

where, since Z_2, \dots, Z_p have mean zero and are independent of Z_1 ,

$$\begin{aligned} E_\phi(I(c_f - Z_1 \geq 0) \Lambda_k^T \sum_{i=1}^p (P_i Z_i)) &= E_\phi(I(c_f - Z_1 \geq 0) \Lambda_k^T P_1 Z_1) \\ &= \Lambda_k^T \frac{\boldsymbol{\Sigma}^{1/2} \beta^*}{\|\boldsymbol{\Sigma}^{1/2} \beta^*\|_2} E_\phi(I(c_f - Z_1 \geq 0) Z_1) \\ &= \Lambda_k^T \frac{\boldsymbol{\Sigma}^{1/2} \beta^*}{\|\boldsymbol{\Sigma}^{1/2} \beta^*\|_2} \int_{-\infty}^{+\infty} I(c_f - x \geq 0) x \phi(x) dx \\ &= \Lambda_k^T \frac{\boldsymbol{\Sigma}^{1/2} \beta^*}{\|\boldsymbol{\Sigma}^{1/2} \beta^*\|_2} \int_{-\infty}^{c_f} x \phi(x) dx \end{aligned}$$

Since $x\phi(x)$ is an odd function and $\phi(x)$ is symmetric, we have

$$\int_{-\infty}^{c_f} x \phi(x) dx = \int_{-\infty}^{c_f} x \phi(x) dx = -\frac{1}{\sqrt{2\pi}} \int_{c_f^2}^{+\infty} \frac{1}{2} \exp(-\frac{z}{2}) dz = -\phi(c_f).$$

Therefore, for $k = 1, \dots, p$, $E_f(I(1 - \mathbf{X}^T \beta^* \geq 0) X_k) = \mu_k \Phi(c_f) - \Lambda_k^T \frac{\boldsymbol{\Sigma}^{1/2} \beta^*}{\|\boldsymbol{\Sigma}^{1/2} \beta^*\|_2} \phi(c_f)$. Hence

$$E_f(I(1 - \mathbf{X}^T \beta^* \geq 0) \mathbf{X}_-) = \boldsymbol{\mu} \Phi(c_f) - \phi(c_f) \boldsymbol{\Sigma}^{1/2} \mathbf{P}_1.$$

Similarly,

$$E_g(I(1 + \mathbf{X}^T \beta^* \geq 0) \mathbf{X}_-) = \boldsymbol{\nu} \Phi(c_g) + \phi(c_g) \boldsymbol{\Sigma}^{1/2} \mathbf{P}_1$$

Then, we have

$$\Phi(c_f) = \Phi(c_g) \quad (24)$$

and

$$\mu\Phi(c_f) - \phi(c_f)\Sigma^{1/2}\mathbf{P}_1 = \nu\Phi(c_g) + \phi(c_g)\Sigma^{1/2}\mathbf{P}_1 \quad (25)$$

From (24), we have $\tilde{c} = c_f = c_g$, which implies

$$\beta_-^T(\mu + \nu) = -2\beta_0^* \quad (26)$$

From (25),

$$\frac{\beta_-^*}{\|\Sigma^{1/2}\beta_-^*\|_2} = \frac{\Phi(\tilde{c})}{2\phi(\tilde{c})}\Sigma^{-1}(\mu - \nu) \quad (27)$$

Let $d_\Sigma(\mu, \nu) = ((\mu - \nu)^T \Sigma^{-1} (\mu - \nu))^{1/2}$ be the Mahalanobis distance between μ and ν and $R(x) = \frac{\phi(x)}{\Phi(x)}$. As $\Sigma^{1/2} \frac{\beta_-^*}{\|\Sigma^{1/2}\beta_-^*\|_2}$ has l_2 norm equal to 1, we have $\|\frac{\Phi(\tilde{c})}{2\phi(\tilde{c})}\Sigma^{-1/2}(\mu - \nu)\|_2 = 1$,

i.e., $R(\tilde{c}) = \frac{d_\Sigma(\mu, \nu)}{2}$. $R(x)$ is a monotonically decreasing function, thus we have $\tilde{c} = R^{-1}\left(\frac{d_\Sigma(\mu, \nu)}{2}\right)$. Meanwhile, $\tilde{c} = c_f = \frac{1 - \beta_0^* - \mu^T \beta_-^*}{\|\Sigma^{1/2}\beta_-^*\|_2}$, we can solve the problem based on (26) and (27),

$$\beta_0^* = -\frac{(\mu - \nu)^T \Sigma^{-1} (\mu + \nu)}{2\tilde{c}d_\Sigma(\mu, \nu) + d_\Sigma^2(\mu, \nu)} \quad (28)$$

From (25),

$$\beta_-^* = \frac{2\Sigma^{-1}(\mu - \nu)}{2\tilde{c}d_\Sigma(\mu, \nu) + d_\Sigma^2(\mu, \nu)} \quad (29)$$

By plugging (28) and (29) into (7), we can calculate $H(\beta^*)$ as

$$H(\beta^*) = \frac{\phi(\tilde{c})}{4}(2\tilde{c} + d_\Sigma(\mu, \nu)) \begin{pmatrix} 2 & (\mu + \nu)^T \\ \mu + \nu & H_{22}(\beta^*) \end{pmatrix} \quad (30)$$

where

$$H_{22}(\beta^*) = \mu\mu^T + \nu\nu^T + 2\Sigma + 2 \left(\left(\frac{\tilde{c}}{d_\Sigma(\mu, \nu)} \right)^2 + \frac{\tilde{c}}{d_\Sigma(\mu, \nu)} - \frac{1}{d_\Sigma^2(\mu, \nu)} \right) (\mu - \nu)(\mu - \nu)^T$$

As we have obtained the analytic form of $H(\beta^*)$, we consider Model 1 in Section 5.1 as an example. In Model 1, $q = 5$, $\mu = (0.1, 0.2, 0.3, 0.4, 0.5, 0, \dots, 0)^T$ and $\nu = (-0.1, -0.2, -0.3, -0.4, -0.5, 0, \dots, 0)^T \in \mathbb{R}^p$ and $\pi^+ = \pi^- = 1/2$. The covariance matrix $\Sigma = (\sigma_{ij})$ consists of nonzero entries $\sigma_{ij} = -0.2$ for $1 \leq i \neq j \leq q$ and other entries equal to 0. From (28) and (29), we have $\beta^* = (0, 1.39, 1.47, 1.56, 1.65, 1.74, 0, \dots, 0)^T$. Based on (30), we derived $H(\beta^*)$ and numerically validated its positive-definiteness.

References

- Natalia Becker, Grischka Toedt, Peter Lichter, and Axel Benner. Elastic scad as a novel penalization method for svm classification tasks in high-dimensional data. *BMC bioinformatics*, 12(1):138, 2011.
- Alexandre Belloni and Victor Chernozhukov. l_1 -penalized quantile regression in high-dimensional sparse models. *The Annals of Statistics*, 39(1):82–130, 2011.
- Peter J Bickel, Ya'acov Ritov, and Alexandre B Tsybakov. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009.
- Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152. ACM, 1992.
- Jean Bourgain and Vitaly D Milman. New volume ratio properties for convex symmetric bodies in \mathcal{R}^n . *Inventiones Mathematicae*, 88(2):319–340, 1987.
- Paul S Bradley and Olvi L Mangasarian. Feature selection via concave minimization and support vector machines. In *ICML*, volume 98, pages 82–90, 1998.
- Tony Cai, Lie Wang, and Guangwu Xu. New bounds for restricted isometry constants. *IEEE Transactions on Information Theory*, 56(9):4388–4394, 2010.
- Tony Cai, Weidong Liu, and Xi Luo. A constrained l_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607, 2011.
- Yin-Wen Chang and Chih-Jen Lin. Feature ranking using linear svm. *Causation and Prediction Challenge Challenges in Machine Learning*, 2:47, 2008.
- Jiahua Chen and Zehua Chen. Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3):759–771, 2008.
- R Dennis Cook, Bing Li, and Francesca Chiaromonte. Dimension reduction in regression without matrix inversion. *Biometrika*, 94(3):569–584, 2007.
- Jiangang Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.
- Eitan Greenshtein et al. Best subset selection, persistence in high-dimensional statistical learning and optimization under l_1 constraint. *The Annals of Statistics*, 34(5):2367–2386, 2006.
- Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1):389–422, 2002.

- Ja-Yong Koo, Yoonkyung Lee, Yuwon Kim, and Changyi Park. A bahadur representation of the linear support vector machine. *The Journal of Machine Learning Research*, 9: 1343–1368, 2008.
- Yi Lin. Support vector machines and the bayes rule in classification. *Data Mining and Knowledge Discovery*, 6(3):259–275, 2002.
- Nicolai Meinshausen and Bin Yu. Lasso-type recovery of sparse representations for high-dimensional data. *The Annals of Statistics*, pages 246–270, 2009.
- Changyi Park, Kwang-Rae Kim, Rangmi Myung, and Ja-Yong Koo. Oracle properties of scad-penalized support vector machine. *Journal of Statistical Planning and Inference*, 142(8):2257–2270, 2012.
- Guilherme V Rocha, Xing Wang, and Bin Yu. Asymptotic distribution and sparsity for l_1 -penalized parametric m-estimators with applications to linear svm and logistic regression. *arXiv preprint arXiv:0908.1940*, 2009.
- CA Rogers. Covering a sphere with spheres. *Mathematika*, 10(02):157–164, 1963.
- Minghu Song, Curt M Breneman, Jinbo Bi, Nagamani Sukumar, Kristin P Bennett, Steven Cramer, and Nihal Tugcu. Prediction of protein retention times in anion-exchange chromatography systems using support vector regression. *Journal of Chemical Information and Computer Sciences*, 42(6):1347–1357, 2002.
- Bernadetta Tarigan and Sara Anna van de Geer. *Adaptivity of Support Vector Machines with L_1 Penalty*. University of Leiden. Mathematical Institute, 2004.
- Bernadetta Tarigan, Sara A Van De Geer, et al. Classifiers of support vector machine type with l_1 complexity regularization. *Bernoulli*, 12(6):1045–1076, 2006.
- Sara A Van de Geer. High-dimensional generalized linear models and the lasso. *The Annals of Statistics*, 36(2):614–645, 2008.
- Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer Science & Business Media, 1995.
- Lan Wang, Yichao Wu, and Runze Li. Quantile regression for analyzing heterogeneity in ultra-high dimension. *Journal of the American Statistical Association*, 107(497):214–222, 2012.
- Lie Wang. The l_1 penalized lad estimator for high dimensional linear regression. *Journal of Multivariate Analysis*, 120:135–151, 2013.
- Lifeng Wang and Xiaotong Shen. On l_1 -norm multiclass support vector machines: methodology and theory. *Journal of the American Statistical Association*, 102(478):583–594, 2007.
- Marten Wegkamp and Ming Yuan. Support vector machines with a reject option. *Bernoulli*, 17(4):1368–1385, 2011.
- Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894–942, 2010.
- Hao Helen Zhang, Jeongyoum Ahn, Xiaodong Lin, and Cheolwoo Park. Gene selection using support vector machines with non-convex penalty. *Bioinformatics*, 22(1):88–95, 2006.
- Xiang Zhang, Yichao Wu, Lan Wang, and Runze Li. A consistent information criterion for support vector machines in diverging model spaces. *Journal of Machine Learning Research*, 17(16):1–26, 2016a.
- Xiang Zhang, Yichao Wu, Lan Wang, and Runze Li. Variable selection for support vector machines in moderately high dimensions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(1):53–76, 2016b.
- Ji Zhu, Saharon Rosset, Trevor Hastie, and Rob Tibshirani. 1-norm support vector machines. *Advances in Neural Information Processing Systems*, 16(1):49–56, 2004.
- Hui Zou. An improved 1-norm svm for simultaneous classification and variable selection. *Journal of Machine Learning Research*, 2:675–681, 2007.

Blending Learning and Inference in Conditional Random Fields

Tamir Hazan

*Technion - Israel Institute of Technology
Haifa, 3200, Israel*

TAMIR.HAZAN@TECHNION.AC.IL

Alexander G. Schwing

*Electrical and Computer Engineering and Coordinated Science Laboratory,
University of Illinois at Urbana-Champaign
Urbana, IL 61801*

ASCHWING@ILLINOIS.EDU

Raquel Urtasun

*University of Toronto
40 St. George Street,
Toronto, ON, M5S 2E4*

URTASUN@CS.TORONTO.EDU

Editor: Sebastian Nowozin

Abstract

Conditional random fields maximize the log-likelihood of training labels given the training data, e.g., objects given images. In many cases the training labels are structures that consist of a set of variables and the computational complexity for estimating their likelihood is exponential in the number of the variables. Learning algorithms relax this computational burden using approximate inference that is nested as a sub-procedure. In this paper we describe the objective function for nested learning and inference in conditional random fields. The devised objective maximizes the log-beliefs — probability distributions over subsets of training variables that agree on their marginal probabilities. This objective is concave and consists of two types of variables that are related to the learning and inference tasks respectively. Importantly, we afterwards show how to blend the learning and inference procedure and effectively get to the identical optimum much faster. The proposed algorithm currently achieves the state-of-the-art in various computer vision applications.

1. Introduction

Learning and inference of structured models drives much of the research in machine learning applications, from computer vision and natural language processing to computational biology. Examples include object detection (e.g., by Felzenszwalb et al. (2010)), parsing (e.g., by Koo et al. (2010)), or protein design (e.g., by Sontag et al. (2008)). The inference problem in these cases involves assessing the likelihood of labelings, whether outlined objects, parse trees, or molecular structures. The learning procedure searches for the parameters that maximize the likelihood of the training set.

Conditional random fields (CRFs) form an effective framework for maximizing the log-likelihood of training labels in structured models. Learning the parameters of these models

can be computationally expensive since the label space of real-world problems is usually exponential in the size of its variables.

When the label structure corresponds to a tree, exactly inferring the likelihood of the training labels can be done in time, linear in the number of variables using sum-product belief propagation as a subroutine. We refer to this approach as nested learning and inference. In contrast, when the label structure corresponds to a general graph, we cannot infer the likelihood exactly. However nested learning and inference can still be applied using approximate inference algorithms such as convex sum-product belief propagation (Wainwright et al., 2005; Heskes, 2006; Meltzer et al., 2009; Hazan and Shashua, 2010). Nevertheless, the approximate inference algorithms might be computationally expensive to be used as a nested subroutine of the learning algorithm.

In this article we suggest to interleave optimization of learning parameters with optimization of inference parameters. We call this approach blending learning and inference in CRFs. For this end we propose an optimization program that maximizes log-beliefs, i.e., probability distributions over subsets of training variables that agree on their marginal probabilities. These beliefs are elements of the local polytope (Wainwright and Jordan, 2008). The log-beliefs objective is concave and consists of two types of variables that are related to the learning and inference tasks respectively. We are able to blend the learning and inference procedures by alternating maximizations over the inference and learning variables. With blending we reach the nested learning-inference optimum much faster.

We also define loss-adjusted beliefs to integrate prior knowledge about the desired inference as well as a parameter that controls the smoothness of the beliefs. In the past and partly due to its efficiency, the presented machine learning algorithm was shown to improve the state-of-the-art in various computer vision tasks, including 2D scene understanding (Yao et al., 2012), 3D scene understanding (Lin et al., 2013), shape reconstruction (Salzmann and Urtasun, 2012), indoor scene understanding (Schwing et al., 2012a; Schwing and Urtasun, 2012), depth estimation (Yamaguchi et al., 2012), flow estimation (Yamaguchi et al., 2013) and visual-language understanding (Fidler et al., 2013). This manuscript extends our previous work (Hazan and Urtasun, 2010) to high order setting while simplifying its theory and proofs. The code is publicly available on <http://www.alexander-schwing.de/projects/GeneralStructuredPrediction/LatentVariables.php>.

The reminder of the paper is organized as follows. In Section 3 we review the parameter learning setting of CRFs that maximize the log-likelihood of training labels given corresponding data instances. We also present the nested learning and inference approach as well as approximate inference algorithms of belief propagation and its convex variants. In Section 4 we describe the objective function for nested learning and inference that maximize the log-beliefs. We then describe a blending algorithm to optimize this objective and describe its convergence properties using convex duality. Next we present loss-adjusted beliefs and the appropriate modifications for blending in Section 5, drawing connections to blending in the structured SVMs setting suggested by Meshi et al. (2010). We then demonstrate the effectiveness of our approach in Section 6. We conclude by describing the generality of our approach, relating blending and convexity to the penalty method.

2. Related work

Learning log-beliefs extends the CRFs framework that maximizes the log-likelihood of conditional Gibbs distributions (cf. Lafferty et al. (2001); Lebon and Lafferty (2002)). Gibbs distributions, also known as Markov random fields, are probability distributions that are defined on a product space using potential functions over subsets of variables. Gibbs probability models often consider exponentially many possible assignments for these variables. In this case, approximate inference methods are used in a black box manner to estimate the gradient and the objective of CRFs, resulting in the nested learning-inference algorithm illustrated in Figure 1. Nested learning-inference algorithms are successfully dealing with real-world problems, e.g.: (Levin and Weiss, 2006) in computer vision, (Yanover et al., 2007) in computational biology and (Sutton and McCallum, 2009) in language processing to name a few.

The current work extends and simplifies our previous work (Hazan and Urtasun, 2010). We simplify the learning objective while formulating it as the maximization of log-beliefs, which are pseudo-marginal probabilities of the Gibbs distribution. We extend the learning procedure while considering pseudo-marginals of Gibbs distributions on any subset of its variables. In the last couple of years, the presented machine learning algorithm was shown to improve the state-of-the-art in various computer vision tasks: Considering outdoor scene understanding (Yao et al., 2012), each (super)pixel is represented by a variable of the Gibbs model and its possible assignments correspond to discrete semantic label, e.g., person, car, tree and so on. Our learning algorithm estimates the parameters that maximize the probability of the training data per variable (i.e., pixel and its observed object) and subsets of variables (e.g., neighborhood of pixels and their observed objects). Considering indoor scene understanding (Schwing et al., 2012a; Schwing and Urtasun, 2012; Lin et al., 2013) each wall or a 3D object in the room is represented by a subset of variables and our learning algorithm estimates the position of these objects while maximizing likelihood of these subsets within the training data. In depth estimation and optimal flow (Yamaguchi et al., 2012, 2013) the variables of the Gibbs models are either continuous or discrete. The continuous variables correspond to the possible hyperplanes that either explain the depth or the optical flow of the super pixels in the training images. The discrete variables maintain consistency between adjacent super pixels. Our learning algorithm estimates the probable hyperplanes and their spatial relations in the training data.

Our approach suggests to blend the inference and learning steps and reach the same optimum as nested approaches while being at least an order of magnitude faster. Blending helps the algorithm to avoid computationally expensive inference algorithms when learning parameters w that are far from optimal. Similar observations have been made in the context of coordinate descent by Tappen et al. (2013). Other approaches for blending CRFs appear are developed by Donke (2011). Lemma 3 describes how to infer non-consistent beliefs. These beliefs are different from the beliefs that are usually derived during the runtime of approximate inference algorithms. The theoretical characterization of the optimal points for the learning-inference procedures are described by Wainwright (2006); Wainwright et al. (2003).

Meshi et al. (2010) describe blending learning and inference in the context of structured SVMs, which are constructed to minimize the loss between the predicted labels and the

observed ones (cf. Taskar et al., 2004; Tschantz et al., 2004; Collins, 2002)). The ideas of blending learning and inference in structured SVMs also appear in (Taskar et al., 2005; Anguelov et al., 2005). In contrast, our work focuses on blending learning and inference when maximizing log-probabilities. Nevertheless, our loss-adjusted beliefs may describe a probabilistic alternative to blended learning in structured-SVMs. When setting the counting numbers $c_r = 0$, we effectively work with zero-one probabilities (i.e., max-beliefs) thus we recover the algorithm of Meshi et al. (2010).

3. Background

Log-likelihood learning in structured models involves data instances $x \in \mathcal{X}$ and their labels $y \in \mathcal{Y}$. The structure is incorporated into the labels which may refer to sequences, grids, or other high-dimensional objects. For every data instance x , its possible labels are described by a set of feature functions $\phi_k : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$, $k \in \{1, \dots, K\}$. A linear combination of the K features is used to score the different labels $y \in \mathcal{Y}$ using the parameters $w \in \mathbb{R}^K$. Formally we obtain the score $\theta(y; x, w)$ of a label via

$$\theta(y; x, w) = \sum_k w_k \phi_k(x, y).$$

The real valued score is mapped to the probability scale via the Gibbs distribution:

$$p(y|x; w) \propto \exp(\theta(y; x, w)). \quad (1)$$

Within the CRF framework, the goal is to learn the parameters of the potential functions to maximize the conditional likelihood of the training data $(x, y) \in \mathcal{S}$:

$$\max_w \sum_{(x,y) \in \mathcal{S}} \log p(y|x; w) - \frac{C}{2} \|w\|_2^2. \quad (2)$$

The regularization term is sometimes considered as a Gaussian prior over the parameters w . The regularized likelihood of CRFs is a concave and smooth function and its optimal parameters may be attained by gradient ascent. The gradient measures the disagreements between the inferred distribution over labels and the groundtruth training labels, i.e.,

$$\frac{\partial \log p(y|x; w)}{\partial w_k} = \sum_{g \in \mathcal{Y}} p(g|x; w) \phi_k(x, g) - \phi_k(x, y). \quad (3)$$

The computational complexity of CRFs is governed by inference, which amounts to evaluating the probability $p(g|x; w)$ for computing the objective and the gradient.

We consider cases in which the labels $y \in \mathcal{Y}$ are n -tuples, i.e., $y = (y_1, \dots, y_n)$, and hence the configuration space is exponential in n . The features describe relations between subsets of elements $r \subset \{1, \dots, n\}$, also called regions or factors. We denote by \mathcal{R}_k the regions required to compute the feature $\phi_k(x, y)$. Importantly, the features are functions of their regions labels $y_r \subset \{y_1, \dots, y_n\}$, i.e.,

$$\phi_k(x, y_1, \dots, y_n) = \sum_{r \in \mathcal{R}_k} \phi_{k,r}(x, y_r). \quad (4)$$

Thus the features define hypergraphs whose nodes represent the n labels indexes, and the regions $\mathcal{R} = \cup_k \mathcal{R}_k$ correspond to its hyperedges. A convenient way to represent a hypergraph is by its region graph. A region graph is a directed graph whose nodes represent the regions and its direct edges correspond to the inclusion relation, i.e., a directed edge from node r to s is possible only if $s \subset r$. We adopt the terminology where the sets $P(r)$ and $C(r)$ represent all nodes that are parents and children of the node r , respectively.

The Hammersley-Clifford theorem (e.g., Lauritzen (1996)) asserts that the Gibbs distributions $p(\hat{y}|x; w)$ defined in Equation (1) correspond to a Markov random field (MRF) whose statistical independencies are described by the joint hypergraph. These independencies are determined by the Markov property: two nodes in the graph are conditionally independent when they are separated by observed nodes. Aji and McEliece (2001) show that whenever the region graph is bipartite and has no cycles, the Markov property provides a low dimensional representation of the Gibbs distribution using its marginal probabilities $p(\hat{y}_r|x; w) = \sum_{\hat{y}_{\setminus r}} p(\hat{y}|x; w)$, namely

$$p(\hat{y}|x; w) = \prod_{r \in \mathcal{R}} p(\hat{y}_r|x; w)^{1-|P(r)|}. \quad (5)$$

In such cases the inference step, i.e., estimating the probabilities $p(\hat{y}|x; w)$ can be performed efficiently using message-passing algorithms. When the region graph has no cycles it is bipartite, therefore it has two types of regions: outer regions, i.e., regions that are not contained by other regions, and inner regions. To differentiate between those regions we denote outer regions by α and inner regions by i . In this case, one can use the belief propagation algorithm to efficiently infer the marginal probabilities without performing exponentially many operations:

Algorithm 1 Sum-product belief propagation

Set $\mathcal{K}_r = \{k : r \in \mathcal{R}_k\}$. For every (x, y) , we set $\theta_r(\hat{y}_r) = \sum_{k \in \mathcal{K}_r} w_k \phi_{k,r}(x, \hat{y}_r)$.

Repeat until convergence:

$$\mu_{\alpha \rightarrow i}(y_i) = \log \left(\sum_{y_\alpha \setminus y_i} \exp \left((\theta_\alpha(y_\alpha) + \sum_{j \in C(\alpha) \setminus i} \lambda_{j \rightarrow \alpha}(y_j)) \right) \right)$$

$$\lambda_{i \rightarrow \alpha}(y_i) = \theta_i(y_i) + \sum_{\beta \in P(i) \setminus \alpha} \mu_{\beta \rightarrow i}(y_i)$$

Output:

$$b_i(y_i) \propto \exp \left(\theta_i(y_i) + \sum_{\alpha \in P(i)} \mu_{\alpha \rightarrow i}(y_i) \right)$$

$$b_\alpha(y_\alpha) \propto \left(\theta_\alpha(y_\alpha) + \sum_{i \in C(\alpha)} \lambda_{i \rightarrow \alpha}(y_i) \right)$$

The marginal probabilities $p(\hat{y}_r|x; w)$ appear as the beliefs $b_r(\hat{y}_r)$. In general, when the region graph has cycles, the belief propagation algorithm is no longer guaranteed to output the marginal probabilities. Nevertheless, when it converges it provides beliefs that agree on their marginal probabilities, namely $\sum_{y_\alpha \setminus y_i} b_\alpha(y_\alpha) = b_i(y_i)$. In some cases the belief propagation algorithm infers beliefs $b_r(\hat{y}_r)$ which approximate well the marginal probabilities, while in other cases it produces non-accurate results or might even fail to converge. A possi-

ble explanation for this behavior comes from the fact that the belief propagation algorithm iterates the stationary points of a non-convex objective called the Bethe free energy (Yedidia et al., 2005). Recently, in an extensive effort to derive converging belief propagation type algorithms, the non-convex Bethe free energy was replaced by convex free energies while introducing nonnegative counting numbers c_r that replace the Bethe coefficients. Consequently, the belief propagation algorithm was replaced by block coordinate descent over the dual program (Heskes, 2006; Meltzer et al., 2009; Hazan and Shashua, 2008). These dual block coordinate descent algorithms belong to the family of the norm-product belief propagation algorithms:

Algorithm 2 Norm-Product Belief Propagation

Set $\hat{c}_i = c_i + \sum_{\alpha \in P(i)} c_\alpha$. Repeat until convergence:

$$\mu_{\alpha \rightarrow i}(y_i) = c_\alpha \log \left(\sum_{y_\alpha \setminus y_i} \exp \left((\theta_\alpha(y_\alpha) + \sum_{j \in C(\alpha) \setminus i} \lambda_{j \rightarrow \alpha}(y_j)) / c_\alpha \right) \right)$$

$$\lambda_{i \rightarrow \alpha}(y_i) = \frac{c_i}{c_i} \left(\theta_i(y_i) + \sum_{\beta \in P(i)} \mu_{\beta \rightarrow i}(y_i) \right) - \mu_{\alpha \rightarrow i}(y_i)$$

Output:

$$b_i(y_i) \propto \exp \left(\theta_i(y_i) + \sum_{\alpha \in P(i)} \mu_{\alpha \rightarrow i}(y_i) \right)^{1/c_i}$$

$$b_\alpha(y_\alpha) \propto \exp \left(\theta_\alpha(y_\alpha) + \sum_{i \in C(\alpha)} \lambda_{i \rightarrow \alpha}(y_i) \right)^{1/c_\alpha}$$

The norm-product algorithm, illustrated in Algorithm 2, reduces to belief propagation when setting its coefficients to $c_r = 1 - |P(r)|$, namely, the Bethe counting numbers. We refer the interested reader to (Wainwright and Jordan, 2008) for more details.

The norm-product algorithm iterates over the fixed point solutions for the variational problem

$$\arg \max_{b \in L(\mathcal{C})} \sum_{r \in \mathcal{R}} b_r(y_r) \theta_r(y_r) + \sum_r c_r H(b_r). \quad (6)$$

The set $L(\mathcal{C})$ is known as the local polytope, and contains probability distributions $b_r(y_r)$ that agree on their overlapping variables, i.e.,

$$L(\mathcal{C}) = \left\{ b_r(y_r) : b_r(y_r) \geq 0, \sum_{y_r} b_r(y_r) = 1, \forall p \in P(r) \sum_{y_p \setminus y_r} b_p(y_p) = b_r(y_r) \right\}. \quad (7)$$

Throughout this work we refer to elements in the local polytope as beliefs.

The variational program given in Equation (6) is concave whenever $c_r \geq 0$. The norm-product algorithm given in Algorithm 2 performs block coordinate descent on its dual program. Therefore it is guaranteed to converge to beliefs that agree on their marginal probabilities. Typically its inferred beliefs approximate the marginal probabilities as well as the belief propagation approximations (Meshi et al., 2009). Thus in its various forms it is used as the inference step when learning the parameters of CRFs. Such nested loop algorithms for performing learning and inference are presented in Figure 1. Unfortunately, iteratively executing the norm-product algorithm as an inference procedure to compute

Learning with nested inference for CRFs

1. Set $\theta_r(y_r; x, w) = \sum_{k \in K_r} w_k \phi_{k,r}(x, y_r)$. Repeat until convergence:
2. Inference: For every $(x, y) \in S$:

$$b^* = \arg \max_{b(\cdot|x) \in L(G)} \sum_{r \in \mathcal{R}} \sum_{y_r \in \mathcal{Y}_r} b_r(y_r|x) \theta_r(y_r; x, w) + \sum_{r \in \mathcal{R}} H(b_r).$$

3. Learning:

$$w_k \leftarrow w_k - \eta \left(\sum_{(x,y) \in S} \sum_{r \in \mathcal{R}} \left(\sum_{y_r} b_r^*(y_r|x) \phi_{k,r}(x, y_r) - \phi_{k,r}(x, y_r) \right) + C w_k \right).$$

Figure 1: Nested (unblended) inference in learning. The inference step is performed in every iteration. The learning step improves the parameters till learning is attained. η is typically referred as the learning rate and may be set as $1/\sqrt{t}$, where t is the iteration index. The abbreviation $b \in L(G)$ describes conditional beliefs $b(\cdot|x)$ for every $x \in S$. Each of these conditional beliefs is in the local polytope, as defined in Equation (7).

the gradient is computationally demanding and this method has not been used widely (see Section 2 for more details). In the following we explore duality to provide the means to blend the learning and inference tasks efficiently.

4. Blending learning and inference

Log-likelihood of Gibbs distributions, as defined in Equation (1) with potentials $\theta(y; x, w)$ that are linear functions of their parameters w , results in a concave program. When using nested (unblended) inference, the learning algorithm executes a concave program for inferring beliefs about its marginal probabilities that are required for computing its gradient. Our main result explicitly defines the concave program whose optimal solutions are the limit points of the nested learning and inference algorithm that is shown in Figure 1. Using this characterization we are able to derive an algorithm that blends the learning and inference steps. Consequently it is orders of magnitude faster than the nested algorithm that uses inference as a black-box algorithm. Since our blended algorithm optimizes a concave program, it is guaranteed to reach the same optimum as the nested algorithm.

Theorem 1 *The limit points of the nested inference and learning algorithm in Figure 1 are described by the optimal points of the following concave program maximizing log-beliefs:*

$$\begin{aligned} & \max_{w, \lambda} \sum_{(x,y) \in S} \sum_{r \in \mathcal{R}} \log b_r(y_r|x; w, \lambda) - \frac{C}{2} \|w\|^2 \\ \text{s.t. } & b_r(y_r|x; w, \lambda) \propto \exp \left(\theta_r(y_r; x, w) + \sum_{c \in C(r)} \lambda_{c \rightarrow r}(y_c; x) - \sum_{p \in P(r)} \lambda_{r \rightarrow p}(y_r; x) \right) \end{aligned}$$

Proof The theorem replaces maximization over $b \in L(G)$ with maximization over λ . Decoupling the maximizations takes the form $\max_w \sum_{(x,y) \in S} \left(\max_{\lambda} \left\{ \sum_{r \in \mathcal{R}} \log b_r(y_r|x; w, \lambda) \right\} \right)$. In the following we show how to derive the result from Lagrange optimality conditions (KKT):

$$\begin{aligned} \lambda^* &= \arg \max_{\lambda} \sum_{r \in \mathcal{R}} \log b_r(y_r|x; w, \lambda) \\ b^* &= \arg \max_{b \in L(G)} \sum_{r \in \mathcal{R}} \sum_{y_r \in \mathcal{Y}_r} b_r(y_r|x) \theta_r(y_r; x, w) + \sum_{r \in \mathcal{R}} H(b_r) \\ &\implies b_r^*(y_r|x) \propto \exp \left(\theta_r(y_r; x, w) + \sum_{c \in C(r)} \lambda_{c \rightarrow r}^*(y_c; x) - \sum_{p \in P(r)} \lambda_{r \rightarrow p}^*(y_r; x) \right) \end{aligned} \quad (8)$$

Using the above Lagrange optimality conditions, the theorem follows since the inference nested within the learning algorithm applies gradient ascent to the following program:

$$\max_w \sum_{(x,y) \in S} \left(\max_{\lambda} \left\{ \sum_{r \in \mathcal{R}} \log b_r(y_r|x; w, \lambda) \right\} - \frac{C}{2} \|w\|^2 \right) = \max_w \sum_{(x,y) \in S} \sum_{r \in \mathcal{R}} \log b_r^*(y_r|x) - \frac{C}{2} \|w\|^2.$$

Equation (8) states the Lagrange optimality conditions for maximum-likelihood maximum-entropy type duality. Consider the constrained inference algorithm in Figure (1) and the Lagrange multipliers $\lambda_{r \rightarrow p}(y_r; x, w)$ for the marginalization constraints $\sum_{y_r} b_r(y_r|x) = b_r(y_r|x)$. Its corresponding Lagrangian is $L(b, \lambda) = \sum_{r, y_r} b_r(y_r|x) \theta_r(y_r; x, w, \lambda) + \sum_r H(b_r)$ where $\theta_r(y_r; x, w, \lambda) = \theta_r(y_r; x, w) + \sum_{c \in C(r)} \lambda_{c \rightarrow r}(y_c; x) - \sum_{p \in P(r)} \lambda_{r \rightarrow p}(y_r; x)$. Its dual function is $q(\lambda) = \max_b L(b, \lambda)$ while $b_r(y_r|x)$ are subject to probability constraints. Conjugate duality between the entropy function and the log-partition function (cf. Wainwright and Jordan (2008)) implies that $q(\lambda) = \sum_r \log \left(\sum_{y_r} \exp(\theta_r(y_r; x, w, \lambda)) \right)$. Since strong duality between the entropy and the log-partition function holds, its Lagrange optimality conditions imply

$$\lambda^* = \arg \min_{\lambda} \sum_{r \in \mathcal{R}} \log \left(\sum_{y_r} \exp(\theta_r(y_r; x, w, \lambda)) \right).$$

The theorem then follows since $\sum_{r \in \mathcal{R}} \sum_{p \in P(r)} \lambda_{r \rightarrow p}(y_r; x) - \sum_{r \in \mathcal{R}} \sum_{c \in C(r)} \lambda_{c \rightarrow r}(y_c; x) \equiv 0$ therefore $\sum_r \log b_r^*(y_r|x) = -q(\lambda^*)$ and the Lagrange optimality conditions in Equation (8) hold. Thus $b_r^*(y_r|x)$ can be replaced by its re-parameterization $\frac{1}{Z_r} \exp(\theta_r(y_r; x, w, \lambda^*))$, with $Z_r = \sum_{y_r} \exp(\theta_r(y_r; x, w, \lambda^*))$. ■

The maximum log-beliefs program describes the variational landscape of nested inference in learning. Thus it can be used to measure the step-size η in Figure 1. For example, the Armijo rule that determines the step size according to the variational neighborhood results in a faster convergence per iteration than using a fixed learning rate.

The nested learning and inference algorithm performs a complete inference step before performing a single learning step. The inference step derives beliefs that agree on their marginal probabilities. In this work we use the maximum log-beliefs concave program to blend the learning and inference steps. Specifically, we use block coordinate ascent steps

to blend learning (optimization w.r.t. w) with incomplete inference steps (optimization w.r.t. λ) that derive beliefs $b_r(y_r|x; w, \lambda)$ that not necessarily agree on their marginal probabilities, i.e., $\sum_{y_p \in \mathcal{Y}_p} b_p(y_p|x; w, \lambda) = b_r(y_r|x; w, \lambda)$. Such an approach is computationally favorable since it does not require to perform a complete inference step for initial learning parameters. Concavity ensures that blending reaches the maximum log-beliefs optimum thus it guarantees to derive consistent beliefs upon convergence.

Performing block coordinate descent on the maximum log-beliefs program in Theorem 1 requires minimizing a block of variables while holding the rest fixed. We begin by describing how to infer the optimal set of variables $\lambda_{r \rightarrow p}(y_r; x)$ that are related to a region and its parents in the graphical model.

Lemma 2 Blended inference: *Consider the program given in Theorem 1. For a given region r , the optimal inference parameters $\lambda_{r \rightarrow p}^*(y_r; x)$, for every $p \in P(r)$, $y_r \in \mathcal{Y}_r$, $x \in \mathcal{S}$, when fixing all other λ takes the following form:*

$$\begin{aligned} \mu_{p \rightarrow r}(y_p; x) &= \log \left(\sum_{y_p \in \mathcal{Y}_p} \exp(\theta_p(y_p; x; w) + \sum_{c \in C(p) \setminus r} \lambda_{c \rightarrow p}(y_c; x) - \sum_{p' \in P(p)} \lambda_{p \rightarrow p'}(y_{p'}; x)) \right) \\ \lambda_{r \rightarrow p}^*(y_r; x) &= \frac{\theta_r(y_r; x; w) + \sum_{c \in C(r)} \lambda_{c \rightarrow r}(y_c; x) + \sum_{p' \in P(r)} \mu_{p' \rightarrow r}(y_{p'}; x)}{1 + |P(r)|} - \mu_{p \rightarrow r}(y_p; x) \end{aligned}$$

Proof The maximum log-beliefs program in Theorem 1 is smooth, concave and unconstrained as a function of λ , therefore the optimum is achieved when the gradient vanishes. Setting

$$\theta_r(y_r; x; w, \lambda) = \theta_r(y_r; x; w) + \sum_{c \in C(r)} \lambda_{c \rightarrow r}(y_c; x) - \sum_{p \in P(r)} \lambda_{r \rightarrow p}(y_p; x) \quad (9)$$

and $b_r(y_p|x; w, \lambda) \propto \exp(\theta_r(y_p|x; w, \lambda))$, the gradient with respect to $\lambda_{r \rightarrow p}(y_p; x)$ takes the form

$$\frac{\partial \sum_{(x, y), r} \log b_r(y_r|x; w, \lambda)}{\partial \lambda_{r \rightarrow p}(y_r; x)} = \sum_{y_p \in \mathcal{Y}_p} b_p(y_p|x; w, \lambda) - b_r(y_r|x; w, \lambda).$$

The optimal dual variables are those for which the gradient vanishes, i.e., the corresponding beliefs agree on their marginal probabilities. When setting $\mu_{p \rightarrow r}(y_r; x)$ as above, the marginalization of $b_p(y_p|x; w, \lambda)$ satisfies

$$\sum_{y_p \in \mathcal{Y}_p} b_p(y_p|x; w, \lambda) \propto \exp(\mu_{p \rightarrow r}(y_r; x) + \lambda_{r \rightarrow p}(y_r; x)).$$

Therefore, by taking the logarithm, the gradient vanishes whenever the beliefs numerators agree up to an additive constant:

$$\sum_{y_p \in \mathcal{Y}_p} b_p(y_p|x; w, \lambda^*) - b_r(y_r|x; w, \lambda^*) = 0 \iff \mu_{p \rightarrow r}(y_r; x) + \lambda_{r \rightarrow p}^*(y_r; x) = \theta_r(y_r; x; w, \lambda^*).$$

The right hand side of the condition almost characterizes completely the optimal variables $\lambda_{r \rightarrow p}^*(y_r; x)$ by $\lambda_{r \rightarrow p}^*(y_r; x) = \theta_r(y_r; x; w, \lambda^*) - \mu_{p \rightarrow r}(y_r; x)$. Unfortunately, $\theta_r(y_r; x; w, \lambda^*)$ depends on $\sum_{p \in P(r)} \lambda_{r \rightarrow p}^*(y_r; x)$ thus it cannot serve as an update rule in its current form.

To complete the proof we require to replace $\sum_{p \in P(r)} \lambda_{r \rightarrow p}^*(y_r; x)$ with another quantity that does not depend on $\lambda_{r \rightarrow p}^*(y_r; x)$ for every $p \in P(r)$ and $y_r \in \mathcal{Y}_r$.

To isolate this quantity we sum both sides of the equality $\mu_{p \rightarrow r}(y_r; x) + \lambda_{r \rightarrow p}^*(y_r; x) = \theta_r(y_r; x; w, \lambda^*)$ with respect to $p \in P(r)$, thus we are able to obtain

$$(1 + |P(r)|) \sum_{p \in P(r)} \lambda_{r \rightarrow p}^*(y_r; x) = |P(r)|(\theta_r(y_r; x; w) + \sum_{c \in C(r)} \lambda_{c \rightarrow r}(y_c; x)) - \sum_{p \in P(r)} \mu_{p \rightarrow r}(y_r; x)$$

Plugging it into the above equation results in the desired block dual ascent update rule. ■

One can verify that since the program is not strictly concave in λ , the optimal solutions can be achieved for every additive shift of $\lambda_{r \rightarrow p}(y_r; x)$. The above lemma describes an analytic solution for the optimal $\lambda_{r \rightarrow p}(y_r; x)$, that are computed in the block coordinate steps of the algorithm. In practice, block coordinate descent with analytic steps provides a significant speedup over conventional gradient methods and can be parallelized and distributed easily, as shown by Schwing et al. (2011).

A learning step updates the weights w so as to maximize the log-beliefs. When using blended inference, the beliefs are not required to agree on their marginal probabilities. However, they are governed by the concave program in Theorem 1. Its concavity guarantees that these beliefs agree on their marginals at the optimum.

Lemma 3 Blended learning: *Consider the program given in Theorem 1. The gradient of its objective function with respect to w_k takes the form:*

$$\sum_{(x, y) \in \mathcal{S}} \sum_{r \in \mathcal{R}_k} \sum_{\hat{y}_r \in \mathcal{Y}_r} b_r(\hat{y}_r|x; w, \lambda) \phi_{k,r}(x, \hat{y}_r) - \phi_{k,r}(x, y_r) + C w_k.$$

Proof We let

$$\log b_r(y_r|x; w, \lambda) = \theta_r(y_r; x; w, \lambda) - \log \left(\sum_{\hat{y}_r} \exp(\theta_r(\hat{y}_r; x; w, \lambda)) \right).$$

The theorem follows by noting that $\theta_r(y_r; x; w) = \sum_{k \in \mathcal{K}_r} w_k \phi_{k,r}(x, y_r)$, recalling the definition of $\theta_r(y_r; x; w, \lambda)$ in Equation (9) and that $b_r(\hat{y}_r|x; w, \lambda)$ which is defined in Theorem 1 is the gradient of its log-partition function, i.e., $\log \left(\sum_{\hat{y}_r} \exp(\theta_r(\hat{y}_r; x; w, \lambda)) \right)$. ■

The computational complexity of the gradient depends on the structure of the features, especially the number of regions and their labels. Our framework prefers features with small regions and reasonable number of labels. Another computational issue relates to the step size η for increasing the objective along the gradient of w_k . In general, the gradient updates verify that the chosen step size η reduces the objective. Theoretically, we can use the fact that the gradient is Lipschitz continuous to predetermine a step size that guarantees ascent. In practice it gives worse performance than searching for a step size depending on the gradient and the objective at any given point.

Lemmas 2 and 3 describe the inference and learning steps for maximizing the log-beliefs maximization in Theorem 1. Since the program is concave, the order of the maximization steps is not important, and as long as all inference and learning parameters are optimized the maximal value is attained. For example, one can maximize the inference variables λ till

Blending learning and inference

1. Set $\theta_r(\hat{y}_r; x, w) = \sum_{k \in K_r} w_k \phi_{k,r}(x, \hat{y}_r)$. Repeat until convergence:
2. For every $(x, y) \in \mathcal{S}$, $r \in \mathcal{R}$, $\hat{y}_r \in \mathcal{Y}_r$, $p \in P(r)$:

$$H_{p \rightarrow r}(\hat{y}_r; x) = \log \left(\sum_{\hat{y}_p \in \mathcal{Y}_p} \exp(\theta_r(\hat{y}_p; x, w)) + \sum_{c \in C(p)} \lambda_{c \rightarrow p}(\hat{y}_c; x) - \sum_{p' \in P(p)} \lambda_{p \rightarrow p'}(\hat{y}_p; x) \right)$$

$$\lambda_{r \rightarrow p}(\hat{y}_r; x) = \frac{\theta_r(\hat{y}_r; x, w) + \sum_{c \in C(r)} \lambda_{c \rightarrow r}(\hat{y}_c; x) + \sum_{p' \in P(r)} H_{p' \rightarrow r}(\hat{y}_p; x)}{1 + |P(r)|} - |H_{p \rightarrow r}(\hat{y}_r; x)|$$
3. Set $b_r(\hat{y}_r|x; w, \lambda) \propto \exp(\theta_r(\hat{y}_r; x, w) + \sum_{c \in C(r)} \lambda_{c \rightarrow r}(\hat{y}_c; x) - \sum_{p \in P(r)} \lambda_{r \rightarrow p}(\hat{y}_r; x))$.

$$w_k \leftarrow w_k - \eta \left(\sum_{(x,y) \in \mathcal{S}} \sum_{r \in \mathcal{R}_k} \sum_{\hat{y}_r \in \mathcal{Y}_r} \left(\sum_{p \in P(r)} b_r(\hat{y}_p|x; w, \lambda) \phi_{k,r}(x, \hat{y}_p) - \phi_{k,r}(x, \hat{y}_r) \right) + C w_k \right).$$

Figure 2: The inference step is described in Lemma 2 and the learning step is described in Lemma 3. The step size η is set to guarantee convergence (e.g., corresponding to the Lipschitz constant or the Armijo rule.) Concavity of the program in Theorem 1 ensures that the blending converges to consistent inferred beliefs, see Theorem 4.

they do not change before optimizing the learning parameters w . We refer to this approach in Figure 1 as nested inference within learning since it performs the approximate inference heuristic described in Section 3 as a black-box solver. Nested learning and inference is computationally unfavorable in general as it requires to infer λ till convergence for every gradient step for learning w . Since concavity ensures that the maximization does not depend on the order of the maximizing steps, it also provides a principled way to blend the learning and inference steps. Particularly, it may learn the w parameters using inferred beliefs $b_r(\hat{y}_r|x; w, \lambda) \propto \exp(\theta_r(\hat{y}_r; x, w, \lambda))$ that do not agree on their marginal probabilities. For this purpose our algorithm infers the parametrized beliefs *differently* than the (outer) beliefs that are computed by the nested learning algorithm in Figure 1. This blending property is important in practice, since shortly upon initialization, where the given parameters w are far from the optimum, it is not advisable to spend time on computing consistent beliefs. Figure 2 summarizes the inference-learning blending algorithm.

The block coordinate descent algorithm is guaranteed to converge, as it monotonically increases the log-beliefs in Theorem 1, which are upper bounded by its dual. Moreover, the values that are generated by the algorithm converge to the program's optimal value (see Theorem 4 for exact statement). It is not immediately clear that the algorithm converges to its optimal value since the program is not strictly concave. Consequently, the sequence of variables $\lambda_{r \rightarrow p}(\hat{y}_r; x)$ generated by the algorithm is not guaranteed to be bounded. As a trivial example, adding an arbitrary constant to the variables, $\lambda_{r \rightarrow p}(\hat{y}_r; x) + c$, does not

change the objective value, hence the algorithm can generate a monotonically decreasing unbounded sequences. Convergence to the optimum holds by convex duality:

Theorem 4 *The learning-inference blending algorithm in Figure 2 for log-beliefs maximization is guaranteed to converge. Moreover, the value of its objective is guaranteed to converge to the global maximum, and its sequence of beliefs are guaranteed to converge to consistent beliefs that are the unique solution of the dual program*

$$\min_{z_{b_r}} \sum_{(x,y) \in \mathcal{S}} \frac{1}{2C} \|z\|^2 - \sum_{r \in \mathcal{R}} H(b_r)$$

$$\text{subject to } b_r(\hat{y}_r|x) \geq 0, \quad \sum_{p \in P(r)} b_r(\hat{y}_p|x) = 1, \quad b_r(\hat{y}_r|x) = \sum_{\hat{y}_p \in \mathcal{Y}_p} b_p(\hat{y}_p|x)$$

$$z_k = \sum_{(x,y) \in \mathcal{S}} \sum_{r \in \mathcal{R}_k} \left(\sum_{\hat{y}_p} b_r(\hat{y}_p|x) \phi_{k,r}(x, \hat{y}_p) - \phi_{k,r}(x, \hat{y}_r) \right)$$

Proof The update rules in Figure 2 iteratively apply the block coordinate ascent rules in Lemmas 2 and 3 thus monotonically increase the primal objective in Theorem 1. This program is concave thus it is bounded by its dual program, therefore the value of its objective is guaranteed to converge. To derive the dual program we construct the Lagrangian

$$L(w, \lambda, b) = -\frac{C}{2} \|w\|^2 + \sum_k w_k \left(\sum_{(x,y) \in \mathcal{S}} \sum_{r \in \mathcal{R}_k} \phi_{k,r}(x, \hat{y}_r) \right) - \sum_{(x,y) \in \mathcal{S}} \log \left(\sum_{\hat{y}_p} \exp(\theta_r(\hat{y}_p; x)) + \sum_{c \in C(r)} \lambda_{c \rightarrow r}(\hat{y}_c; x) - \sum_{p \in P(r)} \lambda_{r \rightarrow p}(\hat{y}_r; x) \right) + \sum_{(x,y) \in \mathcal{S}} \sum_{\hat{y}_p} b_r(\hat{y}_p|x) \left(\theta_r(\hat{y}_p; x) - \sum_k w_k \phi_{k,r}(x, \hat{y}_p) \right).$$

The variables $b_r(\hat{y}_r|x)$ are the Lagrange multipliers for the equality constraints $\theta_r(\hat{y}_r; x) = \sum_k w_k \phi_{k,r}(x, \hat{y}_r)$. The dual program takes the form $q(b) = \max_{w, \lambda} L(w, \lambda, b)$. Setting z_k as above, since the $\|\cdot\|^2$ is the conjugate dual of $\|\cdot\|^2$, the maximization over w takes the form $\max_w \left\{ -\frac{C}{2} \|w\|^2 - \sum_k w_k z_k \right\} = \frac{1}{2C} \|z\|^2$. To complete the derivation of the dual, the maximization over λ , for every (x, y) takes the form $\max_{\lambda} \left\{ \sum_{r, \hat{y}_p} b_r(\hat{y}_p|x) (\theta_r(\hat{y}_p; x) - \sum_{p' \in P(r)} \lambda_{c \rightarrow r}(\hat{y}_c; x) + \sum_{p \in P(r)} \lambda_{r \rightarrow p}(\hat{y}_r; x)) \right\}$. This is the conjugate dual of the log-partition function, which is known to be the entropy function $H(b_r)$. The mixing of the messages between the different regions results in the marginalization constraints in the dual program. An alternative proof for the conjugate duality between the re-parametrized log-partitions and entropies subject to marginalization constraints appears in the more generalized setting of Theorem 5.

Finally, since the dual is strictly convex subject to linear marginalization constraints and the linear moment constraints the convergence properties are a consequence of Tseng and Bertsekas (1987). ■

The convergence of the block coordinate ascent depends on the step size η , which requires to increase the log-beliefs. This can be done by the Armijo rule, or by using the fact that

the function z_k^2 is strongly convex (e.g., Tseng and Bertsekas (1987)) and its gradient is Lipschitz continuous (e.g., Nesterov (2004)). In practice, Theorem 4 describes how to measure the convergence of the algorithm. Specifically, it may be derived from the primal objective value, the dual objective value or the beliefs themselves, as all these quantities converge. Unfortunately, the variables λ might not converge, but if they do converge, their convergence point is optimal.

5. Blending learning and loss adjusted inference

Loss adjusted inference emerges from Support Vector Machines (SVMs) where a loss function indicates preferences between different labels. In the following we consider nonnegative loss functions $\ell_r(\hat{y}_r, \hat{y}_r) \geq 0$ over subsets of variables, while $\ell_r(\hat{y}_r, \hat{y}_r) = 0$. We suggest to augment our learned beliefs with loss adjusted probability models. Given a training example (x, y) , we define the loss adjusted belief model to be

$$b_r(\hat{y}_r|x; w, \lambda) \propto \exp\left(\ell_r(\hat{y}_r, \hat{y}_r) + \theta_r(\hat{y}_r; x, w, \lambda)\right).$$

Note that these beliefs should be conditioned over x as well as the vector $\ell_r(\cdot, \hat{y}_r)$. However, to simplify the notation, we leave this conditioning implicit. The intuition for using the loss as a prior and for deriving loss adjusted beliefs is based on encouraging to learn parameters that decrease probabilities over labels with higher loss with respect to the training labels. The likelihood approach aims at maximizing the beliefs $b_r(\hat{y}_r|x; w, \lambda)$. Since the beliefs are probability distributions, it equivalently aims at minimizing the log-beliefs of all other assignments, namely, $b_r(\hat{y}_r|x; w, \lambda)$ for every $\hat{y}_r \neq y_r$. Since the loss $\ell_r(\hat{y}_r, \hat{y}_r)$ is a nonnegative function it implies that loss-adjusted maximum-likelihood learns parameters that better reduce scores of non-observed training labels, namely $\theta_r(\hat{y}_r; x, w, \lambda)$ for any $\hat{y}_r \neq y_r$. The algorithms for maximizing loss adjusted beliefs follow the derivations presented in Section 4, while replacing $\theta_r(\hat{y}_r; x, w, \lambda)$ with $\theta_r(\hat{y}_r; x, w, \lambda) + \ell_r(\hat{y}_r, \hat{y}_r)$. Letting $z = (x, y)$ we introduce $\theta_r(\hat{y}_r; z, w, \lambda) = \theta_r(\hat{y}_r; x, w, \lambda) + \ell_r(\hat{y}_r, \hat{y}_r)$.

The norm-product approach for inference may use counting numbers c_r to control the peakedness of the beliefs, namely

$$b_r(\hat{y}_r|x; w, \lambda, c) \propto \exp\left(\frac{\theta_r(\hat{y}_r; x, w, \lambda) + \ell_r(\hat{y}_r, \hat{y}_r)}{c_r}\right). \quad (10)$$

As $c_r \rightarrow 0$ this distribution approaches a zero-one probability around the most likely structure, i.e., the desired loss adjusted prediction. Thus the following program blends learning with loss adjusted inference, as well as structured predictions (cf. Meshi et al. (2010)):

Theorem 5 Consider the loss adjusted beliefs given in Equation (10) and their maximum likelihood concave program:

$$\max_{w, \lambda} \sum_{z \in \mathcal{S}} \sum_{r \in \mathcal{R}} c_r \cdot \log b_r(\hat{y}_r|x; w, \lambda, c) - \frac{C}{2} \|w\|^2.$$

Set $\hat{\theta}_r(\hat{y}_r; z, w) = \theta_r(\hat{y}_r; x, w) + \ell_r(\hat{y}_r, \hat{y}_r)$. Then, blending the following loss adjusted learning and inference update rules is guaranteed to converge to the programs optimal value for any

$c_r > 0$.

$$\mu_{p \rightarrow r}(\hat{y}_r; x) = c_p \log \left(\sum_{\hat{y}_p, \hat{y}_r} \exp \left((\hat{\theta}_p(\hat{y}_p; z, w) + \sum_{c \in \mathcal{C}(p)} \lambda_{c \rightarrow p}(y_c; x) - \sum_{p' \in \mathcal{P}(p)} \lambda_{p \rightarrow p'}(\hat{y}_{p'}, x)) / c_p \right) \right)$$

$$\lambda_{r \rightarrow p}(\hat{y}_r; x) = \frac{c_r \left(\hat{\theta}_r(\hat{y}_r; z, w) + \sum_{c \in \mathcal{C}(r)} \lambda_{c \rightarrow r}(y_c; x) + \sum_{p' \in \mathcal{P}(r)} \mu_{p' \rightarrow r}(\hat{y}_{p'}, x) \right)}{c_r + \sum_{p \in \mathcal{P}(r)} c_p} - \mu_{p \rightarrow r}(\hat{y}_r; x)$$

$$w_k \leftarrow w_k - \eta \left(\sum_{(x, y) \in \mathcal{S}} \sum_{r \in \mathcal{R}_k} \left(\sum_{\hat{y}_r \in \mathcal{Y}_r} b_r(\hat{y}_r|x; w, \lambda, c) \phi_{k,r}(x, \hat{y}_r) - \phi_{k,r}(x, y_r) \right) + C w_k \right).$$

Moreover, the beliefs converge to consistent beliefs that are the unique solution of the dual program

$$\max_{b_r, u} \sum_{(x, y) \in \mathcal{S}} \left(c_r H(b_r) + \sum_{\hat{y}_r} b_r(\hat{y}_r|x) \ell_r(\hat{y}_r, \hat{y}_r) \right) - \frac{1}{2C} \|u\|^2,$$

subject to $u_k = \sum_{(x, y) \in \mathcal{S}} \sum_{r \in \mathcal{R}_k} b_r(\hat{y}_r|x) \phi_{k,r}(x, \hat{y}_r) - \phi_{k,r}(x, y_r)$, $b_r(\hat{y}_r|x) \geq 0$, $\sum_{\hat{y}_r} b_r(\hat{y}_r|x) = 1$ and $b_r(\hat{y}_r|x) = \sum_{\hat{y}_p \setminus \hat{y}_r} b_p(\hat{y}_p|x)$.

Proof The update rule for w follows from Lemma 3 and the update rule for λ follows from Lemma 2. Since the program is unconstrained, the optimal λ is attained when the gradient vanishes, or equivalently $\sum_{\hat{y}_p, \hat{y}_r} b_p(\hat{y}_p|x; w, \lambda) = b_r(\hat{y}_r|x; w, \lambda)$, while $b_r(\cdot)$ are the reparametrized beliefs. When setting $\mu_{p \rightarrow r}(\hat{y}_r; x)$ as above, the marginalization of $b_p(\hat{y}_p|x; w, \lambda)$ satisfy $\sum_{\hat{y}_p \setminus \hat{y}_r} b_p(\hat{y}_p|x; w, \lambda) \propto \exp\left(\left(\mu_{p \rightarrow r}(\hat{y}_r; x) + \lambda_{r \rightarrow p}(\hat{y}_r; x)\right) / c_p\right)$. Therefore, by taking the logarithm, the gradient vanishes whenever the beliefs numerators agree

$$\frac{\mu_{p \rightarrow r}(\hat{y}_r; x) + \lambda_{r \rightarrow p}(\hat{y}_r; x)}{c_p} = \frac{\theta_r(\hat{y}_r; z, w) + \sum_{c \in \mathcal{C}(r)} \lambda_{c \rightarrow r}(y_c; x) - \sum_{p \in \mathcal{P}(r)} \lambda_{r \rightarrow p}(\hat{y}_r; x)}{c_r}$$

Multiplying both sides by $c_r c_p$ and summing both sides with respect to $p' \in \mathcal{P}(r)$ we are able to isolate $\sum_{p' \in \mathcal{P}(r)} \lambda_{r \rightarrow p'}(\hat{y}_r; x)$. Plugging it into the above equation results in the desired inference update rule, i.e., $\lambda_{r \rightarrow p}(\hat{y}_r; x)$ for which the partial derivatives vanish.

To prove the duality theorem we show that the primal program is the dual of its dual program using its Lagrangian. For every $r, \hat{y}_r, p \in \mathcal{P}(r)$ we introduce the Lagrange multipliers $\lambda_{r \rightarrow p}(\hat{y}_r; x)$ for the marginalization constraints $b_r(\hat{y}_r|x) = \sum_{\hat{y}_p \setminus \hat{y}_r} b_p(\hat{y}_p|x)$. We also introduce the Lagrange multipliers u_k for the constraints $u_k = \sum_{(x, y) \in \mathcal{S}} \sum_{r \in \mathcal{R}_k} b_r(\hat{y}_r|x) \phi_{k,r}(x, \hat{y}_r) - \phi_{k,r}(x, y_r)$. We let Δ_r refer to the probability simplex constraining the beliefs $b_r(\hat{y}_r|x)$. Then the primal program takes the form $p(\lambda, w) = \max_{b_r \in \Delta_r} L(b, u, \lambda, w)$ which decomposes to $\sum_{z, r} c_r \max_{b_r \in \Delta_r} \{H(b_r) + \sum_{\hat{y}_r} b_r(\hat{y}_r|x) (\ell_r(\hat{y}_r, \hat{y}_r) + \theta_r(\hat{y}_r; x, w, \lambda)) / c_r\} + w^\top d + \max_{u_k} \{w^\top u - \frac{C}{2} \|u\|^2\}$, where d_k are the empirical moments $d_k = \sum_{(x, y) \in \mathcal{S}} \phi_{k,r}(x, y_r)$. Thus the primal is the sum of conjugate dual functions. The primal is then derived since the log-partition function is the conjugate dual of the entropy function and the conjugate dual of $\frac{C}{2} \|u\|^2$ is $\frac{C}{2} \|w\|^2$. The form in the theorem is obtained since $\ell_r(\hat{y}_r, \hat{y}_r) = 0$ and $\sum_r \sum_{p \in \mathcal{P}(r)} \lambda_{r \rightarrow p}(\hat{y}_r; x) - \sum_r \sum_{c \in \mathcal{C}(r)} \lambda_{c \rightarrow r}(y_c; x) \equiv 0$. Finally, the convergence properties are a consequence of Tseng and Bertsekas (1987) similarly to Theorem 4. ■

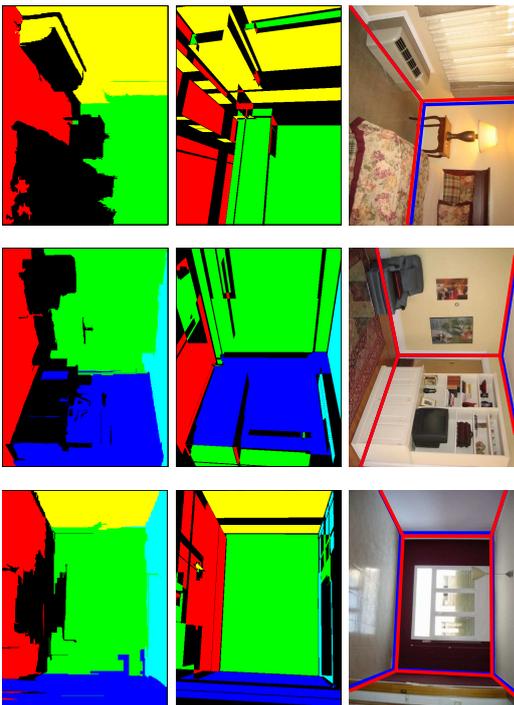


Figure 3: The top row shows three indoor scenes from the layout dataset (Hedau et al., 2009). Orientation maps (Lee et al., 2009) and geometric context (Holm et al., 2007) features for the respective images are illustrated in the center and bottom row respectively.

Note that the log-beliefs are balanced by c_r , to compensate for their exponent peakedness. This balancing makes the objective well defined at $c_r = 0$ as the limit of $c_r \rightarrow 0$. In this case the program is non-smooth and one needs to consider the sub gradient in the form of max-beliefs. However, this setting was already developed, although in the different context of structured-SVMs using dual losses, and we refer the interested reader to Meshi et al. (2010) for more details.

6. Experiments

The effectiveness of the discussed framework was recently illustrated by employing this algorithm as the learning engine for various computer vision tasks: scene understanding (Yao et al. (2012)), Lin et al. (2013)), shape reconstruction (Salzman and Urtasun (2012)) indoor scene understanding (Schwing et al. (2012a); Schwing and Urtasun (2012)), depth estimation (Yamaguchi et al. (2012)), flow estimation (Yamaguchi et al. (2013)) and visual-language understanding (Fidler et al. (2013)). The code is publicly available on <http://www.alexander-schwing.de/projectGeneralStructuredPredictionLatentVariables.php>.

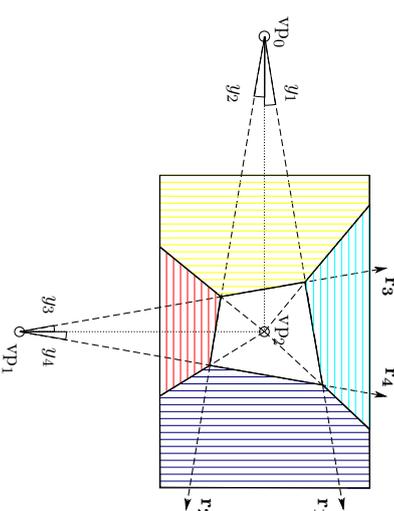


Figure 4: Ray parameterization of the layout given detected vanishing points. A state of a discretized variable y_i gives rise to a particular ray r_i originating from a detected vanishing point. Four rays are sufficient to define a room layout, i.e., the location of the walls.

In the following we demonstrate the various properties of blending learning and inference. For this purpose we elaborate more carefully on a 3D scene understanding application (Schwing et al., 2012a) evaluated on the well known layout dataset (Hedau et al., 2009) containing 314 indoor images. Given a single image similar to the ones illustrated in the top row of Figure 3, we aim at estimating the location of the left, front and right walls as well as ceiling and floor. Formulating this task as a pixelwise semantic segmentation application permits a large number of configurations that are physically not plausible. We therefore constrain the space of configurations by parametrizing the application using three dominant vanishing points as illustrated in Figure 4. The random variables y_1, \dots, y_4 correspond to discretized angles of rays, each one originating from a detected vanishing point. Such a parametrization in terms of rays permits only physically plausible configurations and assumes the world to be Manhattan like, i.e., the observed room is aligned according to the three dominant directions.

To obtain the dominant directions we employ the vanishing point detector of Hedau et al. (2009). Due to the involved randomness it failed in our case on 9 training images and was successful on all 105 test instances. For image x and a given layout hypothesis $y = (y_1, \dots, y_4)$ we compute a 55-dimensional feature vector $\phi(x; y)$ which is constructed based on geometric context (Holm et al., 2007) and orientation maps (Lee et al., 2009) illustrated in the middle and bottom row of Figure 3 respectively.

More specifically, for each of the five hypothesized wall areas (note that the back wall is never observed) obtained from projecting the predicted layout y into the image, we count how frequently geometric context estimates the five different wall labels plus an additional clutter label within each hypothesized wall. This gives rise to a $5 \cdot 6 = 30$ dimensional

$C \backslash c_i$	1e2	1	1e-2	1e-4	1e-6
10.0000	30.92	17.53	13.81	14.24	14.46
1.0000	23.95	16.26	14.46	14.86	14.86
0.1000	17.64	13.69	14.83	14.80	14.80
0.0100	15.83	13.59	14.20	14.25	14.25
0.0010	15.46	13.82	14.00	13.85	13.85
0.0001	16.04	14.09	13.95	13.96	13.97
0	15.72	13.70	13.82	13.91	13.98

$C \backslash c_i$	1e2	1	1e-2	1e-4	1e-6
10.0000	27.28	19.54	17.43	16.46	16.49
1.0000	22.66	17.87	16.55	16.83	16.83
0.1000	19.41	17.43	16.74	16.91	16.96
0.0100	17.98	16.48	17.04	16.86	16.86
0.0010	17.92	17.18	16.95	16.83	16.87
0.0001	17.95	17.06	17.07	16.80	16.71
0	18.01	16.97	17.25	17.04	17.01

Figure 5: Left: Test set percentage pixel error on the layout data set Hedau et al. (2009). Right: Test set pixel classification error in % on the bedroom data set of Hedau et al. (2010).

feature vector. Similarly for orientation map evidence we count how many of the five possible labels fell onto each of the five hypothesized wall areas, resulting in a $5 \cdot 5 = 25$ dimensional vector. Combining both image evidences we hence obtain a feature vector ϕ having a total of $K = 55$ entries, each being represented by a graphical model having nodes $\mathcal{R}_k, k \in \{1, \dots, K\}$.

Note that a vanilla implementation of these color counting features results in potentials of order four for the front wall, and of order three for all other walls. High-order potentials increase the complexity of learning and inference, therefore, the exact decomposition of these high-order potentials into pairwise terms using ‘integral geometry’ Schwing et al. (2012a) is important for tractability.

In our experiments we learn and infer with the same counting numbers. For example, when we learn log-beliefs by setting the counting numbers to 1, we also infer with log-beliefs at test time, namely setting the counting numbers to 1. Figure 5 demonstrates the tradeoffs of learning with various counting numbers. This experiment also compares to blending of structured-SVMs of Meshi et al. (2010) when setting the counting numbers to 0.

In Figure 6 we illustrate the optimized cost function, i.e., the surrogate training loss over wall clock time. We observe that our discussed blended learning approach (‘Ours’) converges quickly, i.e., in less than 50 seconds, to a zero primal dual gap. In contrast it takes the standard learning approach (‘Standard 20’), which performs 20 message passing iterations, more than 600 seconds to converge to the same solution. Hence blended learning is more than 10 times faster on this example. Next we investigate whether a standard message passing approach with 20 iterations is overly pessimistic for this dataset. To this end we use 10 message passing iterations and refer to the obtained result as ‘Standard 10’. Investigating Figure 6 more carefully we observe that 10 message passing iterations are not sufficient to close the duality gap, i.e., the approach does never converge to the same solution.

These plots validate our theoretical results. However, often we are interested in the resulting test set error. Therefore we compare learning which uses the proposed blending approach with the standard technique using the test set performance in Figure 7. More specifically, in Figure 7(a) we illustrate the test set performance of blending with all counting numbers equal to one (‘Blending (1)’) and compare it to the test set performance of blending with all counting numbers equal to zero (‘Blending (0)’) derived by Meshi et al.

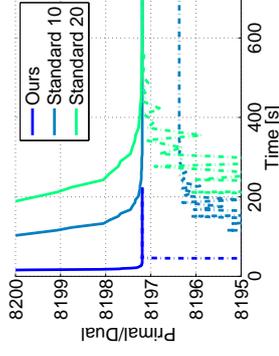


Figure 6: Primal surrogate loss (dashed) and its dual (solid) over wall clock time measured in seconds for blended learning, the standard approach with 10 iterations, and the standard approach with 20 iterations.

(2010), the test set performance of convex learning with 20 message passing iterations with both counting numbers equal to zero (cf. Meshi et al. (2010)) and one, i.e., ‘Standard 20 (0)’ and ‘Standard 20 (1)’ respectively. We observe that the test set accuracy drops significantly faster which is expected since we are able to update the parameter vector more frequently. Similar to the aforementioned surrogate training loss result we observe a performance improvement of more than one order of magnitude, i.e., we obtain accurate test set results more than 10 times faster.

In a next experiment we evaluate the importance of the loss function. The results are illustrated in Figure 7(b). We observe the loss to be important for the case where the counting numbers equal zero. We are able to obtain a performance similar to loss-included setting if the counting numbers are equal to one. However algorithms which do not use a loss function while having counting numbers equal to zero got stuck prematurely and are hence not even visible in Figure 7(b), where the scale was adjusted to fit the other plots.

Next we observe the behavior if we reduce the number of iterations for standard learning from 20 down to 2. Recall, from the surrogate training loss results that the primal-dual gap will not reach zero in this setting. The test set errors are illustrated in Figure 7(c). We observe that the standard method is approaching the blending technique. This indicates that a primal-dual gap is not necessarily important for good generalization performance.

In Figure 7(d) we illustrate that blending offers a wide range of possibilities. Indeed, we are not restricted to performing only a single message passing iteration before updating the parameter vector. Rather any arbitrary scheduling is possible. As illustrated in the results we observe slightly faster convergence when updating the messages twice as frequently as the parameter vector, i.e., we perform two rounds of message passing updates before updating the parameters of the model. This is expected since the distribution is more accurate while message passing updates are quick in this setup. We refer the interested reader to (Schwing et al., 2012a) for additional results exploring counting numbers other than zero and one.

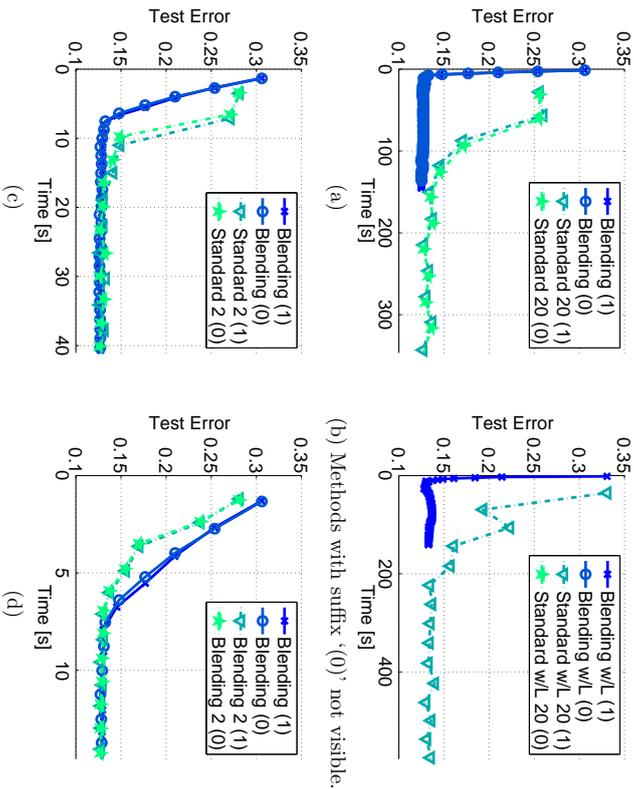


Figure 7: A comparison of the test set error over training time for different parameter settings. In (a) we compare standard learning with blended learning when including a loss function. In (b) we provide the comparison when not using a loss function. Note that using MAP estimation without loss got stuck at a high error. The results in (c) demonstrate the effects when reducing the iterations of the standard learning method, while (d) illustrates the flexibility offered by blending where we use two message passing iterations before updating the parameters.

7. Conclusion and Discussion

In this paper we describe the objective function for nested learning and inference in CRFs, for which approximate inference algorithms are used as a sub-procedure. The devised objective maximizes the log-likelihoods — probability distributions over subsets of variables that agree on their marginal probabilities. This objective is concave and consists of two types of variables that are related to the learning and inference tasks respectively. Therefore, we are able to blend the learning and inference procedures and effectively get to the optimum much faster than nested learning and inference approach, which uses inference algorithms as black-box solvers. We show the computational advantage for using blended algorithms over nested ones. We also provide an efficient C++ implementation with a Matlab wrapper.

This work extends Hazan and Urtasun (2010) while simplifying its theoretical and practical concepts. Specifically, we introduce the learning program as maximizing log-likelihoods, explain the relations between nested and blended learning-inference and derives a blending algorithm for general graphs. This work is extended to latent variables (Schwing et al. (2012b)) and deep learning (Chen* et al. (2015); Schwing and Urtasun (2015)). The effectiveness of the discussed framework was recently illustrated by employing this algorithm as the learning engine for various computer vision tasks: 2D scene understanding (Yao et al. (2012)), 3D scene understanding (Lin et al. (2013)), shape reconstruction (Salzman and Urtasun (2012)) indoor scene understanding (Schwing et al. (2012a); Schwing and Urtasun (2012)), depth estimation (Yamaguchi et al. (2012)), flow estimation (Yamaguchi et al. (2013)) and visual-language understanding (Fidler et al. (2013)). We believe it is interesting to show in the future if this algorithm provides state-of-the-art performance in domains other than computer vision, or whether the statistics in computer vision are used by this approach in a special manner.

The computational complexity of our algorithm depends on norm-products over the labels of regions. Therefore, efficient techniques over large regions in inference can be applied as sub-procedure in our algorithm, e.g., Kohli et al. (2009); Batra et al. (2010); Tatlow et al. (2010, 2011); Tatlow and Zemel (2012).

In our framework, we can enforce the moment matching constraints through general concave functions. These function translate to a regularization in the primal. For computational efficiency we choose the square function but we did not investigate the different moment matching and regularization functions. Moreover, we enforce the marginalization constraints through indicator functions, in order to obtain closed-form solution in the primal block coordinate descent. However, we have shown that using the penalty method we can enforce the marginalization constraints with different convex functions. Further explorations of structured prediction with squared penalties appears in work by Mochi et al. (2015). We leave the affect of general convex functions on moment matching and regularization, as well as marginalization constraints and efficient message-passing to future research.

Interestingly, our approach confirms that the parameters of graph based structured predictors can be efficiently learned in many real-life problems. This validates the intuition behind the theoretical results of Wainwright et al. (2003); Wainwright (2006) which asserts that whenever learning and inference occur together one can use pseudo moment matching for learning the parameters. This concept was put forward in the general framework of learning to reason by Khadon and Roth (1997) and we leave for future research to find different frameworks which have similar learning-prediction robustness.

8. Extensions: regularizations and the penalty method

Duality theory turned out to be very effective in machine learning as it provides a principled way to decompose the different ingredients of the primal objective through its Lagrange multipliers. The dual decomposition in turn provides the means to efficiently estimate the different ingredients while maintaining their consistency using the dual objective.

When dealing with convex programs one usually needs to consider the set of primal feasible solutions while constructing the dual function. We find it simpler to describe

the primal program using extended real-valued convex functions, which are functions that can take on the value of infinity. By using extended real-valued functions we can ignore their domains, i.e., points for which a function attains the value other than infinity, thus simplifying the derivations. The dual programs of extended real valued convex functions $g(\mu)$ are conveniently formulated in terms of their conjugate dual

$$g^*(z) = \max_{\mu} \left\{ \mu^{\top} z - g(\mu) \right\}.$$

Throughout this work we use the following duality theorem, known as the Fenchel duality (cf. Fenchel (1951); Rockafellar (1970); Bertsekas et al. (2003)):

Theorem 6 *Let $f(\cdot), h_1(\cdot), h_2(\cdot)$ be extended real-valued, continuous and convex functions.*

The following are primal and dual programs:

$$\begin{aligned} \min_{\lambda, w} \quad & \sum_{(x,y) \in S_T \times \mathcal{R}} f(\theta_r(\cdot; x, w, \lambda) + \ell_r(\cdot, \hat{y}_r)) + \sum_k h_1(w_k) + \sum_{(x,y), r, \hat{y}_r, p \in P(r)} h_2(\lambda_{r \rightarrow p}(\hat{y}_r; x)) \\ \max_{b_r} \quad & \sum_{(x,y) \in S_T \times \mathcal{R}} \left(-f^*(b_r) + \sum_{\hat{y}_r} b_r(\hat{y}_r | x) \ell_r(\hat{y}_r, \hat{y}_r) \right) \\ & - \sum_k h_1^* \left(\sum_{(x,y), r} \left(\sum_{\hat{y}_r} b_r(\hat{y}_r) \phi_{k,r}(x, \hat{y}_r) - \phi_{k,r}(x, y_r) \right) \right) \\ & - \sum_{(x,y), r, \hat{y}_r, p} h_2^* \left(\sum_{\hat{y}_p, \hat{y}_r} b_p(\hat{y}_p | x) - b_r(\hat{y}_r | x) \right) \end{aligned}$$

Proof The proof goes along the lines of Theorem 5. The dual program takes the form

$$\begin{aligned} \sum_{(x,y) \in S_T \times \mathcal{R}} \left(-f^*(b_r) + \sum_{\hat{y}_r} b_r(\hat{y}_r | x) \ell_r(\hat{y}_r, \hat{y}_r) \right) - \sum_k h_1^*(u_k) - \sum_{(x,y), r, \hat{y}_r, p} h_2^*(\delta_{r \rightarrow p}(\hat{y}_r; x)) \\ \text{s.t.} \quad u_k = \sum_{(x,y), r} \left(\sum_{\hat{y}_r} b_r(\hat{y}_r | x) \phi_{k,r}(x, \hat{y}_r) - \phi_{k,r}(x, y_r) \right) \\ \delta_{r \rightarrow p}(\hat{y}_r; x) = \sum_{\hat{y}_p, \hat{y}_r} b_p(\hat{y}_p | x) - b_r(\hat{y}_r | x) \end{aligned}$$

Constructing its Lagrangian with the Lagrange multipliers $w_k, \lambda_{r \rightarrow p}(\hat{y}_r; x)$, we obtain the primal program $p(w, \lambda) = \max_{b_r, u, \lambda} L(b_r, \lambda, w)$ which decomposes to $\sum_{(x,y), r} \max_{b_r} \{ -f^*(b_r) + \sum_{\hat{y}_r} b_r(\hat{y}_r | x) (\ell_r(\hat{y}_r, \hat{y}_r) - \theta_r(\hat{y}_r; x, w, \lambda)) \} + \sum_{(x,y), r, \hat{y}_r, p} \max_{\delta_{r \rightarrow p}} \{ \delta_{r \rightarrow p}(\hat{y}_r) - h_2^*(\delta_{r \rightarrow p}(\hat{y}_r)) \} + w^{\top} e + \sum_k \max_{u_k} \{ u_k u_k - h_1^*(u_k) \}$, where $e_k = \sum_{(x,y), r} \phi_{k,r}(x, y_r)$. The result then follows since the conjugate dual of $f^*(\cdot), h_1^*(\cdot), h_2^*(\cdot)$ are $f(\cdot), h_1(\cdot), h_2(\cdot)$ respectively. ■

The above formulation describes a dual program with a selection rule $f^*(\cdot)$, a penalty function for learning moment matchings $h_1^*(\cdot)$ and a penalty function $h_2^*(\cdot)$ for fitting the marginalization constraints. Although these penalty functions are conceptually different — $h_1^*(\cdot)$ relates to learning the parameters w and $h_2^*(\cdot)$ relates to inferring the marginal probabilities — they have the same variational interpretation.

The primal program translates the dual penalty functions $h_1^*(\cdot), h_2^*(\cdot)$ to regularization functions $h_1(\cdot), h_2(\cdot)$. Whenever the primal functions are smooth we can use the chain rule to derive the primal program gradients:

$$\begin{aligned} \frac{\partial}{\partial w_k} & : \sum_{(x,y) \in S_T \times \mathcal{R}} \left(\sum_{r \in \mathcal{R}} \left(\sum_{\hat{y}_r} \frac{\partial f(\theta_r(\cdot; z, w, \lambda))}{\partial \theta_r(\hat{y}_r; z, w, \lambda)} \phi_{k,r}(x, \hat{y}_r) - \phi_{k,r}(x, y_r) \right) + \nabla h_k(w_k) \right) \\ \frac{\partial}{\partial \lambda_{r \rightarrow p}(\hat{y}_r; x)} & : \sum_{\hat{y}_p, \hat{y}_r} \left(\frac{\partial f(\theta_r(\cdot; z, w, \lambda))}{\partial \theta_p(\hat{y}_p; z, w, \lambda)} - \frac{\partial f(\theta_r(\cdot; z, w, \lambda))}{\partial \theta_r(\hat{y}_r; z, w, \lambda)} + \nabla h_2(\lambda_{r \rightarrow p}(\hat{y}_r; x)) \right) \end{aligned}$$

The above generalizes the learning-inference blending algorithm for general functions $f(\cdot)$ and regularizations $h_1(\cdot), h_2(\cdot)$. The power of setting $f(\cdot)$ to be the log-partition function is that its derivatives are beliefs therefore we obtain an intuitive probabilistic interpretation. The parameters derivatives result in moment matching constraints

$$\sum_{(x,y) \in S_T \times \mathcal{R}} \left(\sum_{\hat{y}_r} \left(\sum_{r \in \mathcal{R}} \frac{\partial f(\theta_r(\cdot; z, w, \lambda))}{\partial \theta_r(\hat{y}_r; z, w, \lambda)} \phi_{k,r}(x, \hat{y}_r) - \phi_{k,r}(x, y_r) \right) \right).$$

The re-parametrization derivatives with respect to the messages $\lambda_{r \rightarrow p}(\hat{y}_r; x)$ are then marginalization constraints

$$\sum_{\hat{y}_p, \hat{y}_r} \left(\frac{\partial f(\theta_p(\cdot; z, w, \lambda))}{\partial \theta_p(\hat{y}_p; z, w, \lambda)} - \frac{\partial f(\theta_r(\cdot; z, w, \lambda))}{\partial \theta_r(\hat{y}_r; z, w, \lambda)} \right)$$

Moreover, whenever $h_2(\cdot) \equiv 0$, we are able to derive closed-form update rules.

Acknowledgments

We are most grateful to David McAllester for helpful discussions, and the anonymous reviewers for their helpful comments. This work was supported in part by the German-Israeli Foundation grant I-1123-407.6-2013.

References

- S. M. Aji and R. J. McEliece. The generalized distributive law and free energy minimization. In *Allerton*, 2001.
- D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Y. Ng. Discriminative learning of markov random fields for segmentation of 3D range data. In *Proc. CVPR*, 2005.
- D. Batra, A. C. Gallagher, D. Parikh, and T. Chen. Beyond trees: MRF inference via outer-planar decomposition. In *Proc. CVPR*, 2010.
- D. P. Bertsekas, A. Nedić, and A. E. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, 2003.
- L.-C. Chen*, A. G. Schwing*, A. L. Yuille, and R. Urtasun. Learning Deep Structured Models. In *Proc. ICML*, 2015. * equal contribution.

- M. Collins. Discriminative training methods for hidden markov models: Theory and experiments with perception algorithms. In *Proc. ACL*, 2002.
- J. Domke. Parameter learning with truncated message-passing. In *Proc. CVPR*, 2011.
- P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 2010.
- W. Fenchel. *Conver cones, sets, and functions*. Princeton University, Department of Mathematics, 1951.
- S. Fidler, A. Sharma, and R. Urtasun. A sentence is worth a thousand pixels. In *Proc. CVPR*, 2013.
- T. Hazan and A. Shashua. Convergent message-passing algorithms for inference over general graphs with convex free energies. In *Proc. UAI*, 2008.
- T. Hazan and A. Shashua. Norm-product belief propagation: Primal-dual message-passing for approximate inference. *Information Theory*, 2010.
- T. Hazan and R. Urtasun. A Primal-Dual Message-Passing Algorithm for Approximated Large Scale Structured Prediction. In *Proc. NIPS*, 2010.
- Y. Hedau, D. Hoiem, and D. Forsyth. Recovering the Spatial Layout of Cluttered Rooms. In *Proc. ICCV*, 2009.
- Varsha Hedau, Derek Hoiem, and David Forsyth. Thinking Inside the Box: Using Appearance Models and Context Based on Room Geometry. In *Proc. ECCV*, 2010.
- T. Heskes. Convexity arguments for efficient minimization of the Bethe and Kikuchi free energies. *Journal of Artificial Intelligence Research*, 2006.
- D. Hoiem, A. A. Efros, and M. Hebert. Recovering Surface Layout from an Image. *IJCV*, 2007.
- R. Khandon and D. Roth. Learning to reason. *Journal of the ACM (JACM)*, 1997.
- P. Kohli, L. Ladicky, and P. H. S. Torr. Robust higher order potentials for enforcing label consistency. *IJCV*, 2009.
- T. Koo, A. M. Rush, M. Collins, T. Jaakkola, and D. Sontag. Dual decomposition for parsing with non-projective head automata. In *Proc. EMNLP*, 2010.
- J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML*, 2001.
- S. L. Lauritzen. *Graphical models*, volume 17. Oxford University Press, USA, 1996.
- G. Lebanon and J. Lafferty. Boosting and maximum likelihood for exponential models. *Proc. NIPS*, 2002.
- D. C. Lee, M. Hebert, and T. Kanade. Geometric Reasoning for Single Image Structure Recovery. In *Proc. CVPR*, 2009.
- A. Levin and Y. Weiss. Learning to Combine Bottom-Up and Top-Down Segmentation. In *Proc. ECCV*, 2006.
- D. Lin, S. Fidler, and R. Urtasun. Holistic scene understanding for 3d object detection with rgbd cameras. In *Proc. ICCV*, 2013.
- T. Meltzer, A. Globerson, and Y. Weiss. Convergent message passing algorithms—a unifying view. In *Proc. UAI*, 2009.
- O. Meshi, A. Jainovitch, A. Globerson, and N. Friedman. Convexifying the bethe free energy. In *Proc. UAI*, 2009.
- O. Meshi, D. A. Sontag, T. S. Jaakkola, and A. Globerson. Learning efficiently with approximate inference via dual losses. In *Proc. ICML*, 2010.
- O. Meshi, N. Srebro, and T. Hazan. Efficient Training of Structured SVMs via Soft Constraints. In *Proc. AISTATS*, 2015.
- Y. Nesterov. *Introductory lectures on convex optimization: A basic course*. Springer, 2004.
- R.T. Rockafellar. *Convex analysis*. Princeton university press, 1970.
- M. Salzmann and R. Urtasun. Beyond Feature Points: Structured Prediction for Monocular Non-rigid 3D Reconstruction. In *Proc. ECCV*, 2012.
- A. G. Schwing and R. Urtasun. Efficient exact inference for 3d indoor scene understanding. In *Proc. ECCV*, 2012.
- A. G. Schwing and R. Urtasun. Fully Connected Deep Structured Networks. <http://arxiv.org/abs/1503.02351>, 2015.
- A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Distributed message passing for large scale graphical models. In *Proc. CVPR*, 2011.
- A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Efficient structured prediction for 3d indoor scene understanding. In *Proc. CVPR*, 2012a.
- A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Efficient Structured Prediction with Latent Variables for General Graphical Models. In *Proc. ICML*, 2012b.
- D. Sontag, T. Meltzer, A. Globerson, T. Jaakkola, and Y. Weiss. Tightening LP relaxations for MAP using message passing. In *Proc. UAI*, 2008.
- C. Sutton and A. McCallum. Piecewise training for structured prediction. *Machine Learning*, 2009.
- R. Tappeander, P. Richtárik, and J. Gondzio. Inexact coordinate descent: complexity and preconditioning. *arXiv preprint arXiv:1304.5530*, 2013.

- D. Tarlow and R. S. Zemel. Structured output learning with high order loss functions. In *Proc. AISTATS*, 2012.
- D. Tarlow, I. E. Givoni, and R. S. Zemel. Hopmap: Efficient message passing with high order potentials. In *Proc. AISTATS*, 2010.
- D. Tarlow, D. Batra, P. Kohli, and V. Kolmogorov. Dynamic tree block coordinate ascent. *Proc. ICML*, 2011.
- B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. *Proc. NIPS*, 2004.
- B. Taskar, V. Chatalbashev, D. Koller, and C. Guestrin. Learning structured prediction models: A large margin approach. In *Proc. ICML*, 2005.
- P. Tseng and D.P. Bertsekas. Relaxation methods for problems with strictly convex separable costs and linear constraints. *Mathematical Programming*, 38(3):303-321, 1987.
- I. Tschantzaris, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *Proc. ICML*, 2004.
- M. J. Wainwright. Estimating the Wrong Graphical Model: Benefits in the Computation-Limited Setting. *JMLR*, 2006.
- M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 2008.
- M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Tree-reweighted belief propagation algorithms and approximate ML estimation by pseudo-moment matching. In *Proc. Workshop on AI and Stats.*, 2003.
- M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. A new class of upper bounds on the log partition function. *Trans. on Information Theory*, 51(7):2313-2335, 2005.
- K. Yamaguchi, T. Hazan, D. McAllester, and R. Urtasun. Continuous markov random fields for robust stereo estimation. In *Proc. ECCV*, 2012.
- K. Yamaguchi, D. McAllester, and R. Urtasun. Robust monocular epipolar flow estimation. In *Proc. CVPR*, 2013.
- C. Yanover, O. Schueler-Furman, and Y. Weiss. Minimizing and learning energy functions for side-chain prediction. In *RECOMB*. Springer, 2007.
- J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *Proc. CVPR*, 2012.
- J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *Information Theory*, 2005.

Distributed Submodular Maximization

Baharan Mirzasoileiman

*Department of Computer Science
ETH Zurich*

Universitaetstrasse 6, 8092 Zurich, Switzerland

BAHARANM@INF.ETHZ.CH

Amin Karbasi

*School of Engineering and Applied Science
Yale University
New Haven, USA*

AMIN.KARBASI@YALE.EDU

Rik Sarkar

*Department of Informatics
University of Edinburgh
10 Crichton St, Edinburgh EH8 9AB, United Kingdom*

RSARKAR@INF.ED.AC.UK

Andreas Krause

*Department of Computer Science
ETH Zurich
Universitaetstrasse 6, 8092 Zurich, Switzerland*

KRAUSEA@ETHZ.CH

Editor: Jeff Bilmes

Frey, 2007) to active set selection for non-parametric learning (Rasmussen, 2004), to viral marketing (Kenpe et al., 2003), and data subset selection for the purpose of training complex models (Lin and Bilmes, 2011). Many such problems can be reduced to the problem of *maximizing a submodular set function* subject to cardinality or other feasibility constraints such as matroid, or knapsack constraints (Krause and Gomes, 2010; Krause and Golovin, 2012; Lee et al., 2009a).

Submodular functions exhibit a natural diminishing returns property common in many well known objectives: the marginal benefit of any given element decreases as we select more and more elements. Functions such as entropy or maximum weighted coverage are typical examples of functions with diminishing returns. As a result, submodular function optimization has numerous applications in machine learning and social networks: viral marketing (Kenpe et al., 2003; Babaei et al., 2013; Mirzasoileiman et al., 2012), information gathering (Krause and Guestrin, 2011), document summarization (Lin and Bilmes, 2011), and active learning (Golovin and Krause, 2011; Guillory and Bilmes, 2011).

Although maximizing a submodular function is NP-hard in general, a seminal result of Nemhauser et al. (1978) states that a simple greedy algorithm produces solutions competitive with the optimal (intractable) solution (Nemhauser and Wolsey, 1978; Feige, 1998). However, such greedy algorithms or their accelerated variants (Minoux, 1978; Badanidiyuru and Vondrák, 2014; Mirzasoileiman et al., 2015a) do not scale well when the dataset is massive. As data volumes in modern applications increase faster than the ability of individual computers to process them, we need to look at ways to adapt our computations using parallelism.

MapReduce (Dean and Ghemawat, 2008) is arguably one of the most successful programming models for reliable and efficient parallel computing. It works by distributing the data to independent machines: *map* tasks redistribute the data for appropriate parallel processing and the output then gets sorted and processed in parallel by *reduce* tasks.

To perform submodular optimization in MapReduce, we need to design suitable parallel algorithms. The greedy algorithms that work well for centralized submodular optimization do not translate easily to parallel environments. The algorithms are inherently sequential in nature, since the marginal gain from adding each element is dependent on the elements picked in previous iterations. This mismatch makes it inefficient to apply classical algorithms directly to parallel setups.

In this paper, we develop a distributed procedure for maximizing submodular functions, that can be easily implemented in MapReduce. Our strategy is to partition the data (e.g., randomly) and process it in parallel. In particular:

- We present a simple, parallel protocol, called GREDI for distributed submodular maximization subject to cardinality constraints. It requires minimal communication, and can be easily implemented in MapReduce style parallel computation models.
- We show that under some natural conditions, for large datasets the quality of the obtained solution is provably competitive with the best centralized solution.
- We discuss extensions of our approach to obtain approximation algorithms for (not-necessarily monotone) submodular maximization subject to more general types of constraints, including matroid and knapsack constraints.

Abstract

Many large-scale machine learning problems—clustering, non-parametric learning, kernel machines, etc.—require selecting a small yet representative subset from a large dataset. Such problems can often be reduced to maximizing a submodular set function subject to various constraints. Classical approaches to submodular optimization require centralized access to the full dataset, which is impractical for truly large-scale problems. In this paper, we consider the problem of submodular function maximization in a distributed fashion. We develop a simple, two-stage protocol GREDI, that is easily implemented using MapReduce style computations. We theoretically analyze our approach, and show that under certain natural conditions, performance close to the centralized approach can be achieved. We begin with monotone submodular maximization subject to a cardinality constraint, and then extend this approach to obtain approximation guarantees for (not necessarily monotone) submodular maximization subject to more general constraints including matroid or knapsack constraints. In our extensive experiments, we demonstrate the effectiveness of our approach on several applications, including sparse Gaussian process inference and exemplar based clustering on tens of millions of examples using Hadoop.

Keywords: distributed computing, submodular functions, approximation algorithms, greedy algorithms, map-reduce

1. Introduction

Numerous machine learning tasks require selecting representative subsets of manageable size out of large datasets. Examples range from exemplar based clustering (Dueck and

- We implement our approach for exemplar based clustering and active set selection in Hadoop, and show how our approach allows to scale exemplar based clustering and sparse Gaussian process inference to datasets containing tens of millions of points.
- We extensively evaluate our algorithm on several machine learning problems, including exemplar based clustering, active set selection and finding cuts in graphs, and show that our approach leads to parallel solutions that are very competitive with those obtained via centralized methods (98% in exemplar based clustering, 97% in active set selection, 90% in finding cuts).

This paper is organized as follows. We begin in Section 2 by discussing background and related work. In Section 3, we formalize the distributed submodular maximization problem under cardinality constraints, and introduce example applications as well as naive approaches toward solving the problem. We subsequently present our GREEDY algorithm in Section 4, and prove its approximation guarantees. We then consider maximizing a submodular function subject to more general constraints in Section 5. We also present computational experiments on very large datasets in Section 6, showing that in addition to its provable approximation guarantees, our algorithm provides results close to the centralized greedy algorithm. We conclude in Section 7.

2. Background and Related Work

2.1 Distributed Data Analysis and MapReduce

Due to the rapid increase in dataset sizes, and the relatively slow advances in sequential processing capabilities of modern CPUs, parallel computing paradigms have received much interest. Inhabiting a sweet spot of resiliency, expressivity and programming ease, the MapReduce style computing model (Dean and Ghemawat, 2008) has emerged as prominent foundation for large scale machine learning and data mining algorithms (Chu et al., 2007; Ekamayake et al., 2008). A MapReduce job takes the input data as a set of $< key; value >$ pairs. Each job consists of three stages: the *map* stage, the *shuffle* stage, and the *reduce* stage. The map stage, partitions the data randomly across a number of machines by associating each element with a *key* and produce a set of $< key; value >$ pairs. Then, in the shuffle stage, the value associated with all of the elements with the same key gets merged and sent to the same machine. Each reducer then processes the values associated with the same key and outputs a set of new $< key; value >$ pairs with the same key. The reducers' output could be an input to another MapReduce job, and a program in MapReduce paradigm can consist of multiple rounds of map and reduce stages (Karloff et al., 2010).

2.2 Centralized and Streaming Submodular Maximization

The problem of centralized maximization of submodular functions has received much interest, starting with the seminal work of Nemhauser et al. (1978). Recent work has focused on providing approximation guarantees for more complex constraints (for a more detailed account, see the recent survey by Krause and Golovin, 2012). Golovin et al. (2010) consider an algorithm for online distributed submodular maximization with an application to sensor selection. However, their approach requires k stages of communication, which is unrealistic

for large k in a MapReduce style model. Krause and Gomes (2010) consider the problem of submodular maximization in a streaming model; however, their approach makes strong assumptions about the way the data stream is generated and is not applicable to the general distributed setting. Recently, Badanidiyuru et al. (2014) provide a single pass streaming algorithm for cardinality-constrained submodular maximization with $1/2 - \epsilon$ approximation guarantee to the optimum solution that makes no assumptions on the data stream.

There has also been new improvements in the running time of the standard greedy solution for solving SET-COVER (a special case of submodular maximization) when the data is large and disk resident (Cornode et al., 2010). More generally, Badanidiyuru and Vondrak (2014) and Mirzasoileman et al. (2015a) improve the running time of the greedy algorithm for maximizing a monotone submodular function by reducing the number of oracle calls to the objective function. Very recently, Mirzasoileman et al. (2016) provided a fast algorithm for maximizing non-monotone submodular functions under general constraints. In a similar spirit, Wei et al. (2014) propose a multi-stage framework for submodular maximization. In order to reduce the memory and computation cost, they apply an approximate greedy procedure to maximize surrogate (proxy) submodular functions instead of optimizing the target function at each stage. The above approaches are sequential in nature and it is not clear how to parallelize them. However, they can be naturally integrated into our distributed framework to achieve further acceleration.

2.3 Scaling Up: Distributed Algorithms

Recent work has focused on specific instances of submodular optimization in distributed settings. Such scenarios often occur in large-scale graph mining problems where the data itself is too large to be stored on one machine. In particular, Chierichetti et al. (2010) address the MAX-COVER problem and provide a $(1 - 1/e - \epsilon)$ approximation to the centralized algorithm at the cost of passing over the dataset many times. Their result is further improved by Brelloch et al. (2011). Lattanzi et al. (2011) address more general graph problems by introducing the idea of filtering, namely, reducing the size of the input in a distributed fashion so that the resulting, much smaller, problem instance can be solved on a single machine. This idea is, in spirit, similar to our distributed method GREEDY. In contrast, we provide a more general framework, and characterize settings where performance competitive with the centralized setting can be obtained. The present version is a significant extension of our previous conference paper (Mirzasoileman et al., 2013), providing theoretical guarantees for both monotone and non-monotone submodular maximization problems subject to more general types of constraints, including matroid and knapsack constraints (described in Section 5), and additional empirical results (Section 6). Parallel to our efforts (Mirzasoileman et al., 2013), Kumar et al. (2013) has taken the approach of adapting the sequential greedy algorithm to distributed settings. However, their method requires knowledge of the ratio between the largest and smallest marginal gains of the elements, and generally requires a non-constant (logarithmic) number of rounds. We provide empirical comparisons in Section 6.4.



Figure 1: Cluster exemplars (left column) discovered by our distributed algorithm Greedy described in Section 4 applied to the Tiny Images dataset (Torralba et al., 2008), and a set of representatives from each cluster.

3. Submodular Maximization

In this section, we first review submodular functions and how to greedily maximize them. We then describe the *distributed submodular maximization* problem, the focus of this paper. Finally, we discuss two naive approaches towards solving this problem.

3.1 Greedy Submodular Maximization

Suppose that we have a large dataset of images, e.g., the set of all images on the Web or an online image hosting website such as Flickr, and we wish to retrieve a subset of images that best represents the visual appearance of the dataset. Collectively, these images can be considered as *exemplars* that *summarize* the visual categories of the dataset as shown in Fig. 1.

One way to approach this problem is to formalize it as the *k-medoid* problem. Given a set $V = \{e_1, e_2, \dots, e_n\}$ of images (called ground set) associated with a (not necessarily symmetric) dissimilarity function, we seek to select a subset $S \subseteq V$ of at most k exemplars or cluster centers, and then assign each image in the dataset to its least dissimilar exemplar. If an element $e \in V$ is assigned to exemplar $v \in S$, then the cost associated with e is the dissimilarity between e and v . The goal of the *k-medoid* problem is to choose exemplars that minimize the sum of dissimilarities between every data point $e \in V$ and its assigned cluster center.

Solving the *k-medoid* problem optimally is NP-hard, however, as we discuss in Section 3.4, we can transform this problem, and many other summarization tasks, to the problem of maximizing a monotone submodular function subject to a cardinality constraint

$$\max_{S \subseteq V} f(S) \quad \text{s.t.} \quad |S| \leq k. \quad (1)$$

Submodular functions are set functions which satisfy the following natural diminishing returns property.

Definition 1 (c.f., Nemhauser et al. (1978)) A set function $f : 2^V \rightarrow \mathbb{R}$ is submodular, if for every $A \subseteq B \subseteq V$ and $e \in V \setminus B$

$$f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B).$$

Furthermore, f is called monotone iff for all $A \subseteq B \subseteq V$ it holds that $f(A) \leq f(B)$.

We will generally additionally require that f is nonnegative, i.e., $f(A) \geq 0$ for all sets A . Problem (1) is NP-hard for many classes of submodular functions (Feige, 1998). A fundamental result by Nemhauser et al. (1978) establishes that a simple greedy algorithm that starts with the empty set and iteratively augments the current solution with an element of maximum incremental value

$$v^* = \arg \max_{v \in V \setminus A} f(A \cup \{v\}), \quad (2)$$

continuing until k elements have been selected, is guaranteed to provide a constant factor approximation.

Theorem 2 (Nemhauser et al., 1978) For any non-negative and monotone submodular function f , the greedy heuristic always produces a solution $A^{\text{gc}}[k]$ of size k that achieves at least a constant factor $(1 - 1/e)$ of the optimal solution.

$$f(A^{\text{gc}}[k]) \geq (1 - 1/e) \max_{|A| \leq k} f(A).$$

This result can be easily extended to $f(A^{\text{gc}}[l]) \geq (1 - e^{-l/k}) \max_{|A| \leq k} f(A)$, where l and k are two positive integers (see, Krause and Golovin, 2012).

3.2 Distributed Submodular Maximization

In many today’s applications where the size of the ground set $|V| = n$ is very large and cannot be stored on a single computer, running the standard greedy algorithm or its variants (e.g., lazy evaluations, Minoux, 1978; Leskovec et al., 2007; Mirzasoileiman et al., 2015a) in a centralized manner is infeasible. Hence, we seek a solution that is suitable for large-scale parallel computation. The greedy method described above is in general difficult to parallelize, since it is inherently sequential: at each step, only the object with the highest marginal gain is chosen and every subsequent step depends on the preceding ones.

Concretely, we consider the setting where the ground set V is very large and cannot be handled on a single machine, thus must be distributed among a set of m machines. While there are several approaches towards parallel computation, in this paper we consider the following model that can be naturally implemented in MapReduce. The computation proceeds in a sequence of rounds. In each round, the dataset is distributed to m machines. Each machine i carries out computations independently in parallel on its local data. After all machines finish, they synchronize by exchanging a limited amount of data (of size polynomial in k and m , but independent of n). Hence, any distributed algorithm in this model must specify: 1) how to distribute V among the m machines, 2) which algorithm should run on each machine, and 3) how to communicate and merge the resulting solutions.

In particular, the distributed submodular maximization problem requires the specification of the above steps in order to implement an approach for submodular maximization. More precisely, given a monotone submodular function f , a cardinality constraint k , and a number of machines m , we wish to produce a solution $A^{\text{d}}[m, k]$ of size k such that $f(A^{\text{d}}[m, k])$ is competitive with the optimal centralized solution $\max_{|A| \leq k, A \subseteq V} f(A)$.

3.3 Naive Approaches Towards Distributed Submodular Maximization

One way to solve problem (1) in a distributed fashion is as follows. The dataset is first partitioned (randomly, or using some other strategy) onto the m machines, with V_i representing the data allocated to machine i . We then proceed in k rounds. In each round, all machines—in parallel—compute the marginal gains of all elements in their sets V_i . Next, they communicate their candidate to a central processor, who identifies the globally best element, which is in turn communicated to the m machines. This element is then taken into account for computing the marginal gains and selecting the next elements. This algorithm (up to decisions on how break ties) implements exactly the centralized greedy algorithm, and hence provides the same approximation guarantees on the quality of the solution. Unfortunately, this approach requires synchronization after each of the k rounds. In many applications, k is quite large (e.g., tens of thousands), rendering this approach impractical for MapReduce style computations.

An alternative approach for large k would be to greedily select k/m elements independently on each machine (without synchronization), and then merge them to obtain a solution of size k . This approach that requires only two rounds (as opposed to k), is much more communication efficient, and can be easily implemented using a single MapReduce stage. Unfortunately, many machines may select redundant elements, and thus the merged solution may suffer from diminishing returns. It is not hard to construct examples for which this approach produces solutions that are a factor $\Omega(m)$ worse than the centralized solution.

In Section 4, we introduce an alternative protocol GREEDY, which requires little communication, while at the same time yielding a solution competitive with the centralized one, under certain natural additional assumptions.

3.4 Applications of Distributed Submodular Maximization

In this part, we discuss two concrete problem instances, with their corresponding submodular objective functions f , where the size of the datasets often requires a distributed solution for the underlying submodular maximization.

3.4.1 LARGE-SCALE NONPARAMETRIC LEARNING

Nonparametric learning (i.e., learning of models whose complexity may depend on the dataset size n) are notoriously hard to scale to large datasets. A concrete instance of this problem arises from training Gaussian processes or performing MAP inference in Determinantal Point Processes, as considered below. Similar challenges arise in many related learning methods, such as training kernel machines, when attempting to scale them to large data sets.

Active Set Selection in Sparse Gaussian Processes (GPs). Formally a GP is a joint probability distribution over a (possibly infinite) set of random variables \mathbf{X}_V , indexed by the ground set V , such that every (finite) subset \mathbf{X}_S for $S = \{e_1, \dots, e_s\}$ is distributed according to a multivariate normal distribution. More precisely, we have

$$P(\mathbf{X}_S = \mathbf{x}_S) = \mathcal{N}(\mathbf{X}_S; \mu_S, \Sigma_S),$$

where $\mu = (\mu_{e_1}, \dots, \mu_{e_s})$ and $\Sigma_S = [\kappa_{e_i, e_j}]$ are prior mean and covariance matrix, respectively. The covariance matrix is parameterized via a positive definite kernel $\mathcal{K}(\cdot, \cdot)$. As a

concrete example, when elements of the ground set V are embedded in a Euclidean space, a commonly used kernel in practice is the squared exponential kernel defined as follows:

$$\mathcal{K}(e_i, e_j) = \exp(-\|e_i - e_j\|_2^2 / \theta^2).$$

Gaussian processes are commonly used as priors for nonparametric regression. In GP regression, each data point $e \in V$ is considered a random variable. Upon observations $\mathbf{Y}_A = \mathbf{X}_A + \mathbf{n}_A$ (where \mathbf{n}_A is a vector of independent Gaussian noise with variance σ^2), the predictive distribution of a new data point $e \in V$ is a normal distribution $P(\mathbf{X}_e | \mathbf{Y}_A) = \mathcal{N}(\mu_{e|A}, \Sigma_{e|A}^2)$, where mean $\mu_{e|A}$ and variance $\Sigma_{e|A}^2$ are given by

$$\mu_{e|A} = \mu_e + \Sigma_{e,A}(\Sigma_{A,A} + \sigma^2 \mathbf{I})^{-1}(\mathbf{X}_A - \mu_A), \quad (3)$$

$$\Sigma_{e|A}^2 = \sigma_e^2 - \Sigma_{e,A}(\Sigma_{A,A} + \sigma^2 \mathbf{I})^{-1}\Sigma_{A,e}. \quad (4)$$

Evaluating (3) and (4) is computationally expensive as it requires solving a linear system of $|A|$ variables. Instead, most efficient approaches for making predictions in GPs rely on choosing a small—so called *active*—set of data points. For instance, in the Informative Vector Machine (IVM) one seeks a set S such that the *information gain*, defined as

$$f(S) = I(\mathbf{Y}_S; \mathbf{X}_V) = H(\mathbf{X}_V) - H(\mathbf{X}_V | \mathbf{Y}_S) = \frac{1}{2} \log \det(\mathbf{I} + \sigma^{-2} \Sigma_S)$$

is maximized. It can be shown that this choice of f is monotone submodular (Krause and Guestrin, 2005a). For medium-scale problems, the standard greedy algorithms provide good solutions. For massive data however, we need to resort to distributed algorithms. In Section 6, we will show how GREEDY can choose near-optimal subsets out of a dataset of 45 million vectors.

Inference for Determinantal Point Processes. A very similar problem arises when performing inference in Determinantal Point Processes (DPPs). DPPs (Macchi, 1975) are distributions over subsets with a preference for diversity, i.e., there is a higher probability associated with sets containing dissimilar elements. Formally, a point process \mathcal{P} on a set of items $V = \{1, 2, \dots, N\}$ is a probability measure on 2^V (the set of all subsets of V). \mathcal{P} is called *determinantal point process* if for every $S \subseteq V$ we have:

$$\mathcal{P}(S) \propto \det(K_S),$$

where K is a positive semidefinite kernel matrix, and $K_S \equiv [\kappa_{i,j}]_{i,j \in S}$ is the restriction of K to the entries indexed by elements of S (we adopt that $\det(K_\emptyset) = 1$). The normalization constant can be computed explicitly from the following equation

$$\sum_S \det(K_S) = \det(\mathbf{I} + K),$$

where \mathbf{I} is the $N \times N$ identity matrix. Intuitively, the kernel matrix determines which items are similar and therefore less likely to appear together.

In order to find the most diverse and informative subset of size k , we need to find $\arg \max_{|S| \leq k} \det(K_S)$ which is NP-hard, as the total number of possible subsets is exponential (Ko et al., 1995). However, the objective function is log-submodular, i.e. $f(S) = \log \det(K_S)$ is a submodular function (Kulesza, 2012). Hence, MAP inference in large DPPs is another potential application of distributed submodular maximization.

3.4.2 LARGE-SCALE EXEMPLAR BASED CLUSTERING

Suppose we wish to select a set of exemplars, that best represent a massive dataset. One approach for finding such exemplars is solving the k -medoid problem (Kaufman and Rousseeuw, 2009), which aims to minimize the sum of pairwise dissimilarities between exemplars and elements of the dataset. More precisely, let us assume that for the dataset V we are given a nonnegative function $l : V \times V \rightarrow \mathbb{R}$ (not necessarily assumed symmetric, nor obeying the triangle inequality) such that $l(\cdot, \cdot)$ encodes dissimilarity between elements of the underlying set V . Then, the cost function for the k -medoid problem is:

$$L(S) = \frac{1}{|V|} \sum_{v \in V} \min_{e \in S} l(e, v). \quad (5)$$

Finding the subset

$$S^* = \arg \min_{|S|=k} L(S)$$

of cardinality at most k that minimizes the cost function (5) is NP-hard. However, by introducing an auxiliary element e_0 , a.k.a. phantom exemplar, we can turn L into a monotone submodular function (Krause and Gomes, 2010)

$$f(S) = L(\{e_0\}) - L(S \cup \{e_0\}). \quad (6)$$

In words, f measures the decrease in the loss associated with the set S versus the loss associated with just the auxiliary element. We begin with a phantom exemplar and try to find the active set that together with the phantom exemplar reduces the value of our loss function more than any other set. Technically, any point e_0 that satisfies the following condition can be used as a phantom exemplar:

$$\max_{v' \in V} l(v, v') \leq l(v, e_0), \quad \forall v \in V \setminus S.$$

This condition ensures that once the distance between any $v \in V \setminus S$ and e_0 is greater than the maximum distance between elements in the dataset, then $L(S \cup \{e_0\}) = L(S)$. As a result, maximizing f (a monotone submodular function) is equivalent to minimizing the cost function L . This problem becomes especially computationally challenging when we have a large dataset and we wish to extract a manageable-size set of exemplars, further motivating our distributed approach.

3.4.3 OTHER EXAMPLES

Numerous other real world problems in machine learning can be modeled as maximizing a monotone submodular function subject to appropriate constraints (e.g., cardinality, matroid, knapsack). To name a few, specific applications that have been considered range from efficient content discovery for web crawlers and multi topic blog-watch (Chierichetti et al., 2010), over document summarization (Lin and Bilmes, 2011) and speech data subset selection (Wei et al., 2013), to outbreak detection in social networks (Leskovec et al., 2007), online advertising and network routing (De Vries and Vohra, 2003), revenue maximization in social networks (Hartline et al., 2008), and inferring network of influence (Gomez Rodriguez et al., 2010). In all such examples, the size of the dataset (e.g., number of webpages,

size of the corpus, number of blogs in the blogosphere, number of nodes in social networks) is massive, thus GREEDI offers a scalable approach, in contrast to the standard greedy algorithm, for such problems.

4. The GREEDI Approach for Distributed Submodular Maximization

In this section we present our main results. We first provide our distributed solution GREEDI for maximizing submodular functions under cardinality constraints. We then show how we can make use of the geometry of data inherent in many practical settings in order to obtain strong data-dependent bounds on the performance of our distributed algorithm.

4.1 An Intractable, yet Communication Efficient Approach

Before we introduce GREEDI, we first consider an intractable, but communication-efficient two-round parallel protocol to illustrate the ideas. This approach, shown in Algorithm 1, first distributes the ground set V to m machines. Each machine then finds the *optimal* solution, i.e., a set of cardinality at most k , that maximizes the value of f in each partition. These solutions are then merged, and the optimal subset of cardinality k is found in the combined set. We denote this distributed solution by $f(A^d[m, k])$.

As the optimum centralized solution $A^c[k]$ achieves the maximum value of the submodular function, it is clear that $f(A^c[k]) \geq f(A^d[m, k])$. For the special case of selecting a single element $k = 1$, we have $f(A^c[1]) = f(A^d[m, 1])$. Furthermore, for *modular* functions f (i.e., those for which f and $-f$ are both submodular), it is easy to see that the distributed scheme in fact returns the optimal centralized solution as well. In general, however, there can be a gap between the distributed and the centralized solution. Nonetheless, as the following theorem shows, this gap cannot be more than $1/\min(m, k)$. Furthermore, this result is tight.

Theorem 3 *Let f be a monotone submodular function and let $k > 0$. Then, $f(A^d[m, k]) \geq \frac{1}{\min(m, k)} f(A^c[k])$. In contrast, for any value of m and k , there is a monotone submodular function f such that $f(A^c[k]) = \min(m, k) \cdot f(A^d[m, k])$.*

The proof of all the theorems can be found in the appendix. The above theorem fully characterizes the performance of Algorithm 1 in terms of the best centralized solution. In practice, we cannot run Algorithm 1, since there is no efficient way to identify the optimum subset $A^c[k]$ in set V , unless P=NP. In the following, we introduce an efficient distributed approximation – GREEDI. We will further show, that under some additional assumptions, much stronger guarantees can be obtained.

4.2 Our GREEDI Approximation

Our efficient distributed method GREEDI is shown in Algorithm 2. It parallels the intractable Algorithm 1, but replaces the selection of optimal subsets, i.e., $A^c[k]$, by greedy solutions $A_g^c[k]$. Due to the approximate nature of the greedy algorithm, we allow it to pick sets slightly larger than k . More precisely, GREEDI is a two-round algorithm that takes the ground set V , the number of partitions m , and the cardinality constraint k . It first distributes the ground set over m machines. Then each machine separately runs the

Algorithm 1 Inefficient Distributed Submodular Maximization

Input: Set V , #of partitions m , constraints k .

Output: Set $A^d[m, k]$.

- 1: Partition V into m sets V_1, V_2, \dots, V_m .
 - 2: In each partition V_i find the optimum set $A_i^c[k]$ of cardinality k .
 - 3: Merge the resulting sets: $B = \cup_{i=1}^m A_i^c[k]$.
 - 4: Find the optimum set of cardinality k in B . Output this solution $A^d[m, k]$.
-

Algorithm 2 Greedy Distributed Submodular Maximization (GREEDI)

Input: Set V , #of partitions m , constraints κ .

Output: Set $A^{sd}[m, \kappa]$.

- 1: Partition V into m sets V_1, V_2, \dots, V_m (arbitrarily or at random).
 - 2: Run the standard greedy algorithm on each set V_i to find a solution $A_i^{gc}[\kappa]$.
 - 3: Find $A_{\max}^{gc}[\kappa] = \arg \max_A \{F(A) : A \in \{A_i^{gc}[\kappa], \dots, A_m^{gc}[\kappa]\}\}$
 - 4: Merge the resulting sets: $B = \cup_{i=1}^m A_i^{gc}[\kappa]$.
 - 5: Run the standard greedy algorithm on B to find a solution $A_B^{gc}[\kappa]$.
 - 6: Return $A^{sd}[m, \kappa] = \arg \max_A \{F(A) : A \in \{A_{\max}^{gc}[\kappa], A_B^{gc}[\kappa]\}\}$.
-

standard greedy algorithm by sequentially finding an element $e \in V_i$ that maximizes the discrete derivative (2). Each machine i -in parallel-continues adding elements to the set $A_i^{gc}[\cdot]$ until it reaches κ elements. We define $A_{\max}^{gc}[\kappa]$ to be the set with the maximum value among $\{A_1^{gc}[\kappa], A_2^{gc}[\kappa], \dots, A_m^{gc}[\kappa]\}$. Then the solutions are merged, i.e., $B = \cup_{i=1}^m A_i^{gc}[\kappa]$, and another round of greedy selection is performed over B until κ elements are selected. We denote this solution by $A_B^{gc}[\kappa]$. The final distributed solution with parameters m and κ , denoted by $A^{sd}[m, \kappa]$, is the set with a higher value between $A_{\max}^{gc}[\kappa]$ and $A_B^{gc}[\kappa]$ (c.f., Figure 2 shows GREEDI schematically). The following result parallels Theorem 3.

Theorem 4 Let f be a monotone submodular function and $\kappa \geq k$. Then

$$f(A^{sd}[m, \kappa]) \geq \frac{(1 - e^{-\kappa/k})}{\min(m, k)} f(A^c[k]).$$

For the special case of $\kappa = k$ the result of 4 simplifies to $f(A^{sd}[m, \kappa]) \geq \frac{(1-1/e)}{\min(m, k)} f(A^c[k])$. Moreover, it is straightforward to generalize GREEDI to multiple rounds (i.e., more than two) for very large datasets.

In light of Theorem 3, one can expect that in general it is impossible to eliminate the dependency of the distributed solution on $\min(k, m)$ ¹. However, as we show in the sequel, in many practical settings, the ground set V exhibits rich geometrical structure that can be used to obtain stronger guarantees.

1. It has been very recently shown by Mirzasoleiman et al. (2015b) that the tightest dependency is $\Theta(\sqrt{\min(m, k)})$.

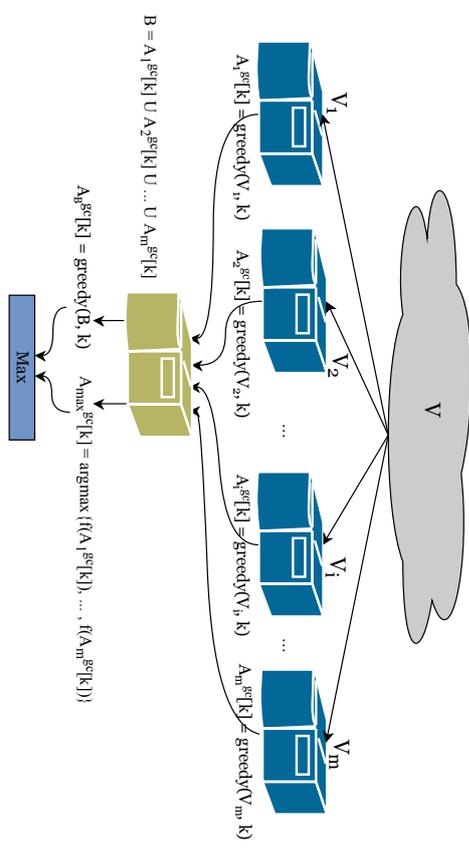


Figure 2: Illustration of our two-round algorithm GREEDI

4.3 Performance on Datasets with Geometric Structure

In practice, we can hope to do much better than the worst case bounds shown previously, by exploiting underlying structure often present in real data and important set functions. In this part, we assume that a metric $d : V \times V \rightarrow \mathbb{R}$ exists on the data elements, and analyze performance of the algorithm on functions that vary slowly with changes in the input. We refer to these as *Lipschitz functions*:

Definition 5 Let $\lambda > 0$. A set function $f : 2^V \rightarrow \mathbb{R}$ is λ -Lipschitz w.r.t. metric d on V , if for any integer k , any equal sized sets $S = \{e_1, e_2, \dots, e_k\} \subseteq V$ and $S' = \{e'_1, e'_2, \dots, e'_k\} \subseteq V$ and any matching of elements: $M = \{(e_1, e'_1), (e_2, e'_2), \dots, (e_k, e'_k)\}$, the difference between $f(S)$ and $f(S')$ is bounded by:

$$|f(S) - f(S')| \leq \lambda \sum_i d(e_i, e'_i). \quad (7)$$

We can show that the objective functions from both examples in Section 3.4 are λ -Lipschitz for suitable kernels/distance functions:

Proposition 6 Suppose that the covariance matrix of a Gaussian process is parametrized via a positive definite kernel $\mathcal{K} : V \times V \rightarrow \mathbb{R}$ which is Lipschitz continuous with respect to metric $d : V \times V \rightarrow \mathbb{R}$ with constant \mathcal{L} , i.e., for any triple of points $x_1, x_2, x_3 \in V$, we have $|\mathcal{K}(x_1, x_3) - \mathcal{K}(x_2, x_3)| \leq \mathcal{L}d(x_1, x_2)$. Then, the mutual information $I(\mathbf{Y}; S; \mathbf{X}) = \frac{1}{2} \log \det(\mathbf{I} + \mathbf{K})$ for the Gaussian process is λ -Lipschitz with $\lambda = \mathcal{L}k^3$, where k is the number of elements in the selected subset S .

Proposition 7 Let $d : V \times V \rightarrow \mathbb{R}$ be a metric on the elements of the dataset. Furthermore, let $l : V \times V \rightarrow \mathbb{R}$ encode the dissimilarity between elements of the underlying set V . Then for $l = d^\alpha$, $\alpha \geq 1$ the loss function $L(S) = \frac{1}{|V|} \sum_{e \in V} \min_{e' \in S} l(e, e')$ (and hence also the corresponding submodular utility function f) is λ -Lipschitz with $\lambda = \alpha R^{\alpha-1}$, where R is the diameter of the ball encompassing elements of the dataset in the metric space. In particular, for the k -medoid problem, which minimizes the loss function over all clusters with respect to $l = d$, we have $\lambda = 1$, and for the k -means problem, which minimizes the loss function over all clusters with respect to $l = d^2$, we have $\lambda = 2R$.

Beyond Lipschitz-continuity, many practical instances of submodular maximization can be expected to satisfy a natural density condition. Concretely, whenever we consider a representative set (i.e., optimal solution to the submodular maximization problem), we expect that any of its constituent elements has potential candidates for replacement in the ground set. For example, in our exemplar-based clustering application, we expect that cluster centers are not isolated points, but have many almost equally representative points close by. Formally, for any element $v \in V$, we define its α -neighborhood as the set of elements in V within distance α from v (i.e., α -close to v):

$$N_\alpha(v) = \{w : d(v, w) \leq \alpha\}.$$

By λ -Lipschitz-continuity, it must hold that if we replace element v in set S by an α -close element v' (i.e., $v' \in N_\alpha(v)$) to get a new set S' of equal size, it must hold that $|f(S') - f(S)| \leq \alpha\lambda$.

As described earlier, our algorithm GREEDI partitions V into sets V_1, V_2, \dots, V_m for parallel processing. If in addition we assume that elements are assigned uniformly at random to different machines, α -neighborhoods are sufficiently dense, and the submodular function is Lipschitz continuous, then GREEDI is guaranteed to produce a solution close to the centralized one. More formally, we have the following theorem.

Theorem 8 Under the conditions that 1) elements are assigned uniformly at random to m machines, 2) for each $e_i \in A^c[k]$ we have $|N_\alpha(e_i)| \geq km \log(k/\delta^{1/m})$, and 3) f is λ -Lipschitz continuous, then with probability at least $(1 - \delta)$ the following holds:

$$f(A^{gd}[m, \kappa]) \geq (1 - e^{-\kappa/k})(f(A^c[k]) - \lambda\alpha k).$$

Note that once the above conditions are satisfied for small values of α (meaning that there is a high density of data points within a small distance from each element of the optimal solution) then the distributed solution will be close to the optimal centralized one. In particular if we let $\alpha \rightarrow 0$, the distributed solution is guaranteed to be within a $1 - e^{-\kappa/k}$ factor from the optimal centralized solution. This situation naturally corresponds to very large datasets. In the following, we discuss more thoroughly this important scenario.

4.4 Performance Guarantees for Very Large Datasets

Suppose that our dataset is a finite sample V drawn i.i.d. from an underlying infinite set \mathcal{V} , according to some (unknown) probability distribution. Let $A^c[k]$ be an optimal solution in the infinite set, i.e., $A^c[k] = \arg \max_{S \subseteq \mathcal{V}} f(S)$, such that around each $e_i \in A^c[k]$, there is

a neighborhood of radius at least α^* where the probability density is at least β at all points (for some constants α^* and β). This implies that the solution consists of elements coming from reasonably dense and therefore representative regions of the dataset.

Let us suppose $g : \mathbb{R} \rightarrow \mathbb{R}$ is the growth function of the metric: $g(\alpha)$ is defined to be the volume of a ball of radius α centered at a point in the metric space. This means, for $e_i \in A^c[k]$ the probability of a random element being in $N_\alpha(e_i)$ is at least $\beta g(\alpha)$ and the expected number of α neighbors of e_i is at least $E[|N_\alpha(e_i)|] = n\beta g(\alpha)$. As a concrete example, Euclidean metrics of dimension D have $g(\alpha) = O(\alpha^D)$. Note that for simplicity we are assuming the metric to be homogeneous, so that the growth function is the same at every point. For heterogeneous spaces, we require g to have a uniform lower bound on the growth function at every point.

In these circumstances, the following theorem guarantees that if the dataset V is sufficiently large and f is λ -Lipschitz, then GREEDI produces a solution close to the centralized one.

Theorem 9 For $n \geq \frac{8km \log(k/\delta^{1/m})}{\beta g(\frac{\varepsilon}{\lambda k})}$, where $\frac{\varepsilon}{\lambda k} \leq \alpha^*$, if the algorithm GREEDI assigns elements uniformly randomly to m processors, then with probability at least $(1 - \delta)$,

$$f(A^{gd}[m, \kappa]) \geq (1 - e^{-\kappa/k})(f(A^c[k]) - \varepsilon).$$

The above theorem shows that for very large datasets, GREEDI provides a solution that is within a $1 - e^{-\kappa/k}$ factor of the optimal centralized solution. This result is based on the fact that for sufficiently large datasets, there is a suitably dense neighborhood around each member of the optimal solution. Thus, if the elements of the dataset are partitioned uniformly randomly to m processors, at least one partition contains a set $A_i^c[k]$ such that its elements are very close to the elements of the optimal centralized solution and provides a constant factor approximation of the optimal centralized solution.

4.5 Handling Decomposable Functions

So far, we have assumed that the objective function f is given to us as a black box, which we can evaluate for any given set S independently of the dataset V . In many settings, however, the objective f depends itself on the entire dataset. In such a setting, we cannot use GREEDI as presented above, since we cannot evaluate f on the individual machines without access to the full set V . Fortunately, many such functions have a simple structure which we call *decomposable*. More precisely, we call a submodular function f *decomposable* if it can be written as a sum of submodular functions as follows (Krause and Gomes, 2010):

$$f(S) = \frac{1}{|V|} \sum_{i \in V} f_i(S)$$

In other words, there is separate submodular function associated with every data point $i \in V$. We require that each f_i can be evaluated without access to the full set V . Note that the exemplar based clustering application we discussed in Section 3.4 is an instance of this framework, among many others. Let us define the evaluation of f restricted to $D \subseteq V$ as

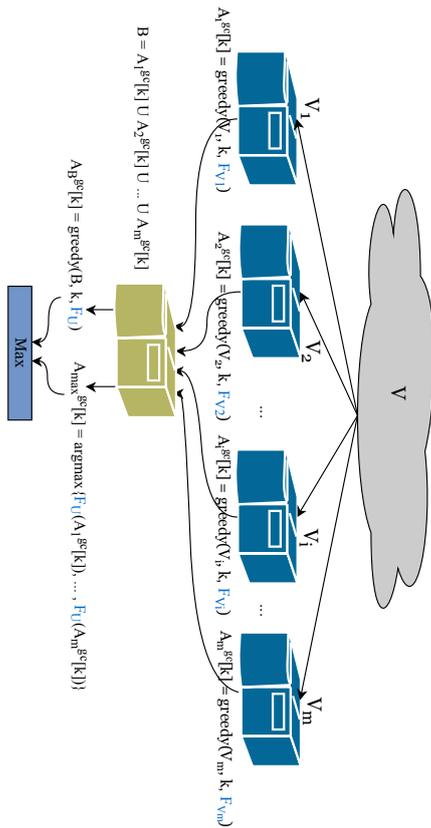


Figure 3: Illustration of our two-round algorithm GREEDI for decomposable functions

follows:

$$f_D(S) = \frac{1}{|D|} \sum_{i \in D} f_i(S)$$

In the remaining of this section, we show that assigning each element of the dataset randomly to a machine and running GREEDI will provide a solution that is with high probability close to the optimum solution. For this, let us assume that f_i 's are bounded, and without loss of generality $0 \leq f_i(S) \leq 1$ for $1 \leq i \leq |V|, S \subseteq V$. Similar to Section 4.3 we assume that GREEDI performs the partition by assigning elements uniformly at random to the machines. These machines then each greedily optimize f_i . The second stage of GREEDI optimizes f_U , where $U \subseteq V$ is chosen uniformly at random with size $|n/m|$.

Then, we can show the following result. First, for any fixed ϵ, m, k , let us define n_0 to be the smallest integer such that for $n \geq n_0$ we have $\ln(n)/n \leq \epsilon^2/(mk)$.

Theorem 10 For $n \geq \max(n_0, m \log(\delta/4m)/\epsilon^2)$, $\epsilon < 1/4$, and under the assumptions of Theorem 9, we have, with probability at least $1 - \delta$,

$$f(A^{gr}[m, \kappa]) \geq (1 - e^{-n/m}) (f(A^c[k]) - 2\epsilon).$$

The above result demonstrates why GREEDI performs well on decomposable submodular functions with massive data even when they are evaluated locally on each machine. We will report our experimental results on exemplar-based clustering in the next section.

4.6 Performance of GREEDI on Random Partitions Without Geometric Structure

Very recently, Barbosa et al. (2015) and Mirrokni and Zadimoghaddam (2015) proved that under random partitioning of the data among m machines, the expected utility of GREEDI will be only a constant factor away from the optimum.

Theorem 11 (Barbosa et al. (2015); Mirrokni and Zadimoghaddam (2015)) *If elements are assigned uniformly at random to the machines, and $\kappa = k$, GREEDI gives a constant factor approximation guarantee (in the average case) to the optimum centralized solution².*

$$\mathbb{E}[f(A^{gr}[m, k])] \geq \frac{1-1/e}{2} f(A^c[k]).$$

These results show that random partitioning of the data is sufficient to guarantee that GREEDI provides a constant factor approximation, irrespective of m and k , and without the requirement of any geometric structure. On the other hand, if geometric structure is present, the bounds from the previous sections can provide sharper approximation guarantees.

5. (Non-Monotone) Submodular Functions with General Constraints

In this section we show how GREEDI can be extended to handle 1) more general constraints, and 2) non-monotone submodular functions. More precisely, we consider the following optimization setting

$$\begin{aligned} & \text{Maximize } f(S) \\ & \text{Subject to } S \in \zeta. \end{aligned}$$

Here, we assume that the feasible solutions should be members of the constraint set $\zeta \subseteq 2^V$. The function $f(\cdot)$ is submodular but may not be monotone. By overloading the notation we denote the set that achieves the above constrained optimization problem by $A^c[\zeta]$. Throughout this section we assume that the constraint set ζ is hereditary, meaning that if $A \in \zeta$ then for any $B \subseteq A$ we also require that $B \in \zeta$. Cardinality constraints are obviously hereditary, so are all the examples we mention below.

5.1 Matroid Constraints

A matroid \mathcal{M} is a pair (V, \mathcal{I}) where V is a finite set (called the ground set) and $\mathcal{I} \subseteq 2^V$ is a family of subsets of V (called the independent sets) satisfying the following two properties:

- *Hereditary property:* $A \subseteq B \subseteq V$ and $B \in \mathcal{I}$ implies that $A \in \mathcal{I}$, i.e. every subset of an independent set is independent.
- *Augmentation property:* If $A, B \in \mathcal{I}$ and $|B| > |A|$, there is an element $e \in B \setminus A$ such that $A \cup \{e\} \in \mathcal{I}$.

2. In fact, Mirrokni and Zadimoghaddam (2015) proved a 0.27-approximation guarantee which is slightly worse than $(1 - 1/e)/2$.

Maximizing a submodular function subject to matroid constraints has found several applications in machine learning and data mining, ranging from content aggregation on the web (Abbassi et al., 2013) to viral marketing (Narayanan and Nanavati, 2012) and online advertising (Streeter et al., 2009).

One way to approximately maximize a monotone submodular function $f(S)$ subject to the constraint that each S is independent, i.e., $S \in \mathcal{I}$, is to use a generalization of the greedy algorithm. This algorithm, which starts with an empty set and in each iteration picks the feasible element with maximum benefit until there is no more element e such that $S \cup \{e\} \in \mathcal{I}$, is guaranteed to provide a $\frac{1}{2}$ -approximation of the optimal solution (Fisher et al., 1978). Recently, this bound has been improved to $(1 - 1/e)$ using the continuous greedy algorithm (Calinescu et al., 2011). For non-negative and non-monotone submodular functions with matroid constraints, the best known result is a 0.325-approximation based on simulated annealing (Gharan and Vondrák, 2011).

Curvature: For a submodular function f , the total curvature of f with respect to a set S is defined as:

$$c = 1 - \min_{j \in V} \frac{f(j|S \setminus j)}{f(j)}.$$

Intuitively, the notion of curvature determines how far away f is from being modular. In other words, it measures how much the marginal gain of an element w.r.t. set S can decrease as a function of S . In general, $c \in [0, 1]$, and for additive (modular) functions, $c = 0$, i.e., the marginal values are independent of S . In this case, the greedy algorithm returns the optimal solution to $\max\{f(S) : S \in \mathcal{I}\}$. In general, the greedy algorithm gives a $\frac{1}{1+c}$ -approximation to maximizing a non-decreasing submodular function with curvature c subject to a matroid constraint (Conforti and Cornuéjols, 1984). In case of the uniform matroid $\mathcal{I} = \{S : |S| \leq k\}$, the approximation factor is $(1 - e^{-c})/c$.

Intersection of Matroids: A more general case is when we have p matroids $\mathcal{M}_1 = (V, \mathcal{I}_1), \mathcal{M}_2 = (V, \mathcal{I}_2), \dots, \mathcal{M}_p = (V, \mathcal{I}_p)$ on the same ground set V , and we want to maximize the submodular function f on the intersection of p matroids. That is, $\mathcal{I} = \bigcap_i \mathcal{I}_i$ consists of all subsets of V that are independent in all p matroids. This constraint arises, e.g., when optimizing over rankings (which can be modeled as intersections of two partition matroids). Another recent application considered is finding the influential set of users in viral marketing when multiple products need to be advertised and each user can tolerate only a small number of recommendations (Du et al., 2013). For p matroid constraints, the $\frac{1}{p+1}$ -approximation provided by the greedy algorithm (Fisher et al., 1978) has been improved to a $(\frac{1}{p} - \epsilon)$ -approximation for $p \geq 2$ by Lee et al. (2009b). For the non-monotone case, a $1/(p + 2 + 1/p + \epsilon)$ -approximation based on local search is also given by Lee et al. (2009b).

p -systems: p -independence systems generalize constraints given by the intersection of p matroids. Given an independence family \mathcal{I} and a set $V' \subseteq V$, let $S(V')$ denote the set of maximal independent sets of \mathcal{I} included in V' , i.e., $S(V') = \{A \in \mathcal{I} \mid \forall e \in V' \setminus A : A \cup \{e\} \notin \mathcal{I}\}$. Then we call (V, \mathcal{I}) a p -system if for all nonempty $V' \subseteq V$ we have

$$\max_{A \in S(V')} |A| \leq p \cdot \min_{A \in S(V')} |A|.$$

Similar to p matroid constraints, the greedy algorithm provides a $\frac{1}{p+1}$ -approximation guarantee for maximizing a monotone submodular function subject to a p -systems constraint (Fisher et al., 1978). For the non-monotone case, Gupta et al. (2010) provided a $p/((p + 1)(3p + 3))$ -approximation can be achieved by combining an algorithm of Gupta et al. (2010) with the result for unconstrained submodular maximization of Buchbinder et al. (2012). This result has been recently tightened to $p/((p + 1)(2p + 1))$ by Mirzasoleiman et al. (2016).

5.2 Knapsack Constraints

In many applications, including feature and variable selection in probabilistic models (Krause and Guestrin, 2005a) and document summarization (Lin and Bilmes, 2011), elements $e \in V$ have non-uniform costs $c(e) > 0$, and we wish to find a collection of elements S that maximize f subject to the constraint that the total cost of elements in S does not exceed a given budget \mathcal{R} , i.e.

$$\max_S f(S) \text{ s.t. } \sum_{v \in S} c(v) \leq \mathcal{R}.$$

Since the simple greedy algorithm ignores cost while iteratively adding elements with maximum marginal gains according (see Eq. 2) until $|S| \leq \mathcal{R}$, it can perform arbitrary poorly. However, it has been shown that taking the maximum over the solution returned by the greedy algorithm that works according to Eq. 2 and the solution returned by the modified greedy algorithm that optimizes the cost-benefit ratio

$$v^* = \arg \max_{\substack{e \in V \setminus S \\ c(e) \leq \mathcal{R} - c(S)}} \frac{f(S \cup \{e\}) - f(S)}{c(v)},$$

provides a $(1 - 1/\sqrt{\epsilon})$ -approximation of the optimal solution (Krause and Guestrin, 2005b). Furthermore, a more computationally expensive algorithm which starts with all feasible solutions of cardinality 3 and augments them using the cost-benefit greedy algorithm to find the set with maximum value of the objective function provides a $(1 - 1/\epsilon)$ -approximation (Sviridenko, 2004). For maximizing non-monotone submodular functions subject to knapsack constraints, a $(1/5 - \epsilon)$ -approximation algorithm based on local search was given by Lee et al. (2009a).

Multiple Knapsack Constraints: In some applications such as procurement auctions (Garg et al., 2001), video-on-demand systems and e-commerce (Kulik et al., 2009), we have a d -dimensional budget vector \mathcal{R} and a set of element $e \in V$ where each element is associated with a d -dimensional cost vector. In this setting, we seek a subset of elements $S \subseteq V$ with a total cost of at most \mathcal{R} that maximizes a non-decreasing submodular function f . Kulik et al. (2009) proposed a two-phase algorithm that provides a $(1 - 1/e - \epsilon)$ -approximation for the problem by first guessing a constant number of elements of highest value, and then taking the value residual problem with respect to the guessed subset. For the non-monotone case, Lee et al. (2009a) provided a $(1/5 - \epsilon)$ -approximation based on local search.

p -system and d knapsack constraints: A more general type of constraint that has recently found interesting applications in viral marketing (Du et al., 2013) and personalized data summarization Mirzasoleiman et al. (2016) which can be cast by combining a

Constraint	monotone submodular functions	Approximation (τ)	non-monotone submodular functions
Cardinality	$1 - 1/e$ (Fisher et al., 1978)	0.325 (Gharan and Vondrák, 2011)	
1 matroid	$1 - 1/e$ (Ghahesou et al., 2011)	0.325 (Gharan and Vondrák, 2011)	
p matroid	$1/p - \epsilon$ (Lee et al., 2009b)	$1/(p + 2 + 1/p + \epsilon)$ (Lee et al., 2009b)	
1 knapsack	$1 - 1/e$ (Sviridenko, 2004)	$1/5 - \epsilon$ (Lee et al., 2009a)	
d knapsack	$1 - 1/e - \epsilon$ (Kuhlé et al., 2009)	$1/5 - \epsilon$ (Lee et al., 2009a)	
p -system	$1/(p + 1)$ (Fisher et al., 1978)	$p/((p + 1)(2p + 1))$ (Mirzasoleiman et al., 2016)	
p -system + d knapsack	$1/(p + 2d + 1)$ (Bardandiyuru and Vondrák, 2014)	$(1 + \epsilon)/(p + 1)(2p + 2d + 1)/p$ (Mirzasoleiman et al., 2016)	

Table 1: Approximation guarantees (τ) for monotone and non-monotone submodular maximization under different constraints.

p -system with d knapsack constraints. For maximizing a monotone submodular function Bardandiyuru and Vondrák (2014) proposed a modified version of the greedy algorithm that guarantees a $1/(p + 2d + 1)$ -approximation. By combining this algorithm with the one proposed in (Gupta et al., 2010), Mirzasoleiman et al. (2016) provided a fast algorithm for maximizing a non-monotone submodular function subject to a p -system and d knapsack constraints with $(1 + \epsilon)/(p + 1)(2p + 2d + 1)/p$ -approximation.

Table 1 summarizes the approximation guarantees for monotone and non-monotone submodular maximization under different constraints.

5.3 GREEDI APPROXIMATION GUARANTEE UNDER MORE GENERAL CONSTRAINTS

Assume that we have a set of constraints $\zeta \subseteq 2^V$ that is hereditary. Further assume we have access to a "black box" algorithm X that gives us a constraint factor approximation guarantee for maximizing a non-negative (but not necessarily monotone) submodular function f subject to ζ , i.e.

$$X : (f, \zeta) \rightarrow A^X \in \zeta \text{ s.t. } f(A^X[\zeta]) \geq \tau \max_{A \in \zeta} f(A). \quad (8)$$

We can modify GREEDI to use any such approximation algorithm as a black box, and provide theoretical guarantees about the solution. In order to process a large dataset, it first distributes the ground set over m machines. Then instead of greedily selecting elements, each machine i -in parallel—separately runs the black box algorithm X on its local data in order to produce a feasible set $A_i^X[\zeta]$ meeting the constraints ζ . We denote by $A_{\max}^{\text{loc}}[\zeta]$ the set with maximum value among $A_i^X[\zeta]$. Next, the solutions are merged: $B = \cup_{i=1}^m A_i^X[\zeta]$, and the black box algorithm is applied one more time to set B to produce a solution $A_B^{\text{loc}}[\zeta]$. Then, the distributed solution for parameter m and constraints ζ , $A^{Xd}[m, \zeta]$, is the best among $A_{\max}^{\text{loc}}[\zeta]$ and $A_B^{\text{loc}}[\zeta]$. This procedure is given in more detail in Algorithm 3.

The following result generalizes Theorem 4 for maximizing a submodular function subject to more general constraints.

Algorithm 3 GREEDI under General Constraints

Input: Set V , #of partitions m , constraints ζ , submodular function f .

Output: Set $A^{Xd}[m, \zeta]$.

- 1: Partition V into m sets V_1, V_2, \dots, V_m .
- 2: In parallel: Run the approximation algorithm X on each set V_i to find a solution $A_i^X[\zeta]$.
- 3: Find $A_{\max}^{\text{loc}}[\zeta] = \arg \max_A \{f(A) \mid A \in \{A_1^X[\zeta], \dots, A_m^X[\zeta]\}\}$.
- 4: Merge the resulting sets: $B = \cup_{i=1}^m A_i^X[\zeta]$.
- 5: Run the approximation algorithm X on B to find a solution $A_B^{\text{loc}}[\zeta]$.
- 6: Return $A^{Xd}[m, \zeta] = \arg \max\{A_{\max}^{\text{loc}}[\zeta], A_B^{\text{loc}}[\zeta]\}$.

Theorem 12 *Let f be a non-negative submodular function and X be a black box algorithm that provides a τ -approximation guarantee for submodular maximization subject to a set of hereditary constraints ζ . Then*

$$f(A^{Xd}[m, \zeta]) \geq \frac{\tau}{\min(m, \rho([\zeta]))} f(A^c[\zeta]),$$

where $f(A^c[\zeta])$ is the optimum centralized solution, and $\rho([\zeta]) = \max_{A \in \zeta} |A|$.

Specifically, for submodular maximization subject to the matroid constraint \mathcal{M} , we have $\rho(A \in \mathcal{I}) = r_{\mathcal{M}}$ where $r_{\mathcal{M}}$ is the rank of the matroid (i.e., the maximum size of any independent set in the system). For submodular maximization subject to the knapsack constraint \mathcal{R} , we can bound $\rho([\zeta] \leq \mathcal{R})$ by $\lceil \mathcal{R} / \min_{v \in \mathcal{V}} c(v) \rceil$ (i.e. the capacity of the knapsack divided by the smallest weight of any element).

Performance on Datasets with Geometric Structure. When the submodular function $f(\cdot)$ and the constraint set ζ have more structure, then we can provide much better approximation guarantees. Assuming the elements of V are embedded in metric space with distance $d : V \times V \rightarrow \mathbb{R}^+$, we say that ζ is *locally replaceable* with respect to a set $S \subseteq V$ with parameter $\alpha > 0$ if

$$\forall S' \subseteq V \text{ s.t. } |S'| = |S| \text{ and } d_{\infty}(S, S') \leq \alpha \Rightarrow S' \in \zeta.$$

Here, we define the distance d_{∞} between two sets S and S' of the same size k as follows. Let M be the set of all possible matchings between S and S' , i.e.,

$$M = \{((e_1, e'_1), \dots, (e_k, e'_k)) \text{ s.t. } e_i \in S \text{ and } e'_i \in S' \text{ for } 1 \leq i \leq k\}.$$

Then $d_{\infty}(S, S') = \min_M \max_i d(e_i, e'_i)$. We require locality only with respect to $A^c[\zeta]$ to ensure that the optimum solution can be well approximated. What the locally replaceable property requires is that as elements of $A^c[\zeta]$ get replaced by nearby elements, the resulting set is also a feasible solution. Combining this property with λ -Lipschitzness will provide us with the following theorem.

Theorem 13 *Under the conditions that 1) elements are assigned uniformly at random to m machines, 2) for each $e_i \in A^c[\zeta]$ we have $|\mathcal{N}_{\alpha}(e_i)| \geq \rho([\zeta])m \log(\rho([\zeta])/\delta^{1/m})$, 3) $f(\cdot)$ is*

λ -Lipschitz, and 4) ζ is locally replaceable with respect to $A^c[\zeta]$ with parameter α , then with probability at least $(1 - \delta)$,

$$f(A^{Xd}[m, \zeta]) \geq \tau(f(A^c[\zeta]) - \lambda\alpha\rho(\zeta)).$$

The above result generalizes Theorem 8 for maximizing non-negative submodular functions subject to different constraints.

Performance Guarantee for Very Large Datasets. Similarly, we can generalize Theorem 9 for maximizing non-negative submodular functions subject to more general constraints. Suppose that our dataset is a finite sample V drawn i.i.d. from an underlying infinite set \mathcal{V} , according to some (unknown) probability distribution. Let $A^c[\zeta]$ be an optimal solution in the infinite set, i.e., $A^c[\zeta] = \arg \max_{S \subseteq \mathcal{V}} f(S)$, such that around each $e_i \in A^c[\zeta]$, there is a neighborhood of radius at least α^* where the probability density is at least β at all points (for some constants α^* and β). Recall that $g: \mathbb{R} \rightarrow \mathbb{R}$ is the growth function where $g(\alpha)$ measures the volume of a ball of radius α centered at a point in the metric space.

Theorem 14 For $n \geq \frac{8\rho(\zeta)m \log(\rho(\zeta)/\delta^{1/m})}{\beta g(\frac{\epsilon}{\lambda\rho(\zeta)})}$, where $\frac{\epsilon}{\lambda\rho(\zeta)} \leq \alpha^*$, if GREEDI assigns elements uniformly at random to m processors and under the conditions that f is λ -Lipschitz, and ζ is locally replaceable with respect to $A^c[\zeta]$ with parameter α^* , then with probability at least $(1 - \delta)$, we have

$$f(A^{Xd}[m, \zeta]) \geq \tau(f(A^c[\zeta]) - \epsilon).$$

Performance Guarantee for Decomposable Functions. For the case of decomposable functions described in Section 4.5, the following generalization of Theorem 10 holds for maximizing a non-negative submodular function subject to more general constraints. Let us define n_0 to be the smallest integer such that for $n \geq n_0$ we have $\ln(n)/n \leq \epsilon^2/(m \cdot \rho(\zeta))$.

Theorem 15 For $n \geq \max(n_0, m \log(\delta/4m)/\epsilon^2)$, $\epsilon < 1/4$, and under the assumptions of Theorem 14, we have, with probability at least $1 - \delta$,

$$f(A^{Xd}[m, \zeta]) \geq \tau(f(A^c[\zeta]) - 2\epsilon).$$

6. Experiments

In our experimental evaluation we wish to address the following questions: 1) how well does GREEDI perform compared to the centralized solution, 2) how good is the performance of GREEDI when using decomposable objective functions (see Section 4.5), and finally 3) how well does GREEDI scale in the context of massive datasets. To this end, we run GREEDI on three scenarios: exemplar based clustering, active set selection in GPs and finding the maximum cuts in graphs.

We compare the performance of our GREEDI method to the following naive approaches:

- *random/random*: in the first round each machine simply outputs k randomly chosen elements from its local data points and in the second round k out of the merged mk elements, are again randomly chosen as the final output.

- *random/greedy*: each machine outputs k randomly chosen elements from its local data points, then the standard greedy algorithm is run over mk elements to find a solution of size k .

- *greedy/merge*: in the first round k/m elements are chosen greedily from each machine and in the second round they are merged to output a solution of size k .

- *greedy/max*: in the first round each machine greedily finds a solution of size k and in the second round the solution with the maximum value is reported.

For GREEDI, we let each of the m machines select a set of size αk , and select a final solution of size k among the union of the m solutions (i.e., among αkm elements). We present the performance of GREEDI for different parameters $\alpha > 0$. For datasets where we are able to find the centralized solution, we report the ratio of $f(A_{\text{dist}}[k])/f(A^c[k])$, where $A_{\text{dist}}[k]$ is the distributed solution (in particular $A^{\text{gd}}[m, \alpha k, k] = A_{\text{dist}}[k]$ for GREEDI).

6.1 Exemplar Based Clustering

Our exemplar based clustering experiment involves GREEDI applied to the clustering utility $f(S)$ (see Sec. 3.4) with $d(x, x') = \|x - x'\|^2$. We performed our experiments on a set of 10,000 *Tiny Images* (Torralba et al., 2008). Each 32 by 32 RGB pixel image was represented by a 3,072 dimensional vector. We subtracted from each vector the mean value, normalized it to unit norm, and used the origin as the auxiliary exemplar. Fig. 4a compares the performance of our approach to the benchmarks with the number of exemplars set to $k = 50$, and varying number of partitions m . It can be seen that GREEDI significantly outperforms the benchmarks and provides a solution that is very close to the centralized one. Interestingly, even for very small $\alpha = \kappa/k < 1$, GREEDI performs very well. Since the exemplar based clustering utility function is decomposable, we repeated the experiment for the more realistic case where the function evaluation in each machine was restricted to the local elements of the dataset in that particular machine (rather than the entire dataset). Fig 4b shows similar qualitative behavior for decomposable objective functions.

Large scale experiments with Hadoop. As our first large scale experiment, we applied GREEDI to the whole dataset of 80,000,000 *Tiny Images* (Torralba et al., 2008) in order to select a set of 64 exemplars. Our experimental infrastructure was a cluster of 10 quad-core machines running Hadoop with the number of reducers set to $m = 8000$. Hereby, each machine carried out a set of reduce tasks in sequence. We first partitioned the images uniformly at random to reducers. Each reducer separately performed the lazy greedy algorithm on its own set of 10,000 images ($\approx 123\text{MB}$) to extract 64 images with the highest marginal gains w.r.t. the local elements of the dataset in that particular partition. We then merged the results and performed another round of lazy greedy selection on the merged results to extract the final 64 exemplars. Function evaluation in the second stage was performed w.r.t. a randomly selected subset of 10,000 images from the entire dataset. The maximum running time per reduce task was 2.5 hours. As Fig. 5a shows, GREEDI highly outperforms the other distributed benchmarks and can scale well to very large datasets. Fig. 5b shows a set of cluster exemplars discovered by GREEDI where Fig. 5c and Fig. 5d show 100 nearest images to exemplars 26 and 63 (shown with red borders) in Fig. 5b.

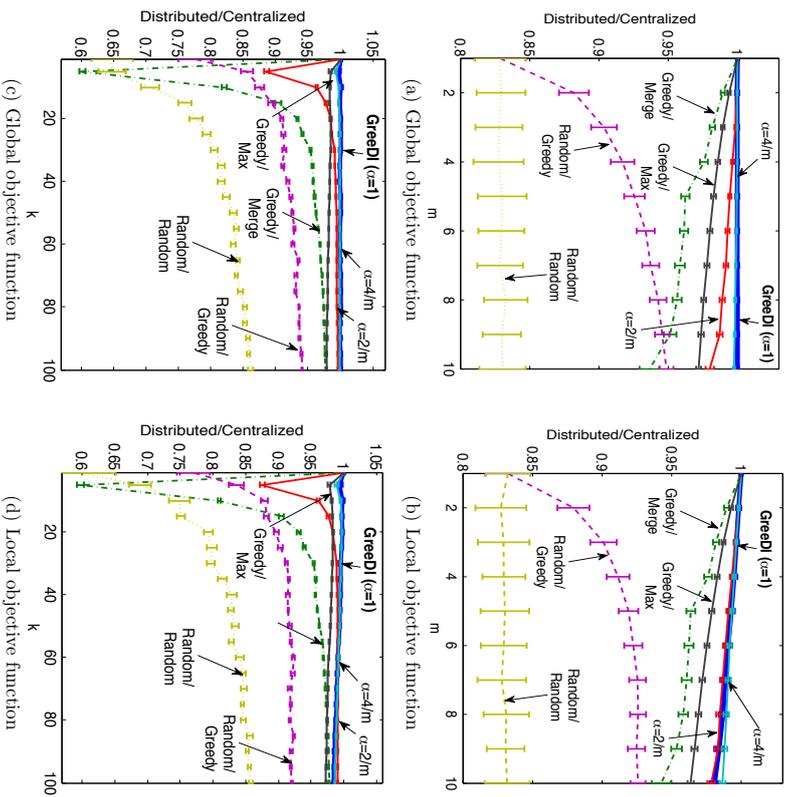


Figure 4: Performance of GREEDI compared to the other benchmarks. a) and b) show the mean and standard deviation of the ratio of distributed vs. centralized solution for global and local objective functions with budget $k = 50$ and varying the number m of partitions. c) and d) show the same ratio for global and local objective functions for $m = 5$ partitions and varying budget k , for a set of 10,000 *Tiny Images*.

6.2 Active Set Selection

Our active set selection experiment involves GREEDI applied to the information gain $f(S)$ (see Sec. 3.4) with Gaussian kernel, $h = 0.75$ and $\sigma = 1$. We used the *Parkinsons Telemor-toring* dataset (Tsanas et al., 2010) consisting of 5,875 bio-medical voice measurements with 22 attributes from people with early-stage Parkinson’s disease. We normalized the vectors to zero mean and unit norm. Fig. 6b compares the performance GREEDI to the

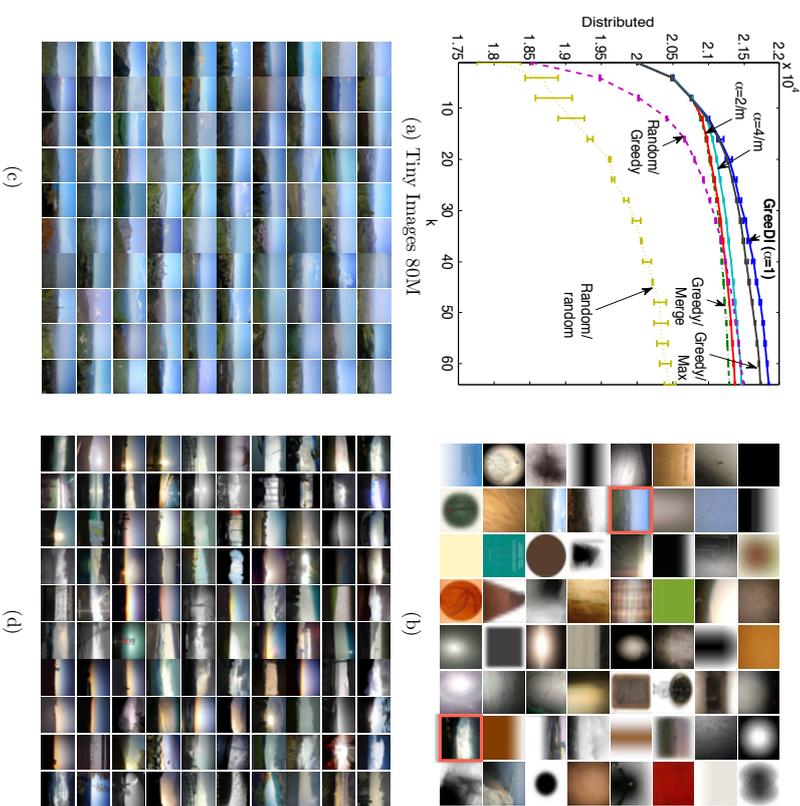


Figure 5: Performance of GREEDI compared to the other benchmarks. a) shows the distributed solution with $m = 8000$ and varying k for local objective functions on the whole dataset of 80,000,000 *Tiny Images*. b) shows a set of cluster exemplars discovered by GREEDI, and each column in c) shows 100 images nearest to exemplars 26 and d) shows 100 images nearest to exemplars 63 in b).

benchmarks with fixed $k = 50$ and varying number of partitions m . Similarly, Fig 6a shows the results for fixed $m = 10$ and varying k . We find that GREEDI significantly outperforms the benchmarks.

Large scale experiments with Hadoop. Our second large scale experiment consists of 45,811,888 user visits from the Featured Tab of the Today Module on Yahoo! Front Page (web, 2012). For each visit, both the user and each of the candidate articles are associated with a feature vector of dimension 6. Here, we used the normalized user features. Our experimental setup was a cluster of 8 quad-core machines running Spark with the number

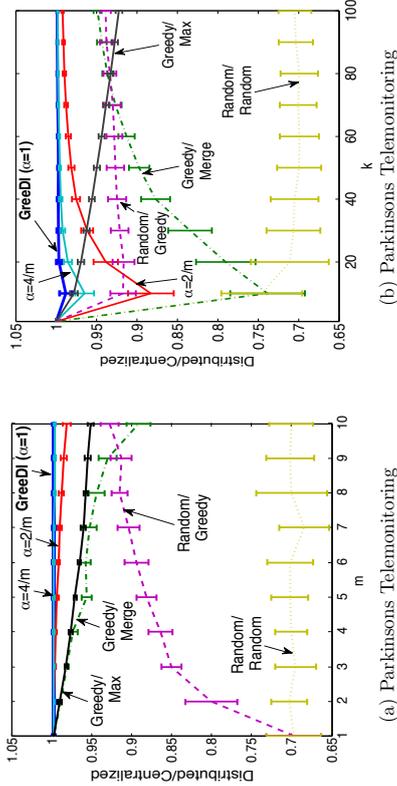


Figure 6: Performance of GREEDI compared to the other benchmarks. a) shows the ratio of distributed vs. centralized solution with $k = 50$ and varying m for *Parkinson's Telemonitoring*. b) shows the same ratio with $m = 10$ and varying k on the same dataset.

of reducers set to $m = 32$. Each reducer performed the lazy greedy algorithm on its own set of $\approx 1,431,621$ vectors ($\approx 34\text{MB}$) in order to extract 256 elements with the highest marginal gains w.r.t the local elements of the dataset in that particular partition. We then merged the results and performed another round of lazy greedy selection on the merged results to extract the final active set of size 256. The maximum running time per reduce task was 12 minutes for selecting 128 elements and 48 minutes for selecting 256 elements. Fig. 7 shows the performance of GREEDI compared to the benchmarks. We note again that GREEDI significantly outperforms the other distributed benchmarks and can scale well to very large datasets.

Performance Comparison. Fig. 8 shows the speedup of GREEDI compared to the centralized greedy benchmark for different values of k and varying number of partitions m . As Fig. 8a shows, for small values of m , the speedup is almost linear in the number of machines. However, for large values of m the running time of the second stage of GREEDI increases and ultimately dominates the whole running time. Hence, we do not observe a linear speedup anymore. This effect can be observed in Fig. 8b. For larger values of k , the speedup is higher on fewer machines, but decreases more quickly by increasing m , as the second stage takes longer to complete.

6.3 Non-Monotone Submodular Function (Finding Maximum Cuts)

We also applied GREEDI to the problem of finding maximum cuts in graphs. In our setting we used a *Facebook-like social network* (Opsahl and Panzarasa, 2009). This dataset includes the users that have sent or received at least one message in an online student community at University of California, Irvine and consists of 1,899 users and 20,296 directed ties. Fig. 9a and 9b show the performance of GREEDI applied to the cut function on graphs. We evaluated the objective function locally on each partition. Thus, the links

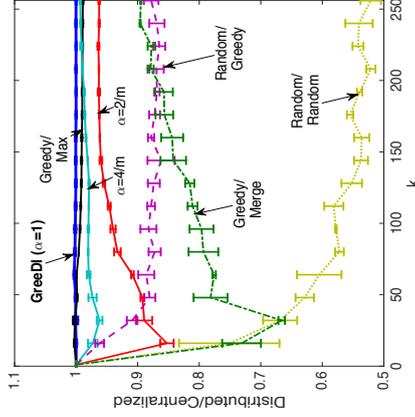


Figure 7: Performance of GREEDI with $m = 32$ and varying budget k compared to the other benchmarks on *Yahoo! Webscope data*.

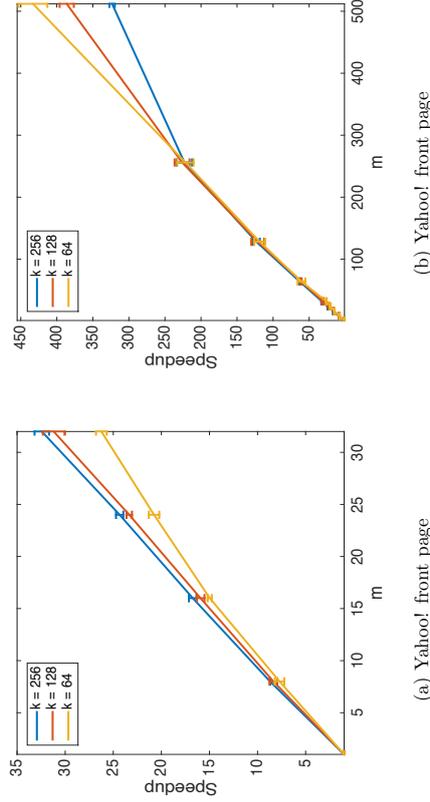


Figure 8: Running time of GREEDI compared to the centralized greedy algorithm. a) shows the ratio of centralized vs. distributed solution with $k = 64, 128, 256$ and up to $m = 32$ machines for *Yahoo Webscope data*. b) shows the same ratio with $k = 64, 128, 256$ and up to $m = 512$ machines on the same dataset. Both experiments are performed on a cluster of 8 quad core machines.

between the partitions are disconnected. Since the problem of finding the maximum cut in a graph is non-monotone submodular, we applied the RandomGreedy algorithm proposed by Buchbinder et al. (2014) to find the near optimal solution in each partition.

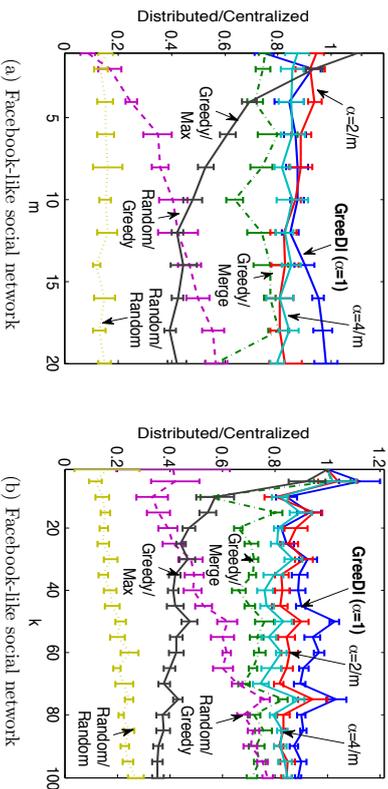


Figure 9: Performance of GREEDI compared to the other benchmarks. a) shows the mean and standard deviation of the ratio of distributed to centralized solution for budget $k = 20$ with varying number of machines m and b) shows the same ratio for varying budget k with $m = 10$ on *Facebook-like social network*.

Although the cut function does not decompose additively over individual data points, perhaps surprisingly, GREEDI still performs very well, and significantly outperforms the benchmarks. This suggests that our approach is quite robust, and may be more generally applicable.

6.4 Comparison with Greedy Scaling.

Kumar et al. (2013) recently proposed an alternative approach—GREEDYSCALING—for parallel maximization of submodular functions. GREEDYSCALING is a randomized algorithm that carries out a number (typically less than k) rounds of MapReduce computations. We applied GREEDI to the submodular coverage problem in which given a collection V of sets, we would like to pick at most k sets from V in order to maximize the size of their union. We compared the performance of our GREEDI algorithm to the reported performance of GREEDYSCALING on the same datasets, namely *Accidents* (Gauts et al., 2003) and *Kosarak* (Bodon, 2012). As Fig 10a and 10b shows, GREEDI outperforms GREEDYSCALING on the *Accidents* dataset and its performance is comparable to that of GREEDYSCALING in the *Kosarak* dataset.

7. Conclusion

We have developed an efficient distributed protocol GREEDI, for constrained submodular maximization. We have theoretically analyzed the performance of our method and showed that under certain natural conditions it performs very close to the centralized (albeit impractical in massive datasets) solution. We have also demonstrated the effectiveness of our approach through extensive experiments, including active set selection in GPs on a dataset

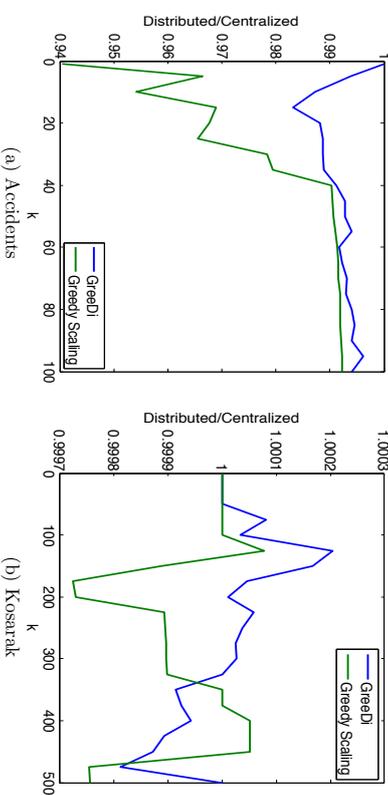


Figure 10: Performance of GREEDI compared to the GreedyScaling algorithm of Kumar et al. (2013) (as reported in their paper). a) shows the ratio of distributed to centralized solution on *Accidents* dataset with 340,183 elements and b) shows the same ratio for *Kosarak* dataset with 990,002 elements. The results are reported for varying budget k and varying number of machines $m = n/\mu$ where $\mu = O(kn^\delta \log n)$ and n is the size of the dataset. The results are reported for $\delta = 1/2$. Note that the results presented by Kumar et al. (2013) indicate that GreedyScaling generally requires a substantially larger number of MapReduce rounds compared to GREEDI.

of 45 million examples, and exemplar based summarization of a collection of 80 million images using Hadoop. We believe our results provide an important step towards solving submodular optimization problems in very large scale, real applications.

Acknowledgments

This research was supported by SNE 200021-137971, DARPA MSEE FA8650-11-1-7156, ERC StG 307036, a Microsoft Faculty Fellowship, an ETH Fellowship, Google Research Faculty Award, and a Scottish Informatics and Computer Science Alliance.

Appendix A. Proofs

This section presents the complete proofs of theorems presented in the article.

A.1 Proof of Theorem 3

⇒ direction:

The proof easily follows from the following lemmas.

Lemma 16 $\max_i f(A_i^c[k]) \geq \frac{1}{m} f(A^c[k])$.

Proof Let B_i be the elements in V_i that are contained in the optimal solution, $B_i = A_i^c[k] \cap V_i$. Then we have:

$$f(A^c[k]) = f(B_1 \cup \dots \cup B_m) = f(B_1) + f(B_2|B_1) + \dots + f(B_m|B_{m-1}, \dots, B_1).$$

Using submodularity of f , for each $i \in \{1 \dots m\}$, we have

$$f(B_i|B_{i-1} \dots B_1) \leq f(B_i),$$

and thus,

$$f(A^c[k]) \leq f(B_1) + \dots + f(B_m).$$

Since, $f(A_i^c[k]) \geq f(B_i)$, we have

$$f(A^c[k]) \leq f(A_1^c[k]) + \dots + f(A_m^c[k]).$$

Therefore,

$$f(A^c[k]) \leq m \max_i f(A_i^c[k]).$$

Lemma 17 $\max_i f(A_i^c[k]) \geq \frac{1}{k} f(A^c[k])$.

Proof Let $f(A^c[k]) = f(\{u_1, \dots, u_k\})$. Using submodularity of f , we have

$$f(A^c[k]) \leq \sum_{i=1}^k f(u_i).$$

Thus, $f(A^c[k]) \leq k f(u^*)$ where $u^* = \arg \max_i f(u_i)$. Suppose that the element with highest marginal gain (i.e., u^*) is in V_j . Then the maximum value of f on V_j would be greater or equal to the marginal gain of u^* , i.e., $f(A_j^c[k]) \geq f(u^*)$ and since $f(\max_i f(A_i^c[k])) \geq f(A_j^c[k])$, we can conclude that

$$f(\max_i f(A_i^c[k])) \geq f(u^*) \geq \frac{1}{k} f(A^c[k]).$$

Since $f(A^d[m, k]) \geq \max_i f(A_i^c[k])$; from Lemma 16 and 17 we have

$$f(A^d[m, k]) \geq \frac{1}{\min(m, k)} f(A^c[k]).$$

⇐ direction:

Let us consider a set of unbiased and independent Bernoulli random variables $X_{i,j}$ for $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, k\}$, i.e., $\Pr(X_{i,j} = 1) = \Pr(X_{i,j} = 0) = 1/2$ and $(X_{i,j} \perp X_{i',j'})$ if $i \neq i'$ or $j \neq j'$. Let us also define $Y_i = (X_{i,1}, \dots, X_{i,k})$ for $i \in \{1, \dots, m\}$. Now assume that $V_i = \{X_{i,1}, \dots, X_{i,k}, Y_i\}$, $V = \bigcup_{i=1}^m V_i$ and $f(S) = H(S)$, where H is the entropy of the subset S of random variables. Note that H is a monotone submodular function. It is easy to see that $A_i^c[k] = \{X_{i,1}, \dots, X_{i,k}\}$ or $A_i^c[k] = Y_i$ as in both cases $H(A_i^c[k]) = k$. If we assume $A_i^c[k] = \{X_{i,1}, \dots, X_{i,k}\}$, then $B = \{X_{i,j} | 1 \leq i \leq m, 1 \leq j \leq k\}$. Hence, by selecting at most k elements from B , we have $H(A^d[m, k]) = k$. On the other hand, the set of k elements that maximizes the entropy is $\{Y_1, \dots, Y_m\}$. Note that $H(Y_i) = k$ and $Y_i \perp Y_j$ for $i \neq j$. Hence, $H(A^c) = k \cdot m$ if $m \geq k$ or otherwise $H(A^c[k]) = k^2$.

A.2 Proof of Theorem 4

Let us first mention a slight generalization over the performance of the standard greedy algorithm. It follows easily from the argument in (Nemhauser et al., 1978).

Lemma 18 *Let f be a non-negative submodular function, and let $A^{gc}[q]$ of cardinality q be the greedy selected set by the standard greedy algorithm. Then,*

$$f(A^{gc}[q]) \geq \left(1 - e^{-\frac{q}{k}}\right) f(A^c[k]).$$

By Lemma 18 we know that

$$f(A_i^{gc}[k]) \geq (1 - \exp(-\kappa/k)) f(A_i^c[k]).$$

Now, let us define

$$B^{gc} = \bigcup_{i=1}^m A_i^{gc}[k],$$

$$A_{\max}^{gc}[k] = \max_i f(A_i^{gc}[k]),$$

$$\bar{A}[k] = \arg \max_{S \subseteq B^{gc}, |S| \leq \kappa} f(S).$$

Then by using Lemma 18 again, we obtain

$$\begin{aligned} f(A^{gd}[m, \kappa]) &\geq \max \{ f(A_{\max}^{gc}[k]), (1 - \exp(-\kappa/\kappa)) f(\bar{A}[k]) \} \\ &\geq \frac{(1 - \exp(-\kappa/k))}{\min(m, k)} f(A^c[k]). \end{aligned}$$

A.3 Proof of Proposition 6

Let K be a positive definite kernel matrix defined in section 3.4.1. If we replace a point $e_i \in S$ with another point $e'_i \in V \setminus S$, the corresponding row and column i in the modified kernel matrix K' will be changed. W.l.o.g assume that we replace the first element $e_1 \in S$ with another element $e'_1 \in V \setminus S$, i.e., $\Delta K = K' - K$ has the following form with non-zero entries only on the first row and first column,

$$\Delta K \equiv K' - K \leq \begin{pmatrix} a_1 & a_2 & \dots & a_k \\ a_2 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_k & 0 & \dots & 0 \end{pmatrix}.$$

Note that kernel is Lipschitz continuous with constant \mathcal{L} , hence we have $|a_i| \leq \mathcal{L}d(e_1, e'_1)$ for $1 \leq i \leq k$. Then the absolute value of the change in the objective function would be

$$\begin{aligned}
 |f(S) - f(S')| &= \left| \frac{1}{2} \log \det(\mathbf{I} + K') - \frac{1}{2} \log \det(\mathbf{I} + K) \right| \\
 &= \frac{1}{2} \left| \log \frac{\det(\mathbf{I} + K')}{\det(\mathbf{I} + K)} \right| \\
 &= \frac{1}{2} \left| \log \frac{\det(\mathbf{I} + K + \Delta K)}{\det(\mathbf{I} + K)} \right| \\
 &= \frac{1}{2} \left| \log[\det(\mathbf{I} + K + \Delta K) \cdot \det(\mathbf{I} + K)^{-1}] \right| \\
 &= \frac{1}{2} \left| \log \det(\mathbf{I} + \Delta K(\mathbf{I} + K)^{-1}) \right|. \tag{9}
 \end{aligned}$$

Note that since K is positive-definite, $\mathbf{I} + K$ is an invertible matrix. Furthermore, since ΔK and K are symmetric matrices they both have k real eigenvalues. Therefore, $(\mathbf{I} + K)^{-1}$ has k eigenvalues $\lambda_i = \frac{1}{1 + \lambda_i} \leq 1$, for $1 \leq i \leq k$, where $\lambda_1' \cdots \lambda_k'$ are (non-negative) eigenvalues of kernel matrix K .

Now, we bound the maximum eigenvalues of ΔK and $\Delta K(\mathbf{I} + K)^{-1}$ respectively. Consider vectors $x, x' \in \mathbb{R}^n$, such that $\|x\|_2 = \|x'\|_2 = 1$. We have,

$$\begin{aligned}
 |x^T \Delta K x'| &= \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{pmatrix}^T \begin{pmatrix} a_1 & a_2 & \cdots & a_k \\ a_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_k & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} x'_1 \\ x'_2 \\ \vdots \\ x'_k \end{pmatrix} \\
 &= \begin{pmatrix} x_1 & x_2 & \cdots & x_k \\ x_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ x_k & 0 & \cdots & 0 \end{pmatrix}^T \begin{pmatrix} \sum_{i=1}^k a_i x_i x'_i \\ a_2 x'_1 \\ \vdots \\ a_k x'_1 \end{pmatrix} \\
 &= |x_1| \sum_{i=1}^k a_i x_i^2 + x'_1 \sum_{i=2}^k a_i x_i \\
 &= |x_1| \cdot \left| \sum_{i=1}^k a_i x_i^2 \right| + |x'_1| \cdot \left| \sum_{i=2}^k a_i x_i \right| \\
 &= |x_1| \cdot \sum_{i=1}^k |a_i x_i^2| + |x'_1| \cdot \sum_{i=2}^k |a_i x_i| \\
 &\leq 2k\mathcal{L}d(e_1, e'_1), \tag{10}
 \end{aligned}$$

where we used the following facts to derive the last inequality: 1) the Lipschitz continuity of the kernel gives us an upperbound on the values of $|a_i|$, i.e., $|a_i| \leq \mathcal{L}d(e_1, e'_1)$ for $1 \leq i \leq k$; and 2) since $\|x\|_2 = \|x'\|_2 = 1$, the absolute value of the elements in vectors x and x' cannot

be greater than 1, i.e., $|x_i| \leq 1, |x'_i| \leq 1$, for $1 \leq i \leq k$. Therefore,

$$\lambda_{\max}(\Delta K) = \max_{x: \|x\|_2=1} |x^T \Delta K x| \leq 2k\mathcal{L}d(e_1, e'_1).$$

Now, let $v_1, \dots, v_k \in \mathbb{R}^n$ be the k eigenvectors of matrix $(\mathbf{I} + K)^{-1}$. Note that $\{v_1, \dots, v_k\}$ is an orthonormal system and thus for any $x \in \mathbb{R}^n$ we can write it as $x = \sum_{i=1}^k c_i v_i$ and we have $\|x\|_2^2 = \sum_{i=1}^k c_i^2$. In order to bound the largest eigenvalue of $\Delta K(\mathbf{I} + K)$, we write

$$\begin{aligned}
 |x^T \Delta K (\mathbf{I} + K)^{-1} x| &= \left| x^T \Delta K (\mathbf{I} + K)^{-1} \sum_{i=1}^k c_i v_i \right| \\
 &= \left| x^T \Delta K \sum_{i=1}^k \lambda_i c_i v_i \right| \\
 &= \left| \sum_{j=1}^k c_j v_j^T \Delta K \left(\sum_{i=1}^k \lambda_i c_i v_i \right) \right| \\
 &= \sum_{i,j=1}^k \lambda_i c_i v_i^T \Delta K v_j \\
 &\stackrel{(a)}{\leq} 2k\mathcal{L}d(e_1, e'_1) \sum_{i,j=1}^k |c_i| |c_j| \\
 &= 2k\mathcal{L}d(e_1, e'_1) \left(\sum_{i=1}^k |c_i| \right)^2,
 \end{aligned}$$

where in (a) we used Eq. 10 and the fact that $\lambda_i \leq 1$ for $1 \leq i \leq k$. Using Cauchy-Schwarz inequality

$$\left(\sum_{i=1}^k |c_i| \right)^2 \leq k \sum_{i=1}^k |c_i|^2$$

and the assumption $\|x\|_2 = 1$, we conclude

$$\begin{aligned}
 |x^T \Delta K (\mathbf{I} + K)^{-1} x| &\leq 2k^2 \mathcal{L}d(e_1, e'_1) \sum_{i=1}^k |c_i|^2 \\
 &\leq 2k^2 \|x\|_2^2 \mathcal{L}d(e_1, e'_1) \\
 &\leq 2k^2 \mathcal{L}d(e_1, e'_1).
 \end{aligned}$$

Therefore,

$$\lambda_{\max}(\Delta K(\mathbf{I} + K)^{-1}) = \max_{x: \|x\|_2=1} |x^T \Delta K (\mathbf{I} + K)^{-1} x| \leq 2k^2 \mathcal{L}d(e_1, e'_1). \tag{11}$$

Finally, we can write the determinant of a matrix as the product of its eigenvalues, i.e.

$$\det(\mathbf{I} + \Delta K(\mathbf{I} + K)^{-1}) \leq (1 + 2k^2 \mathcal{L}d(e_1, e'_1))^k. \quad (12)$$

By substituting Eq. 11 and Eq. 12 into Eq. 9 we obtain

$$\begin{aligned} |f(S) - f(S')| &\leq \frac{1}{2} \left| \log(1 + 2k^2 \mathcal{L}d(e_1, e'_1))^k \right| \\ &\leq \frac{k}{2} \left| \log(1 + 2k^2 \mathcal{L}d(e_1, e'_1)) \right| \\ &\leq k^3 \mathcal{L}d(e_1, e'_1), \end{aligned}$$

where in the last inequality we used $\log(1+x) \leq x$, for $x \geq 0$.

Replacing all the k points in set S with another set S' of the same size, we get

$$|f(S) - f(S')| \leq k^3 \mathcal{L} \sum_{i=1}^k d(e_i, e'_i).$$

Hence, the differential entropy of the Gaussian process is λ -Lipschitz with $\lambda = \mathcal{L}k^3$.

A.4 Proof of Proposition 7

Assume we have a set S of k exemplars, i.e., $S_0 = \{e_1, \dots, e_k\}$, and each element of the dataset $v \in V$ is assigned to its closest exemplar. Now, if we replace set S with another set S' of the same size, the loss associated with every element $v \in V$ may be changed. W.l.o.g, assume we swap one exemplar at a time, i.e., in step i , $1 \leq i \leq k$, we have $S_i = \{e_1, \dots, e'_i, e_{i+1}, \dots, e_k\}$. Swapping the i^{th} exemplar $e_i \in S_{i-1}$ with another element $e'_i \in S'$, 4 cases may happen: 1) element v was not assigned to e_i before and doesn't get assigned to e'_i , 2) element v was assigned to e_i before and gets assigned to e'_i , 3) element v was not assigned to e_i before and gets assigned to e'_i , 4) element v was assigned to e_i before and gets assigned to another exemplar $e_x \in S_i \setminus \{e'_i\}$. For any element $v \in V$, we look into the four cases and show that in each case

$$|l(e'_i, v) - l(e_i, v)| \leq d(e_i, e'_i) \alpha R^{\alpha-1}.$$

- Case 1: In this case, element v was assigned to another exemplar $e_x \in S_i \setminus e_i$ and the assignment doesn't change. Therefore, there is no change in the value of the loss function.
- Case 2: In this case, element v was assigned to e_i before and gets assigned to e'_i . let $a = d(e_i, v)$ and $b = d(e'_i, v)$. Then we can write

$$\begin{aligned} |l(e'_i, v) - l(e_i, v)| &= |a^\alpha - b^\alpha| \\ &= |(a-b)(a^{\alpha-1} + a^{\alpha-2}b + \dots + ab^{\alpha-2} + b^{\alpha-1})| \\ &\leq d(e_i, e'_i) \alpha R^{\alpha-1}, \end{aligned} \quad (13)$$

where in the last step we used triangle inequality $|d(e_i, v) - d(e_i, v)| \leq d(e_i, e'_i)$ and the fact that data points are in a ball of diameter R in the metric space.

- Case 3: In this case, v was assigned to another exemplar $e_x \in S_{i-1} \setminus \{e_i\}$ and gets assigned to e'_i , which implies that $|l(e'_i, v) - l(e_x, v)| \leq |l(e_i, v) - l(e'_i, v)|$, since otherwise e would have been assigned to e_i before.

- Case 4: In the last case, element v was assigned to e_i before and gets assigned to another exemplar $e_x \in S_i \setminus \{e'_i\}$. Thus, we have $|l(e_x, v) - l(e_i, v)| \leq |l(e'_i, v) - l(e_i, v)|$ since otherwise v would have been assigned to e_x before. Hence, in all four cases the following inequality holds:

$$\left| \min_{e \in S_{i-1}} l(e, v) - \min_{e \in S_i} l(e, v) \right| \leq |l(e'_i, v) - l(e_i, v)| \leq d(e_i, e'_i) \alpha R^{\alpha-1}.$$

By using Eq. 13 and averaging over all elements $v \in V$, we have

$$\begin{aligned} |L(S_{i-1}) - L(S_i)| &= \frac{1}{|V|} \sum_{v \in V} \left| \min_{e \in S_{i-1}} l(e, v) - \min_{e \in S_i} l(e, v) \right| \\ &\leq \alpha R^{\alpha-1} d(e_i, e'_i). \end{aligned}$$

Thus, for any point e_0 that satisfies

$$\max_{v \in V} l(v, v') \leq l(v, e_0), \quad \forall v \in V \setminus S,$$

we have $L(\{e_0 \cup S\}) = L(\{S\})$ and thus

$$\begin{aligned} |f(S_{i-1}) - f(S_i)| &= |L(\{e_0\}) - L(\{e_0 \cup S_{i-1}\}) - L(\{e_0\}) + L(\{e_0 \cup S_i\})| \\ &\leq \alpha R^{\alpha-1} d(e_i, e'_i). \end{aligned}$$

Now, if we replace all the k points in set S with another set S' of the same size, we get

$$\begin{aligned} |f(S) - f(S')| &= \left| \sum_{i=1}^k f(S_{i-1}) - f(S_i) \right| \\ &= \sum_{i=1}^k |f(S_{i-1}) - f(S_i)| \\ &\leq \alpha R^{\alpha-1} \sum_{i=1}^k d(e_i, e'_i). \end{aligned}$$

Therefore, for $l = d^\alpha$, the loss function is λ -Lipschitz with $\lambda = \alpha R^{\alpha-1}$.

A.5 Proof of Theorem 8

In the following, we say that sets S and S' are γ -close if $|f(S) - f(S')| \leq \gamma$. First, we need the following lemma.

Lemma 19 *If for each $e_i \in A_i^c[k]$, $|N_\alpha(e_i)| \geq km \log(k/\delta^{1/m})$, and if V is partitioned into sets V_1, V_2, \dots, V_m , where each element is randomly assigned to one set with equal probabilities, then there is at least one partition with a subset $A_i^c[k]$ such that $|f(A_i^c[k]) - f(A_i^c[k])| \leq \lambda \alpha k$ with probability at least $(1 - \delta)$.*

Proof By the hypothesis, the α neighborhood of each element in $A^c[k]$ contains at least $km \log(k/\delta^{1/m})$ elements. For each $e_i \in A^c[k]$, let us take a set of $m \log(k/\delta^{1/m})$ elements from its α -neighborhood. These sets can be constructed to be mutually disjoint, since each α -neighborhood contains $m \log(k/\delta^{1/m})$ elements. We wish to show that at least one of the m partitions of V contains elements from α -neighborhoods of each element.

Each of the $m \log(k/\delta^{1/m})$ elements goes into a particular V_j with a probability $1/m$. The probability that a particular V_j does not contain an element α -close to $e_i \in A^c[k]$ is $\delta^{1/m}$. The probability that V_j does not contain elements α -close to one or more of the k elements is at most $\delta^{1/m}$ (by union bound). The probability that each V_1, V_2, \dots, V_m does not contain elements from the α -neighborhood of one or more of the k elements is at most δ . Thus, with high probability of at least $(1 - \delta)$, at least one of V_1, V_2, \dots, V_m contains an $A^c[k]$ that is $\lambda\alpha k$ -close to $A^c[k]$. ■

By lemma 19, for some V_{i_0} $|f(A^c[k]) - f(A_{i_0}^c[k])| \leq \lambda\alpha k$ with the given probability. Furthermore, $f(A_{i_0}^c[k]) \geq (1 - e^{-\alpha/k})f(A^c[k])$ by Lemma 18. Therefore, the result follows using arguments analogous to the proof of Theorem 4.

A.6 Proof of Theorem 9

The following lemma says that in a sample drawn from distribution over an infinite dataset, a sufficiently large sample size guarantees a dense neighborhood near each element of $A^c[k]$ when the elements are from representative regions of the data.

Lemma 20 *A number of elements: $n \geq \frac{Sk m \log(k/\delta^{1/m})}{\beta g(\alpha)}$, where $\alpha \leq \alpha^*$, suffices to have at least $4km \log(k/\delta^{1/m})$ elements in the α -neighborhood of each $e_i \in A^c[k]$ with probability at least $(1 - \delta)$, for small values of δ .*

Proof The expected number of α -neighbors of an $e_i \in A^c[k]$ is $E[|N_\alpha(e_i)|] \geq Sk m \log(k/\delta^{1/m})$. We now show that in a random set of samples, at least a half of this number of neighbors is realized with high probability near each element of $A^c[k]$.

This follows from a Chernoff bound:

$$\Pr[|N_\alpha(e_i)| \leq 4km \log(k/\delta^{1/m})] \leq e^{-km \log(k/\delta^{1/m})} \leq (\delta^{1/m}/k)^{km}.$$

Therefore, the probability that some $e_i \in A^c[k]$ does not have a suitable sized neighborhood is at most $k(\delta^{1/m}/k)^{km}$. For $\delta \leq 1/k$, $k\delta^{km} \leq \delta^m$. Therefore, with probability at least $(1 - \delta)$, the α -neighborhood of each element $e_i \in A^c[k]$ contains at least $4km \log(1/\delta)$ elements. ■

Lemma 21 *For $n \geq \frac{Sk m \log(k/\delta^{1/m})}{\beta g(\frac{\delta}{k})}$, where $\frac{\delta}{k} \leq \alpha^*$, if V is partitioned into sets V_1, V_2, \dots, V_m , where each element is randomly assigned to one set with equal probabilities, then for sufficiently small values of δ , there is at least one partition with a subset $A^c[k]$ such that $|f(A^c[k]) - f(A_{i_0}^c[k])| \leq \varepsilon$ with probability at least $(1 - \delta)$.*

Proof Follows directly by combining Lemma 20 and Lemma 19. The probability that some element does not have a sufficiently dense $\varepsilon/\lambda k$ -neighborhood with $km \log(2k/\delta^{1/m})$ elements is at most $(\delta/2)$ for sufficiently small δ , and the probability that some partition does not contain elements from the one or more of the dense neighborhoods is at most $(\delta/2)$. Therefore, the result holds with probability at least $(1 - \delta)$. ■

By Lemma 21, there is at least one V_j such that $|f(A^c[k]) - f(A_{j_0}^c[k])| \leq \varepsilon$ with the given probability. And $f(A_{j_0}^c[k]) \geq (1 - e^{-\alpha/k})f(A^c[k])$ using Lemma 18. The result follows using arguments analogous to the proof of Theorem 4.

A.7 Proof of Theorem 10

Note that each machine has on the average n/m elements. Let us define Π_i the event that $n_i/2m < |V_i^c| < 2n_i/m$. Then based on the Chernoff bound we know that $\Pr(\neg\Pi_i) \leq 2\exp(-n_i/8m)$. Let us also define $\xi_i(S)$ the event that $|f_{V_i}(S) - f(S)| < \varepsilon$, for some fixed $\varepsilon < 1$ and a fixed set S with $|S| \leq k$. Note that $\xi_i(S)$ denotes the event that the empirical mean is close to the true mean. Based on the Hoeffding inequality (without replacement) we have $\Pr(\neq \xi_i S) \leq 2\exp(-2n\varepsilon^2/m)$. Hence,

$$\Pr(\xi(S) \wedge \Pi_i) \geq 1 - 2\exp(-2n\varepsilon^2/m) - 2\exp(-n/8m).$$

Let ξ_i be an event that $|f_{V_i}(S) - f(S)| < \varepsilon$, for any S such that $|S| \leq k$. Note that there are at most n^k sets of size at most k . Hence,

$$\Pr(\xi_i \wedge \Pi_i) \geq 1 - 2n^k \exp(-2n\varepsilon^2/m) - \exp(-n/8m). \quad (14)$$

As a result, for $\varepsilon < 1/4$ we have

$$\Pr(\xi_i \wedge \Pi_i) \geq 1 - 4n^k \exp(-2n\varepsilon^2/m).$$

Since there are m machines, by the union bound we can conclude that

$$\Pr(\xi(S) \wedge \Pi_i \text{ on all machines}) \geq 1 - 4mn^k \exp(-2n\varepsilon^2/m).$$

The above calculation implies that we need to choose $\delta \geq 4mn^k \exp(-2n\varepsilon^2/m)$. Let n_0 be chosen in a way that for any $n \geq n_0$ we have $\ln(n)/n \leq \varepsilon^2/(mk)$. Then, we need to choose n as follows:

$$n = \max\left(n_0, \frac{m \log(\delta/4m)}{\varepsilon^2}\right).$$

Hence for the above choice of n , there is at least one V_i such that $|f(A^c[k]) - f(A_{i_0}^c[k])| \leq \varepsilon$ with probability $1 - \delta$. Hence the solution is ε away from the optimum solution with probability $1 - \delta$. Now if we confine the evaluation of $f(A_{i_0}^c)$ to data point only in machine i then under the assumption of Theorem 9 we lose another ε . Formally, the result at this point simply follows by combining Theorem 4 and Theorem 9.

A.8 Proof of Theorem 12

The proof is similar to the proof of Theorem 3 and Theorem 4 and follows from the following lemmas.

Lemma 22 $\max_i f(A_i^c[\zeta]) \geq \frac{1}{m} f(A^c[\zeta])$.

Proof Let B_i be the elements in V_i that are contained in the optimal solution, $B_i = A^c[\zeta] \cap V_i$. Since $A^c[\zeta] \in \zeta$ and ζ is a set of hereditary constraints, we must have $B_i \in \zeta$ as well. Using submodularity of f and by the same argument as in the proof of Lemma 16, we have

$$\begin{aligned} f(A^c[\zeta]) &= f(B_1 \cup \dots \cup B_m) = f(B_1) + f(B_2|B_1) + \dots + f(B_m|B_{m-1}, \dots, B_1) \\ &\leq f(B_1) + \dots + f(B_m). \end{aligned}$$

Since $f(A_i^c[\zeta]) \geq f(B_i)$ we get

$$f(A^c[\zeta]) \leq f(A_1^c[\zeta]) + \dots + f(A_m^c[\zeta]) \leq m \max_i f(A_i^c[\zeta]).$$

■

Lemma 23 $\max_i f(A_i^c[\zeta]) \geq \frac{1}{k} f(A^c[\zeta])$.

Proof The proof follows the outline of the proof of Lemma 17. Let $f(A^c[\zeta]) = f(\{u_1, \dots, u_{\rho(\zeta)}\})$. Since $A^c[\zeta] \in \zeta$ and ζ is a set of hereditary constraints, we have $u_i \in \zeta$. Using submodularity of f , we have

$$f(A^c[\zeta]) \leq \sum_{i=1}^{\rho(\zeta)} f(u_i) \leq \rho(\zeta) f(u^*),$$

where $u^* = \arg \max_i f(u_i)$. Suppose that $u^* \in V_j$, we get

$$f(\max_i f(A_i^c[\zeta])) \geq f(A_j^c[\zeta]) \geq f(u^*) \geq \frac{1}{\rho(\zeta)} f(A^c[\zeta]).$$

Since $f(A^d[m, \rho(\zeta)]) \geq \max_i f(A_i^c[\zeta])$; from Lemma 23 and 22 we have

$$f(A^d[m, \rho(\zeta)]) \geq \frac{1}{\min(m, \rho(\zeta))} f(A^c[\zeta]).$$

For the black box algorithm X with a τ -approximation guarantee, we have

$$f(A_i^X[\zeta]) \geq \tau f(A_i^c[\zeta]).$$

Now, we generalize the definitions used in the proof of Theorem 4

$$\begin{aligned} B^{\text{gc}} &= \cup_{i=1}^m A_i^{\text{gc}}[\zeta], \\ A_{\max}^{\text{gc}}[\zeta] &= \max_i f(A_i^{\text{gc}}[\zeta]), \\ \bar{A}[\zeta] &= \arg \max_{S \subseteq B^{\text{gc}} \& |S| \leq \rho(\zeta)} f(S). \end{aligned}$$

Then using Eq. 15 again, we obtain

$$\begin{aligned} f(A^d[m, \zeta]) &\geq \max \{f(A_{\max}^{\text{gc}}[\zeta]), \tau f(\bar{A}[\zeta])\} \\ &\geq \frac{\tau}{\min(m, \rho(\zeta))} f(A^c[\zeta]). \end{aligned}$$

Note that since we do not use monotonicity of the submodular function in any of the proofs, the results hold in general for constrained maximization of any non-negative submodular function.

A.9 Proof of Theorem 13

Lemma 24 If for each $e_i \in A^X[\zeta]$, $|N_\alpha(e_i)| \geq \rho(\zeta) m \log(\rho(\zeta)/\delta^{1/m})$, and if V is partitioned into sets V_1, V_2, \dots, V_m , where each element is randomly assigned to one set with equal probabilities, then there is at least one partition with a subset $A_i^X[\zeta] \in \zeta$ such that $|f(A^c[\zeta]) - f(A_i^X[\zeta])| \leq \lambda \alpha \rho(\zeta)$ with probability at least $(1 - \delta)$.

The proof is similar to the proof of Lemma 19 by taking disjoint sets of size $m \log(\rho(\zeta)/\delta^{1/m})$ in an α -neighborhood of each $e_i \in A^X[\zeta]$ and showing that with high probability, at least one of the m partitions of V contains elements from α -neighborhoods of each element in the optimal solution. Note that now the size of the optimal solution is at most $\rho(\zeta)$. Since ζ is locally replaceable with parameter α , as elements of $A^c[\zeta]$ gets replaced by nearby elements in their α -neighborhood, the resulting set is also a feasible solution.

By Lemma 24, for some V_i , $|f(A^c[\zeta]) - f(A_i^X[\zeta])| \leq \lambda \alpha \rho(\zeta)$ with the given probability. On the other hand, for the black box algorithm X , we have $f(A_i^X[\zeta]) \geq \tau f(A_i^c[\zeta])$. Therefore, the result follows using arguments analogous to the proof of Theorem 12.

A.10 Proof of Theorem 14

We use the following Lemmas to show that in a sample drawn from a distribution over an infinite dataset, a sufficiently large sample size guarantees a dense neighborhood near each element of the optimal solution.

Lemma 25 A number of elements: $n \geq \frac{8\rho(\zeta)m \log(\rho(\zeta)/\delta^{1/m})}{\beta g(\alpha)}$, where $\alpha \leq \alpha^*$, suffices to have at least $4\rho(\zeta)m \log(\rho(\zeta)/\delta^{1/m})$ elements in the α -neighborhood of each $e_i \in A^c[\zeta]$ with probability at least $(1 - \delta)$, for small values of δ .

Lemma 26 For $n \geq \frac{8\rho(\zeta)m \log(\rho(\zeta)/\delta^{1/m})}{\beta g(\frac{\varepsilon}{\lambda \rho(\zeta)})}$, where $\frac{\varepsilon}{\lambda \rho(\zeta)} \leq \alpha^*$, if V is partitioned into sets V_1, V_2, \dots, V_m , where each element is randomly assigned to one set with equal probabilities, then for sufficiently small values of δ , there is at least one partition with a subset $A_i^c[\zeta]$ such that $|f(A^c[\zeta]) - f(A_i^c[\zeta])| \leq \varepsilon$ with probability at least $(1 - \delta)$.

The proofs follows the same arguments as in the proof of Lemma 20 and 21. Recall that, by assumption ζ is locally replaceable with parameter α . Hence, for $\varepsilon \leq \alpha \lambda \rho(\zeta)$, any set ε -close to the optimal solution is also a feasible solution.

By Lemma 26, there is at least one V_i such that $|f(A_i^c(\zeta)) - f(A_i^c(\zeta))| \leq \epsilon$ with the given probability. Furthermore, for the black box algorithm X , we have $f(A_i^{opt}(\zeta)) \geq \tau f(A_i^c(\zeta))$. Thus the result follows using arguments analogous to the proof of Theorem 12.

A.11 Proof of Theorem 15

Again the proof follows the same line of reasoning as the proof of Theorem 10, except that for a constraint set ζ with $\rho(\zeta) = \max_{S \subseteq \zeta} |S|$, there are at most $n^{\rho(\zeta)}$ feasible solutions. Using the same definitions for Π_i and ξ_i as in the proof of Theorem 10, instead of Eq. 14 we get

$$\Pr(\xi_i \wedge \Pi_i) \geq 1 - 2n^{\rho(\zeta)} (\exp(-2n\epsilon^2/m) - \exp(-n/8m)).$$

As a result, for $\epsilon < 1/4$ and using union bound we conclude that

$$\Pr((\xi_i \wedge \Pi_i) \text{ on all machines}) \geq 1 - 4mn^{\rho(\zeta)} \exp(-2n\epsilon^2/m),$$

which implies that we need to choose $\delta \geq 4mn^{\rho(\zeta)} \exp(-2n\epsilon^2/m)$. Now if n_0 be chosen in a way that for any $n \geq n_0$ we have $\ln(n)/n \leq \epsilon^2/(mb)$, we get $n \geq \max(n_0, m \log(\delta/4m)/\epsilon^2)$.

Bearing in mind that ζ is locally replaceable, there is at least one V_i such that the solution $A_i^c(\zeta)$ is feasible and ϵ away from the optimum solution with probability $1 - \delta$. Now under the assumption of Theorem 14, if we evaluate $f(A_i^c)$ only on machine i , then we lose another ϵ . Now by combining Theorem 12 and Theorem 14 we get the desired result.

References

- Yahoo! academic relations. r6a, yahoo! front page today module user click log dataset, version 1.0, 2012. URL <http://webscope.sandbox.yahoo.com>.
- Zeinab Abbassi, Vahab S Mirrokni, and Mayur Thakur. Diversity maximization under matroid constraints. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 32–40. ACM, 2013.
- Mahmoudreza Bahaei, Baharan Mirzasoleiman, Mahdi Jalili, and Mohammad Ali Safari. Revenue maximization in social networks through discounting. *Social Network Analysis and Mining*, 3(4):1249–1262, 2013.
- Ashwinkumar Badanidiyuru and Jan Vondrák. Fast algorithms for maximizing submodular functions. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1497–1514. SIAM, 2014.
- Ashwinkumar Badanidiyuru, Baharan Mirzasoleiman, Amin Karbasi, and Andreas Krause. Streaming submodular maximization: Massive data summarization on the fly. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 671–680. ACM, 2014.
- Rafael Barbosa, Alina Ene, Hui Nguyen, and Justin Ward. The power of randomization: Distributed submodular maximization on massive datasets. *Proceedings of The 32nd International Conference on Machine Learning*, pages 1236–1244, 2015.
- Guy E Blelloch, Richard Peng, and Kanat Tangwongsan. Linear-work greedy parallel approximate set cover and variants. In *Proceedings of the Twenty-Third Annual ACM Symposium on Parallelism in Algorithms and Architectures*, pages 23–32. ACM, 2011.
- Ferenc Bordon. Kosarak dataset, 2012. URL <http://fimi.ua.ac.be/data/>.
- Niv Buchbinder, Michael Feldman, Joseph Naor, and Roy Schwartz. A tight linear time $(1/2)$ -approximation for unconstrained submodular maximization. In *59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 649–658. IEEE, 2012.
- Niv Buchbinder, Moran Feldman, Joseph Seffi Naor, and Roy Schwartz. Submodular maximization with cardinality constraints. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1433–1452. SIAM, 2014.
- Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.
- Flavio Cherielli, Ravi Kumar, and Andrew Tomkins. Max-cover in map-reduce. In *Proceedings of the 19th International Conference on World Wide Web*, pages 231–240. ACM, 2010.
- Cheng Chu, Saang Kyun Kim, Yi-An Lin, YuanYuan Yu, Gary Bradski, Andrew Y Ng, and Kunle Olukotun. Map-reduce for machine learning on multicore. *Advances in Neural Information Processing Systems*, 19:281, 2007.
- Michele Conforti and Gérard Cornuéjols. Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the rank- ρ -theorem. *Discrete Applied Mathematics*, 7(3):251–274, 1984.
- Graham Cormode, Howard Karloff, and Anthony Wirth. Set cover algorithms for very large datasets. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 479–488. ACM, 2010.
- Sven De Vries and Rakesh V Vohra. Combinatorial auctions: A survey. *INFORMS Journal on Computing*, 15(3):284–309, 2003.
- Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.
- Nan Du, Yingyu Liang, Maria Florina Balcan, and Le Song. Budgeted influence maximization for multiple products. *arXiv preprint arXiv:1312.2164*, 2013.
- Delbert Dueck and Brendan J Frey. Non-metric affinity propagation for unsupervised image categorization. In *IEEE 11th International Conference on Computer Vision (ICCV)*, pages 1–8. IEEE, 2007.
- Jaliya Ekanayake, Shirdeep Pallickara, and Geoffrey Fox. Mapreduce for data intensive scientific analyses. In *IEEE Fourth International Conference on eScience*, pages 277–284. IEEE, 2008.

- Uriel Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.
- Marshall L. Fisher, George L. Nemhauser, and Laurence A. Wolsey. An analysis of approximations for maximizing submodular set functions - II. *Mathematical Programming Study*, (8):73–87, 1978.
- Rahul Garg, Vijay Kumar, and Vinayaka Pandit. Approximation algorithms for budget-constrained auctions. In *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques*, pages 102–113. Springer, 2001.
- Karolien Geurts, Geert Wets, Tom Brijs, and Koen Vanhoof. Profiling of high-frequency accident locations by use of association rules. *Transportation Research Record: Journal of the Transportation Research Board*, 1840(1):123–130, 2003.
- Shayan Oveis Gharan and Jan Vondrák. Submodular maximization by simulated annealing. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1098–1116. SIAM, 2011.
- Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, pages 427–486, 2011.
- Daniel Golovin, Matthew Faulkner, and Andreas Krause. Online distributed sensor selection. In *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 220–231. ACM, 2010.
- Manuel Gomez Rodriguez, Jure Leskovec, and Andreas Krause. Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1019–1028. ACM, 2010.
- Andrew Guillory and Jeff Bilmes. Active semi-supervised learning using submodular functions. In *Proceedings of the Twenty-Seventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-11)*, pages 274–282. AUAI, 2011.
- Anupam Gupta, Aaron Roth, Grant Schoenebeck, and Kunal Talwar. Constrained non-monotone submodular maximization: Offline and secretary algorithms. In *Internet and Network Economics*, pages 246–257. Springer, 2010.
- Jason Hartline, Vahab Mirrokni, and Mukund Sundararajan. Optimal marketing strategies over social networks. In *Proceedings of the 17th International Conference on World Wide Web*, pages 189–198. ACM, 2008.
- Howard Karloff, Siddharth Suri, and Sergei Vassilvitskii. A model of computation for mapreduce. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 938–948. Society for Industrial and Applied Mathematics, 2010.
- Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.
- David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 137–146. ACM, 2003.
- Chun-Wa Ko, Jon Lee, and Maurice Queyranne. An exact algorithm for maximum entropy sampling. *Operations Research*, 43(4):684–691, 1995.
- Andreas Krause and Daniel Golovin. Submodular function maximization. *Tractability: Practical Approaches to Hard Problems*, 3:19, 2012.
- Andreas Krause and Ryan G Gomes. Budgeted nonparametric learning from data streams. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 391–398, 2010.
- Andreas Krause and Carlos Guestrin. Near-optimal nonmyopic value of information in graphical models. In *Proceedings of Uncertainty in Artificial Intelligence (UAI)*, page 5, 2005a.
- Andreas Krause and Carlos Guestrin. A note on the budgeted maximization on submodular functions. Technical Report CMU-CALD-05-103, Carnegie Mellon University, 2005b.
- Andreas Krause and Carlos Guestrin. Submodularity and its applications in optimized information gathering. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(4):32, 2011.
- Alex Kulesza. Determinantal point processes for machine learning. *Machine Learning*, 5(2-3):123–286, 2012.
- Ariel Kulik, Hadas Shachnai, and Tami Tamir. Maximizing submodular set functions subject to multiple linear constraints. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 545–554. Society for Industrial and Applied Mathematics, 2009.
- Ravi Kumar, Benjamin Moseley, Sergei Vassilvitskii, and Andrea Vattani. Fast greedy algorithms in mapreduce and streaming. In *Proceedings of the 25th ACM Symposium on Parallelism in Algorithms and Architectures*, pages 1–10. ACM, 2013.
- Silvio Lattanzi, Benjamin Moseley, Siddharth Suri, and Sergei Vassilvitskii. Filtering: a method for solving graph problems in mapreduce. In *Proceedings of the Twenty-Third Annual ACM Symposium on Parallelism in Algorithms and Architectures*, pages 85–94. ACM, 2011.
- Jon Lee, Vahab S Mirrokni, Viswanath Nagarajan, and Maxim Sviridenko. Non-monotone submodular maximization under matroid and knapsack constraints. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pages 323–332. ACM, 2009a.
- Jon Lee, Maxim Sviridenko, and Jan Vondrák. Submodular maximization over multiple matroids via generalized exchange properties. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 244–257. Springer, 2009b.

- Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *KDD '07: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 420–429. ACM, 2007.
- Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 510–520. Association for Computational Linguistics, 2011.
- Ottie Macchi. The coincidence approach to stochastic point processes. *Advances in Applied Probability*, pages 83–122, 1975.
- Michel Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization Techniques*, pages 234–243. Springer, 1978.
- Vahab Mirrokni and Morteza Zadimoghaddam. Randomized composable core-sets for distributed submodular maximization. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, STOC '15, pages 153–162. ACM, 2015.
- Baharan Mirzasoleiman, Mahmoudreza Babaei, and Mahdi Jalili. Immunizing complex networks with limited budget. *EPL (Europhysics Letters)*, 98(3):38004, 2012.
- Baharan Mirzasoleiman, Amin Karbasi, Rik Sarkar, and Andreas Krause. Distributed submodular maximization: Identifying representative elements in massive data. In *Advances in Neural Information Processing Systems*, pages 2049–2057, 2013.
- Baharan Mirzasoleiman, Ashwin Karbasi, Ashwinkumar Badanidiyuru, and Andreas Krause. Lazy rather than greedy. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015a.
- Baharan Mirzasoleiman, Amin Karbasi, Ashwinkumar Badanidiyuru, and Andreas Krause. Distributed submodular cover: Succinctly summarizing massive data. In *Advances in Neural Information Processing Systems*, 2015b.
- Baharan Mirzasoleiman, Ashwin Badanidiyuru, and Amin Karbasi. Fast constrained submodular maximization: Personalized data summarization. In *Proceedings of The 33rd International Conference on Machine Learning*, 2016.
- Ramasuri Narayanan and Amit A Nanavati. Viral marketing for product cross-sell through social networks. In *Machine Learning and Knowledge Discovery in Databases*, pages 581–596. Springer, 2012.
- George L Nemhauser and Leonard A Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research*, 3(3):177–188, 1978.
- George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1): 265–294, 1978.
- Tore Opsahl and Pietro Panzarasa. Clustering in weighted networks. *Social Networks*, 31(2):155–163, 2009.
- Carl Edward Rasmussen. Gaussian processes in machine learning. In *Advanced Lectures on Machine Learning*, pages 63–71. Springer, 2004.
- Matthew Streeter, Daniel Golovin, and Andreas Krause. Online learning of assignments. In *Advances in Neural Information Processing Systems*, pages 1794–1802, 2009.
- Maxim Sviridenko. A note on maximizing a submodular set function subject to knapsack constraint. *Operations Research Letters*, 32, 2004.
- Antonio Torralba, Rob Fergus, and William T Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.
- Athanasios Tsanas, Max Little, Patrick E McSharry, Lorraine O Raminig, et al. Enhanced classical dysphonia measures and sparse regression for telemonitoring of parkinson’s disease progression. In *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 594–597. IEEE, 2010.
- Kai Wei, Yuzong Lin, Katrin Kirchhoff, and Jeff Bilmes. Using document summarization techniques for speech data subset selection. In *Proceedings of NAACL-HLT*, pages 721–726, 2013.
- Kai Wei, Rishabh Iyer, and Jeff Bilmes. Fast multi-stage submodular maximization. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1494–1502, 2014.

On the properties of variational approximations of Gibbs posteriors

Pierre Alquier
James Ridgway
Nicolas Chopin

ENSAE
3 Avenue Pierre Larousse
92245 MALAKOFF, FRANCE

PIERRE.ALQUIER@ENSAE.FR
JAMES.RIDGWAY@ENSAE.FR
NICOLAS.CHOPIN@ENSAE.FR

Editor: Yee Whye Teh

Abstract

The PAC-Bayesian approach is a powerful set of techniques to derive non-asymptotic risk bounds for random estimators. The corresponding optimal distribution of estimators, usually called the Gibbs posterior, is unfortunately often intractable. One may sample from it using Markov chain Monte Carlo, but this is usually too slow for big datasets. We consider instead variational approximations of the Gibbs posterior, which are fast to compute. We undertake a general study of the properties of such approximations. Our main finding is that such a variational approximation has often the same rate of convergence as the original PAC-Bayesian procedure it approximates. In addition, we show that, when the risk function is convex, a variational approximation can be obtained in polynomial time using a convex solver. We give finite sample oracle inequalities for the corresponding estimator. We specialize our results to several learning tasks (classification, ranking, matrix completion), discuss how to implement a variational approximation in each case, and illustrate the good properties of said approximation on real datasets.

1. Introduction

A Gibbs posterior, also known as a PAC-Bayesian or pseudo-posterior, is a probability distribution for random estimators of the form:

$$\hat{\rho}_\lambda(d\theta) = \frac{\exp[-\lambda r_n(\theta)]}{\int \exp[-\lambda r_n] d\pi} \pi(d\theta).$$

More precise definitions will follow, but for now, θ may be interpreted as a parameter (in a finite or infinite-dimensional space), $r_n(\theta)$ as an empirical measure of risk (e.g. prediction error), and $\pi(d\theta)$ a prior distribution.

We will follow in this paper the PAC (Probably Approximately Correct)-Bayesian approach, which originates from machine learning (Shawe-Taylor and Williamson, 1997;

McAllester, 1998; Catoni, 2004); see Catoni (2007) for an exhaustive study, and Jiang and Tamer (2008); Yang (2004); Zhang (2006); Dalalyan and Tsybakov (2008) for related perspectives (such as the aggregation of estimators in the last three papers). There, $\hat{\rho}_\lambda$ appears as the probability distribution that minimizes the upper bound of an oracle inequality on the risk of *random* estimators. The PAC-Bayesian approach offers sharp theoretical guarantees on the properties of such estimators, without assuming a particular model for the data generating process.

The Gibbs posterior has also appeared in other places, and under different motivations: in Econometrics, as a way to avoid direct maximization in moment estimation (Chernozhuikov and Hong, 2003); and in Bayesian decision theory, as a way to define a Bayesian posterior distribution when no likelihood has been specified (Bissiri et al., 2013). Another well-known connection, although less directly useful (for Statistics), is with thermodynamics, where r_n is interpreted as an energy function, and λ as the inverse of a temperature.

Whatever the perspective, estimators derived from Gibbs posteriors usually show excellent performance in diverse tasks, such as classification, regression, ranking, and so on, yet their actual implementation is still far from routine. The usual recommendation (Dalalyan and Tsybakov, 2012; Alquier and Bian, 2013; Guedj and Alquier, 2013) is to sample from a Gibbs posterior using MCMC (Markov chain Monte Carlo, see e.g. Green et al., 2015); but constructing an efficient MCMC sampler is often difficult, and even efficient implementations are often too slow for practical uses when the dataset is very large.

In this paper, we consider instead VB (Variational Bayes) approximations, which have been initially developed to provide fast approximations of ‘true’ posterior distributions (i.e. Bayesian posterior distributions for a given model); see Jordan et al. (1999); MacKay (2002) and Chap. 10 in Bishop (2006).

Our main results are as follows: when PAC-Bayes bounds are available - mainly, when a strong concentration inequality holds - replacing the Gibbs posterior by a variational approximation does not affect the rate of convergence to the best possible prediction, on the condition that the Kullback-Leibler divergence between the posterior and the approximation is itself properly controlled. Furthermore, for convex risks we show that one can obtain polynomial time algorithms based on optimal convex solvers.

We also provide empirical bounds, which may be computed from the data to ascertain the actual performance of estimators obtained by variational approximation. All the results gives strong incentives, we believe, to recommend Variational Bayes as the default approach to approximate Gibbs posteriors. We also provide a R package¹, written in C++ to compute a Gaussian variational approximation in the case of the hinge risk.

The rest of the paper is organized as follows. In Section 2, we present the notations and assumptions. In Section 3, we introduce variational approximations and the corresponding algorithms. The main results are provided in a general form in Section 4: in Subsection 4.1, we give results under the assumption that a Hoeffding type inequality holds (slow rates) and

1. PACVB package: <https://cran.r-project.org/web/packages/PACVB/index.html>

in Subsection 4.2, we give results under the assumption that a Bernstein type inequality holds (fast rates). Note that for the sake of brevity, we will refer to these settings as ‘‘Hoeffding assumption’’ and ‘‘Bernstein assumption’’ even though this terminology is non-standard. We then apply these results in various settings: classification (Section 5), convex classification (Section 6), ranking (Section 7), and matrix completion (Section 8). In each case, we show how to specialise the general results of Section 4 to the considered application, in order to obtain the properties of the VB approximation, and we also discuss its numerical implementation. All the proofs are collected in the Appendix.

2. PAC-Bayesian framework

We observe a sample $(X_1, Y_1), \dots, (X_n, Y_n)$, taking values in $\mathcal{X} \times \mathcal{Y}$, where the pairs (X_i, Y_i) have the same distribution P . We will assume explicitly that the (X_i, Y_i) 's are independent in several of our specialised results, but we do not make this assumption at this stage, as some of our general results, and more generally the PAC-Bayesian theory, may be extended to dependent observations; see e.g. Alquier and Li (2012). The label set \mathcal{Y} is always a subset of \mathbb{R} . A set of predictors is chosen by the statistician: $\{f_\theta : \mathcal{X} \rightarrow \mathbb{R}, \theta \in \Theta\}$. For example, in linear regression, we may have: $f_\theta(x) = \langle \theta, x \rangle$, the inner product of $\mathcal{X} = \mathbb{R}^d$, while in classification, one may have $f_\theta(x) = \mathbb{1}_{\langle \theta, x \rangle > 0} \in \{0, 1\}$.

We assume we have at our disposal a risk function $R(\theta)$; typically $R(\theta)$ is a measure of the prediction error. We set $\bar{R} = R(\bar{\theta})$, where $\bar{\theta} \in \arg \min R$; i.e. \bar{f}_θ is an optimal predictor. We also assume that the risk function $R(\theta)$ has an empirical counterpart $r_n(\theta)$, and set $\bar{r}_n = r_n(\bar{\theta})$. Often, R and r_n are based on a loss function $\ell : \mathbb{R}^2 \rightarrow \mathbb{R}$; i.e. $R(\theta) = \mathbb{E}[\ell(Y, f_\theta(X))]$ and $\bar{r}_n(\theta) = \frac{1}{n} \sum_{i=1}^n \ell(Y_i, f_\theta(X_i))$. (In this paper, the symbol \mathbb{E} will always denote the expectation with respect to the (unknown) law P of the (X_i, Y_i) 's.) There are situations however (e.g. ranking), where R and r_n have a different form.

We define a prior probability measure $\pi(\cdot)$ on the set Θ (equipped with the standard σ -algebra for the considered context), and we let $\mathcal{M}_+^1(\Theta)$ denote the set of all probability measures on Θ .

Definition 2.1 We define, for any $\lambda > 0$, the pseudo-posterior $\hat{\rho}_\lambda$ by

$$\hat{\rho}_\lambda(\theta) = \frac{\exp[-\lambda r_n(\theta)]}{\int \exp[-\lambda r_n] d\pi} \pi(d\theta).$$

The pseudo-posterior $\hat{\rho}_\lambda$ (also known as the Gibbs posterior, Catoni (2004, 2007), or the exponentially weighted aggregate, Dalalyan and Tsybakov (2008)) plays a central role in the PAC-Bayesian approach. It is obtained as the distribution that minimizes the upper bound of a certain oracle inequality applied to *random* estimators. Practical estimators (predictors) may be derived from the pseudo-posterior, by e.g. taking the expectation, or sampling from it. Of course, when $\exp[-\lambda r_n(\theta)]$ may be interpreted as the likelihood of a

certain model, $\hat{\rho}_\lambda$ becomes a Bayesian posterior distribution, but we will not restrict our attention to this particular case.

The following ‘theoretical’ counterpart of $\hat{\rho}_\lambda$ will prove useful to state results.

Definition 2.2 We define, for any $\lambda > 0$, π_λ as

$$\pi_\lambda(\theta) = \frac{\exp[-\lambda R(\theta)]}{\int \exp[-\lambda R] d\pi} \pi(d\theta).$$

We will derive PAC-Bayesian bounds on predictions obtained by variational approximations of $\hat{\rho}_\lambda$ under two types of assumptions: a Hoeffding-type assumption, from which we may deduce slow rates of convergence (Subsection 4.1), and a Bernstein-type assumption, from which we may obtain fast rates of convergence (Subsection 4.2).

Definition 2.3 We say that a Hoeffding assumption is satisfied for prior π when there is a function f and an interval $I \subset \mathbb{R}_+^*$ such that, for any $\lambda \in I$, for any $\theta \in \Theta$,

$$\left. \begin{aligned} \pi(\mathbb{E} \exp \{\lambda [R(\theta) - r_n(\theta)]\}) \\ \pi(\mathbb{E} \exp \{\lambda [r_n(\theta) - R(\theta)]\}) \end{aligned} \right\} \leq \exp [f(\lambda, n)]. \quad (1)$$

Inequality (1) can be interpreted as an integrated version (with respect to π) of Hoeffding's inequality, for which $f(\lambda, n) \asymp \lambda^2/n$. In many cases the loss will be bounded uniformly over θ ; then Hoeffding's inequality will directly imply (1). The expectation with respect to π in (1) allows us to treat some cases where the loss is not upper bounded by specifying a prior with sufficiently light tails.

Definition 2.4 We say that a Bernstein assumption is satisfied for prior π when there is a function g and an interval $I \subset \mathbb{R}_+^*$ such that, for any $\lambda \in I$, for any $\theta \in \Theta$,

$$\left. \begin{aligned} \pi(\mathbb{E} \exp \{\lambda [R(\theta) - \bar{R}] - \lambda [r_n(\theta) - \bar{r}_n]\}) \\ \pi(\mathbb{E} \exp \{\lambda [r_n(\theta) - \bar{r}_n] - \lambda [R(\theta) - \bar{R}]\}) \end{aligned} \right\} \leq \pi(\exp [g(\lambda, n) |R(\theta) - \bar{R}|]). \quad (2)$$

This assumption is satisfied for example by sums of i.i.d. sub-exponential random variables, see Subsection 2.4 p. 27 in Boucheron et al. (2013), when a margin assumption on the function $R(\cdot)$ is satisfied (Tsybakov, 2004). This is discussed in Section 4.2. Again, extensions beyond the i.i.d. case are possible, see e.g. Wintenberger (2010) for a survey and new results. In all these examples, the important feature of the function g that we will use to derive rates of convergence is the fact that there is a constant $c > 0$ such that when $\lambda = cn$, $g(\lambda, n) = g(cn, n) \lesssim n$.

As mentioned previously, we will often consider $r_n(\theta) = \frac{1}{n} \sum_{i=1}^n \ell(Y_i, f_\theta(X_i))$, however, the previous assumptions can also be satisfied when $r_n(\theta)$ is a U-statistic, using Hoeffding's decomposition of U-statistics combined with the corresponding inequality for sums of independent variables (Hoeffding, 1948). This idea comes from Cl emenceon et al. (2008) and we will use it in our ranking application.

Remark 2.1 We could consider more generally inequalities of the form

$$\frac{\pi(\mathbb{E} \exp \{\lambda[R(\theta) - \bar{R}] - \lambda[r_n(\theta) - \bar{r}_n]\})}{\pi(\mathbb{E} \exp \{\lambda[r_n(\theta) - \bar{r}_n] - \lambda[R(\theta) - \bar{R}]\})} \leq \pi(\exp [g(\lambda, n)[R(\theta) - \bar{R}]^\kappa])$$

that allow using the more general form of the margin assumption of Mammen and Tsybakov (1999); Tsybakov (2004). PAC-Bayes bounds in this context are provided by Catoni (2007). However, the techniques involved would require many pages to be described so we decided to focus on the cases $\kappa = 0$ and $\kappa = 1$ to keep the exposition simple.

3. Numerical approximations of the pseudo-posterior

3.1 Monte Carlo

As already explained in the introduction, the usual approach to approximate $\hat{\rho}_\lambda$ is MCMC (Markov chain Monte Carlo) sampling. Ridgway et al. (2014) proposed tempering SMC (Sequential Monte Carlo, e.g. Del Moral et al. (2006)) as an alternative to MCMC to sample from Gibbs posteriors: one samples sequentially from $\hat{\rho}_\lambda$, with $0 = \lambda_0 < \dots < \lambda_T = \lambda$ where λ is the desired temperature. One advantage of this approach is that it makes it possible to contemplate different values of λ , and choose one by e.g. cross-validation. Another advantage is that such an algorithm requires little tuning; see Appendix B for more details on the implementation of tempering SMC. We will use tempering SMC as our gold standard in our numerical studies.

SMC and related Monte Carlo algorithms tend to be too slow for practical use in situations where the sample size is large, the dimension of Θ is large, or f_θ is expensive to compute. This motivates the use of fast, deterministic approximations, such as Variational Bayes, which we describe in the next section.

3.2 Variational Bayes

Various versions of VB (Variational Bayes) have appeared in the literature, but the main idea is as follows. We define a family $\mathcal{F} \subset \mathcal{M}_+^1(\Theta)$ of probability distributions that are considered as tractable. Then, we define the VB-approximation of $\hat{\rho}_\lambda$: $\hat{\rho}_\lambda$.

Definition 3.1 Let

$$\tilde{\rho}_\lambda = \arg \min_{\rho \in \mathcal{F}} \mathcal{K}(\rho, \hat{\rho}_\lambda),$$

where $\mathcal{K}(\rho, \hat{\rho}_\lambda)$ denotes the KL (Kullback-Leibler) divergence of $\hat{\rho}_\lambda$ relative to ρ : $\mathcal{K}(m, \mu) = \int \log \frac{d\mu}{d\mu} dm$ if $m \ll \mu$ (i.e. μ dominates m), $\mathcal{K}(m, \mu) = +\infty$ otherwise.

The difficulty is to find a family \mathcal{F} (a) which is large enough, so that $\tilde{\rho}_\lambda$ may be close to $\hat{\rho}_\lambda$, and (b) such that computing $\tilde{\rho}_\lambda$ is feasible. Moreover, even when there are algorithms for $\tilde{\rho}_\lambda$ that are efficient in practice, we may, depending on the problem at hand, have more

or less strong guarantees on the quality of the optimization. For example, while in Section 6 we consider a setting where an exact upper bound for the optimization error is available, in Section 8 this is no longer the case.

We now review two types of families popular in the VB literature.

- Mean field VB: for a certain decomposition $\Theta = \Theta_1 \times \dots \times \Theta_d$, \mathcal{F} is the set of product probability measures

$$\mathcal{F}^{\text{MF}} = \left\{ \rho \in \mathcal{M}_+^1(\Theta) : \rho(d\theta) = \prod_{i=1}^d \rho_i(d\theta_i), \forall i \in \{1, \dots, d\}, \rho_i \in \mathcal{M}_+^1(\Theta_i) \right\}. \quad (3)$$

The infimum of the KL divergence $\mathcal{K}(\rho, \hat{\rho}_\lambda)$, relative to $\rho = \prod_i \rho_i$ satisfies the following fixed point condition (Parisi, 1988; Bishop, 2006, Chap. 10):

$$\forall j \in \{1, \dots, d\} \quad \rho_j(d\theta_j) \propto \exp \left(\int \{-\lambda r_n(\theta) + \log \pi(\theta)\} \prod_{i \neq j} \rho_i(d\theta_i) \right) \pi(d\theta_j). \quad (4)$$

This leads to a natural algorithm where we update successively every ρ_j until stabilization.

- Parametric family:

$$\mathcal{F}^{\text{P}} = \{\rho \in \mathcal{M}_+^1(\Theta) : \rho(d\theta) = f(\theta; m) d\theta, m \in M\};$$

and M is finite-dimensional; say \mathcal{F}^{P} is the family of Gaussian distributions (of dimension d). In this case, several methods may be used to compute the infimum. As above, one may use fixed-point iteration, provided an equation similar to (4) is available. Alternatively, one may directly maximize $\int \log[\exp\{-\lambda r_n(\theta)\} \frac{d\pi}{d\theta}(\theta)] \rho(d\theta)$ with respect to parameter m , using numerical optimization routines. This approach was used for instance in Hoffman et al. (2013) with combination of some stochastic gradient descent to perform inference on a latent Dirichlet allocation model. See also e.g. Khan (2014); Khan et al. (2013) for efficient algorithms for Gaussian variational approximation.

In what follows (Subsections 4.1 and 4.2) we provide tight bounds for the prediction risk of $\tilde{\rho}_\lambda$. This leads to the identification of a condition on \mathcal{F} such that the risk of $\tilde{\rho}_\lambda$ is not worse than the risk of $\hat{\rho}_\lambda$. We will make this condition explicit in various examples, using either mean field VB or parametric approximations.

Remark 3.1 An useful identity, obtained by direct calculations, is: for any $\rho \ll \pi$,

$$\log \int \exp[-\lambda r_n(\theta)] \pi(d\theta) = -\lambda \int r_n(\theta) \rho(d\theta) - \mathcal{K}(\rho, \pi) + \mathcal{K}(\rho, \hat{\rho}_\lambda). \quad (5)$$

Since the left hand side does not depend on ρ , one sees that $\bar{\rho}_\lambda$, which minimizes $\mathcal{K}(\rho, \hat{\rho}_\lambda)$ over \mathcal{F} , is also the minimizer of:

$$\bar{\rho}_\lambda = \arg \min_{\rho \in \mathcal{F}} \left\{ \int \tau_n(\theta) \rho(d\theta) + \frac{1}{\lambda} \mathcal{K}(\rho, \pi) \right\}$$

This equation will appear frequently in the sequel in the form of an empirical upper bound.

4. General results

This section gives our general results, under either a Hoeffding Assumption (Definition 2.3) or a Bernstein Assumption (Definition 2.4), on risks bounds for the variational approximation, and how it relates to risks bounds for Gibbs posteriors. These results will be specialised to several learning problems in the following sections.

4.1 Bounds under the Hoeffding assumption

4.1.1 EMPIRICAL BOUNDS

Theorem 4.1 *Under the Hoeffding assumption (Definition 2.3), for any $\varepsilon > 0$, with probability at least $1 - \varepsilon$ we have simultaneously for any $\rho \in \mathcal{M}_+^1(\Theta)$,*

$$\int R d\rho \leq \int \tau_n d\rho + \frac{f(\lambda, n) + \mathcal{K}(\rho, \pi) + \log\left(\frac{2}{\varepsilon}\right)}{\lambda}.$$

This result is a simple variant of a result in Catoni (2007) but for the sake of completeness, its proof is given in Appendix A. It gives us an upper bound on the risk of both the pseudo-posterior (take $\rho = \hat{\rho}_\lambda$) and its variational approximation (take $\rho = \bar{\rho}_\lambda$). These bounds may be computed from the data, and therefore provide a simple way to evaluate the performance of the corresponding procedure, in the spirit of the first PAC-Bayesian inequalities (Shawe-Taylor and Williamson, 1997; McAlester, 1998, 1999). However, these bounds do not provide the rate of convergence of these estimators. For this reason, we also provide oracle-type inequalities.

4.1.2 ORACLE-TYPE INEQUALITIES

Another way to use PAC-Bayesian bounds is to compare $\int R d\hat{\rho}_\lambda$ to the best possible risk, thus linking this approach to oracle inequalities. This is the point of view developed in Catoni (2004, 2007); Dalalyan and Tsybakov (2008).

Theorem 4.2 *Assume that the Hoeffding assumption is satisfied (Definition 2.3). For any $\varepsilon > 0$, with probability at least $1 - \varepsilon$ we have simultaneously*

$$\int R d\hat{\rho}_\lambda \leq \mathcal{B}_\lambda(\mathcal{M}_+^1(\Theta)) := \inf_{\rho \in \mathcal{M}_+^1(\Theta)} \left\{ \int R d\rho + 2 \frac{f(\lambda, n) + \mathcal{K}(\rho, \pi) + \log\left(\frac{2}{\varepsilon}\right)}{\lambda} \right\}$$

7

and

$$\int R d\bar{\rho}_\lambda \leq \mathcal{B}_\lambda(\mathcal{F}) := \inf_{\rho \in \mathcal{F}} \left\{ \int R d\rho + 2 \frac{f(\lambda, n) + \mathcal{K}(\rho, \pi) + \log\left(\frac{2}{\varepsilon}\right)}{\lambda} \right\}.$$

Moreover,

$$\mathcal{B}_\lambda(\mathcal{F}) = \mathcal{B}_\lambda(\mathcal{M}_+^1(\Theta)) + \frac{2}{\lambda} \inf_{\rho \in \mathcal{F}} \mathcal{K}(\rho, \pi_\lambda^2)$$

where we remind that π_λ is defined in Definition 2.2.

In this way, we are able to compare $\int R d\hat{\rho}_\lambda$ to the best possible aggregation procedure in $\mathcal{M}_+^1(\Theta)$ and $\int R d\bar{\rho}_\lambda$ to the best aggregation procedure in \mathcal{F} . More importantly, we are able to obtain explicit expressions for the right-hand side of these inequalities in various models, and thus to obtain rates of convergence. This will be done in the remaining sections. This leads to the second interest of this result: if there is a $\lambda = \lambda(n)$ that leads to $\mathcal{B}_\lambda(\mathcal{M}_+^1(\Theta)) \leq \bar{R} + s_n$ with $s_n \rightarrow 0$ for the pseudo-posterior $\hat{\rho}_\lambda$, then we only have to prove that there is a $\rho \in \mathcal{F}$ such that $\mathcal{K}(\rho, \pi_\lambda)/\lambda \leq c s_n$ for some constant $c > 0$ to ensure that the VB approximation $\bar{\rho}_\lambda$ also reaches the rate s_n .

We will see in the following sections several examples where the approximation does not deteriorate the rate of convergence. But first let us show the equivalent oracle inequality under the Bernstein assumption.

4.2 Bounds under the Bernstein assumption

In this context the empirical bound on the risk would depend on the minimal achievable risk $\bar{\tau}_n$ and cannot be computed explicitly. We give the oracle inequality for both the Gibbs posterior and its VB approximation in the following theorem.

Theorem 4.3 *Assume that the Bernstein assumption is satisfied (Definition 2.4). Assume that $\lambda \in I$ satisfies $\lambda - g(\lambda, n) > 0$. Then for any $\varepsilon > 0$, with probability at least $1 - \varepsilon$ we have simultaneously:*

$$\begin{aligned} \int R d\hat{\rho}_\lambda - \bar{R} &\leq \bar{\mathcal{B}}_\lambda(\mathcal{M}_+^1(\Theta)), \\ \int R d\bar{\rho}_\lambda - \bar{R} &\leq \bar{\mathcal{B}}_\lambda(\mathcal{F}), \end{aligned}$$

where, for either $\mathcal{A} = \mathcal{M}_+^1(\Theta)$ or $\mathcal{A} = \mathcal{F}$,

$$\bar{\mathcal{B}}_\lambda(\mathcal{A}) = \frac{1}{\lambda - g(\lambda, n)} \inf_{\rho \in \mathcal{A}} \left\{ [\lambda + g(\lambda, n)] \int (R - \bar{R}) d\rho + 2\mathcal{K}(\rho, \pi) + 2\log\left(\frac{2}{\varepsilon}\right) \right\}.$$

In addition,

$$\bar{\mathcal{B}}_\lambda(\mathcal{F}) = \bar{\mathcal{B}}_\lambda(\mathcal{M}_+^1(\Theta)) + \frac{2}{\lambda - g(\lambda, n)} \inf_{\rho \in \mathcal{F}} \mathcal{K}\left(\rho, \pi_{\frac{\lambda + g(\lambda, n)}{2}}\right).$$

8

The main difference with Theorem 4.2 is that the function $R(\cdot)$ is replaced by $R(\cdot) - \bar{R}$. This is well known way to obtain better rates of convergence.

5. Application to classification

5.1 Preliminaries

In all this section, we assume that $\mathcal{Y} = \{0, 1\}$ and we consider linear classification: $\Theta = \mathcal{X} = \mathbb{R}^d$, $f_\theta(x) = \mathbf{1}_{\{\theta \cdot x\} > 0}$. We put $r_n(\theta) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{\{f_\theta(X_i) \neq Y_i\}}$, $R(\theta) = \mathbb{P}(Y \neq f_\theta(X))$ and assume that the $\{(X_i, Y_i)\}_{i=1}^n$ are i.i.d. In this setting, it is well-known that the Hoeffding assumption always holds. We state as a reminder the following lemma.

Lemma 1 *Hoeffding assumption (1) is satisfied with $f(\lambda, n) = \lambda^2/(2n)$, $\lambda \in \mathbb{R}_+$.*

The proof is given in Appendix A for the sake of completeness.

It is also possible to prove that Bernstein assumption (2) holds in the case where the so-called margin assumption of Mammen and Tsybakov is satisfied. This condition we use was introduced by Tsybakov (2004) in a classification setting, based on a related definition in Mammen and Tsybakov (1999).

Lemma 2 *Assume that Mammen and Tsybakov's margin assumption is satisfied: i.e. there is a constant C such that*

$$\mathbb{E}[(\mathbf{1}_{f_\theta(X) \neq Y} - \mathbf{1}_{f_\theta(X) \neq Y})^2] \leq C[R(\theta) - \bar{R}].$$

Then Bernstein assumption (2) is satisfied with $g(\lambda, n) = \frac{C\lambda^2}{2n-\lambda}$.

Remark 5.1 *We refer the reader to Tsybakov (2004) for a proof that*

$$\mathbb{P}(0 < \langle \bar{\theta}, X \rangle \mid \leq t) \leq C't$$

for some constant $C' > 0$ implies the margin assumption. In words, when X is not likely to be in the region $\langle \bar{\theta}, X \rangle \simeq 0$, where points are hard to classify, then the problem becomes easier and the classification rate can be improved.

We propose in this context a Gaussian prior: $\pi = \mathcal{N}_d(0, \theta^2 I_d)$, and we consider a VB approach based on Gaussian families. The corresponding optimization problem is not convex, but remains feasible as we explain below.

5.2 Three sets of Variational Gaussian approximations

Consider the three following Gaussian families

$$\begin{aligned} \mathcal{F}_1 &= \left\{ \Phi_{\mathbf{m}, \sigma^2}, \mathbf{m} \in \mathbb{R}^d, \sigma^2 \in \mathbb{R}_+^* \right\}, \\ \mathcal{F}_2 &= \left\{ \Phi_{\mathbf{m}, \sigma^2}, \mathbf{m} \in \mathbb{R}^d, \sigma^2 \in (\mathbb{R}_+^*)^d \right\} \text{ (mean field approximation),} \\ \mathcal{F}_3 &= \left\{ \Phi_{\mathbf{m}, \Sigma}, \mathbf{m} \in \mathbb{R}^d, \Sigma \in \mathcal{S}^{d \times d} \right\} \text{ (full covariance approximation),} \end{aligned}$$

where $\Phi_{\mathbf{m}, \sigma^2}$ is Gaussian distribution $\mathcal{N}_d(\mathbf{m}, \sigma^2 I_d)$, $\Phi_{\mathbf{m}, \Sigma}$ is $\mathcal{N}_d(\mathbf{m}, \text{diag}(\sigma^2))$, and $\Phi_{\mathbf{m}, \Sigma}$ is $\mathcal{N}_d(\mathbf{m}, \Sigma)$. Obviously, $\mathcal{F}_1 \subset \mathcal{F}_2 \subset \mathcal{F}_3 \subset \mathcal{M}_+^d(\Theta)$, and

$$\mathcal{B}_\lambda(\mathcal{M}_+^d(\Theta)) \leq \mathcal{B}_\lambda(\mathcal{F}_3) \leq \mathcal{B}_\lambda(\mathcal{F}_2) \leq \mathcal{B}_\lambda(\mathcal{F}_1). \quad (6)$$

Note that, for the sake of simplicity, we will use the following classical notations in the rest of the paper: $\varphi(\cdot)$ is the density of $\mathcal{N}(0, 1)$ w.r.t. the Lebesgue measure, and $\Phi(\cdot)$ the corresponding c.d.f. The rest of Section 5 is organized as follows. In Subsection 5.3, we calculate explicitly $\mathcal{B}_\lambda(\mathcal{F}_2)$ and $\mathcal{B}_\lambda(\mathcal{F}_1)$. Thanks to (6) this also gives an upper bound on $\mathcal{B}_\lambda(\mathcal{F}_3)$ and proves the validity of the three types of Gaussian approximations. Then, we give details on algorithms to compute the variational approximation based on \mathcal{F}_2 and \mathcal{F}_3 , and provide a numerical illustration on real data.

5.3 Theoretical analysis

We start with the empirical bound for \mathcal{F}_2 (and \mathcal{F}_1 as a consequence), which is a direct corollary of Theorem 4.1.

Corollary 5.1 *For any $\varepsilon > 0$, with probability at least $1 - \varepsilon$ we have, for any $\mathbf{m} \in \mathbb{R}^d$, $\sigma^2 \in (\mathbb{R}_+^*)^d$,*

$$\int \text{Rd}\Phi_{\mathbf{m}, \sigma^2} \leq \int r_n \text{d}\Phi_{\mathbf{m}, \sigma^2} + \frac{\lambda}{2n} + \frac{\frac{1}{2} \sum_{i=1}^d \left[\log \left(\frac{\theta^2}{\sigma_i^2} \right) + \frac{\sigma_i^2}{\theta^2} \right] + \frac{\|\mathbf{m}\|^2}{2\theta^2} - \frac{d}{2} + \log \left(\frac{1}{\varepsilon} \right)}{\lambda}.$$

We now want to apply Theorem 4.2 in this context. In order to do so, we introduce an additional assumption.

Definition 5.1 *We say that Assumption A1 is satisfied when there is a constant $c > 0$ such that, for any $(\theta, \theta') \in \Theta^2$ with $\|\theta\| = \|\theta'\| = 1$, $\mathbb{P}\langle X, \theta \rangle \langle X, \theta' \rangle < 0 \leq c\|\theta - \theta'\|$.*

This is not a strong assumption. It is satisfied when X has an isotropic distribution, and more generally when $X/\|X\|$ has a bounded density on the unit sphere². The intuition

² If the density of $X/\|X\|$ with respect to the uniform measure on the unit sphere is upper bounded by B then $\mathbb{P}\langle X, \theta \rangle \langle X, \theta' \rangle < 0 \leq \frac{B}{2\pi} \arccos(\theta, \theta') \leq \frac{B}{2\pi} \sqrt{5-5\langle \theta, \theta' \rangle} \leq \frac{B}{2\pi} \sqrt{\frac{5}{2}} \|\theta - \theta'\|$.

beyond A1 is that for a “typical” X , a very small change in θ will only induce a change in $\text{sign}(X; \theta)$ with a small probability. When it is not satisfied, two parameters θ and θ' very close to each other can lead to very different predictions, and thus, whatever the accuracy of an approximation of θ , it might still lead to poor predictions.

Corollary 5.2 *Assume that the VB approximation is done on either \mathcal{F}_1 , \mathcal{F}_2 or \mathcal{F}_3 . Take $\lambda = \sqrt{nd}$ and $\vartheta = \frac{1}{\sqrt{n}}$. Under Assumption A1, for any $\varepsilon > 0$, with probability at least $1 - \varepsilon$ we have simultaneously*

$$\int \text{Rd}\hat{\rho}_\lambda \left\{ \leq \bar{R} + \sqrt{\frac{d}{n}} \log(4ne) + \frac{c}{\sqrt{n}} + \frac{1}{4n} \sqrt{\frac{d}{n}} + \frac{2 \log(\frac{\varepsilon}{2})}{\sqrt{nd}} \right\}.$$

See the appendix for a proof. Note also that the values $\lambda = \sqrt{nd}$ and $\vartheta = \frac{1}{\sqrt{n}}$ allow to derive this almost optimal rate of convergence, but are not necessarily the best choices in practice.

Remark 5.2 *Note that Assumption A1 is not necessary to obtain oracle inequalities on the risk integrated under $\hat{\rho}_\lambda$. We refer the reader to Chapter 1 in Catoni (2007) for such assumption-free bounds. However, it is clear that without this assumption the shape of $\hat{\rho}_\lambda$ and $\hat{\rho}_\lambda$ might be very different. Thus, it seems reasonable to require that A1 is satisfied for the approximation of $\hat{\rho}_\lambda$ by $\hat{\rho}_\lambda$ to make sense.*

We finally provide an application of Theorem 4.3. Under the additional constraint that the margin assumption is satisfied, we obtain a better rate.

Corollary 5.3 *Assume that the VB approximation is done on either \mathcal{F}_1 , \mathcal{F}_2 or \mathcal{F}_3 . Under Assumption A1 (Definition 5.1 page 10), and under Mammen and Tsybakov margin assumption, with $\lambda = \frac{2n}{C+2}$ and $\vartheta > 0$, for any $\varepsilon > 0$, with probability at least $1 - \varepsilon$,*

$$\int \text{Rd}\hat{\rho}_\lambda \left\{ \leq \bar{R} + \frac{(C+2)(C+1)}{2} \left\{ \frac{d \log \frac{n}{\vartheta}}{n} + \frac{d\vartheta}{n^2} + \frac{1}{\vartheta} - \frac{d}{\vartheta n} + \frac{2}{n} \log \frac{2}{\varepsilon} \right\} + \frac{\sqrt{d}2c(2C+1)}{n} \right\}.$$

It is possible to minimize the bound with respect to ϑ explicitly, this choice or any constant instead will lead to a rate in $d \log(n)/n$. Note that the rate d/n is minimax-optimal in this context. This is, for example, a consequence of more general results in Lecué (2007) under a general form of the margin assumption. See the Appendix for a proof.

5.4 Implementation and numerical results

For family \mathcal{F}_2 (mean field), the variational lower bound (5) equals

$$\mathcal{L}_{\lambda, \vartheta}(\mathbf{m}, \boldsymbol{\sigma}) = -\frac{\lambda}{n} \sum_{i=1}^n \Phi \left(-Y_i \frac{X_i \mathbf{m}}{\sqrt{X_i \text{diag}(\boldsymbol{\sigma}^2) X_i^T}} \right) - \frac{\mathbf{m}^T \mathbf{m}}{2\vartheta} + \frac{1}{2} \sum_{k=1}^d \left(\log \sigma_k^2 - \frac{\sigma_k^2}{\vartheta} \right),$$

11

while for family \mathcal{F}_3 (full covariance), it equals

$$\mathcal{L}_{\lambda, \vartheta}(\mathbf{m}, \Sigma) = -\frac{\lambda}{n} \sum_{i=1}^n \Phi \left(-Y_i \frac{X_i \mathbf{m}}{\sqrt{X_i \Sigma X_i^T}} \right) - \frac{\mathbf{m}^T \mathbf{m}}{2\vartheta} + \frac{1}{2} \left(\log |\Sigma| - \frac{1}{\vartheta} \text{tr} \Sigma \right).$$

Both functions are non-convex, but the multimodality of the latter may be more severe due to the larger dimension of \mathcal{F}_3 . To address this issue, we recommend using the reparametrization of Oppier and Archambeau (2009), which makes the dimension of the latter optimization problem $\mathcal{O}(n)$; see Khan (2014) for a related approach. In both cases, we found that deterministic annealing to be a good approach to optimize such non-convex functions. We refer to Appendix B for more details on deterministic annealing and on our particular implementation.

We now compare the numerical performance of the mean field and full covariance VB approximations to the Gibbs posterior (as approximated by SMC, see Section 3.1) for the classification of standard datasets; see Table 1. The datasets are all available in the UCI repository³ except for the DNA dataset which is part of the R package mlbench by Leisch and Dimitriadou (2010). When no split between the training sample is provided we split the data in half. The design matrices are centered and scaled before being used. For the Glass dataset we compare the “silicon” class against the other classes.

We also include results for a linear SVM (support vector machine) and a radial kernel SVM; the latter comparison is not entirely fair, since this is a non-linear classifier, while all the other classifiers are linear. Except for the Glass and DNA datasets, the full covariance VB approximation performs as well as or better than both SMC and SVM (while being much faster to compute, especially compared to SMC). Note that some high errors for the VB approximations can be due to the fact that the optimization of the objective is harder (we address this issue in next section).

Interestingly, VB outperforms SMC in certain cases. This might be due to the fact that a VB approximation tends to be more concentrated around the mode than the Gibbs posterior it approximates. Mean field VB does not perform so well on certain datasets (e.g. Indian). This may be due either to the approximation family being too small, or to the corresponding optimisation problem to be strongly multi-modal. We address this issue in next section.

6. Application to classification under convexified loss

Compared to the previous section, the advantage of convex classification is that the corresponding variational approximation will amount to minimizing a convex function. This

3. <https://archive.ics.uci.edu/ml/datasets.html>

12

Dataset	Covariates	Mean Field (\mathcal{F}_2)	Full cov. (\mathcal{F}_3)	SMC	SVM radial	SVM linear
Pima	7	31.0	21.3	22.3	30.4	21.6
German	60	32.0	33.6	32.0	32.0	33.2
Credit						
DNA	180	23.6	23.6	23.6	3.5	5.1
SPECTF	22	08.0	06.9	08.5	10.1	21.4
Glass	10	34.6	19.6	23.3	4.7	6.5
Indian	11	48.0	25.5	26.2	26.8	25.3
Breast	10	35.1	1.1	1.1	1.7	1.7

Table 1: Comparison of misclassification rates (%).

Misclassification rates for different datasets and for the proposed approximations of the Gibbs posterior. The last two columns are the misclassification rate given by a SVM with radial kernel and a linear SVM. The hyperparameters are chosen by cross-validation.

means that (a) the minimization problem will be easier to deal with; and (b) we will be able to compute a bound for the integrated risk after a given number of steps of the minimization procedure.

The setting is the same as in the previous section, except that for convenience we now take $\mathcal{Y} = \{-1, 1\}$, and the risk is based on the hinge loss,

$$r_n^H(\theta) = \frac{1}{n} \sum_{i=1}^n \max(0, 1 - Y_i(\theta, X_i)).$$

We will write R^H for the theoretical counterpart and \bar{R}^H for its minimum in θ . We keep the superscript H in order to allow comparison with the risk R under the 0-1 loss. We assume in this section that the X_i are uniformly bounded, that is, we have almost surely $\|X_i\|_\infty = \max_j |X_{i,j}| < c_x > 0$. Note that we do not require an assumption of the form (A1) to obtain the results of this section, as we rely directly on the Lipschitz continuity of the hinge risk.

6.1 Theoretical Results

Contrary to the previous section, the risk is not bounded in θ , and we must specify a prior distribution for the Hoeffding assumption to hold.

Lemma 3 *Under an independent Gaussian prior π such that each component is $N(0, \theta^2)$, and for $\lambda < \frac{1}{c_x} \sqrt{\frac{n}{2}}$ and with bounded design $|X_{i,j}| < c_x$, Hoeffding assumption (1) is satisfied with $f(\lambda, n) = \lambda^2 / (4n) - \frac{1}{2} \log \left(1 - \frac{\theta^2 \lambda^2 c_x^2}{2n} \right)$.*

The main impact of such a bound is that the prior variance cannot be taken too big relative to λ .

Corollary 6.1 *Assume that the VB approximation is done on either \mathcal{F}_1 , \mathcal{F}_2 or \mathcal{F}_3 . Take $\lambda = \frac{1}{c_x} \sqrt{\frac{n}{\theta^2}}$ and $\vartheta = \frac{1}{\sqrt{d}}$. For any $\varepsilon > 0$, with probability at least $1 - \varepsilon$ we have simultaneously*

$$\left. \int R^H d\hat{\rho}_\lambda \right\} \leq \bar{R}^H + \frac{c_x}{2} \sqrt{\frac{d}{n}} \log \frac{n}{d} + c_x \frac{d}{n} \sqrt{\frac{d}{n}} + \frac{1}{\sqrt{nd}} \left(\frac{2c_x^2 + 1}{2c_x} + 2c_x \log \frac{2}{\varepsilon} \right)$$

The oracle inequality in the above corollary enjoys the same rate of convergence as the equivalent result in the preceding section. In the following we link the two results.

Remark 6.1 *As stated in the beginning of the section we can use the estimator specified under the hinge loss to bound the excess risk of the 0-1 loss. We write R^* and R^{H*} the respective risk for their corresponding Bayes classifiers. From Zhang (2004) (section 3.3) we have the following inequality, linking the excess risk under the hinge loss and the 0-1 loss,*

$$R(\theta) - R^* \leq R^H(\theta) - R^{H*}$$

for every $\theta \in \mathbb{R}^p$. By integrating with respect to $\hat{\rho}^H$ (the VB approximation on any $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3$ of the Gibbs posterior for the hinge risk) and making use of Corollary 6.1 we have with high probability,

$$\hat{\rho}^H(R(\theta)) - R^* \leq \inf_{\theta \in \mathbb{R}^p} R^H(\theta) - R^{H*} + \mathcal{O} \left(\sqrt{\frac{d}{n}} \log \left(\frac{n}{d} \right) \right).$$

6.2 Numerical application

We have motivated the introduction of the hinge loss as a convex upper bound. In the sequel we show that the resulting VB approximation also leads to a convex optimization problem. This has the advantage of opening a range of possible optimization algorithms (Nesterov, 2004). In addition we are able to bound the error of the approximated measure after a fixed number of iterations (see Theorem 6.2).

Under the model \mathcal{F}_1 each individual risk is given by:

$$\rho_{\mathbf{m}, \sigma}(r_i(\theta)) = (1 - \Gamma_i \mathbf{m}) \Phi \left(\frac{1 - \Gamma_i \mathbf{m}}{\sigma \|\Gamma_i\|_2} \right) + \sigma \|\Gamma_i\|_2 \varphi \left(\frac{1 - \Gamma_i \mathbf{m}}{\sigma \|\Gamma_i\|_2} \right) := \Xi_i \left(\begin{pmatrix} \mathbf{m} \\ \sigma \end{pmatrix} \right),$$

writing $\Gamma_i := Y_i X_i$.

Hence the lower bound to be maximized is given by

$$\mathcal{L}(\mathbf{m}, \sigma) = -\frac{\lambda}{n} \left\{ \sum_{i=1}^n (1 - \Gamma_i \mathbf{m}) \Phi \left(\frac{1 - \Gamma_i \mathbf{m}}{\sigma \|\Gamma_i\|_2} \right) + \sum_{i=1}^n \sigma \|\Gamma_i\|_\varphi \left(\frac{1 - \Gamma_i \mathbf{m}}{\sigma \|\Gamma_i\|_2} \right) \right\} - \frac{\|\mathbf{m}\|_2^2}{2\theta} + \frac{d}{2} \left(\log \sigma^2 - \frac{\theta}{\sigma^2} \right).$$

It is easy to see that the function is convex in (\mathbf{m}, σ) , first note that the map

$$\Psi : \begin{pmatrix} x \\ y \end{pmatrix} \mapsto x \Phi \left(\frac{x}{y} \right) + y \varphi \left(\frac{x}{y} \right),$$

is convex and note that we can write $\Xi_i \left(\begin{pmatrix} \mathbf{m} \\ \sigma \end{pmatrix} \right) = \Psi \left(A \begin{pmatrix} x \\ y \end{pmatrix} + b \right)$ hence by composition of convex function with linear mappings we have the result. Similar reasoning could be held for the case \mathcal{F}_2 and \mathcal{F}_3 , where in later the parametrization should be done in \mathbb{C} such that $\Sigma = CC^*$. The bound is however not universally Lipschitz in σ , this impacts the optimization algorithms. In Theorem 6.2 we define a ball around the optimal value of the objective, containing the initial values. We denote it's radius by M . On this ball the objective is Lipschitz (with coefficient L) and optimal convex solvers can be used (e.g. Nesterov (2004) section 3.2.3).

On the class of function $\mathcal{F}_0 = \left\{ \Phi_{\mathbf{m}, \frac{1}{\sigma}}, \mathbf{m} \in \mathbb{R}^d \right\}$, for which our Oracle inequalities still hold we could get faster numerical algorithms. The objective function has Lipschitz continuous derivatives and we would get a rate of $\frac{L}{1+8L^2}$.

Other convex loss could be considered which could lead to convex optimization problems. For instance one could consider the exponential loss.

Theorem 6.2 *Assume that the VB approximation is based on either $\mathcal{F}_1, \mathcal{F}_2$ or \mathcal{F}_3 . Denote by $\tilde{\rho}_k(d\theta)$ the VB approximated measure after the k th iteration of an optimal convex solver using the hinge loss. Fix $M > 0$ large enough so that the optimal approximated mean and variance $\tilde{m}, \tilde{\Sigma}$ are at distance at most M from the initial value used by the solver. Take $\lambda = \sqrt{nd}$ and $\theta = \frac{1}{\lambda}$ then under the hypothesis of Corollary 6.1 with probability $1 - \epsilon$*

$$\int R^H d\tilde{\rho}_k \leq \bar{R}^H + \frac{LM}{\sqrt{1+k}} + \frac{c_x \sqrt{d}}{2} \log \frac{n}{d} + c_x \frac{d}{n} \sqrt{\frac{d}{n}} + \frac{1}{\sqrt{nd}} \left(\frac{2c_x^2 + 1}{2c_x} + 2c_x \log \frac{2}{\epsilon} \right)$$

where L is the Lipschitz coefficient on a ball of radius M defined above.

Note that this result is stronger and more practical than the previous ones: it ensures a certain error level (with fixed probability $1 - \epsilon$) for the k -th iterate of the optimization algorithm, for a known value of k . In contrast, previous results applied to the output of the optimizer "for k large enough".

We find that on average the misclassification error (Table 2) is lower than for the 0-1 loss where we have no guarantees that the maximum is attained.

Dataset	Covariates	Hinge loss	SMC
Pima	7	19.5	22.3
Credit	60	26.2	32.0
DNA	180	4.2	23.6
SPECTF	22	10.1	08.5
Glass	10	2.8	23.3
Indian	11	25.5	25.5
Breast	10	0.5	1.1

Table 2: Comparison of misclassification rates (%). Misclassification rates for different datasets and for the proposed approximations of the Gibbs posterior. The hyperparameters are chosen by cross-validation. This is to be compared to Table 1. The variational Bayes approximation was computed using the R package we developed (see the introduction for a reference).

7. Application to ranking

7.1 Preliminaries

We now focus on the ranking problem. We follow Clémenton et al. (2008) for the definitions of the basic concepts: $\mathcal{Y} = \{0, 1\}$, $\Theta = \mathbb{R}^d$ and $f_\theta : \mathcal{X}^2 \rightarrow \{-1, +1\}$ for $\theta \in \Theta$; $f_\theta(x, x') = 1$ (resp. -1) means that x is more (resp. less) likely to correspond to label 1 than x' . The natural risk function is then

$$R(\theta) = \mathbb{P}[(Y_1 - Y_2)f_\theta(X_1, X_2) < 0]$$

and the empirical risk

$$r_n(\theta) = \frac{1}{n(n-1)} \sum_{1 \leq i \neq j \leq n} \mathbf{1}_{(Y_i - Y_j)f_\theta(X_i, X_j) < 0}.$$

Again, we recall classical results.

Lemma 4 *The Hoeffding-type assumption is satisfied with $f(\lambda, n) = \frac{\lambda^2}{2}$.*

The variant of the margin assumption adapted to ranking was established by Robbiano (2013) and Ridgway et al. (2014).

Lemma 5 *Assume the following margin assumption:*

$$\mathbb{E}[\mathbf{1}_{f_\theta(X_1, X_2)(Y_1 - Y_2) < 0} - \mathbf{1}_{f_\theta(X_1, X_2)(Y_1 - Y_2) < 0}]^2 \leq C[R(\theta) - \bar{R}].$$

Then Bernstein assumption (2) is satisfied with $g(\lambda, n) = \frac{C\lambda^2}{n-1-4\lambda}$.

We focus on linear classifiers, $f_\theta(x, x') = -1 + 2 \times \mathbf{1}_{\langle \theta, x \rangle > \langle \theta, x' \rangle}$. Like in the classification setting, $\langle x, \theta \rangle$ is interpreted as a score related to the probability that $Y = 1|X = x$. We consider a Gaussian prior

$$\pi(d\theta) = \prod_{i=1}^d \varphi(\theta_i; 0, \vartheta^2) d\theta_i$$

and the approximation families will be the same as in Section 5: $\mathcal{F}_1 = \{\Phi_{\mathbf{m}, \sigma^2}, \mathbf{m} \in \mathbb{R}^d, \sigma^2 \in \mathbb{R}_+^*\}$, $\mathcal{F}_2 = \{\Phi_{\mathbf{m}, \sigma^2}, \mathbf{m} \in \mathbb{R}^d, \sigma^2 \in (\mathbb{R}_+^*)^2\}$ and $\mathcal{F}_3 = \{\Phi_{\mathbf{m}, \Sigma}, \mathbf{m} \in \mathbb{R}^d, \Sigma \in \mathcal{S}^{d \times d}\}$.

7.2 Theoretical study

Here again, we start with the empirical bound.

Corollary 7.1 *For any $\varepsilon > 0$, with probability at least $1 - \varepsilon$ we have, for any $\mathbf{m} \in \mathbb{R}^d$, $\sigma^2 \in (\mathbb{R}_+^*)^d$,*

$$\int \text{Rd}\Phi_{\mathbf{m}, \sigma^2} \leq \int r_n d\Phi_{\mathbf{m}, \sigma^2} + \frac{\lambda}{n-1} + \frac{\frac{1}{2} \sum_{j=1}^d \left[\log \left(\frac{\vartheta^2}{\sigma_j^2} \right) + \frac{\sigma_j^2}{\vartheta^2} \right] + \frac{\|\mathbf{m}\|^2}{2\vartheta^2} - \frac{d}{2} + \log \left(\frac{1}{\varepsilon} \right)}{\lambda}.$$

In order to derive a theoretical bound, we introduce the following variant of Assumption A1.

Definition 7.1 *We say that Assumption A2 is satisfied when there is a constant $c > 0$ such that, for any $(\theta, \theta') \in \Theta^2$ with $\|\theta\| = \|\theta'\| = 1$, $\mathbb{P}(\langle X_1 - X_2, \theta \rangle \langle X_1 - X_2, \theta' \rangle < 0) \leq c\|\theta - \theta'\|$. Assumption A2 is just Assumption A1 applied to the distribution of $(X_1 - X_2)$. Intuitively, it means that two parameters close to each other rank X_1 and X_2 in the same way (with large probability).*

Corollary 7.2 *Use either \mathcal{F}_1 , \mathcal{F}_2 or \mathcal{F}_3 . Take $\lambda = \sqrt{\frac{d(n-1)}{2}}$ and $\vartheta = 1$. Under (A2), for any $\varepsilon > 0$, with probability at least $1 - \varepsilon$,*

$$\left. \int \text{Rd}\hat{\rho}_\lambda \right\} \leq \bar{R} + \sqrt{\frac{2d}{n-1}} \left(1 + \frac{1}{2} \log(2d(n-1)) \right) + \frac{c\sqrt{2}}{\sqrt{n-1}} + \frac{1}{(n-1)^{3/2}\sqrt{2d}} + \frac{2\sqrt{2}\log\left(\frac{2e}{\varepsilon}\right)}{\sqrt{(n-1)d}}.$$

Finally, under an additional margin assumption, we have:

Corollary 7.3 *Under Assumption A2 and the margin assumption of Lemma (5), for $\lambda = \frac{n-1}{C+5}$ and $\vartheta > 0$, for any $\varepsilon > 0$, with probability at least $1 - \varepsilon$,*

$$\left. \int \text{Rd}\hat{\rho}_\lambda \right\} \leq \bar{R} + \frac{(C+5)(C+1)}{2} \left\{ \frac{d \log \frac{n}{\vartheta}}{n-1} + \frac{d\theta}{n(n-1)} + \frac{1}{\vartheta} - \frac{d}{\vartheta n-1} + \frac{2}{n-1} \log \frac{2}{\varepsilon} \right\} + \frac{\sqrt{d}4c(C+1)}{n}.$$

It is possible to optimize the bound with respect to ϑ . The proof is similar to the ones of Corollaries 5.2, 5.3 and 7.2.

As in the case of classification, ranking under an AUC loss can be done by replacing the indicator function by the corresponding upper bound given by a hinge loss. In this case we can derive similar results as for the convexified classification in particular we can get a convex minimization problem and obtain result without requiring assumption (A2).

7.3 Algorithms and numerical results

As an illustration we focus here on family \mathcal{F}_2 (mean field). In this case the VB objective to maximize is given by:

$$\mathcal{L}(\mathbf{m}, \sigma^2) = -\frac{\lambda}{n_+ n_-} \sum_{i:Y_i=1, \vartheta:Y_i=0} \Phi \left(-\frac{\Gamma_{ij} \mathbf{m}}{\sqrt{\sum_{k=1}^d (\gamma_{ij}^k)^2 \sigma_k^2}} \right) - \frac{\|\mathbf{m}\|_2^2}{2\vartheta} + \frac{1}{2} \sum_{k=1}^d \left[\log \sigma_k^2 - \frac{\sigma_k^2}{\vartheta} \right], \quad (7)$$

where $\Gamma_{ij} = X_i - X_j$, $n_+ = \text{card}\{1 \leq i \leq n : Y_i = 1\}$, $n_- = n - n_+ = \text{card}\{1 \leq i \leq n : Y_i = 0\}$ and where $(\gamma_{ij}^k)_k$ are the elements of Γ .

This function is expensive to compute, as it involves $n_+ n_-$ terms, the computation of which is $\mathcal{O}(p)$.

We propose to use a stochastic gradient descent in the spirit of Hoffman et al. (2013). The model we consider is not in an exponential family, meaning we cannot use the trick developed by these authors. We propose instead to use a standard descent.

The idea is to replace the gradient by an unbiased version based on a batch of size B as described in Algorithm 4 in the Appendix. Robbins and Monro (1951) show that for a step-size $(\lambda_t)_t$ such that $\sum_t \lambda_t^2 < \infty$ and $\sum_t \lambda_t = \infty$ the algorithm converges to a local optimum.

In our case we propose to sample pairs of data with replacement and use the unbiased version of the derivative of the risk component. We use a simple gradient descent without any curvature information. One could also use recent research on stochastic quasi-Newton-Raphson (Byrd et al., 2014).

For illustration, we consider a small dataset (Pima), and a larger one (Adult). Both datasets are available in the UCI repository⁴. As for the previous experiment the data is scaled and centered. The latter is already quite challenging with $n_+ n_- = 193, 829, 520$ pairs to compare. In both cases with different size of batches convergence is obtained with a few iterations only and leads to acceptable bounds.

In Figure 1 we show the empirical bound on the AUC risk as a function of the iteration of the algorithm, for several batch sizes. The bound is taken for 95% probability, the batch sizes are taken to be $B = 1, 10, 20, 50$ for the Pima dataset, and 50 for the Adult dataset. The figure shows an additional feature of VB approximation in the context of

4. <https://archive.ics.uci.edu/ml/datasets.html>

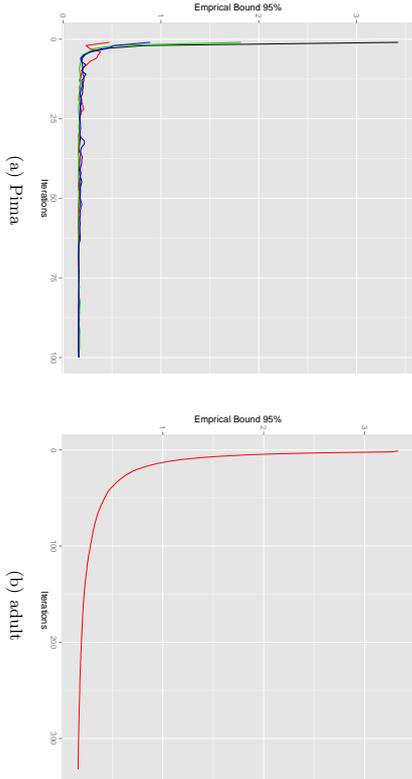


Figure 1: Error bound at each iteration, stochastic descent, Pima and Adult datasets.

Stochastic VB with fixed temperature $\lambda = 100$ for Pima and $\lambda = 1000$ for adult. The left panel shows several curves that correspond to different batch sizes; these curves are hard to distinguish. The right panel is for a batch size of 50. The adult dataset has $n = 32556$ observation and $n+n_- = 193829520$ possible pairs. The convergence is obtained in order of seconds. The bounds are the empirical bounds obtained in Corollary 7.1 for a probability of 95%.

Gibbs posterior: namely the possibility of computing the empirical upper bound given by Corollary 7.1. That is we can check the quality of the bound at each iteration of the algorithm, or for different values of the hyperparameters.

8. Application to matrix completion

The matrix completion problem has received increasing attention recently, partly due to spectacular theoretical results (Candes and Tao, 2010), and to challenging applications like the Netflix challenge (Bennett and Lanning, 2007). In the perspective of this paper, the specific interest of this application is twofold. First, this is a case where the family of approximations is not parametric, but rather of the form (3), i.e. the family of products of independent components. Then, there is no known theoretical result for the Gibbs estimator in the considered model, yet we can still directly bound the loss induced by the variational approximation.

We observe i.i.d. pairs $((X_i, Y_i))_{i=1}^n$ where $X_i \in \{1, \dots, m_1\} \times \{1, \dots, m_2\}$, and we assume that there is a $m_1 \times m_2$ -matrix M such that $Y_i = M_{X_i} + \varepsilon_i$ and the ε_i are centred. Assuming that X_i is uniform on $\{1, \dots, m_1\} \times \{1, \dots, m_2\}$, that $f_\theta(X_i) = \theta_{X_i}$, and taking the quadratic risk, $R(\theta) = \mathbb{E} [Y_i - \theta_{X_i}]^2$, we have that

$$R(\theta) - \bar{R} = \frac{1}{m_1 m_2} \|\theta - M\|_F^2$$

where $\|\cdot\|_F$ stands for the Frobenius norm.

A common way to parameterize the problem is

$$\Theta = \{\theta = UV^T, U \in \mathbb{R}^{m_1 \times K}, V \in \mathbb{R}^{m_2 \times K}\}$$

where K is large; e.g. $K = \min(m_1, m_2)$. Following Salakhutdinov and Mnih (2008), we define the following prior distribution: $U_{:,j} \sim \mathcal{N}(0, \gamma_j I)$, $V_{:,j} \sim \mathcal{N}(0, \gamma_j I)$ where the γ_j 's are i.i.d. from an inverse gamma distribution, $\gamma_j \sim \text{IG}(a, b)$.

Note that VB algorithms were used in this context by Lin and Teh (2007) (with a slightly simpler prior however: the γ_j 's are fixed rather than random). Since then, this prior and variants were used in several papers (e.g. Lawrence and Urtasun, 2009; Zhou et al., 2010). Until now, no theoretical results were proved to the best of our knowledge. Two papers prove minimax-optimal rates for slightly modified estimators (by truncation), for which efficient algorithms are unknown (Mai and Alquier, 2015; Suzuki, 2014). However, using Theorems 4.2 and 4.3 we are able to prove the following: if there is a PAC-Bayesian bound leading to a rate for $\hat{\rho}_\lambda$ in this context, then the same rate holds for $\hat{\rho}_\lambda$. In other words: if someone proves the conjecture that the Gibbs estimator is minimax-optimal (up to log terms) in this context, then the VB approximation will enjoy automatically the same property.

We propose the following approximation:

$$\mathcal{F} = \left\{ \rho(d(U, V)) = \prod_{i=1}^{m_1} u_i(dU_{:,i}) \prod_{j=1}^{m_2} v_j(dV_{:,j}) \right\}.$$

Theorem 8.1 *Assume that $M = UV^T$ with $|U_{i,k}|, |V_{j,k}| \leq C$. Assume that $\text{rank}(M) = r$ so that we can assume that $U_{:,r+1} = \dots = U_{:,K} = V_{:,r+1} = \dots = V_{:,K} = 0$ (note that the prior π does not depend on the knowledge of r through). Choose the prior distribution on the hyperparameters γ_j as inverse gamma $\text{Inv}^{-1}(a, b)$ with $b \leq 1/[2\beta(m_1 \vee m_2) \log(2K(m_1 \vee m_2))]$. Then there is a constant $C(a, C)$ such that, for any $\beta > 0$,*

$$\inf_{\rho \in \mathcal{F}} \mathcal{K}(\rho, \pi_\beta) \leq C(a, C) \left\{ r(m_1 + m_2) \log[\beta k(m_1 + m_2)K] + \frac{1}{\beta} \right\}.$$

See the Appendix for a proof.

For instance, in Theorem 4.3, in classification and ranking we had λ , $\lambda - g(\lambda, n)$ and $\lambda + g(\lambda, n)$ of order $\mathcal{O}(n)$. In this case we would have:

$$\frac{2}{\lambda - g(\lambda, n)} \inf_{\rho \in \mathcal{F}} \mathcal{K} \left(\rho, \pi_{\frac{\lambda + g(\lambda, n)}{2}} \right) = \mathcal{O} \left(\frac{\mathcal{C}(a, C)^r (m_1 + m_2) \log \left[\frac{nb(m_1 + m_2)K}{n} \right]}{n} \right),$$

and note that in this context it is known that the minimax rate is at least $r(m_1 + m_2)/n$ (Koltchinskii et al., 2011).

8.1 Algorithm

As already mentioned, the approximation family is not parametric in this case, but rather of type mean field. The corresponding VB algorithm amounts to iterating equation (4), which takes the following form in this particular case:

$$\begin{aligned} u_j(dU_{j,\cdot}) &\propto \exp \left\{ -\frac{\lambda}{n} \sum_i \mathbb{E}_{V_i U_{-j}} [(Y_{X_i} - (UV^T)_{X_i})^2] - \sum_{k=1}^K \mathbb{E}_{\gamma_j} \left[\frac{1}{2\gamma_k} U_{jk}^2 \right] \right\} \\ v_j(dV_{j,\cdot}) &\propto \exp \left\{ -\frac{\lambda}{n} \sum_i \mathbb{E}_{V_{-j} U} [(Y_{X_i} - (UV^T)_{X_i})^2] - \sum_{k=1}^K \mathbb{E}_{\gamma_j} \left[\frac{1}{2\gamma_k} V_{jk}^2 \right] \right\} \\ p(\gamma_k) &\propto \exp \left\{ -\frac{1}{2\gamma_k} \left(\sum_j \mathbb{E}_U U_{kj}^2 + \sum_i \mathbb{E}_V V_{ik}^2 \right) + (\alpha + 1) \log \frac{1}{\gamma_k} - \frac{\beta}{\gamma_k} \right\} \end{aligned}$$

where the expectations are taken with respect to the thus defined variational approximations. One recognises Gaussian distributions for the first two, and an inverse Gamma distribution for the third. We refer to Lim and Teh (2007) for more details on this algorithm and for a numerical illustration. However, we point out that in this case, while the algorithm seems to work well in practice, there is no theoretical guarantee that it will converge to the global minimum of the problem.

9. Discussion

We showed in several important scenarios that approximating a Gibbs posterior through VB (Variational Bayes) techniques does not deteriorate the rate of convergence of the corresponding procedure. We also described practical algorithms for fast computation of these VB approximations, and provided empirical bounds that may be computed from the data to evaluate the performance of the so-obtained VB-approximated procedure. We believe these results provide a strong incentive to recommend VB as the default approach to approximate Gibbs posteriors, in lieu of Monte Carlo methods. We also developed a R package⁵ for convexified losses (classification and bipartite ranking), applying the ideas of Section 6.

5. PACVB package: <https://cran.r-project.org/web/packages/PACVB/index.html>

We hope to extend our results to other applications beyond those discussed in this paper, such as regression. One technical difficulty with regression is that the risk function is not bounded, which makes our approach a bit less direct to apply. In many papers on PAC-Bayesian bounds for regression, the noise can be unbounded (usually, it is assumed to be sub-exponential), but one assumes that the predictors are bounded, see e.g. Alquier and Bian (2013). However, using the robust loss function of Audibert and Catoni, it is possible to relax this assumption (Audibert and Catoni, 2011; Catoni, 2012). This requires a more technical analysis, which we leave for further work.

Appendix A. Proofs

A.1 Preliminary remarks

Direct calculation yields, for any $\rho \ll \pi$ with $\int r_n d\rho < \infty$,

$$\mathcal{K}(\rho, \pi[r_n]) = \lambda \int r_n d\rho + \mathcal{K}(\rho, \pi) + \log \int \exp(-h) d\pi.$$

Two well known consequences are

$$\begin{aligned} \pi[h] &= \arg \min_{\rho \in \mathcal{M}_+^1(\mathfrak{e})} \left\{ \int h d\rho + \mathcal{K}(\rho, \pi) \right\}, \\ -\log \int \exp(-h) d\pi &= \min_{\rho \in \mathcal{M}_+^1(\mathfrak{e})} \left\{ \int h d\rho + \mathcal{K}(\rho, \pi) \right\}. \end{aligned}$$

We will use these inequalities many times in the followings. The most frequent application will be with $h(\theta) = \lambda r_n(\theta)$ (in this case $\pi[\lambda r_n] = \hat{\rho}_\lambda$) or $h(\theta) = \pm \lambda[r_n(\theta) - R(\theta)]$, the first case leads to

$$\mathcal{K}(\rho, \hat{\rho}_\lambda) = \lambda \int r_n d\rho + \mathcal{K}(\rho, \pi) + \log \int \exp(-\lambda r_n) d\pi, \quad (8)$$

$$\hat{\rho}_\lambda = \arg \min_{\rho \in \mathcal{M}_+^1(\mathfrak{e})} \left\{ \lambda \int r_n d\rho + \mathcal{K}(\rho, \pi) \right\}, \quad (9)$$

$$-\log \int \exp(-\lambda r_n) d\pi = \min_{\rho \in \mathcal{M}_+^1(\mathfrak{e})} \left\{ \lambda \int r_n d\rho + \mathcal{K}(\rho, \pi) \right\}. \quad (10)$$

We will use (8), (9) and (10) several times in this appendix.

A.2 Proof of the theorems in Subsection 4.1

Proof of Theorem 4.1. This proof follows the standard PAC-Bayesian approach (see Catoni (2007)). Apply Fubini's theorem to the first inequality of (1):

$$\mathbb{E} \int \exp \{ \lambda [R(\theta) - r_n(\theta)] - f(\lambda, n) \} \pi(d\theta) \leq 1$$

then apply the preliminary remark with $h(\theta) = \lambda[r_n(\theta) - R(\theta)]$:

$$\mathbb{E} \exp \left\{ \sup_{\rho \in \mathcal{M}_+^1(\Theta)} \int \lambda [R(\theta) - r_n(\theta)] \rho(d\theta) - \mathcal{K}(\rho, \pi) - f(\lambda, n) \right\} \leq 1.$$

Multiply both sides by ε and use $\mathbb{E}[\exp(U)] \geq \mathbb{P}(U > 0)$ for any U to obtain:

$$\mathbb{P} \left[\sup_{\rho \in \mathcal{M}_+^1(\Theta)} \int \lambda [R(\theta) - r_n(\theta)] \rho(d\theta) - \mathcal{K}(\rho, \pi) + \log(\varepsilon) > 0 \right] \leq \varepsilon.$$

Then consider the complementary event:

$$\mathbb{P} \left[\forall \rho \in \mathcal{M}_+^1(\Theta), \quad \lambda \int R d\rho \leq \lambda \int r_n d\rho + f(\lambda, n) + \mathcal{K}(\rho, \pi) + \log \left(\frac{1}{\varepsilon} \right) \right] \geq 1 - \varepsilon.$$

□

Proof of Theorem 4.2. Using the same calculations as above, we have, with probability at least $1 - \varepsilon$, simultaneously for all $\rho \in \mathcal{M}_+^1(\Theta)$,

$$\lambda \int R d\rho \leq \lambda \int r_n d\rho + f(\lambda, n) + \mathcal{K}(\rho, \pi) + \log \left(\frac{2}{\varepsilon} \right) \quad (11)$$

$$\lambda \int r_n d\rho \leq \lambda \int R d\rho + f(\lambda, n) + \mathcal{K}(\rho, \pi) + \log \left(\frac{2}{\varepsilon} \right). \quad (12)$$

We use (11) with $\rho = \hat{\rho}_\lambda$ and (9) to get

$$\lambda \int R d\hat{\rho}_\lambda \leq \inf_{\rho \in \mathcal{M}_+^1(\Theta)} \left\{ \lambda \int r_n d\rho + f(\lambda, n) + \mathcal{K}(\rho, \pi) + \log \left(\frac{2}{\varepsilon} \right) \right\}$$

and plugging (12) into the right-hand side, we obtain

$$\lambda \int R d\hat{\rho}_\lambda \leq \inf_{\rho \in \mathcal{M}_+^1(\Theta)} \left\{ \lambda \int R d\rho + 2f(\lambda, n) + 2\mathcal{K}(\rho, \pi) + 2 \log \left(\frac{2}{\varepsilon} \right) \right\}.$$

Now, we work with $\hat{\rho}_\lambda = \arg \min_{\rho \in \mathcal{F}} \mathcal{K}(\rho, \hat{\rho}_\lambda)$. Plugging (8) into (11) we get, for any ρ ,

$$\lambda \int R d\rho \leq f(\lambda, n) + \mathcal{K}(\rho, \hat{\rho}_\lambda) - \log \int \exp(-\lambda r_n) d\pi + \log \left(\frac{2}{\varepsilon} \right).$$

By definition of $\hat{\rho}_\lambda$, we have:

$$\lambda \int R d\hat{\rho}_\lambda \leq \inf_{\rho \in \mathcal{F}} \left\{ f(\lambda, n) + \mathcal{K}(\rho, \hat{\rho}_\lambda) - \log \int \exp(-\lambda r_n) d\pi + \log \left(\frac{2}{\varepsilon} \right) \right\}$$

and, using (8) again, we obtain:

$$\lambda \int R d\hat{\rho}_\lambda \leq \inf_{\rho \in \mathcal{F}} \left\{ \lambda \int r_n d\rho + f(\lambda, n) + \mathcal{K}(\rho, \pi) + \log \left(\frac{2}{\varepsilon} \right) \right\}.$$

We plug (12) into the right-hand side to obtain:

$$\lambda \int R d\hat{\rho}_\lambda \leq \inf_{\rho \in \mathcal{F}} \left\{ \lambda \int R d\rho + 2f(\lambda, n) + 2\mathcal{K}(\rho, \pi) + 2 \log \left(\frac{2}{\varepsilon} \right) \right\}.$$

This proves the second inequality of the theorem. In order to prove the claim

$$\mathcal{B}_\lambda(\mathcal{F}) = \mathcal{B}_\lambda(\mathcal{M}_+^1(\Theta)) + \frac{2}{\lambda} \inf_{\rho \in \mathcal{F}} \mathcal{K}(\rho, \pi_{\frac{\lambda}{2}}),$$

note that

$$\begin{aligned} \mathcal{B}_\lambda(\mathcal{F}) &= \inf_{\rho \in \mathcal{F}} \left\{ \int R d\rho + \frac{2f(\lambda, n)}{\lambda} + \frac{2\mathcal{K}(\rho, \pi)}{\lambda} + \frac{2 \log \left(\frac{2}{\varepsilon} \right)}{\lambda} \right\} \\ &= \inf_{\rho \in \mathcal{F}} \left\{ -\frac{2}{\lambda} \log \int \exp \left(-\frac{\lambda}{2} R \right) d\pi + \frac{2f(\lambda, n)}{\lambda} + \frac{2\mathcal{K}(\rho, \pi_{\frac{\lambda}{2}})}{\lambda} + \frac{2 \log \left(\frac{2}{\varepsilon} \right)}{\lambda} \right\} \\ &= -\frac{2}{\lambda} \log \int \exp \left(-\frac{\lambda}{2} R \right) d\pi + \frac{2f(\lambda, n)}{\lambda} + \frac{2 \log \left(\frac{2}{\varepsilon} \right)}{\lambda} + \frac{2}{\lambda} \inf_{\rho \in \mathcal{F}} \mathcal{K}(\rho, \pi_{\frac{\lambda}{2}}) \\ &= \mathcal{B}_\lambda(\mathcal{M}_+^1(\Theta)) + \frac{2}{\lambda} \inf_{\rho \in \mathcal{F}} \mathcal{K}(\rho, \pi_{\frac{\lambda}{2}}). \end{aligned}$$

This ends the proof. □

A.3 Proof of Theorem 4.3 (Subsection 4.2)

Proof of Theorem 4.3. As in the proof of Theorem 4.1, we apply Fubini, then (10) to the first inequality of (2) to obtain

$$\mathbb{E} \exp \left\{ \sup_{\rho} \int [\lambda [R(\theta) - \bar{R}] - \lambda [r_n(\theta) - \bar{\pi}_n] - g(\lambda, n)] [R(\theta) - \bar{R}] \rho(d\theta) - \mathcal{K}(\rho, \pi) \right\} \leq 1$$

and we multiply both sides by $\varepsilon/2$ to get

$$\mathbb{P} \left\{ \sup_{\rho} \left[\lambda - g(\lambda, n) \right] \left[\int R d\rho - \bar{R} \right] \geq \lambda \left[\int r_n d\rho - \bar{\pi}_n \right] + \mathcal{K}(\rho, \pi) + \log \left(\frac{2}{\varepsilon} \right) \right\} \leq \frac{\varepsilon}{2}. \quad (13)$$

We now consider the second inequality in (2):

$$\mathbb{E} \exp \left\{ \lambda [r_n(\theta) - \bar{\pi}_n] - \lambda [R(\theta) - \bar{R}] - g(\lambda, n) [R(\theta) - \bar{R}] \right\} \leq 1.$$

The same derivation leads to

$$\mathbb{P} \left\{ \sup_{\rho} \left[\lambda - g(\lambda, n) \right] \left[\int r_n d\rho - \bar{r}_n \right] \geq \lambda \left[\int R d\rho - \bar{R} \right] + \mathcal{K}(\rho, \pi) + \log \left(\frac{2}{\varepsilon} \right) \right\} \leq \frac{\varepsilon}{2}. \quad (14)$$

We combine (13) and (14) by a union bound argument, and we consider the complementary event: with probability at least $1 - \varepsilon$, simultaneously for all $\rho \in \mathcal{M}_+^1(\Theta)$,

$$\left| \lambda - g(\lambda, n) \right| \left[\int R d\rho - \bar{R} \right] \leq \lambda \left[\int r_n d\rho - \bar{r}_n \right] + \mathcal{K}(\rho, \pi) + \log \left(\frac{2}{\varepsilon} \right), \quad (15)$$

$$\lambda \left[\int r_n d\rho - \bar{r}_n \right] \leq \left[\lambda + g(\lambda, n) \right] \left[\int R d\rho - \bar{R} \right] + \mathcal{K}(\rho, \pi) + \log \left(\frac{2}{\varepsilon} \right). \quad (16)$$

We now derive consequences of these two inequalities (in other words, we focus on the event where these two inequalities are satisfied). Using (9) in (15) yields

$$\left| \lambda - g(\lambda, n) \right| \left[\int R d\hat{\rho}_\lambda - \bar{R} \right] \leq \inf_{\rho \in \mathcal{M}_+^1(\Theta)} \left\{ \lambda \left[\int r_n d\rho - \bar{r}_n \right] + \mathcal{K}(\rho, \pi) + \log \left(\frac{2}{\varepsilon} \right) \right\}.$$

We plug (16) into the right-hand side to obtain:

$$\begin{aligned} \left| \lambda - g(\lambda, n) \right| \left[\int R d\hat{\rho}_\lambda - \bar{R} \right] \\ \leq \inf_{\rho \in \mathcal{M}_+^1(\Theta)} \left\{ \left[\lambda + g(\lambda, n) \right] \left[\int R d\rho - \bar{R} \right] + 2\mathcal{K}(\rho, \pi) + 2 \log \left(\frac{2}{\varepsilon} \right) \right\}. \end{aligned}$$

Now, we work with $\hat{\rho}_\lambda$. Plugging (8) into (13) we get

$$\left| \lambda - g(\lambda, n) \right| \left[\int R d\rho - \bar{R} \right] \leq \mathcal{K}(\rho, \hat{\rho}_\lambda) - \log \int \exp[-\lambda(r_n - \bar{r}_n)] d\pi + \log \left(\frac{2}{\varepsilon} \right).$$

By definition of $\hat{\rho}_\lambda$, we have:

$$\begin{aligned} \left| \lambda - g(\lambda, n) \right| \left[\int R d\hat{\rho}_\lambda - \bar{R} \right] \\ \leq \inf_{\rho \in \mathcal{F}} \left\{ \mathcal{K}(\rho, \hat{\rho}_\lambda) - \log \int \exp[-\lambda(r_n - \bar{r}_n)] d\pi + \log \left(\frac{2}{\varepsilon} \right) \right\}. \end{aligned}$$

Then, apply (8) again to get:

$$\left| \lambda - g(\lambda, n) \right| \left[\int R d\hat{\rho}_\lambda - \bar{R} \right] \leq \inf_{\rho \in \mathcal{F}} \left\{ \lambda \int (r_n - \bar{r}_n) d\rho + \mathcal{K}(\rho, \pi) + \log \left(\frac{2}{\varepsilon} \right) \right\}.$$

Plug (16) into the right-hand side to get

$$\begin{aligned} \left| \lambda - g(\lambda, n) \right| \left[\int R d\hat{\rho}_\lambda - \bar{R} \right] \\ \leq \inf_{\rho \in \mathcal{F}} \left\{ \left[\lambda + g(\lambda, n) \right] \int (R - \bar{R}) d\rho + 2\mathcal{K}(\rho, \pi) + 2 \log \left(\frac{2}{\varepsilon} \right) \right\}. \end{aligned}$$

□

A.4 Proofs of Section 5

Proof of Lemma 1. Combine Theorem 2.1 p. 25 and Lemma 2.2 p. 27 in Boucheron et al. (2013). □

Proof of Lemma 2. Apply Theorem 2.10 in Boucheron et al. (2013), and plug the margin assumption. □

Proof of Corollary 5.2. We remind that thanks to (6) it is enough to prove the claim for \mathcal{F}_1 . We apply Theorem 4.2 to get:

$$\begin{aligned} \mathcal{B}_\lambda(\mathcal{F}_1) &= \inf_{(\mathbf{m}, \sigma^2)} \left\{ \int R d\Phi_{\mathbf{m}, \sigma^2} + \frac{\lambda}{n} + 2 \frac{\mathcal{K}(\Phi_{\mathbf{m}, \sigma^2}, \pi) + \log \left(\frac{2}{\varepsilon} \right)}{\lambda} \right\} \\ &= \inf_{(\mathbf{m}, \sigma^2)} \left\{ \int R d\Phi_{\mathbf{m}, \sigma^2} + \frac{\lambda}{n} + 2 \frac{d \left[\frac{1}{2} \log \left(\frac{\theta^2}{\sigma^2} \right) + \frac{\sigma^2}{2\theta^2} \right] + \frac{\|\mathbf{m}\|^2}{2\theta^2} - \frac{d}{2} + \log \left(\frac{2}{\varepsilon} \right)}{\lambda} \right\}. \end{aligned}$$

Note that the minimizer of $R, \bar{\theta}$, is not unique (because $f_\theta(x)$ does not depend on $\|\theta\|$) and we can chose it in such a way that $\|\bar{\theta}\| = 1$. Then

$$\begin{aligned} R(\theta) - \bar{R} &= \mathbb{E} \left[\mathbf{1}_{\langle \theta, X \rangle Y < 0} - \mathbf{1}_{\langle \bar{\theta}, X \rangle Y < 0} \right] \leq \mathbb{E} \left[\mathbf{1}_{\langle \theta, X \rangle \langle \bar{\theta}, X \rangle < 0} \right] \\ &= \mathbb{P} \left(\langle \theta, X \rangle \langle \bar{\theta}, X \rangle < 0 \right) \leq c \frac{\theta}{\|\theta\|} - \bar{\theta} \leq 2c \|\theta - \bar{\theta}\|. \end{aligned}$$

So:

$$\begin{aligned} \mathcal{B}_\lambda(\mathcal{F}_1) &\leq \bar{R} + \inf_{(\mathbf{m}, \sigma^2)} \left\{ 2c \int \|\theta - \bar{\theta}\| \Phi_{\mathbf{m}, \sigma^2}(d\theta) \right. \\ &\quad \left. + \frac{\lambda}{n} + 2 \frac{d \left[\frac{1}{2} \log \left(\frac{\theta^2}{\sigma^2} \right) + \frac{\sigma^2}{2\theta^2} \right] + \frac{\|\mathbf{m}\|^2}{2\theta^2} - \frac{d}{2} + \log \left(\frac{2}{\varepsilon} \right)}{\lambda} \right\}. \end{aligned}$$

We now restrict the infimum to distributions ν such that $\mathbf{m} = \bar{\theta}$:

$$\mathcal{B}(\mathcal{F}_1) \leq \bar{R} + \inf_{\sigma^2} \left\{ 2c \sqrt{d} \sigma + \frac{\lambda}{n} + \frac{d \log \left(\frac{\theta^2}{\sigma^2} \right) + \frac{d\sigma^2}{\theta^2} - d + 2 \log \left(\frac{2}{\varepsilon} \right)}{\lambda} \right\}.$$

We put $\sigma = \frac{1}{\lambda}$ and substitute $\frac{1}{\sqrt{d}}$ for ϑ to get

$$\mathcal{B}(\mathcal{F}_1) \leq \bar{R} + \frac{\lambda}{n} \frac{c\sqrt{d} + d \log(4\frac{\lambda^2}{d}) + r_n^2 + \frac{d^2}{4\lambda^2} + 2 \log\left(\frac{2}{\varepsilon}\right)}{\lambda}.$$

Substitute \sqrt{nd} for λ to get the desired result. \square

Proof of Corollary 5.3. We apply Theorem 4.3:

$$\begin{aligned} & \int (R - \bar{R}) d\bar{\rho}_\lambda \\ & \leq \inf_{\mathbf{m}, \sigma^2} \left\{ \frac{\lambda + g(\lambda, n)}{\lambda - g(\lambda, n)} \int (R - \bar{R}) d\Phi_{\mathbf{m}, \sigma^2} + \frac{1}{\lambda - g(\lambda, n)} \left(2\mathcal{K}(\Phi_{\mathbf{m}, \sigma^2}, \pi) + 2 \log\left(\frac{2}{\varepsilon}\right) \right) \right\} \end{aligned}$$

where $\lambda < \frac{2n}{c_{\mathcal{F}_1}}$. Computations similar to those in the proof of Corollary 5.2 lead to

$$\begin{aligned} \int R d\bar{\rho}_\lambda & \leq \bar{R} + \inf_{\mathbf{m}, \sigma^2} \left\{ \frac{\lambda + g(\lambda, n)}{2c} \frac{\lambda + g(\lambda, n)}{\lambda - g(\lambda, n)} \int \|\theta - \bar{\theta}\| \Phi_{\mathbf{m}, \sigma^2}(d\theta) \right. \\ & \quad \left. + 2 \frac{\frac{1}{2} \sum_{j=1}^d \left[\log\left(\frac{\vartheta^2}{\sigma^2}\right) + \frac{\sigma^2}{\vartheta^2} \right] + \frac{\|\mathbf{m}\|^2}{2\vartheta^2} - \frac{d}{2} + \log\left(\frac{2}{\varepsilon}\right)}{\lambda - g(\lambda, n)} \right\}. \end{aligned}$$

taking $\mathbf{m} = \bar{\theta}$ and $\lambda = \frac{2n}{c_{\mathcal{F}_2}}$, we get the result. \square

A.5 Proofs of Section 6

Proof of Lemma 3. For fixed θ we can upper bound the individual risk such that:

$$0 \leq \max(0, 1 - \langle \theta, X_i \rangle) \leq 1 + \langle \theta, X_i \rangle$$

such that we can apply Hoeffding's inequality conditionally on X_i and fixed θ .

We get,

$$\begin{aligned} \mathbb{E}[\exp(\lambda(R^H - r_n^H)) | X_1, \dots, X_n] & \leq \exp \left\{ \frac{\lambda^2}{8n^2} \sum_{i=1}^n (1 + \langle \theta, X_i \rangle)^2 \right\} \\ & \leq \exp \left\{ \frac{\lambda^2}{4n} + \frac{\lambda^2 c_x^2}{4n} \|\theta\|^2 \right\} \end{aligned}$$

where the last inequality stems from the fact that $(a + b)^2 \leq 2(a^2 + b^2)$ and the fact that we have supposed the X_i to be bounded. We can take the expectation of this term with

respect to the X_i 's and with respect to our Gaussian prior.

$$\begin{aligned} \pi \{ \mathbb{E}[\exp(\lambda(R^H - r_n^H))] \} & \leq \frac{\exp\left(\frac{\lambda^2}{4n}\right)}{(2\pi)^{\frac{d}{2}} \sqrt{\vartheta^2}} \int \exp\left(\frac{\lambda^2 c_x^2}{4n} \|\theta\|^2 - \frac{1}{2\vartheta^2} \|\theta\|^2\right) d\theta \\ & \leq \frac{\exp\left(\frac{\lambda^2}{4n}\right)}{(2\pi)^{\frac{d}{2}} \sqrt{\vartheta^2}} \int \exp\left(-\frac{1}{2} \left[\frac{1}{\vartheta^2} - \frac{\lambda^2 c_x^2}{2n} \right] \|\theta\|^2\right) d\theta \end{aligned}$$

The integral is a properly defined Gaussian integral under the hypothesis that $\frac{1}{\vartheta^2} - \frac{\lambda^2 c_x^2}{2n} > 0$ hence $\lambda < \frac{1}{c_x} \sqrt{\frac{2n}{d}}$. The integral is proportional to a Gaussian and we can directly write:

$$\pi \{ \mathbb{E}[\exp(\lambda(R^H - r_n^H))] \} \leq \frac{\exp\left(\frac{\lambda^2}{4n}\right)}{\sqrt{1 - \frac{\lambda^2 \lambda^2 c_x^2}{2n}}}$$

writing everything in the exponential gives the desired result. \square

Proof of Corollary 6.1. We apply Theorem 4.2 to get:

$$\begin{aligned} \mathcal{B}_\lambda(\mathcal{F}_1) & = \inf_{(\mathbf{m}, \sigma^2)} \left\{ \int R^H d\Phi_{\mathbf{m}, \sigma^2} + \frac{\lambda}{2n} - \frac{1}{\lambda} \log\left(1 - \frac{\vartheta^2 \lambda^2 c_x^2}{2n}\right) + 2 \frac{\mathcal{K}(\Phi_{\mathbf{m}, \sigma^2}, \pi) + \log\left(\frac{2}{\varepsilon}\right)}{\lambda} \right\} \\ & = \inf_{(\mathbf{m}, \sigma^2)} \left\{ \int R^H d\Phi_{\mathbf{m}, \sigma^2} + \frac{\lambda}{2n} - \frac{1}{\lambda} \log\left(1 - \frac{\vartheta \lambda^2 c_x^2}{2n}\right) \right. \\ & \quad \left. + 2 \frac{\frac{1}{2} \sum_{j=1}^d \left[\log\left(\frac{\vartheta^2}{\sigma^2}\right) + \frac{\sigma^2}{\vartheta^2} \right] + \frac{\|\mathbf{m}\|^2}{2\vartheta^2} - \frac{d}{2} + \log\left(\frac{2}{\varepsilon}\right)}{\lambda} \right\}. \end{aligned}$$

We use the fact that the hinge loss is Lipschitz and that the (X_i) are uniformly bounded $\|X\|_\infty \leq c_x$. We get $R^H(\theta) \leq R^H + c_x \sqrt{d} \|\theta - \bar{\theta}\|$ and restrict the infimum to distributions ν such that $m = \bar{\theta}$:

$$\mathcal{B}(\mathcal{F}_1) \leq \bar{R}^H + \inf_{\sigma^2} \left\{ c_x d \sigma^2 + \frac{\lambda}{2n} - \frac{1}{\lambda} \log\left(1 - \frac{\vartheta^2 \lambda^2 c_x^2}{2n}\right) + \frac{d \log\left(\frac{\vartheta^2}{\sigma^2}\right) + \frac{d\sigma^2}{\vartheta^2} + \frac{1}{\vartheta^2} - d + 2 \log\left(\frac{2}{\varepsilon}\right)}{\lambda} \right\}.$$

We specify $\sigma^2 = \frac{1}{\sqrt{dn}}$ and $\lambda = c_x \sqrt{\frac{2n}{d}}$ such that we get:

$$\mathcal{B}(\mathcal{F}_1) \leq \bar{R}^H + c_x \sqrt{\frac{d}{n}} + \frac{\sqrt{\vartheta^2}}{2c_x \sqrt{n}} - c_x \sqrt{\frac{\vartheta^2}{n}} \log\left(1 - \frac{1}{2}\right) + d \frac{c_x \vartheta}{\sqrt{n}} \log\left(\vartheta^2 \sqrt{nd}\right) + c_x \vartheta \frac{d}{\sqrt{nd}} + \frac{1}{\vartheta^2} - d + 2 \log\left(\frac{2}{\varepsilon}\right) \frac{1}{\sqrt{n}}.$$

To get the correct rate we take the prior variance to be $\vartheta^2 = \frac{1}{n}$ by replacing in the above equation we get the desired result.

□

Proof of Theorem 6.2. From Nesterov (2004) (th. 3.2.2) we have the following bound on the objective function minimized by VB, (the objective is not uniformly Lipschitz)

$$\rho^k(r_n^H) + \frac{1}{\lambda} \mathcal{K}(\rho^k, \pi) - \inf_{\rho \in \mathcal{F}_1} \left\{ \rho(r_n^H) + \frac{1}{\lambda} \mathcal{K}(\rho, \pi) \right\} \leq \frac{LM}{\sqrt{1+k}}. \quad (17)$$

We have from equation (11) specified for measures ρ^k probability $1 - \varepsilon$,

$$\lambda \int r_n^H d\rho^k \leq \lambda \int R^H d\rho^k + f(\lambda, n) + \mathcal{K}(\rho^k, \pi) + \log\left(\frac{1}{\varepsilon}\right)$$

Combining the two equations yields,

$$\int R^H d\rho^k \leq \frac{LM}{\sqrt{1+k}} + \frac{1}{\lambda} f(n, \lambda) + \inf_{\rho \in \mathcal{F}_1} \left\{ \rho(r_n^H) + \frac{1}{\lambda} \mathcal{K}(\rho, \pi) \right\} + \frac{1}{\lambda} \log \frac{1}{\varepsilon}$$

We can therefore write for any $\rho \in \mathcal{F}_1$,

$$\int R^H d\rho^k \leq \frac{LM}{\sqrt{1+k}} + \frac{1}{\lambda} f(n, \lambda) + \rho(r_n^H) + \frac{1}{\lambda} \mathcal{K}(\rho, \pi) + \frac{1}{\lambda} \log \frac{1}{\varepsilon}$$

Using equation (11) a second time we get with probability $1 - \varepsilon$

$$\int R^H d\rho^k \leq \frac{LM}{\sqrt{1+k}} + \frac{2}{\lambda} f(n, \lambda) + \rho(R^H) + \frac{2}{\lambda} \mathcal{K}(\rho, \pi) + \frac{2}{\lambda} \log \frac{2}{\varepsilon}$$

Because this is true for any $\rho \in \mathcal{F}_1$ in $1 - \varepsilon$ we can write the bound for the smallest measure in \mathcal{F}_1 .

$$\int R^H d\rho^k \leq \frac{LM}{\sqrt{1+k}} + \frac{2}{\lambda} f(n, \lambda) + \inf_{\rho \in \mathcal{F}_1} \left\{ \rho(R^H) + \frac{2}{\lambda} \mathcal{K}(\rho, \pi) \right\} + \frac{2}{\lambda} \log \frac{2}{\varepsilon}$$

By taking the Gaussian measure with variance $\frac{1}{n}$ and mean $\bar{\theta}$ in the infimum and taking $\lambda = \frac{1}{c_x} \sqrt{nd}$ and $\vartheta = \frac{1}{d}$, we can use the results of Corollary 6.1 to get the result. □

A.6 Proofs of Section 7

Proof of Lemma 4. The idea of the proof is to use Hoeffding's decomposition of U-statistics combined with Hoeffding's inequality for iid random variables. This was done in ranking by Clémengon et al. (2008), and later in Robbiano (2013); Ridgway et al. (2014) for ranking via aggregation and Bayesian statistics. The proof is as follows: we define

$$q_{i,j}^\theta = \mathbf{1}_{(Y_i - Y_j) f_\theta(X_i, X_j) < 0} - R(\theta)$$

so that

$$U_n := \frac{1}{n(n-1)} \sum_{i,j} q_{i,j}^\theta = r_n(\theta) - R(\theta).$$

From Hoeffding (1948) we have

$$U_n = \frac{1}{n!} \sum_{\pi} \frac{1}{\lfloor \frac{n}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} q_{\pi(i), \pi(i + \lfloor \frac{n}{2} \rfloor)}^\theta$$

where the sum is taken over all the permutations π of $\{1, \dots, n\}$. Jensen's inequality leads to

$$\begin{aligned} \mathbb{E} \exp[\lambda U_n] &= \mathbb{E} \exp \left[\lambda \frac{1}{n!} \sum_{\pi} \frac{1}{\lfloor \frac{n}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} q_{\pi(i), \pi(i + \lfloor \frac{n}{2} \rfloor)}^\theta \right] \\ &\leq \frac{1}{n!} \sum_{\pi} \mathbb{E} \exp \left[\frac{\lambda}{\lfloor \frac{n}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} q_{\pi(i), \pi(i + \lfloor \frac{n}{2} \rfloor)}^\theta \right]. \end{aligned}$$

We now use, for each of the terms in the sum we use the same argument as in the proof of Lemma 1 to get

$$\mathbb{E} \exp[\lambda U_n] \leq \frac{1}{n!} \sum_{\pi} \exp \left[\frac{\lambda^2}{2 \lfloor \frac{n}{2} \rfloor} \right] \leq \exp \left[\frac{\lambda^2}{n-1} \right]$$

(in the last step, we used $\lfloor \frac{n}{2} \rfloor \geq (n-1)/2$). We proceed in the same way to upper bound $\mathbb{E} \exp[-\lambda U_n]$. □

Proof of Lemma 5. As already done above, we use Bernstein inequality and Hoeffding decomposition. Fix θ . We define this time

$$q_{i,j}^\theta = \mathbf{1}_{\{(\theta, X_i - X_j)(Y_i - Y_j) < 0\}} - \mathbf{1}_{\{(\bar{\theta}, X_i - X_j)(Y_i - Y_j) < 0\}} - R(\theta) + \bar{R}$$

so that

$$U_n := r_n(\theta) - \bar{r}_n - R(\theta) + \bar{R} = \frac{1}{n(n-1)} \sum_{i \neq j} q_{i,j}^\theta.$$

Then,

$$U_n = \frac{1}{n!} \sum_{\pi} \frac{1}{\lfloor \frac{n}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} q_{\pi(i), \pi(i + \lfloor \frac{n}{2} \rfloor)}^\theta.$$

Jensen's inequality:

$$\begin{aligned} \mathbb{E} \exp[\lambda U_n] &= \mathbb{E} \exp \left[\lambda \frac{1}{n!} \sum_{\pi} \frac{1}{\lfloor \frac{n}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} q_{\pi^{(i)}, \pi^{(i+\lfloor \frac{n}{2} \rfloor)}^\theta} \right] \\ &\leq \frac{1}{n!} \sum_{\pi} \mathbb{E} \exp \left[\frac{\lambda}{\lfloor \frac{n}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} q_{\pi^{(i)}, \pi^{(i+\lfloor \frac{n}{2} \rfloor)}^\theta} \right]. \end{aligned}$$

Then, for each of the terms in the sum, use Bernstein's inequality:

$$\mathbb{E} \exp \left[\frac{\lambda}{\lfloor \frac{n}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} q_{\pi^{(i)}, \pi^{(i+\lfloor \frac{n}{2} \rfloor)}^\theta} \right] \leq \exp \left[\frac{\mathbb{E}((q_{\pi^{(1)}, \pi^{(1+\lfloor \frac{n}{2} \rfloor)}^\theta}^\theta)^2) \frac{\lambda^2}{\lfloor \frac{n}{2} \rfloor}}{2 \left(1 - 2 \frac{\lambda}{\lfloor \frac{n}{2} \rfloor}\right)} \right].$$

We use again $\lfloor \frac{n}{2} \rfloor \geq (n-1)/2$. Then, as the pairs (X_i, Y_i) are iid, we have $\mathbb{E}((q_{\pi^{(1)}, \pi^{(1+\lfloor \frac{n}{2} \rfloor)}^\theta}^\theta)^2) = \mathbb{E}((q_{1,2}^\theta)^2)$ and then $\mathbb{E}((q_{1,2}^\theta)^2) \leq C[R(\theta) - \bar{R}]$ thanks to the margin assumption. So

$$\mathbb{E} \exp \left[\frac{\lambda}{\lfloor \frac{n}{2} \rfloor} \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} q_{\pi^{(i)}, \pi^{(i+\lfloor \frac{n}{2} \rfloor)}^\theta} \right] \leq \exp \left[\frac{C[R(\theta) - \bar{R}] \frac{\lambda^2}{n-1}}{\left(1 - \frac{4\lambda}{n-1}\right)} \right].$$

This ends the proof of the proposition. \square

Proof of Corollary 7.2. The calculations are similar to the ones in the proof of Corollary 5.2 so we don't give the details. Note that when we reach

$$\mathcal{B}_\lambda(\mathcal{F}_1) \leq \bar{R} + \frac{2\lambda}{n-1} + \frac{c\sqrt{d} + d \log(2\lambda) + \frac{d}{4\lambda^2} + 2 \log \left(\frac{2e}{\varepsilon} \right)}{\lambda},$$

an approximate minimization with respect to λ leads to the choice $\lambda = \sqrt{\frac{d(n-1)}{2}}$. \square

A.7 Proofs of Section 8

Proof. First, note that, for any ρ ,

$$\begin{aligned} \mathcal{K}(\rho, \pi_\delta) &= \beta \int (R - \bar{R}) d\rho + \mathcal{K}(\rho, \pi) + \log \int \exp[-\beta(R - \bar{R})] d\pi \\ &\leq \beta \int (R - \bar{R}) d\rho + \mathcal{K}(\rho, \pi). \end{aligned}$$

Now, we define a subset of \mathcal{F} that will be used for the calculation of the bound. We define for $\delta > 0$ the probability distribution $\rho_{U,V,\delta}(d\theta)$ as π conditioned to $\theta = \mu^T$ with μ is

uniform on $\{V(i, \ell), |\mu_{i,\ell} - U_{i,\ell}| \leq \delta\}$ and ν is uniform on $\{V(i, \ell), |v_{i,\ell} - V_{j,\ell}| \leq \delta\}$. Note that

$$\begin{aligned} \int (R - \bar{R}) d\rho_{M,N,\delta} &= \int \mathbb{E}((\theta_X - M_X)^2) \rho_{U,V,\delta}(d\theta) \\ &\leq \int 3\mathbb{E}(((UV^T)_X - M_X)^2) \rho_{U,V,\delta}(d(\mu, \nu)) \\ &\quad + 3 \int \mathbb{E}(((UV^T)_X - (UV^T)_X)^2) \rho_{U,V,\delta}(d(\mu, \nu)) \\ &\quad + 3 \int \mathbb{E}(((\mu^T)_X - (U\nu^T)_X)^2) \rho_{U,V,\delta}(d(\mu, \nu)). \end{aligned}$$

By definition, the first term is 0. Moreover:

$$\begin{aligned} &\int \mathbb{E}(((UV^T)_X - (UV^T)_X)^2) \rho_{U,V,\delta}(d(\mu, \nu)) \\ &= \int \frac{1}{m_1 m_2} \sum_{i,j} \left[\sum_k U_{i,k} (v_{j,k} - V_{j,k}) \right]^2 \rho_{U,V,\delta}(d(\mu, \nu)) \\ &\leq \int \frac{1}{m_1 m_2} \sum_{i,j} \left[\sum_k U_{i,k}^2 \right] \left[\sum_k (v_{j,k} - V_{j,k})^2 \right] \rho_{U,V,\delta}(d(\mu, \nu)) \\ &\leq K r C^2 \delta^2. \end{aligned}$$

In the same way,

$$\begin{aligned} \int \mathbb{E}(((\mu^T)_X - (U\nu^T)_X)^2) \rho_{U,V,\delta}(d(\mu, \nu)) &\leq \int \|\mu - U\|_F^2 \|\nu\|_F^2 \rho_{U,V,\delta}(d(\mu, \nu)) \\ &\leq K r (C + \delta)^2 \delta^2. \end{aligned}$$

So:

$$\int (R - \bar{R}) d\rho_{M,N,\delta} \leq 2K r \delta^2 (C + \delta^2).$$

Now, let us consider the term $\mathcal{K}(\rho_{U,V,\delta}, \pi)$. An explicit calculation is possible but tedious. Instead, we might just introduce the set $G_\delta = \{\theta = \mu^T, \|\mu - U\|_F \leq \delta, \|\nu - V\|_F \leq \delta\}$ and note that $\mathcal{K}(\rho_{U,V,\delta}, \pi) \leq \log \frac{1}{\pi(G_\delta)}$. An upper bound for G_δ is calculated page 317-320 in Alquier (2014) and the result is given by (10) in this reference:

$$\begin{aligned} \mathcal{K}(\rho_{U,V,\delta}, \pi) &\leq 4\delta^2 + 2\|U\|_F^2 + 2\|V\|_F^2 + 2 \log(2) \\ &\quad + (m_1 + m_2) r \log \left(\frac{1}{\delta} \sqrt{\frac{3\pi(m_1 \vee m_2)K}{4}} \right) + 2K \log \left(\frac{\Gamma(a) 3^{a+1} \exp(2)}{b^{a+1} 2^a} \right) \end{aligned}$$

as soon as the restriction $b \leq \frac{\delta^2}{2m_1 K \log(2m_1 K)}, \frac{\delta^2}{2m_2 K \log(2m_2 K)}$ is satisfied. So we obtain:

$$\begin{aligned} \mathcal{K}(\rho_{U,V,\delta}, \pi_\beta) &\leq \beta^2 K r \delta^2 (C + \delta^2) + 4\delta^2 + 2\|U\|_F^2 + 2\|N\|_F^2 + 2\log(2) \\ &\quad + (m_1 + m_2)r \log\left(\frac{1}{\delta} \sqrt{\frac{3\pi(m_1 \vee m_2)K}{4}}\right) + 2K \log\left(\frac{\Gamma(a)3^{a+1} \exp(2)}{b^{a+1} 2^a}\right). \end{aligned}$$

Note that $\|U\|_F^2 \leq C^2 r m_1$, $\|V\|_F^2 \leq C^2 r m_2$ and $K \leq m_1 + m_2$ so it is clear that the choice $\delta = \sqrt{\frac{1}{\beta}}$ and $b \leq \frac{1}{2\beta(m_1 \vee m_2) \log(2K(m_1 \vee m_2))}$ leads to the existence of a constant $\mathcal{C}(a, C)$ such that

$$\mathcal{K}(\rho_{U,V,\delta}, \pi_\beta) \leq \mathcal{C}(a, C) \left\{ r(m_1 + m_2) \log[\beta h(m_1 + m_2)K] + \frac{1}{\beta} \right\}.$$

□

Appendix B. Implementation details

B.1 Sequential Monte Carlo

Tempering SMC approximates iteratively a sequence of distribution ρ_{λ_t} , with

$$\rho_{\lambda_t}(d\theta) = \frac{1}{Z_t} \exp(-\lambda_t r_n(\theta)) \pi(d\theta),$$

and temperature ladder $\lambda_0 = 0 < \dots < \lambda_T = \lambda$. The pseudo-code below is given for an adaptive sequence of temperatures.

Algorithm 1 Tempering SMC

Input N (number of particles), $\tau \in (0, 1)$ (ESS threshold), $\kappa > 0$ (random walk tuning parameter)

Init. Sample $\theta_0^i \sim \pi_\xi(\theta)$ for $i = 1$ to N , set $t \leftarrow 1$, $\lambda_0 = 0$, $Z_0 = 1$.

Loop a. Solve in λ_t the equation

$$\frac{\{\sum_{i=1}^N w_t(\theta_{t-1}^i)\}^2}{\sum_{i=1}^N \{w_t(\theta_{t-1}^i)\}^2} = \tau N, \quad w_t(\theta) = \exp[-(\lambda_t - \lambda_{t-1})r_n(\theta)] \quad (18)$$

using bisection search. If $\lambda_t \geq \lambda_T$, set $Z_T = Z_{t-1} \times \left\{ \frac{1}{N} \sum_{i=1}^N w_t(\theta_{t-1}^i) \right\}$, and stop.

b. Resample: for $i = 1$ to N , draw A_t^i in $1, \dots, N$ so that $\mathbb{P}(A_t^i = j) = w_t(\theta_{t-1}^j) / \sum_{k=1}^N w_t(\theta_{t-1}^k)$; see Algorithm 2 in the appendix.

c. Sample $\theta_t^i \sim M_t(\theta_{t-1}^{A_t^i}, d\theta)$ for $i = 1$ to N where M_t is a MCMC kernel that leaves invariant π_t ; see comments below.

d. Set $Z_t = Z_{t-1} \times \left\{ \frac{1}{N} \sum_{i=1}^N w_t(\theta_{t-1}^i) \right\}$.

The algorithm outputs a weighted sample (w_T^i, θ_T^i) approximately distributed as target posterior, and an unbiased estimator of the normalizing constant Z_{λ_T} .

Step **b.** of algorithm B.1 depends of a resampling algorithm. We choose to use Systematic resampling, see Algorithm 2.

Algorithm 2 Systematic resampling

Input: Normalised weights $W_i^t := w_i(\theta_{t-1}^i) / \sum_{k=1}^N w_k(\theta_{t-1}^i)$.

Output: indices $A^t \in \{1, \dots, N\}$, for $t = 1, \dots, N$.

- a. Sample $U \sim \mathcal{U}([0, 1])$.
- b. Compute cumulative weights as $C^n = \sum_{m=1}^n NW^m$.
- c. Set $s \leftarrow U$, $m \leftarrow 1$.
- d. **For** $n = 1 : N$

While $C^m < s$ **do** $m \leftarrow m + 1$.

$A^n \leftarrow m$, and $s \leftarrow s + 1$.

End For

For the MCMC step, we used a Gaussian random-walk Metropolis kernel, with a covariance matrix for the random step that is proportional to the empirical covariance matrix of the current set of simulations.

B.2 Optimizing the bound

A natural idea to find a global optimum of the objective is to try to solve a sequence of local optimization problems with increasing temperatures. For $\gamma = 0$ the problem can be solved exactly (as a KL divergence between two Gaussians). Then, for two consecutive temperatures, the corresponding solutions should be close enough.

This idea has been coined under several names. It has a long history in variational inference under the name ‘deterministic annealing’; see e.g. Yuille (2010) for an application to Markov random fields. In addition the intermediate results can be of interest in our case for selecting the temperature. One can compute the bound at almost no additional cost as a function of the current risk. In turns this can be used to monitor the bound.

Algorithm 3 Deterministic annealing

Input $(\lambda_t)_{t \in [0, T]}$ a sequence of temperature

Init. Set $m = 0$ and $\Sigma = \vartheta I_d$, the values minimizing KL-divergence for $\lambda = 0$

Loop $t=1, \dots, T$

- a. $m^{\lambda_t}, \Sigma^{\lambda_t} = \text{Minimize } \mathcal{L}^{\lambda_t}(m, \Sigma)$ using some local optimization routine with initial points $m^{\lambda_{t-1}}, \Sigma^{\lambda_{t-1}}$
- b. Break if the empirical bound increases.

End Loop

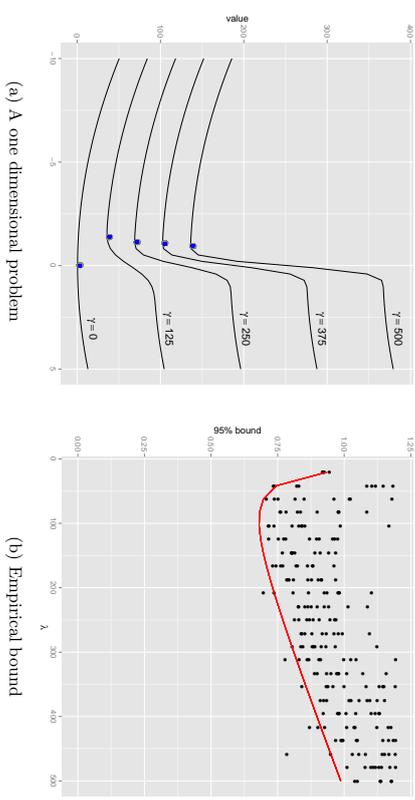


Figure 2: Deterministic annealing on a Pima Indians with one covariate and full model resp.

The right panel gives the empirical bound obtained for the DA method (in red). The dots are the results of direct global optimization based on L-BFGS algorithms (with starting values drawn from the prior). Each optimization problem is repeated 20 times.

We find that using a deterministic annealing algorithm with a limited amount of steps helps in finding a high enough optimum. On the left panel of Figure 2, we can see the one

dimensional case where the initial problem $\gamma = 0$ corresponds to a convex minimization problem and where the increasing temperature gradually complexifies the optimization problem. Figure 2 shows that the solution given by DA is in average lower than randomly initialized optimization.

Appendix C. Stochastic gradient descent

The stochastic gradient descent algorithm used in Section 7 is described as Algorithm 4.

Algorithm 4 Stochastic Gradient Descent

Input B a batch size, an unbiased estimator of the gradient $\tilde{\nabla}_B f$, $\eta \in (0, 1)$ and c

While \neg converged

a. $x_{t+1} = x_t - \lambda_t \tilde{\nabla}_B f(x_t)$

b. Update $\lambda_{t+1} = \frac{1}{(t+\phi)^\eta}$

End Loop

In all our experiments we take $c = 1$ and $\eta = 0.9$.

References

- P. Alquier. Bayesian methods for low-rank matrix estimation: short survey and theoretical study. In S. Jain, R. Munos, F. Stephan, and T. Zeugmann, editors, *Algorithmic Learning Theory*. Springer - Lecture Notes in Artificial Intelligence, 2014.
- P. Alquier and G. Biau. Sparse single-index model. *Journal of Machine Learning Research*, 14(1):243–280, 2013.
- P. Alquier and X. Li. Prediction of quantiles by statistical learning and application to GDP forecasting. In J.-G. Ganascia, P. Lencea, and J.-M. Petit, editors, *Discovery Science*. Springer - Lecture Notes in Artificial Intelligence, 2012.
- J.-Y. Audibert and O. Catoni. Robust linear least squares regression. *Ann. Statist.*, 39(5):2766–2794, 10 2011. doi: 10.1214/11-AOS918. URL <http://dx.doi.org/10.1214/11-AOS918>.
- J. Bennett and S. Lanning. The netflix prize. In *Proceedings of KDD Cup and Workshop 07*, 2007.
- C. M. Bishop. *Pattern Recognition and Machine Learning*, chapter 10. Springer, 2006.

- P. Bissiri, C. Holmes, and S. Walker. A general framework for updating belief distributions. *arXiv preprint arXiv:1306.6430*, 2013.
- S. Boucheron, G. Lugosi, and P. Massart. *Concentration Inequalities*. Oxford University Press, 2013.
- R. H. Byrd, S. L. Hansen, J. Nocedal, and Y. Singer. A stochastic quasi-Newton method for large-scale optimization. *arXiv preprint arXiv:1401.7020*, 2014.
- E. J. Candès and T. Tao. The power of convex relaxation: near-optimal matrix completion. *IEEE Trans. Inform. Theory*, 56(5):2053–2080, 2010. ISSN 0018-9448. doi: 10.1109/TIT.2010.2044061. URL <http://dx.doi.org/10.1109/TIT.2010.2044061>.
- O. Catoni. *Statistical learning theory and stochastic optimization*, volume 1851 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 2004. Lecture notes from the 31st Summer School on Probability Theory held in Saint-Flour, July 8–25, 2001.
- O. Catoni. *PAC-Bayesian supervised classification: the thermodynamics of statistical learning*. Institute of Mathematical Statistics Lecture Notes—Monograph Series, 56. Institute of Mathematical Statistics, Beachwood, OH, 2007.
- O. Catoni. Challenging the empirical mean and empirical variance: A deviation study. *Ann. Inst. H. Poincaré Probab. Statist.*, 48(4):1148–1185, 11 2012. doi: 10.1214/11-AHP454. URL <http://dx.doi.org/10.1214/11-AHP454>.
- V. Chernozhukov and H. Hong. An MCMC approach to classical estimation. *Journal of Econometrics*, 115(2):293–346, 2003.
- S. Clémentçon, G. Lugosi, and N. Vayatis. Ranking and empirical minimization of U-statistics. *Ann. Stat.*, 36(2):844–874, 2008.
- A. S. Dalalyan and A. B. Tsybakov. Aggregation by exponential weighting, sharp PAC-Bayesian bounds and sparsity. *Machine Learning*, 72:39–61, 2008.
- A. S. Dalalyan and A. B. Tsybakov. Sparse regression learning by aggregation and Langevin Monte-Carlo. *Journal of Computer and System Science*, 78(5):1423–1443, 2012.
- P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *J. R. Statist. Soc. B*, 68(3):411–436, 2006. ISSN 1467-9868.
- P. J. Green, K. Latuszynski, M. Pereyra, and C. P. Robert. Bayesian computation: a perspective on the current state, and sampling backwards and forwards. Preprint arXiv:1502.01148, 2015.
- B. Guedj and P. Alquier. PAC-Bayesian estimation and prevision in sparse additive models. *Electronic Journal of Statistics*, 7:264–291, 2013.

- W. Hoeffding. Probability inequalities for sums of random variables. *Annals of Mathematical Statistics*, 10:293–325, 1948.
- M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.
- W. Jiang and M. A. Tanner. Gibbs posterior for variable selection in high-dimensional classification and data mining. *The Annals of Statistics*, 36(5):2207–2231, 2008.
- M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, (37):183–233, 1999.
- M. E. Khan. Decoupled variational Gaussian inference. In *Advances in Neural Information Processing Systems*, pages 1547–1555, 2014.
- M. E. Khan, A. Aravkin, M. Friedlander, and M. Seeger. Fast dual variational inference for non-conjugate latent gaussian models. In *Proceedings of The 30th International Conference on Machine Learning*, pages 951–959, 2013.
- Y. Kolchinskii, K. Lounici, and A. B. Tsybakov. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *The Annals of Statistics*, 39(5):2302–2329, 2011.
- N. D. Lawrence and R. Urtasun. Non-linear matrix factorization with Gaussian processes. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 601–608. ACM, 2009.
- G. Lecué. Méthodes d’agrégation: optimalité et vitesses rapides. Ph.D. thesis, Université Paris 6, 2007.
- F. Leisch and E. Dimitriadou. mlbench: Machine learning benchmark problems. *R package version*, 2:1–1, 2010.
- Y. J. Lin and Y. W. Teh. Variational Bayesian approach to movie rating prediction. *Proceedings of KDD Cup and Workshop*, 7:15–21, 2007.
- D. J. C. Mackay. *Information theory, inference and learning algorithms*. Cambridge University Press, 2002.
- T. T. Mai and P. Alquier. A Bayesian approach for matrix completion: optimal rate under general sampling distribution. *Electronic Journal of Statistics*, 9:823–841, 2015.
- E. Mannen and A. Tsybakov. Smooth discrimination analysis. *The Annals of Statistics*, 27(6):1808–1829, 1999.
- D. A. McAllester. PAC-Bayesian model averaging. In *Proceedings of the Twelfth Annual Conference On Computational Learning Theory, Santa Cruz, California (Electronic)*, pages 164–170. ACM, New-York, 1999.
- D. A. McAllester. Some PAC-Bayesian theorems. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 230–234. ACM, New York, 1998.
- Y. Nesterov. *Introductory lectures on convex optimization*, volume 87. Springer Science & Business Media, 2004.
- M. Opper and C. Archambeau. The variational Gaussian approximation revisited. *Neural computation*, 21(3):786–792, 2009.
- G. Parisi. *Statistical field theory*. Addison-Wesley, New-York, 1988.
- J. Ridgway, P. Alquier, N. Chopin, and F. Liang. PAC-Bayesian AUC classification and scoring. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 658–666. Curran Associates Inc., 2014.
- S. Robbiano. Upper bounds and aggregation in bipartite ranking. *Electronic Journal of Statistics*, 7:1249–1271, 2013.
- H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.
- R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proceedings of the 25th international conference on Machine learning*, pages 880–887. ACM, 2008.
- J. Shawe-Taylor and R.C. Williamson. A PAC analysis of a Bayesian estimator. In *Proceedings of the tenth annual conference on Computational learning theory*, pages 2–9. ACM, 1997.
- T. Suzuki. Convergence rate of Bayesian tensor estimator: Optimal rate without restricted strong convexity. arXiv preprint arXiv:1408.3092 (accepted by ICML2015), 2014.
- A. Tsybakov. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1):135–166, 2004.
- O. Wintenberger. Deviation inequalities for sums of weakly dependent time series. *Electronic Communications in Probability*, 15:489–503, 2010.
- Y. Yang. Aggregating regression procedures to improve performance. *Bernoulli*, 10:25–47, 2004.

- A. Yuille. Belief propagation, mean-field and the Bethe approximation. Technical report, Dept. Statistics UCLA, 2010.
- T. Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, pages 56–85, 2004.
- T. Zhang. Information theoretical upper and lower bounds for statistical estimation. *IEEE Transaction on Information Theory*, 52:1307–1321, 2006.
- M. Zhou, C. Wang, M. Chen, J. Paisley, D. Dunson, and L. Carin. Nonparametric bayesian matrix completion. *Proc. IEEE SAM*, 2010.